Wearable Monitoring System

Sandro Miguel de Sousa Ferreira

Thesis to obtain the Master of Science Degree in

Mestrado em Engenharia Informática e de Computadores

Supervisor: Prof. Renato Jorge Caleira Nunes

Examination Committee

Chairperson: Prof. Miguel Nuno Dias Alves Pupo Correia
Supervisor: Prof. Renato Jorge Caleira Nunes
Member of the Committee: Prof. Alberto Manuel Ramos da Cunha

December 2020

Abstract. As people get older, both the body and mind suffer from it, making them more vulnerable, dependent and prone to various types of age-related illnesses. This means that the need for more regular monitoring increases. With the advancement of technology, it is now possible to remotely assess a person's health 24 hours a day. This leads to more accurate monitoring which leads to better diagnosis and improved quality of life as the patient does not need to travel unnecessarily to a hospital or health clinic, resulting in lower costs to the patient and hospitals. In this work, several technologies and projects address this problem are presented and compared. Next, a low-cost solution for a wearable device that combines health monitoring and indoor localization using a network of nodes is proposed. This solution is intended to be used in a person's house or in a nursing home, it is easy to install, and its main goal is to improve the autonomy and safety of its users with the lowest cost possible, allowing them to age in their home and thus promoting a better quality of life. The developed solution was evaluated and proved to be effective in locating and monitoring the activity level of the person, and in analyzing and forwarding data from the sensors. The solution is modular and can easily be expanded in the future to incorporate more biomedical sensors.

Keywords: Wearable, health, monitoring, localization, low-cost.

Resumo. À medida que as pessoas envelhecem, tanto o corpo como a mente sofrem com isso, tornando-as mais vulneráveis, dependentes e sujeitas a vários tipos de doenças relacionadas à idade. Isso significa um aumento da necessidade de monitorização regular. Com o avanço da tecnologia, agora é possível avaliar remotamente a saúde de uma pessoa 24 horas por dia. Isto traduz-se numa monitorização mais precisa, que leva a um melhor diagnóstico e a melhor qualidade de vida, pois o paciente não precisa de se deslocar desnecessariamente para um hospital ou posto de saúde, resultando em custos mais baixos para o paciente e hospitais. Neste trabalho, diversas tecnologias e projetos que abordam este problema são apresentados e comparados. Em seguida, é proposta uma solução de baixo custo para um dispositivo vestível que combina monitorização de saúde e localização num espaço interior utilizando uma rede de nós. Esta solução destina-se a ser utilizada em casa de idosos ou em lares, é fácil de instalar, e tem como principal objetivo melhorar a autonomia e a segurança dos seus utilizadores com o menor custo possível, permitindo que eles envelheçam na sua habitação e, portanto, promovendo uma melhor qualidade de vida. A solução desenvolvida foi avaliada e mostrou ser eficaz na localização da pessoa, na monitorização do nível de atividade, e na análise e encaminhamento dos dados dos sensores. A solução é modular e pode ser facilmente expandida no futuro para incorporar mais sensores biomédicos.

Palavras-chave: vestível, saúde, monitorização, localização, low-cost

Index

List of Tables

List of Figures

List of Abbreviations

ADC	Analog-to-Digital Converter
AM	Acceleration Magnitude
AoA	Angle of arrival
API	Application Programming Interface
BLE	Bluetooth Low Energy
BMI	Body Mass Index
BPM	Beats per Minute
BS	Base Stations
CoO	Cell of origin
DMP	Digital Motion Processor
DoF	Degrees of Freedom
ECG	Electrocardiogram
GFSK	Gaussian Frequency-Shift Keying
GPS	Global Positioning System
GSM	Global System for Mobile communications
GUI	Graphical User Interface
HR	Heart Rate
I2C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IFTTT	If This, Then That
IMU	Inertial Measurement Unit
IoT	Internet of things
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
LCD	Liquid Crystal Display
LT	Lower Threshold
MS	Mobile Stations
MSC	Mobile Switching Centers
OC	Orientation Change
OMR	Object-Relational Mapper
PCB	Printed Circuit Board
REST	Representational State Transfer
RPD	Received Power Detector
RSSI	Received Signal Strength Indicator
SIM	Subscriber Identification Module
SPI	Serial Peripheral Interface

UT	Upper Threshold
URL	Uniform Resource Locator
Wi-Fi	Wireless fidelity
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

1 Introduction

1.1 Motivation

Human aging is inevitable. A person's body and mind will also age and this leads to a large number of people who are more debilitated, dependent and with age-related illnesses. This creates a growing need for constant monitoring and care. Older people tend to forget about taking medication, are more likely to fall, have difficulty moving, have health problems that need to be checked regularly and dementia is also a problem in this age group. These people are often unable to live by themselves and therefore, must live in a nursing home or with their family. With the advances in technology, there are now many systems that can perform telemedicine and health monitoring. These intelligent systems are capable of monitoring a person at home with quality and reliability, which helps prevent frailty and the worsening of diseases. Monitoring health in the comfort of our home is very useful and can have a lot of advantages. When a person goes to the hospital or medical center to perform medical exams, those exams might not return the best results because the patient could be too stressed and nervous. If a person is being monitored while doing their daily life, the results are more accurate, and the patient does not have to waste a day going to the hospital and to stay in an uncomfortable environment. With the *Internet of Things* (IoT), the patient's information can be quickly sent to a doctor to evaluate the data. With this type of systems, the need for hospitalization can be lessened by the right diagnosis, and time and money are saved for both the patient and the medical facilities.

However, the existing solutions have some negative characteristics. Usually, these systems have a high cost, the autonomy is not the best and are closed products. This means that these systems do not allow to modify or add new sensors and features.

1.2 Objectives

This work proposes a low-cost system that uses a wearable device that can monitor the vital signs of one or more users in an indoor environment. The device uses sensors such as heart rate and an accelerometer to detect abnormal situations and generate alarms related to falls and inactivity. Combined to health monitoring, the system also performs room-level localization by having a network of nodes that communicate by radio frequency inside the desired indoor facility. All the information about the users is sent wirelessly through the network until it reaches the central node that processes and stores the data for later consultation on a web application created for that purpose. All the network and user information can be configured through the web application and all the alerts are configurable for each user. This is all performed in a way that is neither invasive nor uncomfortable to the user. This combination of indoor localization and health monitoring is expected to have a good impact on the user's health diagnosis. It is intended to be used at the user's house or in nursing homes, and it is capable of monitoring several users at the same time.

The main goal is to demonstrate that the low-cost solution proposed in this work, has the potential to be used as a real life health monitoring and indoor localization system. The total system cost must be as low as possible without losing the main features required. The system will monitor the users every day without restricting their daily life, which makes them feel more comfortable by knowing that their health and physical status is constantly being monitored, contributing to their wellbeing and increasing their quality of life.

1.3 Document Structure

The rest of this document is organized as follows: chapter 2 includes related work about communication protocols, indoor localization, fall detection and health monitoring. Chapter 3 presents the proposed health monitoring and localization system. Chapter 4 includes the more technical details about the development of the prototype system. Chapter 5 presents the solution challenges, evaluation and future improvements. And finally, the conclusions are presented in chapter 6.

2 Related Work

2.1 Indoor Localization Techniques

Indoor localization [1] can be defined as any system that can determine the position of something or someone inside of a closed structure continuously and in real time. In order to achieve a reasonably good accuracy, the system needs to be able to handle several problems such as the interference of obstacles like walls, objects and the movement of human beings, and also the multipath effect. The multipath effect occurs when the radio signals are reflected by obstacles and reach the receiving radio by two or more different paths. The performance of a localization system can be evaluated according to the accuracy, responsiveness, coverage, adaptiveness, scalability, cost and complexity of that system. Indoor localization techniques can be divided into three categories: proximity, triangulation and scene analysis.

Proximity detection is the simplest to implement. The position of the mobile node is determined by cell of origin (CoO) method. When the mobile node is detected by more than one beacon, the node's position belongs to the beacon with the strongest signal received.

Triangulation/trilateration use triangles geometry to determine the target location. Techniques that are based on transmission time and on signal property belong to the lateration category (trilateration). The techniques based on the angle of arrival (AoA) belong to the angulation category (triangulation).

Scene analysis can be divided in two techniques: Dead Reckoning and map matching. Dead reckoning estimates the current position based on the last determined position and increments that position based on known or estimated speeds over the time. The major disadvantage of this technique is that the inaccuracy is cumulative because the new position is estimated from the previous one. This results in error position growing over time. Map matching is based on pattern recognition combining a digital map with locating information to obtain the real position of the target. With this method, there is no need to install additional hardware.

2.2 Communication Protocols and Radios

There are many different types of RF protocols available with various advantages and disadvantages. Three important characteristics to consider when choosing a wireless communication method are data rate, range and power consumption.

Wi-Fi - Wireless fidelity (Wi-Fi) [2] includes IEEE (Institute of Electrical and Electronics Engineers) 802.11 standards for wireless local area networks (WLAN). It allows users to surf the internet at broadband speeds when connected to an access point or in ad hoc mode. The signal range is around 100m.

Wi-Fi has a very high data rate but also has high power consumption. A popular Wi-Fi module is ESP8266 [3] and costs around 6 euros.

ZigBee - This protocol [1] is based on IEEE 802.15.4 and it is one of the standards in terms of wireless sensor networks. It supports three types of communication topologies such as point-to-point, point-to-multipoint and mesh topology. It provides a wide coverage area from being capable of multi-hop communication. This module is often used in embedded applications that require low power consumption and low data rates such as home automation and medical data for example. The signal range coverage of a ZigBee in indoor environments is typically 20m to 30m. Distance calculation between two ZigBee nodes is usually carried out from RSSI (Received Signal Strength Indication) values. ZigBee is open to interference from a wide range of signal types using the same frequency which can disrupt radio communication because it operates in the unlicensed ISM (Industrial, Scientific and Medical) bands. The XBee [4] module is a low power radio that allows a microcontroller like an Arduino to communicate wirelessly using Zigbee and costs around 20 euros.

Bluetooth – Bluetooth [1,2] is a wireless standard for wireless personal area networks (WPANs). Almost every Wi-Fi enabled mobile device, such as mobile phone or computer, also has an embedded Bluetooth module. It operates in the 2.4 GHz ISM band. The signal range is around 10m. The benefit of using Bluetooth for exchanging information between devices is that this technology offers high security, low cost, low power and small size. One of the drawbacks of using Bluetooth in localization systems is that, after each location found, it needs to run the device discovery procedure again. This significantly increases latency and the power consumption.

Bluetooth Low Energy – BLE [5] is a wireless personal area network technology. This technology is aimed at areas such as healthcare, fitness, beacons, security and home entertainment industries. Compared to classic Bluetooth, is intended to maintain a similar communication range while providing a reduced power consumption and cost. BLE has been designed as low-power solution for control and monitoring applications. Its range is much shorter than ZigBee but has a much higher data rate. HM-10 [6] is a popular BLE module to work with Arduino and it costs around 10 euros.

GSM – The Global System for Mobile communications [7,8] is the standard for digital mobile communications. GSM technology provides wide range communication so that it can be used from any corner of the world. GSM architecture consists of mobile stations (MS) and base stations (BS) which communicate with each other through radio links. Base stations are connected to the mobile switching centers (MSC) which are responsible for routing the signals to and from the fixed networks. Several databases are defined to perform management and authentication purposes. It has a comprehensive set of security features. However, these modules have some limitations to consider. It is necessary to have a SIM card and the messages or data that is transmitted have a cost associated. SIM900A [9] is a popular GSM module and costs around 10 euros.

CC1101 radio module - CC1101 [10] is a low-cost transceiver designed for low-power wireless applications. It is intended for the ISM and SRD (Short Range Device) frequency bands at 315, 433, 868 and 915 MHz. It also can be programmed for operation in the 300-348 MHz, 387-464 MHz and 779-928 MHz bands.

The radio has a configurable baseband modem that supports several modulation formats, and the data rate can be configured up to 600 kbps. CC1101 provides hardware support for packet handling, data buffering, clear channel assessment, link quality indication, burst transmissions and wake-on-radio. The cost of this radio module is around 4 euros and the transmission range is around 200m to 500m. However, some tests indicate that the actual range can be around 40m to 50m [11].

nRF24L01+ radio module - The nRf24L01+ [12] is a single chip designed to operate in 2.4 GHz worldwide ISM frequency band and uses GFSK modulation for data transmission. This radio is one of the most inexpensive data communication options that can be found (less than 1 euro), making it suitable for ultra-low power wireless applications. The signal range is around 100 meters in open space. If used inside a building the range should be less than 30 meters.

It has parameters that can be configured, such as the channel frequency, output power and air data rate. The air data rate can be one of 250 kpbs, 1Mbps and 2 Mpbs. The operating voltage lies between 1.9 to 3.6V and the logic pins are 5-V tolerant. This means that the radio can easily be connected to any microcontroller like an Arduino without the need to use a logic lever converter. Table 1 shows the nRF24L01+ specifications.

Frequency Range	2.4 GHz ISM Band
Maximum Air Data Rate	2 Mb/s
Modulation Format	GFSK
Max. Output Power	0 dBm
Operating Supply Voltage	1.9 V to 3.6 V
Max. Operating Current	13.5mA
Min. Current(Standby Mode)	26μA
Logic Inputs	5V Tolerant

Table 1. nRF24L01+ transceiver module specification

This radio provides a built-in feature called Multiceiver (Multiples transmitters single receiver). A module can act as a hub and communicate with 6 different transmitters in parallel. This hub can also stop listening and start to act as a transmitter. This gives the opportunity to easily create a network.

The nRF24L01+ transceiver module uses a packet structure known as Enhanced ShockBurst. This structure has a lot of advantages: it allows for variable length payloads (1 to 31 bytes) and it provides each sent packet with an ID, which allows to check whether a message is new or needs to be retransmitted. Each message can request an acknowledgement to be sent when it is received by another device. The transceiver supports automatic packet handling, and it is done entirely by the chip, there is no involvement of the microcontroller.

Comparison - Table 2 displays a comparison between all the devices discussed above regarding consumption power, data rate, range and average price:

	Consumption	Data Rate	Range	Price
ESP8266	170mA	54 Mbps	100m	6€
XBEE	45mA	250 kbps	20m-30m	20€
HM-10	50mA	2 Mbps	10m	10€
SIM900A	500mA	9.6 Mbps	Global	10€
CC1101	14.7mA	600kbps	40m-50m	4€
NRF24L01+	13.5mA	2 Mbps	30m	0.8€

Table 2. Comparison of different radio modules regarding consumption, data rate, range and price

2.3 Fall Detection

As people age, their physical and psychological capacity gets worse, which leads to a greater likelihood of falls. According to the World Health Organization [13], approximately 28-35% of people aged of 65 and over fall each year and the frequency of those falls increases exponentially with age and frailty level. The major causes for hospital admission related to falls are hip fracture, traumatic brain injuries and upper limb injuries. Falls may also result in a post-fall syndrome that includes dependence, loss of autonomy, confusion, immobilization and depression. The longer a fall goes undetected, less likely are the chances of the patient making a full recovery.

Taking all this information into account, the development of fall detection systems for healthcare is a very active area of research. There are basically two types of fall detection systems: context-aware systems and wearable devices (including smartphones) [14]. The context-aware systems are based on cameras, floor sensors, infrared sensors, microphones, and pressure sensors. Although being an area that is widely studied, there is no standardized context-aware technique accepted by the researchers in this field. Wearable-based solutions can combine different sensors such as accelerometers, gyroscopes, inertial sensors and barometers. But the accelerometers are the most used option within the literature [15]. Most of the fall detection techniques based on wearable devices are machine learning (pattern matching) techniques or fixed thresholds techniques [14]. Since the devices compare the sensor values measure and only compare them with a threshold, the accuracy and sensitivity are quite low and it increases the number of false positives. Therefore, the use of machine learning algorithms for predicting falls has become a trend since it results in increased prediction accuracy.

2.3.1 Dynamics of a Fall

In order to detect a fall based on acceleration, the magnitude of acceleration needs to be calculated using the 3 axis accelerometer values (equation 1).

$$\hat{a} = \sqrt[2]{a_x^2 + a_y^2 + a_z^2} \quad (1)$$

A fall event can occur in different directions but usually the acceleration pattern is the same. A typical fall can be divided into 3 sections: the free fall period, the impact period and the inactivity period:

Free fall period - In this period, the subject starts descending towards the ground and the acceleration is less than 1g and tending to 0g.

Impact period – After descending, the subject eventually impacts the ground or other objects and the acceleration rapidly increases to values greater than 3g (2.5g - 3.5g is usually the range used in fall detection systems) [16].

Inactivity period – After the impact, the subject usually remains almost motionless for a very short amount of time. This happens even if the fall event does not have any serious consequences. After the fall, the subject may become unconscious or start moving and trying to get up.

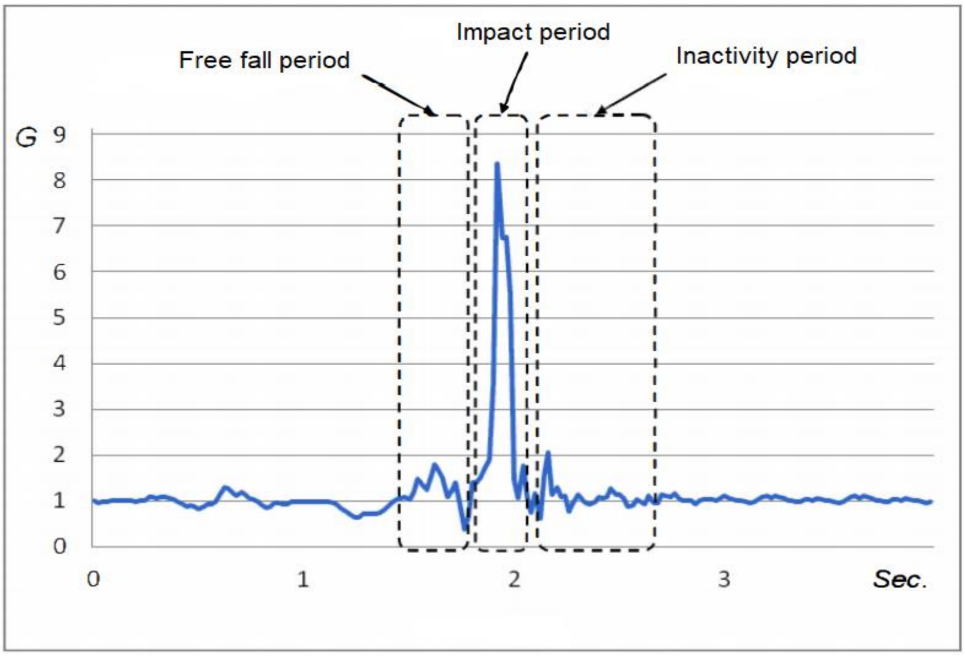

Fig. 1. The three phases of a fall event [17]

2.3.2 Public Datasets

Creating datasets with simulated falls and some usual activities of daily living can help developing better fall detection systems. Most of these datasets are created by healthy young individuals that simulate various types of falls and with a wearable device usually placed on the chest or on the waist.
However, the UMAFall dataset [18] has data from simulated falls with the sensors placed on different regions such as ankle, waist, wrist and head. This dataset is relevant for the proposed system in this work because it has values of acceleration with the sensors placed on the wrist.
The FARSEEING dataset [19] is also very interesting because unlike the other datasets that have data from young individuals simulating falls, the FARSEEING dataset has data gathered by real falls that occurred in older people. The data is gathered from 22 real falls recorded by 15 elderly people with the sensors placed either on the lower back or on the tight. This dataset can therefore assess the veracity of simulated falls comparing them to real falls. In [15], a comparison between these two datasets was made and the results showed a clear difference between the real and simulated falls. This means that the results with simulated falls often cannot validate real cases.

2.3.3 Fall Detection Solutions

The common characteristic of most wearable solutions is that the location of the wearable device is on the chest or on the waist. This is because those regions of the human body are the easier ones to detect a fall event [15,20]. Despite this advantage, it is difficult to hide a wearable device on those positions and it often causes discomfort to the user while performing their daily activities. Due to this lack of usability, users often forget to wear those devices. There has been a lot of research for fall detection systems placed on the wrist because it is an area of the body that does not cause discomfort to the user and can be used just like a regular watch. However, the wrist is probably one of the most difficult places to detect a fall, because people use their hands all the time when performing their daily activities, which means that the accelerations measured are usually greater than if they were measured at the chest or waist regions.

In [21], an easy to use wrist-worn fall detector called Speedy was developed. After a heavy fall, the device detects it and alerts a call center. It can emit an alert even if the user is unconscious or too agitated to ask for help by himself. The system has an accelerometer and uses 3 thresholds to detect a fall. Those thresholds are the acceleration magnitude, the current speed and the vertical speed. The system does not measure the orientation of the user and this is the main problem with this solution. If the user rotates during a fall, there could be many errors in the threshold calculations, and this can result in a poor fall detection.

[22] proposed a solution also based on fixed thresholds but it also adds a threshold for orientation change retrieved from the gyroscope. There is a trigger that activates if the magnitude of acceleration is lower than the lower threshold. After that, If the acceleration goes higher than the upper threshold, another trigger is activated. If the person’s orientation changes within 0.5 seconds of activation of trigger

two, and the orientation change remains for 10 seconds (indicating the person is immobilized), a fall is detected. If any of those triggers fails to occur, all the triggers are reset, and the acceleration readings are taken as normal. It was also implemented a false alarm button that can be pressed within 5 seconds after a fall is detected and a button to ask for help. This solution has also a GPS module that sends the location of the user when an alert is emitted.

In [15], a different approach was developed. The authors also designed a wrist-worn device but they used machine learning for the detection. The project is based on [16] where the author proposed the dynamics of a fall seen in figure 2 in order to detect a fall.

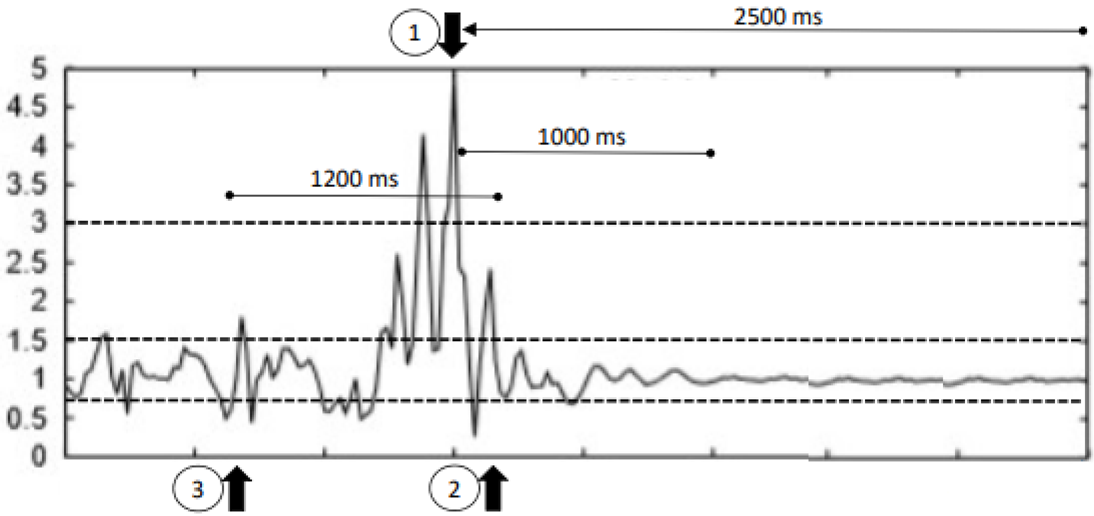

Fig. 2. Graph from [15] based on [16] and it shows the evolution of the magnitude of the acceleration. The X-axis represents the time, and each mark corresponds to 500 ms

[16] proposes that in order to find the peak at peak time pt = t – 2500 (point 1), there must be no other peaks with higher acceleration and that peak acceleration must be higher than 3g. If this holds for 2500ms than the system recognizes a peak. The impact end (ie) (point 2) is the last time the acceleration is higher than 1.5g. The impact start (is) (point 3) is the first sequence of the acceleration is lower than 0.8g and then goes up to values higher than 1.5g. The impact start must belong to the interval [ie-1200ms, pt]. If no impact end is found, it is fixed to pt + 1000ms and if no impact start is found, it is fixed to pt. After calculating these three times, the system can start calculating eight features [14, 15]:

- Average Absolute Acceleration Magnitude Variation, AAMV = $\sum_{t=is}^{ie} \frac{|a_{t+1}-a_t|}{N}$, with N the number of samples in the interval.
- Impact Duration Index, IDI = ie – is.
- Maximum Peak Index, MPI = $max_{t\in[is,ie]}(a_t)$.
- Minimum Valley Index, MVI = $min_{t\in[is-500,ie]}(a_t)$.

- Peak Duration Index, PDI = pe – ps, with ps the peak start defined as the last magnitude below 1.8g occurring before pt and pe is the peak end defined as the first time the magnitude was below 1.8g occurring after pt.
- Activity Ratio Index, ARI, the ratio between the number of samples not in [0.85g, 1.3g] and the total number of samples in the 700ms interval centered in (is + ie)/2.
- Free Fall Index, FFI, the average magnitude in the interval [t_{FFI},pt]. t_{FFI} is the time between the first magnitude below 0.8g occurring up to 200ms before pt; if not found it is set to pt – 200ms.
- Step Count Index, SCI, number of peaks in the interval [pt – 2200, pt].

After the calculations of those features, the eight values are used as input to a neural network. [15] performed threshold optimizations but the results did not show a clear difference to the results from [16]. Despite the good results, they used simulated falls data for the training process. A dataset with wrist data from real falls is needed in order to improve and validate the solution. Machine learning solutions also have a problem when being used in wearable devices because they can consume much more energy compared to fixed threshold solutions.

Apple Watch [23] is a smart watch produced by Apple and it has the capability of detecting falls. When a hard fall is detected by the Apple Watch, an alert appears and allows the user to easily call emergency services or dismiss the alert. If the user is unresponsive for about a minute, a 30-second countdown will begin, and a sounding alarm will start. The alert gets louder, so that someone nearby can hear it. When the countdown ends, an emergency call will be placed automatically, and a message will be sent to the user's emergency contacts. The falls are automatically recorded, and the fall history of the user can be checked. This feature is automatically enabled for users that are 55 years and older and can be turned on for anyone in the Apple Watch app.

2.4 Health Monitoring Systems

2.4.1 AMON

AMON [24] is a wearable medical computer for high risk patients. The system is a watch-like housing mounted on a wrist blood pressure cuff. It is built for high risk patients that require constant monitoring of their vital signs. It continuously monitors and logs, pulse, blood oxygen saturation and the patient's temperature. The level of physical activity is measured by an acceleration sensor. When necessary, blood pressure and electrocardiogram (ECG) can also be taken. The system performs an online analysis of the user's health status and if the results are reasonably different from a predefined user specific range, additional measurements such as ECG and blood pressure are required. If those additional measurements confirm the deviations, the system automatically alerts a doctor. A manual alert can be triggered whenever the patient feels that he needs assistance. All the data logged in the previous hours is sent to the medical center. The medical personnel can communicate with the patient using the simple

LCD display of the device and then can instruct it to perform new measurements. For communication, the system uses a Siemens TC 35 GSM module. The battery can sustain the device for about 24h without recharging.

This system communicates using GSM technology. By using a digital cellular network, high risk patients are allowed to be more mobile. By wearing a small device, their vital signs can be monitored without restrictions and discomfort wherever they are. The main drawbacks of AMON are the cost, limited flexibility and autonomy. The 24h autonomy is not the ideal because the proposed project's objective is to function without maintenance for a much longer time. Also, the cost of that system is not low, due to the fact that it uses a GSM module and an LCD display that substantially increase the price.

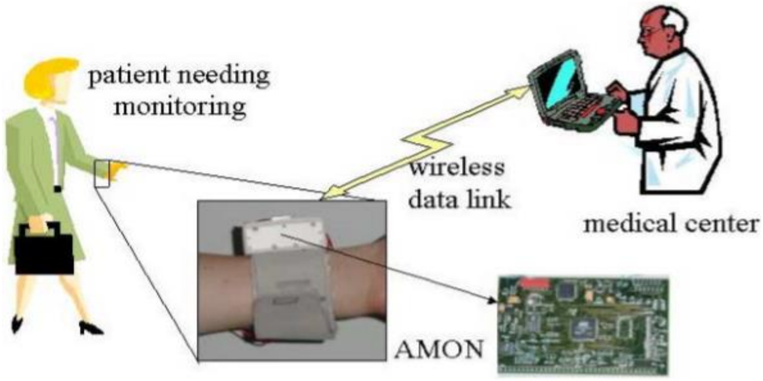

Fig. 3. Prototype of the AMON and its role in the monitoring system

2.4.2 MySignals

MySignals [25] is a development platform for medical devices and eHealth applications. It allows more than 20 biometric sensors such as pulse, breath rate, oxygen in blood, electrocardiogram signals, blood pressure, muscle electromyography signals, glucose levels, galvanic skin response, lung capacity, snore waves, patient position, airflow and body scale parameters (weight, bone, mass, body fat, muscle mass, body water, visceral fat, Basal Metabolic Rate and Body Mass Index). This makes MySignals one of the most complete eHealth platforms in the market. All the data gathered by MySignals is encrypted and sent to the user's private account in a Cloud API. The information can be sent using Wi-Fi or Bluetooth Low Energy 4.0. The data can be visualized in a tablet or smartphone with Android or iPhone Apps and in a TFT screen incorporated on the platform. It also features an alarm/emergency button that provides immediate assistance by just pressing the button. It can be worn as a help pendant around the patient's neck or as an alert watch button around the wrist. Regarding security, it provides SSL secure HTTPS communications to a cloud. This security layer is added on top of the symmetric encryption AES 256 that is performed by MySignals in the application layer.

The architecture used in the system is made by the Libelium company. The microprocessor is an Atmega 2560 and has BLE and Wi-Fi integrated. There is an open-source version of this system. The architecture is compatible with Arduino, the microprocessor is an Atmega 328 (Arduino Uno) and has extra radios such as Bluetooth, ZigBee, 4G, 3G and GPRS (General Packet Radio Service).

MySignals is a very interesting and innovative project. In terms of health monitoring is one of the most complete solutions that exists and is very beneficial in the lives of the people who use it. However, this system's portability, autonomy and cost are not the ideal for the type of solution being proposed in this document. The device has a size that does not allow to be wearable and it needs to be plugged into a power outlet. Finally, the cost of this system is above 1000 euros.

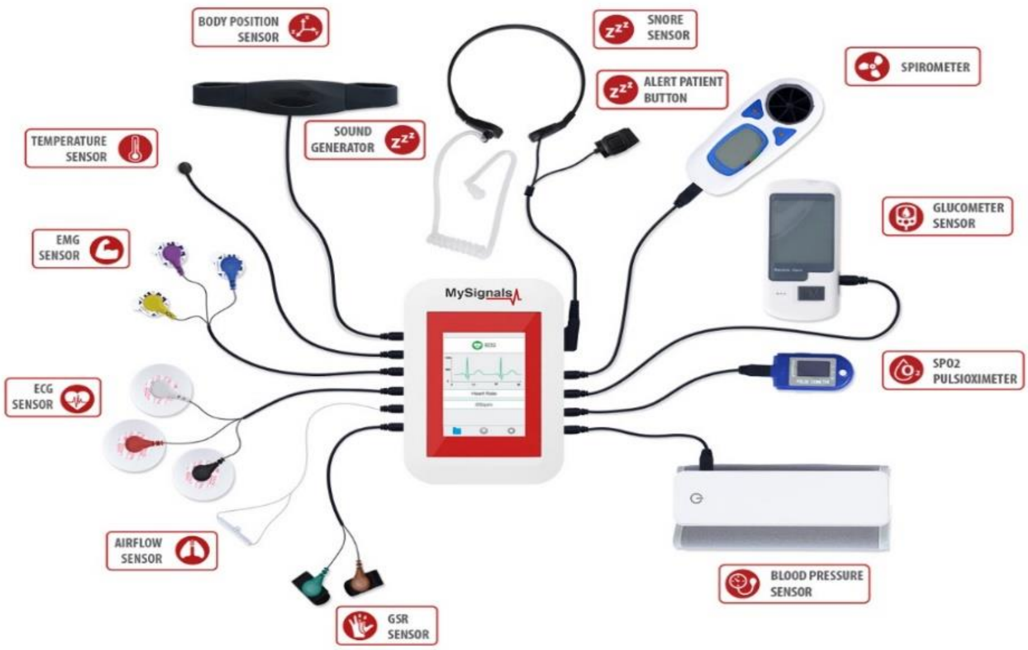

Fig. 4. MySignals software development platform with sensors

2.4.3 HealthBand

HealthBand [26] is a project that detects and locates a person whose health status is in danger. It is designed as a remote rescue system for people who are at risk of having a stroke, cardiac arrest and heart attack. A smartphone is synced with a wearable health monitoring bracelet that can read the user's vital signs such as pulse rate and body temperature. If the bracelet detects life threatening vital readings, the phone synced to the bracelet will automatically call the family to prompt the person's health status. The wearable device contains an ATmega328 chip acting as a standalone Arduino that is connected to a body temperature sensor, a pulse rate sensor and a blood pressure sensor. For communication, the system uses the HC-05 Bluetooth module. The power source is a 3.7v (1000mAh) lithium battery and it can power the device for about 2 days. The device communicates with a mobile app that shows the information gathered by the sensors and the data is refreshed every 30 seconds.

This system was made by a teenager for google science fair. It is not the most technical, detailed and professional solution, but regarding health monitoring, is the project that comes closest to the solution proposed in this document since it has a wearable device and was developed with low cost components.

Fig. 5. HealthBand wearable device and its app application

2.4.4 FrailSafe

Demographic ageing in Europe will increase dramatically over the coming years. Frailty is considered as one of most important issues associated with ageing. As people get older, both their physical and mental conditions get worse and they start to lose independence and quality of life. With some management and early detection, frailty can be delayed or even postponed. This is the main goal of the European Union's FrailSafe Project [27]. This project consists of an integrated assessment system that can estimate the patient's frailty level, provides health monitoring, indoor and outdoor localization, games for cognitive stimulation and generates notifications in case of adverse events. With all the data collected, the system generates a virtual patient model that can be accessed by a health care professional and authorized family members through the platform's dashboard. This dashboard displays the patient's current health condition and suggestions to preventive frailty interventions. The patient's information can be compared with other people with the same age and frailty level.

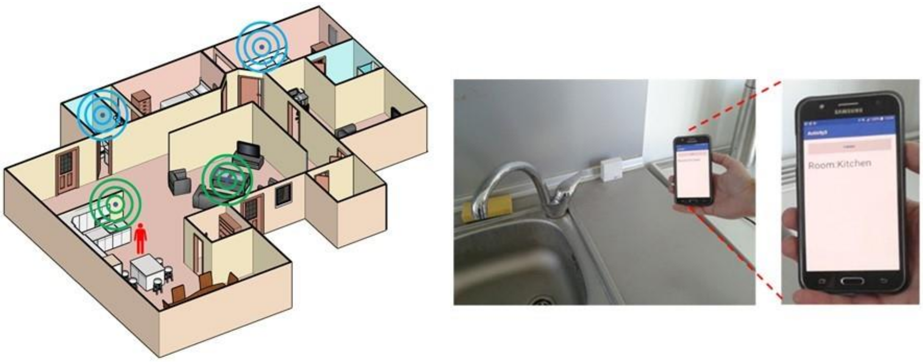

Fig. 6. FrailSafe's indoor localization system overview

Indoor localization - Capturing the indoor movements of the older person around the house can indicate his/her health condition. A person can have difficulty to move due to pain, frequently visits the toilet or have fallen. The indoor localization system of this project has the objective to detect those problems. The system is based in Bluetooth beacons. Several of those beacons are positioned in the older person's home and constantly emit a unique identifier (ID). The device that receives that information is a smartphone or a smartwatch that must be carried by the patient all the time. The mobile device receives the signal from the beacons and estimates the person's current room by calculating the distance between itself and the beacon. This is possible because those beacons use Bluetooth's Received Signal Strength (RSS). In order to locate properly, a training phase needs to be performed. The beacons need to be assigned to their corresponding rooms and the clinical person needs to move randomly in each room for about half a minute. This procedure is necessary because the system needs to learn how the signal strength differs in different rooms and positions.

In these articles [28,29], there is more detail about the system. One of the main focus is to non-technical staff being able to install the system in multiple houses with ease and in a short amount of time. A smartphone was chosen as a tracking device because any smartphone in the market can integrate Bluetooth Low Energy and cloud functionalities, which is necessary to localization and to collect data from multiple users. Finally, a GUI (Graphical User Interface) was implemented to help with the setup procedure of the beacons from non-technical staff. The beacons are small devices that use Bluetooth Low Energy, and their function is to broadcast, between small time intervals, messages that contain their ID. They are placed in appropriate positions in the house, so that they are at least 2 m from each other. The reason for keeping this distance restriction between the beacons is to sufficiently discriminate the rooms with the collected RSS fingerprints. The main reasons that the system is based on the specific hardware choices are the low cost and the availability of the involved devices (beacons, smartphone/smartwatch). The mobile device continuously scans those messages and measures the RSS value received from them. The RSS value is then compared to a set of RSS values (fingerprints) collected during the setup phase and after that, the person's current position can be calculated. The advantage of using fingerprints is that there is no need to transform the RSS value to distance because it depends on the topology of each house and requires a longer setup procedure in order to collect different RSS measurements in various distances from the beacon. In order to capture relevant movement patterns, the patient needs to carry the mobile device for several days to allow a good analysis. The data collected from this indoor application are uploaded to the FrailSafe cloud repositories to be further analyzed by other FrailSafe components. The beacons used were Sensoro beacons [30]. Those beacons currently cost around 15 euros and are certified by Apple IBeacon. In [28,29], 5 to 8 Sensoro beacons were used inside a house to perform the tests. In this system's architecture the beacons do not form a network, they are just placed in certain areas of the house and it is necessary to pair with all those beacons. Also, the beacons are the ones that ping their ID.

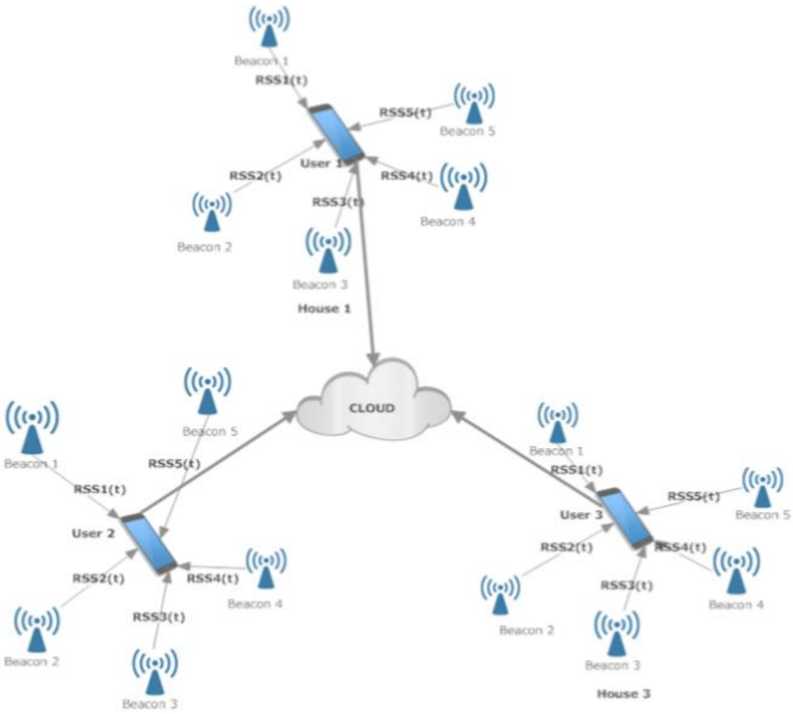

Fig. 7. FrailSafe's indoor localization system architecture

Health monitoring – Regarding health monitoring, FrailSafe project uses a Smartvest developed by a partner company called Smartex [31]. The Smartvest prototype is a "Wearable WBAN – Wireless Body Area Network System" . The system is composed of a sensorized garment, an electronic device and a software tool for visualization of streaming data or downloading the recorded data from the electronic device to a PC and then uploaded to a cloud service. The sensorized garment is a shirt with short sleeves that has two fabric electrodes for electrocardiogram (ECG) monitoring, a fabric piezoresistive sensor for respiration monitoring on the chest, and two small boxes with a 9-Degrees of Freedom (DoF) Inertial Measurement Unit (IMU) sensor that is placed in each sleeve. There is also a pocket for the electronic device with another integrated 9-DoF IMU sensor. All sensors are connected via cables to the device and collects the data from them. After that, a microprocessor elaborates several parameters, and all the raw and processed data are saved on a micro-SD card. That data can be transmitted by Bluetooth to a computer or an Android device for real time data analysis.

The Smartvest will monitor the following parameters:

- Electrocardiogram (ECG)
- RR Standard Deviation (SDNN)
- Posture
- Heart Rate
- Respiration signal
- Activity classification

- RR (distance in milliseconds between 2 QRS complexes)
- Respiration Rate
- Step counter

This garment has been developed to collect several types of information while reducing the discomfort to a minimum. It uses natural material such as cotton to make it breathable and cool. Because the fabric sensors need to be placed in tight connection with the user's thorax and the garment is made to be used by people with different sizes, it has a Velcro strap on the back side of the shirt, in order to adapt the vest to every user's thorax circumference. The main drawback of this system is that the device inside the shirt needs to be charged every day.

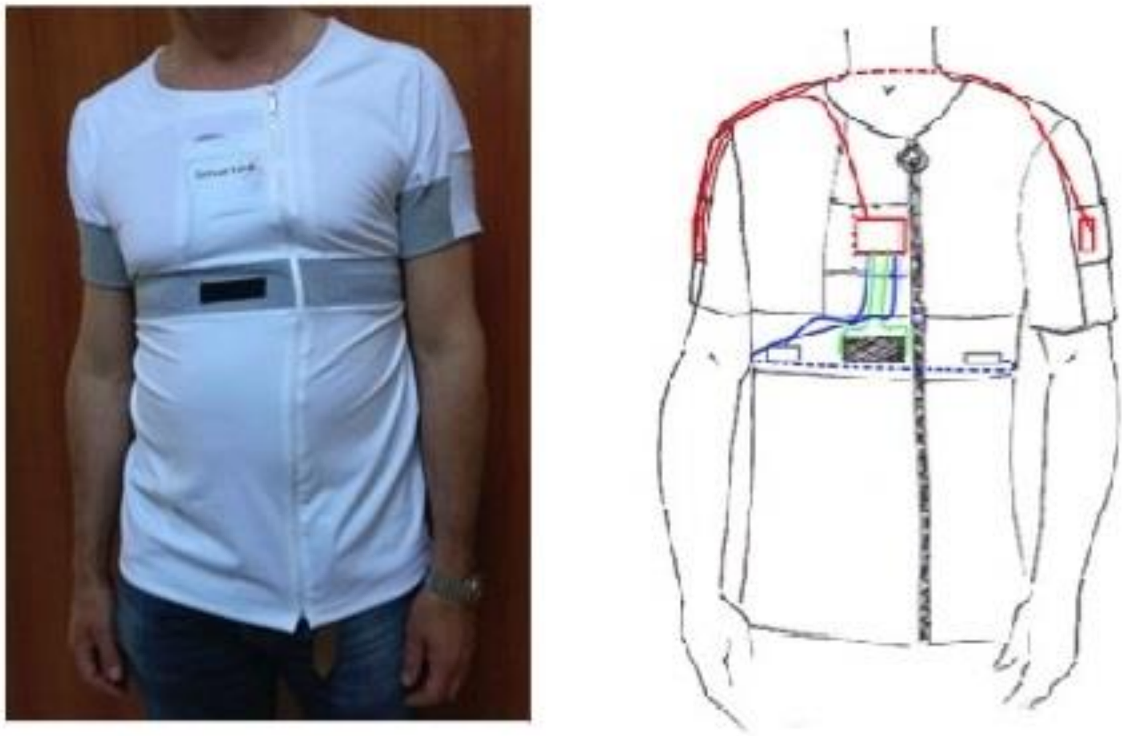

Fig. 8. FrailSafe's wearable monitoring shirt

2.4.5 Other Health Monitoring Systems

Two very similar projects are now presented. These two systems are not products, they are smart health monitoring system prototypes, and both use an Arduino as microcontroller.

The project "Smart health monitoring system" described in [32] proposes a system where several biomedical sensors are interfaced with Arduino UNO microcontroller and get the information from the sensors. The sensors used in this project are temperature and heartbeat sensors. The data is saved in an SD card and then, is sent to a server and to a mobile app wirelessly. The android application developed has a medicine reminder, shows nearby hospitals, shows health care tips and home remedies, shows the sensor data, and has a BMI (Body Max Index) calculator. The medicine reminder feature reminds patients of their dosage timings through an alarm ringing system.

The project “Health Monitoring System using internet of things” presented in [33], is a similar solution where an Arduino Mega with more sensors monitors not only the body temperature and heartbeat, but also respiration, blood pressure and performs ECG. The data is then automatically uploaded to a website by a cloud server called “ThingSpeak”, where the patient’s health status can be monitored easily. The system has also an LCD display implemented that shows the same information. For communication it uses a WIFI module (ESP8266).

2.5 Existing Solutions Comparison

MySignals is the only solution that is a finished product and is one of the most complete products in the market, the other solutions are still prototypes that are under development. FrailSafe is a big project funded by the European Union and is the only solution that also has a system that can perform room-level indoor localization. HealthBand, AMON and FrailSafe present solutions for a wearable device. Both HealthBand and AMON projects propose a wearable device to be used on the wrist, instead, FrailSafe project proposes a shirt as a wearable device (SmartVest). In terms of cost, HealthBand, [32] and [33] projects are the cheapest solutions. Both FrailSafe and MySignals are the most expensive ones.

Table 3 presents a comparison of the existing solutions regarding the communication method, whether the solution has a localization system, whether it proposes a wearable device and the qualitative cost of the system (low, medium or high). These solutions do not satisfy the requisites and objectives defined for document’s work. In the next chapter, it is presented a solution for a low-cost system that performs health monitoring and also indoor localization of several users at the same time using mobile devices and a network of nodes.

	Communication	Localization	Wearable	Cost
AMON	GMS	No	Yes	Medium
MySignals	BLE, Wi-Fi	No	No	High
HealthBand	BLE	No	Yes	Low
FrailSafe	BLE	Yes	Yes	High
[32], [33]	Wi-Fi	No	No	Low

Table 3. Comparison between the existing solutions

3 Proposal

The proposed solution can be divided in two different segments that work together: indoor localization and health monitoring. The goal is to develop a wearable device and a system where one or more people can be localized inside an indoor space and be able to monitor those people regarding basic health characteristics with the lowest cost possible.

3.1 Solution Architecture

The architecture of the proposed solution is composed by several components that interact with each other. It includes a network of fixed and mobile nodes based on the Arduino Nano microcontroller. The mobile nodes are attached to the users and periodically send all the information gathered by their incorporated sensors to the network. The fixed nodes send the data received from the mobile nodes through the network until it reaches the central node.

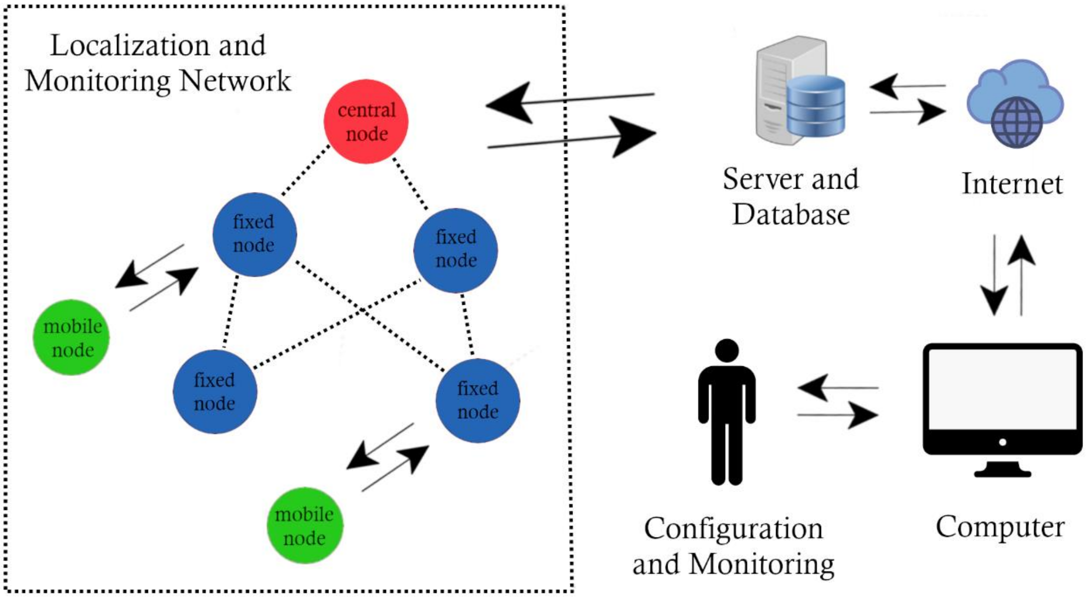

Fig. 9. Overall architecture of the proposed system

The central node collects all that data and sends it through a serial channel to the local server, which is a device with higher memory and processing power such as a computer or a Raspberry Pi. This proposal chooses the Raspberry Pi as the preferred device to be the local server due to its lower cost and energy consumption. The local server runs a server that reads the data sent by the central node, processes it and stores the relevant information in a MySQL database, which is known to be reliable, flexible and easy to use.

The local server also runs a web application that gather information from the database and displays the user data in real time. Before starting to use the system, a setup procedure must be performed in the web application. This procedure consists of inserting all the information that the local server needs to start running. That information helps the server to associate the fixed nodes with their corresponding divisions and the mobile nodes with their corresponding users. The web application can also set specific configurations for each user. These configurations are also sent to the database and the local server forwards the message through the reverse path until it reaches the specific mobile node.

3.2 Radio Selection for the Node Network

In order to choose the communication method for the localization and monitoring network, and taking into account the related work and that the project's objective is to be low cost, 4 radio modules were selected for evaluation: XBee, HM-10, nRF24L01+ and CC1101. The Xbee is the most expensive and the nRF24L01+ is the cheapest alternative. HM-10 uses Bluetooth Low Energy which can be a problem, since all the nodes in the network will need to pair with each other and with every new mobile node. This could result in latency issues and connection problems. Also, regarding cost and power consumption, the nRF24L01+ seems a better choice compared to HM-10. A performance comparison was made between XBee and nRF24L01+ [34]. In this article, the performance metrics compared were throughput measurement, mesh routing recovery time and power consumption. Throughput (equation 2) is the volume of data that is sent successfully in unit time. Basically, it is a measure of how fast the network is.

$$\text{Throughput} = \text{number of bytes sent} \,/\, \text{total transmission time (sec)} \qquad (2)$$

Mesh routing recovery time is the time taken to reconnect a node to the mesh after it gets disconnected. Regarding power consumption, the measurement was taken with the lowest power configuration of the nRF24L01+. After analyzing all those parameters, the nRF24L01+ could provide a better throughput compared to Xbee in almost all scenarios. In the point-to-point communication test, it even outshined XBee by a lot. In terms of power consumption, the nRF24L01+ seems to consume more power than Xbee. This can be improved by developing a better energy efficient protocol that enables dynamic sleep cycles.

Therefore, since one of the main objectives of this project is to be relatively cheap and taking into account the comparison made in terms of cost and speed, the nRF24L01+ seems to be the right choice. It costs no more than 1 euro and has the capability to handle the communication between the nodes and to create the network. The major issue with this module relates to how it measures signal strength. Received Signal Strength Indication (RSSI) measures the power of a present radio signal and is very useful for indoor localization because it detects if a mobile node is getting closer or far away from a receiver. However, this radio does not have RSSI.

The nRF24L01+ only has 1 bit called Received Power Detector (RPD) that returns true if the power of any radio signal detected is equal or greater than -64dbm and returns false otherwise. There is no option to change this value and in order to know what it corresponds in terms of distance, some tests and evaluations need to be performed.

The radio's air data rate and power output can be configured. The possible air data rate values are 250kbps, 1Mbps or 2Mbps. The receiver and the transmitter must be programmed with the same air data rate in order to communicate with each other. A higher air data rate reduces de receiver sensitivity, the probability of on-air collisions is lower, and it consumes less power. Since our objective is to identify the division in which a person is in, reducing the receiver sensitivity will be useful because it avoids receiving the signal from neighbor divisions. Table 4 shows the values that are possible to set for output power and the corresponding power consumption.

SPI RF-SETUP (RF_PWR)	RF output power	DC current consumption
11	0dBm	11.3mA
10	-6dBm	9.0mA
01	-12dBm	7.5mA
00	-18dBm	7.0mA

Table. 4. RF output power settings for the nRF24L01+

3.3 Network Topology

The nRF24L01+ radio module [12] has the capacity to create a network with several nodes that can act as transmitters and receivers. The radio's firmware handles all network procedures, which simplifies the node's software, as not everything is implemented in the Arduino. The network is arranged in a tree topology using the RF24 [35] and RF24Network libraries [36] that were created by TMRh20 to explore the capacities of the radio.

The radio can actively listen up to 6 other modules at the same time. One node is the base, and all other nodes are children of either that node or another. Each node can have up to 5 children, and this can go 5 levels deep. This means that it is possible to create a network with a total of 3125 nodes. Since this project is intended to use in a house or nursing home, this limitation will not be a problem.

The libraries provide an efficient acknowledgement of network-wide transmissions, using dynamic radio and network protocol acknowledgements. Each node has a logical address on the local network and the nodes can join the network without any changes to any existent nodes. The administrator of the network must assign a 15-bit address to each node. The address describes exactly the position of the node within the network tree. It is an octal number, and each digit represents a position in the tree further from the base. The following examples help to understand the network's addressing format:

- Node 00 is the base node .
- Nodes 01-05 are children of the base node.
- Node 021 is the second child of node 01.
- Node 0321 is the third child of node 021

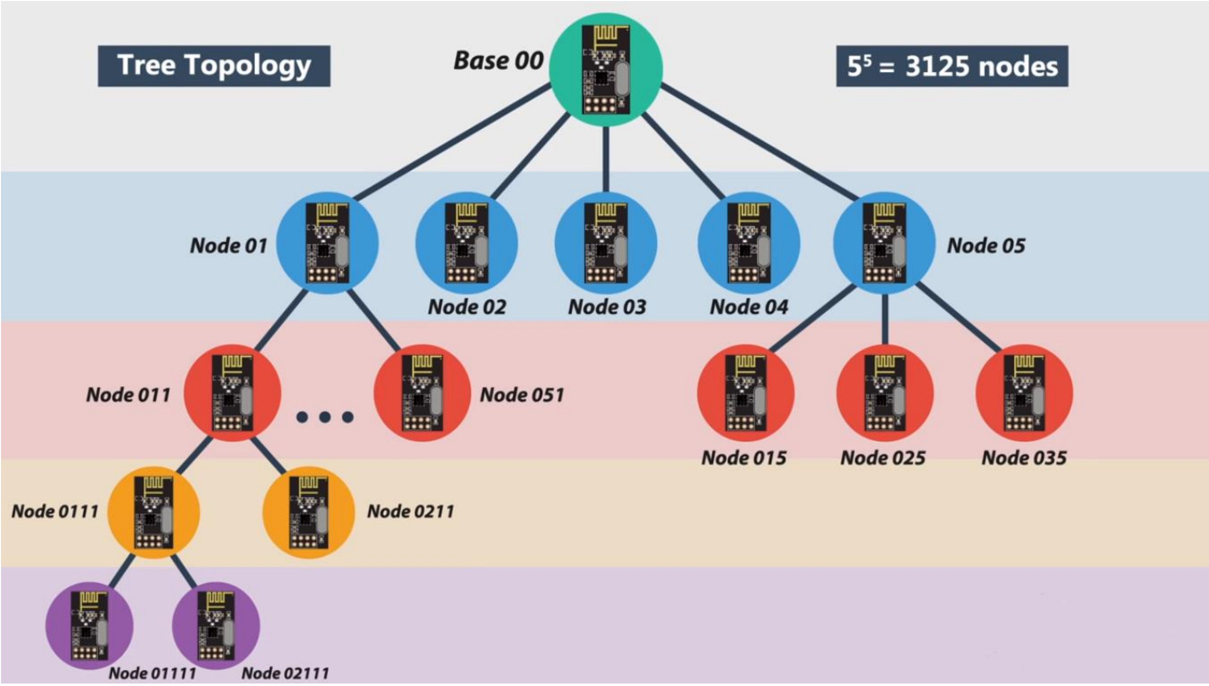

Fig. 10. nRF24L01+ radio network topology [37]

When a message is sent, a header needs to be created with the node address of the receiving node. The network finds the right path to reach that node and sends the message.

In order to localize a person inside a house, a network with fixed nodes in every division must be implemented. Those fixed nodes will receive information from a mobile node that is attached to a user and will send that information to another fixed node until it reaches the central node that can process the information. Therefore, those fixed nodes can only receive and transmit messages between them, which can increase the performance of the fixed nodes due to lower processing.

3.4 Indoor Localization

The indoor detection is one of the main features of this system. Knowing the location of a person inside an indoor space can be very helpful when a person falls, faints or in other several types of situations where the location of people is useful. Developing this type of system with low price components is a great advantage but those components can have some limitations that might affect the reliability of the system. The indoor localization does not need to be very accurate because the objective of this system is to just detect which division the mobile node is at the moment. To achieve this, it is only necessary one fixed node per division. If a division is bigger than usual and there is interest in identifying different areas in it, more than one fixed node will be probably needed.

As previous said in section 3.2, most radios have the RSSI feature but the nRF24L01+ radio have only the RPD bit as a way to measure signal strength. This is an important limitation that needs to be known when developing a localization system. Systems that use radios with RSSI usually compare the strength of all the received signals and assume that the closest node has the highest signal strength, and can determine the distance between the nodes with some accuracy. Because the radio used on this system only have the RPD bit as a way to evaluate the signals received, the common indoor localization methods will not work with this radio due to that limitation. A solution was developed that uses only the RPD bit to perform localization and it allows reliable results. The developed approach consists of a system based on points using the 2 possible RPD values.

The mobile nodes move inside the network and periodically broadcast their ID along with sensors data. The fixed nodes that receive the message from the mobile nodes, add the value of the RPD bit to the message and send that information to the central node or to another fixed node that can send the data to the central node. The central node also works as a fixed node, but it is connected to the local server and sends to it all the information received through a serial channel. The local server runs a python script (server script) that receives the network information and starts processing it. This is where all the localization is performed, which simplifies the software of the nodes.

Regarding the radio configurations, the power output and the air data rate were both set to the maximum. Setting the power output to the maximum helps to increase the distance that the RPD bit still returns "1". Otherwise, the distance was too small to make a difference on the detection. The air data rate was set to the maximum because it reduces the on-air collisions, consumes less power and reduces the sensitivity of the radios, which helps to avoid ambiguities that happen when the radio receives messages from neighbor divisions.

3.4.1 Localization Algorithm

First, the server script associates all the fixed nodes of the network with their corresponding division. Every mobile node has an array with the size of all the fixed nodes that exist on the network. Each value inside this array is associated with a certain fixed node. These values, or confidence points, are the system's confidence that the mobile node is located at a certain fixed node's division. The higher the confidence points for a division, the more confident the system is that the mobile node's location is at that division. When a fixed node is detecting the presence of a mobile node and the RPD value is "0" (low strength), 1 confidence point is attributed to that division. If the RPD value is "1" (high strength), 5 confidence points are attributed to that division. The other divisions that had no messages received have 0 confidence points attributed. This is performed for every mobile node every second within 4 iterations (4 seconds). Each iteration adds up to the previous confidence points based on the new messages received. After the 4 iterations, the division with the highest confidence points is assumed to be where the mobile node currently is. After the detection, all the confidence points are set to zero and the iterations start over again. If all the divisions have 0 confidence points, the system declares an unknown position for that mobile node. The "1" and "5" confidence points that are attributed according to the RPD value, were chosen making sure that the confidence points attributed when the RPD bit is high are substantially greater than the ones attributed by a low RPD. Since there are 4 iterations and a low RPD gives 1 point, some divisions could have 4 confidence points after the 4 iterations. In order to give advantage to a division with a high RPD bit, 5 confidence points were chosen to avoid situations where two divisions have the same confidence points, but the RPD values were different at a certain time.

3.4.2 Dealing with Ambiguities

However, there can still exist blind spots and ambiguities where a mobile node apparently seems to be in two different divisions at the same time. This can occur when a mobile node has two or more divisions with the same confidence points. To solve this, if a division has the highest confidence points at two detections in a row (8 iterations), the mobile node starts the next detection with already 2 confidence points on that division, instead of 0 like the other ones. This is to create a certain resistance to leave the division and solves several ambiguities. That resistance is removed as soon as a different division is assumed as the location of the mobile node. If the mobile node is in a position where there is no detection of the same division two times in a row, the resistance is not created, and the ambiguity cannot be resolved until the mobile node moves or a certain fixed node is changed to a better position. Another implementation that tries to solve the ambiguities is by using the concept of neighbors. All the divisions that have a direct passage to a certain division are its neighbors. Before finishing the detection procedure, the server script checks if the division that now has the highest confidence points can be directly accessed from the mobile node's last division. This automatically ignores the divisions that might be getting confidence points attributed but cannot be accessed from the last division. If a non-neighbor division have the highest confidence points, the system assumes that the mobile node's location remains on its last division.

3.4.3 Implementation Example

Figure 11 shows an example of a 9 nodes network implementation inside a house with the central node placed on the hall. The living room on the figure can be an example of a division that needs 2 nodes because its size might be too big or there is a need to identify certain areas inside it. There are several ways to create a network inside an indoor space. Having very few connections can be a problem because the nodes could be too far away for a message to be received. Lots of connections and levels within the network means that more fixed nodes can receive messages from the mobile nodes and those messages need to take longer paths to reach the central node. This increases the network traffic and the number of ambiguities, and also decreases the probability of a message reaching the central node. The goal is to find a balance between the node's distance and the number of connections.

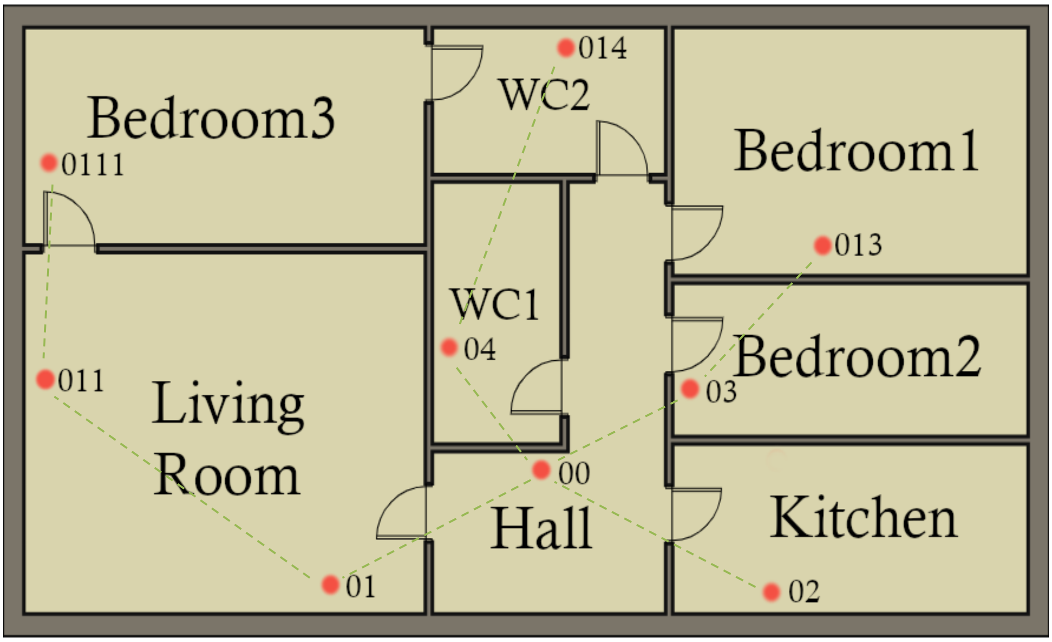

Fig. 11. Example of a network implementation inside a house

3.5 Fixed and Mobile Nodes

The nodes that compose the network were developed with the goal of being small, lightweight, energy efficient and low cost. Both mobile and fixed nodes are made of the same components, but with the difference of the mobile nodes having sensors incorporated. The nodes are built on top of a PCB board that helps with the connection and support of the node's components. It was designed by Professor Renato Nunes from Instituto Superior Técnico and it was created for other projects but can be used on this project as well. Attached to the board are the nRF24L01+ module for radio communications and an Arduino Nano to control everything. There is also 5 leds and 3 buttons attached to the PCB board that are used for testing, alerts and to simulate actuators and sensors.

The mobile nodes are equal to the fixed nodes but have more modules attached. In order to detect falls and inactivity, an MPU6050 [38] accelerometer was attached to the board. And for heart rate measuring, an MAX30102 [39] module was connected. These two modules communicate by the I2C protocol but had no connections made for them inside the PCB, so some wiring and soldering had to be made. Because this is the first prototype, the two extra modules are attached to the board with duct tape and on the empty space that can be found on the board. This is not the best solution, but it works and does not have any impact on the radio communications. A better prototype can be made with a new PCB designed to have connections to all the modules and with a considerable smaller size.

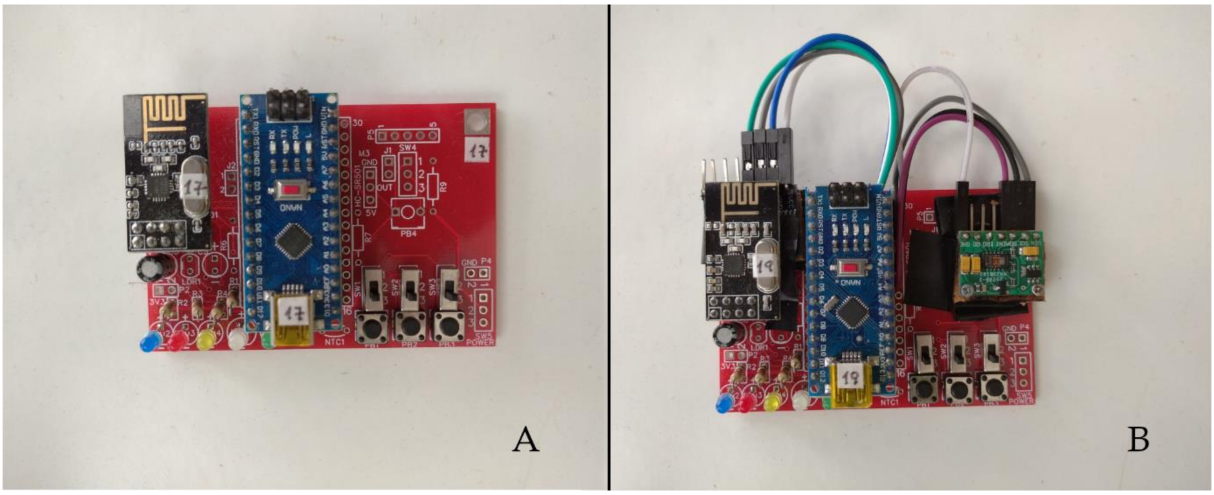

Fig. 12. A – Fixed Node, B – Mobile Node

3.6 Health Monitoring

Through localization it is possible to identify if a person is active and moving around regularly, which already gives some information about the person's health status. If a person goes to the toilet many times throughout the day or stays in a division for a long time, it may indicate that there is a problem. The patient also might have difficulty to move or have fallen. Combining that with sensor information leads to a more correct and detailed diagnosis of the user's health status.

The mobile node is carried by the person being monitored. This device periodically monitors the sensors data and sends that information to the base node to be processed and stored. Those sensors are the MPU6050 and the MAX30102.

MPU6050 - The MPU6050 is a low-power integrated 6-axis device that combines a 3-axis gyroscope, a 3-axis accelerometer and a Digital Motion Processor (DMP). It features 3 16-bit analog-to-digital converters (ADC) for gyroscope outputs and also 3 ADCs for accelerometer outputs. The gyroscope can be programmed to use a full-scale range of $\pm$250º/s, $\pm$500º/s, $\pm$1000 º/s, and ± 2000 º/s. The accelerometer can be programmed to use a full-scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$, and $\pm 16g$. The communication is performed using I2C at 400kHz. Additionally, the sensor has an embedded temperature sensor and an on-chip oscillator with $\pm$1% variation over the operating temperature range.

MAX30102 - The MAX30102 is a low-power integrated pulse oximetry and heart-rate monitor module. The device has internal LEDs, photodetectors, optical elements, and low-noise electronics with ambient light rejection. It communicates though the standard I2C protocol. The module can be shut down by software with zero standby current with the power rails remaining powered. The MAX30102 has an on-chip temperature sensor for calibrating the temperature dependence of the SpO2 subsystem. The sensor integrates Red and IR LED drivers to modulate LED pulses for SpO2 and heart-rate measurements. The LED current can be programmed from 0 to 50 mA and the LED pulse can be programmed from

69μs to 411μs. This allows the algorithm to optimize SpO2 and HR accuracy and power consumption based on use cases.

3.6.1 Fall detection

Detecting a fall is one of the most important features of this project. By combining this feature with indoor localization, it is possible to know where and when a person has fallen, which is very helpful because time is an important factor on fall events. The detection of a fall is urgent and needs to be the detected as soon as possible in order to avoid more consequences to user. That is why the detection is performed on the mobile node itself to avoid delays that might exist. The MPU6050 sensor attached to the mobile node is the only module necessary to detect if the user has fallen. Because the mobile node has limited processing power, machine learning algorithms for the detection were discarded. Instead, a simple fixed thresholds solution was implemented based on [22] and [23]. The major problem with this solution is to differentiate a fall from normal daily activities such as sitting, getting up, walking, and jumping. Having a solution with the sensor placed on the wrist makes the problem even harder due to its high mobility during the day.

Before start using the MPU6050 sensor, it is necessary to perform a calibration procedure to calculate the offsets for each of the accelerometer and gyroscope axis. Adding the offsets calculated to the values measured by the sensor gives a more accurate reading.

The fall detection algorithm constantly reads the MPU6050 values and stores the information on variables for the 3 accelerometer axis, the 3 gyroscope axis and the ambient temperature value. The MPU6050 is configured to operate with an accelerometer scale range of $\pm 8g$ and a gyroscope scale range of $\pm$250 º/s. The node is continuously calculating the magnitude of acceleration (equation 1) and based on the magnitude, the system uses triggers that activate or deactivate according to those values. If all the triggers activate, a fall is detected. The first trigger is activated when the magnitude of acceleration goes below the 0.4g lower threshold (LT), which indicates that the user is falling. After that, if the magnitude of acceleration goes higher than the 5g upper threshold (UT) within the next 500ms, the second trigger is activated (impact period). If during the next 500ms after the second trigger being activated, the system detects an orientation change (OC) greater than 200 º/s, the third trigger activates. OC is calculated using equation 1 but replacing the 3 accelerometer values with the gyroscope values. After waiting for 3 seconds, a blue LED turns on indicating that the system is suspicious of a fall event. This 3 second period is to make sure the fall event is finished, because some falls can take a longer time and would make the detection less correct. Next, for 60 seconds, the system checks once again the orientation change of the user. If there is a movement greater than 100 º/s, the system assumes that the user is conscious and moving freely. Therefore, the suspicious fall is cancelled, and the LED turns off. If the user presses button2 within that period, the suspicious fall is also cancelled. However, if there is no movement detected from the user during the 60 seconds period, the system now is certain that a fall has occurred, and the user might be unconscious or unable to call for help. A white LED that could simulate an actuator such as a buzzer turns on and the user's emergency contact automatically receives an SMS message with information about the time and the location of the fall event. An advantage of this solution is that the system automatically sends an alert without any interaction from the user, which is useful because the user might be physically unable to call for help after a fall.

Figure 13 shows the flowchart of the fall detection algorithm where AM is the acceleration magnitude, OC is the orientation change, LT is the lower threshold (0.4g) and UT is the upper threshold (5g). These threshold values were selected based on the literature but can easily be adjusted for specific cases of certain users.

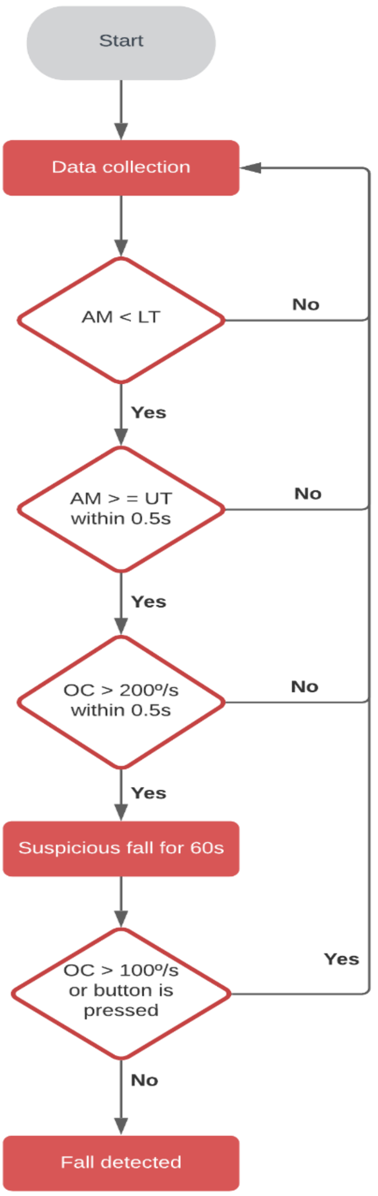

Fig. 13. Flowchart of the fall detection algorithm

3.6.2 Inactivity detection

Detecting inactivity is also very important because, even if a fall was not detected, the person could have fainted or there is an unusual loss of mobility that needs to be evaluated. The system does not have the capability of detecting unusual inactivity according to the user's daily movement patterns. In order to do that, machine learning algorithms will be probably needed. This system only detects a large period of inactivity during a configurable time interval.

The inactivity detection starts by calculating the orientation change. If for a predefined amount of minutes (60 seconds by default), the orientation change remains below 10 º/s, an inactivity event is generated and an alert is sent the same way as the fall detection alert, and a white LED turns on that could simulate an actuator such as a buzzer. Pressing button2, cancels the alarm and turns off the LED. If there is an orientation change higher than 10º/s during the predefined period, the timer is reset, and it starts counting from the beginning.

3.6.3 Heart rate measuring

The heart rate, or pulse, is the number of times the heart beats per minute (bpm). When a person is relaxed and calm, the normal heart rate is usually between 60 and 100 bpm. Having a heart rate lower than 60 does not necessarily mean that it is a medical problem. People with a lot of physical activity or that are very athletic can have heart rates values as low as 40 bpm. Other factors that can influence heart rate values are the air temperature, the body position, emotions, the body size and some medications.

In this solution, the MAX30102 sensor is always on standby mode to avoid abnormal readings and to save battery. If a user presses button1 (the leftmost button), the sensor wakes and waits for a finger to be placed on the sensor within the next 5 seconds. If no finger is placed, the sensor returns to standby mode. If there is a finger placed on the sensor, it starts measuring the heart rate of the user for 20 seconds. It senses the heartbeats and performs some calculations to get the bpm value. After gathering several of those values, an average is calculated and updated along the 20 seconds measuring period. After the measuring procedure, a red or green LED turn on depending if the value measured belongs within the acceptable bpm range, which is 40-160 bpm by default (the range can be predefined and adjusted to the user). After 5 seconds the feedback LED will turn off and the sensor returns to the standby mode. Figure 14 shows the flowchart of the heart rate measurement algorithm.

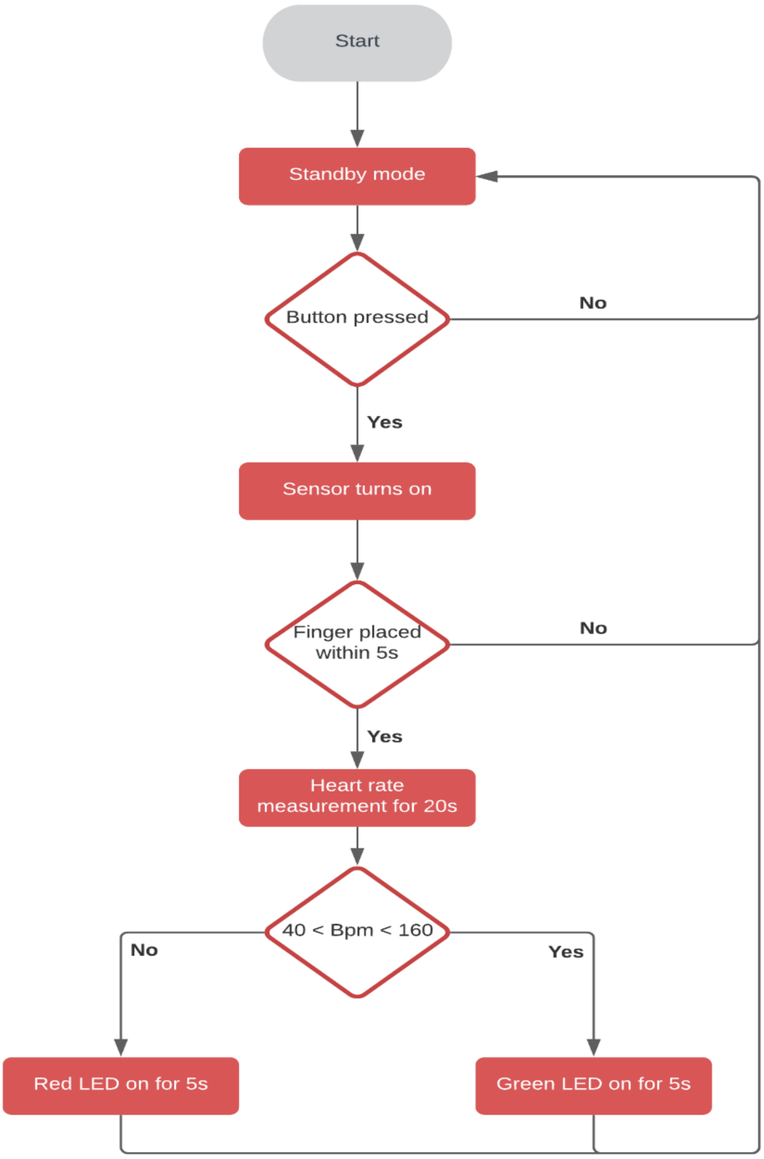

Fig. 14. Flowchart of the heart rate measurement algorithm

3.7 Web Application

3.7.1 Web application architecture

The web application offers access to every element of the system and allows the analysis and management of the network and user data. Django is a python web framework that runs as a backend server creating and managing all the database models. Django was selected as the backend framework for this solution due to previous experience in developing web applications with it, and because Python has libraries and tools that help to read the serial data from the central node very easily. Django has a powerful feature called Object-Relational Mapper (ORM) which enables a good interaction with the database. In order for the frontend to access the database, a RESTful API was created using a framework called Django REST framework. A RESTful API uses HTTP requests to access and use the data. That data can be read, updated, created and deleted using the methods GET, PUT, POST and DELETE. The Django REST framework serializes the data from the Django ORM and allows access and updates via the RESTful API. The frontend of the web application is created by the React framework inside the Django project. React is a frontend JavaScript library that handles user interfaces. Django loads a single HTML template and let React manage all the frontend. It gets and posts data via the Django API and uses the static files from Django when a user visits the URL.

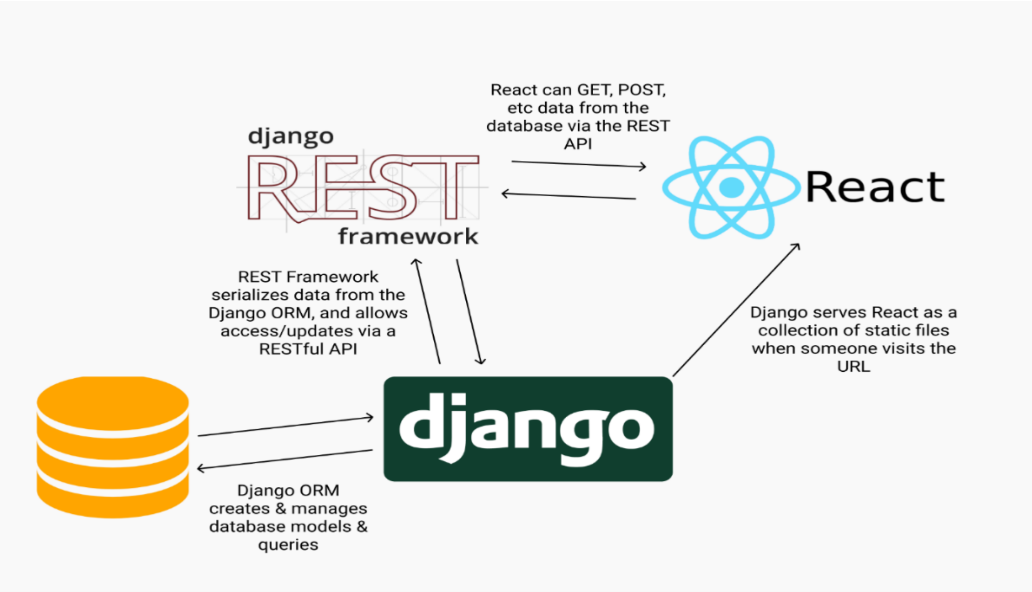

Fig. 15. Web application architecture. Django application using a RESTful API to allow React to use data from the database [40]

3.7.2 Web Application GUI

The web application gives access to all the information processed by the system. It has a simple and intuitive design that helps the user to analyze the data with ease. It was created with the purpose of being able to be used by people that have a basic knowledge on how to use a computer, thus avoiding the need for training. It offers three main functionalities that will be described next.

Dashboard – The dashboard is the main page of the application. It displays a table with the name of all the users stored on the database and their current location. This information is obtained by constantly getting information from the database by calling the API. There is a search bar that filters the users on the table. A radio button allows to choose whether users are filtered by name or by the current division. Finally, it is possible to get more information about each user by clicking on their name.

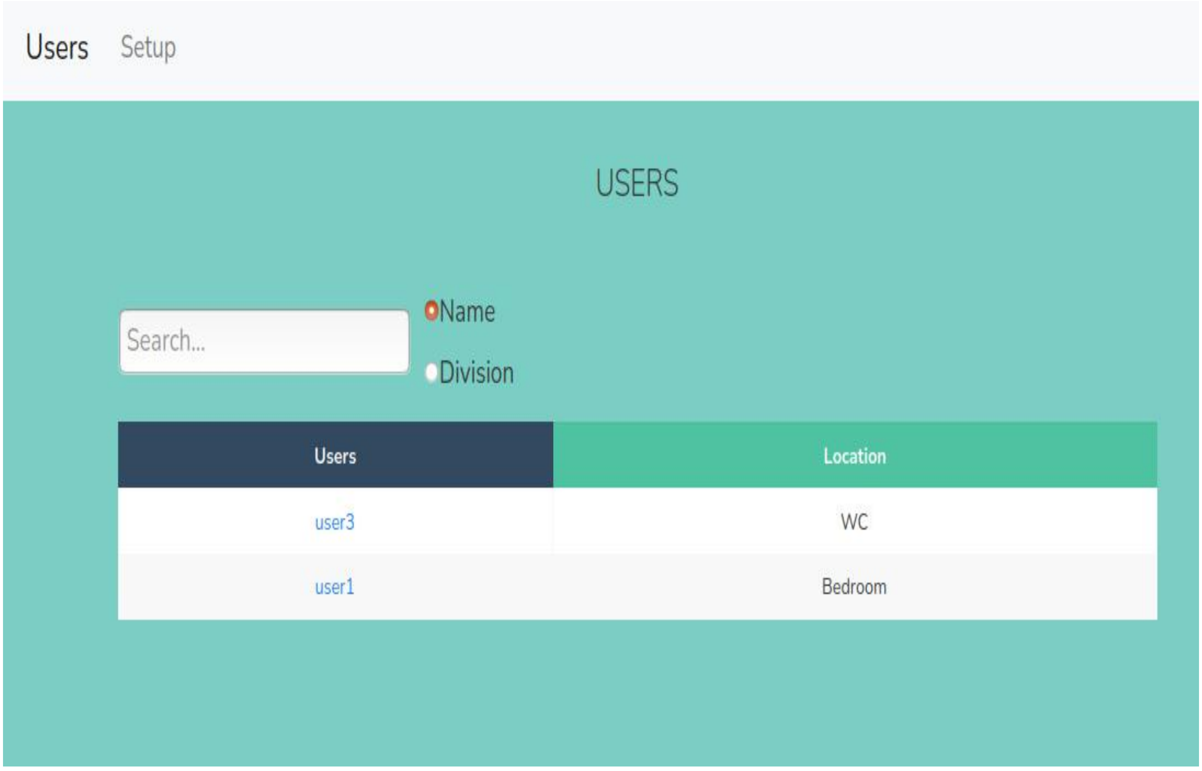

Fig. 16. Dashboard of the web application

User page - After clicking on a user, the app redirects to a user page where it shows more detailed information about the user such as the current location, the history of divisions that the user already went, the history of falls, the previous heart rate readings and the last heart rate measurement. The division and fall history components display the information divided by location and date. The heart rate history component displays the data divided by the bpm value and date. The component that displays the last heart rate value measured, gives feedback according to the value measured (good/bad). There is also a section to manage the alarms related to the user. All the alarms are configurable to each user and are simple to manage. A switch button can be turned on or off for each of 4 configurable alarms (inactivity, fall, heart rate and emergency button). Regarding the inactivity detection alarm, there are two input boxes that indicate the time range for a detection to be activated. The default range is 09:00h - 23:00h. This feature helps to avoid false alarms that can happen during the night when the user is sleeping for example. The time range of the inactivity detection is configurable because each user might have a different daily schedule.

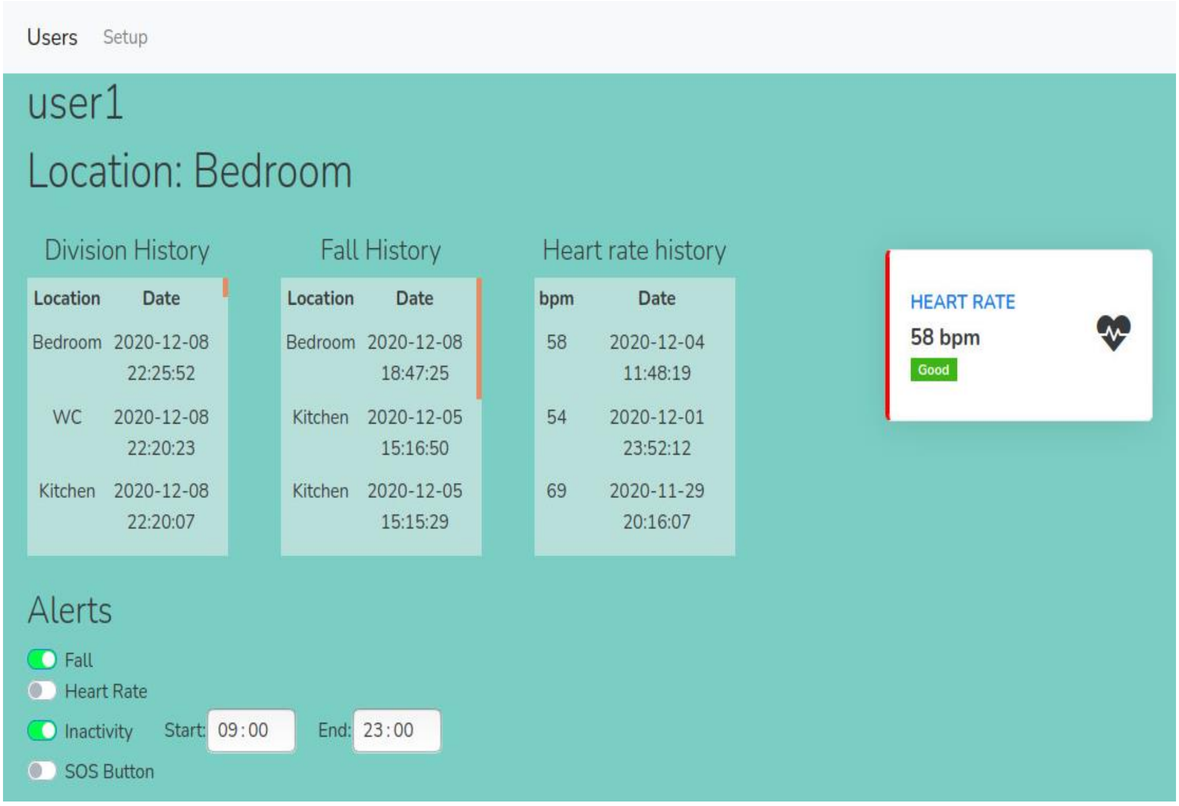

Fig. 17. User page of the web application

Setup page – Finally, the application has a setup page that is accessed by clicking on "setup", which is inside a navbar located on top of the web application. The purpose of this page is to setup the network before running the server script, which handles all the network information. The setup is divided in two sections: the mobile nodes and the fixed nodes. At the mobile nodes section, users are created by inserting their corresponding mobile node ID, username and a phone number to send alert notifications. At the fixed nodes section, it is required to insert the fixed node ID, its corresponding division, and the ID's of all its neighbors separated by commas. For both sections there are buttons to add or delete users and fixed nodes. There is also another way to setup the network. By clicking on the "file" button, the application requests a JSON file and after selecting the file, all its information about the mobile and fixed nodes is displayed on the setup page. It can also be edited the same way as adding the information manually and it is even possible to add more information on top of it. Creating a JSON file with all the information needed to setup the network can be very useful by being more practical and time saving. Figure 18 shows a JSON file example to setup the network. When everything is done, clicking the upload button will send all the setup information to the database and the server script can start running. The setup procedure is very easy to perform and because the information is stored on the database, there is no need to repeat the procedure if the web application needs to be restarted. Finally, the "reset" button will delete all the data inside the database. The setup page also provides a short message feedback when performing a reset or an upload procedure. Every time the setup page is revisited, all the database information is written on the setup form boxes, which helps the user to know the network configurations.

```
{
"users":[
          {
          "user_id": "12",
          "name": "user1",
          "phone_number": "--"
     },
     {
          "user_id": "22",
          "name": "user2",
          "phone_number": "--"
     },
     {
          "user_id": "3",
          "name": "user3",
          "phone_number": "961234567"
     }
        ],
"f_nodes":[
          {
          "fixed_node_id": "0",
          "division": "Hall",
          "neighbours": "1,2"
     },
     {
          "fixed_node_id": "1",
          "division": "Kitchen",
          "neighbours": "0,2"
     },
     {
          "fixed_node_id": "2",
          "division": "WC",
          "neighbours": "0,1"
     }
          ]
}
```

Fig. 18. Example of a JSON file to setup the network

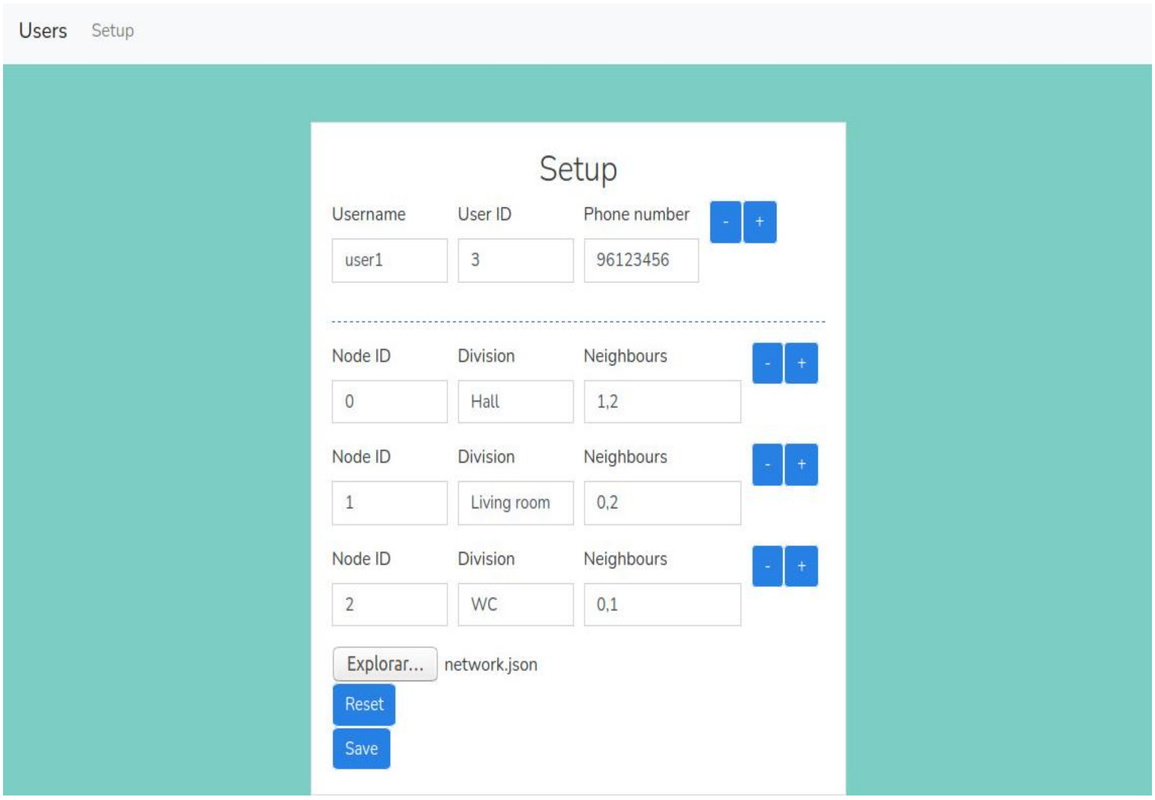

Fig. 19. Setup page of the web application

3.8 Alerts

A system that generates lots of alerts all the time will not be usable because people will tend to ignore them after a while. That is why the alerts of this system are configurable for every user and are only emitted when there is an urgent event, which will hopefully avoid more serious consequences. Besides the falls, inactivity and heart rate alerts, the user can also press the SOS button (the rightmost button) of the mobile node, which automatically sends an alert to the system. The alerts have two ways to be emitted. The first one is on the mobile node itself, where leds turn on simulating actuators such as a buzzer. The second one is an SMS message that is sent to the user's defined emergency phone number. This phone number could belong to a user's family member, to the user's nursing home workers or even to the user's doctor for example. The SMS alert works by using a web platform called "IFTTT" [41]. IFTTT's name comes from the programming conditional statement "if this, then that". It is a web service that allows users to create chains of conditional statements called applets that are triggered by changes on several other web services. The platform basically connects apps, devices and services from different developers and triggers automations. The platform also has a mobile application that is useful when performing cellphone related automations, which is this system's case.

After creating an account on the platform, it is possible to access an URL with a personal key that will be used to emit an alert. When an alarm event is detected, the server script sends a post request to the URL gathered before and adds to it the details of the event. A predefined phone that must be connected to the internet and has the IFTTT application installed, receives the notification from the platform and automatically sends an SMS message to the user's emergency phone number. Figure 20 shows all the possible SMS messages that can be sent by the system from the perspective of the emergency phone user.

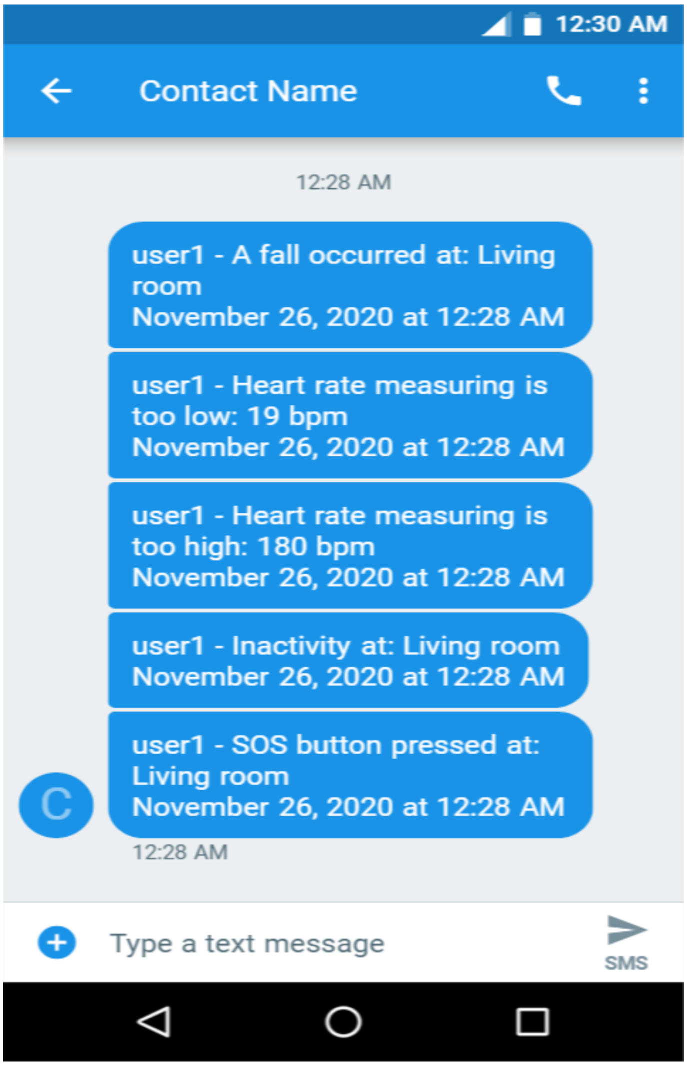

Fig. 20. All possible SMS messages sent from the application

All the functionalities of the proposed solution were addressed. In the next chapter, the implementation of each of those features is explained in more detail.

4 Solution Details

In this chapter, the implementation details about the indoor localization, the database, the alerts system and the web application are covered.

4.1 Indoor localization

4.1.1 Nodes and Network Details

All the Arduino code for the nodes was developed on the Arduino IDE platform. In order to compile the code, it was necessary to select “Arduino Nano” as the type of board, select the USB port to send the data by serial and to select the bootloader as “old bootloader”. It was developed code for the 3 types of nodes: mobile node, fixed node and central node. The Arduino code is divided in 2 major functions: Setup and Loop functions. The code inside the Setup function runs only once at the node’s initialization. The Loop function is called right after the Setup and runs in a loop until the node is disconnected or restarted. The central node acts in the same way as the fixed nodes, which means that it can also receive messages from the mobile nodes. The only difference is that it sends the data through a serial channel to the local server.

The mobile nodes move freely through the network but need their messages to be received by a fixed node. The mobile nodes do not know which fixed node is closest to their position in order to send the message to the central node. Therefore, a broadcast solution where the mobile node sends the information to all nearby fixed nodes is needed. The RF24Network library provides a multicast feature that sends a message to all the nodes inside a multicast level. A multicast level is basically a shared address where multiple nodes can listen and send messages. By setting all the fixed nodes to multicast level 1 and all the mobile nodes to multicast level 2, the message from the mobile nodes can reach the central node from every position of the network. The 3 types of nodes declare the same message structure that is used to perform the radio communications. All the nodes must have the same message structure declared in order to correctly read the messages received. The structure of the messages is composed by the ID of the mobile node that sent the message, the RPD value of the fixed node that receives the message, information about the sensors (in this prototype, the sensor information is the heart rate measurement and the existence of falls and inactivity) and if the mobile node has started (this field will be explained later).

On the mobile node’s Setup function, buttons, LEDs and the nRF24L01+ radio are initialized. The radio configurations such as the data rate, the power output and the multicast level are set. Finally, both the heart rate and accelerometer sensors are initialized, and the heart rate sensor is set to standby mode. The setup function for the fixed nodes and the central node is the same as the mobile node’s function but without the configuration of the sensors.

On the mobile node's Loop function, the network is regularly checked by calling an update method. Next, all the inactivity, fall and heart rate procedures are run, and its values are stored. The RF24Network provides a feature that helps identifying different types of messages. A character can be attributed to the header of a message. This is useful for the fixed nodes to check if the message received comes from a mobile node or from a fixed node. Therefore, before sending the message, every mobile node sets the header type to "M".
Finally, all the health monitoring variables and the mobile node's ID are sent every second to multicast level 1.

The fixed nodes simply receive the message from the mobile nodes and add their RPD value to it. After that, they send it to the central node. If an intermediate fixed node receives the message from another fixed node, the message is not changed by adding a different RPD value because the message sent by the first fixed node is not a multicast message but a direct message to the central node.
The central node first checks if it is receiving the message directly from the mobile node. If that is the case, it calculates its RPD value. After that, it sends all the values of the message received through a serial channel to the local server. The message sent to the local server is a text message where all the values are separated by commas.

4.1.2 Localization Algorithm Details

The local server runs the server script, which starts a setup procedure where it gets all the information from the database and creates an array with all the mobile nodes and another one with all the fixed nodes. The number of both mobile and fixed nodes is always configurable. These 2 arrays are filled with information from the database. On the mobile nodes array, it is also created a confidence points array for each mobile node, that will be used to determine the position of the nodes. The server script now has everything it needs to start processing the received data.

The server script is constantly waiting for a message to arrive from the central node. When it arrives, the server script starts to divide the message by commas into several data. One of those values is the ID of the mobile node that sent the message to the central node. After knowing that, the server script now can store the information received at the correct index of the mobile nodes array.

The localization detection algorithm consists of a system of confidence points based on the RPD value received. These confidence points are attributed to the confidence points array of each mobile node. The message received also has information about which fixed node received the message. The only difference is if instead of a fixed node ID, the server script receives the mobile node's ID. This means that it was the central node that received the message. After knowing all that, it can associate the confidence points to the correct division.

Next, the server script iterates through the mobile nodes and finds, for each of them, the division with the most confidence points. If more than one division has the maximum confidence points, an ambiguity is detected. It now starts performing the procedures already explained in section 3.4 that try to solve the ambiguities. The previous location of the mobile node is important for the detection because it is used to check whether the node's current position is a neighbor of its previous position, which avoids ambiguities. When a mobile node turns on, there is no previous position. Because of that, there is a value inside the message received by the server script that gives information on whether the mobile node was recently turned on. If that is the case, the node's starting point is set to its current position. This implementation also helps in a scenario where the mobile node is disconnected for some reason and it restarts on a different division. If the division where the node restarted is not a neighbor of its last position, the detection will be inaccurate. Therefore, resetting the node's previous location solves the problem.

After finding the mobile node's position, that information is stored. The last procedure is to reset all the confidence points (explained in 3.4.1), add any resistance points (explained in 3.4.2) when that is the case and set all node positions to unknown. After the reset procedure, the cycle starts over again.

4.2 Database

The system uses two database models, one for the fixed nodes and the other for the users. These models are then migrated to a MySQL database, which is the default Django database.

The fixed node model has the following fields:

- **fixed_node_id –** Stores the ID of the fixed node and it is the primary key of this model.
- **division –** Stores the name of the division where the fixed node is placed.
- **neighbors –** Stores the IDs of all the fixed nodes that could be the next user position, from a certain division.

The user model has the following fields:

- **user_id –** Stores the ID of the mobile node and it is the primary key of this model.
- **name –** The username selected to identify the user.
- **location –** The current location of the mobile node.
- **division_history –** The current and previous locations of the mobile node.
- **sensor_max –** The value of the user's last heart rate measurement.
- **bpm_history –** The time and values of all the user's previous heart rate measurements.
- **fall –** Boolean value about the existence of a fall event.
- **fall_history –** The time and location of all the user's previous falls.
- **fall_alert –** "1" or "0" value representing whether the fall detection is turned on or off.
- **bpm_alert –** "1" or "0" value representing whether the heart rate detection is turned on or off.
- **inactivity_alert –** 2 values representing whether the inactivity detection is turned on or off and if the current time is within the inactivity range.

- **inactivity_range –** Time range for the inactivity detection to be on.
- **phone_number** – The emergency phone number chosen to receive the alerts from a specific mobile node.
- **sos_alert –** "1" or "0" value representing whether the SOS emergency is turned on or off.

4.3 Alerts

Regarding the alert configurations, each of the 4 toggle switches in the user interface trigger a corresponding function when they are pressed. Each function updates the corresponding database field according to the toggle switch state. For example, if the heart rate alert has changed to "ON", a function is called that updates the database field "bpm_alert" to "1". The fall and SOS alert configurations work the same way as the heart rate. However, the inactivity alert configuration has some differences. The database field "inactivity_alert" has 2 numbers divided by a comma instead of only 1 number like the other 2 alert configurations. This is due to the time range feature that the inactivity alert has. The first number of the "inactivity_alert" database field corresponds to the state of its toggle switch. The second number means whether the current time is within the time range defined at the inactivity configuration. If the toggle switch is turned off, the second number is "0" because it has no use due to the alarm being deactivated. Therefore, the possible values that can be at the "inactivity_alert" database field are:

- "0,0" – Alert is turned off.
- "1,0" – Alert is turned on and the current time does not belong to the defined time range.
- "1,1" – Alert is turned on and the current time is within the defined time range.

The server script is in charge to regularly check the state of the inactivity alert and if the current time is within the defined time range, and to update the "inactivity_alert" database field. If there is a state change on any one of the 4 alert configurations, the server script sends a serial message to the central node with the new configuration values and the ID of the corresponding mobile node. The server script also sends the alerts configuration data when it restarts and also when a mobile node restarts. This procedure makes sure that the mobile nodes always know the defined alert configurations. When the central node receives the message, it automatically sends those values to multicast level 2 and to multicast level 1 and with the message's header type set to "F". The reason behind sending the message to both multicast levels is because the mobile node might be too far away from the central node to receive the multicast level 2 message. Sending to multicast level 1 can help to decrease the distance because now the fixed nodes that are closer to the central node also receive the message. The fixed nodes that receive the message first check if the character of the header type is an "F". If that is true, the same procedure as the central node is performed. The fixed nodes send to multicast levels 1 and 2 until the message eventually reaches the mobile node.

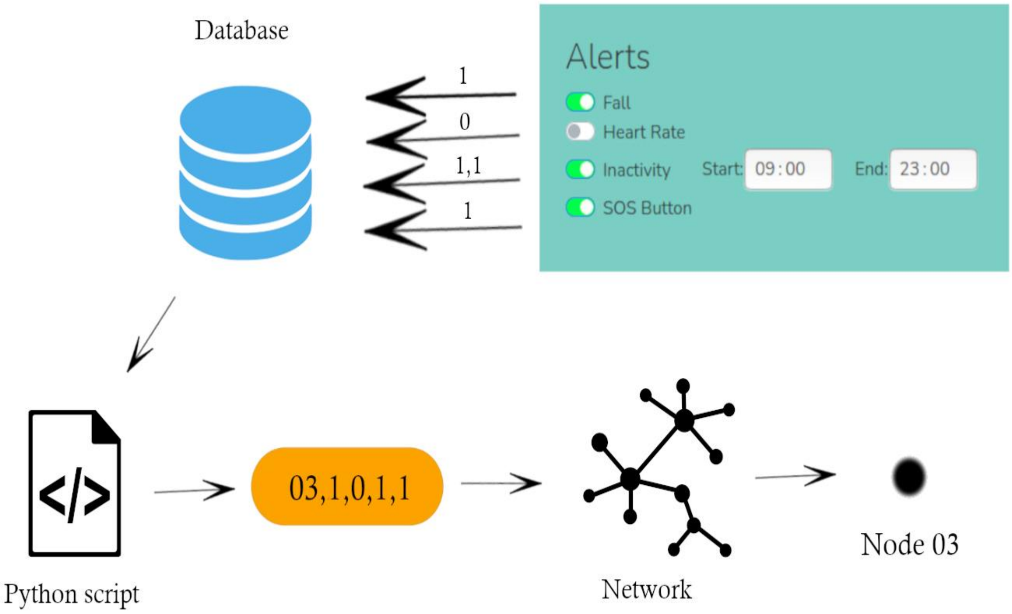

Fig. 21. Example of an alert configuration message sent by the alert system

Figure 21 shows an example of an alert configuration message sent by the alert system. On this example, some alert configurations were performed to the user that has the mobile node 03 attached. After the configuration, the values are sent to the database and the server script will get the data from it. In this example it is assumed that the current time is within the inactivity time range. After processing the data and adding the destination mobile node's ID, the server script sends the information to the central node and it will eventually reach mobile node 03.

When the message reaches the mobile node, it checks if the message received has its mobile node's ID. If that is the case the node can read the message. According to the values inside the message, the node can activate or deactivate the inactivity, fall, heart rate and SOS alerts. The other mobile nodes will also receive the message, but it is ignored because the ID present in the message does not belong to them.

After the detection of an emergency event, the mobile node sends that information to the central node 10 times (fall and inactivity) or continuously (SOS and heart rate). The reason for this is because there might be situations where a message gets lost and does not reach the central node. By continuously sending that an emergency event has occurred, the chances of the information reaching the central node increase. When any of the 4 alerts is being emitted by a mobile node, the server script immediately sends a POST request to the IFTTT URL and an SMS message is sent to a predefined phone number.

4.4 Web Application and Running Procedure

The Django app offers two URLs that are used when the REST API is accessed by the frontend. Those URLs are "/api/users/" and "/api/fixed_nodes/". The server script was created as a custom Django command instead of a general script file. The reason for this is because with this approach, the server script can access the database information and process it more easily, and also avoids any missing files or directory related problems due to being saved inside the web application's project.

The Dashboard, User and Setup pages were created by 3 corresponding files at the React project. Each of them is composed by JavaScript and HTML code. The JavaScript code is responsible for the interaction between the user and the web components, and to get and update the database information through the REST API URLs created before. The HTML code renders the page via an HTML template created by the Django Application. For the pages styling, a CSS file was used combined with Bootstrap components. Bootstrap [42] is popular frontend framework that features numerous HTML and CSS templates for UI interface elements.

In order to run the application and start using the system it is necessary to first start the server. On a terminal, inside the project's folder, inserting the command "python3 manage.py runserver" starts the Django application. After that, the application can already be accessed locally through the URL "127.0.0.1:8000/". It is also possible to access the application through several devices that are connected to the same local network by providing the IP of the central computer into the terminal command (example: "python3 manage.py runserver 192.168.1.4:8000"). The other devices can access the application by entering the IP of the central node as a URL.

After starting the server and performing a setup procedure to the network if it was necessary, the server script can be run inserting the command "python3 manage.py script" on a new terminal. Every time the database gets cleared and a setup procedure is performed from scratch, the server script needs to be restarted.

5 Evaluation

5.1 Indoor localization

Before the implementation, the goal for the radio configurations was to set the air data rate to the maximum because it consumes less power and reduces the radio sensitivity, and to set the power output to the minimum because it also consumes less power and the RPD bit will only return “1” when the mobile node is close enough to the fixed node. After some tests with this implementation, the results showed that the best configuration for this project is both the air data rate and the power output set to the maximum level. The reason for the power output change is because when it was set to the minimum, the RPD bit returned “1” when the mobile node was only a few centimeters apart from the fixed node. This would make the RPD feature useless when performing indoor localization. By increasing the power output at incremental steps, the best solution was found when it was set to the maximum. At this configuration, the RPD bit returns “1” at a few meters from the fixed node. The results vary a lot due to the radio orientation of the mobile and fixed nodes, but the maximum power output was the only configuration that gave the most consistent RPD value “1” inside a division.

As stated earlier in section 4.1, in order for the mobile nodes to move freely throughout the network and sending messages to the central node independent of the position was a challenge. But using the multicast feature where the fixed nodes were set to multicast level 1 and the mobile nodes to multicast level 2 the issue was solved, and it works without any problems.

Another challenge was the lack of the RSSI feature. The detection algorithm using a system of points based on the RPD value does not have the same reliability as solutions that have radios with RSSI. And due to the detection algorithm logic, there is no immediate detection. There is always some detection delay when a user is moving through the divisions. If a user performs some specific movements such as for example: if during the detection of a division, the user goes immediately to another division, depending on how fast the user was moving, the correct detection of the new division could only happen after almost 8 seconds. This is because, even if the user is now on a new division, he may have been in the old division long enough for it to have the most confidence points. However, the indoor localization implemented on this system does not have to be perfect in terms of time delays and overall accuracy, because the purpose of this system is to give a general idea of “where” and “when” a user went, which is useful for sending alerts and to analyze the user’s movement patterns. Besides that, since the system is intended to be used by the elderly, it is not expected that the users would move that quickly.

Despite accepting that a time delay will exist during the detection, the goal was to reduce the delay as much as possible. With that purpose, the detection time was set to 1 second and the mobile nodes sent their messages every 200ms. The results showed a very rapid change of detected divisions. Detecting the user’s movement patterns would be impossible due to the constant change in the detected divisions. This inconsistent behavior was not very useful for the purposes of this project. After some experiments, the detection interval was set to 4 seconds. A time interval of 4 seconds seems a reasonable time for a person to move between divisions. And the mobile nodes only send the messages every second instead of every 200ms, which reduces the chances of sporadic changes of the RPD value. Therefore, it is more important to have a consistent detection with a few seconds delay, that an immediate detection that can vary the results in a very short time. Nevertheless, these values can be confirmed and improved by performing more and better tests.

The position of the fixed nodes is also an important factor that can influence the detection accuracy. During the tests, the network was created with one fixed node per division and the detection had good results. The ideal position should be at an elevated place that has no objects that could be in the way of the radio signals. Therefore, placing the fixed nodes at the ceiling or at the higher part of the walls are theoretically the best solutions. However, most indoor facilities are not prepared to install those fixed nodes at those positions, requiring some wiring installation procedures. And sometimes, some positions might create blind spots and ambiguities, which will require a new installation on another place. An alternative is to just place the nodes at the division's power outlets. Despite having the possibility of objects obstructing the radio signals, it is a more convenient and cheaper solution. If a fixed node is getting a lot of obstructed signals, the fixed node can simply be plugged to another power outlet. Another suggestion for the fixed nodes installation is to place them in positions where a user entering the division will pass relatively close to the fixed node, such as near the entrance of the division. The reason for this placement suggestion, is to get a high RPD bit as soon as possible to prevent the neighbor divisions from influencing the detection. It is also required to place the nodes relatively far from each other, so the divisions could be differentiated. Finding a balance between the distance of the nodes and their position inside each division will give the best results.

An additional feature that could help improving the localization system was implemented at first but then it was discarded. This feature was the possibility of the central node to dynamically change the radio's power output of a selected mobile node. With this approach, the existence of ambiguities could reduce even more. The idea is that when ambiguities appear, the mobile node reduces its radio's power output incrementally until the ambiguities disappear and a more consistent detection is achieved. This solution was abandoned because of the increased complexity that comes with it. The nodes needed to be constantly changing the power output and the network traffic would increase a lot. Besides that, the central node would have to manage all of that, and the performance of the nodes could decrease. However, this feature is still a possibility that can be achieved in the future.

After performing some tests with a simple network created with the fixed nodes connected to the walls power outlets, using the 4 seconds time interval and the confidence points values explained in section 3.4, the detection accuracy was correct most of the time. However, some ambiguities and incorrect results appeared sporadically at certain positions of the mobile nodes. But overall, taking into account all the limitations previously explained, the localization system has a good performance and acceptable accuracy for the purpose of this project.

5.2 Health monitoring

The fall detection algorithm implemented on this system was based on [21, 23] but it has the challenge of the sensor being placed on the wrist, which can cause more false alarms. The detection cannot be tested reliably by simulating real falls because the node is not attached to the wrist and is not made to support major impacts. Simulating normal daily activities like walking, sitting down and getting up, do not trigger an alert because even if acceleration caused by those activities triggers a suspected fall, the inactivity detection that is performed after that, will discard most of the suspicions. Tests were performed by dropping the mobile node from different heights into soft surfaces such as a pillow, a mattress, and a carpet. The algorithm detected almost every fall, but a lot of the detections would probably be false alarms on a real situation. The reason for this, is because the values of g force measured by the sensor on impact were higher than expected. Based on literature [15], the value of g force on fall events is usually between 2.5g to 3.5g, but the MPU6050 sensor measured values around 10g and higher, even when performing drops from small heights. A possible explanation to those results is the mobile node's small mass. As described in Newton's second law of motion, force equals mass times acceleration (equation 3). Therefore, even a small force caused by a small drop can cause a big acceleration of a mobile node that has a small mass. Based on that, attaching the mobile node to a human body is expected to reduce the acceleration measured to values that are closer to those in the literature.

$$F = ma \tag{3}$$

Regarding the inactivity detection, it has a simple algorithm that works as expected. If there is little to no movement detected by the accelerometer within a certain time, an alert is emitted. However, there are still some problems that might appear. False alarms could happen by a bad configuration such as setting the inactivity time range to a period of time that the user might be usually inactive (example: user sleeping during the day). False alarms could also happen when the user is not wearing the mobile device. This situation could happen for example when the user goes to take a bath, the device is removed previously because it is not waterproof. If the user forgets to wear the device after the bath, it will eventually emit an inactivity alert.

The heart rate values measured by the MAX30102 sensor were compared to a heart rate monitoring device and the results were quite similar. However, this implementation has some problems. The first problem is the fact that the sensor is not constantly monitoring the user. When the user wants to know its heart rate value, he needs to wake up the sensor by pressing a button and to place his finger on the sensor. The second issue is that there is no feature that helps the user's finger to stay firmly in place during the measurement period. This can cause incorrect readings that could lead to false alarms if the user's finger moves too much or is placed badly on the sensor.

At first, an LM-35 temperature sensor [43] was planned to be implemented, but later that idea was discarded because the sensor could not measure a reliable and consistent body temperature on the wrist. In order to have a decent measure, the sensor should be attached to another area of the body such as the user's armpit. However, variables such as the length of the sensor wires and trying to attach the sensor to the armpit without causing discomfort to the user were the reason to discard this body temperature feature.

Depending on the alert configurations, the mobile node can activate or deactivate certain monitoring algorithms, which saves processing power. The health monitoring algorithms are light, efficient and fast. This results in a 55% memory occupation of the mobile node's Arduino Nano, which indicates a good performance even if every detection algorithm is configured to be active all the time. This also means that there is some space for more biomedical sensors to be implemented without affecting the mobile node's performance.

5.3 Web application and Server Script

The web application has a simple and intuitive design that helps a user to analyze the data without any previous training. It displays the current data without noticeable delays, updates the network information easily and gives feedback to the user. The communication between the server script and the web application though the REST API also works without any problems or delays. The major disadvantage is the need to restart the server every time a setup procedure is performed. However, it can be studied the possibility of implementing a file in the future that automatically handles this problem.

5.4 Power consumption

The power consumption is important in the development of systems that have mobile devices. Although the objective of this work is to validate the functionality, architecture and proposed approach, the power consumption must be a relevant feature in future iterations of this project. The power consumption of the mobile and fixed nodes was measured through a multimeter. The fixed nodes consume about 35.5mA and the mobile nodes consume 42mA. The difference of the values is mostly due to the accelerometer incorporated on the mobile node. Using a small 2000mAh battery, the mobile node can run for about 48 hours. This autonomy is not the best for this type of system because the user would need to charge the device every 2 days. Some procedures can be performed in order to reduce the mobile node's power consumption. By setting the radio to be asleep for most of the time and only waking up every second to send and receive messages, the power consumption drops to 26mA. This increases the running time of the mobile node to 77 hours using the same 2000mAh battery. However, this solution requires that the alert configuration messages need to be transmitted by the fixed nodes only when the mobile node's radio is awake. This solution was implemented but it was then discarded due to bad performance. The fixed nodes stored all the configurations that were sent by the application and when they receive a message from a mobile node, that mobile node also receives its configuration message. This seemed to be functioning well but when more than one mobile node is connected to the network, the mobile nodes eventually stop receiving the configurations from the fixed nodes. The fixed nodes probably could not handle such message traffic or data collision was occurring. A better algorithm or perhaps, a new radio with better capabilities could solve this issue. Finally, the power consumption could be reduced even more by increasing the time that the radio stays asleep during long inactivity periods such as at night when the user is sleeping. During that period, the radio could only wake up every 5 minutes for example, which will substantially decrease the power consumption.

5.5 Alerts

The alert system is very configurable, which allows to set different configurations according to the characteristics of each user and the mobile nodes can be configured regardless of their position on the network. During the tests, the alert configuration messages reached the desired mobile nodes through the network without problems. However, there are no acknowledgement messages to confirm that the configurations have reached their destination. The reason for that, is because the communications are performed with the multicast feature, which disables the acknowledgement and retransmission capacities of the nRF24L01+ radio. Otherwise, data collisions and missing packets could occur. The same problem applies with the periodic messages sent by the mobile nodes. The solution to this problem is to implement an acknowledgement message that goes from the destination to the origin confirming the reception of the event message and to retransmit if necessary. The biggest problem that prevents this implementation is that multicast is always necessary in the communication between the mobile nodes and the network. However, the power consumption algorithm suggested in section 5.4 is also a good approach to solve this problem if it can be implemented in the future. If a fixed node receives a periodic message from a mobile node, it now knows that mobile node's ID. Therefore, the fixed node can send a direct message with the configurations to the mobile node using the acknowledgments and retransmit capacities of the radio. However, all of this increases the complexity of the algorithms and causes a lot of stress to the network because several fixed nodes may be receiving the multicast messages from the mobile node, which causes multiple acknowledgement messages to reach the central node at the same time. And this is only for the configurations message. Adding acknowledgements to confirm that an alert message has reached the central node increases even more the complexity and stress to the network. This new algorithm will need to handle all of this. Nevertheless, multicast makes multiple fixed nodes to send the same message to the destination, which hopefully can ensure that at least a message is successfully received. If the system can detect the location of a user, it means that it is receiving the mobile node's periodic messages. Therefore, it will also receive any alerts generated by the user. On top of that, when an alert event is generated, the mobile node constantly sends that alert message to the network.

The SMS messages sometimes were duplicated or were received with some delay, but it was no longer than 10 seconds. However, the most relevant disadvantage is that the SMS message alert depends on the internet connection by both the local server and the sending mobile phone.

5.6 Future improvements

5.1.1 Hardware

Future improvements for this system could be the creation of a new mobile node prototype that can actually be attached to a person’s wrist. A PCB designed specifically for this project can create a mobile node much smaller than the existing one. Connecting a small battery and using straps to attach the node to the wrist creates a usable mobile node. The paper [44] shows the mains characteristics of a wearable device for medical purposes. The device must be designed properly in terms of size, weight, performance, and reliability.

Another improvement could be the addition of more sensors and actuators such as a temperature sensor to measure body temperature through the wrist and a buzzer that can be triggered by the alerts. Another suggestion is to replace certain components. The MAX30102 sensor could be replaced with a sensor that could measure continuously the heart rate of a user through the wrist. The SMS alert system has some reliability problems and could be improved by replacing the IFTTT solution with a GSM module connected to the local server. Finally, power efficient components such as a new microcontroller or radio could replace the existing ones. However, changing the mobile node’s components could eventually increase the node’s total cost.

5.1.2 Software

Regarding the health monitoring algorithms, all the parameters such as the fall detection thresholds, the sensor settings, the period of inactivity and the heart rate acceptable range should be calibrated using real life tests. The same applies to the indoor localization algorithm, as the confidence points and the number of iterations should also be calibrated to increase the detection accuracy. More user information could be added to the system and accessed through the web application. That information could be the user’s age, height, weight, medical history and known health problems for example. Finally, implementing an acknowledgment algorithm would help with the communication reliability of this solution.

6 Conclusions

Currently, more and more people are getting older and need more medical care and to be monitored more often. The human body suffers a lot with age, making it more prone to various problems and illnesses. But it is not just the body that suffers with age, the mind also becomes more fragile. Older people tend to get lost and to forget many things such as not taking medication, for example. Nowadays it is possible to monitor the elderly at home or in a nursing home 24 hours a day, and in a non-intrusive way. The person's health condition can then be sent remotely to a medical professional for evaluation. This saves a lot of time and money for both the users and medical facilities and could provide better diagnosis because the user is being monitored in a comfortable environment and doing his daily life.

In this work, several existing solutions and technologies that approach the health monitoring and indoor localization areas were studied and compared. After studying the positive and negative aspects of these solutions, and taking into account existing technologies, a low-cost solution that consists of a wearable device was developed, which can be used in a house or nursing home. The solution implemented performs room-level localization, allows to monitor the heart rate of the user, has an SOS button, and detects falls and inactivity. It then sends alerts that can be received via SMS message to the user's emergency contact. Combining the localization with the health analysis, will help to perform a better and more reliable diagnosis. Suggestions for future improvements to this system were proposed, such as designing a new PCB board, replacing certain components, improving the mobile node's power consumption, and calibrating certain parameters within the indoor localization and health monitoring algorithms.

The results show that even with some limitations and accuracy problems, the proposed system have an acceptable performance and proves that this low-cost approach on wearable health monitoring and indoor localization can be achieved. In the end, by performing some improvements and calibrations, a real life system that has a lower cost compared to other existing solutions could be implemented in the future based on the proposed solution in this work. This type of equipment is very relevant because it allows people to be safer, allowing them to age at home, enjoying greater comfort and better quality of life.

References

1. Zahid Farid, Rosdiadee Nordin, and Mahamod Ismail, “Recent Advances in Wireless Indoor Localization Techniques and System,” Journal of Computer Networks and Communications, vol. 2013, Article ID 185138, 12 pages, 2013.
2. J. Lee, Y. Su and C. Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei, 2007, pp. 46-51
3. Espressif Systems. (2018). ESP8266 Datasheet Arduino Core Documentation. Retrieved from https://datasheet4u.com/datasheet-pdf/EspressifSystems/ESP8266/pdf.php?id=853652 (accessed on 4 January 2020).
4. Digi International, Inc. (2009). XBee/XBee-PRO RF Modules. Retrieved from https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf (accessed on 4 January 2020).
5. Gomez, C.; Oller, J.; Paradells, J. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. Sensors 2012, 12, 11734-11753.
6. JN Huamao Technology Co., Ltd. (2018). Bluetooth 4.0 BLE module Datasheet. Retrieved from http://jnhuamao.cn/index.asp (accessed on 4 January 2020).
7. Peinado, A. (2004). Privacy and authentication protocol providing anonymous channels in GSM. Computer Communications, 27(17), 1709–1715.
8. Chi-Chun Lo, & Yu-Jen Chen. (1999). Secure communication mechanisms for GSM networks. IEEE Transactions on Consumer Electronics, 45(4), 1074–1080.
9. SIMCom. (2009). SIM900A Datasheet Hardware Design. Retrieved from https://datasheet4u.com/share_search.php?sWord=sim900a (accessed on 4 January 2020).
10. Texas Instruments. (2015). CC1101 Low-Power Sub-1 GHz RF Transceiver datasheet (Rev. I). Retrieved from http://www.ti.com/lit/ds/symlink/cc1101.pdf (accessed on 5 January 2020).
11. Texas Instruments support forums. Available online: https://e2e.ti.com/support/wireless-connectivity/other-wireless/f/667/t/217416?cc1101-range-is-very-low. (accessed on 17 December 2020).
12. Nordic Semiconductor. (2008). nRF24L01+ Single Chip 2.4 GHz Transceiver Preliminary Product Specification v1.0. Retrieved from https://lastminuteengineers.com/datasheets/nrf24L01+2.4ghz-transceiver-datasheet.pdf (accessed on 5 January 2020).
13. W. H. O. Ageing and L. C. Unit, WHO global report on falls prevention in older age. Geneva, CH: World Health Organization, 2008.
14. Khan, S.S.; Hoey, J. Review of fall detection techniques: A data availability perspective. Med. Eng. Phys. 2017, 39, 12–22.
15. Khojasteh, S., Villar, J., Chira, C., González, V., & de la Cal, E. (2018). Improving Fall Detection Using an On-Wrist Wearable Accelerometer. Sensors, 18(5), 1350.
16. Abbate, S.; Avvenuti, M.; Bonatesta, F.; Cola, G.; Corsini, P.; Vecchio, A. A smartphone-based fall detection system. Pervasive Mob. Comput. 2012, 8, 883–899

17. Hsieh, S. et al. “A wrist -worn fall detection system using accelerometers and gyroscopes.” Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control (2014): 518-523.
18. Casilari, E.; Santoyo-Ramón, J.A.; Cano-García, J.M. UMAFall: A multisensor dataset for the research on automatic fall detection. Procedia Comput. Sci. 2017, 110, 32–39.
19. Bagala, F.; Becker, C.; Cappello, A.; Chiari, L.; Aminian, K.; Hausdorff, J.M.; Zijlstra, W.; Klenk, J. Evaluation of accelerometer-based fall detection algorithms on real-world falls. PLoS ONE 2012, 7, e37062.
20. Chaudhuri, S.; Thompson, H.; Demiris, G. Fall detection devices and their use with older adults: A systematic review. J. Geriatr. Phys. Ther. 2014, 37, 178–196)
21. Degen, T., Jaeckel, H., Rufer, M., & Wyss, S. (n.d.). SPEEDY:a fall detector in a wrist watch. Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.
22. Kumar, H. R., Janardhan, S., Prakash, D., & Prasanna Kumar, M. K. (2018). Fall Detection System using Tri- Axial Accelerometer. 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT).
23. Apple Watch. Helping Your Patients Identify Early Warning Signs. Available online: https://www.apple.com/healthcare/apple-watch/ (accessed on 16 December 2019).
24. P. Lukowicz, U. Anliker, J. Ward, G. Troster, E. Hirt, and C. Neufelt, “AMON: A wearable medical computer for high risk patients,” in Proc. 6th ISWC, Oct. 7-10, 2002, pp. 133–134.
25. Libelium Comunicaciones Distribuidas S.L. (2019). MySignals Sw eHealth and Medical IoT Development Platform Technical Guide. Retrieved from http://www.libelium.com/downloads/documentation/mysignals_technical_guide.pdf (accessed on 5 January 2020).
26. ASCAS. (2015). HealthBand: A Remotely Monitored Health Status Bracelet. Retrieved from https://www.instructables.com/id/HealthBand/ (accessed on 30 December 2020).
27. FrailSafe Project. Available online: https://frailsafe-project.eu/ (accessed on 5 January 2020).
28. T. Tegou, I. Kalamaras, K. Votis and D. Tzovaras, "A low-cost room-level indoor localization system with easy setup for medical applications," 2018 11th IFIP Wireless and Mobile Networking Conference (WMNC), Prague, 2018, pp. 1-7.
29. M. G. Tsipouras, N. Giannakeas, T. Tegou, I. Kalamaras, K. Votis and D. Tzovaras, "Assessing the Frailty of Older People using Bluetooth Beacons Data," 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Limassol, 2018, pp. 5-11.
30. Sensoro. Available on https://www.sensoro.com/en/Beacons. (Accessed on 6 January 2020).
31. Smartex. Available online: http://www.smartex.it/en/ (accessed on 5 January 2020).
32. T. Khan and M. K. Chattopadhyay, "Smart health monitoring system," 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), Indore, 2017, pp. 1-6.
33. Patil, V., Thakur, S. S., & Kshirsagar, V. (2018). Health Monitoring System Using Internet of Things. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS).
34. Piyare, Rajeev, and Seong-ro Lee. "Performance analysis of XBee ZB module based wireless sensor networks." International Journal of Scientific & Engineering Research 4.4 (2013): 1615-1621.

35. RF24 Library. Available online: http://tmrh20.github.io/RF24/ (accessed on 18 December 2020).
36. RF24Network Library. Available online: http://tmrh20.github.io/RF24Network/ (accessed on 18 December 2020).
37. howtomechatronics.com. Available online: https://howtomechatronics.com/tutorials/arduino/how-to-build-an-arduino-wireless-network-with-multiple-nrf24l01-modules/ (accessed on 5 January 2020).
38. InvenSense Inc. (2013). MPU6050 Datasheet. Retrieved from https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf (accessed on 15 December 2020).
39. Maxim Integrated Products. (2015). MAX30102 Datasheet. Retrieved from https://www.alldatasheet.com/datasheet-pdf/pdf/859400/MAXIM/MAX30102.html (accessed on 16 December 2020).
40. React on Django: Getting Started. Available online: https://blog.usejournal.com/react-on-django-getting-started-f30de8d23504 (accessed on 16 December 2020).
41. IFTT helps every thing work better together. Available online: https://ifttt.com/ (accessed on 16 December 2020).
42. Bootstrap. Available online: https://getbootstrap.com/ (accessed on 29 December 2020).
43. Texas Instruments. (2017). LM-35 Datasheet. Retrieved from https://www.ti.com/lit/ds/symlink/lm35.pdf (accessed on 16 December 2020).
44. E. Jovanov T. Martin and D. Raskovic. “Issues in wearable computing for medical monitoring applications: A case study of a wearable ecg monitoring device.” In Digest of Papers. Fourth International Symposium on Wearable Computers. IEEE Comput. Soc, 2000.