

**CSE 323 (Operating System)**  
**Project Final Report**  
**Section: 08**

**Project Title: Sound Pollution Detector**



**Group No: 01**

**Submitted By**

Townim Faisal Chowdhury (1721327042)

A.S.M. Irfan (1520297042)

Yakut Marzan (1430034042)

**Submitted On: August 28, 2019**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**  
**NORTH SOUTH UNIVERSITY**  
**SUMMER 2019**

## Table of Contents

1. Introduction .....	3
2. Executive summary.....	3
2. Goals.....	3
3. Model and Model Integration.....	3
4. Requirement Definition.....	4
5. Feasibility .....	4
6. Design.....	4
7. Product Launching .....	5
8. Future Implementation .....	5
9. Project Contribution.....	5
10. Important Project's Snippets .....	6

## 1. Introduction

In recent decade, the massive industrial rise and the huge increase in the number of vehicles have caused the levels of sound pollution to increase in Dhaka city. We made an app which is helpful for the people because by using this app people are enabled to monitor the amount of sound pollution surrounding them. This pollution is measured by mobile microphone.

## 2. Goals:

The main goal of this app is, with the help of this app we can indicate decibel by gauge. It will help display the current noise reference. It will also notify the users about the current noise. It will also be able to display decibel by graph line.

## 3. Model and Model Integration:

The whole process of software development is divided into separate phases. For the breakdown of project activities into linear sequential phases, where the next phase depends on the previous phase for completing their task is the Waterfall model.

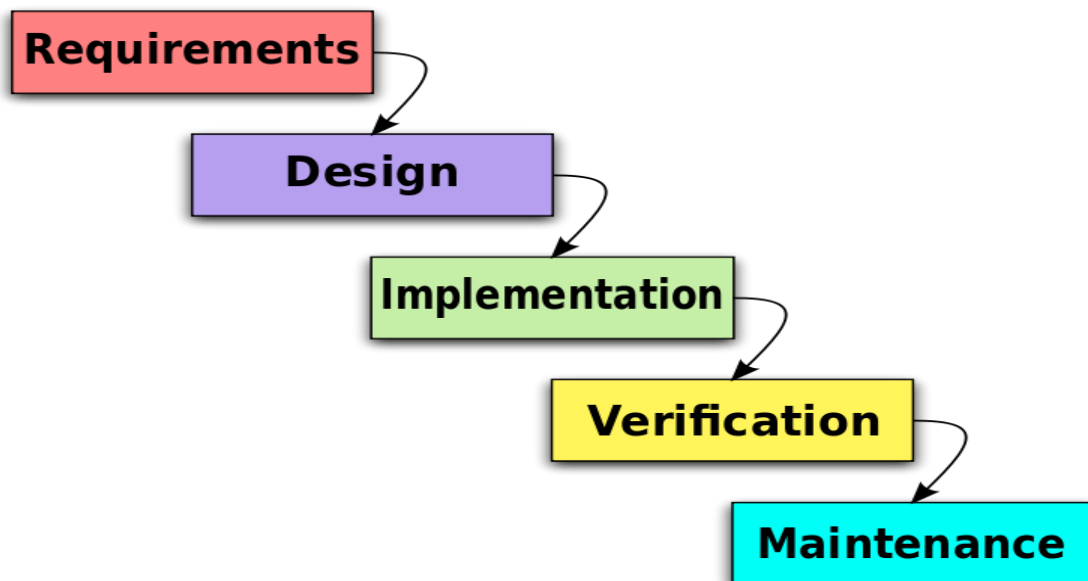


Figure: Waterfall Model

The procedure how we have integrated “Waterfall model” is described below with the phases.

**Requirement Gathering and Analysis:** We collected all possible requirements of the system to be developed in this phase and documented in a requirement specification document.

**System Design:** We studied the requirement specifications from first phase in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

**Implementation:** With inputs from the system design, the system was first developed in small programs called units, which are integrated in the next phase.

**Verification:** All the units developed in the implementation phase were integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

**Deployment and Maintenance:** Once the verification had done; the product was deployed. In maintenance, we have not done any significant work as the product was not launched in the market.

#### **4. Requirement Definition:**

The application was tested on API level 24(Nougat) and 26(Oreo) with the minimum SDK version set to 19.

- API level 24(Nougat) and 26(Oreo)
- Minimum SDK version 19
- Android studio
- Device using Android operating system
- Device supporting Microphone

#### **5. Feasibility:**

It is feasible in our time frame to achieve our goal. The complexity level is low compared to other related work and this is user friendly as well.

#### **6. Design:**

For implementing the project, this designed algorithm is used.

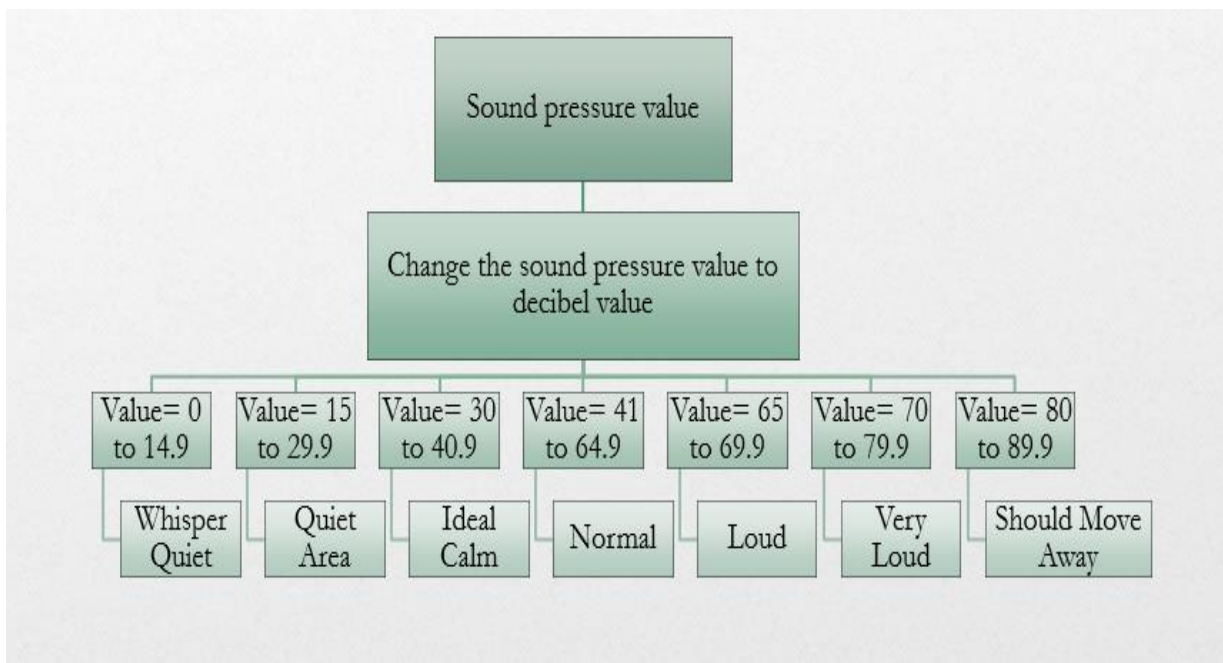


Figure: Algorithm Design

## 7. Product Launching:

We have plan to launch the product as an android app in Google Play Store. Here the link of play store is given: <https://play.google.com/store?hl=en>

Later, we have the planning to make an iOS version of this app for iPhone user.

## 8. Future Implementation:

We want to spread this among all the people so that they can remain safe all the time whenever necessary and they can use this sound pollution detector whenever they need it. We have plan to notify other user about this specific area. We have also a plan to implement alert message to the local police about the noise pollution. We will try to improve more accuracy of measuring noise. If we get a good response from the users, we may think about commercial perspective of this app.

## 9. Project Contribution:

The below table shows the project contribution of each group members.

Name	Project Contribution
Townim Faisal Chowdhury	Logical implementation of project in code, Project report, Project presentation
A.S.M. Irfan	UI/UX design, Project report, Project presentation
Yakut Marzan	UI/UX design, Project report, Project presentation

## 10. Important Project's Snippets:

All the important code snippets including the pictures and the table of calculating error is given here:

```
MyMediaRecorder.java ×
9 public class MyMediaRecorder {
10     public File myRecAudioFile ;
11     private MediaRecorder mMediaRecorder ;
12     public boolean isRecording = false ;
13
14     public float getMaxAmplitude() {
15         if (mMediaRecorder != null) {
16             try {
17                 return mMediaRecorder.getMaxAmplitude();
18             } catch (IllegalArgumentException e) {
19                 e.printStackTrace();
20                 return 0;
21             }
22         } else {
23             return 5;
24         }
25     }
26
27     public File getMyRecAudioFile() { return myRecAudioFile; }
28
29
30
31     public void setMyRecAudioFile(File myRecAudioFile) { this.myRecAudioFile = myRecAudioFile; }
32
33
34
35     public boolean startRecorder(){
36         if (myRecAudioFile == null) {
37             return false;
38         }
39         try {
40             mMediaRecorder = new MediaRecorder();
41
42             mMediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
43             mMediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
44             mMediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
45             mMediaRecorder.setOutputFile(myRecAudioFile.getAbsolutePath());
46
47             mMediaRecorder.prepare();
48             mMediaRecorder.start();
49             isRecording = true;
50             return true;
51         } catch (IOException exception) {
52             mMediaRecorder.reset();
53             mMediaRecorder.release();
54             mMediaRecorder = null;
55             isRecording = false ;
56             exception.printStackTrace();
57         } catch (IllegalStateException e){
58             stopRecording();
59             e.printStackTrace();
60             isRecording = false ;
61         }
62         return false;
63     }
64 }
```

```

C MainActivity.java x
281     data.clearValues();
282     data.removeDataSet( index: 0);
283     data.addDataSet(set1);
284 }else {
285     data = new LineData(set1);
286 }
287
288     data.setValueTextSize(9f);
289     data.setDrawValues(false);
290     mChart.setData(data);
291     mChart.getLegend().setEnabled(false);
292     mChart.animateXY( durationMillisX: 2000, durationMillisY: 2000);
293     mChart.invalidate();
294     isChart=true;
295 }
296
297 }
298 /* Sub-chart analysis */
299 private void startListenAudio() {
300     thread = new Thread((Runnable) () -> {
301         while (isThreadRun) {
302             try {
303                 if(bListener) {
304                     volume = mRecorder.getMaxAmplitude(); //Get the sound pressure value
305                     if(volume > 0 && volume < 1000000) {
306                         World.setDbCount(20 * (float) (Math.log10(volume))); //Change the sound pressure value to the decibel value
307                         // Update with thread
308                         Message message = new Message();
309                         message.what = 1;
310                         handler.sendMessage(message);
311                     }
312                 }
313             }
314             if(refreshed){
315                 Thread.sleep( millis: 1200);
316                 refreshed=false;
317             }else{
318                 Thread.sleep( millis: 200);
319             }
320         } catch (InterruptedException e) {
321             e.printStackTrace();
322             bListener = false;
323         }
324     }
325 );
326 thread.start();
327 }

```

```
75     }
76     speedometer.refresh();
77     minVal.setText(df1.format(World.minDB));
78     mmVal.setText(df1.format( number: (World.minDB+World.maxDB)/2));
79     maxVal.setText(df1.format(World.maxDB));
80     curVal.setText(df1.format(World.dbCount));
81     updateData(World.dbCount, time: 0);
82     if(World.dbCount>0 && World.dbCount<=14.9){
83         txtStatus.setText("Wisper Quiet");
84     }
85     }
86     else if (World.dbCount>=15 && World.dbCount<=29.9){
87         txtStatus.setText("Quiet Area");
88     }
89     }
90     else if (World.dbCount>=30 && World.dbCount<=40.9){
91         txtStatus.setText("Ideal Calm");
92     }
93     }
94     else if (World.dbCount>=41 && World.dbCount<=69){
95         txtStatus.setText("Normal Conversation");
96     }
97     }
98     else if (World.dbCount>=65.1 && World.dbCount<=69.9){
99         txtStatus.setText("LOUD");
100    }
101    }
102    else if (World.dbCount>=70 && World.dbCount<=79.9){
103        txtStatus.setText("Very Loud Music");
104    }
105    }
106
107    else if (World.dbCount>=80 && World.dbCount<=89.9){
108        txtStatus.setText("Should Move away");
109    }
110    }
111
112    else if (World.dbCount>=90 && World.dbCount<=110){
113        txtStatus.setText("Concert , MOVE AWAY");
114    }
115    }
116    else if (World.dbCount>=110){
117        txtStatus.setText("IMMEDIATLY MOVE AWAY");
118    }
119    }
120
```



