BigData

LAB – I

Submitted by

Mahesh Vemula

16158759

Part 1:

Group Members:

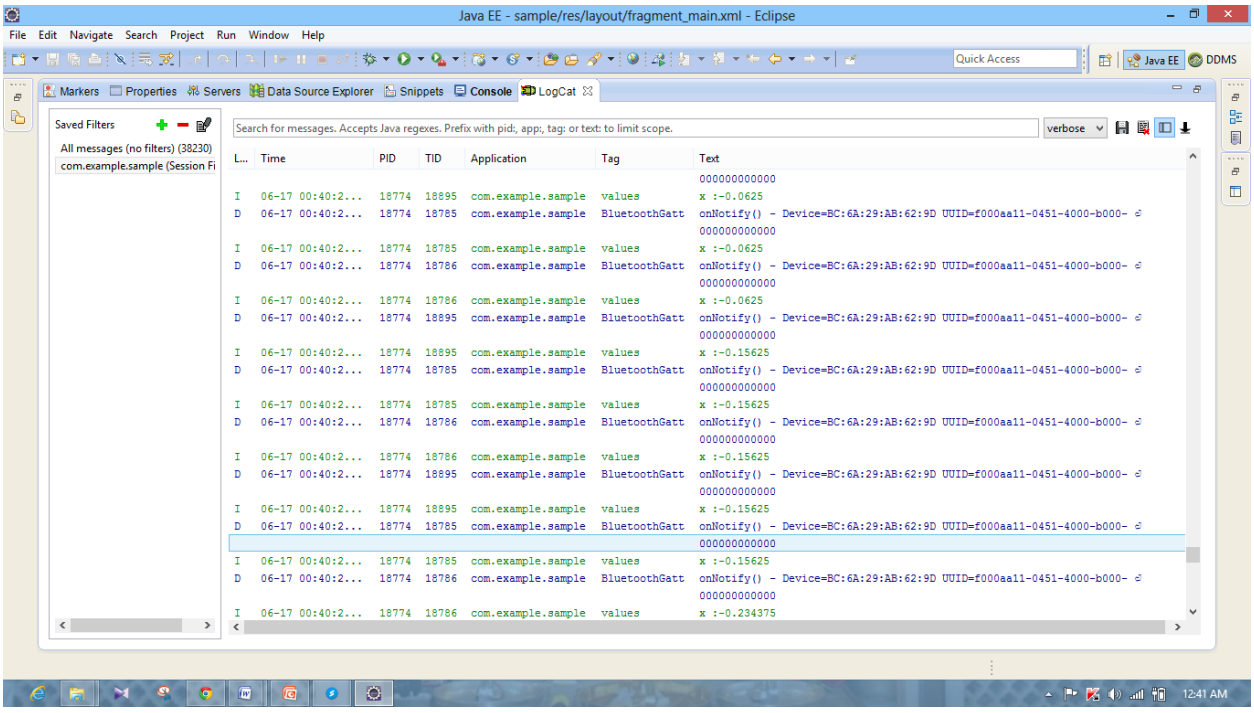
- **Deshpande,Aditya**
- **Meka,Tej Kiran**
- **Jagadish Rao**
- **Mahesh Vemula**

Task1: Android application using TI sensor tag.

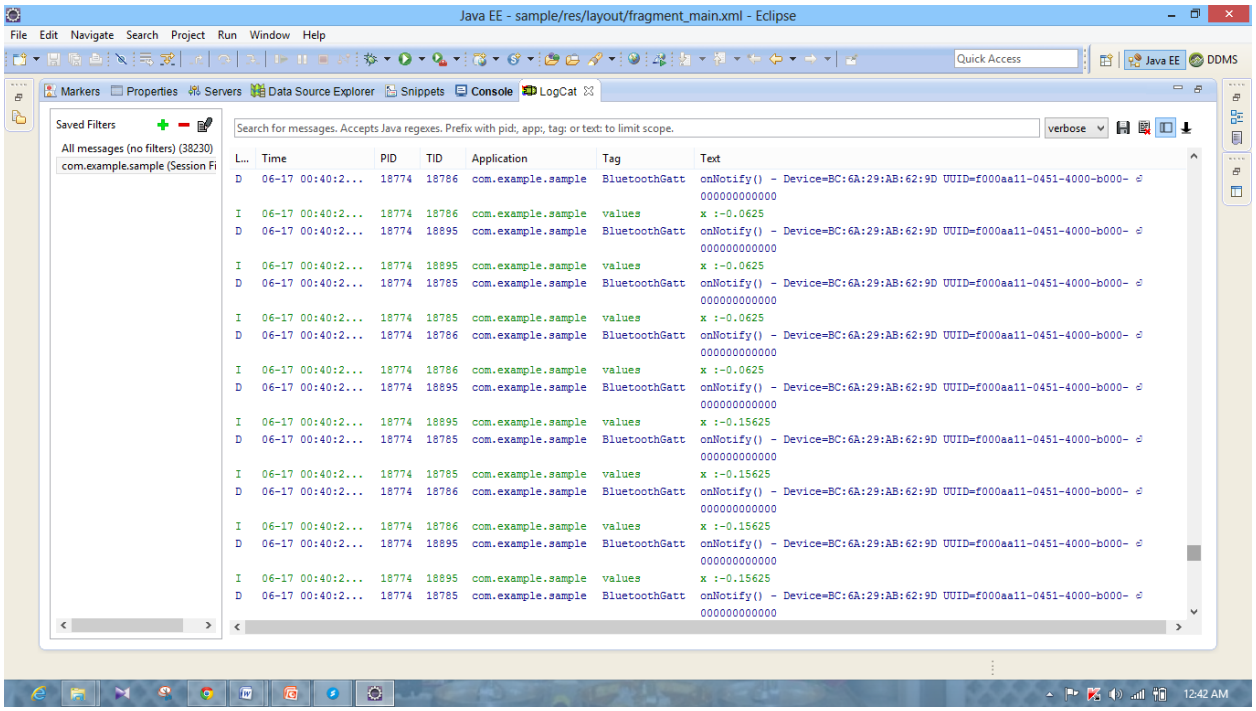
Step 1: Exatract the application from download file.

Step 2: Run the application on the device after enbling debugging mode.

Step 3: Observe the output in eclipse and take the screen shots of result.



Output showing the results of sensor values


Java EE - sample/res/layout/fragment_main.xml - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Java EE
DDMS
Markers Properties Servers Data Source Explorer Snippets Console LogCat
Saved Filters
All messages (no filters) (38230)
com.example.sample (Session Fi
Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.
verbose
L... Time PID TID Application Tag Text
D 06-17 00:40:2... 18774 18786 com.example.sample BluetoothGatt onNotify() - Device=BC:6A:29:AB:62:9D UUID=f000aa11-0451-4000-b000- 000000000000
I 06-17 00:40:2... 18774 18786 com.example.sample values x :-0.0625
D 06-17 00:40:2... 18774 18895 com.example.sample BluetoothGatt onNotify() - Device=BC:6A:29:AB:62:9D UUID=f000aa11-0451-4000-b000- 000000000000
I 06-17 00:40:2... 18774 18895 com.example.sample values x :-0.0625
D 06-17 00:40:2... 18774 18785 com.example.sample BluetoothGatt onNotify() - Device=BC:6A:29:AB:62:9D UUID=f000aa11-0451-4000-b000- 000000000000
I 06-17 00:40:2... 18774 18785 com.example.sample values x :-0.0625
D 06-17 00:40:2... 18774 18786 com.example.sample BluetoothGatt onNotify() - Device=BC:6A:29:AB:62:9D UUID=f000aa11-0451-4000-b000- 000000000000
I 06-17 00:40:2... 18774 18786 com.example.sample values x :-0.0625
D 06-17 00:40:2... 18774 18895 com.example.sample BluetoothGatt onNotify() - Device=BC:6A:29:AB:62:9D UUID=f000aa11-0451-4000-b000- 000000000000
I 06-17 00:40:2... 18774 18895 com.example.sample values x :-0.15625
D 06-17 00:40:2... 18774 18785 com.example.sample BluetoothGatt onNotify() - Device=BC:6A:29:AB:62:9D UUID=f000aa11-0451-4000-b000- 000000000000
I 06-17 00:40:2... 18774 18785 com.example.sample values x :-0.15625
D 06-17 00:40:2... 18774 18786 com.example.sample BluetoothGatt onNotify() - Device=BC:6A:29:AB:62:9D UUID=f000aa11-0451-4000-b000- 000000000000
I 06-17 00:40:2... 18774 18786 com.example.sample values x :-0.15625
D 06-17 00:40:2... 18774 18895 com.example.sample BluetoothGatt onNotify() - Device=BC:6A:29:AB:62:9D UUID=f000aa11-0451-4000-b000- 000000000000
I 06-17 00:40:2... 18774 18895 com.example.sample values x :-0.15625
D 06-17 00:40:2... 18774 18785 com.example.sample BluetoothGatt onNotify() - Device=BC:6A:29:AB:62:9D UUID=f000aa11-0451-4000-b000- 000000000000
12:42 AM

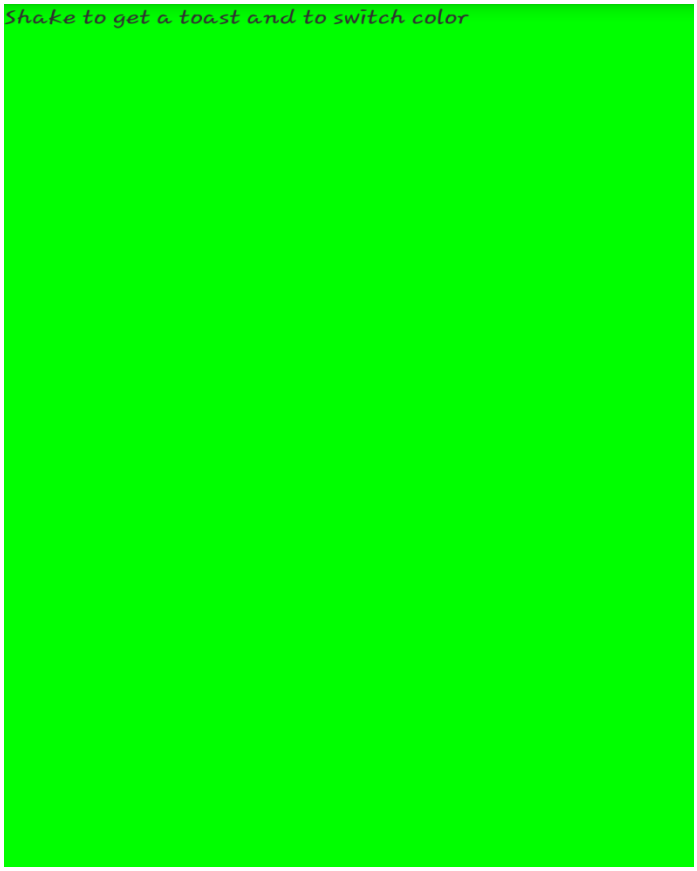
Capture-2

Task 2: Mobile sensor with Android sensor app

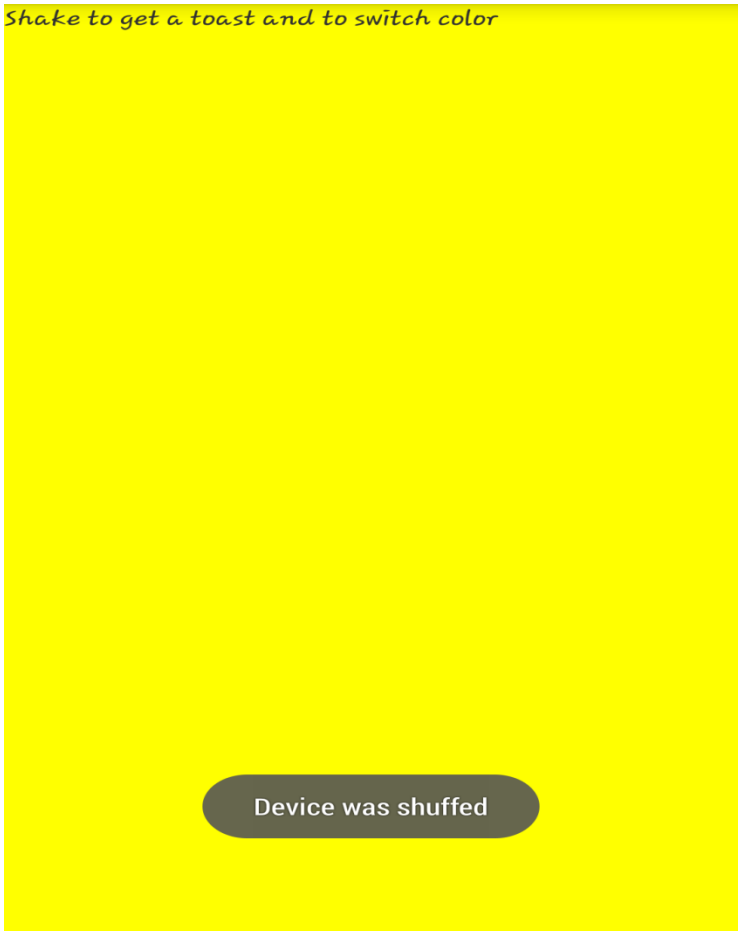
Step 1: Exatract and modify the application from blackboard.

Step 2: Run the application on the device after enbling debugging mode.

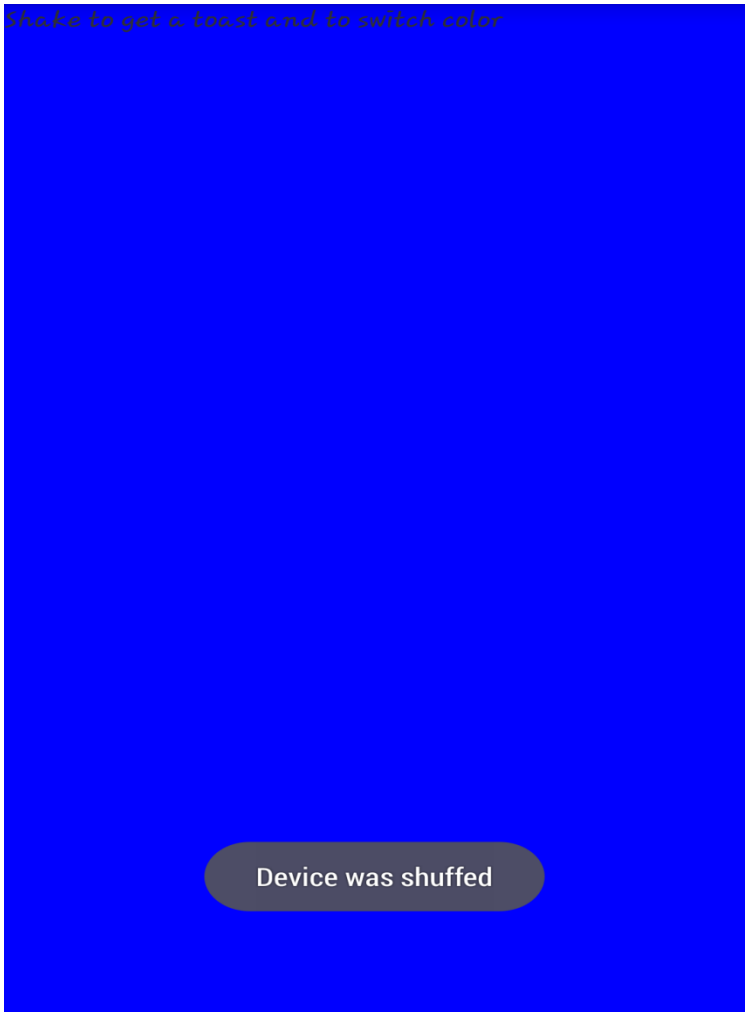
Step 3: Observe the output and take the screen shots of result.



First screen of the application



When we shake the device the sensor detects and gives sensor ouput. By detecting output we change the colour of the device.



Again when we shake the device the sensor detects and gives sensor ouput. By detecting output we change the colour of the device.

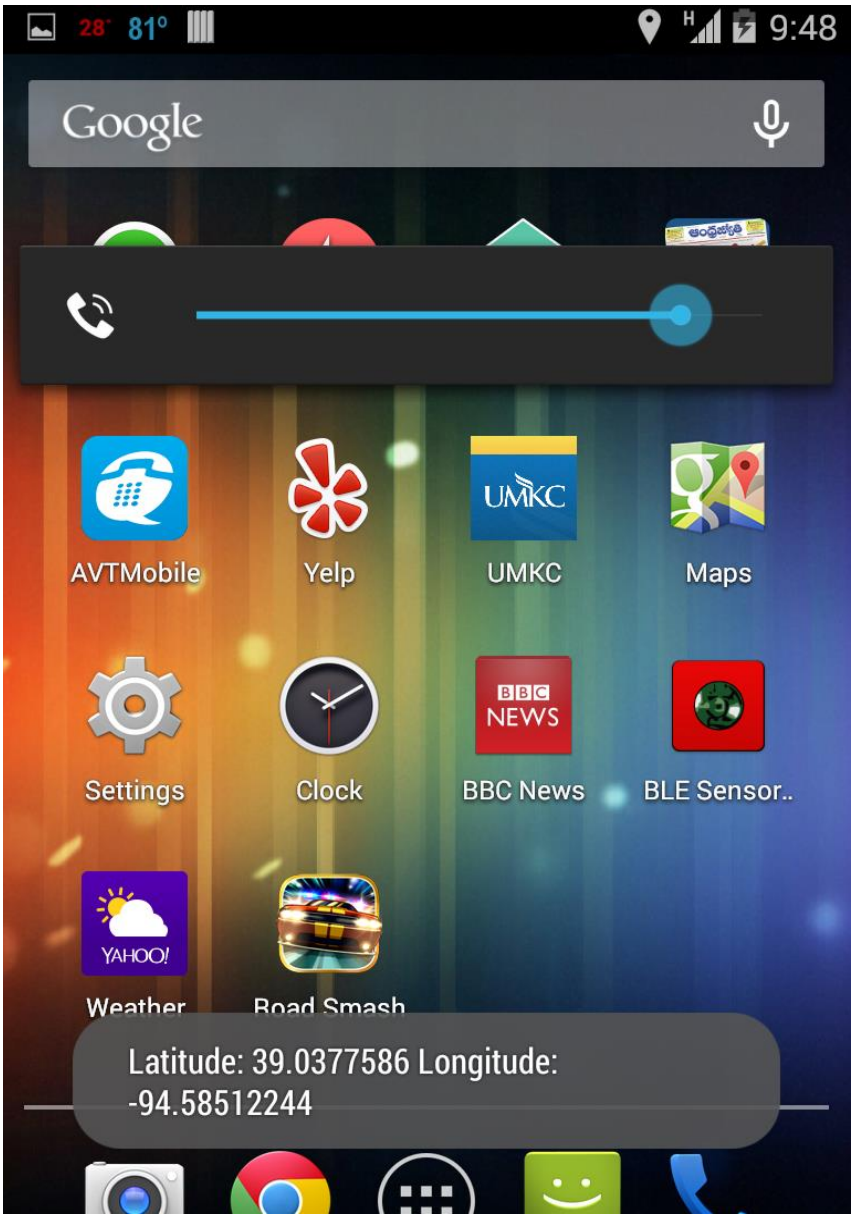
Task3: Geosensing android application

Step 1: Exatract and modify the application and add libraries.

Step 2: Run the application on the device after enbling debugging mode.

Step 3: Observe the output and take the screen shots of result.

First screen detecting the geo loacation and Displaying address using google API service call with geo location coordinates.

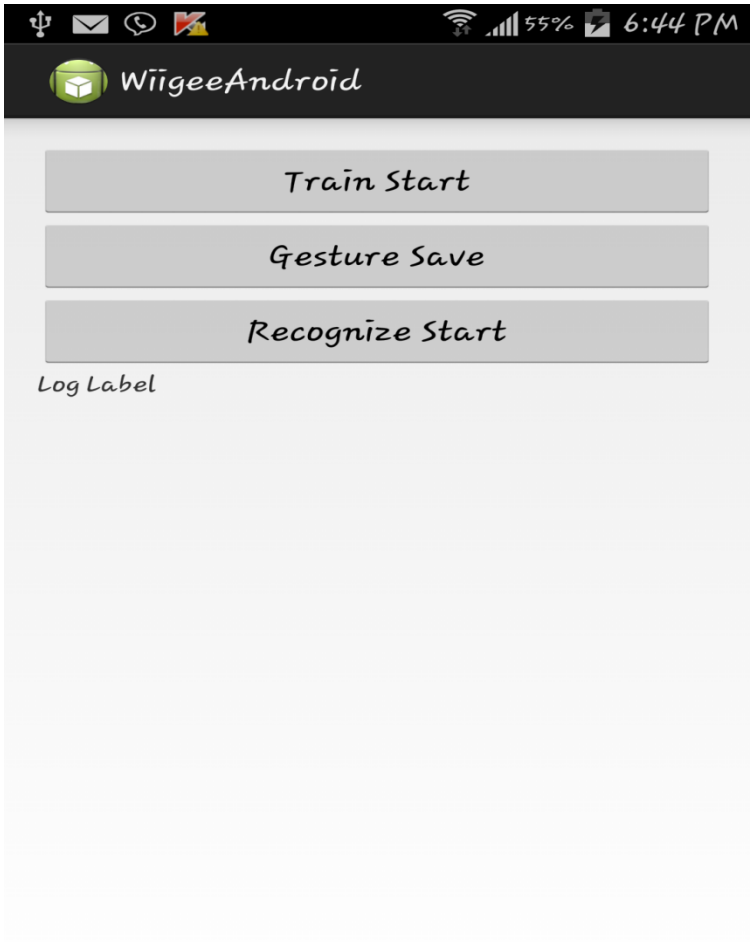


Task4: Wiigee app with Android smartphone

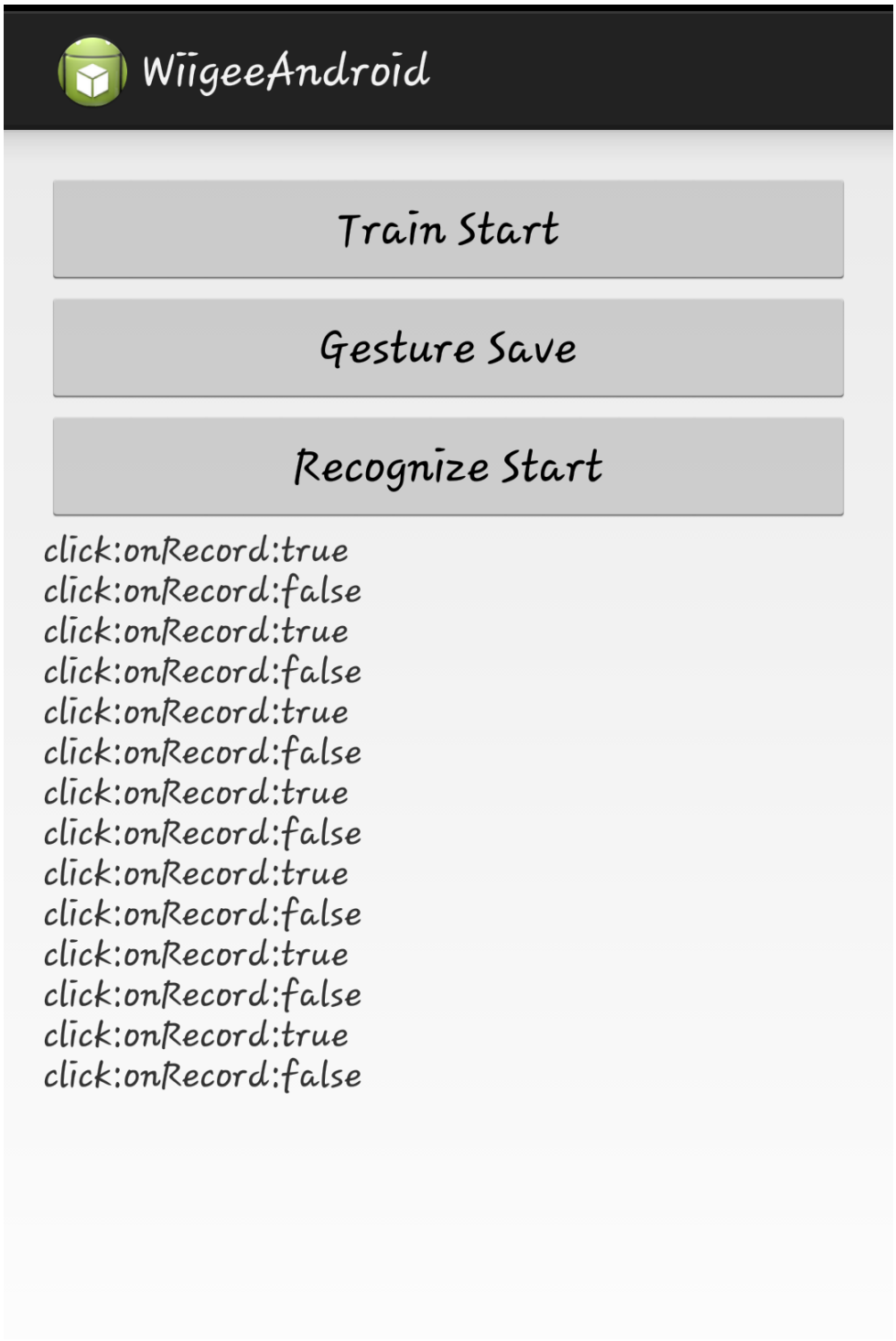
Step 1: Exatract and modify the application. Modify build path.

Step 2: Run the application on the device after enbling debugging mode.

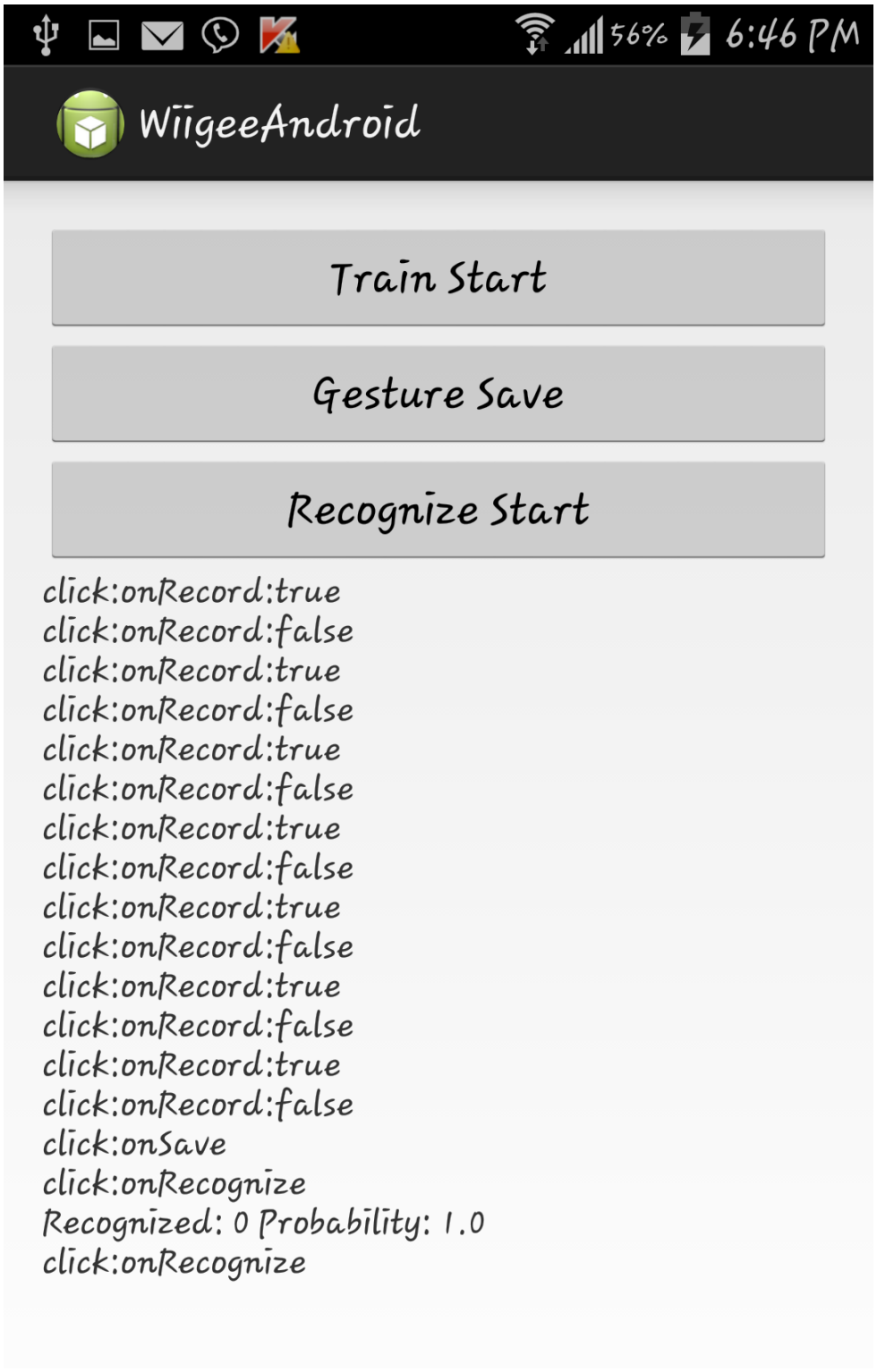
Step 3: Observe the output and take the screen shots of result.



First screen of the application



When we capture gesture click on record a gesture and after motion click on stop. After few samples click on save gesture.



Now to check the gesture we click on recognize and make motion. If motion matches it shows probability of gesture match.

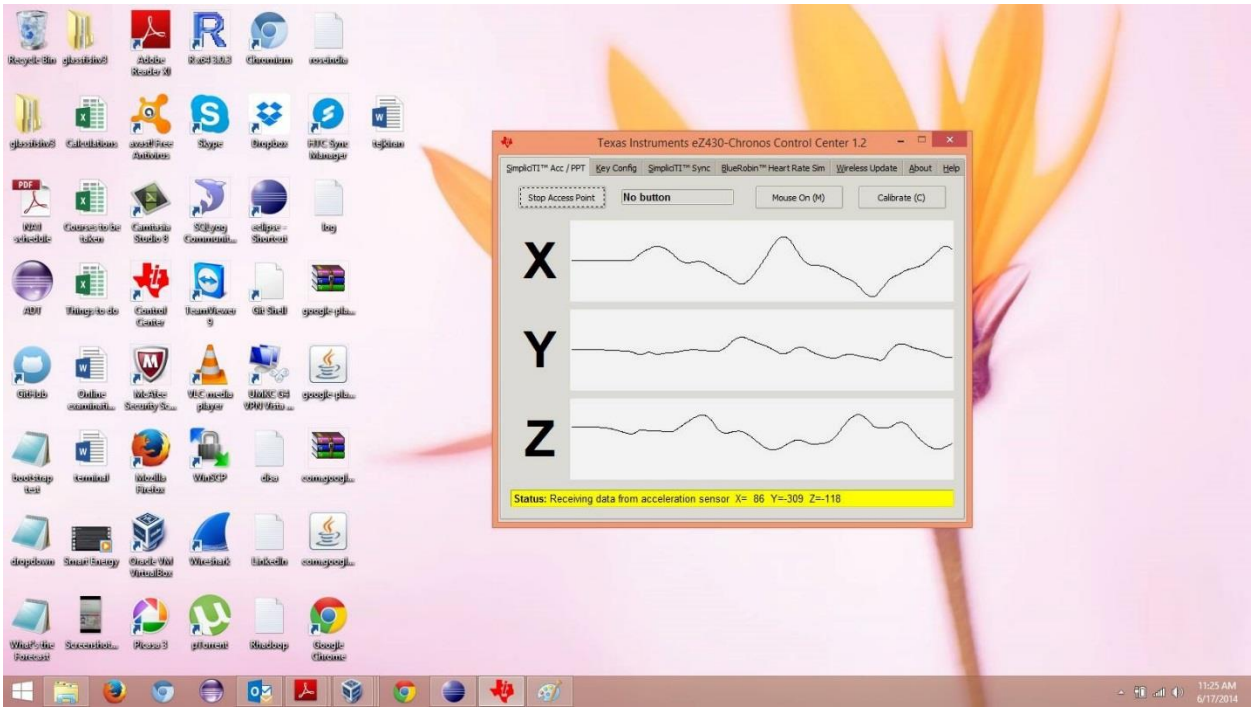
Task5: Application using chronus watch

Step 1: Install the chronus watch drivers.

Step 2: Enable ACC mode on watch.

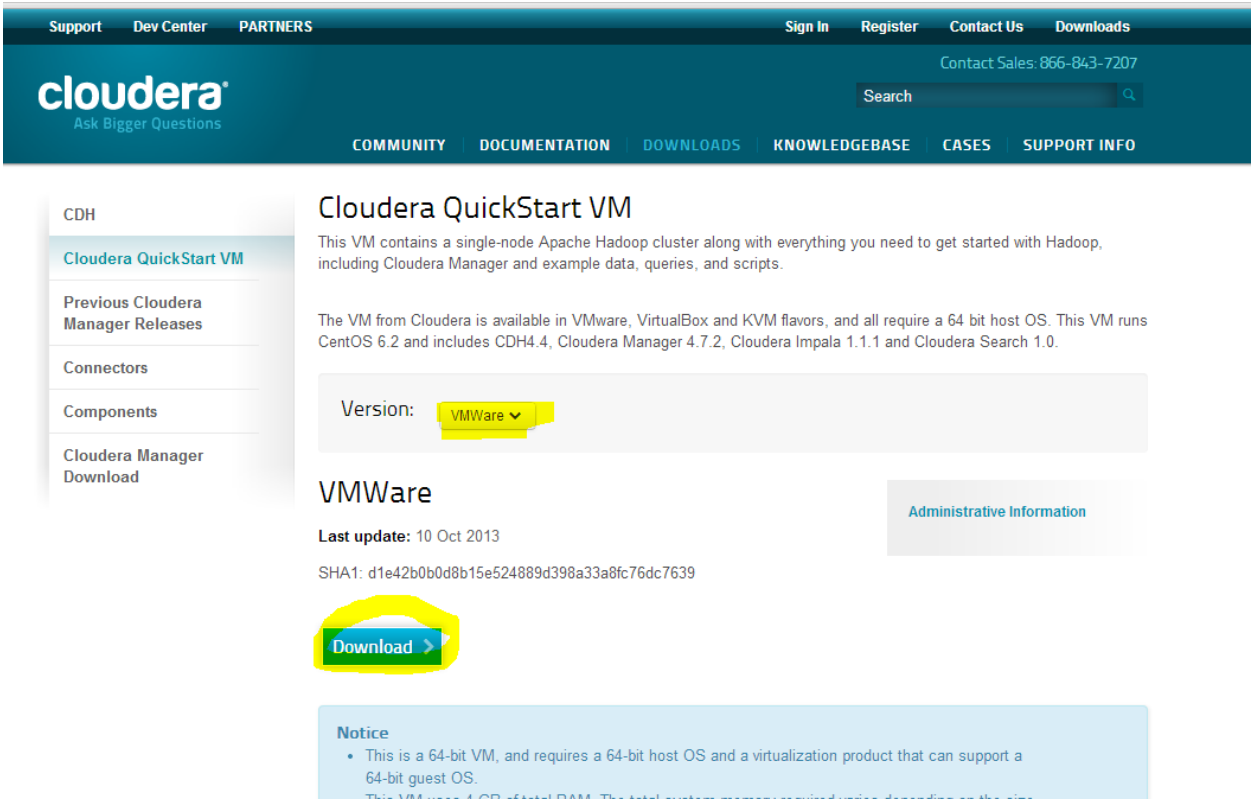
Step 3: Click start capture on the application.

NOTE:

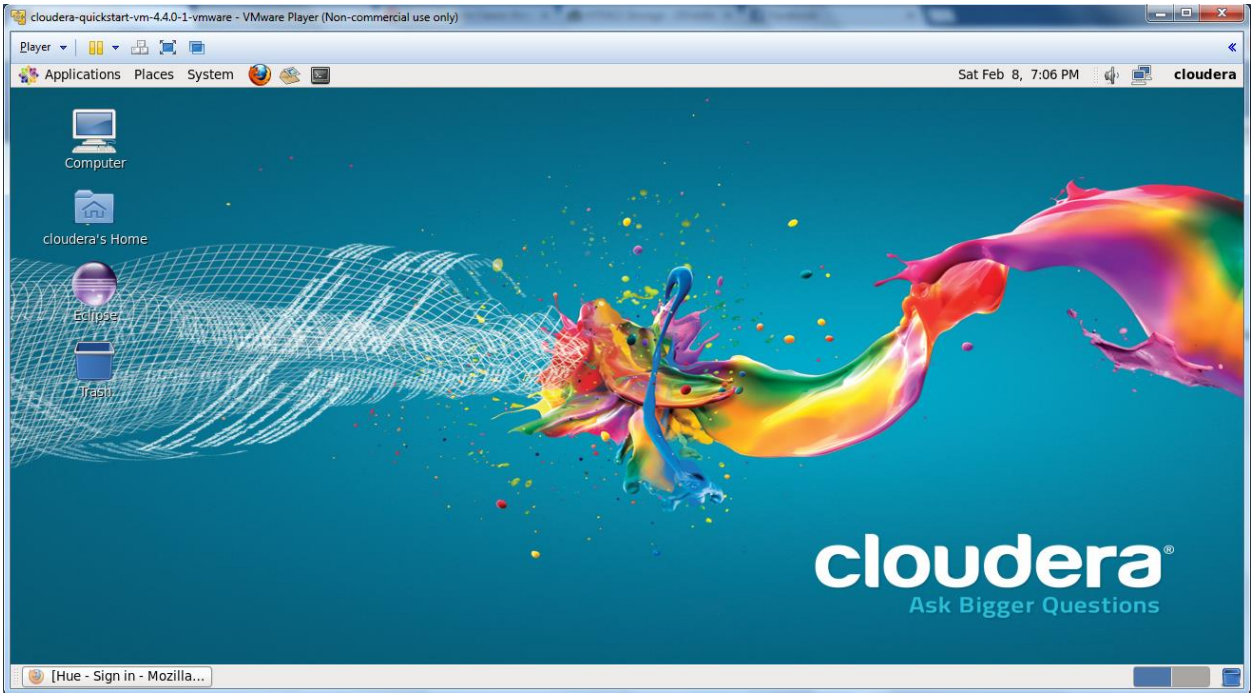


Part 2: Cloudera/MapReduce**: Download the Cloudera Image, implement the WordCount MapReduce and run it.**

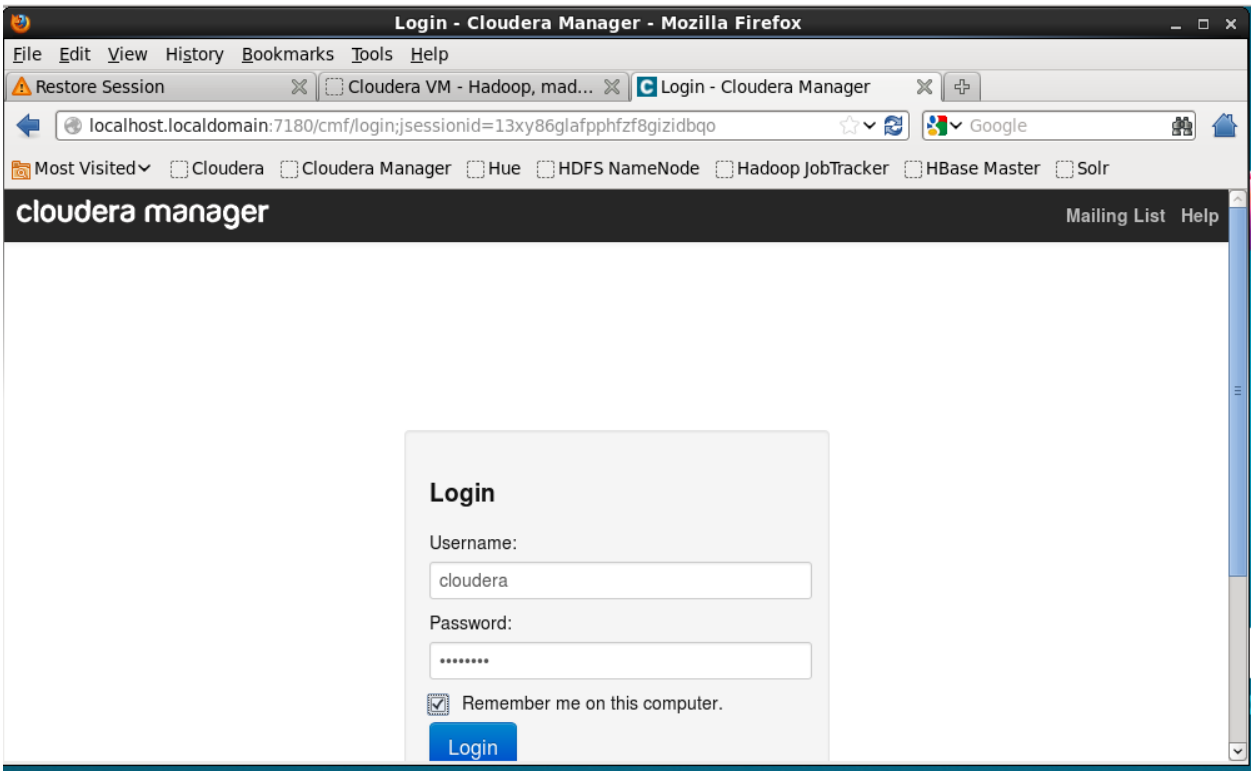
Step 1: Downloading cloud image



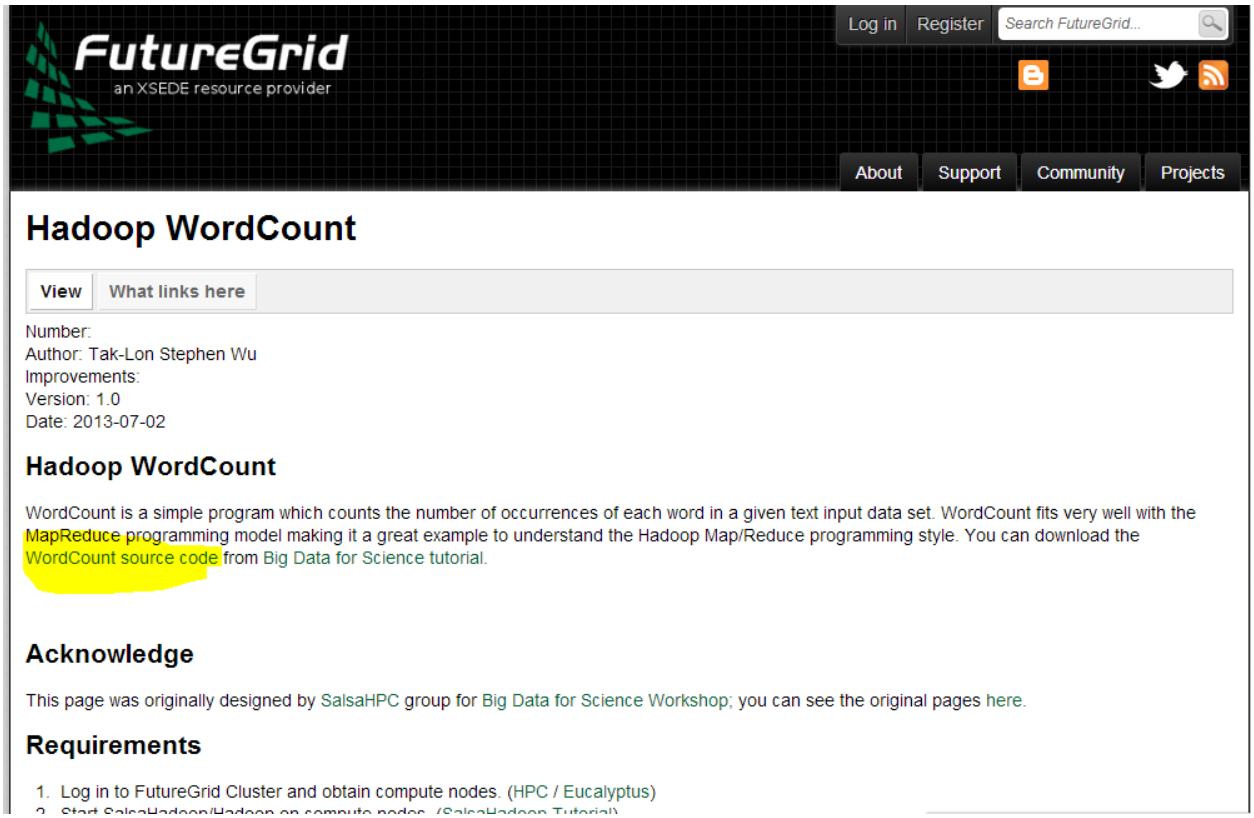
Step 2: open image using VM player



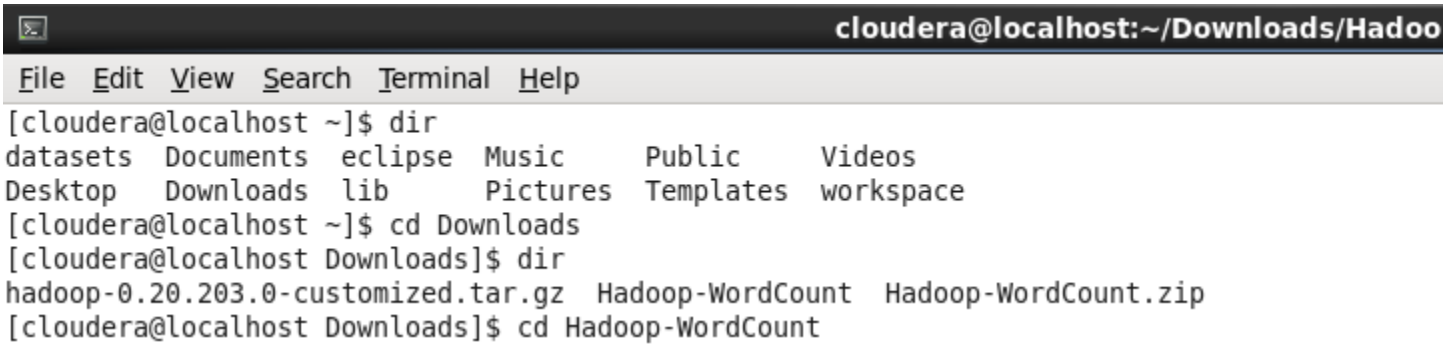
Step 3: open cloudera manager entering the credentials.



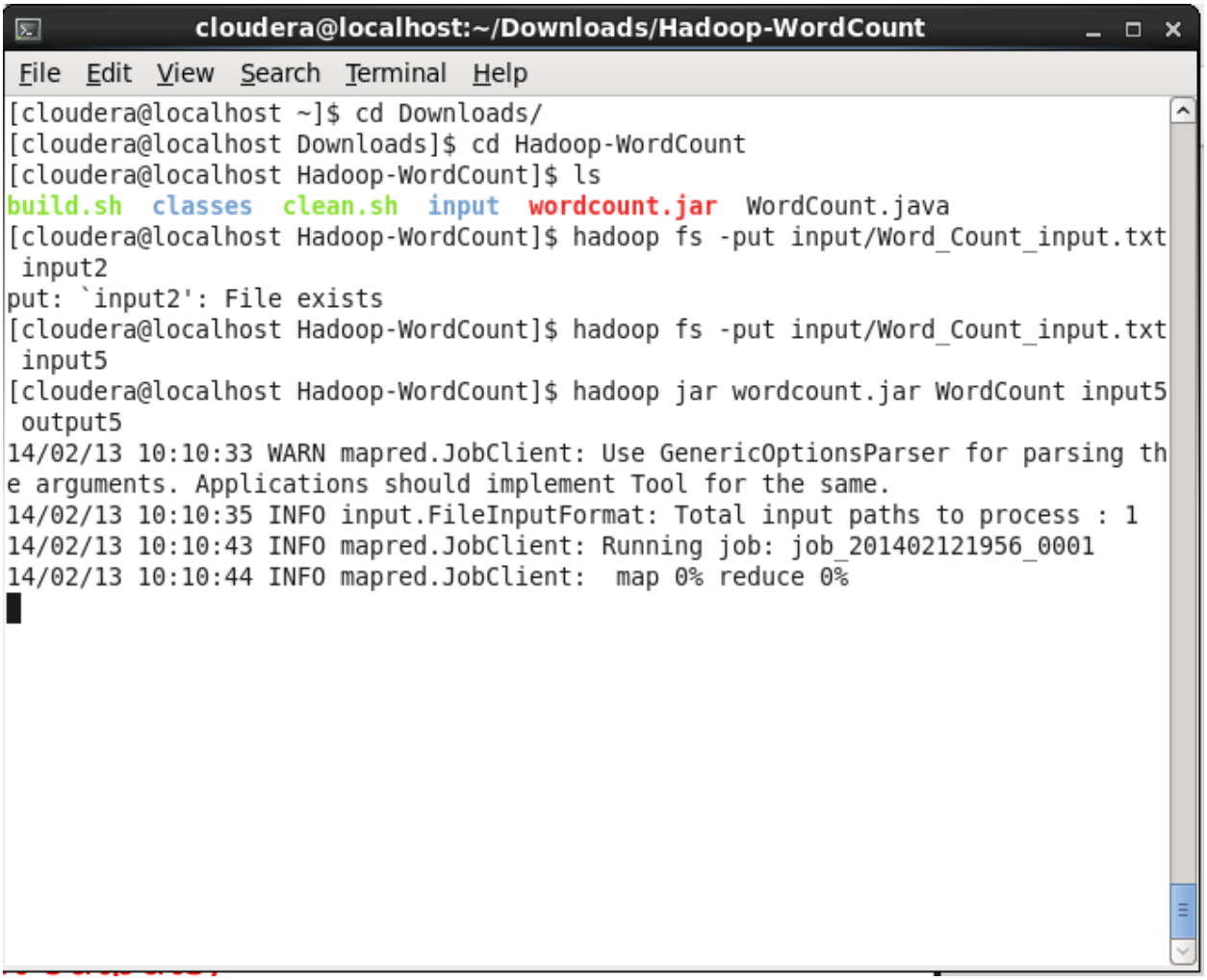
Step 4 : Downloading Word Count file and running using terminal. Download Page:



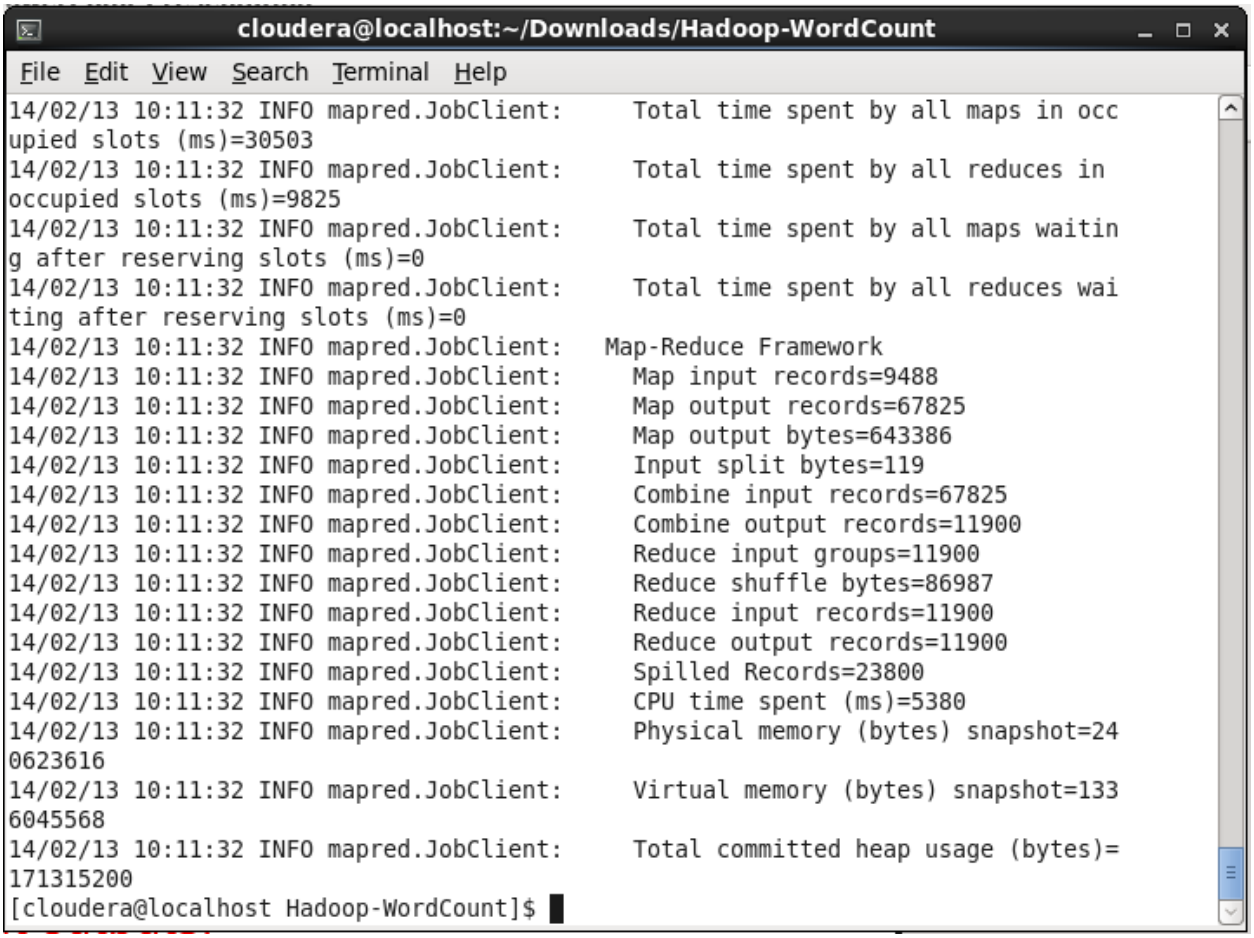
Step 5: Extracting the files



Step 5: Input text folder to processing folder



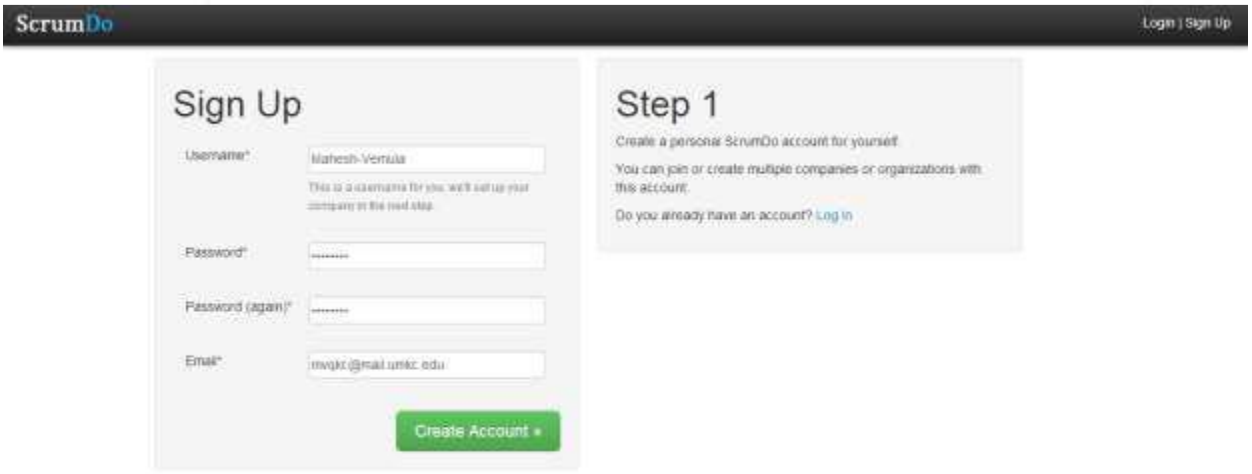
Step 6: now running the word count


cloudera@localhost:~/Downloads/Hadoop-WordCount
File Edit View Search Terminal Help
14/02/13 10:11:32 INFO mapred.JobClient: Total time spent by all maps in occupied slots (ms)=30503
14/02/13 10:11:32 INFO mapred.JobClient: Total time spent by all reduces in occupied slots (ms)=9825
14/02/13 10:11:32 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0
14/02/13 10:11:32 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0
14/02/13 10:11:32 INFO mapred.JobClient: Map-Reduce Framework
14/02/13 10:11:32 INFO mapred.JobClient: Map input records=9488
14/02/13 10:11:32 INFO mapred.JobClient: Map output records=67825
14/02/13 10:11:32 INFO mapred.JobClient: Map output bytes=643386
14/02/13 10:11:32 INFO mapred.JobClient: Input split bytes=119
14/02/13 10:11:32 INFO mapred.JobClient: Combine input records=67825
14/02/13 10:11:32 INFO mapred.JobClient: Combine output records=11900
14/02/13 10:11:32 INFO mapred.JobClient: Reduce input groups=11900
14/02/13 10:11:32 INFO mapred.JobClient: Reduce shuffle bytes=86987
14/02/13 10:11:32 INFO mapred.JobClient: Reduce input records=11900
14/02/13 10:11:32 INFO mapred.JobClient: Reduce output records=11900
14/02/13 10:11:32 INFO mapred.JobClient: Spilled Records=23800
14/02/13 10:11:32 INFO mapred.JobClient: CPU time spent (ms)=5380
14/02/13 10:11:32 INFO mapred.JobClient: Physical memory (bytes) snapshot=240623616
14/02/13 10:11:32 INFO mapred.JobClient: Virtual memory (bytes) snapshot=1336045568
14/02/13 10:11:32 INFO mapred.JobClient: Total committed heap usage (bytes)=171315200
[cloudera@localhost Hadoop-WordCount]$

Step 7: output file

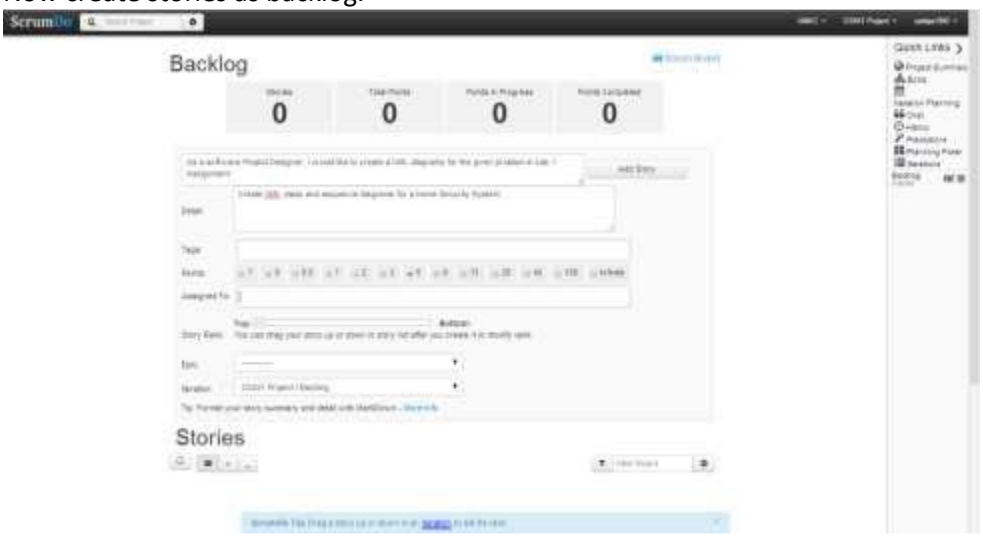
Part 3: Task1 – ScrumDo :

Step1: Signing up into ScrumDo by providing required details.

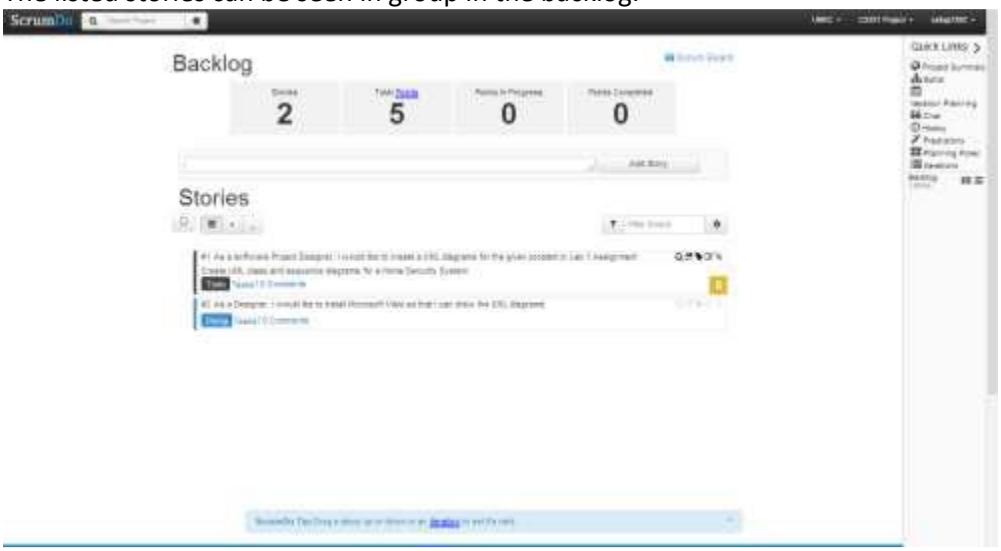


Step2: Now create a organization and add Project

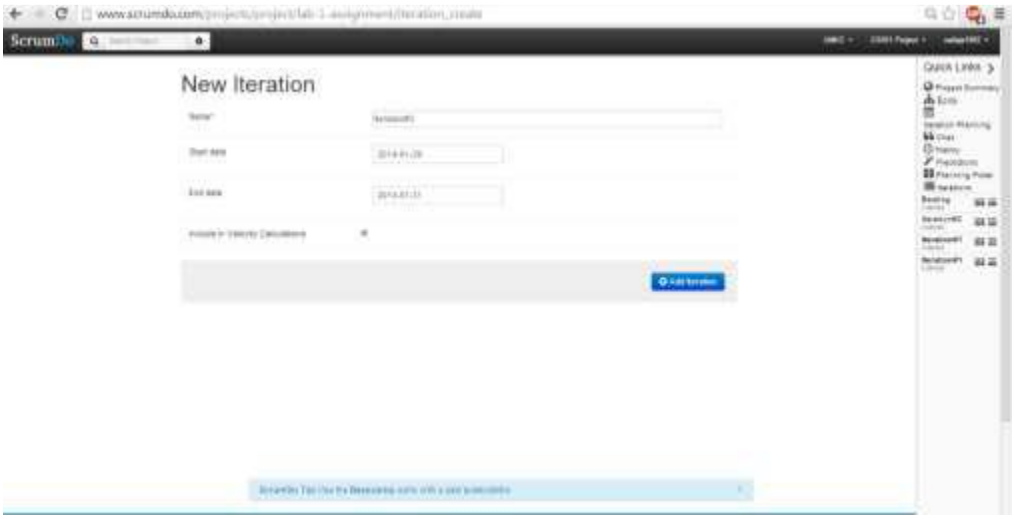
Now create stories as backlog.

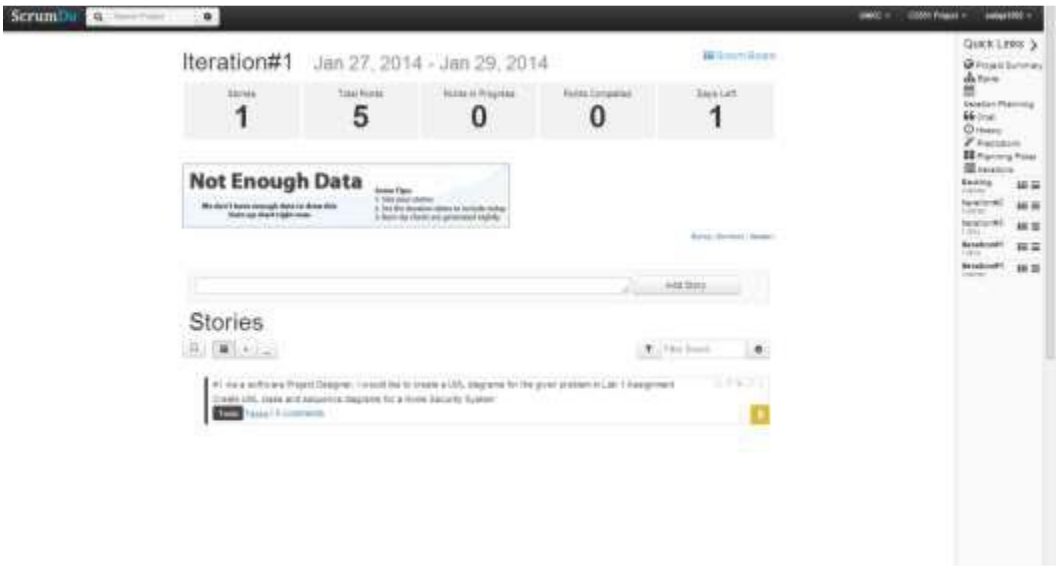


The listed stories can be seen in group in the backlog.



Add iterations

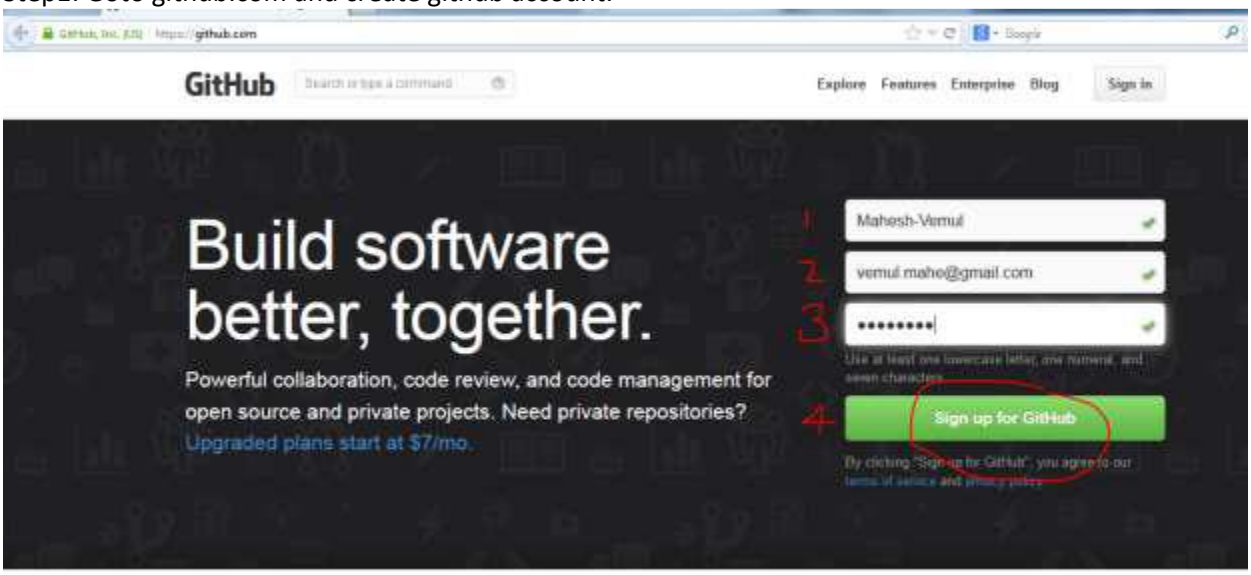




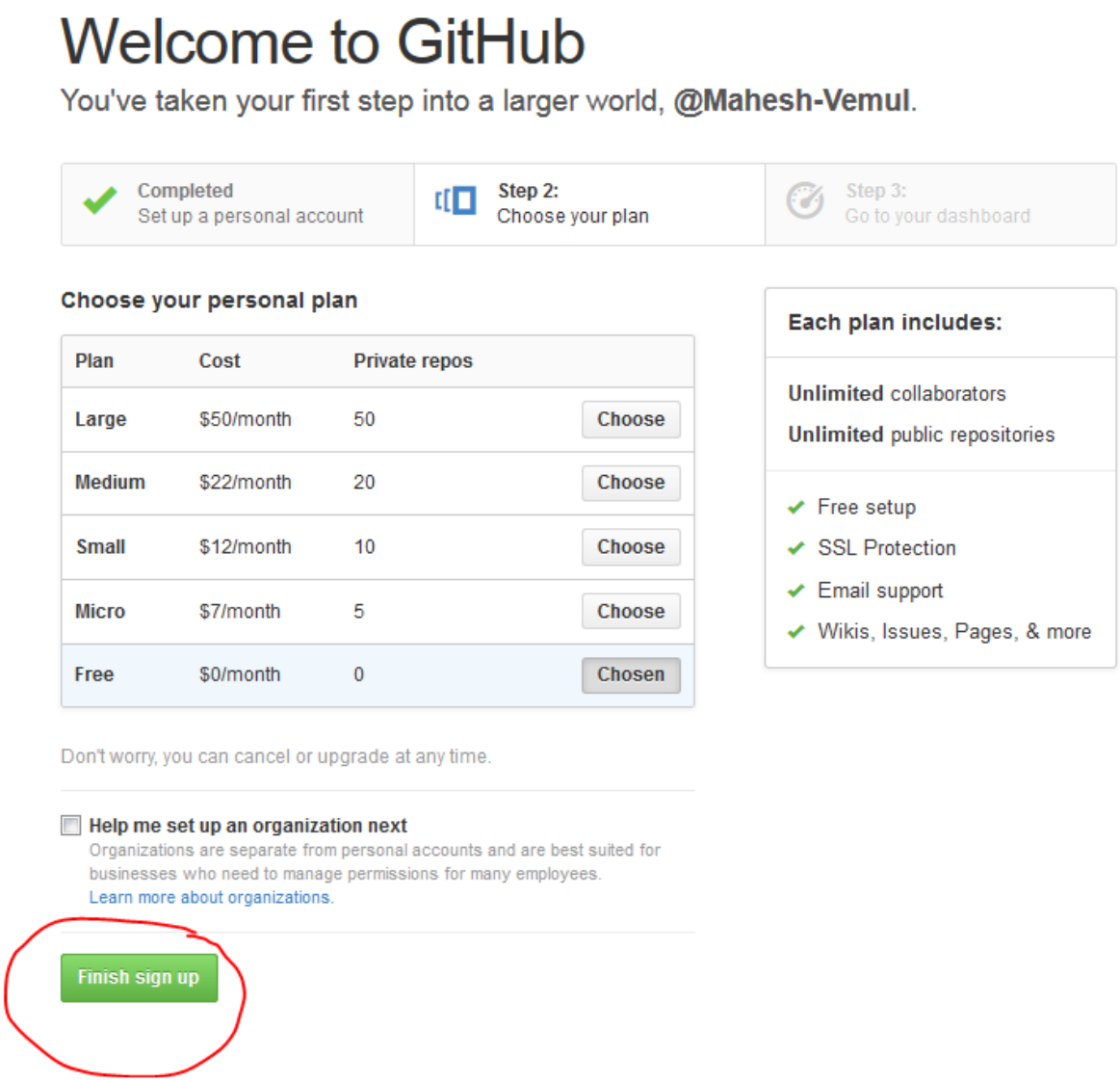
Now create team and add team members

Task2-Github:

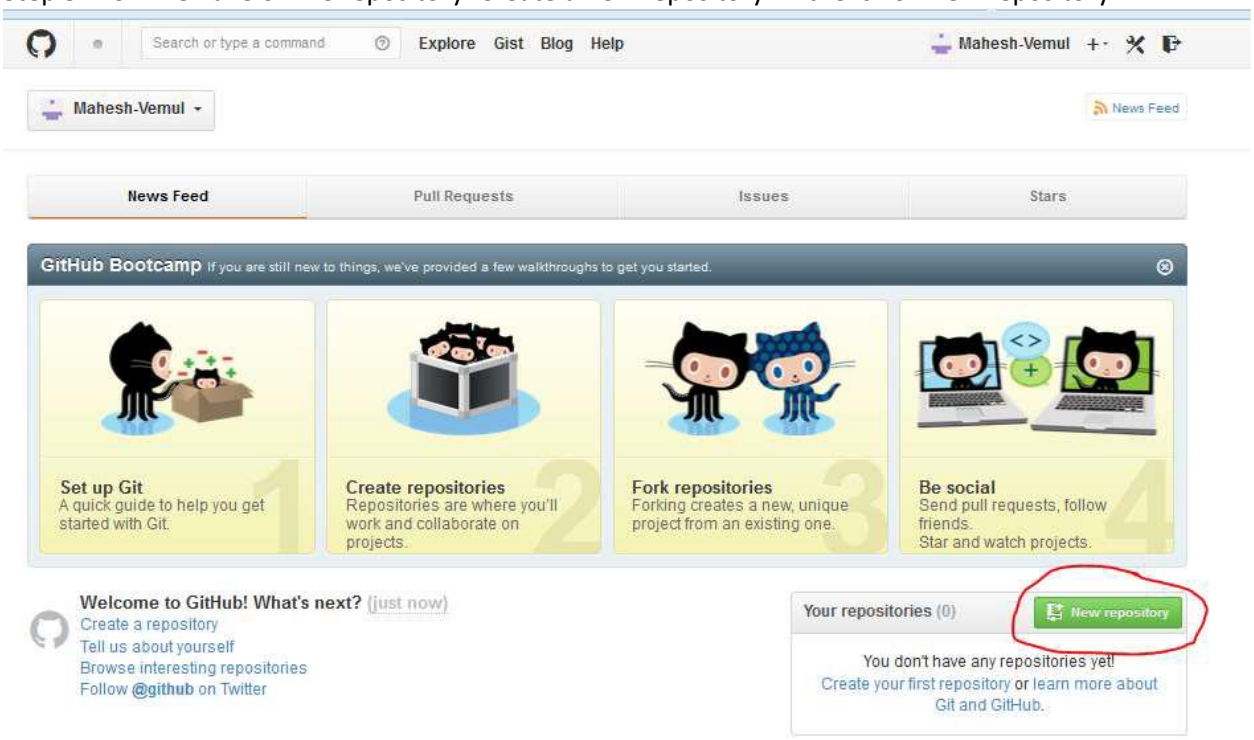
Step1: Goto github.com and create github account.



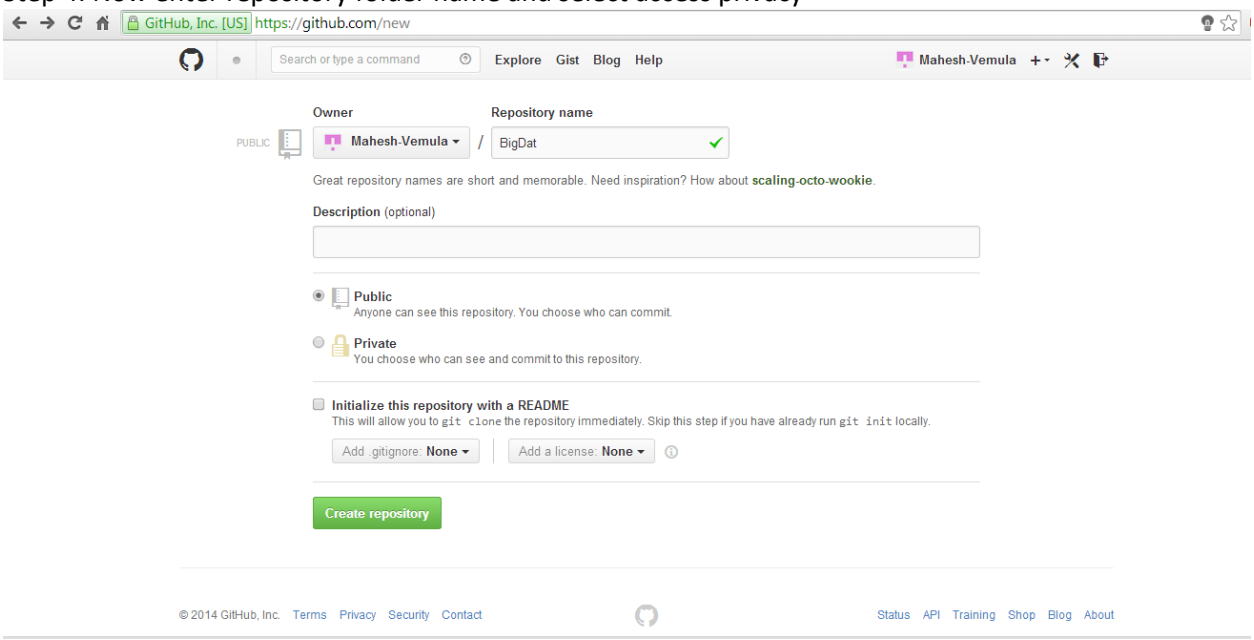
Step 2: Select your plan and click finish



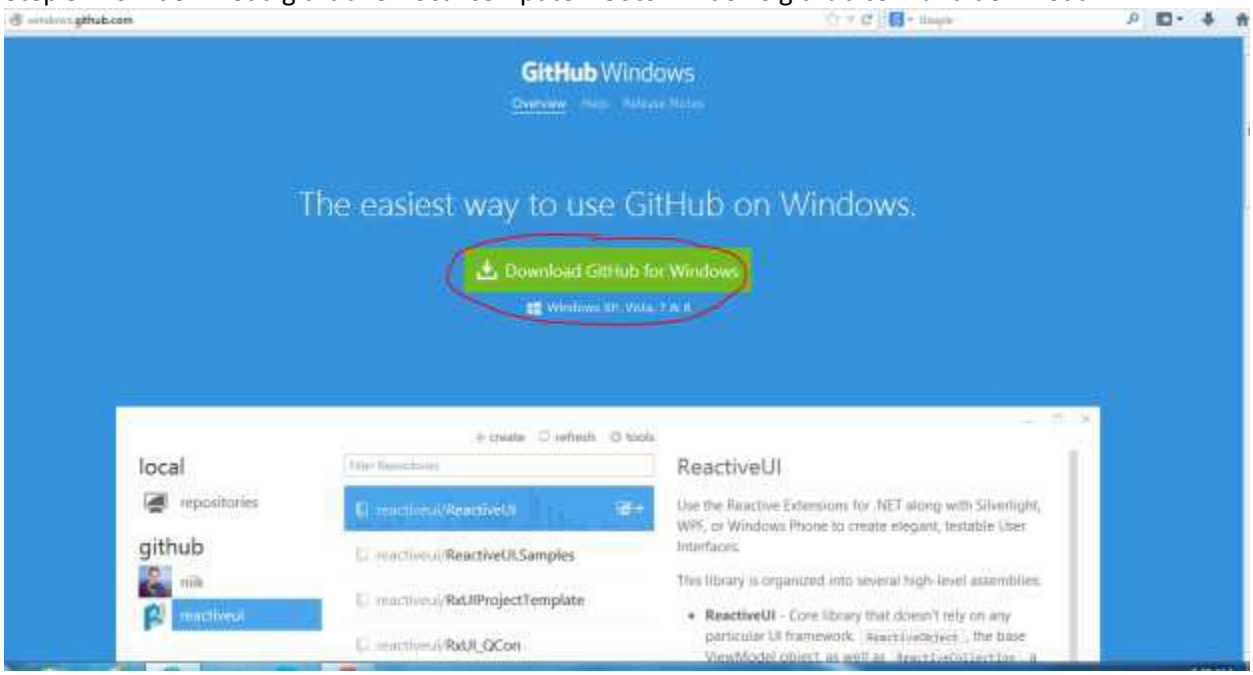
Step 3: Now we have online repository. Create a new repository in it. Click on new repository



Step 4: Now enter repository folder name and select access privacy

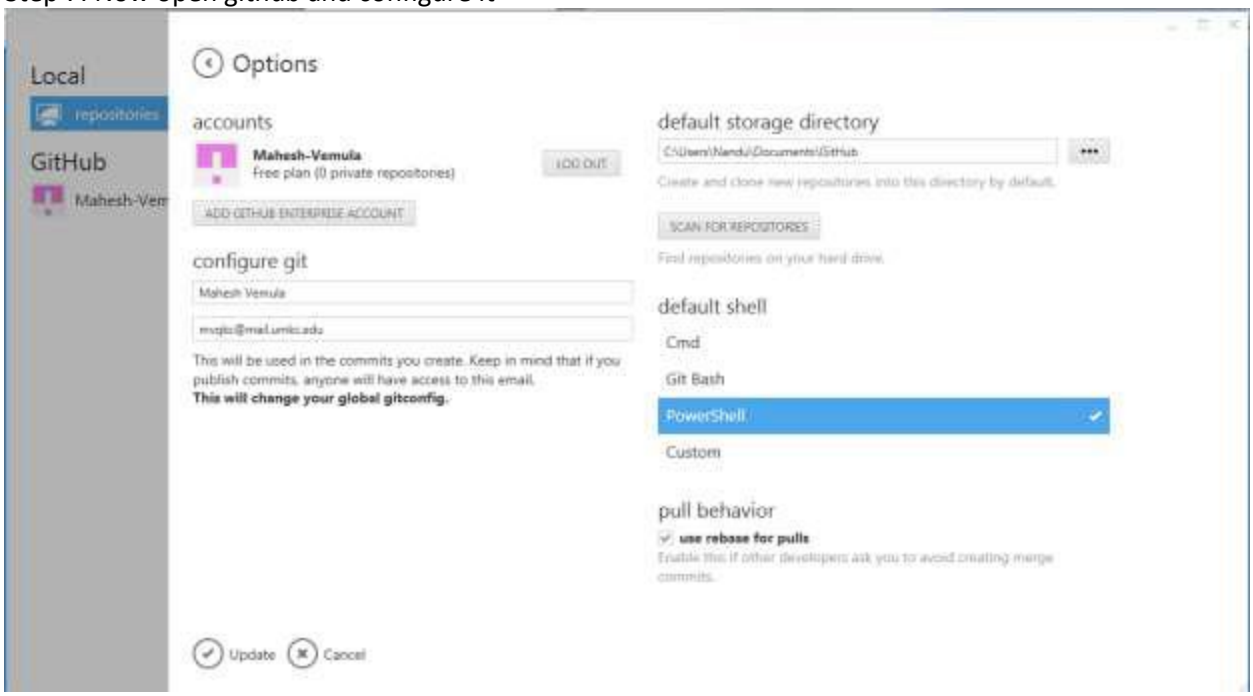


Step 5: Now download github for local computer. Goto windows.github.com and download

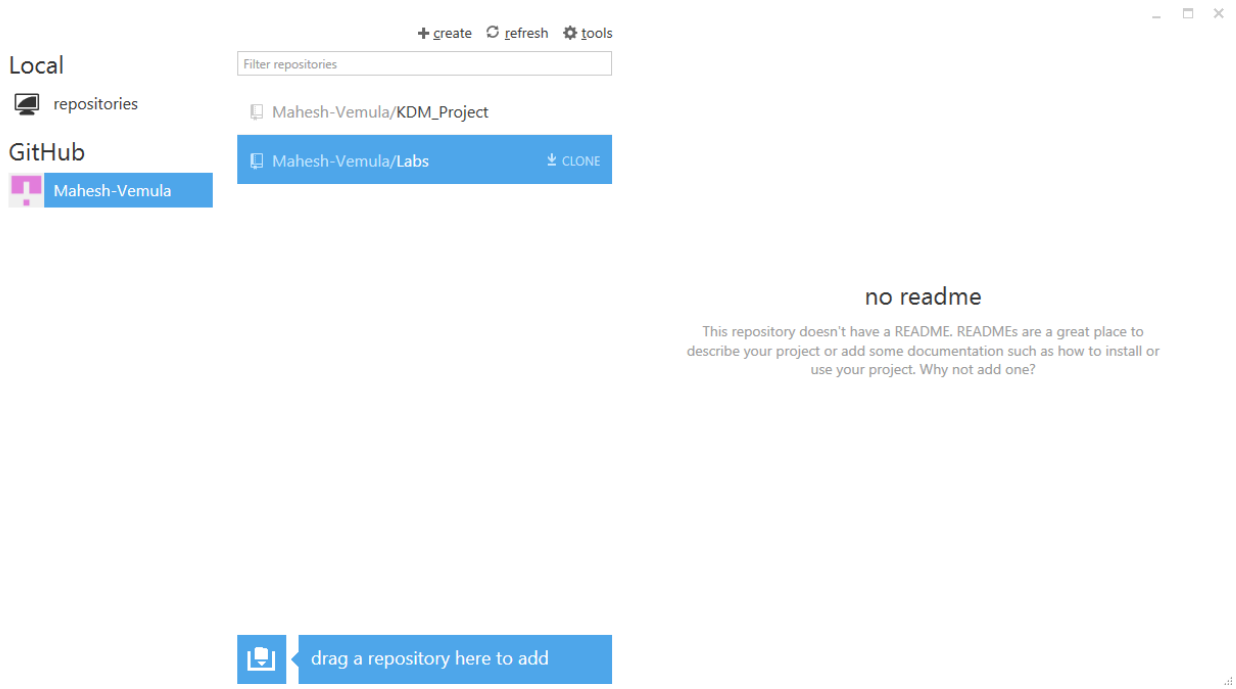


Step 6: After downloading install github.

Step 7: Now open github and configure it



Step 8: Github repository is now ready to sync.



END of REPORT