Cairo University

Faculty of engineering

Department of Computer Engineering

Psionica

A Graduation Project Report Submitted

to

Faculty of Engineering, Cairo University

in Partial Fulfillment of the requirements of the degree

of

Bachelor of Science in Computer Engineering.

Presented by

Islam Ahmed Mohamed

Mohamed Abobakr Gad

Mohamed Ibrahim Gaballah

Omar Tarek Mohamed

Supervised by

Prof. Ihab Talkhan

19/7/2022

Abstract

There are millions of disabled people who can't do their daily tasks without relying on others, which makes them feel that they are a burden to their families and even society, all of which badly affects their mental health.

Thanks to recent technological evolution, we can tackle problems and confront challenges we didn’t imagine that we could do one day. Among the notable technologies in this revolution, we have Artificial Intelligence, the Internet of Things, Neuroscience, and many others we used in this project to address this problem.

Psionica helps Quadriplegic patients to communicate with people and control electronic devices by reading and analyzing brain signals using an Electroencephalography (EEG) headset. The actions inferred from these signals are then transferred to a mobile application through Wi-Fi.

The mobile application enables the patient to control his phone using his brain signals only. This is done by taking brain signals and applying signal processing techniques then classifying the action using machine learning techniques to control a virtual mouse pointer created by the app. The mobile application will enable patients or users to do many essential tasks such as: controlling the smartphone, controlling a wheelchair, and controlling a laptop mouse pointer. It can also monitor and collect statistics on his status to be used in case of a diagnosis or emergency.

الملخص

هناك الملايين من المعاقين الذين لا يستطيعون أداء مهامهم اليومية دون الاعتماد على الآخرين ، مما يجعلهم يشعرون بأنهم عبء على عائلاتهم وحتى المجتمع ، كل ذلك يؤثر بشكل سيء على صحتهم النفسية.

بفضل التطور التكنولوجي الحديث ، نحن قادرون على معالجة المشاكل ومواجهة التحديات التي لم نتخيل في يوم من الأيام أن نستطيع معالجتها. من بين التقنيات البارزة في هذه الثورة ، لدينا الذكاء الاصطناعي وإنترنت الأشياء وعلم الأعصاب والعديد من التقنيات الأخرى التي استخدمناها في هذا المشروع لمعالجة هذه المشكلة.

يساعد سايونيكا مرضى الشلل الرباعي على التواصل مع الناس والتحكم في الأجهزة الإلكترونية من خلال قراءة وتحليل إشارات الدماغ باستخدام سماعة تخطيط كهربية الدماغ. يتم بعد ذلك نقل الإجراءات المستنبطة من هذه الإشارات إلى تطبيق الجوال عبر الواى فاى.

يتيح التطبيق للمريض التحكم في هاتفه باستخدام إشارات دماغه فقط. يتم ذلك عن طريق أخذ إشارات الدماغ وتطبيق تقنيات معالجة الإشارات ثم تصنيف الإجراء باستخدام تقنيات التعلم الآلي للتحكم في مؤشر الماوس الافتراضي الذي تم إنشاؤه بواسطة التطبيق. سيمكن تطبيق الهاتف المحمول المرضى أو المستخدمين من القيام بالعديد من المهام الأساسية مثل: التحكم في الهاتف الذكي ، والتحكم في كرسي متحرك ، والتحكم في مؤشر فأرة الكمبيوتر المحمول. كما يمكنه رصد وجمع بعض الإحصائيات عن حالته لاستخدامها في حالة التشخيص أو الطوارئ.

ACKNOWLEDGMENT

We would like to express our thanks and gratitude to our supervisor Prof. Ihab Talkhan for his excellent advice, help, support, respect, and guidance throughout this year.

We also would like to thank our doctors for their phenomenal efforts. Lastly, we would like to thank our families for their continuous support.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
BCI	Brain Computer Interface
CapEx	Capital Expenditures
CSP	Common Spatial Interface
CSS	Cascading Style Sheets
Db4	Daubechies 4
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
DTFT	Discrete-Time Fourier Transform
DWT	Discrete Wavelet Transform
EEG	Electroencephalography
EGP	Egyptian Pound
ERP	Event-Related Potential
FFT	Fast Fourier Transform
FBCSP	Filter Bank Common Spatial Pattern
FT	Fourier Transform
GDF	General Data Format
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IOT	Internet of Things
IP	Internet Protocol
KNN	K Nearest Neighbors
MI	Motor Imagery
Npy	NumPy
OpEx	Operating Expenses
OVR	One-Vs-Rest
PCA	Principal Component Analysis
PSD	Power Spectral Density
SBFS	Sequential Backward Floating Selection

SSVEP	Steady State Visually Evoked Potential
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
SW	Software
UI	User Interface
USA	United States of America
USD	United States Dollar
UX	User Experience

List of Symbols

ω	Angular Velocity
ψ	Mother wavelet function

Contacts

Team Members

Name	Email	Phone Number
Islam Ahmed Mohamed	eslam1092@hotmail.com	+201113586932
Mohamed Abobakr Gad	mo.abobakr.11@gmail.com	+201119554431
Mohamed Ibrahim Gaballah	mohammedibrahimgaballah@gmail.com	+201145916604
Omar Tarek Mohamed	omartarek9984@gmail.com	+201226286647

Supervisor

Name	Email	Phone Number
Prof. Ihab Talkhan	italkhan@cu.edu.eg	+201223105504

This page is left intentionally empty

Chapter 1: Introduction

Psionica helps Quadriplegic patients to communicate with people and control electronic devices by reading and analyzing brain signals using an Electroencephalography (EEG) headset. The actions inferred from these signals are then transferred to a mobile application through Wi-Fi. The mobile application will enable patients or users to do many essential tasks.

1.1. Motivation and Justification

There are millions of disabled who can't do their daily tasks without relying on others, which makes them feel that they are a burden to their families and even society, all of which badly affects their mental health. Moreover, we chose this problem because one of the team members has a quadriplegic relative who inspired us to tackle this problem and try to find a solution to make his life and others' lives a bit easier and more independent.

1.2. Project Objectives and Problem Definition

Our objective is to offer a solution for disabled people through a mobile app that could help them in doing their essential daily tasks such as controlling their wheelchair, controlling their laptop mouse, notifying their doctor in case of emergency, and giving insights about their health.

1.3. Project Outcomes

Our project outcome is a mobile app controlled by an EEG headset that reads and analyzes the user's brain signals providing three main features which are Controlling his wheelchair, laptop mouse, and notifying the doctor in case of emergency and giving insights about the user's health.

1.4. Document Organization

In this document we cover all the important details of our project in order to help the readers to understand the whole project. It is divided into chapters and each one covers a certain topic about our project. Chapter one gives a brief introduction to our project showing our motivation, project objectives, and project outcomes. Chapter two contains a market feasibility study which shows our targeted customers, competitors, and financial analysis. Chapter three contains the literature survey which focuses on what is currently used in practice and the needed concepts that helped us to come out with this project. In chapter four we discuss the system design and architecture of our project in full detail. Chapter five covers all our testing trials, setups, and environment.

Ending with chapter six which concludes the whole report and shows the challenges we faced, the experience we acquired, and our intended future work to enhance our project.

Chapter 2: Market Feasibility Study

Making one's life who suffers from totally counting on others for doing his daily tasks more independent is like a lifeline. So, we believe that one day, there won't be a quadriplegic that doesn't use our project if it is affordable to him.

2.1. Targeted Customers

Our targeted customers are the disabled, especially the quadriplegic.

2.2. Market Survey

research involves analyzing a given market in order to gain insight into the buying potential and attributes of the target audience for a product or service. It is like a crystal ball. It helps provide predictive analytics for a new concept, product, or service before it is introduced into a market. As a result, brands can make strategic changes to a concept before launch and are likely to see a higher return on investment [1]. So, as we introduced before, we don't have competitors who directly provide the same product. Thus, we will introduce some competitors that provide products near to our idea.

2.2.1. Eye-controlled wheelchair

Figure 2.1: Eye-controlled wheelchair

This was a project developed by students at the school of engineering, University of Maryland, college park, USA. This project proposes a system to aid people with motor disabilities by restoring their ability to move effectively and effortlessly by utilizing an eye-controlled electric wheelchair [2].

2.2.2. EEG Controlled Wheelchair

Figure 2.2: EEG controlled wheelchair

This was a project developed by students at the Faculty of Engineering and Technology, Multimedia University, Malaysia. This project proposes a system to aid people with motor disabilities by restoring their ability to move effectively and effortlessly by utilizing their brain signals to control the wheelchair. They were using the Emotiv EPOC headset to record the EEG signals. They ended with average accuracy on 5 subjects equal to 76% and the highest accuracy was on the fourth subject equal to 85% [3].

2.3. Business Case and Financial Analysis

2.3.1. Business Case

The components of our product are:

- EEG headset
- Electric wheelchair
- Mobile app

We will set our price to be covering the product cost plus a little profit for the first 2 years. Our project cost is 1,675 United State Dollars (USD) which is currently equal to 31,626 Egyptian Pound (EGP). We will initially put only 10% profit. Therefore, the price for the first 2 years will be 1842 USD which is currently equal to 34,788 EGP. For the first year, we expect our sales to gradually increase and manage to sell about 500 units. Then, we will sell about 1500 units during the second year.

2.3.2. Financial Analysis

We calculated our project's cash flow as shown in Figure 2.3. It consists of two parts. The first part is capital expenditures (CapEx) which are one-time spending. Our Capex was the price of our four laptops, and the mobile app deployment fees. The second part is operating expenses (OpEx) which are recurring payments. Our Opex was the salaries of Software (SW) engineers developing the product, cost of product components per each unit like the electric wheelchairs and EEG headset, marketing expenses, salesmen's salaries, rent for our company's place, and bills like gas, water, and electricity. Our revenue comes from our sales only. We are expecting to start selling our product in September 2022, increasing the number of units that would be sold monthly until it reaches 65 and we keep selling 65 units a month.

So, as calculated in our project's cash flow, our break-even point would be in May 2023 which is reasonable according to our product's price and the number of sold units.

		Jan-22	Feb-22	Mar-22	Apr-22	May-22	Jun-22	Jul-22	Aug-22	Sep-22	Oct-22	Nov-22	Dec-22	Jan-23	Feb-23	Mar-23	Apr-23	May-23	Jun-23
CapEx																			
	Laptop	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500
	app deployment fees	0	0	0	0	0	0	0	470	0	0	0	0	0	0	0	0	0	0
OpEx																			
	Salaries	32000	32000	32000	32000	32000	32000	32000	32000	32000	32000	32000	32000	32000	32000	32000	32000	32000	32000
	Electric Wheelchair	0	0	0	0	0	0	0	535500	535500	535500	535500	535500	535500	535500	535500	535500	535500	535500
	EEG Headset	0	0	0	0	0	0	0	792120	792120	792120	792120	792120	792120	792120	792120	792120	792120	792120
	Rent	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
	Bills	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500
	Marketing	0	0	0	0	0	0	0	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000	20000
	Sales	0	0	0	0	0	0	0	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
SubTotal		38000	38000	38000	38000	38000	38000	38000	1396090	1395620	1395620	1395620	1395620	1395620	1395620	1395620	1395620	1395620	1395620
Accumulative		38000	76000	114000	152000	190000	228000	266000	1662090	3057710	4453330	5848950	7244570	8640190	10035810	11431430	12827050	14222670	15618290
Revenue																			
	Units Count	0	0	0	0	0	0	0	0	10	20	40	50	55	55	60	60	65	65
	Units Sales	0	0	0	0	0	0	0	0	347880	695760	1391520	1739400	1913340	1913340	2087280	2087280	2609100	2609100
SubTotal		0	0	0	0	0	0	0	0	347880	695760	1391520	1739400	1913340	1913340	2087280	2087280	2609100	2609100
Accumulative		0	0	0	0	0	0	0	0	347880	1043640	2435160	4174560	6087900	8001240	10088520	12175800	14784900	17394000
Profit		-38000	-76000	-114000	-152000	-190000	-228000	-266000	-1662090	-2709830	-3409690	-3413790	-3070010	-2552290	-2034570	-1342910	-651250	562230	1775710
Taxes																			443927.5
Net Profit																			1331782.5

Figure 2.3: Cash flow table

So, as a result of our search, market survey, and financial analysis we noticed the following observations:

- Quadriplegic percentage is about 0.02 % of the United States of America (USA) population according to the National Library of Medicine [4], hence this percentage tends to be more if generalized all over the world
- We are the first to provide this complete product package in the market
- Our product is not limited to a specific region but is a global product
- Our product pricing is reasonable and somehow affordable

Accordingly, by assuming only 5% of Quadriplegics would buy our product with our settled price, therefore our project is feasible.

Chapter 3: Literature Survey

This chapter is divided into two parts. In the first part, we will discuss the backgrounds that are necessary for a complete understanding of our project. The topics we will cover are Brain-Computer Interface (BCI) and classification of signals. In the second part, we will give a literature review of the recent publications related to our project.

3.1. Background on BCI

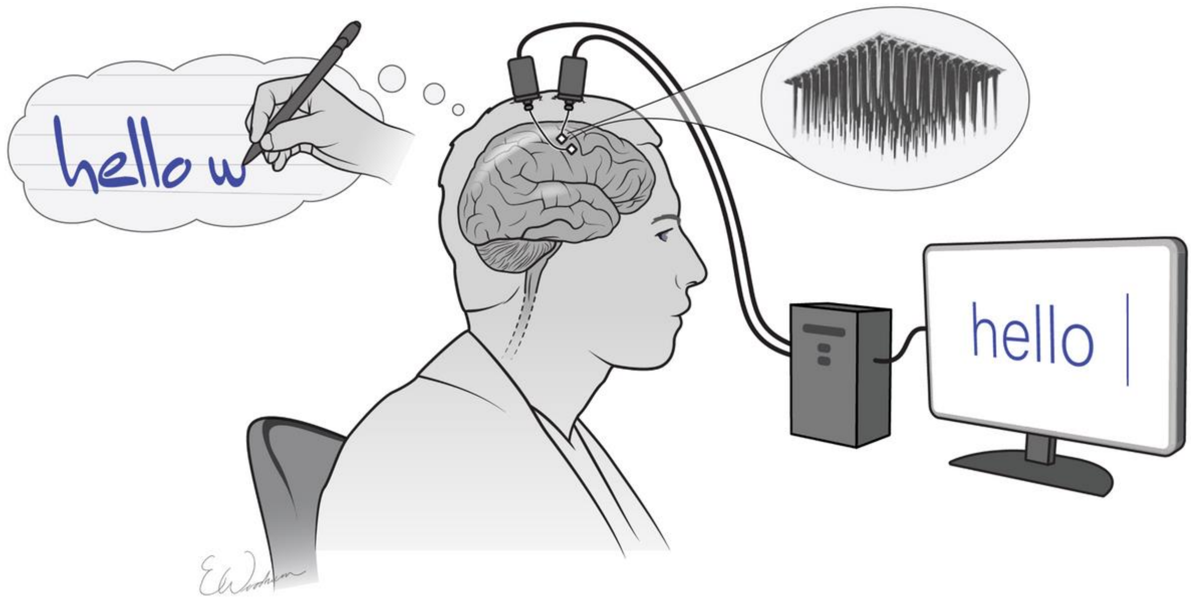


Figure 3.1: Illustration of the operation principle of BCI

3.1.1. BCI definition

Brain-computer interface is a technology that enables users to control external devices using their brain signals. It provides a way of bypassing the neuromuscular system. This allows humans with disabilities to control external devices without physical interaction. The basic idea is to create an interface that enables the communication between the brain and the device by transferring the brain signals directly to the device [5]. So, for example, to move a mouse cursor, the signals carrying the command are transmitted to the computer directly, instead of taking the regular neuromuscular route to the hand.

In order to infer the command that the user wants, electric signals from neurons are collected and then passed to a computer system that translates them into actions using machine learning techniques. The brain signals are collected either by non-invasive electrodes or by implanting invasive devices into the brain. One of the many benefits of BCI technology is helping people with paralysis to control an electric wheelchair or write a book and use their mobiles. The only requirement to be able to use BCI is conscious thought.

3.1.2. EEG-based BCI

Electroencephalography is one of the most well-known techniques that are used in the implementation of non-invasive BCI systems. EEG is a method to record brain signals by recording the electrical activity on the scalp with the aid of non-invasive electrodes [6]. Positioning the scalp electrodes follows the 10-20 system which is the standard for recording scalp EEG [7].

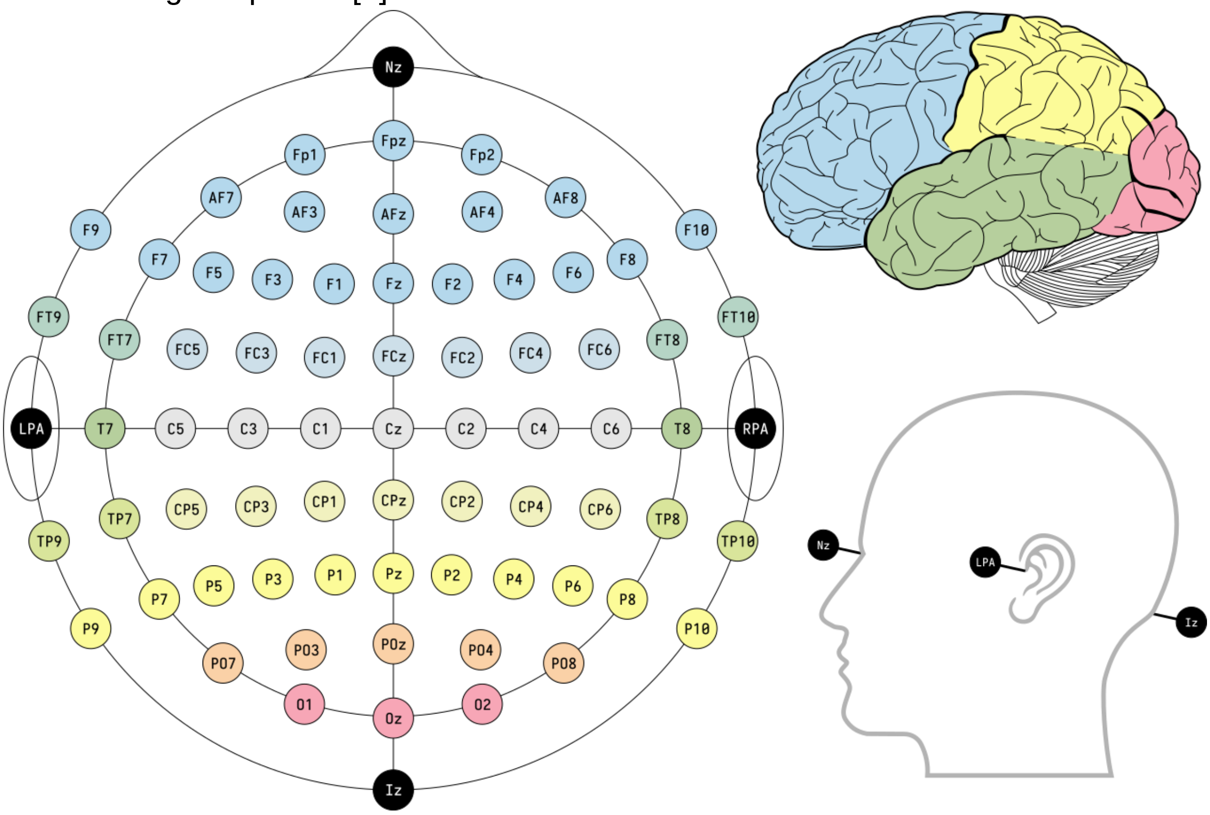


Figure 3.2: The 10-20 system

The resulting electrogram is then diagnosed to perform a specific task such as diagnosing epilepsy or inferring a command for BCI. Some of the most commonly used paradigms of EEG-based BCI are Event Related Potential (ERP), Steady State Visually Evoked Potential (SSVEP), and Motor Imagery (MI).

3.1.2.1. ERP - P300

Event Related Potentials are the EEG changes that occur in response to a specific event or stimulus. These changes represent the systematic way of processing the stimulus in the brain. Different ERP waveforms are classified according to their amplitude and latency. The most discussed use of ERP is the P300. The waveform of the P300 is shown in Figure 3.3.

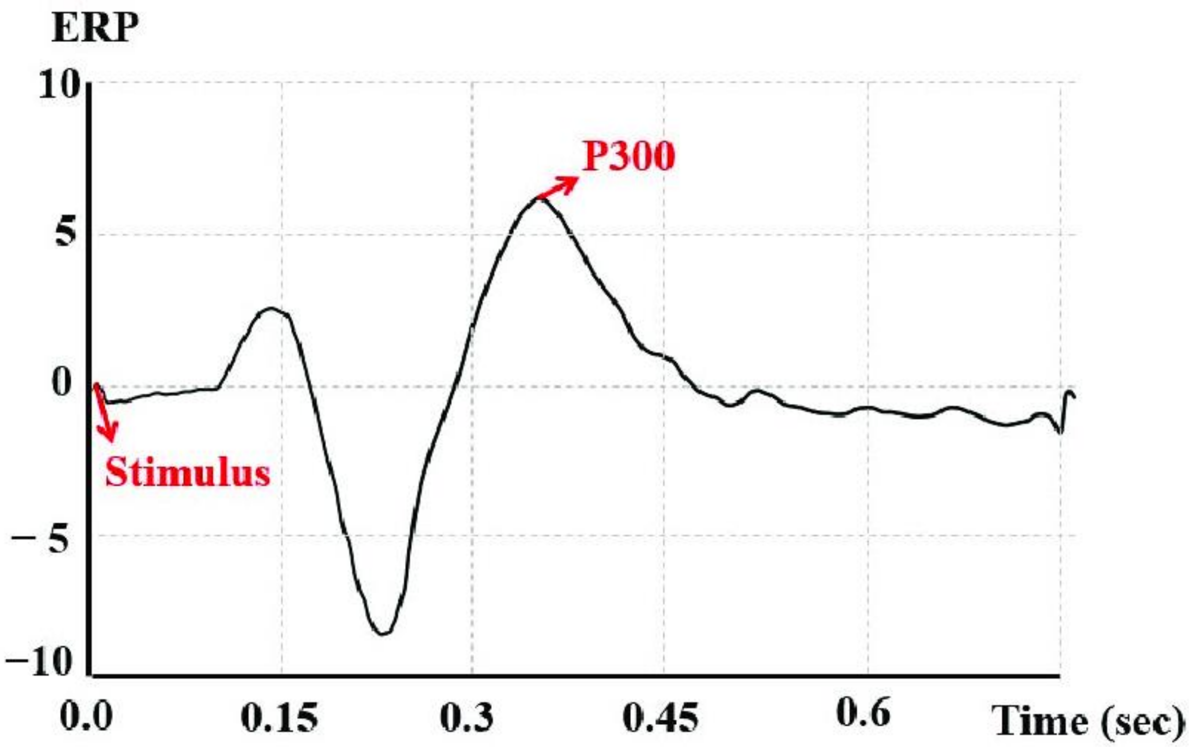


Figure 3.3: P300 waveform

P300 is usually elicited by an oddball paradigm with the odd event acting as the stimulus for the P300. Its waveform is a positive deflection that is peaking - typically - around 300ms after the stimulus [8]. Its latency depends on the difficulty of discriminating the stimulus as well as the mental performance of the subject (Shorter latency indicates a superior mental performance compared to longer latencies). On the other hand, the amplitude of the peak depends on the improbability of the stimulus (greater information results in larger waves) [9].

P300 is usually used in creating spellers where a matrix of letters is displayed with the rows and columns of the matrix flashing randomly one at a time. The wave could be elicited when the desired row or column flashes, yielding the detection of the desired letter [10].

3.1.2.2. SSVEP

Steady State Visually Evoked Potentials, similar to ERP, are electric changes in the brain in response to a stimulus, with the difference being the type of the stimulus. When the subject focuses on a target that is flickering with a specific frequency, the brain generates electrical activity with the same frequency or a multiple of it [11]. SSVEP has a high signal-to-noise ratio and is considered relatively immune to artifacts, thus it provides high accuracies when used in BCI systems [12].

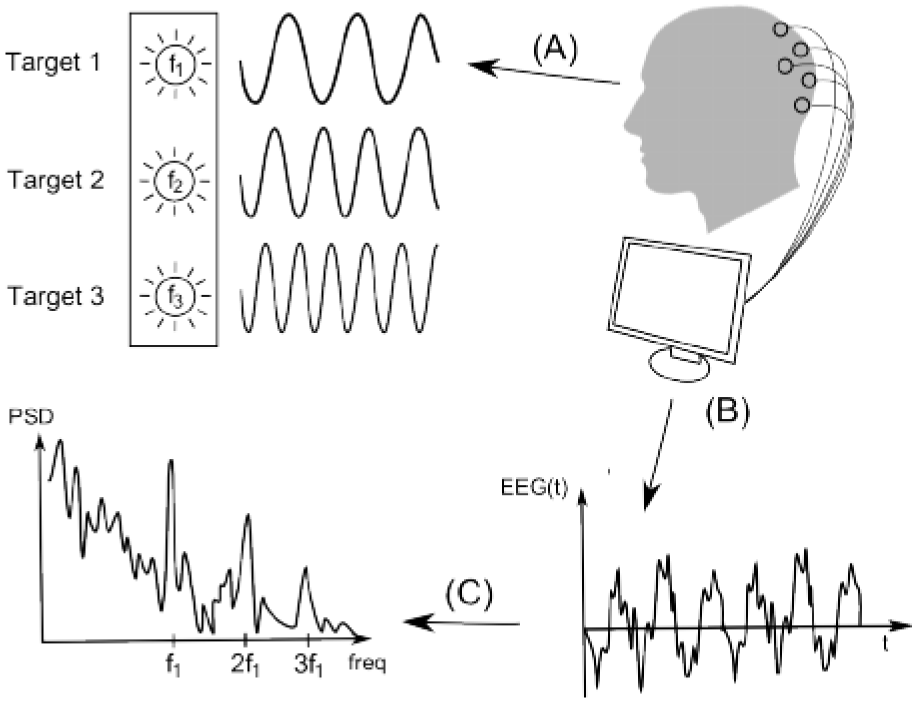


Figure 3.4: SSVEP-based BCI

Most common way of designing a BCI system that relies on SSVEP is by displaying several targets to the user, each target flickers with a different frequency. Targets could represent symbols, numbers, letters or commands that the user wants to select one from [13]. The user is asked to focus on the desired target. The target is then deduced by analyzing the Power Spectral Density (PSD) of the EEG signals.

3.1.2.3. Motor Imagery

Motor Imagery is the process where a person imagines performing the action without actually performing it [14]. This is the most convenient paradigm for designing BCI for disabled people as most of them can't perform the action. In MI-based BCI, the EEG signals of the user are recorded while imagining performing the action. Then, by using machine learning techniques, the action is classified and sent to the external device to be executed. MI suffers from the same problems (noise and artifacts) as other paradigms [15] in addition to other problems such as inter-subject variability and random noise [16]. Random noise poses a bigger challenge for classification in MI-based systems because waveforms for actions are not known in advance, unlike ERP or SSVEP.

Unlike other paradigms, MI doesn't need an interface that creates a stimulus. Theoretically, all the user needs to use the device is to wear the headset and think about the desired command. MI has gained a lot of popularity because of this easy and intuitive way of operation [17].

3.2. Background on classification of EEG

Electroencephalogram consists of multiple signals that are the difference between the output of each electrode with a reference electrode (obtained by applying a differential amplifier). Therefore, Digital Signal Processing (DSP) techniques are used to analyze the signals and extract useful information. This is usually achieved by transforming signals. This information is then used to train a machine learning model. Different paradigms of EEG have different difficulties when it comes to classification. Next, we will discuss the most commonly used discrete transforms in EEG classification and the most common feature extraction methods.

3.2.1. Transforms

3.2.1.1. Discrete Fourier transform

Fourier transform (FT) decomposes signals into their frequency components, that is each frequency that exists in the signal along with its magnitude. In order to apply the Discrete-Time Fourier Transform (DTFT) on finite data, Discrete Fourier Transform (DFT), which is based on sampling the DTFT, is used. DFT is usually implemented using the Fast Fourier Transform (FFT) which is an efficient implementation for the DFT [18]. The result of the DFT is of the same size as the input. DFT is an invertible operation which means that we can reconstruct the original signal from the transform. DTFT is represented by the following equation [19]:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$

DFT is represented by the following equation [20]:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}}$$

Fourier transform gives information about what frequencies exist in the signal as well as their amplitude but doesn't provide information about their position in the signal in the time domain; this is why it is ideal for stationary signals, which are signals whose spectral contents do not change with time.

3.2.1.2. Short-time Fourier transform

In order to overcome the poor time resolution problem of DFT, Short-Time Fourier Transform (STFT) was developed. It provides a time-frequency representation of the signal [21]. The idea is to divide the signal into smaller segments and compute the Fourier transform for each segment separately then add them all up. The signal is assumed to be stationary within the segment. STFT is implemented by taking a window function of a fixed length and moving it along the signal and computing the FT at each section. The procedure is illustrated in Figure 3.5.

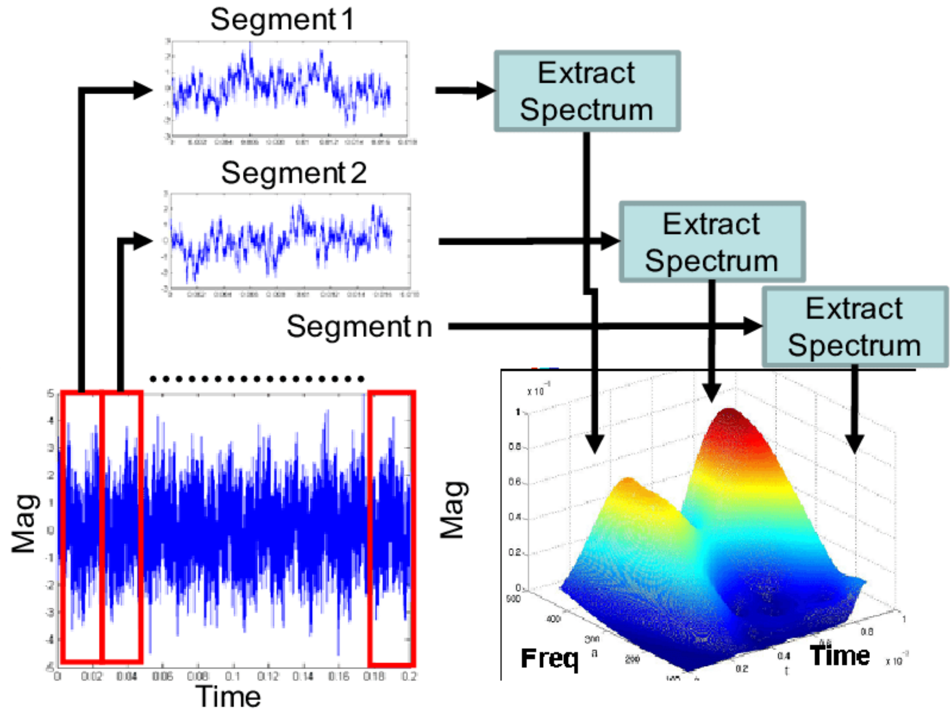


Figure 3.5: STFT explanation

The output is usually plotted using a spectrogram that views the relationship between frequency and amplitude in the different segments. An example of a spectrogram is shown in Figure 3.6.

The fixed window size of the STFT means that both time and frequency resolutions have to be fixed for the whole signal. Another limitation is that since the window size is finite, the frequency resolution decreases. Generally, low-frequency components need a high-frequency resolution as they last for a long period of time while high-frequency components need a high time resolution as they appear as short bursts in the time domain. If we decrease the size of the moving window, we get a higher time resolution but low-frequency resolution. On the other hand, increasing it increases the frequency resolution and decreases the time resolution. This trade-off is the source of the STFT limitations.

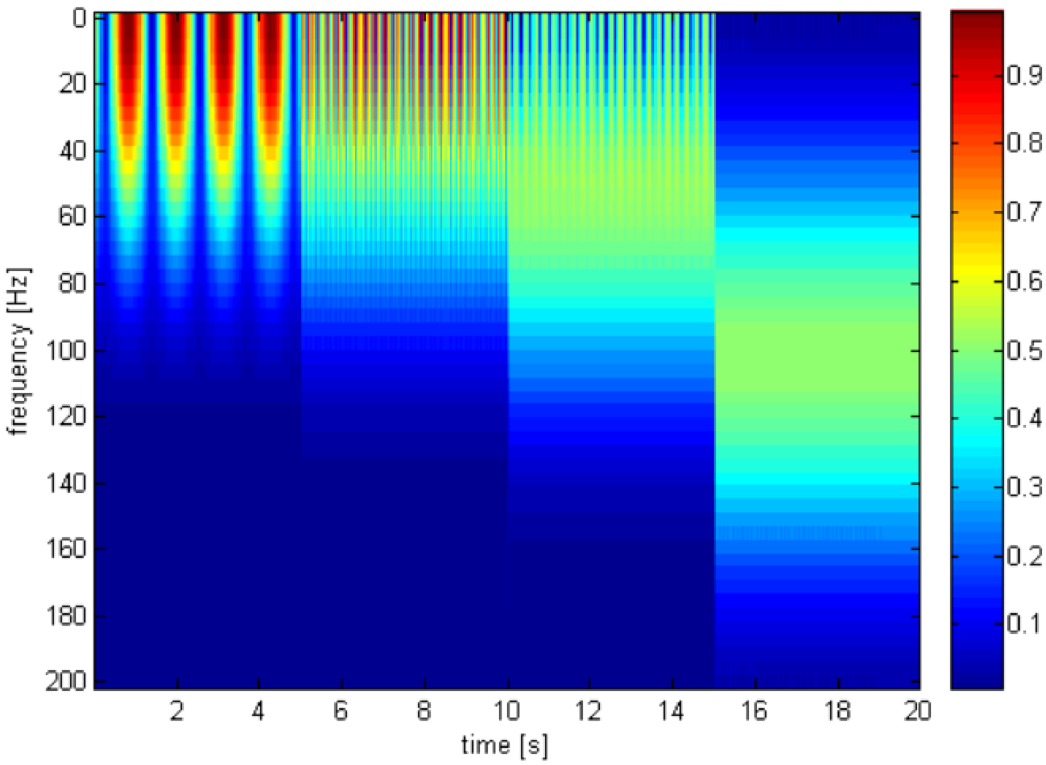


Figure 3.6: Example of a spectrogram

3.2.1.3. Discrete wavelet transform

Discrete Wavelet Transform (DWT) overcomes the limitations of STFT as it provides a multi-resolution time-frequency representation of signals. Instead of using trigonometric functions as basis, a set of mutually orthogonal wavelet functions are used. Wavelets are wave-like oscillations that are spatially localized which means they have a nonzero value only over a part of the total signal length. The two most important properties of wavelets are that they can be translated and scaled [22]. Translation means changing their spatial location while scaling means stretching or squishing the wavelet. We slide the wavelet on the signal and multiply them at each point to get the coefficient at that point for that particular scale. This operation is known as convolution [23].

$$F(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} f(t)\psi^*(\frac{t-\tau}{s})dt$$

In the previous equation, Larger s means a more stretched wavelet where ψ symbol represents the mother wavelet function which is the wavelet function without translation or scaling. The convolution is repeated with different scales to get the response of different frequencies starting with a small value and progressively increasing the scale.

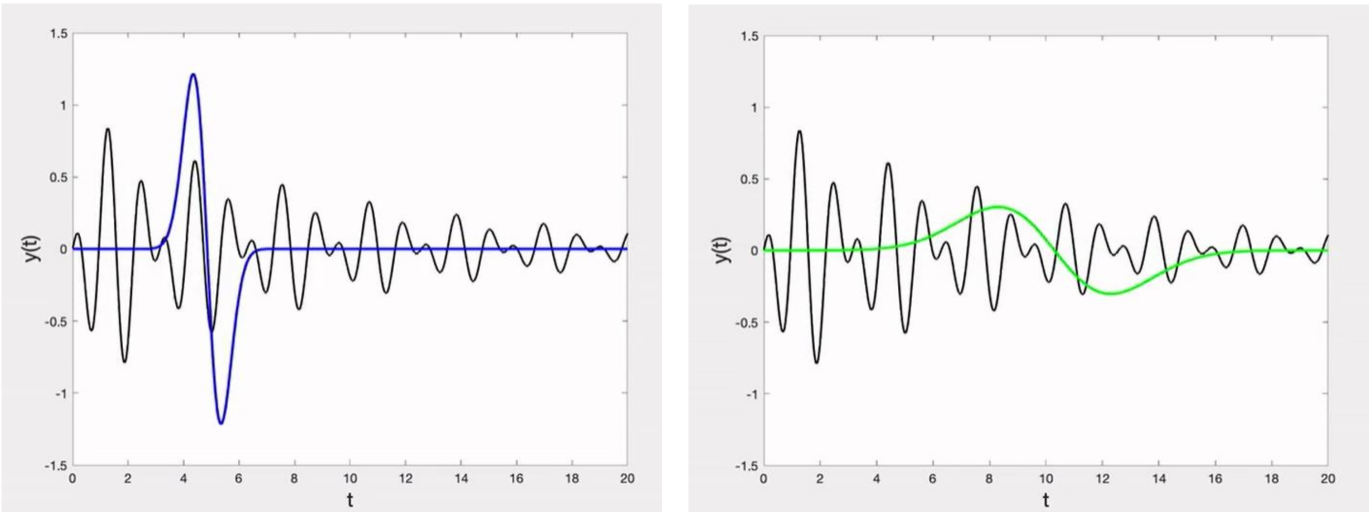


Figure 3.7: Convolving wavelets of different scales with the signal

In DWT, the wavelets are discretely sampled. The DWT formula is given by:

$$D[a, b] = \frac{1}{\sqrt{b}} \sum_{m=0}^{p-1} f[t_m]\psi[\frac{t_m - a}{b}]$$

a and b in the equation are τ and s in the continuous formula. If a and b are dyadic (based on powers of 2) then the DWT can be computed efficiently by multilevel decomposition which is done by passing the signal into low pass and high pass filters to get the approximation and details coefficients respectively [24]. The low pass filter removes some details, so its output is considered an approximation of the signal while the output of the high pass filter contains only the details. The step of applying filters is called decomposition. In multilevel decomposition, this step is repeated on the approximation several times to get different levels of details. Since the output of each filter is the same length as the original, both outputs are downsampled. The multilevel decomposition is shown in Figure 3.8.

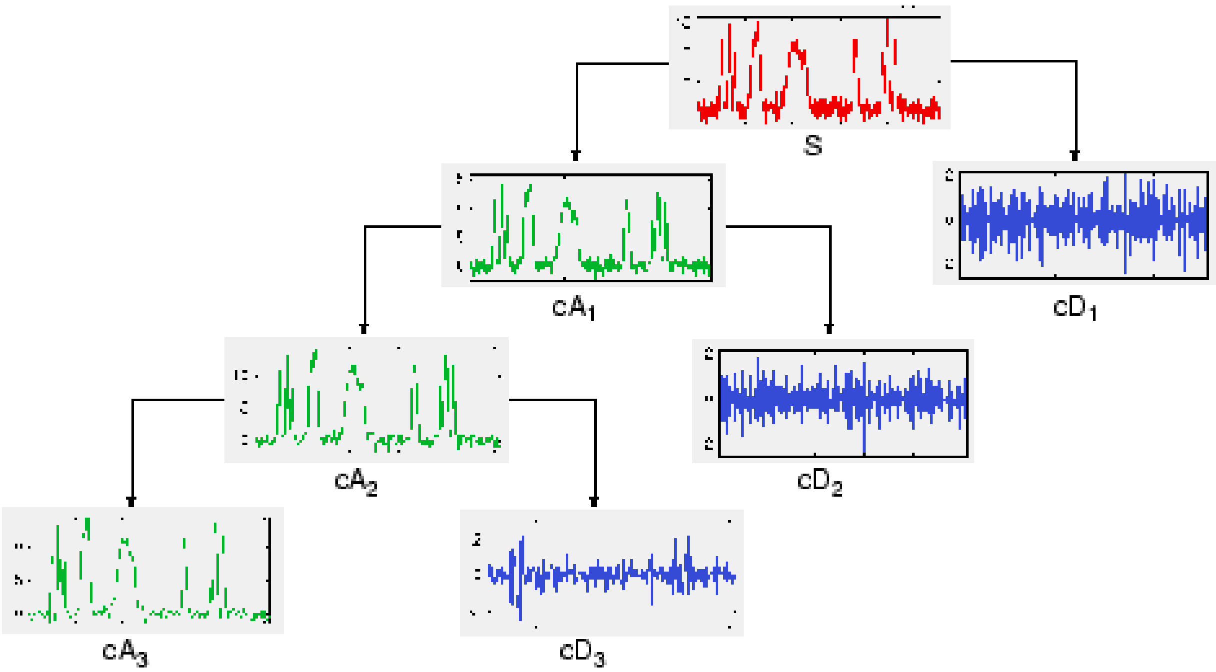


Figure 3.8: multilevel decomposition tree

DWT is invertible so the original signal can be reconstructed from the wavelet coefficients. There is a wide variety of wavelets that can be used. If the characteristic shape of the signal features is known, then we can choose the wavelet that best matches that shape.

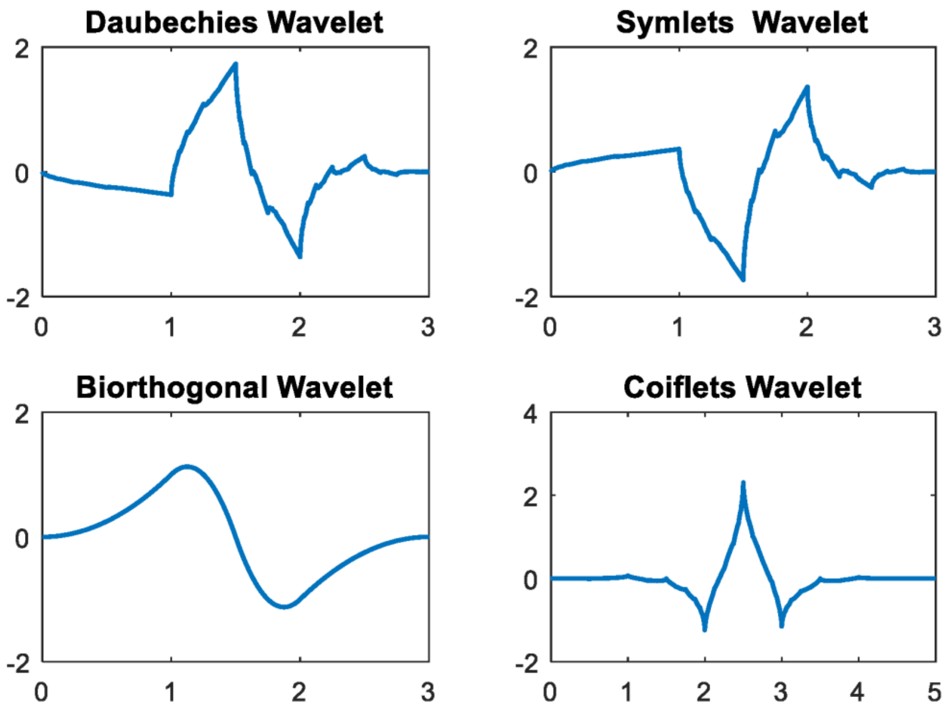


Figure 3.9: Example of wavelets

3.2.2. Feature Extraction

After transforming the signals using DSP techniques, they become more suitable for feature extraction. Next, we will discuss the most commonly used feature extraction methods in EEG classification.

3.2.2.1. Statistical techniques

Statistical features are the most easily extracted features used in EEG classification. Statistical features include mean, median, variance, standard deviation, skewness, and Kurtosis. These features are usually extracted in the frequency domain; however, they could also be extracted from the time domain. The period-amplitude analysis is extracted from the time domain and includes features like the wave duration, number of waves, and peak amplitude. Several types of entropy are also used as features, Shannon's entropy as the most common [25].

3.2.2.2. Common spatial patterns

Common Spatial Pattern (CSP) is a procedure that applies a linear transformation on the input data to map it into a more discriminative space. The transformation is done using spatial filters. The new time series after transformation have variances that are optimal for the discrimination of classes. The original algorithm that is used on two classes applies spatial filters to the data and then maximizes the variance of one while minimizing that of the other [26]. Therefore, the two classes get more discriminative features that could be fed to a machine learning model. New data points are transformed using the same spatial filters before classification. The formula of the spatial filters is given by:

$$w = argmax_w \frac{||wX_1||^2}{||wX_2||^2}$$

There are several ways of solving the maximization problem, the most common is simultaneous diagonalization of the covariance matrices which is also called the generalized eigenvalue decomposition. The algorithm is extended for multiclass problems by using a One-Vs-Rest (OVR) approach where the variance between each class and all other classes is maximized [27].

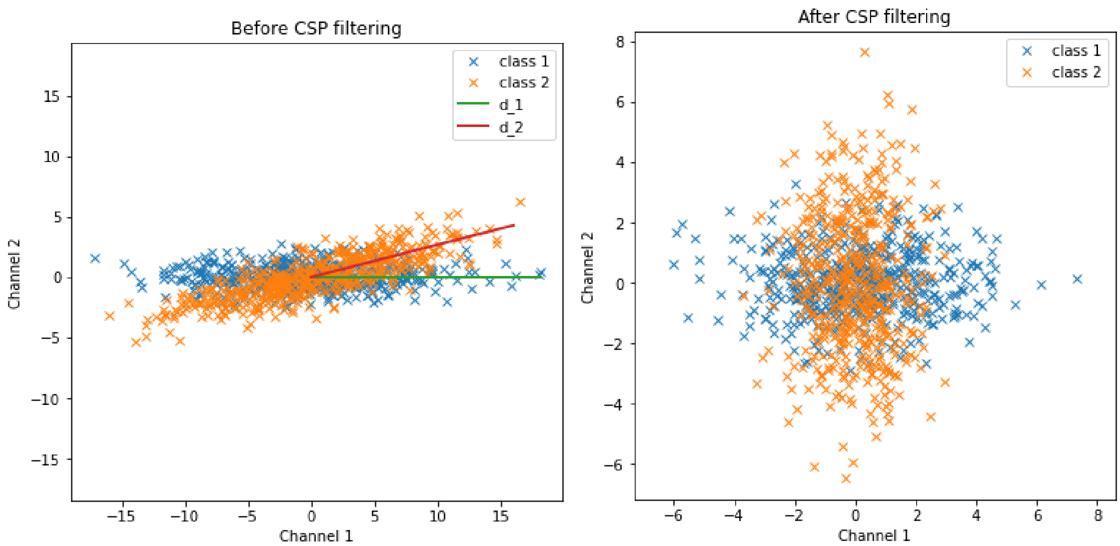


Figure 3.10: Effect of CSP on overlapping data

As shown in figure 3.10, after the transformation, the variance difference is maximized over each channel separately, so one class has high variance on channel 1 and low on channel 2 and the second class has the opposite.

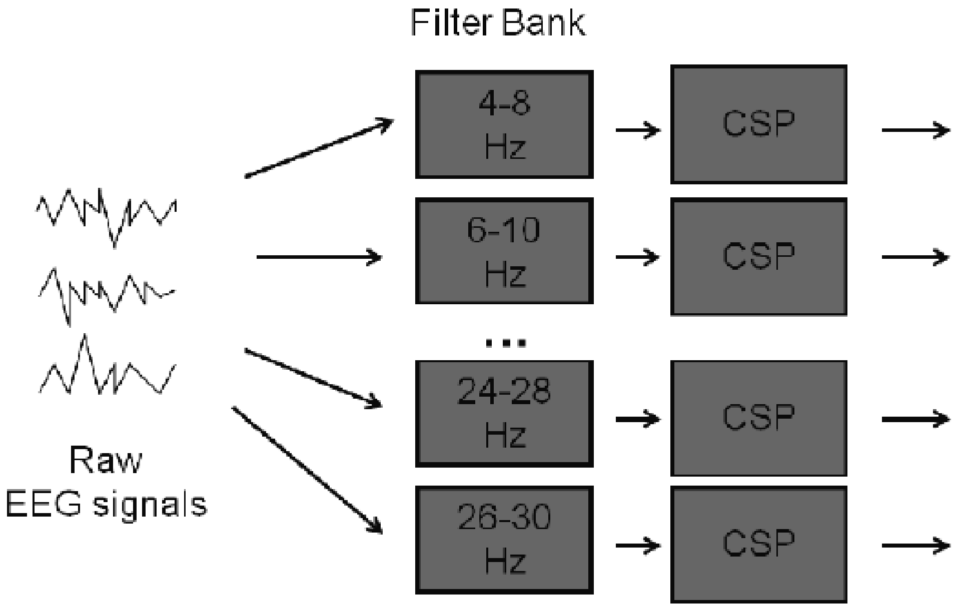


Figure 3.11: Filter Bank CSP

One variant of the CSP is the Filter Bank Common Spatial Pattern (FBCSP). The idea is to filter the signal in several frequency bands using a filter bank: hence the name. CSP is then applied to each of the resulting frequency bands separately [28]. This is usually followed by feature selection algorithms in order to select the best features that will be used for classification.

3.3. Comparative Study of Previous Work

In this section, we will give a comparison between techniques discussed in the previous sections.

3.3.1. Transforms

The DWT is the improvement of FFT and STFT. FFT is not suitable for non-stationary signals (like EEG) as it doesn't give information about the position of the frequencies. Nevertheless, it overcomes the fixed resolution problem of STFT by using different scale wavelets. It was also proven by experiment that DWT gives better resolution than STFT even for clinically interpretable data [29].

3.3.2. Feature Extraction

The following table shows the kappa value of the CSP and statistical features on all subjects. Last column shows the average of all subjects.

Table 3.1: comparison between statistical features and CSP on evaluation data

algorithm	S1	S2	S3	S4	S5	S6	S7	S8	S9	avg
Statistical	0.69	0.34	0.71	0.44	0.16	0.21	0.66	0.73	0.69	0.52
CSP	0.68	0.42	0.75	0.48	0.40	0.27	0.77	0.75	0.61	0.57

3.3.3. Classifiers

Table 3.2: Comparison between classifiers on Neurology Research Center EEG dataset [30]

Algorithm	Accuracy
SVM	95.6%
Naive Bayes	84.4%
KNN	86.1%

3.4. Implemented Approach

Firstly, we apply DWT using Daubechies 4 (db4) wavelet and apply multilevel decomposition of 7 levels. We chose the DWT as it is much better at approximating EEG signals frequency components as explained in the previous section. Then based on the comparative study, we use CSP on the approximations and details of the DWT and concatenate the result. Finally, we used the Support Vector Machine (SVM) classifier based on the comparative study.

Chapter 4: System Design and Architecture

This is in fact the most important chapter in the whole booklet. In this chapter, we shall be describing all of our main modules along with other modules that help accomplish our goal of enabling quadriplegic people to control any devices they want with their brains.

4.1. Overview and Assumptions

Our system is composed of the following main modules: the headset (the simulator), the machine learning model, the mobile phone, and the wheelchair. The different parts of our system are connected using a Wi-Fi router that enables our devices to exchange information.

From now on we will be referring to the headset as the simulator and we will also use them interchangeably and you should know that both refer to the same module. We will also describe the elegant way of connecting these devices that support adding even more devices to the network seamlessly to enable the quadriplegic person to control even more devices as needed. Not only the quadriplegic person will be able to control the wheelchair, but it can control even extra devices.

Since the price of the headset is very high, we have replaced it with a very elegant piece of software we call the simulator, that We shall describe in detail a little bit later. We have also replaced the electric wheelchair with a prototype 4-wheeled car since the electrical wheelchair costs a lot.

4.2. System Architecture

The system as shown in figure 4.1 is composed of the 3 main modules connected through a Wi-Fi router. These devices actually communicate through a simple rest Application Programming Interface (API). This rest API has 4 routes /F to indicate forward movement, /B for backward movement, /L for left movement, and /R for right movement.

In our system, the simulator or headset is monitoring the brain activity of the person. Whenever an activity is detected in the brain of the person. This activity is classified as the user intending to move in one of four directions forward, backward, left, or right.

Whenever a new action is detected by the headset it is sent to the mobile phone. The mobile phone receives the action and uses this action to move the virtual cursor on the mobile's screen. And if the mobile app is in the wheelchair control mode the mobile phone sends these actions to the wheelchair to move it in different directions. Depending on the selected mode the user actions are sent to the corresponding device by the mobile phone.

All of the communication among different devices of our network is done through Hypertext Transfer Protocol (HTTP) request as shown in figure 4.1.

4.2.1. Block Diagram

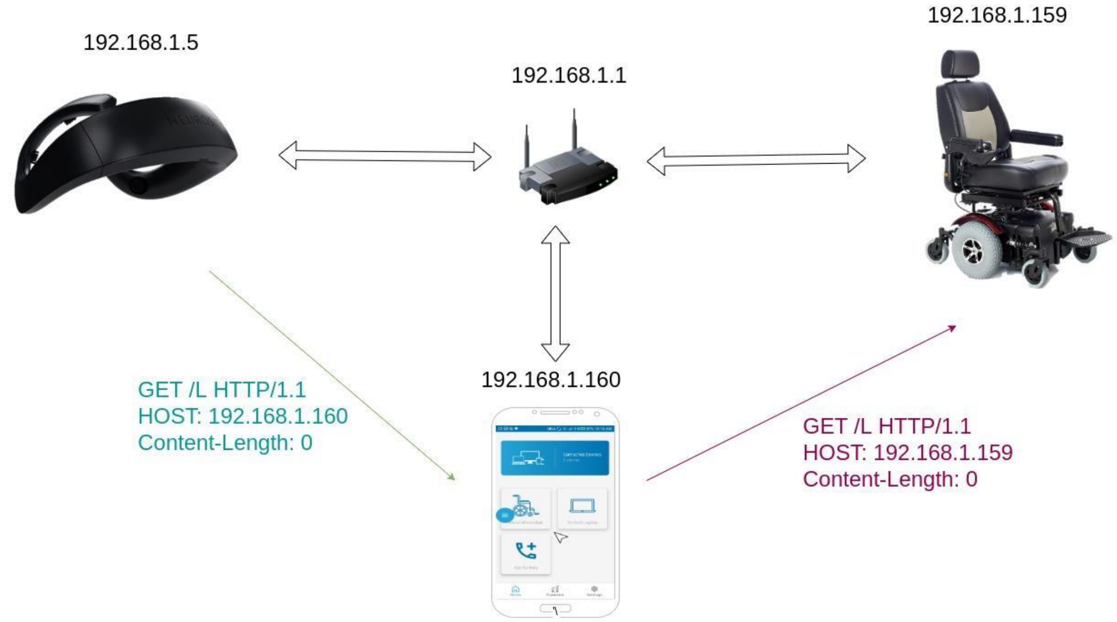


Figure 4.1: Block diagram of the Psionica System

4.3. Simulator (Headset) module

In this section we will be talking about the simulator program that is used to mimic the original headset. The original headset is named Neurosity crown this headset has built-in Wi-Fi and a quad-core processor and can be programmed. To mimic this headset, we developed a computer program that has 2 parts the frontend part that is developed using Hypertext Markup Language (HTML), Cascading Style

Sheets (CSS), JavaScript, and the backend part that is developed using flask framework.

● **Inputs:**

○ A dataset consists of many samples of one user and these samples contain the true action or sequence of actions the user wants to think of.

● **Outputs:**

○ the predicted action that the model thinks the user wants.

4.3.1. Functional Description

The simulator module is responsible for simulating the process of a user thinking of some action and then predicting what this action is based on EEG recorded data of him thinking of the action.

In the real world while the user will be wearing the headset. If the user wants to move to the left, he would be asked to imagine himself moving his right arm. If he wants to move left, he has to imagine himself moving his left arm, and in case of moving forward he has to imagine himself moving his tongue back and forward, and in case he wants to move backward he has to imagine himself moving his legs (see figure 4.2).

If the user isn't thinking of any of their previous imagination our machine learning solution can easily detect that. And this way no action will be taken.

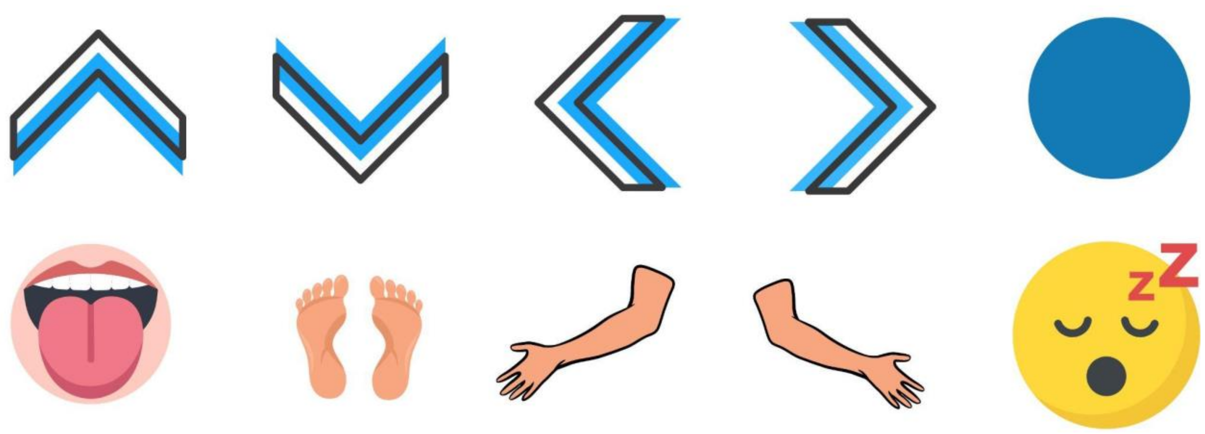


Figure 4.2. Organ Imagination and its corresponding action

To mimic the headset, we designed the simulator (shown in figure 4.3). The User Interface (UI) of the simulator is very simple and is composed of 4 parts.

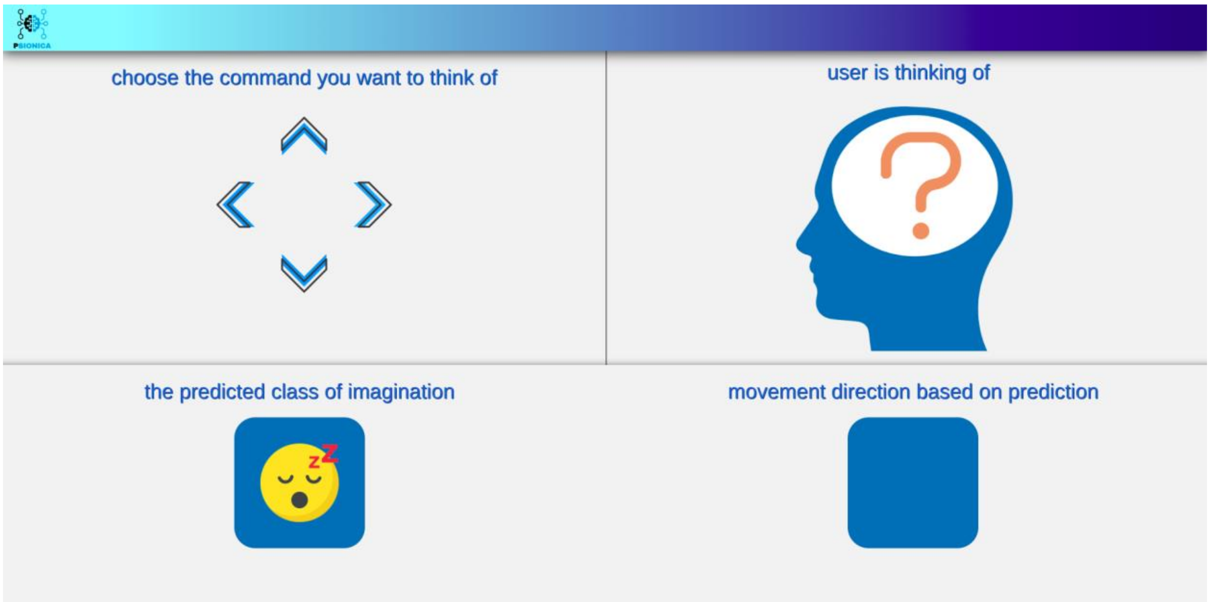


Figure 4.3: The simulator when the user is idle

The upper left part of the simulator's UI is composed of 4 buttons that are shaped like arrows; each arrow corresponds to the user intending to move in one of four directions. Let's say the user wants to move backward and thus clicks the back button or arrow.

The upper right part of the UI shows what the user is thinking of right now. And as we know since the user is intending to move backward, he has to imagine himself moving his legs, which is shown in the upper right part (see figure 4.4).

The bottom left part of the UI shows what the machine learning model has predicted. If the prediction was correct as shown in figure 4.4 then both the bottom left, and the upper left parts would be showing the same imagination. And if the model's prediction was incorrect as shown in figure 4.5, where the user is imagining that he is moving his leg, but the model predicted that he is imagining moving his tongue.

In the bottom right part, the predicted action that corresponds to the predicted imagination is shown. In figure 4.4 you can see the predicted action is the same as the intended one which is moving backward, since the model predicted our imagination correctly. But in the second case shown in figure 4.5, the model mispredicted our imagination and instead of moving backward the predicted action is to move forward.

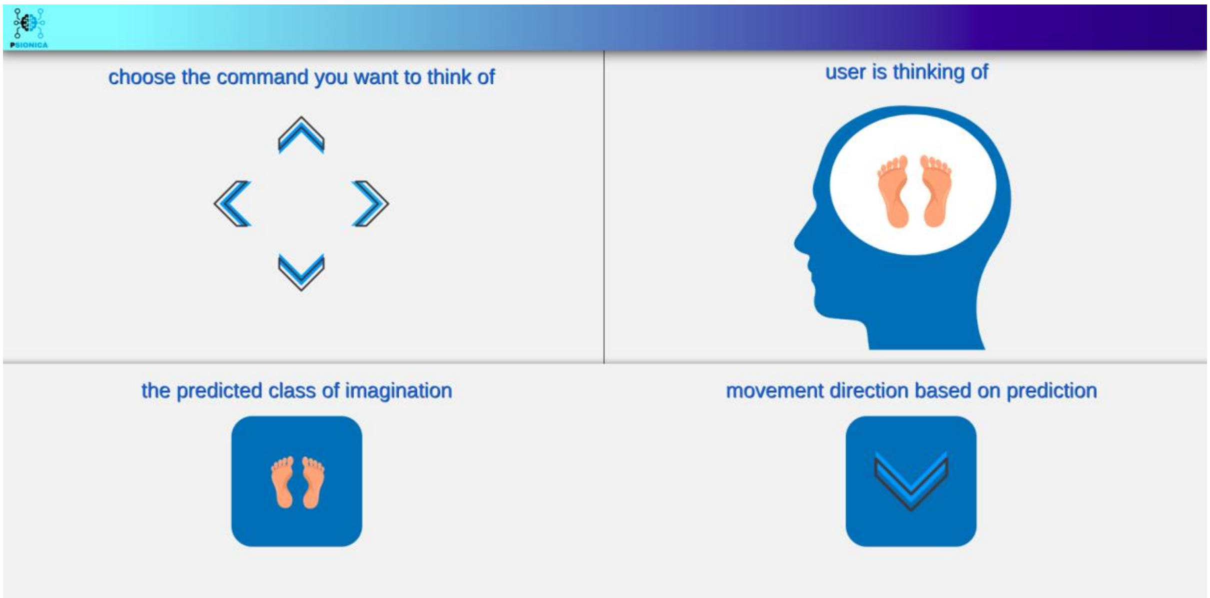


Figure 4.4: Simulator when user's imagination is classified correctly

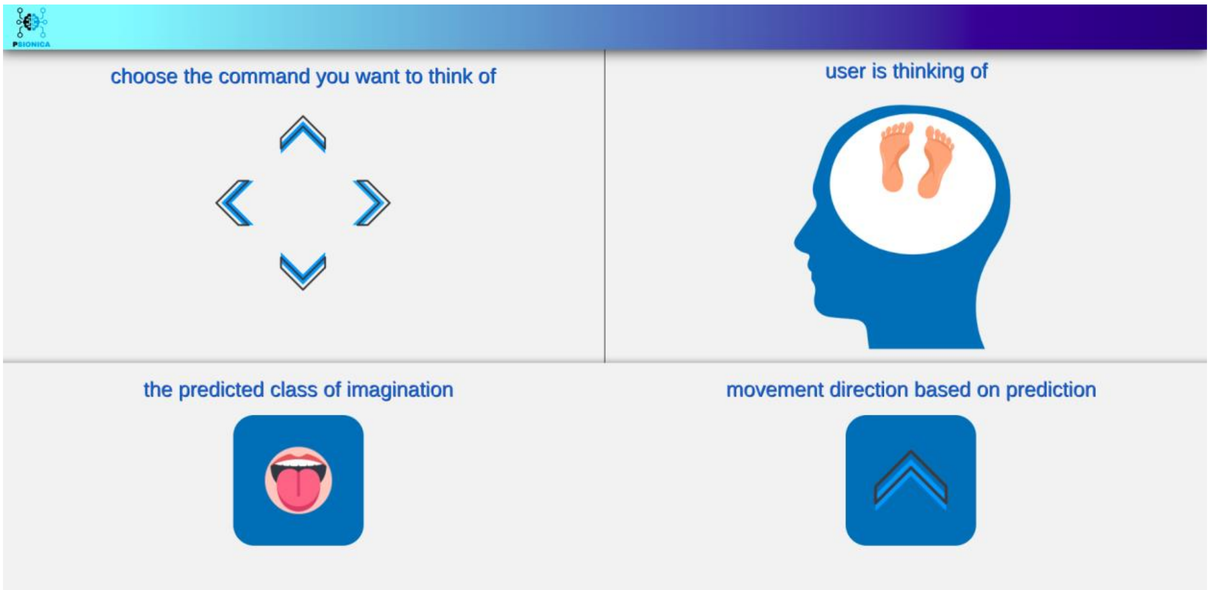


Figure 4.5: Simulator when user Imagination is misclassified

4.3.2. Modular Decomposition

The simulator is a computer program with a web UI. The simulator is divided into 2 parts: the frontend part which is written using HTML, CSS, and JavaScript, and the backend part which is written using flask which is a python framework for developing web applications.

The frontend part is responsible for the UI part and communicates with the backend through HTTP requests. The backend part is responsible for the core of the simulation and for passing test data to the model and getting the predicted actions then sending them to the frontend which in turn sends the actions to the mobile phone.

4.3.3. Design Constraints

Each sample of user thinking must be 3 seconds long and must be recorded using an EEG Headset with 25 EEG electrodes.

4.4. Signal Preprocessing module

In this section, we will discuss more information about the dataset and how we dealt with it since this is considered the first stage that the dataset passes by.

- **Input:**
    - Dataset of recorded brain signals for 9 users in a general data format
- **Output:**
    - Filtered dataset in a proper format where desired actions are separated

4.4.1. Functional Description

Signal preprocessing goal is to provide the model with well-formatted and filtered data. Fortunately, the data we acquired was exposed to some preprocessing procedures. The data was sampled at 250 Hz and a bandpass filter was applied to the data between 0.5 Hz and 100 Hz. So, all data was bounded within this range. In order to decrease the effect of noise, an additional 50 Hz notch filter was enabled to suppress the noise. Also, during the recording of the data from the subjects, the sensitivity of the amplifier was set to 100 µV. Then, data is recorded and saved in General Data Format (GDF). So, these were the needed preprocessing techniques to make data cleansing. We also checked the data by applying some preprocessing like a bandpass filter, but it gave us the same results which made us confident about the data cleansing. On the other hand, the data was recorded in a format that needs to be changed. We managed to read the data and change its format from GDF into NumPy format (npy). This procedure is done once then we always deal with the npy files.

4.4.2. Modular Decomposition

We have two models in our system. The first one is to classify the data whether the user is thinking of action, or he is idle. While the second model is to classify data into one of the four actions in case the user was thinking. So, to make the data prepared for these two models we have divided the data in a reasonable way.

We grouped the four action samples together and the idle samples together. Idle samples are the data where the user wasn't thinking of action. He was just keeping his eyes open (looking at a fixation cross on screen), closed, or just moving his eyes.

We also divided the idle data into samples equal in size to action samples where the size is 3 seconds equal to 750 Hz. Therefore, all samples became equal in size, well-formatted, and ready for passing through the following phase.

4.4.3. Design Constraints

The data should be amplified with suitable sensitivity. There are no constraints on the maximum or minimum frequencies of the signals as it will be subjective to a bandpass filter that will ignore any unneeded frequency bands.

4.5. Machine Learning Model module

In this section, we will discuss the module of the machine learning model. It is considered the core module in our project or the brain of our project. In this module, the signals are analyzed, and the decisions are taken.

- **Input:**
    - Preprocessed data
- **Output:**
    - Classified actions

4.5.1. Functional Description

The goal of the model is to classify the input data correctly with high accuracy as much as possible. We have two models each with three stages. The first stage is loading the preprocessed data and splitting it into training data and testing data with percentages of 80% and 20% respectively.

The second stage is feature extraction. In this stage, we finally decided to use 2 techniques that yield better efficiency after trying many different techniques. We tried statistical techniques and feature selection techniques like the Principal Component Analysis (PCA). We also tried Sequential Backward Floating Selection (SBFS) which is a technique used to select k optimal features from n extracted features. We have mentioned some important techniques in detail in chapter 3 so we will not go deeply into details. Finally, we decided to apply DWT and CSP. In this way, we extracted the needed features, and the second stage has come to an end.

The third stage was making standardization for the extracted features and choosing the classifier. We chose different classifiers such as K Nearest Neighbors (KNN), Support Vector Machine (SVM), and many others until we realized that the SVM is the most suitable one that got the best accuracy depending on our extracted features.

4.5.2. Modular Decomposition

As we mentioned in the previous point, we are using these techniques for both two models. The first model is a binary classifier that classifies the data if idle or thinking. The second model is a multi-class classifier that classifies the four actions in the case of thinking.

4.6. Mobile Application module

The Mobile Application is the main interface of our system. It’s responsible for connecting the headset and the model with all other devices to give the user the capability to control each device he/she wants.

- **Input:**
    - The classified action from the model
- **Output:**
    - Applying action to the mouse cursor or sending it to the selected device

4.6.1. Functional Description

The mobile application offers a UI for interfacing with the user. The application consists of some connected devices that the user can communicate with to control them. These devices can be extended to any number of devices that the user needs to handle his essential daily tasks easily. For our system, this list contains controlling his wheelchair, calling his doctor in case of emergency and we added controlling his laptop mouse for future work. This list can be shown in figure 4.6.

The application assumes that the user is connected to the network that contains the list of devices that it will be communicating with. It opens a socket connection on port 12345 to receive the classified actions that are sent by the headset and starts to apply these actions according to the current mode.

When the user chooses a device to connect with. The application changes the mode of the application and sends the device IP and opens the port to another component of the application to start sending the incoming actions. This process can be shown in figure 4.7.

The application also provides a statistics page that will be responsible for showing the user's health condition and some statistics about him like heart rate and other observations.

The last UI page is the settings. This page contains the system preferences that the user can edit like the doctor's phone number and the connected devices' IPs. Figure 4.8 shows the settings page.

In addition to communicating with the connected devices, the application also provides the functionality of controlling the mobile phone itself by providing a mouse cursor and an action menu that are independent of the application itself.

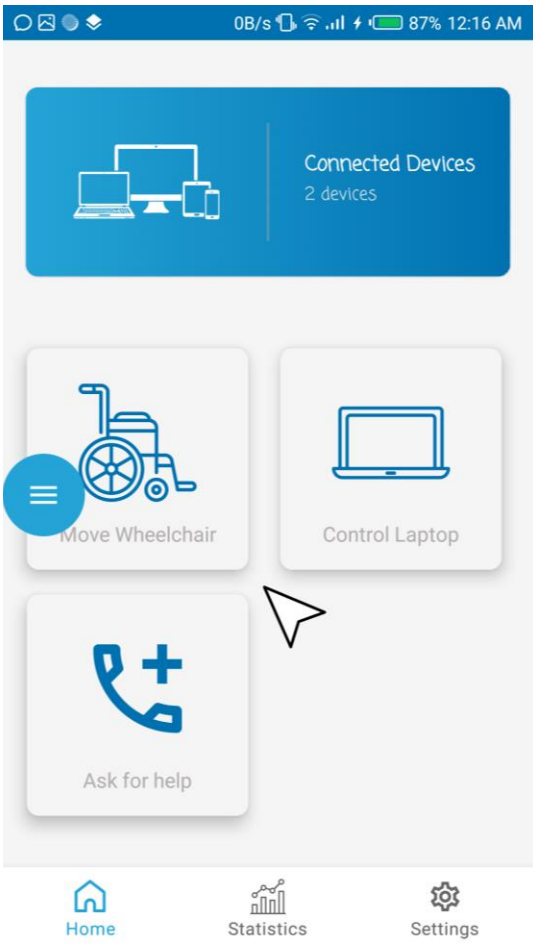


Figure 4.6: devices list

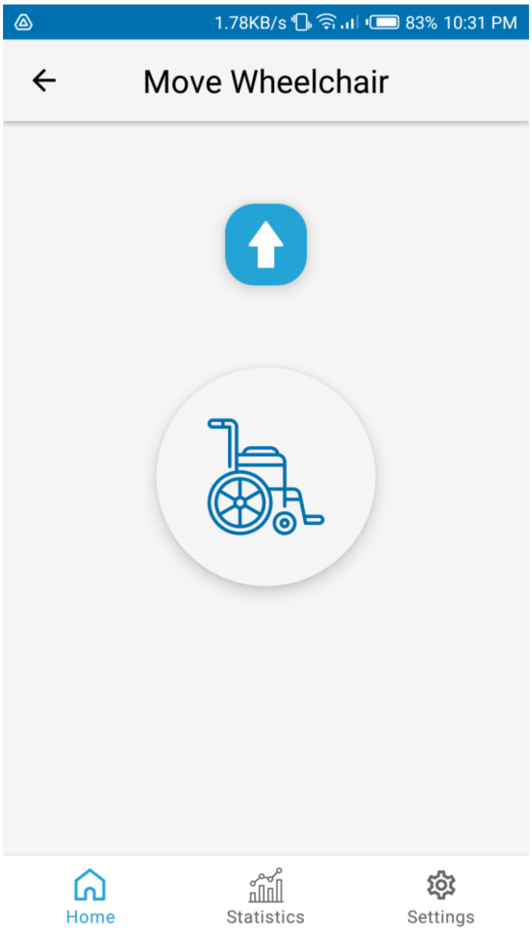


Figure 4.7: sending incoming actions to the wheelchair

All this is done thanks to the Accessibility services API [31] provided by the Android framework that is designed to help users with disabilities use their android devices and apps. It is a long-running service that helps users process information on the screen and lets them interact with the device.

As the number of actions is limited and there are only 4 actions due to the trade-off between the accuracy and the model, the application needs to provide an easy way to do all the functionalities of the mobile phone with these 4 actions only.

The application uses the 4 actions to move the mouse cursor in the 4 main directions. By default, when the user moves the cursor to a new position and stops, the application clicks automatically after some time. In this way, the application handles the click events easily.

The other events are handled by an action menu added to the mobile screen. When the user clicks on it, it opens buttons for other actions like pressing the back button, home button, and Recent apps buttons.

If the user doesn’t want the automatic click, another action is added to go to idle mode. This mode stops the clicks except on the actions menu. If the actions menu is clicked, the user exits the idle mode and returns to automatic click again. Figure 4.9 shows the actions menu when it’s opened.

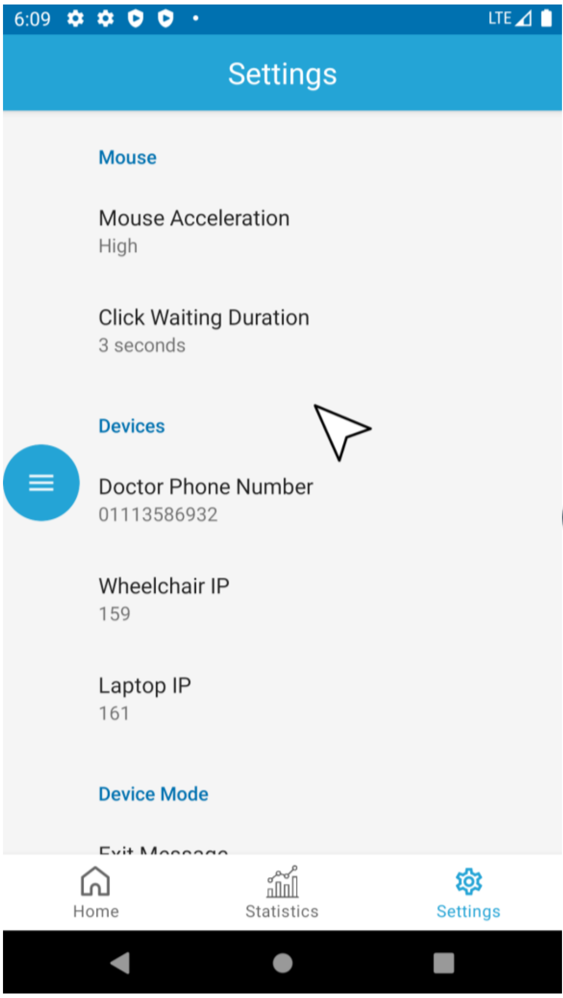


Figure 4.8: settings page

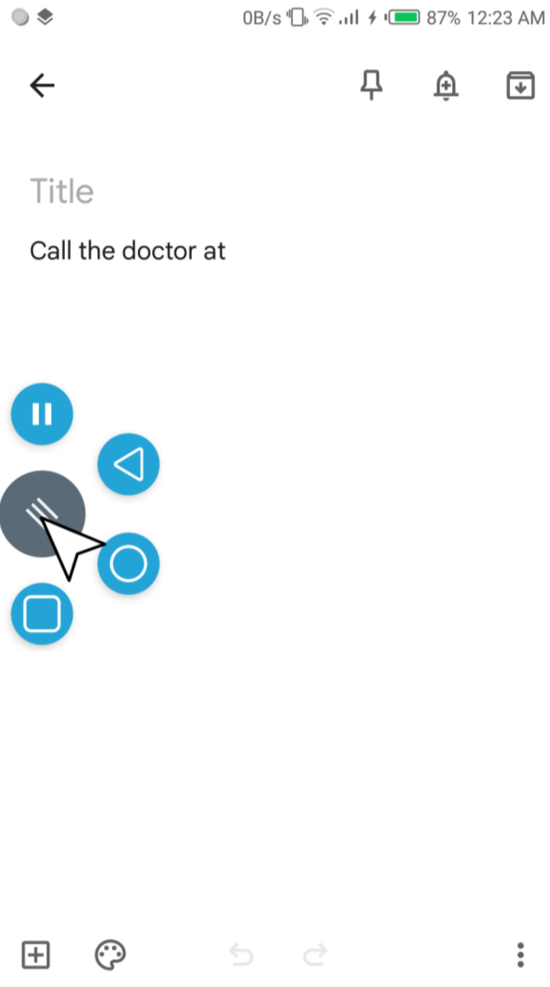


Figure 4.9: actions menu

4.6.2. Modular Decomposition

The mobile application comes with two main components each one is responsible for some of the mentioned functions:

- **Application UI:**
  - Enables the user to choose between different devices to connect with.
  - Communicates with the accessibility service to provide it with the data of the device and update the UI according to the received action.
  - It can also monitor the user's health condition.

- **Accessibility Service:**

- The accessibility service is a separate component. It can run even if the application is not opened.
- Draws the mouse cursor and the actions menu that the user can use inside any application on the mobile.
- Receives the classified action from the headset and the model through the network Wi-Fi.
- Applies action based on the current mode that the user selected.
- If the user selects any device, It sends the received action to that device through the network Wi-Fi.
- If the user didn’t select any device. It enables the user to use the mobile phone by moving the cursor and applying accessibility actions like clicking on the screen and pressing the back button and other actions.

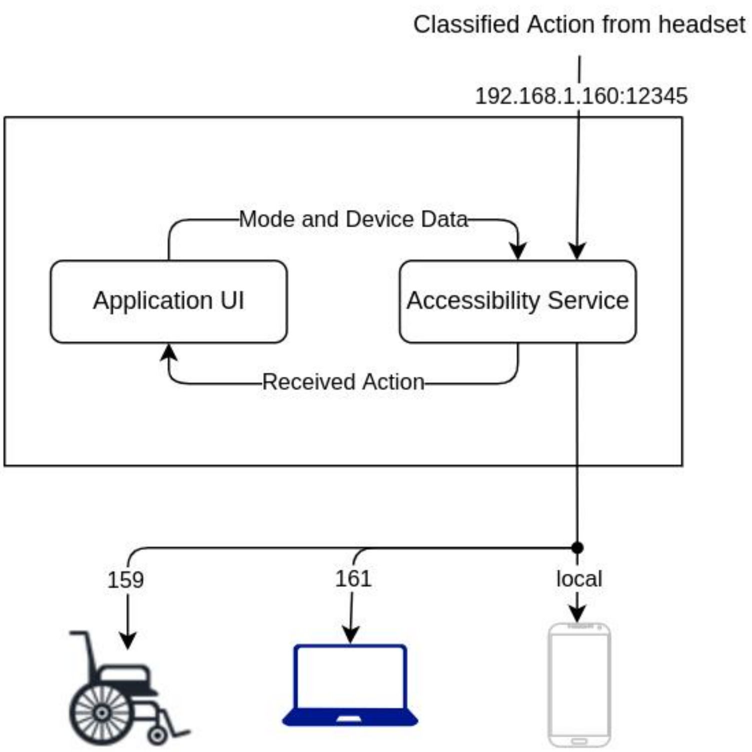


Figure 4.10: modular decomposition of the mobile application module

4.6.3. Design Constraints

- All devices should be connected to the same network.
- The Mobile is connected using local IP that ends with 160.

- The user should Allow some permission to be able to use the Application like Turning on Psionica Accessibility service and allowing to draw overlay and other permissions.

4.7. Wheelchair module

In this section we will be talking about the wheelchair module or as It should be called the car module and how it is able to achieve the required movement functionality.

- **Inputs:**
    - an HTTP request on the following URL 192.168.1.159:12345 on one of the following routes /F, /B, /L, or /R to drive the car in the forward, backward, left, or right direction.
- **Outputs:**
    - the output is the wheelchair moving in the requested direction forward, Backward, Left, or Right
    - the car shouldn't move in the backward direction if there is anything blocking it behind.

4.7.1. Functional Description

The wheelchair (shown in figure 4.11) will move in the requested direction but with the exception that it won't move backward if there is anything behind. We detect that there is anything behind the wheelchair by using an ultrasonic sensor to detect if an obstacle is present behind the wheelchair.

The wheelchair is programmed to occupy a fixed IP address which is 192.168.1.159 and it has a web server listening on port 12345 to any incoming HTTP requests to drive the car in the intended direction.

The car listens to incoming HTTP requests on the following routes /F, /B, /L, or /R to drive the car in the forward, backward, left, or right direction as requested by the user.

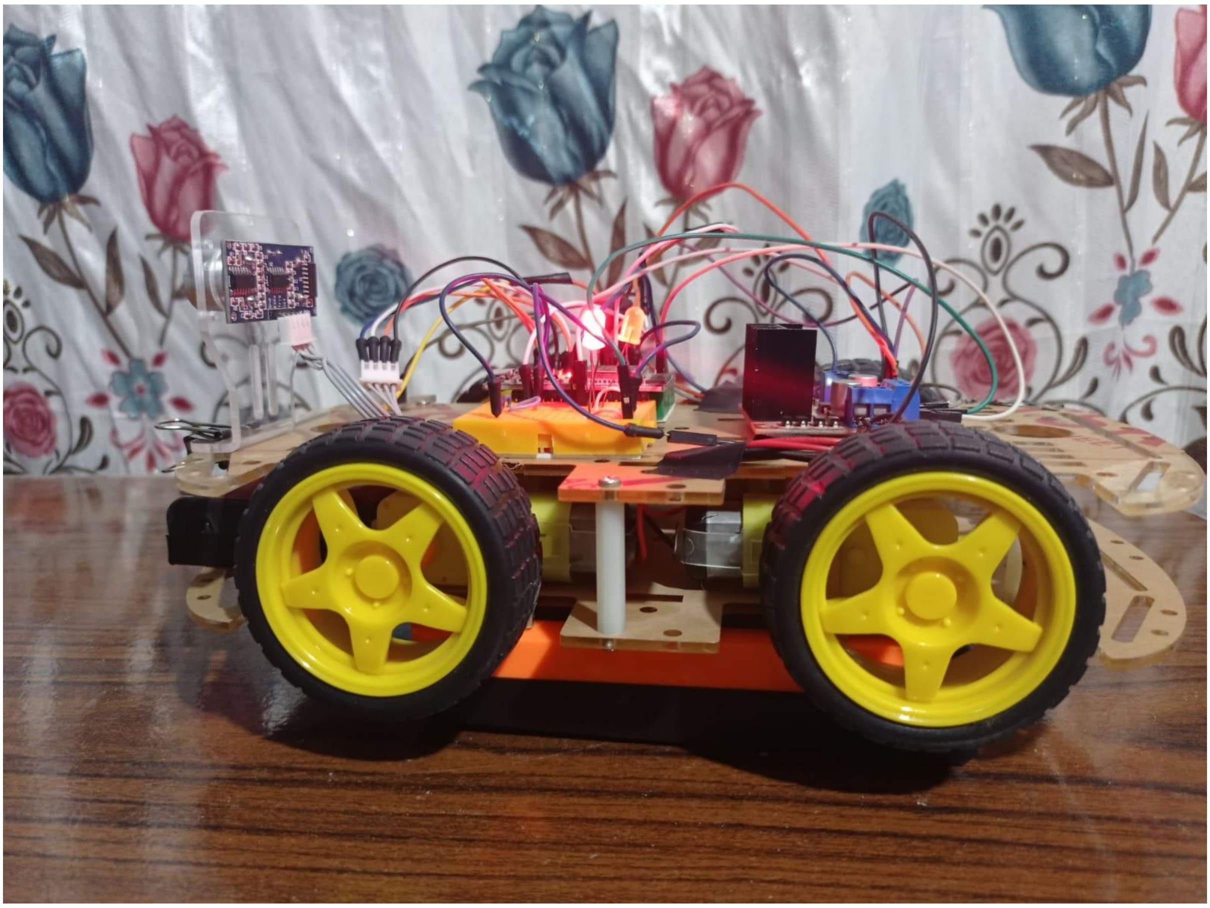


Figure 4.11: the wheelchair as a 4WD prototype car

4.7.2. Modular Decomposition

This module has hardware components that are controlled by a very robust highly mature software that adheres to that uses object-oriented programming concepts to abstract each component of the hardware.

The hardware components of the wheelchair (shown in figure 4.12) are as follows:

- **ESP32 Controller with onboard Wi-Fi:** this is the brain of the wheelchair that hosts the software logic of the wheelchair.
- **Four-wheeled car:** with each wheel attached to a different 12v dc motor.
- **L298N motor driver:** a chip that enables esp32 to control 12v dc motors.
- **ultrasonic:** this ultrasonic is used to avoid the user crashing when moving backward.

- **12v chargeable lithium battery:** this is the battery that powers all the components on the car.

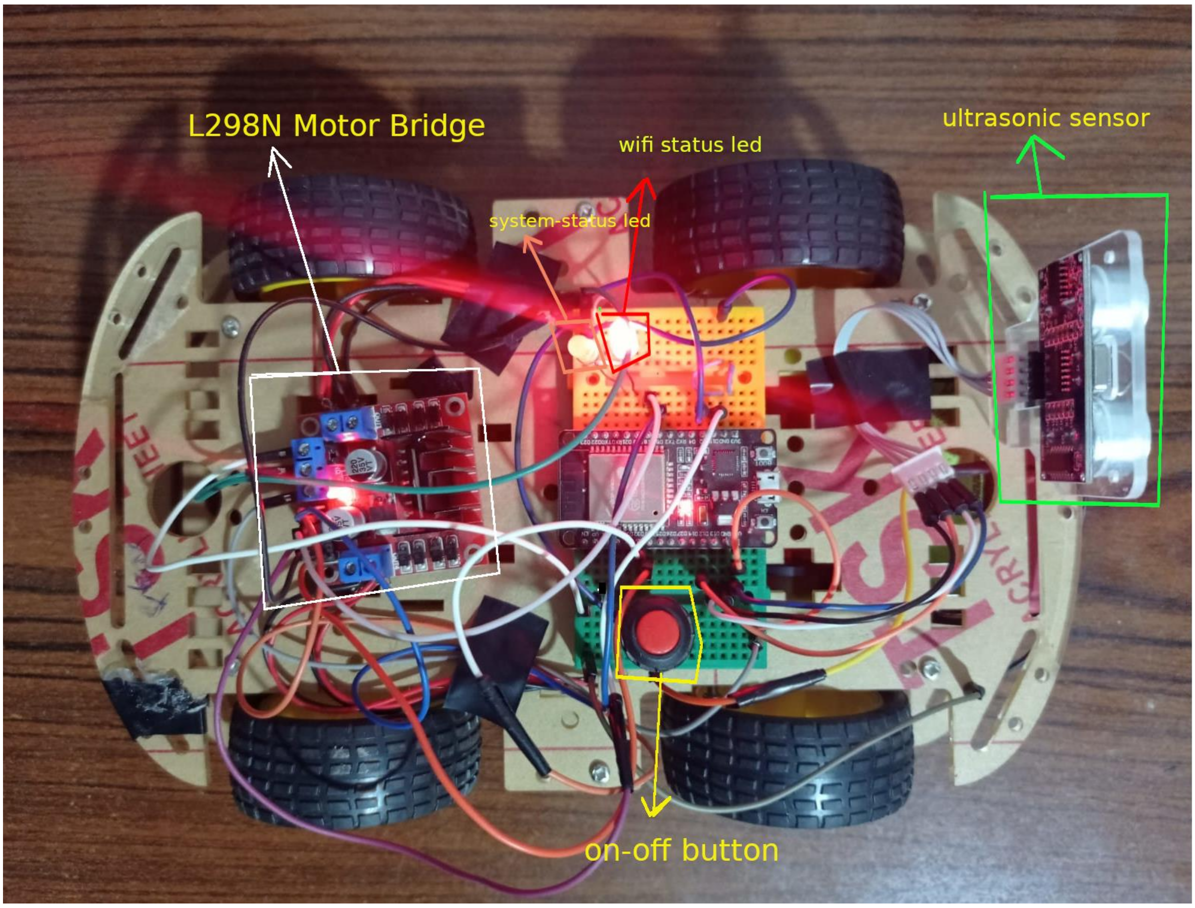


Figure 4.12: hardware components of the prototype wheelchair

The software components of the wheelchair (as shown in figure 4.13) are:

- **station**
    - The station is the main component that integrates all of the other modules together
    - The station integrates the web server, the Wi-Fi functionality, the car, and the status LED together.
    - The station has the Wi-Fi functionality that connects the car to the Wi-Fi network and reconnects it in case of router restarts or crashes.
- **Web Server**

- The station also has the web server listening for HTTP requests on port 12345 to requests to move the car
- If an HTTP request arrives on the route /F to the web server, the car moves forward
- And if an HTTP request arrives on the route /B it moves backward
- And the same goes for /L and /R that make the car move left and right respectively.

- **LEDs**
  - This is a simple class that is used to turn and off the status LEDs for the car
  - The car has 2 status LEDs
  - The red led is used to indicate that the car is connected to the Wi-Fi successfully.
  - The red led goes off if the Wi-Fi is disconnected and goes on again when the car reconnects successfully to the Wi-Fi.
  - The orange led blinks all the time to indicate that the microcontroller is up and running.
- **Car**
  - The car class is responsible for operating the 4 motors of the car and the ultrasonic sensor
- **Motor**
  - The motor class is responsible for operating the motors of the car.
- **Sonar**
  - The sonar class is used to control the back ultrasonic of the car
  - The sonar sends an echo sound that when colliding with an object returns.
  - Based on the time the echo took to come back, the distance to the object in centimeters is calculated using the following equation

$$distance(cm) = \frac{duration * SOUND\_SPEED}{2}$$

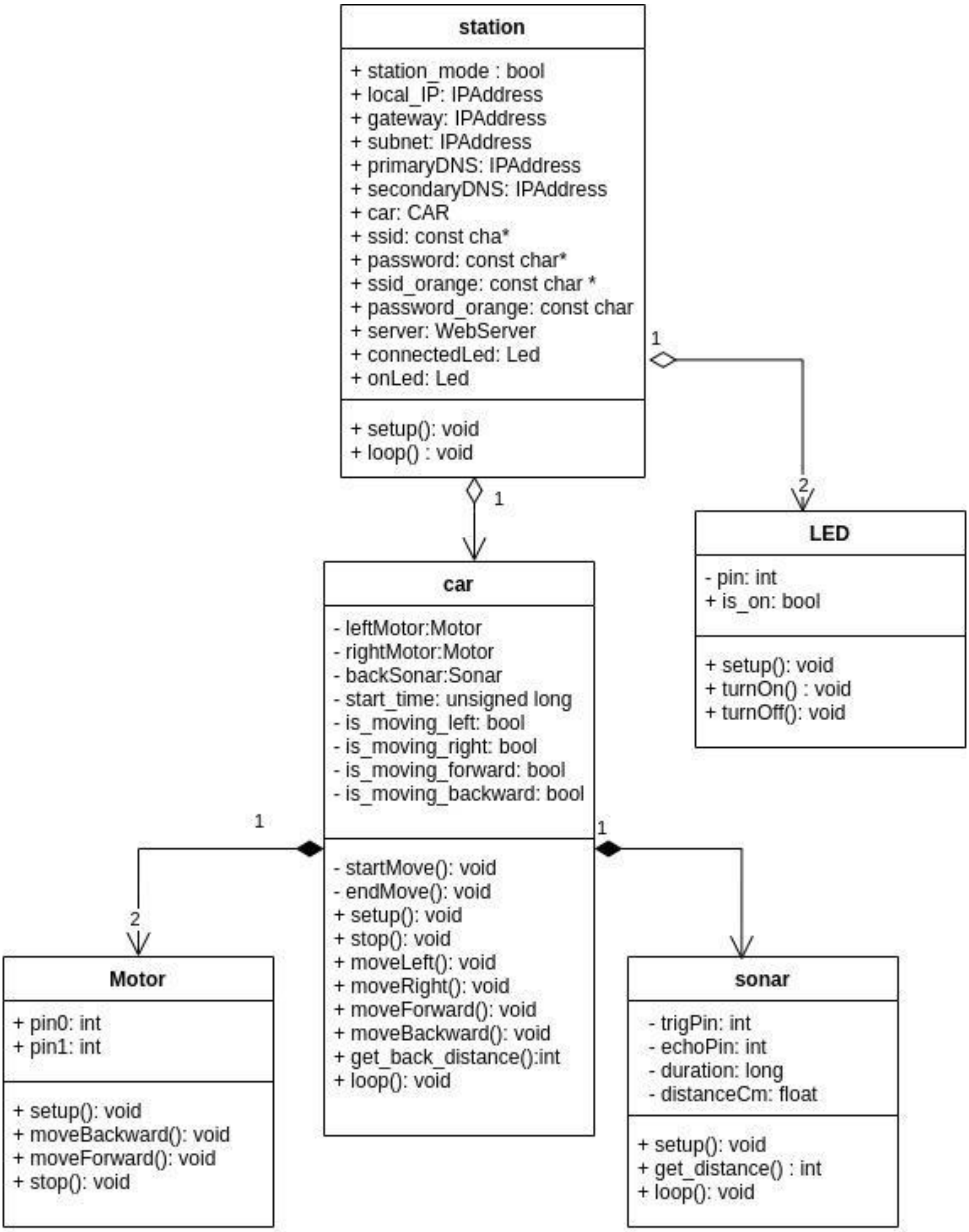


Figure 4.13: UML class diagram of the wheelchair

4.7.3. Design Constraints

The wheelchair controller must be connected to a fast Wi-Fi router to be able to respond in real-time.

Also, no other device on the same Wi-Fi can occupy a fixed IP as the one occupied by the car.

4.8. Integration module

One of the most critical aspects of our system is to be able to integrate all of the involved components. Not only could we integrate the modules of our system, but also, we could develop a system that allows the addition of new devices to the network seamlessly and enable the mobile phone and thus the quadriplegic person to control these devices as well.

4.8.1. Functional Description

The components of our system can communicate together through a local Wi-Fi network. This happens by making all the devices join the same Wi-Fi router.

In our system, devices communicate with each other using HTTP requests to send the actions. The headset sends the actions to the mobile phone after detecting the action using the machine learning model, then once the mobile phone receives the action it decides which device should the action go to. The action received could be used to control the virtual cursor of the mobile phone, or it could be used to control the user's wheelchair. It could also be used to control any other device on the network given it is listening to the action sent by the mobile phone.

In our network devices, the need to receive actions like the mobile phone and wheelchair must assume a fixed IP and must listen on port 12345. The simulator or headset doesn't have to assume a fixed IP (see figure 4.1).

4.8.2. Design Constraints

Some devices must assume a fixed IP like the mobile phone and the wheelchair. Also since we are using raw HTTP the security of our system is somehow limited.

Chapter 5: System Testing and Verification

Since our project is supposed to be used by quadriplegic people, we had to test it thoroughly to make sure everything works smoothly and avoid any bugs or defects.

5.1. Testing Setup

Each module has its own setup since they are completely different. The setup for each module is as follows:

- Simulator: the models and testing data files should be added to the simulator folder to start the testing.
- Models: For testing this module, the model and test data should be loaded successfully.
- Wheelchair: the Wi-Fi router should be on and the wheelchair's red led should be on.

5.2. Testing Plan and Strategy

Each of the modules was tested independently to make sure it works correctly as desired. Then, we started to incrementally integrate the different parts of the product.

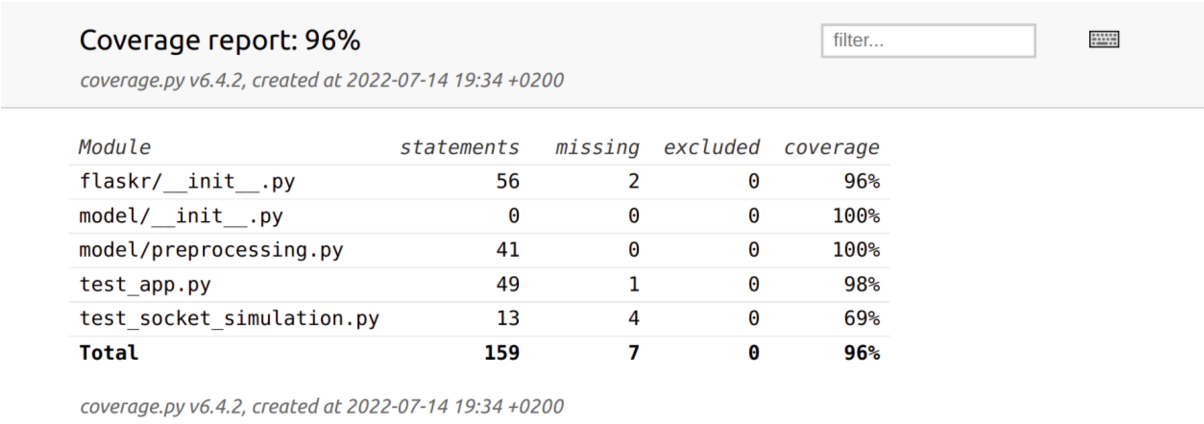
5.2.1. Module Testing

5.2.1.1. Testing The Simulator

To test the simulator (shown in figure 4.3) we did both manual testing, that is we started the app and started testing that the UI behaves as expected and that indeed the predicted actions are successfully sent to the mobile app.

To make sure the simulator functions as expected we developed scenarios to test the different behaviors and to test the edge cases.

Also, we developed unit testing scripts to test the backend part of the app that is written in flask and we could achieve coverage of 96% as shown in figure 5.1.



Coverage report: 96%

filter...

coverage.py v6.4.2, created at 2022-07-14 19:34 +0200

Module	statements	missing	excluded	coverage
flaskr/__init__.py	56	2	0	96%
model/__init__.py	0	0	0	100%
model/preprocessing.py	41	0	0	100%
test_app.py	49	1	0	98%
test_socket_simulation.py	13	4	0	69%
Total	**159**	**7**	**0**	**96%**

coverage.py v6.4.2, created at 2022-07-14 19:34 +0200

Figure 5.1: Coverage report of the simulator's backend part

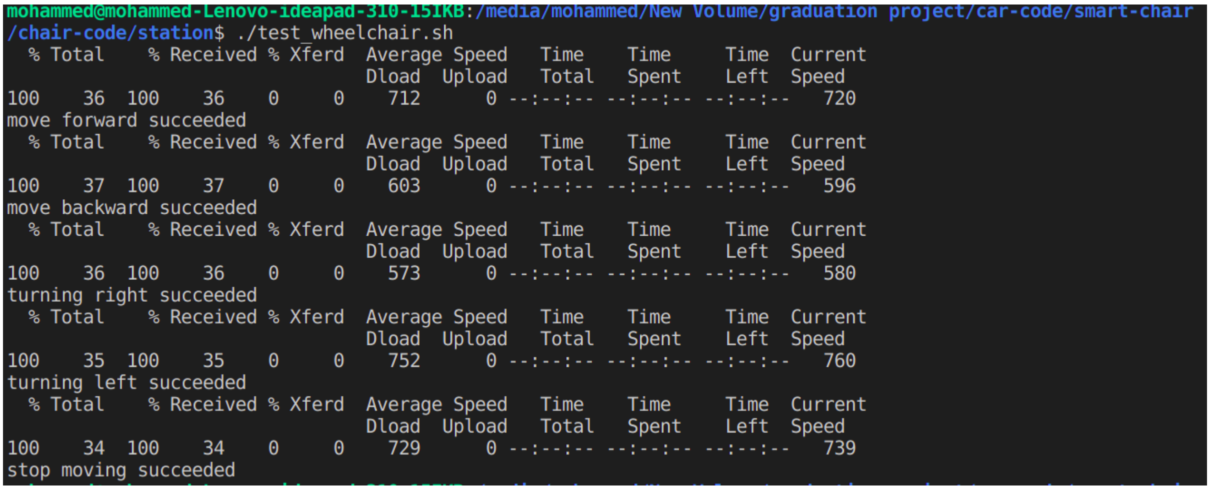
5.2.1.2. Testing The Model

The machine learning model consists of two individual models each with its own data. The datasets were split at the beginning of the development into 80% training data and 20% test data. We tested each model on its test data and also calculated the 10-fold cross-validation.

5.2.1.3. Testing the wheelchair

To test the wheelchair (shown in figure 5.2) we did some stress testing to see how fast it will reply to send actions by sending a lot of commands at once and we could see that it behaved as expected.

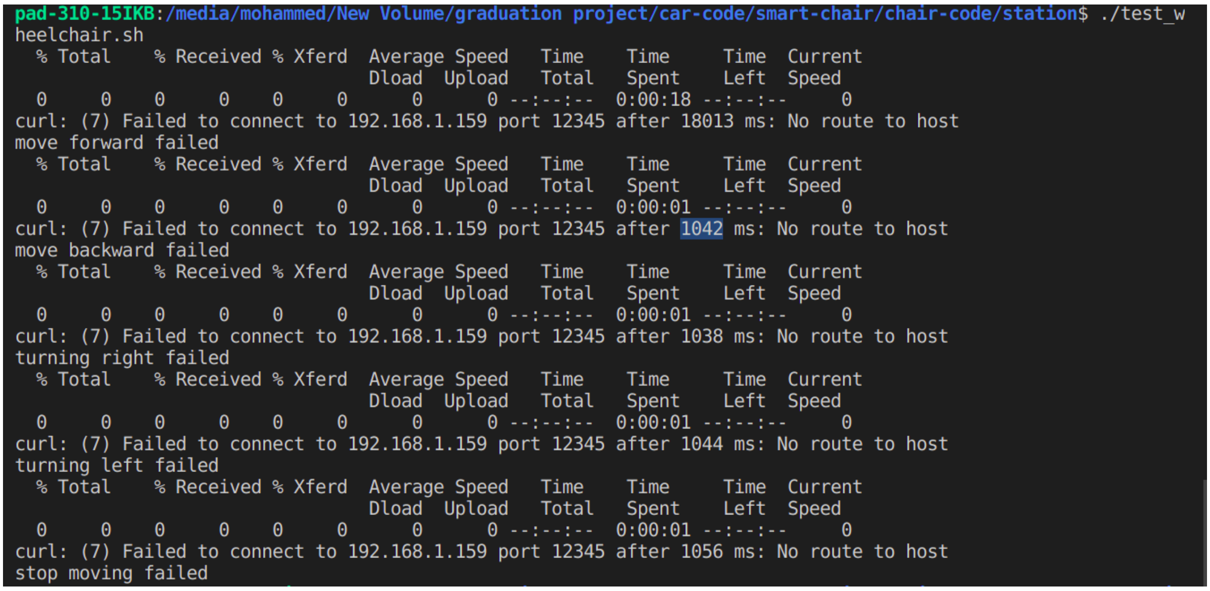
We also developed a bash script that tests that the wheelchair replies to the sent actions and requests. This bash script uses curl Linux command to send HTTP requests to the car and parses the HTTP responses from the car to make sure that the wheelchair received the request and that it moved in the right direction. Shown in the figure 5.2 is a successful run of the test script.



```
mohammed@mohammed-Lenovo-ideapad-310-15IKB:/media/mohammed/New Volume/graduation project/car-code/smart-chair
/chair-code/station$ ./test_wheelchair.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    36  100    36    0     0    712      0 --:--:-- --:--:-- --:--:--   720
move forward succeeded
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    37  100    37    0     0    603      0 --:--:-- --:--:-- --:--:--   596
move backward succeeded
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    36  100    36    0     0    573      0 --:--:-- --:--:-- --:--:--   580
turning right succeeded
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    35  100    35    0     0    752      0 --:--:-- --:--:-- --:--:--   760
turning left succeeded
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    34  100    34    0     0    729      0 --:--:-- --:--:-- --:--:--   739
stop moving succeeded
```

Figure 5.2: successful run of the wheelchair testing

In the figure 5.3 we show a screenshot of an unsuccessful run of the bash script as the car was turned off to test the script behavior.



```
pad-310-15IKB:/media/mohammed/New Volume/graduation project/car-code/smart-chair/chair-code/station$ ./test_w
heelchair.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:18 --:--:--     0
curl: (7) Failed to connect to 192.168.1.159 port 12345 after 18013 ms: No route to host
move forward failed
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
curl: (7) Failed to connect to 192.168.1.159 port 12345 after 1042 ms: No route to host
move backward failed
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
curl: (7) Failed to connect to 192.168.1.159 port 12345 after 1038 ms: No route to host
turning right failed
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
curl: (7) Failed to connect to 192.168.1.159 port 12345 after 1044 ms: No route to host
turning left failed
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
curl: (7) Failed to connect to 192.168.1.159 port 12345 after 1056 ms: No route to host
stop moving failed
```

Figure 5.3: unsuccessful run of the wheelchair testing

5.2.2. Integration Testing

The integration testing strategy is illustrated in Figure 5.4. We started by integrating the wheelchair with the mobile to make sure the communication is working correctly. Then, we integrated the simulator to test the process of sending an action to the application without the machine learning model. After making sure that everything works correctly, we integrated the machine learning model and the EEG data and carried out the final integration testing.

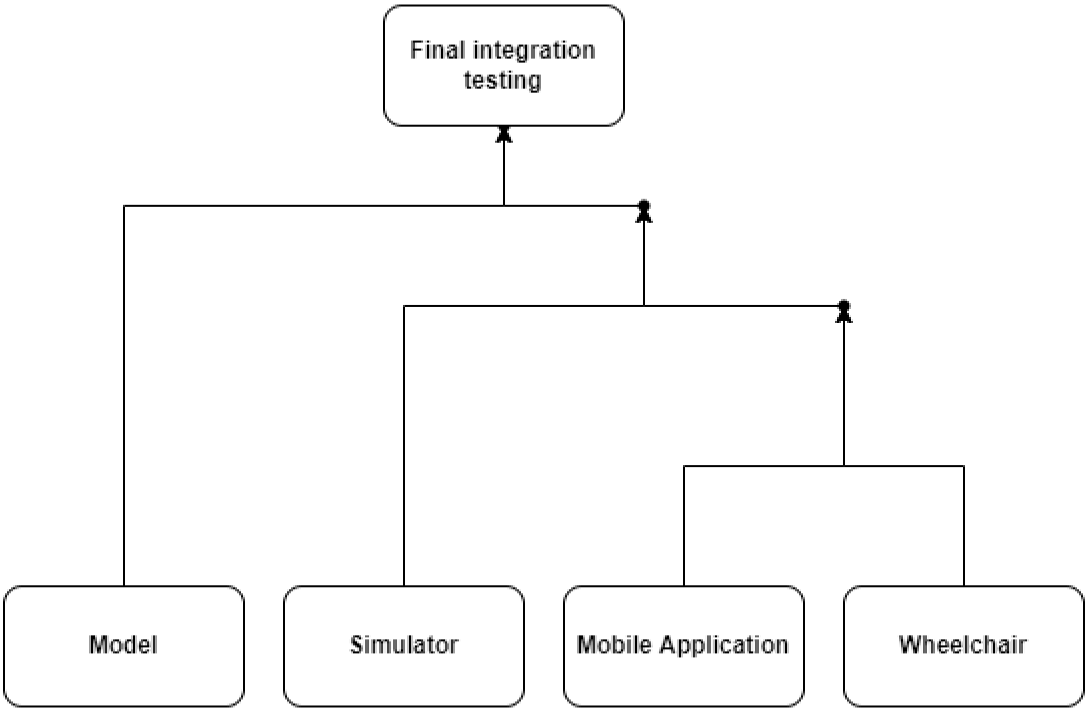


Figure 5.4: Integration testing plan

5.3. Comparative Study to Previous Work

Table 5.1: 10-fold cross-validation comparison on our data

algorithm	S1	S2	S3	S4	S5	S6	S7	S8	S9	avg
CSP	0.64	0.42	0.79	0.36	0.21	0.28	0.62	0.77	0.71	0.48
Our Model	0.62	0.53	0.62	0.42	0.46	0.32	0.86	0.77	0.56	0.517

Chapter 6: Conclusion and Future Work

In previous chapters, we have shown that there are millions of disabled people who can't live their lives independently and need help almost all the time. Then we introduced our easy-to-use system that helps these people to control their electronic devices like a wheelchair, mobile phone, and laptop with less effort using just their thinking and imagination only and thoroughly explained different components of it how they work together.

In this chapter, we will take a look at how our journey was. We will describe the challenges we faced throughout working on the project, and the skills and experiences we learned. Then we will give our conclusions and summarize what we achieved. Finally, we will talk about how we can improve our project in the future.

6.1. Faced Challenges

The Brain-Computer Interface and Recording electrical activity from the brain using EEG technologies and interpreting it to control machines are new and challenging topics in the last few years. Working on a system that relies on these technologies introduced a variety of challenges. In this section, we will discuss these challenges and show how we overcome these challenges in our system.

6.1.1. Fundraising

Since our project consists of many components in order to get the full experience. Some are necessary like the EEG headset and mobile phone while others are subsidiaries such as wheelchair and laptop. We assumed that the user that would buy our project would already have a smartphone. So, the headset and a wheelchair are still needed, and they are expensive components that cost more than 30,000 EGP. This pushed us to apply for funds. So, we sent our project proposal to many fundraisers. Although we were accepted by Itida and we were provided with 10,000 EGP, it was not enough to buy even one of the two components. Regrettably, we weren’t able to make use of the fund.

6.1.2. EEG headset availability

One of the most challenges we faced during working on this project was finding the EEG headset. Firstly, we started to search for factories in Egypt that manufacture headsets, but we found that it's neither manufactured in Egypt nor available during this period due to coronavirus which reduced the productivity rate of the factories and they partially stopped exporting. These days after many countries reduced their precaution measures, factories started to export the headset again, but they noted that the manufacturing process plus the delivery would take about 4 months. At this point, we completely agreed to work on plan B which needs more effort to add a new module, simulator, that simulates the headset and uses a recorded dataset. This manages us to overcome the lack of the headset.

6.1.3. Number of supported actions

It's known in the field of analyzing brain signals that as the number of actions increases the accuracy greatly decreases. So, this was a challenge to get more than two actions and keep the accuracy acceptable. We began to work on gradually increasing the number of actions with keeping eye on the accuracy until we reached four actions with very satisfactory accuracy. Then we added some innovative ideas that helped us to go beyond this number and reached more actions like click, back, go to the homepage and get currently opened pages.

6.1.4. Real-time system

Since we didn't get the EEG headset for the reasons stated before, we decided to make a simulator with a very informative and splendid UI that will give the whole experience as if it was real-time, however, we are working offline owing to the recorded dataset.

6.1.5. Motor imagery paradigm accuracy

Motor imagery is a paradigm that yields the least accuracy as it depends totally on imagination without any movement or any means which is explained deeply in chapter 3. Despite it being challenging to get high accuracy using motor imagery, we insisted on using this paradigm to make the user more comfortable and doesn't have to make any physical moves. Also, there are approaches that mix different paradigms together to increase the number of actions but it's a trade-off between getting more actions and the ease of use and comfortability of the user.

6.1.6. Lack of generalization

Everyone has slightly different brain signals when he intends to do something. So, there is no identical pattern all people do to express their desire for doing something. This fact obliged researchers and developers who are concerned with neuroscience and analyzing brain signals to keep in mind this point. We found in papers that we have to train our model per subject and don't depend on gathering all subjects' data and train the model once on this aggregated data. Of course, this makes the model lack generalization, but this is essential in this field in order to acquire a better accuracy.

6.2. Gained Experience

Throughout this project we were exposed to different technologies, tools, and aspects from which we gained great experience that can be summarized in the following:

- We gained the experience of researching and reading many papers about a certain problem.
- We applied our past skills and knowledge in AI to build a model from scratch.
- We gained experience working on a project containing hardware and software components together.
- We learned how to develop a mobile app.
- We gained the experience of integrating all projects' parts together.
- We gained experience working on a large-scale project for a long time while keeping consistency and motivation to get splendid results.
- We learned much information about signal processing and transformation and put this information into work.

6.3. Conclusions

In this section we will illustrate the effect of this project, after describing its specs and functions in detail, on the life of the quadriplegic and disabled.

Psionica from now on will be their helpful friend that they will never dispense with. They will be able to:

- Depend on themselves in their motion by controlling their wheelchair.
- Enjoy using their smartphones and surfing the internet by controlling their mobile phones through our app.
- Call their doctor immediately in case of an emergency.
- Use their laptop by controlling its mouse just by their brain signals.
- Get insights about their health.
- Customize some app functions depending on their preferences.

6.4. Future Work

One of the most important strength points of this project is the extensibility. We could attach any new device to our network, and it will be controlled through the mobile app. So, this strength point opens the door for large future work that could be done such as:

- Controlling laptop
- Adding more models and devices helping in getting insights from the user
- Adding experimental sessions in the application in order to collect data from user to train on it to increase models' accuracies
- Increasing features at the app like making users memorize some locations and making the app guide the wheelchair towards these locations using Path Planning.
- Increasing the security of the network connecting different devices together

References

[1] DriveResearch, “What is Market Survey Research? | Defining the Benefits and Process of a Market Survey”, 2022, https://www.questionpro.com/blog/market-survey/, last accessed: 6/7/2022.

[2] Mahmoud Dahmani, Muhammad E. H. Chowdhury, Amith Khandakar, Tawsifur Rahman, Khaled Al-Jayyousi, Abdalla Hefny, Serkan Kiranyaz, “An Intelligent and Low-Cost Eye-Tracking System for Motorized Wheelchair Control”, sensors, vol. 20, no. 14, pp. 3936-3963, 2020.

[3] Sim Kok Swee, Desmond Teck Kiang Kho, Lim Zheng You, “EEG Controlled Wheelchair”, MATEC Web of Conferences, vol. 51, 02011, 2016

[4] Kurtzke, J F. “Epidemiology of spinal cord injury.” Neurologia, neurocirugia, psiquiatria vol. 18,2-3 Suppl: pp. 157-91. 1977.

[5] Willett, F.R., Avansino, D.T., Hochberg, L.R. et al. “High-performance brain-to-text communication via handwriting”. Nature, vol. 593, no. 7858, pp. 249–254, 2021.

[6] Niedermeyer E.; da Silva F.L. “Electroencephalography: Basic Principles, Clinical Applications, and Related Fields”, Lippincott Williams & Wilkins, fifth edition, 2004.

[7] Klem, G H et al. “The ten-twenty electrode system of the International Federation. The International Federation of Clinical Neurophysiology.” Electroencephalography and clinical neurophysiology. Supplement vol. 52, pp. 3-6, 1999.

[8] Picton, T W. “The P300 wave of the human event-related potential.” Journal of clinical neurophysiology: official publication of the American Electroencephalographic Society, vol. 9, no. 4, pp. 456-79, 1992.

[9] Sur, Shravani, and V K Sinha. “Event-related potential: An overview.” Industrial psychiatry journal, vol. 18, no. 1, pp. 70-3, 2009.

[10] Cuntai Guan, M. Thulasidas and Jiankang Wu, "High performance P300 speller for brain-computer interface," IEEE International Workshop on Biomedical Circuits and Systems 2004., pp. S3/5/INV-S3/13, 2004.

[11] Beverina, Fabrizio & Palmas, Giorgio & Silvoni, Stefano & Piccione, Francesco & Giove, Silvio. “User adaptive BCIs: SSVEP and P300 based interfaces”. PsychNology Journal, vol 1, pp. 331-354. 2003.

[12] Norcia AM, Appelbaum LG, Ales JM, Cottereau BR, Rossion B. “The steady-state visual evoked potential in vision research: A review”, Journal of Vision, vol. 15, no. 6:4, pp. 1-46. 2015.

[13] Singla, Rajesh. "SSVEP-Based BCIs". Evolving BCI Therapy - Engaging Brain State Dynamics, edited by Denis Larrivee, IntechOpen, 2018.

[14] Mulder, Th. “Motor imagery and action observation: cognitive tools for rehabilitation.” Journal of neural transmission, vol. 114, no. 10, pp. 1265-78, 2007.

[15] Zhu, Yanlong & Wang, Zhongyu & Dai, Chenglong & Pi, Dechang, Artifact Removal Methods in Motor Imagery of EEG, pp. 287-294, 2017.

[16] S. R. Sreeja, J. Rabha, K. Y. Nagarjuna, D. Samanta, P. Mitra and M. Sarma, "Motor Imagery EEG Signal Processing and Classification Using Machine Learning Approach," 2017 International Conference on New Trends in Computing Sciences (ICTCS), pp. 61-66, 2017.

[17] Park, Sangin et al. “Improving Motor Imagery-Based Brain-Computer Interface Performance Based on Sensory Stimulation Training: An Approach Focused on Poorly Performing Users.” Frontiers in neuroscience vol. 15, no. 732545, 2021.

[18] Stoica, Petre & Moses, Randolph, “Spectral Analysis of Signals”, Prentice Hall, 2005.

[19] Prandoni, Paolo, and Martin Vetterli, “Signal processing for communications”, CRC Press, 2008.

[20] Strang, G., “Wavelets. American Scientist”, vol. 82, no.3, pp. 250–255, 1994.

[21] Sejdić E.; Djurović I.; Jiang J., "Time-frequency feature representation using energy concentration: An overview of recent advances". Digital Signal Processing, vol. 19, no. 1, pp. 153–183, 2009.

[22] Addison, P.S., “Wavelet transforms and the ecg: A review”, Physiological Measurement, vol. 26, no. 5, pp. R155, 2005.
[23] Madhur, S. “Improving signal resolution and reducing experiment time in electron spring resonance spectroscopy via data processing methods.”, Ph.D. thesis, Graduate School of Cornell University, 2018.
[24] Mallat, S. G. “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,” IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 11, Issue 7, pp. 674–693, 1989.
[25] Stancin I, Cifrek M, Jovic A., “A Review of EEG Signal Features and their Application in Driver Drowsiness Detection Systems”, Sensors, vol. 21, no. 11, pp. 3786, 2021.
[26] G. Pfurtscheller, C. Guger and H. Ramoser, "EEG-based brain-computer interface using subject-specific spatial filters", Engineering applications of bio-inspired artificial neural networks, Lecture Notes in Computer Science, Vol. 1607, no. 1999, pp. 248-254, 2006.
[27] Darvish ghanbar K, Yousefi Rezaii T, Farzamnia A, Saad I, “Correlation-based common spatial pattern (CCSP): A novel extension of CSP for classification of motor imagery signal”, PLOS ONE, vol. 16, no. 3, pp. e0248511, 2021.
[28] Lotte, Fabien., “A Tutorial on EEG Signal Processing Techniques for Mental State Recognition in Brain-Computer Interfaces”, springer, 2014.
[29] Kiymik, M. & Guler, Inan & Dizibuyuk, Alper & Akin, Mehmet, “Comparison of STFT and wavelet transform methods in determining epileptic seizure activity in EEG signals for real-time application.”, Computers in biology and medicine, vol. 35, pp. 603-16, 2005.
[30] Shoka, Athar & Dessouky, Mohamed & el-sherbeny, Ahmed & El-Sayed, Ayman. (2019).
Literature Review on EEG Preprocessing, Feature Extraction, and Classifications Techniques.
Menoufia Journal of Electronic Engineering Research. 28. 292-299. 10.21608/mjeer.2019.64927.
[31] Google, “AccessibilityService | Android Developers”, 2022,
https://developer.android.com/reference/android/accessibilityservice/AccessibilityService , last accessed: 2022-07-13

Appendix A: Development Platforms and Tools

Psionica helps Quadriplegic patients to communicate with people and control electronic devices by reading and analyzing brain signals. This is done using many tools and technologies. In this section, we will discuss the used tools and programming languages for both Hardware and Software platforms.

A.1. Hardware Platforms

For the electric wheelchair, we represented it as a robotic car. The car uses ESP32 Controller with onboard Wi-Fi to control four wheels attached with a 12v dc motor and a L298N motor driver. The car is powered by a 12v chargeable lithium battery.

For the mobile, the application works on Android 7.0 or higher with a minimum SDK version equal to 24. The application tested on different mobile phones; these are the specs of one of them:

- Operating system: Android 8.1
- CPU: Octa-core 1.3 GHz Cortex-A53
- RAM: 3 GB

A.2. Software Platforms

A.2.1. Programming languages

- C++: we used C++ for programming the esp32 controller of the wheelchair.
- Python: we used it for developing the machine learning model and for the backend part of the simulator
- JavaScript: we used JavaScript for developing the frontend part of the simulator.
- Kotlin: used for android application development

A.2.2. Libraries and Frameworks

- Esp32 Wi-Fi communication library: we used it for managing Wi-Fi connection with the router.

- ArduinoJson Library: used for serializing the JSON requests and responses to and from clients on the esp32.
- Arduino Library: used for controlling the hardware of the esp32.
- Flask: used flask to develop the backend part of the simulator.
- Python unittest: we used it to develop unit test scripts to test the simulator's backend part.
- Python coverage: we used it to get the coverage report for the backend part unit testing.
- Sklearn: machine learning library used for feature extraction and classification
- Scipy: signal processing library used for noise removal and feature extraction
- MNE: we used it for loading and interpreting EEG recorded data.
- OS: we used this library for creating and managing model and datasets files and folders.

A.2.3. Tools and Platforms

- Esp32: we used esp32 as a controller for the wheelchair.
- GitHub: version control platform to share code among developers.
- Arduino Studio: used for programming the robotic car.
- Android Studio: a software tool designed specifically for Android development.
- Linux bash scripting: we used it for writing automation scripts to automate the wheelchair testing.
- VScode: we used it for developing the simulator's code.
- ClickUp: a platform used for project planning and task management.

A.2.4. markup languages

- Html: we used it for defining the structure of the simulator's frontend.
- CSS: we used it for styling the frontend of the simulator's frontend.

Appendix B: Use Cases

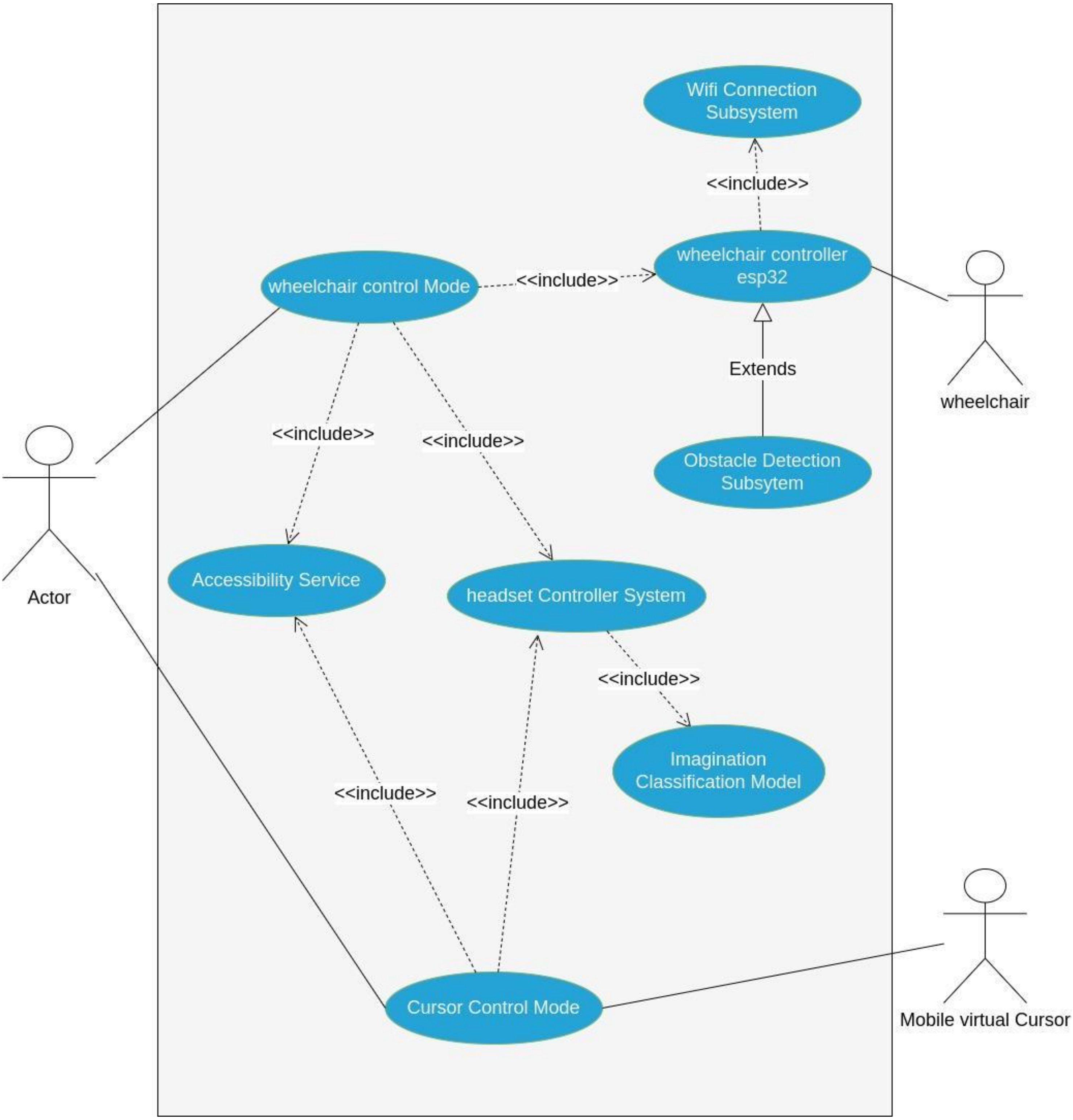


Figure B.1: Psionica Use Case Diagram

Appendix C: User Guide

We deliberately designed a simple and easy-to-use mobile application that will give, in our opinion, the best user experience. In this section, we will take a tour of our application and show the different screens that will appear during usage.

Since our application allows the user to make a call to a number of his choice that he can change it whenever he needs. User needs to allow the app to make and manage calls as shown in figure C.1.

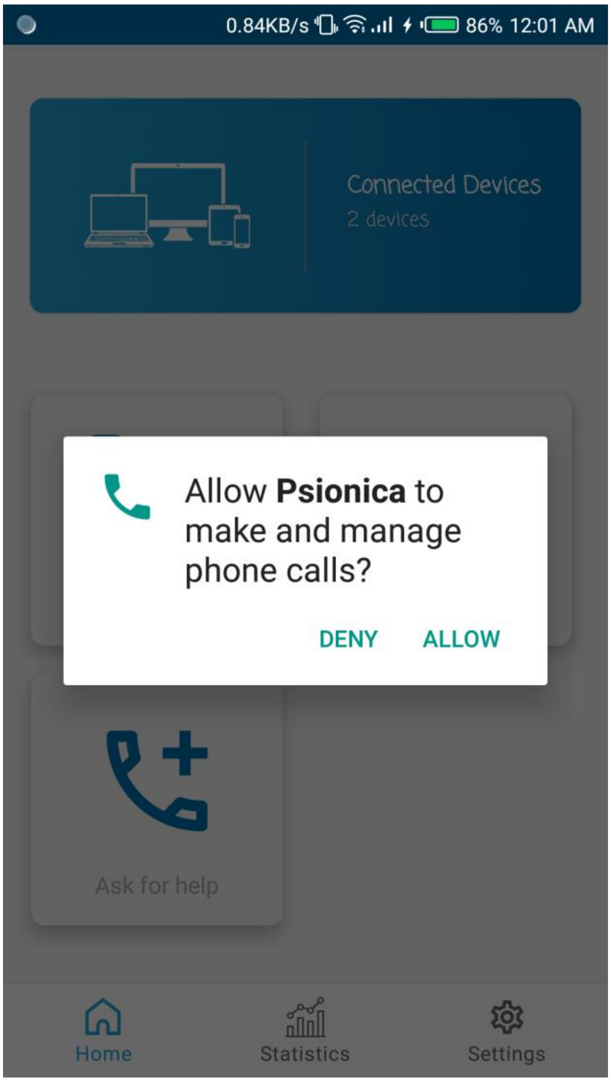


Figure C.1: Allow making and managing calls

Also, the user has to connect to the Wi-Fi and use the specified static local IP address specified in the pop-up message as shown in figure C.2.

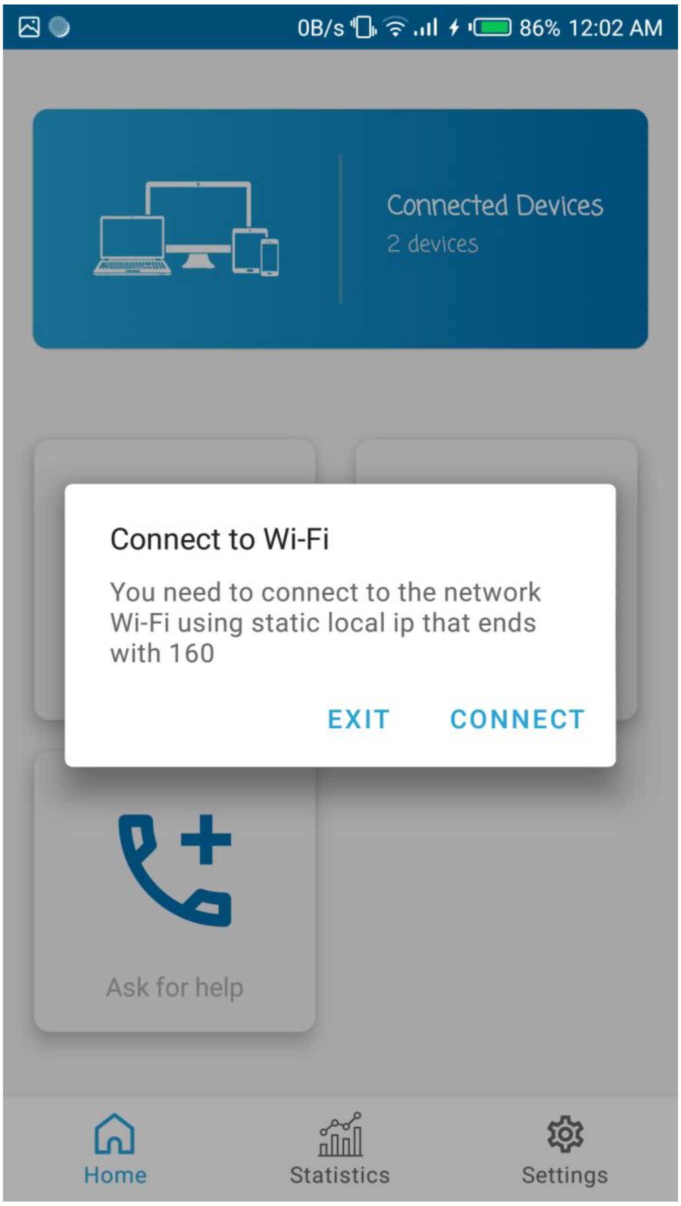


Figure C.2: Connect to Wi-Fi using static IP

You need to allow the overlay as well to be able to use the application and get the benefit of the elegant design as shown in figure C.3.

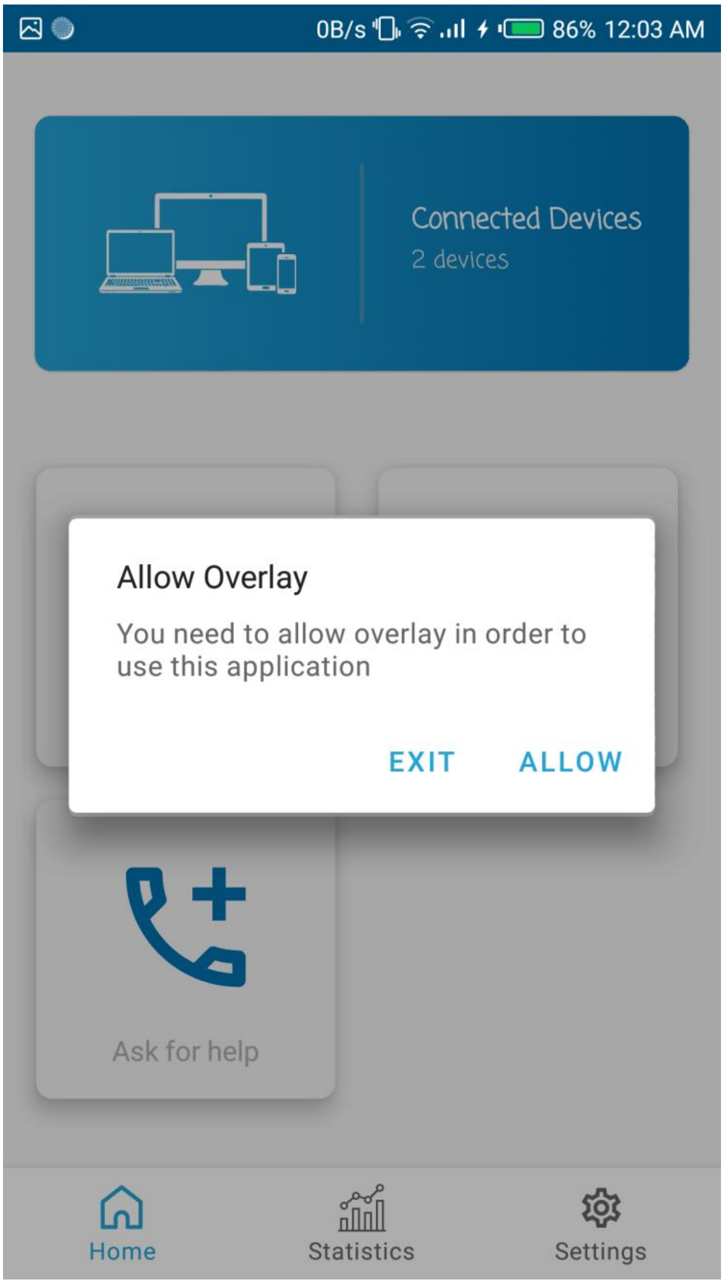


Figure C.3: Allow the overlay

As we introduced that our application uses a mouse pointer on the screen. So, this needs the user to turn on the accessibility service as shown in figure C.4.

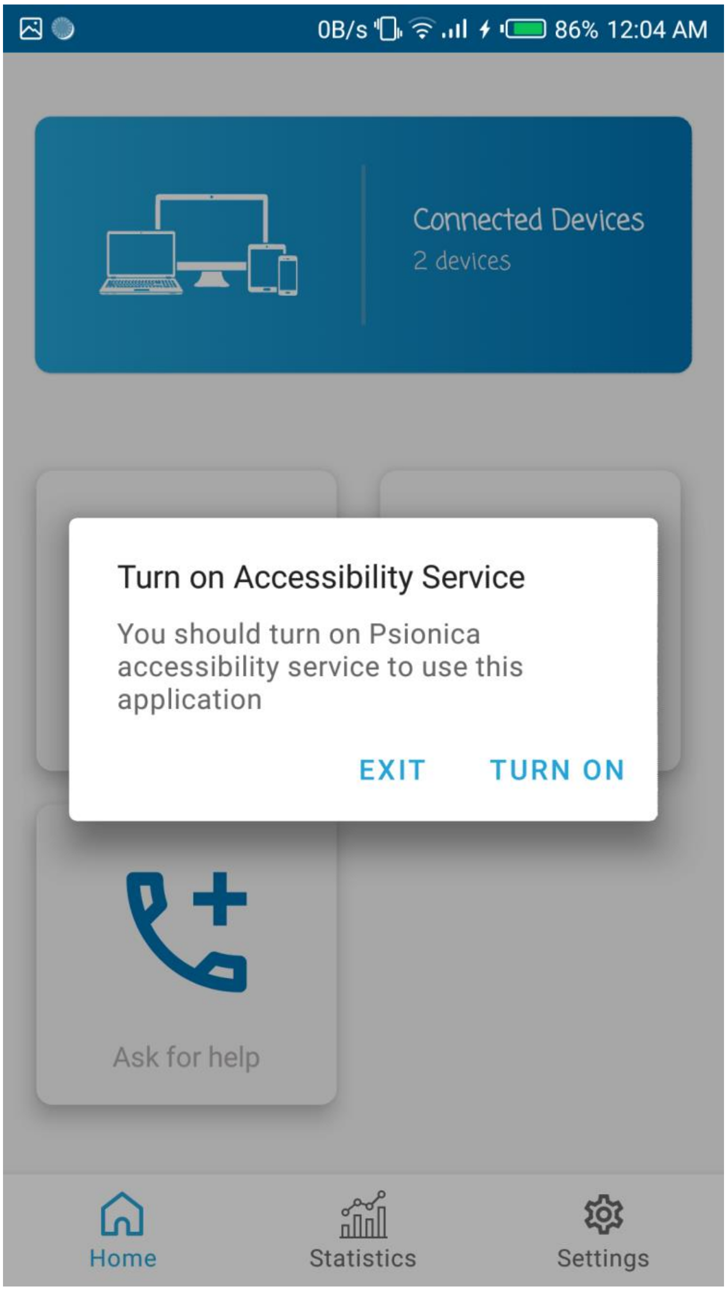


Figure C.4: Turn on accessibility service

After giving the permissions needed before, now it's time to have a look at the home page of the app as shown in figure C.5 and then we will explain every part of the app.

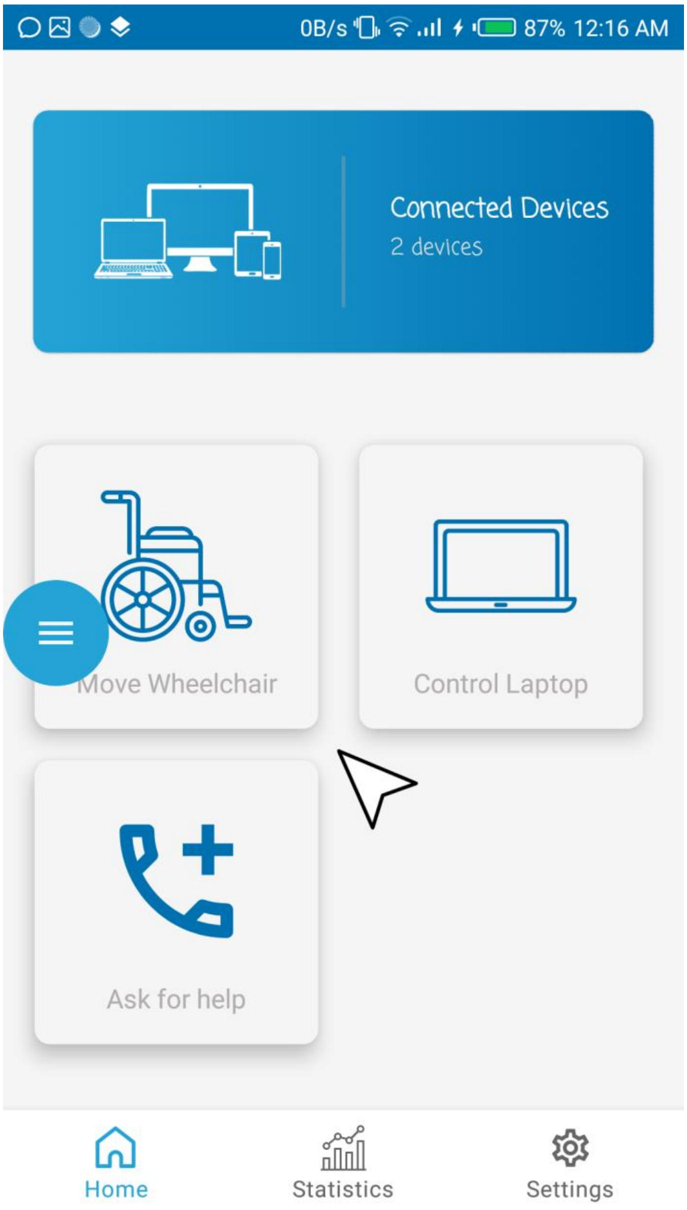


Figure C.5: Home page

Now, we will go through the wheel-chair mode where the user is able to control the wheelchair by his brain signals as shown in figure C.6.

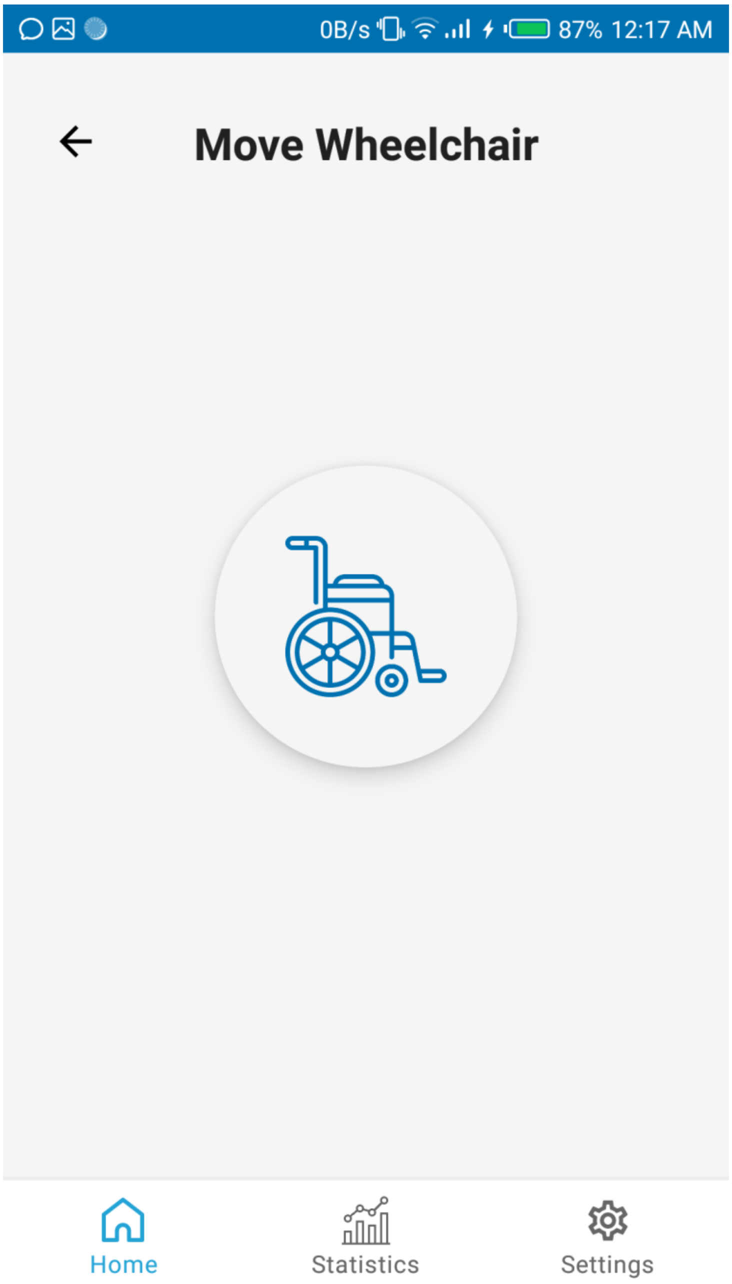


Figure C.6: Wheelchair page

Also, to enhance our UX, we show an icon with the action classified sent by the simulator after analyzing the brain signals as shown in figure C.7.

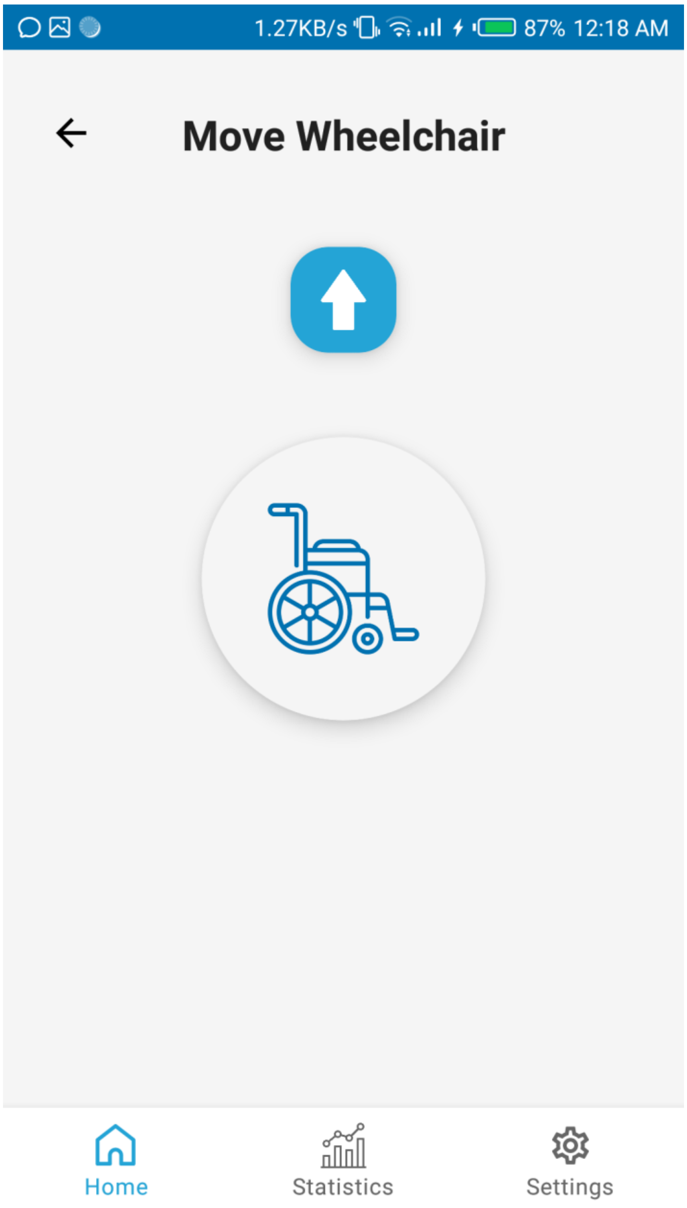


Figure C.7: Wheelchair page with the taken action

In light of the fact that we provide four actions, we got this splendid idea to make a pop-up message every period of time to get out of the current mode or to continue. This pop-up message is shown in figure C.8.

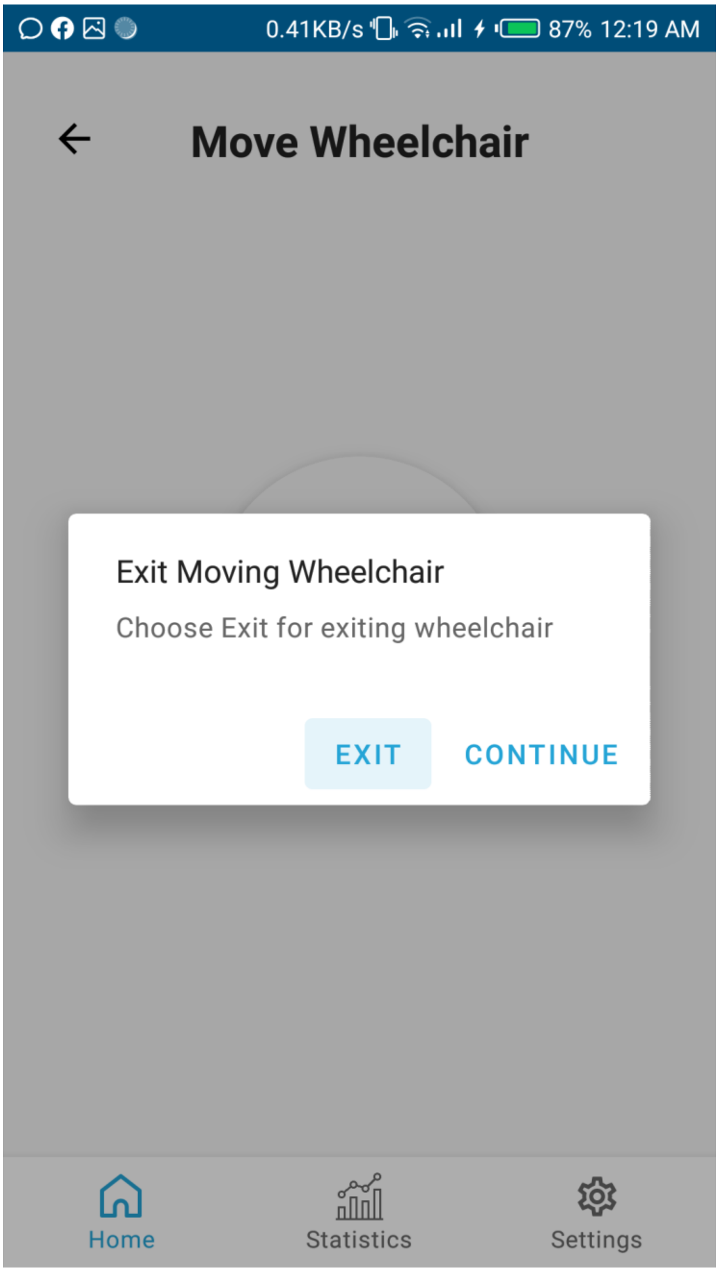


Figure C.8: Exit wheelchair mode

Also, as we introduced that we could make calls directly from the app just by clicking the ask for help mode as shown in figure C.9.

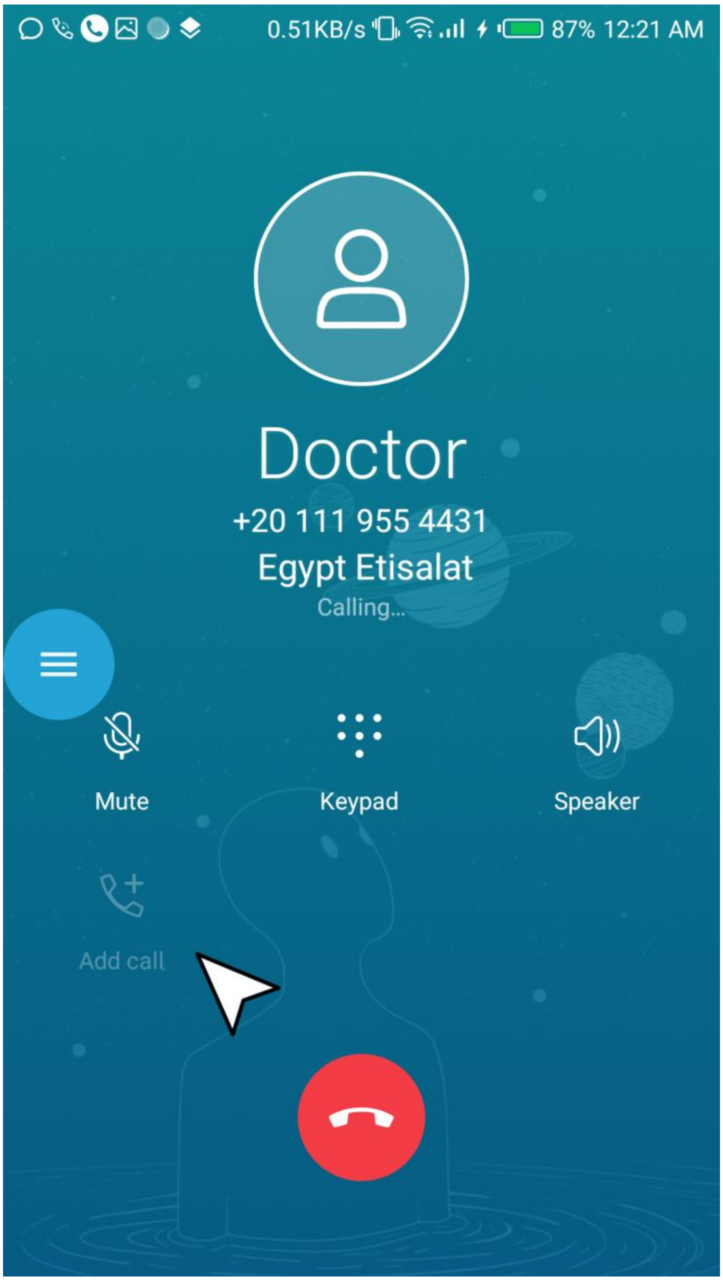


Figure C.9: Make call

Since our vision is to give the user a complete experience as much as possible, we preserved the mouse cursor outside the app as well. Therefore, the user became able to enjoy the whole experience just with his brain signals and not limited to being inside the app as shown in figure C.10.

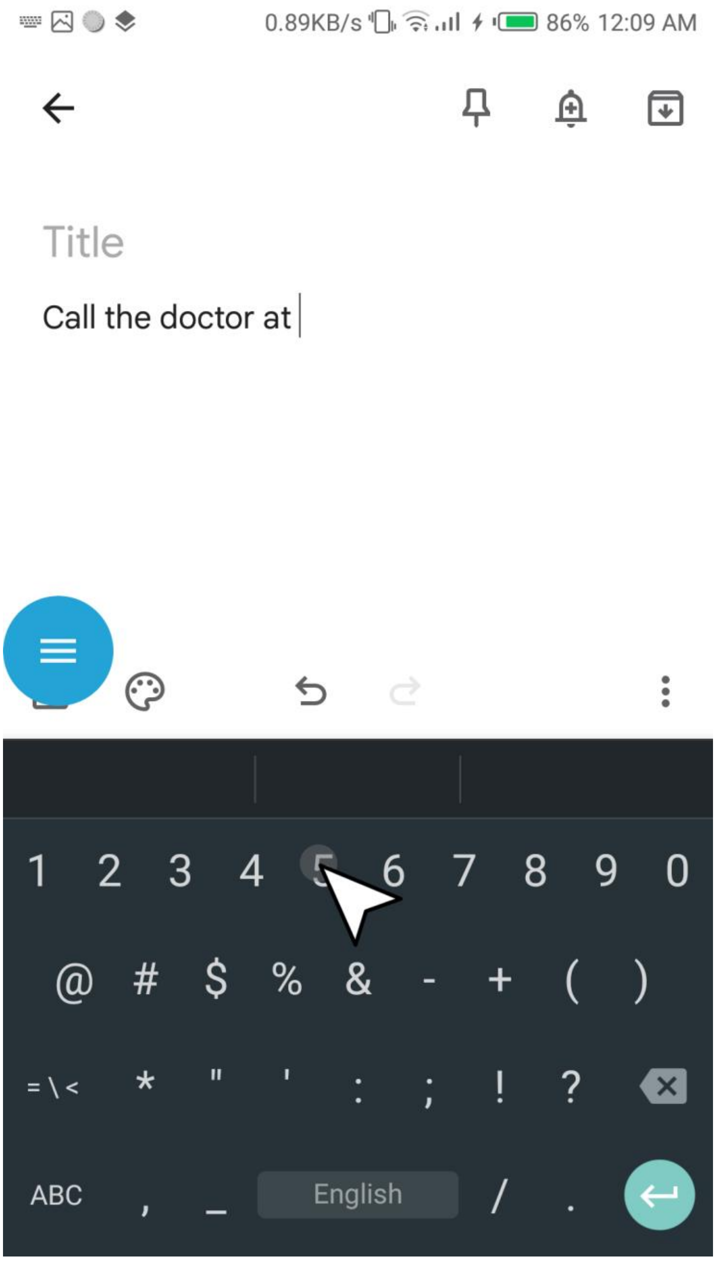


Figure C.10: Writing notes

Moreover, we added a button on the left edge of the screen which is a menu button. When this button is clicked, it opens another 4 buttons which are idle mode, back, home, and recent pages. It's designed in an easy simple way as shown in figure C.11.

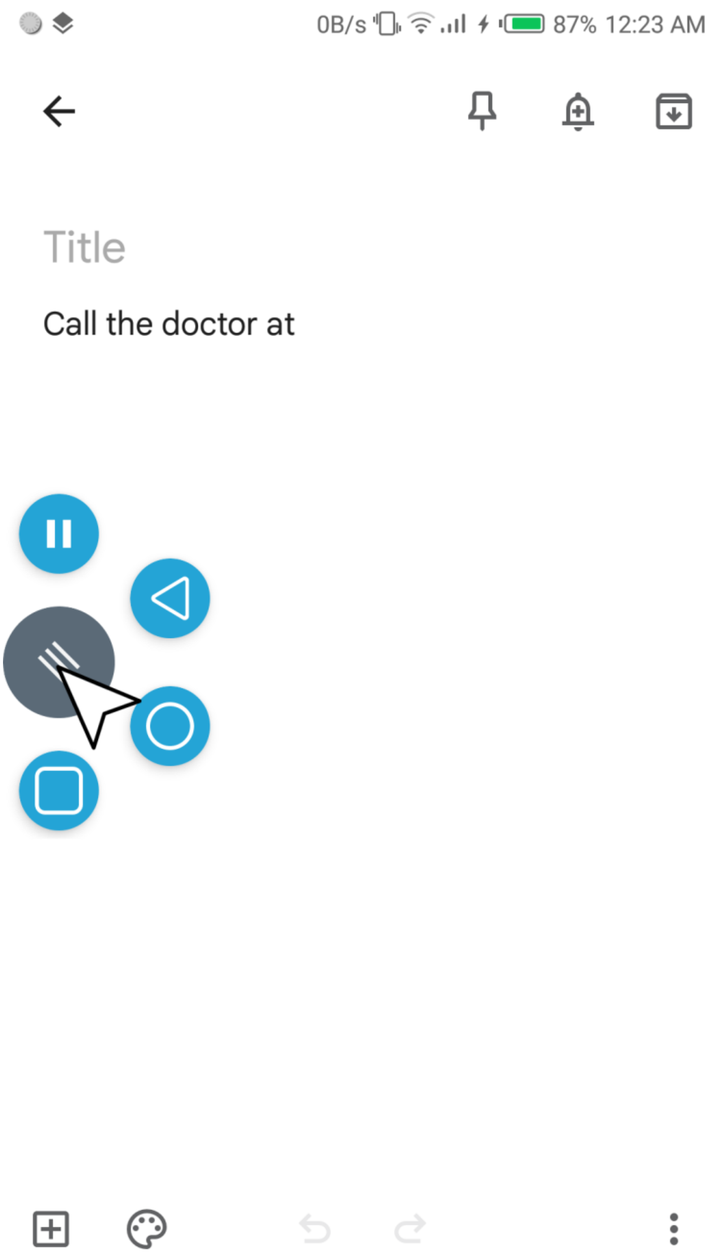


Figure C.11: Menu button

In order to increase the application customization, we allowed the user to make some changes through the settings page. The user can change the IP addresses of both the wheelchair and laptop. Furthermore, the user can change the phone number used in the ask for help mode as shown in figure C.12.

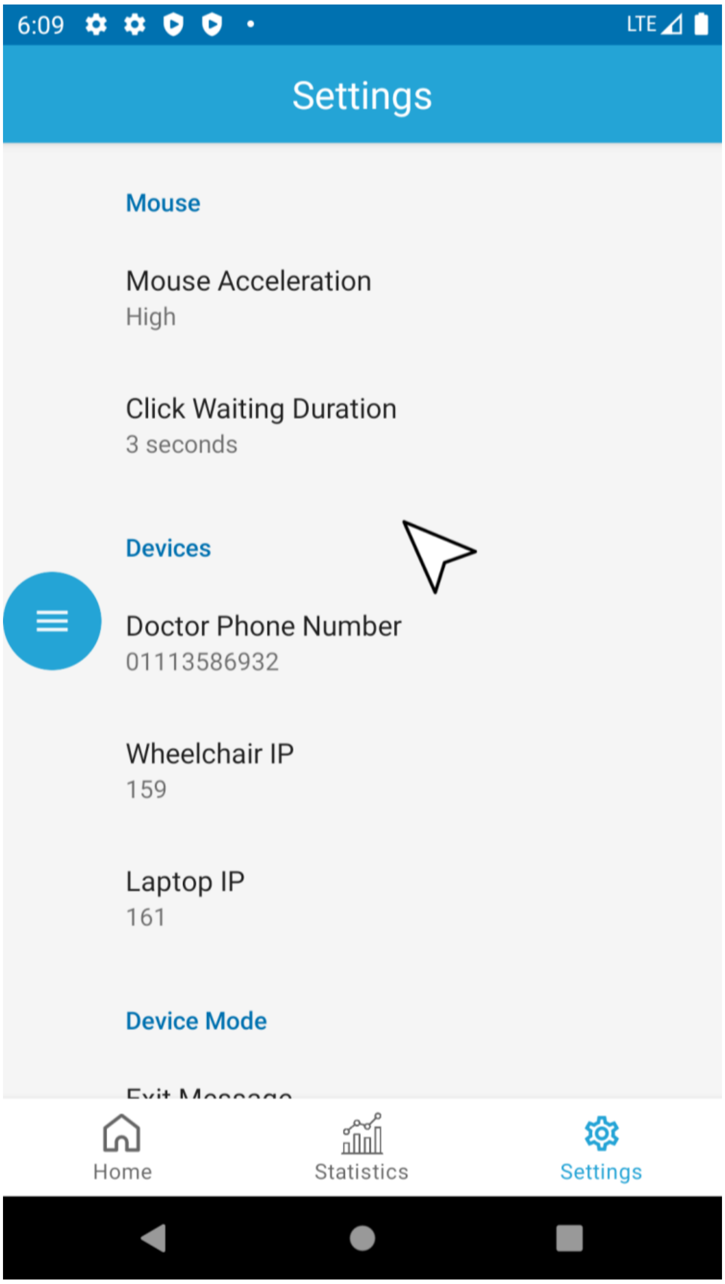


Figure C.12: Settings page

This photo illustrates the easy way of changing the phone number on the settings page. We also validate the phone number entered by the user to make sure it's not faulty as shown in figure C.13.

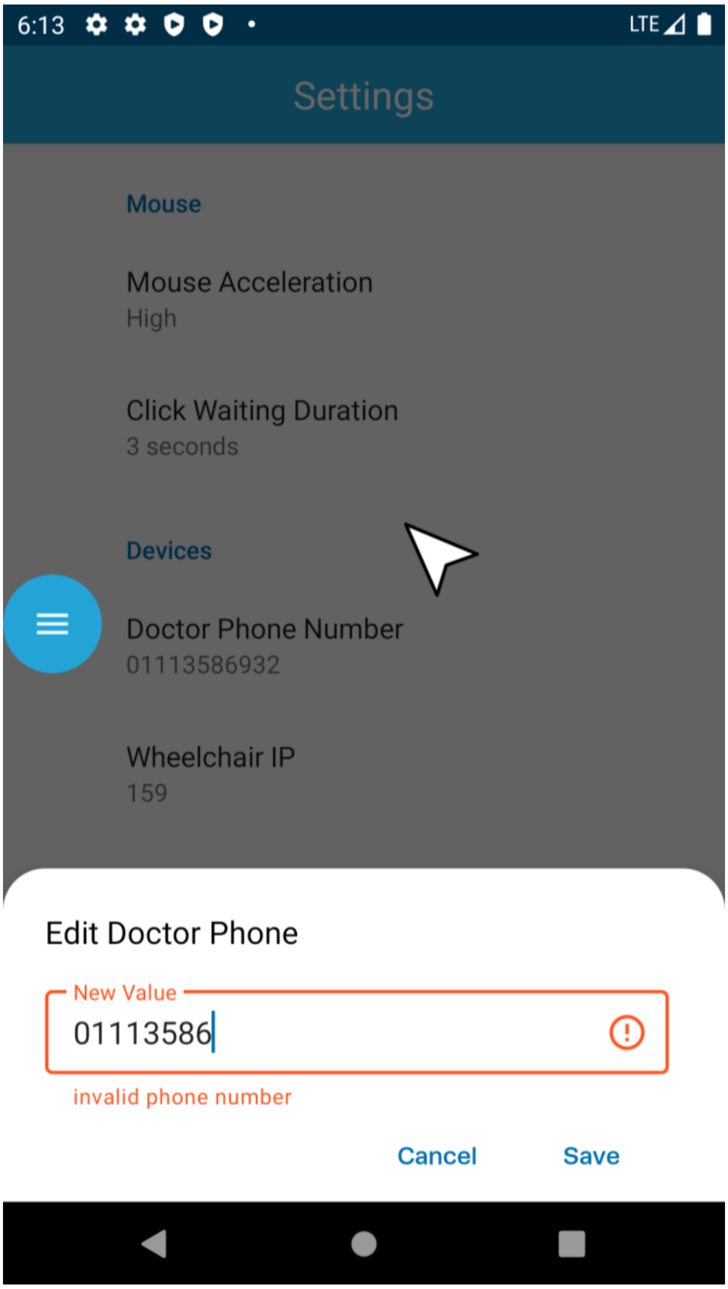


Figure C.13: Changing and validating phone number

Appendix D: Code Documentation

Action predict_action (intended_action)

This function is an essential part of the simulation process. This function takes as input the action the user is intending to take, then it selects a random sample from the test samples where the user was thinking of that same action and passes this sample through the model to get the predicted action. The predicted is most of the time equal to the intended action and sometimes it isn't. That's why our simulation is very much close to what happens in the real world.

Parameters

- Intended_action: the action the user is intending to take.

Return value

- predicted_action: the predicted action that the model classified based on randomly selected samples of data from the user.

List featurize (x_train) -

It applies Discrete Wavelet Transform on the training data and returns the coefficients which are used as features.

Parameters

- x_train: training data which is a 3d array(events,channels,samples).

Return value

- coefficients: List of coefficients as features from the training data.

Class CSP

This class has 2 member functions used to apply CSP on the input data samples.

1. Fit_Transform

It fits the CSP on the input data (training data and coefficients of the previous feature extraction stage). So CSP instances are ready to extract features in the transformation step.

Parameters

- x_train: training data which is a 3d array(events,channels,samples).
- y_train: the labels of the data samples which is a 1d array

Return value

- New_arr: a transformed array that contains the intended features.

2. Transform

It extracts features from the input data (testing data) to be used by the classifier during the classification of the testing data.

Parameters

- x_test: training data which is a 3d array(events,channels,samples).

Return value

- New_arr: a transformed array that contains the intended features.

Appendix E: Feasibility Study

E.1. Technical Feasibility

Psionica is technically feasible because of different factors such as:

- We successfully run our application on a mobile phone with ordinary specs. So, it’s expected to work on the majority of mobile phones.
- The ordinary hardware capabilities are sufficient for training our model.
- There are papers that discuss ways to solve the problem of analyzing brain signals and BCI challenges.

E.2. Legal Feasibility

Psionica is legally feasible due to the following reasons:

- It uses public software and libraries for development.
- It uses publicly available datasets that are provided by BCI competitions.

E.3. Market Feasibility

Psionica is market feasible since:

- This whole package from controlling a wheelchair, mobile phone, laptop mouse, calling your doctor immediately in case of emergency, having insights, and having a user-customized application was not provided before.
- Its pricing is reasonable