# Nidra: An Extensible Android Application for Recording, Sharing and Analyzing Breathing Data Collected During Sleep

Jagat Deep Singh

Thesis submitted for the degree of
Master in Programming and Networks
60 credits

Department of Informatics
Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

Autumn 2019

# Nidra: An Extensible Android Application for Recording, Sharing and Analyzing Breathing Data Collected During Sleep

Jagat Deep Singh

Nidra: An Extensible Android Application for Recording, Sharing and Analyzing Breathing Data Collected During Sleep

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

# Acknowledgements

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Leo a diam sollicitudin tempor. Elit ullamcorper dignissim cras tincidunt lobortis feugiat vivamus. Consectetur a erat nam at lectus urna duis. Ullamcorper sit amet risus nullam eget felis eget nunc. Sollicitudin nibh sit amet commodo. Volutpat maecenas volutpat blandit aliquam etiam erat. Sed viverra ipsum nunc aliquet bibendum enim facilisis gravida neque. Metus dictum at tempor commodo. Nunc vel risus commodo viverra maecenas accumsan lacus. Vitae justo eget magna fermentum iaculis eu non diam. Habitant morbi tristique senectus et. Sed enim ut sem viverra aliquet. Lectus mauris ultrices eros in cursus.

Turpis massa tincidunt dui ut ornare lectus. Elit sed vulputate mi sit amet mauris commodo quis imperdiet. Etiam non quam lacus suspendisse faucibus interdum posuere lorem ipsum. Morbi non arcu risus quis. Quis viverra nibh cras pulvinar mattis nunc sed. Tellus cras adipiscing enim eu turpis egestas. Nec tincidunt praesent semper feugiat nibh sed. Ipsum dolor sit amet consectetur. Duis convallis convallis tellus id interdum. Nulla aliquet enim tortor at.

# Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Leo a diam sollicitudin tempor. Elit ullamcorper dignissim cras tincidunt lobortis feugiat vivamus. Consectetur a erat nam at lectus urna duis. Ullamcorper sit amet risus nullam eget felis eget nunc. Sollicitudin nibh sit amet commodo. Volutpat maecenas volutpat blandit aliquam etiam erat. Sed viverra ipsum nunc aliquet bibendum enim facilisis gravida neque. Metus dictum at tempor commodo. Nunc vel risus commodo viverra maecenas accumsan lacus. Vitae justo eget magna fermentum iaculis eu non diam. Habitant morbi tristique senectus et. Sed enim ut sem viverra aliquet. Lectus mauris ultrices eros in cursus.

Turpis massa tincidunt dui ut ornare lectus. Elit sed vulputate mi sit amet mauris commodo quis imperdiet. Etiam non quam lacus suspendisse faucibus interdum posuere lorem ipsum. Morbi non arcu risus quis. Quis viverra nibh cras pulvinar mattis nunc sed. Tellus cras adipiscing enim eu turpis egestas. Nec tincidunt praesent semper feugiat nibh sed. Ipsum dolor sit amet consectetur. Duis convallis convallis tellus id interdum. Nulla aliquet enim tortor at..

# Contents


**List of Listings** **vi**

**List of Tables** **vii**

**List of Figures** **ix**

**I Introduction and Background** **1**

**1 Introduction** **3**
- 1.1 Background and Motivation . . . . . 3
- 1.2 Problem Statement . . . . . 3
- 1.3 Limitations . . . . . 5
- 1.4 Research Methods . . . . . 5
- 1.5 Contributions . . . . . 5
- 1.6 Thesis Outline . . . . . 5

**2 Background** **7**
- 2.1 CESAR: Project Structure . . . . . 7
  - 2.1.1 Extensible Data Acquisition Tool . . . . . 7
  - 2.1.2 Extensible Data Stream Dispatching Tool . . . . . 9
  - 2.1.3 Flow Sensor Kit . . . . . 12
- 2.2 Android OS . . . . . 13
  - 2.2.1 Android Architecture . . . . . 13
  - 2.2.2 Application Components . . . . . 14
  - 2.2.3 Process and Threads . . . . . 18
  - 2.2.4 Inter-Process Communication (IPC) . . . . . 19
  - 2.2.5 Data and File Storage . . . . . 19
  - 2.2.6 Architecture Patterns . . . . . 20
  - 2.2.7 Power Management . . . . . 21
  - 2.2.8 Bluetooth Low Energy . . . . . 22

**3 Related Work** **23**

**II Design and Implementation** **25**

**4 Analysis and High-Level Design** **27**

4.1 Requirement Analysis . . . . . . . . . . . . . . . . . . . . . . 28
4.1.1 Stakeholders . . . . . . . . . . . . . . . . . . . . . . . 28
4.1.2 Resource Efficiency . . . . . . . . . . . . . . . . . . . . 28
4.1.3 Security and Privacy . . . . . . . . . . . . . . . . . . . 28
4.2 High-Level Design . . . . . . . . . . . . . . . . . . . . . . . 29
4.2.1 Task Analysis . . . . . . . . . . . . . . . . . . . . . . . 29
4.3 Seperation of Concerns . . . . . . . . . . . . . . . . . . . . . 31
4.3.1 Recording . . . . . . . . . . . . . . . . . . . . . . . . . 31
4.3.2 Sharing . . . . . . . . . . . . . . . . . . . . . . . . . . 34
4.3.3 Modules . . . . . . . . . . . . . . . . . . . . . . . . . . 36
4.3.4 Analytics . . . . . . . . . . . . . . . . . . . . . . . . . 37
4.3.5 Storage . . . . . . . . . . . . . . . . . . . . . . . . . . 38
4.3.6 Presentation . . . . . . . . . . . . . . . . . . . . . . . . 40
4.4 Data Structure . . . . . . . . . . . . . . . . . . . . . . . . . 42
4.4.1 Data Formats . . . . . . . . . . . . . . . . . . . . . . . 42
4.4.2 Data Entities . . . . . . . . . . . . . . . . . . . . . . . 44
4.4.3 Data Packets . . . . . . . . . . . . . . . . . . . . . . . 47

**5 Implementation** **51**
5.1 Android Components . . . . . . . . . . . . . . . . . . . . . . 51
5.1.1 Inter-Process Communication: AIDL . . . . . . . . . . 51
5.2 Flow Sensor Kit Development . . . . . . . . . . . . . . . . . . 53
5.3 Implementation of Concerns . . . . . . . . . . . . . . . . . . . 53
5.3.1 Recording . . . . . . . . . . . . . . . . . . . . . . . . . 53
5.3.2 Sharing . . . . . . . . . . . . . . . . . . . . . . . . . . 58
5.3.3 Modules . . . . . . . . . . . . . . . . . . . . . . . . . . 61
5.3.4 Analytics . . . . . . . . . . . . . . . . . . . . . . . . . 63
5.3.5 Storage . . . . . . . . . . . . . . . . . . . . . . . . . . 65
5.3.6 Presentation . . . . . . . . . . . . . . . . . . . . . . . . 66

**III Evaluation and Conclusion** **71**

**6 Evaluation** **73**
6.1 Experiments and Measurements . . . . . . . . . . . . . . . . 73
6.1.1 Experiment A: Orchestral Concert to Analyze Musical Absorption using Nidra to Collect Breathing Data 73
6.1.2 Experiment B: 8-hours recording . . . . . . . . . . . . 78
6.1.3 Experiment C: User-Experience . . . . . . . . . . . . . 78
6.1.4 Experiment D: Creating a Simple Module . . . . . . . 78
6.2 Main Findings . . . . . . . . . . . . . . . . . . . . . . . . . . 78

**7 Conclusion** **79**

**Appendix** **80**

**A Power Data** **83**

# Listings


4.1 My Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
4.2 My Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
4.3 My Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
4.4 My Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
5.1 My Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
5.2 My Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
5.3 My Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
5.4 My Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
5.5 My Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62

# List of Tables


4.1 Example entry in record table . . . . . . . . . . . . . . . . . . . 45
4.2 Example entry in sample table . . . . . . . . . . . . . . . . . . . 46
4.3 Example entry in module table . . . . . . . . . . . . . . . . . . 46

6.1 Device models used during the concert . . . . . . . . . . . . 74
6.2 Day 1—Duration: 1 hour & Roof Samples: 5145 . . . . . . . . 76
6.3 Day 2—Duration: 50 mins. & Roof Samples: 4288 . . . . . . 76

# List of Figures


1.1 Structure of the project, separating functionality into three independent layers [**daniel**] . . . . . . . . . . . . . . . . . . . 4

2.1 Sharing the collected data between multiple applications [**gjoby**] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
2.2 Sharing the collected data between multiple applications [**daniel**] . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
2.3 Recording . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
2.4 Recording . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
2.5 Entity Relationship Diagram . . . . . . . . . . . . . . . . . . 20

4.1 Recording . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
4.2 Recording . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
4.3 Sharing . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
4.4 Modules . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
4.5 Analytics . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
4.6 Storage . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.7 Presentation . . . . . . . . . . . . . . . . . . . . . . . . . . 41
4.8 Modules . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

5.1 Applications components . . . . . . . . . . . . . . . . . . . 52
5.2 Implementation of recording functionality: (A) Start a Recording . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
5.3 Implementation of recording functionality: (B) Stop a Recording . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
5.4 Implementation of recording functionality (C) . . . . . . . . 57
5.5 Implementation of sharing functionality (A): Exporting one or all Records . . . . . . . . . . . . . . . . . . . . . . . . . . 58
5.6 Implementation of sharing functionality (B) . . . . . . . . . . 60
5.7 Implementation of module functionality(A): Add a Module 62
5.8 Implementation of module functionality(B): Launch a Module 63
5.9 Implementation of analytics functionality (A): Display a Graph for a Single Record . . . . . . . . . . . . . . . . . . . . 64
5.10 Entity Relationship Diagram . . . . . . . . . . . . . . . . . . 65
5.11 The recording screen displayed to the user; with the screen: (A) during a recording, (B) real-time analytics, and (C) finalizing the recording . . . . . . . . . . . . . . . . . . . . . . 67
5.12 The sharing screen displayed to the user . . . . . . . . . . . . 68

5.13 The module screen displayed to the user . . . . . . . . . . . . . 69
5.14 The analytics screen displayed to the user . . . . . . . . . . . . 69

6.1 Concert Day 1—Mobile Device B . . . . . . . . . . . . . . . . . . 75
6.2 Concert Day 2—Mobile Device B . . . . . . . . . . . . . . . . . . 75

# Part I

# Introduction and Background

# Part II

# Design and Implementation

# Part III

# Evaluation and Conclusion

# Chapter 6

# Evaluation

## 6.1 Experiments and Measurements

### 6.1.1 Experiment A: Orchestral Concert to Analyze Musical Absorption using Nidra to Collect Breathing Data

The experiment was conducted in collaboration with master student Joachim Dalgard at the University of Oslo at the faculty of *RITMO: Centre for Interdisciplinary Studies in Rhythm, Time and Motion*.

The goal of the experiment was to analyze musical absorption, which is a state an individuals ability and willingness allows music to draw them into an emotional experience and becomes unaware of time and space. In order to analyze the effect of musical absorption on individuals, we gathered 20 participants who were experienced listeners with musical education. The participants attended an orchestral concert by Richard Strauss' Alpine Symphony—a symphonic poem that portrays the experience of eleven hours spent climbing an Alpine mountain—that lasted around 50 minutes at Oslo Concert Hall on third of April and fourth of April 2019.

However, the motivation for this experiment for our application can be summarized into: (1) to test the application in a real-life and crowded environment, in order to analyze whether other mobile devices interfere or obstruct with the signals between the collecting sensor and the device; (2) to test whether the samples gathered are meaningfull, in the sense that the application is collecting the samples from the sensors correctly, and handling unexpected disconnections; and (3) to test the application on different Android OS versions, and to put the application in the hands of participants.

The participants were divided into two groups to attend the concert on the two dates. Each participant was equipped with a wireless electromyographic sensor from DELSYS in order to measure heart rate, and a Flow sensor kit to measure respiration during the concert. RITMO

| Model | Samsung Galaxy S9 | OnePlus 3T | Google Pixel XL |
|---|---|---|---|
| Operating System | Android 8.0 | - | Android 9.0 & Android 7.1.2 |
| Chipset | Exynos 9810 | Qualcomm MSM8996 Snapdragon 821 | Qualcomm MSM8996 Snapdragon 821 |
| CPU | Octa-core | Quad-core | Quad-core |
| GPU | Mali-G72 MP18 | Adreno 530 | Adreno 530 |
| RAM | 4 GB | 6 GB | 4 GB |
| Battery | Li-Ion 3000 mAh | Li-Ion 3400 mAh | Li-Ion 3450 mAh |
| Bluetooth | 5.0, A2DP, LE, aptX | 4.2, A2DP, aptX HD, LE | 4.2, A2DP, LE, aptX |

Table 6.1: Device models used during the concert

had multiple Flow sensors for disposal; however, they had no suitable mobile application that could record with these sensors. Also, with their equipment they experienced that Flow sensor kits tended to disconnect every 10-15 minutes, resulting in fragmented recordings for a single session. Therefore, they reached out to *Insitute for Informatics* in hopes of a solution. Our application was a suiting match for both parties, hence a collaboration was formed [? do i need this]. We arranged for six Android devices and reached out to the participants to bring their Android devices if they had one. There were ten participants in each group and six assessed devices. As a precaution, we decided to give out the devices to the participants who scored highest on a test performed on beforehand.

During the concert, there were approximately 800 attendees on the first day and approximately 1500 attendees on the second day. We assume that most of the attendees had a mobile device, and a few shares of them had BlueTooth activated on the device. Based on these estimates, we were able to replicate an environment (on a larger scale) where other devices might interfere with the signals between the collecting sensor and the application. Also, we were able to install the application on multiple mobile devices with different Android OS versions and put the application in the hands of the participants.

#### 6.1.1.1 Preperations

In order to partake in the experiments, we had to ensure that the participant had the sensor placed accurately on their body and the mobile devices were configured correctly. Additional tests were also performed in order to ensure the application worked as intended, and also to prevent any unforeseen events or bugs that could have occurred during the recording.

**Device Configuration** The device models in our disposal had to be configured with the applications to enable recording on Nidra. First, the data stream dispatching module was installed on the devices. Second, the sensor wrapper for the Flow sensor kit was configured with one Flow sensor on beforehand—in order to reduce the time to set up the mobile device and the sensor on the participant before the concert. Lastly, the Nidra application was initiated with a unique name (A-F), as such to distinguish each participant with the sensor and device model. In Figure 6.1, the device models are listed with their specifications and the OS ... [Skriv mer]

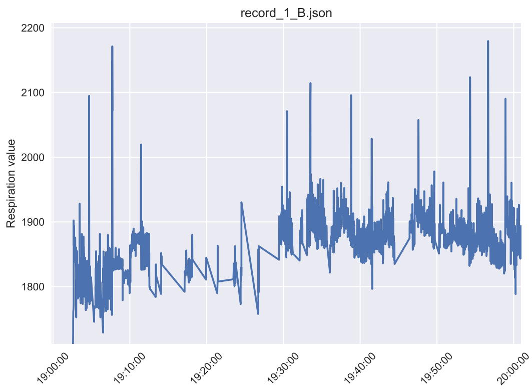


Figure 6.1: Concert Day 1—Mobile Device B

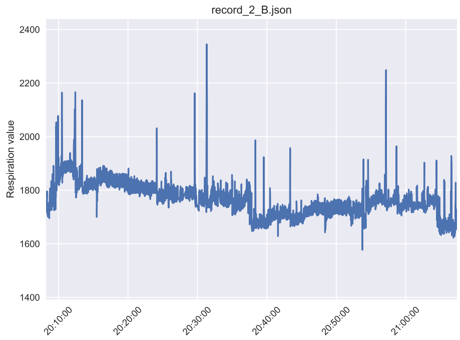


Figure 6.2: Concert Day 2—Mobile Device B

**Body Placement** The respiration value is based on the participant body circumference, and changes are associated with breathing. The participant was instructed to place the sensor around their thorax (just below the armpits), in order to measure the expansion and contraction of the rib cage.

#### 6.1.1.2 Results

We gathered all of the recordings from the various devices by using the sharing functionality in the application and sent it to our application. There were thirteen mobile devices combined for both dates—one of the participants had an Android device—however, the application crashed on one of the mobile devices during the recording. Therefore, we have access to twelve recordings from the concerts.

Figure 6.1 and Figure 6.2 present two of these twelve recordings (rest can is found in Appendix B) that are of most interest to us in a time-series graph. The Y-axis represents the respiration (breathing) value, and the X-axis the time of respiration value acquisition—*day one* of the concert started at the time of 19:00 and ended at 20:00, while *day two* started at the time of 20:10 and ended 21:00. The graphs are plotted with Python and the library MatPlotLib in order to analyze and evaluate the sample values; however, an identical graph is also presented in the application.

From the thesis "..." by Løberg, we can group the signals strengths based on four types of breathing patterns: (1) normal breathing—normal exhaling and inhaling (12-18 breaths per minute); (2) no breathing—close to flat rates over a long period; (3) shallow breathing—rapid inhaling and exhaling; and (4) deep breathing—prolonged inhaling or exhaling (also denoted as fluctuations). From the Figures, we can see that the respiration value is stable with some fluctuations and disconnections. Disconnections are defined as when the line is sloping or benching in an extended period (e.g., 20 seconds or more), while fluctuations are samples that spikes (deep breathing) in the graph. We should keep in mind that the margin for the normal breathing respiration value can vary throughout the recording based on body or sensor position and movements (e.g., sitting relax or tense in a chair).

Figure 6.1 has many disconnections (further analyzed later) with unstable breathing. There are multiple occurrences of deep breathing throughout the recording, with some shallow breathing at the start and the end of the recording. Hypothetically, deep breathing might be a sign of musical absorption; however, we have no concrete analysis of this matter, and it is out of the scope for this experiment motivation. Moving on, Figure 6.2 shows fewer disconnects compared to Figure 6.1, and with much more concise breathing patterns and resemblance to normal breathing. Although, there are noticably deep breathing throughout the recording, with a few shallow breathings at the beginning. Conclusively, both Figures shows no nuances for no breathing during the recording. However, there are signs of normal breathing, with few instances of shallow and deep breathing.

#### 6.1.1.3 Analysis & Discussion

We will discuss all twelve records by analyzing the respiration values. It is of special interest to find occurrences of disconnections—when the samples are stagnant in more extended period—and the time the sensor is disconnected during the recording.

| Model | Samples Count | Loss Count | Loss Percentage | Disconnection Count | Disconnection Time |
|---|---|---|---|---|---|
| A | 5145 | 0 | 0 % | 0 | 00:00 |
| B | 3363 | 1782 | 34 % | 13 | 19m:14s |
| C | 4189 | 956 | 19 % | 7 | 8m:20s |
| D | 3501 | 1644 | 32 % | 5 | 18m:30s |
| E | 5144 | 1 | 0.02 % | 0 | 00:00 |
| F | 5145 | 0 | 0 % | 0 | 00:00 |

Table 6.2: Day 1—Duration: 1 hour & Roof Samples: 5145

| Model | Samples Count | Loss Count | Loss Percentage | Disconnection Count | Disconnection Time |
|---|---|---|---|---|---|
| A | 4286 | 2 | 0.05 % | 0 | 00:00 |
| B | 4161 | 127 | 3 % | 4 | 1m:25s |
| C | 4286 | 2 | 0.05 % | 0 | 00:00 |
| D | 2576 | 1712 | 40 % | 7 | 24m:35s |
| E | 4285 | 3 | 0.06 % | 0 | 00:00 |
| F | 4288 | 0 | 0 % | 0 | 00:00 |

Table 6.3: Day 2—Duration: 50 mins. & Roof Samples: 4288

The recordings for the six devices for the two dates are represented in Table 6.2 and Table 6.3. Each table exhibits data that is extracted from each recording from the mobile device and can be characterized as:

**Roof Sample** The expected number of samples that can be acquired in the period of the recording (based on the frequency of sample output by Flow Sensor).

**Sample Count** The number of samples that were gathered in the duration of the recording.

**Loss Count** The number of missing samples based on `Roof Samples`. This can be calculated as:

$$Loss\ Count = Roof\ Samples - Samples\ Count \tag{6.1}$$

**Loss Percentage** The percentage of missing samples based on the expected samples. This can be calculated as:

$$Loss\ Percentage = \left(1 - \frac{Samples\ Count}{Roof\ Samples}\right) * 100 \tag{6.2}$$

**Disconnection Count** Is the number of disconnections that occurred within the duration of the recording.

**Disconnection Time** Is the accumulated time of disconnection.

The device models A, E, and F are noticeably accurate (with some noises which presumably occurred during parsing), there are no apparent disconnects occurred during the recording for these models both of the days. In contrast, device model B, C, D shows an outburst of disconnections during recording. Especially, model D has high loss percentage both of the days, which is reflected in the disconnection time. Also, model B had a high loss percentage on the first day; however, significantly less loss percentage the second day. Similar resemblance can also be seen in model C.

We could conclude on the fact that some mobile device models or some sensors is malfunctioned, and for this reason we see higher loss rate. Unfortunately, with insufficent data to point out whether that is correct, is is something we cannot conclude on. Moreover, [Skriv mer om Android OS version].

#### 6.1.1.4 Conclusion

To summarize the experiment, we were able to test the application in a real-life and crowded environment. The application managed to record samples that lasted up to 1 hour with the Flow senor and various device models with different Android OS versions. However, one of the application crashed, and we were unable to find the source of the problem. Based on samples from the recording, it is identified that the Flow sensor has a tendency of disconnecting, but the application managed to provide a continuous data stream by reconnecting with the sensor during recording. In the end, the records were successfully shared across applications, which enabled us to analyze the recordings.

To conclude, the goal of the experiment was to analyze musical absorption during a concert, but that is not the motivation for our application. However, we provided an application that collected the data during the concert. Thus, the collected data can then be used to conclude a correlation between breathing and musical absorption. Also, inconsistency in the sampling of the recordings makes it difficult to determine the source of the problem. We could argue that the analysis of the aggregated data in the tables might indicate that the sensor or the mobile is a malfunctioned, however, due to insufficient data this conclusion cannot be drawn. Although, the application managed to collect data with the use of the Flow sensor kit, in an environment filled with other mobile devices that could have interfered with the signals, as well as reconnecting to the sensors during the recording. [Reflect more on the motivation goals].

### 6.1.2 Experiment B: 8-hours recording

### 6.1.3 Experiment C: User-Experience

### 6.1.4 Experiment D: Creating a Simple Module

## 6.2 Main Findings

# Appendix