



---

# 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)

## **I. 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. It will also collect data from the heart rate monitor of the smartwatch to measure the rigor of the User's activity. GPS data on the watch will determine the location of the user. We will also use datetime metrics to determine the day and the time that the user typically does certain activities.

The sources of data above will be collected by the user, but we also want to combine this with several API's to make better predictions, and add suggestions to give back to the user. We will use speech to text API to query the application for information. We will use the Google Location API to develop a richer context of the user's most frequented locations, and suggest possible locations that might interest the user. We will implement the Yelp API, so that when the application predicts when the user eats, it will make suggestions on places to go.

All of the data we will collect will be organized using the Hadoop file system and implemented using Amazon Web Services. We will discuss the technologies more in this documentation. The project is broken down into four phases. For the first Increment, we began the data collection to measure several common movements. We are initially using Naïve Bayes to classify these movements. We have also began research on several of the API's that we plan on implementing.

## **II. Project Goal and Objectives:**

The main implementation of the project will consist of the following specification based objective and subjective analysis along with a sketched perspective of goal:

### **1. Overall Goal:**

The goal of the project is to predict a user's activity at any time during a user's schedule. A large dataset is provided by the smartwatch and the machine is trained on the basis of machine learning algorithms as to how the coordinates will affect the motion of the sensor. Machine is trained and the provided data is analyzed based on the reading from the various sensors of the smartwatch. The location information will provide a set of variable features such as restaurants, parks, etc. near to the naive user. APIs will provide the analysis of information and data regarding the review and further circumspect options are brought forward to help the smartwatch analyze the data and provide a predictive routine. A variable set of sensors will collect the proximity, heart rate, accelerometer based motion readings, etc. to analyze data set. The analysis of this large datasets will return recommended notifications to the user for a viable routine.

### **2. Specific objectives:**

- Predict user routine by analysis of large datasets through machine learning techniques.
- Return geolocation based places in proximity such as restaurants, parks, etc.
- Sensor dataset provides specific activity based readings.
- Provide recommended services by sending notification to user depending on time and routine.

### **3. Specific features:**

Sensors in the watch serve as primary source for providing data which is required for analysis. The accelerometer will provide a three dimensional reading with regard to the position. The heart rate sensor will provide the heart beat rate which can be used to obtain the magnitude of force exerted while performing a particular activity. GPS will provide information about the location and the APIS will return data to the user depending upon the routine's requirements and suggestive notifications will be sent to the smartwatch while the connection to another remote device or

machine such as a smartphone holds true. It will create a complicate alliance and thus the machine is to be trained using various machine learning algorithms. Data clusters are classified using the classification techniques intended to provide more information about the routine prediction. There are other sensors in the watch such as ambient light sensor, gyroscope, magnetic field sensor, step counter, gravity, linear acceleration, significant motion, etc. which are used collaboratively to obtain a dataset.

#### **4. Significance:**

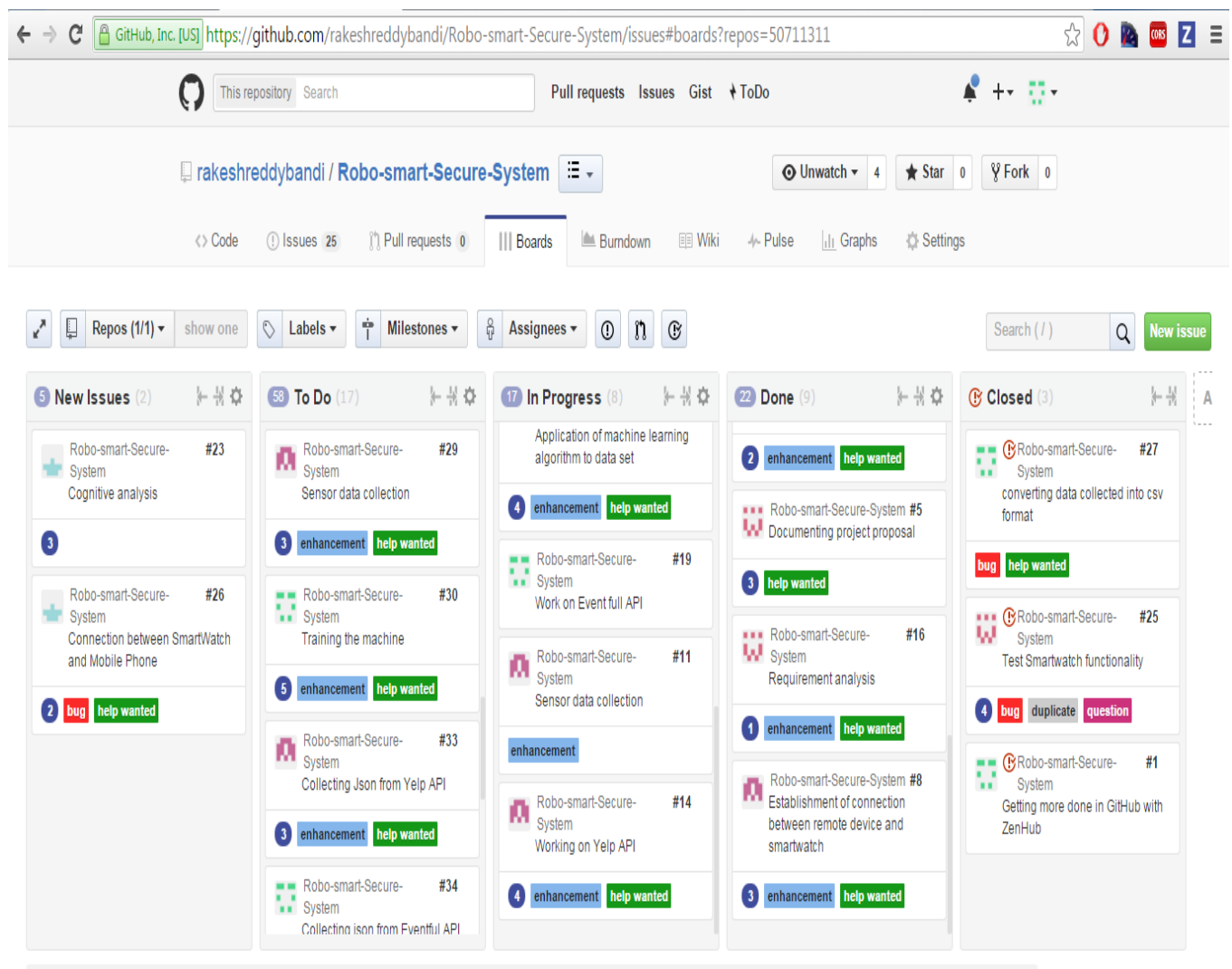
The most important aspect of the project is that an analysis of big data is done in order to predict the routine of a specific user. It will generate a pattern based on the different readings to generate a more sophisticated clustered activity set. This will help in obtaining a clear cut view of a user's routine plus a more generative dimension to a time schedule. Sensors present in the smartwatch will be crucial in collection of data, which in turn is fed into the machine through various machine learning techniques. A stable schedule would prompt the user to receive a collaborative approach to obtain the output as a predictive routine analysis.

### III. 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.

The screenshot displays a GitHub repository page for 'Robo-smart-Secure-System' by user 'rakeshreddybandi'. The URL is <https://github.com/rakeshreddybandi/Robo-smart-Secure-System/issues#boards?repos=50711311>. The repository has 4 Unwatch, 0 Stars, and 0 Forks. The 'Boards' tab is selected, showing a Kanban board with five columns: New Issues (2), To Do (17), In Progress (8), Done (9), and Closed (3). Each column contains a list of issues with their titles, IDs, and labels like 'enhancement', 'bug', 'help wanted', and 'question'.

Column	Issue ID	Issue Title	Labels
New Issues (2)	#23	Robo-smart-Secure-System Cognitive analysis	
	#26	Robo-smart-Secure-System Connection between SmartWatch and Mobile Phone	bug, help wanted
To Do (17)	#31	Robo-smart-Secure-System Predicting the user activities	enhancement, help wanted
	#32	Robo-smart-Secure-System Adding SmartWatch data as training data	enhancement, help wanted
	#24	Robo-smart-Secure-System Notification to Smart Watch	enhancement, help wanted
	#37	Robo-smart-Secure-System	
In Progress (8)	#17	Robo-smart-Secure-System Work on Geo Location API	enhancement
	#12	Robo-smart-Secure-System Application of machine learning algorithm to data set	enhancement, help wanted
	#19	Robo-smart-Secure-System Work on Event full API	
	#11	Robo-smart-Secure-System Sensor data collection	
Done (9)	#10	Robo-smart-Secure-System Research on APIs	enhancement, help wanted, question
	#6	Robo-smart-Secure-System Refine and rebuild project proposal	enhancement
	#22	Robo-smart-Secure-System Modular Description	enhancement, help wanted
	#5	Robo-smart-Secure-System Documentation project proposal	
Closed (3)	#27	Robo-smart-Secure-System converting data collected into csv format	bug, help wanted
	#25	Robo-smart-Secure-System Test Smartwatch functionality	bug, duplicate, question
	#1	Robo-smart-Secure-System Getting more done in GitHub with ZenHub	

### 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.

The screenshot displays the GitHub interface for the repository 'rakeshreddybandi / Robo-smart-Secure-System'. The URL in the browser is <https://github.com/rakeshreddybandi/Robo-smart-Secure-System/issues#boards?repos=50711311>. The repository has 4 Unwatch, 0 Stars, and 0 Forks. The 'Boards' tab is selected, showing a Kanban board with the following columns:

- New Issues**: 3 issues.
  - Issue #23: Robo-smart-Secure-System Cognitive analysis (help wanted).
  - Issue #26: Robo-smart-Secure-System Connection between SmartWatch and Mobile Phone (bug, help wanted).
- To Do**: 17 issues.
  - Issue #13: Robo-smart-Secure-System Testing (enhancement, help wanted).
  - Issue #15: Robo-smart-Secure-System Maintenance (bug, enhancement, help wanted).
  - Issue #18: Robo-smart-Secure-System Debugging (bug, help wanted, question).
  - Issue #31: Robo-smart-Secure-System (no details visible).
- In Progress**: 8 issues.
  - Issue #4: Robo-smart-Secure-System Analyzing the machine learning algorithms (help wanted).
  - Issue #20: Robo-smart-Secure-System Architecture design (enhancement, help wanted).
  - Issue #21: Robo-smart-Secure-System Working on Weather API (enhancement, help wanted).
  - Issue #17: Robo-smart-Secure-System Work on Geo Location API (no details visible).
- Done**: 9 issues.
  - Issue #2: Robo-smart-Secure-System Project Proposal Discussion (enhancement).
  - Issue #7: Robo-smart-Secure-System Knowledge discovery of smarwatch (enhancement, question).
  - Issue #3: Robo-smart-Secure-System Finalizing project plan (enhancement).
  - Issue #10: Robo-smart-Secure-System Research on APIs (no details visible).
- Closed**: 3 issues.
  - Issue #27: Robo-smart-Secure-System converting data collected into csv format (bug, help wanted).
  - Issue #25: Robo-smart-Secure-System Test Smartwatch functionality (bug, duplicate, question).
  - Issue #1: Robo-smart-Secure-System Getting more done in GitHub with ZenHub (no details visible).

## 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.

The screenshot displays the GitHub interface for the repository 'rakeshreddybandi / Robo-smart-Secure-System'. The URL in the browser is <https://github.com/rakeshreddybandi/Robo-smart-Secure-System/issues#boards?repos=50711311>. The repository has 4 Unwatch, 0 Star, and 0 Fork. The navigation bar includes links for Code, Issues (25), Pull requests (0), Boards, Burndown, Wiki, Pulse, Graphs, and Settings. The main content area shows a Kanban board with five columns: New Issues (2), To Do (17), In Progress (8), Done (9), and Closed (3). Each column contains a list of issues with their titles, numbers, and labels (enhancement, bug, help wanted, duplicate, question). The issues are as follows:

Column	Issue Number	Issue Title	Labels
New Issues (2)	#23	Robo-smart-Secure-System Cognitive analysis	
	#26	Robo-smart-Secure-System Connection between SmartWatch and Mobile Phone	bug, help wanted
To Do (17)	#36	Robo-smart-Secure-System Refining all the features	enhancement, help wanted
	#38	Robo-smart-Secure-System Final Documentation	enhancement, help wanted
	#39	Robo-smart-Secure-System Final Documentation Review	enhancement, help wanted
	#19	Robo-smart-Secure-System Work on Event full API	enhancement, help wanted
In Progress (8)	#11	Robo-smart-Secure-System Sensor data collection	enhancement
	#14	Robo-smart-Secure-System Working on Yelp API	enhancement, help wanted
Done (9)	#5	Robo-smart-Secure-System Documenting project proposal	enhancement, help wanted
	#16	Robo-smart-Secure-System Requirement analysis	enhancement, help wanted
	#8	Robo-smart-Secure-System Establishment of connection between remote device and smartwatch	enhancement, help wanted
Closed (3)	#27	Robo-smart-Secure-System converting data collected into csv format	bug, help wanted
	#25	Robo-smart-Secure-System Test Smartwatch functionality	bug, duplicate, question



## 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.

The screenshot displays the GitHub interface for the repository 'rakeshreddybandi / Robo-smart-Secure-System'. The top navigation bar includes links for Pull requests, Issues, Gist, and To Do. The repository name and statistics (Unwatch, Star, Fork) are visible. Below the navigation bar, the 'Boards' tab is selected, showing a Kanban board with five columns: New Issues (2), To Do (17), In Progress (8), Done (9), and Closed (3). Each column contains a list of issues with their titles, numbers, and labels. The issues are categorized by labels such as 'enhancement', 'bug', 'help wanted', 'duplicate', and 'question'. The board provides a visual overview of the project's progress and upcoming tasks.

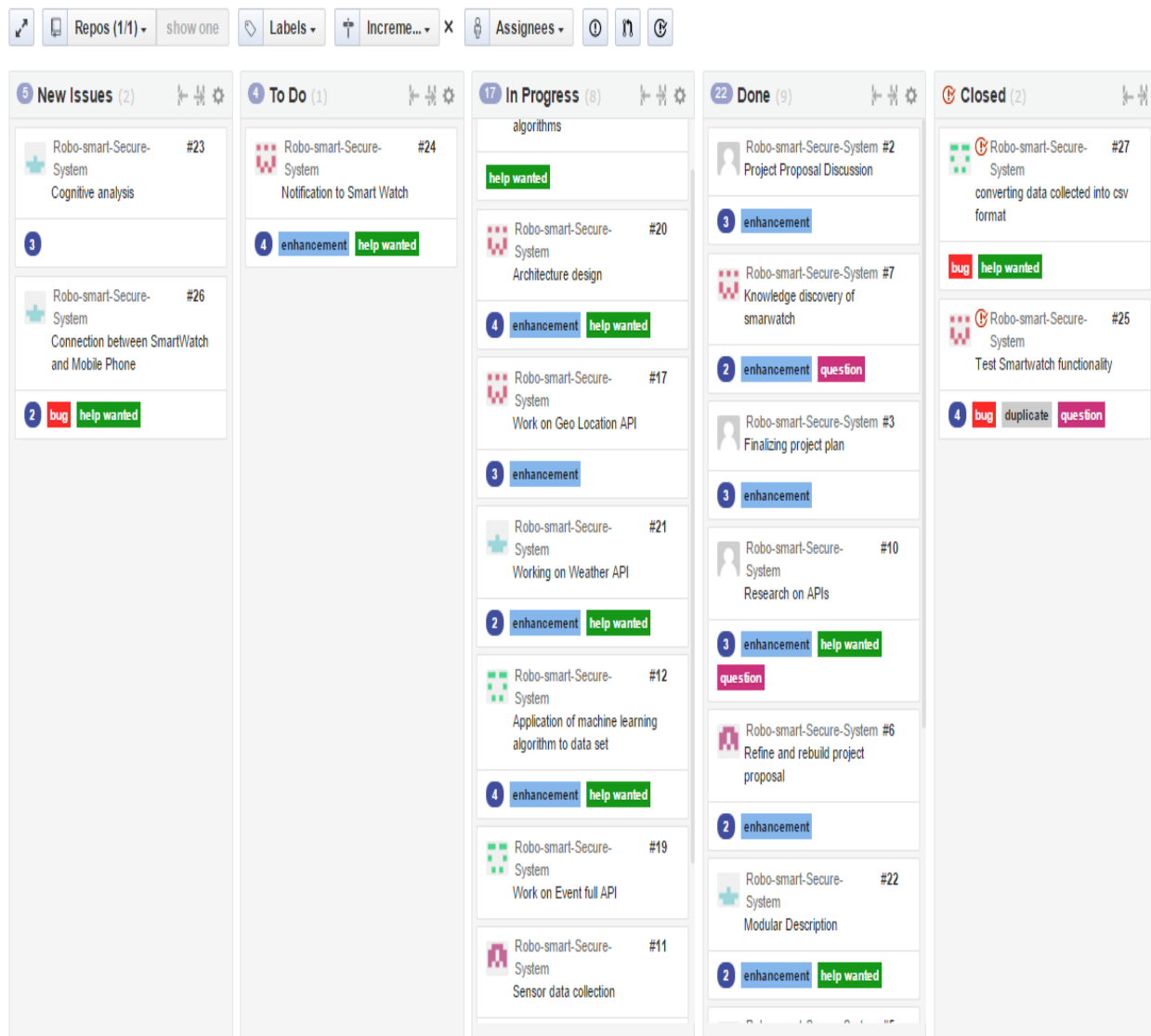
Column	Issue Number	Issue Title	Labels
New Issues (2)	#23	Robo-smart-Secure-System Cognitive analysis	
	#26	Robo-smart-Secure-System Connection between SmartWatch and Mobile Phone	bug, help wanted
To Do (17)	#34	Robo-smart-Secure-System Collecting json from Eventful API	enhancement
	#35	Robo-smart-Secure-System Analyze Location Data	enhancement, help wanted
	#36	Robo-smart-Secure-System Refining all the features	enhancement, help wanted
	#38	Robo-smart-Secure-System	
In Progress (8)		Application of machine learning algorithm to data set	enhancement, help wanted
	#19	Robo-smart-Secure-System Work on Eventful API	
	#11	Robo-smart-Secure-System Sensor data collection	enhancement
	#14	Robo-smart-Secure-System Working on Yelp API	enhancement, help wanted
Done (9)	#5	Robo-smart-Secure-System Documenting project proposal	enhancement, help wanted
	#16	Robo-smart-Secure-System Requirement analysis	enhancement, help wanted
	#8	Robo-smart-Secure-System Establishment of connection between remote device and smartwatch	enhancement, help wanted
Closed (3)	#27	Robo-smart-Secure-System converting data collected into csv format	bug, help wanted
	#25	Robo-smart-Secure-System Test Smartwatch functionality	bug, duplicate, question
	#1	Robo-smart-Secure-System Getting more done in GitHub with ZenHub	

## 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:

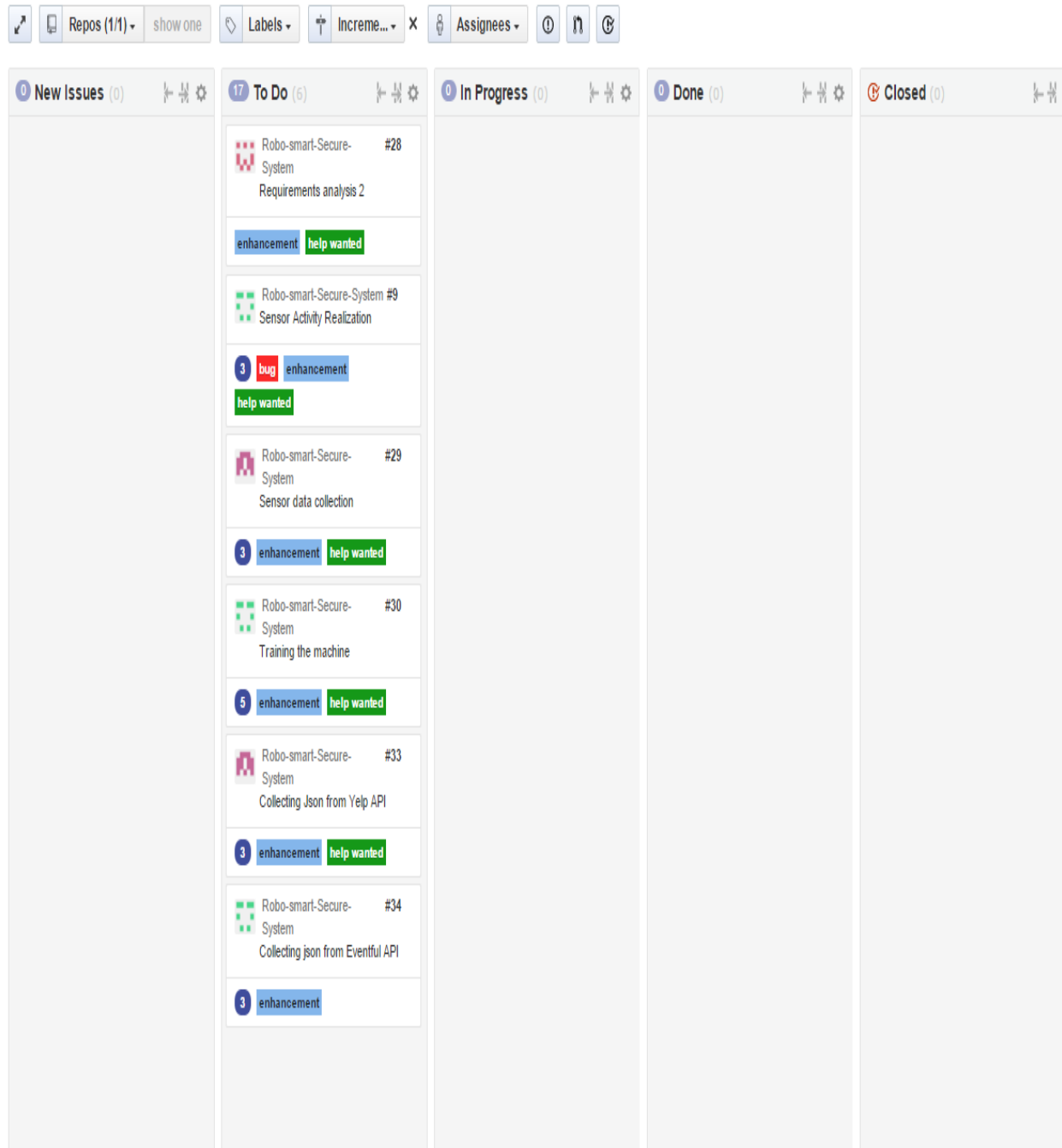
### Increment 1:

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



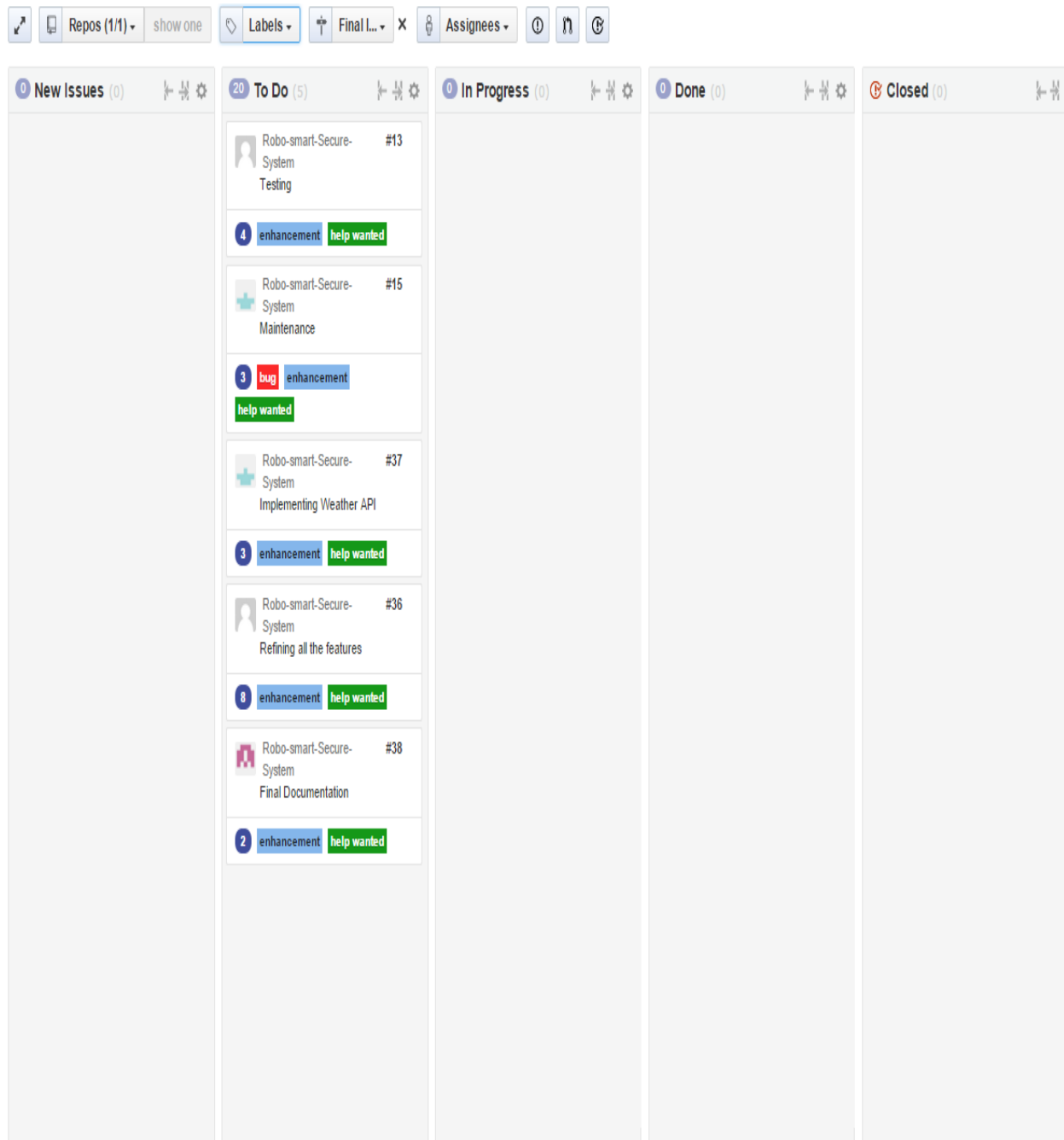
## Increment 2:

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



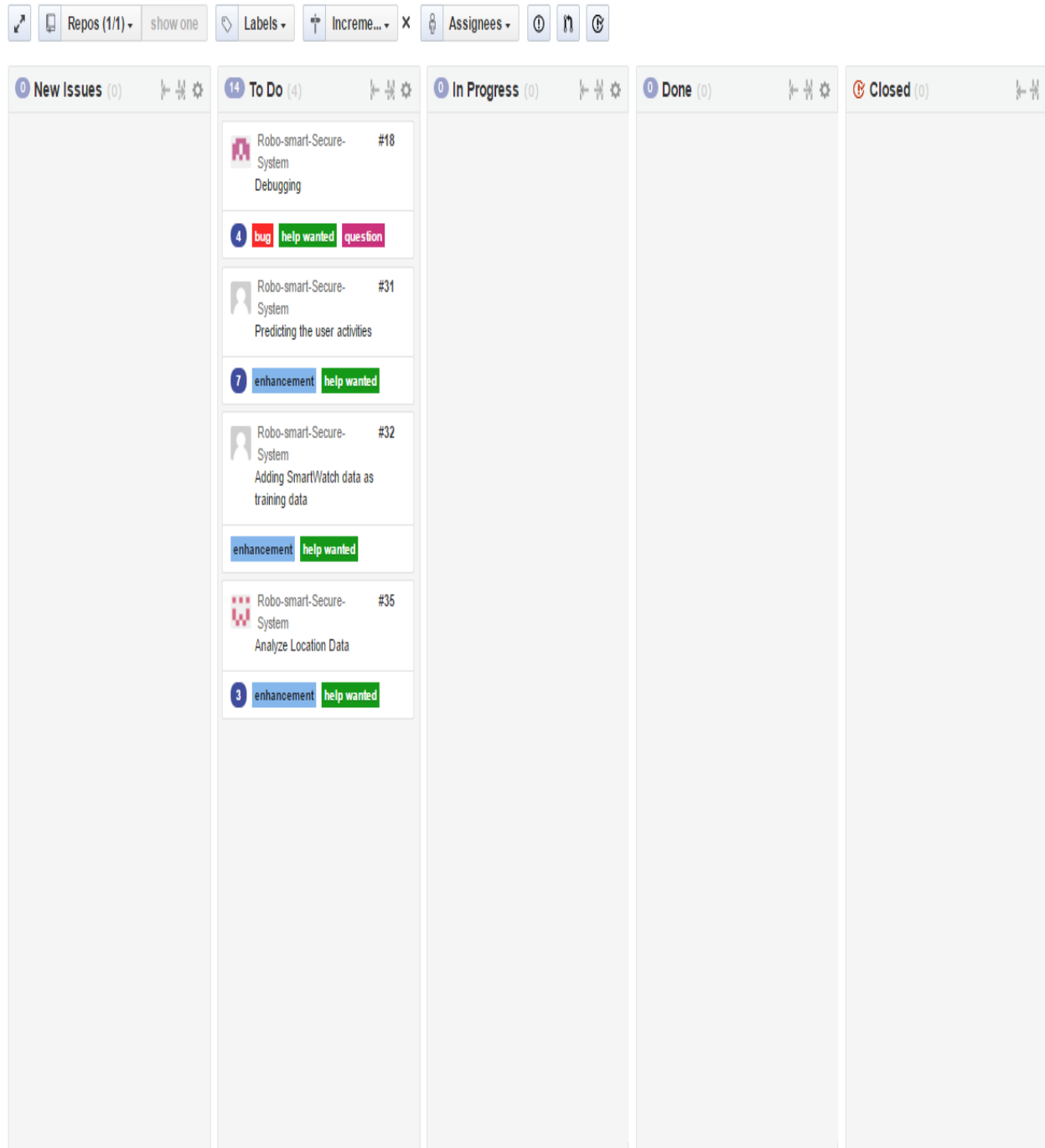
### Increment 3:

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



## Final Increment:

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

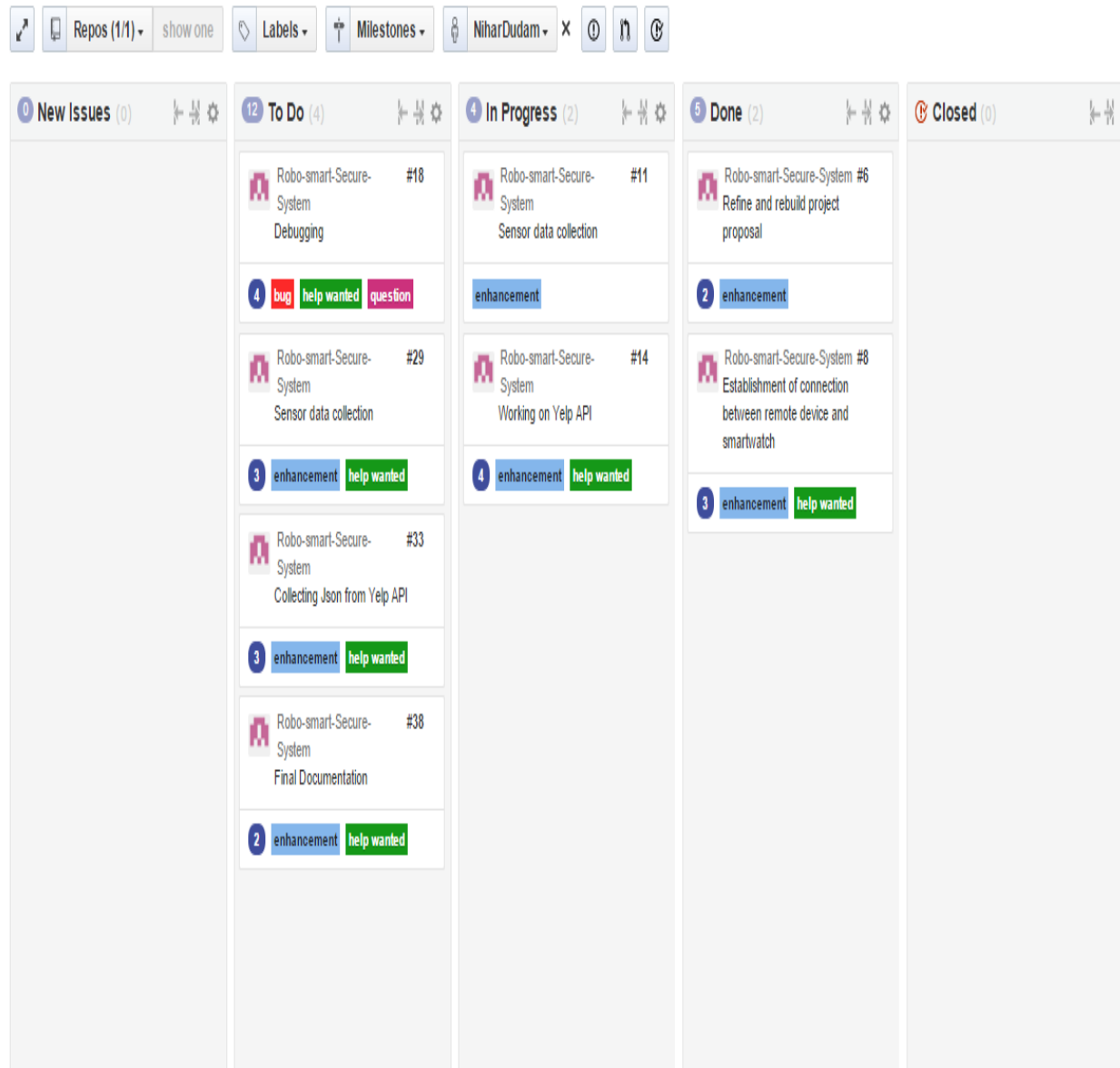


### 3. Project Members and Task Responsibility:

This section deals with the timeline of the project for each member along with the task responsibility:

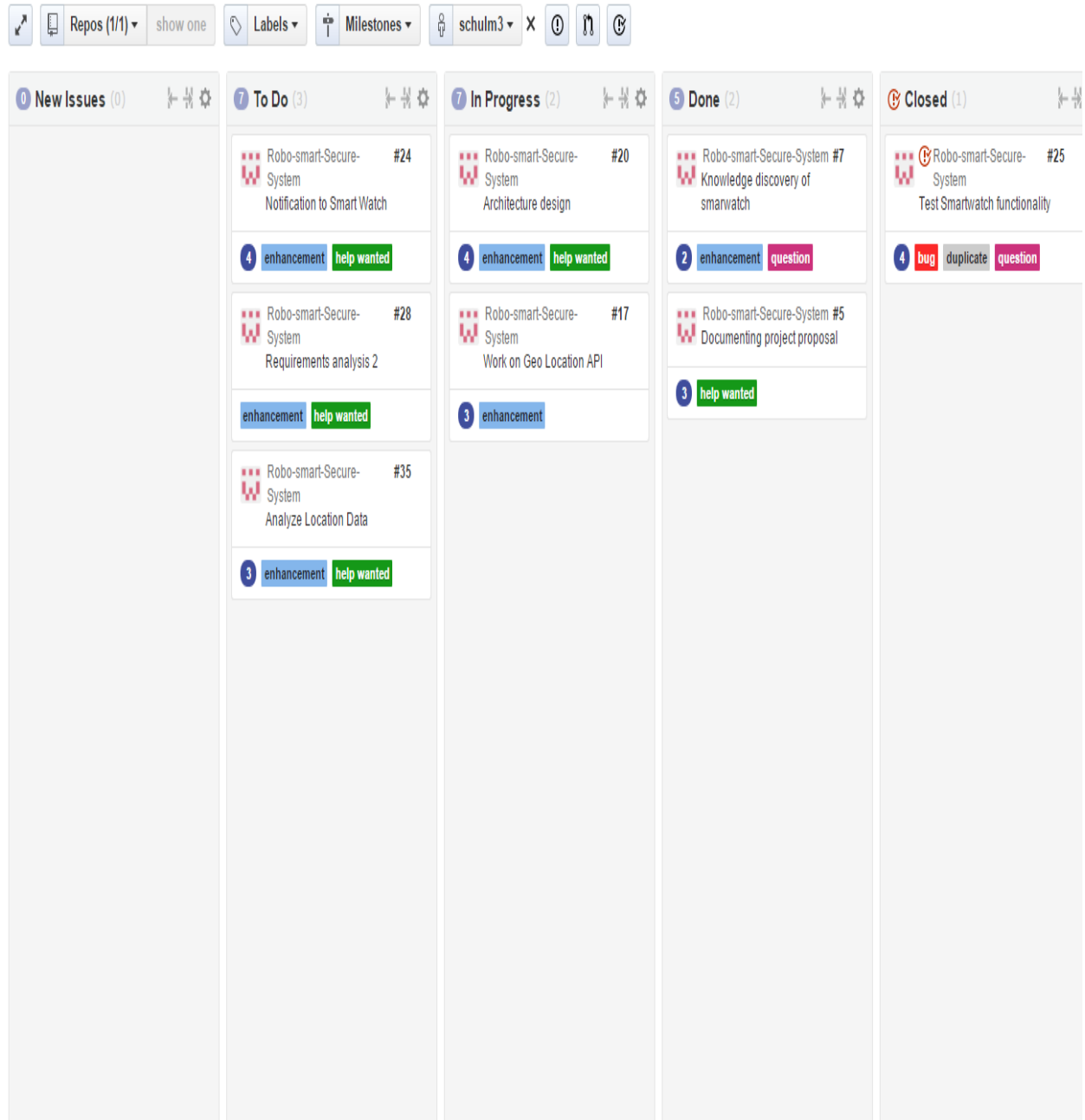
#### a. Nihar Dudam

The tasks assigned to Nihar in order to facilitate the completion of the project has the following diagram:



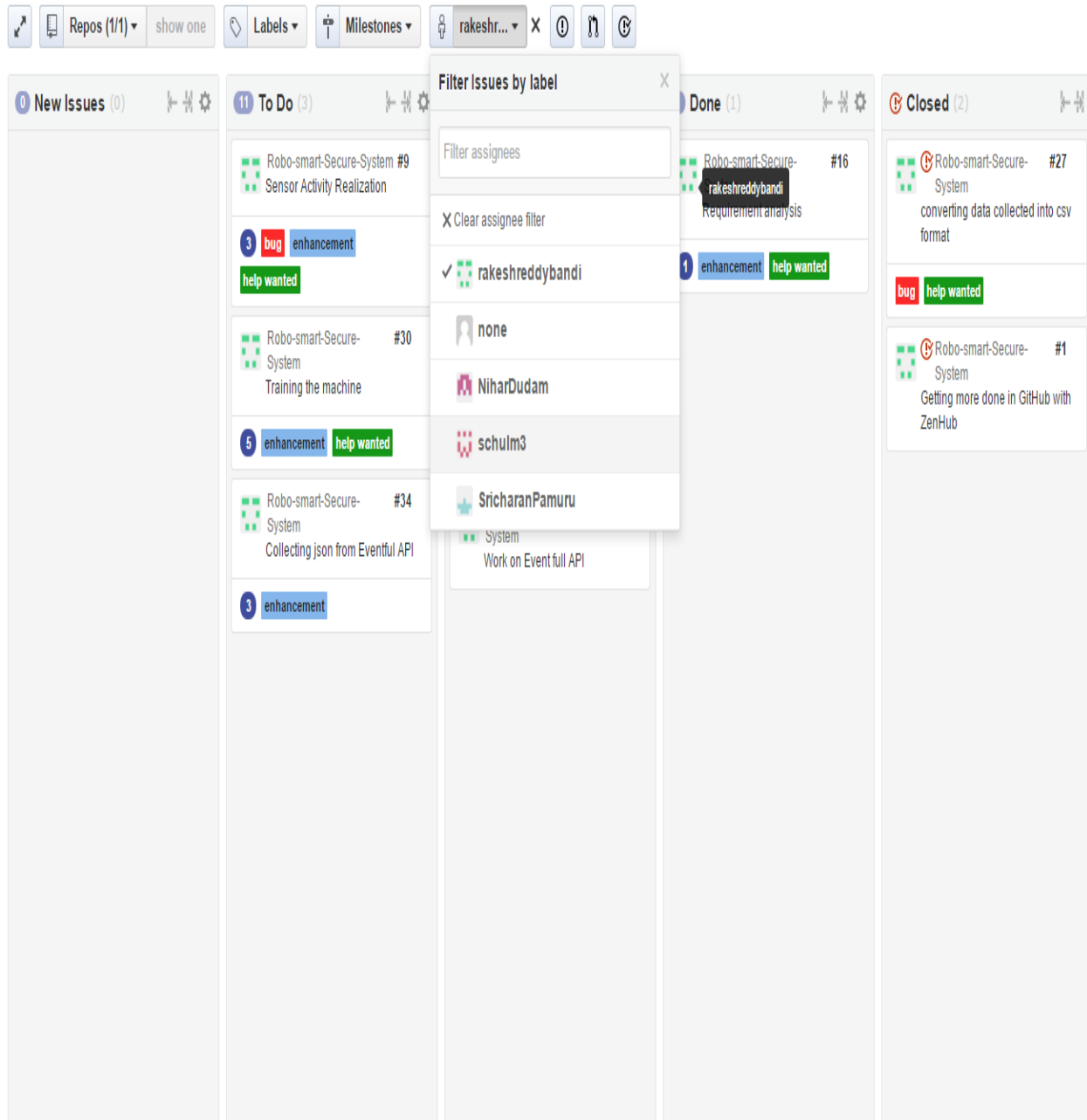
## b. Mark Schultz

The tasks assigned to Mark in order to facilitate the completion of the project has the following diagram:



### c. Rakesh Reddy Bandi

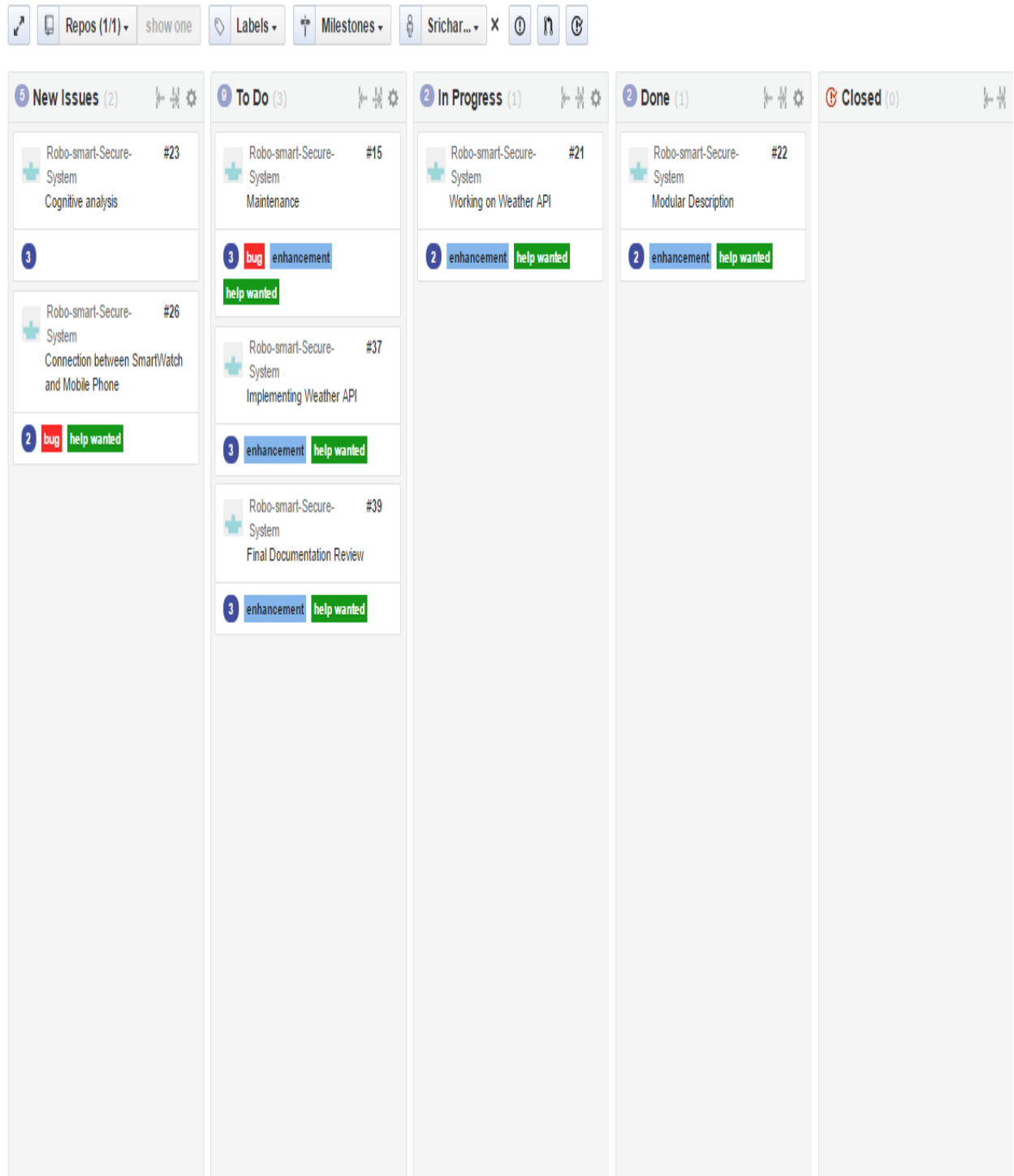
The tasks assigned to Rakesh in order to facilitate the completion of the project has the following diagram:





#### d. Sricharan Pamuru

The tasks assigned to Sricharan in order to facilitate the completion of the project has the following diagram:



#### 4. 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.

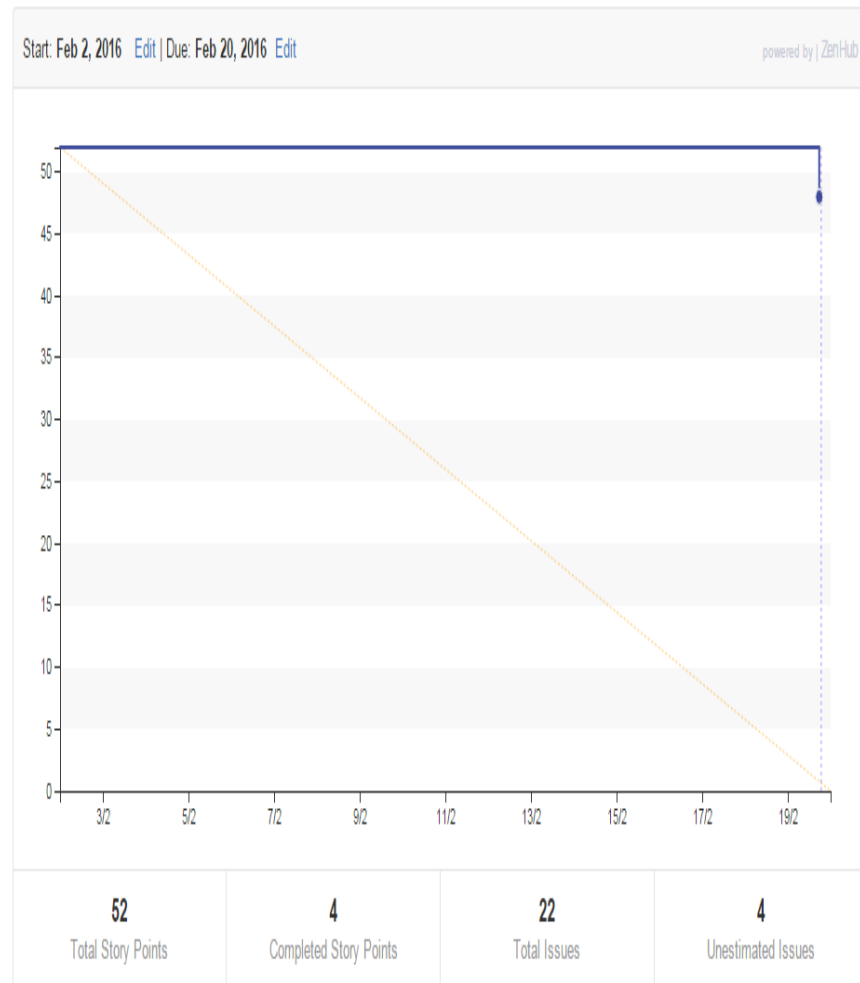
##### Increment 1:

The burndown chart of increment 1:

### Increment 1

[Edit Milestone](#) [Change Milestone](#)

In this increment group need to discuss about project plan and features. How can we implement all the features discussed ?. We should also finalize about the first increment tasks to be done.



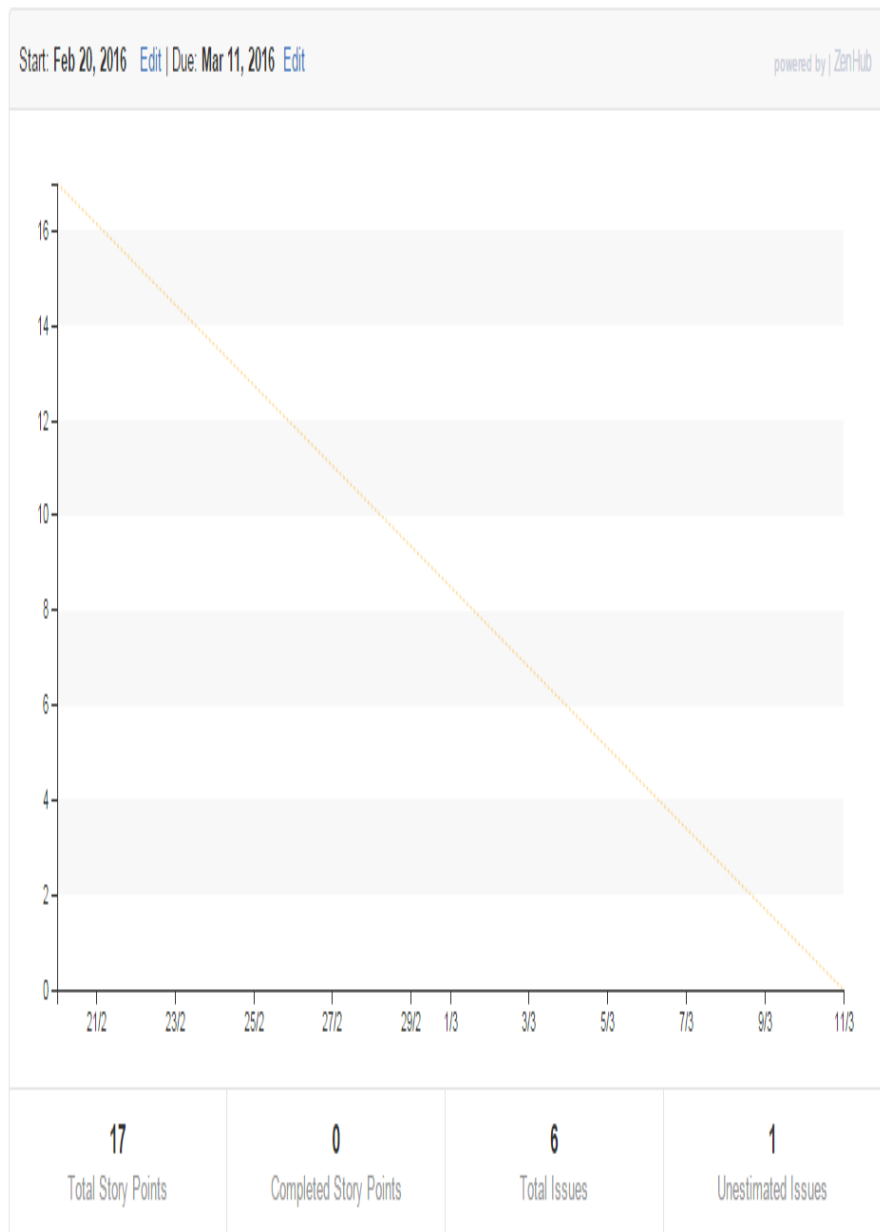
## Increment 2:

The burndown chart of increment 2 is as follows:

### Increment 2



Additional features must be implemented in this increment.



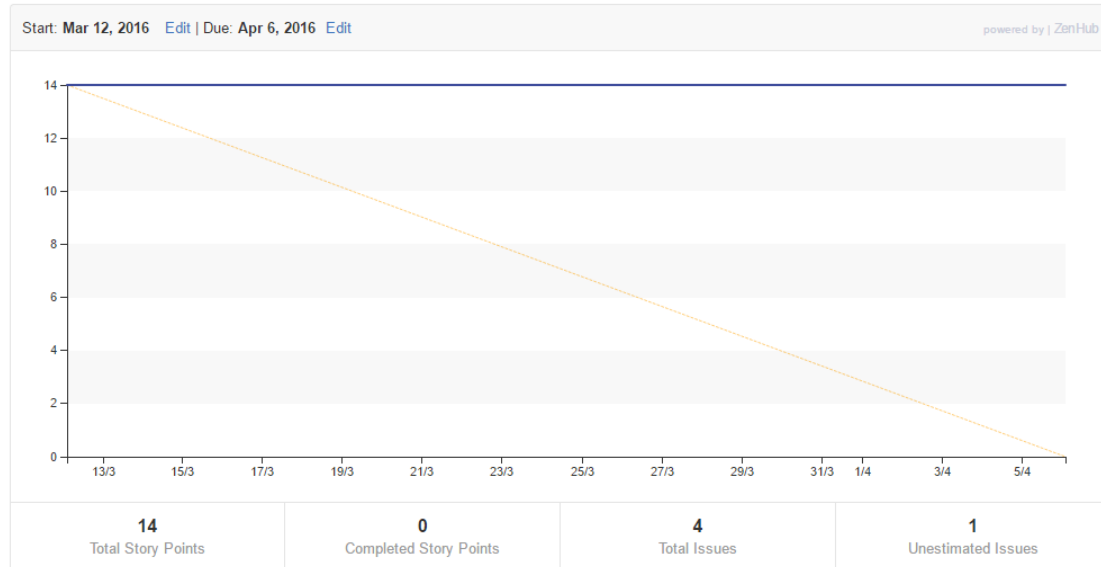
## Increment 3:

The burndown chart of increment 3 is:

### Increment 3

 Edit Milestone  Change Milestone

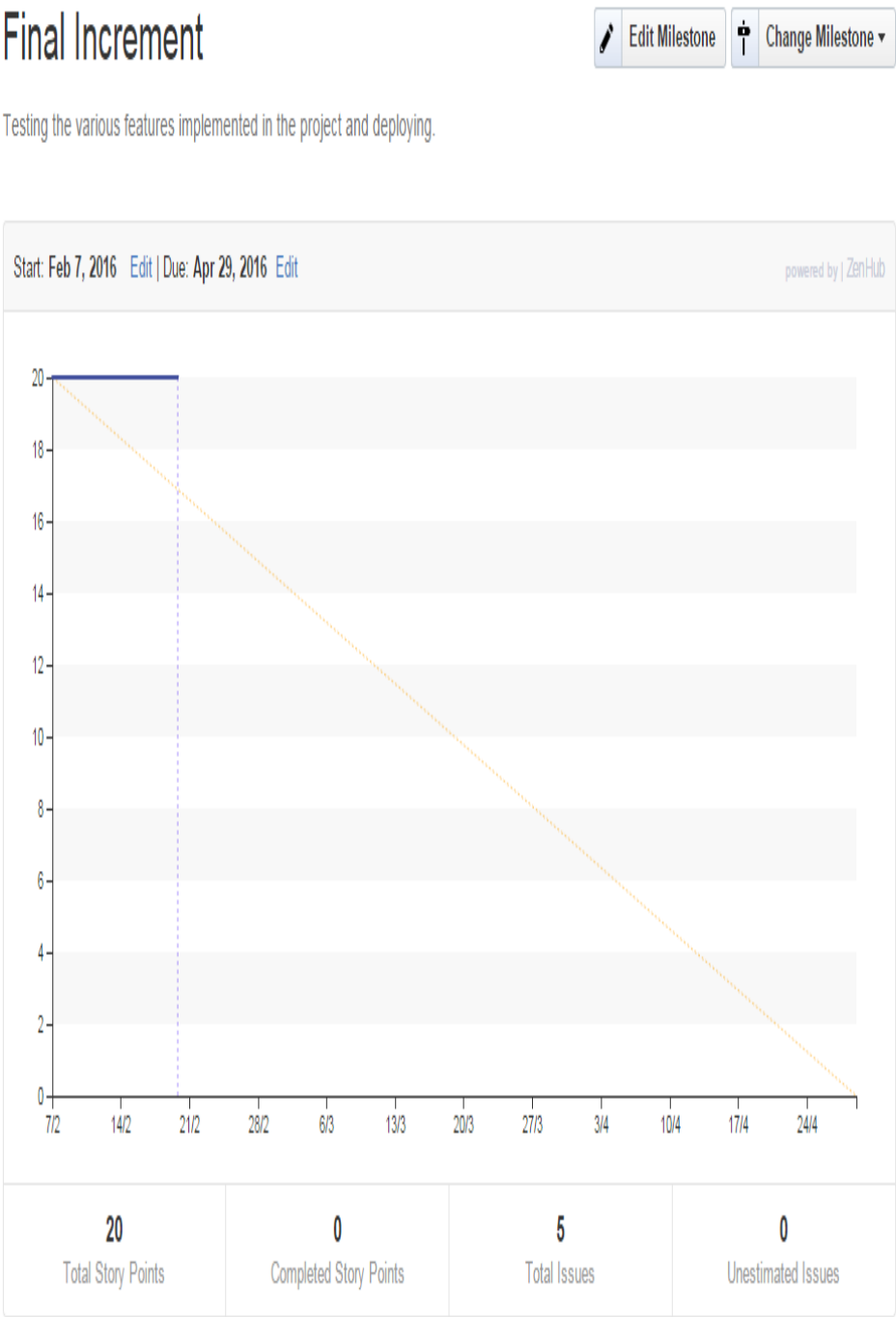
Discussed features must be implemented in this increment.



Increment 3		
Repository	Issues	Story Points
User-Routine-Predictive-Analysis	#31 Predicting the user activities	7
User-Routine-Predictive-Analysis	#18 Debugging	4
User-Routine-Predictive-Analysis	#35 Analyze Location Data	3
User-Routine-Predictive-Analysis	#32 Adding SmartWatch data as training data	Not estimated

Final Increment:

The final increment has the burndown chart as follows:



#### **IV. 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.

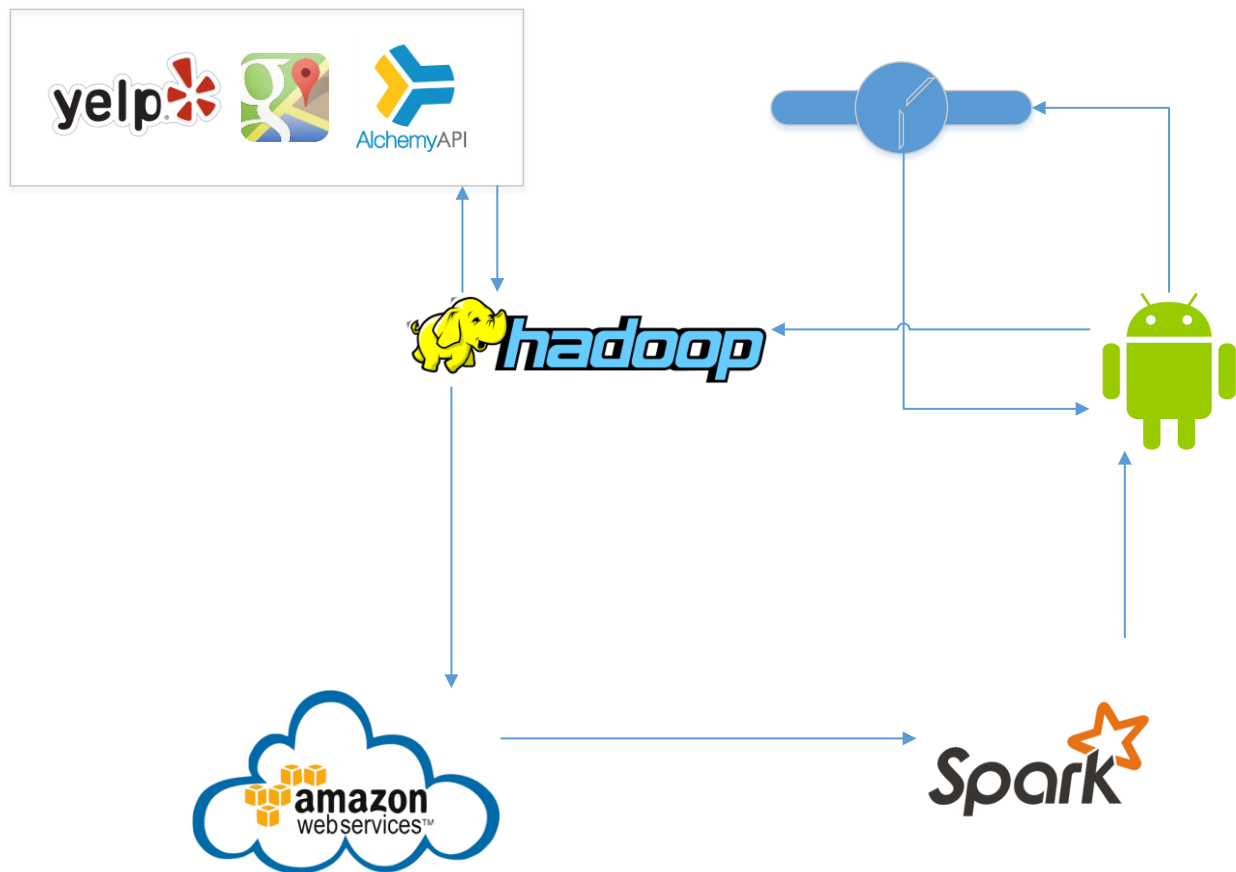
We have our locations, and we have our activities, now we want to link the two together. For example, say we are studying in the library. Assuming the library is cluster 1, our application would append '1::studying' to a text file containing this data. Because the user enjoys studying, we may get the 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 an activity using collaborative filtering, and predict the next location using the maximum probability of the next location.

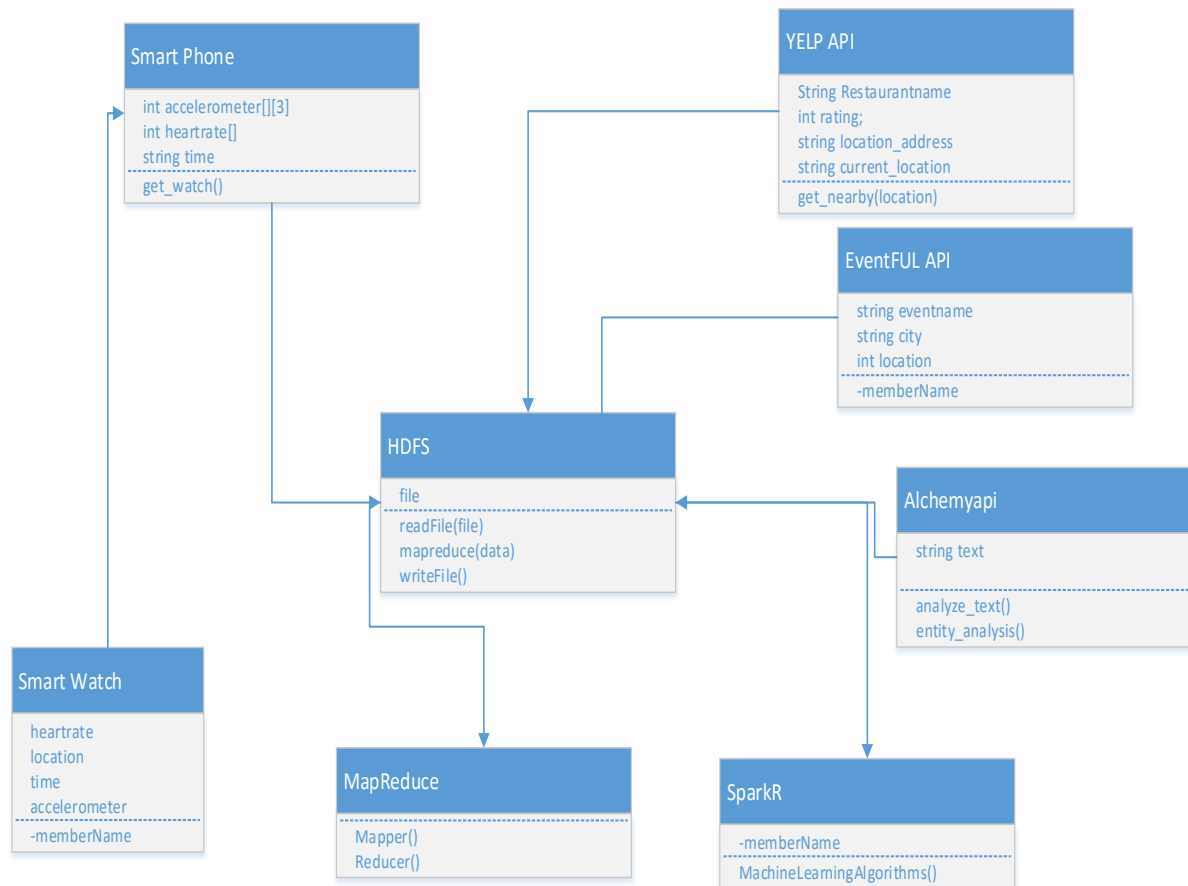
## 1. Design of Features:

### a. 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.

## b. Class Diagram:



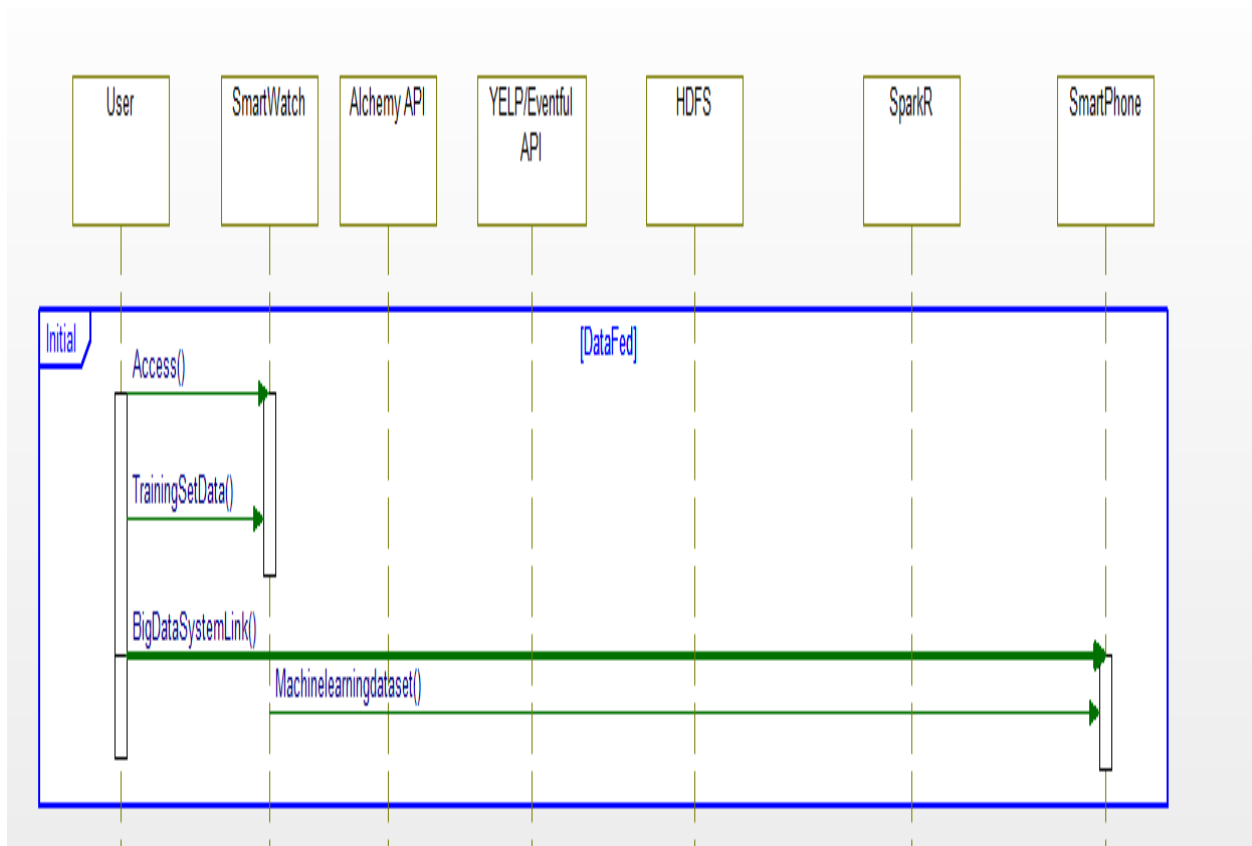
The class diagram we have drawn outlines the structure of our current project. We have three classes for three APIs, which provide data to the HDFS class. Also providing data to the HDFS is the smart phone, which calls `get_watch()` to retrieve smart watch data. The smart watch collects heartrate, location, time, and accelerometer data. HDFS will implement MapReduce and SparkR classes to analyze data it is given. SparkR will implement several of the machine learning algorithms, which will be discussed in more detail later in this paper.



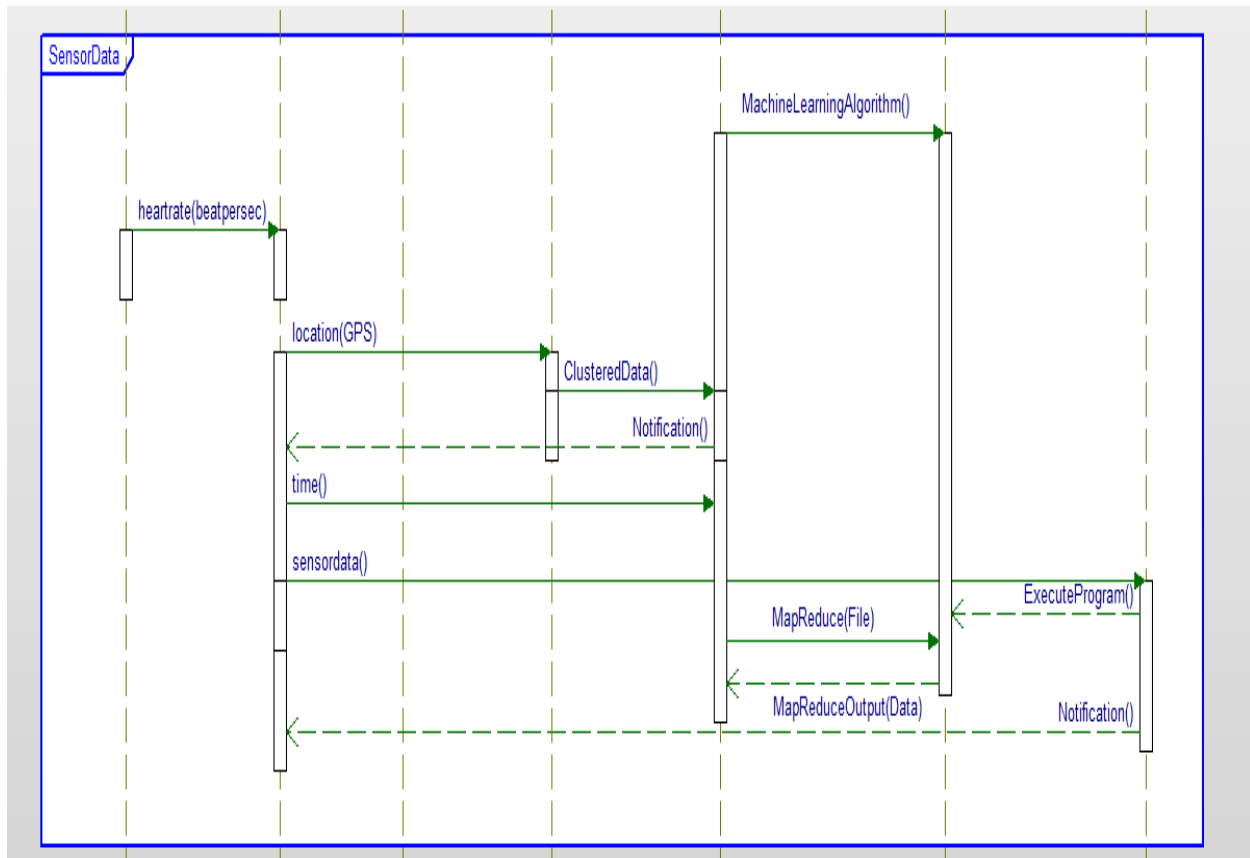
### c. Sequence Diagram:

The sequence diagram is shown as below. The sequence diagram is

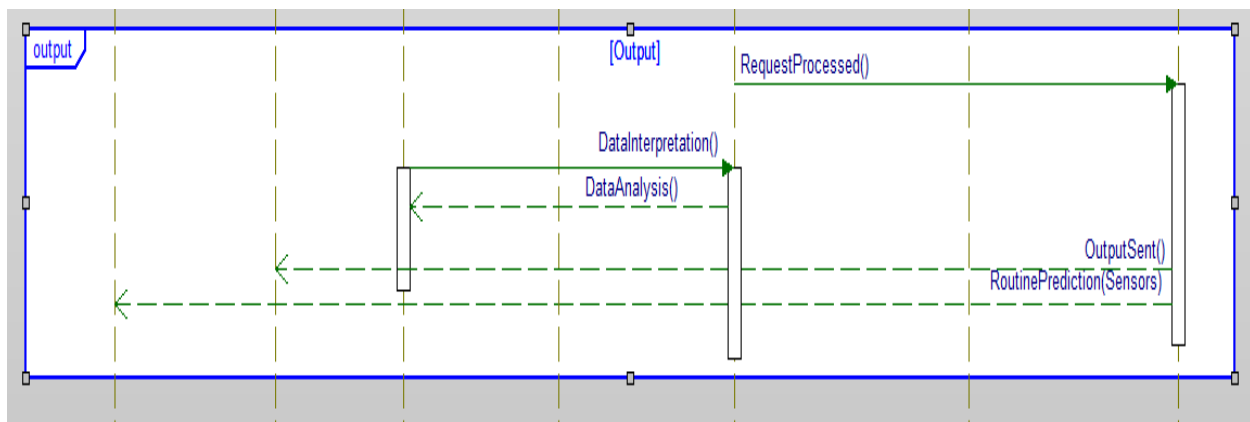
Disintegrated and attached:



The datafed process would mainly consist of all the classes to initially set up communication. The access and training set data is collected and then related with the smartwatch by the user as there is a setup of data fed into the remote machine. The machine learning dataset would be communicated through the smart phone and the device learns as to what the user pattern is.



The sensor data would basically do the operations involved mainly from the HDFS to obtain a programmed output that provides the user as to what the routine is or rather should be.



The final output would basically be a clear cut approach and the user gets a routine predicted.

#### **d. Machine Learning Algorithms:**

In Increment one, we have implemented a naïve bayes algorithm in R to classify datasets of user movements. The naïve bayes is a classifier that uses Bayes theorem to determine the most likely outcome. Bayes Theorem is written as such:  $P(A|B) = \frac{P(A)*P(B|A)}{P(B)}$ . In this equation, P(A) is the prior, P(B|A) is the likelihood, and P(B) is the evidence. This algorithm relies on P(A) and P(B) being independent of one another. While in reality, this is rarely the case, it provides an estimate that is extremely effective in classification.

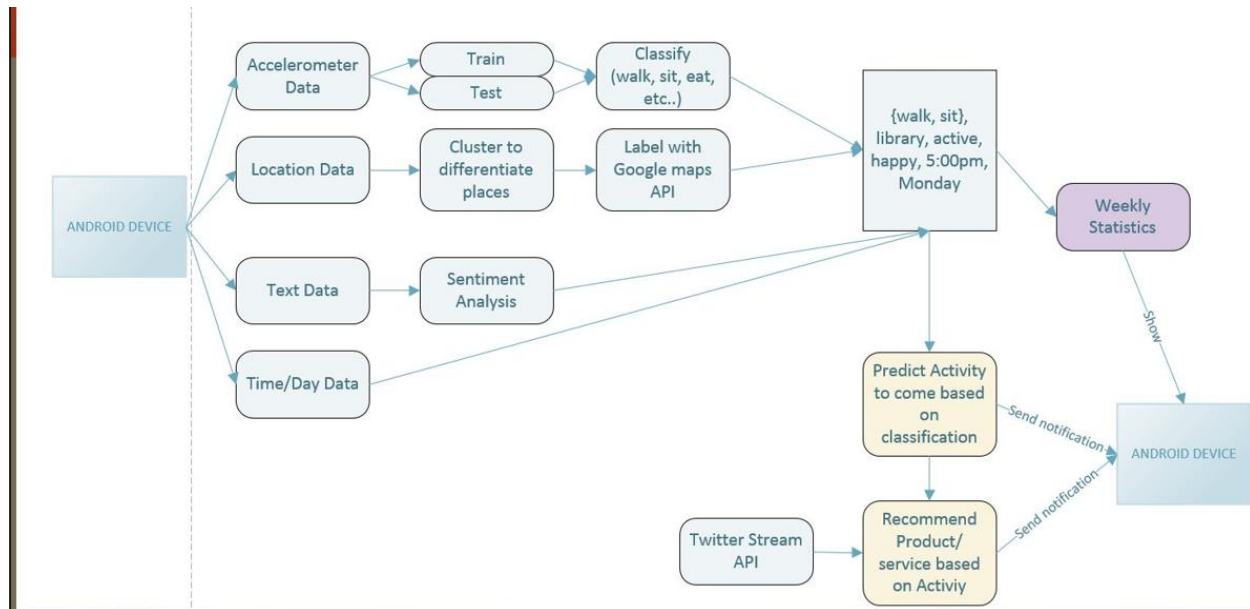
In future increments, we plan on using a several more machine learning algorithms implemented on SparkR to scale larger sets of data.

#### **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.

## 2. Implementation

### ➤ Workflow of project:



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

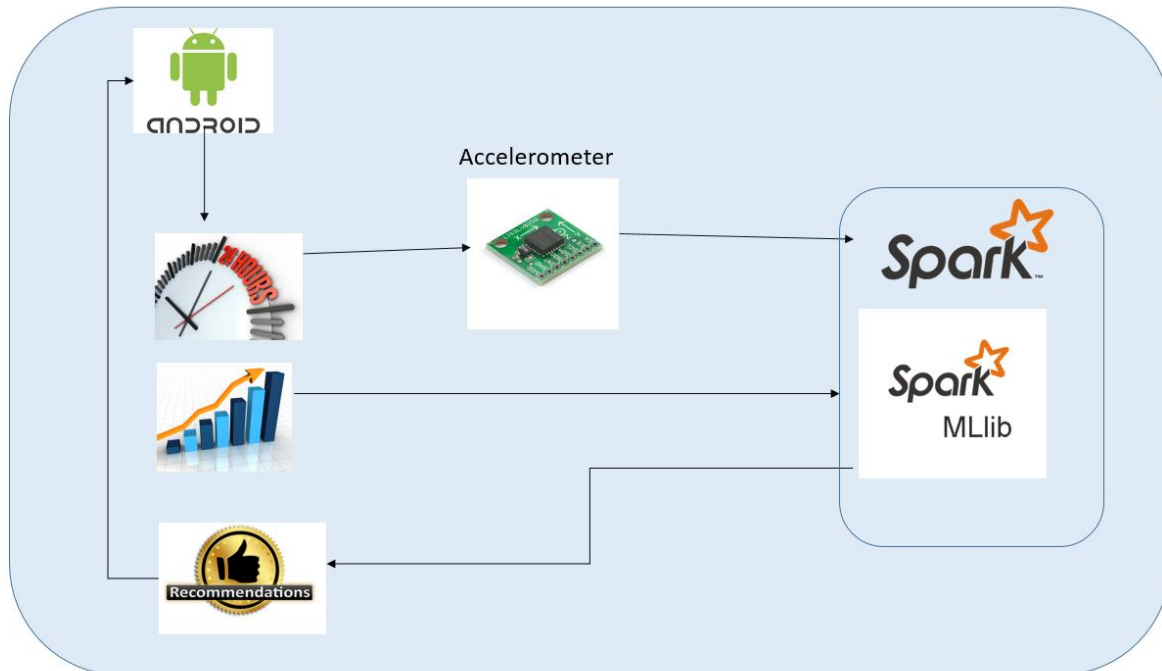
#### Mobile Application for Collecting the Sensor Data:

- 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.

➤ **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

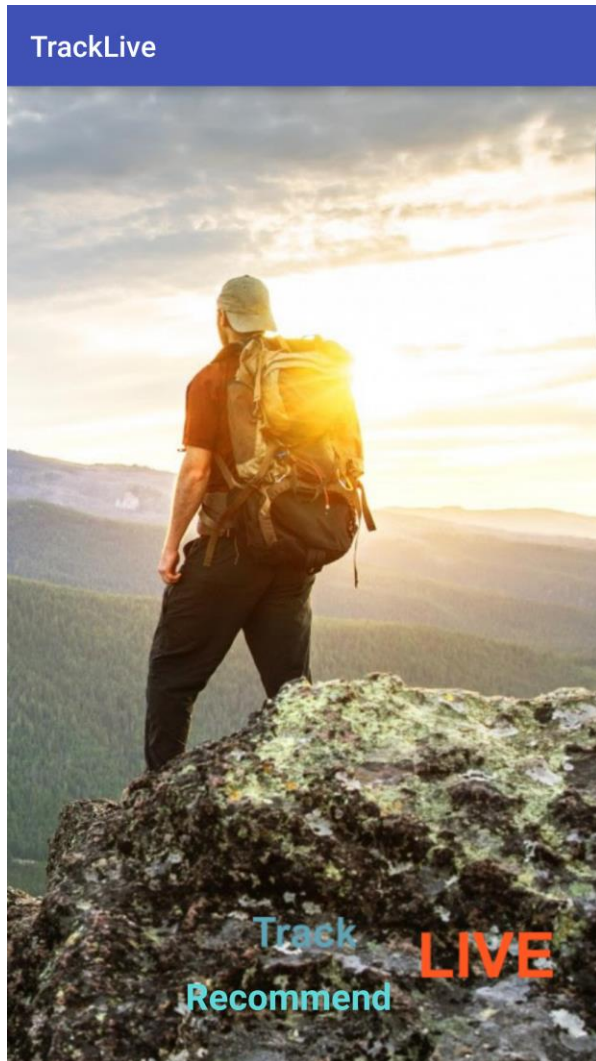
➤ **Machine Learning Application:**

**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

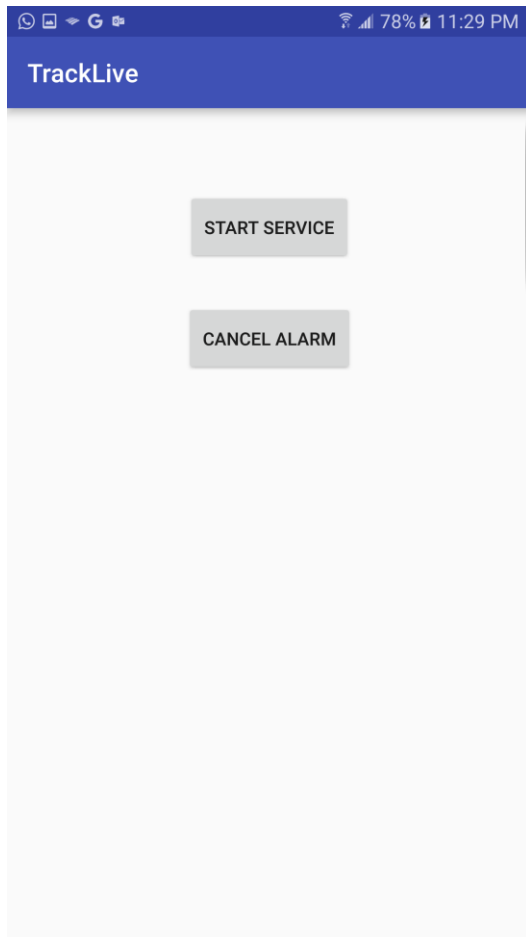
### 3. Implementation

#### 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

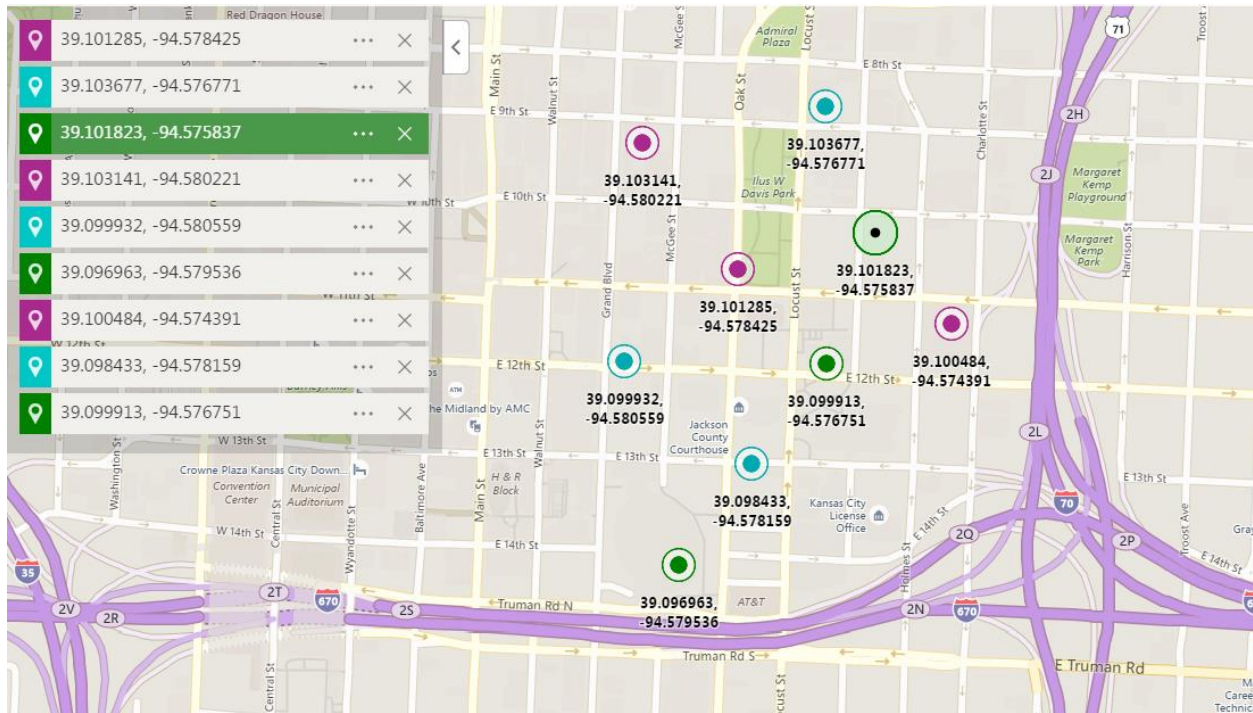


The next screen is followed by two basic buttons, where one is to start the service. The other button used is to cancel alarm.

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:

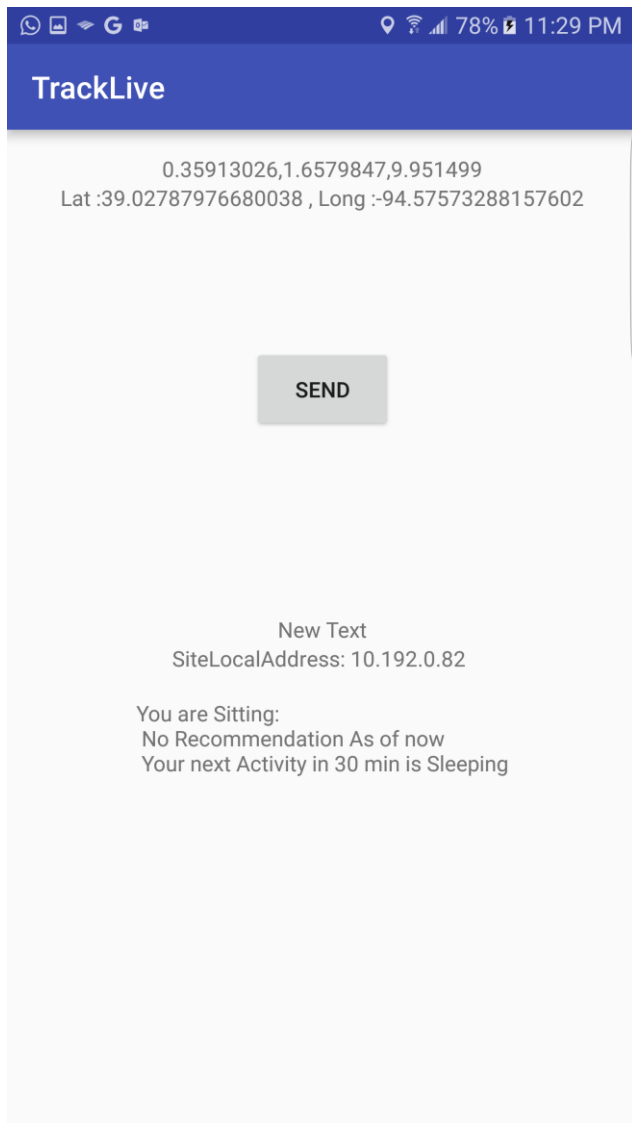


Clusters created near KC-Downtown.

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

```
Run NaiveBayesExample
16/04/06 22:58:26 INFO Executor: Finished task 0.0 in stage 6.0 (TID 13). 3117 bytes result sent to driver
16/04/06 22:58:26 INFO TaskSetManager: Finished task 0.0 in stage 6.0 (TID 13) in 485 ms on localhost (1/1)
16/04/06 22:58:26 INFO DAGScheduler: ResultStage 6 (take at NaiveBayes.scala:216) finished in 0.486 s
16/04/06 22:58:26 INFO TaskSchedulerImpl: Removed TaskSet 6.0, whose tasks have all completed, from pool
16/04/06 22:58:26 INFO DAGScheduler: Job 5 finished: take at NaiveBayes.scala:216, took 0.496154 s
6.0
16/04/06 22:58:26 INFO SparkContext: Invoking stop() from shutdown hook
16/04/06 22:58:26 INFO SparkUI: Stopped Spark web UI at http://192.168.56.1:4040
16/04/06 22:58:26 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
16/04/06 22:58:26 INFO MemoryStore: MemoryStore cleared
Compilation completed successfully in 2s 440ms (29 minutes ago) 107:36 CRLF+ UTF-8+
```

5. Final output screen shown on the android application.



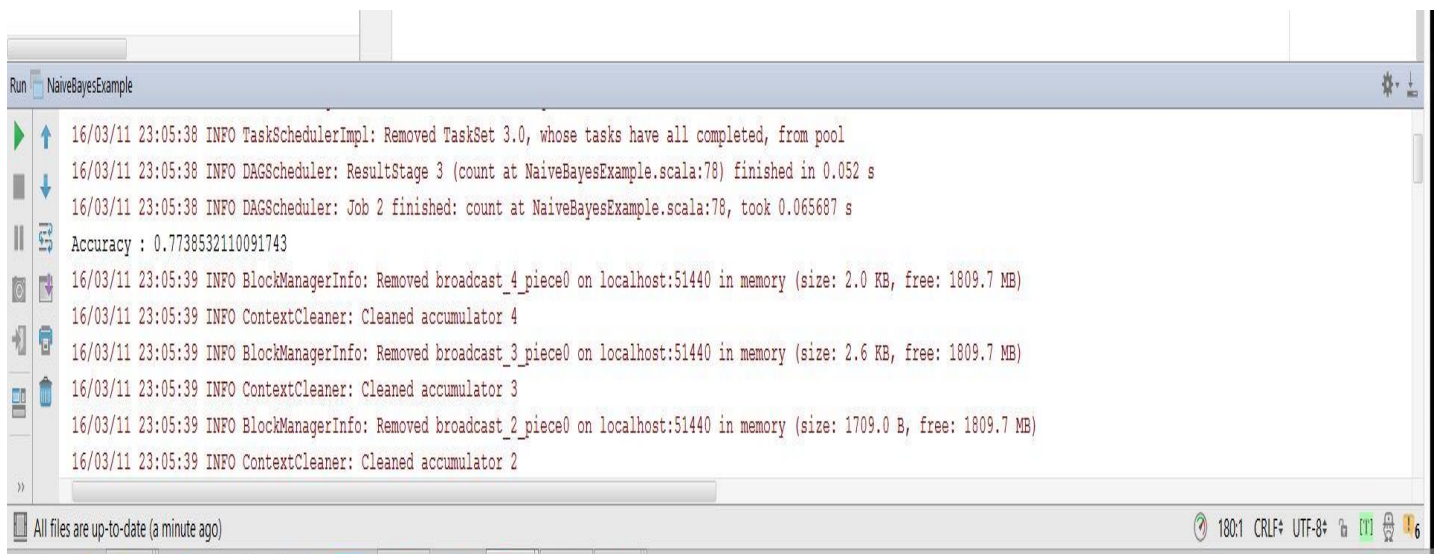
The final implementation screen is obtained from Spark where the server resides. The socket is opened to listen continuously. The main functions occurring here include knowing the current activity from the user accelerometer data. The classification occurs based on the saved model. The probability of 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.

## 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%:



```
Run NaiveBayesExample
16/03/11 23:05:38 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
16/03/11 23:05:38 INFO DAGScheduler: ResultStage 3 (count at NaiveBayesExample.scala:78) finished in 0.052 s
16/03/11 23:05:38 INFO DAGScheduler: Job 2 finished: count at NaiveBayesExample.scala:78, took 0.065687 s
Accuracy : 0.7738532110091743
16/03/11 23:05:39 INFO BlockManagerInfo: Removed broadcast_4_piece0 on localhost:51440 in memory (size: 2.0 KB, free: 1809.7 MB)
16/03/11 23:05:39 INFO ContextCleaner: Cleaned accumulator 4
16/03/11 23:05:39 INFO BlockManagerInfo: Removed broadcast_3_piece0 on localhost:51440 in memory (size: 2.6 KB, free: 1809.7 MB)
16/03/11 23:05:39 INFO ContextCleaner: Cleaned accumulator 3
16/03/11 23:05:39 INFO BlockManagerInfo: Removed broadcast_2_piece0 on localhost:51440 in memory (size: 1709.0 B, free: 1809.7 MB)
16/03/11 23:05:39 INFO ContextCleaner: Cleaned accumulator 2
All files are up-to-date (a minute ago)
180:1 CRLF+ UTF-8+ 6
```

## 2. Displaying Confusion Matrix for the User Activity :



The screenshot shows a console window titled "Run NaiveBayesExample". The console contains 10 lines of text, each stating "Predicted rating for the moves is: (1.0,1.0)". The window has a standard IDE interface with a toolbar on the left and a status bar at the bottom. The status bar indicates "All files are up-to-date (2 minutes ago)" and shows the encoding "59045 CRLF+ UTF-8+".

```
Run NaiveBayesExample
Predicted rating for the moves is: (1.0,1.0)
Predicted rating for the moves is: (1.0,1.0)
Predicted rating for the moves is: (1.0,1.0)
Predicted rating for the moves is: (1.0,1.0)
Predicted rating for the moves is: (1.0,1.0)
Predicted rating for the moves is: (1.0,1.0)
Predicted rating for the moves is: (1.0,1.0)
Predicted rating for the moves is: (1.0,1.0)
Predicted rating for the moves is: (1.0,1.0)
Predicted rating for the moves is: (1.0,1.0)
```

All files are up-to-date (2 minutes ago) 59045 CRLF+ UTF-8+

## 5. Project Management

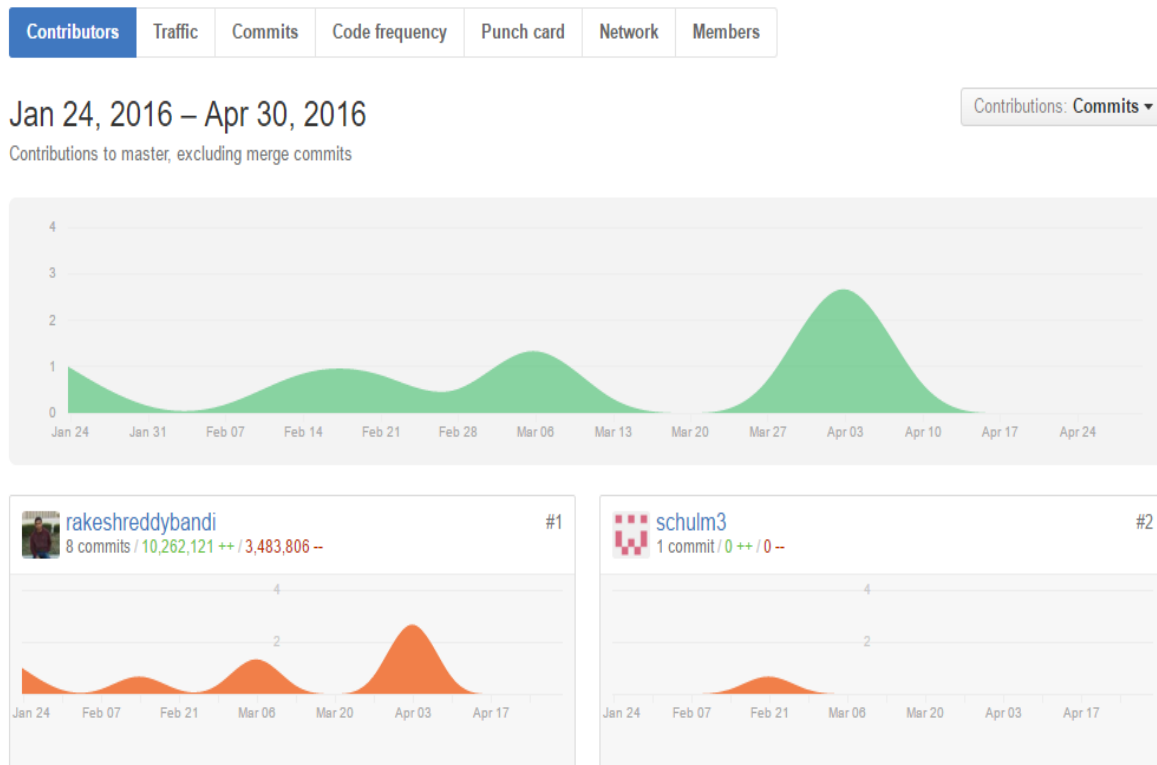
### Increment 4:

The following image shows how the image is categorized and the implementation using the Zenhub Tool.

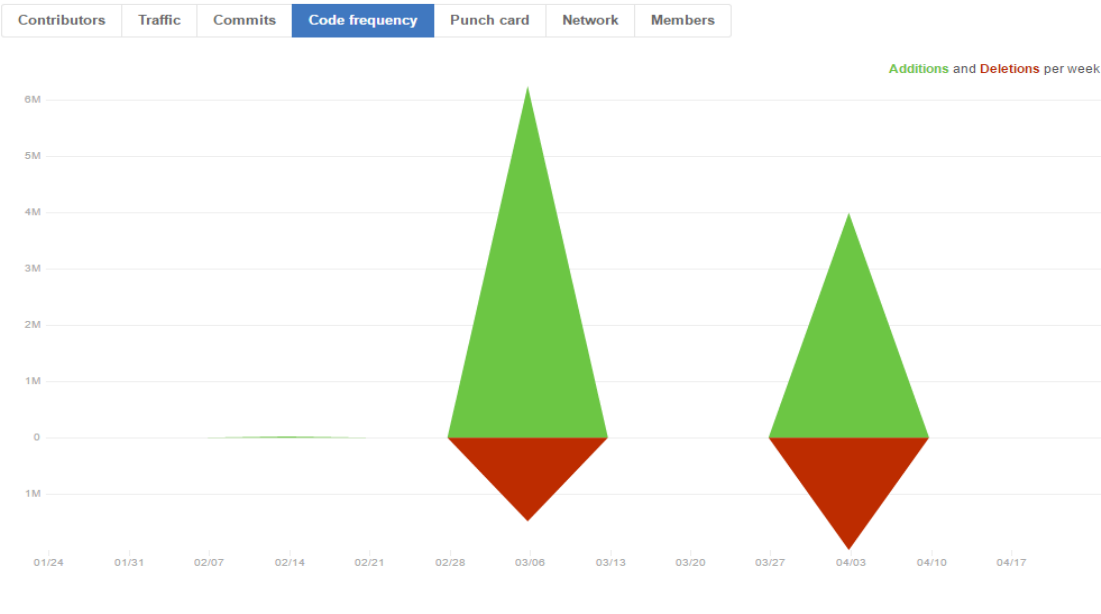
### Graphs:

The following graphs show the analysis of the project management done over the period between first and second increment.

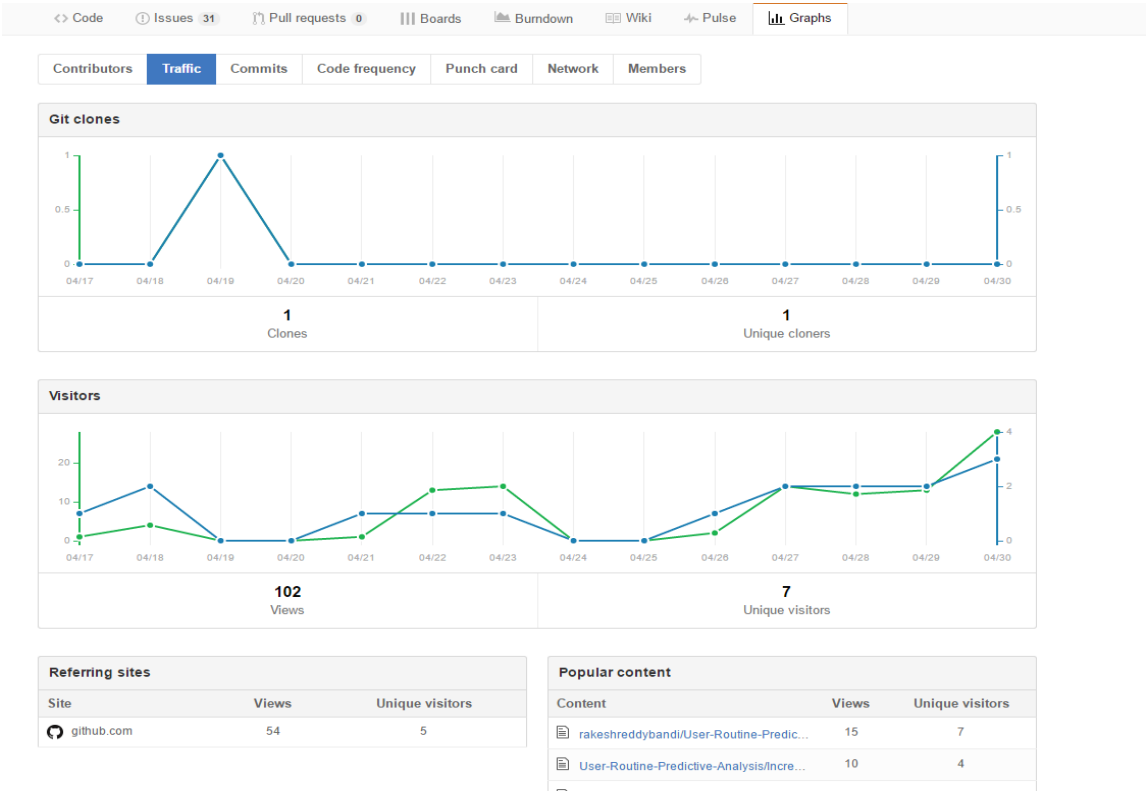
### Contributors:



Code Frequency:



Traffic :



Increment 4 ZenHub tool:

This repository Search Pull requests Issues Gist To Do

rakeshreddybandi / User-Routine-Predictive-Analysis Unwatch 4 Star 0 Fork 0

Code Issues 31 Pull requests 0 Boards Burndown Wiki Pulse Graphs

Repos (1/1) show one Labels Milestones Assignees

Search ( / ) New issue

New Issues To Do In Progress Done Closed Add a Pipeline ...

User-Routine-Predictive-Analysis #38 Final Documentation enhancement help wanted

User-Routine-Predictive-Analysis #39 Final Documentation Review enhancement help wanted

User-Routine-Predictive-Analysis #23 Cognitive analysis enhancement help wanted

User-Routine-Predictive-Analysis #15 Maintenance bug enhancement help wanted

User-Routine-Predictive-Analysis #13 Testing enhancement help wanted

User-Routine-Predictive-Analysis #18 Debugging bug help wanted question

User-Routine-Predictive-Analysis #30 Training the machine enhancement help wanted

User-Routine-Predictive-Analysis #4 Analyzing the machine learning algorithms help wanted

User-Routine-Predictive-Analysis #36 Refining all the features enhancement help wanted

User-Routine-Predictive-Analysis #32 Adding SmartWatch data as training data enhancement help wanted

User-Routine-Predictive-Analysis #31 Predicting the user activities enhancement help wanted

User-Routine-Predictive-Analysis #35 Analyze Location Data enhancement help wanted

User-Routine-Predictive-Analysis #28 Requirements analysis 2 enhancement help wanted

User-Routine-Predictive-Analysis #21 Working on Weather API enhancement help wanted

User-Routine-Predictive-Analysis #37 Implementing Weather API enhancement help wanted

User-Routine-Predictive-Analysis #5 Documenting project proposal help wanted

User-Routine-Predictive-Analysis #3 Finalizing project plan enhancement

User-Routine-Predictive-Analysis #2 Project Proposal Discussion enhancement

User-Routine-Predictive-Analysis #10

- Implementation status report:

### Work completed

Member	Task Description	Member Responsibility	Time (hours)	Comments /Issues
<b>Rakesh</b>	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	
<b>Nihar</b>	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	
<b>Mark</b>	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	
<b>Sricharan</b>	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	



## 6. Bibliography:

- i. Komninos, Andreas, and Mark Dunlop. "Text input on a smart watch." *Pervasive Computing, IEEE* 13.4 (2014): 50-58.
- ii. 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.
- iii. <http://www.motorola.com/us/products/moto-360>
- iv. <https://www.thegrommet.com/robome-app-enabled-robot>
- v. <https://www.yelp.com/developers/documentation/v2/overview>
- vi. <https://api.eventful.com/>