POLITECNICO DI TORINO

Academic Year 2019-2020

# Interdisciplinary Project Report

# RFED

**Team**:

Anastasiya Isgandarova

Cansu Ilter

Lorenzo Bellone

Victor de Castro Morini

# Contents

**1 Introduction** **5**
1.1 Context Analysis . . . . . 5
1.2 Project Objective . . . . . 5
1.3 Related Works . . . . . 6

**2 Proposed System** **8**
2.1 Communication Protocols . . . . . 9
2.2 Hardware . . . . . 10
2.2.1 Global Tag BLE Development Kit . . . . . 10
2.2.2 ESP32-DevKitC . . . . . 10
2.2.3 Raspberry Pi 3B+ . . . . . 11
2.3 Software . . . . . 11
2.3.1 Database . . . . . 11
2.3.2 ESP32 Development Framework . . . . . 12
2.4 Security . . . . . 14
2.5 Project Implementation . . . . . 15
2.5.1 Beacons Detection Strategy . . . . . 16
2.5.2 ESP32 Wi-Fi MESH MQTT Implementation . . . . . 19
2.5.3 Database Structure and Strategies . . . . . 20
2.5.4 Prototype Packaging . . . . . 22

**3 Experimental Results** **24**
3.1 Scenario 1 . . . . . 25
3.2 Scenario 2 . . . . . 26
3.3 Scenario 3 . . . . . 28

**4 Business Model** **30**

**5 Conclusion and Future Works** **31**

# List of Figures


1 RFED concept . . . 9
2 ESP-MESH Software Stack. Ref: Espressif . . . 12
3 ESP-MESH Network Architecture. Modified from: Espressif . . . 13
4 Espressif Mesh Development Framework (ESP-MDF) stack with highlighted MESH Wi-Fi (Mwifi) Application Programming Interface (API), the main driver of RFED. Modified from: Espressif Github . . . 14
5 The graphical representation of the communication network. . . . 15
6 Shifting Flow . . . 17
7 Detection of Patient Arrival and Patient Departure . . . 18
8 Beacon Strategy Flow . . . 18
9 Simplified flowchart of the Wi-Fi, MQTT and MESH implementation code, flashed in the ESP32 devices. . . . 19
10 ESP32 Menu . . . 20
11 Web application forms to insert a new patient and to remove a leaving one. . . . 21
12 Flowchart representing the strategy applied by the subscriber to update the database when a new message is received from the ESP mesh network. . . . 22
13 RFED ESP32 development board box model . . . 23
14 RFED Raspberry Pi (RPi) box model . . . 23
15 Highlighted PCB and IPEX antenna. Two different models of ESP32. Modified from: Espressif . . . 24
16 Graphical representation of the environment in which the project has been tested. 24
17 Real durations (green bars) compared to measured ones (red bars) for each phase and patient of the first scenario. . . . 26
18 Real durations (green bars) compared to measured ones (red bars) for each phase and patient of the second scenario. . . . 27
19 Real durations (green bars) compared to measured ones (red bars) for each phase and patient of the third scenario. . . . 29
20 Canvas Business Model . . . 30
21 Bill of Materials . . . 33


# List of Tables


1 Benchmarking Results . . . 8

2 Data discovery, defence and detection requirements demanded by modern privacy regulations and implemented by MongoDB. . . . . 16
3 Numerical results from scenario 1 . . . . . 25
4 Numerical results from scenario 2 . . . . . 28
5 Numerical results from scenario 3 . . . . . 30
6 The comparison between proposed and delivered project . . . . . 32

# Acronyms

**AP** Access Point.

**API** Application Programming Interface.

**BLE** Bluetooth Low Energy.

**BOM** Bill of Materials.

**ED** Emergency Department.

**ESP-IDF** Espressif IoT Development Framework.

**ESP-MDF** Espressif Mesh Development Framework.

**GDPR** General Data Protection Regulation.

**GUI** Graphical User Interface.

**IEEE** Institute of Electrical and Electronics Engineers.

**IP** Internet Protocol.

**IRTLS** Indoor Real-time locating systems.

**JSON** JavaScript Object Notation.

**MAC** Medium Access Control.

**MCU** Microcontroller.

**MQTT** Message Queue Telemetry Transport.

**Mwifi** MESH Wi-Fi.

**NoSQL** Not Only Structured Query Language.

**OS** Operational System.

**OTA** Over the Air.

**PCB** Printed Circuit Board.

**PoE** Power over Ethernet.

**RDBMS** Relational Database Management System.

**RESTful** Representational State Transfer.

**RFID** Radio-frequency identification.

**RPi** Raspberry Pi.

**RSSI** Received Signal Strength Indicator.

**RTLS** Real-time locating systems.

**SBC** Single Board Computer.

**SoC** System On a Chip.

**SSL** Secure Sockets Layer.

**TLM** Telemetry.

**TLS** Transport Layer Security.

**UID** Universal Identifier.

**URL** Uniform Resource Locator.

**UUID** Universally Unique IDentifier.

**WLAN** Wireless Local-Area Network.

# 1 Introduction

## 1.1 Context Analysis

In the past few years, hospitals and Emergency Department (ED) are experiencing a stressful both financial and operational situation due to their low patients' throughput. Their capacity tends to saturate because of the lack of personnel, beds, and equipment, and a physical increase in the capacity is usually out of question due to the limited financial resources.
Rather than only relying on the increase of the physical capacity, an alternative solution is to increase the service capacity, thus improving the patients' throughput [1]. This could be made possible through the analysis of patient tracking data, understanding which is the most congested unit. Patient tracking is already widely used in health processes to improve the flow of patients in medical environments, but usually, these measurements are obtrusive, involving medical personnel to manually add time and location data, thus keeping them from carrying out medical tasks and increasing the probability of errors and inaccuracies.
A more innovative and technology-based solution, as the one presented in this report, regards the exploitation of Indoor Real-time locating systems (IRTLS) [2], who unobtrusively take the needed data, with no personnel involved. Apart from the analysis of the flow of patients, a real-time tracking method could also be exploited to track indoor patients' movements in order to guarantee the safety for those patients with unstable mental conditions (e.g. Alzheimer or dementia) or to provide a medical asset tracking system to identify, track and manage medical assets in healthcare environments [3].

## 1.2 Project Objective

This report aims to present a working prototype of an indoor real-time patients' location system suited to health facilities, aiding to the identification of bottlenecks and to the increase of patients' throughput. Before presenting the proposed solution, it is important to discuss which specific requirements this project wants to achieve and which are the major constraints that must be satisfied. The MoSCoW method [4] was implemented to define and prioritize system requirements. Hence, the characteristics of the prototype are divided as follows:

- MUST
  - track patients in specific checkpoints inside the health facility, sending location information to a data collector;
  - avoid to interfere with medical equipments;

  - respect the privacy of the patients and the involved facility.

- SHOULD
  - be able to process the acquired data to extract useful information on the flow of patients.

- COULD
  - have a Graphical User Interface (GUI) to help medical personnel to better interact with the system.

- WON'T (this time)
  - have a mobile application.

To meet these requirements, constant comparison with previous works and benchmark solutions found in the current state-of-the-art has been carried out along the entire development of the project (section 1.3). The decision of devices, software, and protocols to employ has been carefully accomplished and presented in the section 2 keeping in mind the requirements and constraints of the project. Eventually, a working prototype is demonstrated in section 3 together with a discussion of the main limitations and advantages of the current proposed solution in a realistic environment.

## 1.3 Related Works

The introduction of IRTLS in health facilities is still in its infancy, as confirmed by Yoo et al. [5] in a recent study. Few vendors integrate software-based systems for patient flow monitoring and, albeit numerous hospitals have recently started to assess the benefits of these systems, they do not always share their experiences or data, making the comparison between different solutions and environments very hard to define.
Fernandez-Llata et al. [6] present a system based on Real-time locating systems (RTLS) to introduce a more automatic method for clinical pathways [7] deployment. They propose a process mining-based methodology providing a graphical representation of the process' execution by taking valid information from the exploited RTLS. The location system is Radio-frequency identification (RFID)-based and it has been provided by MySphera [8]. Each patient is identified by a RFID bracelet [9], which periodically sends frames through the ZigBee protocol [10] to beacons situated in fixed points. Then, the beacons, through the Ethernet protocol [11], forward the patient's information to a location server that will evaluate the positions.

The graphical view of the process is obtained through the process mining application named "PALIA ILS", based on the process discovery algorithm called "PALIA" [12]. This system was validated in the General Hospital of Valencia (Spain) for three months, showing how efficient the integration between IRTLS and process mining techniques is by presenting a graphical view of patients' transitions from one unit to another one, highlighting the most followed path through color code. This system has proved to be well accepted by the health personnel, helping them to achieve a better knowledge of the behavior in a clinical pathway as well as allowing the RTLS technicians to detect failures in the localization process. The project that is going to be presented in this report, differently from the work proposed by Fernandez-Llata et al., is more focused on the realization of the RTLS system. However, given the structure of the data collector and the organization of the patients' information, efficient queries can be performed to achieve valid and useful information about the flow of patients and bottlenecks in the health facility.

Yoo et al.[5] suggest a real-time asset tracking system based on Bluetooth Low Energy (BLE) [13] and Wi-fi technology [14]. In this case, they do not want to track people but medical assets that are subjected to high mobility within the facility. The tracking system on each asset connects to the existing Wi-Fi infrastructure, providing information at the room level. Moreover, thanks to BLE beacons situated in each room, the asset can also send more detailed information regarding its position inside the room. Although this system proved to be satisfactory from the end-users perspective, many limitations should be solved, like the battery drainage problem, the strict dependency on the existing Wi-Fi infrastructure, and the cost of the active tags. Even if the purpose is slightly different, the system proposed in this report will try to address these problems by submitting a flexible, energy-efficient, and inexpensive infrastructure.

Drazen and Rhoads [1] discuss how the use of tracking tools can improve patient flow in hospitals. More in detail, they compare two methodologies: IRTLS and existing event-driven data. The hospitals should decide which methodology they need according to what kind of patient flow challenge the organization has to face. If the facility needs a high-level of accuracy to track not only patients but also staff and assets, then IRTLS is the best choice. If instead, the facility is already provided with a rich information environment and does not require a real-time accuracy, then the event-driven data should be the solution. The paper provides a set of best practices to be followed by the medical staff to efficiently introduce the proper tracking tool. Finally, five real cases of health delivery organizations that have explored these technology systems to improve patient flow are presented. The ability to manage patient flow significantly improved in all the cases, with a general reduction of patient visit length, an increase of admitted patients, an increase of the bed capacity, and a great decision-making help thanks to the

detailed information that the staff can obtain through efficient queries. The work carried out by Drazen and Rhoads gives a strong and detailed motivation of why hospitals should introduce patient tracking in their systems and how much the study of innovative solutions is important in this field.
Moreover, there are some other companies implementing similar IRTLS projects that are worth to be mentioned. To have a better idea about the complexity, scope and the used technologies, the competitor analysis has been performed using the products given in Table 1.

| Company Name | Technology Used | Scope of the Product |
|---|---|---|
| Go Codes [15] | QR codes | Medical Asset Tracking |
| Centrak [16] | RFID tags, Gen2IR | Medical Asset Tracking , Patient Flow |
| Borda [17] | RFID tags | Patient and Asset Tracking |
| MySphera [8] | RFID tags | Patient Flow, Asset and Patient Tracking |

Table 1: Benchmarking Results

As Table 1 shows, the involved technology varies according to the scope of the product. In order to implement more precise patient and asset tracking systems, RFID tags are commonly used, which is a more expensive solution compared to the systems exploiting BLE beacons and QR codes.

# 2 Proposed System

The proposed approach wants to directly track the patients in each zone of an ED, thus collecting information on their path together with the time spent in different zones. To do so, the patients have to be provided with a wearable device that identifies itself and sends messages related to the patient's identity during the entire journey. Relay nodes are placed in specific checkpoints that identify with the main zones of the health facility. When a relay node receives a message from a patient's wearable, it will add to this message the information about its location and a timestamp, forwarding the modified information to a gateway. At this point, the latter will further process the message placing it in a database with a given structure. Thanks to the database organization and its properties, very efficient queries can be performed to extract useful information about the patient's flow and bottlenecks of the facility. The Figure 1 depicts the concept of our system, named **RFED**. In the following sections a more accurate description of the system's structure is presented, while in section 2.5 a finer representation of the project implementation will be described.

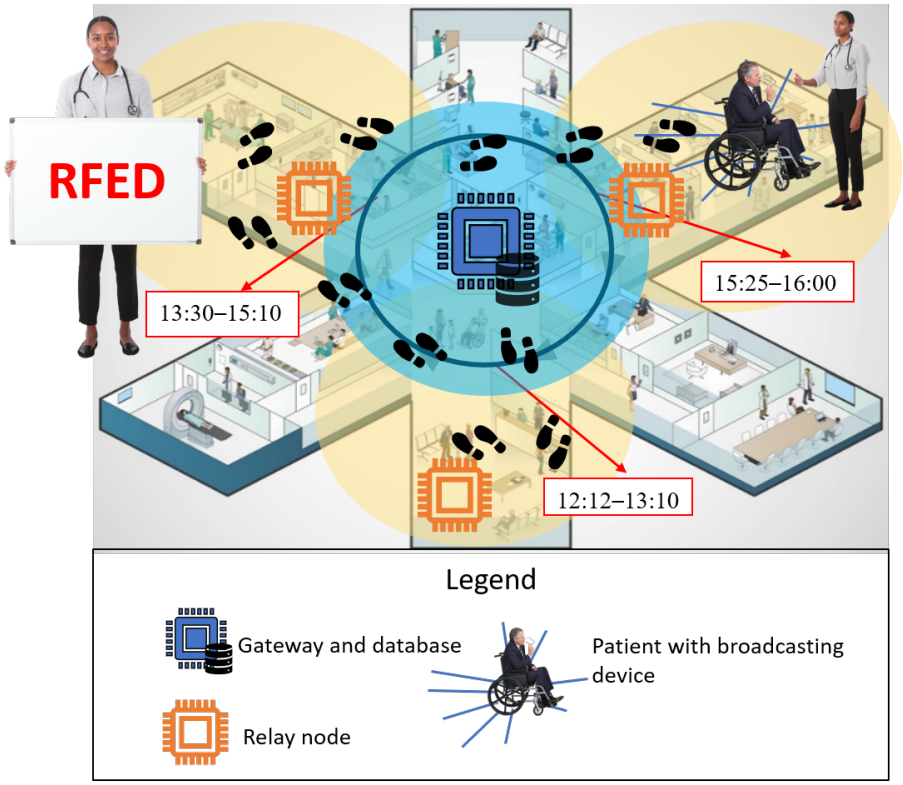


Figure 1: RFED concept

## 2.1 Communication Protocols

The communication protocols exploited to send patient related information are fundamental to determine the final costs, the security, and the energy requirements of the system. The communication flow was divided in two phases:

1. from the patient's wearable device to fixed small gateways in each checkpoint;
2. from each checkpoint to the main data collector.

How much the medical staff is involved in the patient tracking depends on the first part, where it is possible to use either active (RFID, Bluetooth, Wi-Fi) or passive (QR codes, bar codes) solutions: the former increase the automation of the entire process as well as the costs, while the latter is cheaper, albeit they need a major commitment from the medical staff. By looking the given constraints and requirement, and considering different comparisons among the existing solutions [18], [19] ,[10], the final choice is to provide each patient with a **Bluetooth Low Energy (BLE) beacon** device that broadcasts, through this low power wireless protocol, location information to the relay nodes in each checkpoint. In this way the automation of the process in enhanced while the costs and energy consumption are kept low.
The second communication phase involves information transfer from the relay nodes to the

gateway. Many existing solutions involve the use of the already existing hospital internet infrastructure to perform this task, but, this solution, can be unfeasible when the infrastructure does not reach all the departments or even exist. Moreover, sending these typologies of information through the internet may bring to a lot of security issues. For these reasons the choice was to deploy a private **Wi-Fi Mesh network**, where all the gateways can act as relays for other gateways until the gateway is reached. The application-level protocol that is used to forward these messages is the **Message Queue Telemetry Transport (MQTT)** protocol [20] since it can handle several messages at the same time as well as a lightweight messaging transport.

## 2.2 Hardware

### 2.2.1 Global Tag BLE Development Kit

The Global Tag BLE Development Kit is a BLE Beacon kit proposed to test beacons and to develop BLE softwares. Being more expensive than its competitors, the kit provides BLE beacons with embedded nRF52810 chip (Nordic Semiconductor) supplied with CR2032 battery (DURACELL®), support for the beacon formats both Eddystone (Universal Identifier (UID), Telemetry (TLM), Uniform Resource Locator (URL))[1] and iBeacon [2] protocols, built-in accelerometer and temperature & humidity sensors and well designed silicon wristbands. In addition, Global Tag provided technical support by phone and email and technical documentation.

The quality of the product, protocols, design of the wristband and technical support were the key parameters in the choice of the Global Tag BLE Development Kit over their competitors.

### 2.2.2 ESP32-DevKitC

The ESP32-DevKitC development board from Espressif was the chosen hardware to be the bridge between the wearable BLE beacons and the gateway. This low-footprint low cost hardware is powered by the System On a Chip (SoC) ESP32, which provides 2.4 GHz Wi-Fi (full Institute of Electrical and Electronics Engineers (IEEE) 802.11 b/g/n Wi-Fi Medium Access Control (MAC) protocol), BLE specifications complient and security due to the embedded ATECC608A chip[3].

On the software side, Espressif provides the Espressif IoT Development Framework (ESP-IDF) and the ESP-MDF which ease the prototype development. More details about these development frameworks is covered in section 2.3.2.

[1] `https://developers.google.com/beacons/eddystone`
[2] `https://developer.apple.com/ibeacon/`
[3] `https://www.microchip.com/wwwproducts/en/ATECC608A`

Therefore, taking into consideration the technical requirements of the project, time to market, prototype design and cost, this Microcontroller (MCU) was a great fit to detect the BLE beacons broadcasted by wearable devices, filter, and relay information to the gateway via Wi-Fi mesh network.

### 2.2.3 Raspberry Pi 3B+

The RPi 3B+ is the chosen Single Board Computer (SBC) to be the gateway of the project and act as a 2.4GHz Wi-Fi hotspot(IEEE 802.11b/g), message broker, data processor and database.

On the hardware side, it provides enough computational power, Wi-Fi dual-band IEEE 802.11b/g/n/ac, storage capacity via Micro-SD and Power over Ethernet (PoE), IEEE 802.3af-2003. The last feature, which eases the installation, reduce cost and cables, was the differential that this model was chosen over its previous version, the 3B.

On the software side, MongoDB, the chosen database for the project, is not able to run on the official RPi Foundation Operational System (OS), the 32-bit Raspbian, but on the 64-bit Ubuntu Mate, which is available up to the RPi 3B+.

Therefore, with these hardware and software characteristics, the RPi 3B+ would be able to act as a Wi-Fi 2.4Ghz Access Point (AP) and message broker for the ESP32 mesh network, database and data processor, fulfilling all the necessary technical requirements of the project gateway. In addition to the technical aspects, the group expertise and cost of the hardware were also taken into consideration in order to guarantee a short time to market and low cost of the prototype.

## 2.3 Software

### 2.3.1 Database

Database systems are used to provide an important way of monitoring and improving the value of healthcare services. By having the appropriate data stored in a database, it makes easier both to identify the bottlenecks of the health system and to improve the quality of the system itself.

It is possible to use either Not Only Structured Query Language (NoSQL) or Relational Database Management System (RDBMS). RDBMS is a database management system based on the relational model of data and it is a completely structured way of storing data. On the other hand, NoSQL databases are able to handle unstructured data, which is growing far more rapidly than structured ones and does not fit the relational schemes of RDBMS [21], [22].

There are different NoSQL databases such as Couchbase[4], Casandra[5], Amazon DynamoDB[6], and MongoDB[7]. Each NoSQL database has its own features and the architecture of the database depends on the application. Taking into account all the features, the final choice is to use **MongoDB** as a database for this project [23].

MongoDB is a cross-platform, document-oriented database that provides high performance and availability and easy scalability. It is highly optimized for storage JavaScript Object Notation (JSON) format data and database with Representational State Transfer (RESTful) API. One of the benefits offered by MongoDB is the possibility to use dynamic schemas that eliminates the existence of a predefined structure.

The data inside the database can be stored either locally or on the cloud. Since MongoDB is chosen as a database for the project, one of the options is to use MongoDB Atlas[8], which is a fully-managed cloud database. It is worth to use a cloud database for back-up and real-time information. The final choice is to store data only locally at this time due to budget constraints (Atlas involves a paid subscription).

### 2.3.2 ESP32 Development Framework

In order to implement the communication between ESP32 devices and the gateway, RPi, the ESP-MESH network protocol is used. The software stack of this protocol is built atop the Wi-Fi Driver/FreeRTOS[9], as depicted in the software stack diagram, Figure 2.

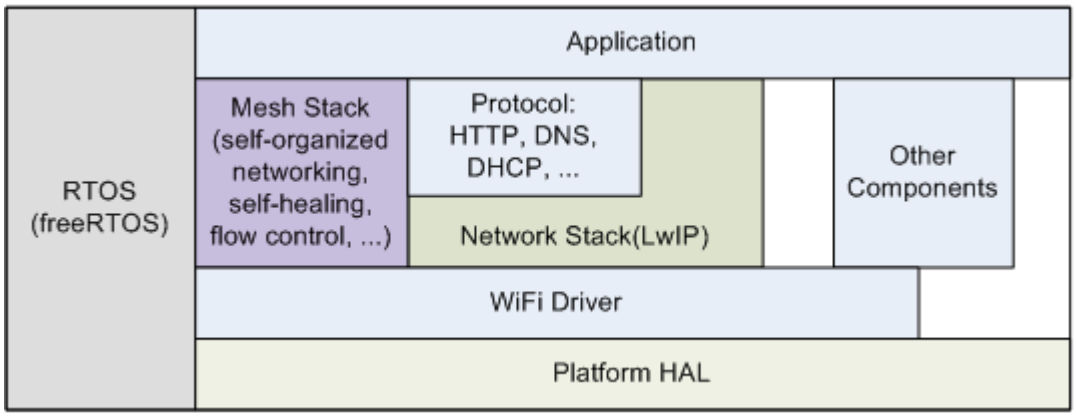



Figure 2: ESP-MESH Software Stack. Ref: Espressif

[4] https://www.couchbase.com/
[5] https://cassandra.apache.org/
[6] https://aws.amazon.com/dynamodb/
[7] https://www.mongodb.com/
[8] https://www.mongodb.com/cloud/atlas
[9] https://docs.espressif.com/projects/esp-idf/en/stable/api-reference/network/esp_mesh.html

ESP-MESH is a multiple hop (multi-hop) network, allowing several ESP32 devices to be connected under the same Wireless Local-Area Network (WLAN) even though if not connected directly nor in range of the AP, Figure 3. In addition, the ESP-MESH is self-organizing and self-healing, meaning the network can be built and maintained autonomously[10]. Therefore, exploiting ESP-MESH, the system covers large areas maintaining its independence from the client's facility and avoiding network maintenance. Possible drawbacks of this network could be overload of upstream, latency, hard to troubleshoot, packet loss due to parent node switching and link or node failure leading to idle nodes[11].

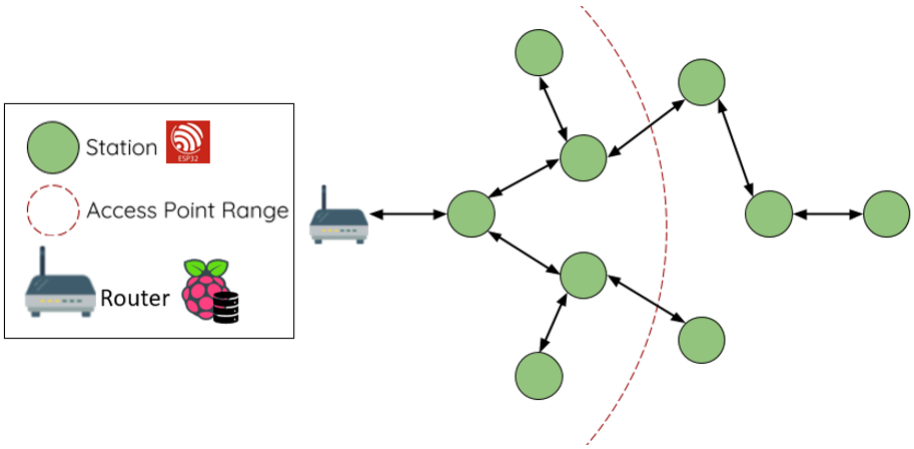



Figure 3: ESP-MESH Network Architecture. Modified from: Espressif

The implementation is done by exploiting ESP-MDF, based on the ESP-MESH network protocol software stack. ESP-MDF is built on top of ESP-IDF that essentially contains API (software libraries and source code) for ESP32 and scripts to operate the toolchain. ESP-IDF API is mainly composed of: Bluetooth (Classic, BLE, BLE-MESH), Networking (Wi-Fi, Ethernet, Internet Protocol (IP) and Application Layer), Peripherals (I2C, SPI, ADC, DAC, CAN, UART, etc) and System (FreeRTOS, Watchdogs, Sleep Modes, Over the Air (OTA), etc)[12].

The ESP-MDF contains the Mwifi API, a networking API, which encapsulates the Wi-Fi and ESP-MESH API, and add other functionalities[13]. Mwifi is the main driver of the RFED project. Figure 4 shows the ESP-MDF stack on top of the ESP-IDF, with the highlighted Mwifi under Utils.

[10] https://docs.espressif.com/projects/esp-idf/en/stable/api-guides/mesh.html
[11] https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/mesh.html
[12] https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/index.html
[13] https://docs.espressif.com/projects/esp-mdf/en/latest/api-reference/mwifi/index.html

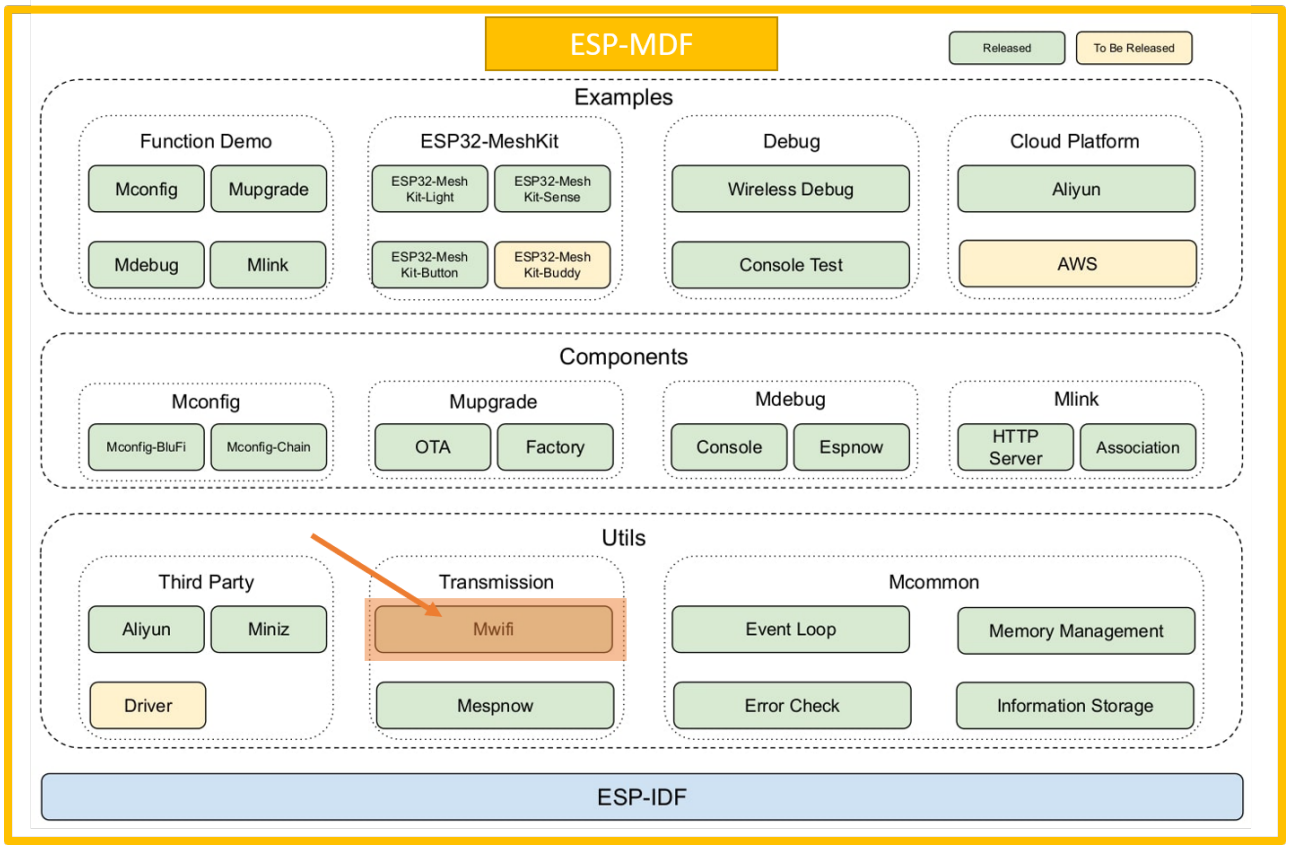



Figure 4: ESP-MDF stack with highlighted Mwifi API, the main driver of RFED. Modified from: Espressif Github

## 2.4 Security

This work has to deal with sensitive information of patients, given that the messages that are exchanged in the system bring knowledge about personal and private details. In Europe, the main privacy law whose aim is to regulate how companies can use all the typologies of personal data is called General Data Protection Regulation (GDPR) [24]. Although GDPR compliancy involves several and multidisciplinary aspects, the attention is drawn on the technical ones, which are:

1. encrypted exchange of messages between gateways;

2. efficient data discovery, secure data storage and quick breach detection at the database.

In the message exchange between gateways, data are encrypted at the transport layer through Secure Sockets Layer (SSL) protocol, while Transport Layer Security (TLS) protocol is exploited when the RPi sends the information to the main data collector [25]. A figurative representation of the scenario is shown in figure 5.

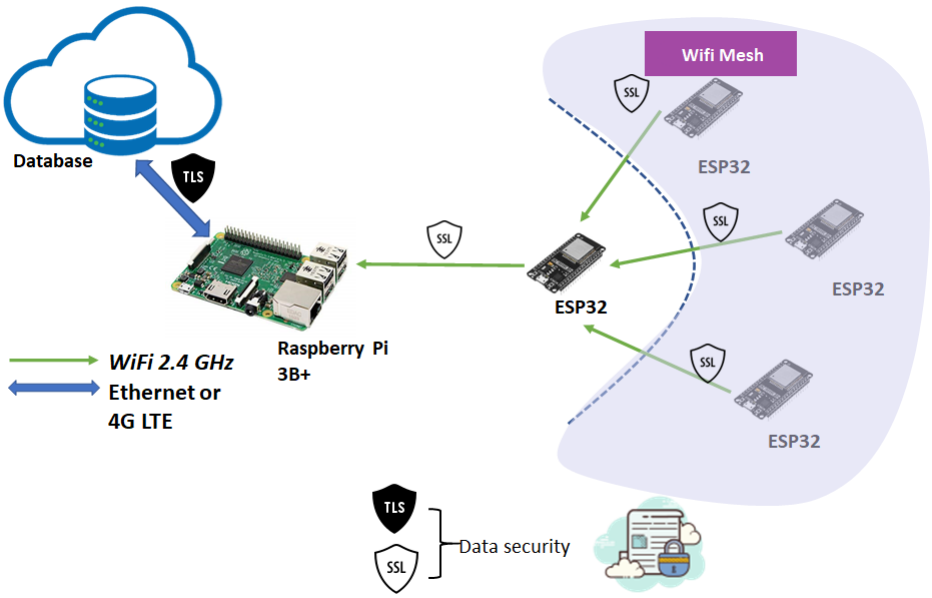



Figure 5: The graphical representation of the communication network.

As mentioned in section 2.3, the data collector that is in charge of data storage and queries is MongoDB. This specific database is also able to provide all the capabilities to implement data discovery, defence and detection demanded by the privacy regulations. Table 2 shows the strategies implemented by MongoDB in order to facilitate the GDPR compliancy.

Detailed information on these security strategies are well explained on the official MongoDB documentation [26], while how exactly MongoDB implements them in order to facilitate GDPR compliancy is presented in [27].

## 2.5 Project Implementation

As already discussed (section 2), the main idea of the project is to build a prototype that can track patients in each ED zone and aids the identification of the bottlenecks in the system.

The functionality of RFED system, from a standard ED's patient flow, starts when the patient enters the ED, then the triage nurse fills out the necessary data into a GUI, as part of the RFED system, and provides each patient with the wearable BLE wristband. The patient's position is tracked by using these wearable beacon bracelets. The **BLE** beacon packets are broadcasted from the patient's beacon bracelets and detected by SoC ESP32 devices. Later, the information of the patient's location is relayed in the **ESP32 MQTT Mesh** until it reaches the ESP32 root node. The latter delivers the messages to the gateway, **RPi**. The proposed solution is different from the ones that were discussed in section 1.3 because first it is cheaper since BLE beacons are low cost and have long battery life. Second, it has a flexible structure

| **Discover** | **Defend** | **Detect** |
|---|---|---|
| Identify Personal data<br>• Queries and Analytic | Access Control<br>• Authentication (SCRAM, x.509 certificates)<br>• Authorization (RBAC)<br>• IP White listing | Monitor and Report<br>• Real-Time alerting |
| Personal Data Retention<br>• TTL indexes | Pseudonymization and Encryption<br>• Read only views<br>• Field-Level Encryption<br>• TLS/SSL Network Encryption | Audit<br>• MongoDB Audit Log<br>• MongoDB Change Streams |
| | Resilience<br>• Replica Sets | |
| | Data Sovereignty<br>• MongoDB Atlas Global Clusters | |

Table 2: Data discovery, defence and detection requirements demanded by modern privacy regulations and implemented by MongoDB.

that is independent from the hospital's existing network structure. Third, it has a user-friendly design, which is a plus for patients who wear these wristbands.

### 2.5.1 Beacons Detection Strategy

Beacon detection strategy is based on the arrival and departure of the patients inside the facility. Each BLE bracelet broadcasts its Universally Unique IDentifier (UUID), Namespace ID and Received Signal Strength Indicator (RSSI) values. UUID value is unique for each bracelet and it is used to identify which patient is detected. Namespace ID is a filtering criterion. All Global Tag beacons have the same Namespace ID value so that it is used to eliminate other detected BLE beacons in the environment. The RSSI value is the second filtering criterion which is used to eliminate the beacons out of the room. The RSSI threshold value is determined according to the structure and the size of the room. It is fixed to -65 dB in the testing scenarios which

are explained in section 3.

The Global Tag BLE beacons can be configured as both iBeacon and Eddystone (UID, URL, TLM) advertising frame types. The decision is made as Eddystone since it allows further sensor data implementations, which is discussed in section 5. The advertising packet type is changed to Eddystone UID using the "NRF-Connect" mobile application.

The BLE beacons which are in the range of the checkpoint ESP32 are detected over a 20 seconds period. In the filtering stage, the detected beacons are filtered according to their RSSI and Namespace ID value.Then, ARR_20 stores the filtered beacons within current 20 seconds period. There are also ARR_40 and ARR_60, which store the beacons of one period before and two period before respectively. In order to make the comparison of the stored beacons over time, a shifting procedure is implemented. At the end of each period, ARR_20 is shifted to ARR_40 and similarly, ARR_40 is shifted to ARR_60. The shifting procedure is depicted in Figure 6.

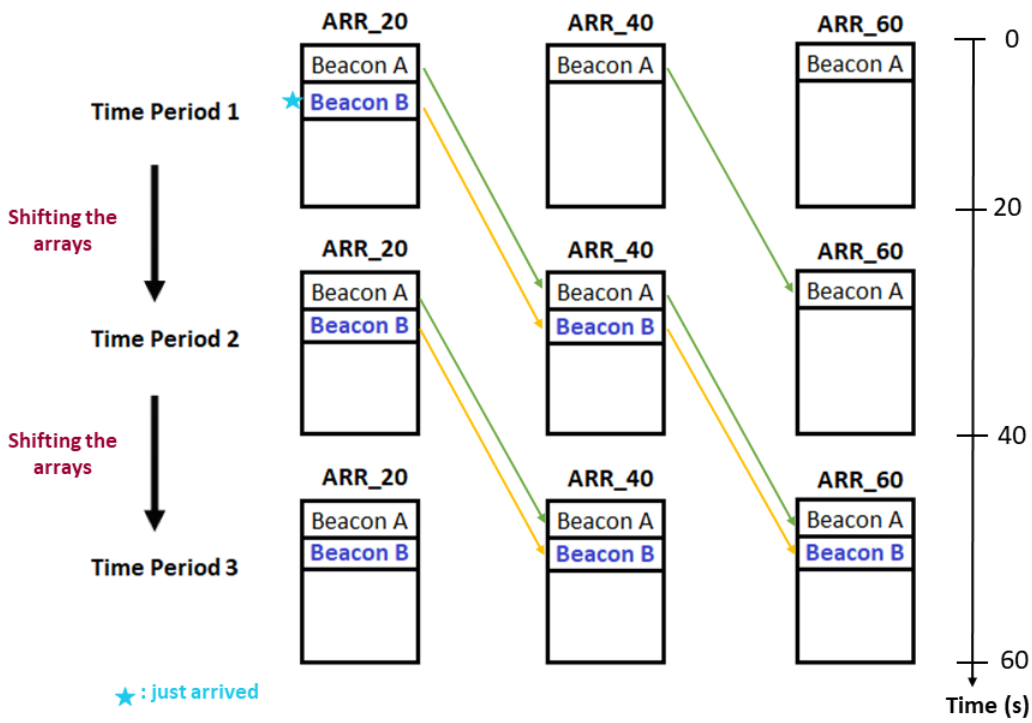


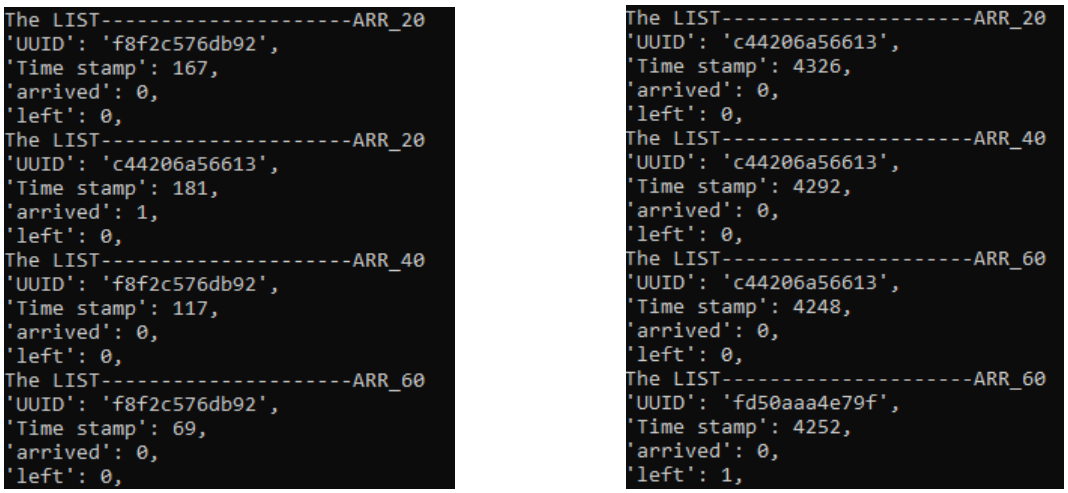

Figure 6: Shifting Flow

In order to detect the arrived patients, the patients in ARR_20 are compared with the ones stored in ARR_40 and ARR_60. If the patient with the same UUID does not exist in both arrays, that patient is labeled as "arrived" = "1". Similarly, for departure detection, the

patients in ARR_60 are compared with the patients of ARR_40 and ARR_20. If the patient exists in none of them, that patient will be labeled as "left" = "1".

For both arrival and departure detection, two sample cases are provided below.

As an example case (Figure 7a), the beacon with "UUID": "c44206a56613" is just detected and stored in ARR_20. It is labeled as "arrived" = "1" since it doesn't exist in ARR_40 and ARR_60. The beacon with "UUID": "f8f2c576db92" has been detected for three consecutive periods. It means that the patient is still in the room and was previously labeled as "arrived" = "1".




(a) Patient Arrival (b) Patient Departure

Figure 7: Detection of Patient Arrival and Patient Departure

In the second example case (Figure 7b), the beacon with "UUID": "fd50aaa4e79f" has not been detected for two consecutive time periods (ARR_40 and ARR_20). Therefore, it is labeled as "left" = "1".

In case of an arrival or a departure detection, a JSON format message is created to be sent in the mesh network. The structure of the message is discussed in the section 2.5.3.

Therefore, the detection strategy is as depicted in Figure 8.

Figure 8: Beacon Strategy Flow

### 2.5.2 ESP32 Wi-Fi MESH MQTT Implementation

The logic for the MQTT and mesh network implementation is depicted and explained in Figure 9.

From the presented flowchart, it can be inferred that first the Wi-Fi mesh is configured and initialized, followed by the MQTT. Then the main loop starts, retrieving the detected beacons information stored in a ring buffer (section 2.5.1). If the ring buffer is empty, it returns to the loop, otherwise the retrieved item is sent to the mesh network in JSON format, with the ESP32 root node as the destination. To ease modifications during the development and validation phase, the ESP32 configuration menu[14] was customized to the RFED project, avoiding typos and other mistakes before flashing(Figure 10).

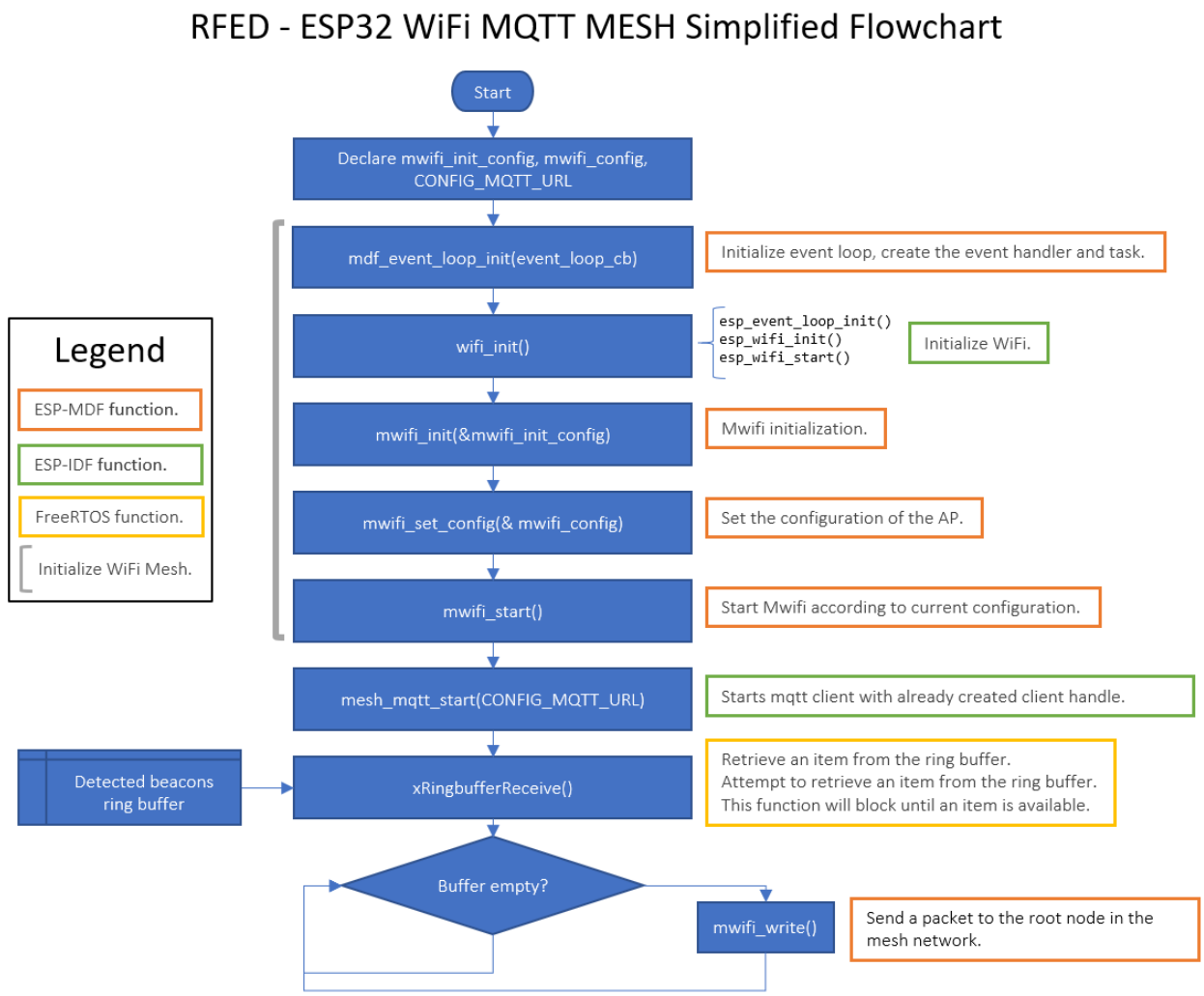



Figure 9: Simplified flowchart of the Wi-Fi, MQTT and MESH implementation code, flashed in the ESP32 devices.

[14] https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/kconfig.html

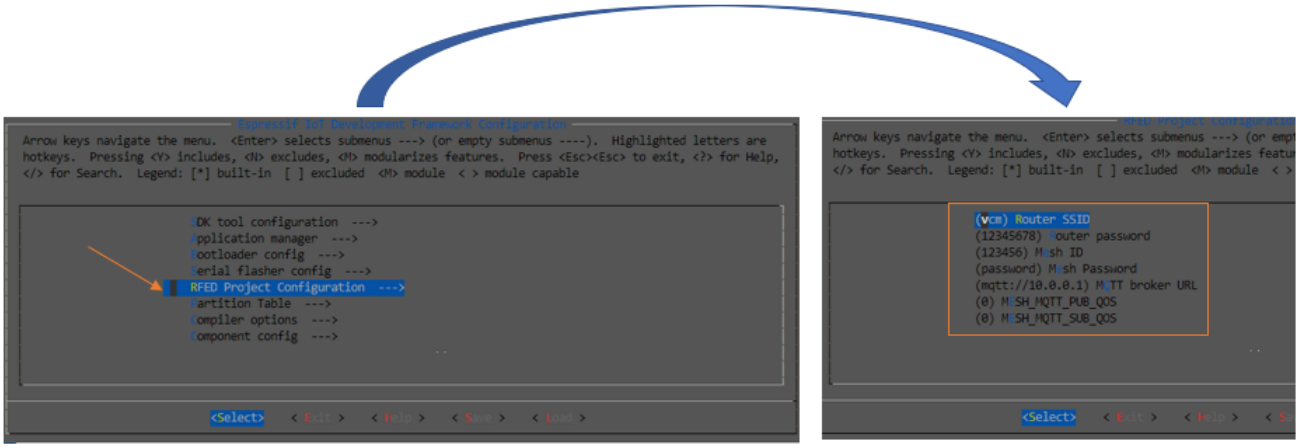



Figure 10: ESP32 Menu

### 2.5.3 Database Structure and Strategies

The final database structure is organised in four collections:

- **Patient_ID**: a mapping collection that associates a specific patient with a unique identifier;
- **MAC_IDs**: another mapping collection to associate the BLE beacons' MAC addresses with the patient who is wearing them;
- **RTPatients**: a real-time collection that provides information about the current zone where a patient is being detected;
- **PermanentPatients**: the collection that provides the historical data about the path followed by the patients together with a duration field (how long has the patient remained in a certain zone).

When a new patient arrives, it has to be provided with a unique identifier and associated with the MAC address of the wristband that will be assigned to her/him. This operation can be easily done through simple web application forms (Figure 11): the first one takes as input the name, surname, date of birth of the patient together with the MAC address of the beacon that will be used by her/him. The application will automatically update the "Patient_ID" and "MAC_IDs" collections in a way that the patient will always be recognized with the same id, even if she/he has been previously admitted in the same health facility. The second form, instead, is used when the patient leaves the facility, in order to dissociate the beacon's MAC address from that patient.

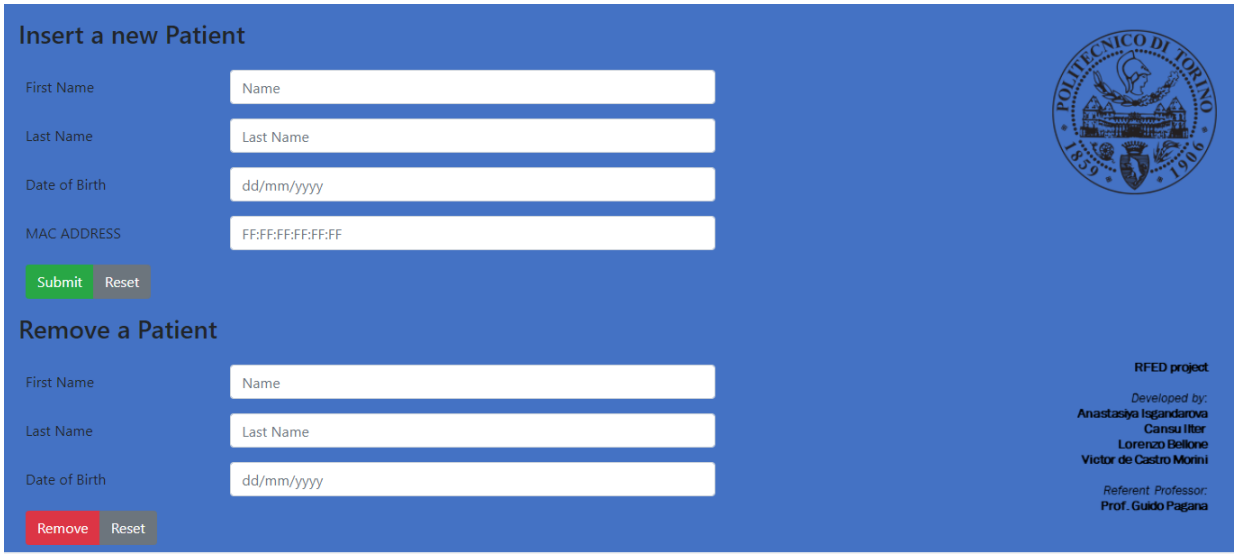



Figure 11: Web application forms to insert a new patient and to remove a leaving one.

The other two collections (RTPatients and permanentPatients), instead, are automatically updated by the RPi according to the received messages. The gateway acts as an MQTT subscriber, which receives messages in JSON format with the following structure:

$$\{"mac_address" : x, "room" : y, "timestamp" : z, "detected" : boolean\}$$

As soon as the subscriber receives a message from the ESP mesh network, it maps the beacon's MAC address with the unique patient ID (from the "MAC_IDs" collection). Then, it has to check the "detected" field. If it is "True" the patient is detected in the specified room for the first time, hence, a new document needs to be inserted in the real-time collection giving information about the new position of the patient. A document in the "RTPatient" collection has the structure: $\{patient_id, room, timestamp\}$.
If the detected field is "False", instead, it means that the patient left that room; the correspondent document must be removed from the real-time collection and a new document has to be added in the permanent collection, including for how long has the patient remained in that room. A "permanentPatient" document has the structure: $\{patient_id, room, timestamp, duration\}$. An accurate flowchart of the subscriber strategy is shown in Figure 12.

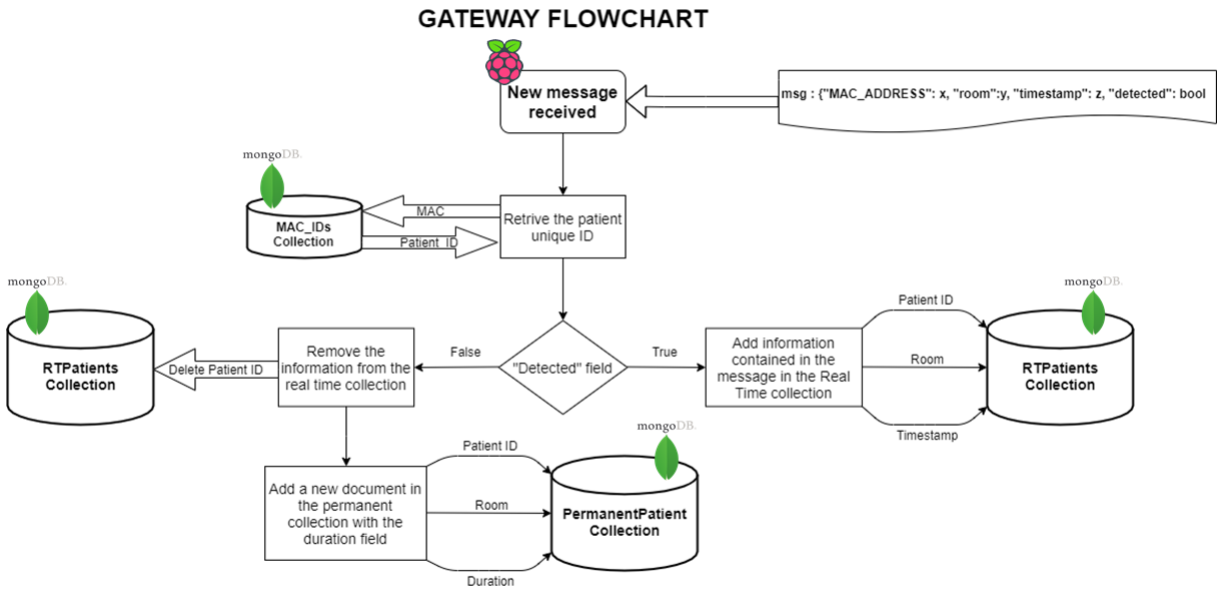



Figure 12: Flowchart representing the strategy applied by the subscriber to update the database when a new message is received from the ESP mesh network.

Sometimes it can happen that an arrival message of a patient in a certain room arrives at the RPi before his/her departure message from the previous room. This situation is properly handled by considering the arrival message also as a departure message from the previous room. When the real departure message will arrive at the RPi, it will have no effect on the database.

### 2.5.4 Prototype Packaging

The packaging was designed for the ESP32 (Figure 13) and for the RPi (Figure 14).

For 3D printing the models, the printer specifications should be taken into account and the 3D model should be reviewed accordingly.

It is recommended to change the ESP32 Development Kit with Printed Circuit Board (PCB) antenna to one with an IPEX antenna if the boxes are decided to be used. The Figure 15 illustrates the visual difference among them.

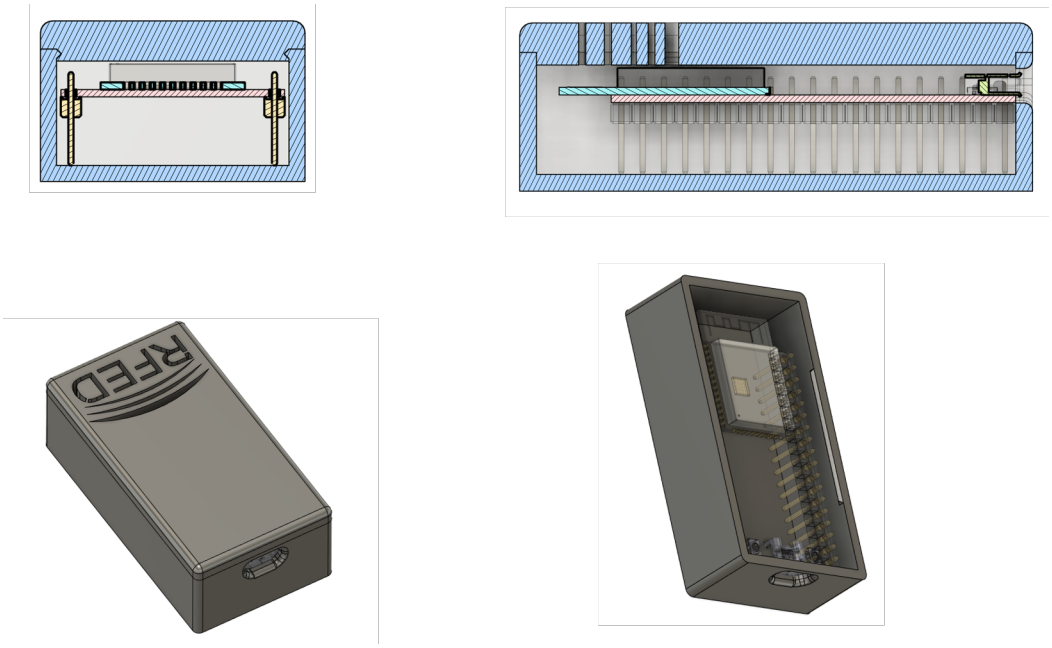



Figure 13: RFED ESP32 development board box model

Figure 14: RFED RPi box model

| Module | Description | Chip Embedded | Dimensions (mm) | Pins | Flash (MB) | PSRAM (MB) | Antenna | Development Board |
|---|---|---|---|---|---|---|---|---|
| ESP32-S2-WROVER | ESP32-S2-WROVER integrates ESP32-S2. It is a powerful, generic Wi-Fi MCU module that has a rich set of peripherals. | ESP32-S2 | 18x31x3.3 | 42 | 4,8,16 | 2 | PCB antenna | ESP32-S2-Saola-1 |
| ESP32-S2-WROVER-I | ESP32-S2-WROVER-I integrates ESP32-S2. It is a powerful, generic Wi-Fi MCU module that has a rich set of peripherals. | ESP32-S2 | 18x31x3.3 | 42 | 4,8,16 | 2 | IPEX antenna | ESP32-S2-Saola-1 |

Figure 15: Highlighted PCB and IPEX antenna. Two different models of ESP32. Modified from: Espressif

# 3 Experimental Results

The prototype has been realized and tested in the environment depicted in Figure 16.

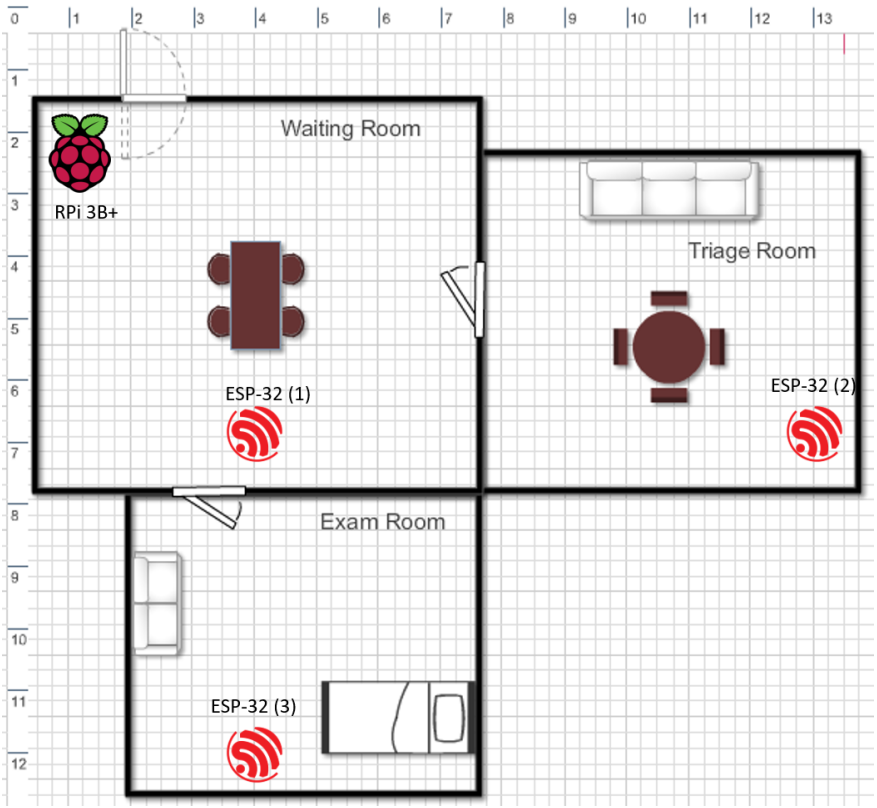



Figure 16: Graphical representation of the environment in which the project has been tested.

Each room has been provided with one ESP32 while the RPi has been placed in the "Waiting room". The ESP32 node in the waiting room acts also as the root node, thus receiving the

messages that come from the triage and exam rooms, and forwarding them to the gateway. Once the mesh network is established (as explained in section 2.5.2), three patients (that will be identified as patient A, B and C) can be simulated through the three available BLE beacons. Three slightly different scenarios have been tested, each one described by specific phases for each patient, with the aim at comparing how accurately the system can measure for how long the patients remain in certain rooms.

Each scenario was tested only once, therefore not statistically meaningful, but it can still provide us with insights regarding the prototype functionality.

## 3.1 Scenario 1

For the first scenario, two patients (A and B) and two rooms (waiting and exam) are considered. The experiment proceeds as follows:

1. A and B arrive at the same time in the waiting room (waiting1 phase).
2. After one minute, patient A moves from the waiting room to the exam room. B joins A in the exam room after another minute (exam1 phase).
3. After two minutes A comes back in the waiting room, while B comes back in waiting room after another minute. When B also arrives in the waiting room, after 1 minute both patients leave the facility (waiting2 phase).

The results of the first scenario can be appreciated and compared in table 3 and graph 17.

| Scenario's Phase | Patient | Real Duration (s) | Measured Duration (s) |
|---|---|---|---|
| Waiting1 | A | 60 | 74 |
| Waiting1 | B | 120 | 116 |
| Exam1 | A | 180 | 173 |
| Exam1 | B | 180 | 195 |
| Waiting2 | A | 120 | 145 |
| Waiting2 | B | 60 | 81 |

Table 3: Numerical results from scenario 1

The numerical results obtained from the first scenario are very promising and accurate. Each patient's path is correctly detected by the ESP32 boards and the error between the real duration and measured one is always below twenty seconds. Since the main goal of this project is to detect bottlenecks, the accuracy obtained in the duration measurements is satisfying for now.

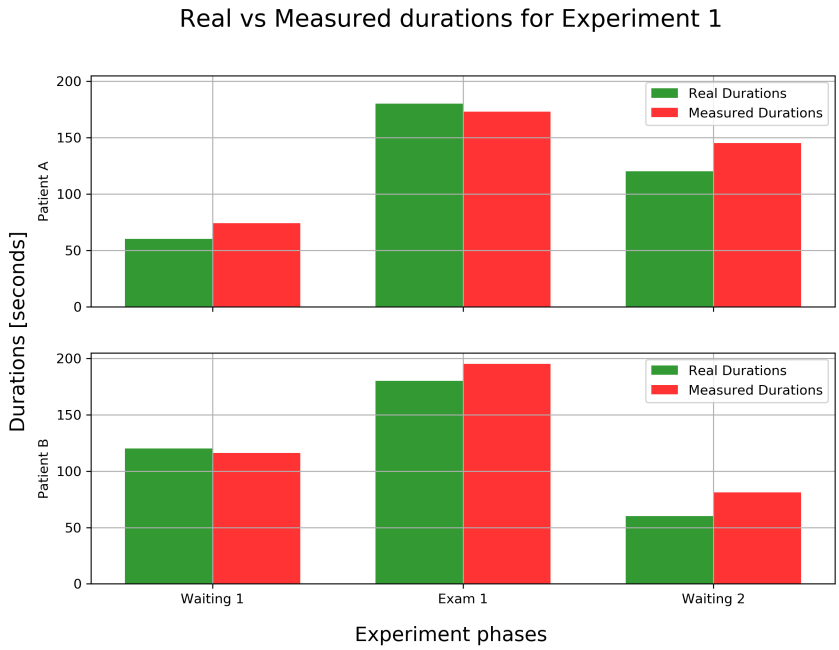



Figure 17: Real durations (green bars) compared to measured ones (red bars) for each phase and patient of the first scenario.

## 3.2 Scenario 2

In the second scenario, three patients have been considered (A, B and C) together with the same two rooms (waiting and triage), following these steps:

1. Each patient arrives in the waiting room one minute after the previous patient arrival (waiting1 phase).

2. After three minutes from his own arrival, each patient leaves the waiting room and reaches the exam room (exam1 phase).

3. Once again, after three minutes in the exam room, each patient comes back in the waiting room. The patients stay in the waiting room for other three minutes and then leave (waiting2 phase).

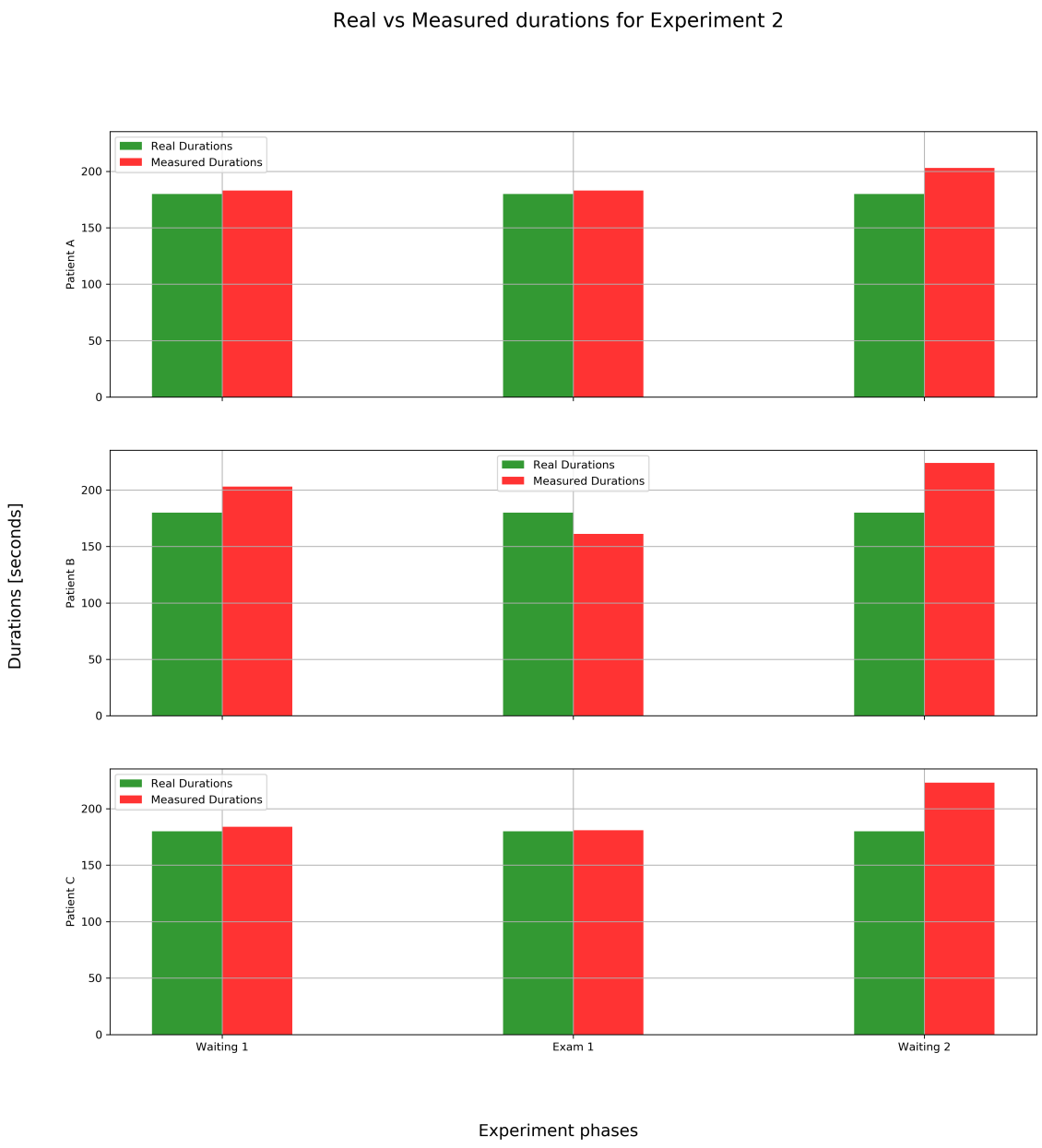



Figure 18: Real durations (green bars) compared to measured ones (red bars) for each phase and patient of the second scenario.

The inclusion of an additional patient has brought a slightly deterioration of the results, but only for the last phase. In fact, all the patients' paths are correctly detected and the duration errors are below twenty seconds for all the rooms but the last one, where the error increases up to forty seconds. Although the accuracy is still acceptable for this project's purposes, it has been important to discover the entity of this worsening results, in order to make the system scalable with the number of patients.

| Scenario's Phase | Patient | Real Duration (s) | Measured Duration (s) |
|---|---|---|---|
| Waiting1 | A | 180 | 183 |
| Waiting1 | B | 180 | 203 |
| Waiting1 | C | 180 | 184 |
| Exam1 | A | 180 | 183 |
| Exam1 | B | 180 | 161 |
| Exam1 | C | 180 | 181 |
| Waiting2 | A | 180 | 203 |
| Waiting2 | B | 180 | 224 |
| Waiting2 | C | 180 | 223 |

Table 4: Numerical results from scenario 2

### 3.3 Scenario 3

For the third scenario, still three patients have been considered while the last ESP32 has been involved in the mesh, exploiting also the "Triage Room":

1. Patients A, B and C arrive at the same time in the waiting room (waiting1 phase). After two minutes patient A leaves and goes in the triage (triage1 phase) while patient B reaches the exam room (exam1 phase). Patient C will never leave the waiting room.

2. After two minutes, patients A and B perform a room swap (exam2 and triage2 phases).

3. After other two minutes, the two patients will come back in the waiting room (waiting2 phase), where patient C is still located. The three patients will leave the facility after two minutes.

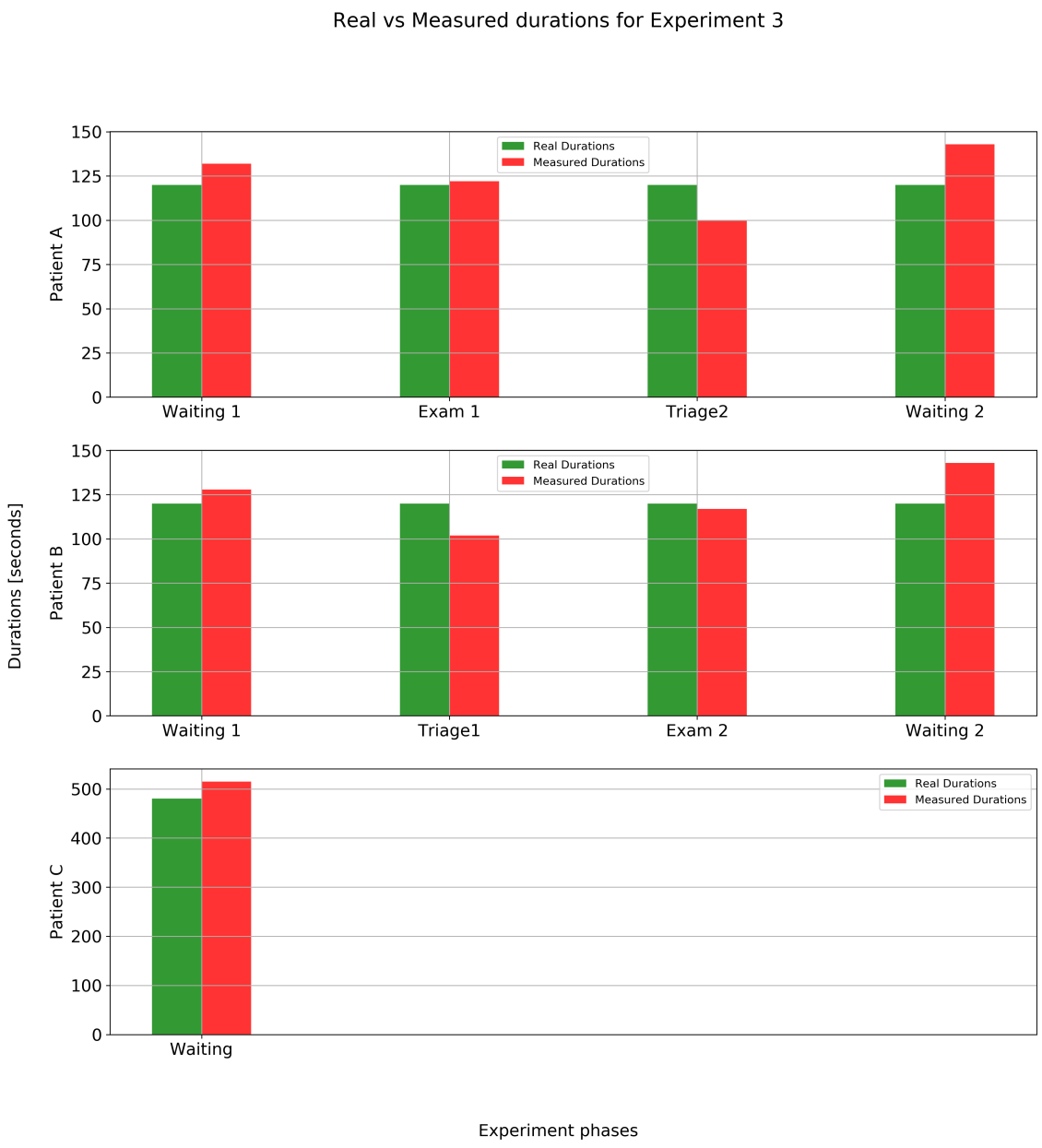



Figure 19: Real durations (green bars) compared to measured ones (red bars) for each phase and patient of the third scenario.

The more complex scenario of the third experiment seems to have no bad effects on the final results. The patients' paths are still correctly detected, even considering the room swap. The duration errors are still below 20 seconds, except for patient C, where the error is thirty-five seconds. However, as long as the maximum error remains below the minute, the system perfectly achieves the desired accuracy for bottlenecks detection.

| Scenario's Phase | Patient | Real Duration (s) | Measured Duration (s) |
|---|---|---|---|
| Waiting1 | A | 120 | 132 |
| Waiting1 | B | 120 | 128 |
| Waiting | C | 480 | 515 |
| Exam1 | A | 120 | 122 |
| Triage1 | B | 120 | 102 |
| Triage2 | A | 120 | 117 |
| Exam2 | B | 120 | 100 |
| Waiting2 | A | 120 | 143 |
| Waiting2 | B | 120 | 143 |

Table 5: Numerical results from scenario 3

# 4 Business Model

The Canvas Business Model is developed in order to reflect the business systematically by using nine key building blocks. It describes the business and how it will continue in the future. The business model Canvas for the RFED project is given in Figure 20.

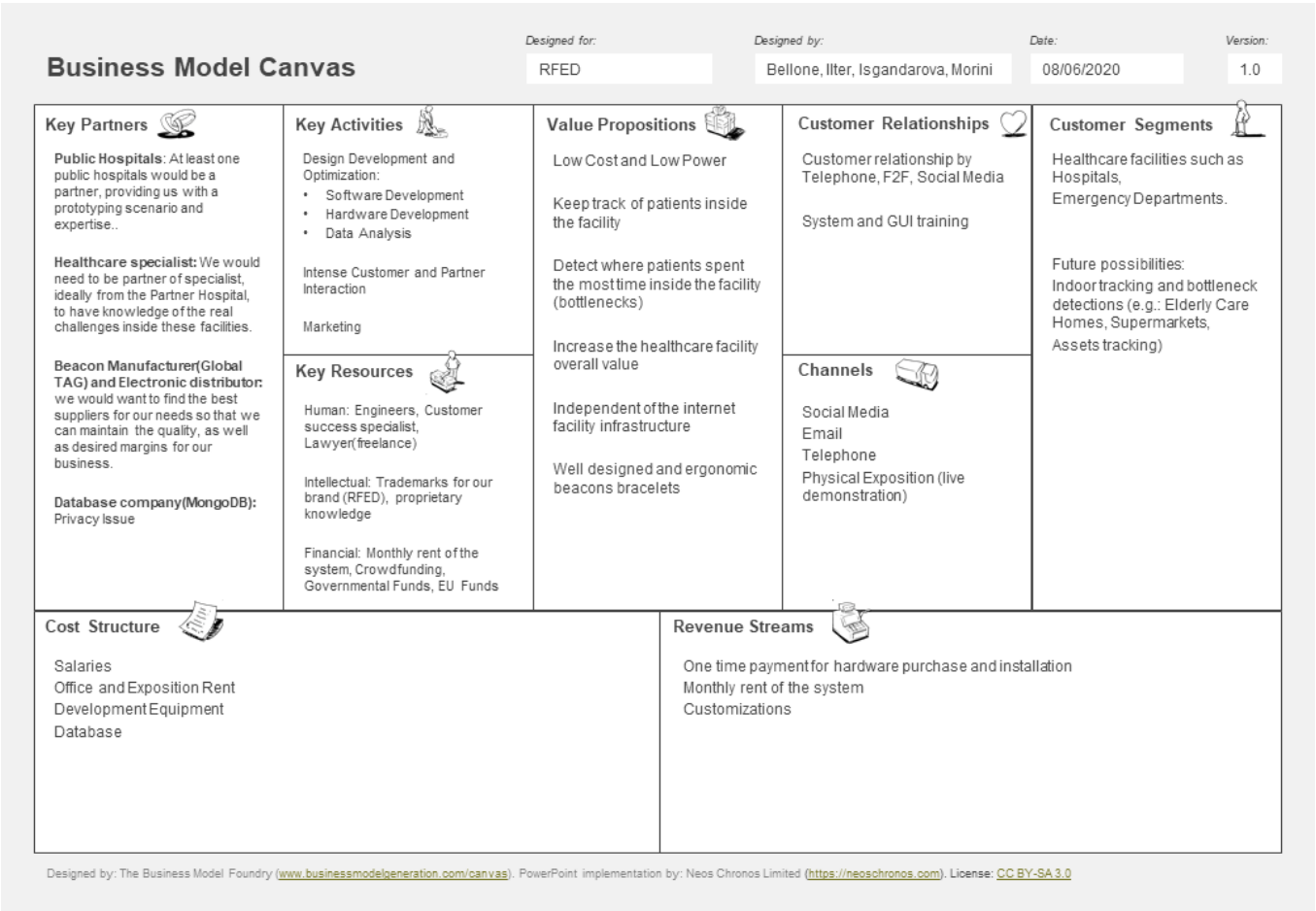



Figure 20: Canvas Business Model

# 5 Conclusion and Future Works

In this report, after the definition of the problem and the analysis of previous works, the RFED architecture has been defined (section 1.3). The objective of this project is to build a prototype that can track patients in each zone of an ED to aid the identification of bottlenecks of the system in an inexpensive way. The experiments that have been carried out have shown that the developed prototype installed in a simple environment brings satisfying results, although more tests should be performed to have reliable statistics. Said that, the patients' paths are always correctly identified and the evaluated duration in each room always with acceptable errors.

Future testing and simulation is needed to a scenario in which the number of rooms and patients will be larger, since the network could show limitations and bottlenecks.

A more detailed comparison of the project objectives discussed by using the MoSCoW method and the actual project delivery is presented in Table 6.

The main difference between the implemented project and the proposed one is the local rather than cloud storage. One of the possible future implementations of the project can be to adopt the cloud storage (e.g. MongoDB Atlas) together with proper security strategies. Another future implementation could be the integration of the sensor data (e.g. temperature, humidity and acceleration) which are already available in the chosen BLE beacon bracelets in order to forward other valid information together with the patient's location. In order to further integrate patients' information that are sent over the network, patients' anamnesis could also be inserted in the data flow, with the goal of a faster and accurate fetch of medical histories.

| **Project Goals** | **Implemented Project** |
|---|---|
| MUST (M)<br>• track the patients in checkpoints, collecting the data into a database<br>• avoid interference and be harmless<br>• respect the privacy | • collection of indoor location data is performed by the proposed prototype<br>• the devices do not interfere with medical equipment given their low power transmission and are completely harmless to the patient<br>• the prototype cannot guarantee compliancy with privacy issues at the moment, but it has all the capabilities to implement them |
| SHOULD(S)<br>• provide data mining | • database structure allowing to perform data mining |
| COULD (C)<br>• has a GUI<br>• Live Migration on the Cloud | • provides a GUI to insert and remove patients in the database<br>• local storage due to budget constraints |
| WON'T (W)<br>• have a mobile application | • doesn't have a mobile application |

Table 6: The comparison between proposed and delivered project

Eventually, in order to evaluate the cost-efficiency of the implemented project, the Bill of Materials (BOM) is calculated in Figure 21 for an example system. The total cost depends on the dimensions of the infrastructure and the number of patients that are going to be tracked.

## BILL OF MATERIALS(BOM)

**Product: RFED Minimum System**

**Quantity: 1**

**Details: 5 checkpoints. (5 rooms (±15m²))**

| COMPONENT | DESCRIPTION | BASE QTY | COST PER UNIT | Reference Model |
|---|---|---|---|---|
| ESP32DevKitC | SoC Dev Board | 7+ | € 9.00 | https://www.espressif.com/en/products/devkits/esp32-devkitc/overview |
| Wall Mount AC Adapters | <500m A, 5V | 7+ | € 6.72 | e.g. mouser |
| USB A/micro B Cable | <500m A | 7+ | € 2.15 | e.g. mouser |
| Raspberry Pi 4 Desktop kit | Gateway | 1 | € 129.90 | https://www.raspberrypi.org/products/raspberry-pi-4-desktop-kit/ |
| Global Tag BLE | 20 Beacons and 20 Bracelets | 1* | € 300.00 | https://www.global-tag.com/ble-beacon-kit/ |
| Mongo DB Atlas | Storage up to 10 GB/2 GB RAM | 1 | € 57/month | https://www.mongodb.com/pricing |
| | | | | |
| | | **TOTAL COST:** | € 612.00 | |
| **For each additional checkpoint add 1 QTY of +.** | | | | |
| *** Calculate one beacon and one bracelet per patient.** | | | | |
| **One check point is needed for every ±10m.** | | | | |

Figure 21: Bill of Materials

The documentation and the codes implemented for the RFED project can be found at this GitHub link: `https://github.com/vcmorini/rfed`.

# References


[1] E. Drazen and J. Rhoads, "Using tracking tools to improve patient flow in hospitals", *California Health Care Foundation*, p. 21, Library Catalog: www.chcf.org. [Online]. Available: https://www.chcf.org/publication/using-tracking-tools-to-improve-patient-flow-in-hospitals/ (visited on 05/15/2020).

[2] D. Dardari, P. Closas, and P. M. Djuric, "Indoor Tracking: Theory, Methods, and Technologies", en, *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, Apr. 2015, ISSN: 0018-9545, 1939-9359. DOI: 10.1109/TVT.2015.2403868. [Online]. Available: http://ieeexplore.ieee.org/document/7042271/ (visited on 05/12/2020).

[3] L. Gholamhosseini, F. Sadoughi, and A. Safaei, "Hospital real-time location system (a practical approach in healthcare): A narrative review article", vol. 48, ISSN: 2251-6085. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6500521/.

[4] M. P., "Moscow method: For setting requirements in order of priority", 2017. [Online]. Available: https://www.toolshero.com/project-management/moscow-method/ (visited on 05/12/2020).

[5] S. Yoo, S. Kim, E. Kim, E. Jung, K.-H. Lee, and H. Hwang, "Real-time location system-based asset tracking in the healthcare field: Lessons learned from a feasibility study", en, *BMC Medical Informatics and Decision Making*, vol. 18, no. 1, p. 80, Dec. 2018, ISSN: 1472-6947. DOI: 10.1186/s12911-018-0656-0. [Online]. Available: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-018-0656-0 (visited on 05/12/2020).

[6] C. Fernandez-Llatas, A. Lizondo, E. Monton, J.-M. Benedi, and V. Traver, "Process mining methodology for health process tracking using real-time indoor location systems", *Sensors*, vol. 15, no. 12, pp. 29 821–29 840, Nov. 30, 2015, ISSN: 1424-8220. DOI: 10.3390/s151229769. [Online]. Available: http://www.mdpi.com/1424-8220/15/12/29769 (visited on 05/11/2020).

[7] L. Kinsman, T. Rotter, E. James, P. Snow, and J. Willis, "What is a clinical pathway? Development of a definition to inform the debate", *BMC Medicine*, vol. 8, p. 31, May 2010, ISSN: 1741-7015. DOI: 10.1186/1741-7015-8-31. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2893088/ (visited on 05/13/2020).

[8] *MYSPHERA, IoT & RTLS for Healthcare*, en, Library Catalog: www.mysphera.com. [Online]. Available: https://www.mysphera.com (visited on 05/13/2020).

[9] F. Chetouane, "An overview on rfid technology instruction and application", *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 382–387, 2015, 15th IFAC Symposium onInformation Control Problems inManufacturing, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.06.111. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S240589631500350X.

[10] M. Chellappa, S. Madasamy, and R. Prabakaran, "Study on zigbee technology", Apr. 2011, pp. 297–301, ISBN: 978-1-4244-8678-6. DOI: 10.1109/ICECTECH.2011.5942102.

[11] *Ethernet | computer networking technology*, en, Library Catalog: www.britannica.com. [Online]. Available: https://www.britannica.com/technology/Ethernet (visited on 05/13/2020).

[12] C. Fernandez-Llatas, B. Valdivieso, V. Traver, and J. Benedi, "Using process mining for automatic support of clinical pathways design", *Methods in molecular biology (Clifton, N.J.)*, vol. 1246, pp. 79–88, Nov. 2015. DOI: 10.1007/978-1-4939-1985-7_5.

[13] C. Gomez, J. Oller Bosch, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology", *Sensors (Basel, Switzerland)*, vol. 12, pp. 11 734–53, Dec. 2012. DOI: 10.3390/s120911734.

[14] *Wi-Fi | networking technology*, en, Library Catalog: www.britannica.com. [Online]. Available: https://www.britannica.com/technology/Wi-Fi (visited on 05/14/2020).

[15] *Track medical samples using qr codes and a mobile app*, en-US. [Online]. Available: https://www.gocodes.com/track-medical-samples-using-qr-codes-and-a-mobile-app.

[16] *Real time location system (rtls) for hospitals*, en-US. [Online]. Available: https://www.centrak.com/products//%20Real-time-location-services.

[17] *Iot for healthcare*, en-US. [Online]. Available: http://www.bordatech.com/tracking_rtls.html.

[18] E. Ferro and F. Potorti, "Bluetooth and wi-fi wireless protocols: A survey and a comparison", *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.

[19] J. Č. Gambiroža, T. Mastelić, P. Šolić, and M. Čagalj, "Capacity in lorawan networks: Challenges and opportunities", in *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, 2019, pp. 1–6.

[20] D. Soni and A. Makwana, "A survey on mqtt: A protocol of internet of things(iot)", Apr. 2017.

[21] S. Mukherjee, "The battle between nosql databases and rdbms", May 2019. DOI: 10.15680/IJIRSET.2019.0805107.

[22] *Nosql databases explained*. [Online]. Available: `https://riak.com/resources/nosql-databases/index.html?p=9937.html` (visited on 05/25/2020).

[23] H. Krishnan, M. Elayidom, and T. Santhanakrishnan, "Mongodb – a comparison with nosql databases", *International Journal of Scientific and Engineering Research*, vol. 7, pp. 1035–1037, May 2016.

[24] (). General Data Protection Regulation (GDPR) – Official Legal Text. en-US. Library Catalog: gdpr-info.eu, [Online]. Available: `https://gdpr-info.eu/` (visited on 05/26/2020).

[25] H. K. Lee, T. Malkin, and E. Nahum, "Cryptographic Strength of SSL/TLS Servers: Current and Recent Practices", en, p. 9,

[26] *Security — MongoDB Manual*, en, Library Catalog: docs.mongodb.com. [Online]. Available: `https://docs.mongodb.com/manual/security` (visited on 05/27/2020).

[27] "The Changing Face of Global Data Privacy: GDPR, CCPA, and Beyond.", en, *A MongoDB White Paper*, p. 20, 2019. (visited on 03/03/2020).