UNIVERSITÄT OSNABRÜCK

INSTITUTE OF COGNITIVE SCIENCE

Master's Thesis

Design and implementation of an Android app to record, store and manage EEG signals

Author:
Rodolfo Adrián ROJAS CAMPOS

Supervisors:
M.Sc. Johannes LEUGERING
Prof. Dr. Gordon PIPA

Submitted at the Institute of Cognitive Science
Osnabrück University
August, 2019

Contents

1 Introduction **6**

1.1 App description . . . 6

1.2 Why is this app necessary? . . . 6

1.2.1 Epilepsy in the third-world . . . 6

1.2.2 The boom of mobile technology . . . 7

1.2.3 A possible intervention . . . 7

1.3 Previous research . . . 8

1.4 Aim of this project . . . 9

1.5 Thesis outline . . . 10

2 Method **11**

2.1 Definition of design guidelines . . . 11

2.2 Software development platform . . . 11

2.3 Software development methodology . . . 12

2.4 Design of the app architecture . . . 12

2.5 Process of implementation . . . 14

3 Results **16**

3.1 App's name and icon . . . 16

3.2 EEGDroid app . . . 16

3.2.1 Main activity (*MainActivity.java*) . . . 17

3.2.2 Record activity (*Record.java*) . . . 18

3.2.3 Play activity (*Display.java*) . . . 20

3.2.4 Manage activity (*ManageSessions.java*) . . . 22

3.2.5 Learn activity (*Learn.java*) . . . 23

3.2.6 Tutorial activity (*Tutorial.java*) . . . 25

3.2.7 Epibot activity (*Epibot.java*) . . . 26

3.2.8 Time-frequency analysis activity (*TFAnalysis.java*) . . . 28

3.2.9 Settings activity (*Settings.java*) . . . 29

3.2.10 Info activity (*Info.java*) . . . 30

3.3 Tests . . . 30

4 Discussion **33**

4.1 Difficulties faced . . . 33

4.2 Shortcomings . . . 33

4.3 Achievements . . . 35

4.4 Future steps . . . 36

5 Bibliography **38**

A Appendix: Content of Learn activity **40**

A.1 Introduction 40
A.2 What is a seizure? 41
A.3 What is not a seizure? [1] 41
A.4 What types of seizures are? [5] 41
A.5 What causes seizures? 42
A.6 What is a provoked seizure? 42
A.7 What is epilepsy? 42
A.8 How can epilepsy be diagnosed? [2] 43
A.9 What are the underlying mechanisms of epilepsy? [10] 43
A.10 What are the risk factors for epilepsy? [2] 43
A.11 How can epilepsy be treated? [7] 44
A.12 How many people suffer epilepsy? 45
A.13 Where can I learn more? 46
A.14 Bibliography 46

B Appendix: Content of Tutorial activity **47**

B.1 What is the Traumschreiber? 47
B.2 What is EEG? 47
B.3 How EEG works? 47
B.4 Why is EEG important? 49
B.5 How can I use the Traumschreiber? 49
B.6 Bibliography 49

List of Figures

1 Royce (1970) Waterfall model . . . 12
2 EEGDroid's architecture . . . 13
3 Activity's lifecycle. Taken from AndroidDocumentation (2019) . . . 14
4 EEGDroid's icon . . . 16
5 Main activity . . . 17
6 Record activity: disconnected, recording and casting . . . 18
7 Record activity: saving or deleting session . . . 19
8 Record activity: server printing incoming signals . . . 20
9 Play activity: reproducing a session . . . 20
10 Play activity: different speeds and channels . . . 21
11 Manage activity . . . 22
12 Learn activity: index, content and offline warning . . . 24
13 Tutorial activity . . . 26
14 Epibot activity . . . 27
15 Time-frequency analysis activity: results . . . 28
16 Username and ID in .CSV file . . . 29
17 Settings activity . . . 29
18 EEDDroid running in MotoG5S Plus . . . 31
19 EEDDroid running in Huawei Mediapad T5 . . . 32
20 Comparison Main Activity between versions . . . 34
21 Comparison Record Activity between versions . . . 35
22 Comparison Epibot between versions . . . 37
23 Vagus nerve stimulation. Taken from [7] . . . 44
24 Deep brain stimulation. Taken from [7] . . . 45
25 Traumschreiber . . . 48
26 Example of EEG session . . . 48
27 YouTube tutorial video . . . 49

Abstract

In the present work, we design an Android app to collect, plot, record, save, manage, and share EEG sessions, using the Traumschreiber EEG device (Appel et al. (2016)). Additionally, information about epilepsy is presented in the app in a clear and accessible way. A chatbot expert in epilepsy is embedded in the app, with whom the user can chat and evacuate doubts about these topics.

We follow the waterfall methodology (Royce (1970)) and the design was based on basic predefined guidelines. An architecture was design with a total of 10 different modules, in which the main functionality of the app was split.

As a result of the implementation phase, we produced a reliable and stable app. All the code is available online[1]. Once the app was implemented, different processes of testing and correction of bugs were performed.

In this document, we offer a description of the conceptualization, design, implementation, and test phases that compose the complete project. We present the goal and explain the functioning of each one of the modules that contain this app (named activities), as well as some interesting details about the implementation.

Finally, we analyze the challenges, weaknesses and strengths of the app. Based on the weaknesses of our implementation, we propose future steps to be implemented.

[1]https://github.com/adrocampos/EEG-Droid

1 Introduction

1.1 App description

The main goal of this project was to design an app to allow the user to connect with the Traumschreiber, plot, record and save EEG sessions, as well as manage the session in his smartphone / tablet, rename, delete or send them to other devices.

To achieve this goal, we carried out a process of software development during five months, including the design, implementation, and test phases. This process produced a well-designed app with multiple capabilities that is easy to handle and use.

The functionality of the resulting app was split into two major sections: the first section is called "utilities", which includes the record, play, and manage activities. Within the record activity, the user can connect the smartphone / tablet with the Traumschreiber and plot, record or cast the incoming signals. Within the play activity, the user can reproduce recorded EEG signals at different speeds and observe in detail what happened at a particular time point. Using the manage activity, the user can see the stored EEG sessions, rename or delete them, as well as share them using different platforms like email or Bluetooth.

The second section is called "learn" and presents the user essential and reliable information about epilepsy and provides the opportunity to include tutorial content, in order to help the user to operate the Traumschreiber and participate in experiments. Additionally, we included in the app a chatbot called Epibot, as a new interactive way to access content about epilepsy. The user can chat with it and ask frequent questions about epilepsy.

Furthermore, the user can perform time-frequency analysis of the EEG sessions and visualize the result of the analysis. Also, a settings configuration was included, in which the user can save personal data (username and ID) and this information is included as part of the EEG sessions.

1.2 Why is this app necessary?

1.2.1 Epilepsy in the third-world

Epilepsy affects approximately 50 million people worldwide and accounts for about 1 % of the global burden of disease. From all the epilepsy patients, 80 % live in developing countries. Developing countries usually present marked inequalities on the distribution of health care resources among the rich and the poor. Private health care, usually accessible for those with a higher income, may be relatively equivalent to first-world care, while the care for the rest of the population could be deficient or nonexistent (Birbeck (2010a)).

To make a good diagnosis of epilepsy, it is necessary to count with a detailed history of the seizure and to use neurodiagnostics. One key factor that allows a correct diagnostic and treatment of epilepsy is electroencephalography (EEG). EEG is an electrophysiological method for measuring the electrical activity of the brain neurons (Sazgar and Young (2019)). It is the most useful diagnostic procedure for diseases like epilepsy and narcolepsy. It allows health care professionals to confirm the presence of abnormal electrical activity and gives information regarding the type of seizure disorder and the location of the seizure focus (Browne and Holmes (2008)).

Evaluations using EEG are usually carried out in a laboratory or a hospital and require a considerable effort and cost for the institutions and patients. The price of the electroencephalograph rounds the thousands of EUR. In developing countries, due to economic limitations, the needed tools to perform neurodiagnostics (e.g., EEG, neuroimaging) might be very limited or nonexistent. Additionally, as part of the limited resources, the number of neurology professionals could be very low (Birbeck (2010a)).

According to Birbeck (2010a) in context of such shortage of experts, the most effective role for an expert to assume is the one of educator, advisor, and advocate. In the developing world, part of the job as a neurologist is to educate the population about the disease, to provide them with the right guidelines and information, as well as to work with local authorities to ensure that essential care is available.

1.2.2 The boom of mobile technology

On the other hand, the use of mobile technology and the internet is widely extended all over the world. The Mobile Economy Report of 2018 (GSM-Association (2018)) reported that in 2018, a total of 5 billion people were connected with some kind of mobile communication worldwide. From these 5 billion, 57 % have access to smartphones, and 43 % are users of mobile internet. By the year 2025, an increase is expected in the number of mobile internet users to 77 %. This growth will be mainly driven by developing countries, particularly India, China, Pakistan, Indonesia, and Bangladesh, as well as Sub-Saharan Africa and Latin America.

As part of this mobile technology expansion, Google's Android operative system is taking the lead in the market share. According to StatCounter (2017), in the year 2017, Android overcome Windows as the most used operative system in the world.

In 2019 Android represents 75.27 % of the mobile operative system market in the world, very far away from his main competitor, Apple's iOS with 22.74 of the market shares (StatCounter (2019)). Android is expanded between the population, and writing applications for this platform allows to reach a bigger public compared with iOS.

1.2.3 A possible intervention

Considering all these elements, in the Neuroinformatics department emerged the idea of using a low-cost portable EEG (called Traumschreiber, more details in the next pages), as a tool to help to collect data and diagnose epilepsy in developing countries. Given the extended access to mobile connectivity and the Android operative system, it was considered convenient to develop an app that working together with the Traumschreiber, allows easy and low-cost EEG data collection and transmission. An Android app and the Traumschreiber could allow medical doctors and patients to run out experiments that help to establish an adequate diagnose on all those regions where EEG is inaccessible.

Additionally, due to the shortage of experts in developing countries previously mentioned, it was considered necessary that the app also contained a learning section. The app should help to inform and educate people about the different diseases that affect the brain and the possible condition which they might suffer. This goal becomes particularly important considering the many misconceptions associated with epilepsy present in some cultures and the social stigma derived from them, causing social and economic problems to patients and families (Birbeck (2010b)).

1.3 Previous research

The EEG-Droid app was developed in the middle of an already established research ecosystem. It counted with several contributions from different researchers and coders. We present a small review of such collaborations in the next paragraphs.

In the first place, the Traumschreiber was designed by Appel et al. (2016) as a high-tech sleep mask for professional sleep experiments, that combines sleep recording and sleep manipulation. It has eight electrographic channels with reliable data quality and secure wireless data transfer via Bluetooth. It is an open hardware-software project, usable in home-settings, portable, battery-driven, and low-cost.

First tests of the Traumschreiber were made by Appel (2018) as part of his Ph.D. project. This work was centered in the evaluation of the Traumschreiber as a sleep mask to make crowd-based EEG recordings during sleep. Four different experiments were carried out as part of the evaluation. In these experiments, the Traumschreiber proved to be a reliable source of information for EEG data.

The first attempt to create an application for portable EEG recording was developed by Jäkel (2017) as part of his bachelor's thesis. The goal of Jäkel (2017) was to develop an application that allows to handle the Traumschreiber data, introduce the system to participants, and help them to use it. This thesis established some of the design elements for the application followed by this work, for example: receiving EEG data, provide an easy-to-use application, include a starter guide that explains the app and how to use the Traumschreiber and make possible to save and export data. Nonetheless some difficulties were experimented by Jäkel's project to solve the wireless connection between the Traumschreiber and Android systems. Spite of these bugs, Jäkel (2017) was able to develop the first Android app that served as a ground for future implementations. In order to track the evolution of the implementation process, the version number 0.1 was assigned to Jäkel's app.

A second version (v0.2) of the app was developed by a group of Cognitive Science students as part of the Hack4Health Hackathon[2] celebrated from the 12th to the 16th of February of 2018 in Osnabrück. The working team was composed by Marc Vidal, Renato Garita, Andrei Achziger and Adrián Rojas-Campos. During this event, a new independent app was created[3]. This app could connect to an Android device, receive its signals, and plot them in real time. The first version of a chatbot expert in epilepsy (named later Epibot) was also developed during these days. This version of the app was improved during the next months by Vidal and Rojas-Campos, as a part of tasks as research assistants in the Neuroinformatics Department.

Parallel to this development, Kai Fritsch developed a Time-Frequency analysis as part of his Lab Rotation in the Neuroinformatics Department. The implementation and test of the analysis were developed using artificial data given that real data taken by the Traumschreiber was not available at the moment. This app is also included as a part of this project.

Some months later, the project was pushed forward with Marc Vidal's master thesis. With his work, Vidal (2019) improved the version 0.2 of the app, implementing a better and more detailed plotting of the brain waves and allowing saving of the recordings as CSV files, as well as adding the option to change the gain of signals when recording. This version number 0.3 of the app was included almost entirely in this work[4]. Some minor improvements were implemented to make the user interface language independent, using icons instead of words in some functionalities. In this work was also tested the app using an experimental setting, proving good evidence about the functionality of the app as a data collector in experimental settings.

The version of the app developed in this project is considered as 0.4.

1.4 Aim of this project

Considering previous works, the main goal of this thesis was to design, implement and test a general use Android app, that together with the Traumschreiber device, allows the user to record, plot, analyze, manage and share EEG recordings.

A big part of this project was to unify all different pieces of software produced previously which worked independently, in one single app that is able to perform all functionalities provided by previous and aggregate new capabilities.

The general idea behind the design of this app is that an average user can learn how to operate the Traumschreiber, how to record brain signals with it, manage the files (rename or delete them) and share them using a smartphone. At the same time, the user gets access to reliable information about different diseases that affect the brain (in this case Epilepsy) and use tools of Artificial Intelligence to chat with a robot expert in Epilepsy as part of the learning process.

[2]https://www.hack4health.de/

[3]https://github.com/StatefulMind/eegdroid/

[4]https://github.com/mvidaldp/Traumschreiber-mobileEEG

1.5 Thesis outline

In this document, we will present the process of development: the platform used to develop the app and the software development methodology followed. Then we will expose the results: the process of design and the resulting architecture, as well as details about the process of implementation. We will present a basic description of the functionality of every one of the 10 activities that form this app. For each activity, we will show its goal, followed by an explanation of how this goal was achieved. We will also display the interaction between activities, as well as the results of the testing phase. Finally, we will show the difficulties, shortcomings, achievements, and future steps from this work. As appendix, we included the content of the learning units included in the app.

2 Method

2.1 Definition of design guidelines

Previous to the design of the app, a set of main design guidelines was defined. This set of principles guided the process of design and implementation, helping on the making of important decisions during later stages of development. Before the definition of the guidelines, the documentation and material offered by Google for developers[5] were consulted, and it served as a base for the next steps.

The first important guideline was to produce an app that was easy to use. The potential public for the app was considered: research assistants, medical doctors, nurses, and patients, all form part of the potential group of users. Therefore, it was essential to develop an app that allows data collection without difficulties and without requiring expert knowledge or expertise with EEG.

The second guideline was to produce an app that was not only functional but also educational. The idea is that even if the user never heard before about EEG, he/she would be able to operate the app and learn basic knowledge about this technology, how it works and why it is important for the diagnosis of certain diseases.

As a third guideline, we considered relevant to reduce and limit the amount of text present in the app as much as possible. The intention was to make a friendly design that facilitates the process of translation to other languages.

The fourth guideline was to follow a goal-oriented design. A goal-oriented design means that the goal of the app was split between several different modules, and every module had a clear and defined goal behind it. With the work of all modules combined, the main goal of the app is achieved.

As the last guideline was to program the app in a modular way. This means that every module works independently from each other. Every module can be suppressed, modified, and improved independently from each other, as well as new modules can be integrated into the system without great effort. In Android apps, modules should be implemented as activities (from now on, we will refer to modules as activities).

2.2 Software development platform

The implementation of the app was made using Android Studio. Android Studio is the officially supported software for Android development. The programming language used was mainly Java, except for the graphical elements that are handled in XML. As part of the development process, the software was tested using a Nokia 7.1 device with Android 9.

[5] https://developer.android.com/docs

2.3 Software development methodology

The software development methodology followed by this project was the traditional waterfall model (Royce (1970)). Following the waterfall model, the first step was to examine the system and software requirements, later comes the design of the software architecture, followed by the implementation in code and the verification and test of the app as the last step. The last step of Royce (1970) model (migration, support, and maintenance of the system) was not needed because the app is not yet published or distributed between users.

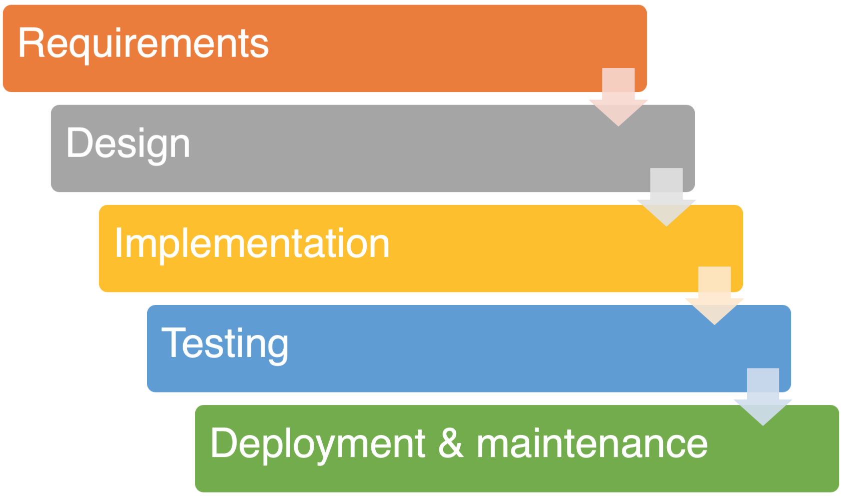


Figure 1: Royce (1970) Waterfall model

The first step was to determine the requirements and goal of the app and the design guidelines to follow. Then we proceeded to elaborate the architecture, according to the requirements and the guidelines previously defined. Once the design of the architecture was finished, the implementation process started.

2.4 Design of the app architecture

The requirements were defined as a series of goals to achieve with the app based on it's general purpose. For that, the general goal of the app was divided into several specific goals to be fulfilled.

As we presented in the introduction section, the main goal of the app is: "Allow the user to record, plot, analyze, manage, and share EEG recordings, as well as provide the user with useful and reliable information about Epilepsy and EEG".

This primary goal was subdivided into specific goals:

- Allow the user to record EEG signals provided by the Traumschreiber
- Allow the user to play EEG sessions previously recorded
- Allow the user to manage the EEG sessions: rename, delete and share sessions

- Present reliable information to the user about Epilepsy
- Present the user the necessary information about EEG and how to operate the Traumschreiber to run experiments
- Provide the user with a chatbot expert in epilepsy as an educational tool
- Allow the user to perform time-frequency analysis of previously recorded EEG sessions

Additionally, in order to improve the usability of the app, some additional extra elements were considered essential to implement:

- Allow the user to save some information (username, user ID number)
- Allow the user to choose the directory to save the EEG recordings
- Present to the user information about the app

Based on these goals, an architecture was designed. The functionality of the app was divided into three main categories: Utilities, Learn, and Extras, each one of them with three different activities. The Time-frequency analysis was placed under Extras, given its specificity and the fact that the average user might not use it very often. Following a modular design, most of the specific goals were assigned to one activity.

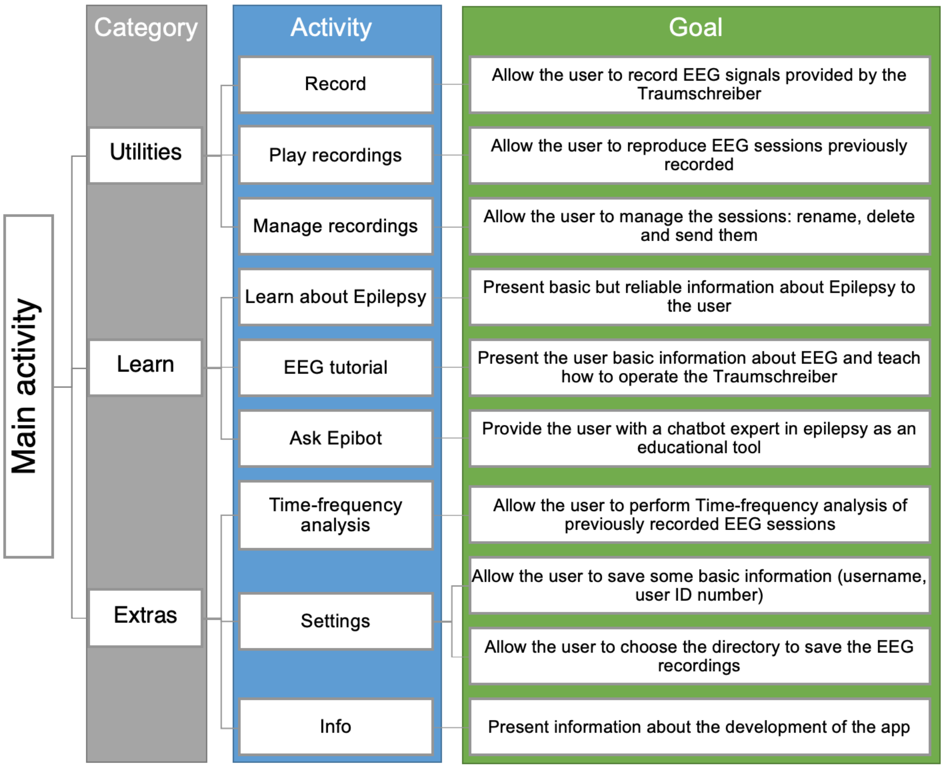


Figure 2: EEGDroid's architecture

One additional activity was added (main activity), to work as a central hub, welcoming the user and allowing the transition between the different activities, for a total of 10 activities.

2.5 Process of implementation

On Android, every module can be implemented as an activity. An activity requires the declaration of a new class that extends the superclass Activity. On each activity, the code is divided into different sections that are called depending on the stage of the activity life cycle. Therefore, the different functions and instructions in one activity needed to be split between the different sections of the activity life cycle. At the same time, every activity works together with an XML file that handles the graphical elements.

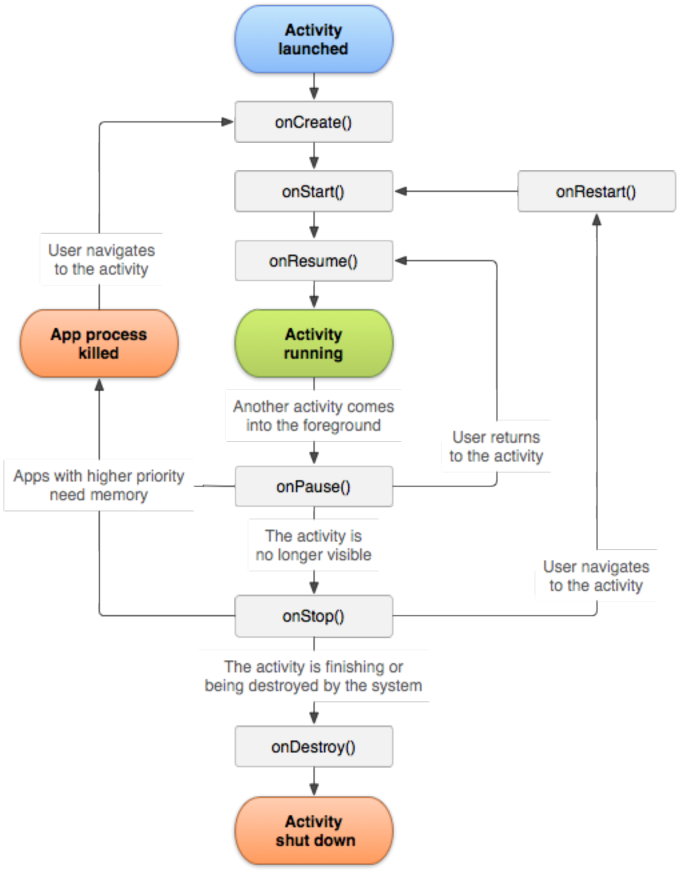


Figure 3: Activity's lifecycle. Taken from AndroidDocumentation (2019)

The complete implementation process of all activities required approximately four months. As part of this process, all different previous activities were collected. A new Android app was created with 10 independent activities and the previous existent software was adapted to the new activities (record, epibot, and time-frequency analysis). This process of adaptation usually required a deep understanding of the code written by others and implicated a significant amount of effort. Some modifications were made to adapt the old software to our design guidelines. For example, text buttons were removed and replaced by image buttons of language-independent meaning. Once all previous software was working correctly, the new activities were implemented (main activity, play, manage, learn, tutorial, settings, and info).

Once all the activities were running correctly, a revision of the graphical design was performed. The main goal of this revision was to make an effort to unify the appearance of all activities and to provide the app with a friendly and professional look. Changes were made in the colors of the app, the size of the text and space between elements on the screen.

As the last step, a series of test were run to the app to check its correct functioning. For example, it was essential to check that the app did not present crashes, that it ran smoothly, that it was able to perform extensive recordings without problems, that it was able to play long recordings within considerable loading time, as well as to run in different devices with different screen sizes. A detailed explanation of the tests and results is offered in the results section.

3 Results

The process of software development under the waterfall methodology produced an android app composed of 10 different activities, that together achieve the general goal of the project. In this section, we will provide a basic description of the app produced and the results of the test to which the app was submitted. Every activity it is accompanied by the name of the file inside the project. All the code written was saved and backed up using a GitHub repository and it can be found online[6].

We consider necessary to remark that here we present exclusively a final version of the software. The development of this app took many weeks and required a lot of research, exploration, and experimentation, until we found the best solution available. All that process of exploration and trial and error is not documented here, given that it does not contain a high scientific value, but it was a considerable amount of the time invested in this work.

3.1 App's name and icon

Previously to the implementation of the app, a prototype name needed to be chose for the app. We wanted to select a name that express the functionality of the app, but also shows the educative goal of it. Therefore, the name "EEGDroid" was selected. Also, a new logo was selected for the app. For the logo we wanted to create a simple logo with a minimalist design.

Figure 4: EEGDroid's icon

3.2 EEGDroid app

In this section, we will present all the activities contained by the app. For each activity, we will present the goal, the appearance, how it can be operated by the user, and a basic description of how it works. We follow the order in which the activities are presented in the slide menu of the app.

[6] https://github.com/adrocampos/EEG-Droid

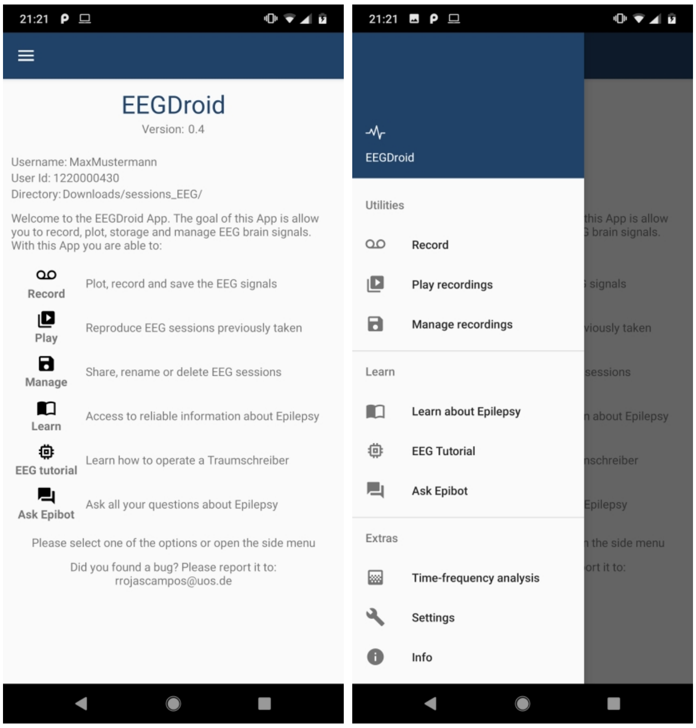
3.2.1 Main activity (*MainActivity.java*)

The function of the main activity is to serve as a central unit in the app. It is the default activity that runs when the app is launched and redirects the user to all the other activities that perform functions.

The main activity welcomes the user and displays essential information about the user configurable in settings activity (username, user ID, and directory to save or load the EEG sessions). It also provides the user with a short description about the function of every activity. For design considerations, it was essential to limit the amount of text on this page; therefore, the information offered is basic.

From the main activity, the user can run all other possible functions of the app, using the slide menu in the top left corner. Additionally, when the app is executed verifies the existence of a folder to save and load the EEG sessions. By default, the name of the folder is "sessions_EEG". In case that the folder does not exist, the activity creates it under the directory "Downloads".

Android provides two ways to save data for apps: internal and external storage. We consider that using external storage (the downloads folder) is more convenient than internal memory, considering that it makes the sessions easier to share with other apps or devices and protect them from being deleted if the app is uninstalled or corrupted.



(a) Welcome screen (b) Slide menu

Figure 5: Main activity

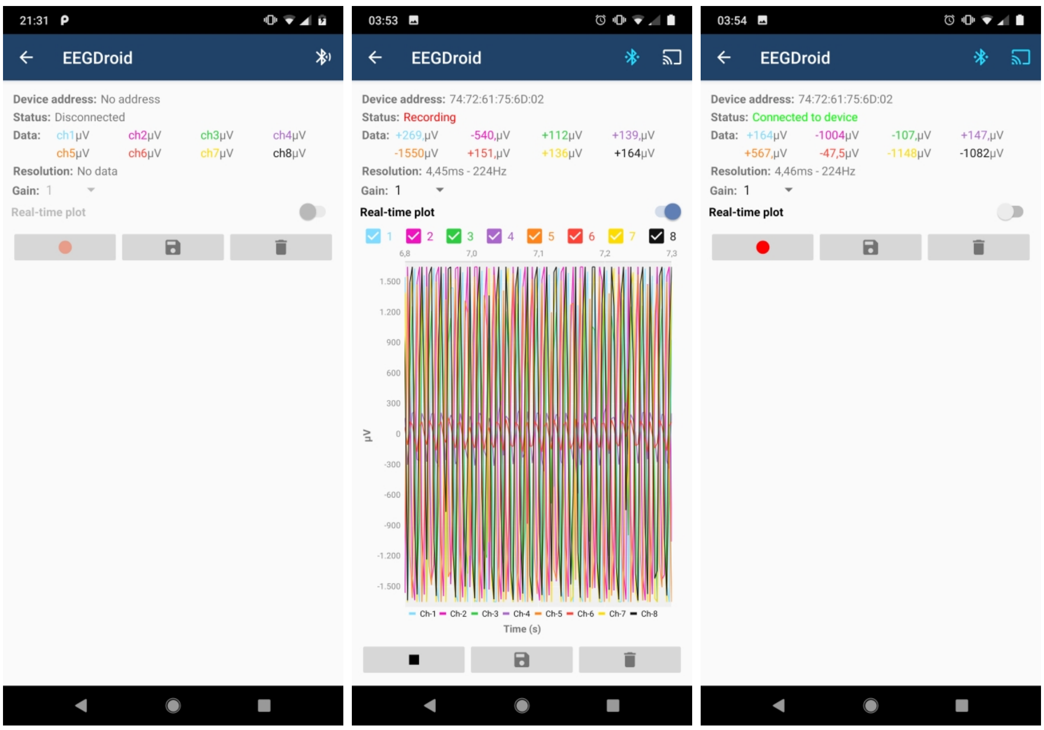
3.2.2 Record activity (*Record.java*)

The function of the record activity is to connect the app with the Traumschreiber, plot, and collect the incoming data transmitted by it.

The functionality of the record activity depends on two states: connected to the Traumschreiber or not. While there is no connection, the functionality is disabled, and the activity only allows to search for devices and connect. Once that a connection is detected, the incoming data is printed on the main page, and the other functionalities are available. The group of buttons at the bottom of the screen adapts dynamically depending on the state of the connection and if the app is recording or not.

To establish a Bluetooth connection, the record activity needs to call temporarily a new activity called DeviceScan activity (*DeviceScanActivity.java*). DeviceScan searches for all the Traumschreiber devices available and shows them in a list. Once the user chooses one Traumschreiber, the information about this Traumschreiber is sent back to the record activity, and record activity creates a service that streams the incoming data. If the connection with the Traumschreiber is interrupted, these options will be disabled, and the activity will go back to the no-connection state.

If data is being received (the Bluetooth icon should be light blue) and the record button is pressed, the data received starts to be collected in an *ArrayList* of floats. During this process, the recording is taking place. If the user returns to the main activity, the recording will be canceled



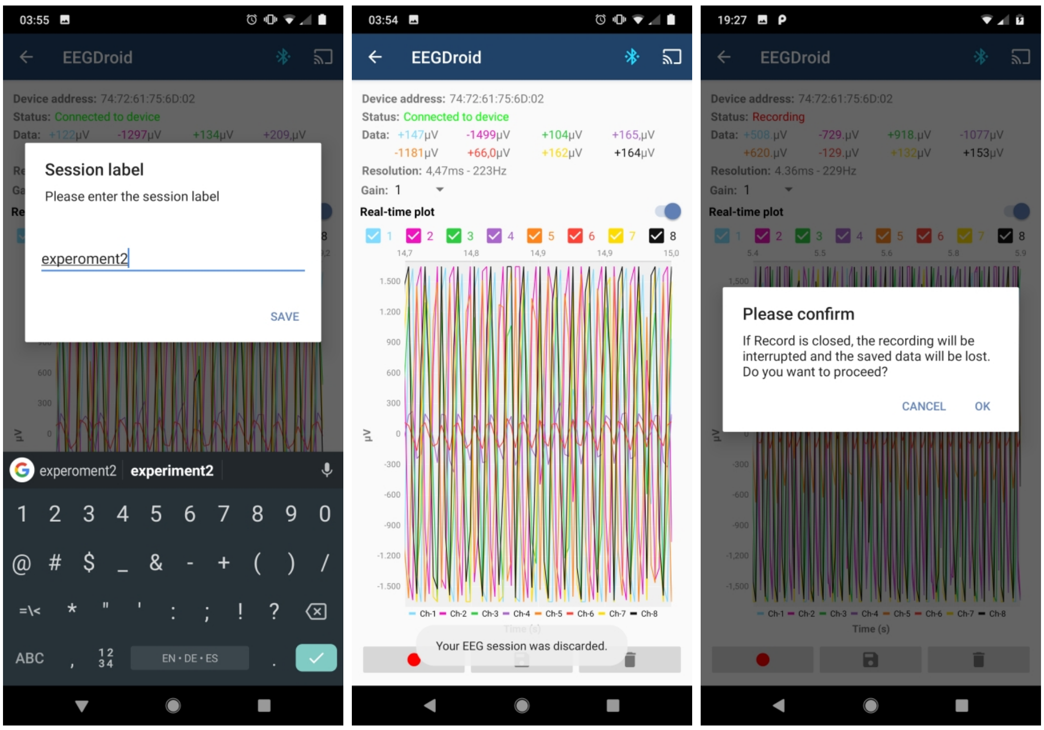
(a) Disconnected (b) Recording (c) Casting

Figure 6: Record activity: disconnected, recording and casting

(a warning will show up first), but the user can block the phone or make use of different apps and the transmission of data will keep going.

Once the device is recording, the record button turns into a stop button. If the stop button is pressed, the app stops recording the data and show two different options to the user. The user can discard the recording (for example, in case something went wrong with the electrodes, and the experiment must start again). If the user decides to discard the recording, the *ArrayList* with the data is substituted by an empty *ArrayList*, and the app allows to record again. On the other hand, if the user decides to save the current recording, the app will ask for a name of the data file. If the user introduces no name, "default" will be used as a name. Once a name is chosen, the data contained in the *ArrayList* is saved as a CSV file. The name of the CSV file is composed by the date and hour of the recording and the name provided. The CSV file contains in the second row all the critical information about the recording, including the username and ID number settled in settings. The file is created and saved under the direction chosen in the settings activity (by default: \Downloads\sessions_EEG).

Additionally, we implemented a casting function. Once a Traumschreiber is connected, a cast icon appears next to the Bluetooth icon. The icon disappears if the connection Bluetooth is interrupted. If the cast option is selected, the smartphone/tablet will start to cast the incoming data to a server using the WebSocket protocol. The IP address and the port of the server are



(a) Saving session (b) Session discarded (c) Disconnection warning

Figure 7: Record activity: saving or deleting session

specified in the settings. If the cast button is pressed while casting, the transmission of the data will be interrupted.

In this work, we test the transmission of the data using a Wireless Local Area Network and a basic Java server that receives the signal, converts the string into doubles and print the array of numbers. The code from the Java server can be found online[7].

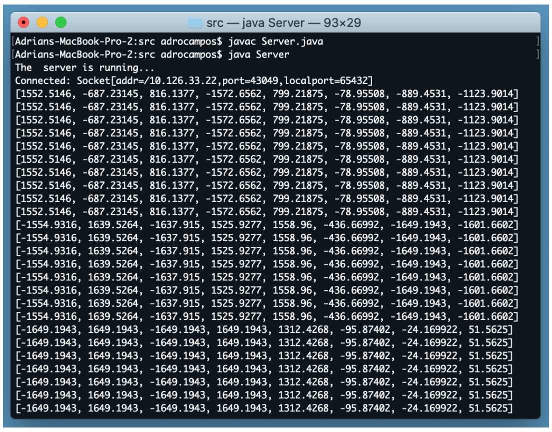


Figure 8: Record activity: server printing incoming signals

3.2.3 Play activity (*Display.java*)

The goal of the play activity is to load, plot, and reproduce saved EEG sessions. For that, the user needs to select one of the EEG stored in the selected folder. Once one file is selected, the data will be loaded into memory, and the respective plot will be generated. Horizontal screen orientation is forced to achieve better visualization of the plots. Given the high frequency of the recordings and the significant number of data points, the process of loading the data could take some time, depending on the speed of the processor and the size of the file.

[7] https://github.com/adrocampos/EEGDroidJavaServer

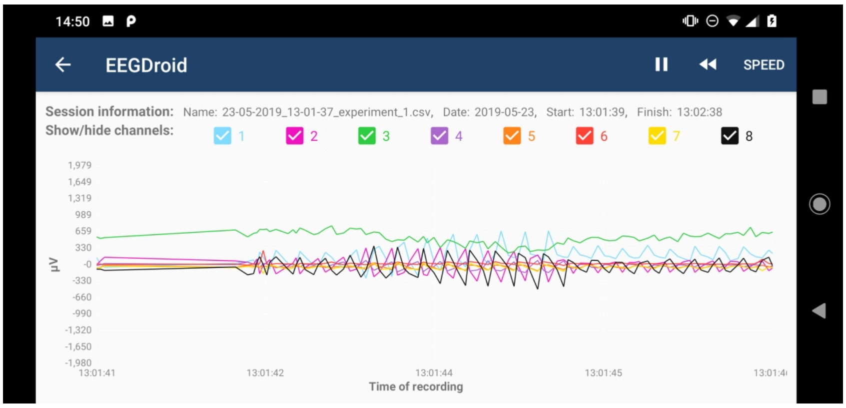


Figure 9: Play activity: reproducing a session

Once the data is loaded, and the plot is rendered, the user is capable of navigating the recording using the touchscreen. Similarly to the record activity, the y-axis represents the microvolts and the x-axis time. In this case, the x-axis starts at the time when the recording started, and it extends until the recording stopped, showing a window of 5 seconds on screen. If the user touches of the data points on the screen, detailed information about this data point will be shown for some seconds.

If the play button is touched, the EEG session will start to reproduce like a video in real time. To play recordings, the app needs to create an independent thread that handles the movement of the plot through the x-axis. Once the activity is reproducing the session, the user can pause it (this will interrupt the execution of the thread and save a checkpoint), continue playing (starts a new thread from the checkpoint) or rewind the session (kills the thread and moves the window to the beginning of x-axis).

The user is also able to configure the speed of the reproduction of a session. Seven different options are offered (x0.25, x0.5, x1 (real time), x2, x4, and x8). The speed of reproduction depends on two variables: *jump_x* and *period*. *jump_x* is the distance in the x-axis that the screen shift to the right every unit of *period*, and a *period* is the number of *jump_x* performed in one second. Therefore, the real-time speed is the result of $jump\_x = 1$ and $period = 1$. Analogously, the change in the speed depends on the change of these two variables: the slow motion speed x0.5 results from $jump\_x = 1$ and $period = 2$ (the shift of one second in the x-axis will take two seconds of real time) and the fast speed x4 works thanks to $jump\_x = 4$ and $period = 1$ (shift of four seconds in the x-axis per second in real time). The variables *jump_x* and *period*, serve as arguments that modify the rate of execution of lines of code on the thread.

It is important to add that not all data points are displayed in the plot. This configuration is controlled by the variable $density = 10$, which means that just one value every ten is used in the plot. This consideration was made given the limited hardware resources available on portable devices and the high density of information contained in the file.

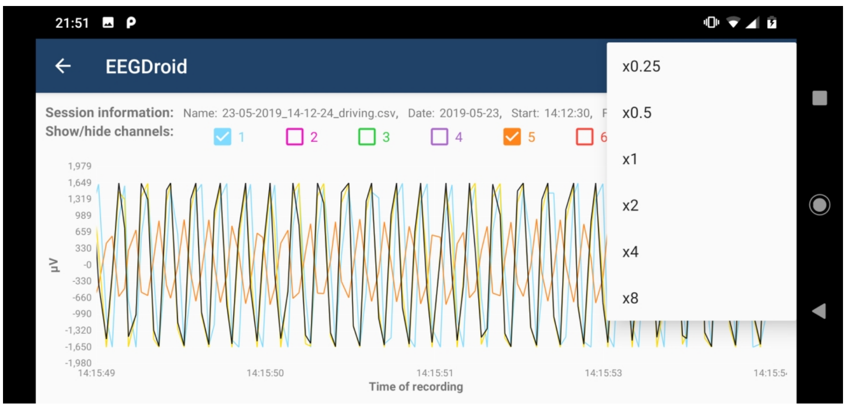


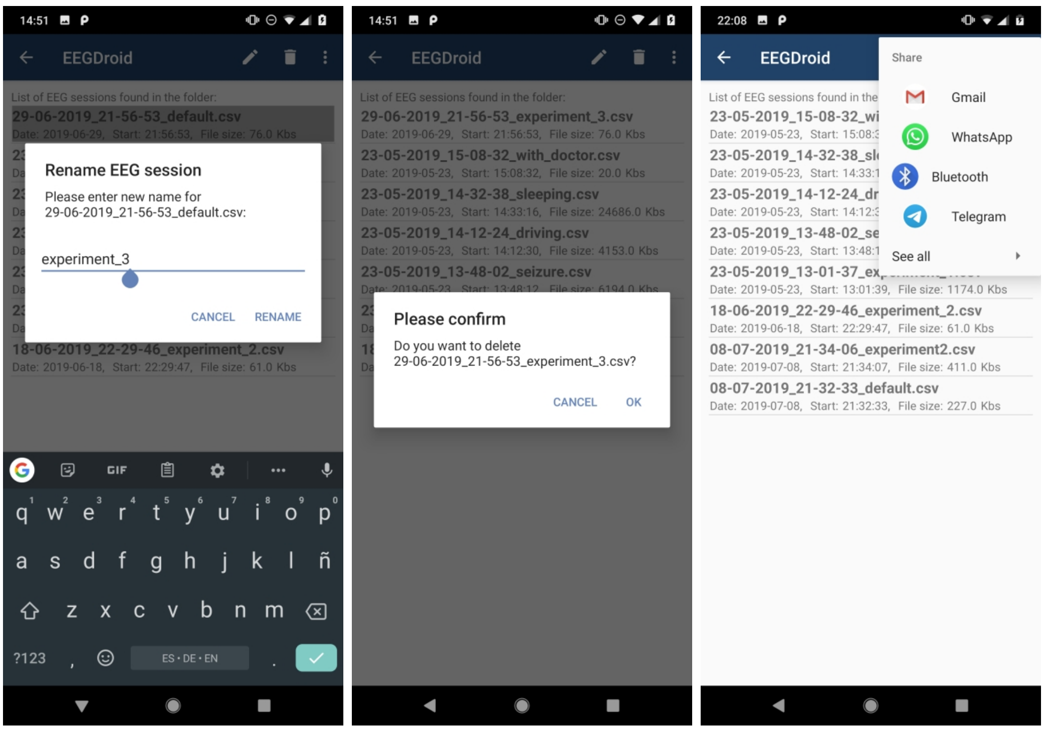
Figure 10: Play activity: different speeds and channels

3.2.4 Manage activity (*ManageSessions.java*)

The function of the manage activity is to allow the user to manage the EEG sessions stored in the device. Manage the sessions means that the user can rename, delete, or share them. When the manage activity is launched, it reads the external directory selected in settings (by default: \Downloads\sessions_EEG) and fills a list with all the CSV files founded in the folder. For each of the files, different information is read from the metadata and display: the name of the file, the date of creation, the time when the recording start and the file size. The files are presented alphabetically.

To display the list of files, a new class named *SessionAdapter* was defined. The function of the adapters is to configure how every element on the list will look and which information will be display. The *SessionAdapter* works together with *session_row.xml* that defines the rendering of every element on the list. From the list of files, the user can select one. When a file is selected, the row in the list will turn gray. If a file is selected, the app allows the user to perform different tasks over this specific file.

In the first place, the user can rename the file. If the user chooses this option, a new name will be asked using a dialog window. Once the new name is introduced, the activity will change the name of the file and refresh the list of files in the screen. The user is not allowed to change the first characters from the name: date and hour of creation. Only the last section of the name



(a) Rename function (b) Delete function (c) Share function

Figure 11: Manage activity

is available to change. Also, the user does not need to introduce ".csv" at the end of the name to prevent a change in the file format, because the activity includes it by itself.

The manage activity also allows deleting EEG session files from the directory, without the need to explore the directory with additional apps. For that, the user needs to select one of the files and press the delete button in the action bar. Once the user presses the delete button, a confirmation message will appear. If the user wants to proceed with the deletion, he/she must press "ok" in the confirmation. If that is the case, the activity will proceed to delete the file.

Finally, the managing activity allows the user to share the CSV file using different kinds of applications. To achieve that the user needs to select one of the recordings and press the share option under the menu. A list of possible apps will be displayed according to the installed apps in the device. Between the sharing possibilities, we can find Bluetooth, SMS, and email, and many others.

3.2.5 Learn activity (*Learn.java*)

The primary function of the learn activity is to provide the user with essential and reliable information about Epilepsy in a friendly and easy-to-understand way. To achieve it, two different development stages were needed. First, a review of the literature was made with the goal of collect necessary and updated information about this syndrome. All this information was compiled in a single markdown file. The second part of the task was to find a way to display this information in the app that it is also easy to explore and update.

The content offered to the users is exposed in a FAQ (frequently asked questions) format. We offer information to answer the next questions:

- What is a seizure?
- What is not a seizure?
- What type of seizures are?
- What causes seizures?
- What is a provoked seizure?
- What is epilepsy?
- How can epilepsy be diagnosed?
- What are the mechanisms of epilepsy?
- What are the risk factors for epilepsy?
- How can epilepsy be treated?
- How many people suffer epilepsy?

At the end of the document, we offer references to diverse web pages specialized in educating about the topic. All the content present in this section can be found in the appendix or online[8].

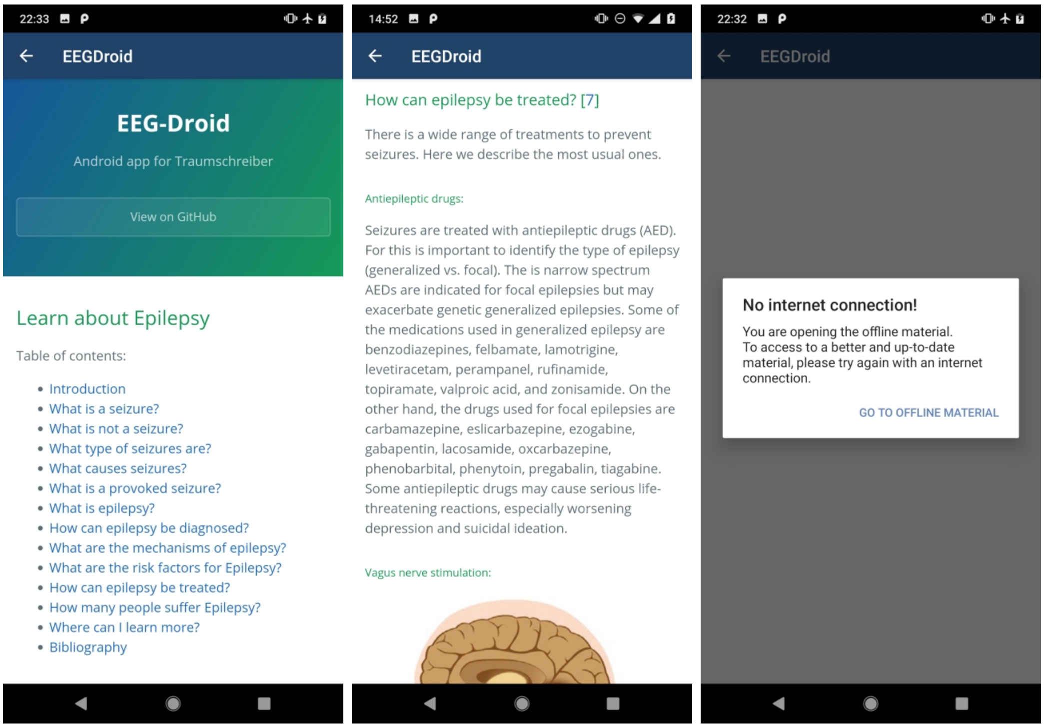
It is necessary to consider that creating an extensive revision of the literature about epilepsy and all the surrounding topics is a demanding task that is out of the scope of this thesis. Therefore, we limit the present work to a proof of concept, showing one possible way to present learning material and presenting some information as an example.

Once the content of the document was filled, the question of how to embed inside the app appeared. For this, it was especially important that the content had a nice and clear appearance, that it would be easy to navigate through the page and easy to update. The solution founded was to use a simple *WebView* as the only element of the activity, together with GitHub Pages and Jekyll. In this way, all the documentation found in as part of the GitHub (including the markdown files with the content) is also offered as HTML files as part of the website of the project.

The only job of the activity is to load the webpage using *WebView* and display the content. In this way, to update the content showed on the activity is only needed to update the markdown file in the GitHub repository.

In case that the user has no internet connection, a local copy of the web page is loaded from the assets of the app. If the activity detects no internet connection, it will warn the user before

[8] https://adrocampos.github.io/EEG-Droid/learn.html



(a) Index of contents (b) Information about epilepsy (c) Offline content warning

Figure 12: Learn activity: index, content and offline warning

opening an offline version of the content. This offline version looks similar to the one online, but all the links will get to online resources. Another problem with the offline version is that it only can be updated when the app is updated. Therefore we insist on the user the importance of accessing the online material.

3.2.6 Tutorial activity (*Tutorial.java*)

In the same way as the learn activity, the primary purpose of the tutorial activity is to offer the user information about the Traumschreiber, EEG, and its importance for the diagnostic of diseases as epilepsy. An important goal is to teach the user how to use the Traumschreiber to record EEG signals, how to place the electrodes and under which conditions the experiments should be done. To prepare the information, a bibliographical revision was made. The content was collected in a markdown file in a FAQ format, with the following questions:

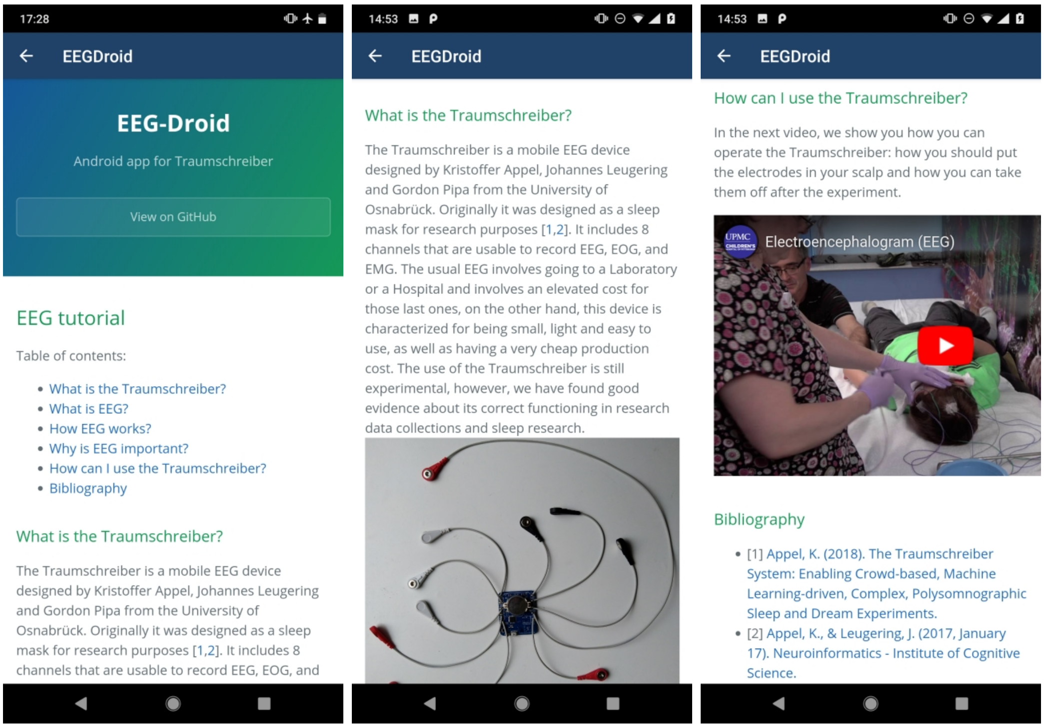
- What is the Traumschreiber?
- What is EEG?
- How EEG works?
- Why is EEG important?
- How can I use the Traumschreiber?

To display the information, the same technique as in learn activity was implemented: *WebView* that reads an online source managed by GitHub and an offline copy in case of no connection. The content can be accessed online[9] or in the appendix of this work.

Unfortunately, we were not able to answer the last question: "How can I use the Traumschreiber?" After looking at previous works that have used the Traumschreiber in experimental context (Appel (2018); Vidal (2019)) we considered that the configuration of the electrodes is highly dependent on the specific experiments to be performed, and for that, we need a specific experimental setting that is lacking in this work. However, we believe that using instructional videos to teach the user, in the way that Appel (2018) did it in his work, is a potentially adequate idea to implement in the future. For this reason, we embedded temporally a YouTube instructional video about epilepsy, that eventually should be replaced by the adequate video that teaches users how to use the Traumschreiber.

Again, replacing this video by another or any other change of the content can be easily made by applying the changes in the markdown file and submitting the changes in the GitHub repository.

[9] https://adrocampos.github.io/EEG-Droid/tutorial.html



(a) Index of contents (b) Content (c) Embedding of tutorial video

Figure 13: Tutorial activity

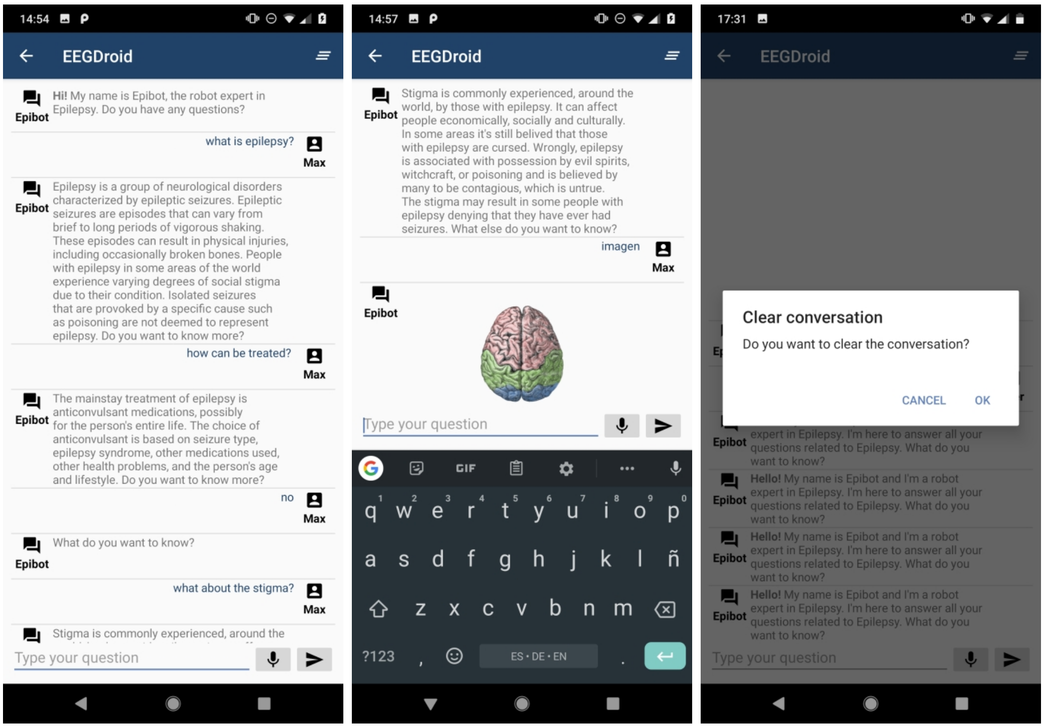
3.2.7 Epibot activity (*Epibot.java*)

Considering the lack of experts and the vast penetration of technology in developing countries mentioned in the first chapter, we considered necessary to use artificial intelligence tools to provide a virtual expert in epilepsy. One possible solution to this problem is to create a chatbot filled with specialized knowledge about epilepsy and EEG that works as an interactive way to inform and teach. The implementation of the chatbot was out of the limits of this work. Therefore we decided to use IBMs technology called Watson Assistant[10]. Watson Assistant is a service to build conversational interfaces part of IBM's Watson project. The chatbot can be feed through a website where the developer can complete the information of the knowledge base.

The first version of this activity was developed as an independent app, as part of the work in the neuroinformatics department. The chatbot was established and filled with information from different online sources. In this work, we integrated this independent app to our app and made some modifications to the functioning and graphical interface of the chatbot.

The Epibot activity handles the embedment of the chatbot services to Android. At the background, it works with an *ArrayList* of messages that are displayed on the screen. For that, the app has to take the input of the user and communicate with the IBM services, send the input, and wait for the answer. Once the answer from the services is received, it is saved and showed in the

[10] https://www.ibm.com/cloud/watson-assistant/



(a) Text answers (b) Image answers (c) Clear conversations

Figure 14: Epibot activity

screen as part of the conversation. For this reason, the epibot activity always requires an internet connection. However, while the activity is closed, the conversation is saved in the internal memory of the app and is loaded again when the activity is reopen, in this way the user can access to old conversation even if they do not have internet access at the moment (the possibility of sending messages will be disabled).

In the same way as the manage Activity, a particular class of adapter needs to be called to display the *ArrayList* in the screen. For this reason, the Message Adapter class is defined (*MessageAdapter.java*) that, together with *adapter_view_layout.xml* defined how a message in the list is presented. Here we manage to change the color and the direction of the text depending on who is the author of the text. The user also has the option to clean the conversation if wanted, by pressing the button in the top right corner. By clicking this, the *ArrayList* of messages is erased and substituted by an empty *ArrayList*, and the view of the screen is refreshed.

Some extra features were implemented, with the goal of make the conversation similar to the chat applications widely used. A vibration function was implemented, and it is activated every time that the user receives content from Epibot. A delay function was also implemented, that delay presenting the messages in the list in some fractions of a second according to the length of the message, to give a human-like interaction with the user (as if someone were typing). However, this function was deactivated and commented on the final version of this app.

3.2.8 Time-frequency analysis activity (*TFAnalysis.java*)

This activity was developed by Kai Fritsch as an independent app, as part of his Lab Rotation in the Neuroinformatics department. The main goal is to analyze the EEG measurements and display time-frequency representations a to the user. A time-frequency representation helps to study a signal in time and frequency domains simultaneously, and it is produced by time-frequency matemathical transformations (Cohen (1995)).

Given that the app to record EEG signals using Android was not available yet, Fritsch developed this activity using an artificial measurement as a CSV document that was read from the assets of the app. For this reason, the main work on this activity was to implement the procedure of select one of the EEG sessions in the directory and to adapt the process of leading the data to the CSV format defined.

To achieve the selection of an EEG session, we used the same strategy as in the play activity. We read the directory and display a dialog with the list of CSV files founded under the directory. Once the user picks one, the time-frequency analysis is performed using this information.

When one file is selected, the data is loaded into the memory, and the analysis is performed (this task can take some seconds, depending on the length of the file). Once the analysis is performed, the results are shown on the screen using an *ImageView*. The user can change different settings: channel, time, and overlap, according to the type of information that wants to access.

In the process of testing, we discover that if the EEG session was too short (around 5 seconds) and contained only a few data, the process of analysis failed because a division by zero was performed. Considering that recordings of only 5 seconds will not be frequent, a try-and-catch exception was added. If the recording was too short, the activity does not crash but returns to the main activity and informs the user.

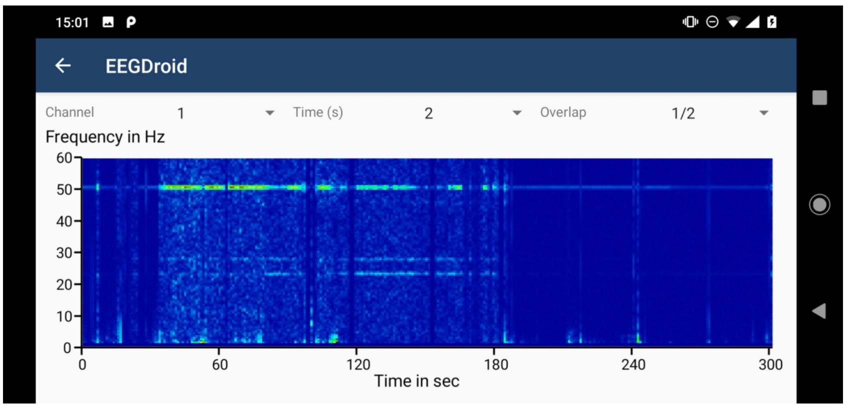


Figure 15: Time-frequency analysis activity: results

3.2.9 Settings activity (*Settings.java*)

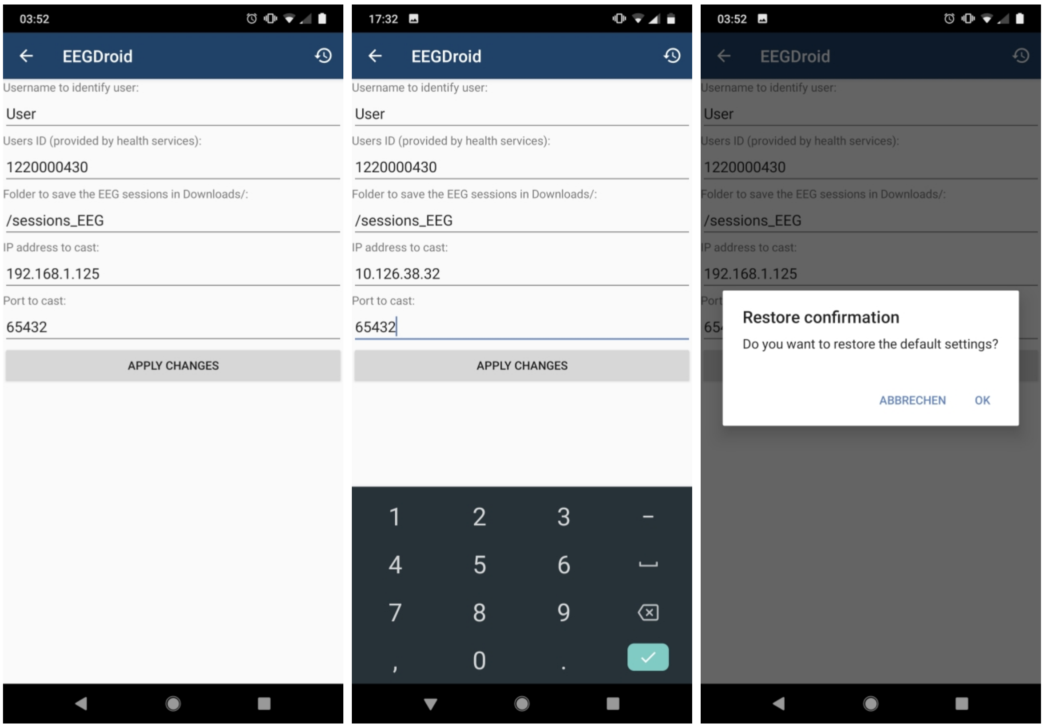
Considering that in an experimental setting different subjects may be tested, we considered essential to include information about the user as part of the EEG session. For that, the apps bring the possibility to include a username and a user ID, and this information will be included in the CSV file.

Username	User ID	Session ID	Session Tag	Date	Shape (rows x colum	Duration (ms)	Starting Time	Ending Time	Resol
MaxMustermann	1220000430	b70d5e5f-9264-4a0b	experiment2	08-07-2019_21-34-0	4517x8	105843	21:33:27	21:33:48	
Time	Ch-1	Ch-2	Ch-3	Ch-4	Ch-5	Ch-6	Ch-7	Ch-8	
0	-1649.1943	1100.5371	-496.28906	21.75293	15.307617	506.7627	230.41992	1111.8164	
3	1255.2246	-1326.123	1410.7178	-1462.2803	1463.8916	1649.1943	-1649.1943	1649.1943	
5	1620.9961	-1625.8301	1637.915	-1285.0342	955.5176	626.001	-1649.1943	-667.08984	
53	-1295.5078	1413.9404	-1384.9365	1577.4902	-1613.7451	-1382.5195	1265.6982	-1541.2354	
59	-1649.1943	1649.1943	-1649.1943	1649.1943	-1649.1943	-1635.498	1618.5791	-299.70703	
63	571.2158	-790.35645	992.5781	-1131.1523	1140.8203	1490.4785	-1285.0342	1649.1943	

Figure 16: Username and ID in .CSV file

The user is also able to change the directory where the EEG sessions are being stored (but always as a subdirectory of the Downloads folder), the IP address of the server and the port for the data casting. To all information, the apps offer initial default information that can be changed by the user. Once the user introduces new information and clicks the button ”Apply changes” the information is stored, and the app restarts using the new information.

Instead of saving the variables as fields of a class or part of the main activity, Android provides



(a) Settings (b) Modify values (c) Restore default values

Figure 17: Settings activity

the opportunity to save small pieces of information as key-value storage that can be accessed from any activity inside the app. Using the function *sharedPreferences*, we save the information of the user in the Main Activity and using settings the user can modify the information stored in *sharedPrerefences*. Finally, the user is also able to reset the default values for the different variables. Pressing the button on the action bar, in the top right part of the screen, the default values are restored in the text fields. To make the changes to be valid, the user should press "Apply changes".

3.2.10 Info activity (*Info.java*)

The goal of the Info activity is to provide the user with some basic information about the app, the project behind the app and the people who create it. For this, we use the same implementation as in the learn activity and tutorial activity: a *WebView* that reads an online markdown file.

3.3 Tests

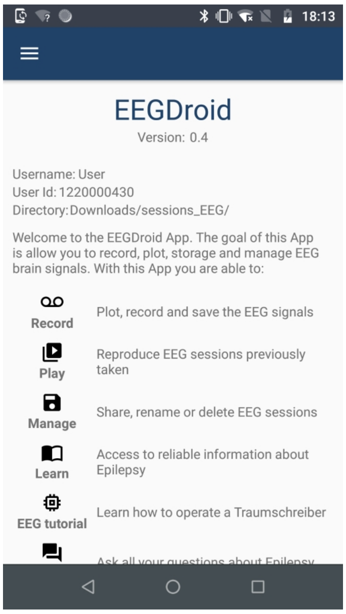
As a part of the software development process, testing the app as a whole, and the functioning of the different activities was necessary. It is important to remark that during the implementation process, there was a constant process of testing and fixing problems, standard in every software development. Most of the bugs in the code were solved during the same coding. The goal of this testing phase was to find problems and errors related to the whole functioning of the app and the interactions between the different activities.

The testing consisted of two stages: test the functioning of the app in different hardware and simulating the typical, expected usage of the app. The first stage was performed using three different devices: two smartphones: Nokia 7.1 and Moto G5S Plus, and a tablet Huawei Mediapad T5. This part of the testing put his emphasis on the graphical user interface.

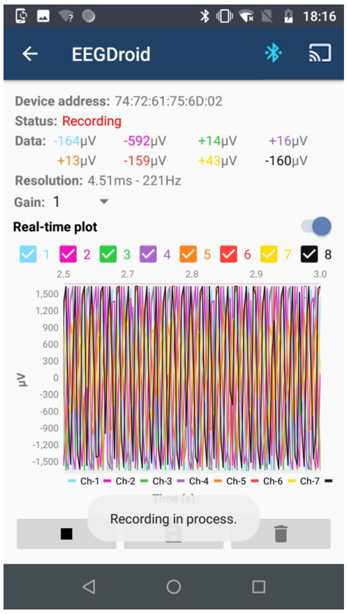
The goal was to assure the graphical user interface look similar, independent from the size of the screen specific to each device. During the process of testing, some problems were discovered regarding the size of the elements in the screen. Given that the developing process was carried out using the same device (Nokia 7.1), some of the settings of the apps were adapted to this device and were not relative according to the size of the screen. For example: when we tested the app using a device with a smaller screen, we realize that some information on the main activity was left out of the screen and it was not possible to down scroll the page.

This first stage of testing showed a series of bugs and problems that were corrected afterwards. The final result is an app that presents a GUI adaptable to the size of the screen of the device where it is running.

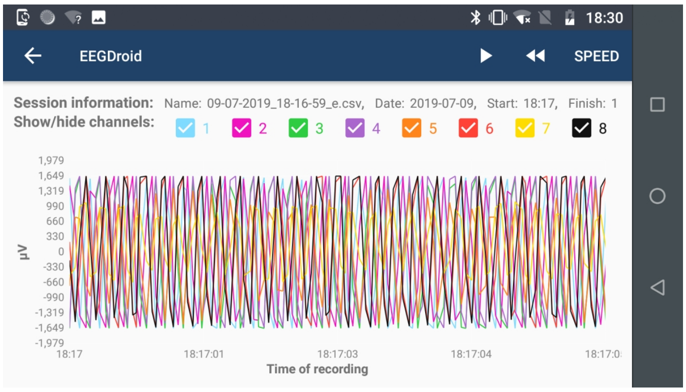
The second part of the testing phase was to simulate the expected usage of the app from the user, as well as some possible unexpected behaviors (for example, closing the record activity while recording). We also defined random sequences of different uses of the app and performed on the device (for example: learning for 4 minutes, recording for 7 and then playing the recording at x8



(a) Main activity (scrollable)



(b) Record activity

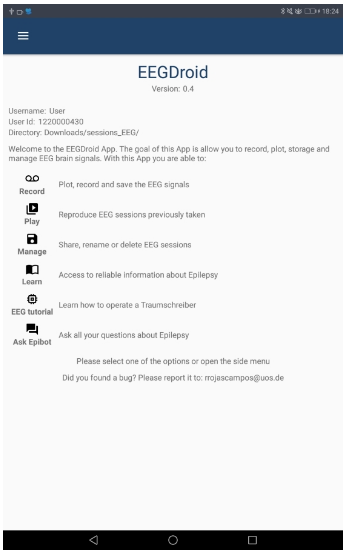


(c) Play activity

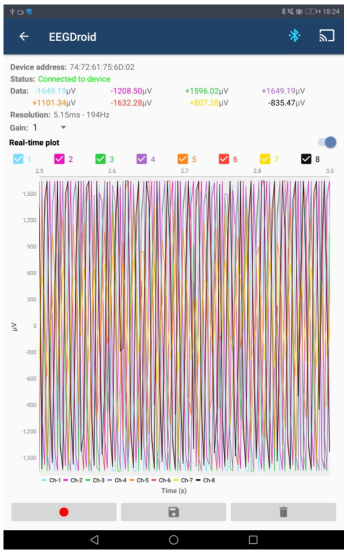
Figure 18: EEDDroid running in MotoG5S Plus

speed). As a final step, we test the transmission of the data and the capability of the app to save the data, making recordings of different duration (in minutes: 1, 5, 10 and 20).

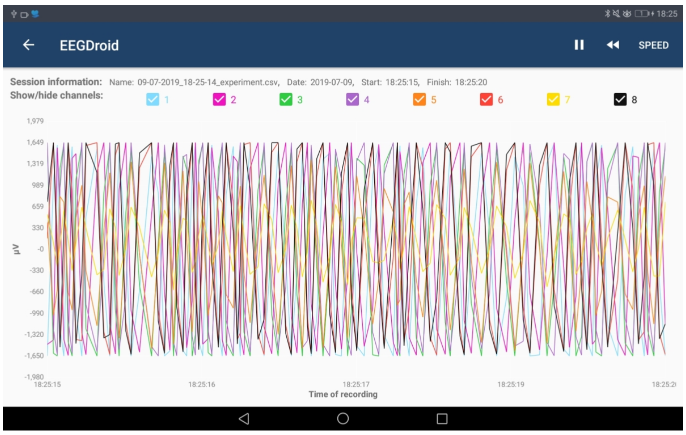
From this second stage of testing, no major problem was detected. The app works fluently, and there were no crashes during the functioning. However, we recognize that the testing executed did not follow a rigorous methodology and the quality of the app needs to be assured testing it under "hard" conditions: low RAM, no space in the hard drive, problems with connectivity, and others — more about this in "Future steps" in the next chapter.



(a) Main activity



(b) Record activity



(c) Play activity

Figure 19: EEDDroid running in Huawei Mediapad T5

4 Discussion

In the next pages, we offer an overview of the work performed for this thesis. We make a balance between the difficulties, weaknesses, and strengths. In the end, we present improvement possibilities and future steps that we consider essential during the next stages of the project.

4.1 Difficulties faced

The main difficulty faced during the development of this work was the little amount of time available to perform the work. The process of implementation took approximately four months, which limited the time available for all the other planned phases. The extension of the period dedicated to implementation was caused by two main reasons: the different pieces of code written by different people often demanded a significant amount of time to be understood and adapted, mainly because of the lack of clear documentation and the high complexity of the tasks performed. The second reason is that the lack of a computer science background from this author caused that the process of research and creation of solutions sometimes took more time than usually needed.

A minor difficulty faced was that the hardware used to implement the app was not up-to-date technology. The MacBook Pro 2012 used for the implementation worked fine, but on some occasions the resources demanded by Android Studio caused overheat of the laptop and long waiting time to execute the changes in the app. This situation might have caused an increase in the time required for the implementation.

A third difficulty founded was to define the amount of detail about the software to present in this report. This task was challenging because the implementation of the app is highly complex and requires basic knowledge about the Android development platform. Finding the right point between bringing detailed information and make an enjoyable-to-read text was a big question, and we are not sure of having achieved it correctly.

4.2 Shortcomings

The first and most crucial limitation from this work is the lack of a rigorous testing methodology for the app. The testing carried out was rudimentary and did not expose the app to usage in other conditions as low memory, lack of space in the hard drive, low energy, and others that can be experimented in the regular usage of the app. A more extensive and detailed testing process needs to be performed, in order to find and fix bugs that could appear during those conditions and with that assure the quality of the software.

The second most notable limitation is the graphical interface. Even if the functionality of the app works correctly, the graphical design of the app is still far away from the look of a professional app. This was provoked because of the functionality of the app was prioritized over the design, plus the lack of knowledge and practical skills about graphical design from this author.

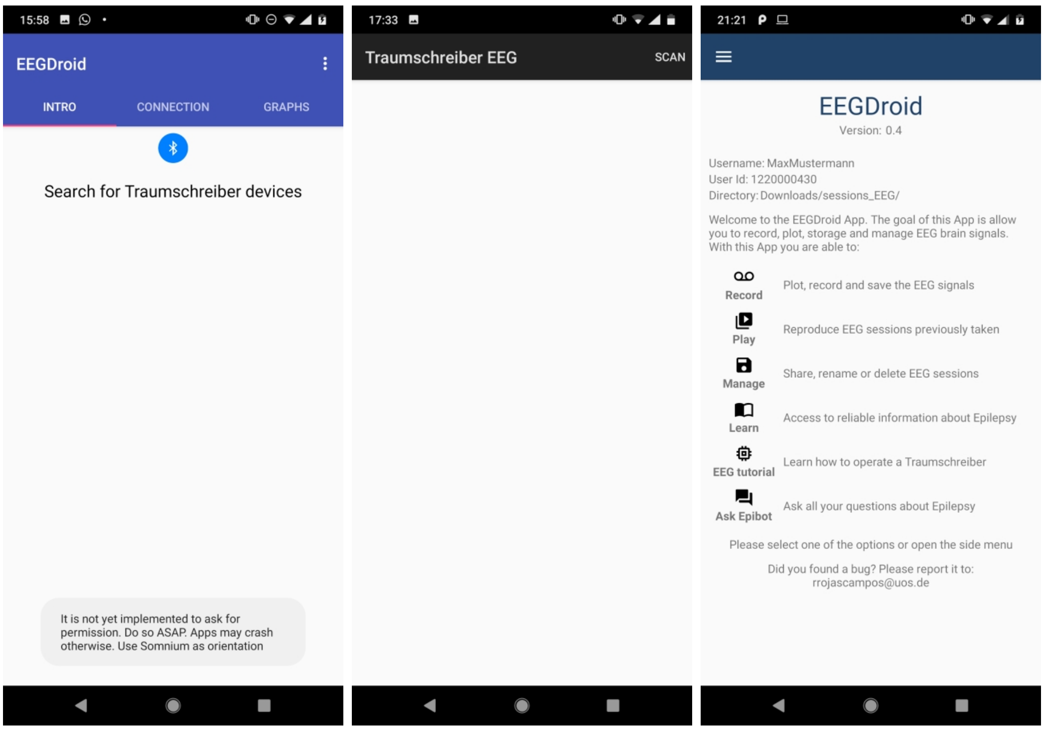
A third limitation is that the app is not accessible to the population with perceptual impairments. Most apps today are designed in a way that adjusting some parameters in the settings allows the look of the app to adapt to the needs of the user, for example: given the opportunity to people with visual impairments to using the app with a bigger font or highest contrast.

Another limitation that needed to be mentioned is the limited freedom provoked by the usage of CSV files as the format to save recordings. Saving and managing the data collected in CSV files causes several difficulties when we want to extract information about an EEG session. Most of the information is contained inside the document in the second row, which requires to open the document or to read the metadata of the document that is usually limited.

As a limitation number five, we point at the limited analysis possibilities offered by the app. The only possible analysis is the time-frequency analysis, and we do not offer the user tools to understand the results of this analysis. Using the app, a user can record the EEG sessions, but the information that can extract from them is minimal.

Additionally, we consider that the amount of information provided in the learn and tutorial activities are not sufficient. The information presented comes from reliable sources but is not extensive. Also, we limit the educational goal to the presentation of information, and we never check how good the users acquire the information.

Finally, but not less relevant, we are not satisfied with the level of innovation and original



(a) v0.2 (b) v0.3 (c) v0.4

Figure 20: Comparison Main Activity between versions

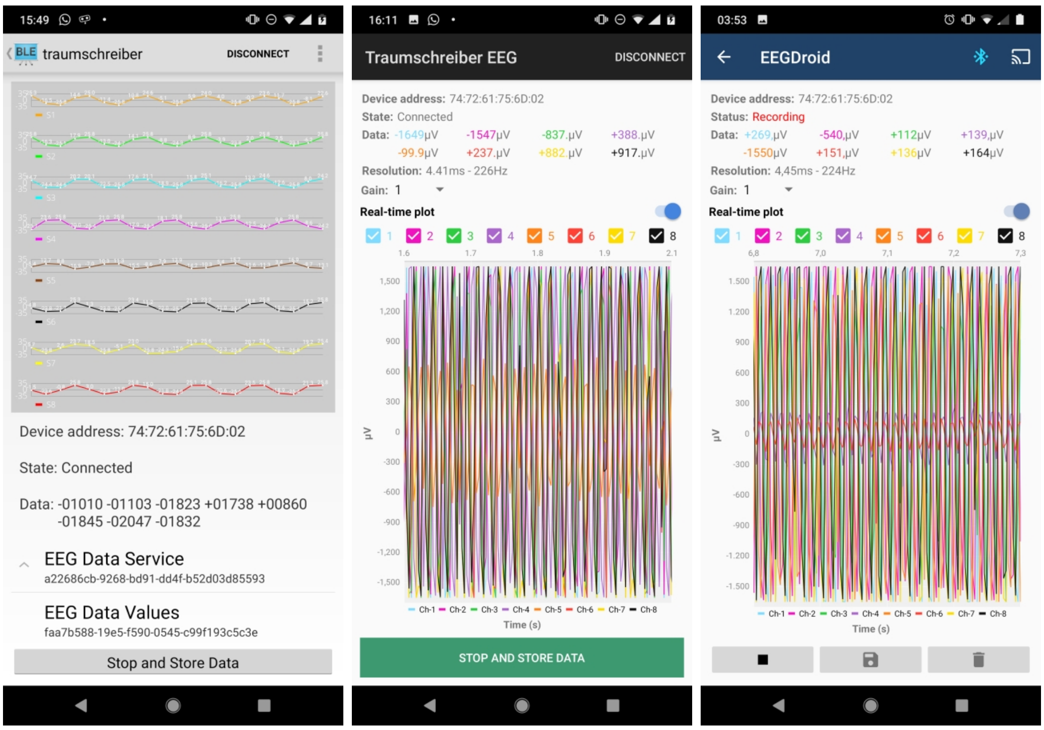
content offered in this work. Most of the technical work consisted of integrating software made by other coders in a way that works together. Without a doubt, this was a challenging, meaningful, and needed assignment to be performed, but the original contribution from this author was modest.

4.3 Achievements

Besides all the already mentioned shortcomings of our app, this work achieves its primary goal of design, implement and test a general use app, that allows the user to record, plot, analyze, manage and share EEG recording, as well as to learn some basic concepts about Epilepsy and EEG. All different pieces of software are integrated into a single and stable app that works without problems and achieve all the specific goals defined in the architecture.

The design of the app is less language dependent, and icons of universal meaning substitute some of the text from the previous software. The task of translating the app to another language is now reduced to change the content in the document *strings.xml*, allowing eventually to have several documents of strings and selecting the desired language when the installation of the app is taking place.

Even if the graphical interface of the app is not equivalent to those developed by software companies, we believe that this app fulfills his role as proof of concept, showing the feasibility of the project and its practical potential.



(a) v0.2 (b) v0.3 (c) v0.4

Figure 21: Comparison Record Activity between versions

Additionally, one strong point from this work is the definition of future steps in the development process. Given that the complete process from conceptualization to testing of the app was made, we can point the main flaws and provide ideas about how these can be solved.

This work is one solid step between many, in the project to create a good quality app that helps to crowd collect data from patients with epilepsy, help the patients to understand better their disease and medical doctors to establish a reliable diagnostic in economically deprived contexts.

4.4 Future steps

Based on the shortcomings of our work, we present different possible improvements that can be implemented in the future as part of the project. The first element to improve is to submit the app to a rigorous procedure of testing and quality assurance. The app needs to be tested in more and different devices. Additionally, an alpha testing procedure can be performed between a reduced number of participants.

The appearance and graphical user interface need to be improved. Right now, the app has a rudimentary interface that allows functionality, but there is no professional graphic design behind it. Before the app is distributed between users, it is vital to have a close look at the design of the screens and how these could be changed and improved.

Another critical element is to include accessibility options as a part of the settings of the app. Modifying the size of the text, background color, or implementing a voice-reading tool is possible in an easy way thanks to inbuilt tools provided by Android.

One possible solution to the limitations caused by the CSV files is to implement an object-oriented solution. We believe that the implementation of a different type of representation in memory to the recordings can be useful and can allow more flexible management of the recordings and information extraction. Concretely, we think about creating a new class that represents a recording of EEG sessions and has fields with relevant information as the time of start, time of finish, duration, username, type of experiment and even GPS localization while recording. To be able to share the information, it is necessary to implement methods to export the recording as CSV or JSON files.

Regarding Epibot, we consider that the content of the chatbot can be improved with information only from scientific sources. One possible solution is, instead of using an activity to chat, redirect the user to an already utilized messages app and implement the chatbot there. For example: implement a Telegram account for the chatbot. This solution has the advantage that the chatbot can be used without to need to install the EEGdroid app and therefore could be accessible to more people.

Furthermore, different types of data analysis can be included in the app to allow the user to extract information from the EEG sessions. This process could require the training of algorithms to analyze the incoming streaming of signals, and therefore, only can be implemented when there is enough information collected.

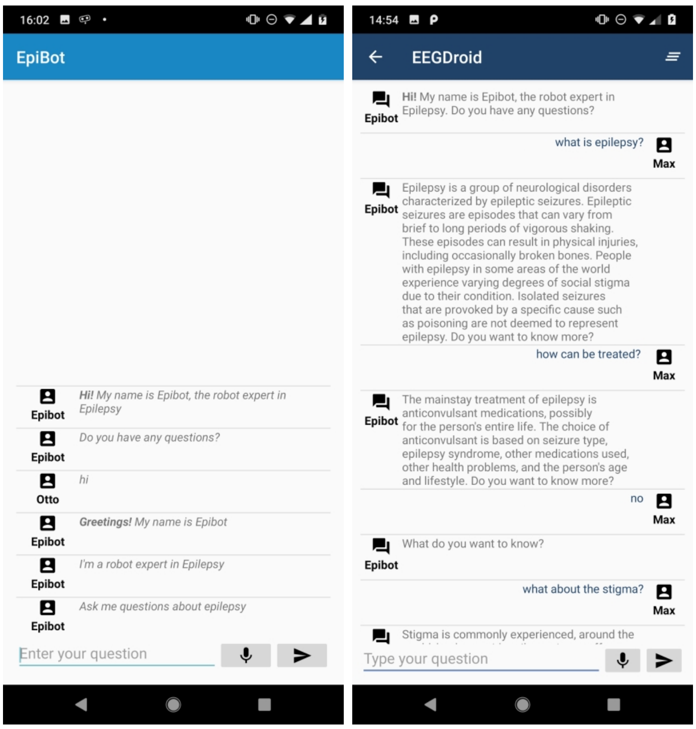
Another critical point to improve is to test the learning content offered in the app. It is essential to measure how understandable the content is and how effective it is to transmit the information. One possible idea is, instead of creating our own content, to collaborate with NGOs that already have as part of their mission to provide educative material about epilepsy (for example The Epilepsy Foundation[11]) and that have good quality educative material available.

Another vital task is to develop usability testing studies to measure how simple is the app to use and how well designed it is from the user point of view. However, we consider that it is essential to wait until the app is in a more advanced stage, so the usability tests provide information about a closer final user experience.

As additional future step, we consider important to develop different tools to receive EEG signals using a server. Different kind of applications can be implemented, for example, using a WebSocket server using JavaScript and Google Graphs to perform real-time on-screen plotting of the data.

One last possible improvement could be to implement an activity that allows the laboratory or health center to contact the user to acquire valuable information and vice-versa. Questionnaires to measure essential variables could be displayed as part of the app.

[11] https://www.epilepsy.com/



(a) v0.2 (b) v0.4

Figure 22: Comparison Epibot between versions

5 Bibliography

AndroidDocumentation (2019). Understand the activity lifecycle. https://developer.android.com/guide/components/activities/activity-lifecycle. Accessed: 2019-07-08.

Appel, K. (2018). *The Traumschreiber System: Enabling Crowd-based, Machine Learning-driven, Complex, Polysomnographic Sleep and Dream Experiments*. PhD thesis, Universität Osnabrück, Osnabrück.

Appel, K., Leugering, J., and Pipa, G. (2016). Traumschreiber: measuring and manipulating human sleep with a portable high-quality but low-cost polysomnographic system. *Journal of sleep research*, (25):158.

Birbeck, G. L. (2010a). Epilepsy care in developing countries: Part i of ii. *Epilepsy currents*, 4(10):75–79. doi:10.1111/j.1535-7511.2010.01362.x.

Birbeck, G. L. (2010b). Epilepsy care in developing countries: Part ii of ii. *Epilepsy currents*, 5(10):105–110. doi: 10.1111/j.1535-7511.2010.01372.x.

Browne, T. R. and Holmes, G. L. (2008). *Handbook of Epilepsy*. Lippincott Williams and Wilkins, Philadelphia, United States, 4 edition.

Cohen, L. (1995). *Time-frequency Analysis: Theory and Applications*. Prentice-Hall, New Jersey, USA.

GSM-Association (2018). *The Mobile Economy 2018*. Retreived from: https://www.gsma.com/mobileeconomy/wp-content/uploads/2018/02/The-Mobile-Economy-Global-2018.pdf.

Jäkel, M. (2017). Developing and designing a mediator smartphone application for portable sleep eeg recording and biofeedback with focus on the end user experience. Bachelor's thesis.

Royce, W. W. (1970). Managing the development of large software systems. *Proceedings of IEEE WESCON*, (26):1–9.

Sazgar, M. and Young, M. G. (2019). *Absolute Epilepsy and EEG Rotation Review*. Springer International Publishing, 1 edition. doi:10.1007/978-3-030-03511-2.

StatCounter (2017). Android overtakes windows for first time [press release]. http://gs.statcounter.com/press/android-overtakes-windows-for-first-time. Accessed: 2019-07-01.

StatCounter (2019). Website analytics made easy. https://de.statcounter.com/. Accessed: 2019-07-01.

Vidal, M. (2019). Auditory evoked potentials with mobile eeg. Master's thesis, Universität Osnabrück, Osnabrück.

A Appendix: Content of Learn activity

In appendix A, all the content from the Learn activity is presented. This content is the result of a bibliographical review from scientific sources, and it is presented to the users of the app as learning material. The goal of this content is to familiarize the users with basic concepts about epilepsy and redirect them to other sources of information.

Table of contents:

- Introduction
- What is a seizure?
- What is not a seizure?
- What type of seizures are?
- What causes seizures?
- What is a provoked seizure?
- What is epilepsy?
- How can epilepsy be diagnosed?
- What are the mechanisms of epilepsy?
- What are the risk factors for epilepsy?
- How can epilepsy be treated?
- How many people suffer epilepsy?
- Where can I learn more?
- Bibliography

A.1 Introduction

In the next paragraphs, we write some answers to the most usually ask questions about seizures and epilepsy. All the information collected here is obtained from scientific sources, and we try to provide reliable and precise information about these topics. However, this guide leaves many details out and should be considered only as an overview of the matter. If you have doubts about your health or think that you may suffer from seizures or epilepsy, you should visit a doctor. Only a physician can establish a reliable diagnostic. Early detection of this condition could help to improve the quality of life and minimize the risk of further complications.

A.2 What is a seizure?

An epileptic seizure is a transient occurrence of signs and symptoms due to abnormal excessive or synchronous neuronal activity in the brain [3]. The period during which the seizure occurs is called the ictus period. The aura is the earliest part of the seizure recognized (and often the only part remembered by the patient), and it could act as a warning. The period immediately after a seizure is the postictical period [10].

A.3 What is not a seizure? [1]

Not every episode of shaking or loss of awareness is necessarily a seizure. Several different conditions can produce episodes of shaking or loss of consciousness, for example, syncopes, migraines, transient ischemic attacks, and sleep disorders. To evaluate an episode, it should be taken into account:

- Precipitating factors: what were the circumstances before the event?
- Prodrome: what did the patient experience before the event?
- Time course: how long last the seizure?
- Stereotypy: do the events happen the same way every time?
- Behavior during factors
- Nature or recovery: the recovery was made with prolonged confusion, or was it rapid?

In order to arrive at the proper diagnosis, it is essential to also consider other non-epileptic etiologies.

A.4 What types of seizures are? [5]

Seizures are usually named by how they begin and develop:

- Generalized onset: originating from and rapidly engaging networks of both brain hemispheres of the brain
- Focal onset: originating within the networks of one hemisphere of the brain. Focal onsets can be further classified by the level of awareness (aware vs. impaired awareness) and motor vs. non-motor
- A seizure may be “unclassified” if there is not enough information or inability to place it in other categories.

A.5 What causes seizures?

It is vital to identify the origin of a seizure for treatment and prognostication. Some of the epilepsies may fit into more than one category [2]. Seizures can be caused by:

- Genetic causes: around 15% of people with epilepsy have a first-degree relative with epilepsy. The risk of suffering epilepsy is three times higher if a first-degree relative has epilepsy. Genetic epilepsies account for 15-20% of all epilepsies [6].
- Metabolic causes: epilepsy is common in disorders from inborn errors of metabolism (for example, vitamin B responsive disorders or disorders of amino acid). Many metabolic epilepsies are also considered epileptic encephalopathies, in which the seizures may contribute to severe cognitive and behavioral impairments [2].
- Structural: some structural problems are associated with seizures, for example, malformations of cortical development, hippocampal sclerosis, tuberous sclerosis complex, brain tumors, vascular events (hypoxic-ischemic injury, subarachnoid hemorrhage, stroke) and traumatic brain injury [2].
- Central nervous system infections: some infections are related to the development of epilepsy (meningitis, encephalitis, neurocysticercosis, and cerebral abscess) [2].
- Autoimmune causes: some autoimmune diseases as multiple sclerosis and Ramussen encephalitis can provoque seizures [2].

A.6 What is a provoked seizure?

Provoked seizures occur as the result of the seizure threshold being transiently lowered by some disturbance. They do not count toward a diagnosis of epilepsy. Some causes of provoked seizures could be medications, recreational drugs, alcohol withdrawal, barbiturate or benzodiazepine withdrawal, and metabolic [1].

A.7 What is epilepsy?

Epilepsy is a complex symptom caused by a variety of pathologic processes in the brain [10]. Epilepsy is characterized by an enduring predisposition to generate epileptic seizures, and by the neurobiological, cognitive, psychological, and social consequences of this condition. The definition of epilepsy requires the occurrence of at least one epileptic seizure [3]. This means that epilepsy is a disease of the brain defined by any of the following conditions [4]:

- At least two unprovoked (or reflex) seizures occurring in less than 24 hours

- One unprovoked (or reflex) seizure and a probability of further seizures similar to the general recurrence risk (at least 60%) after two unprovoked seizures, occurring over the next ten years
- Diagnosis of an epilepsy syndrome

A.8 How can epilepsy be diagnosed? [2]

Any of the following criteria can be used to diagnose epilepsy.

- At least two unprovoked (or reflex) seizures occurring in less than 24 hours
- One unprovoked (or reflex) seizure and a probability of further seizures similar to the general recurrence risk (at least 60%) after two unprovoked seizures, occurring over the next ten years
- Diagnosis of an epilepsy syndrome: although epilepsy is mostly a clinical diagnosis, EEG may help to establish a reliable diagnosis.

A.9 What are the underlying mechanisms of epilepsy? [10]

The seizures occur due to abnormal neuronal discharges. Although the causes of epilepsy are many, the major disorder is derived from abnormal synchronous discharges of network neurons. The events that lead from the seizure are not entirely understood, some hypothesis point to a decrease in the synaptic inhibition of the neurons, increase in synaptic excitation, alteration in potassium and calcium currents or changes in the extracellular ion concentrations.

A.10 What are the risk factors for epilepsy? [2]

Some of the factors that could increase the probability of developing epilepsy are:

- Complex febrile seizures of extended duration (more than 15 min) or more than one seizure in 24 hours
- Family history of epilepsy
- Head trauma with loss of consciousness
- Meningitis or encephalitis
- Perinatal distress

A.11 How can epilepsy be treated? [7]

There is a wide range of treatments to prevent seizures. Here we describe the most usual ones.

Antiepileptic drugs:

Seizures are treated with antiepileptic drugs (AED). For this, it is vital to identify the type of epilepsy (generalized vs. focal). The narrow spectrum AEDs are indicated for focal epilepsies but may exacerbate generalized genetic epilepsies. Some of the medications used in generalized epilepsy are benzodiazepines, felbamate, lamotrigine, levetiracetam, perampanel, rufinamide, topiramate, valproic acid, and zonisamide. On the other hand, the drugs used for focal epilepsies are carbamazepine, eslicarbazepine, ezogabine, gabapentin, lacosamide, oxcarbazepine, phenobarbital, phenytoin, pregabalin, tiagabine. Some antiepileptic drugs may cause life-threatening reactions, especially worsening depression and suicidal ideation.

Vagus nerve stimulation:

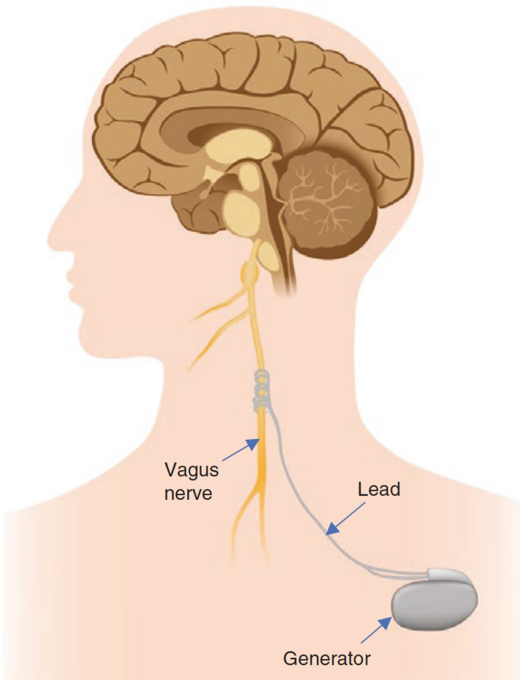


Figure 23: Vagus nerve stimulation. Taken from [7]

Additionally, other types of treatment can be used to fight seizures, for example, the vagus nerve stimulation. This stimulation consists of placing subcutaneously a small pulse generator in the left chest with a lead attached to the left vagus nerve. This nerve is stimulated at regular intervals.

Responsive neurostimulation:

Consist of a closed-loop system that continuously records the electrocorticographic activity and provides responsive electrical stimulation when specific patterns are detected. The neurostimulator should be implanted cranially and connected to one or two depth or subdural cortical strip leads.

Deep brain stimulation

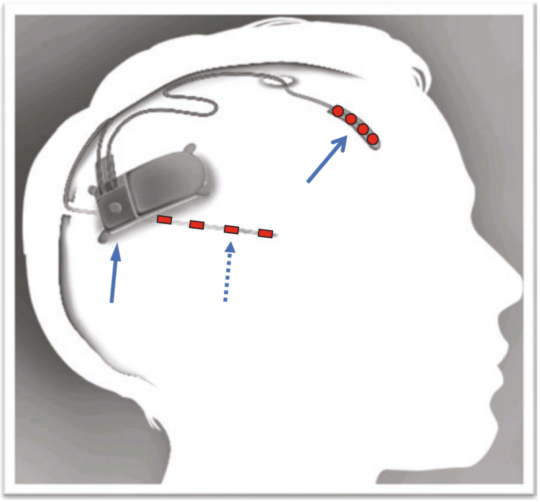


Figure 24: Deep brain stimulation. Taken from [7]

Bilateral electrodes are implanted in the anterior nucleus of the thalamus. It has proved to be useful, but its mechanism is unknown. It can help with reducing the seizure severity, improvement in the quality of life, and neurological measures.

Diet Therapy

Particular types of diet can help to reduce the frequency of the seizures. The Ketogenic Diet, which consists of a ratio of 3:1 between fat and carbohydrates and protein, has proved to reduce the seizures, especially with children and adolescents. The Modified Atkins Diet presented effectivity similar to Ketogenic. However, both diets have adverse effects and can be quickly discontinued by the patients [7].

Epilepsy Surgery

Approximately 30% of the patients with epilepsy will keep having seizures despite the used of medications [8]. This is called drug-resistant epilepsy (DRE). A patient is considered to have DRE when two antiepileptic drugs have failed in improving his / her condition. Surgery is an option for patients with this kind of epilepsy. A pre-surgical evaluation is needed to select between the different kinds of surgical interventions. This evaluation consists of a series of exams and monitoring with to goal of determining the specific region of the intervention. According to the region of the brain intervened, the surgery can produce significant secondary effects that affect the life quality of the patient [7].

A.12 How many people suffer epilepsy?

Epilepsy has a prevalence of 0.5 - 2% amongst the general population. In two-thirds of cases, the onset of the disease occurs during childhood or adolescence [8]. 80% of persons with epilepsy are in the developing world [9].

A.13 Where can I learn more?

There is a lot of information on the internet about epilepsy. However, it is essential to be careful about how we read this information. Here we present some recognized sources of information about seizures and epilepsy, with which you can learn more about his topic.

- World Health Organization about epilepsy
- Epilepsy.org (United States)
- National Institute of Neurological Disorders and Stroke (United States)
- International League against epilepsy (United States)
- National Health Service (United Kingdom)

A.14 Bibliography

1 Sazgar M., & Young M.G. (2019) What Is Not a Seizure?. In: Absolute Epilepsy and EEG Rotation Review. Springer, Cham

2 Sazgar M., & Young M.G. (2019) Seizures and Epilepsy. In: Absolute Epilepsy and EEG Rotation Review. Springer, Cham

3 Fisher RS, van Emde Boas W, Blume W, et al. Epileptic seizures and epilepsy: definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE). Epilepsia. 2005;46(4):470–2.

4 Fisher RS, Acevedo C, Arzimanoglou A, et al. ILAE official report: a practical clinical definition of epilepsy. Epilepsia. 2014;55(4):475–82.

5 Fisher RS, Cross JH, D'Souza C, et al. Instruction manual for the ILAE 2017 operational classification of seizure types. Epilepsia. 2017;58(4):531–42

6 Jallon P, & Latour P. Epidemiology of idiopathic generalized epilepsies. Epilepsia. 2005;46(Suppl 9):10–4.

7 Sazgar M., & Young M.G. (2019) Treatment of Seizures. In: Absolute Epilepsy and EEG Rotation Review. Springer, Cham

8 Aronica, E., & Mühlebner, A. (2017). Neuropathology of epilepsy. In Handbook of Clinical Neurology.

9 Birbeck, G. (2010). Epilepsy care in developing countries: part I of II. Current Review in Clinical Science, 10(4).

10 Browne, Th. & Holmes, G. (2008). Handbook of Epilepsy (4th edition). Wolters Kluwer

B Appendix: Content of Tutorial activity

In appendix B, all the content from the Tutorial activity is presented. This content is the result of bibliographical review from scientific sources, and it is presented to the users of the app as learning material. The goal of this content is to familiarize the users with basic concepts about EEG, and display information about how to perform the experiments.

Table of contents:

- What is the Traumschreiber?
- What is EEG?
- How EEG works?
- Why is EEG important?
- How can I use the Traumschreiber?
- Bibliography

B.1 What is the Traumschreiber?

The Traumschreiber is a mobile EEG device designed by Kristoffer Appel, Johannes Leugering and Gordon Pipa from the University of Osnabrück. Originally it was designed as a sleep mask for research purposes [1,2]. It includes eight channels that are usable to record EEG, EOG, and EMG. The usual EEG involves going to a laboratory or a hospital and requires an elevated cos. On the other hand, the Traumschreiber is characterized for being small, light and easy to use, as well as having a very cheap production cost. The use of the Traumschreiber is still experimental. However, we have found good evidence about its correct functioning in research data collections and sleep research.

B.2 What is EEG?

Electroencephalography, shortened as EEG, it is an electrophysiological monitoring method for the brain. In other words, it is a test used to record the electrical activity of the neurons in the brain. It is noninvasive and requires the use of electrodes placed along the scalp. Those electrodes measure the postsynaptic potentials of the neurons. [3]

B.3 How EEG works?

A typical EEG display graphs voltages on the vertical axis and time on the horizontal axis, providing a near real-time display of ongoing cerebral activity.

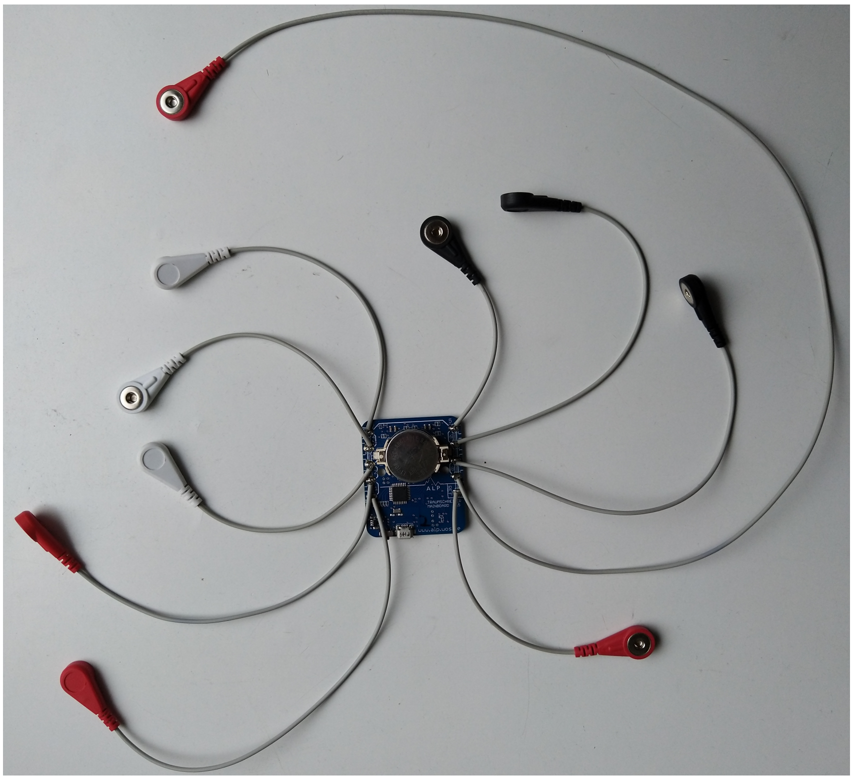


Figure 25: Traumschreiber

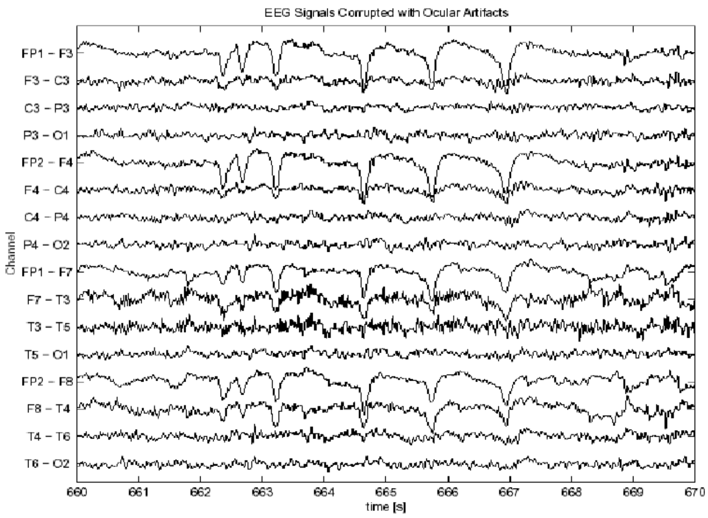


Figure 26: Example of EEG session

To record the signals, EEG uses the principle of differential amplification. The recording is made from voltage differences between different points, using a pair of electrodes that compares one active site with another site of reference. Through the measure of differences in electrical potentials are discernible EEG waveforms generated [6].

B.4 Why is EEG important?

EEG is the most useful diagnostic procedure for Epilepsy. Epilepsy causes abnormalities in the EEG readings, and during a stroke, unusual behavior in the brain signals is recorded [4]. EEG confirms the presence of abnormal electrical activity and gives information regarding the type of seizure disorder and the location of the seizure focus [5]. Epilepsy allows answering the following questions: Does the patient have epilepsy? Where is the epileptogenic zone of the brain? How good is therapy?

B.5 How can I use the Traumschreiber?

In the next video, we show you how you can operate the Traumschreiber: how you should put the electrodes in your scalp and how you can take them off after the experiment.

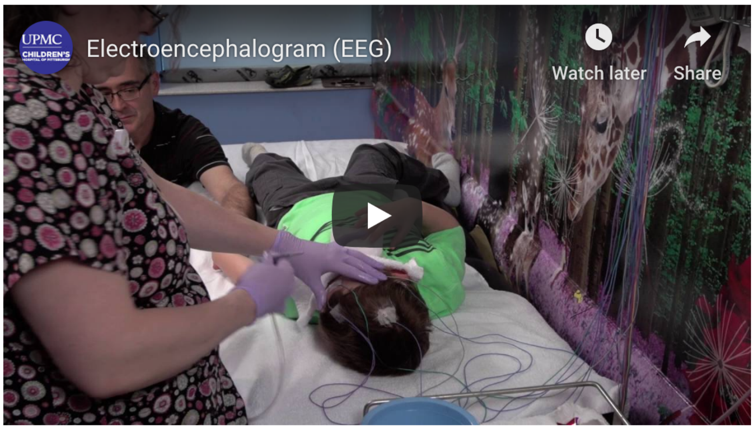


Figure 27: YouTube tutorial video

B.6 Bibliography

1 Appel, K. (2018). The Traumschreiber System: Enabling Crowd-based, Machine Learning-driven, Complex, Polysomnographic Sleep, and Dream Experiments.

2 Appel, K., & Leugering, J. (2017, January 17). Neuroinformatics - Institute of Cognitive Science.

3 Sazgar M., & Young M.G. (2019) Overview of EEG, Electrode Placement, and Montages. In: Absolute Epilepsy and EEG Rotation Review. Springer, Cham

4 Noachtar, S. & Rémi, J. (2009). The role of EEG in epilepsy: A critical review. Epilepsy Behavior, 12, 22-23

5 Browne, Th. & Holmes, G. (2008). Handbook of Epilepsy (4th edition). Wolters Kluwer

6 Britton JW, Frey LC, Hopp JL et al. (2016) Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants. Chicago: American Epilepsy Society