**HackyTeam Solution for Problem Statement 2**

**Git URL**: https://github.com/rajesharma470/GainSightHappyHack.git

**Overview**:

**Description**:

**REST side service** will take the data streams and set to queues of various Storm **Spouts** which does the main processing. Aggregated data will be updated in DB(**MongoDB**).

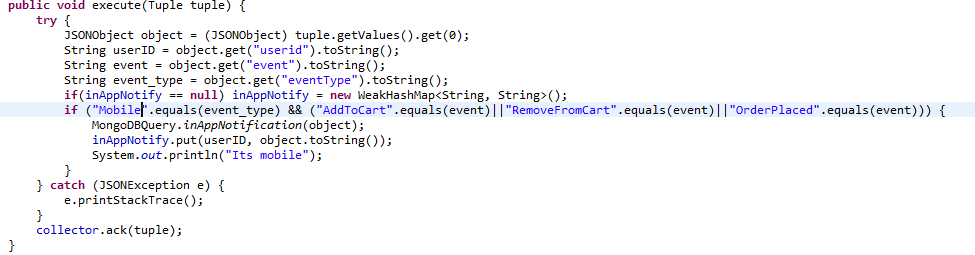
Client/UI interface will be updated with real time data statistics.

**Hypothesis:**

1. For identifying the users for delivering in-app notification, we have assumed that those users who have ‘event\_type’ as ‘Mobile’ and have ‘Ordered’ or ‘AddToCart’ will be notified. Those events with ‘event\_type’ is ‘Mobile’ and are with events ‘Order’, ‘AddToCart’ will be updated in ‘inApp\_notify’ table. Assuming that there is another service which pools this table based on last hour and can retrieve the details.

Sample Flow:

1. Request with POST data is hit at ‘http://localhost:8080//com.happy.hackweb/activity’ then ‘HappyHackJSONService.createXmlEventsInJSON’ method receives the data
2. ‘HappyHackJSONService.createXmlEventsInJSON’ sets the topologies(Apache Storm), its one time only for the first request
3. The request received is converted to JSONObject and set in ‘inApp.queue’
4. ‘InAppNotificationSpout’ will poll the queue and emit the data to ‘NotificationBolt’
5. NotificationBolt will process the request based on whether its ‘Mobile’ and ‘AddToCart’, ‘RemoveFromCart’, ‘OrderPlaced’. It will update the MongoDB with the event in table ‘inApp\_notify



1. For sending promotional emails, based on number of event actions for each user we will be maintaining a overall count in DB. Assuming that there is another service which will take help of these entries to send promotional emails
2. For most active geographies, user activities are tracked and from user pin details we retrieve the geography details (like state, district, city) with timestamp. DB entries are done with all these user-geography details. Assuming that UI system or something similar can use this data

**Tools/Technologies:**

REST Layer – Jersey

Web Container – Tomcat

Real time processor – Apache Storm

Database – MongoDB

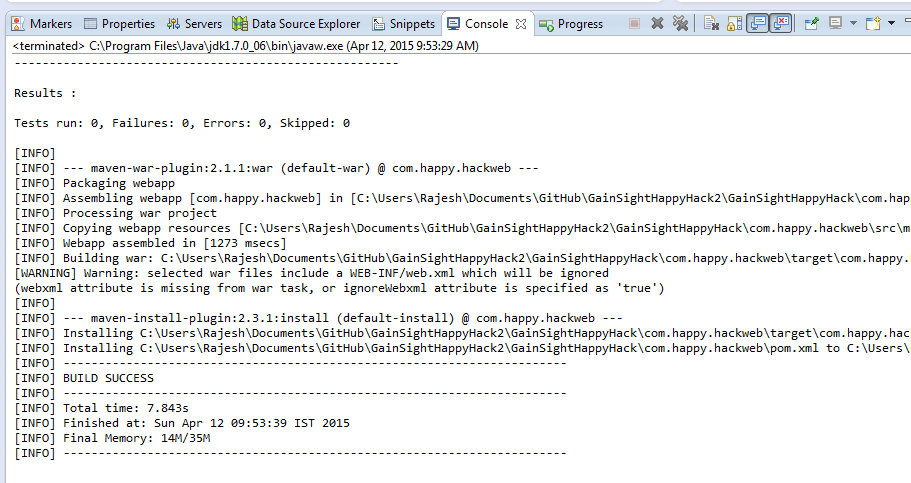
Programming language - Java

Testing tool – Apache Jmeter

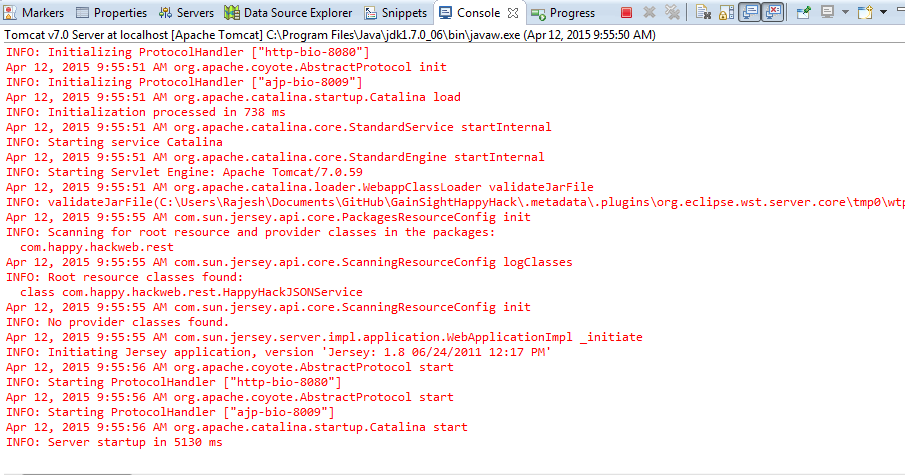
Packaging/build tool – Apache Maven

**Installation/Testing:**

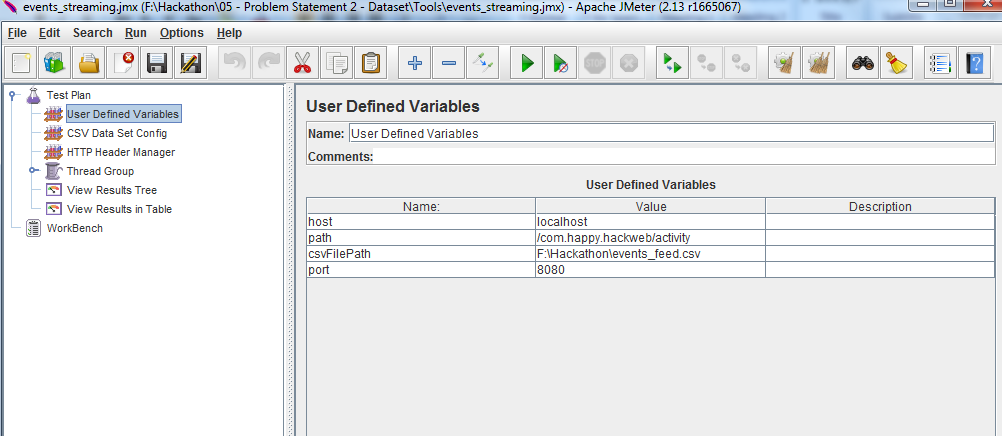
1. Load all the master csv files (users\_master, products\_master, pin\_master) into ‘gainsight’ collection in mongoDB
2. ‘mvn clean install’ on ‘com.happy.hackweb’ to build the war.



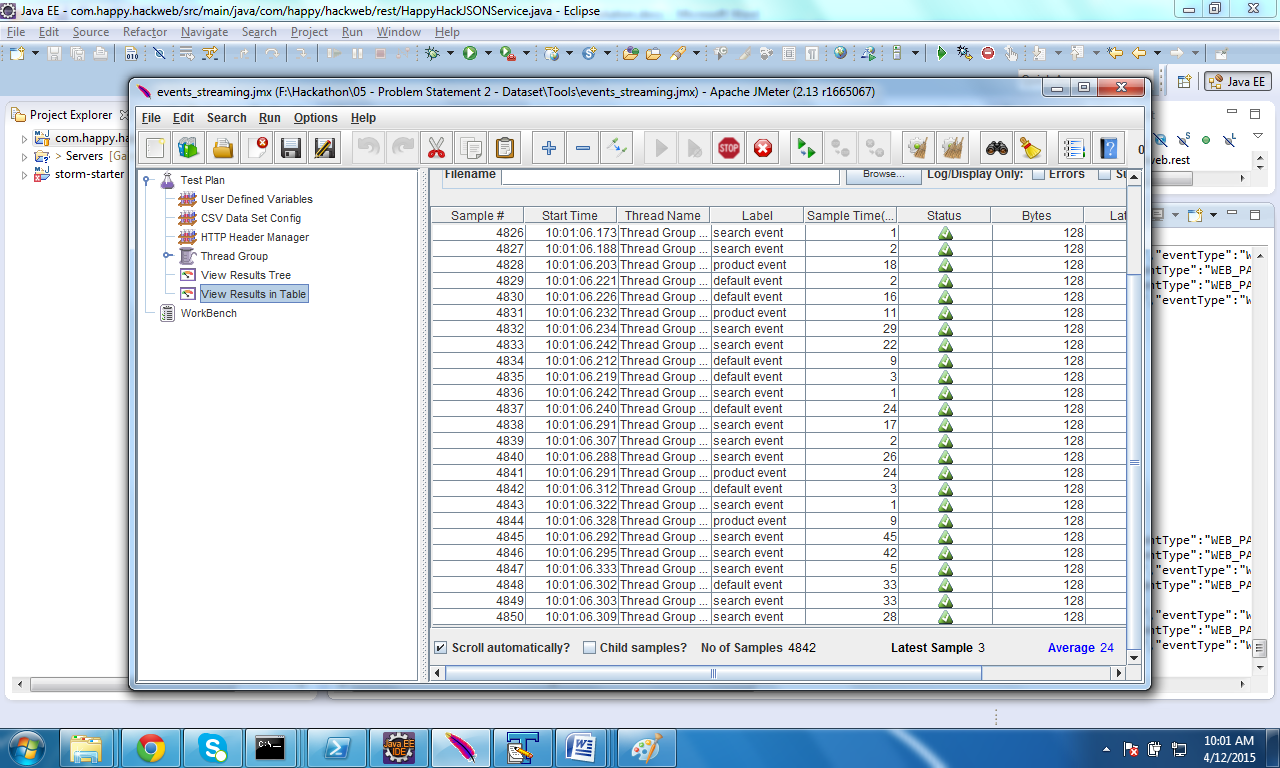
1. Deploy in Apache tomcat server



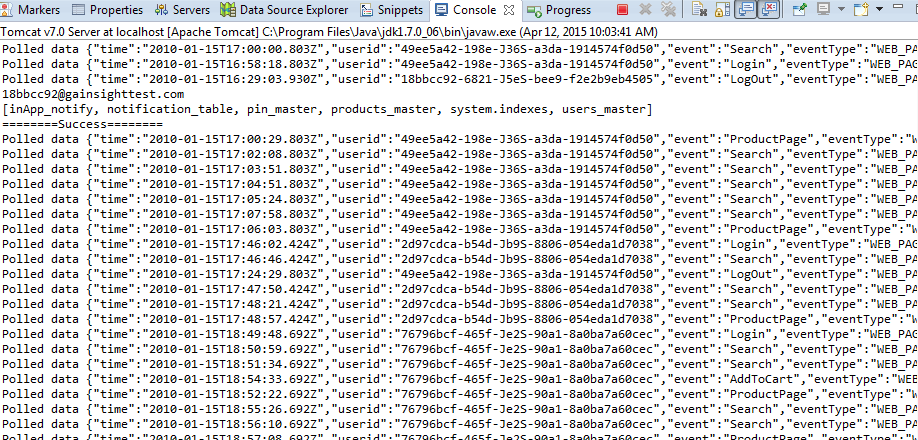
1. Set the testing tool Jmeter for sending the streamed data



1. Start Jmeter run



1. Check for streams are processed



1. Verify the data in DB



**Improvements/pending items:**

1. UI can be added to retrieve data from the notification/analysis tables.
2. Active Geographies implementation has been partially completed.
3. MongoDB integration still can be improved.