INSTITUTO
UNIVERSITÁRIO
DE LISBOA

EldyIoT - IoT Assistive System for Elderly

David Emanuel Magalhães Lourenço

Master in Computer Engineering

Supervisor:
PhD Octavian Adrian Postolache, Associate Professor with habilitation,
ISCTE-IUL

Co-Supervisor:
PhD Francisco António Bucho Cercas, Full Professor,
ISCTE-IUL

November, 2021

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

EldyIoT - IoT Assistive System for Elderly

David Emanuel Magalhães Lourenço

Master in Computer Engineering

Supervisor:
PhD Octavian Adrian Postolache, Associate Professor with habilitation,
ISCTE-IUL

Co-Supervisor:
PhD Francisco António Bucho Cercas, Full Professor,
ISCTE-IUL

November, 2021

Acknowledgments

In the first place, I would like to express my gratitude to Professor Prof. Doctor Octavian Postolache, my master thesis' supervisor, for the opportunity to work in this area. His continued support was fundamental to conclude this dissertation.

This research was supported by Instituto de Telecomunicações, IT-IUL that provide technical support and required equipment's and devices for my thesis.

I want to thank my family and friends, for the unconditional support, encouragement, and motivation that was given in recent years. Thank you all!

Resumo

A Internet das Coisas (IoT) é uma das tecnologias mais promissoras para o futuro próximo. A IoT penetrou em muitos setores, como cidades inteligentes, casas inteligentes, carros inteligentes, saúde, agricultura e manufatura. No sector da saúde, o número de casos de uso, tem aumentado como consequência do incremento na qualidade e eficácia na prestação de serviços hospitalares e de cuidados. Este trabalho, apresenta uma solução para reduzir a distância entre cuidadores e pacientes de forma não intrusiva, permitindo que informações sobre o estado de saúde e mobilidade dos pacientes estejam disponíveis e acessíveis em qualquer lugar, e a qualquer hora. Consecutivamente, aumentará a confiança dos cuidadores em relação ao estado de saúde dos pacientes acompanhados e, por outro lado, aumentará a confiança dos pacientes de que o seu estado de saúde está a ser monitorizado com frequência. Assim, esta tese apresenta um framework para monitorização física de idosos baseada em dispositivos smartwatch. Como parte integrante da solução, apresentamos uma aplicação móvel cross-platform e uma estrutura de software montada para dar suporte aos processos de recolher, processamento, armazenamento, exibição e análise de dados.

Palavras-chave: Smartwaches, Cuidados de Saúde, Doenças Crônicas, Atendimento/Cuidados Domiciliários

Abstract

The Internet of Things (IoT) is one of the most promising technologies for the near future. IoT has penetrated many industries such as smart cities, smart homes, smart cars, healthcare, agriculture and manufacturing. In the health sector, the number of use cases has increased as a consequence of the improvement in the quality and effectiveness of the provision of hospital and care services. This work presents a solution to reduce the distance between caregivers and patients in a non-intrusive way, allowing information about the health status and mobility of patients to be available and accessible anywhere, anytime. Consecutively, it will increase the confidence of caregivers in relation to the health status of monitored patients and, on the other hand, it will increase the confidence of patients that their health status is being monitored frequently. Thus, this thesis presents a framework for physical monitoring of elderly people based on smartwatch devices. As an integral part of the solution, we present a cross-platform mobile application and a software framework assembled to support the processes of collecting, processing, storing, displaying and analyzing data.

Keywords: Smartwatches, Health Care, Chronic Diseases , Home Care

Contents

List of Figures

List of Tables

List of Acronyms

API - application programming interface
CRUD – Create, Update, Delete
BPM - Beats per minute
SQL - Structured Query Language
NoSQL - No Structured Query Language
JSON - JavaScript Object Notation

CHAPTER 1

Introduction

1.1 Motivation and Overview

In recent decades, we have witnessed an increase in the average life expectancy of the world's population to figures never seen before. In 2015, for the first time, the world average mark of 60 years was reached. And according to some studies, by 2050 the elderly population is expected to double and more than triple by 2100 [1]. One out of every six people is aged 60 or over, and in the near future we can expect to reach the one out of five mark [2]. In Europe, 25% of the population is aged 60 and over. In particular, the aging population creates increasing health welfare problems and care healthcare system costs. With the increase in the aging population, healthy aging can no longer be an ignored issue. Most health problems faced by older people are associated with chronic diseases. Many of these can be prevented or delayed by engaging in healthy behaviors. Other health problems can be effectively controlled, especially if they are detected early enough. Even for people with declining health capacity, supportive environments and monitoring can ensure that they live dignified lives with continued personal growth [2].

There is no such thing as a typical elderly person. Aging is not linear and each organism ages differently. [3]. Therefore, monitoring the health status of the elderly plays an extremely important role in improving their quality of life. These trends make the integration of new technologies and innovative solutions necessary and inevitable to meet new challenges such as the growing aging of the population and the consequent pressure on health systems. The follow-up and monitoring of elderly people are associated with periodic consultations and visits to doctors. . The possibility that there may be some symptom or change in the patient's health status that is not felt at that exact moment, ends up camouflaging a diagnosis that could be important and easy if accompanied with data collected by wearable devices. The IoT is proving to be the solution to many of the problems we face today. Thus, the manufacturing industry, and the healthcare industry are two examples of industries with great potential when it comes to transforming and reinventing processes. The possibilities of improving classical medicine and the well-being of humans are a reality more and more present in our days [4]. Healthcare has been one of the areas of greatest investment because of the recognition of value and the potential economic impacts that IoT can bring [5].

As we can see in Figure 1, the cost savings and productivity gains generated through “smart” device monitoring and adaptation are projected to create $0.2 trillion to $1.6 trillion in value in the health care and human sector, $1.2 trillion to $3.7 trillion in operation management and factories global manufacturing, and 0.9 to $3.7 trillion in municipal energy, service provision, traffic control and resource management over the next decade. The Internet of Things offers a global potential economic impact of $4 trillion to $11 trillion a year in 2025 [5].

Figure 1 - IOT economic global impacts [5]

The trend to make the medical industry more digital can have direct impacts on the quality of service provided, the effectiveness of diagnoses and subsequent treatment. The continuous follow-up of elderly and chronically ill people become possible with the constant monitoring and collection of data ready for analysis. Although currently, the vast majority of data from IoT systems does not serve to add value, we have a strong working tool to support important decision making for caregivers and physicians [5].

1.2 Objectives and Research Questions

The objective of this thesis is to develop a system for monitoring the health status and daily activity for elderly people. The system is characterized by hardware and software. The hardware materialized by smart devices will be responsible for collecting data that will later be processed and stored by using developed software components . As hardware, a wearable device will be used, to be the least intrusive possible. Thus, a smartwatch-type device will be used. The intention with this is to collect as much information as possible so that we can obtain important data about the patient's health status without the patient having to worry about it.

For the configuration and visualization of patient data by caregivers, a mobile application has been developed. The data can be viewed graphically and in a medium/long term perspective to facilitate analysis and comparison.

1. What are the advantages of having an elderly monitoring system?
2. What are the advantages of storing elderly physiological measurement data from smartwatch sensors?
3. Is it possible to detect abnormalities in a patient's health status with the aid of smartwatch sensors?
4. Is it possible to perform monitoring of an elderly person in a non-intrusive way?
5. Is it possible to create a monitoring system for the elderly with the aid of a smartwatch designed for fitness monitoring?

1.3 Research and Planning Methods

As a research method, the Design Science Research [6] method was used.

The purpose of adopting a DSR process was to provide a model for problem identification, goal definition, design, preparation, demonstration, evaluation, and communication of the research result. The authors Peffers and Tuunanen in [6] suggest using a model with six sequential activities:

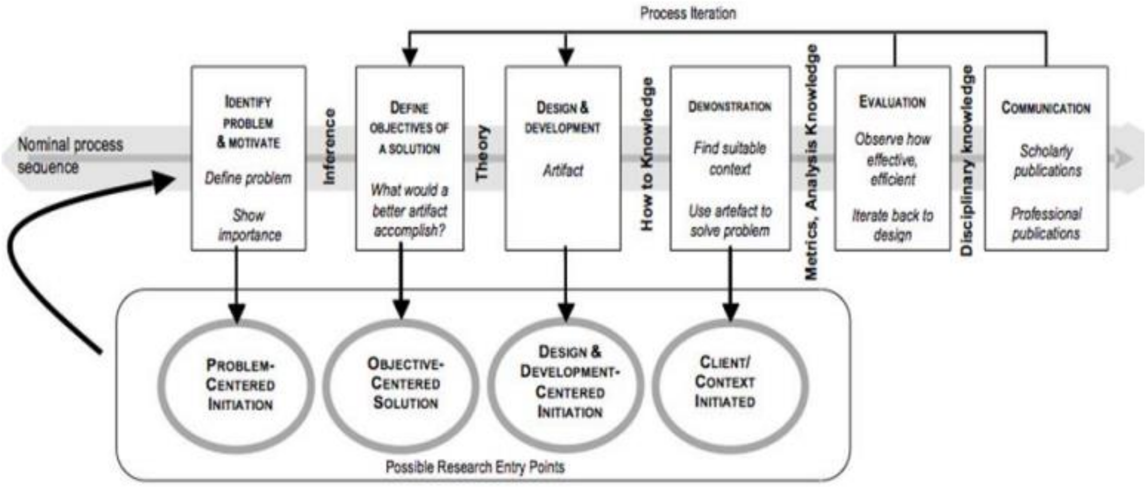


Figure 2 – Design Science Research Methodology Process Model [7]

- Phase 1 - Problem Identification - defining the problem and justifying the value of a solution.
- Phase 2 - Goal and Solution Definition - Deducing the goals of a solution from the problem definition and knowledge of what is possible and feasible.
- Phase 3 - Solution Design and Development - Activity that includes determining the desired functionality of the artifact and its architecture, and then creating the actual artifact.
- Phase 4 - Demonstration - Demonstrating that the use of the implemented artifact, solves one or more instances of the problem.
- Phase 5 - Evaluation - Observing and concluding how well the artifact supports a solution to the problem. This activity involves comparing the objectives of the solution with the actual observed results of using the artifact in the demonstration.
- Phase 6 - Communication - Communication of the problem and its importance. Communication of the artifact: of its usefulness and novelty, the rigor of its design, and its effectiveness for

In DSR methodologies, phases 2, 3, 5 and 6 work in an agile and cyclical way. With this, at the end of either phase 5 or phase 6, there is the possibility of returning to phases 2 or 3 to rethink the solution, the objectives, the architecture developed, or to adjust the implementation.

1.4 Structure of the Thesis

This thesis consists of 7 Chapters. Chapter 2 includes a brief introduction to the topics covered and a literature review. Chapter 3 describes the architecture and functioning of the proposed system. Chapter 4 introduces the software that supports the application. Chapter 5 presents the hardware selected for the development of this system. Chapter 6 explain the Results and Discussions and Chapter 7 presents the conclusions.

CHAPTER 2

State of the Art

2.1 Overview

This chapter describes in a theoretical way, some scientific concepts of importance for the thesis under study as well as the state of the art and literature review.

The increasing rate of the world's aging population has become an increasingly important issue. Population aging is a worldwide trend and an increasingly serious problem. In the following figure we present a image with the evolution of the number of people over 60 years old.

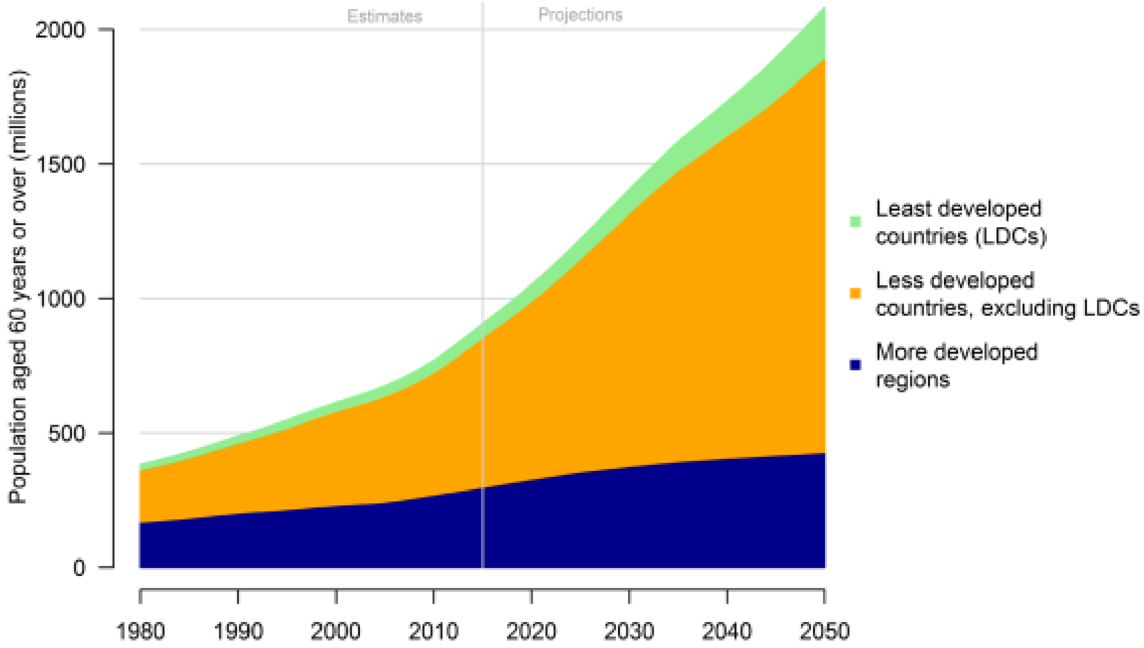


Figure 3 - Number of persons aged 60 years or over by development group,1 from 1980 to 2050 [8]

Among other factors, the increase in the population over 60 years of age creates growing problems in health care needs and care costs, increasing pressure on social and health care systems. Population aging is being felt in different regions of the globe, and it is estimated that by 2030, 34 nations will have reached the standard for an over-aged society, and that by 2050, 21.5% of the world's population will be 60 years of age or older [9].

Most older adults prefer to live at their home, thus with technological assistance that maintains and supports their independent, active, healthy, and functional lives rather than going nurse house or hospital. With technological advancement, we have been able to achieve the goal of person-centered care by providing older adults with easily accessible tools for aging at home, which also improves their quality of life. The main characteristics of home care are personalized care, respect for the value of the person, and the creation of a social environment that supports psychological needs. Since it is known that mental health influences physical health. Caregiving is an activity that involves work and emotions and can be called a "labor of love. The concepts of home care and aging at home have been valued and studied in countries around the world; telemedicine system implementations and the acceptance of geotechnology have also attracted the attention of scholars. This, however, remains an area in need of evolution and research [9].

2.2 Literature review

2.2.1 IoT

The Internet of Things (IoT) describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. With more than 10 billion devices connected today, the number is expected to rise to 22 billion by 2025 [10].

Despite the current media hype, the internet of things is not a new concept. Although unaware of the power of the concept, the first Internet of Things device was a Coca-Cola machine at Carnegie Mellon University in the early 1980s. Via the web, programmers could check the status of the machine and determine if there would be a cold drink waiting for them if they decided to make the trip to the machine.

Despite the advances in the 1980s, it was in the late 1990s, Kevin Ashton had and documented the vision of what would become the foundations of the Internet of Things (IoT) while preparing a presentation for Procter & Gamble. The concept was simple, but powerful.

"If we had computers that knew everything there was to know about things - using the data they collected without any help from us - we would be able to track, count, reduce waste, loss and cost. We would know when things needed to be replaced, repaired, and whether they were fresh or past their best condition. We need to empower computers with their own means of gathering information, so they can see, hear, and smell the world for themselves." RFID technology and sensors allow computers to observe, identify and understand the world - without the limitations of human-sourced data." [11]

At the time, this vision required major technological improvements and humanity was not ready for them. Questions quickly arose regarding means of communication, infrastructure, relationships between devices, among others. There were more questions than answers for everything involving the IoT concept in 1999. Currently, many of these obstacles have been solved and we have seen a great evolution in this area.

Over the years, some technological advances have made IoT possible, among them:

- Access to low-cost, low-power sensor technology. Affordable and reliable sensors are making IoT technology possible for more manufacturers.
- Connectivity. A host of network protocols for the internet has made it easy to connect sensors to the cloud and to other “things” for efficient data transfer.

- Cloud computing platforms. The increase in the availability of cloud platforms enables both businesses and consumers to access the infrastructure they need to scale up without having to manage it all.
- Machine learning and analytics. With advances in machine learning and analytics, along with access to varied and vast amounts of data stored in the cloud, businesses can gather insights faster and more easily. The emergence of these allied technologies continues to push the boundaries of IoT, and the data produced by IoT also feeds these technologies.
- Conversational artificial intelligence (AI). Advances in neural networks have brought natural-language processing (NLP) to IoT devices (such as digital personal assistants Alexa, Cortana, and Siri) and made them appealing, affordable, and viable for home use, [10].

With the unlocking of the technology needed to put this powerful idea into practice, the difficulties previously encountered in putting the IoT into practice have become fewer. As such, many use cases have emerged and have been implemented since then [12] .

- In robotics, wide efficiency in the processing of numerous sensors data combined with the sophistication of mechanical actuators and networks that connect them in the production of machines that aim to be completely and/or almost completely autonomous in controlling their actions in the physical world. Autonomous vehicles are an example [12] .

- In Mechanical Engineering, where sensors and actuators used by weapons, robot and exams, remote medical diagnostics and surgical applications that make use of 'haptic sensory technologies' are great examples [12] .

- In Healthcare, where sensors embedded in clothing collect information about the patient's health status, communicating wirelessly with facilities run by doctors and researchers, and receiving new instructions in real time for controlling the dosage of medication to be administered based on the patient's needs [12] .

- In Ecology and Environmental Sciences, where an extensive network of sensors and actuators can control and manage air and water quality, where weather forecasting and climate measurement are used to improve agricultural production [12] .

- In production process control, one of the earliest and most mature uses of IoT, evolves through application advances in robotics, computer-aided design (CAD), and computer-aided manufacturing (CAM). Just as many data centers work autonomously, in the future we think so will production and supply control [12] .

Another case of applicability of IoT in production processes are automotive assembly lines and industrial assembly lines where an error can create a large loss.[10]

- Transportation and logistical systems benefit from a variety of IoT applications. Fleets of cars, trucks, ships, and trains that carry inventory can be rerouted based on weather conditions, vehicle availability, or driver availability, thanks to IoT sensor data. The inventory itself could also be equipped with sensors for track-and-trace and temperature-control monitoring. The food and beverage, flower, and pharmaceutical industries often carry temperature-sensitive inventory that would benefit greatly from

IoT monitoring applications that send alerts when temperatures rise or fall to a level that threatens the product. [10]

The Internet of Things has taken on an increasingly important role when it comes to increasing and improving the quality of human life. As such, it has been a strong object of acceptance. The increase in technological capacity in terms of infrastructure and supply of devices with different utilities, has increased the number of connected devices almost exponentially. In the following figure we can observe the evolution and almost exponential growth of the number of connected devices in the last years.

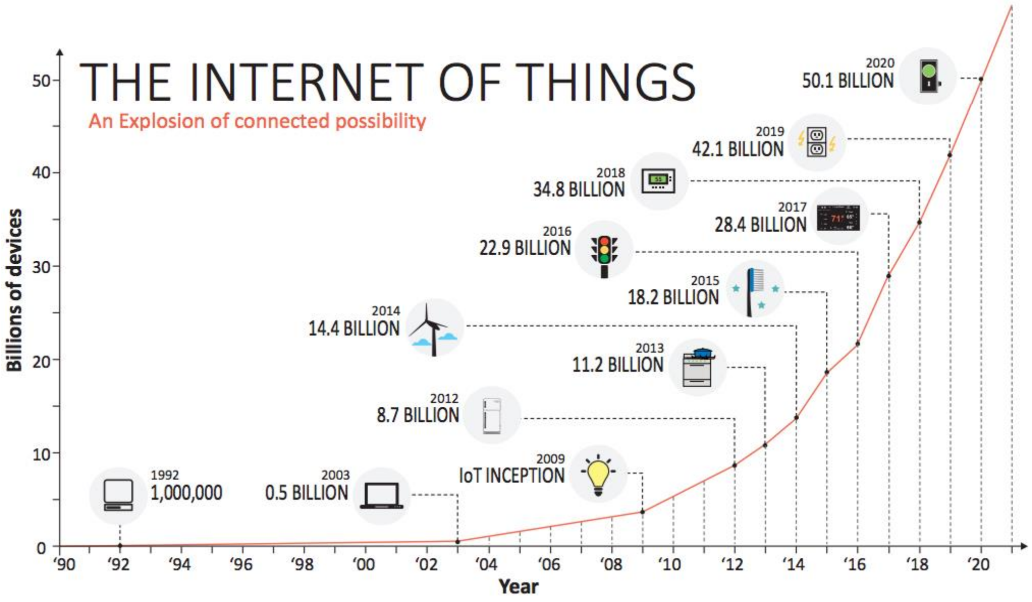
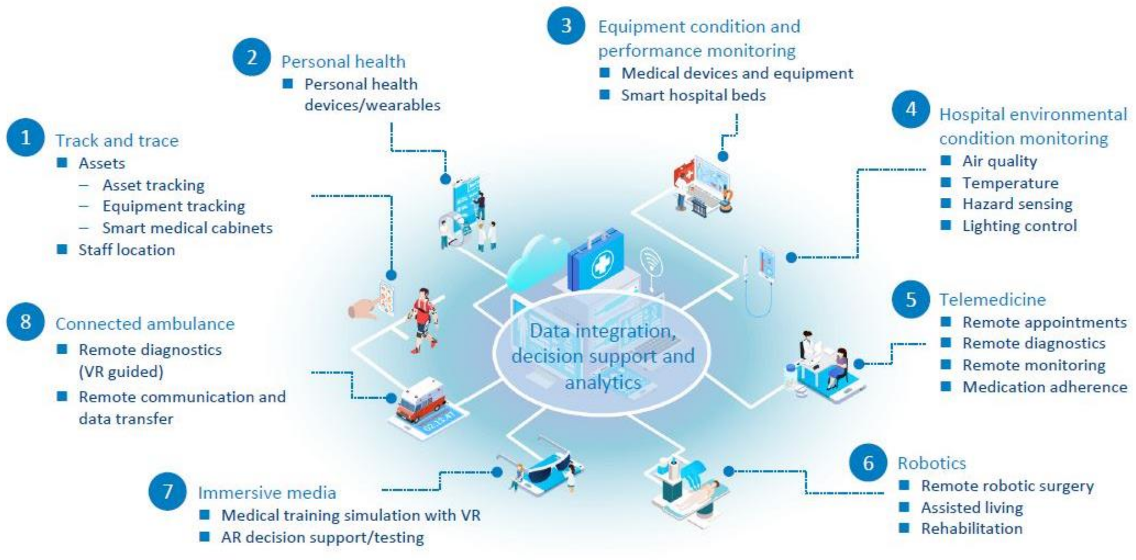


Figure 4– The connected devices evolution [13]

2.2.2 IoT for healthcare

The Internet of Things (IoT) has penetrated many industries: smart cities, smart homes, agriculture, manufacturing, museums, events, smart cars, and the healthcare industry. There are many Internet of Things use cases in Healthcare and this is one of the fastest growing application fields when it comes to embracing and utilizing the benefits of the Internet of Things to its full power. [13]

In figure 5, we can see a diagram explaining some current uses of IoT applied to healthcare. Being Healthcare one of the industries with the most room for process redesign, we can observe, several applicability's of the technology in small different areas within healthcare. IoT is useful not only in monitoring and assisting patients, but also, in facilitating the work of nurses, doctors, and caregivers. There are case studies of VR simulation systems to train doctors in specific situations such as operations [14] . There are case studies of robots capable of assisting and performing surgery [15], as well as there are case studies for more basic functions such as monitoring ambient conditions within hospitals, waiting rooms and operating rooms, Telemedicine, and collecting health data from patients without them having such awareness [16].



Source: Arthur D. Little analysis

Figure 5-– IoT contributions on healthcare diagram [17]

Waerness and Ringer [9] asserted that caregiving is an activity that involves both labor and emotions. By its nature, caregiving is tedious, monotonous, and time-consuming labor that requires constant and immediate response; in addition, the contributions are often ignored. The increasing demand for home care, along with a rapid growth in the older adult population, cannot be satisfied solely by caregivers and resources from governmental agencies. Through the aid of technology and wireless networks, systems and services can be properly developed for older adults, providing them with healthy, safe, independent, comfortable, and active aging lifestyles. [9]

IoT comes to health care to achieve the goal of reducing costs associated with the treatment and monitoring of patients, aggravated by the aging population. Besides the economic aspect, this partnership will help the human aspect, helping the active staff, making their task easier and allowing them to reach more people, with less difficulty. For the patient, the IoT also brings some benefits since, given the high volume of data collected on their health status, it becomes possible to make the care provided, more personalized and centered on the patient.
In sum, healthcare presents one of the most potential and attractive areas for the implementation of IoT solutions, especially in remote health monitoring, chronic diseases and elderly care. In a general point of view, the focus of IoT technologies in healthcare
is based on the prevention, early pathology detection and home care. [18]

2.2.3 Wearable tecnologies

Wearable technology or wearables are the group of intelligent electronic devices (electronic device with micro-controllers) that are worn close to and/or on the surface of the skin, where they detect, analyze, and transmit information concerning e.g. vital signs, and/or ambient data and which allow in some cases immediate biofeedback to the wearer. [19]

With technological advances and the ability to produce sensors at increasingly competitive prices, nowadays we can find wearable devices in almost every object we use in our days. In figure 6, we have some examples of wearable devices present in today's market.

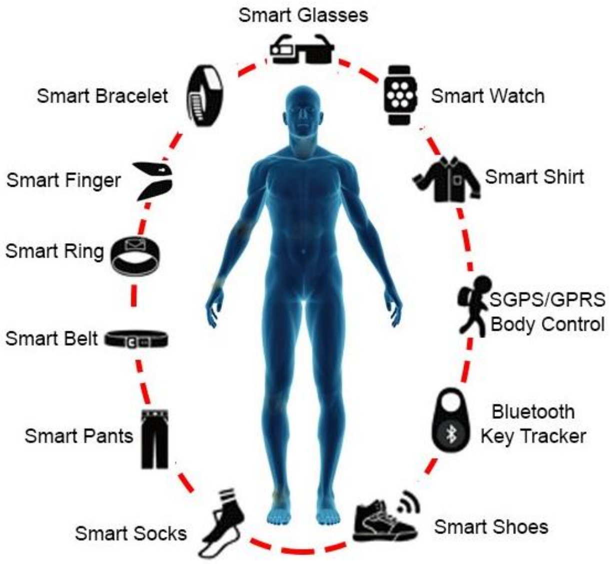


Figure 6 – Wearable tecnologies [20]

Examples of Wearable Technology

The past few years have seen rapid development and introduction of wearable technology products adapted for medical and healthcare uses. These include:

- In Louisville, Kentucky, wearable devices made by AIR Louisville are being used to monitor local air quality, measure pollutants, and identify hotspots for residents with respiratory problems [21].
- Cyrcadia Health has developed iTBra, an intelligent patch that can detect early signs of breast cancer and transmit the information to a lab for analysis[22].
- Wearable medical alert monitors are extending greater mobility and independence to the elderly and impaired [23].
- Smart tattoos that contain flexible electronic sensors are being developed to monitor heart and brain activity, sleep disorders, and muscle function. While these are temporary, even inks are being explored [24].
- A smartwatch for people with Parkinson's disease tracks symptoms and transmits the data so that more personalized training plans can be developed. [23]
- Child monitoring devices equipped with GPS are available from many manufacturers for as little as $25. [23]

2.2.4 Databases

With the advance of technology, databases and their variances have been meeting the needs that new challenges have posed. As such, there are several types of databases, each with its pros and cons. Databases can be divided into two major groups. Relational and non-relational.

Relational Database:

Relational databases are based on the relational model, an intuitive and straightforward way of representing data in tables. Conceptually, in a relational database, each row in the table is a record with a unique ID called a key. Each key is unique and the table columns contain attributes of the data and each record usually has a value for each attribute, making it easy to establish relationships between data points. [25]

Examples: MySQL, SQLite and MariaDB

Non-Relational Databases:

Non-relational databases are approaches that do not use the tabular row and column schema found in the type of databases described earlier. Instead, they use a storage model optimized for the specific requirements of the type of data stored. For example, data can be stored as simple key/value pairs, as JSON documents, or as a graph consisting of edges and vertices. [26]

Examples: MongoDB, Oracle NoSQL Database, DocumentDB.

In table 1, we present some advantages and disadvantages considered during the research and study of the database to be chosen for the development of the proposed solution.

Database	**Advantages**	**Disadvantages**
MongoDB	-Distributed database. -Flexibility in insert, delete and update operations due to the absence of dependency rules between tables. -GridFS - functionality responsible for storing large files. [27] -High scalability High Availability for data-intensive operations. [27]	Possible difficulty in learning a new and different approach.[27]

MySQL	-Fast and Secure[28] -Highly Compatible with Hosting Systems -Free version. [28] -It has many features, even considering that it is a free database. [28]	-In order to have access to support within the free option, you must pay. [28] -Free version does not offer Tread Pooling, decreasing the availability of database connection. [28] -Low migration compatibility with other databases. -Very expensive paid version. [28] -Reference - SQLite vs MySQL - What's the Difference and Which One to Use [28] -Dependency relationships between tables that can hinder operations (crud) -Difficulty in handling large data volume transactions.
MariaDB	-The System is fast and stable. -It offers Tread Pooling and support 200.000 simultaneous connections [28]	-No guarantee for updates and additional versions. -Not supported. -JSON data type support as of version 10.2 -Difficult to handle large data volume transactions.[29]
SQLite	-Small size database. -Operates without the need for a server. Runs as part of the application -Suitable for development and testing [28]	-Does not have security and user management features. -Not suitable for scalability. [28] -Not suitable for handling large volumes of data. [28]

Table 1- Advantages and disadvantages of the databases under study

For the study of the database to be chosen, 4 different types of databases were analyzed. Three relational databases and one non-relational database. Relational databases are closely tied to business rules and dependencies between tables. This on the one hand is important in applications where the need to create relationships between tables is apparent, but on the other hand is of low value when dealing with data structures with poor relationships between the parts involved. Given the nature of the application, the key point in choosing the type of database to use rested on 4 important pillars:

- Scalability
- Reliability
- Availability
- High performance for a large volume of data

As the number of users increases, the database will be increasingly called upon with insert requests and basic information queries. Scalability plays an important role in ensuring that the application can grow without major problems, availability ensures service satisfaction, and high performance with large amounts of data will ensure fast interaction and data exchange between the parties interacting with the database.

Over the past few decades, the volumes of transacted data have increased exponentially. Non-relational databases have been running after the damage in order to counteract the difficulty in dealing with these large volumes. NoSQL databases emerge as a solution to meet this need and have been gaining more and more relevance[27].

After analysis, and in response to the requirements necessary for the good performance of the system to be implemented, we decided to implement a mongoDB type database.

2.2.5 Backend Technologies

In most applications, the backend is the crucial part for the correct functioning of the application, so during the planning of the applications great attention is paid to the choice of technologies and solutions to be adopted.

For the implementation of the proposed system, it was necessary to study technologies that were flexible enough to be able to function as a cross-platform technology. The goal was to run the backend independently of the target machine's operating system. In Figure 7, we present the top 5 popular backend technologies.

Considering this, some technologies were studied and next some conclusions regarding the study will be presented:

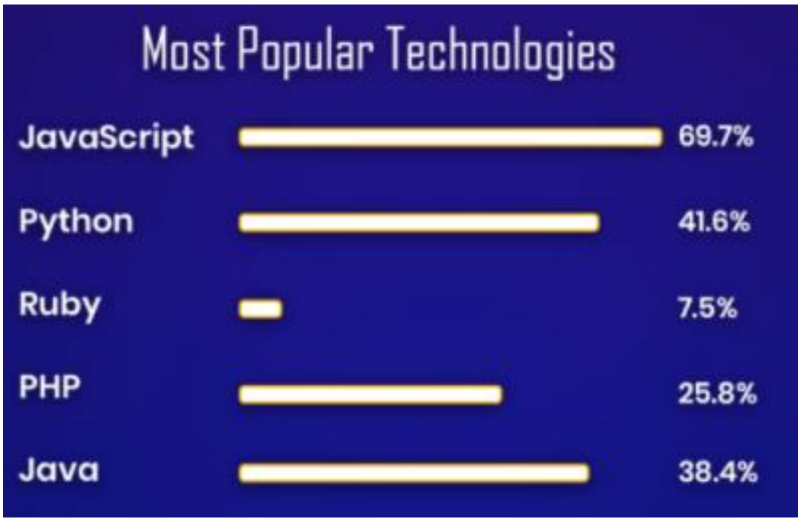


Figure 7- Top 5 backend technologies [30]

Java - Java is considered a multipurpose programming language. The availability of numerous frameworks makes it even more demanded. It has a good reputation among IT specialists. Java is good for large enterprise projects that require complex computations and difficult data processing but needs a JVM-ready environment to compile and run the application in question. Given this limitation, we decided that this would not be the best approach [31].

PHP – PHP is widely used as a backend language for web development but performs best for small to medium development projects. Its architecture is event driven. In terms of performance, PHP loses out to languages like JavaScript. Both languages make use of the File System to interpret files. The difference is that JavaScript processes the next request while waiting for the response from the previous one and PHP waits for the response [31] [32].

Python - Python is considered an easy language to learn. It is very popular when it comes to medium to large application development but in some cases, it may not be the highest performing solution when it comes to web and cross-platform development. Compared to other technologies like Java and JavaScript, Python has much lower response times for server-side operations [33][31].

Ruby - It is best suited for both small and medium sized projects. When comparing performance with technologies like JavaScript, the differences are noticeable especially in response times as a server. Ruby is an interpreted tool, which means that the code is not compiled into a machine version. Therefore, it is more difficult for hardware to read and process quickly. On the other hand, Node.js offers a real-time compiler that allows translating the code directly into machine code[34].

Node.js - As we can see from the previous figure, JavaScript is taking over as the most popular language for building backend applications.

Node.js is a server-side framework and works well with real-time handling of large amounts of information. This JavaScript framework is surrounded by a very large number of frameworks that satisfy most of the language's needs. Some examples are :

Express.js - framework responsible for web server exposure.

Mongoose.js - framework for interaction and management of mongo databases.

Axios.js - HTTP client framework.

Based on the previous study, the language that offered us the most guarantees was JavaScript, and so we chose to make use of the Node.js framework features.

2.2.6 Frontend Technologies

With the technological advances we witness daily, the search for strategies that make developers' work easier has been on the rise. Regarding the development of mobile applications, this is no exception. In figure 8, we can observe the growing interest in cross-platform application development.

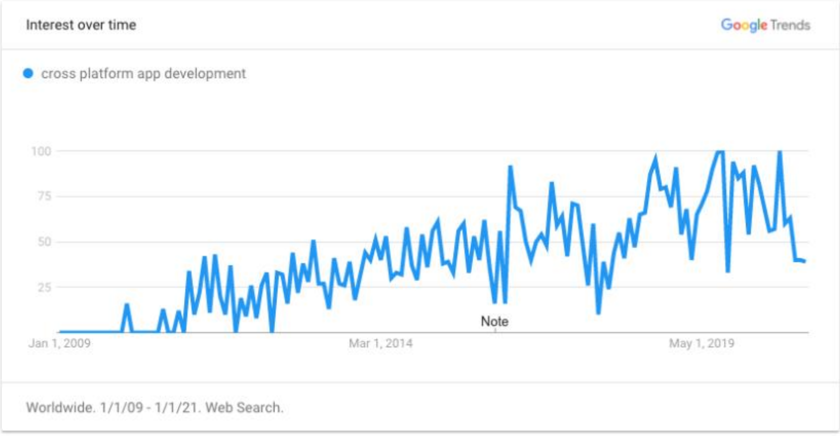


Figure 8– Cross-platform interest over time[36]

After some research and analysis of several studies [*35]*[37][38] , React Native is pointed out as the best solution for cross-platform application development in almost all of them. React Native is a JavaScript-based mobile application framework that allows you to build native-rendered mobile applications for iOS and Android. This framework gives the developer the possibility to save time by writing code optimized to run on many different platforms. Such as android and iOS. React Native was first developed and used by Facebook in the development of their application and subsequently, other large companies worldwide such as Uber, Skype, Instagram and Pinterest have taken advantage of its benefits.

Given the evidence of React Native throughout its history, after research, we decided to use it. Since we are dealing with a JavaScript based framework, the integration with node.js becomes easier.

2.2.7 Ambient Assisted Living (ALL)

One of the most useful IoT scenarios are quality of life care scenarios. Sensors and health monitoring devices are used to collect and make useful information available to doctors or other family members to improve treatment and responsiveness [39]. An Ambient Assisted Living (AAL) can be implemented through an IoT platform. An AAL is designed to provide better quality of life and access to healthcare services for these patients, in order to extend their independent living in the comfort of their homes [40]. By allowing patients to carry medical sensors to monitor different parameters, such as body temperature, blood pressure and respiratory activity, it enables personalized health care, tracking users' daily activities and making suggestions to improve their lifestyle and prevent health problems [41].

2.2.8 Smartwatch and IoT Applications for Healthcare

In the first two surveys, two solutions for monitoring dementia patients are presented. Many of these patients receive home care to maintain their independence. Mobility and physical activity are very significant points when it comes to the independence and quality of life of patients and the non-progression of the disease. Increasing age and cognitive impairment are closely linked to reduced mobility. Therefore, wearable devices become great potential allies for monitoring mobility-related parameters in a non-intrusive way [42], [43]. In addition to the poor control regarding paraments such as mobility, the treatment of dementia as well as the treatment of most chronic diseases follows an unwanted approach. Patient monitoring and reporting of physical and cognitive decline depend on visits by health care providers (doctors, general practitioners, nurses) or on the patient going to the hospital. Given the non-constant nature of these pathologies, the method of medical monitoring has obvious limitations. Given this scenario, it is becoming increasingly clear that constant monitoring of patients in recovery can be a great asset in their recovery, treatment, or stabilization of the disease. As such, data collection and analysis become a point of extreme importance. Costas Boletsis and Simon McCallum [42] have proposed a smart system that can provide additional, potentially useful information to dementia patients by taking advantage of a smartwatch to record parameters such as blood oxygen, body temperature, sweat quality, ambient temperature as well as data from an accelerometer, [42].

Taking advantage of wearable devices, smartwaches have assumed a prominent role in this research, being part of most of the analyzed studies. In the same context, but this time not so focused on disease management in particular, but rather on health care at home, S. Feng and S. Chiou [43] created a monitoring and assistance platform consisting of 4 modules that among them distribute functions such as scheduling daily care services, scheduling medication delivery, notifications, appointment scheduling, messaging, an emergency button and a module responsible for collecting health parameters such as blood pressure, blood sugar, body weight and blood alcohol. [43] In the same direction of thought comes the ROAMM [25] positioning itself also as a monitoring and assistance platform with monitoring of parameters like heartbeat, and GPS data but which in turn enriches the knowledge about the user's health status with programmed requests for qualificative inputs of parameters like pain, fatigue and mood. In a slightly different technological vein comes the combination of indoor location monitoring and wearable devices for the rehabilitation of elderly people. The framework in question combines proximity sensors with smartwaches that, based on signal exchanges, have the intelligence to calculate the user's distance to landmarks scattered around the house. Thus, it becomes possible to map their indoor movements and infer their daily activity. [44] A Wrist sensors [45] emerges with a slightly different paradigm and presents its greatest contribution for being an application capable of pairing with compatible devices, collecting data that can serve as an object for study for the evolution of intervention algorithms and possible detection of possible health problems of the monitored patients.

Technological development and the constant evidence that IoT can be a strong ally in improving patient care, create ideas that there is room for improvement in the processes involved in these activities. As such, in [46] a cloud computing-based approach is proposed to integrate all hospital records of a patient and propose to keep them centralized. The proof of concept of this proposal was made with the integration of a temperature sensor with the rest of the system, centralizing the patient's temperature data in a database. The data collected by the sensors is subsequently accessible and displayed on a web page accessible via a browser. This way, a process that would have to be done by medical staff, became automatic and available for consultation almost immediately. In [24] proposes a system that re-enforces the argument that emphasizes the importance of collecting and analyzing patient data. The monitoring and collection of patient health status information takes the burden off the health systems, saving time for doctors and nurses who can focus on analyzing the results and making the service more efficient. This system collects heartbeat, blood pressure, and ECG data with the help of sensors. After collection and processing, if any abnormalities are found, alerts are sent to the physician responsible for the patient and the history of the data is made available on a web page. [47]

CHAPTER 3

System Framework

3.1 Overview:

This chapter will present a detailed description of developed system for this thesis, as well as all the systems that somehow integrate with the solution.

In section II, the system architecture will be presented, and each component will be explained in detail. Detailing their purpose and importance in the implemented system. In section III, a flow diagram will be presented so that we can have an idea of the course of actions throughout the functioning of the framework.

3.2 Architecture Description

The EldyIoT framework includes a hardware block and a software block. The hardware combines a set of functions capable of collecting data from the patient. The data is collected by a set of sensors embedded in smartwatches. As an integral part of the solution, we have a back-end structure built on a cloud infrastructure capable of supporting the process of data collection, processing, storage and subsequent availability for consumption by the application. The data used in the framework is data obtained from the API provided by Fitbit. These data are stored in a remote database and made available for consumption via the API. As a data visualization mode, we have a mobile application capable of offering a visualization of the collected data in a personalized, simple way and in the form of graphs for easy understanding. It is possible to observe the data in a timeline of up to 3 months in order to have a medium/long term perspective.

The structure of the EadyIoT framework can be divided into 4 different blocks, with different functions to perform, but working towards a common goal. In Figure 9 we present the EldyIoT Architecture Block Diagram.

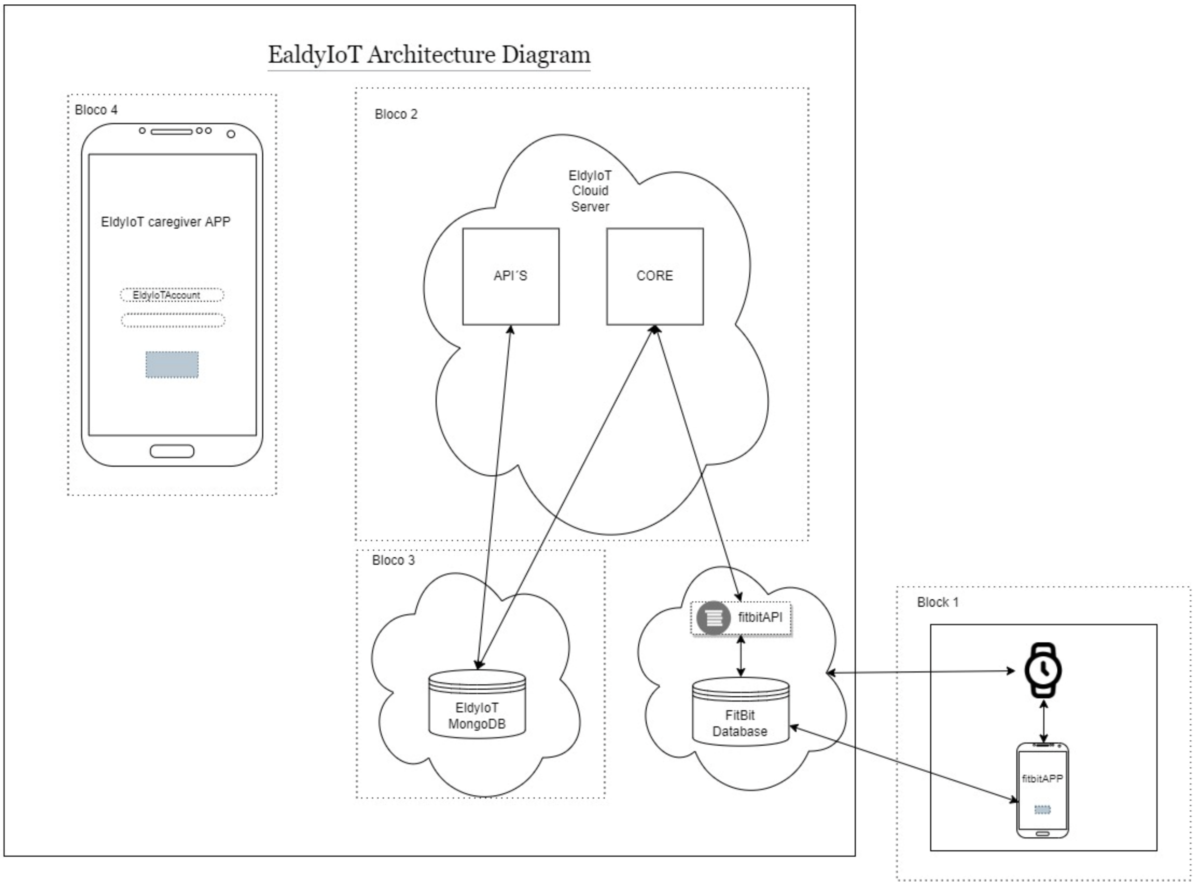


Figure 9-– EldyIoT Architecture Block Diagram

3.2.1 Fitbit Infrastructure

Block 1 described in Figure 10 is responsible for data collection and is part of the structure provided by Fitbit. This block includes the smartwatch Fitbit Versa 3 that we decided to use, the Fitbit database and the Fitbit smartphone application. The information flow occurs from the smartwatch to the mobile app via Bluetooth and from the mobile app to the Fitbit database via the internet. This pairing and synchronization can be forced in the application, but if not, it is done periodically every 15 min.

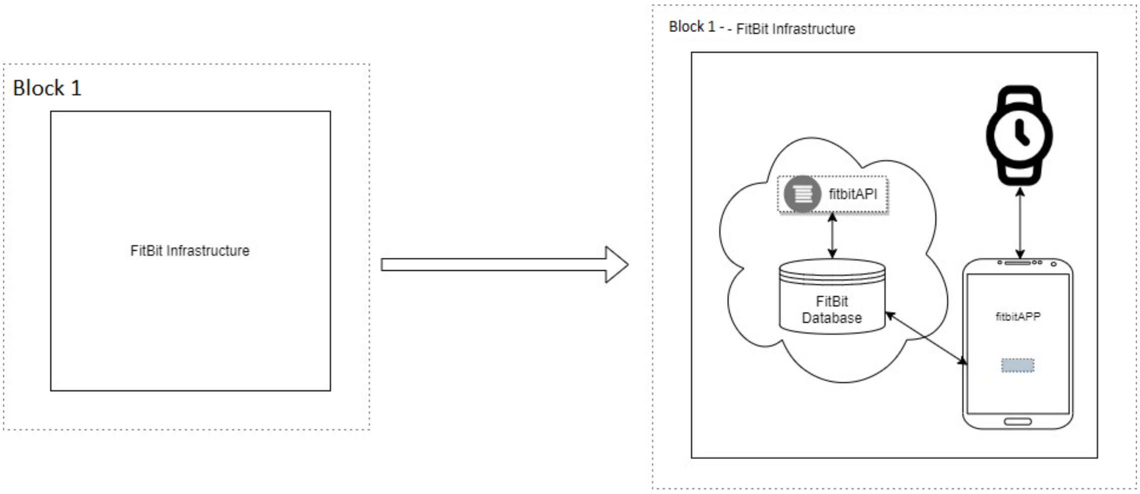


Figure 10– EldyIoT Block 1 block diagram

The Fitbit Versa 3, can collect:

- information regarding distances traveled,
- burned calories ,
- heart rate (HR) beat per minute (bpm),
- climbed floors,
- quality of the patient's sleep.

The data are collected via the smartwatch and are available for consultation through API.

3.2.2 Data preparing and processing

The data preparing and processing is performed at the block 2 level described in Figure 11

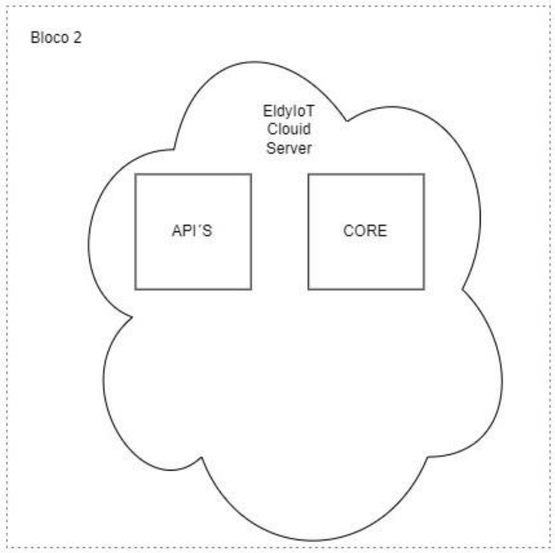


Figure 11– EldyIoT data preparing and processing - Block 2

Block 2 is the main block of the framework. This is where data is consumed via the Fitbit API, processed, and prepared to be stored in the framework's database. This block is running on a cloud server and is divided into two modules, the CORE module, and the API module. The CORE module has two processes running periodically to keep the information in the framework database consistent and up to date with the data collected by the smartwatch and stored in the Fitbit database. The first process is responsible for updating information regarding distances traveled, calories burned, heart rate per minute, floors climbed and runs every 3 min. The second process is responsible for querying and updating the long-term query data used for presentation in graphical way.

3.2.3 Data Storage

The data storage is presented in Block 3. It is in this block that the data that is collected by the smartwatch's built-in sensors is stored. Before storing the data, it is processed, and we only extract the important and relevant information for the framework.

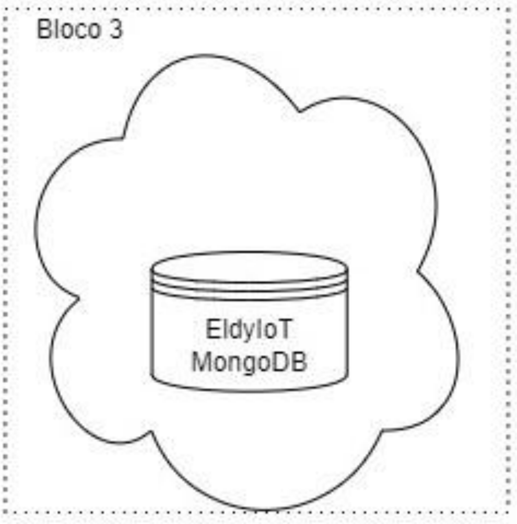


Figure 12- EldyIoT data storage - Block 3

3.2.4 Data Visualization

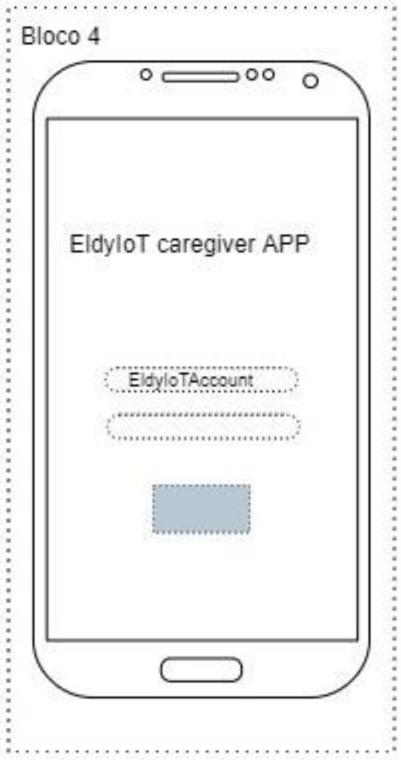


Figure 13- EldyIoT data visualization - Block 4

Block 4 is presented in Figure 13 and represents the data visualization and presentation block of our framework. This data visualization is done via mobile application and gives caregivers the possibility to access information extracted via built-in sensors quickly and as simply as possible. It is possible to visualize daily patient data as well as to make long-term analyses with the aid of graphs. The possibility of a mobile application solution arose in order to make the process of information extraction and monitoring as quick and simple as possible. Smartphones have assumed an increasingly predominant role in our lives and have been seen as a source of solution for many problems we have throughout the day, [48] as such, we wanted to take advantage of that in our framework as well.

Since a patient is often in the care of several caregivers, the application supports multi-caregivers. Multiple caregivers, provided they are authorized to do so, can access information about the patient's health status.

3.3 System Data Flow Diagram

In figure 14, the flow diagram of the information flow in the system is expressed. The execution of the process, includes the following steps:

I. Configuration of the kit with the pairing of the Smartphone with the Smartwatch via Bluetooth and respective login in the Fitbit app with the email provided in the EldyIoT kit.

II. Start of the use of the Smartwatch by the patient and start of data collection.

III. Data transfer from the smartwatch to the smartphone automatically.

IV. Transfer of data from Smartphone to Fitbit database automatically.

V. Periodic consumption of the data in the Fitbit database via API.

VI. Processing the data from the Fitbit structure and storing them in the EldyIoT database.

VII. Login in the EldyIoT application by the caregivers with the accounts provided for this purpose.

VIII. Consumption of patient data in monitoring via the EldyIoT framework.

IX. Consulting the data from the Fitbit database.

Note: steps VIII and IX are bidirectional since the information flow is in both directions. First in the consultation of the data and second, in the transmission of the data coming from the consultation.

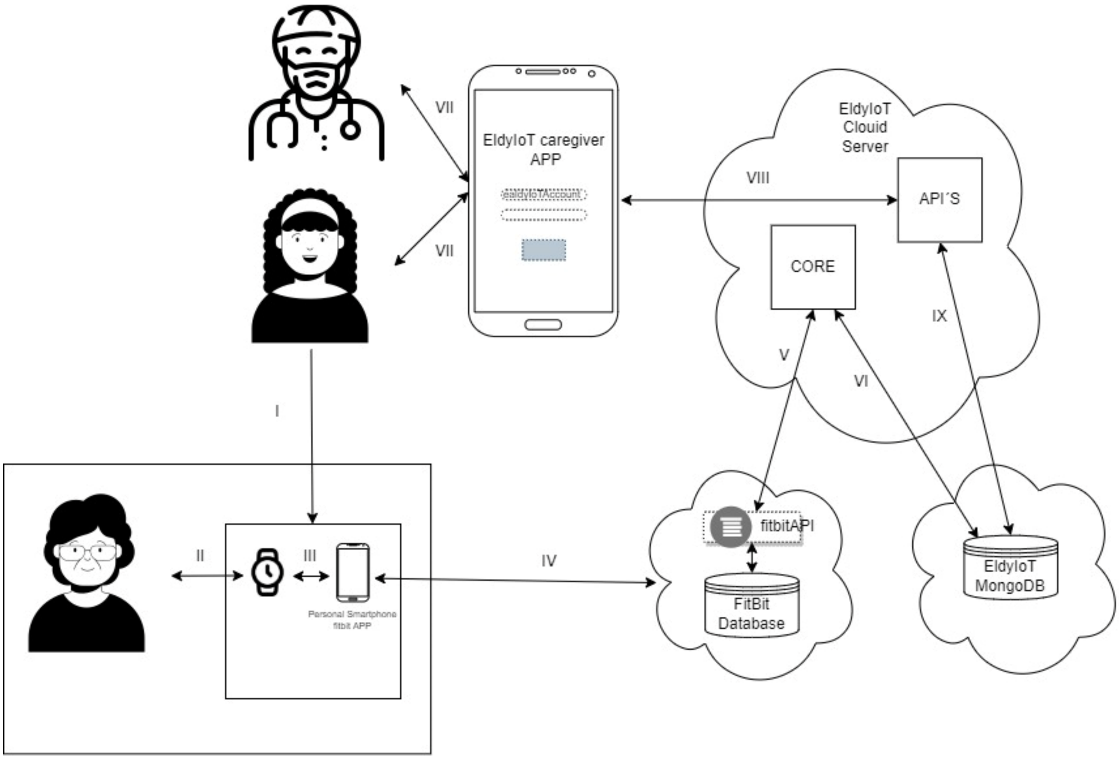


Figure 14– EldyIoT system flow diagram

.

CHAPTER 4

Hardware - Smartwatch Device

4.1 Overview

This chapter aims to describe in more details the device chosen to support the EldyIoT framework. The device is a key part of the framework because it includes built-in sensors that provide data associated with monitoring of the patients during daily life. The type of device chosen to perform patient monitoring is a smartwatch-type device. The choice of this type of device is based on the fact that in the EldyIoT framework we give priority to data collection in a non-intrusive way and without the patient remembering that he or she is being monitored. The device chosen for this work is a smartwatch, from Fitbit, model Fitbit Versa 3. Fitbit has a strong culture of supporting research and has some important tools to support it. For example, data consumption API's and relevant documentation on how we can take advantage of the Fitbit framework. The used device is presented in Figure 15.

Figure 15- Fitbit Versa 3 Kit

4.2 Smartwatch Device - Sensors Layer

The smartwatch Fitbit Versa 3 is equipped with a set of sensors that are used to follow the user during the daily life. Thus, the following sensors are mentioned:

- GPS + GLONASS System - Global Positioning/Navigation System used to calculate distance traveled more accurately.
- SpO2 sensor - Pulse oximeter to measure oxygen levels in the body used to measure oxygen saturation of the blood.
- Optical heart rate sensor - Heart measurement system that is used to collect data regarding cardiac activity.
- 3-axis accelerometer - Acceleration measurement sensor that is used to calculate steps performed by the user during his activity.
- Altimeter - Altitude measurement sensor used to calculate climbed floors
- Ambient light sensor -to perform the measurement of light conditions that may affect the user activity
- Microphones - Sound sensor that may be used for the measurement of the noise in indoor or outdoor condition
- Temperature sensor – that is used to measure the Skin temperature.

The sensors described above, act in such a way as to offer a set of functions capable of improving the daily life of their users. Next, we will go into detail about how each sensor, or set of sensors, is used to gather information about its user.

Regarding Heartrate data, the sensor used for this purpose is the optical heart rate sensor. When our heart pumps blood, our blood vessels contract based on the volume of blood in them. The optical heart rate sensor is located on the back of the smartwatch and fires green LED flashes several times a second. Uses light-sensitive photodiodes to detect these volume changes in the blood vessels above your wrist. The device then calculates how many times your heart beats per minute (bpm). The optical heart rate sensor detects a range of 30-220 bpm. Green LEDs are used because they maximize the signal detected in blood vessels close to the skin surface. The optical heart rate sensor also uses infrared light to determine when the device is on your wrist to improve the accuracy of heart rate data. [49]

The Fitbit Versa recognizes that its user is asleep when he or she has not moved for more than an hour. From the smartwatch we can record the different stages of sleep and accordingly calculate the sleep efficiency. Fitbit estimates the patient's sleep stages by taking advantage of a combination of movement patterns and heart rate. While the patient is sleeping the device, tracks beat-to-beat changes in their heart rate. The changes in heart rate, known as heart rate variability (HRV), fluctuate as the patient transitions between the stages of light sleep, deep sleep, and REM sleep.[50] In addition to sleep stages, we can also obtain sleep efficiency, calculated as follows:

$$\text{Efficiency} = \text{time asleep} / (\text{total time in bed} - \text{time to fall asleep})$$

Sleep efficiency is an indicator that shows the ratio between the time the patient was in bed and the time the patient was asleep. This parameter must be observed with some care since it may not be very accurate for patients who spend a large part of their day bedridden. The Fitbit classifies the sleep efficiency of a patient as [51]:

- Excellent: from 90-100
- Good: from 80-89
- Fair: from 60-79
- Poor: Less than 60

Regarding the calculation of distance traveled, Fitbit devices use two different ways for calculating the distances traveled by their users.[52] The following formula can be used, where the stride length is determined by your height and gender:

$$\text{steps} \times \text{stride length} = \text{distance traveled}$$

Or, when the GPS is active, the information extracted from it is used to calculate the distance traveled. To count the steps taken by the user, the smartwatch makes use of the 3-axis accelerometer.[52]

To calculate the number of floors climbed by the user, the Fitbit smart watch, takes advantage of the altimeter sensor. Given the difference between buildings and height of floors, Fitbit has defined that a new floor, is reached when we observe an altitude gain equivalent to 10 feet or 3 meters. In addition, the device uses barometric pressure differences to confirm the altitude change. Floors climbed on fitness equipment, or going up and down the same stair step, will not be counted. [52]

For calculating the calories expended by users, the Fitbit smartwatch combines a few different inputs. Firstly, the basal metabolic rate (BMR) - the rate at which the patient burns calories at rest to maintain vital body functions (including breathing, blood circulation and heart rate) - and secondly, daily activity data. Heart rate data is also considered, especially for estimating the calories burned during exercise. [52]

Your BMR is based on the physical data you enter in your Fitbit account (height, weight, gender, and age) and accounts for at least half of the calories expended per day.

CHAPTER 5

Software: Server Side, APP and Database

5.1 Overview

In this chapter the software involved in the Fitbit system will be presented. First, we will present the server that supports the process of collecting and processing the data collected by the smartwatch, second, the mobile application, and last, the database.

5.2 Server Side

It is on the server side that the most intelligent part of the framework runs and plays its role. The core module is responsible for information management, data acquisition, processing, and storage in the application database, centralizing the information collected from each patient. The server and the application exchange information using a REST type architecture. The application makes HTTP requests, and the server responds with HTTP responses. Given the need, the API module arises, which is responsible for exposing REST API'S to make the stored data available to be consumed by other services such as the mobile application.

5.2.1 Data acquisition

In the core module, three distinct processes are running that are extremely important for the data acquisition process. These processes run periodically and make use of the API's exposed by Fitbit to acquire data related to each patient. Each of the processes, has the objective of responding to different needs.

The first process runs every 3 minutes and aims to acquire current data to show the patient's health status at that moment.

The second process runs once a day, at 11.59 pm, to collect data regarding calories burned, steps taken, distances traveled, and floors climbed, including data from the last day.

The third process runs once a day at 9am, to collect up-to-date information regarding the user's last 3 months of sleep, including the last night.

5.2.2 Data processing

The core module, besides having an important role in data acquisition, also assumes an important role in the processing of the received data. The HTTP responses sent by the Fitbit API'S and received by the server, arrive in JSON (JavaScript Object Notation) format and often with information not relevant to the framework. In the core module, in each of the previously mentioned processes, the data is processed with the objective of preparing it to be stored and later, consumed by other services such as the mobile application.

5.2.3 Data Storage

The data is stored in a dedicated database with the goal of centralizing all the important information to be made available and consulted by other services.

We chose a Mongo database to support our application and this choice is justified by the need to use a database capable of handling large volumes of data at high speed and with an architecture suitable for expansion. [53] The volume of data stored to run our application and to compare history regarding each patient's health status will lead to increased data in the dedicated database. As such, speed, efficiency, and ease of query was another major factor in the decision. [54]

The documents needed to support the application were thought out in advance and created independently in the code running in the core module.

Figure 16 present all the documents required for the proper functioning of the framework.

```
//LastSleep Document
{
  date: {type: String},
  timeInBed: {type: Number},
  minutesrestless: {type: Number},
  minutesAsleep: {type: Number},
  minutesAwake: {type: Number},
  minutesToFallAsleep: {type: Number},
  token: { type: String }
}
```

```
//User Document
{
  name: { type: String },
  email: { type: String },
  password: { type: String },
  token: { type: String },
  prefixtoken: { type: String },
  mediumtoken: { type: String },
  suffixestoken: { type: String },
}
```

```
//Calories Document
{
  dateTime: { type: String },
  value: { type: Number },
  token: { type: String }
}
```

```
//Device Docuument
{
  id: { type: String },
  value: { type: String },
  token: { type: String }
}
```

```
//Distance Document
{
dateTime: { type: String },
value: { type: Number },
token: { type: String }
}
```

```
//Floor Document
{
  dateTime: { type: String },
  value: { type: Number },
  token: { type: String }
}
```

```
//HeartRate Document
{
  dateTime: {type: String},
  value: {type: Number},
  token: { type: String }
}
```

```
//SleepRate Document
{
  dateTime: {type: String},
  value: {type: Number},
  token: { type: String }
}
```

```
//Step Document
{
  dateTime: {type: String},
  value: {type: Number},
  token: { type: String }
}
```

```
//GenericObject Document
{
id: {type: String},
value: {type: Number},
uni: {type: String},
token: {type: String}
}
```

Figure 16 - EldyIoT database documents

As we can see in Figure 16, each document is accompanied by a token. This token is generated internally by Fitbit and works as a unique identifier for each device. In the EldyIoT framework each patient has a device and consequently each token also identifies a patient. In Figure 17, we present the association explained above.

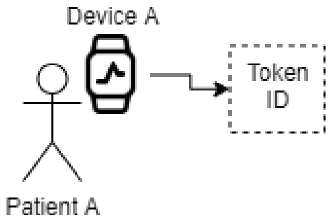


Figure 17– Association between patient - device - token

The token works as an ID that on the one hand ensures that in the database, each patient/device information is identified differently, but on the other hand, ensures that each caregiver only has access to the information of the patient/device in their care. Figure 18 shows a diagram with the summarized idea.

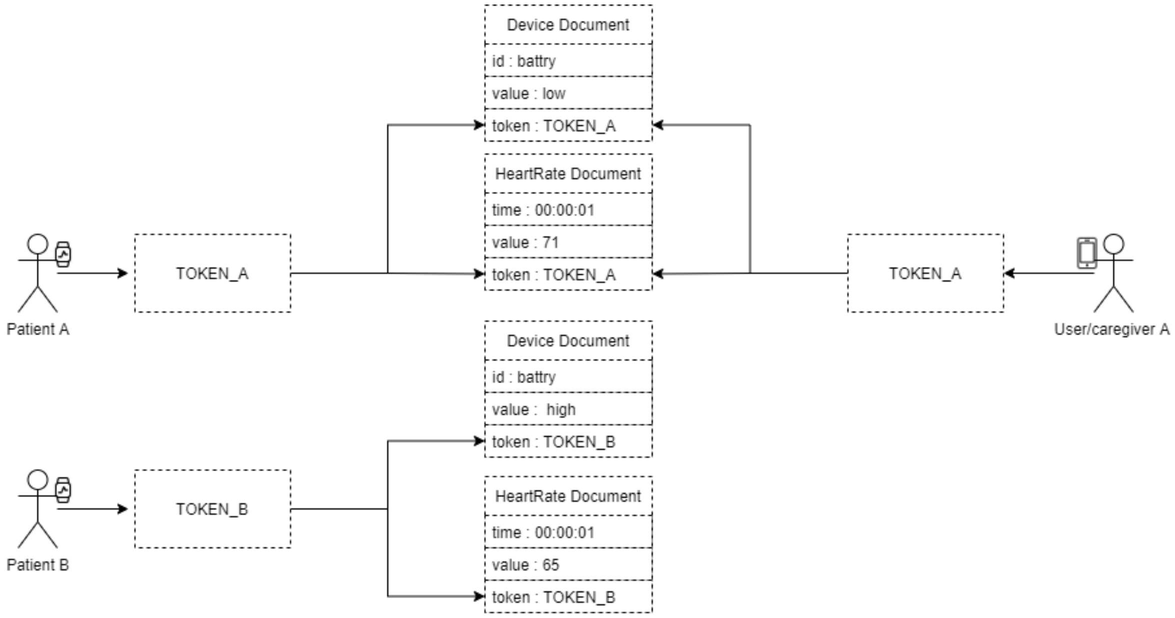
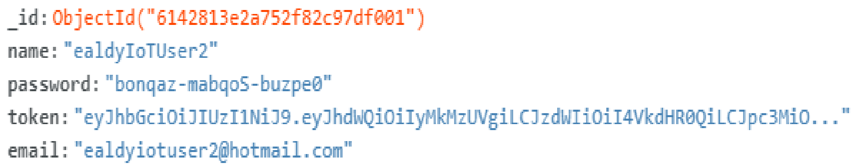


Figure 18– EldyIoT secure token system

The user document is the caregiver document. This document stores information about each user account. As we can see in Figure 19, is saved Information such as name, email, password, and the token of the device of the patient in your care. The access token makes the link between the caregiver and the patient to be monitored.



```
_id: ObjectId("6142813e2a752f82c97df001")
name: "ealdyIoTUser2"
password: "bonqaz-mabqoS-buzpe0"
token: "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMkMzUVgiLCJzdWIiOiI4VkdHR0QiLCJpc3MiO..."
email: "ealdyiotuser2@hotmail.com"
```

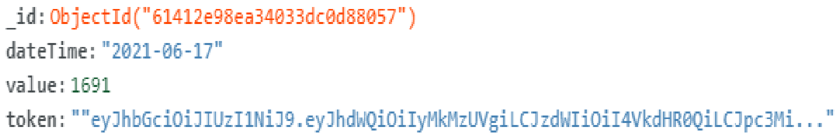
Figure 19– Inserted User Document example

On figure 20 we present a device document that stores information such as battery status, battery percentage, device type and device model. Each of these parameters is accompanied by the token that associate a device and an account as we saw earlier.

```
_id: ObjectId("614de82cea9d806e5ccbaffa")
id: "battery"
value: "Low"
token: "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMkMzUVgiLCJzdWIiOiI4VkdHR0QiLCJpc3MiO..."
```

Figure 20– Inserted Device Document example

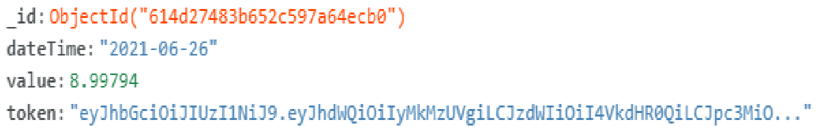
The calorie document stores the information regarding the calories expended by a patient in each day and identified by the token. In Figure 21 is presented the Calories Document



```
_id: ObjectId("61412e98ea34033dc0d88057")
dateTime: "2021-06-17"
value: 1691
token: ""eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMkMzUVgiLCJzdWIiOiI4VkdHR0QiLCJpc3Mi..."
```

Figure 21– Inserted calories document example

Important parameter that is part of the group of acquired parameters is motor activity expressed on distance document which structure is presented in Figure 22.

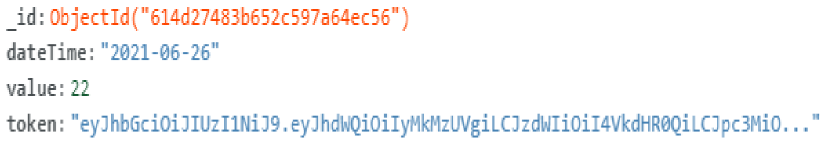


```
_id: ObjectId("614d27483b652c597a64ecb0")
dateTime: "2021-06-26"
value: 8.99794
token: "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMkMzUVgiLCJzdWIiOiI4VkdHR0QiLCJpc3MiO..."
```

Figure 22– Inserted distance document example

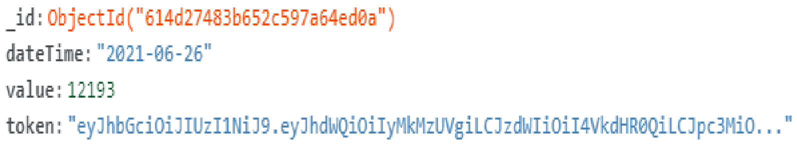
The distance document stores the information regarding the distance traveled by a patient on a given day.

The other two ways to measure the patient's motor activity are floors climbed and steps taken. In figures 23 and 24, we present the documents of floors and steps respectively.



```
_id: ObjectId("614d27483b652c597a64ec56")
dateTime: "2021-06-26"
value: 22
token: "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMkMzUVgiLCJzdWIiOiI4VkdHR0QiLCJpc3MiO..."
```

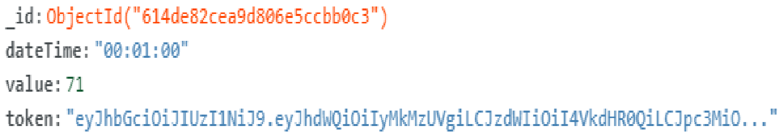
Figure 23– Inserted floors document example



```
_id: ObjectId("614d27483b652c597a64ed0a")
dateTime: "2021-06-26"
value: 12193
token: "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMkMzUVgiLCJzdWIiOiI4VkdHR0QiLCJpc3MiO..."
```

Figure 24– Steps Document example

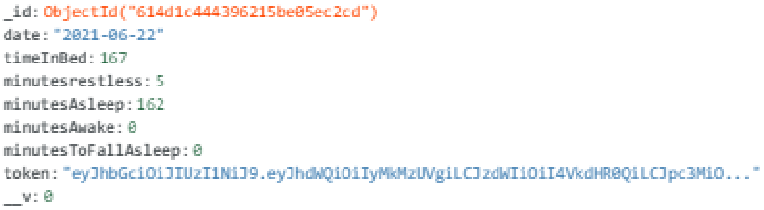
In figure 25 we present the heartbeat document. This document stores the information regarding the heartbeats by a patient on a given day, minute and second.



```
_id: ObjectId("614de82cea9d806e5ccbb0c3")
dateTime: "00:01:00"
value: 71
token: "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMkMzUVgiLCJzdWIiOiI4VkdHR0QiLCJpc3MiO..."
```

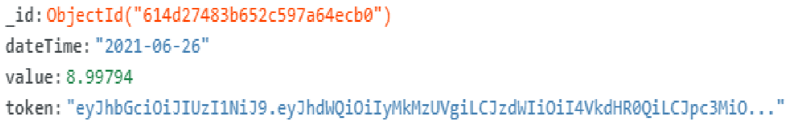
Figure 25– Inserted heart rate document example

Next, in Figure 26 and 27 we present the documents regarding the measurement of the patients' sleep quality. Two documents will be presented, each with a different purpose. The last sleep document stores the information regarding how the last night of sleep was used by a patient and the sleep document, stores information regarding the sleep efficiency of a patient (identified by the token) on a given day.



```
_id: ObjectId("614d1c444396215be05ec2cd")
date: "2021-06-22"
timeInBed: 167
minutesrestless: 5
minutesAsleep: 162
minutesAwake: 0
minutesToFallAsleep: 0
token: "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMkMzUVgiLCJzdWIiOiI4VkdHR0QiLCJpc3MiO..."
__v: 0
```

Figure 26– Inserted last sleep document example



```
_id: ObjectId("614d27483b652c597a64ecb0")
dateTime: "2021-06-26"
value: 8.99794
token: "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMkMzUVgiLCJzdWIiOiI4VkdHR0QiLCJpc3MiO..."
```

Figure 27– Inserted sleep efficiency document example

Regarding the last sleep document, the smartwatch has mechanisms that can understand how we use the time we spend in bed during the night. As such, it stores information such as the time until we fall asleep, the time in deep sleep, the time awake, and the time in light sleep.

As a complement, Sleep efficiency is a parameter used by Fitbit, which aims to show how the night is used to rest and recover energy. The efficiency value comes from the formula:

time asleep / (total time in bed - time to fall asleep)

5.2.4 Communication – WEB API

In this section we will address one of the main characteristics of a smart watch based IoT system. Its ability to communicate with external systems when it comes to receive and provide data collected by smart watch. First, we will address how the communication between server and application is done in our framework and second, we will present the contracts used in our API´s in more detail.

Communications methods:

For the implemented framework, the use of the web API is a mandatory requirement for the communication between the server and the mobile application (Node.js , React Native). An API is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange. [55]

It is used to query information stored in the database, to register users and to start processes. This API is mounted on the server and whenever it is necessary to query information stored in the EldyIoT database, register a user, or start a process, the way to do it is via HTTP (Hypertext Transfer Protocol) invocation. The HTTP architecture was built to allow communication between clients and servers and to function as a request-response type protocol between client and server (Figure 28), for example, a client, submits an HTTP request to the server, and the server responds with a response to the client. This response contains information regarding the specific request made.

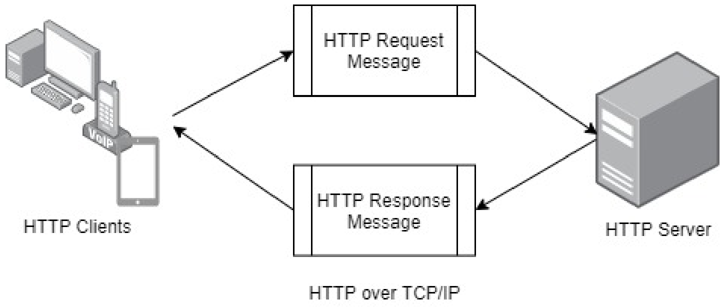


Figure 28– HTTP Flow Diagram

The most used methods of the HTTP communication protocol are:

- GET requests: request the data from a specific resource.
- POST requests: submit the data to be processed in a specific resource.
- PUT requests: allow you to change the data for a specific resource.
- DELETE requests: allow you to erase the data of a specific resource. [56]

Web API

Next, we present the table that includes all the methods implemented to interact with the WEB API exposed on our server.

GET /startBackend/{email}	This post method was created to start the process of data collection and processing by the server. This method receives as a request parameter, the email of the user who is logging in and internally, knows which access token to use to start the processes required for the operation of the application.
GET /actual/{email}	This get method is designed to return a user's most current information regarding Steps taken, Calories burned, Walks climbed, Heart rate measured, and Distance traveled. It receives as parameter the email of the logged user and internally it will know which patient is in question and consequently the data to return.
GET /device/{email}	This get method is designed to return the most current information regarding the device and its status. Information such as battery status, device id, device version and device type are returned.
GET /steps/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to steps taken by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /steps/3/valores/{email}	This get method was created to return the values relative to the steps taken by the patient in the last 3 months. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /distance/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to distances traveled by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.

GET /distance/3/valores/{email}	This get method was created to return the values relative to the daily distance traveled by the patient in the last 3 months. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /floor/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to floors climbed by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /floor/3/valores/{email}	This get method was created to return the values relative to floors climbed daily by the patient in the last 3 months. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /calories/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to calories burned by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /calories/3/valores/{email}	This get method was created to return the values relative to calories expended daily by the patient in the last 3 months. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /batimentos/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to steps taken by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /batimentos/3/valores/{email}	This get method was created to return the daily heartbeat values of the patient in the last 24h. It receives as parameter the email of the logged user and internally will know which patient is being monitored and therefore the data to return.

GET /sleep/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to the patient's sleep efficiency. It receives as parameter the email of the logged user and internally will know which patient is being monitored and therefore the data to return.
GET /sleep/3/valores/{email}	This get method was created to return the values relative to the patient's daily sleep efficiency in the last 3 months. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /lastsleep/valores/{email}	This get method was created to return information about the quality of sleep of the patient during the last night. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.

Table 2– Web API details

5.3 App Mobile

This section aims to describe and show in more detail the mobile application developed to present the data collected by the smartwatch sensors and further processed by the framework. The mobile application was developed entirely using react native. The choice of this language was because we wanted to develop a cross platform application model that would run on both android and iOS.

First the main set of features of the mobile application will be presented. Next, the screens of the application will be detailed and finally, the flow diagram of the application will be presented.

5.3.1 Application Main Features

The application was developed to provide a smart interface for the user that can be materialized by the caregiver and meet the need to present the data collected by the smartwatch in a quick, simple, and intuitive way. Therefore, among the main features of the application we have:

- Visualization of patient data in a graphical and simplified way
- Visualization of health status patient data - Give the caregiver the ability to analyze health status data with the goal of recognizing abnormal values in parameters such as heartbeat.

- Visualization of mobility patient data - Give the caregiver the ability to analyze physical mobility data as distance traveled, floors climbed, steps taken, and calories burned with the goal of recognizing unhealthy patterns and abnormalities.
- Visualization of sleep quality data - Give the caregiver the ability to analyze sleep quality data with the goal of detecting abnormal sleep disturbances.
- Real time alerts - generation of alerts on the caregiver's cell phone in cases of danger or anomalies.
- Real time smartwatch status monitoring - the caregiver can access the status of the device used by the patient and assist in case of need.

5.3.2 Application Screens

Below is a description of each of the screens in the application, as well as their usefulness for the end result and the flow of the application.

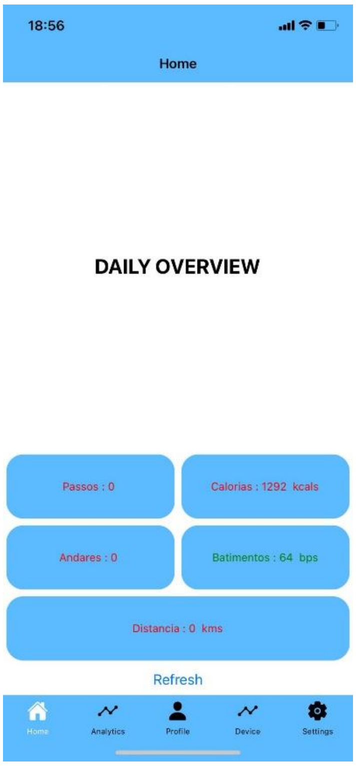
Splash Screen	This screen serves as input to the application. While the application is starting, this screen is displayed with the system icon.
WelcomeScreen	This screen is the screen shown after loading the application and was created to meet the caregiver's initial need to choose whether to login with their existing account or register with a new account.
SignInScreen	This screen was created to satisfy the need to login the caregiver. Login is via email and password, features a button to confirm the action and a link to the Sign-Up page.
SignUpScreen	This screen was created to satisfy the caregiver's Initiating Registration need. For registration a name, email and password are required. This screen features a button to confirm the action and a link to the Sign-In page.
HomeScreen	This screen is the screen displayed after login and shows an information box relating to patient mobility parameters in real time. Among the monitored parameters are the number of steps taken so far, the number of calories used, floors climbed, steps taken and the last beat value per minute (bpm) measured by the smartwatch. In addition to the frame, it also has a refresh button capable of forcing the page to be refreshed.
SettingsScreen	The Settings screen was developed to give the user some extra features and add some customization to the application. On this screen we can change the light theme (by default) to the dark theme, we can change the navigation mode in the application, report a problem and log out of the application.

DeviceScreen	The device's screen was developed to meet the caregiver's need to have access and visibility to the smartwatch used by the patient. It is extremely important for the correct monitoring of the patient that the smartwatch has a battery so that we do not have monitoring gaps. Therefore, this screen allows the observation of variables such as: smartwatch battery, smartwatch id, version, and type.
ProfileScreen	The profile screen was developed to give visibility to the user's personal data of the logged-in account.
AnalyticsSteps	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the steps taken daily by the patient in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for the number of steps taken for an adult aged 65 and 80 years.
AnalyticsDistances	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the distance traveled daily by the patient in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years.
AnalyticsCalories	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the calories burned by the patient daily in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years.
AnalyticsFloors	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the floors climbed daily by the patient in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years.

AnalyticsHeart	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the heartbeats measured daily for each patient. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years.
AnalyticsSleep	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the patient's sleep quality in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years. In addition to the charts explained above, for sleep monitoring, we also present a pie chart that shows how the patient's last night.

Table 3 -- Screens datails

In Figure 29 is presented the HomeScreen associated with patient's status regarding steps taken, calories burned, floors climbed, distance traveled, and last heart rate measurement as talked about earlier in the table.



Passos : 0

Calorias : 1292 kcals

Andares : 0

Batimentos : 64 bps

Distancia : 0 kms

Figure 29– HomeScreen - Daily patient information screen

The purpose of this screen is to give us a quick and clear perception of the patient's health status and how well the daily goals for active and healthy aging are being met or not. These goals are defined based on values for a healthy elderly person. On the screen, the information is loaded dynamically, and is displayed in different colors depending on whether the stipulated goal is being met. Green, when the goal is reached. In red when the goal has not been reached. The refresh button forces the data to be updated whenever the caregiver wants.

In Figure 30, is presented the DeviceScreen. This screen presents the device status screen with some information about the device. Thus, the battery status is presented, which can be useful in case an informal caregiver does not visit the patient daily and does not have daily contact with the smartwatch, as well as the device id when several Fitbit are monitored at the same time.

DEVICE OVERVIEW

Battery: Medium

BatteryLevel: 35

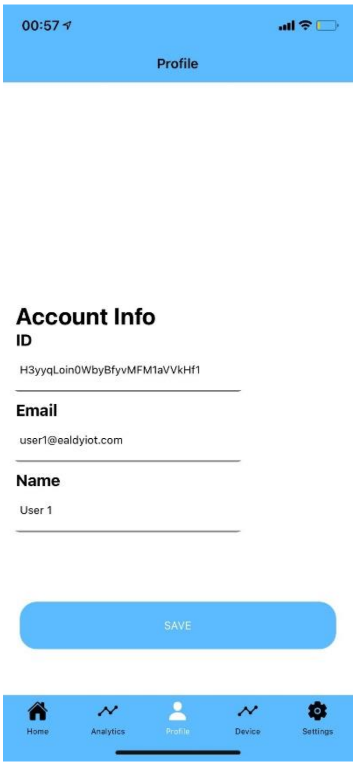
DeviceVersion: Versa 3

DeviceId: 1606614522

DeviceType: TRACKER

Figure 30– DeviceScreen - Device information screen

In Figure 31, the user account information screen is presented on ProfileScreen. In this screen the account id, the registration email and the patient's name are considered.



Account Info

ID

H3yyqLoin0WbyBfyvMFM1aVVkHf1

Email

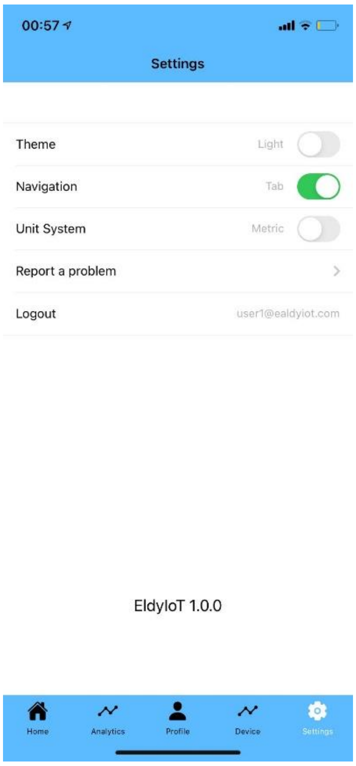
user1@ealdyiot.com

Name

User 1

Figure 31– ProfileScreen - Patient Account Information Screen

In Figure 32, the SettingsScreen of the application is presented. Through this screen elements of theme changing , from light mode to dark mode. We have the possibility to change the navigation, from a bar at the bottom of the screen to a side bar, as well as logout of the account in question.



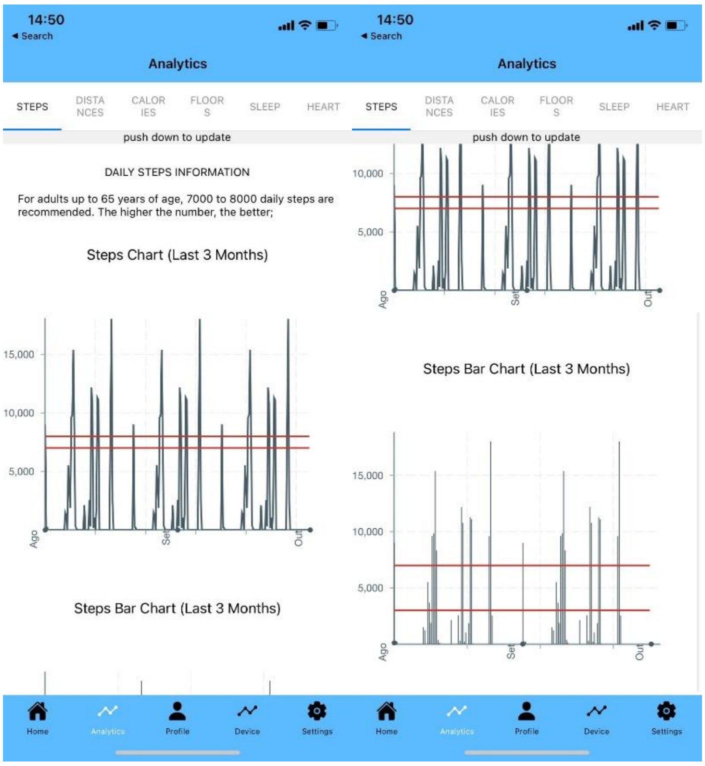
Theme Light
Navigation Tab
Unit System Metric
Report a problem >
Logout user1@ealdyiot.com

EldyIoT 1.0.0

Figure 32– SettingsScreen

Next, in figures 33 to 38, we will show the data analysis screens, built in such a way that a medium/long term analysis of the data collected by the smartwatch is possible. Among them, the number of daily steps over time, distance traveled daily over time, calories burned daily over time, number of floors climbed daily over time, daily sleep quality and heart rate measured over the course of the day. For all the above indicators, we have the possibility to analyze the data collected from the patient in graphical form, facilitating the diagnosis in case any anomaly is detected. All graphs are accompanied by upper and lower limits according to the recommended for adults between 65 and 80 years. The only exception is the sleep quality analysis screen. This screen has an additional graph, which gives an idea of how the time the patient has been in bed has been distributed among the different sleep states. Next, we present the limits assumed to be healthy for each of the parameters analyzed and monitored:

- Distance traveled daily - 5 to 6 km (the higher the better) [59]
- Steps taken daily – 7.000 to 8.000 steps (the higher the better) [59]
- Calories expended daily – 1.800 to 2.800 [60]
- Heartbeat interval - 60 to 100 bpm [57]
- Sleep Efficiency – 80 to 100 [51]



push down to update

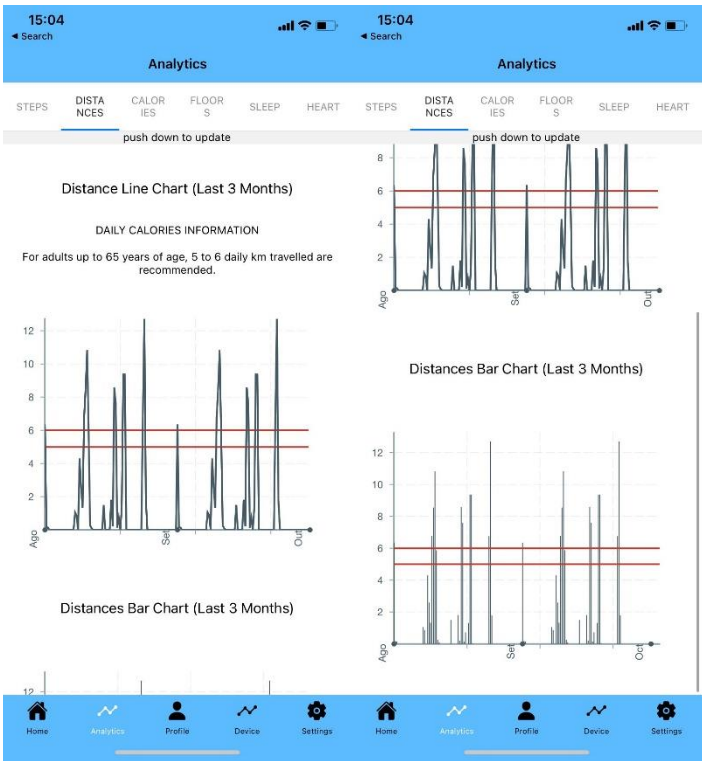
DAILY STEPS INFORMATION

For adults up to 65 years of age, 7000 to 8000 daily steps are recommended. The higher the number, the better;

Steps Chart (Last 3 Months)

Steps Bar Chart (Last 3 Months)

Figure 33– AnalyticsSteps - Step Analysis Screens (Line and Bar Graph)



push down to update

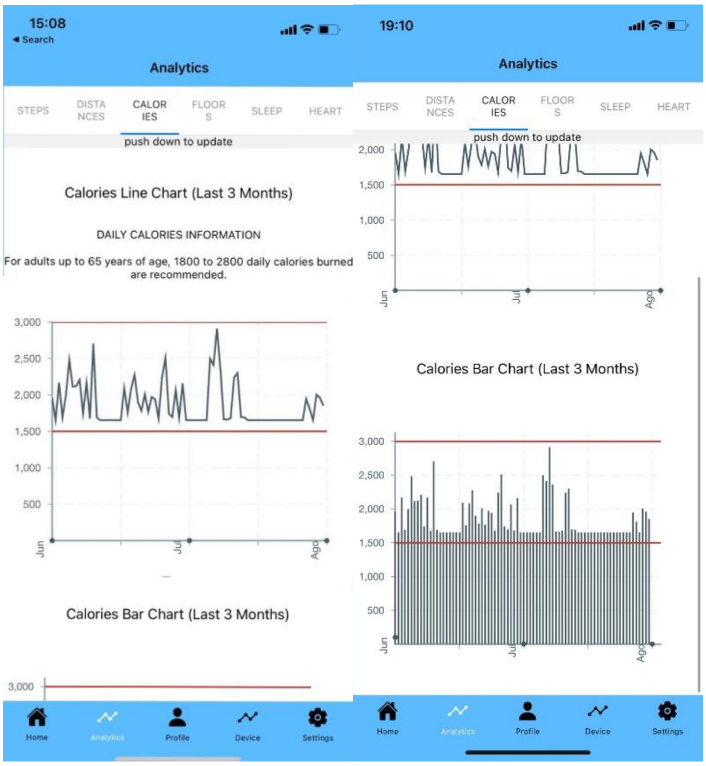
Distance Line Chart (Last 3 Months)

DAILY CALORIES INFORMATION

For adults up to 65 years of age, 5 to 6 daily km travelled are recommended.

Distances Bar Chart (Last 3 Months)

Figure 34– AnalyticsDistance - Distance Analysis Screens (Line and Bar Graph)



DAILY CALORIES INFORMATION

For adults up to 65 years of age, 1800 to 2800 daily calories burned are recommended.

Figure 35– AnalyticsCalories - Calories Analysis Screens (Line and Bar Graph)

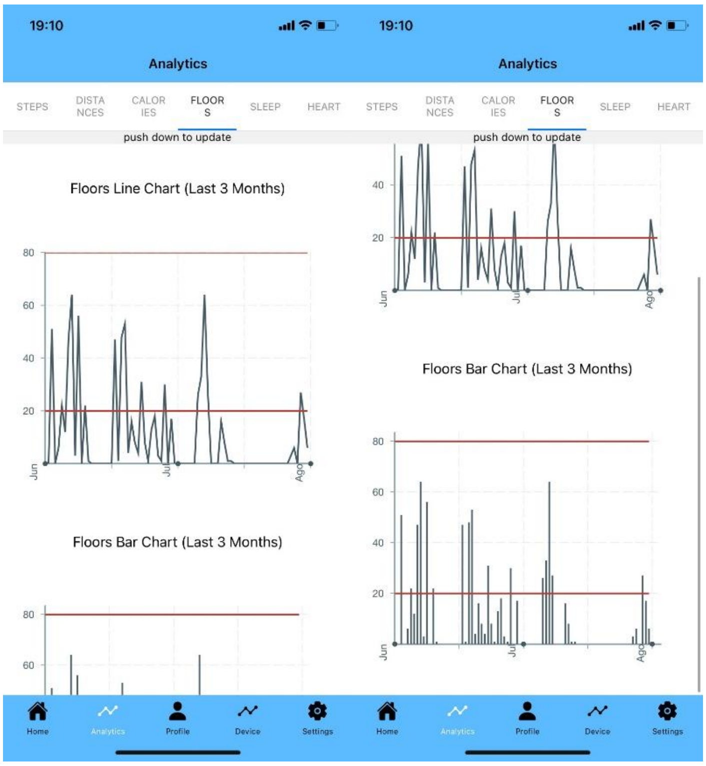
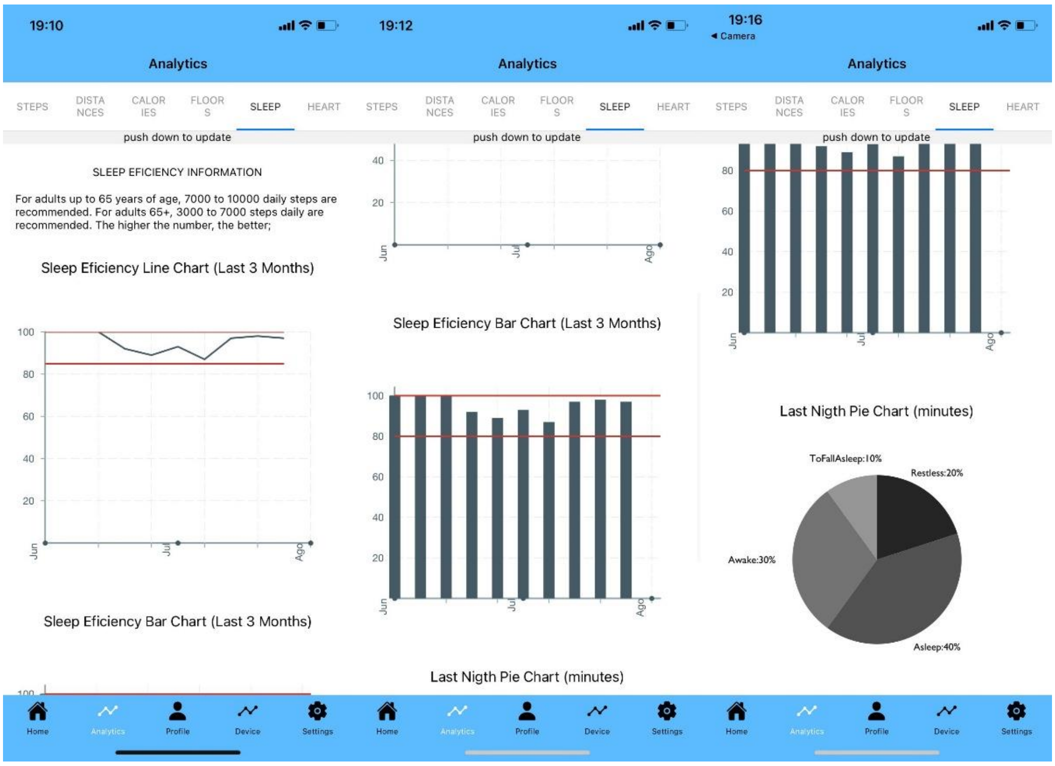


Figure 36– AnalyticsFloors - Floors Analysis Screens (Graph of Lines and Bars)



SLEEP EFICIENCY INFORMATION

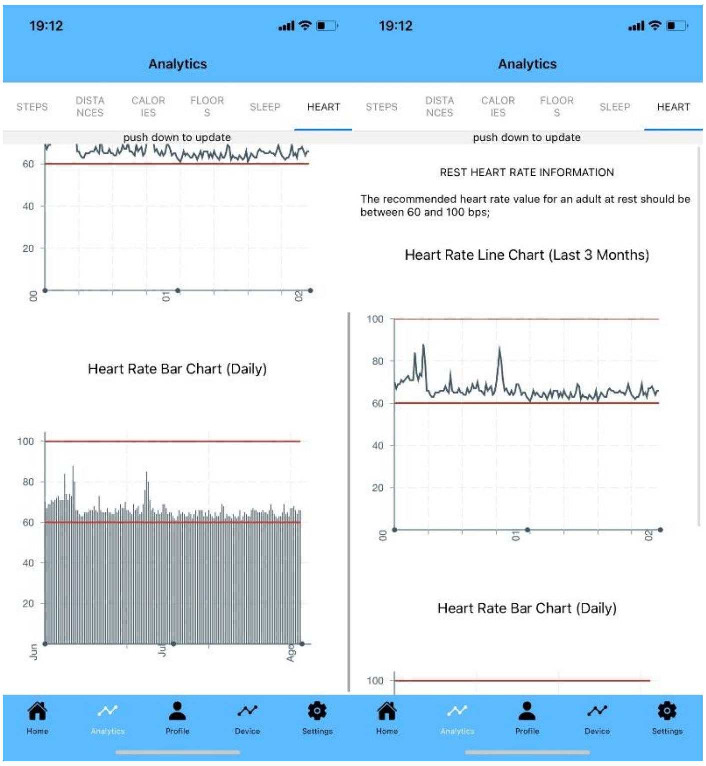
For adults up to 65 years of age, 7000 to 10000 daily steps are recommended. For adults 65+, 3000 to 7000 steps daily are recommended. The higher the number, the better;

Sleep Eficiency Line Chart (Last 3 Months)

Sleep Eficiency Bar Chart (Last 3 Months)

Last Nigth Pie Chart (minutes)

Figure 37– AnalyticsSleep - Sleep Analysis Screens (Line, Bar and Circular Graph)



REST HEART RATE INFORMATION

The recommended heart rate value for an adult at rest should be between 60 and 100 bps;

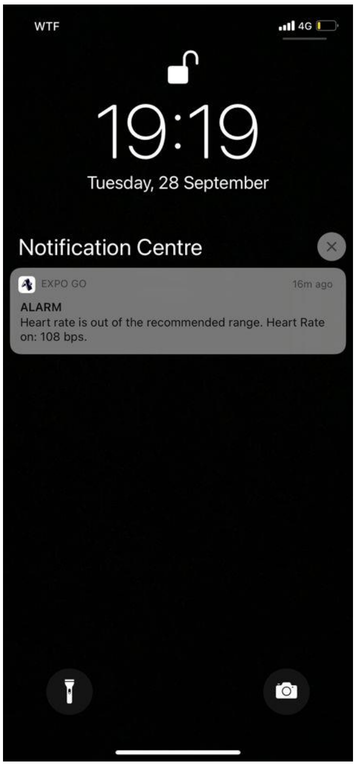
Heart Rate Line Chart (Last 3 Months)

Heart Rate Bar Chart (Daily)

Figure 38– AnalyticsHeart - Heartbeat Analysis Screens (Line and Bar Graph)

Besides the analysis screens capable of detecting and showing anomalies in the patients' health status, we think that in case of emergency it might not be enough. As such, we thought of adding value to the solution and enabling the caregiver to receive alerts whenever any parameter of the health status of the patient in his/her care is at risk values. Therefore, the caregiver doesn't need to constantly look at the analysis graphs to be reassured that the patient in his or her care is doing well. Therefore, the next two screens are designed to meet that need.

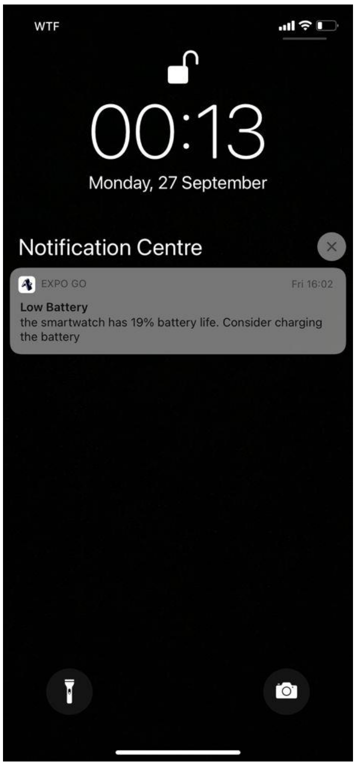
In figure 39, we present a representation of a notification generated by the application on the caregiver's device with an alert of a heart rate value outside the limits stipulated a normal for the patient in question. These alerts are generated when abnormal bpm values are detected, below or above normal values.



ALARM
Heart rate is out of the recommended range. Heart Rate on: 108 bps.

Figure 39– Higher-than-expected heart rate alert

In figure 40, we present the screen that shows the representation of a notification generated by the application on the caregiver's device with an alert regarding the battery status of the device (smartwatch). This type of notification can be very useful in cases where the caregiver is not with the patient every day. Therefore, he or she will receive successive alerts whenever the battery status changes and is below 20%.



Low Battery
the smartwatch has 19% battery life. Consider charging the battery

Figure 40– Smartwatch battery low alert

5.3.3 Application Flow Chart

In figure 41 we present the flow diagram of the application.

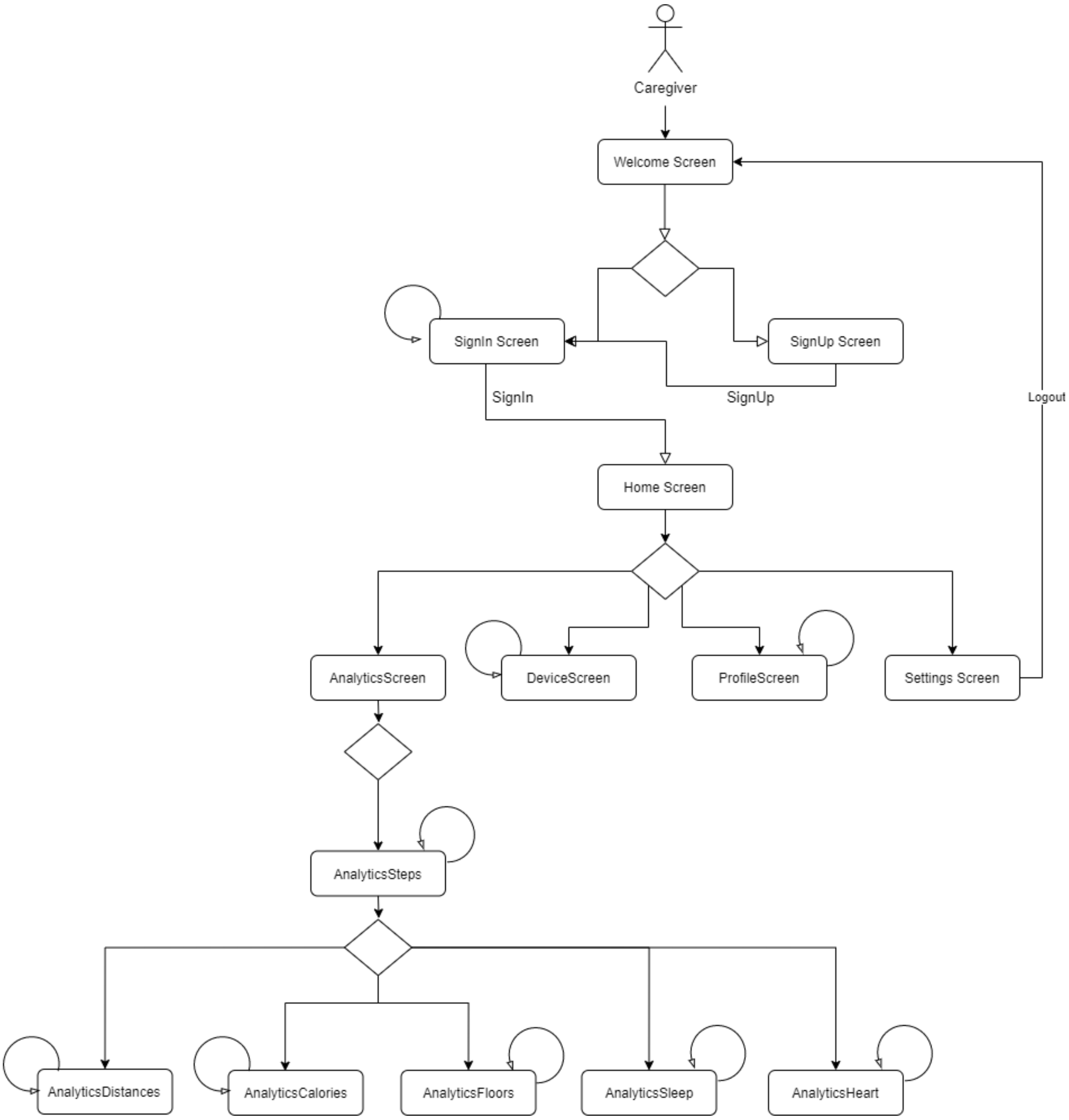


Figure 41- Mobile app flow diagram

CHAPTER 6

Results and Discussions

6.1 Overview

This chapter is intended to describe and show how the framework was tested, as well as the conclusions that were drawn from the tests. Tests were made with real volunteers so that we could get real feedback.

6.2 Tests and Results

The tests done on the EldyIoT system consisted of obtaining data during the same period of time, on different days, and in the course of three different types of activities. In the following table, we present the implemented test cases.

Activity Type	Observation time	Activity	Observed Data
1	30 min	Resting	HeartRate
2	30 min	Sleeping	HeartRate
3	30 min	Walking	HeartRate

Table 4 – EldyIoT test cases

The patients in the study were required to wear the smartwatch during their daily activities and were thus monitored while resting, walking, and sleeping. For patients without walking routines, they were specially asked to do so, if possible, so that the system could be tested and we could draw conclusions about their health status.

In the following table we present the patients who volunteered to test the proposed solution. Patient A and Patient B with relatively close ages of 71 and 75, body mass index within expectations and both female. A third patient was also introduced, male, aged 83 and with a body mass index above the desired one.

	Paciente A	Paciente B	Paciente C	Paciente D
Age	71	75	25	83
BMI (Kg/m2)	22.15	24.14	25.31	28
Gender	F	F	M	M

Table 5- Study patient information

The results obtained for the monitoring of Patient A, Patient B, and Patient C during activity 1 will be presented in figures 42, 43 and 44.

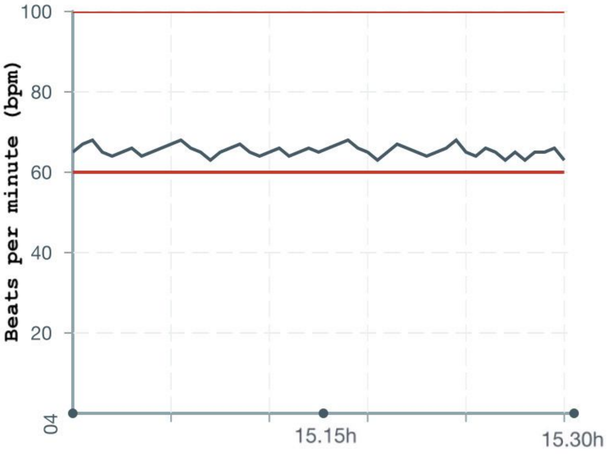


Figure 42– Patient A Heart rate data extracted during Activity 1 (Resting) (bpm/h)

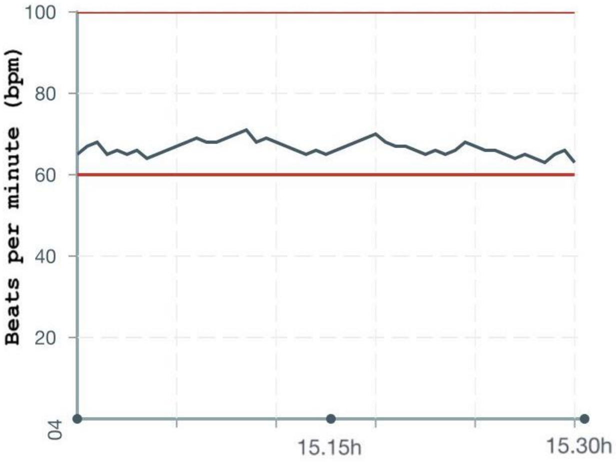


Figure 43– Patient B Heart rate data extracted during Activity 1 (Resting) (bpm/h)

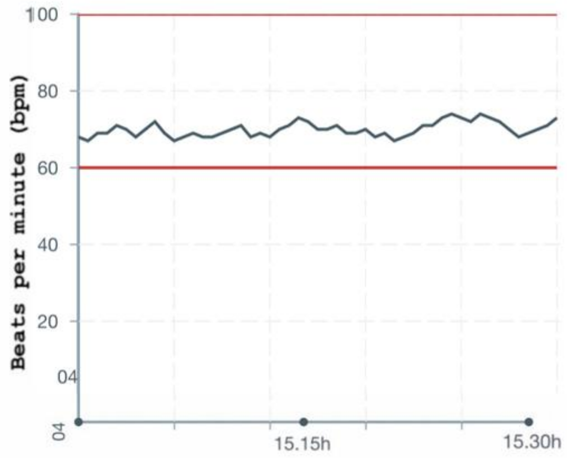


Figure 44– Patient D Heart rate data extracted during Activity 1 (Resting) (bpm/h)

The tests performed on the previous patients, correspond to a data collection interval of 30 min while they were at rest. As we can see, we obtained relatively similar results, without much cause for alarm or anomalies to point out. Patient D presents an average bpm, higher than the previous ones. With increasing age, the heart loses capacity and begins to need to pump more times per minute in order to deliver the oxygen needed for the rest of the body to function. This increase may be natural but given the scenario it is a monitoring point to be considered for the future.

Data were also collected regarding the patients' cardiac behavior during two other activities. Next, in Figures 45, 46, and 47, we will present the results obtained from monitoring Patients A, B, and D respectively, in activity 2.

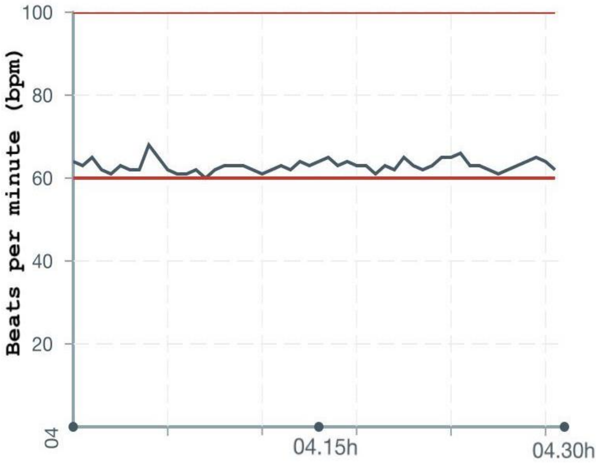


Figure 45– Patient A Heart rate data extracted during Activity 2 (Sleeping) (bpm/h)

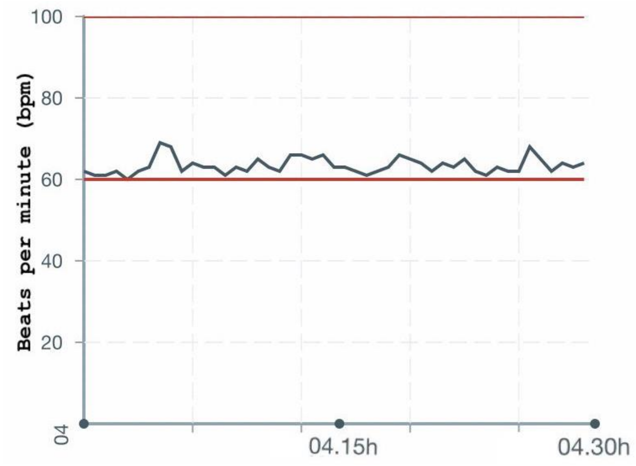


Figure 46– Patient B Heart rate data extracted during Activity 2 (Sleeping) (bpm/h)

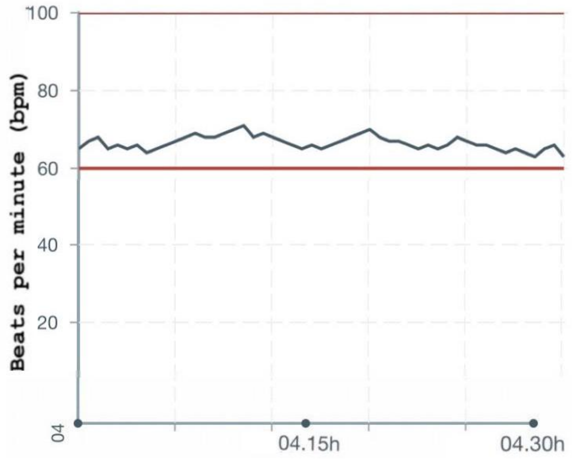


Figure 47– Patient D Heart rate data extracted during Activity 2 (Sleeping) (bpm/h)

During rest, the heart tends to reduce its activity since the body does not need as much oxygen to perform its function. The central nervous system calms down and body pressure tends to decrease[57]. The results obtained from monitoring the patients, mirror this idea.

During activity 3, the volunteer patients were asked to go for a walk so that it was possible to monitor their cardiac status in physical activity. Figures 48, 49 and 50 show the data collected for each of the patients.

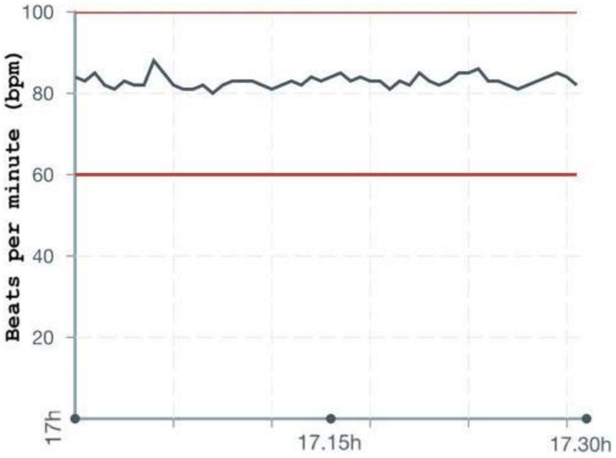


Figure 48– Patient A Heart rate data extracted during Activity 3 (Walking) (bpm/h)

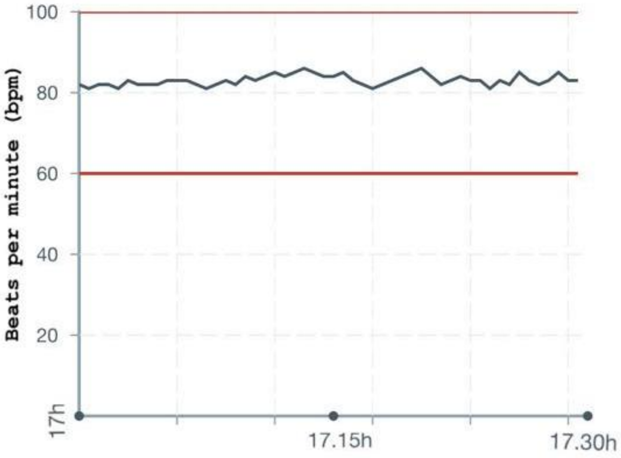


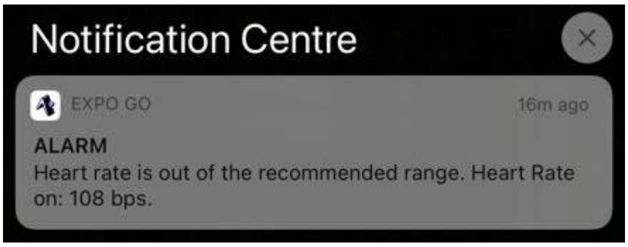
Figure 49– Patient B Heart rate data extracted during Activity 3 (Walking) (bpm/h)

Figure 50– Patient D Heart rate data extracted during Activity 3 (Walking) (bpm/h)

By observing the previous graphs, we were able to observe an increase in overall heart rate, but without much cause for alarm. Contrary to what was observed in the previous graphs, we did not observe an increase in overall heart rate for patient D.

According to the Heart Attack and Stroke Symptoms, [57] the recommended beats per minute values for an adult at rest range from 60 to 100. For moderate physical activity for adults over the age of 70, the range is 75-128 bpm [58]. Thus, the results obtained from the experiments fit the pattern. We were able to observe the variations in the heart rate of each patient according to the activity being performed, and to monitor and analyze that the healthy limits are not exceeded. During the experiments, a history was created for observation in the database and no alerts were generated.

Throughout the experimentation a fourth patient was introduced, so that we could test the application's alert system. As such, the system was used during sports practice with the objective of generating alerts. In Figure 51, we have the result of this experiment.



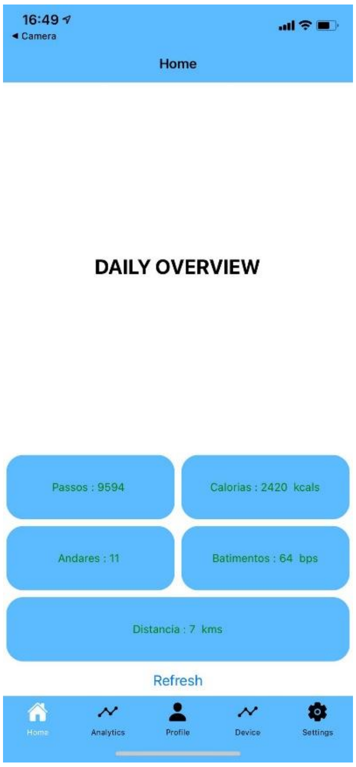
ALARM
Heart rate is out of the recommended range. Heart Rate on: 108 bps.

Figure 51– Alarm generated during the extraction of data from the young adult during physical activity -Patient D.

In addition to information regarding the heart rate of each individual, information regarding the state of mobility during the observation period was also extracted and stored. As such, history was created regarding the number of steps taken, floors climbed, calories burned, distance covered and sleep efficiency.

At the end of the experiments with each of the older volunteers, information regarding their daily mobility status was taken and analyzed.

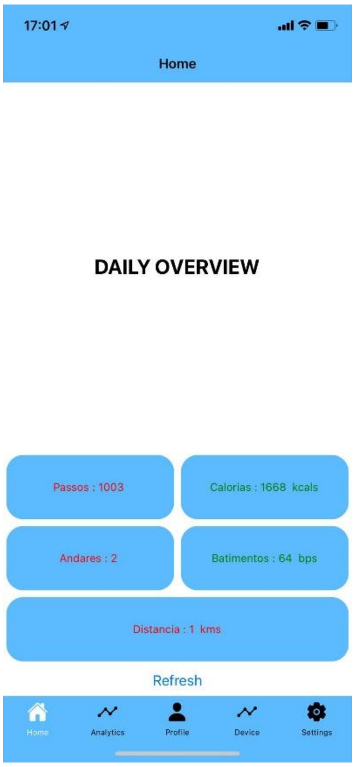
In Figure 52, we can observe that the mobility and health status of patient A is within the expected range in relation to the defined parameters. On the other hand, regarding patient B (Figure 53), although no health alerts have been generated, we can conclude that the lifestyle he led throughout the day is not close to a healthy one. As such, and for a healthier and more active aging, it is important to consider some changes in their routines. The changes in the daily routine of each elderly patient will always depend on their physical and mental lucidness, so the exercise of changing routines will have to be a delicate process and preferably accompanied by a doctor able to advise the way to follow.



DAILY OVERVIEW

Passos : 9594
Calorias : 2420 kcals
Andares : 11
Batimentos : 64 bps
Distancia : 7 kms

Figure 52 – Physical mobility and health status data extracted from a patient A



DAILY OVERVIEW

Passos : 1003
Calorias : 1668 kcals
Andares : 2
Batimentos : 64 bps
Distancia : 1 kms

Figure 53 – Physical mobility and health status data extracted from patient B

In addition to the heart rate measured during the night for each of the patients, sleep efficiency was also monitored for both cases. In Figure 52 and 53 we can see the results of this monitorization. As such we can see that as is normal, with increasing age, the quality of sleep is also affected. We went from a sleep efficiency of 97% for the young adult to an efficiency of 75 for the elderly. Looking at the pie charts, we can also see that the time in bed was not used equally for both patients. The young adult spent more time sleeping. This fact increases sleep efficiency. On the other hand, the older person took longer to fall asleep and had more sleep interruptions during the night. With this, he spent more time awake during the night.

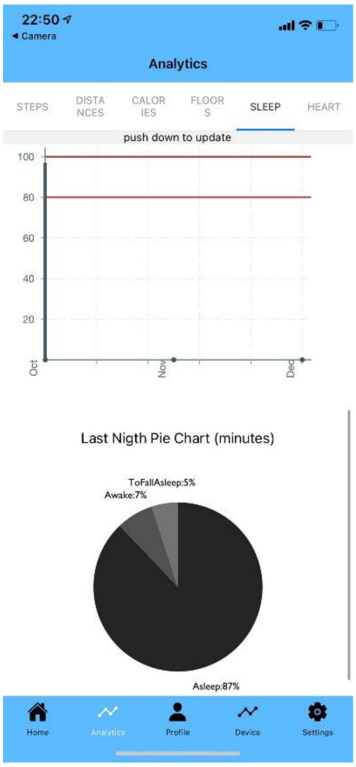


Figure 54 – Last night sleep status extracted from patient D.

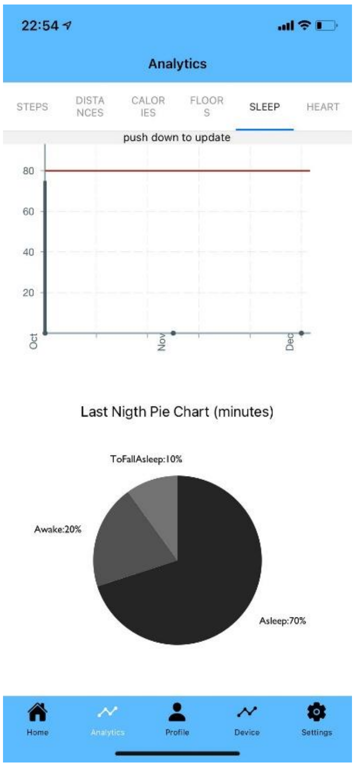


Figure 55 – Last night sleep status extracted from patient B

CHAPTER 7

Conclusion and Future Works

7.1 Conclusion

This system was developed to provide a mobile, non-invasive, and effective option for monitoring patients' mobility parameters and health status. One of the main goals was on the one hand, to increase caregivers' confidence that their patients were doing well, and on the other hand, to decrease patients' fear that something was not right with their health status without anyone monitoring it. Finally, is the idea of meeting the need to monitor important patient health parameters so that abnormalities can be detected as soon as possible. The collection of biometric data from each patient reduces the need for visits to the hospital to confirm the patient's health status since the patient can be monitored from a distance. Choosing Fitbit as a partner for the development of this solution turned out to be a good bet since, given their nature of support for research, they already had some infrastructure to support external solutions such as APIs for data consumption. Despite the limitations imposed by the system such as not being able to access the watch's sensors directly, or not being able to consume API´s data for more than 150 times per hour, I conclude that with this IoT-based mobility and health monitoring system, the entire process of data collection, processing and observation is automated, as well as the constant monitoring, increases the confidence of caregivers and patient, giving more peace of mind to their daily lives. As a caregiver I can have access to the health status parameters of the patient being observed and as a patient, I can be relaxed about the fact that I am being monitored in almost real time. The history of the data collected from the patient is stored and presented in a way that facilitates the analysis process, allowing a faster diagnosis in case of an anomaly.

7.2 Future Works

The system developed is a system with plenty of potential to match existing day-to-day needs as well as to be further developed and improved. Below are some suggestions for improvement:

-Implement multi-patient.

-Implement patient alarms

-Increase the variety of collected health status data.

-Allow configuration of parameters for steps taken, distance traveled, heart rate, distance traveled, and calories burned recommended by caregivers and doctors.

-Application of machine learning to detect anomaly patterns in heartbeat data.

References

[1] ONU, “Envelhecimento - Nações Unidas - ONU Portugal.” https://unric.org/pt/envelhecimento/ (accessed Aug. 03, 2021).

[2] Fundo de População das Nações Unidas, “Envelhecimento no Século XXI.” [Online]. Available: https://www.unfpa.org/sites/default/files/pub-pdf/Portuguese-Exec-Summary_0.pdf.

[3] D. M. Chan, “Relatório mundial de envelhecimento e saúde,” *Organ. Mund. da saúde*, 15AD, Accessed: Aug. 05, 2021. [Online]. Available: www.who.int.

[4] “5 Big Bets for Enterprise IoT in 2020 - InfoVision.” https://www.infovision.com/5-big-bets-for-enterprise-iot-in-2020/ (accessed Aug. 06, 2021).

[5] “Unlocking the Potential of the Internet of Things | McKinsey.” https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world# (accessed Aug. 06, 2021).

[6] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research.” Accessed: Jan. 10, 2021. [Online]. Available: http://www.tuunanen.fi.

[7] “Design Science Research,” 2020.

[8] Department of Economic and Social Affairs, “World Population Ageing 2017 Highlights,” *United Nations*, p. 41, 2017, [Online]. Available: https://www.un.org/en/development/desa/population/publications/pdf/ageing/WPA2017_Highlights.pdf.

[9] S.-C. C. J. Y. W.-T. H. Shu-Ping FENG, “SYSTEM DESIGN AND USABILITY OF HOMECARE SMARTWATCHES,” *J. Sci. Des.*, vol. 1, no. 2, 2018, doi: 10.11247/jsd.1.2_2_11.

[10] “What Is the Internet of Things (IoT)? | Oracle Portugal.” https://www.oracle.com/pt/internet-of-things/what-is-iot/ (accessed Sep. 29, 2021).

[11] “‘An Introduction to the Internet of Things (IoT)’ Part 1. of ‘The IoT Series,’” 2013. Accessed: Dec. 23, 2020. [Online]. Available: https://www.abiresearch.com/press/over-5-billion-wireless-connectivity-chips-will-sh.

[12] “Evolution of IoT begs for new thinking, new value systems.” https://internetofthingsagenda.techtarget.com/opinion/Evolution-of-IoT-begs-for-new-thinking-new-value-systems (accessed Sep. 29, 2021).

[13] “Internet of Things 101 | SmartBear Blog.” https://smartbear.com/blog/internet-of-things-101/ (accessed Oct. 02, 2021).

[14] “VR Technology: Immersive Training for Students and Surgeons.”

https://healthtechmagazine.net/article/2020/11/how-surgeons-use-vr-technology-train-and-adapt (accessed Nov. 01, 2021).

[15] “Surgical Robots for Minimally Invasive Procedures. | maxon group.” https://www.maxongroup.com/maxon/view/application/Surgical-Robots-for-Minimally-Invasive-Procedures (accessed Nov. 01, 2021).

[16] “How is an IoT-based Ambience Monitoring System beneficial for Healthcare Industry?” https://www.biz4intellia.com/blog/how-iot-based-ambience-monitoring-system-is-beneficial-for-healthcare-industry/ (accessed Nov. 01, 2021).

[17] “Now is the time for healthcare providers to get into IoT | Arthur D Little.” https://www.adlittle.com/en/insights/report/now-time-healthcare-providers-get-iot (accessed Oct. 02, 2021).

[18] J. Pedro, N. Rei, and I. B. Filho, “Study, Selection and Evaluation of an IoT Platform for Data Collection and Analysis for medical sensors,” *Fed. Univ. Rio Gd. do Norte*, no. July, 2018.

[19] “Wearable technology - Wikipedia.” https://en.wikipedia.org/wiki/Wearable_technology (accessed Oct. 02, 2021).

[20] H. A. J. Joel J. P. C. Rodrigues, Senior Member, IEEE, Dante B. R. Segundo and and V. H. C. de A. Murilo H. Sabino, Rafael M. Prince, Jalal Al-Muhtadi, “Enabling Technologies for the Internet of Health Things,” p. 9, 2018, [Online]. Available: https://www.researchgate.net/publication/322261039_Enabling_Technologies_for_the_Internet_of_Health_Things.

[21] “Air Quality Monitoring | LouisvilleKY.gov.” https://louisvilleky.gov/government/air-pollution-control-district/air-quality-monitoring (accessed Nov. 01, 2021).

[22] “Hi-tech iTBra a breakthrough for Asian women at high risk of breast cancer | South China Morning Post.” https://www.scmp.com/lifestyle/health-wellness/article/2180728/hi-tech-itbra-breakthrough-asian-women-high-risk-breast (accessed Nov. 01, 2021).

[23] “Wearable Technology Definition.” https://www.investopedia.com/terms/w/wearable-technology.asp (accessed Oct. 02, 2021).

[24] “Smart Tattoo Technology: The Future Within Skin • Tattoodo.” https://www.tattoodo.com/articles/smart-tattoo-technology-the-future-within-skin-14567 (accessed Nov. 01, 2021).

[25] “O que É um Banco de Dados Relacional | Oracle? | Oracle Brasil.” https://www.oracle.com/br/database/what-is-a-relational-database/ (accessed Nov. 02, 2021).

[26] “Dados não relacionais e NoSQL - Azure Architecture Center | Microsoft Docs.” https://docs.microsoft.com/pt-pt/azure/architecture/data-guide/big-data/non-relational-data (accessed Nov. 02, 2021).

[27] “Vantagens de um Banco de Dados NoSQL, MongoDB.” https://blog.4linux.com.br/vantagens-

de-um-banco-de-dados-nosql-mongodb/ (accessed Nov. 02, 2021).

[28] “SQLite vs MySQL – Qual a Diferença e Qual Usar.” https://www.hostinger.com.br/tutoriais/sqlite-vs-mysql#Pros-e-Contras-SQLite-vs-MySQL (accessed Nov. 02, 2021).

[29] “MariaDB ou MySQL: saiba qual tecnologia de banco de dados escolher.” https://rockcontent.com/br/blog/mariadb/ (accessed Nov. 02, 2021).

[30] “Top 10 Backend Technologies | Low-code backend to build modern apps.” https://blog.back4app.com/backend-technologies/#1_JavaScript (accessed Nov. 02, 2021).

[31] “Node.js Backend Development: Features, Benefits, Prices - Existek Blog.” https://existek.com/blog/node-js-backend-development/ (accessed Nov. 02, 2021).

[32] “Node.js vs PHP For Backend Development: Which One Should You Pick?” https://www.ideamotive.co/blog/nodejs-vs-php-for-web-development (accessed Nov. 02, 2021).

[33] “Node.js vs Python — What to Choose in 2021 — TechMagic.” https://www.techmagic.co/blog/node-js-vs-python-what-to-choose/ (accessed Nov. 02, 2021).

[34] “Node.js vs Ruby on Rails: Which Is the Best for Web Development?” https://www.techmagic.co/blog/nodejs-vs-ruby-on-rails/ (accessed Nov. 02, 2021).

[35] “Why Node.js is Great for Backend Development? | Engineering Education (EngEd) Program | Section.” https://www.section.io/engineering-education/why-nodejs-is-great-for-backend-development/#main-competitors-of-nodejs-in-backend-development (accessed Nov. 02, 2021).

[36] “Top 10 Best Cross-Platform App Development Frameworks in 2021.” https://www.codenameone.com/blog/top-10-best-cross-platform-app-development-frameworks-in-2021.html (accessed Nov. 02, 2021).

[37] “11 Popular Cross-Platform Tools for App Development (Updated Version 2021) | Hacker Noon.” https://hackernoon.com/9-popular-cross-platform-tools-for-app-development-in-2019-53765004761b (accessed Nov. 02, 2021).

[38] “Cross-Platform Mobile Development: Five Best Frameworks | SaM Solutions.” https://www.sam-solutions.com/blog/cross-platform-mobile-development/ (accessed Nov. 02, 2021).

[39] A. W. A. A. Da Xu, “The Internet of Things—A survey of topics and trends,” 2014.

[40] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, “The internet of things for health care: A comprehensive survey,” *IEEE Access*, vol. 3, pp. 678–708, Jun. 2015, doi: 10.1109/ACCESS.2015.2437951.

[41] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of things: Vision, applications

and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012, doi: 10.1016/J.ADHOC.2012.02.016.

[42] C. Boletsis, S. McCallum, and B. F. Landmark, "The use of smartwatches for health monitoring in home-based dementia care," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9194, pp. 15–26, doi: 10.1007/978-3-319-20913-5_2.

[43] J. R. Thorpe, B. H. Forchhammer, and A. M. Maier, "Development of a sensor-based behavioral monitoring solution to support dementia care," *J. Med. Internet Res.*, vol. 21, no. 6, Jun. 2019, doi: 10.2196/12013.

[44] R. Ramezani *et al.*, "A combination of indoor localization and wearable sensor-based physical activity recognition to assess older patients undergoing subacute rehabilitation: Baseline study results," *J. Med. Internet Res.*, vol. 21, no. 7, Jul. 2019, doi: 10.2196/14090.

[45] V. Kilintzis, C. Maramis, and N. Maglaveras, "Wrist sensors - An application to acquire sensory data from Android Wear® smartwatches for connected health," 2017, doi: 10.1109/BHI.2017.7897221.

[46] R. K. Kodali, G. Swamy, and B. Lakshmi, "An implementation of IoT for healthcare," 2016, doi: 10.1109/RAICS.2015.7488451.

[47] S. L. G. L. J. Divyabharathi, "REMOTE PRESCRIPTION AND I-HOME HEALTHCARE BASED ON IoT," *2017 Int. Conf. Innov. Green Energy Healthc. Technol.*, 2017, [Online]. Available: https://ieeexplore.ieee.org/document/8094069.

[48] B. Magazine, "Smartphones and how they impact our daily lives," Aug. 25, 2019. https://bmmagazine.co.uk/business/smartphones-and-how-they-impact-our-daily-lives/ (accessed Aug. 16, 2021).

[49] "How do I track heart rate with my Fitbit device?" https://help.fitbit.com/articles/en_US/Help_article/1565.htm (accessed Nov. 01, 2021).

[50] "How do I track my sleep with my Fitbit device?" https://help.fitbit.com/articles/en_US/Help_article/1314.htm (accessed Nov. 01, 2021).

[51] "What's sleep score in the Fitbit app?" https://help.fitbit.com/articles/en_US/Help_article/2439.htm (accessed Nov. 01, 2021).

[52] "How does my Fitbit device calculate my daily activity?" https://help.fitbit.com/articles/en_US/Help_article/1141.htm (accessed Nov. 02, 2021).

[53] "Advantages of NoSQL | MongoDB." https://www.mongodb.com/nosql-explained/advantages (accessed Sep. 13, 2021).

[54] "Advantages of MongoDB | MongoDB." https://www.mongodb.com/advantages-of-mongodb (accessed Sep. 13, 2021).

[55] "What is API: Definition, Specifications, Types, Documentation | AltexSoft."

https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/ (accessed Sep. 29, 2021).

[56] “Hypertext Transfer Protocol – Wikipédia, a enciclopédia livre.” https://pt.wikipedia.org/wiki/Hypertext_Transfer_Protocol (accessed Sep. 21, 2021).

[57] “All About Heart Rate (Pulse) | American Heart Association.” https://www.heart.org/en/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure/all-about-heart-rate-pulse (accessed Nov. 02, 2021).

[58] “Target Heart Rates Chart | American Heart Association.” https://www.heart.org/en/healthy-living/fitness/fitness-basics/target-heart-rates (accessed Nov. 02, 2021).

[59] “Recommended Walking Distances for Seniors.” https://www.fgblawfirm.com/blog/2018/june/recommended-walking-distances-for-seniors/ (accessed Nov. 04, 2021).

[60] “Calorie Requirements for Seniors.” https://www.verywellfit.com/calorie-requirements-for-older-people-2223969 (accessed Nov. 04, 2021).

Appendix A – Article

The 9th IEEE International Conference on E-Health and Bioengineering - EHB 2021
Grigore T. Popa University of Medicine and Pharmacy, Web Conference, Romania, November 18-19, 2021

EldyIoT - IoT Monitoring System For Elderly People

David Lourenço [1], Octavian Postolache[1,2],
ISCTE-Instituto Universitário de Lisboa, Lisboa, Portugal,
Instituto de Telecomunicações, Lisboa, Portugalopostolache@lx.it.pt

***Abstract*— The Internet of Things (IoT) is one of the most promising technologies for the near future. IoT has penetrated. into many industries like smart cities, smart homes, agriculture, manufacturing, museums, events, smart cars, and the healthcare industry. There are a huge number of IoT Use Cases in the Healthcare Industry. With this work, we present a solution to decrease the distance between caregivers and patients in a non-intrusive way, allowing information regarding the health status and mobility of patients to be available and accessible anywhere. As such, this thesis presents a framework for physical monitoring of the elderly based on smartwatch devices. As an integral part of the solution, there is a hybrid mobile application and a software structure assembled to support the processes of data collection, processing, storage, and display.**

Keywords—Internet of Things; Health Care Perversive; Wearable Sensors; Health Care; Ambient Assisted Living.

I. INTRODUCTION

The Internet of Things (IoT) describes the network of physical objects-"things"-that are embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet. Healthcare is one of the fastest-growing industries to adopt and utilize the benefits of IoT in all its might. [1] With over 10 billion devices connected today, the number is expected to scale to 22 billion by 2025. [2] The increasing rate of the world's aging population has become an increasingly important thematic. Population aging is a worldwide trend and an increasingly serious issue. Among other factors, the increase in the population over 60 years of age creates growing problems in health care needs and care costs, increasing pressure on social and health care systems. Population aging is being felt in different regions of the globe, and it is estimated that by 2030, 34 nations will have reached the standard for an over-aged society and that by 2050, 21.5% of the world's population will be 60 years of age or older. [3] Most older adults prefer to age in a place where they live in conjunction with technological assistance that maintains and supports their independent, active, healthy, and functional lives rather than going to places where they are not as comfortable. With technological advancement, we have been able to achieve the goal of person-centered care by providing older adults with accessible tools for aging at home, improving their quality of life. The main goals of home care should be personalized care, respect for the value of the person, and the creation of a social environment that supports the psychological needs of the patient. Healthy aging is directly linked to active aging and, as such, it becomes a fact of extreme importance to analyze the evolution regarding physical activity in older patients. On the other hand, abnormal heart rhythms represent one of the most common pathologies among the elderly. In one study of 147 active, healthy elderly subjects, arrhythmias were observed in 93.2%. In another study on 98 subjects of both sexes, aged 60 to 85 years, arrhythmias were found in at least 80% [13]. The concepts of home care and aging at home have been valued and studied in countries around the world; This, however, remains an area in need of evolution and research. [3]

II. STATE OF THE ART

In the first two surveys, two solutions for monitoring dementia patients are presented. Several of these patients receive home care to maintain their independence. Mobility and physical activity are very significant points when it comes to the independence and quality of life of patients and the non-progression of the disease. Increasing age and cognitive impairment are closely linked to reduced mobility. Therefore, wearable devices become great potential allies for monitoring mobility-related parameters in a non-intrusive method. [4] [5]

In addition to minor control regarding paraments such as mobility, the treatment of dementia as well as the treatment of most chronic diseases follows an unwanted approach. Monitoring of patients and reporting of physical and cognitive decline depends on visits by health care operators (physicians, general practitioners, nurses) or the patient's trips to the hospital. Given the non-constant nature of these pathologies, the method of medical monitoring has obvious limitations. Given this scenario, it is becoming increasingly clear that constant monitoring of patients in recovery can be a great asset in their recovery, treatment, or stabilization of the disease. As a consequence, data collection and analysis become a point of extreme importance. Costas Boletsis and Simon McCallum proposed a smart system capable of providing additional and potentially useful information regarding dementia patients by

IEEE 978-1-6654-4000-4/21/$31.00 ©27October2021 IEEE

advantage of a smartwatch to record parameters such as blood oxygen, body temperature, sweat quality, ambient temperature as well as data from an accelerometer. [4] Also taking advantage of wearable devices, smartwatches have taken a prominent role in this research being part of most of the studies analyzed. In the same context but this time not so focused on the aid of diseases in particular but on health care at home, S. Feng and S. Chiou came up with a monitoring and assistance platform consisting of 4 modules that among them distribute functions such as scheduling daily care services, scheduling the supply of medication, notifications, appointments, sending messages, an emergency button and a module responsible for collecting health parameters such as blood pressure, blood sugar, body weight, and blood alcohol. [5] In the same line of thought comes ROAMM [6] positioning itself also as a monitoring and assistance platform with monitoring of parameters like a heartbeat, and GPS data but that in turn enriches the knowledge about the user's health status with programmed requests for classification inputs of parameters like pain, fatigue, and mood. In a slightly different technological vein comes the combination of indoor location monitoring and wearable devices for the rehabilitation of old people. The framework in question combines proximity sensors with smartwatches that, based on signal exchanges, have the intelligence to calculate the user's distance to landmarks scattered around the house. Therefore, it becomes possible to map the indoor displacements and infer their daily activity. [7] Wrist sensors [8] come with a slightly different paradigm and present their greatest contribution by being an application capable of pairing with compatible devices, collecting data that can serve as an object for study for the evolution of intervention algorithms, and possible detection of possible health problems of the monitored patients.

The technological development and the constant evidence that IoT can be a strong ally in improving patient care, create ideas that there is room for improvement of the processes involved in these activities. As such, in [9] a cloud computing-based approach is proposed to integrate all hospital records of a patient and proposes to keep them centralized. The proof of concept of this proposal was done by integrating a temperature sensor with the rest of the system, centralizing the patient's temperature data in a database exposed on a web server. This way, a process that would have to be done by medical personnel, became automatic and available for consultation almost immediately. In [10] a system is proposed that again strengthens the argument that emphasizes the importance of collecting and analyzing patient data. The monitoring and collection of patient health status information take the burden off the health systems, saving time for doctors and nurses who can focus on analyzing the results and making the service more efficient. This system collects heartbeat, blood pressure, and ECG data with the help of sensors. After collection and processing, if any abnormalities are found, alerts are sent to the physician responsible for the patient, and the history of the data is made available on a web page.

III. SYSTEM DESCRIPTION

The EaldyIoT Framework includes a hardware block and a software block. The hardware combines a set of functions capable of collecting data from the patient under study via sensors embedded in smartwatches. As an integral part of the hardware and support for collecting patient data, the Fitbit versa 3 smartwatches were chosen. As an integral part of the solution, we have a backend structure built on a cloud infrastructure capable of supporting the process of processing and storing data that will be made available for consumption by the mobile application. The data used in the framework are data obtained post-prior and periodically from the API provided by Fitbit. This data is stored in a remote database and made available for consumption via the API. As a way of visualizing the data, a mobile application was created that can offer a visualization of the collected data in a simple way to facilitate understanding. It is possible to observe the data on a timeline of up to 3 months to have a medium/long-term perspective. In addition to collecting data, the system also processes it, generating alerts when an abnormal heartbeat value is detected. In addition, low battery notifications are also generated.

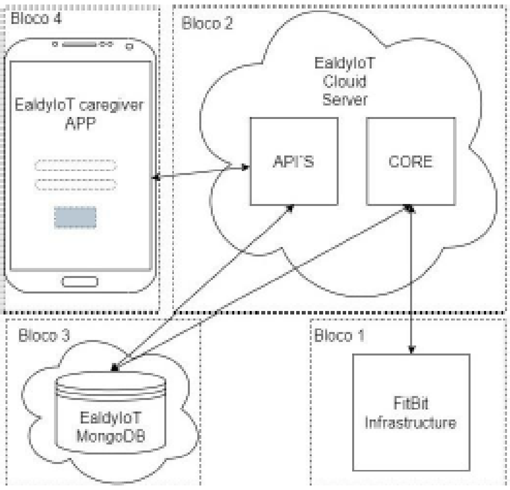


Fig 1 – EaldyIoT Architecture Diagram

A. *Hardware:*

The Fitbit Versa 3 is equipped with some important features and sensors to performance its function, among them:

- 6-day battery life efficiency
- GPS + GLONASS System - Global Positioning System
- SpO2 sensor - Pulse oximeter to measure oxygen levels
- Optical heart rate sensor - Heart measurement system
- 3-axis accelerometer - used to calculate steps taken

- Altimeter - used to calculate floors climbed. One floor up is equivalent to a 3 m height gain
- Ambient light sensor - Ambient light sensor.
- Microphones - Sound sensor.
- Temperature sensor - Skin temperature sensor.

Fig 2 - Fitbit smartwatch

B. *Software:*

The structure of the EaldyIoT framework can be divided into 4 different blocks. Block 1 is responsible for data collection and is part of the structure provided by Fitbit. This block includes the smartwatch that we decided to use, the Fitbit database, and the Fitbit smartphone application. The information flow occurs from the smartwatch to the mobile app via Bluetooth and from the mobile app to the Fitbit database via the internet. This pairing and synchronization can be forced in the application, but if not, it is done periodically every 15 min. Block 2 is the brain of the framework. This is where data is consumed via the Fitbit API, processed, and stored in the framework's database. This block is running on a cloud server and is divided into two modules, the CORE module, and the API module. The CORE module keeps the information in the framework database consistent and up to date with the data collected by the smartwatch. and the API module displays the information for consumption. Block 3 is where the data is stored. Before storing the data, it is processed, and we only extract the important and relevant information for the framework. Block 4 is the data visualization block. This data visualization is done via mobile application and gives caregivers the possibility to access information extracted as simply as possible. It is possible to visualize daily patient data as well as to make long-term analyses with the aid of graphs. The possibility of a mobile application solution arose to make the process of information extraction and monitoring as quick and simple as possible.

IV. RESULTS AND DISCUSSIONS

Several experiments have been done, and here, the results of two of them will be presented. Next, we present and explain part of the results obtained during 3 different activities. Resting, walking and sleeping. For this sample, we only used 30 min of monitoring of each activity.

TABLE I. PATIENT INFORMATION

Learning set	Test set		
	Patient A	*Patient B*	*Patient C*
Age	71	75	25
BMI (Kg/m2)	22.15	24.14	25.31
Gender	F	F	M

Resting:

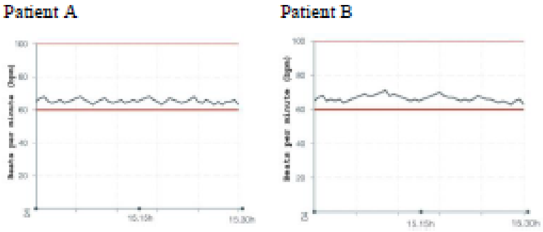


Fig 3 – Heart rate data extracted during rest (bpm/h)

Sleeping:

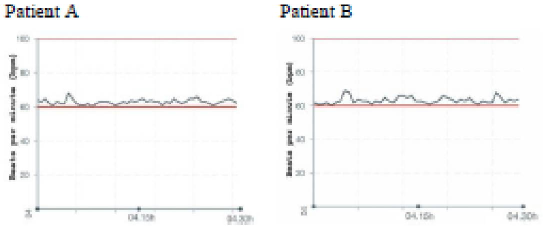


Fig 4 – Heart rate data extracted during sleep (bpm/h)

Wallking:

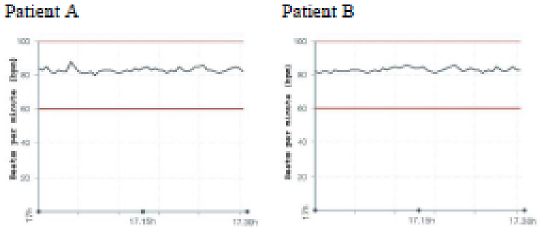
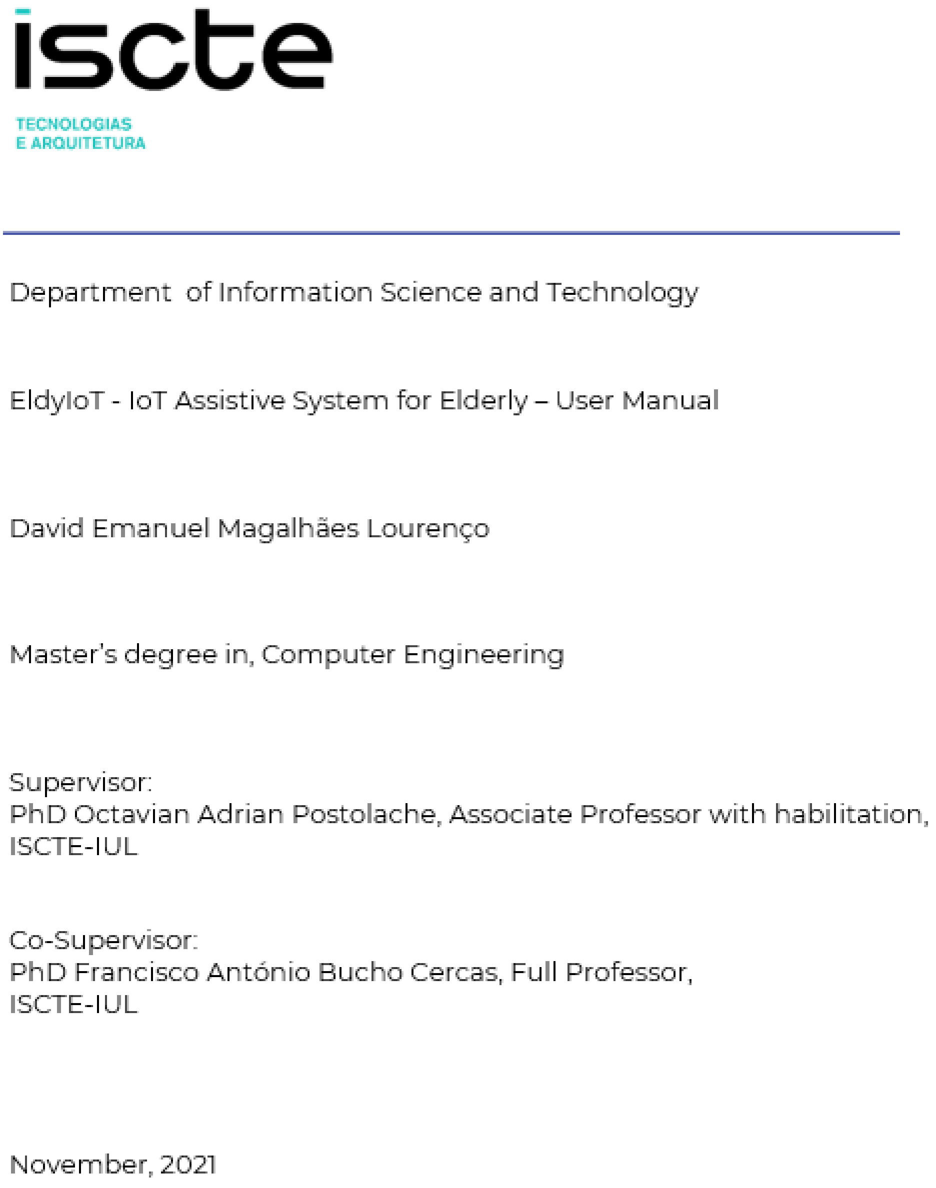


Fig 5 – Heart rate data extracted during the walk (bpm/h)

According to the Heart Attack and Stroke Symptoms organization, the recommended beats per minute values for an adult at rest range from 60 to 100. For moderate physical activity for adults over the age of 70, the range is 75-128 bpm. Thus, the results obtained from the experiments fit the pattern. We were able to observe the variations in the heart rate of each patient according to the activity being performed, and to monitor and analyze that the healthy limits are not exceeded. During the experiments, a history was created for observation in the database and no alerts were generated. Throughout the experimentation a third patient was introduced, so that we could test the application's alert system. As such, the system was used during sports practice with the objective of generating alerts. In Fig 6, we have the result of this experiment.

Appendix B – User Manual



iscte
TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

EldyIoT - IoT Assistive System for Elderly – User Manual

David Emanuel Magalhães Lourenço

Master's degree in, Computer Engineering

Supervisor:
PhD Octavian Adrian Postolache, Associate Professor with habilitation,
ISCTE-IUL

Co-Supervisor:
PhD Francisco António Bucho Cercas, Full Professor,
ISCTE-IUL

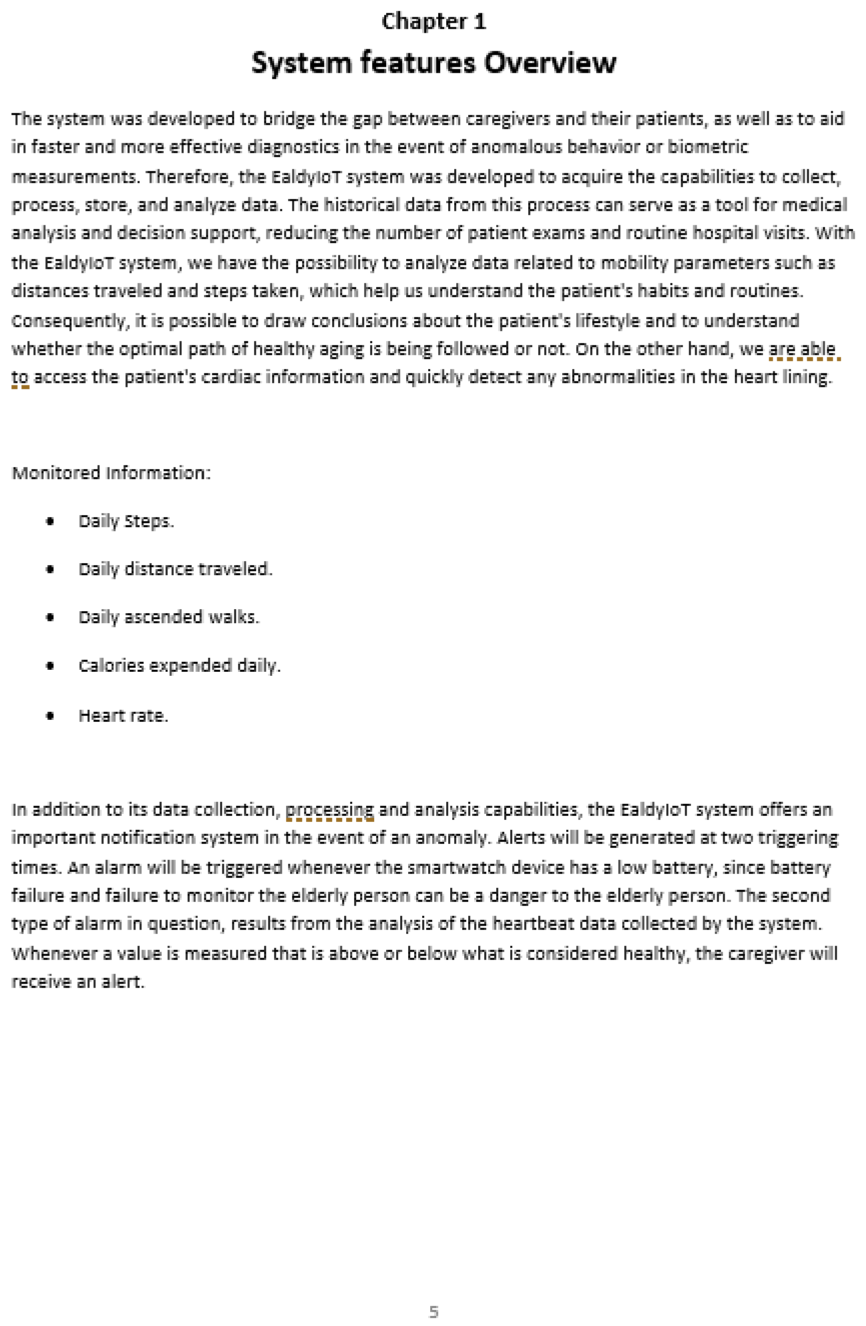
November, 2021

Contents

List of figures

User Manual

This manual aims to show the features available in the EaldyIoT framework and explain how they work. In chapter 1, it intends to explain the functionalities of the smartwatch that has the purpose of collecting data related to the patient, so that we can build a history to support some decision making. In chapter 2, it intends to explain the functionalities of the application that serves as a support for the knowledge and analysis of the patient's health status. Chapter 3 intends to explain how to install the application.



Chapter 1

System features Overview

The system was developed to bridge the gap between caregivers and their patients, as well as to aid in faster and more effective diagnostics in the event of anomalous behavior or biometric measurements. Therefore, the EaldyIoT system was developed to acquire the capabilities to collect, process, store, and analyze data. The historical data from this process can serve as a tool for medical analysis and decision support, reducing the number of patient exams and routine hospital visits. With the EaldyIoT system, we have the possibility to analyze data related to mobility parameters such as distances traveled and steps taken, which help us understand the patient's habits and routines. Consequently, it is possible to draw conclusions about the patient's lifestyle and to understand whether the optimal path of healthy aging is being followed or not. On the other hand, we are able to access the patient's cardiac information and quickly detect any abnormalities in the heart lining.

Monitored Information:

- Daily Steps.
- Daily distance traveled.
- Daily ascended walks.
- Calories expended daily.
- Heart rate.

In addition to its data collection, processing and analysis capabilities, the EaldyIoT system offers an important notification system in the event of an anomaly. Alerts will be generated at two triggering times. An alarm will be triggered whenever the smartwatch device has a low battery, since battery failure and failure to monitor the elderly person can be a danger to the elderly person. The second type of alarm in question, results from the analysis of the heartbeat data collected by the system. Whenever a value is measured that is above or below what is considered healthy, the caregiver will receive an alert.

Chapter 2

EaldyIoT Smartwatch

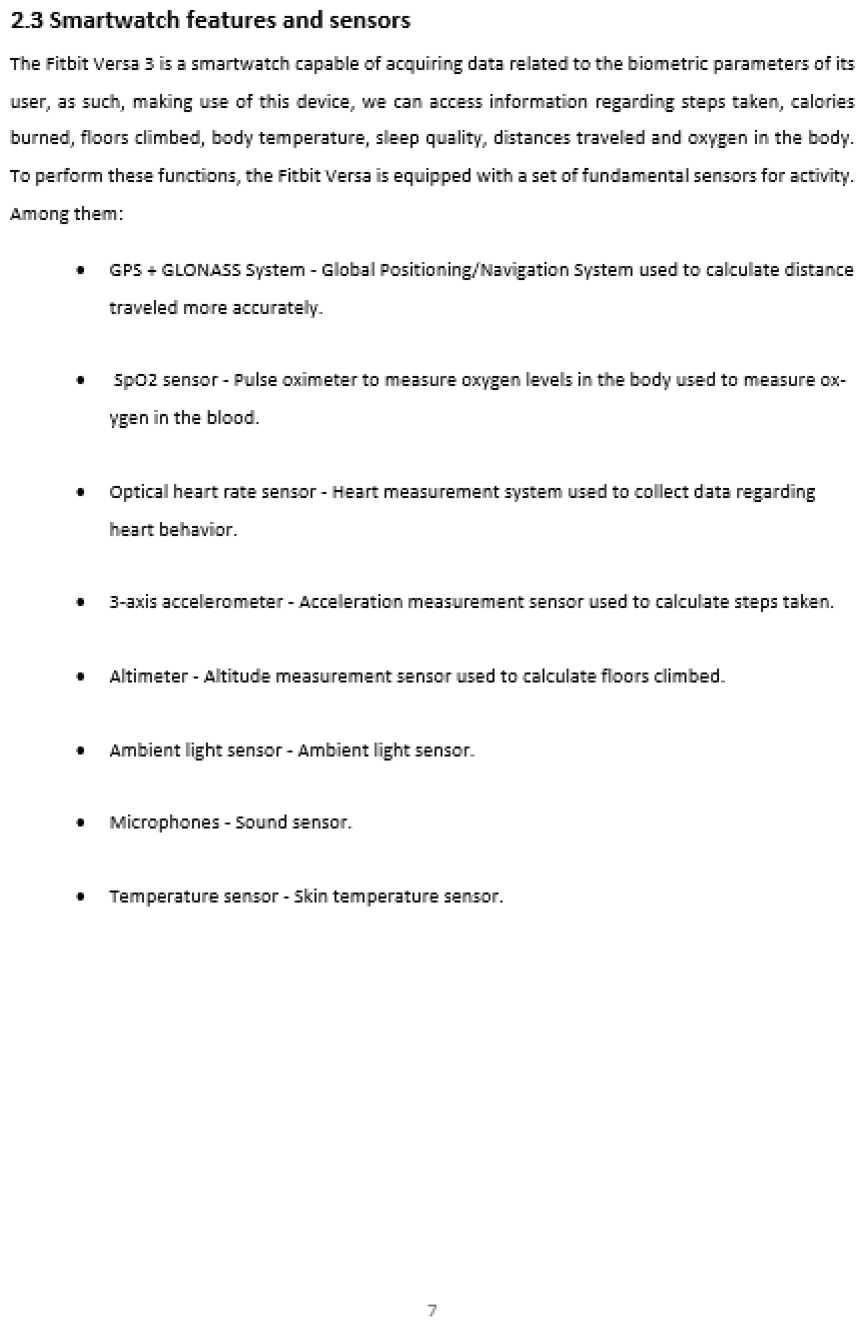
2.1 Overview

This chapter aims to present the smartwatch used in the framework as well as explain its functionality. Section 2.2 presents the smartwatch kit from the factory, and section 2.3 presents the smartwatch used to develop the solution and its functions in detail.

2.2 Smartwatch Kit

The smartwatch presented in this chapter is the Fitbit Versa 3. Figure 2 shows the delivered kit, consisting of smartwatch, USB charger, and a silicone bracelet.

Figure 1 - Fitbit Smartwatch Kit



2.3 Smartwatch features and sensors

The Fitbit Versa 3 is a smartwatch capable of acquiring data related to the biometric parameters of its user, as such, making use of this device, we can access information regarding steps taken, calories burned, floors climbed, body temperature, sleep quality, distances traveled and oxygen in the body. To perform these functions, the Fitbit Versa is equipped with a set of fundamental sensors for activity. Among them:

- GPS + GLONASS System - Global Positioning/Navigation System used to calculate distance traveled more accurately.
- SpO2 sensor - Pulse oximeter to measure oxygen levels in the body used to measure oxygen in the blood.
- Optical heart rate sensor - Heart measurement system used to collect data regarding heart behavior.
- 3-axis accelerometer - Acceleration measurement sensor used to calculate steps taken.
- Altimeter - Altitude measurement sensor used to calculate floors climbed.
- Ambient light sensor - Ambient light sensor.
- Microphones - Sound sensor.
- Temperature sensor - Skin temperature sensor.

Chapter 3

EaldyIoT Mobile Application

4.1 Overview

This chapter aims to describe the functionalities of the mobile application developed for the presentation of the data collected by the smartwatch sensors and further processed by the framework.

The main function of the EaldyIoT application is data visualization. In the EaldyIoT system, the application offers the caregiver an easy and quick analysis of data related to the patient, as well as serving as a support tool to detect anomalies in the patient's health status. Alerts are generated in case of detection of abnormal conditions, and this reduces the time to diagnose or provide service in case of emergency. In section 4.2 we will explain in detail how each of the screens of the application works.

4.2 Mobile Application Screens

In figure 2, we have the login screen. On this screen, we can login to start viewing the patient data associated with our account as well as we can proceed to the creation of a user account.

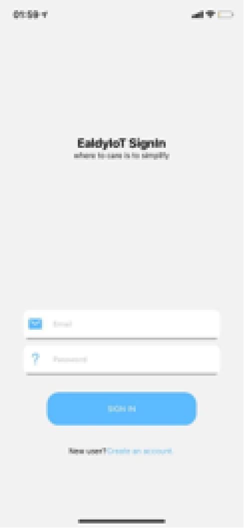


Figure 2 – Sign In Screen

Sign Up Screen

Figure 3 shows the registration screen, where we can create an EaldyIoT account or, in turn, jump to the application's login screen.

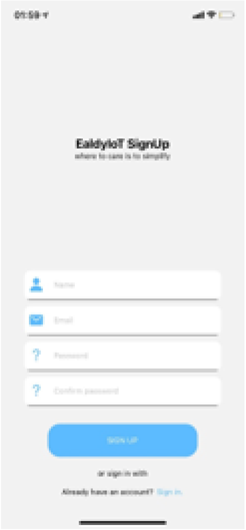


Figura 3 – Sign Up Screen

Account Screens

In figure 4, we have the account information screen, where the account id, email and name of the main user of the account are shown.

Account Info

ID

Email

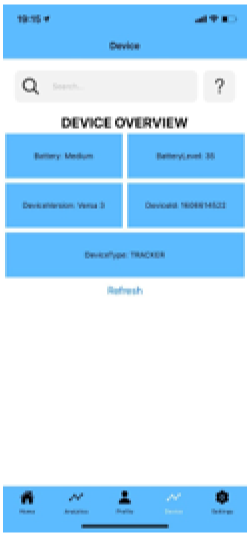
Name

User 1

Figure 4 – Account Info

Device Screen

In Figure 5, we have the device status screen with some information about the device. We can see the battery status, which can be useful in case an informal caregiver does not visit the patient daily and does not have daily contact with the smartwatch, as well as the device ID, which can be useful in case it is necessary to contact fitbit and ask for some clarification on the functioning of the device itself. In addition, it is possible to force an update of the information by means of a refresh button.



DEVICE OVERVIEW

Battery: Medium

BatteryLevel: 35

DeviceVersion: Versa 3

DeviceType: TRACKER

Figure 5 – Definition's screen

Analytic Screens

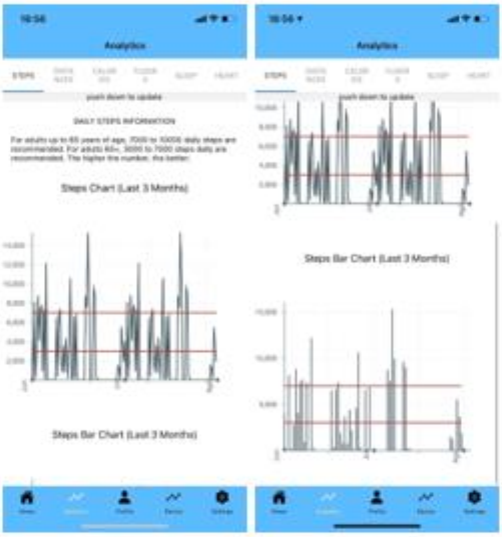
The analysis screens are one of the most important and useful parts of the system in question. From the detailed and careful analysis of the measured parameters, we can draw important conclusions about the patient's health status, as well as his daily routines.

The Graph in figure 6, presents us with a snapshot of information regarding the mobility and health status of the monitored patient. The purpose of this screen is to give a quick and clear perception of the patient's health status and whether or not the daily goals of active and healthy aging are being met. On the screen, information is dynamically loaded, and is displayed in different colors depending on the achievement of the stipulated goal. Green when the goal is reached. Red when the goal has not been reached.

Next, in figures 7 to 12, the data analysis screens will be shown, built in such a way that a medium/long term analysis of the data collected by the smartwatch is possible. These include daily number of steps over time, daily distance traveled over time, calories burned daily over time, daily number of floors climbed over time, daily sleep quality, and heart rate measured over the course of the day. For all the above indicators, we have the possibility to analyze the data collected from the patient in graphical form, facilitating the diagnosis in case any anomaly is detected. All graphs are accompanied by upper and lower limits according to the recommended for adults between 65 and 80 years. The only exception is the sleep quality analysis screen. This screen has an additional graph, which gives an idea of how the time the patient spent in bed was distributed between the different sleep states.

In figure 6 we have a refresh button, which forces a page refresh and returns the most up-to-date data. In any of the other data analysis screens, we can also force the data to be updated but, this time, by swiping down on the screen.

Figure 6 – Daily Overview Screen



DAILY STEPS INFORMATION

For adults up to 65 years of age, 7000 to 10000 daily steps are recommended. For adults 65+, 3000 to 7000 steps daily are recommended. The higher the number, the better.

Figure 7 – Step Analysis Screens (Line and Bar Graphs)

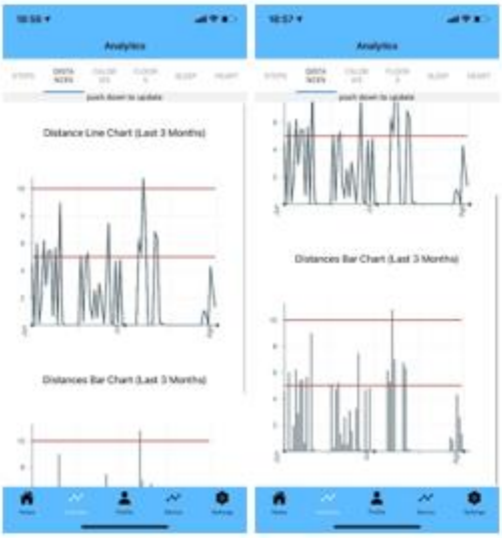


Figure 8 – Distance analysis screens (Line and Bar Graphs)

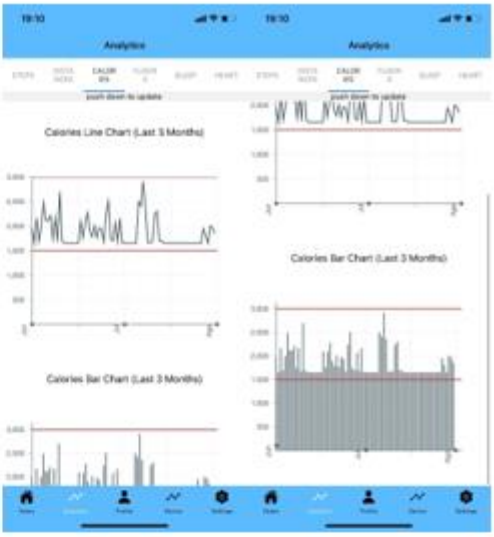


Figure 9 – Calorie's analysis screens (Line and Bar Graphs)

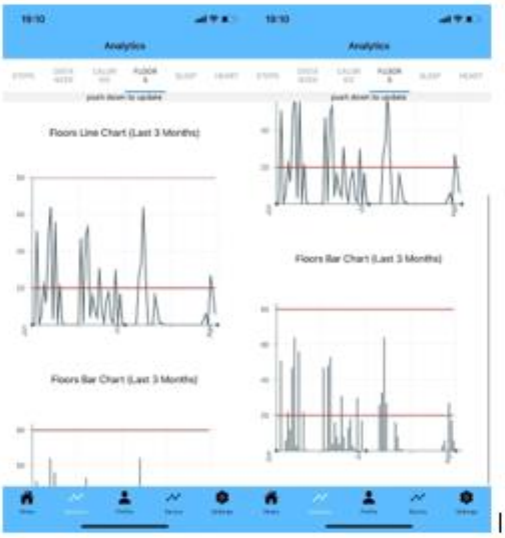


Figure 10 – Floors Analysis Screens (Line and Bar Graphs)

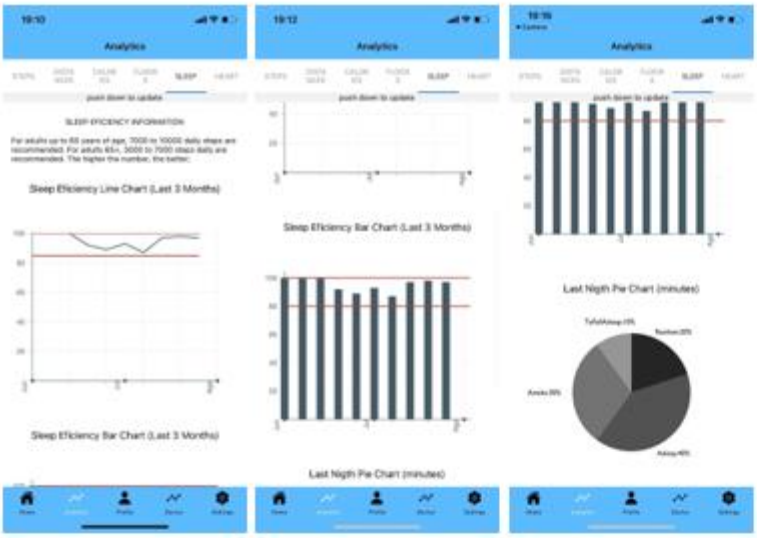


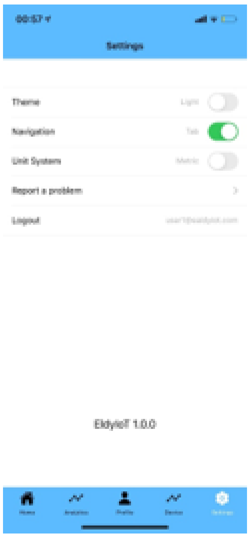
Figure 11 – Sleep analysis screens (Line graph, bar graph and circle graph)

The recommended heart rate value for an adult at rest should be between 60 and 100 bpm;

Figure 12 – Heartbeat analysis screens (Line and Bar Graphs)

Settings Screen

In Figure 13, we have the application's settings screen. We have the possibility to change the theme, from light mode to dark mode. We have the possibility to change the navigation, from a bar at the bottom of the screen to a sidebar, as well as to logout of the account in question.



Settings

Theme Light

Navigation Tab

Unit System Metric

Report a problem

Logout

EldyIoT 1.0.0

Figure 13 – Settings Screen

Notification System

In figure 14, we have the screen showing the representation of a notification generated by the application on the caregiver's device with an alert of a heart rate value outside the limits stipulated as normal for the patient in question. These alerts are generated when abnormal bpm (beats per minute) values are detected, below or above normal values

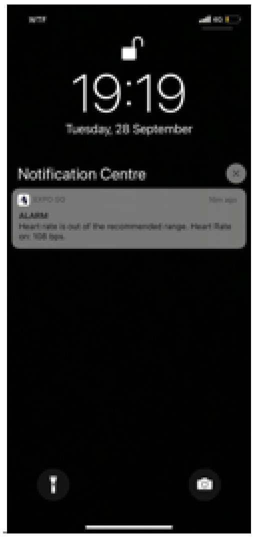
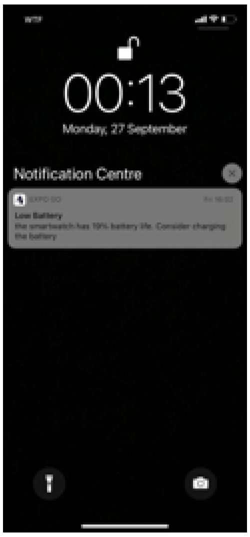


Figure 14 – Heartbeat above expected alert

In figure 15, we have the screen that shows the representation of a notification generated by the application on the caregiver's device with an alert regarding the battery status of the device (smartwatch). This type of notification can be very useful in cases where the caregiver is not with the patient every day. Therefore, he or she will receive successive alerts whenever the battery status changes and is below 20%.



00:13

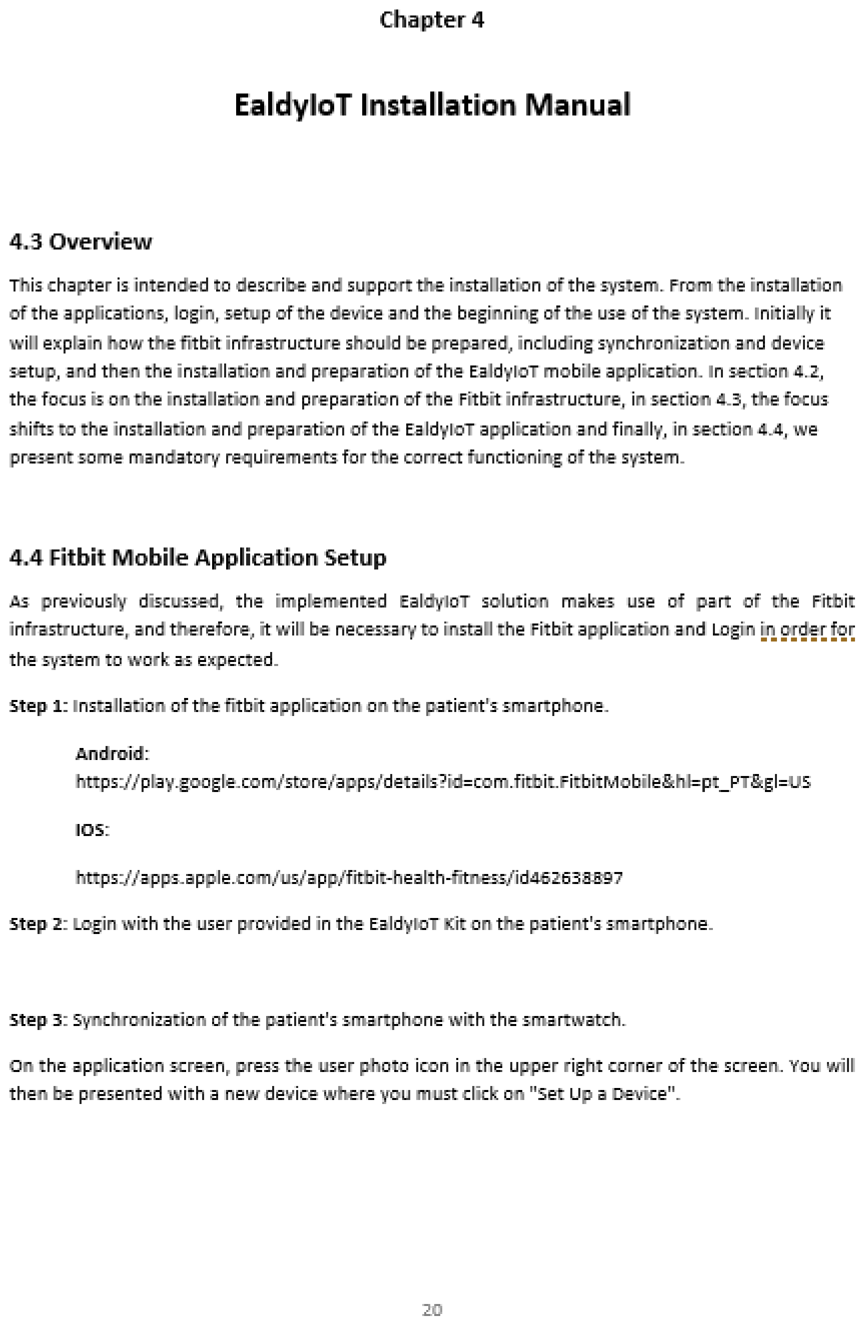
Monday, 27 September

Notification Centre

Low Battery

the smartwatch has 19% battery life. Consider charging the battery

Figure 15 – Discharged smartwatch battery alert



Chapter 4

EaldyIoT Installation Manual

4.3 Overview

This chapter is intended to describe and support the installation of the system. From the installation of the applications, login, setup of the device and the beginning of the use of the system. Initially it will explain how the fitbit infrastructure should be prepared, including synchronization and device setup, and then the installation and preparation of the EaldyIoT mobile application. In section 4.2, the focus is on the installation and preparation of the Fitbit infrastructure, in section 4.3, the focus shifts to the installation and preparation of the EaldyIoT application and finally, in section 4.4, we present some mandatory requirements for the correct functioning of the system.

4.4 Fitbit Mobile Application Setup

As previously discussed, the implemented EaldyIoT solution makes use of part of the Fitbit infrastructure, and therefore, it will be necessary to install the Fitbit application and Login in order for the system to work as expected.

Step 1: Installation of the fitbit application on the patient's smartphone.

Android:
https://play.google.com/store/apps/details?id=com.fitbit.FitbitMobile&hl=pt_PT&gl=US

IOS:

https://apps.apple.com/us/app/fitbit-health-fitness/id462638897

Step 2: Login with the user provided in the EaldyIoT Kit on the patient's smartphone.

Step 3: Synchronization of the patient's smartphone with the smartwatch.

On the application screen, press the user photo icon in the upper right corner of the screen. You will then be presented with a new device where you must click on "Set Up a Device".

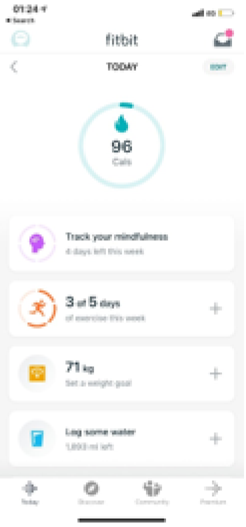


Figure 16 - Fitbit App Menu Screen

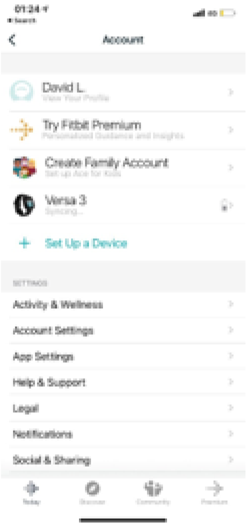


Figure 17 - Fitbit App Account Screen

Next you will be shown a different screen and asked to choose the device you want to configure. For this you should select the "Versa 3". To finish pairing, simply follow the steps shown on the application screen.

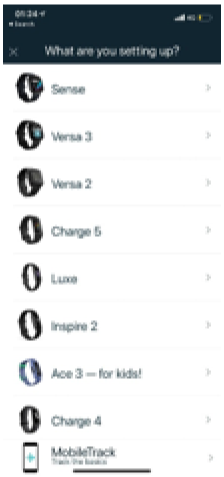


Figure 18 - Fitbit App Setting up Screen

Figure 19 - Fitbit smartwatch synchronization menu screen

Step 4: Enable the background run of the Fitbit app.

Android:

Settings -> Applications -> Fitbit APP -> Run in Background On

IOS:

Settings -> Scroll down to apps -> Select Fitbit app -> Update background app EaldyIoT

4.5 Mobile Application Setup

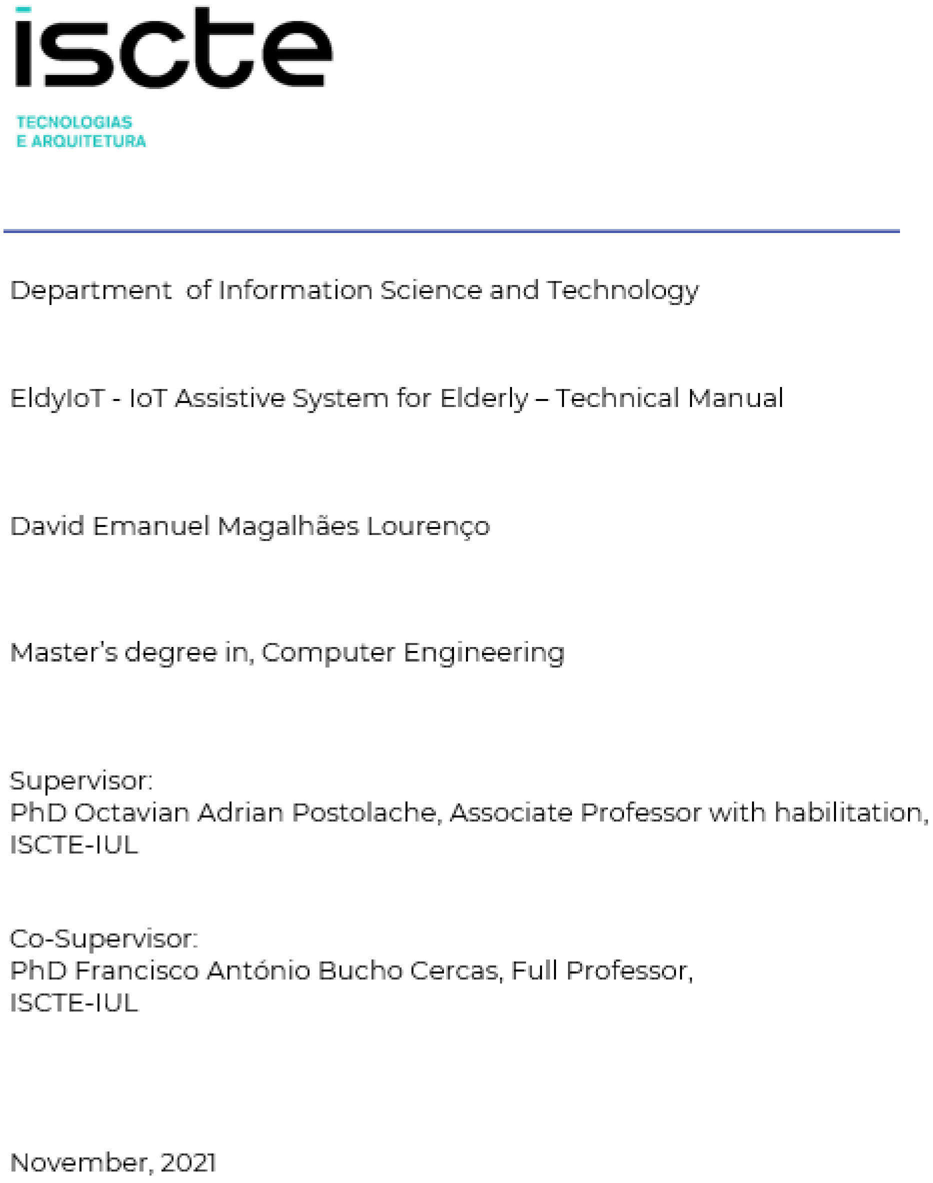
Step 1: Download and Install the EaldyIoT application on the Smartphone of the caregiver(s).

Step 2: Login with the user or users provided in the EaldyIoT Kit.

4.6 Mandatory Requirements:

- Internet connection active on the patient's smartphone.
- Battery saving deactivated on the patient's smartphone.

Appendix C – Technical Manual



Department of Information Science and Technology

EldyIoT - IoT Assistive System for Elderly – Technical Manual

David Emanuel Magalhães Lourenço

Master's degree in, Computer Engineering

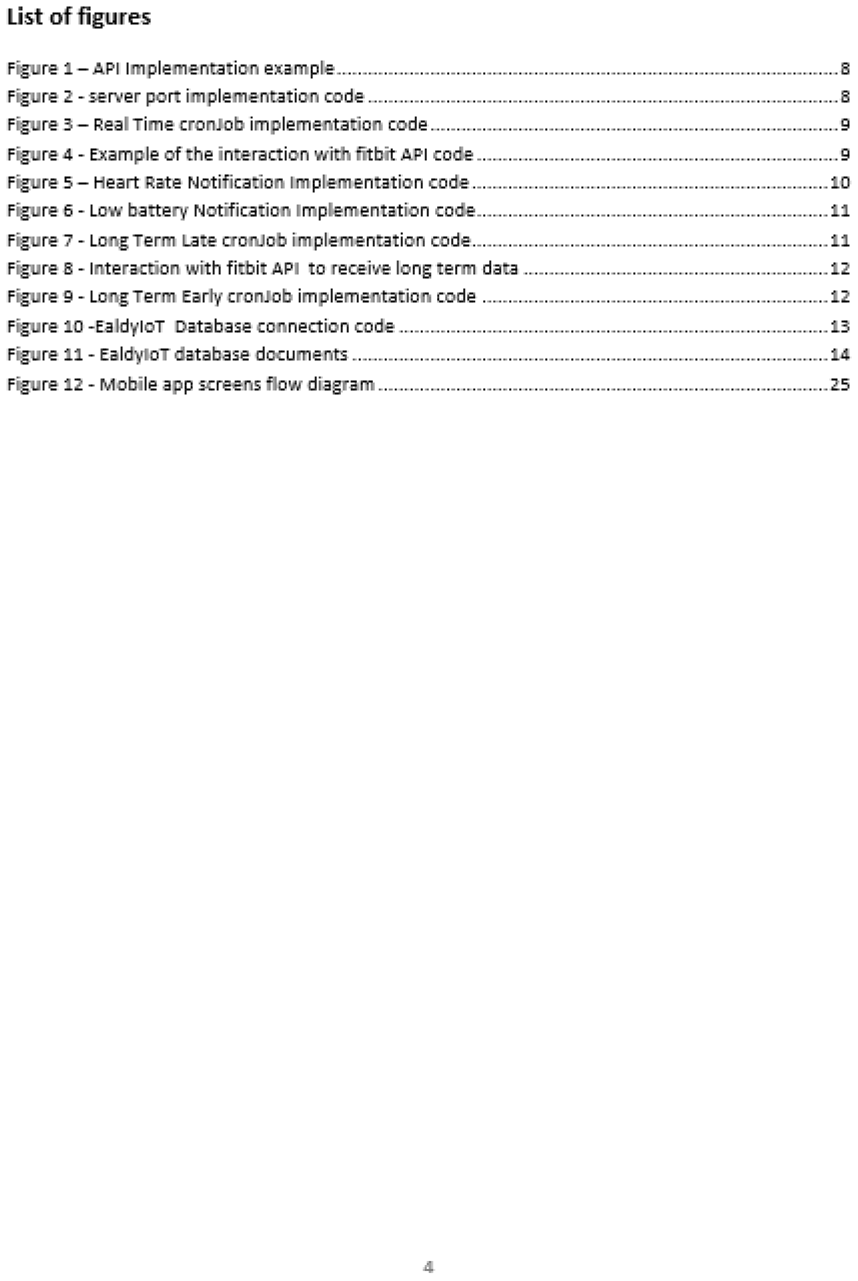
Supervisor:
PhD Octavian Adrian Postolache, Associate Professor with habilitation,
ISCTE-IUL

Co-Supervisor:
PhD Francisco António Bucho Cercas, Full Professor,
ISCTE-IUL

November, 2021

2

Contents



List of figures

List of Tables

Technical Manual

This manual aims to show the features available in the EaldyIoT framework and explain how they work. In chapter 1, it intends to explain the functionalities of the smartwatch that has the purpose of collecting data related to the patient, so that we can build a history to support some decision making. In chapter 2, it intends to explain the functionalities of the application that serves as a support for the knowledge and analysis of the patient's health status. Chapter 3 intends to explain how to install the application.

9

Chapter 2

Server Side

2.1 Overview

This chapter aims to clarify the implementation made in the development of the Server to support the EaldyIoT application. We'll go into more technical details at the code level as well as the structure of the application itself.

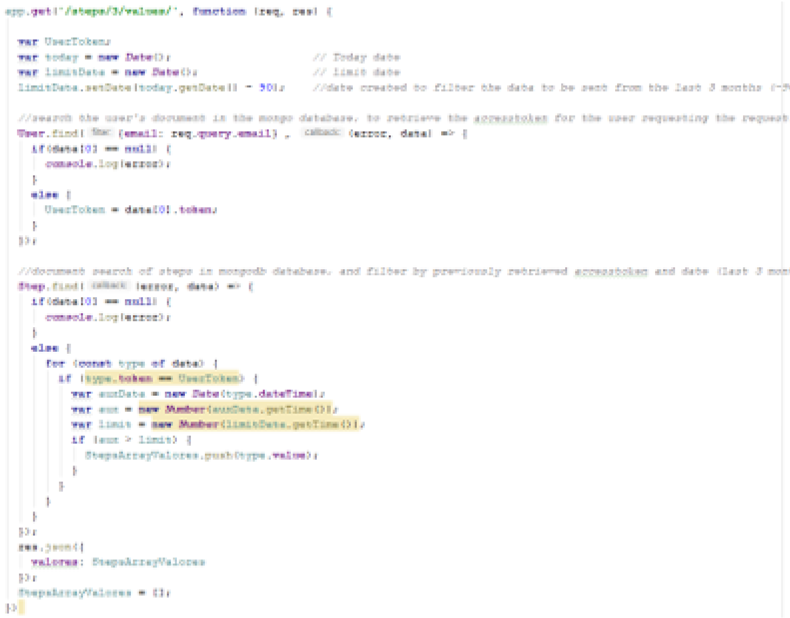
2.2 NodeJS Application

On the server side, the entire application is built in NodeJS. The application consists of two blocks with completely different functions. The first, with the function of collecting data from the Fitbit infrastructure, processing and storing it in the EaldyIoT database. The second module is capable of exposing API's that will make available the information stored by the previous process. When the server starts, both functions are available. The server starts to synchronize data from Fitbit database as well as APIS are available for possible requests.

Programing:

It is in App.js File that the code of all APIs exposed in the application is found. The exposed APIs aim to ensure communication between the mobile application and the server. This is how the application has access to the information it makes available on the screens.

Next, we have the explanatory example of the code snippet that exposes the API of the steps taken by the patient in the last 3 months.



```
app.get('/steps/3/values/', function (req, res) {

  var UserToken;
  var today = new Date();                      // Today date
  var limitDate = new Date();                  // limit date
  limitDate.setDate(today.getDate() - 90);     //date created to filter the data to be sent from the last 3 months (~90 days)

  //search the user's document in the mongo database, to retrieve the accesstoken for the user requesting the request.
  User.find( filter: {email: req.query.email} , callback: (error, data) => {
    if(data[0] == null) {
      console.log(error);
    }
    else {
      UserToken = data[0].token;
    }
  });

  //document search of steps in mongodb database, and filter by previously retrieved accesstoken and date (last 3 months)
  Step.find( callback: (error, data) => {
    if(data[0] == null) {
      console.log(error);
    }
    else {
      for (const type of data) {
        if (type.token == UserToken) {
          var auxDate = new Date(type.dateTime);
          var aux = new Number(auxDate.getTime());
          var limit = new Number(limitDate.getTime());
          if (aux > limit) {
            StepsArrayValores.push(type.value);
          }
        }
      }
    }
  });
  res.json({
    valores: StepsArrayValores
  });
  StepsArrayValores = [];
});
```

Figure 1 – API Implementation example

At the time of request, the mobile application sends to the server, the email of the account that is currently logged in, and by the way, requesting the information. As a way to ensure security, the server goes to the EaldyIoT database and checks if the user in question has legal access to any patient. If so, the information regarding that patient is returned, if not, an error is returned. The example above is related to the steps taken by the patient, but the same system works for all other API's. In the next chapter, we will present each of them in detail. Each of these API's is programmed to be exposed in port 3000 of the machine running.

```
const port = process.env.PORT || 3000;
```

Figure 2 - server port implementation code

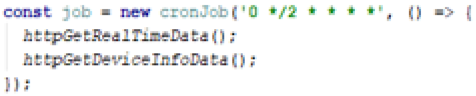
Therefore, any request is accessible via http:

Ex: http://ip:3000/steps/3/values/

Regarding the data synchronization process between the fitbit database and the EaldyIoT database, we can divide it into 3 sub-processes.

- core_real_time_data.js
- core_long_term_data_early.js
- core_long_term_data_late.js

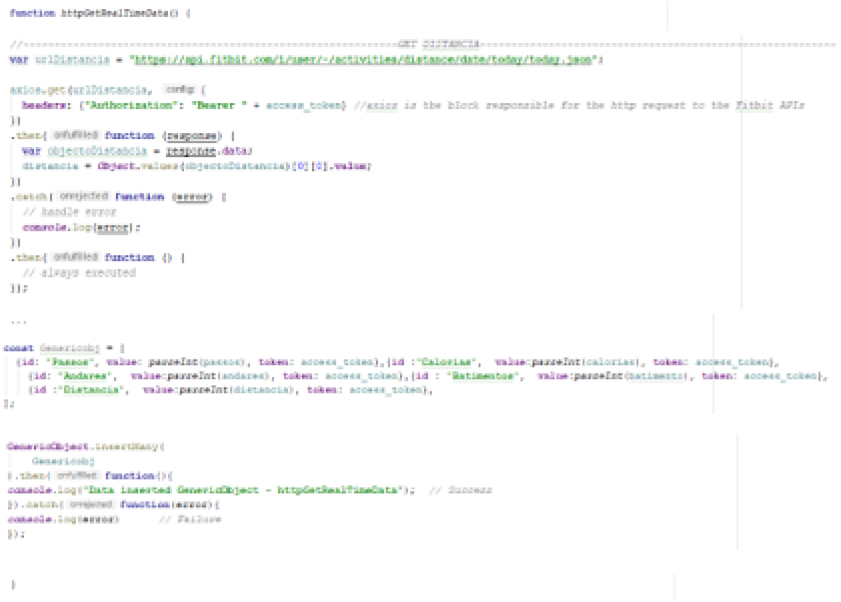
Each of the processes runs periodically and handles data for different purposes. The core_real_time_data.js process runs every 2 min in order to keep the objects, Generic Object and Device, with the most up-to-date information possible. Trying to provide information such as battery status, steps taken, calories used, floors climbed and distances covered in real time.



```
const job = new cronJob('0 */2 * * * *', () => {
  httpGetRealTimeData();
  httpGetDeviceInfoData();
});
```

Figure 3 – Real Time cronJob implementation code

In Figure 3 we can see the cronJob implementation responsible for running the functions httpGetRealTimeData() and httpGetDeviceInfoData() every two minutes.



```
function httpGetRealTimeData() {

//==============================================================GET DISTANCIA=============================================================
var urlDistancia = "https://api.fitbit.com/1/user/-/activities/distance/date/today/today.json";

axios.get(urlDistancia, {
  headers: {"Authorization": "Bearer " + access_token} //axios is the block responsible for the http request to the Fitbit APIs
})
.then( function (response) {
  var objectoDistancia = response.data;
  distancia = Object.values(objectoDistancia)[0][0].value;
})
.catch( function (error) {
  // handle error
  console.log(error);
})
.then( function () {
  // always executed
});

...

const Genericobj = [
  {id: "Passos", value: parseInt(passos), token: access_token},{id :"Calorias",  value:parseInt(calorias), token: access_token},
   {id: "Andares",  value:parseInt(andares), token: access_token},{id : "Batimentos",  value:parseInt(batimento), token: access_token},
   {id :"Distancia",  value:parseInt(distancia), token: access_token},
];


GenericObject.insertMany(
    Genericobj
).then( function(){
console.log("Data inserted GenericObject - httpGetRealTimeData");  // Success
}).catch( function(error){
console.log(error)      // Failure
});


 }
```

Figure 4 - Example of the interaction with fitbit API code

In figure 4, we have the code excerpt of the function httpGetRealTimeData() in which the information regarding the distance traveled by the patient until the exact time of the request is collected, the object responsible for the Real Time data (Generic Object) is created and inserted in the Mongo database.

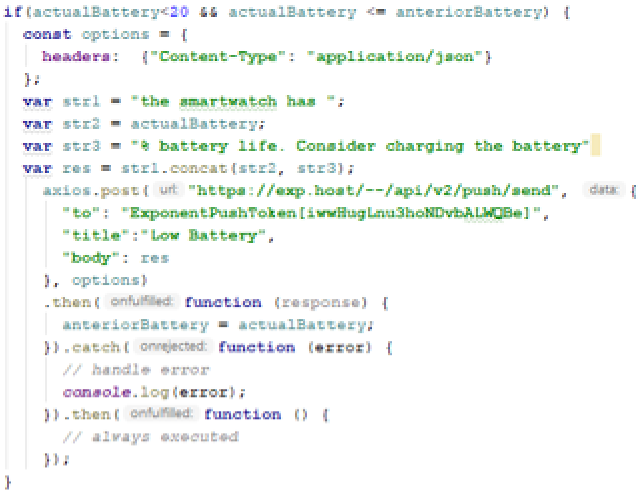
As we can see, the GenericObject object is constituted by other values besides the km traveled by the patient, as such, the previous code snippet only explains how the distance and all other values are collected.

Axios is used as an HTTP client to make requests to the mongo API. Then the received object is processed, and useful information is removed. The GenericObject object is created and inserted into the mongo database to which we connect at the beginning of the code. Regarding the connection to the database, it will be explained in the section on databases. It is also in the core_real_time_data.js process that alerts are generated. Two types of alerts are generated, related to cardiac monitoring, and related to the device status. To generate alerts in the form of notifications, the Expo system was used. Expo notifications.

```
if(batimento_anterior!=batimento) {
  if((batimento>90 && passos == passosAntigo && passos != 0)|| batimento < 50) {
    var str1 = "Heart rate is out of the recommended range. Heart Rate on: ";
    var str2 = batimento;
    var str3 = " bpm."
    var body = str1.concat(str2, str3);

    axios.post( url: "https://exp.host/--/api/v2/push/send",  data: {
      "to": "ExponentPushToken[iwwHugLnu3hoNDvbALWQBe]",
      "title":"ALARM",
      "body": body
    }, options)
    .then( onfulfilled: function (response) {
      batimento_anterior=batimento;
    })
    .catch( onrejected: function (error) {
     // handle error
      console.log(error);
    })
    .then( onfulfilled: function () {
      // always executed
    });
  }
```

Figure 5 – Heart Rate Notification Implementation code

In the previous figure, we have the code snippet responsible for generating alerts related to cardiac monitoring. While data relating to the patient's cardiac function are extracted, they are processed and verified that they are within expected values. As we can see in the previous figure, if the registered heartbeat is above 90 bpm and the patient is stopped, or the heartbeat is below 50 bpm, an alert will be generated.

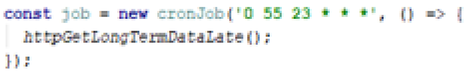


```
if(actualBattery<20 && actualBattery <= anteriorBattery) {
  const options = {
    headers:  {"Content-Type": "application/json"}
  };
  var str1 = "the smartwatch has ";
  var str2 = actualBattery;
  var str3 = "% battery life. Consider charging the battery"
  var res = str1.concat(str2, str3);
    axios.post( url: "https://exp.host/--/api/v2/push/send",  data: {
      "to": "ExponentPushToken[iwwHugLnu3hoNDvbALWQBe]",
      "title":"Low Battery",
      "body": res
    }, options)
    .then( onfulfilled: function (response) {
      anteriorBattery = actualBattery;
    }).catch( onrejected: function (error) {
      // handle error
      console.log(error);
    }).then( onfulfilled: function () {
      // always executed
    });
}
```

Figure 6 - Low battery Notification Implementation code

In the previous figure, we have the code snippet related to the implementation of alerts related to the device's battery status, which follows the same structure and logic.

The core_long_term_data_late.js process that runs every day at 23.55h.



```
const job = new cronJob('0 55 23 * * *', () => {
  httpGetLongTermDataLate();
});
```

Figure 7 - Long Term Late cronJob implementation code

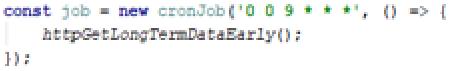
This process collects data on distances covered, calories used, floors climbed and steps taken in the last 3 months, including the last 24h. Although the following figure only shows the code snippet related to the collection of data on the distance traveled, for calories, floors climbed and steps taken, the way we collect the remaining data is the same, just changing the URL of the GET request.

```
//--------------------------------------------------------------GET DISTANCE----------------------------
var str1 = "https://api.fitbit.com/1/user/-/activities/distance/date/";
var str2 = "/15min.json";
var res = str1.concat(today,"/" ,threeMonthsEarlierDate ,str2);

axios.get(res,  config: {
  headers: {"Authorization": "Bearer " + access_token}
})
.then( onFulfilled: function (response) {
   Distance.remove( filter: (error, data) => {
     if(error) {
     console.log(error);
     }
   });
    var distancesArray = Object.values(response.data)[0];
    var output = [];

    function logArrayElements(element, index, array) {
        output.push({ dateTime: element.dateTime, value: element.value, token: access_token })
    }

    distancesArray.forEach(logArrayElements);

    Distance.insertMany(
      output
  ).then( onFulfilled: function(){
    console.log("Data inserted Distance - httpGetLongTermData")  // Success
  }).catch( onRejected: function(error){
    console.log(error)      // Failure
  });
}).catch( onRejected: function (error) {
  // handle error
  console.log(error);
}).then( onFulfilled: function () {
  // always executed
});
```

Figure 8 - Interaction with fitbit API to receive long term data

In the first 3 lines of code the URL is dynamically mounted. With the date of the day of order and with the date for the day 3 months before. After that, Axios is used again as http client to make the call to the mongo APIs, later processing the data and inserted via Mongoose into the mongo database.

And finally, core_long_term_data_early.js runs every day at 9am and collects data on how the past nights went up to the day in question.



```
const job = new cronJob('0 0 9 * * *', () => {
    httpGetLongTermDataEarly();
});
```

Figure 9 - Long Term Early cronJob implementation code

The way in which data is collected, processed and stored follows the same contours as the previous ones.

2.3 Database

The database used for the development of the system is a mongodb database. Mongoose was used as an Object Data Modeling (ODM) library to manage the relationship between the NodeJS code and the database itself.

Database Connection:

```
const mongoose = require('mongoose');

const URI ="mongodb+srv://root:root@cluster0.ifkyy.mongodb.net/EaldyIoT?retryWrites=true&w=majority";

const connectDB = async () => {
  await mongoose.connect(URI,  options: {
    useUnifiedTopology: true,
    useNewUrlParser: true
  });
  console.log('db connected..!');
};

module.exports = connectDB;
```

Figure 10 -EaldyIoT Database connection code

Connection to the database is done via URI asynchronously.

Database Documents:

Given the non-relational nature of the mongo database, the documents used for the functioning of the system will be presented and explained below.

```
//LastSleep Document
{
  date: {type: String},
  timeInBed: {type: Number},
  minutesrestless: {type: Number},
  minutesAsleep: {type: Number},
  minutesAwake: {type: Number},
  minutesToFallAsleep: {type: Number},
  token: { type: String }
}
```

```
//User Document
{
  name: { type: String },
  email: { type: String },
  password: { type: String },
  token: { type: String },
  prefixtoken: { type: String },
  mediumtoken: { type: String },
  suffixestoken: { type: String },
}
```

```
//Calories Document
{
  dateTime: { type: String },
  value: { type: Number },
  token: { type: String }
}
```

```
//Device Document
{
  id: { type: String },
  value: { type: String },
  token: { type: String }
}
```

```
//Distance Document
{
dateTime: { type: String },
value: { type: Number },
token: { type: String }
}
```

```
//Floor Document
{
  dateTime: { type: String },
  value: { type: Number },
  token: { type: String }
}
```

```
//HeartRate Document
{
  dateTime: {type: String},
  value: {type: Number},
  token: { type: String }
}
```

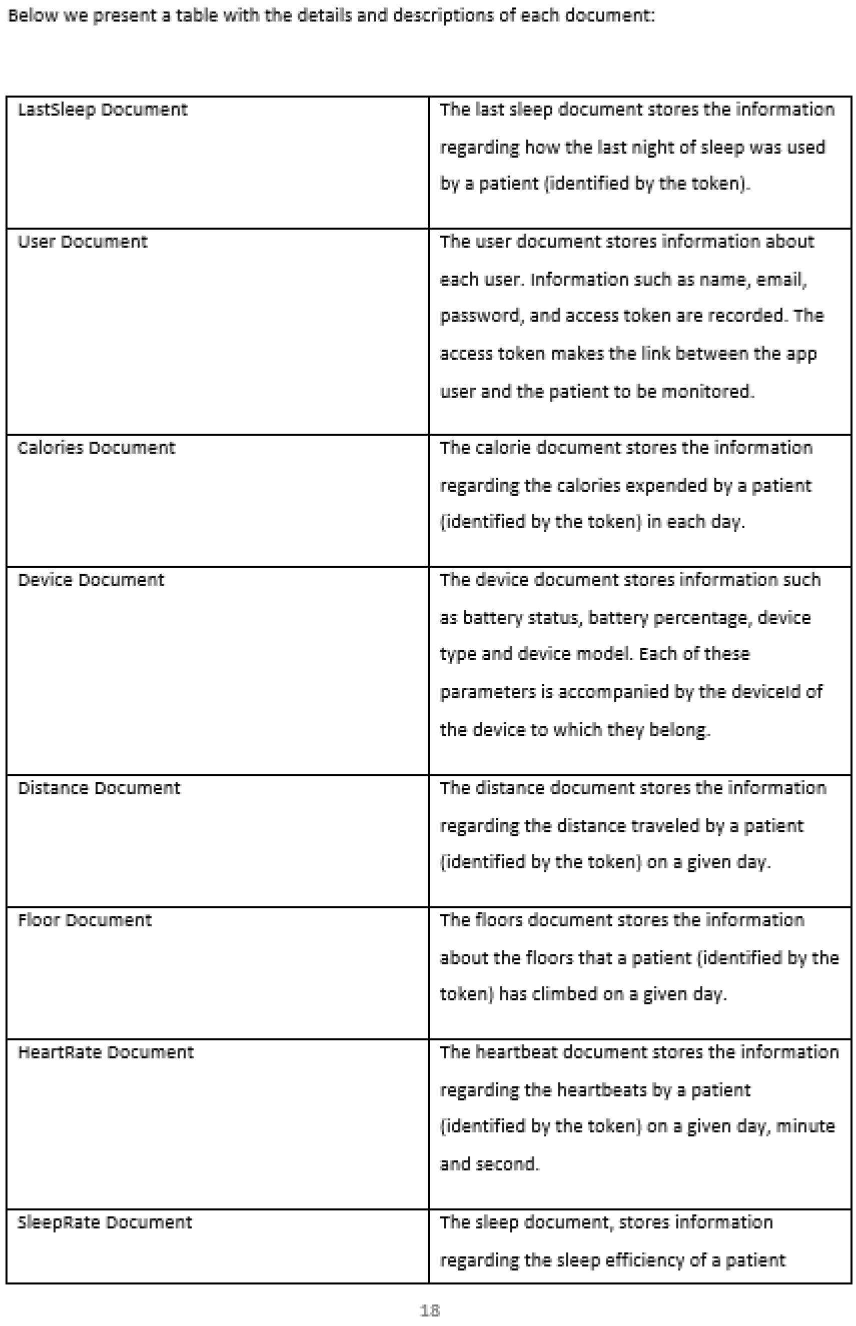
```
//SleepRate Document
{
  dateTime: {type: String},
  value: {type: Number},
  token: { type: String }
}
```

```
//Step Document
{
  dateTime: {type: String},
  value: {type: Number},
  token: { type: String }
}
```

```
//GenericObject Document
{
id: {type: String},
value: {type: Number},
uni: {type: String},
token: {type: String}
}
```

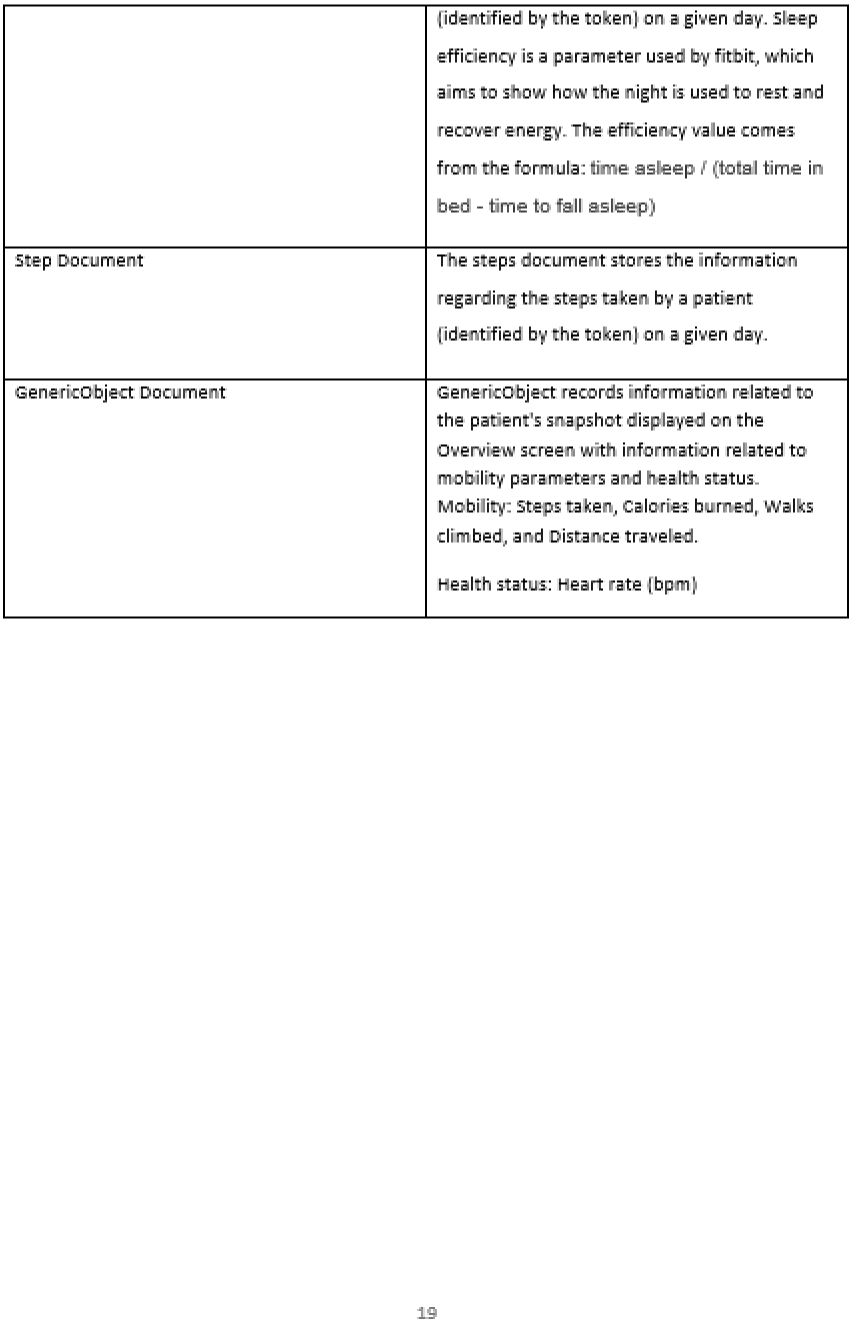
Figure 11 - EaldyIoT database documents

17

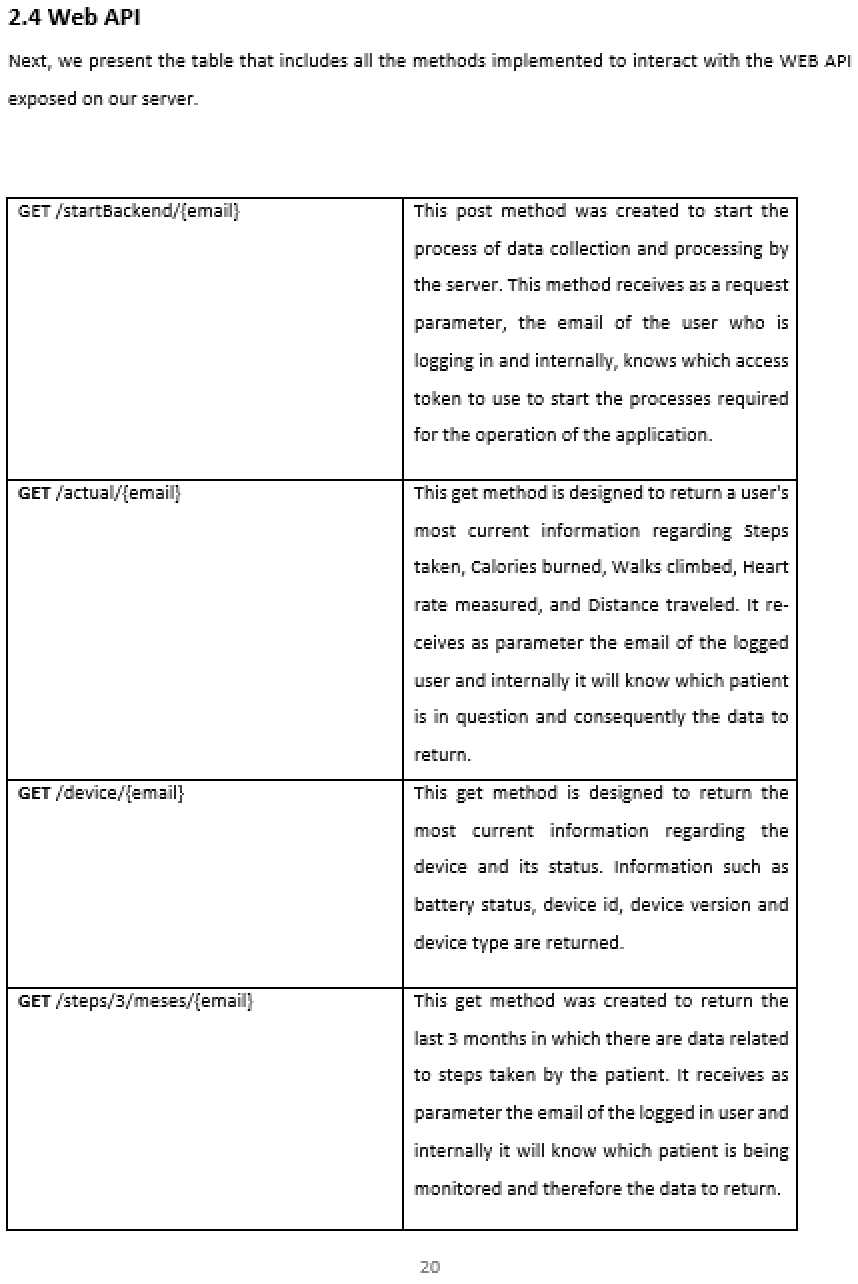


Below we present a table with the details and descriptions of each document:

LastSleep Document	The last sleep document stores the information regarding how the last night of sleep was used by a patient (identified by the token).
User Document	The user document stores information about each user. Information such as name, email, password, and access token are recorded. The access token makes the link between the app user and the patient to be monitored.
Calories Document	The calorie document stores the information regarding the calories expended by a patient (identified by the token) in each day.
Device Document	The device document stores information such as battery status, battery percentage, device type and device model. Each of these parameters is accompanied by the deviceId of the device to which they belong.
Distance Document	The distance document stores the information regarding the distance traveled by a patient (identified by the token) on a given day.
Floor Document	The floors document stores the information about the floors that a patient (identified by the token) has climbed on a given day.
HeartRate Document	The heartbeat document stores the information regarding the heartbeats by a patient (identified by the token) on a given day, minute and second.
SleepRate Document	The sleep document, stores information regarding the sleep efficiency of a patient



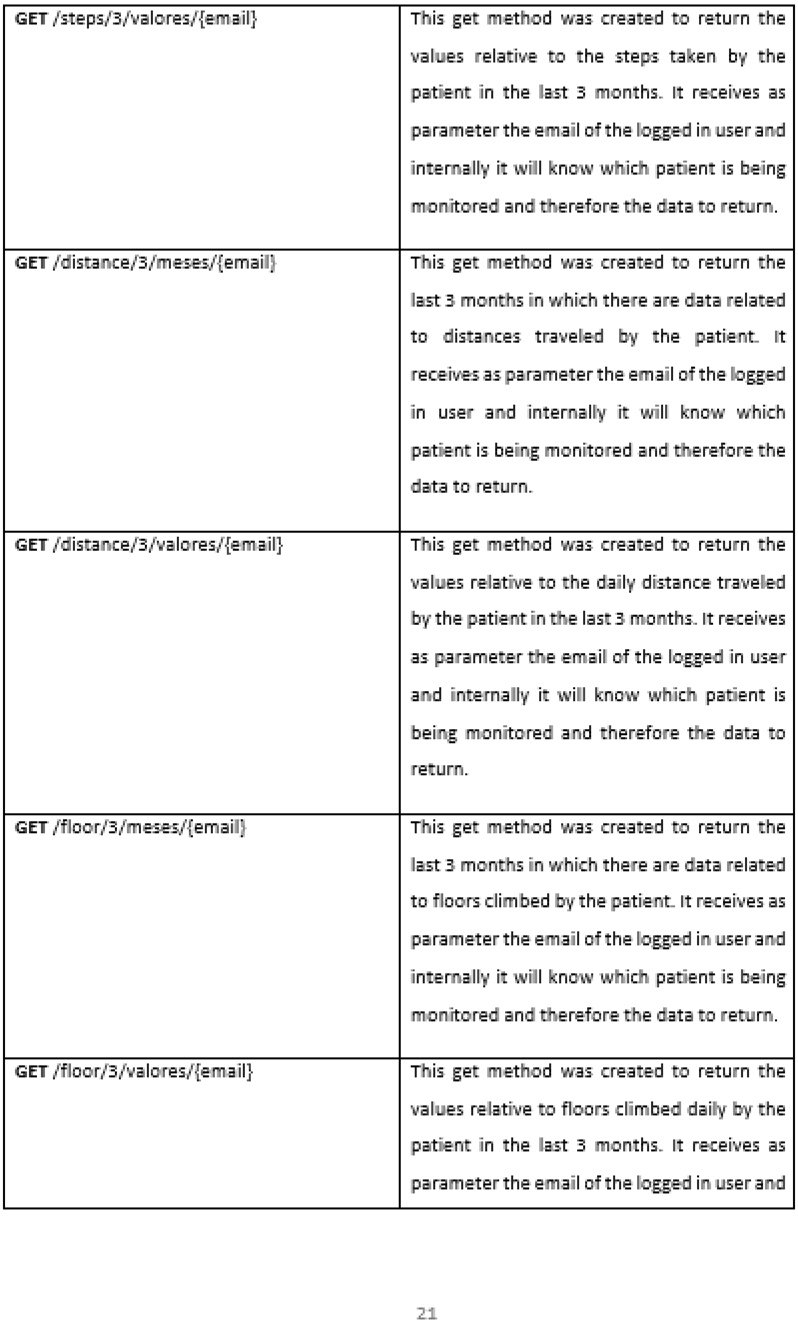
	(identified by the token) on a given day. Sleep efficiency is a parameter used by fitbit, which aims to show how the night is used to rest and recover energy. The efficiency value comes from the formula: time asleep / (total time in bed - time to fall asleep)
Step Document	The steps document stores the information regarding the steps taken by a patient (identified by the token) on a given day.
GenericObject Document	GenericObject records information related to the patient's snapshot displayed on the Overview screen with information related to mobility parameters and health status. Mobility: Steps taken, Calories burned, Walks climbed, and Distance traveled. Health status: Heart rate (bpm)



2.4 Web API

Next, we present the table that includes all the methods implemented to interact with the WEB API exposed on our server.

GET /startBackend/{email}	This post method was created to start the process of data collection and processing by the server. This method receives as a request parameter, the email of the user who is logging in and internally, knows which access token to use to start the processes required for the operation of the application.
GET /actual/{email}	This get method is designed to return a user's most current information regarding Steps taken, Calories burned, Walks climbed, Heart rate measured, and Distance traveled. It receives as parameter the email of the logged user and internally it will know which patient is in question and consequently the data to return.
GET /device/{email}	This get method is designed to return the most current information regarding the device and its status. Information such as battery status, device id, device version and device type are returned.
GET /steps/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to steps taken by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.



GET /steps/3/valores/{email}	This get method was created to return the values relative to the steps taken by the patient in the last 3 months. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /distance/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to distances traveled by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /distance/3/valores/{email}	This get method was created to return the values relative to the daily distance traveled by the patient in the last 3 months. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /floor/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to floors climbed by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /floor/3/valores/{email}	This get method was created to return the values relative to floors climbed daily by the patient in the last 3 months. It receives as parameter the email of the logged in user and

	internally it will know which patient is being monitored and therefore the data to return.
GET /calories/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to calories burned by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /calories/3/valores/{email}	This get method was created to return the values relative to calories expended daily by the patient in the last 3 months. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /batimentos/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to steps taken by the patient. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /batimentos/3/valores/{email}	This get method was created to return the daily heartbeat values of the patient in the last 24h. It receives as parameter the email of the logged user and internally will know which patient is being monitored and therefore the data to return.
GET /sleep/3/meses/{email}	This get method was created to return the last 3 months in which there are data related to the patient's sleep efficiency. It receives as parameter the email of the logged user and

	internally will know which patient is being monitored and therefore the data to return.
GET /sleep/3/valores/{email}	This get method was created to return the values relative to the patient's daily sleep efficiency in the last 3 months. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.
GET /lastsleep/valores/{email}	This get method was created to return information about the quality of sleep of the patient during the last night. It receives as parameter the email of the logged in user and internally it will know which patient is being monitored and therefore the data to return.

Tabela 1 - Web API details

Chapter 3

Mobile App

3.1 Overview

The mobile application was developed entirely using react native. The choice of this language was because we wanted to develop a hybrid application model that would run on both android and iOS. As support tools, the Axios library was used as an HTTP client and to support the requests to the server.

This chapter aims to present the application's functionalities, as well as describe each screen present in the application.

3.2 Application Main Features

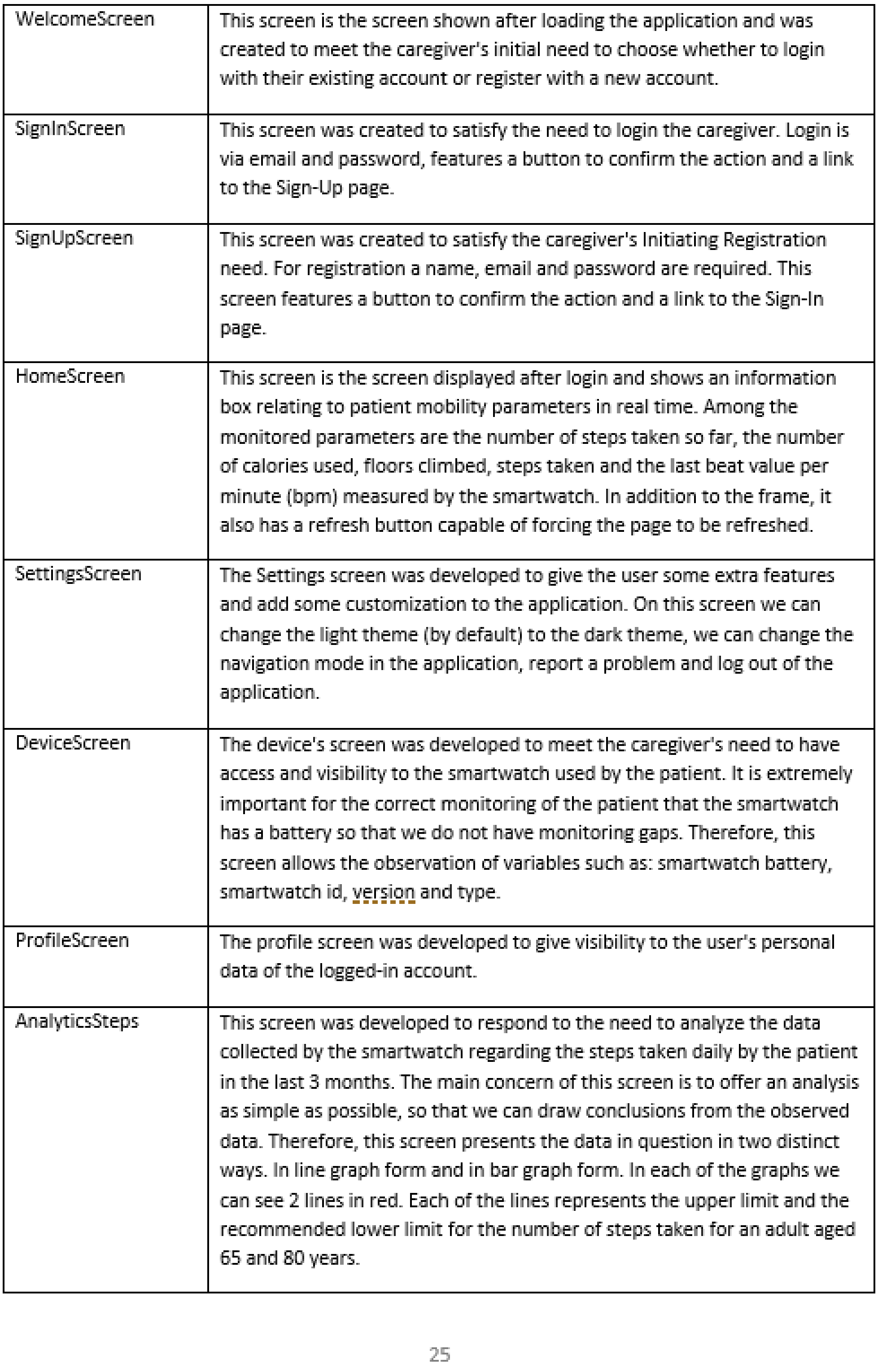
The mobile application appeared in the EaldyIoT system to meet the need to present the data collected by the smartwatch in a quick, simple, and intuitive way. Therefore, among the main features of the application we have:

- Visualization of patient data in a graphical and simplified way
- Visualization of health status patient data - Give the caregiver the ability to analyze health status data with the goal of recognizing abnormal values in parameters such as heartbeat.
- Visualization of mobility patient data - Give the caregiver the ability to analyze physical mobility data as distance traveled, floors climbed, steps taken, and calories burned with the goal of recognizing unhealthy patterns and abnormalities.
- Visualization of sleep quality data - Give the caregiver the ability to analyze sleep quality data with the goal of detecting abnormal sleep disturbances.
- Real time alerts - generation of alerts on the caregiver's cell phone in cases of danger or anomalies.
- Real time smartwatch status monitoring - the caregiver can access the status of the device used by the patient and assist in case of need.

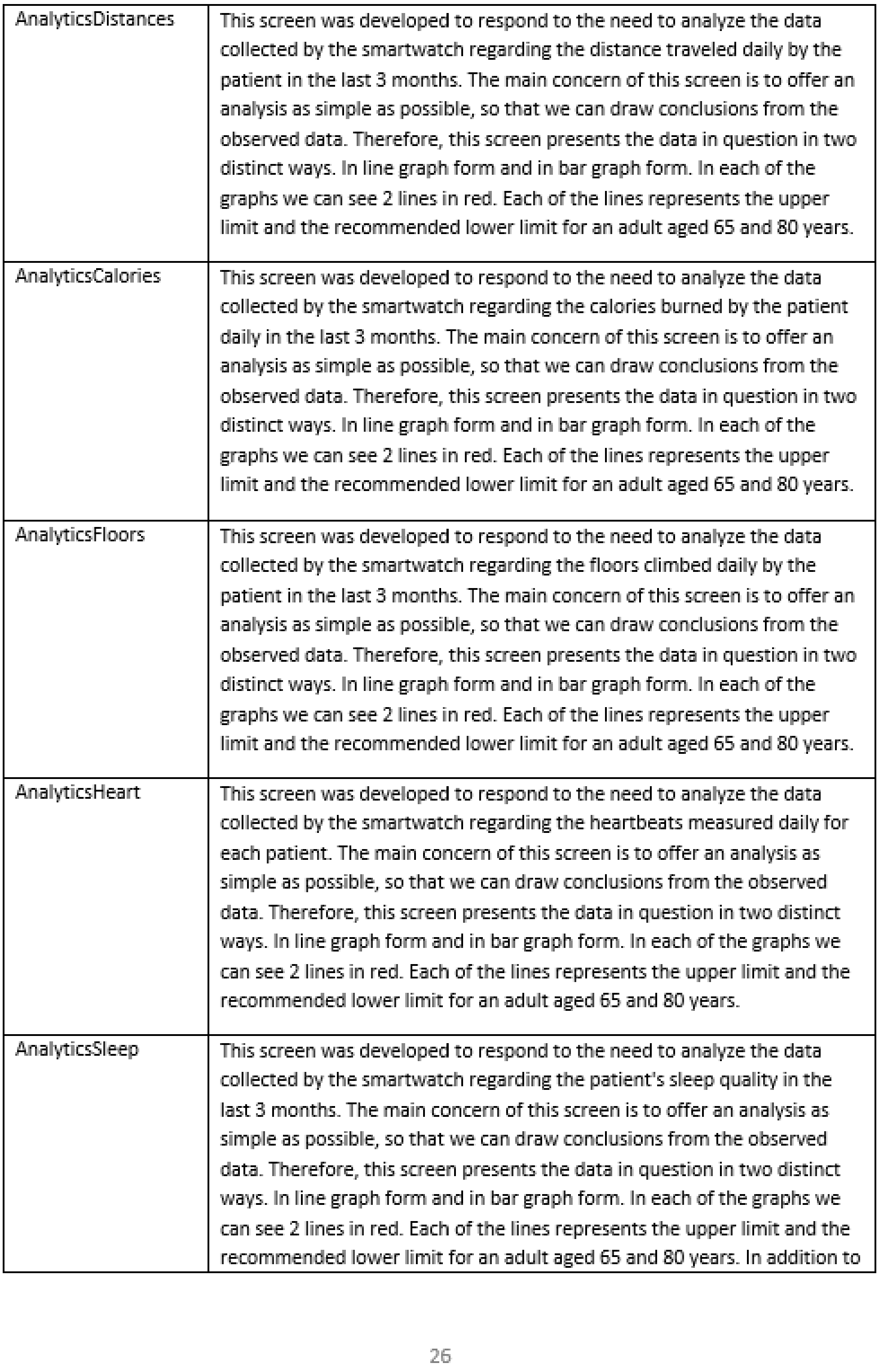
3.3 Screens

Below, we have a description of each of the application's screens, as well as its usefulness for the final result and the application's flow diagram.

Splash Screen	This screen serves as input to the application. While the application is starting, this screen is displayed with the system icon.



WelcomeScreen	This screen is the screen shown after loading the application and was created to meet the caregiver's initial need to choose whether to login with their existing account or register with a new account.
SignInScreen	This screen was created to satisfy the need to login the caregiver. Login is via email and password, features a button to confirm the action and a link to the Sign-Up page.
SignUpScreen	This screen was created to satisfy the caregiver's Initiating Registration need. For registration a name, email and password are required. This screen features a button to confirm the action and a link to the Sign-In page.
HomeScreen	This screen is the screen displayed after login and shows an information box relating to patient mobility parameters in real time. Among the monitored parameters are the number of steps taken so far, the number of calories used, floors climbed, steps taken and the last beat value per minute (bpm) measured by the smartwatch. In addition to the frame, it also has a refresh button capable of forcing the page to be refreshed.
SettingsScreen	The Settings screen was developed to give the user some extra features and add some customization to the application. On this screen we can change the light theme (by default) to the dark theme, we can change the navigation mode in the application, report a problem and log out of the application.
DeviceScreen	The device's screen was developed to meet the caregiver's need to have access and visibility to the smartwatch used by the patient. It is extremely important for the correct monitoring of the patient that the smartwatch has a battery so that we do not have monitoring gaps. Therefore, this screen allows the observation of variables such as: smartwatch battery, smartwatch id, version and type.
ProfileScreen	The profile screen was developed to give visibility to the user's personal data of the logged-in account.
AnalyticsSteps	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the steps taken daily by the patient in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for the number of steps taken for an adult aged 65 and 80 years.



AnalyticsDistances	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the distance traveled daily by the patient in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years.
AnalyticsCalories	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the calories burned by the patient daily in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years.
AnalyticsFloors	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the floors climbed daily by the patient in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years.
AnalyticsHeart	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the heartbeats measured daily for each patient. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years.
AnalyticsSleep	This screen was developed to respond to the need to analyze the data collected by the smartwatch regarding the patient's sleep quality in the last 3 months. The main concern of this screen is to offer an analysis as simple as possible, so that we can draw conclusions from the observed data. Therefore, this screen presents the data in question in two distinct ways. In line graph form and in bar graph form. In each of the graphs we can see 2 lines in red. Each of the lines represents the upper limit and the recommended lower limit for an adult aged 65 and 80 years. In addition to

	the charts explained above, for sleep monitoring, we also present a pie chart that shows how the patient's last night

Tabela 2 - Screens datails

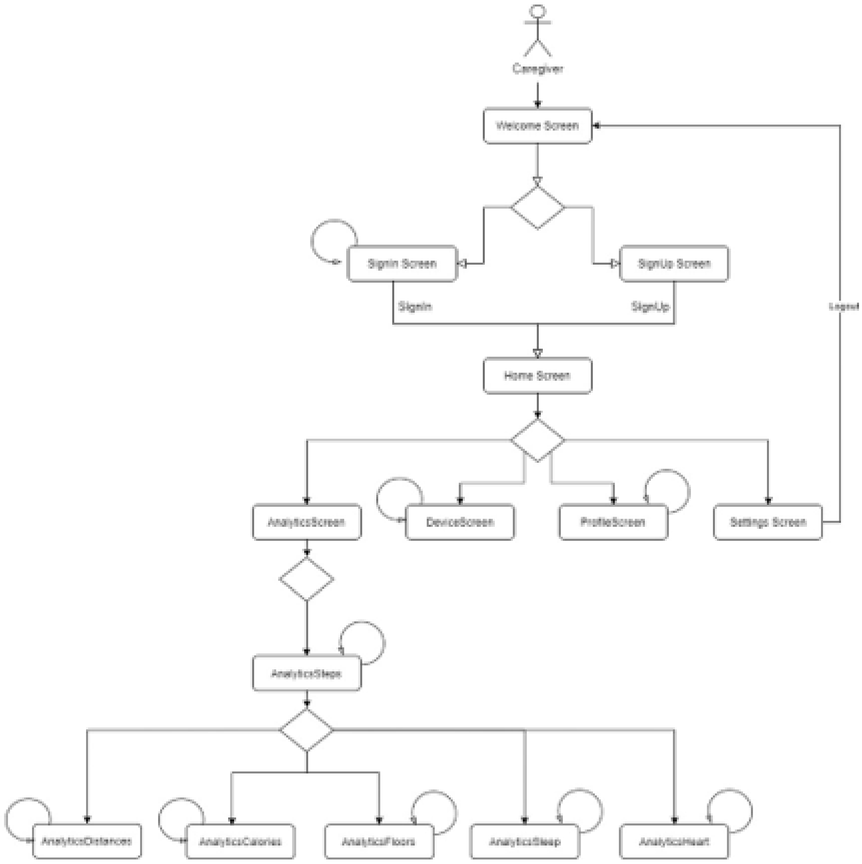


Figure 12 - Mobile app screens flow diagram