CS5542 BIG DATA ANALYTICS AND APPS

Summary Report (05/11/2016)

Project Group -7

By

Harshini Medikonda (14)

Abhiram Ampabathina (1)

Hirenbhai Harshadbhai Shah(27)

Dinesh Reddy (19)

I.INTRODUCTION:

The heart rate of a person depends on age, gender, daily physical activity, mental stress and many other activities/conditions. Furthermore, there is no proper equipment that can keep a track of heart beat rate. We intend to do a system that can collect the person's daily heart rate activity, store it in a database, analyze the heart rate and the activity the person is performing. Moreover, the application can analyze the data and recommend mental or physical activities to be performed by the user to keep the heart rate optimal. It can also suggest the timings of the abnormal heart rate. All these would give a clear idea of the medical condition of the user and the better usage of it can help in a longer life.

II.PROJECT GOAL AND OBJECTIVES:

OVERALL GOAL:

The goal of the project is to build a system that can take care of the user's health. This heart rate system is an android application which he can view even through the smart watch. This application works with the heart rate sensor embedded in the smart watch. It can observe the patterns of the heart rate and determine the health condition. It recommends the user with the necessary physical and mental activity.

SPECIFIC OBJECTIVE:

The objectives that would be achieved are as follows:

- Collect the heart rate and step count of the user
- Store the heart rate in regular intervals
- Get the heart rate onto HDFS per day basis
- Analyze it using machine learning algorithms.
- Notifying the health conditions using smart watch and smart phone
- Recommend the activities to be done by the user.
- Have a medical record, convenient and cost efficient.
- Suggest users items with items the has a particular nutrition.
- Recommend user with the other related items having same nutritional values

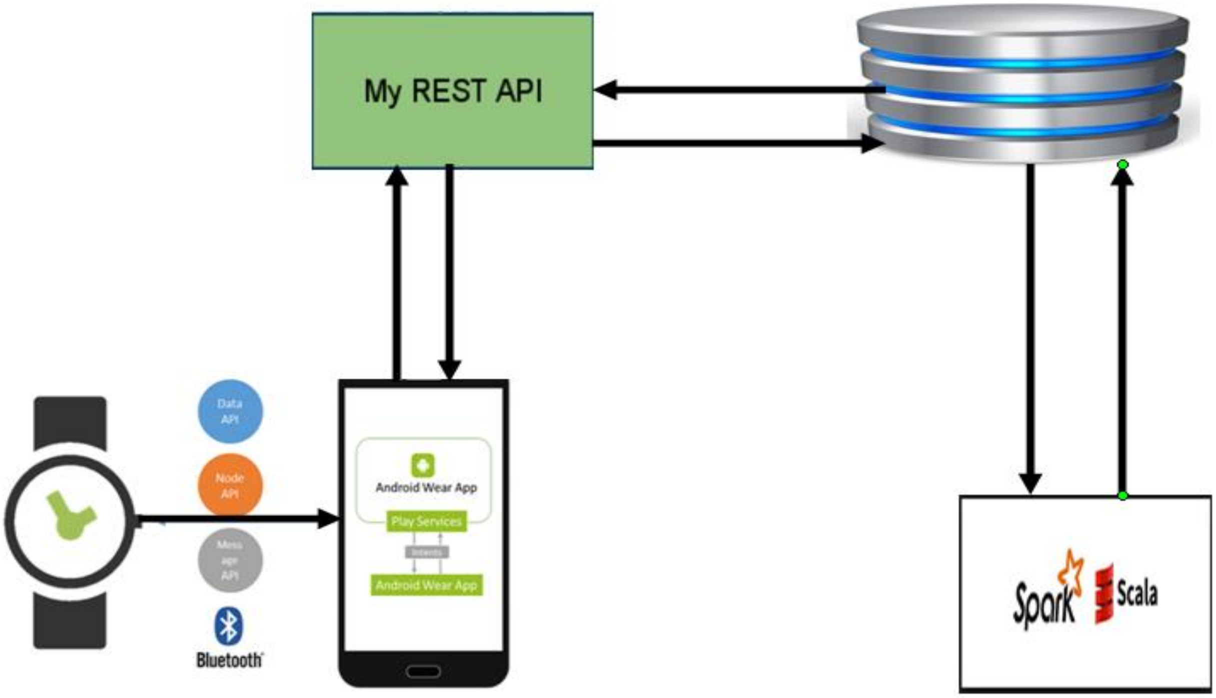
SPECIFIC FEATURES:

The specific features designed in the project are:

- Heart beat analysis
- Sleep pattern analysis
- Notification of current health condition
- Recommendation of health care
- Recommendation of products of good nutritional value
- Suggesting user with the products having a particular nutrition
- Select any date and view the corresponding day patterns

II. OUR SOLUTIONS:

ARCHITECTURE:



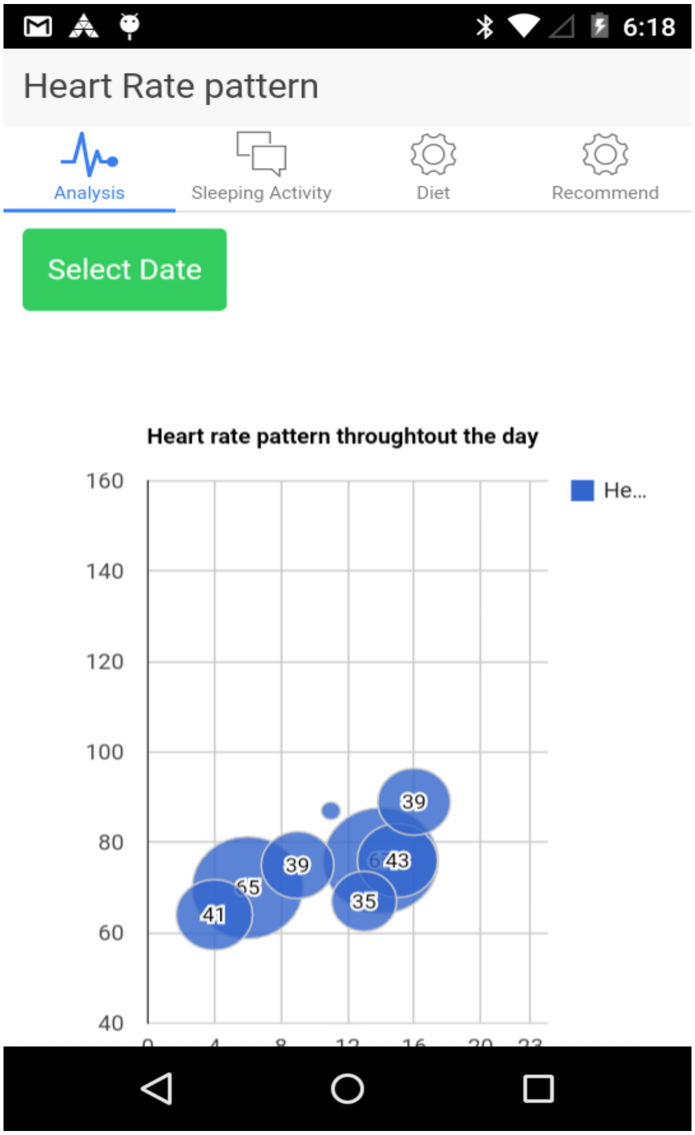
APPLICATION:

The application consists of the feature that tells the heart rate pattern of the day. It analyzes the heart beat collected throughout the day. This is done using Apriori algorithm in Machine Learning. For this application to run, two devices are required:

1. Andriod wear
2. Android mobile

where you can visualize the results.

From this visualization, one can get to know during which hour the heat pattern went extremely high and how many times this happened and the location of it. So, this would caution the user and he can take extra care.

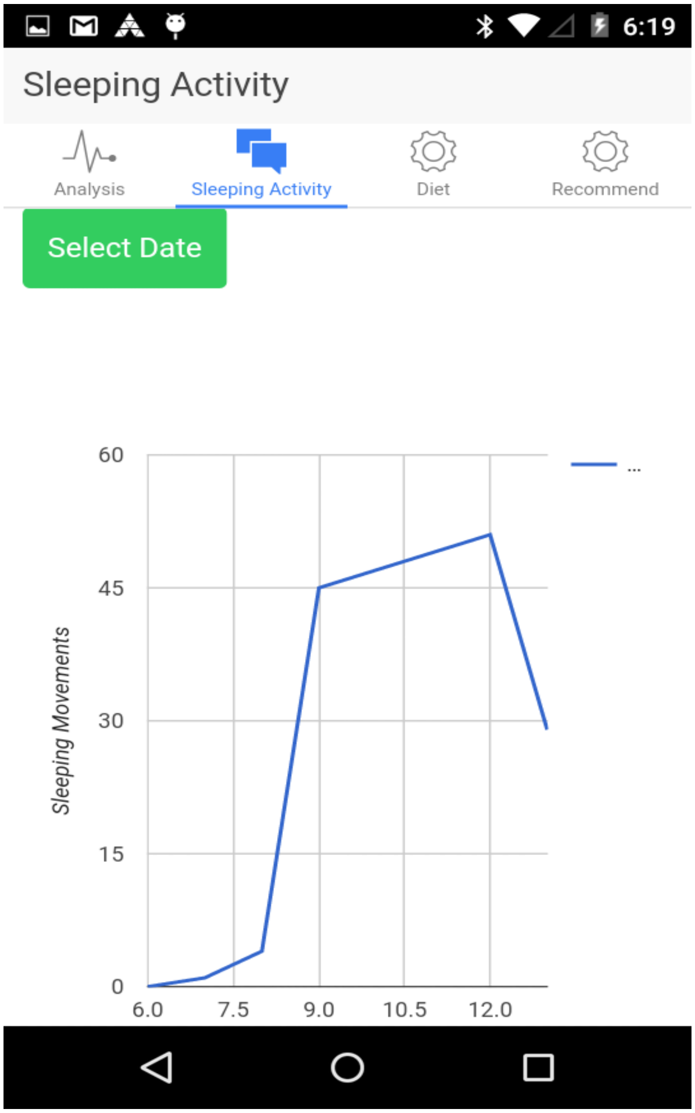


Heart Rate pattern

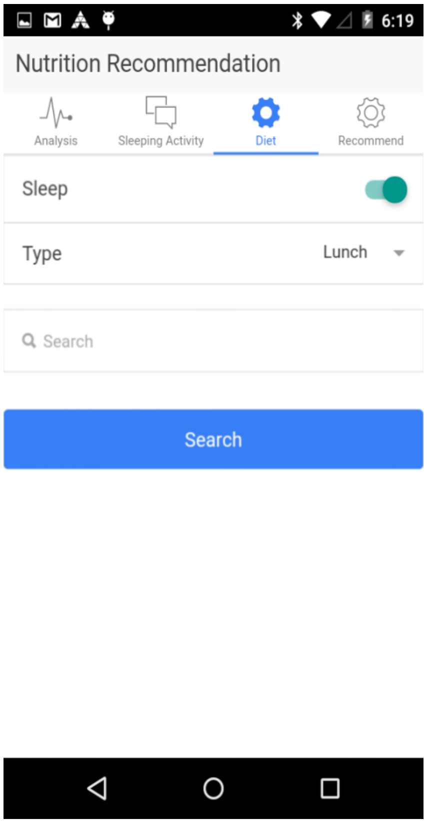
Sleeping Activity:

There is one more feature in the application where the user can the change the date and view that particular day's pattern accordingly. This graph gives the user about the idea of how well he slept last night. The user has to turn on the toggle when he is about to sleep and turn it off once he wakes up.

The analysis is done between those timings. A Spark Job is run that analyzes the movements of the user, whether he was continuously moving or not. This data is collected from the accelerometer sensor in the watch that he wears each night.



The screenshot of toggle button is given below:

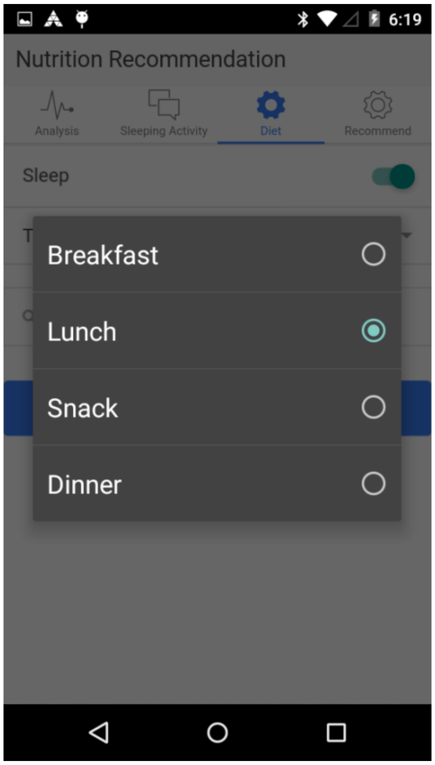


Sleep ●

Type Lunch

DIET:

The user can prefer the meal type as shown in the below screenshot.
It helps in recommending the user. Taking into consideration the nutritional values.



Breakfast ○
Lunch ●
Snack ○
Dinner ○

RECOMMENDATIONS:

In recommendations tab, we have 2 features:

1. User can choose what item has to be present in his diet

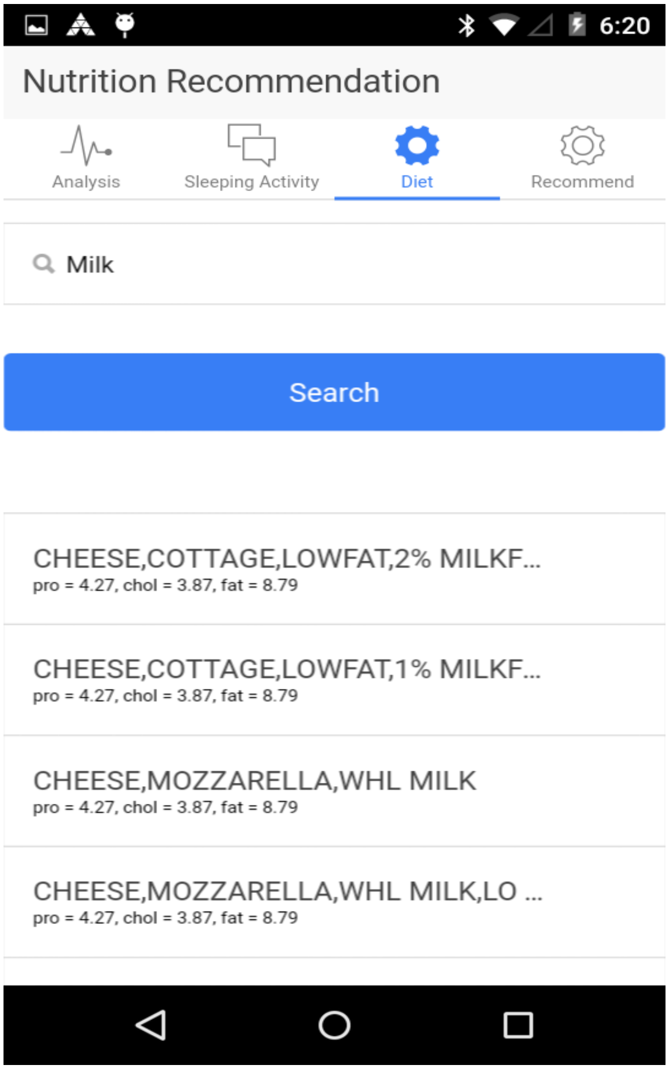
   Example : milk

2. We give the user the items containing milk. Then the user can select the items based on the nutritional values mentioned. Then we recommend similar items based on the nutritional values of the user preferred item.

Motivation behind this is:

Everyday, we think too much select an item what to eat. Moreover we do not know the nutritional values of that particular item.

The items database was collected from National Nutrition Database which was available online. Got access to it after requesting for permission and it was so helpful.

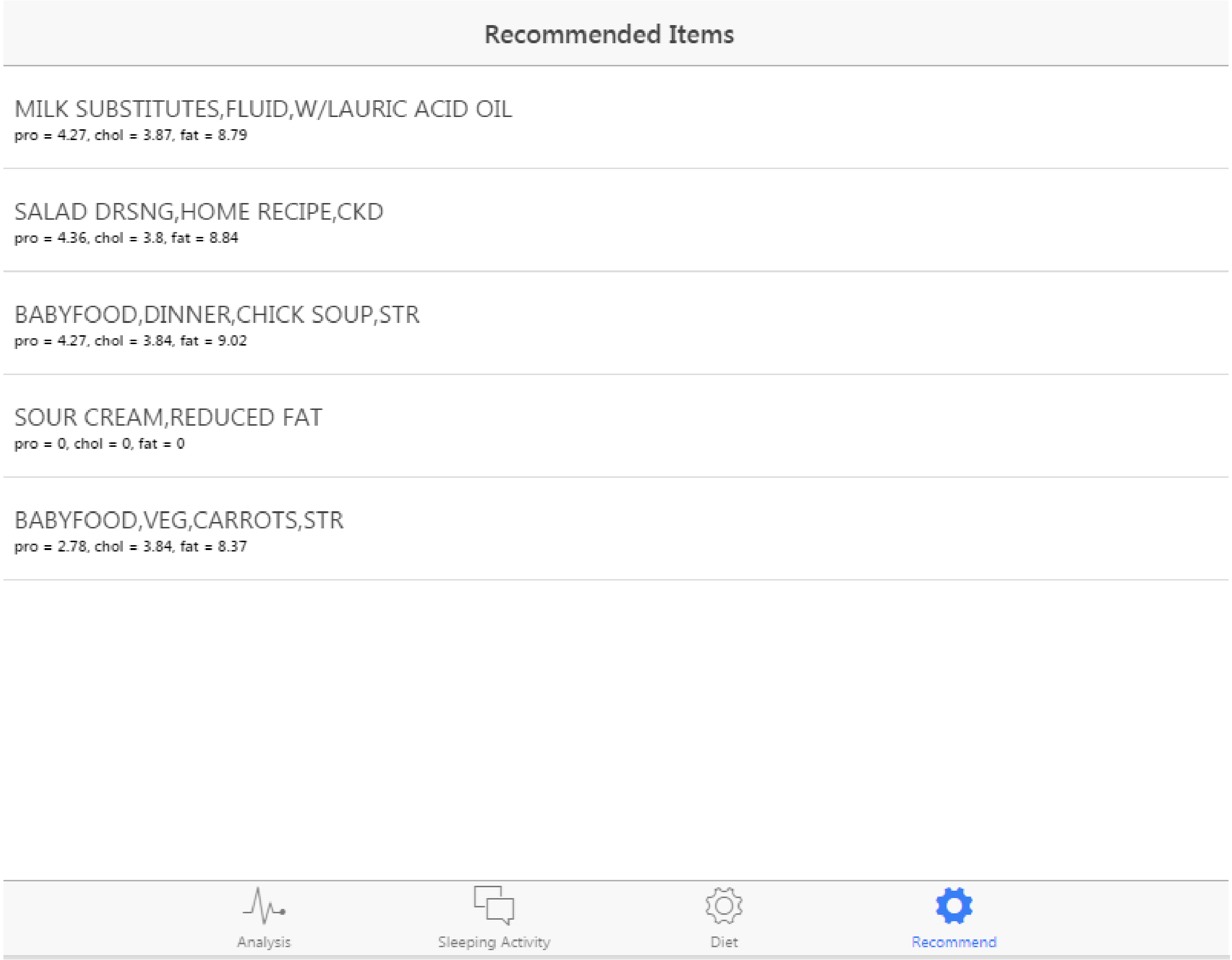


CHEESE,COTTAGE,LOWFAT,2% MILKF...
pro = 4.27, chol = 3.87, fat = 8.79

CHEESE,COTTAGE,LOWFAT,1% MILKF...
pro = 4.27, chol = 3.87, fat = 8.79

CHEESE,MOZZARELLA,WHL MILK
pro = 4.27, chol = 3.87, fat = 8.79

CHEESE,MOZZARELLA,WHL MILK,LO ...
pro = 4.27, chol = 3.87, fat = 8.79



Recommended Items

MILK SUBSTITUTES,FLUID,W/LAURIC ACID OIL
pro = 4.27, chol = 3.87, fat = 8.79

SALAD DRSNG,HOME RECIPE,CKD
pro = 4.36, chol = 3.8, fat = 8.84

BABYFOOD,DINNER,CHICK SOUP,STR
pro = 4.27, chol = 3.84, fat = 9.02

SOUR CREAM,REDUCED FAT
pro = 0, chol = 0, fat = 0

BABYFOOD,VEG,CARROTS,STR
pro = 2.78, chol = 3.84, fat = 8.37

MACHINE LEARNING ALGORITHMS:

We used 2 machine learning algorithms

1. Apriori Algorithm:

The Apriori algorithm is an algorithm for mining frequent datasets. Since the heart rate of the user when collected it produces similar data and frequent datasets are formed. This enables the use of this algorithm to determine the patterns.

The FP-growth algorithm is described in the paper Han et al., Mining frequent patterns without candidate generation, where "FP" stands for frequent pattern. Given a dataset of transactions, the first step of FP-growth is to calculate item frequencies and identify frequent items. Different from Apriori-like algorithms designed for the same purpose, the second step of FP-growth uses a suffix tree (FP-tree) structure to encode transactions without generating candidate sets explicitly, which are usually expensive to generate. After the second step, the frequent itemsets can be extracted from the FP-tree. Inspark.mllib, its been implemented a parallel version of FP-growth called PFP, as described in Li et al., PFP: Parallel FP-growth for query recommendation. PFP distributes the work of growing FP-trees based on the suffices of transactions, and hence more scalable than a single-machine implementation. We refer users to the papers for more details.

spark.mllib's FP-growth implementation takes the following (hyper-)parameters:

- minSupport: the minimum support for an itemset to be identified as frequent. For example, if an item appears 3 out of 5 transactions, it has a support of 3/5=0.6.
- numPartitions: the number of partitions used to distribute the work.

In our project folder, more information is available about this algorithm which helped us to implement this.

2. Food recommendation according to nutrients:

For recommendation use case we are planning to recommend user food items as per nutrition he is following or meal plan he is taking.
We have collected data from USDA national nutrient data and made it in required format after some clean up. We dumped it to mysql as structured format. We have over 9,000 food items and 25 different nutrients.
We are using cosine similarity function for finding the similarity between users' selected or consumed item and the other items available to the database. We compare individual item with

other item to make pair of two rows. We calculate cosine similarity between these two rows and store result as (key, pair) value where key is id of two food items and value is cosine similarity which lies between 0 to 1. Here, we don’t have any negative values so it lies between 0 to 1 else it would be between -1 to 1.

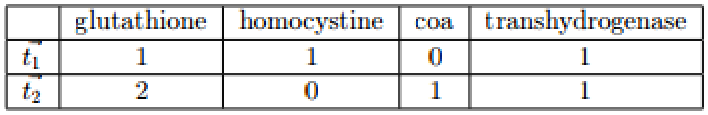
Below is the simple example of cosine similarity.

$$cos(\vec{t_1},\vec{t_2}) = \frac{\vec{t_1} \cdot \vec{t_2}}{\|\vec{t_1}\| \|\vec{t_2}\|}$$

To calculate cosine similarity between two texts t1 and t2, they are transformed in vectors as shown in the Table 1.

For example, a cosine similarity can be computed as below for two texts

$$\frac{1 \cdot 2 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1}{\sqrt{1^2 + 1^2 + 0^2 + 1^2}\sqrt{2^2 + 0^2 + 1^2 + 1^2}} \simeq 0.72$$



	glutathione	homocystine	coa	transhydrogenase
$\vec{t_1}$	1	1	0	1
$\vec{t_2}$	2	0	1	1

We ran this using apache spark core and spark SQL. We used JDBC connecter for java to store result back to the mysql database. Below is the snap shot for one item with its most similar 5 items.

DATA:

Stored data in Spark SQL and accessed them as RDD’s. The structure was simple to understand, and easy to consume data from it, so we preferred this.

We have continuous real time data where we collected heart rate, accelerometer, and the information of nutrients was already available.

In android studio, we formatted the data in the way we require it so we can directly process it on Spark.

EVALUATION:

This application was evaluated and tested several times among the members of the team and it worked perfectly fine as expected. We tested with extreme values as well and it showed perfect results.

ACCURACY:

The values of heart rate and accelerometer were so accurate and we can see the real time values on the screen and we had a meta data field in the heart rate which tells us how accurate the values are.

More over, our testing of recommendation system with different lambda values helped in understanding the nutritional accuracy of the system.

At times, the sensor used to behave exceptionally that it showed zeros even when the person is wearing, in such situations we eliminated that case where it showed zeros to get good results in patterns.

RUNTIME PERFORMANCE:

We tested the HeartFit application several times among the members of the team and it worked perfectly fine as expected. We tested with extreme values as well and it showed perfect results.

Point to Note: We tested the application by wearing the application while sleeping to observe the sleeping pattern and the application worked well. It tracked the movements correctly and plotted the graph accurately.

While running the application, you have the option to go a previous date and view the heart and sleeping patterns of it. This gives us to keep a track of our health history.

RELATED WORK:

1. We referred several documents to find what patterns could be useful for a use and also as TA's suggested we went through ARGUS application which gave us an idea of sleeping pattern. A special thank you for their support as we were able to improve in every increment.

   Mentioning few of the links that we referred:

ftp://ftp.tik.ee.ethz.ch/pub/students/2010-HS/SA-2010-26.pdf

http://www.ceng.metu.edu.tr/~e1395557/foodRecSys.pdf

https://bioinformatics.oxfordjournals.org/content/suppl/2009/10/24/btp613.DC1/bioinf-2008-1835-File004.pdf

http://blog.echen.me/2012/02/09/movie-recommendations-and-more-via-mapreduce-and-scalding

https://github.com/pocmo/SensorDashboard

FUTURE WORK:

We implemented the features that we wanted to implement, but in future we want to alert the user with some emergency if some extreme heart patterns are observed.
This could help in the betterment of the project.

We referred to several documents to find out what patterns could be useful for a user and also as TA’s suggested we went through ARGUS application which gave us an idea of sleeping pattern. A special thank you for their support because of whom were able to improve in every increment.