Remote Monitoring System for Physiotherapy Clinics

Pedro Filipe Marques de Oliveira

Dissertação de Mestrado

Orientador na FEUP: Prof. Joaquim Gabriel Magalhães Mendes

Co-Orientador na FEUP: Prof. Miguel Pimenta Monteiro

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Faculdade de Engenharia da Universidade do Porto

Mestrado Integrado em Bioengenharia

Setembro de 2018

Remote Monitoring System for Physiotherapy Clinics

Abstract

The aging of the population and the lack of physical exercise have led to an increase in the needs of Rehabilitation services, namely Physiotherapy, which ensure through adequate exercises, procedures and equipment utilization (ultrasound, electrotherapy, thermotherapy, etc.), physical, emotional and cognitive well-being.

In ambulatory Rehabilitation and Physiotherapy clinics, patients must perform in a systematic, continuous and repetitive manner a set of previously defined clinical treatments. During part of his treatment, the patient is performing therapeutic actions that imply that he is alone for some time, although supervised. However, at the end of each of these therapeutic acts, there is no system of automated warning to the clinician, causing the patient to potentially wait longer than necessary. This situation is highly burdensome to the efficient management of the clinic and painful to the patient, and may even lead to accidents.

Regarding the problems previously descriibed, it was developed a solution composed of three modules was developed: Remote module, placed near the patient in which the professional introduces the treatment time (equivalent to the current system); a remote module, consisting of an Android application for the professional to consult the status of the treatment and its times, with the possibility of being notified when it finishes and emergency cases; and a server module, which interfaces with the others to store in the database the information that flows within the system.

This solution allows to digitally connect the health professional, the patient's treatment area, the patient and the clinic. The system was designed to fit the existing layout and ensure scalability and expandability.

A prototype is currently being installed at the clinic of the CMM Group - Medical Centers and Rehabilitation, at the unit: CMM Centro Clínico FisioEstarreja.

Resumo

O envelhecimento da população e a falta de exercício físico têm originado um aumento das necessidades de serviços de Reabilitação, nomeadamente de Fisioterapia que garantam através de exercícios adequados, procedimentos e utilização de equipamentos (ultrassonoterapia, eletroterapia, termoterapia, etc) a manutenção das funcionalidades biomecânicas, bem-estar físico, emocional e cognitivo.

Nas clínicas de Reabilitação e Fisioterapia ambulatória, os pacientes devem executar de forma sistemática, contínua e repetitiva um conjunto de tratamentos clínicos previamente definidos. Durante parte do seu tratamento, o paciente está a realizar atos terapêuticos que implicam que esteja algum tempo só, embora supervisionado. Contudo, no final de cada um destes atos terapêuticos, não existe um sistema de aviso automático ao clínico, levando a que potencialmente o paciente permaneça em espera mais tempo do que o necessario. Esta situação é altamente gravosa para a gestão eficiente da clinica e penosa para o paciente, podendo inclusivé dar origem a acidentes.

Para solucionar este problema, desenvolveu-se nesta dissertação uma solução composta por três módulos: Módulo remoto, colocado próximo do paciente no qual o profissional introduz o tempo do tratamento (equivalente ao atual sistema); um módulo remoto, constituido por uma aplicação Android para que o profissional consulte o estado do tratamento e respetivos tempos, com possibilidade de ser notificado em casos do fim deste e emergência; e um módulo de servidor, em interface com os restantes para guardar em base de dados a informação que flui dentro do sistema.

Esta solução permite ligar digitalmente o profissional de saúde, o local de tratamento do paciente, o paciente e a clínica. O sistema foi desenhado de modo a adaptar-se ao *layout* existente e garantir a escalabilidade e expansibilidade.

Atualmente está em instalação um demonstrador na clínica do Grupo CMM – Centros Médicos e Reabilitação, na unidade: CMM Centro Clínico FisioEstarreja.

Acknowledgements

I would like first and foremost to thank my supervisor, Prof. Dr. Joaquim Gabriel, for his constant very helpful and very commendable supervision and assistance.

Furthermore, I would like to acknowledge my co-supervisor, Prof. Dr. Miguel Pimenta Monteiro, for all the advices, technical guidance and availability.

Additionally, I would to show my appraisement to technician Pedro Alves, for all his assistance in circuit printing phase and constructive ideas.

I also want to express my acknowledgment to all the professionals, friends, colleagues and family that supported all this time, you understood it in the worst moments and celebrated it in the best. Without you, it would have been almost impossible.

I would like to acknowledge CMM – "Centros Médicos de Reabilitação" company and its CEO and Founder Gabriel Martins Costa (Physical Therapist) for the opportunity.

Last but not the least, I thank my family, especially my parents, for believing since the first year when, looking in the eyes, I told them that I would achieve it with distinction. They, as well as my closest friends, supported me surpassing difficulties and achieving goals keeping me motivated and showing their pride throughout the last five years.

Contents

1 Introduction 1
1.1 Physiotherapy Expenditures and Demography Aging 2
1.2 Project Framework and Motivation 6
1.3 Project Goals 7
1.4 The Current Clinical Problem 8
1.5 Dissertation Outline 9
2 State of Art 11
2.1 Other Similar Systems 11
2.2 Software 18
2.3 Android 18
2.4 Server 19
2.5 Hardware 24
3 The Proposed Solution 27
3.1 Remote Module 28
3.1.1 Microcontroller 30
3.1.2 7-Segment Display Choice 33
3.1.3 Enclosing Box – 3D Printing 33
3.1.4 PCB Design 34
3.2 Server Module 36
3.2.1 MQTT Tree Structure 37
3.2.2 Database 37
3.2.3 Mobile Android Application 38
4 Developed Solution 39
4.1 System Interface Description 39
4.2 Overall System Dataflow 42
4.2.1 Patient Treatment Tracking 42
4.2.2 Connect and Disconnect Tracking 44
5 Demonstration 47
5.1 Remote Module 47
5.1.1 Prototype 47
5.1.2 Printed Circuit Board 48
5.1.3 NodeMCU Programming 49
5.1.4 Remote Module State Sequence 50
5.2 Professional Module – Android 52
5.2.1 Application Main Features 52
5.3 Server Module 55
5.3.1 SQLite Database 56
5.3.2 HTTP WebPage 56

6 Conclusions and Future Developments 61
6.1 Conclusions 61
6.2 Future Developments 62
References 63
ANEXO A: OKW's Commercial Box External Measures 67

Abbreviations and Symbols

List of abbreviations:

ADT	Android Development Tools
AJAX	Asynchronous JavaScript And XML
APFISIO	Associação Portuguesa de Fisioterapeutas
CAPTE	Commission on Accreditation in Physical Therapy Education
CDN	National Board of Directors
CEO	Chief Executive Officer
CICA	Centro de Informática Prof. Correia de Araújo
CRM	Customer Relationship Management
DNS	Domain Name System
EAGLE	Easily Applicable Graphical Layout Editor
EDA	Electronic Design Automation
FEUP	Faculdade de Engenharia da Universidade do Porto
FTP	File Transfer Protocol
GPIO	General Purpose Input/Output
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated development environment
IIS	Internet Information Services
iOS	iOperating System
IoT	Internet of Things
ISR	Interrupt Srvice Routine
LED	Light Emitting Diode
LWT	Last Will and Testament
M2M	Machine-to-Machine
MQTT	Message Queuing Telemetry Transport
PC	Personal Computer
PCB	Printed Circuit Board
PHP	Hypertext Preprocessor
PPACA	Patient Protection and Affordable Care Act - Obamacare

PT	Professional Therapists
QoS	Quality of Service
RGB	Red Green Blue
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol and Internet Protocol
UI	User Interface
UML	Unified Modeling Language
USA	United States of America
WWW	World Wide Web
XML	Extensible Markup Language

List of Figures

Figure 1.1 – U.S Outpatient rehabilitation expenditures increase since early 2000's until today [4]. 3

Figure 1.2 - U.S. Professional Therapists' graduates. Number of actual and projected graduates, 2981–2019 [7]. 3

Figure 1.3 - Population resident in Portugal, 1 January 1980 to 2060 (estimates and projections) [8]. 4

Figure 1.4 - Percentage distribution of population by age groups - central scenario, 1980-2060 (estimates and projections) [8]. 5

Figure 1.5 - Age pyramid of the population. (I) January 1, 2008 and 2030; (II) January 1, 2008 and 2060 [8]. 6

Figure 2.1 - PhysioTrack Software browser interface [9]. 12

Figure 2.2 - PhysioTrack Software smartphone interface [9]. 12

Figure 2.3 - *TrackActive* smartphone interface [10]. 13

Figure 2.4 - Smartphone and laptop with patient data chart for professional therapist analysis [9]. 14

Figure 2.5 - Laptop, iOS, smartphone, tablet interface for exercise tracking and Patient-Professional communication [11]. 14

Figure 2.6 - AccelerWare Browser Interface with appointments [12]. 15

Figure 2.7 - Spok's role is to connect the current nursery machines the professionals to the their communication devices [13]. 16

Figure 2.8 - Examples of medical communication devices for notification and their interfaces [13]. 16

Figure 2.9 - ALCAD's system simplified workflow [14]. 17

Figure 2.10 - Android Studio IDE interface [16]. 19

Figure 2.11 – Android Empty Activity with "Hello World" TextView [19]. 20

Figure 2.12 - MQTT architecture, Clients and Broker [24]. 22

Figure 2.13 - AJAX technology working server-client process [28]. 23

Figure 2.14 - Sequence of processes once the interrupt routine is triggered. The Line "Interrupt Request" represents the trigger timing [33]. 24

Figure 2.15 - Simple overview of the I^2C wiring [35]. 25

Figure 3.1 - Detailed modular distribution of the presented system and implemented communication protocols. 27

Figure 3.2 - Remote Module's schematic circuit. 28

Figure 3.3 - Remote Module's Fritzing drawing. 29

Figure 3.4 - D1 Mini ESP8266 Module pinout [42]. 32

Figure 3.5 - Example of 7-segment display. 33

Figure 3.6 - Alphanumeric digital display [45]. 33

Figure 3.7 - OKW's enclosure box (external/general appearance) [46]. 34

Figure 3.8 - OKW's enclosure box (L size internal platine measures) [46]. 34

Figure 3.9 - Remote Module's board structure. 36

Figure 3.10 - Database Relational Model. 37

Figure 4.1 - Detailed hardware and software distribution of the presented system with implemented communication protocols. 39

Figure 4.2 - Remote Module State Diagram of a single patient treatment. 42

Figure 4.3 - System's example treatment status dataflow synchronized with column data insertion. The given example is the same as the one given in the text. 44

Figure 4.4 - System's example disconnections dataflow synchronized with column data insertion. The given example is the same as the one given in the text. 45

Figure 5.1 – Remote Module images closed: a) TOP perspective; b) UP perspective. 47

Figure 5.2 - Remote Module closed turned on with peripheral legend. 48

Figure 5.3 – Representative images of the PCB: a) TOP perspective; b) BOTTOM perspective. 49

Figure 5.4 – Sequence of Remote Module States for Professional-Module Actions. (I) Rest state; (II) Setting time while in rest state; (III) and (IV) counting down time; (V) Treatment time ended; (VI) Professional/Emergency call. 51

Figure 5.5- Application's UML Use Case diagram. 52

Figure 5.6 - Application's user interface flow. (I) Activities' flow; (II) fragments' flow. 54

Figure 5.7 - Active fragment, Treatment and synchronized update. 54

Figure 5.8 - Both Application interface on Emergency Call . (I) Emergency acknowledge by notification pressing; (II) Emergency acknowledge with application maximized. 55

Figure 5.9 - CICA's Server files. 55

Figure 5.10 – Example of treatment's data stored in database. 56

Figure 5.11 - Example of microcontroller connection data stored in database. 56

Figure 5.12 - System's Webpage with empty (Green) and waiting professional acknowledge (Yellow). 57

Figure 5.13 - System's Webpage with clinic's map. Scrolled view from Figure 5.12. 57

Figure 5.14 – Server-Client AJAX Simplified communication. 58

Figure 5.15 – Activity diagram of Browser-Server AJAX communication. 59

List of Tables

Table 1 - Microcontroller comparison 31

1 Introduction

This project was proposed by "CMM – Centros Médicos e de Reabilitação" (Aveiro, Portugal) to fight the lack of performance of the clinics. CMM is a company mainly focused in outpatient Rehab, specially Physiotherapy with eleven medical centers spread over Portugal, and processing more than 5,000 patients daily.

Currently, the Physiotherapy clinics' workflow is usually not optimized. Among others, the health professionals take care of several patients at a time, being unfocused and dispersed by constant machine buzzing, which leads to stress, disorganization and lack of clinics' activity control.

Given that, it is proposed a centralized system to simultaneously manage the group of patients, their treatments as well as control their evolution between different states. This system enables specific professional notification, independent treatment tracking the information for future management department's analysis and resource optimization.

According to the site Reabilitech, in 2015, it was registered the most significant growth of approximately 43% of the demand for these physiotherapy professionals in Brazilian health plans [2].

Regarding Portugal's scenario, in the last half-century, it is seen the development of physiotherapy as the third largest in the healthcare area and the most representative profession in rehabilitation. According to Emanuel Vital, chair of Associação Portuguesa de Fisioterapeutas (APFISIO) National Board of Directors (CDN), the profession has grown exponentially in last ten years, from little more than 2,000 professionals at the beginning of this century to the current 11,000 physiotherapists.

Another topic emphasized by chairman Emanuel Vital is that the future is being built today with notorious attention being given by physiotherapy world aiming an increasing investment in scientific research in the next years [3].

1.1 Physiotherapy Expenditures and Demography Aging

Outpatient rehabilitation patient is someone who is not hospitalized overnight but visits a hospital, clinic or associated facility for diagnosis or treatment. In United States of America (USA), this type of rehabilitation is a $29.6 billion industry, which is expected to grow 7% annually. In 2018 is reaching approximately 90%, of all outpatient rehabilitation spending, as illustrated in Figure 1.1 [4], [5]. This market is highly fragmented with the largest 50 competitors comprising less than 25% of the market.

Considering recurring society analysis, there are recognized numerous and positive factors driving long-term growth on patient population:

- Aging population;
- Unhealthy youth lifestyle trends;
- New government regulations increasing patient access to physical therapy, taking as an example the Patient Protection and Affordable Care Act (PPACA), also noun as the Obamacare [6];
- Outpatient rehabilitation, comparatively to surgery or hospitalization, is significantly less costly, but with similar clinical effectiveness.

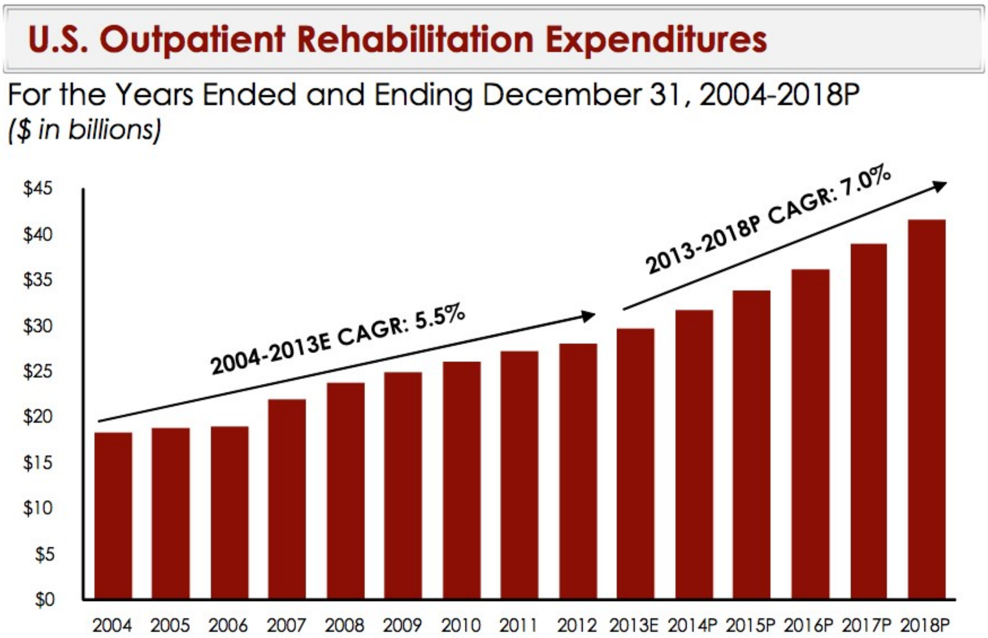
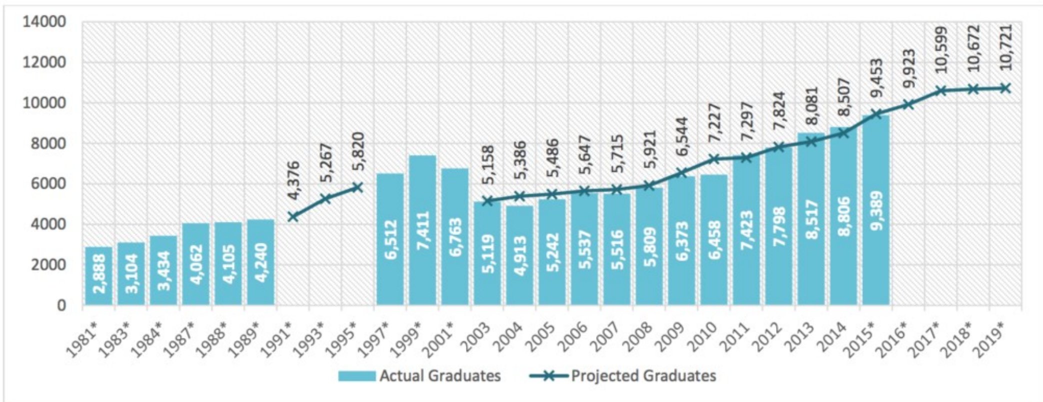


Figure 1.1 – U.S Outpatient rehabilitation expenditures increase since early 2000's until today [4].

According to the Bureau of Labor and Statistics, it is expected an increment of 25% of the physical therapist jobs between 2016 and 2026. As illustrated in Figure 1.2, the number of physical therapist jobs reached a total of 239,000 in 2016. With that number growing at a rate of about 4% as the Professional Therapists (PT) schools pump out nearly 10,000 graduates each year [7].



* Indicates years where one or the other, not both, graduate numbers were available.

Figure 1.2 - U.S. Professional Therapists' graduates. Number of actual and projected graduates, 2981–2019 [7].

According to the Commission on Accreditation in Physical Therapy Education (CAPTE), new PTs graduates are expected to reach record highs of more than 10,600 professionals in 2018.

By year, in the USA, approximately, 10,000 professionals are being graduated from the PT program, becoming this job market more competitive each year [7].

It is expected that in the next 50 years the population of Portuguese residents will remain in the 10 million. However, the trend of demographic aging will continue, projecting that in 2060 there are about three elderly people per each young person.

Figure 1.3 retracts several aging scenarios that will be further analyzed.

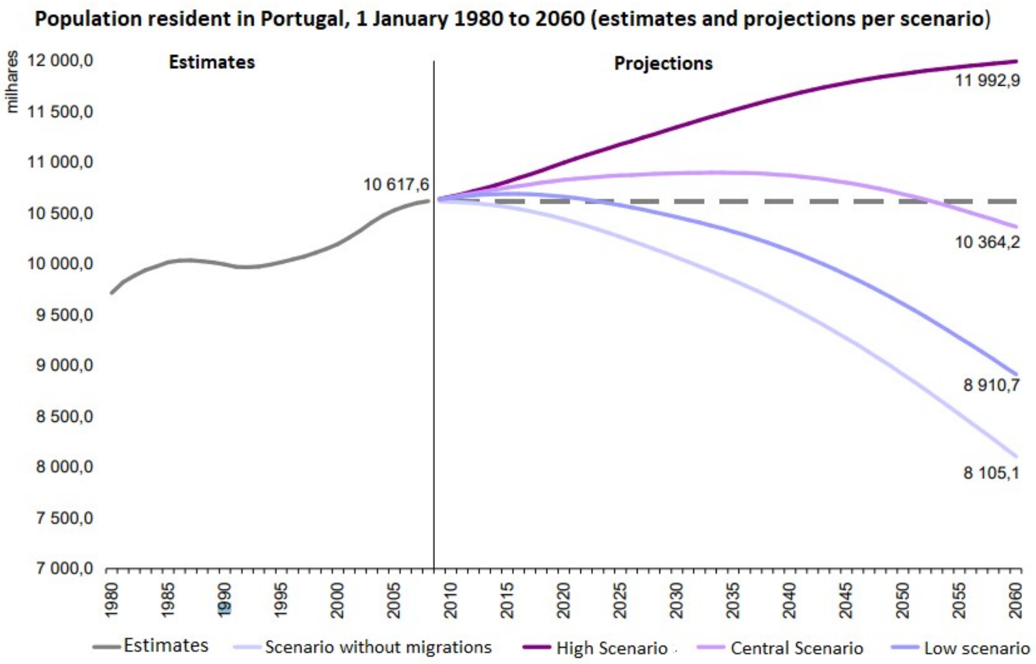


Figure 1.3 - Population resident in Portugal, 1 January 1980 to 2060 (estimates and projections) [8].

According to the results obtained in the central scenario, the resident population in Portugal will continue to increase to 2034, reaching 10,898.7 thousand individuals, the year from which the population will decrease [8].

According to Figure 1.4, in any of the scenarios considered, it is expected a reduction in the proportion of young people (under 15 years) from 15.3% in 2008 to 11.9% in 2060. The same scenario is predicted for the working-age population (from 67.2% in 2008 to 55.7% in the central scenario). Still concerning this age gap, in any of the scenarios considered, the results indicate a reduction of the proportions for the age groups "15 to 24", "25 to 39" and "40 to 54", as opposed to increasing the proportion of the age group "55 to 64”.

In opposition, the elderly population, aged 65 or over is expected to almost double, increasing in the central scenario from 17.4% in 2008 to 32.3% in 2060 [8].

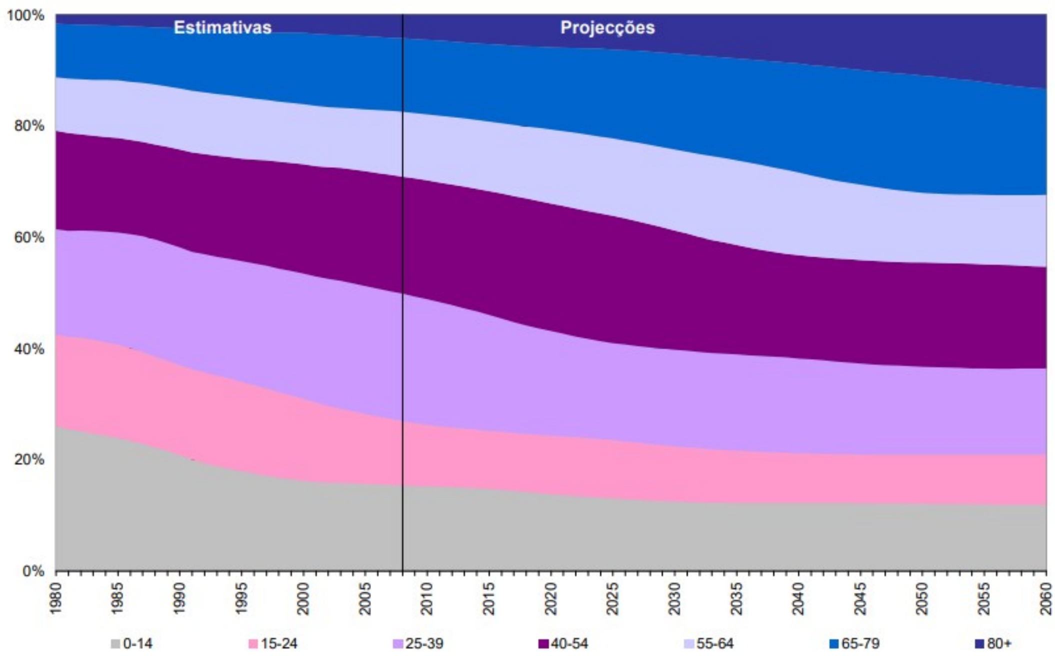


Figure 1.4 - Percentage distribution of population by age groups - central scenario, 1980-2060 (estimates and projections) [8].

Regarding the Figure 1.5, age pyramids present different forms for each scenario, both in 2030 and 2060. By 2030, the effects of the different scenarios are particularly observable in the active and young ages. In 2060, the impact of the different scenarios will be visible at all ages, albeit more sharply at the young and dynamic ages, showing the effects of aging populations [8].

Each pyramid has two sizes. The right one corresponds to female statistics, and the left one corresponds to male statistics.

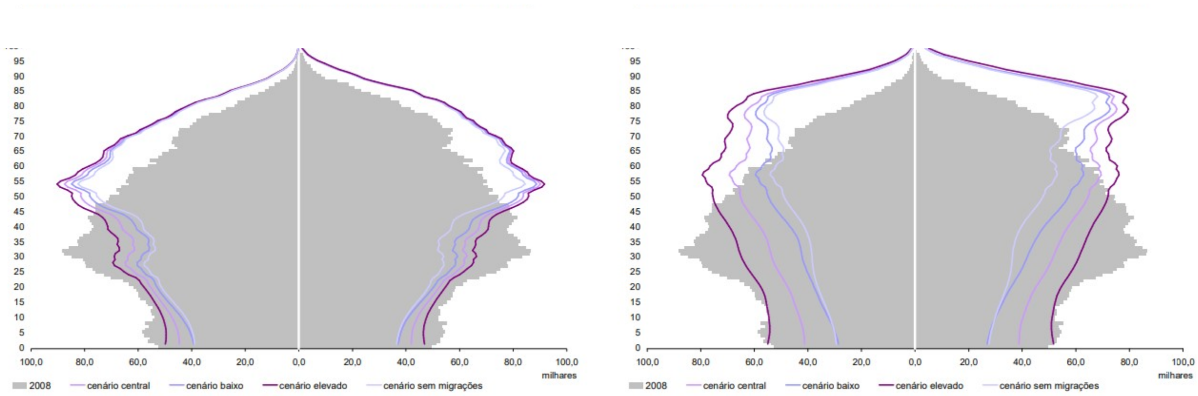


Figure 1.5 - Age pyramid of the population. (I) January 1, 2008 and 2030; (II) January 1, 2008 and 2060 [8].

Considering the decrease in the young population, simultaneously with the increase in the elderly population, the index of aging population will increase. Focusing central scenario, in the year 2060, there will be 271 elderly people for every 100 young people, more than double the projected value for 2009 (116 elderly people per 100 young people).

The verification of positive migratory balances will not be sufficient to mitigate demographic aging, even if there is a more significant contribution of the fertility component. The migration balances and higher fertility levels, as well as the rising of the average life expectancy, will only allow the population-aging [8].

To attend the increasing demand for better and more efficient healthcare conditions either for patient, clinic and professional. A research investment opens the door for integrating other areas of knowledge, like engineering.

1.2 Project Framework and Motivation

In this work, it is proposed an auxiliary clinical system to increase the performance of the healthcare professionals, improving the digitization of daily procedures, and consequently improving the care quality and security of the patients, focusing the development of a product for future market implementation in partnership with the company.

From the company’s perspective, the present project aims a scalable and duplicable prototype for real testing in clinical environment with user-friendly mobile interface and a simple device interaction near the client for system digitalization and optimization purposes. The aim of this project is the development of a designed system in order to facilitate specialist's daily routine while having minimum impact on daily workflow, by keeping the professional's number of inputs very similar to the non-digital current system.

This project fits the company's self-commitment to continuously improve all processes of quality health care training while strengthening the relationship between the healthcare provider and the client. The integration of technology as an innovation inside a traditional medical field is an audacious complement for clinic method and organization aiming the global client satisfaction.

1.3 Project Goals

This system was developed in partnership with the company "CMM – Centros Médicos e Reabilitação", fulfilling the initial requirements proposed as well as the "on-the-fly" changes specified in personal meetings during the development of the prototype. The objective of the company is clearly identified as providing real-time information about patient and treatment, compartment occupation information display and digitalized timed schedules.

These features should be attended while maintaining the overall system low-cost, robust, scalable and modular so that it can fit several types of clinics' activity and follow its increase.

In this way, this project encompasses a component of hardware and software development specific for the problem faced. The whole system is composed by three modules: Remote Module, Server Module and Professional Module.

The Remote Module is a microcontroller with *Wi-Fi* connection and 7-segment display located in each place where monitoring is needed such as rooms and devices; The Server Module is a real-time server responsible for supporting the system information with database; The Professional Module is an Android Application with the purpose of displaying the information from the devices of interest, as well as provide a simple user interaction.

1.4 The Current Clinical Problem

In physiotherapy clinics, each health professional is responsible for several patients, in different treatment stages that may require, or not, the presence of the professional. The treatments can be performed in individual cabins, gym exercises or washing baths. Currently, these therapeutics exercises are programmed directly on the equipment, or on nearby analog alarm clock devices that ring when defined time ends.

In particular, at CMM, the current solution has no light indication of communication ability, being similar to the countdown timer, as shown in Figure 3.1.

Figure 3.1 - Example of Extra Large Display Digital Timer [36].

Thus, the current operation mode causes constant buzzing in the working area, a lack of patient tracking, lack of central information of clinical timings and doesn't allow a log of the activities.

This optimization is mainly based on a designed user-friendly interface capable of providing all the information required by the specialist as well as the ability to store relevant clinical information for further human and facility resources optimization.

The proposed solution uses a smartphone as the remote communication interface to the system. In addition to being a device well known by all clinical staff, it eliminates "not informative" buzzing, while lowering the confusion with the clinician-device assignment and adds the history consultation feature.

Taking a closer look to the local compartment module, the developed idea is keeping the device-professional interaction as similar as it is on the current digital device, where he enters de time with two buttons, minutes and seconds incremental. Using this as reference, it is replicated the same idea as the double button setup for time setting and another two buttons, the "OK" button for acknowledges and timer starting and the emergency button for the real-time call mentioned.

1.5 Dissertation Outline

This document is separated into **six** chapters.

- **Chapter 1 - Introduction**

The **present** chapter describes the demographic aging problem, project motivation, the aim of the developed system and the contributions of this dissertation.

- **Chapter 2 – State of Art**

The **second** chapter, a State of Art review, describes other similar systems for patient-professional communication inside or outside, it is described some relevant features about Android programming and protocols used. It also explains the microcontroller interrupts routines used, the I2C protocol and the Node.JS server framework.

In this chapter, it is also outlined the clinic problem as constant buzzing, lack of digitalization and the need for optimization of human and facilities resources.

- **Chapter 3 – The Proposed Solution**

The results are presented in the **third** chapter, subdivided by system module.

- **Chapter 4 – Developed Solution**

The **fouth** chapter presents, with examples, the system's modules interfaces, describing which and why module and communication protocol are used.

Also in this chapter**, t**he system in-depth dataflow analysis is discussed, analysing more technical aspects of messages, server, microcontrollers and single module programming.

- **Chapter 5 - Demonstration**

The results are discussed in the **fifth** chapter, where the final prototype system is showed.

- **Chapter 6 – Conclusion and Future Development**

The **sixth** chapter of this document is the Conclusion, where the conclusions are established and the future system development are outlined.

2 State of Art

In this chapter, it will be addressed the current similar solutions to the proposed system, technology, development software, communication protocols and software background for system design and conception.

2.1 Other Similar Systems

In the market, there are several similar systems for digitalization of hospitals, nursery and physiotherapy clinics, which are presented hereafter.

The most notorious are Physiotrack [9], TrackActive [10], MyphysioApp [11] that were launched to strengthen the communication between patient and physiotherapist outside clinic buildings. The professional prescribes the exercises between appointments according to patients' feedback through the mobile application while the system keeps the history of exercises and patient responses through performance quiz.

These systems are an impactful advance in patient daily life, but it is meaningless for internal activity optimization.

Physiotrack Software

PhysioTrack is a web-based software that works on any modern web browser. Patients can view their exercises on their smartphones and send feedback to the professional. Handouts can be generated electronically, on paper or by mobile app. Generally, Physiotrack is an "exercise prescription software".

This software provides a free patient mobile phone application available for Android and iOperating System (iOS*)*, that receives exercise prescriptions, appointment reminders, clinical instructions and marketing communications. The system presents a complete range of exercise photos and patient instructions (more than 2,000 exercise images) matching the physiotherapy tools/software available [9].

In brief, the software allows unlimited communication between patients and the health professional. Figure 2.1 and Figure 2.2 illustrate the designed interface proposed where the patient and the clinician can interact.

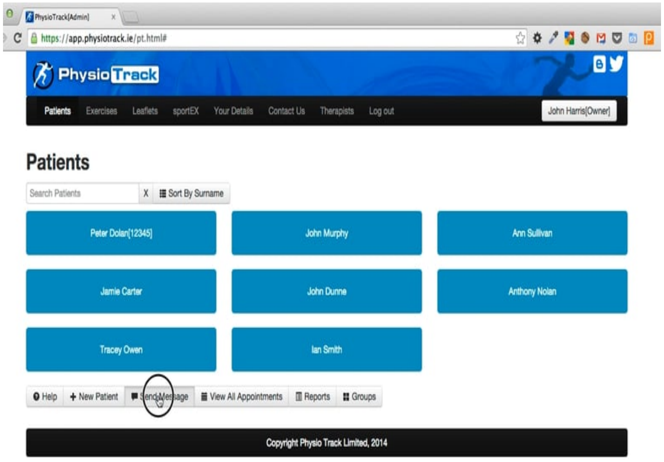


Figure 2.1 - PhysioTrack Software browser interface [9].

Figure 2.2 - PhysioTrack Software smartphone interface [9].

TrackActive Software

TrackActive is an exercise prescription software, customizable and evidence-based. Patients can access their rehabilitation programs via web login or by mobile app. The system has high-quality exercises for all body areas and injuries with hundreds of evidence-based exercises complete with videos provided by the developers and the community completing the database.

The system enables the professional to save time so he can supervise other patients. The application, Figure 2.3, is designed so that the professional can create a program and send it directly to the patient in under 5 minutes. The same characteristic is achieved on the Personal Computer (PC), laptop or tablet platform, Figure 2.4 [10].

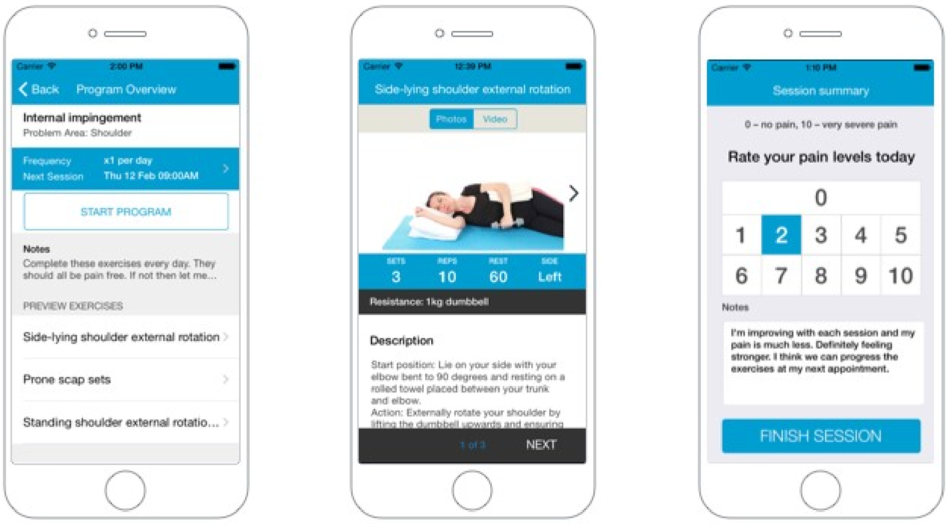


Figure 2.3 - *TrackActive* smartphone interface [10].

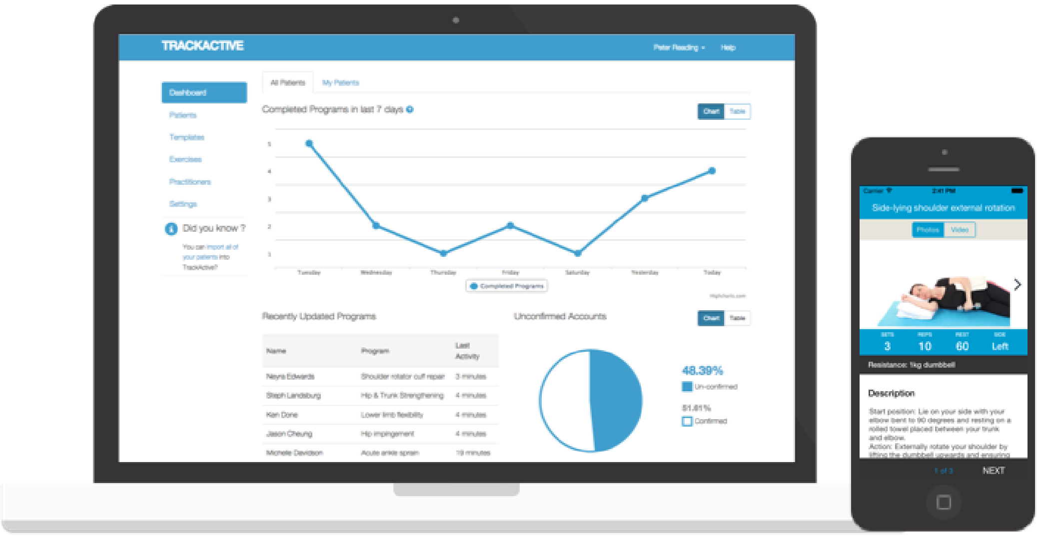


Figure 2.4 - Smartphone and laptop with patient data chart for professional therapist analysis [9].

MyPhysioApp Software

MyPhysioApp was created in 2012 by Active Bodies (Adelaide, Australia). It is used in physiotherapy practices throughout the world to deliver specific exercise programs that compliment in-clinic treatment. It provides exercise videos and instructions directly to the phone, reading information, links, supplying the client with practice information and special offers.

As illustrated in Figure 2.5, the system runs in mobile phone (iOS and Android), tablet, laptop, and can be used either by the professional to insert the exercises, or by the patient to see them and provide feedback [11].

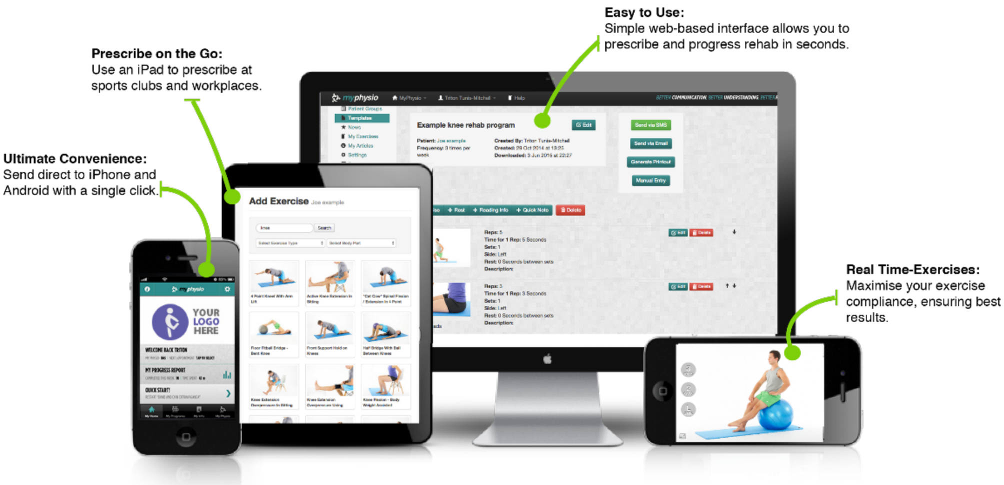


Figure 2.5 - Laptop, iOS, smartphone, tablet interface for exercise tracking and Patient-Professional communication [11].

AccelerWare Software

AccelerWare developed the Customer Relationship Management (CRM) system [12]. This system considers the database of clients, staff, programs, resources, among others. It makes contact and communication very easy, allowing the professional to manage recurring appointments in a series, automatically remind patients of consultation times, upload patient details and other similar functions quickly and easily, Figure 2.6 [12].

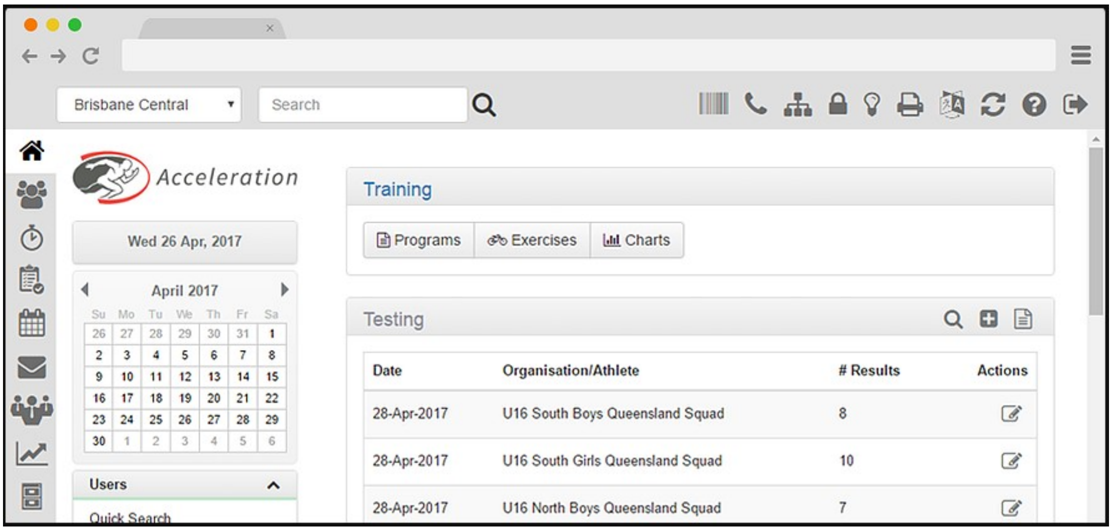


Figure 2.6 - AccelerWare Browser Interface with appointments [12].

Despite all the advantages, this system would be an overkill for statistics and communication without solving the facilities and staff performance issue.

Spok Clinical Alerting Notification

A different type of information systems needed in clinics are the alerting systems like Spok Clinical Alerting Notification. This system centralizes all alerts notifications among compartment in a hospital and redirects it to professional's pagers, smartphone or tablets. It is an integrated system with hospitals' machines where all the alerts are sent to the responsible professional, Figure 2.7.

Although this system has the centralization feature, it presents a limitation in the interface, once nursery machines' alerts should be compatible with the notification system. This factor leads to too many expenses in restructuring the whole clinical hardware, Figure 2.8 [13].

Figure 2.7 - Spok's role is to connect the current nursery machines the professionals to the their communication devices [13].

Figure 2.8 - Examples of medical communication devices for notification and their interfaces [13].

The Patient-Nurse System by ALCAD

A notification system that enables the workflow and is already in the market is The Patient-Nurse System made by ALCAD Electronics S.L (Gipuzkoa, Spain). It is a communication solution and nursing management that employs Internet Protocol technology (IP) and that works both wired or wireless. The solution consists of a set of push buttons and wireless sensors that allow a better knowledge of the state of the patient/resident. The information is transmitted to the software of management (monitoring system) that allows to record all events and extract reports from the alarms and the assistance work.

Despite several sensors and professional interfaces, it does not solve the problem of the disorganization, since none of the health professionals has a centralized notification device. To access that information, the professional should move to a specific place in the clinic. Figure 2.9 presents a simplified workflow when ALCAD system is installed [14].

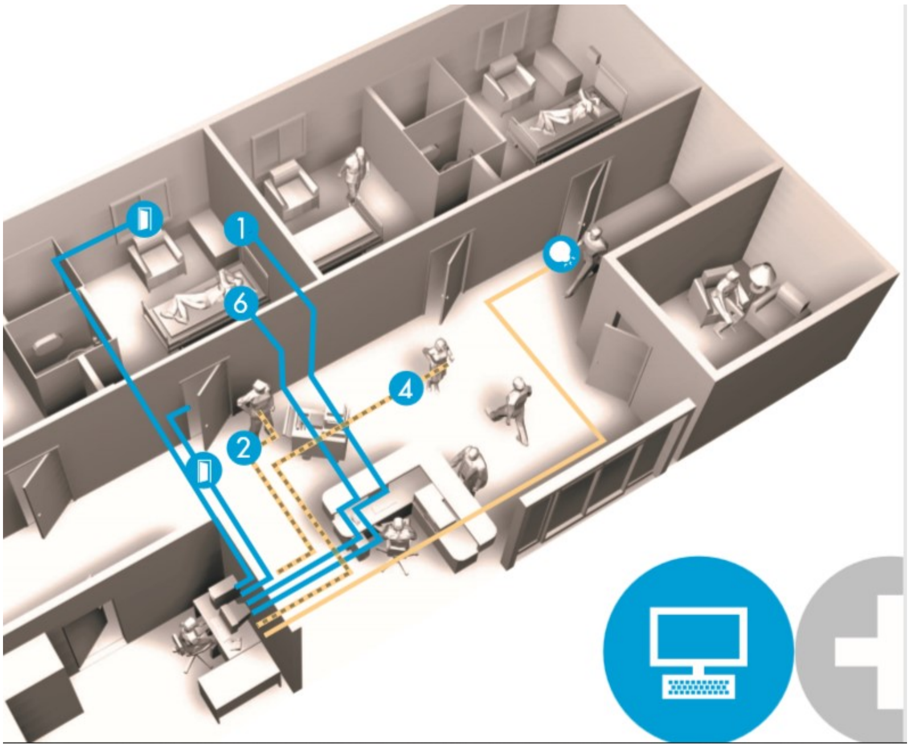


Figure 2.9 - ALCAD's system simplified workflow [14].

According to the market solutions, it is raised the need of a solution that essentially integrates interfaces and patient tracking of ALCAD system while keeping the professional informed and raising notification and critical phases. This innovation allows a digitalization of daily procedures and consequent resources optimization as well as developing a system that fills the proponent company requirements.

From a market point of view, the modular requirement should be focused to keep the system scalable and adaptable to different clinics and conditions [14].

2.2 Software

Looking now to the software implementation, the system requires different programming languages, as well as communication protocols that better fit the system requirements.

Starting by the Remote Module, a regular controller can be programmed in for example in C/C++ compilers, and might support several wireless application level protocols as Message Queuing Telemetry Transport (MQTT), HyperText Transfer Protocol (HTTP), File Transfer Protocol (FTP), Domain Name System (DNS), WebSocket, among others. For the system described, it was thought to use MQTT protocol as the overall implemented protocol for real-time processes as it is already used in IoT applications and HTTP for a simple, informative webpage display.

The professional's device can be a smartphone or tablet with Android operative system. Regarding the Android application, the compiler Android Studio Integrated Development Environment (IDE) is widely used as a well-stablished compiler for mobile computing development in professional or university environment.

2.3 Android

Android Studio

Android Studio is an IDE for Google's Android operating system designed specifically for Android development. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development, Figure 2.10.

The Android application was coded in Android Studio software, utilizing both JAVA for the logical part and Extensible Markup Language (XML), similar to Hypertext Markup Language (HTML), for layout components' rearrangement.

This software presents several features that make it a well-established IDE in Android community such as [15]:

- Flexibility;
- Being a unified environment for all Android devices;
- Having "Instant Run" to apply changes to running applications without having to compile a new installer;

- Checking tools to detect performance;
- Usability;
- Version compatibility;

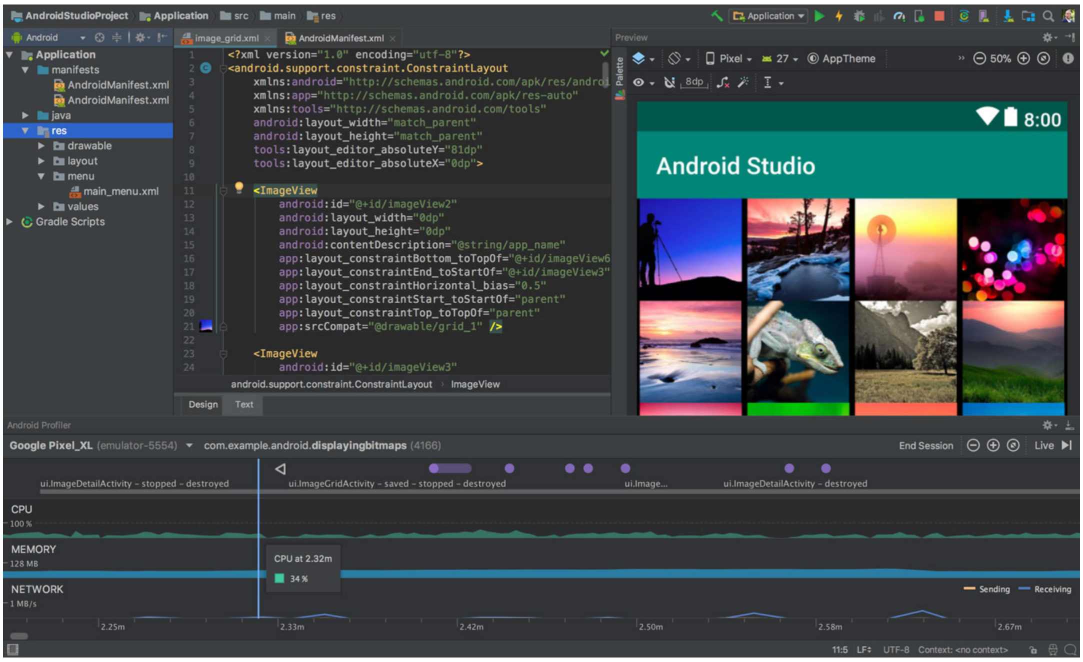


Figure 2.10 - Android Studio IDE interface [16].

2.4 Server

Server Background

Regarding the Server Module, several query languages might be used to manage a database, but it is decided since early stages of the project that a regular Structured Query Language (SQL) language like SQLite is enough for system requirements emphasized by the higher student's proficiency from previous course lectures.

At this time, the Server Module should be capable of supporting MQTT protocol while interfacing SQLite, which is accomplished by a server capable of real-time WebSocket communication and database communication. This feature could be accomplished by Node server, Apache Server, Internet Information Services (IIS) server, but it was chosen a Node.JS one due to its high number of applications in real time and IoT systems with MQTT interface and SQLite one.

Android Background Service

Service is an application component capable of performing User Interface (UI). A Service may continue to run in the background even if the user switches to another application like when minimizing a task. For example, a service can handle network transactions, play music, perform file I / O, or interact with a content provider, all from the background [17].

Android Activity

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place the UI.

While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows or embedded inside of another activity [18]. Figure 2.11 presents an Android simple Empty Activity with the usual "Hello World" TextView as an example.

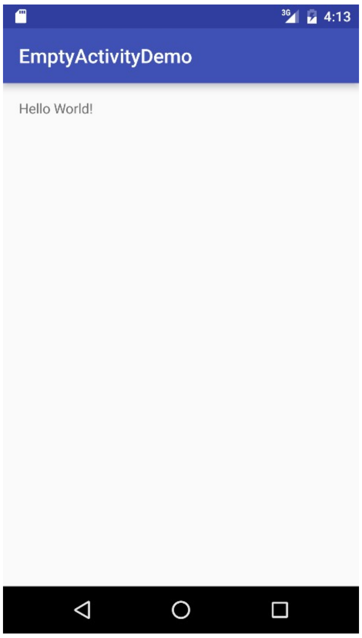


Figure 2.11 – Android Empty Activity with "Hello World" TextView [19].

Android *Fragments*

A fragment represents the behavior or a part of the user interface in an activity. A fragment is similar to a section of an activity, which has its cycle, receives its input events and can be handled with the running activity, functioning as a "subactivity" that can be reused in different activities.

These types of components must always be embedded in an activity and its lifecycle is directly impacted by the host activity life cycle. For example, when the activity is paused, all fragments are paused as well, and when activity is destroyed, all fragments are also destroyed. However, while an activity is running each fragment is independently processed [20].

MQTT

MQTT is a Machine-to-Machine (M2M) connectivity protocol. It is thought to be a lightweight publish/subscribe messaging transport. It is convenient for connections with remote locations where a small code footprint is required, or the network coverage is low.

The MQTT public broker is responsible for redirecting the messages published to the specific topic by clients to the ones that are subscribing to those topics. It is responsible for receiving all messages, filtering the messages, determining which clients are subscribing the topic, and redirecting the message to subscribing clients, as illustrated in Figure 2.12.

This protocol has been used in devices communicating to a broker via wireless in a range of home automation and small device scenarios. It is also useful for mobile applications because of its small size, low power usage, minimized data packets, and efficient redirection of information to several receivers [21].

Last Will & Testament

In MQTT, it is used the Last Will and Testament (LWT) feature to notify other clients about an ungracefully disconnected client. Each client can specify its last will message when it connects to a broker. This message is a regular MQTT message with a topic, retained message flag, Quality of Service (QoS), and Payload. The broker stores the message until it detects that the client has disconnected ungracefully.

In response to the ungraceful disconnect, the broker sends the last-will message to all subscribed clients of the last-will message topic [22].

Keep-Alive Connection Parameter

MQTT includes a keep-alive function that provides an alternative solution for the issue of half-open connections, making it possible to verify if the connection is still open.

Keep Alive ensures that the broker-client connection is still open and both are aware of being connected. When the connection between the broker and the client is established, the client communicates a time interval in seconds to the broker. This interval stipulates the maximum period of time that the broker and the client may not communicate with each other [23].

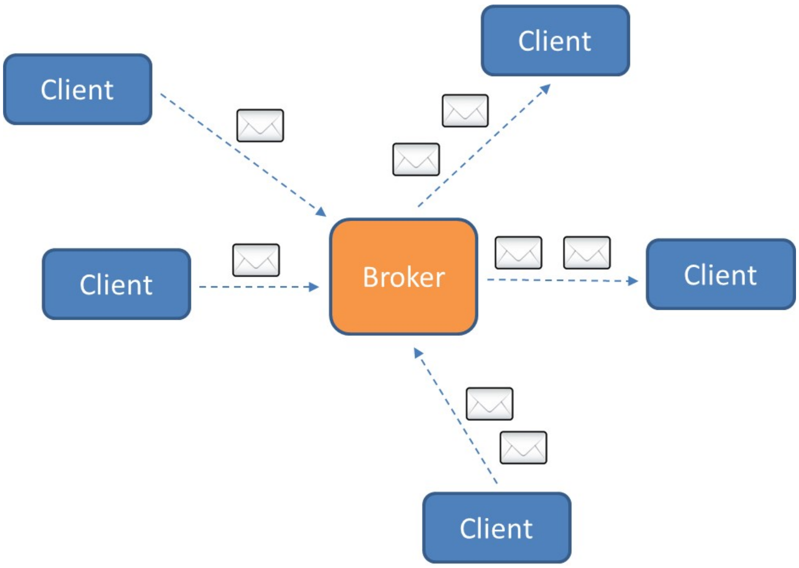


Figure 2.12 - MQTT architecture, Clients and Broker [24].

HTTP

HyperText Transfer Protocol (HTTP) is an application-layer protocol used primarily on the World Wide Web (WWW). HTTP uses a client-server model where the web browser end is the client in communication with the webserver, where the website is hosted. The browser uses HTTP, which is transported over Transmission Control Protocol and Internet Protocol (TCP/IP) to communicate to the server and retrieve Web content for the user.

HTTP is a standardized protocol and has been rapidly assumed over the Internet because of its simplicity. It is a stateless and connectionless protocol [25].

AJAX

Asynchronous JavaScript and XML (AJAX) is a technology used by interactive web applications to make HTTP requests to a server without causing page transitions

[26]. It supports information in various formats, including JSON, XML, HTML and text files.

The most attractive feature of AJAX is its 'asynchronous' nature, which means it can communicate with the server, exchange data, and refresh the page without having to reload the page [27]. Figure 2.13 retracts the AJAX Browser-Server communication.

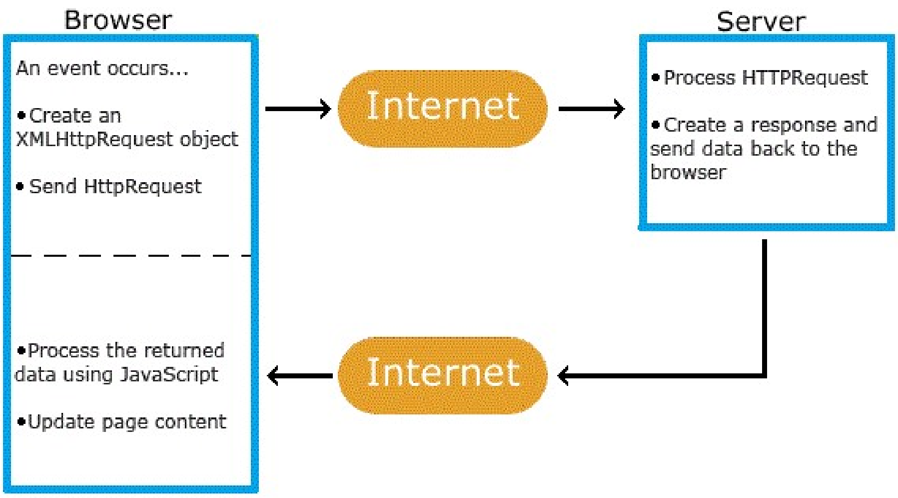


Figure 2.13 - AJAX technology working server-client process [28].

Node.JS

Node.JS is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Node.JS lets developers use JavaScript to write Command Line tools for server-side to create dynamic web page content before the page is sent to the client's web browser. Consequently, Node.JS represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server side and client side scripts [29].

JavaScript is a well-stablished platform for server deployment, making available several interfaces with another platforms and languages, the idea of SQL and MQTT interface will need the use of "*mqtt*" and "*sqlite-3*" libraries through "*npm*" installer [30].

Internet of Things

Nowadays, Internet of Things (IoT) is one of the buzzwords that is pushing ahead the development of new applications that make use of low-cost sensors and communication. One of the most spoken is the Industry 4.0, where there is an overall digitalization of companies' activity so that it can be totally monitored and optimized. IoT is characterized by adding low-cost microcontrollers with internet access to regular devices enabling the control and centralization of information leading to a proactivity increase.

IoT is a future tendency and its application to health purposes enters the iHealth area, where conventional health devices become connected to internet, allowing a similar digitalization and control comparatively to industry's [31].

2.5 Hardware

Emergency Interrupts

Interrupts are events that cause the microprocessor or microcontroller to stop the execution of the task that is performing, work in a different task temporarily and come back to resume the task. Figure 2.14 shows the actions of a microcontroller when an external stimulus triggers an interrupt.

Interrupts can be external or internal. In the present project, only external interrupts were used that are triggered by an event external to the microcontroller.

External interrupts are very useful because they can receive data from sensors without continually asking the sensor if the data is changed. In other words, interrupts allow to avoid polling (constant input request).

Naturally, when an interrupt occurs, it is handled in an Interrupt Service Routine (ISR) which corresponds to a code section that executes when the interrupt happens [32].

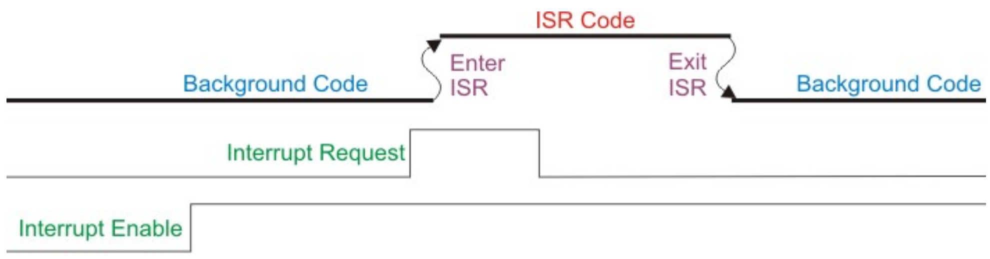


Figure 2.14 - Sequence of processes once the interrupt routine is triggered. The Line "Interrupt Request" represents the trigger timing [33].

I^2C Communication Protocol

This protocol was named by Philips as I^2C and allows the use of many standardized components, which can perform various functions, as well as enable the efficient exchange of information between them.

Each I^2C bus consists of two signals: Serial Clock (SCL) and Serial Data (SDA). SCL is the clock signal, and SDA is the data signal.

As illustrated in Figure 2.15, the devices connected in Inter IC have a fixed address (each component receives a specific address), and allows to be configured to receive or transmit data; in this way, they can be classified as masters (MASTER) or slaves (SLAVE). One of the advantages of the I^2C standard is that it does not set the transmission speed (frequency) because it will be determined by the MASTER circuit (transmission of the SCL) [34].

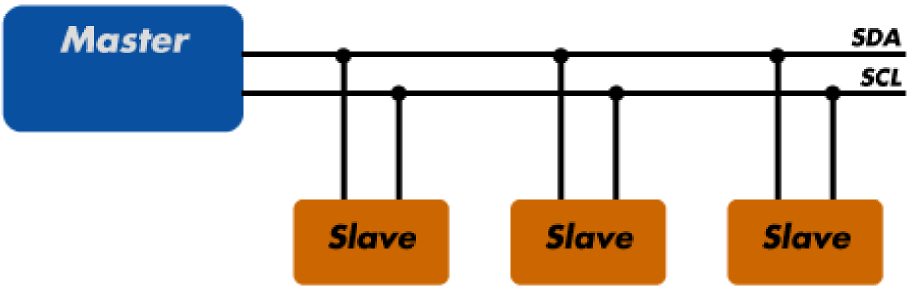


Figure 2.15 - Simple overview of the I^2C wiring [35].

3 The Proposed Solution

The proposed solution is divided into three modules: Remote Module, Server Module and Professional Module, as it is shown in Figure 3.1. The communication between the different modules follows two distinct wireless protocols: MQTT and HTTP.

In the next subsection, it will be explained the modules used in this project, both hardware and the software components, as illustrated in the scheme of Figure 3.1. In the case of the hardware, the selection of the parts will be discussed. Whereas, in the case of the proposed software, the main stages of the methodology will be described in more detail.

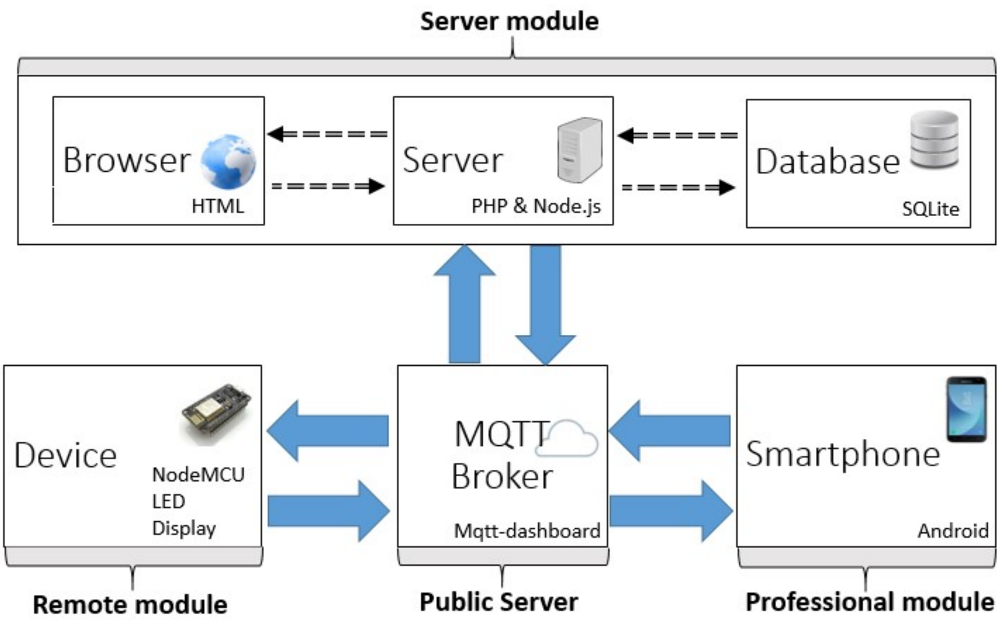


Figure 3.1 - Detailed modular distribution of the presented system and implemented communication protocols.

3.1 Remote Module

This Remote Module should be installed near the patient. This restriction will facilitate the configuration of the software by the professionals, such as the countdown timer and the send of relevant information to the central server.

Regarding the professional requirements, this module presents a reduced number of buttons to simplify the professionals' procedures. The schematic circuit and Fritzing drawing are shown in Figure 3.2 and Figure 3.3, respectively. As illustrated in Figure 3.3, this hardware is composed of four buttons and a display with a 7-segment technology.

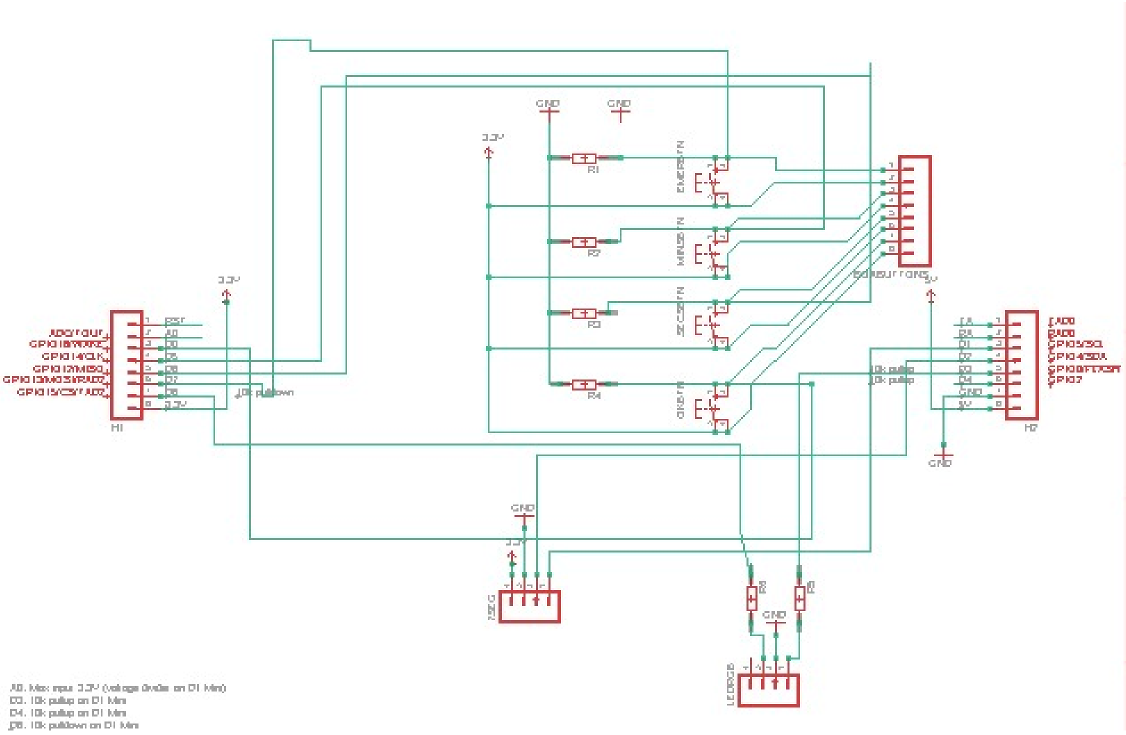


Figure 3.2 - Remote Module's schematic circuit.

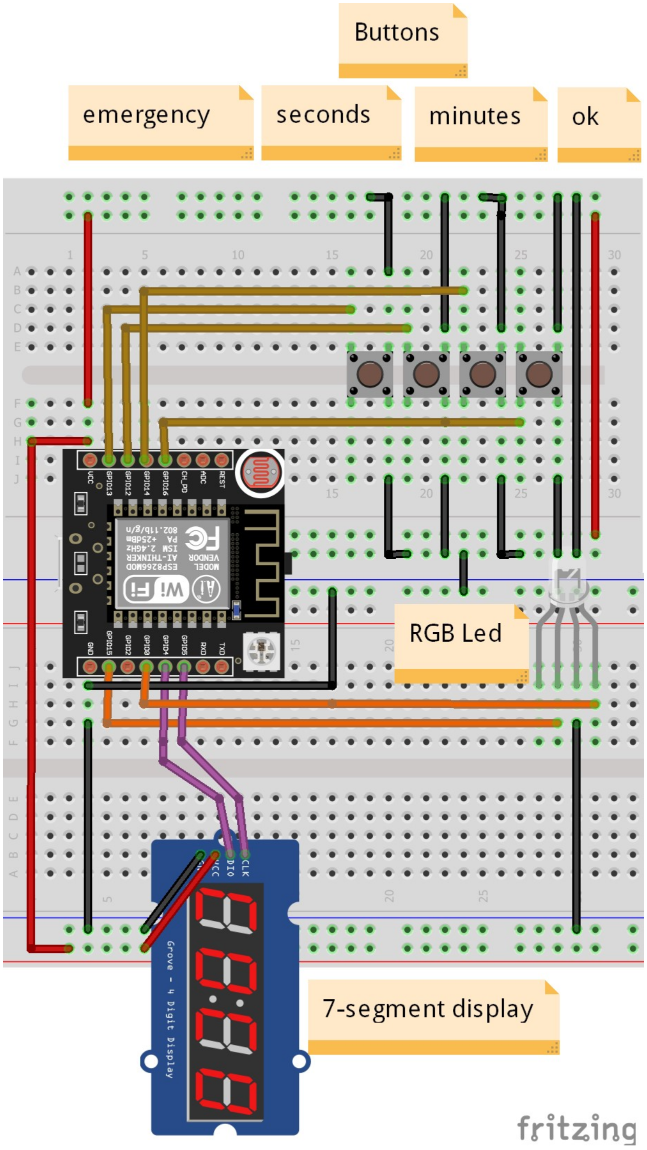


Figure 3.3 - Remote Module's Fritzing drawing.

Following, it is presented a components' list and discussed their utility in the following sections.

Components list:

- ESP8266/NodeMCU D1 Mini
- 7-Segment display
- RGB LED
- 4 Push-Buttons
- 3D printed box
- Homemade Printed Circuit Board (PCB)

Attending the mentioned characteristics, the module has as major parts the microcontroller, a three-color Light-Emitting Diode (RGB LED) for status signaling and a 7-segment display for time display.

This device includes a powerful RGB LED to report the status of the system, avoiding the need to check the display for quick information. The LED might be green, yellow or red, depending on the status of the treatment, respectively: free to be used, on use or on wait/emergency.

3.1.1 Microcontroller

For this work, it was carried out a depth research with the purpose of electing the microcontroller that best suits the needs. The main characteristics should be: small size, facilitating its portability and handling, low-cost and open-source to lower the programming cost, Wi-Fi compatible to allow inter-module communication.

There were mainly five controllers that were found for the purpose: Arduino, ESP8266 series and Raspberry Pi series, Table 1.

Table 1 - Microcontroller comparison

Controller	Arduino Uno[37]	Raspberry Pi[38]	ESP8266 (NodeMCU D1 mini)[39], [40]	ESP32 [41]
Size	7.6 x 6.4 x 1.9 cm	8.6 x 5.4 x 1.7 cm	3.2 x 2.5 x 1.7 cm	5.9 x 2.8 x 1.7 cm
Memory Flash	32Kb	4MB	80Kb	520Kb
Clock Speed	16 MHz	700 MHz	80MHz	160MHz
On Board Network	None	Yes	Yes	Yes
Input Voltage	7 to 12V	5V	3.3V	3.3V
Operating System	None	Linux	None	None
Price	25€	30€	<8€	<15€
Integrated Development Environment	Arduino IDE	Linux support	Arduino IDE	Arduino IDE
Digital pins	14	41 GPIO Pins	9	None
Analog pins	6		1	
SPI communication compatible	Yes	Yes	Yes	Yes
I^2C communication compatible	Yes	Yes	Yes	Yes

By analyzing the Table 1, it is possible to conclude that although the Arduinos fulfilled the requirements, it lacks the Wi-Fi communication, as in the base platform and is costly while compared with ESP solutions.

Similarly, the Raspberry Pi was discarded because of the increased costs and being "overkill" to use a fully working Linux operative system for a simple task that a conventional microcontroller would be enough.

The NodeMCU, derivative of ESP8266 microcontroller, is an emergent micro in the field. This particularity makes it faster than the Arduino, cheaper and it incorporates already a Wi-Fi module.

Despite any other arguments, the cost was the most predominant characteristic from all, since the cost of the controller with integrated Wi-Fi was less than $5 which is much lower than any others without Wi-Fi modules. The main reason for the cost priority as to deal with the significant number of the nods that may be necessary to use in a single clinic.

Figure 3.4 shows the microcontroller used and its pinout arrangement.

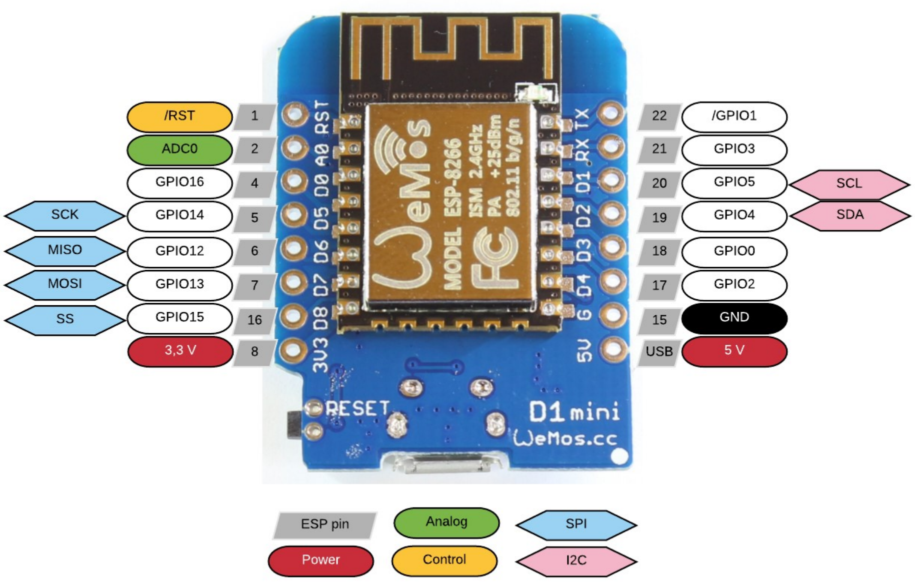


Figure 3.4 - D1 Mini ESP8266 Module pinout [42].

3.1.2 7-Segment Display Choice

In this case, the primary requirement was a great visibility, with a high level of contrast to facilitate the visualization of the information. In this way, it was selected the 7-segment (Figure 3.5), instead of an alphanumeric display (Figure 3.6), given the requirements defined above [43], [44].

Figure 3.5 - Example of 7-segment display.

Figure 3.6 - Alphanumeric digital display [45].

3.1.3 Enclosing Box – 3D Printing

The electronic components should be enclosed to protect them against falls and liquid.

For the prototype phase, a box was 3D printed with the same dimensions of OKW company (Buchen, Germany) commercial model, Figure 3.7. The drawings are presented in Annex A - [46].

Figure 3.7 - OKW's enclosure box (external/general appearance) [46].

3.1.4 PCB Design

After defining the box, the PCB was designed to fit on it. Figure 3.8 shows a detailed diagram of the fitting board, with the illustration of the holes where PCB should be screwed to the enclosing box.

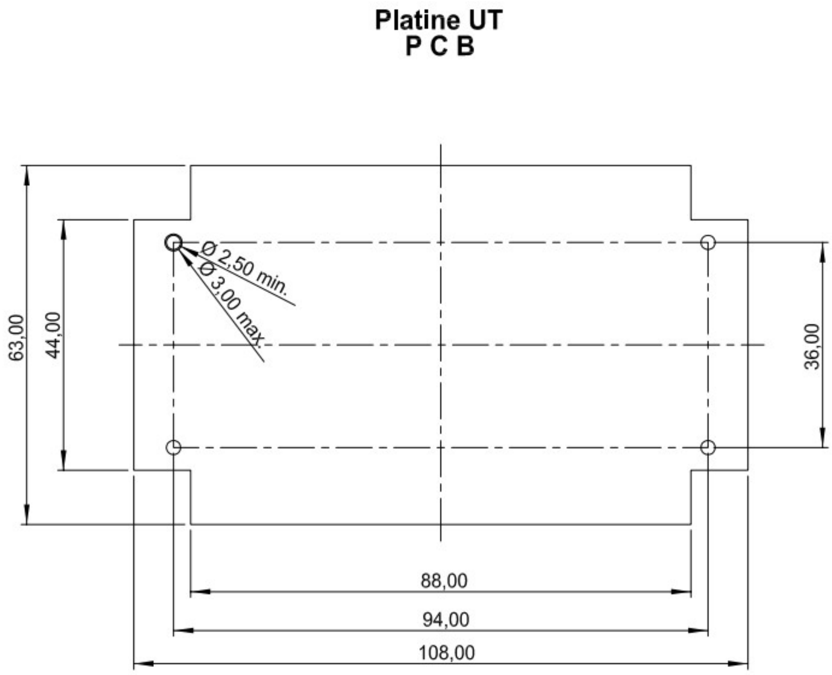


Figure 3.8 - OKW's enclosure box (L size internal platine measures) [46].

In this way, the PCB design present in Figure 3.9 was drawn considering the measures above and having the ergonomics aspects in mind. It shows the scheme of the

PCB with the “location” of each of the components previously described (led, buttons and others).

Using as reference the numbers presented in Figure 3.9, the button (4) and the button (5), have as main function the minute and seconds incremental, respectively. They were placed above the minutes and seconds number in the 7-segment display (3), and the “OK” button (6) at their right as it is shown in the figure. Those buttons are connected to the female headers (7) so that when external switches are plugged into the enclosing box, they can be easily connected to the board without further soldering.

The RGB LED (2) was placed in the upper left area, while the emergency button was placed out of the most active zone (buttons). The same way, the 7-segment was rearranged at the middle of the board and the microcontroller is going to be on the top (1).

EAGLE, an acronym for Easily Applicable Graphical Layout Editor, belongs to AutoDesk, Inc. (San Rafael, California). It is widely used by educationalists, students and professionals because of its rich yet straightforward interface with large component library cross-platform support. Autodesk EAGLE is an Electronic Design Automation (EDA) software. Enabling PCB designers to connect schematic diagrams, component placement, PCB routing, and comprehensive library content [24], [47].

The reason for this arrangement between the box and PCB was leaving the area on the bottom free aiming for future upgrades, namely for a battery.

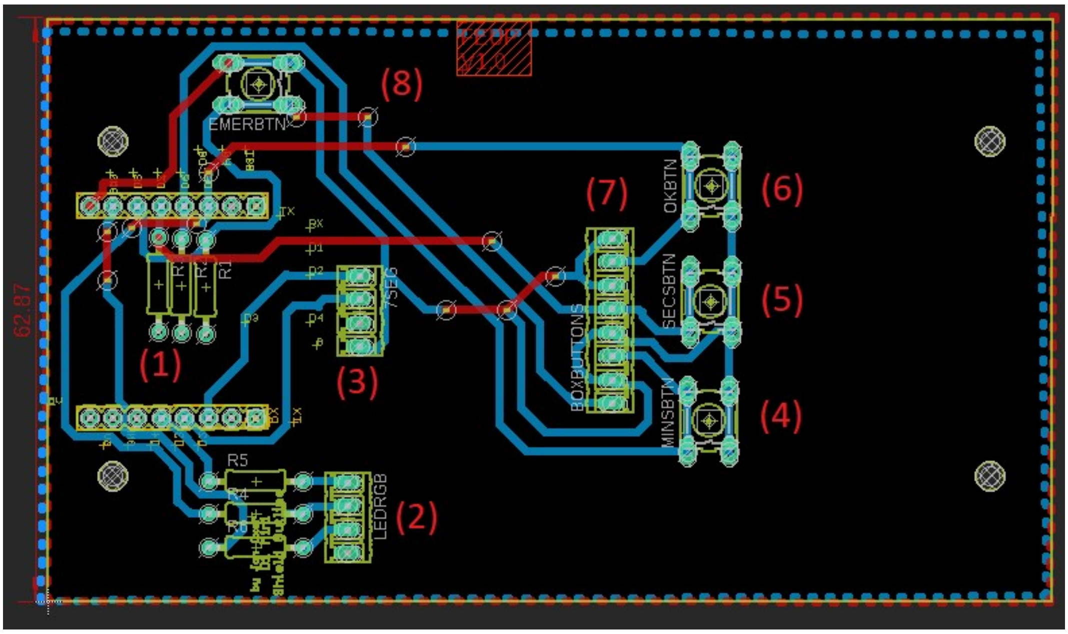


Figure 3.9 - Remote Module's board structure.

3.2 Server Module

The Server Module is divided into two parts: the Node.JS server and the SQLite Database. The choice of Node.JS for server implementation was made taking into account that the server should be constantly running when using a Publish-Subscriber protocol. In this way, a Node.JS server was implemented using the MQTT and SQLite libraries, being responsible for gathering information on MQTT channels and storing it in the system database. The Node.JS server acts as a client in the MQTT broker, by listening to every topic which corresponds to the device.

Hereafter are presented the software components of the Server Module.

Components list:

- Node.JS server
- SQLite Database
- Hypertext Preprocessor (PHP) server

3.2.1 MQTT Tree Structure

Every MQTT broker works by topics which can be divided into different subtopics. In this project, every information runs under the topic "*CMM*", which is divided into subtopics "*CMM/will*", lately explained for disconnection purposes and "*CMM/Room*" further subdivided for single device topic: device one – "*CMM/Room/1*", device two – "*CMM/Room/2*".

For the prototype phase, it was used the publicly available MQTT Broker: "*mqtt-dashboard*" [48].

3.2.2 Database

The choice of SQLite language for database management was based on my proficiency, cost-free and robust enough for the system's needed tasks: updating and requesting data.

The information stored in the database was discussed with the company representative to mirror as much as possible the workflow timings: treatment beginning, device number, minutes, seconds, end times, professional acknowledgments and emergency. For further system control, the remote devices connection tracking is stored, Figure 3.10 presents the relational model composed of the two needed tables: treatment information and connections table.

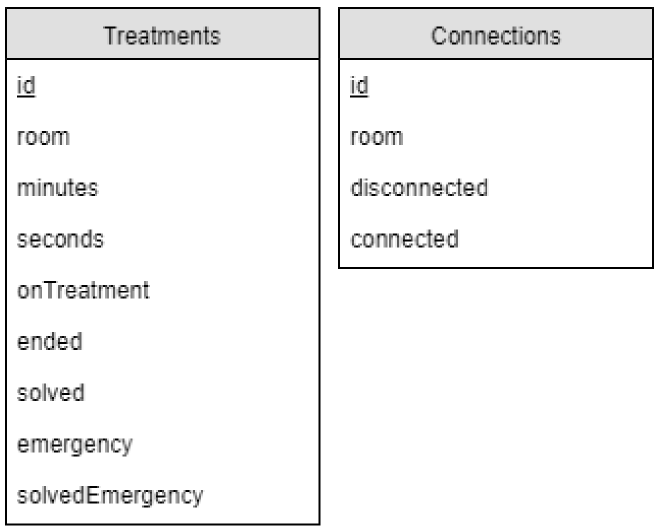


Figure 3.10 - Database Relational Model.

This optimization of the clinic workflow is not an objective of this dissertation, focusing the data collection and the needed hardware and software components.

3.2.3 Mobile Android Application

A mobile Android application integrates the system integrated as the professional's end to display information and acknowledge insertion. The health professional can personalize the app's interface to show countdown times, emergency calls from patients and enables them to inform the system that they have been already notified. The application updates automatically the historical data for every device under monitoring. The remote devices to monitor are added and delete in Subscribe fragment/Tab.

The application development was made focusing the ease of use, since it is supposed to be handled under stress situations aiming not technologically proficient professionals, so the menus were designed to be "user-friendly" and highly intuitive, with few buttons and readable menu display.

4 Developed Solution

This chapter describes the communication between the components of the system apart from the database is based on Wi-Fi wireless communication using the protocols HTTP or MQTT (mostly the last one).

The general overview of the system is presented in Figure 4.1.

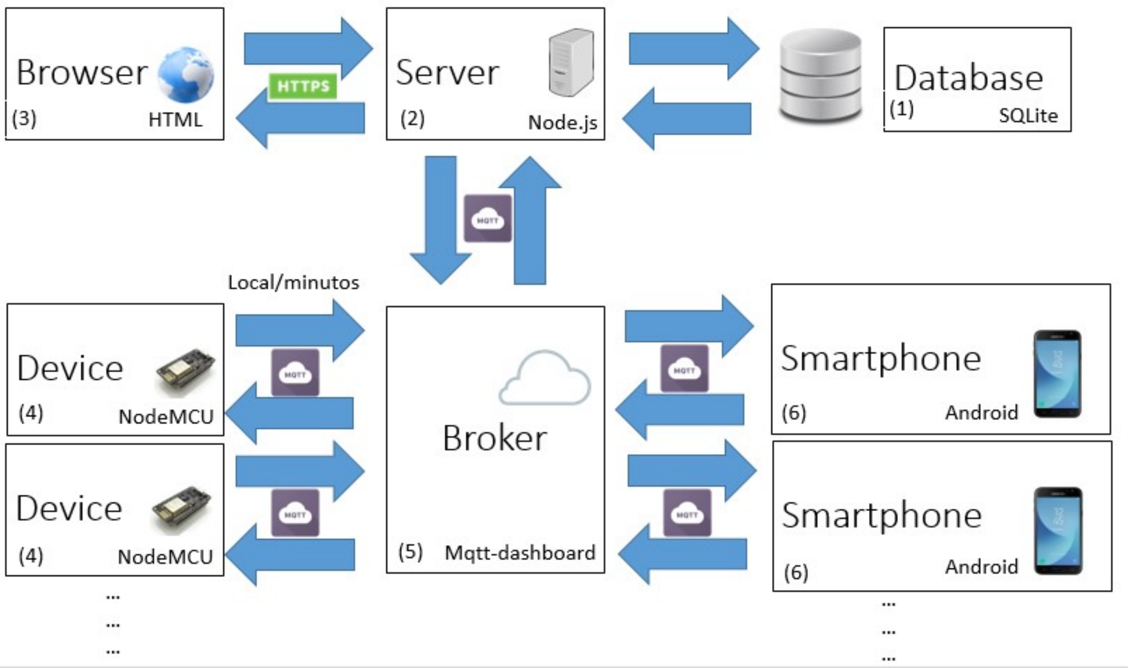


Figure 4.1 - Detailed hardware and software distribution of the presented system with implemented communication protocols.

4.1 System Interface Description

The first part of the system is the database, running in SQLite language, where all the information is stored as it was described in earlier sections.

The new data is inserted in the database whenever the health professional starts a treatment. When the time ends, the professional is notified, and the system is waiting for the acknowledge that is logged together with the time stamp. The same happens for emergency calls.

The second part is a Node.JS server running in the same directory as the SQLite database, currently in FEUP CICA's servers (Centro de Informática Prof. Correia de Araújo), written in JavaScript language that has the role of MQTT client subscribing every topic in the MQTT broker. This Node.JS server is responsible for adding or updating the database information according to the activity passing in every broker's topic. In the presented system, it was used a free public broker named "*mqtt-dashboard*".

The next component is the browser which gives a central simplified overview display of all running tasks. It is a standard HTTP page that displays every device that is in any state besides showing the general clinical status. This webpage is supposed to be visible to the professionals and managers.

The fourth part of the system is the device component, it is constituted by the microcontroller NodeMCU and it is placed near every monitoring device like rooms, muscling machines, beds and any other patient treatment places. These devices are used on purpose similar to the current provided analog solution, where the clinician inserts the time in an alphanumerical display that immediately starts the countdown.

The NodeMCU is Arduino IDE compatible, so it was all coded in C++ just like any other Arduino microcontroller.

Another feature added to this component is the possibility for the clients to call a professional either by an emergency or irregularity with direct notification for the professional, which will be detailed in the next topic.

The sixth part of the system is the Android interface the professional might have like a smartwatch, smartphone or tablet with "CMM Professional UI" application installed. This app acts as a MQTT Client that subscribes and publishes different broker's topics to update it automatically, in real time. Additionally, it sends the alarm acknowledgments to the server.

As it was already mentioned, the app was designed taking into consideration that this interface should be used under stress situations. Once it is initialized, minimizing or blocking the screen will not affect the update and background tasks.

The designed interface is simple allowing only individual activities. Regarding these premises, there were designed three Android fragments as tabs: the first one to see the active and finished/on wait treatments, other for the historical data and another to choose the devices of interested to be notified. This last one controls the MQTT topics that the devices are subscribing on the public broker.

Once an emergency arises, the Android device ringtone is called and a priority is assigned to the on-call device to popup a message, therefore just after acknowledging this alert the professional sees other information.

4.2 Overall System Dataflow

This chapter describes further technical, focusing the flow of data between the modules, protocols, flow direction and the messages' structure.

4.2.1 Patient Treatment Tracking

This chapter explains the flow of treatment information among the different agents, through a simple example, Figure 4.2.

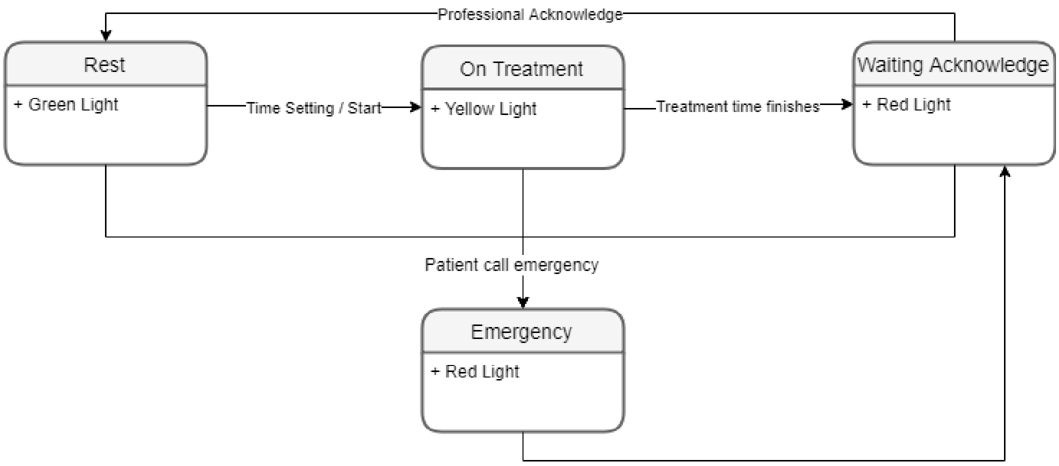


Figure 4.2 - Remote Module State Diagram of a single patient treatment.

The system's dataflow is triggered once the professional enters the treatment time in the device and presses the "OK" button. Once it is done, the microcontroller publishes a message for the *mqtt-dashboard* broker like "*2/15/Begin*" to topic "*CMM/Room/4*", taking as an example two minutes and fifteen seconds treatment starting at device number four.

The Node.JS server is a MQTT Client subscribing topic "*CMM/#*", which means that it receives every message published inside "*CMM*" topic. That way, it will receive the forwarded message from device four and split it by "/" and add two minutes and fifteen seconds treatment to the SQLite database. This treatment instance is stored at the local time, so filling the column treatment with this message that has just been received.

The same process occurs to the Android, which is subscribing the same topic for device number four, with the same message. The countdown timer is set and time is displayed in the main interface.

As soon as the called timer ends, if no emergency calls are made, the microcontroller will be responsible for sending a finishing message to the broker like "*2/15/End*" to the topic "*CMM/Room/4*". This will update the database's "End" column in the last room row to the local date at the time. The same process occurs in the smartphone when it receives the ending notification for the specific device, it internally updates the device variable state and refreshes the user interface, moving the timer from the active tab to finished tab, waiting for the professional acknowledge.

Once the treatment ends and the patient is put on waiting, the professional acknowledge can be done from the smartphone, by pressing the button available in the "Finished" sector or by pressing the "OK" button in the compartment device. Regardless of the source of the acknowledge, it is published to the topic "*CMM/Room/4*" the message "*2/15/Solved*", the other is instantly updated and the column "solved" of the last treatment on room four is also updated with the local time.

The dataflow described is the usual for the normal flow executions, without disconnections from the MQTT broker/Wi-Fi router or emergency calls.

In the case of an emergency, a message like "*2/15/Emer*" will be published to the topic "*CMM/Room/4*", which, similarly to other cases, will update the emergency column with the local date at the time.

In Figure 4.3, it is presented the described dataflow example with the synchronized column data insertion.

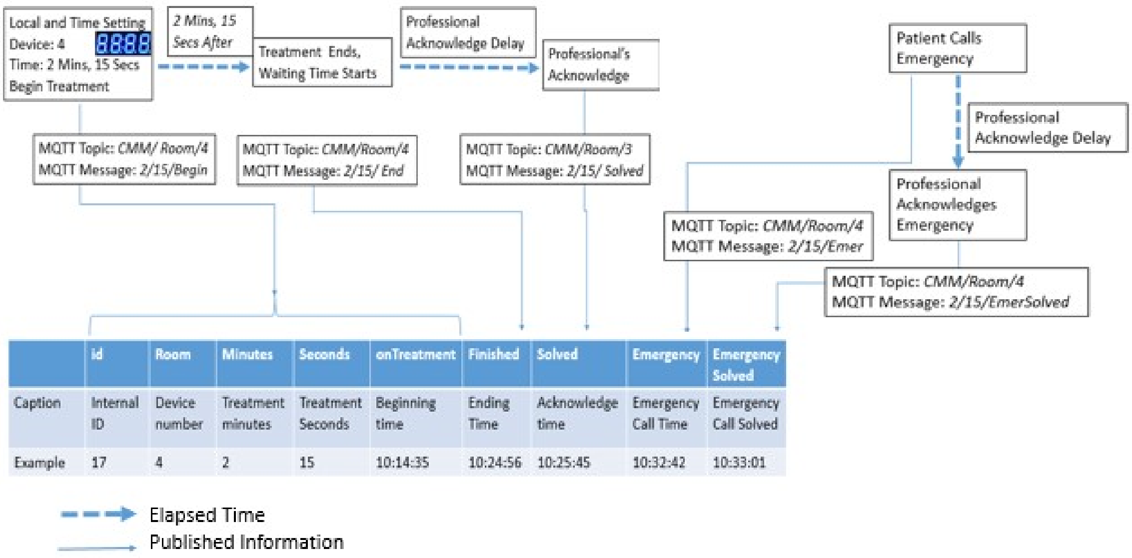


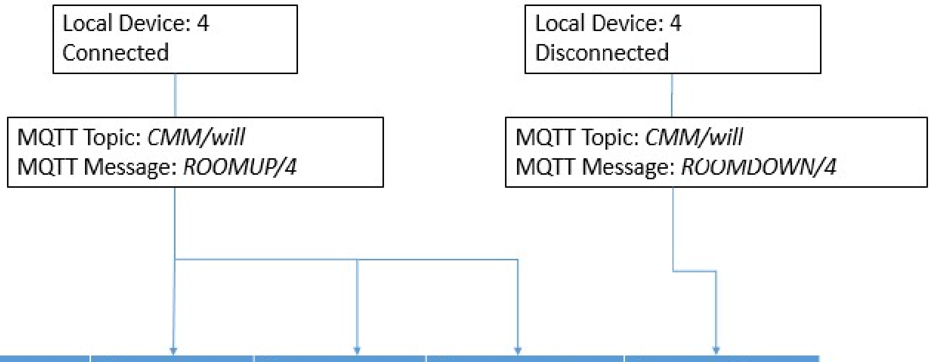
Figure 4.3 - System's example treatment status dataflow synchronized with column data insertion. The given example is the same as the one given in the text.

4.2.2 Connect and Disconnect Tracking

The connected and disconnected time emerged from the fact that in a wireless system, it is normal to have component disconnections that must be handled by themselves. Beyond the reconnection to the Wi-Fi router and MQTT broker on the Remote Module, the connections and disconnections times are recorded in the server every time the microcontroller connects and disconnects.

The connection data, from communication protocol perspective, is trivial to accomplish since the connection is an active process. Inversely, the disconnection detection from the server is random and is not a consequence of any other process. With that characteristic in mind, it is used the Last Will & Testament feature of the MQTT broker when a client stops answering to the Keep-Alive message.

Once a client disconnects from a specific topic, every other client connected to the same topic receives a particular last will message, which in the current system is "*ROOMDOWN/4*" read by the Node.JS server on the "*CMM/will*" topic leading to a table update. The dataflow image in Figure 4.4 shows the described database updates.



	id	Room	Disconnected	Connected
Caption	Internal ID	Device number	Disconnected Time	Connected Time
Example 1	17	4	*2018-04-17 17:30:55*	*NULL*
Example 2	18	4	*NULL*	2018-04-17 17:30:40

Figure 4.4 - System's example disconnections dataflow synchronized with column data insertion. The given example is the same as the one given in the text.

5 Demonstration

In this section, it will be presented the demonstration and further discussion of some noticeable points in respect to product development, programming language or code, the tip followed or any other aspect that may seem relevant for each module: Remote, Professional and Server.

5.1 Remote Module

This module is composed by the microcontroller, its electronical peripherals: 7-segment display, buttons and LED's connected by PCB board and the enclosing 3D box.

5.1.1 Prototype

The final prototype aspect is presented in Figure 5.1. The components were arranged to be intuitive, with the minutes and seconds buttons above the display while the emergency button is maintained distant from the most used device's zones. Figure 5.2 retracts Remote Module appearance and legends the components' arrangement.

Figure 5.1 – Remote Module images closed: a) TOP perspective; b) UP perspective.

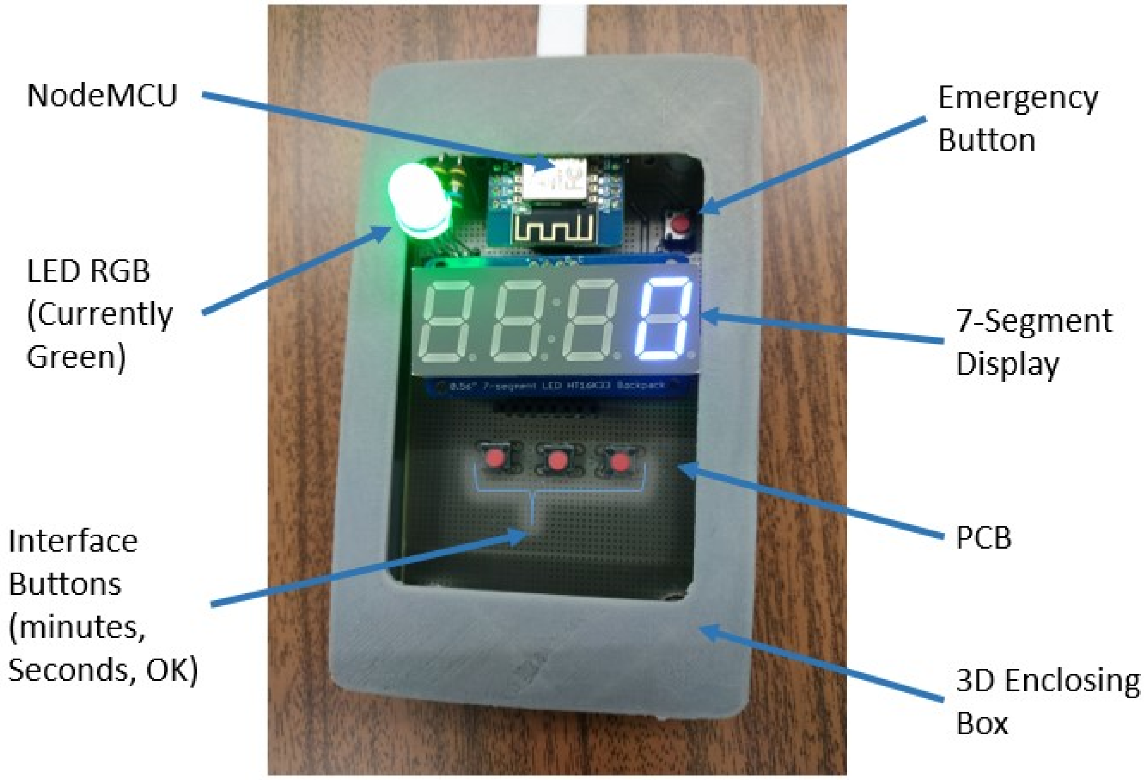


Figure 5.2 - Remote Module closed turned on with peripheral legend.

5.1.2 Printed Circuit Board

The PCB does not add any different features since it is only for robustness and fixation purposes. The soldering connections and the headers for the components provide the strength needed for daily use and to build the consistent trust that the installed product keeps the same quality of service along the use.

Both Figure 5.3 a) and b) show the top and the bottom layers of the printed PCB. These are the printed version form of the board presented in Figure 3.9.

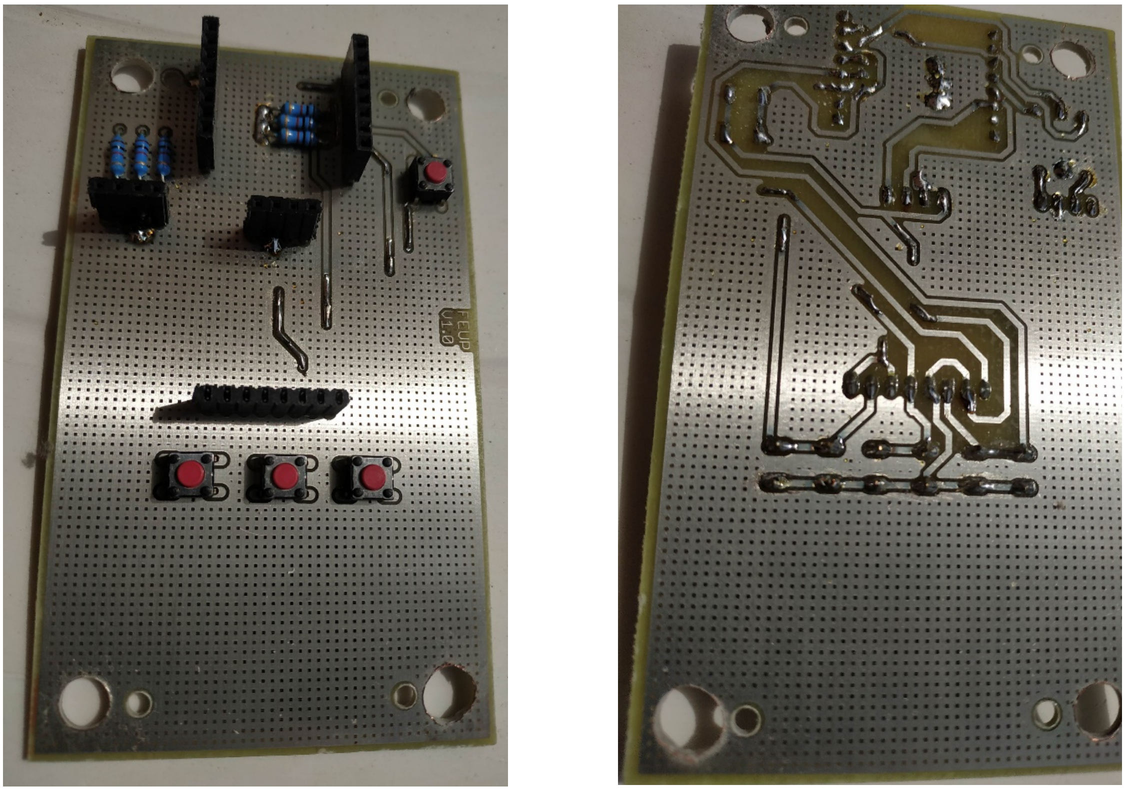


Figure 5.3 – Representative images of the PCB: a) TOP perspective; b) BOTTOM perspective.

5.1.3 NodeMCU Programming

The microcontroller has the active role of keeping the connection open to the broker, and the I/O signals (buttons, LED).

The Setup Routine

The setup routine is responsible for initiating the communication with the 7-segment (I^2C), Wi-Fi and MQTT communication as well as setting the input, output and interrupt pins.

The Loop Routine

The Loop routine defines the procedures that will be repeated forever. This routine is responsible for controlling the button inputs and outputs in order to update time and control the LED status. Every cycle, the treatment remain time is displayed in the 7-Segment display.

When the time is defined, an emergency is called, a message is received from the MQTT Broker, the NodeMCU automatically establishes a Wi-Fi communication with MQTT protocol to the broker or the internet access point (both are tested every cycle). At the end of the routine, the RGB LED status is updated.

Regarding the emergency component, it makes use of a hardware interrupt. Once the emergency button is pressed, an emergency routine is executed to notify the server and the professional as soon as possible.

5.1.4 Remote Module State Sequence

Figure 5.4 is the practical result of the state diagram presented in Figure 4.2 with Professional-Module actions needed. Regarding the Figure 5.4, the enumerated seven states will be following detailed.

When turned on, after establishing the connections to broker and Wi-Fi access point (where “9999” is displayed on the 7-Segment) the module starts in (I), with the green light to indicate the associated device is free/empty. The example shows a thirty-second treatment.

Once the professional inserts the treatment time with the minutes and seconds buttons, it is displayed on the 7-Segment, as seen on (II). The minutes and seconds buttons act as incrementers, one minute and fifteen seconds, respectively. As soon as the professional presses the “OK” button, the LED becomes yellow, the countdown starts, the beginning message “0/30/Begin” is published to device’s topic and states (III) and (IV) are passed.

When the thirty seconds time ends, treatment finishes, the light becomes red, “0/30/End” message is published to the device’s topic and the module is now waiting, on state (V), for the professional to acknowledge by pressing the “OK” button. When it happens, the module returns to state (I), and “0/30/Solved” message is published to the device’s topic. This acknowledge might also be done through the smartphone, what will cause the remote to module update the same way,

The state (VI) is triggered anytime the patient calls the professional or emergency state is activated by pressing the emergency button, the module stops its recurring state, publishes “0/30/Emer” to the corresponding topic and waits for acknowledge. This functionality is achieved through the use of NodeMCU’s external interrupts.

When on this state, the light becomes red (despite appearing to be yellow on the figure) and it waits like in state (V) for professional acknowledge to return to state (I). This acknowledge is characterized by publishing "0/30/EmerSolved" either by smartphone or pressing the "OK" button in the module.

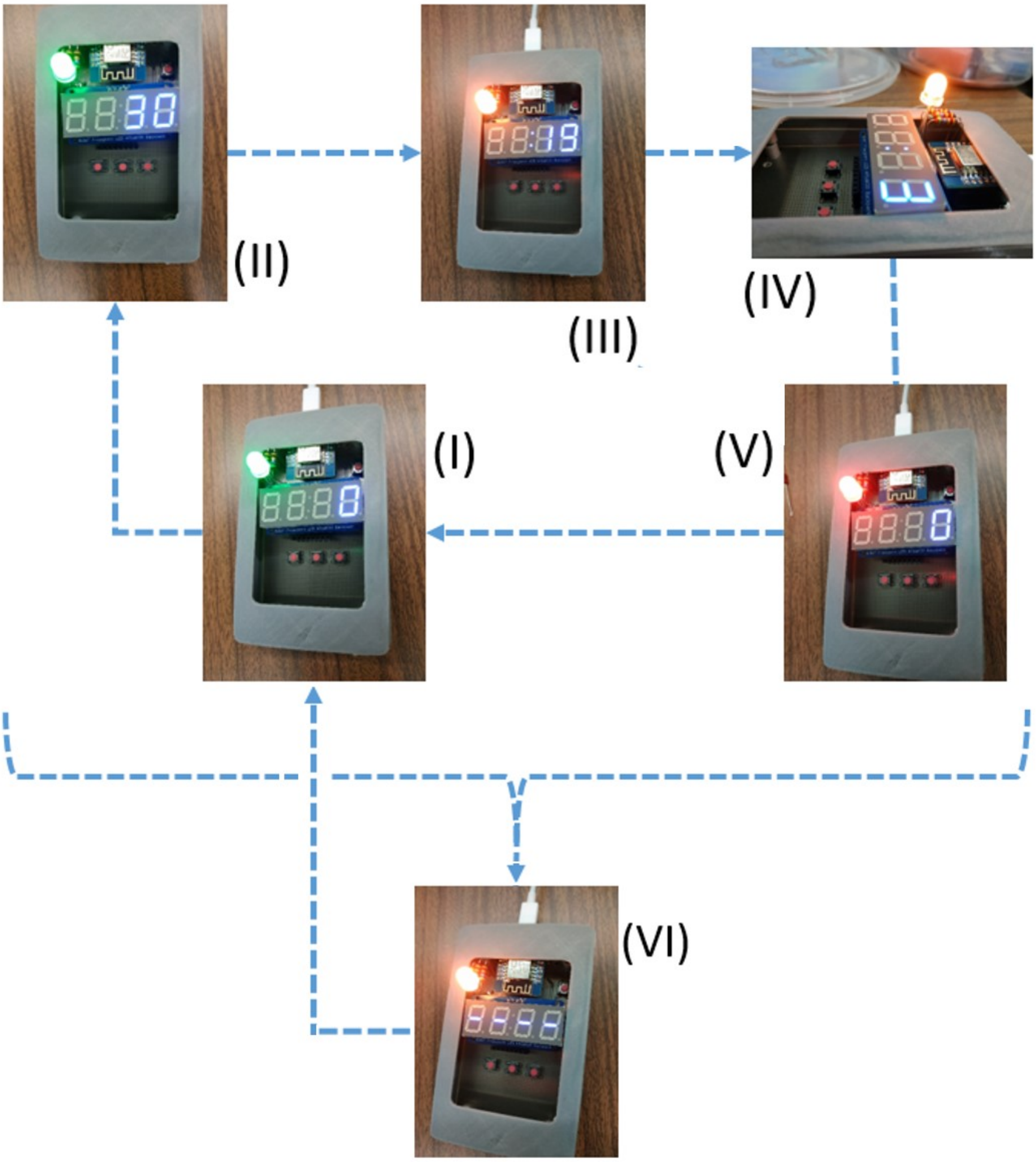


Figure 5.4 – Sequence of Remote Module States for Professional-Module Actions. (I) Rest state; (II) Setting time while in rest state; (III) and (IV) counting down time; (V) Treatment time ended; (VI) Professional/Emergency call.

5.2 Professional Module – Android

This module is constituted by the smartphone application, providing several tools to keep the professional connected to the system and consequently to its patients. In Figure 5.5, it is shown the Unified Modeling Language (UML) Use-Case Diagram for Application interaction between the physiotherapy professional and Android Application.

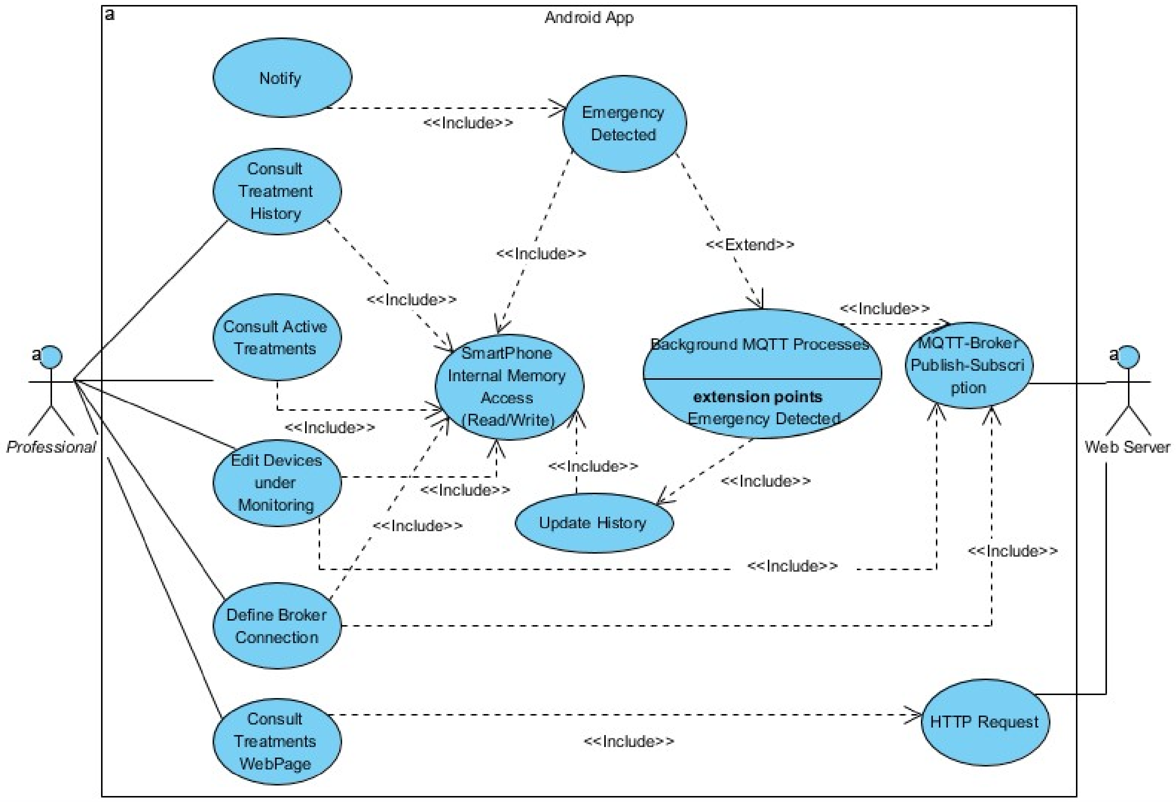


Figure 5.5- Application's UML Use Case diagram.

5.2.1 Application Main Features

The professional application is designed for any technically proficient person, so it is "user-friendly" and its proportions menu, menu items, toolbars and buttons are similar to other applications in the Play Store, to make it more intuitive for the users.

The application has three primary interfaces as fragments, which are switch according to a tabbed Android component: The Active fragment, where active treatments are shown, the History fragment where the subscribed devices' notification history is shown and the Subscribe fragment, where the devices to monitor can be defined.

Considering both commercial and professional aspects, the application was designed so that it has minimum actions from the professional. Apart from the device specifications of the Settings fragment, the notification acknowledgments only need a single button pressing.

Since it is designed for professional purposes, once it is open, it updates itself and calls the professional attention even if it is minimized. That is possible because the communication block of code is inserted in a Background Service, so it is run in a parallel thread to the main application interface. Any other fragments or activity run separately to the communication thread for performance and interface purposes.

The devices that the professional wants to get notified are editable at any time and saved in smartphones' internal memory, that way the professional is specifically notified by the device's activity he is attending or paying attention. The device's specificity is achieved by the different topic subscription on the broker.

For the professionals that still are not used to device rearrangement, a single activity is added for further device and its place mapping.

Application development

The application development was made according to good practices measures. Some aspects that must be emphasized are the use of adapters, since they allow to change the list of items that appear so that it facilitates the interaction, making it more intuitive by adding different components like clickable buttons, images and effects.

For device under monitorization data is saved even if the application is closed using TinyDB library. This library provides a simplified interface with Android's Shared Preferences to store data in memory with key-value association.

Another feature that is almost unrecognized is made when a new device is subscribed, the application starts an internal history request, so the History fragment is updated in real time displaying the data that is stored in the server even if the application is closed. This is done by publishing the device number to the topic "*CMM/GetHistory*", like it was described in the previous section.

Application Activity Flow

This section emphasizes the application interactions in current activity workflow (Figure 5.6), emergency case (Figure 5.7) and the way a treatment is displayed (Figure 5.8) from the user perspective.

Figure 5.6 - Application's user interface flow. (I) Activities' flow; (II) fragments' flow.

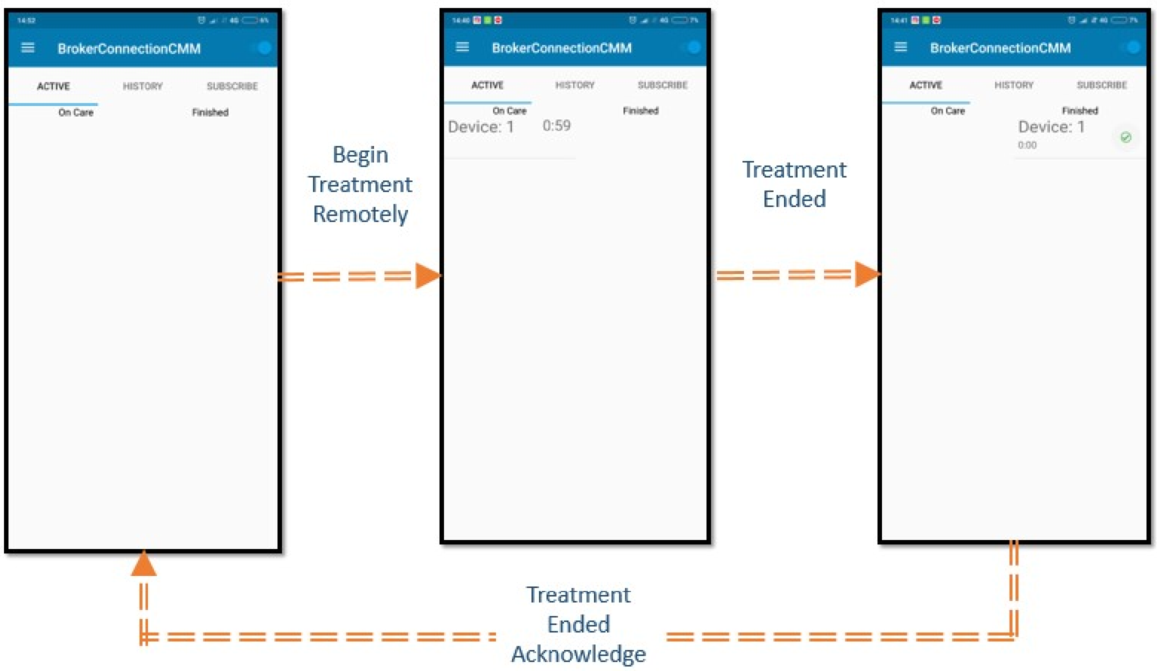


Figure 5.7 - Active fragment, Treatment and synchronized update.

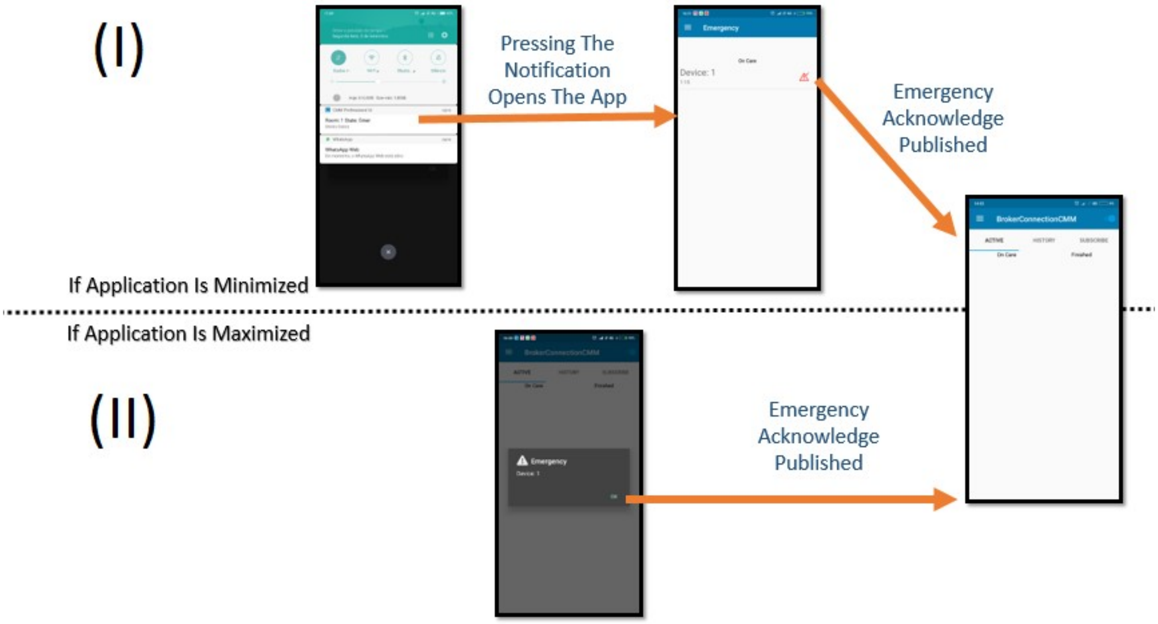


Figure 5.8 - Both Application interface on Emergency Call . (I) Emergency acknowledge by notification pressing; (II) Emergency acknowledge with application maximized.

5.3 Server Module

This module uses Node.JS and PHP. The server component has a passive role in the system, since it only answers to external stimulus. The server saves and redirects data when that activity is triggered on the topic that is subscribing.

Figure 5.9 shows the final Server Module files, currently on FEUP CICA's public servers. This end has the HTTP and Node.JS components.

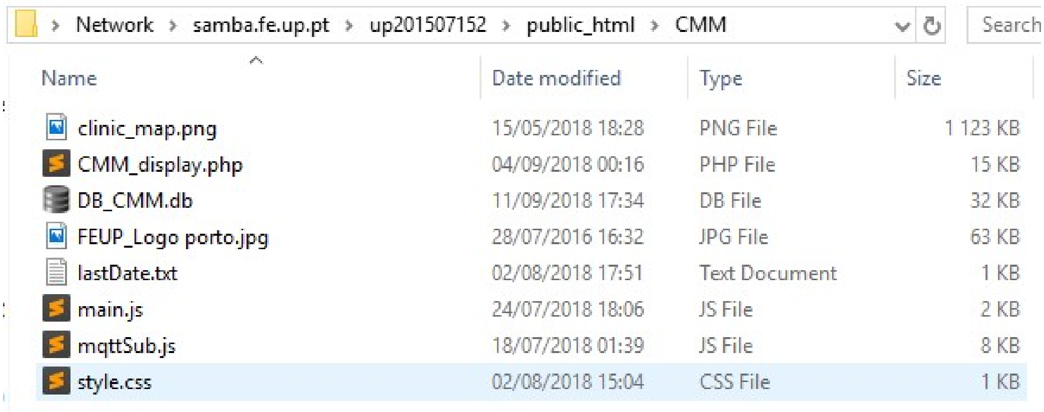


Figure 5.9 - CICA's Server files.

5.3.1 SQLite Database

The final database arrangement is the same as the described on relational model in section 3.2 with two tables, one for storing treatment data (Figure 5.10)and other for remote device connection tracking (Figure 5.11).

	id	room	minutes	seconds	onTreatment	ended	solved	emergency	solvedEmergency
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	810	4	45	15	2018-07-30 16:50:01	2018-07-30 15:16:01	2018-07-30 15:16:01	2018-07-30 15:16:01	2018-07-30 15:16:01
2	811	1	25	15	2018-08-01 16:04:01	NULL	NULL	NULL	NULL
3	812	2	45	15	2018-07-30 16:50:01	2018-07-30 15:25:01	2018-07-30 15:35:01	NULL	NULL
4	813	3	40	15	2018-07-30 16:50:01	2018-07-30 15:16:01	2018-07-30 15:16:01	2018-07-30 15:25:01	2018-07-30 15:25:01
5	814	25	35	15	2018-07-30 16:50:01	NULL	NULL	NULL	NULL

Figure 5.10 – Example of treatment's data stored in database.

	id	room	disconnected	connected
	Filter	Filter	Filter	Filter
1	94	1	NULL	2018-04-17 1...
2	95	1	2018-04-17 1...	NULL
3	96	1	NULL	2018-04-17 1...
4	97	1	2018-04-17 1...	NULL
5	98	1	NULL	2018-04-17 1...
6	99	1	2018-04-17 1...	NULL
7	100	1	NULL	2018-04-17 1...

Figure 5.11 - Example of microcontroller connection data stored in database.

5.3.2 HTTP WebPage

The page was designed to be positioned in a central clinic zone, where it could be easily seen by professionals. The information displayed, as seen in Figure 5.12 and Figure 5.13, is simplified by the device corresponding timers and the treatment state follows the Remote Module's color: green for free device state, yellow on treatment state and red on wait or emergency status.

Since it was written in regular HTML, it can be seen on every browser: https://paginas.fe.up.pt/~up201507152/arduino/CMM_display.php?CMMTrack.

Below the device sequence, Figure 5.13, there is an image with the clinic map for information purposes.

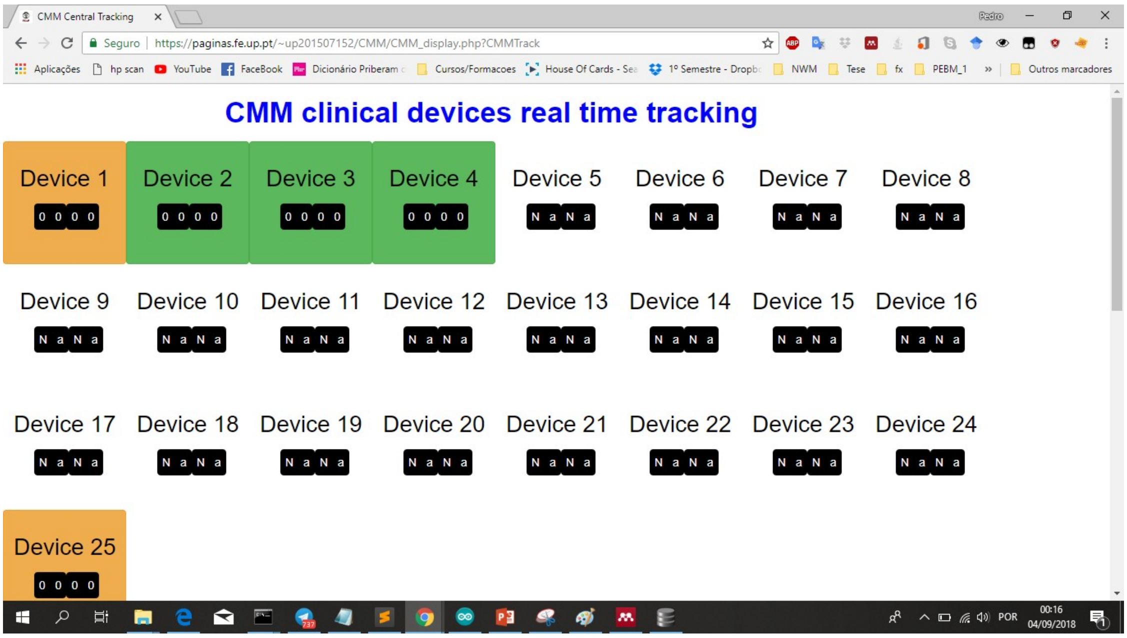


Figure 5.12 - System's Webpage with empty (Green) and waiting professional acknowledge (Yellow).

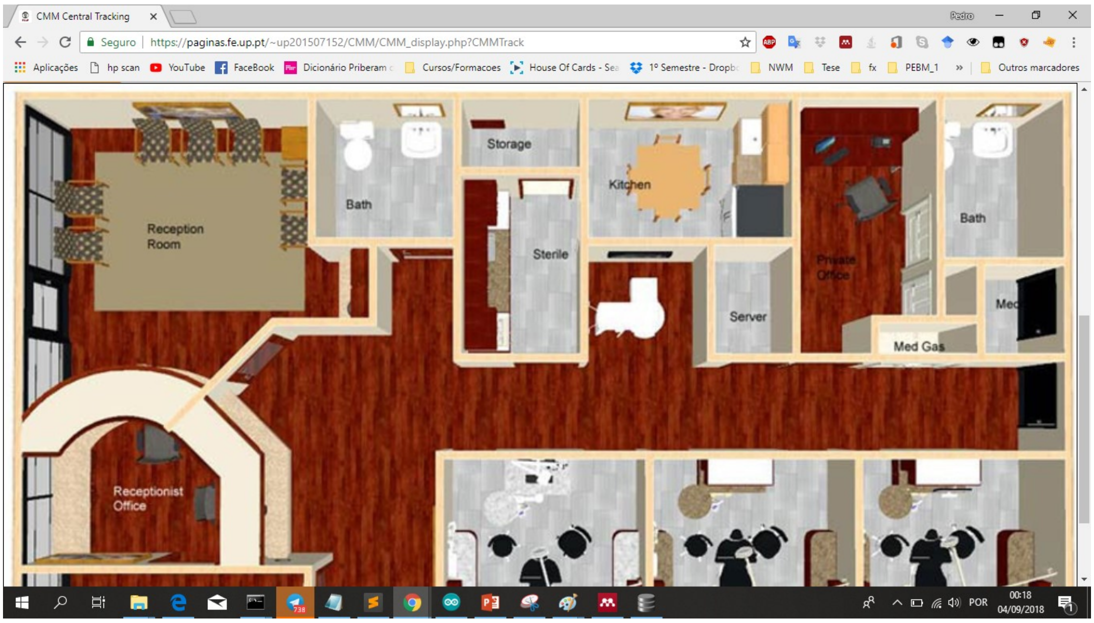


Figure 5.13 - System's Webpage with clinic's map. Scrolled view from Figure 5.12.

This Webpage has an AJAX component, described in Figure 5.14. Since the page information is stored in the database and it must be updated on the client, an AJAX technology side was implemented to the HTTP server. This way, when the page is loaded from the client side, on parallel background activity, it continues to execute the HTTP request without page update.

These HTTP requests aim to ask the server if there is new information, if so, a HTTP GET request is sent to the server for regular browser webpage update, otherwise the page keeps the same aspect.

This new information request lays in last database update. Every time the database is changed, its time is recorded on a "lastDate.txt" file. The client side saves the last update date and iteratively compares it with the last update date through the AJAX request.

Figure 5.15 is presented an Activity Diagram of the sequence of processes enumerated from 1 to 4 in Figure 5.14 between the two entities: Server and Client.

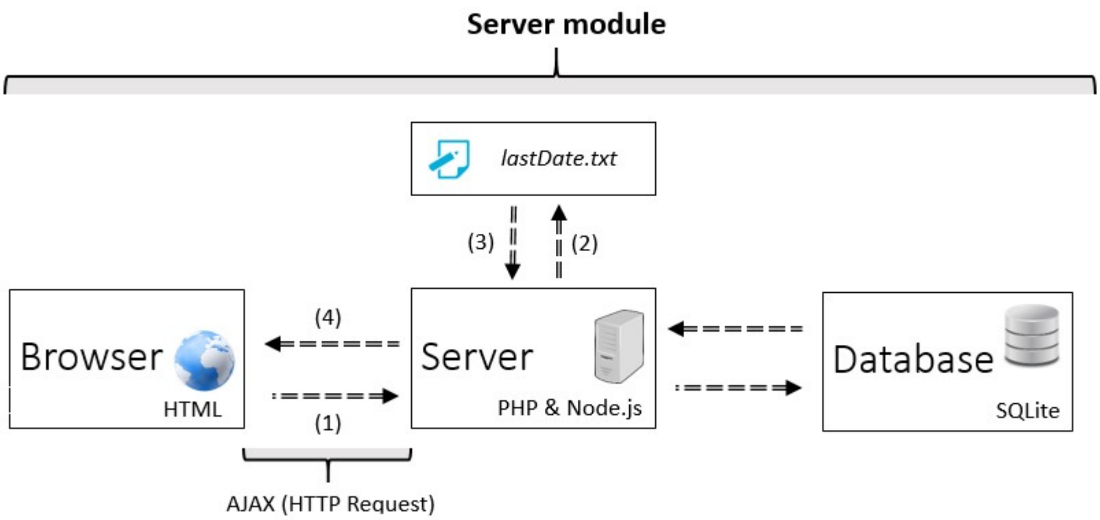


Figure 5.14 – Server-Client AJAX Simplified communication.

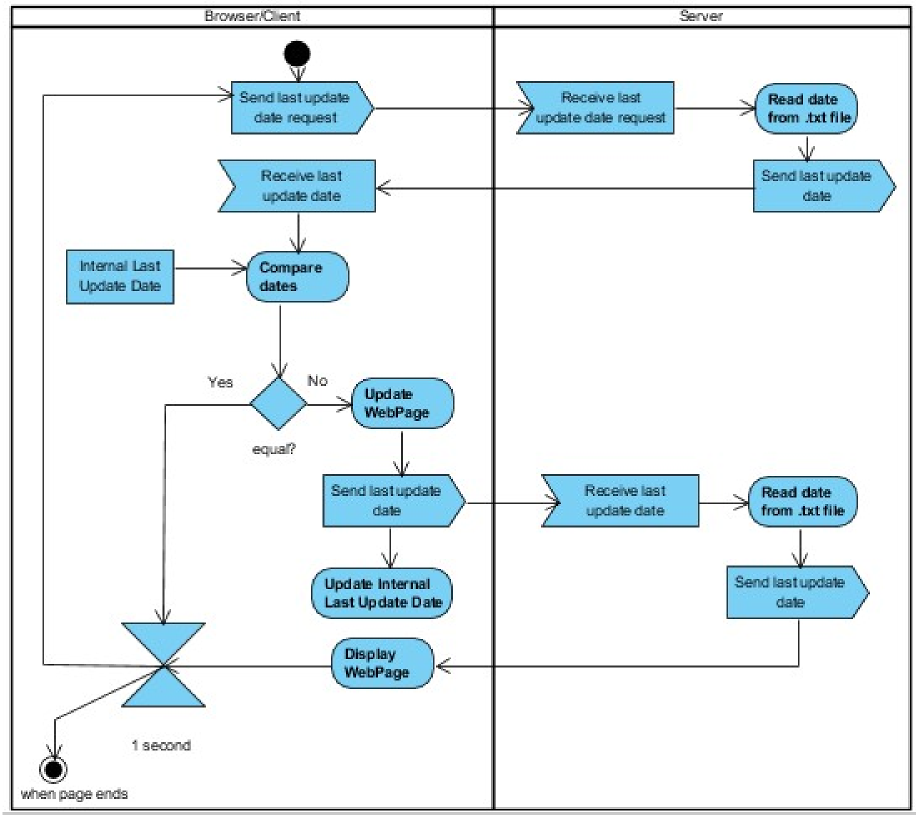


Figure 5.15 – Activity diagram of Browser-Server AJAX communication.

6 Conclusions and Future Developments

6.1 Conclusions

The primary motivation for this project was the lack of information on daily activity of nowadays physiotherapy clinics' workflow. The actual systems are either focused on the patient physical activity, or in the communication between patient (outside the clinic) and professionals; and not to support the clinics management operations (human resources and materials). The proposed system intends to be integrated seamless in the daily activity of the clinic and gather the information of real-time operations in order to allow the optimization.

The system was successfully developed and tested as an adaptable, scalable and low-cost modular system with user-friendly interface composed by three modules: Remote Module, Professional Module and Server Module.

The Remote Module intends to be near the patient to facilitate the configuration by the professionals, using microcontroller to countdown, display time and acknowledgements, while presenting a reduced number of buttons;

The Professional Module is the professional's end, being constituted by a smartphone application to provide several tools to keep the professional connected to the system and consequently to its patients;

The Server Module is the central point that has the passive role to external stimulus. The server saves and redirects data when that activity is triggered.

Analyzing the described system and comparing it with initial requirements, the developed prototype fulfills the preliminary and "on-the-fly" requirements and is fully functional. It is capable of tracking the clinical and its patients' activity without identifying them for data protection purposes.

The prototype is now being installed in the clinic to get the user's feedback to improve both design and functionalities.

6.2 Future Developments

The first topic to be considered is a fully running Node.JS interface much like PM2 process machine, which allows a better configuration, log management, monitoring other performance tracking statistics. The development of an iOS application is also a development for the case apple smartphones are intended to be used.

Development of smartphone or laptop management API for consulting and working the historical treatment data analysis in real time.

Contacting the prototype enclosing producer – OKW company – and a PCB company for cost analysis for mass production. This component also includes the integration of external switches that will be connected to the 8-header already soldered to the PCB.

The last topic is the full integration of the present system in the clinic, developing an application programming interface (API), which will be accomplished by co-working with the informatics department, so that the information becomes accessible through the actual software and the transference of the software to a server provided by the clinic.

References

[1] APFISIO, "Sobre a Fisioterapia.", url: http://www.apfisio.pt/o-que-e-a-fisioterapia/sobre-a-fisioterapia/, last accessed in 05-09-2018.

[2] Reabilitech , "Novas perspectivas para o futuro da fisioterapia.", url: https://www.reabilitech.com.br/novas-perspectivas-para-o-futuro-da-fisioterapia/, last accessed in 03-09-2018.

[3] APFISIO, "A Fisioterapia cresce em Portugal -.", url: http://www.apfisio.pt/fisioterapia-cresce-portugal/, last accessed in 03-09-2018.

[4] Webster, "Outpatient | Definition of Outpatient by Merriam.", url: https://www.merriam-webster.com/dictionary/outpatient. last accessed in 05-09-2018].

[5] Agility Health , "Investor Presentation," 2017.

[6] MSH INTERNATIONAL, " O que deve saber sobre a lei PPACA / ACA.", url: https://www.msh-intl.com/pt/europa/particulares/definicao-aca.html, last accessed in: 05-09-2018]

[7] PTProgress, "Physical Therapist Jobs Outlook: A 10 Year Forecast of Massive Growth.", url: https://www.ptprogress.com/physical-therapist-jobs-outlook/, last accessed in 03-09-2018.

[8] INE, "Portal do Instituto Nacional de Estatística.", url: https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_publicacoes&PUBLICACOESpub_boui=65944632&PUBLICACOESmodo=2&xlang=pt, last accessed in: 03-09-2018.

[9] Physiotrack, url: https://www.physiotrack.co.uk/, last accessed in 04-02-2018.

[10] TrackActive, url: http://www.trackactive.co/, last accessed in 04-02-2018.

[11] MyPhysio App, url: http://www.myphysioapp.com/, last accessed in 04-02-2018.

[12] Accelerware, url: http://www.accelerware.com/, last accessed in 04-02-2018.

[13] Spok, "Clinical Alerting and Notification Systems for Hospitals.", url: http://www.spok.com/solutions/clinical-alerting-notification, last accessed in 04-02-2018].

[14] ALCAD, "Nurse Call System.", url: http://www.alcad.net/uploads/publicaciones/catalogos/pdf/Nurse%20Call%20System%20Catalogue%201.pdf, last accessed in 03-09-2018.

[15] Android Developers, "Android Studio release notes.", url: https://developer.android.com/studio/releases/, last accessed in 03-09-2018.

[16] Android Developers, " Download Android Studio and SDK tools", url: https://developer.android.com/studio/, last accessed in 16-09-2018.

[17] Android Developers, "Serviços", url: https://developer.android.com/guide/components/services, last accessed in 03-09-

2018.

[18] Android Developers, "Activity", url: https://developer.android.com/reference/android/app/Activity, last accessed in 05-09-2018.

[19] Techotopia, "A Guide to the Android Studio Designer Tool - Android 6", url: https://www.techotopia.com/index.php/A_Guide_to_the_Android_Studio_Designer_Tool_-_Android_6, last accessed in 16-09-2018.

[20] Android Developers, "Fragmentos", url: https://developer.android.com/guide/components/fragments, last accessed in 03-09-2018.

[21] HiveMQ, "MQTT Essentials", url: https://www.hivemq.com/mqtt-essentials/, last accessed in 13-09-2018.

[22] HiveMQ, "MQTT Essentials Part 9: Last Will and Testament.", url: https://www.hivemq.com/blog/mqtt-essentials-part-9-last-will-and-testament, last accessed in 03-09-2018.

[23] HiveMQ, "MQTT Essentials Part 10: Keep Alive and Client Take-Over.", url: https://www.hivemq.com/blog/mqtt-essentials-part-10-alive-client-take-over, last accessed in 03-09-2018.

[24] maxEmbedded, "PCB Design using EAGLE - Part 1: Introduction to EAGLE and Software Environment .", url: http://maxembedded.com/2014/06/pcb-design-eagle-part-1-introduction-eagle-software-environment/, last accessed in 08-09-2018.

[25] WhatIs, "What is HTTP (Hypertext Transfer Protocol)?", url: https://searchwindevelopment.techtarget.com/definition/HTTP, last accessed in 03-09-2018.

[26] HttpWatch, "AJAX.", url: https://www.httpwatch.com/httpgallery/ajax/, last accessed in 17-09-2018.

[27] MDN , "Primeiros Passos - Guia de desenvolvimento da Web.", url: https://developer.mozilla.org/pt-PT/docs/Web/Guide/AJAX/Como_começar, last accessed in 11-09-2018.

[28] Stackify, "How to Return AJAX Response from Asynchronous JavaScript Call.", url: https://stackify.com/return-ajax-response-asynchronous-javascript-call/, last accessed in 16-Sep-2018.

[29] W3schools, "Node.js Introduction.", url: https://www.w3schools.com/nodejs/nodejs_intro.asp, last accessed in 3-09-2018.

[30] W3schools "Node.js NPM." [Online], url: https://www.w3schools.com/nodejs/nodejs_npm.asp, last accessed in 03-09-2018.

[31] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[32] Techtutorialsx "ESP8266: External interrupts.", url: https://techtutorialsx.com/2016/12/11/esp8266-external-interrupts/, last accessed in 03-09-2018.

[33] Embeddedrelated, "Introduction to Microcontrollers - Interrupts - Mike Silva.", url: https://www.embeddedrelated.com/showarticle/469.php, last accessed in 16-09-2018.

[34] Sparkfun, "I2C - learn.sparkfun.com.", url: https://learn.sparkfun.com/tutorials/i2c, last accessed in 03-09-2018.

[35] Mikroe, "I2C - Everything you need to know.", url: https://www.mikroe.com/blog/i2c-everything-need-know, last accessed in 07-09-

2018.

[36] Escali, “Escali Timers.”, url: https://escali.com/timers, last accessed in 07-09-2018.

[37] Arduino, “Arduino Uno Rev3.”, url: https://store.arduino.cc/arduino-uno-rev3, last accessed in 03-09-2018.

[38] WIRED UK, “Raspberry Pi 3: features, price and release date”, url: https://www.wired.co.uk/article/raspberry-pi-three-wifi-bluetooth-release-price-cost, last accessed: 03-09-2018.

[39] Zonamaker, “WeMos D1 Mini - Introdução e Primeiros Passos.”, url: https://www.zonamaker.com.br/wemos-d1-mini-introducao-e-primeiros-passos/, last accessed in 03-09-2018.

[40] Wemos, “D1 mini [WEMOS Electronics].” url: https://wiki.wemos.cc/products:d1:d1_mini, last accessed in 03-09-2018.

[41] Espressif Systems, “ESP32 Datasheet Espressif Systems,” 2016.

[42] EscapeQuotes, “WeMos D1 mini pins and diagram.”, url: https://escapequotes.net/esp8266-wemos-d1-mini-pins-and-diagram/, last accessed in 07-09-2018.

[43] Engineers Garage, “LCD-016M002B MECHANICAL DATA.”, 2016

[44] Holtek, “HT16K33 - Product Details.”, url: http://www.holtek.com/productdetail/-/vg/HT16K33, last accessed in 03-09-2018.

[45] Amazon, “Amazon.in: Buy MY TechnoCare 1602 16x2 LCD Alphanumeric [JHD162A] Display Yellow 162A/2x16 for 8051, AVR, PIC, Arduino, ARM Microcontroller Kit Yellow BackLight | Industrial R&D | Student Hobby DIY Project Online at Low Prices in India | MY TechnoCare Reviews & Ratings.” url: https://www.amazon.in/MY-TechnoCare-Alphanumeric-Microcontroller-Industrial/dp/B071HPJ8B8, last accessed in 07-09-2018.

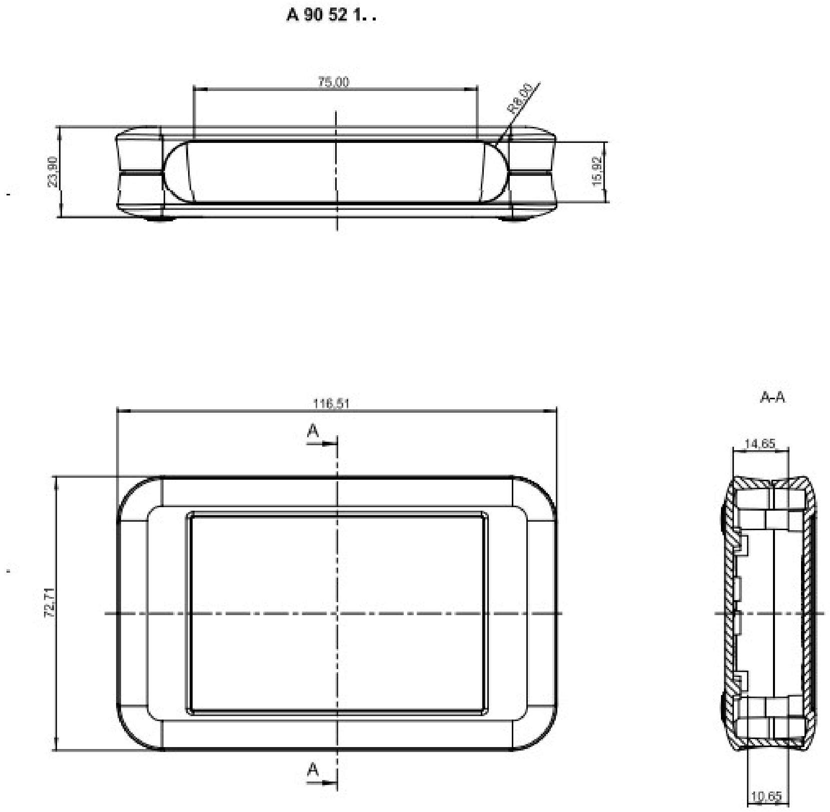
[46] OKW, “2.3 Soft-Case | Caixas Portáteis em Grandes Formatos | OKW.”, url: https://www.okw.com/pt/category/4752e520-6960-11e5-b123-8eba63e66ed5/products?vs=af314fc0-c2e5-11e2-8e2c-0050568225d7%24%24cd733c40-d51f-11e2-a4df-0050568225d7, last accessed in 03-Sep-2018.

[47] Autodesk, “EAGLE | PCB Design Software.”, url: https://www.autodesk.com/products/eagle/overview, last accessed in 08-09-2018.

[48] Mqtt-dashboard, “MQTT Dashboard.”, url: http://www.mqtt-dashboard.com/, last accessed in 14-Sep-2018.

ANEXO
A: OKW's Commercial Box External Measures

OKW's commercial box external measures. These measures were reproduced in 3D printed box projecting.



Annex A - OKW's Commercial Box External Measures