USER ROUTINE PREDICTIVE ANALYSIS

BIG DATA ANALYTICS AND APPLICATIONS PROJECT

SPRING 2016

Track LIVE

Recommend LIVE

SUBMITTED BY

RAKESH REDDY BANDI (02)

NIHAR DUDAM (06)

SRICHARAN PAMURU (20)

MARK SCHULTZ (26)

Contents

1. Project Proposal

Motivation:

Safety is an important measure of human sustainability. It would be essential to create an application which would favor location tracking. Remote locations need to be accessed primarily and accessibility to network should be reviewed. In case of any emergency, when the user is not within reach, an emergency procedure needs to be incorporated. Most importantly, the user's activity needs to be recommended or rather reminded.

Goal and Objectives:

The important aspect of this application would be to promote security and safety, thereby incrementing an application interface which is user friendly. The integration of SmartPhone and SmartWatch would guarantee the implementation of GPS signals with the respective user, thereby eliminating situations which are dangerous.

When the user is out of bounds and/or rather in a location where remote sensing and network connections have created a failure, the SmartWatch which is in contact with the user will not respond. This creates an imbalance during communication between the SmartWatch and SmartPhone. As a result of this, the SmartPhone would be required to react in a manner as programmed. A certain protocol will include dialing an emergency number. It should be able to perform speech recognition and voice control. It may eliminate a timeframe in which people may be facing dire consequences. The main goal of our project is to recommend the user activities at particular time periods.

The SmartWatch is planned to be programmed in such a way, that data obtained establishes a list of places which are frequented to by the user. Premature suggestion during routing of locations would be facilitated and relatively make it easier to predict the user's daily activity and a schedule can be inculcated into the device. This can be mainly designed to perform an iterative search pattern for travel and alleviate security of the user in a potential clock time frame. The most significant pattern that needs to be analyzed in this context would be to identify large sets of data collected and distributed across the user's activity period.

Significance:

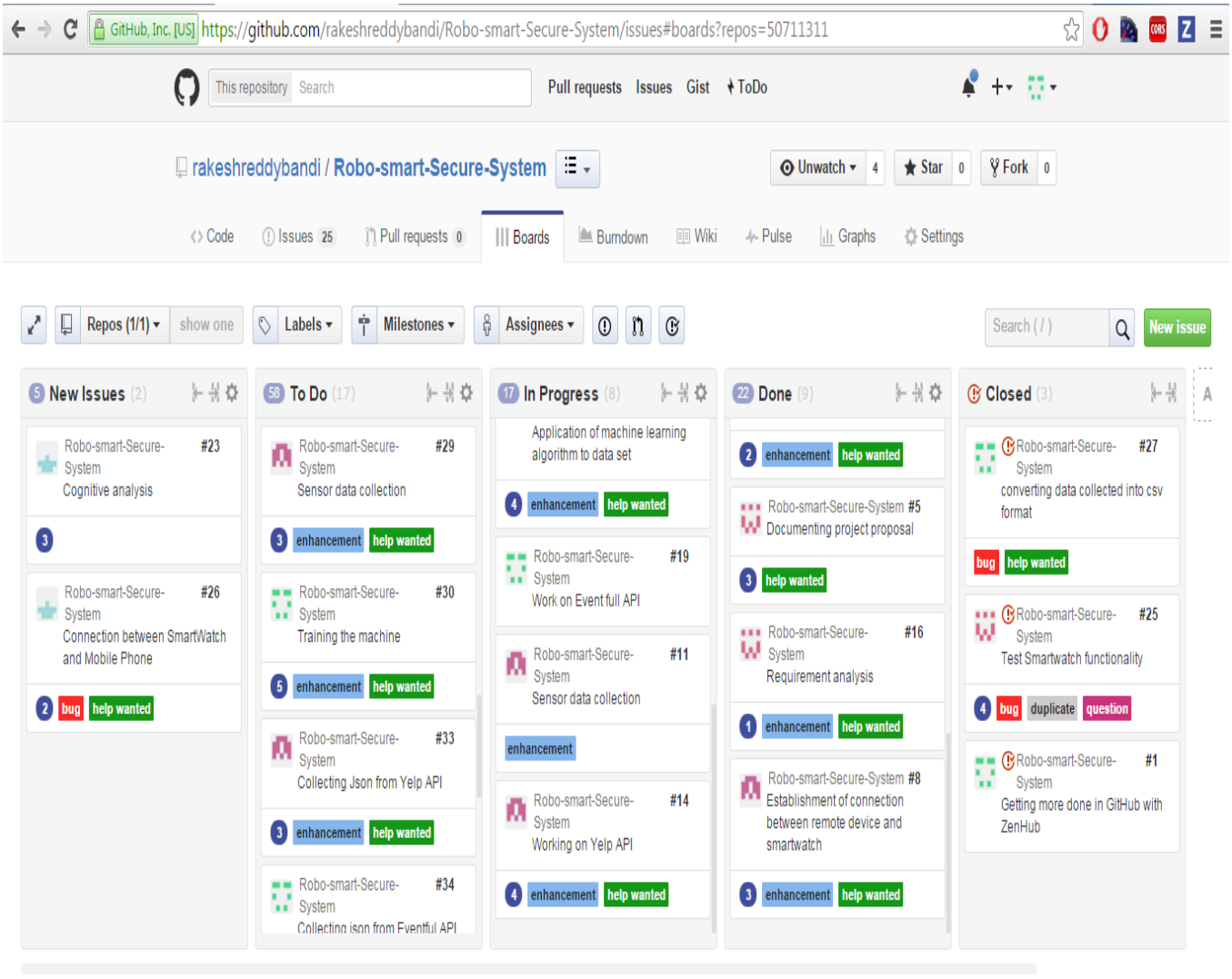
The term 'Smart' associated with both the SmartWatch and SmartPhone is to render the device with protocols for autonomous interactivity and communication across the network. An impetus in Big Data Analytics can be gained by implementation of a secure line between the devices to implement the application of secure activity. It is primarily aimed at providing security to the user in remote locations and analyze data sets. We have implemented an application which tracks the user and provides live recommendations of activity based on time and his location.

2. Project Plan:

1. Schedule for entire project:

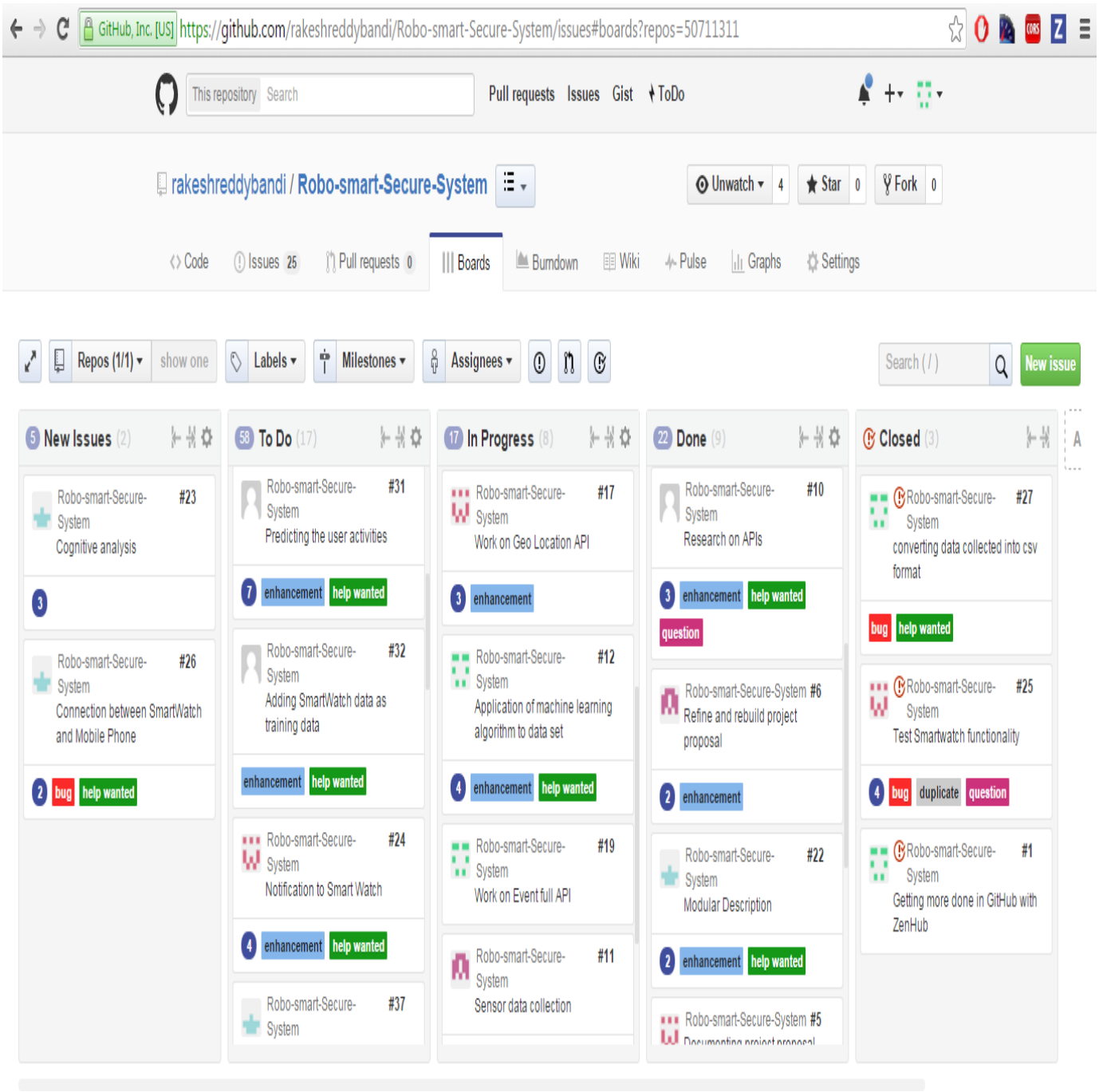
The entire schedule for the project is shown as follows where the respective use case and process step up with all the modules and the time frame is categorized specifically to illustrate the step by step fulfilment of the entire processes done such as feature design and feature implementation.

Screenshot 1: The following screenshot demonstrates the overall perspective of implementation in which the project is being done. The To do and In Progress have a significantly high number of modules and are continued in the following images.



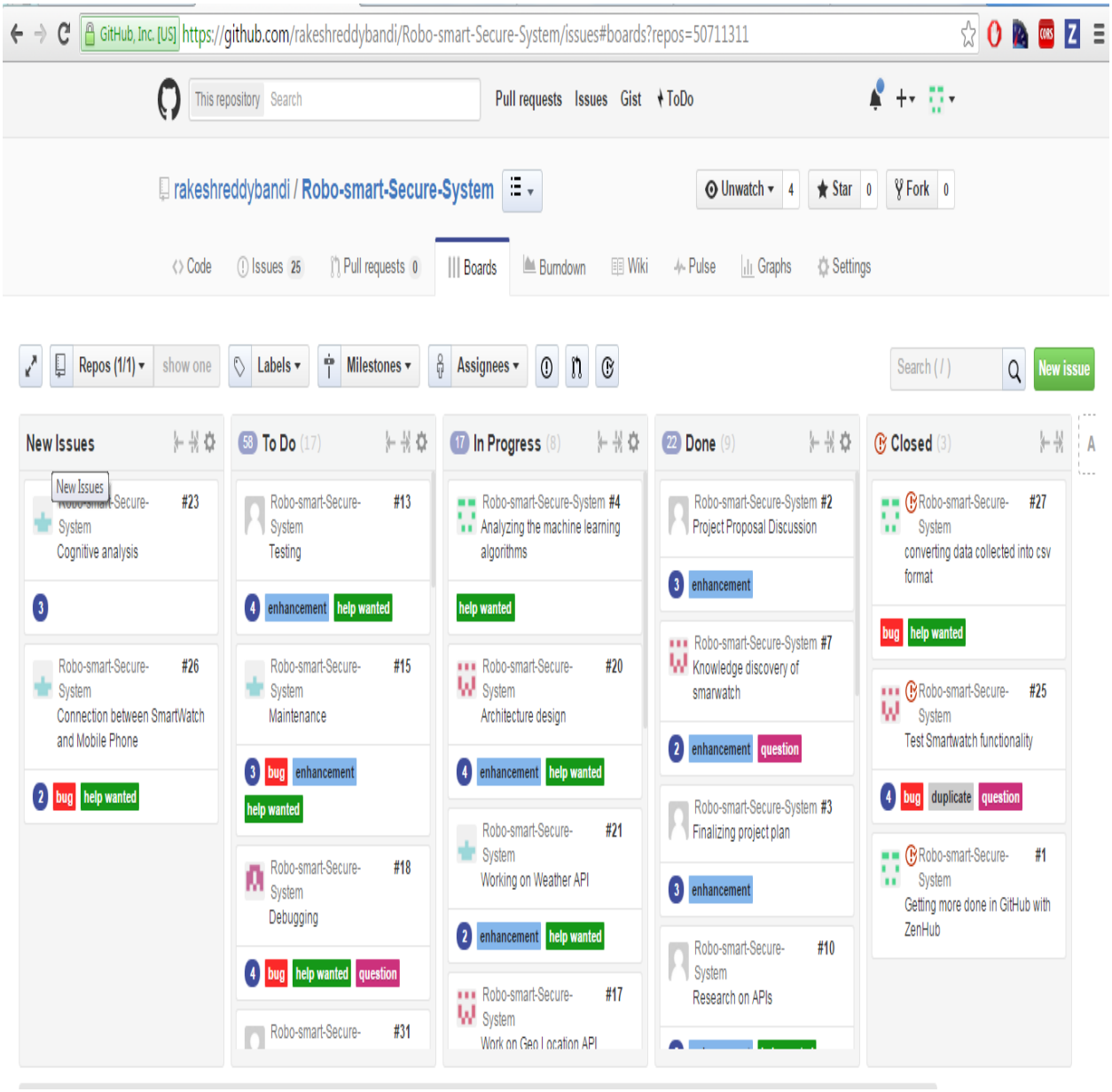
Screenshot 2:

The following screenshots shows the second listing of the To Do phases and In Progress where there is an increase in the number of elements and tasks to be done. There is an incentive in populating the respective tasks and assign them to the members as per core requirement and generative design.



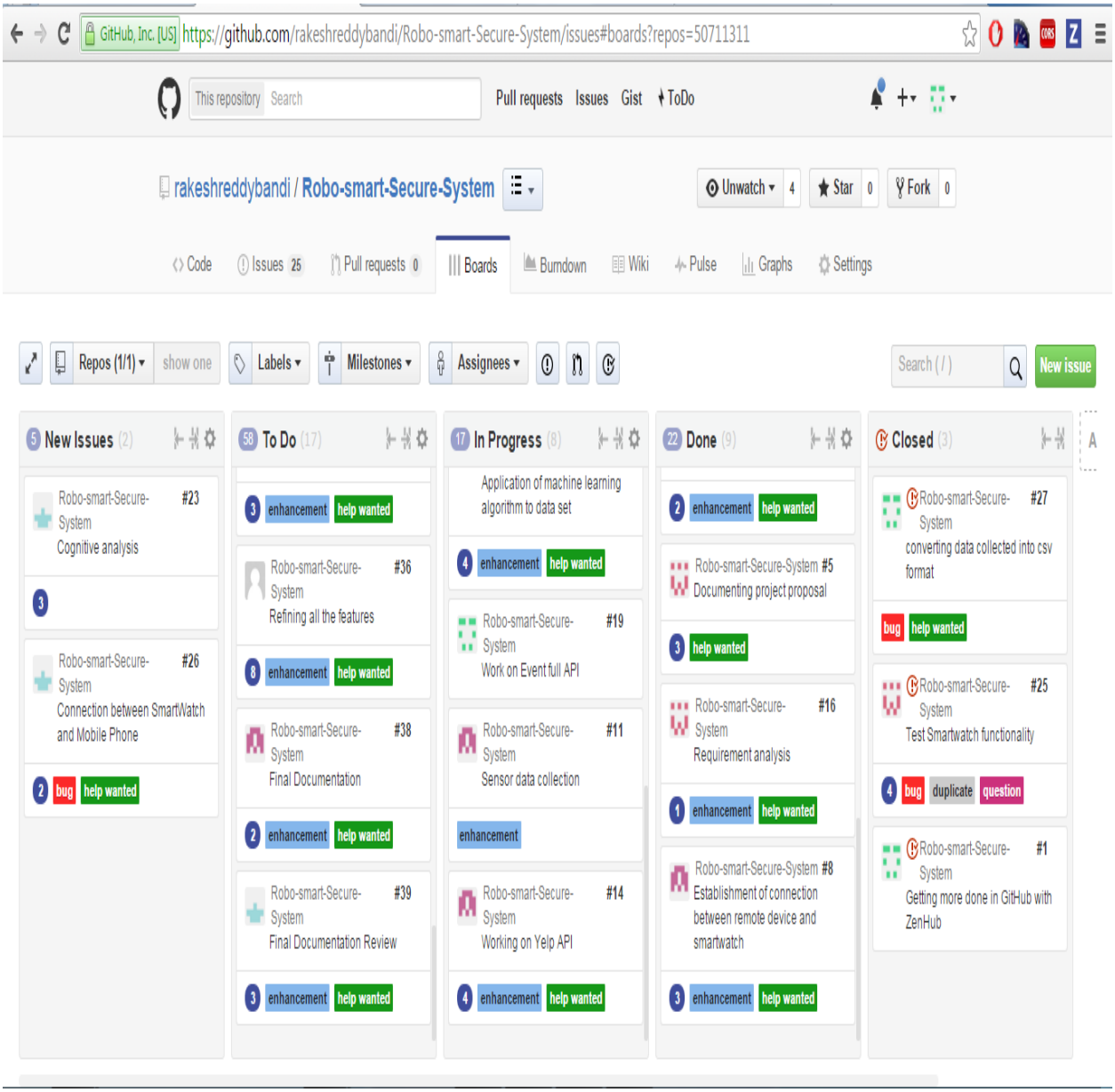
Screenshot 3:

The following screenshot shows the phase where the tasks in progress have been concentrated mainly on the design specifications. It combines along with the done tasks to stabilize the development of project in due course to simultaneously bring out the working modules.



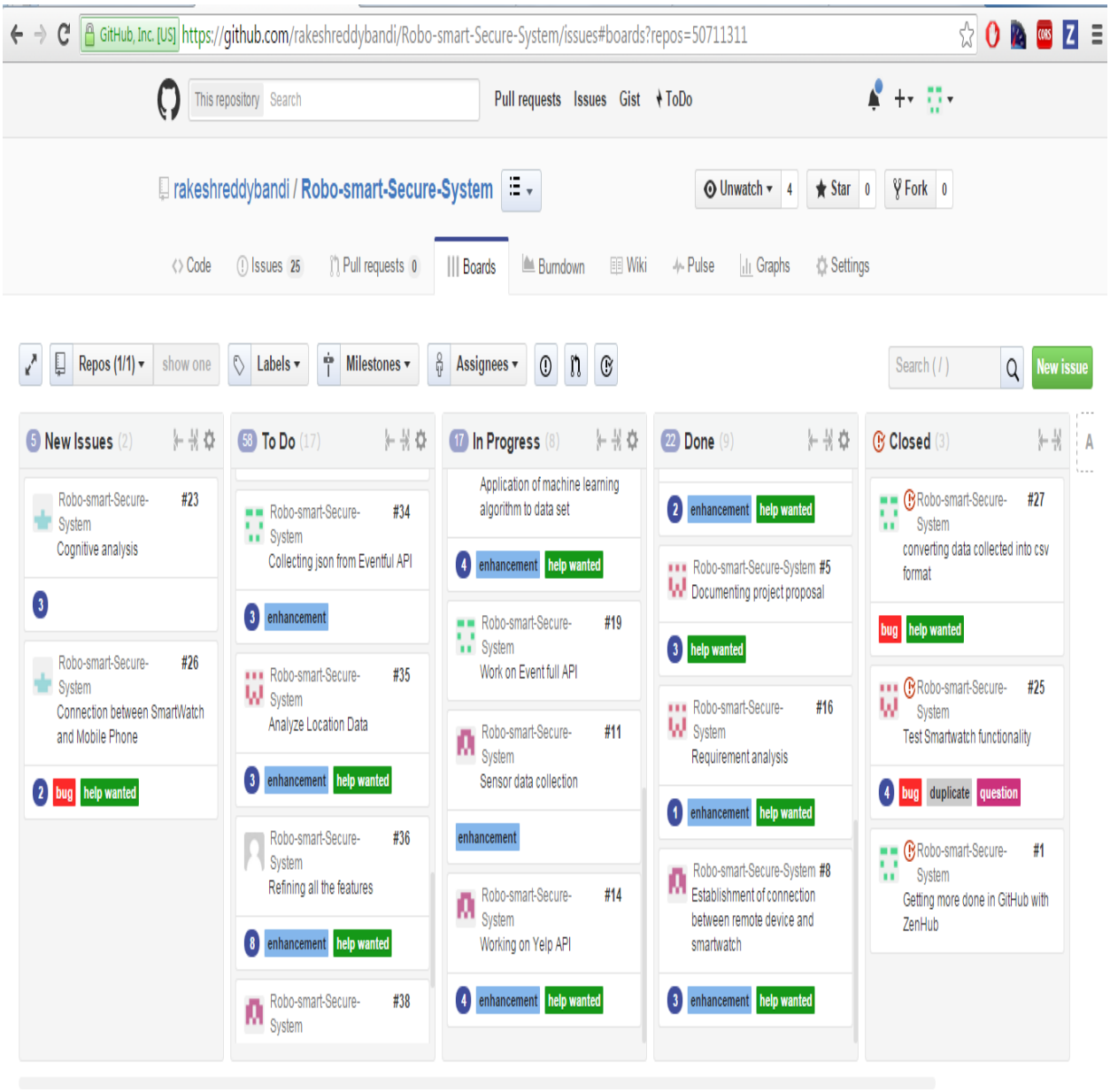
Screenshot 4:

The screenshot here shows the listing done where the penultimate requirement analysis and engineering is done to bring out the resulted output that is tested. The maintenance is to be done and especially debugging to negate results which are not in favour of project deployment.



Screenshot 5:

The entire project schedule has been roughly shown here. A detailed insight of project plan is done in such a manner that there might be few more additional development or phase build in due course of the project. It shows a correlation of the various modules included as per specification and features.

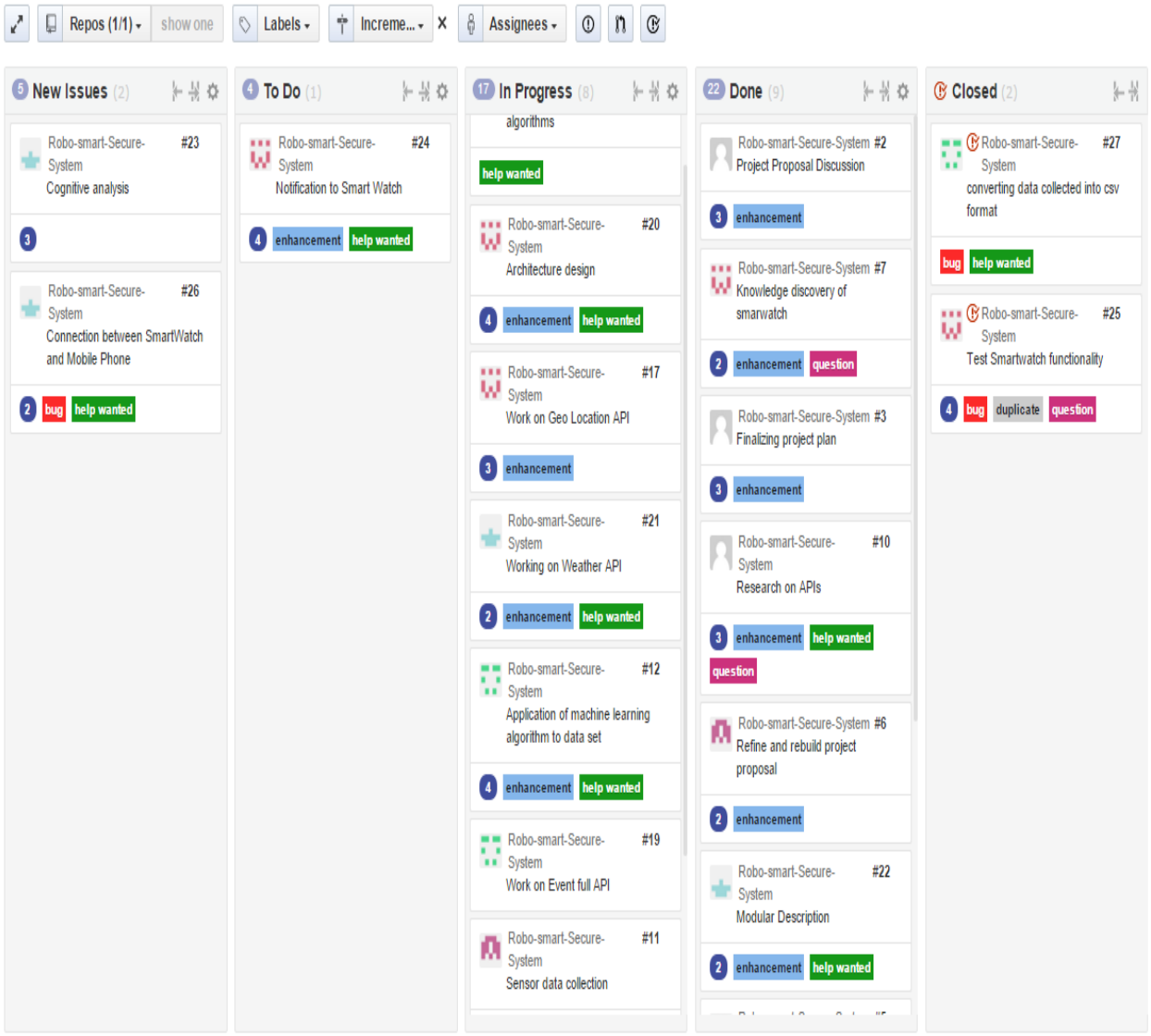


2. *Project Timeline:*

The implementation of the project is covered with a specific timeline and this is shown as follows to implement it using Zenhub:

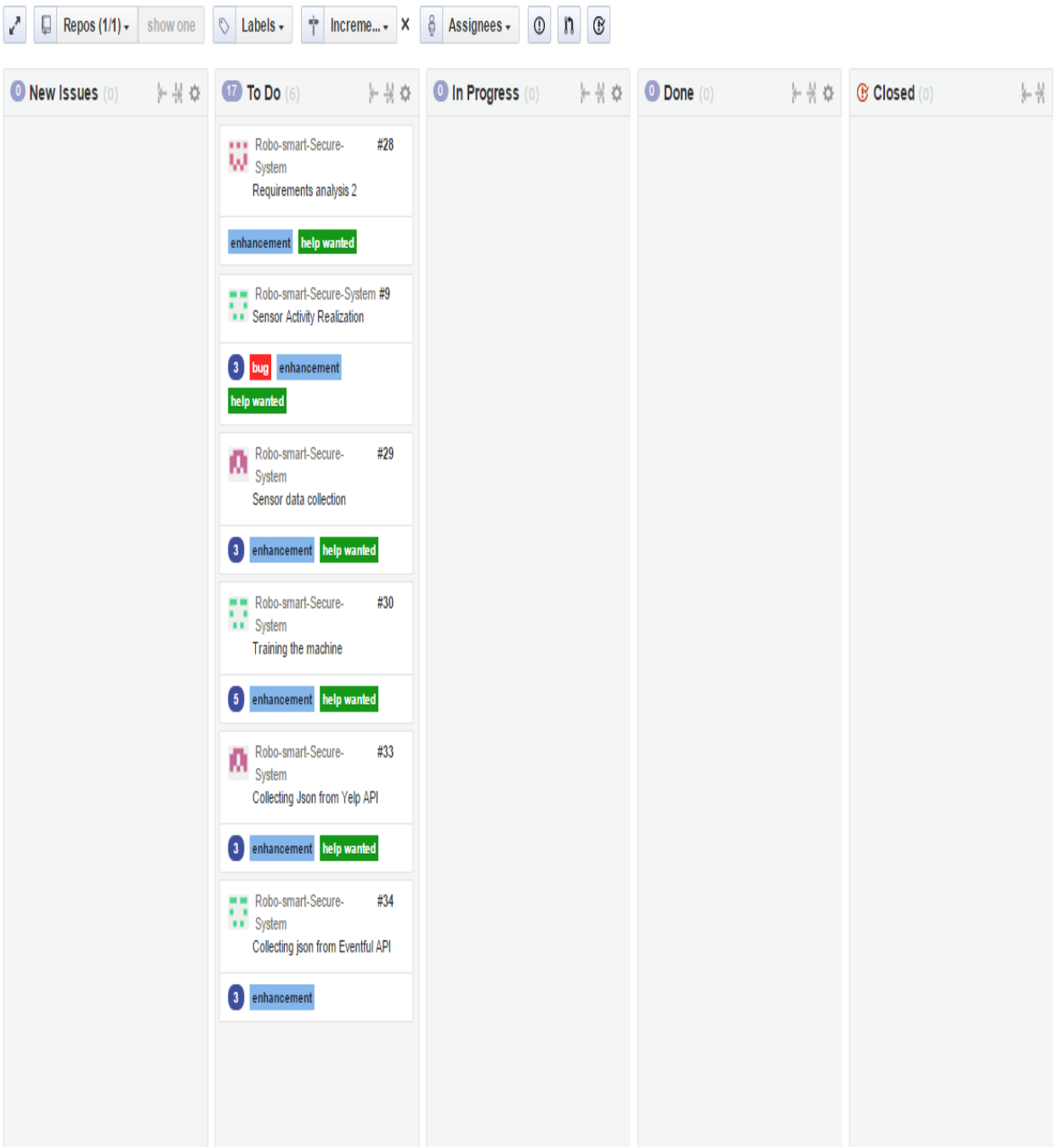
1. Increment 1:

The first increment has the following tasks to be completed by 02/19/2016 with tasks assigned as such:



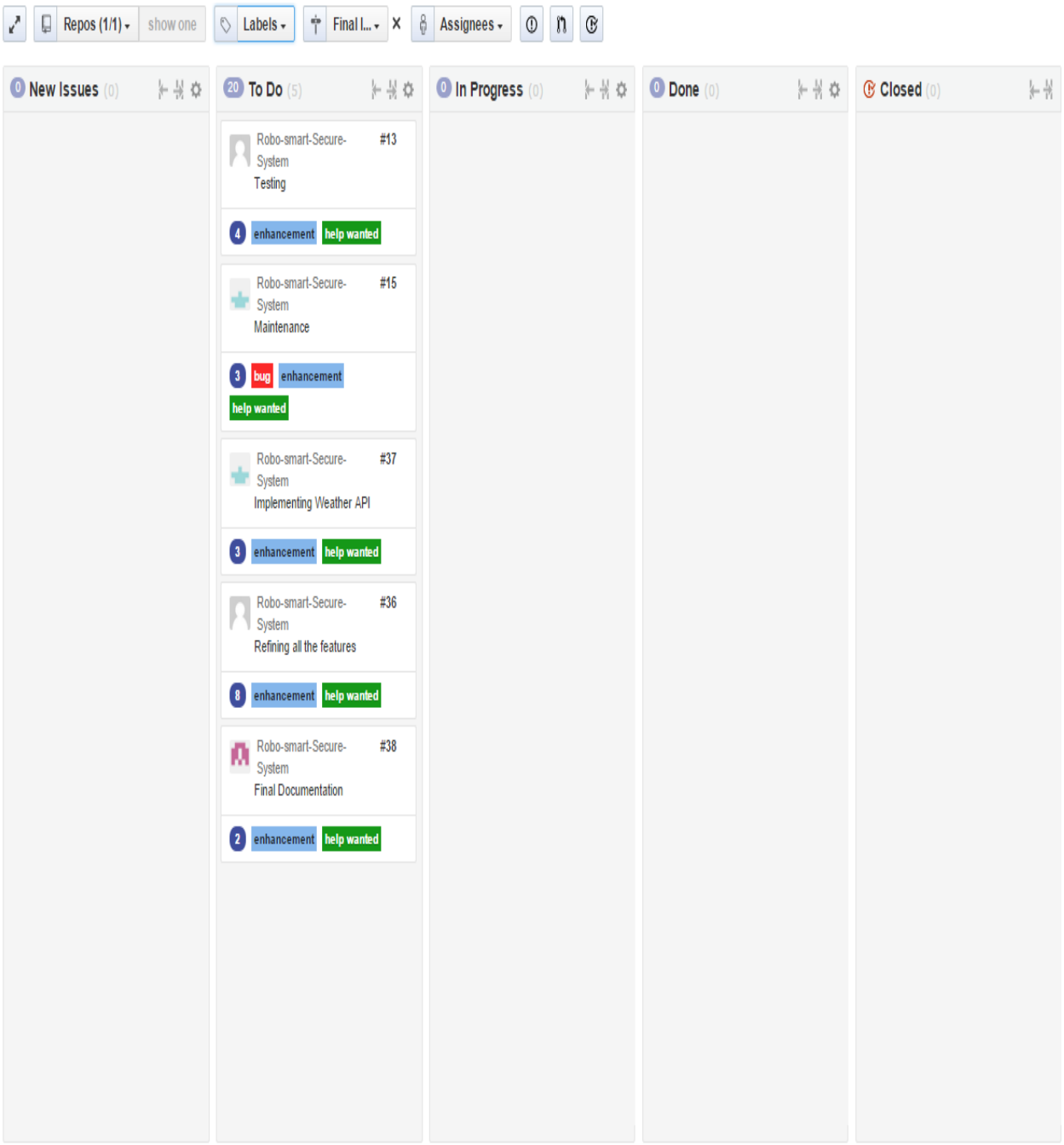
2. Increment 2:

The timeline for the second increment due on 03/11/2016 will have the graph as follows:



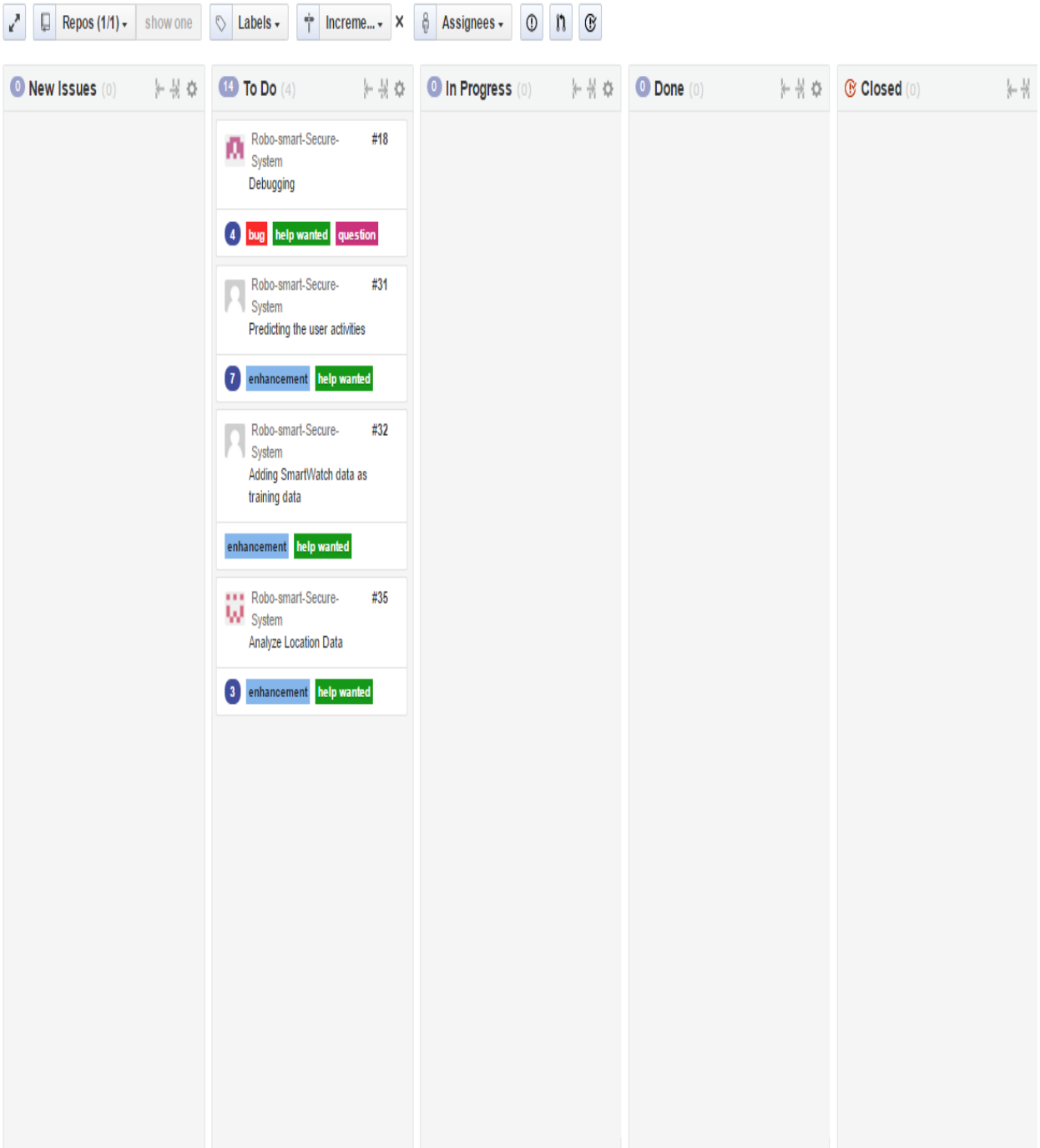
3. Increment 3:

The timeline for the third increment due on 04/06/2016 will have the graph as follows:



4. Final Increment:

The timeline for the final increment due on 04/29/2016 will have the graph as follows:

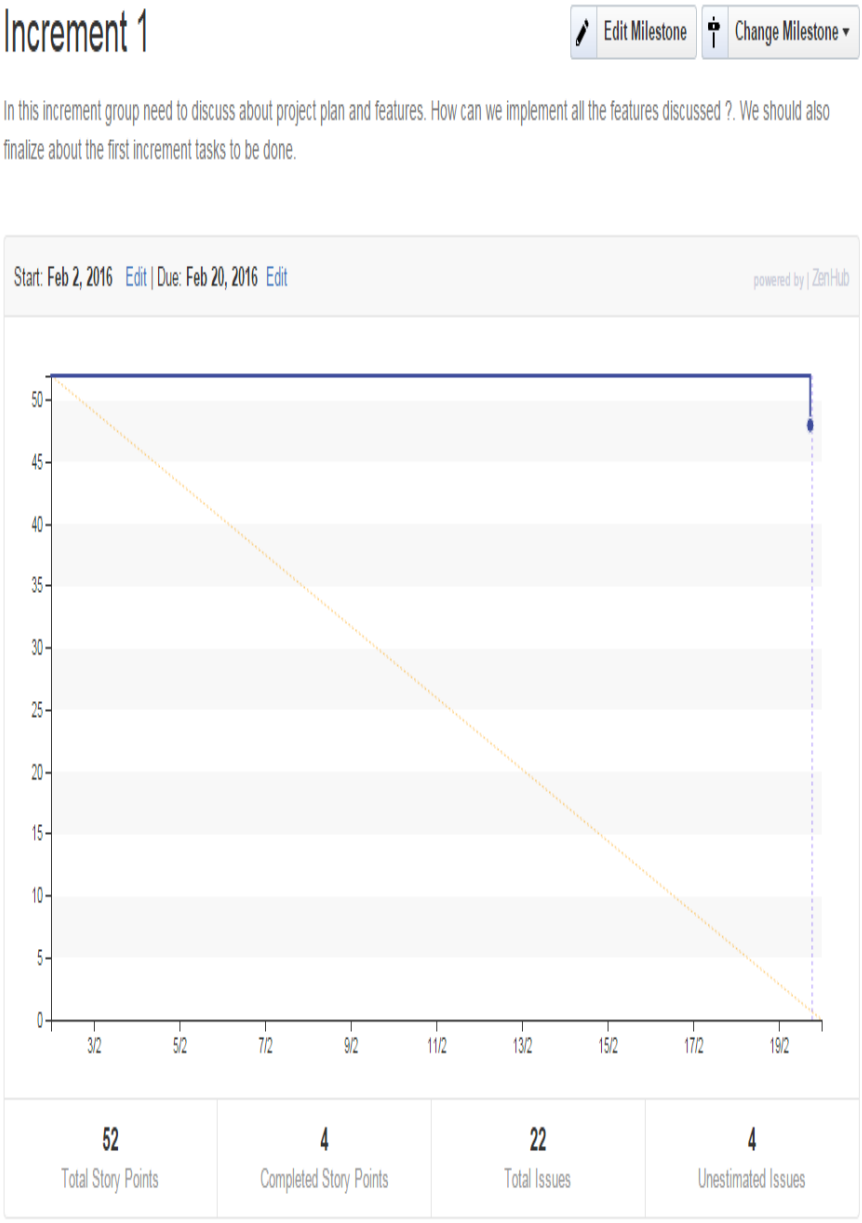


3. Burndown charts:

The burndown charts would basically represent the time deployment of project as per the various phases. It would instantiate the modules of time and its according development over the schedule.

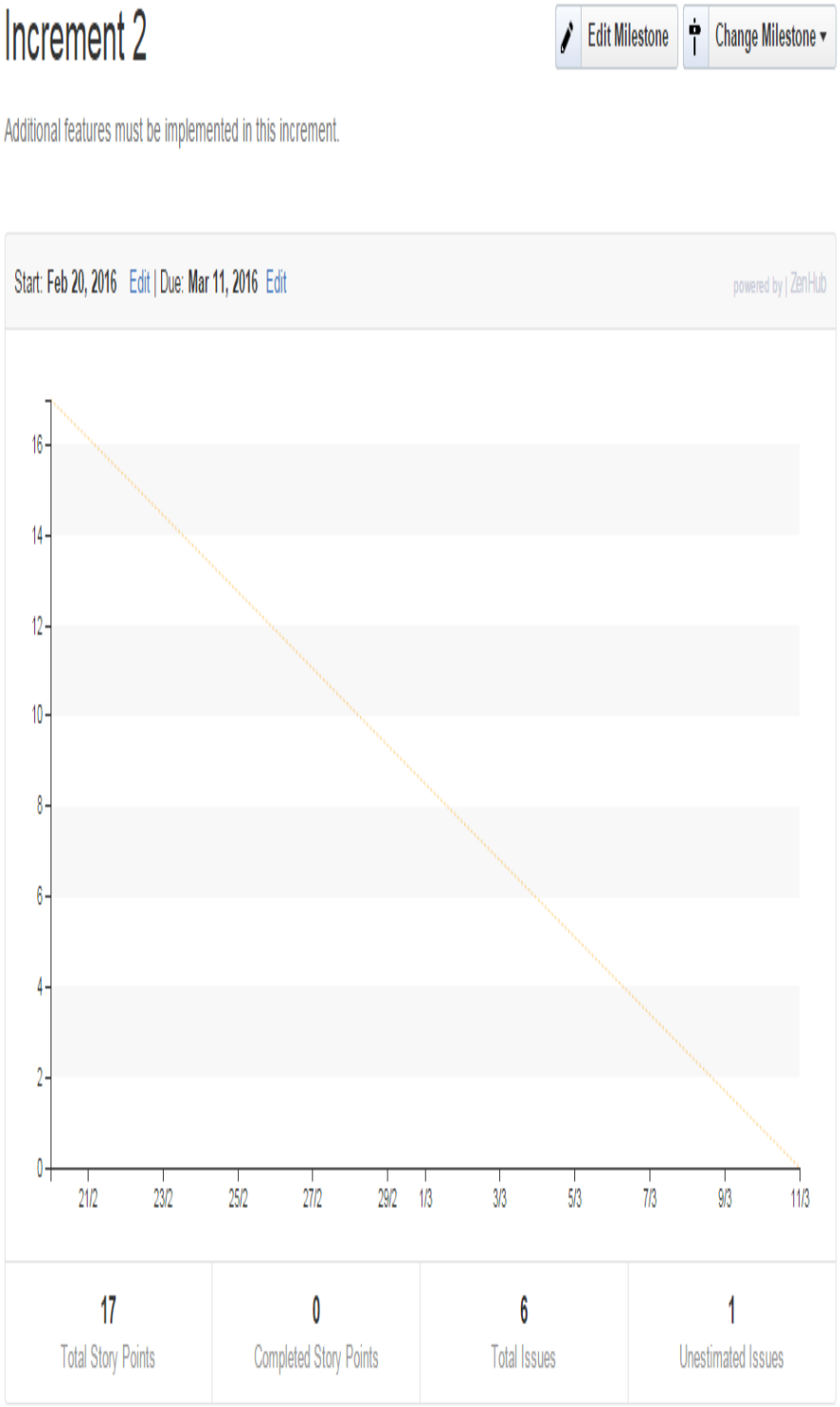
Increment 1:

The burndown chart of increment 1:



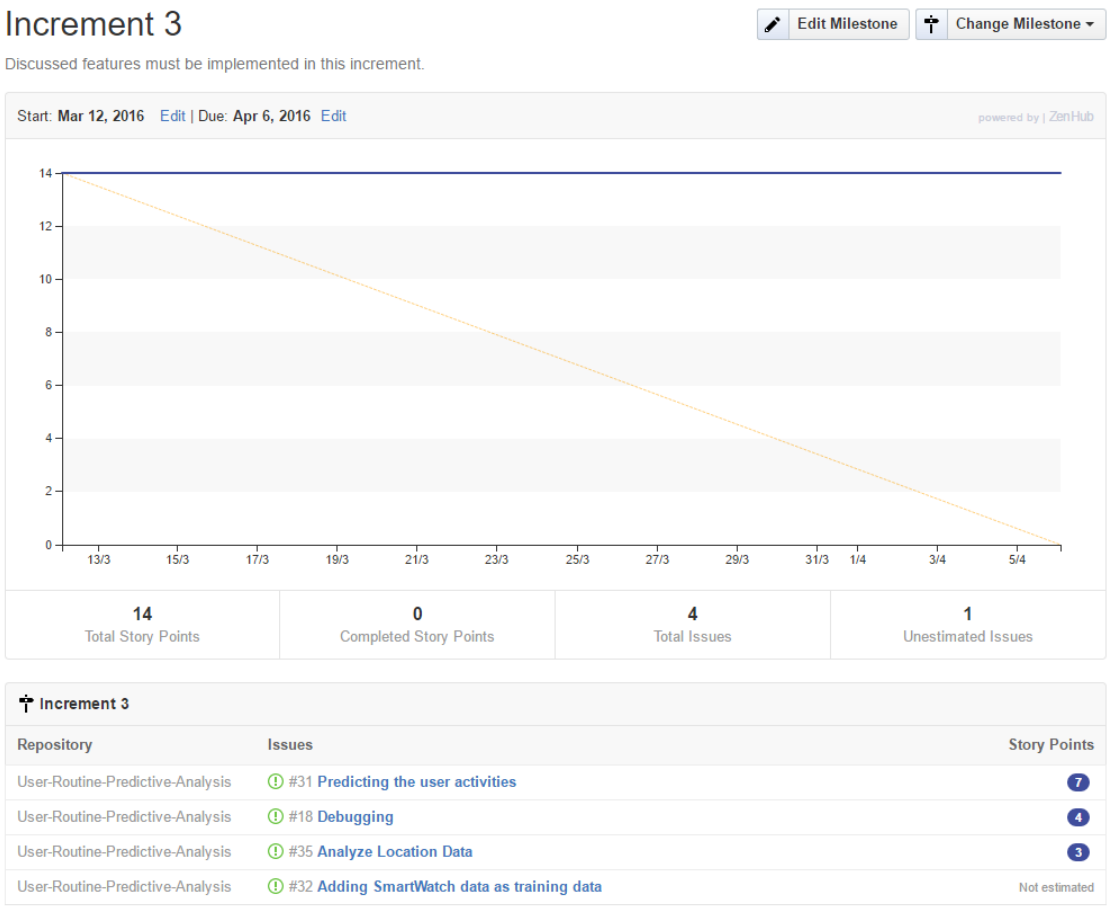
Increment 2:

The burndown chart of increment 2 is as follows:



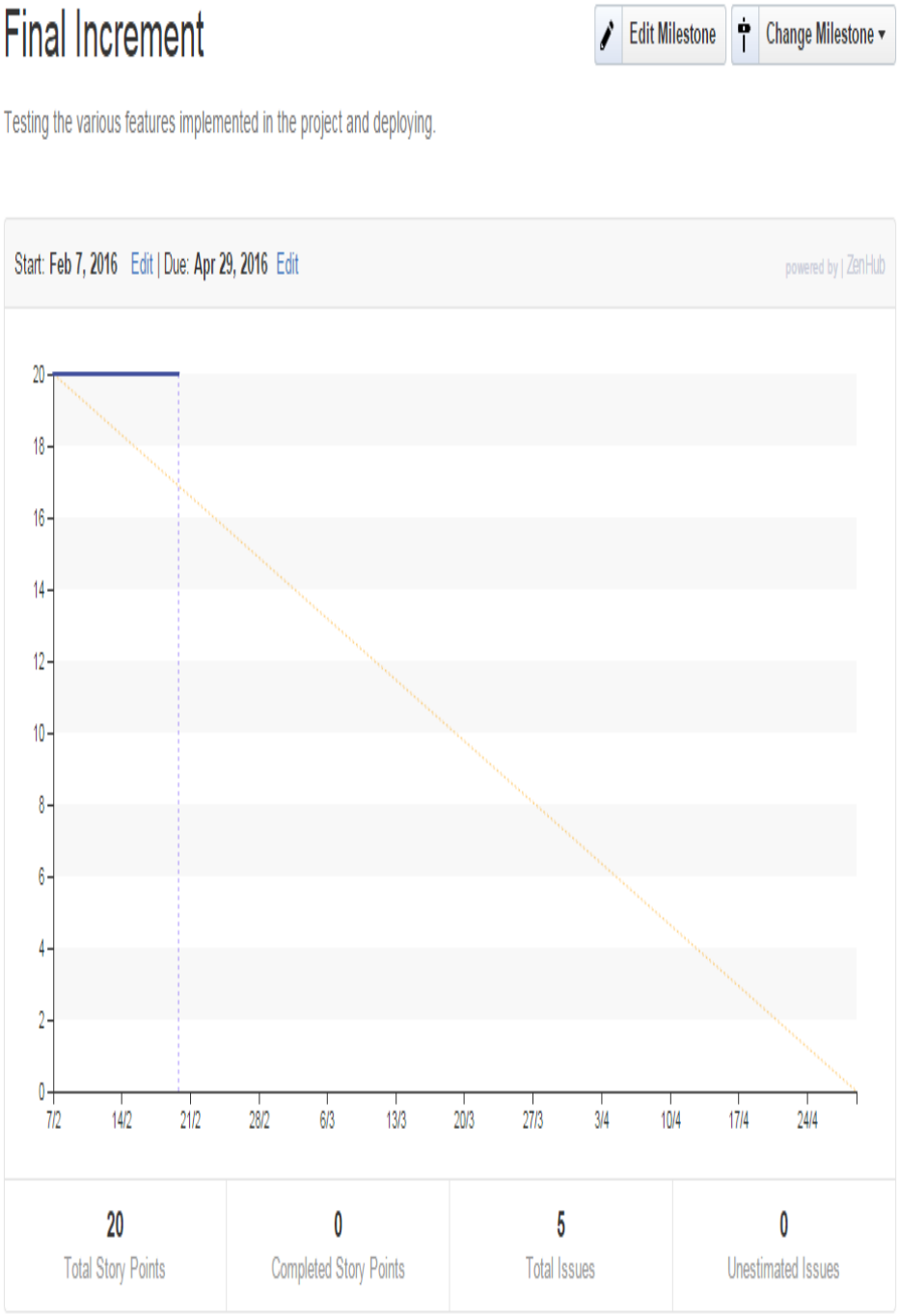
Increment 3:

The burndown chart of increment 3 is:



Final Increment:

The final increment has the burndown chart as follows:



3. Increment Reports

1. First increment Report:

In the first increment, we have implemented notification sending from the smartphone to the watch. The phone calls from the Alchemy API to retrieve entities, text, and image data. So far, the application can analyze text that is given to it. With this text, it can determine important entities, such as a name or a place. It can also look for keywords in the text. The application also is able to perform sentiment analysis, and can rate the text as positive or negative. The application can also look at images and classify them based on their characteristics. In our testing, the image is successfully classified. Once these features are performed, the smart watch is given a notification with the analysis.

We have also written a Naïve Bayes algorithm in R to classify datasets of accelerometer data. We used online repositories to retrieve the data we needed. In this repository, we have accelerometer data on walking, using the telephone, standing up from chair, sitting in chair, pouring water, lying in bed, getting out of bed, eating soup, eating meat, drinking out of a glass, descending from the stairs, combing hair, climbing the stairs, and brushing teeth. For this increment, we chose walking, getting out of bed, and climbing the stairs to analyze. We have also begun collecting the data from the smartwatch user. We used the same movements—walking, getting out of bed, and climbing the stairs—to measure activity. We loaded these data sets into R, wrote a code for naïve Bayes algorithm, and implemented the code. We classified the data by choosing the classification with the highest probability. This implementation is still separate from the phone application, but we plan on doing so in the next increment.

1. Implementation

Mobile Client Implementation (Smart Watch, Smart Phone, Robot): Mobile Application for Collecting the Sensor Data:

- We have developed an application which is specifically for collecting the user smart phone and smart watch sensor data.
- In this application we have collected the Sensor data and we stored in to the csv files in the Device memory.
- Later we use this data to train over model to predict the user patterns regularly.
- This application mainly gets the all sensor data values from the Sensors such as Gyroscope, Accelerometer, Proximity, Light, Magnetic and Heart Rate in the Smart watch and smart phone.
- Apart from this we also collected the Location's latitude and longitude at which the sensor data collected and we store the data in csv file.

Text Analysis and Image Analysis Application:

- We have implemented the Text and Image analysis of Data and Image respectively using the alchemy API.
- The Smart phone calls from the Alchemy API to retrieve entities, text, and image data.
- The application can analyze text that is given to it. With this text, it can determine important entities, such as a name or a place. It can also look for keywords in the text.
- The application also is able to perform sentiment analysis, and can rate the text as positive or negative. The application can also look at images and classify them based on their characteristics. In our testing, the image is successfully classified. Once these features are performed, the smart watch is given a notification with the analysis.

Machine Learning Application:

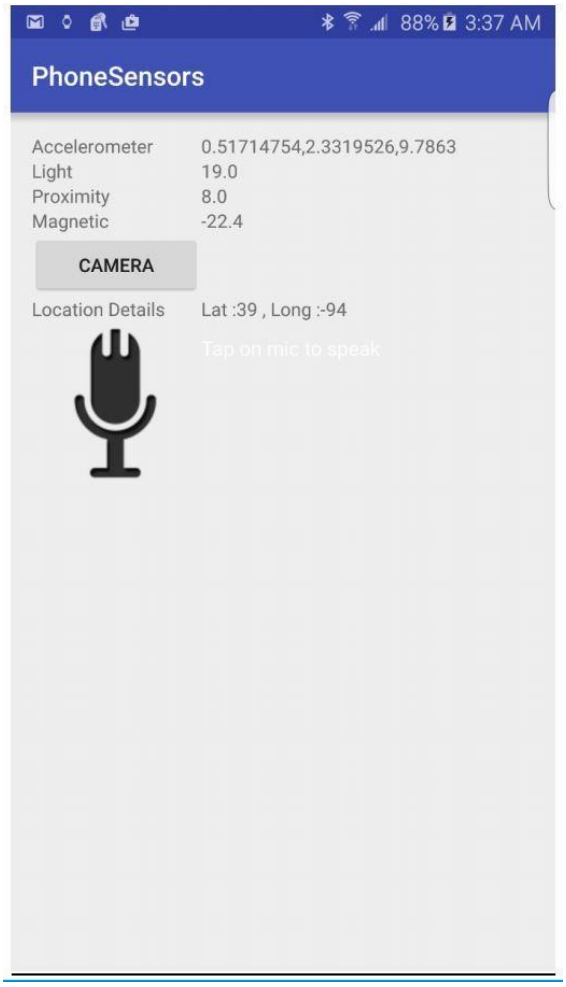
Machine Learning Classification Algorithm- Naïve Bayes Classification Algorithm to predict the User Activity:

- We have predicted the user activity using the Naïve Bayes Classification algorithm.

- We have trained the model based on the data obtained for different activities such as walking, getting out of bed and climbing up the stairs.
- So using the training data, we have predicted the user activity for the particular accelerometer readings using the Naïve Bayes Classification algorithm.
- Naïve Bayes Classification algorithm:
- Calculate the Mean and Standard deviation for each of column in the data set.
- The likelihood for each of the axis for activity is calculated.
- The likelihood of each of the axis reading was calculated using the Gaussian distribution function.
- Using the Naïve Bayes posterior probability formula we have computed the probability of occurrence of posterior.
- Posterior probability is directly proportional to product of prior probability and likelihood.
- Using the posterior probability we have predicted which activity is most likely to perform.

2. Deployment:

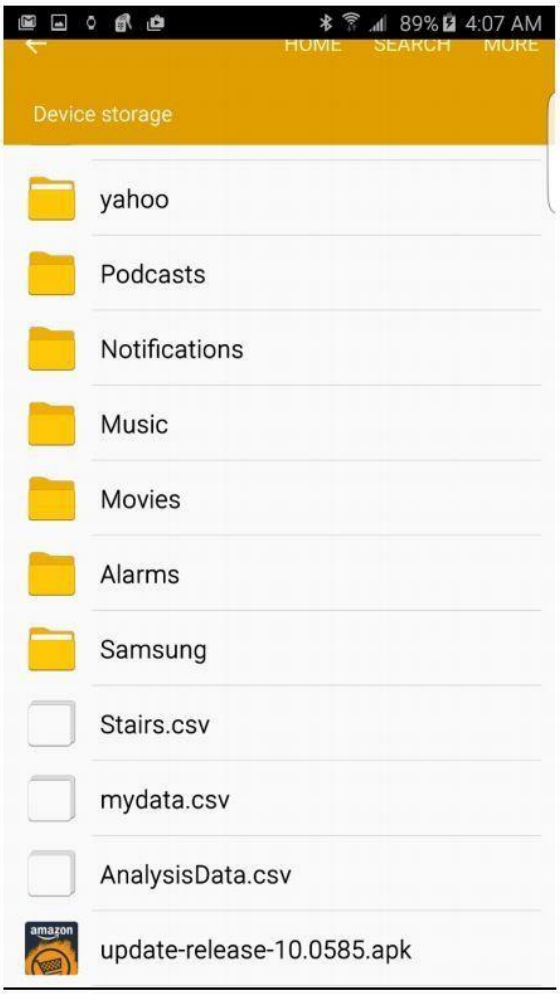
1. SmartPhone/SmartWatch Sensors Data Storage Application:



The various sensors in both mobile and smart watch gives the data values continuously for the user activities.

We store these values in the Internal storage of the Device.

2. SmartPhone/SmartWatch Sensors Data Storage Application File Explorer View:



We can see the data sets as .CSV files in the Internal memory of Android Devices.

Machine Learning Model:

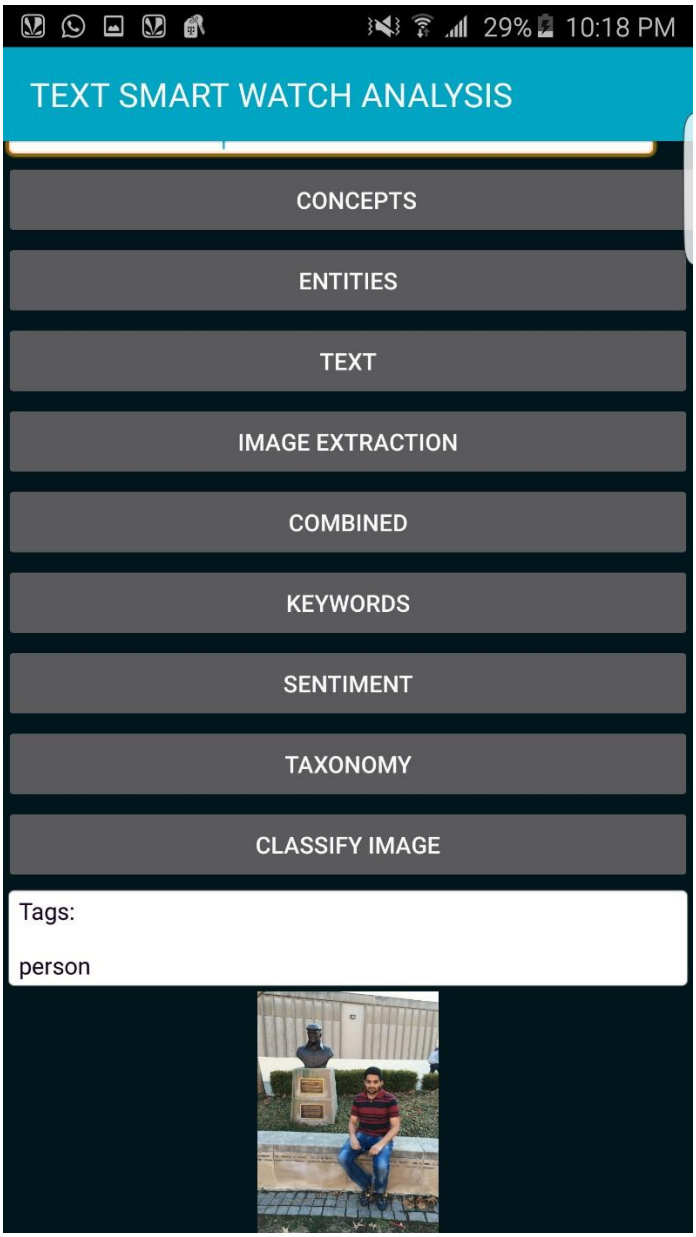
1. Model predicting the User Activity as Getting up the Bed

2. Model Predicting the User Activity as Climbing the Stairs

3. Model Predicting the User Activity as Walking

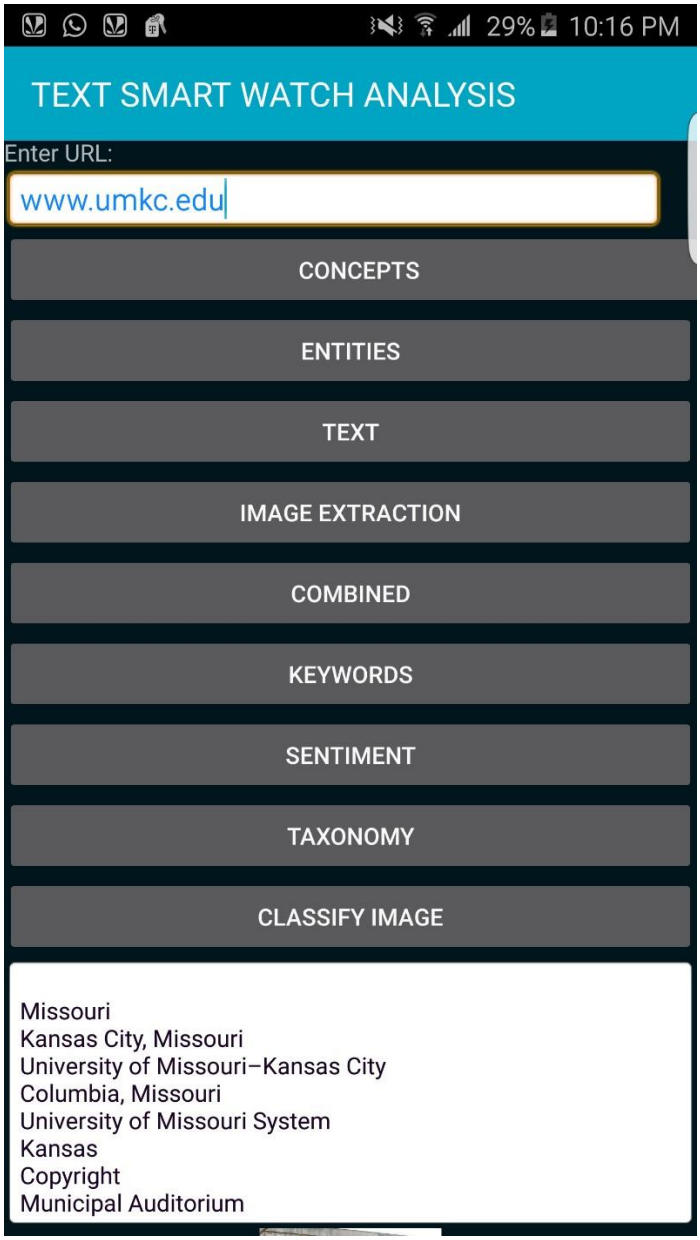
TEXT AND IMAGE ANALYSIS APPLICATION-ALCHEMY API:

1.



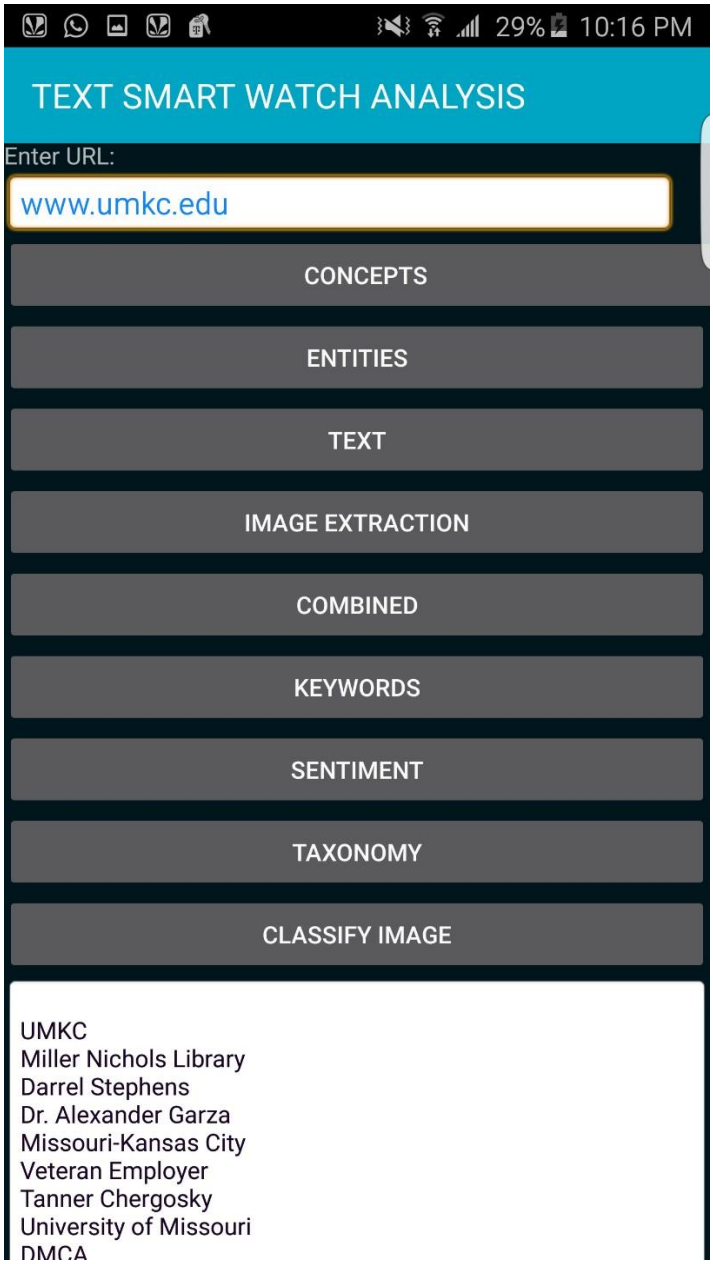
The above figure illustrates about classify image button where it does image analysis of the image provided and gives respective tags.

2.



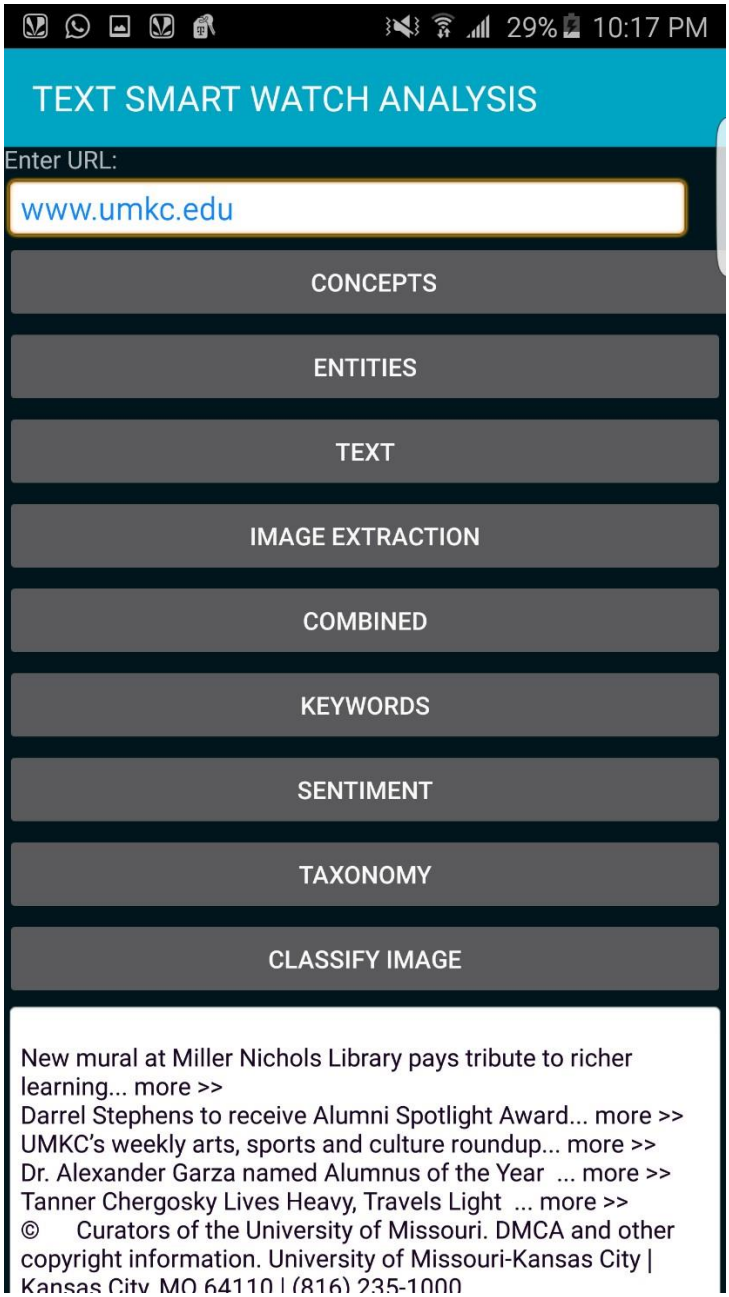
The above figure explains about the **concepts of** URL mentioned in the text box.

3.



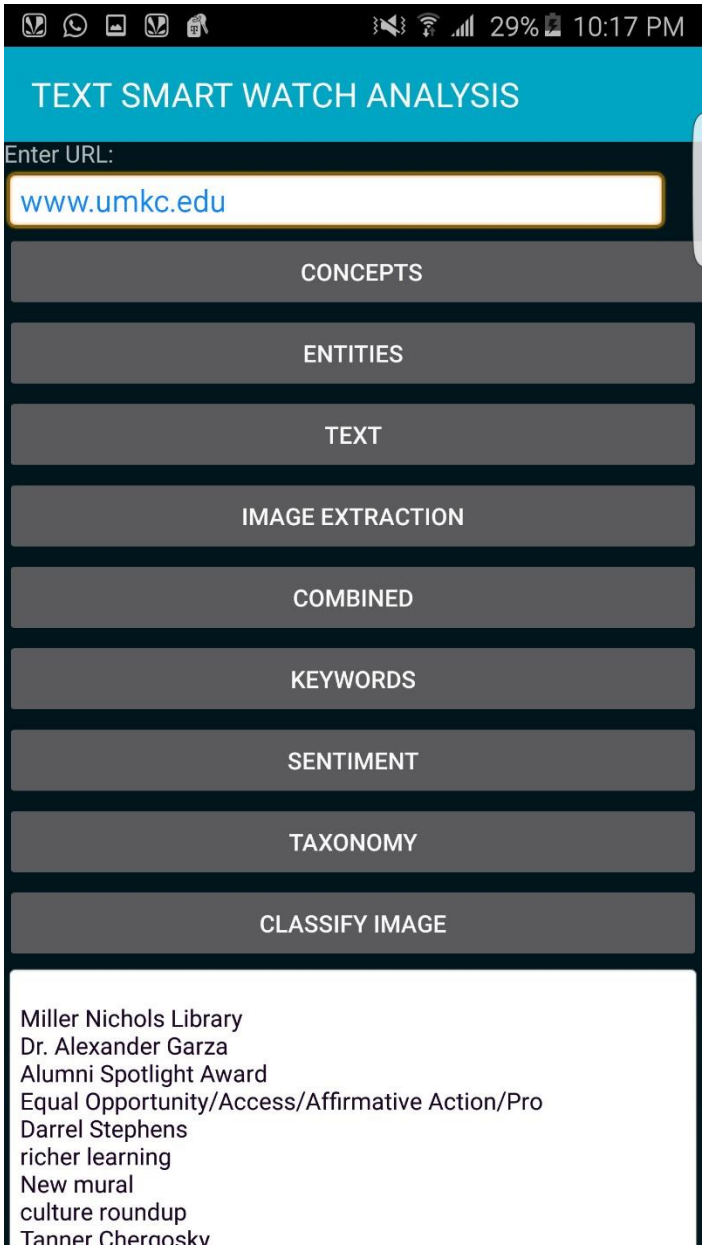
The above figure illustrates about the various **entities** in the URL.

4.



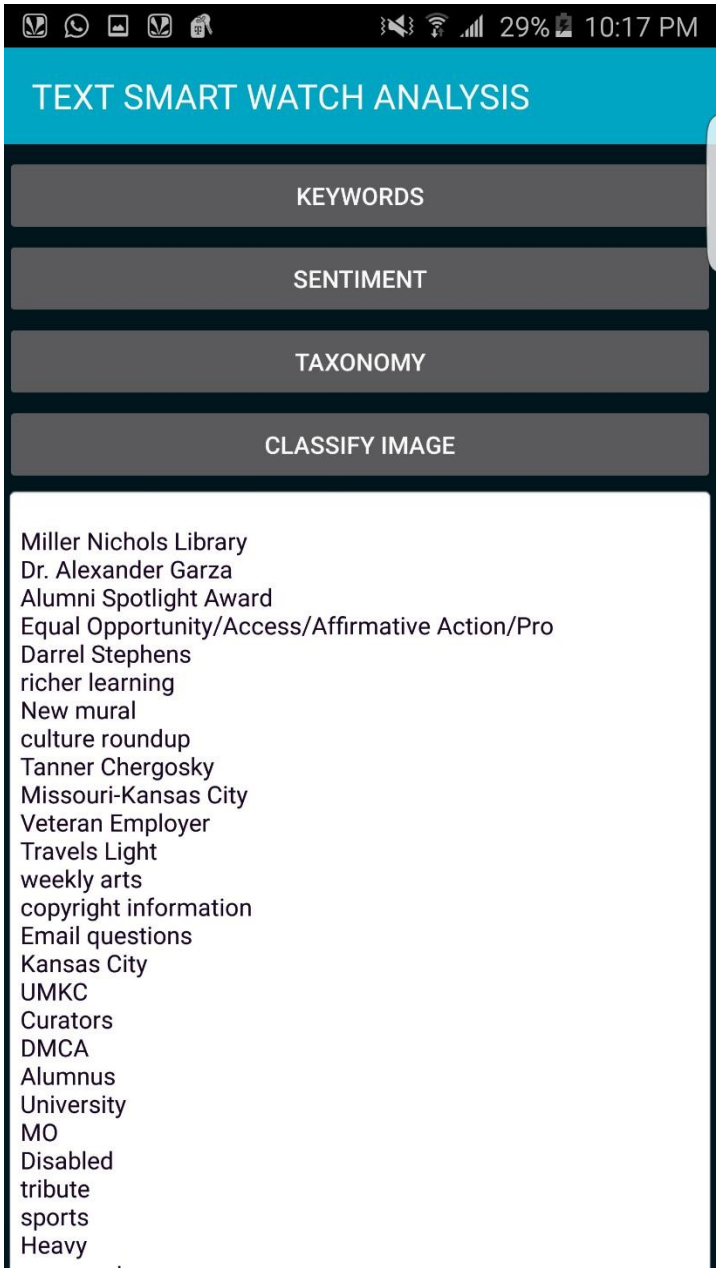
The above figure explains about the **text** present in the URL.

5.



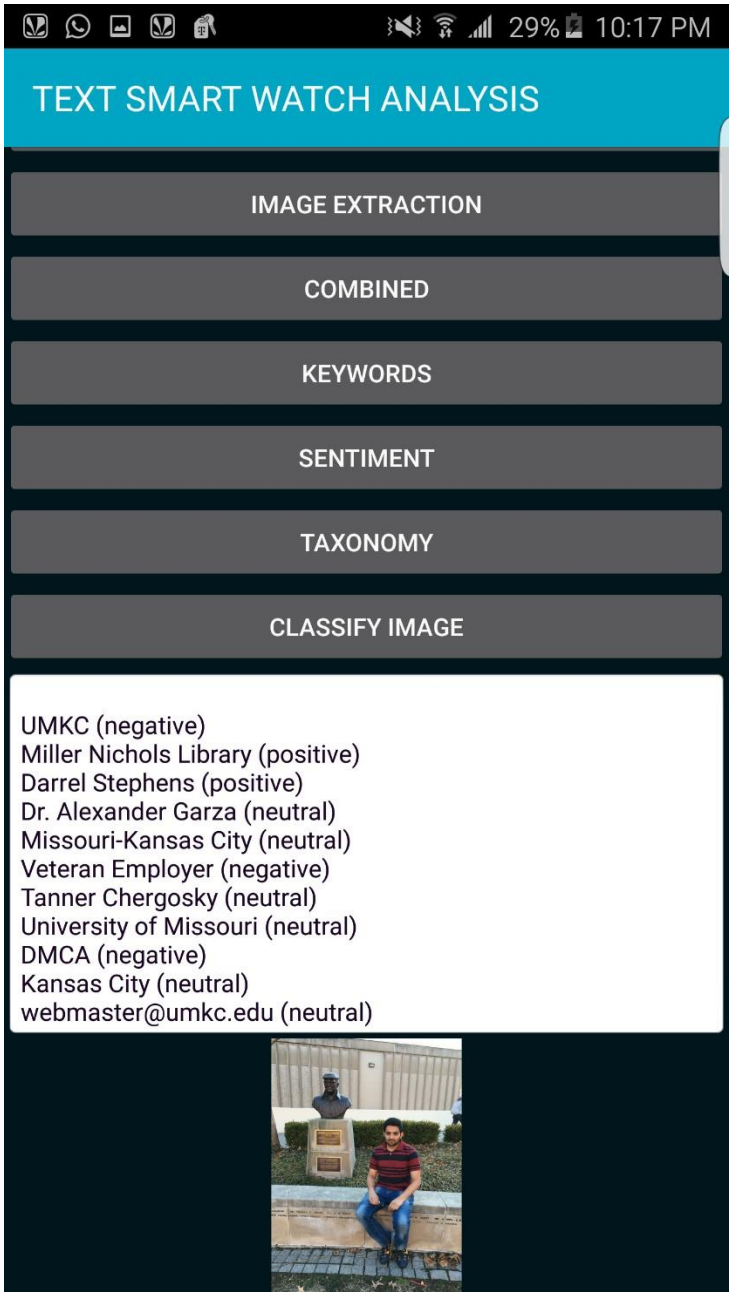
The above figure explains about the **combined** feature of the application.

6.



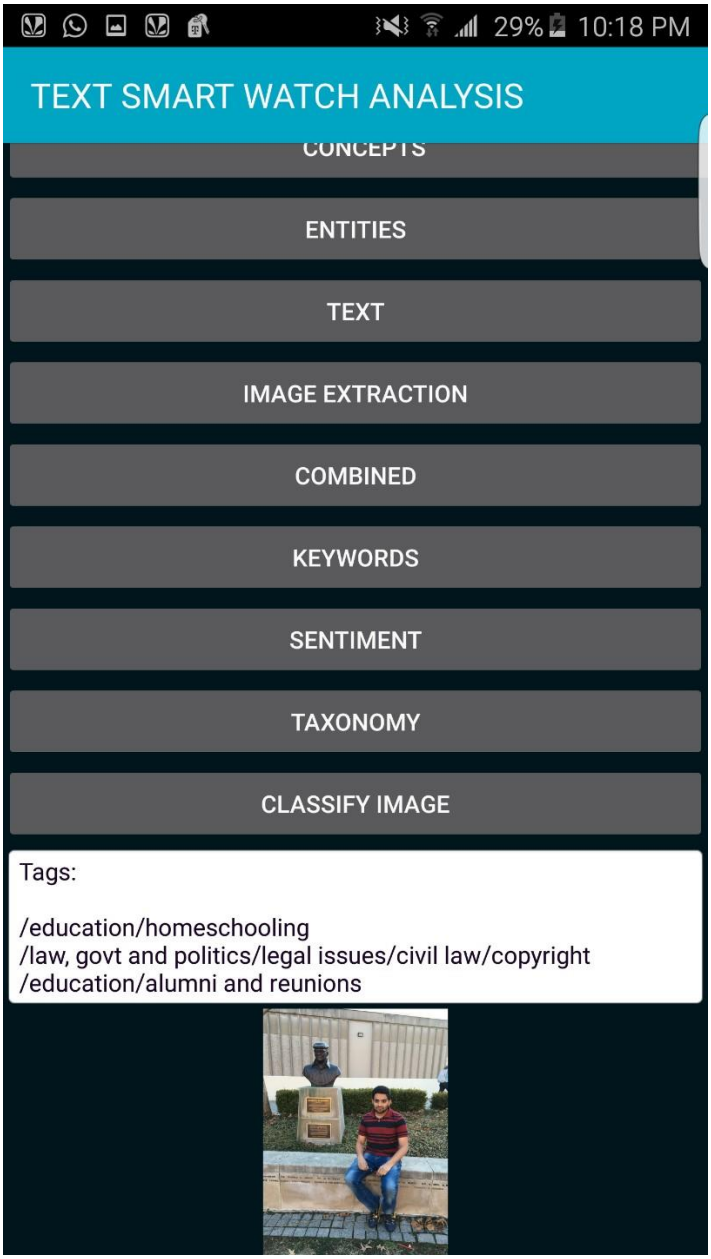
The above figure explains about the **keywords** button in the application.

7.



The above figure explains about **sentimental analysis** in application.

8.



The above figure illustrates about the **taxonomy** of the URL.

2. Second Increment Report:

In this increment, we have focused on extending our Naïve Bayes algorithm from increment one and implementing it in Spark. Although R is a great tool for testing this algorithm, we need the capability of streaming large amounts of data dynamically moving forward. Spark streaming handles this quite well.

We have also extended our project to include sentence data. Eventually, we want to be able to recommend restaurants to the user based on their location and Restaurants nearby that users are praising. In this increment, we have created a program that can read text and make recommendations sent back to the phone.

Because our goal is to recommend restaurants to the User based on sentences sent in the application, we need to do two things. First, we have tested a Semantic analyzer that will tell us if the sentence is positive or negative. After collecting this information about the sentence, we use a Collaborative Filter algorithm to rank restaurants based on the user's preferences.

Finally, we must send these recommendations to the user. We have extended the notification sending from the previous lab and extended it to the restaurant rankings.

Moving forward, we must accomplish a few goals. First of all, we must improve the quality of the user interface. The user interface is key to making our application a product that people will want to use. Next, we need to integrate several of the features that we have been developing independent of one another. This project will only be successful if all of these features work together efficiently. We also must extend our Machine learning algorithm to include the location of the user and the time of day.

1.Implementation

Mobile Client Implementation (Smart Watch, Smart Phone, Robot):

Mobile Application for Collecting the Sensor Data:

- We have developed an application which is specifically for collecting the user smart phone and smart watch sensor data.
- In this application we have collected the Sensor data and we stored in to the csv files in the Device memory.
- Later we use this data to train over model to predict the user patterns regularly.
- This application mainly gets the all sensor data values from the Sensors such as Gyroscope, Accelerometer, Proximity, Light, Magnetic and Heart Rate in the Smart watch and smart phone.
- Apart from this we also collected the Location's latitude and longitude at which the sensor data collected and we store the data in csv file.

Sending and receiving data to Spark:

- We have developed an application which sends a string of text to Spark
- Spark analyzes the data, and returns a positive or negative sentiment using Sentiment Analysis
- Based on the statement, we are recommending to the user a list of movies

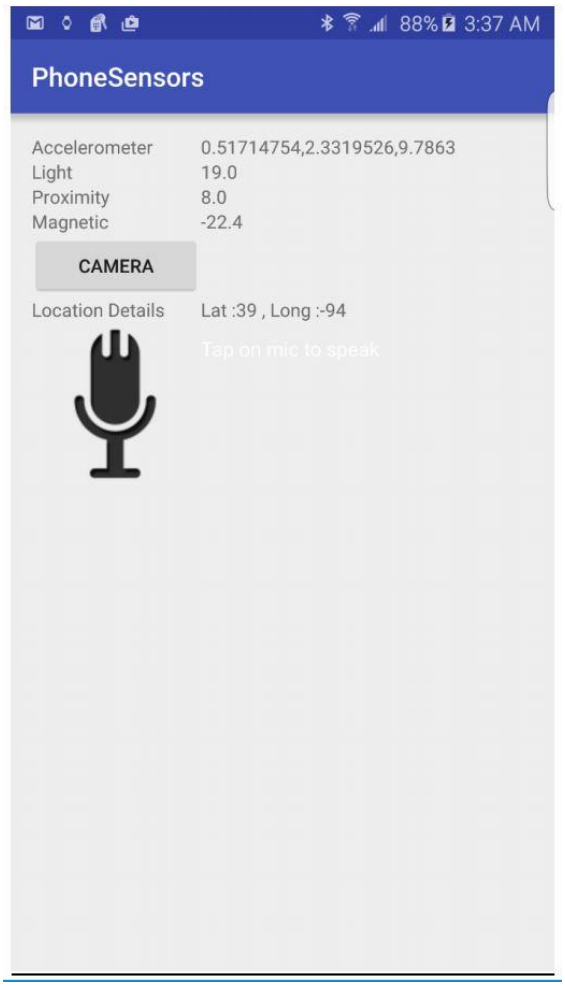
Machine Learning Application:

Machine Learning Classification Algorithm- Naïve Bayes Classification Algorithm to predict the User Activity:

- We have predicted the user activity using the Naïve Bayes Classification algorithm.
- We have trained the model based on the data obtained for different activities such as walking, getting out of bed and climbing up the stairs.
- So using the training data, we have predicted the user activity for the particular accelerometer readings using the Naïve Bayes Classification algorithm.
- Naïve Bayes Classification algorithm:
    1. Calculate the Mean and Standard deviation for each of column in the data set.
    2. The likelihood for each of the axis for activity is calculated.
    3. The likelihood of each of the axis reading was calculated using the Gaussian distribution function.
    4. Using the Naïve Bayes posterior probability formula we have computed the probability of occurrence of posterior.
    5. Posterior probability is directly proportional to product of prior probability and likelihood.
- Using the posterior probability we have predicted which activity is most likely to perform.
- The Machine learning algorithm runs using Spark

2. Deployment:

1. SmartPhone/SmartWatch Sensors Data Storage Application:

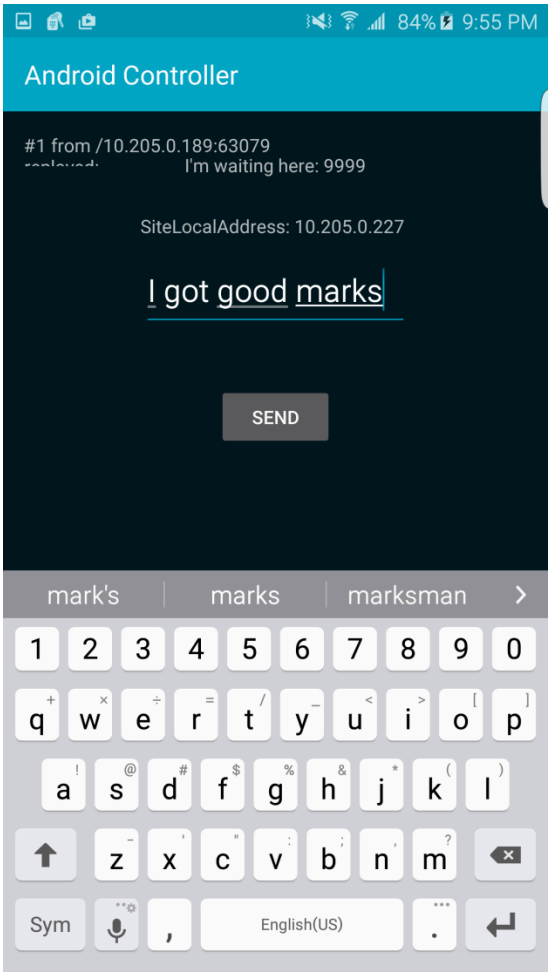


The various sensors in both mobile and smart watch gives the data values continuously for the user activities.

We store these values in the Internal storage of the Device.

2. Sending Data to Spark :

In this application, the user can type a command and send it to Spark.



Here we see that the message has been received.

```
16/03/11 21:55:10 INFO DAGScheduler: ResultStage 1 (collect at MainStreaming.scala:25) finished in 0.027 s
16/03/11 21:55:10 INFO DAGScheduler: Job 8 finished: collect at MainStreaming.scala:25, took 0.064004 s
I got good marks
16/03/11 21:55:10 INFO BlockManagerInfo: Removed broadcast_1_piece0 on localhost:63060 in memory (size: 1037.0 B, free: 1127.2 MB)
16/03/11 21:55:10 INFO ContextCleaner: Cleaned accumulator 2
16/03/11 21:55:10 WARN : Your hostname, DESKTOP-P023Q7N resolves to a loopback/non-reachable address: fe80:0:0:0:0:5efe:acd:bd%net5, but we couldn't find any ext
16/03/11 21:55:11 INFO ReceiverSupervisorImpl: Starting receiver again
16/03/11 21:55:11 INFO ReceiverTracker: Registered receiver for stream 0 from 10.205.0.189:63018
16/03/11 21:55:11 INFO ReceiverSupervisorImpl: Starting receiver
```

3. Sentiment Analysis

Once the message is received, we analyze the tweet for positive or negative sentiment. Below we show the code that we used.

```
    .setAppName("SparkStreaming")
    .set("spark.executor.memory", "4g").setMaster("local[*]")
  val ssc= new StreamingContext(sparkConf,Seconds(2))
  val sc=ssc.sparkContext
  val ip=InetAddress.getByName("10.205.0.227").getHostName
  val lines=ssc.socketTextStream(ip,9999)
 // lines.saveAsTextFiles("output")
 val command= lines.map(x=>{
   x.toString()

 })
 command.foreachRDD(
 rdd=> rdd.collect().foreach(text => {

   println(text)
    rdd.saveAsTextFile("output")
    val sentimentAnalyzer: SentimentAnalyzer = new SentimentAnalyzer
    val tweetWithSentiment: TweetWithSentiment = sentimentAnalyzer.findSentiment(text)
    System.out.println("SENTIMENT is"+tweetWithSentiment)
    if(tweetWithSentiment.toString().contains("positive")){
      SimpleRecommendation.recommend(rdd.context )
    }

 })
```

And the results:

```
16/03/11 21:55:13 INFO ReceiverSupervisorImpl: Stopped receiver 0
 done [1.3 sec].
Adding annotator sentiment
16/03/11 21:55:14 INFO JobScheduler: Added jobs for time 1457754914000 ms
SENTIMENT isTweetWithSentiment [line=I got good marks, cssClass=sentiment : positive]
16/03/11 21:55:14 INFO MemoryStore: Block broadcast_3 stored as values in memory (estimated size 59.6 KB, free 104.8 KB)
16/03/11 21:55:14 INFO MemoryStore: Block broadcast_3_piece0 stored as bytes in memory (estimated size 13.8 KB, free 118.6 KB)
16/03/11 21:55:14 INFO BlockManagerInfo: Added broadcast_3_piece0 in memory on localhost:63060 (size: 13.8 KB, free: 1127.2 MB)
```

We are also storing the text for reccomendations:

1. **Creating Recommendations and sending them as notifications to the Smartphone**

We want to send the user useful reccomendations in this application. Therefore, we have created restaurant suggestions based on the user's preference.

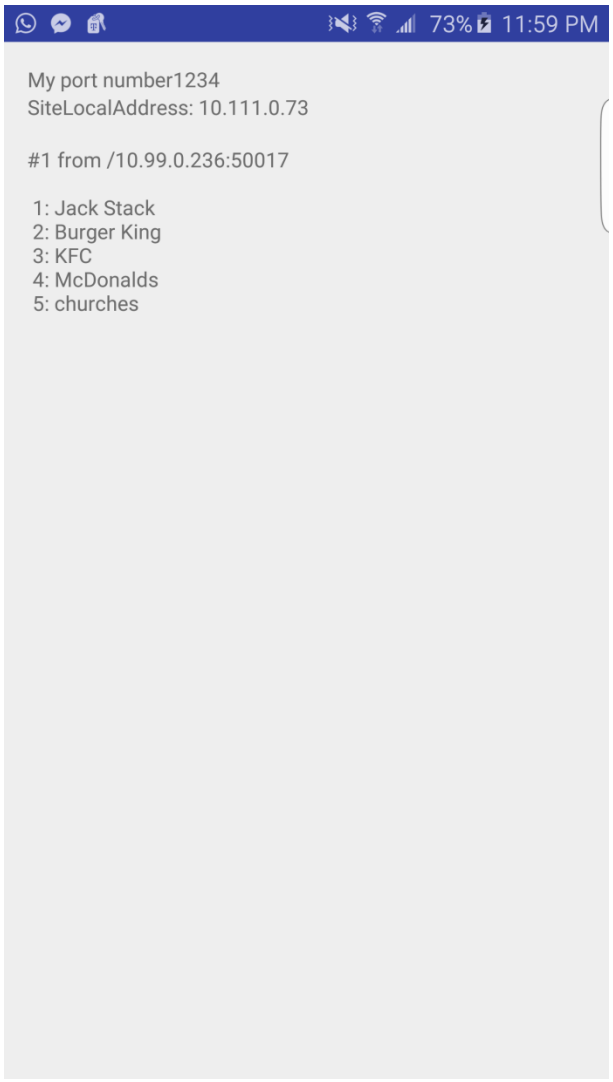
First, we break it down into training and testing data. The code below shows how we have done this.

```
val numPartitions = 4
val training = ratings.filter(x => x._1 < 6)
  .values
  .union(myRatingsRDD)
  .repartition(numPartitions)
  .cache()
val validation = ratings.filter(x => x._1 >= 6 && x._1 < 8)
  .values
  .repartition(numPartitions)
  .cache()
val test = ratings.filter(x => x._1 >= 8).values.cache()

val numTraining = training.count()
val numValidation = validation.count()
val numTest = test.count()

println("Training: " + numTraining + ", validation: " + numValidation + ", test: " + numTest)
```

Next, we show the collaborative filtering model used to train the model:

```
// train models and evaluate them on the validation set

val ranks = List(8, 12)
val lambdas = List(0.1, 10.0)
val numIters = List(10, 20)
var bestModel: Option[MatrixFactorizationModel] = None
var bestValidationRmse = Double.MaxValue
var bestRank = 0
var bestLambda = -1.0
var bestNumIter = -1
for (rank <- ranks; lambda <- lambdas; numIter <- numIters) {
  val model = ALS.train(training, rank, numIter, lambda)
  val validationRmse = computeRmse(model, validation, numValidation)
  println("RMSE (validation) = " + validationRmse + " for the model trained with rank = "
    + rank + ", lambda = " + lambda + ", and numIter = " + numIter + ".")
  if (validationRmse < bestValidationRmse) {
    bestModel = Some(model)
    bestValidationRmse = validationRmse
    bestRank = rank
    bestLambda = lambda
    bestNumIter = numIter
  }
}
```

We send the results to the smartphone:



Machine Learning Model:

1. Model predicting the User Activity as Getting up the Bed

2. Model Predicting the User Activity as Climbing the Stairs

3. Model Predicting the User Activity as Walking

3. Implementation status report:

a. Work completed

Member	Task Description	Member Responsibility	Contribution %	Time (hours)	Comments /Issues
Rakesh	1. Implementing Naïve Bayes Machine Learning Algorithm in Spark. 2. Collecting data from smart phone.	Develop, design, build and testing the task	100	22	
Nihar	1. Stream data from Smart phone to Spark. 2. Sentimental Analysis on the text received from the user	Develop, design, build and testing the task	100	20	
Mark	1. Recommendation system using Collaborative Filtering 2. Collecting restaurant data for the recommendation system.	Develop, design, build and testing the task	100	22	
Sricharan	1. Sending the recommendations to Smart phone. 2. Sentimental Analysis.	Develop, design, build and testing the task	100	20	

Work yet to be completed

Member	Task Description	Member Responsibility	Time (hours)	Comments /Issues
Rakesh	1. Implementing Machine Learning Algorithms in Spark R 2. Sending notification of predicted data. 3. Predicting the user patterns regularly. 4. Implementing API's.	Develop, design, build and testing the task	40	
Nihar	1. Implementing Machine Learning Algorithms in Spark R. 2. Collecting more sensor data sets. 3. Predicting the user patterns regularly. 4. Implementing API's.	Develop, design, build and testing the task	40	
Mark	1. Implementing Machine Learning Algorithms in Spark R 2. Sending notification of extracted data 3. Improving GUI. 4. Predicting the user patterns regularly.	Develop, design, build and testing the task	40	
Sricharan	1. Implementing Machine Learning Algorithms in Spark R 2. Sending notification of extracted data. 3. Testing the implemented features. 4. Predicting the user patterns regularly.	Develop, design, build and testing the task	40	

3. Third Increment Report:

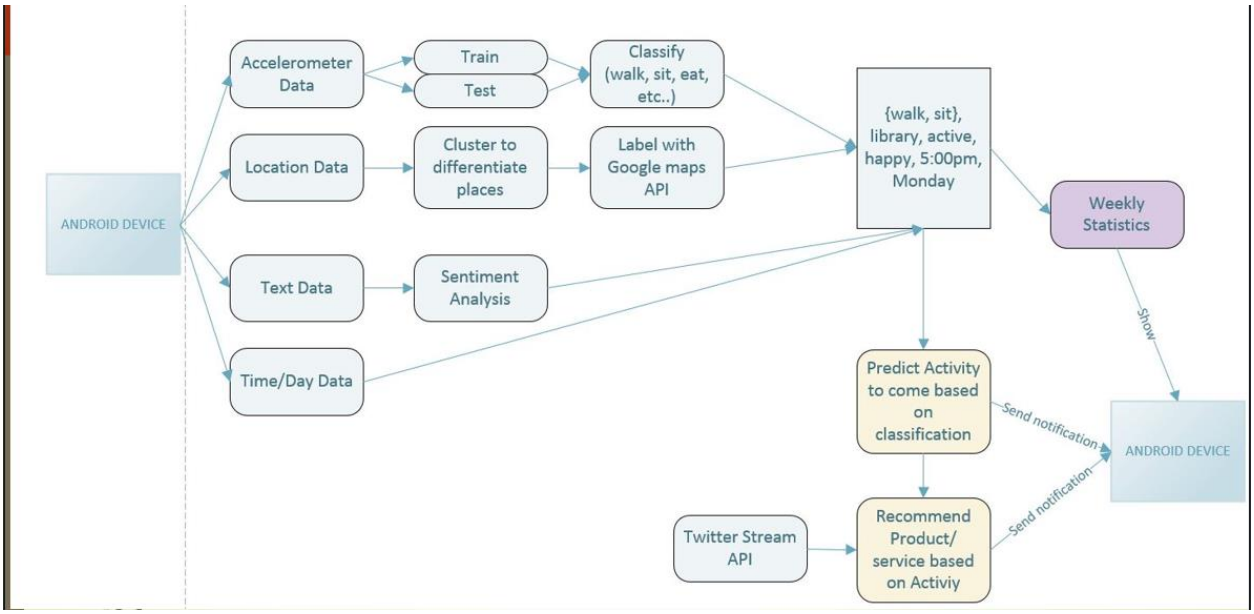
In this increment, we have implemented several pieces of our project. First, we have successfully used k-means clustering to establish frequent locations of our user, given information about latitude and longitude. We have tested this data by creating a large set of input data around a specific point in downtown Kansas City. As expected, the clusters are evenly dispersed around said point. When inserting one specific location into the data multiple times, the clustering method adjusts accordingly.

Once the cluster centers are found, the clusters then receive labels. The clusters detect frequent locations, and the labels provide additional value to the said location. We export the k centers to Google Maps API to determine the address of each center. Furthermore, we use this to determine the type of location (bookstore, library, coffee shop, etc.).

Also, we have built a socket connection which live streams data from the mobile app to the spark stream. With the spark stream, we are able to implement our naïve Bayesian model in order to classify specific movements based on accelerometer data. Such movements include walking, eating, and climbing the stairs. While these small movements may not seem like provide much insight alone, we will be using it as data to predict even larger trends in routines. The socket also provides other data, including the latitudes and longitudes for location detection, time, and day.

1. Implementation

Workflow of project:



Mobile Client Implementation (Smart Watch, Smart Phone, Robot):

Mobile Application for Collecting the Sensor Data:

- We have developed an application which is specifically for collecting the user smart phone and smart watch sensor data.
- In this application we have collected the Sensor streaming data and sending to Spark.
- Later we use this data to train over model to predict the user patterns regularly.
- This application mainly gets the all sensor data values from the Sensors such as Gyroscope, Accelerometer, Proximity, Light, Magnetic and Heart Rate in the Smart watch and smart phone.
- Apart from this we also collected the Location's latitude and longitude at which the sensor data
  Collected is sent to Spark.

Clustering the location data using K-means clustering technique:

- We have used k-means clustering technique to estimate user's frequent locations.
- Tested this clusters by creating large input data around KC-downtown.
- Once cluster centers are found they are labeled according to time zone.
- We export k centers to Google Maps API to determine the address of each center which can be used to determine the type of location.

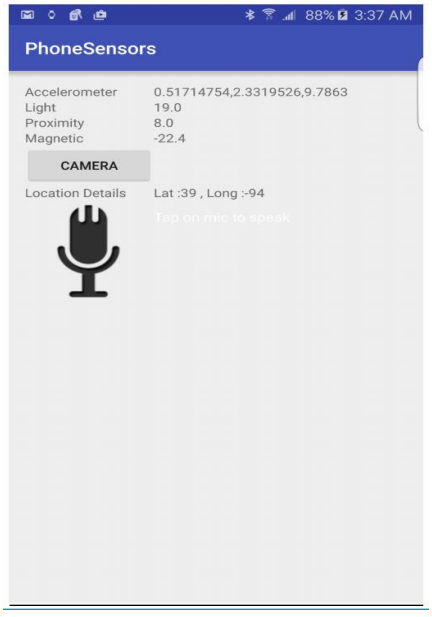
Machine Learning Application:

Classification Algorithm- Naïve Bayes Classification Algorithm to predict the User Activity:

- We have predicted the user activity using the Naïve Bayes Classification algorithm.
- We have trained the model based on the data obtained for different activities such as walking, getting out of bed and climbing up the stairs.
- So using the training data, we have predicted the user activity for the particular accelerometer readings using the Naïve Bayes Classification algorithm.
- Naïve Bayes Classification algorithm:
    4. Calculate the Mean and Standard deviation for each of column in the data set.
    5. The likelihood for each of the axis for activity is calculated.
    6. The likelihood of each of the axis reading was calculated using the Gaussian distribution function.
    7. Using the Naïve Bayes posterior probability formula, we have computed the probability of occurrence of posterior.
    8. Posterior probability is directly proportional to product of prior probability and likelihood.
- Using the posterior probability, we have predicted which activity is most likely to perform.
- The Machine learning algorithm runs using Spark

2. Deployment:

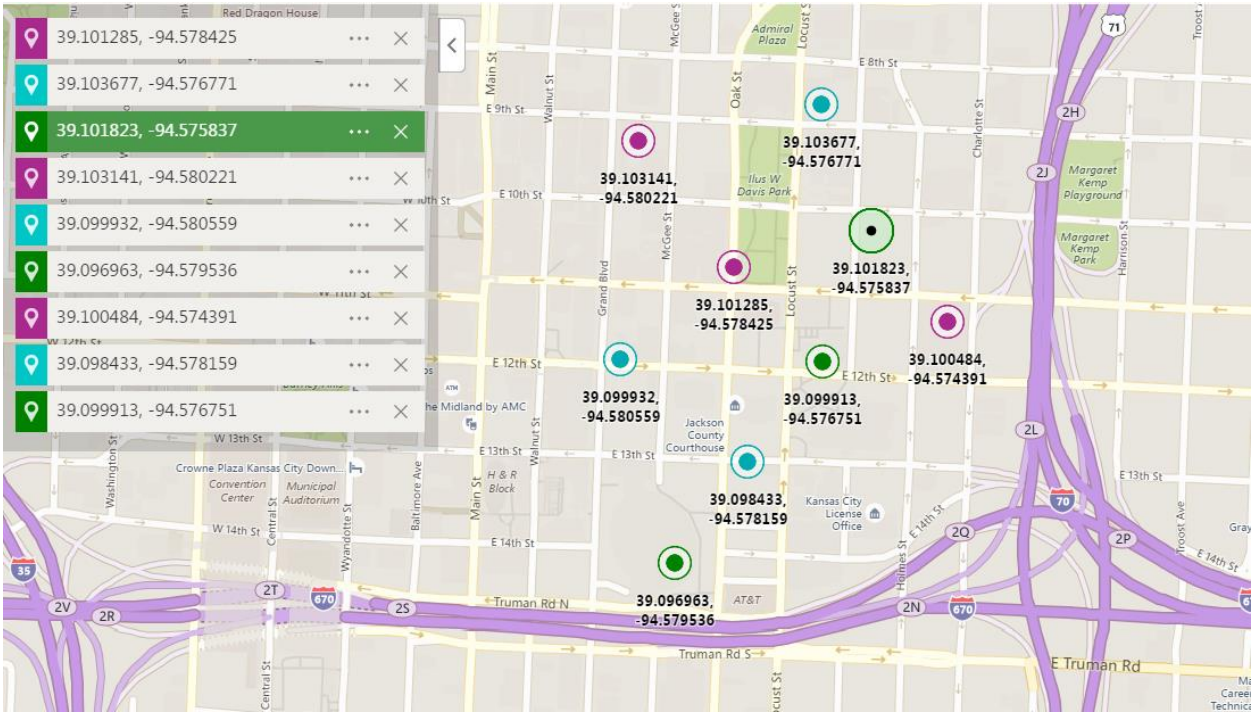
1. SmartPhone/SmartWatch Sensors Data Storage Application:



The various sensors in both mobile and smart watch gives the data values continuously for the user activities.

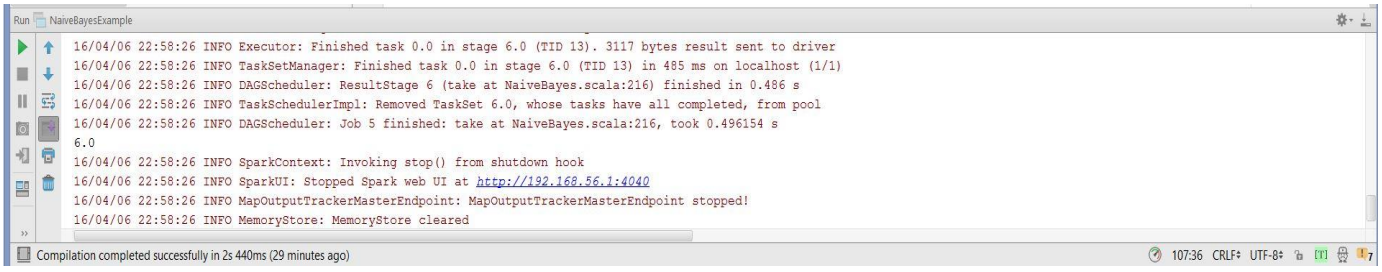
We store these values in the internal storage of the Device.

2. Clustering the location data using k-means:



Clusters created near KC-Downtown.

3. Displaying current activity of user activity in Spark where the data is taken from the Device

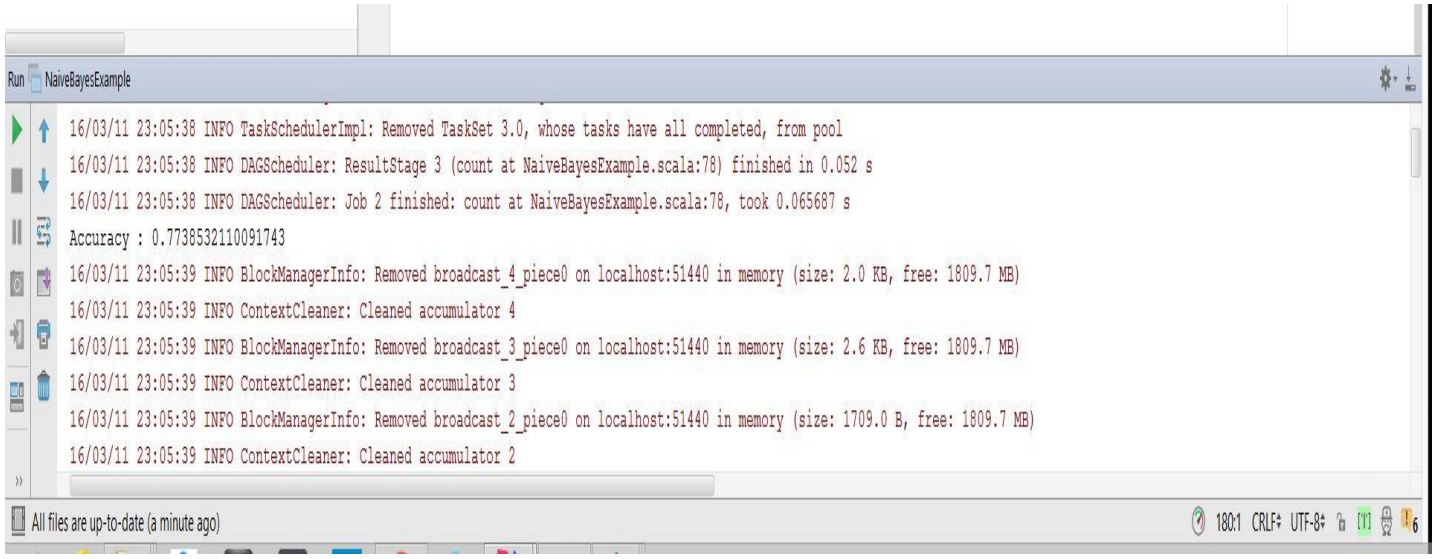


Supervised Learning Algorithm Machine Learning Model:

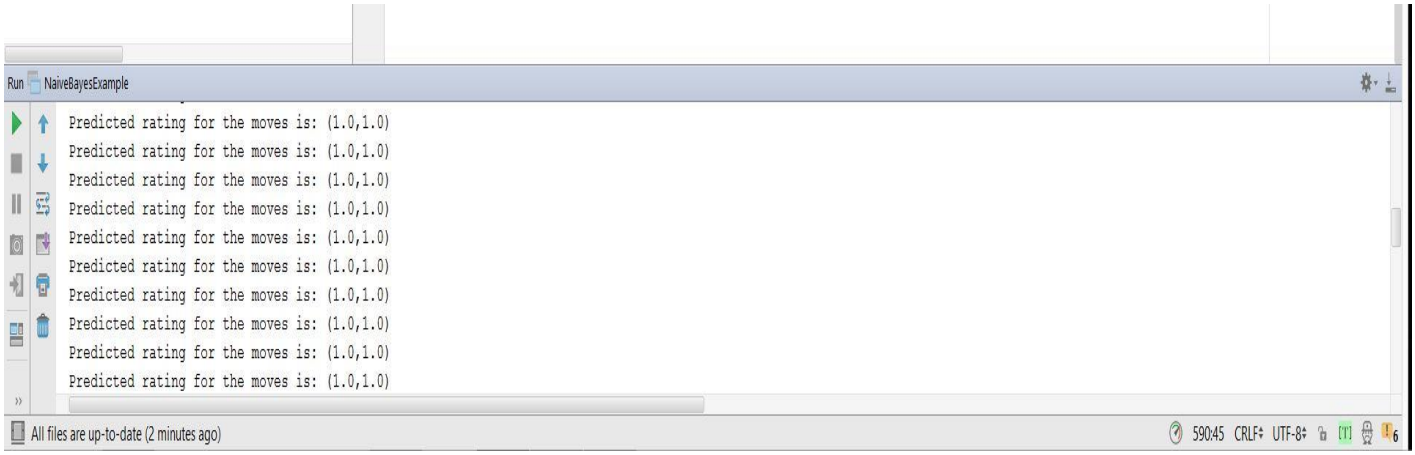
Classification of User movements and activities using the Naïve Bayes Classification:

- We have divided the data into two parts testing data and training data.
- Training data—60%
- Testing data ---40%
- We have trained the model to predict the different user activities and movements such as sitting up the chair, sitting down the chair, walking, sleeping etc.,
- We used the Multinomial distribution and we have **normalized** the input data values to fall between 0-1 to increase the accuracy.
- Normalized is done using MLLib function.
- After Normalizing the features, We have converted the RDD into **labelled Vector RDD**
- Accuracy is calculated by validating the results predicted from the testing data to the actual values. We have got accuracy of 77.38%
- We can use this model to classify the incoming data from the user.
- We have saved this model to predict the later data from the user. From this data we provide the overall stats of user time spending on activities.

1. Model Accuracy – nearly 77%:



2. Displaying Confusion Matrix for the User Activity :



3. Implementation status report:

a. Work completed

Member	Task Description	Member Responsibility	Contribution %	Time (hours)	Comments /Issues
Rakesh	1. Sending streaming data from phone to Spark. 2. Prediction of specific activity based on streaming data. 3. Labelling the clustered location data.	Develop, design, build and testing the task	100	22	
Nihar	1. Clustering location data using k-means. 2. Downtown location data collection.	Develop, design, build and testing the task	100	20	
Mark	1. Clustering location data using k-means. 2. Collecting location data set for clustering.	Develop, design, build and testing the task	100	22	
Sricharan	1. Socket connection between phone and spark. 2. Labelling the clustered data.	Develop, design, build and testing the task	100	20	

b. Work yet to be completed

Member	Task Description	Member Responsibility	Time (hours)	Comments /Issues
Rakesh	1. Implementing Machine Learning Algorithms in Spark R 2. Sending notification of predicted data. 3. Predicting the user patterns regularly. 4. Implementing API's.	Develop, design, build and testing the task	40	
Nihar	1. Implementing Machine Learning Algorithms in Spark R. 2. Collecting more sensor data sets. 3. Predicting the user patterns regularly. 4. Implementing API's.	Develop, design, build and testing the task	40	
Mark	1. Implementing Machine Learning Algorithms in Spark R 2. Sending notification of extracted data 3. Improving GUI. 4. Predicting the user patterns regularly.	Develop, design, build and testing the task	40	
Sricharan	1. Implementing Machine Learning Algorithms in Spark R 2. Sending notification of extracted data. 3. Testing the implemented features. 4. Predicting the user patterns regularly.	Develop, design, build and testing the task	40	

4. *Fourth Increment Report:*

The main features of our project involve the following: linking activities to locations, predicting current and future activity based on probabilistic trends, and making recommendations based upon said activities and locations.

To begin, we set the common locations by k-means clustering. The reason we cluster locations is so that we can encompass movement within the cluster, and we can ignore noise in between the clusters (driving/walking between the locations). Once we have these clusters set, we can map latitudes and longitudes to a specific location. For our demonstration, we have selected four locations.

Next, we use accelerometer data to model certain activities. We extract features from the accelerometer data and use them to classify the data. We then perform naïve Bayes on the extracted data. With this, we are able to determine the approximate.

We have our locations, and we have our activities, now we want to link the two together. For example, say we are studying in the library. Assuming the library is cluster 1, our application would append '1::studying' to a text file containing this data. Because the user enjoys studying, we may get the result '1::studying' multiple times in a row. The application checks for this and only adds a new row if it is unique from the row above.

We also want to link changes in location clusters to predict the next state. Say we have '1::studying' followed by '3::eating'. We want to create another file that keeps track of this change between locations. This file has the format '1::3'.

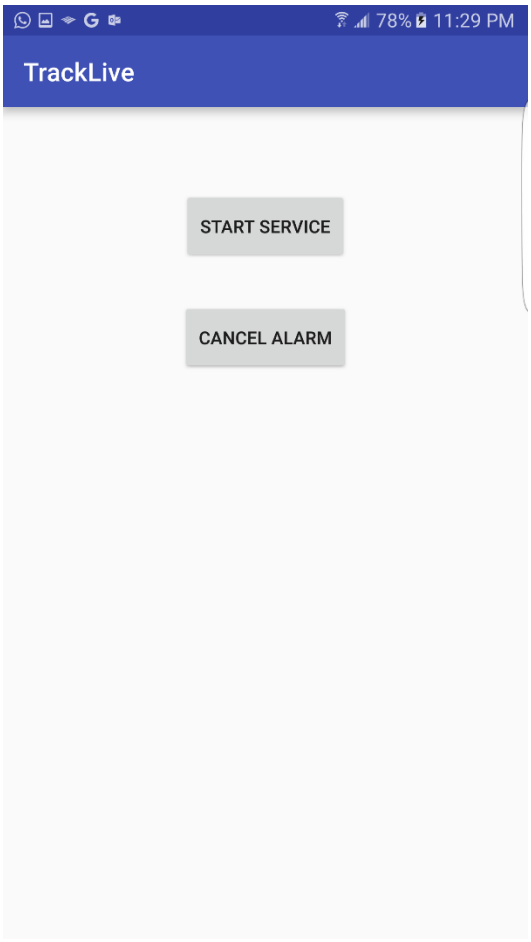
With the given data, the application will recommend and activity using collaborative filtering, and predict the next location using the maximum probability of the next location.

1. Implementation

1.Android side implementation

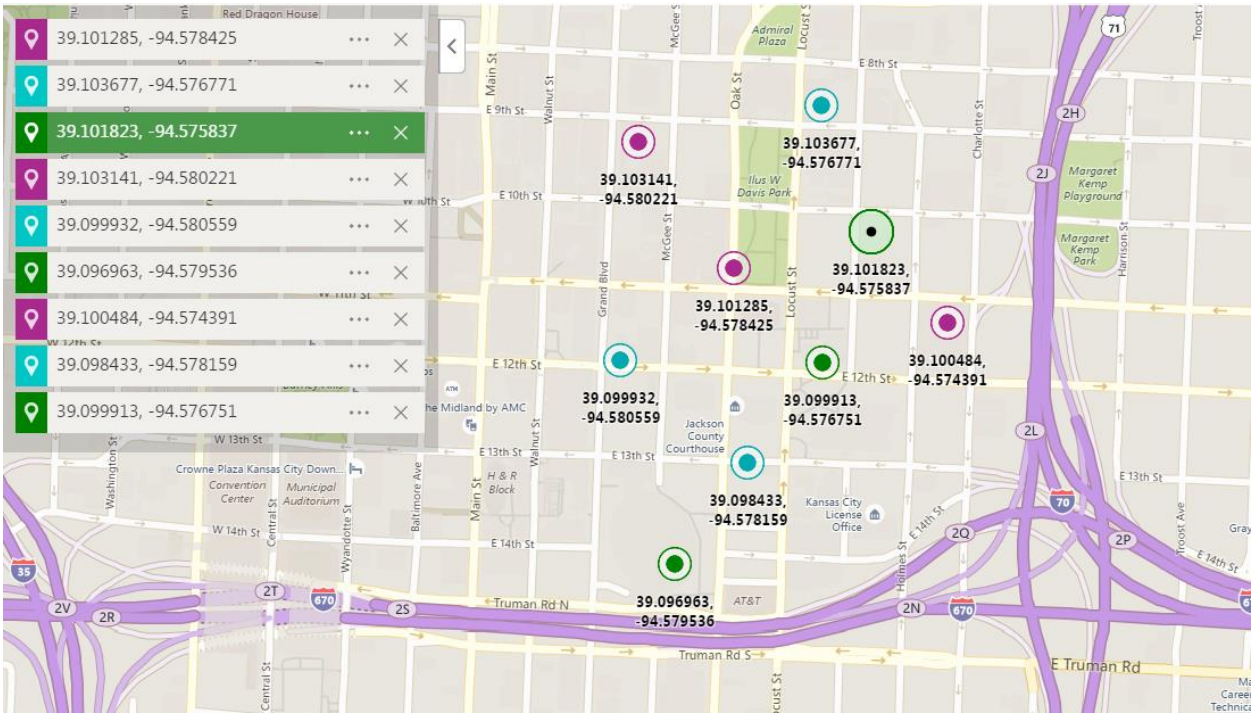
The android application is started after clicking on the logo. It basically provides the recommendation of activities to the user based on his history and statistics of activity done at specific time periods.

2. Menu screen obtained on the application



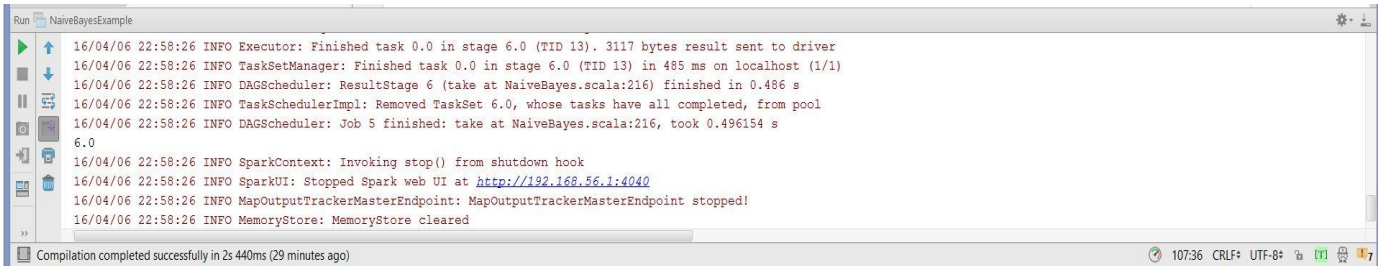
The next screen is followed by two basic buttons, where one is to start the service. The other button used is to cancel alarm.
The service is triggered every half an hour. The current activity and future activity within half an hour is received by the application. The recommendations are received based on the future activities.

3. Clustering the location data using k-means:

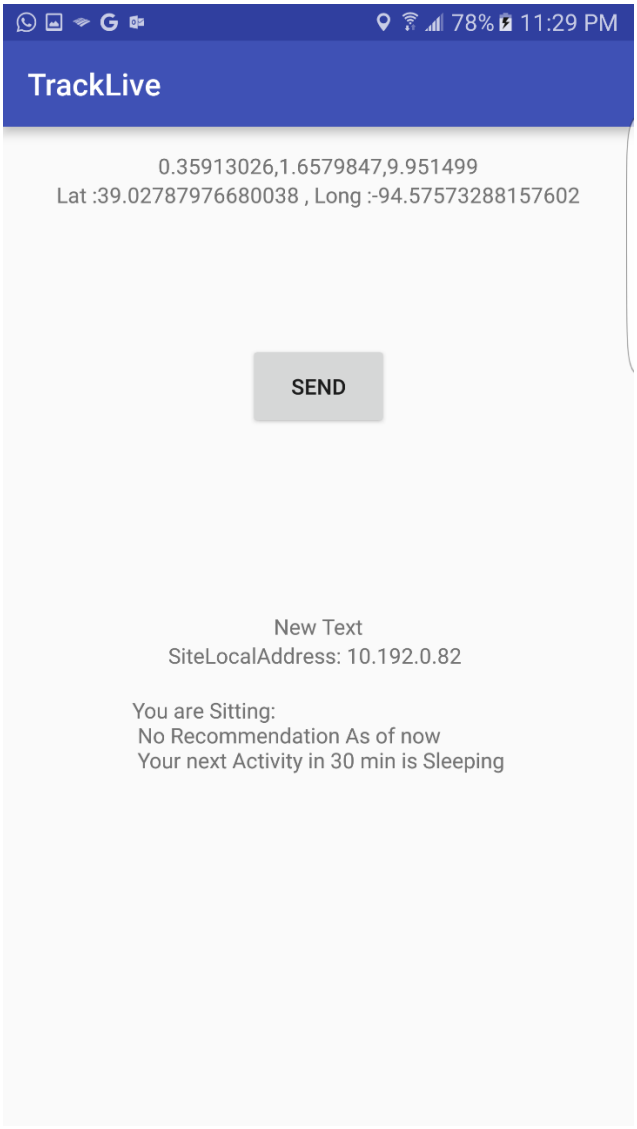


Clusters created near KC-Downtown.

4. Displaying current activity of user activity in Spark where the data is taken from the Device



5. Final output screen shown on the android application.



2. Deployment

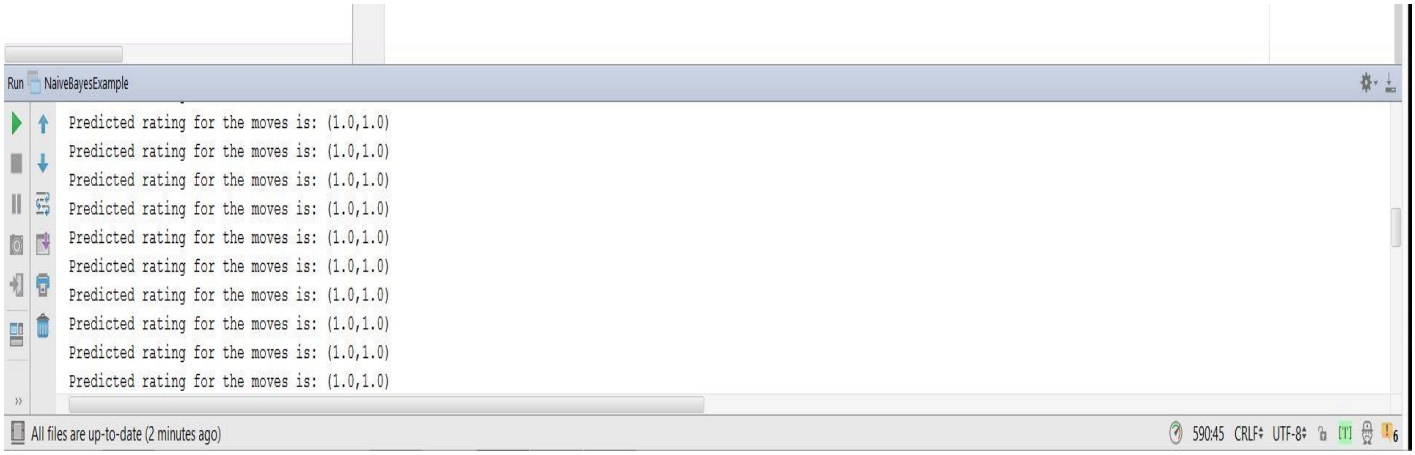
The final implementation screen is obtained from Spark where the server resides. The socket is opened to listen continuously. The main functions occurring here include knowing the current activity from the user accelerometer data. The classification occurs based on the saved model. The probability of next event occurring is calculated and selected. The next event is recommended and all data is pushed back to the client. The respective recommendation is returned and shown as above.

Supervised Learning Algorithm Machine Learning Model:

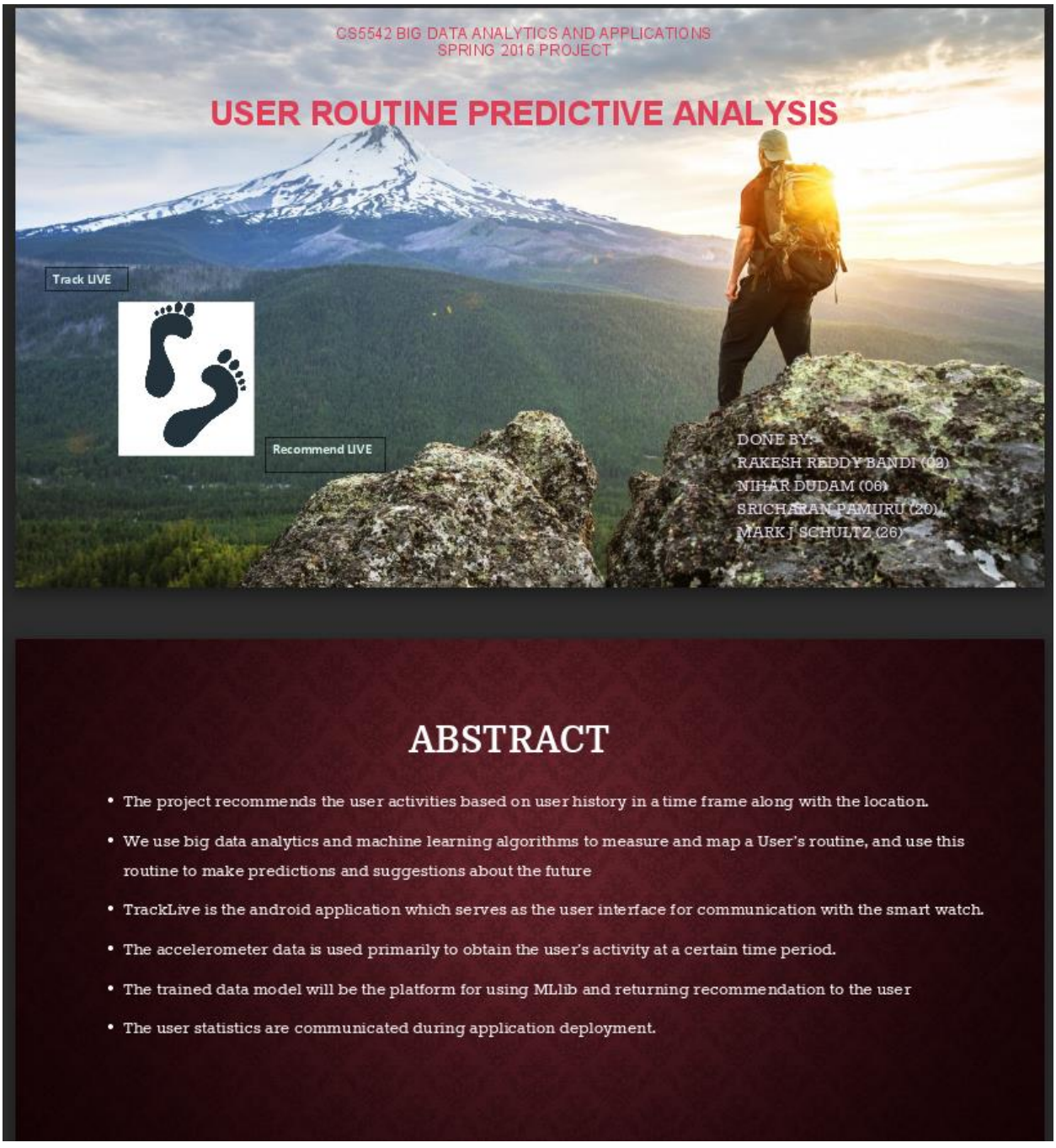
Classification of User movements and activities using the Naïve Bayes Classification:

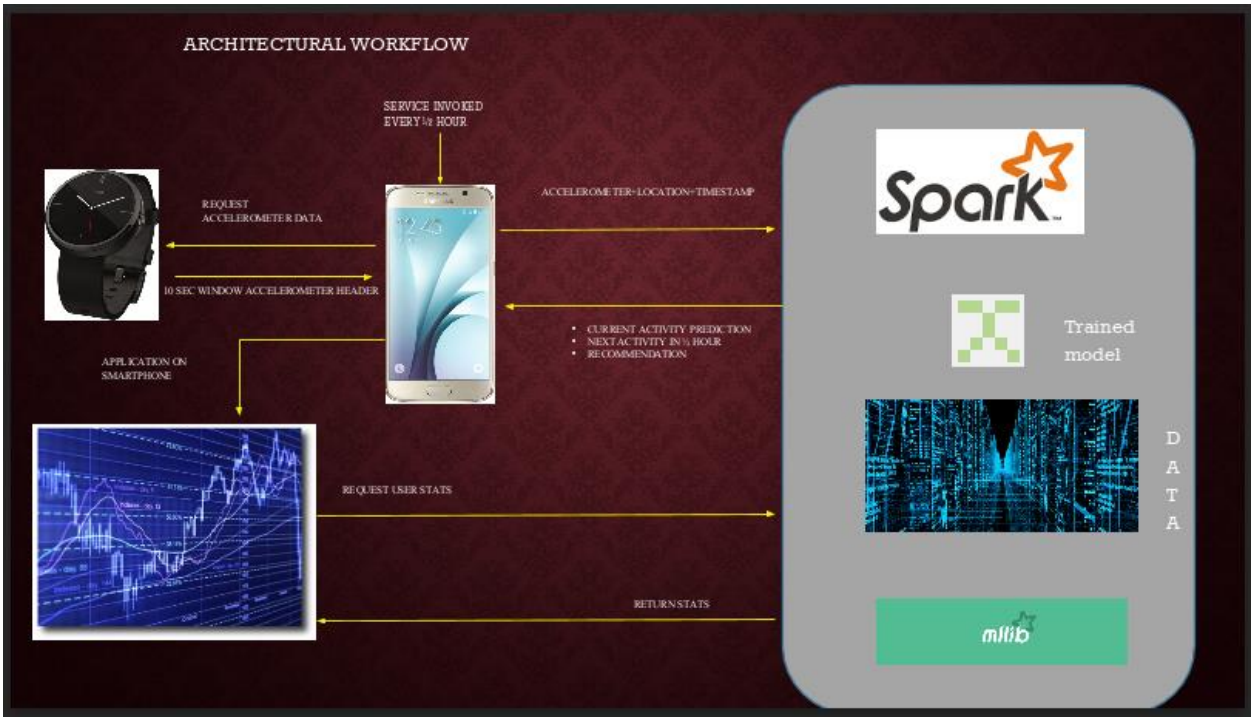
- We have divided the data into two parts testing data and training data.
- Training data—60%
- Testing data ---40%
- We have trained the model to predict the different user activities and movements such as sitting up the chair, sitting down the chair, walking, sleeping etc.,
- We used the Multinomial distribution and we have **normalized** the input data values to fall between 0-1 to increase the accuracy.
- Normalized is done using MLLib function.
- After Normalizing the features, We have converted the RDD into **labelled Vector RDD**
- Accuracy is calculated by validating the results predicted from the testing data to the actual values. We have got accuracy of 46.38%
- We can use this model to classify the incoming data from the user.
- We have saved this model to predict the later data from the user. From this data we provide the overall stats of user time spending on activities.

1. Displaying Confusion Matrix for the User Activity :



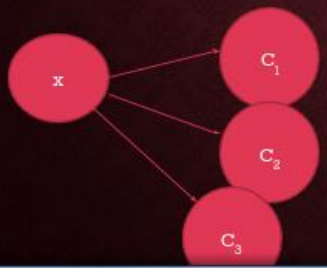
5. Presentation slides:

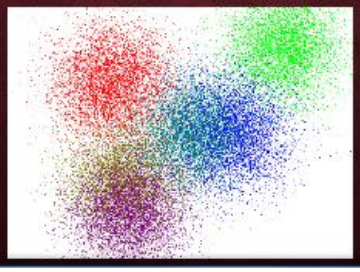


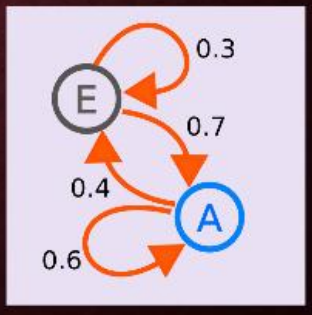


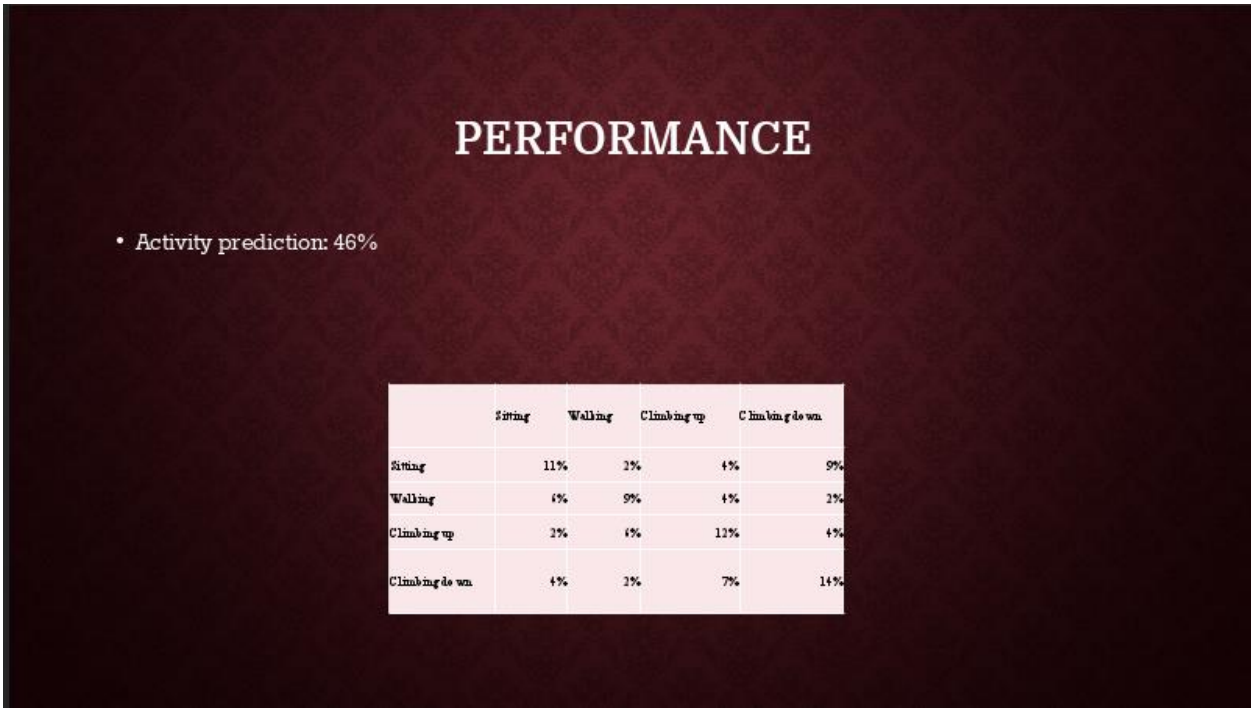
ALGORITHMS USED

- Naïve Bayes with Feature extraction for the accelerometer data
- Location Clustering (with k-means clustering)
- Markov Chain for state dependent location prediction









6. Github URL:

https://github.com/SCE-UMKC/BigData-Spring-2016-User-Routine-Predictive-Analysis

7. Youtube URL:

https://youtu.be/8PvDgS6vCkQ

8. Project Summary

a. Introduction:

Big data has become a buzzword in the field of Computer Science. As technology advances, we are finding new ways to collect and analyze data. With the invention of the smart watch, we have been able to expand this collection of data to areas that had not been possible years ago. In addition, it has never been easier for the user to retrieve data from their device when needed. Our project combines these two conveniences into one application. We will use big data to measure and map a User's routine, and use this routine to make predictions and suggestions about the future.

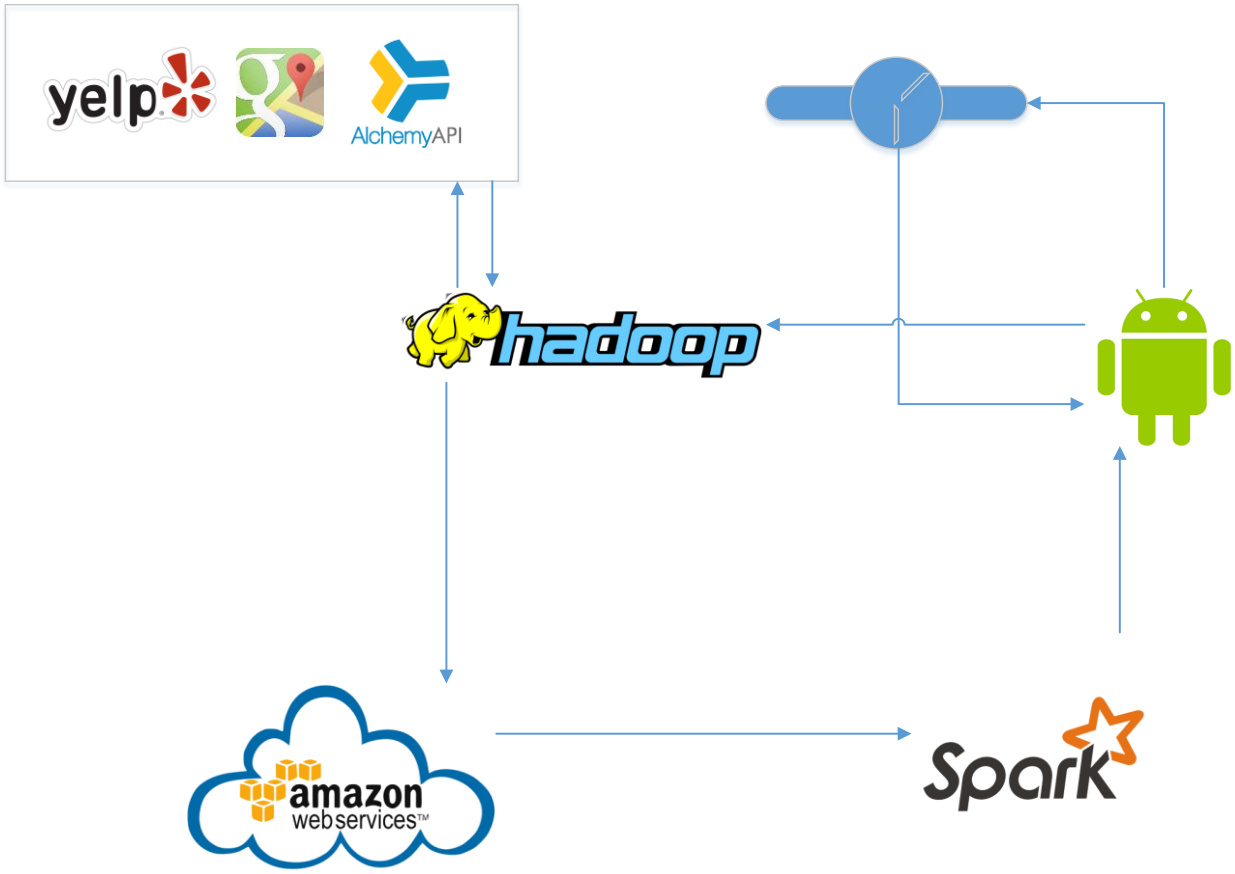
We plan on applying several sources for data collection. First of all, the applications collect data from movements of the watch to classify patterns in certain movements. It will use these movements to guess the activity that the User may be doing at the moment. The applications that we have designed is TrackLive which tracks a user's live activity and provides the user with an interpretation of what the next activity is to be done. Basically, the machine is trained with a set of activities in a corresponding time frame to notice what the activity will be at that time.

Based on machine learning algorithms, we are able to train the data and the accelerometer in the smart watch is the sensor that checks the location coordinates of the user. It is mapped with the time period and the activity that takes place is shown to the user through the application. Other key ideas which can be further implemented include user's preference of restaurants, libraries and other places which is in the vicinity. The user interface will provide recommendations on what kind of park or restaurant he would like to go.

The efficiency can be improved in analyzing more data sets and producing significant patterns among them. Our application is pretty much based on a single user's pattern and in order to expand the reach of the users we may have to inculcate features where the initial grace period serves as the training period. Overall, the application is effective and efficient in recommending the activity to the user.

b. Design of Features:

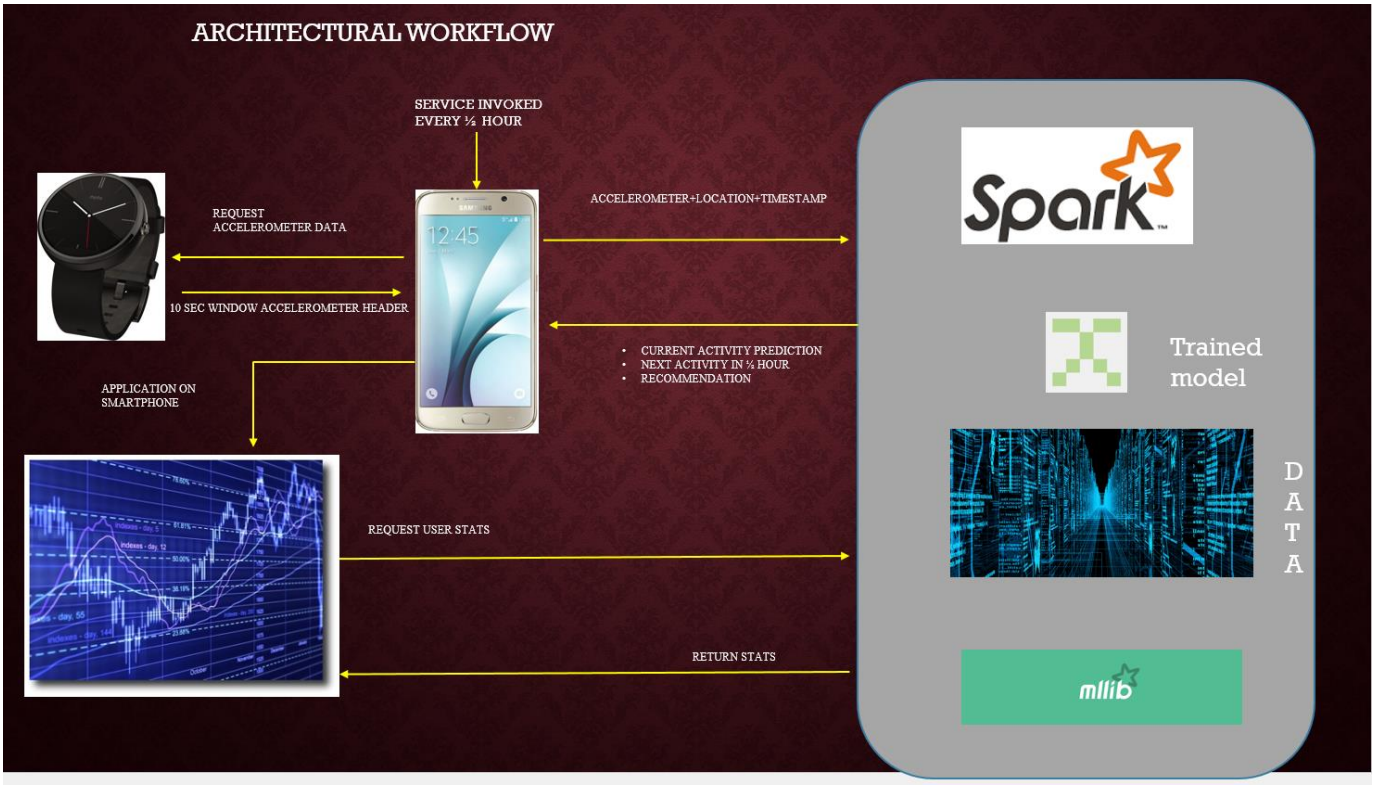
i. Architecture:



Our application begins and ends with the smartwatch user interface. The user will input to the watch. watch sends data to the Hadoop File System through the Android application. The Hadoop System will organize data that comes both from the user and from APIs. The application will then send data to the Amazon Web Services to run analytics on the data. We will implement MapReduce programs in Spark and Spark R. We will use several machine learning algorithms for prediction. After running the Query in spark, the data is sent back to the Android application, which can then be viewed by the user on their smartwatch.

c. Application

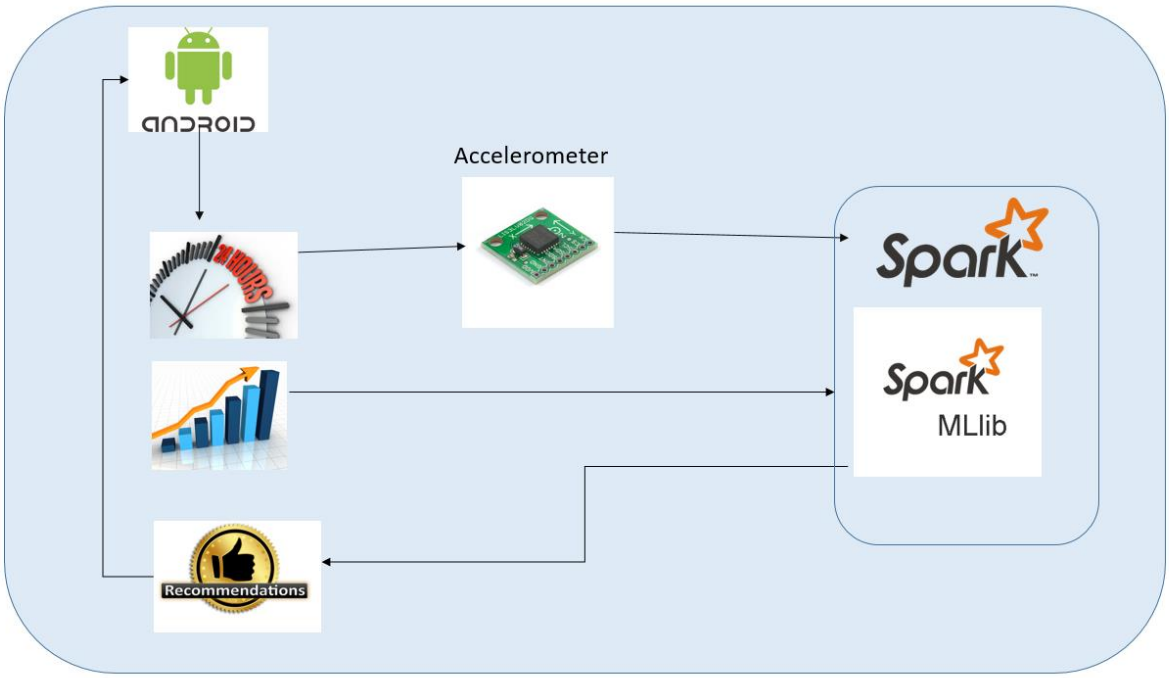
i. Workflow of project:



Mobile Application for Collecting the Sensor Data:

- We have developed an application which is specifically for collecting the user smart phone and smart watch sensor data.
- In this application we have collected the Sensor streaming data and sending to Spark.
- Later we use this data to train over model to predict the user patterns regularly.
- This application mainly gets the all sensor data values from the Sensors such as Gyroscope, Accelerometer, Proximity, Light, Magnetic and Heart Rate in the Smart watch and smart phone.
- Apart from this we also collected the Location's latitude and longitude at which the sensor data
  Collected is sent to Spark.

ii. Functional Outline:



The Android application will check the specific time period and determines location and other data based on sensors. The accelerometer data will help in understanding the user activity currently going on. The Spark system consists of the machine learning libraries where collaborative filtering, classification algorithms are run. Based on the statistical data present, the recommendation is returned to the user.

Android Application will trigger every half hour events to receive the current activity. The recommendation is basically returned based on the future activity. The application has features such as:

- Show statistics
- Most visited places
- Time spent on activities
- Do no track me

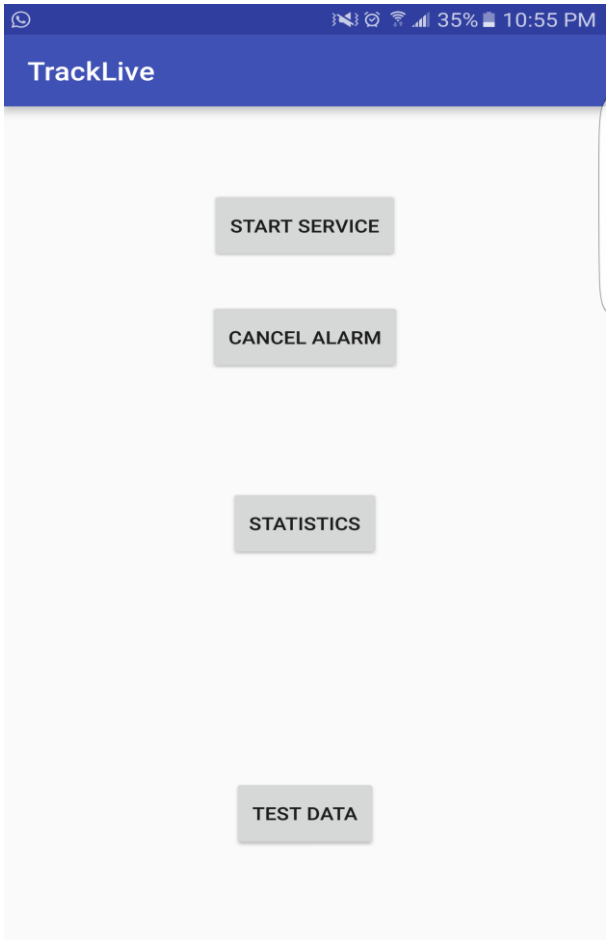
When the socket is opened it continuously runs and the request is obtained to return results to the client. The main functions here are:

- Knowing the user's current activity based on accelerometer data
- Calculate probability of next event occurrence
- Selection of events.
- Recommend events such as eating, movies, library visit.
- Data pushed back to client

iii. Android side implementation

The android application is started after clicking on the logo. It basically provides the recommendation of activities to the user based on his history and statistics of activity done at specific time periods.

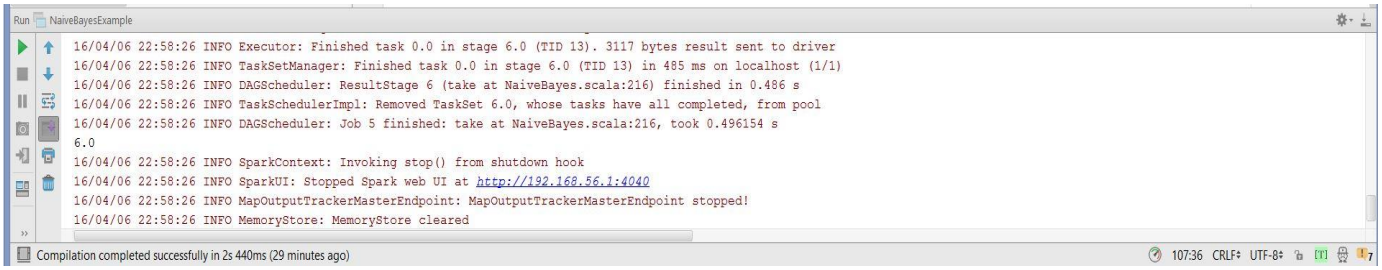
Menu screen obtained on the application



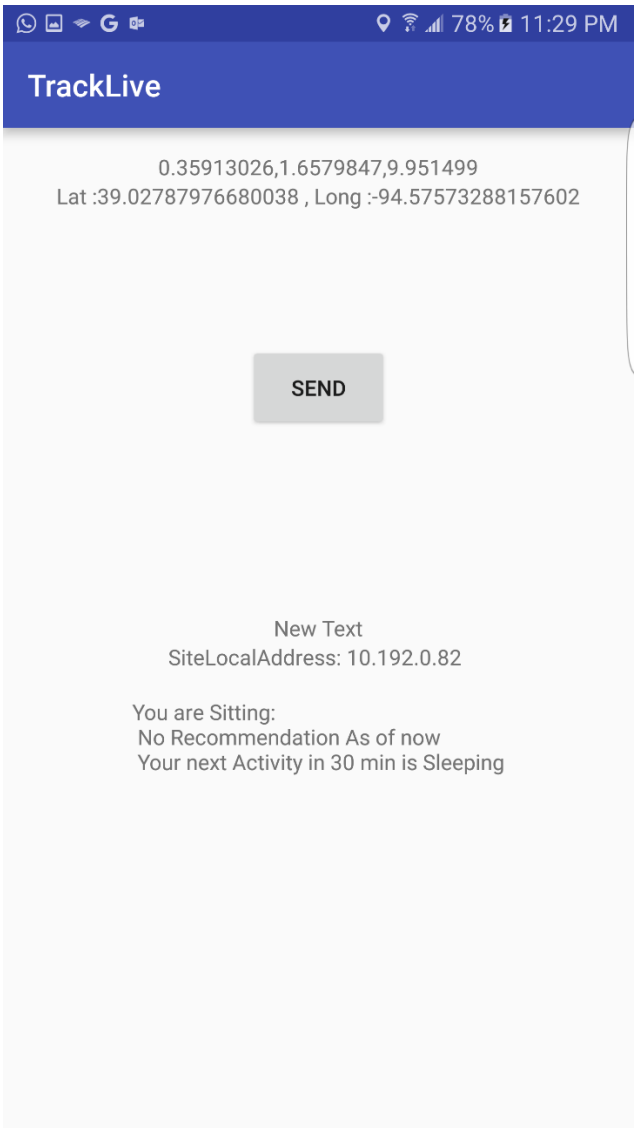
The next screen is followed by two basic buttons, where one is to start the service. The other button used is to cancel alarm.

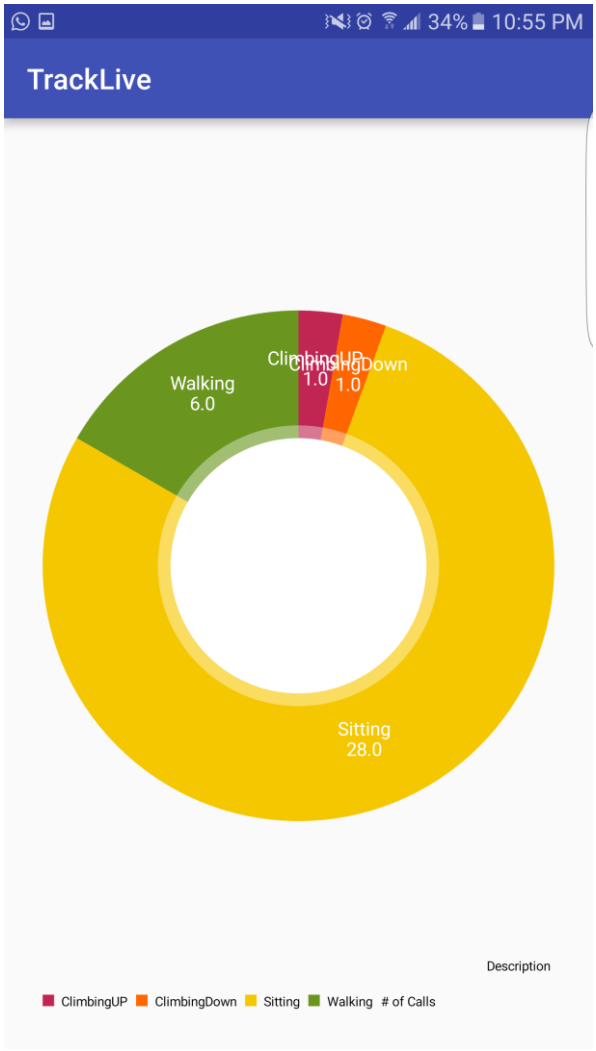
The service is triggered every half an hour. The current activity and future activity within half an hour is received by the application. The recommendations are received based on the future activities.

Displaying current activity of user activity in Spark where the data is taken from the Device



Final output screen shown on the android application





The final implementation screen is obtained from Spark where the server resides. The socket is opened to listen continuously. The main functions occurring here include knowing the current activity from the user accelerometer data. The classification occurs based on the saved model. The probability of next event occurring is calculated and selected. The next event is recommended and all data is pushed back to the client. The respective recommendation is returned and shown as above.

d. Machine Learning Application:

Machine Learning Classification Algorithm- Naïve Bayes Classification Algorithm to predict the User Activity:

- We have predicted the user activity using the Naïve Bayes Classification algorithm.
- We have trained the model based on the data obtained for different activities such as walking, getting out of bed and climbing up the stairs.
- So using the training data, we have predicted the user activity for the particular accelerometer readings using the Naïve Bayes Classification algorithm.
- Naïve Bayes Classification algorithm:
- Calculate the Mean and Standard deviation for each of column in the data set.
- The likelihood for each of the axis for activity is calculated.
- The likelihood of each of the axis reading was calculated using the Gaussian distribution function.
- Using the Naïve Bayes posterior probability formula we have computed the probability of occurrence of posterior.
- Posterior probability is directly proportional to product of prior probability and likelihood.
- Using the posterior probability we have predicted which activity is most likely to perform.

K-means Clustering to Define Discrete Locations:

The mobile devices send us latitude and longitude data based on the User's location. While this is useful information, there are slight variations between locations. For example, if a user is on one side of the library, and walks to the other side of the library, this can return a slightly different latitude. With k-means clustering, we set the number of locations being clustered as the value of k. We then randomly select cluster centers based on the data. After that, we take the mean of each cluster, and relocate the center to the mean. We continue this process for a set number of iterations, until the cluster centers align with the location.

Markov Chain for Location Prediction:

It is very likely that one event will follow another in a person's daily routine. For example, after spending several hours at the library, let us say that the user almost always go directly home. There is a chance that the user may stop for food after a particularly wearing study session. However, such an instance is the exception not the rule. Mapping probability of the next state of the user based on the current state of the user is the main concept behind the Markov chain. First, we use mapreduce to build a data file with the first location cluster followed by the second location cluster. We reduce any state where the user stays in the same location. After doing this, we calculate the probability of each state change, based on the current state. We only needed this process for the next state, but this process can be carried out

to predict several states into the future. While beginning this process, we did not fully implement this part of the solution in the final project.

e. Datasets:

A dataset is a collection of related sets of information iterated across an activity centric output. The dataset collected as input from the smartwatch will be fed via machine learning algorithms to the remote machine for it to be analyzed. The data/information is obtained from the different sensors which are required for providing an analytic comparison. Output is retrieved from the values fed. Supervised learning is in the initial stages since the machine is trained or rather intelligence provided regarding what to react in which case. After the training data set is provided, unsupervised learning comes to the frame. Here, the machine is able to analyze the data and compare with the training set. It will also be able to learn in case the user does an activity different from the routine. A notification is sent but nonetheless learning is also done by the machine, which in our case is the smart phone connected via USB.

The algorithm takes location data (latitude and longitude) and accelerometer data. The location data does not send on the same row as the accelerometer data, so it is important that we reformat the data in order to get it all on one line. We also do the same thing with time and date data. The result is a clean csv of data. In addition, we can also label the data when we are familiar with the location. Below is a small example taken from the Miller Nichols Library.

X	Y	Z	Day	Date	Time	Lat	Long	Place
12.23677	-10.6207	0.289698	Monday	4/4/2016	9:00am	39.03508	-94.5765	Miller Nicholas Library
3.57574	-5.83108	-0.21907	Monday	4/4/2016	9:00am	39.03508	-94.5764	Miller Nicholas Library
3.854665	-10.7667	-2.36068	Monday	4/4/2016	9:00am	39.03508	-94.5765	Miller Nicholas Library
4.021062	-11.3617	1.297657	Monday	4/4/2016	9:00am	39.03511	-94.5765	Miller Nicholas Library
13.15015	-3.41293	-1.04148	Monday	4/4/2016	9:00am	39.03513	-94.5764	Miller Nicholas Library

Source: The accelerometer data is collected from smart watch and Smart phone for various activities like sitting, walking, etc. which serves as training data for the model.

Format of the data: The accelerometer data collected has several columns in the order of X(x-co-ordinate), Y(y-co-ordinate), Z(z-co-ordinate),Day(day of the week), Date(format of mm-dd-yyyy), Time, Lat (latitude), Long(Longitude), Plac

f. Evaluation:

Once we have the data processed, we want to determine what activities are taking place and where they are taking place. In order to do this, we will take a segment of this data, perform feature

extraction, and then do naïve Bayes on the accelerometer data. This gives us something tangible rather than an x, y and z axis. We also perform clustering on the location data. We do this to convert slight variations in the latitude and longitude (variation within a particular location is somewhere around 10E-5 degrees) into discrete locations. If we know the label, we can add it to the equation.

g. Accuracy:

The accuracy of the project is measured as Activity prediction which is around 46%. We believe that the accuracy could be improved by implementing more feature extractions. We also believe that a random forest tree may potentially provide a more accurate solution than a naïve Bayes classifier. The clustering prediction gave us the correct answer when given a discrete set of locations, however in order to expand our solution to a real world application, we would need to implement a solution to dynamically add new clusters.

h. Performance:

Performance can be illustrated by the following confusion matrix:

Classifier results		Truth data					
		Sitting	Walking	Climbing UP	Climbing DOWN	Classification overall	Producer Accuracy (Precision)
	Class 1	11	2	4	9	26	42.308%
	Class 2	10	9	4	2	25	36%
	Class 3	2	6	12	4	24	50%
	Class 4	4	2	7	14	27	51.852%
	Truth overall	27	19	27	29	102	
	User Accuracy (Recall)	40.741%	47.368%	44.444%	48.276%		
Overall accuracy (OA):	45.098%						
Kappa[1]:	0.267						

i. Related Work(References)

- Komninos, Andreas, and Mark Dunlop. "Text input on a smart watch." Pervasive Computing, IEEE 13.4 (2014): 50-58.
- Lutze, Rainer, and Klemens Waldhor. "A Smartwatch Software Architecture for Health Hazard Handling for Elderly People." Healthcare Informatics (ICHI), 2015 International Conference on. IEEE, 2015.
- http://www.motorola.com/us/products/moto-360
- https://www.thegrommet.com/robome-app-enabled-robot
- https://www.yelp.com/developers/documentation/v2/overview
- https://api.eventful.com/

j. Future Work

The project can be further improvised if we can obtain the recommendations of the user as per his location and suggest restaurants, libraries, parks, recreation places. It can be implemented in this perspective and the user interface can be enhanced so that the simulation is easy and effective for all users on a regular basis. It is a one-user application and his personal training can be done by including options to train the app. This would improve the significance and entirety of the cause.

9. Project Management

a. Project Management Report:

Everyone in the team worked together to build this application.

Member	Task Description	Member Responsibility	Points allocated
Bandi Rakesh Reddy (2)	1. Activity Recognition using NaiveBayes 2. Extracted features from accelerometer data from SmartPhone 3. Android Application for SmartPhone (Which triggers every 30min) 4. Android Background Service for SmartWatch which listens for SmartPhone Request.	Develop, design, build and testing the task	25
Dudam Nihar (6)	1. Collecting more sensor data sets. 2. Finding patterns from mapped Locations 3. Sentimental Analysis for text data	Develop, design, build and testing the task	25
Mark J Schultz (20)	1. Clustering and labeling the Locations 2. Mapping every Location 3. Implementing Recommendation System	Develop, design, build and testing the task	25
Pamuru Sricharan (26)	1. Collecting more sensor data sets 2. Implementing Recommendation System for Restaurants	Develop, design, build and testing the task	25

b. Final Project Evaluation:

Our final product did not meet all of the requirements we had hoped to accomplish. Given the learning curve of the tools and techniques we used in class, the time constraint was much tighter than we had anticipated. Ideally, we could continue our work to improve the scope and potential range of users. The design process was satisfactory and a majority of our goals were achieved. The agile process helped us assess ourselves and provide each team member with an opportunity to work and integrate at each increment. The project plan schedule was somewhat deviating, like we spent a lot of time on certain components. The extraction of features was a topic which consumed a lot of time.

The management structure within the group was divided in a way that Android Application and user side design was managed correspondingly with data collection and integration. Machine learning algorithms were implemented simultaneously to broaden the research. It was a new topic and we had to spend significant time to understand the concepts and write code. In case the project was to be implemented in the real world, it would be necessary for us to make the user interface more interactive. We should also be able to have a service running where the training is done in the initial time period. It would be better if the tutorials were more detailed and a bit slow so that we have more time to understand each concept and the projects are done successfully within the stipulated time. Overall, it was a wonderful experience for all team members and equal contribution has helped us to come up with this application.