Smart Hear - Intelligent Hearing for Android

Final Project Package

Submitted on **09th May 2016**

Group 1:

1. Ragunandan Rao Malangully (13) **2. Ravi Kiran Yadavalli (31)**

GitHub Project URL: https://github.com/SCE-UMKC/BigData-Spring-2016-SmartHear

YouTube URL: https://www.youtube.com/watch?v=bhRDAIpw78k

Presentation URL: https://www.dropbox.com/s/zzxdpst8wwqykn1/UMKC_CS5542.pptx?dl=0

Project Proposal

<u>Title of the Project:</u> Intelligent Hearing for Android

Ragunandan Rao Malangully (13) **Ravi Kiran Yadavalli (31)**

<u>Motivation:</u>

Hearing disability is one of the major concerns challenging the mankind. Medical technology has come a long way to cater the needs of the facing such challenges. These advancements have a high cost associated with it. Hearing aid still remains a dream for a common man. A solution to this is a hearing aid that is smart and at the same time affordable. The application performs context aware analysis to perform noise cancellation and enhance the hearing experience of the user. An additional feature of the application is to notify the user important sounds such as alarm, phone ring etc. through a notification sent to his mobile device.

<u>Goal and Objectives:</u>

The objective of this project is to accumulate and analyze the important features of an audio signal. Then process the features through selected machine learning algorithms. The processing framework would be big data framework which would provide faster processing and reduce the application processing time. Feature can be defined in multiple ways. A *feature* is a unit of functionality of a software system that satisfies a requirement, represents a design decision, and provides a potential configuration option.

From a user perspective and a developer perspective it is important to know about the features of the application which will help them to know the use of the applications, basic requirement to use the application semantics and structure of the code respectively. Identifying the required important features would be the most important aspect of the process. Once this is acquired applying the models on these features would yield us the desired result. The project involves feature extraction and weighing the features to pick the appropriate feature vectors. This feature extraction can be done at the client. Next comes applying a learning algorithm that can predict a model for the features provided as input. This would be done using the SPARK MLib package.

<u>Significance:</u>

Today most of us have an access to smart phones. In the USA alone Android smart phones constitutes 53 percent of the mobile market. Choosing Android as the platform for developing the application was ideal to reach bulk of the smart phone users.

System Features:

- An Android application that serves as the smart hearing aid and also provides a user friendly model for the feature extraction.
- A machine learning framework that can provide a model based on the features received from the client end.
- Analysis of audio files to observe their characteristics.
- Collection of features present in an audio file and weighing the features to select the vital features dynamically.
- To send text notifications on various events based on the pattern of previous audios.
- To be able to detect and send alarms on to the device in potential hazardous situations.
- Integration of the client and the processing framework for a complete application that can perform as a smart hearing device.

Related Work:

1. RDF → a formal representation for unstructured data such as text data and sensor data.

2. Machine learning algorithms → Study about classification and regression, clustering, dimensionality reduction, feature extraction.

3. SPARQL → it is a semantic query language for databases used to retrieve and manipulate data in RDF format.

4. Apache Spark → an engine for big data processing, which performs in-memory computations.

Requirements Specifications:

1. An android mobile device with minimum version of Android 5.0 (Lollipop).
2. Android wearable devices with minimum version of Android 5.0 (Lollipop) .

Bibliography:

I. Megha Sharma,"Automatic Architecture Recovery Model", Thesis Fall 2014 in University of Missuori Kansas City

II. Sven Apel, Christian Kastner :An Overview of Feature-Oriented Software Development, Published in Journal of Object Technology, Vol. 8, No. 5, July–August 2009

III. Samad Paydar, Mohsen Kahani : A Semantic Web Based Approach for Design Pattern Detection from Source Code. In : 2012 2nd International conference on Computer and Knowledge Engineering (ICCKE), October 18-19, 2012

IV. Hendrik Speidel, Philipp Wendler, Alexander von Rhein, Dirk Beyer : Detection of feature interactions using feature-aware verification. Proceeding ASE '11 Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering

V. Yashwanth Rao Dannamaneni, Semantic Code Search and Analysis for Open Source Projects, MS Thesis, University of Missouri – Kansas City, May, 2014

Smart Hear - Intelligent Hearing for Android

Project Plan and First Iteration Report

Submitted on **02 February 2016**

Group 1:

1. Ragunandan Rao Malangully (13) **2. Ravi Kiran Yadavalli (31)**

I. Introduction

This document intends to provide an overall description of the project named “Smart Hear” in detail. The project schedule and the plan of action is also discussed. The proposal document submitted would give an insight about what the project is about. The main outcome of the first increment was the high and low level design of the application. As of current state we have not deviated from our initial proposal that we submitted earlier.

II. Project goal and objectives

Overall goal

The overall goal is to provide a hearing aid through the use of the smart phone that is accessible to every person today. The smart phone can be utilized in a way that it can act similar to the hearing dog that is what motivated us to take up this idea and try to implement using the smart phone.

This application can be used to provide benefits to the special abled people who face challenges in listening to sound. All the features come at the cost of installing an application that is freely provided to the user. No costs attached.

Objectives

- To provide a smart hearing aid to the user.
- To implement features to provide the user flexibility in varying certain key features of sound like frequency, tempo etc.
- To provide notifications to the user when there are important events like a door bell ring or an alarm that goes off.
- To develop a user friendly application.
- To ensure that we have a light weight application on the client end.
- To make sure that the context is recognized and the user settings are changed accordingly.
- Implement knowledge discovery to provide a summary of a topic that the user is listening or recording.
- Provide user customizable settings that the user can adjust to suit his or her needs.
- Ensure that the application can deliver its functionality at minimal cost of operation.
- Provide a user guide to understand and use the application.

III. Project background and related work

The app is not a new invention or a brand new concept. But the implementation aspect of it is what makes it unique from the others. The following are the projects that are similar to what our idea is.

- Resound LiNX: It is a smart hearing aid kit that can integrate with Apple Iphone and other android devices. The device does noise cancellation and also amplifies the sound that the user is listening to. The device can also integrate with smart watches and provide very handy functionality. But the downfall is that it is an additional device that has to be purchased and is does not have notification features to the user.
- Siemens Hearing Aids: These were introduced at the International CES 2015. They are pretty close to what an interactive hearing aid can deliver and seamlessly work hand in hand with smart devices. They require a specific app to connect the device with the hand held devices and only then can they provide the functionality. On the flipside the product is priced high and is not for everyone to afford it.

Significance

The major significance of the application lies in the fact that the all the useful services are provided under one system. There is no need of another device for the hearing enhancements. The user can access all the features through his smart phone. Even form the implementation aspect the usage of big data framework would ensure that the application can process the data at high performance rates. This would also reduce the cost of the creating and operating the application. User customizable settings is one of the major significance of the project.

IV. Proposed system

Requirements specifications

- The application will provide users facility to login with their Google plus account.
- User should have the flexibility to login with his Facebook account.
- The system will display the user his information along with his profile picture.

- An Android application that serves as the smart hearing aid and also provides a user friendly model for the feature extraction.
- A machine learning framework that can provide a model based on the features received from the client end.
- Analysis of audio files to observe their characteristics.
- Collection of features present in an audio file and weighing the features to select the vital features dynamically.
- To send text notifications on various events based on the pattern of previous audios.
- To be able to detect and send alarms on to the device in potential hazardous situations.

- Integration of the client and the processing framework for a complete application that can perform as a smart hearing device.

Workflow analysis

The first and the most fundamental workflow of the application is that of the designing the UI for the system. This involves selecting the proper HTML elements and arranging the elements in an appropriate manner. It includes making use of the CSS attributes to style the UI of the application. Each page of the application is to be custom designed for suiting the needs of the service or feature being provided over that page. This would also include capturing the sound from the device and storing it in the device storage.

Second task would be do perform basic filtering and other alterations to the sound using libraries such as high pass, band pass etc. The frequency, tempo, pitch and other features of the sound can be altered and extracted individually.

The third workflow deals with the performing model discovery or applying the best fit model over the features extracted. This would involve applying machine learning algorithms like Decision tree etc. to classify the source of the sound.

Technological and architectural requirements

On the technological aspect the project is pretty much dependent on the native Android libraries for implementing the client functionality. This would also include some external Java or Android based libraries or plugins for managing the operations on the sound recorder. The primary development language for the client would Java and XML for the UI part.

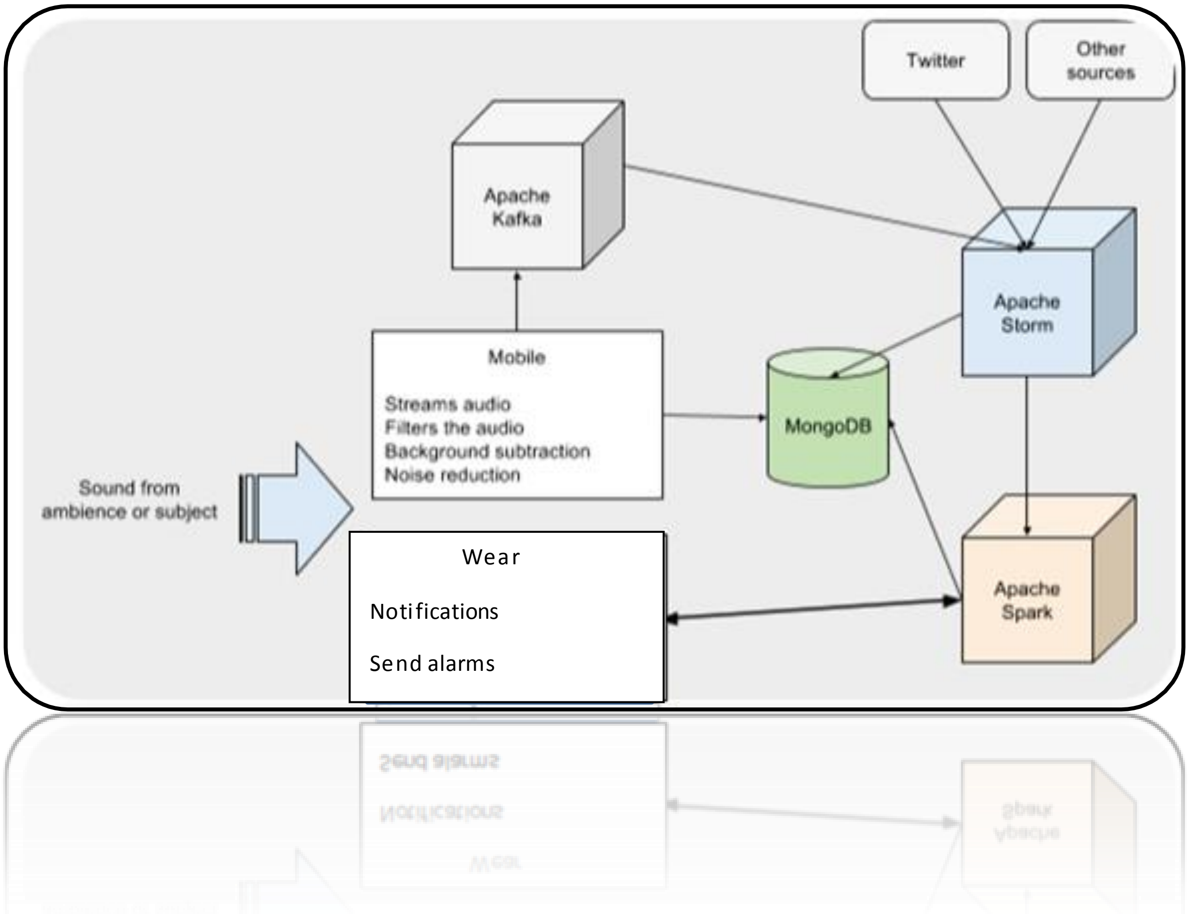
On the server end it would be mostly Scala that would be running on Spark to integrate with the client to perform the machine learning aspect of the application. At the current stage the database that would be in use would MongoDB.

In terms of the architectural framework we are using a traditional 3 layer architecture but the interaction between the layers may vary depending on the situation. In terms of the traditional scenario the recording is present in the database and then the spark engine processes the data to categorize the audio data. In the real time recording scenario the UI layer directly interacts with the spark engine to process the data as it records. The framework would ensure this flexibility is provided to the user.

Framework specifications

The system would be a flexible framework that can incorporate the services that are customized to suit the user and application needs. The system architecture would comprise of a client server architecture in which our application would be a client accessing service hosted or offered through different servers. Internet would be the medium of communication between the application and the services hosted in the web. One of the service that the application offers requires the traditional 3 tier architecture where in the data is stored through the UI layer into the business layer and ends up in the database. The system architecture is represented in the diagram below.

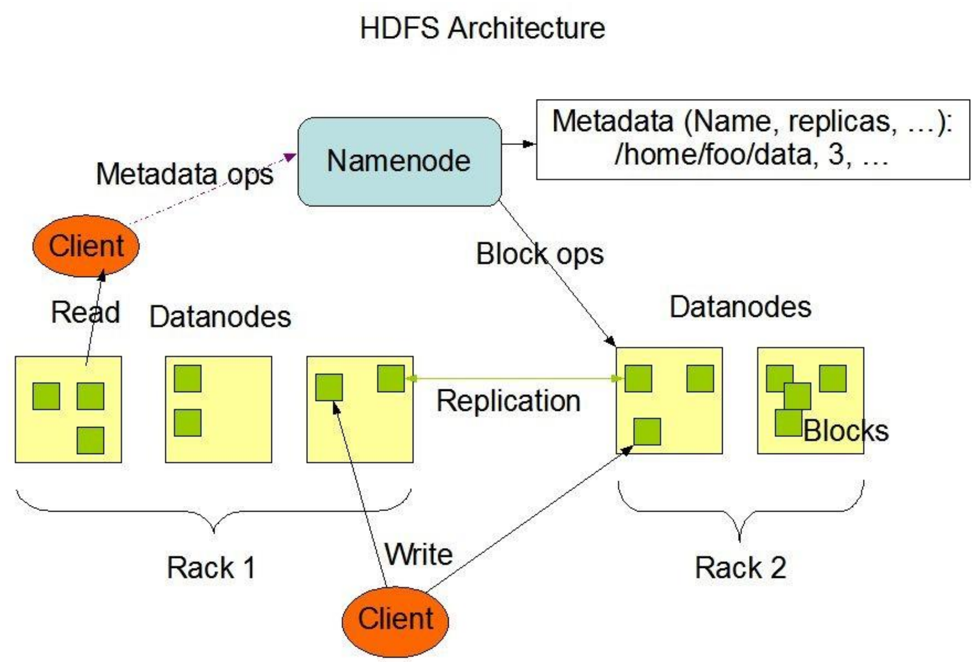
Architecture Diagram:



System specification

Apache Hadoop

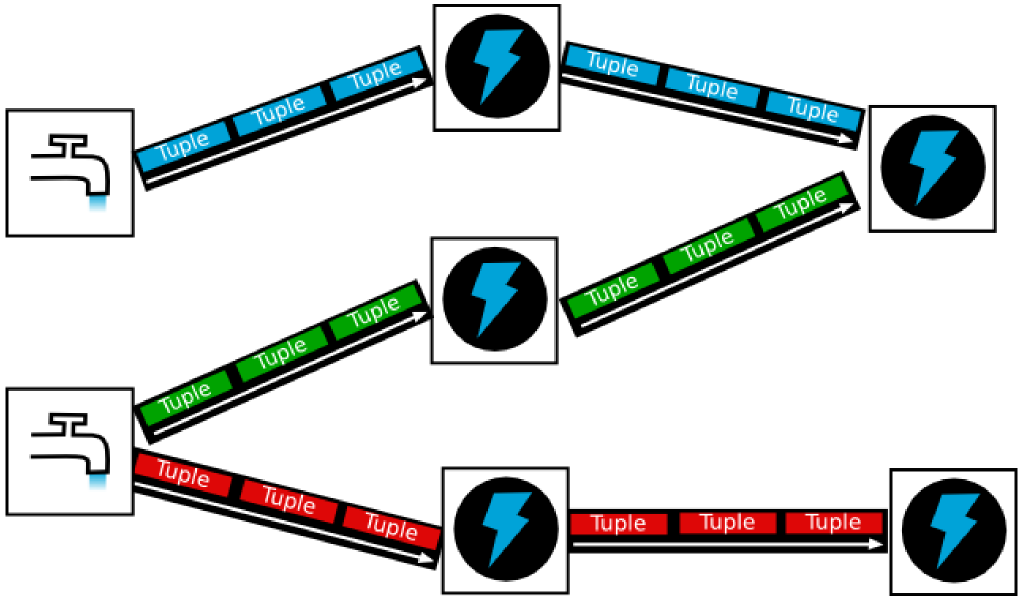
Apache Hadoop is open source framework that processes large amount of distributed data sets among the cluster of nodes using the programming models. Hadoop is developed to be scalable, reliable and distributed computing. It is designed to scale from single server to a cluster of nodes where each node offers local computations and storage. It have two parts namely HDFS and MapReduce. It uses HDFS to store the data where it breaks the input data and distributes to nodes, this enables the parallel processing of data. The MapReduce distributes the programs across the nodes where they performs operations on the data stored in HDFS. Hadoop has YARN which helps in cluster resource management and job scheduling.



3.3.2. Apache Storm

Apache Storm is a distributed computational framework which process high volume of real-time data. It is extremely fast and process millions of records on each node of a cluster at a time. It supports any programming language. Apache storm has three abstractions named spout, bolt and topology. Spout is the source of data, It receives data from kafka or from twitter API or from any other source of information. Bolt processes the input streams of data and can has the ability to produce many output data streams. It works with all the logical computations like functions, filter etc., The topology is the network of spouts and bolts where the edges are connected to bolts which are output streams of spouts.

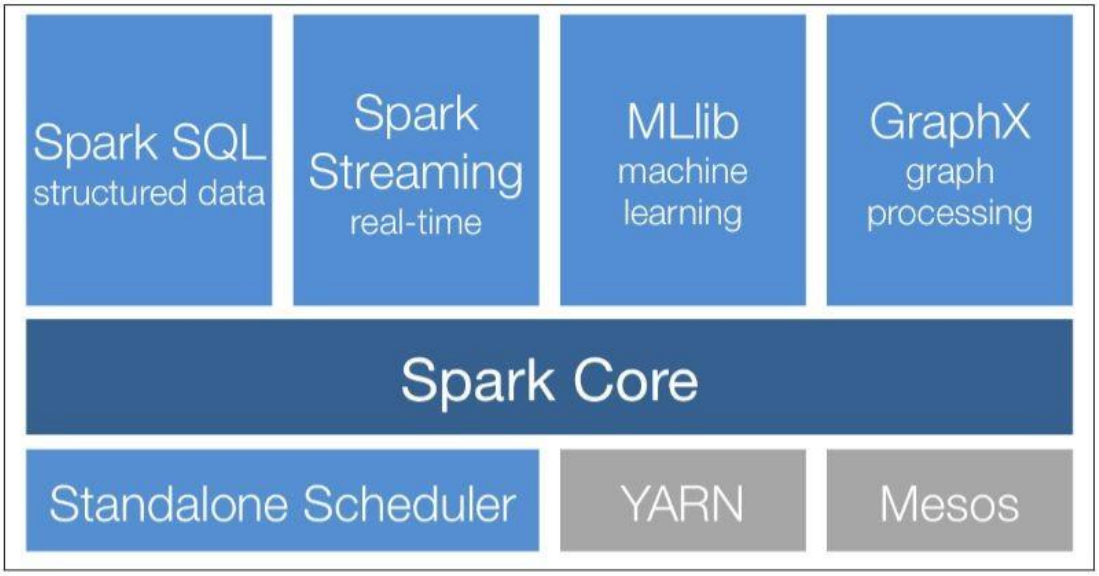
Strom has a Nimbus node which uploads computations,distributes code across the nodes, monitors the computations and instantiates the workers. The zoo-keeper coordinates the cluster and the nimbus node. The supervisor nodes are the worker nodes of storm which works based on the commands of nimbus node. Some of the storm use cases are real-time analytics, distributed RPC and more. Storm is fault-tolerant and is scalable and is easy to setup and operate.



3.3.3. Apache Spark

Apache Spark is a cluster computing platform which has been designed to be fast. Spark extends mapreduce to support more types of computations, Interactive queries and stream data processing. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for structured data processing and using SQL and Apache Hive.The spark

streaming is a spark package which process the live streaming data. MLlib package of spark provides common machine learning algorithms like classifications, regressions, collaborative filtering for machine learning, GraphX package of spark provides API for graph manipulations. Spark has designed to scale up on thousands of nodes in order to achieve this flexibility it supports different cluster managers like apache mesos, Hadoop YARN. Spark computes distributed datasets on the files stored in the file system. Even if the dataset is lost it recomputes using Lineage graphs. These RDD's supports two operations named transformations and actions. Transformations produces new RDD's whereas the actions bring the results back to the driver programs.

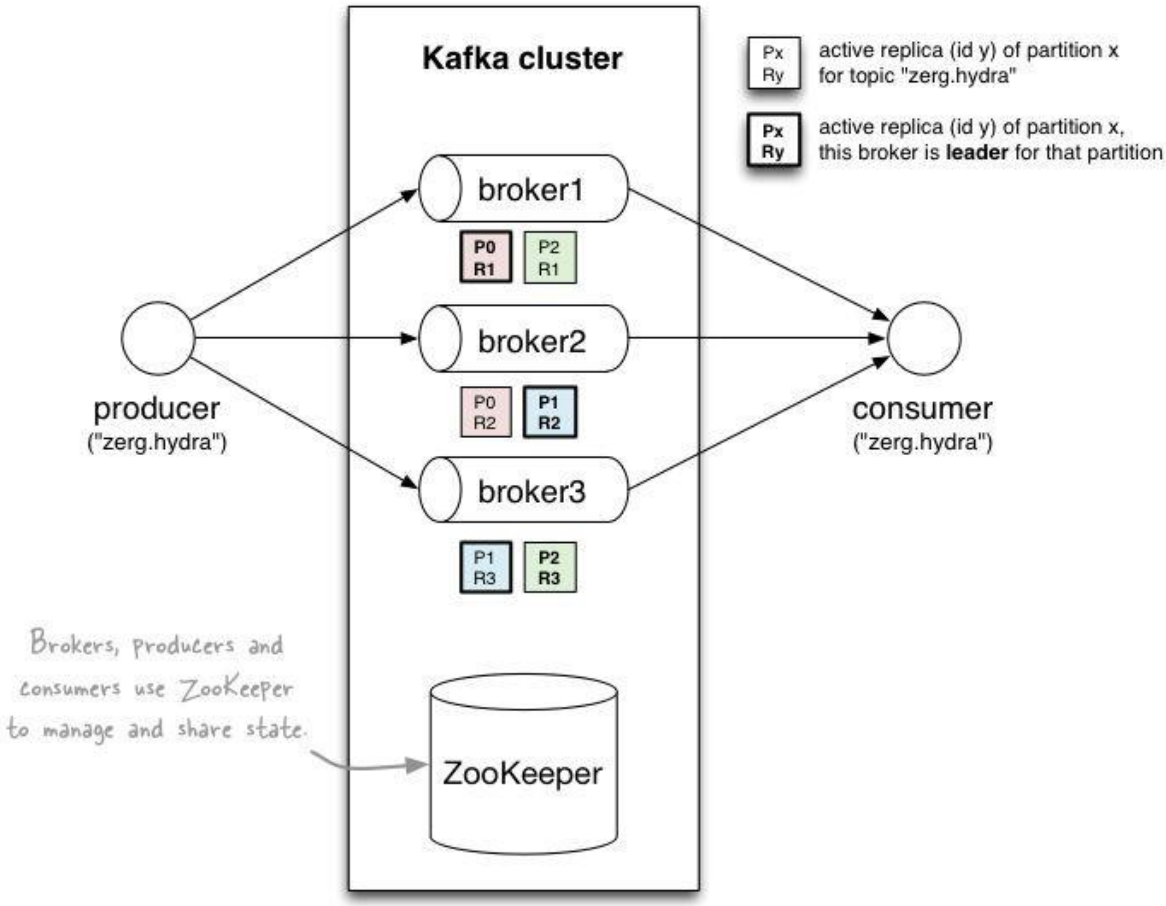


3.3.4. Apache Kafka

Apache Kafka is a high distributed messaging system with replicated,partitioned commit log services. It was originated at Linkedin. Implemented in scala and java. Kafka supports high throughput to support huge volume of data, Supports real-time processing of these feeds to create new desired feeds, Supports large data backlogs to handle any system faults, Guarantees fault-tolerance in case of machine failure. Supports low-latency delivery. Kafka has high reads and high writes. Kafka has main 3 core concepts Producer, Brokers, Consumers.

Producers write data to Brokers.Consumers read data from Brokers for further processing. All these processes are distributed. Data is stored in the form of 'Topics'. Topics are splitted into partitions ,these partitions are later replicated. Kafka treats each partition of topics

as a log. Partitions are immutable, ordered sequence of messages which will be appended continuously. Consumers subscribe to these 'topics' to receive data transferred by producer. Kafka requires Zookeeper. Zookeeper is used by brokers and consumers but not by producers. In newer versions of Kafka v0.9, zookeeper is only used by brokers, consumers uses topics instead of Zookeeper.



3.3.5. MongoDB

MongoDB is a open-source document based database which stores the data in JSON format. MongoDB is classified as a NOSQL database. It can be used as a file system by using the concept of load balancing and replications.

After processing the audio signal the classification of the signal which obtained is stored in the MongoDB. We store the classification in three collections named

1. Users
2. Locations and
3. Settings

It stores the audio files, users, user ids, tags,latitude, longitude, time in the collections.

3.3.6. R (RStudio)

R is one of the famous data analytics tool available and being used by researchers from many decades. R is well known as a Software environment as well as a programming language for graphics and statistical computing used majorly for data mining by data miners and statisticians.

Statistical and Graphical techniques such as Linear and Nonlinear modeling, clustering, classification and others are implemented using R and its libraries. When compared to most of the statistical computing languages, R has much stronger Object Oriented Programming facilities. Static graphs is the other major strength of R.

RStudio is the open source IDE which supports R programming language and it is available for free in two editions. One is the RStudio Desktop edition which is used to run desktop applications locally. The other is the RStudio Server which uses web browsers to access RStudio running on a remote linux server.

V. Project plan

The proposed project plan is outlined the by the screen shot from the Kanban tool. The project is divided into four phases each of which is guide lined by a test driven approach. Each iteration has four tasks categorized by stories of designing phase, building the service through implementation and testing. The final task is to store the completed tasks for future integration.

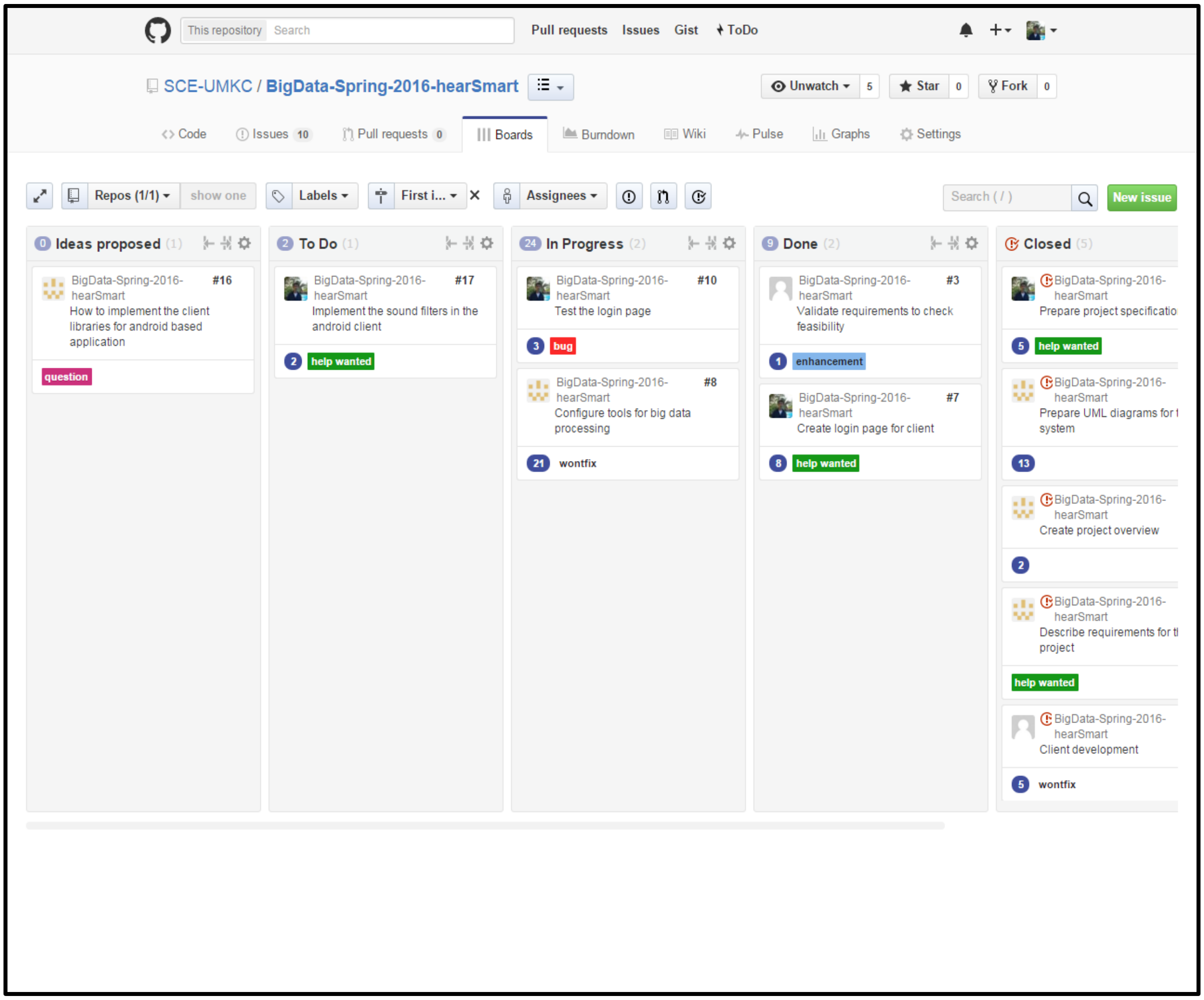


Figure 2: Depicting the schedule and plan for phase 1 of the project.

Phase 1:

This phase mainly deals with the designing the system for the implementation phase. The tasks of this phase mainly focuses on the UML diagram and collecting the necessary information for the realization and the implementation of the system. The end result of the phase 1 are initial screens of the application. These are just the rudimentary level implementation screens and these may change during application development process.

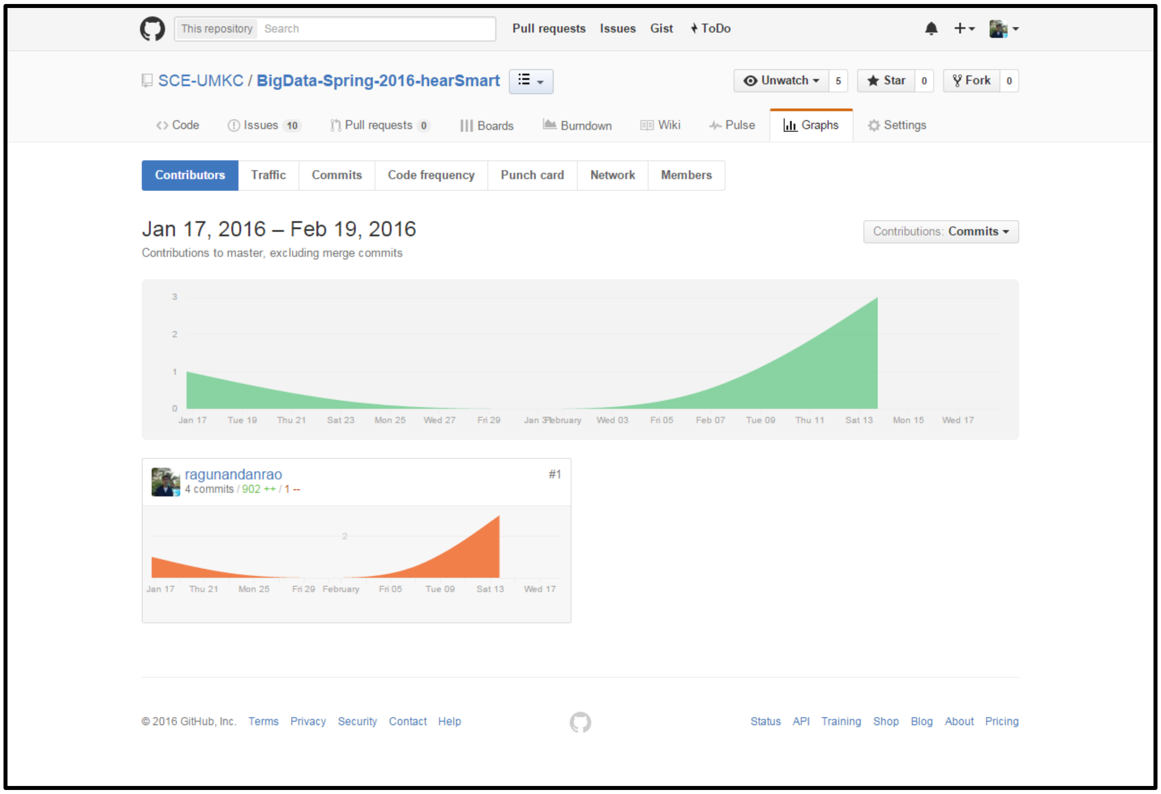


Figure 3: Depicting the contribution of each person to the repository

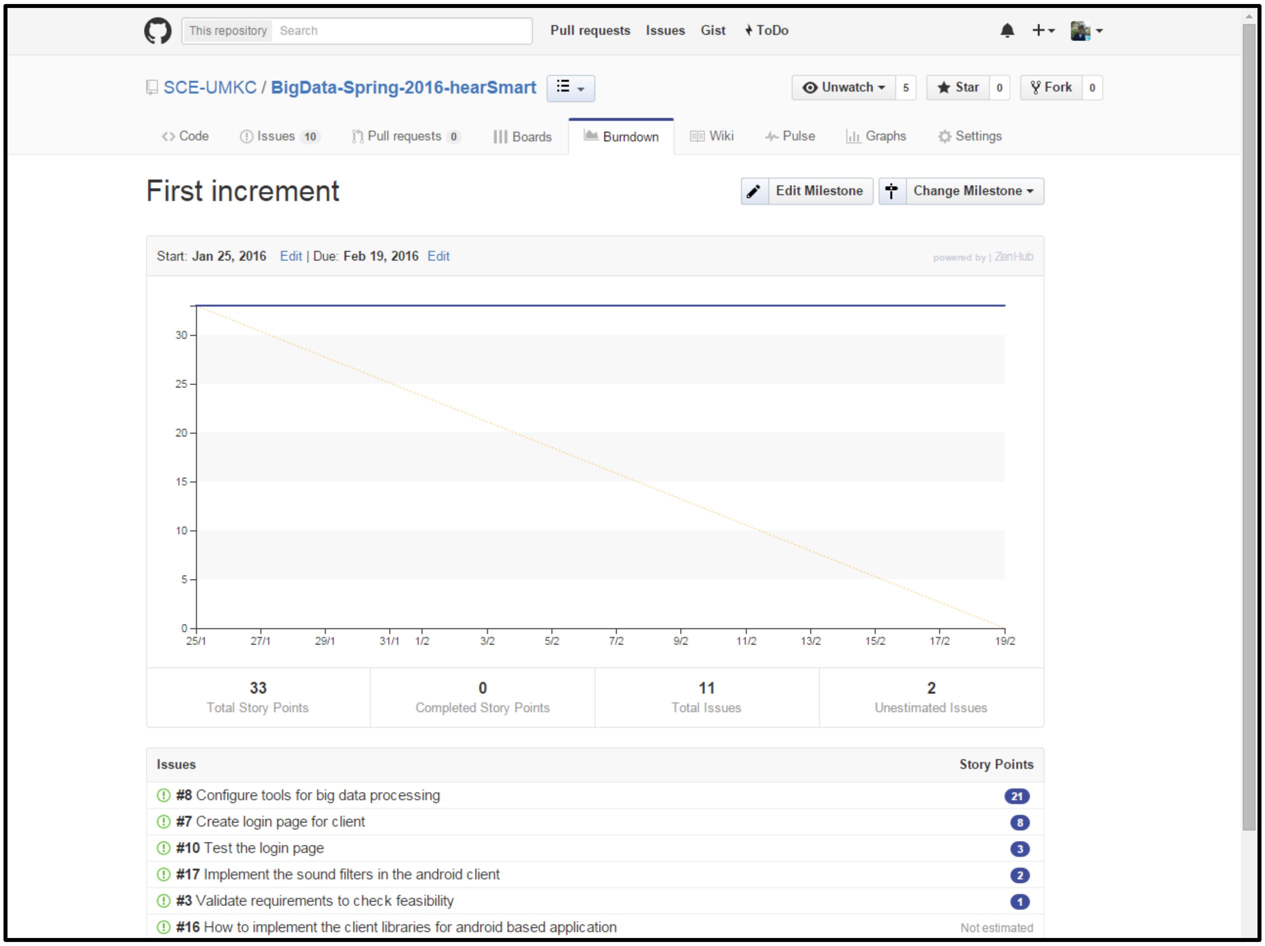


Figure 4: Depicting the analytical graph for the milestone of first increment.

Timelines of the project

The timelines of the project schedule are in coherence with the timelines of submission for each of the phase. As of the current state there are no deviations from the project schedule.

VI. First Increment Report

This document is a report of first iteration of work performed on the Smart Hear Mobile App. This App proposes to implement various web services into single App. The document emphasizes on the pictorial representation of the application using different implementations which gives an insight on internal system. This document intends to provide an overall description of the project named "Smart Hear".

The main outcome of the first increment is the high and low level designs of the App. As of current state we have not deviated from our initial proposal that we submitted earlier. We have taken care of the implementation of Class Diagrams and Sequence Diagrams which showcase the flow of our application. The blueprint of the application is generated using the Wireframes and are available in this document.

During the course of the first increment our main focus was on the idea and to get to know the feasibility of the various use cases provided in the proposal. We mainly concentrated on the finding the resources for the project in terms of papers on the topic and also in terms of the work that already done. Currently the implementation is very minimal and we believe that it can be caught up very soon. The team's immediate goal is to get a basic application that record sounds from the microphone of the smart phone and perform mutations such as frequency, pitch other feature modulations.

Collaboration diagram for the high level design of the system

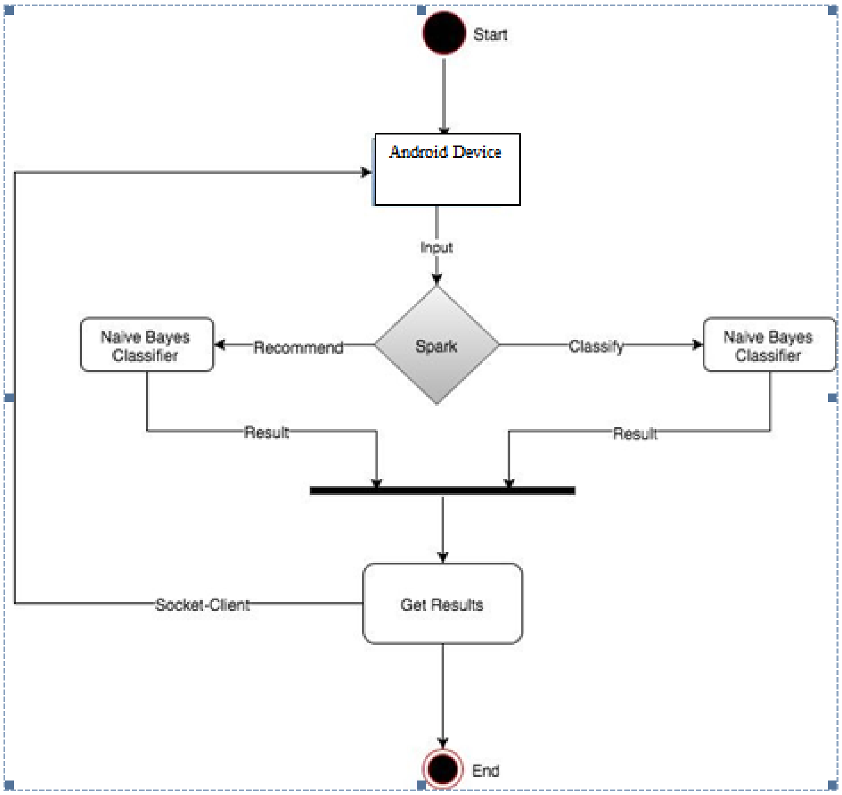


Figure 1

Figure 1 demonstrates a class diagram which explains High Level Design Architecture of the System.

Sequence diagram for each of the web service interaction

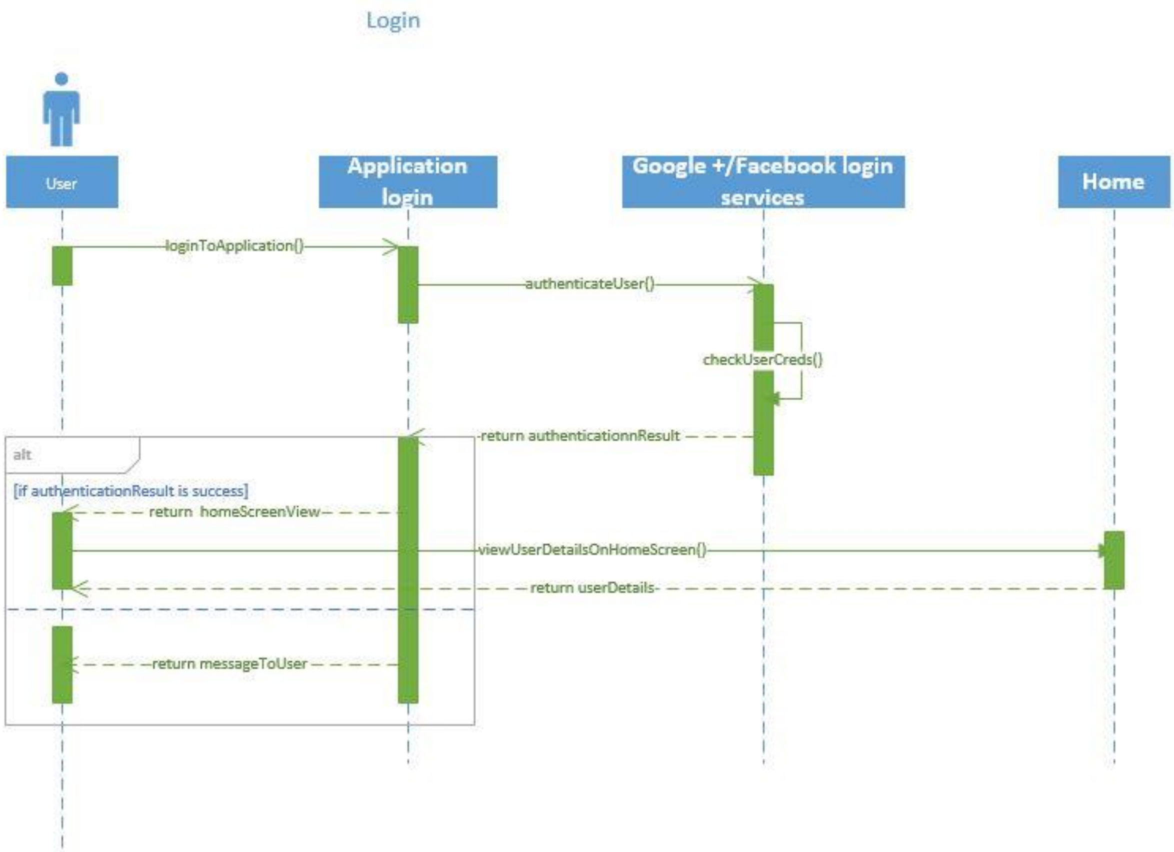


Figure 2

Figure 2 demonstrates a sequence diagram about how a user can Login to the App by using different Sign In options like Facebook, Google+.

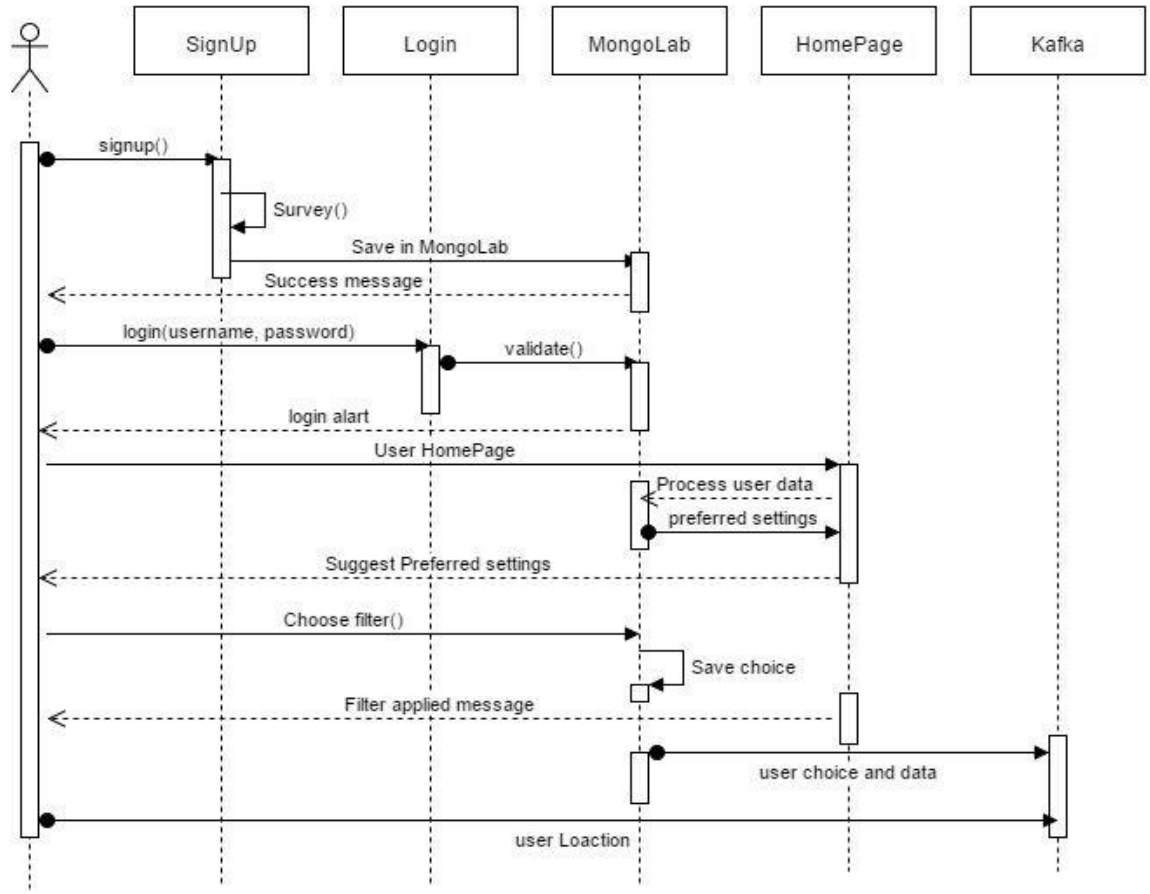


Figure 3

Figure 3 demonstrates a Sequence diagram about how a user can get the user's preferred settings.

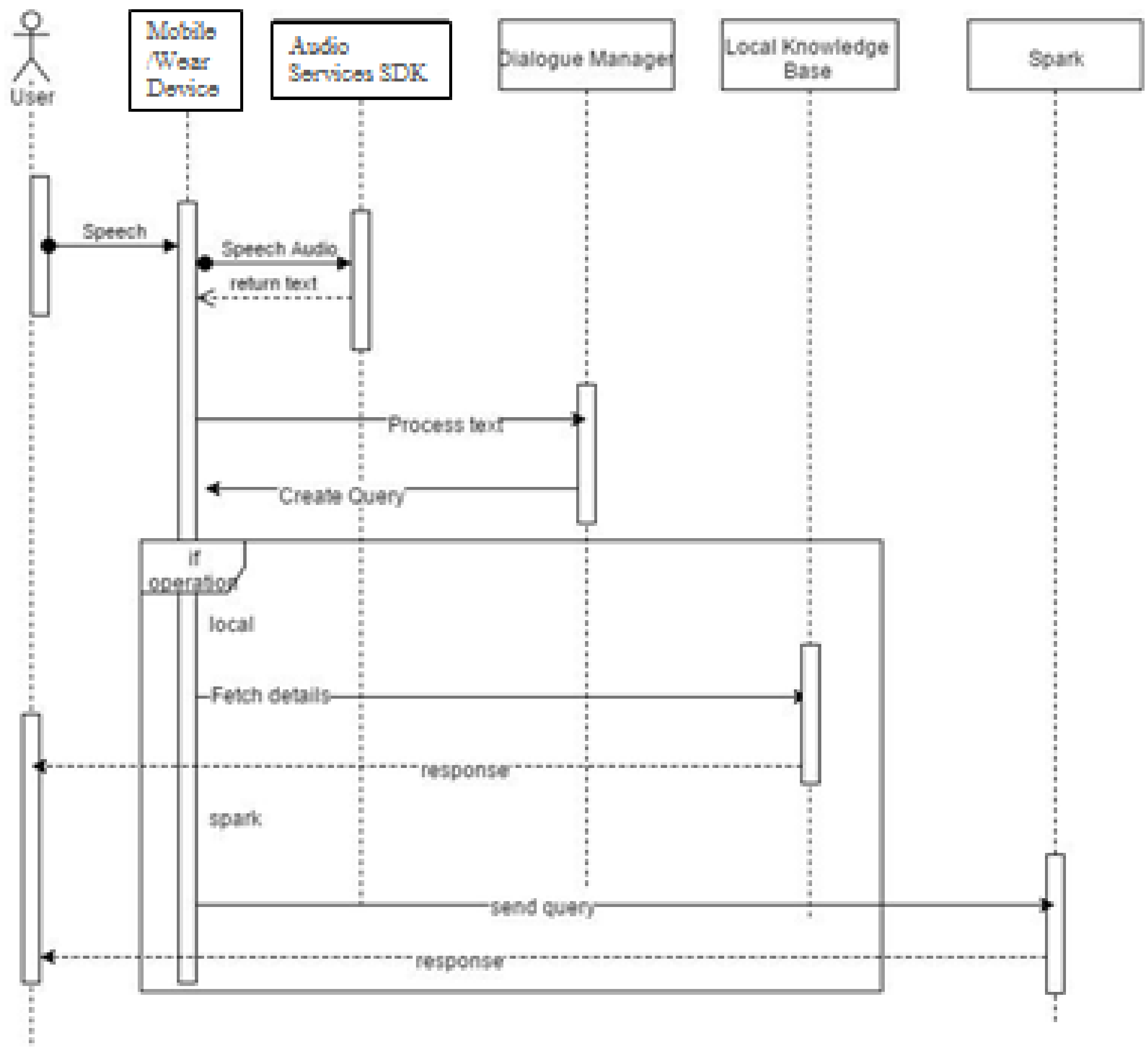


Figure 4

Figure 4 demonstrates a sequence diagram about how a user can use this App to Fetch Audio details and analyze results .

.

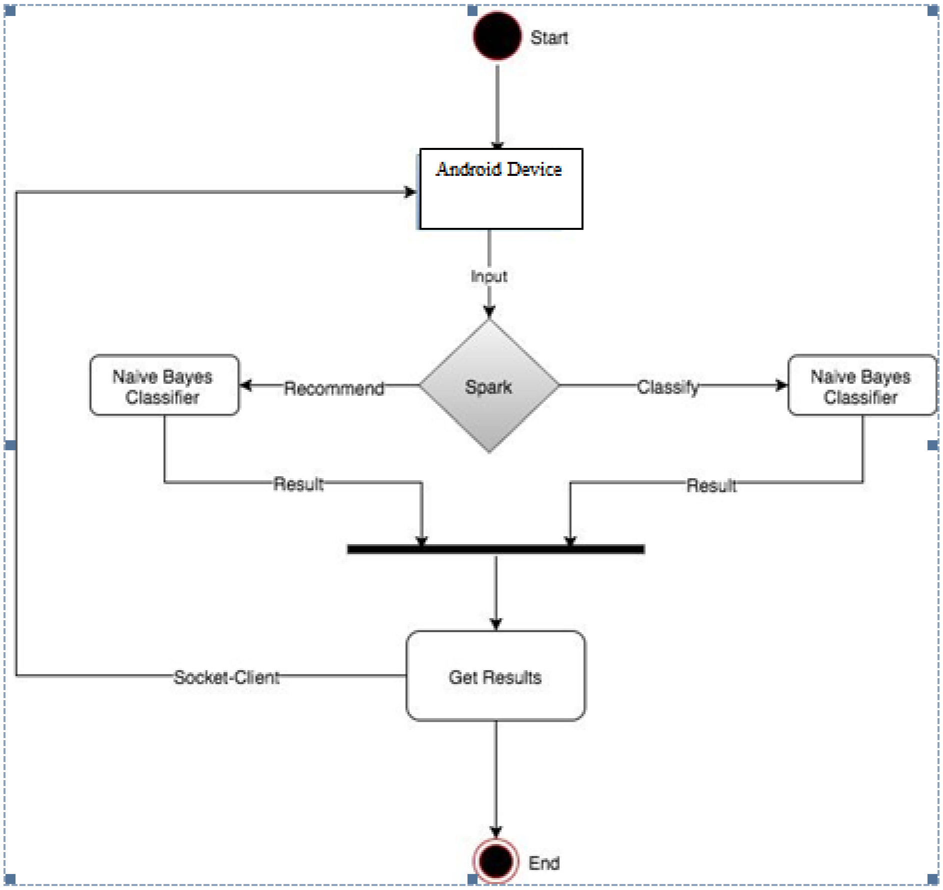


Figure 5

Figure 6 demonstrates a Activity diagram about how a user can get the results of the Specified Audio Filters.

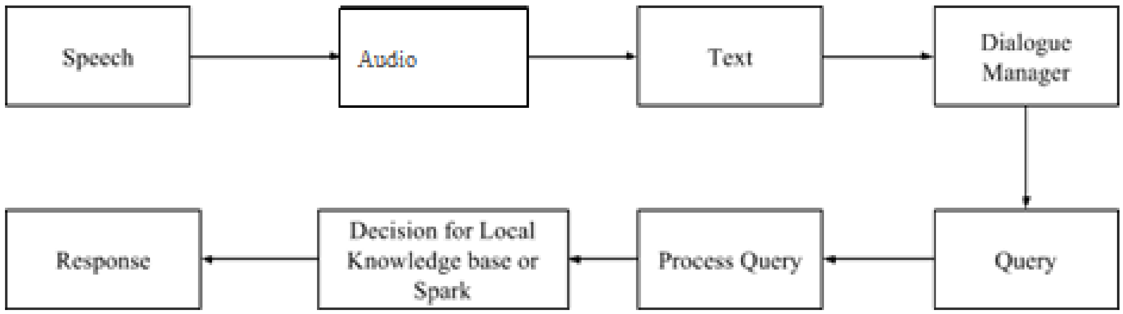


Figure 6

Figure 6 demonstrates a work flow diagram about how a user can generate responses from Speech.

Activity diagram of the system

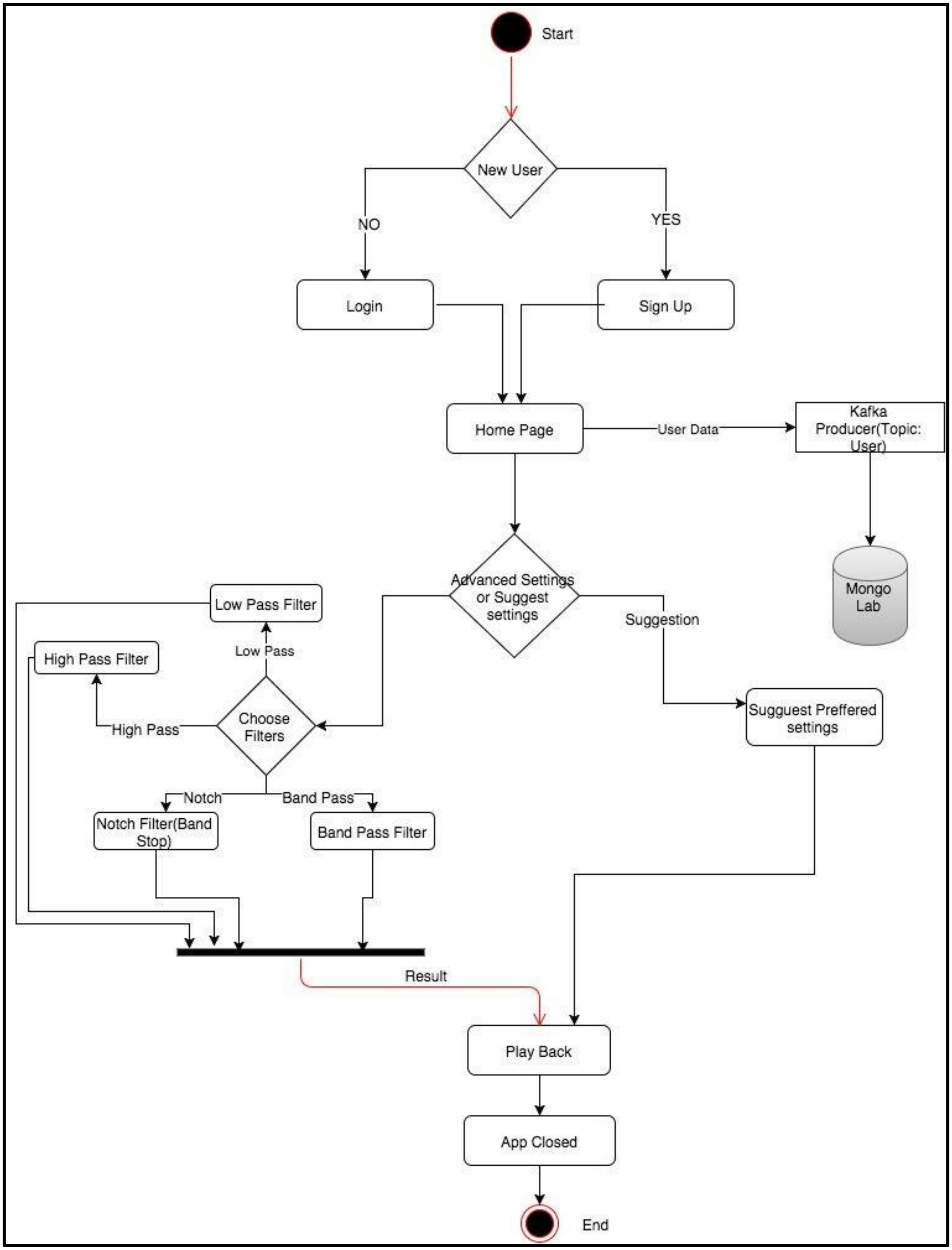
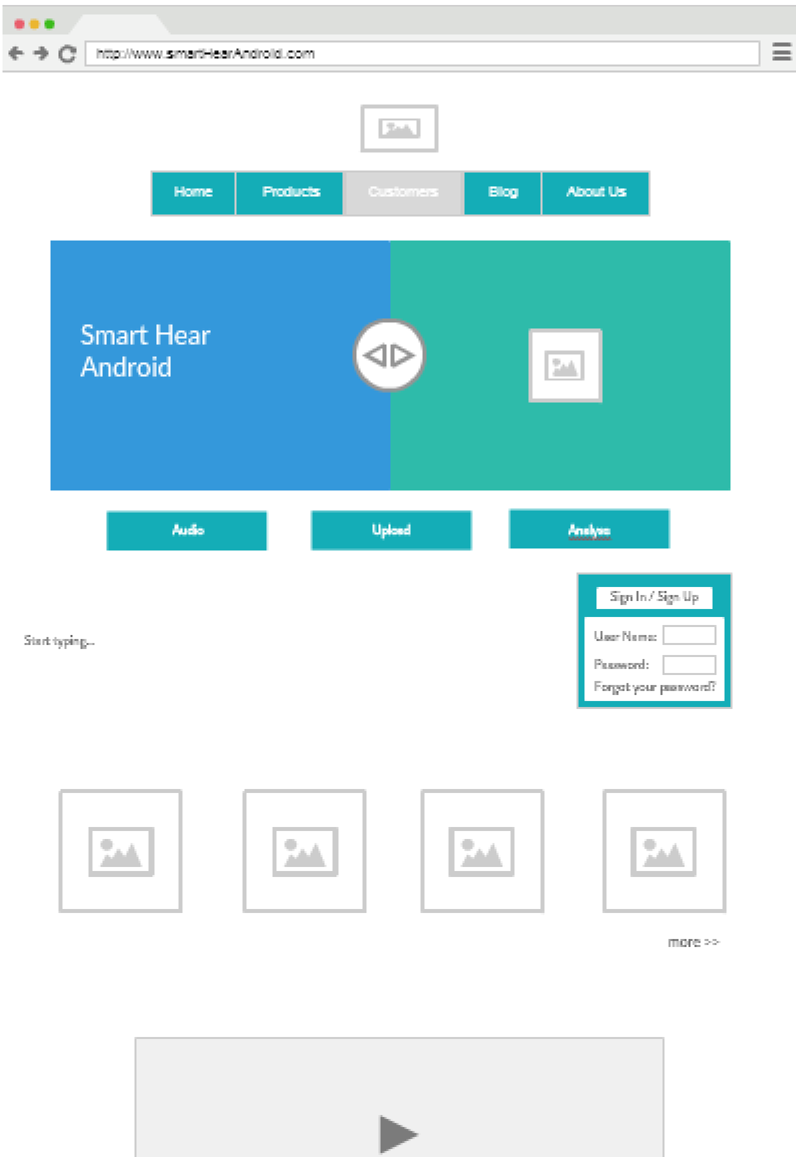
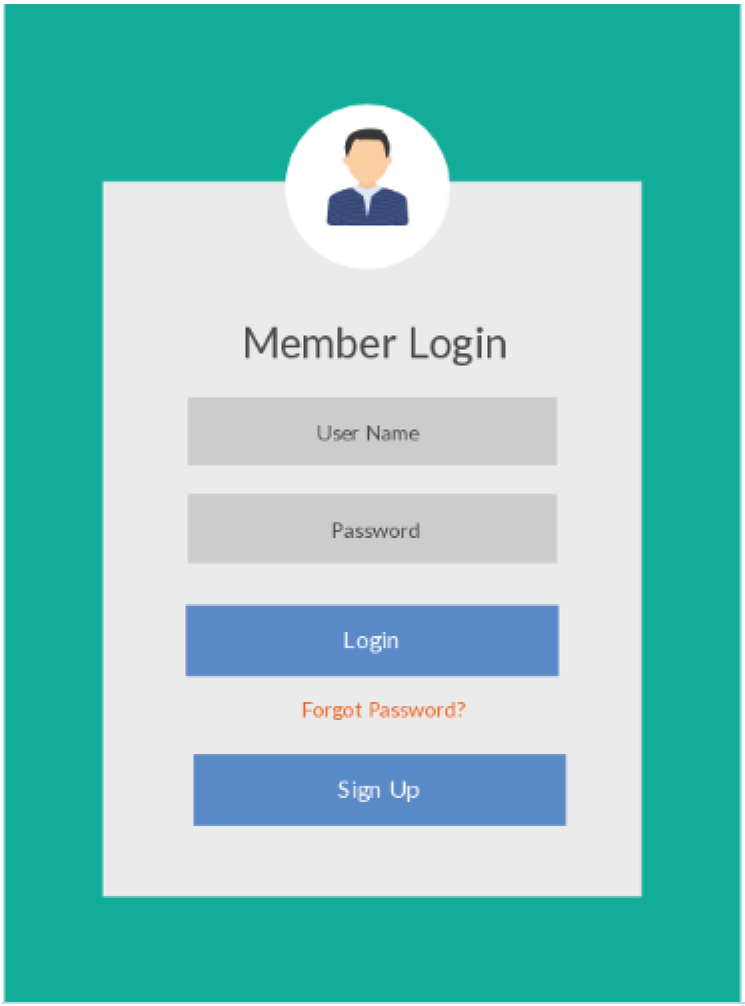


Figure 9

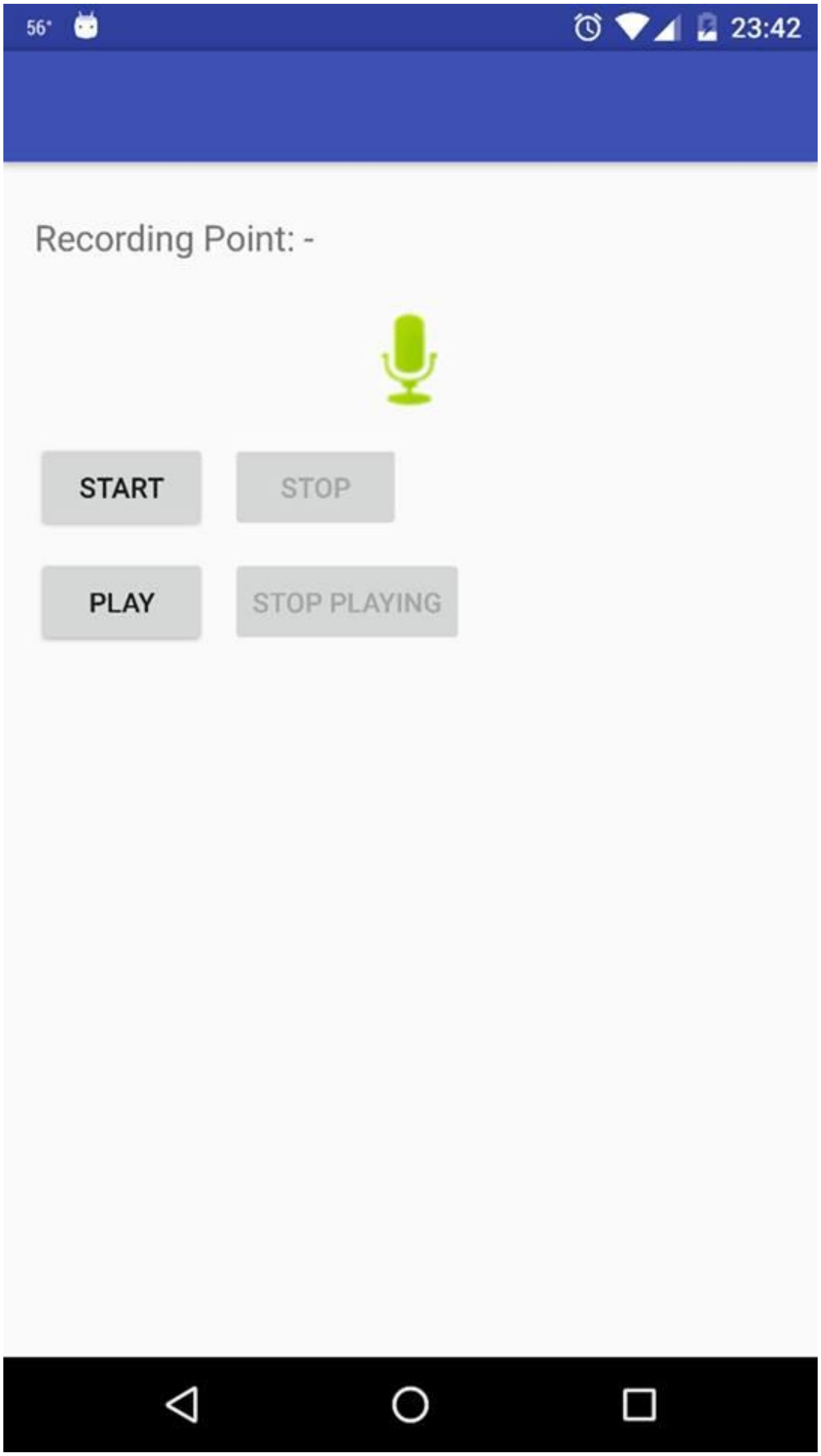
Figure 9 demonstrates a state diagram which explains High Level Design Architecture of the System.

Wireframes for the application




Member Login
User Name
Password
Login
Forgot Password?
Sign Up

Screenshots



VII. Bibliography

- https://play.google.com/store/apps/details?id=mg.locations.track5&hl=en
- https://play.google.com/store/apps/details?id=com.fsp.android.friendlocator&hl=en
- http://www.raywenderlich.com/120177/beginning-android-development-tutorial-installing-android-studio
- http://developer.android.com/tools/building/building-studio.html
- https://usa.bestsoundtechnology.com/ces/#
- http://www.hearingreview.com/products/new-product-technology/
- http://www.resound.com/en-US/hearing-aids/resound-linx2#.Vse5ZPkrLcs

Smart Hear - Intelligent Hearing for Android

Project Second Iteration Report

Submitted on **11 March 2016**

Group 1:

1. **Ragunandan Rao Malangully (13)**
2. **Ravi Kiran Yadavalli (31)**

I. Introduction

This document intends to provide an overall description of the project named “Smart Hear” in detail. The project schedule and the plan of action is also discussed. The proposal document submitted would give an insight about what the project is about. The main outcome of the Second increment was the high and low level design of the application. As of current state we have not deviated from our initial proposal that we submitted earlier.

II. Project goal and objectives

Overall goal

The overall goal is to provide a hearing aid through the use of the smart phone that is accessible to every person today. The smart phone can be utilized in a way that it can act similar to the hearing dog that is what motivated us to take up this idea and try to implement using the smart phone.

This application can be used to provide benefits to the special abled people who face challenges in listening to sound. All the features come at the cost of installing an application that is freely provided to the user. No costs attached.

Objectives

- To provide a smart hearing aid to the user.
- To implement features to provide the user flexibility in varying certain key features of sound like frequency, tempo etc.
- To provide notifications to the user when there are important events like a door bell ring or an alarm that goes off.
- To develop a user friendly application.
- To ensure that we have a light weight application on the client end.
- To make sure that the context is recognized and the user settings are changed accordingly.
- Implement knowledge discovery to provide a summary of a topic that the user is listening or recording.
- Provide user customizable settings that the user can adjust to suit his or her needs.
- Ensure that the application can deliver its functionality at minimal cost of operation.
- Provide a user guide to understand and use the application.

III. Project background and related work

The app is not a new invention or a brand new concept. But the implementation aspect of it is what makes it unique from the others. The following are the projects that are similar to what our idea is.

- Resound LiNX: It is a smart hearing aid kit that can integrate with Apple Iphone and other android devices. The device does noise cancellation and also amplifies the sound that the user is listening to. The device can also integrate with smart watches and provide very handy functionality. But the downfall is that it is an additional device that has to be purchased and is does not have notification features to the user.
- Siemens Hearing Aids: These were introduced at the International CES 2015. They are pretty close to what an interactive hearing aid can deliver and seamlessly work hand in hand with smart devices. They require a specific app to connect the device with the hand held devices and only then can they provide the functionality. On the flipside the product is priced high and is not for everyone to afford it.

Significance

The major significance of the application lies in the fact that the all the useful services are provided under one system. There is no need of another device for the hearing enhancements. The user can access all the features through his smart phone. Even form the implementation aspect the usage of big data framework would ensure that the application can process the data at high performance rates. This would also reduce the cost of the creating and operating the application. User customizable settings is one of the major significance of the project.

IV. Proposed system

Requirements specifications

- The application will provide users facility to login with their Google plus account.
- User should have the flexibility to login with his Facebook account.
- The system will display the user his information along with his profile picture.

- An Android application that serves as the smart hearing aid and also provides a user friendly model for the feature extraction.
- A machine learning framework that can provide a model based on the features received from the client end.
- Analysis of audio files to observe their characteristics.
- Collection of features present in an audio file and weighing the features to select the vital features dynamically.
- To send text notifications on various events based on the pattern of previous audios.
- To be able to detect and send alarms on to the device in potential hazardous situations.

- Integration of the client and the processing framework for a complete application that can perform as a smart hearing device.

Workflow analysis

The first and the most fundamental workflow of the application is that of the designing the UI for the system. This involves selecting the proper HTML elements and arranging the elements in an appropriate manner. It includes making use of the CSS attributes to style the UI of the application. Each page of the application is to be custom designed for suiting the needs of the service or feature being provided over that page. This would also include capturing the sound from the device and storing it in the device storage.

Second task would be do perform basic filtering and other alterations to the sound using libraries such as high pass, band pass etc. The frequency, tempo, pitch and other features of the sound can be altered and extracted individually.

The third workflow deals with the performing model discovery or applying the best fit model over the features extracted. This would involve applying machine learning algorithms like Decision tree etc. to classify the source of the sound.

Technological and architectural requirements

On the technological aspect the project is pretty much dependent on the native Android libraries for implementing the client functionality. This would also include some external Java or Android based libraries or plugins for managing the operations on the sound recorder. The primary development language for the client would Java and XML for the UI part.

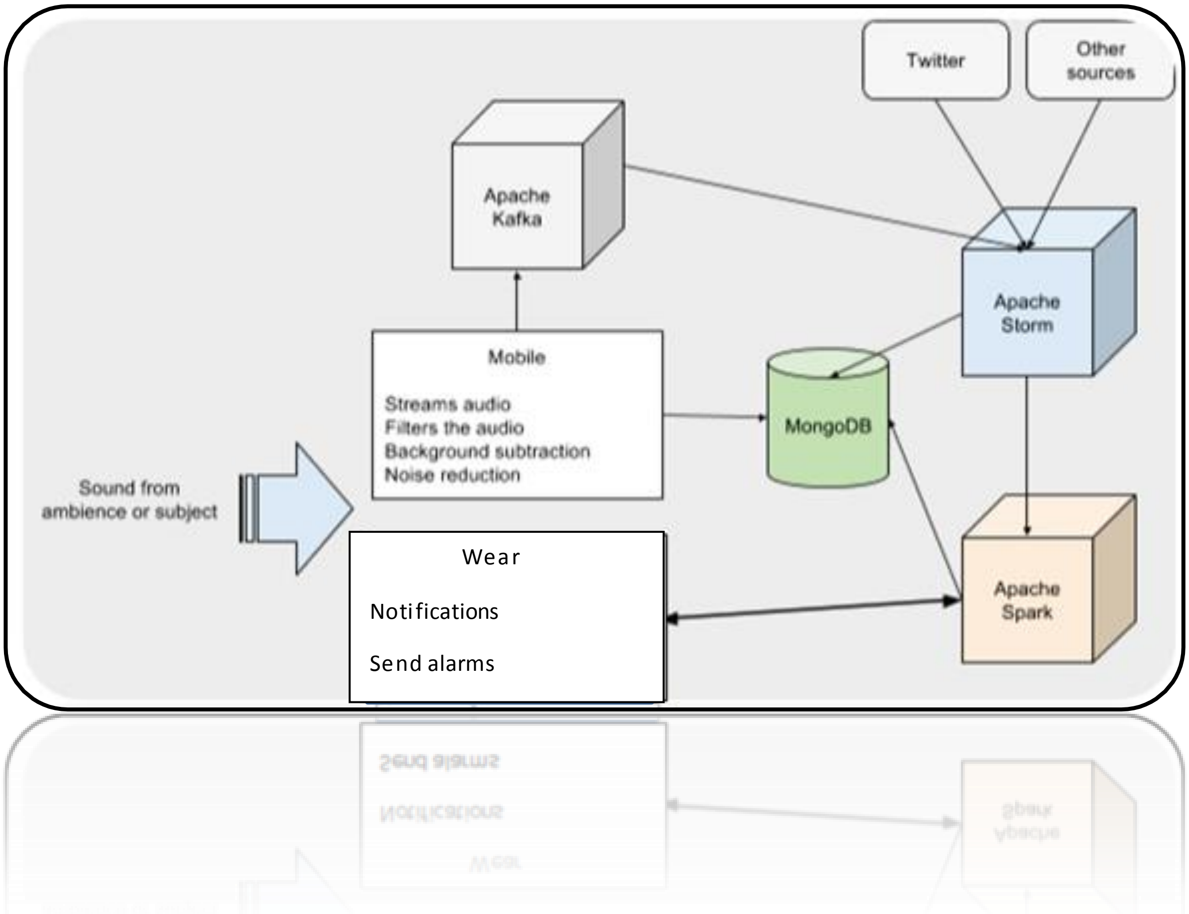
On the server end it would be mostly Scala that would be running on Spark to integrate with the client to perform the machine learning aspect of the application. At the current stage the database that would be in use would MongoDB.

In terms of the architectural framework we are using a traditional 3 layer architecture but the interaction between the layers may vary depending on the situation. In terms of the traditional scenario the recording is present in the database and then the spark engine processes the data to categorize the audio data. In the real time recording scenario the UI layer directly interacts with the spark engine to process the data as it records. The framework would ensure this flexibility is provided to the user.

Framework specifications

The system would be a flexible framework that can incorporate the services that are customized to suit the user and application needs. The system architecture would comprise of a client server architecture in which our application would be a client accessing service hosted or offered through different servers. Internet would be the medium of communication between the application and the services hosted in the web. One of the service that the application offers requires the traditional 3 tier architecture where in the data is stored through the UI layer into the business layer and ends up in the database. The system architecture is represented in the diagram below.

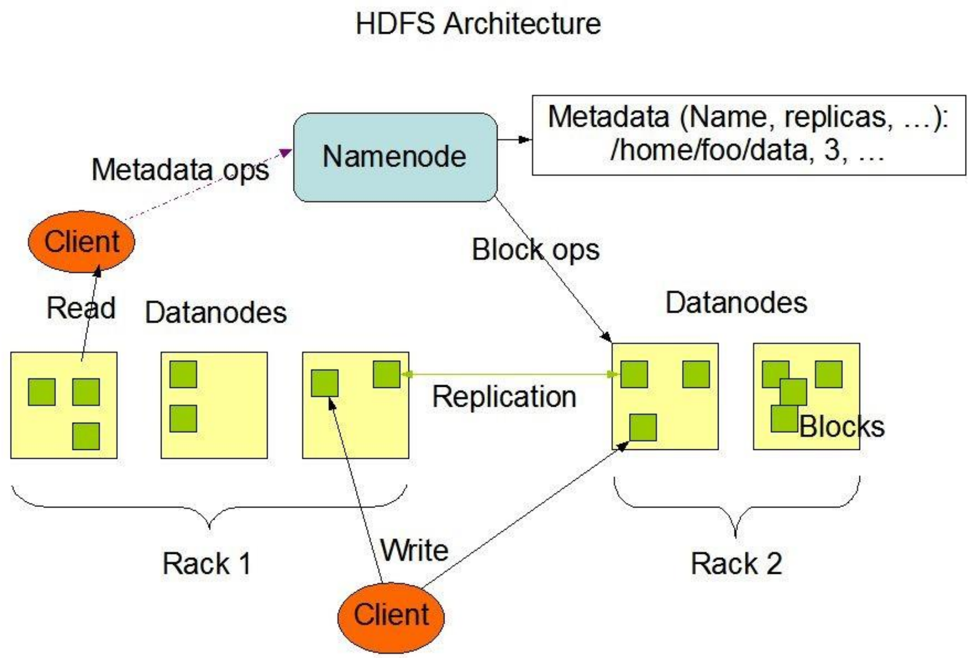
Architecture Diagram:



System specification

Apache Hadoop

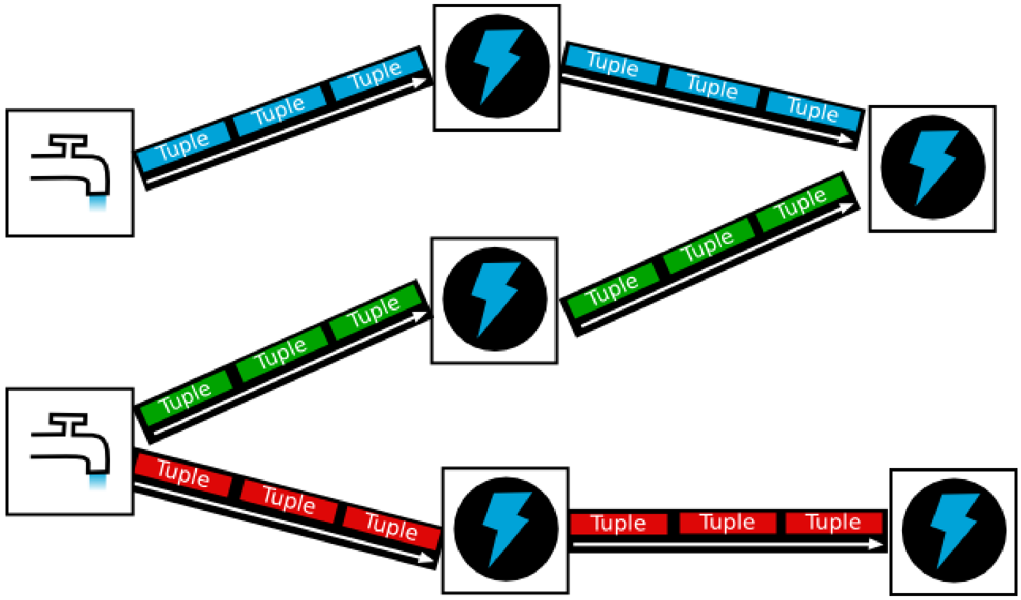
Apache Hadoop is open source framework that processes large amount of distributed data sets among the cluster of nodes using the programming models. Hadoop is developed to be scalable, reliable and distributed computing. It is designed to scale from single server to a cluster of nodes where each node offers local computations and storage. It has two parts namely HDFS and MapReduce. It uses HDFS to store the data where it breaks the input data and distributes to nodes, this enables the parallel processing of data. The MapReduce distributes the programs across the nodes where they performs operations on the data stored in HDFS. Hadoop has YARN which helps in cluster resource management and job scheduling.



3.3.2. Apache Storm

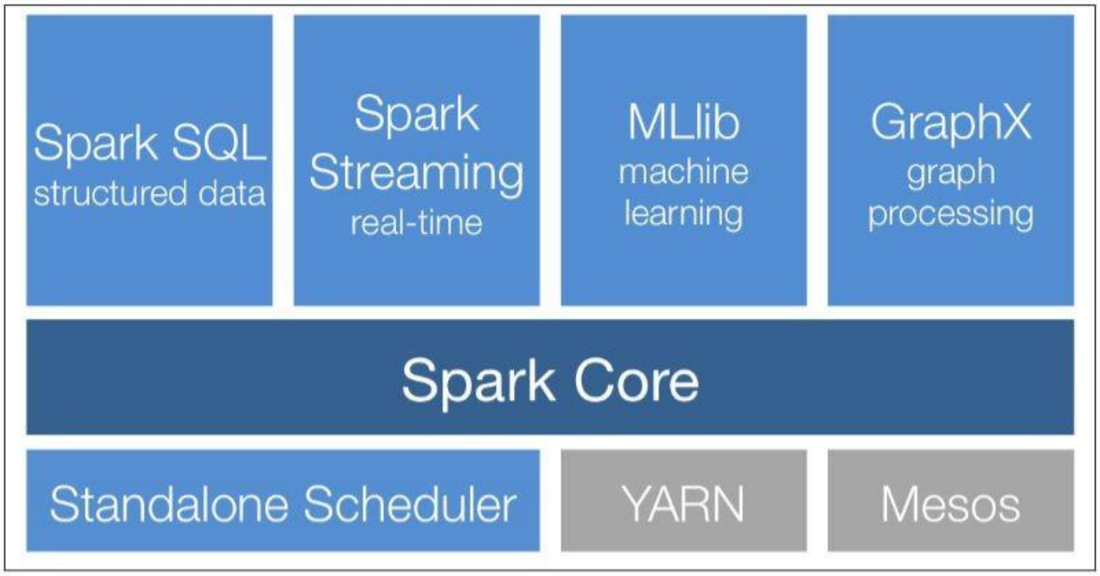
Apache Storm is a distributed computational framework which process high volume of real-time data. It is extremely fast and process millions of records on each node of a cluster at a time. It supports any programming language. Apache storm has three abstractions named spout, bolt and topology. Spout is the source of data, It receives data from kafka or from twitter API or from any other source of information. Bolt processes the input streams of data and can has the ability to produce many output data streams. It works with all the logical computations like functions, filter etc., The topology is the network of spouts and bolts where the edges are connected to bolts which are output streams of spouts.

Strom has a Nimbus node which uploads computations, distributes code across the nodes, and monitors the computations and instantiates the workers. The zoo-keeper coordinates the cluster and the nimbus node. The supervisor nodes are the worker nodes of storm which works based on the commands of nimbus node. Some of the storm use cases are real-time analytics, distributed RPC and more. Storm is fault tolerant and is scalable and is easy to setup and operate.



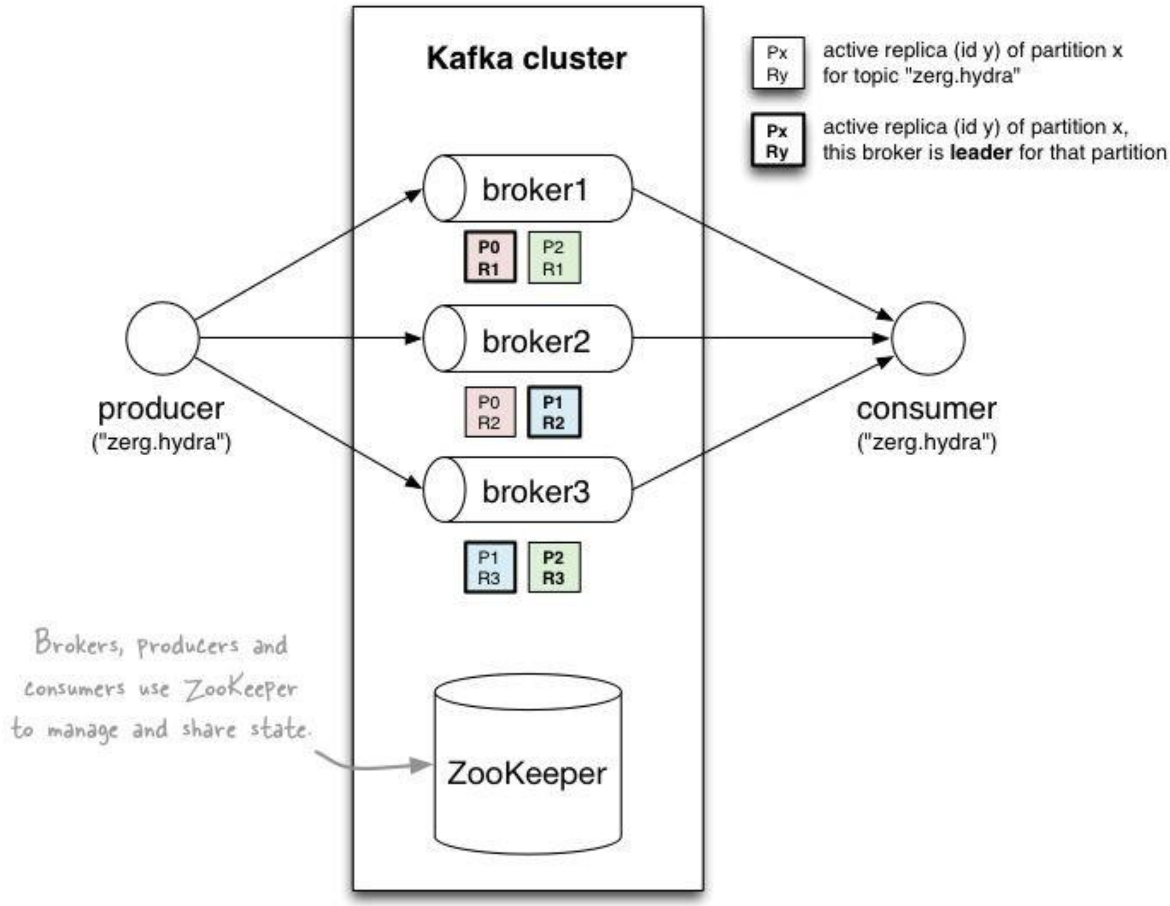
3.3.3. Apache Spark

Apache Spark is a cluster computing platform which has been designed to be fast. Spark extends MapReduce to support more types of computations, Interactive queries and stream data processing. It provides high level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for structured data processing and using SQL and Apache Hive.



3.3.4. Apache Kafka

Apache Kafka is a high distributed messaging system with replicated, partitioned commit log services. It was originated at LinkedIn. Implemented in scala and java. Kafka supports high throughput to support huge volume of data, Supports real-time processing of these feeds to create new desired feeds, Supports large data backlogs to handle any system faults, Guarantees fault tolerance in case of machine failure. Supports low latency delivery. Kafka has high reads and high writes. Kafka has main 3 core concepts Producer, Brokers, Consumers.



3.3.5. MongoDB

MongoDB is an open source document based database which stores the data in JSON format. MongoDB is classified as a NOSQL database. It can be used as a file system by using the concept of load balancing and replications.

After processing the audio signal the classification of the signal which obtained is stored in the MongoDB. We store the classification in three collections named

1. Users
2. Locations and
3. Settings

It stores the audio files, users, user ids, tags, latitude, longitude, time in the collections.

3.3.6. R (RStudio)

R is one of the famous data analytics tool available and being used by researchers from many decades. R is well known as a Software environment as well as a programming language for graphics and statistical computing used majorly for data mining by data miners and statisticians.

Statistical and Graphical techniques such as Linear and Nonlinear modeling, clustering, classification and others are implemented using R and its libraries. When compared to most of the statistical computing languages, R has much stronger Object Oriented Programming facilities. Static graphs is the other major strength of R.

RStudio is the open source IDE which supports R programming language and it is available for free in two editions. One is the RStudio Desktop editi on which is used to run desktop applications locally. The other is the RStudio Server which uses web browsers to access RStudio running on a remote linux server.

V. Project plan

The proposed project plan is outlined the by the screen shot from the Kanban tool. The project is divided into four phases each of which is guide lined by a test driven approach. Each iteration has four tasks categorized by stories of designing phase, building the service through implementation and testing. The final task is to store the completed tasks for future integration.

Phase 2:
This phase mainly deals with the designing the system for the implementation phase. The tasks of this phase mainly focuses on the UML diagram and collecting the necessary information for the realization and the implementation of the system. The end result of the phase 1 are initial screens of the application. These are just the rudimentary level implementation screens and these may change during application development process.

Timelines of the project

The timelines of the project schedule are in coherence with the timelines of submission for each of the phase. As of the current state there are no deviations from the proje ct schedule.

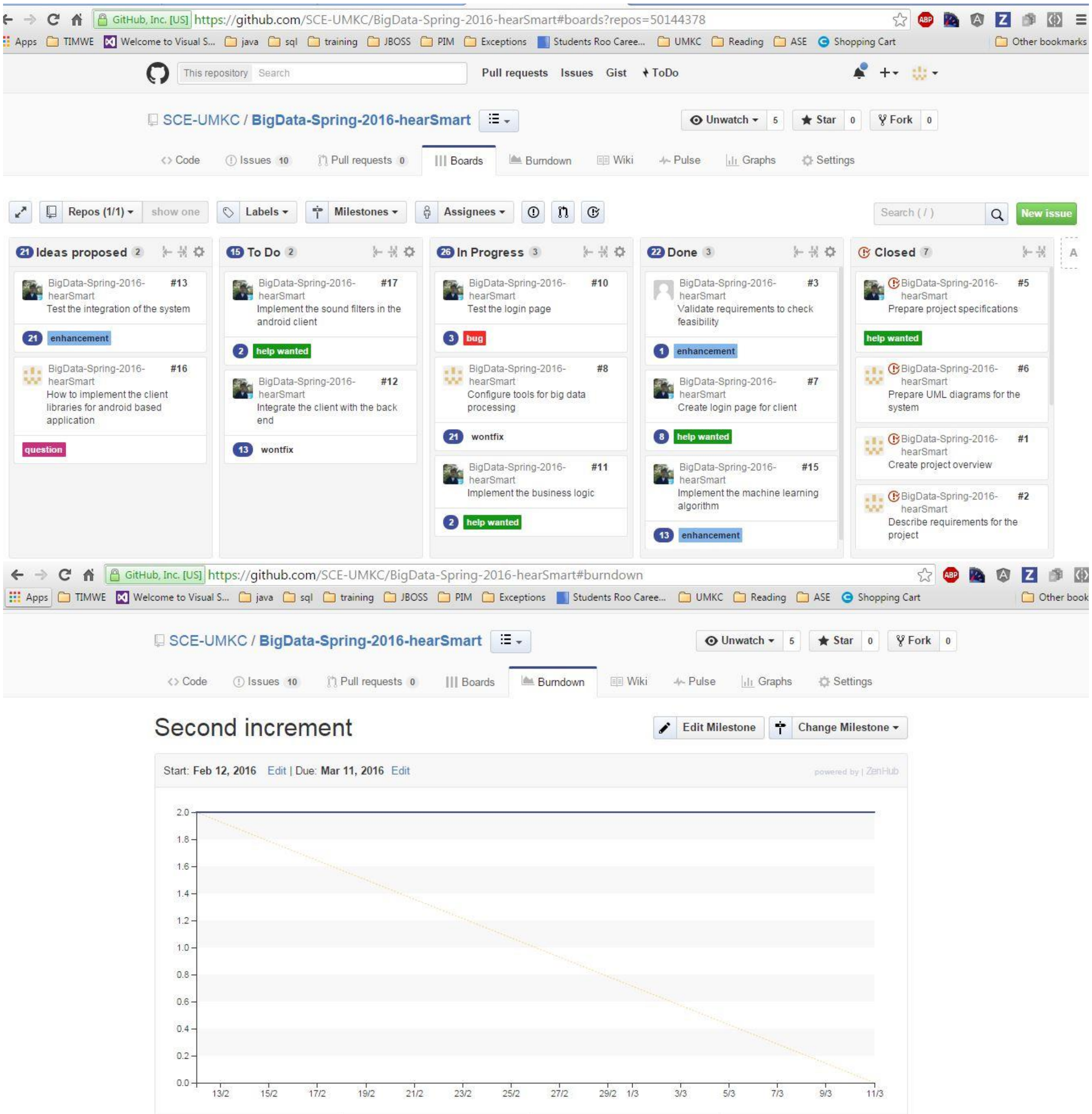


Figure 2: Timelines and commits incurred during second Increment.

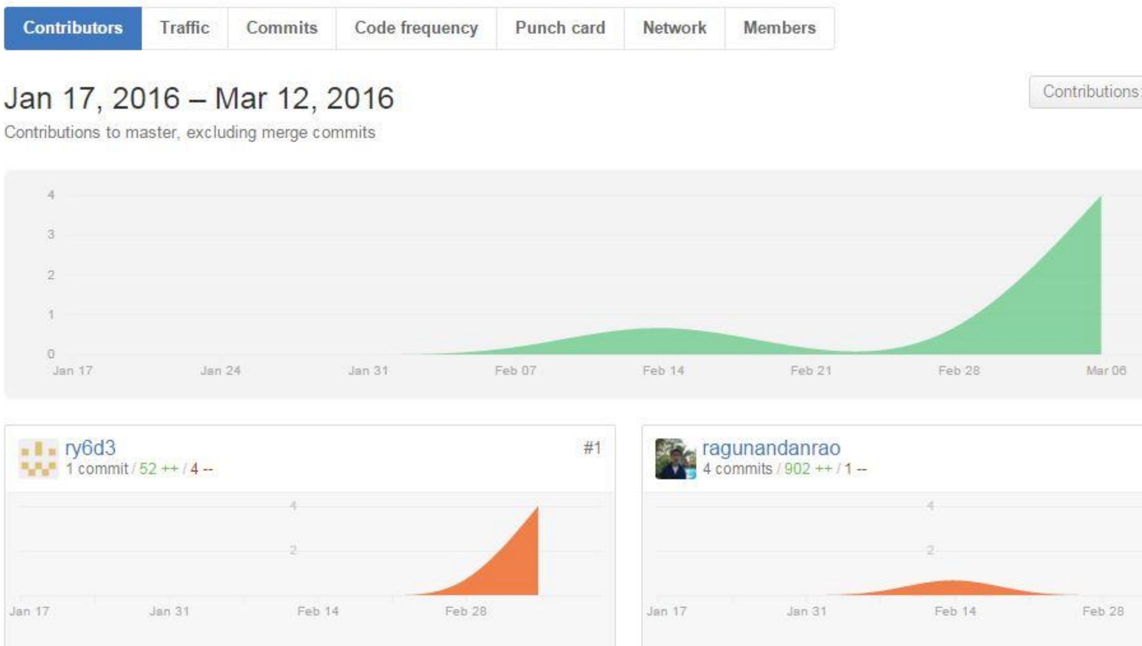


Figure 3: Depicting the contribution of each person to the repository Figure 2: Depicting the schedule and plan for phase 2 of the project.

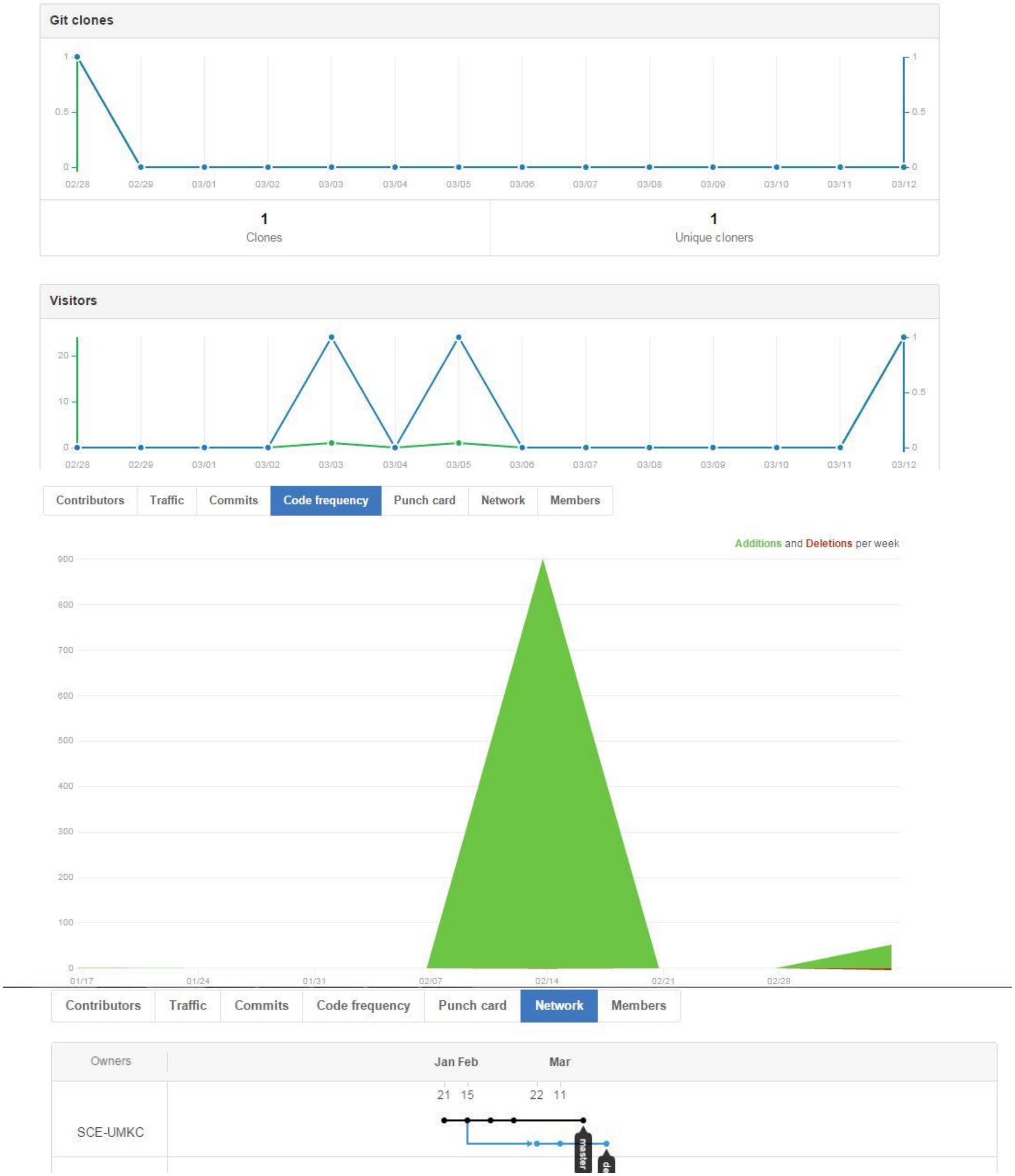


Figure 4: Depicting the analytical graph for the milestone of second increment.

VI. Second Increment Report

This document is a report of second iteration of work performed on the Smart Hear Mobile App. This App proposes to implement various web services into single App. The document emphasizes on the pictorial representation of the application using different implementations which gives an insight on internal system. This document intends to provide an overall description of the project named "Smart Hear".

The main outcome of the Second increment is the high and low level designs of the App. As of current state we have not deviated from our initial proposal that we submitted earlier. We have taken care of the implementation of Class Diagrams and Sequence Diagrams which showcase the flow of our application. The blueprint of the application is generated using the Wireframes and are available in this document.

During the course of the Second increment our main focus was on the idea and to get to know the feasibility of the various use cases provided in the proposal. We mainly concentrated on the finding the resources for the project in terms of papers on the topic and also in terms of the work that already done. Currently the implementation is very minimal and we believe that it can be caught up very soon. The team's immediate goal is to get a basic application that record sounds from the microphone of the smart phone and perform mutations such as frequency, pitch other feature modulations.

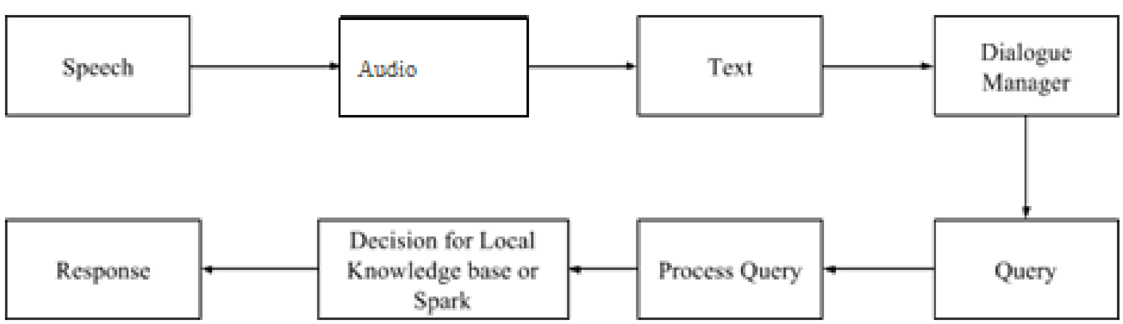


Figure 6

Figure 6 demonstrates a work flow diagram about how a user can generate responses from Speech.

Activity diagram of the system

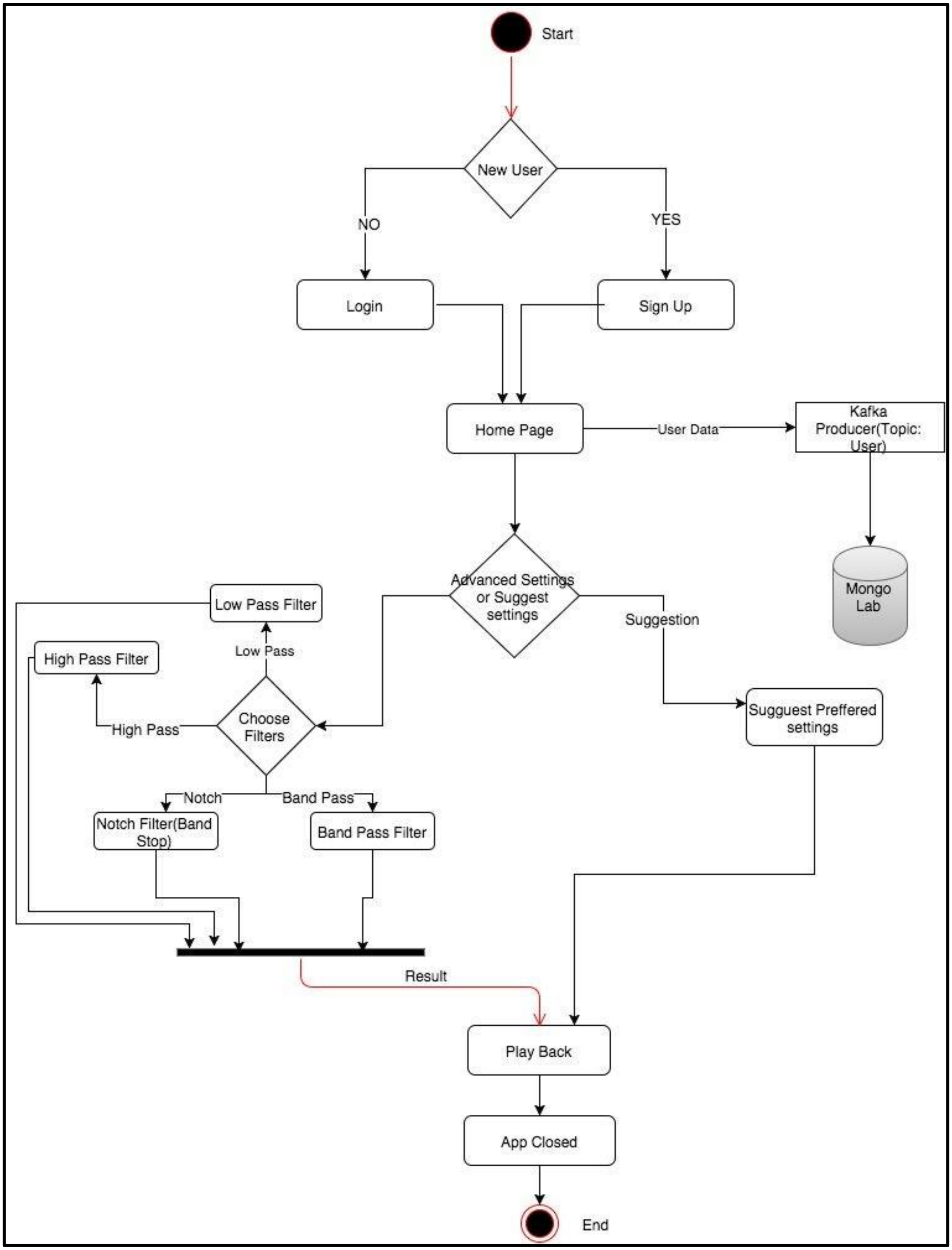
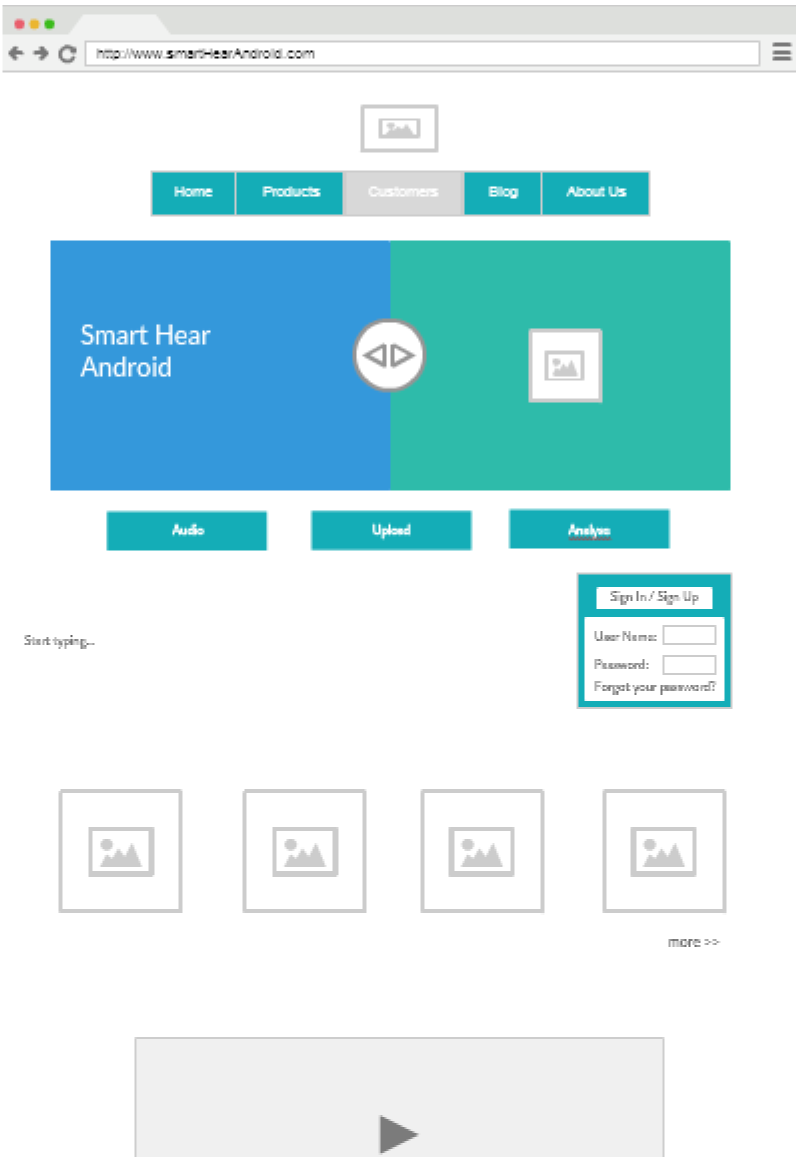
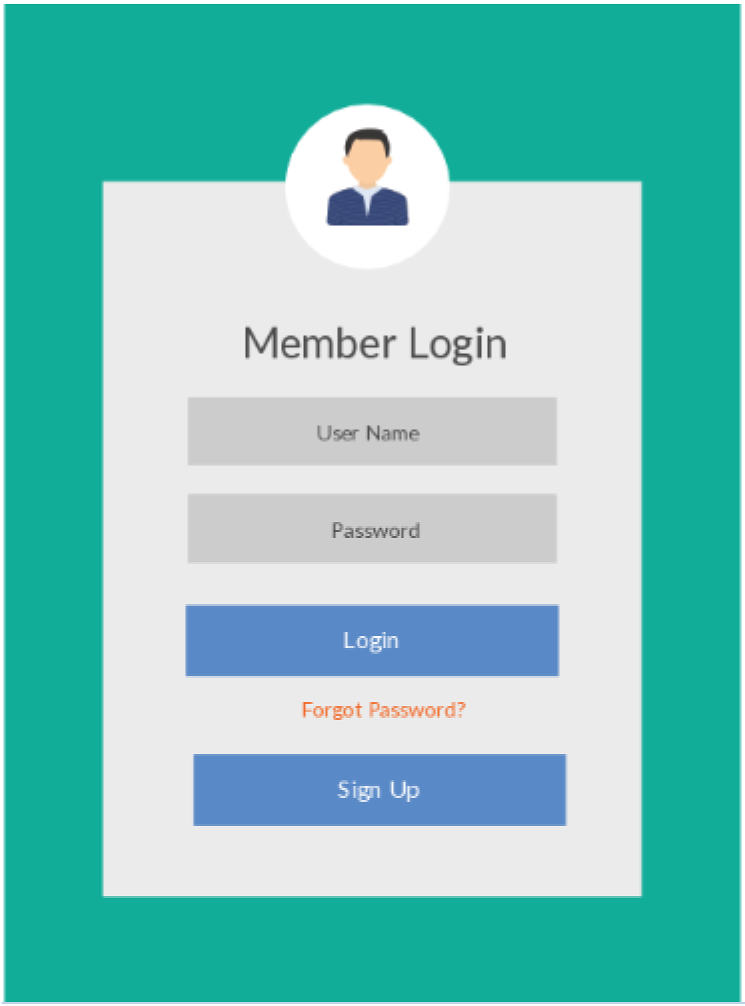


Figure 9

Figure 9 demonstrates a state diagram which explains High Level Design Architecture of the System.

Wireframes for the application




Member Login
User Name
Password
Login
Forgot Password?
Sign Up

Initial Screen for Application

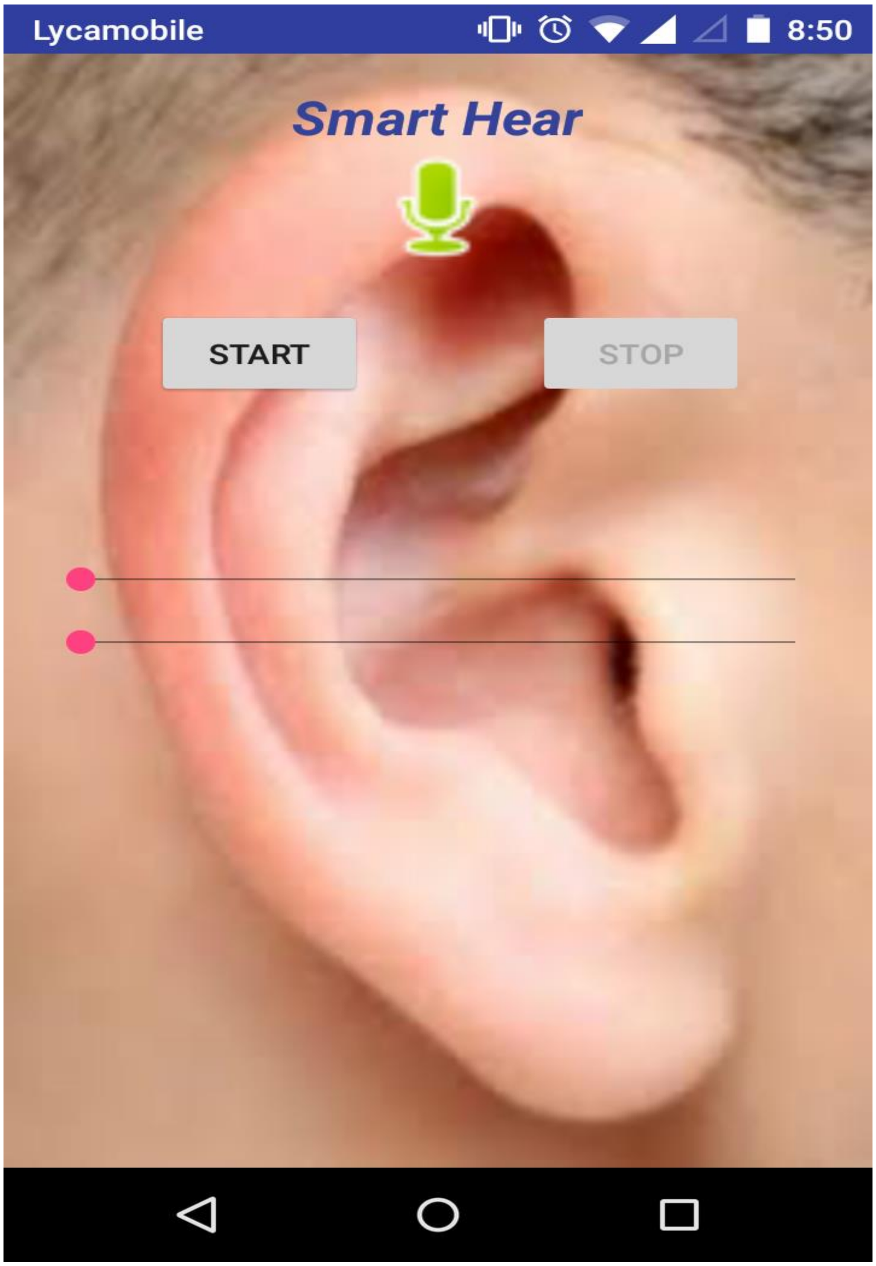


Figure: Home Screen of the application

Low Pass Filter Figure

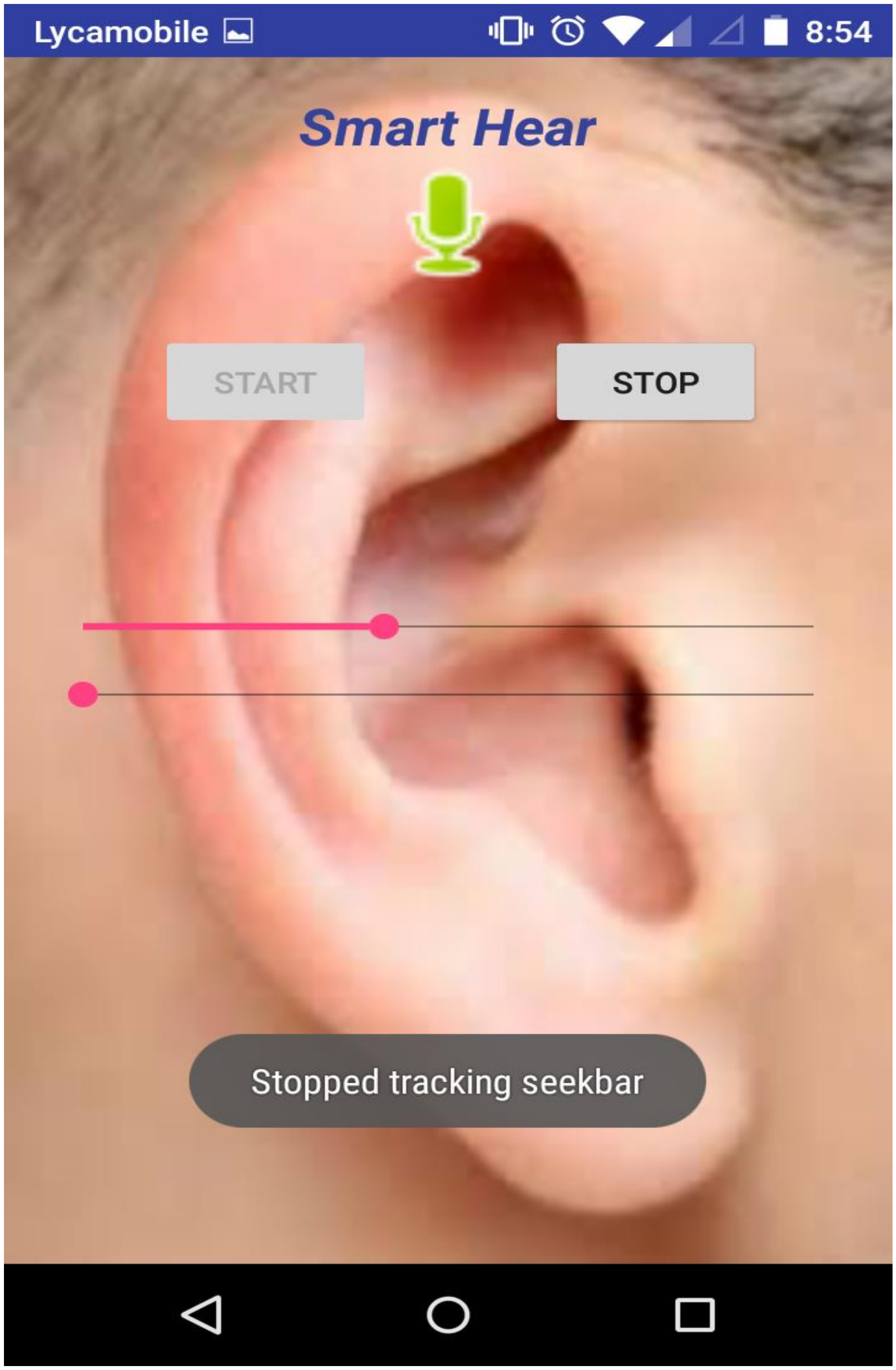


Figure: Low Pass Filter Figure

High Pass Filter Figure

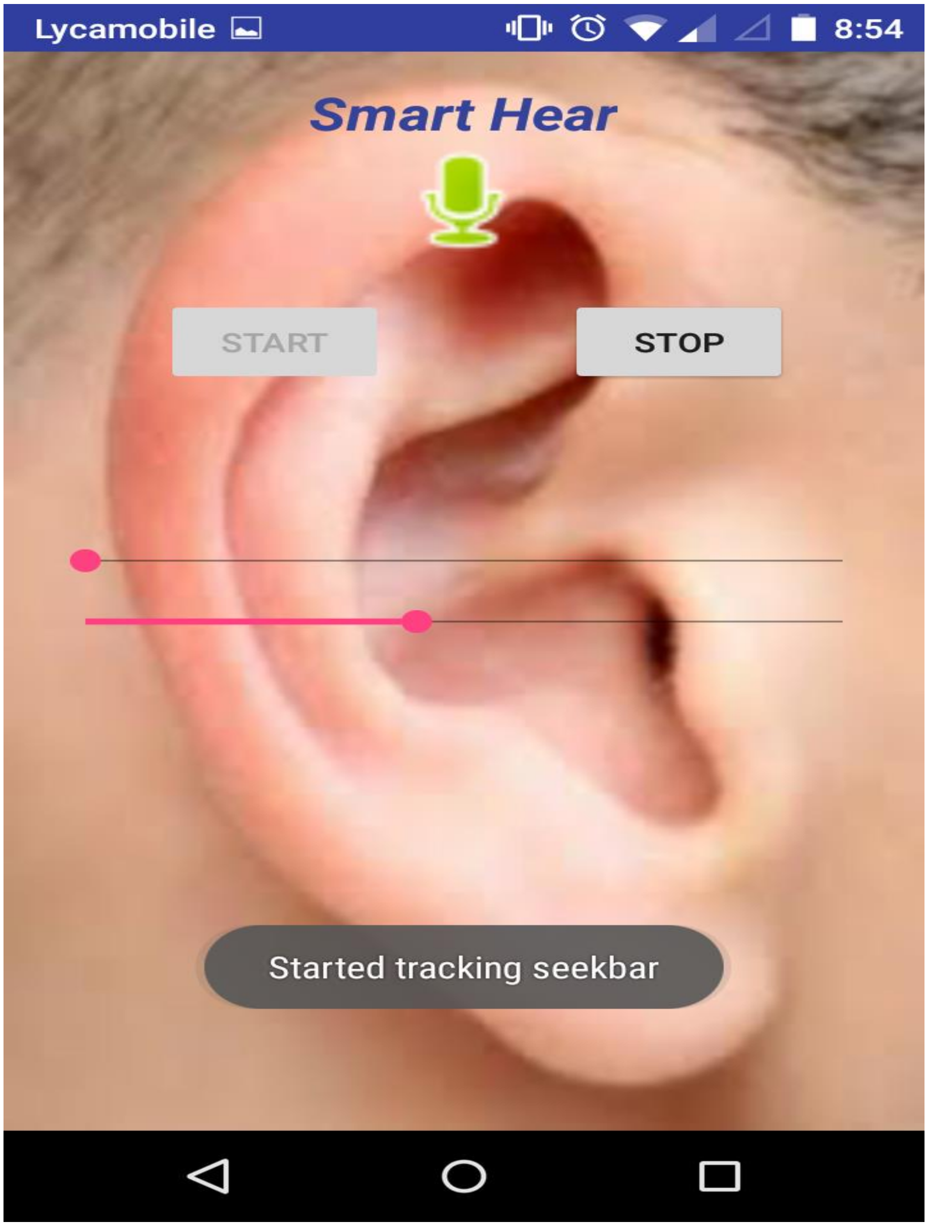


Figure: When High Pass Filter is passed

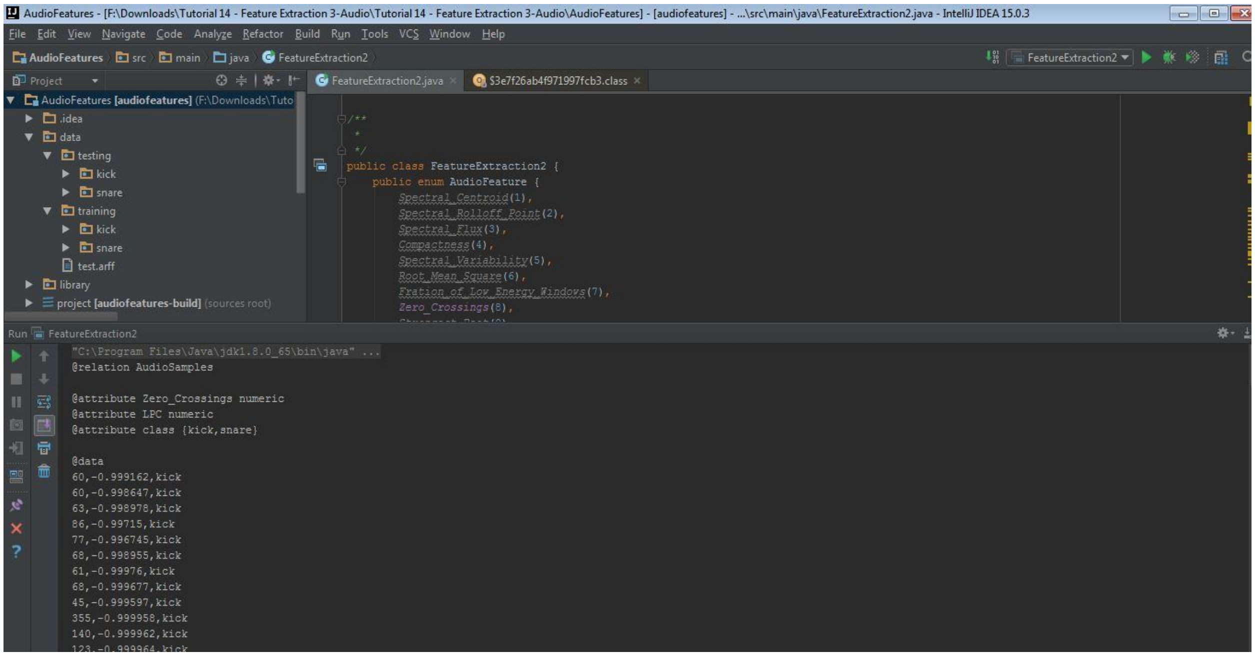


Figure: Results from Audio Filters Applied

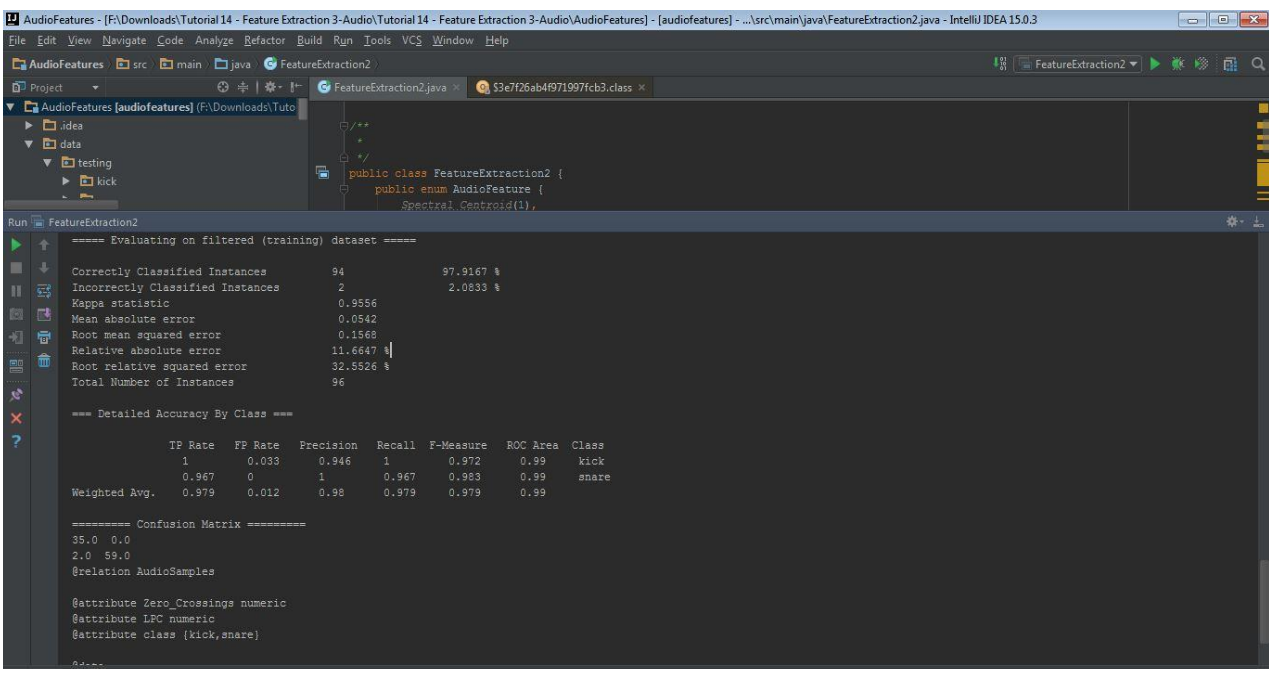


Figure: Using J Audio Feature Extraction

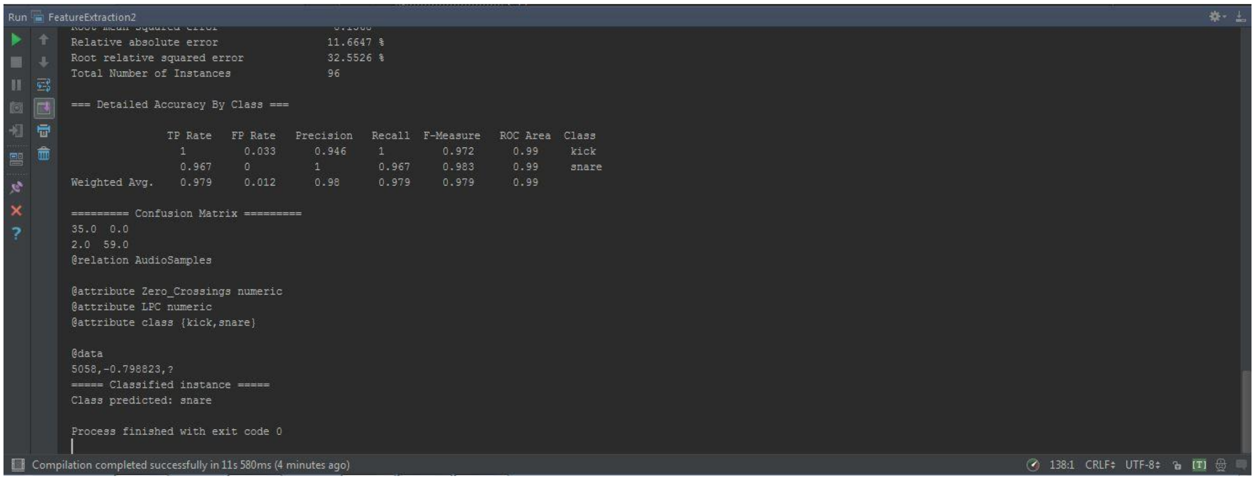


Figure: Outputs showing Confusion matrix

VII. Bibliography

- https://play.google.com/store/apps/details?id=mg.locations.track5&hl=en
- https://play.google.com/store/apps/details?id=com.fsp.android.friendlocator&hl=en
- http://www.raywenderlich.com/120177/beginning-android-development-tutorial-installing-android-studio
- http://developer.android.com/tools/building/building-studio.html
- https://usa.bestsoundtechnology.com/ces/#
- http://www.hearingreview.com/products/new-product-technology/
- http://www.resound.com/en-US/hearing-aids/resound-linx2#.Vse5ZPkrLcs

Smart Hear - Intelligent Hearing for Android

Project Third Iteration Report

Submitted on **06th April 2016**

Group 1:

1. **Ragunandan Rao Malangully (13)**
2. **Ravi Kiran Yadavalli (31)**

I. Introduction

This document intends to provide an overall description of the project named "Smart Hear" in detail. The project schedule and the plan of action is also discussed. The proposal document submitted would give an insight about what the project is about. The main outcome of the Third increment was the high and low level design of the application. As of current state we have not deviated from our initial proposal that we submitted earlier.

II. Project goal and objectives

Overall goal

The overall goal is to provide a hearing aid through the use of the smart phone that is accessible to every person today. The smart phone can be utilized in a way that it can act similar to the hearing dog that is what motivated us to take up this idea and try to implement using the smart phone.

This application can be used to provide benefits to the special abled people who face challenges in listening to sound. All the features come at the cost of installing an application that is freely provided to the user. No costs attached.

Objectives

- To provide a smart hearing aid to the user.
- To implement features to provide the user flexibility in varying certain key features of sound like frequency, tempo etc.
- To provide notifications to the user when there are important events like a door bell ring or an alarm that goes off.
- To develop a user friendly application.
- To ensure that we have a light weight application on the client end.
- To make sure that the context is recognized and the user settings are changed accordingly.
- Implement knowledge discovery to provide a summary of a topic that the user is listening or recording.
- Provide user customizable settings that the user can adjust to suit his or her needs.
- Ensure that the application can deliver its functionality at minimal cost of operation.
- Provide a user guide to understand and use the application.

III. Project background and related work

The app is not a new invention or a brand new concept. But the implementation aspect of it is what makes it unique from the others. The following are the projects that are similar to what our idea is.

- Resound LiNX: It is a smart hearing aid kit that can integrate with Apple Iphone and other android devices. The device does noise cancellation and also amplifies the sound that the user is listening to. The device can also integrate with smart watches and provide very handy functionality. But the downfall is that it is an additional device that has to be purchased and is does not have notification features to the user.
- Siemens Hearing Aids: These were introduced at the International CES 2015. They are pretty close to what an interactive hearing aid can deliver and seamlessly work hand in hand with smart devices. They require a specific app to connect the device with the hand held devices and only then can they provide the functionality. On the flipside the product is priced high and is not for everyone to afford it.

Significance

The major significance of the application lies in the fact that the all the useful service s are provided under one system. There is no need of another device for the hearing enhancements. The user can access all the features through his smart phone. Even form the implementation aspect the usage of big data framework would ensure that the application can process the data at high performance rates. This would also reduce the cost of the creating and operating the application. User customizable settings is one of the major significance of the project.

IV. Proposed system

Requirements specifications

- The application will provide users facility to login with their Google plus account.
- User should have the flexibility to login with his Facebook account.
- The system will display the user his information along with his profile picture.
- An Android application that serves as the smart hearing aid and also provides a user friendly model for the feature extraction.
- A machine learning framework that can provide a model based on the features received from the client end.
- Analysis of audio files to observe their characteristics.
- Collection of features present in an audio file and weighing the features to select the vital features dynamically.
- To send text notifications on various events based on the pattern of previous audios.
- To be able to detect and send alarms on to the device in potential hazardous situations.
- Integration of the client and the processing framework for a complete application that can perform as a smart hearing device.

Workflow analysis

The first and the most fundamental workflow of the application is that of the designing the UI for the system. This involves selecting the proper HTML elements and arranging the elements in an appropriate manner. It includes making use of the CSS attributes to style the UI of the application. Each page of the application is to be custom designed for suiting the needs of the service or feature being provided over that page. This would also include capturing the sound from the device and storing it in the device storage.

Second task would be do perform basic filtering and other alterations to the sound using libraries such as high pass, band pass etc. The frequency, tempo, pitch and other features of the sound can be altered and extracted individually.

The third workflow deals with the performing model discovery or applying the best fit model over the features extracted. This would involve applying machine learning algorithms like Decision tree etc. to classify the source of the sound.

Technological and architectural requirements

On the technological aspect the project is pretty much dependent on the native Android libraries for implementing the client functionality. This would also include some external Java or Android based libraries or plugins for managing the operations on the sound recorder. The primary development language for the client would Java and XML for the UI part.

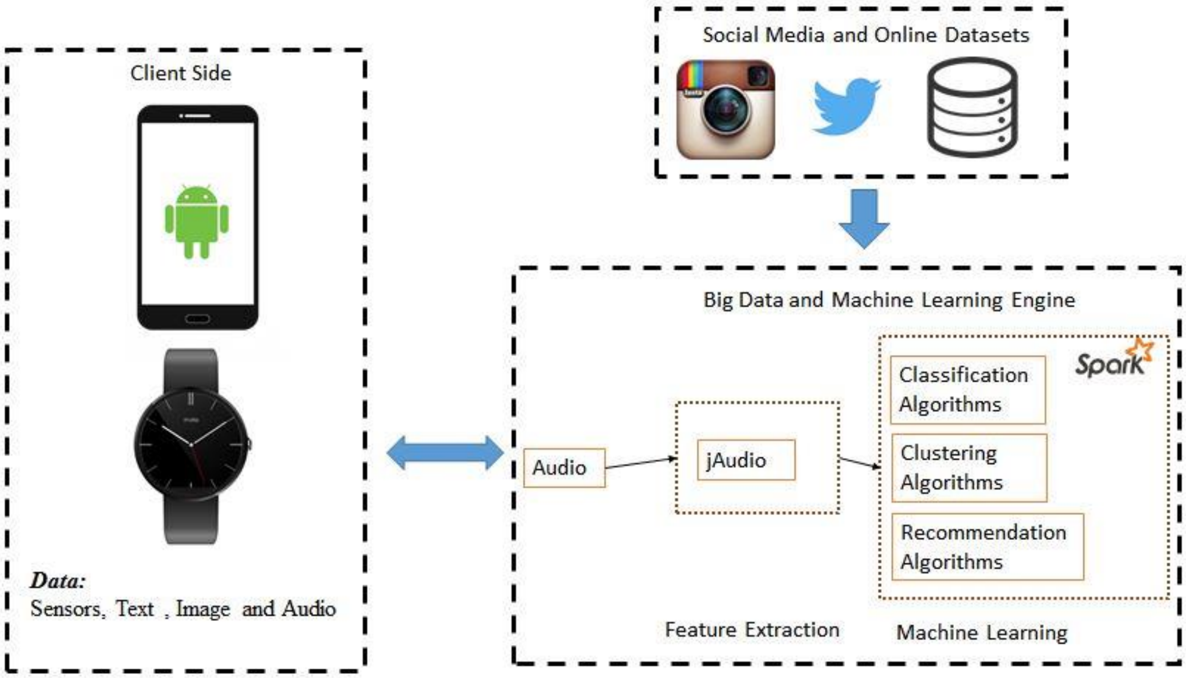
On the server end it would be mostly Scala that would be running on Spark to integrate with the client to perform the machine learning aspect of the application. At the current stage the database that would be in use would MongoDB.

In terms of the architectural framework we are using a traditional 3 layer architecture but the interaction between the layers may vary depending on the situation. In terms of the traditional scenario the recording is present in the database and then the spark engine processes the data to categorize the audio data. In the real time recording scenario the UI layer directly interacts with the spark engine to process the data as it records. The framework would ensure this flexibility is provided to the user.

Framework specifications

The system would be a flexible framework that can incorporate the services that are customized to suit the user and application needs. The system architecture would comprise of a client server architecture in which our application would be a client accessing service hosted or offered through different servers. Internet would be the medium of communication between the application and the services hosted in the web. One of the service that the application offers requires the traditional 3 tier architecture where in the data is stored through the UI layer into the business layer and ends up in the database. The system architecture is represented in the diagram below.

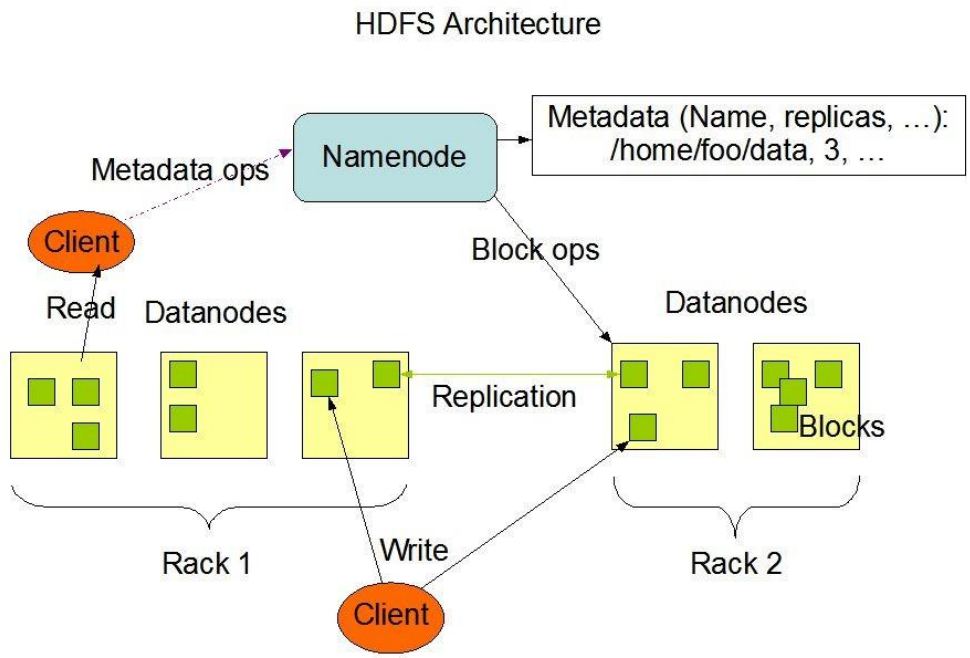
Architecture Diagram:

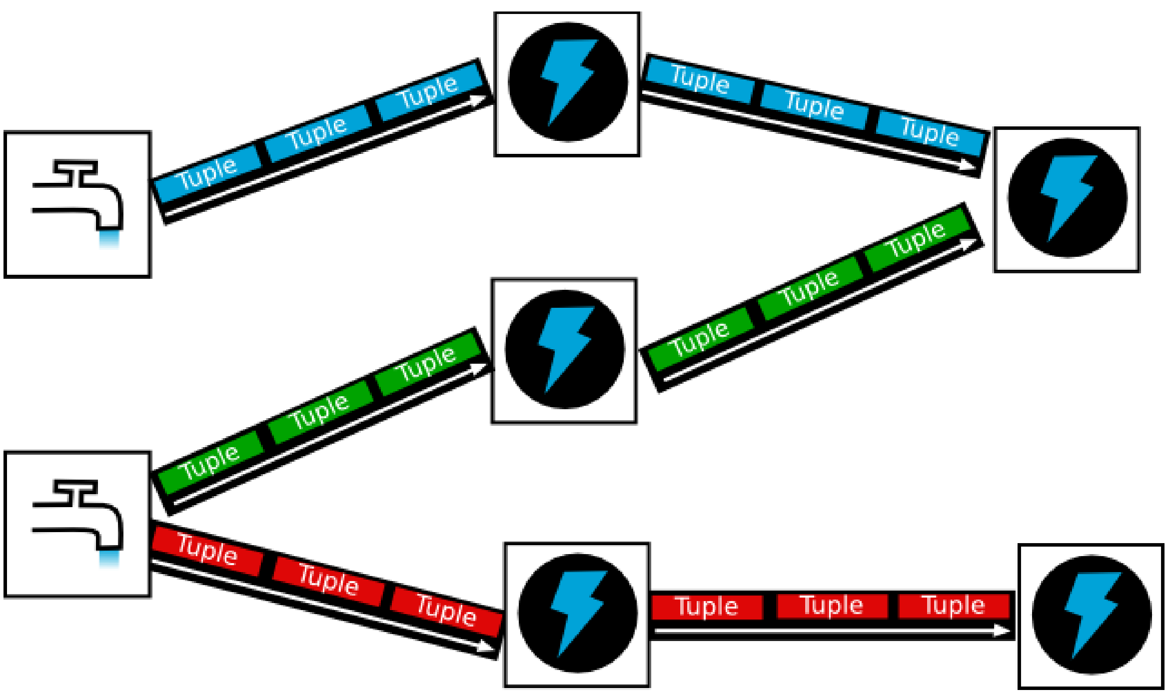


System specification

Apache Hadoop

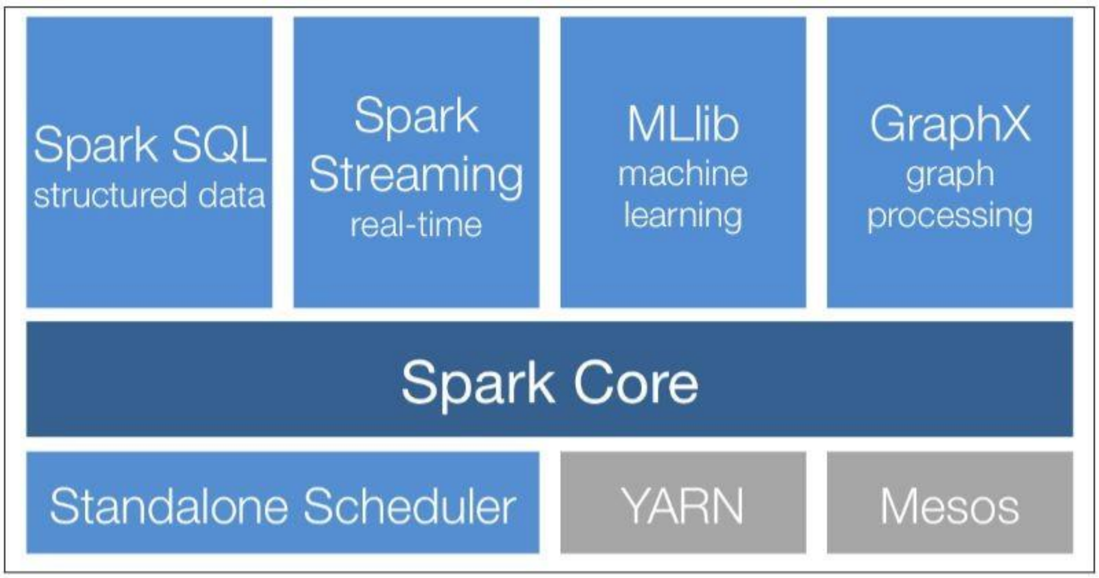
Apache Hadoop is open source framework that processes large amount of distributed data sets among the cluster of nodes using the programming models. Hadoop is developed to be scalable, reliable and distributed computing. It is designed to scale from single server to a cluster of nodes where each node offers local computations and storage. It has two parts namely HDFS and MapReduce. It uses HDFS to store the data where it breaks the input data and distributes to nodes, this enables the parallel processing of data. The MapReduce distributes the programs across the nodes where they perform operations on the data stored in HDFS. Hadoop has YARN which helps in cluster resource management and job scheduling.


HDFS Architecture
Metadata ops
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Rack 2
Client


Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple
Tuple

3.3.3. Apache Spark

Apache Spark is a cluster computing platform which has been intended to be fast. Spark extends MapReduce to support more types of computations, Interactive queries and stream data processing. It provides high level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for structured data processing and using SQL and Apache Hive.



3.3.4. MongoDB

MongoDB is an open source document based database which stores the data in JSON format. MongoDB is classified as a NOSQL database. It can be used as a file system by using the concept of load balancing and replications.

After processing the audio signal the classification of the signal which obtained is stored in the MongoDB. We store the classification in three collections named

1. Users
2. Locations and
3. Settings

It stores the audio files, users, user ids, tags, latitude, longitude, time in the collections.

3.3.5. R (RStudio)

R is one of the famous data analytics tool available and being used by researchers from many decades. R is well known as a Software environment as well as a programming language for graphics and statistical computing used majorly for data mining by data miners and statisticians.

Statistical and Graphical techniques such as Linear and Nonlinear modeling, clustering, classification and others are implemented using R and its libraries. When compared to most of the statistical computing languages, R has much stronger Object Oriented Programming facilities. Static graphs is the other major strength of R.

RStudio is the open source IDE which supports R programming language and it is available for free in two editions. One is the RStudio Desktop edition which is used to run desktop applications locally. The other is the RStudio Server which uses web browsers to access RStudio running on a remote Linux server.

V. Project plan

The proposed project plan is outlined the by the screen shot from the Kanban tool. The project is divided into four phases each of which is guide lined by a test driven approach. Each iteration has four tasks categorized by stories of designing phase, building the service through implementation and testing. The final task is to store the completed tasks for future integration.

Phase 2:
This phase mainly deals with the designing the system for the implementation phase. The tasks of this phase mainly focuses on the UML diagram and collecting the necessary information for the realization and the implementation of the system. The end result of the phase 1 are initial screens of the application. These are just the rudimentary level implementation screens and these may change during application development process.

Timelines of the project

The timelines of the project schedule are in coherence with the timelines of submission for each of the phase. As of the current state there are no deviations from the project schedule.

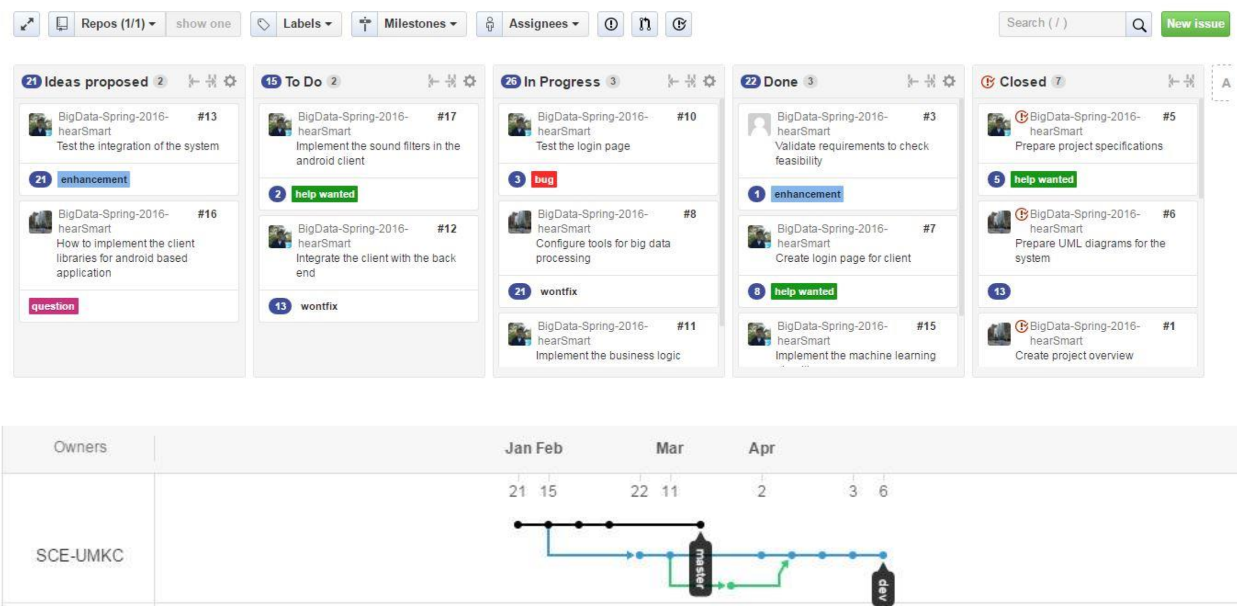


Figure 2: Timelines and commits incurred during Third Increment.

Figure 3: Depicting the contribution of each person to the repository

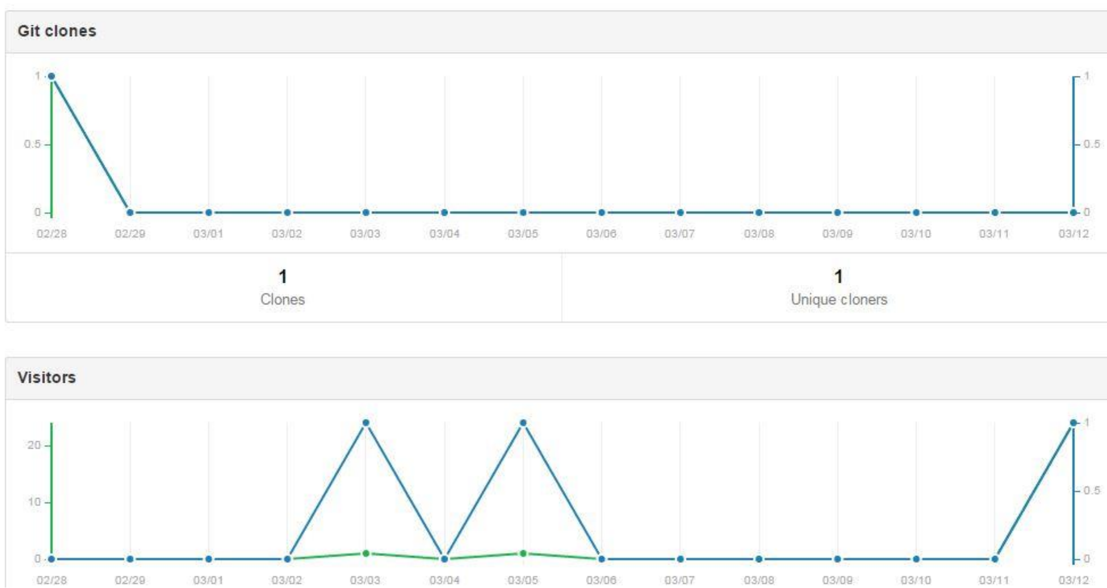


Figure 4: Depicting the schedule and plan for phase 3 of the project.

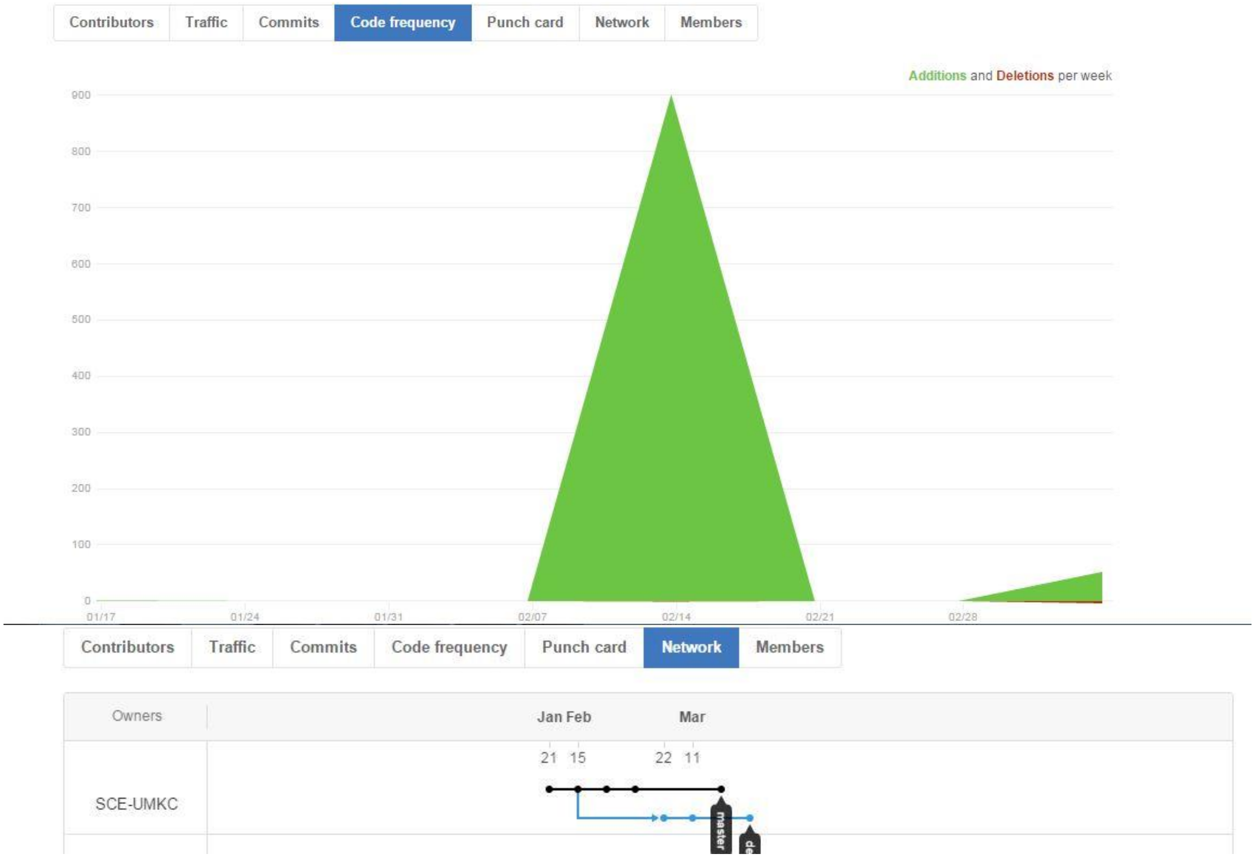


Figure 5: Depicting the analytical graph for the milestone of Third increment.

VI. Third Increment Report

This document is a report of Third iteration of work performed on the Smart Hear Mobile App. This App proposes to implement various web services into single App. The document emphasizes on the pictorial representation of the application using different implementations which gives an insight on internal system. This document intends to provide an overall description of the project named “Smart Hear”.

The main outcome of the Third increment is the high and low level designs of the App. As of current state we have not deviated from our initial proposal that we submitted earlier. We have taken care of the implementation of Class Diagrams and Sequence Diagrams which showcase the flow of our application. The blueprint of the application is generated using the Wireframes and are available in this document.

During the course of the Third increment our main focus was on the idea and to get to know the feasibility of the various use cases provided in the proposal. We mainly concentrated on the finding the resources for the project in terms of papers on the topic and also in terms of the work that already done. Currently the implementation is very minimal and we believe that it can be caught up very soon. The team's immediate goal is to get a basic application that record sounds from the microphone of the smart phone and perform mutations such as frequency, pitch other feature modulations.

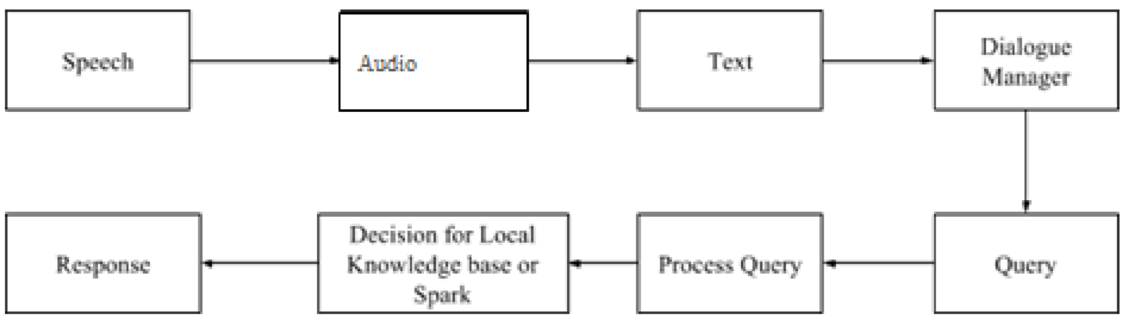


Figure 6

Figure 6 demonstrates a work flow diagram about how a user can generate responses from Speech.

Activity diagram of the system

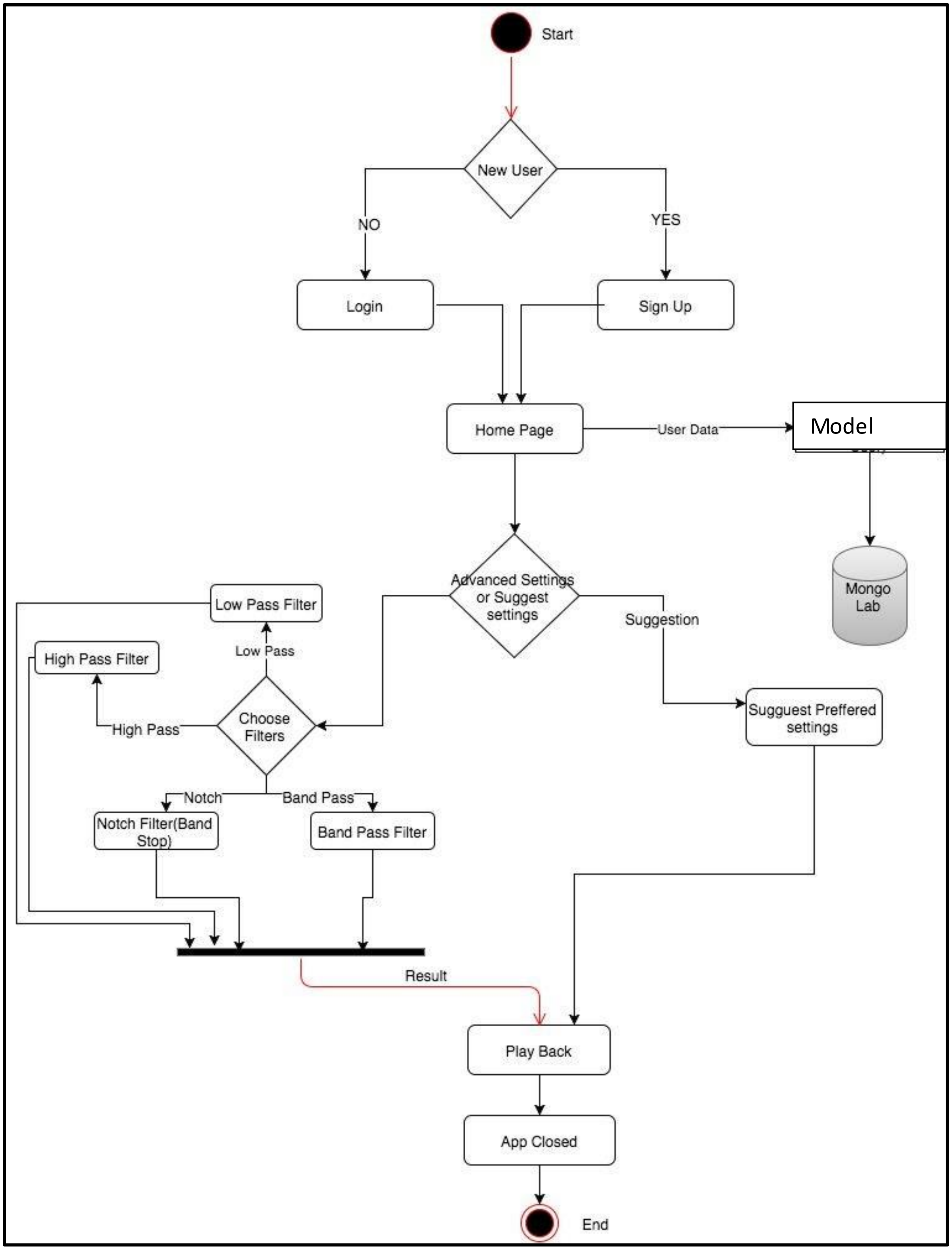
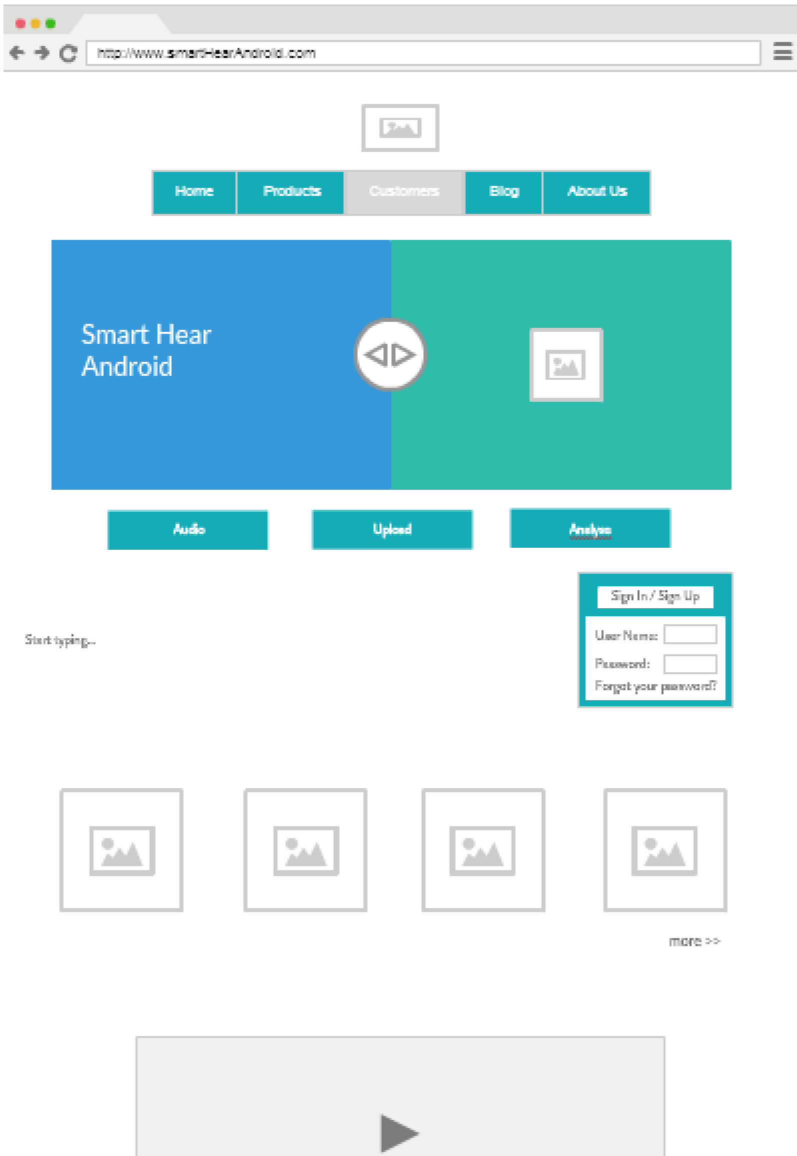
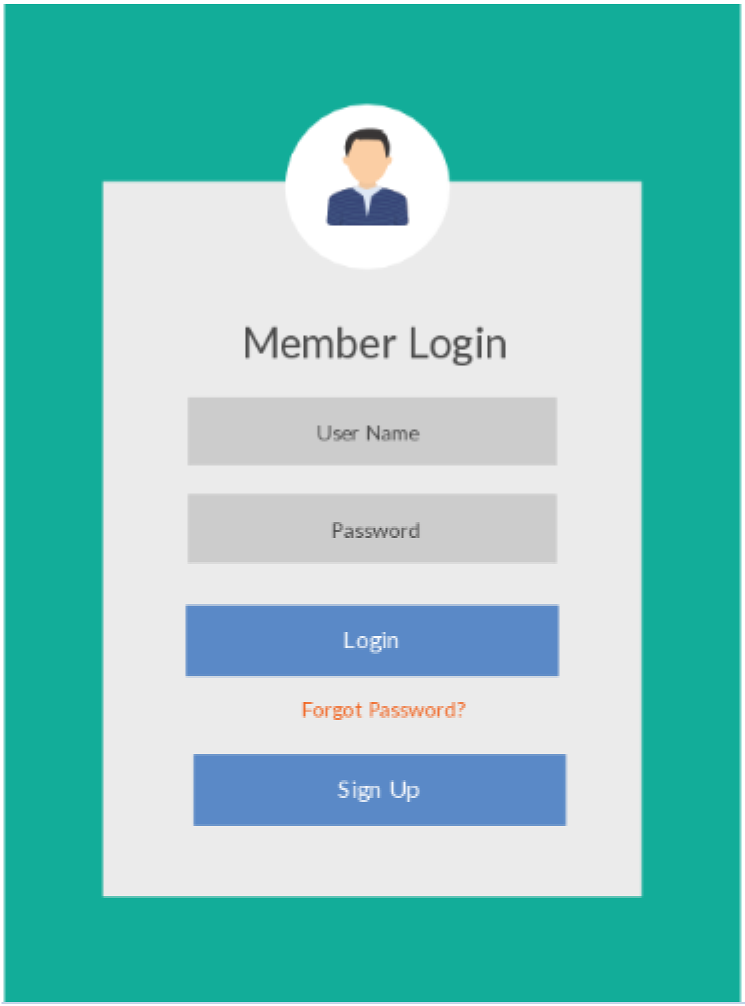


Figure 9

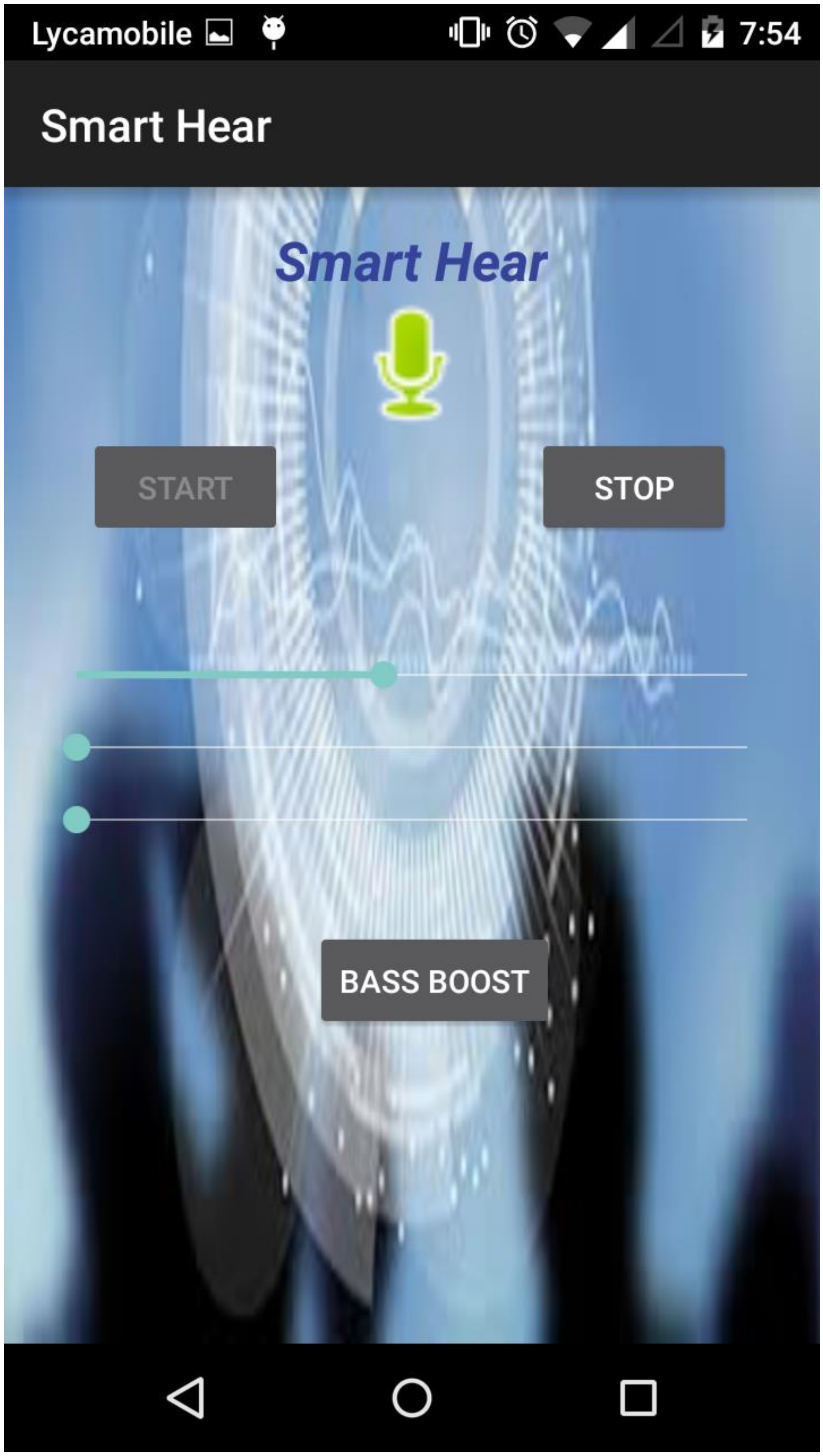
Figure 9 demonstrates a state diagram which explains High Level Design Architecture of the System.

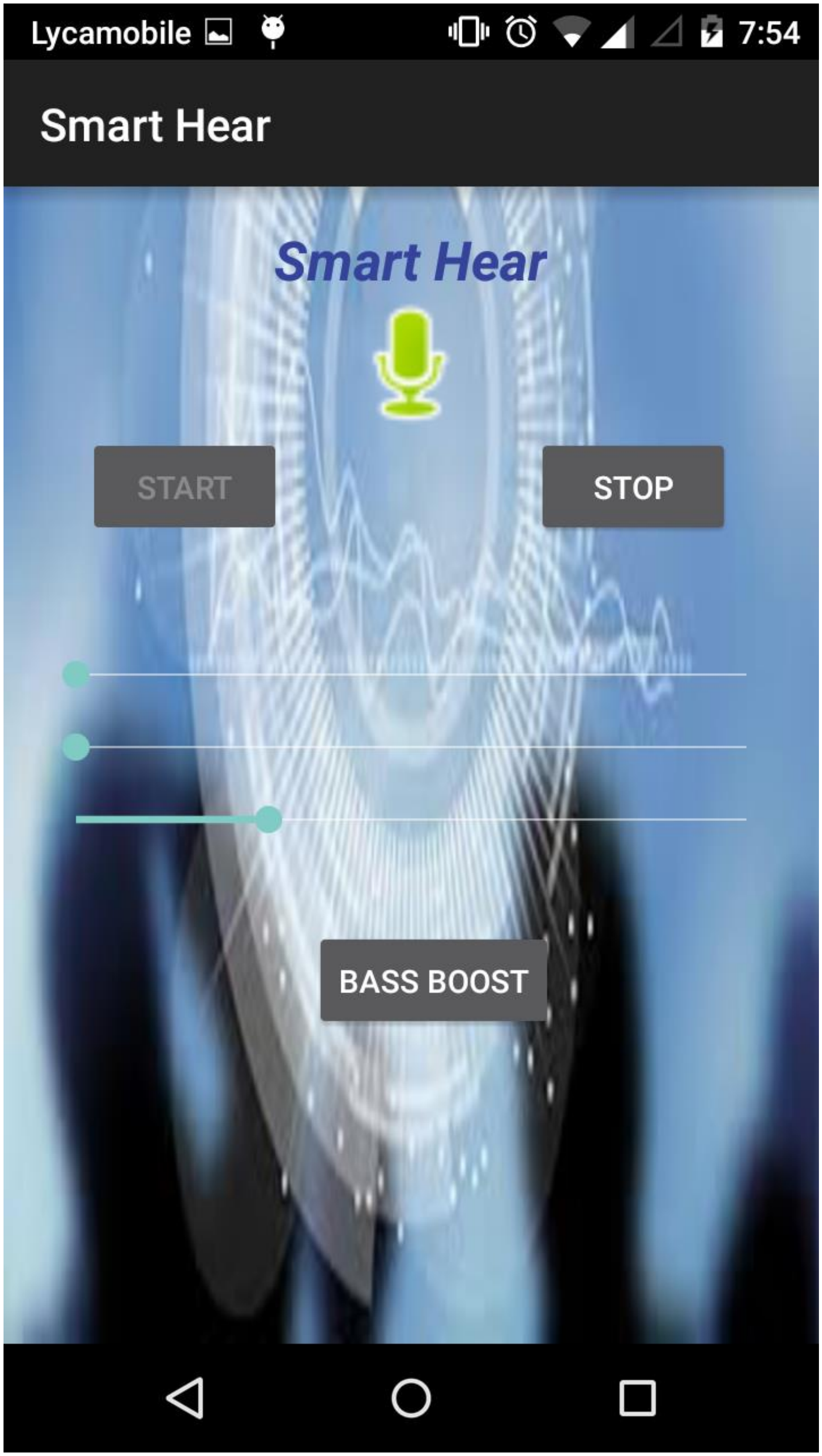
Wireframes for the application




Member Login
User Name
Password
Login
Forgot Password?
Sign Up

Initial Screen for Application




Lycamobile
7:54
Smart Hear
Smart Hear
START
STOP
BASS BOOST

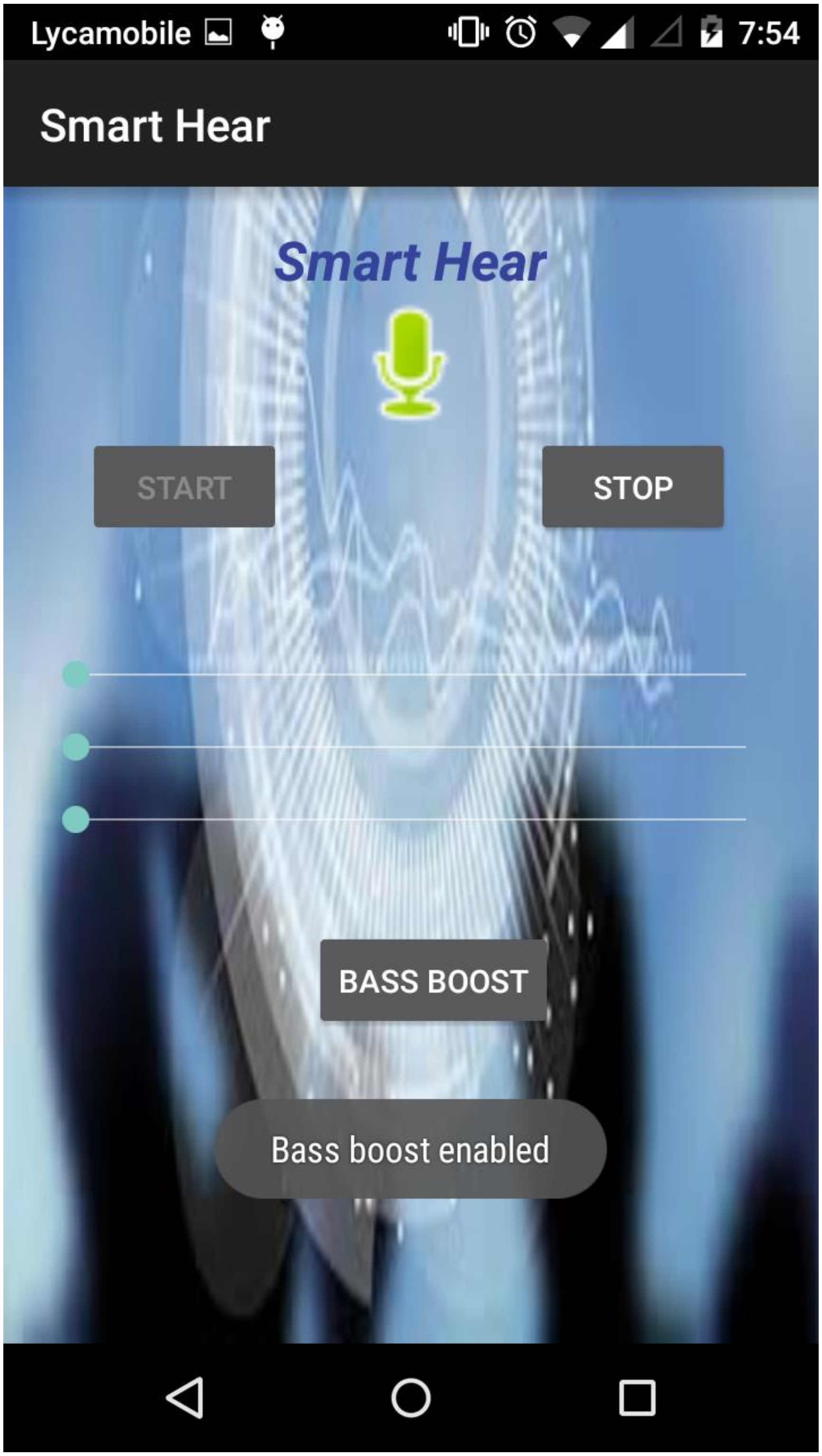


Figure: Results from Audio Filters Applied

VII. Bibliography

- https://play.google.com/store/apps/details?id=mg.locations.track5&hl=en
- https://play.google.com/store/apps/details?id=com.fsp.android.friendlocator&hl=en
- http://www.raywenderlich.com/120177/beginning-android-development-tutorial-installing-android-studio
- http://developer.android.com/tools/building/building-studio.html
- https://usa.bestsoundtechnology.com/ces/#
- http://www.hearingreview.com/products/new-product-technology/
- http://www.resound.com/en-US/hearing-aids/resound-linx2#.Vse5ZPkrLcs

Smart Hear - Intelligent Hearing for Android

Project Fourth Iteration Report

Submitted on **29th April 2016**

Group 1:

1. **Ragunandan Rao Malangully (13)** 2. **Ravi Kiran Yadavalli (31)**

I. Fourth Increment Report

This increment mainly focused on extracting the features at the client end using some real time audio feature extraction library. We had experimented using various audio libraries such as TarsosDSP, musicg and Jaudio. Based on the results and the ease of implementation we decided to use Jaudio feature extraction library for both the client and the spark server engine. This required us to extract the features based on raw bytes of audio data. This was one of the phases of work being progressed. On the other hand we had to build the model by training it using audio data in .wav format. The library used to extract features here was Jaudio. We initially put effort into using Naive Bayesian algorithm which yielded pretty low results in terms of accuracy, approximately 28 percent. Then we decide to train the model using the Random Forest algorithm which improved the accuracy to 72 percent. This accuracy was measured by providing a subset of training data as testing data and then calculating the predicted versus the actual class to compute the accuracy. Once trained the model was saved for future use and reduction in time of execution in prediction.

In the later part of this period we also introduced various preset effects that simulate the hearing experience of home, office, outdoor, classroom and finally bass effect. This is being provided as a drop down for the user to choose from. Based on the option selected by the user we vary the frequency filter values to suit the context that the user has selected. This is a manual implementation of learning the context by asking the user to choose his context. Then on the client end we also provided the feature where there would be a service continually detecting sound. On detection of sound above a cut-off amplitude it would open the application. The user can then click on record button to analyze the sound. This decision is left to the user. Once the app is minimized the service resumes its job of detecting sound. This was done to ensure that we do not run the service all the time conserving the battery power. The implementation was done using a background service in android and we used the MediaRecorder class in android to detect any sound generated.

The next part of work was to create a socket between the android client and the spark server engine. This socket was used to share the features extracted on the client side with the spark server. Once the features were received by the server it would process the features using saved model to predict a class that best matches the model. This was sent back to the client using another socket. Once the client received the prediction result we displayed a notification to the user with what was happening and also provided a vibration effect to alert the user. Also we had some improvements in terms of the look and feel of the application. We have changed the background of the app and also provided an icon for the app launcher.

We have provided the screenshots of the application that is a product of this increment and the combined efforts of previous increments.

II. Screenshots for the increment

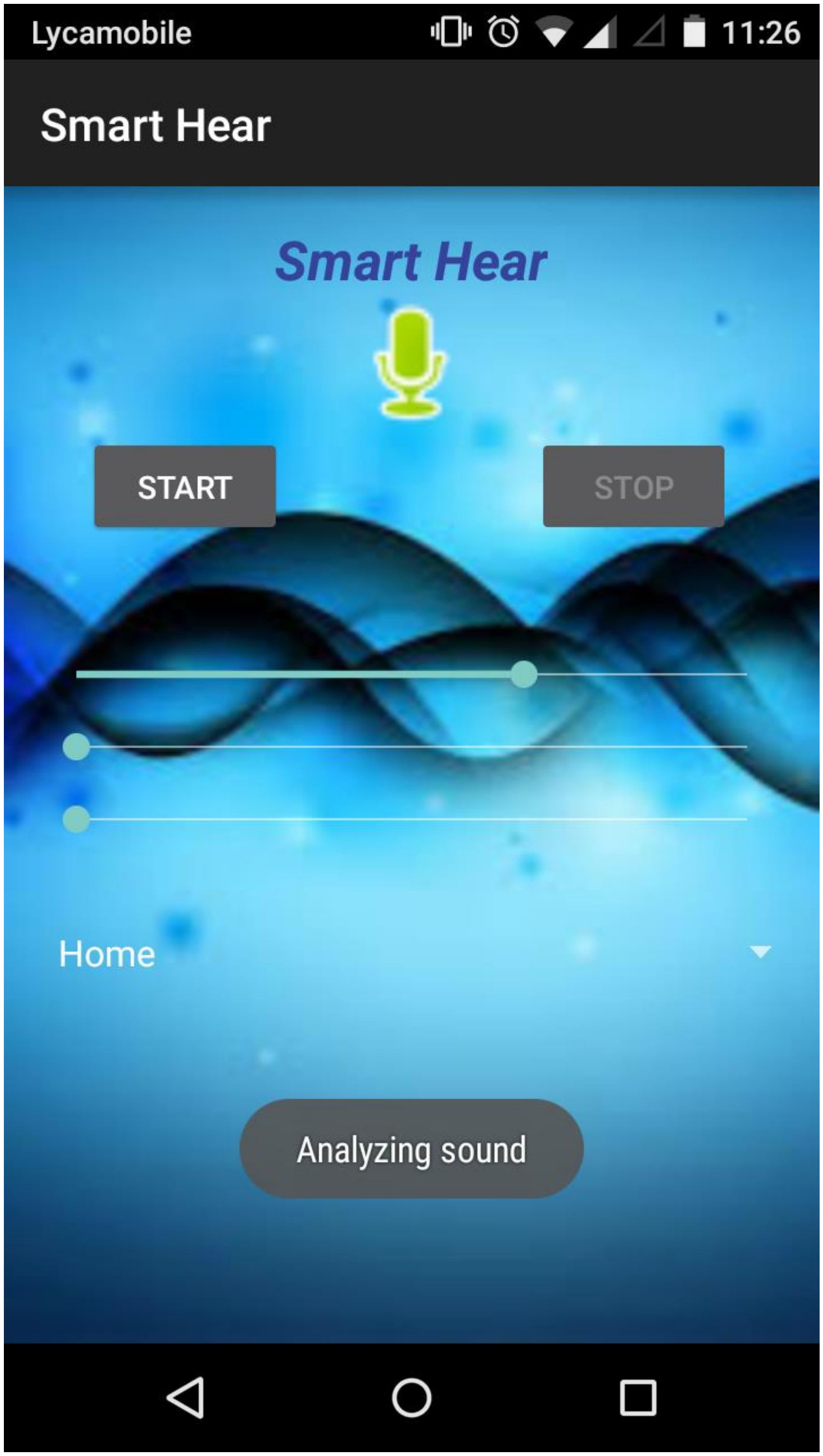


Figure 1 : Initial screen of the application after a sound is detected.

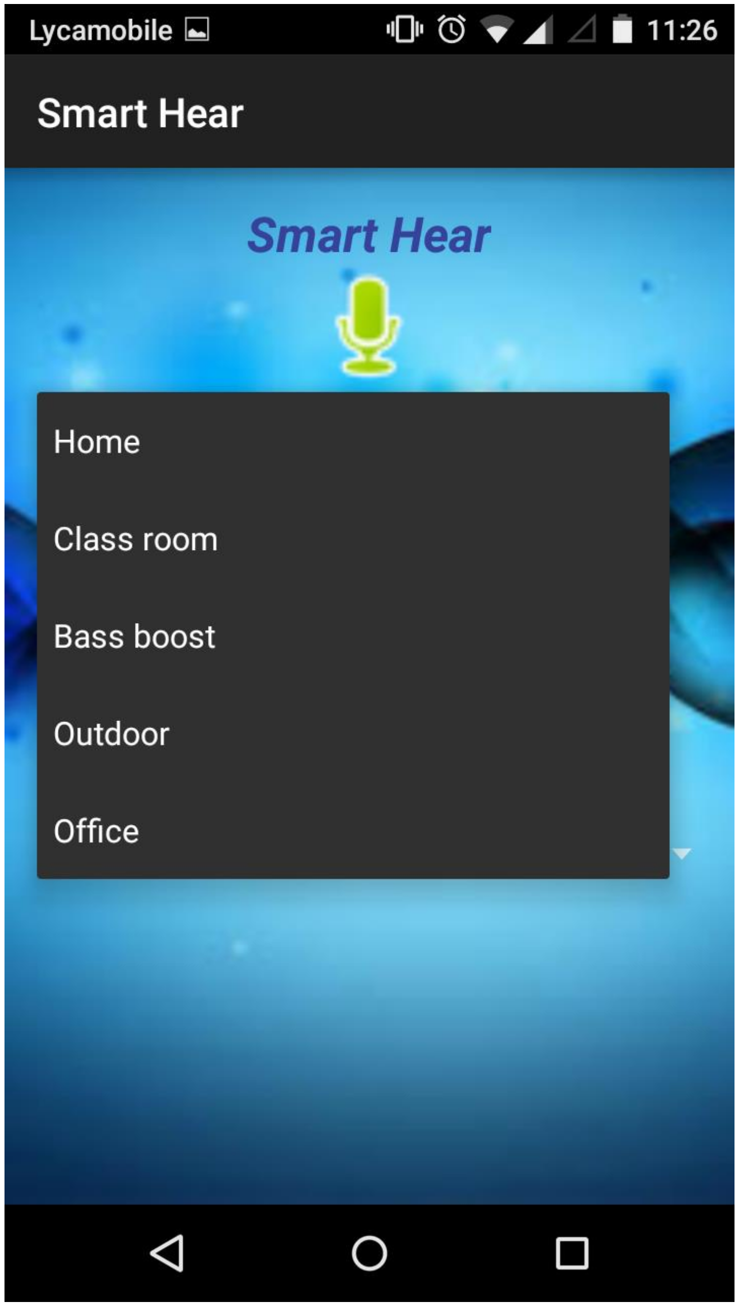


Figure 2: The screen with the different context options for the user to pick from.

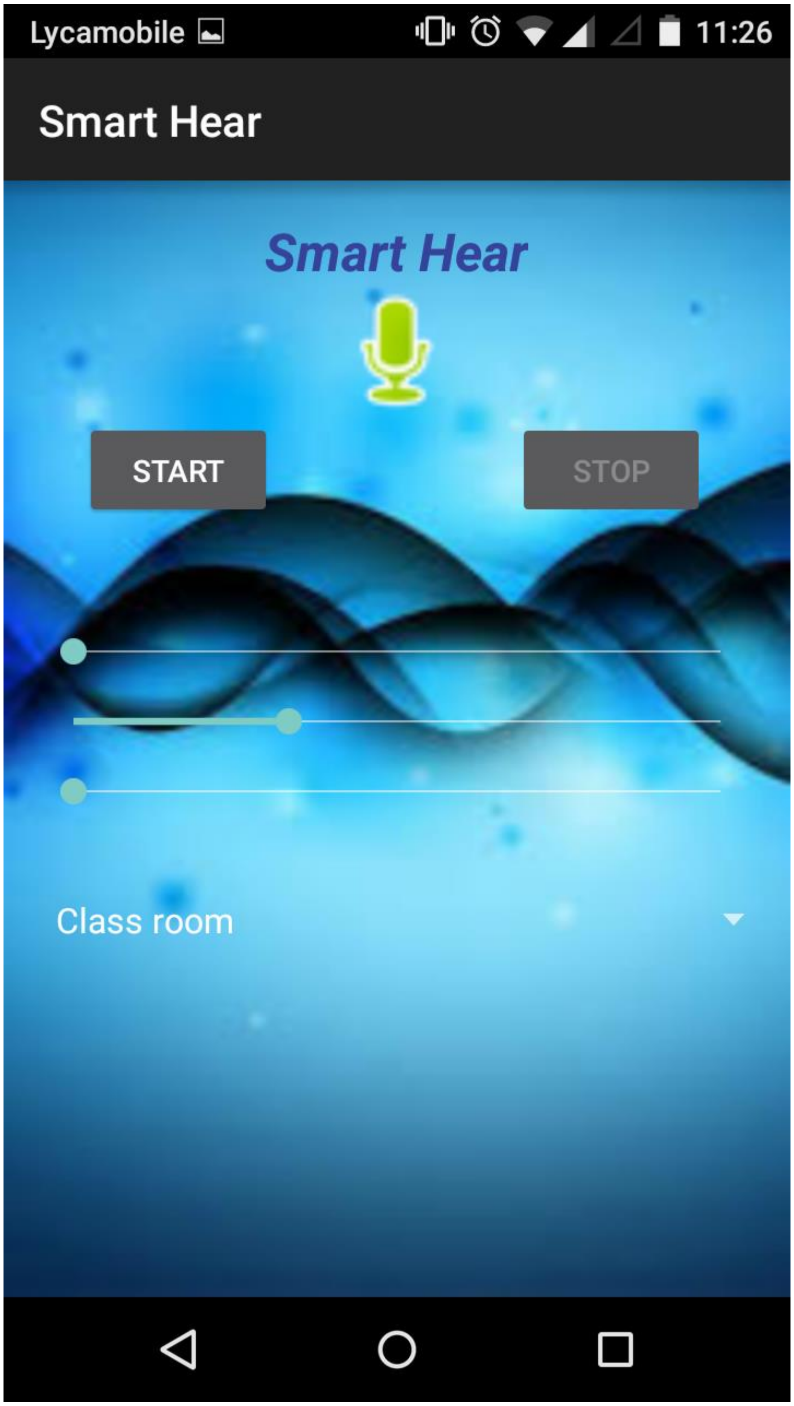


Figure 3: One of the sample screen showing the settings for one of the context selected.

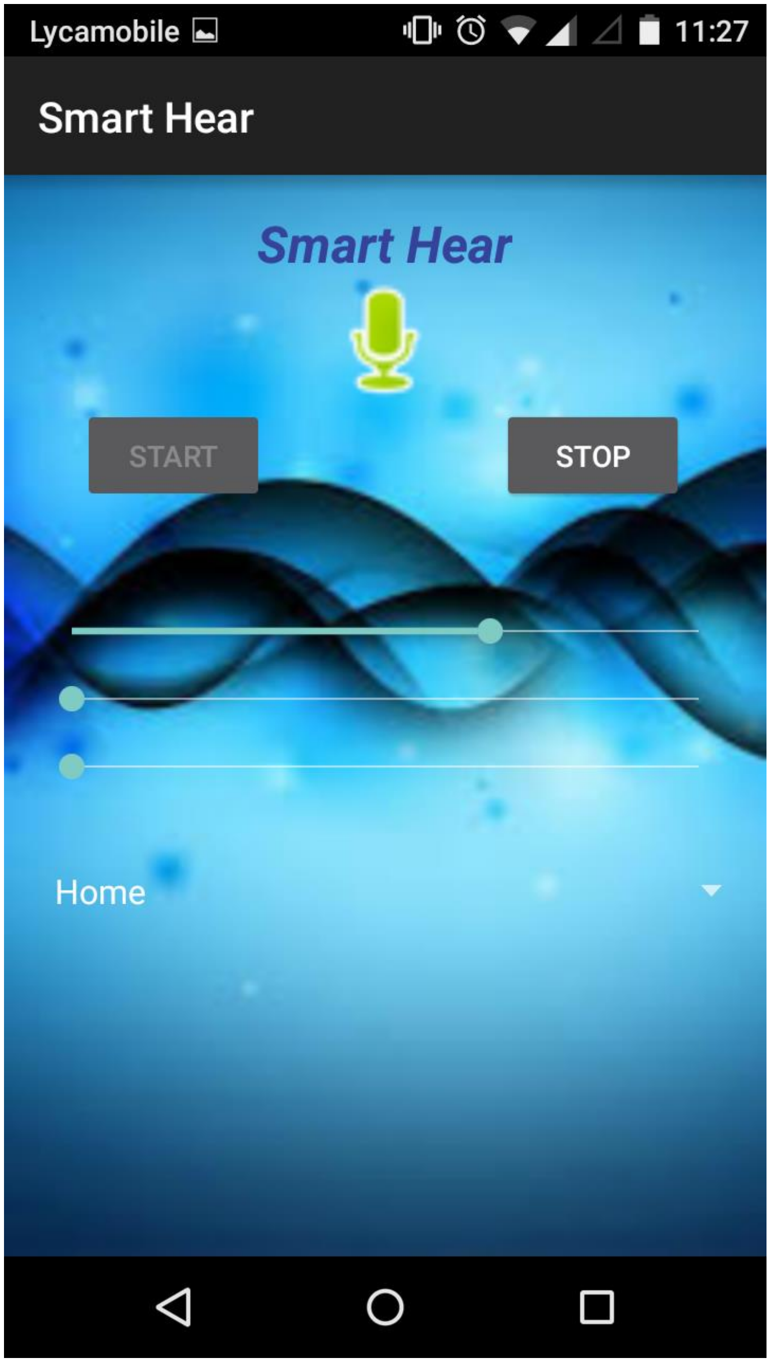


Figure 4: The screen shows the start of the recording activity which would be sent to server for analysis.

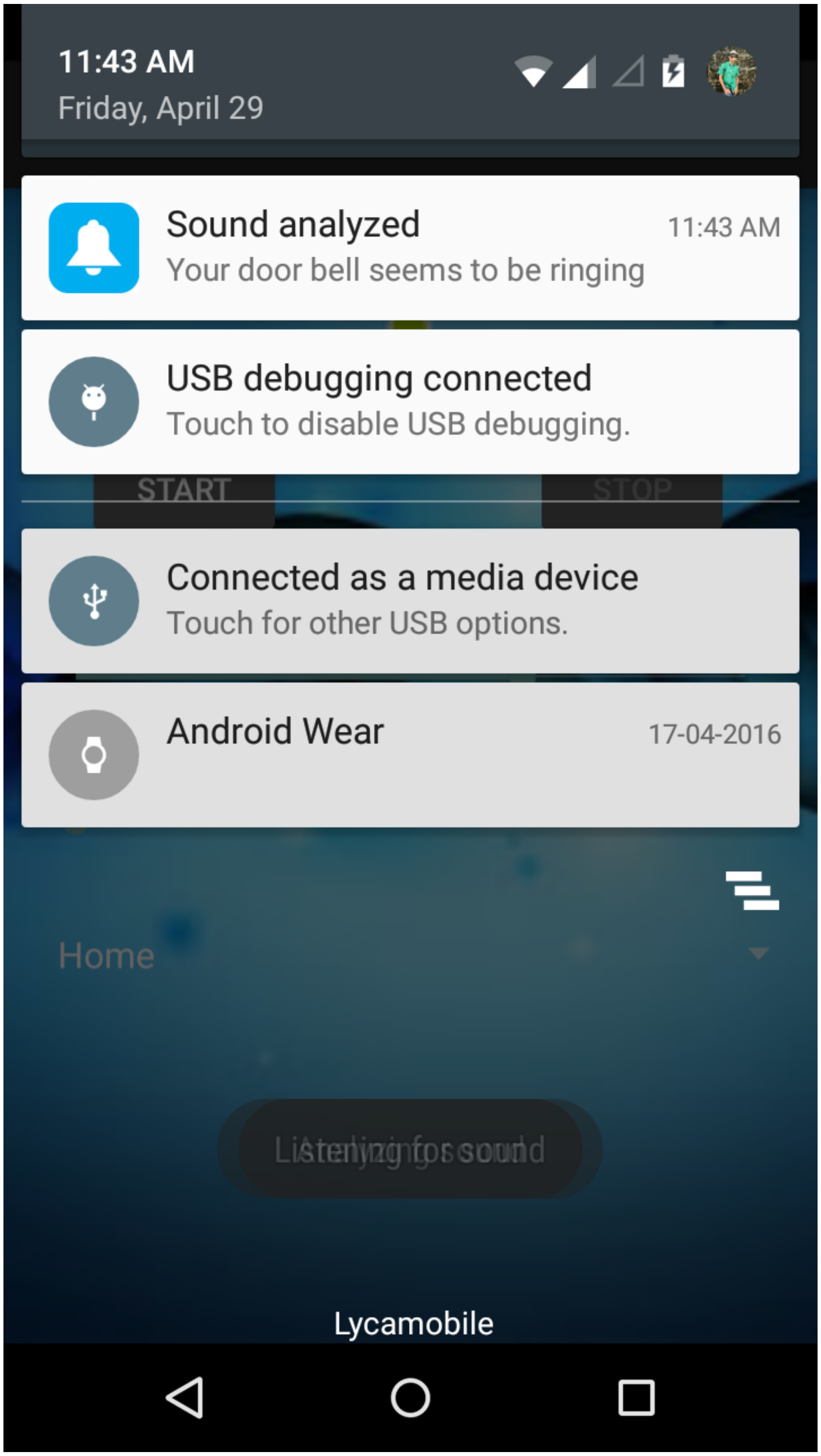


Figure 5: The notification to the user that is shown after the server has done analysis on the features extracted from client.

III. Bibliography

- https://play.google.com/store/apps/details?id=mg.locations.track5&hl=en
- https://play.google.com/store/apps/details?id=com.fsp.android.friendlocator&hl=en
- http://www.raywenderlich.com/120177/beginning-android-development-tutorial-installing-android-studio
- http://developer.android.com/tools/building/building-studio.html
- https://usa.bestsoundtechnology.com/ces/#
- http://www.hearingreview.com/products/new-product-technology/
- http://www.resound.com/en-US/hearing-aids/resound-linx2#.Vse5ZPkrLcs

SmartHear - Intelligent Hearing for Android

Final Project Report

Submitted on **09th May 2016**

Group 1:

1. Ragunandan Rao Malangully (13) **2. Ravi Kiran Yadavalli (31)**

I. Introduction

This document intends to provide an overall description of the project named "Smart Hear" in detail. The project schedule and the plan of action is also discussed. The proposal document submitted would give an insight about what the project is about. Audio detection and analysis is one of the major fields of research that is yet to reach its fullest potential. A subset of this research is to aid the people challenged in hearing sounds. Hearing dog is one such solution that notifies the user on hearing any sound. With the growth in the audio data size and the need to perform real time analysis on the streaming audio, it is very puzzling as to why it is an unexplored area on the big data platform. These concepts were the motivation behind our goal of aiding the challenged users using an audio detection and analysis application using the big data tools. The focus of the study was to evaluate the various machine learning algorithms performance in terms of audio signal detection and analysis with respect to varying features used for analysis.

II. Project goal and objectives

Overall goal

The overall goal is to provide a hearing aid through the use of the smart phone that is accessible to every person today. The smart phone can be utilized in a way that it can act similar to the hearing dog that is what motivated us to take up this idea and try to implement using the smart phone.

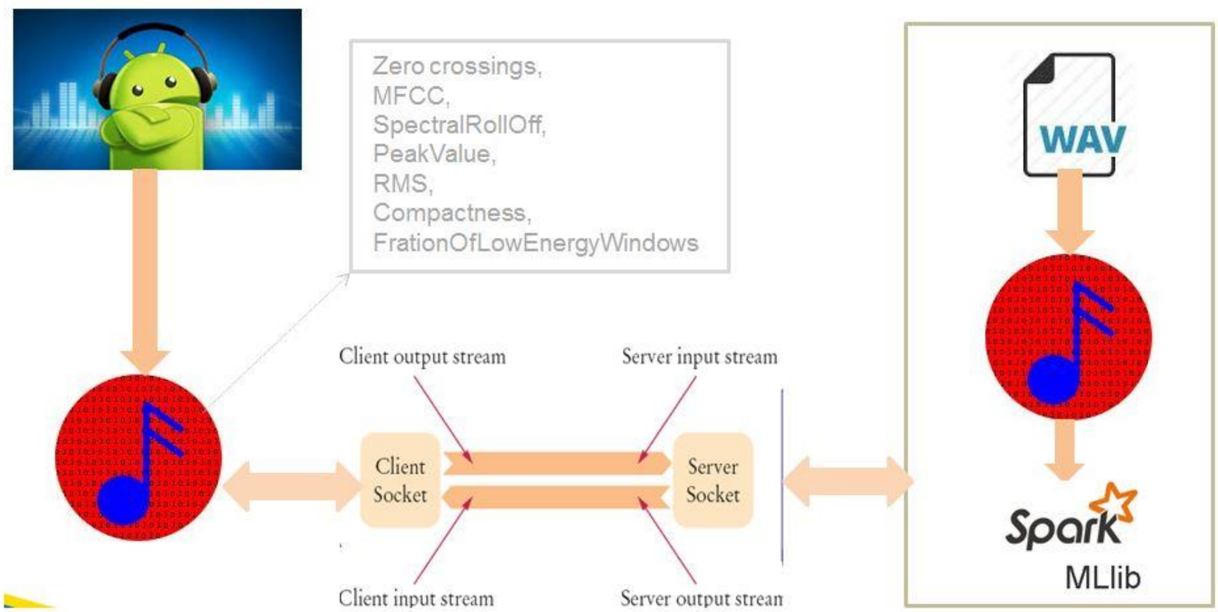
This application can be used to provide benefits to the special abled people who face challenges in listening to sound. All the features come at the cost of installing an application that is freely provided to the user. No costs attached.

Objectives

- To provide a smart hearing aid to the user.
- To implement features to provide the user flexibility in varying certain key features of sound like frequency, tempo etc.
- To provide notifications to the user when there are important events like a door bell ring or an alarm that goes off.
- To make sure that the context is recognized and the user settings are changed accordingly.
- To ensure that the response from the server is at acceptable limits and is accurate.
- To present the ideal set of features for better accuracy of the audio class detection.
- To evaluate the various machine learning algorithms and pick the best one for audio detection and prediction of audio class.
- Provide user customizable settings that the user can adjust to suit his or her needs.
- Ensure that the application can deliver its functionality at minimal cost of operation.

III. Architecture & Application

The architecture of the application is pretty simple. We have an android client that records the audio sounds and sends the audio to JAudio library for feature extraction. The JAudio library extracts the seven features mentioned in the architecture and also adds the contextual information. This information is then sent to the spark server engine through a socket connection. The spark server is trained with .wav audio files and real time data from the device. The .wav files are sent to the JAudio library for feature extraction. These features are then used to train the model at the server end. Trained models are then saved to files. Each context has a different model that is trained and saved for that context. Based on the context information a model for prediction is selected. Then the features sent from the client are used to predict the audio class. This information then is sent back to the android client through a socket connection. Based on the information sent from the server the android client creates a notification for the user to alert him regarding the audio event.



Machine Learning (Algorithms)

- Real time dynamic model training by collecting features using Spark MLlib from client side feature set.
- Multiple models have been built based on several parameters like efficient set of feature set, machine learning model, number of classes.
- Server client connection established through socket connection.
- The serval experiments conducted among Naive Bayes, Random Forest and Decision Tree algorithms with varying number of features.

IV. Data

In the training phase of the application twenty audio classes were used in the model. Each model had fifteen audio files for training and validation purpose. To improve the accuracy and to add the real time nature to the application some real time data collected from the device was also provided to the model as training data. Context aware systems are a component of ubiquitous computing or pervasive computing environment. In our current work, we have devised four important contexts based on the geographical prevalence.

- Home Context
- Classroom Context
- Outdoor Context
- Office Context

Design of context aware model captures important aspects in audio analysis such as a noise level and mining and training of sounds in location specific model. Our experiments have proved that a context aware model has 2.19 % more accuracy than an aggregation of all the contexts as a single general context.

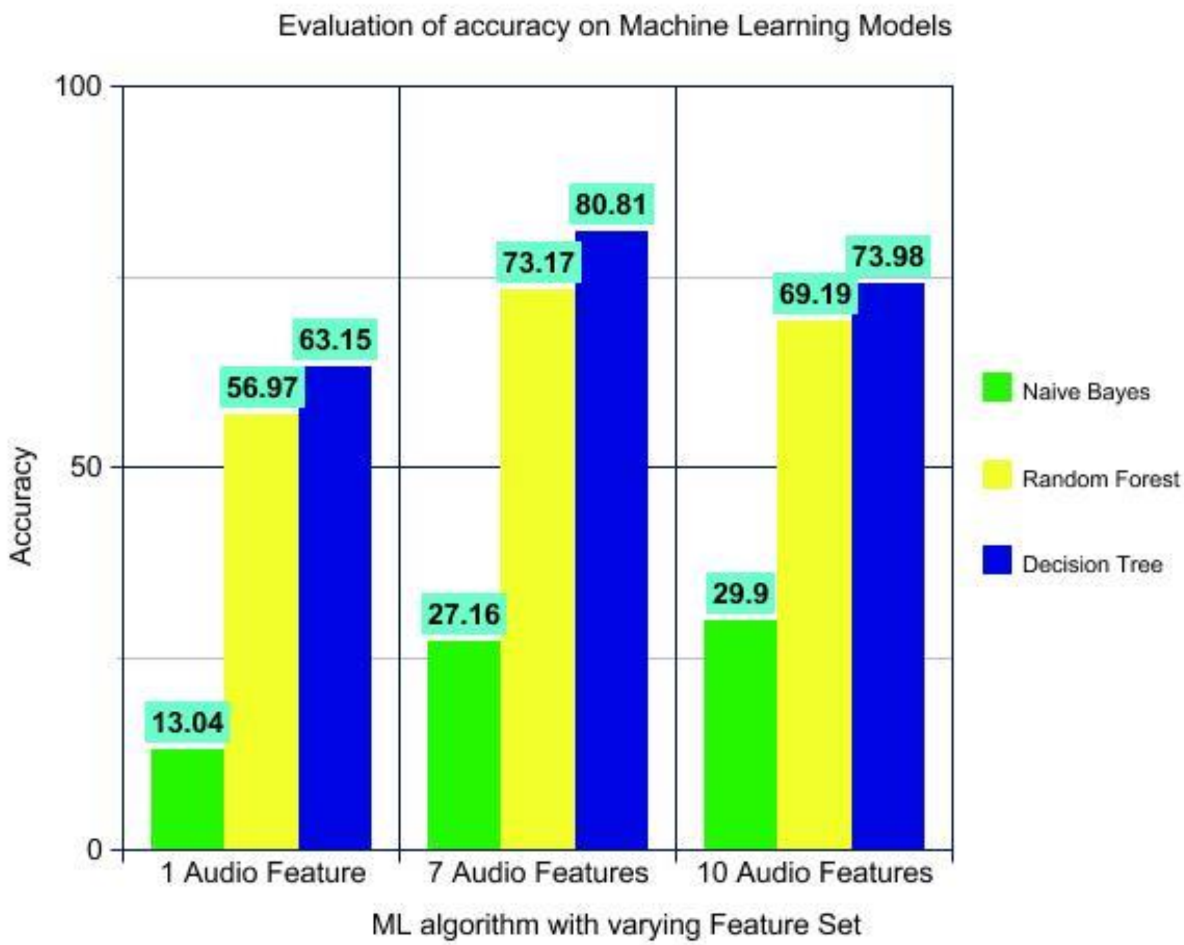
Feature Extraction

- Audio feature extraction is extracting properties, such as beat points, statistical summaries, along with many other less obviously useful properties.
- These properties can then be fed to machine learning toolkits (Here it is Apache Spark) to automatically extract properties (such as artist or genre) from unknown music.
- More interesting applications include prediction of sounds based on the location and redundant time-based daily activities or events.
- Based on the AudioRecorder and MediaRecorder API's provided by Android the audio byte data is captured and analyzed to extract feature set using JAudio library and send to spark for machine learning.
- The features used in the application are Zero crossings, MFCC, SpectralRollOff, PeakValue, RMS, Compactness, and FrationOfLowEnergyWindows as they resulted in the most accuracy for the analysis.

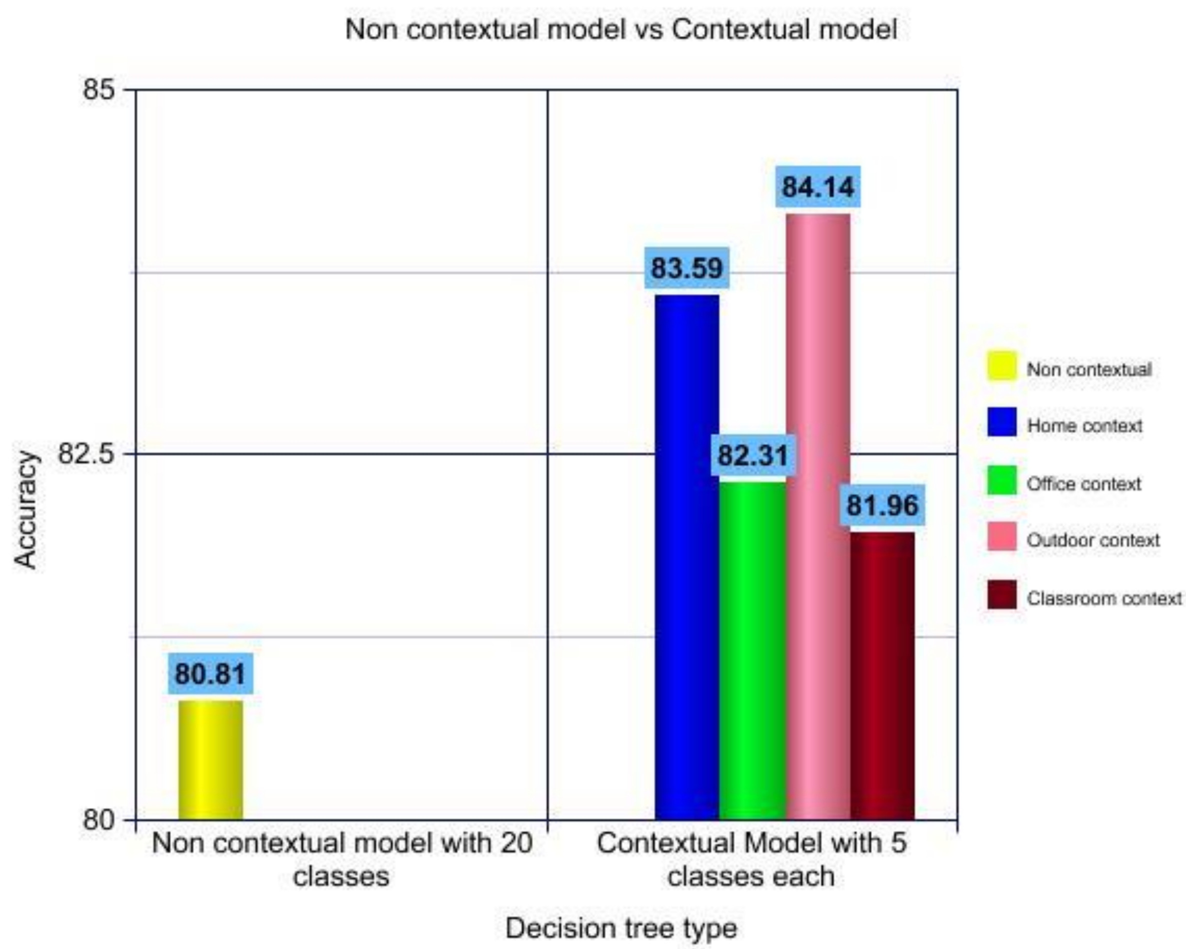
V. Evaluation and Accuracy

There were a series of experiments performed in the study of the application improvements. The following are the evaluation results and analysis for the experiments.

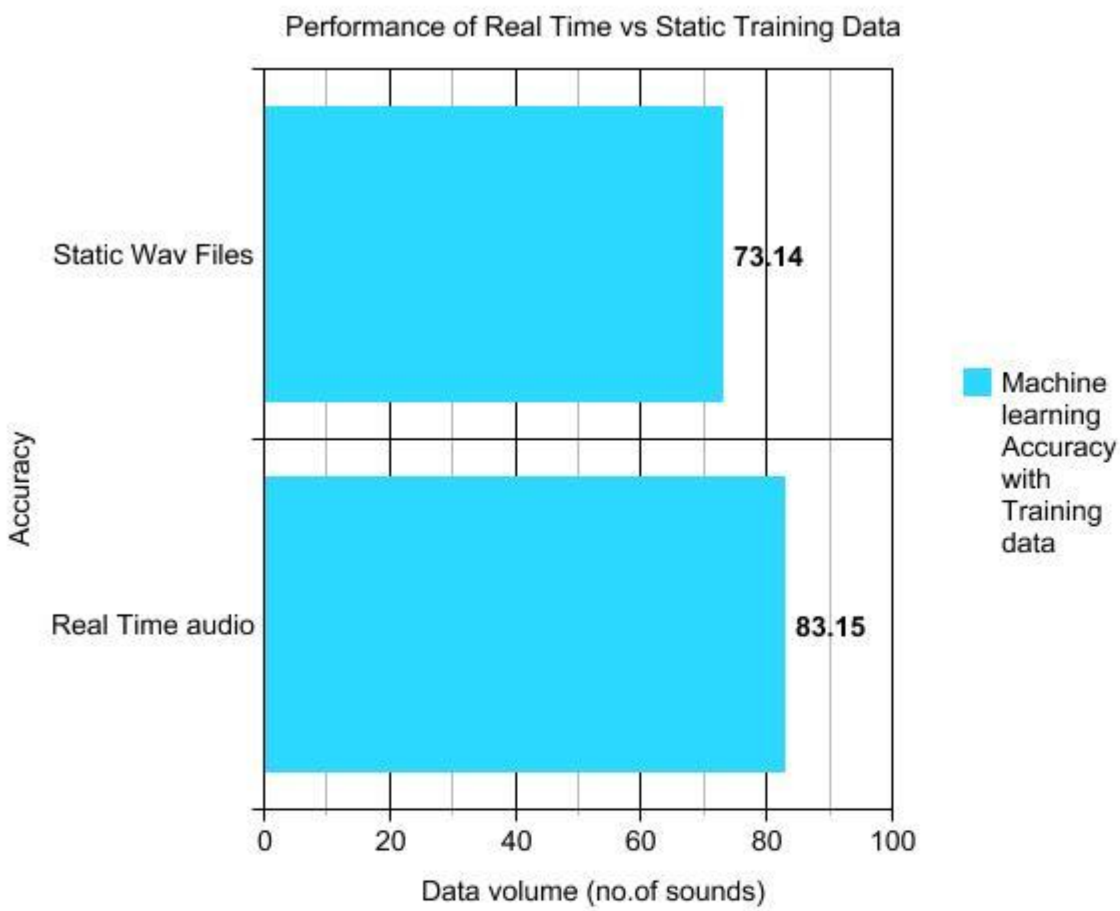
1. Finding the ideal features set for improving the accuracy of the prediction model. In this evaluation the focus was to find a match for the best model in terms of highest accuracy. At the same time varying the features set size to minimize the effort on the model while trying to increase the accuracy. The figure below presents the case that based on the comparison for accuracy for Naïve Bayes, Random Forest and Decision Tree algorithms with 1, 7 and 10 features respectively. From the figure it can be deduced that Decision Tree with 7 features (Zero crossings, MFCC, SpectralRollOff, PeakValue, RMS, Compactness, and FrationOfLowEnergyWindows) resulted in the best accuracy. So it was used in the application.



2. Comparison of the accuracy for Non contextual model versus the contextual model. In this exercise we had two models, one where there was no context information for the model. There was only a single model with 20 classes and training data for all the 20 classes. On the other hand we had 4 different models each for one of four contexts. Each contextual model had 5 classes each and training data was limited only to these 5 classes for each model. Then we compared the accuracy of non-contextual model vs each of the contextual model. As the following figure depicts the contextual models outperform the non-contextual model as the contextual models each have limited classes to predict which in turn improves accuracy of the model.



3. In this experiment the focus was on the study of the accuracy of the model prediction with only static data in training versus model prediction of model with real time data included in training. It was clearly evident that with static data in training the accuracy of the model was only 73 percent as audio data varies with a lot of factors such as noise, context etc. When real time data was also added for training of the model there was an increase in the accuracy to 83 percent as most of the real time data captured the variance of the audio data in terms of noise and other factors. This is depicted in the figure below.



VI. Runtime Performance

The application had acceptable performance in terms of the run time required for the entire work flow. In terms of the model training and model saving the run time required was 20 minutes and 34 seconds for the non-contextual model. While it around 6 minutes for each of the four contextual models. The runtime for one cycle of audio recording, feature extraction, prediction of audio class and notification to the user was as following.

- Audio recording – 4 seconds
- Feature extraction and audio class detection at server – 3.42 seconds.
- Response from server to client and notification to the user – 1.89 seconds.

VII. Future Work

The current model predicts the sound class in a particular context from a set of four contexts and 5 classes each with 83.00 % accuracy on average from several experiments conducted from thousands of .wav sound files and over 250 hours of real time audio data.

- Based on the current analysis, the audio features apparently vary from device to device and the noise levels. Arriving at a common model for prediction for all the devices.
- Improving the accuracy of the model with more classes and dynamical addition of data for training of the model.
- The challenge ahead of us is to devise a model to handle device configuration dynamically and adjust the noise levels accordingly.
- Collaborative learning from multiple devices to improve the prediction of model.
- Automatically detecting the user context and efficient usage of the battery for optimal performance while saving the energy consumption.

VIII. Related work(References)

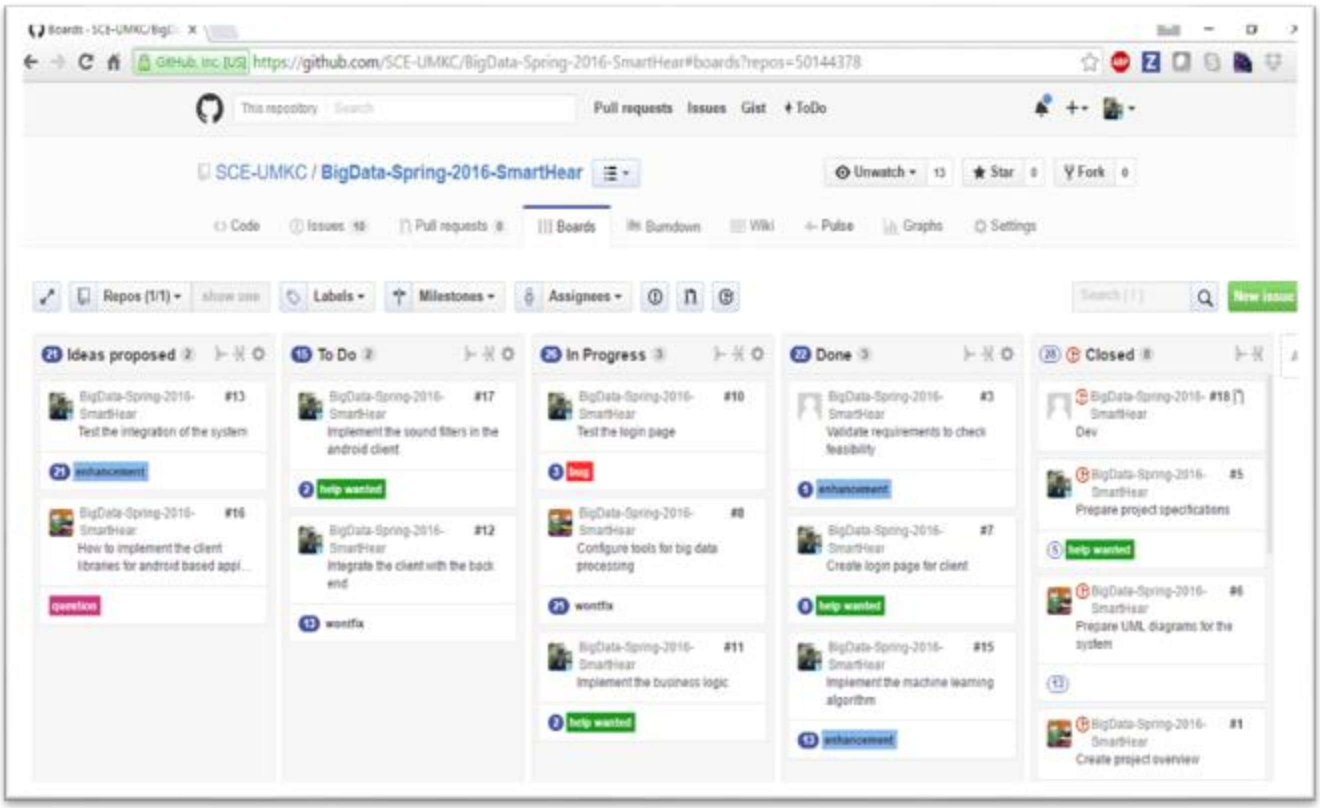
- https://play.google.com/store/apps/details?id=mg.locations.track5&hl=en
- https://play.google.com/store/apps/details?id=com.fsp.android.friendlocator&hl=en
- http://www.raywenderlich.com/120177/beginning-android-development-tutorial-installing-android-studio
- http://developer.android.com/tools/building/building-studio.html
- https://usa.bestsoundtechnology.com/ces/#
- http://www.hearingreview.com/products/new-product-technology/
- http://www.resound.com/en-US/hearing-aids/resound-linx2#.Vse5ZPkrLcs
- http://jmir.sourceforge.net/publications/ISMIR_2005_jAudio.pdf

IX. Project Management

The project was managed mainly as two parts. The client side part and the server side part. As there were two members in the team the work was divided equally for both the members. Ragunandan Rao was responsible for development and testing of the client side module. While Ravi Kiran was responsible for the development and testing of the server side module. The ZenHub was used for creating milestones for each increment and the issues were assigned to each team member. This was used for tracking the contribution. Even the documentation was divided into parts for contribution. PFB the screenshot of ZenHub for the management throughout the project.

Ragunandan Rao Malangully – 50 percent

Ravi kiran Yadavalli – 50 percent.



X. Final project evaluation

The project completed has been a very good implementation of the inception concept. Most of the requirements that were drafted were implemented. The design of the project was not straight forward. The use of agile process really helped the cause as we had to change some of the design policies intermediately. This would not have been possible if agile process was not in place. Agile process is definitely something that we would practice for every project that we would take up. The schedule of the project was not fully practiced as there were some points in the timeline where the required amount of work or goal was not achieved. This caused some alterations in the work schedule but in the end things fell into place. Mostly team meetings weekly was the main source of the project progress and management discussion. As a team we would be updating each other weekly regarding the previous week work completion as well as goals for next week. Also we had the ZenHub for our reference. We did have conflicts on how we can divide the work for each team member but once we had a discussion within the team the problem was solved. For the project the most time consuming part was to figure out to use the best model and to extract features. We believe a little more study on the topi c of the project before taking up the task would really help the cause. We believe if this was a real world project then there would be more refinement in terms of the flow of the project like preprocessing of the audio signal before feature extraction. Improvement of the UI. More customization options. Also we could have seen a mixture of models for prediction of audio class. The recommendation for next year would be that if the topics of interest were provided to the students then it would be better. As w e are of the view that students would be having a limited scope in choosing the topic for their project. This would limit the functionality of the project and also the application would not give an impression of solving real time challenges.