iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA

PhysioEnabler - Intelligent sensor system to aid motor rehabilitation with a web application

Pedro Miguel Assis Batoca

Master in Telecommunications and Computer Engineering

Supervisor
PhD Octavian Adrian Postolache, Associated Professor (with Aggregation)
Iscte - Instituto Universitário de Lisboa

Co-Supervisor
PhD Américo Manuel Carapeto Correia, Full Professor
Iscte - Instituto Universitário de Lisboa

November, 2022

iscte
TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

PhysioEnabler - Intelligent sensor system to aid motor rehabilitation with a web application

Pedro Miguel Assis Batoca

Master in Telecommunications and Computer Engineering

Supervisor
PhD Octavian Adrian Postolache, Associated Professor (with Aggregation)
Iscte - Instituto Universitário de Lisboa

Co-Supervisor
PhD Américo Manuel Carapeto Correia, Full Professor
Iscte - Instituto Universitário de Lisboa

November, 2022

Acknowledgment

I would like to thank my family and close friends for the encouragement and support to finish this dissertation.

A word of appreciation to my supervisor, Professor Doctor Octavian Postolache for his availability, support, and commitment so that I could finish my master's degree.

Lastly, I express my thank to Instituto de Telecomunicações (IT) and ISCTE-Instituto Universitário de Lisboa for supplying the needed material for this project.

Resumo

A reabilitação física é um tema atual devido ao envelhecimento da população em geral, mas também ao aumento de doenças crónicas, havendo já diversas iniciativas e estudos para encontrar soluções inovadoras nesta área. Um dos maiores desafios é a falta de dados que possam ajudar a diagnosticar e tratar de forma mais adequada os pacientes.

Os dispositivos de ajuda à mobilidade mais comuns permitem melhorar o dia a dia dos pacientes, na medida que lhes providenciam uma maior independência nas suas atividades, mas, não permitem a recolha de dados para análise. Com os avanços nas tecnologias de IoT, é possível dotar estas ajudas com sensores e outros dispositivos de modo a extrair dados que permitam aos fisioterapeutas tomar melhores decisões e influenciar positivamente o tratamento de um paciente.

O protótipo apresentado nesta dissertação propõe o uso de sensores de força, IMU (sensor que combina acelerômetro, giroscópio e magnetómetro) e RFID (permite a identificação a partir de sinais rádio) a uma canadiana e utilizar um microcontrolador ligado aos sensores para extrair esta informação de modo a enviá-la para um endpoint na cloud via protocolo MQTT. O processamento destes dados é feito por cloud functions que também armazenam o resultado. A informação é disponibilizada tanto para fisioterapeutas como para pacientes num front-end desenvolvido em Python. A aplicação permite também que sejam criados planos de tratamento customizados de acordo com as necessidades de cada paciente que também podem ser consultados pelos vários utilizadores incluindo o fisioterapeuta e o paciente, utilizador da canadiana inteligente.

Palavras-chave: Internet das coisas, Internet das coisas médicas, Sensores inteligentes, Reabilitação física, Aplicação web, Cloud

Abstract

Physical rehabilitation is a current topic due to the global aging population and an increase in chronic diseases and there are several initiatives and studies to bring new and innovative solutions in this area. One of the main challenges is the lack of data that can help to diagnose and provide more adequate treatments to patients.

The current walking aids help improve the patients' day-to-day by providing greater independence in their activities but don't allow the extraction of any objective data for analysis. With the advances in IoT technologies, it is possible to enhance the aids' functionality with sensors and other devices to extract information that can help physiotherapists improve their decisions and influence the patients' course of treatment.

The prototype presented in this dissertation proposes to add force, IMU (sensor that combines accelerometer, gyroscope, and magnetometer), and RFID (allows the identification with radio signals) sensors to a crutch and to use a microcontroller connected to these sensors to extract the data and send it to an endpoint in the cloud via MQTT protocol. The data processing takes place with cloud functions that also store the results. The information is available for patients and physiotherapists to view and analyze in a front-end developed in Python. The application also allows the creation of custom exercise plans according to the patient's needs and is available for physiotherapists and patients to view.

Keywords: Internet of Things, Internet of Medical Things, Smart Sensors, Physical Rehabilitation, Web Application, Cloud

Contents

CHAPTER 1

Introduction

Physiotherapy is a science dedicated to improving a patient's physical condition that can be related to a physical or nervous issue [**1**]. Physical therapy training needs can come due to natural aging, accidents, events such as AVCs, degenerative diseases, or others [**1**]. In this dissertation, the focus is the development of compatible smart IoT equipment that can aid gait rehabilitation during planned training sessions.

According to a 2020 study from the United Nations, the number of people aged 65 or older surpasses 727 million worldwide and estimates that by 2050, this number will be 1.5 billion [**2**]. World Health Organization report on falls prevention in older age also points out that with natural aging, the likelihood of falls by elderly people also increases, and usually when it happens, there is a need for physical rehabilitation [**3**]. The number of people living with chronic disease also keeps increasing, making this a mandatory topic [**4**]. Estimates indicate that one in three people would benefit from rehabilitation however it isn't available to everyone [**4**]. According to WHO in low and middle-income countries, half of the population doesn't have access to it [**4**]. To address this need, WHO created an initiative called Rehabilitation 2030 to challenge stakeholders worldwide to expand the funding for health systems, improve research to develop new solutions, and expand the availability of these services to the population worldwide [**4**].

The always-increasing number of patients is not the only challenge that physiotherapists face. There is also the lack of a system that allows an individual to use a crutch or other type of aid and automatically obtain, save and analyze the data generated by the used equipment [**5**]. This information will allow physiotherapists to improve their decisions regarding the treatment, which can lead to a faster recovery for the patient.

The goal of this dissertation is to create a platform capable of helping patients by tracking training sessions, allowing a physiotherapist to evaluate the patient progress.

1.1. Objectives

The goal of this dissertation is to improve the data that is available to physiotherapists from physical rehabilitation sessions. To achieve this it is proposed the development of an IoT system composed of a crutch with a microcontroller and sensors attached to it. This crutch has a companion web application capable of displaying the collected data and creating treatment plans for the patient. To meet this goal the following objectives need to be attained:

- Create a prototype of a smart crutch capable of extracting the patient's metrics during the session.

- Employ an RFID sensor to identify the patient.
- Design and implement algorithms to extract metrics from the sensor's data.
- Training results presentation

To fulfill the mentioned objectives, the following tasks are considered:

- Develop and deploy a smart sensor on the crutch level.
- Development of a software module to extract sensors' data and send it to cloud storage.
- Definition and implementation of an endpoint to receive the data and store the data in the defined storage.
- Design and implementation of a user management solution.
- Development of a web application to display the information to the users.

1.2. Structure of this work

This dissertation is divided into four main chapters. Chapter 2 provides an overview of the current state of the art of the solutions already developed to tackle this need. It illustrates solutions similar to the one proposed in this work, using several pieces of equipment for rehabilitation sessions such as a crutch, walker, cane, shoe, and a tao ball. A comparison between application protocols is also presented to support the protocol choice. Chapter 3 presents the system architecture, showcasing the microcontroller, sensors, and underlying services used for communication, computing, and storage. The application is also described as well as the functionalities available. Chapter 4 provides the results of the several use cases used in this dissertation. Chapter 5 presents the conclusions from this work and future work.

CHAPTER 2

State of the art

This chapter presents the current state of the art and approaches of IoT systems designed to improve rehabilitation treatments. Section 2.1 describes the concept of IoT with an emphasis on the internet of medical things and its impact on society. Section 2.2 presents the application protocols considered for this work, HTTP, MQTT, and AMQP, and a comparison between them. The final section, 2.3, provides an overview of similar solutions that use a crutch, a walker, and a cane.

2.1. Internet of things - IoT

The Internet of Things describes the network of physical objects or things, usually embedded devices with sensors running software that can connect and exchange data with other devices or systems over the internet [**6**]. IoT devices assist society in multiple fields such as healthcare, education, industry, autonomous driving, and smart cities [**6**]. The average consumer can also leverage these systems in their house with ordinary objects such as bulbs (e.g. Philips hue) or refrigerators (e.g. some LG models). Studies indicate that the IoT market is growing with an expected 16.44 billion devices by 2025 [**7**] and projected revenue growth from 389.2 billion in 2020 to 677.4 billion by 2025 [**8**].

2.1.1. Internet of medical things - IoMT

The growth in the number of devices and revenue for IoT systems paved the way for the development of new solutions in healthcare that was named the Internet of Medical Things or IoMT. IoMT systems are mainly composed of sensors and electronic circuits to acquire biomedical signals from a patient, a processing unit to process the biomedical signals, a network device to transmit the biomedical data over a network, a temporary or permanent storage unit, a visualization platform with artificial intelligence schemes to take decisions according to the convenience of a physician [**5**]. There are several examples of the usage of this kind of systems such as remote health monitoring or fall detection for elderly people. Remote monitoring can be achieved by using low-power and weight sensors attached to the patient that records vital information such as heart rate, respiration rate, and blood oxygen saturation. This data is then uploaded to an endpoint to be processed and displayed on a platform to be analyzed [**5**]. To detect the fall of elderly people it is possible to use real-time monitoring systems that use sensors such as a gyroscope, an accelerometer, and a vibration one to detect if the patient fell and alert the authorities [**5**].

2.2. Microcontroller

This section presents three models of small form factor microcontrollers that were considered for the smart crutch prototype. The three candidates were the Arduino Nano 33 IoT, Raspberry Pi Pico W, and ESP32.

2.2.1. Arduino Nano 33 IoT

The Arduino Nano 33 IoT was launched in 2019 as part of the nano family, the small form factor models from Arduino. This model uses an Arm Cortex-M0+ dual-core processor, the RAM is 256KB, a clock speed of 48 MHz, a flash memory of 1 MB, and an operating voltage of 3.3 V. For connectivity, it supports WI-FI, Bluetooth, and Bluetooth low energy, 14 digital pins for input/output, 8 analog pins and for development the C/C++ languages. It supports 1 UART, 1 SPI, and 1 I2C, offers a 6-axis IMU capable of tracking the movement of small objects, and a security chip capable of offering IoT security features such as store endpoints and certificates/keys used for connectivity [**9**] [**10**].

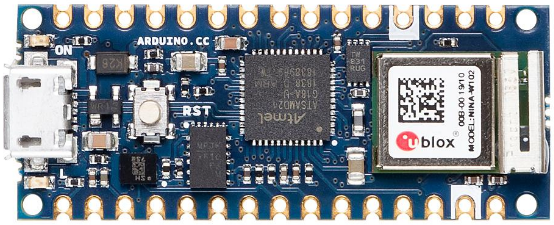


Figure 1. Arduino Nano 33 IoT [**11**]

2.2.2. Raspberry Pi Pico W

Raspberry Pi Pico W was launched in 2022, and it was the iteration of the first microcontroller launched by the Raspberry Pi Foundation, the Raspberry Pi Pico. Both Raspberry Pis use the RP2040 chip that provides an Arm Cortex-M0+ dual-core processor. The RAM is 264KB, the clock speed is 133MHz, the flash memory is 2 MB, the operating voltage is between 1.8-5.5V DC, and only the W model offers WI-Fi. In terms of GPIO, it offers 26 pins plus 3 analog pins, and for other interfaces, it has 2 UART, 2 I2C, 2 SPI, and 16 PWM channels. For firmware development, it supports both MicroPython and C/C++ [**12**].

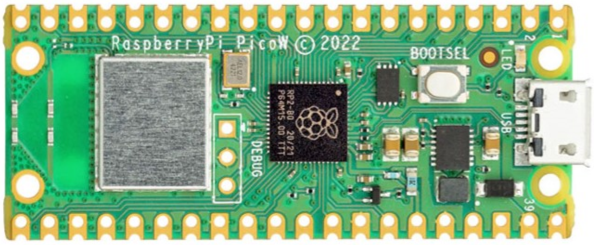


Figure 2. Raspberry Pi Pico W [**13**]

2.2.3. Esp 32

ESP32 is a microcontroller launched by ExpressIf in 2015 and the successor to the ESP8266. This model uses a Tensilica Xtensa LX6 32-bit dual-core processor, the RAM is 520KB, a clock speed of up to 240 MHz, a flash memory of 4 MB, and an operating voltage between 2.2-3.6V. In terms of connectivity, it offers WI-FI, Bluetooth, and Bluetooth low energy, and the GPIO offers 34 programmable pins. It also supports 2 I2S, 2 I2C, 3 UART, and 4 SPI and has 16 PWM channels. For development, it supports both MicroPython and C/CC++ [**14**].

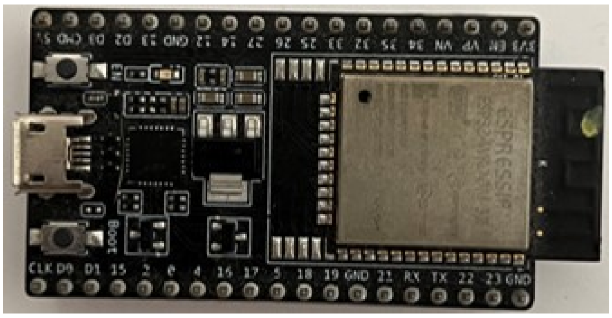


Figure 3. Esp32

2.2.4. Remarks

Comparing the three devices, all satisfy the requirements for the prototype. In terms of pricing, Raspberry Pi Pico W costs 7.49€, the ESP32 15.38€, and the Arduino 20,60€ [**15**] [**16**] [**17**]. The choice was the ESP32 because it allows connecting more sensors for the future due to having more analog, I2C, and SPI pins, it offers Bluetooth connecitivity, and it was already available to use, so there was no need to purchase. The Arduino would be a good choice if the IMU wasn't aimed at small objects because it would be cheaper than having a stand-alone IMU sensor. However, it would be difficult to add more sensors because there wouldn't be any I2C and SPI interfaces available.

2.3. MicroPython vs C for firmware development

MicroPython was developed in 2013 by Damien George and is an implementation of Python focused on microcontrollers. It shares the same advantages that Python has, its ease to read and write, several available libraries that can help to develop faster and is memory safe [**18**].

C is a programming language that was created in the 1970s and is currently the preferable way to program microcontrollers [**19**] [**18**]. However, it has a steeper learning curve when compared MicroPython, and is more challenging because there are missing features that many modern languages have, such as garbage collection. However, it is much faster than others, for instance, python.

Comparing both, even though the MicroPython can be easier to use for developing and testing quickly it has performance disadvantages when compared to C. As the author of [**18**] shows, the performance deficit is 45% when testing the operations: floating point;

CRC computation; Hash computation for the same hardware. For the smart crutch prototype, the firmware was developed in C due to the performance gains and availability of libraries and documentation to implement the needed functionalities.

2.4. Application layer protocols

This section presents the application protocols under consideration for this dissertation, HTTP, MQTT, AMQP, and the rationale to choose between them.

2.4.1. HyperText Transfer Protocol (HTTP)

HTTP is an application-level protocol that implements a client-server architecture and uses the TCP protocol [**20**]. It is unidirectional and synchronous and provides methods such as GET, POST, PUT, and DELETE. To send new data to the server the clients need to establish a new connection leading to higher overhead and more latency for IoT systems [**21**]. Another article shows that HTTP can consume up to four times more CPU than MQTT or AMQP [**20**].

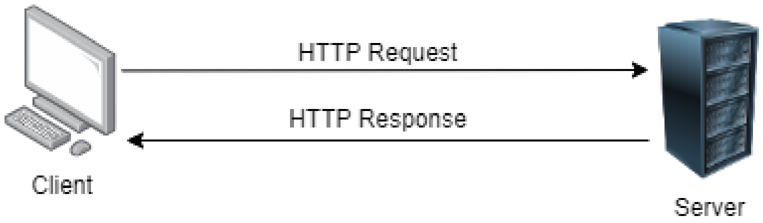


Figure 4. HTTP message diagram

2.4.2. Message Queue Telemetry Transport (MQTT)

MQTT leverages a publish-subscribe architecture and is meant to be used by small, low power, low cost, and low memory devices in high latency, low-bandwidth, or unreliable networks making it the most appropriate for IoT [**20**]. MQTT is based on TCP/IP, thus providing reliable communication for unstable links [**20**]. It has three main components:

(1) The MQTT Broker is a server that receives the messages from the publisher and forwards them to the appropriate subscribers by checking the topic subscribed to [**21**]
(2) The publisher is the one who sends the message (data to be shared) and the topic (the string that the MQTT broker uses to route the messages to the appropriate subscribers) [**21**].
(3) The subscriber receives messages from the MQTT Broker related to the subscribed topic [**21**].

MQTT provides three levels of Quality of Service (QoS), 0 - at most once, 1 - at least one, and 2 - exactly once [**20**]. The first level does not provide any confirmation that the message is received [**22**]. Level two guarantees that message is received by sending the message until it receives the confirmation from the broker via a puback message [**22**]. The last level assures that the message is received exactly once and there are no lost messages however, the end-2-end delay is expected to be higher [22].

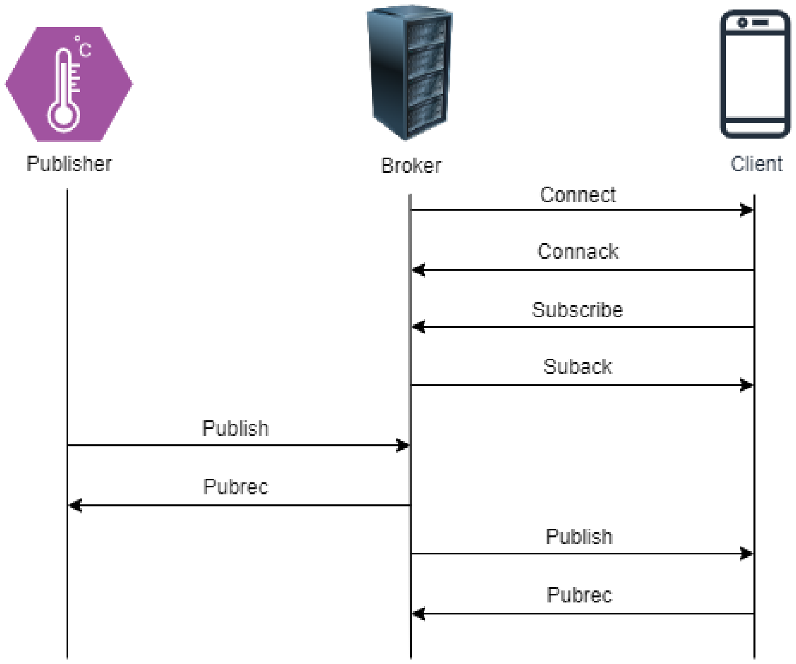


Figure 5. MQTT message diagram

2.4.3. Advanced Message Queuing Protocol (AMQP)

AMQP is a protocol created in 2003 by John O'Hara at JPMorgan Chase that leverages request/response or publish/subscribe architecture and is designed for reliability, security, and interoperability. For the transport layer, TCP is used, and TLS/SSL and SASL for security [21]. To send data, either the publisher or the consumer must create an exchange with a given name and broadcast it to the network. Consumers are required to create a queue and attach it to the exchange [21]. When the broker receives a new message, it transfers it to the exchange that will route it to the queue and is kept there until the customer retrieves it, unlike MQTT [23]. In terms of QoS, it offers two levels, Unsettle Format (not reliable and at least once) and Settle Format (reliable and at most once) [21].

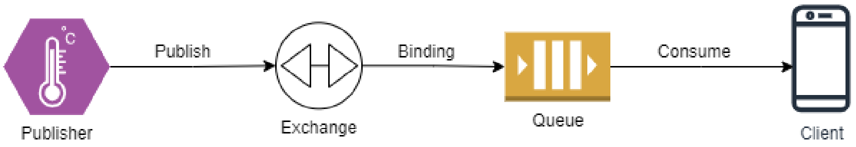


Figure 6. AMQP diagram

2.4.4. Remarks

Comparing the protocols, HTTP is the more CPU hungry and the one that has a higher message overhead. MQTT is similar in terms of CPU usage, however, has a lower message overhead and requires less bandwidth and resources when compared to AMQP, thus providing a better battery life. With these points in mind, MQTT was the chosen protocol.

2.5. Walking aids

This section presents the importance of walking aids and other innovative solutions that share the same goal as this dissertation, which is to tackle the lack of data that physiotherapists have at their disposal when making treatment decisions for their patients.

2.5.1. Why are walking aids so relevant?

Walking aids help elderly people or patients with impairment in the lower limbs. For patients, these aids can be temporary until they recover and don't need them anymore. However, for the elderly, these devices can provide them greater independence in daily activity. [**24**] There are various types of equipment, crutches, walkers, cranes, and standing aids that are prescripted by physiotherapists according to the patient's needs [**25**]. The importance of walking aids can't be ignored since some studies concluded that in case of a fall, patients without equipment are more prone to sustain severe injuries [**25**]. Like everything in life, there are also downsides, mainly that these devices are sometimes hard to use and require great strength which can tire the patient in day-to-day activities [**25**], and also don't provide metrics to be analyzed by the physiotherapist. So there is room for improvement on these devices.

2.5.2. Smart Crutch

In [**26**], is presented a system with three main components a crutch, a mobile application, and a remote database. The smart crutch features an Arduino Nano and sensors attached to it, such as IMU, a force sensor, and a load cell. The generated data allows to compute the metrics of orientation and balance of the patient and the force applied in the crutch. The information is shared with the mobile application via Bluetooth which, then stores it in a remote database. The application allows the visualization of the data in real time and track the overall progress of the patient. As mentioned, improvements can be made by allowing more types of equipment or the use of two crutches simultaneously. The main limitations of this solution are the lack of exercise plans, the need to start and end the session on the application, and that this solution is only available for physiotherapists which do not allow for the patients to perform sessions outside the clinic.

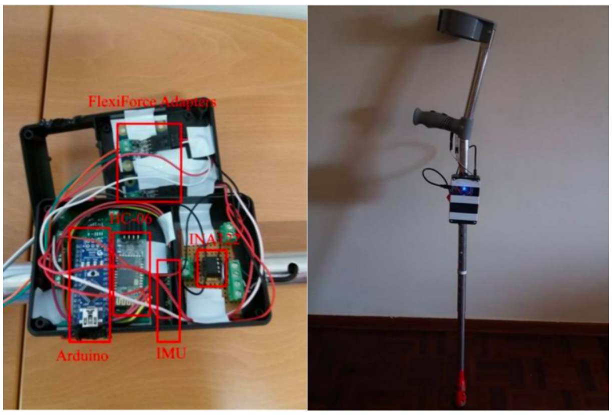


Figure 7. Smart crutch prototype [**26**]

2.5.3. Smart Walker

In [27] is presented a system with four main components, an RFID module, a walker, a web application, and a mobile application. The module registers the RFID from the patient's card into the database. The walker has an Arduino Mega and several sensors attached, including an IMU, ultrasonic range sensors, and load cells. The generated data allows to calculate the metrics of the balance of the patient. For instance, the bilateral elevation of the walker, number of steps, and force exercise in the walker's feet. There are two ways to share the information, the first is when the physiotherapist or the patient starts the session using an RFID card, and the data is sent via HTTP and can only be analyzed at the end of the session. The other is when the physiotherapist starts the session on the mobile application. In this case, the data is available in real-time and is sent to the web server. Both web and mobile applications offer the possibility to create and view patients, view treatment plans and search past sessions of each patient. The real-time view of the session and the patient's progression is only available in the mobile application. The improvements suggested are using MQTT as the communication protocol and replacing the RFID sensor with a fingerprint one.

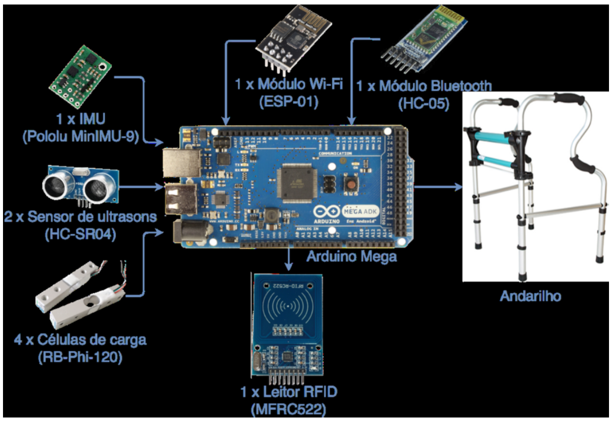


Figure 8. Smart walker prototype [27]

In [28] there is another example of a Smart Walker with different functionalities. The prototype offers three features, which are a walking aid, sit-to-stand aid, and an electric scooter. In the walking aid functionality, the walker measures the applied force, determines the motion commands such as speed and acceleration, and allows the walker to move as if it was a lightweight passive walker. The sit-to-stand feature has to be triggered by the patient and allows to support a part of the user's weight thus reducing the load on the lower limbs, which can be a more effective way of performing this exercise. Lastly, the electric scooter allows going from place to place. There is a GUI that provides overall function control, exercise database management, and status monitoring. Additionally, there are two push buttons for emergency stops. Tests show that the system's main targets are patients with muscle weakness or spinal cord injuries.

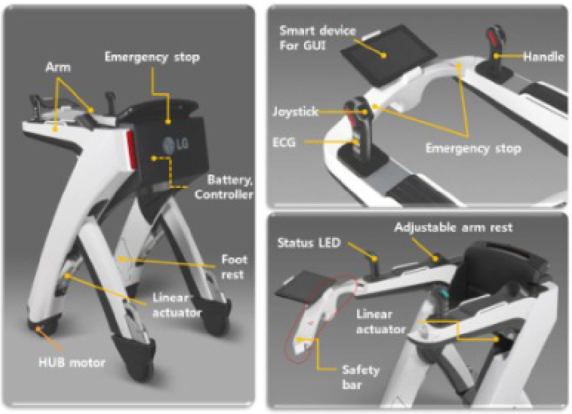


FIGURE 9. Smart walker prototype [**28**]

2.5.4. Smart Cane

In [**29**] is another approach to improve treatments using a low-cost system and a popular tool, a cane (or other variations like a quad-cane). It leverages a Gizduino (an Arduino-based microcontroller) with a load cell and an accelerometer attached. This system monitors the weight-bearing performance of the patient by using a sound alarm to increase or decrease the force applied to the cane. Sessions start automatically when the cane is moved and the data is uploaded by Bluetooth to be analyzed later in a GUI developed in StampPlot Pro (an open-source program). The main disadvantages of this solution are the lack of treatment plans, the lack of a mobile or a web application to view the session data, and accidental movements of the cane that can start a new session (e.g. cleaning).

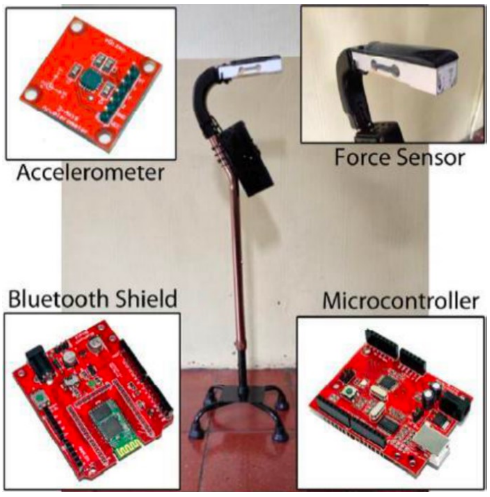


FIGURE 10. Smart cane prototype [**29**]

CHAPTER 3

System description

3.1. Overview

The goal of this system is to provide data to physiotherapists so that they can objectively analyze the patient's condition, track their progress, and ultimately help accelerate the treatment. The high-level architecture used is shown below:

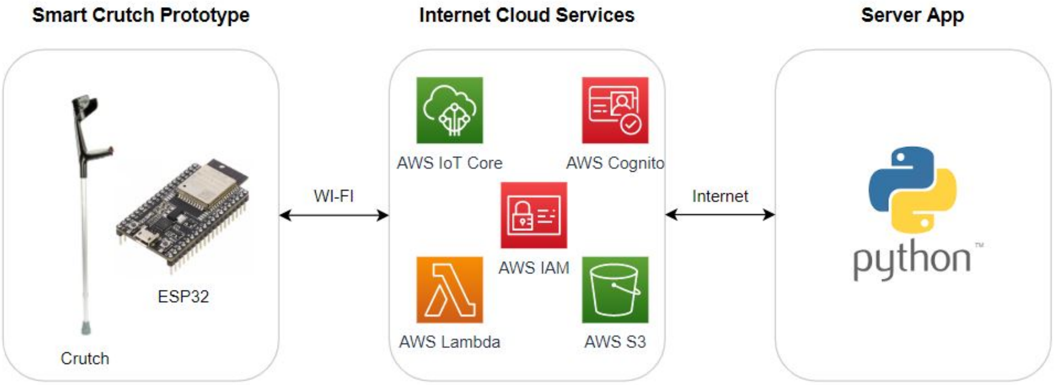


FIGURE 1. Physical Therapy Gait Assessment: High Level Architecture

As seen in the figure above, this system can be divided into three main blocks. The first is the smart crutch prototype, the second is the cloud services, and the final is the server app. The smart crutch prototype includes a microcontroller and a set of sensors. These sensors extract dynamic information about force and accelerations that allow the calculation of specific metrics. The data is necessary for patient gait analysis and is aggregated by the microcontroller and then sent to the cloud services via WI-FI to be processed. The cloud services are responsible for the MQTT broker, user management, permission management, data processing, and storage. Amazon Web Service (AWS) Identity and Access Management is the service responsible for the permission management for AWS services and resources [**30**]. AWS IoT Core provides connectivity, scalability, and management for IoT devices easily and reliably without physical servers [**31**]. It supports MQTT, HTTPS, MQTT over WebSocket, and LoRaWAN as communication protocols [**31**]. MQTT provides less overhead than using MQTT over WebSockets and for this reason, it was the choice [**32**]. AWS Lambda is a serverless cloud computing service that runs code responding to an event [**33**]. AWS Simple Storage Service or S3 is a cloud object storage service that provides high availability and redundancy storage at a competitive price [**34**]. AWS Cognito is a cloud-native user authentication and management service that can be integrated into custom applications simply [**35**]. The last block is the server app built in Python that provides access to the session data for the patients and physiotherapists as well as training plans and administrative tasks for physiotherapists and administrators..

3.2. User types

The application has three different types of users with different rights regarding the system. Patients are allowed to view and change their profiles and view past sessions and training plans. Physiotherapists can create new patients, and view the entire patient list, training plans, and session data where they can also add notes. The other type of user is the administrator which manages the physiotherapists.

3.3. System flow diagram

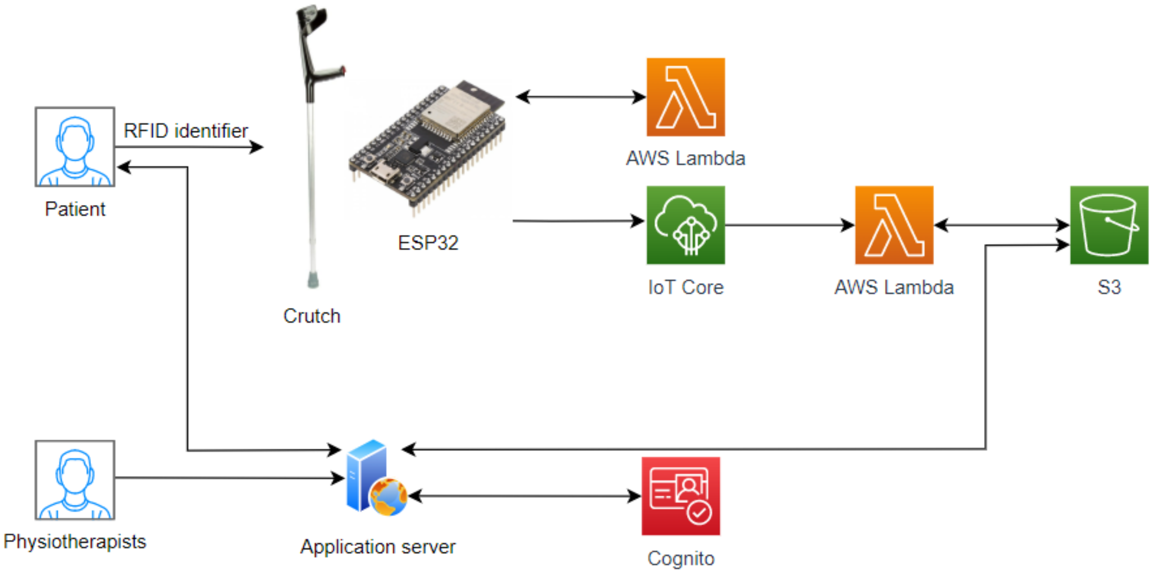


FIGURE 2. System flow diagram, patients use an RFID tag to start sessions and AWS Cognito to login into the application

The flow diagram below allows to better understand how the system works and how to start a new session. Firstly, the patient has to approach his RFID tag or card to the RFID reader embedded in the crutch, and then the system makes a call to a lambda function with the RFID identifier as input to fetch the interval to use between measurements for that patient. Afterward, the session starts and the patient does the exercises. The data is sent using the MQTT protocol to the IoT Core service that triggers another lambda to process the data and store it on cloud storage. To finish the session, the patient has to approach the RFID identifier to the crutch. The metrics are then extracted from the raw data by a third lambda that will be detailed in the software chapter. Patients and physiotherapists can access the session data by logging in to the application server with their credentials. The user profiles are automatically assigned when logged in and can perform the actions described in the previous section.

CHAPTER 4

Hardware

This chapter presents the hardware used in this dissertation's prototype. It leverages several sensors connected to a microcontroller that acquires the data necessary for metrics calculation. The sensing part is composed of an RFID reader RC522, RFID tags, a FlexiForce A201, a load cell SNC2C6, and the AltiIMU 10 v-4. To aggregate the generated data and send it to the cloud, the ESP32 was the chosen microcontroller.

4.1. Sensors

This section presents the models and specifications of the RFID reader, RFID tags, flexiForce sensor, flexiForce adapter, load cell, INA 22, and AltiIMU 10 v-4 used on the prototype.

4.1.1. RC522 RFID Sensor

RFID or Radio Frequency Identification is used to identify the patient and has two components, a transponder or tag inside an object to be identified and a transceiver or reader [**36**]. The RC522 is a low-cost RFID reader that generates a 13.56MHz electromagnetic field that is the standard frequency used by the RFID tags, has a read range of 5 centimeters, is compatible with the interfaces SPI, I2C, and UART [**36**].

Operating Voltage	2.5V to 3.3V
Max. Operating Current	13-26mA
Min. Current(Power down)	10µA

Table 1. RC522 RFID reader specs [**36**]

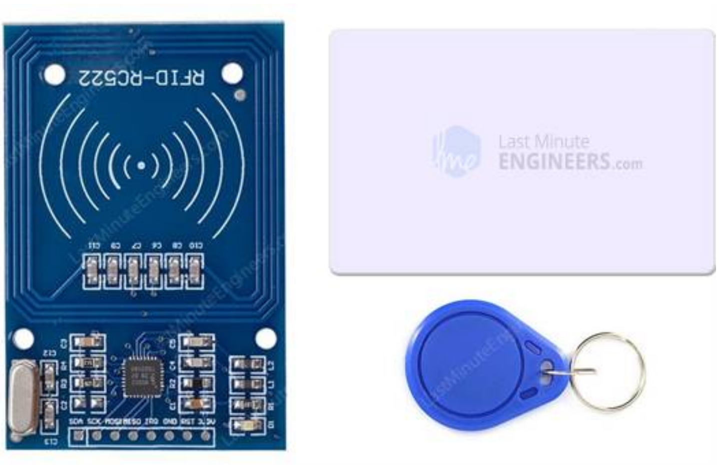


Figure 1. RC522 sensor and identifiers [**36**]

4.1.2. Load Cell SNC2C6

To measure the force applied to the crutch a load cell is used. These sensors convert a mechanical force into an electric signal (voltage) and usually use a Wheatstone bridge as a conditioning circuit [**37**]. The SNC2C6 load cell sensor, figure 2, was the choice for this prototype since it is accurate and has a measurement range of up to 50 Kg or 490N. This range can be easily increased to 100 kg if a patient needs it. However, taking into account the weight distribution, 50 kg is considered appropriate [**38**]. However, it has a very low sensibility of only 0.8 - 2.0 mV/V, thus providing a very low output signal voltage. To tackle this the sensor is coupled with an INA122, a precision instrumentation amplifier for accurate, low noise differential signal acquisition that helps to provide a finer output voltage [**39**]. The gain of the INA122 is calculated by the equation G = 5 + 200 kΩ / R. The resistance used had 1 kΩ so the gain provided is 205.

Non-linearity	±0.3% of Full Scale Repeatability ±0.3% of Full Scale
Hysteresis	±0.3% of Full Scale Precision ±0.1% of Full Scale
Sensitivity	2.0 ±0.1 mv/V
Operating Temperature	20ºC to 70ºC

TABLE 2. SNC2C6 load cell specs [**38**]

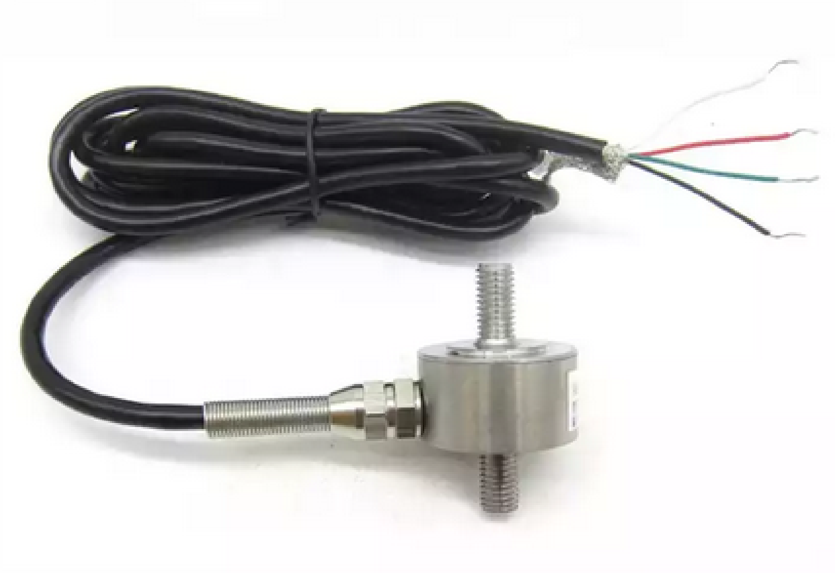


FIGURE 2. Load cell sensor [**38**]

4.1.3. FlexiForce A201

The FlexiForce A201, figure 3 is a thin, flexible, and non-intrusive piezoresistive force sensor that is ideal for assembling proof of concepts [17]. It can measure forces up to 445

N or 45 Kg in a low response time of less than 5µsec while providing the durability of over 3 million actuations [40]. This sensor was used to measure the force applied to the hand support of the crutch.

Linearity (Error)	<±3% of Full Scale (Line drawn from 0 to 50% load)
Repeatability	<±2.5% (Conditioned Sensor, 80% of Full Force Applied)
Hysteresis	< 4.5% of Full Scale (Conditioned Sensor, 80% of Full Force Applied)
Drift	<5% / logarithmic time (Constant Load of 111 N)
Response Time	<5µsec
Operating Temperature	-40°C - 60°C

Table 3. FlexiForce A201 specs [40]

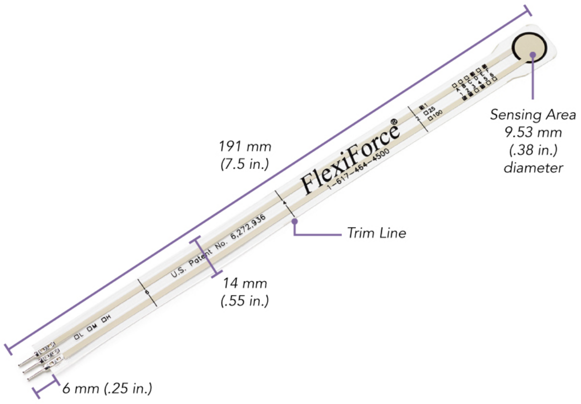


Figure 3. FlexiForce A201 sensor [41]

In order to reduce the noise from the measurements, a FlexiForce Adapter, figure 4, was added. It also allows for the sensor to be powered by the microcontroller and allows it to read the data from the sensor in an analog port making it easier to extract sensor data.

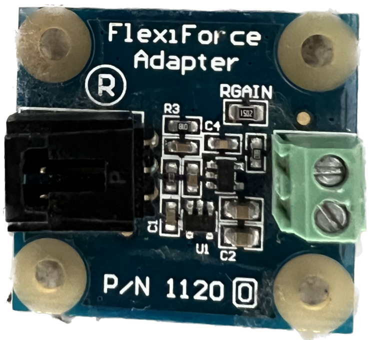


Figure 4. FlexiForce Adapter

4.1.4. AltiIMU 10 v-4

With this prototype, one of the goals was to track the patient's movement, the walked steps, and balance. To be able to retrieve these metrics there was the need for an inertial measurement unit or IMU because this type of sensor combines sensors such as an accelerometer and gyroscope capable of recording the patient's movement. The choice was the Pololu AltIMU-10 v4, figure 5, an IMU board that combines an LPS25H digital barometer, L3GD20H 3-axis gyroscope, and LSM303D 3-axis accelerometer and 3-axis magnetometer thus providing this sensor a 10 degree of freedom (10DOF) [**42**]. It will be responsible for providing the data used to calculate the orientation and steps of the patients. This sensor uses I2C to communicate with reads to the microcontroller and each read for the gyro, accelerometer, and magnetometer uses 16 bits [**42**]. It features a low-dropout linear voltage regulator that allows it to be powered from 2.5 to 5.5V [**42**]. The measurement range of the gyro, accelerometer, and magnetometer can be found in the table below:

Gyro	±245, ±500, or ±2000°/s
Accelerometer	±2, ±4, ±6, ±8, or ±16 g
Hysteresis	±2, ±4, ±8, or ±12 gauss

Table 4. Measurement range of the gyro, accelerometer, and magnetometer [**42**]

Figure 5. ALTIMU-10 - v4 sensor [**42**]

To compute the Euler angles based on the data from the accelerometer, gyroscope, and magnetometer there is the need for sensor fusion that is done by a Kalman Filter.

Kalman Filter is a recursive algorithm that is used for a series of measurements from sensors over time, containing statistical noise and other inaccuracies, and provides estimates of the desired parameters based on the fusion of this data. It is commonly used for GPS. The algorithm has process phases such as: the prediction phase and the estimation phase. In the prediction phase, the Kalman filter constructs a matrix of the uncertainties for each measurement. Once the prediction phase is complete, the Kalman filter yields the estimates of current state variables, continuing to update their uncertainties. When the outcome of the next measurement is available, the estimates are computed by a weighted average, providing higher accuracy [**43**].

4.2. ESP32

The ESP32, figure 6 was the choice for this prototype due to its low cost, energy consumption, interface compatibility, and available libraries for firmware development. In terms of specs, this microcontroller packs a Tensilica Xtensa LX6 dual-core processor in 40nm technology, a Bluetooth and WiFi interface supports up to four 16-MB of external QSPI flash and SRAM and, the crystal oscillator is limited to 40 MHz due to firmware [**14**]. In terms of interfaces, it provides compatibility with several interfaces such as SPI (serial peripheral interface), CAN, and I2C [19]. It is suited for all kinds of environments and has an operating voltage between 2.7 and 3.6V [**14**].

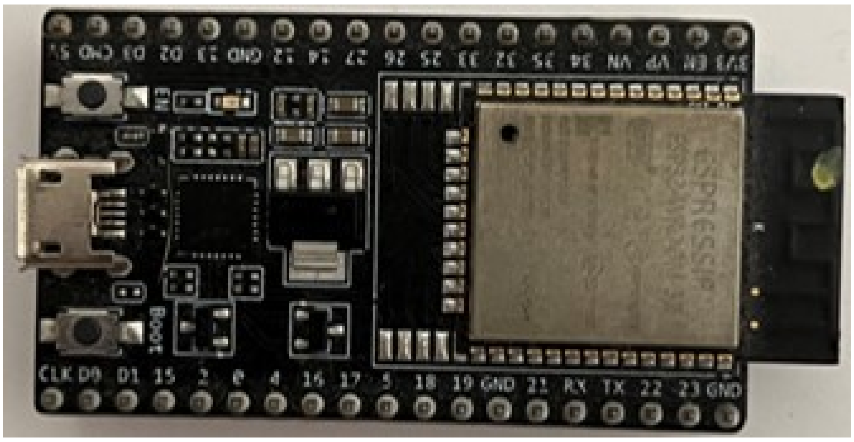


Figure 6. ESP32

4.3. Communication protocols

The goal of this section is to characterize the different wired and wireless communication protocols used to retrieve the data from the sensors using the ESP32. The protocols used are Universal Asynchronous Reception and Transmission (UART) for communication with the computer, Inter-integrated-circuit (I2C) for the IMU AltiIMU 10 v-4 sensor, and Serial Peripheral Interface (SPI) for the RC522 RFID sensor. The voltages associated with force measurement channels are acquired using the ESP32 ADC characterized by 12 bits resolution.

4.3.1. Universal Asynchronous Reception and Transmission (UART)

Universal Asynchronous Reception and Transmission (UART) is a serial communication protocol that allows communication between a host and other devices (eg. a computer) [**44**]. It offers bi-directional, asynchronous, and serial data transmission [**44**]. This protocol has two data lines connected between two devices, one to transmit (TX) and another to receive (RX) [**44**]. UART supports three modes: simplex, which only transfers data in one direction; half-duplex, data transfer data in two directions but not at the same time; full-duplex, where data transmission occurs in both directions simultaneously [44]. To transfer data the device converts the data into serial form and then transmits it to the other device that has to re-convert the data to read it [44]. The main advantages of this protocol are is simple to use and has vast documentation online, and allows parity bit for error checking [44]. On the other hand, the main disadvantages are the limited size frame of 9 bits, and the baud rates of each UART must be within 10% of each other to prevent data loss and a low transmission speed [44].

4.3.2. Inter-integrated-circuit (I2C)

Inter-integrated-circuit or I2C is another serial communications protocol however it is intended to communicate with microcontrollers and similar devices, modules, and sensors while UART is aimed at device-device communications [44]. This protocol requires two wires to exchange information between devices, SCL (serial line clock) for synchronizing the transmission and SDA (serial data line acceptance port) for sending and receiving data [44]. These wires are connected to a shared bus and use an address system thus allowing to have multiple devices connected using the same wires [44]. Collisions are solved by the first device that puts a dominant bit on the bus. The tradeoff is that this protocol can become slower than Serial Peripheral Interface (SPI) [44]. The main advantages are the low pin usage and the flexibility because it allows for multiple master/slave devices [44]. The main disadvantages are the slower speed when compared to SPI and more complexity when the number of devices increases [44].

4.3.3. Serial Peripheral Interface (SPI)

Serial Peripheral Interface (SPI) is another full-duplex serial communication protocol more focused on the microcontrollers [44]. It can be faster in processing than I2C due to its simplicity. Theoretically, there is no device limit (only hardware) and is used when the speed is necessary or when information changes fast [44]. SPI requires 4 lines: master output, slave input (MOSI); master data input, slave data output (MISO); clock signal (SCLK), generated by the master device; and slave-enabled signal (NSS), controlled by the master device, one for each slave device [44]. The main advantages of this protocol are the simplicity, speed, separate MISO, and MOSI lines to receive and transmit data simultaneously [44]. The main disadvantages are the need for four lines (MOSI, MISO, NCLK, NSS) meaning that more pins are occupied. Only supports one master, error checking and acknowledgment have to be implemented by the application [44].

4.4. Prototype

This section presents the final smart crutch prototype and its assembly. As already mentioned, it uses an ESP32, a load cell, a flexiForce, an RFID, and an IMU sensor mounted in a printed circuit board (PCB) attached to a crutch.

4.4.1. Crutch

A crutch shifts the body weight of a patient from the legs to the arms and torso to help patients stand up and walk. The common causes for crutch use are lower extremity injuries such as fractures and/or neurological impairment. It also helps patients with acute and chronic injuries to maintain mobility and independence. Crutch use may not be appropriate for younger and older individuals because the patient needs to have torso strength and coordination to use it successfully. It is also important to mention that the crutch needs to be adjusted to the patient's height to don't provoke any injury [**45**].

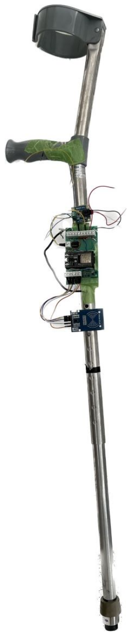


Figure 7. Crutch

4.4.2. Schematics

Figure 8 shows the PCB design for all the components of the smart crutch prototype and the system is already designed to support an additional flexiForce sensor for the hand support of the crutch.

4.4.3. Printed Circuit Board (PCB)

Printed circuit boards or PCBs allow the connection of electric components and reduce the cables needed. These boards can be made from several materials such as FR4 (fiberglass), copper, solder mask, and others. The one used in this prototype uses fiberglass with the components welded to the board [**46**]. Although the PCB design was performed, this prototype used a protoboard, a similar approach to connect all the components. Figure 9 shows the board used and the place for each component.

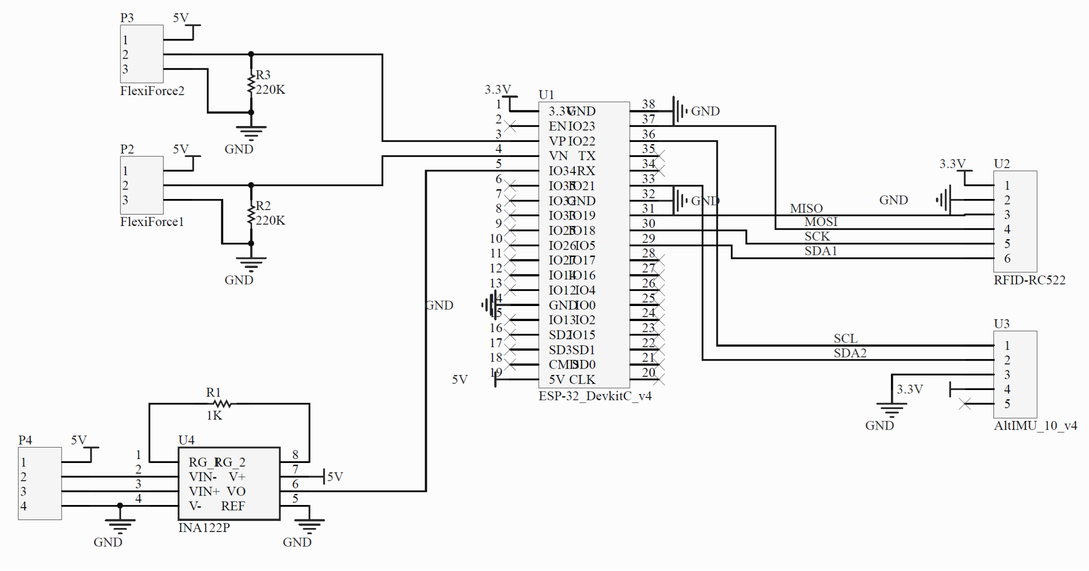


Figure 8. Smart sensor signal conditioning schematics

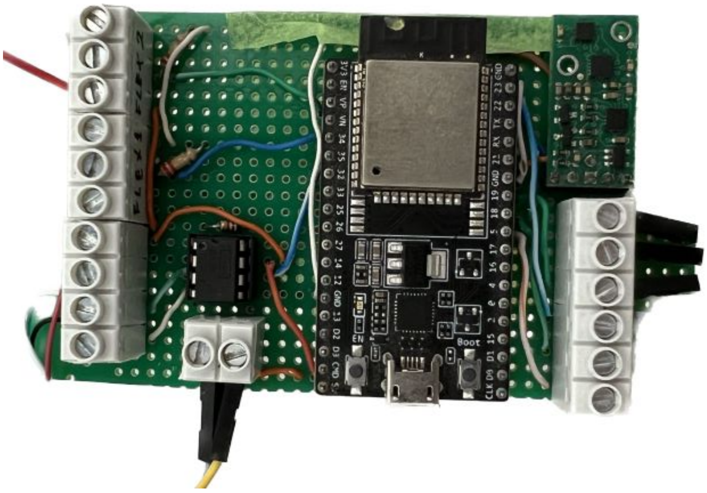


Figure 9. PCB

CHAPTER 5

Software

This section presents the software used in this dissertation for data acquisition, primary processing, and communication. It presents the script used to extract the data from the ESP32, the underlying cloud services used, and the web application. The architecture used is:

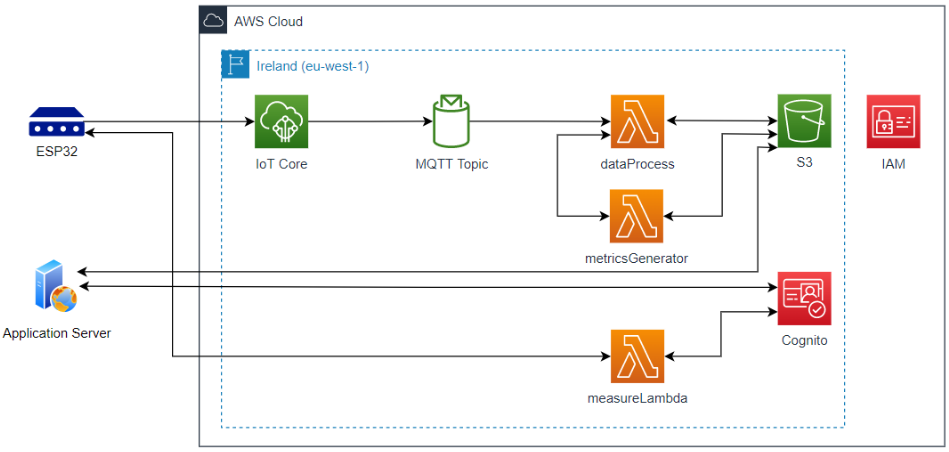


Figure 1. System Software Architecture (S3 refers to the AWS Simple Storage Service, the MQTT topic is the string used by the broker to forward the messages, IAM is the AWS Identity and Access Management service, and AWS Cognito is the service responsible for the user management)

5.1. ESP32 script

The script was developed in Arduino IDE, an open-source tool that allows the development of code using several code libraries and then uploads it to a board. This code is called a sketch and requires two methods:

- Setup(): Runs once either when the board has power or when the reset button is pressed. It initiates variables, pin modes, and libraries [**47**].
- Loop(): runs the code inside continuously and usually has a delay between executions [**47**].

The following libraries are used to develop the software firmware of the system:

- WiFiClientSecure: enables the use of SSL certificates to establish new connections
- PubSubClient: a client for MQTT communication
- ArduinoJson: a JSON library for this board
- WiFi: enables network connection either local or to the internet

- HTTPClient: allows making HTTP requests to a web server
- SPI: enables communication with SPI devices, in this case with the RFID sensor
- MFRC522: library for the MFRC522 sensor that communicates via SPI and allows to read/write an RFID Card or Tag using the ISO/IEC 14443A/MIFARE interface.
- stdlib: used to convert the HTTP response with the measurement delay for the patient from String to Integer
- Wire: enables communication with I2C/TWI devices
- LSM303: library to communicate with the accelerometer on the IMU sensor
- L3G.h: used to retrieve measurements from the gyroscope on the IMU sensor

Additionally are also imported two files, secrets.h and MinIMU9AHRS.h. The first one contains WI-FI credentials, certificates, and a private key to allow the ESP32 to connect to AWS IoT Core. The second includes definitions of constants and variables used to calculate the Euler Angles required by the used library.

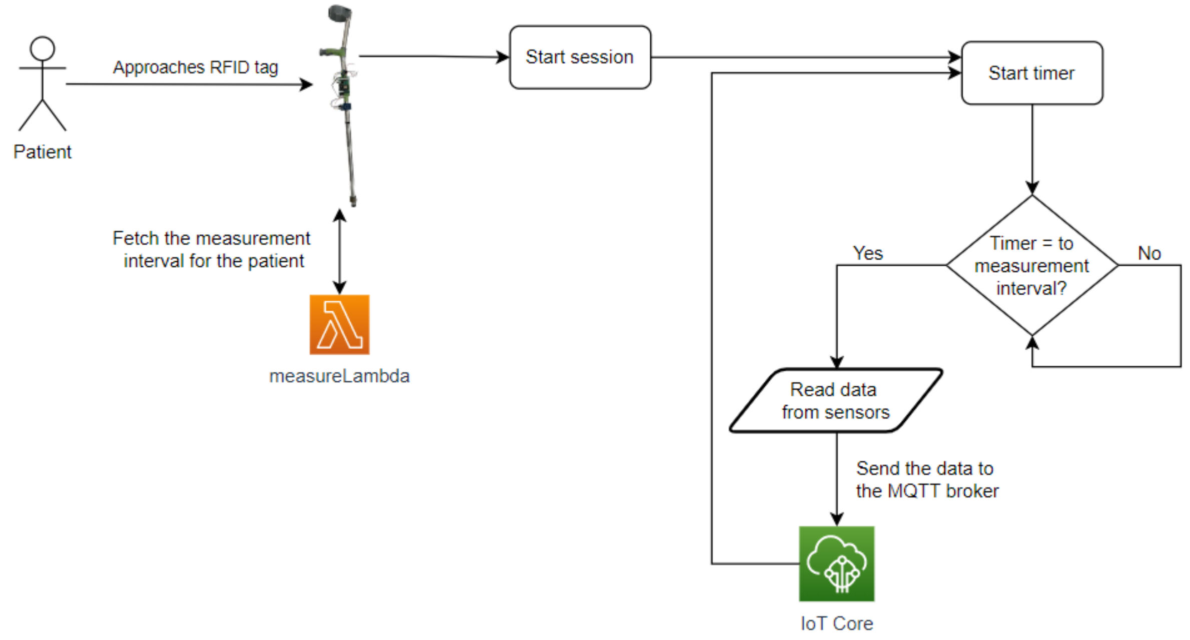
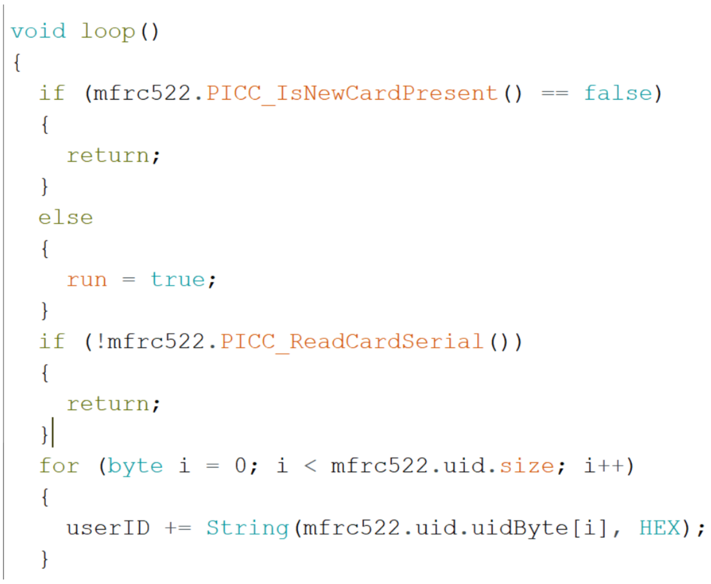

Patient
Approaches RFID tag
Start session
Start timer
Fetch the measurement interval for the patient
measureLambda
Timer = to measurement interval?
Yes
No
Read data from sensors
Send the data to the MQTT broker
IoT Core

FIGURE 2. Data acquisition flow chart

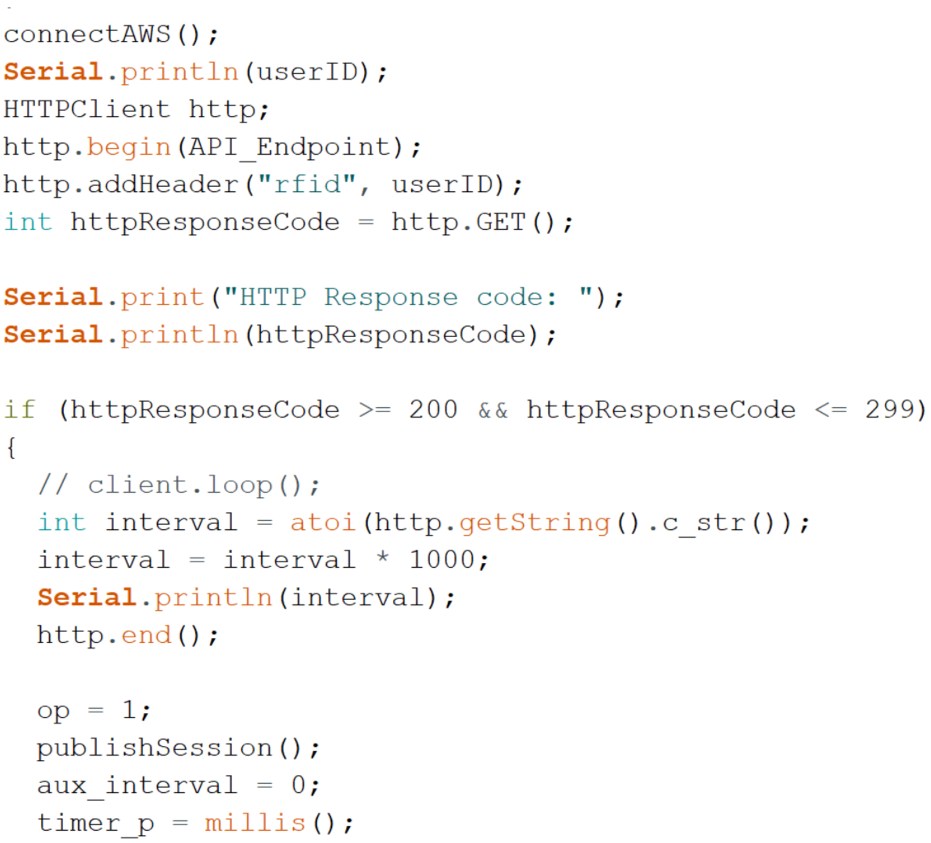
Figure 2 describes on a high level how the data acquisition works. First, a patient needs to approach the RFID tag to the crutch's reader afterward, the crutch requests the measurement interval for that patient from a cloud function (measureLambda). Then it connects to the IoT Core (MQTT broker) and the system sends a JSON packet to mark the session as initiated. The measurements for each sensor, load cell, flexiForce, IMU, and the Euler Angles are also sent in a JSON packet to IoT Core to be processed only when the timer reaches the measurement interval. The timer is then set to zero and starts counting toward the interval value. When the patient decides to finish the session, the patient needs to reapproach the tag to the sensor, and the system sends the last packet that indicates that the session is over and resets the RFID tag value.



```
void loop()
{
  if (mfrc522.PICC_IsNewCardPresent() == false)
  {
    return;
  }
  else
  {
    run = true;
  }
  if (!mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    userID += String(mfrc522.uid.uidByte[i], HEX);
  }
```

FIGURE 3. Loop method - RFID identification

Figure 3 presents the code used to manage the readings of the RFID. When a patient is not using the crutch, this method doesn't do anything since an RFID tag needs to approach to start the system. When a patient uses the tag, the system reads the identifier that is used to start a new session.



```
connectAWS();
Serial.println(userID);
HTTPClient http;
http.begin(API_Endpoint);
http.addHeader("rfid", userID);
int httpResponseCode = http.GET();

Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);

if (httpResponseCode >= 200 && httpResponseCode <= 299)
{
  // client.loop();
  int interval = atoi(http.getString().c_str());
  interval = interval * 1000;
  Serial.println(interval);
  http.end();

  op = 1;
  publishSession();
  aux_interval = 0;
  timer_p = millis();
```

FIGURE 4. Loop method - Connect to IoT Core, fetch the measurement interval and start the session

Figure 4 describes the code used to connect to IoT Core, fetch the measurement interval and start the session. The system starts by calling the method ”connectAWS” which will use the credentials to securely establish a connection to the broker. Afterward, it uses the user’s RFID identifier to make a request to a cloud function and receive the measurement interval. If the request has a positive response code, the system sets the variable interval with the value received. With the information needed, the system starts the session by calling the ”publishSession” method that sends the JSON packet with the user information and starts the timer.

```
if ((millis() - timer) >= 20) // Main loop runs at 50Hz
{
  counter++;
  timer_old = timer;
  timer = millis();
  if (timer > timer_old)
  {
    G_Dt = (timer - timer_old) / 1000.0; // Real time of loop run.
    //We use this on the DCM algorithm (gyro integration time)
    if (G_Dt > 0.2)
      G_Dt = 0; // ignore integration times over 200 ms
  }
  else
    G_Dt = 0;

  // *** DCM algorithm
  // Data adquisition
  Read_Gyro();  // This read gyro data
  Read_Accel(); // Read I2C accelerometer

  if (counter > 5) // Read compass data at 10Hz... (5 loop runs)
  {
    counter = 0;
    Read_Compass();     // Read I2C magnetometer
    Compass_Heading(); // Calculate magnetic heading
  }

  // Calculations...
  Matrix_update();
  Normalize();
  Drift_correction();
  Euler_angles();
```

FIGURE 5. Loop method - Euler angles calculation

Figure 5 shows the application of the Kalman Filter in the code design. The system starts with the acquisition of values for the prediction phase from the accelerometer, gyroscope, and magnetometer and then uses these estimations to provide the most accurate values for the Euler angles.

```
  if (millis() - timer_p >= interval)
  {
    timer_p = millis();
    loadCell = analogRead(34);
    flexiForce1 = analogRead(36);
    flexiForce2 = analogRead(39);
    aux_roll = ToDeg(roll);
    aux_pitch = ToDeg(pitch);
    aux_yaw = ToDeg(yaw);
    compass.read();
    gyro.read();
    snprintf(imu, sizeof(imu),
      "A:%6d,%6d,%6d-M:%6d,%6d,%6d-G:%6d,%6d,%6d",
      compass.a.x, compass.a.y, compass.a.z, compass.m.x,
      compass.m.y, compass.m.z,
      gyro.g.x, gyro.g.y, gyro.g.z);

    publishMessage();
    client.loop();
  }
}
op = 0;
publishSession();
run = false;
Serial.println("End session");
delay(1000);
userID = "";
```

Figure 6. Loop method - Send the data from the sensors and end session

Figure 6 shows the moment the system reads the data from the sensors. When the timer reaches the measurement interval the values from the sensors are assigned to variables. Afterward, the system sends a JSON packet to IoT Core with the measurement information to be processed. In this figure, it is also possible to see what happens when the patient reproaches the RFID card, the system sends a JSON packet that will trigger the end of the session and the data processing.

5.2. Terraform

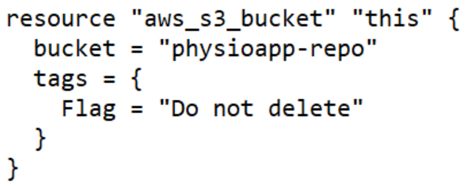
To create the cloud resources for this solution there were two possibilities. The first one was the native one from AWS called CloudFormation, and the other was TerraformIt is simpler to develop the cloud functions because it only needs a file with the code and it can generate and upload the zip automatically when in the case of CloudFormation the user has to generate a zip, then upload it to S3, and only afterward can create the function.

HashiCorp Terraform is an infrastructure as code tool that allows defining resources such as virtual networks or machines on any available platform (cloud or on-prem) in a declarative language called HCL which is very easy to understand. To manage infrastructure, terraform calls the destiny platform's APIs based on the user's code. The main advantage is that these configurations can be reused, versioned, and shared. An example can be if someone deletes a resource, it is possible to relaunch it without wasting time on the setup.

Terraform also offers the possibility to use custom workflows with security, name, or other checks that help to manage the infrastructure. Nevertheless, the core workflow has three steps:

- Write: The user writes the code that defines the resources to create, for instance, virtual storage, machines, etc.
- Plan: Terraform describes the changes generated by the code. These can be the creation, update, or deletion of one or more resources.
- Apply: In case a user approves the changes, Terraform performs the defined operations in the correct order, respecting any resource dependencies.

In this dissertation, Terraform is used to manage all the cloud resources such as user permissions, cloud functions, and other services described in the next section. An example can be found below:



```
resource "aws_s3_bucket" "this" {
  bucket = "physioapp-repo"
  tags = {
    Flag = "Do not delete"
  }
}
```

Figure 7. Creation of an S3 bucket

There was another choice for managing the resources called CloudFormation, an AWS native tool. It has disadvantages, such as a higher overhead when creating cloud functions. This overhead is due to having to generate a zip, then upload it to S3, and only afterward it is possible to create the function. In Terraform the user only has to specify where the file is, and the code will do this process.

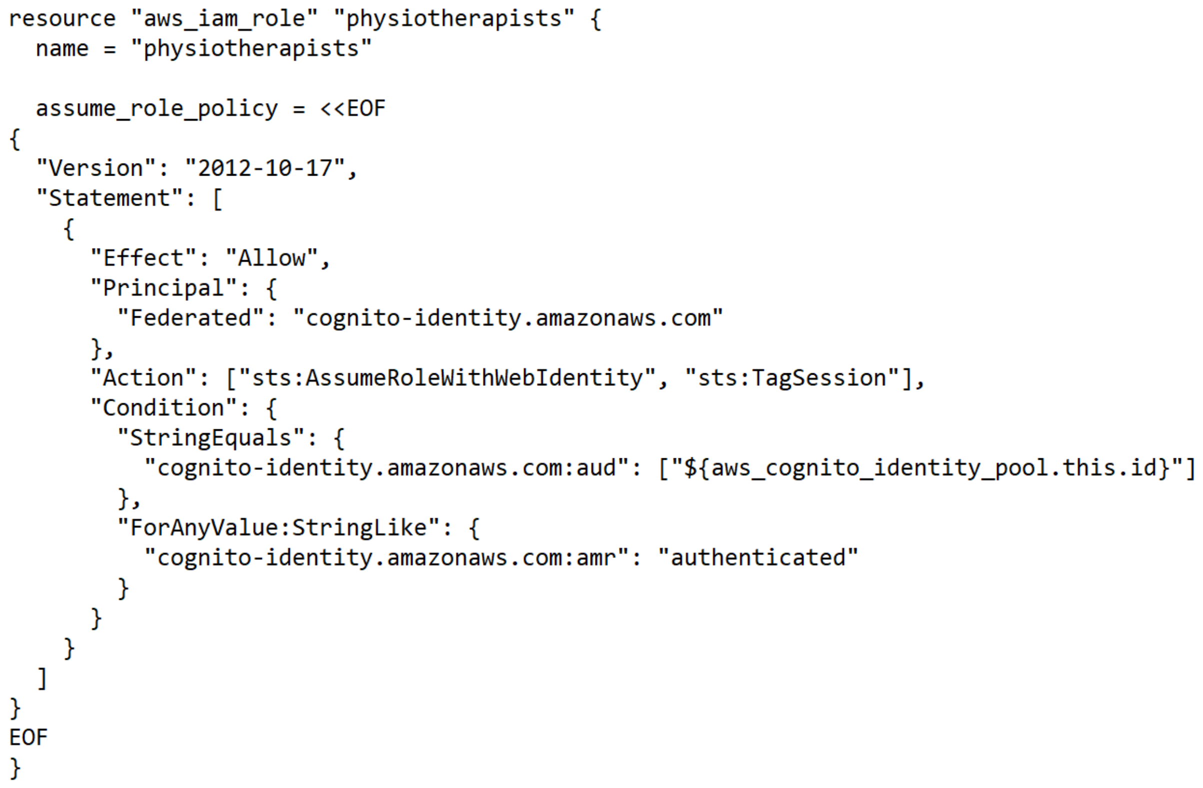
5.3. Cloud services

Cloud services have been revolutionizing the IT world due to the broad capabilities offered, price, and abstraction of the underlying infrastructure like physical servers, networking, and storage. According to Gartner, this market keeps growing at a fast pace being worth 410,915 billion dollars in 2021 with an expected increase to 599,840 billion dollars in 2023 [**48**]. Aside from the price and the other factors listed above, one of the great advantages is that the cloud offers managed services and platform services that enable developers to rapidly deploy their ideas and applications while having less management overhead in terms of infrastructure. With these advantages in mind, it was decided that cloud services would be the most suited solution to host the infrastructure supporting this application. When comparing the top 3 cloud services providers, Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform, AWS was the choice. This provider offers a broader range of services with some having a free tier. It is also the standard in terms of security and reliability being the more mature of the three [**49**]. For the development of this dissertation are used five cloud services, AWS Identity and Access

Management, AWS IoT Core, AWS Lambda, Amazon Simple Storage Service (Amazon S3), and Amazon Cognito.. For the development of this dissertation are used five cloud services, AWS Identity and Access Management, AWS IoT Core, AWS Lambda, Amazon Simple Storage Service (Amazon S3), and Amazon Cognito.

5.3.1. AWS Identity and Access Management (IAM)

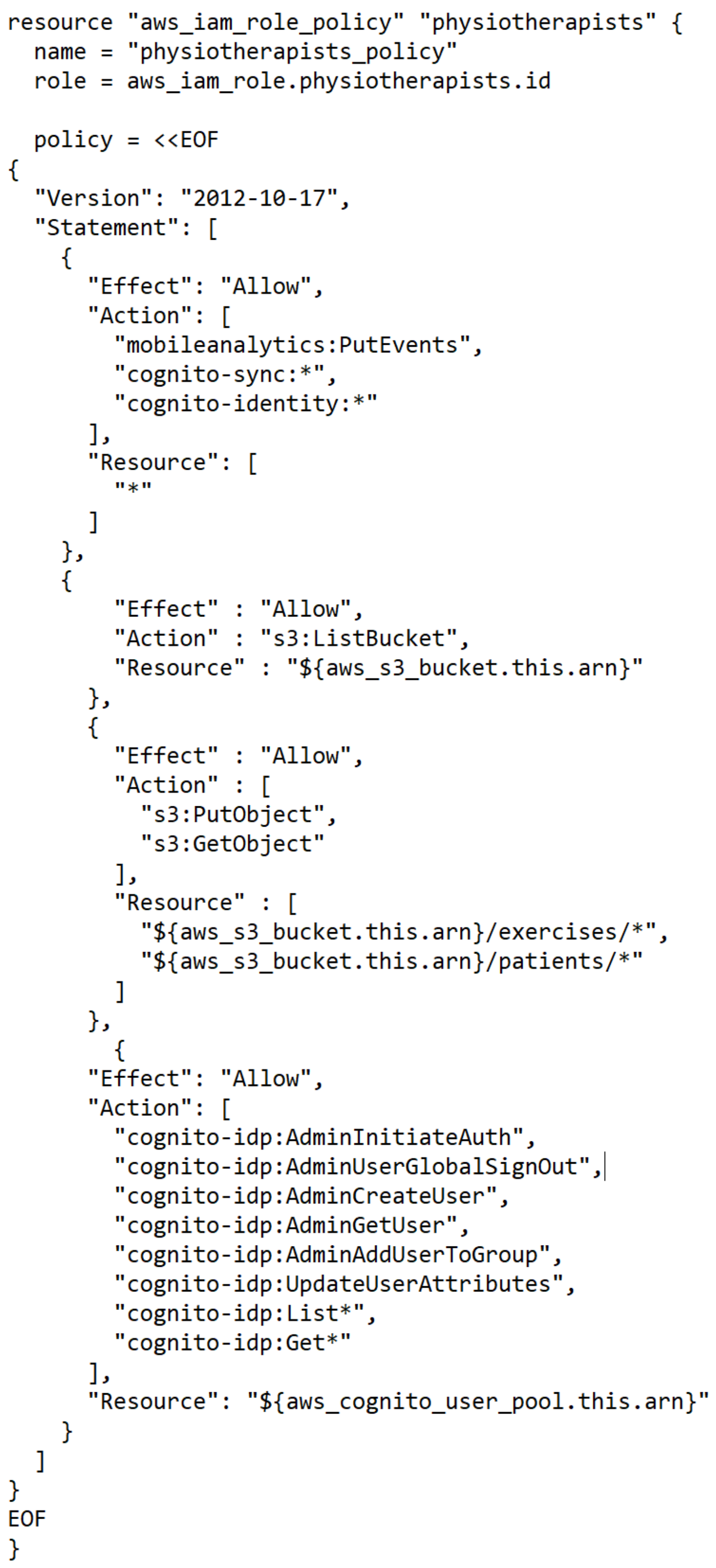
AWS Identity and Access Management is the service responsible for the permission management for AWS services and advocates the least-privilege mindset. It provides a fine-grained level of control because it can reach the resource level and can combine several conditions for a user to access a resource. IAM offers user and permissions management by allowing the creation of users, roles, the leverage of managed policies created by AWS, or the creation of custom ones. These policies are attached to user groups, users, or roles that can be assumed by users, other accounts, or services that gain the permissions defined in the policies attached to those roles. In this application, IAM is used to create custom policies and the roles used by other services in AWS, and for user management, AWS Cognito was the choice.



```
resource "aws_iam_role" "physiotherapists" {
  name = "physiotherapists"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": ["sts:AssumeRoleWithWebIdentity", "sts:TagSession"],
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": ["${aws_cognito_identity_pool.this.id}"]
        },
        "ForAnyValue:StringLike": {
          "cognito-identity.amazonaws.com:amr": "authenticated"
        }
      }
    }
  ]
}
EOF
}
```

FIGURE 8. IAM Role for the physiotherapist

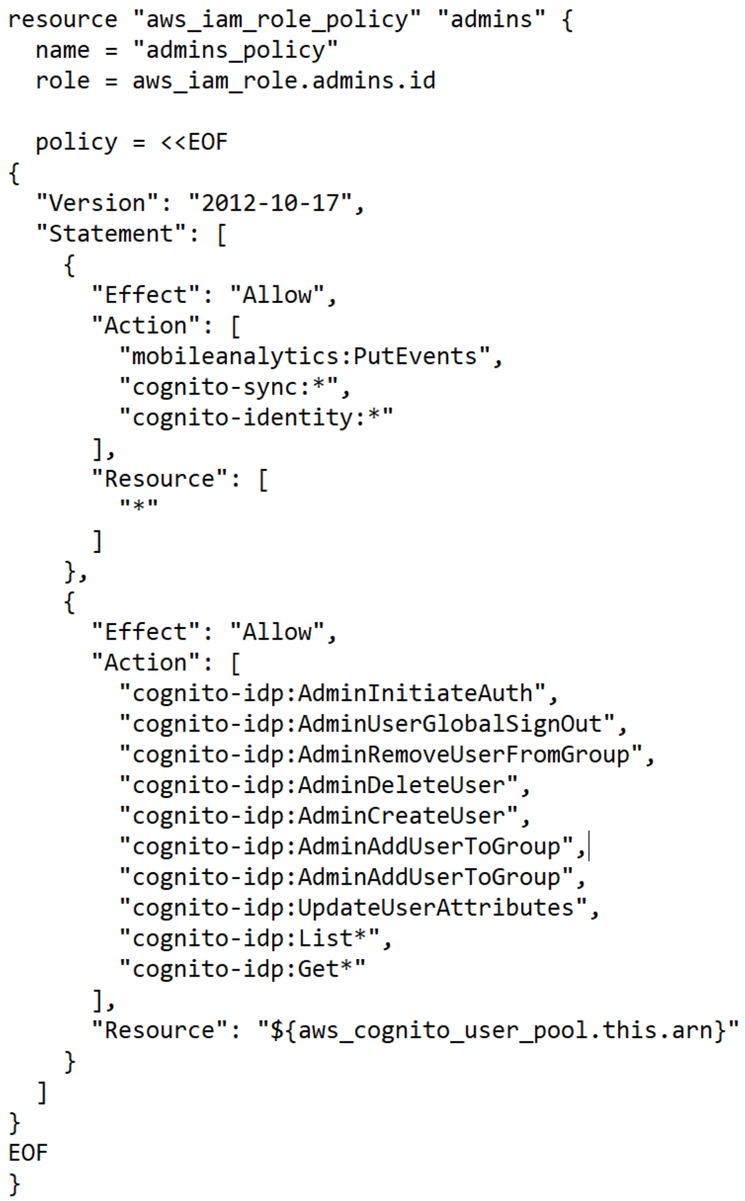
Figure 8 shows the role definition for the physiotherapist. The name of this role is physiotherapists and the assume role policy parameters ensure that only users authenticated with the user management service can assume this role. The assume role policy definition for admins and patients is the same and the mapping between users and roles is defined in AWS Cognito.



```
resource "aws_iam_role_policy" "physiotherapists" {
  name = "physiotherapists_policy"
  role = aws_iam_role.physiotherapists.id

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "mobileanalytics:PutEvents",
        "cognito-sync:*",
        "cognito-identity:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
        "Effect" : "Allow",
        "Action" : "s3:ListBucket",
        "Resource" : "${aws_s3_bucket.this.arn}"
      },
      {
        "Effect" : "Allow",
        "Action" : [
          "s3:PutObject",
          "s3:GetObject"
        ],
        "Resource" : [
          "${aws_s3_bucket.this.arn}/exercises/*",
          "${aws_s3_bucket.this.arn}/patients/*"
        ]
      },
        {
      "Effect": "Allow",
      "Action": [
        "cognito-idp:AdminInitiateAuth",
        "cognito-idp:AdminUserGlobalSignOut",
        "cognito-idp:AdminCreateUser",
        "cognito-idp:AdminGetUser",
        "cognito-idp:AdminAddUserToGroup",
        "cognito-idp:UpdateUserAttributes",
        "cognito-idp:List*",
        "cognito-idp:Get*"
      ],
      "Resource": "${aws_cognito_user_pool.this.arn}"
    }
  ]
}
EOF
}
```

Figure 9. IAM Role policy for the physiotherapists

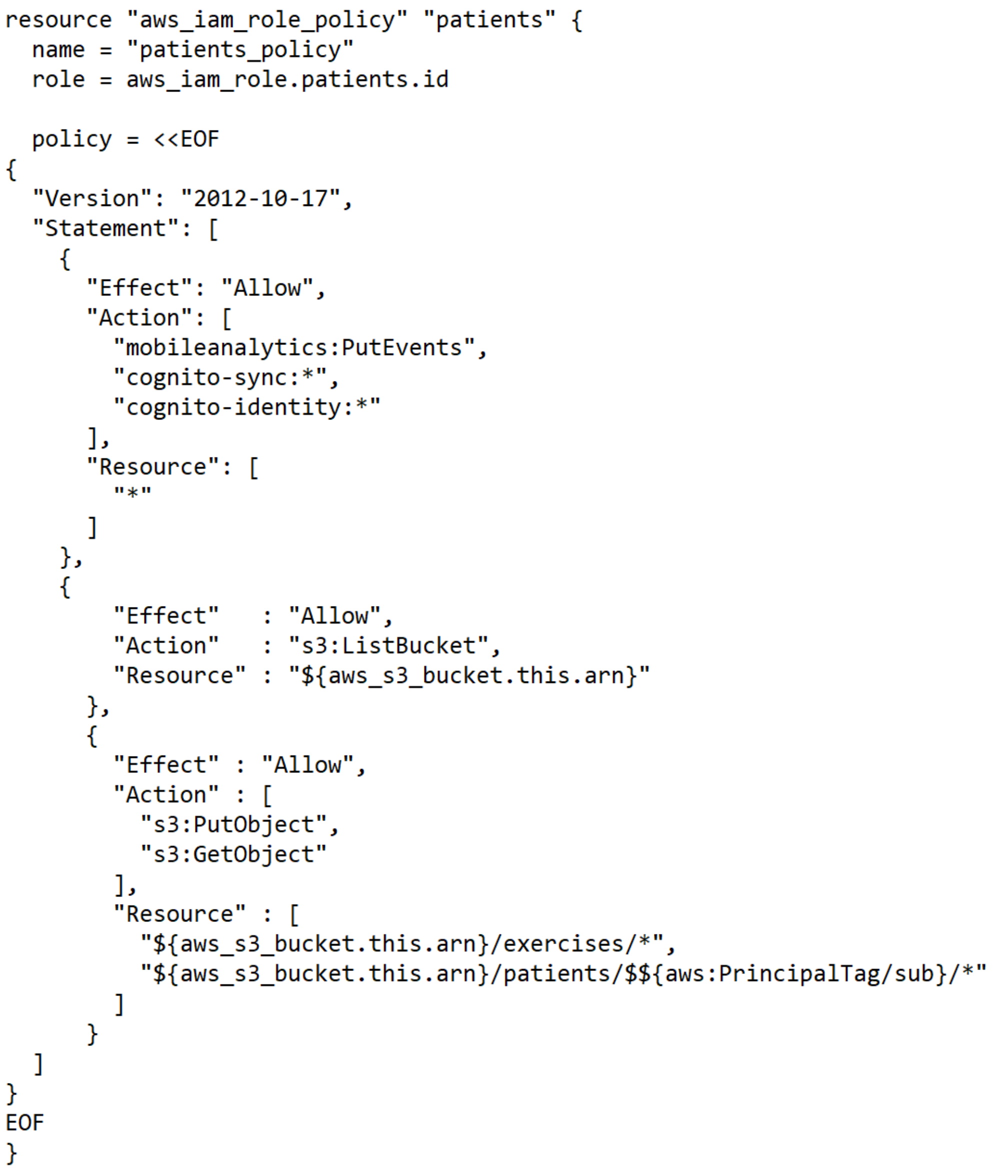
For a role to have access to the resources it needs to have permissions, that are given by an IAM policy. Figure 9 describes the policy assigned to the physiotherapists' role, it gives access to the S3 bucket where the data of the sessions is stored so that the physiotherapists can view and create files (when creating exercise plans) and allows view and create patients. This policy also gives physiotherapist the possibility to update their personal data.



```
resource "aws_iam_role_policy" "admins" {
  name = "admins_policy"
  role = aws_iam_role.admins.id

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "mobileanalytics:PutEvents",
        "cognito-sync:*",
        "cognito-identity:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cognito-idp:AdminInitiateAuth",
        "cognito-idp:AdminUserGlobalSignOut",
        "cognito-idp:AdminRemoveUserFromGroup",
        "cognito-idp:AdminDeleteUser",
        "cognito-idp:AdminCreateUser",
        "cognito-idp:AdminAddUserToGroup",
        "cognito-idp:AdminAddUserToGroup",
        "cognito-idp:UpdateUserAttributes",
        "cognito-idp:List*",
        "cognito-idp:Get*"
      ],
      "Resource": "${aws_cognito_user_pool.this.arn}"
    }
  ]
}
EOF
}
```

FIGURE 10. IAM Role policy for the admins

Figure 10 shows the policy given to the admin role. Compared with the physiotherapists, admins can also manage their information and have the option to delete users, however, don't have access to the S3 bucket where the session data is stored.

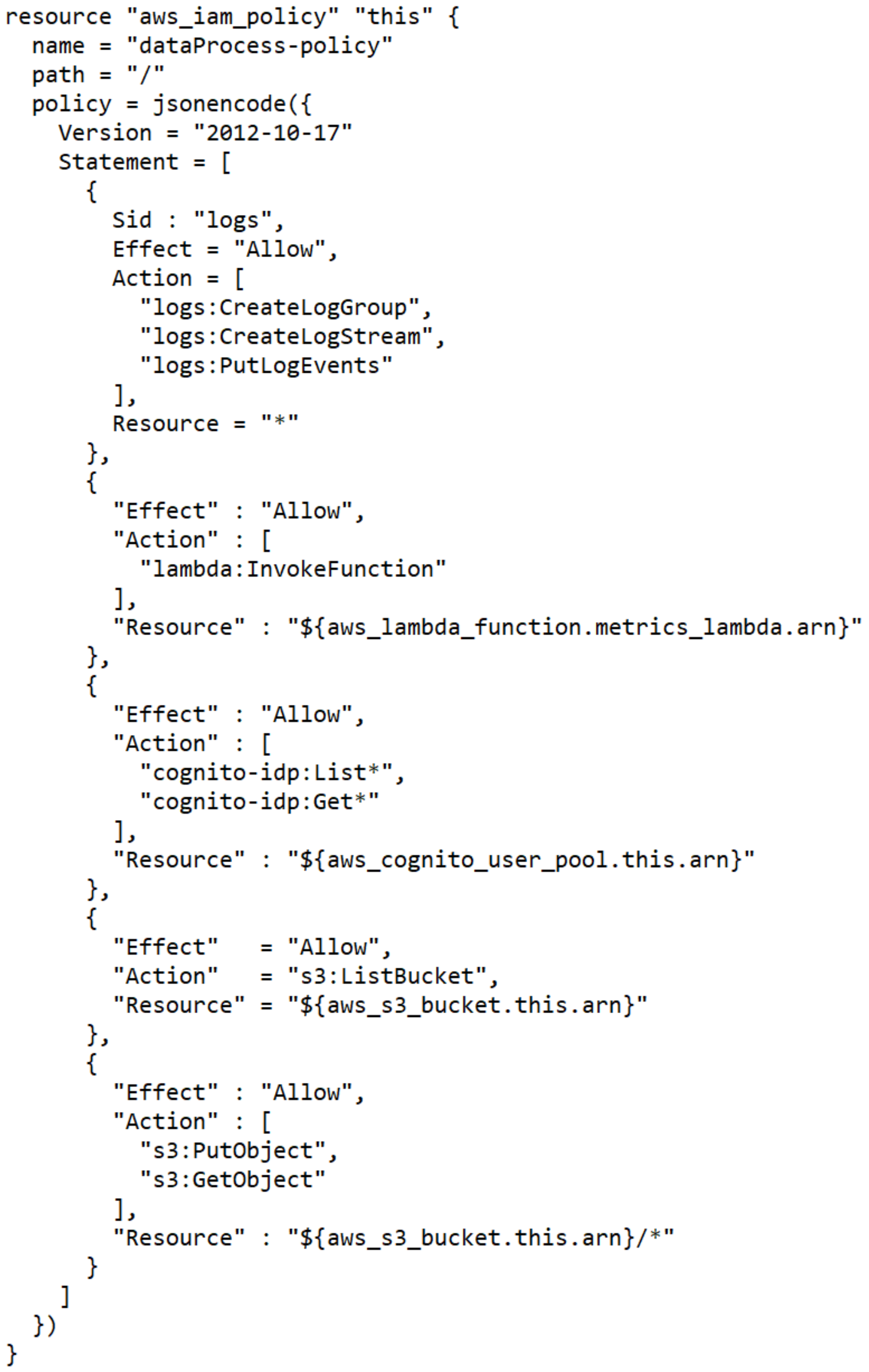


```
resource "aws_iam_role_policy" "patients" {
  name = "patients_policy"
  role = aws_iam_role.patients.id

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "mobileanalytics:PutEvents",
        "cognito-sync:*",
        "cognito-identity:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
        "Effect"   : "Allow",
        "Action"   : "s3:ListBucket",
        "Resource" : "${aws_s3_bucket.this.arn}"
      },
      {
        "Effect" : "Allow",
        "Action" : [
          "s3:PutObject",
          "s3:GetObject"
        ],
        "Resource" : [
          "${aws_s3_bucket.this.arn}/exercises/*",
          "${aws_s3_bucket.this.arn}/patients/$${aws:PrincipalTag/sub}/*"
        ]
      }
  ]
}
EOF
}
```

Figure 11. IAM Role policy for the patients

Figure 11 presents the policy of the patient role. Compared with the admins and physiotherapists, the patients can't view and manage users, they can only manage their information. To view the session data and exercise plans they also need to have access to the S3 bucket however, they can only view their own data.

The cloud functions in AWS Lambda also need to have IAM roles and policies to be able to access the services needed. The main difference between the roles of the users and lambda is the principal which defines who can assume the role. In this case, the principal has to be the lambda service "lambda.amazonaws.com", and there is no condition to verify if the user is authenticated or not. In terms of IAM policies the main differences is that the lambdas have also permissions to create execution logs to check for errors and can't create new users.



```
resource "aws_iam_policy" "this" {
  name = "dataProcess-policy"
  path = "/"
  policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Sid : "logs",
        Effect = "Allow",
        Action = [
          "logs:CreateLogGroup",
          "logs:CreateLogStream",
          "logs:PutLogEvents"
        ],
        Resource = "*"
      },
      {
        "Effect" : "Allow",
        "Action" : [
          "lambda:InvokeFunction"
        ],
        "Resource" : "${aws_lambda_function.metrics_lambda.arn}"
      },
      {
        "Effect" : "Allow",
        "Action" : [
          "cognito-idp:List*",
          "cognito-idp:Get*"
        ],
        "Resource" : "${aws_cognito_user_pool.this.arn}"
      },
      {
        "Effect"   = "Allow",
        "Action"   = "s3:ListBucket",
        "Resource" = "${aws_s3_bucket.this.arn}"
      },
      {
        "Effect" : "Allow",
        "Action" : [
          "s3:PutObject",
          "s3:GetObject"
        ],
        "Resource" : "${aws_s3_bucket.this.arn}/*"
      }
    ]
  })
}
```

Figure 12. IAM Role policy for the dataProcess lambda

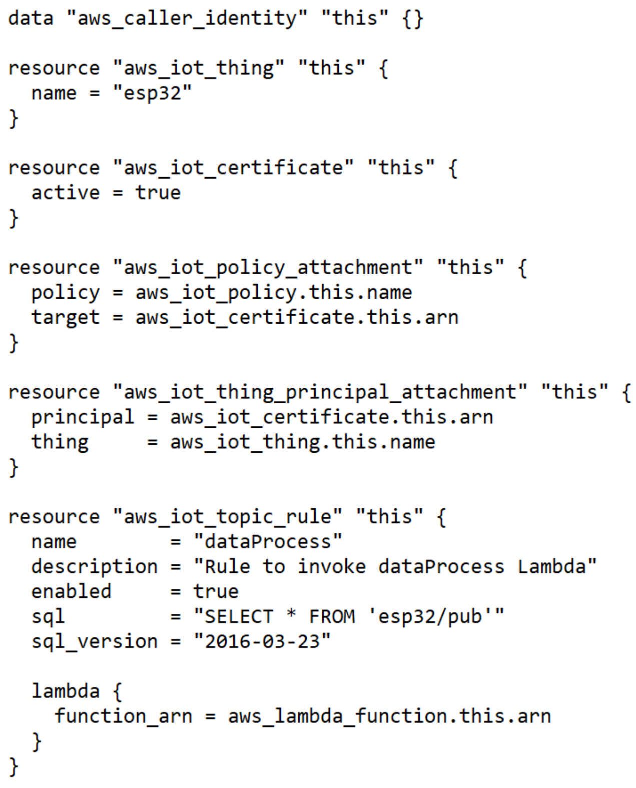
Figure 12 shows the policy assigned to the dataProcess role as an example. This policy allows the lambda to generate execution logs, invoke the metricsGenerator lambda, view the users in Cognito, and get and put files on the S3 bucket.

5.3.2. AWS IoT Core

AWS IoT Core provides connectivity, scalability, and management for IoT devices quickly and reliably without physical servers. It supports MQTT, HTTPS, MQTT over WSS, and

LoRaWAN as communication protocols. It integrates with many other services using rules to filter, transform, and perform actions with the data received. The main functionality leveraged in this work is the MQTT Broker, responsible for obtaining the messages from the ESP32, and a custom rule that forwards the data to AWS Lambda. Security-wise, this service guarantees encryption on all points of the connection, and to exchange data there is always the need to have a proven identity. AWS IoT supports the AWS method of authentication (called 'SigV4'), X.509 certificate-based authentication, and customer-created token-based authentication (through custom authorizers). For this dissertation, MQTT connectivity is secured with certificates mapped with policies to authorize devices or applications to access this service.

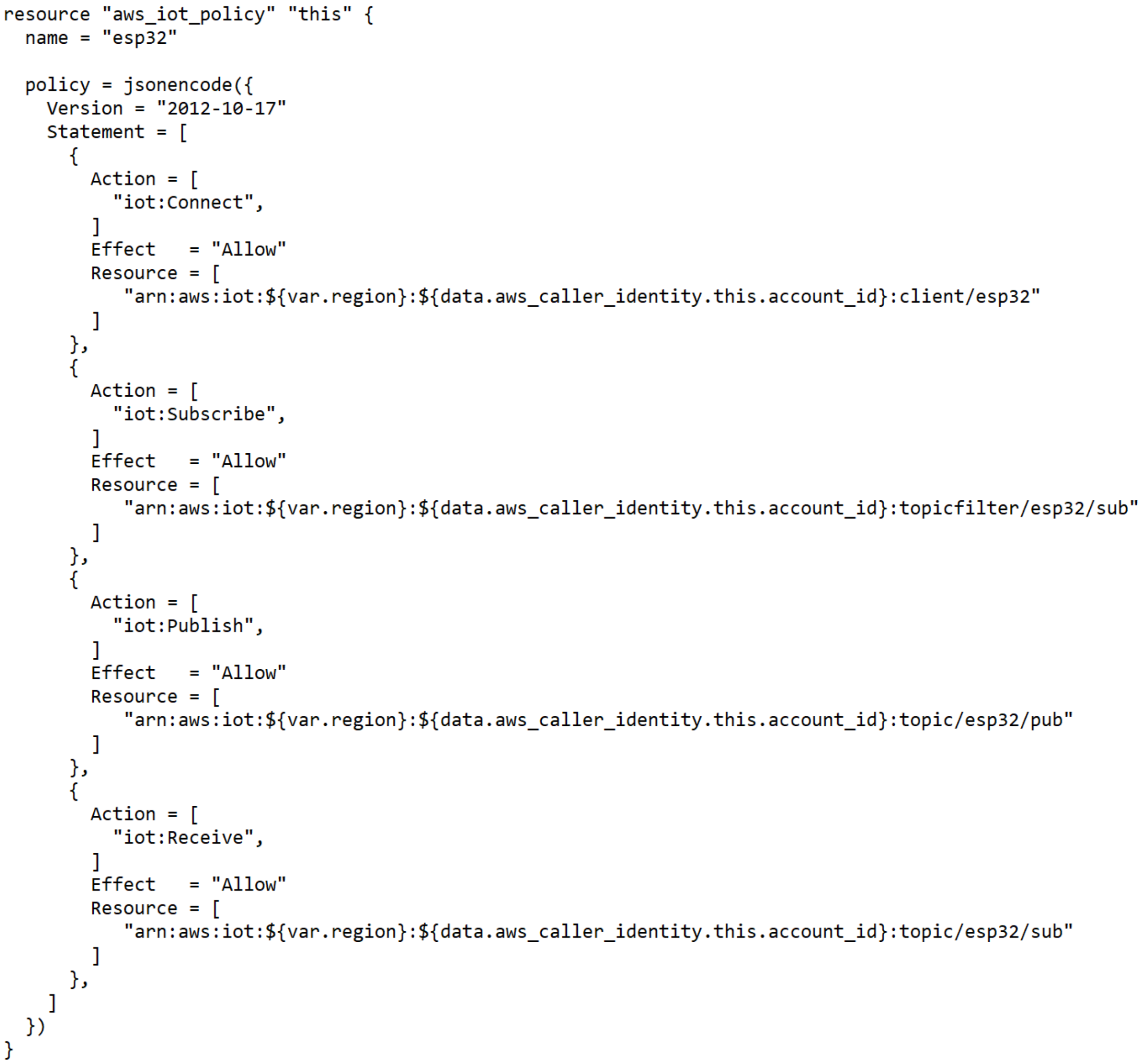
To securely send the data to IoT Core via MQTT, are used three resources, a thing, a certificate, and a policy. A thing is a logical or physical device identified by its name that can be categorized into types and have custom attributes. This name is also used as the default MQTT client id. A certificate authenticates the client, and the policy provides the defined permissions to a client. A custom rule is also employed to forward the content from the MQTT topic to AWS lambda.



```
data "aws_caller_identity" "this" {}

resource "aws_iot_thing" "this" {
  name = "esp32"
}

resource "aws_iot_certificate" "this" {
  active = true
}

resource "aws_iot_policy_attachment" "this" {
  policy = aws_iot_policy.this.name
  target = aws_iot_certificate.this.arn
}

resource "aws_iot_thing_principal_attachment" "this" {
  principal = aws_iot_certificate.this.arn
  thing     = aws_iot_thing.this.name
}

resource "aws_iot_topic_rule" "this" {
  name        = "dataProcess"
  description = "Rule to invoke dataProcess Lambda"
  enabled     = true
  sql         = "SELECT * FROM 'esp32/pub'"
  sql_version = "2016-03-23"

  lambda {
    function_arn = aws_lambda_function.this.arn
  }
}
```

FIGURE 13. IoT Core - Terraform code to create the thing, the certificate and to create the rule for the esp32/pub topic

Figure 13 presents the code used to create the thing, the certificate, and the assignment of the policy created earlier to the certificate. Additionally, it also shows the rule created to forward all the traffic from the esp32/pub topic to the dataProcess lambda.



```
resource "aws_iot_policy" "this" {
  name = "esp32"

  policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Action = [
          "iot:Connect",
        ]
        Effect   = "Allow"
        Resource = [
           "arn:aws:iot:${var.region}:${data.aws_caller_identity.this.account_id}:client/esp32"
        ]
      },
      {
        Action = [
          "iot:Subscribe",
        ]
        Effect   = "Allow"
        Resource = [
           "arn:aws:iot:${var.region}:${data.aws_caller_identity.this.account_id}:topicfilter/esp32/sub"
        ]
      },
      {
        Action = [
          "iot:Publish",
        ]
        Effect   = "Allow"
        Resource = [
           "arn:aws:iot:${var.region}:${data.aws_caller_identity.this.account_id}:topic/esp32/pub"
        ]
      },
      {
        Action = [
          "iot:Receive",
        ]
        Effect   = "Allow"
        Resource = [
           "arn:aws:iot:${var.region}:${data.aws_caller_identity.this.account_id}:topic/esp32/sub"
        ]
      },
    ]
  })
}
```

FIGURE 14. IoT Core - Terraform code to create the IoT Core policy

Figure 14 details the policy created for the certificate. With this policy, the device can connect to the IoT Core service subscribe and receive data in the topic ”esp32/sub” and publish to the ”esp32/pub” topic which is the one used by the esp32 to send the data.

5.3.3. AWS Lambda

AWS Lambda is a cloud computing service that runs code responding to an event. There are several types of events such as new files added to a bucket, changes in a database, HTTP requests, or other services. Lambda supports several programming languages like python, java, go, and others. In this service, users only pay for the time that the functions run and have one million execution free per month and only need to create the code, the rest of the infrastructure is managed by AWS. This prototype uses three lambdas, dataProcess, metricsGenerator, and measureLambda which are described in this subsection.

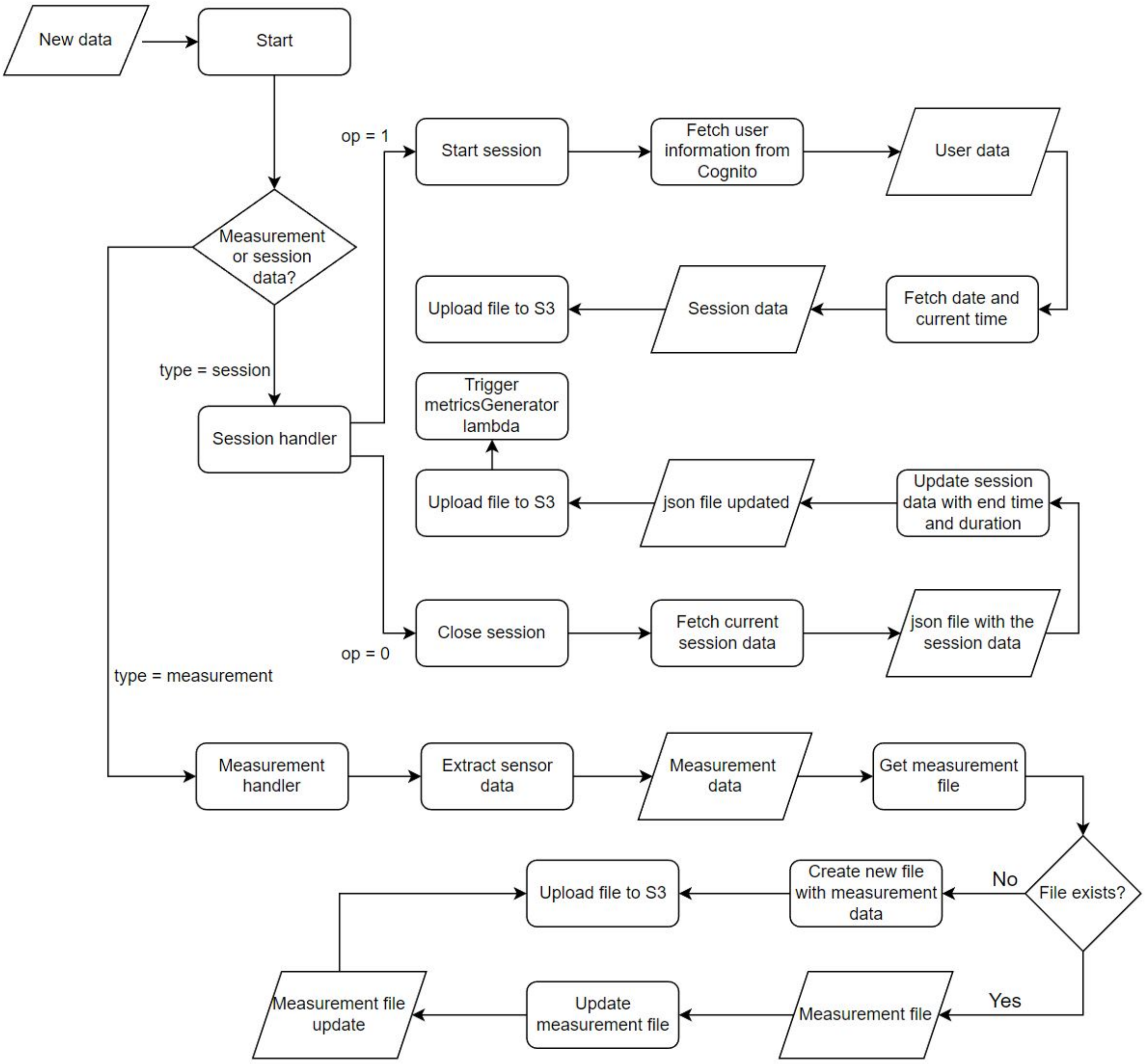


FIGURE 15. Data process lambda flowchart

The dataProcess function is responsible for processing the data sent by the ESP32. The first method that runs is the main, which checks the type value from the message, if it is a session it executes the session handler method, otherwise is the measurementHandler. The first one either creates the record of a new session or ends a session. When starting a new session (op=1), the function generates the current timestamp, fetches the user information from Cognito with the fetchUserData method, and saves this file in the patient's directory on the S3 bucket. If it needs to close the session(op=0), it searches for the last session record, updates the end time and duration attributes, re-uploads the file to S3, and triggers the metricsGenerator lambda. When the contents of a new message are a new measurement, the measurementHandler method extracts the sensor readings, and for the data from the IMU, it splits them into accelerometer, magnetometer, gyro, and angles (pitch, roll, and yaw) for better processing. Afterward, it searches for the current session and checks if the measurements file exists, if so, it updates the file with the new data otherwise, it creates the file.

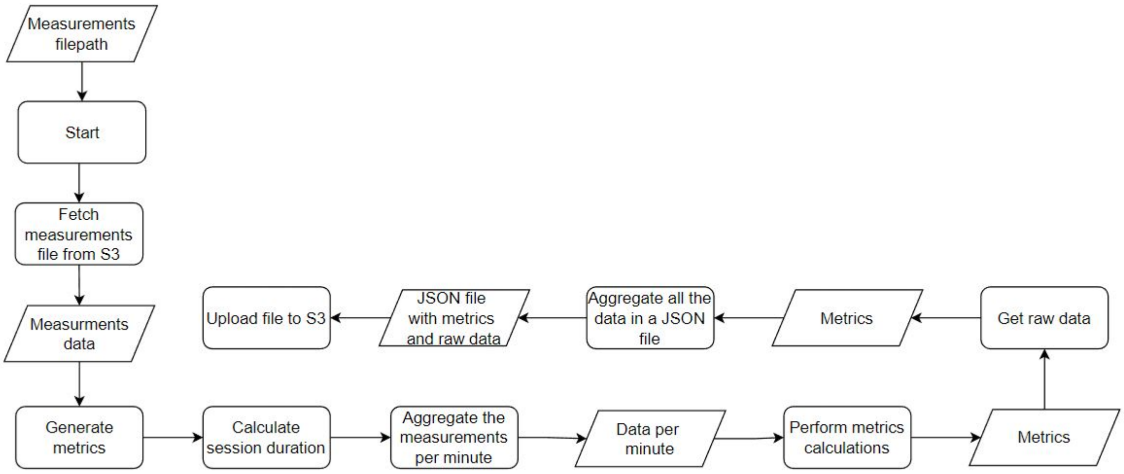

Measurements filepath
Start
Fetch measurements file from S3
Measurments data
Generate metrics
Calculate session duration
Aggregate the measurements per minute
Data per minute
Perform metrics calculations
Metrics
Get raw data
Metrics
Aggregate all the data in a JSON file
JSON file with metrics and raw data
Upload file to S3

Figure 16. Metrics generator lambda flowchart

This lambda generates all the metrics that are to be shown in the web application. It starts by calculating the duration of the session in minutes and then aggregates all the measurements by minute. Afterward, it calculates the maximum, minimum, average, variance, and standard deviation for the load cell and flexiForce and the average and variation for the Euler angles. Additionally, it also provides the data of the acceleration (in x,y, and z) during the sessions as well as the evolution of the Euler Angles. The data is then stored in a JSON file on the bucket.

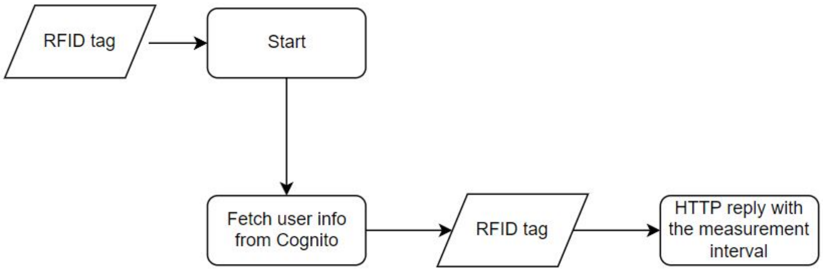
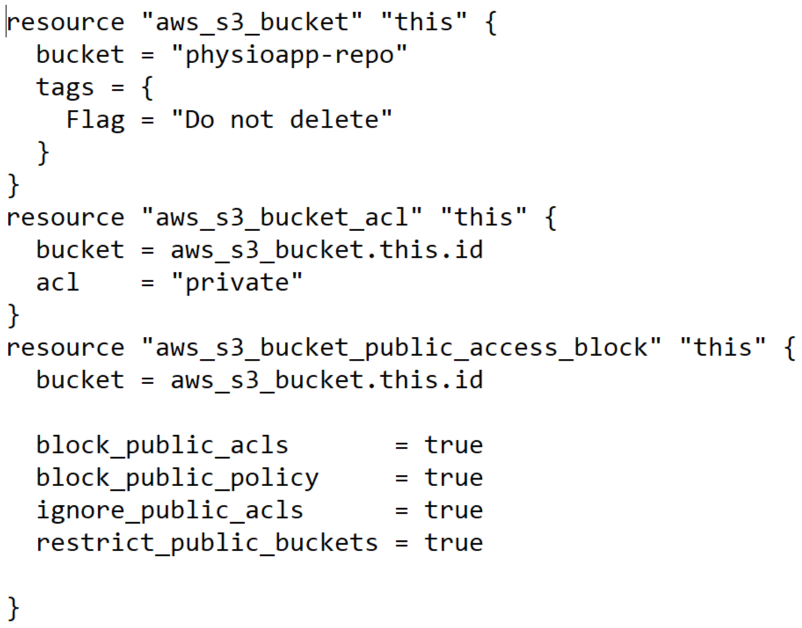

RFID tag
Start
Fetch user info from Cognito
RFID tag
HTTP reply with the measurement interval

Figure 17. Measure lambda flowchart

Compared with previous lambdas, this one is less complex. It is triggered by an HTTP request made by the ESP32 asking for the measurement interval for the patient. The function starts by finding the user that has the RFID from the request with the getUserInfo method, and after finding it, it creates an HTTP response with the measurement interval.

5.3.4. AWS Simple Storage Service (S3)

AWS Simple Storage Service or S3 is a cloud object storage service that provides high availability and redundancy storage at a competitive price. Users store the files in buckets up to 5 terabytes in size that are replicated across three data centers in the same region (geographic region ex. Frankfurt). It is also possible to copy the data automatically to another region for maximum redundancy. Another feature of this service is that it offers the possibility of hosting static websites without the need to deploy a server. In terms of security, bucket access can be either public or private depending on the use case, and the permissions are controlled at the bucket level and/or IAM.



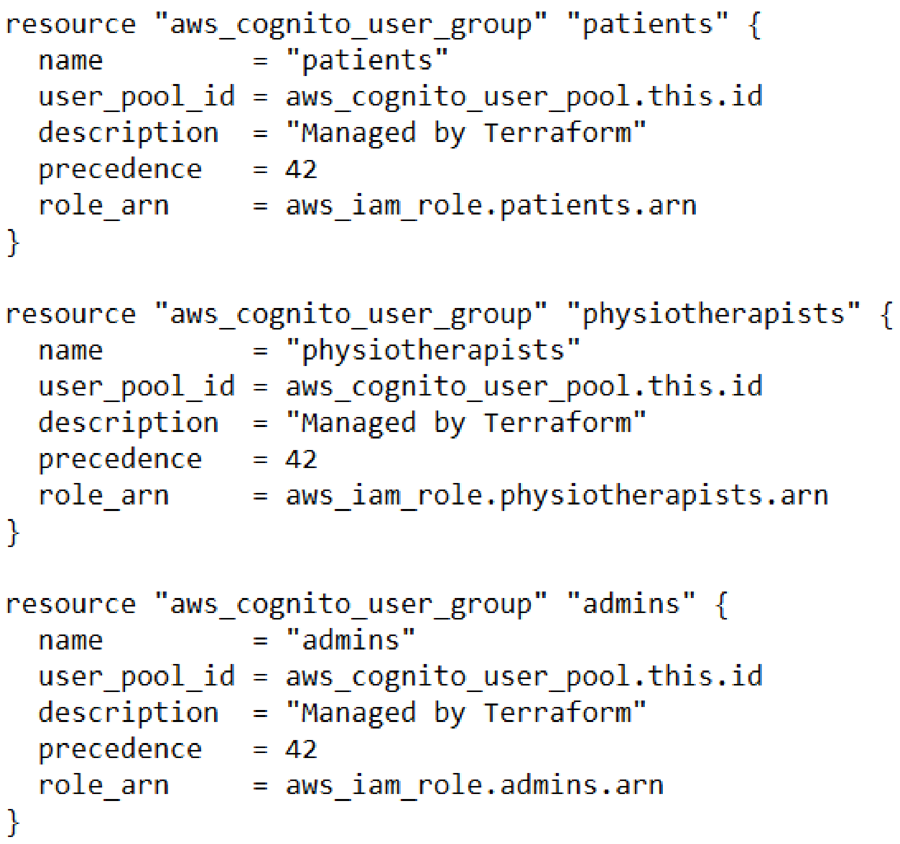
```
resource "aws_s3_bucket" "this" {
  bucket = "physioapp-repo"
  tags = {
    Flag = "Do not delete"
  }
}
resource "aws_s3_bucket_acl" "this" {
  bucket = aws_s3_bucket.this.id
  acl    = "private"
}
resource "aws_s3_bucket_public_access_block" "this" {
  bucket = aws_s3_bucket.this.id

  block_public_acls       = true
  block_public_policy     = true
  ignore_public_acls      = true
  restrict_public_buckets = true

}
```

Figure 18. S3 - Terraform code to create the bucket

Figure 18 presents the code that creates the S3 bucket. The bucket access control list is set to private to guarantee that only the account resources can access the bucket in addition to the block public access resource.

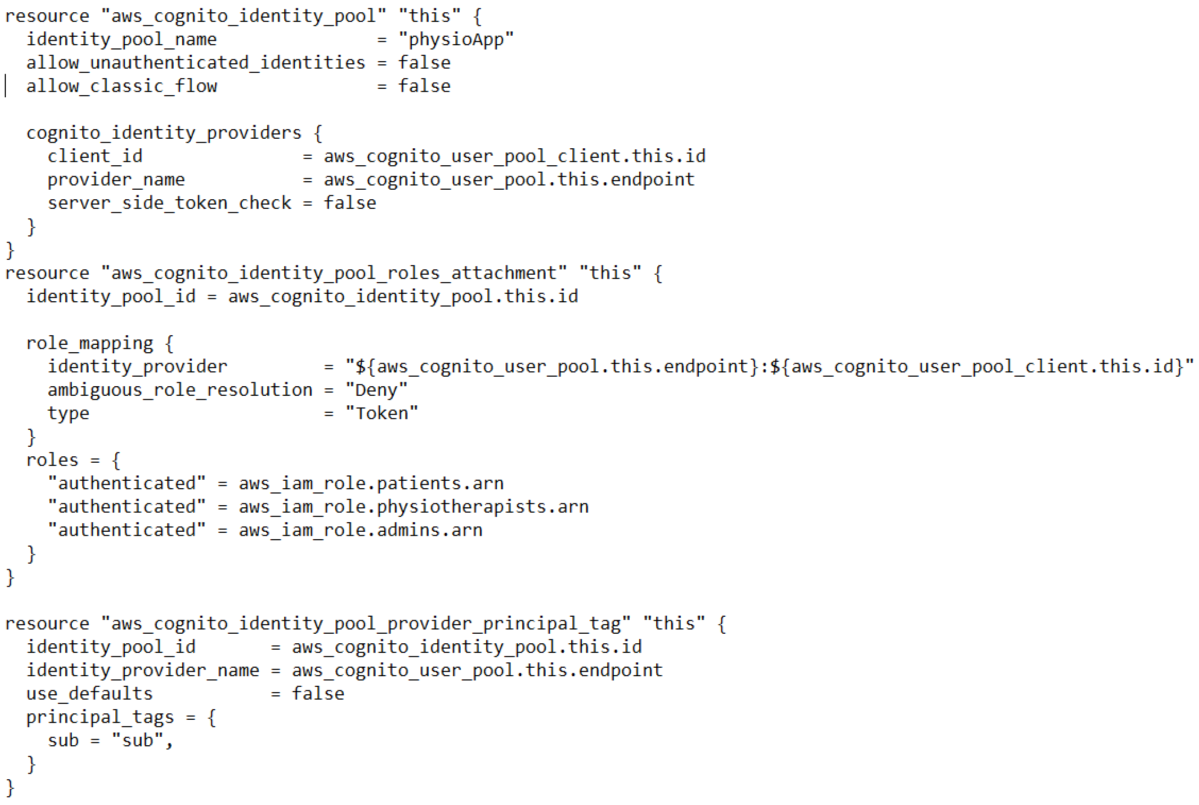
5.3.5. AWS Cognito

AWS Cognito is a cloud-native user authentication and management service that can be integrated into custom applications in a simple way. Since this application uses AWS resources, mainly S3 buckets, this was the only service that offered user management and permissions mapping to access the resources. This service allows users to self-register easily by offering a user interface and APIs for custom register pages and the same for the login. It supports integration with external identity providers, such as Facebook, Google, SAML 2.0, and OpenID Connect. Cognito is split into two services, user pools, and federation identities. The first one stores the information about each user, allows the creation of user groups, the users' sign-up, and sign-in experience, and provides the information to integrate with the custom application. On the other hand, federation identities based on defined criteria provide a mapping between the user and its IAM permissions so that users can access AWS resources.



```
resource "aws_cognito_user_group" "patients" {
  name         = "patients"
  user_pool_id = aws_cognito_user_pool.this.id
  description  = "Managed by Terraform"
  precedence   = 42
  role_arn     = aws_iam_role.patients.arn
}

resource "aws_cognito_user_group" "physiotherapists" {
  name         = "physiotherapists"
  user_pool_id = aws_cognito_user_pool.this.id
  description  = "Managed by Terraform"
  precedence   = 42
  role_arn     = aws_iam_role.physiotherapists.arn
}

resource "aws_cognito_user_group" "admins" {
  name         = "admins"
  user_pool_id = aws_cognito_user_pool.this.id
  description  = "Managed by Terraform"
  precedence   = 42
  role_arn     = aws_iam_role.admins.arn
}
```

Figure 19. Cognito - User groups definition



```
resource "aws_cognito_identity_pool" "this" {
  identity_pool_name               = "physioApp"
  allow_unauthenticated_identities = false
  allow_classic_flow               = false

  cognito_identity_providers {
    client_id               = aws_cognito_user_pool_client.this.id
    provider_name           = aws_cognito_user_pool.this.endpoint
    server_side_token_check = false
  }
}
resource "aws_cognito_identity_pool_roles_attachment" "this" {
  identity_pool_id = aws_cognito_identity_pool.this.id

  role_mapping {
    identity_provider         = "${aws_cognito_user_pool.this.endpoint}:${aws_cognito_user_pool_client.this.id}"
    ambiguous_role_resolution = "Deny"
    type                      = "Token"
  }
  roles = {
    "authenticated" = aws_iam_role.patients.arn
    "authenticated" = aws_iam_role.physiotherapists.arn
    "authenticated" = aws_iam_role.admins.arn
  }
}

resource "aws_cognito_identity_pool_provider_principal_tag" "this" {
  identity_pool_id       = aws_cognito_identity_pool.this.id
  identity_provider_name = aws_cognito_user_pool.this.endpoint
  use_defaults           = false
  principal_tags = {
    sub = "sub",
  }
}
```

Figure 20. Cognito - Identity pool definition

The user pool to manage the users is called physioApp and users can log in with their email and password. The users' attributes are also specified, each user needs to have a name, an email, a phone number, a birthdate, a gender, an address, and a nif. The RFID, measurement interval, and device are only for patients. The figure shows three groups, admins, physiotherapists, and patients which are mapped with the defined roles in IAM. The Cognito identity pool is also specified, this service is responsible for assigning the role to each user by verifying the users' group when this logs in. The last resource, the principal tag resource, allows using the patients' user id to restrict access to each patient's data in the S3 bucket.

5.4. Application server

As described in the previous section, the data processing is made using serverless functions in AWS lambda that save the information to show in the web application in JSON files on the S3 bucket. The application is a front-end to display the data to the users in a presentable way.

This application uses flask, a python framework that is easy to use, lightweight, and provides several out-of-box functionalities such as a built-in web server and debugger that helps to speed up experimentation and error fixing. It also uses HTML, CSS, and JavaScript to design pages visible to the user. Additionally, the graphs used to display session data are created by chart JS, one of the most popular free JavaScript libraries for this functionality.

As mentioned in the system description, there are three types of users. Patients can view session data and the exercise plans created. Physiotherapists can create new patients, view session data, and create/view patients' exercise plans. Lastly, the admins can create new physiotherapists.

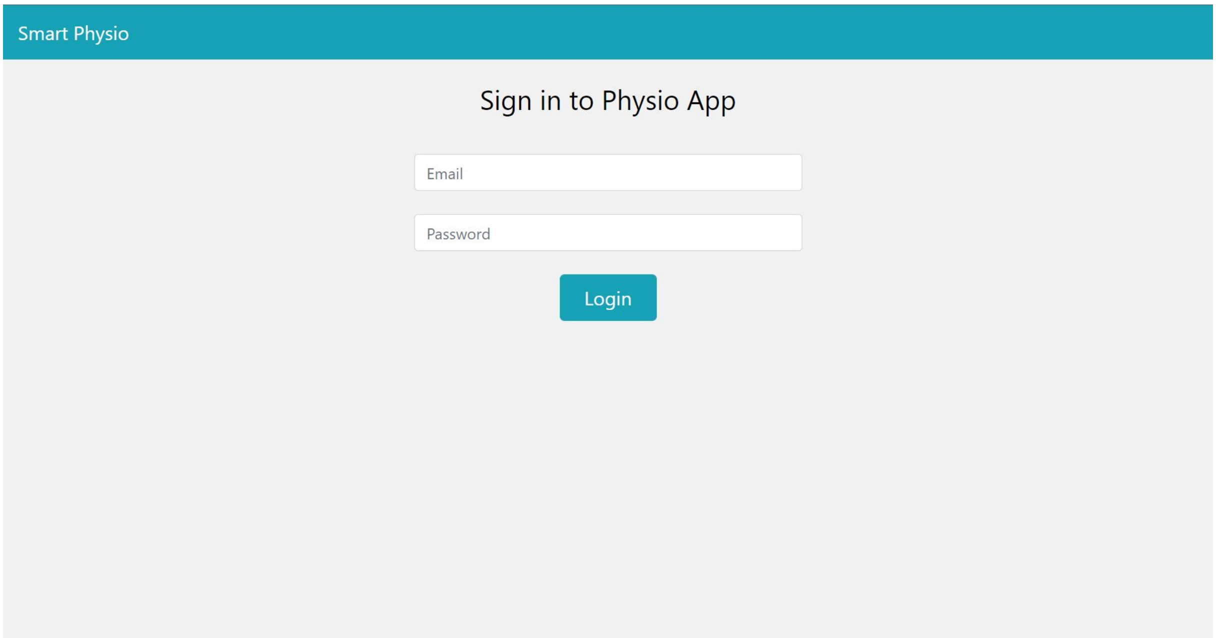
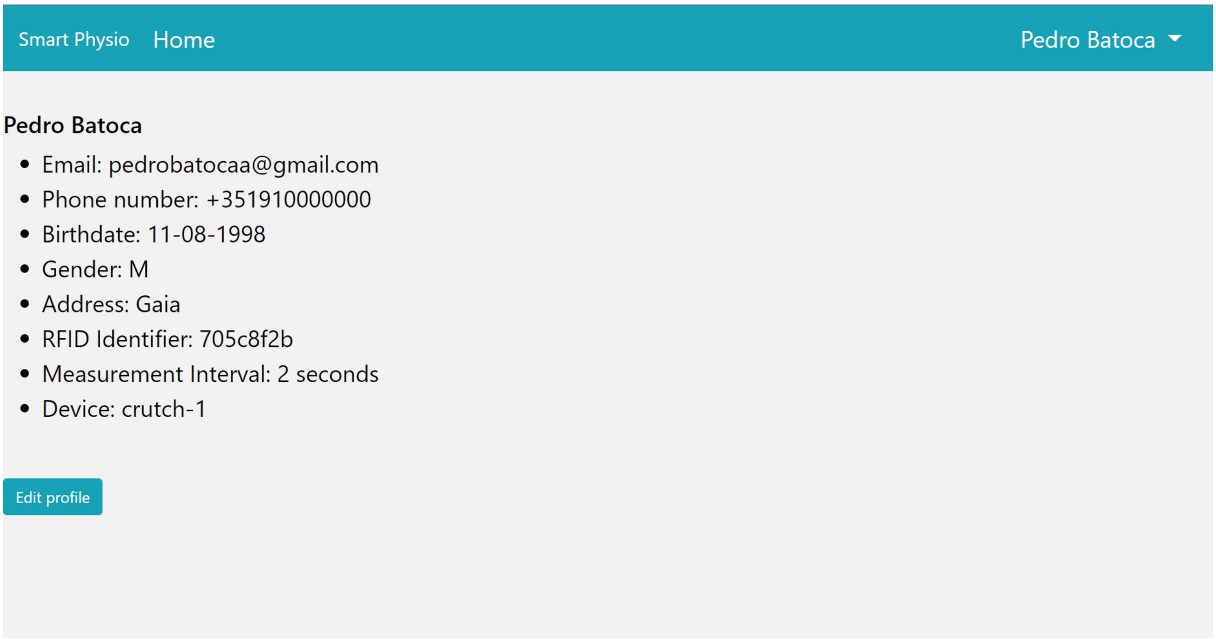


FIGURE 21. Login page

Figure 21 shows the login screen for the users. After logging in the users are forward to their homepages that are detailed on the users functionalities.



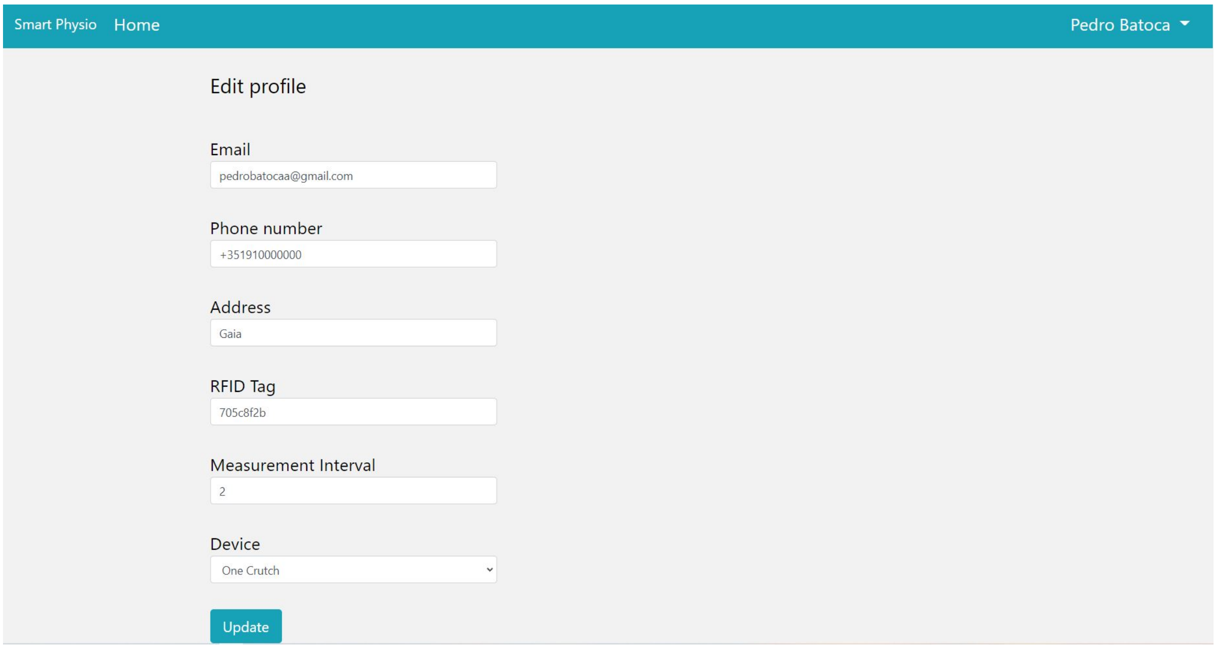
Pedro Batoca

- Email: pedrobatocaa@gmail.com
- Phone number: +351910000000
- Birthdate: 11-08-1998
- Gender: M
- Address: Gaia
- RFID Identifier: 705c8f2b
- Measurement Interval: 2 seconds
- Device: crutch-1

FIGURE 22. User profile page

All users can view their profile, to achieve it, they need to click on their name and then on profile. Afterward, a page with their data will appear as seen in Figure 22

To change any of its attributes, the patient has to click on the "Edit profile" button, and then a new page will appear with text boxes to change the desired information:



Edit profile

Email: pedrobatocaa@gmail.com

Phone number: +351910000000

Address: Gaia

RFID Tag: 705c8f2b

Measurement Interval: 2

Device: One Crutch

FIGURE 23. Edit user profile page

5.4.1. Patient functionalities

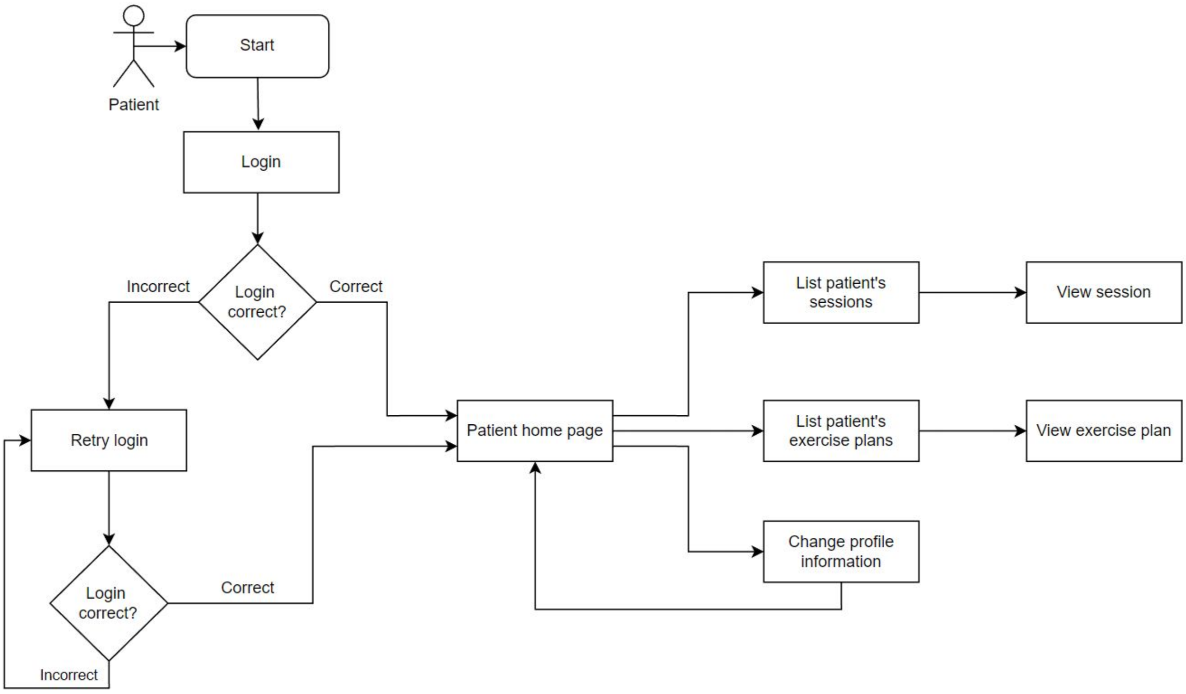
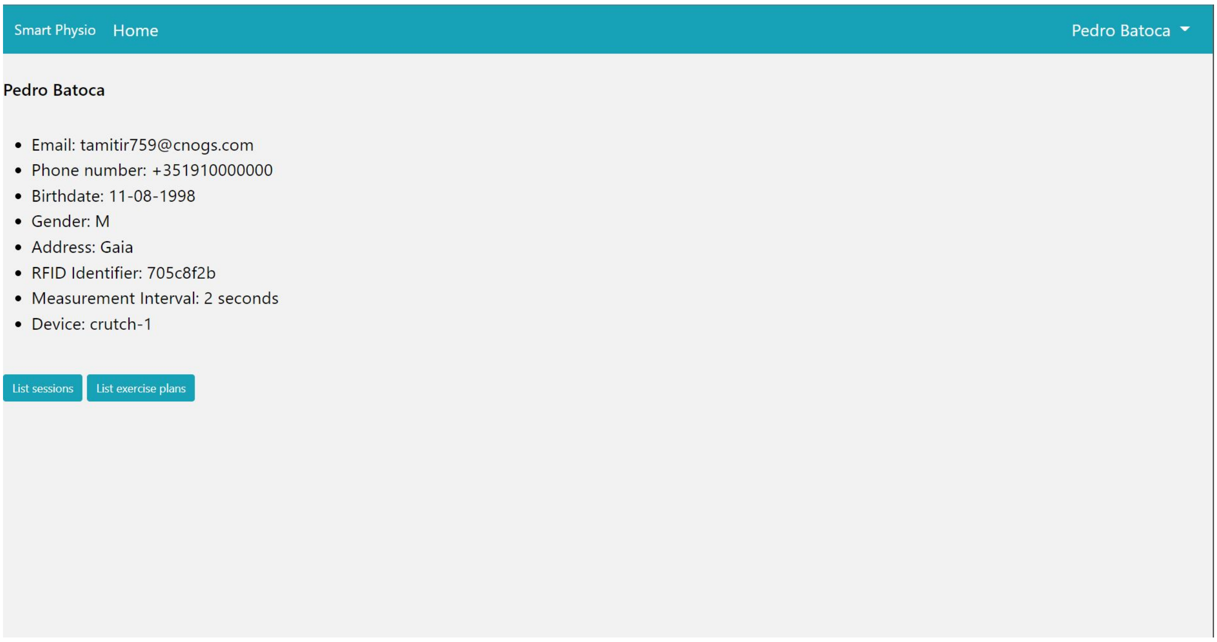


FIGURE 24. Patient flowchart

After the login, patients are sent to their homepage, in which they have their information available, alongside two buttons, one to view the list of sessions and the other to view the exercise plans.

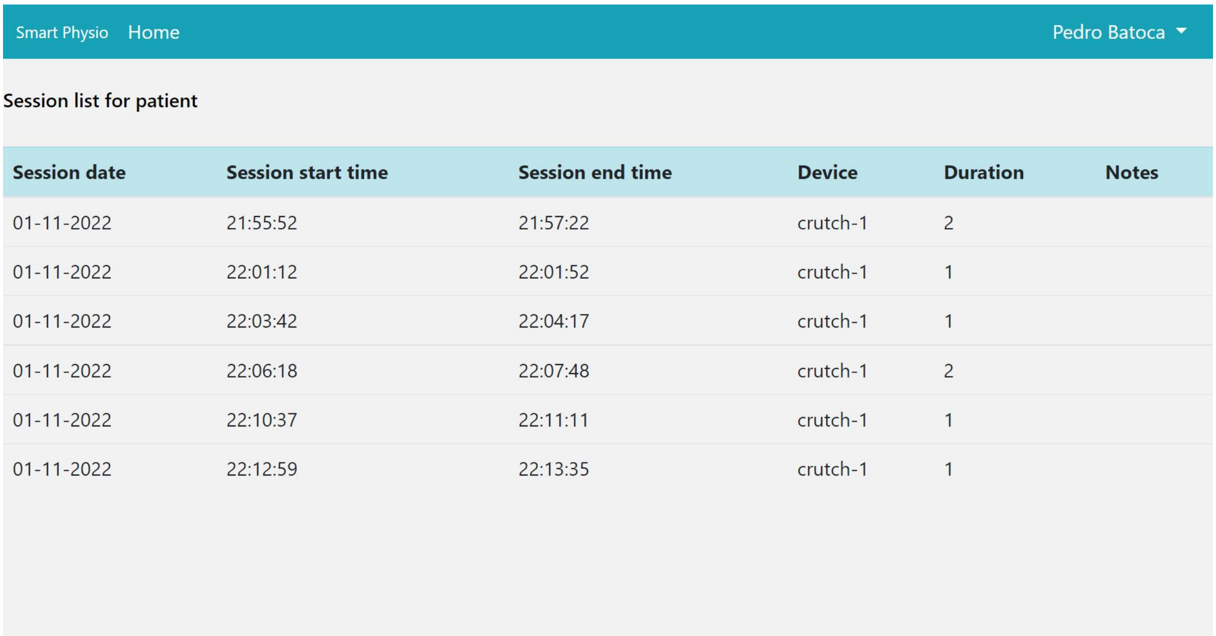


Pedro Batoca

- Email: tamitir759@cnogs.com
- Phone number: +351910000000
- Birthdate: 11-08-1998
- Gender: M
- Address: Gaia
- RFID Identifier: 705c8f2b
- Measurement Interval: 2 seconds
- Device: crutch-1

FIGURE 25. Patient homepage

If a patient desires to view his sessions, he has to click on the list sessions button and then is forwarded to another page with this information:

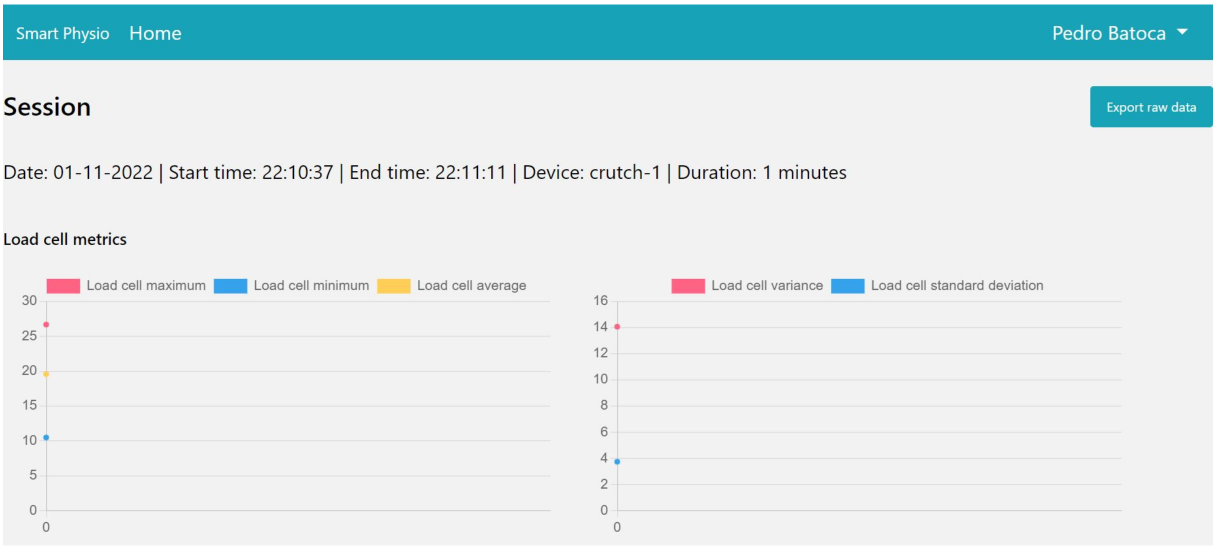


Session list for patient

Session date	Session start time	Session end time	Device	Duration	Notes
01-11-2022	21:55:52	21:57:22	crutch-1	2	
01-11-2022	22:01:12	22:01:52	crutch-1	1	
01-11-2022	22:03:42	22:04:17	crutch-1	1	
01-11-2022	22:06:18	22:07:48	crutch-1	2	
01-11-2022	22:10:37	22:11:11	crutch-1	1	
01-11-2022	22:12:59	22:13:35	crutch-1	1	

FIGURE 26. Session list page

To view the data of the desired session, the patient needs to click on the table row corresponding to the session and will be forwarded to a page with the session data.



Session

Date: 01-11-2022 | Start time: 22:10:37 | End time: 22:11:11 | Device: crutch-1 | Duration: 1 minutes

Load cell metrics

FIGURE 27. Session page with the session information and load cell metrics

On figure 27, it is possible to view the session information such as the date, start and end time, device, and duration. It is also possible to extract all of the raw data to a CSV file by clicking on the ”Export raw data” button. The metrics data is split into multiple figures for better viewing and in this one is possible to view the metrics of the load cell such as maximum, minimum, average, variation, and standard deviation.

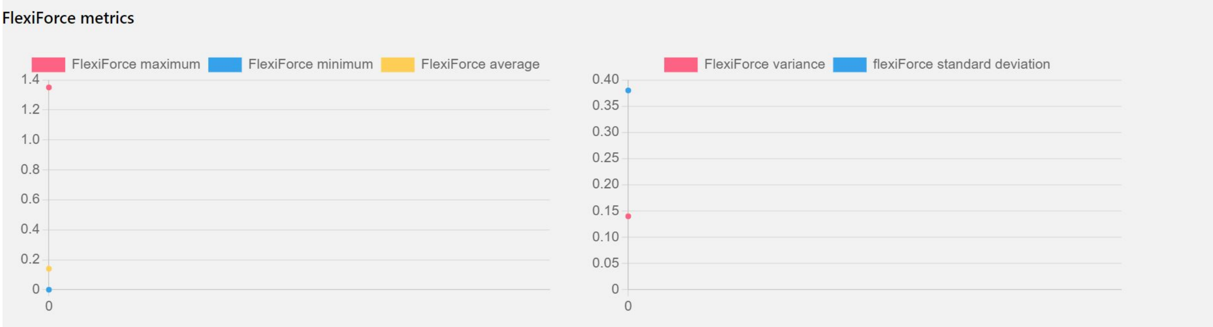


FIGURE 28. Session page with the flexiForce metrics

Figure 28 shows the metrics for the FlexiForce sensor, which include the maximum, minimum, average, variation, and standard deviation.

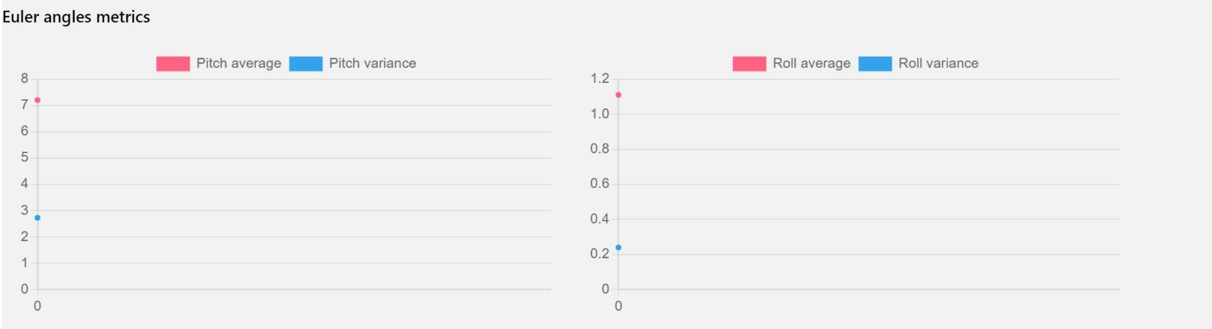


FIGURE 29. Session page with the Pitch and Roll metrics

Figure 29 shows the average and variance metrics for the pitch and roll angles

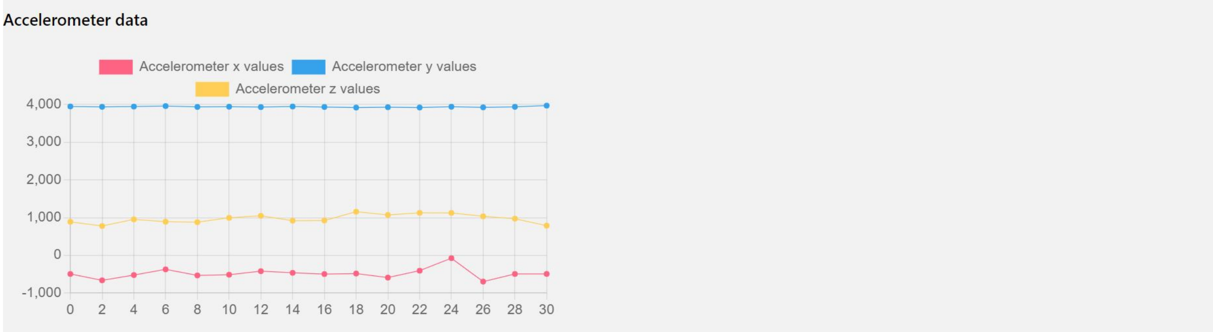


FIGURE 30. Session page with the accelerometer evolution

Figure 30 shows the accelerometer in three axes, x, y, and z and allows us to better understand the patient's movements during the session.

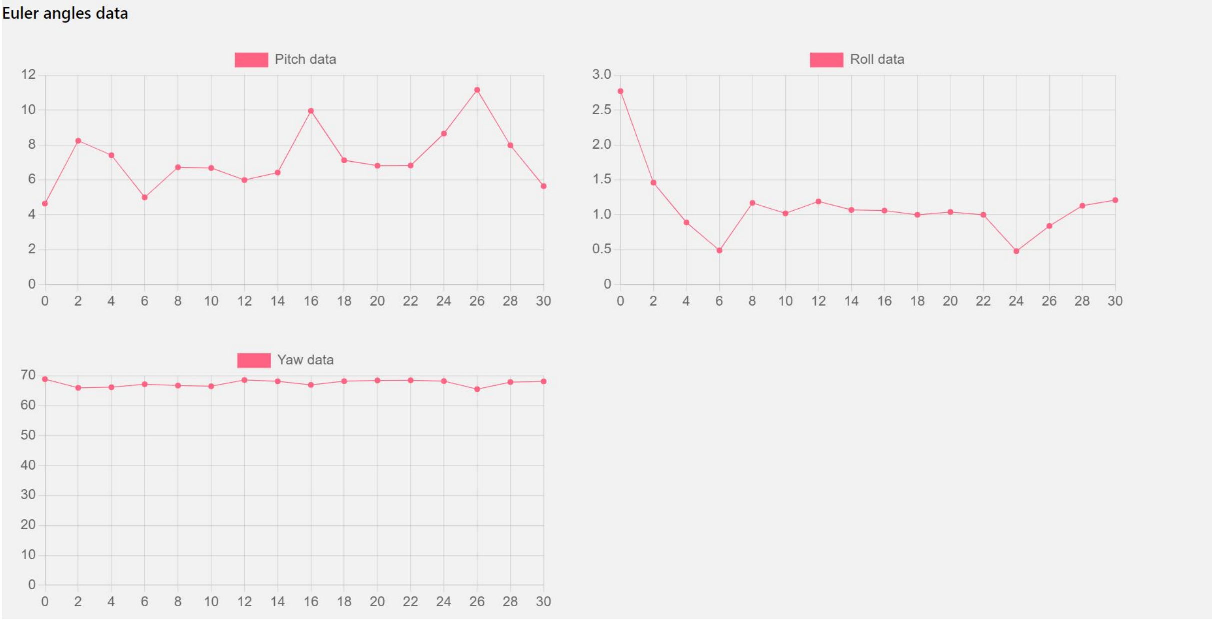
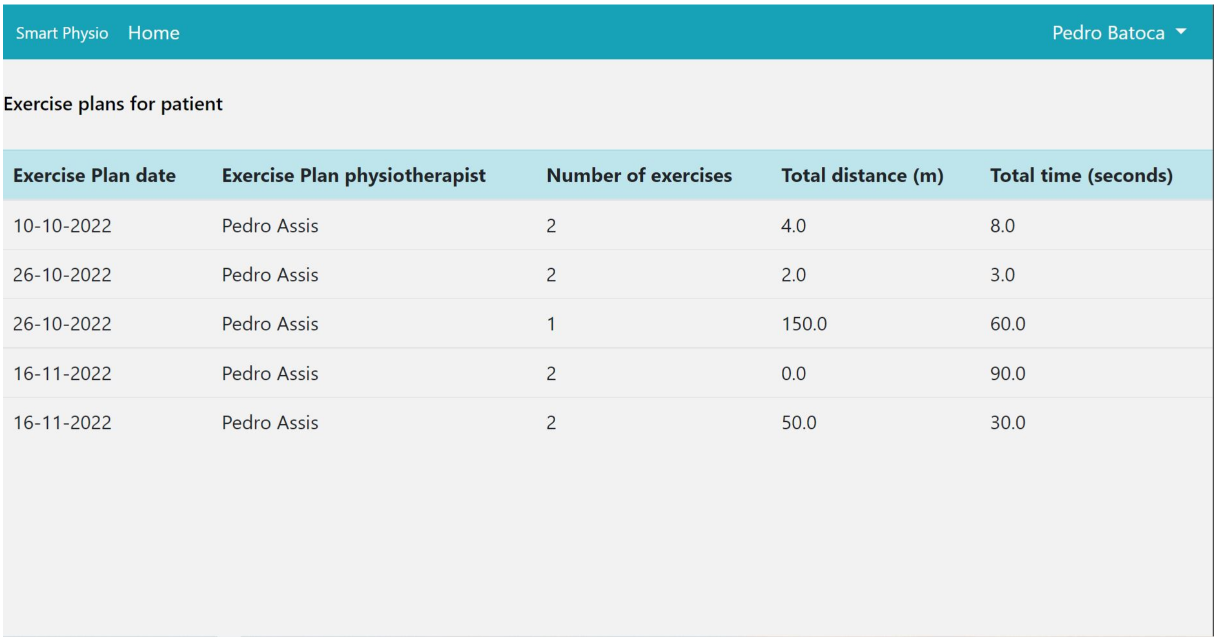


FIGURE 31. Session page with the accelerometer evolution

Figure 31 shows the pitch, roll and yaw evolution during the session to understand the number of steps that a patient gave and the balance with the goal to understand if there is a need for a walking aid or not.

If a patient wishes to see the the exercise plans, he has to click on list exercise plans and is forwarded another page:

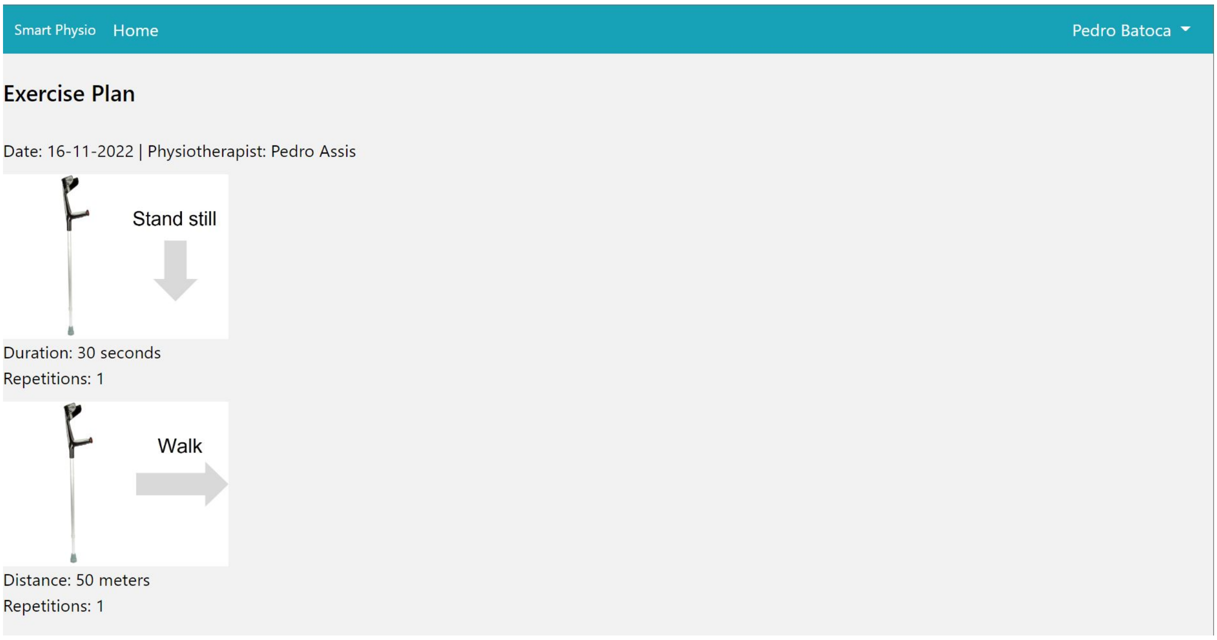


Exercise plans for patient

Exercise Plan date	Exercise Plan physiotherapist	Number of exercises	Total distance (m)	Total time (seconds)
10-10-2022	Pedro Assis	2	4.0	8.0
26-10-2022	Pedro Assis	2	2.0	3.0
26-10-2022	Pedro Assis	1	150.0	60.0
16-11-2022	Pedro Assis	2	0.0	90.0
16-11-2022	Pedro Assis	2	50.0	30.0

FIGURE 32. Exercise plans list page

To view the desired exercise plan, the patient needs to click on the table row corresponding to the plan and will be forwarded to a page with the exercise plan data.



Exercise Plan

Date: 16-11-2022 | Physiotherapist: Pedro Assis

Duration: 30 seconds
Repetitions: 1

Distance: 50 meters
Repetitions: 1

FIGURE 33. Exercise plan example

Figure 33 shows an example of an exercise plan where a user can view the exercises prescripted by the physiotherapist in terms of the exercise type, duration, distance, and the number of repetitions.

5.4.2. Physiotherapist functionalities

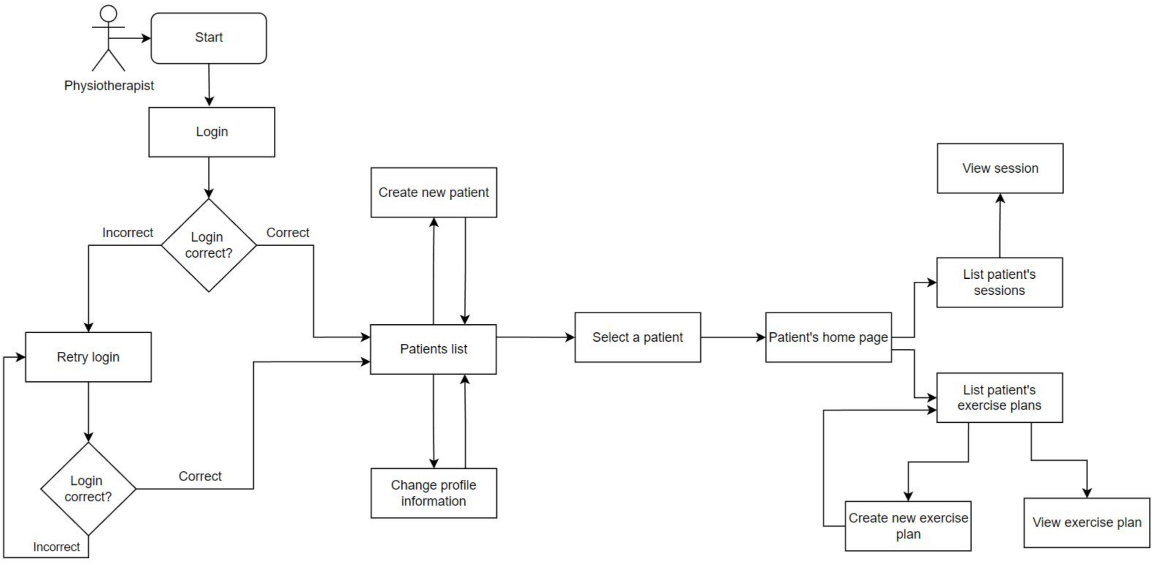
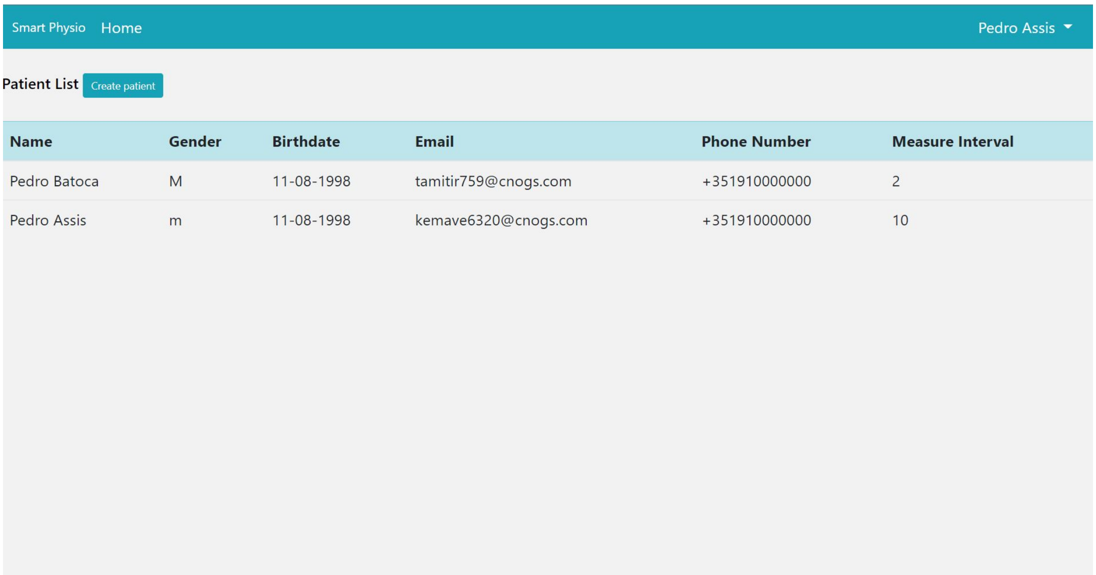


FIGURE 34. Physiotherapist flowchart

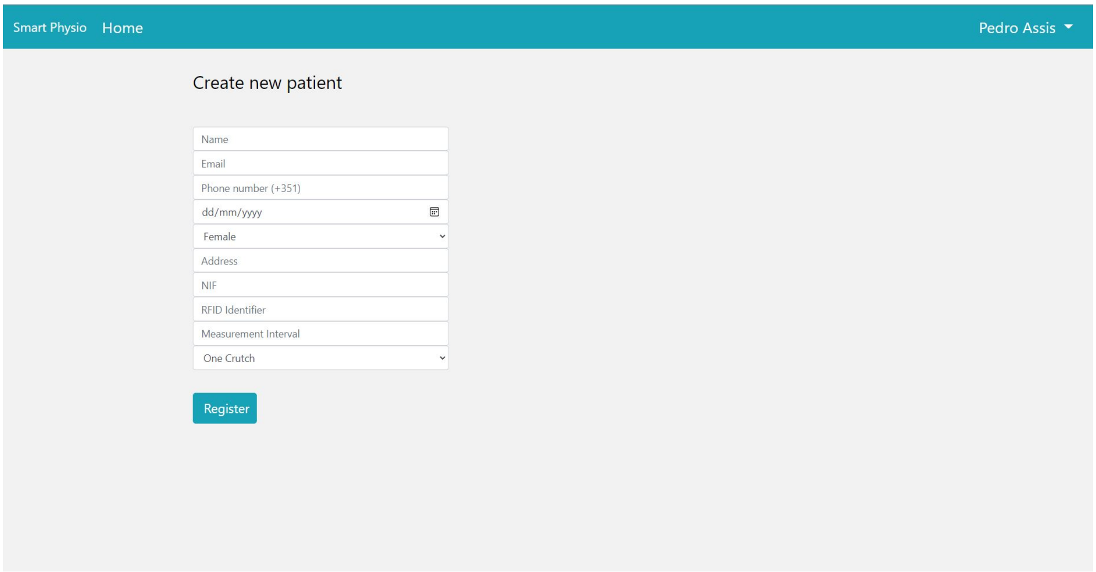
After the login, physiotherapists are sent to their homepage, in which they have the list of patients and a button to create new patients if needed.



Name	Gender	Birthdate	Email	Phone Number	Measure Interval
Pedro Batoca	M	11-08-1998	tamitir759@cnogs.com	+351910000000	2
Pedro Assis	m	11-08-1998	kemave6320@cnogs.com	+351910000000	10

FIGURE 35. Physiotherapist homepage

If a physiotherapist wishes to create a new patient, he has to click on the ”Create patient” button and is forwarded to other page.

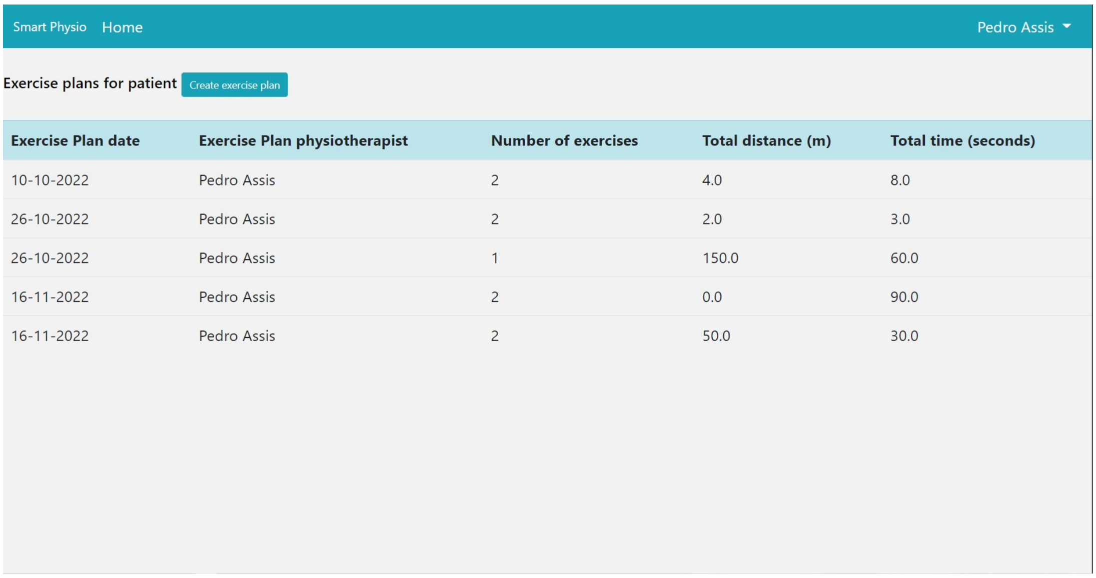


Name
Email
Phone number (+351)
dd/mm/yyyy
Female
Address
NIF
RFID Identifier
Measurement Interval
One Crutch

FIGURE 36. Create patient page

After the patient is created, the patient receives an email with a temporary password that needs to be changed after the first login.

After selecting a patient, the physiotherapist has the same functionalities as the patient in terms of viewing sessions and exercise plans, the only change is in the exercise plans.

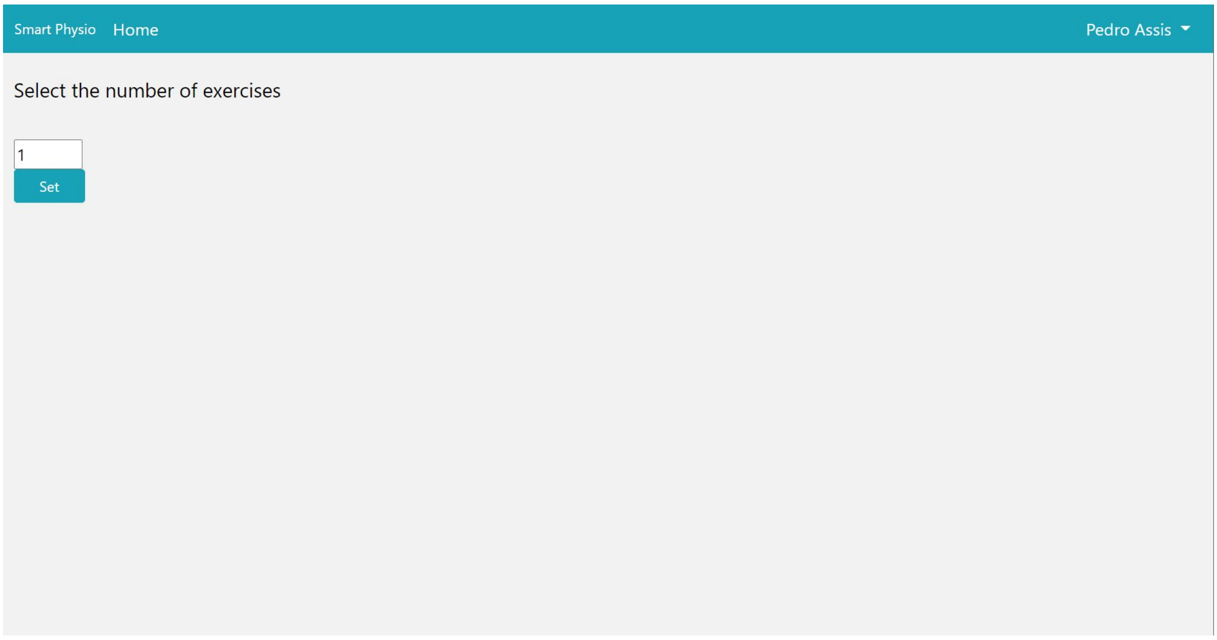


Exercise plans for patient

Exercise Plan date	Exercise Plan physiotherapist	Number of exercises	Total distance (m)	Total time (seconds)
10-10-2022	Pedro Assis	2	4.0	8.0
26-10-2022	Pedro Assis	2	2.0	3.0
26-10-2022	Pedro Assis	1	150.0	60.0
16-11-2022	Pedro Assis	2	0.0	90.0
16-11-2022	Pedro Assis	2	50.0	30.0

FIGURE 37. Exercise plans list page for the physiotherapist

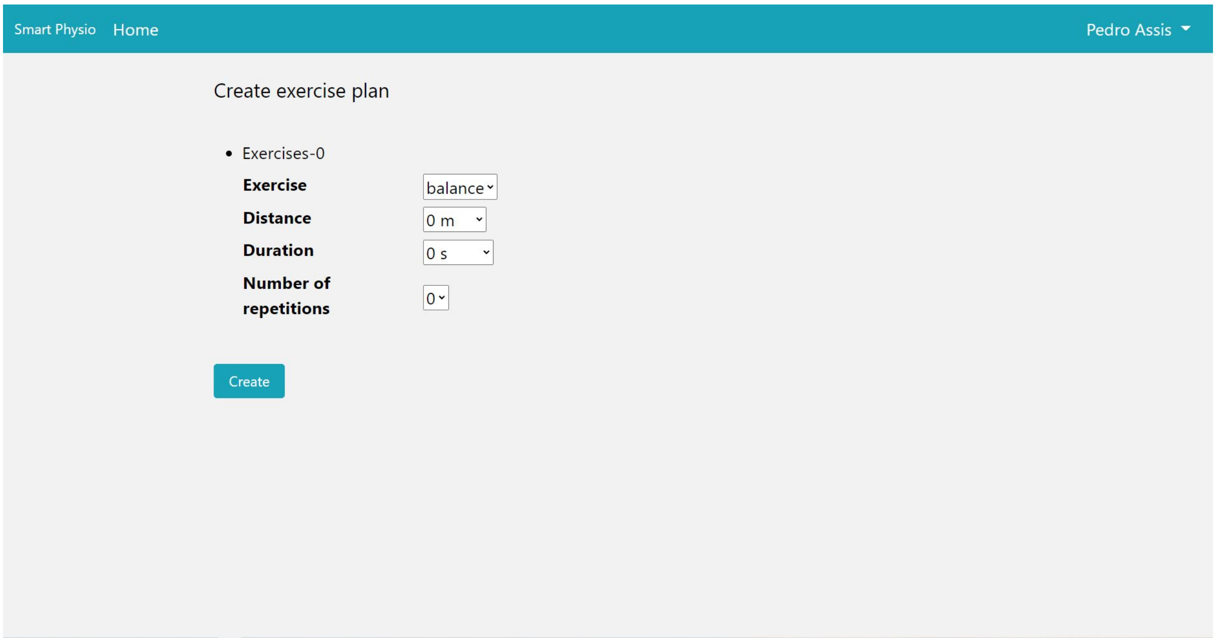
Figure 37 shows the list of exercise plans for the physiotherapist. Compared with the patient, it is possible to see that there is an additional button that allows the physiotherapist to create exercise plans. To create a new exercise plan, the physiotherapist has to first specify the desired number of exercises as shown in figure 38 and then is forwarded to a different page as shown in figure 39 where is possible to select the exercise type, duration or distance and the number of repetitions. The exercise types are mapped with images stored in the S3 bucket so that for each exercise there is a picture of it to present to the patient.



Select the number of exercises

1

FIGURE 38. Number of exercises of the plan



Create exercise plan

- Exercises-0

Exercise: balance

Distance: 0 m

Duration: 0 s

Number of repetitions: 0

FIGURE 39. Exercise plans list page for the physiotherapist

5.4.3. Admin functionalities

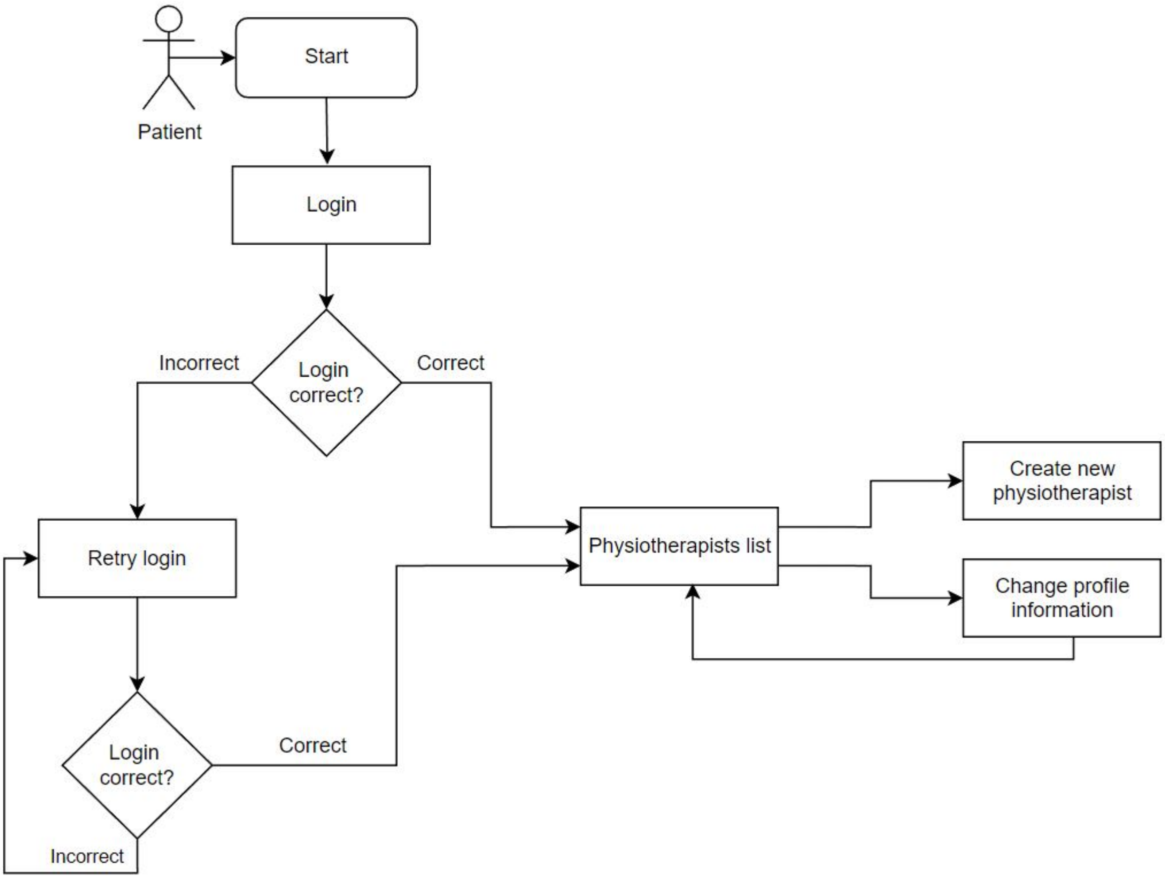
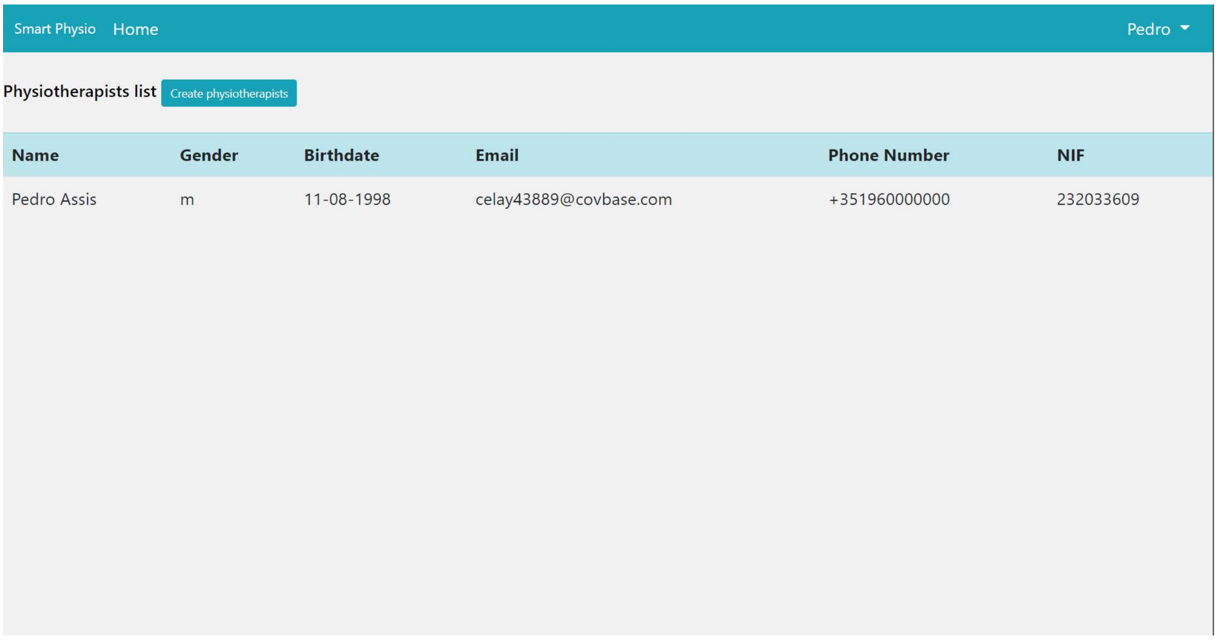


FIGURE 40. Admin flowchart

After the login, admins are sent to their homepage, in which they have the list of physiotherapists and a button to create new physiotherapists if needed.



Physiotherapists list

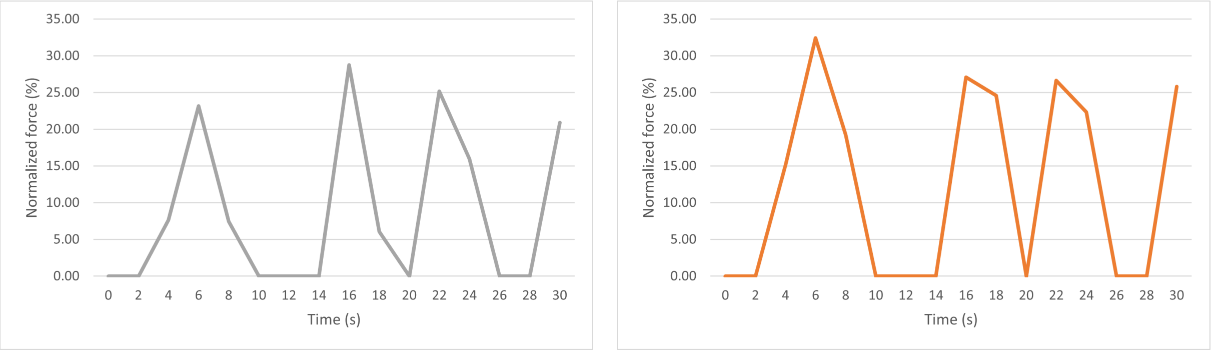
Name	Gender	Birthdate	Email	Phone Number	NIF
Pedro Assis	m	11-08-1998	celay43889@covbase.com	+351960000000	232033609

Figure 41. Admin homepage

CHAPTER 6

Results

The system was tested by two healthy volunteers, a male and a female aged 24 and 25, with a respective BMI of 25.2 and 24.8. All the executed tests had a duration of 30 seconds. The first one was to walk in a straight line, the second was a balance exercise which consists of staying still, and the last one was also a balance exercise however, the goal was to simulate a patient that needs to rely on a crutch to stand still. It was possible to gather data regarding the force applied on the bottom and hand support of the crutch as well as the Euler angles. These angles are based on a Kalman filter applied to the measurements from the accelerometer, magnetometer, and gyroscope as explained in the hardware chapter, and provide reliable information about the rotation of the crutch during the exercises, mainly if the patient gave a step or inverted the marching way. The data was recorded by the crutch and sent to the cloud for storage, and Microsoft Excel was used to generate the following graphs.

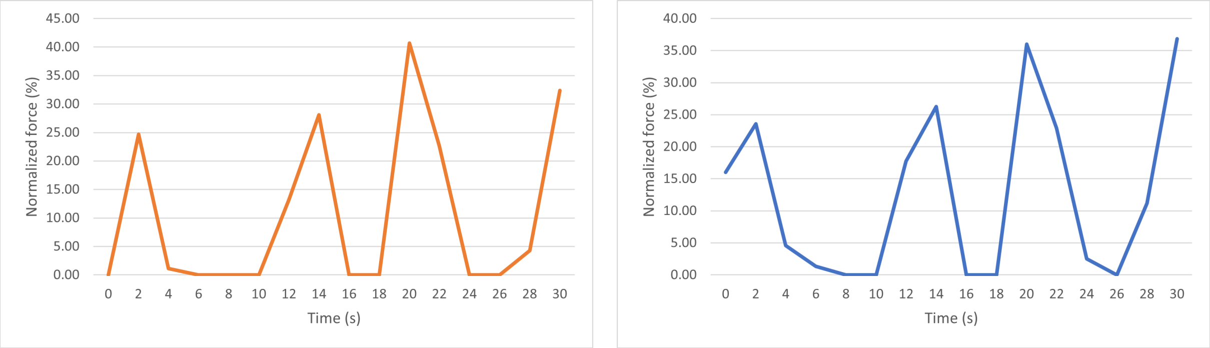


(A) FlexiForce measurements

(B) Load cell measurements

FIGURE 1. Force evolution during walking for the female volunteer

Figure 1, shows the force induced on the sensors present in the crutch as a function of the duration of the exercise for the female patient. The force is normalized to the maximum sensor value and is presented as a percentile value. The different sensors, loadcell, and flexiforce show agreeable results for both cases, when we see a peak for one sensor, there is a corresponding peak on the other. For the load cell, the average force applied was 12.09%, the minimum was 0%, and the maximum was 32.42%. For the flexiForce, the minimum was also 0%, the maximum was 28.76%, and the average was 8.44%. It is possible to see that the values reach 0 when there is a higher variation of Euler angles which means that the crutch is in the air while the patient is taking a step.

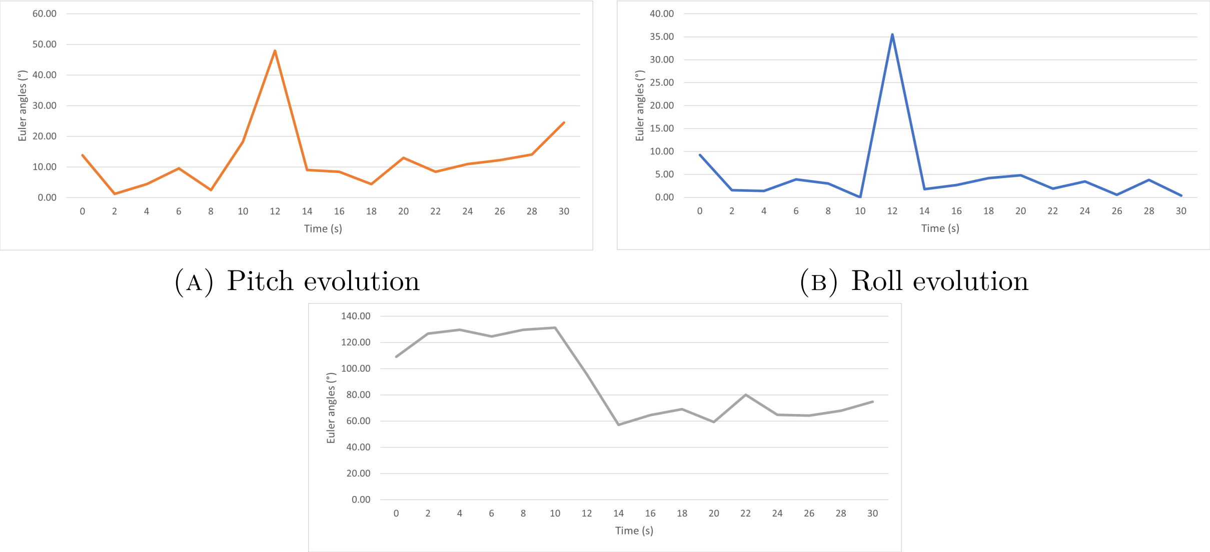


(A) FlexiForce measurements (B) Load cell measurements

FIGURE 2. Force evolution during walking for the male volunteer

Figure 2 shows the same information for the male patient. The different sensors, loadcell, and flexiforce show agreeable results for both cases, when we see a peak for one sensor, there is a corresponding peak on the other. For the load cell, the average force applied was 12.43%, the minimum was 0%, and the maximum was 36.83%. For the flexiForce, the minimum was also 0%, the maximum was 40.67%, and the average was 10.45%. It was also possible to verify that the values reach 0 when there is a higher variation of Euler angles which means that the crutch is in the air while the patient is taking a step.

Comparing both figures it is possible to see that both gave approximately the same number of steps(4), the average for the load cell is similar for both patients however the male volunteer applied a higher force on the hand support which might be related to a higher weight when compared to the other volunteer.

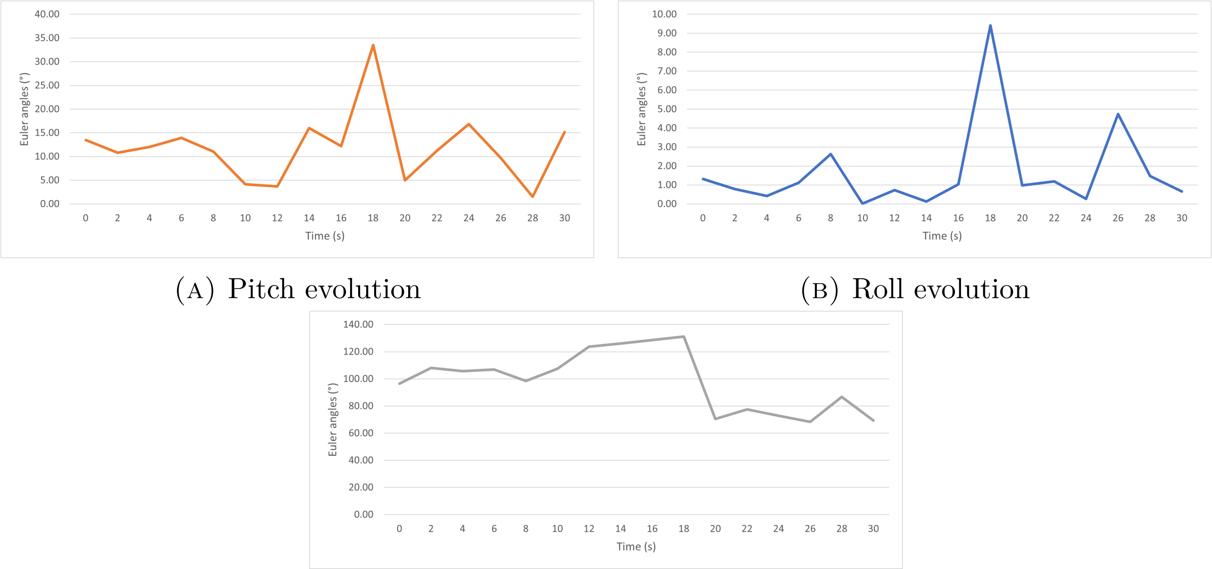


(A) Pitch evolution (B) Roll evolution

(C) Yaw evolution

FIGURE 3. Roll, pitch and yaw evolution during walking for the female volunteer

Figure 3, shows the roll and pitch angles evolution (in degrees) when the volunteer is walking for the female. For the pitch angle, the average was 12.67º, the maximum was 47.95º and the minimum was 1.21º. For the roll angle, the maximum was 35.50º, the minimum was 0.03º, and the average was 4.90º. Lastly, for the yaw angle, the average was 90.52º, the maximum was 131,24º and the minimum was 57.24º. It is possible to see that the maximum for the pitch and roll happened when the patient took more time to give a step (between 10-14 seconds), and the yaw also suffered the biggest drop also at this moment.



(A) Pitch evolution

(B) Roll evolution

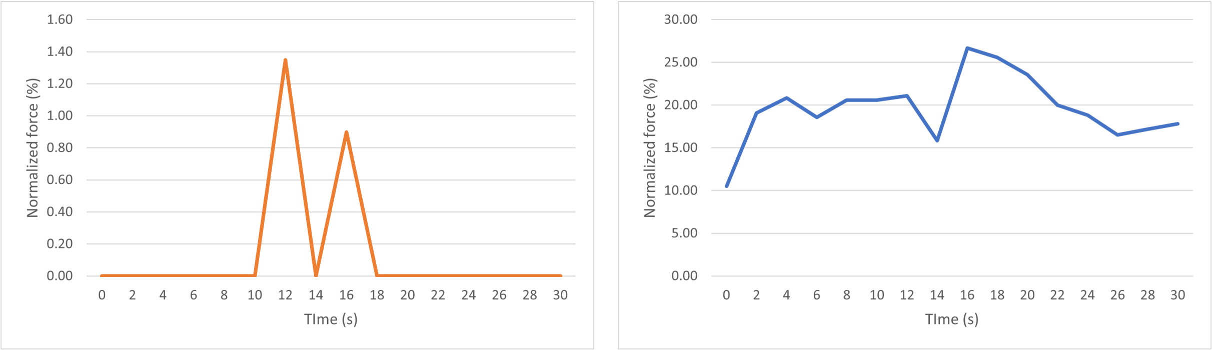
(C) Yaw evolution

Figure 4. Roll, pitch and yaw evolution during walking for the male volunteer

Figure 4 shows the same information as in 3 for the male patient. For the pitch angle, the average was 11.90º, the maximum was 33.52º and the minimum was 1.54º. For the roll angle, the maximum was 9.41º, the minimum was 0.01º, and the average was 1.68º. Lastly, for the yaw angle, the average was 98.60º, the maximum was 131,14º and the minimum was 68.3º.

Comparing both figures, it is possible to see that the male volunteer was more balanced because the variation between the pitch and roll averages and the minimum and maximum values were lower.

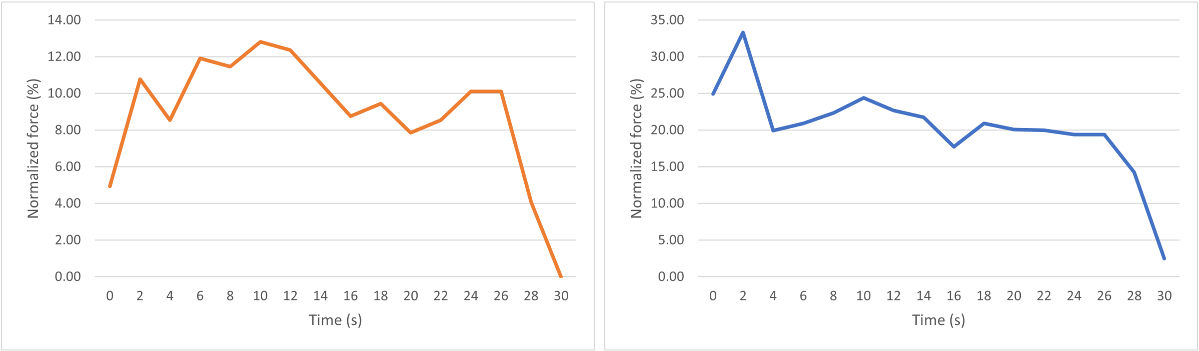
Figure 5, shows the force induced on the sensors present in the crutch as a function of the duration of the balancing exercise for the female patient. The force is normalized to the maximum sensor value and is presented as a percentile value.. For the load cell, the average force applied was 19.58%, the minimum was 10.50%, and the maximum was 26.67%. For flexiForce, the minimum was 0%, the maximum was 1.35%, and the average was 0.14%.



(A) FlexiForce measurements

(B) Load cell measurements

Figure 5. Force evolution during the balancing exercise for the female volunteer



(A) FlexiForce measurements

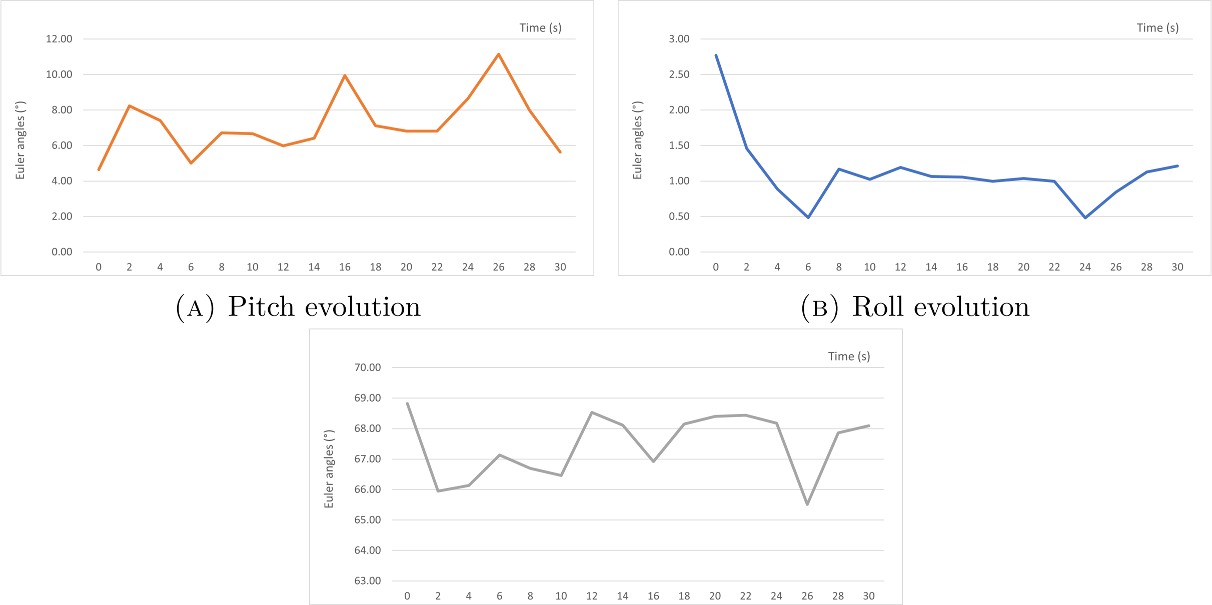
(B) Load cell measurements

Figure 6. Force evolution during the balancing exercise for the male volunteer

Figure 6, shows the same information for the male patient. The different sensors, loadcell, and flexiforce show agreeable results, when we see a peak for one sensor, there is a corresponding peak on the other. For the load cell, the average force applied was 20.29%, the minimum was 2.50%, and the maximum was 33.33%. For flexiForce, the minimum was 0%, the maximum was 12.81%, and the average was 8.89%. The lowest values were reached at the end of the session when the patient was stopping the session.

Comparing both figures it is possible to see that in this case there are no decreases to 0 and then spikes, which means that the volunteers did not take any steps during the exercise which was the goal, and the maximum forces applied were lower than the walking exercise, and the female patient probably was not holding or using the crutch properly which explains the measurements of the load cell.

Figure 7, shows the roll and pitch angles evolution (in degrees) during the balancing exercise for the female patient. For the pitch angle, the average was 7.20º, the maximum was 11.15º and the minimum was 4.64º. For the roll angle, the maximum was 2.77º, the minimum was 0.48º, and the average was 1.11º. Lastly, for the yaw angle, the average was 67.46º, the maximum was 68.83º and the minimum was 65.51º. It is possible to see that the deviation from the maximum and minimum values to the average is small, which



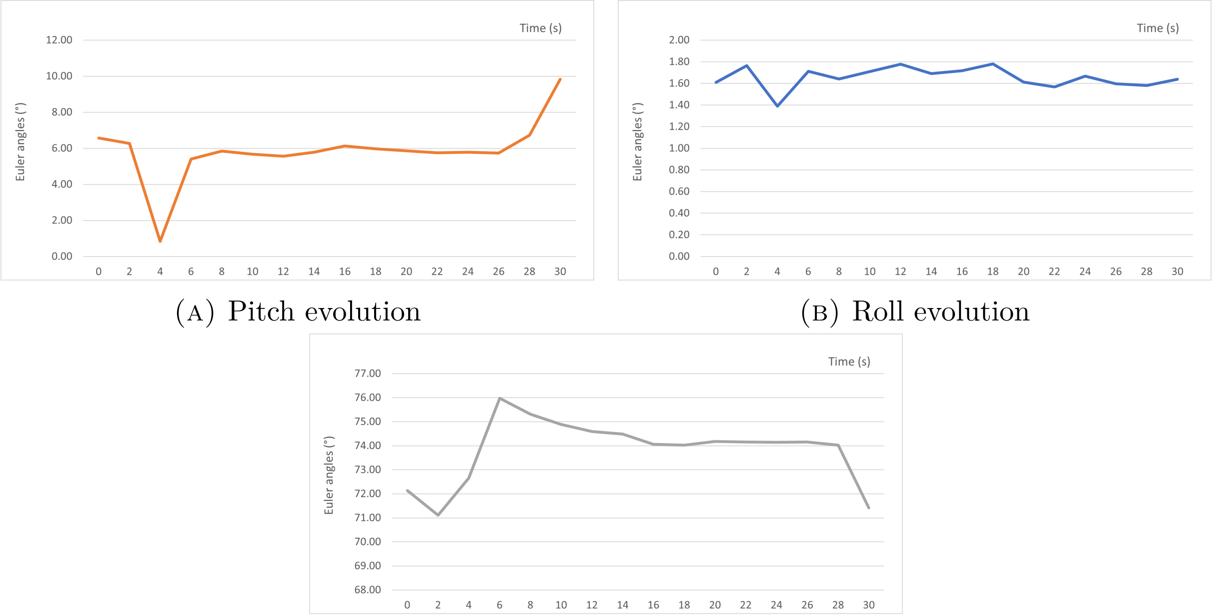
(A) Pitch evolution

(B) Roll evolution

(C) Yaw evolution

FIGURE 7. Roll, pitch and yaw evolution during the balancing exercise for the female volunteer

means that the patient was steadily holding the crutch during the exercise, as evidenced by the volunteer not making force on the crutch.



(A) Pitch evolution

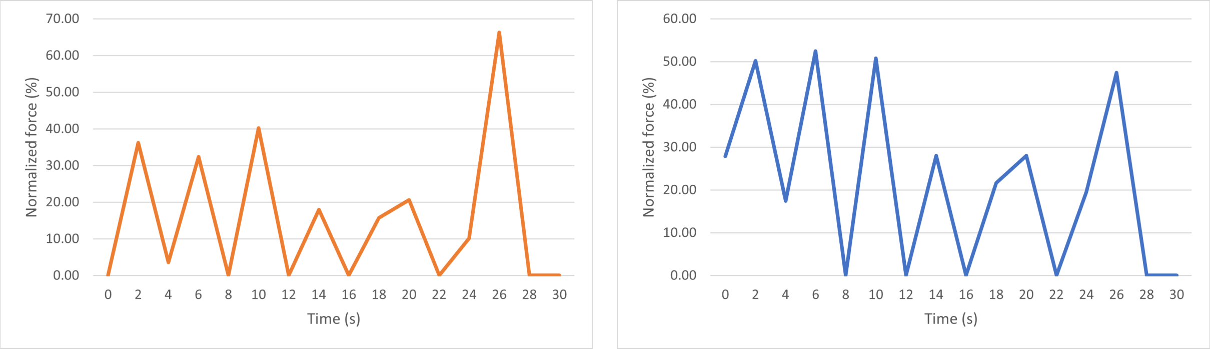
(B) Roll evolution

(C) Yaw evolution

FIGURE 8. Roll, pitch and yaw evolution during the balancing exercise for the male volunteer

Figure 8 shows the same information for the male volunteer. For the pitch angle, the average was 5.86º, the maximum was 9.84º and the minimum was 0.84º. For the roll angle, the maximum was 1.78º, the minimum was 1.39º, and the average was 1.65º. Lastly, for the yaw angle, the average was 73.83º, the maximum was 75.97º and the minimum was 71.11º. There are angles that show two spikes, one on the 4th second and the other on the 30th second, associated with an instant movement of the crutch.

From figures 7 and 8, it is not possible to compare the performance of both volunteers because the female data shows that the exercise was not performed correctly.

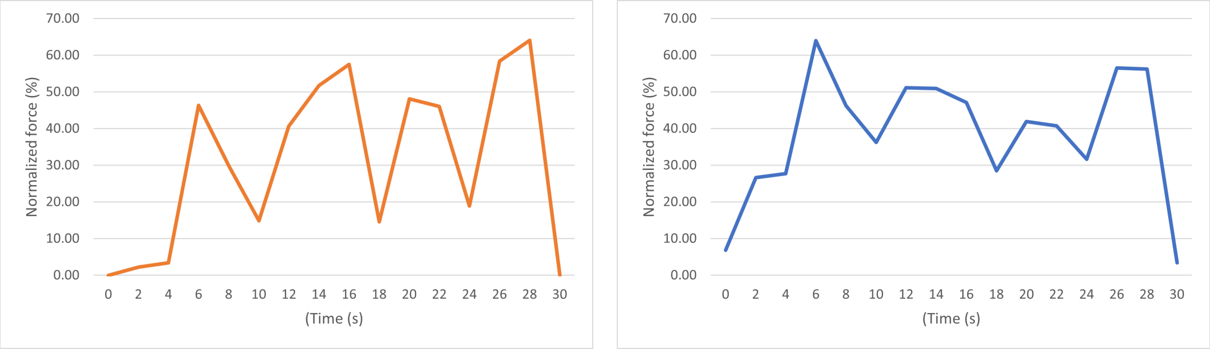


(A) FlexiForce measurements

(B) Load cell measurements

Figure 9. Force evolution during the balancing simulation exercise for the female volunteer

Figure 9, shows the force induced on the sensors present in the crutch as a function of the duration of the balancing exercise for the female patient. The force is normalized to the maximum sensor value and is presented as a percentile value. The different sensors, loadcell, and flexiforce show agreeable results, when we see a peak for one sensor, there is a corresponding peak on the other. For the load cell, the average force applied was 21.45%, the minimum was 0%, and the maximum was 52.20%. For flexiForce, the minimum was 0%, the maximum was 66.29%, and the average was 15.20%.



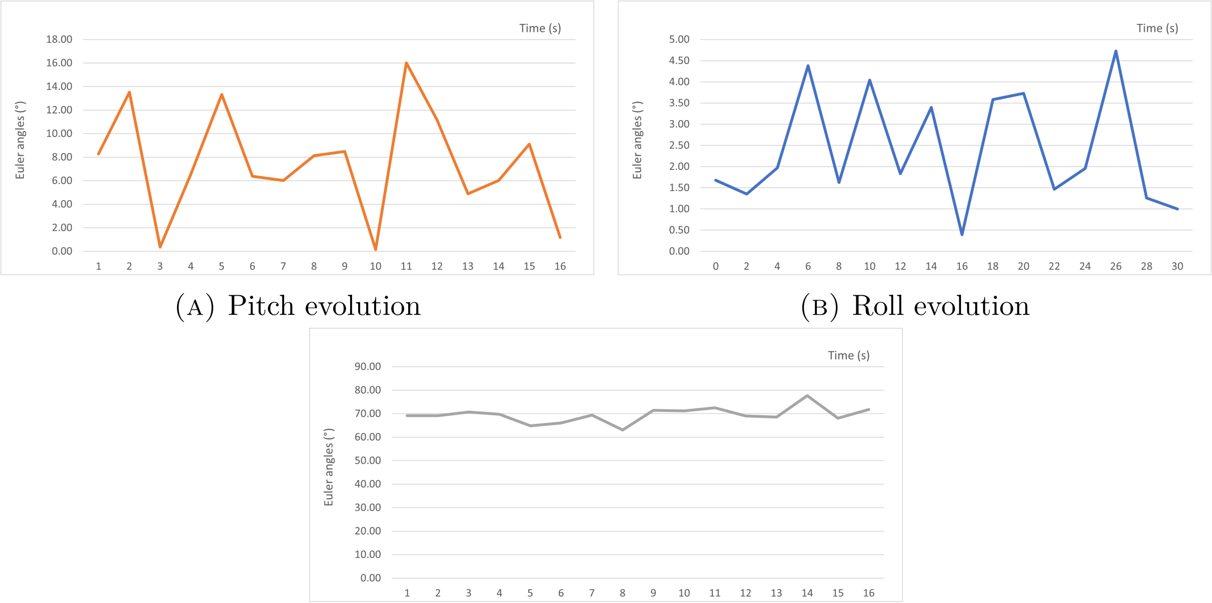
(A) FlexiForce measurements

(B) Load cell measurements

Figure 10. Force evolution during the balancing simulation exercise for the male volunteer

Figure 10, shows the same information for the male patient. TThe different sensors, loadcell, and flexiforce show agreeable results, when we see a peak for one sensor, there is a corresponding peak on the other. For the load cell, the average force applied was 38.48%, the minimum was 3.42%, and the maximum was 64.00%. For flexiForce, the minimum was 0%, the maximum was 64.04%, and the average was 31.04%. As expected, the maximum and minimum values vary much more from the average when simulating the need for a walking aid.

Comparing both figures for both volunteers it was possible to verify in the values the expected behavior when a patient needs to use a walking aid.



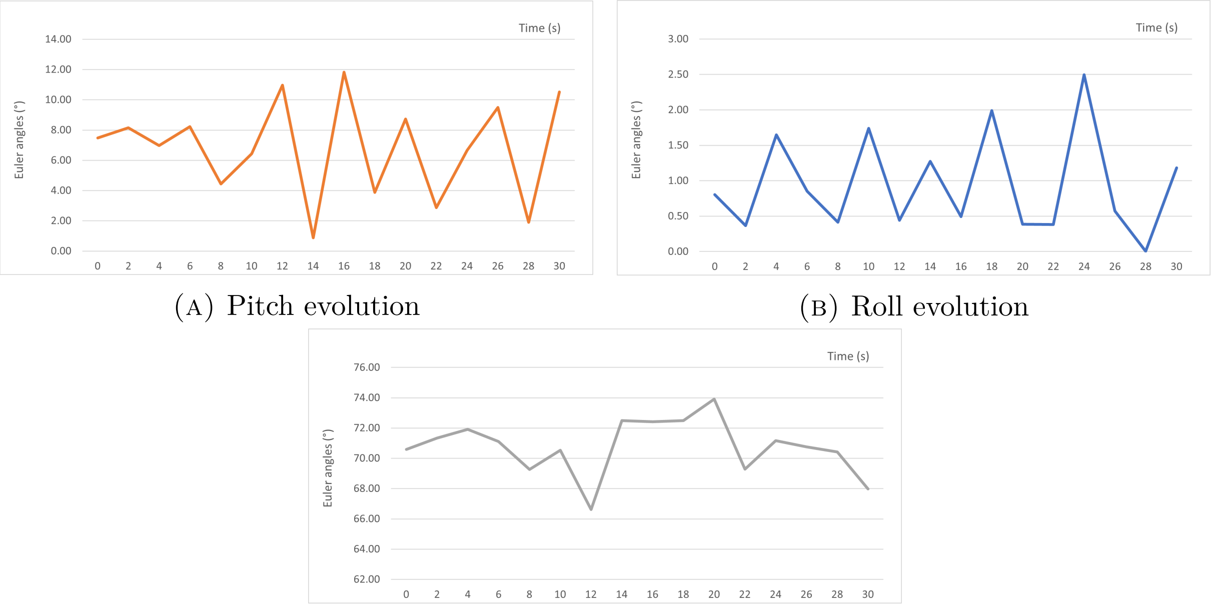
(A) Pitch evolution

(B) Roll evolution

(C) Yaw evolution

FIGURE 11. Roll, pitch and yaw evolution during the balancing simulation exercise for the female volunteer

Figure 11, shows the roll and pitch angles evolution (in degrees) during the balancing exercise for the female patient. For the pitch angle, the average was 7.47º, the maximum was 16.01º and the minimum was 0.15º. For the roll angle, the maximum was 4.73º, the minimum was 0.39º, and the average was 2.40º. Lastly, for the yaw angle, the average was 69.53º, the maximum was 77.64º and the minimum was 63.04º. As expected there are greater variations in the angles when compared with exercise 2 that are associated with the variations in the force applied.



(A) Pitch evolution

(B) Roll evolution

(C) Yaw evolution

FIGURE 12. Roll, pitch and yaw evolution during the balancing simulation exercise for the male volunteer

Figure 12 shows the same information for the male volunteer. For the pitch angle, the average was 6.84º, the maximum was 11.83º and the minimum was 0.88º. For the roll angle, the maximum was 2.50º, the minimum was 0º, and the average was 0.94º. Lastly, for the yaw angle, the average was 70.77º, the maximum was 73.90º and the minimum was 66.62º. It was also possible to verify that there are greater variations in the angles when compared with exercise 2 that are associated with the variations in the force applied.

From figures 11 and 12, it is possible to conclude that the male patient was more balanced during the exercise due to a lower variation from the minimum and maximum to the average of the angles measured.

CHAPTER 7

Conclusions and Future work

7.1. Conclusions

Physical therapy aims to help patients recover from impairments and provide better conditions for elderly people or patients that need walking aids on their day-to-day to have greater independence. As mentioned, a large part of the population that needs this kind of care does not have access to physical rehabilitation and it is imperative to change this. The World Health Organization already took a step in this direction with the Rehabilitation 2030 to provide physical rehabilitation to the whole population however, it is also needed to keep investigating and develop solutions that can help to achieve this goal. Alongside this need, there is also the objective to provide objective data to physiotherapists to improve patients' treatment and recovery times.

The contribution of this dissertation is the development of a smart crutch and a web application to provide data to patients and physiotherapists. The prototype of the smart crutch uses off-the-shelf components mainly, a crutch, a load cell, a flexiForce sensor, an IMU, an RFID reader, and an ESP32. This setup allows the retrieval of metrics associated with the force applied both on the hand support, the bottom of the crutch, and the patient balance (acceleration, pitch, roll, and yaw), thus allowing objective evaluation of gait rehabilitation. The data was extracted successfully and by leveraging cloud services, it was possible to have a remote storage location in S3 that provides redundancy and high availability which is mandatory for healthcare systems. Regarding the safety of the user data, each patient only has access to their information, and this solution can be extended to other clinics with complete isolation guaranteeing data security. It was also possible to shift the processing of the information to AWS lambda right after a session ends. In addition to the crutch, a web application provides information about each session and exercise plan for both patients and physiotherapists. There is also the possibility to create new patients and physiotherapists if needed. The results proved that this is a capable solution to provide the metrics to physiotherapists to analyze the patient's need for physical rehabilitation. With a larger sample of data from healthy and unhealthy individuals, it would be possible to create a model that would allow us to understand if a patient is healthy or not.

7.2. Future work

The design prototype aims to create an end-to-end solution for physiotherapy clinics however, some items would need to be refined, mainly:

- Develop new metrics related to the patient training outcome

- Create machine learning models capable to identify patients that need walking aids for their safety and gait rehabilitation
- Increase the number of volunteers to assure the reliability of the data
- Add more types of equipment to the system such as wheelchairs, to measure the balance, the force applied on its arms, and also walkers where is possible to measure the balance of the patient, number of steps, bilateral elevation of the walker and the force applied on the walker's feet.

Bibliography

[1] NHS, "Physiotherapy." https://www.nhs.uk/conditions/physiotherapy/ Acccessed: 07 June 2022, nov 2018.

[2] United Nations Department of Economic and Social Affairs, Population Division, "World population ageing 2020 highlights: Living arrangements of older persons." https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/undesa_pd-2020_world_population_ageing_highlights.pdf, Acccessed: 07 June 2022, 2020.

[3] World Health Organization, "Who global report on falls prevention in older age." https://extranet.who.int/agefriendlyworld/wp-content/uploads/2014/06/WHo-Global-report-on-falls-prevention-in-older-age.pdf, Acccessed: 07 June 2022, 2007.

[4] World Health Organization, "Rehabilitation 2030." https://www.who.int/initiatives/rehabilitation-2030, Acccessed: 07 June 2022, 2017.

[5] S. Vishnu, S. J. Ramson, and R. Jegan, "Internet of medical things (iomt) - an overview," in *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*, pp. 101–104, 2020.

[6] Oracle, "What is iot?." https://www.oracle.com/pt/internet-of-things/what-is-iot/, Acccessed: 07 June 2022.

[7] Lionel Sujay Vailshery, "Number of internet of things (iot) connected devices worldwide from 2019 to 2030." https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/, Acccessed: 07 June 2022, 2022.

[8] Lionel Sujay Vailshery, "Internet of things (iot) total annual revenue worldwide from 2019 to 2030." https://www.statista.com/statistics/1194709/iot-revenue-worldwide/, Acccessed: 07 June 2022, 2022.

[9] Arduino Team, "Deep dive with dario: A closer look at the new arduino nano 33 iot." https://blog.arduino.cc/2019/05/24/getting-to-know-the-new-arduino-nano-33-iot/, Acccessed: 26 November 2022, 2019.

[10] Arduino Team, "Arduino nano 33 iot." https://docs.arduino.cc/hardware/nano-33-iot, Acccessed: 26 November 2022, 2022.

[11] Farnell, "Arduino nano 33 iot." https://pt.farnell.com/productimages/large/en_GB/3404692-40.jpg, Acccessed: 26 November 2022, 2022.

[12] Phil King, "What is a raspberry pi pico w and what can you use it for?." https://www.makeuseof.com/raspberry-pi-pico-w-what-you-can-use-it-for/, Acccessed: 26 November 2022, 2022.

[13] Robert Mauser, "Microcontrolador raspberry pi pico w com wifi - raspberry pi." https://mauser.pt/images/97710982c91efd0e11dcb58be9fa9ce1.jpg, Acccessed: 26 November 2022, 2022.

[14] Espressif Systems, "Esp32 wroom32 datasheet." https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf, Acccessed: 18 July 2022, 2022.

[15] Robert Mauser, "Microcontrolador arduino nano 33 iot - arduino." https://mauser.pt/catalog/product_info.php?products_id=096-7553, Acccessed: 26 November 2022, 2022.

[16] Robert Mauser, "Módulo wifi - espressif esp32-s2-saola-1m (wroom)." https://mauser.pt/catalog/product_info.php?cPath=1667_2889_2619&products_id=096-8586, Acccessed: 26 November 2022, 2022.

[17] Robert Mauser, "Microcontrolador raspberry pi pico w com wifi - raspberry pi." https://mauser.pt/catalog/product_info.php?cPath=1667_2620_1672\&products_id=095-0597, Acccessed: 26 November 2022, 2022.

[18] V. M. Ionescu and F. M. Enescu, "Investigating the performance of micropython and c on esp32 and stm32 microcontrollers," in *2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pp. 234–237, 2020.

[19] Robert Keim, "Introduction to the c programming language for embedded applications." https://www.allaboutcircuits.com/technical-articles/introduction-to-the-c-programming-language-for-embedded-applications/, Acccessed: 26 November 2022, 2019.

[20] C. B. Gemirter, Ç. Şenturca, and Ş. Baydere, "A comparative evaluation of amqp, mqtt and http protocols using real-time public smart city data," in *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pp. 542–547, 2021.

[21] N. Naik, "Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http," in *2017 IEEE International Systems Engineering Symposium (ISSE)*, pp. 1–7, 2017.

[22] HiveMQ, "Quality of service (qos) 0,1, and 2 mqtt essentials: Part 6." https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/, Acccessed: 07 June 2022, 2015.

[23] N. Q. Uy and V. H. Nam, "A comparison of amqp and mqtt protocols for internet of things," in *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, pp. 292–297, 2019.

[24] Home Nursing Group, "The importance of using a mobility aid." https://homenursinggroup.com.au/about-us/news/the-importance-of-using-a-mobility-aid/, Acccessed: 14 November 2022, 2022.

[25] Physiopedia, "Walking aids." https://www.physio-pedia.com/Walking_Aids, Acccessed: 14 November 2022, 2022.

[26] P. Martim, V. L. Frango, O. Postolache, and Y. Yang, "Smart object for physical rehabilitation assessment," in *2018 International Conference and Exposition on Electrical And Power Engineering (EPE)*, pp. 0678–0682, 2018.

[27] C. Nave, Y. Yang, V. Viegas, and O. Postolache, "Physical rehabilitation based on smart walker," in *2018 12th International Conference on Sensing Technology (ICST)*, pp. 388–393, 2018.

[28] G. Yuk, Y. Y. Chang, H. Park, B. J. Dan, and W. S. Chang, "Smart walker development based on experts' feedback for elderly and disabled," in *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 552–553, 2013.

[29] J. Mercado, G. Chu, E. J. Imperial, K. G. Monje, R. M. Pabustan, and A. Silverio, "Smart cane: Instrumentation of a quad cane with audio-feedback monitoring system for partial weight-bearing support," in *2014 IEEE International Symposium on Bioelectronics and Bioinformatics (IEEE ISBB 2014)*, pp. 1–4, 2014.

[30] Amazon Web Services, "Aws identity and access management (iam) features." https://aws.amazon.com/iam/features/?nc=sn&loc=2, Acccessed: 16 July 2022, 2022.

[31] Amazon Web Services, "Aws iot core features." https://aws.amazon.com/iot-core/features/?nc=sn&loc=3, Acccessed: 16 July 2022, 2022.

[32] Ian Craggs, "Understanding the differences between mqtt and websockets for iot." https://www.hivemq.com/blog/understanding-the-differences-between-mqtt-and-websockets-for-iot/, Acccessed: 26 November 2022, 2022.

[33] Amazon Web Services, "Aws lambda features." https://aws.amazon.com/lambda/features/, Acccessed: 16 July 2022, 2022.

[34] Amazon Web Services, "Aws s3 features." https://aws.amazon.com/s3/features/?nc=sn&loc=2, Acccessed: 16 July 2022, 2022.

[35] Amazon Web Services, "Aws cognito features." https://aws.amazon.com/cognito/details/, Acccessed: 16 July 2022, 2022.

[36] LastMinuteEngineers.com, “What is rfid? how it works? interface rc522 rfid module with arduino.” https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/, Acccessed: 16 July 2022, 2022.

[37] OMEGA Engineering, “Load cells and force sensors.” https://www.omega.com/en-us/resources/load-cells, Acccessed: 18 July 2022, 2020.

[38] Alibaba, “Snc2c6 cheap tension compression load cell.” https://www.alibaba.com/product-detail/SNC2C6-cheap-tension-compression-load-cell_60289554523.html, Acccessed: 18 July 2022, 2022.

[39] Burr-Brown Corporation, “Ina122.” https://www.ti.com/lit/ds/symlink/ina122.pdf, Acccessed: 31 August 2022, 1997.

[40] Tekscan Inc., “Flexiforce™ standard model a201.” https://www.tekscan.com/sites/default/files/FLX-Datasheet-A201-RevI.pdf, Acccessed: 18 July 2022, 2021.

[41] Tekscan Inc., “Flexiforce a201 sensor.” https://www.tekscan.com/products-solutions/force-sensors/a201?tab=specifications-performance, Acccessed: 18 July 2022, 2022.

[42] Mixtrónica, “Altimu-10 - v4 gyro accelerometer compass and altimeter.” https://mixtronica.com/giroscopio-acelerometros-sensores/7387-altimu-10-v4-gyro-accelerometer-compass-and-altimeter.html, Acccessed: 18 July 2022, 2022.

[43] V. M. Monica and K. G. J. Nigel, “Object tracking based on kalman filter and gait feature extraction,” in *2017 International Conference on Inventive Systems and Control (ICISC)*, pp. 1–5, 2017.

[44] Yida, “Uart vs i2c vs spi – communication protocols and uses.” https://www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-vs-spi-communication-protocols-and-uses/, Acccessed: 18 July 2022, 2019.

[45] Warees M. Warees; Lisa Clayton; Matthew Slane, “Crutches.” https://www.ncbi.nlm.nih.gov/books/NBK539724/, Acccessed: 7 November 2022, 2022.

[46] Sparkfun, “Pcb basics.” https://learn.sparkfun.com/tutorials/pcb-basics/all, Acccessed: 7 November 2022, 2022.

[47] Arduino, “Arduino sketches.” https://docs.arduino.cc/learn/programming/sketches, Acccessed: 26 November 2022, 2022.

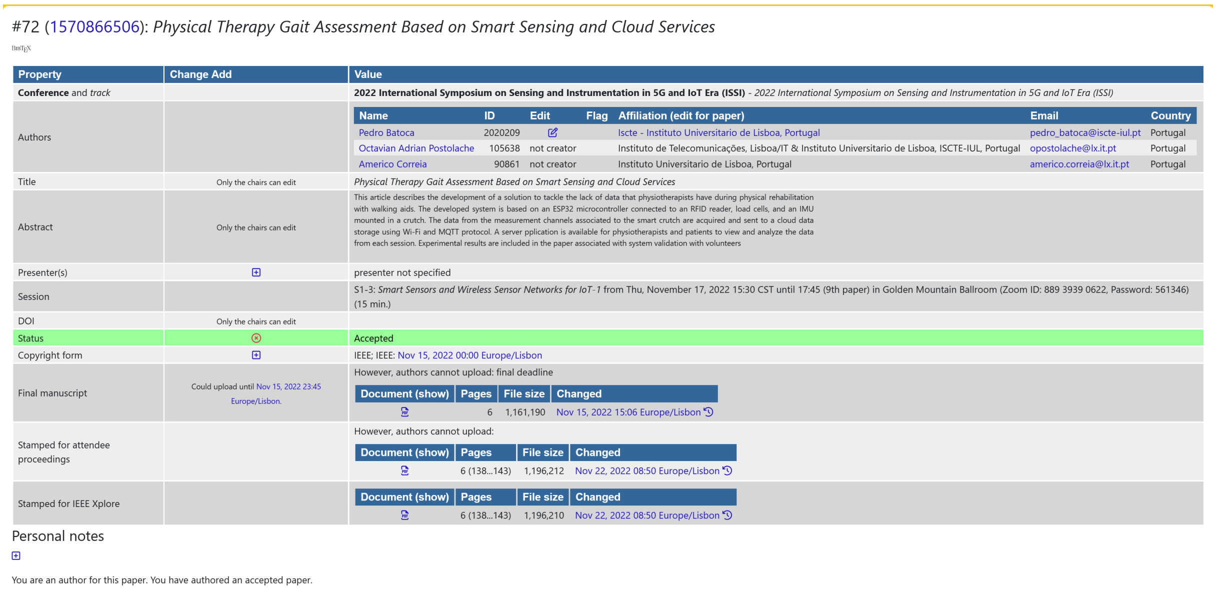
[48] Meghan Rimol, “Gartner forecasts worldwide public cloud end-user spending to reach nearly $500 billion in 2022.” https://www.gartner.com/en/newsroom/press-releases/2022-04-19-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-500-billion-in-2022, Acccessed: 26 November 2022, 2022.

[49] Shanika Wickramasinghe, “Aws vs azure vs gcp: Comparing the big 3 cloud platforms.” https://www.bmc.com/blogs/aws-vs-azure-vs-google-cloud-platforms/, Acccessed: 26 November 2022, 2021.

CHAPTER 8

Annex A - Article

The article below was accepted and presented at the 3rd International Symposium on Sensing and Instrumentation in 5G & IoT Era, ISSI 2022, which took place between the 17th and 18th of November 2022 in Shanghai, China.



#72 (1570866506): *Physical Therapy Gait Assessment Based on Smart Sensing and Cloud Services*

BibTeX

<table>
<tr><th>Property</th><th>Change Add</th><th>Value</th></tr>
<tr><td><b>Conference</b> and <i>track</i></td><td></td><td><b>2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI)</b> - <i>2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI)</i></td></tr>
<tr><td>Authors</td><td></td><td><table>
<tr><th>Name</th><th>ID</th><th>Edit</th><th>Flag</th><th>Affiliation (edit for paper)</th><th>Email</th><th>Country</th></tr>
<tr><td>Pedro Batoca</td><td>2020209</td><td>✎</td><td></td><td>Iscte - Instituto Universitario de Lisboa, Portugal</td><td>pedro_batoca@iscte-iul.pt</td><td>Portugal</td></tr>
<tr><td>Octavian Adrian Postolache</td><td>105638</td><td>not creator</td><td></td><td>Instituto de Telecomunicações, Lisboa/IT &amp; Instituto Universitario de Lisboa, ISCTE-IUL, Portugal</td><td>opostolache@lx.it.pt</td><td>Portugal</td></tr>
<tr><td>Americo Correia</td><td>90861</td><td>not creator</td><td></td><td>Instituto Universitario de Lisboa, Portugal</td><td>americo.correia@lx.it.pt</td><td>Portugal</td></tr>
</table></td></tr>
<tr><td>Title</td><td>Only the chairs can edit</td><td><i>Physical Therapy Gait Assessment Based on Smart Sensing and Cloud Services</i></td></tr>
<tr><td>Abstract</td><td>Only the chairs can edit</td><td>This article describes the development of a solution to tackle the lack of data that physiotherapists have during physical rehabilitation with walking aids. The developed system is based on an ESP32 microcontroller connected to an RFID reader, load cells, and an IMU mounted in a crutch. The data from the measurement channels associated to the smart crutch are acquired and sent to a cloud data storage using Wi-Fi and MQTT protocol. A server pplication is available for physiotherapists and patients to view and analyze the data from each session. Experimental results are included in the paper associated with system validation with volunteers</td></tr>
<tr><td>Presenter(s)</td><td>⊞</td><td>presenter not specified</td></tr>
<tr><td>Session</td><td></td><td>S1-3: <i>Smart Sensors and Wireless Sensor Networks for IoT-1</i> from Thu, November 17, 2022 15:30 CST until 17:45 (9th paper) in Golden Mountain Ballroom (Zoom ID: 889 3939 0622, Password: 561346) (15 min.)</td></tr>
<tr><td>DOI</td><td>Only the chairs can edit</td><td></td></tr>
<tr><td>Status</td><td>⊗</td><td>Accepted</td></tr>
<tr><td>Copyright form</td><td>⊞</td><td>IEEE; IEEE: Nov 15, 2022 00:00 Europe/Lisbon</td></tr>
<tr><td>Final manuscript</td><td>Could upload until Nov 15, 2022 23:45 Europe/Lisbon.</td><td>However, authors cannot upload: final deadline<table>
<tr><th>Document (show)</th><th>Pages</th><th>File size</th><th>Changed</th></tr>
<tr><td>PDF</td><td>6</td><td>1,161,190</td><td>Nov 15, 2022 15:06 Europe/Lisbon</td></tr>
</table></td></tr>
<tr><td>Stamped for attendee proceedings</td><td></td><td>However, authors cannot upload:<table>
<tr><th>Document (show)</th><th>Pages</th><th>File size</th><th>Changed</th></tr>
<tr><td>PDF</td><td>6 (138...143)</td><td>1,196,212</td><td>Nov 22, 2022 08:50 Europe/Lisbon</td></tr>
</table></td></tr>
<tr><td>Stamped for IEEE Xplore</td><td></td><td><table>
<tr><th>Document (show)</th><th>Pages</th><th>File size</th><th>Changed</th></tr>
<tr><td>PDF</td><td>6 (138...143)</td><td>1,196,210</td><td>Nov 22, 2022 08:50 Europe/Lisbon</td></tr>
</table></td></tr>
</table>

Personal notes

⊞

You are an author for this paper. You have authored an accepted paper.

FIGURE 1. Edas confirmation of accepted paper

Physical Therapy Gait Assessment based on Smart Sensing and Cloud Services

Pedro Batoca
Instituto de Telecomunicações
ISCTE-Instituto Universitário de Lisboa
Lisboa, Portugal
pedro_batoca@iscte–iul.pt

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

Américo Correia
Instituto de Telecomunicações
ISCTE-Instituto Universitário de Lisboa
Lisboa, Portugal
americo.correia@iscte-iul.pt

***Abstract*—This article describes the development of a solution to tackle the lack of data that physiotherapists have during physical rehabilitation with walking aids. The developed system is based on an ESP32 microcontroller connected to an RFID reader, load cells, and an IMU mounted in a crutch. The data from the measurement channels associated with the smart crutch are acquired and sent to a cloud data storage using Wi-Fi and MQTT protocol. A server application is available for physiotherapists and patients to view and analyze the data from each session. Experimental results are included in the paper associated with system validation with volunteers.**

***Keywords*—Internet of Things, Internet of Medical Things, Smart Sensors, Physical Rehabilitation, Web Application, Cloud**

I. Introduction

Physiotherapy is a science dedicated to improving a patient's physical condition that can be related to a physical or nervous issue [1]. This need can come due to natural aging, accidents, events such as AVCs, degenerative diseases, or others [1]. In this dissertation, the focus is physical rehabilitation through periodic rehabilitation sessions.

According to a 2020 study from the United Nations, the number of people aged 65 or older surpasses 727 million worldwide and estimates that by 2050, this number will be 1.5 billion [2]. World Health Organization report on falls prevention in older age also points out that with natural aging, the likelihood of falls by elderly people also increases, and usually when it happens, there is a need for physical rehabilitation [3]. The number of people living with chronic disease also keeps increasing, making this a mandatory topic [4]. Estimates indicate that one in three people would benefit from rehabilitation however it isn't available to everyone [4]. According to WHO in low and middle-income countries, half of the population doesn't have access to it [4]. To address this need, WHO created an initiative called Rehabilitation 2030 to challenge stakeholders worldwide to expand the funding for health systems, improve research to develop new solutions, and expand the availability of these services to the population worldwide [4].

The always-increasing number of patients is not the only challenge that physiotherapists face. There is also the lack of a system that allows an individual to use a crutch or other type of aid and automatically obtain, save and analyze the data generated by the device used [5]. This information will allow physiotherapists to improve their decisions regarding the treatment, which can lead to a faster recovery for the patient.

The goal of this article is to create a platform capable of helping patients track their sessions, allowing a physiotherapist to observe and possibly evaluate the patient progress, and include training plans for the patients.

In [6], is described another system that helps to tackle this need. It has three main components a crutch, a mobile application, and a remote database. The crutch features an Arduino Nano and several sensors attached to it, such as IMU, load cell, gait, and Bluetooth. The data generated allows computing the metrics of orientation and balance of the patient and the force applied in the crutch. The information is shared with the mobile application via Bluetooth which, then stores it in a remote database for redundancy. The application allows to view the data in real-time as well as to create and view physiotherapists, view or start sessions, and track the overall progress of the patient. As mentioned, improvements can be made by allowing more types of equipment or the use of two crutches simultaneously as well as providing training plans and exercises in the application to allow the patient to train without going to the clinic.

In [7] is presented a system with four main components, an RFID module, a walker, a web application, and a mobile application. The module registers the RFID from the patient's card into the database. The walker has an Arduino Mega and several sensors attached, including an IMU, WI-FI, Bluetooth, ultrasounds, and load cells. The data generated allows calculating the metrics of orientation and balance of the patient, bilateral elevation of the walker, number of steps, and force exercise in the walker's feet. Both web and mobile applications offer the possibility to create and view the patients, the treatment plan and search past sessions of each patient. The real-time view of the session and the patient's progression is only available in the mobile application. The improvements suggested are using MQTT as the communication protocol and replacing the RFID sensor with a fingerprint one.

II. System Description

The goal of this system is to provide data to physiotherapists so that they can objectively analyze the patient's condition, track their progress, and ultimately help accelerate the treatment. The high-level architecture used is shown below:

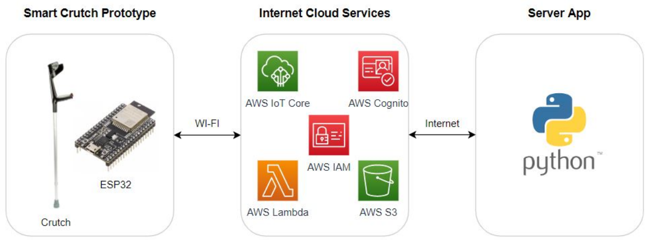


Fig. 1. Physical Therapy Gait Assessment: High Level Architecture

As seen in Fig. 1, this system can be divided into three main blocks. The first is the smart crutch prototype, the second is the cloud services, and the final is the server app. The smart crutch prototype includes a microcontroller and a set of sensors. These sensors extract dynamic information about force and accelerations that allow the calculation of specific metrics. The data is necessary for patient gait analysis and is aggregated by the microcontroller and then sent to the cloud services via WI-FI to be processed.

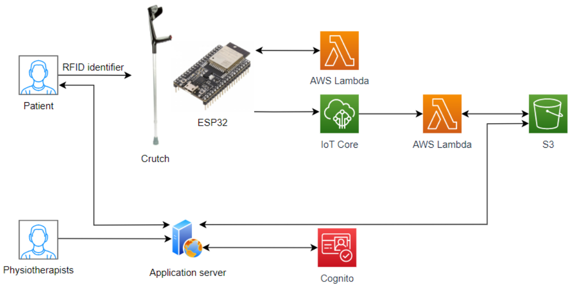


Fig. 2. System flow diagram

The flow diagram in Fig. 2 allows to better understand how the system works and how to start a new session. Firstly, the patient has to approach his RFID tag or card to the RFID reader embedded in the crutch, and then the system makes a call to a lambda function to fetch the interval to use between measurements. Afterward, the session starts and the patient does the exercises. The data is sent using the MQTT protocol to the IoT Core service that triggers another lambda to process the data and store it on an S3 bucket. To finish the session, the patient has to approach the RFID identifier to the crutch. The metrics are then extracted from the raw data by a third lambda that will be detailed in the software chapter. Patients and physiotherapists can access the session data by logging in to the application server with their credentials. The user profiles are automatically assigned when logged in and can perform the actions described in the previous section.

A. Hardware

This prototype leverages several sensors connected to a microcontroller that acquires signals necessary for metrics calculation. The sensing part is composed of an RFID RC522, a FlexiForce A201, a load cell SNC2C6, and the IMU AltiIMU 10 v-4. To aggregate the generated data and send it to the cloud, the ESP32 was the microcontroller chosen. The RFID is used to identify the patient and has two components, a transponder or tag inside an object to be identified and a transceiver or reader [8]. The RC522 module was used due to its low cost and compatibility. This module generates a 13.56MHz electromagnetic field used to communicate with the RFID tags (ISO 14443A standard tags), has a read range of 5 centimeters, and is compatible with the interfaces SPI, I2C, and UART [8]. To measure the force applied to the crutch a load cell is used. These sensors convert a mechanical force into an electric signal (voltage) and usually use a Wheatstone bridge as a conditioning circuit [9]. The SNC2C6 load cell sensor was the choice for this prototype because it offers accurate measurement and a range of up to 500N, 1000N, or even 1500N [10]. For this prototype, the load cell had 500N of range since the group of volunteers had a known BMI, and the results didn't show any signs of saturation during the performed tests. In case of need, the load cell can be easily replaced. However, it has a very low sensibility of only 0.8 - 2.0 mV/V, thus providing a very low output signal voltage. To tackle this the sensor is coupled with an INA122, a precision instrumentation amplifier for accurate, low noise differential signal acquisition that helps to provide a finer output voltage [11]. The gain of the INA122 is calculated by the equation $G = 5 + 200\ k\Omega / R$. The resistance used had 1 kΩ so the gain provided is 205.

The FlexiForce A201 is a thin, flexible, and non-intrusive piezoresistive force sensor that is ideal for assembling proof of concepts [17]. It can measure forces up to 445 N or 45 Kg in a low response time of less than 5μsec while providing the durability of over 3 million actuations [12]. This sensor was used to measure the force applied to the hand support of the crutch. The Pololu AltIMU-10 v4 is an inertial measurement unit (IMU) board that combines an LPS25H digital barometer, L3GD20H 3-axis gyroscope, and LSM303D 3-axis accelerometer and 3-axis magnetometer thus providing this sensor a 10 degree of freedom (10DOF) [13]. It will be responsible for providing the data used to calculate the orientation and steps of the patients. This sensor uses I2C to communicate with reads to the microcontroller and each read for the gyro, accelerometer, and magnetometer uses 16 bits [13]. It features a low-dropout linear voltage regulator that allows it to be powered from 2.5 to 5.5V when it only needs 3.3V [13].

The ESP32 was the choice for this prototype due to its low cost, energy consumption, interface compatibility, and available libraries for scripting development. In terms of specs, this microcontroller packs a Tensilica Xtensa LX6 dual-core processor in 40nm technology, a Bluetooth and

WiFi interface supports up to four 16-MB of external QSPI flash and SRAM and, the crystal oscillator is limited to 40 MHz due to firmware [14]. In terms of interfaces, it provides compatibility with several interfaces such as SPI (serial peripheral interface), CAN, and I2C [19]. It is suited for all kinds of environments and has an operating voltage between 2.7 and 3.6V [14].

When it comes to the interfaces used for the communication between the ESP32 and the sensors, the RFID uses SPI, the IMU I2C, and both FlexiForce and LoadCell use analog pins. SPI uses the master-slave architecture where communication is full-duplex, and connections are done by MOSI (Master Output Slave Input) and MISO (Master Input Slave Output). In turn, I2C is half-duplex and is implemented with SDA (Serial Data) and SCL (Serial Clock) [15]. The RFID uses SPI, while the IMU uses an I2C bus. The load cell and the FlexiForce are digitized using the ESP32 onboard analog-to-digital converter (ADC).

The sensors are connected to the ESP32 as it is presented in Fig. 3.

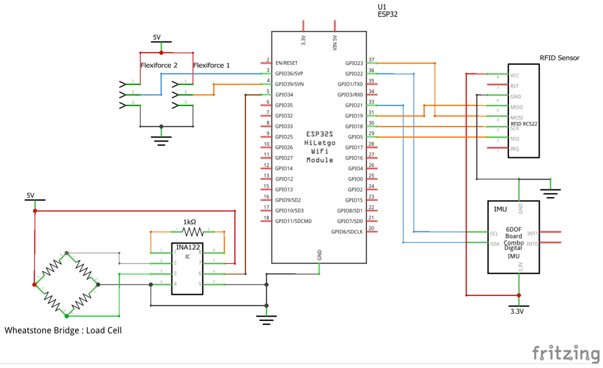


Fig. 3. Schematic of the sensing apparatus

B. Software

The architecture of the system can be seen below:

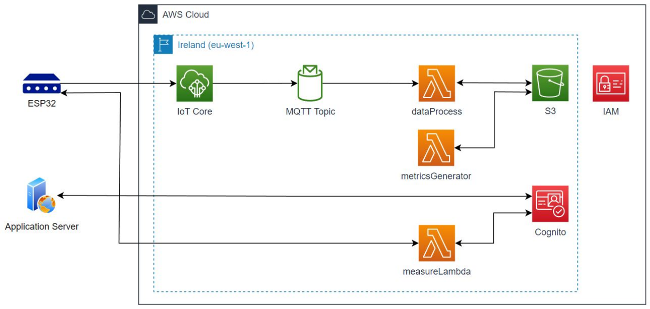


Fig. 4. System Software Architecture (S3 refers to the AWS Simple Storage Service, the MQTT topic is the string used by the broker to forward the messages, IAM is the AWS Identity and Access Management service, and AWS Cognito is the service responsible for the user management)

As mentioned in the system overview, this solution uses the ESP32 as its microcontroller, a python application server, and leverages several cloud services in its architecture. The ESP32 runs a C program responsible for establishing the connection, extracting, and sending sensor data to the cloud. Firstly the microcontroller starts by initiating the sensing part and establishing a connection to the IoT Core service from AWS. Afterward, it requests the measurement interval to a lambda (cloud function) via HTTP Get request to be used in the session and prompts the user to pass the RFID tag. Once the user approaches the tag, the session starts, and the measurements are sent to IoT Core through MQTT until the user passes the identifier again to terminate the session.

MQTT leverages a publish-subscribe architecture and is meant to be used by small, low power, low cost, and low memory devices in high latency, low-bandwidth, or unreliable networks. MQTT is based on TCP/IP and has three main components [16] [17] [18]:

1) The MQTT Broker is a server that receives the messages from the publisher and forwards them to the appropriate subscribers by checking the topic subscribed to.
2) The publisher is the one who sends the message (data to be shared) and the topic (the string that the MQTT broker uses to route the messages to the appropriate subscribers).
3) The subscriber receives messages from the MQTT Broker related to the subscribed topic.

On the other hand, HTTP is an application-level protocol that implements a client-server architecture and uses the TCP protocol. It is unidirectional and synchronous and provides methods such as GET, POST, PUT, and DELETE [16] [17].

The messages present in Fig. 5 and explained below are used to transmit the data that was acquired by ESP32.

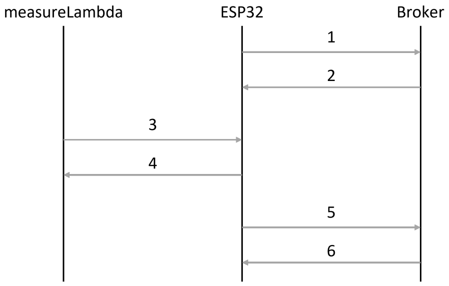


Fig. 5. Messages between microcontroller and server

Message 1) is an MQTT Connect message. 2) is the respective Connack 3) is an HTTP Get message that triggers the measureLambda function, and 4) is a message with the measuring interval defined for the patient. 5) is an MQTT Publish message where the RFID tag is sent and marks the start of the session and 6) is another Publish message with the data acquired by the sensors. This last message repeats itself every time that the measurement interval is passed. When an RFID tag or card is presented again, a final message is sent to end the session.

The cloud services are responsible for the MQTT broker, user management, permission management, data processing, and storage. Amazon Web Service (AWS) Identity and Access Management is the service responsible for the permission management for AWS services and resources [19]. AWS IoT Core provides connectivity, scalability, and management for IoT devices easily and reliably without physical servers [20]. It supports MQTT, HTTPS, MQTT over WSS, and LoRaWAN as communication protocols [20].AWS Lambda is a serverless cloud computing service that runs code responding to an event [21]. AWS Simple Storage Service or S3 is a cloud object storage service that provides high availability and redundancy storage at a competitive price [22]. AWS Cognito is a cloud-native user authentication and management service that can be integrated into custom applications simply [23]. The last block is the server app built in Python that provides access to the session data for the patients and physiotherapists as well as training plans and administrative tasks for physiotherapists and administrators.

Once the data reaches the cloud, the dataProcess lambda function is triggered. This function starts by evaluating if the incoming message is related to the start/end of a session or to recording a new measurement. If it is to start a new session, it stores the data related to it in an s3 bucket by creating a new folder in the user's path and a file inside with the date, RFID tag, start time, and device used. If a new measurement arrives, the function starts by locating the session and then adds the data related to each sensor to a measurements file inside the session folder. When the session ends, lambda calculates the session duration, adds the end time, and updates the information file. Afterward, the metricsGenerator lambda initiates. This function aggregates the raw data into averages for each sensor by the minute and stores them in the bucket. The app server leverages the flask framework and the boto3 library to interact with the AWS services. The app has three different types of users according to their respective roles. Patients are allowed to view and change their profiles and view past sessions and training plans. Physiotherapists can create new patients and view the patient list, training plans, and session data. The other type of user is the administrator that has the function of managing the physiotherapists. These roles are mapped with the roles assigned in AWS Cognito.

III. Results

This system was tested by two volunteers, a male and a female aged 24 and 25. All the executed tests had a duration of 30 seconds. The first one was to walk an imposed distance, the second was a balance exercise which consists of staying still, and the last one was also a balance exercise however, the goal was to simulate a patient that needs to rely on a crutch to stand still. Alongside the force applied on the bottom and the hand support of the crutch, the Euler angles were also retrieved. These angles are based on the measurements from the accelerometer, magnetometer, and gyroscope and provide reliable information about the rotation of the crutch during the exercises, mainly if the patient gave a step or inverted the marching way. All of this data was recorded by the crutch and sent to the cloud for storage, and Microsoft Excel was used to generate the following graphs.

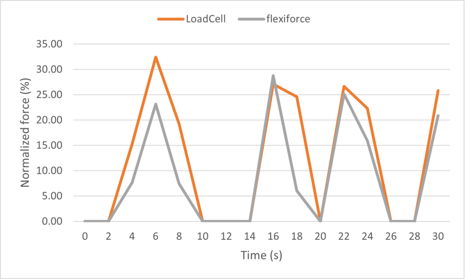


Fig. 6. Force evolution during walking for the female volunteer

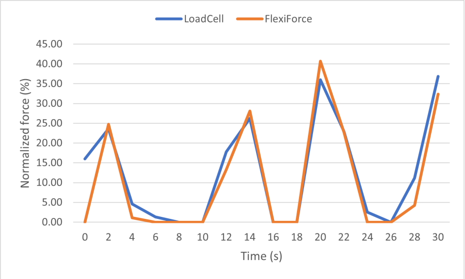


Fig. 7. Force evolution during walking for the male volunteer

Fig. 6, shows the force induced on the sensors present in the crutch as a function of the duration of the exercise for the female patient, and Fig. 7 shows the same information for the male patient. The force is normalized to the maximum sensor value and is presented as a percentile value. The different sensors, loadcell, and flexiforce show agreeable results for both cases. When the patient is walking, the sensor in the crutch handle and the foot of the crutch shows an increase in the force applied. Comparing both figures it is possible to see that the male volunteer applied a homogenous force across the crutch and, the female made more force in the load cell. It is also possible to see that the male patient exercised a higher strength overall.

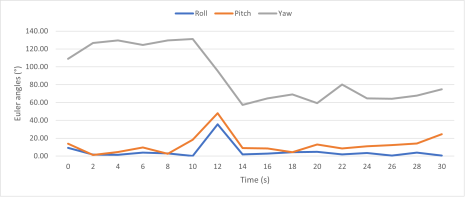


Fig. 8. Roll and pitch evolution during walking for the female volunteer

Fig. 8, shows the roll and pitch angles evolution when the patient is walking for the female patient, and Fig. 9 shows the same information for the male patient. The roll and pitch angle values are in degrees and were calculated by the ESP32. It is possible to see the variations with the pitch angle that translates into the steps of the patient and the roll angle

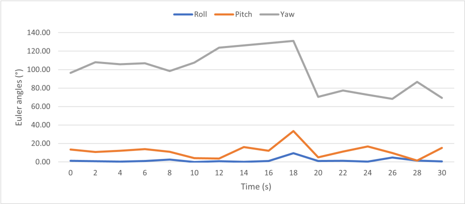


Fig. 9. Roll and pitch evolution during walking for the male volunteer

that provide information about the balance. Comparing both figures it is possible to see that the male volunteer was more balanced than the female one and both gave approximately the same number of steps.

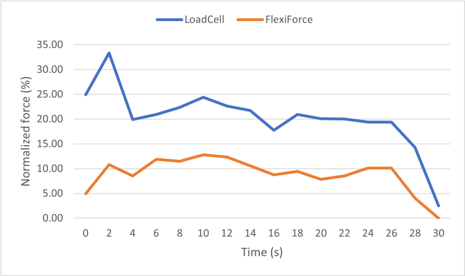


Fig. 10. Force evolution during the balancing exercise for the male volunteer

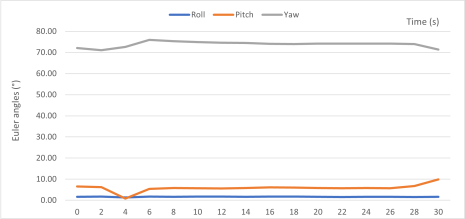


Fig. 11. Roll and pitch evolution during the balancing exercise for the male volunteer

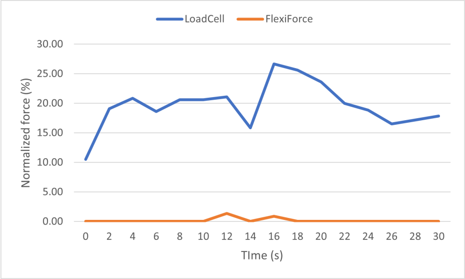


Fig. 12. Force evolution during the balancing exercise for the female volunteer

Fig. 10, shows the force induced on the sensors present in the crutch as a function of the duration of the balancing exercise for the male patient, and Fig. 11 shows the roll and pitch angles evolution for the same exercise. The results are accordingly expected because it is possible to see little variations of the pitch and roll angles over time meaning that the volunteer is balanced. Also, it is possible to conclude that the force applied to the hand support shows that the patient does not need a crutch.

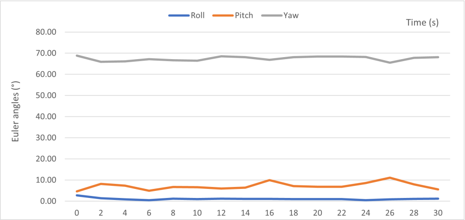


Fig. 13. Roll and pitch evolution during the balancing exercise for the female volunteer

Fig. 12 and Fig. 13 show the exercise results for the female patient and it is possible to take the same conclusions, the force applied on the hand supports constant at 0 overtime showing that this volunteer is also healthy. The main difference between the results is that pitch variation was higher.

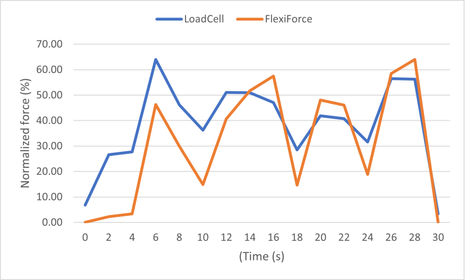


Fig. 14. Force evolution during the balancing simulation exercise for the male volunteer

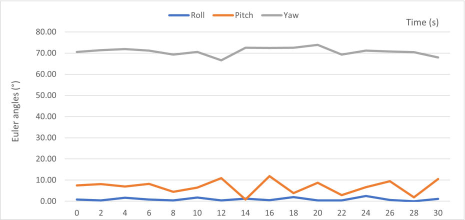


Fig. 15. Roll and pitch evolution during the balancing simulation exercise for the male volunteer

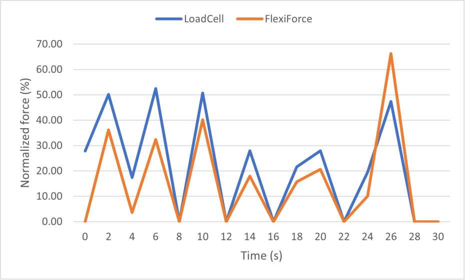


Fig. 16. Force evolution during the balancing simulation exercise for the female volunteer

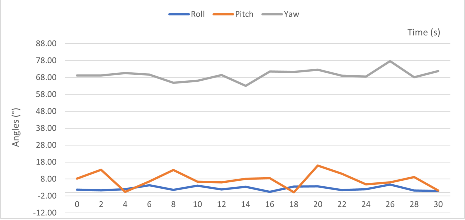


Fig. 17. Roll and pitch evolution during the balancing simulation exercise for the female volunteer

Fig. 14, shows the force induced on the sensors present in the crutch as a function of the duration of the balancing simulation exercise for the male patient, and Fig. 15 shows the roll and pitch angles evolution for the same exercise. The results confirm the expectation of seeing a significantly higher force applied on both ends of the crutch to support the patient's weight. There are also greater variations of the roll and pitch angle related to the decrease in the patient's balance ability.

Fig. 16 and Fig. 17 show the exercise results for the female patient and validate the conclusions taken from the male patient analysis. When a patient needs physical support, both the force applied on the crutch and the need for balance will be significantly higher. This need for balance translates to a greater variation in the roll and pitch angles.

IV. Conclusions

A smart crutch as a walking aid solution for physical rehabilitation is proposed in the paper. The developed prototype uses off-the-shelf components such as crutches and sensors it is a relatively cheap build. This solution presents functionalities related to the objective evaluation of gait rehabilitation. This diagnosis can be based on the data provided by the system about the force applied at the bottom and hand grip of the crutch as well as, imu information like pitch, roll, and yaw angles. In addition to the crutch, there is also a web application where physiotherapists can view all of this information and define exercise plans to improve the patient's condition faster.

In terms of future work, this can be the development of new metrics related to the patient training outcome as well as the development of machine learning models capable to identify patients that need walking aids for their safety and gait rehabilitation.

V. Acknowledgment

This research is supported by Instituto de Telecomunicações (IT) and ISCTE-Instituto Universtiário de Lisboa.

References

[1] NHS, "Physiotherapy." https://www.nhs.uk/conditions/physiotherapy/ Acccessed: 07 June 2022, nov 2018.

[2] United Nations Department of Economic and Social Affairs, Population Division, "World population ageing 2020 highlights: Living arrangements of older persons." https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/undesa_pd-2020_world_population_ageing_highlights.pdf, Acccessed: 07 June 2022, 2020.

[3] World Health Organization, "Who global report on falls prevention in older age." https://extranet.who.int/agefriendlyworld/wp-content/uploads/2014/06/WHo-Global-report-on-falls-prevention-in-older-age.pdf, Acccessed: 07 June 2022, 2007.

[4] World Health Organization, "Rehabilitation 2030." https://www.who.int/initiatives/rehabilitation-2030, Acccessed: 07 June 2022, 2017.

[5] S. Vishnu, S. J. Ramson, and R. Jegan, "Internet of medical things (iomt) - an overview," in *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*, pp. 101–104, 2020.

[6] P. Martim, V. L. Frango, O. Postolache, and Y. Yang, "Smart object for physical rehabilitation assessment," in *2018 International Conference and Exposition on Electrical And Power Engineering (EPE)*, pp. 0678–0682, 2018.

[7] C. M. A. R. d. A. Nave, "Iophyr - physical rehabilitation iot system: sistema de reabilitação motora baseado em andarilhos inteligentes e iot.," 2018.

[8] LastMinuteEngineers.com, "What is rfid? how it works? interface rc522 rfid module with arduino." https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/, Acccessed: 16 July 2022, 2022.

[9] OMEGA Engineering, "Load cells and force sensors." https://www.omega.com/en-us/resources/load-cells, Acccessed: 18 July 2022, 2020.

[10] Alibaba, "Snc2c6 cheap tension compression load cell." https://www.alibaba.com/product-detail/SNC2C6-cheap-tension-compression-load-cell_60289554523.html, Acccessed: 18 July 2022, 2022.

[11] Burr-Brown Corporation, "Ina122." https://www.ti.com/lit/ds/symlink/ina122.pdf, Acccessed: 31 August 2022, 1997.

[12] Tekscan Inc., "Flexiforce™ standard model a201." https://www.tekscan.com/sites/default/files/FLX-Datasheet-A201-RevI.pdf, Acccessed: 18 July 2022, 2021.

[13] Mixtrónica, "Altimu-10 - v4 gyro accelerometer compass and altimeter." https://mixtronica.com/giroscopio-acelerometros-sensores/7387-altimu-10-v4-gyro-accelerometer-compass-and-altimeter.html, Acccessed: 18 July 2022, 2022.

[14] Espressif Systems, "Esp32 wroom32 datasheet." https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf, Acccessed: 18 July 2022, 2022.

[15] F. Leens, "An introduction to i2c and spi protocols," *IEEE Instrumentation & Measurement Magazine*, vol. 12, no. 1, pp. 8–13, 2009.

[16] N. Naik, "Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http," in *2017 IEEE International Systems Engineering Symposium (ISSE)*, pp. 1–7, 2017.

[17] M. Bansal and Priya, "Application layer protocols for internet of healthcare things (ioht)," in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, pp. 369–376, 2020.

[18] N. Q. Uy and V. H. Nam, "A comparison of amqp and mqtt protocols for internet of things," in *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, pp. 292–297, 2019.

[19] Amazon Web Services, "Aws identity and access management (iam) features." https://aws.amazon.com/iam/features/?nc=sn&loc=2, Acccessed: 16 July 2022, 2022.

[20] Amazon Web Services, "Aws iot core features." https://aws.amazon.com/iot-core/features/?nc=sn&loc=3, Acccessed: 16 July 2022, 2022.

[21] Amazon Web Services, "Aws lambda features." https://aws.amazon.com/lambda/features/, Acccessed: 16 July 2022, 2022.

[22] Amazon Web Services, "Aws s3 features." https://aws.amazon.com/s3/features/?nc=sn&loc=2, Acccessed: 16 July 2022, 2022.

[23] Amazon Web Services, "Aws cognito features." https://aws.amazon.com/cognito/details/, Acccessed: 16 July 2022, 2022.