Malmö University

Faculty of Technology and Society

Examensarbete för kandidatexamen I datavetenskap - 15hp

Integration of Bluetooth Sensors in a Windows-Based Research Platform

Integration av Bluetooth-Sensorer i en Windows-baserad Forskningsplattform

Rohan Samandari

Program:	System development
Supervisors:	Dario Salvi
	Carl Magnus Olsson
	Carmen Carrasco López
Examiner:	Reza Malekian

Spring 2021

Abstract

This thesis describes how to build a solution for transmitting data from an Electroencephalography (EEG) device to a server in real-time while guiding the user through a number of predefined exercises. This solution will be used by Spinal Cord Injury (SCI) patients suffering from neuropathic pain, in order to understand if it is possible to predict such pain from EEG. The collected data will help clinicians analyze the brain activity data from patients who can submit the data from their home. To accomplish this development task, an application was built that connects to a portable EEG device, gather brain activity data from patients, guides patients through a set of imaginary tasks and sends the data to a server. This project made use of a Software Development Kit (SDK) for the Python programming language and a web sockets server written in JavaScript. The application was tested both in terms of usability and end-to-end latency, showing high usability and low latency. The proposed solution will support a clinical trial in Spain with 40 SCI patients.

Keywords: Bluetooth sensors, EEG device, Electroencephalography device, EEG connection, Bitbrain, Python

Acknowledgement

There are many who helped me along the way on this journey. I want to take a moment to thank them for being a support, both as helping me and as being patient by my side.

First of all, I wish to thank my supervisors. Carmen Carrasco López, and Carl Magnus Olsson who went above and beyond to help me reach my goal. A special thanks to Dario Salvi not only as a supervisor but even as a leader, whose help gave this study a form.

To Reza Malekian, as examiner to this thesis whose comments and feedbacks made a better version of this thesis.
To Bitbrain support with the helps and explanation I received from them, which means a lot in this work.
To my friends and family members who participated one of the tests. I am grateful for their help and appreciate it. It would be difficult to gain a result without their help.

Finally, to my wife, Masoumeh, and my daughter, Aida: your love and understanding helped me through the tough times. Without you believing in me, I never would have made it so long. I am forever grateful for your patience and understanding. It is time to celebrate; you earned this degree right along with me.

Last but not least, I want to thank me for believing in me and to being such a hard-working.

Table of Contents

Abbreviations

CSV	Comma-separated values
EC	European Commission
EEG	Electroencephalography
FR	Functional Requirement
GUI	Graphical User Interface
IDE	Integrated Development Environment
NFR	Non-Functional Requirement
SCI	Spinal Cord Injury
SD	Secure Digital
SDK	Software Development Kit
UI	User Interface
UML	Unified Modeling Language
VAS	Visual Analogue Scale
VSC	Visual Studio Code
M	Must
S	Should
C	Could
W	Wont

List of figures

List of tables

1. Introduction

Spinal Cord Injury (SCI) is a destructive neurological and pathological state of the central nervous system which is producing morbidity, psychological stress, and physical dependency [1]. Patients who have this disease are not able or willing to visit a hospital for submitting their tests. One cause visiting a clinic is emergency pains. The pain can be reflected in the changes of the brain waves which can be collected using an EEG device [2].
Electroencephalography (EEG) is a recording of electrical activity in the brain using small, metal disks called electrodes which are attached to the scalp. Brain cells communicates to each other through electrical impulses and these impulses are active all the time. The activity of the brain is shown as waves on an EEG recording [2]. To see or read these waves there is a need to an EEG device that is connected to a monitor.

Nowadays it is not possible to ignore the role of computer technology in health care. The communication between healthcare staff, the analysis of any results and managing and transmitting data, are happening with the help of computer technology [3].
Some technologies are used in places such as hospitals and clinics. Some patients who are not able to travel to the clinics to submit their samples are neither able to submit them from home, due to lack of possibilities such as technology and devices. There are fortunately some devices that can help the patients to measure their brain activity. These devices are such as a computer or a tablet and an EEG cap. The EEG cap is a device that can be placed on a patient's head and read the brain activity from the patient's scalp. Some of these devices has the ability to store EEG data into a memory card and is also able to communicate to a computer with a connected cable or in other cases with its bluetooth sensor [4].
There are SCI patients who are suffering from neuropathic pain. These suffering are followed by mild, moderate, or severe pain. Treatment to these pains is usually handled by medication. A European Commission (EC) research project is trying to predict these pains and treat them before the pain hits. One part missing in this project is a possibility of connection between the EEG device and a computer through bluetooth, together with an instruction which guides the patient to perform a test. This possibility is a sort of an application which, as described, guides a patient through a test while it gathers data from brain activity via a bluetooth EEG device.

1.1 Main aim and research questions

The main goal of this thesis is to help SCI patients, submit their EEG records in an easy way and from their home. This is done by developing an application which can receive data from an EEG device, show the data and impedance level on an interface to the user and send them to a clinician in real time, while it stores the data locally on the application device for further data analysis after the session.

The work within this paper represents a technical development perspective of the 1st; integration of the EEG device with a tablet application, 2nd; showing EEG data received from the device, live and in real-time for the user, 3rd; storing and sending the data collected to a server for further analysis in the EC project, and 4th; guiding the patient to perform an imaginary test while it collects data (see table 1). This test is divided into 5 tasks which are about: keeping the eyes open, keeping them closed and imagining a movement of respective arm while the application shows some randomly right and left arrows. The tasks are further explained in section 5.2.

Research question 1:

How can Electroencephalography data be transmitted to a clinician, remotely and in real-time, through a computer technology?

Research question 2:

How usable can a such application be?

1.2 Limitations

The first identified limitation is the knowledge about the programming language. The required programming language is Python which takes a while to learn, and this leads to the lack of time. For usability test it would be more relative to have some of EC project's patients to perform the test. But in that case this thesis must have an ethical approval to have the patients as participants. Information about the ethical approval is needed, came a bit later when the test was planned. To applicate for the ethical approval and find participants, would make a delay for the result. To avoid application for the approval, the test is performed without data gathering from the participants. And participants chosen are among relatives and persons who are involved in this thesis.

1.3 Contributions

A working application can help the EC project to succeed the pain prediction. The success of EC project will help the SCI patients to deal with the pain better by handling the medication more accurate, and the treatment received before the pain may happen. This study can be a part of a medical treatment for SCI patients who are suffering from severe pains.

2. Research methods and requirements

This section briefly explains the chosen methods. The methods have been compared to each other in Oates book [5], and some parts of the method have been analyzed and planed. It is not only the methods analyzed but also the plans, limitations in the project and the communications with the project owner has been discussed. These parts have been discussed iteratively as well as the project process.

2.1 Chosen methods

The main method to accomplish this thesis and answer the research questions is to create some experiences while developing an application. The first method, for gathering information about other related documented researches, is Literature Review. This method is chosen because it is suitable for gathering information about other researches. Another method chosen is Design and Creation, which can help the thesis to find an answer to both research questions. The further knowledge after the development of the application will be to analyze the usability of the application. In this case the Observation method is chosen.

2.1.1 Literature Review

To research and find information about the SCI patients, bluetooth EEG devices, bluetooth applications, and related works it needs to use Literature Review method. Literature Review is a method for using to gather information about other documented researches [5].

2.1.2 Design and Creation

Design and Creation, one of the methods which is described in Oates book [5], is one of the chosen methods for this project. With this method it is possible to find an answer for the first research question. Oates describes it as a well-documented method which suits to develop an application [5].

2.1.3 Observation

Observation is another data generation method which is also described by Oates [5] and is used to answer the second research question. As Oates describes it, it is suitable when a participant is performing a task and an observer is observing him or her. By observation, Oates means watching the participant, taking notes and predicting the next step of the participant [5].

2.2 Literature Review

To read about other related works and to have a better understanding about what is done and what is missing, can a method called Literature Review be used [5]. It is important to know about previous research and related works to determine what has been discovered. Even how a such application can be developed needs some reviews. The literature that are going to be reviewed should be related to the work and also be based on comprehensive understanding of the current knowledge of the topic [6]. This method helps the thesis to have a better understanding about each issue, such as existing related research, EEG, SCI, and bluetooth applications.
According to Oates [5], there are some steps which are used to have a better result in this research method. They are as following:

1. **Searching**: How a researcher can do a better research of information
2. **Obtaining:** On which way the researcher can find the literatures. Libraries and search engine are two examples in this part.
3. **Assessing**: How to access the creditably of a source. These sources are books, journal articles, conference articles and webpages.
4. **Reading**: How to find a relevant literature is by reading some part of the founded literature. The main part is the abstract level of a literature.
5. **Critically evaluating**: How to read a previous work depends on the information in the paper and which type of the literature the work has been published in.
6. **Writing critical review**: The researcher should write a review about the researches, the related works and how they have been in help. Another important thing is that if this paper did not repeat the previous works. This means that this thesis has accomplished a further knowledge about the topic.

2.3 Design and Creation

This method is broken down in five parts which uses an iterative process, as Oates describes it [5]. These five parts are as following: “awareness”, “suggestion”, “development”, “evaluation” and “conclusion” [5]. Another level of the Design and Creation method is the System Development methodology which is a part of development phase. These steps will be further explained in detail in the following section.

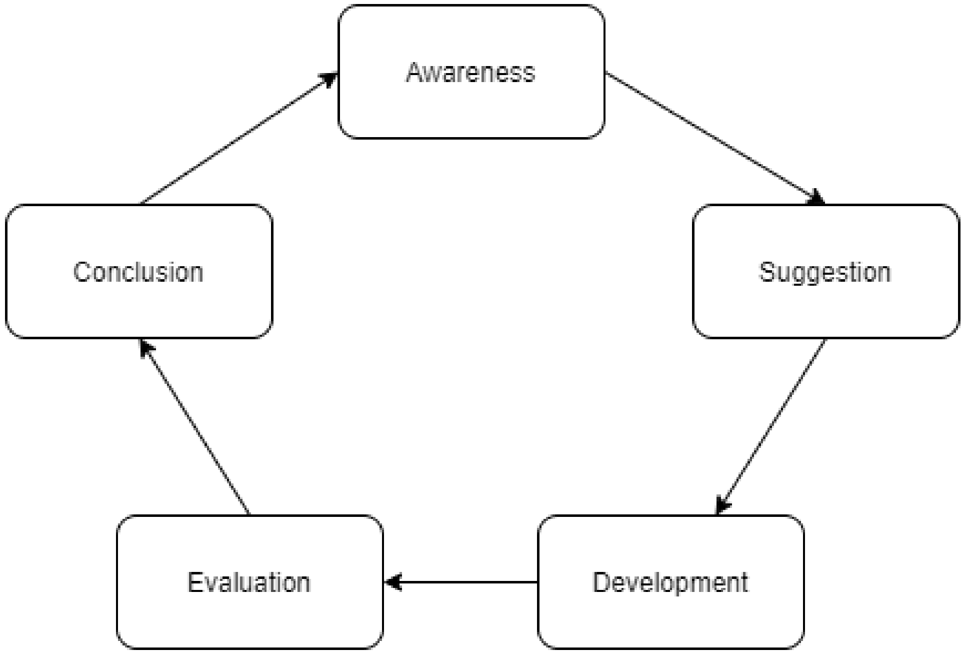

Figure 1. The iterative Design and Creation process

2.3.1 Awareness

In this part some information about a problem or a need is gathered [5], which is already defined by the project owner. In the other hand the further information and detailed explanation about the design and functions is discussed between the project owner, client, and the developer.

2.3.2 Suggestion

Ideas about how the problem will be solved, is discussed in this part, and details are documented and used in the developing part [5]. The preliminary designs and functions are also discussed. UML diagrams is used to identify which part and functions are needed and where they are placed in the application.

2.3.3 Development

The implementation of ideas is started and is done in this part. All the discussed and documented designs and functions is implemented one by one. Some changes came during the development when the client and the project owner have been in touch in the project. This means that the project is iterating between awareness, suggestion, and the development part. The project needs a system development method which should be followed [5]. The development method is the agile method which allows the project to run iteratively between each part [7].

2.3.4 Evaluation

After the developing part it is time to compare the product with the expected system that has the functions and the design explained in section 2.8. Those requirements are decided to be implemented in the application, and in this part, they are examined how satisfied they are implemented.

2.3.5 Conclusion

In the last part, the project will be documented, and the knowledge gained will be identified. It also needs to identify any loose ends and unexpected results that cannot yet be explained by the analyzes which could be a subject for further research.

2.3.6 System development methodology

Oates [5], describes in system development methodology section that a research report should contain an explanation and documentation of how the developer has worked through the project. The project is broken down in four stages; analysis, design, implementation, and testing [5]. This method is chosen and is related to the project because of its flexibility, which means that every stage can be visited and changed while the other stages are in process.

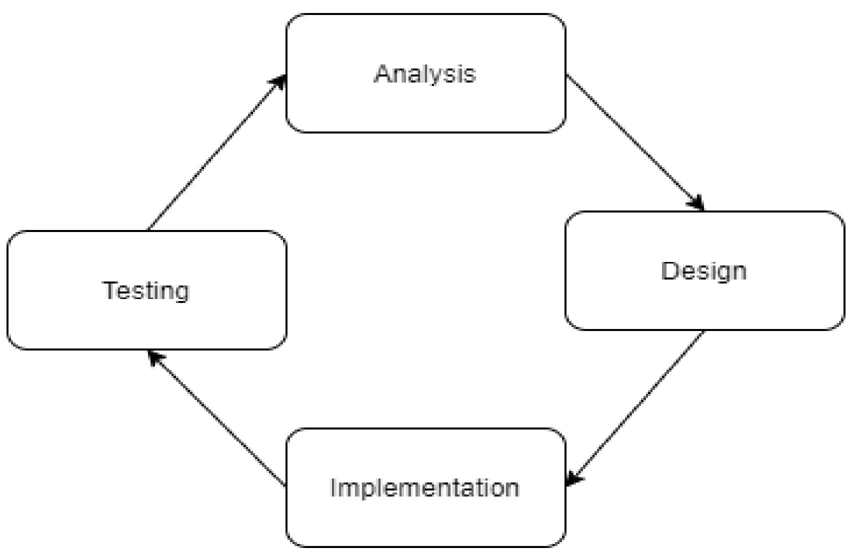

Figure 2. The iterative System Development process

2.4 Observation

There are several kinds of Observation that are related to this project and are explained in this section. One of those which is relevant to this project is Systematic Observation. In this method the type of the events which will be observed, is pre-defined and a

schedule for noting the observation is pre-designed [5]. The kind of observation used in this paper is called Think-aloud observation. This method is a pre-defined observing test which has three moments: 1, Define the users or participants. 2, explain them the tasks, and 3, do not talk and let the user use the system and do the talking [8].
While user goes through the tasks the talking is recorded and some notes of what he or she is doing is taken.

2.5 Method discussion

To choose the best method for this accomplishment, requires both knowledge and experience. The first moment in this thesis is about to have a better understanding about the project, related works and other information that increase the knowledge about the topic. To achieve this knowledge the Literature Review fits the research. This method makes it easier for the two following methods to be accomplished in a satisfied way.

First following method is to experience developing av an application and analyze the result to answer the first research question. The one method which could fit in this project is Experiment. Experiment method is also about to experiment and test several solutions and then compare the results [5]. This method is not chosen because the only way to answer the research questions is to create one application and there is not enough time to experiment several solutions. In the other hand the Experiment method requires a thesis which also should be proved, which is not the goal of this thesis. A reason to choose Design and Creation method is because it is about creating a system. The other method, which is a part of the Design and Creation method, is called System Development Methodology. This method helps the project to be accomplished in a way that can help the question to be answered. There are other methods in this category, but the System Development Methodology is suitable because of its flexibility.

For the second research question there are multiple methods, described by Oates [5], which can help the question to get an answer. One of them is to let some participants use the application and be observed by an observer. Another method is to let a participant use the application and give feedbacks or answer similar pre-designed questions. An interview with a clinician as a participant would be a great help to approve what is well done and what is missing. This method is called Interview and could be a good choice. A combination of those mentioned methods can result to a satisfied result. But as the developing part takes the most time of this thesis, there will be some simple methods to getting an answer to the second question. The observation of some participants takes less time and answer to more wonderings. A Think-aloud method fits in this paper for getting a better user test result. The benefits of this method are that it takes less time, and the observer is not involved in the test which means that the observer is not affecting the user. The other benefit is the user can test the system freely and trying to solve any conflicts by him or herself [8].

2.6 Hardware

The needs for implementing this project are an EEG device which will be placed on a user's head that can gather and transmit the data to a Windows-based platform. This device is available in two parts, an EEG amplifier, and a Dry-EEG headset [9]. Both parts are used in this thesis which consists of 8 main parts, adjustable headband, ear clip, amplifier, micro memory card, connection cable, power switch, light, and battery. Figure 1 presents the device design.

There are 12 channels placed on the headset with sensors connected to user's scalp. An ear clip is placed on the left ear lobe which connects the ground [9]. The amplifier connects to the headset via a cable mounted on the headset. Amplifier is equipped with the bluetooth feature for connection to other devices and has a place for a micro memory card. A button is placed on the amplifier which turns the device on and off. There is also a light on the amplifier which is blinking when the device is on but not connected. The constant light shows that the device is connected to another device via bluetooth. The device cannot connect and run by itself, and an application must have the control over the device.

Figure 3. Bitbrain Electroencephalography device

A computer which supports bluetooth connection, is needed for the development and the testing, and a Windows-based tablet which also supports bluetooth connection, is needed for further use by a patient or user.

2.7 Software

There are many free Integrated Development Environments (IDE) for Python programming language, which can be used for developing of the software, but the chosen IDE is Microsoft Visual Studio Code. The main reason of the choice is the IDE's simplicity and functionality.
There is another required tool which should be used to connect to the bluetooth EEG device. That tool is called Bitbrain Software Development Kit (SDK). There are other libraries and SDKs which also should be used to perform the project. To test the EEG cap's connection and receiving and showing the signals, it requires to download and install an application that is called Bitbrain Viewer [10]. This application allows the developer to check the connection and the signals. This application can only be connected to Bitbrain company's bluetooth EEG devices and with the help of its SDK. One of these devices is used in this study.

Another tool which is required for delivering and sharing the document, implemented codes and other materials is GitHub software which is even available on a website [11].

2.7.1 Microsoft Visual Studio Code

Microsoft has a free IDE which supports several programing languages and is popular IDE among developers [12]. This software will let the developer to develop any kind of applications with many choices of programing languages. The programing language which will be used for developing this application is Python. This is because there is an available Bitbrain Python SDK that makes the connection to the device possible.

2.7.2 Bitbrain SDK

Bitbrain SDK allows the data acquisition from the EEG cap to a Bitbrain programming platform [13]. This SDK can be used to a third-party application that uses the Bitbrain hardware.

2.8 Requirements

Due to some discussion with the project owner some functions are identified and decided to be implemented in the application. These requirements are the part of the study which are necessary for a working system. Not only a working system is the focus on this paper but even answering the questions require to implement these functions. The requirements are divided into two categories, functional and non-functional requirements and are shown in the following table. The first column contains a specific ID for each requirement. And in the "Description" column the requirement is shortly described. In the "Priority" column, the priority of the requirement is marked by a letter. These letters are used to describe the prioritization of a requirement. This prioritization technique is described further in Cobb's book [14], which is shortly described in this section. This technique is called MoSCoW method and is defined as following:

- **M**ust have – this priority means that the requirements labeled with "Must" have to be implemented and if it is not implemented the project delivery should be considered as failure.
- **S**hould have – this label on a requirement means that the requirement has also to be implemented but including in the delivery box is not necessary.
- **C**ould have – means that it is nice to have this requirement in the delivery box.
- **W**on't have – means that the requirement will not be implemented and delivered at this time.

The table is divided into two parts. Functional requirements and Non-Functional requirements. The identification of an ID for a requirement, is based on an abbreviation of the type of the requirement. The abbreviation is followed by a number which describes the implementation order of the requirement. As an example, see FR1 which stands for Functional Requirement number 1. and NFR1 stands for Non-Functional Requirements number 1.

Req.ID	**Description**	**Priority**
FR1	Connect to Mobistudy server through WebSocket	M
FR2	User should be able to login onto the Mobistudy	M
FR3	Ask the user for the pain level by a Visual Scale	M
FR4	Connect to the Bitbrain portable EEG sensor through Bluetooth	M
FR5	The user should be able to check the signals from the EEG device	M
FR6	Collect data from the sensor and send them to the server	M
FR7	Produce CSV files for the collected data	M
FR8	Guide the user through imaginary tasks while collecting data	M
FR9	Store and send, up to 256 signals per second	M
FR10	Show all 256 signals per second on the UI	C
FR11	Show a few signals per second on the UI	S
FR12	The ability to change the GUI language into at least two languages	C
NFR1	Be usable (Should be easy to understand and to use the application)	M
NFR2	Be reliable (Should not crash during running)	M
NFR3	The code should be readable by other developers	S
NFR4	Have a latency lower than 5 seconds from EEG device to the server	S
NFR5	Relatively visible and touchable buttons for patients with disabilities	M

Table 1. Functional and Non-Functional Requirements

2.9 Evaluation and Testing

As the project requires, there should be a test moment of the application. And as an answer to the questions the result should be analyzed and documented. The planed event for that is to design and describe some test scenarios that are going to be performed at the last part. The test result should meet all the requirements and the application should be further developed if any requirements have not been fulfilled. The chosen test method for this project is a variant of black-box testing [15]. In this kind of testing and in this project, the application will receive an amount of data as input and the results which are shown on the GUI, stored on the files locally, and the data which has been sent to the server will be compared and analyzed.

3. Results

This section further explains the results of this work. Those parts which are explained are as following:

- Literature review
- Design
- Functionality
- EEG cap integration
- Latency tests
- Usability tests

3.1 Literature Review

I did some literature research about the CSI, EEG, bluetooth application, bluetooth devices, computer technology in health care, and related similar works, on Malmö University Libsearch library. As it was needed for other tools and methods, some other research on Google Scholar is done. Research on Google Scholar library helped to find information and data for how to use Python language and other necessary development tools. To choose these tools and how to use them it took some online research on related webpages as well.

A few of the used literature sources are books about planning, requirement gathering, implementing, testing, structure of the thesis and to identify research methods. Several literature and documentation have been taken into consideration while other literatures that were less related to this paper, has been disregarded.

To understand and compare these literature sources, it was needed to read the abstracts of several chosen articles by relevant titles. After reading the abstracts the amount of the literature was reduced depending on how relevant they were for the work.

The research data of literature research is included in the following table.

Searched Key words	Libsearch	Google Scholar	Date
Spinal Cord Injury	196571		29/3
Spinal Cord Injury patients	55378		29/3
Computer technology role in clinic	869		29/3
Computer technology role in health care	5675		29/3
Electroencephalography application	17869		15/4
Electroencephalography	238743		15/4
Bluetooth Python	90		15/4
Bluetooth EEG headset	35		4/5
Arduino		1900	29/3
Creating and converting to CSV files		32400	24/3
Testing methods		76000	29/3

Table 2. Results for combination of key words and date of access

3.1.1 Related works

Due to some research about similar works, there are some works related to the same topic. Of these related studies one is chosen, which is about testing an EEG device and what data it can receive. The main aim for this research is to find some similar works to create a better understanding about how these devices work.

Using an EEG device via bluetooth is related to this work that helps this paper to get a better result. A work related to this thesis is an EEG device user test about checking the connectivity, receiving data and filtering the received data [16]. Aim of this related paper is to contribute to the area of remote control and interpretation of EEG data which are gathered from a bluetooth EEG device. To filter noise from the EEG headset to the connected device is one of the focus on this paper. The EEG device in this work is not the same. The similar features are the bluetooth connection and data transmitting. The headset described in the paper finds a specific Arduino bluetooth device and it can then connect to it. For this connection it requires some prior configuration. The modules role must be set to master which means that the control over connecting is set to this headset and not to other devices [16]. Arduino Board [17] is another bluetooth device which is required for both connection and the data transmitting from the EEG headset. The Arduino device should be successfully connected to be able to receive data packet [17]. Each data packet consists of 3 parts: 1, packet header which provide data stream synchronization; 2, packet payload which ensures that new data fields can be added to the packet without breaking any packet parsers; and 3, packet checksum which checks the data integrity. The data payload consists of several data rows and each data row is a series of bytes which presents the type and the value of the data transmitted [16]. This data stores even locally on a memory card on the Arduino.

This related work helps the thesis to have a better understanding about the Bitbrain EEG device and how it performs the connection and the transmitting of data.

3.2 Development

As it was discussed in section 2.3.6, the development and testing has been accomplished iteratively which means that some functions have been tested during the development. In the following table the result of the Functional and Non-Functional requirements is shown. In Implemented "Impl" column, the status of the implemented function is marked by "✔" symbol which means that the function has been implemented and tested and it works as it is described. The "**X**" symbol represents that the function is either not implemented or it does not work as it is described.

Req.ID	Description	Impl	Prio
FR1	Connect to Mobistudy server through WebSocket	✔	M
FR2	User should be able to login onto the Mobistudy	✔	M
FR3	Ask the user for the pain level by a Visual Scale	✔	M
FR4	Connect to the Bitbrain portable EEG sensor through Bluetooth	✔	M
FR5	The user should be able to check the signals from the EEG device	✔	M
FR6	Collect data from the sensor and send them to the server	✔	M
FR7	Produce CSV files for the collected data	✔	M
FR8	Guide the user through imaginary tasks while collecting data	✔	M
FR9	Store and send, up to 256 signals per second	✔	M
FR10	Show all 256 signals per second on the UI	**X**	C
FR11	Show a few signals per second on the UI	✔	S
FR12	The ability to change the GUI language between two languages	✔	C
NFR1	Be usable	✔	M
NFR2	Be reliable (Should not crash during running)	✔	M
NFR3	The code should be readable by other developers	✔	S
NFR4	Have a latency lower than 5 seconds from EEG device to the server	✔	S
NFR5	Relatively visible and touchable buttons for patients with disabilities	✔	M

Table 3. Implemented and not implemented functions.

3.3 Functionality

The implemented and tested functions are further explained in this section.
These functions have been developed in parallel and sometimes iteratively. Some functions have not been fully implemented before other functions were developed. The iterative process helped the application to be demonstrated as soon as they were done which led to get feedbacks for some changes that could improve the functions. The implementation of each functional requirements is briefly explained in following subsections.

3.3.1 FR1 *(server connection),* FR2 *(user login)*

In the first User Interface (UI), the user can use his or her account information to login to the Mobistudy webpage [18]. The account information used to login is also stored in a text file locally for future in logins. The user is even able to log out from his or her account and login to another account (see figure 4 and 5). This function connects and logs the user into the server and the connection runs while the application is running. This connection transmits the test data to the server and stores it into a specific user database.

Figure 4. Main GUI

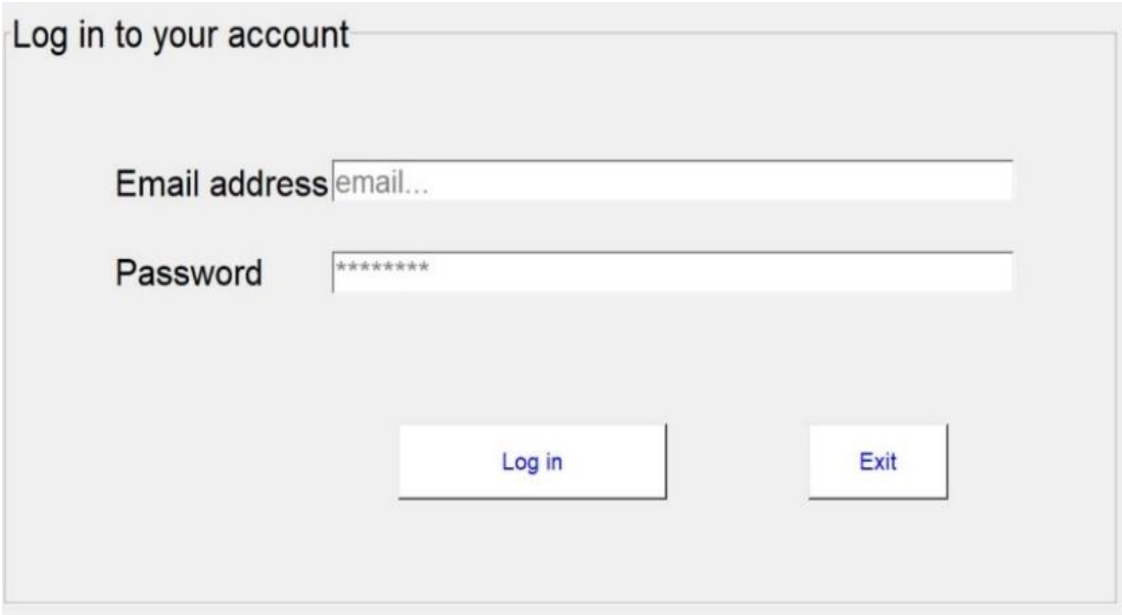

Figure 5. Login GUI

3.3.2 FR3 *(pain level insertion)*

A Graphical User Interface (GUI) is displayed, in which the user can choose a pain level between 0 and 10 (see figure 6). The chosen digit is stored locally and is sent later to the server.

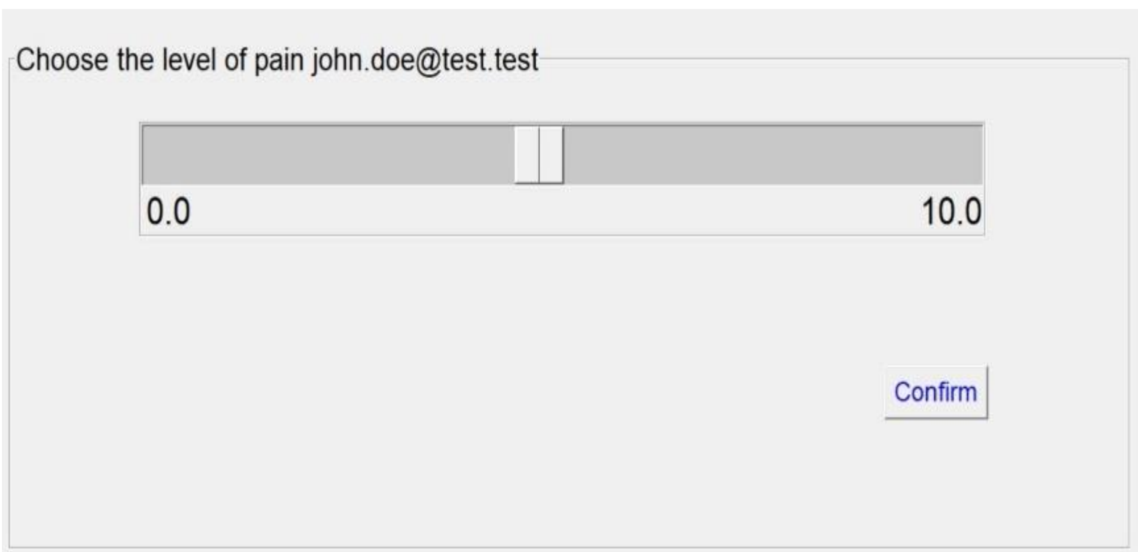

Figure 6. VAS GUI

3.3.3 FR4 *(EEG connection)*

Connection to the EEG device is performed automatically without any user's involvement, via bluetooth, if the device is switched on. Otherwise, after 5 seconds of trying, the application shows a message that the device is switched off and it should be turned on (see figure 7). The user should turn on the device and can then press "Try again" button so the application begins searching for an in advanced indicated device. The connection is made by using the Bitbrain SDK [13]. The SDK controls the searching, connection and data receiving. The application only needs to call the connection method in the Bitbrain's Python API which is based on a C++ SDK. The connection runs until the "Disconnect" method in SDK is called, or the running of the connection is set to 'False'.
This connection runs by a separate thread which means that the GUIs can run in parallel.

Figure 7. EEG Connection GUI

3.3.4 FR5 *(EEG signal check)*

An interface is showing a graph of data on 9 different rows for 9 connected channels. This interface updates 3 times per second and allows the user to see and check the signals of each channel (see figure 8). There are 9 small labels placed on a scalp image which shows where the signals come from. These labels show even the quality of connection of each channel with 5 different colors. Gray for not connected sensors to the scalp. Red shows a connection on the scalp not receiving data. When the connection is there but is very weak the color turns orange. A yellow-colored label means that the connection is fair enough and the receiving data is trustable. The color turns green when the connection is as good as possible. This data is received from the Bitbrain SDK by calling two of its Python API's methods. One is getting the connection quality while the other gets the data from each channel.

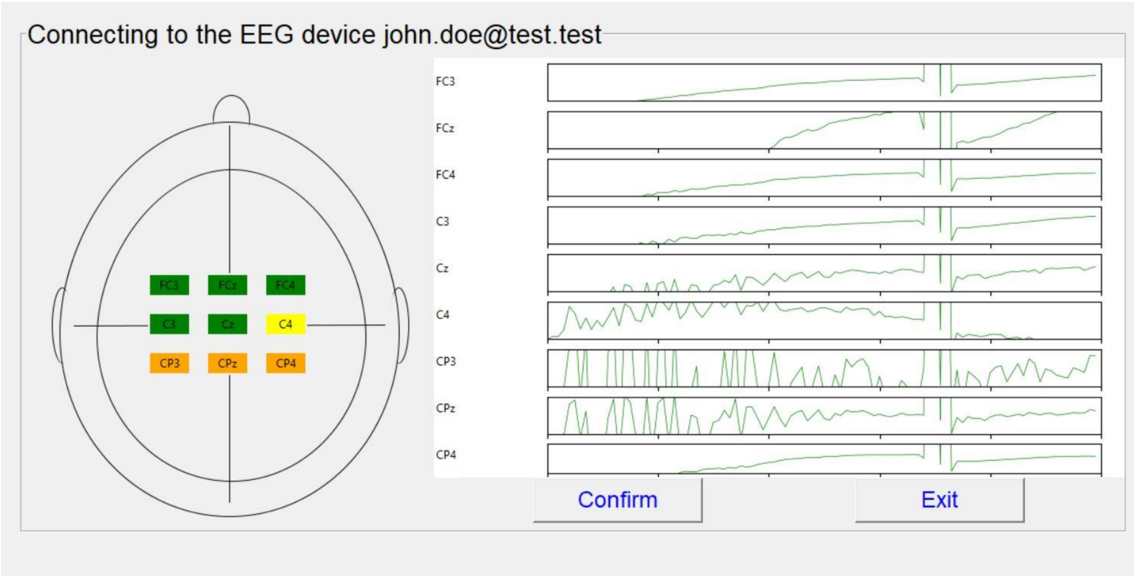

Figure 8. Signal Check GUI

3.3.5 FR6 *(data collection)*, FR9 *(256 signals/sec data transmission)*

While the device is not disconnected, as it is described in section 3.3.3, the application continues calling the Read_Data method in the SDK which receives data from each channel. That is happening by calling several methods in the SDK which control the device. Data shown on the Signal GUI are not stored, and neither send to the server, until the test has begun. When the test begins, application starts to store the data transmitted from EEG cap to a file (described in section 3.3.6) and to send the data to the server. The Mobistudy server takes care of receiving, collecting, and storing the data transmitted from the application.

The data transmits to the application as a packet. Each packet contains 8 samples from each channel. There are 9 channels transmitting samples which means the application receives 9 packets with 8 samples in each, per once. This process continues to up to 32 packets per second (see table 4).

Channel 1

Pack1	Smpl1	Smpl2	Smpl.x	Smpl8
Pack2	Smpl1	Smpl2	Smpl.x	Smpl8
Pack.z	Smpl1	Smpl2	Smpl.x	Smpl8
Pack32	Smpl1	Smpl2	Smpl.x	Smpl8

Channel 2

Pack1	Smpl1	Smpl2	Smpl.x	Smpl8
Pack2	Smpl1	Smpl2	Smpl.x	Smpl8
Pack.z	Smpl1	Smpl2	Smpl.x	Smpl8
Pack32	Smpl1	Smpl2	Smpl.x	Smpl8

Channel y

Pack1	Smpl1	Smpl2	Smpl.x	Smpl8
Pack2	Smpl1	Smpl2	Smpl.x	Smpl8
Pack.z	Smpl1	Smpl2	Smpl.x	Smpl8
Pack32	Smpl1	Smpl2	Smpl.x	Smpl8

Channel 9

Pack1	Smpl1	Smpl2	Smpl.x	Smpl8
Pack2	Smpl1	Smpl2	Smpl.x	Smpl8
Pack.z	Smpl1	Smpl2	Smpl.x	Smpl8
Pack32	Smpl1	Smpl2	Smpl.x	Smpl8

Table 4. Data received from all channels per second.

Each sample value ranges between -200 000 and 200 000 as a float number. These float numbers are converted to the bytes to be send easily to the server. In this case each sample is about 4 bytes. A timestamp is also sent to the server which is converted to Unix timestamp (described in section 3.4.2) from the time when the connection to the EEG device has started. This timestamp is sent together by each sample.

Total transmitted data per second is 256 samples per channel, that is 2304 samples per second in total. This data is stored to a Comma-Separated Values (CSV) file locally and also sent to the Mobistudy server.

3.3.6 FR7 *(CSV file creation)*

When the Signal GUI runs, it calls another method which creates two separate CSV files. One file contains the patient's information, date and time of the file creation, pain level chosen by the user, timestamp for each test task, and a description of beginning and ending of a task (see figure 9). The reason to use the CSV files is that it makes it easier to read data from the files. The file creates columns for each comma sign which is inserted by the application [19]. The application sets the name of the file with date and time stamp to specify in what time the test has taken place.

Another CSV file which stores the collected data has 10 columns within the first one is a timestamp which declares the time of sample received. Each row contains a sample from each channel (see figure 10).

Column1	Column2	Column3	Column4
Patient:	john.doe@test.test		
Date and Time:	2021-04-30 15:41:23.720842		
Pain level:	6.2		
15:42:18:386013	Keep your eyes open	Begins	
15:42:26:478118	Keep your eyes open	Ends	
15:42:26:478118	Keep your eyes closed	Begins	
15:42:34:598621	Keep your eyes closed	Ends	
15:42:34:598621	Pause	Begins	
15:42:40:696987	Pause	Ends	
15:42:40:696987	Message = Imaginary	Begins	
15:42:45:819094	Message = Imaginary	Ends	
15:42:45:819094	Arrow Task A	Begins	
15:42:45:819094	Right		
15:42:49:825011	Right		
15:42:53:876249	Right		
15:42:58:947449	Arrow Task A	Ends	
15:42:58:947449	Pause	Begins	
15:43:7:11178	Pause	Ends	
15:43:7:11178	Arrow Task B	Begins	
15:43:7:12177	Right		
15:43:11:98368	Right		

Figure 9. The CSV file which stores the test tasks events.

Time	FC3	FCZ	FC4	C3	Cz	C4	CP3	CPz	CP4
15:42:6:372024	-66644	-50527	-17345	-59055	18227	-31634	-21612	-3423	-22034
15:42:6:372024	-66665	-50548	-17364	-59080	18423	-30237	-21723	-4824	-21911
15:42:6:372024	-66690	-50572	-17375	-59081	20561	-15881	-19459	2194	-21091
15:42:6:372024	-66695	-50559	-17353	-59043	22163	-5620	-17475	9469	-20721
15:42:6:373021	-66660	-50527	-17332	-59021	20232	-18383	-19195	4915	-21370
15:42:6:373021	-66643	-50527	-17339	-59040	18384	-30537	-21398	-2626	-21885
15:42:6:373021	-66661	-50534	-17343	-59052	18305	-31067	-21822	-5075	-21936
15:42:6:373021	-66674	-17341	-59030	19956	-20260	-19500	4022	-21439	-47635
15:42:6:384364	-66681	-50553	-17353	-59039	22223	-5205	-17474	9290	-20607
15:42:6:384364	-66657	-50521	-17333	-59011	20569	-16262	-18855	5958	-21227
15:42:6:385365	-66633	-50512	-17327	-59028	18453	-30171	-21268	-2146	-21920
15:42:6:385365	-66647	-50530	-17339	-59057	18297	-31319	-21831	-5046	-21905
15:42:6:385365	-66671	-50560	-17349	-59069	19933	-20216	-20157	-167	-21275
15:42:6:385365	-66682	-50557	-17341	-59046	22109	-6051	-17643	8563	-20705
15:42:6:385365	-66658	-50526	-17332	-59019	20920	-14017	-18530	6916	-21144
15:42:6:385365	-66636	-17356	-59064	20190	-18306	-19857	816	-21245	-47511
15:42:6:405471	-66649	-50519	-17331	-59034	18183	-32041	-21908	-5153	-22024
15:42:6:405471	-66667	-50540	-17341	-59052	19604	-22338	-20505	-1279	-21442
15:42:6:407461	-66683	-50539	-17334	-59029	22047	-6465	-17761	8080	-20645
15:42:6:407461	-66660	-50503	-17303	-58984	21214	-12215	-18246	7733	-21069
15:42:6:408461	-66618	-50490	-17301	-58992	18827	-27780	-20801	-420	-21817
15:42:6:408461	-66624	-50512	-17324	-59028	18231	-31768	-21839	-4792	-21951

Figure 10. The CSV file which stores the samples received from each channel.

3.3.7 FR8 *(imaginary test instruction)*

As it is already explained in section 3.3.5 and 3.3.6 the data is collected while the user is guided through a test. The test has 5 tasks that the user should perform in a time of around 17 minutes. As soon as the "Ready" button is pressed on the task GUI (see figure 11), the user must go through the tasks. An admin or user has the control over the time of tasks and the pauses. The time for each task and paus is written on a text file and is stored locally. The file is available for a user to make changes for the task times. The tasks and the time of the tasks as default, are as following:

1. Count down 3 seconds
2. Keep your eyes open 2 minutes
3. Beep sound between the task 2 seconds
4. Keep your eyes closed 2 minutes
5. Pause 5 seconds
6. Show message "Imaginary task" 5 seconds
7. Showing right and left arrows 60 times
8. Shows each arrow 3 seconds
9. Beep between the arrows 2 seconds
10. Pause 2 minutes
11. Showing right and left arrows 60 times
12. Shows each arrow 3 seconds
13. Beep between the arrows 2 seconds
14. Pause 10 seconds
15. Showing message "Thank you" until the user clicks on "Exit" button.

Figure 11. GUI for the test.

3.3.8 FR10 *(show all signals on GUI)*, FR11 *(show few signals on GUI)*

Due to the lack of the time, the requirement by ID FR10 could not be accomplished. And according to importance of the requirement FR10 and its priority the application could be delivered without accomplishing this requirement. However, the FR11 is implemented, and application shows the samples on a graph on the Signal GUI (see figure 8). These signals are shown with color red if connections are weak. The color changes to green when a connection from EEG electrodes to brain receives. Interface updates 3 times per second though it is not so fast to see each action by user. The reason for this graph is to see if the application receives any signals from the brain.

3.3.9 FR12 *(language alternatives)*

At the start screen, the user can choose the application language between three languages, English, Spanish and Swedish (see figure 4). There are three files with the three languages that store each word used in the application. A user clicks on a chosen language button and the application reads each word from the chosen file and uses it as the application language.

3.4 Non-Functional requirements

This section is divided into two subsections. In the first subsection, the accomplished Non-functional requirements are explained. And in the second section, the requirement NFR4 is tested and explained.

3.4.1 NFR1 *(usable)*, NFR2 *(reliable)*, NFR3 *(readable code)*, NFR5 *(visible)*

The code is documented by each row of a new code and each method are explained what they are receiving, what they are doing under the process and what they are returning. The scripts are separated by their functionality and their category. Some of scripts includes a big number of rows and the reason is to be as simple and as fast as possible. An included simplified UML diagram with no attributes and no methods, shows the relation between each script. Due to lack of the knowledge about Python programming language, there are no classes used in this system. But the scripts are separated by their category, and they work together. The SDK files are used in this project is connected via "py_bbt_driver".

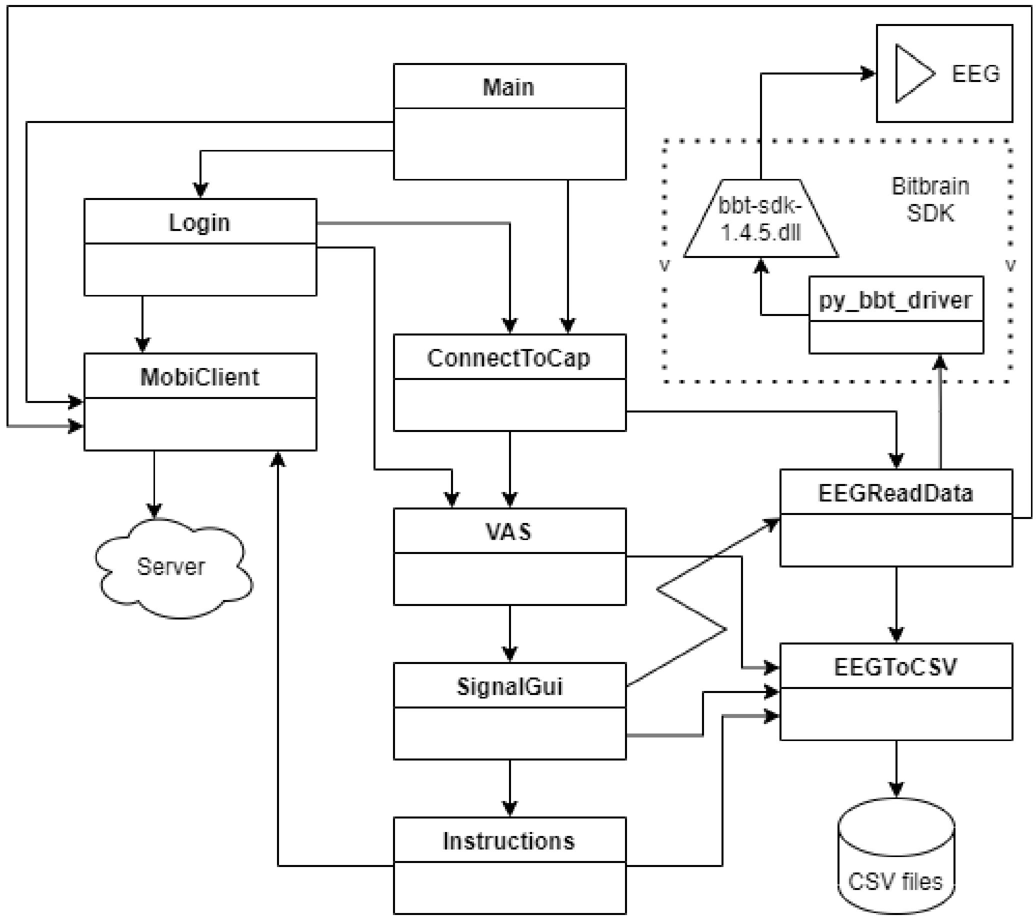

Figure 12. UML diagram

The design is suitable for intended patients with some disabilities. The buttons are designed relatively big and visible so the patients can easily see, reach, and touch them. The fonts and the colors chosen are as simple as possible, so they do not disturb the patient's focus on the test. According to the client the buttons are suitable for the patients involved in the EC project.

3.4.2 NFR4 *(less than 5 seconds latency)*

One of requirement functions and even a part of the first research question is to find out how to transmit data in real-time. By real-time it is meant to have a latency lower than 5 seconds, for transmitting data from the EEG device to the server. A local server which is similar to Mobistudy server has been used in this project. The server stores received data into CSV files which is a similar version to the CSV file stored locally on the platform (see figure 9 and 10). The data file from each channel is stored on a CSV file which also includes one column for time stamp. For the latency test, the idea is to include one more column by two different clock stamps. One of the columns stores the time stamp which shows the time of when the data is sent from the application, measured on the sending device.

The other column stores the time stamp of the time data is received to the server, measured by the server. This timestamp is converted to a Unix timestamp even called Unix Epoch [20].
A test scenario has been designed to measure the latency of data transmission. The idea is to run the server on a different computer than the one where application is running but on the same local network. A tester places the EEG device on his or her head, then the tester disconnects the ground electrode (the ear clip, see figure 3) of the device at a decided time, and notes the time by hour, minute and second. The tester connects the ground again after 20 seconds. This process repeats in two different times, and it happens twice during each process. The disconnection of the ground electrode is observable in the generated signals. The default signal while the ground is disconnected is a float amount with the value of "-1002070.0390625". When connection and disconnection of the ground is performed, the values of each channel change in a way that can be easily observed in the data.
To measure the latency, the disconnection time was compared to the timestamp stored on a file in correspondence of where the signals changed to the fixed lowest value (see table 5 and 6). When disconnecting, the value should decrease to its default value, which is described above. And when connecting the value increases to the signals received from the brain.
An included table with 3 columns and 4 rows explains the times written on a note, for each occasion and each time of the test. The first column includes a code which tells about the occasion and the round in the occasion. To clarify the code, see the row 1 and 2 "Occasion 1, Round 1 and Occasion 1, round 2" in table number 5. This means that the application is run once, but the disconnection and connection of the ground sensor happened twice. The column 2 and 3 shows the time of disconnection and connection of the ground sensor.
To compare the result of the time noted, with the time stored on CSV file on the server, the table number 6 is included. In the second column in the table the converted timestamp is shown as some digits. These Unix numbers are then converted to actual dates and times using the online Epoch Converter [21]. The received date and time tell us about the difference between the time written on a note and the timestamp stored on the file. These differences are the result of the latency for data transmitting from the EEG device to the server. For a better understanding the related cells are shown by the same color.
The test is performed twice as described and got the following result.

O1 + R1	Disconnection = 14:20:00	Connection = 14:20:20
O1 + R2	Disconnection = 14:21:00	Connection = 14:21:20
O2 + R1	Disconnection = 18:30:00	Connection = 18:30:20
O2 + R2	Disconnection = 15:43:30	Connection = 18:43:50

Table 5. The noted time

Timestamp by changed signals	1620908400221	= 14:20:00
Timestamp by changed signal	1620908420025	= 14:20:20
Timestamp by changed signal	1620908460040	= 14:21:00
Timestamp by changed signal	1620908480065	= 14:21:20
Timestamp by changed signal	1620923402002	= 18:30:02
Timestamp by changed signal	1620923422002	= 18:30:22
Timestamp by changed signal	1620924212012	= 18:43:32
Timestamp by changed signal	1620924232018	= 18:43:52

Table 6. The converted times from the CSV file

In the first test with two times disconnection of the ground sensor, and with a space of 1 minute, the result shows a 0 second latency.

In the second test and also with two times disconnection of the ground sensor, but with a space of 13 minutes between both disconnections, the result shows a 2 second latency each time. The maximum measured latency is therefore 2 seconds.

3.5 Usability Test

According to a discussion with the project owner and the limitation about the ethical approval (read section 1.2), it is decided to get help from some relative participants. The selection of participants has also been limited. The main aim was to have between 3 to 7 participants from different genders and ages. Due to lack of time, we contented us by 4 participants. The defined participants are from social network of the researcher, of which two were women and two were men. Of the male participants is one at age 18 and the other at age 21. The two female participants are at age 28 and 58.
As Oates [5] describes it, the participant should be informed about the project and the aim of it. An information about this paper (section 1), the application and the test were given to participants. All participants went through same test and same method for observation data gathering. A mobile phone recorder application was used to record voice of participants' think-aloud thoughts while they talked during the test. Observer is sitting next to participant and take notes of participants' behaviors and navigating through the application. To ensure the participants about non-data gathering, the stored data of each test is removed after the test. The cap is placed on participant's head only to make it feel natural. The amplifier is switched on so the application can be run. For the tests of the application and its usability, a fake server is created which is similar to Mobistudy server. This server collects the data and stores it on a file locally and does not send data to Mobistudy server.

Some pre-information about what to do was explained to participants. These are: 1, An account information, "an e-mail address and password", are written on a note. 2, The amplifier and its turn on and off button. 3, EEG signal colors and meaning of them. The color on the GUI shows which channels have a good connection. 4, The signal waves. 5, What to do in right and left arrow task (Should lift the right or the left arm depending on displayed arrow).

The instructions about the test are explained in the following steps:

Participant:

1- logs in to an account.
2- turns on the EEG cap.
3- chooses a pain level
4- checks the signals and tries to place the EEG cap as good as possible.
5- confirms the connection and goes to the imaginary test level.
6- chooses to mute or unmute the voice guide, by clicking on “Mute” button.
7- follows the instructions and performs each task after another.
8- clicks on “Exit” button when a “Thank you” message is shown

When the test is completed, the researcher removes the data from the memory to respect participants’ privacy. A little pre-designed survey is handed to participant after the test and the participant is asked to rate the different features of the application (see table 5).

3.5.1 Result of the test

Some lack in defined pre-information is detected while the participants participated the test. First, that the first participant was not aware about to connect the ear clip (see figure 1) to his ear to connect the ground of EEG device. One of the participants noticed that she did not know how to place the EEG device and how to adjust it tight to the head so the application could receive good signals. Two of them thought aloud about the meaning of the dark screen without any text on it, while doing the imaginary tasks. One guessed that the dark screen means do nothing and the other one guessed the dark screen as a pause in the test. One issue was that they did not know when the connection was good enough so they could continue to the test. They all got the answer that at least yellow color should be shown for each channel (see figure 8). Yellow and green means that the connection is good enough for gathering data.
These problems were adjusted to facilitate the test for next following participants. The first participant got help during the test.

After each test, the researcher showed the participant where the data files were gathered, removed them while the participant were there, and made participants sure that the files were permanently removed from the local device.

3.5.2 Feedbacks

Feedbacks, thoughts, and notes from participants were analyzed to get a result of the usability test. This result can be used to answer the research question about the usability of the application.

Navigation:

All participant navigated easily through the application. They read each text and understood them easily. They found the buttons very simple designed and easy to find. They guessed correctly what it would happen if they pressed any of the buttons. The only button they were unsure about was the “Exit” button. They all guessed that the button would close the application. One participant asked what if a user accidentally clicks the “Exit” button. After they tried clicking on the button and faced a message that was asking them if they were sure to exit, they appreciated the feature.

Design:

“Few colors used, and very simple designed application puts the focus on the tasks and not on the application” is one of the participant’s feedbacks about the application.

Another important thing in design is feedbacks about any actions in an application. According to Norman [22] a user should know what he or she has done and what he or she is going to do. The feedbacks tell or show the user that for example he or she has pressed a button, and some action is happening. Norman describes further that a good design means that a user should not be confused over an action [22].

Participant behaviors:

Because of the information before the test, participants were already aware and ready for some actions. All participants noticed directly to turn on the device when the application showed that it was trying to connect to the EEG cap, and they found easily how to turn on the device. Some of them tried to put on the device directly after turning it on, while the others waited for the instructions.

To check the signals when the device was placed on the head, they tried to figure out how to tight the device and how to place it correctly. One participant got help with correcting the EEG device on the head while the others found it out after a few tries. Some of them tried to understand if moving body or head would affect the signal waves.

All participants performed the imaginary test in a same way. They tried to follow the instructions but all of them missed the first arrow shown in the imaginary task. They were trying to figure out what to do in this task. On the second round of the arrow task all participants knew what to do.

Rating of the features

The expected results of each function on the user interface are shown in the following table. To have a better understanding of the result, features on the user interface is rated between 0 to 10, where 0 is least satisfied and 10 is most satisfied. Participants are shown in the first column where “m” stands for male and “f” stand for female. The number in the parentheses is the age of the participants. The usability is categorized in 6 categories. Visibility is about how to see the text and buttons. Simplicity explains how simple it is to use the application *(Note that this is only about the application and not the EEG device)*. To receive rating about appearance, colors, lines, size, etc. the design column is included. Feedbacks about each action is another category which is decided to get ratings. Instruction 1 decides for if participant is aware or can guess the next step. Instruction 2 is another category for the test guiding, which is about how clear the task instruction is. The last column shows the average of the rates.

Participant	Visibility	Simplicity	Design	Feedback	Instr.1	Instr.2	Average
1 (18, m)	9	9	7	9	7	8	8.2
2 (21, m)	9	9	8	9	8	8	8.5
3 (28, f)	9	8	8	9	7	7	8.0
4 (58, f)	7	7	8	8	5	7	7.0

Table 7. Features rated by participants

3.6 Summary of the result

The result of reviewing literatures, implementing of the application and its feature, the latency test, and the usability test is summarized in this section. In summary, the research questions are answered. To answer the research questions some literature about similar works and how a similar application would work has been reviewed. The literature review helped the process to have a better understanding about the project, patients, different EEG devices and Bluetooth applications. It even guided the paper through a research structure and gave a comprehensive image over the project. The literature review even helped the paper to have a good structure and plan the project. The choice of methods and how to implement the application is another benefit of the literature review. Developing of the system has been accomplished through the implementation of the required functions, testing them, discuss them, and analyzing them after each implementation and iteratively in several times. These discussions led to a better understanding of the functions and by that, the functions could be improved. To understand the connection between the EEG device and application it was needed to understand the SDK published by Bitbrain company [23]. The SDK packet which

performed the connection and data receiving from the EEG device, included two Python scripts. These scripts helped the project to connect the application to the SDK. The understanding of the SDK has been necessary for the EEG connection. Some issues with the Bitbrain SDK connection were met in the project. To solve these issues a contact to Bitbrain support [24] was needed. This contact helped to get a solution and to succeed the connection to the EEG device. The solution has been about to have the right version of the Python installed on the computer and to know which methods to call from the Bitbrain SDK. The functions are implemented as they were required, and the application performs what it was meant to perform. The developing of application, implementing of functions, bluetooth connecting to EEG device, and transmitting data by a low latency to a server, is a relevant answer to the paper's first research question, which is handled in a following separate subsection.
The latency and usability tests are also designed, managed, and performed that have led to a result. 4 participants have participated the usability test and have been observed while performing the tests. The different phase and levels of the application have also been rated by these participants. The result of this test is answering the second research question.

Usability test result could have been much different if the participants were real patients from the EC-project.

3.6.1 Research Question 1

The result of implementing and developing is a relative answer to the research question: "How can Electroencephalography data be transmitted to a clinician, remotely and in real time, through a computer technology?". To clarify and summarize the result, the answer will be as following steps:

1- Implement a GUI for user
2- Have a bluetooth supported EEG device available
3- Implement the connection to a server
4- Use the device's prepared SDK or any other tools which performs the connection
5- Understand the SDK or the tools
6- Use the right methods to connect and read data from the EEG device
7- Transmit the received data to a server where a clinician has access
8- The application should do as little as possible between receiving the data from EEG and transmitting them to server, to have as low latency as possible.

Without any clinician involved in the project or in the result testing, we assume that the data received to the server means that a clinician can see, read, and analyze the data at the clinic.

3.6.2 Research Question 2

To answer the second research question, "How usable can a such application be?", the result of the usability test can be used. All 4 participants have rated an almost high score to the different phases of the application. The average rates for these different design features are as following:

1- Visibility 34 out of 40
2- Simplicity 33 out of 40
3- Design 31 out of 40
4- Feedback 35 out of 40
5- Instruction 1 27 out of 40
6- Instruction 2 30 out of 40

This result can be interpreted in different ways depending on which situation and what application these are about. For a such application which is only meant to connect, receive data, transmit data, and guide a patient through some tasks this result can give an answer to the question.

4. Prototype demonstration

In this section the explanations are divided into two categories. First category is about who can use this prototype [25], and when and where the prototype can be used. The second category is explaining how the application can be used. This category is a kind of user instructions.

4.1 When, where and by whom the prototype can be used

This prototype can be used by any user who has a Mobistudy account and a Bitbrain bluetooth EEG device to connect to the application. But as far as that, the SCI patients involved in the project are going to use it for a research test. This test will be performed daily in several weeks by a patient who is a participant to the project. This application can even be used by other patients in other situations after the research is accomplished.

4.2 How to use the prototype

To use the prototype, it requires to have the EEG hardware side by side of the prototype. The application can only run on a windows-based platform. It requires also to have an account created on a Mobistudy website [18]. In the first screen the user will log in to his or her account on Mobistudy. On the next screen connection to the EEG device is shown. The application will automatically connect to the device if it is already on. After connecting to the device, the device should now be placed correctly on patient's head, and the ear clip should connect the cap to its ground so the application can receive the data from each channel of device. In the following screen the user is asked to choose the level of pain. The level can be chosen between 0 as no pain, and 10 as the highest level of pain (see figure 6). After choosing the pain level, the application will run another screen where there are 9 dots shown with gray color from the beginning and it will turn to other colors depending on which signal quality it receives. In the same screen the user can now see the brainwaves on a graph. After connecting to the device and receiving data the patient is redirected to a black screen window, where he or she gets some tasks to perform. The first tasks are about opening eyes and keeping them open and closing eyes and keeping them close. The following task is an imaginary task, and the patient does not have to do anything physically, but only imagine. In this level the application shows a left or right arrow which is chosen randomly that means the patient should imagine that he or she is moving or lifting the respectively arm. This task is going on, on two occasions and 60 times for each occasion. After the arrow task A and B are performed, the first task about opening eyes will be asked again. The application will show a "Thank you" message at the end which means the user can now click on "Exit" button to close the application.

5. Analysis

There have been other already existing applications for transmitting data from other EEG devices with cable connections. However, the project described in this thesis needed a simple application for transmitting data from a specific EEG device and guiding patients through a test, while it is gathering the brain activity data. Even though the Bitbrain company has already a data reading and storing application by the name Bitbrain Viewer, such application does not include guiding during the test.

A real-time data is preferred to be received to a clinician through a server, so a patient can submit the samples as he or she is yet at home. Through this paper and this developed application the purpose of transmitting data via bluetooth, and server is achieved which leads to a continued research for the EC project. Subject to time and knowledge the application could be developed by further features and even a signal GUI application for the recipient in the other side of the server. The application transmits data to a CSV file on server, and the CSV file can be read by other applications such as a simple Notepad or a Microsoft Excel program.

All the required functions together helped to understand the needs in the application for a temporary use in EC project. There are several difficulties and some disadvantages that affect the using of the application together with the EEG device. One of them is that the user is not able to see the connection signals during the test. If the connections are lost or any channel is unconnected or even has a bad connection the user will not notice that. The only thing the user is aware of, is a pre-instruction given by the clinician about how to use the device and application, and what to do and not to do during the test. One of the not-to-do instructions is that the user should not remove the device from his or her head during the test. And one of the to-do instructions is to sit still and not move the body during the test but only do what the application asks them to do.

The result shows that the application is accomplished on a way that it gives an answer to the first research question. According to the timestamps on notes and comparison with the timestamps stored on the csv files, the application transmits the data from the EEG device to the server by a latency lower than 5 seconds. This means that the data transmits in a real-time.

According to the result of the usability test the application could get a better design which can explain what the user should do in different situations. Regarding to the second research question the application can be used by a user without major worries and considering that the user will perform the test during a several week period. This means that the user can get to use the application after the first use.

6. Discussion

During this study it has been multitude factors that have affected the result.
Some factors have limited this project and they have been affecting the result. The first factor which cannot be ignored is the lack of the knowledge about the programming language, used in the development. The knowledge could be improved subject to a longer time. By an improved knowledge it would be possible to even improve the design, features, connections, and latency. This means that the other factor which has been affecting the result is the lack of time. Subject to the time, the knowledge could be improved, and the implementation and the research could achieve a better result as well.

The usability has been tested by several participants who were not SCI patients which is another limitation for the research. Even one SCI patient as a participant could be a big help for the usability test. In that case the result of usability test could be more real, and the feedbacks could even improve the usability of application.

Some other factors have affected the research study positively, which have helped the project to achieve this result. One of the most important of them is the availability of the EEG device. Method decisions and planning structures are other opportunities that helped the project to have a clear image. Support from the Bitbrain company which answered some unclear questions and wonderings could be a limitation depending on the answer, but however it helped the development and connection to the EEG device to be implemented. The simplicity of the tools and the availability of the SDK enabled the development, the connection, and the testing of the system.

Understanding the programming language and the Bitbrain SDK have been one of the challenges which complicated the process. To understand and develop the application, I had to do many exercises and to do online research to find features and function's solution. There have been many discussions with supervisor to understand the SDK and how to call its methods. It has been complicating the process and took a while to learn the language and understand the SDK to succeed the implementation.
There are other methods in the SDK which are not used in this application. These methods are about to check: the type of the device, number of available channels, hardware version, battery level, capability of the SD card, is any SD card placed in the device, to reconnect, to start storing on a memory card and other similar methods. These methods have not been needed considering to the requirements.

The focus has been on the goal of this paper and also to answer the research questions. By this focus the work has achieved a result which has been tested in detail. Each function has been tested and the result is included in the paper. The code is available on GitHub [25] and can be tested and run by anyone, and although the EEG device is needed as well as a Mobistudy account or using other similar servers. The figures and the tables show all research and test result in detail. Another test of the application would give the almost related result, depending on the participants for usability test and internet connection for latency test. It is also possible for any other developers to develop a similar application. What they need as a first tool is the related knowledge and the tools as both hardware and software.

The hope is that this application helps the involved researchers in the project to succeed reading, analyzing, and understanding the brain activity signals. This success means a lot in clinical researches to predict pain. The computer technology has always been a help in healthcare, and to predict the pain and avoid it, the computer technology will have an increased and meaningful effect in clinical researches.

By receiving data in real-time and with the knowledge of analyzing and understanding the data to predict the pain, as well as quick action for medical treatment, it would be a help for SCI patients and even other patients to avoid or at least decrease the unwanted pains.

7. Conclusion and further research

The aim of this project has been to answer the research questions and through a developing of an application getting a result that leads to gaining an improved data transmission.

Through the results of the implementing and testing, the first research question will get an answer. The result describes how it is possible to transmit the data in real-time and remotely from a platform. On the way along the result, it is noticeable that it is possible to transmit data from a patient to a clinician through a bluetooth device and remotely. Although there is already other cable connected devices, this study focused on a bluetooth connection to an EEG device. There have been some needs that are identified at the beginning of this paper. Some of these identified tools are the EEG device and its company's SDK concern to the device. These tools beside the knowledge and structures helped this thesis to get a result. A result which is about a developed application which connects to a server, connects to an EEG device, reads data from the device, stores the data, and transmits the data to a server with a latency faster than 5 seconds. The latency less than 5 seconds is one of the requirements which also has been implemented and tested. The result of the latency test is between 0 and 2 seconds which means that the received data, from the EEG device, via bluetooth, through a research platform, and via server, is transmitted to a clinician in a time of 2 seconds. In the research question it is asked if this transmitting can be in real-time. The 2 seconds latency result of the latency test describes that the data transmitting from the device to the clinician can be in real-time.

As for the second research question an application has been developed and been ready for a usability test. With several participants the test has been accomplished and the result of the test has been documented. Participants have been doing well during the test and no complicated tasks have been founded. The usability got a rate between 7.0 and 8.5 out of ten as a result. As feedback for the application the participant mentioned the simplicity of the design and navigating on it. The colors used in the application makes the user to set the focus on the tasks and not on the using of the application. The second research question gets the answer about how usable it is for a patient. The only missing part in this test is relative participants that were not available. This can also lead to an irrelevant feedback from participants. The test result would have been different with real EC-patients as participants.

As for future work, this study can be a start point for developers and researchers who wants a better version of this application. Not only the developers but even other clinical researchers can use this system to test the usability, and the quality of the data transmitting. The project owner has decided to do the usability test on real patients from the EC project by several weeks latency after this paper is submitted. The result of the usability test on real patients means a lot in this study.

Some other functions and features which could be suitable in this application are to see the signals on a separate GUI, to implement the reconnection of the device if it disconnects by any reason. This prototype can even be an example to compare with other similar prototypes. Namely the future researchers can compare their work with this study.

8. References

[1] A. Anjum *et al.*, “Spinal cord injury: Pathophysiology, multimolecular interactions, and underlying recovery mechanisms,” *Int. J. Mol. Sci.*, vol. 21, no. 20, pp. 1–35, 2020, doi: 10.3390/ijms21207533.

[2] “EEG (Electroencephalogram),” *Mayo Clinic*, 2020. https://www.mayoclinic.org/tests-procedures/eeg/about/pac-20393875 (accessed Apr. 15, 2021).

[3] P. F. Brennan and I. Strombom, “Improving health care by understanding patient preferences: The role of computer technology,” *J. Am. Med. Informatics Assoc.*, vol. 5, no. 3, pp. 257–262, 1998, doi: 10.1136/jamia.1998.0050257.

[4] J. Cionek, “EEG recording caps,” *Brain Support*, 2020. https://www.brainlatam.com/blog/eeg-recording-caps-2000 (accessed Apr. 15, 2021).

[5] B. J. Oates, *Researching Information Systems and Computing*. London: SAGE publications Ltd, 2012.

[6] L. A. Machi and B. . McEvoy, *The literature review: Six steps to success*. London: SAGE publications Ltd, United Kingdom, 2009.

[7] R. F. Simon Bennett, Steve McRobb, *Object-Oriented Systems Analysis and Design Using UML*. New York: Mc Graw Hill Education, 2010.

[8] J. Nielsen, “Thinking aloud: The #1 Usability Tool,” *Nielsen Norman Group*, 2012. https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/ (accessed May 05, 2021).

[9] “Bitbrain,” *Bitbrain*. https://www.bitbrain.com/neurotechnology-products/dry-eeg/diadem (accessed Apr. 05, 2021).

[10] Bitbrain, “Bitbrain Viewer,” 2020, [Online]. Available: https://www.bitbrain.com/search/node/bitbrain viewer.

[11] “Github,” 2021. https://github.com/ (accessed May 21, 2021).

[12] L. Thung, “Visual Studio Code: How Microsoft’s ‘any OS, any programming language, any software’ plan is paying off,” *ZDNet*, 2021. https://www.zdnet.com/article/visual-studio-code-how-microsofts-any-os-any-programming-language-any-software-plan-is-paying-off/ (accessed Apr. 15, 2021).

[13] “Bitbrain,” *Bitbrain*, 2011. https://www.bitbrain.com/neurotechnology-products/software/programming-tools (accessed Apr. 19, 2021).

[14] C. G. Cobb, *The project Manager's Guide to Mastering Agile*. New Jersey: John Wileys & Sons, Inc, 2015.

[15] I. Burnstein, *Practical Software Testing*. New York: Springer Publishing Company, 2003.

[16] M. Rabek and K. Žakova, "Ball Levitation Using EEG Headset via Bluetooth," *ICETA 2018 - 16th IEEE Int. Conf. Emerg. eLearning Technol. Appl. Proc.*, pp. 457–462, 2018, doi: 10.1109/ICETA.2018.8572045.

[17] M. Margolis, *Arduino Cookbook*. O'Reilly Media, Inc, 2011.

[18] "Mobistudy," *Malmö University*, 2021. https://mobistudy.org/ (accessed Apr. 19, 2021).

[19] Bigcommerce, "BIGCOMMERCE," 2020. https://www.bigcommerce.com/ecommerce-answers/what-csv-file-and-what-does-it-mean-my-ecommerce-business/ (accessed May 06, 2021).

[20] Unix, "UnixTimeStamp," 2021. https://www.unixtimestamp.com/ (accessed May 13, 2021).

[21] Epoch, "Epoch converter," 2021. https://www.epochconverter.com/.

[22] D. A. Norman, *The design of everyday things*. Cambridge, Mass: MIT Press, 2013.

[23] Bitbrain, "Bitbrain." https://www.bitbrain.com/ (accessed May 10, 2021).

[24] Support, "Bitbrain Contact," 2021. https://www.bitbrain.com/neurotechnology-contact (accessed Apr. 05, 2021).

[25] R. Samandari and D. Salvi, "Github," 2021. https://github.com/23Rohan/MobiEEG (accessed May 21, 2021).