WakeUpNet

Smart Pomodoro for productivity management

Mariana Marçal Vargas

Thesis to obtain the Master of Science Degree in

Telecommunications and Informatics Engineering

Supervisors: Prof. Teresa Maria Sá Ferreira Vazão Vasques

Examination Committee

Chairperson: Prof. Paulo Jorge Pires Ferreira
Supervisor: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Members of the Committee: Prof. João Paulo Baptista de Carvalho

October 2017

Acknowledgments

I would like to start to to acknowledge my dissertation supervisor, Prof. Teresa Vazão, for her support that went so much further that the guidance during the developing of this thesis.

It is important for me to mention the rule that my professors and other employees from the institutions where I studied had. I was fortunate to meet amazing people that turned my academic journey in such an inspiring experience. I need to mention the names of Prof. Isabel Laís, Prof. Rui Valadas and Carla Boura, along with my dissertation supervisor again, who guided me to discover my personal value and face every challenge with the motivation to grow, academic and personally.

I would also like to thank my family for their encouragement and caring over all these years and specially in this final line of my master's degree. My mom, who emigrated in a late time of her life to pursue a successful career, gave me such a life lesson of how we are never too old, too weak or too limited to accomplish our goals. My godmother, who always guided me in how to turn my efforts in rewards. My aunt, who was always so kind and dedicated in helping me in my mother's physical absence. And the memory of my father, that makes me certain that he would be so proud of the success I've achieved so far under our family's name.

To all the friends and colleagues I was lucky to collect during my whole life. I owe them the most memorable moments, in the funniest and in the hardest moments. Specially, Tiago Castro and Duarte Faria, who always reminded of my personal value when I couldn't see it by myself; and André Pires, without who my academic and personal life during the college years would be much more lonely. Also, Afonso Olveira who was always available to help me with the struggles faced during the developing of WakeUpNet project.

Last but not least, I need to thank Tuna Mista do Instituto Superior Técnico, the group I integrated during my journey in this university. This group totally influenced me to be more creative, hardworking, perfectionist, and always finding joy in every responsibility I need to take.

To all of you, my sincere acknowledgements.

Abstract

Performance, time management, organization and productivity are common concerns of every student and worker. Everyone has the goal to perform his own tasks the best they can in the smallest period time as possible. This document presents a proposal for a time management system and increased productivity during the study or the work of the user. The present proposal has the innovative concept of being adaptable to the user, functioning according to their brain activity, captured at all times by a non-intrusive sensor system. The solution we propose is a mobile application, which acts as a timer to account for the time spent studying / working and taking a pause individually; and suggest the user when are the best times to take their breaks according to their actual state of fatigue. After each use, the user can analyze his performance, see the evolution of their productivity results over time and in comparison with other users. The solution presented was conceived taking into account the solutions that already existed in the market, with similar functionalities and objectives. It was thus possible to develop a solution not only capable of integrating the main advantages of the existing systems, but also of solving some of its limitations. The proposed system was then implemented and evaluated, taking into account the implementation options taken.

Keywords

Attention, Productivity, Time Management, Neurofeedback, Sensors, Mobile Application

Resumo

Desempenho, gestão de tempo, organização e produtividade são preocupações comuns de cada aluno ou trabalhador. Todos têm o objetivo de realizar as suas próprias tarefas o melhor possível no menor tempo possível. Este documento apresenta uma proposta para um sistema de gestão de tempo e aumento da produtividade durante o estudo ou o trabalho do utilizador. A proposta que se apresenta tem o conceito inovador de ser adaptável ao utilizador, funcionando de acordo com a sua actividade cerebral, capturada a cada momento por um sistema não intrusivo de sensores. A solução que propomos passa por uma aplicação móvel, que funciona como um timer para contabilizar o tempo passado a estudar/-trabalhar e em pausa, individualmente; e indica ao utilizador quais são os momentos mais indicados para fazer as suas pausas, de acordo com o seu estado real de cansaço. Após cada utilização, o utilizador poderá analisar a sua performance, ver a evolução dos seus resultados de produtividade ao longo do tempo e em comparação com outros utilizadores. A solução apresentada foi idealizada tendo em conta as soluções que já existiam no mercado, com funcionalidades e objectivos semelhantes. Assim foi possível desenvolver uma solução capaz de não só integrar todas as vantagens dos sistemas já existentes, mas também que solucionasse algumas das suas limitações. O sistema proposto foi então implementado e avaliado, tendo em conta as opções de implementação tomadas.

Palavras Chave

Concentração, Produtividade, Gestão de tempo, Neurofeedback, Sensores, Aplicação Móvel

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AC Alternating Current

ADHD Attention Deficit Hyperactivity Disorder

ADD Attention Deficit Disorder

BCI Brain Computer Interface

BES Best Effort Sleep

CPU Central Processing Unit

EEG Electroencephalography

EMG Electromyography

GSM Global System for Mobile Communications

HTTP Hypertext Transfer Protocol

IDE Integrated Development Environment

IST Instituto Superior Técnico

MDT MindSet Development Tools

NAPE Núcleo de Apoio ao Estudante

OS Operating System

PC Personal Computer

SDK Software Development Kit

UI User Interface

TGAM ThinkGear ASIC module

TGAT ThinkGear ASIC chip

TGC ThinkGear Connector

TCP Transmission Control Protocol

1

Introduction

1.1 Context and Motivation

Performance, time management, organization and productivity are common concerns of every student and worker. Everyone has the goal to perform his own tasks the best they can in the smallest period time as possible. Failing in accomplish the goals may cause anxiety, which, at the longterm, decreases the person ability to cope with the defined objectives [6]

Time is then one important factor for monitoring a person's performance. Based on previous work on time-boxing [6] and cognitive techniques [6], *Cirillo* proposed the Pomodoro technique. This technique uses a time management concept that consists in dividing the work day in work sets with small breaks in between. The work sets have a settled duration of 25 minutes and the breaks have a settled duration of 5 minutes. The main goals that were defined are: define how much effort an activity requires; cut-down on interruptions; estimate the effort for activities; create a time-table.The technique have been successfully used for work time management by individuals as well as teams [7].

In the technological era it is not surprising that mobile applications reproducing this technique have appeared fast and growing. These applications reproduce the technique through a digital timer that alternates between the count of the 25 minutes of work duration and the count of the 5 minutes for breaks duration. They alert the user when to stop working and start pausing, and vice-versa, according to the counter timer information. Different applications offer additional features for the proposes of time management, work organization and performance enhancement but all of them share the disadvantage of taking time as universal dimension for every user performing every task[1][2][3].

[1] ClearFocus: http://www.clearfocus.com , last access on January 2017.
[2] PomoDone: https://pomodoneapp.com , last access on January 2017
[3] KanbanFlow: https://kanbanflow.com, last access on January 2017

Nevertheless, the basic technique and related applications fail to adapt the time periods to the user state: sometimes the break occurs too soon and the users felt they will be able to perform their work for a longer period of time, while other times it will occur too late and the users felt already tired when the break occurred. In a time where context-aware applications are a common place, it is easy to foresee everyone would prefer one solution in this scope. Context-aware systems use context to provide the user information and services, that are relevant to his personal needs, recurring to sensors to infer the required information [8]. Also, the research about new fields of application for cheap portable sensors, able to read biological human signs and act accordingly, has been growing. Those are called biofeedback technologies [9]

We found the existent solutions in the biofeedback scope can be divided in two categories: intrusive and non intrusive. The intrusive solutions require external equipment to sense the user vital signals, while the non intrusive only use the device embedded sensors [4].

Looking at the current technology state of the art, we came up with the idea for an intelligent timer, fed by brain sensing signals in a way that could follow the real user concentration, able to dynamically manage their set of work and breaks, according to user's current individual state of focus.

1.2 Problem statement

Although the Pomodoro technique has been successfully used during years for now, and has already been accurately reproduced by several mobile applications, a big flaw remains in every version of the technique: the lack of user adaptation. Each person is different, have different work rhythms and different workdays according to his mood, his energy and the task in progress.

The only way to develop a system that could behavior according to the user biological status would be using biological sensors. Although there are already many solutions in terms of biofeedback technology, all of them offer a drawback as well. Non intrusive solutions are not accurate, and intrusive solutions are not pleasant for the user.

There is a need for an intermediate solution, i.e., a system for time management proposes, but able to read the user brain activity accurately and not intrusively, and dynamically evolve according to the sensed data.

1.3 Original Contribution and Proposed solution

WakeUpNet proposes to be an affordable, portable and user friendly tool that helps the user to organize his work in time schedules adapted to his own states of mind, with a timer fed by brain sensing signals. It aims as well to be a productivity enhancement system, by reporting to the user informative data that can help him to improve its work.

This way, we propose a system that integrates the biofeedback technologies with the time management techniques, helping every user to manage his time according to his personal capacity and offering the possibility to improve his individual productivity to his best.

WakeUpNet offers an innovative solution in two ways:

1. **Biological sensing analysis balancing accuracy and intrusiveness:** Recurring to biofeedback technologies, WakeUpNet offers an innovative solution being accurate, but not intrusive to the user. This is accomplished by using brain activity sensors integrated in a object of common use, in a way the user would not notice the sensors existence at all.

2. **Flexible time managing according to the user personal needs:** In terms of time managing, WakeUpNet offers a new solution through a flexible timer. This flexibility is achieved first by giving the user the option of choosing its own duration of work and break sets every time he uses the system. However, the system accomplishes a higher level of flexibility by tacking the user choices only as a reference, recurring to the biofeedback techniques to anticipate or extend breaks according to user real focus status.

1.4 Document Overview

This thesis is structured in six sections, as follows.

Chapter one enlighten the context of the time management problems and presents the motivation for the conception of WakeUpNet, identifies the problem, proposes a solution to it and clarifies the original contribution of the proposed solution, ending with the description of the structure of the dissertation.

Chapter two details the work done previously in the project context. Begins with the details of the human brain operation. Follows the description of how to train the human brain through neurofeedback technology. Next, the techniques and devices currently used to measure the mental status are detailed, as well as the most used techniques for improving the human attention. Finally, it's made an overview through the current system and mobile applications with similar goals to the solution proposed.

Chapter three concerns the architecture designed to the proposed solution. It begins with an overview of WakeUpNet objectives, followed by the disclosure of the system requirements to implement the given solution. Next, it's presented a brief description of the system functional overview and is made a proposal for the system architecture, detailing the most complex operations and algorithms.

Chapter four concerns the implementation of the proposed solution. First,the decisions for tools and technologies used are discussed. Secondly, the details of the implementation process concerning every component and algorithms of the proposed architecture are detailed. Finally are stated the Limitations and Restrictions faced during the implementation.

Chapter five explains how the proposed solution implemented was tested and validated, detailing the methodology, test types, results and analysis of functional as well as user tests.

Chapter six presents the conclusions and future work to be done.

2

Related Work

For the implementation of a system with WakeUpNet requirements, a research about the the brain operation was the starting point. This research is minutely described in section 2.1.

After this, it was needed to understand how the human mind states can be detected and measured, as well as how they can be improved. Section 2.2 is dedicated to this concerns.

Follows the discover of what methods and devices currently exist for capturing and monitoring brain activity and for what proposes they have been applied, in section 2.3.

Then, it is important to understand human attention itself. What it is, how can it be measured, what factors influence it and how can it be improved. This is discussed in section 2.4.

Finally, it is important to find out what mobile applications and systems currently use a similar technology and/or have goals similar to WakeUpNet. This helps to decide what features the system must include and how its architecture must be defined. This subject is then detailed in section 2.5.

In the end of this chapter are summarized the most important informations achieved trough this research, in the section 2.6.

2.1 Brain

WakeUpNet aims to be a trustful attention monitoring system. The most accurate way of measuring a human focus is reading his brain activity. So, the first subject to understand the scope of this project is the brain organ itself.

The understanding of brain's functional anatomy is essential for the accurate localization of the sensors for monitoring the organ's activity. In figure 2.1 is shown the functional neuronal structure of the human brain.

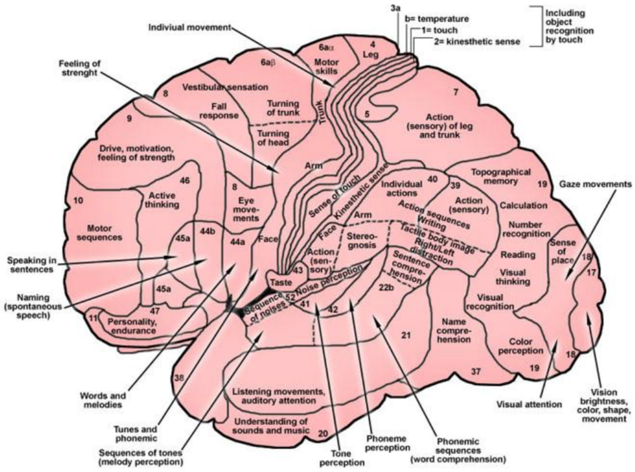


Figure 2.1: Functional localization within cerebral cortex [1]

The figure shows that there is not only one brain area responsible for the cognitive process. The cognitive function may involve the active thinking area (46 in Figure 2.1) and other functions, such as calculation, number recognition, reading and visual thinking which have distant locations. And further than that, most complex tasks involve multiple functional circuits that link many structures. That means that is not possible to designate, accurately and completely, which brain areas perform certain cognitive or effective functions [10].

Another important factor needed to understand for the conception of WakeUpNet is how to measure brain activity.

The brain cells, neurons, communicate with each others by individually sending tiny electrochemical signals [11] with particular frequencies bands, that may be called brainwaves.

Brainwaves are usually categorized into five different frequency bands known as Delta, Theta, Alpha, Beta and Gamma waves. Each of these wave types often correlates with different mental states [11]. Table 2.1 lists the different frequency bands and their associated mental states.

Table 2.1: Brainwaves Categories

Wave Type	**Frequency**	**Mental State**
Delta (δ)	0-4 Hz	Deep sleep
Theta (θ)	4-8 Hz	Creativity, dream sleep, drifting, thoughts
Alpha (α)	8-12 Hz	Relaxation, calmness, abstract thinking
Beta (β)	12-30 Hz	Relaxed focus, high alertness
Gamma (γ)	$>$30 Hz	Problem solving and memory

Every single human always has some degree of each of these brainwave bands present in different parts of the brain. For example, when someone is becoming drowsy, there are more Delta and slow Theta brainwaves creeping in. And if someone is somewhat inattentive to external things and their mind is wandering, there is more Theta waves present [9].

The individual activity of neurons contributes with small electrical current, that generates a signal strong enough to be detected by an Electroencephalography (EEG) device [11]. This technology was very important for this project, and will be detailed in subsection 2.3.1.

2.2 Neurofeedback

After the discover of how to capture and how to use brainwaves, in the late 1960s, researchers made one more step discovering that it was possible to recondition, retrain or learn different brainwave patterns. This was naturally followed as a method to achieve it: EEG biofeedback (neurofeedback), a technique for retraining brainwave patterns through operant conditioning [9].

As stated in the previous section, the states of mind of every human being are related with the amount of brainwave bands present in his brain, in every moment. Since people with mental disorders are no more than people with abnormal levels of some brainwave bands, medicine was the main field where this technique was used in the early years. Specifically, when an excessive amount of slow waves are present in the executive (frontal) parts of the brain, it becomes difficult to control attention, behavior, and/or emotions [9]. A person in this condition generally have problems with concentration, memory, controlling their impulses and moods or with hyperactivity. They can not focus very well and exhibit diminished intellectual efficiency.

Similar to other parts of our body, the brain function can be trained as well as the EEG can be rectified [12]. With continuous feedback, coaching, and practice, healthier brainwave patterns can usually be retrained in most people [9]. Neurofeedback allows the user to see his brainwaves on a computer screen a few thousands of a second after they occur. Being aware of them, it gives him the ability to influence and change them [9].

Neurofeedback has been appearing in the form of ”serious games”, games for medical, educational and personal proposes. The medical proposes focus on the treatment of psychological conditions such as Attention Deficit Disorder (ADD)/Attention Deficit Hyperactivity Disorder (ADHD), stroke, head injury, deficits following neurosurgery, uncontrolled epilepsy, depression, anxiety, obsessive-compulsive disorder, or other brain-related conditions [9]. The educational proposes, refer to the motorization and evaluation of the attention levels of the trainees at regular intervals during the training period [3]. Finally, the personal proposes include mobile applications for sleep and meditation monitoring. Later on this section, it will be stated and discussed how the neurofeedback is being used in all of this different scopes.

2.3 Techniques and devices for brain activity capture and monitoring

2.3.1 Techniques

EEG is a non-invasive technique recording the electrical potential over the scalp which is produced by the activities of brain cortex and reflects the state of the brain. It is an easy and portable way to monitor brain activities with the help of suitable signal processing and classification algorithms [12].

EEG-based systems have many different applications, one of the main ones is in the health scope. It has been used in humans with motor debilities, including those that result from spinal cord injury or amyotrophic lateral sclerosis, so that they are able to control a computer cursor in two dimensions. This technology as also been used by motor-intact individuals to command robots to manipulate objects and has the potential to be applied in operating limb prosthetics [13].

The bridge between the brain and the EEG device is called Brain Computer Interface (BCI) [2] and consists in a communication system that enables people to communicate and control devices by mere ”thinking” or expressing intent [14].

EEG is the method of signal recording used for noninvasive BCIs, that is, that do not require human penetration. And, with the development of EEG wireless devices, all the factors needed were accomplished for the expansion of EEG applications out of the lab, for personal use [12]. Consequently, we assisted to a proliferation of commercial devices for personal utilization in the recent years [2].

2.3.2 Portable Devices

Nowadays there are many inexpensive mobile non-invasive, dry, bio-sensors able to read the electrical activity of the brain to determine states of attention and relaxation, with an embedded BCI. Examples of producers are AvatarEEG Solutions [1], Neurosky [2], InteraXon [3] and Emotiv Systems [4].

Neurosky stands out of the other producers of EEG devices since their products are the most low-cost, when one of the main problems about EEG and BCI devices is precisely their expensive price about thousands dollars. Mindwave from Neurosky costs only about 80 dollars, what makes it an affordable option. Besides that, Neurosky products are easy to use and developed for leisure and not clinical use [2]. For this propose, the company produces four different headset devices, technologically similar, whose differences are resumed in appendix 8. Since Mindwave Mobile ended to be used for WakeUpNet implementation we will focus on it.

[1] Avatar EEG Solutions: https://www.egi.com/research-division/research-division-research-products/avatar, last access in January 2017

[2] Neurosky: site http://neurosky.com, last access in January 2017

[3] InteraXon: site http://www.choosemuse.com, last access in January 2017

[4] Emotiv: site https://www.emotiv.com, last access in January 2017

2.3.2.A Mindwave Device

The main function of the Mindwave device is to measure brainwave signals and monitor the attention (similar to concentration) and meditation (similar to relaxation) levels of individuals. ThinkGear ASIC module (TGAM) is the technology that allows the EEG solution, working together with dry-electrodes to sense the brain activity, filter out extraneous noise and electrical interference and convert it to digital power. A ThinkGear ASIC chip (TGAT) is embedded with TGAM, programmed with NeuroSky eSense, analog-to-digital converter, amplification off head detection, noise filtering for Electromyography (EMG) and 50/60Hz Alternating Current (AC) powerline interference [2].

It consists of eight main parts: ear clip, flexible ear arm, battery area, power switch, adjustable head band, sensor tip, sensor arm and, inside, a thinkgear chipset [2].

The headset's reference and ground electrodes are on the ear clip and the EEG electrode is on the sensor arm, resting on the forehead above the eye. The placement of the sensor arm is the ideal because this is a forehead area with minimal hair, what offers EEG clarity to enable the accurate delivery of raw and powerband. Moreover, this location is ideal to measure higher cognitive processes including Attention and Meditation algorithms. Given the proximity to the eye, it also enables blink detection [2].

The principle of Mindwave headset operation is quite simple. Two dry sensors are used to detect and filter the EEG signals. The sensor tip detects electrical signals from the forehead of the brain. At the same time, the sensor pick up ambient noise generated by human muscle, computers, light bulbs, electrical sockets and other electrical devices. The second sensor, ear clip, is a ground reference, which allows thinkgear chip to filter out the electrical noise. Figure 2.2 presents the device design.

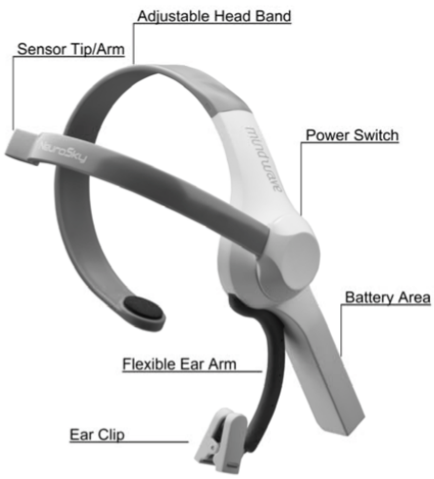


Figure 2.2: Mindwave Mobile device design [2]

Mindwave Mobile is compatible with Mac Operating System (OS), Windows and mobile devices with iOS and Android, and as Neurosky offers free Developer Tools it is easy to create innovative apps, compatible with different platforms, that respond to a user's brainwaves and mental states. It is called

Neurosky MindSet Development Tools (MDT) and can be free downloaded from the Neurosky Store [5].

MDT is a collection of drivers, sample code, and documentation describing how to develop applications for several software platforms, including Personal Computer (PC), Symbian, and even lower level platforms such as micro-controllers like the Arduino.

MDT supports supports many different coding languages, including C++, C#, Java and J2ME, and provides the ThinkGear Connector (TGC). TGC is a daemon-like software that runs on Windows or Mac OS X, for opening a Transmission Control Protocol (TCP) port on the user's local computer so that applications can connect to it and retrieve MindWave data. As long as the TGC is running on one of the supported platforms and connected to a Mindwave device, any application written in any language that can communicate through TCP sockets can connect to the TGC to read data from the Mindwave device [15].

Finally, it is important to clarify what is NeuroSky eSense to understand its importance for the WakeUpNet implementation. eSense is the algorithm developed by NeuroSky to quantify mental states. To calculate eSense, the NeuroSky ThinkGear technology amplifies the raw brainwave signal and removes the ambient noise and muscle movement. eSense algorithm is then applied to the remaining signal, resulting in the interpreted eSense meter values.

Mindwave Mobile offers four EEG algorithms that work with its TGAM: Attention (similar to concentration), Meditation (similar to relaxation), BandPower and Blink Detection [15]. For WakeUpNet, only the first two were used.

2.4 Attention

Concentration or sustained attention is the basic cognitive ability of a person to perform any task or develop a skill. The timely monitoring of this ability is important when his/her performance enhancement is concerned [3].

One of the most commonly used ways of improving human concentration consists in managing the time associated to the realization of a given task, by defining short breaks between working periods. This is usually known as Pomodoro technique and will be discussed in the following section.

Knowing the factors that have impact on the attention is rather important, but one need to find out some way of improving it. The appearance of neurofeedback techniques showed that it was possible to retrain the brain. Neurofeedback games introduces one method for training the brain in order to reach a focused state of mind the faster and the longer through video games. This topic will be presented in subsection 2.4.2.

[5] Neurosky: http://developer.neurosky.com/ , last access on October 2017

2.4.1 Pomodoro technique

The Pomodoro technique was created with the aim of using time as a valuable ally to accomplish what we want to do and to empower us to continually improve our work or study processes [6]. It appeared as personal time management tool for individual work proposes, but, currently, is also used for team work, e.g. in extreme programming teams [7].

The traditional Pomodoro consists in organizing the work time in cycles (”pomodoros”) of 30 minutes long: 25 minutes of work plus a 5 minute break. Frequently implemented through kitchen timers shaped like a pomodoro (the Italian for tomato) [6] soon appeared software applications with similar timers as we will present later.

The inventor of Pomodoro based the methodology on scientific proofs that 20 to 45 minutes of working sets can maximize human attention and mental activity, if followed by a short break. Pomodoro pretends to be a different way of seeing time, alleviating anxiety. Abstracting time, Pomodoro encourages conscientiousness, concentration and clarity through effective time management [7]. And a better use of the mind enables, for itself, facilitates learning. Besides that, Pomodoro is an easy-to-use and unobtrusive tool that reduces the complexity of applying the technique while favoring continuity and allows the user to concentrate his efforts on the activities the user wants to accomplish [6].

The big flaw of this technique is being so restricted and unadaptable to the user. After adaptable techniques like neurofeedback have been discovered, it is possible to create working and studying enhancement methods based on the real activity of the brain. Close to this, a technique that bases the human concentration performance only in time sets seems weak.

2.4.2 Neurofeedback games

Neurofeedback games are computer games in the field of health and education, developed to improve one or more cognitive abilities of the users.

The difference between neurofeedback games and regular gaming applications is that, while the last ones require a graphics tool for the design of the gaming environment and user interface along with devices like keyboard, mouse or a joystick type of controllers to control the game; EEG based games use “thoughts” as inputs to directly control game.

So, in order to play neurofeedback games, the user needs an EEG reading device, like Mindwave from Neurosky, and an interpreter (like a smartphone, a computer or a PlayStation).

In figure 2.3 is an example of an architecture for a neurofeedback game that explains in a simple way the game functionality.

The EEG device safely measures and outputs EEG power spectrums (the brainwaves).

The software provided with the EEG device is enough to connect the dispositive to the computer,

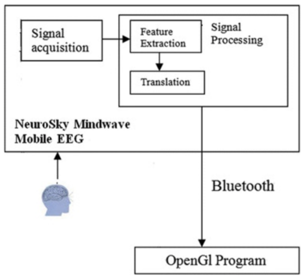


Figure 2.3: Block Diagram of a Neurofeedback Implementation [3]

allowing the game to be run according to the player's states of mind as detected by EEG headset with NeuroSky's ThinkGear sensor. The software runs as a background process continuously by keeping an open socket on the computer system, thereby allowing applications to connect to it. The applications receive information from the connected ThinkGear devices, in digital form, through Bluetooth connectivity.

The input to this gaming application is through the attention values, that are indicated and reported on a meter with a relative eSense scale of 1 to 100, being 50 considered the threshold value for a fair attention [3]. There are alternative methods for scaling states of mind, but it is not proved they are better than eSense scale [12].

When the threshold value is crossed, the application sends some reward feedback to the player. For example, if the game objective is to enforce the user to move a ball in direction of a certain destination, by being concentrated; the players ball will move forward towards the destination when the threshold value is reached, else it descends [3].

Another examples of existent neurofeedback game are "Dancing Robot", a simple single-player where the player is required to control the speed of robot through concentration level while the robot is dancing; and "Brain Chi" where controls a little boy to fight against evil bats using a protection ball. The size of the protection ball is controlled by the concentration level of the player and to win this game, the player needs to increase the size of the ball by concentration to eliminate all the bats [12].

Analyzing the effects of this type of games in the human brain it is concluded that similar to other parts of our body, the brain function can be trained as well as the EEG can be rectified. Neurofeedback can help an healthy person to learn how to to increase special components of EEG signals with neurofeedback, to enhance his brain functions [12]. Based on this information, one of the objectives of WakeUpNet is to incorporate, somehow, one or more levels of neurofeedback games so the user can practice how to reach and maintain his focus anytime.

2.5 Applications and Systems

There is a big diversity of mobile applications that share some characteristics with WakeUpNet. An overview through them is important to select the most important features WakeUpNet must include, as well as inspire for the conception of WakeUpNet architecture.

This mobile applications include software of Pomodoro timers; applications for monitoring and improving relaxation and attention; sleep monitoring applications and devices and finally an application for the monitoring of drivers vigilance.

2.5.1 Pomodoro applications

In the recent years, with the technological boost, a big variety of software applications that apply the Pomodoro technique appeared.

All these applications have a basic common functionality - the digital timer, configured to break down work time in 25 minute intervals separated by 5 minutes length breaks. But, the software implementation has a great advantage: it allows to take the Pomodoro technique to a next level by including several additional features that improve the traditional technique. This features include forms of minimizing user's distractions, integration with other systems and statistics and reporting.

- **ClearFocus:** This app provides features for reducing external distractions include disabling the Wi-Fi connection and setting the device to silent automatically during the working time and resetting the previous configurations during the break time[6].
- **PomoDone:** PomoDone, for example, besides not being a mobile application but a web and desktop app, is a good example of how the integration with other systems his a good complement to the timer. In this case, the app can be synchronized with other productivity applications, like Asana, Wunderlist and Evernote. While these applications collect the user's tasks, PomoDone keeps track of the time spent with every one of them[7].
- **KanbanFlow:** KanbanFlow goes even further. In this case, the timer is not the main feature, but one of them. The application integrates a big variety of project management tools. It consists of a board providing an overview of all the work situation, including the tasks in different development status, organized by swimlanes, colors, labels and filters; the sub tasks that compose each task, and allows the collaboration with multiple participants and the integration with platforms such Dropbox and Google Drive for file management. A Pomodoro timer appears as an optional feature to track the time spent in each track. Alternatively, the user can use just a regular stopwatch timer or log time manually as he prefers. Finally, the application provides useful reports and statistics, allowing the user to see how much time was spent on different tasks, to keep track of his own time, to review what his team members have been working on, to find out what type of interruptions are causing the productivity loss and to use that information to adjust the work process[8].

[6] ClearFocus: http://www.clearfocus.com , last access on January 2017
[7] PomoDone: https://pomodoneapp.com , last access on January 2017
[8] KanbanFlow: https://kanbanflow.com, last access on January 2017

Analyzing the Pomodoro mobile applications offered nowadays, they all have the same disadvantage mentioned in section 2.4.1: they are strict and unadaptable to the user. If the user is in a high level of concentration at the end of the 25 working minutes, interrupting his productivity to enforce a break may not be helpful. It would be better to wait until he starts getting distracted to make that pause and in the case he already worked more than the defined 25 minutes allow him to make a bigger break. Besides that, during the break, the user may be still thinking about work and not actually relaxing at all and the 5 minutes of break time may not always be enough.

Besides this flaw, Pomodoro applications offer some valuable characteristics, namely the capacity to minimize user distractions shutting off Wi-Fi and activating silent mode during the work time as a measure for improving user's focus. Statistics and reporting are also an important part, with the goal of showing to the user his progress in terms of the time he's able to maintain his focus, how often he needs a pause and suggesting the set up configuration that is more suitable for his natural work rhythm.

2.5.2 Meditation applications

Another type of applications related to this project scope are Meditation applications.

This type of applications have the objective of helping the user to train his own mind to reach a high level of mindfulness.

- **Headspace app:** Headspace accomplish the cited propose with daily sessions of meditation tutorials with ten minutes of duration. The objective is to give the user the necessary tools to apply the lessons during all day, during his ordinary life. The app triggers reminders so that the user would not miss his daily session and attributes rewards for regular meditation in order to keep him motivated[9].

  After the free program is complete, the user can access additional inspiring packs and sessions on a range of specific topics, such stress, happiness and appreciation. Headspace was made for desktop use, but it is possible to download the sessions for portable devices.

  One interesting feature of this application is that it is integrable with Google Fit and Google Calendar. Google Fit is a platform for personal analysis, monitoring and store of fitness information. The integration allows the user to see his mediation progress in Google Fit, with the reports and statistics provided for this platform.

- **Muse app** Muse claims to be the first tool in the world that gives accurate, real-time feedback on what is happening in the brain when a person meditates[10].

[9] Headspace: https://www.headspace.com, last access on January 2017
[10] Muse: http://muse.mu, last access on January 2017

It aims to teach, to incentive and to monitor user's meditation. The user is submitted to attention-based training exercises during three minutes sessions and receives awards for his achievements.

But the big innovation is the connection with a headband, connected with the portable device through Bluetooth, that is able to read brain signals. When using the application, Muse begins for taking a snapshot of the brain in his natural state. This snapshot is used as a reference to understand the user brain signals. Next, the brain signals are used to be concerted into the sounds of wind. When the mind is calm and settled, the user hears calm and settled winds. Instead, when his mind is active the winds pick up and blow. After each session, and over time, the user can track his progress through a series of graphs and charts.

The objective and the strategy to accomplished by both applications is similar. However, although Headspace aims to help the user to train his own brain in order to achieve deeper and longer focus, just as the Pomodoro applications, lacks in considering the real brain activity. The solution offered by Muse is definitely more robust and accurate, since it takes advantage of the possibility of collecting biological data from the users to show them the concrete progress of their work.

2.5.3 Sleep monitoring applications

The best way to monitor sleep quantity and quality is through polysomnography, a study that includes monitorization of brain waves, eye movements, muscle contractions, blood oxygen levels, snoring and restless. For monitoring all of this, it is needed to execute a serie of complex techniques and use different set of equipments, such as EEG, electrooculography, pulse oximetry, microphones and cameras; what makes polysomnography too much complex and expensive to be implemented in a large-scale long-term sleep monitoring.

But with the emerging quantified-self apps and wearable devices, it became possible for common people to measure and keep track of sleep duration, patterns and quality [4]. The alternatives for personal use can be grouped in two: Intrusive and non intrusive.

The unobtrusive ones do not require much user intervention nor the use of external devices to collect biological data (e.g. headbands). But the intrusive methods, besides that disadvantage, are able to achieve better are more accurate results.

Some examples of unobstrusive methods are:

- **Sleep Cycle and Sleep Time:** Mobile applications like SleepCycle[11] and Sleep Time[12] offer a set of interesting functionalities in the scope of sleep monitoring. They use the device embedded sensors - like the accelerometer, light and sound sensors - to determine the phases of the user

[11] SleepCycle: https://www.sleepcycle.com , last access on January 2017
[12] Sleep Time: http://www.azumio.com/s/sleeptime/index.html, last access on January 2017

sleep. The objective is to wake up the user in the most appropriate sleep phase, acting like a smart alarm. They also collect the sleep data and integrate them in graphs and statistics so the user can see his complete history.

- **Best effort sleep** Best Effort Sleep (BES) is another unobtrusive method discussed by [4], that recurs only to smartphone usage patterns (e.g. the time and length of smartphone usage and recharge events) and environmental observations (e.g. prolonged silence darkness) to predict sleep duration.

To understand the effectiveness of the unobtrusive methods in general, and the BES method in particular, let us see a comparison of it with other three methods with intrusive characteristics - Zeo, Fitbit and Jawbone - resumed in table 2.2.

Table 2.2: Characteristics of different sleep monitoring methods

	Sleep Monitoring Method	**User Setup Required**
Zeo	Headband	Yes
Jawbone	Wristband	Yes
SWP	Smartphone's embbeded accelerometers	Yes
BES	Smartphone usage patterns	No

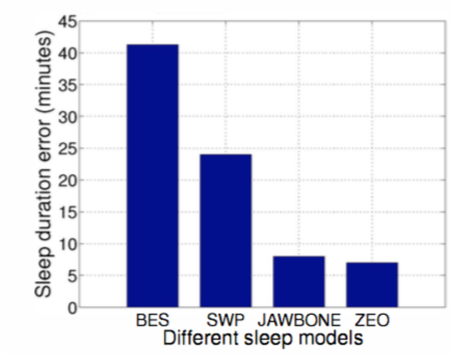


Figure 2.4: Overall sleep duration error for BES compared to the three alternative sleep monitoring systems (SWP, Jawbone, Zeo). [4]

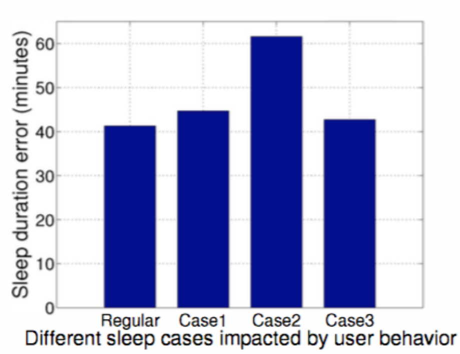


Figure 2.5: BES sleep duration error when users behave as expected ("regular") compared to three examples of atypical sleep behavior (i.e., corner cases). Specially, these behaviors are: Case1 - user sleeps with the room lights on; Case2 - user has a prolonged nap during the day; Case3 - user fails to recharge their phone. [4]

The results of the comparison tests, depicted in figures 2.4 and 2.5, show that the unobtrusive form produce much more sleep duration errors than the intrusive forms, situation that is a aggravated when the ideal conditions for BES use (user sleeping with the room lights off; user sleeping only at night during all day and user recharging his phone in the moment of sleep) are not met [4].

The conclusion is that besides being more expensive and intrusive, the options for sleep monitoring that involve wearables and user configuration are more accurate that the unobtrusive alternative ones [4].

2.5.4 Drivers vigilance system

Another field of research for the use of EEG signals is the monitoring of human's capacity of driving.

Driver's fatigue is a concern since it involves a big safety risk and that motivated researches to explore ways of estimate and report in real-time the driver attention [5]. In order to accomplish that, a real-time vital signs monitoring system is required. This system is able to read the main driver's vital signs and emotional parameters such as heart rate, blood pressure, oxygen saturation, glucose level, body temperature and muscle stamina. This implies an oxygen saturation sensor, electrocardiography and electrical conductance of the skin which can be used to detect the driver emotional status [5]. Additional to this, is the measuring of driver's concentration through EEG.

Figure 2.6 shows an appropriate form of the integration of the sensors needed to a system that collects user biological data, interpret it and triggers an alarm in case of the driver's medical status is not in good condition (and consequently, not eligible to drive). Additionally, sends the user vital information to his doctor that may confer it at any time over Internet and is saved on a database.

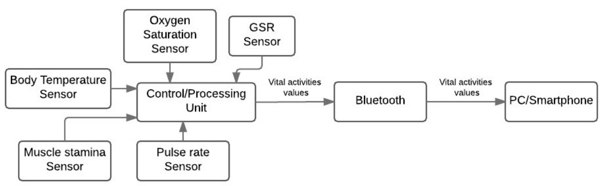


Figure 2.6: Sensors integration for an attention monitoring system [5]

The control unit of the proposed design must receive the measured signals from all the sensors, process signals, convert it to real vital activities values in form of digital waves and put this data somehow that can be transmitted for the receiver device using Bluetooth. It is the first main component of the real-time vital signs monitoring system, that incorporates the sleep detection and warning system to ensure that the driver medical status is in good condition and check if the driver is eligible to drive. The other ideal components and the way they work as a whole are shown in figure 2.7.

As mentioned above, the various sensors send the vital activities measurements to the control unit, that treats the data and transmits it via the Bluetooth module for every connected driver device separately. The driver devices interpret the received data and if there are some abnormal values among the

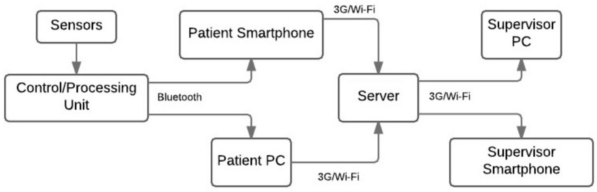


Figure 2.7: Real-time vital signs monitoring system architecture [5]

vital activities measurements they trigger the driver alarm. In any case, they send the signals to the server.

The server receives the vital activities measurements from all the drivers, puts this data in an organized format and saves it in a text file. Then, sends the information to the doctor devices being able to choose which device - smartphone or personal computer - he wants to see his data from his personal applications. The doctor can send his recommendations to the driver as well [5].

The architecture presented in figure 2.7 was the starting point for the conception of WakeUpNet architecture.

2.6 Synthesis and discussion

This chapter presented the main subjects that were considered relevant for WakeUpNet design - an application that offers a dynamic Pomodoro by adjusting the time intervals to the person's status.

The research about the brain operation contributed to understand how brain activity can be read and measured. The conclusion was that the presence of specific brainwaves are associated with correspondent states of mind. Namely, the capture of Betha and Gamma waves will the determinant to detect user's attention.

Neurofeedback is a technique for accurately and non-intrusively measure and improve the user concentration. It allows training the brain to reach and maintain certain mind states such as attention or relaxation. This technology is suitable for being integrated in applications for productivity assessment and enhancement like WakeUpNet.

EEG is a suitable technique for brain activity monitoring. It is non-invasive, easy and portable. Initially used for medical proposes, nowadays it is integrated in portable devices commercialized for the general public.

Neurosky stands out as manufacturer of portable EEG systems for offering the most economic solutions, with high accuracy performance. Their products use are much effective for the propose of reading brain waves, what makes them perfect for the implementation of WakeUpNet.

The Pomodoro technique has been widely used as a tool for productivity enhancement and time management. Applications that offer a software implementation of this technique made it easier to use, but fail to include the user natural work pace and response to stress and drowsiness. There is a market lack of performance and work improvement applications with this characteristic, that WakeUpNet pretends to fulfill.

Neurofeedback games, on the other hand, take advantage of EEG technology exactly for reading the brain activity and develop user's capacity for reaching determined mind states, namely attention, as fast and longer as possible.

The ideal application for concentration enhancement would have a combination of this three elements: a smart timer, recurring to EEG to determine the best time for making pauses and turn back to work and reporting useful information for performance improvement.

Meditation and Sleep monitoring applications are already used in everyday life. The most invasive systems for this propose are the most accurate, so an intermediate method that is not invasive enough to be disinterested is the ideal. Still, an application for the work time organization and focus improvement during the workday is still missing and, once again, that is an innovation that WakeUpNet intend to be.

Drivers monitoring is one of the fields of application of EEG technology that meets a very close objective to WakeUpNet. The architecture of this systems is a good example and contribution for the conception of WakeUpNet architecture.

3

WakeUpNet Architecture

The design of a system architecture before its implementation is indispensable to prescribe the structure of the system being developed. Recurring to the information about the existent applications mentioned in section 2.5 it was possible to conclude what were the features that WakeUpNet wanted to include and what components its architecture must integrate to provide them.

This chapter begins with an overview of WakeUpNet objectives in section 3.1, followed by the disclosure of the system functional and non functional requirements that were needed to implement the given solution, in section 3.2. Section 3.3 presents a brief description of the system functional overview. Then, it is made a proposal for the system architecture. This begins with an Architecture overview in section 3.4, and follows the description of every architecture component in section 3.5.

Finally, the most complex operations are detailed in dedicated chapters: the communication between the two major architecture components, in section 3.6, and the details of the core system behavior is given in section 3.7.

3.1 Goals and Overview

WakeUpNet has the goal of improving productivity by dynamically adapting the user's activity to the brain status and displaying information about the productivity of the user's work day.

In order to accomplish that, it first aims to be an intelligent Pomodoro timer, where the work cycles and breaks are not rigid to the traditional cycles of 25-5 minutes. Instead, the user determines the time period during which he pretends to work and pause. This time period is only a reference time, since the real period times are adaptable to the user's brain activity: if the user is tired, the system can anticipate a pause; if the user still haven't rested enough after a break, this break can be extended; if the user is being productive will work during the time chosen by himself.

To display useful data about user's productivity that may be used for statistic proposes, the user must be able to access data about his performance in three forms:

1. After each session: After every WakeUpNet utilization (for now on we'll call it session), the system must retrieve data about the user performance during the last system run.

2. Along all the sessions: WakeUpNet must retrieve data about all the user sessions performed in the past, showing his evolution.

3. Comparing with other users: the user can compare its session results with the results achieved by other WakeUpNet users with similar profile characteristics.

WakeUpNet was idealized on its best form, suitable for commercial proposes, what is a very ambitious goal for a final dissertation project. During the implementation, several modifications were made in the initial architecture. The next the next section details the major requirements for the system design on its fullness.

3.2 Requirements

To accomplish the goals defined in previous section the complete platform of WakeUpNet must satisfy a set of functional and not functional requirements.

3.2.1 Functional requirements

In terms of functional requirements, i.e., requirements that describe a system functionality, the following ones were defined:

- **Authentication:** Proper authentication operations, as well as another security issues, such as authorization and encryption of transmitted data must be assured. This is an important requirement since the users want to make sure its personal data is not disclosed to undesired entities.

- **Configuration:** WakeUpNet aims to provide a personalized experience to every user, so the user must be able to configure the system according to its personal needs and preferences. Consequentially, the system needs to be configurable, in a way the user can define his work duration, according to his individual needs and goals.

- **Time management:** The system must work according to the Pomodoro Technique concept, alternating between work and break moments that occur at the most appropriate times and with the ideal duration to the user state of concentration and relaxation. The system must keep track of the time the user spend in each work and break moments, respectively.

- **Sensing data gathering:** WakeUpNet must be able to accurately collect biological data in terms of the user brain activity information, namely his status of focus and calmness. This data must be collected continuously during the system utilization.
- **Context Awareness:** The system operation must dynamically change according to the stream of data collected every moment.
- **Data processing:** The sensing data must be collected at every moment and correctly be interpreted in order to influence the system operation; also it has to retrieve useful and trustful results to the user.
- **Data analysis and presentation:** At the end of each session, the user will want to know about his performance and, probably, about his evolution hrough several WakeUpNet utilizations, as well as his position in comparison with the other WakeUpNet users. So, the system must display the accurate results, concerning graphs and statistics of the user performance in an appealing and explicit form.
- **User Adaptability:** One of the main goals of WakeUpNet is to operate according to the individual characteristics of each user, not only in terms of its personal preferences, but about its personal work flow every moment.

3.2.2 Non functional requirements

Non functional requirements refer to global qualities and system attributes. Follows the non functional requirements that have been identified and must be respected:

- **Mobility and Portability:** The system must be accessible through different mobile devices, so the user can use it in any workspace with the devices he has access at every moment. This implies the system to use wireless technology to data transfer and to maintain the stored data accessible every time and everywhere.
- **Availability in case of disconnection:** Since WakeUpNet is made for using during work or study, the user may prefer to execute it with no Internet connection, what may be a factor for loosing focus. So, the system must be able to run in case of Internet disconnection, even if some of the functionalities become compromised. For example, the system may have local computation capacity to generate and display the results of the current session in its end; however to access the results of the sessions in comparison with other users, Internet connection is required.

- **Performance;** The system execution and processing must result in acceptable times for responding to user actions and retrieving data.
- **Scalability:** To display the user results along time, e.g., in the last month, the system needs to keep record of the user results in every run made during the last 30 days. Moreover, to display the user results comparing with other users, the system needs to maintain a history of every user results. This implies the system needs to deal with increasing use and size of data, never discarding an acceptable response time.
- **Extensibility:** The implementation must take future growth into consideration, in terms of addition of new functionalities or through modification of the existing ones.
- **Modularity:** The component services must be weekly interlinked to increase the responsiveness in dynamic environments where available services are not available.
- **Accuracy:** The system aims to collect accurate sensing data and deliver the user trustful results.
- **Low Cost:** WakeUpNet must be low cost. This implies using hardware and services with affordable prices.
- **Usability:** The users must be able to access WakeUpNet to accomplish their objectives learning how to use it easily, through an intuitive interface and suitable documentation. The user must find it non intrusive and comfortable to use. This implies a special concern for the sensors implementation, since the common user is not usually familiar with sensors utilization. So, the sensors must be similar to an object the user is familiar with, like headphones. To allow the user use it during his ordinary routine, the sensors also must be wireless.

3.3 Functional Overview

To use WakeUpNet, the user must use a brainwave sensor device and a mobile device through which he can access WakeUpNet mobile application.

An example of a common WakeUpNet use case is show in figure 3.1.

The first task the user must execute is authentication. In case the user is a new user, he must sign up first. Sign up regards to not only inserting credentials (username and password) through the user interface, but also its profile data, such name, age, gender and professional activity (worker or student). This information is personal and so it is kept private, requiring authentication any time the system is accessed again.

Secondly, the user must configure WakeUpNet timer, preparing an initial setup of the pretended work and break duration time. The system will use that value as reference to keep track of time, but the real

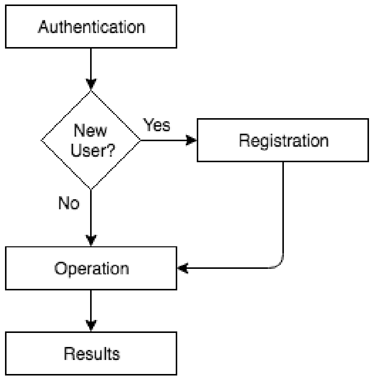


Figure 3.1: Use Case Example

user focus and drowsiness status will affect its operation. This is accomplished recurring to a wearable device with EEG technology, in order to monitor the brain activity. This device must be portable, easy to use, affordable and accurate.

When finishing his work session, the system reports the user useful information about his performance. After using WakeUpNet, information about his latest performance is displayed in the user interface. The system is also able to report statistical information about his performance along time, on demand.

The data gathered from each user, including his personal information, his work sessions' statistics and preferences is kept in a database managed by an external server. Although being confidential, the data model is prepared for further implementation of user groups (by age or gender, for example) to make reports, statistics and rankings of users into profiles, so that each user can compare his performance with the reference values of the majority. This would allow the system to recommend him the most suitable options according to his profile as well. To accomplish this, a technology standard for exchanging data over an algorithm for interpreting the sense data is needed as well.

3.4 Architecture overview

By definition, WakeUpNet is a context-aware system. Context-aware systems are defined by using context to provide user-relevant information and / or services, with user-dependent relevance, including end-user applications that use context information provided by sensors. Context is any information that is used to characterize a situation of an entity. In this case, the context is the brain activity that characterizes the user mental situation and enriches the user interaction with the application.

Context-aware systems can be local or distributed. Local systems are systems in which sensors and applications are tightly-coupled (usually through a direct physical connection). Distributed systems do not have a direct physical connection between the sensor and the application. As a consequence of that loose coupling, it is possible to have multiple applications receiving information from the same sensor. Also, it is possible that multiple dispersed sensors produce information to be consumed by a single application [8].

In the way WakeUpNet uses its sensors, it assumes a distributed system nature not having a direct physical connection with the end-user application and allowing each user to compare its own data with the other application users.

According to this, WakeUpNet is also inserted in the category of distributed non collaborative systems. Distributed collaborative systems are systems that help two or more dispersed humans accomplish a common goal, when non-collaborative systems support only individual goals [8]. Since WakeUpNet main goal is to monitor user's performance during study (or work) individually, its nature is non-collaborative.

WakeUpNet architecture design starts to follow the guidelines of the typical context-aware system. The simplest form of a context-aware system architecture is depicted in figure 3.2.

End-user application
Middleware
Sensor

Figure 3.2: Typical context-aware architecture

According to the figure, a context-aware system architecture includes three layers:

- **Sensors layer**: produces contextual information. Generic contextual information can be location, identity, activity, or time, corresponding roughly to the primal questions: where, who, what, and when.

- **Middleware layer**: establish the communication and coordinate issues between the distributed components. This layer is optional, because a simple context-aware system can be designed with a end-user application communicating directly with the sensor. However, in more complex systems, additional infrastructural components are desirable, namely: Decision Support Tools, Context Repositories and Context Processing components. The mentioned components reduce the system complexity, improve maintainability and promote reuse [8].

- **End-user applications layer**: consumes contextual information, i.e., uses the processed information to provide a service.

Considering the enumerated layers, WakeUpNet architecture was designed as follows: On its higher level, it is described by the User components and a Remote Server. The User components concern the system components directly accessed by each user, including a Sensor component and a User Device component.

WakeUpNet architecture fits in the typical context-aware system architecture as depicted is figure 3.3.

End-user application	User Device
Middleware	Remote Server, User Device
Sensor	Sensor

Figure 3.3: Framework of WakeUpNet architecture in the typical context-aware architecture

The Sensor layer is accomplished by the Sensor module, that integrates the brainwave sensors required to gather the user mental states. The end-user application layer is fulfilled by a mobile application running WakeUpNet system in a mobile device, what composes the User Device module. The middleware supporting this application will be integrated in the User Device module as well, supported by a Remote Server module.

3.4.1 Network architecture

Figure 3.4 depicts the main modules of WakeUpNet network architecture.

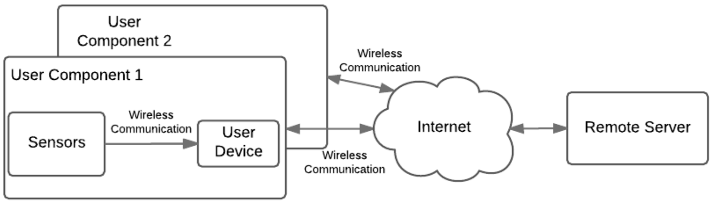


Figure 3.4: WakeUpNet distributed modules

As stated before, WakeUpNet comprises three main modules that are detailed as follows:

- **Sensors module:** Sensors are responsible for collecting the user brain activity raw data. To meet the defined requirements, physical sensors are required, i.e., hardware sensors capable of capturing sensing data. These physical sensors need to be supported by a portable device of common use, easy to configure and comfortable for using for a long period of time, being as less intrusive as possible. As was cited before, a set of headphones is recommend for this propose. The sensing data needs to be sent to the user device through a standard communication technology, easy to find in the available devices. This technology needs to support wireless low range and low bandwidth communications, since the system operation, running in the user device, will be executed close to the sensors.

- **User Device module:** The system operation needs to be performed through a mobile application running in a mobile device. The user device must be able to receive the data collected by the sensors, in raw format, and support its interpretation. WakeUpNet operation requires the interpretation of the sensing data to perform its operation accordingly, and the User Device Component needs to locally support processing algorithms for the collected raw data to assure the availability in case of disconnection and reduce the bandwidth. The remote server will only support some additional features that allow a larger latency and scalability, as will be discussed later. The User Device is also the component through which the user directly operates with WakeUpNet system, as it integrates the user interface.

- **Remote Server module:** The Remote Server consists of a scalable cloud system supporting a relational database with capacity to store all the data received from each User Component of WakeUpNet network. It also supports processing algorithms for data treatment, in order to generate the reports that may be requested by the User Components. The User Component and the Remote Server communicate through a communication standard that supports large range and full-scale networks operation. This communication must be wireless to assure the system scalability and mobility.

The User Component of figure 3.4 is composed by the Sensors and the User Device where the system will operate supported by the Remote Server. These three modules are described subsequently in the next section.

3.5 Modules architecture

An architecture solution for WakeUpNet implementation is proposed, as depicted in figure 3.5, minutely describing the modules identified in figure 3.4 and the most important information flows. Follows the details of each modules of the proposed architecture.

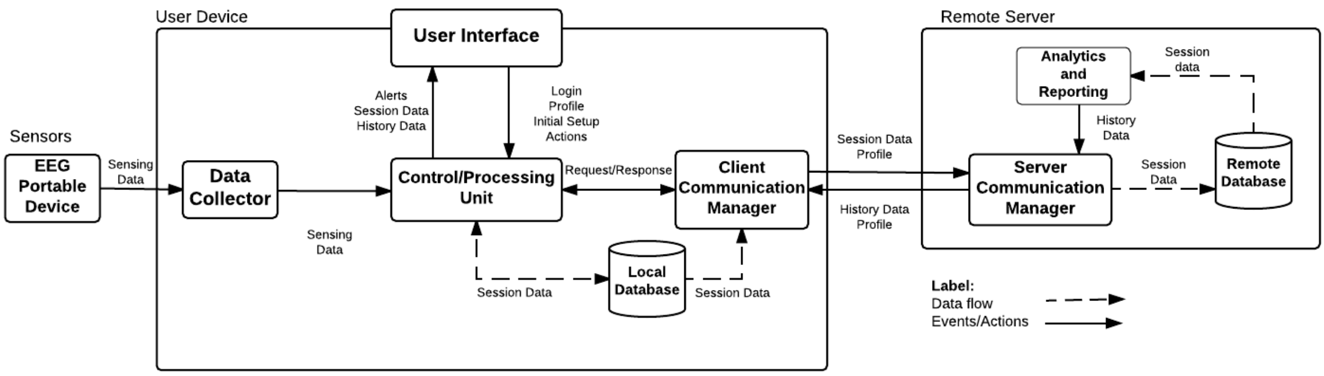


Figure 3.5: WakeUpNet Architecture

3.5.1 Sensors module

The Sensors module is responsible for measuring brain's electrical activity and transform it into a set of waves of different frequencies which may be processed to infer the mental state. This module must also be able to record information continuously and pair with a single User Device to send the data streaming through a persistent wireless communication channel. The data collecting and sending must occur in almost real-time.

As previously studied in section 2, two different type of sensing devices may be used to fulfill this goal: intrusive and non-intrusive. Non intrusive approaches do not reach the level of accuracy WakeUpNet proposes. This means that, the embedded sensors of a mobile device would not be an accurate solution; instead an intrusive solution, concerning external sensing equipment, is required.

Since the Sensors module will use an off-the-shelve equipment, from our architectural point of view it comprises a single module: the EEG portable device. The EEG device is an accurate technique for accurately recording the state of the human brain. As explained in section 2 it is considered a non-invasive technique in terms of not penetrating the human scalp, but in terms of user comfort it is really invasive, what raises some concerns to meet the Usability requirement. The solution will be to integrate the EEG sensors in a wearable common object the user would find acceptable to use, for example, a set of headphones through which he may listen to music during his work.

The sensing data collected from EEG is sent to User Device module in raw format, i.e., the percentage of each brainwave frequency detected every moment.

3.5.2 User Device module

The User Device module consists of a mobile device running an application designed to support WakeUpNet functionalities. The application interacts with the Sensors module, the Remote Server module and the end-user. Hence, at least at this first stage, it is the most complex module of WakeUpNet architecture.

Keeping in mind the modularity requirement, its functions were organized into five main components (see figure 3.5 User Interface, Data Collector, CPU, Client Communication Manager and Local Database. Information flows between them using well-defined interfaces, which are also represented in the same figure.

The next paragraphs details each one of these components.

3.5.2.A User Interface component

The User Interface component allows the direct interaction between the user and the system, which makes this component crucial to accomplish many of the requirements identified, as will be explained below.

The most effective form of User Interface (UI) to interact with a mobile device would be a graphical user interface integrated with a touch screen display. To meet the Usability requirement, this UI needs to be simple and intuitive. This means that:

- The interface elements (e.g. buttons, views, menus) should be easy to understand and operate in a consistent form.
- Error messages should explain how to recover from the error.
- Undo should be available for most actions.
- Actions which cannot be undone should ask for confirmation.

The conceptual model proposed for this UI, depicted in figure 3.6, defines it as a collection of screens linked sequentially, each one concerning one action the user can introduce in the system.

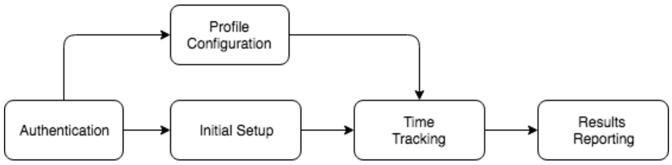


Figure 3.6: WakeUpNet UI conceptual model: operations

Functionalities There are five different types of user actions: Authentication, Profile Configuration, Initial Set up, Time tracking and Results Reporting. Each one of them is responsible for triggering a given operation, as follows:

- **Authentication operation:** is the first action every user performs, since the system needs to receive the user authentication before start operating. In this operation, the user inserts his credentials in an input field, namely the username and password. At the first utilization of WakeUpNet, the user needs to register his credentials in the system as well as configure its personal profile data.
- **Profile Configuration operation:** this operation is needed when a user wants to start using the system for the first time. Profile is an intermediate step between the Authentication and Initial Setup operations, which basically consists in a form the user must fill with his personal data fields, namely his age, gender and occupation (student or worker, and the field of studies or work). The user is only forced to take this at the first system utilization, but he can return here anytime he wants to change its profile data, e.g., in case of changing its occupation.
- **Initial Setup operation**: After successfully logging into the system, the user must fill a form describing his goals for the current session, namely: the duration of work time and break time periods he aims to accomplish. To make the usability requirement, the form to fill should have some default values so the user can use them as guideline in the first utilizations, for example, the values of 25 minutes of work and 5 minutes of break of the traditional Pomodoro technique. The user is asked to fill this information each time he enters the system to start a new session, since his goal may changes in different occasions. After the system starts running, he can not return here.

- **Time Tracking operation:** During the session time is the most relevant information that must be showed to the user. Hence, the Time Tracking operation displays the timer that keeps track of the duration of the spent work and break time periods, in terms of hours and minutes. This includes the Start and Stop buttons through the user can tell the system when he pretends to begin and finish his working sessions.

- **Results Reporting operation:** At the end of session users want to know about his performance. The Results Reporting displays all the relevant information about the user performance. By default, it displays the results of the working sessions that just ended. It also includes buttons so that the user may request more information, about his history through time or in terms of comparing his statistical results with the other system users.

  The history data about the user is represented by graphics of the user performance along several WakeUpNet utilizations, e.g., the last week. History data about the different WakeUpNet users is displayed by a graph showing how the user performance takes place in comparison with other users with similar profile characteristics.

3.5.2.B Data Collector component

The Data Collector is the first component in the User Device to deal with the sensing data collected by Sensors module. Its main operation is to gather the raw information, process it and send it to the CPU component. The sensors collect the EEG signal data in its raw format, i.e., a set of five waves with different frequencies that determine brain activity. Recalling section 2 those are alpha (α), beta (β), gamma (γ) and theta (θ).

The Data Collector applies a Feature Extraction method as basis for a classification algorithm which results in the categorizing of the mental states of focus and relaxation. To this categorization is applied a quantifying algorithm to determine if the processed signals indicate the user mental state [16]. This can be classified as focused, not focused, relaxed or not relaxed. The classified data is sent to CPU, whose operation depends on it.

Since the signals are collected continuously, the Data Collector needs to operate continuously as well.

3.5.2.C CPU component

The CPU component is the core of WakeUpNet functionalities. It is responsible for all the system decisions based on the defined operational algorithm and inputs received from the Data Collector, User Interface and Communication Manager.

Table 3.1 summarizes the functionalities supported by CPU and what components are also involved in them.

Operation	Involved Components
Authentication	UI, Client Communication Manager
Initial Set up	UI, Local Database
Profile Configuration	UI, Local Database
Time tracking	UI, Data Collector
Results reporting	UI, Local Database, Client Communication Manager
Database Synchronization	Client Communication Manager

Table 3.1: CPU supported functionalities and components involved

Functionalities CPU receives inputs from different components and based on them different operations are executed. The details of each one of the operations will be detailed next.

- **Authentication:** The first time the user performs Authentication in the system, an interaction with the Remote Server is required and, consequently, Internet connection is needed. This is a big restriction to WakeUpNet use since the user needs to be connected to the Internet to use it for the first time, but it Is the only way to guarantee a secure authentication process.

  What happens in the first Authentication operation, is the user is asked to registry his credentials: username and password. Those are sent to the Remote Server, that includes it, encrypted, in its remote database and sends back an authentication token, that is basically an UserID the server recognizes that user. The following times the user accesses the system he will not be asked to login again since this UserID is enough to recognize the user as itself.

  Note that although the interaction is between the CPU and the Remote Server component, they do not communicate directly. The Client Communication Manager is the one who mediates the CPU external communication, and that is why this module appears in the table 3.1 instead of the Remote Server.

- **Profile Configuration:** Whenever a new user signs up, he needs to define his personal information. Profile Configuration concerns a Local Database update with a new user and his personal data fields, through the process described in the subsection about the UI component. The Remote Server keeps the profile data as well, to assure the data replication and to make some computations necessary to report History results.

- **Initial Set Up:** This is an intermediate step required to perform the Time Tracking operation. Time Tracking is actually one of the most complex features managed by CPU. The timer displayed is a smarter version of the Pomodoro Technique, tracking the time the user spends either working either making a break, separately and cumulatively. To determine the beginning and the ending of work and break sessions, CPU supports the Initial Set Up to gather the user pre-defined Work and Break time duration. Then, it recurs to the sensing information about the user mental status to manage the timer. This operational algorithm is WakeUpNet main functionality and has a subsection dedicated to its details: subsection 3.2.

- **Time tracking:** During this operation, CPU keeps track of the following results information is generated:
  - Number of breaks - each time the user starts a break, CPU increments a counter of the performer breaks.
  - Total time and average time of breaks - CPU keeps track of the total time spent during breaks (between the work moments) and computes the average time spent in every break time.
  - Total time and average time of work - this process is similar to the previous one, but concerns the time spent during work phases (between break instants)
- **Results Reporting:** Concerning this operation, two different cases are defined: reporting of Session data and History data.
  - Reporting Session data concerns only the Local Database where all user data is stored and updated. By default, each WakeUpNet utilization ends with the displaying of the performance results of the current session using the attributes gathered during the Time Tracking execution, as stated above. Reporting session results also includes the displaying of the user evolution thought several sessions. In that case, CPU must query the Local Database to gather the results stored during, e.g., the last month. This data is treated in order to generate a graph displayed by the UI component.
  - Reporting History data, on the other hand, requires the access to data stored in the Remote Database. This database integrates the results information sent by every WakeUpNet user. The CPU can not query this database directly, so this operation involves the Client Communication Manager to manage the communication to the Remote Server component.
- **Database Synchronization** The Database Synchronization is a background task performed periodically to sync the Local and the Remote Database. It requires detecting if there is a network connection available to the Remote Server and query the Local Database in order to determine if there are unsynced data to send. If this two conditions verify, the Client Communication Manager is asked to manage the unsynced data sending.

3.5.2.D Local Database component

The Local Database stores the User Session data after it has been processed, every WakeUpNet run. There is no need to store all the raw data, as it would be a really big issue in terms of the storage capacity and usability, further in the mobile scope. Otherwise, only the data concerning the results generated in the Time Tracking operation are stored after each WakeUpNet run. Therefore, two different sets of information were considered as relevant:

- **User profile information:** it comprises the UserID and personal data, namely, the age, gender and occupation (worker or student). The passwords are stored encrypted, in order to make it safe.
- **User session information:** that includes the results obtained after each WakeUpNet running, as stated in the CPU module description.
- **Synchronization information:** it is needed, since the data stored locally needs to be periodically updated to the Remote Server, where all users sessions information is maintained to support the Results Reporting functionality. There is a need to keep sync tags that identify what data is or not updated in the Remote Server.

Next paragraphs enlighten the information models used.

Local Database model Figure 3.7 represents the Entity-Relationship Model of this Local Database.

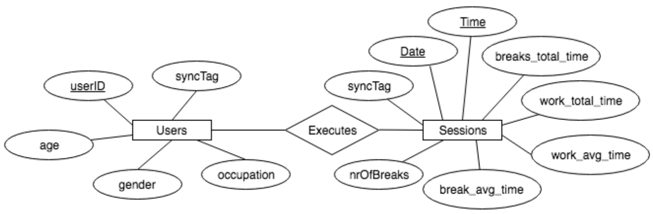


Figure 3.7: WakeUpNet Local Database Entity-Relationship Model

This conceptual model defines two entities - Users and Sessions - and a relationship between them - Executes, as follows:

- **Users entity:** it is defined by five attributes: UserID, Age, Gender, Occupation and syncTag. The UserID is the authentication token generated by the Remote Server when a user registers for the first time. The user does not know what is its UserID, this is an abstraction just to trace the user data in the system safely, because it is not required the username and password to circulate through the communication channels. Also, it would not be safe to store the used credentials in a mobile device since the most of information is shared with another applications. Except for the UserID and syncTag, the other attributes are chosen when inserted by the user itself in its first login.
- **Sessions entity:** it is defined by eight attributes: syncTag, Time, Date, Number of Breaks, Work Total Time, Work Average Time, Breaks Total Time and Breaks Average Time. Except for the syncTag, these attributes are generated by the CPU during system run, as in the Time tracking

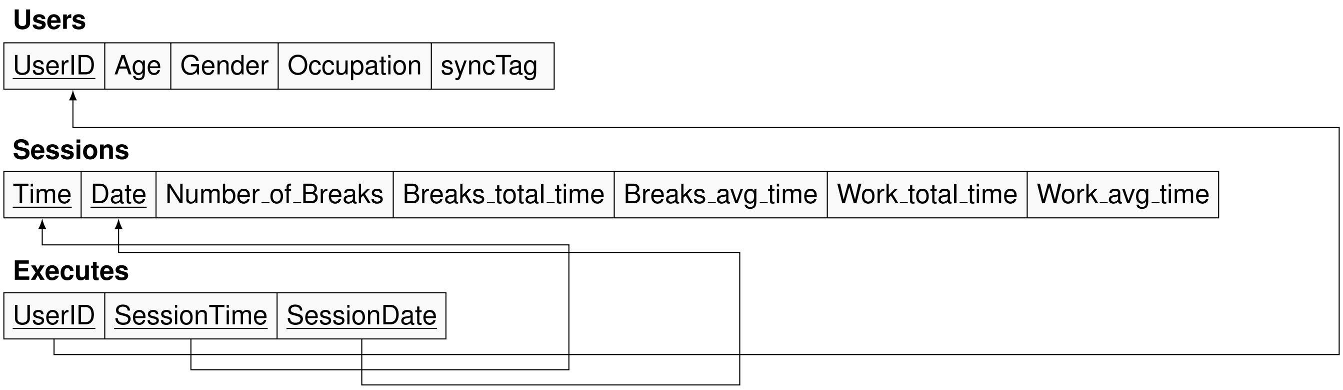


Figure 3.8: WakeUpNet Local Database Relational Model

operation previously explained. Time and Date are the identifier of the session. These attributes allow keeping track of the order by the sessions were executed, which is important to trace the graphs of the users performance along time.

In both entities there is a common attribute: The syncTag. This is a boolean type that defines if the user profile or the sessions results information are or not updated in the server. That action corresponds to the

- **Executes relationship:** is a m:n relationship, also called many-to-many relationship. This means that many users execute many sessions.

Local Database tables The described Entity-Relationship model is translated to the Relational Model figured in figure 3.8. This relational model is the guide to the database schema.

According to this schema, three tables are required: Users, Sessions and Executes.

- **Users:** This table has five columns, one for each Users entity attribute. All the columns entries are filled with String types, except the Age column that accepts the Integer type and for the syncTag column that accepts the boolean type. The UserID is the primary key of this table, i.e., it is the entry that identifies each record in the Users table. When the user registers in the system, a new entry in this table is added with the attributes selected by the User in the Profile component of the UI. Everytime the user updates one of this fields, this table is updated as well.

- **Sessions:** This table has eight columns, accepting entries of the Integer type, except for the Time column that accepts the SQL Time type, the Date column that accepts SQL Date type and the syncTag that accepts the boolean type. Time and Date compose the primary key. It is necessary to have a composite primary key, i.e., a primary key of two or more columns from the table to discriminate sessions executed in the same day.

  Everytime the user finishes one WakeUpNet session, a new entry in this table is added with the results data generated during the Time Tracking operation of CPU component.

- **Executes:** This table contemplates columns with UserID and the respective Time and Date. This is made by a referential integrity constraint, which means ff the data of Users and Sessions table is amended, Executes shall be verified to keep the data consistent. The entries in Executes are then Foreign Keys, i.e., are entries extracted from other tables.

3.5.2.E Client Communication Manager component

The Client Communication Manager takes part of the Communication Manager, a service that allows the sharing of information between the user device and the Remote Server by CPU command. This communication is specified in next subsection that concerns the integration between the components.

3.5.3 Remote Server module

As described before, there is a need to store all the data from all the system users as well as guarantee a local database backup, the user access to his data from different devices, and a safe method of authentication.

Mobile devices have limited storage and processing capacities as well as particular security issues. Besides, the local storing only assures the system would run in one single device. So, this can only be accomplished maintaining a remote database database properly synced with the local database. A remote database allows the local database having only the necessary data each moment, having additional data required available under request. WakeUpNet architecture solves this through the Remote Server component, composed by the Server Communication Manager, the Remote Database, and the Analytics and Reporting, as depicted in figure 3.5.

The next subsections details each one of the components.

3.5.3.A Server Communication Manager component

The Server Communication Manager takes part of the Communication manager, the service that allows the sharing of information between the User Device and the Remote Server components.

The Remote Server is independent from the User Device. To make the communication between the two components through the Internet possible, it acts as a web service. A proper API to abstract the communication to the database is proposed. This API is integrated through the Server Communication Manager that is responsible for the server-side of the communication. This component is able to interpret the requests processed by the Client Communication Manager, create the matching responses and send it back to the User Device.

The information sent back is the History Data, information the system reports the user, about all the other users in the system, for comparison.

This communication process can't be isolated from the Client Communication Manager operation, so it is detailed in subsection 3.6, dedicated to the interactions between the User Device and the Remote Server.

3.5.3.B Remote Database component

The Remote Database is fulfilled with the content of each user local databases, including:

- **User profile information:** The user credentials, namely username and password, and its personal data, namely, its age, gender and occupation (worker or student). The passwords are stored encrypted, in order to make it safe.
- **User session information:** The results obtained after each WakeUpNet running, as stated in the CPU component description.

The next paragraphs details the database model and associated tables.

Remote Database model This information is defined by an entity-relationship model, as shown in figure 3.9.

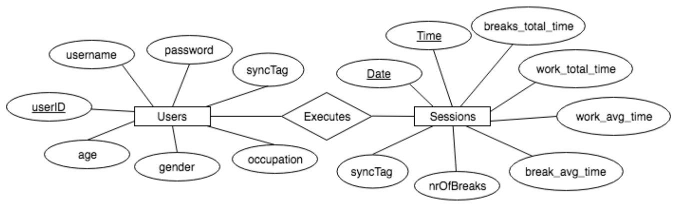


Figure 3.9: WakeUpNet Database Entity-Relationship Model

Similar to the conceptual model of the local database, figured in 3.7, the Remote Database conceptual model defines two entities - Users and Sessions - and a relationship between them - Executes, as follows:

- **Users entity:** in this case, the Users entity is defined by six attributes: UserID, Username, Password, Age, Gender and Occupation. As stated before, the UserID is the authentication token, associated with the user credentials, generated by the Remote Server when a user registers for the first time. Except for the UserID, the other attributes are received from the User Device at the first Authentication operation performed by the CPU. The username and password attributes are then just stored in the remote server, after an encryption method, with a service that allows the decryption key to be stored anywhere else separately.
- **Sessions entity:** it is defined by only seven attributes: Time, Date, Number of Breaks, Work Total Time, Work Average Time, Breaks Total Time and Breaks Average Time. This attributes are

received from the User Device as well, anytime a Database Synchronization operation is made (as will be discussed later).

- **Executes relationship:** it is a relationship identical to the one in the local database so we don't need to repeat.

There's no need to keep synced tags in the remote database, only in the local, as will be explained in subsection 3.6.2.

Remote Database tables The relational model that guides the database schema is depicted in figure 3.10.

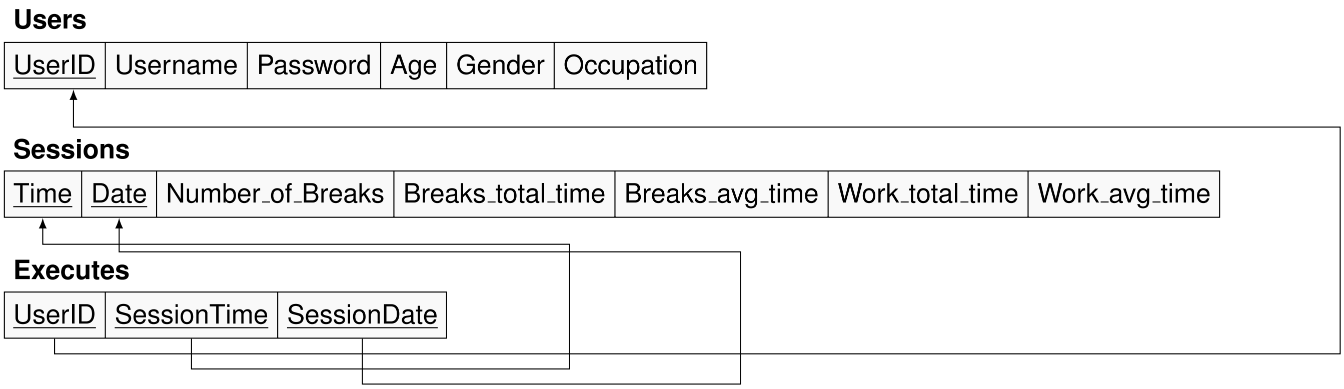


Figure 3.10: WakeUpNet Database Relational Model

According to this schema, three tables are required: Users, Sessions and Executes. Their contents are the following:

- **Users:** the table has six columns, one for each Users entity attribute. UserID, Age, Gender and Occupation have the same type as in the local database. UserID is also the primary key. This database additionally includes entries for the password and username fields.

- **Sessions:** the table has seven columns, being similar to the remote database Sessions table but without the syncTag field. This table is fulfilled as well by the Database Synchronization process.

- **Executes:** the table is similar to the one in the local database.

3.5.3.C Analytics and Reporting

The Analytics and Reporting component allows a dynamic integration of the database, i.e., the query processing during execution by requested from the User Device, and supports Results Reporting and Database Synchronization functionalities. The operational algorithm that makes this possible is detailed in section 3.6.

3.6 Remote Server and User Device interactions

The Mobile Device module and the Central Server module communicate through the Client Communication Manager component and the Server Communication Manager component, respectively, via Wi-Fi or Global System for Mobile Communications (GSM), trough a trustful communication protocol based in TCP, in a typical client-server interaction.

The two main interactions are about retrieving History results to the user, regarding the results of the other users in the system; and the synchronization of the remote and local databases. This functionalities are then only available with Internet connection.

As will be discussed, the Remote Server only responds to User Device requests, does not send requests to it.

To assure the user has access to the most recent results, the operation of Database Synchronization is always performed before the History Reporting.

3.6.1 History Reporting

Under user order, the Mobile Device may send requests for History reports, relative to other users individual progress, keeping each users data anonymous but traceable.

The user needs to select the type of results he pretends to obtain. The options are as follows:

1. **Number of breaks:** e.g., the user makes in average three breaks per session, and pretends to know if that's less or more than expected.

2. **Duration of continuous work without breaks :**e.g., the user works in average twenty minutes until he takes a break, and pretends to know if that's less or more than expected.

3. **Duration of breaks :** e.g. the user takes breaks of 10 minutes in average and pretends to know if that's more or less than expected.

4. **Percentage of breaks time per work time:** e.g. the user spends, in average, 20 minutes taking a break for every 60 minutes of work and pretends to know if that's more or less than expected.

As stated in the description of the UI component, the user must select what is the type of History data he wants to see. This selection is important to relief the server operation, since he only has to query and retrieve the selected data.

The Client Communication Manager module converts the option selected to an unique numeric identifier, as stated in table 4.1. The user can request for more than one option at once, so the Client Communication Manager creates an integer array of the options selected. The Server Communication Manager module contains the same table to interpret the information solicited.

Selected Option	Identifier
Option 1	0
Option 2	1
Option 3	2
Option 4	3

Figure 3.11: Communication Manager interpretation of the user results selection

The user also needs to select a date range, with the following options:

1. **Results of the last six months:** the user pretends to consult it's evolution during the last six months in comparison with the results of the last six months obtained by another users with a similar profile;

2. **Results of the last month:** the user pretends to consult it's evolution during the last thirty days in comparison with the results of the last month obtained by other users with a similar profile;

3. **Results of the last week:** the user pretends to consult it's evolution during the last seven days in comparison with the results of the last week obtained by other users with a similar profile.

This options are interpreted for the client managers components the same way as already stated.

The system only gives the user date selection options according to the information kept in the local database, since the user ca not access results of the last month if he only uses WakeUpNet one week ago. Also, it is important to notice the option 4 refers to the results of the last session in comparison with the average

The Analytics and Reporting component establishes the connection to the database, collects the requested information from it, and sends the response to the Mobile Device, properly encapsulated and encrypted for the Server Communication Manager. Since the query to the database must retrieve the data about the users with the same profile characteristics, the request from User Device needs to include the user identification to determine in which profile he fits in.

Figure3.12 summarizes the messages exchanged in this communication action.

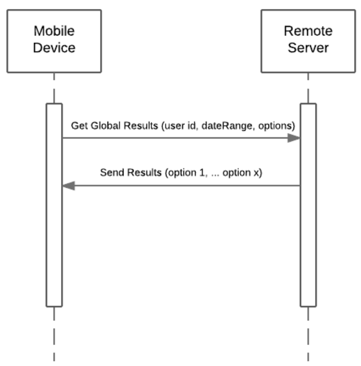


Figure 3.12: Messages exchanged between User Device and Remote Server to access the remote database

Analytics and Reporting algorithm In order to understand how the Analytics and Reporting operates let us remind the general WakeUpNet operation. In the beginning, the Analytics and Reporting starts to select the profile characteristics (age, gender and occupation) matching the UserID received. Then, it selects all the users that have the same profile fields. Finally, selects the sessions performed by those users, in the date range received. This selection depends on the options received in the Request. After gathering the results of all users with the desired profile and in the defined data range, it computes the average of the results, and the Server Communication Manager sends this number back to the User. In the case the user has selected more than one option of results, the Remote Server needs to send an average number for every option.

Algorithm 3.1 details, in pseudo-code, the Analytics and Reporting operations in the Remote Server to provide the results requested.

3.6.2 Database Syncronization Process

The Remote Server database is kept up to date because the User Device sends the unsynced session and profile information to the Server in a pull-based fashion.

The first time the user inserts its profile data in the system, or every time he changes it, the syncTag of Users entity in the Local Database is defined as 0. The same way, every time the user performs an WakeUpNet session, generates a new entry in the Sessions table, with the syncTag attribute is defined as 0.

Periodically, the CPU queries the local database to collect the data tagged as unsynced. If that is the case, he infers the Internet connection availability to send the unsynced data to the remote database. If there is no Internet connection, he keeps periodically inferring its availability to send the unsynced data as soon as possible.

Figure 3.13 summarizes the messages exchanged in this communication action.

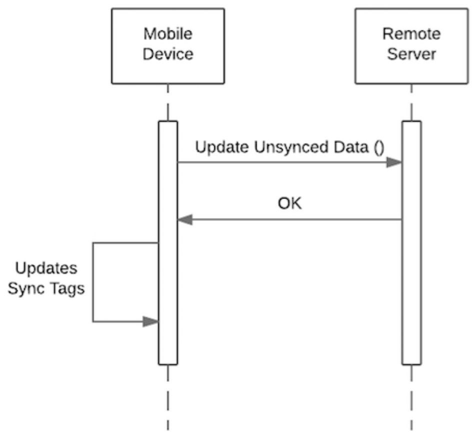


Figure 3.13: Messages exchanged between User Device and Remote Server to synchronize databases

Algorithm 3.1: Analytics and Reporting Algorithm

Int @userID $\longleftarrow userID$;
Date @dateRange $\longleftarrow dateRange$;
Array Integer @Options $\longleftarrow options$;

Queries database to: select profile attributes (age, gender, occupation) of the user registered with the received UserID;
Int @age $\longleftarrow age$;
String @gender $\longleftarrow gender$;
String @occupation $\longleftarrow occupation$;

Queries database to: select all users registered with the profile attributes selected;

Array Integer @UserIDs $\longleftarrow userID$\\Array with the UserId of the selected users;
Array Integer @Results_To_Send\\Array with the Results to send

Switch (option: @Options):

Case 1: **For each** UserID **in** Array @UserIDs: {
Queries database to: select attribute Number_Of_Breaks from all the sessions performed in the selected data range;
Array Integer @Results1 $\longleftarrow$ Number_Of_Breaks} ;
@Results_To_Send $\longleftarrow$ average of values in Array Integer @Results1 ;
Break ;

Case 2: **For each** UserID **in** Array @UserIDs:{
Queries database to: select attribute Work_avg_Time from all the sessions performed in the selected data range;
Array Integer @Results2 $\longleftarrow$ Work_avg_Time};
@Results_To_Send $\longleftarrow$ average of values in Array Integer @Results2 ;
Break ;

Case 3: **For each** UserID **in** Array @UserIDs:{
Queries database to: select attribute Break_avg_Time from all the sessions performed in the selected data range;
Array Integer @Results3 $\longleftarrow$ Break_avg_Time};
@Results_To_Send $\longleftarrow$ average of values in Array Integer @Results3 ;
Break ;

Case 4: **For each** UserID **in** Array @UserIDs:{
Queries database to: select attribute Break_Total_Time from all the sessions performed in the selected data range;

Queries database to: select attribute Work_Total_Time from all the sessions performed in the selected data range;
Array Integer @Results4 $\longleftarrow$ Break_Total_Time\Work_Total_Time};
@Results_To_Send $\longleftarrow$ average of values in Array Integer @Results4 ;
Break ;

Send **@Results_To_Send** to User Device;

3.7 CPU operation - finite state machine

As stated before, the CPU is the core component of WakeUpNet system, being responsible for all system decisions. Its specification is depicted as a finite state machine, shown in figure 3.14.

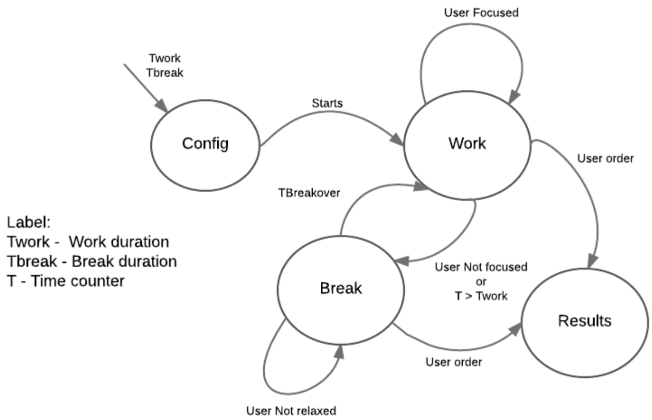


Figure 3.14: State diagram of WakeUpNet functionality

This finite state machine include four states: Config, Work, Break and Results. Each transition to each one of them is trigered by a given event, as follows:

- **Config status:** concerns the initial setup required to launch the system operation. This status is only and always accessed in the beginning of the system run and the user ca not turn back to it again.
- **Work status:** concerns the monitoring of the user mental activity during work.
- **Break status:** deals with the user mental activity during break.
- **Results status:** is the final status, only accessed in the end of the system run. It is the status where the user can see the Results of the just ended session and ask for History results if he wants.

CPU operational algorithm This section details the algorithm executed by the CPU in the FSM previously described. The CPU operational algorithm is represented in the pseudo-code shown in 3.2.

When the system is launched it starts in Config status. The UI receives the user inputs concerning the duration of the work sets he pretends to execute, for example, the typical 25 minutes of the Pomodoro technique; as well as the duration of the breaks, for example, the typical 5 minutes of the Pomodoro technique. In 3.2 the duration of the working time is represented by the constant Twork and breaking time by Tbreak.

Algorithm 3.2: Proposed CPU operational algorithm

$Boolean$ user_is_focused = true;
$Boolean$ user_is_relaxed = true;

Config():

TWork $\longleftarrow$ $userinput$;
TBreak $\longleftarrow$ $userinput$;

Work status routine():

while T $<$ Twork **do**
    Attention analysis();
    if user_is_focused = false **then**
        Break status routine();
Break status routine();

Break status routine():

if T $=$ Tbreak **then**
    Relaxation analysis();
if user_is_relaxed = true **then**
    Work status routine();
else
    Break status routine();

Once configured, the system does not turn back to this state. This means the user only configures the system one time per utilization. After leaving the Config status, the systems alternates between the states of Work and Break, until the user command to end the session and reach Results status.

Leaving Config status always follows the Work status. CPU communicates with the UI to start the timer. The real time is represented by the variable T in 3.2 and it's one of the main factors that influence CPU operation. This variable is incremented while time goes by and it is constantly compared to the Twork to determine if the user already worked during the period of time he pretends to. At the same time, the CPU reads the result of the brainwaves data streaming processing, received from the Data Collector, relative to the user Attention status.

The Break status proceeds Work status in two different cases:

- The real time equals the input Work time
- The user is not focused

This means that if CPU receives information of a low level of concentration and the timer has not reached yet the working time defined by the user in the initial setup, the system jumps to the Break status. Otherwise, the system only reaches the Break status when the timer equals the work time defined by the user.

Break status procedures are different from Work status because breaks are typically smaller than the work duration. The default procedure is the Break status take the duration inputed in Config status - indetified as Tbreak in 3.2. However, a parallel analysis to the data streaming influence CPU decisions as well. The difference is this analysis only occurs after the break time is totally spent. At that moment, CPU receives information about the user Relaxation status. This information defines if the user has rested enough, or not, during his break. If not, the Break status procedure is repeated. Otherwise, the system must return to the Work status. This means the Work status only proceeds Break status in one case:

- The real time equals the input Break time and the user is relaxed

The system keeps alternating between Work and Break status until the user commands it to stop. That can happen in any state, and when happens, the system evolves to the Results status.

Sensing Data analysis The functions ”Attention analysis()” and ”Relaxation analysis()” of the pseudo-code 3.2 are the more complex operations of the system and need to be explained in detail. These functions concern how the sensing streaming analysis is done to determine the user mind status. The respective algorithms are depicted in algorithm 3.3.

The sensing data is captured by the sensors continuously. This means, a set of five brainwave frequencies is captured every second. The Data Collector receives this data in a buffer, identifying the begin and end of each set. To each data set is applied a quantification algorithm, to determine a value of concentration and relaxation on a scale from 0 to 100. These values are inserted in a different buffer that is sent to the CPU to be interpreted.

The CPU needs to interpret the level of attention and relaxation of the user; the first, during work/study, and the second, during breaks. In the scale of 0 to 100, the values below 50 represent lows values of attention and relaxation, respectively. 50 is then the threshold below which the user is considered out of focus or not relaxed. However, the values fluctuate in time, so it is more accurate to only consider the user is out of focus or not relaxed if he generates an average of values below 50 after a certain period of time. We considered that five minutes is a reasonable time to determine the user mind status, i.e., the followed rule is ”If the user generates an average of values is inferior to 50 for attention/relaxation during the last five minutes he is considered not focused/not relaxed”.

To accurately identify the user mind status, it is then necessary to continuously determine his attention/relaxation values in the last five minutes. This is achieved through two circular buffers, one for attention, other for relaxation. This buffer has the size of 300. This value is the value of seconds in five minutes.

The objective of this analysis is make the system evolve according do the user needs. If, in the last five minutes of work the user is considered out of focus, he must start a break; or if, in the last five

Algorithm 3.3: CPU algorithm for sensing data analysis

```
Boolean user_is_focused = true;
Boolean user_is_relaxed = true;
Integer threshold_constant = 50 ;
Integer first_analysis_constant = 300 ;
Integer following_analysis = 60 ;

Attention analysis():;
while attention_buffer lenght < first_analysis_constant do
    addFirstElement(attention_buffer) ⟵ streamingdata;
compute average;
if average < threshold_constant then
    user_is_focused = false;
else
    counter = following_analysis while counter! = 0 do
        removeLastElement(attention_buffer) ;
        addFirstElement(attention_buffer) ⟵ streamingdata;
        counter − 1

Relaxation analysis(): while relaxation_buffer lenght < first_analysis_constant do
    addFirstElement(relaxation_buffer) ⟵ streamingdata;
compute average;
if average < threshold_constant then
    user_is_relaxed = true;
else
    counter = following_analysis while counter! = 0 do
        removeLastElement(relaxation_buffer) ;
        addFirstElement(relaxation_buffer) ⟵ streamingdata;
        counter − 1
```

minutes of break the user is considered not relaxed, he must extend his break a few more minutes.

So, the first analysis is made after the first five minutes of work, for attention measure, or after the last five minutes of break, for relaxation measure. To make a continuous analysis, the respective buffer must delete the oldest values and insert the new ones, captured every second. This guarantees the analysis is always being made only to values captured in the latest 300 seconds.

To minimize the CPU effort for this operation, the forwarding analysis just need to be made every other minute. This means, after the first analysis, made with the first 300 values collected, the following analysis are made over new 60 values.

3.8 Synthesis and discussion

It is pretended that WakeUpNet system would be an innovative system that incorporates time management with the user's natural work rhythm through brain activity sensing. Additionally, it must be able to report the user his performance results over time as well as in comparison with other users performance results.

Keeping this overall goals in mind, the authors identified the system functional and not functional requirements that should be took into account for the architecture design. This resulted in an architecture composed by three main components:

Sensors: A set of headphones integrates EEG sensors to read user's brainwaves, without intrusiveness, inferring his states of mind, mainly the state of focus and the state of relaxation.

User Device: A mobile device runs a dedicated application that supports WakeUpNet functionalities. This application comprises the User Interface, the Data Collector, the CPU, the Local Database and the Client Communication Manager.

The User Interface allows the interaction between the user and the system. It is responsible for receiving the user inputs, such as his authentication, the initial configuration, the profile information and commands for actions. It is responsible as well for displaying the alerts and results during the application running.

The Data Collector gathers the information received from the sensors, processes it into a classification of mental states (focused/not focused or relaxed/not relaxed) and sends this processed data to the CPU.

The CPU makes all the system decisions based on the defined operational algorithm and input received from the Data Collector or User Interface. As this is the core component of the system, supported by complex algorithms that must be carefully took in account.

The Local Database stores individual information about the user results through the several WakeUpNet uses.

Client Communication Manager allows the sharing of information between the user device and the Remote Server.

Remote Server: This component is composed by the Server Communication Manager and a Remote Database. The Remote Database stores processed data about all the users of the application that may be requested for the user in order to monitor his performance in comparison to the others. The Server Communication Manager receives the requests to consult the Remote Database, sent from the Client Communication Manager and sends back the History data required.

4

Implementation

As stated in the previous section, WakeUpNet was designed it its full potential, to commercial proposes. However, this project was implemented as a final dissertation project, so as expected, many implementation decisions need to took in account goals with particular nature in terms of technologies available and submission date restrictions.

In this chapter are discussed the decisions for tools and technologies used in the implementation, in section 4.1.

Follows the details of the implementation process concerning every component and algorithms of the proposed architecture: first, concerning the sensors component, in section 4.2.1; then, the user device component, in section 4.2.2 and finally, the Remote Server, in section 4.2.3.

After this, the more important and complex algorithms performed by the CPU, i.e, the algorithm for sensing data analysis and the operational algorithm are detailed in section 4.3.

Finally, the limitations and restrictions faced during the implementation are stated in section 4.4.

4.1 Options

The first implementation decision concerned the Sensor component. According to the proposed architecture this should be a portable and non intrusive EEG device.

Through the research shown in section 2.2, Neurosky's Mindwave Mobile device ended to be the best option for the implementation, as it was an accurate brainwave sensor compatible with mobile devices through Bluetooth.

The authors of this project took the decision of using only Mindwave Mobile device sensors to get brain activity information, using the device in its commercial form. According to the proposed architecture, the device sensors should be extracted and integrated in a set of headphones, to meet the Usability requirement. That option was discarded since that process would be too much time and effort consuming, when the ending results about the sensing reading would be exactly the same. However, it is important to note that if this system would be implemented for commercial proposes, this step would be definitely important to reduce its intrusiveness.

Another reason for using Mindwave Mobile was the tool set for developers provided by Neurosky. As stated in section 2, Neurosky provides a toolkit compatible with both Android and iOS, the two biggest operative systems for mobile environments. This tool set is named NeuroSky MDT and made it easy to use the BCI technology of NeuroSky's MindWave headset. This allowed to simplify the sensing data processing, as it already provided an algorithm that characterizes the raw data into a quantization scale for many brain status.

The second option was to implement the mobile application that supports the system in the User Device component in Android OS, because not only is the most common mobile OS used but also because it can be easily developed through the Android Studio Integrated Development Environment (IDE)[1], which is free for download, and offers an intelligent code editor for a java language, a language with the authors were already familiar with; and an emulator with instant run. So for this case, between the many developer toolkits provided by Neurosky, the Android Developer Tools 4.2[2] was used.

Another important implementation decision concerns the displaying of Results. When planning the Implementation and Testing, the authors concluded the test phases would require an unsustainable number of hours to complete, for the reasons that will be discussed in the 4.4 section. The only sustainable tests concern the results of the just ended session of every user, so that was the main focus of the implementation. This decision affected the implementation of the databases as well. In these case, there was no need to dedicate time and effort to implement a remote and a local database, properly synced. The authors decided to implement only the Remote Database, able to record several sessions of every user, although only the record of one session was used for tests. This decision was made to assure the Scalability and Extensibility requirements, as only implementing the Local Database wouldn't meet them.

The remote server was set recurring to the MySql database server provided for Instituto Superior Técnico (IST), since it's accessible for free, easy to use and the authors were already familiarized with it. Consequentially, the remote database was implemented as a simple relational MySQL database.

To support WakeUpNet mobile application, as stated in section 3.5, a mobile device was used. The authors opted for a One Plus 3T smartphone was used, since it was the equipment we had immediately available and fulfilled all the User Device requirements.

[1]Android Studio: https://developer.android.com/studio/index.html, last access on September 2017
[2]Neurosky Store: https://store.neurosky.com/products/android-developer-tools-4, last access on October 2017

4.2 Modules implementation

4.2.1 Sensors module

As already stated in the previous section, the Sensors module was implemented recurring to the Neurosky's Mindwave Mobile in its commercial form, i.e., an headband able to pair with mobile devices through Bluetooth technology.

Regarding the proposed architecture, this component should be able to send Sensing Data to the User Device. This was implemented recurring to the Android Developer Tools 4.2.

This kit included Application Standards Document and Icon Images, Stream Software Development Kit (SDK) for Android and EEG Algorithm SDK for Android and EULA (End User License Agreement). Each SDK includes a sample project and documents which teach the developer how to use them. Stream SDK for Android contains everything is needed to connect the Android application to the Mindwave Mobile headset via bluetooth, and receive the sensing data from it. EEG Algorithm SDK for Android is used to analyze and further interpret EEG data from NeuroSky's headset or TGAM module. It also includes free algorithms for the quantizing several mental status, being the Attention and the Meditation algorithms used in this case.

This kit also allowed to simplify the Data Collector as will be explained in the next section.

4.2.2 User Device module

WakeUpNet mobile application is one of the key components of the system architecture, as seen in figure 3.5. This section will explain how every component works in the software scope.

4.2.2.A User Interface component

To help the understanding of the User Interface, figure 4.1 shows WakeUpNet application front-end, including its five activities - Authentication, Profile Configuration, Initial Setup, Time Tracking and Results Reporting - each activity one corresponding to a screen in the application.

Functionalities These activities are responsible for:

- **Authentication Activity:** is responsible for the authentication for registered users, containing a form for introducing its credentials - username and password.
- **Profile Configuration Activity:** the first time the user uses WakeUpNet, he must be directed to the Profile Configuration Activity that contains a form to insert its credentials - username and password - as well as its personal data fields - age, gender and occupation. Next time he visits the

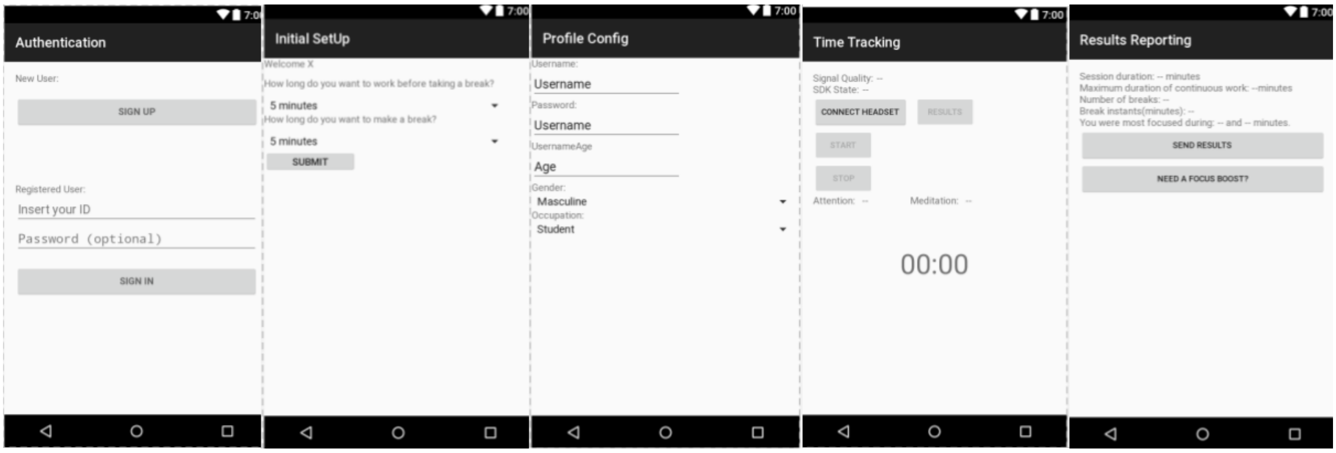


Figure 4.1: WakeUpNet mobile application front-end

application, he won't need to insert its password. As mentioned in subsection 4.1 this steps are only implemented in the front-end scope, since there was no possibility to test it.

- **Initial Setup Activity:** is where the user inserts the initial setup, concerning the period of time during which he pretends to work and to make a break. This time periods are pre-defined to be close to the values of the traditional Pomodoro Technique that concerns 25 minutes of work and 5 minutes of break: For working time, the user may chose to work during 15, 25, 35, 45 or 60 minutes; and for break time, the user may chose to pause during 5, 10 or 15 minutes.

- **Time Tracking activity:** concerns the Time Tracking screen identified in the proposed architecture. The User Interface of this activity displays four Buttons - Connect Headset, Start, Stop and Results - the signal quality status and a timer.

- **Results Reporting Activity:** displays the information about what happened during the session. Recall that "Session" is the name of a full access of WakeUpNet system, including work and break periods.

  As seen in image 4.1, WakeUpNet shows five different types of information:

  – **Session Duration:** is the total duration of the session, including break and work time periods.

  – **Maximum number of continuous work:** is the largest period time in Work status.

  – **Number of breaks:** is the total number of breaks made during the session.

  – **Break Instants:** instants when breaks occurred.

  – **You were must focused during x and y minutes:** informs about the time period during which the user were most focused

This result fields are not exactly the same as planned in the proposed architecture because, after the decision of only displaying the results of every user latest session, there was need to provide the most useful results for individual use.

Results Reporting activity also includes the Send Results button in case the user pretends to send the displayed information and the sensing data captured for the remote database. This option allows the creation of graphs showing user progress over time, as well as the comparison with other users with similar profiles. However, the user can not to be comfortable with the idea of having personal brain activity information stored in a remote server, so results data is only sent under his order.

Finally, the button with label ”Need a focus boost?” is a primary idea for supporting the functionality of suggestions for performance improving. In the presence of Internet connection, this button should redirect the user for a neurofeedback game that can help him, trough time, to train their brain operation, similar to what was studied in section 2.4. However, as seen in that projects, the construction of a neurofeedback game would extent the scope of this project. In alternative, the button redirects the user to a page in Drive Store where he can download neurofeedback games.

4.2.2.B Data Collector component

As mentioned earlier in this section, this component was implemented recurring to the Android Developer Tools 4.2. This toolkit simplified a lot this component implementation since it already included quantization algorithms for the mental states: the eSense algorithms. For WakeUpNet were used two, as follows:

- **eSense Attention meter:** indicates the intensity of a user's level of mental "focus" or "concentration", such as that which occurs during intense concentration and directed (but stable) mental activity. Distractions, wandering thoughts, lack of focus, or anxiety may lower the Attention meter level.
- **eSense Meditation meter:** indicates the level of a user's mental "calmness" or "relaxation". Meditation is related to reduced activity by the active mental processes in the brain. Distractions, wandering thoughts, anxiety, agitation, and sensory stimuli may lower the Meditation meter levels.

For each different type of eSense (i.e. Attention, Meditation), the meter value is reported on a relative eSense scale of 1 to 100 outputting every second. Neurosky considers five categories of eSense interval values, shown in table 4.1.

On this scale, a value between 1 to 20 indicates "strongly lowered" levels of the eSense and a value between 20 to 40 indicates "reduced" levels of the eSense. These levels may indicate states of distraction, agitation, or abnormality.

Table 4.1: Categorization of eSense scale intervals

1-20	||	Strongly lowered
20-40	||	Reduced
40-60	||	Neutral
60-80	||	Slightly elevated
80-100	||	Elevated

Similarly, on the other end of the scale, a value from 60 to 80 is considered “slightly elevated”, and may be interpreted as levels tending to be higher than normal (levels of Attention or Meditation that may be higher than normal for a given person). Values from 80 to 100 are considered “elevated”, meaning they are strongly indicative of heightened levels of that eSense [15].

The CPU component also has a similar table to interpret the data received.

The Data Collector continuously captures eSense values for both algorithms every second.

4.2.2.C CPU component

The implementation of CPU functionalities differs from the planned in the proposed architecture as depicted in table 4.2. Basically, all the operations are the same, but omitting the existence of a Local Database.

Operation	Involved Components
Authentication	UI, Client Communication Manager
Initial Set up	UI
Profile Configuration	UI, Client Communication Manager
Time tracking	UI, Data Collector
Results reporting	UI, Client Communication Manager

Table 4.2: CPU functionalities implemented and components involved

Every activity identified before in UI implementation details has a CPU operation behind it.

Functionalities As stated before, the CPU performs different operation, depending on the inputs received from the external components. No modifications have been made to the FSM, but since there is no local database, a few details need to be explained to understand how it works. They will be explained next:

- **Authentication:** the first operation in every WakeUpNet session is authentication, during the Authentication activity. If the user already is registered in the application, CPU creates a new user object, whose attributes are gathered from the Remote Database. For every Session stored in the Remote Database matching the user identification, a new Session object is created with the stored attributes. All this sessions are inserted in a new ArrayList of Session objects, that is associated as the User object' Sessions attribute.

As the authentication functionality implementation does not add any innovative characteristic to this project, we would not deepen about it.

- **Profile configuration:** if it is the first time the user uses WakeUpNet, as stated before, he is forwarded to the Profile Configuration activity. CPU creates a new User object as well, however, its attributes are defined according to the user inputs in the fields previously identified in UI implementation details for this activity. In that case, it is the first Session the User performs. That means, the attribute Sessions is periodically filled as null, until the first Session is completed. However, a new Session object, for the current WakeUpNet run, is already created, with every attribute set to zero.

- **Initial Setup:** takes place after Authentication. As described in the details of UI component, this takes place in the Initial SetUp activity, where the user must choose, between the given options, the time of work and break during he pretends to work and take a break, respectively.

  Additionally, a new Session object is created as well, corresponding to the new Session the user is about to perform as soon as he enters the Time Tracking activity. The sessionID attribute is immediately set with the value of the user's Sessions ArrayList length plus one. The other attributes are temporally set with value zero, until they can be updated with the session results.

  In the regular use of the application, Initial SetUp activity is followed by Time Tracking activity, to where the chosen values are sent, and where the CPU time tracking operation is executed.

- **Time tracking** operation is the process through each time spent during breaks and work is tracked. The timer displayed is pre-configured to execute the Work and Break moments during the times the user chose in Initial SetUp Activity. However, the operation is conducted according to the sensing data the CPU receives from the Data Collector component, as defined in the proposed architecture.

- **Results reporting:** consists in displaying the results gathered during the Time tracking operation, as will be detailed later, in the UI.

Pairing with Mindwave devices To make these operations possible, the first step to use the system is to pair the mobile device in use with Mindwave headset through Bluetooth, which requires the User Device to have the Bluetooth connection enabled. In order to the user understand how he must use

WakeUpNet easily, meeting the Usability requirement, the CPU infers the state of the Bluetooth connection. If there is no Bluetooth connection, an advertisement indicating the user must enable Bluetooth connection appears. Otherwise, the Headset Button is enabled.

Next, to connect the device with WakeUpNet system, Connect Headset button must be triggered - and that's why, at the moment of launching Time Tracking activity, it is the only button enabled. The user must be properly using the headset at this time, what implies the device has charged batteries and is positioned correctly.

When the system recognizes the headset, the UI shows the signal quality obtained. There are three possible levels of signal quality - Good, Medium and Poor.

We found the algorithms for Attention and Meditation only run under a Good signal quality level. As the quality level fluctuates constantly under conditions that are hard to identify, there's no way to assure the system will run under a Good signal quality every time. To guarantee that valid EEG data is acquired at least in the first instants of the session, after the headset is recognized, the Start button is enabled only when a Good signal quality is achieved.

The Time tracking operation only occurs under user order, toggling the Start Button.

Local data structures As stated before, java is the programming language used. Since it is a object-oriented programming language, the most of entities are managed through objects. The structure of WakeUpNet domain is modelled by the UML depicted in figure 4.2.

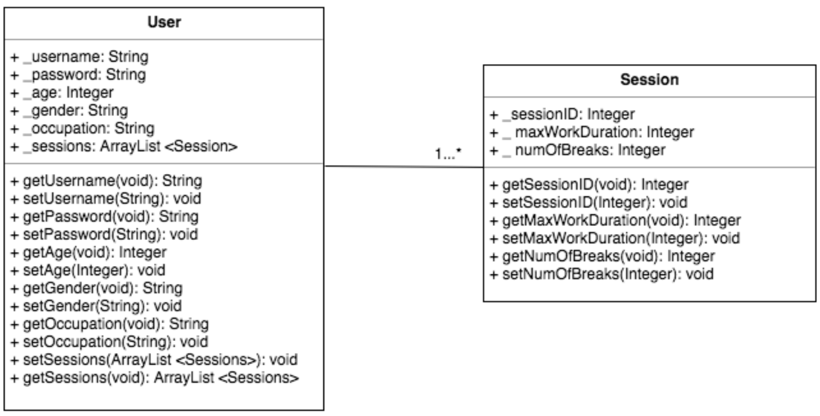


Figure 4.2: WakeUpNet UML Class Diagram

According to the UML of figure 4.2 there are two class objects: User and Session.

The User object contains the following attributes:

- **username and password:** the credentials inserted in the Authentication Activity;

- **age, gender** and **occupation:** the profile fields inserted in the Profile Configuration activity;

- **sessions:** arraylist of the Sessions the user as already performed in previous WakeUpNet utilizations.

The Session object contains the following attributes:

- **sessionID:** this is an incremental unique identifier, attributed to each session belonging to a determined user.
- **maxWorkDuration:** the largest period time in Work status. Matches the ”Maximum number of continuous work” field displayed in Results Reporting activity ;
- **numOfBreaks:** total number of breaks made during the session. Matches the ”Number of breaks” field displayed in Results Reporting activity

Pairing with MindWave devices algorithm Algorithm 4.1 shows the pseudo-code of how this first steps to take before starting the work session were implemented.

Algorithm 4.1: WakeUpNet Set Up Connection

```
Read ConnectHeadset button state ;
if ConnectHeadset button state = TRIGGERED then
Read signal quality state ;
if signal quality state != GOOD then
Wait for Good signal quality
else
    Enable Start Button;
    Read Start Button state;
    if Start Button state = TRIGGERED then
    Enable Stop Button;
    Start Timer;
    CPUDecision();
```

In Algorithm 4.1 is shown that the main goal of the first steps is to reach the statement ”Start CPUDecision()”. This represents the start of the CPU operational algorithm, that was detailed in section 4.3 as it faced some implementation particularities relatively to the proposed architecture.

During Time tracking, CPU gathers the five type of information results identified in the UI implementation details, who are passed as an Intent to the Results Reporting activity.

The ”number of breaks” and the ”Maximum number of continuous work” values are used to set the ”maxWorkDuration” and the ”NumOfBreaks” attributes of the Session object. This session object is added to the User object’ Session ArrayList. Only after this step, the new results may be updated in the Remote Database, if the user manifests its will to do it, pressing the Results Button. The reason why only this two types of results information match Session objects attributes and are sent to Remote Database concern the Extensibility requirement identified in the 3 section. Although the implementation focused on displaying the user results of the just ended session, a future work to extend WakeUpNet implementation must display the results of every user registered in the system. This results will be the ones identified in the proposed solution, that includes the "maxWorkDuration" and the "NumOfBreaks" but not the remaining ones implemented to offer the user the best individual information results for WakeUpNet testing.

4.2.2.D Client Communication Manager component

As described in the proposed architecture of chapter 3, the client communication manager is a triggering process, i.e., makes requests that trigger a a server reaction. This component was then implemented through an REST API developed in node.js, that abstracts the communication between the User Device module and the Remote Server module. Table 4.3 summarizes the client requests and the respective server actions.

TAG	Params	HTTP verb	Request
/user	username	Get	Request to know if the user is already registered in the Remote Database or not.
/newUser	username, nrOfBreaks, maxWorkDuration, sessionNr	Post	Request to include in the Remote Database information about a new user.
/getSessions	username	Get	Request to get the information about the user's sessions performed in previous WakeUpNet uses.
/sendData	nrOfBreaks, maxWorkDuration, sessionNr	Post	Request to update the Remote Database with information about a new session of the user.

Table 4.3: Client Communication Requests

As seen in table 4.3, the Client Communication Manager can send four types of requests to the Server Communication Manager, each one abstracted through one request tag known by both the Client and the Server Communication Managers: "\user", "\newUser", "\getSessions", "\sendData".

\user tag identifies the request made in Authentication Activity to determine if the username inserted refers to a user that is already registered in the system or not. If the user is registered, the Client Communication Manager sends the \getSessions request, to load every Session the user already made. Otherwise, if the user is not registered yet, the \newUser request is sent. In that case, the server will prepare its database to accommodate a new user, with the fields of the Results he will obtain in the end of the Session he's about to start after the authentication procedures. Finally, the \sendData request is performed when the user gets in the Results Reporting activity, to update the remote database with the results he got in the session that has just been completed.

4.2.3 Remote Server module

As in the proposed architecture, the Remote Server was implemented with an Server Communication Manager and a Remote Database.

The server provides a webservice, being accessed through Hypertext Transfer Protocol (HTTP) requests, sent from the Client Communication Manager. The HTTP requests are interpreted for the Server Communication Manager, implemented as a Rest API written in node.js. This API translates the request tags, previously stated in table 4.3, into the actions to perform on the remote database. This translation is depicted in figure 4.4.

TAG	Params	HTTP verb	Response
/user	username	Get	Database is queried for a user registered with a given username.
/newUser	username, nrOfBreaks, maxWorkDuration, sessionNr	Post	A new database entry is created for a new user with the given username and the given results fields.
/getSessions	username	Get	Database is queried for all the records concerning the session data relative to the user with the given username.
/sendData	nrOfBreaks, maxWorkDuration, sessionNr	Post	Database is updated to add to the user records the data of a new session.

Table 4.4: Server Communication Responses

As stated before, due to the faced restrictions, the Remote Database was implemented in a very simple format.

To test WakeUpNet only was required to store the results from one session for every user. For that, only one table was required and then implemented, according to the Relational Model stated in figure 4.3.

Results

Username	SessionNr	maxWorkDuration	nrOfBreaks

Figure 4.3: WakeUpNet implemented Remote Database Relational Model

4.3 CPU operation

As you may recall, WakeUpNet most complex algorithms are the CPU operational and the sensing data analysis algorithms, idealized in section 3 by the algorithms 3.2 and 3.3, respectively.

Recalling the operational algorithm, it defines a finite state machine, detailed in 3.7, that determines the system can operate in four different states - Config, Work, Break and Results. The system only stays in both Config and Results status once time a session, when the user accesses Initial SetUp activity and the Results Reporting activity, respectively. The operational algorithm implementation followed the 3.2 with some implementation particularities that will be discussed later in this session.

Sensing analysis The implementation of sensing data analysis, however, faced more complexity than the expected in the 3.3 algorithm.

First, a different threshold value was considered. According to the information exposed in table 4.1, the right value of threshold to define the states of ”Focused” and ”Not focused”, as well as ”Relaxed” and ”Not relaxed” is 40. If the proposed threshold of 50 was used, the system could consider users in the category of ”Neutral” as ”Not Focused” and ”Not Relaxed”, resulting in erroneous anticipated and extended breaks.

Similar to the planned algorithm 3.3, the data streaming is constantly captured, processed and CPU inserts the Attention and Meditation values in two circular buffers, respectively. eSense values are captured every second being the measured values subject to normal ranges of variance and fluctuation, during to personal and environmental conditions. So, the attention and meditation analysis is made after the average of values captured during a defined period of time.

However, we realized that to run Mindwave EEG algorithms a Good signal quality is required. When in Medium quality level, the latest eSense values read are just repeated every instant, what needs to be considered to compute the accurate average.

This issue was solved attributing the value ”999”, a value totally out of the eSense scale, for every value read under a signal quality other than Good. This number represents an invalid read value, since its real numeric value would only be the repetition of the last value read under a Good signal quality. On the other hand, for every value read under a Good signal quality, the read numeric eSense value is inserted in the buffer.

The circular buffer is then fulfilled with valid and invalid values, being only the valid ones considered to the average computation. There’s need to define a minimum quantity of valid values to compute an accurate average. We decided for half of the circular buffer length, i.e., 150 valid values are required to compute the average of sensing data.

Sensing analysis algorithm The sensing analysis occurs in the instants foreseen in algorithm 3.3, i.e.:

- **Attention verification:** occurs after the first 5 minutes (300 seconds) of work, and again after every other minute until the work time pre-defined is reached or the user is considered out of focus;
- **Meditation verification :** occurs when the break time pre-configured is reached, considering the last five minutes of break, and again after the extended break time ends.

For this reason, the buffer was implemented with a capacity of 300 values as in the proposed architecture. When full, the new read values are inserted in the buffer and the older ones are removed, following a FIFO fashion, according to the nature of a circular buffer.

If, at the moment of sensing analysis, the minimum quantity of valid values required to compute the average of the read values is not met, the sensing analysis just doesn’t occur. However, in the next instant of sensing analysis, the buffer will be fulfilled with some new values, so, if there are enough valid ones, the analysis can be completed.

In the case the system can never get enough valid values, the work and break are managed according to the user inputs made in Initial SetUp activity.

Pseudo-code 4.2 summarizes data sensing and the circular buffer operation.

Algorithm 4.2: Implemented algorithm for CPU sensing data analysis

```
Boolean user_is_focused = true;
Boolean user_is_relaxed = true;
Integer threshold_constant = 40 ;
Integer first_analysis_constant = 300 ;
Integer following_attention_analysis = 60 ;
Integer following_meditation_analysis = TBreak \2;

Fill buffer( Buffer ):
if Buffer lenght < first_analysis_constant then
    removeLastElement(Buffer)
if Signal quality level = Good : then
    addFirstElement(Buffer) ⟵ streamingdata;
else
    addFirstElement(Buffer) ⟵ 999;

Attention analysis(Buffer):
if Valid values > Buffer lenght\2 then
    compute average;
    if average > threshold_constant then
        user_is_focused = false ;

Meditation analysis(Buffer) :
if Valid values > Buffer lenght\2 then
    compute average;
    if average < threshold_constant then
        user_is_relaxed = false ;
```

Notice the meditation analysis suffered an adjustment. After the first analysis, ensuing verifications doesn't happen every other minute, as planned in the proposed architecture, but only in the end of the extended break time period. This happens because we decided to give the user more decision capacity. The user is never forced to extend his break, alternatively, if he hasn't rested enough enough during his chosen break time, an Alert Dialog is created to ask him if he pretends to extend his break or to return to work. If he decides to extend his break, a verification every other minute could lead to return to work only after one extra minute of break. In terms of Usability, this is very confusing to the user. So, the following verifications are only made after the extended time of break as been completed. We defined this extra time as half of the pre-configured break time.

CPU operational algorithm After defining this way the moments of verification, CPU operational algorithm was implement according to the pseudo-code stated in 4.3.

Algorithm 4.3: Implemented CPU operational algorithm

```
Boolean user_is_focused = true;
Boolean user_is_relaxed = true;
Integer TWork;
Integer TBreak;

SetUp( user input) : {
TWork ⟵ userinput;
TBreak ⟵ userinput;
}

Work status routine( TWork ) : {
while T < TWork do
    Attention analysis();
    if user_is_focused = false then
        Break status routine ( TBreak );
Break status routine ( TBreak ) );
}

Break status routine ( TBreak ) : {
if T = TBreak then
    Meditation analysis();
if user_is_relaxed = true then
    Work status routine ( TWork );
else
    Break status routine ( TBreak\2 )
}
```

4.4 Limitations and constraints

During WakeUpNet implementation were faced many limitations.

First, the device limitations. Choosing Mindwave Mobile as the EEG sensor made it easy in some issues but made it hard in others as well.

The device is complicated to use. The device used was an old version and the connection with WakeUpNet through Bluetooth, a naturally instable technology, resulted in many situations of trouble to pair Mindwave with mobile devices for no identified reasons.

Also, we found out the device requires a Good signal quality to collect Attention and Meditation sensing data. To accomplish this quality level, well charged batteries and correct device positioning are required. The first issue is easy to solve - just changing batteries to new ones - but the second one can take much time to solve.

Besides, if the signal quality decreases to a Poor level during some time, the system was configured to reset. This restriction was imposed by the developer toolkit as default, and the authors decided to maintain it since there was no point in using the system if there was no sensing data being collected. The problem here is the signal quality is very instable, fluctuating in time. Many testing sessions started with a signal quality in a Good level, but this level suddenly decreased after a considerable period of time, interrupting the session without generating any results.

Additionally, the device manifested overheating problems, often turning off after some hours of use. This totally conditioned the duration of time a day spent using the device, for implementation and for testing proposes.

WakeUpNet is supposed to be used during a considerable amount of time since its finality is to monitor work or study, activities that the users supposedly perform during hours. To test the full proposed architecture, there would be required at least 15 users, that should use the system several times, along time. Recalling the options of History results the users can select, every user in the system should at least use WakeUpNet once a day for one week. Recall as also that one WakeUpNet session, which includes several work and break sets, may took hours to complete. In the best scenario, if every user only used WakeUpNet for sessions with one hour of duration, the Test phase would took 525 hours, i.e., more than 21 days. This is too long for an optimistic scenario.

Finally, the headset was really intrusive, becoming uncomfortable after a few minutes of usage.

Due to the device intrusiveness, the overheating problem and the instability of the reached signal quality, it was not possible to perform tests for hours in a row, not even more than one user a day. For the same reason, the testing users could not benefit from Results activity in its full potential, since they only had access to the results of one single session. This totally conditioned the implementation and test conditions and, consequentially, some of implementation goals were dropped out or ended not to be tested.

The extent of the tests duration also made it difficult to find testing users available during all the test phases.

4.5 Synthesis and discussion

The implementation of a system naturally faces more limitations and unexpected issues that the architecture design couldn't foresee. The main restrictions faced during WakeUpNet implementation lead us to the implementation options as follows:

- **Sensors:** This component was implemented through a Mindwave Mobile sensor in its commercial format. This way, the component was implemented through a portable device, able to accurately read human's mental status and send the sensing data to a mobile device via Bluetooth. The decision to use this equipment for sensing data allowed a simpler implementation of the Data Collector, since the device producers provided developer kits to easily integrate the data analysis in a mobile application. However, this equipment option also lead to another problems. First, using it in its commercial form discarded the usability requirement of implementing a non intrusive system. Second, the device was difficult to use and consequently lead to an extended period for testing proposes.

- **User Device:** This component included:

  - **User Interface:** supports the reception of user inputs, such as his authentication, the initial setup, the profile information and commands for actions. It also displays the timer, the alerts and the results during the application running.

  - **Data Collector:** this component was implemented with the Android Developer Tools provided by Neurosky. It supports the reception of the sensing data, receive through the Sensor component, and classifies the mental sates of attention and relaxation in a quantitative scale from 1 to 100.

  - **CPU:** is the support for the user actions, namely the Authentication, Initial Set Up, Profile Configuration, Time tracking and Results Reporting. To support the Time tracking operation, defined by an intricate operational algorithm, it infers the user mental status according to the information received from The Data Collector and a complex sensing data analysis algorithm. The mentioned algorithms are the most complex and determinant to the system.

  - **Client Communication Manager:** supports the communication between the User Device and the Remote Server through Wi-Fi. It is implemented by a node.js API that abstracts the communication details in the client-side. It supports four different requests:

    * Request to know if the user is already registered in the Remote Database or not.
    * Request to include in the Remote Database information about a new user.
    * Request to get the information about the user's sessions performed in previous WakeUp-Net uses.
    * Request to update the Remote Database with information about a new session of the user.

- **Remote Server:** This component was composed by the Server Communication Manager, that processed the requests from User Device and a Remote Database. The Remote Database is accessed according to the stated possible requests. The actions performed to each request type are:

  1. Server queries the database for a user registered with a given username.
  2. Server creates a new entry in the database for a new registered user.
  3. Server queries the database for all the session data records relative to the user with the given username.
  4. Server updates adds to the user records the data results of a new session.

5

Tests

To validate WakeUpNet system in its full dimensions, two types of tests were executed: functional tests and user tests.

This chapter describes the different types of tests in sections 5.1 and 5.2, in terms of the methodologies used, the test scenario in which they were performed, the results obtained and the analysis of them.

5.1 Functional Tests

Functional tests consisted in using the system in different phases of implementation and keep record of the use conditions. This tests aimed to conclude how the different technologies used could be integrated in order to meet the requirements.

Functional tests objective is to test the equipment and implementation before testing with real users.They were planned to be performed after each phase of implementation.

5.1.1 Methodology

As stated before, functional tests were performed after the following phases of implementation:

- **Phase One:** at this phase, the first draft of the mobile application was implemented, including only the Main activity. At this time, the timer was set up to work by the traditional Pomodoro Technique rules, i.e., considering 25 minutes for work duration, with 5 minutes of break in between. MDT was used to integrate MindWave device in the system, already capturing properly brain activity data.

- **Phase Two:** at this phase, the first steps of CPU component were implemented, regarding the attention analysis algorithm. The timer was still working according to the traditional Pomodoro Technique rules.
- **Phase Three:** in this phase the timer was set to work as a smart Pomodoro timer, where the pre-defined work and break durations are still the traditional 25 and 5 minutes, respectively; however the system could start the breaks in anticipation, if the collected attention data were manifested user distraction.

After phase one, the system was launched several times, for a minimum duration of 50 minutes, time enough to gather results about two work sessions and one break in between. Sensing values were read to a .dat file to produce graphs recurring to gnuPlot program. This graphs show clearly how the sensing values captured changed in time.

After phase two, a new functional test was performed, concerning different issues, as we will discuss later. The experience acquired in the first functional tests made us to realize it was better to perform test for a smaller duration of 25 minutes. First, because the device becomes uncomfortable to use after a few minutes, mainly because of the ear clip. Secondly, reducing the duration of the test to half would allow us to perform more system runs. Finally, at this time we wanted test issues related only with the attention analysis, which occurs during the work time (the first 25 minutes of the session). This way, we launched the system 30 times, during 25 minutes each, and taking notes of some events, detailed later in this chapter.

Regarding phase three, the User Tests to be discussed in next subsection were performed.

5.1.2 Test scenario

The computer used for tests was a Macbook Pro Retina and the portable device was a One Plus 3T with 7.1.1 Android OS version.

All the functional tests were performed in the same local: office 11.3, in IST at Taguspark campus. This office is located in the second floor of the building, in the internal area, with a window to the main corridor. It has air conditioner, the light conditions were not very good, since it does has a window to the outside and might be a little noisy, when people circulate in the corridor. Hence, to minimize the noise, the window was closed. During the tests, the author was the only person in the room.

The sensors component, as stated before, consisted in the MindWave Mobile Device in its commercial form. It was equipped with full charged batteries. We used Duracell Ultra Power batteries, which have a power check functionality that allowed us to verify the state of batteries charge. After every test, this state was verified and the batteries were changed every time they were under 75% of charge.

Every new test, the system was totally configured from the beginning: the mobile application was launched again, mobile device Bluetooth was turn off and on again and the sensors device was paired with the mobile device again.

Since the system is supposed to be used during an intellectual activity, the authors performed the tests using the system while developing it at the same time.

5.1.3 Test types

Two different set of tests were realized:

- **Validation of the sensing data capturing:** these tests were performed after the first phase of implementation was done. The objective was to understand how the sensing data was read and interpreted by the Sensor module and sent to the User Device module

- **Tests to the signal quality:** after the first phase of tests, we needed to understand how the functionalities affected by the signal quality reached would develop. The objective was to understand, during a normal run, what is expected from the system in terms of the time spent during the pre-configuration and the number of sensing analysis made.

5.1.4 Results and analysis

Sensing data validation The tests to the validation of sensing reads resulted in the production of the graphs depicted in figure 5.1. As already explained, MDT offers algorithms for reading EEG values, i.e.,

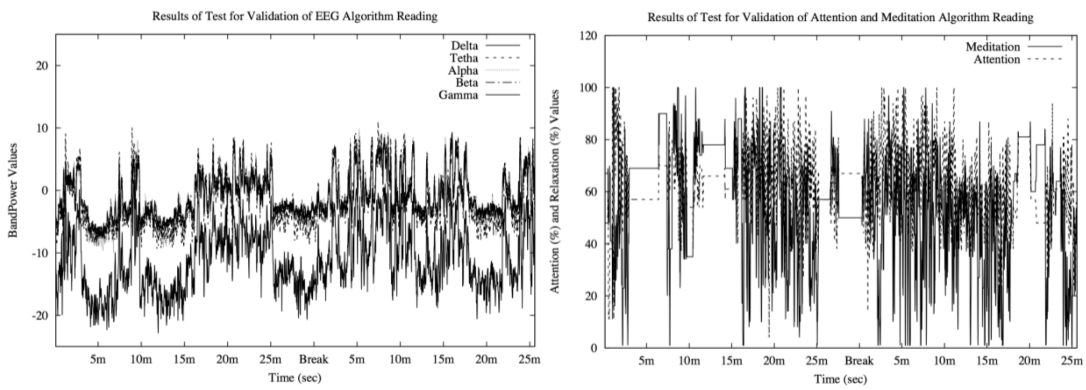


Figure 5.1: Results of functional test for validation of sensing data

the values of each brainwave frequency; and algorithms for quantizing Attention and Meditation mental status in a scale from 0 to 100, the eSense scale.

The first graph shows the variation of the EEG raw sensing values read in time. The second graph shows the eSense values for Attention and Meditation read and processed in time.

Since at every second is captured a value for every single brainwave (five in total), after 50 minutes of usage, more than 15000 values were captured, making it hard to interpret. We then individualized the values captured during only the first minute as depicted in graph 3 of figure 5.2. In graph 4 of the same figure are shown the respective Attention and Meditation eSense values produced at the same time, resulting from the MDT processing algorithm.

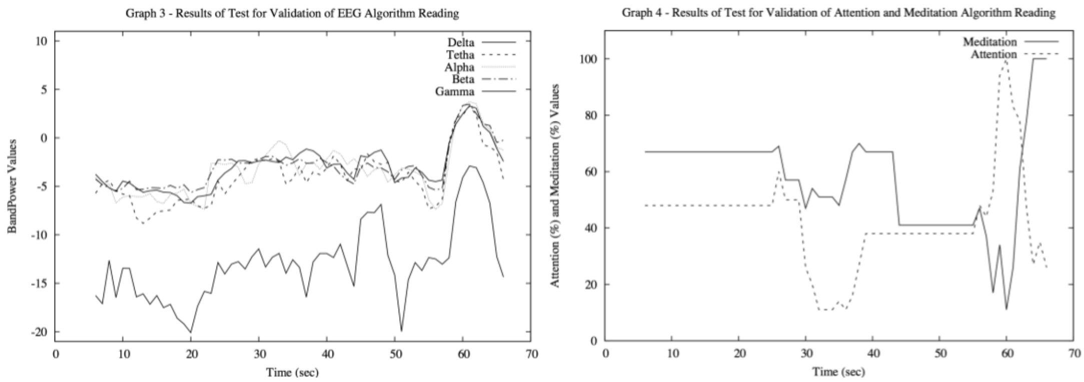


Figure 5.2: Results of functional test for validation of sensing data

Analyzing the graphs in figure 5.2, we can not find an intuitive relation between the two of them. Although in the period around the 60 seconds all the bandpower values increase and the eSense values for the both mental status also increase. This could indicate a direct proportion relation. However, analyzing the full period we easily conclude that it was just a mere coincidence. The fact that, in the first 20 seconds, the eSense values are constant, but the BandPower values are slightly disperse got our attention.

Signal quality validation We start to took in account how the quality signal changed during the captures. As we mentioned before, the signal quality variates between three levels: Good, Medium and Poor. MDT documentation informed us the SDK pauses when the signal is in the level Poor, only running again when the signal quality returns to an higher level.

Then we performed another test,with the same methodology, but now taking in account the signal quality during the process. The results are shown in graph depicted in figure 5.3. This graph only concerns a short period of time, but the results obtained for the total duration of the test can be found in appendix 8.

The information in this graph show us the level of signal quality effectively affects the sensing data processing. When the device only gets a Medium signal quality, the raw data is not converted in eSense values, instead, the last eSense values obtained are just repeated until a Good signal quality is reached again. In terms of the raw data, the previous graphs were enough to conclude that they were always readings, since the values are different every second.

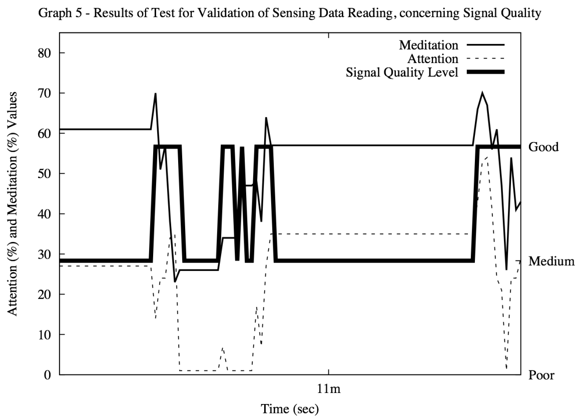


Figure 5.3: Results of the eSense values captured for Attention and Meditation for the first functional test and the respective signal quality at every second

Taking this into account, we concluded we had two options to implement the algorithm for attention and meditation analysis:

1. Use only the raw data collected and develop an original algorithm for converting the five brainwave frequencies captured every second into a quantization scale.

2. Use the MDT algorithm for mapping the raw values in eSense scale, and implement the system with methods for balancing the signal quality.

After some research, the authors found that Neurosky does not provide any information about the process MDT uses for converting the raw values in eSense values. We also found that these actually require really complex calculations [16]. Besides that, we do not have anyone in the team from the neuroscience area that can give us the knowledge to understand what needs to be done nor the confidence in the way used to process data and obtain meaning from it. With this, the authors concluded that developing a process to interpret the raw sensing data would be too much complex and time consuming for our project, and our best option would be trust in the processing algorithm offered by the MDT and use it for WakeUpNet.

The measures took to obtain the required signal quality to run the system properly were then:

- To perform a sensing analysis, whose consequence can be starting a break or returning to work, the system must have achieved a minimum quantity of values collected under a Good signal quality (the so called valid values). This is because, when in the other levels of signal quality, the device can not read eSense values.

- To launch the system, it is necessary to reach a Good signal quality first. This assures the system is launched automatically collecting valid values.

In consequence of the first measure, the system can be used without making a decision after the user mental status. We had to determine what we could expect from this situation. We then counted the number of times the system performs a decision based on the sensing analysis. The results are shown in figure 5.4.

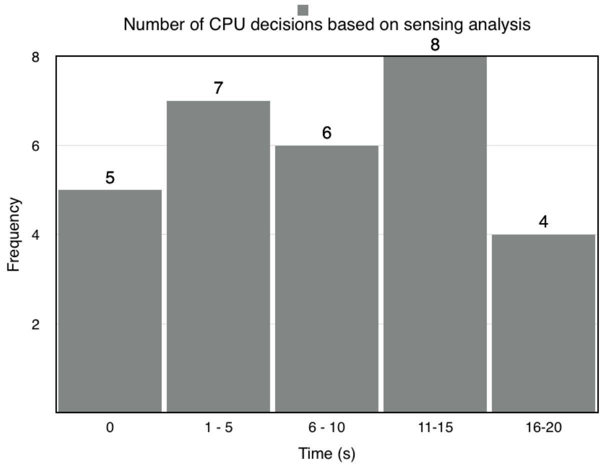


Figure 5.4: Results of functional test concerning the number of CPU decisions performed

If the system runs under its full potential, if the work duration is configured to last 25 minutes, it must perform a decision 20 times: the first, after the first five minutes of run, and after it, every minute.

According to the figure, after launching the system 30 times, in 5 runs the system could not get valid values to perform a sensing analysis at all, working as a regular timer with pre-configured duration for working and break. This means, the system was not able to execute its main functionality for about 17% of cases. Also, in 7 times, the system only made less then 5 decisions what is about 23% of the cases.

However, in 8 times, i.e., approximately 27% of the cases, the system was able to perform 11 to 15 decisions, and in 4 timer, i.e., approximately 13%, the system was able to perform 16 to 20 decisions. The remaining times, the system performed 6 to 10 decisions.

In terms of performance, this values are considered low, since in most of the times the system is not able to perform its main operation. However, the system still runs offering all the functionalities, but with fix times setup by the user, for the duration of work and breaks.

The second decision affected the configuration necessary to use the system. The signal quality reached is affected by the position of the device in user's head what is a difficult situation to manage. Sometimes, to reach the necessary level takes much time trying to fix the device position. Since this is an usability concern, we counted the time required to perform this configuration step. The results are shown in figure 5.5.

After launching the system 30 times, 6 times the system was not able to detect the device at all. This means, 20% of times, the overall system failed, what is a reliability concern. However, is not a critical

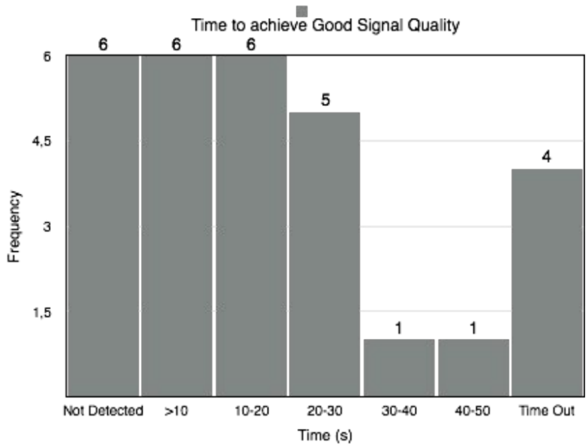


Figure 5.5: Results of functional test concerning the signal quality

concern since in the remaining 80% of times, the system can be launched and it will run properly until the end of the session. Also note that, the device used was always the same, and it was some years old, so we cannot assure if its operation could have been affected through use and time.

We considered that more than 50 seconds the system should be considered unreachable. That situation happen 4 times, i.e., 13% of the cases.

However, in 6 of the cases the system took less than 10 seconds to launch, and also 6 of the cases the system also took 10 to 20 seconds to achieve the signal quality level necessary to launch. In overall, 40% of times the system needs less than 20 seconds to reach a Good signal quality level.

This numbers indicate the signal quality achievement is an instable process, since it it is the same number of times the system reaches the maximum signal quality in less than ten seconds as the number of times the system can not detect the sensor device at all.

5.2 User Tests

The functional tests allowed us to determine how the analysis of sensing data should be implemented in the system, taking the signal quality necessary in account. After this, it was necessary to implement the CPU Decision based on the data analysis made, what corresponds to the phase three of implementation stated in the previous section.

After this, it was necessary to test the system in a use case scenario, to understand if its operation occurs as expected (never discarding the limitations identified by the previous tests).

For that propose, user tests were performed to conclude about the user perception of the system. Our proof of concept also aimed to determine if a biofeedback application could substantial improve the traditional Pomodoro Technique.

5.2.1 Methodology

The test users were selected among IST students that would be studying similar subjects, during the interim tests examination period. We tried to achieve a sample as representative as possible of the students in this institution.

The first activity to every user was to read WakeUpNet tests documentation, shown in appendix 8. This documentation provided the candidates to test users the necessary information to understand if they were suitable to perform the tests and how this would occur.

After that, test users were selected according to a selection survey, annexed in appendix 8. This survey assured the users sample was homogeneous and had the characteristics we pretended to test the system. We set the goal of finding a minimum of 15 users with the pretended characteristics. After selecting our sample, the users were contacted to schedule the test.

Before testing, the users were asked to fill the Pre-Test survey available in appendix 8. This survey finality was to understand in what conditions, that could affect they focus status, the users will perform the test, in terms of the number of hours of sleep, how long they ate the last meal and how many coffees they had. These conditions were selected with the collaboration of Dra. Sofia Sá, a psychologist of Núcleo de Apoio ao Estudante (NAPE). Taking these factors into account we aimed to relate them with the test results. The Pre-Test also queried about how long the user usually studies, how long are usually his breaks and also asked for a self-assessment of his attention status in the day of the test.

Then, the users performed the test, that consisted in testing WakeUpNet system running in two different modes, corresponding to different testing phases:

1. The traditional Pomodoro technique - in this phase, the users were asked to study during 25 minutes and after it, make a 5 minutes break. After this, it was expected the users to understand how the Pomodoro technique works.

2. The smart version of the Pomodoro technique - in this phase, the system was able to adapt the duration of study to the user's attention status. This test phase did not have a defined time period, but taking the goal of our proof-of-concept, the pre-defined times for work and break were 25 and 5, respectively. The system was able to recommend the users to make breaks of 5 minutes every time the user his lacking focus

The sensing data captured during the test was sent to the authors by email.

Finally, after completing the two testing phases, the users were asked to fill the Pos-test survey, seen in appendix 8 that gave us the results of how user perceived WakeUpNet system in general, and in comparison with the traditional Pomodoro technique in particular. This survey also queried about the system intrusiveness.

5.2.2 Test scenario

The tests were realized in conditions as much as possible similar to the conditions of the functional tests. Hence, the same equipment was used although the software version of WakeUpNet differs. The tests were also realized in the same office, to assure they were executed in the same environmental conditions of light, sound and temperature.

Since the MindWave Mobile headset is difficult to configure and fails sometimes (as stated in the results of the functional tests) the authors were present during the tests to provide the necessary support. However they did not interfere in the user activity.

User tests were performed over a sample of 18 test users.

The sample is characterized in terms of personal, professional and social characterization, as displayed in figures 5.6, 5.7 and 5.8, respectively.

Figure 5.6: Personal characterization of the sample analyzed

The sample is characterized for 72.2% males and 27.8% females. Usually, the samples of user for testing are composed in a way to find gender equality, but our goal was to find a sample representative of the IST students population. In terms of age, 93.3% of the testing users was aged between 18 to 25 years old.

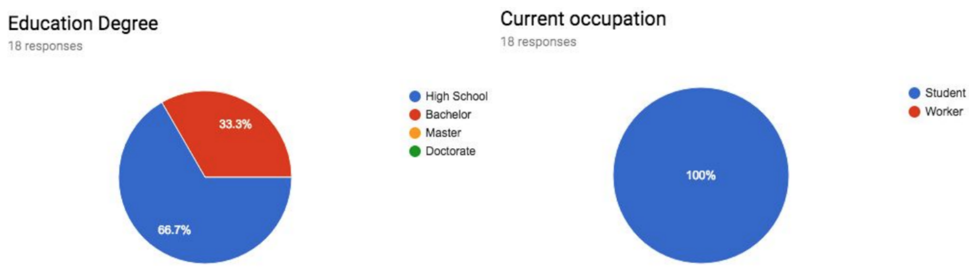


Figure 5.7: Academic characterization of the sample analyzed

In terms of education, the sample is characterized for having an high school degree in 66.7% of the cases, and a bachelor degree in 33.3% of the cases. This happens because the testing users were both bachelor's and master's students at the time.

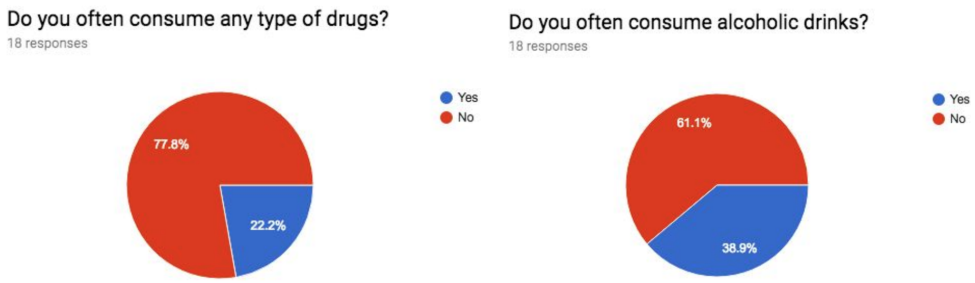


Figure 5.8: Habits characterization of the sample analyzed

Finally, some health habits that could lead to different attention status results were analyzed as well. To obtain an homogeneous sample, we could seek for people who never consume any type of drugs and alcohol but the authors did not consider that would be a realistic sample of a population of university students. Our sample was then composed by 77.8% of people who do not often consume any type of drugs, 22.2% of people who often do it; 61.1% of people who do not often consume alcoholic drinks, and 28.9% who consume it often.

5.2.3 Test types

The user tests were performed by surveys and analysis of the sensing data captured during the test.

- **Pre-Test survey:** as stated in the methodology, the users were asked to answer a survey before the test, the Pre-test survey, and another one after the test, the Post-test survey.

  The answers to the pre-test survey aimed to conclude about:

  – **The study habits of the users:** Our hint is that a person who usually studies for hours without needing a break do not find the Pomodoro Technique useful. To understand how the users perceive the system, we needed to understand how they are used to organize their study.

  – **The biological conditions of the users during the test:** As stated before, we contacted a psychologist to understand what conditions mostly affect students performance.

  – **The activity performed during the test:** We let the users perform the study activity they needed. We pretended to take in account how different activities could be related with different concentration levels.

  – **The user perception of their own performance:** Although the biological factors could indicate the users were not in the ideal conditions to achieve a good performance, the users could not perceive it.

- **Post-Test survey:** aims to analyse how users perceived the system. The Post-test survey tested:

- **Users classification of the experience with the traditional Pomodoro Technique**
- **Users classification of the experience with the smart version Pomodoro Technique**, with focus analysis to recommend the most suitable times for making a break according to it.
- **How the users classify both of the techniques, in comparison with one another:** Even if the users liked both of the techniques or none of them, they can have an opinion of what is better than the other.
- **The intrusiveness of the system:** We wanted the conclusion if the use of MindWave Mobile sensors in its commercial format would really be a limitation for WakeUpNet system.

• **Sensing data:** we also tested the system operation, designing graphs of the users concentration data in time, to analyze their performance during the test in a functional perspective. The aim is to understand if the results gathered matched the users expectation after the test.

5.2.4 Results and analysis

We achieved 15 valid results of tests. In 3 from the total 18 user tests performed, the system failed for not being able to detect the sensor, as we expected according to the results of the functional test to the signal quality issues, discussed in the previous subsection.

Pre-Test survey The answers to the Pre-Test survey, are depicted in figures 5.9, 5.10, 5.11 and 5.12.

Questions Q1 and Q2 of the Pre-Test survey characterized the study habits of the users. The results are depicted in figure 5.9. As depicted these graphs, 66.7% of the users state to study for 25 minutes to one hour before making a break, 13.3% study for more than one hour until making a break and the remaining 20% study for less than 25 minutes. In terms of duration of the breaks, 10 from the 15 users make breaks from 5 to 25 minutes, what is the big majority of the sample.

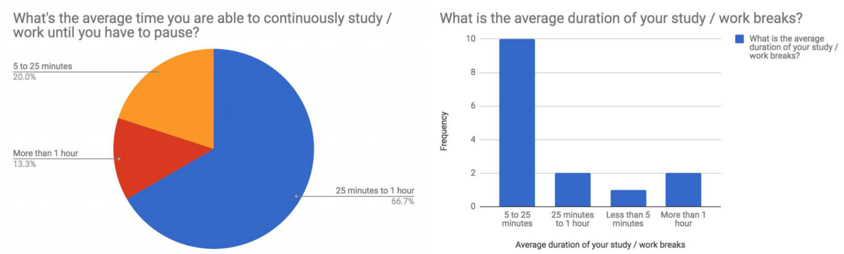


Figure 5.9: Characterization of the study habits: (answers to Q1 and Q2)

Regarding the biological conditions that could affect user concentration status, they are covered by questions Q3, Q4 and Q5 of the Pre-Test survey. The results are illustrated in figure 5.10. The results are as follows:

- 53.3% of the users had 6 to 8 hours of sleep, and 20% had more than 8 hours what indicates the majority of the user should not have tiredness affecting their concentration status.
- 46.7% of the students had their last meal in less than one hour. This results are not conclusive. This either can mean the most of the users were not hungry during the test, not having that factor diminishing their focus, either can mean the most of the users were experiencing the state of laziness shortly after a big meal.
- Finally, 46.7% of users did not have coffee at the moment of performing the test. Once again, this results are inconclusive because it did not had information if the users usually have or not coffee or other energetic drinks. They either could be lacking of the caffeine their body is used to or not. Also, 26.7% of users had one coffee, and 26.7% had two or more. They either could be feeling stressed and anxious due to the presence of caffeine in their body or not because they are used to that condition.

To take accurate conclusions about the answers to this questions the authors of this project had to gather more information about the users habits and biological conditions, preferably with the help of a psychologist or health professional but unfortunately we had not that possibility until the end of the project.

Figure 5.10: Characterization of the biological conditions (answers to Q3, Q4 and Q5)

Regarding the activity performed during the test, the results obtained by the answer of question Q6 are depicted in figure 5.11. The results show that the seven of the fifteen users were making practical exercises, for example, calculations, during the test. Five were reading and four were writing. The remaining ones were programming. We pretended to determine how different types of activities require and stimulate different levels of concentration, but to infer about it we need a bigger sample and again, the help of professionals to validate the results.

Finally, the results of last question, used to assess the user's perception of their focus is depicted in refPreTest4. The answers to the last question of the pre-test showed that six of the users classify their focus status in three, in a scale of one to five. Five user classify it as four. The objective of this answer was to map the users own perception of their attention with the results obtained of the test. Our guess is

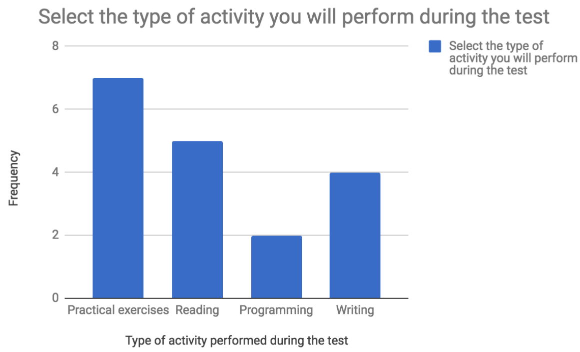


Figure 5.11: Activities performed during the test (answer to Q6)

that some users may do not have the perception of their real tiredness and study performance. However, once again, to take an accurate conclusion we needed to take in account all the factors concerned in the last questions, with a bigger sample and professional help.

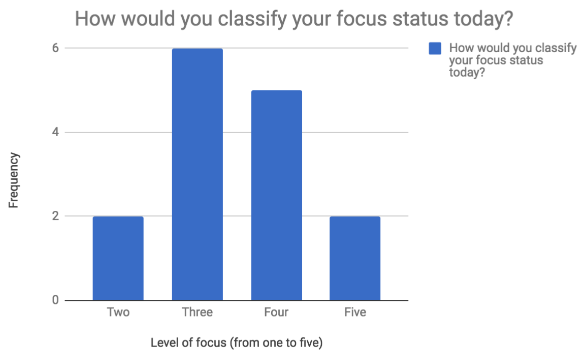


Figure 5.12: Characterization of the users perception of their own performance (answer to Q7)

Post-Test survey- Pomodoro technique The answers to the Pos-Test survey, concerning the experience with the traditional Pomodoro technique are depicted in figures 5.13, 5.14, 5.15 and 5.16.

Figure 5.13 depicts how user perceived the Pomodoro technique, by stating about the validity of break time. This result from the analyis of the answers to question Q1. As can be observed:

- 46.7%, consider the moment to take a break, according to the traditional Pomodoro technique was made in the most suitable moment.
- the remaining users considered that either the break did not occurred in the right moment or it is not relevant to have a break.

These are expectable results since our first statement of the traditional Pomodoro Technique is precisely that it does not fit well for everyone.

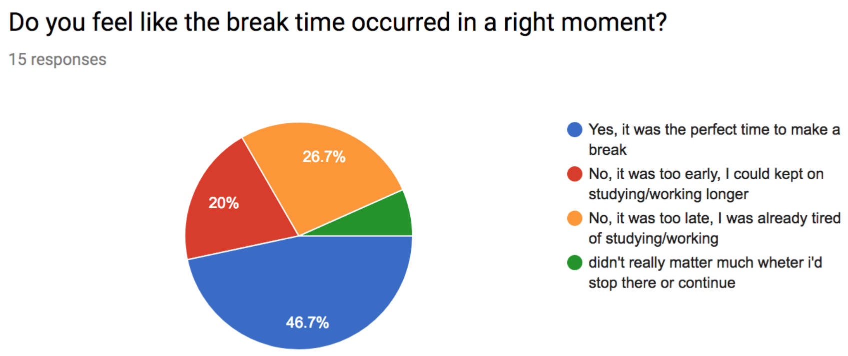


Figure 5.13: Users classification of the experience with traditional Pomodoro technique (answer to Q1)

However, we wanted to know if, even in that conditions, the users perceived this technique as useful. The next figures, illustrate the results, obtained by the answers of questions Q2, Q3 and Q4. The main results are then:

- 53.3% of users classify the utility of the traditional Pomodoro technique for time management during study as four, in a scale from one to five.
- 46.7% of users classify the utility of the traditional Pomodoro technique for improving their study performance as four.
- 60% of users would use again the traditional Pomodoro technique.

These answers reveal that, approximately half of the users think the technique is suitable for them.

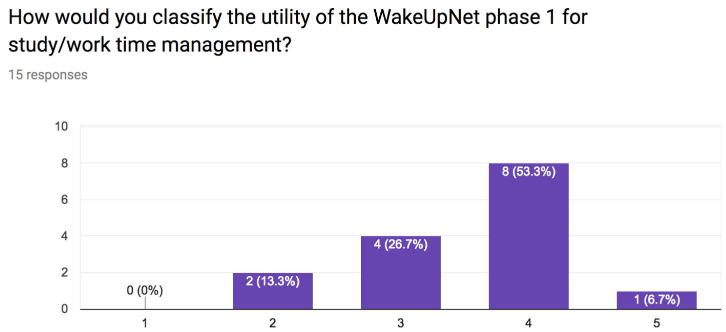


Figure 5.14: Users classification of the experience with traditional Pomodoro technique (answer to Q2)

Post-Test survey- WakeUpNet intelligent Pomodoro technique The answer to the Post-Test Survey regarding the smart version of the Pomodoro technique, are depicted in figures 5.17, 5.18, 5.19 and 5.20.

The main results of the experience with the smart version of Pomodoro Technique are then:

- 53.3% of tests users consider the system did not detect when they needed a break.

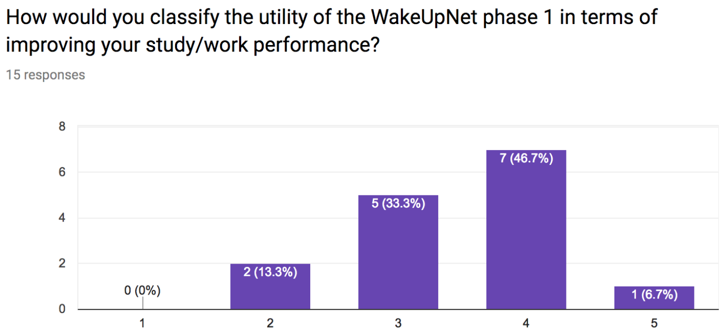


Figure 5.15: Users classification of the experience with traditional Pomodoro technique (answer to Q3)

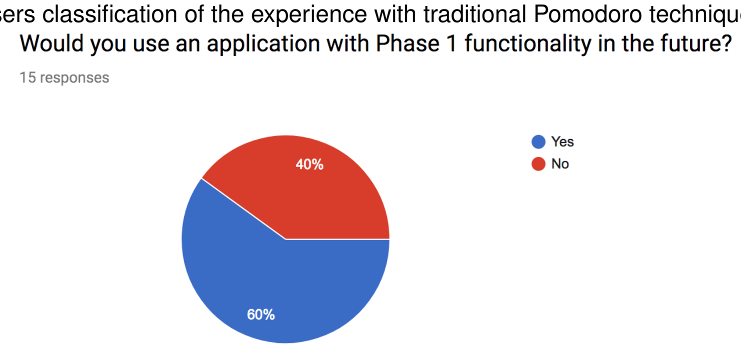


Figure 5.16: Users classification of the experience with traditional Pomodoro technique (answer to Q4)

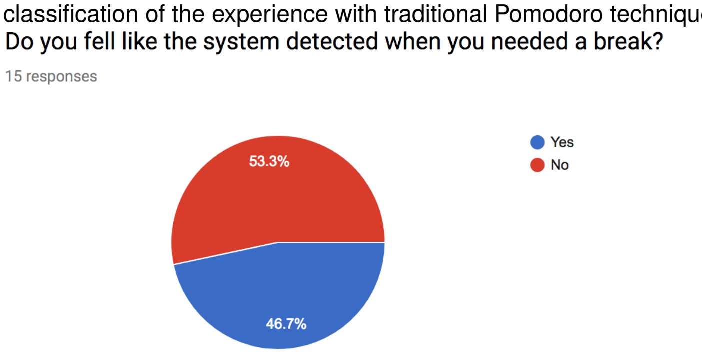


Figure 5.17: Users classification of the experience with smart Pomodoro technique (answers to Q5)

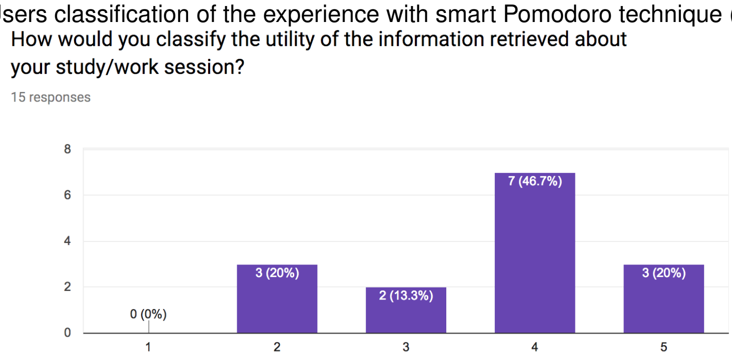


Figure 5.18: Users classification of the experience with smart Pomodoro technique (answers to Q6)

- 46.7% classify the utility of the information retrieved about their study session as four, in a scale of one to five.
- Only 33.3% of users classify the utility of the traditional Pomodoro technique for time management during study as four, in a scale from one to five. But 26.7% classify as five.

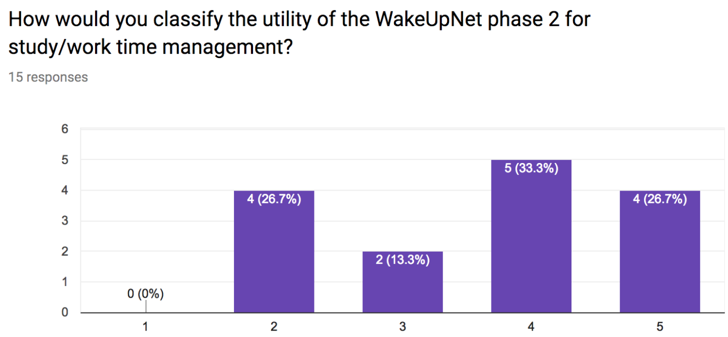


Figure 5.19: Users classification of the experience with smart Pomodoro technique (answers to Q7)

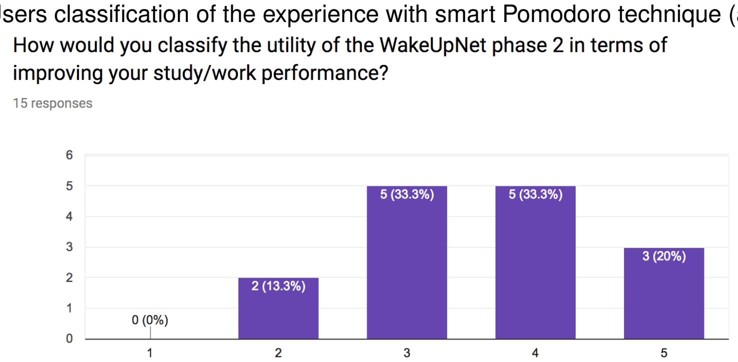


Figure 5.20: Users classification of the experience with smart Pomodoro technique (answers to Q8)

- Only 33.3% of users classify the utility of the traditional Pomodoro technique for improving their study performance as four, in a scale from one to five, but 20% classify as five. Also only 33.3% classify as three.

- 66.6% of users would use again the smart version of the Pomodoro technique.

If the majority of users do not think the system detect when they needed a break, we conclude these users do not think the system worked as it supposed to be.

In terms of improving study performance, there are more users (53.3%) classifying the traditional Pomodoro Technique as Four in a scale of One to Five, than users giving the same classification for the smart version of the Pomodoro Technique (33.3%). However, more users (26.7%) classify the smart version as Five, when only 6.7% gave this classification to the Traditional Pomodoro Technique. This means, 60% of users classify both of the techniques with an utility level of more than Four for time management.

In terms of improving study performance, there are more users (46.7%) classifying the traditional Pomodoro Technique as Four in a scale of One to Five, than users giving the same classification for the smart version of the Pomodoro Technique (33.3%). However, more users (20%) classify the smart version as Five, when only 6.7% gave this classification to the Traditional Pomodoro Technique. This means, in general, half of the users (53.3% for the traditional version and 53.4% to the smart version) classify both of the techniques with an utility level of more than Four.

Post-Test survey- comparison of techniques The answers to the questions about the comparison of both methods are depicted in figures 5.21 and 5.22:

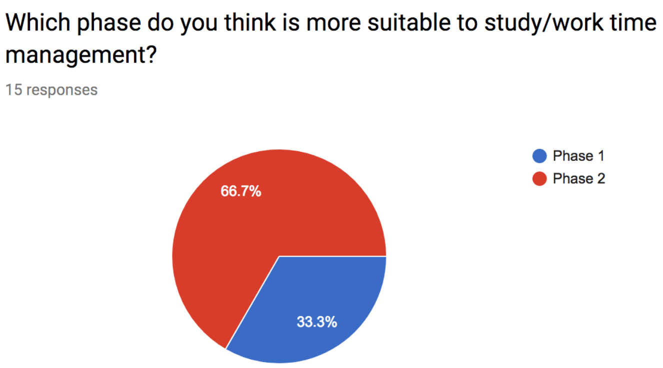


Figure 5.21: Comparison of traditional and smart Pomodoro techniques (answer to Q9)

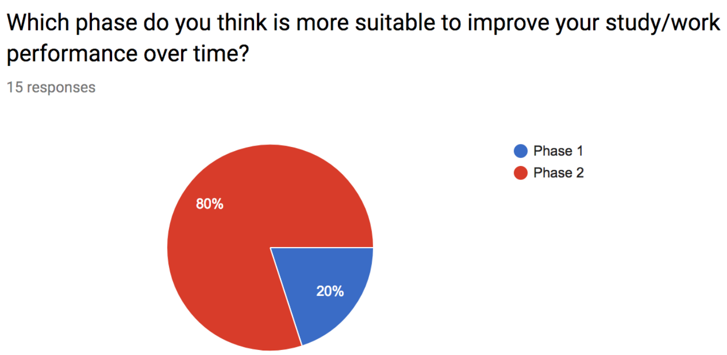


Figure 5.22: Comparison of traditional and smart Pomodoro techniques (answer to Q9

The results are that 66.7% of users consider the smart version of the Pomodoro Technique was more suitable for their time management during study and 80% of users consider the smart version of the Pomodoro Technique was more suitable for improving their study performance. This means that the majority of the users prefer the smart version of the Pomodoro Technique although they found the experiences with both the techniques were similar.

Post-Test survey- intrusiveness of the system The answers about the intrusiveness of the system are depicted in figure 5.23. The results show that 40% of users classify the system comfort of three in a scale of one to five, and 33.3% classify as two. This classification is very low, as made us conclude the system intrusiveness is really one limitation that must be improved.

Sensing data analysis Finally, the results of the data collected during the tests are found in annex 8. We pretended to gather these results in order to determine if the users perception of the system was accurate. However, this relation became very difficult for many reasons.

Three different cases were used to illustrate the different cases that we found.

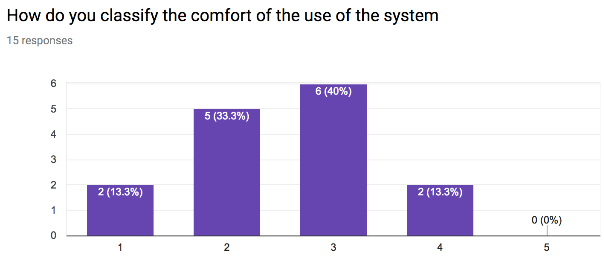


Figure 5.23: Intrusiveness of the sysyem (answer to Q11

The first case discussed is user D. The user manifested being out of focus after 6 minutes of test, making a break of five minutes in that moment. The test ended before 20 minutes of test because the user chose that way. The results obtained by the sensing analysis are depicted in figure 5.24.

As it can be observed in the figure 5.24, the data was collected under a Good signal quality since there are no constant moments in the graph. The numeric values in specific instants represent the average of the attention values collected. The attention threshold below which the user is considered out of focus is also evidential. According to the results gathered we can state that, in that case there is clear perception of the user about the right time to make a break and the smart Pomodoro provided the opportunity to interrupt the work after that instant.

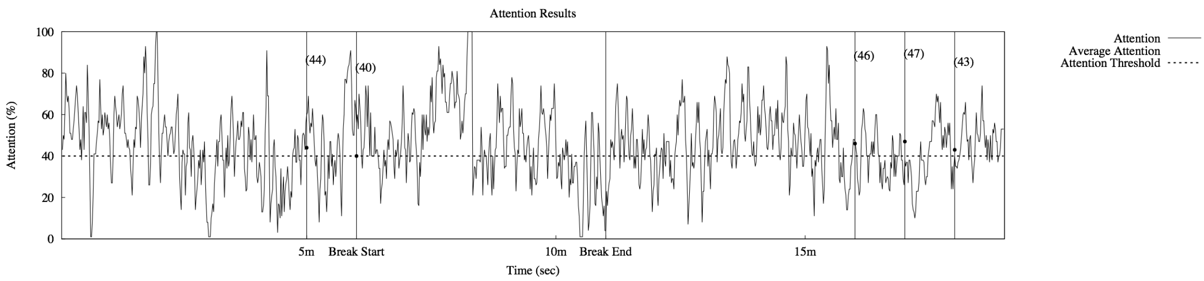


Figure 5.24: Attention results of the user D

In another case, user L, made a break after only 5 minutes of working. He stated he usually works from 25 to 1 hour until making a break, but also stated that the break happened after too long and he was already tired. The results of the sensing analysis are reported in figure 5.25 and proofed to match the users expectation. This is acceptable since the test to the smart version of Pomodoro Technique was made after the test to the traditional version, and the test may be performed after the user had already studied for a long time in that day. It is possible he was already tired after that moment, but we had to had more information about to conclude about it.

Another complicated case is User M, for example. The user stopped the test before making any break. The sensing data analysis is depicted in figure 5.26. The results reported that the user was focused until the end of the test and so there was no valid reason to make a break. So his answers about the break being made in the most suitable moment for it, or not, are not valid. However, we cannot discard the possibility that he already tired and the system didn't detect it properly. Since we give the users the chance to use the smart version of the system for how long they wanted, as stated in the

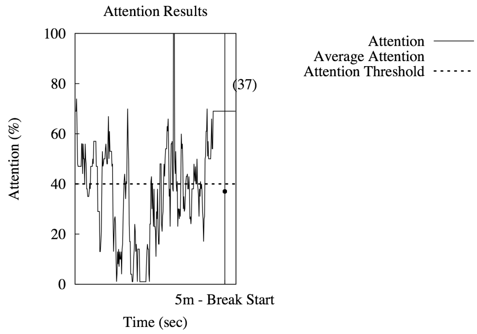


Figure 5.25: Attention results of the user L

documentation provided for them, we were not prepared for dealing with this types of particularities.

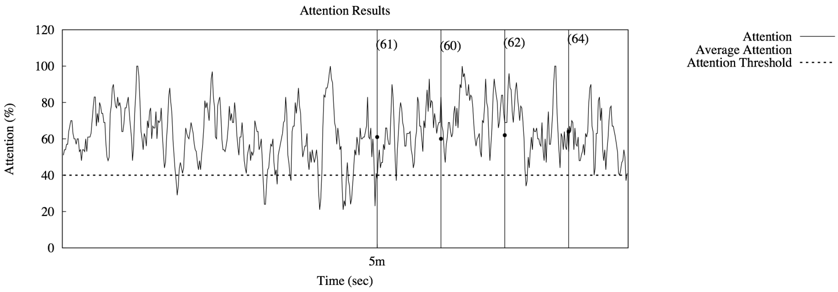


Figure 5.26: Attention results of the user M

5.3 Synthesis and discussion

The evaluation of the system in terms of functional tests was crucial to take some implementation decisions more carefully.

The validation of capture of sensing data let us conclude the system required a Good signal quality level to properly collect the sensing data. This made us implement the system in a form that, to perform sensing analysis, it must achieve a minimum quantity of values collected under a Good signal quality (the so called valid values).

The tests show that in 17% of cases the system is not able to gather the minimum quantity of valid values required. 23% of the cases it only achieves that minimum 1 to 5 times, 20% of times it achieves it 6 to 10 times, 27% of the cases, the system was able to perform 11 to 15 decisions, and finally, in approximately 13% of cases, the system was able to perform 16 to 20 decisions.

In terms of performance, this values are considered low, since in most of the times the system is not able to perform its main operation.

Also, to launch the system, it is necessary to reach a Good signal quality first to assures the system is launched automatically collecting valid values. This affected the time required to launch the system and we tested that factor as a performance metric. The results are that 40% of times, the system tooks less than 20 seconds to achieve a Good signal quality. Also, in 13% of the cases the system times out before achieving the required level of signal quality. Finally, 20% of the times the system can't recognize the sensor device at all. This results are not good, but we cannot state that it's sensor fault. The sensor used for testing was old and can be not be updated or damaged.

The User Tests helped us to understand how the users perceived the system and how they would compare it with the traditional Pomodoro Technique.

The results of the user tests show that 53.3% users don't think the system detect when they needed a break.

In terms of utility for time management, there are more users (53.3%) classifying the traditional Pomodoro Technique as Four in a scale of One to Five, than users giving the same classification for the smart version of the Pomodoro Technique (33.3%). However, more users (26.7%) classify the smart version as Five, when only 6.7% gave this classification to the Traditional Pomodoro Technique. This means, 60% of users classify both of the techniques with an utility level of more than Four for time management.

In terms of improving study performance, there are more users (46.7%) classifying the traditional Pomodoro Technique as Four in a scale of One to Five, than users giving the same classification for the smart version of the Pomodoro Technique (33.3%). However, more users (20%) classify the smart version as Five, when only 6.7% gave this classification to the Traditional Pomodoro Technique. This means, in general, half of the users (53.3% for the traditional version and 53.4% to the smart version) classify both of the techniques with an utility level of more than Four.

Also, 66.7% of users consider the smart version of the Pomodoro Technique was more suitable for their time management during study and 80% of users consider the smart version of the Pomodoro Technique was more suitable for improving their study performance. This means that the majority of the users prefer the smart version of the Pomodoro Technique although they found the experiences with both the techniques were similar.

And finally, the answers about the intrusiveness show that 40% of users classify the system comfort of three in a scale of one to five, and 33.3% classify as two. This classification is very low, as made us conclude the system intrusiveness is really one limitation that must be improved.

Some results were inconclusive, since the concentration of human beings is a complex process and it is affected by many factors. To infer about the biological conditions in which the users performed the test we needed help of health professionals, what was not possible at the time of the tests.

6

Conclusion

6.1 Conclusions

WakeUpNet system aimed to be an innovative system, incorporating the most relevant factors that are proved to influence human's concentration and work productivity.

The research about the state of the art in the context of the project helped to identify the main functionalities that the system should support. The result was the conception of a system that combines time management with the user's natural work rhythm, suggestions for performance improvement, reports of the user performance over time and in comparison with the results of other users.

The solution proposed pretended to be a better version of the Pomodoro technique, and offer an innovative solution in two aspects: Biological sensing analysis balancing accuracy and intrusiveness, and flexible time managing according to the user personal needs.

The objective of becoming a non intrusive neurofeedback system was not met, due to the option to implement the sensors component with an existent sensing device in its commercial form. Although, the drawback of not taking this option would seriously compromise the other functionalities, since the implementation of an algorithm for the analysis of human brainwaves is complex and requires medical knowledge that is out of the scope of the project.

The system had to be implemented in a experimental version, and certainly has several aspects to be improved. It was possible to evaluate its performance and functionality, as well as the interest of users in using their functionalities. The tests don't show the most satisfactory results, but were determinant to identify what aspects should be improved, which are mainly related with the option the authors took for the sensors.

6.2 System Limitations and Future Work

The first improvement we suggest for WakeUpNet system is definitely the use of a less intrusive way for collecting sensing data. This could be met by implementing the sensors in an object of common use, like an headphones set. The intrusiveness of the system ended to be a requirement not met. Also, the use of Mindwave Mobile device as sensor, raised performance concerns related to the signal quality achieved for the system. It is important to find a way of minimizing this problem, using sensors more reliable. Another improvement related with the sensing data is the algorithm for processing the raw data relative to the brainwaves captured. To assure the system accuracy, different options of algorithms should be tested to score them and be sure the system uses the best option.

The system is missing the functionality of comparison of results with other users. This implies several alterations in the implementation of the Remote Server, that needs to accommodate much more data than it does now and have a better algorithm for retrieving the History reports requested. It also requires that more security concerns to be take in account since the personal data of every user is going to be stored in the cloud. The functionality of showing individual results of the user could also be improved, showing the progression of the user during several uses of WakeUpNet. The results displayed should also be displayed graphically, what would enrich the information displayed.

Another limitation of the system is the lack of Local Database. The consequence is a system that requires Internet connection to reach the Remote Database and gather all the data required. This is important to improve, since a system for study and work performance should not obey the user to be connected to the Internet if he prefers to be off line to minimize distractions.

The final limitation that needs to be solved is related with the user tests to evaluate the system. First, having one only sensor device, the tests had to be performed individually. Taking in account their duration, this limitations made the testing time much longer than expected. It was necessary the users to test the several phases of the system in the same day, and being the sensors so uncomfortable, most of them decided to interrupt the test sooner. Also, a bigger sample was required, and each user should have used the system more times. The test also needs to be performed in more controlled conditions. The system pretends to be adaptable to every user, however, the testing conditions need to be more unified to formulate accurate conclusions. By letting the users to perform the test for how long they wanted to study and whatever they wanted to study, our results ended to be more disperse than supposed. Finally, the analysis of the human concentration contemplates many different factors that need to identified and analyzed by health professionals. The authors were careful to take the biological conditions of users in account to the experience, but the know-how to interpret it was missing. The results were mainly inconclusive.

One improvement the authors also suggest to the system is to include, in future, a functionality for suggesting the user solutions to train their brain activity trough neurofeedback. This would maximize the integration of the sensor component to another user benefit. According to the research made in the state of the art of this project, we learn how the brain can be trained to achieve determined mind status, as in playful ways, such as neurofeedback games. The system could integrate the possibility to suggest the user activities to improve his results and performance trough neurofeedback techniques.

Finally, the research and tests ended to focus in the attention mental status. For future work, the status of meditation should be deepened, as well as other possible biological indicators of human mental activity, e.g., blink detection. The system could also take advantage of the smartphones embedded sensors to collect environmental data, like sound or light, and suggest the user to improve this conditions in order to improve their performance.

7

Bibliography

Bibliography

[1] A. I. Rughani and P. L. Penar, “Brain anatomy,” 2015.

[2] W. Sałabun, “Processing and spectral analysis of the raw eeg signal from the mindwave,” *Przeglad Elektrotechniczny*, vol. 90, no. 2, pp. 169–174, 2014.

[3] G. U. Navalyal and R. D. Gavas, “A dynamic attention assessment and enhancement tool using computer graphics,” *Human-centric Computing and Information Sciences*, vol. 4, no. 1, p. 11, Jul 2014. [Online]. Available: https://doi.org/10.1186/s13673-014-0011-0

[4] Z. Chen, M. Lin, F. Chen, N. D. Lane, G. Cardone, R. Wang, T. Li, Y. Chen, T. Choudhury, and A. T. Campbell, “Unobtrusive sleep monitoring using smartphones,” in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013 7th International Conference on*. IEEE, 2013, pp. 145–152.

[5] M. S. Saleab, M. A. A. El Ghany, R. M. Toma, and K. Hofmann, “Real-time sleep detection and warning system to ensure driver’s safety based on eeg,” in *Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2016 IEEE 19th International Symposium on*. IEEE, 2016, pp. 1–6.

[6] F. Cirillo, “The pomodoro technique (the pomodoro),” *Agile Processes in Software Engineering and*, vol. 54, no. 2, 2006.

[7] N. B. Moe, P. J. Ågerfalk *et al.*, *Agility across time and space: Implementing agile methods in global software projects*. Springer Science & Business Media, 2010.

[8] P. Alves and P. Ferreira, “Distributed context-aware systems,” Technical Report RT/22/2011, INESC-ID, Tech. Rep., 2011.

[9] D. C. Hammond, “What is neurofeedback?” *Journal of neurotherapy*, vol. 10, no. 4, pp. 25–36, 2007.

[10] R. D. Hunt, “The neurobiology of adhd,” 2006. [Online]. Available: http://www.medscape.com/viewarticle/41543_7

[11] A. Heraz and C. Frasson, “Predicting the three major dimensions of the learner’s emotions from brainwaves,” *International Journal of Computer Science*, vol. 2, no. 3, pp. 187–193, 2007.

[12] Q. Wang, O. Sourina, and M. K. Nguyen, “Eeg-based” serious” games design for medical applications,” in *Cyberworlds (cw), 2010 international conference on*. IEEE, 2010, pp. 270–276.

[13] I. S. Kotchetkov, B. Y. Hwang, G. Appelboom, C. P. Kellner, and E. S. Connolly Jr, “Brain-computer interfaces: military, neurosurgical, and ethical perspective,” *Neurosurgical focus*, vol. 28, no. 5, p. E25, 2010.

[14] P. A. Pour, T. Gulrez, O. AlZoubi, G. Gargiulo, and R. A. Calvo, “Brain-computer interface: Next generation thought controlled distributed video game development platform,” in *Computational Intelligence and Games, 2008. CIG’08. IEEE Symposium On*. IEEE, 2008, pp. 251–257.

[15] “Mindwave user guide,” 2011. [Online]. Available: http://developer.neurosky.com/docs/lib/exe/fetch.php?media=mindwave_user_guide_en.pdf

[16] N.-H. Liu, C.-Y. Chiang, and H.-C. Chu, “Recognizing the degree of human attention using eeg signals from mobile sensors,” *Sensors*, vol. 13, no. 8, pp. 10 273–10 286, 2013.

8

Appendixes

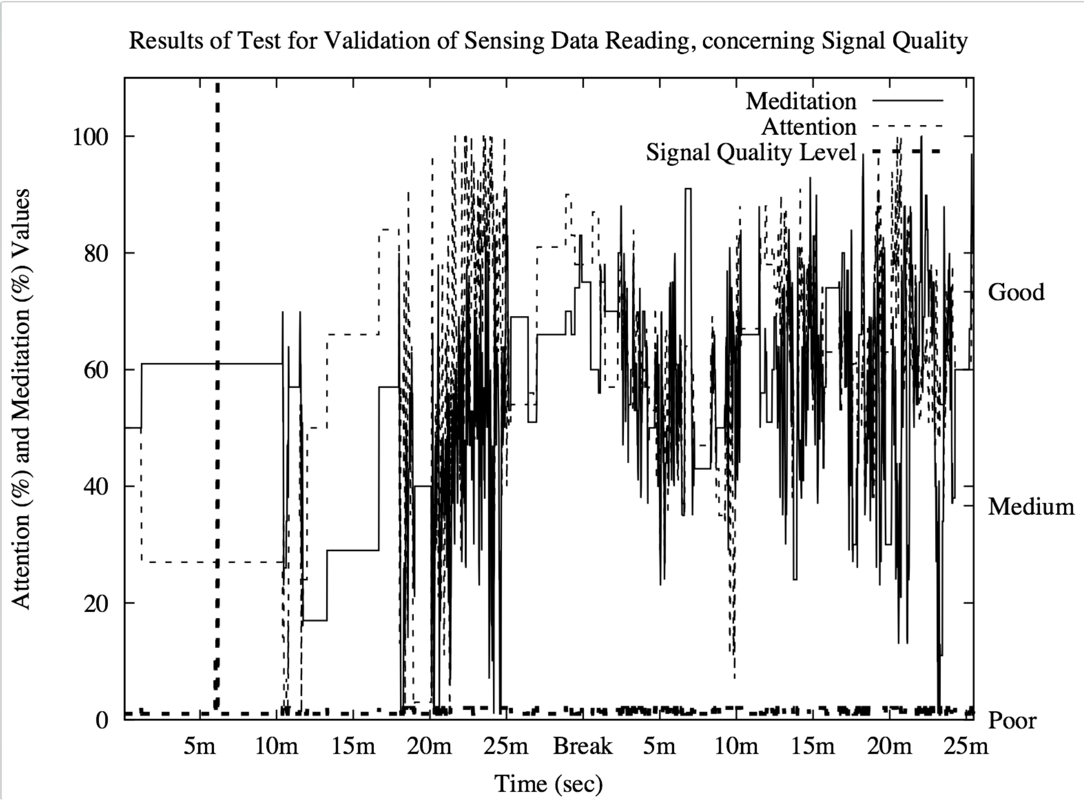
Appendix A: Table of comparison between the different Neurosky products

	Win	Mac	And	iOS	EEG Sensor	Sensor Type	Wireless Output	Receiver	Ref/Gnd Type	Battery	Other
MindWave Mobile	√	√	√	√	TGAM	Dry	Bluetooth	Standard BT (not included)	Earclip	1x AAA	
MindWave	√	√			TGAM	Dry	RF	USB RF receiver (included)	Earclip	1x AAA	
MindSet	√	√	√		TGAM	Dry	Bluetooth	Standard BT (USB dongle included)	Earphone pad	Built-in rechargable Li-Ion battery	Built-in stereo audio earphone speakers, Built-in microphone input
MindBand *	√	√	√		TGAM	Dry	Bluetooth	Standard BT (USB dongle included)	Earclip	Built-in rechargable Li-Ion battery	Headband form factor

For developers: All four devices transmit their output according to the same serial data protocol (ThinkGear Communications Protocol). Therefore, any app developed according to the NeuroSky Development Tools (ThinkGear SDKs) can be used with all four devices.

* For research and development purposes only

Appendix B: Results of functional test for validation of sensing data reading, concerning signal quality, for 50 minutes



Appendix C: Documentation for Test Users

WakeUpNet

WakeUpNet is a time management system that concerns user's focus status.

This system is composed by a brain sensor and a mobile application for Android OS. The brain sensor captures user's brain status in order to determine his level of attention. The mobile application tracks time spent and user's performance.

How WakeUpNet is going to be tested

WakeUpNet is going to be tested in three phases.

In the first phase, the system is only a mobile application of the classic Pomodoro Technique (see below) that also captures brain activity. The application produces a file with user's brain activity data during the work session.

This test phase takes 30 minutes long: During 25 minutes the user must study/work and after it, make a 5 minutes break. During both the study/work time as the break time the user must not put off the brain sensors.

In the second phase, the system will be able to adapt the duration of study/work to the user's attention status.

This test phase doesn't have a defined time period. The user can study/work for how long he wishes. The system will recommend him to make 5 minutes breaks everytime the user his lacking focus, and the user must act according to it.

In the third phase, the system will be able to adapt the duration of breaks to the user relaxation status as well.

In a first test session, the users will test, sequentially, the first and second phase, to compare them through a survey. Similarly, another day, a third test session will happen to test the third phase.

Test conditions

The users will be provided with the brain sensor and a smartphone to run the mobile application during the test sessions.

The user must use the system while doing some regular task of his study or work field and answer a survey about his experience with WakeUpNet. All the test results, answered surveys and any documentation with personal data will remain anonymous and will be used to statistic proposes only.

To ideal test conditions, the users must have similar characteristics. In order to accomplish that, the users must fill a selection survey to determine if they have the characteristics needed to an accurate system test. For example, regular drug consumers may have his focus status altered and therefore are not suitable for testing WakeUpNet. The selection survey will be anonymous as also.

All the selected users must be available to test the system all the four phases.

The Pomodoro Technique

The classic Pomodoro Technique is a time management technique that consists in working in 25 minute cycles with 5 minute breaks between them. The first phase of the test reproduces this technique.

The brain sensor and how to use it

The brain sensor required to use WakeUpNet system is a headband that the user must use just like headphones. It includes an earclip that must be clipped in user's left ear and an sensor tip that must be adjusted to stay in front of the user's forehead.

Test Instructions

First and Second phases:

1. Read this documentation carefully.
2. Answer the pre-test survey.
3. Set up WakeUpNet system by:
    a. Turn on the phone's Bluetooth
    b. Turn on the Brain Sensor and put it on;
    c. Launch WakeUpNet application
    d. Insert your Personal Identifier in the in the Login field
    e. Click the Connect Headset button and wait the message "connected" appear. You may need to wait some time untill the signal quality level is displayed.
    f. Adjust the headset until you reach a good signal quality and the Start button becomes available
    g. Click the Start button, that will launch the first phase test.
    h. Start studying/working during 25 minutes long, until WakeUpNet indicate you the 5 minute Break time will start
    i. When you receive the notification for break time, stop your study. You may do wherever you want, but you must keep using the brain sensor
    j. When the break time finishes, begins the test for phase two. You must do restart your study/work during for the time you wish. The system will recommend you to make more 5 minutes breaks according to your focus status.
4. Once you finish, please answer the post-test survey.

Appendix D: Selection Survey

WakeUpNet - Selection Survey

This survey will determine if the candidate is ideal for testing WakeUpNet system.
All questions are mandatory and your answers will be kept confidential and used for statistical proposes only.
The estimated time for completing the survey is less than 5 minutes.
Thank you!

*Required

1. **Please chose a Personal Identifier Example: your first name.** *

   The Personal Identifier is needed to track the test results over the multiple test phases and the respective survey answers. Please remember your ID so you can insert in the mobile application and surveys.

   ____

2. **Gender** *

   Mark only one oval.

   - ◯ Male
   - ◯ Female

3. **Age**

   Mark only one oval.

   - ◯ 18-25
   - ◯ 26-30
   - ◯ 31+
   - ◯ Other: ____________________

4. **Education Degree**

   Mark only one oval.

   - ◯ High School
   - ◯ Bachelor
   - ◯ Master
   - ◯ Doctorate
   - ◯ Other: ____________________

5. **Current occupation**

   Mark only one oval.

   - ◯ Student
   - ◯ Worker
   - ◯ Other: ____________________

6. **Education and training subject** *

7. **Do you often consume any type of drugs?**
Mark only one oval.

- Yes
- No

8. **Do you often consume alcoholic drinks?**
Mark only one oval.

- Yes
- No

Powered by
Google Forms

Appendix E: Pre-test Survey

WakeUpNet - Pre-test survey

This survey must be completed before the test is completed.
All questions are mandatory and your answers will be kept confidential and used for statistical proposes only.
The estimated time for completing the survey is less than 5 minutes.

*Required

Personal Data

This section of the questionnaire, plus the selection survey, will serve to characterise the population sample under analysis.

1. **Insert your Personal Identifier** *
   You must enter the same personal identifier you entered in the selection survey. This is needed to track your test results and survey answers, maintaining anonymity.

   ____

2. **What's the average time you are able to continuously study / work until you have to pause?** *
   Mark only one oval.
   - Less than 5 minutes
   - 5 to 25 minutes
   - 25 minutes to 1 hour
   - More than 1 hour

3. **What is the average duration of your study / work breaks?** *
   Mark only one oval.
   - Less than 5 minutes
   - 5 to 25 minutes
   - 25 minutes to 1 hour
   - More than 1 hour

Current user status

The following answers will help to determine what is your biological status during the test

4. **Number of sleep hours last night** *
   Mark only one oval.
   - Less than 4 hours
   - 4 to 6 hours
   - 6 to 8 hours
   - More than 8 hours

5. **What time you had your last meal?** *
Mark only one oval.

- Less than 1 hour
- 1 to 2 hours
- More than 2 hours

6. **How many coffees did you have today?** *
Mark only one oval.

- None
- 1
- 2
- 3
- More than 3

7. **Select the type of activity you will perform during the test** *
Tick all that apply.

- Reading Ex: reading slides, books, scientific articles, notes
- Writing Ex: writing notes, a scientific article, a report
- Practical exercises Ex: performing tests, performing calculations
- Programming
- Other: ______________________

8. **How would you classify your focus status today?** *
Mark only one oval.

	1	2	3	4	5	
Totally out of focus						Highly Concentrated

Appendix F: Post-test Survey

WakeUpNet - Post-Test Survey

This survey must be completed before the test is completed.
All questions are mandatory and your answers will be kept confidential and used for statistical proposes only.
The estimated time for completing the survey is less than 5 minutes.

*Required

1. **Insert your Personal Identifier** *
You must enter the same personal identifier you entered in the selection survey and mobile application. This is needed to track your test results and survey answers, maintaining anonymity.

___

Phase 1

Questions about the phase 1 (Classic Pomodoro Technique)

2. **Do you feel like the break time occurred in a right moment?** *
Mark only one oval.
- Yes, it was the perfect time to make a break
- No, it was too early, I could kept on studying/working longer
- No, it was too late, I was already tired of studying/working
- Other: ___

3. **How would you classify the utility of the WakeUpNet phase 1 for study/work time management?** *
Mark only one oval.

	1	2	3	4	5	
Totally useless						Totally usefull

4. **How would you classify the utility of the WakeUpNet phase 1 in terms of improving your study/work performance?** *
Mark only one oval.

	1	2	3	4	5	
Totally useless						Totally usefull

5. **Would you use an application with Phase 1 functionality in the future?** *
Mark only one oval.
- Yes
- No

Phase 2

Questions about the phase 2 (Interactive)

6. **Do you fell like the system detected when you needed a break?** *
Mark only one oval.

- Yes
- No

7. **How would you classify the utility of the information retrieved about your study/work session?**
Mark only one oval.

	1	2	3	4	5	
Totally Useless						Totally Usefull

8. **How would you classify the utility of the WakeUpNet phase 2 for study/work time management?** *
Mark only one oval.

	1	2	3	4	5	
Totally useless						Totally usefull

9. **How would you classify the utility of the WakeUpNet phase 2 in terms of improving your study/work performance?** *
Mark only one oval.

	1	2	3	4	5	
Totally useless						Totally usefull

10. **Would you use an application with Phase 2 functionality in the future?** *
Mark only one oval.

- Yes
- No

Phases comparison

Please answer the following questions about the comparison of the two phases tested.

11. **Which phase do you think is more suitable to study/work time management?** *
Mark only one oval.

- Phase 1
- Phase 2

12. **Which phase do you think is more suitable to improve your study/work performance over time?** *
Mark only one oval.

- Phase 1
- Phase 2

Overall System

Please answer the following questions about the overall system

13. **How do you classify the comfort of the use of the system** *

Mark only one oval.

	1	2	3	4	5	
Totally Uncomfortable	◯	◯	◯	◯	◯	Totally Confortable

Powered by Google Forms

Appendix G: Post-test Survey Answers

Figure 8.1: Results of the test with User A

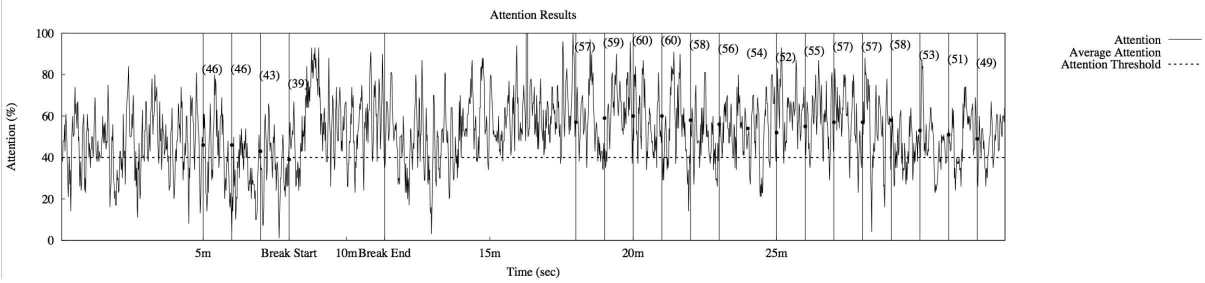


Figure 8.2: Results of the test with User B

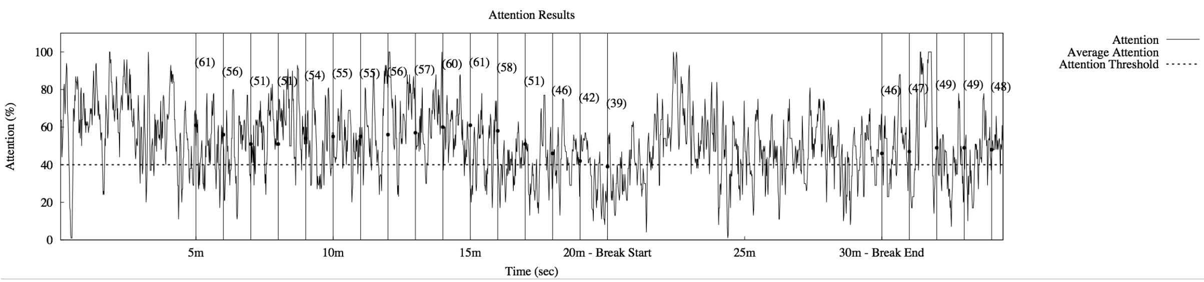


Figure 8.3: Results of the test with User C

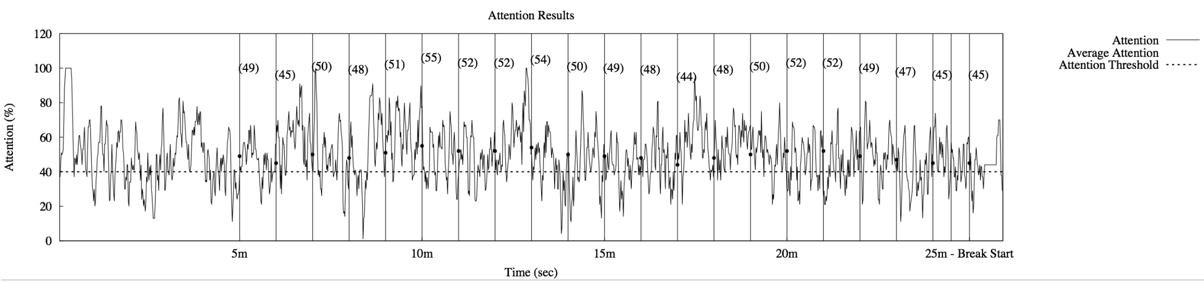


Figure 8.4: Results of the test with User E

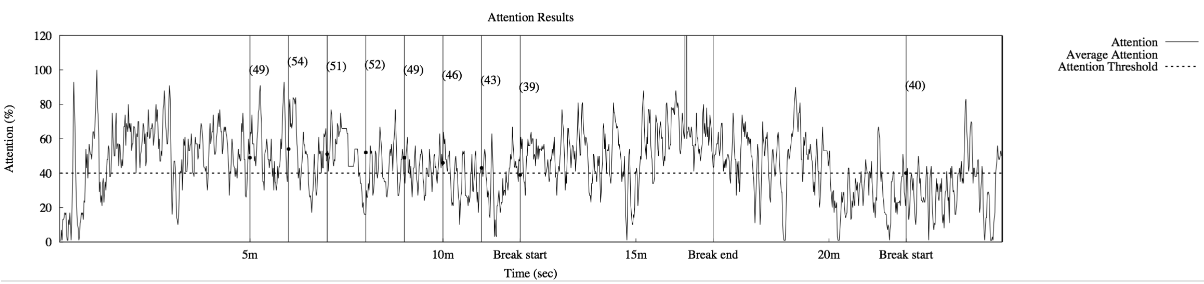


Figure 8.5: Results of the test with User F

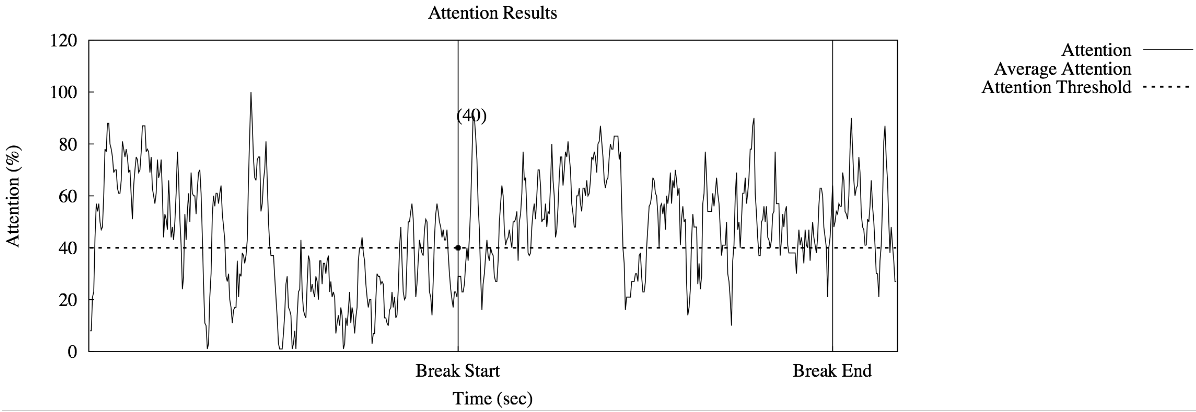


Figure 8.6: Results of the test with User G

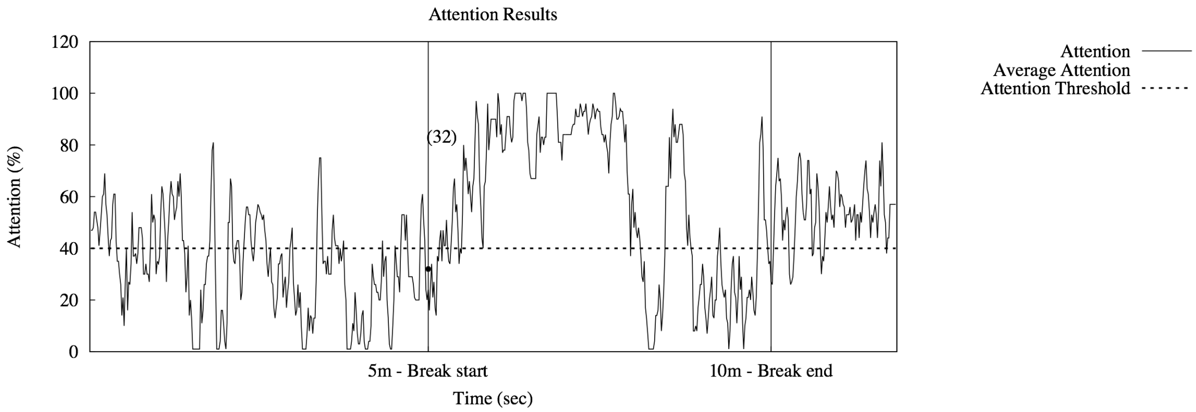


Figure 8.7: Results of the test with User H

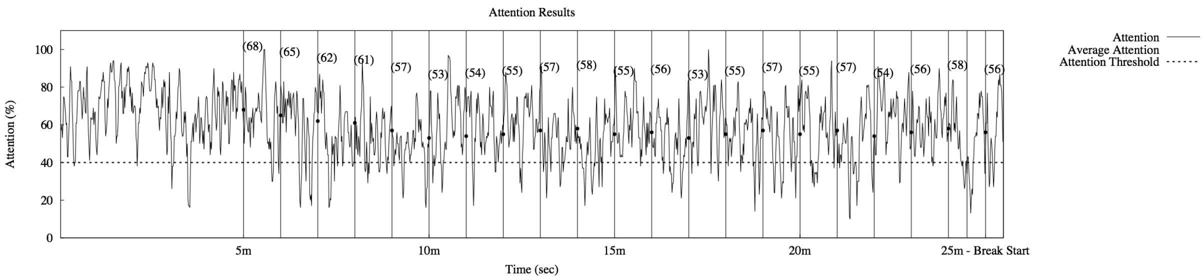


Figure 8.8: Results of the test with User I

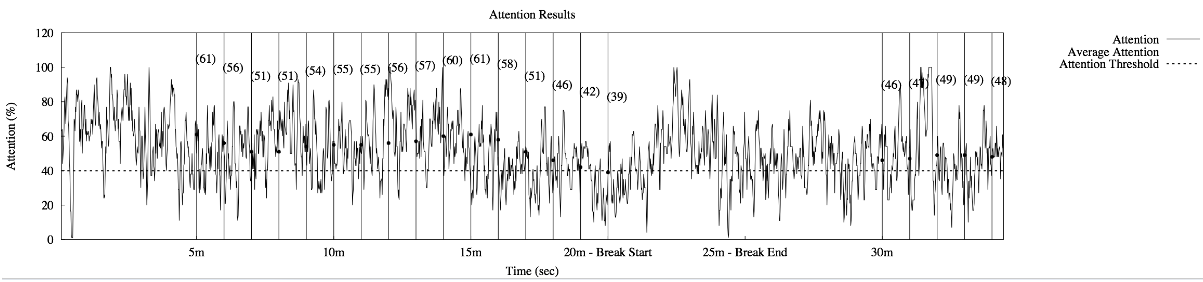


Figure 8.9: Results of the test with User J