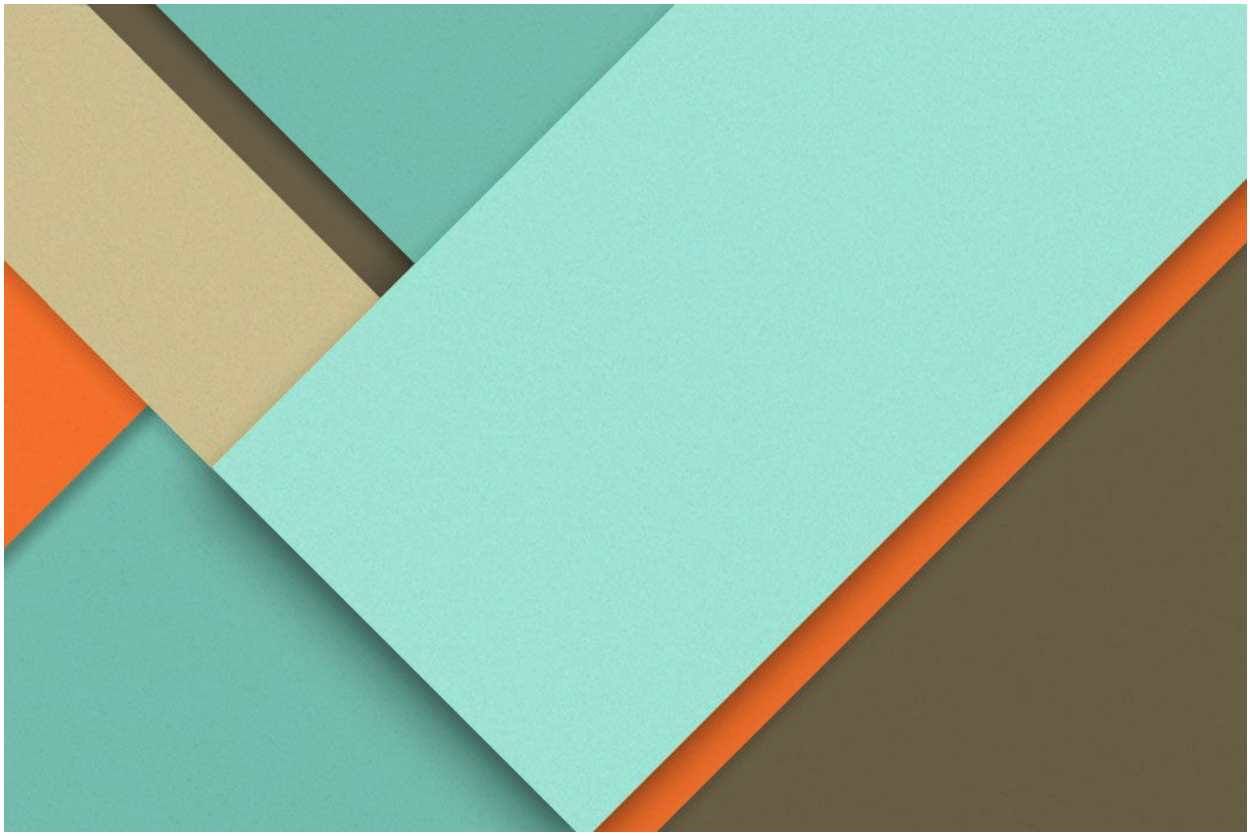


CS 5542 Project Increment 3

Gharibi, Mohamed (7)

Huang, Wei (9)

Sattineni, Vipin Reddy (24)

Xia, Ting (30)

Introduction

Nowadays, the technology is getting improved day after day and hour after hour. From this point, all the people start looking for new ideas and projects which help to simplify the daily issues for the people. Moreover, as we know the data in different types through different social media websites and applications are getting improved and increased every single second as well. Especially, when we talk about large amount of data and how to use this data in a sufficient way to serve us in a better way.

There are many majors and many different ways to get use of this data so that it would be sufficient for all the people in an easy, fast and simple way. For example, having an application for storing the patient's' record is one of the best ideas, so that whenever something happens to this patient, all what the doctors have to do is checking in his last records about his health in the previous days and his previous illnesses. In this way, we used the data which we collected for all the patients in a better way and display it in a good application that is easy to be read.

In our project, we will do almost the same. We will use the technology (such as the smartphone or smartwatch) to get help and serve us in a better way in our daily life. Our project is about reading and monitoring some people's health parameters and make necessary assistance if necessary. Also, it can make life easy by recognizing and fetching the objects we want through controlling of smartwatch or smartphone. The smartphone and smartwatch will be paired, so the communication between the two can be easy and quite useful. We will do these through using big data tools such as spark which is a very good tool to deal with such data and analyse it.

In the next pages, we will discuss about the class and sequence diagrams for the project and we will explain the objectives of this project, APIs were used and the other tools and programming languages.

Project Goal and Objectives

- Overall goal(revised)

  The main purpose of this project is to provide realistic assistance to adult,especially for seniors in 24/7 based. Such personal aid functions has become more critical in modern lives. Specifically, our “Baymax Duo” will provide very personalized health monitoring functions, which include body health monitoring, environmental safety and indoor objects identification functions such that to improve user’s life quality and safety.

- Specific objectives(problem statement)

  Since we are focusing on application development for adults (especially for seniors). There are always issues related to unpredictable events, such as sudden illness, life threats that come from others. It is necessary to address those issues and develop certain applications to provide convenient assistant and reduce those potential threat in an urgent, precautious and effective manners.

- Specific features

  By using health monitoring functions, if a user is experiencing potential health threat (such as heart attack, stroke, fall down, lost conscious etc), by matching real time data reading with the database to make assistance if necessary to increase patients’ surviving probabilities. At the same time, it can send out emergency signals to the user’s relatives, local police stations, hospitals etc. whenever it is necessary.

  By utilizing image recognition function, the smartphone can capture human face characters and compare it with specific database to identify if this “unknown” person is a criminal or just his/her friends and send the analyzed results to the user.

  Since there are some issues exist in Romo me and movement functions for the robot, we will cut the Indoor objects identification implement in our project. And we will put more time on developing other functional areas such as machine learning for the heart rate monitoring and face recognitions implementations by utilize machine learning concept.

  The smartwatch and smartphone are paired. In that case, the communication between the two devices could be easy. Notifications can be sent in two-way. The user can control the smartphone by just controlling the smartwatch, which would make life easy.

- Significance

  With the development of modern civilization, people concern health, safety, and convenience much more than before. How to enjoy a life with high quality becomes a valuable question considered by commercial market. We have four concerns according to the motivation. First, due to unawareness, some people suffer severely from sudden heart attack, which may even threaten the lives. In this case, Baymax Duo we propose could provide necessary assistance, such as bring medicine to these people and send text message to relevant people or hospital to save the lives. Second, criminal recognition is of great significance to protect home safety. Baymax Duo provides such function to identify criminals and friends to reinforce home safety. In that way, potential threaten to home could be eliminated as much as possible. Third, with the help of Baymax Duo, some easy work, such as fetching objects could be done by just making arm gestures. This will not only realize a more comfortable life, but also help improve the life quality of elderly people or disabled people. Also, the smartwatch and smartphone are paired. The two devices can communicate with each other easily. This will make life easy by taking advantages of the two and controlling one device through another one. Based on the three major functions, our Baymax Duo has the practical value and infinite market potentials without any doubts.

Project Plan (revised)

1) Schedule for the four different increments

A. Increment 1

List of work:

a. Project design, including scenario, use case specification, and feature design.
b. Design class diagram
c. Design sequence diagram
d. Design architecture diagram
e. Manage sensors in both smartwatch and phone.
f. Implement AlchemyAPI in phone app.

Scenario & use case specification & feature design & feature implementation:

Use case 1: send phone sensor information to smartwatch

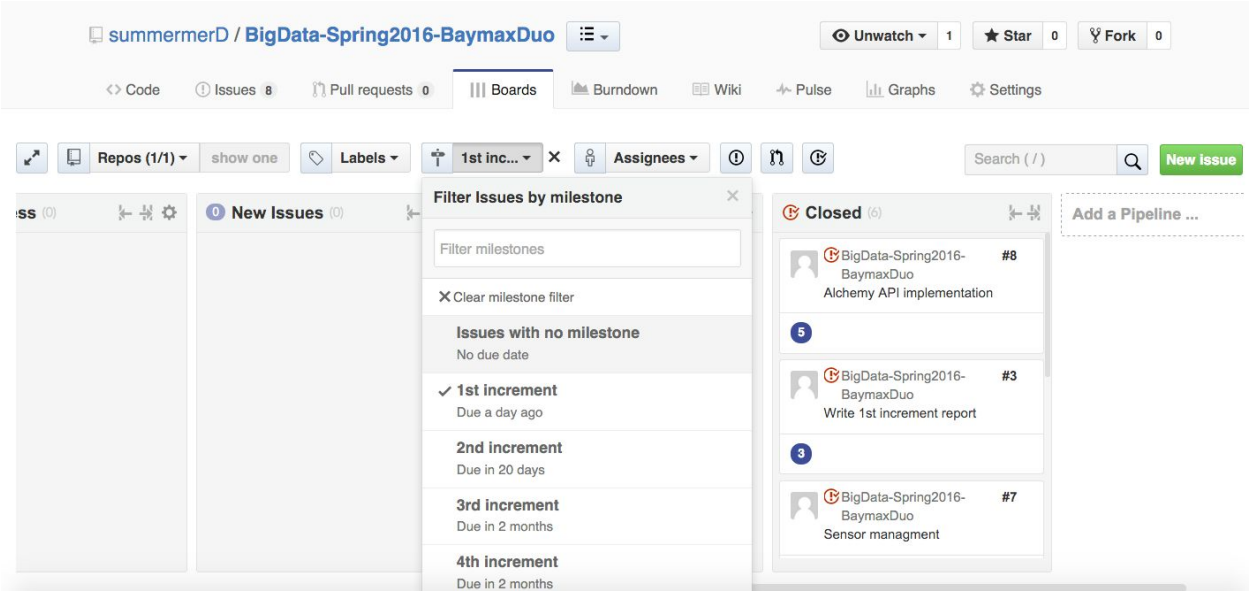
The phone has many sensors, such as accelerometer, light sensor, magnetic sensor, etc. The detected results can be sent to smartwatch by clicking the button on the phone. The user who wears the smartwatch can get the information.

We implement it by combining the sensor code with the notification code.

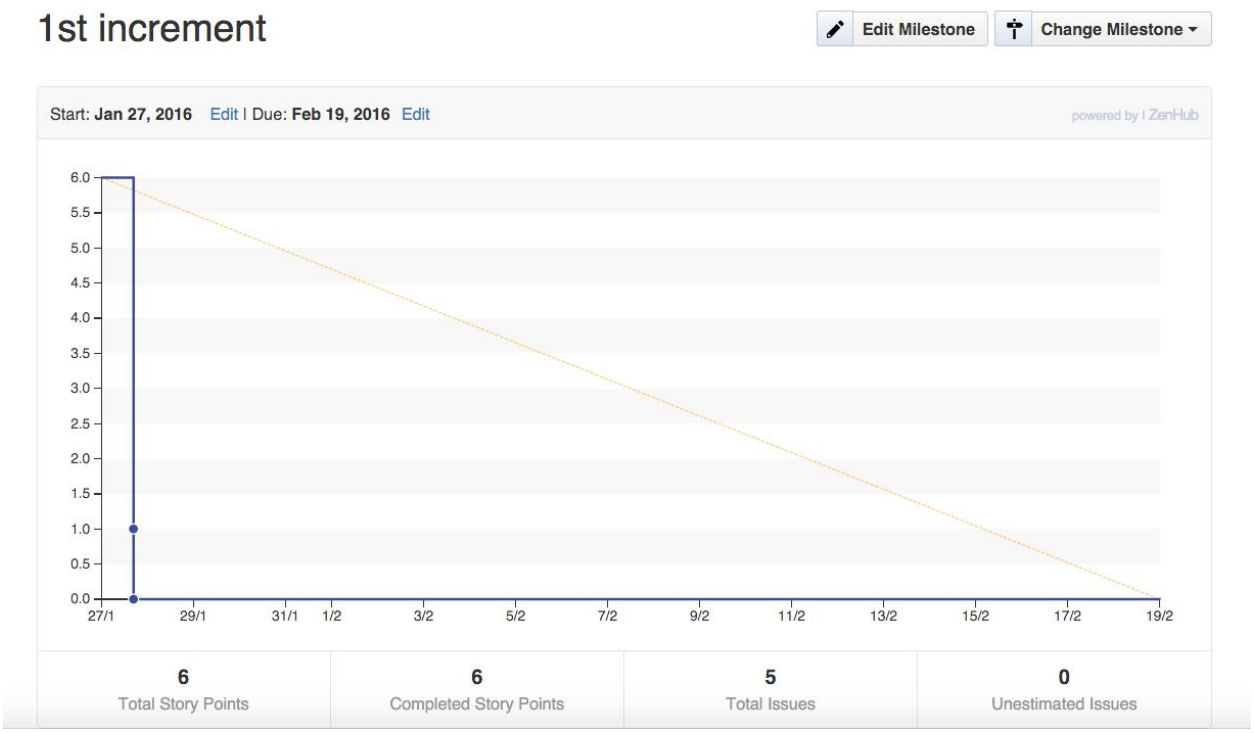
Use case 2: analyze image information by Alchemy API

We implement Alchemy API in android phone by using android Alchemy SDK. Alchemy API can provide image recognition as well as text analysis. So when a user chooses one image and clicks the button, the Alchemy API will analyze the image and give the category of the image it detects.

Boards:



Burndown chart:



B. Increment 2

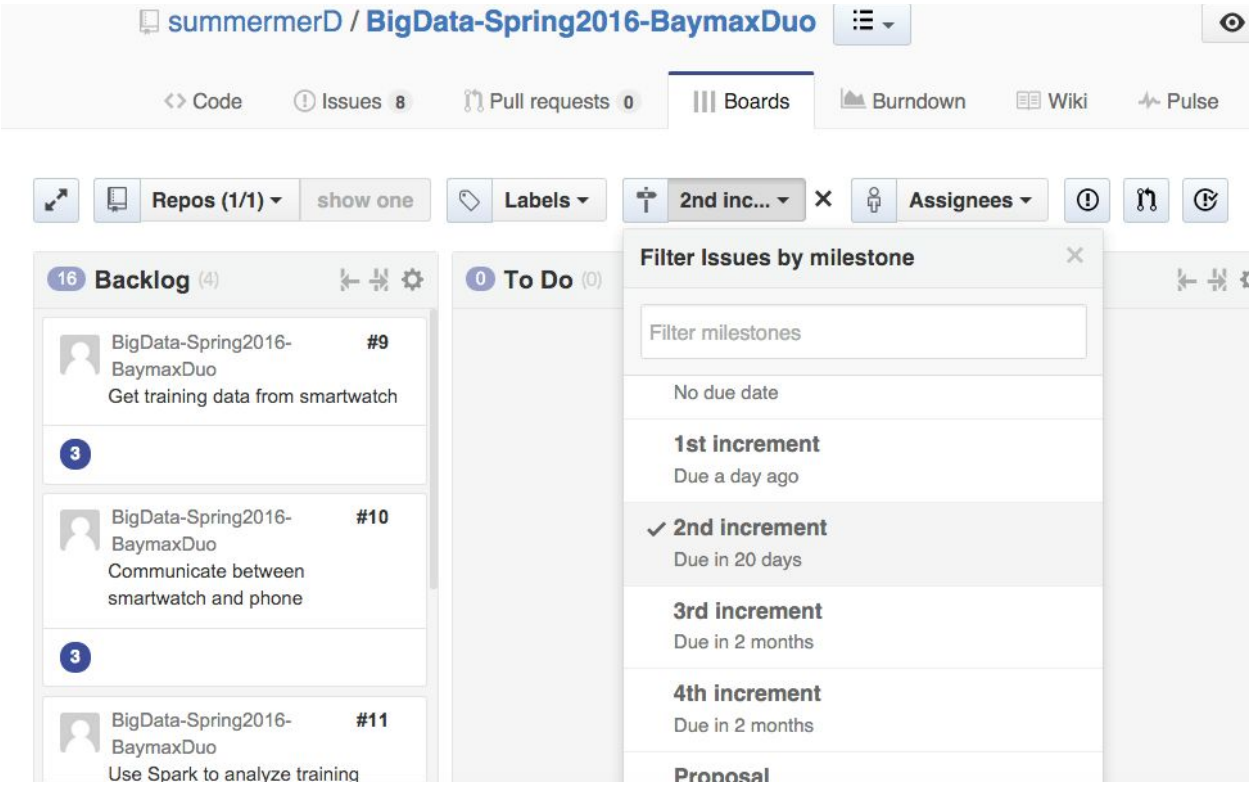
List of work:

a. Get training data from smartwatch
b. Communicate between smartwatch and phone
c. Use Spark to analyze training data sets obtained from smartwatch and phone.

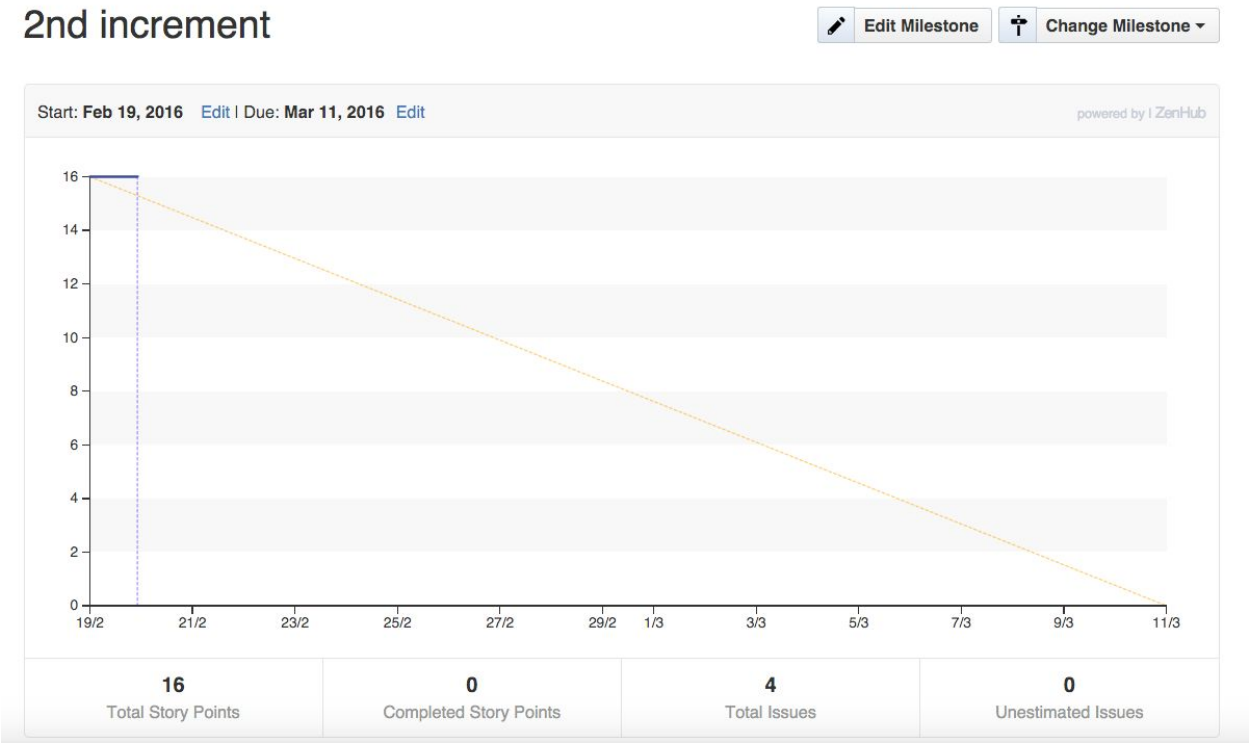
Scenario & use case specification & feature design & feature implementation:

Use case 1: We will measure the heart rate of the user and get lots of training data. Then figure out the normal range of the user. The training data can be sent to phone and analyzed by Spark.

Boards:



Burndown chart:



C. Increment 3

List of work:

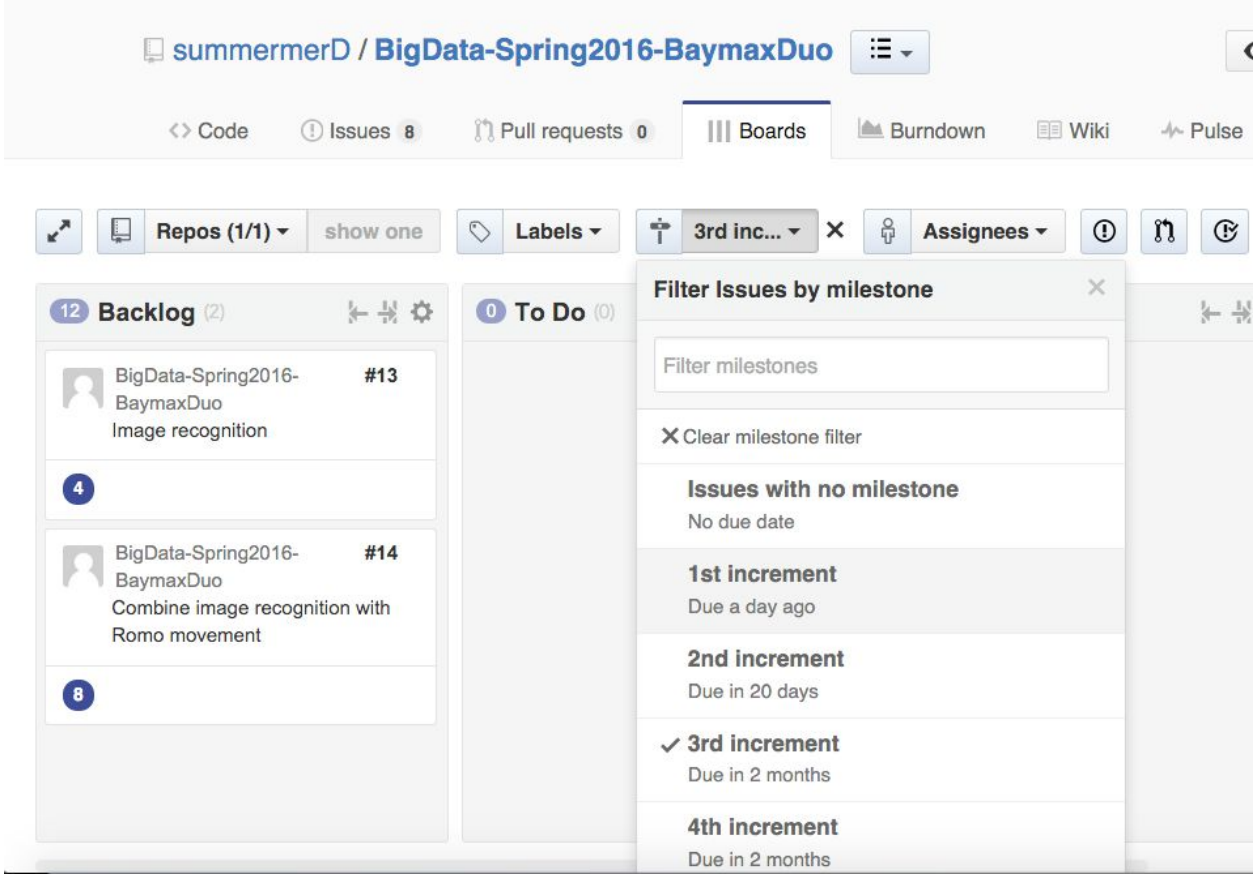
a. Image/objects recognition.
b. Combine image recognition with Romo movement.

Scenario & use case specification & feature design & feature implementation:

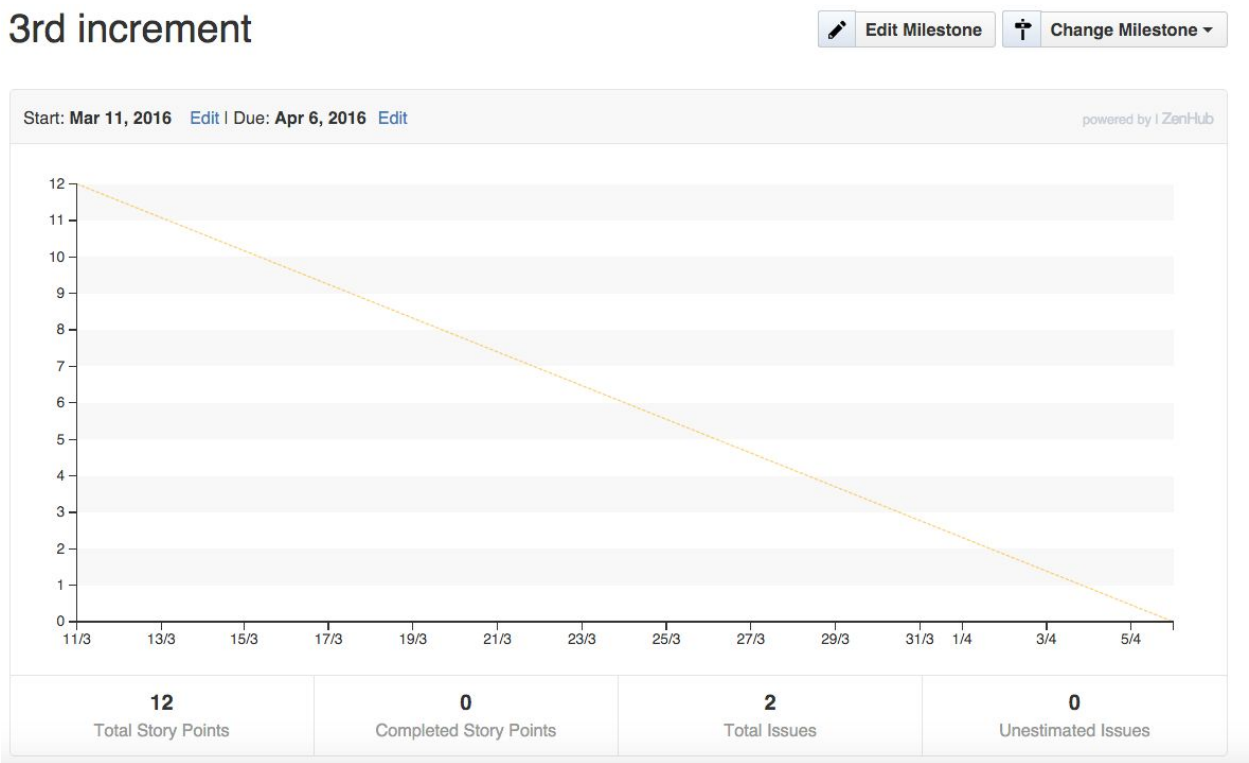
Use case 1: The Romo can take photos of objects when required. Then use image recognition API (e.g. Alchemy API) to recognize the object. The result can be sent to smartwatch/phone.

Use case 2: The user can send a command to let the Romo to fetch something to him/her. By making gestures the Romo can arrive at the object and take phone. If the object matches with the command, the Romo can get it and return back to the user.

Boards:



Burndown chart:



D. Increment 4

List of work

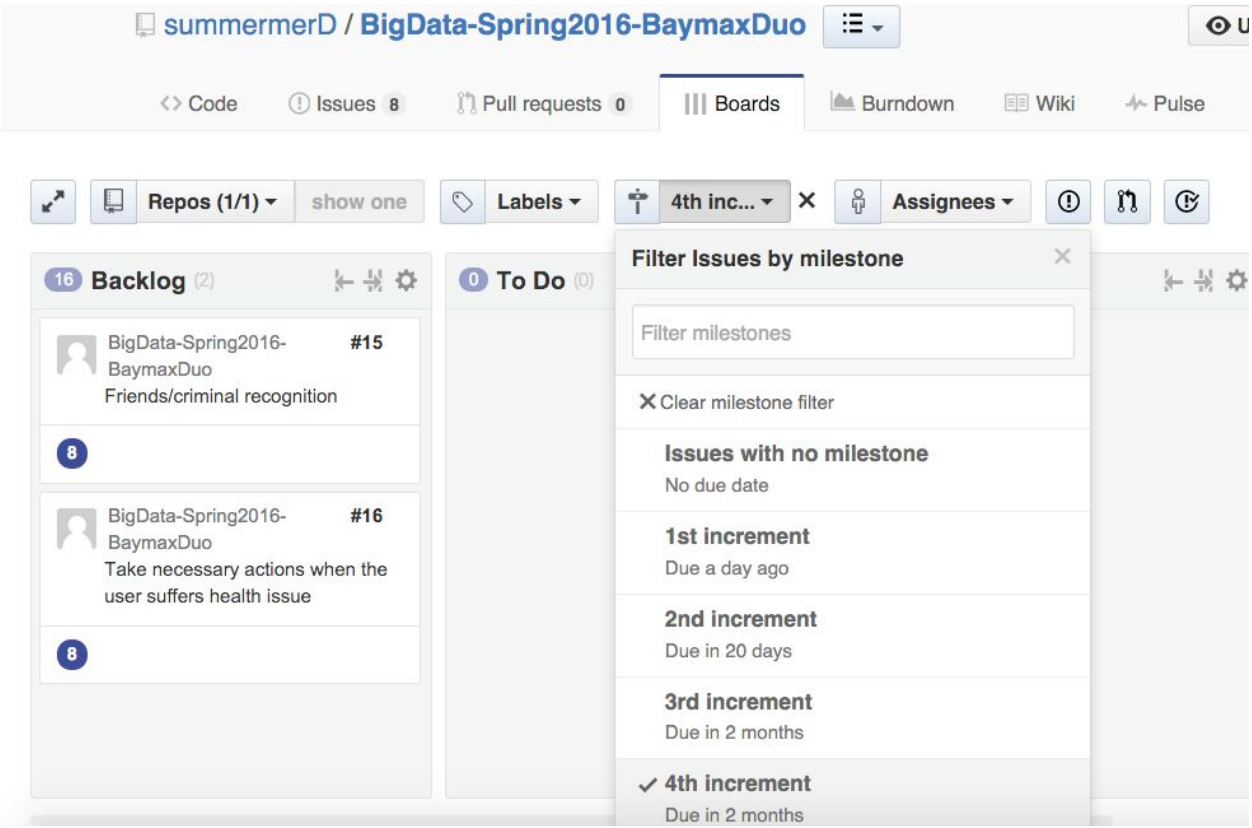
a. Friends/criminal recognition.
b. Take necessary actions when the user suffers health issue.

Scenario & use case specification & feature design & feature implementation:

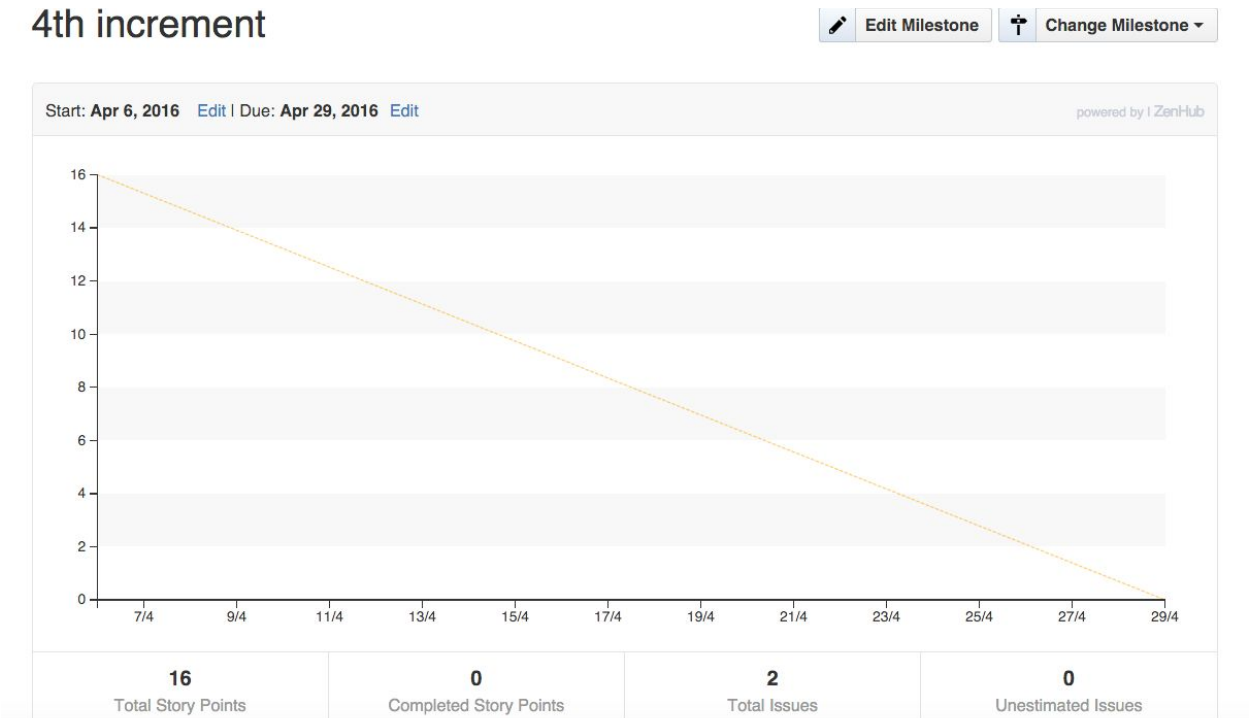
Use case 1: When someone knocks the door, the Romo can take photo of him/her and send the photo to phone and the phone will analyze it. If he/she is a criminal, the results will be sent to the user and let the user make the decision (e.g. call police). If he/she is a friend, the result will be sent to user and let the user make the decision (e.g. open the door). We will use some APIs to analyze friend/criminal.

Use case 2: When the user who wears the smartwatch has the health issue (heart rate abnormal), the smartwatch will send message the related people and let the Romo come to the user (the Romo has the necessary medicine with it).

Boards:



Burndown chart:



2) Project Timelines, Members, Task Responsibility

In general, each of our team members will be have a very clear separation regarding to each section of the tasks. For instance, Mohamed will be responsible for testing all the exist applications that we have developed and report potential issues to us. Vipin will be responsible for modifying the existing issues and also looking for opportunities to improve the functionalities of all apps. Ting and Wei will be constantly working on developing new apps and considering all sorts of possibilities that how the new apps could fit into our project and how those functions might interact with others. By doing in this, we could form a basic project development cycle and guaranty that the smoothness of all the functions.

In terms of timelines, we are considering launch the basic functions such as sensors, AlchemyAPI on phone apps. Before the increment 3 due time, we already have a very clear picture of how many actual data we can stored and processed by using either Spark, Hadoop, R etc. and categorize those trained data to implement the potential situations and apply to the possible test data set and see how accurate they could be. Before increment 3, we are planning to implement face recognition features in project. By the time of increment 3 due, this feature should be relatively mature and it can perform the corresponding task in an acceptable level. By increment 4, we should have all the tuned functions. Lastly, we will compare human face image with criminal database to check if this person will be a match with anybody.

Second Increment Report

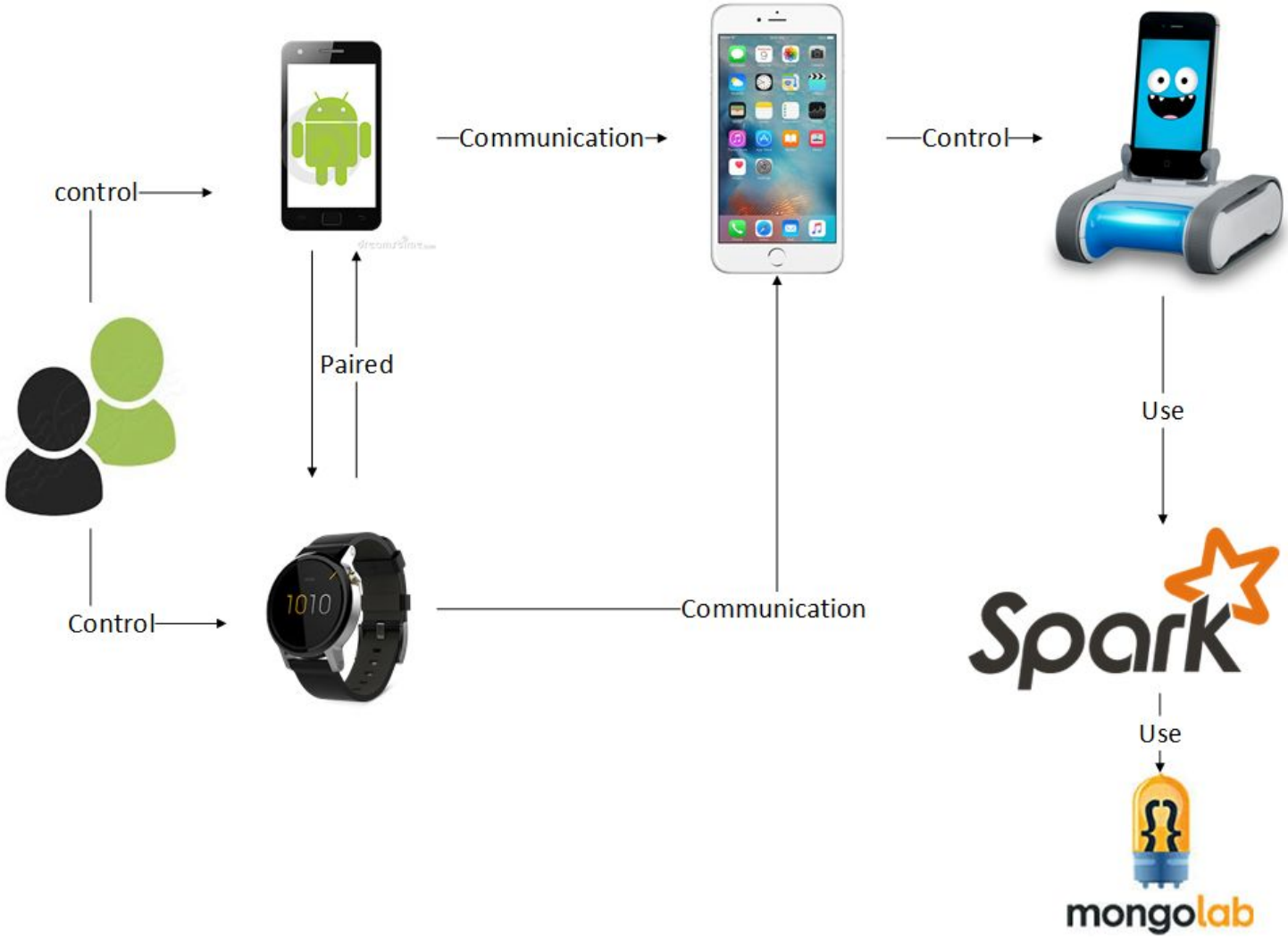
All the features we have developed for this increment, write a documentation describing the design, implementation, testing, and deployment (including the precise descriptions and screenshots).

1. Existing services, widget, API or source code you used in your project.
    a. Basic environment sensors such as light, proximity, magnetic sensors.
    b. Alchemy API
    c. Spark MLlib
    d. Twitter streaming
    e. Stanford Core NLP
    f. Alternating Least Squares (ALS)
2. Design of Features
    A. Architecture diagram/Class diagram/Sequence diagram

    Architecture diagram:

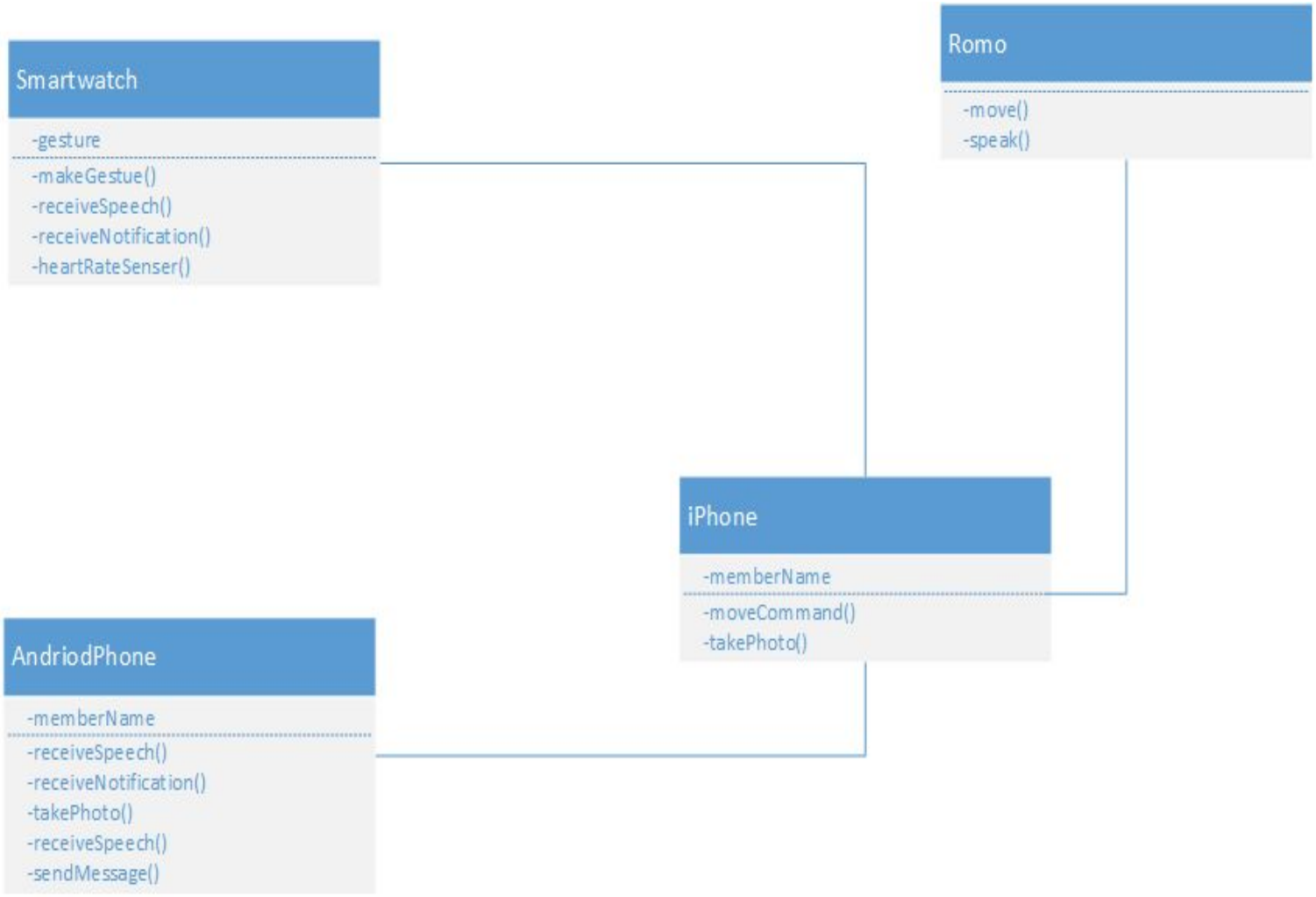
It contains three devices: the android smartwatch, android phone, iPhone. The user can control the android phone and smartwatch to communicate with the iPhone.

Also, iPhone can take photo when required. The photo will be analyzed and the results will be sent to the user. Spark is used to analyzed the data sets and Mongolab is used to store the data. The smartwatch and phone are paired, so the communication is easy between the two.



Class diagram:

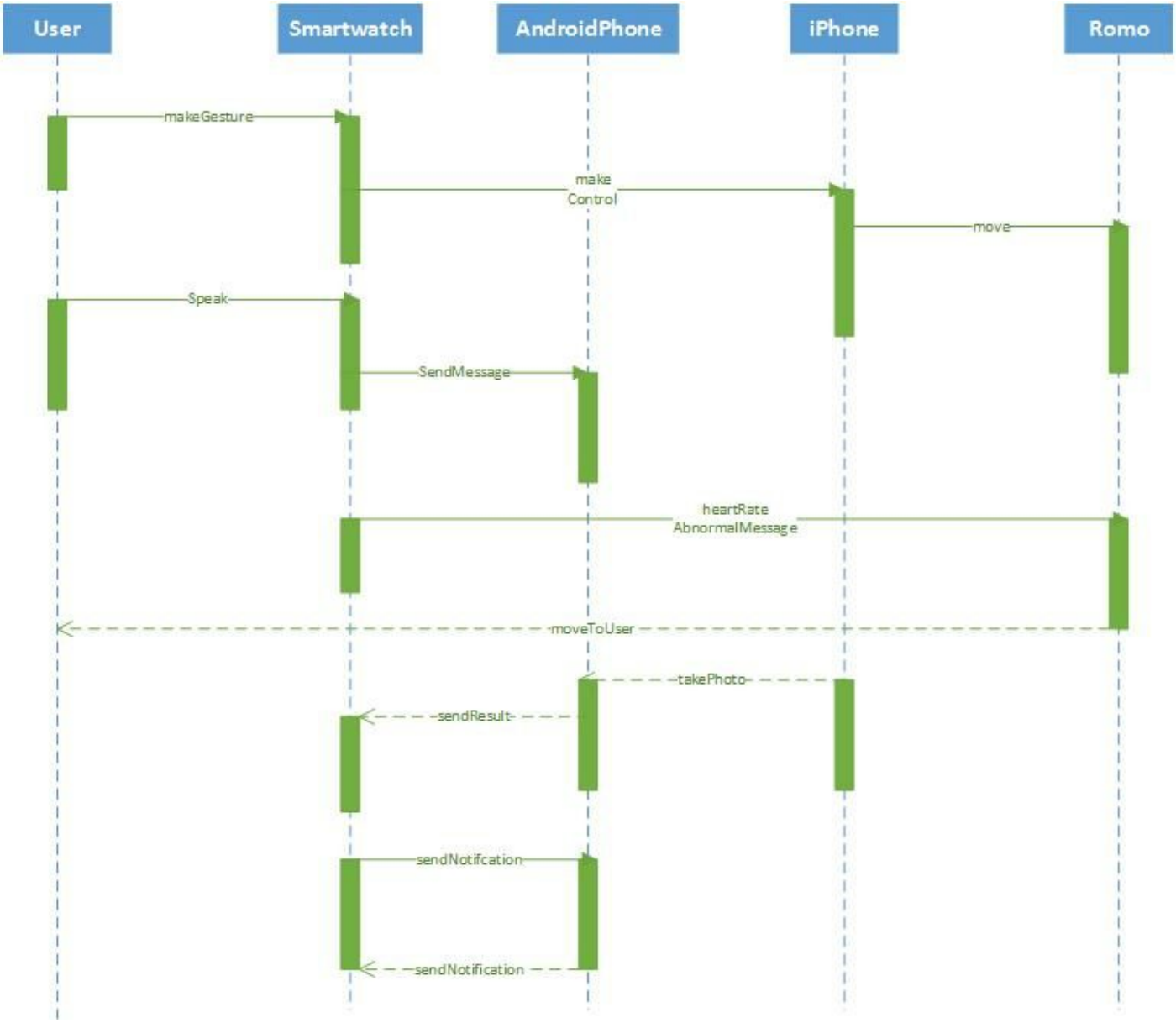
Three classes are listed as follows: smartwatch, android phone, iPhone. The methods are included in the diagram.



Sequence diagram:

Three scenarios:

1. Speech control
2. Heart rate detection and action
3. Communication between smartwatch and android phone.



B. Hadoop/Machine Learning Algorithms

In Hadoop/Machine learning, we are planning to use it as one of our core techniques to processing the data. In the later on project increment, it will behaved as the part that "force" the robot to react upon certain data we have collected or we have trained to the data. For example, in SparkR, we can use it to process certain amount of data to calculate K-Means, K centroid and Expectation Maximization(EM) and use it as a best representation of certain amount of data which fall under a defined category. Consequently, we can shrink the static data size into smaller ones so that it is easier to process. Specifically, the following services are used in our project.

a. **Use Spark MLlib**:
   i. k-means analysis of heart rate.
   ii. Categorize collected tweets using Feature Extraction – TF-IDF.
b. **Sentiment analysis:** Stanford Core NLP is used to do sentiment analysis of collected tweets
c. **Recommendation:** Alternating Least Squares (ALS) is used to recommendation to the user.

C. Datasets
   a. Twitter streaming to collect tweets
   b. Sensor data get from smartphone/smartwatch

D. Implementation

o Mobile Client Implementation (Smartwatch, Smartphone)

We will implement the mobile Client by utilizing smartwatch and smartphone

o Machine Learning Application

- Personal health: By comparing heart rate history to determine to determine user's current health conditions. If monitoring exceeding certain limitations, instant alert notification will be sent to watch to draw user's attention. Otherwise, display as "Normal".
- Environmental safety: Base on people's face characteristic(may requires collecting large amount of face features data online), by training robot on matching captured human's face image with certain database to identify if specific person is an "enemy" or "friend".

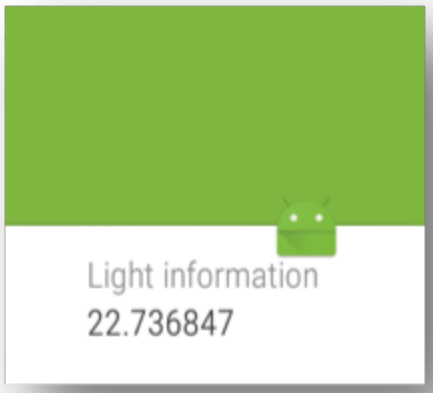
E. Deployment

o Deploy your mobile app to smartphones and describe it including the screenshots

1) Collect sensor information and send it to smartwatch

We also used the sensors in our project. Using Android studio, we combined the notification method and the sensor data collection method to send information from phone sensor to smartwatch.

Notification sent to smartwatch:

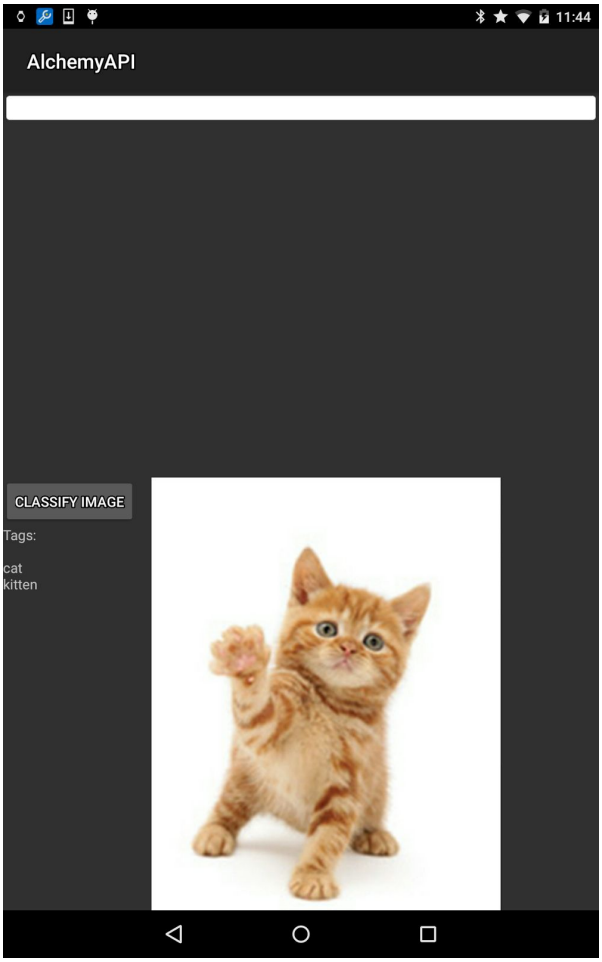

Light information
22.736847


Magnetic
information
16.799927


Accelerometer
information
-0.771286,1.40872
19,9.60051

2) Image recognition (Use Alchemy API)

After we have completed the first increment, now we could say that we have a nice page to process the pictures using Alchmey API as we can see from the picture below:



When a user chooses an image then click “classify image” Alchmey API could recognize the picture and provide the tags for this picture (i.e. it was successfully recognize the cat image).

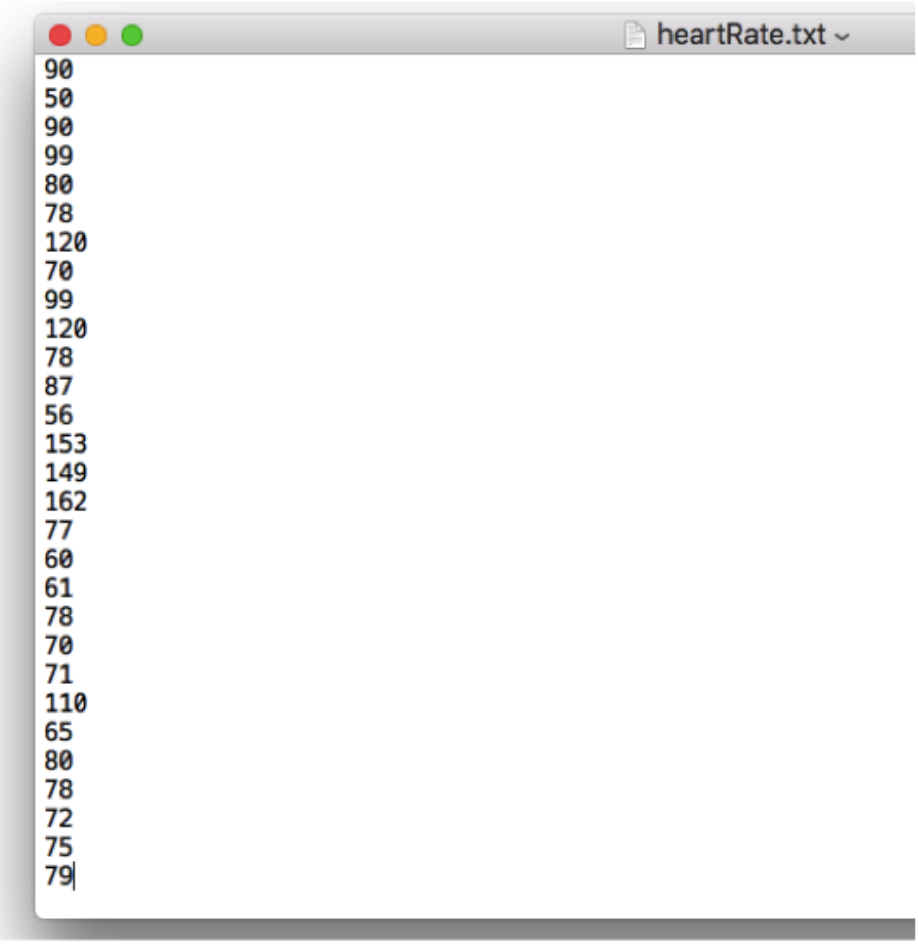
Of course, as you can see in the picture above that this picture was not deployed on the computer, it was deployed on the smartphone device. Just like that, we can implement this API on the smartwatch we have. Moreover, Alchemy API could help us in our project to classify the owner of the picture if he is a friend or not, since we are making our project as a home assistant application.

And before we keep talking about the other features, we would like to mention that all the features could be implemented on the smartwatch as well, since we already linked the smartwatch with our computer so that we can control it in a proper way. For example we already did some parts of machine learning using the smartwatch. i.e. we prepared a data set related to our project then we perform K-means and K-median over this set of data.

3) Heart rate analysis (Use Spark MLlib)

We also completed all the second increment features such as linking spark with the smartphone so that we could be able to implement a smart application with big data analytics which is related to our project. So we used K-means method in Spark MLib to analyze heart rater. As you will see below we defined five clusters: high danger, high normal, low, low danger. After Spark analyzes three clusters, the cluster center will be sent to android phone using Android socket connection.

Data input (heart rate):



Run logs:

```
16/02/24 22:28:06 INFO TaskSchedulerImpl: Removed TaskSet 21.0, whose tasks have all comple
16/02/24 22:28:06 INFO DAGScheduler: ResultStage 21 (collect at KMeansModel.scala:147) fini
16/02/24 22:28:06 INFO DAGScheduler: Job 19 finished: collect at KMeansModel.scala:147, too
Final Centers:
[59.33333333333333]
[154.66666666666666]
[116.66666666666666]
[76.0]
[93.0]
16/02/24 22:28:06 INFO SparkUI: Stopped Spark web UI at http://192.168.1.7:4040
16/02/24 22:28:06 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopp
16/02/24 22:28:06 INFO MemoryStore: MemoryStore cleared
16/02/24 22:28:06 INFO BlockManager: BlockManager stopped
16/02/24 22:28:06 INFO BlockManagerMaster: BlockManagerMaster stopped
```

Android phone received:

4) Tweets word count

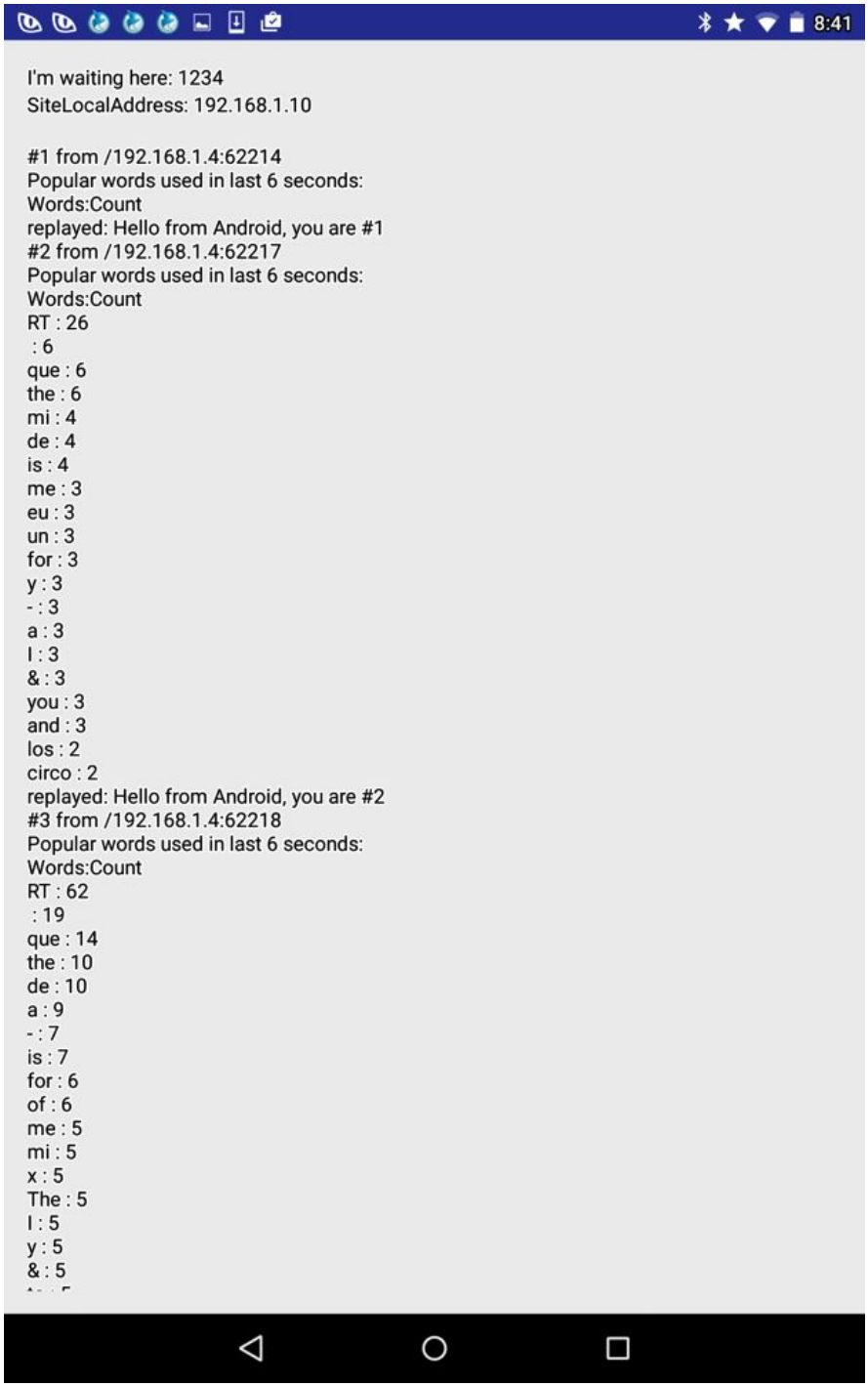
Moreover, we collected the tweets using Twitter streaming. After that, we made a map reduce over the collected data and finally we collected the results in a string and then send it to android phone.

Run log:

```
16/02/27 20:41:16 INFO DAGScheduler: Job 9 finished: count

Popular words used in last 6 seconds (415 total):
RT (26 times)
 (6 times)
que (6 times)
the (6 times)
mi (4 times)
de (4 times)
is (4 times)
me (3 times)
eu (3 times)
un (3 times)
for (3 times)
y (3 times)
– (3 times)
a (3 times)
I (3 times)
& (3 times)
you (3 times)
and (3 times)
los (2 times)
circo (2 times)
16/02/27 20:41:17 INFO MemoryStore: Block input-0-1456627
```

Sent to smartphone.



5) Category analysis of tweets collected.

```
PREDICTION
16/02/27 21:01:07 INFO MemoryStore: Block input-0-1456628467000 stored as bytes in mem
16/02/27 21:01:07 INFO BlockManagerInfo: Added input-0-1456628467000 in memory on loca
16/02/27 21:01:07 WARN BlockManager: Block input-0-1456628467000 replicated to only 0
16/02/27 21:01:07 INFO BlockGenerator: Pushed block input-0-1456628467000
16/02/27 21:01:07 INFO SparkContext: Starting job: foreach at TwitterCategoryAnalysis.
16/02/27 21:01:07 INFO DAGScheduler: Got job 31 (foreach at TwitterCategoryAnalysis.sc
16/02/27 21:01:07 INFO DAGScheduler: Final stage: ResultStage 30 (foreach at TwitterCa
16/02/27 21:01:07 INFO DAGScheduler: Parents of final stage: List()
16/02/27 21:01:07 INFO DAGScheduler: Missing parents: List()
16/02/27 21:01:07 INFO DAGScheduler: Submitting ResultStage 30 (MapPartitionsRDD[74] a
16/02/27 21:01:07 INFO MemoryStore: Block broadcast_36 stored as values in memory (est
16/02/27 21:01:07 INFO MemoryStore: Block broadcast_36_piece0 stored as bytes in memor
16/02/27 21:01:07 INFO BlockManagerInfo: Added broadcast_36_piece0 in memory on localh
16/02/27 21:01:07 INFO SparkContext: Created broadcast 36 from broadcast at DAGSchedul
16/02/27 21:01:07 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 30 (M
16/02/27 21:01:07 INFO TaskSchedulerImpl: Adding task set 30.0 with 2 tasks
16/02/27 21:01:07 INFO TaskSetManager: Starting task 0.0 in stage 30.0 (TID 131, local
16/02/27 21:01:07 INFO TaskSetManager: Starting task 1.0 in stage 30.0 (TID 132, local
16/02/27 21:01:07 INFO Executor: Running task 0.0 in stage 30.0 (TID 131)
16/02/27 21:01:07 INFO Executor: Running task 1.0 in stage 30.0 (TID 132)
16/02/27 21:01:07 INFO BlockManager: Found block rdd_65_1 locally
16/02/27 21:01:07 INFO BlockManager: Found block rdd_65_1 locally
16/02/27 21:01:07 INFO BlockManager: Found block rdd_65_0 locally
16/02/27 21:01:07 INFO BlockManager: Found block rdd_65_0 locally
16/02/27 21:01:07 INFO JniLoader: successfully loaded /var/folders/_n/x33gb1l54b132v0p
16/02/27 21:01:07 INFO Executor: Finished task 1.0 in stage 30.0 (TID 132). 2044 bytes
16/02/27 21:01:07 INFO Executor: Finished task 0.0 in stage 30.0 (TID 131). 2044 bytes
16/02/27 21:01:07 INFO TaskSetManager: Finished task 1.0 in stage 30.0 (TID 132) in 82
16/02/27 21:01:07 INFO TaskSetManager: Finished task 0.0 in stage 30.0 (TID 131) in 84
16/02/27 21:01:07 INFO TaskSchedulerImpl: Removed TaskSet 30.0, whose tasks have all c
16/02/27 21:01:07 INFO DAGScheduler: ResultStage 30 (foreach at TwitterCategoryAnalysi
16/02/27 21:01:07 INFO DAGScheduler: Job 31 finished: foreach at TwitterCategoryAnalys
hashtag.food
hashtag.other
hashtag.food
hashtag.other
hashtag.food
hashtag.food
hashtag.food
hashtag.food
16/02/27 21:01:07 INFO StreamingContext: Invoking stop(stopGracefully=false) from shut
```

6) Recommendation to user

We analyzed the tweets and made a recommendations, we made a sentiment analysis as well. Now, we are able to push all the results to the android phone.

Step 1: Define 10 categories:

Step 2: Collect tweets as training data to categorize the tweets into these 10 categories using keywords searching.

Step 3: Collect tweets again for recommendation training. In order to do rating, four items should be collected:

1) UserId. It's the tweet's userId which should be converted into integer.
2) Category. It is analyzed using feature extraction – TF-IDF by the training data collected in step 2.
3) Rating. Use sentiment analysis to give the rating for each tweet.

4) Timestamp. Tweet creation time which should be converted into integer.

These four items should be written into one file called “rating.txt”.

Step 4: Get the category mapping file called “category.txt”.

Step 5: Get the recommendation for one particular user. The recommendation is the categories that the user prefers.

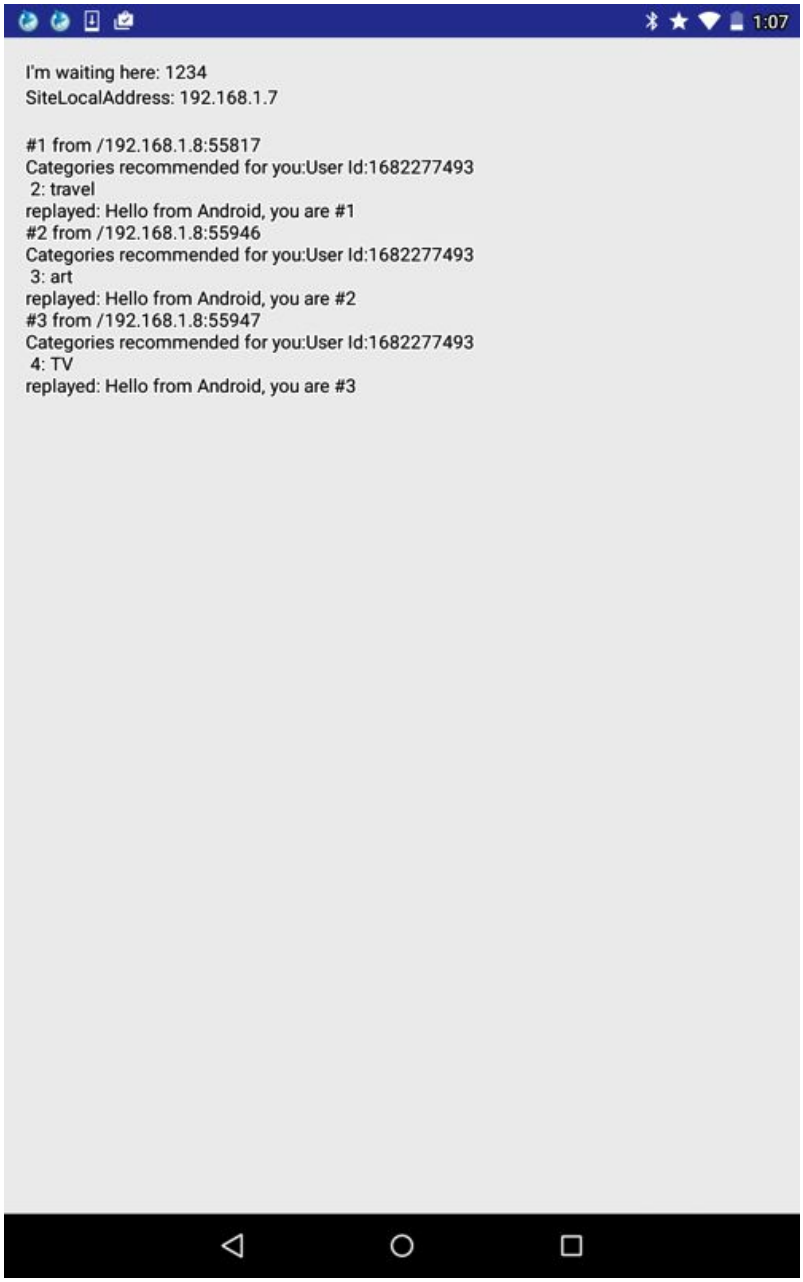
Step 6: Send the results to the smartphone/smartwatch

Run log:

```
RMSE (validation) = 0.16005871649500103 for the model trained with rank = 8, lambda = 0.1, and numIter = 10.
RMSE (validation) = 0.18066115598146681 for the model trained with rank = 8, lambda = 0.1, and numIter = 20.
RMSE (validation) = 3.692744729379982 for the model trained with rank = 8, lambda = 10.0, and numIter = 10.
RMSE (validation) = 3.692744729379982 for the model trained with rank = 8, lambda = 10.0, and numIter = 20.
RMSE (validation) = 0.15739262495617246 for the model trained with rank = 12, lambda = 0.1, and numIter = 10.
RMSE (validation) = 0.180025657757478 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
RMSE (validation) = 3.692744729379982 for the model trained with rank = 12, lambda = 10.0, and numIter = 10.
RMSE (validation) = 3.692744729379982 for the model trained with rank = 12, lambda = 10.0, and numIter = 20.
The best model was trained with rank = 12 and lambda = 0.1, and numIter = 10, and its RMSE on the test set is 1.8075950816211317.
The best model improves the baseline by -61.31%.
Categories recommended for you:
 1: art
 2: travel
 3: TV

Process finished with exit code 0
```

Sent to smartphone



7) Image recommendation

In our application we provided many services that could help a user a lot to improve his search and recommendation part. For example, some people like animals a lot so in this case we recommend animals pages and pictures for a user depends on the animal species he or she likes. Others, want to teach their children about animals and differences between them, so that we also implemented a service which allows you to recognize the animal species from many different categories. First of all, we implemented the text sentiment analysis which allows us to know what the user likes and hates, so that if the user typed something good about a cat, we know this user likes cats by using sentiment analysis.

In addition, through our application you will be able to choose an animal picture and process it using Spark. Our application will process the picture using Spark and will let you know weather this animal is a cat or dog, etc. We have ten animal categories to recognize which animal you like. Then, we will recommend you pages and pictures for the same animal you like.

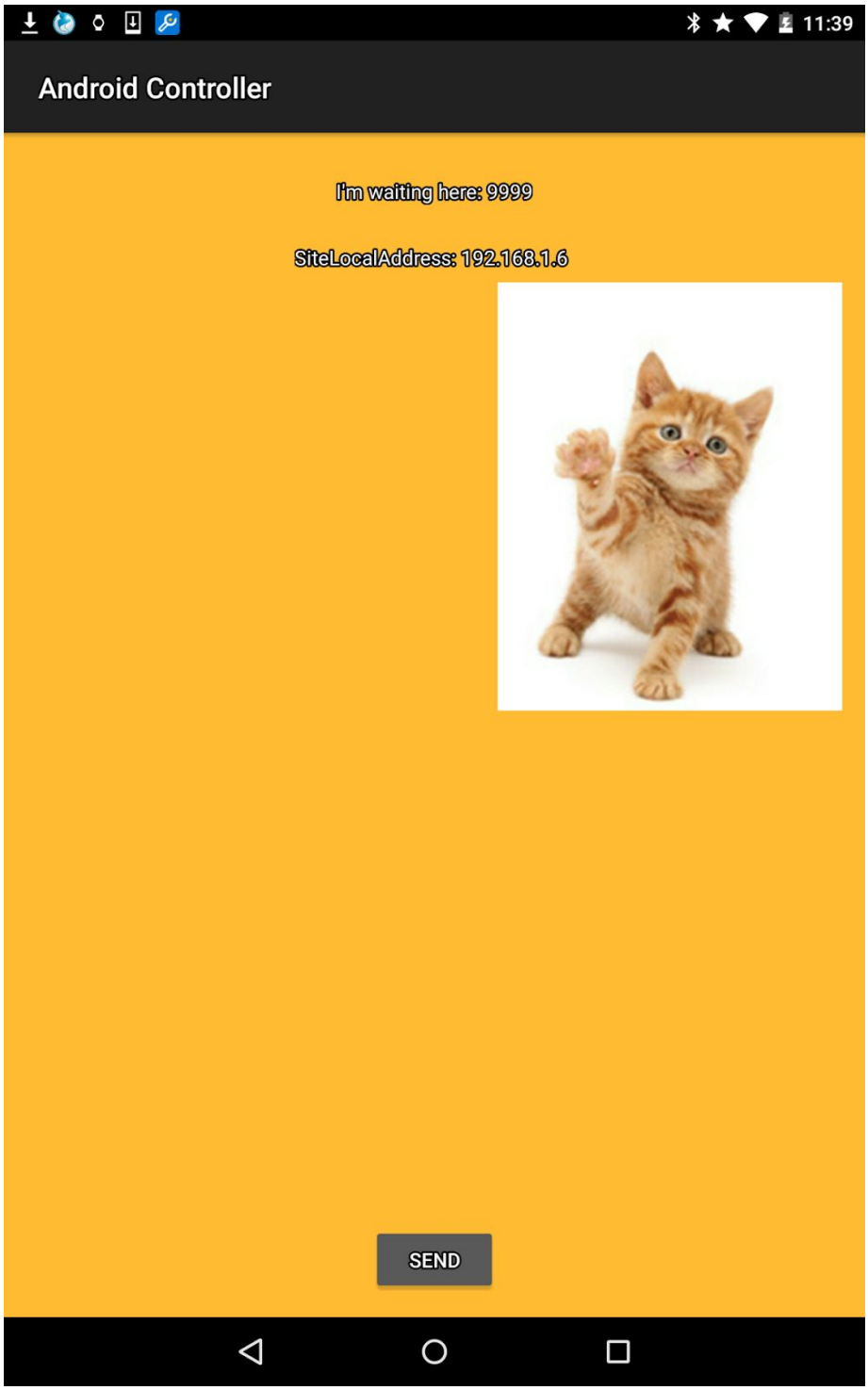
Through Spark, we will compare the picture you uploaded with another picture from Instagram and will let you know the opinion about this picture depending on the information we already had (Sentiment Analysis). Here is the main part of machine learning since every time we run another picture, our knowledge about this user increases. Also, the recommendation part will keep getting better and better.

In our application we were so strict for comparing the pictures in Spark so that the user will get the best result and later this result will be compared to the ten animal categories we used in our application. These categories contain animals' species such as [dogs, cats, birds, etc.]. The main part is to define the picture for which category it belongs so that all the coming recommendations are related to this result.

Step 1: Define 9 categories:

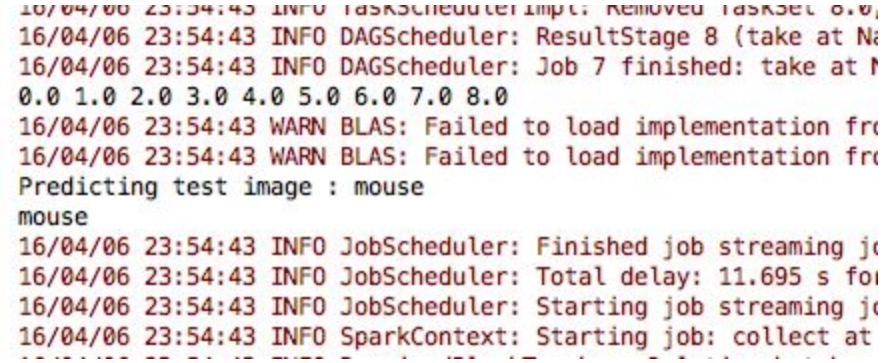
Step 2: Collect images as training data to categorize the image into these 9 categories using image tag searching.

Step 3: Collect image from device



Step 4: Train the images in Spark.

Step 5: Categorize the image data.



Step 6: Do recommendation of the user according to sentimental analysis.

Step 7: Compare the image category collected from device and the user recommendation category.

Step 8: Send the comparison result to the device.

8) Movement of Romo

After we made sure that all these features working properly, we started to think about the movement of the robot. Most of the code for the movement is ready and we will use Xcode tool to write and edit the movement part of the robot using Object C language. So far, we were doing well, until we faced some problems with the robot. The robot is not moving so far. So we are waiting to check whether we will keep working on the movement or we will just complete the project without this part. As you can see from the first increment we highlighted the movement part in red because the problem from the robot itself not from the code. So, if the problem solved we will complete our project plan and do the movement. Otherwise, we have to cut out the movement part and add another features instead.

o Implementation status report

Work completed:

Descriptions and timelines

- Jan 28- Feb 4, our team tried to get familiar with the basic functions that watch can do with android studio. For example, how to send a non-static notification from android studio to emulator. And we discussed how we could use this features in our coming project.
- Feb 5-Feb 11, by utilizing the source code provided in tutorial lab 3, we created heart rate monitoring function and pass the reading data to the tablet. Related numbers show on the tablet as it showed on watch.ni
- Feb 12- Feb 17, we are currently trying to see if there are any functionalities we could possibly utilize by using hadoop and meta-data processing techniques in our project when we deal with large amount of data etc.
- Feb 18- Mar 11, we focused on machine learning part using spark MLlib and send the information to smartphone.

Responsibility (Task, Person)

- Testing for existing features between phone and watch and identify potential issues that may exist in previous code. Provide suggestions on which specific part of functions need to be improved in terms of user experience, easiness, capabilities of

each functions so that to guarantee the smooth transpassing between existing and further functions we will implement----Gharibi, Mohamed

- Based on lab/tutorial examples to modified detail source code and implement specific functions to match with the need our project ----Huang, Wei
- Assisted Wei and Xia on the detail code implementation . Formatted user's function layout on smartphone----Sattineni, Vipin Reddy
- Use Spark MLlib to do analysis of big data, like category analysis, recognition. Focus on heart rate part. Tested on all the existing command based on our project particular definitions and from user's watch----Xia, Ting

Time taken (#hours)

- 38 hours in total

Contributions (members/percentage)

- Gharibi, Mohamed/ 25%
- Huang, Wei/ 25%
- Sattineni, Vipin Reddy/ 25%
- Xia, Ting/ 25%

Work to be completed

Description

- Start thinking about how to utilize the image/camera features on the phone and combine it with our project. May considering capture image of people's face and compare with a general database for checking purposes.
- Start trying to implement objective C code to realize simply robot moving (forward, back, left, right etc)

Responsibility (Task, Person)

- Testing for existing features between phone and watch and identify potential issues that may exist in previous code. Provide suggestions on which specific part of functions need to be modified so that to guarantee the smooth function overflow when new functions are added in the future. Get involved in robot motion controls by using objective C---- Gharibi, Mohamed
- Based on Mohamed suggestion improve previous features(app layout, functions stabilities etc). Based on lab/tutorial examples to modify detail source code and add

into new functions as needed. Get involved in robot motion controls by using objective C and image capture(sending certain notifications based on criteria defined) ---Huang, Wei

- Get involved in robot motion controls by using objective C. Doing more research on how to realize two-ways communication between watch and phone and then sending real time data/notifications to each other---Sattineni, Vipin Reddy
- Get involved in robot motion controls by using objective C. Doing research on how to train data to recognize human face and automatically comparing with certain database at backstage. Then execute certain command by definition or from user's watch--Xia, Ting

Time to be taken (estimated #hours)

- Approximately 5-8 hours in tuning the new features such as how to control the robot to move by using objective C and RomoMe functions.
- Another 5-8 hours in improving the previous features such as how to send message from watch to phone or from phone to watch in the real time manner.

Issues/Concerns

- Though we start considering new features such as image capture/recognition, automatic robot traceback features, we are not sure how well our overall program can handle the more complex situations since large amount of data will flow into the system(especially image file etc)

Bibliography

1. Android Developers website: http://developer.android.com/index.html
2. IBM Alchemyapi: https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/services-catalog.html
3. CS 5542 lab tutorial source code 4,5,6,7