# 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

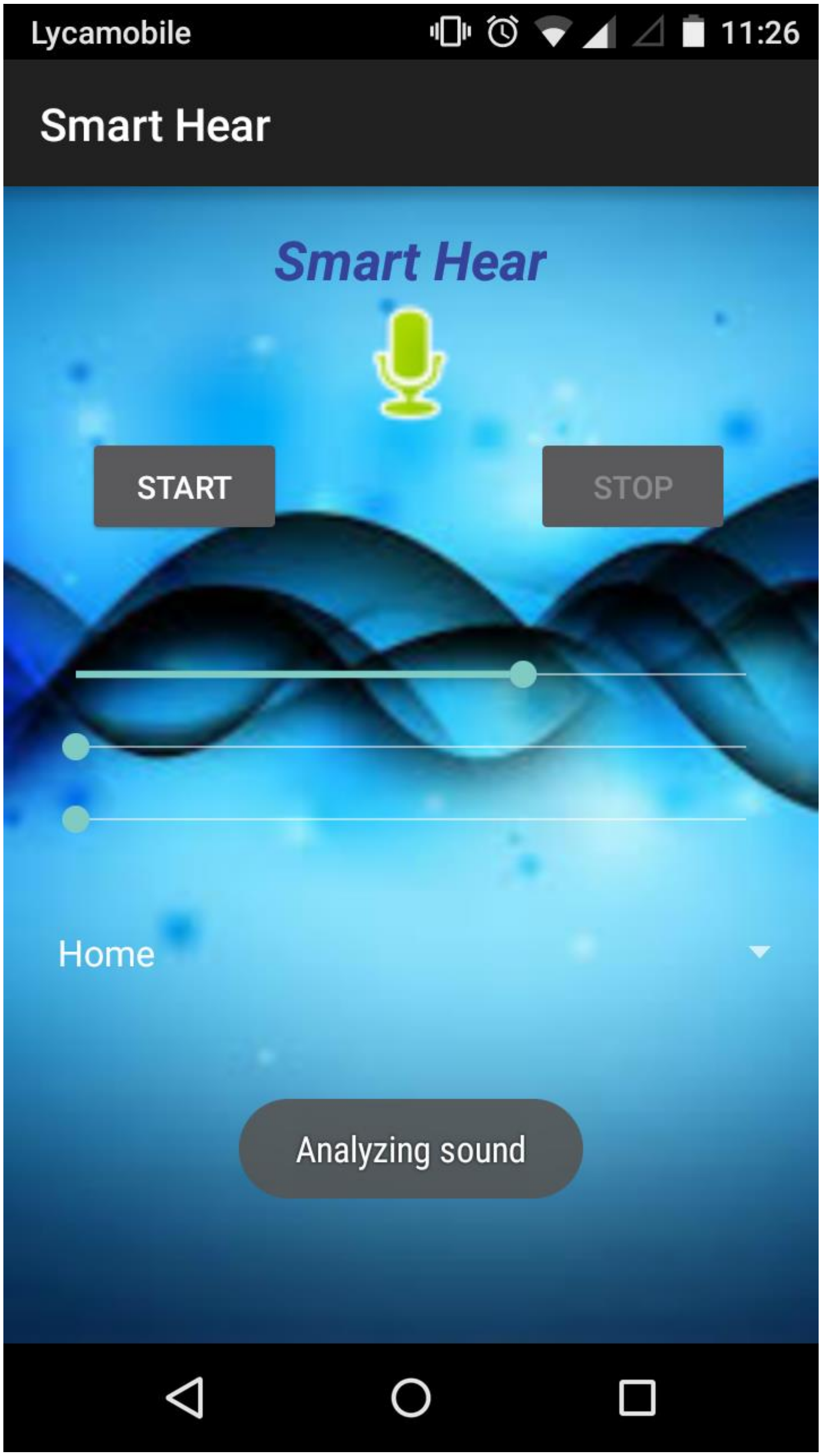
This increment mainly focused on extracting the features at the client end using some real time audio feature extraction library. We had experimented using various audio libraries such as TarsosDSP, musicg and Jaudio. Based on the results and the ease of implementation we decided to use Jaudio feature 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



Analyzing sound

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

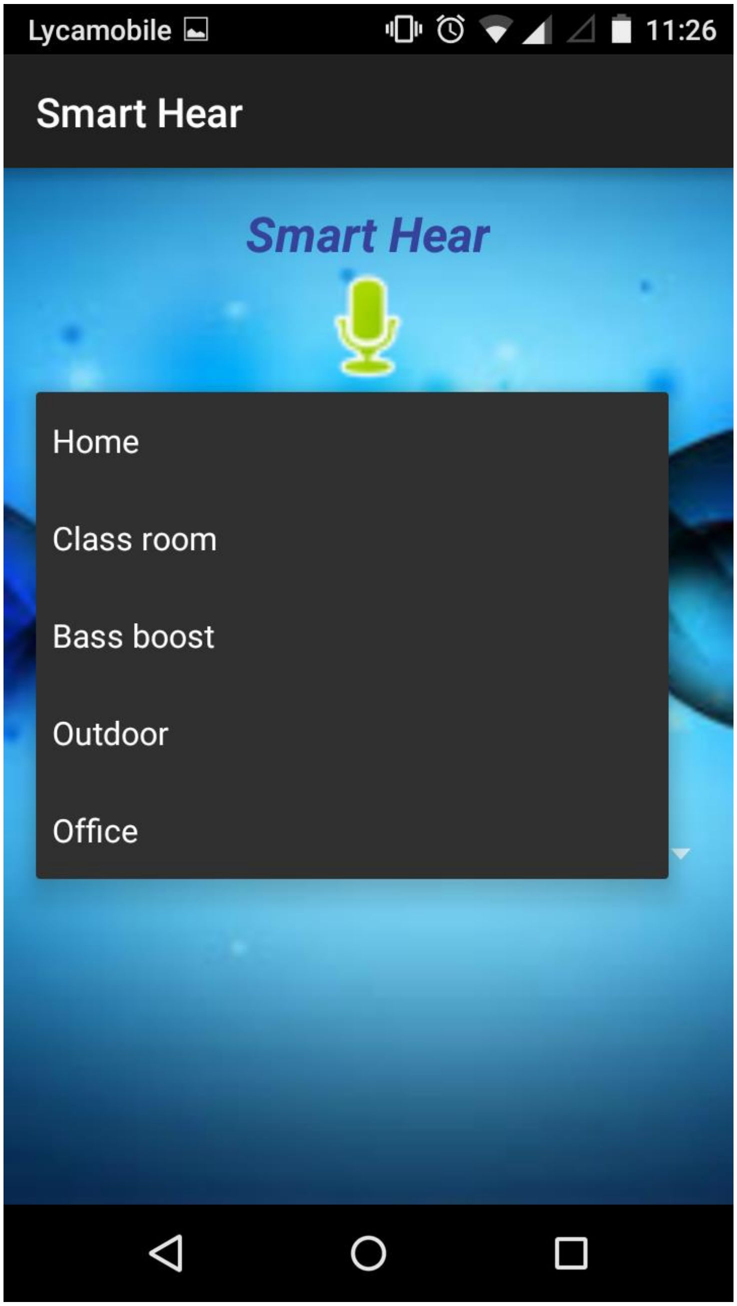


Figure 2: The screen with the different context options for the user to pick from.

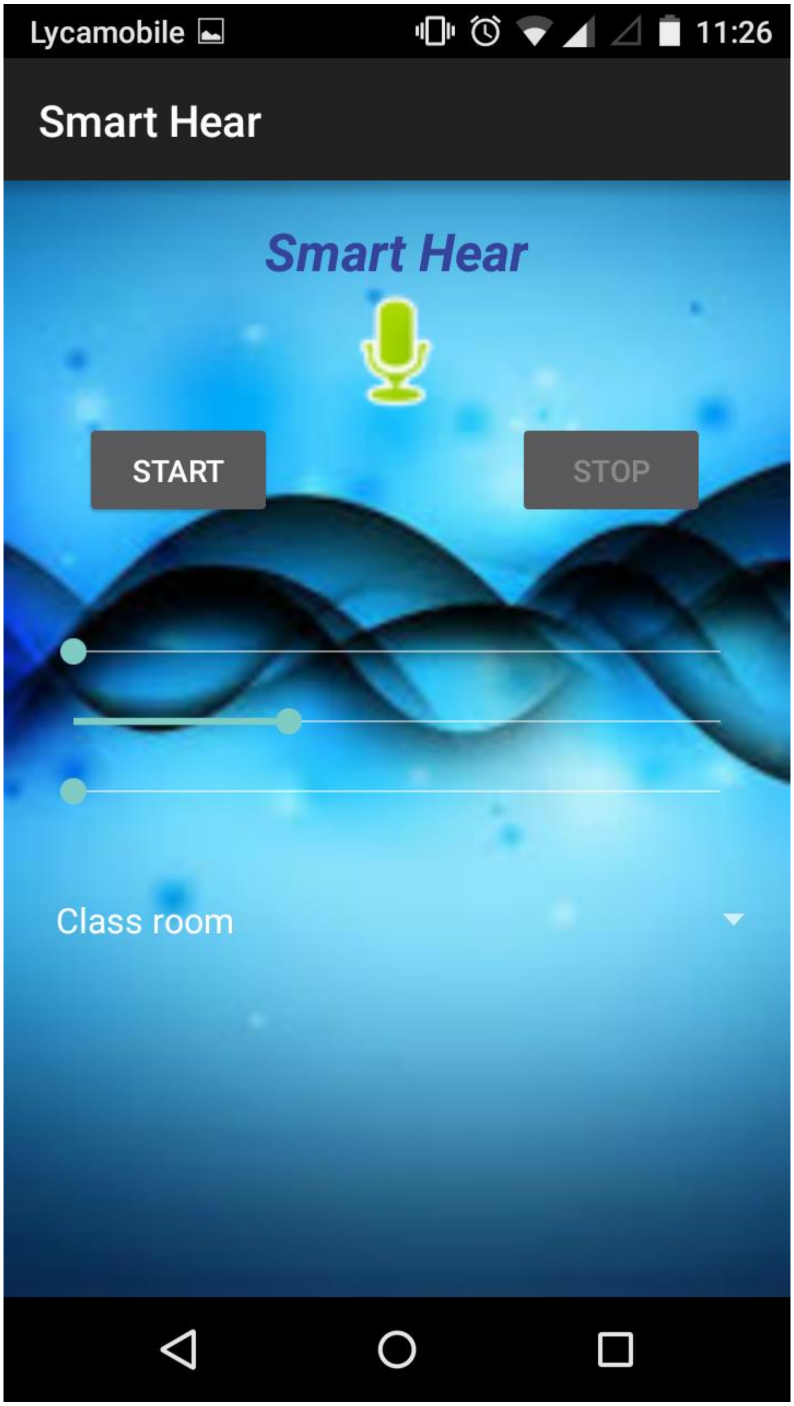


Figure 3: One of the sample screen showing the settings for one of the context selected.

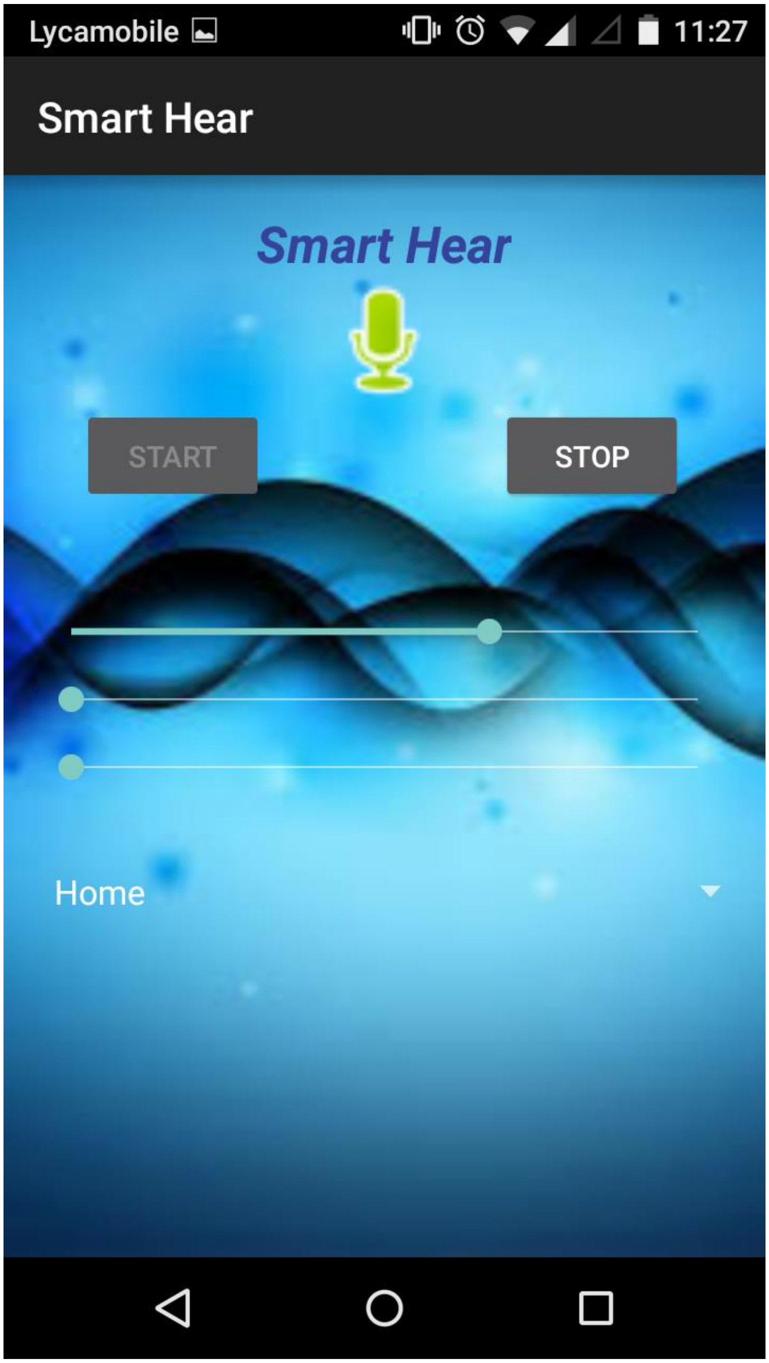
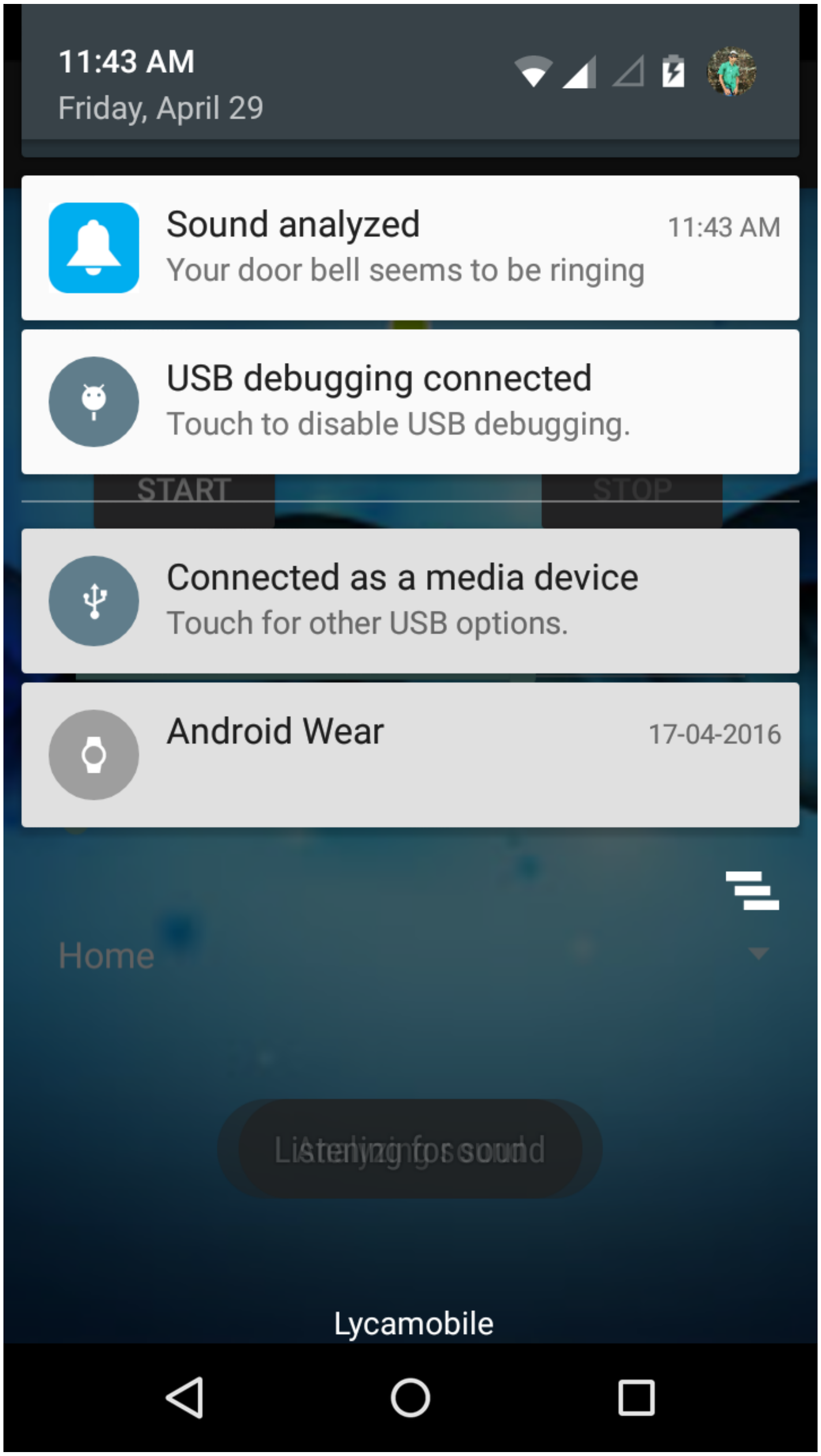


Figure 4: The screen shows the start of the recording activity which would be sent to server for analysis.



11:43 AM
Friday, April 29

**Sound analyzed** 11:43 AM
Your door bell seems to be ringing

**USB debugging connected**
Touch to disable USB debugging.

**Connected as a media device**
Touch for other USB options.

**Android Wear** 17-04-2016

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