INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Remote Health Monitoring System for the Elderly based on Mobile Computing and IoT

Rafael da Silva Cardoso

Master in Telecommunications and Computer Engineering

Supervisor:
PhD Octavian Adrian Postolache, Associate Professor with aggregation
Iscte - Instituto Universitário de Lisboa

Co-Supervisor:
PhD Carlos Eduardo Dias Coutinho, Assistant Professor
Iscte - Instituto Universitário de Lisboa

November, 2022

TECHNOLOGY
AND ARCHITECTURE

Department of Information Science and Technology

Sistema de Monitorização Remota da Saúde para Idosos baseado em Computação Móvel e IoT

Rafael da Silva Cardoso

Mestrado em Engenharia de Telecomunicações e Informática

Supervisor:
Doutor Octavian Adrian Postolache, Professor associado com agregação
Iscte - Instituto Universitário de Lisboa

Co-Supervisor:
Doutor Carlos Eduardo Dias Coutinho, Professor auxiliar
Iscte - Instituto Universitário de Lisboa

Novembro, 2022

Acknowledgements

First of all, I would like to thank my advisor associate professor Dr. Octavian Postolache for all the support and guidance throughout the work. First of all, for having accepted to be my supervisor and propose a dissertation topic that captured my attention and dedication. For the trust deposited in me and total freedom of ideas that I had and wanted to develop in the work, for having made available all the materials needed and for guiding me to the creation of the scientific article later submitted, accepted and presented at an international conference.

I would also like to thank my co-advisor assistant professor. Dr. Carlos Coutinho, for all the support and guidance during all stages of this work. Mainly, for having helped a lot in the writing part of the dissertation, for having helped in the creation of an organizational methodology that allowed me to go through all the steps of the thesis without ever getting lost and for having helped a lot in the improvement of my presentation skills of the work done.

I would also like to thank ISCTE, because it allowed all this work to be done without any bureaucratic constraints, providing all the materials and resources available.

My acknowledgments go also to Instituto de Telecomunicações IT-IUL that supported my work during the whole period providing me technical advices and devices.

A thanks to several colleagues and friends, who helped me with their knowledge and experience in the development of the work.

I also want to thank my family, for their support and motivation in finishing my master's degree.

This work was also supported by Fundação para a Ciência e a Tecnologia, I.P. (FCT) [ISTAR Projects: UIDB/04466/2020 and UIDP/04466/2020] together with Instituto de Telecomunicações, and allowed for the submission, acceptance and presentation of a

scientific paper on an international conference in Shangai, ”2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI)”, on the 17th of November 2022.

To all those I have listed my sincere ”Thank you”.

Resumo

Este documento apresenta o trabalho realizado na tese de Mestrado em Engenharia de Telecomunicações e Informática e descreve o desenvolvimento, implementação e validação de um Sistema de Monitorização Remota da Saúde para Idosos.

Devido à crescente inovação tecnológica ao longo dos anos, a esperança média de vida dos seres humanos está a aumentar anualmente. Embora seja um excelente passo em frente para a humanidade, tem levado à população mais idosa a ser propensa a doenças e acidentes, tais como quedas. Neste trabalho, efectua-se um estudo sobre a literatura existente em sistemas não intrusivos de monitorização remota da saúde, com vista à concepção e implementação de um sistema IoT capaz de identificar quedas e monitorizar dados cardíacos. Foi concebida uma Revisão Sistemática da Literatura (SLR), tendo em conta literatura existente sobre sistemas de monitorização da saúde, algoritmos de detecção de quedas e IoT. A metodologia Design Science Research (DSR) foi utilizada para procurar melhorar os conhecimentos tecnológicos sobre o tema desta dissertação, através da criação de um artefacto inovador.

O sistema inclui um relógio inteligente (LILYGO T-WATCH-2020-V2), programável em C sob a IDE Arduino para detectar quedas e um dispositivo de monitorização fotopletismográfico (PPG) baseada num oxímetro Onyx 9560 Bluetooth, capaz de medir a percentagem de oxigénio no sangue (SpO2) e o ritmo cardíaco. Fornece ainda monitorização remota através de um website para visualizar dados em direto sobre a saúde do utilizador. O sistema foi testado em voluntários para mostrar a eficácia dos sistemas de monitorização remota da saúde em idosos.

Palavras-chave: Monitorização da Saúde, Deteção de Quedas, Fotopletismografia, Investigação Científica de Design, Revisão Sistemática da Literatura

Abstract

This document presents the work done in the Master's thesis in Telecommunications and Computer Engineering and describes the development, implementation and subsequent of a Remote Health Monitoring System for the Elderly based on Mobile Computing and IoT.

Due to increasing technological innovation over the last decades, the average life expectancy of humans is increasing year-by-year. Although this is an excellent step forward for humanity, it has led older population to being more prone to illness and accidents such as falls. In this work a study is made on the existing literature in non-intrusive remote health monitoring systems, towards the design and implementation of an IoT system capable of identifying falls and monitor cardiac data. A Systematic Literature Review (SLR) method was considered, taking into account the existing literature on remote health monitoring systems, fall detection algorithms and IoT. The Design Science Research (DSR) methodology was used to seek to enhance technology and science knowledge about this dissertation's topic, through the creation of an innovative artifact.

The system includes a smart watch (LILYGO T-WATCH-2020-V2), programmable in C under Arduino IDE to detect falls and a photoplethysmography monitoring unit (PPG) based on a Onyx 9560 Bluetooth oximeter, capable of measuring the user's blood oxygen percentage (SpO2) and heart rate, in real time. It also provides remote monitoring through a user-friendly website to visualize live data about the health status of the user. The system was tested in volunteers to show the effectiveness of remote health monitoring systems for the elderly population.

Keywords: Health Monitoring, Fall Detection, Photoplethysmography, Design Science Research, Systematic Literature Review

Contents

List of Figures

List of Tables

Code Listings

Glossary

IoT - Internet of Things

PPG - Photoplethysmography

BCG - Ballistocardiography

SpO2 - Oxygen saturation

ISCTE - Instituto Superior de Ciências do Trabalho e da Empresa

ISTA - Iscte School of Technology and Architecture

HRV - Heart Rate Variance

MQTT - Message Queuing Telemetry Transport

SLR - Systematic Literature Review

DSR - Design Science Research

IC - Inclusive Criteria

EC - Exclusive Criteria

CC - Cloud Computing

FC - Fog Computing

EC - Edge Computing

SaaS - Software-as-a-service

IaaS - Infrastructure-as-a-service

PaaS - Platform-as-a-service

MEC - Mobile Edge Computing

MCC - Mobile Cloud Computing

BI - Business Intelligence

ADL - Activities of Daily Living

HTML - HyperText Markup Language

PHP - Personal Home Page

a_{th} - Acceleration threshold

LVGL - Light and Versatile Graphics Library

BMI - Body Mass Index

CHAPTER 1

Introduction

This chapter aims to present the dissertation topic, starting with its motivation and framework. Then elaborate the respective research questions, as well as, define its main objectives and the research process applied in the preparation of the dissertation.

1.1. Motivation and Framework

With the significant increase in new technologies starting in the middle of the 20th century, the average life expectancy has increased exponentially, as shown in figure 1, reaching an impressive figure of 79 years in 2019 in Europe [1]. The increase in average life expectancy raises a new problem. The elderly population is increasingly exposed to disease and health threatening events such as falls. This is called frailty, which is

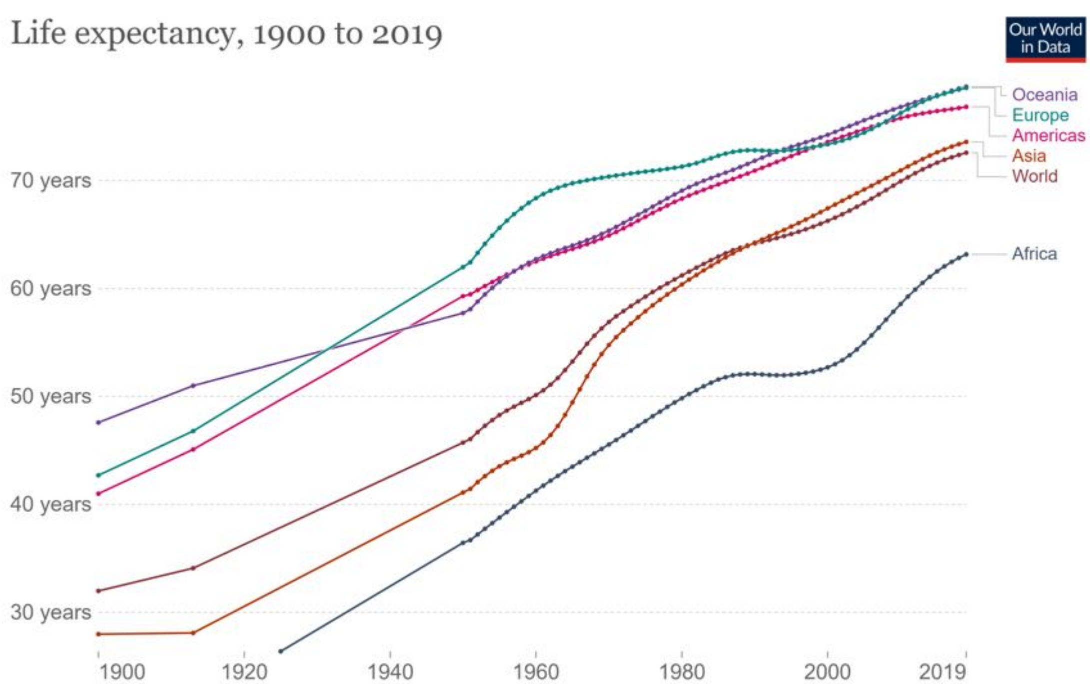


Figure 1. Average life expectancy through the years [1].

a measure of vulnerability that comes as a consequence of cumulative decline in many physiological systems during a lifetime [2]. The increased frailty of elderly people leads to a relatively poorer quality of life in terms of health, because despite living longer, they live with many more health problems.

Quality of life is a very subjective topic. Not only does it involve health, but is also influenced by family life, social and financial status, friends and many others. However, a study of older people over 80 years old, indicates that for 96% of the older people who indicated that they had a poor quality of life, health is a determining factor in their quality of life [3].

Therefore, it is necessary to innovate and create technologies that not only increase the average life expectancy but also provide a higher quality of life for the elderly by monitoring their health-related physiological data.

This innovation is based on the growing area of IoT where the number of IoT devices has been increasing over the years. The development of smart sensors, smart devices, advanced lightweight communication protocols made the possibility of interconnecting medical things to monitor biomedical signals and diagnose the diseases of patients without human intervention and termed as Internet of Medical Things (IoMT) [4]. One projection made indicates a growth from 8 billion devices in 2017, to 20 billion devices in 2020 [5]. The applications for IoT are endless and exist in all areas of our daily lives, not only in healthcare. Some examples of the areas are agriculture [6], automotive [7], telecommunication [8], among many others [9].

It is necessary to provide the older population with tools that not only allow them to live longer, but also to live with better quality of life, detecting unexpected accidents and monitoring health related data that can anticipate diseases. These tools must be based on IoT because it is the present and the future of technology, and thus improve the living conditions of the elderly, health wise.

1.2. Research Questions

Within the scope of the topic under study, the research questions that motivate the analysis and subsequent response are the following:

(1) How can a non-intrusive health remote monitoring system influence the health of an older patient?
(2) How does such system improve the quality of life of the elderly?
(3) How to design such system so that the response to an emergency is robust and effective?

With these research questions, this work aims to innovate in remote health monitoring, especially in the area of elderly health.

This work also defined a set of hypothesis to be proposed to reply to each research question. Each hypothesis comprises a prediction about how two or more variables will interact. It's like making an educated guess about what will happen in an experiment [10].

- **Hypothesis 1:** By being a system capable of detecting falls, monitor cardiac data and able to send warnings to caregivers or even in extreme cases to authorities in cases of medical emergencies, the patient will directly benefit from better health over time and quick responses in cases of distress.
- **Hypothesis 2:** As seen in the previous subsections, health is the determining factor in quality of life, for the older population. By improving their health conditions, their quality of life is directly improving as well.
- **Hypothesis 3:** The system needs to be user-friendly to be perceptible to the average caretaker, capable of rapid response to emergencies by having low latency levels and needs to be a robust system in case of power failures, hardware failures, among other unforeseen events.

1.3. Objectives

The goal of this dissertation is to create a non-intrusive remote health monitoring system, based on Internet of Things (IoT) sensors, capable of monitoring the living conditions of the elderly and in cases of life-threatening danger send warnings to the proper authorities, possibly saving lives. The diagram of the expected system architecture can be seen in figure 2.

This work will focus then, on creating a system capable of:

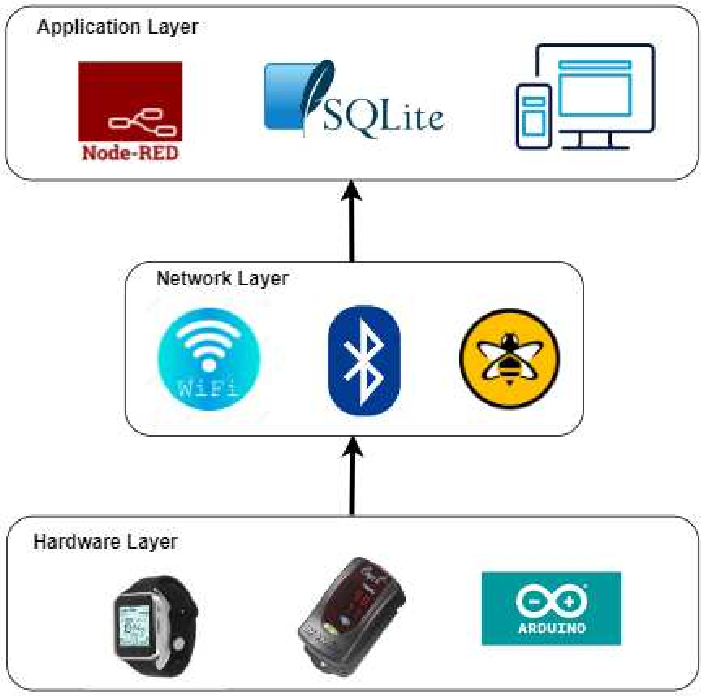


Figure 2. Expected IoT system diagram

- Identifying falling situations for the user by accessing the accelerometer sensor on the Lily-Go T-Watch-2020 V2 smart watch;
- Measuring SpO2 and heart rate data of the user in real time, using a Onyx 9560 oximeter, displaying the values on the Lily-Go T-Watch-2020 V2 smart watch via Bluetooth communication, through Node-Red;
- Locating the user through GPS location on the watch, in cases of medical emergency;
- Processing the data in the cloud, store the data in an SQLite database and provide a user-friendly visualization website;
- Sending warnings to health entities in the mentioned emergencies.

This work was carried out using the facilities of the ISCTE Campus and used hardware provided by the Iscte School of Technology and Architecture (ISTA), and Instituto de Telecomunicações IT-IUL. This work has as a differentiating factor from previous reported works, the fact that it includes not only the monitoring of cardiac factors but also the ability to identify falling situations. In addition, it has a visual interface to remotely monitor the health status of elderly users.

1.4. Research Process

In this dissertation work, the research process based on Design Science Research (DSR) was followed. DSR is a problem-solving paradigm that aims to improve human knowledge through the creation of novel products. Simply said, DSR aims to improve technology and science knowledge bases by developing new artifacts that solve problems while also improving the environment in which they are implemented [11].

The DSR process, as shown in figure 3 includes six steps [12]:

- Identification of the problem and motivation;
- Definition of objectives for a solution;
- Design and development;
- Demonstration;
- Evaluation;
- Communication;

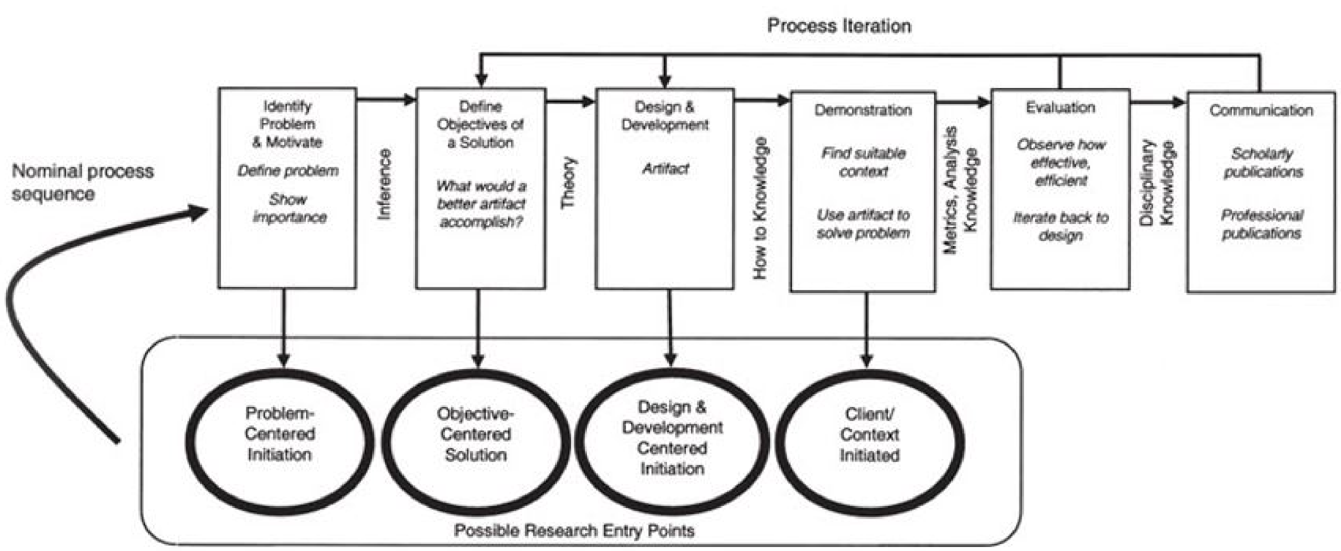


Figure 3. DSR Methodology Process Model (Adapted from Peffers et al. (2008)) [11]

The first step begins with identifying and defining the research problem as well as justifying the value of a solution. Then, defining the objective for that solution is the second step. After that, in the third step, the design and development of the artifact is done, by determining the artefact's desired functionality and its architecture. The fourth step consists in demonstrating the use of the artifact to solve instances of the research problem. Afterwards, in the fifth step, an evaluation of the solution is made, based on a comparison of the goals and the actual results obtained through the use of the artefact. Lastly, all aspects of the problem and the designed artifact are communicated to the relevant stakeholders, being other researchers or other relevant audiences, like professionals on the same area of work.

The DSR process also includes four different entry points [12]: **The problem centered approach**, starting in step one, for research where the concept came from observing the situation or through a report from a previous project that proposed future research. **The objective-centered solution**, starting in step two, mostly for research that needs to develop an artifact. **The design and development-centered approach**, starting in step three, being suitable for situations where the artefact already exists, but needs further developments. **The client/context initiated solution**, starting in step four, for research based on observing a practical solution that has worked.

In this work, to apply the DSR model, the problem centered approach was used, due to the fact that the addressed problem is already known. This approach can be seen in figure 4.

The first step of the DSR process, where a thorough knowledge of the problem at hand and the state of the solution is required, is represented in Chapter 2, literature review. The second step, is represented in section 1.3. In the third step, the design and functionalities of the system will be defined. In the fourth step, data collected by the system will be compared to data values deemed usual. The fifth step consist of conducting tests to evaluate the performance of the system as a solution to the problem found. In the sixth step the work is shared with the interested parts via a submission of a scientific article.

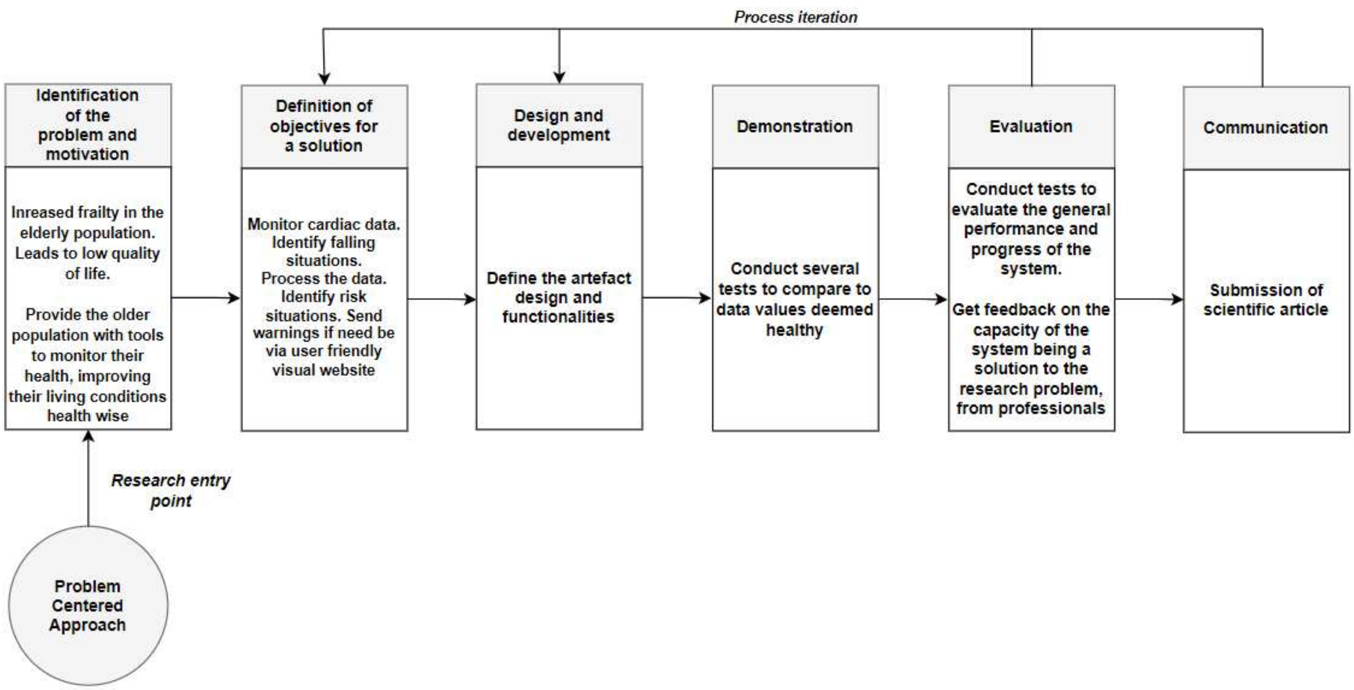


Figure 4. DSR Methodology adapted from figure 3

1.5. Dissertation Outline

After the introduction, the organization of the dissertation follows the following scheme:

- **Chapter 2** : Presents the state of the art and the research methodology used;
- **Chapter 3** : Defines the necessary requirements and the architecture of the system, as well as containing the complete system implementation;
- **Chapter 4** : Presents, analyses and discusses the results and limitations of the remote health monitoring system;
- **Chapter 5** : Provides a conclusion to the work. Presents possibilities for future work in the area of the dissertation.

CHAPTER 2

State of The Art

This chapter presents the review of the literature that was beneficial to this dissertation. It will provide a good technical knowledge and starting bases for the realisation of the necessary work.

For a focused and organised research, a systematic review of the related work encountered is defined. In the literature review, the focus will be on reviewing existing literature on remote health monitoring systems. Reviewing how IoT data is processed, to define the best strategy to use to process the data captured by the IoT sensors on the system, by looking at the various computing options used in other similar IoT systems. This way there is a clearer idea where it makes most sense to process the data generated. Closer to where the data is generated to improve performance, for instance, or other options.

Finally, reviewing how the wireless communication is handled in other existing IoT systems is needed, ensuring the correct functioning of the communication between the sensors and the system, analysing the Message Queuing Telemetry Transport (MQTT) protocol.

2.1. Systematic Review

To conduct a systematic review, to analyse and understand the related work already done in the field of this dissertation, the Systematic Literature Review (SLR) method was used. In health care, systematic reviews and meta-analyses have become increasingly relevant [13]. Therefore, to gain a better insight into the area of remote health monitoring, this research method was conducted on the available literature.

To conduct an SLR, it is necessary to define a main search database. The database for this work was the Scopus tool. In the need, to acquire additional information for

Including	Excluding	Number of related papers
All open access	Not being open access	610
From 2014 onwards	Older than 2014	511
In the area of Computer Science, Engineering and Medicine	Not in the area of Computer Science, Engineering and Medicine	415
Written in English and Portuguese	Written in other than Portuguese or English	413
Document type: Article	Not an Article	298

Table 1. Inclusion and Exclusion criteria defined in the search for related work.

the literature review, a secondary database, provided by Google, called Google Scholar, was used.

The search in the Scopus tool was performed through a query, while in the Google Scholar tool the search was only performed when necessary, through keywords.The query run was initially: "(health AND remote AND monitor AND systems)". This query proved to be too generic as the number of related papers found was 2234. With such a high number, it is impossible to make a SLR of this magnitude individually.

So, to highlight the most relevant works among the enormous amount of results obtained, some Inclusive Criteria (IC) and Exclusive Criteria (EC) were defined. These criteria are found in Table 1.

Even with the ICs and ECs defined, we still have a substantially large number of related articles. So, we apply an additional exclusive criteria that removes articles related only to medicine. This leaves us with 137 articles. The next step to slim down this number is to skim through the abstract of the remaining articles and select those that appear to be more relevant to the work to be carried out.

After this step there were 55 articles left, which were then analysed for the next subsection, in which we will review how monitoring systems are developed.

2.2. Related Work

2.2.1. Remote Patient Monitoring by Diseases

To analyse some of the existing remote patient monitoring systems, it is necessary to take into account whether or not the wearable devices used in this systems are of the intrusive type. In this work, the focus falls only on the non-intrusive sensors because these are the ones used in the prototype of this work.

In the work carried out by [14], non-intrusive sensing methods are approached. One of the methods approached is the PPG Sensing Method.

Many vital signs, such as heart rate, respiration rate and blood pressure, have been measured using PPG sensing, which uses a light source to emit light into tissue and a photo-detector to gather light reflected from or transmitted through the tissue. Within each cardiac cycle, the signal recorded by this approach indicates the pulsatile blood volume variations of peripheral microvasculature generated by pressure pulse. The sensor unit is traditionally in direct contact with the skin [14].

Recent research shows that there are already several devices that measure PPG in various different areas of the body, such as a PPG ring [15], capable of monitoring a patient's heart rate, oxygen saturation, and heart rate variability. Another example is PPG measuring with eyeglasses [16] through measuring blood volume variations on the nose. In another work done by [17], PPG measurements are done in the ear lobe, to monitor disease outbreak to prevent traffic accidents, while driving.

Another non-intrusive method to be analysed is the BCG sensing method. This method consists on obtaining a representation of the heart beat-induced repetitive movements of the human body, occurring due to acceleration of blood as it is ejected and moved in the large vessels [18]. The sensor captures these vibrations of the blood pumping and thus monitors the cardiac activity. It can be measured non-invasively without the use of electrodes, textiles, or anything else on the subject's body. As a result, BCG is well adapted to continuously monitoring cardiopulmonary function at night during sleep [19].

Several structures are already in place prepared to capture BCG data. The work carried out by [18], uses a chair as a sensor structure to measure BCG data. Another example is sleep monitoring using BCG sensing in beds like the work done by [20].

In the work carried out by [19], in which BCG was measured with four bed sensors, each one at each corner of the bed, it is observed that pre-processing of the raw BCG data is necessary to normalise the obtained values.

To remove the low frequency respiratory components, the data is filtered using a fourth order Butterworth band-pass filter. Then, output was normalized as max value was 0.5 and min value was -0.5. In figure 5(a) we observe the extracted typical wave showing the rough largest peaks detected in intervals of approximately 0.7 seconds. So, the next step is to observe segments for each peak in that interval. That is achieved by defining 0.4s before and 0.4s after each peak, figure 5(b).

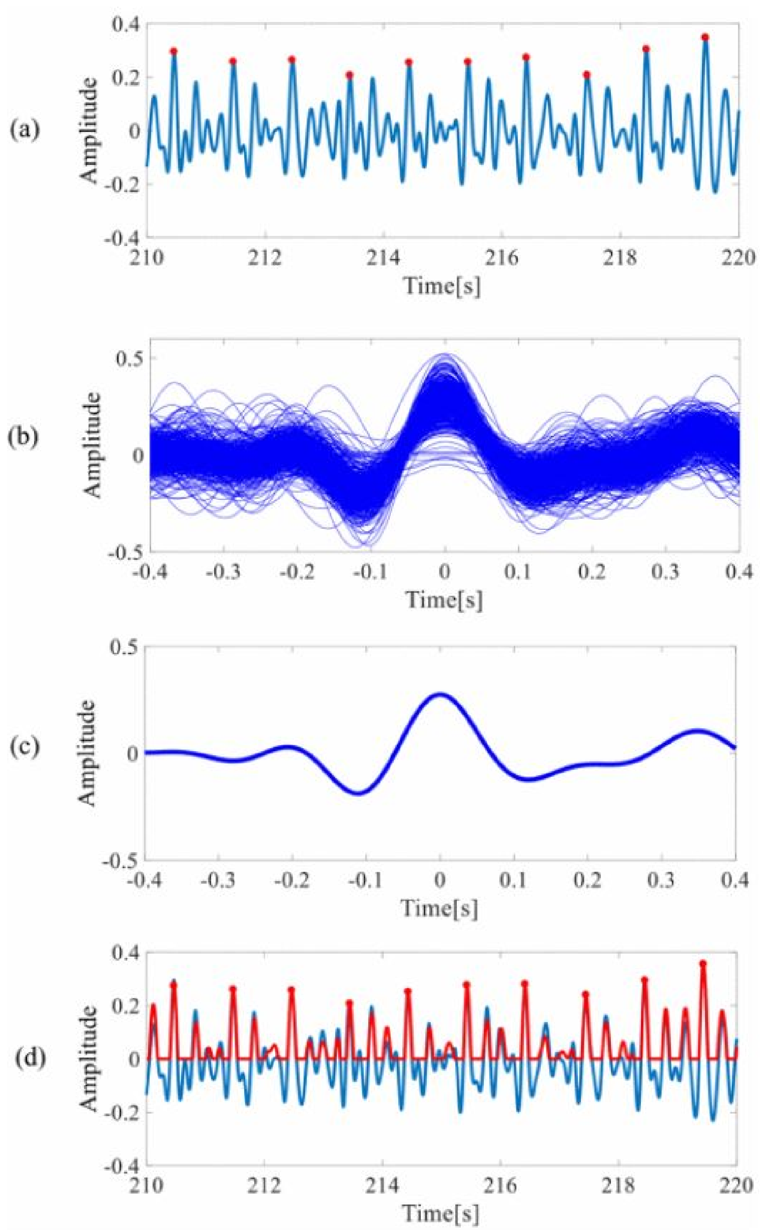


Figure 5. BCG extraction analysis [19]

In the figure small differences between the different peaks are observable, so the next step to normalise the BCG waveform, is to average all the peaks, thus reducing the effect of these differences, figure 5(c).

After that, correlations between typical wave of BCG and pre-processed BCG were calculated, replacing negative values as zero. It was found again that the interval between peaks was around 0.7 seconds. These peaks were defined as heart beats, as seen in figure 5(d) as red loops marked on the red curve line.

To evaluate the accuracy of the measurements, the beat-to-beat intervals from defined peaks were calculated and compared to blood pressure data, considered correct, for the same intervals. The results are found in figure 6.

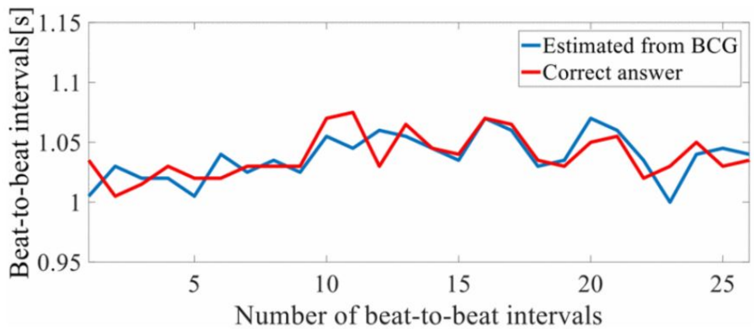


Figure 6. Beat-to-beat intervals calculated from blood pressure (red line) and BCG (blue line) [19]

To estimate the accuracy of BCG measurements, the precision was calculated, using the following equation:

$$Precision = \frac{correct}{correct + incorrect}, \tag{2.1}$$

If the calculated beat-to-beat interval slips over 30 (ms) from the correct answer, that interval was defined as incorrect.

The results showed that it was extremely difficult to calculate Heart Rate Variability (HRV) based on the BCG measured on users lying on their sides. They showed that 71.5% of beat-to-beat intervals can be detected by the algorithm designed in this work, for the average signal of the two top of the bed sensors. Finally, it was confirmed that the individual differences of each user present in the experiment, were the biggest factor for accurate estimations.

2.2.2. Remote Monitoring Patients in Falling Situations

Fall detection systems are an important kind of monitoring devices, particularly for the elderly population, where falls can have serious health effects. Main challenge fall detection systems face is differentiating a fall from activities of daily living [21]. Many of these systems are based on the accelerometer sensor. So, algorithms are needed to distinguish a fall from, for example, picking up an object from the ground.

The work carried out by [22] concludes that current falls detection research is hampered by a number of flaws. The fact that simulated fall scenarios are only performed by young healthy volunteers, the fact that simulated falls do not always represent real fall situations and the fact that falls are usually performed onto thick mats that provide a cushioning effect and change the characteristics of the fall impact from that of a real fall are some of these shortcomings.

In the work carried out by [23], the fall detection system, uses a threshold based fall detection algorithm. The first feature to define the event of a fall, is defining the threshold of the sum acceleration and rotation angle information. When an actual fall occurs, the impact of the human body with the ground produces a visible peak in the cumulative acceleration, which has a magnitude of:

$$|a| = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{2.2}$$

Where a_x, a_y and a_z represent each axis acceleration of the accelerometer. This magnitude is the first step to distinguish high intensity movements from others. However, high intensity activities, like jumping, also produce peak values, which mean that additional detection features are required.

The second feature, is an angle calculated based on acceleration measurements. By separating the gravity components before and after human's fall, then it is possible to calculate the rotation angle of accelerometer coordinate in 3D space, which is also equivalent to the rotation angle of the gravity vector relative to the fixed coordinate. The rotation of gravity vector in fixed coordinate is shown in figure 7.

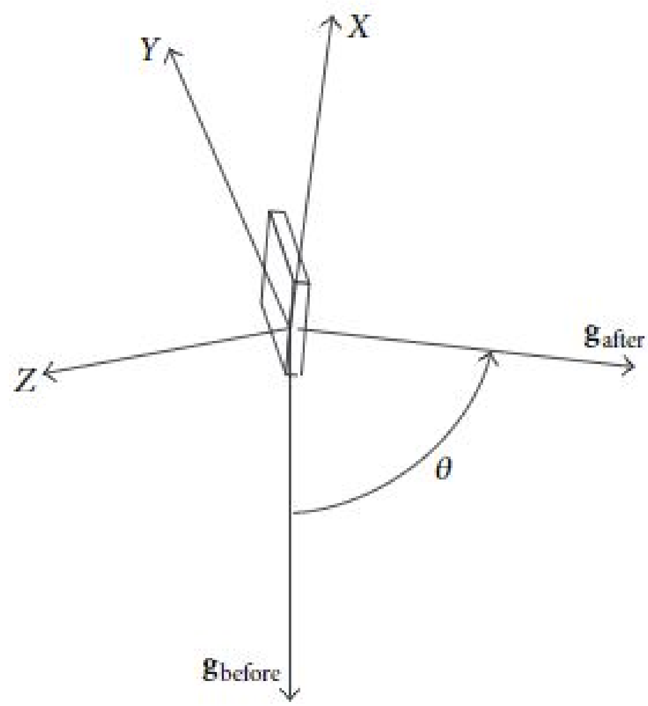


Figure 7. Rotation of gravity relative to fixed coordinate [23]

The rotation angle θ is obtained using the quaternion rotation method. When an object is falling, θ approaches 90°. In this work, the wearable device is mounted on human's waist and the device records g_{before}. The fall detection state machine is presented in figure 8.

$a_{threshold}$ means the threshold of sum acceleration magnitude and $t_{threshold}$ means the threshold of oscillation time duration after the break of $a_{threshold}$.

The sum acceleration $|a|$ will be determined once measurements in three separate axes have been taken. When a real fall happens, sum acceleration will reach peak value of $|a| \geq a_{threshold}$.

When the act of falling is over and the elderly is already lying on the floor, $|a| \approx |g|$. Then the acceleration $|a|$ is recorded as g_{after} and the quaternion angle of rotation θ is calculated, based on the information g_{after} and g_{before}. Lastly, it is considered a fall if $\theta = 90°$.

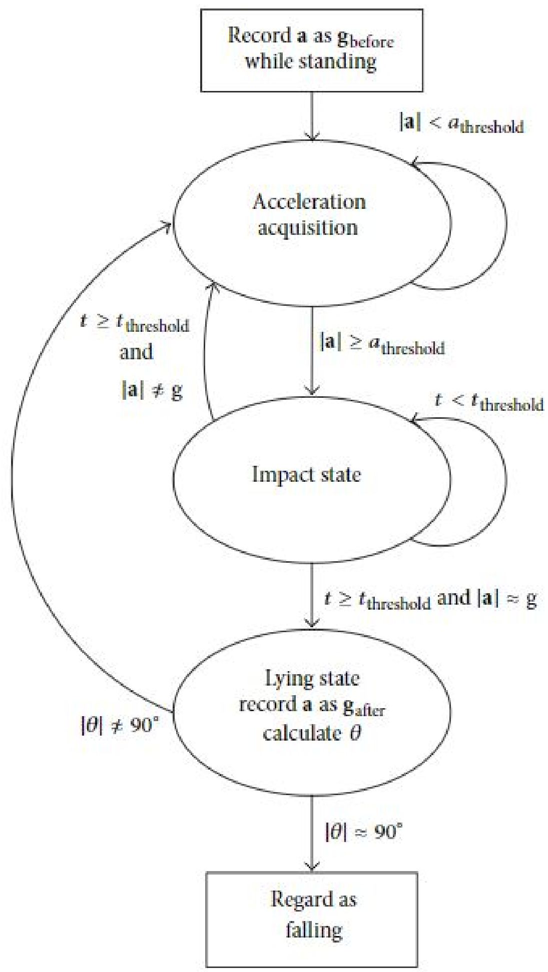


Figure 8. Fall detection algorithm state machine [23]

This work concludes that normal activity of resting also has similar rotation as falling and it may trigger false falling alarms. The correct definition of the paramater $a_{threshold}$, is the most important factor to distinguish falling from lying down. It also concludes that it is important to experiment with a group of people who are more differentiated in terms of age, gender and weight, in order to improve the reliability and robustness of the threshold.

2.3. Computing Data

With the exponential growth in the number of IoT systems, such as smart cities, smart homes, among others, comes the need to increase the available resources, such as increasing network bandwidth and decrease latency in data transmission [24].

In cases of emergency, it is imperative to have no delay in data transmission. Otherwise the possibility of irreversible damage can become a reality. In this work, decreasing latency as much as possible can be as critical as saving a user's life.

Affected systems with insufficient bandwidth limitations, make existing networks unable to send all data from IoT devices in a timely manner to the cloud. This can lead to information being lost, leading to inaccurate decision-making.

To address the problems of centralised computing systems and to mitigate the demands of IoT systems explained above, new computing strategies are emerging such as: Cloud Computing (CC), Edge Computing(EC), Fog Computing(FC) [24]. The architecture for these new processing strategies in IoT systems can be seen in figure 9. In this work, this architecture will be the basis for the way we process the data coming from the sensors.

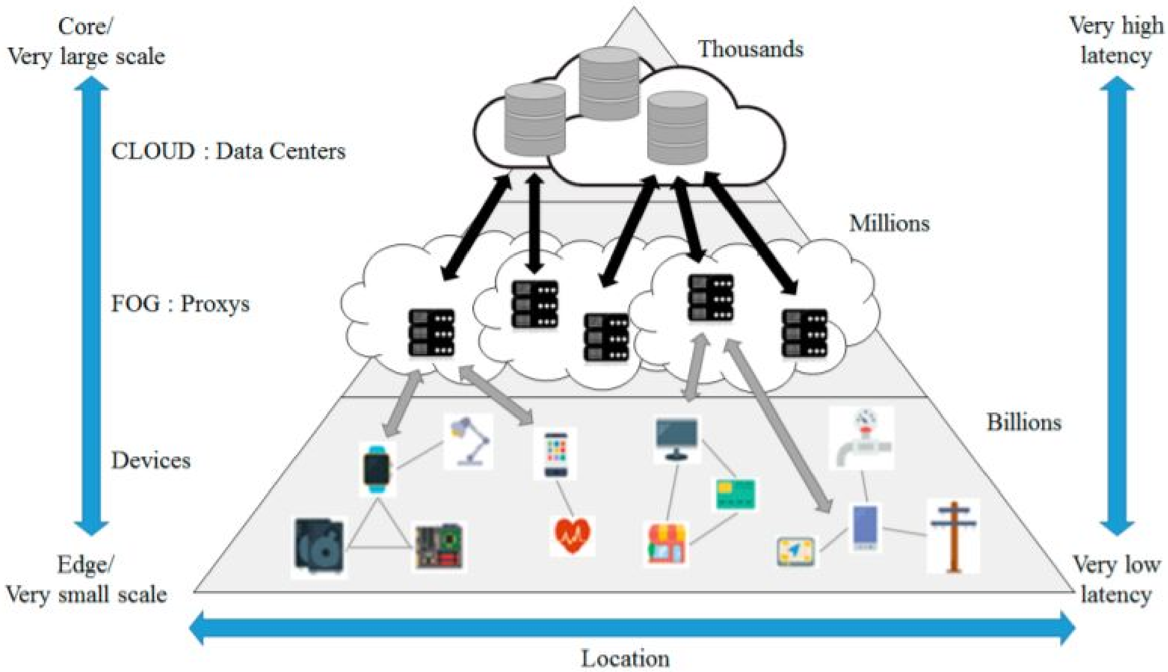


Figure 9. IoT system architecture [25]

2.3.1. Cloud Computing

Cloud Computing has revolutionised the way we process IoT data by providing various services over the internet. It allows you to store data in remote databases in the cloud, rather than in local storage. This makes it exponentially easier to access data, as only an internet connection is required.

There are 3 main cloud computing services [26]:

- **Software-as-a-service (SaaS)**: Provides the user with software running on infrastructures owned by the cloud provider. These applications are designed to be accessed by several users at the same time, via an internet connection. Examples of SaaS include Google Mail, Google Docs, and so forth [27];
- **Platform-as-a-service (PaaS)**: Development service offered that users can access via the internet. Instead of using software provided, users use this service to develop software of cloud apps and applications, that are delivered via the Internet (PaaS Cloud) [27]. Some examples of PaaS are Google AppEngine [27];

- **Infrastructure-as-a-service (IaaS)**: This is the delivery of servers, storage, network and operating system, as a service. The user has access to a virtual computer, storage, network infrastructure, computing resources for deploying and running software. Examples of IaaS are Amazon's EC2, IBM Cloud and Microsoft Azure [27].

There are 5 main actors present in the CC architecture [28]: The Cloud Provider, which is the entity providing the service. The Cloud Consumer, being the entity using the service. The Cloud Broker, that mediates the providers and the consumers, and manages performance and delivery of the services. The Cloud Carrier, serving as middle ground between consumers and providers, supplying connectivity between them. Lastly, the Cloud Auditor, that conducts independent assessments of the Cloud infrastructure in areas like security, performance and others.

Therefore, CC has the advantage of lowering costs, increasing the efficiency and robustness of IoT systems, unlimited storage capacity and easy accessibility.

2.3.2. Edge Computing

Edge computing is a technological innovation that enables the services and utilities of Cloud computing to be brought closer to IoT devices at the periphery of the network, where the data is generated [29]. This, guarantees fast processing and low levels of latency.

According to the study described in [30], for mobile devices, Mobile Edge Computing (MEC), achieves lower latency, consumes less power saving more energy and enhances privacy and security for mobile applications, in comparison with Mobile Cloud Computing (MCC).

EC is present in many areas [31], such as:

- **Manufacturing**, to monitor processes, apply machine learning and perform real-time analysis to improve product quality and product and detect manufacturing problems in the product;
- **Farming**. Using sensors to monitor fertilizer density and water use as well as optimize harvesting, guaranteeing that crops are harvested in peak condition;

- **Network optimization**, by measuring network performance for users throughout the internet and using analytics to discover the most reliable, low-latency network path for each user's data;
- **Healthcare**. Due to increased patient data, a growing need arises to manage, process and store the data. Peripheral computing helps here by applying machine learning and automation to access data. It assists doctors in identifying data that requires immediate attention in order to improve patient care and eliminate health incidents.

In this work, it is essential that the processing is optimised and that the network presents minimum latency levels due to the sensitivity of the information generated by the sensors. It is important to act quickly in cases of emergency.

2.3.3. Fog Computing

Fog computing was a term coined by Cisco, to enable applications on billions of connected devices, already connected in the Internet of Things (IoT), to run directly at the network edge [32].

Fc's objectives are very similar to Ec's in many respects. Being, moving cloud computing's storage, networking, and computing capabilities to the edge of networks, offloading cloud data centers and lowering service latency for end users [33].

The key distinction between EC and FC is where the processing takes place. EC occurs on the devices where the sensors are installed, or at a gateway physically close to the sensors. As a result, FC can add another processing layer between the cloud and IoT devices. The first benefit is increased data traffic efficiency and a reduction in latency. However, the biggest advantage in adding the extra layer is relieving cloud storage, since the cloud would now only store and process relevant data [34].

2.4. MQTT Protocol

The MQTT protocol, released by IBM in 1999, is the de-facto standard for IoT messaging [35]. MQTT publish/subscribe protocol provides a scalable and reliable way to connect devices over the Internet. Today, MQTT is used by many companies to connect millions of devices to the Internet, like Amazon Web Services, McAfee, Red Hat and Cisco [36].

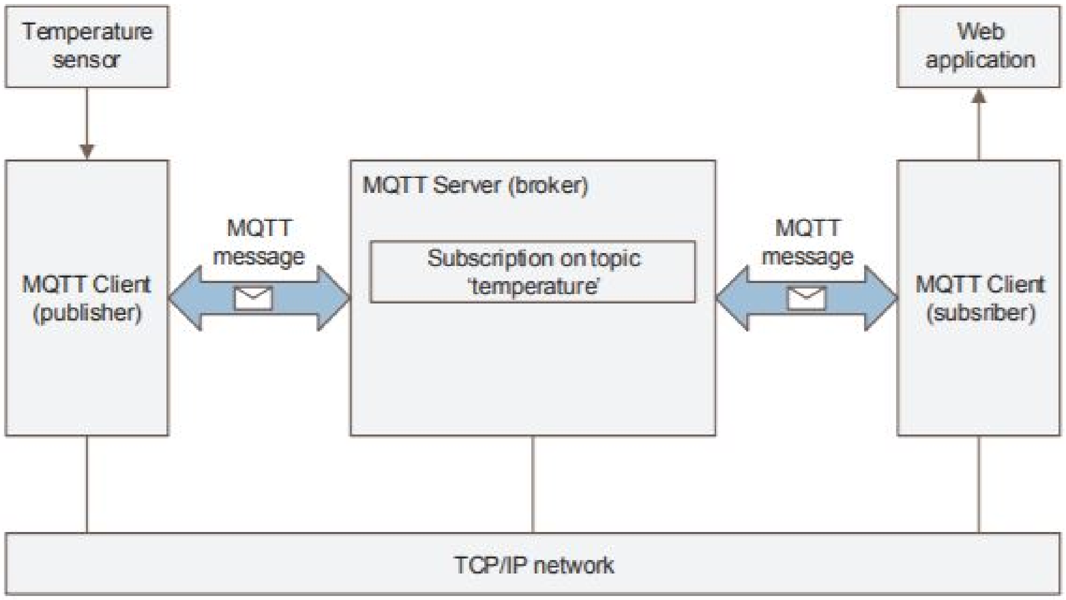


Figure 10. MQTT protocol model [38]

The protocol architecture contains 3 main actors [37]:

- **Publisher:** MQTT client that publishes a message to an MQTT broker defining a topic that the broker can use to forward the message to interested clients;
- **Broker:** MQTT broker receives messages published by clients, filters the messages by topic, and distributes them to subscribers;
- **Subscriber:** MQTT client that sends a SUBSCRIBE message to the MQTT broker, listening to messages with the subscribed topic.

The protocol model is shown in figure 10. Using the MQTT protocol to handle the communications of the system, we increase scalability, reduce bandwidth consumption and distribute information more efficiently.

CHAPTER 3

System Description

To meet the objective of creating a system capable of identifying possible fall situations, and through accessing the accelerometer sensor of the smartwatch, firmware was developed to detect falls, via an algorithm designed in the Arduino IDE using C programming language.

To meet the objective of capturing cardiac data, the Onyx 9560 oximeter was used. Through Bluetooth connection, the system can gather the data and provide the system with a visualization of the same.

To connect the different devices and stakeholders of the system, we use the Node-RED server.

To visualize the data a public website was created.

Figure 11 shows the final architecture of the system.

3.1. Requirements

This work focused on developing a system to be used for the majority of the day. The longer it is in use, the more effective the user's health monitoring will be. So, to meet all the intended objectives,the requirements were identified.

- **Many of the health monitoring systems currently in use are intrusive and affect the normal course of a user's daily activity for long periods of time.**
  - Requirement: Create a remote and non-intrusive health monitoring system. Remote, so that users do not have to travel to the providers of these services. Non-intrusive, so as not to disrupt users' normal daily lives.

- **Many health monitoring systems are not prepared to be used autonomously by the user, requiring the help of health professionals to be used.**

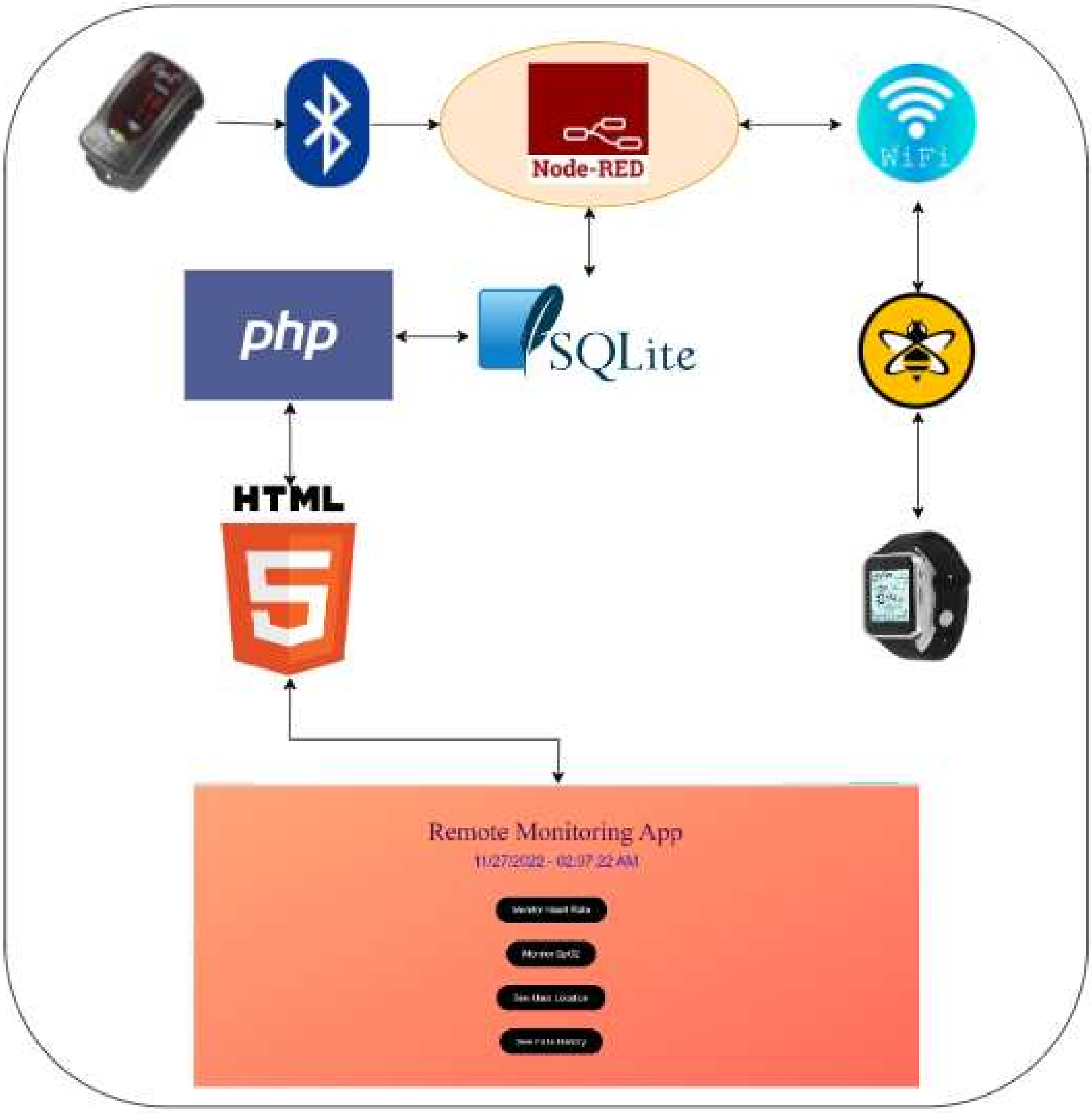


Figure 11. Final System Architecture

- Requirement: The system must be user-friendly for all the considered users. By only having to wear the watch and place the oximeter on the finger when needed, it is guaranteed that the system does not need professional supervision to be useful and to function.

- **Some remote health monitoring systems are not prepared to issue alerts in cases of medical urgency. They only have the functionality to collect data and store it for future queries from health entities for example.**
  - Requirement: The system must be capable of not only capturing, storing and processing data, but also be able to issue alerts in cases where the captured data indicates that there is a medical urgency.

- **Some health monitoring systems do not include real-time analysis by the caregivers, which means that the analysis of the patient's health conditions cannot be done in real time.**
  - Requirement: The system must have the capacity to allow the remote visualisation in real time of the captured data, since they are cardiac data and the location of the user, which analysed in real time can be important in critical cases.

- **Many IoT-based monitoring systems suffer from hardware failures, due to the fact that they are sometimes very complex, with several possible points of failure.**
  - Requirement: The system needs hardware that is not only simple to use, but also robust to failure, thus ensuring the viability of the system in the long run.

- **Many health monitoring systems are one-dimensional. They monitor only one or two factors and are not prepared to integrate more monitoring elements.**
  - Requirement: It is necessary that the system is capable of being scaled in the future to allow the inclusion of new IoT sensors. This will ensure the interoperability and future viability of the system, allowing for the constant evolution and improvement of remote user health monitoring.

3.2. Hardware

The hardware used in this system was the LILYGO T-WATCH-2020 V2 and the Onyx 9560 oximeter. This hardware was chosen in order to meet the requirements indicated above. Remote, non-intrusive and high-performance hardware was given priority.

3.2.0.1. ***LILYGO T-WATCH-2020 V2:***

The LILYGO T-WATCH-2020 V2, seen in figure 12, was design to be interacted with, networked, programmed, and worn. It incorporates an ESP32 microcontroller with integrated Wi-Fi and Bluetooth, which is convenient to connect to the Internet and is simple to program and build.

Figure 12. LILYGO T-WATCH-2020 V2

The main library used to base the development of the firmware was the Xinyuan-LilyGO/TTGO_TWatch_Library. The user interface of the watch is programmed using the Light and Versatile Graphics Library (LVGL) library as a basis.

It's powered by a rechargeable lithium battery and has a 1.54 inch watch capacitive touch screen using OCA process.

Other features and components of this watch are the possibility to access a BMA423 accelerometer sensor which is a 12 bit, digital, triaxial acceleration sensor with intelligent on-chip motion-triggered interrupt features, a DRV260SL vibration motor and an AXP202 power management unit, designed to be a highly-integrated power system management that is optimized for applications requiring single-cell Li-battery (Li - Ion/Polymer) and multiple output DC-DC converters.

The watch provides as the user interface display, an LCD ST7789V which is a single-chip controller/driver with a graphic type TFT-LCD and a touchscreen function. A block diagram to demonstrate the clock specification can be seen in the figure 13.

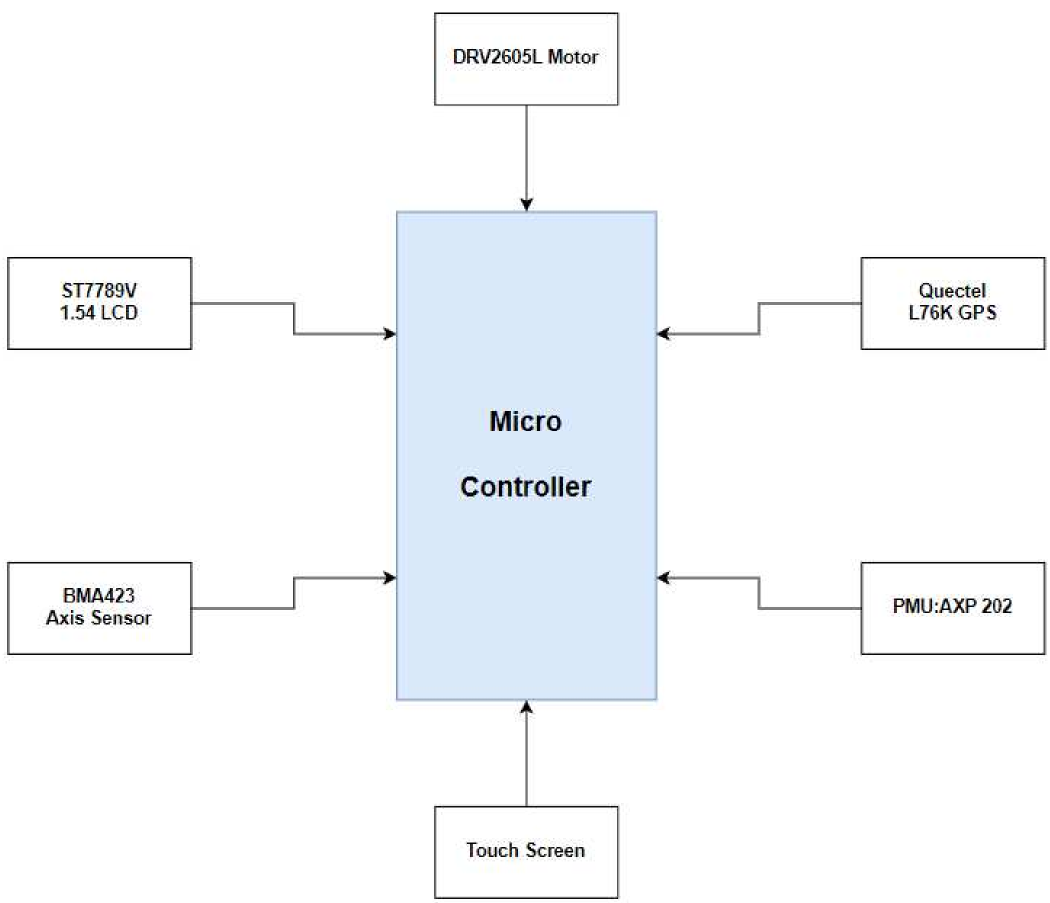


Figure 13. LILYGO T-WATCH-2020 V2 specifications - Block diagram

The LILYGO T-WATCH-2020 V2 introduces the innovation of the GPS functionality, by accessing a L76K GNSS sensor embedded in the watch, and the ability to store data on a memory card.

3.2.0.2. ***Onyx 9560 Oximeter:***

The Onyx 9560, seen in figure 14, is a pulse and blood oxygen saturation monitoring device, designed for individuals who desire a Bluetooth oximeter, allowing them to upload measurements to their phone or digital device. It also offers a range of up to 100 meters, and allows the user mobility during use, unlike many other oximeters.

Figure 14. Onyx 9560 Bluetooth oximeter

The oximeter provides excellent reliability and accuracy, allowing readings in various conditions, using Nonin PureSAT and PureLight technology.

Other features are easy carriability due to low weight, and an auto on/off - power saving feature.

3.3. Software

The software used in this work was the Arduino IDE to program the LILYGO T-WATCH-2020 V2, the Node-RED to connect all devices in our system, SQLite as a database, and both HTML and PHP to develop the website.

3.3.1. *Arduino IDE:*

The Arduino Integrated Development Environment - or Arduino Software (IDE) - is a free software that contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the microcontroller hardware to upload programs and communicate with them.

This is the software used to program the ESP32 microcontroller embedded in the LILYGO T-WATCH-2020 V2. The firmware developed to communicate between the microcontroller, the oximeter and the Node-RED server was successfully developed and is explained and demonstrated in the following section, System implementation.

3.3.2. *Node-RED:*

Node-RED is a programming tool for connecting hardware devices, APIs and online services. Node-RED is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware. This is where all system components are connected and communicate with each other and the database, via Wi-Fi using the MQTT protocol and via Bluetooth in the oximeter case.

The main reason for using Node-RED as the programming tool for connecting devices of the system is due to its flow-based programming simplicity and application flexibility.

The main functionalities of the Node-RED server in this system are:

- Implement an intermediary entity that enables MQTT clients to communicate - MQTT broker.
- Subscribe and publish to different topics, using the MQTT protocol.
- Extract, transform and load the data into an SQLite database.
- Manage the database.
- Provide a a monitoring dashboard for tracking the reliability of the system.

This functionalities can be seen in figure 15.

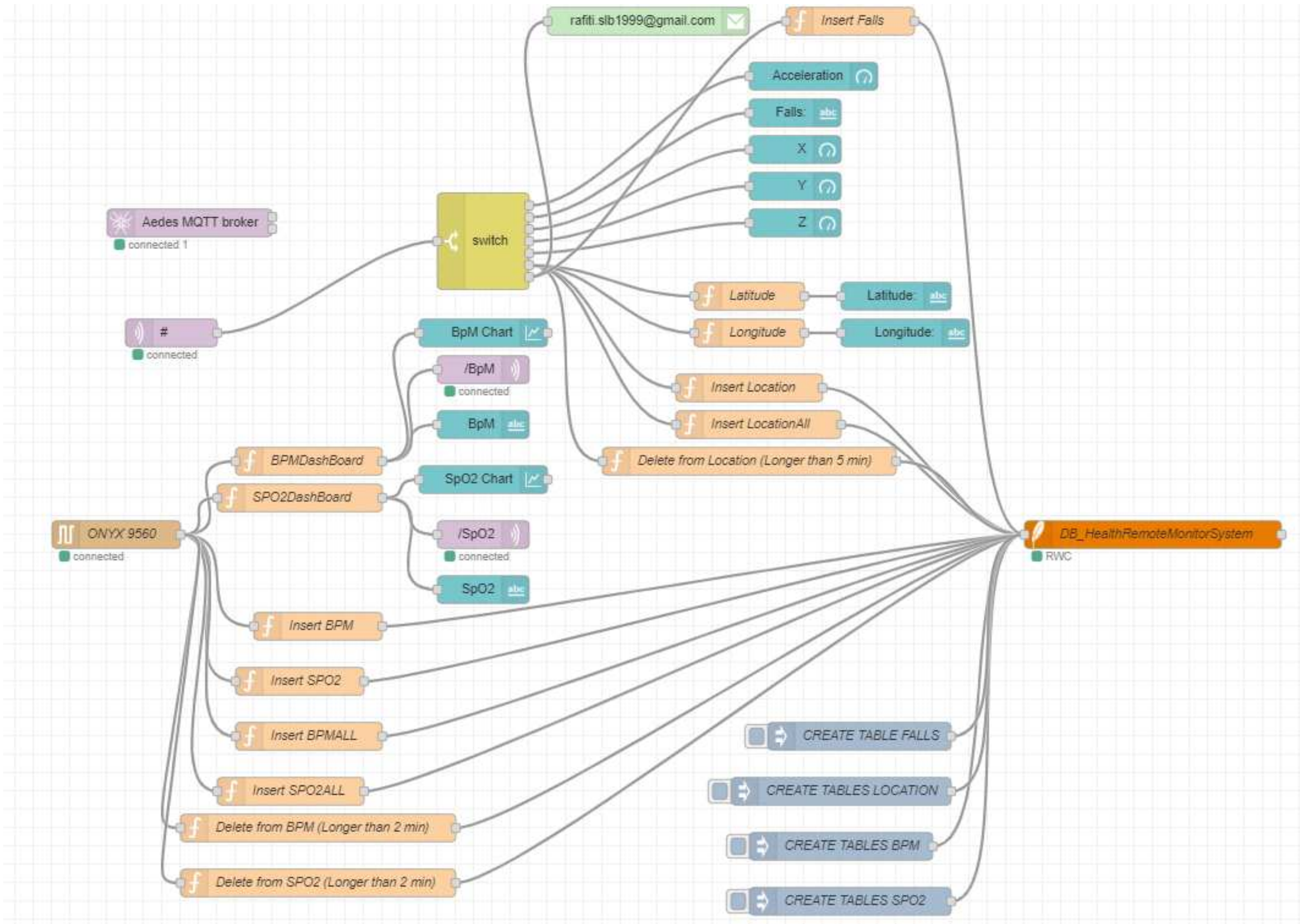


Figure 15. Node-RED flow

3.3.3. *SQLite:*

SQLite is an open-source SQL database that stores data to a text file on a device. One of SQLite's greatest advantages is that it can run nearly anywhere.

SQLite is a popular choice as the database to back small to medium-sized websites, because it requires no configuration and stores information in ordinary disk files.

The use of SQLite as the database for this system is justified because of its flexibility and the fact that it is better prepared to deal with a smaller flow of data.

The database model of this system can be seen in figure 16.

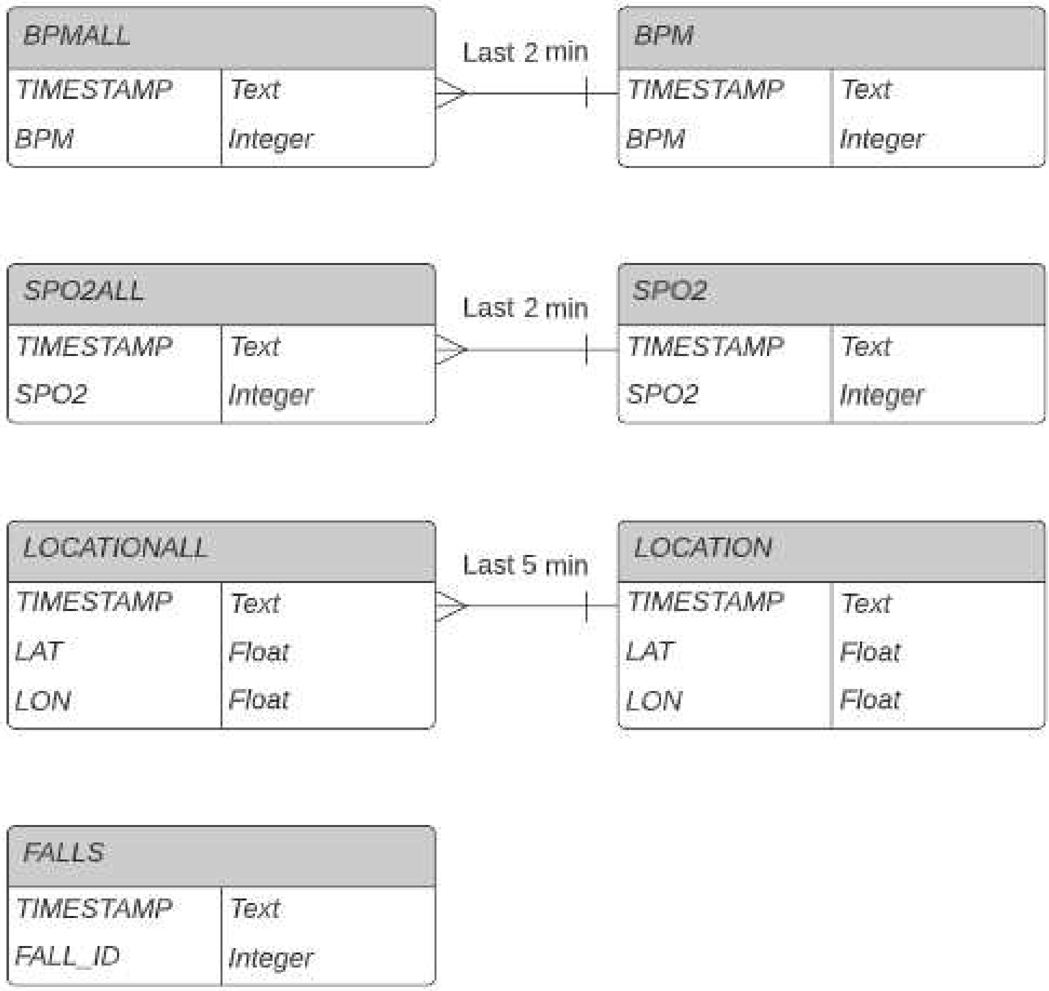


Figure 16. ”DB_HealthRemoteMonitorSystem” - SQLite database model

The final database of the system includes 7 tables. 4 tables for cardiac readings, 2 tables for GPS coordinate readings and a table for the fall alerts. The tables with ”ALL” in the name, have all the data records since the start-up of the system, e.g. the ”BPMALL” table. The ”BPM” table, only has the records from the last 2 minutes. This way, the system can present a much lighter table if cardiac data is to be monitored live.

3.3.4. *HTML:*

HTML stands for HyperText Markup Language. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes. HTML is used for web development, Internet navigation and web documentation.

3.3.5. *PHP:*

PHP: Hypertext Preprocessor (originally called Personal Home Page), is a popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP is also a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

A script is a set of programming instructions that are interpreted at runtime. Server-side scripts, such as PHP, are interpreted on the server, while client-side scripts are interpreted by the client application, browser for example, as is the case with JavaScript.

3.4. System Implementation

3.4.1. Fall detection algorithm

To design the fall detection algorithm, we need firmware that accesses and configures the BMA423 accelerometer embedded in the smart watch. This software has been developed:

```
void setup(void) //runs once at the beginning of the program.
{
  Serial.begin(115200); //Estabilishes serial communication at a baud rate of
    115200 bps
  WiFi.begin(ssid, pass); //Connects the micro controller to Wi-Fi
  client.begin(BROKER_IP, 1883, net); //Connects to the MQTT broker in port
    1883
```

```
client.onMessage(messageReceived); //Function called to enable receiving
 messages from subscribed MQTT topics
connect();

//Initialization of the Watch and its interface
  ttgo = TTGOClass::getWatch();
  ttgo->begin();
  ttgo->openBL();
  ttgo->tft->setTextFont(1);
  ttgo->tft->fillScreen(TFT_BLACK);
  (...)
  // Accelerometer parameter structure
  Acfg cfg;
  cfg.odr = BMA4_OUTPUT_DATA_RATE_100HZ;
  cfg.range = BMA4_ACCEL_RANGE_2G;
  cfg.bandwidth = BMA4_ACCEL_NORMAL_AVG4;
  cfg.perf_mode = BMA4_CONTINUOUS_MODE;
  // Configure the BMA423 accelerometer
  sensor->accelConfig(cfg);
  // Enable BMA423 accelerometer
  sensor->enableAccel();
  (...)
  }
```

Listing 1. Accelerometer specification

In the design of the fall detection algorithm, two decision factors were considered. Only if these two factors met the requirement, a fall was identified and thus caregivers received an alert from the system.

The first factor is comparing the magnitude of the acceleration ($|a|$) calculated with the linear acceleration per accelerometer axis (X, Y and Z), with a predefined maximum acceleration threshold (a_{th}) The magnitude of the acceleration was calculated using the following equation:

$$|a| = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{3.1}$$

Where a_x, a_y and a_z represent each axis acceleration of the accelerometer.

To obtain the mentioned acceleration values, software was defined to map the values of each axis acceleration to be between 0 and 100, where 0 is no acceleration, and 100 is maximum acceleration. Then, when the magnitude of acceleration is calculated (”acc_avg”) we compare it the magnitude registered in the moment before, using the absolute value generated by the comparison (”diff”). If this difference is bigger then the predefined acceleration threshold (a_{th}), the system activates the second factor to confirm a possible fall.

```
void loop() //Continuosly running
{
    (...)
     Accel acc;
    // Get acceleration data
    bool res = sensor->getAccel(acc);
    if (res == false) {
        Serial.println("getAccel FAIL");
    } else {
        // Show the data
        int new_x= acc.x; //X-Axis
        new_x = map(new_x, -1200, 1200, 0, 100);
        int new_y= acc.y; //Y-Axis
        new_y = map(new_y, -1200, 1200, 0, 100);
        int new_z= acc.z; //Z-Axis
        new_z = map(new_z, -1200, 1200, 0, 100);
        (...)
     //Magnitude of Acceleration
        int acc_avg = sqrt((new_x * new_x) + (new_y * new_y) + (new_z * new_z)
   );
```

```
        //Comparing the magnitude of acceleration with the one registered in the previous loop iteration
        int diff = abs(acc_avg-acc_1);
         //Storing the value for the next moment in time
        acc_1 = acc_avg;
        if( diff < 70){
          }else{
          //Activates the second factor - Threashold Surpassed
           ConfirmFall();
          }
    (...)
    }
```

Listing 2. First fall detection factor - Acceleration threshold exceeded

To reach optimal performance, a_{th} is tested in the next chapter of this work, to obtain the value with the highest hit percentage, in the detection of true falls.

The second factor helps a lot in detecting false positives and is implemented in the touchscreen interface of the smart watch. If the user does not click a button in the touchscreen interface, indicating that it was a false fall alarm in a period of time after the first factor, a possible fall is confirmed, and the alarm for the caregivers will be sent.

The system also provides a feature, during the period of time defined as 20 seconds, after the first factor occurs. This feature consists of implementing the vibration of the watch, indicating to the user that the acceleration limit has been exceeded. Thus, the user has the possibility, by means of a click, to annul the detection of the possible fall, thus being a false alarm. This implementation is possible by accessing the vibration motor embedded in the watch, DRV2605L Motor

When the fall is confirmed, the system sends through Wi-Fi, using the MQTT protocol, a message to Node-RED via the topic ”/Fall Alert”. Node-RED being a subscriber to the topic, receives the message and then sends the alarm to the caregivers via email.

This software implementation can be seen down below.

```
//Starts the vibration accessing the DRV Motor.
  void ConfirmFall(){
          drv->setWaveform(0, effect);  // play effect
    // Start the vibrating motor
            drv->go();
            delay(200);
        drv->setWaveform(1, 0);       // end waveform
    //Boolean indicator to indicate that the false alarm button hasn´t been clicked
        possibleFall=1;
    }
(...)
void loop()
{
    (...)
     // Function that listens to the user interface to check if the user presses the button
    lv_task_handler();
     //if the button hasn´t been pressed -> possibleFall = 1
     if(possibleFall == 1){
          iterador = iterador + 1;
          //if 20s of vibrating has passed:
          if(iterador >= 99){
              //Fall confirmation has occured
              nquedas = nquedas + 1;
              //PUBLISH TO TOPIC /FALL ALERT and Node-Red generates Alert
              client.publish("/Fall Alert!!", "O utilizador sofreu uma queda!");
              //resets the boolean indicator to stop the vibration
              possibleFall = 0;
          }
```

```
    }else{ // User pressed the button indicating a false alarm of fall
            iterador = 0;
    }
   (...)
}
```

Listing 3. Second fall detection factor - User confirmation

In the Node-RED server, an MQTT Broker is defined on a set port (1883), and an MQTT listener is also defined to subscribe to all messages sent from the smartwatch to different topics (#). This can be seen in figure 17. The figure also shows a "switch" node that for different topics, does different actions. For the messages published to the topics of Acceleration, X, Y, Z and Falls, the Node-RED dashboard was designed to show these values.

When a fall is confirmed and a message is published to the topic "/Fall Alert", Node-RED stores the alert and the time and day of the alert on the SQLite database. In addition sends the alarm to the caregivers via email.

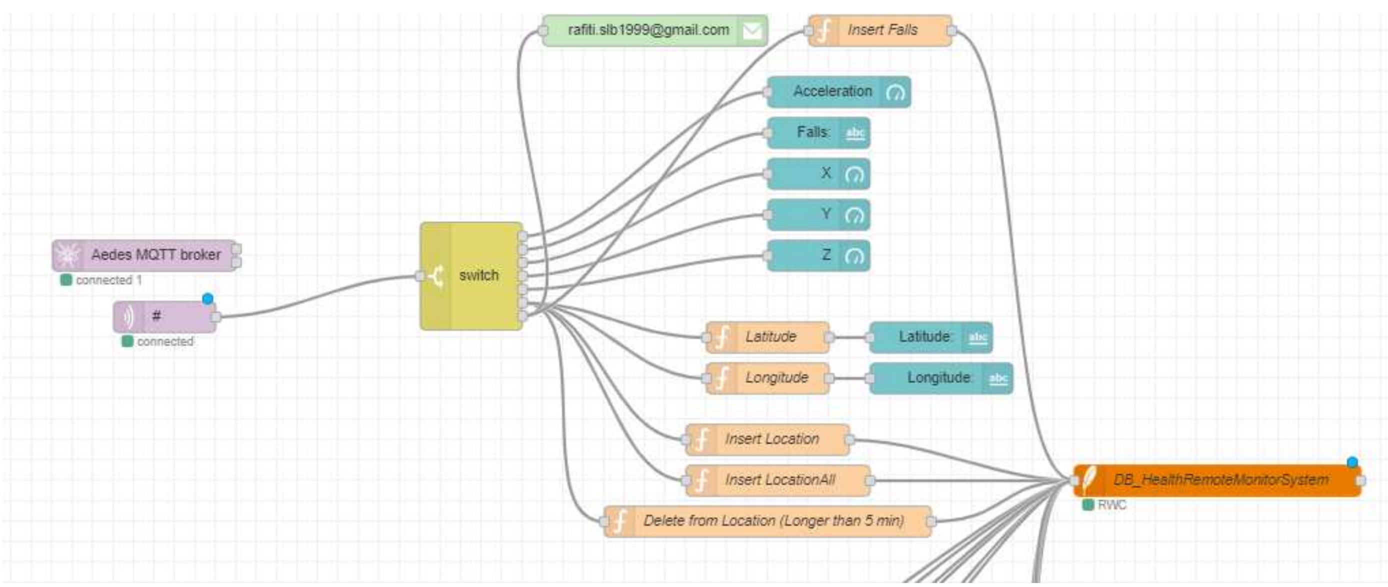


Figure 17. NodeRED - Fall detection algorithm

The system also has the possibility to display the real-time location of the user in case the healthcare entities which to locate the user of the system, in case of emergencies like a fall.

The smartwatch publishes the coordinates via MQTT protocol using the topic ”/coord”. The Node-Red subscribes this topic, and stores the coordinates in the ”LOCATION” and the ”LOCATIONALL” tables in the system’s SQLite database. This firmware is seen below:

```
static void display_info(void)
{
    static uint32_t updatePeriod;
    if (millis() - updatePeriod < 20000) { // Updates the GPS location every
    20 seconds
        return;
    }
    updatePeriod = millis();
    if (gps->location.isUpdated()) {
        //Gets the Latitude to 6 decimal places
         float latsmall= gps->location.lat();
         lat = String(latsmall,6);
        //Gets the Longitude to 6 decimal places
         float lngsmall= gps->location.lng();
         lng = String(lngsmall,6);
         //Sends the coordinates as (lat,lon) format
         String coord = ((String) lat ) + ',' + ((String) lng );
        //Publishes them to the /coord topic
         client.publish("/coord", coord); //PUBLISH TO TOPIC /coord
    }
    delay(20);
}
```

Listing 4. Get GPS coordinates and publish them on the MQTT topic - ”/coord”

An additional function is executed every time the server receives GPS coordinates. This function is used to delete the coordinate records from the LOCATION table, that have been captured longer than 5 minutes ago. These way we can have a table with the most recent locations of the user only. This is achieved by executing the following query:

```sql
DELETE FROM LOCATION
WHERE "TIMESTAMP" <= datetime('now', '-5 minutes', 'localtime');
```

Listing 5. SQL - Delete records from 'LOCATION' every 5 minutes

A final representation of the algorithm designed and implemented to detect falls can be seen in figure 18.

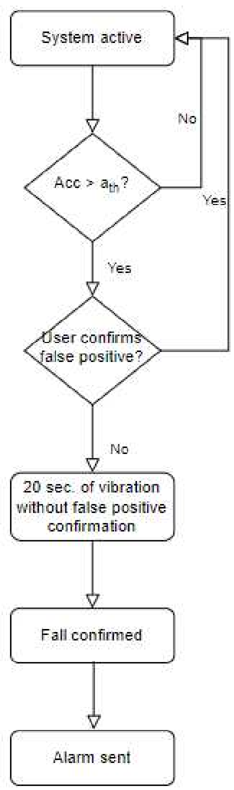


Figure 18. Fall detection algorithm - State diagram

3.4.2. Cardiac monitoring

The system also has the ability to collect heart rate (BpM) and oxygen saturation (SpO2) values using an Onix 9560 oximeter. The device captures and communicates data with the Node-RED platform via Bluetooth every second, as seen in figure 19, in the node 'ONYX 9560'.

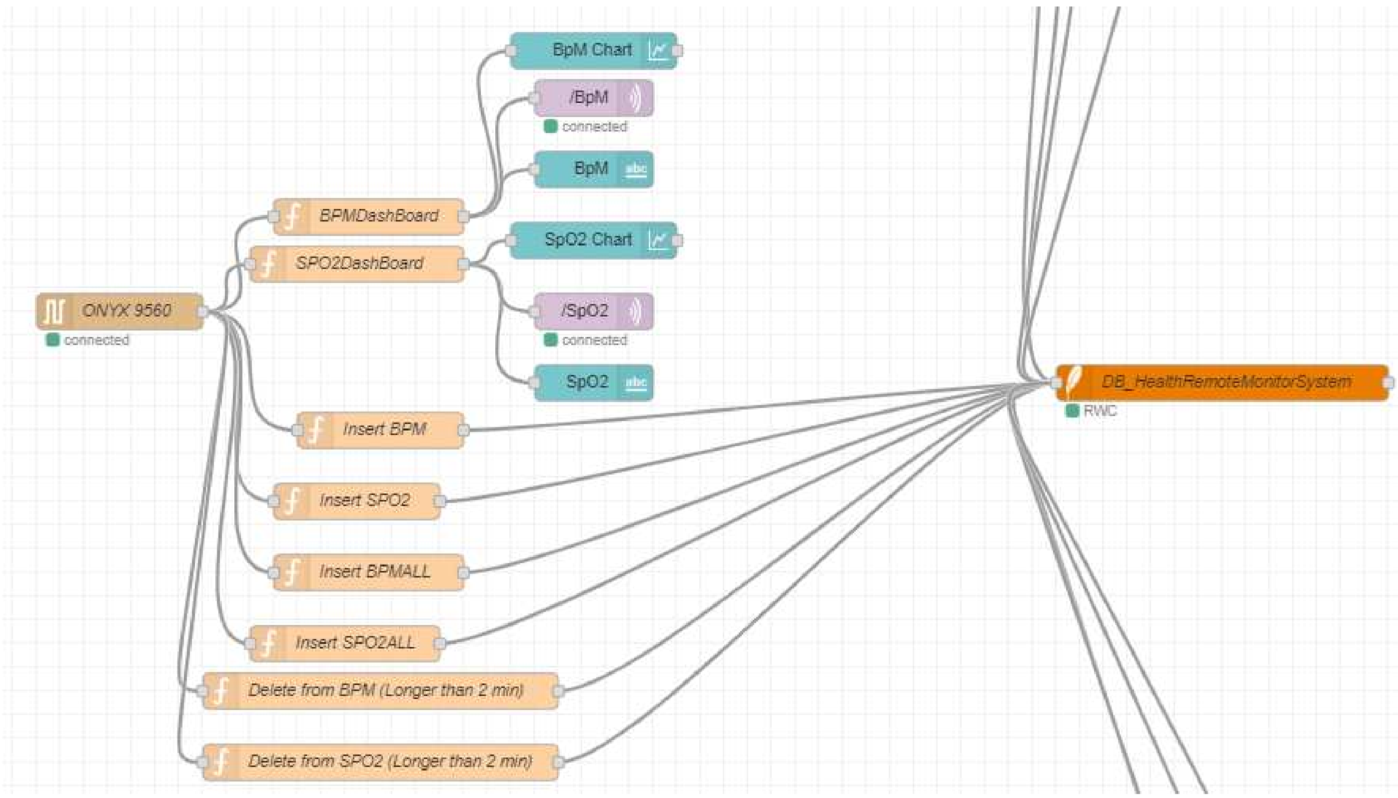


Figure 19. NodeRED - Cardiac Monitoring

Node-RED then forwards messages to the smartwatch via the MQTT protocol, through MQTT out nodes (purple) seen in the figure, publishing the heart rate values to the "/BpM" topic and publishing the blood oxygen saturation values to the "/Spo2" topic.

Node-RED stores the cardiac data into an SQLite database, through querys indicated in the function nodes (orange). Each type of data (BpM and SpO2) is stored in two different tables. One table has all the data collected from the beginning of the use of the system, coupled with the timestamp of when the data was collected.

The other table only has the data collected in the last two minutes, coupled with the timestamp of when the data was collected. This table was created to enable the real-time visualization of both the SpO2 and BpM, without having to see tons of data prior to when the visualization is occurring, guaranteeing the access to live data is efficient.

To implement this table, a query to delete data older than two minutes is implemented, every time a reading arrives to Node-RED. This ensures that we only display the current data in the website for visualization created. The query is the following for the BpM example:

```sql
  DELETE FROM BPM
  WHERE "TIMESTAMP" <= datetime('now', '-2 minutes', 'localtime');
```

Listing 6. SQL - Delete records from ’BPM’ every 2 minutes

In order to show the values of BpM and SpO2 on its user interface, the smartwatch subscribes to the topics ”/BpM’ and ”/SpO2”, and receives the readings from the oximeter through Node-RED. This firmware development can be seen below:

```cpp
void connect() {
//Connecting to WiFi
  Serial.print("checking wifi...");
  while (WiFi.status() != WL_CONNECTED) {  //Not able to connect to WiFi
    Serial.print(".");
    delay(1000);
  }
  while (!client.connect(DEV_NAME, MQTT_USER, MQTT_PW)) {   //Not able to
    connect to the MQTT Broker
    Serial.print(".");
    delay(1000); //try again after 1 second
  }
//When it connects subscribes to both topics "/BpM" and "/SpO2"
  Serial.println("\nconnected!");
```

```
  client.subscribe("/BpM"); //SUBSCRIBE TO TOPIC /BpM
  client.subscribe("/Sp02"); //SUBSCRIBE TO TOPIC /Sp02

}
//When a message is published to the topics subscibed
void messageReceived(String &topic, String &payload) {
  //If it is "/BpM" topic
  if(topic == "/BpM"){
  //Store the BpM reading (in the payload of the message) into the variable
    BpM = payload;
  }
  //If it is "/Sp02" topic
  if(topic == "/Sp02"){
  //Store the Sp02 reading (in the payload of the message) into the variable
    Sp02 = payload;
  }
}
```

Listing 7. Cardiac data collection through MQTT

When the values are stored in the variables, the software builds the user interface of the watch to show these values through the LVGL library. To do these, first we reset the user interface in the area of the cardiac readings by placing a black rectangle in that same area. Then we set the cursor to the position we want to write, select the color of the letters and what to write, using ”tft->println()”.

```
void loop()
{
    (...)
    //Resets the previous values by  placing a black rectangle in that same
   area
     tft->fillRect(0, 85, 250, 53, TFT_BLACK);
    //Write Sp02 Reading in the user interface
```

```
        tft->setCursor(50, 85); //Interface position to write
        ttgo->tft->setTextColor(TFT_ORANGE, TFT_BLACK); // Orange picked
        tft->print("SpO2:"); //Label
        ttgo->tft->setTextColor(TFT_WHITE, TFT_BLACK); // White picked
        tft->println(SpO2); //Cardiac value

    //Write BpM Reading in the user interface
        ttgo->tft->setTextColor(TFT_ORANGE, TFT_BLACK); // Orange picked
        tft->setCursor(50, 110); //Interface position to write
        tft->print("BpM:"); //Label
        ttgo->tft->setTextColor(TFT_WHITE, TFT_BLACK); // White picked
        tft->println(BpM); //Cardiac value
(...)
    }
```

Listing 8. Cardiac data user interface display

The full visualization of the user interface of the smartwatch can be seen in the next subsection where all other elements used in the systems graphical user interface are defined.

3.4.3. System Graphical User Interface

3.4.3.1. *Remote Monitoring App - Website*

To visualize all the data captured and monitor the health of the user, a website was created using HTML, PHP and JavaScript to present the data in a user friendly and objective way to health entities and the user itself.

In figure 20, the main page of the website is displayed.

The website offers 4 options for visualization. The page ”Monitor Heart Rate” presents the user’s heartbeat in real time, while the ”Monitor SpO2” page shows the blood oxygen saturation values also in real time.

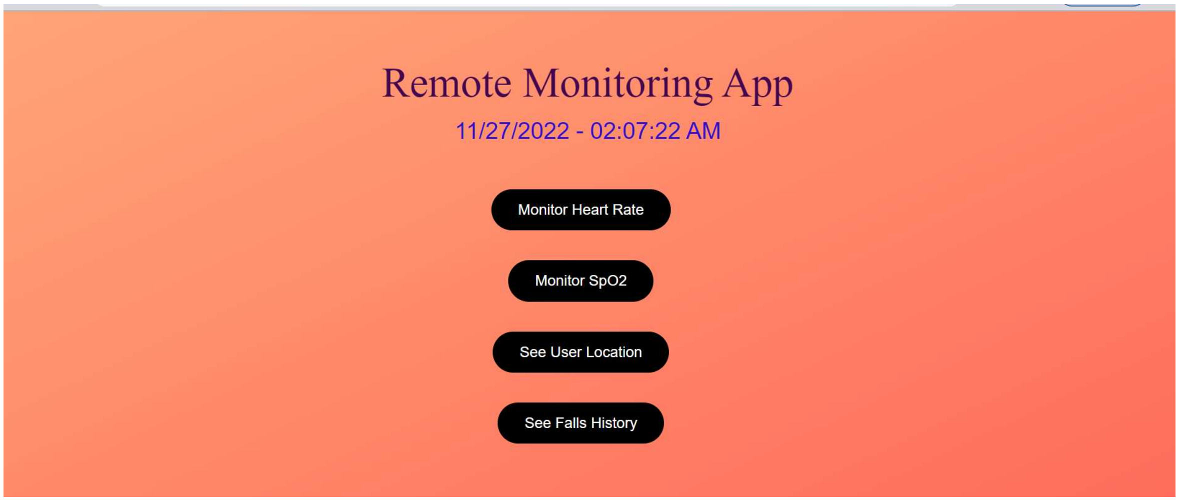


Figure 20. System graphical user interface (Website) - Main page

The monitoring of SpO2 and Heart Rate in real time, is done by consulting the table of data readings in the last 2 minutes, through PHP and JavaScript in the firmware of the website. An example of the firmware for these pages is the BpM monitoring page:

```
    <?php
    $pdo = new PDO('sqlite:DB_HealthRemoteMonitorSystem.db'); //Connects to DB
    $statement = $pdo->query("SELECT * FROM BPM"); //write SQL statement
    $rows = $statement->fetchAll(PDO::FETCH_ASSOC); //run SQL statement
    $stackBPM = array(); //initializes the arrays
    $stackTSBPM = array();
    foreach($rows as $row)    {
    //Value in column 'BPM' to array $stackBPM
    array_push($stackBPM,$row['BPM']);
    //Value in column 'TIMESTAMP' to array $stackTSBPM
    array_push($stackTSBPM,$row['TIMESTAMP']);
    } ?>
<style>  (...)  </style> //Style of the page using CSS
```

```
<!DOCTYPE html>
<html>
```

```
<--!library for the line graph-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.
    js"></script>
<body>
<canvas id="myChart" style="width:100%;max-width:600px"></canvas>
<script>
<--!Alocates the arrays to different objects-->
    var obj = <?php echo json_encode($stackBPM); ?>;
    var objTS = <?php echo json_encode($stackTSBPM); ?>;

new Chart("myChart", { <--!Creates the chart-->
  type: "line",
  data: {
    labels: objTS,
    datasets: [{
      fill: true,
      pointRadius: 1,
     (...)
});
<--!Creates both buttons to check Historic and return to the Main Page of the
    Website-->
</script>
<div class="wrapper">
<button onclick="location.href='http://remotemonitoringsystemapp.42web.io/
    BPMALL.php'" type="button"  class="button">
         Consult BPM Historic</button></div><h2></h2>
<div class="wrapper">
<button onclick="location.href='http://remotemonitoringsystemapp.42web.io/'"
    type="button"  class="button">
         Return to App Menu</button>
         </div>
</body>
```

```
</html>
```

Listing 9. Webpage development - BPM monitoring page

Figures 21 and 22, show the real time remote cardiac monitoring of the system.

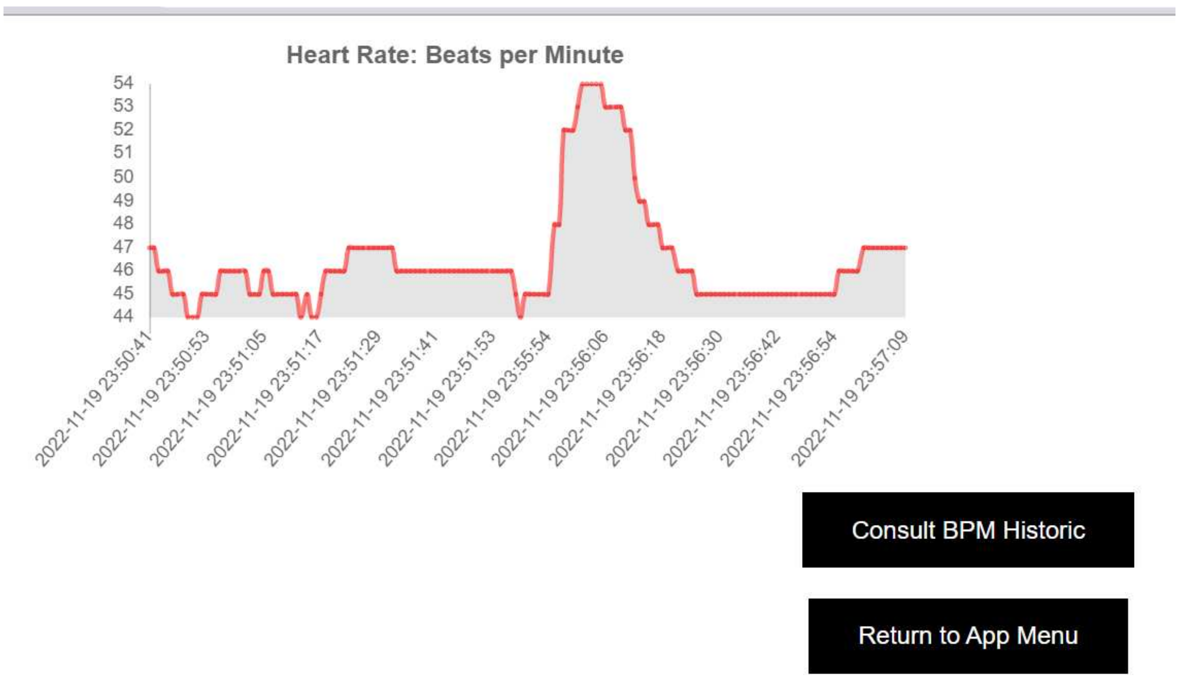


Figure 21. System visualization (Website) - Heart rate monitoring

These 2 monitoring pages offer the possibility to consult the cardiac history since the system started capturing data, by clicking the ”Consult BPM Historic” and the ”Consult SpO2 Historic” buttons in the website, as seen in the firmware developed.

Figures 23, show the full historic of cardiac data for the beats per minute of the user available to be monitored. The SpO2 example follows the same logic

By analysing the figure it is possible to see the cardiac data over a long period of time. This can be very useful for long-term monitoring of patients. This can lead to the discovery of adverse health conditions, such as arrythmia for example.

The ”See User Location” page shows the current location of the user of the system. As seen before, the system also has the possibility to display the real-time location of the user, using GPS coordinates obtained from the smart watch, stored in the database.

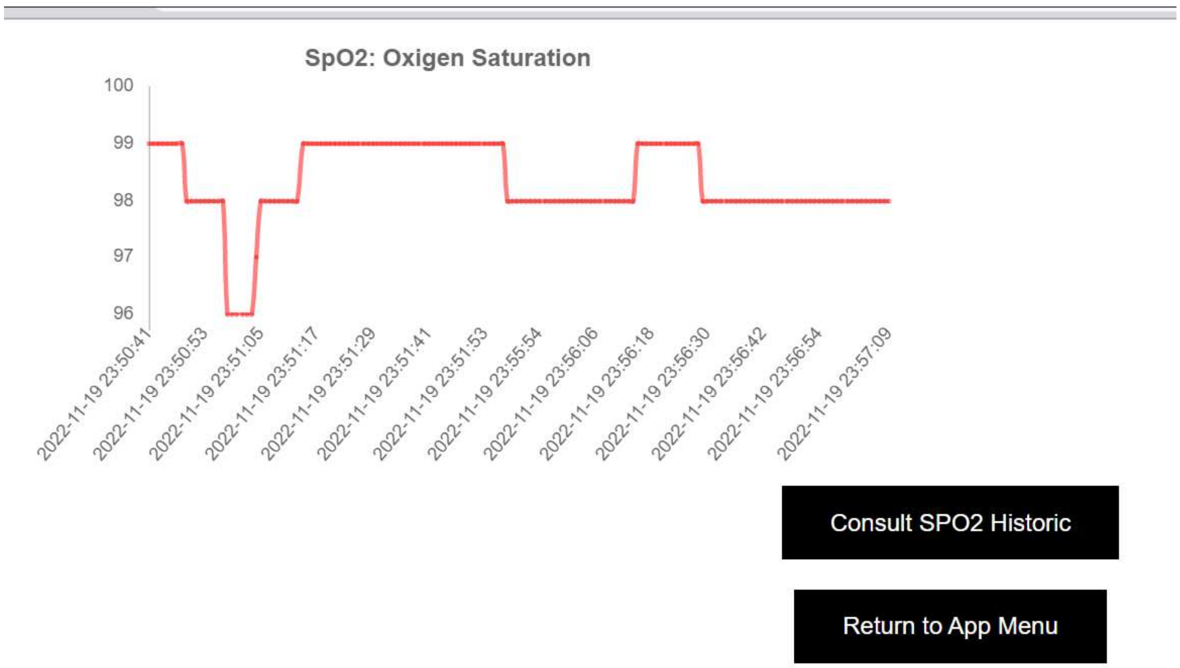


Figure 22. System visualization (Website) - Oxigen saturation monitoring (SpO2)

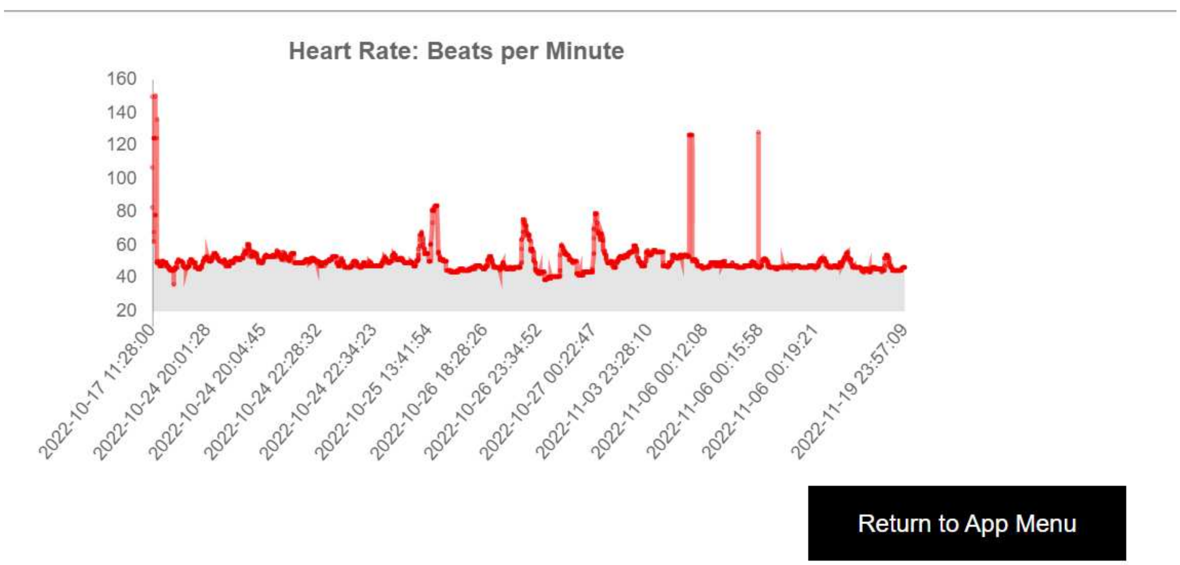


Figure 23. System visualization (Website) - Historic Heart rate monitoring

The smartwatch publishes the coordinates via MQTT protocol using the topic "/coord". The Node-Red subscribes this topic, and stores the coordinates in the "LOCATION" table also in the system's SQLite database.

Then, through a Google Maps API, we generate the map with the last known location of the user. This map can be seen in figure 24.

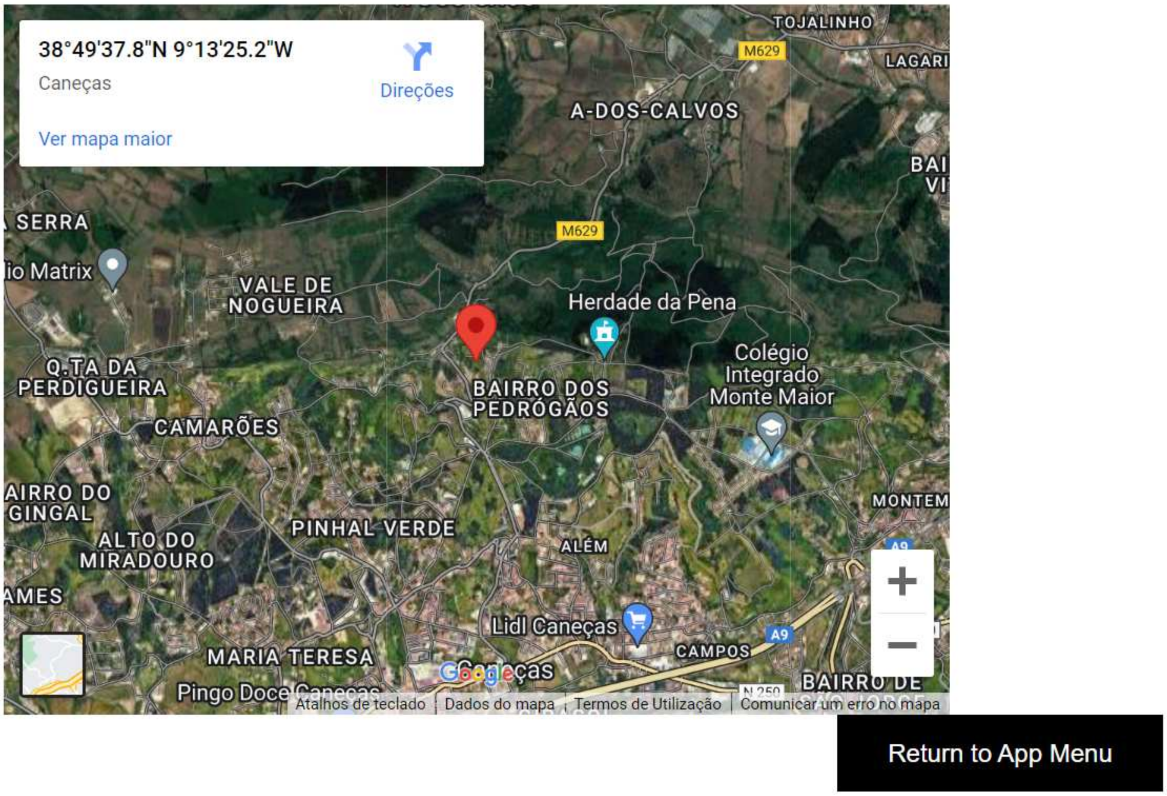


Figure 24. System visualization (Website) - Current location

The firmware developed to use the Google Maps API followed 4 steps.:

- Select the locations present in the SQLite database in the ”LOCATION” table;
- Store each field of the table to corresponding array (e.g. ”array_push($stack-LAT,$row['LAT'])”);
- Arrange the most recent coordinates fetched in ('lat'.'long') format;
- Prepare the link to get the location through the API with the base URL, Key generated through google Account, the coordinates and the selected zoom of the map.

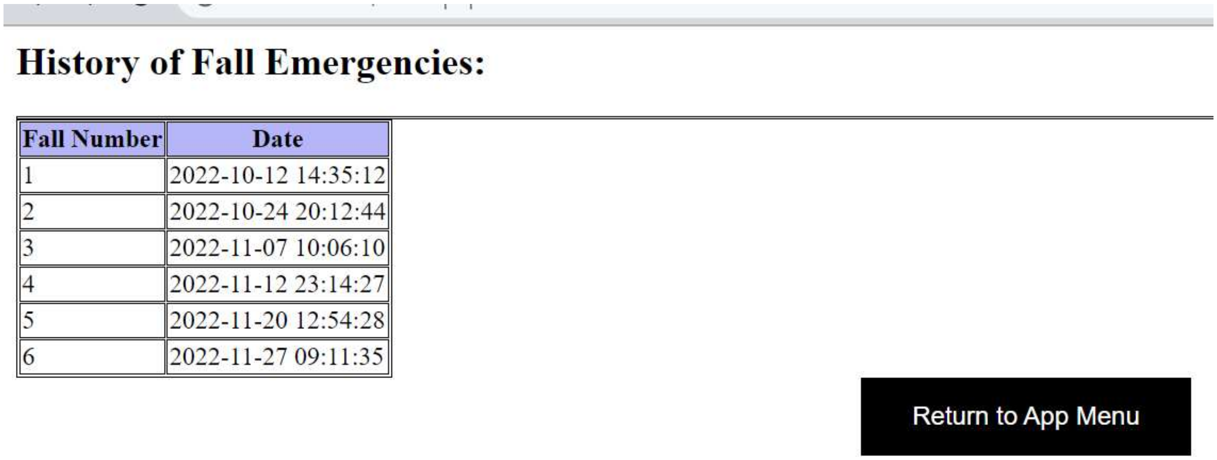
This software can be seen below:

```
<?php
```

```
    $pdo = new PDO('sqlite:DB_HealthRemoteMonitorSystem.db');//Connects to DB
    $statement = $pdo->query("SELECT * FROM LOCATION");//write SQL statement
    $rows = $statement->fetchAll(PDO::FETCH_ASSOC);//run SQL statement
    $stackLAT = array();//Initializes the arrays for each column of the table
    $stackLON = array();
    $stackTSLoc = array();
    foreach($rows as $row)
    {
    //Store each field of the table to corresponding array
    array_push($stackLAT,$row['LAT']);
    array_push($stackLON,$row['LON']);
    array_push($stackTSLoc,$row['TIMESTAMP']);
    }
    //Arrange the coordinates in ('lat'.'long') format
    $lat = end($stackLAT);
    $comma=",";
    $lat .= $comma;
    $lon = end($stackLON);
    $latlong = $lat . $lon;
    //Prepare the link to the location with the URL, Key generated through
   google Account, the coordinates and the selected zoom of the map
    $link1 = "https://www.google.com/maps/embed/v1/place?key=
   AIzaSyD86WNILiswgc7JeKL7WAbwvZb5piIxQwE&q=";
    $link1 .= $latlong;
    $link1 .="&zoom=14";
?>
    <style>(...)
```

Listing 10. Webpage development - User location page

Finally, the website offers the possibility to check the history of fall alerts of the user sent by the system by clicking the ”See Falls History” button. This page can be seen in figure 25



History of Fall Emergencies:

Fall Number	Date
1	2022-10-12 14:35:12
2	2022-10-24 20:12:44
3	2022-11-07 10:06:10
4	2022-11-12 23:14:27
5	2022-11-20 12:54:28
6	2022-11-27 09:11:35

Return to App Menu

Figure 25. System visualization (Website) - Falls history

3.4.3.2. *Remote Monitoring App - Final Smartwatch User Interface*

Figure 26 displays the final display of the watch, showing not only the date and time, but also showing the heart rate and SpO2 data, if the user is measuring them through the oximeter. These values are obtained by subscribing to the "/SpO2" and "/BpM" topics, as seen previously.

Figure 26. Lilygo T-watch-2020 v2 - final display

The watch also features a bar that shows how intense the movement is, mirroring the magnitude of total acceleration. The higher the acceleration the more filled the bar is. Finally, a button that, given a possible fall, the user can click to confirm a false detection of fall as explained in section 3.4.1.

3.4.3.3. *Remote Monitoring App - Node-RED Server Dashboard*

The dashboard presented in figure 27, is a Node-RED server dashboard created to monitor the system in the server side. It has data related to linear acceleration in each axis, acceleration magnitude, the number of falls registered for the user, GPS coordinates and cardiac monitoring data.

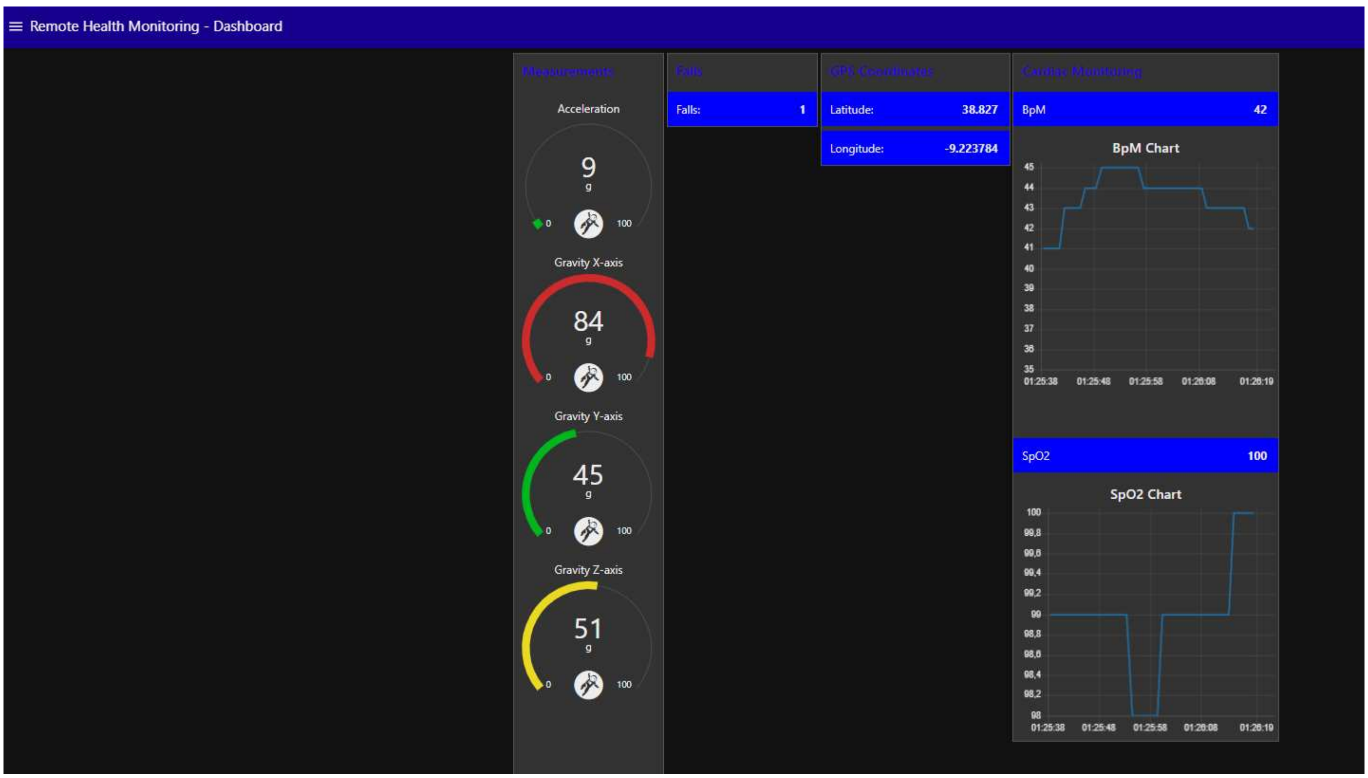


Figure 27. Node-RED Dashboard

CHAPTER 4

Remote Health Monitoring System - Results and Discussion

4.1. Fall Detection Algorithm

To validate the system, several different types of simulated falls were tested, using a thin-width mattress as a landing platform. The platform can be seen in figure 28.

Figure 28. Landing platform

The tests were made on three volunteers, two men and one woman (23-year-old M, 62- year-old M, and a 59-year-old F). The physiological characteristics of the individuals tested can be seen in the following table:

Gender	Age	Height (m)	Weight (kg)	BMI
M	23	1,75	70,4	23,0
M	62	1,71	63,3	22,7
F	59	1,48	55,8	25,5

Table 2. Test subjects physiological characteristics

The table has the Body Mass Index (BMI) characteristic. This indicator is calculated using the following formula:

$$BMI = \frac{weight(kg)}{[height(m)^2]} \tag{4.1}$$

With the metric system, the formula for BMI is weight in kilograms divided by height in meters squared. BMI is an easy screening method for weight category. Some of them are underweight (<18.5), healthy weight (18.5–24.9), overweight (25–29.9), and obesity

(>30). By analysing the table, all male subjects in this tests are considered healthy weight. The female subject enters the overweight category, even if only by a small margin.

Forward falls, sideways falls, and backward falls as if it were a slide, were simulated onto the mattress as the first tests. Each type of fall was executed 20 times by each volunteer of the test.

The other type of tests performed, focused on activities of daily living (ADL) such as walking, running at different speeds, and vertical jumping, analysing whether the system identified false fall alarms in these high-intensity activities. During these tests, cardiac data monitoring was also done to validate the proper functioning of the system.

In the first test a maximum acceleration limit (a_{th}) of 70% was used. First the acceleration values obtained with the accelerometer are transformed to be between 0 and 100, where 0 is no acceleration and 100 is maximum acceleration. By testing with several different values of (a_{th}), the most correct value to serve as the maximum acceleration limit can be estimated, thus maximizing the accuracy of the fall detection algorithm.

Type of Fall	False Negatives	True Positives
Forward	13,3%	86,7%
Sideways to the Left	43,3%	56,7%
Sideways to the Right	26,7%	73,3%
Backward	28,3%	71,7%

Table 3. Test 1 - Types of falls: (a_{th}) = 70%:

Type of Fall	False Negatives	True Positives
Forward	53,3%	46,7%
Sideways to the Left	81,7%	18,3%
Sideways to the Right	73,3%	26,7%
Backward	71,7%	28,3%

Table 4. Test 1 - Types of falls: (a_{th}) = 84%:

In the first test it is observed that when evaluating possible falls, the limiting factor of maximum acceleration has a great impact on the correct detection of a possible fall.

For an a_{th} = 70%, most of the fall events are correctly detected, which is the main goal of the system. Increasing a_{th}, many of the possible falls go unnoticed.

Analyzing these tables, it is observable that frontal drops are the most easily detected by the algorithm, so they do not depend so much on a_{th}.

The side falls tests show that falls to the side where the user places the watch are less easily detected.

In the second test, as expected, for a larger a_{th}, fewer false positives are detected by the system.

Type of ADL	False Positives	True Negatives
Walk	0%	100%
Run - Slow Pace	0%	100%
Run - Medium Pace	40%	60%
Run - Fast Pace	93,3%	6,7%
Jumping	10%	90%

Table 5. Test 2 - Types of ADL's: (a_{th}) = 70%:

Type of ADL	False Positives	True Negatives
Walk	0%	100%
Run - Slow Pace	0%	100%
Run - Medium Pace	0%	100%
Run - Fast Pace	36,6%	63,3%
Jump	0%	100%

Table 6. Test 2 - Types of ADL's: (a_{th}) = 84%:

For an a_{th} of 70%, fast-paced activities were detected as falls most of the time. As the system is designed for elderly users, the focus becomes more on detecting possible falls and minimizing possible undetected falls.

4.2. Cardiac Monitoring

To test the proper functioning of the remote monitoring of cardiac data, the tests to the different ADL's were performed, using the oximeter on the right index finger, to observe the evolution of the heartbeat during the tests.

In the graph seen in figure 30, the change in heart rate and SpO2 values over the course of a 20-second run, with a subsequent 40 seconds of rest, is shown, for the 3 different running intensities tested.

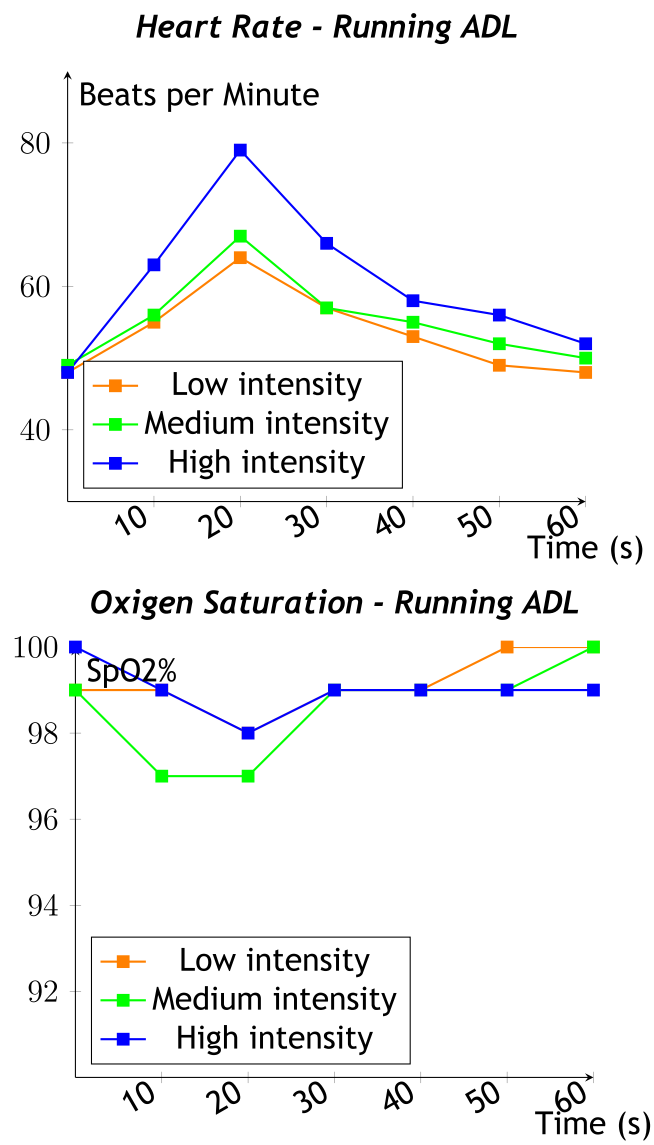


Figure 30. Heart rate and oxygen saturation (SpO2) monitoring - Running activity

By analyzing figure 30, it is visible the expected increase in heart rate at the moment of the run ([0,20] seconds), and the consequent slowing down of the heart rate in the following 40 seconds as a consequence of the resting moment. It is also noted that the SpO2 values tend to decrease during the physical activity (first 20 seconds), although these changes are not very noticeable because of the short duration of the run. The results show that the system can indeed detect falling situations and monitor cardiac data when the oximeter is in the user's finger. However, the limitations of the fall detection algorithm are important in the final definition of the system's performance. The biggest limitation is the fact that it is impossible to recreate real fall situations in a human being. The testing of falls assumes that the tester knows he is going to simulate it, so it won't be close to what a real fall would be, since these are always unexpected.

Another limitation of the testing is that the tests were not performed for every type of possible user of the system. It is practically impossible to cover all the possible characteristics of the possible users of the system, whether they are users with reduced mobility, users with other types of conditions that involve abrupt arm movements, leading to false detection of falls, among other examples.

The system proves to be quite capable of identifying cases of falls and due to the second false alarm confirmation factor, the user can trigger the system in cases of false detections, to avoid false alarms to the caregivers.

Taking this second factor into account, a a_{th} of 70% was selected as the best acceleration threshold value, as the system is more prepared to detect false positives, than to avoid false negatives, so there is no need to increase the value of a_{th}.

To test the influence of the users' physiological characteristics on the effectiveness of the fall detection algorithm, figure 31 shows a graph of the percentage of successfully detected falls during the test phase for an acceleration threshold of 70%, per participant per type of fall.

$$\%FallsCorrectlyDetected = \frac{Detected\_Falls}{N\_Falls\_Tested} \tag{4.2}$$

Analysing the graph, it can be seen that there are some discrepancies between the results of each subject. Although it can be said that it may be random within the scope of each test, it is noticeable that for a backward fall, the younger individual has better results than the older ones. This may be due to the fact that simulating a fall backwards involves taking more risks and, therefore, enhances the movement, detecting falls more easily.

It is also observable that for the female subject, the results were not as good as for the male subjects. This indicates that for a person with less weight and less height, the

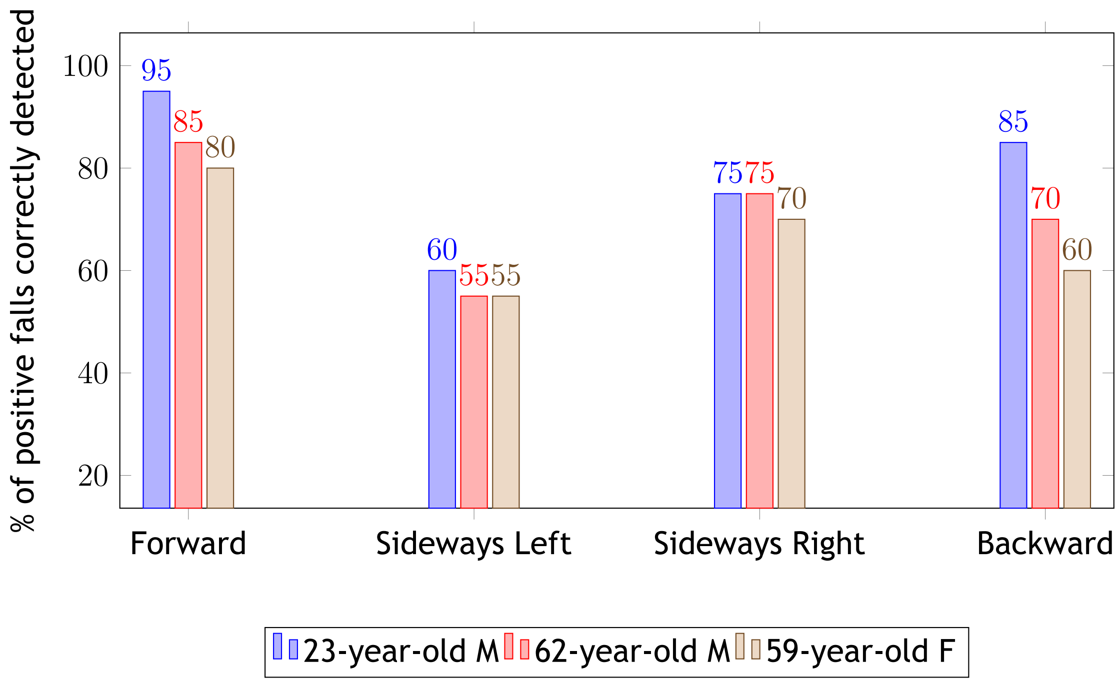


Figure 31. Influence of the users' characteristics on the effectiveness of the fall detection algorithm

acceleration threshold must be revised to be even smaller, because the gravitational acceleration in a fall depends on body mass.

All these factors indicate that physiological characteristics like, age, body type,body mass and more are a determining factor in the success of a fall detection algorithm.

With the system's ability to detect possible falls and display cardiac and SpO2 values on the watch, the user and their caregivers can have better control over the user's health, leading to a higher quality of life.

4.3. Development difficulties

Despite being a system that meets the proposed objectives and fulfils the hypotheses outlined, it was a work that encountered some development difficulties.

The greatest implementation difficulty was the firmware for the LILYGO T-WATCH-2020 V2 smart watch. The greatest difficulty of this work was the elaboration of the firmware for the LILYGO T-WATCH-2020 V2 smart watch. Although it is an embeddable device, it has a whole language and libraries, never personally used before. Some of these difficulties were:

- Connecting the smartwatch to my local Wi-Fi.
- Managing large upload times of firmware developed.
- Changing the graphical interface, without harming previously developed elements.
- Connecting the oximeter to the watch via Bluetooth.

Other implementation difficulties arose from the fact that some central aspects of the work were forced to be changed midway through the work such as the PPG sensor to be used for cardiac monitoring. This was due to unexpected incompatibilities and lack of performance that made the system inoperable.

Other difficulties, also mentioned earlier in this work, were the difficulties in perfecting the fall detection algorithm, since some test factors are undeniable, such as the fact that the tests are not made for every possible type of user and the fact that those who participate in the tests simulate a fall, which may not be 100% accurate with what a real fall would be.

Nevertheless, and taking into account the difficulties already mentioned regarding the fall detection algorithm, this system meets all the proposed objectives and proved to be robust and reliable to remain in operation and continue to be worked on in the future.

CHAPTER 5

Conclusions and Future Work

5.1. Conclusions

This work focuses on a distributed system that could improve the quality of life of the older population by monitoring their mobility and physiological parameters. It is a system capable of identifying possible falls and allowing remote monitoring of cardiac data. It allows sending alarms in emergency situations and allows consulting the user's location through GPS, all this through a monitoring website.

Appropriate firmware was developed to satisfy the system requirements. Validation tests have been carried out, with various results associated with user mobility together with cardiac monitoring including high-intensity movements.

This work may emerge as a foundation for future improvements in the area of health monitoring for the elderly. For this it is necessary to base all the next innovations in the area of health monitoring on IoT, so that the entire elderly population has access to these systems. Only then, it will be possible to serve an entire aging population, even those who live more isolated.

If the balance between demand and supply of remote health monitoring systems is achieved, we can finally give the elderly not only longer lives, but also more quality of life as their health will be monitored closely.

Despite the difficulties in perfecting the fall detection algorithm, the findings presented suggest that based on this system, it is possible to provide reliable remote health monitoring for elderly users, also considering its reduced costs.

5.2. Future Work

It would be beneficial, to explore ways of improving the system created under this work. Most notably, the refinement of the fall detection algorithm. Having ways to adapt the system according to physiological characteristics of each subject, for example. Perhaps adding a third factor, which would help to dissipate the situations of false fall detection due to high intensity movements. Also improving the visual component of the system, by elaborating a more aesthetically appealing website and mobile app.

In the cardiac monitoring phase, it would be advantageous to monitor other factors using oximeters, such as heart rate variability, which is the amount of time between heartbeats. This variability can indicate imbalances in the autonomic nervous system.

It would also be fruitful to pursue further research about monitoring the health of the elderly to validate that non-intrusive remote health monitoring systems based on IoT in an aging society, has practical value. Further research in areas like ballistocardiogram (BCG) monitoring (24h cardiac monitoring), diabetes (blood glucose monitoring), asthma, among others.

CHAPTER 6

Attachments

6.1. Scientific article

Submitted, accepted and presented the article to 2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI), on the 17th of November 2022.

Paper ID: 1570865524.

Title: Remote Health Monitoring System for the Elderly based on mobile computing and IoT.

Remote Health Monitoring System for the Elderly based on mobile computing and IoT

Rafael Cardoso
Instituto de Telecomunicações
Instituto Universitário de Lisboa (ISCTE)
Lisbon, Portugal
rscoa@iscte-iul.pt

Octavian Postolache
Instituto de Telecomunicações
Instituto Universitário de Lisboa (ISCTE)
Lisbon, Portugal
opostolache@lx.it.pt

Carlos Coutinho
Instituto Universitário de Lisboa (ISCTE), ISTAR
Lisbon, Portugal
carlos.eduardo.coutinho@iscte-iul.pt

***Abstract*—Due to the increasing technological innovation over the last decades, the average life expectancy of a human being has been increasing exponentially. Although this is an excellent step forward for humanity, it has led older population to being more prone to illness, making them more vulnerable to accidents such as falls. In this article a study is made on the existing literature in non-intrusive remote health monitoring systems, towards the design and implementation of an IoT system capable of identifying fall situations and monitor cardiac data. A Systematic Literature Review (SLR) method was considered in this work, focused on reviewing the existing literature on remote health monitoring systems, having fall detection algorithms, based in IoT. The Design Science Research (DSR) methodology was used to seek to enhance technology and science knowledge about this paper's topic, through the creation of an innovative artifact.**

The system includes a smart watch (Lily-Go T-Watch-2020 V2), programmable in C under Arduino IDE to detect falls and a photoplethysmography monitoring unit (PPG) based on a Onyx 9560 Bluetooth oximeter, capable of measuring the user's blood oxygen percentage (SpO2) and heart rate, in real time. It also provides remote monitoring through a user-friendly website to visualize live data about the status of the user. The system was tested in volunteers to show the effectiveness of remote health monitoring systems for the elderly population.

***Index Terms*—Health Monitoring, Fall Detection, PPG, DSR, SLR**

I. Introduction

With the significant increase in new technologies starting in the middle of the 20th century, the average life expectancy has increased exponentially, as shown in Fig. 1, reaching an impressive figure of 79 years in 2019 [1]. The increase in average life expectancy raises a new problem. The elderly population is increasingly exposed to disease and health threatening events such as falls. This is called frailty, which is a measure of vulnerability that comes as a consequence of cumulative decline in many physiological systems during a lifetime [2]. The increased frailty of elderly people leads to a relatively poorer quality of life in terms of health, because despite living longer, they live with many more health problems. Quality of life is a very subjective topic. Not only does it involve health, but is also influenced by family life, social and financial status, friends and many others. However, a study of older people over 80 years old, indicates that for 96% of the older people who indicated that they had a poor quality of life, health is a determining factor in their quality of life [3]. Therefore, it is necessary to innovate and create technologies that not only increase the average life expectancy but also provide a higher quality of life for the elderly by monitoring their health-related physiological data. This innovation is based on the growing area of IoT where the number of IoT devices has been increasing over the years. One projection made indicates a growth from 8 billion devices in 2017, to 20 billion devices in 2020 [4]. The applications for IoT are endless and exist in all areas of our daily lives, not only in healthcare. Some examples of the areas are agriculture, automotive, telecommunication, industry, among many others [5]. It is necessary to provide the older population with tools that not only allow them to live longer, but also to live with more quality of life, detecting unexpected accidents and monitoring health related data that can anticipate diseases. These tools must be based on IoT because it is the present and the future of technology, and thus improve the living conditions of the elderly, health wise.

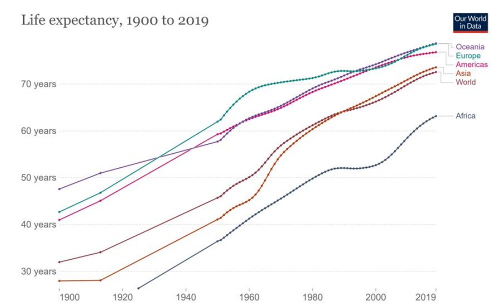


Fig. 1: Average life expectancy through the years [1]

The goal of this article is to create a non-intrusive remote health monitoring system, based on Internet of Things (IoT) sensors, capable of monitoring the living conditions of the elderly and in cases of life-threatening danger send warnings to the proper authorities, possibly saving lives. The diagram of the expected system architecture can be seen in Fig. 2.

This paper will focus on creating a system capable of:

- Identifying falling situations for the user by accessing

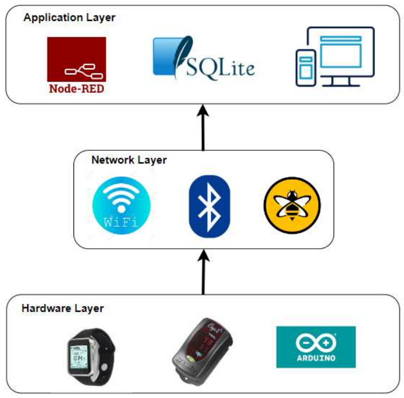


Fig. 2: Remote monitor system architecture

the accelerometer sensor on the Lily-Go T-Watch-2020 V2 smart watch;

- Measure SpO2 and heart rate data of the user, using a Onyx 9560 oximeter, displaying the values on the Lily-Go T-Watch-2020 V2 smart watch via Bluetooth communication, through Node-Red;
- Able to locate the user through GPS location on the watch, in cases of medical emergency;
- Processing the data in the cloud, store the data in an SQLite database and provide a user-friendly visualization website;
- Sending warnings to health entities in the mentioned emergencies.

This work was carried out using the facilities of the ISCTE-IUL Campus and used hardware provided by the Iscte School of Technology and Architecture (ISTA).

II. Related Work & Methodology

This chapter reviews the literature beneficial to this paper. It provides a good technical knowledge and starting bases for the realization of the necessary work.

For a focused and organized research, a systematic review of the related work encountered was defined. Then, the focus was on reviewing existing literature on remote health monitoring systems.

To analyze and understand the related work already done in the field of this paper, the Systematic Literature Review (SLR) method was used. In health care, systematic reviews and meta-analyses have become increasingly relevant [6]. Therefore, to gain a better insight into the area of remote health monitoring, this research method was conducted on the available literature.

To conduct an SLR, it is necessary to define a main search database. The database for this work was the Scopus tool. In the need, to acquire additional information for the literature review, a secondary database, provided by Google, called Google Scholar, was used.

The search in the Scopus tool was performed through a query, while in the Google Scholar tool the search was only performed when necessary, through keywords. The query run was initially: ”(health AND remote AND monitor AND systems)”. This query proved to be too generic as the number of related papers found was 2234. With such a high number, it is impossible to make a SLR of this magnitude individually.

So, to highlight the most relevant works among the enormous number of results obtained, some Inclusion Criteria (IC) and Exclusive Criteria (EC) were defined. These criteria are found in Table 1.

Including	**Exclusive**	**Number of related papers**
All open access	Not being open access	610
From 2014 onwards	Older than 2014	511
In the area of Computer Science, Engineering and Medicine	Not in the area of Computer Science, Engineering and Medicine	415
Written in English or Portuguese	Written in other than Portuguese or English	413
Document type: Article	Not an Article	298

TABLE I: Inclusion and Exclusion criteria defined in the search for related work.

Even with the ICs and ECs defined, there were still a substantially large number of related articles. So, the authors applied an additional exclusive criterion that removes articles related only to medicine. This leaves us with 137 articles. The next step to slim down this number is to skim through the abstract of the remaining articles and select those that appear to be more relevant to the work to be carried out.

After this step there were 55 articles left, which were then analyzed for the next subsection, reviewing how monitoring systems are developed.

Fall detection systems are an important kind of monitoring devices, particularly for the elderly population, where falls can have serious health effects. Main challenge fall detection systems face is differentiating a fall from activities of daily living.

The work carried out by [7] concludes that current fall detection research is hampered by several flaws. The fact that simulated fall scenarios are only performed by young healthy volunteers, the fact that simulated falls do not always represent real fall situations and the fact that falls are usually performed onto thick mats that provide a cushioning effect and change the characteristics of the fall impact from that of a real fall are some of these shortcomings.

In the work carried out by [8], the fall detection system, uses a threshold-based fall detection algorithm. The first feature to define the event of a fall, is defining the threshold of the sum acceleration and rotation angle information. When an actual fall occurs, the impact of the human body with the ground produces a visible peak in the cumulative acceleration, which can be determined by the equation:

$$|a| = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{1}$$

Where a_x, a_y and a_z represent each axis acceleration of the accelerometer. This magnitude is the first step to distinguish high intensity movements from others. However, high intensity activities, like jumping, also produce peak values, which mean that additional detection features are required.

The second feature is an angle calculated based on acceleration measurements. By separating the gravity components before and after a human fall, then it is possible to calculate the rotation angle of accelerometer coordinate in 3D space, which is also equivalent to the rotation angle of the gravity vector relative to the fixed coordinate.

This work concludes that normal activity of resting also has similar rotation as falling and it may trigger false falling alarms. The correct definition of the parameter $a_{threshold}$, is the most important factor to distinguish falling from lying down. It also concludes that it is important to experiment with a group of people who are more differentiated in terms of age, gender, and weight, in order to improve the reliability and robustness of the threshold.

Within the scope of the topic under study, the research questions that motivate the analysis and subsequent response are the following:

1) How can a non-intrusive health remote monitor system influence the health of an older patient?
2) How does the system improve the quality of life of the elderly?
3) How to design the system so that the response to an emergency is robust and effective?

With these research questions, this work aims to innovate in remote health monitoring, especially in the area of elderly health.

This work also defined a set of hypothesis to be proposed to reply to each research question. Each hypothesis comprises a prediction about how two or more variables will interact. It's like making an educated guess about what will happen in an experiment [9].

- **Hypothesis 1:** By being a system capable of detecting falls, monitor cardiac data and able to send warnings to caregivers or even in extreme cases to authorities in cases of medical emergencies, the patient will directly benefit from better health over time and quick responses in cases of distress.
- **Hypothesis 2:** As seen in the previous subsections, health is the determining factor in quality of life, for the older population. By improving their health conditions, their quality of life is directly improving as well.
- **Hypothesis 3:** The system needs to be user-friendly to be perceptible to the average caretaker, capable of rapid response to emergencies by having low latency levels and needs to be a robust system in case of power failures, hardware failures, among other unforeseen events.

In this paper, the research process based on Design Science Research (DSR) was followed. DSR is a problem-solving paradigm that aims to improve human knowledge through the creation of novel products. Simply said, DSR aims to improve technology and science knowledge bases by developing new artifacts that solve problems while also improving the environment in which they are implemented [10].

The first step begins with identifying and defining the research problem as well as justifying the value of a solution. Then, defining the objective for that solution is the second step. After that, in the third step, the design and development of the artifact is done, by determining the artifact's desired functionality and its architecture. The fourth step consists in demonstrating the use of the artifact to solve instances of the research problem. Afterwards, in the fifth step, an evaluation of the solution is made, based on a comparison of the goals and the actual results obtained through the use of the artifact. Lastly, all aspects of the problem and the designed artifact are communicated to the relevant stakeholders, being other researchers or other relevant audiences, like professionals on the same area of work.

The DSR process also includes four different entry points [11]: **The problem centered approach**, starting in step one, for research where the concept came from observing the situation or through a report from a previous project that proposed future research. **The objective-centered solution**, starting in step two, mostly for research that needs to develop an artifact. **The design and development-centered approach**, starting in step three, being suitable for situations where the artifact already exists, but needs further developments. **The client/context initiated solution**, starting in step four, for research based on observing a practical solution that has worked.

In this work, the DSR model was applied using the problem centered approach due to the fact that the addressed problem is already known.

III. System Description

A. Hardware

The hardware used in this system was the LILYGO T-WATCH-2020 V2 and the Onyx 9560 oximeter.

The LILYGO T-WATCH-2020 V2 is based on a design concept that can be interacted with, networked, programmed, and worn. It incorporates Wi-Fi/Bluetooth, which is more convenient to connect to the Internet and is simple to program and build, with ESP32. The big difference from V1 to V2 is the addition of the GPS function and the ability to store data on a memory card.

The Onyx 9560 is a pulse and blood oxygen saturation monitoring device, designed for individuals who desire a Bluetooth oximeter, allowing them to upload measurements to their phone or digital device. It also offers a range of up to 100 meters, and allows the user mobility during use, unlike many other oximeters.

B. Software

The Arduino Integrated Development Environment - or Arduino Software (IDE) - is a free software that contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the microcontroller hardware to upload programs and communicate with them. This is the software we use to program the LILYGO T-WATCH-2020 V2.

Node-RED is a programming tool for connecting hardware devices, APIs and online services. Node-RED is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware. This is where all system components are connected and communicate with each other and the database, via Wi-Fi using the MQTT protocol and via Bluetooth in the oximeter case.

SQLite is an open-source SQL database that stores data to a text file on a device. One of SQLite's greatest advantages is that it can run nearly anywhere. Its flexibility and the fact that it is better prepared to deal with a smaller flow of data, was the decisive factor in choosing it as a database.

HTML stands for HyperText Markup Language. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes. HTML is used for web development, Internet navigation and web documentation.

C. System Overview

The system was designed to detect possible cases of falling using the accelerometer sensor embedded in the smartwatch. A fall detection algorithm, designed in the Arduino IDE using C programming language was also considered.

The fall identification algorithm has two fall detection factors. The first is comparing the acceleration (Acc) calculated with the linear acceleration per accelerometer axis (X, Y and Z), with a predefined maximum acceleration limit (a_{th}), which is tested in the next chapter of this work, to obtain the value with the highest hit percentage, in the detection of true falls.

The second factor helps a lot in detecting false positives and is implemented in the touchscreen interface of the smart watch. If the user does not click a button in the touchscreen interface, indicating that it was a false fall alarm in a period of time after the detection, a possible fall is confirmed, and the alarm for the caregivers will be sent.

The system also has the ability to collect heart rate and SpO2 data using an Onix 9560 oximeter. The device communicates with the Node-RED platform via Bluetooth, which forwards messages to the smartwatch via the MQTT protocol, publishing the heart rate values to the "/BpM" topic and publishing the blood oxygen saturation values to the "/Spo2" topic.

Node-RED stores all the data collected in an SQLite database. The website, coded using HTML and PHP, is used to visualize this data in a user friendly and objective way.

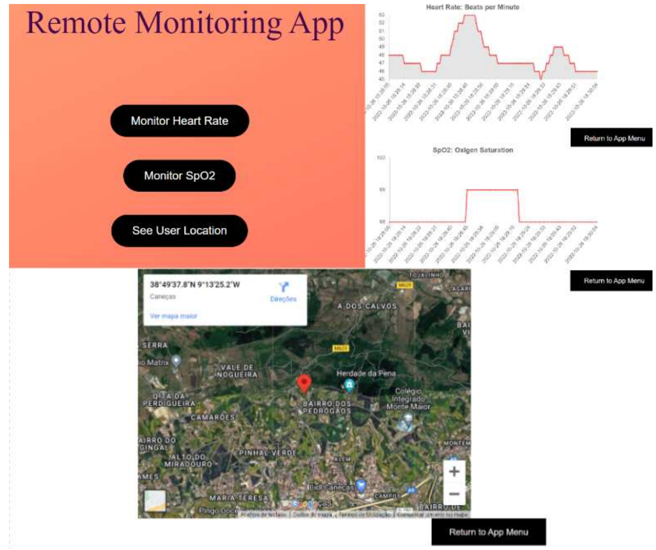


Fig. 3: System visualization (Website) - Heart Rate, SpO2 and current GPS location of the user

Fig. 4: Lilygo T-watch-2020 v2 - final display

The system also has the possibility to display the real-time location of the user, using GPS coordinates obtained from the smart watch, stored in the database.

Fig.3 presents the visualization of the collected data via the website created. Thus, the website offers 3 options for visualization. The page "Monitor Heart Rate" presents the user's heartbeat, while the "Monitor SpO2" page shows the blood oxygen saturation values. Finally, the "See User Location" page shows the current location of the user of the system.

Fig. 4 displays the visualization through the watch, showing not only the date and time, but also showing the heart rate and SpO2 data, if the user is measuring them through the oximeter. The watch also features a bar that shows how intense the movement is and a button that, given a possible fall, the user can click to confirm the false positive.

IV. Results and Discussions

To validate the system, several different types of simulated falls were tested, using a thin-width mattress as a landing platform. The tests were made on three volunteers, two men

Type of Fall	False Negatives	True Positives
Forward	13,3%	86,7%
Sideways to the Left	43,3%	56,7%
Sideways to the Right	26,7%	73,3%
Backward	28,3%	71,7%

TABLE II: Types of falls: (a_{th}) = 70%:

Type of Fall	False Negatives	True Positives
Forward	53,3%	46,7%
Sideways to the Left	81,7%	18,3%
Sideways to the Right	73,3%	26,7%
Backward	71,7%	28,3%

TABLE III: Types of falls: (a_{th}) = 84%:

and one woman (23-year-old M, 62- year-old M, and a 59-year F). Forward falls, sideways falls, and backward falls as if it were a slide, were simulated onto the mattress. Each type of fall was executed 20 times by each volunteer of the test.

The other type of tests performed, focused on activities of daily living (ADL) such as walking, running at different speeds, and vertical jumping. During these tests, cardiac data monitoring was also done to validate the proper functioning of the system.

In the first test a maximum acceleration limit (a_{th}) of 70% was used. First the acceleration values obtained with the accelerometer are transformed to be between 0 and 100, where 0 is no acceleration and 100 is maximum acceleration. By testing with several different values of (a_{th}), the most correct value to serve as the maximum acceleration limit can be estimated, thus maximizing the accuracy of the fall detection algorithm.

In the second test the system was faced with cases of false positives results in various day-to-day activities (ADL), like walking, running, and jumping.

In the first test it is observed that when evaluating possible falls, the limiting factor of maximum acceleration has a great impact on the correct detection of a possible fall.

For an a_{th} = 70%, most of the fall events are correctly detected, which is the main goal of the system. Increasing a_{th}, many of the possible falls go unnoticed.

Analyzing these tables, the authors concluded that frontal drops are the most easily detected by the algorithm, so they do not depend so much on a_{th}.

Type of ADL	False Positives	True Negatives
Walk	0%	100%
Run - Slow Pace	0%	100%
Run - Medium Pace	40%	60%
Run - Fast Pace	93,3%	6,7%
Jumping	10%	90%

TABLE IV: Types of ADL's: (a_{th}) = 70%:

Type of ADL	False Positives	True Negatives
Walk	0%	100%
Run - Slow Pace	0%	100%
Run - Medium Pace	0%	100%
Run - Fast Pace	36,6%	63,3%
Jump	0%	100%

TABLE V: Types of ADL's: (a_{th}) = 84%:

The side falls tests show that falls to the side where the user places the watch are less easily detected.

In the second test, as expected, for a larger a_{th}, fewer false positives are detected by the system.

For an a_{th} of 70%, fast-paced activities were detected as falls most of the time. As the system is designed for elderly users, the focus becomes more on detecting possible falls and minimizing possible undetected falls.

To test the proper functioning of the remote monitoring of cardiac data, the tests to the different ADL's were performed, using the oximeter on the right index finger, to observe the evolution of the heartbeat during the tests.

In the graph seen in Fig. 6, the change in heart rate and SpO2 values over the course of a 20-second run, with a subsequent 40 seconds of rest, is shown, for the 3 different running intensities tested.

By analyzing Fig.6, it is visible the expected increase in heart rate at the moment of the run ([0,20] seconds), and the consequent slowing down of the heart rate in the following 40 seconds as a consequence of the resting moment. It is also noted that the SpO2 values tend to decrease during the physical activity (first 20 seconds), although these changes are not very noticeable because of the short duration of the run. The results show that the system can indeed detect falling situations and monitor cardiac data when the oximeter is in the user's finger. However, the limitations of the fall detection algorithm are important in the final definition of the system's performance. The biggest limitation is the fact that it is impossible to recreate real fall situations in a human being. The testing of falls assumes that the tester knows he is going to simulate it, so it won't be close to what a real fall would be, since these are always unexpected.

Another limitation of the testing is that the tests were not performed for every type of possible user of the system. It is practically impossible to cover all the possible characteristics of the possible users of the system, whether they are users with reduced mobility, users with other types of conditions that involve abrupt arm movements, leading to false detection of falls, among other examples.

The system proves to be quite capable of identifying cases of falls and due to the second false alarm confirmation factor, the user can trigger the system in cases of false detections, to avoid false alarms to the caregivers.

Taking this second factor into account, a a_{th} of 70% was selected as the best acceleration threshold value, as the system

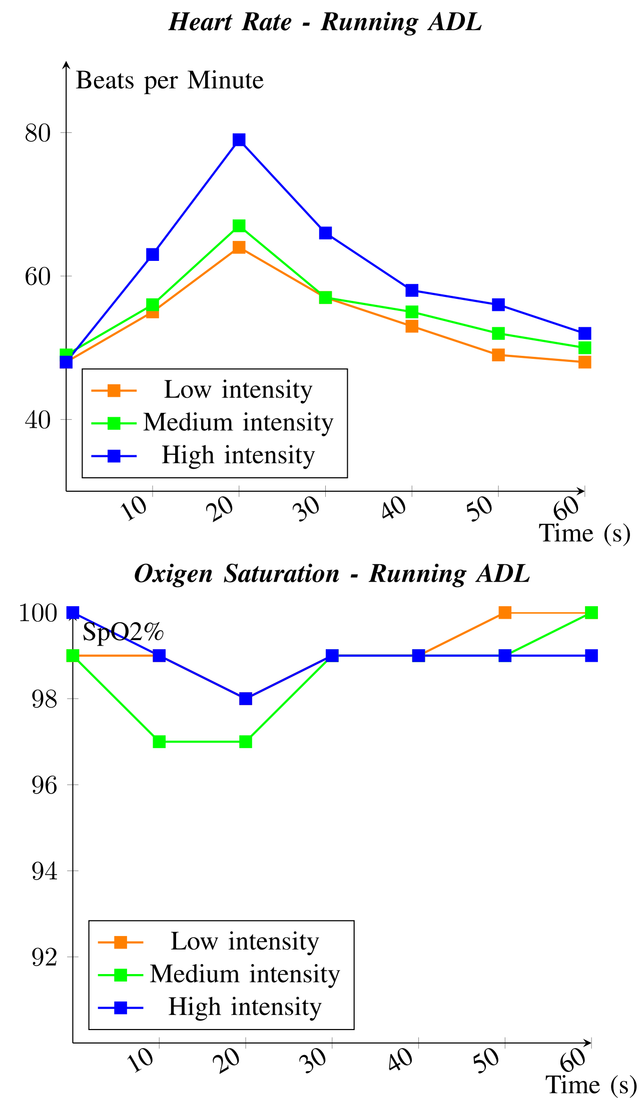


Fig. 6: Heart rate and oxygen saturation (SpO2) monitoring - Running activity

is more prepared to detect false positives, than to avoid false negatives, so there is no need to increase the value of a_{th}.

With the system's ability to detect possible falls and display cardiac and SpO2 values on the watch, the user and their caregivers can have better control over the user's health, leading to a higher quality of life.

Conclusion

This work focuses on a distributed system that could improve the quality of life of the older population by monitoring their mobility and physiological parameters. Appropriate firmware was developed to satisfy the system requirements including real-time monitoring of falls and cardiac data. Validation tests have been carried out, with various results associated with user mobility together with cardiac monitoring including high-intensity movements.

Despite the difficulties in perfecting the fall detection algorithm, the findings presented suggest that based on this system, it is possible to provide reliable remote health monitoring for elderly users, also considering its reduced costs.

It would be fruitful to pursue further research about monitoring the health of the elderly to validate that non-intrusive remote health monitoring systems based on IoT in an aging society, has practical value. Further research in areas like ballistocardiogram (BCG) monitoring (24h cardiac monitoring), diabetes (blood glucose monitoring), asthma, among others.

Acknowledgment

This work was supported by Fundação para a Ciência e a Tecnologia, I.P. (FCT) [ISTAR Projects: UIDB/04466/2020 and UIDP/04466/2020].

References

[1] . O.-O. Max Roser and H. Ritchie, "Life expectancy,"Our World in Data, 2013,https://ourworldindata.org/life-expectancy.

[2] A. Clegg, J. Young, S. Iliffe, M. O. Rikkert, and K. Rockwood, "Frailty in elderly people," The Lancet, vol. 381, pp. 752–762, 9868 2013, issn: 1474547X. doi: 10. 1016/S0140-6736(12)62167-9.

[3] F. M. Xavier, M. Ferraz, N. Marc, N. U. Escosteguy, and E. H. Moriguchi, "Elderly people s definition of quality of life," Brazilian Journal of Psychiatry, vol. 25, pp. 31–39, 2003.

[4] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in 2018 IEEE Security and Privacy Workshops (SPW), 2018, pp. 29–35. doi: 10.1109/SPW.2018.00013.

[5] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," Wireless Personal Communications, vol. 58, pp. 49–69, 1 May 2011, issn: 09296212. doi: 10.1007/S11277-011-0288-5.

[6] D. Moher, A. Liberati, J. Tetzlaff, et al., "Preferred reporting items for systematic reviews and meta-analyses: The prisma statement," PLoS Medicine, vol. 6, no. 7, 2009, cited By 34924. doi: 10.1371/journal.pmed.1000097. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-68049122102&doi=10. 1371%2fjournal.pmed.1000097&partnerID=40&md5=ae3854790b86846c4e4519473a3a3417. vol. 10, pp. 57–76, 1 Jan. 2019, issn: 18685145. doi: 10.1007/S12652-017-0598-X.

[7] T. Shany, S. Redmond, M. Narayanan, and N. Lovell, "Sensors-based wearable systems for monitoring of human movement and falls," IEEE Sensors Journal, vol. 12, no. 3, pp. 658–670, 2012, cited By 197. doi: 10.1109/JSEN.2011.2146246. [Online]. Available: https : / / www . scopus . com / inward / record . uri ? eid = 2 - s2 . 0 - 84856846340 & doi = 10 . 1109 % 2fJSEN . 2011 . 2146246 & partnerID = 40 & md5 = 5477ee5e67102e71b40bc608bd03bfdf.

[8] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, "Development of a wearable-sensor-based fall detection system," International Journal of Telemedicine and Applications, vol. 2015, 2015, cited By 134. doi: 10 . 1155 / 2015 / 576364. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84924370518&doi=10. 1155%2f2015%2f576364&partnerID=40&md5=9f3cbbe7c346b8f9c9a8edc23909b661

[9] Hasa, Difference between hypothesis and research question: Meaning, features, characteristics, usage, Aug. 2016. [Online]. Available: https : / / pediaa . com / difference-between-hypothesis-and-research-question/.

[10] J. v. Brocke, A. Hevner, and A. Maedche, "Introduction to design science research," in Sep. 2020, pp. 1–13, isbn: 978-3-030-46780-7. doi: 10.1007/978-3- 030-46781-4_1.

[11] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," Journal of Management Information Systems, vol. 24, no. 3, pp. 45–77, 2007. doi: 10 . 2753 / MIS0742 - 23 1222240302. eprint: https://doi.org/10.2753/MIS0742-1222240302. [Online]. Available: https://doi.org/10.2753/MIS0742-1222240302.

6.2. System firmware developed - LILYGO T-WATCH-2020 V2:

```
#define LILYGO_WATCH_2020_V2
#define LILYGO_WATCH_LVGL

#include <LilyGoWatch.h>


//--------------------------------------------
#include <WiFi.h>

#define BROKER_IP    "192.168.1.126"
#define DEV_NAME     "device_random_name"
#define MQTT_USER    ""
#define MQTT_PW      ""

#define SerialMon                   Serial

const char ssid[] = "LT132428 4315";
const char pass[] = "s0>096B0";

#include <MQTT.h>

WiFiClient net;
MQTTClient client;
unsigned long lastMillis = 0;
//--------------------

TTGOClass *ttgo;
TFT_eSPI *tft;
BMA *sensor;

TinyGPSPlus *gps;
```

```

Adafruit_DRV2605 *drv;
uint8_t effect = 16;


boolean possibleFall = 0; //queda Possivel


int acc_1 = 50; //generic number to initialize the acceleration
int nquedas=0; // Number of falls
String lng = (String) 0;  //Longitude
String lat = (String) 0;  //Latitude

String SpO2 = "";
String BpM = "";

lv_obj_t * bar1;

int iterador = 0; //iterator to control the vibration process during the 20s
    period after a possible fall detected, so as not to be constantly vibrating
    ;

uint32_t targetTime = 0;       // for next 1 second timeout

byte omm = 99;
boolean initial = 1;
byte xcolon = 0;
unsigned int colour = 0;

static uint8_t conv2d(const char *p)
{
    uint8_t v = 0;
    if ('0' <= *p && *p <= '9')
```

```
        v = *p - '0';
    return 10 * v + *++p - '0';
}

uint8_t hh = conv2d(__TIME__), mm = conv2d(__TIME__ + 3), ss = conv2d(__TIME__ + 6); // Get H, M, S from compile time


//---------------------------
void connect() {
  Serial.print("checking wifi...");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(1000);
  }

  Serial.print("\nconnecting...");
  while (!client.connect(DEV_NAME, MQTT_USER, MQTT_PW)) {
    Serial.print(".");
    delay(1000);
  }
  Serial.println("\nconnected!");
  client.subscribe("/BpM"); //SUBSCRIBE TO TOPIC /BpM
  client.subscribe("/SpO2"); //SUBSCRIBE TO TOPIC /SpO2

}

void messageReceived(String &topic, String &payload) {
  Serial.println("incoming: " + topic + " - " + payload);

  if(topic == "/BpM"){
    Serial.println("Recebi um BpM");
    BpM = payload;
```

```
  }
  if(topic == "/SpO2"){
    Serial.println("Recebi um SpO2");
    SpO2 = payload;
  }
}
//---------------------------------------------------


static void event_handler(lv_obj_t *obj, lv_event_t event)
{
    if (event == LV_EVENT_CLICKED) {
        Serial.printf("Falso positivo confirmado\n");
        possibleFall = 0;
    }
}


void setup(void)
{
  Serial.begin(115200);

  //---------------------------------------------------------

  WiFi.begin(ssid, pass);

  client.begin(BROKER_IP, 1883, net);
  client.onMessage(messageReceived);
  connect();

  //---------------------------------

    ttgo = TTGOClass::getWatch();
```

```
  ttgo->begin();
  ttgo->openBL();
  ttgo->tft->setTextFont(1);
  ttgo->tft->fillScreen(TFT_BLACK);
  ttgo->tft->setTextColor(TFT_YELLOW, TFT_BLACK); // Note: the new fonts do not draw the background colour
  targetTime = millis() + 1000;

nquedas=0;


  //Receive objects for easy writing
  tft = ttgo->tft;
  sensor = ttgo->bma;


  gps = ttgo->gps;
  ttgo->trunOnGPS();
  ttgo->gps_begin();
  gps =  ttgo->gps;




   drv = ttgo->drv;
   ttgo->lvgl_begin();

     ttgo->enableDrv2650();

  drv->selectLibrary(1);

  // I2C trigger by sending 'go' command
  // default, internal trigger when sending GO command
```

```
  drv->setMode(DRV2605_MODE_INTTRIG);

  // Accel parameter structure
  Acfg cfg;

  cfg.odr = BMA4_OUTPUT_DATA_RATE_100HZ;

  cfg.range = BMA4_ACCEL_RANGE_2G;

  cfg.bandwidth = BMA4_ACCEL_NORMAL_AVG4;

  cfg.perf_mode = BMA4_CONTINUOUS_MODE;

  // Configure the BMA423 accelerometer
  sensor->accelConfig(cfg);

  // Enable BMA423 accelerometer
  sensor->enableAccel();


  lv_obj_set_style_local_bg_color (lv_scr_act(), LV_OBJ_PART_MAIN,
 LV_STATE_DEFAULT, LV_COLOR_BLACK);

  lv_obj_t *label;


      lv_obj_t *btn1 = lv_btn_create(lv_scr_act(), NULL);
      lv_obj_set_event_cb(btn1, event_handler);
      lv_obj_align(btn1, NULL, LV_ALIGN_CENTER, 0, 90);

      label = lv_label_create(btn1, NULL);
      lv_label_set_text(label, "NO FALL");

```

```

        bar1 = lv_bar_create(lv_scr_act(), NULL);
        lv_obj_set_size(bar1, 150, 20);
        lv_obj_align(bar1, NULL, LV_ALIGN_CENTER, 0, 45);
        lv_bar_set_anim_time(bar1, 1000);
        lv_bar_set_value(bar1, 0, LV_ANIM_ON);



    tft->setTextFont(4);
    tft->setTextColor(TFT_WHITE, TFT_BLACK);


}

void loop()
{
//----------------------------


client.loop();
  if (!client.connected()) {
    connect();
  }


showClock(); //Show the current time and date on the watch;

 while (ttgo->hwSerial->available()) {
        int r = ttgo->hwSerial->read();
        ttgo->gps->encode(r);
    }
    display_info();
```

```


Accel acc;



    // Get acceleration data
    bool res = sensor->getAccel(acc);

    if (res == false) {
        Serial.println("getAccel FAIL");
    } else {
        // Show the data

        int new_x= acc.x;
        new_x = map(new_x, -1200, 1200, 0, 100);
        client.publish("/x", (String) new_x);

        int new_y= acc.y;
        new_y = map(new_y, -1200, 1200, 0, 100);
        client.publish("/y", (String) new_y);

        int new_z= acc.z;
        new_z = map(new_z, -1200, 1200, 0, 100);
        client.publish("/z", (String) new_z);


     //Magnitude of Acceleration
        tft->setCursor(80, 100);
        int acc_avg = sqrt((new_x * new_x) + (new_y * new_y) + (new_z * new_z)
   );
        int diff = abs(acc_avg-acc_1);
        lv_bar_set_value(bar1, diff, LV_ANIM_ON);
```

```
        client.publish("/acc", (String) diff); //PUBLISH TO TOPIC /acc
        acc_1 = acc_avg;


       client.publish("/fall", (String) nquedas); //PUBLISH TO TOPIC /nquedas

        if( diff < 70){

          }else{

           ConfirmFall();

          }

     lv_task_handler();


     if(possibleFall == 1){
          iterador = iterador + 1;
          if(iterador % 10 == 0){
            drv->go();
            Serial.println("Vibracao");
          // wait a bit
            delay(100);
             drv->setWaveform(1, 0);       // end waveform

          }

          if(iterador >= 99){ //+- 20s
              nquedas = nquedas + 1;
              client.publish("/Fall Alert!!", "O utilizador sofreu uma queda!"
   ); //PUBLISH TO TOPIC /FALL ALERT
              possibleFall = 0;
```

```
            }
      }else{ // User carrega no botão
             iterador = 0;
      }



      delay(200);

         tft->fillRect(0, 85, 250, 53, TFT_BLACK);


         tft->setCursor(50, 85);
         ttgo->tft->setTextColor(TFT_ORANGE, TFT_BLACK); // Orange
         tft->print("SpO2:");
         ttgo->tft->setTextColor(TFT_WHITE, TFT_BLACK); // White
         tft->println(SpO2);
         ttgo->tft->setTextColor(TFT_ORANGE, TFT_BLACK); // Orange
         tft->setCursor(50, 110);
         tft->print("BpM:");
         ttgo->tft->setTextColor(TFT_WHITE, TFT_BLACK); // White
         tft->println(BpM);




     }

}



  void ConfirmFall(){
```

```
    // set the effect to play
          drv->setWaveform(0, effect);  // play effect

          // play the effect!
            drv->go();
            Serial.println("AQUI");

          // wait a bit
            delay(200);


        drv->setWaveform(1, 0);       // end waveform


        possibleFall=1;


    }



void showClock(){

      //----------------------------
    if (targetTime < millis()) {
        targetTime = millis() + 1000;
        ss++;              // Advance second
        if (ss == 60) {
            ss = 0;
            omm = mm;
            mm++;            // Advance minute
```

```
        if (mm > 59) {
            mm = 0;
            hh++;          // Advance hour
            if (hh > 23) {
                hh = 0;
            }
        }
    }

    if (ss == 0 || initial) {
        initial = 0;
        ttgo->tft->setTextColor(TFT_GREEN, TFT_BLACK);
        ttgo->tft->setCursor (8, 52);
        ttgo->tft->print(__DATE__); // This uses the standard ADAFruit small font


    }

    // Update digital time
    byte xpos = 6;
    byte ypos = 0;
    if (omm != mm) { // Only redraw every minute to minimise flicker
        // Uncomment ONE of the next 2 lines, using the ghost image demonstrates text overlay as time is drawn over it
        ttgo->tft->setTextColor(0x39C4, TFT_BLACK);  // Leave a 7 segment ghost image, comment out next line!
        ttgo->tft->setTextColor(TFT_BLACK, TFT_BLACK); // Set font colour to black to wipe image
        // Font 7 is to show a pseudo 7 segment display.
        // Font 7 only contains characters [space] 0 1 2 3 4 5 6 7 8 9 0 : .
```

```
            ttgo->tft->drawString("88:88", xpos, ypos, 7); // Overwrite the
    text to clear it
            ttgo->tft->setTextColor(0xFBE0, TFT_BLACK); // Orange
            omm = mm;

            if (hh < 10) xpos += ttgo->tft->drawChar('0', xpos, ypos, 7);
            xpos += ttgo->tft->drawNumber(hh, xpos, ypos, 7);
            xcolon = xpos;
            xpos += ttgo->tft->drawChar(':', xpos, ypos, 7);
            if (mm < 10) xpos += ttgo->tft->drawChar('0', xpos, ypos, 7);
            ttgo->tft->drawNumber(mm, xpos, ypos, 7);
        }
    }

    }




    static void display_info(void)
{
    static uint32_t updatePeriod;
    if (millis() - updatePeriod < 20000) {
        return;
    }
    updatePeriod = millis();
    if (gps->location.isUpdated()) {

         float latsmall= gps->location.lat();
         lat = String(latsmall,6);

         float lngsmall= gps->location.lng();
```

```
        lng = String(lngsmall,6);

        String coord = ((String) lat ) + ',' + ((String) lng );

        client.publish("/coord", coord); //PUBLISH TO TOPIC /coord


    }
    delay(20);
}
```

References

[1] E. O.-O. Max Roser and H. Ritchie, “Life expectancy,” *Our World in Data*, 2013, https://ourworldindata.org/life-expectancy.

[2] A. Clegg, J. Young, S. Iliffe, M. O. Rikkert, and K. Rockwood, “Frailty in elderly people,” *The Lancet*, vol. 381, pp. 752–762, 9868 2013, issn: 1474547X. doi: 10.1016/S0140-6736(12)62167-9.

[3] F. M. Xavier, M. Ferraz, N. Marc, N. U. Escosteguy, and E. H. Moriguchi, “Elderly people s definition of quality of life,” *Brazilian Journal of Psychiatry*, vol. 25, pp. 31–39, 2003.

[4] S. Vishnu, S. J. Ramson, and R. Jegan, “Internet of medical things (iomt) - an overview,” in *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*, 2020, pp. 101–104. doi: 10.1109/ICDCS48716.2020.243558.

[5] R. Doshi, N. Apthorpe, and N. Feamster, “Machine learning ddos detection for consumer internet of things devices,” in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 29–35. doi: 10.1109/SPW.2018.00013.

[6] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor, “Iot and agriculture data analysis for smart farm,” *Computers and Electronics in Agriculture*, vol. 156, pp. 467–474, 2019, issn: 0168-1699. doi: https://doi.org/10.1016/j.compag.2018.12.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169918308913.

[7] X. Krasniqi and E. Hajrizi, “Use of iot technology to drive the automotive industry from connected to full autonomous vehicles,” *IFAC-PapersOnLine*, vol. 49, no. 29, pp. 269–274, 2016, 17th IFAC Conference on International Stability, Technology and Culture TECIS 2016, issn: 2405-8963. doi: https://doi.org/10.1016/j.ifacol.

2016.11.078. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896316325162.

[8] A. Ahmed and G. Ahmed, "Benefits and challenges of internet of things for telecommunication networks," *Telecommunication Networks-Trends and Developments*, pp. 105–124, 2019.

[9] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless Personal Communications*, vol. 58, pp. 49–69, 1 May 2011, issn: 09296212. doi: 10.1007/S11277-011-0288-5.

[10] Hasa, *Difference between hypothesis and research question: Meaning, features, characteristics, usage*, Aug. 2016. [Online]. Available: https://pediaa.com/difference-between-hypothesis-and-research-question/.

[11] J. v. Brocke, A. Hevner, and A. Maedche, "Introduction to design science research," in Sep. 2020, pp. 1–13, isbn: 978-3-030-46780-7. doi: 10.1007/978-3-030-46781-4_1.

[12] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007. doi: 10.2753/MIS0742-1222240302. eprint: https://doi.org/10.2753/MIS0742-1222240302. [Online]. Available: https://doi.org/10.2753/MIS0742-1222240302.

[13] D. Moher, A. Liberati, J. Tetzlaff, *et al.*, "Preferred reporting items for systematic reviews and meta-analyses: The prisma statement," *PLoS Medicine*, vol. 6, no. 7, 2009, cited By 34924. doi: 10.1371/journal.pmed.1000097. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-68049122102%5C&doi=10.1371%5C%2fjournal.pmed.1000097%5C&partnerID=40%5C&md5=ae3854790b86846c4e4519473a3a3417.

[14] Y. L. Zheng, X. R. Ding, C. C. Y. Poon, *et al.*, "Unobtrusive sensing and wearable devices for health informatics," *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 1538–1554, 5 2014, issn: 15582531. doi: 10.1109/TBME.2014.2309951.

[15] H. Asada, P. Shaltis, A. Reisner, S. Rhee, and R. Hutchinson, "Mobile monitoring with wearable photoplethysmographic biosensors," *IEEE engineering in medicine*

and biology magazine : the quarterly magazine of the Engineering in Medicine & Biology Society, vol. 22, pp. 28–40, May 2003. doi: 10.1109/MEMB.2003.1213624.

[16] Y. Zheng, B. Leung, S. Sy, Y. Zhang, and C. Poon, "A clip-free eyeglasses-based wearable monitoring device for measuring photoplethysmograhic signals," cited By 24, 2012, pp. 5022–5025. doi: 10.1109/EMBC.2012.6347121. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84870852294%5C&doi=10.1109%5C%2fEMBC.2012.6347121%5C&partnerID=40%5C&md5=ac73796251a5e43af8af18ac81bd5

[17] Y.-H. Lin, C.-F. Lin, and H.-Z. You, "A driver's physiological monitoring system based on a wearable ppg sensor and a smartphone," *Communications in Computer and Information Science*, vol. 223 CCIS, pp. 326–335, 2011, cited By 15. doi: 10.1007/978-3-642-23948-9_36. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-80655146188%5C&doi=10.1007%5C%2f978-3-642-23948-9_36%5C&partnerID=40%5C&md5=c59e0615502c04eb213195de2e478104.

[18] E. Pinheiro, O. Postolache, and P. Girão, *Theory and developments in an unobtrusive cardiovascular system representation: Ballistocardiography*, 2010. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3111731/#R1.

[19] M. Nagura, Y. Mitsukura, T. Kishimoto, and M. Mimura, "A practical bcg measuring system with bed sensors and algorithm for heartbeat detection," in *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, 2018, pp. 317–321. doi: 10.1109/AMC.2019.8371110.

[20] J. Paalasmaa, H. Toivonen, and M. Partinen, "Adaptive heartbeat modeling for beat-to-beat heart rate measurement in ballistocardiograms," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 6, pp. 1945–1952, 2015. doi: 10.1109/JBHI.2014.2314144.

[21] L. P. Malasinghe, N. Ramzan, and K. Dahal, "Remote patient monitoring: A comprehensive study," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 57–76, 1 Jan. 2019, issn: 18685145. doi: 10.1007/S12652-017-0598-X.

[22] T. Shany, S. Redmond, M. Narayanan, and N. Lovell, "Sensors-based wearable systems for monitoring of human movement and falls," *IEEE Sensors Journal*, vol. 12,

no. 3, pp. 658–670, 2012, cited By 197. doi: 10.1109/JSEN.2011.2146246. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84856846340%5C&doi=10.1109%5C%2fJSEN.2011.2146246%5C&partnerID=40%5C&md5=5477ee5e67102e71b40bc608bd03bfdf.

[23] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, "Development of a wearable-sensor-based fall detection system," *International Journal of Telemedicine and Applications*, vol. 2015, 2015, cited By 134. doi: 10.1155/2015/576364. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84924370518%5C&doi=10.1155%5C%2f2015%5C%2f576364%5C&partnerID=40%5C&md5=9f3cbbe7c346b8f9c9a8edc23909b6

[24] K. Bierzynski, A. Escobar, and M. Eberl, "Cloud, fog and edge: Cooperation for the future?" In *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 62–67. doi: 10.1109/FMEC.2017.7946409.

[25] H.-J. Cha, H.-K. Yang, and Y.-J. Song, "A study on the design of fog computing architecture using sensor networks," *Sensors*, vol. 18, no. 11, 2018, issn: 1424-8220. doi: 10.3390/s18113633. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3633.

[26] I. Odun-Ayo, M. Ananya, F. Agono, and R. Goddy-Worlu, "Cloud computing architecture: A critical analysis," in *2018 18th International Conference on Computational Science and Applications (ICCSA)*, 2018, pp. 1–7. doi: 10.1109/ICCSA.2018.8439638.

[27] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 27–33. doi: 10.1109/AINA.2010.187.

[28] M. De Donno, K. Tange, and N. Dragoni, "Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog," *IEEE Access*, vol. 7, pp. 150 936–150 948, 2019. doi: 10.1109/ACCESS.2019.2947652.

[29] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019, issn: 0167-739X. doi: https://doi.org/10.1016/j.future.2019.02.050. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18319903.

[30] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017. doi: 10.1109/COMST.2017.2745201.

[31] S. J. Bigelow, *What is edge computing? everything you need to know*, Dec. 2021. [Online]. Available: https://searchdatacenter.techtarget.com/definition/edge-computing.

[32] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, “An overview of fog computing and its security issues,” *Concurrency and Computation: Practice and Experience*, vol. 28, no. 10, pp. 2991–3005, 2016. doi: https://doi.org/10.1002/cpe.3485. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.3485. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3485.

[33] M. Mukherjee, R. Matam, L. Shu, *et al.*, “Security and privacy in fog computing: Challenges,” *IEEE Access*, vol. 5, pp. 19 293–19 304, 2017. doi: 10.1109/ACCESS.2017.2749422.

[34] *Understanding edge computing vs. fog computing*. [Online]. Available: https://www.onlogic.com/company/io-hub/edge-computing-vs-fog-computing/.

[35] D. Soni and A. Makwana, “A survey on mqtt: A protocol of internet of things (iot),” in *International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017)*, vol. 20, 2017.

[36] A. Solovev, *What is the mqtt protocol and why choosing it for iot devices*, Jan. 2022. [Online]. Available: https://www.integrasources.com/blog/mqtt-protocol-iot-devices/.

[37] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, “Internet of things: Survey and open issues of mqtt protocol,” in *2017 International Conference on Engineering MIS (ICEMIS)*, 2017, pp. 1–6. doi: 10.1109/ICEMIS.2017.8273112.

[38] K. Grgić, I. Špeh, and I. Heđi, “A web-based iot solution for monitoring data using mqtt protocol,” in *2016 International Conference on Smart Systems and Technologies (SST)*, 2016, pp. 249–253. doi: 10.1109/SST.2016.7765668.