

# Assist Robot

Phase I

Project Report

**Project Team – 8**

Team Members

Priyadarsini Nidadavolu(17)

Deepthi Priyadarshini Penmetsa(22)

Dheeraja Vallabhaneni(28)

Tej Kumar Yentrapragada(33)

## Table of Contents

Section	Description
1	Introduction
2	Project Goal and Objectives
3	Project Plan
	3.1 Schedule
	3.2 Project Timelines
	3.2.1 Team Members
	3.2.2 Tasks and Responsibilities
	3.3 Burndown Chart
4	First Increment Report
	4.1 Existing API
	4.2 Design of Features
	4.3 Implementation
	4.4 Deployment
5	Project Management
	5.1 Implementation Status Reports
6	Bibliography

## **1. Introduction:**

The main goal of the project is to help people in finding their misplaced objects. Basically humans have a tendency to forget their belongings somewhere in their house and search for it for hours together. For example, if I have an important business meeting to attend, but I don't remember where I placed my car keys, then I will be in huge loss. So to prevail in these circumstances here comes our Friendly Robot- My Friend which could assist me in keeping track of my personal things. So what this robot will do is that it will have entire map (laser scan) of the building and objects in the building in its memory. So we will feed the robot with the objects that are highly important to us, like car keys, some files, phone and laptop. This robot will keep its eye on these objects and notify their location to its master upon request.

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

The primary goals of our project is described below:

- To implement a module which has an interaction with robot. Eg: You can ask few questions to the robot and the Robot will be responding to you back. You can ask the robot about your misplaced phone. So that it will answer you after it had found the phone.
- To make the robot learn about the personal items like laptops, phones, watch, keys etc.
- To design a robot which can find the learned objects that are misplaced in a building.
- To send a notification to your smart watch when it finds the lost object.
- To remind the user about his day to day events that were previously taught.
- To make the robot act as an assistant in getting things specified by the user. ( Mr Robot – Get me my phone).
- To make a single robot act as assistant to all the people living in same house. It recognizes the user first and then assists that particular user in finding the belongings.

### 3. Project Plan:

#### 3.1 Schedule:

**Stories:** Four user stories had been created as part of Iteration 1. Here are the snapshots for the stories which are in closed and opened state.

This screenshot shows the GitHub Issues page for the repository 'npdarsini/Assist-Robot'. The search bar at the top contains the query 'is:open author:npdarsini'. The results show two issues:

- Test - Different Objects** (enhancement): #4 opened 35 minutes ago by npdarsini. Project - Increment 1.
- Capture and feed the Images** (enhancement): #3 opened 36 minutes ago by npdarsini. Project - Increment 1.

The interface includes standard GitHub features like filters, labels, and milestones. A ProTip at the bottom suggests editing multiple issues at once.



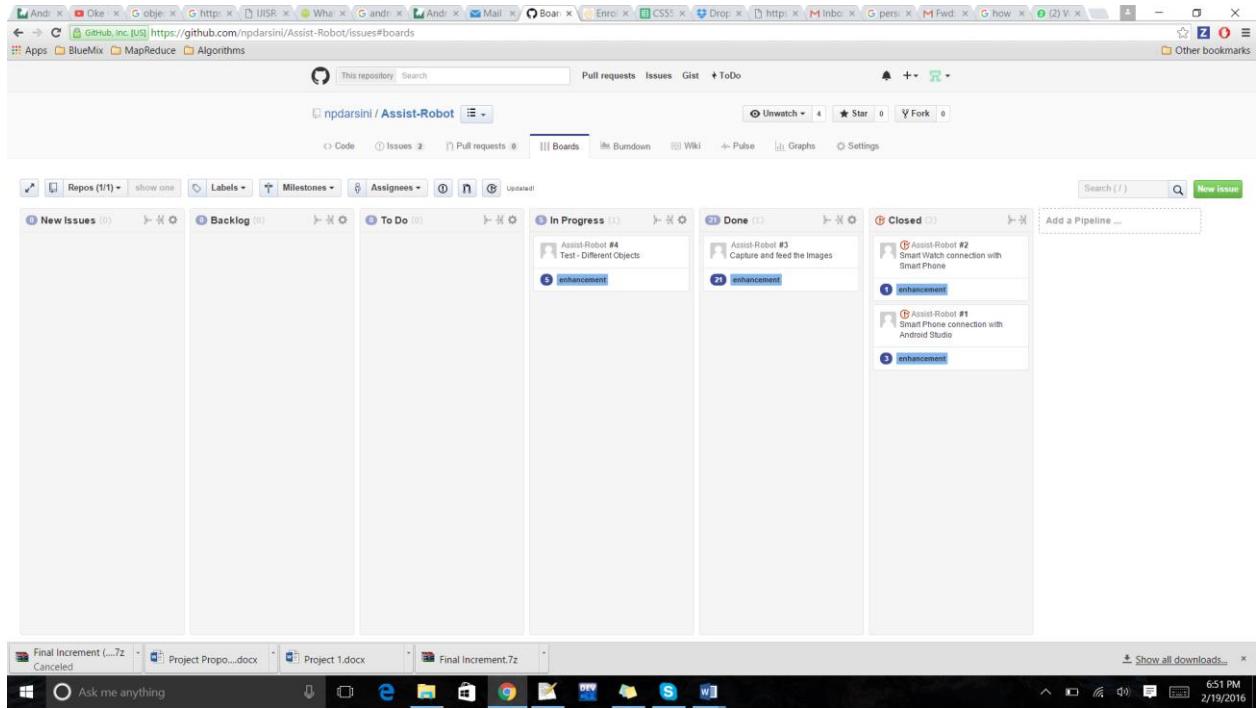
This screenshot shows the GitHub Issues page for the repository 'npdarsini/Assist-Robot'. The search bar at the top contains the query 'is:closed author:npdarsini'. The results show two issues:

- Smart Watch connection with Smart Phone** (enhancement): #2 opened 40 minutes ago by npdarsini. Project - Increment 1.
- Smart Phone connection with Android Studio** (enhancement): #1 opened 42 minutes ago by npdarsini. Project - Increment 1.

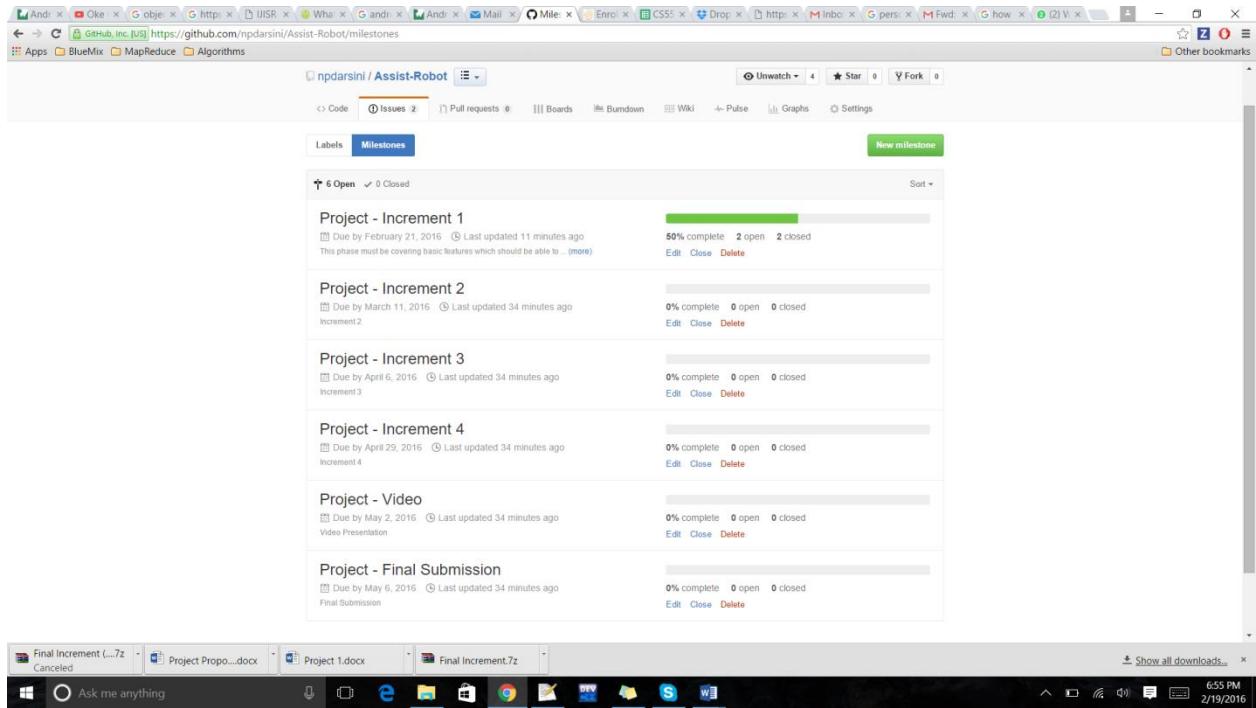
The interface includes standard GitHub features like filters, labels, and milestones. A ProTip at the bottom suggests mixing and matching filters to narrow down the search.



## Board:

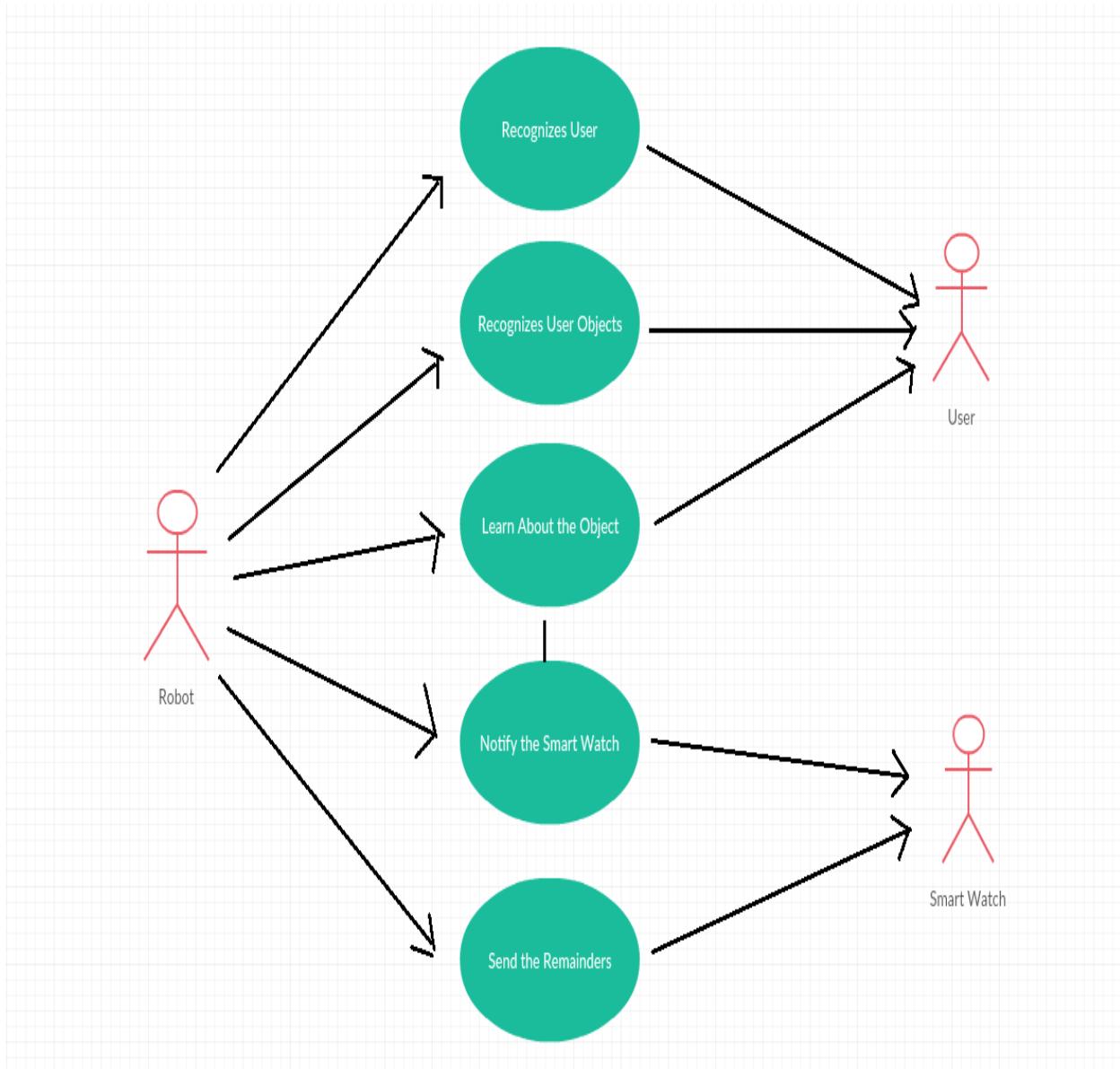


## Milestones:

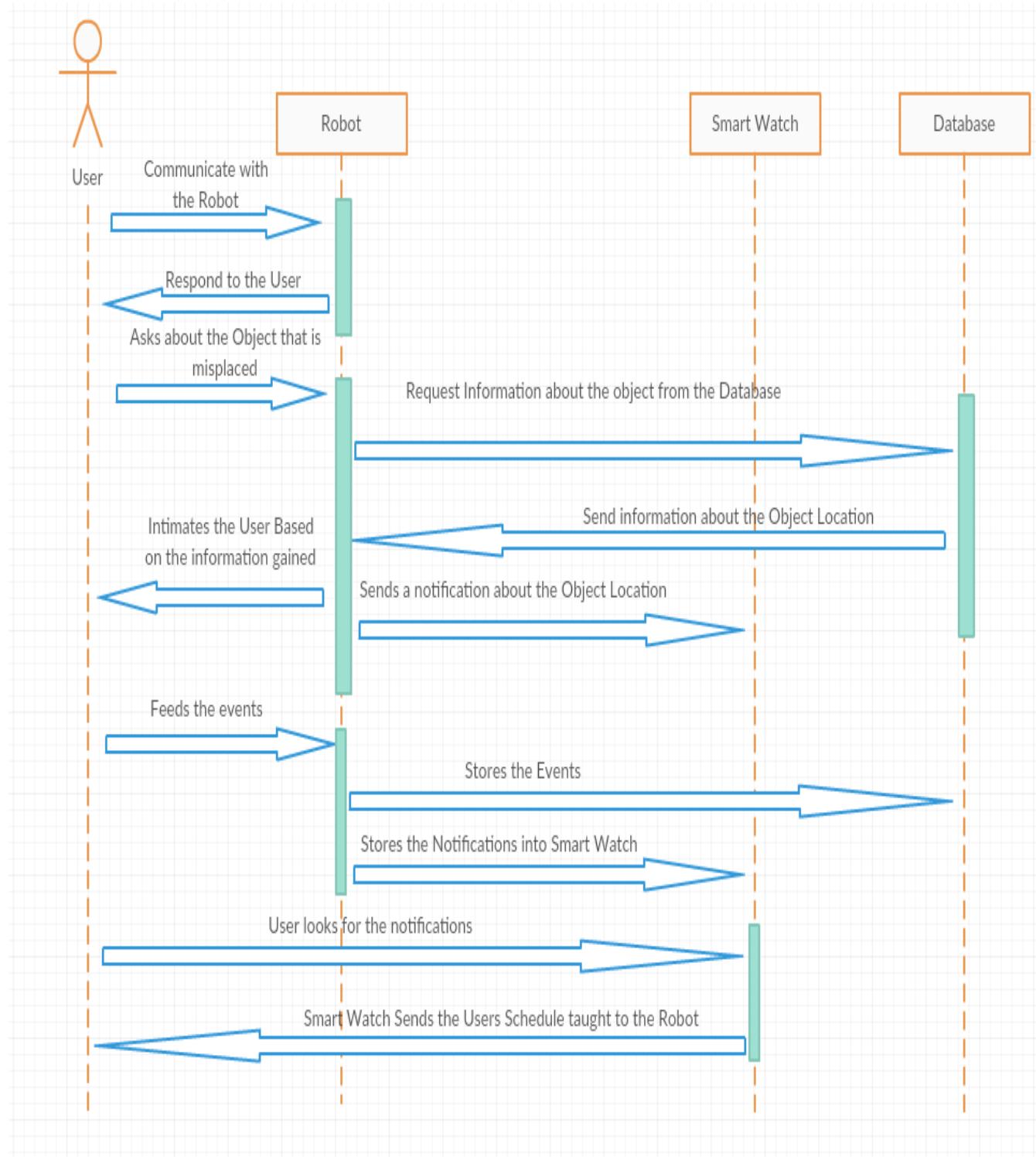


### 3.1.1 UML Diagrams:

#### Use Case Diagram



## Sequence Diagram:



### **3.2 Project Timelines:**

<b>Increment</b>	<b>Deadline</b>
<b>Increment 1</b>	19 February 2015
<b>Increment 2</b>	11 March 2016
<b>Increment 3</b>	6 April 2016
<b>Increment 4</b>	29 April 2016
<b>Final Submission</b>	6 May 2016

#### **3.2.1 Team Members:**

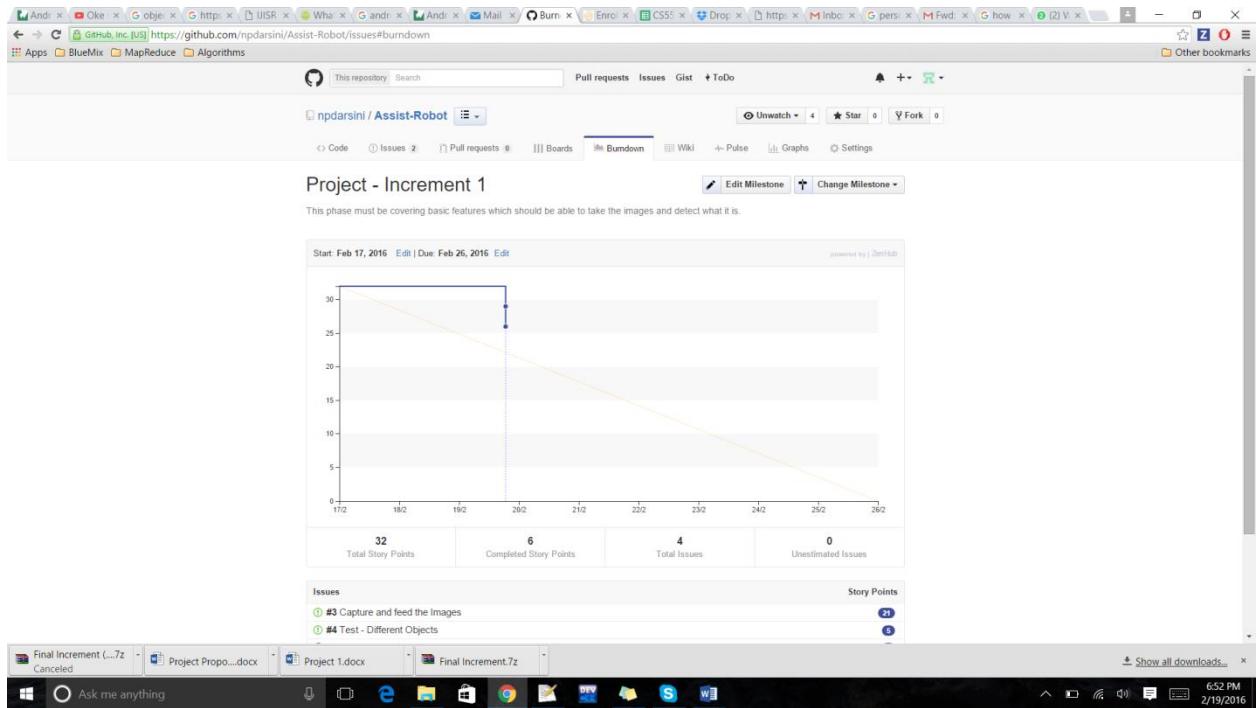
1. Priyadarsini Nidadavolu – 17
2. Deepthi Priyadarshini Penmetsa – 22
3. Dheeraja Vallabhaneni – 28
4. Tej Kumar Yentrapragada – 33

#### **3.2.2 Tasks and Responsibilities:**

- **Machine Learning and R Programming** – Deepthi Priyadarshini Penmetsa
- **Spark and Hadoop Technologies** – Priyadarsini Nidadavolu
- **Objective C and IOS Programming** – Tej Kumar Yentrapragada
- **Android Programming** - Dheeraja Vallabhaneni

### 3.3 Burndown Chart:

#### Burndown:



## 4. First Increment Report

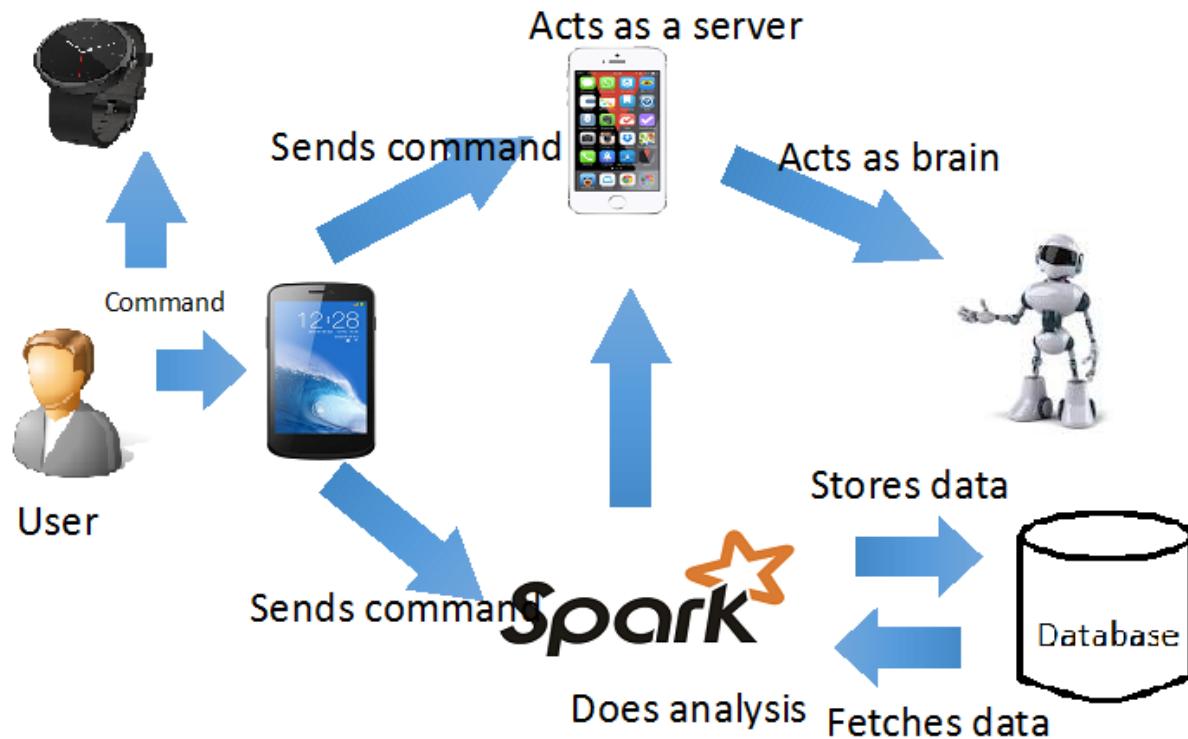
### 4.1 Existing API:

#### IBM Alchemy API

This API basically performs machine learning and natural language processing techniques. Some of its features include semantic text analysis, sentimental analysis, deep learning, face detection and reorganization, speech to text and vice versa conversions etc. In this we had used this API in order to recognize the objects that we want to teach the Robot.

Achievements upon using this API – The Robot could identify basic objects like laptop, phone, bottle etc.

## 4.2 Design of Features:



The architecture of our system could be like the user can give commands to the client device which is android phone. Further the Iphone which acts as a server could take commands from the android phone and passes it to the Robot. The Robot performs the necessary actions of the received command and return back to the Android Phone. It also sends the notifications to the Android Smart Watch. The Android device can also pass the command to the Spark and fetch the data from the database (MongoDB, Hadoop DB).

### System Features

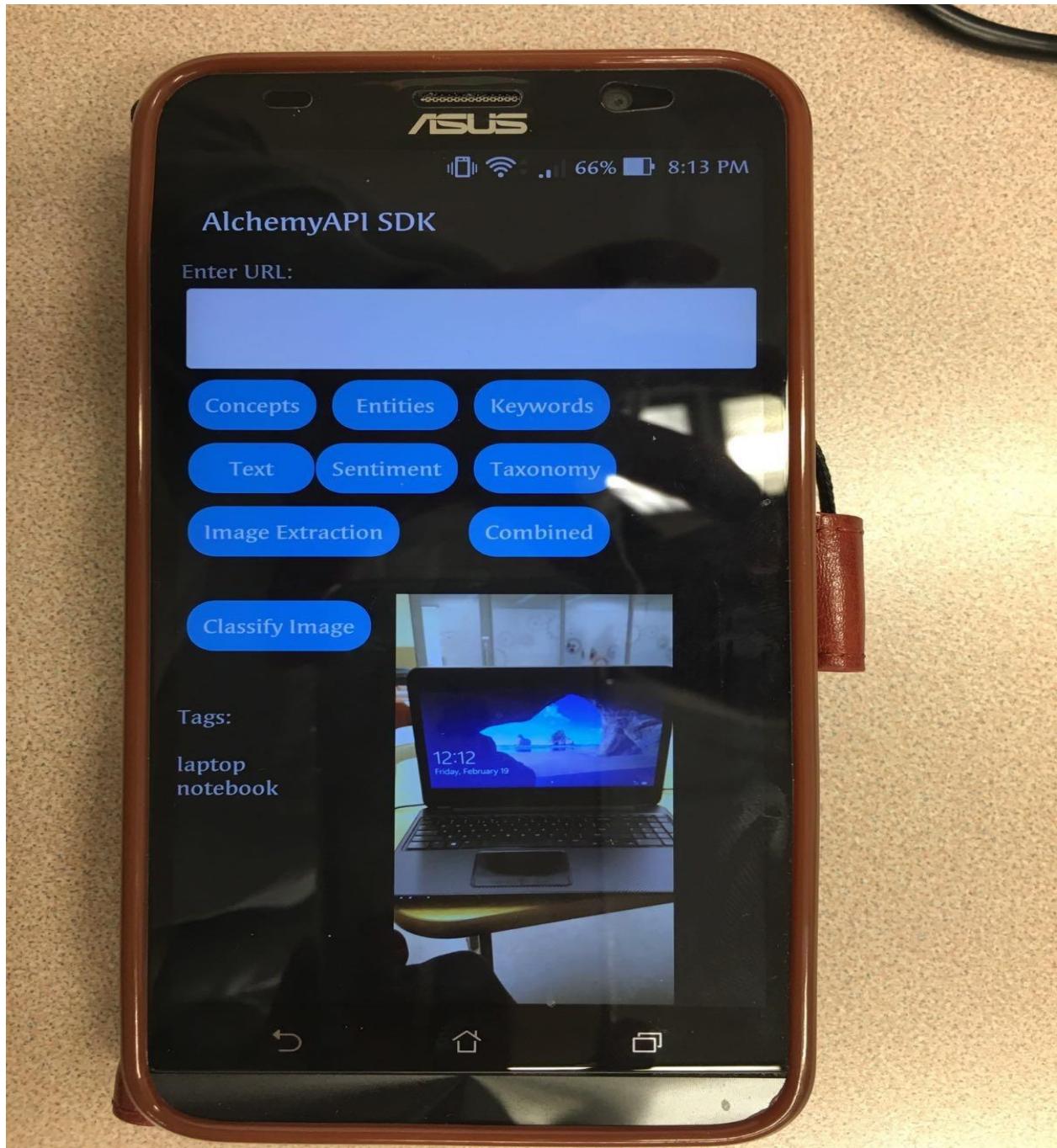
These are the following features we developed as part of this phase:

We had used IBM's Alchemy API and able to make our Robot to detect the object and return the object name as a result.

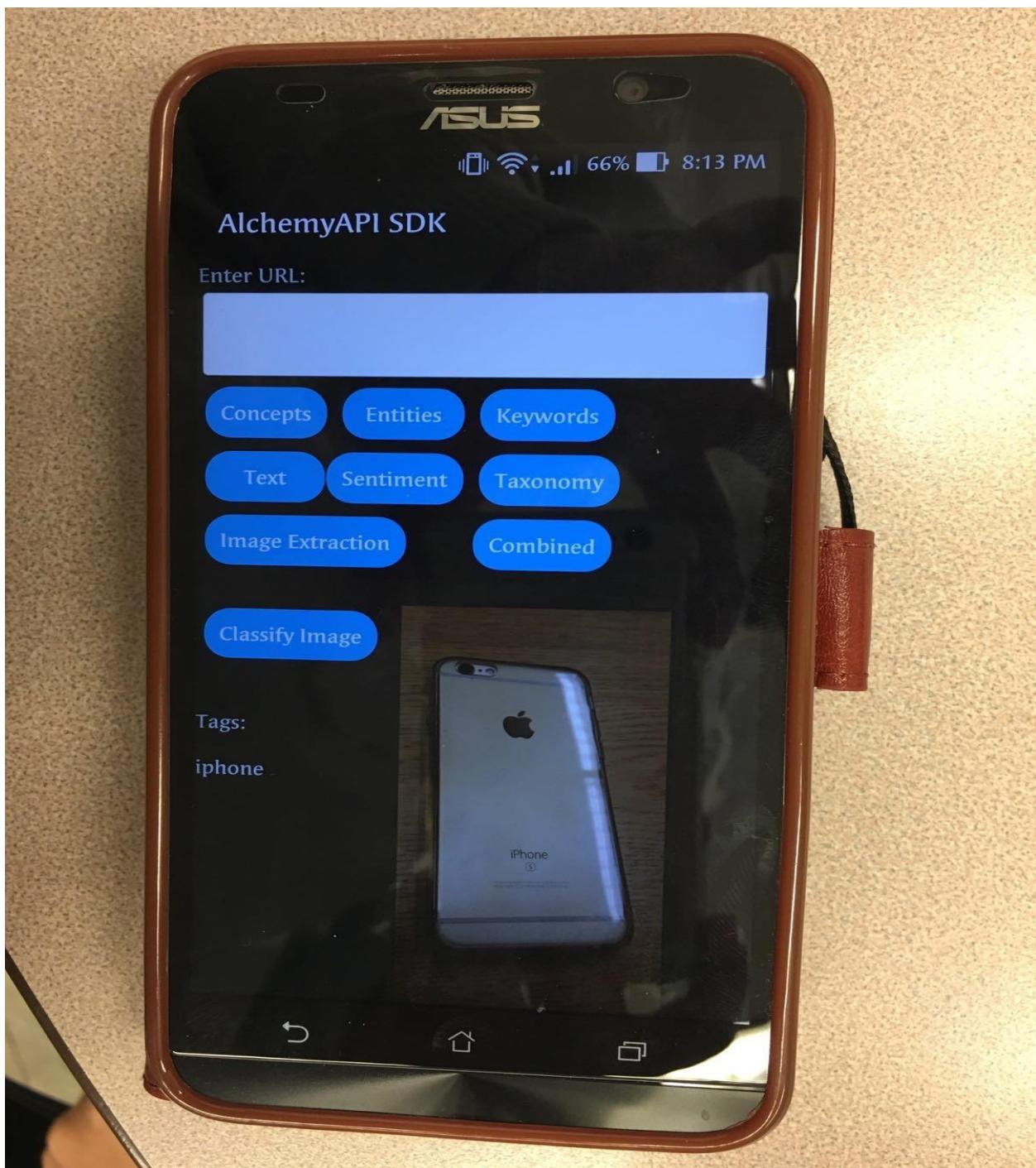
## 4.3 Implementation:

### Mobile Client Implementation – Snapshots

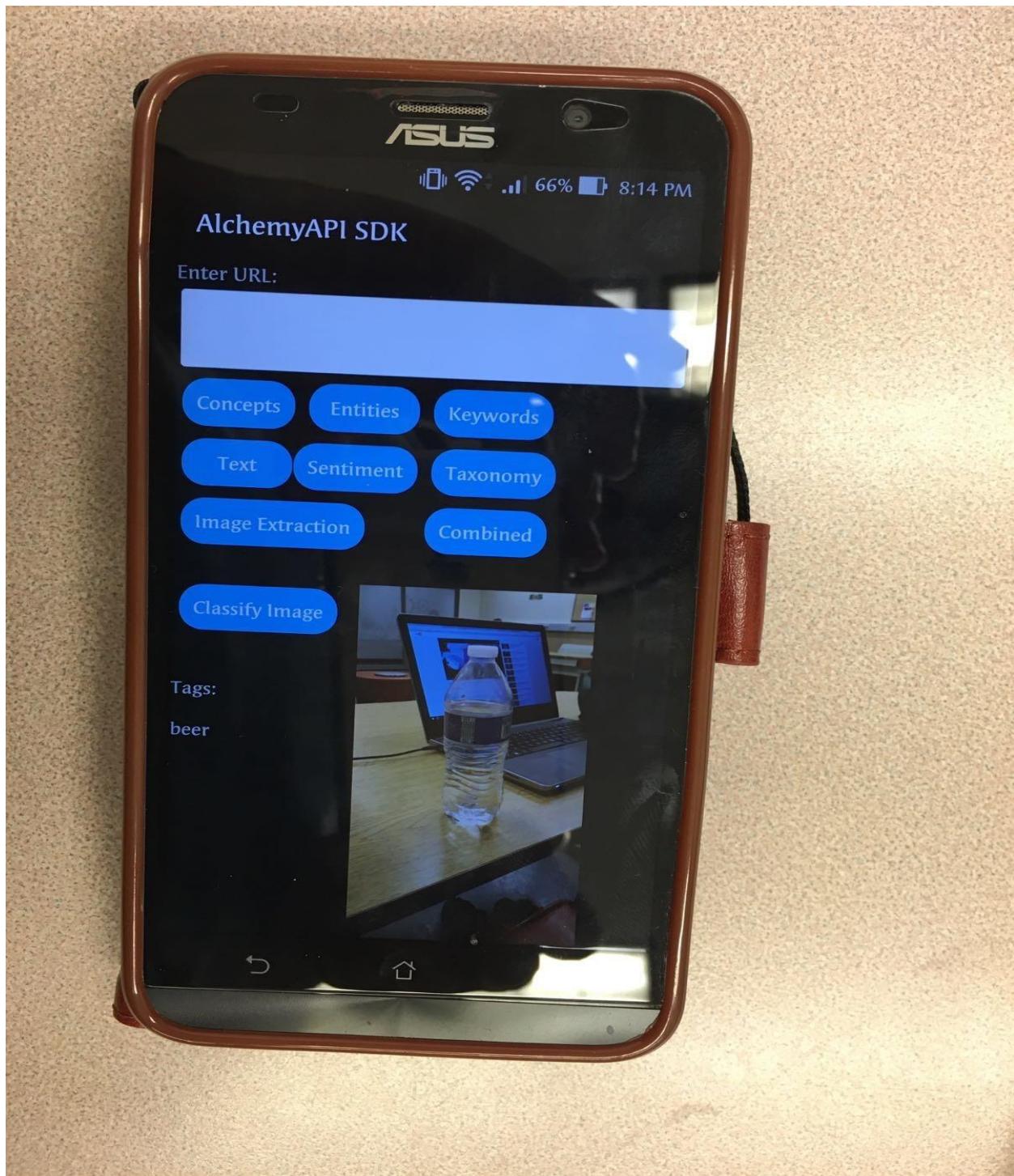
This snapshot shows us that the application is able to identify the object and names its Laptop.



This snapshot shows us that the application is able to identify the object and names it as an Iphone.



This snapshot shows us that the application is able to identify the bottle.



## 4.4 Deployment:

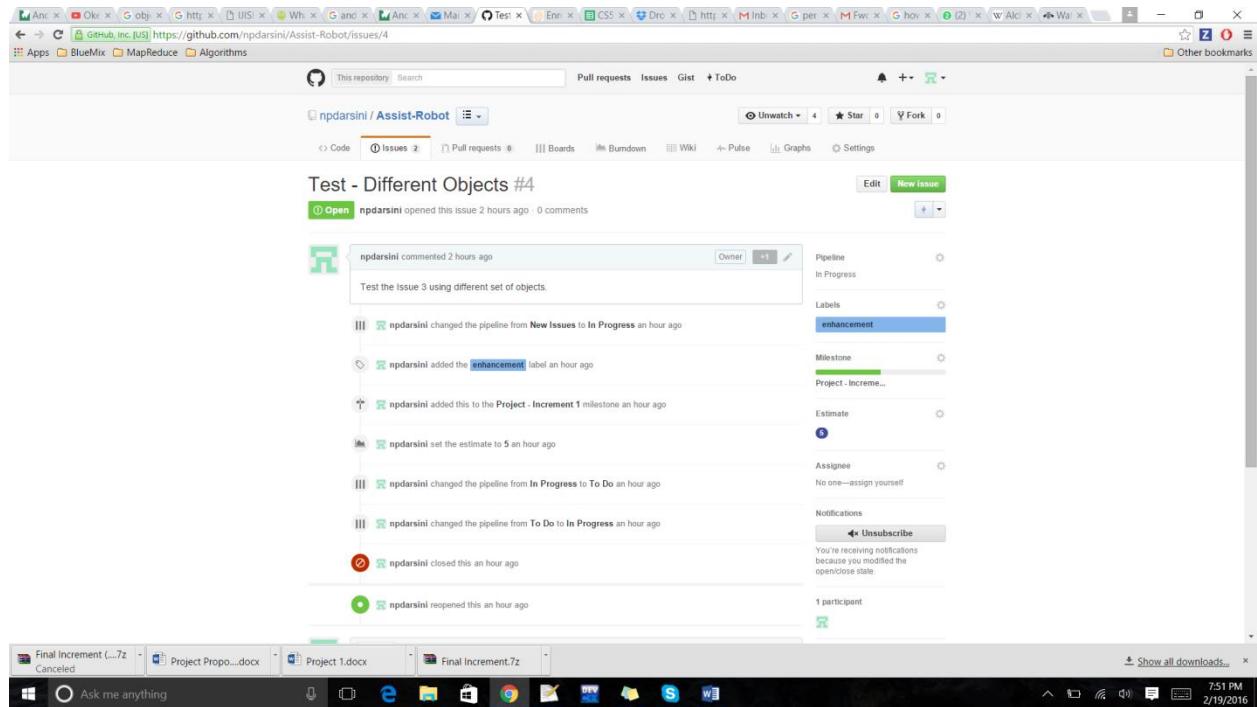
### Git Hub Link:

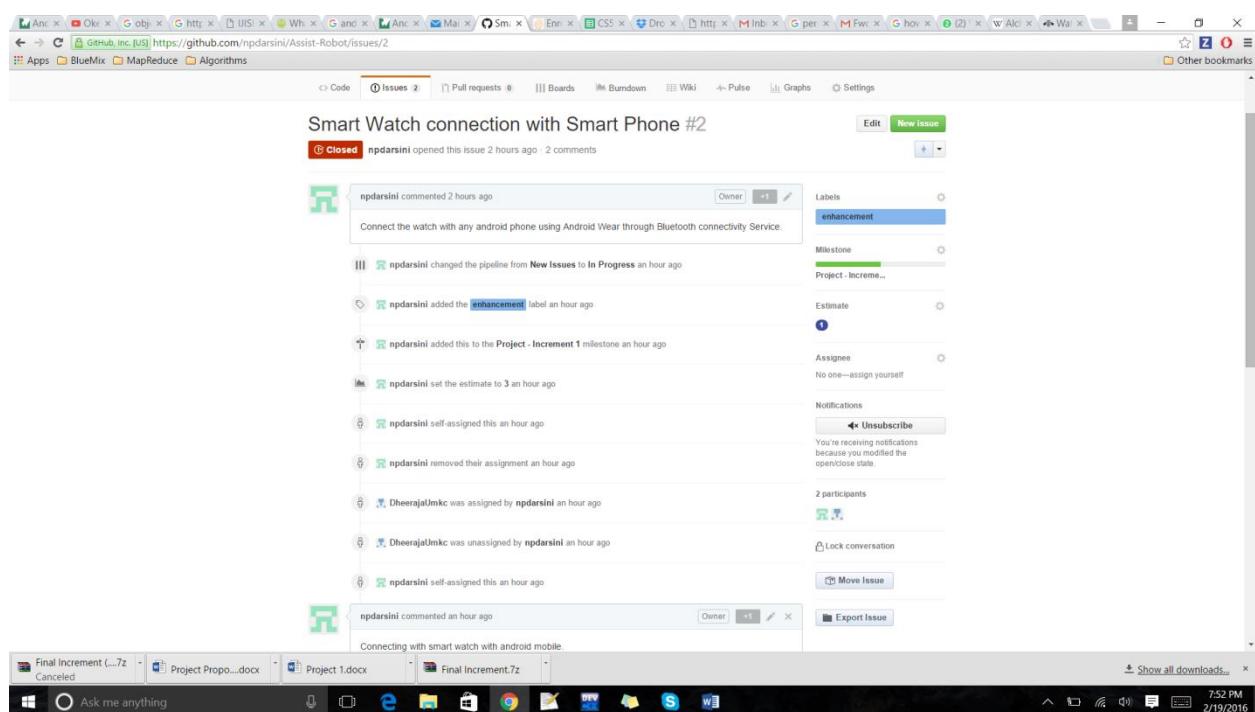
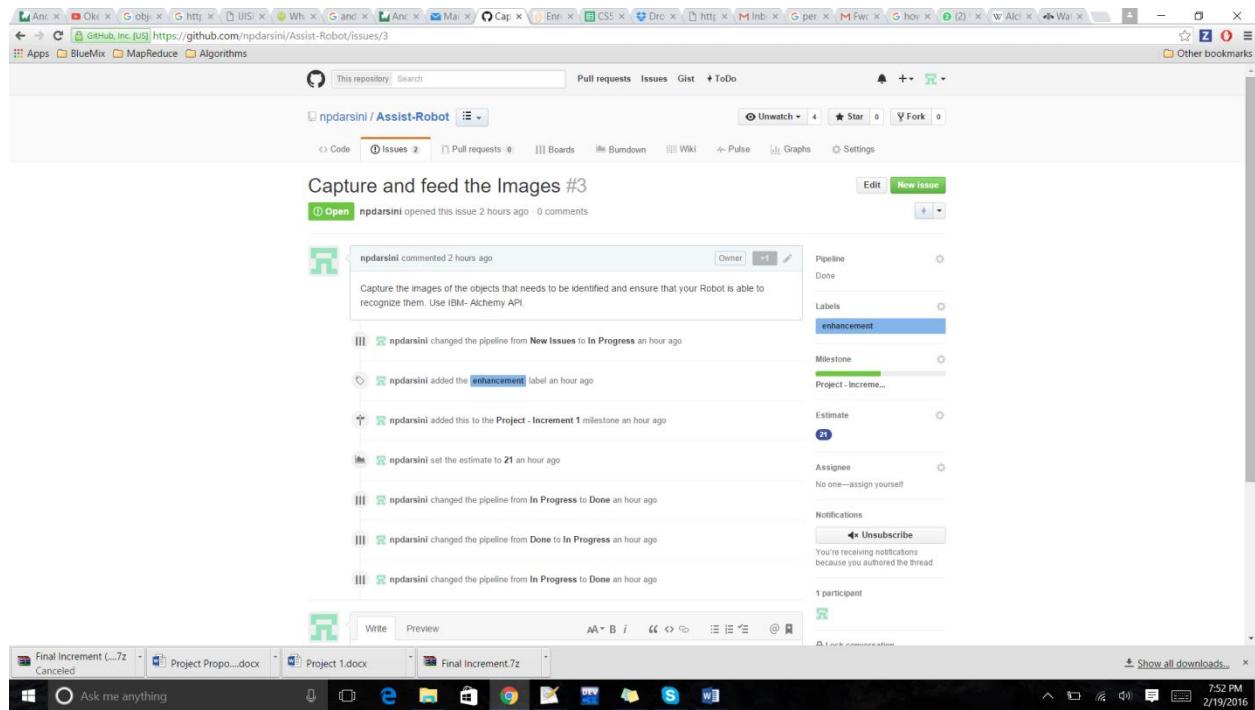
<https://github.com/npdarsini/Assist-Robot>

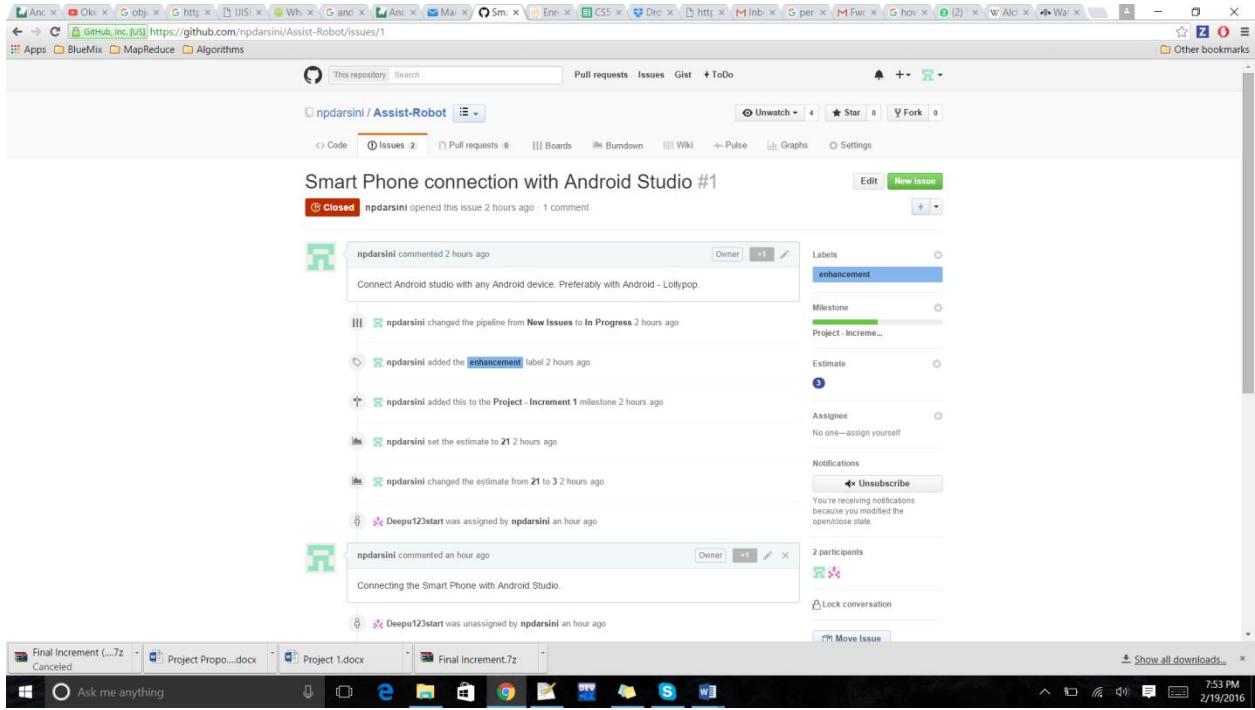
## 5. Project Management:

### 5.1 Implementation status report:

20% implementation has been implemented. This phase involves the basic object identification that is required for finding and location the objects. The team members has an equal contribution towards the development and it took around 2 complete days to give an outlook for this phase.







## Bibliography:

Lab Tutorials and the material provided by Dr. Lee.

# Assist Robot

Phase II

Project Report

**Project Team – 8**

Team Members

Priyadarsini Nidadavolu(17)

Deepthi Priyadarshini Penmetsa(22)

Dheeraja Vallabhaneni(28)

Tej Kumar Yentrapragada(33)

## Table of Contents

Section	Description
1	Introduction
2	Project Goal and Objectives
3	Project Plan
	3.1 Schedule
	3.2 Project Timelines
	3.2.1 Team Members
	3.2.2 Tasks and Responsibilities
	3.3 Burndown Chart
4	Increment Report
	4.1 Increment based on Phases
	4.1.1 Phase 1 - Existing API
	4.1.2 Phase 2 - Recommendation System
	4.2 Design of Features
	4.3 Implementation
	4.4 Deployment
5	Project Management
	5.1 Implementation Status Reports
6	Bibliography

## **1. Introduction:**

The main goal of the project is to help people in finding their misplaced objects. Basically humans have a tendency to forget their belongings somewhere in their house and search for it for hours together. For example, if I have an important business meeting to attend, but I don't remember where I placed my car keys, then I will be in huge loss. So to prevail in these circumstances here comes our Friendly Robot- My Friend which could assist me in keeping track of my personal things. So what this robot will do is that it will have entire map (laser scan) of the building and objects in the building in its memory. So we will feed the robot with the objects that are highly important to us, like car keys, some files, phone and laptop. This robot will keep its eye on these objects and notify their location to its master upon request. Additionally our Robot who will be an eFriend who will help us to choose the furniture to our home. Also our robot will suggest us the top rated books.

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

The primary goals of our project is described below:

- To implement a module which has an interaction with robot. Eg: You can ask few questions to the robot and the Robot will be responding to you back. You can ask the robot about your misplaced phone. So that it will answer you after it had found the phone.
- To make the robot learn about the personal items like laptops, phones, watch, keys etc.
- To design a robot which can find the learned objects that are misplaced in a building.
- To build a recommendation system which will be able to recommend the list of books which are rated high and are related to our interests.
- To build a recommendation system which will suggest us about the latest furniture details, their quality and from which brand/ shop we could purchase from. This feature enables us to decorate our houses with rich interior designing.
- To send a notification to your smart watch when it finds the lost object.
- To remind the user about his day to day events that were previously taught.
- To make the robot act as an assistant in getting things specified by the user. ( Mr Robot – Get me my phone).

- To make a single robot act as assistant to all the people living in same house. It recognizes the user first and then assists that particular user in finding the belongings.

### 3. Project Plan:

#### 3.1 Schedule:

**Stories:** Four user stories had been created as part of Iteration 1. Here are the snapshots for the stories which are in closed and opened state.

The screenshot shows the GitHub Issues page for the repository 'npdarsini / Assist-Robot'. The 'Issues' tab is selected. There are two open issues listed:

- ① Test - Different Objects enhancement
- ② Capture and feed the Images enhancement

Both issues were opened 35 minutes ago by npdarsini. The interface includes filters, labels, milestones, and a 'New issue' button.



The screenshot shows the GitHub Issues page for the repository 'npdarsini / Assist-Robot'. The 'Issues' tab is selected. There are two closed issues listed:

- ① Smart Watch connection with Smart Phone enhancement
- ② Smart Phone connection with Android Studio enhancement

Both issues were opened 49 minutes ago by npdarsini. The interface includes filters, labels, milestones, and a 'New issue' button.

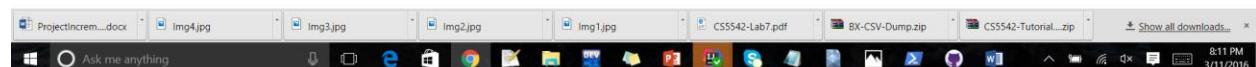


**Stories:** Eight user issues had been created as part of Iteration 2. Here are the snapshots for the stories which are in closed and opened state.

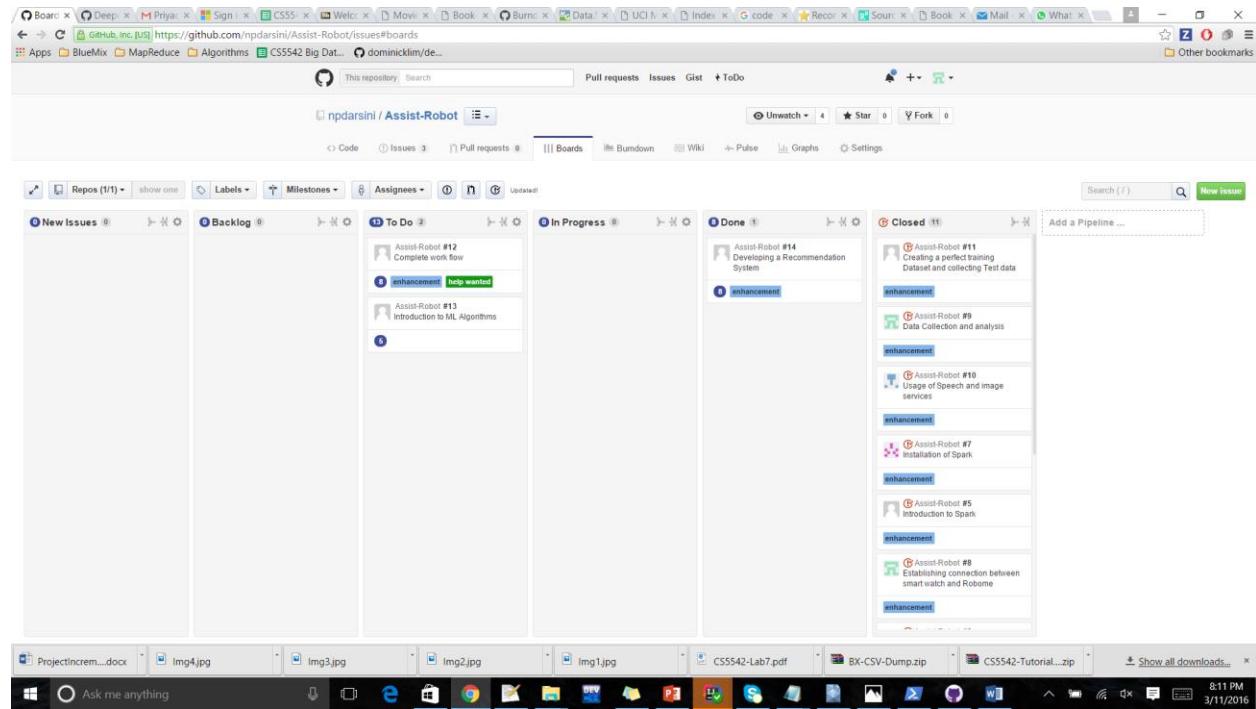
This screenshot shows the GitHub Issues page for the repository 'npdarsini/Assist-Robot'. The search bar contains the query 'is:issue milestone:"Project - Increment 2" is:closed'. The results list seven closed issues and one open issue. The open issue is titled 'Creating a perfect training Dataset and collecting Test data' and was opened 3 hours ago by DheerajaUmkc. The closed issues are: 'Usage of Speech and image services', 'Data Collection and analysis', 'Establishing connection between smart watch and Robome', 'Installation of Spark', 'Features of Robome', 'Introduction to Spark', and 'Developing a Recommendation System'. All closed issues were opened 4 days ago by DheerajaUmkc.



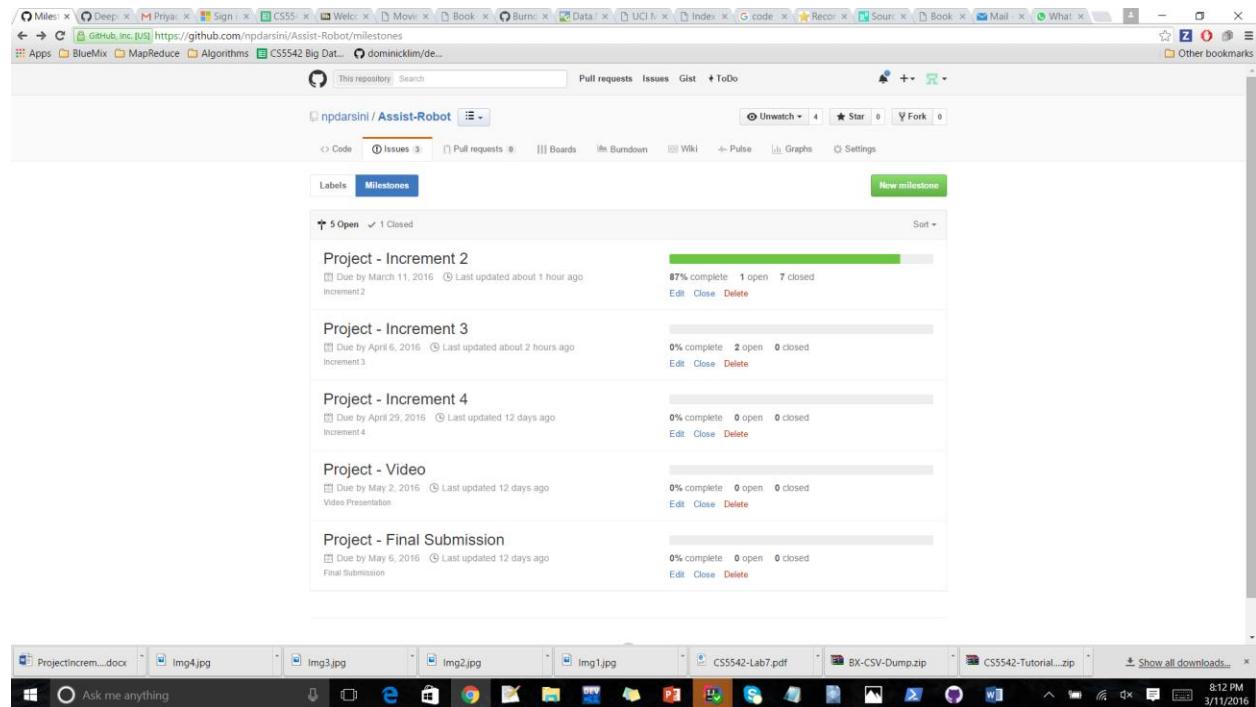
This screenshot shows the GitHub Issues page for the repository 'npdarsini/Assist-Robot'. The search bar contains the query 'is:issue milestone:"Project - Increment 2" is:open'. The results list one open issue titled 'Developing a Recommendation System' and one closed issue. The open issue was opened 3 hours ago by DheerajaUmkc and is currently in the 'Done' status.



## Board:

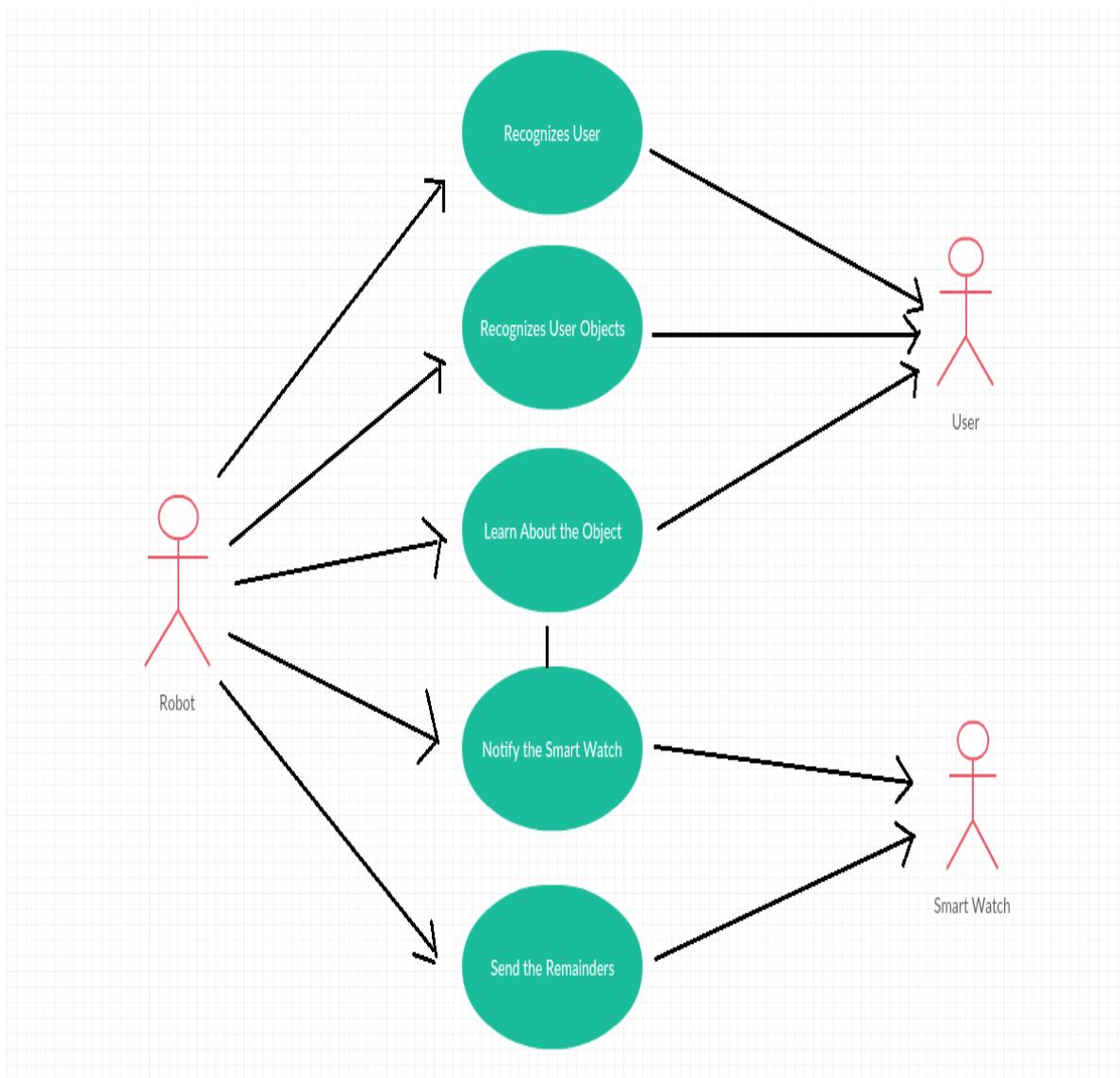


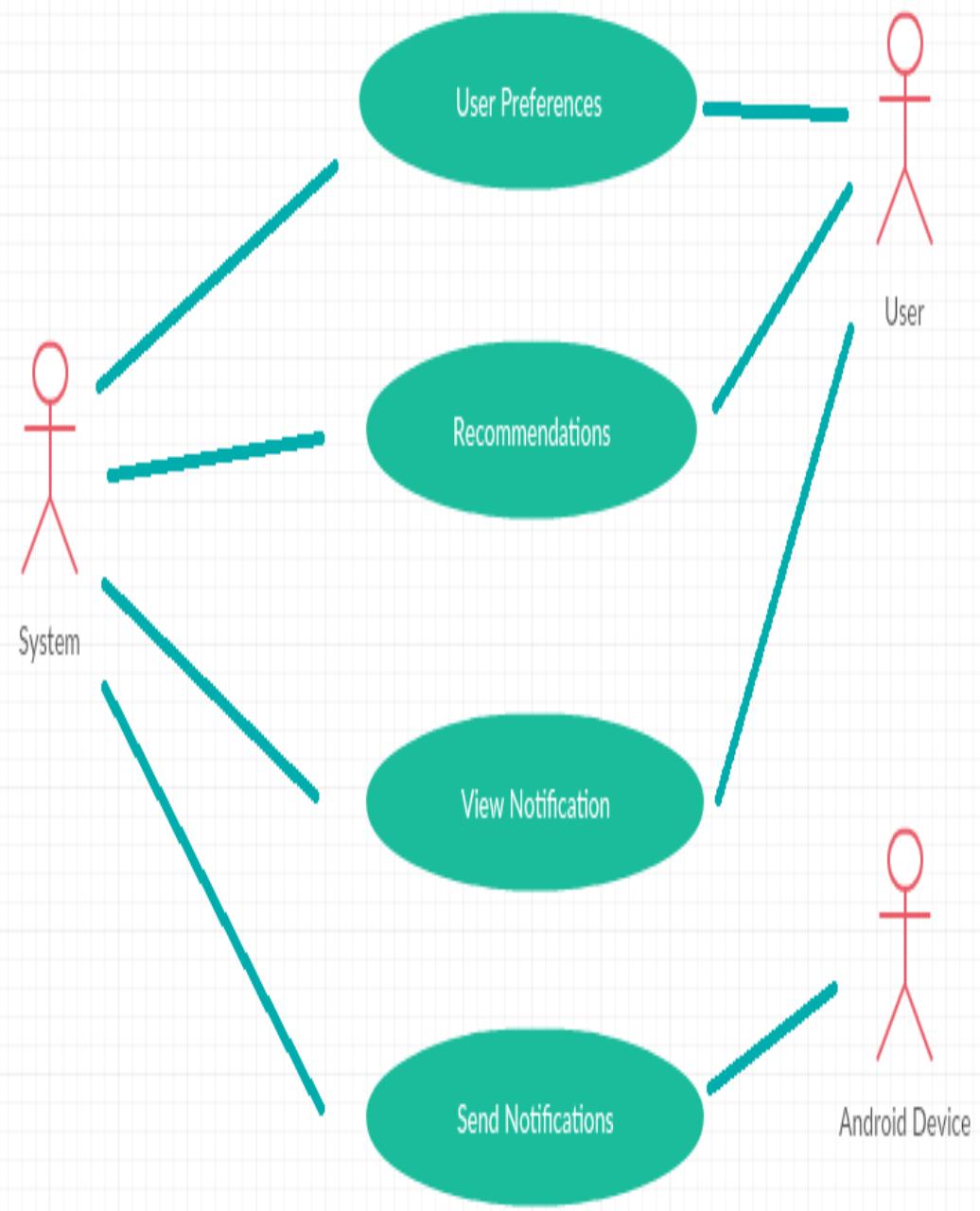
## Milestones:



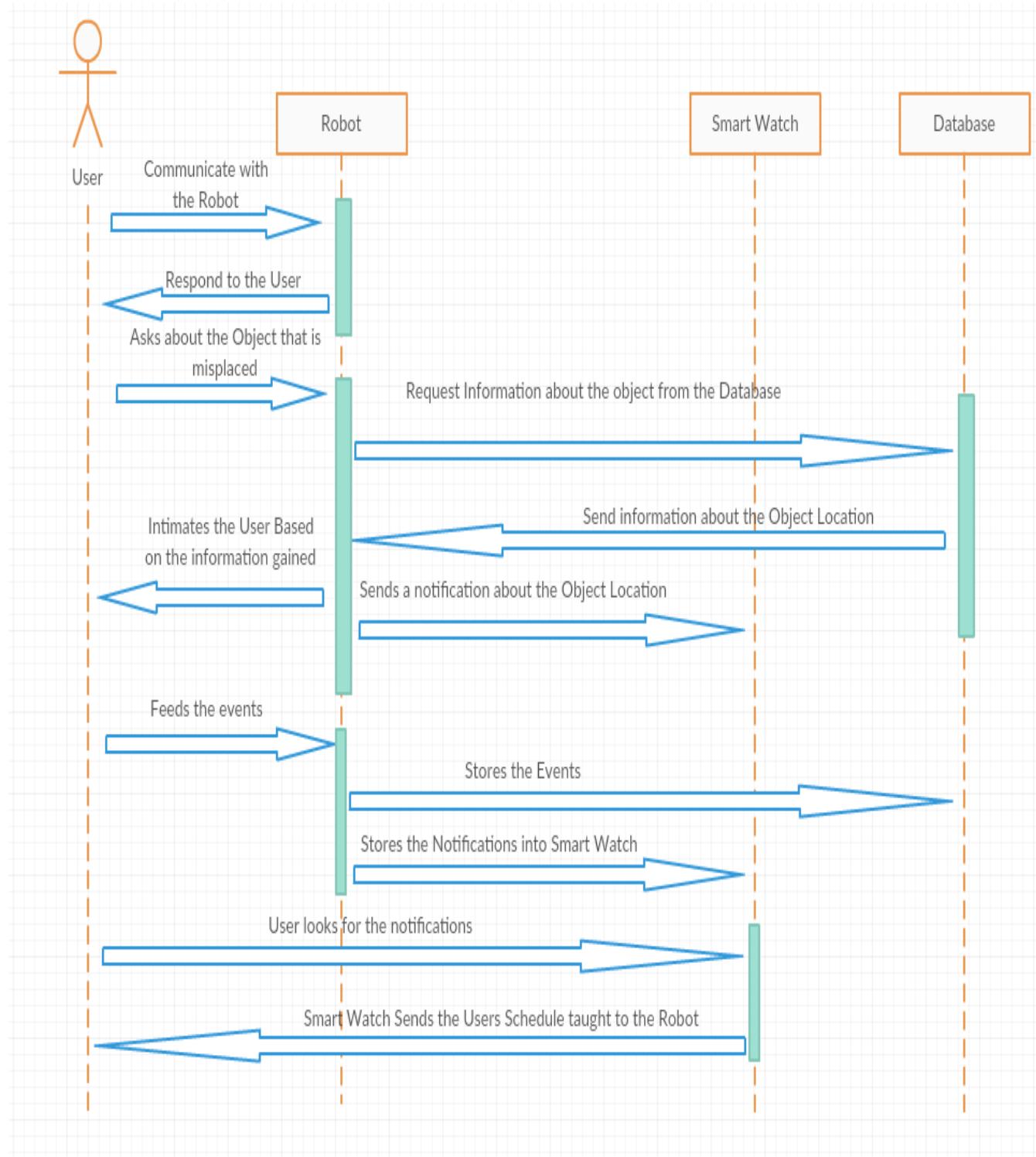
### 3.1.1 UML Diagrams:

#### Use Case Diagram

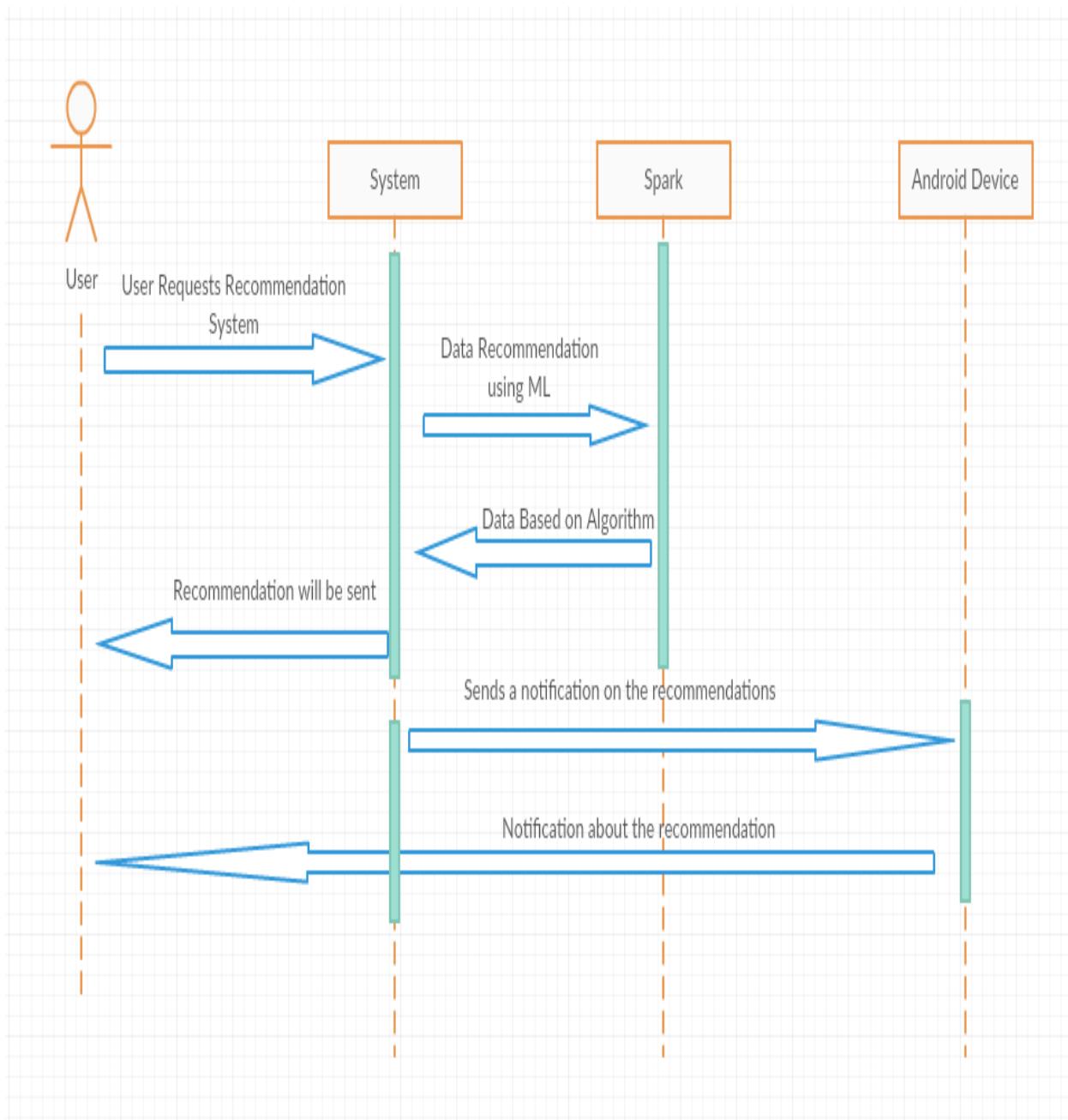




## Sequence Diagram:



## Sequence Diagram for Recommendation System



### **3.2 Project Timelines:**

Increment	Deadline
<b>Increment 1</b>	19 February 2015
<b>Increment 2</b>	11 March 2016
<b>Increment 3</b>	6 April 2016
<b>Increment 4</b>	29 April 2016
<b>Final Submission</b>	6 May 2016

#### **3.2.1 Team Members:**

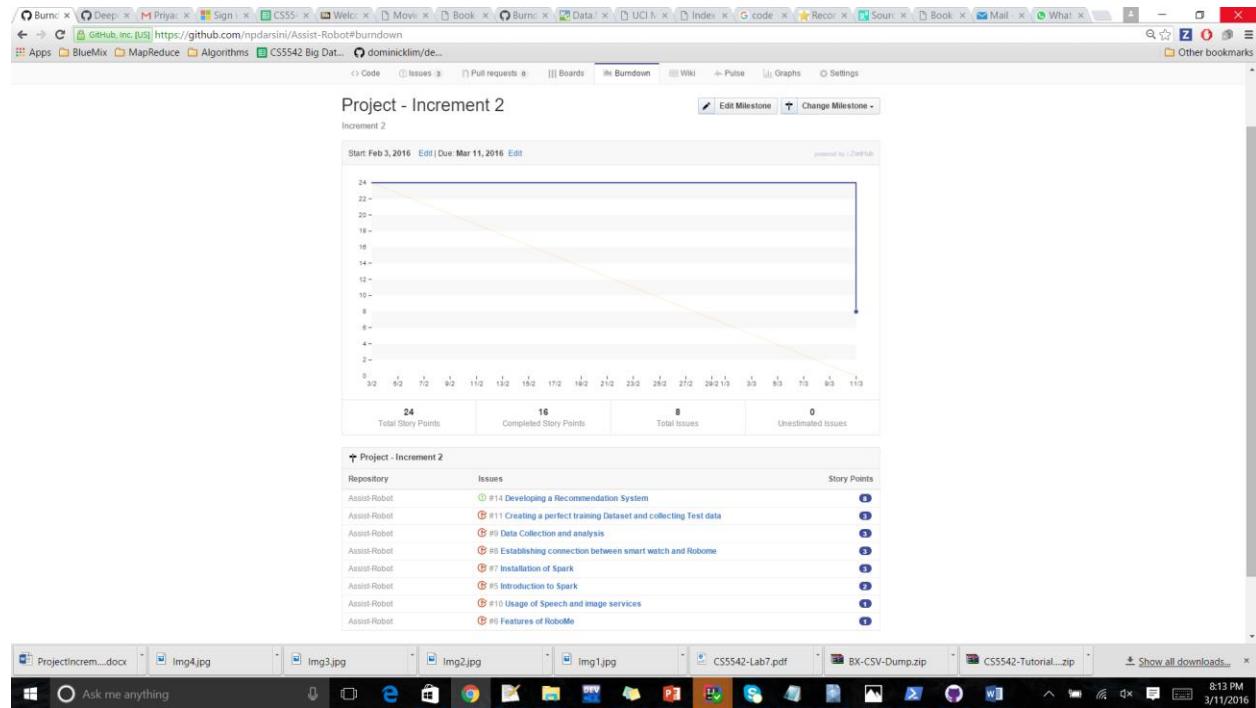
1. Priyadarsini Nidadavolu – 17
2. Deepthi Priyadarshini Penmetsa – 22
3. Dheeraja Vallabhaneni – 28
4. Tej Kumar Yentrapragada – 33

#### **3.2.2 Tasks and Responsibilities:**

- **Machine Learning and R Programming** – Deepthi Priyadarshini Penmetsa
- **Spark and Hadoop Technologies** – Priyadarsini Nidadavolu
- **Objective C and IOS Programming** – Tej Kumar Yentrapragada
- **Android Programming** - Dheeraja Vallabhaneni

### 3.3 Burndown Chart:

#### Burndown:



## 4. Increment Report

### 4.1 Incremental Explanations

#### 4.1.1 Phase 1 -Existing API:

##### IBM Alchemy API

This API basically performs machine learning and natural language processing techniques. Some of its features include semantic text analysis, sentimental analysis, deep learning, face detection and reorganization, speech to text and vice versa conversions etc. In this we had used this API in order to recognize the objects that we want to teach the Robot.

Achievements upon using this API – The Robot could identify basic objects like laptop, phone, bottle etc.

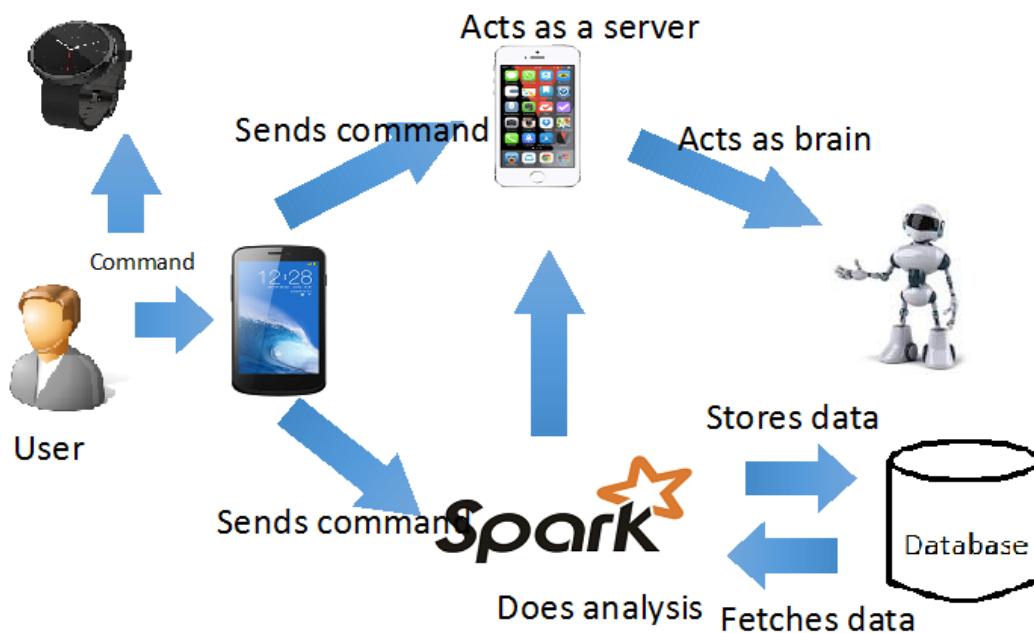
#### **4.1.2 Phase 2 - Recommendation System:**

In this phase we had developed two recommendation systems which can recommend the user about the popular furniture showrooms and the famous books. In this we provided the training set with user information (uid, name, ratings etc.,), furniture information (list of showrooms, location) and the book information (name, author etc.,).

The recommended notification has been sent to the android device (smart watch/phone) using Spark-Android Socket programming techniques.

#### **4.2 Design of Features:**

The architecture of our system could be like the user can give commands to the client device which is android phone. Further the Iphone which acts as a server could take commands from the android phone and passes it to the Robot. The Robot performs the necessary actions of the received command and return back to the Android Phone. It also sends the notifications to the Android Smart Watch. The Android device can also pass the command to the Spark and fetch the data from the database (MongoDB, Hadoop DB). Our system will be able to recommend the user based on the trained data sets and the notifications will be sent to the android device.



## **System Features**

The following are the features that were developed as part of Phase I:

We had used IBM's Alchemy API and able to make our Robot to detect the object and return the object name as a result.

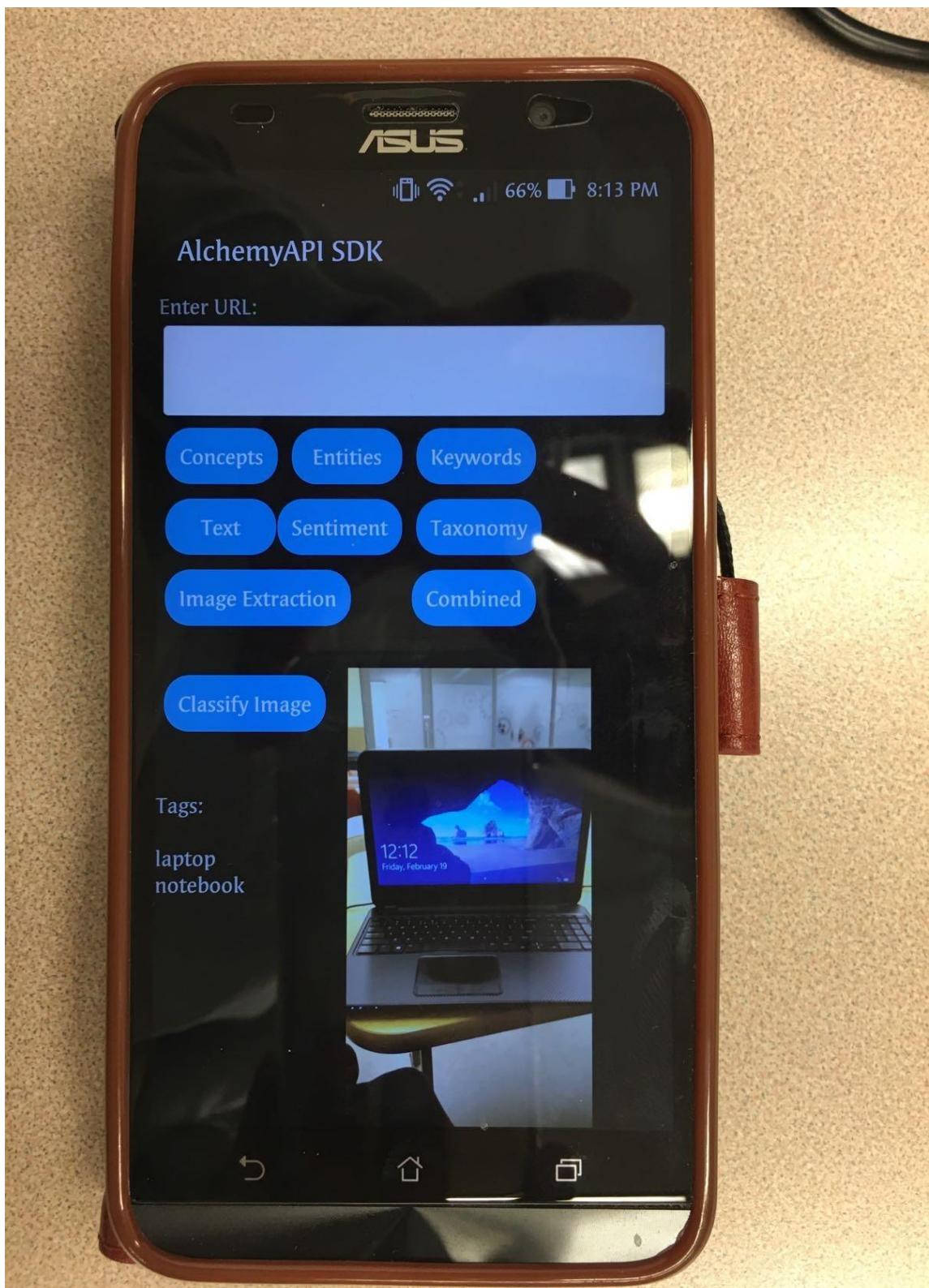
The following are the features that are developed as part of Phase II:

We had used machine learning algorithms to develop a recommendation system. In this phase we had developed two recommendation systems with which the system will be able to suggest top rated books to the user based on his interests and the furniture showrooms which could be available for cheaper prices with the location. Basically, we had provided the training data sets and the user preferences which serves as a key inputs for the system. We were also able to connect our system to the android device to which the recommendations has been sent.

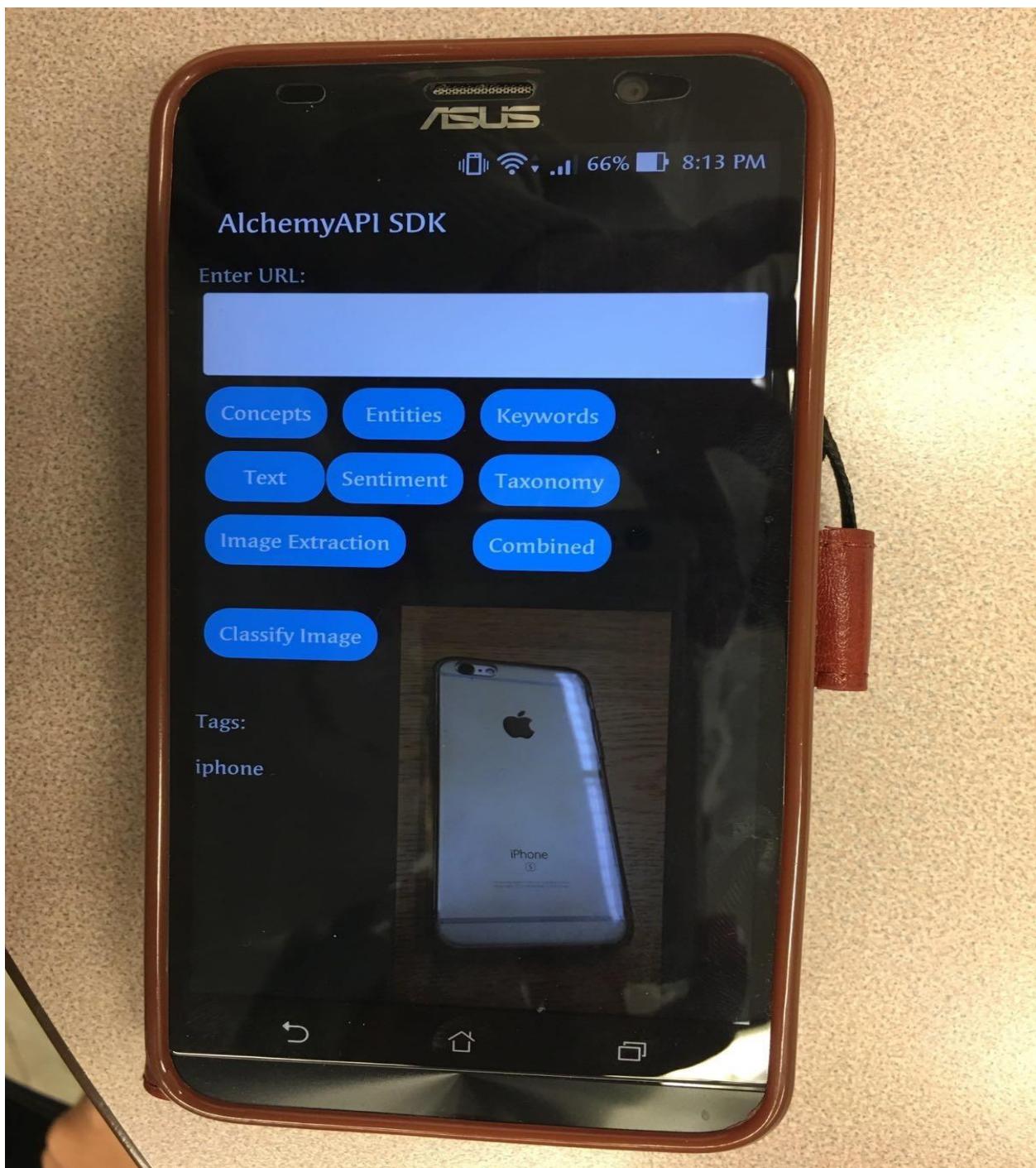
### **4.3 Implementation:**

#### **Mobile Client Implementation – Snapshots**

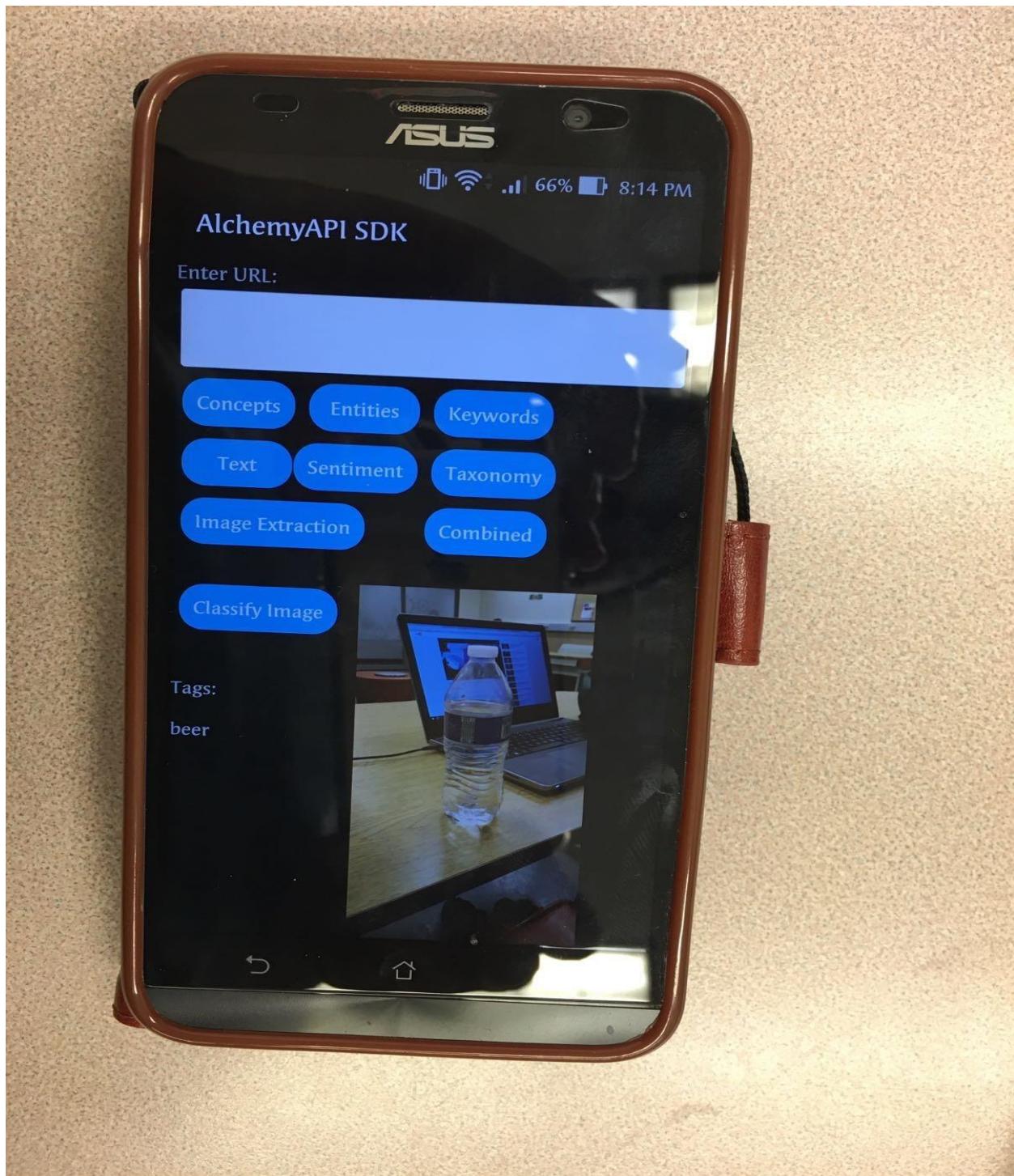
This snapshot shows us that the application is able to identify the object and names its Laptop.



This snapshot shows us that the application is able to identify the object and names it as an Iphone.



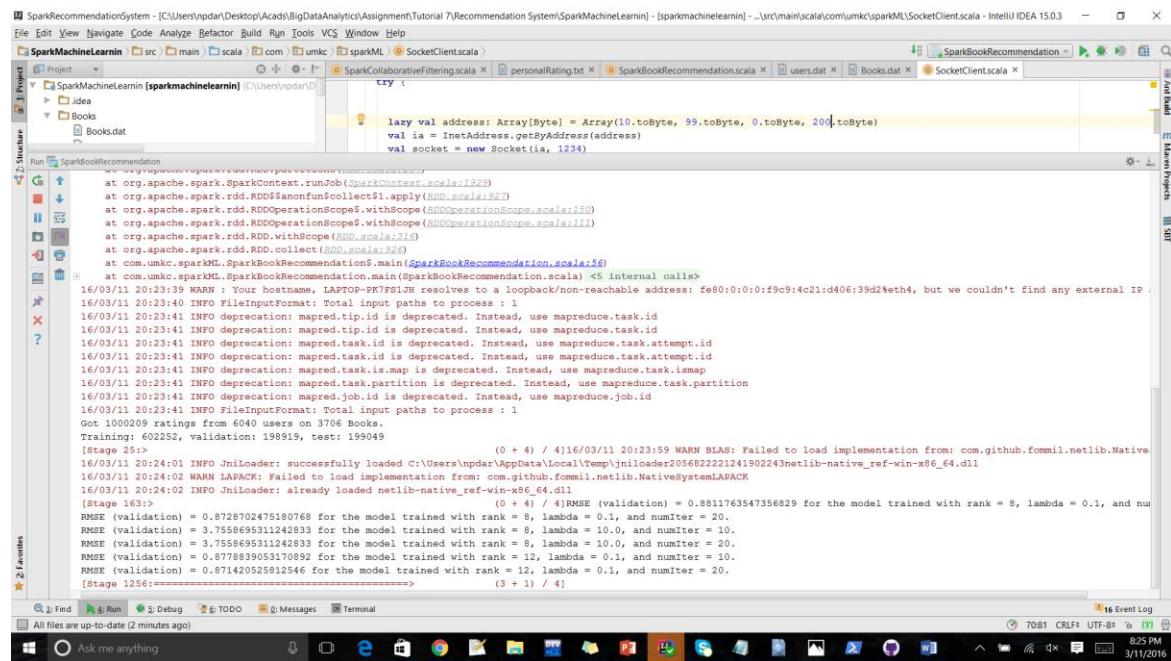
This snapshot shows us that the application is able to identify the bottle.



## Recommendation System Snapshots:

### Books Recommendation System:

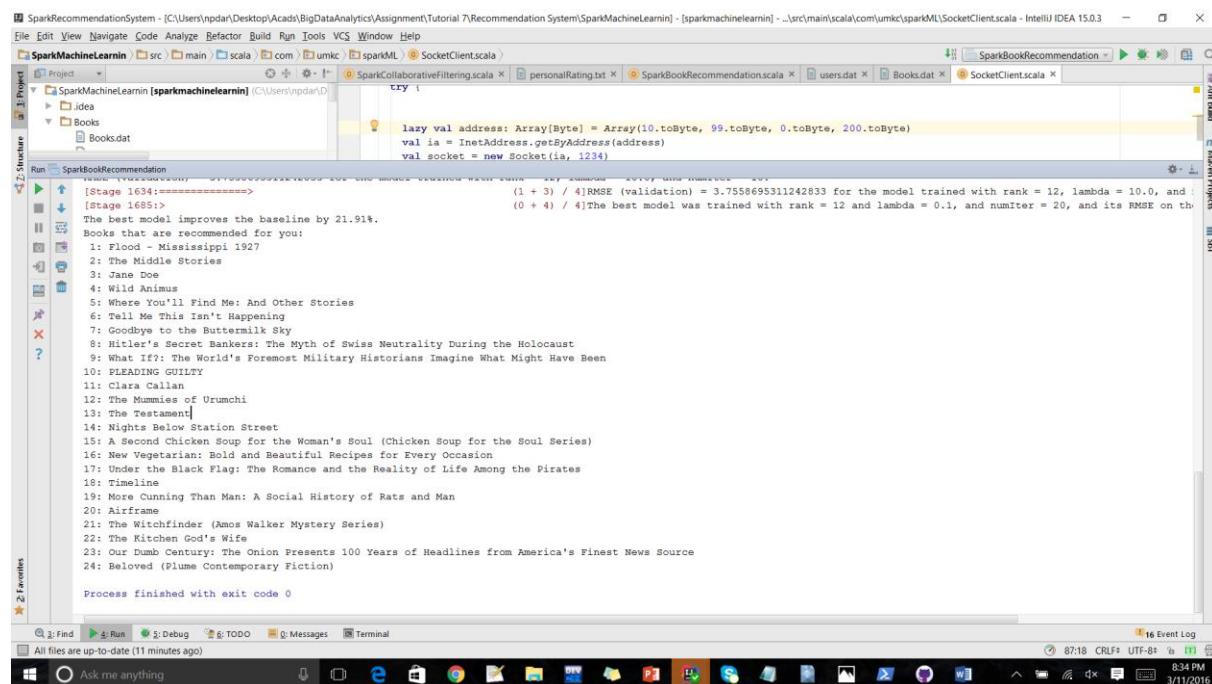
Phase at which the analyzation of training data set is taking place.



```
try {
    lazy val address: Array[Byte] = Array(10.toByte, 99.toByte, 0.toByte, 200.toByte)
    val ia = InetAddress.getByAddress(address)
    val socket = new Socket(ia, 1234)
} catch {
    case e: UnknownHostException =>
        System.out.println("WARN : Your hostname, " + e.getMessage() + " is not reachable")
        e.printStackTrace()
}

@Stage 25:> [Stage 25:=====>]
[Stage 163:> [Stage 163:=====>]
[Stage 164:> [Stage 164:=====>]
[Stage 165:> [Stage 165:=====>]
The best model improves the baseline by 21.9%.
Books that are recommended for you:
1: Flood - Mississippi 1927
2: The Middle Stories
3: Jane Doe
4: Wild Animals
5: Where You'll Find Me: And Other Stories
6: Tell Me This Isn't Happening
7: Goodbye to the Buttermilk Sky
8: Hitler's Secret Bankers: The Myth of Swiss Neutrality During the Holocaust
9: What If?: The World's Foremost Military Historians Imagine What Might Have Been
10: FEADING GUILTY
11: Clara Callan
12: The Mummies of Urumchi
13: The Testament
14: Nights Below Station Street
15: A Second Chicken Soup for the Woman's Soul (Chicken Soup for the Soul Series)
16: New Vegetarian: Bold and Beautiful Recipes for Every Occasion
17: Under the Black Flag: The Romance and the Reality of Life Among the Pirates
18: Timeline
19: More Cunning Than Man: A Social History of Rats and Man
20: Airframe
21: The Witchfinder (Amos Walker Mystery Series)
22: The Kitchen God's Wife
23: Our Dumb Century: The Onion Presents 100 Years of Headlines from America's Finest News Source
24: Beloved (Plume Contemporary Fiction)
```

### Recommended Books



```
try {
    lazy val address: Array[Byte] = Array(10.toByte, 99.toByte, 0.toByte, 200.toByte)
    val ia = InetAddress.getByAddress(address)
    val socket = new Socket(ia, 1234)
} catch {
    case e: UnknownHostException =>
        System.out.println("WARN : Your hostname, " + e.getMessage() + " is not reachable")
        e.printStackTrace()
}

@Stage 164:> [Stage 164:=====>]
[Stage 165:> [Stage 165:=====>]
The best model improves the baseline by 21.9%.
Books that are recommended for you:
1: Flood - Mississippi 1927
2: The Middle Stories
3: Jane Doe
4: Wild Animals
5: Where You'll Find Me: And Other Stories
6: Tell Me This Isn't Happening
7: Goodbye to the Buttermilk Sky
8: Hitler's Secret Bankers: The Myth of Swiss Neutrality During the Holocaust
9: What If?: The World's Foremost Military Historians Imagine What Might Have Been
10: FEADING GUILTY
11: Clara Callan
12: The Mummies of Urumchi
13: The Testament
14: Nights Below Station Street
15: A Second Chicken Soup for the Woman's Soul (Chicken Soup for the Soul Series)
16: New Vegetarian: Bold and Beautiful Recipes for Every Occasion
17: Under the Black Flag: The Romance and the Reality of Life Among the Pirates
18: Timeline
19: More Cunning Than Man: A Social History of Rats and Man
20: Airframe
21: The Witchfinder (Amos Walker Mystery Series)
22: The Kitchen God's Wife
23: Our Dumb Century: The Onion Presents 100 Years of Headlines from America's Finest News Source
24: Beloved (Plume Contemporary Fiction)

Process finished with exit code 0
```

We had also sent the recommended books to the smart phone as a notification:



## Furniture Malls Recommendation System

```
16/03/10 07:16:28 INFO FileInputFormat: Total input paths to process : 1
Got 1000209 ratings from 6040 users on 3706 FurnitureMalls.
Training: 602252, validation: 198919, test: 199049
[Stage 25:>          (0 + 4) / 4]16/03/10 07:16:25 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeDenseMatrix64F
16/03/10 07:16:28 INFO JniLoader: successfully loaded C:\Users\DEEPU\AppData\Local\Temp\jniloader1966739238328846810\netlib-native_ref-win-x86_64.dll
[Stage 27:>          (0 + 4) / 4]16/03/10 07:16:33 WARN LAPACK: Failed to load implementation from: com.github.fommil.netlib.NativeDenseMatrix64F
16/03/10 07:16:33 INFO JniLoader: already loaded netlib-native_ref-win-x86_64.dll
RMSE (validation) = 0.8815801121709103 for the model trained with rank = 8, lambda = 0.1, and numIter = 10.
RMSE (validation) = 0.8726203182715503 for the model trained with rank = 8, lambda = 0.1, and numIter = 20.
RMSE (validation) = 3.7558695311242833 for the model trained with rank = 8, lambda = 10.0, and numIter = 10.
[Stage 950:>          (0 + 4) / 4]RMSE (validation) = 3.7558695311242833 for the model trained with rank = 8, lambda = 10.0, and numIter = 20.
[Stage 1125:=====          (3 + 1) / 4]RMSE (validation) = 0.8772284010651425 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
RMSE (validation) = 0.8710227453579589 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
[Stage 1595:>          (0 + 4) / 4]RMSE (validation) = 3.7558695311242833 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 = 20, and its RMSE on the test set is 0.8690494992690084.
The best model improves the baseline by 21.95%.
```

This Screenshot shows the Furniture Malls recommended to you:

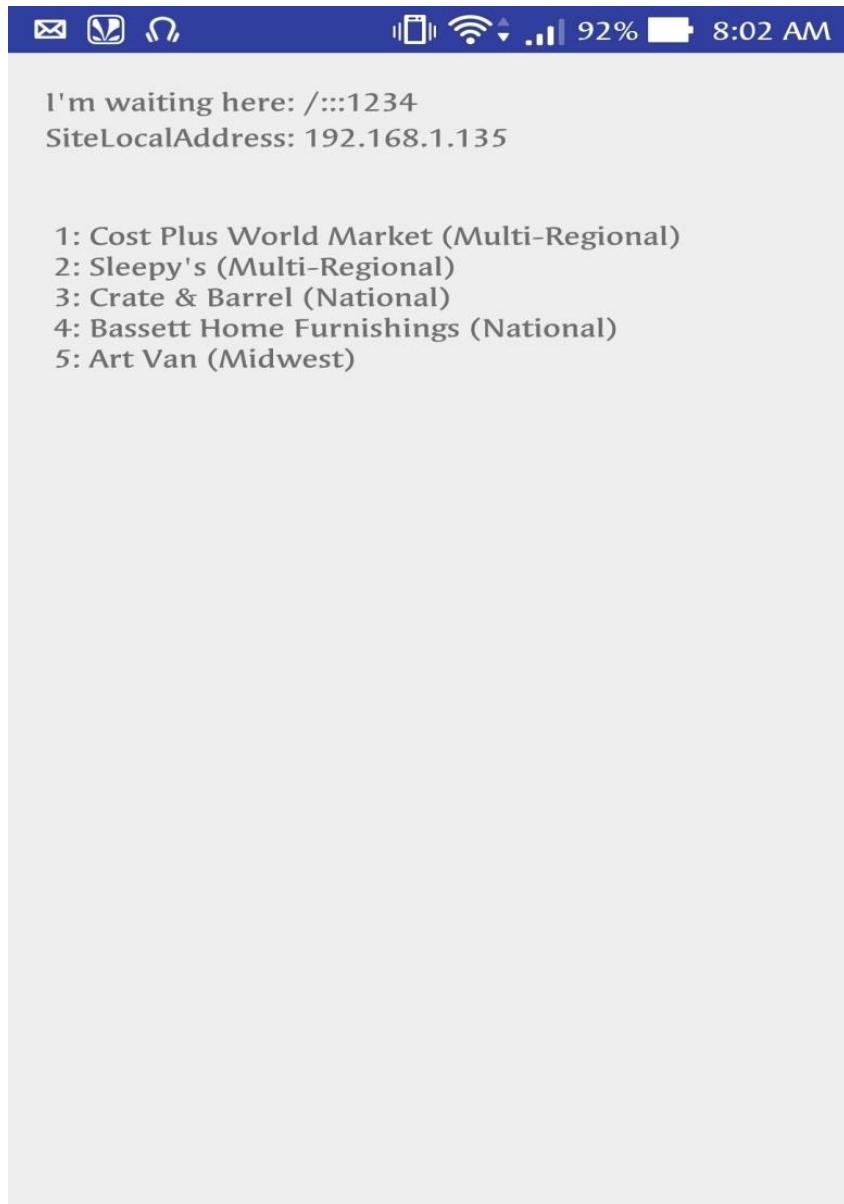
The screenshot shows the IntelliJ IDEA 15.0.1 interface with the following details:

- Project Structure:** Shows the project structure with files like `SparkCollaborativeFiltering.scala`, `SparkMovieRecommendation.scala`, and `FurnitureShopRating.txt`.
- Run Output:** Displays the Scala code and its execution output. The output includes:

  - INFO messages about deprecated methods (`mapred.task.partition`, `mapred.job.id`).
  - Total input paths: 1.
  - Training data: 1000209 ratings from 6040 users on 3706 FurnitureMalls.
  - Training parameters: Rank 8, Lambda 0.1, NumIter 10.
  - Validation RMSE: 0.8815801121709103.
  - Model selection: Stage 27 (Rank 8, Lambda 0.1, NumIter 20).
  - Test RMSE: 0.8690494992690084.
  - Improvement: Baseline improved by 21.95%.

- Bottom Status:** Shows "Process finished with exit code 0".
- Terminal:** Shows the command "Compilation completed successfully in 31s 964ms (7 minutes ago)".

The notification has been sent to the android mobile which shows the recommended furniture list.



#### **4.4 Deployment:**

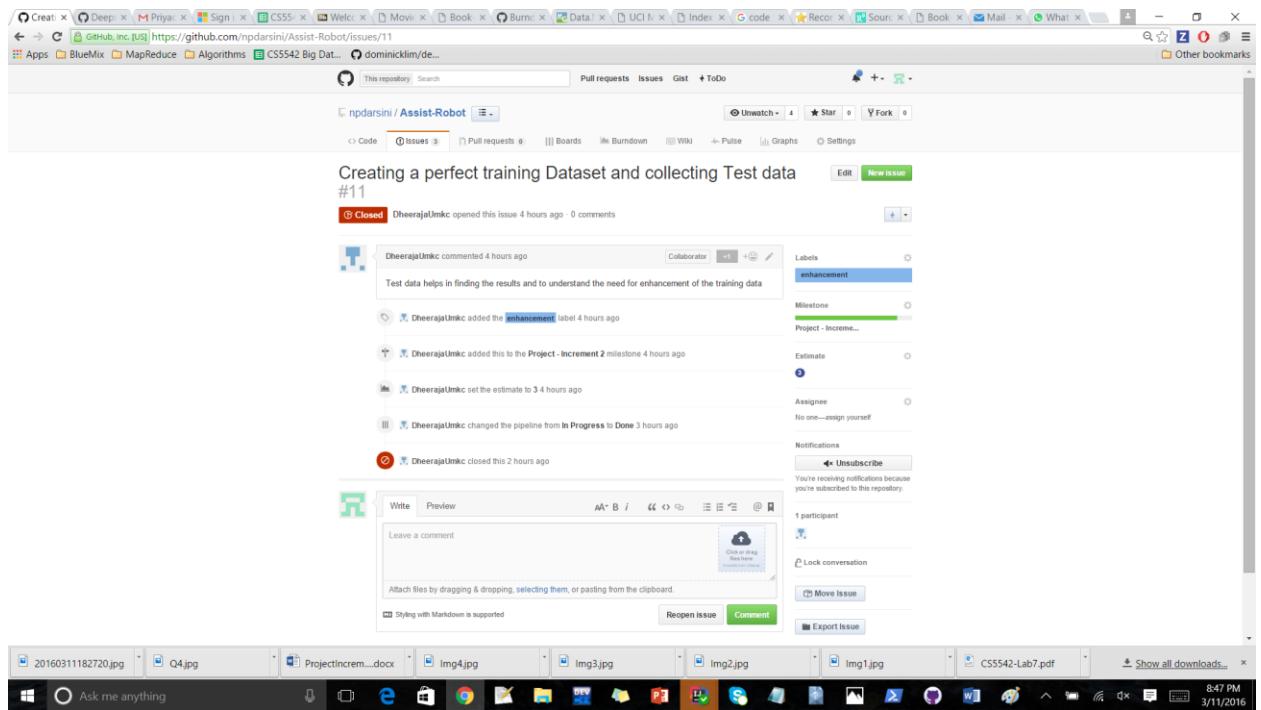
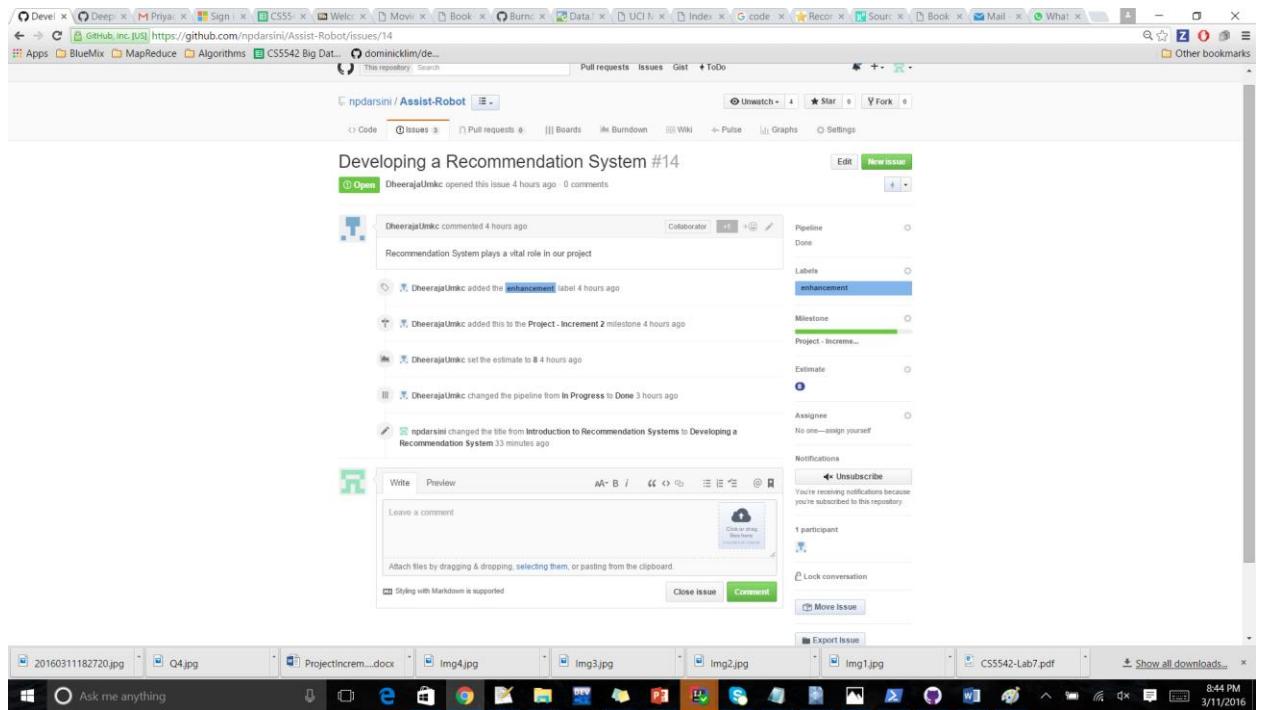
##### **Git Hub Link:**

<https://github.com/npdarsini/Assist-Robot>

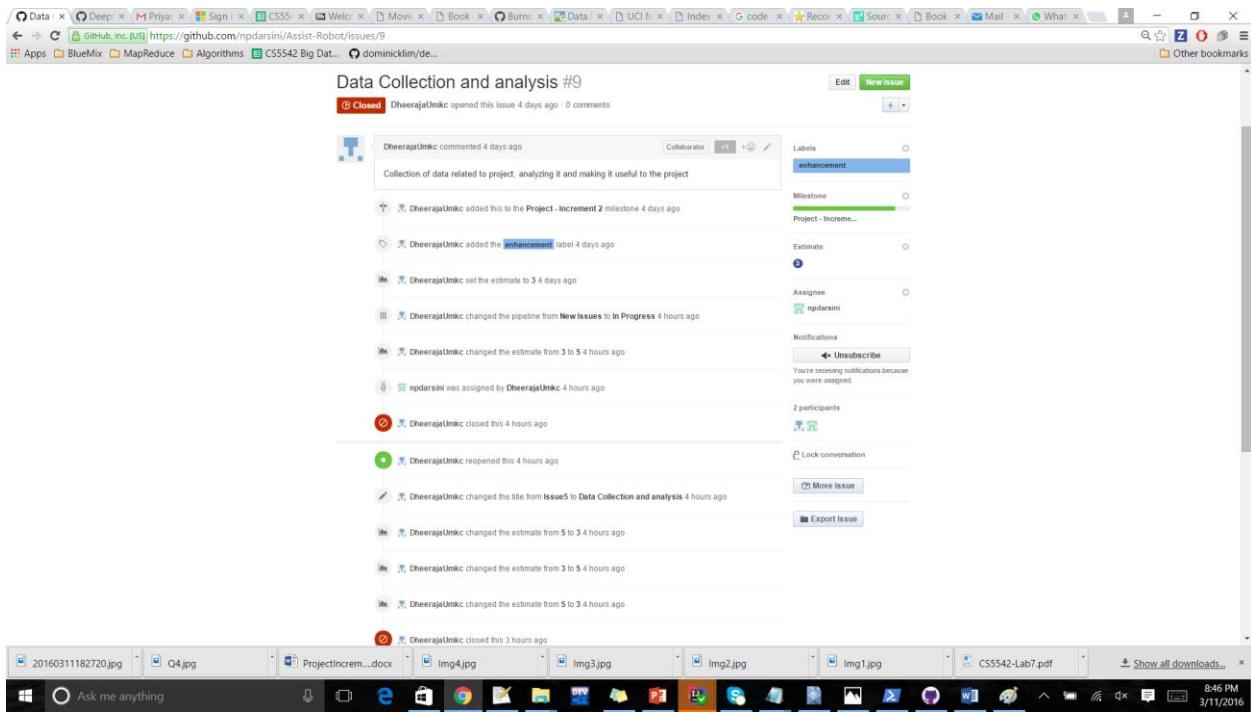
### **5. Project Management:**

#### **5.1 Implementation status report:**

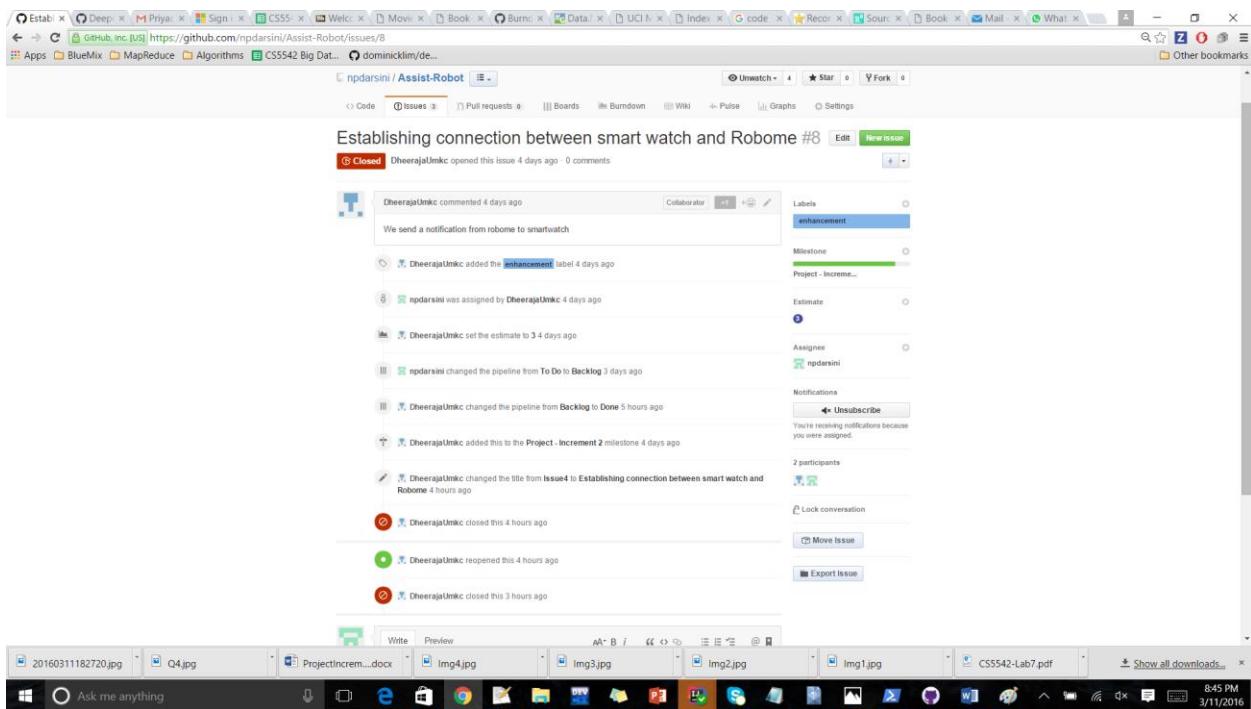
35% implementation has been implemented. This phase involves the development of recommendation systems with which the user will be notified about the top rated books and the best place to buy his/her furniture. Also the notification will be sent to the connected android device, to make the user work easier. The team members has an equal contribution towards the development and it took around 5 complete days to give an outlook for this phase. First we had tried developing a recommendation system for books, and had used this knowledge to develop the Furniture Recommendation System. Later the notification has been sent to the android device. Assuming the user will be carrying the smart android device every single time, we thought that this could be as an innovative thought as the user don't need to look for the advice in the system every time.



Firefox browser window showing a GitHub issue titled "Data Collection and analysis #9". The issue is closed by DheerajUmkc. The description states: "Collection of data related to project, analyzing it and making it useful to the project". Labels: enhancement. Milestone: Project - Increment 2. Estimate: 3 hours. Assignee: npdarsini. Notifications: You're receiving notifications because you were assigned. Participants: 2. Conversation history shows multiple comments from DheerajUmkc and npdarsini, including changes in title, estimate, and pipeline status.



Firefox browser window showing a GitHub issue titled "Establishing connection between smart watch and Robome #8". The issue is closed by DheerajUmkc. The description states: "We send a notification from robome to smartwatch". Labels: enhancement. Milestone: Project - Increment 2. Estimate: 3 hours. Assignee: npdarsini. Notifications: You're receiving notifications because you were assigned. Participants: 2. The conversation history shows multiple comments from DheerajUmkc and npdarsini, including changes in title, estimate, and pipeline status.



Installation of Spark #7

**Closed** Dheeraj@Umkc opened this issue 4 days ago · 0 comments

Dheeraj@Umkc commented 4 days ago Since usage of spark makes the project more flexible because of availability of RDD's

Dheeraj@Umkc added the enhancement label 4 days ago

dunil210 was assigned by Dheeraj@Umkc 4 days ago

Dheeraj@Umkc added this to the Project - increment 2 milestone 4 days ago

Dheeraj@Umkc set the estimate to 3-4 days ago

npdarsini changed the pipeline from To Do to Backlog 3 days ago

Dheeraj@Umkc changed the title from Issue3 to Installation of Spark 4 hours ago

dunil210 was unassigned by Dheeraj@Umkc 4 hours ago

Dheeraj@Umkc changed the pipeline from Backlog to Done 4 hours ago

Deepur123start was assigned by Dheeraj@Umkc 4 hours ago

Dheeraj@Umkc closed this 4 hours ago

Dheeraj@Umkc reopened this 4 hours ago

Dheeraj@Umkc closed this 4 hours ago

The screenshot shows a GitHub issue page for a project titled 'Installation of Spark'. The issue is labeled as 'Closed' and was opened by 'Dheeraj@Umkc' 4 days ago. The description of the issue is: 'Since usage of spark makes the project more flexible because of availability of RDD's'. The issue has been updated several times, with the most recent update being a closure 4 hours ago. The GitHub interface includes a sidebar with labels like 'enhancement', 'Milestone', 'Project - increment...', 'Estimate', 'Assignee', and 'Notifications'. The bottom of the screen shows a Windows taskbar with various open files and a system clock indicating 8:45 PM on 3/11/2016.

Features of RoboMe #6

**Closed** Dheeraj@Umkc opened this issue 4 days ago · 0 comments

Dheeraj@Umkc commented 4 days ago Going through the features of Robome and what all can be derived using its basic features

Dheeraj@Umkc added enhancement, question labels 4 days ago

Deepur123start was assigned by Dheeraj@Umkc 4 days ago

Dheeraj@Umkc added this to the Project - increment 2 milestone 4 days ago

Dheeraj@Umkc set the estimate to 5-4 days ago

npdarsini changed the pipeline from In Progress to To Do 3 days ago

npdarsini changed the pipeline from To Do to In Progress 3 days ago

Dheeraj@Umkc changed the pipeline from In Progress to Done 5 hours ago

Dheeraj@Umkc removed the enhancement label 4 hours ago

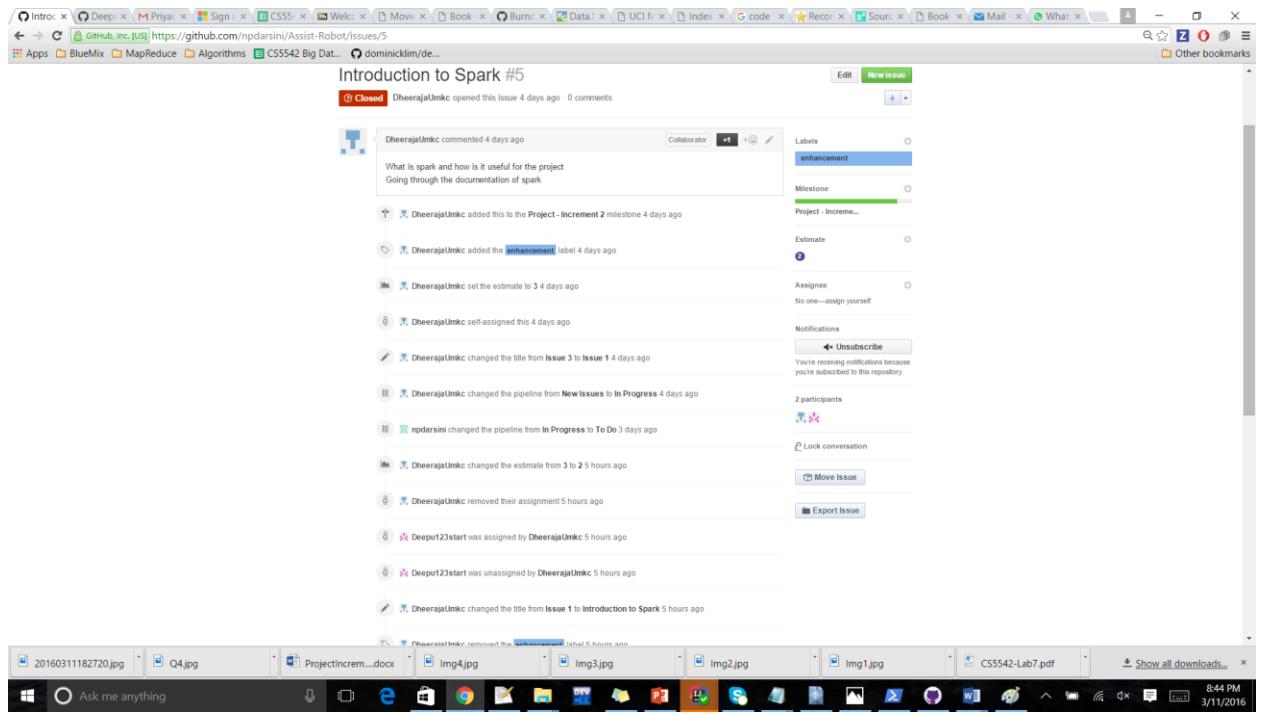
Deepur123start was unassigned by Dheeraj@Umkc 4 hours ago

Dheeraj@Umkc changed the estimate from 5 to 1 hours ago

npdarsini was assigned by Dheeraj@Umkc 4 hours ago

npdarsini was unassigned by Dheeraj@Umkc 4 hours ago

The screenshot shows a GitHub issue page for a project titled 'Features of RoboMe'. The issue is labeled as 'Closed' and was opened by 'Dheeraj@Umkc' 4 days ago. The description of the issue is: 'Going through the features of Robome and what all can be derived using its basic features'. The issue has been updated several times, with the most recent update being a closure 4 hours ago. The GitHub interface includes a sidebar with labels like 'question', 'Milestone', 'Project - increment...', 'Estimate', 'Assignee', and 'Notifications'. The bottom of the screen shows a Windows taskbar with various open files and a system clock indicating 8:45 PM on 3/11/2016.



## Bibliography:

Lab Tutorials and the material provided by Dr. Lee.

# Assist Robot

Phase III

Project Report

**Project Team – 8**

Team Members

Priyadarsini Nidadavolu(17)

Deepthi Priyadarshini Penmetsa(22)

Dheeraja Vallabhaneni(28)

Tej Kumar Yentrapragada(33)

## Table of Contents

Section	Description
1	Introduction
2	Project Goal and Objectives
3	Project Plan
	3.1 Schedule
	3.2 Project Timelines
	3.2.1 Team Members
	3.2.2 Tasks and Responsibilities
	3.3 Burndown Chart
4	Increment Report
	4.1 Increment based on Phases
	4.1.1 Phase 1 - Existing API
	4.1.2 Phase 2 - Recommendation System
	4.1.3 Phase 3 – Image Classification System
	4.2 Design of Features
	4.3 Implementation
	4.4 Deployment
5	Project Management
	5.1 Implementation Status Reports
6	Bibliography

## **1. Introduction:**

The main goal of the project is to help people in finding their misplaced objects. Basically humans have a tendency to forget their belongings somewhere in their house and search for it for hours together. For example, if I have an important business meeting to attend, but I don't remember where I placed my car keys, then I will be in huge loss. So to prevail in these circumstances here comes our Friendly Robot- My Friend which could assist me in keeping track of my personal things. So what this robot will do is that it will have entire map (laser scan) of the building and objects in the building in its memory. So we will feed the robot with the objects that are highly important to us, like car keys, some files, phone and laptop. This robot will keep its eye on these objects and notify their location to its master upon request. Additionally our Robot who will be an eFriend who will help us to choose the furniture to our home. Also our robot will suggest us the top rated books.

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

The primary goals of our project is described below:

- To implement a module which has an interaction with robot. Eg: You can ask few questions to the robot and the Robot will be responding to you back. You can ask the robot about your misplaced phone. So that it will answer you after it had found the phone.
- To make the robot learn about the personal items like chargers, phones, watch, keys etc.
- To design a robot which can find the learned objects that are misplaced in a building.
- To build a recommendation system which will be able to recommend the list of books which are rated high and are related to our interests.
- To build a recommendation system which will suggest us about the latest furniture details, their quality and from which brand/ shop we could purchase from. This feature enables us to decorate our houses with rich interior designing.
- To send a notification to your smart watch when it finds the lost object.
- To remind the user about his day to day events that were previously taught.
- To make the robot act as an assistant in getting things specified by the user. (Mr Robot – Get me my phone).

- To make a single robot act as assistant to all the people living in same house. It recognizes the user first and then assists that particular user in finding the belongings.

### 3. Project Plan:

#### 3.1 Schedule:

**Stories:** Four user stories had been created as part of Iteration 1. Here are the snapshots for the stories which are in closed and opened state.

The screenshot shows the GitHub Issues page for the repository 'npdarsini / Assist-Robot'. The 'Issues' tab is selected. There are two open issues listed:

- ① Test - Different Objects enhancement
- ② Capture and feed the Images enhancement

Both issues were opened 35 minutes ago by npdarsini. The interface includes filters, labels, milestones, and a 'New issue' button.



The screenshot shows the GitHub Issues page for the repository 'npdarsini / Assist-Robot'. The 'Issues' tab is selected. There are two closed issues listed:

- ① Smart Watch connection with Smart Phone enhancement
- ② Smart Phone connection with Android Studio enhancement

Both issues were opened 49 minutes ago by npdarsini. The interface includes filters, labels, milestones, and a 'New issue' button.



**Stories:** Eight user issues had been created as part of Iteration 2. Here are the snapshots for the stories which are in closed and opened state.

Issues | Pull requests | Issues | Gist | ToDo

npdarsini / Assist-Robot

Issues

Filters: Is issue milestone: "Project - Increment 2" is closed

1 Open ✓ 7 Closed

- Creating a perfect training Dataset and collecting Test data (enhancement)
- Usage of Speech and image services (enhancement)
- Data Collection and analysis (enhancement)
- Establishing connection between smart watch and Robome (enhancement)
- Installation of Spark (enhancement)
- Features of Robome (question)
- Introduction to Spark (enhancement)

ProTip! Adding no label will show everything without a label.

Issues | Pull requests | Issues | Gist | ToDo

npdarsini / Assist-Robot

Issues

Filters: Is issue milestone: "Project - Increment 2" is open

1 Open ✓ 7 Closed

- Developing a Recommendation System (enhancement)

ProTip! Mix and match filters to narrow down what you're looking for.

**Stories:** Six user issues had been created as part of Iteration 3. Here are the snapshots for the stories which are in closed and opened state.

The screenshot shows the GitHub Issues page for the repository npdarsini/Assist-Robot. There are 17 issues listed:

- Notification to Android Device - Recognized Object (enhancement) #18 (closed)
- Training data sets - Creation with different objects (enhancement) #17 (closed)
- Exploring different Classification algorithms - Random Forest, Decision Tree (enhancement) #16 (closed)
- Image classification (enhancement) #15 (closed)
- Developing a Recommendation System (enhancement) #14 (closed)
- Introduction to ML Algorithm (enhancement) #13 (closed)
- Creating a perfect training Dataset and collecting Test data (enhancement) #11 (closed)
- Usage of Speech and image services (enhancement) #10 (closed)
- Data Collection and analysis (enhancement) #9 (closed)
- Establishing connection between smart watch and Robome (enhancement) #8 (closed)
- Linking Modules (enhancement) #20 (open)
- Classification based on the Streaming Data (enhancement) #19 (open)
- Streaming Data Collection (enhancement) #12 (open)

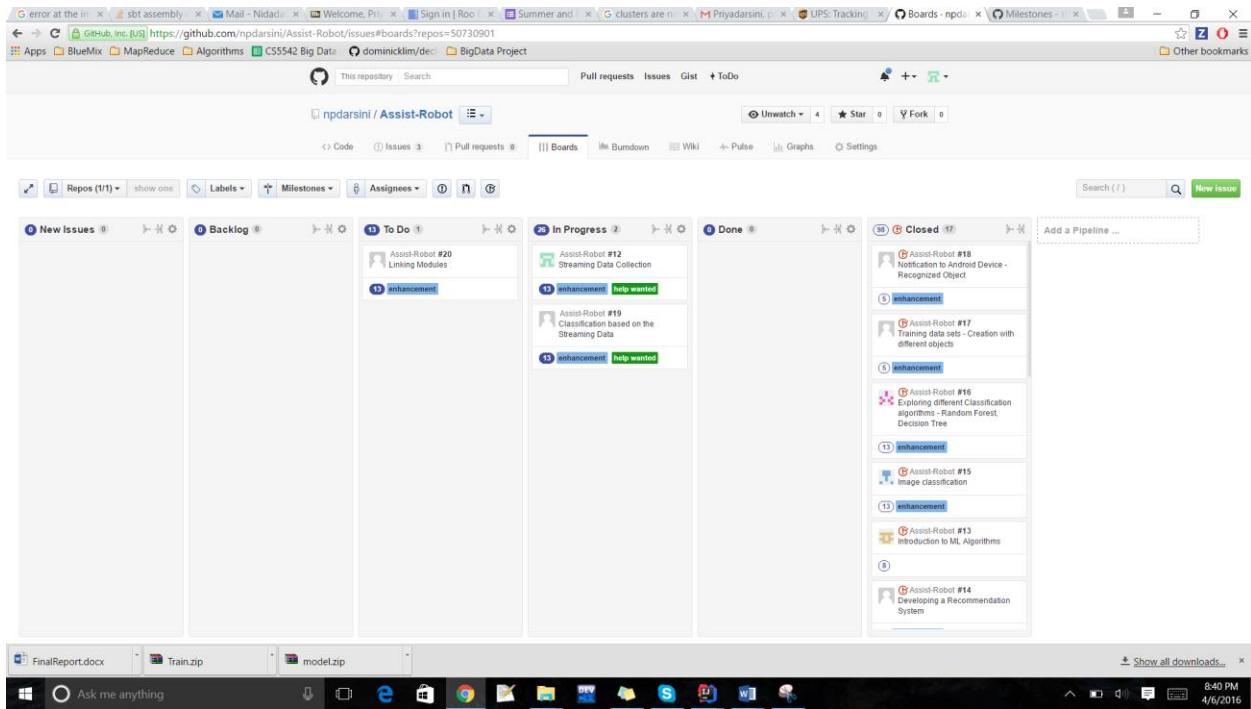
The screenshot shows the GitHub Issues page for the repository npdarsini/Assist-Robot. There are 3 open issues:

- Linking Modules (enhancement) #20
- Classification based on the Streaming Data (enhancement) #19
- Streaming Data Collection (enhancement) #12

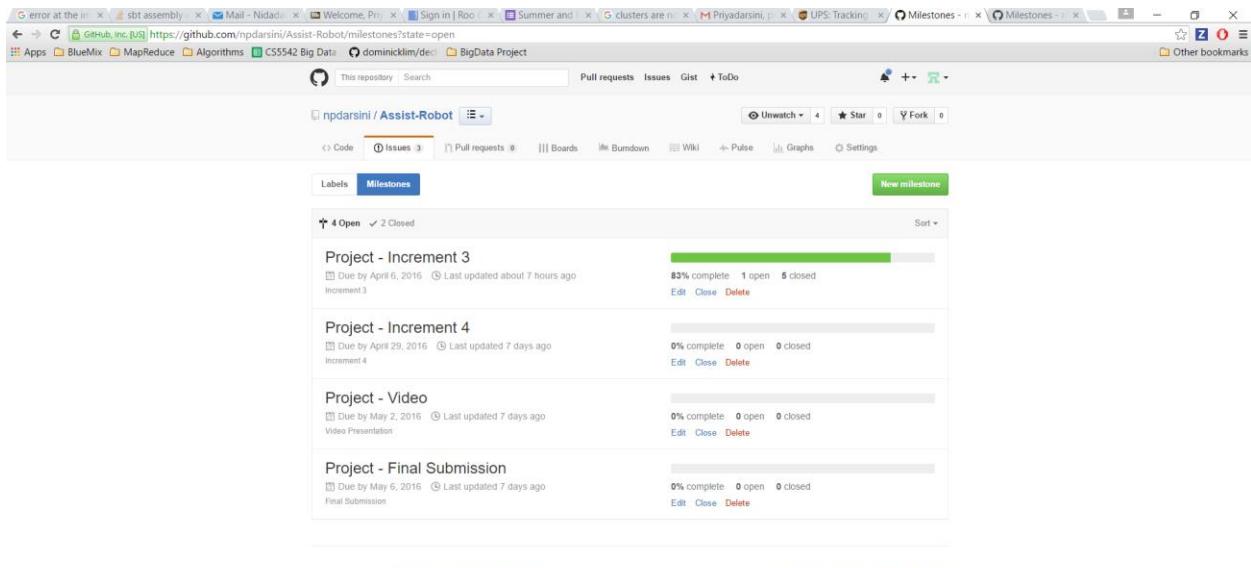
The screenshot shows a Windows taskbar with three open files:

- FinalReport.docx
- Train.zip
- modelZip

## Board:

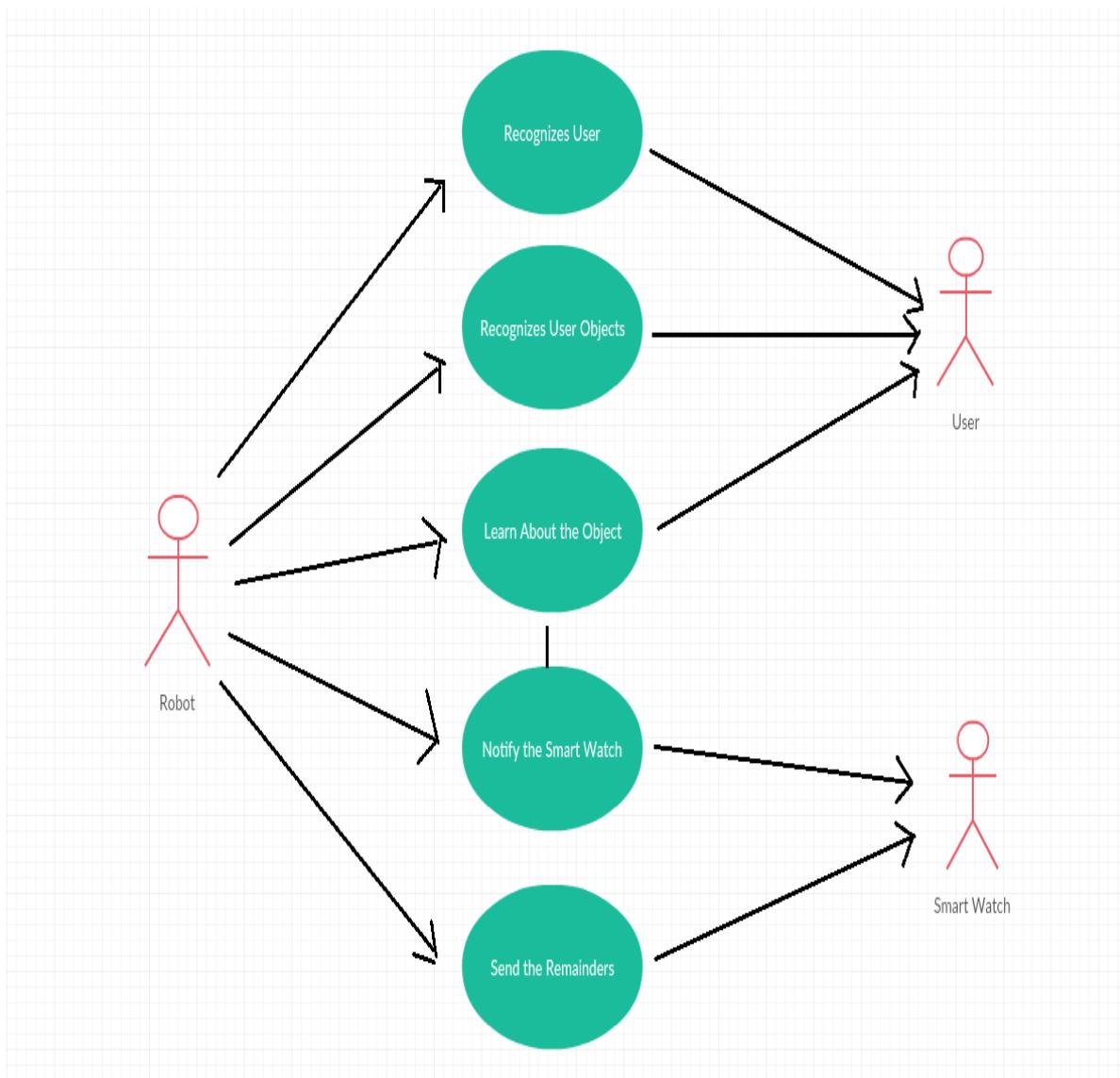


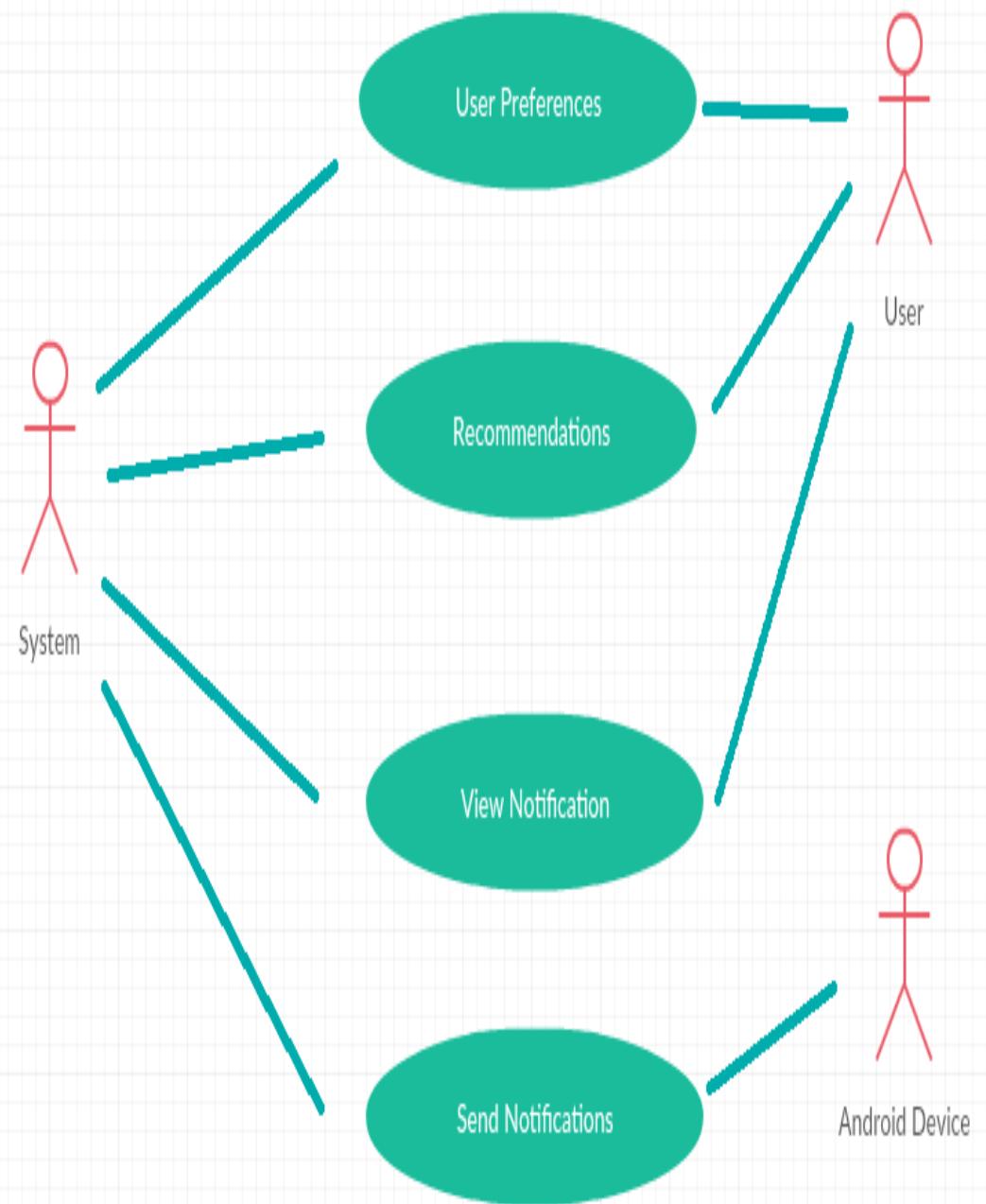
## Milestones:



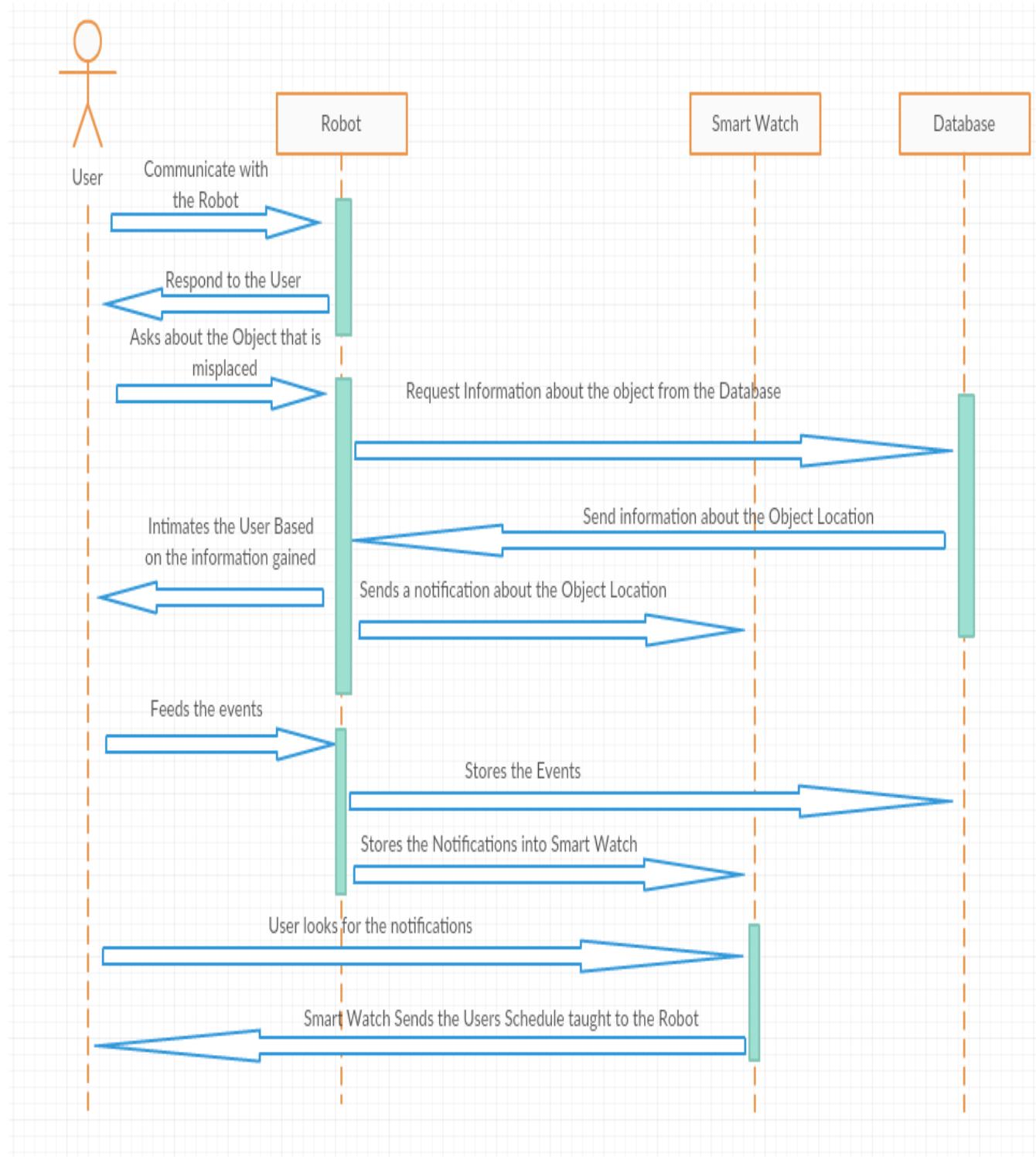
### 3.1.1 UML Diagrams:

#### Use Case Diagram

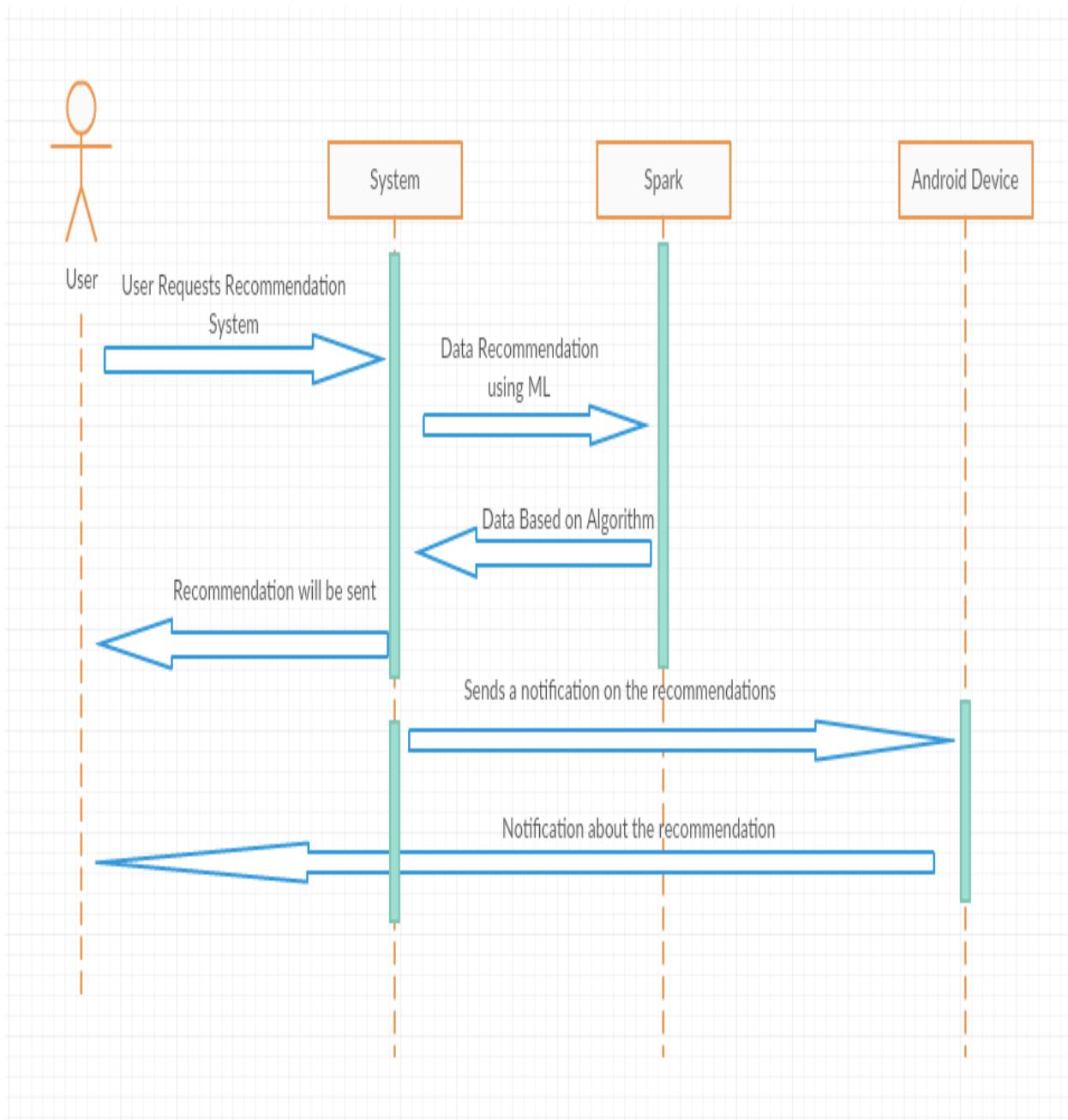




## Sequence Diagram:



## Sequence Diagram for Recommendation System



### **3.2 Project Timelines:**

Increment	Deadline
<b>Increment 1</b>	19 February 2015
<b>Increment 2</b>	11 March 2016
<b>Increment 3</b>	6 April 2016
<b>Increment 4</b>	29 April 2016
<b>Final Submission</b>	6 May 2016

#### **3.2.1 Team Members:**

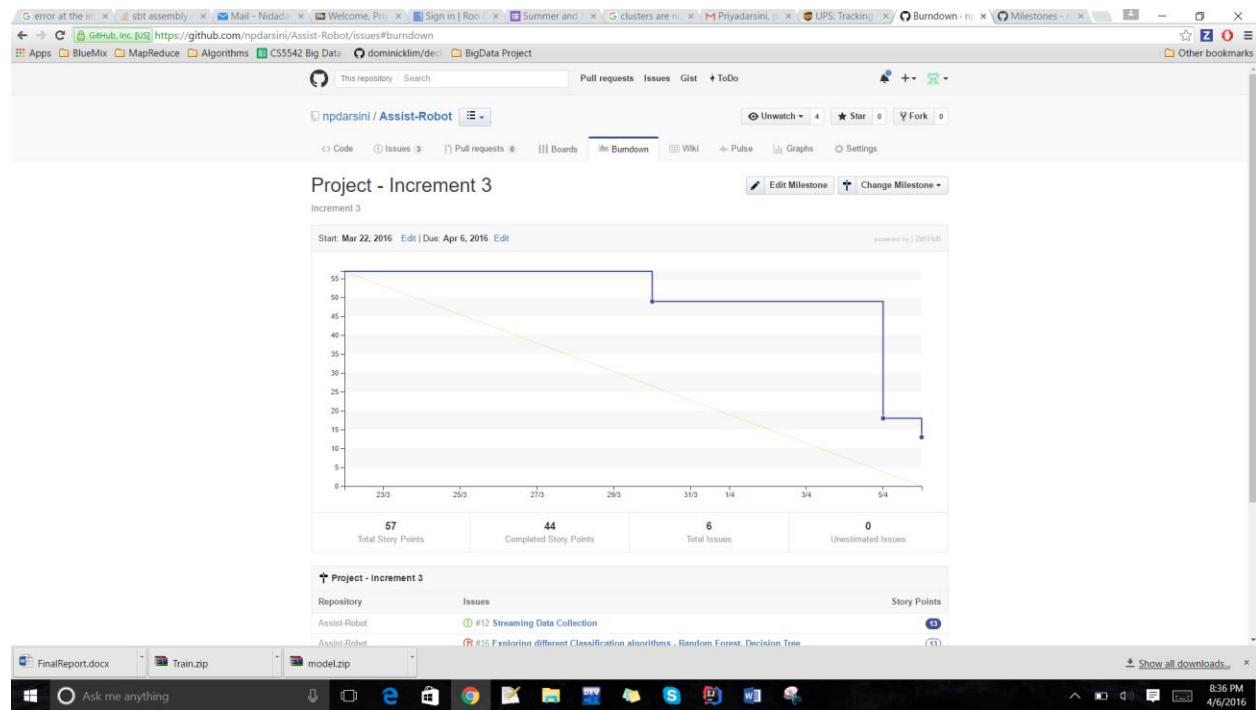
1. Priyadarsini Nidadavolu – 17
2. Deepthi Priyadarshini Penmetsa – 22
3. Dheeraja Vallabhaneni – 28
4. Tej Kumar Yentrapragada – 33

#### **3.2.2 Tasks and Responsibilities:**

- **Machine Learning and R Programming** – Deepthi Priyadarshini Penmetsa
- **Spark and Hadoop Technologies** – Priyadarsini Nidadavolu
- **Objective C and IOS Programming** – Tej Kumar Yentrapragada
- **Android Programming** - Dheeraja Vallabhaneni

### **3.3 Burndown Chart:**

#### **Burndown:**



## **4. Increment Report**

### **4.1 Incremental Explanations**

#### **4.1.1 Phase 1 -Existing API:**

##### **IBM Alchemy API**

This API basically performs machine learning and natural language processing techniques. Some of its features include semantic text analysis, sentimental analysis, deep learning, face detection and reorganization, speech to text and vice versa conversions etc. In this we had used this API in order to recognize the objects that we want to teach the Robot.

Achievements upon using this API – The Robot could identify basic objects like laptop, phone, bottle etc.

#### **4.1.2 Phase 2 - Recommendation System:**

In this phase we had developed two recommendation systems which can recommend the user about the popular furniture showrooms and the famous books. In this we provided the training set with user information (uid, name, ratings etc.,), furniture information (list of showrooms, location) and the book information (name, author etc.,).

The recommended notification has been sent to the android device (smart watch/phone) using Spark-Android Socket programming techniques.

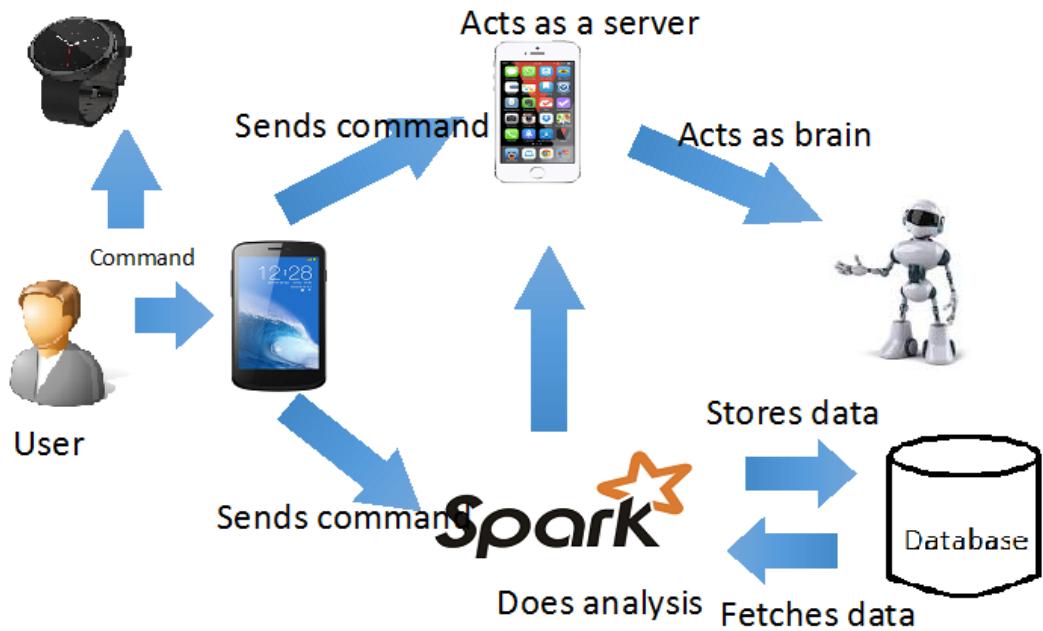
#### **4.1.3 Phase 3 – Image Classification:**

In this phase we had implemented Image Classification system using Random Forest Machine Learning Classifier Algorithm. In this we had provided the training data set which has the different kinds of objects like keys, charger, watch, spectacles, phone. Basically, we generated the key descriptors and created the clusters and histograms out of it. Based on this features, the classifier predicts the image from the testing data set that has been provided.

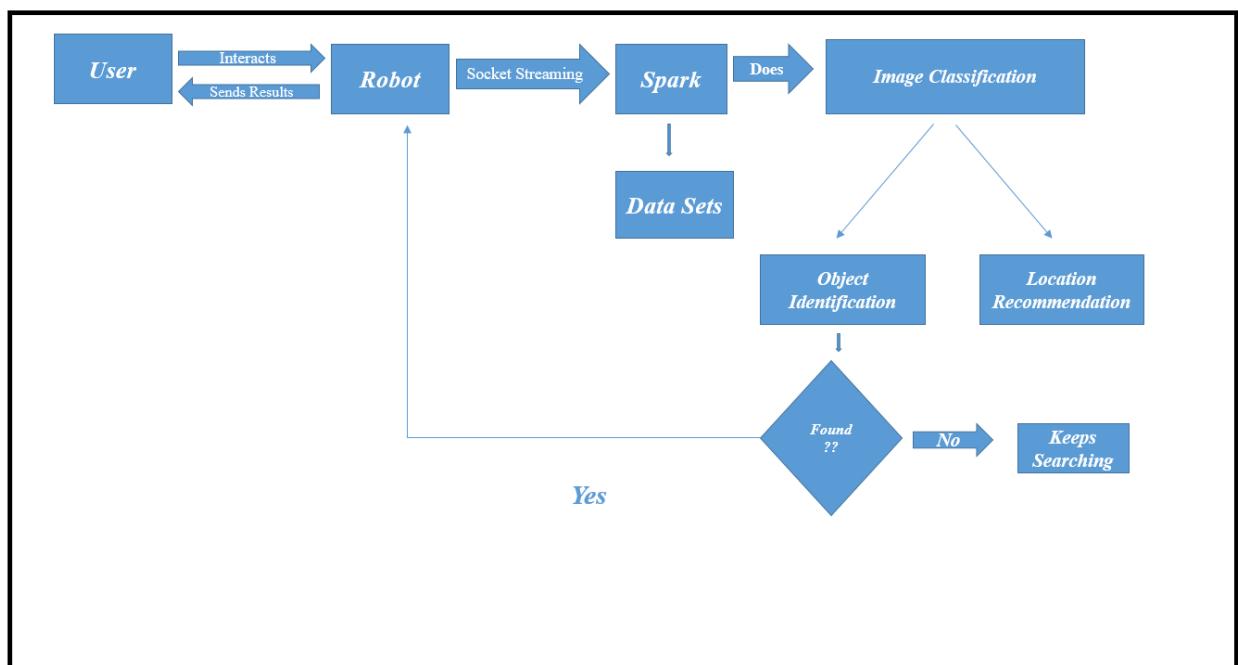
The notification about the classified image will be sent to the android device (smart watch/phone) using Spark-Android Socket programming techniques.

### **4.2 Design of Features:**

The architecture of our system could be like the user can give commands to the client device which is android phone. Further the Iphone which acts as a server could take commands from the android phone and passes it to the Robot. The Robot performs the necessary actions of the received command and return back to the Android Phone. It also sends the notifications to the Android Smart Watch. The Android device can also pass the command to the Spark and fetch the data from the database (MongoDB, Hadoop DB). Our system will be able to recommend the user based on the trained data sets and the notifications will be sent to the android device.



### Work Flow:



## **System Features**

The following are the features that were developed as part of Phase I:

We had used IBM's Alchemy API and able to make our Robot to detect the object and return the object name as a result.

The following are the features that are developed as part of Phase II:

We had used machine learning algorithms to develop a recommendation system. In this phase we had developed two recommendation systems with which the system will be able to suggest top rated books to the user based on his interests and the furniture showrooms which could be available for cheaper prices with the location. Basically, we had provided the training data sets and the user preferences which serves as a key inputs for the system. We were also able to connect our system to the android device to which the recommendations has been sent.

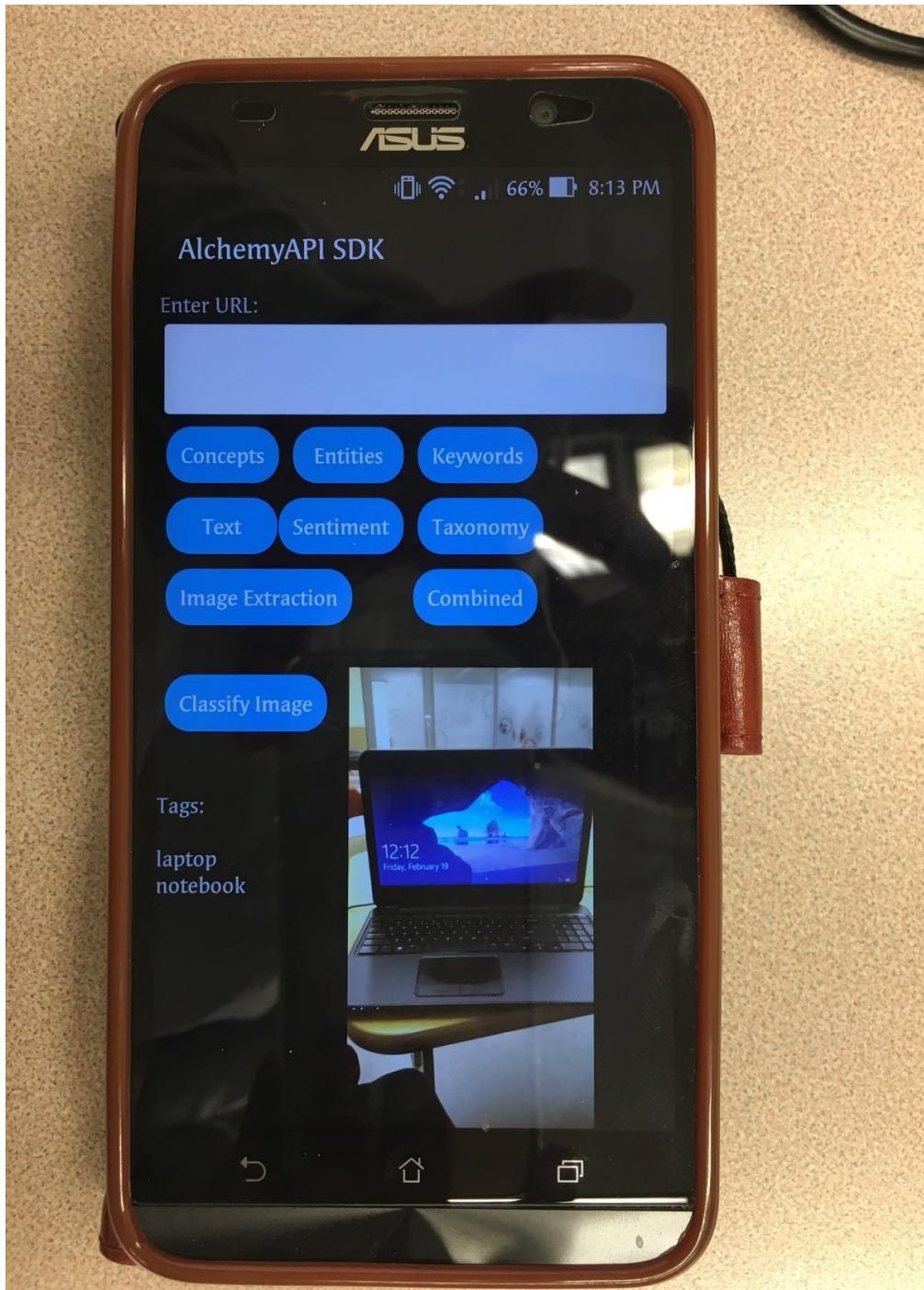
The following are the features that are developed as part of Phase III:

We had used Random Forest Classifier Algorithm to develop an Image classification system. In this phase we had provided the system with the training data set which consists the sample images of the different types of objects like keys, watch, spectacles, phone etc. We are able to create the clusters and histograms out of the provided data and was able to send a notification to the smart device about the object that has been predicted.

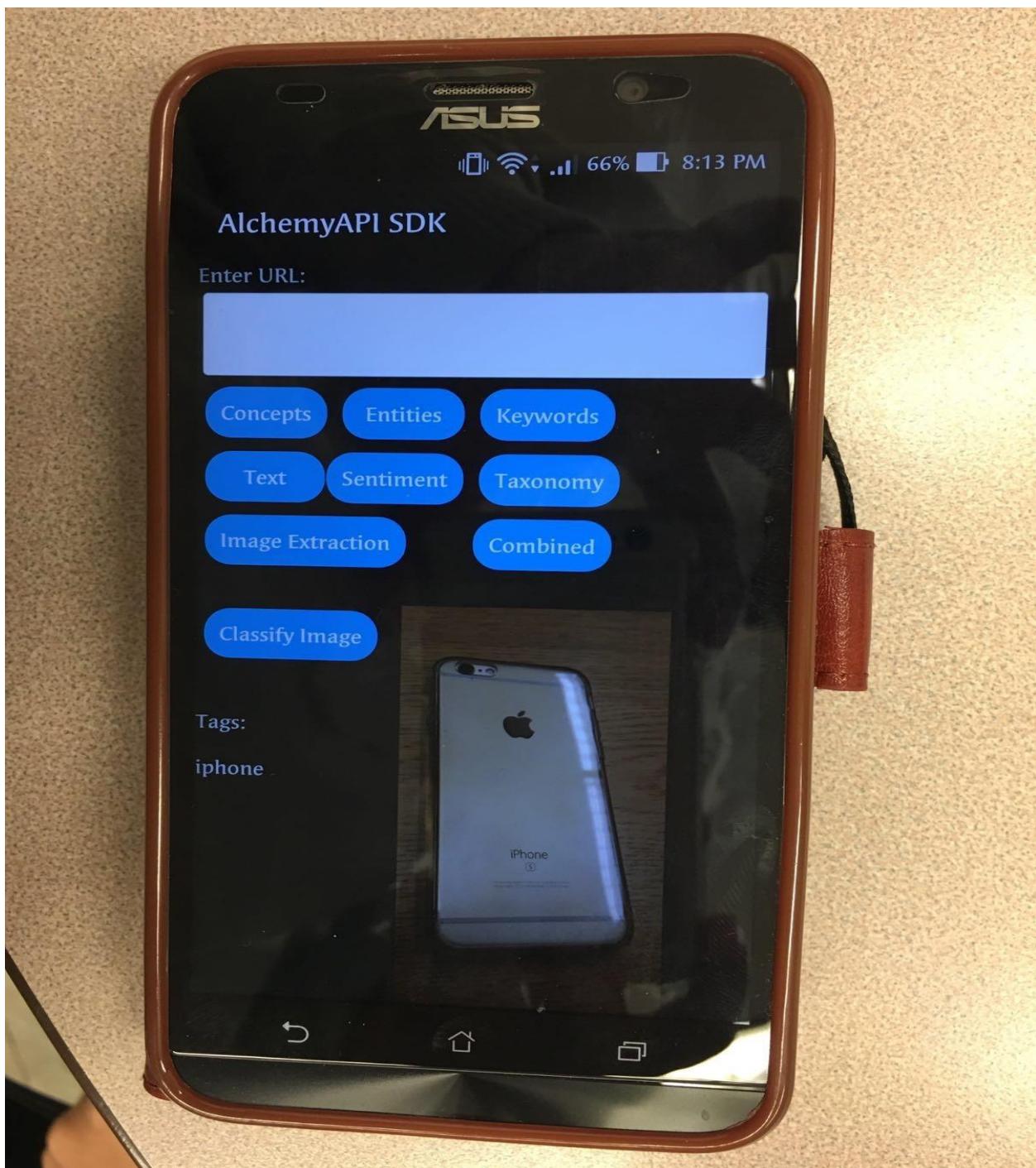
## 4.3 Implementation:

### Mobile Client Implementation – Snapshots

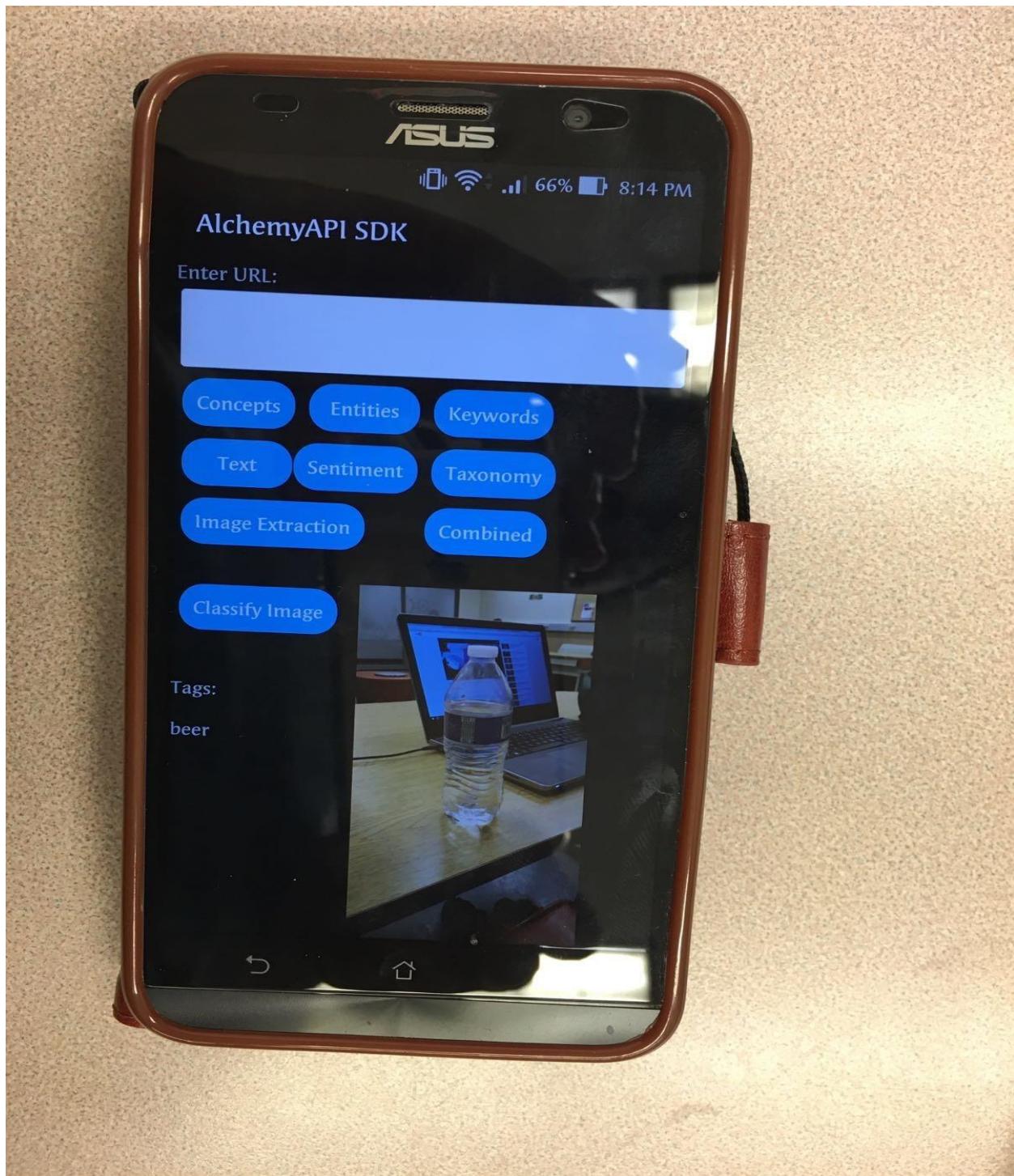
This snapshot shows us that the application is able to identify the object and names its Laptop.



This snapshot shows us that the application is able to identify the object and names it as an Iphone.



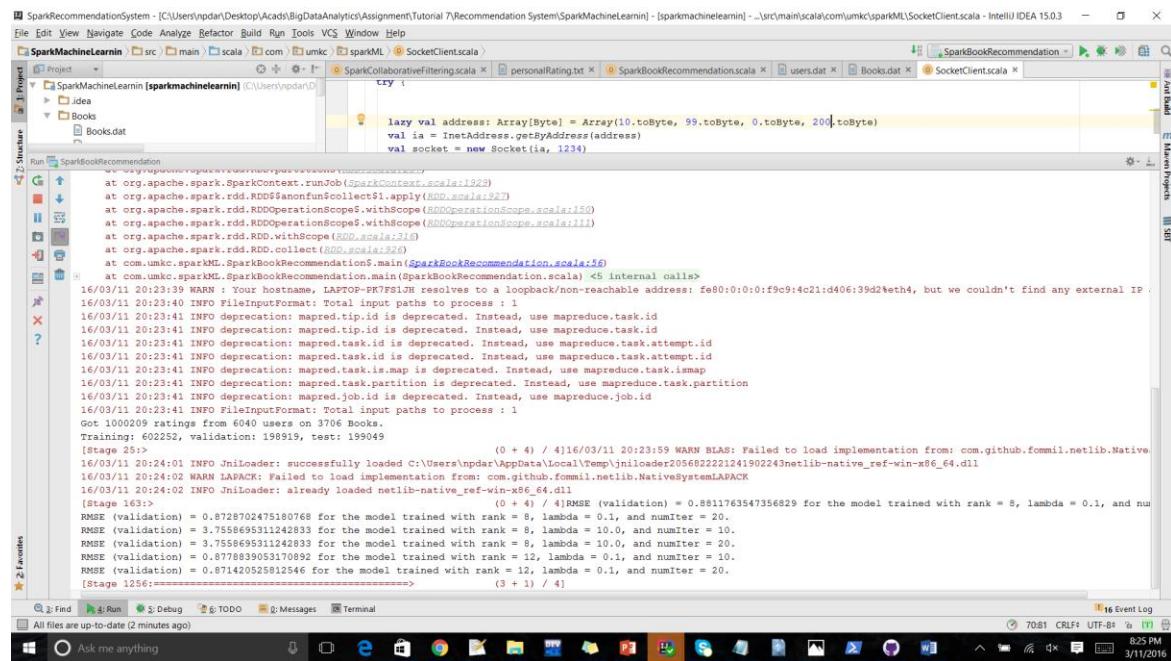
This snapshot shows us that the application is able to identify the bottle.



## Recommendation System Snapshots:

### Books Recommendation System:

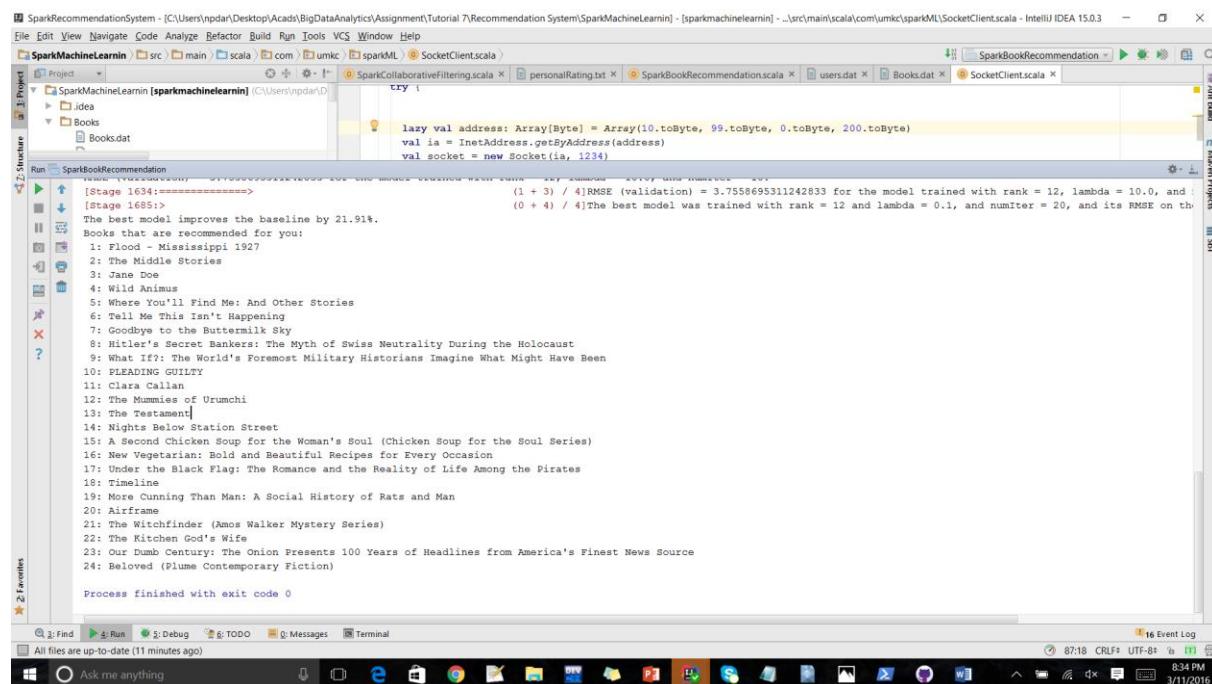
Phase at which the analyzation of training data set is taking place.



```
try {
    lazy val address: Array[Byte] = Array(10.toByte, 99.toByte, 0.toByte, 200.toByte)
    val ia = InetAddress.getByAddress(address)
    val socket = new Socket(ia, 1234)
} catch {
    case e: UnknownHostException =>
        System.out.println("WARN : Your hostname, " + e.getMessage() + " is not reachable")
        e.printStackTrace()
}

@Stage 25:> [Stage 25:=====>]
[Stage 163:> [Stage 163:=====>]
[Stage 164:> [Stage 164:=====>]
[Stage 165:> [Stage 165:=====>]
The best model improves the baseline by 21.9%.
Books that are recommended for you:
1: Flood - Mississippi 1927
2: The Middle Stories
3: Jane Doe
4: Wild Animals
5: Where You'll Find Me: And Other Stories
6: Tell Me This Isn't Happening
7: Goodbye to the Buttermilk Sky
8: Hitler's Secret Bankers: The Myth of Swiss Neutrality During the Holocaust
9: What If?: The World's Foremost Military Historians Imagine What Might Have Been
10: FEADING GUILTY
11: Clara Callan
12: The Mummies of Urumchi
13: The Testament
14: Nights Below Station Street
15: A Second Chicken Soup for the Woman's Soul (Chicken Soup for the Soul Series)
16: New Vegetarian: Bold and Beautiful Recipes for Every Occasion
17: Under the Black Flag: The Romance and the Reality of Life Among the Pirates
18: Timeline
19: More Cunning Than Man: A Social History of Rats and Man
20: Airframe
21: The Witchfinder (Amos Walker Mystery Series)
22: The Kitchen God's Wife
23: Our Dumb Century: The Onion Presents 100 Years of Headlines from America's Finest News Source
24: Beloved (Plume Contemporary Fiction)
```

### Recommended Books



```
try {
    lazy val address: Array[Byte] = Array(10.toByte, 99.toByte, 0.toByte, 200.toByte)
    val ia = InetAddress.getByAddress(address)
    val socket = new Socket(ia, 1234)
} catch {
    case e: UnknownHostException =>
        System.out.println("WARN : Your hostname, " + e.getMessage() + " is not reachable")
        e.printStackTrace()
}

@Stage 164:> [Stage 164:=====>]
[Stage 165:> [Stage 165:=====>]
The best model improves the baseline by 21.9%.
Books that are recommended for you:
1: Flood - Mississippi 1927
2: The Middle Stories
3: Jane Doe
4: Wild Animals
5: Where You'll Find Me: And Other Stories
6: Tell Me This Isn't Happening
7: Goodbye to the Buttermilk Sky
8: Hitler's Secret Bankers: The Myth of Swiss Neutrality During the Holocaust
9: What If?: The World's Foremost Military Historians Imagine What Might Have Been
10: FEADING GUILTY
11: Clara Callan
12: The Mummies of Urumchi
13: The Testament
14: Nights Below Station Street
15: A Second Chicken Soup for the Woman's Soul (Chicken Soup for the Soul Series)
16: New Vegetarian: Bold and Beautiful Recipes for Every Occasion
17: Under the Black Flag: The Romance and the Reality of Life Among the Pirates
18: Timeline
19: More Cunning Than Man: A Social History of Rats and Man
20: Airframe
21: The Witchfinder (Amos Walker Mystery Series)
22: The Kitchen God's Wife
23: Our Dumb Century: The Onion Presents 100 Years of Headlines from America's Finest News Source
24: Beloved (Plume Contemporary Fiction)

Process finished with exit code 0
```

We had also sent the recommended books to the smart phone as a notification:



## Furniture Malls Recommendation System

```
16/03/10 07:16:28 INFO FileInputFormat: Total input paths to process : 1
Got 1000209 ratings from 6040 users on 3706 FurnitureMalls.
Training: 602252, validation: 198919, test: 199049
[Stage 25:>          (0 + 4) / 4]16/03/10 07:16:25 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeDenseMatrix64F
16/03/10 07:16:28 INFO JniLoader: successfully loaded C:\Users\DEEPU\AppData\Local\Temp\jniloader1966739238328846810\jniLibs\NativeRef\win-x86_64.dll
[Stage 27:>          (0 + 4) / 4]16/03/10 07:16:33 WARN LAPACK: Failed to load implementation from: com.github.fommil.netlib.NativeDenseVector64F
16/03/10 07:16:33 INFO JniLoader: already loaded netlib-native_ref-win-x86_64.dll
RMSE (validation) = 0.8815801121709103 for the model trained with rank = 8, lambda = 0.1, and numIter = 10.
RMSE (validation) = 0.8726203182715503 for the model trained with rank = 8, lambda = 0.1, and numIter = 20.
RMSE (validation) = 3.7558695311242833 for the model trained with rank = 8, lambda = 10.0, and numIter = 10.
[Stage 950:>          (0 + 4) / 4]RMSE (validation) = 3.7558695311242833 for the model trained with rank = 8, lambda = 10.0, and numIter = 20.
[Stage 1125:=====          (3 + 1) / 4]RMSE (validation) = 0.8772284010651425 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
RMSE (validation) = 0.8710227453579589 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
[Stage 1595:>          (0 + 4) / 4]RMSE (validation) = 3.7558695311242833 for the model trained with rank = 12, lambda = 10.0, and numIter = 20.
RMSE (validation) = 3.7558695311242833 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 = 20, and its RMSE on the test set is 0.8690494992690084.
The best model improves the baseline by 21.95%.
```

This Screenshot shows the Furniture Malls recommended to you:

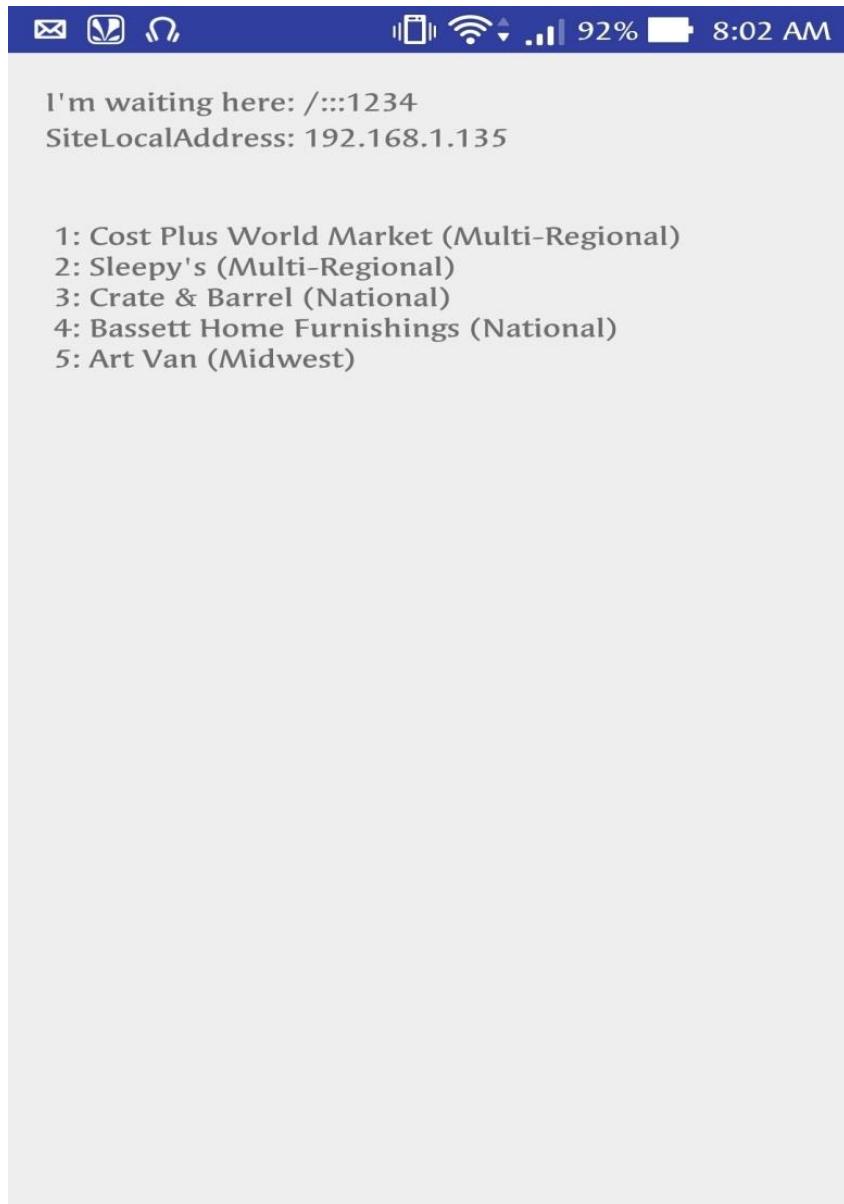
The screenshot shows the IntelliJ IDEA 15.0.1 interface with the following details:

- Project Structure:** Shows the project structure with files like `SparkCollaborativeFiltering.scala`, `SparkMovieRecommendation.scala`, and `FurnitureShopRating.txt`.
- Run Output:** Displays the Scala code and its execution output. The output includes:

  - INFO messages about deprecated mapred.\* methods.
  - INFO message about the number of ratings: "Got 1000209 ratings from 6040 users on 3706 FurnitureMalls."
  - Training parameters: "Training: 602252, validation: 198919, test: 199049".
  - Model training progress: Stages 25, 27, 950, 1125, and 1595.
  - Final RMSE values: 0.8690494992690084 (test set).
  - A list of recommended furniture malls: Cost Plus World Market (Multi-Regional), Sleepy's (Multi-Regional), Crate & Barrel (National), Bassett Home Furnishings (National), Art Van (Midwest).
  - INFO messages at the end: RemoteActorRefProvider\$RemotingTerminator shutting down, remote daemon shutdown, and remoting shut down.
  - Process finished with exit code 0.

- Terminal:** Shows the command "Compilation completed successfully in 31s 964ms (7 minutes ago)".
- Status Bar:** Shows "43:1 CRLF: UTF-8:" and other standard status bar information.

The notification has been sent to the android mobile which shows the recommended furniture list.

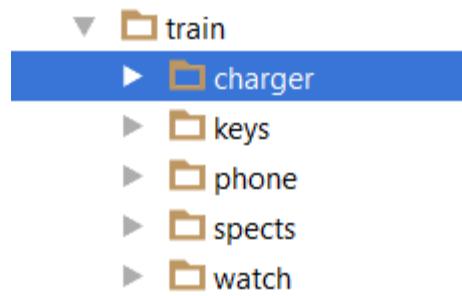


## Image Classification

The Training dataset is full with images of objects the user want to identify which were got misplaced. And the test dataset is the streaming images which are captured by the robot while trying to identify the misplaced object. So when the robot finds the image, it classifies the image and notifies about the object to the users device.

Screenshots of the image classification:

Train data:



Identifying Key descriptors from Training dataset

```
// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTress = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100
```

Key Descriptors 1556 x 128  
Key Descriptors 1334 x 128  
Key Descriptors 1000 x 128  
Key Descriptors 692 x 128  
Key Descriptors 515 x 128  
Key Descriptors 781 x 128  
Key Descriptors 20587 x 128  
Key Descriptors 24357 x 128  
Key Descriptors 2381 x 128  
Key Descriptors 24357  
Key Descriptors 17111 x 128  
Key Descriptors 27961 x 128  
Key Descriptors 3518 x 128  
Key Descriptors 27961

## Cluster formation:

The screenshot shows the IntelliJ IDEA interface with the 'Image\_Classification' project open. The 'IPApp.scala' file is the active editor, containing Scala code for a machine learning model. The 'Run' tool window displays the application's log output, which includes various INFO and WARN messages related to the file input format and task context. The status bar at the bottom right indicates the date as 4/6/2016.

```
// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100
```

Logs from IPApp:

```
16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 64,986B for [point, values, list, element] DOUBLE: 12,800 values, 102,912B raw, 64,939B comp, 1 pages, encodings: [RLE_PLAIN]
16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 64B for [point, indices, list, element] INT32: 100 values, 14B raw, 30B comp, 1 pages, encodings: [RLE_PLAIN]
16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 64,817B for [point, values, list, element] DOUBLE: 12,800 values, 102,912B raw, 64,779B comp, 1 pages, encodings: [RLE_PLAIN]
16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 66,094B for [point, values, list, element] DOUBLE: 12,800 values, 102,912B raw, 66,047B comp, 1 pages, encodings: [RLE_PLAIN]
16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 62B for [point, type] INT32: 100 values, 10B raw, 28B comp, 1 pages, encodings: [BIT_PACKED, PLAIN_DICTIONARY, RLE], d=1
16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 50B for [point, type] INT32: 100 values, 7B raw, 27B comp, 1 pages, encodings: [BIT_PACKED, RLE, PLAIN]
16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 53B for [point, indices, list, element] INT32: 100 values, 14B raw, 30B comp, 1 pages, encodings: [RLE, PLAIN]
16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 65,899B for [point, values, list, element] DOUBLE: 12,800 values, 102,912B raw, 65,852B comp, 1 pages, encodings: [RLE]
16/04/06 17:57:26 INFO FileOutputCommitter: Saved output of task 'attempt_201604061757_0058_m_000002_0' to file:/C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial/parquet/part-r-00002
16/04/06 17:57:26 INFO FileOutputCommitter: Saved output of task 'attempt_201604061757_0058_m_000000_0' to file:/C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial/parquet/part-r-00000
16/04/06 17:57:26 INFO FileOutputCommitter: Saved output of task 'attempt_201604061757_0058_m_000001_0' to file:/C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial/parquet/part-r-00001
16/04/06 17:57:26 INFO FileOutputCommitter: Saved output of task 'attempt_201604061757_0058_m_000003_0' to file:/C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial/parquet/part-r-00003
16/04/06 17:57:26 INFO ParquetFileReader: Initiating action with parallelism: 5
Save Clusters to data3/model/clusters
16/04/06 17:57:26 INFO FileInputFormat: Total input paths to process : 1
16/04/06 17:57:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:57:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:57:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:57:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:57:27 INFO deprecation: mapred.min.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize
16/04/06 17:57:27 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskK
16/04/06 17:57:27 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskK
16/04/06 17:57:27 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskK
16/04/06 17:57:27 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskK
16/04/06 17:57:28 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.
16/04/06 17:57:28 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.
16/04/06 17:57:28 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.
16/04/06 17:57:28 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.
```

Histogram generation based on size specified:

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** Shows a tree view of the project structure under 'Image\_Classification'.
- Code Editor:** Displays the file 'IPApp.scala' containing Scala code for a machine learning application.
- Run Output:** Shows the terminal output of the application's execution, displaying multiple histograms being generated.
- Status Bar:** At the bottom, it says "Compilation completed successfully in 7s 29ms (20 minutes ago)".
- Bottom Icons:** Standard Windows-style icons for file operations like Find, Run, Terminal, and TODO.

## Model has been created based on the Random Forest Classifier Algorithm

The screenshot shows the IntelliJ IDEA interface. The top navigation bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help. The title bar indicates the project is 'Image\_Classification' and the current file is 'IPApp.scala'. A message in the top right says 'Platform and Plugin Updates' and 'IntelliJ IDEA is ready to update'. The left sidebar shows the project structure with 'Project' selected, displaying 'data3' and its subfolders: 'model', 'test', 'keys', 'phone', 'specs', 'watch', and 'train'. The main code editor window contains Scala code for a Random Forest classifier. The bottom run log window shows the command-line output of the application running, including logs from 'ColumnChunkPageWriteStore', 'FileOutputCommitter', and 'ParquetFileReader'. The bottom status bar shows the date and time as '4/6/2016 6:09 PM'.

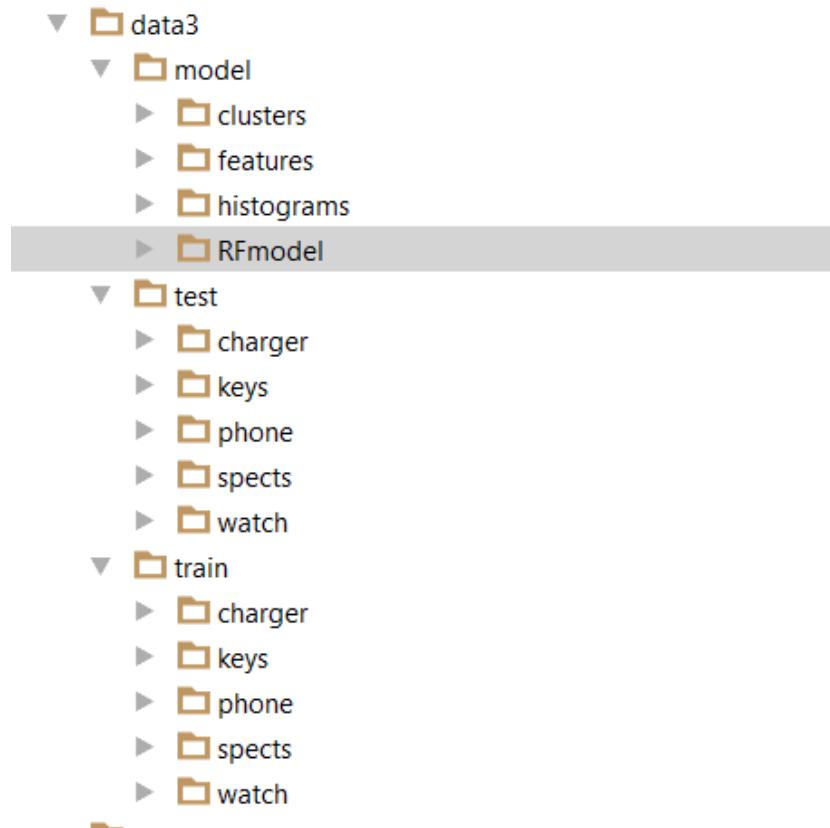
```

// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100

val numTrees = 4 to(5, 1)
val strategies = List("all", "sqrt", "log2", "onethird")
val maxDepths = 3 to(6, 1)
val impurities = List("gini", "entropy")

var bestModel: Option[RandomForestModel] = None
var bestErr = 1.0
val bestParams = new mutable.HashMap[Any, Any]()
var bestNumTrees = 0
var bestNumTreesSubSet = ""

```



## Confusion Matrix:

IntelliJ IDEA 15.0.3

```
// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100
```

Run IPApp

```
1.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0
0.0 0.0 1.0 0.0
0.0 0.0 0.0 1.0
1.0
numTrees 5 featureSubsetStrategy onethird impurity entropy maxDepth 5
16/04/06 17:59:04 WARN DecisionTreeMetadata: DecisionTree reducing maxBins from 100 to 15 (= number of training instances)
Test Error = 0.0
=====
Confusion matrix =====
1.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0
0.0 0.0 1.0 0.0
0.0 0.0 0.0 1.0
1.0
numTrees 5 featureSubsetStrategy onethird impurity entropy maxDepth 6
16/04/06 17:59:05 WARN DecisionTreeMetadata: DecisionTree reducing maxBins from 100 to 15 (= number of training instances)
Test Error = 0.25
=====
Confusion matrix =====
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
0.75
Best Err 0.0
Best params (featureSubsetStrategy,all) (numTrees,4) (maxDepth,3) (impurity,gini)
16/04/06 17:59:05 WARN DecisionTreeMetadata: DecisionTree reducing maxBins from 100 to 15 (= number of training instances)
16/04/06 17:59:06 INFO FileOutputCommitter: Saved output of task 'attempt_201604061759_1280_m_000000_4915' to file:/C:/Users/npdar/Desktop/Acads/BigDataAnalytics/Assignment/Tut...
16/04/06 17:59:07 INFO CodecConfig: Compression: GZIP
16/04/06 17:59:07 INFO ParquetOutputFormat: Parquet block size to 134217728
16/04/06 17:59:07 INFO ParquetOutputFormat: Parquet block size to 134217728
```

Ask me anything

Compilation completed successfully in 7s 29ms (19 minutes ago)

Event Log: 7754 LF: UTF-8 6:10 PM 4/6/2016

IntelliJ IDEA 15.0.3

```
(0..0.4)
(3.0..3)
(3.0..3)
(2.0..3)
(2.0..2)
(3.0..2)
(0..0.2)
(3.0..2)
(2.0..2)
(3.0..2)
(0..0.2)
(2.0..1)
(1.0..1)
(0..0.1)
(3.0..1)
(0..0.1)
(1..0.1)
(1..0.1)
(1..0.0)
(3..0.0)
(0..0.0)
(0..0.0)
(0..0.0)
(1..0.0)
(2..0.0)
0.3333333333333333
=====
Confusion matrix =====
3.0 2.0 1.0 1.0 0.0
2.0 3.0 1.0 1.0 0.0
2.0 0.0 2.0 3.0 0.0
0.0 0.0 1.0 2.0 0.0
4.0 1.0 1.0 0.0 0.0
0.3333333333333333
16/04/06 18:00:51 INFO RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
16/04/06 18:00:51 INFO RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.
```

Process finished with exit code 0

Ask me anything

Compilation completed successfully in 7s 29ms (21 minutes ago)

Event Log: 25771 LF: UTF-8 6:12 PM 4/6/2016

## Predicted Results Snapshots from the testing dataset

Screenshot of IntelliJ IDEA 15.0.3 showing the code editor and run log for the `Image_Classification` project.

**Code Editor:**

```
// testImageClassification(sc)

val testImages = sc.wholeTextFiles(s"${IPSettings.INPUT_DIR}/*/*.jpg")
val testImagesArray = testImages.collect()
var predictionLabels = List[String]()
testImagesArray.foreach(f => {
    val splitStr = f._1.split("/")
    val predictedClass: Double = classifyImage(sc, splitStr(1))
    val segments = f._1.split("/")
    val cat = segments(segments.length - 2)
    val GivenClass = IMAGE_CATEGORIES.indexOf(cat)
    println(s"Predicting test image : " + cat + " as " + IMAGE_CATEGORIES(predictedClass.toInt))
    predictionLabels = predictionLabels + ":" + GivenClass :: predictionLabels
})
```

**Run Log:**

```
16/04/06 17:46:37 INFO CodecPool: Got brand-new decompressor [.gz]
16/04/06 17:46:37 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 128
16/04/06 17:46:37 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 59 records.
16/04/06 17:46:37 INFO InternalParquetRecordReader: at row 0. reading next block
16/04/06 17:46:37 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 59
16/04/06 17:46:37 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 138 records.
16/04/06 17:46:37 INFO InternalParquetRecordReader: at row 0. reading next block
16/04/06 17:46:37 INFO InternalParquetRecordReader: Got brand-new decompressor [.gz]
16/04/06 17:46:37 INFO InternalParquetRecordReader: block read in memory in 2 ms. row count = 138
Predicting test image : charger as charger
file:///C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial 9/CS5542 - Tutorial 9 Code/Image_Classification/data3/test/charger/charger4.jpg
16/04/06 17:46:37 INFO FileInputFormat: Total input paths to process : 1
16/04/06 17:46:37 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:46:37 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:46:37 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:46:37 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:46:38 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskAttempt
16/04/06 17:46:38 INFO ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskAttempt
16/04/06 17:46:38 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.
```

**System Tray:**

- Ask me anything
- Event Log: 2661 LF: UTF-8
- 547 PM 4/6/2016

Screenshot of IntelliJ IDEA 15.0.3 showing the code editor and run log for the `Image_Classification` project.

**Code Editor:**

```
// Empty categoricalFeatureInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeatureInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100

val numOFTrees = 4 to(5, 1)
val strategies = List("all", "sqrt", "log2", "onethird")
val maxDepths = 3 to(6, 1)
val impurities = List("gini", "entropy")

var bestModel: Option[RandomForestModel] = None
var bestErr = 1.0
var bestParams = new mutable.HashMap[Any, Any]()
var bestNumTrees = 0
var bestFeatureSubSet = ""
```

**Run Log:**

```
16/04/06 18:00:26 INFO InternalParquetRecordReader: at row 0. reading next block
16/04/06 18:00:26 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 7 records.
16/04/06 18:00:26 INFO InternalParquetRecordReader: at row 0. reading next block
16/04/06 18:00:26 INFO CodecPool: Got brand-new decompressor [.gz]
16/04/06 18:00:26 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7
16/04/06 18:00:26 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7
16/04/06 18:00:26 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 7 records.
16/04/06 18:00:26 INFO InternalParquetRecordReader: at row 0. reading next block
16/04/06 18:00:26 INFO CodecPool: Got brand-new decompressor [.gz]
16/04/06 18:00:26 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7
Predicting test image : spects as spects
16/04/06 18:00:27 INFO FileInputFormat: Total input paths to process : 1
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
```

**System Tray:**

- Ask me anything
- Event Log: 25468 LF: UTF-8
- 605 PM 4/6/2016

Image\_Classification - [C:\Users\npdar\Desktop\Acads\BigDataAnalytics\Assignment\Assignment\Tutorial 9\CS5542 - Tutorial 9 Code]\Image\_Classification] - [image\_classification] - ..\src\main\scala\IPApp.scala - IntelliJ IDEA 15.0.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Image\_Classification > data3 > model >

Project Packages Project Files Problems

IPApp.scala IPSettings.scala build.sbt

```
// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100

val numOFTrees = 4 to(5, 1)
val strategies = List("all", "sqrt", "log2", "onethird")
val maxDepths = 3 to(6, 1)
val impurities = List("gini", "entropy")

var bestModel: Option[RandomForestModel] = None
var bestErr = 1.0
val bestParams = new mutable.HashMap[Any, Any]()
var bestnumTrees = 0
var bestFeatureSubSet = ""
```

Run IPApp

```
16/04/06 18:00:18 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 7 records.
16/04/06 18:00:18 INFO InternalParquetRecordReader: at row 0. reading next block
16/04/06 18:00:18 INFO InternalParquetRecordReader: block read in memory in 0 ms. row count = 7
16/04/06 18:00:18 INFO CodecPool: Got brand-new decompressor [.gz]
16/04/06 18:00:18 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7
Predicting test image : phone as phone
16/04/06 18:00:18 INFO FileInputFormat: Total input paths to process : 1
16/04/06 18:00:18 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:18 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:18 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:18 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:18 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA
16/04/06 18:00:18 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA
16/04/06 18:00:18 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.
16/04/06 18:00:18 INFO InternalParquetRecordReader: at row 0. reading next block
16/04/06 18:00:18 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA
16/04/06 18:00:18 INFO CodecPool: Got brand-new decompressor [.gz]
```

Find Run Terminal TODO Event Log

Compilation completed successfully in 7s 29ms (16 minutes ago)

2153:72 LF+ UTF-8 6:05 PM 4/6/2016

Image\_Classification - [C:\Users\npdar\Desktop\Acads\BigDataAnalytics\Assignment\Assignment\Tutorial 9\CS5542 - Tutorial 9 Code]\Image\_Classification] - [image\_classification] - ..\src\main\scala\IPApp.scala - IntelliJ IDEA 15.0.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Image\_Classification > data3 > model >

Project Packages Project Files Problems

IPApp.scala IPSettings.scala build.sbt

```
// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100

val numOFTrees = 4 to(5, 1)
val strategies = List("all", "sqrt", "log2", "onethird")
val maxDepths = 3 to(6, 1)
val impurities = List("gini", "entropy")

var bestModel: Option[RandomForestModel] = None
var bestErr = 1.0
val bestParams = new mutable.HashMap[Any, Any]()
var bestnumTrees = 0
var bestFeatureSubSet = ""
```

Run IPApp

```
16/04/06 17:59:53 INFO CodecPool: Got brand-new decompressor [.gz]
16/04/06 17:59:53 INFO InternalParquetRecordReader: block read in memory in 0 ms. row count = 7
16/04/06 17:59:53 INFO CodecPool: Got brand-new decompressor [.gz]
16/04/06 17:59:53 INFO InternalParquetRecordReader: block read in memory in 18 ms. row count = 7
16/04/06 17:59:53 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA
16/04/06 17:59:53 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 7 records.
16/04/06 17:59:53 INFO InternalParquetRecordReader: at row 0. reading next block
16/04/06 17:59:53 INFO CodecPool: Got brand-new decompressor [.gz]
16/04/06 17:59:53 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7
Predicting test image : keys as keys
16/04/06 17:59:53 INFO FileInputFormat: Total input paths to process : 1
16/04/06 17:59:53 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:59:53 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:59:53 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:59:53 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:59:53 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:59:53 INFO ParquetRecordReader: Initiating counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA
16/04/06 17:59:53 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA
16/04/06 17:59:53 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA
```

Find Run Terminal TODO Event Log

Compilation completed successfully in 7s 29ms (17 minutes ago)

1765:55 LF+ UTF-8 6:07 PM 4/6/2016



47% 9:39 PM

Hello World!

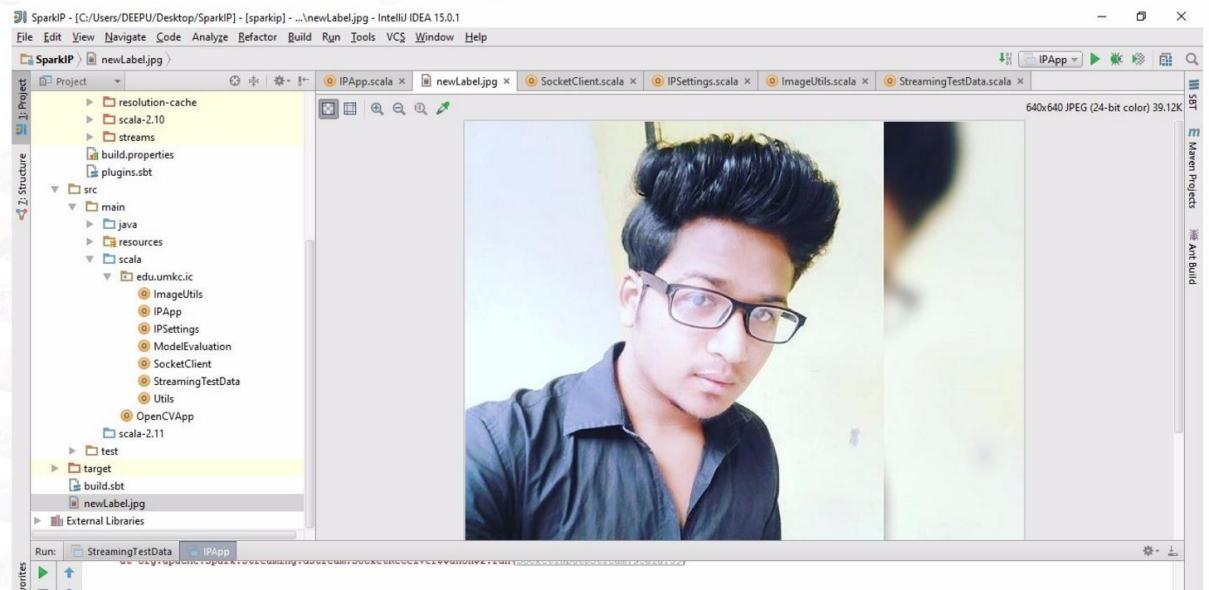
**NOTIFY WEARABLE**

I'm waiting here: 1234

SiteLocalAddress: 192.168.0.21

#1 from /192.168.0.23:53290

Predicted test image:spects/nreplayed: Hello from  
Android, you are #1



#### **4.4 Deployment:**

#### **Git Hub Link:**

<https://github.com/npdarsini/Assist-Robot>

### **5. Project Management:**

#### **5.1 Implementation status report:**

70% implementation has been implemented. This phase involves the development of Image classification report. Basically the system will classify the object based on the provided image and will notify the user about the object. The team members has an equal contribution towards the development and it took around 4 complete days to give an outlook for this phase. First we had tried developing a system with the static training and testing data sets and later we had made the testing part to be the streaming data which comes from the android device. Upon classifying the image the notification will be sent to the user device. Assuming the user will be carrying the smart android device every single time, we thought that this could be as an innovative thought as the user don't need to spend a lot of time for finding the lost objects.

Classification based on the Streaming Data #19

npdarsini opened this issue 33 minutes ago · 0 comments

npdarsini commented 33 minutes ago · No description provided

npdarsini added enhancement, help wanted labels 32 minutes ago

Write Preview AA- B i Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard

Styling with Markdown is supported

Close issue Comment

Pipeline In Progress

Labels enhancement Help wanted

Milestone No milestone

Estimate 13

Assignee No one—assign yourself

Epic Not inside an Epic

Notifications Unsubscribe You're receiving notifications because you authored the thread.

1 metric issue

Show all downloads...

FinalReport.docx Train.zip model.zip

Ask me anything

9:06 PM 4/6/2016

Notification to Android Device - Recognized Object #18

DheerajaUmkc opened this issue a day ago · 0 comments

DheerajaUmkc commented a day ago · To check if our project becomes better using Audio mining techniques  
Not utilized

DheerajaUmkc added the enhancement label a day ago

DheerajaUmkc added this to the Project - Increment 3 milestone a day ago

DheerajaUmkc set the estimate to 5 a day ago

DheerajaUmkc closed this 7 hours ago

npdarsini changed the title from To utilize audio mining to Notification to Android Device - Recognized Object 35 minutes ago

Write Preview AA- B i Leave a comment

Labels enhancement

Milestone Project - Increment 3

Estimate 5

Assignee No one—assign yourself

Epic Not inside an Epic

Notifications Unsubscribe You're receiving notifications because you're subscribed to this repository.

1 participant

Show all downloads...

FinalReport.docx Train.zip model.zip

Greetings!

9:06 PM 4/6/2016

[Training data sets - Creation with different objects #17](https://github.com/npdarsini/Assist-Robot/issues/17)

**Closed** DheerajaUmkc opened this issue a day ago · 0 comments

DheerajaUmkc commented a day ago  
Making use of text mining techniques

DheerajaUmkc added the enhancement label a day ago

DheerajaUmkc added this to the Project - Increment 3 milestone a day ago

DheerajaUmkc set the estimate to 5 a day ago

DheerajaUmkc assigned Deepu123start and unassigned Deepu123start a day ago

DheerajaUmkc closed this a day ago

DheerajaUmkc reopened this a day ago

DheerajaUmkc closed this a day ago

npdarsini changed the title from Usage of Text mining to Training data sets - Creation with different objects 37 minutes ago

**Labels**  
enhancement

**Milestone**  
Project - Increment 3

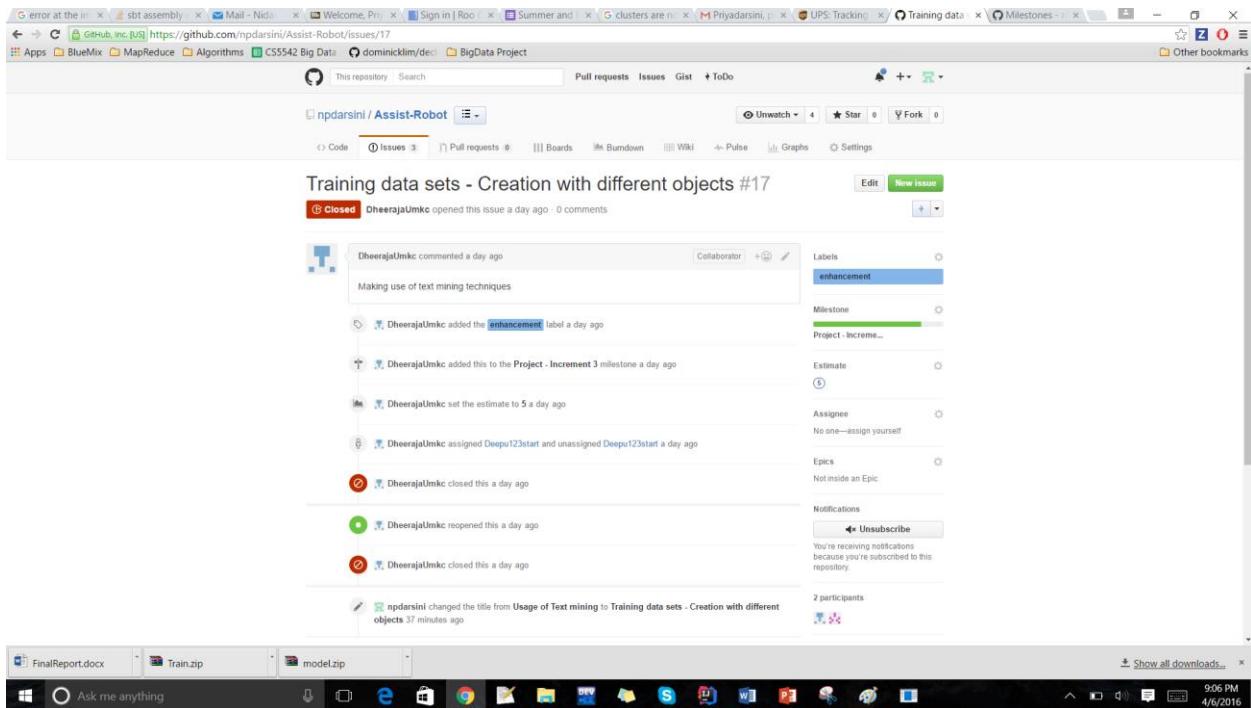
**Estimate**  
(5)

**Assignee**  
No one—assign yourself

**Epic**  
Not inside an Epic

**Notifications**  
Unsubscribe  
You're receiving notifications because you're subscribed to this repository.

2 participants



[Exploring different Classification algorithms - Random Forest, Decision Tree #16](https://github.com/npdarsini/Assist-Robot/issues/16)

**Closed** DheerajaUmkc opened this issue 7 days ago · 0 comments

DheerajaUmkc commented 7 days ago  
Trying to make use of classification algorithms and figure out the best that suits our application

Deepu123start was assigned by DheerajaUmkc 7 days ago

DheerajaUmkc added the enhancement label 7 days ago

DheerajaUmkc added this to the Project - Increment 3 milestone 7 days ago

DheerajaUmkc set the estimate to 8 7 days ago

DheerajaUmkc changed the estimate from 8 to 13 7 days ago

DheerajaUmkc closed this a day ago

npdarsini changed the title from Exploring different supervised learning algorithms to Exploring different Classification algorithms - Random Forest, Decision Tree 38 minutes ago

**Labels**  
enhancement

**Milestone**  
Project - Increment 3

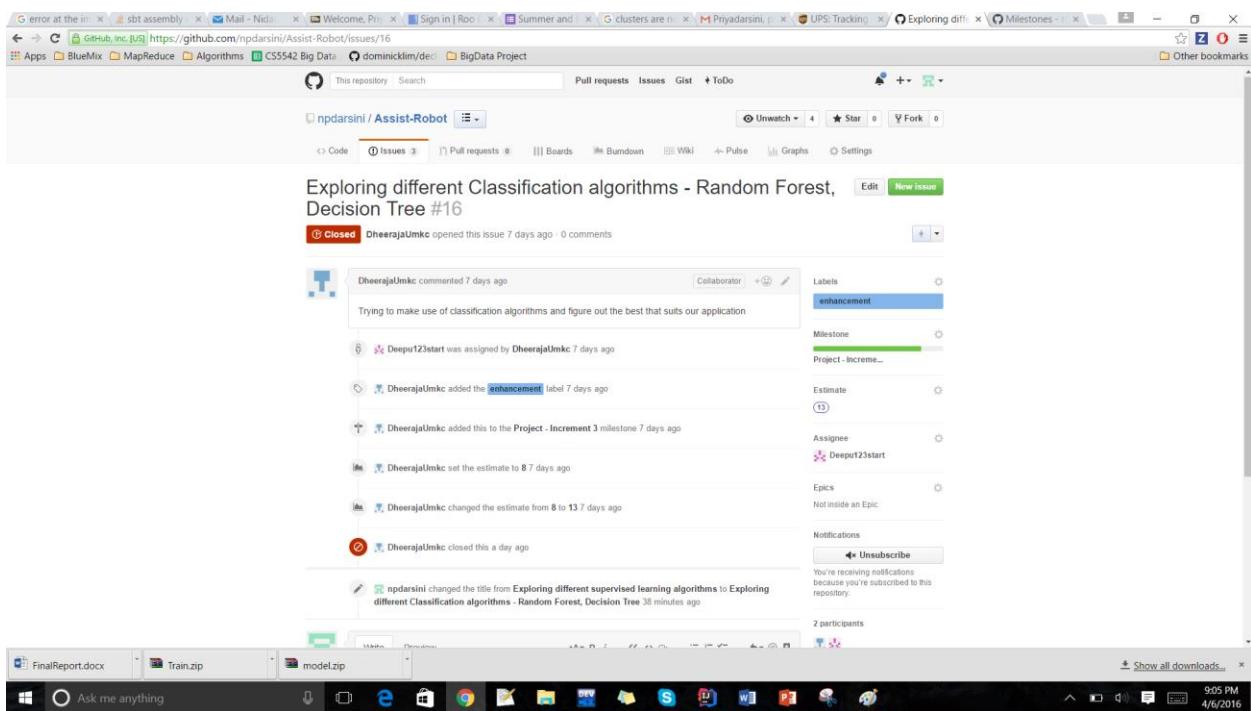
**Estimate**  
(13)

**Assignee**  
Deepu123start

**Epic**  
Not inside an Epic

**Notifications**  
Unsubscribe  
You're receiving notifications because you're subscribed to this repository.

2 participants



GitHub Issues - npdarsini / Assist-Robot

**Image classification #15**

**Closed** DheerajaUmkc opened this issue 7 days ago · 0 comments

DheerajaUmkc commented 7 days ago  
Image classification plays a vital role in our project

DheerajaUmkc self-assigned this 7 days ago

DheerajaUmkc added the **enhancement** label 7 days ago

DheerajaUmkc added this to the **Project - Increment 3** milestone 7 days ago

DheerajaUmkc set the estimate to 13 7 days ago

DheerajaUmkc changed the pipeline from **In Progress** to **Done** a day ago

DheerajaUmkc closed this a day ago

**Labels**: enhancement

**Milestone**: Project - Increment 3

**Estimate**: 13

**Assignee**: DheerajaUmkc

**Epic**: Not inside an Epic

**Notifications**: Unsubscribe

1 participant

Leave a comment

FinalReport.docx Train.zip model.zip

Ask me anything

9:05 PM 4/6/2016

GitHub Issues - npdarsini / Assist-Robot

**Creating a perfect training Dataset and collecting Test data #11**

**Closed** DheerajaUmkc opened this issue 4 hours ago · 0 comments

DheerajaUmkc commented 4 hours ago  
Test data helps in finding the results and to understand the need for enhancement of the training data

DheerajaUmkc added the **enhancement** label 4 hours ago

DheerajaUmkc added this to the **Project - Increment 2** milestone 4 hours ago

DheerajaUmkc set the estimate to 3 4 hours ago

DheerajaUmkc changed the pipeline from **In Progress** to **Done** 3 hours ago

DheerajaUmkc closed this 2 hours ago

**Labels**: enhancement

**Milestone**: Project - Increment 2

**Estimate**: 3

**Assignee**: No one—assign yourself

**Notifications**: Unsubscribe

1 participant

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

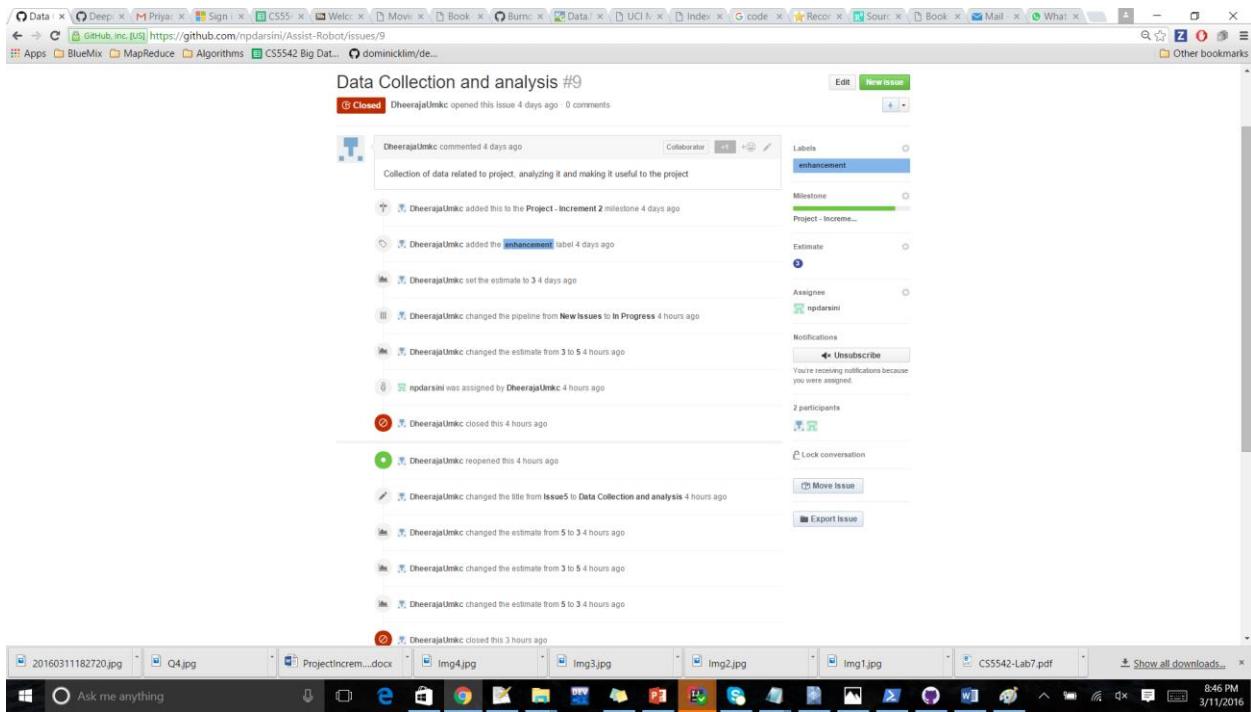
Comment Reopen issue Export issue

20160311182720.jpg Q4.jpg ProjectIncrem...docx Img4.jpg Img3.jpg Img2.jpg Img1.jpg CS5542-Lab7.pdf

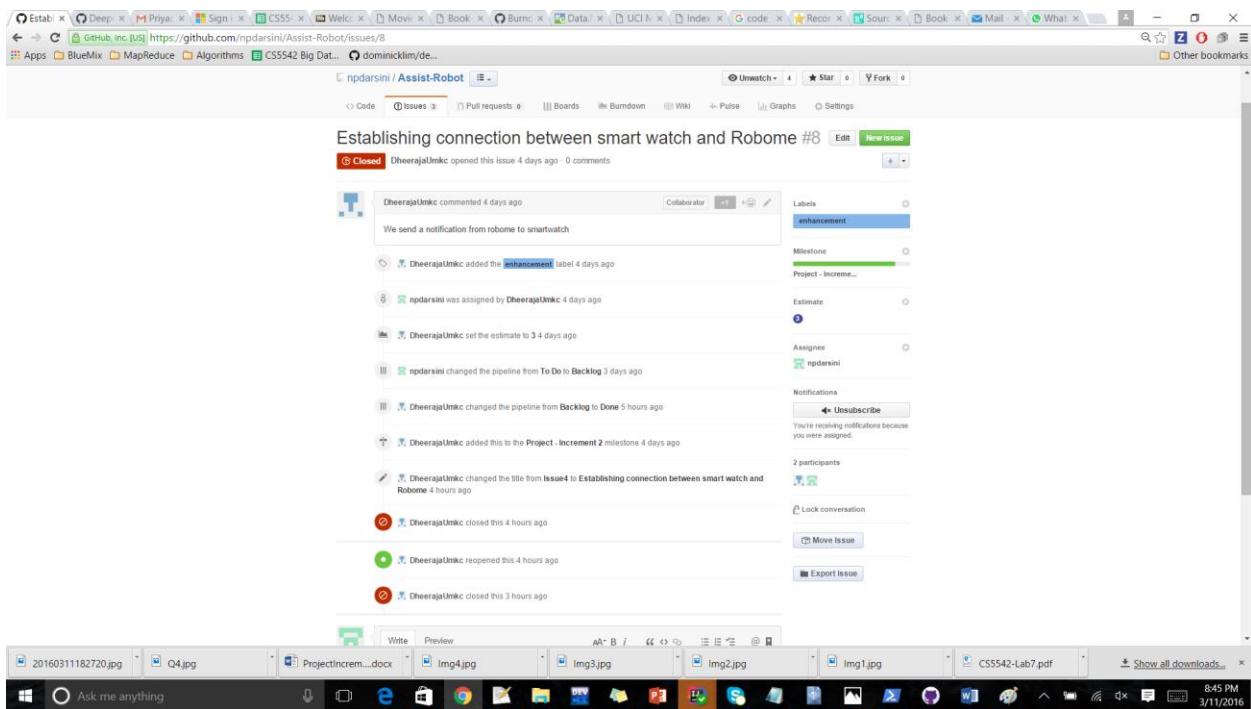
Ask me anything

8:47 PM 3/11/2016

Firefox browser window showing a GitHub issue titled "Data Collection and analysis #9". The issue is closed by DheerajUmkc. The description states: "Collection of data related to project, analyzing it and making it useful to the project". Labels: enhancement. Milestone: Project - Increment 2. Estimate: 3 hours. Assignee: npdarsini. Notifications: You're receiving notifications because you were assigned. Participants: 2. Conversation history shows multiple comments from DheerajUmkc and npdarsini, including changes in title, estimate, and pipeline status.



Firefox browser window showing a GitHub issue titled "Establishing connection between smart watch and Robome #8". The issue is closed by DheerajUmkc. The description states: "We send a notification from robome to smartwatch". Labels: enhancement. Milestone: Project - Increment 2. Estimate: 3 hours. Assignee: npdarsini. Notifications: You're receiving notifications because you were assigned. Participants: 2. The conversation history shows multiple comments from DheerajUmkc and npdarsini, including changes in title, estimate, and pipeline status.



Installation of Spark #7

**Closed** Dheeraj@Umkc opened this issue 4 days ago · 0 comments

Dheeraj@Umkc commented 4 days ago Since usage of spark makes the project more flexible because of availability of RDD's

Dheeraj@Umkc added the enhancement label 4 days ago

dunil210 was assigned by Dheeraj@Umkc 4 days ago

Dheeraj@Umkc added this to the Project - increment 2 milestone 4 days ago

Dheeraj@Umkc set the estimate to 3-4 days ago

npdarsini changed the pipeline from To Do to Backlog 3 days ago

Dheeraj@Umkc changed the title from Issue3 to Installation of Spark 4 hours ago

dunil210 was unassigned by Dheeraj@Umkc 4 hours ago

Dheeraj@Umkc changed the pipeline from Backlog to Done 4 hours ago

Deepur123start was assigned by Dheeraj@Umkc 4 hours ago

Dheeraj@Umkc closed this 4 hours ago

Dheeraj@Umkc reopened this 4 hours ago

Dheeraj@Umkc closed this 4 hours ago

The screenshot shows a GitHub issue page for a project titled 'Installation of Spark'. The issue is labeled as 'Closed' and was opened by 'Dheeraj@Umkc' 4 days ago. The description of the issue is: 'Since usage of spark makes the project more flexible because of availability of RDD's'. The issue has been updated multiple times with comments from 'Dheeraj@Umkc' and 'npdarsini'. Labels include 'enhancement'. Milestones and estimates are also visible. The GitHub interface includes a sidebar with various project-related links like 'Code', 'Issues', 'Pull requests', 'Boards', 'Burndown', 'Wiki', 'Graphs', and 'Settings'.

Features of RoboMe #6

**Closed** Dheeraj@Umkc opened this issue 4 days ago · 0 comments

Dheeraj@Umkc commented 4 days ago Going through the features of Robome and what all can be derived using its basic features

Dheeraj@Umkc added enhancement, question labels 4 days ago

Deepur123start was assigned by Dheeraj@Umkc 4 days ago

Dheeraj@Umkc added this to the Project - increment 2 milestone 4 days ago

Dheeraj@Umkc set the estimate to 5-4 days ago

npdarsini changed the pipeline from In Progress to To Do 3 days ago

npdarsini changed the pipeline from To Do to In Progress 3 days ago

Dheeraj@Umkc changed the pipeline from In Progress to Done 5 hours ago

Dheeraj@Umkc removed the enhancement label 4 hours ago

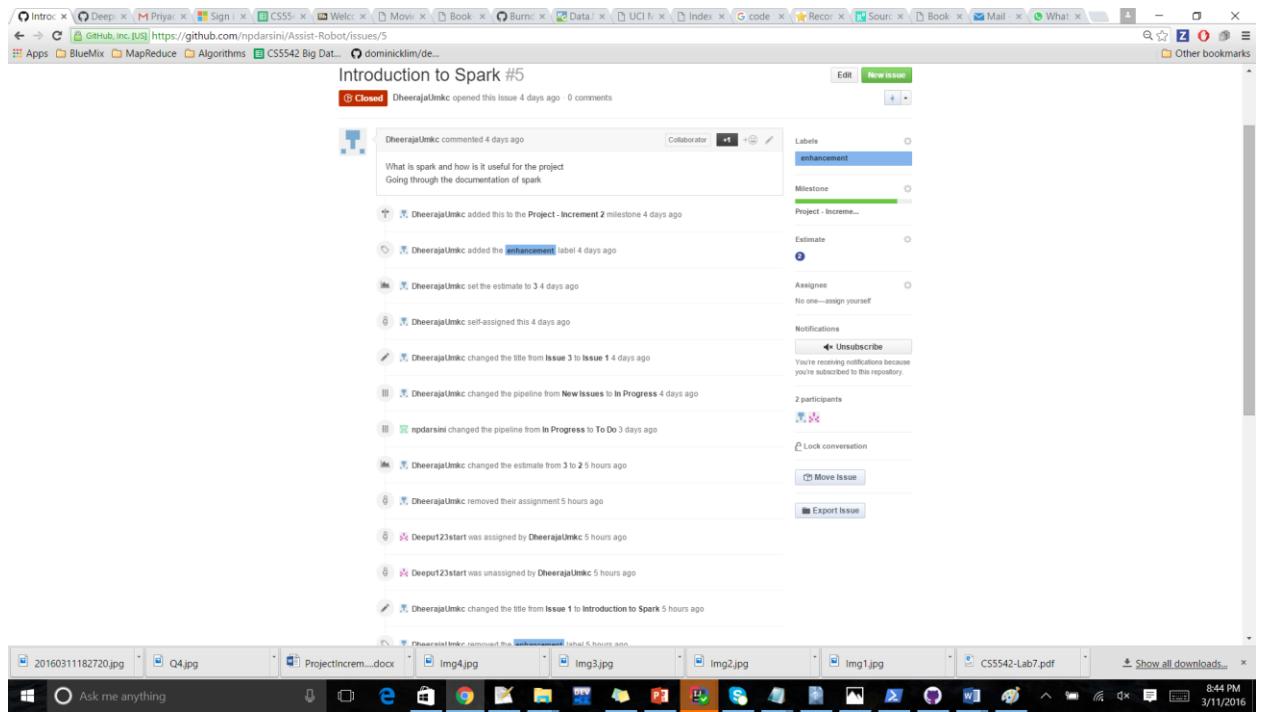
Deepur123start was unassigned by Dheeraj@Umkc 4 hours ago

Dheeraj@Umkc changed the estimate from 5 to 1 hours ago

npdarsini was assigned by Dheeraj@Umkc 4 hours ago

npdarsini was unassigned by Dheeraj@Umkc 4 hours ago

The screenshot shows a GitHub issue page for a project titled 'Features of RoboMe'. The issue is labeled as 'Closed' and was opened by 'Dheeraj@Umkc' 4 days ago. The description of the issue is: 'Going through the features of Robome and what all can be derived using its basic features'. The issue has been updated multiple times with comments from 'Dheeraj@Umkc' and 'npdarsini'. Labels include 'enhancement' and 'question'. Milestones and estimates are also visible. The GitHub interface includes a sidebar with various project-related links like 'Code', 'Issues', 'Pull requests', 'Boards', 'Burndown', 'Wiki', 'Graphs', and 'Settings'.



## Bibliography:

Lab Tutorials and the material provided by Dr. Lee.

# Assist Robot

## Project Report

### Project Team – 8

#### Team Members

Priyadarsini Nidadavolu(17)

Deepthi Priyadarshini Penmetsa(22)

Dheeraja Vallabhaneni(28)

Tej Kumar Yentrabragada(33)

## Table of Contents

Section	Description
1	Introduction
2	Project Goal and Objectives
3	Tools and Frameworks
4	Project Plan
	4.1 Schedule
	4.2 Project Timelines
	4.2.1 Team Members
	4.2.2 Tasks and Responsibilities
	4.3 Burndown Chart
5	Increment Report
	5.1 Increment based on Phases
	5.1.1 Phase 1 - Existing API
	5.1.2 Phase 2 - Recommendation System
	5.1.3 Phase 3 – Image Classification System
	5.2 Design of Features
	5.3 Implementation
	5.4 Deployment
6	Project Management
	6.1 Implementation Status Reports
7	Future Scope
8	Bibliography

## **1. Introduction:**

The main goal of the project is to help people in finding their misplaced objects. Basically humans have a tendency to forget their belongings somewhere in their house and search for it for hours together. For example, if I have an important business meeting to attend, but I don't remember where I placed my car keys, then I might miss my meeting or can be late to the schedule. So to prevail in these circumstances here comes our Friendly Robot- My Friend which could assist me in keeping track of my personal things. So what this robot will do is that it will have entire map (laser scan) of the building and objects in the building in its memory. So we will feed the robot with the objects that are highly important to us, like car keys, some files, phone and laptop. This robot will keep its eye on these objects and notify their location to its master upon request. Additionally our Robot who will be an eFriend who will help us to choose the furniture to our home. Also our robot will suggest us the top rated books.

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

The primary goals of our project is described below:

- To implement a module to create an android application where the user is provided with the login page. The user will be provided with the features of registering the new account or to login with an existing account.
- To implement an android module where user identification is done using facial reorganization.
- To implement an android module where there must be a camera facility. The objects image must be captured and should save the image to the SD card.
- To implement a module which could implement client server module where the client (android device) will be sending the streaming images to server (Spark).
- To implement an image classification module using machine learning techniques in which the image that comes from the android device should be classified.
- To implement a recommendation module where the location of the classified image should be recommended.
- To implement a module where the notifications about the classification and recommendation will be sent to an android device.

- To build a recommendation system which will suggest us about the latest furniture details, their quality and from which brand/ shop we could purchase from. This feature enables us to decorate our houses with rich interior designing.

### **3. Tools and Frameworks**

#### **Android Studio:**

An official Integrated Developer Environment which we used for Android application development. It is based on IntelliJ IDEA with powerful code editor and developer tools embedding flexible ‘Gradle’ based build system. It also support drag and drop editing which helped us in developing rich User Interface. It also had built in feature like emulator where we can test our developed applications without being deployed on actual device.

#### **IntelliJ IDEA:**

It's a Java Integrated Development Environment which helps to develop software applications with the support of various languages like Scala, SQL, Java which we had used for our project. It provides variety of build and packaging tools like Maven, Grunt, Gradle, SBT etc.

**Maven** – A build automation tool developed by Apache organization that is used for adding dependencies and external JARS for JAVA based projects.

**SBT** – An open source tool which can run on cross platform for building Scala and JAVA based projects. An added advantage of this build tool is that it can compile Scala projects by integrating them with many test frameworks. It has a special feature called Ivy with which it provides dependency management.

#### **Spark:**

Spark is an open source framework that runs on top of Hadoop clustering systems. It is experimentally proved that Spark can process the data at least 10 times faster than when it runs on local disk. Spark has a variety of features and it supports various languages like R, SQL, Python, Scala and JAVA. In the project we had used Spark features like MLlib, streaming and data frames. Spark MLlib is a Machine Learning library which has the algorithms like Naive Bayes, Random Forest, and Alternate Least Squares etc.

## 4. Project Plan:

### 4.1 Schedule:

**Stories:** Four user stories had been created as part of Iteration 1. Here are the snapshots for the stories which are in closed and opened state.

This screenshot shows the GitHub Issues page for the repository 'npdarsini/Assist-Robot'. The search bar contains the query 'is:open issue author:npdarsini'. The results show two open issues:

- Test - Different Objects [enhancement] #4 opened 35 minutes ago by npdarsini ↑ Project - Increment 1
- Capture and feed the Images [enhancement] #3 opened 38 minutes ago by npdarsini ↑ Project - Increment 1



This screenshot shows the GitHub Issues page for the repository 'npdarsini/Assist-Robot'. The search bar contains the query 'is:closed issue author:npdarsini'. The results show two closed issues:

- Smart Watch connection with Smart Phone [enhancement] #2 opened 40 minutes ago by npdarsini ↑ Project - Increment 1
- Smart Phone connection with Android Studio [enhancement] #1 opened 42 minutes ago by npdarsini ↑ Project - Increment 1



**Stories:** Eight user issues had been created as part of Iteration 2. Here are the snapshots for the stories which are in closed and opened state.

Issues | Pull requests | Issues | Gist | ToDo

npdarsini / Assist-Robot

Issues

Filters: Is issue milestone: "Project - Increment 2" is closed

1 Open ✓ 7 Closed

- Creating a perfect training Dataset and collecting Test data (enhancement)
- Usage of Speech and image services (enhancement)
- Data Collection and analysis (enhancement)
- Establishing connection between smart watch and Robome (enhancement)
- Installation of Spark (enhancement)
- Features of Robome (question)
- Introduction to Spark (enhancement)

ProTip! Adding no label will show everything without a label.



Issues | Pull requests | Issues | Gist | ToDo

npdarsini / Assist-Robot

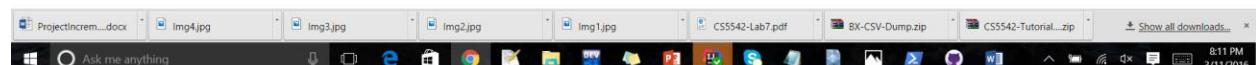
Issues

Filters: Is issue milestone: "Project - Increment 2" is open

1 Open ✓ 7 Closed

- Developing a Recommendation System (enhancement)

ProTip! Mix and match filters to narrow down what you're looking for.



**Stories:** Six user issues had been created as part of Iteration 3. Here are the snapshots for the stories which are in closed and opened state.

The screenshot shows the GitHub Issues page for the repository npdarsini/Assist-Robot. There are 17 issues listed:

- Notification to Android Device - Recognized Object (enhancement) #18 (closed)
- Training data sets - Creation with different objects (enhancement) #17 (closed)
- Exploring different Classification algorithms - Random Forest, Decision Tree (enhancement) #16 (closed)
- Image classification (enhancement) #15 (closed)
- Developing a Recommendation System (enhancement) #14 (closed)
- Introduction to ML Algorithm (enhancement) #13 (closed)
- Creating a perfect training Dataset and collecting Test data (enhancement) #11 (closed)
- Usage of Speech and image services (enhancement) #10 (closed)
- Data Collection and analysis (enhancement) #9 (closed)
- Establishing connection between smart watch and Robome (enhancement) #8 (closed)
- Linking Modules (enhancement) #20 (open)
- Classification based on the Streaming Data (enhancement) #19 (open)
- Streaming Data Collection (enhancement) #12 (open)

The screenshot shows the GitHub Issues page for the repository npdarsini/Assist-Robot. There are 3 open issues:

- Linking Modules (enhancement) #20
- Classification based on the Streaming Data (enhancement) #19
- Streaming Data Collection (enhancement) #12

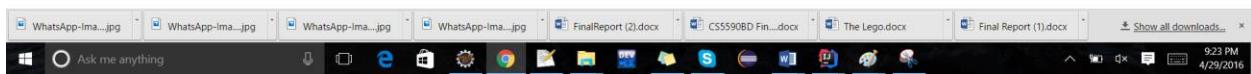
The screenshot shows a Windows taskbar with three open files:

- FinalReport.docx
- Train.zip
- modelZip

## Stories:

This screenshot shows the GitHub Issues page for the repository npdarsini/Assist-Robot. The search bar at the top contains the query "is:issue is:open". The results list six open issues, each with a title, a brief description, and a status of "In Progress".

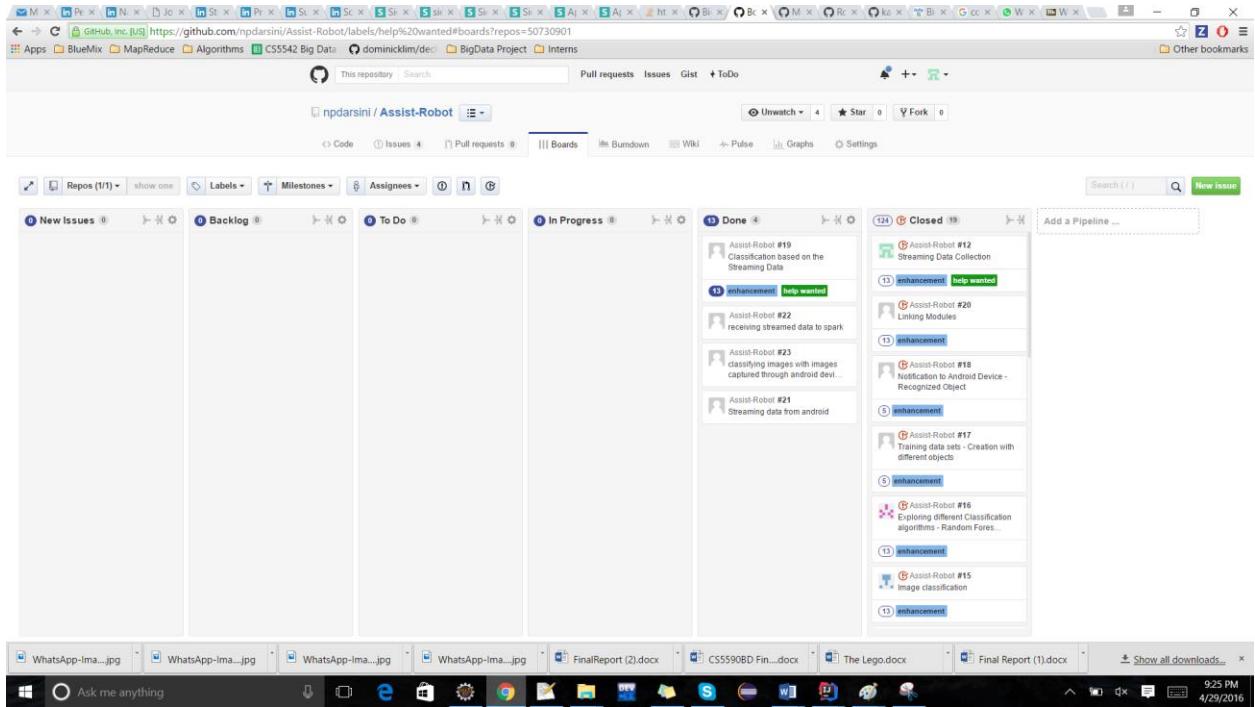
Issue Title	Description	Status
classifying images with images captured through android device	#23 opened a day ago by Deepu123start Project - Increment 4	In Progress
receiving streamed data to spark	#22 opened a day ago by Deepu123start Project - Increment 4	In Progress
Streaming data from android	#21 opened a day ago by Deepu123start Project - Increment 4	In Progress
Linking Modules enhancement	#20 opened 23 days ago by npdarsini Project - Increment 4	To Do
Classification based on the Streaming Data enhancement help wanted	#19 opened 23 days ago by npdarsini Project - Increment 4	In Progress
Streaming Data Collection enhancement help wanted	#12 opened on Mar 11 by DheerajajUmlc Project - Increment 3	In Progress



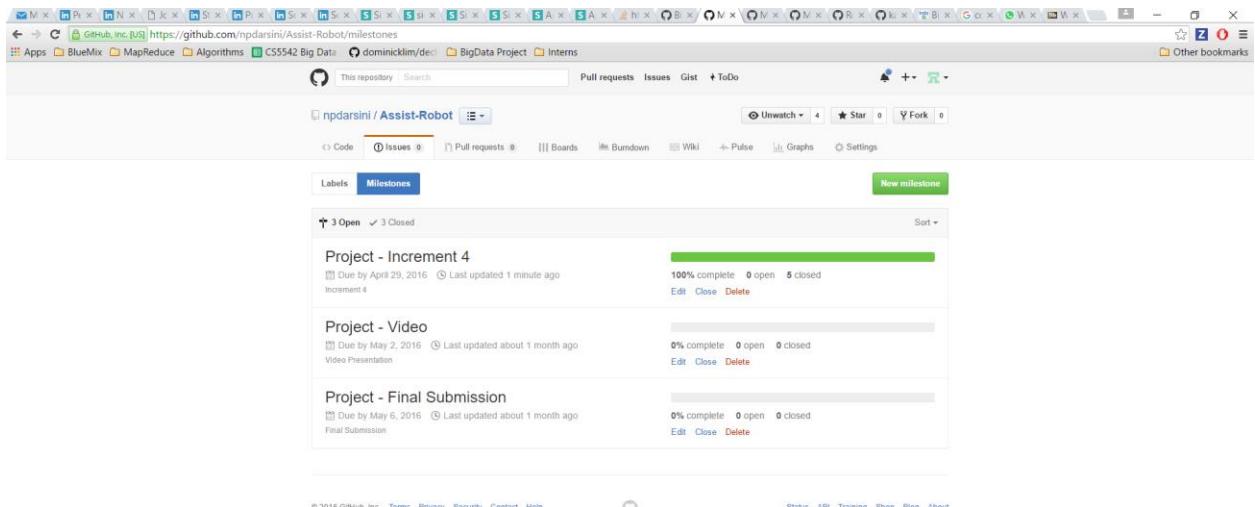
This screenshot shows the GitHub Issues page for the repository npdarsini/Assist-Robot. The search bar at the top contains the query "is:issue is:closed". The results list 23 closed issues, each with a title, a brief description, and a status of "Closed".

Issue Title	Description	Status
classifying images with images captured through android device	#23 opened a day ago by Deepu123start Project - Increment 4	Closed
receiving streamed data to spark	#22 opened a day ago by Deepu123start Project - Increment 4	Closed
Streaming data from android	#21 opened a day ago by Deepu123start Project - Increment 4	Closed
Linking Modules enhancement	#20 opened 23 days ago by npdarsini Project - Increment 4	Closed
Classification based on the Streaming Data enhancement help wanted	#19 opened 23 days ago by npdarsini Project - Increment 4	Closed
Notification to Android Device - Recognized Object enhancement	#18 opened 24 days ago by DheerajajUmlc Project - Increment 3	Closed
Training data sets - Creation with different objects enhancement	#17 opened 24 days ago by DheerajajUmlc Project - Increment 3	Closed
Exploring different Classification algorithms - Random Forest, Decision Tree enhancement	#16 opened on Mar 30 by DheerajajUmlc Project - Increment 3	Closed
Image classification enhancement	#15 opened on Mar 30 by DheerajajUmlc Project - Increment 3	Closed
Developing a Recommendation System enhancement	#14 opened on Mar 11 by DheerajajUmlc Project - Increment 2	Closed

## Board:

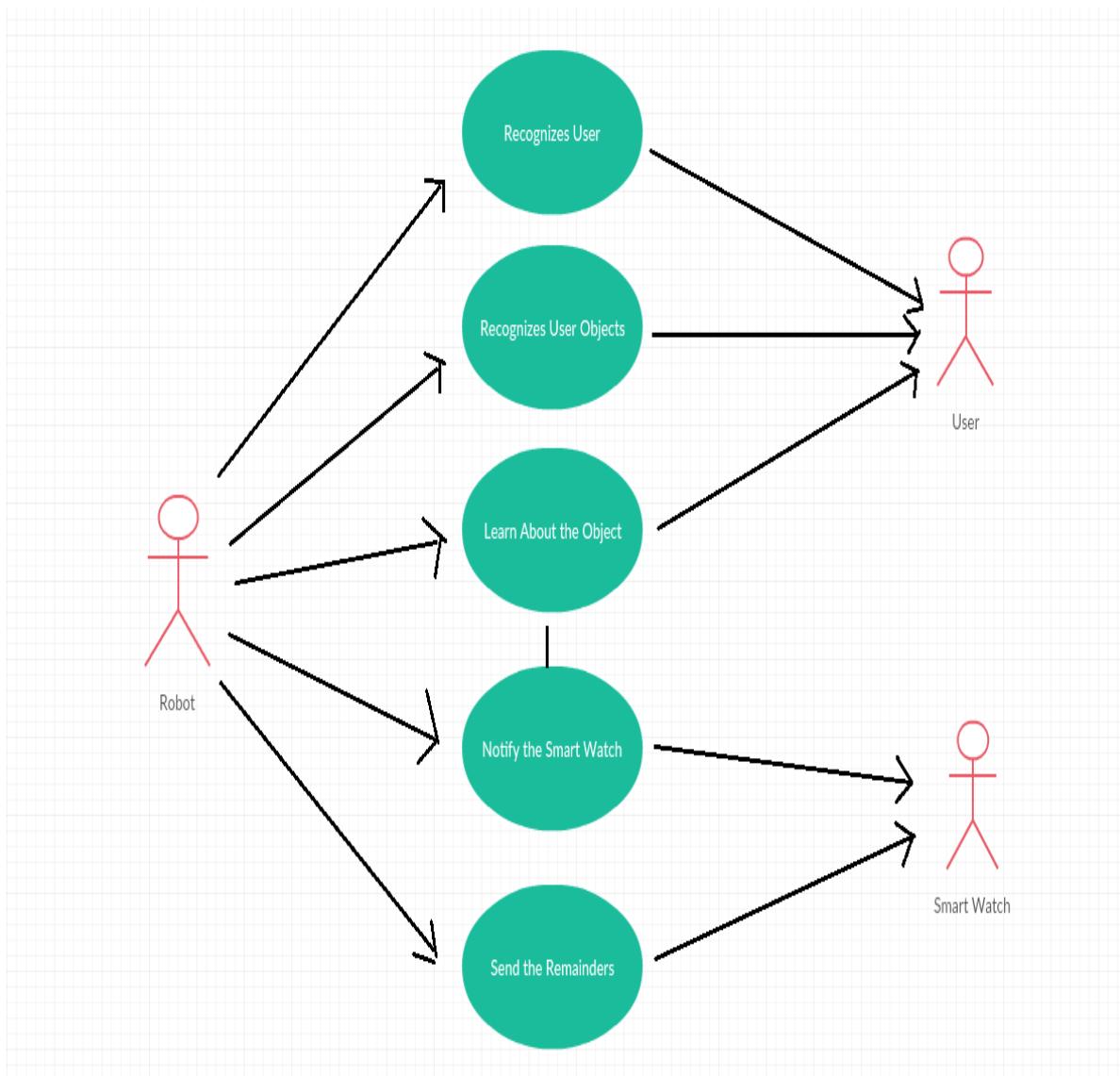


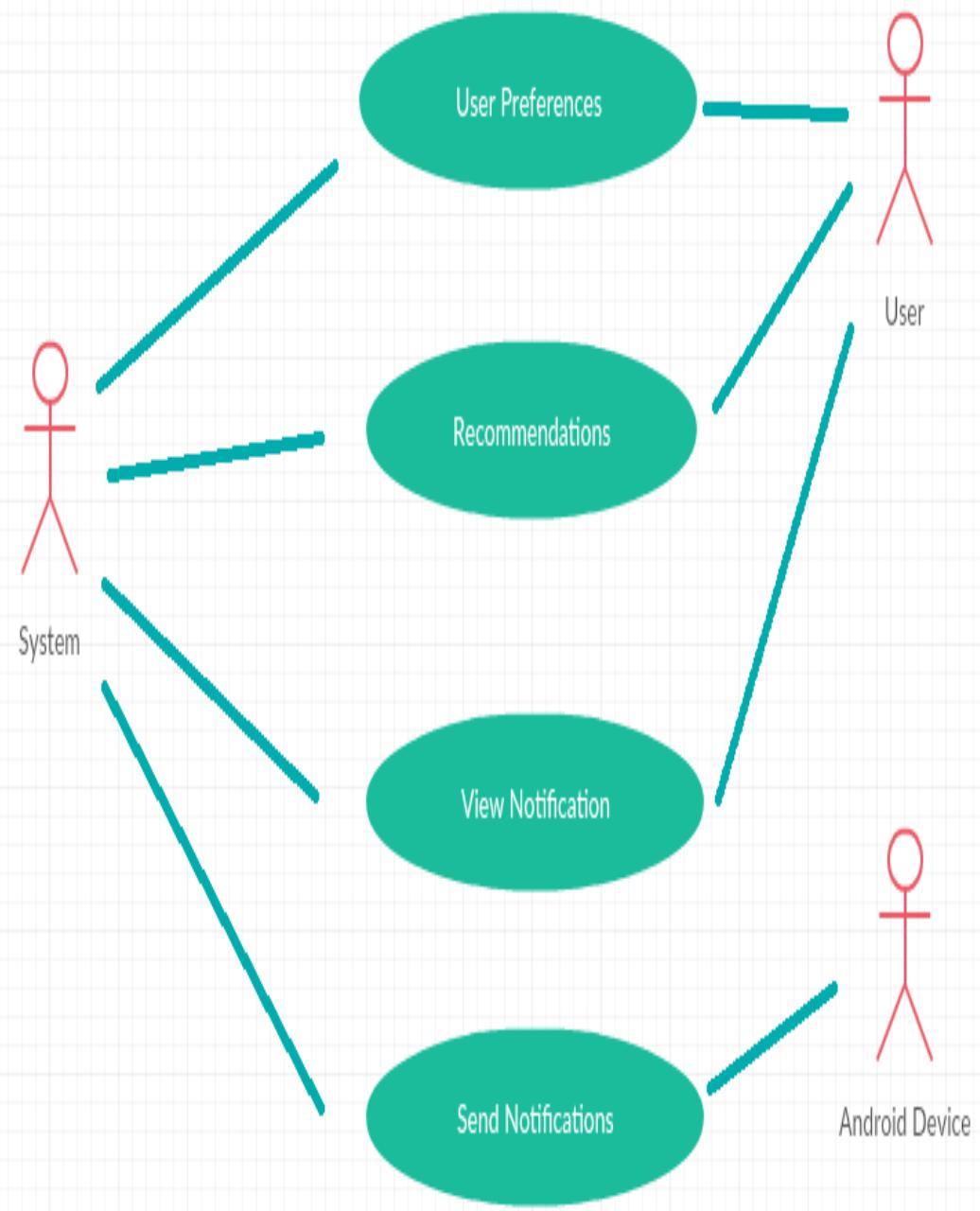
## Milestones:



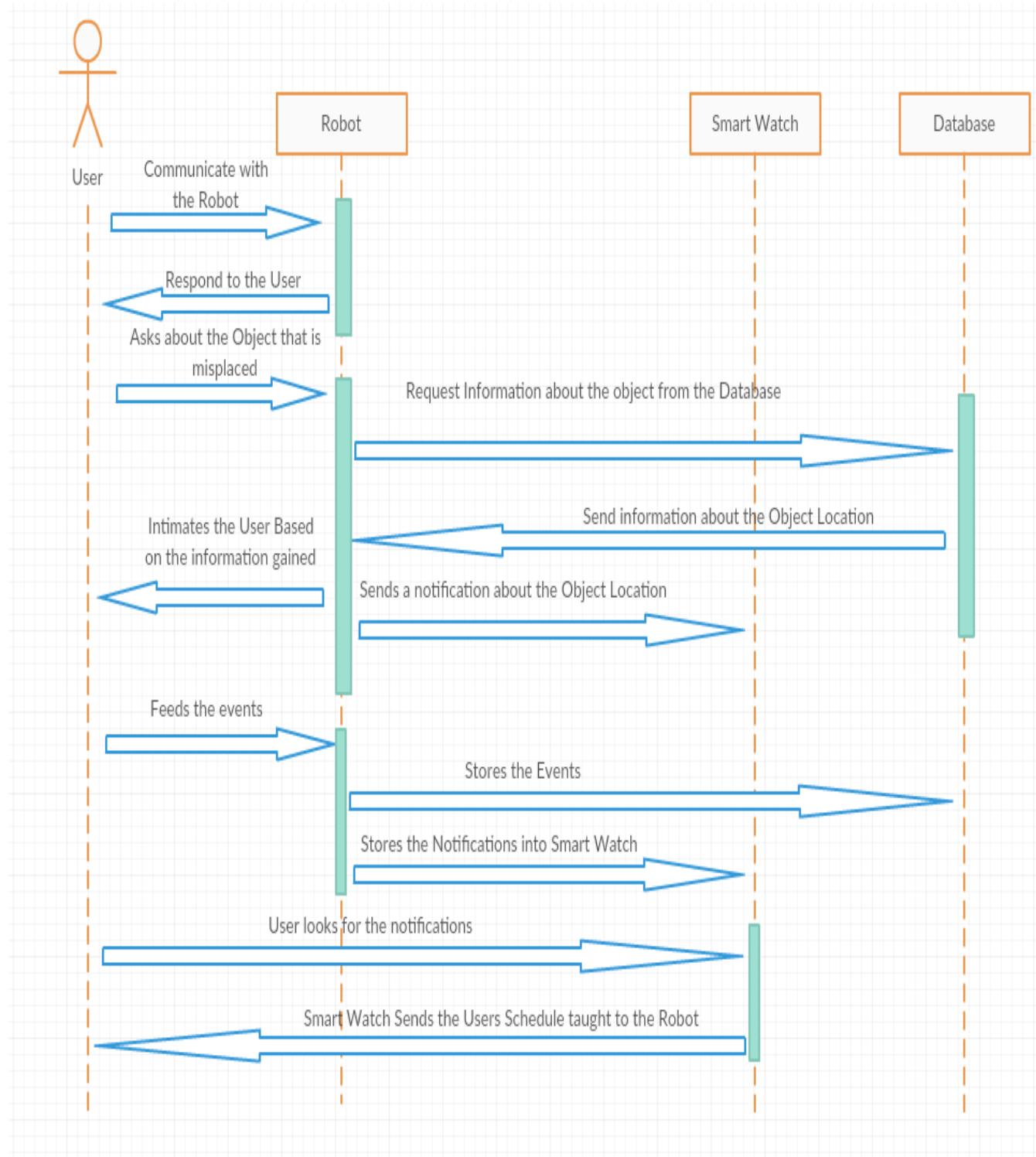
#### 4.1.1 UML Diagrams:

##### Use Case Diagram

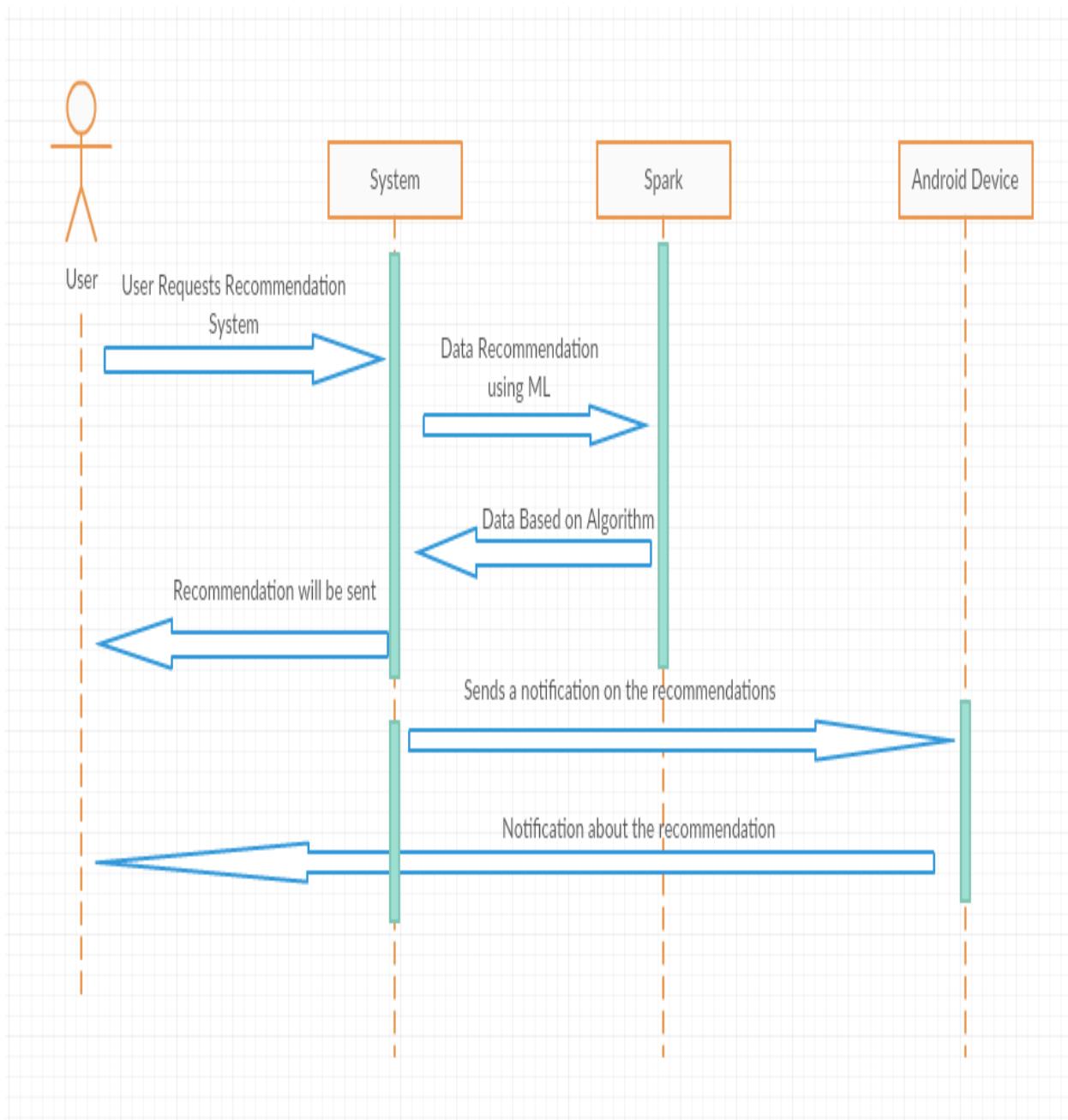




## Sequence Diagram:



## Sequence Diagram for Recommendation System



## **4.2 Project Timelines:**

<b>Increment</b>	<b>Deadline</b>
<b>Increment 1</b>	19 February 2015
<b>Increment 2</b>	11 March 2016
<b>Increment 3</b>	6 April 2016
<b>Increment 4</b>	29 April 2016
<b>Final Submission</b>	6 May 2016

### **4.2.1 Team Members:**

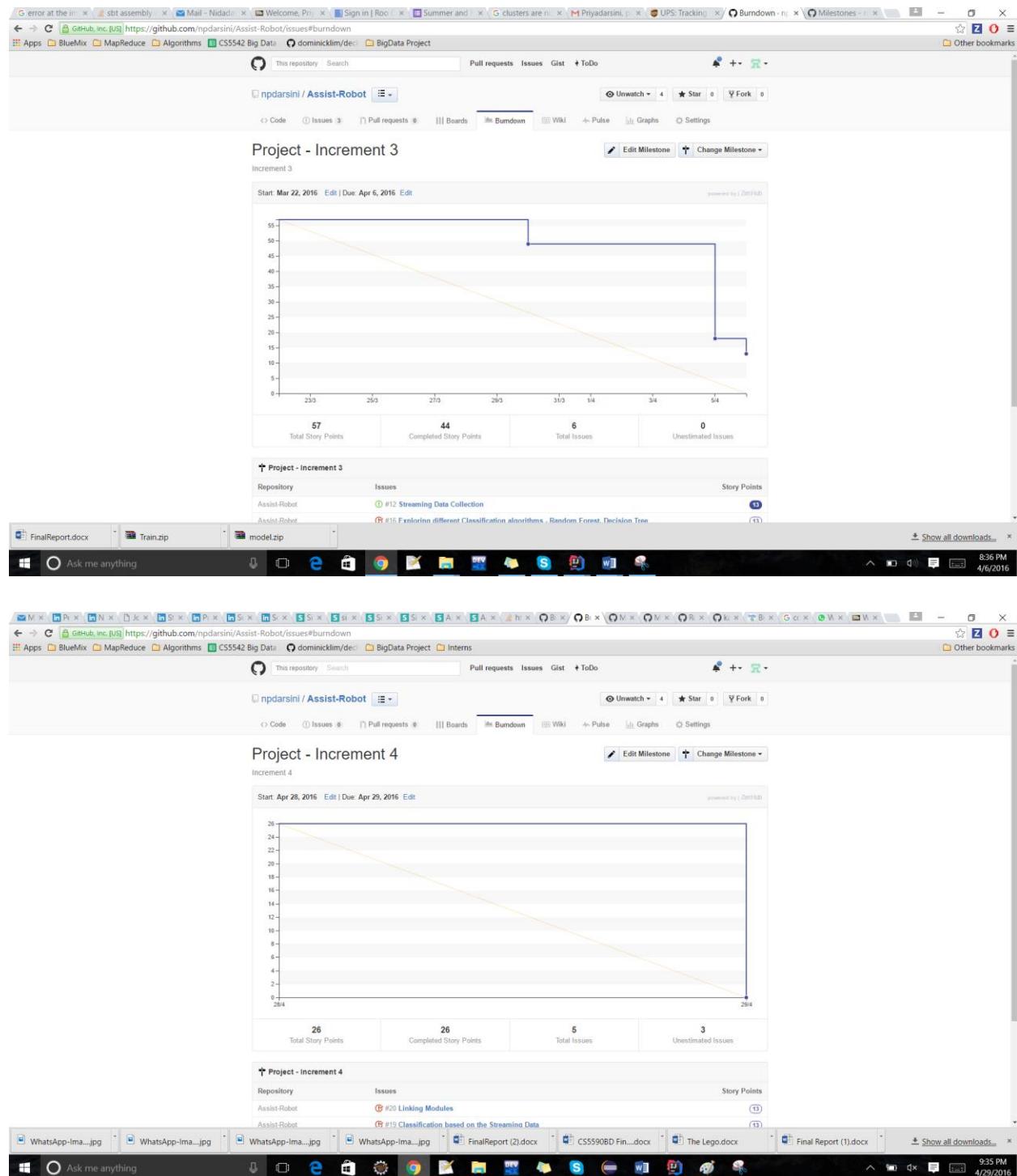
1. Priyadarsini Nidadavolu – 17
2. Deepthi Priyadarshini Penmetsa – 22
3. Dheeraja Vallabhaneni – 28
4. Tej Kumar Yentrapragada – 33

### **4.2.2 Tasks and Responsibilities:**

<b>Name</b>	<b>Responsibilities</b>
<b>Priyadarsini Nidadavolu</b>	<b>Machine Learning with Spark</b>
<b>Deepthi Priyadarshini Penmetsa</b>	<b>Spark and Hadoop Technologies</b>
<b>Tej Kumar Yentrapragada</b>	<b>Android Programming and Spark MLLib</b>
<b>Dheeraja Vallabhaneni</b>	<b>Android Programming</b>

## 4.3 Burndown Chart:

### Burndown:



## **5. Increment Report**

### **5.1 Incremental Explanations**

#### **5.1.1 Phase 1 -Existing API:**

##### **IBM Alchemy API**

This API basically performs machine learning and natural language processing techniques. Some of its features include semantic text analysis, sentimental analysis, deep learning, face detection and reorganization, speech to text and vice versa conversions etc. In this we had used this API in order to recognize the objects that we want to teach the Robot.

Achievements upon using this API – The Robot could identify basic objects like laptop, phone, bottle etc.

#### **5.1.2 Phase 2 - Recommendation System:**

In this phase we had developed two recommendation systems which can recommend the user about the popular furniture showrooms and the famous books. In this we provided the training set with user information (uid, name, ratings etc.,), furniture information (list of showrooms, location) and the book information (name, author etc.,).

The recommended notification has been sent to the android device (smart watch/phone) using Spark-Android Socket programming techniques.

#### **5.1.3 Phase 3 – Image Classification:**

In this phase we had implemented Image Classification system using Random Forest Machine Learning Classifier Algorithm. In this we had provided the training data set which has the different kinds of objects like keys, charger, watch, spectacles, phone. Basically, we generated the key descriptors and created the clusters and histograms out of it. Based on this features, the classifier predicts the image from the testing data set that has been provided.

The notification about the classified image will be sent to the android device (smart watch/phone) using Spark-Android Socket programming techniques.

#### **5.1.4 Final Phase**

This phase basically projects full pledged implementation of our project.

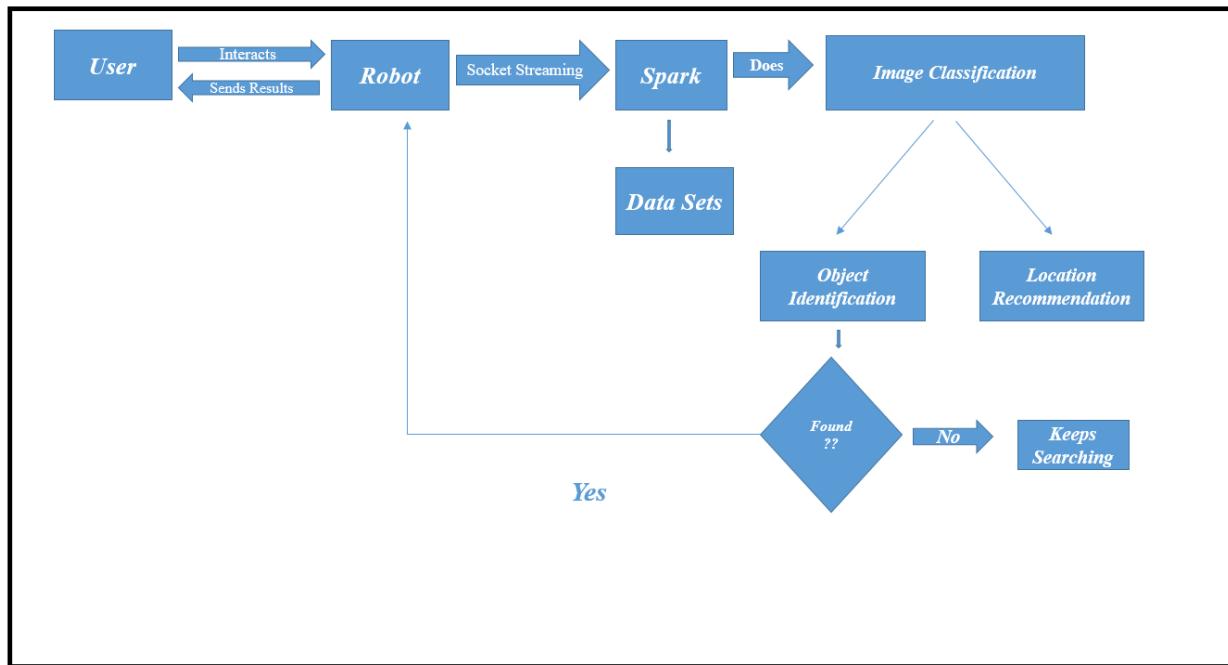
Login to the android application either by using Email ID as a user id or by facial recognition. The user is provided to register his own account or can be able to login with the existing signon. The login page is developed using Java in Android Studio IDE. Upon logging in, the camera option will be made available to the user. Android device will be capturing the images and store it in the SD card using android.os.Environment library.

A client server module has been developed to send these streaming images to spark testing directory. The Spark Server will accept the connection from android client and will receive the base64 formatted image string. At the Spark Server side, these images will be decoded and classified using Random Forest algorithm. This classified result will be send to the user device in the form of notification. Based on the image that is classified, a recommendation system has been built. If the object is out of reach, it will give the recommended places that the object can be placed based on the training data. This will also be sent to the user in the form of notification.

#### **5.2 Design of Features:**

The architecture of our system could be like the user can login to the android application with the facial recognition or with the user credentials. Image capture could be done and will be sent to spark as a streaming data via socket. Image Classification based on the training and testing data will be done and further a recommendation system has been developed. The outcomes from these algorithms will be sent to the android devices as a notification.

## Work Flow:



## System Features

The following are the features that were developed as part of Phase I:

We had used IBM's Alchemy API and able to make our Robot to detect the object and return the object name as a result.

The following are the features that are developed as part of Phase II:

We had used machine learning algorithms to develop a recommendation system. In this phase we had developed two recommendation systems with which the system will be able to suggest top rated books to the user based on his interests and the furniture showrooms which could be available for cheaper prices with the location. Basically, we had provided the training data sets and the user preferences which serves as a key inputs for the system. We were also able to connect our system to the android device to which the recommendations has been sent.

The following are the features that are developed as part of Phase III:

We had used Random Forest Classifier Algorithm to develop an Image classification system. In this phase we had provided the system with the training data set which consists the sample images of the different types of objects like keys, watch, spectacles, phone etc. We are able to create the clusters and histograms out of the provided data and was able to send a notification to the smart device about the object that has been predicted.

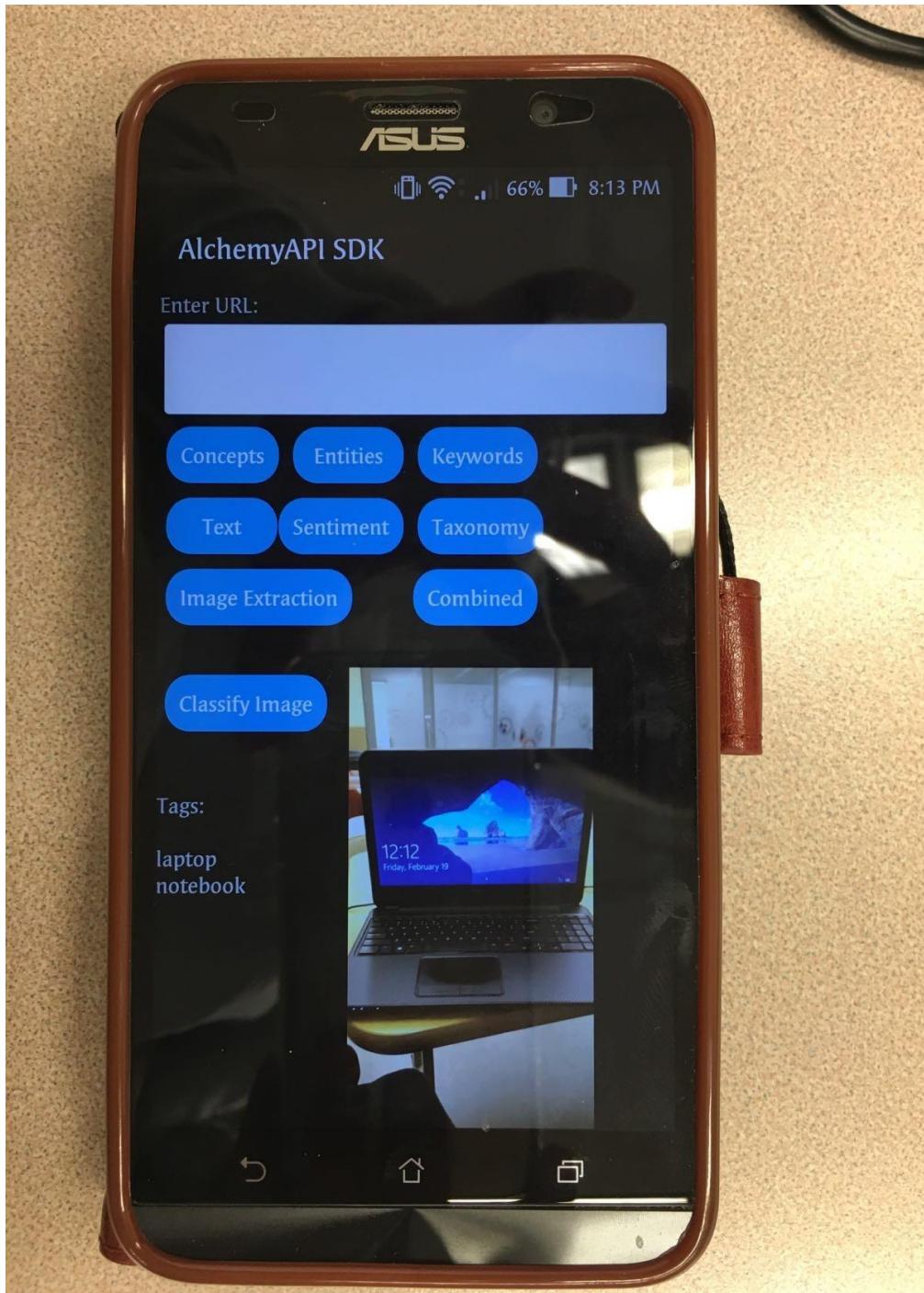
The following features has been added for the final phase:

In the previous phase Image classification is done with the static testing data set. In this phase we had added the functionality of streaming image classification. In this the streaming data will be coming from the android device to the Spark server. A login page for the android application has been developed and embedded with the functionality of facial recognition and user registration options. Upon logging in to the application the image capture option is implemented with the facility to store it to the device SD card.

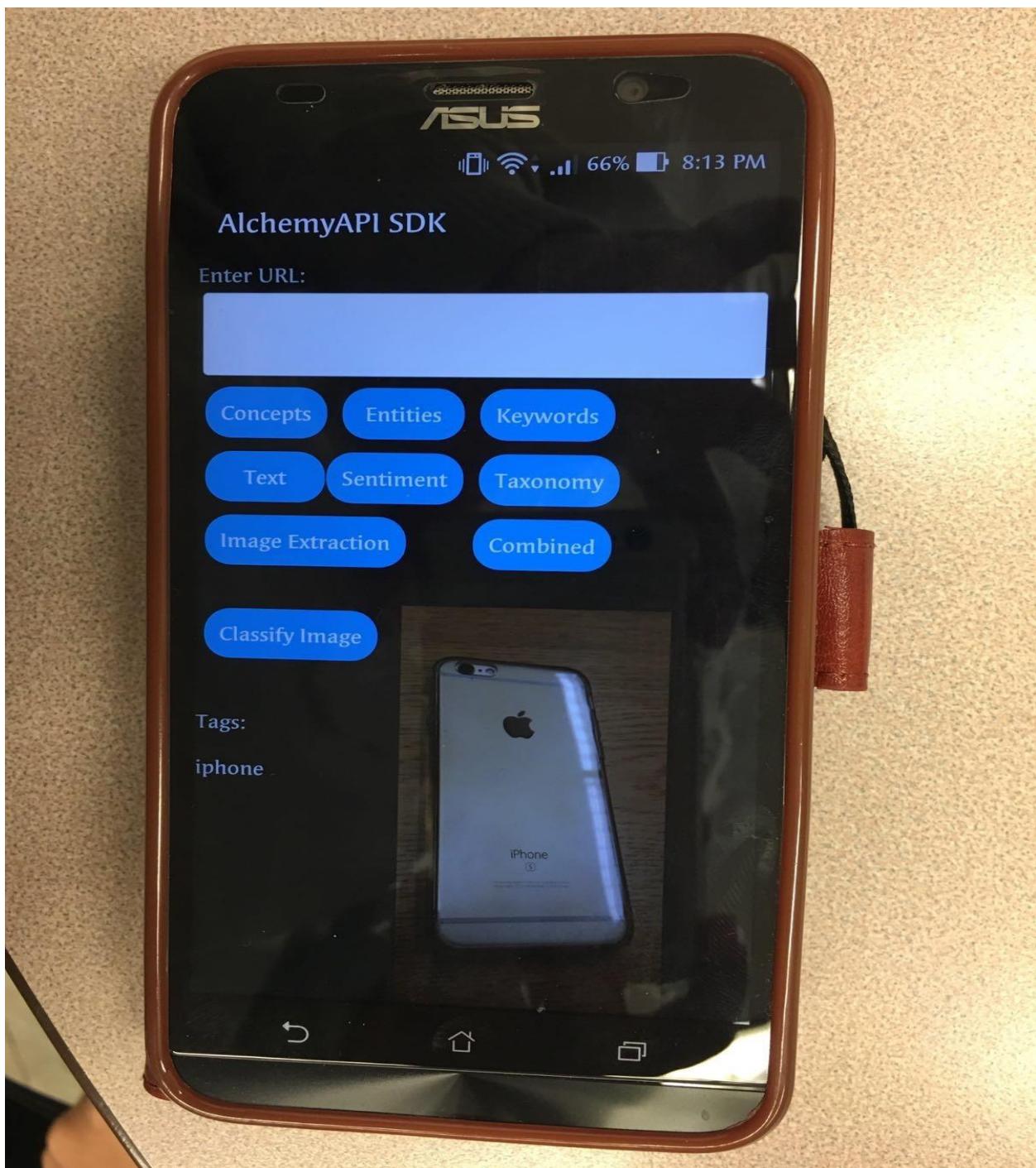
## 5.3 Implementation:

### Mobile Client Implementation – Snapshots

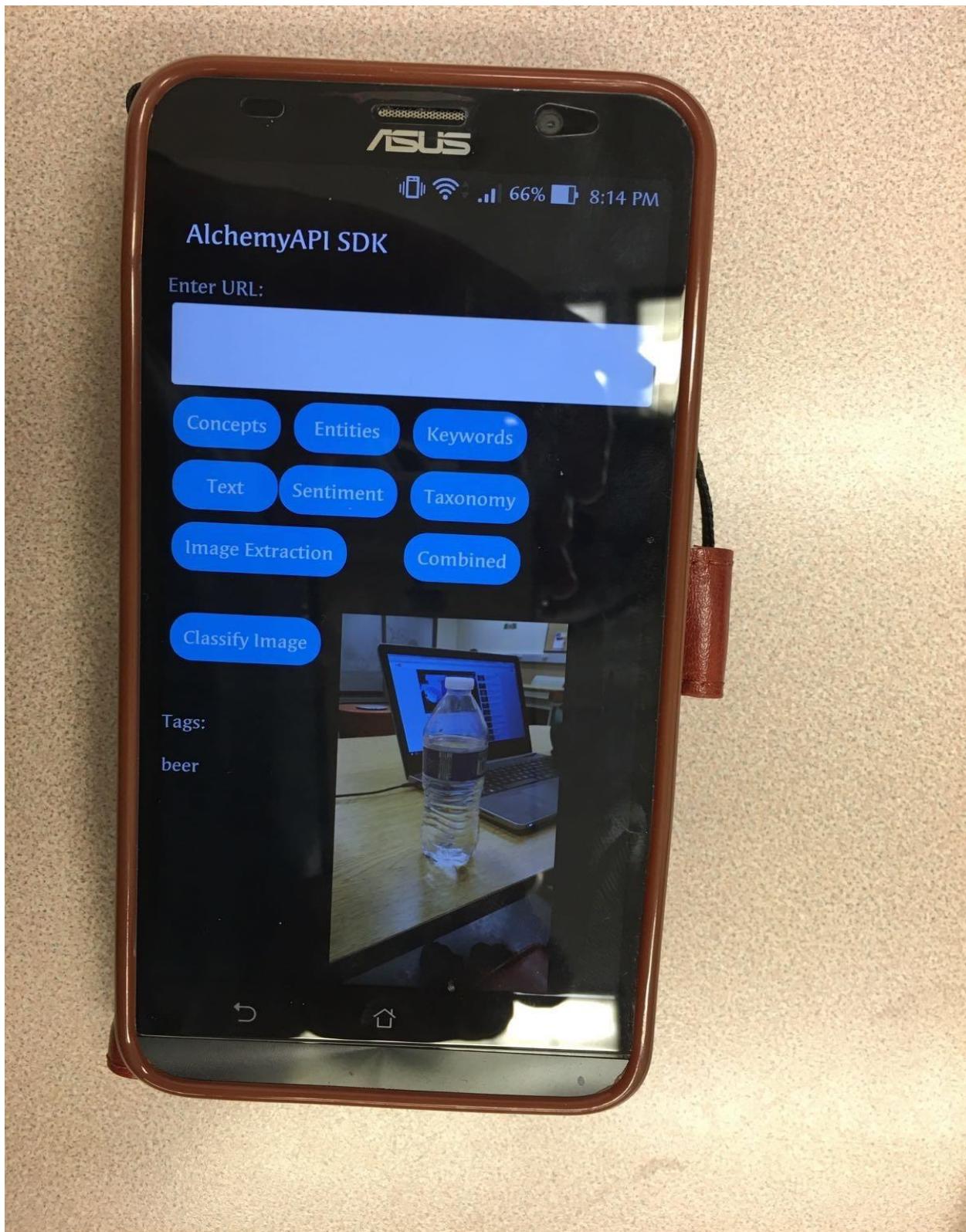
This snapshot shows us that the application is able to identify the object and names its Laptop.



This snapshot shows us that the application is capable of identifying the object and names it as an Iphone.



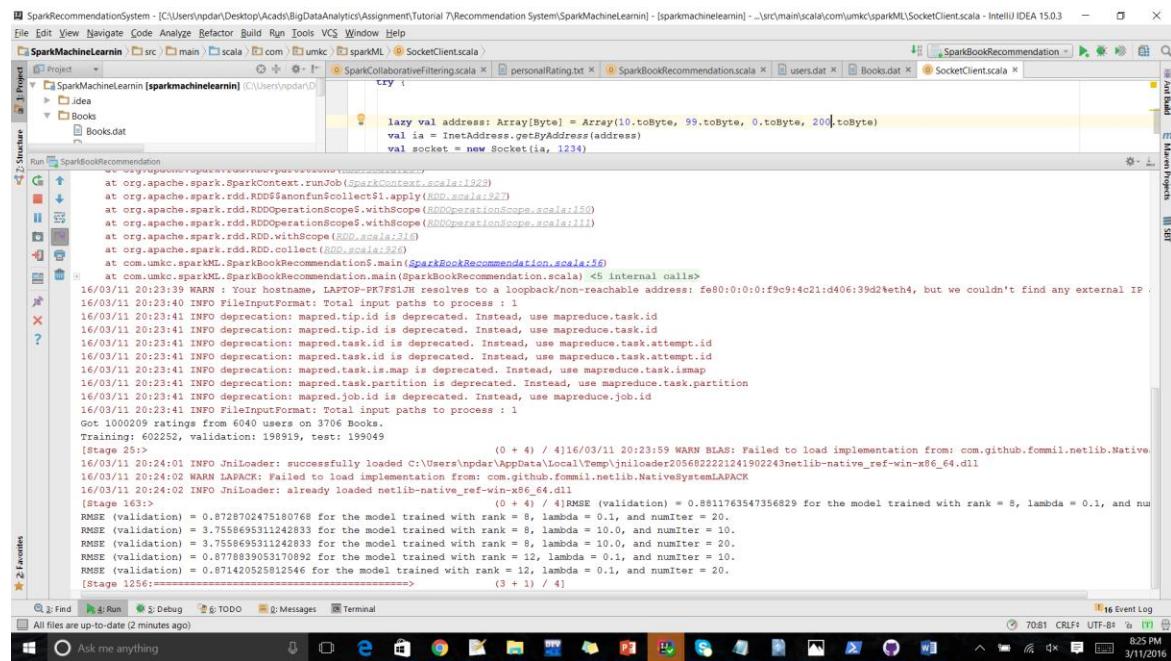
This snapshot shows us that the application is able to identify the bottle.



## Recommendation System Snapshots:

### Books Recommendation System:

Phase at which the analyzation of training data set is taking place.

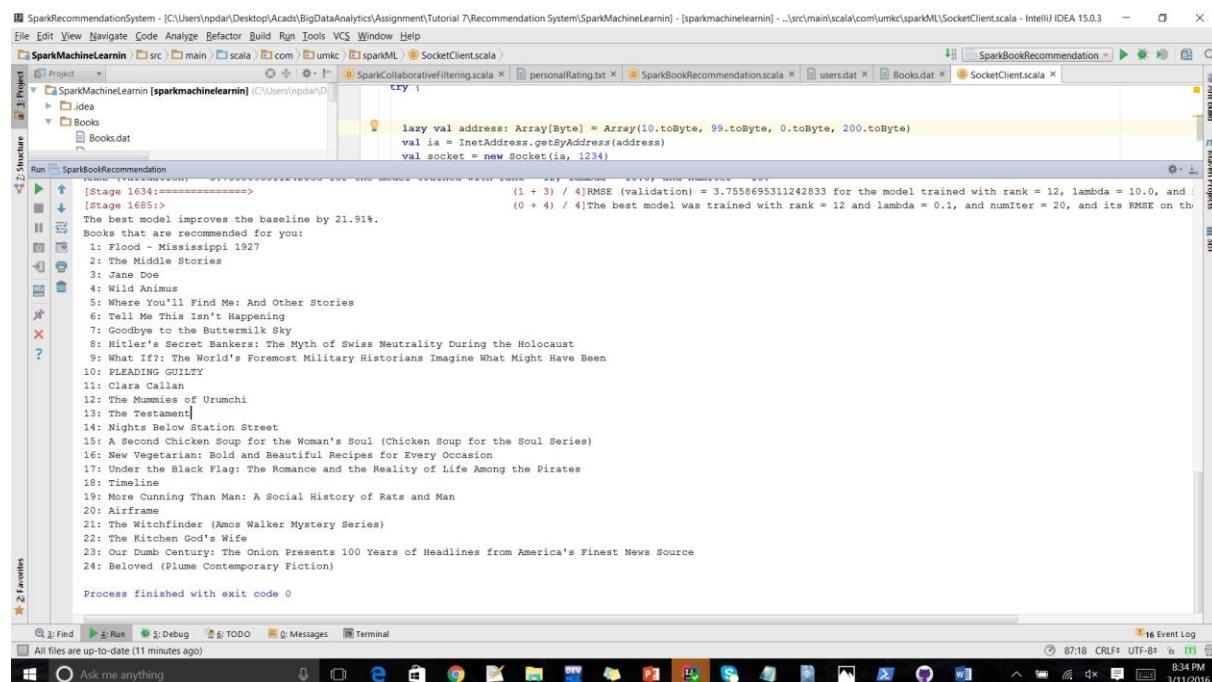


```
try {
    lazy val address: Array[Byte] = Array(10.toByte, 99.toByte, 0.toByte, 200.toByte)
    val ia = InetAddress.getByAddress(address)
    val socket = new Socket(ia, 1234)
} catch {
    case e: UnknownHostException =>
        System.out.println("WARN : Your hostname, " + e.getMessage() + " is not reachable")
        e.printStackTrace()
}
```

```
at org.apache.spark.SparkContext.runJob(sparkContext, scala$1:1929)
at org.apache.spark.rdd.RDD$$anonfun$collect$1.apply(RDD.scala$1:947)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala$1:150)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala$1:117)
at org.apache.spark.rdd.RDD.withScope(RDD.scala$1:31)
at org.apache.spark.rdd.RDD.collect(RDD.scala$1:26)
at com.umkc.sparkml.SparkBookRecommendation$.main(SparkBookRecommendation.scala$1:26)
at com.umkc.sparkml.SparkBookRecommendation.main(SparkBookRecommendation.scala$1:26)
16/03/11 20:23:39 WARN : Your hostname, Laptops-OptiPlex-5090 is not reachable, so it can't connect to any external IP.
16/03/11 20:23:40 INFO FileInputFormat: Total input paths to process : 1
16/03/11 20:23:41 INFO depreciation: mapred.tip.id is deprecated. Instead, use mapreduce.task.id
16/03/11 20:23:41 INFO depreciation: mapred.tip.id is deprecated. Instead, use mapreduce.task.id
16/03/11 20:23:41 INFO depreciation: mapred.task.id is deprecated. Instead, use mapreduce.task.attempt.id
16/03/11 20:23:41 INFO depreciation: mapred.task.id is deprecated. Instead, use mapreduce.task.attempt.id
16/03/11 20:23:41 INFO depreciation: mapred.task.is.map is deprecated. Instead, use mapreduce.task.ismap
16/03/11 20:23:41 INFO depreciation: mapred.partition is deprecated. Instead, use mapreduce.task.partition
16/03/11 20:23:41 INFO depreciation: mapred.job.id is deprecated. Instead, use mapreduce.job.id
16/03/11 20:23:41 INFO FileInputFormat: Total input paths to process : 1
Got 1000209 ratings from 6040 users on 3706 Books.
Training: 602252, validation: 198919, test: 199049
[Stage 25]> (0 + 4) / 416/03/11 20:23:59 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.Native
16/03/11 20:24:01 INFO JniLoader: successfully loaded C:\Users\npdar\AppData\Local\Temp\jniloader20562221241902243netlib-native_ref-win-x86_64.dll
16/03/11 20:24:02 WARN LAPACK: Failed to load implementation from: com.github.fommil.netlib.NativeSystemLAPACK
16/03/11 20:24:02 INFO JniLoader: already loaded netlib-native_ref-win-x86_64.dll
[Stage 163]> (0 + 4) / 4RMSE (validation) = 0.8728702475180768 for the model trained with rank = 8, lambda = 0.1, and numIter = 20.
RMSE (validation) = 0.8728702475180768 for the model trained with rank = 8, lambda = 0.1, and numIter = 20.
RMSE (validation) = 3.7558695311242833 for the model trained with rank = 8, lambda = 10.0, and numIter = 10.
RMSE (validation) = 3.7558695311242833 for the model trained with rank = 8, lambda = 10.0, and numIter = 20.
RMSE (validation) = 0.8778839053170892 for the model trained with rank = 12, lambda = 0.1, and numIter = 10.
RMSE (validation) = 0.8714205281812546 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
[Stage 1256]> (3 + 1) / 4
```

Ask me anything 70:81 CRLF1 UTF-8 8:25 PM 3/11/2016

### Recommended Books



```
try {
    lazy val address: Array[Byte] = Array(10.toByte, 99.toByte, 0.toByte, 200.toByte)
    val ia = InetAddress.getByAddress(address)
    val socket = new Socket(ia, 1234)
} catch {
    case e: UnknownHostException =>
        System.out.println("WARN : Your hostname, " + e.getMessage() + " is not reachable")
        e.printStackTrace()
}
```

```
[Stage 1634:=====]
[Stage 1685:>
The best model improves the baseline by 21.9%.
Books that are recommended for you:
1: Flood - Mississippi 1927
2: The Middle Stories
3: Jane Doe
4: Wild Animals
5: Where You'll Find Me: And Other Stories
6: Tell Me This Isn't Happening
7: Goodbye to the Buttermilk Sky
8: Hitler's Secret Bankers: The Myth of Swiss Neutrality During the Holocaust
9: What If?: The World's Foremost Military Historians Imagine What Might Have Been
10: FEADING GUILTY
11: Clara Callan
12: The Mummies of Urumchi
13: The Testament
14: Nights Below Station Street
15: A Second Chicken Soup for the Woman's Soul (Chicken Soup for the Soul Series)
16: New Vegetarian: Bold and Beautiful Recipes for Every Occasion
17: Under the Black Flag: The Romance and the Reality of Life Among the Pirates
18: Timeline
19: More Cunning Than Man: A Social History of Rats and Man
20: Airframe
21: The Witchfinder (Amos Walker Mystery Series)
22: The Kitchen God's Wife
23: Our Dumb Century: The Onion Presents 100 Years of Headlines from America's Finest News Source
24: Beloved (Plume Contemporary Fiction)
```

```
Process finished with exit code 0
```

Ask me anything 87:18 CRLF1 UTF-8 8:34 PM 3/11/2016

We had also sent the recommended books to the smart phone as a notification:



## Furniture Malls Recommendation System

```
16/03/10 07:16:28 INFO FileInputFormat: Total input paths to process : 1
Got 1000209 ratings from 6040 users on 3706 FurnitureMalls.
Training: 602252, validation: 198919, test: 199049
[Stage 25:>                                         (0 + 4) / 4]16/03/10 07:16:25 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeDenseMatrix64F
16/03/10 07:16:28 INFO JniLoader: successfully loaded C:\Users\DEEPU\AppData\Local\Temp\jniloader1966739238328846810netlib-native_ref-win-x86_64.dll
[Stage 27:>                                         (0 + 4) / 4]16/03/10 07:16:33 WARN LAPACK: Failed to load implementation from: com.github.fommil.netlib.NativeDenseMatrix64F
16/03/10 07:16:33 INFO JniLoader: already loaded netlib-native_ref-win-x86_64.dll
RMSE (validation) = 0.8815801121709103 for the model trained with rank = 8, lambda = 0.1, and numIter = 10.
RMSE (validation) = 0.8726203182715503 for the model trained with rank = 8, lambda = 0.1, and numIter = 20.
RMSE (validation) = 3.7558695311242833 for the model trained with rank = 8, lambda = 10.0, and numIter = 10.
[Stage 950:>                                         (0 + 4) / 4]RMSE (validation) = 3.7558695311242833 for the model trained with rank = 8, lambda = 10.0, and numIter = 20.
[Stage 1125:=====                                         (3 + 1) / 4]RMSE (validation) = 0.8772284010651425 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
RMSE (validation) = 0.8710227453579589 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
[Stage 1595:>                                         (0 + 4) / 4]RMSE (validation) = 3.7558695311242833 for the model trained with rank = 12, lambda = 10.0, and numIter = 20.
RMSE (validation) = 3.7558695311242833 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 = 20, and its RMSE on the test set is 0.8690494992690084.
The best model improves the baseline by 21.95%.
```

This Screenshot shows the Furniture Malls recommended to you:

The screenshot shows the IntelliJ IDEA 15.0.1 interface with the following details:

- Project Structure:** Shows the project structure with files like SparkCollaborativeFiltering.scala, SparkMovieRecommendation.scala, FurnitureShopRating.txt, sample\_libsvm\_data.txt, and test.data.txt.
- Code Editor:** Displays the Scala code for SparkMovieRecommendation, which includes a call to sys.exit(1).
- Run Output:** Shows the command-line output of the application execution. It includes the following log entries:
  - INFO deprecation: mapred.task.partition is deprecated. Instead, use mapreduce.task.partition
  - INFO deprecation: mapred.job.id is deprecated. Instead, use mapreduce.job.id
  - INFO FileInputFormat: Total input paths to process : 1
  - Got 1000209 ratings from 6040 users on 3706 FurnitureMalls.
  - Training: 602252, validation: 198919, test: 199049
  - [Stage 25:> (0 + 4) / 4]16/03/10 07:16:25 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeDenseMatrix64F
  - [Stage 27:> (0 + 4) / 4]16/03/10 07:16:33 INFO JniLoader: successfully loaded C:\Users\DEEPU\AppData\Local\Temp\jniloader1966739238328846810netlib-native\_ref-win-x86\_64.dll
  - [Stage 950:> (0 + 4) / 4]RMSE (validation) = 3.7558695311242833 for the model trained with rank = 8, lambda = 10.0, and numIter = 20.
  - [Stage 1125:===== (3 + 1) / 4]RMSE (validation) = 0.8772284010651425 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
  - [Stage 1595:> (0 + 4) / 4]RMSE (validation) = 3.7558695311242833 for the model trained with rank = 12, lambda = 10.0, and numIter = 20.
  - RMSE (validation) = 3.7558695311242833 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 its RMSE on the test set is 0.8690494992690084.
  - The best model improves the baseline by 21.95%.
- Bottom Status Bar:** Shows "Compilation completed successfully in 31s 964ms (7 minutes ago)" and other system information.

The notification has been sent to the android mobile which shows the recommended furniture list.

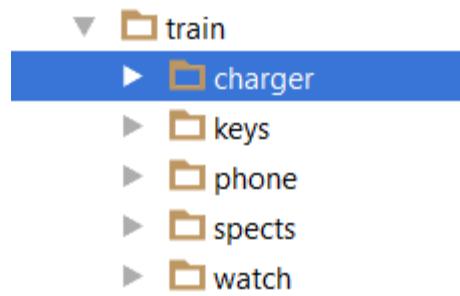


## Image Classification

The Training dataset is full with images of objects the user want to identify which were got misplaced. And the test dataset is the streaming images which are captured by the robot while trying to identify the misplaced object. So when the robot finds the image, it classifies the image and notifies about the object to the users device.

Screenshots of the image classification:

Train data:



Identifying Key descriptors from Training dataset

```
// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTress = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100
```

Key Descriptors 1556 x 128  
Key Descriptors 1334 x 128  
Key Descriptors 1000 x 128  
Key Descriptors 692 x 128  
Key Descriptors 515 x 128  
Key Descriptors 781 x 128  
Key Descriptors 20587 x 128  
Key Descriptors 24357 x 128  
Key Descriptors 2381 x 128  
Key Descriptors 24357  
Key Descriptors 17111 x 128  
Key Descriptors 27961 x 128  
Key Descriptors 3518 x 128  
Key Descriptors 27961

## Cluster formation:

```
// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100
```

16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 64,986B for [point, values, list, element] DOUBLE: 12,800 values, 102,912B raw, 64,939B comp, 1 pages, encodings: [RLE, PLAIN]

16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 53B for [point, indices, list, element] INT32: 100 values, 14B raw, 30B comp, 1 pages, encodings: [RLE, PLAIN]

16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 64,817B for [point, values, list, element] DOUBLE: 12,800 values, 102,912B raw, 64,770B comp, 1 pages, encodings: [RLE, PLAIN]

16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 66,094B for [point, values, list, element] DOUBLE: 12,800 values, 102,912B raw, 66,047B comp, 1 pages, encodings: [RLE, PLAIN]

16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 62B for [point, type] INT32: 100 values, 10B raw, 28B comp, 1 pages, encodings: [BIT\_PACKED, PLAIN\_DICTIONARY, RLE], data type: INT32

16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 50B for [point, values, list, element] INT32: 100 values, 7B raw, 27B comp, 1 pages, encodings: [BIT\_PACKED, RLE, PLAIN]

16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 53B for [point, indices, list, element] INT32: 100 values, 14B raw, 30B comp, 1 pages, encodings: [RLE, PLAIN]

16/04/06 17:57:26 INFO ColumnChunkPageWriteStore: written 65,899B for [point, values, list, element] DOUBLE: 12,800 values, 102,912B raw, 65,852B comp, 1 pages, encodings: [RLE, PLAIN]

16/04/06 17:57:26 INFO FileOutputCommitter: Saved output of task 'attempt\_201604061757\_0058\_m\_000002\_0' to file:/C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial/parquet/part-r-00002

16/04/06 17:57:26 INFO FileOutputCommitter: Saved output of task 'attempt\_201604061757\_0058\_m\_000000\_0' to file:/C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial/parquet/part-r-00000

16/04/06 17:57:26 INFO FileOutputCommitter: Saved output of task 'attempt\_201604061757\_0058\_m\_000001\_0' to file:/C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial/parquet/part-r-00001

16/04/06 17:57:26 INFO FileOutputCommitter: Saved output of task 'attempt\_201604061757\_0058\_m\_000003\_0' to file:/C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial/parquet/part-r-00003

Save Clusters to data3/model/clusters

16/04/06 17:57:26 INFO FileInputFormat: Total input paths to process : 1

16/04/06 17:57:27 INFO ParquetFileReader: Initiating action with parallelism: 5

16/04/06 17:57:27 INFO ParquetFileReader: Initiating action with parallelism: 5

16/04/06 17:57:27 INFO ParquetFileReader: Initiating action with parallelism: 5

16/04/06 17:57:27 INFO ParquetFileReader: Initiating action with parallelism: 5

16/04/06 17:57:27 INFO deprecation: mapred.min.split.size is deprecated. Instead, use mapreduce.input.fileinputformat.split.minsize

16/04/06 17:57:27 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskK

16/04/06 17:57:27 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskK

16/04/06 17:57:27 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskK

16/04/06 17:57:27 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskK

16/04/06 17:57:28 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.

16/04/06 17:57:28 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.

16/04/06 17:57:28 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.

16/04/06 17:57:28 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.

Histogram generation based on size specified:

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The `IApp.scala` file is open in the editor.
- Code Editor:** The code defines a `CategoricalFeaturesInfo` object with various parameters like `numClasses`, `numTrees`, `featureSubsetStrategy`, `impurity`, `maxDepth`, and `maxBins`.
- Run Configuration:** A configuration named `IApp` is selected, showing a histogram of the input data.
- Terminal Output:** The output of the run shows the execution of the `train` command, which generates histograms for each feature. The histograms are large, with sizes ranging from approximately 400 to 9,351,296 bins. The output includes the number of bins, the range of the bins, and the count of data points falling into each bin.
- Bottom Status Bar:** Shows the message "Compilation completed successfully in 7s 29ms (20 minutes ago)".

## Model has been created based on the Random Forest Classifier Algorithm

The screenshot shows the IntelliJ IDEA interface. The top navigation bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help. The title bar indicates the project is 'Image\_Classification' and the current file is 'IPApp.scala'. A message in the top right says 'Platform and Plugin Updates' and 'IntelliJ IDEA is ready to update'. The left sidebar shows the project structure with 'Project' selected, displaying 'data3' and its subfolders: 'model', 'test', 'keys', 'phone', 'specs', 'watch', and 'train'. The main code editor window contains Scala code for a Random Forest classifier. The bottom run log window shows the command-line output of the application running, including logs from 'ColumnChunkPageWriteStore', 'FileOutputCommitter', and 'ParquetFileReader'. The bottom status bar shows the date and time as '4/6/2016 6:09 PM'.

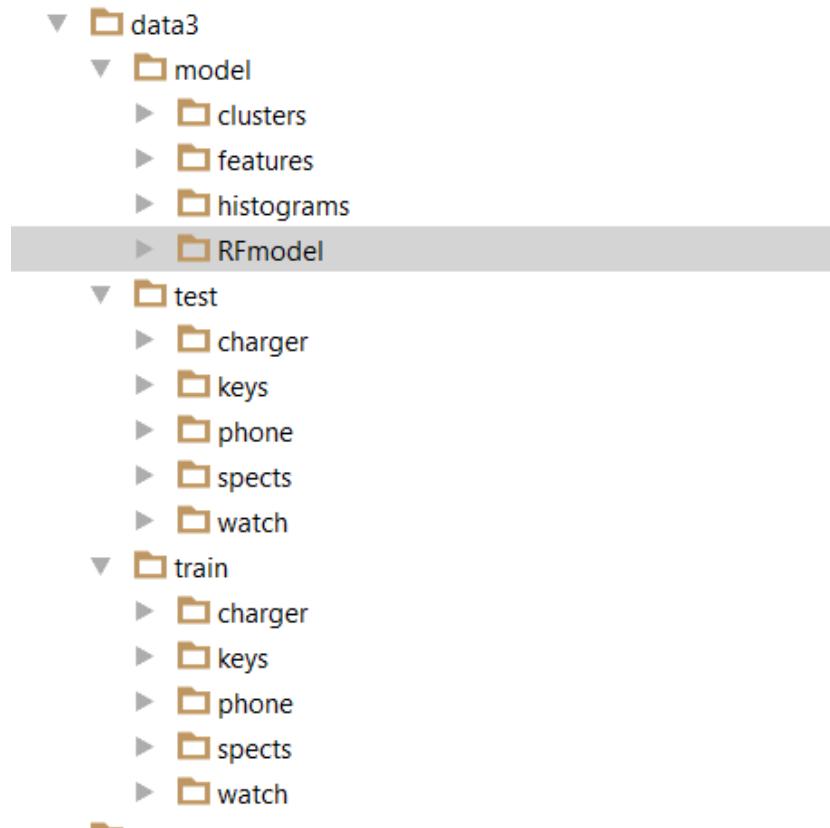
```

// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100

val numTrees = 4 to(5, 1)
val strategies = List("all", "sqrt", "log2", "onethird")
val maxDepths = 3 to(6, 1)
val impurities = List("gini", "entropy")

var bestModel: Option[RandomForestModel] = None
var bestErr = 1.0
val bestParams = new mutable.HashMap[Any, Any]()
var bestNumTrees = 0
var bestNumTreesSubSet = ""

```



## Confusion Matrix:

Screenshot of IntelliJ IDEA 15.0.3 showing the code and output for generating a confusion matrix.

```

// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100

```

Output window shows the following data:

```

1.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0
0.0 0.0 1.0 0.0
0.0 0.0 0.0 1.0
1.0
numTrees 5 featureSubsetStrategy onethird impurity entropy maxDepth 5
16/04/06 17:59:04 WARN DecisionTreeMetadata: DecisionTree reducing maxBins from 100 to 15 (= number of training instances)
Test Error = 0.0
=====
Confusion matrix =====
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
0.0 0.0 0.0 1.0
1.0
numTrees 5 featureSubsetStrategy onethird impurity entropy maxDepth 6
16/04/06 17:59:05 WARN DecisionTreeMetadata: DecisionTree reducing maxBins from 100 to 15 (= number of training instances)
Test Error = 0.25
=====
Confusion matrix =====
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
0.75
Best Err 0.0
Best params (featureSubsetStrategy,all) (numTrees,4) (maxDepth,3) (impurity,gini)
16/04/06 17:59:05 WARN DecisionTreeMetadata: DecisionTree reducing maxBins from 100 to 15 (= number of training instances)
16/04/06 17:59:06 INFO FileOutputCommitter: Saved output of task 'attempt_201604061759_1280_m_000000_4915' to file: C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tut...
16/04/06 17:59:07 INFO CodecConfig: Compression: GZIP
16/04/06 17:59:07 INFO ParquetOutputFormat: Parquet block size to 134217728

```

Event Log shows 7754 LF+ UTF-8 messages at 6:10 PM on 4/6/2016.

Screenshot of IntelliJ IDEA 15.0.3 showing the code and output for generating a confusion matrix.

```

(0..,4)
(3,0,3)
(2,0,3)
(2,0,2)
(3,0,2)
(0,0,2)
(3,0,2)
(2,0,2)
(3,0,2)
(0,0,2)
(2,0,1)
(1,0,1)
(0,0,1)
(3,0,1)
(0,0,1)
(1,0,1)
(1,0,1)
(1,0,0)
(3,0,0)
(0,0,0)
(0,0,0)
(0,0,0)
(1,0,0)
(2,0,0)
0.3333333333333333
===== Confusion matrix =====
3.0 2.0 1.0 1.0 0.0
2.0 3.0 1.0 1.0 0.0
2.0 0.0 2.0 3.0 0.0
0.0 0.0 1.0 2.0 0.0
4.0 1.0 1.0 0.0 0.0
0.3333333333333333
16/04/06 18:00:51 INFO RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
16/04/06 18:00:51 INFO RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.

Process finished with exit code 0

```

Event Log shows 2577.1 LF+ UTF-8 messages at 6:12 PM on 4/6/2016.

## Predicted Results Snapshots from the testing dataset

Screenshot of IntelliJ IDEA showing the code for `testImageClassification` in `IPApp.scala`. The code reads images from a directory, classifies them using a model, and prints the results.

```
// testImageClassification(sc)

val testImages = sc.wholeTextFiles(s"${IPSettings.INPUT_DIR}/*/*.jpg")
val testImagesArray = testImages.collect()
var predictionLabels = List[String]()
testImagesArray.foreach(f => {
    val splitStr = f._1.split("/")
    val predictedClass: Double = classifyImage(sc, splitStr(1))
    val segments = f._1.split("/")
    val cat = segments(segments.length - 2)
    val GivenClass = IMAGE_CATEGORIES.indexOf(cat)
    println(s"Predicting test image: " + cat + " as " + IMAGE_CATEGORIES(predictedClass.toInt))
    predictionLabels = predictedClass + ":" + GivenClass :: predictionLabels
})

Predicting test image : charger as charger
file:///C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial 9/CS5542 - Tutorial 9 Code/Image_Classification/data3/test/charger/charger4.jpg
16/04/06 17:46:37 INFO FileInputFormat: Total input paths to process : 1
16/04/06 17:46:37 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:46:37 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:46:37 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:46:37 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 17:46:38 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskAttempt
16/04/06 17:46:38 INFO ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskAttempt
16/04/06 17:46:38 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.
```

Screenshot of IntelliJ IDEA showing the code for `testImageClassification` in `IPApp.scala`. The code reads images from a directory, classifies them using a model, and prints the results.

```
// Empty categoricalFeaturesInfo indicates all features are continuous.
val numClasses = 10
val categoricalFeaturesInfo = Map[Int, Int]()
// val numTrees = 10 // Use more in practice.
// val featureSubsetStrategy = "auto" // Let the algorithm choose.
// val impurity = "gini"
// val maxDepth = 4
val maxBins = 100

val numOFTrees = 4 to(5, 1)
val strategies = List("all", "sqrt", "log2", "onethird")
val maxDepths = 3 to(6, 1)
val impurities = List("gini", "entropy")

var bestModel: Option[RandomForestModel] = None
var bestErr = 1.0
val bestParams = new mutable.HashMap[Any, Any]()
var bestnumTrees = 0
var bestFeatureSubSet = ""

Predicting test image : spectre as spectre
file:///C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial 9/CS5542 - Tutorial 9 Code/Image_Classification/data3/test/spectre/spectre1.jpg
16/04/06 18:00:26 INFO InternalParquetRecordReader: at row 0, reading next block
16/04/06 18:00:26 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 7 records.
16/04/06 18:00:26 INFO InternalParquetRecordReader: at row 0, reading next block
16/04/06 18:00:26 INFO CodecPool: Got brand-new decompressor [.gz]
16/04/06 18:00:26 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7
16/04/06 18:00:26 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7
16/04/06 18:00:26 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 7 records.
16/04/06 18:00:26 INFO InternalParquetRecordReader: at row 0, reading next block
16/04/06 18:00:26 INFO CodecPool: Got brand-new decompressor [.gz]
16/04/06 18:00:26 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7
Predicting test image : spectre as spectre
file:///C:/Users/npdar/Desktop/Acds/BigDataAnalytics/Assignment/Tutorial 9/CS5542 - Tutorial 9 Code/Image_Classification/data3/test/spectre/spectre2.jpg
16/04/06 18:00:27 INFO FileInputFormat: Total input paths to process : 1
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
16/04/06 18:00:27 INFO ParquetFileReader: Initiating action with parallelism: 5
```

Image\_Classification - [C:\Users\npdar\Desktop\Acads\BigDataAnalytics\Assignment\Assignment\Tutorial 9\CS5542 - Tutorial 9 Code]\Image\_Classification] - [image\_classification] - ..\src\main\scala\IPApp.scala - IntelliJ IDEA 15.0.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Image\_Classification > data3 > model >

Project Packages Project Files Problems

IPApp.scala x IPSettings.scala x build.sbt x

// Empty categoricalFeaturesInfo indicates all features are continuous.  
 val numClasses = 10  
 val categoricalFeaturesInfo = Map[Int, Int]()  
 // val numTrees = 10 // Use more in practice.  
 // val featureSubsetStrategy = "auto" // Let the algorithm choose.  
 // val impurity = "gini"  
 // val maxDepth = 4  
 val maxBins = 100

val numOFTrees = 4 to(5, 1)  
 val strategies = List("all", "sqrt", "log2", "onethird")  
 val maxDepths = 3 to(6, 1)  
 val impurities = List("gini", "entropy")

var bestModel: Option[RandomForestModel] = None  
 var bestErr = 1.0  
 val bestParams = new mutable.HashMap[Any, Any]()  
 var bestnumTrees = 0  
 var bestFeatureSubSet = ""

Run IPApp

16/04/06 18:00:18 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 7 records.  
 16/04/06 18:00:18 INFO InternalParquetRecordReader: at row 0. reading next block  
 16/04/06 18:00:18 INFO InternalParquetRecordReader: block read in memory in 0 ms. row count = 7  
 16/04/06 18:00:18 INFO CodecPool: Got brand-new decompressor [.gz]  
 16/04/06 18:00:18 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7  
 Predicting test image : phone as phone  
 16/04/06 18:00:18 INFO FileInputFormat: Total input paths to process : 1  
 16/04/06 18:00:18 INFO ParquetFileReader: Initiating action with parallelism: 5  
 16/04/06 18:00:18 INFO ParquetFileReader: Initiating action with parallelism: 5  
 16/04/06 18:00:18 INFO ParquetFileReader: Initiating action with parallelism: 5  
 16/04/06 18:00:18 INFO ParquetFileReader: Initiating action with parallelism: 5  
 16/04/06 18:00:18 INFO ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA  
 16/04/06 18:00:18 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA  
 16/04/06 18:00:18 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 100 records.  
 16/04/06 18:00:18 INFO InternalParquetRecordReader: at row 0. reading next block  
 16/04/06 18:00:18 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA  
 16/04/06 18:00:18 INFO CodecPool: Got brand-new decompressor [.gz]

Find Run Terminal TODO Event Log

Compilation completed successfully in 7s 29ms (16 minutes ago)

2153.72 LF+ UTF-8 6:05 PM 4/6/2016

Image\_Classification - [C:\Users\npdar\Desktop\Acads\BigDataAnalytics\Assignment\Assignment\Tutorial 9\CS5542 - Tutorial 9 Code]\Image\_Classification] - [image\_classification] - ..\src\main\scala\IPApp.scala - IntelliJ IDEA 15.0.3

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Image\_Classification > data3 > model >

Project Packages Project Files Problems

IPApp.scala x IPSettings.scala x build.sbt x

// Empty categoricalFeaturesInfo indicates all features are continuous.  
 val numClasses = 10  
 val categoricalFeaturesInfo = Map[Int, Int]()  
 // val numTrees = 10 // Use more in practice.  
 // val featureSubsetStrategy = "auto" // Let the algorithm choose.  
 // val impurity = "gini"  
 // val maxDepth = 4  
 val maxBins = 100

val numOFTrees = 4 to(5, 1)  
 val strategies = List("all", "sqrt", "log2", "onethird")  
 val maxDepths = 3 to(6, 1)  
 val impurities = List("gini", "entropy")

var bestModel: Option[RandomForestModel] = None  
 var bestErr = 1.0  
 val bestParams = new mutable.HashMap[Any, Any]()  
 var bestnumTrees = 0  
 var bestFeatureSubSet = ""

Run IPApp

16/04/06 17:59:53 INFO CodecPool: Got brand-new decompressor [.gz]  
 16/04/06 17:59:53 INFO InternalParquetRecordReader: block read in memory in 0 ms. row count = 7  
 16/04/06 17:59:53 INFO CodecPool: Got brand-new decompressor [.gz]  
 16/04/06 17:59:53 INFO InternalParquetRecordReader: block read in memory in 18 ms. row count = 7  
 16/04/06 17:59:53 INFO ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA  
 16/04/06 17:59:53 INFO InternalParquetRecordReader: RecordReader initialized will read a total of 7 records.  
 16/04/06 17:59:53 INFO CodecPool: Got brand-new decompressor [.gz]  
 16/04/06 17:59:53 INFO InternalParquetRecordReader: block read in memory in 1 ms. row count = 7  
 Predicting test image : keys as keys  
 16/04/06 17:59:53 INFO FileInputFormat: Total input paths to process : 1  
 16/04/06 17:59:53 INFO ParquetFileReader: Initiating action with parallelism: 5  
 16/04/06 17:59:53 INFO ParquetFileReader: Initiating action with parallelism: 5  
 16/04/06 17:59:53 INFO ParquetFileReader: Initiating action with parallelism: 5  
 16/04/06 17:59:53 INFO ParquetFileReader: Initiating action with parallelism: 5  
 16/04/06 17:59:53 INFO ParquetRecordReader: Initiating action with parallelism: 5  
 16/04/06 17:59:53 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA  
 16/04/06 17:59:53 WARN ParquetRecordReader: Can not initialize counter due to context is not a instance of TaskInputOutputContext, but is org.apache.hadoop.mapreduce.task.TaskA

Find Run Terminal TODO Event Log

Compilation completed successfully in 7s 29ms (17 minutes ago)

1765.55 LF+ UTF-8 6:07 PM 4/6/2016



47% 9:39 PM

Hello World!

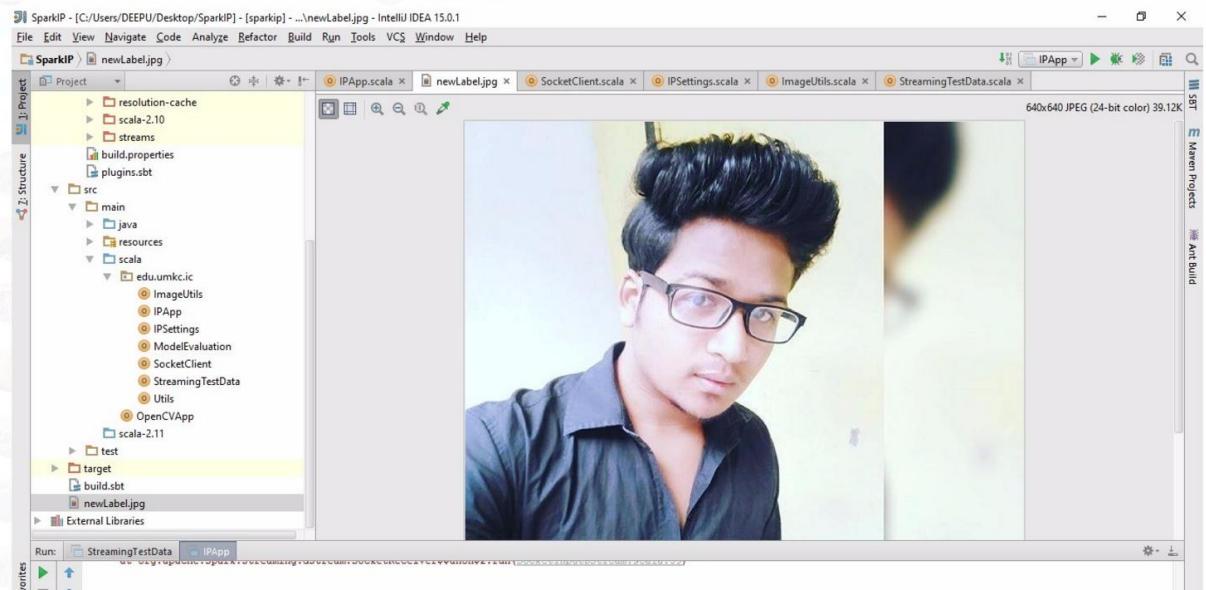
**NOTIFY WEARABLE**

I'm waiting here: 1234

SiteLocalAddress: 192.168.0.21

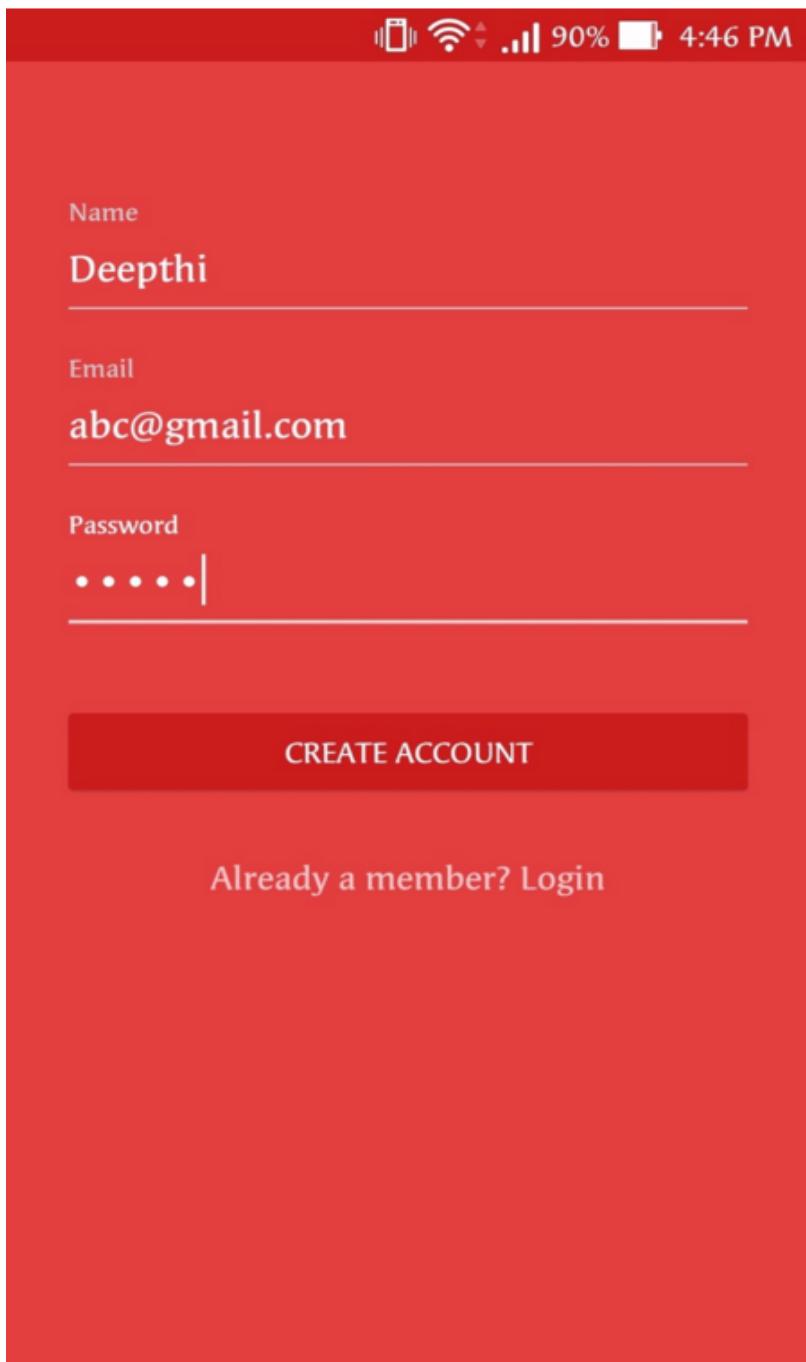
#1 from /192.168.0.23:53290

Predicted test image:spects/nreplayed: Hello from  
Android, you are #1

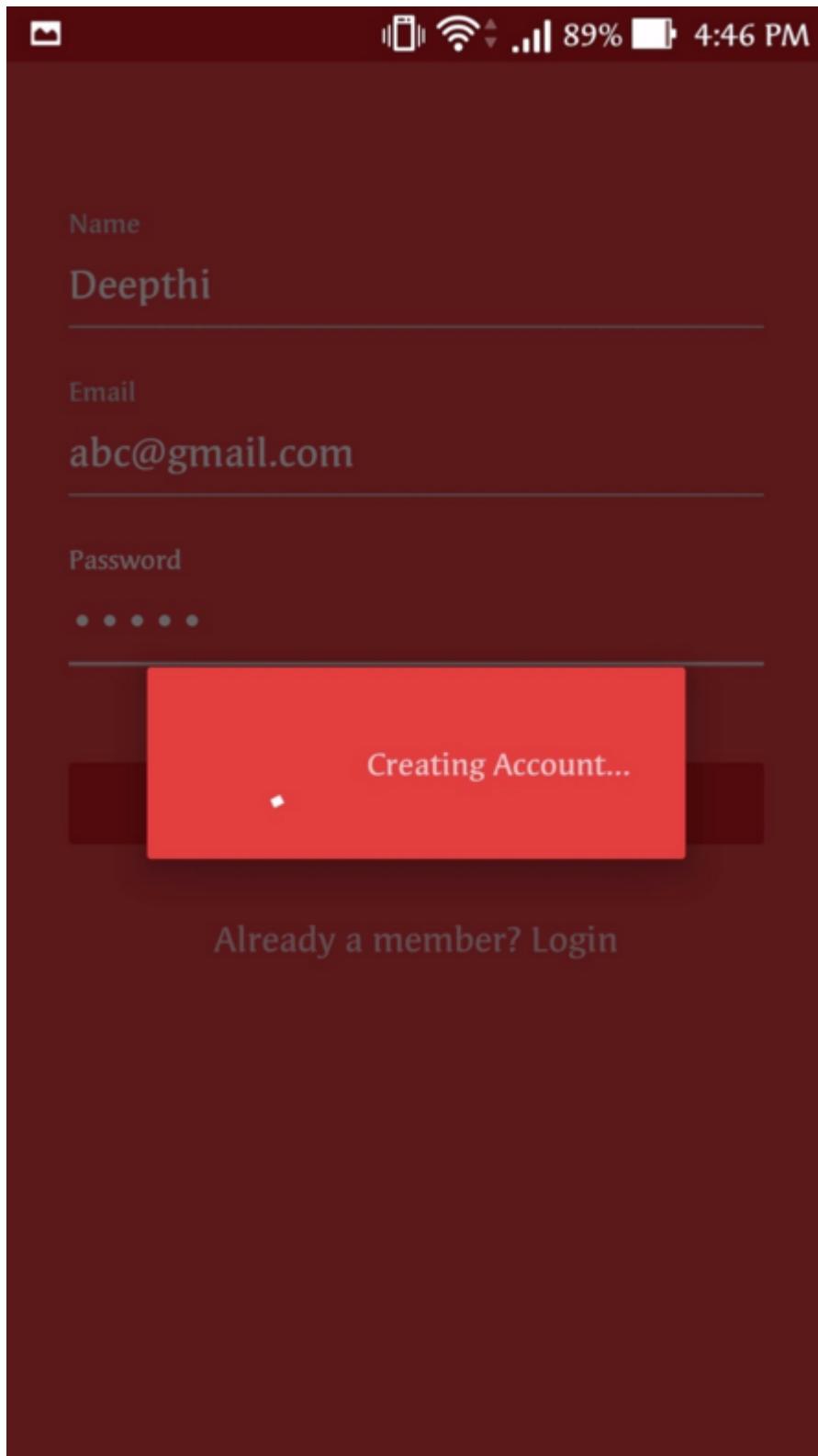


**Final Phase Snapshots:**

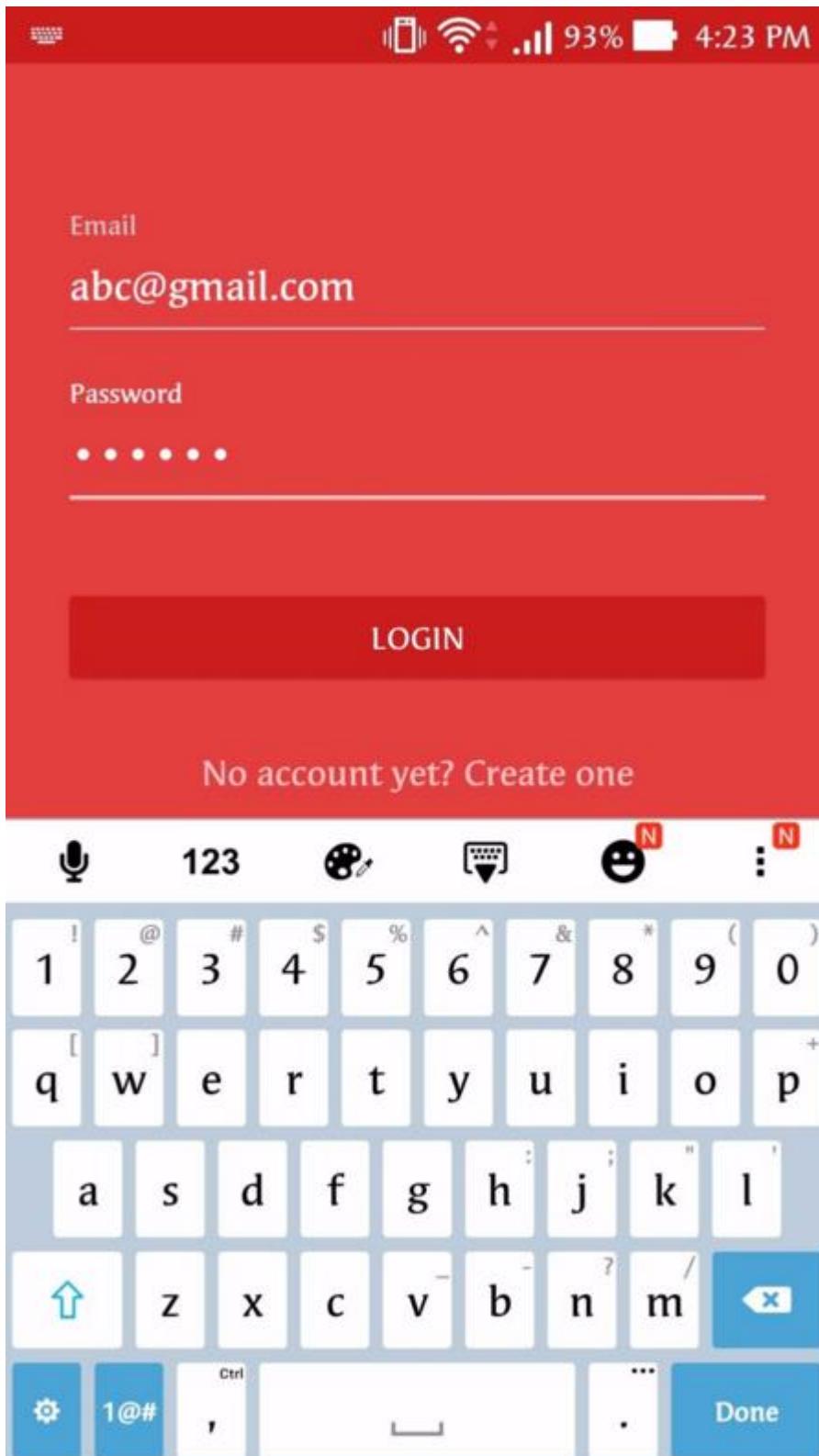
Android application - Registration page



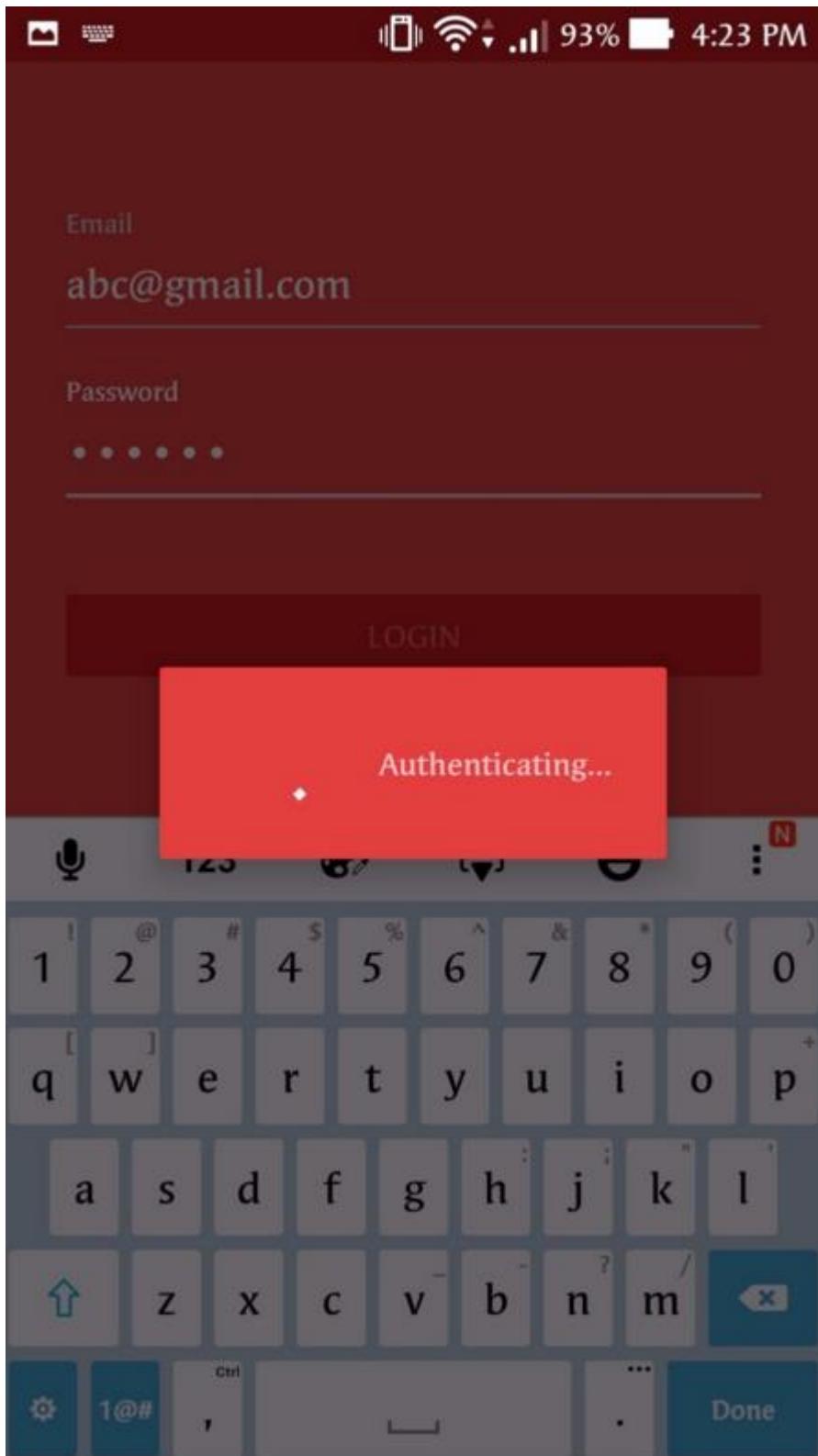
Snapshot to show that the account is being creating.



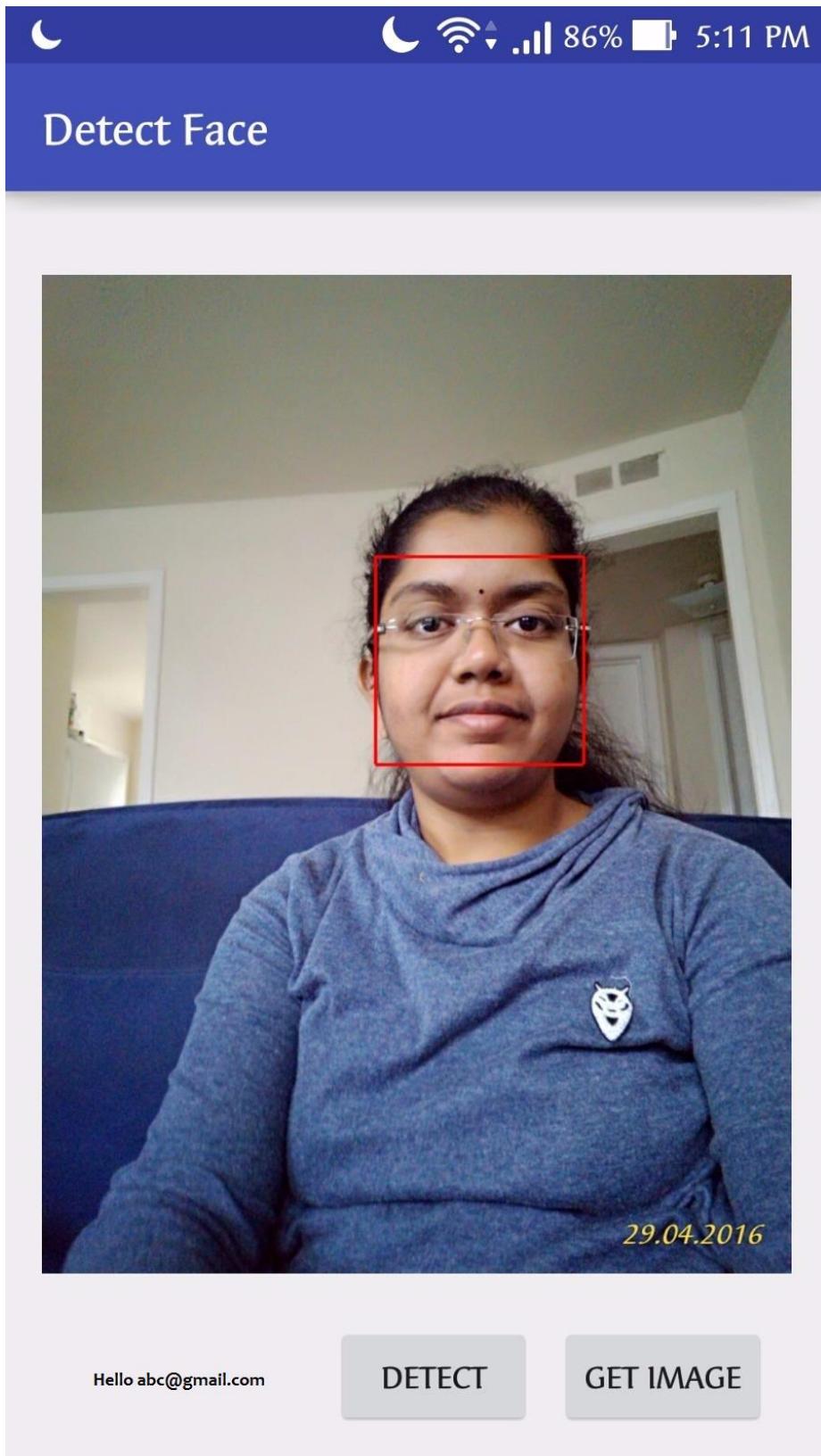
User login page after registration.



## Account authentication Snapshot



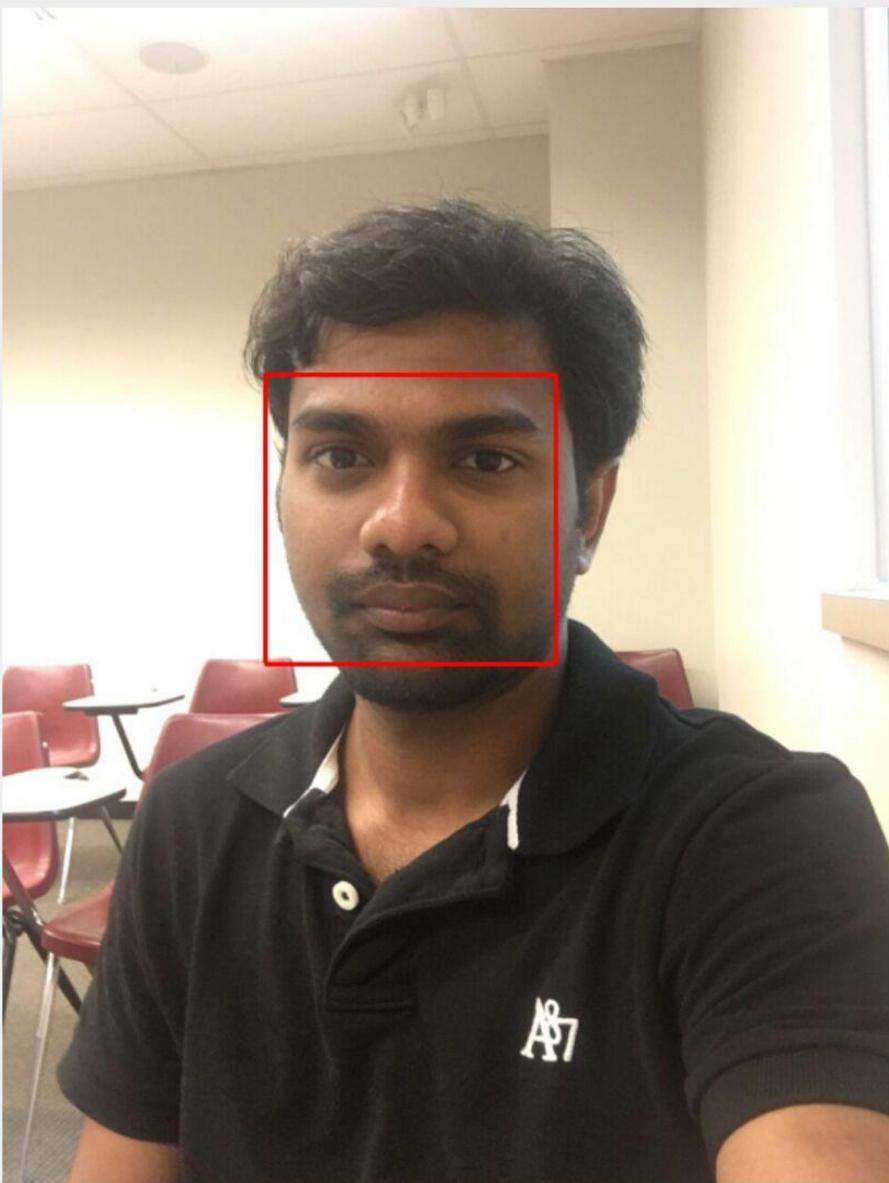
## Facial Recognition





78% 8:34 PM

## Detect Face

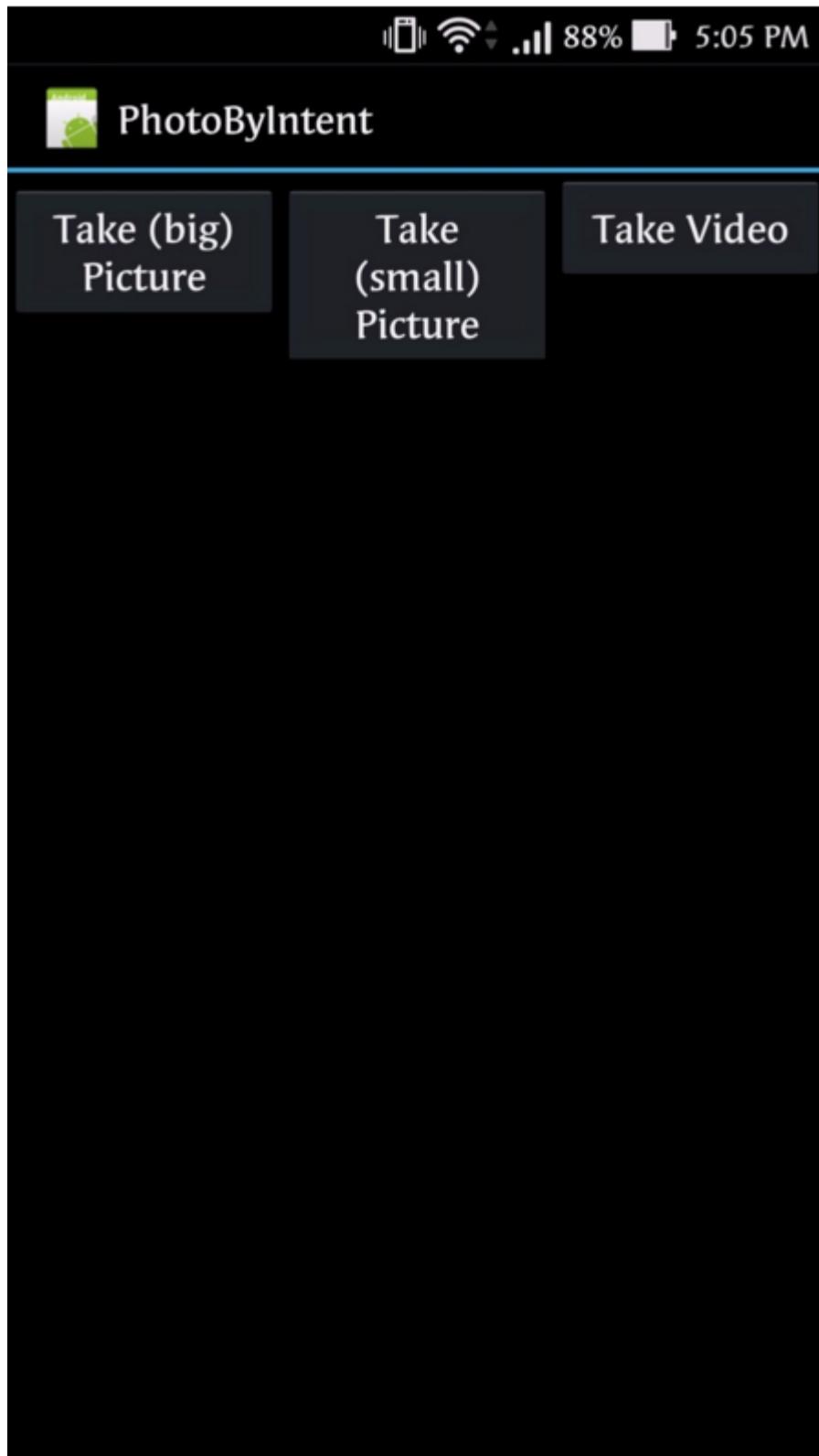


Hello tej@gmail.com

DETECT

GET IMAGE

Option to take an image





78% 8:37 PM



PhotoByIntent

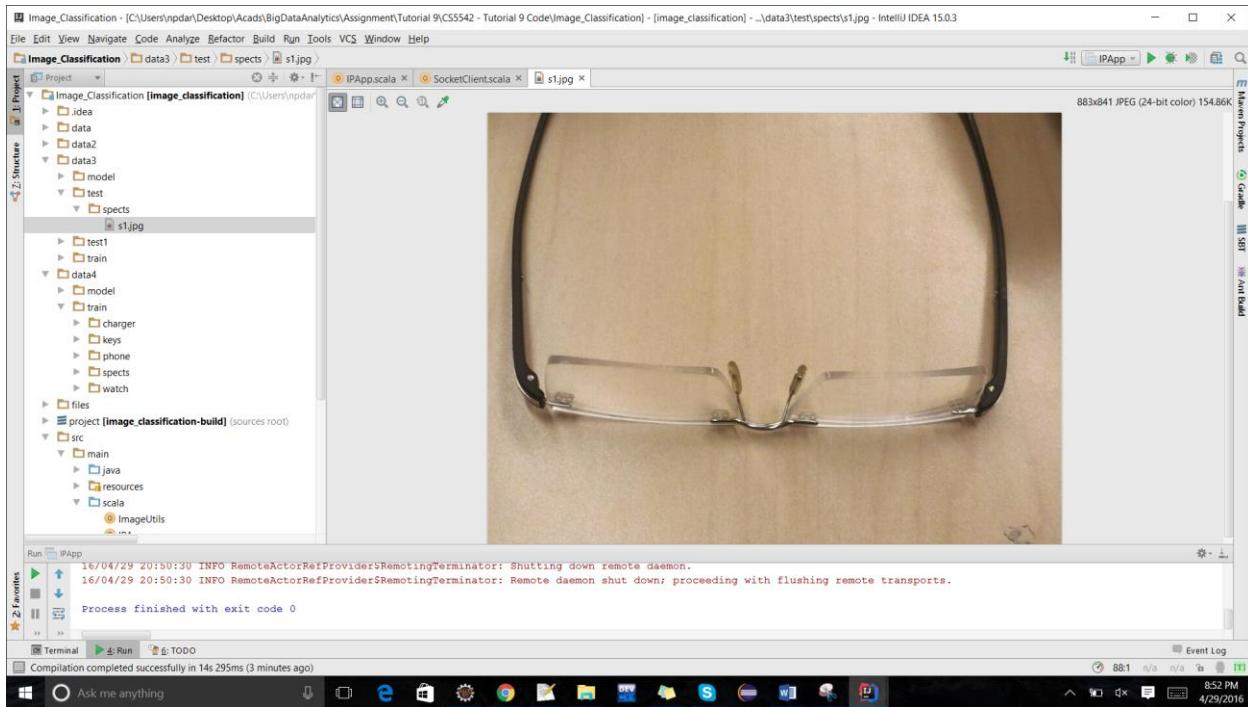
Take (big)  
Picture

Take  
(small)  
Picture

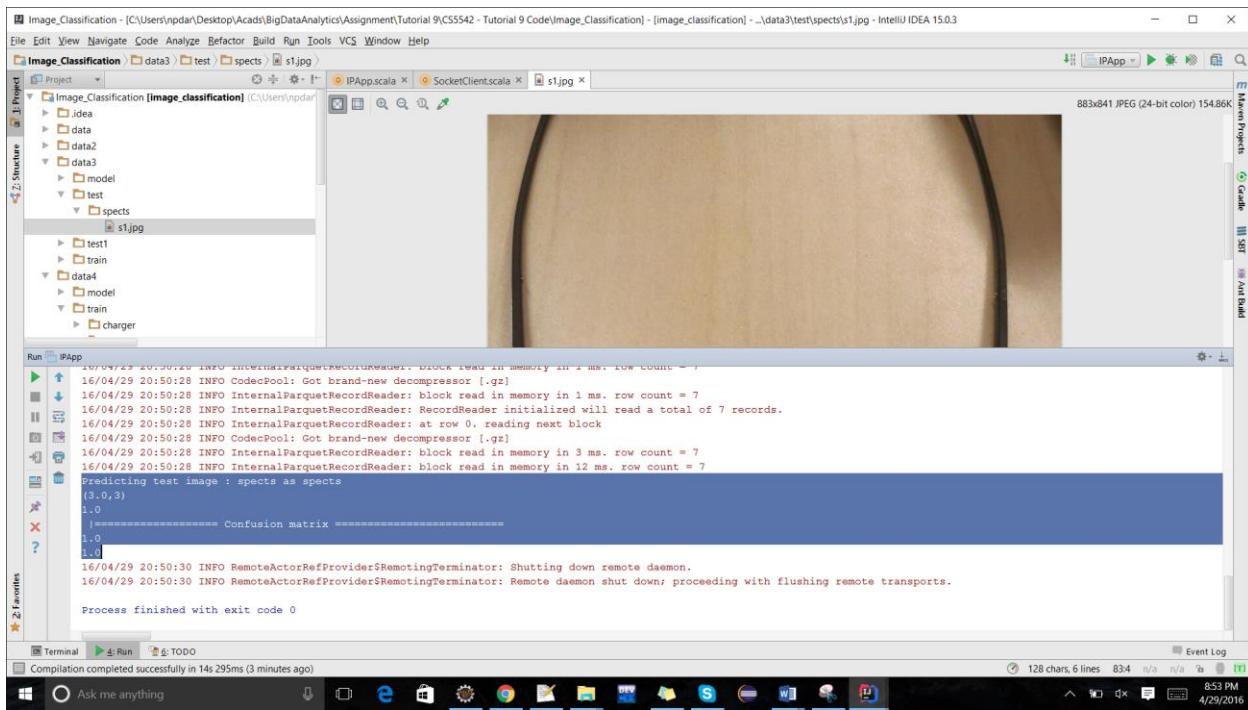
Take Video



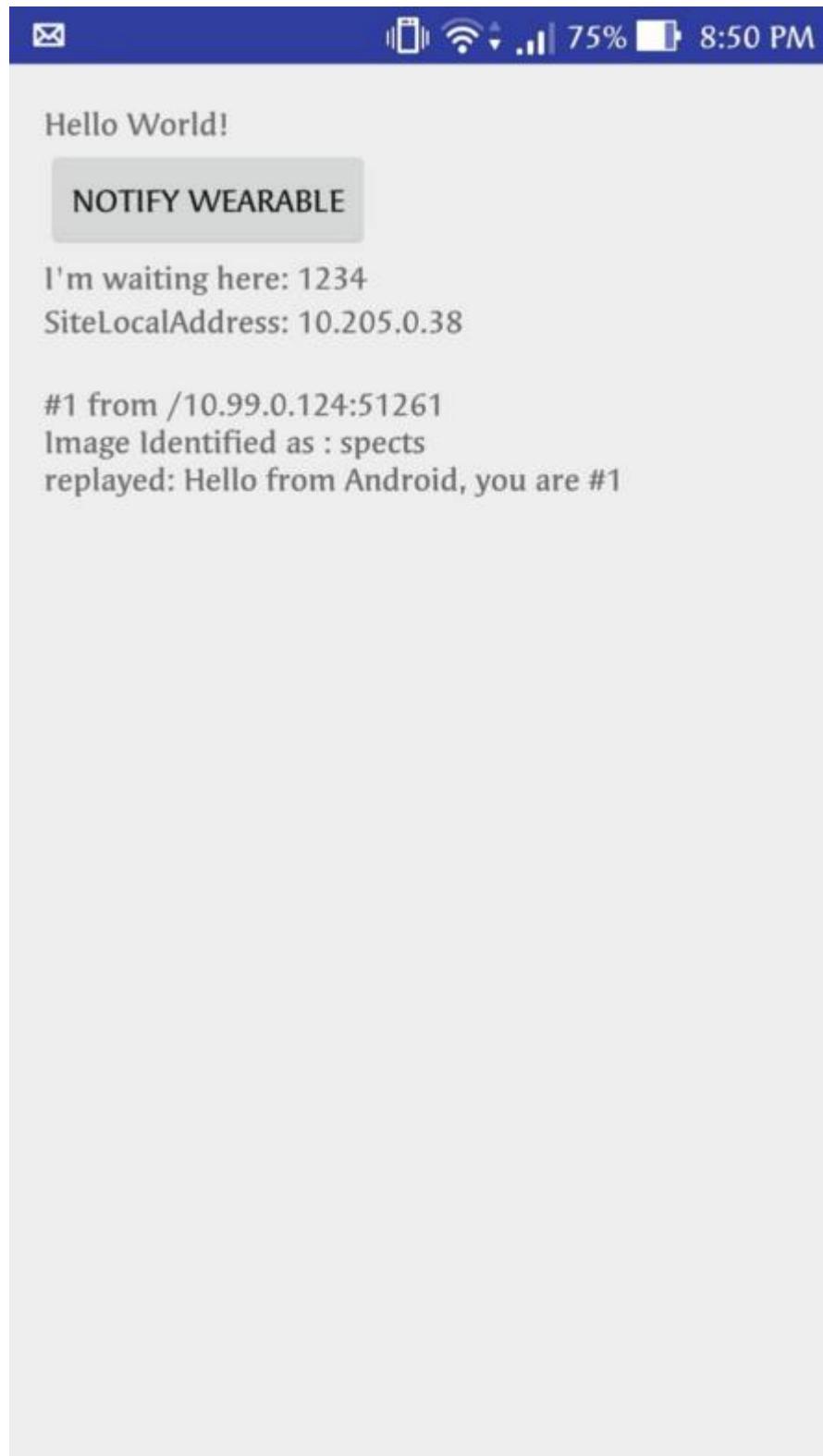
Snapshot shows that the image has been saved into the Spark directory after receiving from Android client.



## Image classification – Object classified as Specs



Notification to smart phone.



Recommendation system – To recommend the location of the object classified.

```
object SparkCollaborativeFiltering {
    def main(args: Array[String]) {
        System.setProperty("hadoop.home.dir", "F:\\winutils")
        val sparkConf = new SparkConf().setMaster("local[*]").setAppName("SparkNaiveBayes").set("spark.driver.memory", "3g").set("spark.executor.memory", "3g")
        val sc = new SparkContext(sparkConf)

        val data = sc.textFile("test.data.txt")
        val ratings = data.map(_.split(',') match { case Array(user, item, rate) =>
            Rating(user.toInt, item.toInt, rate.toDouble)
        })
    }
}

// Build the recommendation model using ALS
```

Run SparkMovieRecommendation

```
RMSE (validation) = 0.8793401396638422 for the model trained with rank = 6, lambda = 0.1, and numIter = 10.  
RMSE (validation) = 0.8724263651153606 for the model trained with rank = 6, lambda = 0.1, and numIter = 20.  
RMSE (validation) = 3.7558695311242833 for the model trained with rank = 6, lambda = 10.0, and numIter = 10.  
RMSE (validation) = 3.7558695311242833 for the model trained with rank = 6, lambda = 10.0, and numIter = 20.  
RMSE (validation) = 0.8784604585989175 for the model trained with rank = 12, lambda = 0.1, and numIter = 10.  
RMSE (validation) = 0.8705540380736105 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.  
RMSE (validation) = 3.7558695311242833 for the model trained with rank = 12, lambda = 10.0, and numIter = 10.  
[Stage 1634:> (0 + 4) / 4]RMSE (validation) = 3.7558695311242833 for the model trained with rank = 12, lambda = 10.0, and numIter = 20.  
[Stage 1687:> (0 + 4) / 4]The best model was trained with rank = 12 and lambda = 0.1, and its RMSE on the validation set is 0.8705540380736105.  
The best model improves the baseline by 22.00%.  
possible locations for finding objects that are recommended for you:  
1: on the table in living room  
2: in the cupboard @ bedroom  
3: underneath bed  
4: in the kitchen  
5: backyard garden
```

16/04/29 17:03:58 INFO RemoteActorRefProvider\$RemotingTerminator: Shutting down remote daemon.  
16/04/29 17:03:58 INFO RemoteActorRefProvider\$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.

Compilation completed successfully in 35s 940ms (today 5:00 PM)

Event Log

88 CRLF: UTF-8: ENG US 901 PM 4/29/2016

Notification to Smart phone about recommendation



## **5.4 Deployment:**

### **Git Hub Link:**

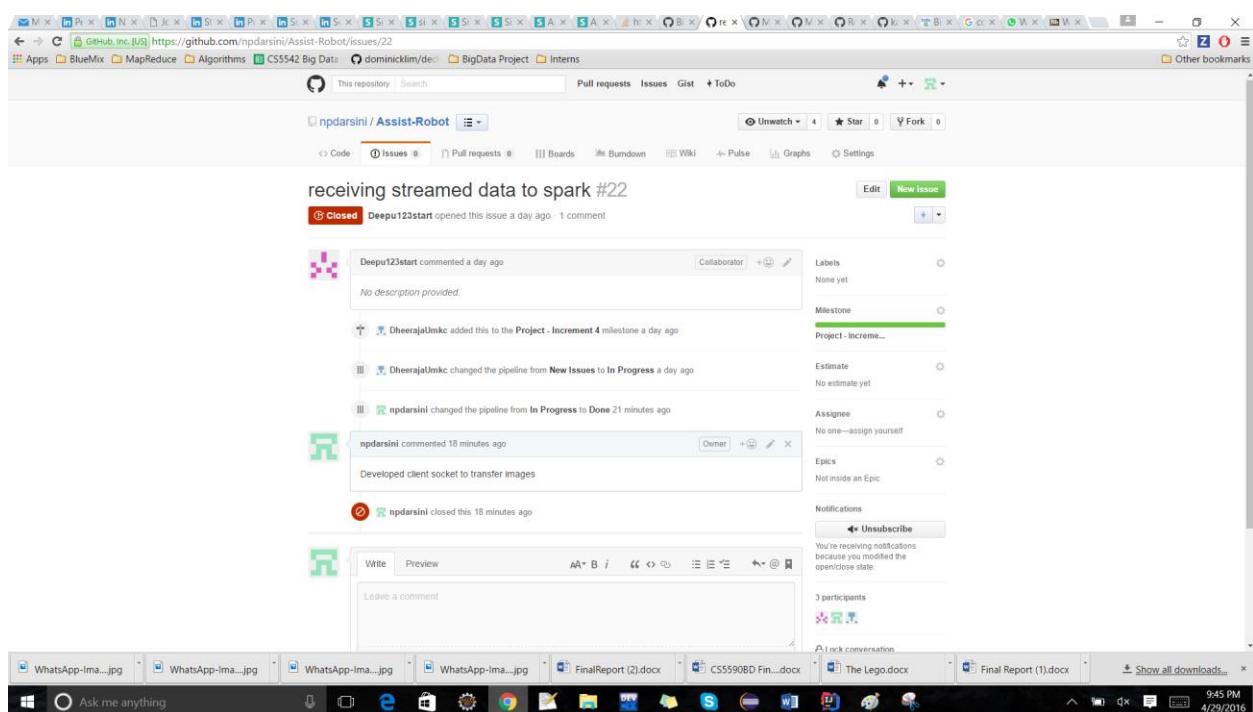
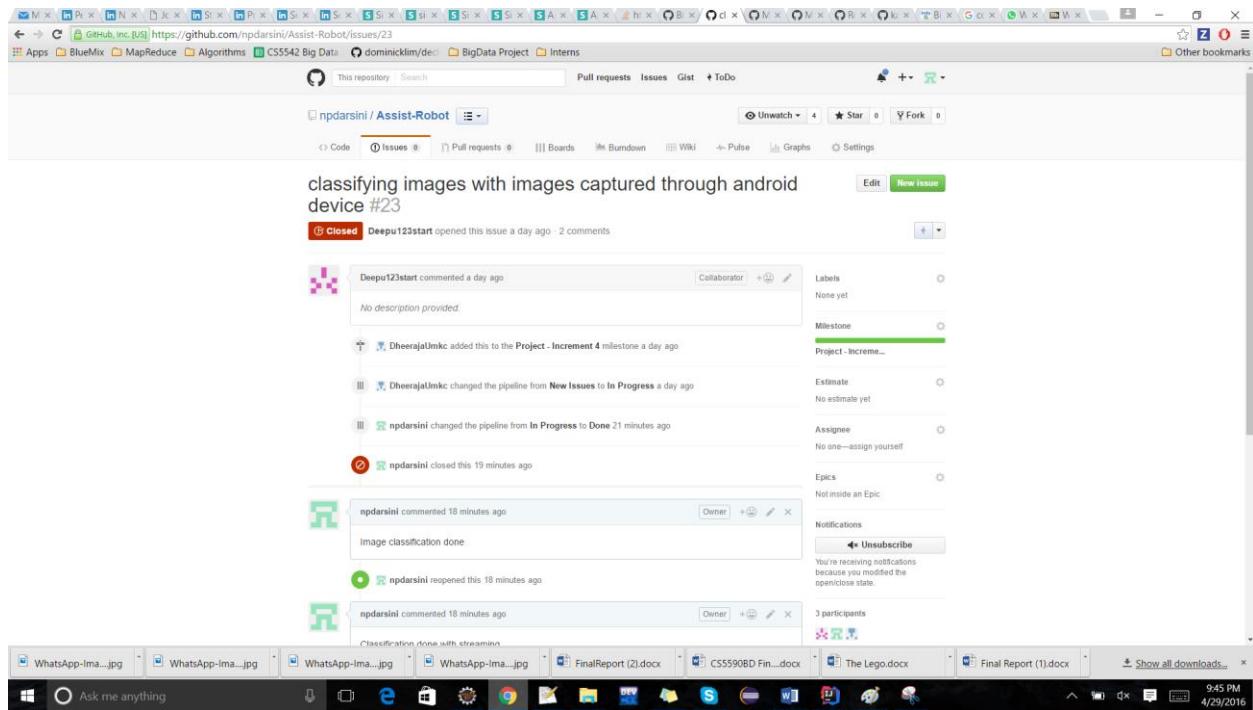
<https://github.com/npdarsini/Assist-Robot>

## **6. Project Management:**

### **6.1 Implementation status report:**

This phase involves the development of Image classification report with the streaming data. Basically the system will classify the object based on the provided image and will notify the user about the object. The image streaming is done from the android device for which we had created an application with the login option. This application can capture the image and transfers the image to the Spark IP address. Upon classifying the image the notification will be sent to the user device. Also a smart recommendation system has been developed which takes the classified image as a base input. Using this input data, it recommends the object location to the user. Assuming the user will be carrying the smart android device every single time, we thought that this could be as an innovative thought as the user don't need to spend a lot of time for finding the lost objects.

Name	Task	Contribution	Time Spent
Priyadarshini N	<b>1. Data Streaming from Android to Spark through Socket communication.</b> <b>2. Image Classification with streaming data from Android using Random Forest Algorithm technique.</b>	100	25
Deepthi P	<b>1. Data Streaming from Android to Spark through Socket communication.</b> <b>2. Recommendation System for finding object location with ALS collaborative filtering technique.</b>	100	25
Tej Kumar Y	<b>1. Android camera module implementation to capture images.</b> <b>2. Face recognition login.</b>	100	20
Dheeraja V	<b>1. Android login page</b> <b>2. Notification on android device.</b>	100	15



GitHub, Inc. [US] https://github.com/npdarsini/Assist-Robot/issues/21

Apps BlueMix MapReduce Algorithms CS5542 Big Data dominicklim/de... BigData Project Interns Other bookmarks

This repository Search Pull requests Issues Gist + ToDo

npdarsini / Assist-Robot Issues Boards Wiki Pulse Graphs Settings

### Streaming data from android #21

**Closed** Deepu123start opened this issue a day ago · 1 comment

Deepu123start commented a day ago  
No description provided

DheerajUmikc changed the pipeline from New Issues to In Progress a day ago

DheerajUmikc added this to the Project - Increment 4 milestone a day ago

npdarsini changed the pipeline from In Progress to Done 21 minutes ago

npdarsini changed the pipeline from Done to To Do 21 minutes ago

npdarsini changed the pipeline from To Do to In Progress 21 minutes ago

npdarsini changed the pipeline from In Progress to Done 21 minutes ago

npdarsini commented 18 minutes ago  
Achieved using client socket programming.

npdarsini closed this 18 minutes ago

Lock conversation

Notifications

Unsubscribe

You're receiving notifications because you modified the specifice state.

3 participants

https://github.com/npdarsini/Assist-Robot/pulse

WhatsApp-Ima...jpg WhatsApp-Ima...jpg WhatsApp-Ima...jpg WhatsApp-Ima...jpg FinalReport (2).docx CS5590BD Fin...docx The Lego.docx Final Report (1).docx Show all downloads...

Ask me anything

GitHub, Inc. [US] https://github.com/npdarsini/Assist-Robot/issues/20

Apps BlueMix MapReduce Algorithms CS5542 Big Data dominicklim/de... BigData Project Interns Other bookmarks

This repository Search Pull requests Issues Gist + ToDo

npdarsini / Assist-Robot Issues Boards Wiki Pulse Graphs Settings

### Linking Modules #20

**Closed** npdarsini opened this issue 23 days ago · 0 comments

npdarsini commented 23 days ago  
No description provided

npdarsini added the enhancement label 23 days ago

npdarsini set the estimate to 13 23 days ago

DheerajUmikc added this to the Project - Increment 4 milestone a day ago

npdarsini closed this 22 minutes ago

Write Preview AA<sup>a</sup> B i  Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Reopen issue Comment

Notifications

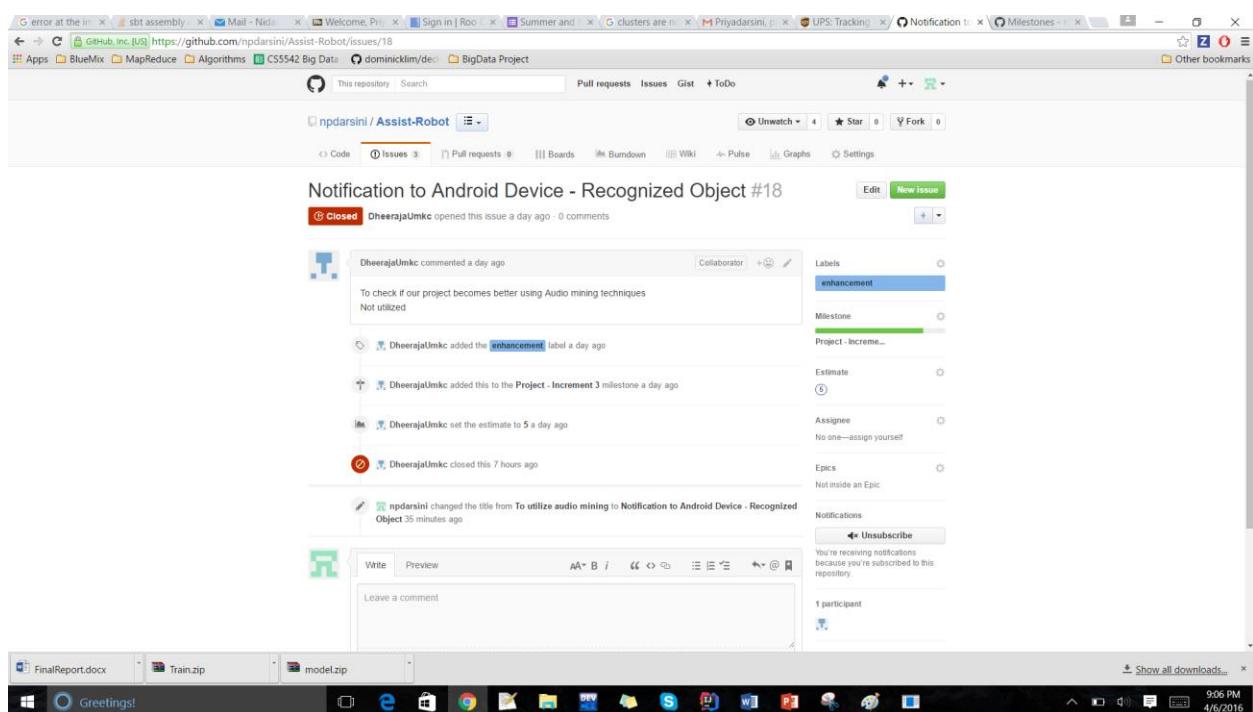
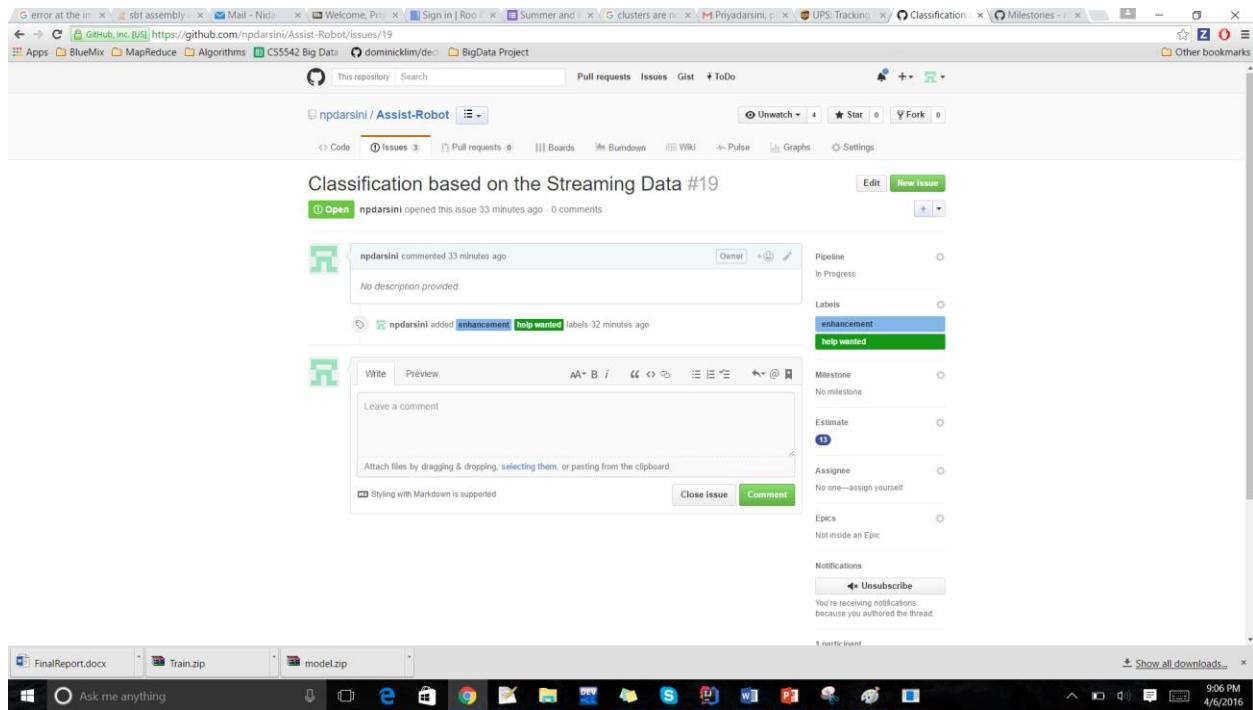
Unsubscribe

You're receiving notifications because you modified the open/close state.

2 participants

WhatsApp-Ima...jpg WhatsApp-Ima...jpg WhatsApp-Ima...jpg WhatsApp-Ima...jpg WhatsApp-Ima...jpg FinalReport (2).docx CS5590BD Fin...docx The Lego.docx Final Report (1).docx Show all downloads...

Ask me anything



[Training data sets - Creation with different objects #17](https://github.com/npdarsini/Assist-Robot/issues/17)

**Closed** DheerajaUmkc opened this issue a day ago · 0 comments

DheerajaUmkc commented a day ago  
Making use of text mining techniques

DheerajaUmkc added the enhancement label a day ago

DheerajaUmkc added this to the Project - Increment 3 milestone a day ago

DheerajaUmkc set the estimate to 5 a day ago

DheerajaUmkc assigned Deepu123start and unassigned Deepu123start a day ago

DheerajaUmkc closed this a day ago

DheerajaUmkc reopened this a day ago

DheerajaUmkc closed this a day ago

npdarsini changed the title from Usage of Text mining to Training data sets - Creation with different objects 37 minutes ago

**Labels**  
**enhancement**

**Milestone**  
Project - Increment 3

**Estimate**  
(5)

**Assignee**  
No one—assign yourself

**Epics**  
Not inside an Epic

**Notifications**  
Unsubscribe

You're receiving notifications because you're subscribed to this repository.

2 participants

FinalReport.docx Train.zip modelZip Show all downloads 9:06 PM 4/6/2016

[Exploring different Classification algorithms - Random Forest, Decision Tree #16](https://github.com/npdarsini/Assist-Robot/issues/16)

**Closed** DheerajaUmkc opened this issue 7 days ago · 0 comments

DheerajaUmkc commented 7 days ago  
Trying to make use of classification algorithms and figure out the best that suits our application

Deepu123start was assigned by DheerajaUmkc 7 days ago

DheerajaUmkc added the enhancement label 7 days ago

DheerajaUmkc added this to the Project - Increment 3 milestone 7 days ago

DheerajaUmkc set the estimate to 8 7 days ago

DheerajaUmkc changed the estimate from 8 to 13 7 days ago

DheerajaUmkc closed this a day ago

npdarsini changed the title from Exploring different supervised learning algorithms to Exploring different Classification algorithms - Random Forest, Decision Tree 38 minutes ago

**Labels**  
**enhancement**

**Milestone**  
Project - Increment 3

**Estimate**  
(13)

**Assignee**  
Deepu123start

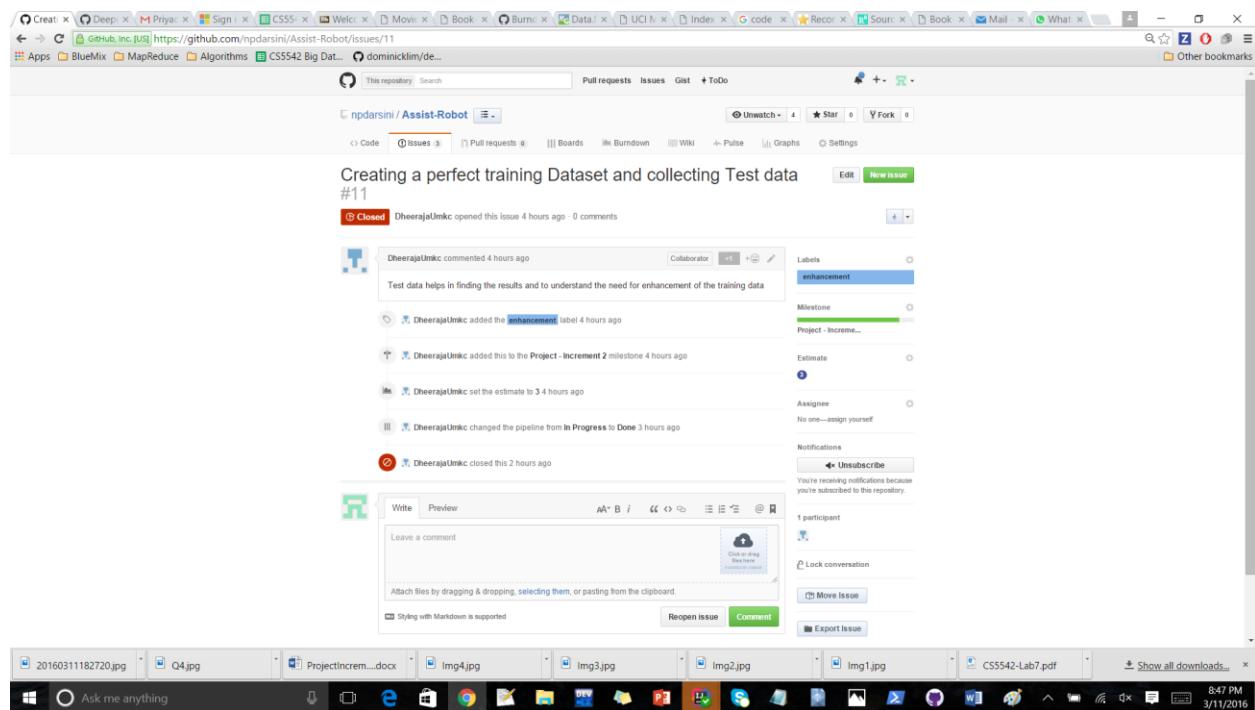
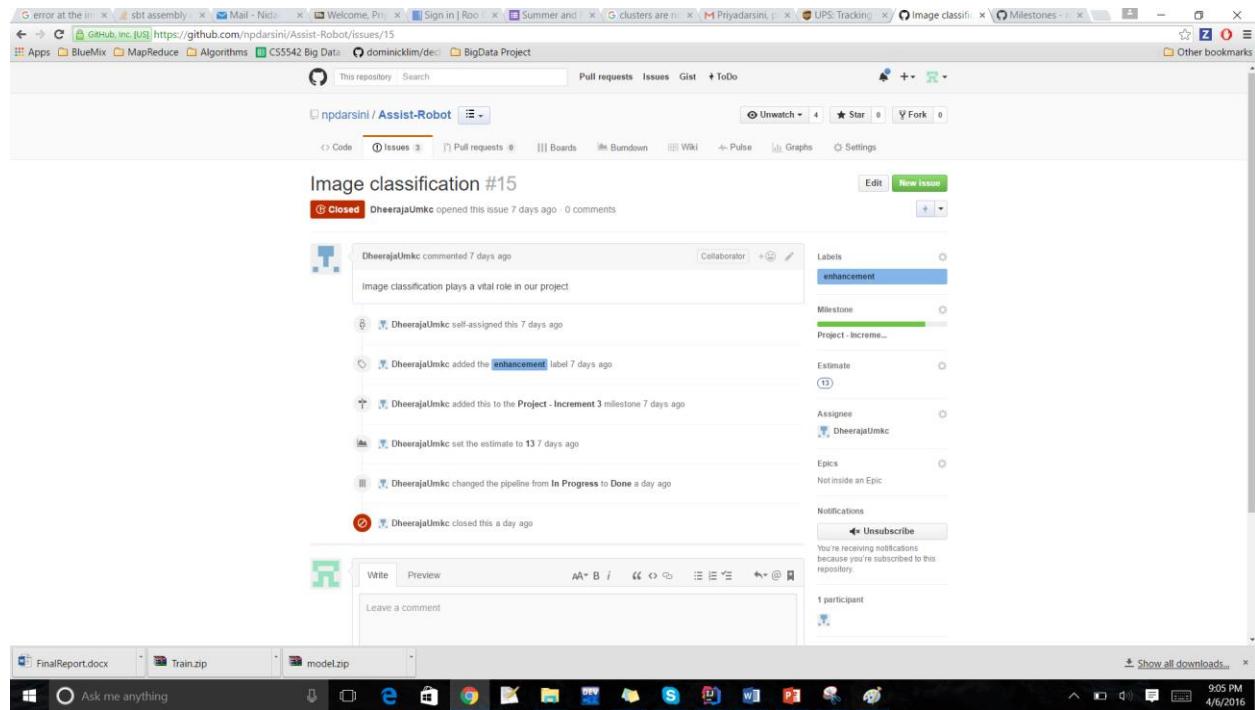
**Epics**  
Not inside an Epic

**Notifications**  
Unsubscribe

You're receiving notifications because you're subscribed to this repository.

2 participants

FinalReport.docx Train.zip modelZip Show all downloads 9:05 PM 4/6/2016



GitHub Issues: Data Collection and analysis #9

**Closed** DheerajUmkc opened this issue 4 days ago · 0 comments

Labels: enhancement

Milestone: Project - Increment 2

Estimate: 3 hours

Assignee: npdarsini

Notifications: Unsubscribe

Participants: 2

Lock conversation

More Issue

Export Issue

Comments:

- DheerajUmkc commented 4 days ago: Collection of data related to project, analyzing it and making it useful to the project.
- DheerajUmkc added this to the Project - Increment 2 milestone 4 days ago.
- DheerajUmkc added the enhancement label 4 days ago.
- DheerajUmkc set the estimate to 3 hours ago.
- DheerajUmkc changed the pipeline from New Issues to In Progress 4 hours ago.
- DheerajUmkc changed the estimate from 3 to 5 hours ago.
- npdarsini was assigned by DheerajUmkc 4 hours ago.
- DheerajUmkc closed this 4 hours ago.
- DheerajUmkc reopened this 4 hours ago.
- DheerajUmkc changed the title from Issue#9 to Data Collection and analysis 4 hours ago.
- DheerajUmkc changed the estimate from 5 to 3 hours ago.
- DheerajUmkc changed the estimate from 3 to 5 hours ago.
- DheerajUmkc changed the estimate from 5 to 3 hours ago.

Downloads: 20160311182720.jpg, Q4.jpg, ProjectIncrem...docx, Img4.jpg, Img3.jpg, Img2.jpg, Img1.jpg, CS5542-Lab7.pdf

Windows Taskbar: Ask me anything, 8:46 PM, 3/11/2016

GitHub Issues: Establishing connection between smart watch and Robome #8

**Closed** DheerajUmkc opened this issue 4 days ago · 0 comments

Labels: enhancement

Milestone: Project - Increment 2

Estimate: 3 hours

Assignee: npdarsini

Notifications: Unsubscribe

Participants: 2

Lock conversation

More Issue

Export Issue

Comments:

- DheerajUmkc commented 4 days ago: We send a notification from robome to smartwatch.
- DheerajUmkc added the enhancement label 4 days ago.
- npdarsini was assigned by DheerajUmkc 4 days ago.
- DheerajUmkc set the estimate to 3 hours ago.
- npdarsini changed the pipeline from To Do to Backlog 3 days ago.
- DheerajUmkc changed the pipeline from Backlog to Done 5 hours ago.
- DheerajUmkc added this to the Project - Increment 2 milestone 4 days ago.
- DheerajUmkc changed the title from Issue#8 to Establishing connection between smart watch and Robome 4 hours ago.
- DheerajUmkc closed this 4 hours ago.
- DheerajUmkc reopened this 4 hours ago.
- DheerajUmkc closed this 3 hours ago.

Downloads: 20160311182720.jpg, Q4.jpg, ProjectIncrem...docx, Img4.jpg, Img3.jpg, Img2.jpg, Img1.jpg, CS5542-Lab7.pdf

Windows Taskbar: Ask me anything, 8:45 PM, 3/11/2016

GitHub Issues - Installation of Spark #7

**Closed** DheerajUmkc opened this issue 4 days ago · 0 comments

DheerajUmkc commented 4 days ago Since usage of spark makes the project more flexible because of availability of RDD's

DheerajUmkc added the enhancement label 4 days ago

dunil210 was assigned by DheerajUmkc 4 days ago

DheerajUmkc added this to the Project - Increment 2 milestone 4 days ago

DheerajUmkc set the estimate to 3-4 days ago

npdarsini changed the pipeline from To Do to Backlog 3 days ago

DheerajUmkc changed the title from Issue3 to Installation of Spark 4 hours ago

dunil210 was unassigned by DheerajUmkc 4 hours ago

DheerajUmkc changed the pipeline from Backlog to Done 4 hours ago

Deepur123start was assigned by DheerajUmkc 4 hours ago

DheerajUmkc closed this 4 hours ago

DheerajUmkc reopened this 4 hours ago

DheerajUmkc closed this 4 hours ago

Ask me anything

GitHub Issues - Features of RoboMe #6

**Closed** DheerajUmkc opened this issue 4 days ago · 0 comments

DheerajUmkc commented 4 days ago Going through the features of Robome and what all can be derived using its basic features

DheerajUmkc added enhancement question labels 4 days ago

Deepur123start was assigned by DheerajUmkc 4 days ago

DheerajUmkc added this to the Project - Increment 2 milestone 4 days ago

DheerajUmkc set the estimate to 5-4 days ago

npdarsini changed the pipeline from In Progress to To Do 3 days ago

npdarsini changed the pipeline from To Do to In Progress 3 days ago

DheerajUmkc changed the pipeline from In Progress to Done 5 hours ago

DheerajUmkc removed the enhancement label 4 hours ago

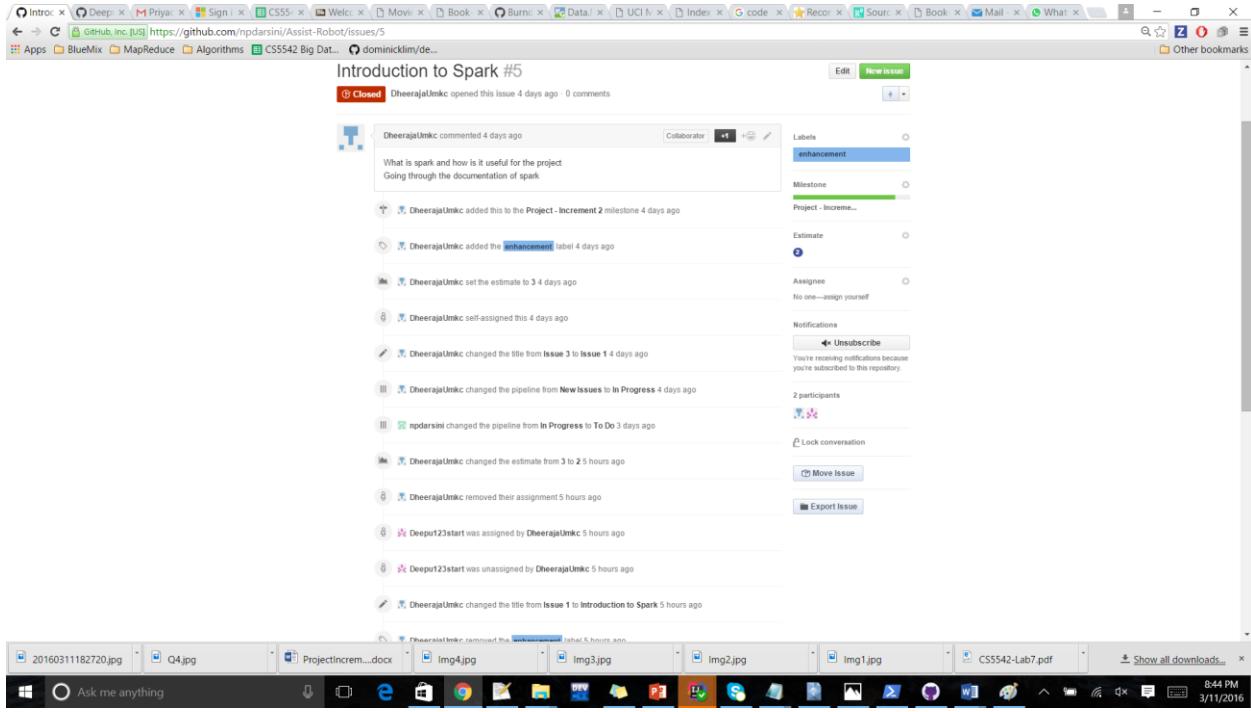
Deepur123start was unassigned by DheerajUmkc 4 hours ago

DheerajUmkc changed the estimate from 5 to 4 hours ago

npdarsini was assigned by DheerajUmkc 4 hours ago

npdarsini was unassigned by DheerajUmkc 4 hours ago

Ask me anything



## 7. Future Scope:

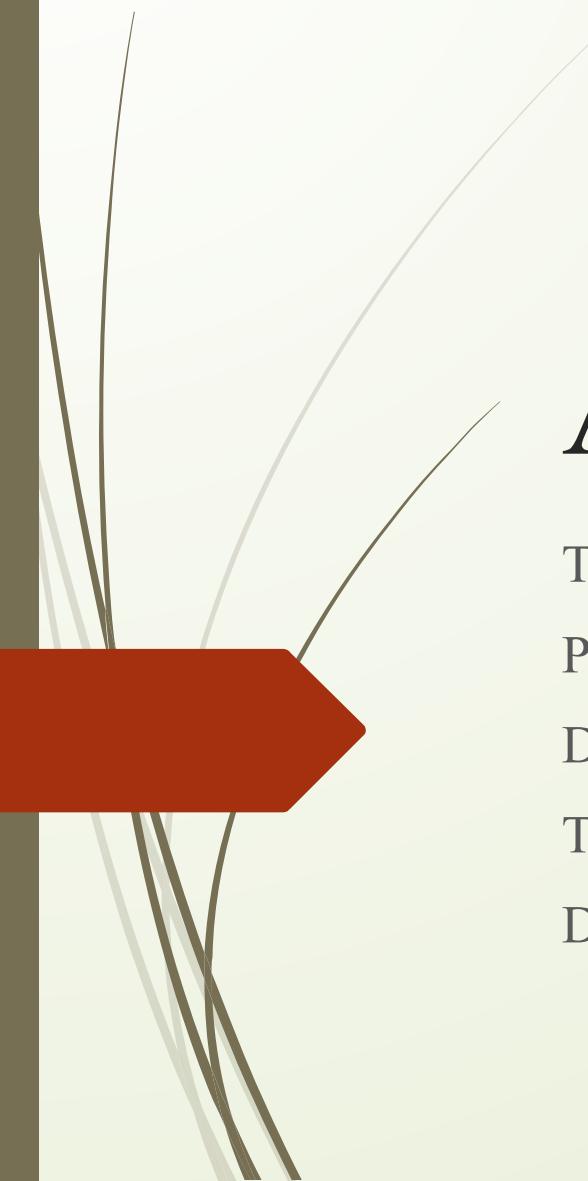
To implement a Robo based system that can move around and track the objects. Upon tracking, the user will be notified about the object location. To explore SparkSQL to store the captured images so that we will be able to deal with the massive data. To explore much about the Machine learning algorithms so as to achieve better accuracy in Image Classification. To embed Natural Language Processing techniques into the current working project.

## 8. Bibliography:

Lab Tutorials and the material provided by Dr. Lee.

SPARK - <https://spark.apache.org/docs/latest/programming-guide.html#external-datasets>

Spark Recommendation - <http://spark.apache.org/docs/latest/mllib-collaborative-filtering.html>



# Assist Robot

Team 8

Priyadarsini Nidadavolu(17)

Deepti Priya Darshini Penmetsa(22)

TejKumar Yentrapragada(33)

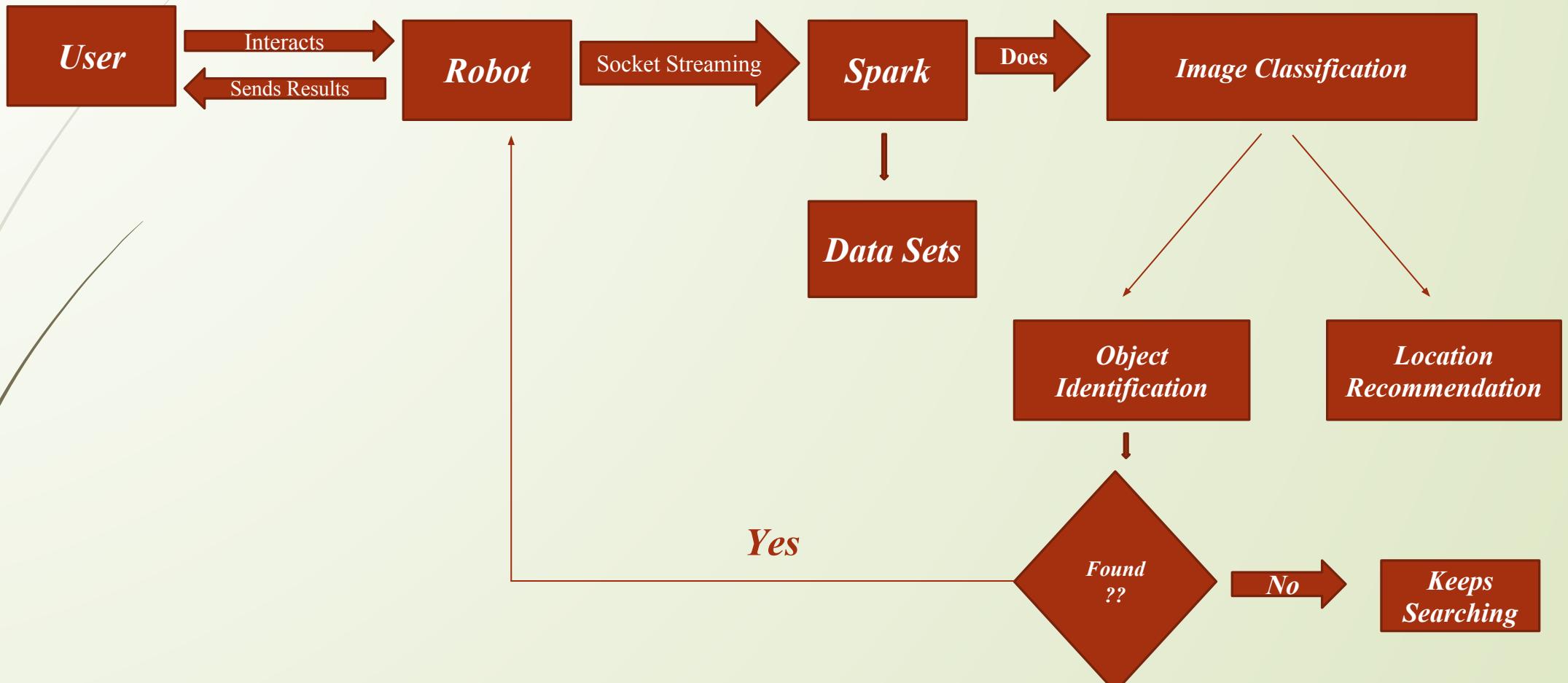
Dheeraja vallabhaneni(28)



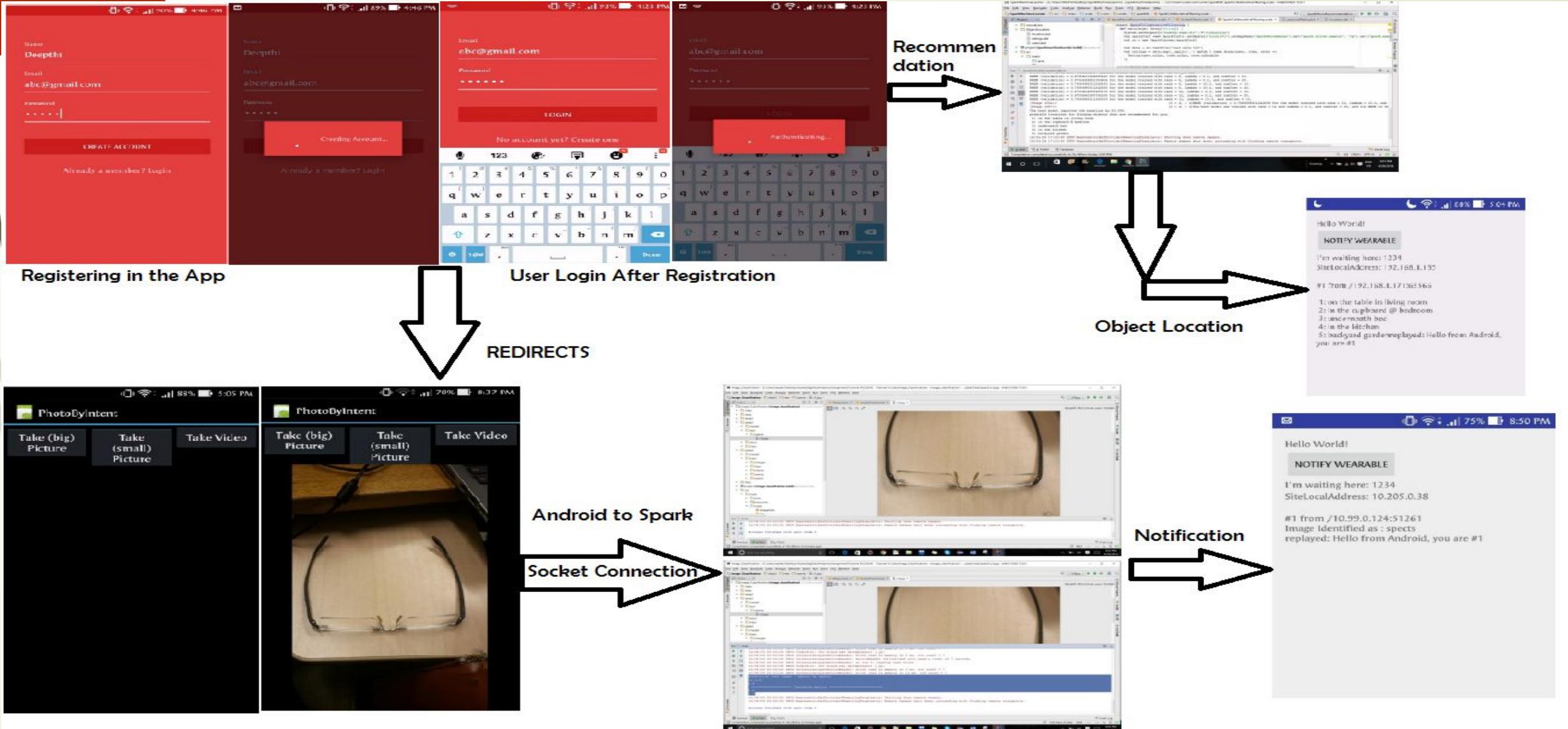
# Features and Data Sets used

- To find the misplaced things like keys, watch, phone etc.,
- To build a recommendation system which should be able to recommend the Possible Locations of the articles.
- To make a robot to serve multiple people within a house. (Face Recognition/User Login)
- To send notifications to either user phone or smart watch about the object identified and also the possible location of the article that was misplaced.
- The first dataset could be a set with list of images of the articles with which we need to train our robot.
- The dataset which specifies the location of the articles that is preferred by the user will be our second dataset.
- The dataset which specifies the list of users, whose commands needs to be taken.

# Workflow



# Flow Screenshots





# Advantages

- With this Application the user could be able to locate his lost objects in his house
- The user need not stress himself in memorizing the location of the most important objects that he uses frequently
- Time saving



*Thank You*

# Assist Robot

## Project Report

### Project Team – 8

#### Team Members

PriyadarsiniNidadavolu(17)

DeepthiPriyadarshiniPenmetsa(22)

DheerajaVallabhaneni(28)

Tej Kumar Yentrapragada(33)

## Table of Contents

Section	Description
1	Introduction
	1.1 Motivation and Objectives
2	Solutions
	2.1 Architecture
	2.2 Algorithms
	2.3 Data Sets
	2.4 Evaluation
	2.5 Accuracy and Performance
3	References
4	Future Work
5	GitHub Link
6	YouTube Link
7	Presentation Link

## **1. Introduction:**

The main goal of the project is to help people in finding their misplaced objects. Basically humans have a tendency to forget their belongings somewhere in their house and search for it for hours together. For example, if I have an important business meeting to attend, but I don't remember where I placed my car keys, then I will be in huge loss. So to prevail in these circumstances here comes our Friendly Robot- My Friend which could assist me in keeping track of my personal things. So what this robot will do is that it will have entire map (laser scan) of the building and objects in the building in its memory. So we will feed the robot with the objects that are highly important to us, like car keys, some files, phone and laptop. This robot will keep its eye on these objects and notify their location to its master upon request. Additionally, our Robot who will be an eFriend who will help us to choose the furniture to our home. Also our robot will suggest us the top rated books.

### **1.1 Project Goal and Objectives:**

The primary goals of our project is described below:

- To implement a module which has an interaction with robot. E.g. You can ask few questions to the robot and the Robot will be responding to you back. You can ask the robot about your misplaced phone. So that it will answer you after it had found the phone.
- To make the robot learn about the personal items like chargers, phones, watch, keys etc.
- To design a robot which can find the learned objects that are misplaced in a building.
- To build a recommendation system which will be able to recommend the list of books which are rated high and are related to our interests.

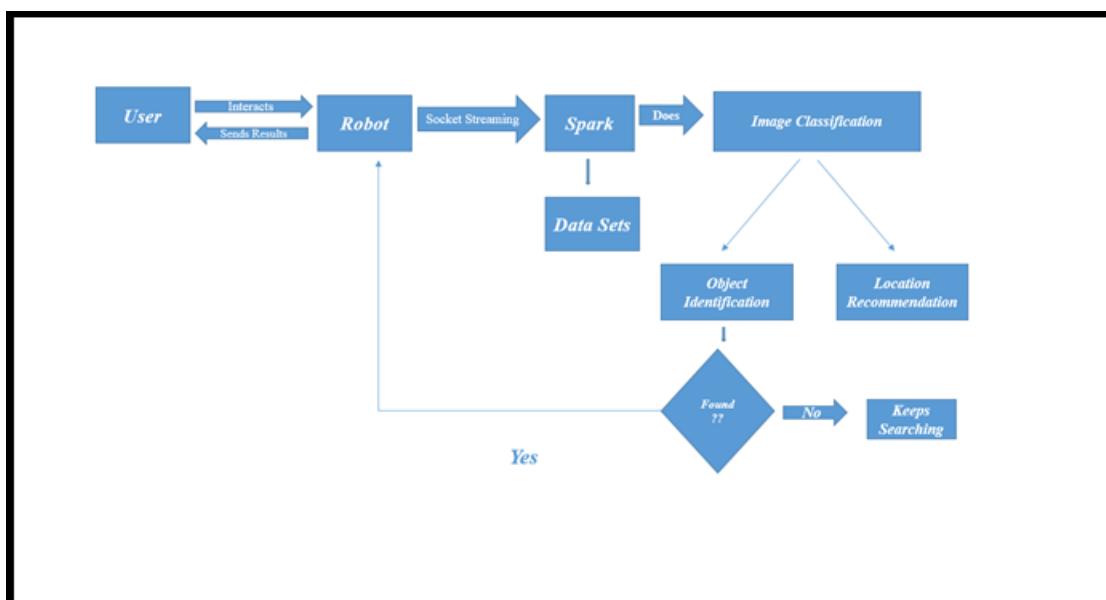
- To build a recommendation system which will suggest us about the latest furniture details, their quality and from which brand/ shop we could purchase from. This feature enables us to decorate our houses with rich interior designing.
- To send a notification to your smart watch when it finds the lost object.
- To remind the user about his day to day events that were previously taught.
- To make the robot act as an assistant in getting things specified by the user. (Mr Robot – Get me my phone).
- To make a single robot act as assistant to all the people living in same house. It recognizes the user first and then assists that particular user in finding the belongings.

## 2. Solutions

In order to achieve the goals of this project we might have to use the techniques of machine learning, a few of natural language processing algorithms. The machine will be fed with the testing and training data sets with which it must build its own capability of decision making.

### 2.1 Architecture

User will be interacting with the Robot/Android Device communicating his/ her requirements and then the requirement of the user will be streamed to spark where decision making takes place and the results will be sent back to the user's device.



The above work flow is justified in the project by the way the user interaction with the Android device happens through the login page where the user need to feed the images manually for which we had implemented a camera module. The image feed will be streamed to Spark framework where the machine learning algorithms like Random Forest and Alternate Least Squares(ALS) has been implemented. Upon building a decision system, there will be a classification that happens on the type of object. This type will be analyzed using the ALS algorithm and a recommendation about the location of the object will be notified to the user via smart phone/ watch.

## 2.2 Algorithms

The two prominent machine learning algorithms has been used as part of this project.

- Random Forest (Image Classification)

The objective here is to classify the image based on the object category that it contains. Image classification using random forest has proven to give better results compared to decision trees. Decision tree is encountered with over fitting problem if the dataset is huge. So random forests are trained on various parts of the same training dataset by averaging multiple decision trees. This solves the problem of over fitting that comes with using individual decision trees.

- Alternate Least Squares Collaborative filtering (Recommendation System)

ALS Collaborative filtering identifies patterns of users to make targeted recommendations. This technique is mainly used in recommendation systems to fill the missing entries in user-item association matrix.

## 2.3 Data Sets

There are three data sets that we had used as part of this project. They are:

- The first dataset with a list of images of the articles with which we need to train our robot.(IMAGENET database).
- Secondly, the dataset which specifies the location of the articles that is preferred by the user.
- Third dataset which specifies the list of users, whose commands needs to be taken.

## **2.4 Evaluation:**

For classifying images we used Random forests to predict the images, where the training dataset is categorized to 6 categories with each category consisting of about 60 images. The testing dataset is with images that are streamed from android device. With this setup the accuracy noticed is around 75 percent. As the dataset in the training category increases by 20 percent of the current dataset the accuracy decreases gradually.

Similarly in recommendation system, the dataset consists of users information, user ratings and location data where objects could be found. So with ALS collaborative filtering the object is mapped to the possible locations. So the accuracy with technique is around 60 percent.

## **2.5 Accuracy and Performance:**

Using Random forests for image classification the accuracy was around 75 percent. With ALS collaborative filtering in recommendation system the accuracy found is 60 percent.

## **3 References**

Lab Tutorials and the material provided by Dr. Lee.

SPARK - <https://spark.apache.org/docs/latest/programming-guide.html#external-datasets>

Spark Recommendation - <http://spark.apache.org/docs/latest/mllib-collaborative-filtering.html>

## **4 Future Work**

To implement a Robo based system that can move around and track the objects. Upon tracking, the user will be notified about the object location. To explore SparkSQL to store the captured images so that we will be able to deal with the massive data. To explore much about the Machine learning algorithms so as to achieve better accuracy in Image Classification. To embed Natural Language Processing techniques into the current working project.

## **5. Project Management**

This project involves the development of Android application with the login page option for users registration and login. A camera module need to be provided on the android application for the user to capture the object images. And the streaming module need to be handled in order to

send these captured images to Spark framework. An Image classification takes place over here on the streaming data. Basically the system will classify the object based on the provided image and will notify the user about the object. Also a smart recommendation system has been developed which takes the classified image as a base input. Using this input data, it recommends the object location to the user. Assuming the user will be carrying the smart android device every single time, we thought that this could be as an innovative thought as the user don't need to spend a lot of time for finding the lost objects.

Name	Task	Contributio n	Time Spent	Grade
Priyadarsini N	<p><b>1. Data Streaming from Android to Spark through Socket communication.</b></p> <p><b>2. Image Classification with streaming data from Android using Random Forest Algorithm technique.</b></p>	100	25	100

Deepthi P	<b>1. Data Streaming from Android to Spark through Socket communication.</b>  <b>2. Recommendation System for finding object location with ALS collaborative filtering technique.</b>	100	25	100
Tej Kumar Y	<b>1. Android camera module implementation to capture images.</b>  <b>2. Face recognition login.</b>	100	20	100
Dheeraja V	<b>1. Android login page</b>  <b>2. Notification on android device.</b>	100	15	100

### Final Project Evaluation

Initially we have a plan of implementing a module of ROBOC, where the robot will be roaming around the house and takes the images through its censored IOS device. As there is no Robot functionality involved as part of the resource management, we came up with an idea of implementing an android module where the user needs to capture the images and these images will be streamed to Spark. This adjustment of requirement even in the last stages of the project happens only in Agile Methodology. This Agile methodology basically provides flexibility to the user's requirements and timelines. The usage of the tools like ZenHub, versionOne etc keeps track of the effort of an individual. In the current world, I strongly suggest the developers to provide an option to the client for restructuring the requirements which could be done only through Agile methodology.

There is an equal contribution from each of my team member towards the project and each one is discipline enough while carrying out their tasks.

The Agile methodology could be handled effectively by practicing Scrum system which happens in the real time industries. I suggest to practice these scrum policy (once in a week or so).

## **5. GitHub Link**

<https://github.com/SCE-UMKC/BigData-Spring-2016-AssistRobot>

## **6. YouTube Link**

<https://www.youtube.com/watch?v=HjZGMTrZMIk>

## **7. Presentation Link**

[https://drive.google.com/file/d/0B6iN7IIkj7l\\_bVJxZUVjZXlwTDJmZW1CLVRRbGJ5ek9kcndr/view?usp=sharing](https://drive.google.com/file/d/0B6iN7IIkj7l_bVJxZUVjZXlwTDJmZW1CLVRRbGJ5ek9kcndr/view?usp=sharing)