1 2 9 0

UNIVERSIDADE Ð COIMBRA

André Cristóvão Ferreira da Silva

TeleAlert

Telemonitoring based on IoT and Location Based Services

Dissertation in the context of the master's in informatics engineering, specialization in Software Engineering advised by Telma Mota and Teacher Mário Zenha Rela and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2023

Acknowledgments

I would like to take this moment to express my heartfelt gratitude to the individuals who have played a pivotal role in my academic journey, particularly during my college years. These years were a transformative phase in my life, and I owe my accomplishments to the unwavering support of those who stood by me.

Foremost, I want to extend my deep appreciation to my family, whose constant encouragement and provision of the necessary conditions paved the way for my educational success. Their continuous support and dedication have been instrumental in shaping my path. To my parents, sister, and grandparents my most sincere thanks.

To my circle of friends, especially Mário, I owe a debt of gratitude for being my steadfast companion throughout my academic pursuit. Your presence, whether during moments of triumph or adversity, has provided not only solace but also a sense of camaraderie that made the college experience truly memorable. Your ability to strike a harmonious balance between offering much-needed distractions and being pillars of support reflects the strength of our enduring friendships.

I wish to acknowledge Teacher Mário Zenha Rela, whose revisions and invaluable guidance significantly improved this dissertation. Your insights and recommendations have contributed immensely to the quality of my academic work.

I am also deeply thankful to Altice Labs, particularly Telma Mota, for suggesting the dissertation topic and for the trust and guidance provided during the challenging journey of my academic course. Additionally, I would like to thank the team, Medeiros, Ricardo, and Bruno Miguel, who helped me through the whole process.

Finally, my heartfelt appreciation goes to my girlfriend, Mariana, who has been a constant source of support and encouragement throughout my academic endeavors. Her unwavering belief in me, her ability to lift me when I faltered, and her enduring motivation have been the cornerstone of my personal and academic growth.

In conclusion, these college years have been a transformative chapter in my life, and I am profoundly grateful to each and every one of you for your unwavering belief in me. Your support has been instrumental in my academic achievements and personal growth, and I am deeply thankful for your presence on this journey.

Abstract

With the advancements in all branches of medicine, the average life expectancy is now higher than ever. However, this increase also entails some negatives since with advancing age, diseases arise that severely impact people's quality of life. Therefore, it is important to keep expanding a digitalizing the healthcare system by implementing solutions that help people take care of their well-being, either on their own or with the help of their loved ones. It is important to keep improving the care flow and make it more efficient for both patients and caregivers.

The burden on national health services and the lack of medical professionals reinforced the need to find alternatives to keep people away from medical centers while maintaining effective supervision. Thus, eHealth/Assisted Living has been growing with the need to improve quality of life.

Solutions like teleconsultation and telemonitoring already allow formal and informal caregivers to keep up with the status of their patients without the need to be always present; these services make it viable to detect risk situations in advance and act on them as quickly as possible. While monitoring vital signs (clinical telemonitoring) has become increasingly important, tracking personal information such as location, quality of steps, falls, routines and daily activities (non-clinical) can also help detect perilous situations.

In this context, Altice Labs' work proposal intends to specify, and implement a home service that offers safety and comfort to people in vulnerable situations (e.g., old age, reduced mobility, light dementia) by using non-clinical data collected by IoT devices installed in their homes (e.g., smart plugs and bulbs, motion detectors, door/window sensors), location information (indoor and outdoor) and other related data. This service aims to progressively increase the safety, prevention, and autonomy of people with health limitations and advanced age. The objective is to explore several telemonitoring scenarios using different devices to create the safest environment possible. To this end, informal caregivers (e.g., relatives) will be considered responsible for monitoring the people in their care, and as such, they will receive data collected from their devices, as well as notifications and alerts, that allow them to act accordingly as quickly as they can. This will be materialized in a web application called TeleAlert composed of multiple microservices that collect non-clinical data from multiple external sources over a public cloud following a B2C business model.

Keywords

Telelocation, Telemonitoring, Assisted Living, Alerts, eHealth

Resumo

Com o progresso de todas as áreas da medicina, a esperança média de vida é agora maior do que nunca. No entanto, este aumento também acarreta alguns contras, pois com o avançar da idade surgem doenças que afetam gravemente a qualidade de vida das pessoas. É, por isso, importante continuar a expandir e a digitalizar o sistema de saúde, implementando soluções que ajudem pessoas a manter o seu bem-estar, seja por conta própria ou com a ajuda dos seus entes queridos. É importante continuar a melhorar o fluxo de cuidados e torná-lo mais eficiente para pacientes e cuidadores.

A sobrecarga nos serviços nacionais de saúde e a falta de profissionais médicos reforçaram a necessidade de encontrar alternativas para manter as pessoas afastadas dos centros médicos, mantendo uma boa supervisão. Assim, a eHealth/Assisted Living tem vindo a crescer com a vontade de melhorar a qualidade de vida.

Soluções como a teleconsulta e a telemonitorização já permitem a cuidadores formais e informais acompanhar o estado dos seus pacientes sem a necessidade de estarem sempre presentes. Estes serviços tornam viável a deteção de situações de risco antecipadamente e que se aja sobre as mesmas o mais rapidamente possível. Embora a monitorização de sinais vitais (telemonitorização clínica) se tenha tornado cada vez mais marcante, o rastreamento de informações pessoais como localização, qualidade dos passos, quedas, rotinas e atividades diárias (não-clínicas) pode, igualmente, ajudar a detetar situações perigosas.

Neste contexto, a proposta de trabalho da Altice Labs visa especificar e implementar um serviço domiciliário que ofereça segurança e conforto a pessoas em situações vulneráveis (por exemplo, idade avançada, mobilidade reduzida, demência ligeira) utilizando dados não-clínicos recolhidos por dispositivos IoT instalados nas suas casas (por exemplo, tomadas e lâmpadas inteligentes, detetores de movimento, sensores de portas/janelas), informações de localização (interior e exterior) e outros dados relacionados. Este serviço visa aumentar progressivamente a segurança, prevenção e autonomia de pessoas com limitações de saúde e idade avançada. O objetivo é explorar vários cenários de telemonitorização usando diferentes dispositivos para criar o ambiente mais seguro possível. Para isso, os cuidadores informais (por exemplo, parentes) serão considerados responsáveis pela monitorização das pessoas ao seu cuidado, e como tal, receberão os dados recolhidos dos seus dispositivos, bem como notificações e alertas, que lhes permitam agir em conformidade o mais rapidamente possível. Este conceito será materializado numa aplicação web chamada TeleAlert composta por vários microserviços que recolhem dados não-clínicos de várias fontes externas através de uma nuvem pública seguindo um modelo de negócio B2C.

Palavras-chave

Telelocalização, Telemonitorização, Assisted Living, Alertas, eHealth

Table of Contents

Acronyms

AI	Artificial Intelligence
ALB	Altice Labs
API	Application Programming Interface
ACID	Atomicity, Consistency, Isolation, Durability
B2B	Business to Business
B2B2C	Business to Business to Consumer
B2C	Business to Consumer
BMI	Body Mass Index
C4	Context, Containers, Components, Code
CDS	Clinical Decision Making
DTOs	Data Transfer Objects
ECG	Electrocardiogram
EHTEL	European Health Telematics Association
EMR	Electronic medical records
GPS	Global Positioning System
ICT	Information and Communications Technology
IMIA	International Medical Informatics Association
IoT	Internet of Things
IPS	Indoor Positioning System
I&D	Investigation and Development
JPA	Java Persistence API
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
M&E	Monitoring and Evaluation
ML	Machine Learning
MoSCoW	Must, Should, Could, Won't have
R&D	Research and Development
RFID	Radio-Frequency Identification
ROI	Return of Investment
UI	User Interface
UX	User Experience

WPS	Wi-Fi Positioning System

List of Figures

List of Tables

1 Introduction

Population aging due to low fertility rates and high life expectancy is a haunting problem for most developed countries. With the evolution of society and medicine, people have been able to live longer than ever before as treatments, hygiene products, and items of first necessity are now available to a wider range of people.

Also, in some countries, fertility rates are lower than ever, since people, in general, are reducing the number of kids they have. For example, China implemented, in the past, strong measures to control population growth even though those policies have now been relieved. In Portugal, families voluntarily postpone having kids due to their professional duties or the lack of financial conditions to raise them.

Even though higher life expectancy has some benefits, it also raises concerns since the age increase has not come with a commensurate improvement in quality of life. Living longer also implies a higher risk of getting serious health problems, such as chronic diseases (e.g., dementia, arthritis, and heart disease) and several types of impairments (e.g., vision and hearing loss, mobility limitations).

This means that more and more people require constant supervision and care, leading to overburdened healthcare systems. Technology has the potential to help patients and facilitate caregivers to make better decisions and manage medical conditions beyond the clinical settings. Solutions such as smart homes, portable devices, and wearables can help providers deliver care to a growing population of older adults.

eHealth is becoming the key factor in the evolution and future of healthcare. This means that health services can now be delivered through the Internet by offering health professionals an easier way to accompany the older population and make informed decisions based on data collected from a variety of sources and devices.

Given this context and the need to prevent and guarantee the well-being of the population in general and older adults in particular, this dissertation will focus on developing monitoring services based on complementary information to clinic data (since there are already several commercial products in the clinical area, including SmartAL, a product from Altice Labs). Information, such as location and IoT data (for ex: invasion sensing, smart home plugs, bulb activity, motion, and fall detection), will be collected and processed to develop services essentially devoted to older adults and their informal caregivers (e.g., family members). Information from different devices will feed decision engines that should be able to evaluate specific conditions and detect abnormal behaviors and risky situations. Under a more holistic well-being concept, patients and caregivers will benefit from these services in addition to pure clinical-based applications. The next sections will explain the concept and the work to be done in more detail.

1.1 Context

eHealth can be defined as the usage of information and telecommunication technology to enhance and allow better delivery of healthcare services through the Internet [1]. The concept represents virtually everything related to medicine and computers, so it ranges from online check-ups to devices that collect information about people to be analyzed by doctors. So, under this concept, Teleconsultation and Telemonitoring, are two fundamental tools of healthcare that allow a more effective patient follow-up [2].

Telehealth is especially important when it comes to older people. It provides this group with a lot of benefits; concerning teleconsultation, it allows people with reduced mobility or another status of fragility to have access to a specialist without having to go to the hospital to get an appointment [3]; complementarily, telemonitoring allows patients to send data directly to their doctors using portable devices that can easily be used at home and measure vital signs, such as temperature and blood pressure. Doctors can then get a better grasp of the patient's problems from a long distance. The collected data can be used by these professionals to make informed decisions in critical situations. Some telemonitoring solutions can already warn doctors automatically when there is a problem using machine learning and Artificial Intelligence. These algorithms are also used to analyze the collected data and suggest some actions while always leaving the responsibility of the final decisions to the physicians to avoid ethical concerns [4].

Life expectancy has risen by 3 years for every 10 years that have passed since 1950, with some exceptions like the '90s when AIDS cut short many lives (mainly in Africa) and the Soviet Union collapse that also led to a high mortality rate [5]. In 2022, the share of the global population aged 65 years or above is around ten percent [6]. Figure 1 shows a map of the average life expectancy by country in 2019 [7].

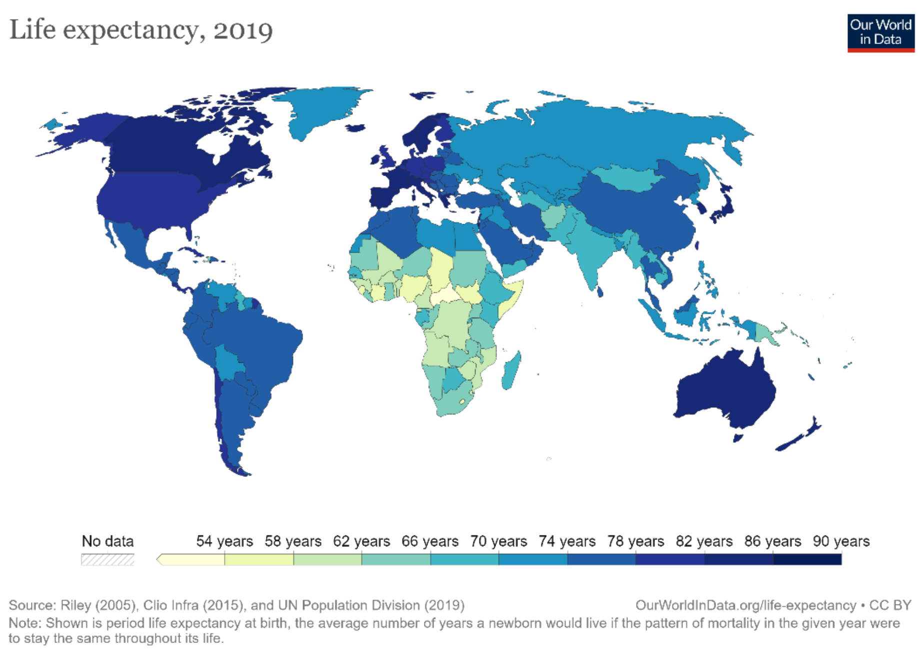


Figure 1: Life Expectancy World Map in 2019 [7]

As a result of the growth in life expectancy, technologies applied to eHealth become even more important, as in recent years the quality of life has not improved at the same rate [8].

The reality is that over time older adults are more likely to develop diseases that impact their quality of life; some of these diseases greatly affect their mobility, such as arthritis; others like sarcopenia and osteoporosis make people more likely to be seriously injured due to muscle weakness and bone fragility (osteoporotic fractures), respectively; diseases like dementia affect people's ability to interact normally with the world around them and limit a person's ability to think and learn; chronic diseases such as hypertension and diabetes also grow on incidence with age and are among the main causes of death; and also, common problems like cataracts and poor hearing prevent people from perceiving the world around them accurately. In short, many limiting conditions can occur with age [8] [9].

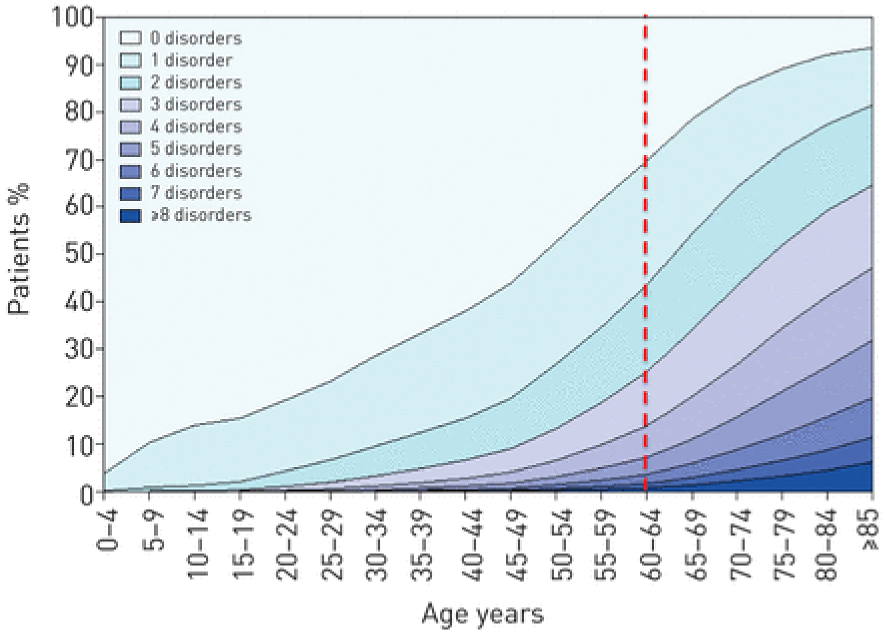


Figure 2: Number of chronic comorbidities by age stratum [10]

As can be seen in the graph in Figure 2, 60% of patients above 60 years old have 1 chronic disorder if not more. When reaching 85-year-old only about 5% have none of these diseases but a similar percentage has 8 or more of those conditions. However, the same medical progress responsible for increasing life expectancy can also help to minimize the resulting problems. This raises the important question of what type of medical research we should invest in in the future [8]. Committing to prevention and improvement of physical well-being has some obvious advantages, but it is not enough. A person with a condition is primarily focused on their pain or discomfort and not on their recovery. So, the common result is to have an unhappy or depressed person, not someone on the way to becoming healthy. Healthier people are usually more motivated and productive and have better relationships with others around them. On a wider scale, the focus on preventive well-being can extend longevity, mitigate mental health problems, and may also help to improve the results of future clinical and behavioral interventions [11].

A good quality of life can be achieved by either slowing the aging of the body or making changes in the environment around it to make life either less complicated or less of a hardship [8]. The first is achieved by investing in medicine to discover new treatments and new technologies that help either with surgeries and procedures or with recovery. Concerning the second, building devices and software solutions can help with prevention, treatment, and recovery.

1.2 Motivation

The work of this dissertation was proposed by **Altice Labs**, the research and development branch of the Altice group. Altice Labs is a telecommunications company that develops innovative products and services for ICT markets. It also participates in international and national projects in collaboration with universities and other I&D institutes to provide knowledge through internships and trainees to become future professionals[1].

[1] Altice Labs - https://www.linkedin.com/company/altice-labs

As part of Altice Labs's portfolio, there are two Telehealth products worth mentioning: Medigraf [12] has been providing teleconsultation services to several institutions for more than twenty years, and a more recent and advanced product, SmartAL [13] provides telemonitoring services.

SmartAL[2] consists of a technological ecosystem that aims to simplify people's daily lives from both a health and social point of view. It is a telemonitoring tool where end-users (patients, family members, and caregivers) may follow in real time the health status of the patients. It allows the collection of information from clinical and non-clinical devices (e.g., oximeter, personal band) and other data sources such as questionnaires. It gives an overview of the patient's health status to both patients and caregivers. The collected data (e.g., vital signs) is then compared with thresholds previously configured by the physicians and if the values are outside the limits, alerts are sent, so that actions can be taken by the health professionals according to their severity (e.g., schedule a teleconsultation, consult a specialist, prescribe medication). SmartAL is available as a mobile app, TV app, and Web App and it runs on practically all kinds of devices. It also adapts its functionality to the profile of the user to meet different needs, including patients, doctors, informal caregivers, administrators, installers, volunteers, and others.

SmartAL operates in the areas of prevention, control, and monitoring of chronic diseases, recovery and medium-term treatments, and other disabling conditions (e.g., old age). Currently, the product is available on the market as a B2B model, meaning it is sold to institutions, such as hospitals, nursing homes, and municipalities that then make the services available to their patients/constituents. However, following the last **uberization**[3] trends, the B2C solution is already under development as Proof of Concept (PoC), so that telehealth services may be offered directly to the client/consumer who will also be the end-user. This dissertation will focus on developing services that will integrate with this new B2C architecture which is being designed and implemented according to the most advanced cloud principles and technologies.

The new services to be conceived and developed came with the need to create well-being functionalities capable of complementing clinical offers also under development by Altice Labs. Therefore, the proposal of Altice Labs aims at creating applications capable of helping informal caregivers and relatives take care of people who need supervision, such as elders and old adults with mild health problems (e.g., disorientation, poor mobility skills, light dementia). The B2C SmartAL prototype already includes teleconsultation and outdoor location services and the work developed under this dissertation will be integrated into that prototype to create a larger and more diverse bundle of services.

1.3 Objectives

The main objective of this dissertation is to conceive and develop a new service called TeleAlert under a Confort@Home[4] concept. This concept intends to create a well-being offer composed of modular services that can be used as a bundle or individually. The main goal of this offer is to keep the end-users safe and healthy.

[2] SmartAL - https://www.alticelabs.com/wp-content/uploads/2022/01/BR_SMARTAL_ALB_EN.pdf
[3] A business model in which services are offered on demand through direct contact between a customer and a supplier, usually via mobile technology.
[4] Concept created by AlticeLabs that defines comfort and well-being experienced within the confines of one's home.

To accomplish this objective, what is proposed is to collect data from different external systems/devices and based on this, evaluate the routines of the end-users, and apply service logic to identify risky situations employing rules and/or analyzing patterns.

There are four envisaged services to be developed under this dissertation that will integrate the TeleAlert offer:

i. A collector of indoor location-related data such as positioning, walking speeds, and step intensity (the external system is based on an indoor non-invasive optic-sensor solution that is being developed by IT Aveiro[5] - Instituto de Telecomunicações de Aveiro, under the scope of a European project called SAFEHOME[6]);
ii. A collector of outdoor location-related data that will receive information, such as GPS coordinates, from a gateway that interacts with a smartwatch.
iii. A home IoT data collector that will consider data coming from an API offered by an Altice Labs solution called SmartHome[7] devices such as smart plugs and bulbs, motion, and door/window status detectors.
iv. A decision engine that will receive information from the three collectors and will apply rules and/or algorithms to evaluate routines/patterns to identify unusual events, abnormal situations, and risks to trigger alerts accordingly. Furthermore, the output of each device/data source will be carefully analyzed and explored to achieve the best result. An investigation will be conducted to decide which is the better approach; several rule engines will be evaluated. In the future, machine learning algorithms can also be used, exclusively or in conjunction for analysis.

Based on this analysis, alerts will be triggered and sent to the caregiver (in the future, may also be sent to the patient). This information can then be combined with other B2C applications, such as clinical telemonitoring apps to give the caregiver a complete view of the person's status under analysis. Be noted that clinical data can also be used to perform this type of analysis, but this work will focus **only** on environmental information as previously explained and as represented in Figure 3.

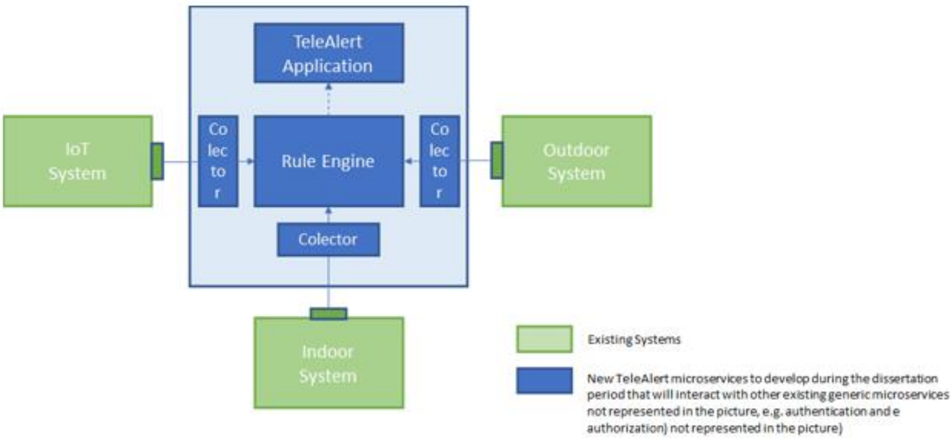


Figure 3: High-Level Architecture of TeleAlert

[5] https://www.it.pt/ITSites/Index/3
[6] https://safe-home.care/
[7] https://www.alticelabs.com/products/SmartHome/

To summarize, this internship will develop a group of microservices that will compose the service named **TeleAlert.** This service will use information collected through non-clinical devices to analyze daily routines and detect potential risk situations related to the people under supervision. As a result, alerts will be sent to their informal caregivers. Rules will be configured statically in the rule engine based on the habits of the person or group of people. In the future, they can be inferred using machine learning mechanisms.

The application to be developed in this dissertation works as **proof of concept** that complementary information such as location and IoT data can also contribute to detecting possible disturbances related to the well-being of the people in care. The developed microservices will integrate a larger existent microservice architecture developed by Altice Labs and hosted in a public cloud infrastructure.

1.4 Results

At the end of this dissertation, the following results were achieved:

- A detailed state of the art around the themes of telemonitoring, indoor, outdoor, and IoT devices, and existing TeleAlert competitors.
- Documentation around the specification and modeling of the system including personas, use cases, functional and non-functional requirements, database models, wireframes, and changes in the existing SmartAL architecture.
- The system was developed according to the must-have and should-have requirements.
- The system was tested through manual testing.
- The developed application was successfully deployed, making it accessible for testing and evaluation.
- The system's indoor component was validated in a realistic scenario of use. The outdoor component was still validated but in a different scenario.

The internship concluded with the creation of a proof of concept that shows its viability in the real world. This encompassed developing collectors and a rule server that sends information and alerts to the frontend to be visualized and help caregivers with decision-making.

The application not only encompasses all the demanded functionalities but also integrates supplementary features classified as should-have.

While the developed application fulfills the immediate project needs and objectives, it is recognized that conducting more comprehensive testing could have bolstered its validity. Moreover, opportunities for future enhancements and refinements remain. These include delving into new functionalities, enriching the user interface, and introducing further notifications. Also, introducing Artificial Intelligence (AI)/ Machine Learning (ML) algorithms to better shape the alert conditions.

The internship has established the groundwork for forthcoming development, and the pinpointed areas for enhancement provide a well-defined path for advancing the application's capabilities and user experience.

1.5 Document Structure

This chapter served as an introduction to this dissertation. During this chapter, the motivation and context beyond the project as well as the results and document structure are on display allowing for an overall view of the full project.

Chapter 2 will dive into the methodology adopted during the internship as well as the work planning, tasks, the risks associated with the implementation of the project, and the thresholds of success.

After, in chapter 3, the state of the art will be on full display. The topics surrounding business models for eHealth systems, the Internet of Things, positioning devices, telemonitoring, and finally assisted decision-making systems will be covered in this section.

The following chapter will focus on background information and the state of practice of the system already built by Altice Labs showing the services it provides and its current architecture.

The fifth chapter will start with the specification of the new microservices by introducing the use cases, the requirements of the application to be developed, and updated architecture and technologies.

Finally, before the conclusion, chapters 6 and 7 detail the process of implementation, integration, deployment, testing, and validation.

1.6 Summary

The current chapter serves as a light introduction to the work being done, giving an initial overview of the problem at hand while providing context on the issue of high life expectancy, age-related diseases, and ways of achieving a better quality of life. It also displays the origin, motivation, and objectives of the dissertation, introducing Altice Labs and briefly the prototype of which the work of this dissertation will become a part, SmartAL B2C.

Moving forward, it is expected to dive further into the themes underlying this dissertation and what approaches were used to develop the intended system.

2 Work Plan

In this section, the work plan for both semesters is explained in detail. Along with it, the risk assessment and thresholds of success are also showcased.

2.1 First Semester

The project started with the planning phase. This included the production of the pre-dissertation document and the necessary investigation of the topics surrounding the work to be included in the state-of-the-art. After that, started the requirement-gathering phase, where collaborative discussions with the team took place to identify the essential functionalities and features of the system. These requirements served as a foundation for the subsequent development activities. Regular weekly meetings were held with the Altice Labs team to discuss progress, adapt to new requirements, and address challenges encountered.

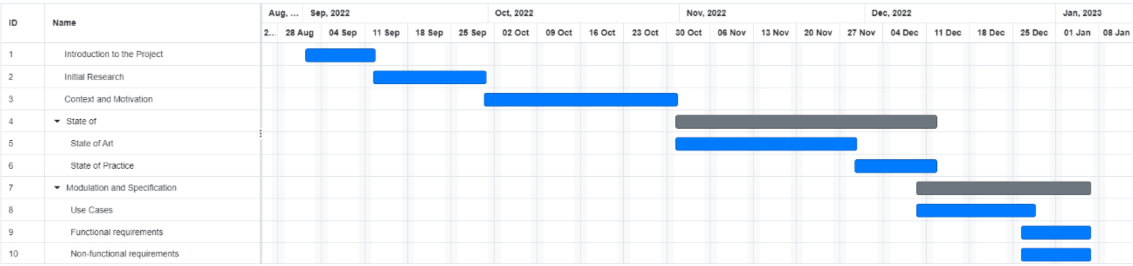


Figure 4: First Semester Gantt Chart

2.2 Second Semester

Following the initial requirements-gathering phase, the project proceeded with a more iterative and agile approach. Backend microservices, including the collectors of information and the rule engine, were developed according to the established requirements and specifications. Once the backend microservices were completed, the project proceeded with the application design and frontend development phase. These components were developed using suitable technologies to create an intuitive user interface.

The integration phase followed the backend and frontend development. First, the collectors and the rule engine backend microservices were integrated to form a cohesive system and then the web application was placed on top. Attention was given to ensuring seamless communication, data accuracy, and security between the components. Integration with other necessary prebuilt services also took place during this phase.

As the project progressed, the team embraced an agile mindset, allowing for flexibility and adaptation. New requirements were evaluated, prioritized, and incorporated into the development process. This iterative approach, combined with the regular meetings, facilitated the continuous evolution of the system to meet changing needs, including the complete restructuring of the outdoor collector service to allow a new device integration. Moving forward, the validation phase was at hand. Real data was collected for a month from a volunteer’s home. Indoor sensors and IoT devices were installed to monitor an old adult. Before, during, and after this period several validation activities were conducted to verify that the system fulfilled the intended objectives and met the stakeholders' requirements. It is also important to know that the whole development was accompanied by manual testing.

Finally, this document was concluded and thoroughly revised by the internship team.

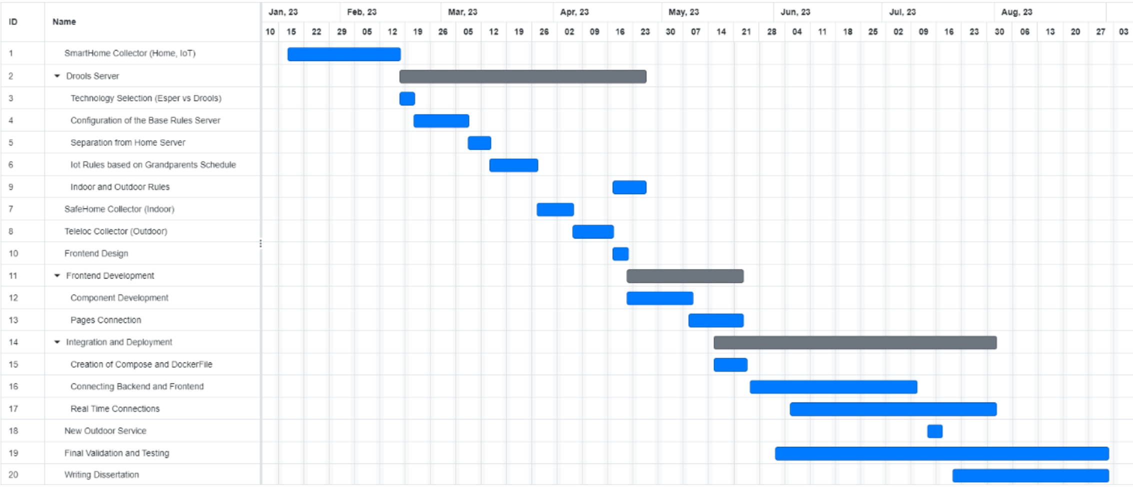


Figure 5: Second Semester Gantt Chart

2.3 Risk Assessment

Risk Analysis is a proven way of identifying and assessing factors that could negatively affect the success of either a business or a project. It allows one to examine the risks that the team and organization face and helps decide how to move forward, as well as how to solve possible problems through mitigation actions.

The following risks were identified for the project:

1. The TeleAlert solution depends on external systems, some of them under development; so, delays on these systems may cause delays in the overall result.
2. The TeleAlert solution depends on already existing SmartAL microservices that might need changes to accommodate the new development. Those changes may cause delays.
3. Poor knowledge of frontend technologies may lead to longer development times.
4. Lack of a real scenario for validation may lead to inconclusive results.

To classify the risks the matrix in Figure 6 was used. Each risk was evaluated from one to four by likelihood and severity.

*Figure 6: 4*4 Risk Matrix*

The table below shows the values of the likelihood and severity of each risk and the corresponding mitigation plan.

Table 1: Mitigation Plan per Risk

Risk	Mitigation Plan	Likelihood	Severity	Risk Level
1	Frequent communication with the other teams to make sure that what can be advanced is being implemented.	4	3	12
2	Preemptive communication of the changes needed to other services.	3	2	6
3	Use the beginning of the second semester to learn frontend technologies through tutorials and interaction with AlticeLabs's specialized personnel.	1	3	3
4	Finding a person who lives alone to make sure the validation is done in a realistic scenario.	2	4	8

Regarding the first risk, it was successfully mitigated. Due to the nature of the collectors and communication between teams, the delays in development were minimal. The structure of each message was shared as soon as possible allowing for early development of the collectors.

The second risk was not entirely mitigated, or at least as fast as desirable. Even though the changes needed were communicated, my lack of knowledge of the system caused some parts of the services to be restructured a bit too late. Nevertheless, last-minute refinements allowed the system to be fully integrated with the help of the ALB team.

The third risk was successfully mitigated. Tutorials and specialized personnel contributed heavily toward the fast frontend development.

Finally, the last risk was reasonably mitigated since the grandma of a member of the SafeHome team allowed for all the devices to be installed at her house. The validation period was cut short, however, to only a month and only two weeks of data were useful.

2.4 Threshold of Success

The Threshold of Success of a project represents the minimum set of conditions that must be met for the project to be considered successful. If even one of the conditions is not met, the project will not succeed. The following conditions were accorded:

1. All functional requirements classified as must-haves must be developed.
2. All non-functional requirements should be respected.
3. The system should be tested manually and validated in a realistic scenario.
4. Results must be evaluated.

To gauge the level of success and ascertain the accomplishment of goals, specific and well-defined objectives were established. The requirements were broken down into individual tasks, and upon the completion of each task, it underwent evaluation by a designated team member.

In terms of testing, it was agreed that backend testing should be mostly manual as it is not the main focus of the dissertation. Similarly, the testing of the frontend would primarily involve manual procedures, with a specific emphasis on its integration with the backend API. Additionally, the system should be validated in a realistic scenario. For more comprehensive insights into the testing and validation process, please refer to Chapter 7 Testing and Validation.

2.5 Summary

In this chapter, it was covered how the project was planned and developed, the risks involved, and what needs to happen for this work to be considered successful. This helped to achieve a better understanding of the project at hand and allowed for bettering the decision-making process.

The next chapter will expand further on the context of the project while detailing the state-of-the-art concepts and technologies surrounding the project.

3 State-of-the-art

To fully grasp the potential of this project, it is necessary to have a comprehensive overview of the current state of the art and related technologies. By examining their current capabilities and limitations, it will be possible to better understand the potential impact and significance of the work being proposed. In addition, it will also allow identifying potential challenges or opportunities that might come up during development and in the future.

3.1 Introduction to Telemedicine

The first documented instance of telemedicine happened around the year 1897, twenty years after Bell invented the telephone. According to Adam Darkins and Margaret Cary, in that year a doctor diagnosed croup (a disease that affects airways with a distinctive cough) through a simple telephone call and reported the case in a medical journal. However, despite this early appearance, in the next decades, remote diagnostics were reserved for specific expeditions and space missions [14].

During the following centuries, technology started to be applied to medicine. The introduction and development of x-rays, telegraphs, and other similar technologies opened many doors in the world of telemedicine as these allowed for new forms of diagnosis [15]. Also, the use of analog telephone lines during those years enabled the transmission of electrocardiograms (ECGs) between hospitals and patients to contact health professionals more frequently [5].

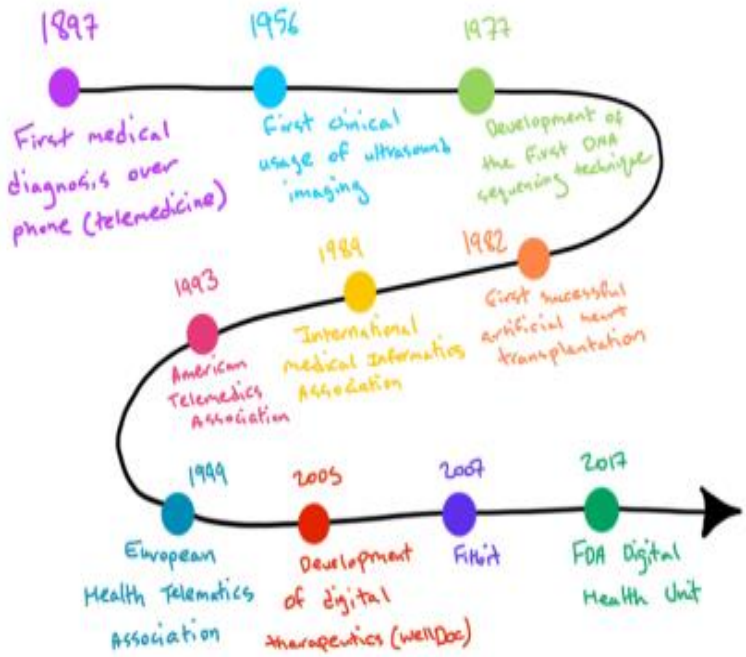


Figure 7: Milestones of Digital Health [65]

At the beginning of the 21st century, another major technological development triggered the area of digital health. The World Wide Web gave users further and easier access to medicine and healthcare. At the same time, therapists started to offer online and digital means to prevent and treat clinical and psychological disorders (online consultation). Furthermore, vital signs, other body measurements, and medical images can now be shared more effectively [15]. As can be seen in Figure 7, Companies like WellDoc[8] were founded in that era and focused on the development of digital therapeutics. Also, FitBit[9] was created at the climax of this period (in 2007) and started to produce wireless fitness devices to monitor vitals, like heart rate and calories burned.

Looking into the future, digitalization in healthcare and the use of new technologies to improve safety and well-being all around the world will continue to increase and improve.

3.2 Business Model

To better know how to target the system of this dissertation it is important to investigate its business model. Most businesses tend to fit into two categories of relationships with their clients: B2B (Business-to-Business), which consists of a business that sells a product to another business company, and B2C (Business-to-Consumer) which focuses on providing a product or a service directly to individual customers [16].

Even though there are other growing types of relationships, like B2B2C (Business-to-Business-to-Consumer) that do business with other business institutions and also mediate their relationship with their customers behaving as a third-party company, these are usually less frequent. This kind of model faces difficulties with high demands because the service must satisfy both the customer and the third party, increasing the complexity of the business relationship and making customer experience management harder [17].

B2C is usually the choice for newer businesses because it is a simpler relationship. First, there is a much bigger possible base of clients. A company might find a lot of possible individual customers in the market but a way lower number of companies ready to negotiate a service [16] [18]. Marketing is also easier for B2C, and they can expect a better and faster ROI (Return on Investment) [19]. A good example of a B2C model is the one used by the Uber[10] company and other similar businesses, which is now being applied to other areas like healthcare.

3.2.1 Service Uberization

Uber is the company that created an application that allows drivers and deliverers to connect directly with customers. Their service allows any person who needs a taxi to get one with just a click. This is a simple business idea that made Uber the biggest taxi company in the world without owning a single car. In 2009, it opened its business and immediately hit 1 million in annualized gross in just 16 months. In 2020, 10 years later it amassed $26.61 billion in gross bookings from its ridesharing business [20].

Uber identified a service with already great demand, and added a technology layer to it, making consumption easier and more convenient for both drivers and passengers. This started a trend of creating on-demand apps that connect the client directly to the provider, known as **Uberization.** This concept has been extended to other industries like housing (Airbnb), deliveries (Amazon), movies and series (Netflix), education, fitness, beauty, and finally healthcare [21].

[8] https://www.welldoc.com/
[9] https://www.fitbit.com/global/eu/home
[10] https://www.uber.com/us/en/about/

3.2.2 Uberization of Healthcare

As mentioned previously, a higher life expectancy led to an increased percentage of the population suffering from chronic ailments. As a direct consequence, there is a bigger-than-ever mismatch between the demand and supply of healthcare professionals. The coronavirus pandemic showed the world how scarce the workforce can be in times when it is most needed. [22]

The virtualization of healthcare services may very well be the solution for this. Just like Uber revolutionized the taxi industry, **Uberization[11]** can revolutionize healthcare if correctly applied. By introducing the usage of technologies like smartphones, the Internet, 5G, mobile apps, and GPS, patients will have more than one way of connecting with health professionals Remote treatment and emergency services can now be provided without patients needing to leave their houses [23].

Uberization can improve the availability of healthcare. There are three main ways this can be achieved. The first approach is to improve access to professionals through methods like online appointments and home doctors. This minimizes unnecessary trips, waiting times, and crowding in healthcare centers. The second method is to intensify the use of healthcare managers. This allows a doctor to monitor vital information to be used for diagnosing, exams, and appointment requests by the patient eliminating human errors in the process. Finally, mobile health will allow the usage of AI-related technologies to diagnose with the assisted intervention of doctors [23].

Many of these approaches can only be achieved with the adoption of smart devices, such as IoT (Internet of Things) devices.

3.3 Internet of Things (IoT)

The term Internet of Things (IoT) first came to use in 1999 by the computer scientist Kevin Ashton when trying to create the first RFID (Radio-Frequency IDentification) chips to track products through the supply chain [24]. The term defines a network of physical objects that includes sensors, cameras, and other technologies to collect and share data with other devices. IoT has become one of the most important technologies of this century because it allows simple communication between people, objects, and processes. It allows connecting everything to the Internet, making sharing and collecting data possible with almost no human intervention. In this way, each interaction with and between these objects can be adjusted automatically [25].

3.3.1 IoT in eHealth

IoT has seen success in many industries, one of them being health care. Due to the complexity and constraints of the industry, it took a long time for innovations to adapt to the needs of the industry, but progress can already be noted in areas like teleconsultation and remote monitoring. Its main benefits reside in enhancing the capabilities of preventive medicine, accelerating patient data processing, reducing error or miscalculation due to human error, better monitoring of medicine intake in patients, and increasing alert of hospital staff [26].

---

[11] "Uberization is replacing middlemen in business with high-tech, such as mobile apps or platforms where you can find a supplier for your demand. The platform facilitates interaction between a customer and the service provider and monitors service quality." https://lectera.com/info/articles/uberization

Many great companies have started investing in IoT. For example, Pfizer[12], best known for its contribution to COVID-19 vaccines, used IoT devices to aid in production and distribution; through IOT sensors they tracked and monitored shipments of the vaccines and guaranteed that the shipments were maintained in safe temperatures. Another example is the Phillips' Capsule Medical Device Information Platform[13] which is being used to connect medical devices and to help feed EMRs (Electronic Medical Records) in hospitals with additional patient information; it also allows for vital signs monitoring and clinical surveillance services. The monitoring system gives the possibility for doctors to access patient data and act on high-risk events more quickly [27].

The most common employment of IoT in health care is remote patient monitoring. Devices can collect information that includes vital signs such as heart rate, glucose, blood pressure, and temperature used for informed decision-making without the patient even being present. Other uses include depression, mood, and other conditions like Parkinson's and Alzheimer's disease monitoring [25] [28].

For these feats to be achieved there are several devices involved, for example, vital data collecting devices (oximeter, glucometer, thermometer, blood pressure monitor), cameras and motion sensors, tags, and RFID chips. Even daily devices like smartwatches and bands can now be used to collect information and visualize it through apps.

In addition, another interesting feature of some of these devices is location detection. They allow not just to keep track of runs and walks using GPS (Global Position System) but also to define geofences and safe places. These features are being used to help monitor people with light physical impairments and mental diseases that affect decision-making (dementia and Alzheimer's) and prevent them from wandering around and getting lost. However, to gain a complete picture of their journey beyond the outdoor environment it is also important to understand how people move within their homes (or institutions) and be able to detect when they are at risk indoors. In this case, IPS (Indoor Location Positioning) should be used as a complement to GPS devices.

In this project, SmartHome IoT devices such as plugs, bulbs, motion detectors, and door/window status sensors along with SAFEHOME optic sensors will be installed in the house of a person to be able to collect real-time data. Also, a simple wearable such as a smartwatch/band will be considered to collect additional information by adding an outdoor component to the system.

3.4 Global Positioning System

The Global Positioning System (GPS) is a satellite-based system that provides users with geographical positioning. The basis of GPS is a set of satellites orbiting the Earth and a network of ground-based stations that monitor these satellites' signals. A typical GPS receiver picks up these signals and uses them to pinpoint its location on Earth [29].

The GPS operates on a simple principle: it uses signals sent from multiple satellites to calculate a receiver's precise location. Each of these satellites sends a signal that includes its location and the time the signal was transmitted. The receiver then calculates the distance to each satellite based on the time it takes for the signal to reach it. With distance measurements from at least twenty-four satellites, the receiver can determine its location using a process

[12] https://www.pfizer.com/
[13] https://capsuletech.com/capsule-platform

known as trilateration. Essentially, each distance measurement provides a sphere of possible locations, with the intersection of these spheres defining the receiver's location [30].

The best part is that it can be used by any device. Most portable devices, like computers, mobile phones, and other smart devices known as wearables already carry that functionality allowing for almost universal access to it.

3.4.1 GPS Applications

The precision and global reach of GPS have made it an essential tool for a diverse range of applications. It is mostly used daily by people all over the world for location and navigation. Just using the internet and GPS connection, people now can use their phones to know where they are and get to anywhere, they want [29].

GPS is also a fundamental tool for geography and cartography-related professions since it can be used for creating detailed maps and carrying out land surveys [31]. It has also been employed for military purposes for a variety of objectives, including guidance for troops, vehicles, and missiles. It also helps coordinate strategies via real-time troop and asset tracking [29] [32].

The same mapping capabilities and real-time updates can also help predict natural disasters such as tsunamis and storms since its use is not restricted to just land [29] [33]. Finally, one of its most important applications that is sometimes overlooked is healthcare.

3.4.2 GPS in Healthcare

There are different ways that GPS has been utilized in the healthcare sector. The first application is centered around patients who require continuous monitoring due to certain conditions that affect their perception of the world around them. GPS can help track their location and send alerts in case they get lost and other emergencies. [34]

This can be especially important in elderly monitoring. GPS-enabled devices can be used to monitor the movements of elderly people, especially those with dementia and Alzheimer’s disease, who might wander and find it difficult to return home. These devices can also provide an emergency call function, enabling the elderly to seek immediate help [34].

Furthermore, GPS can be a great help to the physically disabled, especially when they don't have someone to assist them. GPS locators can help caretakers keep a tab on their location and provide immediate assistance if required [34]. As well as ambulance and paramedic services being able to leverage GPS for routing and navigation during emergencies [35]. It also allows for geo-fencing and safe place creation that permits alerts to be generated when the patient leaves these places in case it gets lost or wanders off.

In conclusion, GPS technology has truly revolutionized various sectors, with its impact permeating healthcare as well. It has significantly improved the safety and efficiency of care for patients, specifically those with high-risk health conditions.

During this project, one of the collectors will obtain information from an outdoor service connected to a smartwatch that besides its GPS location will be able to send georeferenced alerts of outside falls, SOS requests, geofence leaves, low battery, and others.

3.5. Indoor Position Systems

Indoor Positioning Systems (IPS) are characterized by a network of devices that work on a smaller scale than the GPS. In other words, the IPS is used where the GPS cannot locate accurately. GPS lacks precision when it comes to areas like houses or buildings, due to signal attenuation caused by the material that makes up the buildings. In addition, reflection, and interference from other signals can further impair accuracy. Therefore, huge efforts are needed to make GPS as precise as IPS [36].

This technology has seen success in many areas like manufacturing where it is used to control assets, logistics, and emergency response to increase labor safety and efficiency and reduce costs and production time. Some shopping centers use it to enhance the customer experience and for location-based marketing. In multi-story car parks where GPS is generally not available, intelligent IPS-based parking systems can be used to assess occupancy. Finally, in hospitals, IPS can be used to help visitors find a specific service or exit, as well as locate equipment and patients [37].

IPS still must deal with stochastic errors and imprecisions but can easily compensate for those. It can take information from multiple device types and cross-reference it to handle ambiguity and enable error compensation.

A large variety of devices can be used as part of IPS. Some examples include Wi-Fi and Bluetooth antennas, digital cameras and clocks, tags (RFID), mobile devices, and beacons placed in known locations. For example, the sensors created by the SAFEHOME team attempt to determine the location through the combination of the waves generated by the vibration of the steps of a person.

3.5.1 Functions of IPS

There are three main functions of IPS. The first and most common focus is on solely locating targets [38].

The second allows locating and tracking. This type can track a path followed by a person or object. It starts by determining the initial location of the target and every time a predefined interval passes it registers the location again, creating a path [38].

The last type is identification and segregation. These tend to be used when the number of targets being observed is greater than one. The system can segregate the target from the rest of the movement in the room through a tag or a mobile device associated with it [38].

These three methods often go hand in hand as a system often requires more than one to comply with the specification. The SAFEHOME sensor currently only allows for the first method. However, in conjunction with the collector being produced in this dissertation, the second function will be easily achieved by creating a history of past locations. In the future, the SAFEHOME team will work to be able to distinguish different people but for the moment what they have is enough to allow for proof of concept.

3.5.2 How can IPS be achieved – wireless technologies and others

As mentioned before several technologies can be used to achieve IPS, but the most popular are wireless-based Wi-Fi Positioning Systems (WPS) which are based on measuring the received signal and applying Wi-Fi fingerprinting localization methods based on wireless access points [39]. Bluetooth can also be used to infer proximity and can be combined with indoor mapping (iBeacons from Apple[14]). Choke point concepts specify that the usage of reader antennas (usually RFID ones) positioned strategically can determine the position of a tag [40]. Meanwhile, Grid concepts defend that smaller devices with smaller ranges can be arranged in a grid to determine location. Long-range sensors that can cover an entire floor use data like angle and distance to determine location. The Angle of Arrival measures the angle of different sensors and combines information to determine location. Time of Arrival uses the time the signal takes to go from a tag to a sensor to calculate the position of the target, even though they usually require a complex clock synchronization system to combine the time values information acquired. Some methods measure the signal strength to calculate distance [41]. [42]

In this project, a system of optic sensors developed under the scope of the project SAFEHOME will be used. These sensors tend to adopt a grid-like positioning to better measure the waves caused by steps and calculate distances from the wave strength received by several sensors.

3.5.3 Mathematical Methods

Two methods are well-known to improve indoor location:

The first method is an Empirical one. Using a large set of known locations and an algorithm like the k-nearest neighbor, a system can infer the positioning of a given object. This method is, however, quite inaccurate when the environment around the system changes a lot [43].

The second method is Mathematical Modeling. This method calculates location using signal propagation, distances, and angles. Using trigonometry as a basis, location is calculated using trilateration and triangulation [44]. There are even some more complex methods like Bayesian statistical analysis (probabilistic model), Kalman filtering (for estimating proper value streams under noise conditions), and the Sequential Monte Carlo method (for approximating the Bayesian statistical models) [45].

The SAFEHOME system to be used calculates the distances using triangulation through the intensity of the waves that reach each sensor.

3.6 Telemonitoring

After studying and understanding better how to obtain information about non-clinical IoT devices and other types of sensors, it is important to know how all this can be applied to healthcare. Telemonitoring is defined as the usage of information technology to monitor patients at a distance to prevent, predict, and cushion situations that could lead to harm or even the death of a patient. The collected information can then be shown to health professionals to help them with diagnosis or simply to keep a close eye on a patient at risk [46].

This makes it highly important that the collected data is of great quality. Bad data can lead to an erroneous diagnosis or wrong assessment of the situation of the patient and therefore to an inadequate treatment solution. This can even mean a misguided recommendation of a medicine that can negatively impact the patient. On the other hand, good quality data not only helps patients to receive better care but also helps extra research and analysis of related conditions. The conclusions and diagnosis can only be as good as the data gathered [47].

[14] iBeacons - https://www.valuewalk.com/apple-inc-aapl-ibeacon-with-micromapping

3.6.1 Types, Technologies, and Devices

Usually, the data collected in telemonitoring systems are clinically related, like heart rate, blood pressure, glucose levels, and weight as can be seen in Figure 8. This branch of telemonitoring is known as biotelemetry. It uses devices such as scales, blood pressure devices, and cholesterol meters. This information can then be compiled and visualized through a convenient application or interface (web and/or mobile) for a formal caregiver to analyze [48].

Cardiovascular:	Metabolic:
Heart rate[7,16,27,31,34]	Body weight[14]
Fetal Heart rate[15]	Basal metabolic rate[4]
Blood pressure[5–7,16,26,27]	Blood glucose[18]
ECG[7,10-12,16,22,26-28,34]	Blood lactate[34]
Pacemaker parameters[10,11]	Blood ethanol[34]
Stethoscopy[13]	Diet[17,18,48]
Hematologic:	Physical activity[18]
Coagulation (INR)[19]	Temperature[7,31,34]
Respiratory:	Urologic:
Pulse oximetry[16,26,27]	Intravesical pressure[2]
Spirometry[8,9,45]	Obstetrics-gynecology:
Respiratory rate[7,26,28,31]	Intrauterine pressure[20]
CO_2 production[4]	
O_2 consumption[4]	
Neurologic:	Others:
EEG[30]	Movements[28,34]
EMG[35]	Drug therapy[18]
Intracranial pressure[23]	Geographical location (GPS)[31,34]
	Home activity[21]

ECG, electrocardiogram; INR, international normalized ratio; EEG, electroencephalogram; GPS, global positioning system.

Figure 8: Cases of Successful Data Collection [46]

However, as can be seen in Figure 8, sometimes non-clinical information can also be useful to infer additional risk situations. Non-clinical telemonitoring typically uses location and routine-related data to achieve this. It is usually focused on knowing if a person's location and body positioning are familiar and considered normal during the day. Several technologies and devices can be used, such as wearables, pendants, belts, and cameras. In addition, IPS and IoT devices like smart switches, plugs, door, and movement sensors, air quality detectors, noise sensors, fall detectors, and cameras can be used to determine where and how the person is inside their residence and if they are generally safe. These services can be combined to have information about the patient everywhere he goes. Beyond that, behavioral and routine analytics can be used to detect possible risks [49].

3.6.2 Who benefits from telemonitoring and the difficulties that arise

After understanding what telemonitoring is, it is important to understand who benefits from it, how, and the difficulties they face while using it.

The first beneficiary is the patient. Telemonitoring means they don’t have to go to the hospital more than strictly necessary, reducing the costs and time wasted. It also means they can be better followed by doctors, reducing isolation, and raising safety and autonomy in their lives. In addition, provides them with tools to better understand and manage their condition, as well as actively participate in their treatments. Finally, it offers the patient more ways to contact their doctor strengthening the patient-provider relationship and improving satisfaction [50].

Consequentially, formal caregivers, such as doctors and nurses, also benefit. With telemonitoring, medical professionals start having remote monitoring as a new way of reaching their patients. They can be notified strictly when necessary, raising their availability for situations that have higher urgency. Trained administrative support people can also help with the basic triage. Diagnosis is facilitated by the constant information collection provided by telemonitoring devices. More quality data means better diagnosis. Lastly, they also benefit from the strengthened patient-provider relationship as it creates loyalty [50].

Finally, the informal caregivers also have their daily tasks facilitated. They get to accompany the patient without always having to be present. They can also act faster as they are usually closer to the patient and provide the first assistance according to the urgency of the situation before contacting the doctor. Ultimately, it reduces the stress of being away from the patient [50].

However, it is not all pros. There are still challenges this technology must overcome. Not everyone is going to agree to be monitored every day as this technology requires consent from the patient to legally collect their data. This becomes harder when a great percentage of users have increased difficulties understanding the technology being used. This is a direct consequence of population aging that can even affect their ability to use the system. The contrary may also happen: if the person using the system is too knowledgeable about privacy issues, they might become skeptical about using it. In addition, even the hospital staff can offer resistance because of the lack of time or simply because they are not very comfortable using new technologies. In this case, they must be properly trained to use the systems, and this is usually expensive to afford. This type of system is usually expensive, increasing the financial burden on the healthcare institution [51].

3.7 Assisted Decision-Making Systems with Alerts

After writing about the basic concepts and technologies that allow for data collection and basic telemonitoring services, it is important to describe what decision support systems and what they can offer.

Assisted decision-making systems are known as offering Clinical Decision Support (CDS) [52]. CDS systems are intended to assist caregivers with decision-making and usually aggregate multiple services, but the three main ones are going to be detailed in the next subsections. The examples provided below will be focused on non-clinical CDS that include alerts that are more relevant to this work.

3.7.1 Data collection and selection

The first microservices to be developed are the collectors of data through location and IoT devices. The usage of these services implies that the house of the patient needs to have the technology to do so. In other words, smart light switches, smart plugs, doors, and movement sensors need to be installed to collect information. These kinds of devices collect mostly non-clinical information and can be used to detect or infer if something wrong is happening (e.g., the person fell, or the person is moving slower than normal, or not moving at all, etc.).

Before using this information, though, it goes through a selection to increase data quality. The first reason to perform this operation is that sometimes devices may send incorrect information that needs to be removed. The second reason is that maybe only part of the collected information is useful to the caregiver and therefore too much data can make decision-making harder than needed. This data can then be sent to a target-designated microservice and organized in such a way that facilitates the analysis [53]. These are the first two rules for proper data analysis.

As part of this dissertation, the collected information (IoT, indoor and outdoor) will be displayed in a convenient web interface to facilitate its visualization to the caregiver. Also, the associated risks will be sent as alerts if the person in care falls, doesn't move for a long time, strays too far from the house, or is in a weird spot of the house at a certain.

3.7.2 Alerts

The next step of the process consists of using the data collected through the devices to detect patterns in the patient's daily life. If these patterns are not respected and a deviation from the usual routine is noticed, it might indicate a risk. For example, if the person is not able to complete a task around the house in the expected time interval, it might mean they passed out or just got distracted by something else. Additional sensor data can be used to disambiguate.

The microservice developed should have some way to detect these deviations and associate actions when and if they are or are not fulfilled. Actions can be, for example, notifying the informal caregiver that the rule was not satisfied. The caregiver can then make an informed decision about the situation. Either calling the patient, the emergency number or even ignoring the notification/alert are all valid options after having access to the data [53].

Alerts can range from situations as simple as falls to situations like the patient/elder (from now on called dependent) not getting up at the normal waking hour. In this dissertation, it will be attempted to use information from various collectors to determine and warn the caregiver of potential risks. The combination of the data from the three collectors will facilitate the creation of useful alerts.

3.7.3 Alert Adjustment

The constant monitoring of the daily routine may also indicate deviations from it. These deviations can have many origins such as the decreased ability to perform a certain task or even cognitive decline [54].

Therefore, sometimes notifications might need adjustments. If a notification/alert is generated as a result of a condition not being fulfilled and every time it is ignored by the caregiver, then maybe it does not represent any danger. If a plug in the living room has been turned off and the movement sensor detects motion after the person's usual bedtime that may indicate a risky situation. But if this happens every day, maybe the person's bedtime is later than expected or simply the time changed for some well-known reason (e.g., a change in medication). In any case, both these scenarios require condition changes to better evaluate the risk. This can be achieved through routine analysis using statistical techniques, machine learning (ML), or artificial intelligence (AI) [53].

AI/ML will not be the focus of this dissertation as the developed application will be a proof of concept but later the SmartAL team can evolve the project to accommodate this line of work.

3.8 Examples of real-world products

Knowing now the type of services intended to be developed under the scope of this dissertation let us investigate some products in the market. Some real-world companies have already started building systems that use Monitoring and Evaluation (M&E) tools, like the previously described services, and as such existing products will be used as a reference for this project.

Aerial Technologies[15], for example, developed a system capable of inferring indoor positioning through Wi-Fi wave distortion. The Wi-Fi devices installed in the house of the customer collect information that is then passed through AI algorithms (machine learning and predictive analytics). These algorithms can also be customized and allow for AI-powered feedback. The information collected is then organized and sent to a mobile interface. This system can provide eldercare, remote patient monitoring, security, and other home-related services (e.g., energy management) using features that include presence, motion, location, and fall detection. It is possible to handle sleep metrics, notifications and alerts, pet motion filtering, and historical analysis. This technology, however, has a clear inclination toward indoor components completely ignoring outdoor functions.

Another example of this type of service has been created by **Cognitive Systems Corporation**[16]. The product called **Wi-Fi Motion**, as the name implies, uses ripples in the Wi-Fi signal to determine motion. IoT devices in the user's house become motion sensors without deterring from their original purpose or slowing down the network. Complex algorithms are used to differentiate human motion from animal or object motion (e.g., fan movement). The user can view motion in real-time on their phone and define rules related to the movement or the lack of it; then when the rule is not fulfilled the system produces a notification. Other actions include home automation (e.g., variation in lighting and thermostat when a person gets home). All of these allow services like home and wellness monitoring to be offered to the client[17]. Again, this one too disregards outdoor information and is very focused on movement.

The above Wi-Fi-related systems are subject to some challenges. The detection of small gestures, crowd counting, object occlusion, real-time processing requirements, and multipath propagation [55]. Also, all seem to have no routine analysis component. This means that even though they can detect falls and movement very accurately they do not seem to have alerts regarding the scheduling of the users (times of meals, bed, medication, etc.). In addition, they do not infer risk situations based on IoT devices other than positioning and movement ones.

Regarding outdoor-focused products, in the market, there are several examples: **Fitbit**[18], **Zepp Life**[19], **Galaxy Wearable**[20], and **Apple Watch**[21]. Associated with specific wearables, they offer features like basic vital signs, exercise, and sleep monitoring, and sometimes advise on what to do to reduce some risk factors such as high visceral fat, high BMI, and lack of water intake. Also, they help track outdoor activities like walking, running, and cycling. Their apps

---

15 https://aerial.ai/
16 Cognitive Systems - https://www.cognitivesystems.com/
17 https://www.cognitivesystems.com/applications
18 FitBit - https://www.fitbit.com/global/us/home
19 Zepp Life - https://play.google.com/store/apps/details?id=com.xiaomi.hm.health&hl=pt_PT&gl=US
20 Galaxy Wearable - https://play.google.com/store/apps/details?id=com.samsung.android.app.watchmanager&hl=en&gl=US
21 Apple Watch - https://www.apple.com/pt/watch/

allow users to set fitness goals and monitor their progress. Most of these devices are connected to clouds and offer public APIs to access data (i.e., **Google FIT**[22] and **Apple Health Kit**[23]).

Senior Safety[24] has produced an app that is indicated for informal caregivers who want to monitor people in their care. The **Senior Safety App** offers features that are useful if the dependent has dementia or other cognitive challenges, this app can alert the caregivers when one of their dependents wanders outside a set boundary, when there is high ambient noise around them, or when a fall is detected. The GPS tracking, low-battery alerts, and inactivity alerts allow caregivers to stay informed if anything unusual is happening. However, it requires the possession of a mobile phone by the dependent.

The previous systems do not consider any indoor information. So, the market is somehow fragmented. There does not seem to exist a solution in the market that considers both outdoor and indoor information.

When it comes to IoT products, several options are available, but they are not really healthcare-oriented and focus more on self-safety like for example **Amazon Alexa Connect Devices**[25], **Samsung SmartThings**[26], **Apple HomeKit**[27] (also known as Apple Home), and even Altice Labs' **SmartHome**[28] but indirectly allow for elderly supervision.

Regarding apps that consider routine analysis for risk assessment, there are simple solutions like **Snug Safety**[29] that allows for a daily check-in and if it does not happen it contacts a preconfigured set of emergency numbers. **Lotsa Helping Hands**[30] and **Caring Village**[31] both offer services to manage appointments and support one or more caregivers surrounding the dependent.

In the clinical area, **Medisafe**[32] created a **Pill Reminder & Medication Tracker** app that provides reminders to take pills and manage medications. It sends alerts when it is time to take medicine and when prescription refills are due. Also provides drug interaction warnings to avoid any potential health hazards.

All these applications and many others in the market are interesting but very specific. There are no transversal solutions that put the user in the center and collect information from different types of devices. It is difficult to find a product that uses IoT, indoor and outdoor information from a variety of different device options to create a unified alert service that covers a wide range of situations. This is the proposition of this internship: to create a user-friendly app that receives information from many sources and displays it to the user (in this case the informal caregiver) to help with decision-making.

---

22 Google Fit - https://www.google.com/fit/
23 Apple Health Kit - https://developer.apple.com/documentation/healthkit
24 Senior Safety - https://www.seniorsafetyapp.com/
25 Amazon Alexa - https://alexa.amazon.com/
26 Samsung Smart Things - https://www.smartthings.com/
27 Home App - https://www.apple.com/home-app/
28 SmartHome - https://www.alticelabs.com/products/SmartHome/
29 Snug Safety - https://www.snugsafe.com/?fpr=gwg15&fp_sid=apps
30 Lotsa Helping Hands - https://lotsahelpinghands.com/
31 Caring Village - https://caringvillage.com/
32 Medisafe - https://www.medisafe.com/

3.9 Summary

In summary, this chapter provided a comprehensive overview of the type of devices and services in the area, with a specific focus on their potential to enhance decision support. The dissertation aims to integrate these technologies to provide more comfortable long-distance experiences for patients at home while also supporting informal caregivers in making informed decisions.

The chapter analyzed related products and services by highlighting the benefits and difficulties faced in the field. This information will guide the project in its effort to achieve the proposed goals of developing a transversal telemonitoring proof of concept. The findings and insights presented in this chapter serve also as a foundation for future research and development, paving the way for more efficient and patient-centric delivery of healthcare services.

4 State of The Practice

After covering the state-of-the-art it is important to describe the already existing Altice Labs system and all the relevant background information. Therefore, the current chapter is going to detail the current state of the practice of the SmartAL ecosystem, more specifically the B2C facet of it. This includes a small introduction, services, and the "as is" architecture of the system

4.1 SmartAL

SmartAL[33] or Smart Assisted Living is a brand but also a suit of the telemonitoring services developed by Altice Labs. As already briefly mentioned in this document, the solution aims to simplify people's daily lives. Its main objective is to offer a tool where end-users may follow in real time the health status of the patients. It is also customizable to meet the needs of different users, including patients, doctors, informal caregivers, administrators, installers, etc.

The commercial offer targets the B2B healthcare market, meaning institutions such as hospitals, seniors' homes, and municipalities while the new B2C solution that is still under development directly targets final end-users (e.g., patients, doctors, caregivers). The overall SmartAL B2C architecture has already been defined according to the latest technologies and some initial services have been developed as proof of concept. This new platform deployed in a public cloud is for the time being a working space for Research and Development (R&D) projects.

4.2 SmartAL B2C Services

The idea behind the B2C offer is to gradually build a pool of simple and atomic care services that can be used separately or in conjunction to create a more holistic experience.

The B2C solution under development already offers some telecare services such as Teleconsultation and Telelocation. The teleconsultation service offers a patient the chance to get an online appointment with a specialist whenever and wherever they need, while the telelocation service (based on outdoor GPS devices) allows for informal caregivers to accompany people in their care. In addition, there are two new services – Telemonitoring and TeleAlert. Both developed in an academic context.

Initially, the TeleAlert service was supposed to use the output of the Telelocation, meaning the same devices and related GPS data handled by the second service were to be used in TeleAlert. This, however, was not possible due to significant changes in the outdoor location devices and the way they communicate. A new outdoor location gateway had to be developed and integrated from scratch. This slightly delayed the overall integration and validation process, but in the end, it was successfully done.

The web portal is already online for PoC[34] and it includes the new TeleAlert application[35].

[33] https://www.alticelabs.com/products/ehealth-smart-assisted-living/
[34] https://smartalb2c.ddns.net/wordpress/
[35] https://smartalb2c.ddns.net/telealert/

4.3 SmartAL B2C Architecture

The SmartAL B2C microservice architecture by the time the project started is represented in the following picture:

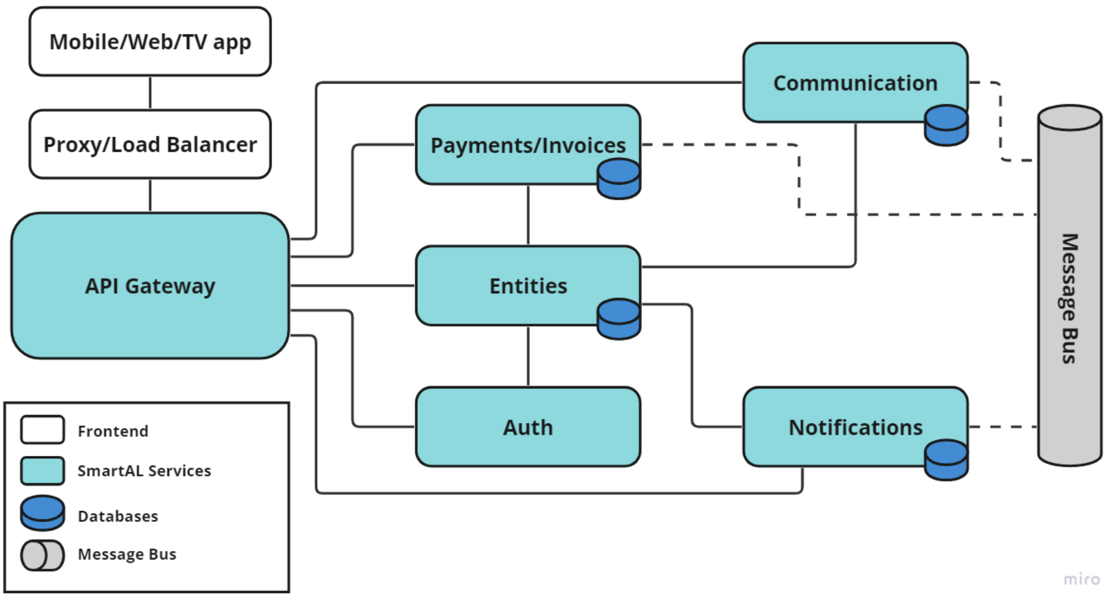


Figure 9: SmartAL B2C Architecture "as is"

As can be seen in Figure 98, the system has a frontend composed of applications and a proxy. They are responsible for displaying information to the user in an understandable and organized way and serving as an intermediate/load balancer for the app's requests.

The backend of the applications is generically composed of microservices, databases, and a common message bus. The **API gateway** is the component responsible for creating the bridge between the external web clients and the other microservices. The **Entities** microservice is the component responsible for managing users and profiles. It also "assists" the **Auth** component which oversees authentication and authorization. The **Payments/Invoices** microservice is responsible for all the payments related to the different applications and can connect to external payment and invoice services. The **Notification** component receives the packets that are placed in the message bus by other services and sends emails, SMSs, and push notifications/alerts in case of need. Finally, the **Communication** service also receives events from other services and allows for real-time updates needed for most SmartAL services.

There are other components not mentioned here as they are specific and related to other applications and therefore not relevant to this dissertation. Further details on the architecture will be described in Chapter 5 Requirements and Architecture.

4.4 Summary

This chapter briefly describes the main principles and services of the SmartAL system B2C solution.

In addition, a small overview of the current architecture was also shown. The microservices implemented in this dissertation will be integrated and make use of preexisting ones to achieve the objectives of this internship. This integration will be detailed in the next chapter along with the updated architecture.

5 Requirements and Architecture

This chapter outlines the various needs and constraints that must be considered to ensure the success and effectiveness of the project.

5.1 Introduction

Before starting the development of the project, it is crucial to understand the requirements that must be met by the **TeleAlert** service. By thoroughly revising the requirements of the project, it is possible to ensure that it can meet all necessary criteria and function properly. The full specification process includes the definition of personas, macro and micro use cases, storyboards, and functional and non-functional requirements.

In addition, it is also important to define the overall architecture of the system as well as the new microservices and how they will be incorporated. Finally, the layout and structure of the application will be designed and specified, as well as its interaction with the new microservices. By describing the overall integrated system, we can gain a clear picture of how it is intended to function.

5.2 Personas

The current section will detail some personas associated with **TeleAlert**. These personas represent the user types that may interact with the system. By anticipating their needs, motivations, and frustrations, it is possible to get a better understanding of what our target audience desires and offer a more user-centered experience.

The two following figures describe examples of possible users of the system. One is for an informal caregiver named Vanessa and another represents her mother Elizabeth who currently lives alone and is her “dependent”, meaning the person in their care.

Vanessa works in sales and is married with kids. She attempts to take care of her mother on the side, conciliating that with her work and family life. She often tries to help her mom with her daily tasks.

Elizabeth has been living alone since her husband passed. She attempts to keep an active lifestyle despite her old age. Recently she developed light dementia making it harder to keep up with everything she would like to do. Beyond that she has been suffering from arthritis for some time now, making her usual walks a bit more difficult.

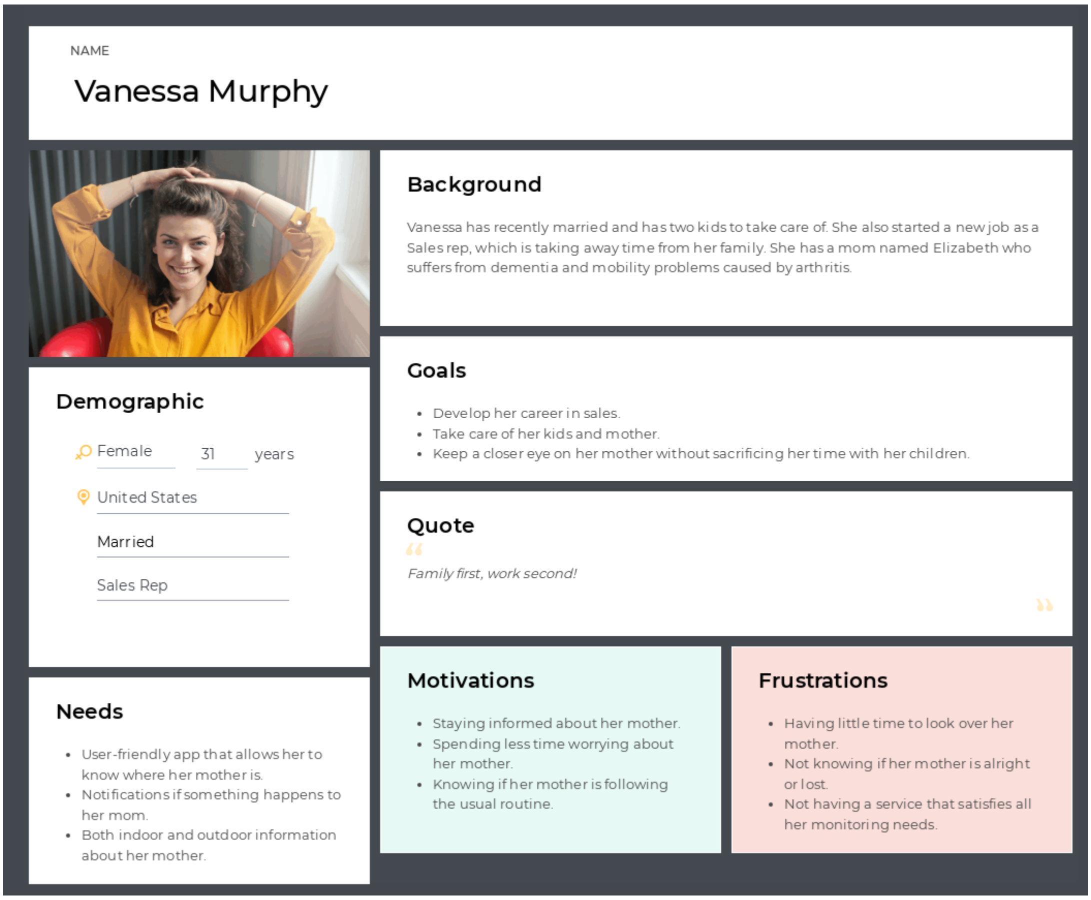


Figure 11: Vanessa Murphy's Persona

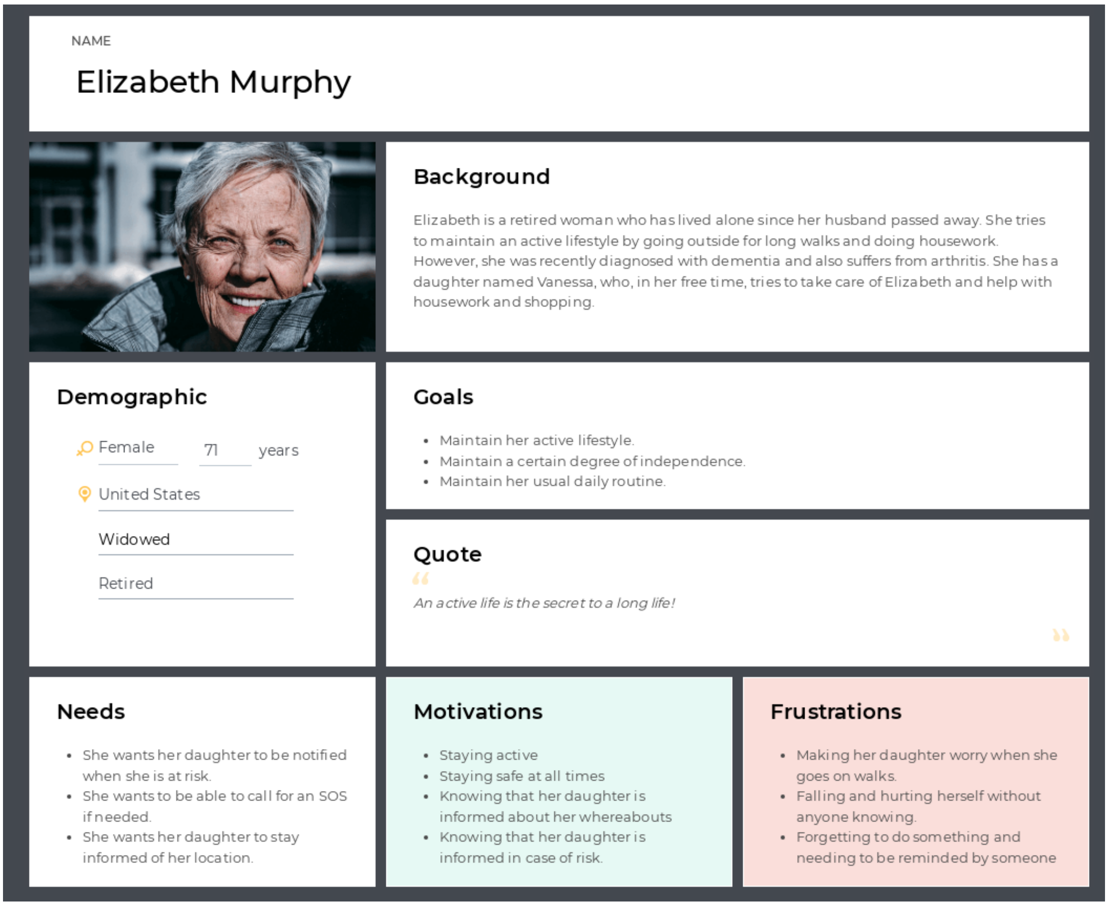


Figure 10: Elizabeth Murphy's Persona

5.3 Use Cases and Storyboards

Use Cases describe the interactions between a system and its users. They can be used to explore the various ways in which the system functions and the actions users can take. Storyboards turn the use cases into concrete scenarios that represent hypothetical situations in real life. Together, these can help define the functional requirements of the system. The aforementioned personas will be used to define the storyboards. The lighter-colored bubbles represent the macro use cases that are central to the system, while the darker ones represent the micro use cases.

5.3.1 Use Case Diagram

Figure 12 depicts the overall use case diagram. In lighter blue, are represented the individual macro use cases that define the TeleAlert system. The next chapters will detail these.

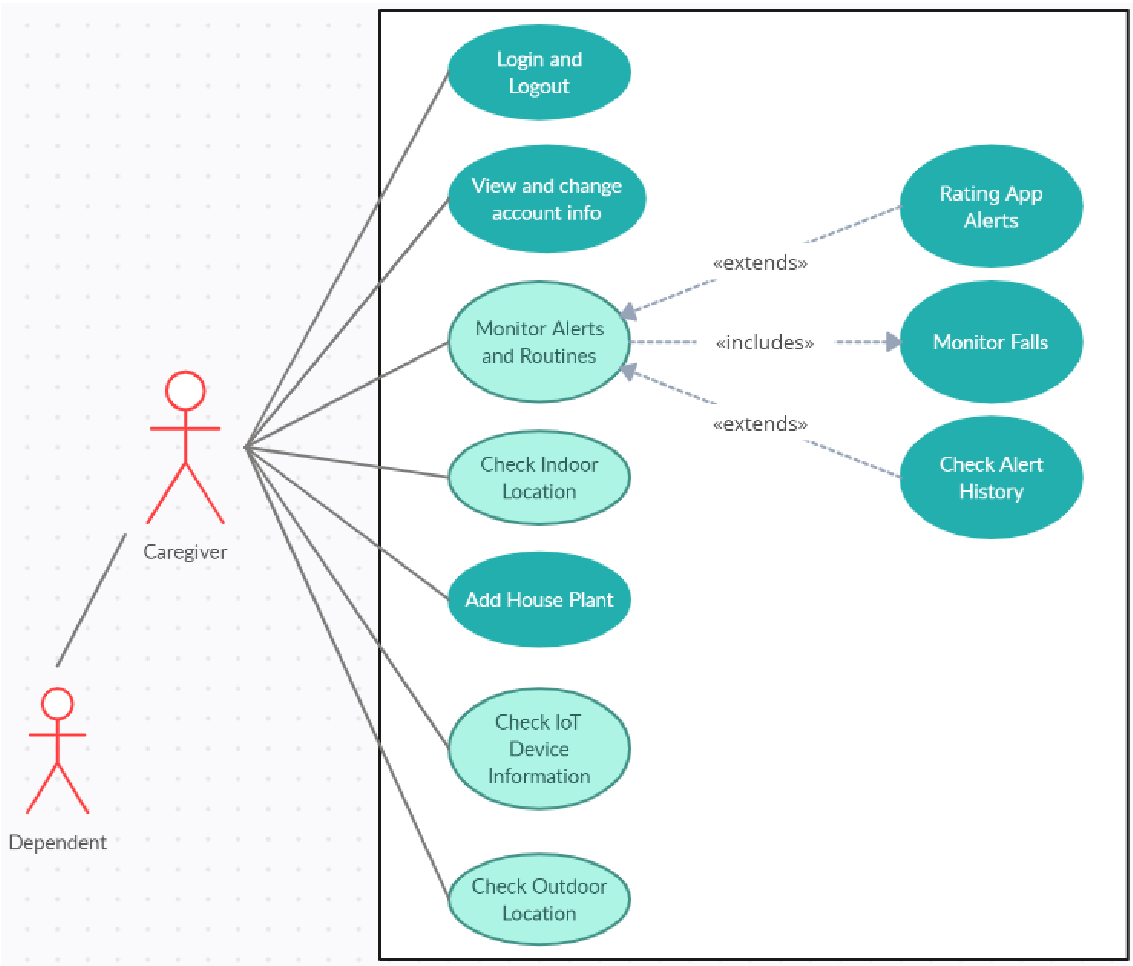


Figure 12: Overall Use Case Diagram

5.3.2 Macro Use Cases

Macro use cases are high-level descriptions of the main goals and the associated functionality of the provided service. They give an overview of the system's capabilities and are used to better define the scope of a project. The following tables will detail the macro use cases of the system to be developed.

Table 2: Macro Use Case – Check Indoor Location

Name	Check Indoor Location	
Actors	Informal Caregivers and Dependents indirectly	
Description	Older people who live alone often have informal caregivers to help them with daily tasks and check if everything is normal. However, these people cannot be around all day because they still have their normal lives to follow. Therefore, a service that can provide them with information about the people in their care while they are away would be of immense help. Indoor location, movement speed, step quality, and other related information can help the caregiver infer possible risks. The caregiver should be able to view the indoor location of their dependents.	
Storyboards	**Nº1**	Elizabeth suffers from arthritis, cannot make huge efforts, and cannot safely leave the house alone. His daughter Vanessa often comes over to help with tasks like cleaning and cooking. However, she has a 9-5 work that does not allow her to be over as much as she would like. To ease her mind through the day, she bought an indoor location service, so she could accompany her mother’s location and contact Elizabeth if she deems it necessary. One day, Elizabeth was still in bed way past her normal wake-up time, and her daughter immediately called her to find out she was feeling sick.
	Nº2	Elizabeth suffers from arthritis. This reduces her mobility and causes pain. Vanessa noticed a pattern whenever she was about to fall. Usually, Elizabeth starts to walk slower with heavier steps. While Vanessa is there, she can help her get up and contact emergency services if needed. However, when Vanessa leaves, she has no way of knowing if Elizabeth fell and hurt herself. So, Vanessa decided to install a telemonitoring service that warns her when situations like this happen so she can immediately assist Elizabeth as needed.

Table 3: Macro Use Case – Check IOT Device Status

Name	Check IoT Device Status	
Actors	Informal Caregivers and Dependents indirectly	
Description	Smart home technology has been gaining consistent popularity over the last few years. IoT devices like smart switches and movement, door, and window sensors are being used to create interconnected systems that collect information and use it to improve safety and the overall quality of life of the users. One of its purposes is to help elderly people living alone. The caregiver should be able to check the data of the devices to better assess risk situations.	
Storyboards	**Nº1**	Elizabeth lives alone and her daughter only visits after 6 p.m. So, during the day Vanessa worries that her mother who suffers from dementia might wander off and forget to take her medication. Therefore, she decides to install door and motion sensors in her mother’s house to take better control of her daily activities and warn her when Elizabeth leaves the house or fails to follow her usual routine.
	Nº2	Elizabeth has mobility problems related to arthritis. She takes medicine for it that she saves in the kitchen to consume with her meals. However, Vanessa knows that she sometimes fails to eat and take her medicine. So, she installed motion sensors in her mother’s kitchen to know about her movements. This way she can call Elizabeth and remind her to eat and take the medicine.

Table 4: Macro Use Case – Check Outdoor Location

Name	Check Outdoor Location	
Actors	Informal Caregivers and Dependents indirectly	
Description	Outdoor location information can also be useful to determine risk situations outside houses and institutions. By defining geofences and safe places around the houses it is possible to send alerts to caregivers only when necessary. This information can complement the indoor information allowing for an extension of the range in which the dependent is monitored. The caregiver should be able to view the outdoor location of their dependents.	
Storyboard	**Nº1**	The entrance sensor that Vanessa installed indicates that her mom left the house. So, the indoor system stopped being useful for monitoring her mom. Vanessa knows that her mom sometimes goes for a walk or visits a nearby shop. So, to have more information on her routine she bought Elizabeth a smartwatch to be able to track her and check if her mom is straying too far from her house or is in a usual shopping spot. If her mom goes too far Vanessa can call her to guide her back home.
	Nº2	Whenever Elizabeth leaves the house, she is still at risk of falling and hurting herself. So, Vanessa bought her mom a smartwatch with fall detection capabilities, where alerts can be triggered and sent automatically to Vanessa.

Table 5: Macro Use Case – Monitor Alerts and Routines

Name	Monitor Alerts and Routines	
Actors	Informal Caregivers	
Description	A routine is defined as a group of activities a person performs day after day around the same time. People usually have activities they perform either in a certain order or at a specific time or day of the week. Rules can be established based on these routines to be able to detect deviations. Alerts can be sent when a particular rule is not respected. In addition, these rules can then be adjusted through Machine Learning to better fit the monitored person's time scheduling. Outdoor, indoor, and IoT information will be used to check the configured conditions and therefore deviations from the rules.	
Storyboards	**Nº1**	Elizabeth takes medicine at exactly 9 a.m. She usually wakes up around half past 8, so she has time to have her breakfast before. Her daughter Vanessa knows that sometimes she forgets to set up the alarm, so she can subscribe to a service that sends her an alert if her mother is still in bed around 9. For that, she installed some movement sensors. If the sensors detect no movement in the room, Vanessa will be alerted so she can take the necessary measures for his mother's safety.
	Nº2	Vanessa set up a movement detection system in the house to monitor her mother, Elizabeth. Usually, her mother goes to bed around 10 p.m. so she set up a service that warns her when she does not. However, every night Vanessa got an alert that she was still up and every night she called, and everything was fine, so she dismissed the alert as not being important. As it happens her mother goes to bed later than expected because of her new medication that must be taken late in the evening. The system should learn that movement is always present after the initially configured bedtime and that the alerts are being dismissed as not important. So, it can adjust the rule to a new time when it normally does not detect movement, meaning the person is probably asleep.

5.3.3 Micro Use Cases

Micro use cases will be used to expand on the previous macro use cases to describe the primary operations of the systems and make it simpler to extract the functional requirements more thoroughly. Most of these will refer to the external system needed to collect data from existing devices and sensors (i.e., SmartHome for IoT, SAFEHOME for indoor, and also from the new OGTW – outdoor gateway developed by AlticeLabs).

Table 6: Micro Use Case - Login and logout

Name	Login and logout
Actors	Informal Caregivers
Pre-requirements	The caregiver needs to have created an account using the TeleAlert application.
Description	The caregiver should be able to log in and log out using the pre-built login process from SmartAL.
Storyboard	After creating an account Vanessa can now log in to TeleAlert and log out if needed.

Table 7: Micro Use Case – View and change account info

Name	View and change account info
Actors	Informal Caregivers
Pre-requirements	The caregiver needs to have created an account using the TeleAlert application.
Description	The caregiver should be able to view and change any account info they need
Storyboard	Vanessa changed her phone number recently, so she updated it in the app. Additionally, she updated the account’s password which she had not changed in a while.

Table 8: Micro Use Case – Add the house plant

Name	Add the house plant
Actors	Informal Caregivers
Pre-requirements	The caregiver needs to have created an account and a dependent to associate the house plant with.
Description	The caregiver should be able to add a plant to the dependent’s house so that they can later see the indoor collected information correctly mapped to the image.
Storyboard	Vanessa can now add her mom’s house plant to the system by selecting an image and inserting the dimensions.

Table 9: Micro Use Case – Monitor falls

Name	Monitor falls
Actors	Informal Caregivers
Pre-requirements	The caregiver needs to have created an account, and a dependent with an associated house plant with at least one device able to detect falls.
Description	Depending on the device the caregiver can receive a notification about a fall or simply a phone call, given that the caregiver's number has been previously configured in the device.
Storyboard	Vanessa was notified that her mom had fallen and received a phone call. Nevertheless, she immediately went to her mom's home to check if she was okay.

Table 10: Micro Use Case – Check notification history

Name	Check notification history
Actors	Informal Caregivers
Pre-requirements	The caregiver needs to have created an account and a dependent with associated devices.
Description	The caregiver should be able to check the alerts from a long time ago.
Storyboard	When visiting the doctor with Elizabeth, Vanessa wanted to let the doctor know how many times Elizabeth had fallen in the last month. Vanessa then checks the history of alerts and can access the number by seeing older alerts.

Table 11: Micro Use Case – Rate app alerts

Name	Rate app alerts
Actors	Informal Caregivers
Pre-requirements	The caregiver needs to have created an account and a dependent with associated devices.
Description	The caregiver should be able to rate the alerts negatively or positively according to how accurate they are. The system can then learn from the given feedback.
Storyboard	The system detected Elizabeth was out for more than 1 hour. So, it alerted Vanessa that decided to immediately call her mom and concluded that it would become part of her mom's routine because on Wednesday Elizabeth would start to go visit a relative. Vanessa can then rate this alert as not important, so the system adjusts to the situation next week.

5.4 Requirements

This section presents the initial list of functional requirements based on the use cases. Priorities were assigned to each functional requirement together with the ALB team according to the **MoSCoW prioritization scale**. Additionally, some non-functional requirements were defined as well.

5.4.1 Functional Requirements

After describing the use cases, the list of the resulting functional requirements is presented in Table 15. According to the MoSCoW scale, the **Must Have** requirements are essential and directly address the core functionalities of the system and the **Should Have** requirements enhance the user experience. The **Could Have** requirements are less essential for the system but still desirable while the **Won't Have** requirements were excluded due to the time constraint of the project.

Table 12: Functional Requirements

Nº	Functional Requirement	Priority
FR1	The system shall collect indoor location information from the SAFEHOME system.	Must have
FR2	The system shall collect IoT device information from the SmartHome system.	Must have
FR3	The system shall collect outdoor location and device information from the outdoor service system (OGTW).	Must have
FR4	The system shall run the collected information by a rule engine to generate alerts that represent risk situations.	Must have
FR5	The caregivers shall be able to check the location of their dependents both indoors and outdoors.	Must have
FR6	The caregivers shall be able to check indoor and outdoor device information.	Must have
FR7	The caregivers shall be able to check alerts about their dependents and their notification history.	Must have
FR8	The caregivers shall be able to add a plant image of the house to be able to visualize the indoor location information.	Must have
FR9	The system shall update all data in real-time.	Must have
FR10	The caregivers shall be able to visualize all the collected information through a Web App.	Must have
FR11	The caregivers shall be able to log in and log out using SmartAL's prebuilt components.	Should have
FR12	The caregivers shall be able to alter their personal information using SmartAL's prebuilt components.	Should have
FR13	The caregiver shall be able to edit the plant of the house.	Could have
FR14	The caregiver shall be able to manage rules and alter them at will.	Won't have
FR15	The caregiver shall be able to rate alerts using a satisfaction scale.	Won't have
FR16	The system shall be able to adjust rules according to the alert ratings.	Won't have

5.4.2 Non-functional Requirements

Some non-functional technological requirements were defined and are listed below in Table 16. These requirements were decided together with the team to make the new microservices coherent with the rest of the SmartAL system.

Table 13: Non-functional Requirements

Nº	Non-functional Requirement
NFR1	The system's backend shall be protected against unauthenticated and unauthorized requests using JWT[36].
NFR2	The system shall work on at least Chrome and Edge web browsers.
NFR3	The system's frontend shall be developed using React (and Typescript optionally).
NFR4	The system's backend is advised to be developed using either Spring Boot or Node.js.
NFR5	The system's data should be saved in a PostgreSQL database.

[36] Java Web Tokens are used for authentication and authorization purposes. The tokens are signed either using a private secret or a public/private key.

5.5 Technologies

Following the requirements outlined previously, this section presents the technologies used in TeleAlert. The selection of the technologies was guided by the need to fulfill the requirements more efficiently and effectively. This section provides a detailed description of each technology and the reasons for its selection.

The SmartAL ecosystem is an aggregate of small services, each performing a specific function, meaning it follows a microservice architecture. This allows for great scalability, fault isolation, continuous integration and delivery. It is also very developer-friendly as varied languages can be used for each microservice, allowing for greater freedom [56]. This architecture also favors containerization during development using Docker. This will be expanded later in Chapter 6 Implementation.

5.5.1 Backend

When it comes to backend technologies, many other microservices have recently been developed. So, they can be taken as the basis for TeleAlert.

Concerning the **databases**, following requirement **NFR5**, **PostgreSQL** was the choice. This database offers compatibility with many computing languages; it is highly scalable allowing for its usage in high-traffic applications; it supports ACID[37] attributes that ensure integrity and reliability; it provides good data availability and resiliency which are extremely important in systems like this. Other advantages include good code quality, availability of advanced features, large community, and extensibility [57].

As stated, in requirement **NFR4**, the advisable options for the **backend server** were Node.js, a cross-platform open-source server environment, and Spring Boot, a Java-based framework to create microservices. In the table below, it is possible to see a brief comparison between the two:

Table 14: Spring Boot vs Node.js [58]

	Spring Boot	**Node.js**
Single vs Multi-threaded	Multi	Single
Concurrency	May block with a larger number of connections	Does not block with a larger number of connections.
I/O model	Blocking	Non-blocking, event-driven
Language	Java -> vast ecosystem of libraries, frameworks, and tools	JavaScript -> simple and versatile
Community	Large and diverse	Large and diverse
Ecosystem	Offers a set of predefined configurations and best practices, good for consistency and simplicity	Wide range of options to add features and functionality
Team skills	If familiar with it, it is better.	If familiar with it, it is better.
Terminology	Better development process.	Consider if familiar with JavaScript.
Runtime	Java Virtual Machine (JVM) is portable and secure.	Faster at run time.
Scalability	Handles multithreading well	Handles concurrent connections well
Effectiveness	May be slower due to JVM	Known for fast performance

[37] Atomicity, Consistency, Isolation, Durability

Best Usage	Enterprise Applications, Microservices, Batch Processing, Web Applications	Real-Time Applications, API development, Single Page Applications, Streaming Applications

Spring Boot was selected due to its rapid development, great ecosystem, scalability, and performance, but most importantly its support for microservice applications like SmartAL. Beyond that, data received by TeleAlert will be collected from queues and the process of interpreting these is favored by the multithreaded nature of Spring Boot. Finally, as can be seen in the table above, familiarity with the language of the framework must be considered in the choice of the framework and I am most familiar with Spring Boot and Java [58]. Even though Node.js might be better for real-time applications, there is already a service responsible for those. More about it later in this chapter.

Finally, it was decided by the team that a rule engine was good enough to prove the concept of the TeleAlert service, which is based on receiving data, configuring rules, and creating alerts depending on whether or not the conditions are met. Rule engine servers are useful for separating information logic from business logic, allowing for better readability, maintainability, and reusability [59]. The two most popular Rule Engines are Drools and Esper. So, the team recommended those to be studied and compared. The table below is the result of that study:

Table 15: Esper vs Drools [60]

	Esper	Drools
Throughput	Better for a smaller number of rules	Better for a bigger number of rules and events
Execution time	Fast for a small number of rules and events	Fast for a large number of rules and events
Run engine as service	Yes, but only the enterprise edition	Yes
High Availability	Yes	No
Scalability	Not as much	Very scalable

In conclusion, Drools is more scalable and has more options than the non-enterprise Esper version. It is also better for a bigger number of rules as needed to describe the routine of a user and for a bigger number of events generated by several users [60]. Besides, Drools is a better implementation for object-oriented systems like Spring Boot and the language is Java-based adding familiarity to it [61].

5.5.2 Frontend

Compared to the backend, the frontend technology was an easier decision since the existing applications already use a very recent and common free and open-source library. So, regarding requirement **NFR3,** React was chosen as the main framework.

React is a JavaScript library that allows building fast, simple, and scalable frontends for web applications. It is a very flexible framework that allows for components to be reused very easily. It speeds up development as it is very easy to learn, allows for the creation of custom reusable components, has Virtual DOM that allows for fast rendering, has reduced load times, has several development tools, and despite being a recent framework already has a large community. In addition, there are already a lot of extensions that enhance the coding environment of the programmer [62]. SmartAL already has several components that include functions like login and logout, that are built in React and therefore it is easier to integrate new React services on the main web application.

When it came to the frontend the only choice to be made was between JavaScript or TypeScript since React can be used with both. It was decided that TypeScript was to be used as it favors object-oriented programming complementing the Spring Boot backend. Furthermore, it allows for early detection of bugs during development that will not grow further into production [63].

5.6 Architecture

In the preceding section, we delved into the realm of the technologies chosen for the project, laying a solid foundation for the forthcoming exploration of architectural design. This section focuses on presenting the key architectural decisions that underpin the development of our system. Central to this chapter is the utilization of simplified C4[38] diagrams to depict the architectural structure and its components. More details will be provided in the next chapter when presenting the implementation, where the system's functionality will be on full display. This being said, the code diagram will be excluded from this chapter.

5.6.1 Context Diagram

The following context diagram gives a high-level view of the relationships of SmartAL (which includes TeleAlert) with the external systems. It helps to understand the scope and the overall environment of the system.

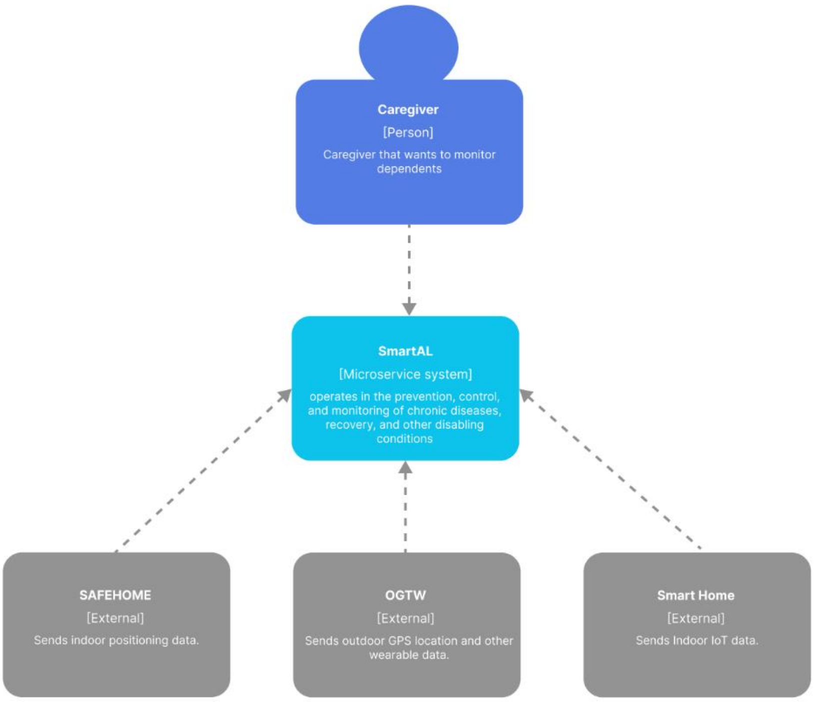


Figure 13: SmartAL Context Diagram

[38] Context, Containers, Components, Code

Figure 13 shows the caregiver as the main user of the application and the three systems from which it needs to collect information. In this diagram, the user interacts with SmartAL to register, log in, and perform other basic operations, and only then invokes TeleAlert microservices to proceed with the visualization of the location, alerts, and other device-related data. The three external services send JSON[39] messages that are interpreted by the new components of the system.

5.6.2 Container Diagram

The container diagram represents the system's architecture on a container level. This means containers within the system and their interactions. Containers can be defined as lightweight, isolated units that encapsulate software and its dependencies, facilitating scalability and portability. In this diagram, each container (e.g., web servers, databases) is depicted as a separate box, and the relationships between containers are shown with arrows indicating data or communication flows. This type of diagram enables developers to gain better insights into how different containers collaborate to form the overall system, aiding in conceiving the system and assisting in decision-making for scalability and maintenance strategies.

Figure 14 displays the microservice architecture adopted by SmartAL. In green are the new containers to be developed, the **TeleAlert Backend** that collects the needed information, and the **TeleAlert Frontend** that displays it. Some other services are needed for the system to be fully functional; between these new containers, the **Api-Gateway** will map the requests to the correct service; the **Auth** service will be responsible for authenticating and authorizing the user and all related requests; the **Entities** service will assist the TeleAlert service by providing additional user data needed to the frontend; the **Databases** service will store the data coming from the new TeleAlert service; finally, the **Communication** service will be responsible for collecting messages from the **Message Bus** and distributing real-time updates to the TeleAlert frontend while the **Notifications** service holds a similar responsibility but is notification-focused.

In addition, the Message Bus also holds another important feature - it is responsible for distributing data regarding indoor and outdoor location, and outdoor device information sent to it by external services through the **Rabbitmq-Proxy** to the TeleAlert service. Note that, an additional external Message Bus will be used for the indoor device data since the SmartHome team made it available to us to facilitate the integration process.

[39] JavaScript Object Notation

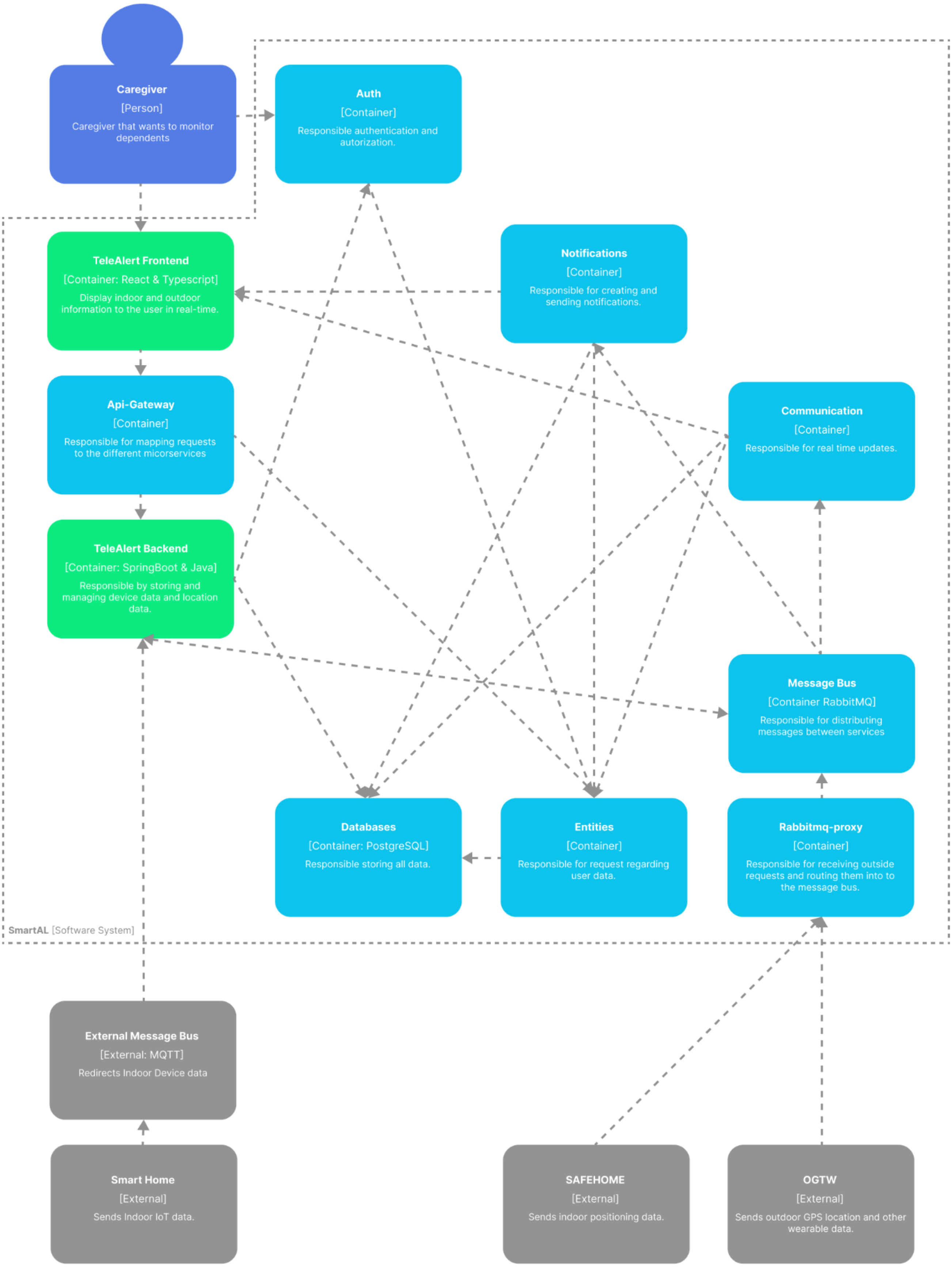


Figure 14: SmartAL Container Diagram

5.6.3 Component Diagrams

The component diagram is used in systems architecture design to depict the internal structure and organization of individual software components within the system. In this diagram, each component is illustrated as a distinct box or shape, with labels to identify its functionalities and responsibilities. The connections between components illustrate how they collaborate to achieve specific tasks or functionalities. This type of diagram helps developers understand the modular design of the system, enabling them to identify reusable components and comprehend the overall architecture complexity.

The frontend presented in Figure 15 already has some functionalities that can be reused from previously developed micro frontends (represented in light blue). The **Microfrontend-Auth** component is already used in many other services as it provides the basic login page and allows for authentication and authorization through JWT. **Microfrontend-Common** also offers components that can be reused from other apps like the notification popup, historical information, and user profile pages. The TeleAlert frontend only needs to be integrated with the functionalities provided by these microservices, so its core development can then be focused on displaying the data sent by the backend.

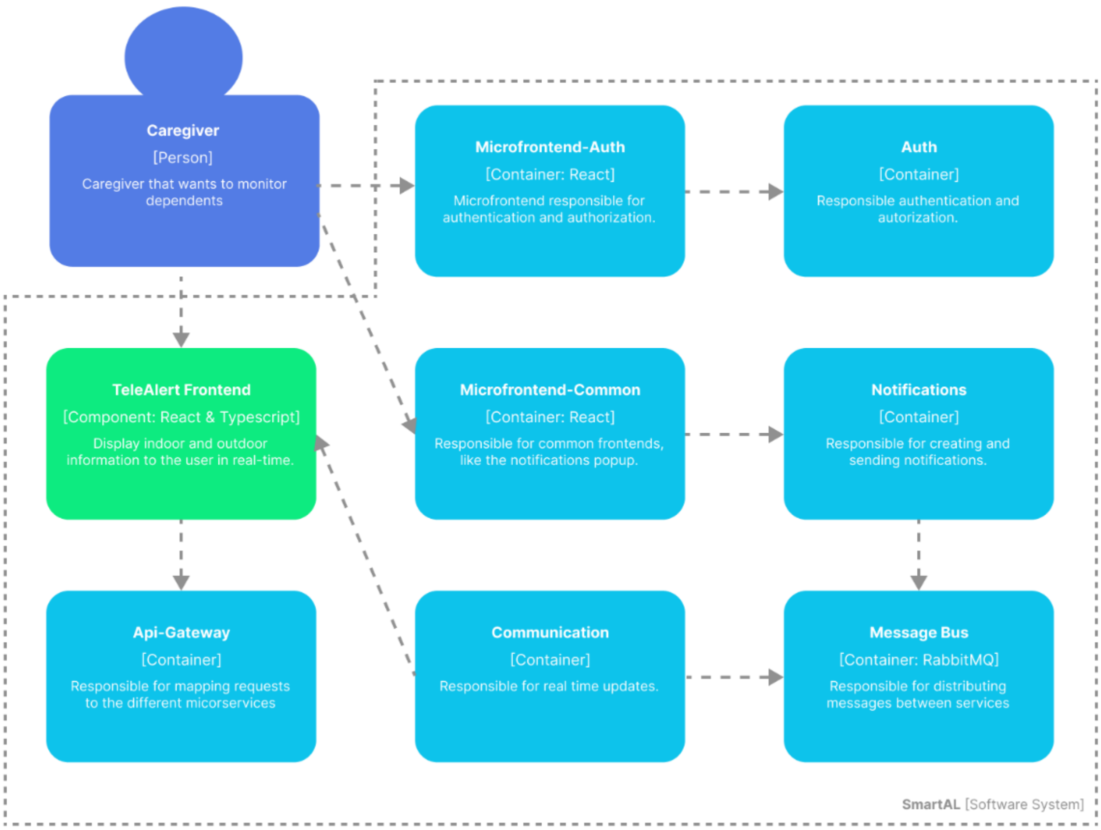


Figure 15: SmartAL Frontend Component Diagram

Following the microservices architecture principle of having a self-contained functionality per service, it was decided to separate the backend into four new services (as can be seen in Figure 16):

- The three collectors, called from now on **Home**, **Indoor,** and **Outdoor**, for simplicity purposes, are responsible for receiving messages from the three outside services and storing them in a database; they should be able to both receive requests coming through the Api-Gateway and send real-time updates through the message queue to the frontend; all the requests directed toward these servers should be authenticated and authorized before being answered.

- The **Rules** server - the three collectors should communicate with the rules engine which is responsible for receiving updates and asking for data to assess risk situations (i.e., predefined conditions. Alerts that result in these situations should be created and sent to the message bus to be later collected by the Notification server and ultimately shown in the TeleAlert application.

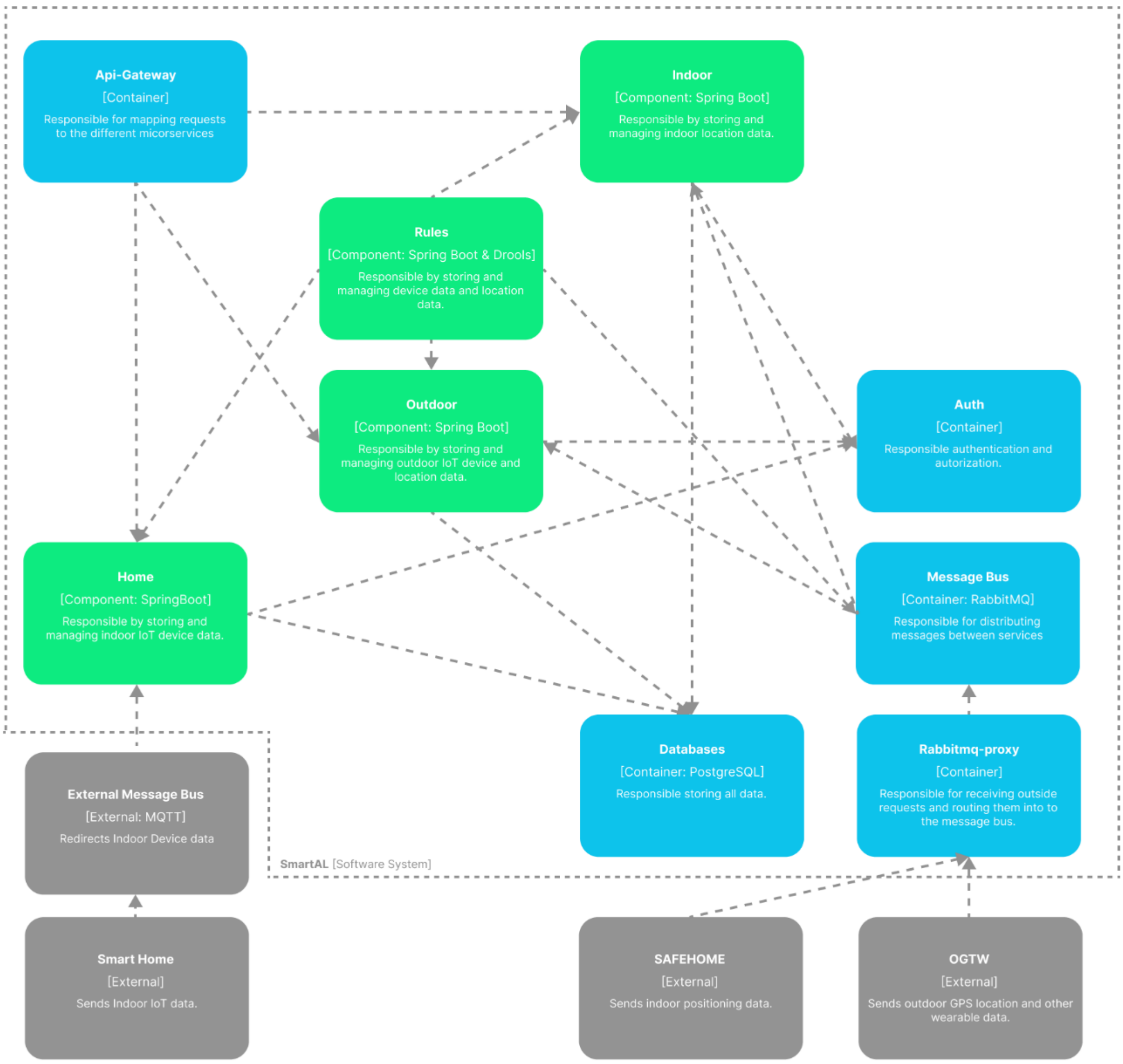


Figure 16: SmartAL Backend Component Diagram

5.7 Database Models

Having presented the project architecture and gained a clear understanding of how SmartAL's microservices interconnect, we can now proceed with the definition of the DB data model. These diagrams were built to accommodate the messages from each external system.

The messages received, however, will not be shown in this section, but in the next chapter dedicated to implementation.

Starting with the SmartHome collector, the below diagram contains all the relevant entities, attributes, and relationships:

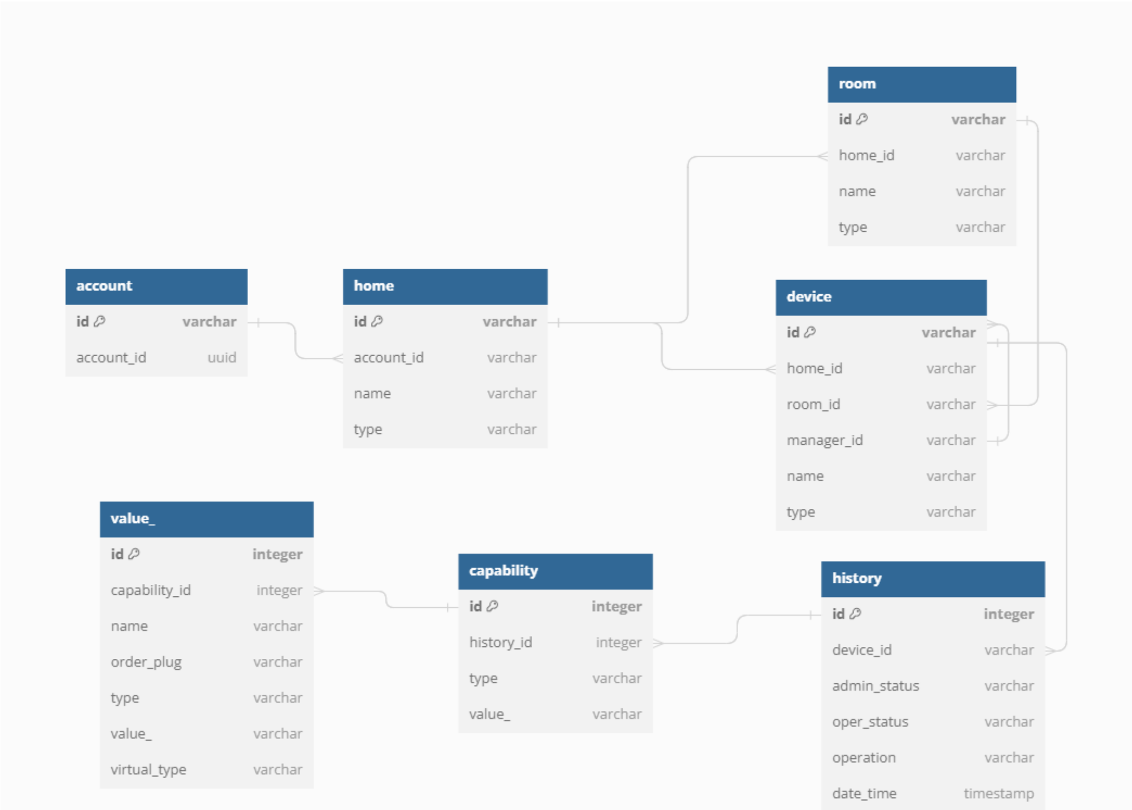


Figure 17: Home Database Diagram

The **Home** diagram is structured like a tree with the caregiver account at the top. It is important to know that it includes an account ID internal to SmartAL and another is used as the primary key which corresponds to the account ID of that same person inside the SmartHome system. Each account can have several homes, even if the SmartHome app does not yet support that option. Each home has rooms and devices, and each room has devices. This is because a device may not have a specific identified division in the house and the home connecting directly to its devices also allows for an easier search. Each device should have a history list that represents all its states through time. Finally, each device has a list of capabilities that can change, and each capability can have a value or a value list.

Moving on, Figure 18 represents the Indoor collector's database diagram:

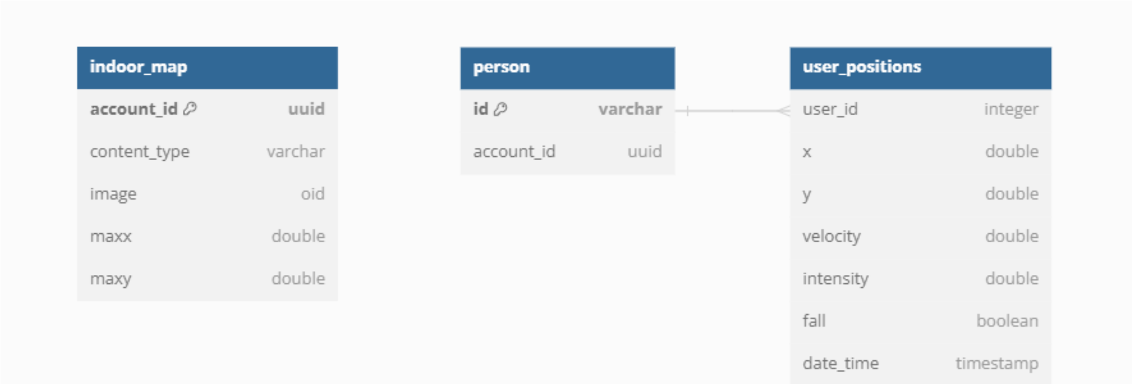


Figure 18: Indoor Database Diagram

The **Indoor** diagram is a bit simpler. The central entity Person (i.e., a dependent) holds an account ID referring to their caregiver's account in SmartAL. Each person entity has a history of user_positions (i.e., dependent's coordinates in the house) at a given time plus some additional data (fall, velocity, and intensity of the steps given in the room). Associated with the account, there is an isolated table unrelated to the messages that will save the image of the plant where the indoor location will be displayed.

Finally, information coming from the **Outdoor** collector database structure is represented in the below diagram:

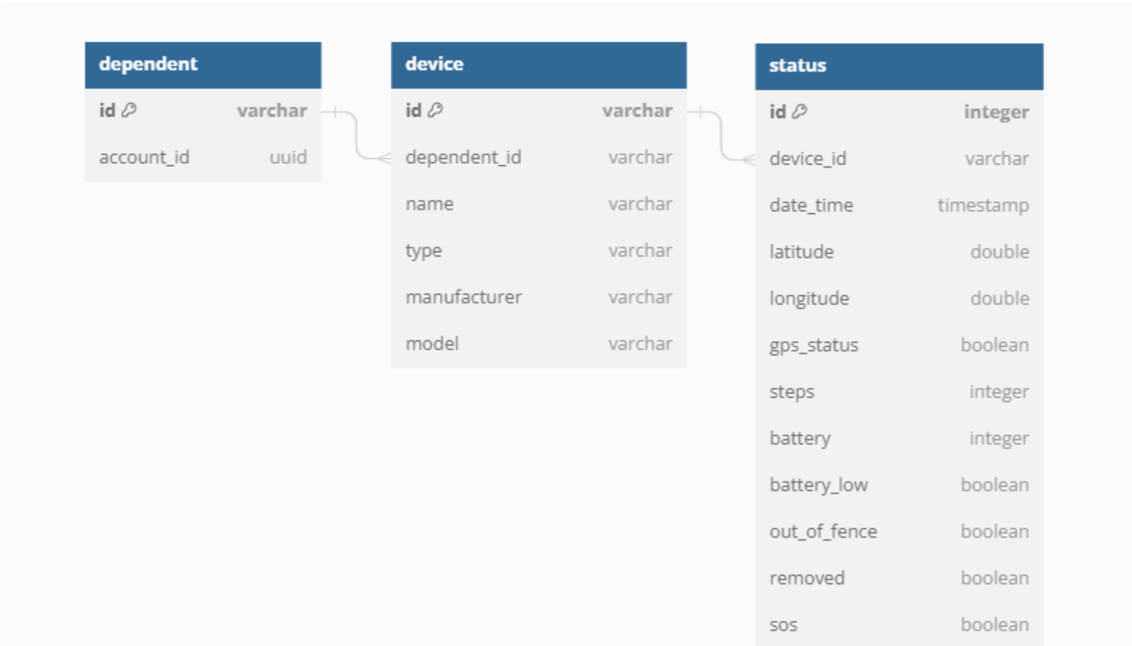


Figure 19: Outdoor Database Diagram

This last diagram contains device information of wearables owned by the dependent and its current. It will not be a very common situation, but each dependent can have multiple devices, and each device has a list of statuses. The location of the user will be the most recent status entry for all of his devices. Some additional information about the user is also provided.

5.8 Mockups

This section shows the main mockups needed to provide the application functionalities that were listed in the requirement section. These offer a basis for understanding how the application should look like. They were used to guide the frontend development. SmartAL's already existing functionality like the notification popup or the login page are not shown in this section so refer to the Implementation chapter for those. These mockups were created using Figma following a preexisting Altice Labs Storybook.

5.8.1 SmartAL Storybook

The SmartAL team defined a storybook to guide the implementation of new frontends. A storybook is essentially a library or a playground for User Interface (UI) components [64]. This library holds information on the general style of the web application, user interface, inputs, templates, and containers.

Under the general style, there are details about the color palette, icons, and typography[40]. On the user interface section, it shows examples of how the icons, buttons, user images, and many other structures should look like. There is also a section that shows all the options for user input ranging from simple text inputs to date and file inputs. Under the template section, some examples of how pages should be structured are shown. And finally, on the container tab, we can see popups and other containers that can be used.

All these examples are reusable and can be called in the React code reducing time spent on styling and formatting.

Regarding the color palettes to be followed, there are 3: brand, feedback, and auxiliary. Only the first two were used since the frontend developed requires few variations in colors.

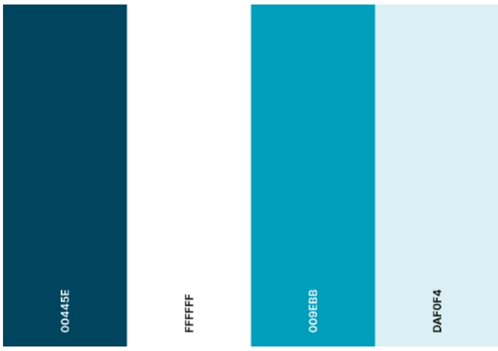


Figure 20: SmartAL brand colors

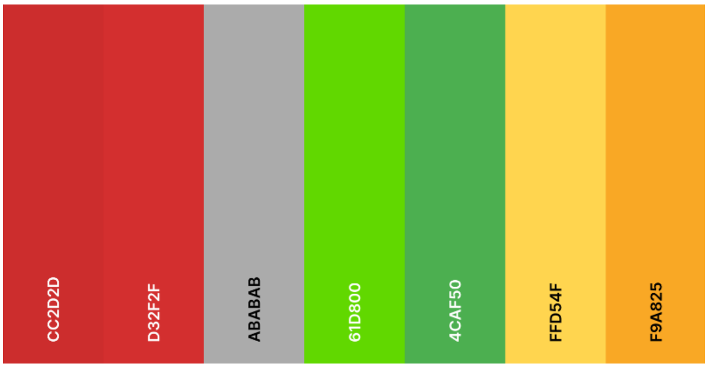


Figure 21: SmartAL feedback colors

[40] font, size and weight of the application texts

5.8.2 Mockup Design and Explanation

In this subsection, a showcase of the TeleAlert application design will be displayed along with a detailed explanation of the different pages of the web application.

The main page was designed to access the location and the device data from both indoor and outdoor sources. So as the location of the dependent is the main focus of the application, it should be the first thing the caregiver sees after logging in. Additionally, alerts are easily accessible on the upright corner of the application using already existing micro frontends, common to other applications. In this new application, both the alert popups and alert history are used. These will be shown later in more detail, in the implementation chapter.

The two main pages, location and devices (icon of a watch on the upright corner), have a common structure with two tabs regarding the indoor (*"Em casa")* and outdoor (*"Fora de Casa")* information. Regarding the location page, we have the example of Figure 22.

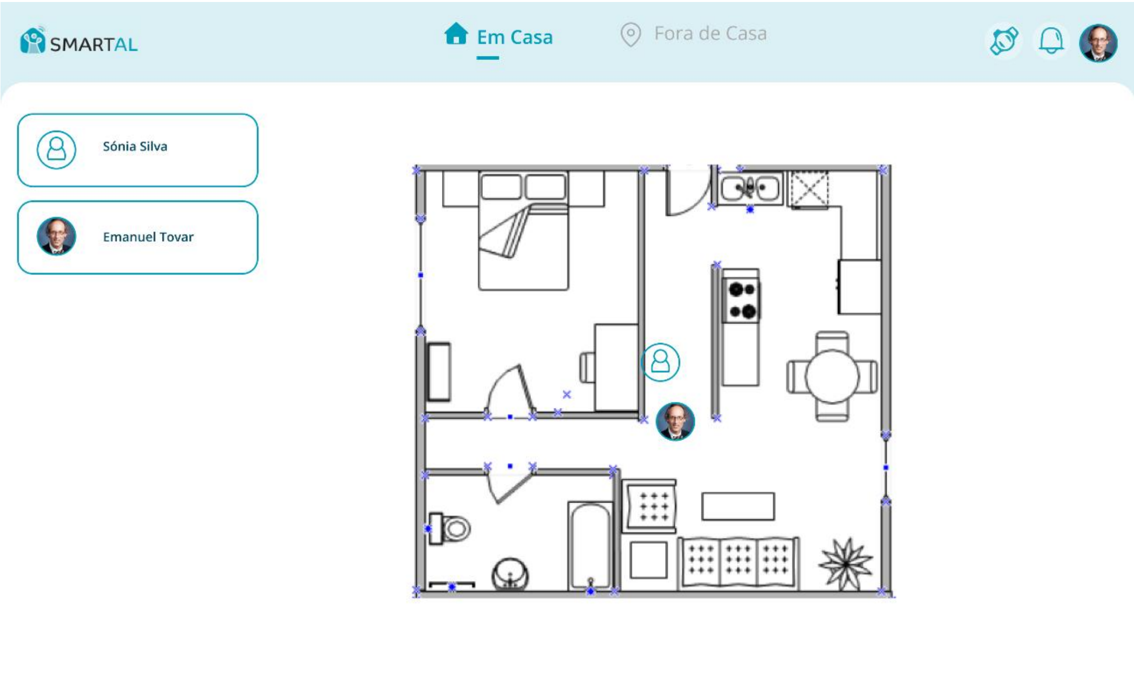


Figure 22: Indoor Location Mockup

In short, on the first screen, the caregiver will be able to see the location of their dependents on the map using different icons/pictures. Lateral tabs are used to distinguish the users. Additionally, the navigation bar contains the SmartAL logo and tabs, highlighting in dark blue the tab the user is currently on. Note that the application base language is Portuguese and so the tabs are named in that same language. On the top corner, there are 3 buttons that, respectively; allow access to the device page; the notification button opens the notification popup with the most recent alerts; and finally, the profile button gives access to the user account data and logout. The last two, as they are part of the common framework, will be shown in the implementation section.

Moreover, an additional screen is needed for the user to add the plant of the house:

Figure 23: Indoor Plant Form Mockup

As can be seen in Figure 23, a file input field is necessary to receive the image (PNG, JPEG …) of the plant. To facilitate the mapping of the location over the image, the user is requested to crop the picture to match properly the house limits. Also, to correctly map the markers of the dependents, the user must provide the dimensions of the house in meters. The user is only allowed to submit the plant when all these inputs are correctly filled with valid information.

Regarding the outdoor location, a map is reproduced as follows:

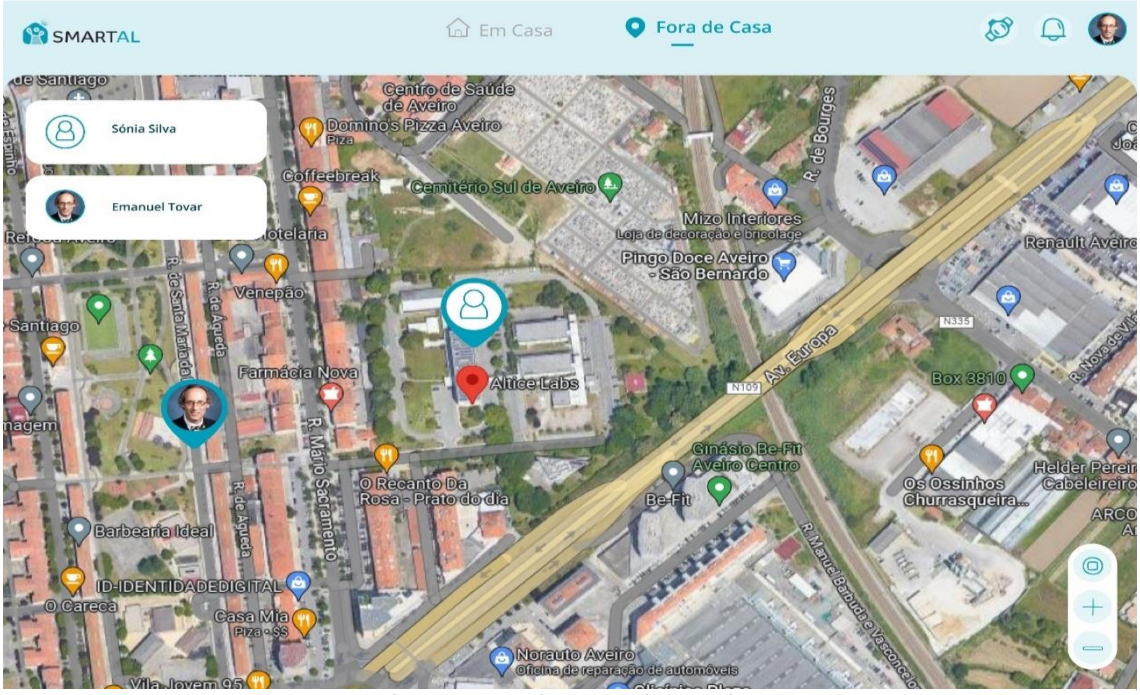


Figure 24: Outdoor Location Mockup

The outdoor screen holds a similar structure to the indoor location. This time, the dependent tabs in the left corner should also be clickable to center the map around the selected person. Also, as in all these kinds of maps, buttons to zoom in and out of the map should exist, and they will be on the lower corner.

Concerning the device screen, it is also divided into two tabs: indoor and outdoor (Figure 25).

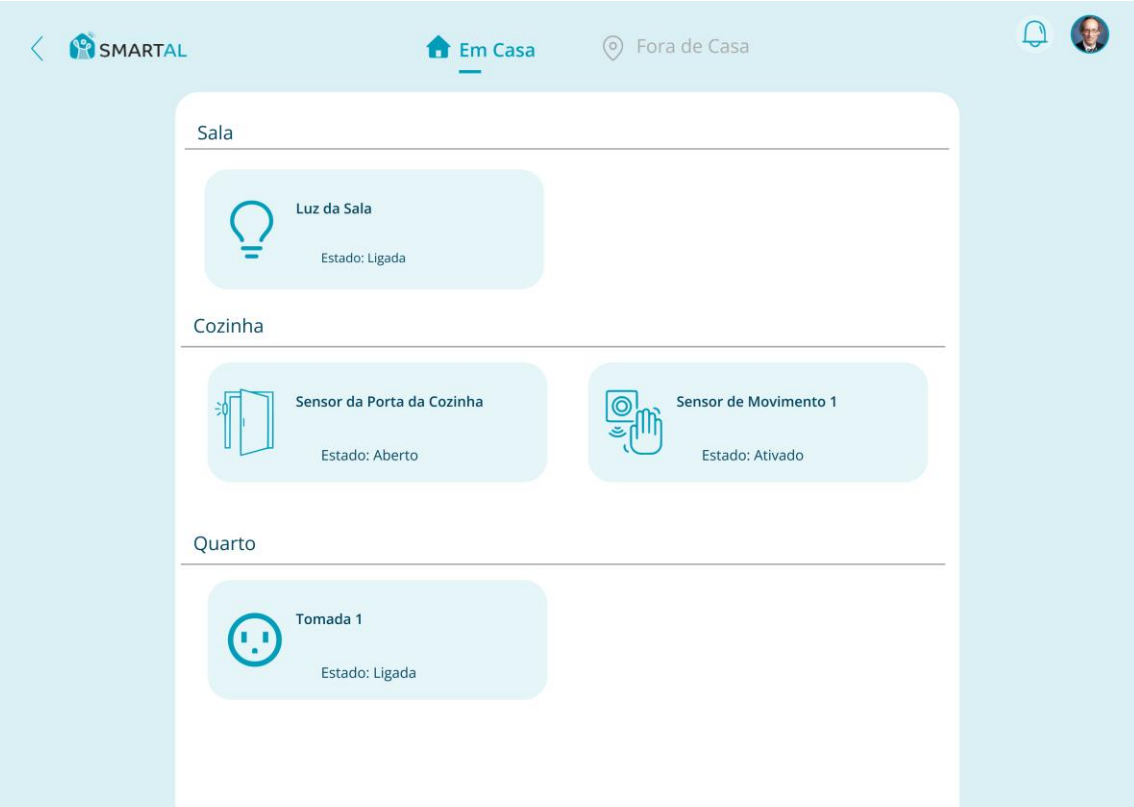


Figure 25: Indoor Devices Mockup

The navbar has a similar structure to the location screen. A back arrow is provided to the main location screen.

The body of the screen consists of sections matching the divisions of the house with devices. Each division has a list of device cards. Each device card displays an icon representing the device and the device name. In addition, it should also display at least information regarding the state of the device.

The outdoor device screen holds a similar structure as can be seen in the below figure:

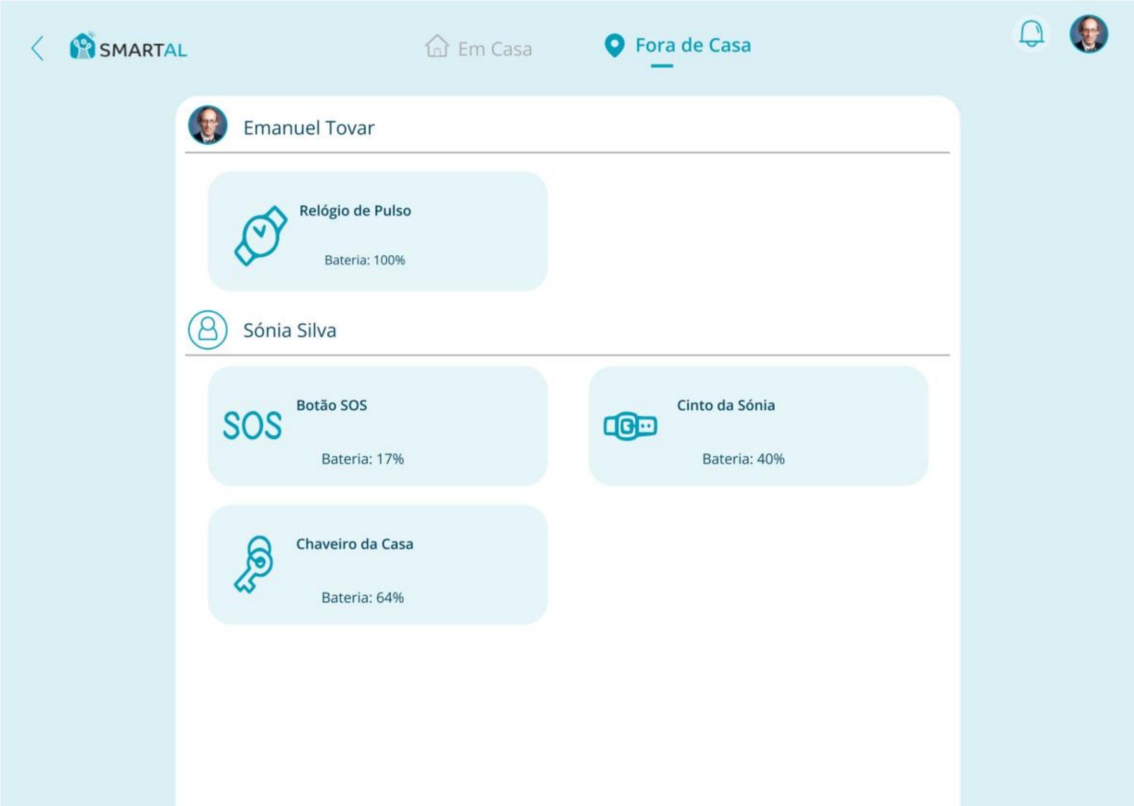


Figure 26: Outdoor Devices Mockup

In the case of the outdoor screen, the devices/wearables are organized by dependent, and if possible, at least the battery should be displayed as additional information. This design allows for further information to be added later if deemed necessary.

5.9 Summary

In conclusion, this chapter provided a comprehensive overview of TeleAlert, encompassing use cases, requirements, architecture, data model, mockups, and technologies to be used. The combination of these essential elements has set a strong foundation for the forthcoming implementation phase. By examining real-world use cases, it was possible to gain valuable insights into users' needs, which impacted the definition of key functionalities and features. The established requirements and architectural design, paired with the selected technologies, ensure a scalable, flexible, and user-centric telelocation platform. The carefully designed data model and user interface mockups further contribute toward an easier implementation phase.

The upcoming Implementation chapter will focus on transforming the conceptual ideas, requirements, and designs discussed in this chapter into a tangible and fully functional TeleAlert service. This phase will involve hands-on coding to make sure the system operates seamlessly. Throughout this process, an iterative approach will be maintained, incorporating regular feedback from the ALB team to address any potential issues and continuously enhance the platform's functionality.

6 Implementation

This chapter is going to showcase all the needed actions to realize the TeleAlert solution and go from conceptual design to a fully functional reality.

It starts by expanding the methodology employed to guide the development process, ensuring efficiency and effectiveness in delivering the desired outcome. Then, the version control practices will be presented that allow tracking changes and maintaining the integrity of the codebase through the development lifecycle.

Furthermore, the deployment strategies for both the frontend and backend components of TeleAlert will be explained. Carefully orchestrating the deployment process will make it possible to ensure the solution is readily accessible and performs as it should in real-world scenarios.

The combination of a well-defined methodology, robust version control, and thoughtful deployment practice will be instrumental in developing TeleAlert. The final objective is to deliver a reliable service that offers users a useful experience.

6.1 Methodology

This subsection describes the methodology employed during the development of the TeleAlert service. The project combined elements of both agile and waterfall approaches to accommodate the initial planning and the evolving nature of the implementation. So, the chosen development method was both iterative and incremental (Figure 27) to satisfy the needs [65].

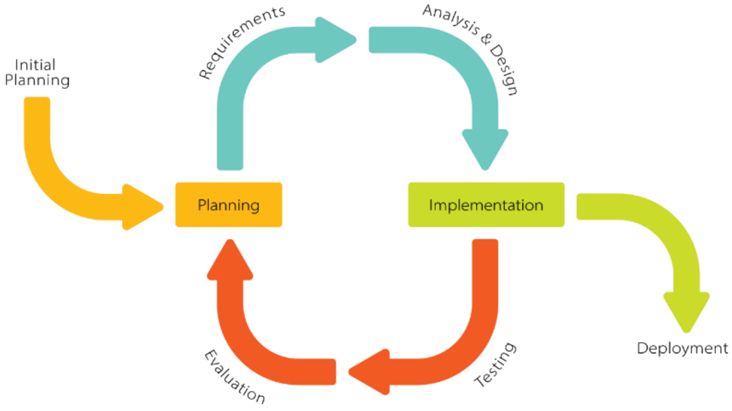


Figure 27: Iterative and Incremental Development

Throughout the development process, regular weekly meetings with the team played a crucial role in adapting to changes. When problems emerged, they were discussed in these meetings, and necessary adjustments were made to incorporate the solutions found.

Each microservice included various cycles of implementation to allow for a better final result. After this phase, each functionality was tested manually while making sure nothing from previous cycles was ruined by the changes made or new functionalities added.

Posterior to testing, the team evaluated the system to make sure everything was going as planned and changes were applied correctly.

In short, the weekly team meetings were used to report on the work accomplished, as well as to gather feedback and possible changes. These meetings were used as an opportunity to discuss progress, address any concerns, and ensure the system followed the requirements. The chosen methodology facilitated the development and focus by breaking down the process into a sequence of smaller tasks. It also allowed for continuous improvement across various aspects, including development, planning, and architecture. This iterative approach was good for ongoing refinement and optimization, resulting in a better application.

6.2 Version Control

The selected version control tool to be used through the development of TeleAlert was **Git**, a version control system created to efficiently manage projects of any size [66]. It also offers a powerful and distributed approach that permits developers to work independently and collaboratively. This flexibility enhances productivity and provides resilience to the project.

Git's powerful branching and merging capabilities enable work on specific features or fixes in isolation, without disrupting the main codebase. This practice minimizes conflicts and allows maintaining a stable main branch while integrating new features through well-managed merges between branches and commits. By effectively utilizing branches, we ensure a streamlined and organized development process.

By having TeleAlert as well as all other SmartAL services available on GitHub, any modifications to the services that interact with TeleAlert become automatically visible, resulting in a streamlined integration process between the various services. Utilizing GitHub as a centralized repository for SmartAL's services enhances communication and coordination among development teams, facilitating efficient collaboration and seamless updates.

6.3 Deployment

Deployment, in the context of software development, refers to the process of making a software application or system available for use in a production environment. It involves taking the codebase and other necessary resources developed during the software development lifecycle and deploying them on servers or cloud infrastructure to make the application accessible to end users.

In the context of a microservices architecture, it means deploying individual microservices independently of each other. Unlike traditional monolithic applications, where the entire application is deployed as a single unit, microservices allow each service to be developed, tested, and deployed as a separate and autonomous entity. During this internship, **Docker** was used to create containers and allow running multiple services on the same machine while still in an isolated environment. It is easier to deploy and orchestrate services using docker-compose. It also facilitated the integration of all the existing services needed for TeleAlert to function as a full system.

Additionally, for frontend deployment, Nginx was used as a front-facing web server and reverse proxy in various situations, including microservices, and traditional web applications. Its ability to handle high loads and optimize application delivery makes it a popular choice for many websites, web applications, and online services. TeleAlert’s frontend and other micro frontends were deployed using a Nginx docker container allowing for client-side requests to be distributed to the appropriate server.

In later phases of production, Docker and Nginx played crucial roles in facilitating the cloud deployment of TeleAlert. Docker's containerization allows packaging the entire application, including Nginx as a reverse proxy and the micro frontends, into lightweight and portable containers. This portability enables the deployment of the same Docker containers consistently across various cloud environments, ensuring a seamless deployment process. The SmartAL team is, however, responsible for the cloud deployment. It was only used in the validation phase of the project to make sure the system performed as expected in a realistic scenario. More about it in Chapter 7 Testing and Validation.

6.4 Backend Development

This section provides an overview of the development process involved in creating the backend collectors and rule engine. By delving into the details, it is possible to gain a comprehensive understanding of how these were built, structured, and interact with each other.

6.4.1 Messages

Several messages containing information related to location and devices were considered. Previously in section 5.7 Database Models, it was showcased the database model and it was mentioned that they were created having in mind the messages from the several services that would be behind TeleAlert, but no messages were shown. In this subsection, an overview of the messages received will be presented.

Starting with the most complex messages, the SmartHome messages related to indoor device information. These messages are sent through the SmartHome MQTT message server, as mentioned in the previous chapter. They came in topics structured in the following manner: **<account id>/<home id>/<operation type>**. Just from the topic alone, it is possible to determine which account and home that update refers to. Additionally, the operation type gives information on what kind of message to expect. SmartHome offers multiple types of messages but only two were relevant to the service being developed. And those were the ones coming from the topics that had **HomeChangedApp** and **DeviceChangedApp** as operation types. The following images show the basic structure of these messages.


```
{
"home": {
  "id": "string",
  "name": "string",
  "devices": [],
  "rooms": [
    {
      "id": "string",
      "name": "string",
      "type": "string",
      "homeId": "string",
      "devices": []
    }
  ],
  "default": "boolean",
  "addons": [],
  "security": {
    "armDelay": [],
    "alarmDelay": [],
    "partners": []
  },
  "type": "string",
  "partners": []
  }
}
```

```
{
  "device": {
    "id": "string",
    "name": "string",
    "operStatus": "string",
    "adminStatus": "string",
    "type": "string",
    "capabilities": [
      {
        "type": "string",
        "value": "string",
        "availableValues": [
          "string",
          "string",
          "string"
        ]
      }
    ],
    "roomId": "string",
    "roomName": "string",
    "isSupported": "boolean",
    "pairedTime": "int"
  },
  "status": "string",
  "stateChanged": "boolean"
}
```


Figure 28: SmartHome Messages: Home Update (right) and Device Update (left)

As can be seen in Figure 28, the Home update sends a tree-like structure that shows the organization of the devices in the house. Each house has a list of rooms, and each room has a list of devices. Devices that do not belong to any room are placed in the device list of the house. This message is sent every time a device is added to a room, changes room, and more importantly, when the account is created, meaning it is the starting message. When the changes are at the device level, the device update is used instead, reducing the amount of information sent with every message. This can be any action done to the device like connecting it or turning it on.

Moving forward, the SAFEHOME messages related to indoor location are simpler. These are deposited by the rabbitmq-proxy in the RabbitMQ message bus of SmartAL under the exchange **indoor.exchange** and the routing key **indoor.location**. They have the following structure:

```json
{
  "user": "string",
  "x": "double",
  "y": "double",
  "intensity": "double",
  "velocity": "double",
  "fall": "boolean"
}
```

Figure 29: SAFEHOME Message

This message simply identifies the dependent it refers to and provides their location while giving additional data relating to walking speed and step intensity. Finally, it also sends information about falls.

Last, there are the new outdoor service messages relating to both outdoor location and wearables. Similarly to the indoor location messages are deposited by the rabbitmq-proxy in the RabbitMQ message bus of SmartAL, but in this case under the exchange **outdoor.exchange** and the routing key **outdoor.event**. They have the following structure:

```json
{
  "userId": "string",
  "device": {
    "id": "string",
    "manufacturer": "string",
    "model": "string",
    "type": "string"
  },
  "date": "string",
  "location": {
    "latitude": "string",
    "longitude": "string",
    "gps_status": "boolean"
  },
  "battery": {
    "life": "int",
    "low": "boolean"
  },
  "activity": {
    "steps": "int",
    "out_of_fence_zone": "boolean",
    "removed_watch": "boolean",
    "fall_alarm": "boolean",
    "SOS_alarm": "boolean"
  }
}
```

Figure 30: Outdoor Service Message

As can be seen in Figure 30, the message indicates the user it belongs to. Beyond that, it shows the date and time it was generated as well as the user location, battery, and activity information.

6.4.2 Collectors

The collectors were the first part of the project to be developed since the rule engine and frontend depend on it. During their development, an event-driven approach was used to design the server to respond to events as they occurred. This also facilitated interaction with other SmartAL services, based on the events the collectors received.

All of the collectors have a similar structure as they hold a similar function in the same framework, Spring Boot. They need to receive, interpret, and save the messages in a database. After that, they can send a message to the rule server and a real-time update to the frontend through the Communication server (refer to the Specification chapter). Having this in mind, the collectors have the following file tree as a base:

```
src
├── main
│   ├── java/pt/ptin/smartal/b2c/<service_name>
│   │   ├── configurations
│   │   ├── controllers
│   │   ├── DTOs
│   │   ├── entities
│   │   ├── repositories
│   │   ├── services
│   │   │   ├── RabbitMQService.java
│   │   │   ├── (MqttService.java -> if needed)
│   │   │   └── ...
│   │   └── Application.java
│   └── resources
│       ├── static
│       ├── templates
│       └── application.properties
└── test
```

Figure 31: Collectors File Structure

The structure is not complex. There is a folder that holds all the configurations of the project related to RabbitMQ, MQTT, JSON Mapper, REST, and even JWT.

The controller folder holds all the REST files that describe the endpoints of the application. These can be called by the frontend of the application but also by the rule server. Later in this chapter, details about the endpoints created will also be covered.

Data Transfer Objects (DTOs) are used to send simplified information to the other services and therefore each collector needs a folder for them. These objects are sent as a response to requests or real-time updates to the frontend.

Last, two folders hold the JPA[41] entities and a repository for each entity. These entities define how the data will be saved in the database and the repositories define the functions to be called for a desired query to the database.

Notable files include the RabbitMQ service that receives and sends messages; Application.java that initiates the application; and finally the application.properties where all the environment variables are saved. In addition, the home collector needs an MQTT service to receive the messages. Ultimately, this structure offers good organization, functionality separation, and ease of search, facilitating development.

6.4.3 Rule Engine Server and Rule Implementation

The Rule engine server's main function as the name implies is to apply rules. It trades requests with all three collectors to make sure it obtains the information needed to perform its task. Its main task is detecting situations of risk regarding the users of the devices and generating notifications to inform the caregiver of those situations. That being said, it still holds a similar structure to the collectors.

```
src
├── main
│   ├── java/pt/ptin/smartal/b2c/rules
│   │   ├── configurations
│   │   ├── controllers
│   │   ├── DTOs
│   │   ├── entities
│   │   ├── repositories
│   │   ├── services
│   │   │   ├── RabbitMQ.java
│   │   │   ├── DroolsService.java
│   │   │   └── ...
│   │   └── Application.java
│   └── resources
│       ├── rules
│       ├── static
│       ├── templates
│       └── application.properties
└── test
```

Figure 32: Rules Server File Structure

Similarly to the controllers, there is still a RabbitMQ service file but in this case its purpose changes. Now that file's function is to feed notification objects into SmartAl's message bus for the Notification service to redirect to the correct frontend service and user.

Additionally and central to this service, the Drools service file calls the rule files located in the resource/rules folder and applies the rules present in that file to the objects passed into the rule.

Several rules were created during this internship. Some are simpler than others, but all of them attempt to detect a risk or situation that might or may already be affecting the

[41] Java Persistence API

dependent. Some rules are applied directly to updates however some need more information to be applied. As mentioned in section 5.5 Technologies, it was decided that the rules were to be implemented using Drools. In Drools, the rule file has a specific structure to be followed.

```
//import all java libraries needed to the file
import pt.ptin.smartal.b2c.rules.DTOs.PersonDTO;
import pt.ptin.smartal.b2c.rules.DTOs.PositionDTO;
import pt.ptin.smartal.b2c.rules.services.NotificationService;
import java.util.UUID;
import java.util.Date;

// declare global variables for the file
// notification service is the java service that sends notifications of every type
global NotificationService notificationService;

//declare the dialect to be used in the file
dialect  "mvel"

//from here on write rules
//make sure they have different names as it severs as their id
rule "Check if a person fell"
when
    $person: PersonDTO()
    $position: PositionDTO(fall == "true") from $person.userPositions
then
    notificationService.sendFallNotification($person.accountId, "high", new java.util.Date(System.currentTimeMillis()), $person.id)
end
```

Figure 33: Drools File Structure

Figure 33 shows that the Drools files normally start with all the imports they need to run similarly to a normal Java file. Right after, global variables are declared. In this project's scenario, the notification service is used to send all types of alerts. Later in this chapter, the notification types will be shown but for now, the focus will be on the rules.

After the global variables are declared, the dialect of choice needs to be selected. In this dissertation, mvel was selected. After that, the developer can write any rule he wants. One example can be seen at the end of the file. This rule needs to receive a PersonDTO device to run correctly. When firing the rules, these objects need to be placed in the working memory of the file builder along with the globals needed. The following picture depicts the container creation and variable insertion for this file.

```
public void applyUpdatePersonRules(PersonDTO person) throws IOException {
    //create a kie container for a file
    KieContainer kieContainer = kieContainer(filePrefix:"UpdatePersonRules");
    try {
        // start a session for that container
        KieSession kieSession = kieContainer.newKieSession();

        //insert variables
        kieSession.insert(person);

        //insert globals
        kieSession.setGlobal(identifier:"notificationService", notificationService);

        //fire the rules
        kieSession.fireAllRules();

        //dispose of the session
        kieSession.dispose();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figure 34: Rules Call Example

Figure 34 shows how the rules are fired. First, a container for a file is created, then a session for that container. Both the objects that are going to be evaluated by the rules and the globals for the rule file are then added to the session. Only then, the rules can be fired. Finally, the session is disposed of.

It is important to understand the structure of a rule file and how the rules are triggered. The basic rule has the following structure:

```
rule "rule name"
when
    //condition
then
    //action if the condition is met
end
```

Figure 35: Basic Rule Structure

A rule name uniquely identifies the rule within the file. If there are two rules with the same name an error will be displayed when the container is built. The rest is similar to the structure of an if in any other programming language. In the *when* section, the condition to be tested is written in the chosen dialect (in this internship, mvel). If the condition is verified, the Java code written in the *then* section runs. Here, the code to be run will always be a function from the notification service class to create and send a notification to the message bus (refer back to Figure 32 for an example). The alerts, however, will only be covered later in this chapter in section 6.6 Integration.

Many rules were implemented during this internship, but this section will only go over the used templates. All rules implemented can be obtained from changes to these templates. Several objects will be mentioned during the rule template showcase so refer back to section 5.7 Database Models to review these structures. The first template refers to an indoor service update.

```
rule "Check if something happened to the dependent inside"
when
    $person: PersonDTO()
    $position: PositionDTO(/*condition on the current dependent position*/) from $person.userPositions
then
    //send notification (currently only for falls)
end
```

Figure 36: Indoor Update Rules Template

The rule type in Figure 36 receives an object that represents the dependent and was sent from the Indoor collector. It tests for conditions within the current dependent position. Currently, only this can attest to falls. Next the Outdoor update rule template:

```
rule "Check something happened to the dependent outside"
when
    $dependent: DependentDTO()
    $device: OutdoorDeviceDTO() from $dependent.devices
    $status: StatusDTO(/*condition on the current status of the user*/) from $device.statusList
then
    //send notification in case the condition verifies it self (SOS, fall, left geofence, removed device .....)
end
```

Figure 37: Outdoor Update Rule Template

This rule holds a similar structure to the one before. For an outdoor update, it checks the current user status through several conditions like falls, SOS calls, dependent removing a device, low battery on the device, bad GPS connection, and others. (This will also be covered later in this chapter, in the integration section like the other alerts).

The next rule template refers to an update from the Home collector.

```
rule "Check if any unexpected activity happened"
when
    $device: DeviceDTO(id == /*device id*/)
    $history: HistoryDTO(
                        dateTime.after(new Date(dateTime.getYear(), dateTime.getMonth(), dateTime.getDate(), 0, 0)) && //start time
                        dateTime.before(new Date(dateTime.getYear(), dateTime.getMonth(), dateTime.getDate(), 8, 30)) && // end time
                        dateTime.getDay() in (0, 1, 2, 3, 4, 5, 6) //days of the week this gets tested (sunday=0, monday=1 ......)
                    ) from $device.historyList
    $capability: CapabilityDTO(type == "power", value == "on") from $history.capabilities // capability to be tested
then
    //send notification if the device update shows unexpected activity
end
```

Figure 38: Home Update Rule Template (Unexpected Activity)

Figure 38's rule, as complex as it looks, only tests for unexpected activity in the house. For a certain device, if there is activity between two time frames where it should not (e.g., light on during the night) the rule will be activated. Additionally, it can be configured for only some days of the week.

The remaining templates left in this chapter are no longer referring to updates from the collector. These kinds of rules need to be fired every minute and they need the information on the whole history of a device or a dependent. So, a request is made to the collectors for these rules in specific.

```
rule "Check if dependent has missed an activity in their routine"
when
    $device: DeviceDTO(id == /*device id*/)
    not( //cause what's being checked is if the dependent did not activate something between the two times
        $history: HistoryDTO(
            dateTime.after(new Date(dateTime.getYear(), dateTime.getMonth(), dateTime.getDate(), 10, 0)) && //start time
            dateTime.before(new Date(dateTime.getYear(), dateTime.getMonth(), dateTime.getDate(), 12, 30)) && //end time
            dateTime.getDay() in (0, 1, 2, 3, 4, 5, 6) //days of the week this gets tested (sunday=0, monday=1 ......)
        ) from $device.historyList and
        $capability: CapabilityDTO(type == "power", value == "off") from $history.capabilities //check the device never activated
    )
    eval(new Date().getHours() == 12 && new Date().getMinutes() == 30) // only send notification when the time interval ends
then
    //send notification
end
```

Figure 39: Routine Analysis Rule

The rule represented in Figure 39 tries to detect a deviation from a dependent's routine. From a certain home service device, it checks its whole history to know if it was ever activated, and if it was not, the rule fires a notification. An example of an application can be a motion sensor placed in the kitchen not activating when the dependent was supposed to take their medication at a certain time. These rules can also be configured to be tested only on some weekdays.

The last rule template tries to check if a dependent is walking as they normally do. The intensity or velocity of the walk can be an indicator that something is not right. So, for the last *n* minutes, this kind of rule checks if the dependent's velocity and intensity are way below or above a baseline. This baseline is calculated by creating several buckets for both intensity and velocity and then distributing the values through all those buckets. The bucket with the most values has its average taken and used as a baseline. This method allows for outliers to be excluded from the baseline value for those parameters. Figure 40 shows the template for the rule:

```
rule "Check if the dependent is walking slower, faster, heavier or lighter than normal"
when
    $person : PersonDTO()
    $baseline : Baseline() // baseline object that holds the baseline velocity and intensity

    $recentPositions : List() from collect(
        PositionDTO(
            dateTime >= new java.util.Date(System.currentTimeMillis() - (9 * 60 * 1000)) //last n minutes n=9
        ) from $person.userPositions //list user positions that are in the last n minutes
    )

    $filteredPositions : List() from collect(
        PositionDTO(
            velocity < $baseline.getVelocity() - 1, // can also be getIntensity and a bigger or smaller difference from the baseline
            dateTime >= new java.util.Date(System.currentTimeMillis() - (9 * 60 * 1000))
        ) from $person.userPositions //list user positions that are in the last n minutes and the speed is way below the baseline
    )

    eval($recentPositions.size() > /*number of readings for the rule to activate*/ && $recentPositions.size() == $filteredPositions.size())
then
    //send notification if all the positions in the last n minutes have been out of the norm
end
```

Figure 40: Walking Manner Rule Template

For now, these were the rules templates defined. However, the system is scalable allowing many other rules to be added in the future with a combination of more data to the collectors and a deeper analysis of the data already being received.

6.4.5 Authentication

As mentioned in the Specification chapter TeleAlert needs to be able to use JWTs for user authentication and authorization. JWTs consist of three parts: the header, the payload (containing user information such as the account ID and role of the user in the application), and the signature. The authentication process using JWT is illustrated in the figure below.

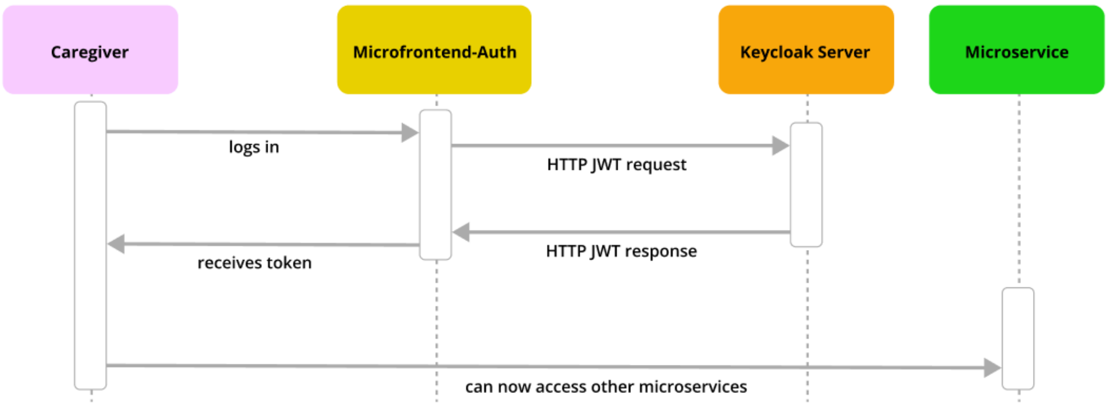


Figure 41: JWT sequence diagram

When a user logs in successfully using the Microfrontend-Auth component, the API generates a JWT and returns it to the browser, where it is stored in either local storage or cookies. Subsequently, when the user makes API calls, the JWT is included in the request header, and the system verifies its validity. Each endpoint then confirms if the token is invalid or non-existent, denying the request if any of those situations happen. Furthermore, this ensures that users only interfere with or access their account information.

These tokens also allow for authorization making sure users from a specific role can only access the endpoints available for that role. This feature will not be used in TeleAlert since it is a caregiver "only" application, but it is still important to mention as the application may allow access to dependents in the future.

To bolster JWT security, a timeout of 5 minutes has been decided by the SmartAL team when configuring their keycloak server. After this period, the JWT becomes invalid. To handle this, a refresh token is also sent to the browser and stored. When the JWT expires, an automated API call triggers a designated endpoint responsible for refreshing the JWT. This process issues a new refreshed JWT along with a new refresh token.

By utilizing JWT-based authentication along with these additional security measures, the API establishes robust user authentication and protects sensitive user data from unauthorized access. In addition, JWT enables a stateless server architecture, eliminating the need to maintain user-related state information and enhancing scalability. Adopting JWT paves the way for the usage of a single sign-on functionality, allowing users to authenticate once and gain seamless access to multiple applications or services. In summary, JWT-based authentication ensures security, scalability, and potential for a convenient user experience.

6.4.6 Endpoints

The collectors and rule engine server both have endpoints that can be called in case some information needs to be traded between them or with the user of the web application. In this subsection, all of these are going to be listed and explained.

While naming the endpoints, it was attempted to follow the best practices and naming conventions of REST API URIs like using nouns to represent resources instead of verbs, no trailing forward slashes, hyphens instead of camel case for readability, no underscores or upper-case letters and others [67].

Starting with the home collector these were the endpoints developed:

- General /:
  - GET /health-check: serves as a health check to the API. Returns code 200 if the app is running as expected.
- Device /device:
  - GET /: returns all the devices saved to the database and is used by the rules servers to apply routine rules.
  - GET /account: returns all devices associated with the account retrieved from the JWT token. It is used by the frontend and the user needs to be authorized and authenticated.

Following with the indoor collector endpoints:

- General /:
  - GET /health-check: serves as a health check to the API. Returns code 200 if the app is running as expected.
- Location /location:
  - GET /: returns all the dependents saved to the database and is used by the rules server to apply walking rules.
  - GET /account: returns all devices associated with the account t retrieved from the JWT token. It is used by the frontend and the user needs to be authorized and authenticated.
- Indoor Map /indoor-map:
  - POST /: saves indoor map image and information about the map to the account that is retrieved from the JWT token. It is used by the frontend and the user needs to be authorized and authenticated.

- GET /: returns the indoor map image from the account that is retrieved from the JWT token. It is used by the frontend and the user needs to be authorized and authenticated.
- GET /data: returns the indoor map information from the account that is retrieved from the JWT token. It is used by the frontend and the user needs to be authorized and authenticated.

Third, the outdoor collector endpoints:

- General /:
  - GET /health-check: serves as a health check to the API. Returns code 200 if the app is running as expected.
- Location /location:
  - GET /account: returns all devices associated with the account retrieved from the JWT token. It is used by the frontend and the user needs to be authorized and authenticated.
- Device /device:
  - GET /account: returns all devices associated with the account retrieved from the JWT token. It is used by the frontend and the user needs to be authorized and authenticated.

Finally, the rules engine server:

- General /:
  - GET /health-check: serves as a health check to the API. Returns code 200 if the app is running as expected.
- Update /update:
  - POST /outdoor: applies rules to an outdoor update.
  - POST /home: applies rules to a home update.
  - POST /indoor: applies rules to an indoor update.

As can be seen by the endpoints listed above the servers are not really endpoint intensive yet. Very few are required to satisfy what the frontend needs, but more are vital to satisfy the rule server. The current organization allows for easy expansion of the system to support more endpoints for new functionalities of the frontend showcased in the next section.

6.5 Frontend Development

This subsection presents an overview of the frontend's organization and the different pages and components that integrate the application. The added micro frontends will also be shown during this chapter. Additionally, the web pages will be presented in a walkthrough manner to get a better understanding of how the app can be used.

6.5.1 Project Structure

The frontend development was realized using React.js as mentioned in Chapter 5 Requirements and Architecture. The following file structure was created by the SmartAL team and made available to this internship.

```
.
└── main
    ├── public
    │   ├── languages
    │   └── logos
    └── src
        ├── components
        ├── dtos
        ├── hooks
        ├── mappers
        ├── pages
        ├── requestBodies
        ├── schemas
        ├── services
        ├── store
        ├── styles
        ├── types
        └── utils
```

Figure 42: Frontend File Structure

The project has two main folders: public and src. The public folder holds the languages and logos folders. The languages folder holds files that will be used in the future to translate the app. Regarding the logos folder, SmartAL has already built-in icons and logos that are accessible through the storybook but depending on the new service it may need to be extended with additional images.

The src folder holds most of the code that was produced during the internship. The components folder holds the components created during this project and the pages folder has the main components that build the web pages of the application.

The service folder is the one that contains all the requests made to the backend allowing for components to call these interfaces when needed.

The schema folder has the objects represented as they come from the backend with all the information associated with them. These can be converted into DTOs and back to schemas by the mappers present in the mappers folder. DTOs are the objects that simplify the amount of information passed into the components making them more lightweight and, in this project, they are located in the DTOs folder.

The project already had the basic style associated with the components. For example, the headers already have a predefined font, weight, and size and backgrounds already have a base color. However, new styles needed in the project are saved in CSS files located in the styles folder.

There are still some other folders. The store folder, for example, holds the socket interfaces for now. These folders can later be needed for other functionalities.

6.5.2 Components

The frontend of TeleAlert consists of several essential components, each designed to provide users with a seamless and intuitive experience. These components play distinct roles in ensuring good application usability and functionality. Delving into each of these components will help in understanding how they contribute to the overall user experience.

First and central to this application is the navbar. It is one of the main components that compose the pages of the application, and it usually carries the tabs that allow changing from indoor to outdoor information. It also holds the SmartAL logo and the buttons that access the common micro frontend popups: account and alerts.

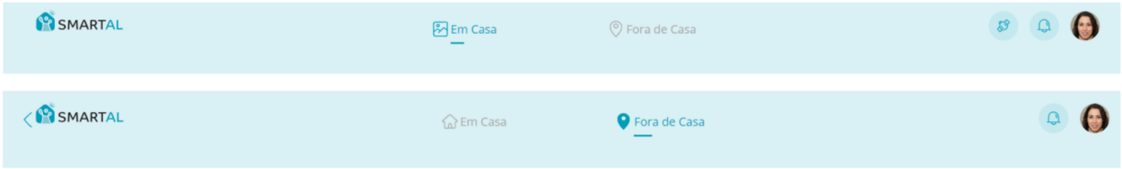


Figure 43: TeleAlert Navbar

In Figure 43, there are examples of the navbar. The top one corresponds to the location screen when the indoor tab is selected and the bottom one to the outdoor device screen. The main difference between them is the tab selected and depending on the case, the availability of the device button or an arrow to go back to the location screen. A tab change causes a change in the URL path of the page triggering the update of the page according to the tab selected. It is worth mentioning that by the time this image was taken the icon of a filled house was still not added to the storybook and so a placeholder was used instead. This is going to be recurring in the next components too.

Next, there is the plant form. This form was created using the Form component included in the existing storybook library. It allows for the insertion of a single image file and the corresponding length and width of the house.

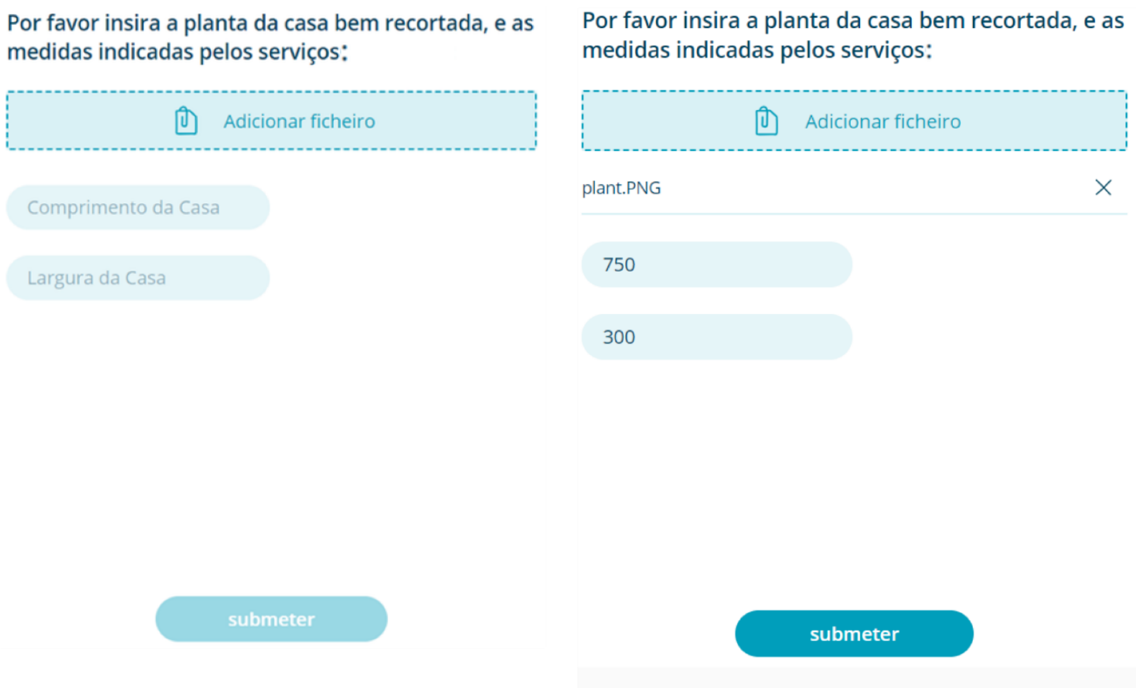


Figure 44: Plant Form

As can be seen in Figure 44, the form can only be submitted when filled. The image must be properly cropped for the pins to be mapped correctly on top. Additionally, the values for the dimensions of the house must match the ones provided by the SAFEHOME system or the mapping won't work as expected. This is not practical for the user, but other solutions are going to be explored by the team in the future. After the plant is submitted correctly, the page jumps to the indoor location immediately.

The Person Tab component will be present on both the indoor and outdoor maps allowing for the caregiver to easily see who is present on the map and their name (Figure 45).

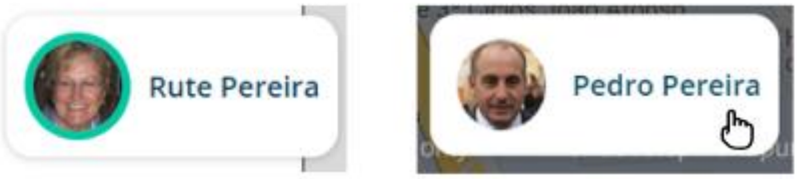


Figure 45: Indoor(left) and Outdoor(right) person tabs

The person tab is composed of the dependent's picture and name. In the indoor version, the user picture is surrounded with a color generated from the dependent's ID, to help distinguish the users on the plant map, in case no profile picture is provided. The outdoor does not need colors as the tab is clickable and centers the outdoor map around the user that was selected.

In some scenarios, the dependent might not be registered in SmartAL so the following will be shown by default:

Figure 46: Unregistered User Tab

The plant container is simply an image and some markers similar to the tab's pictures (Figure 47).

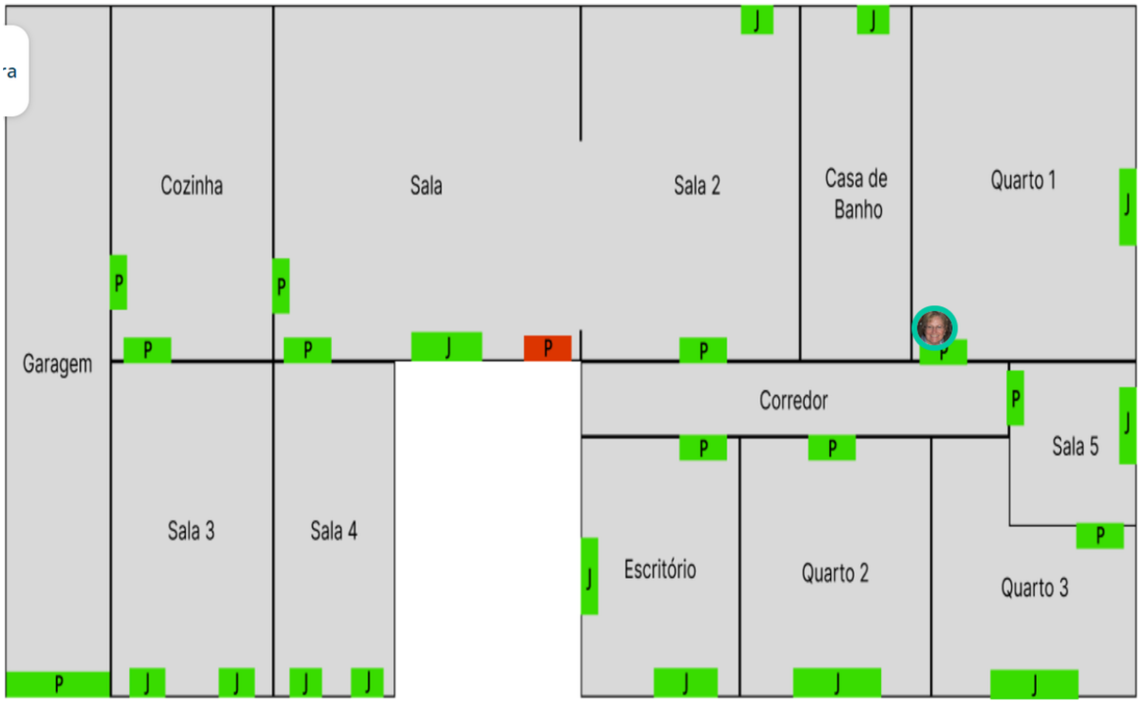


Figure 47: Plant Container

The users are mapped based on the size ratio of the image when it is loaded with the dimensions of the house provided in the plant form. Note the dependent marker has an associated color based on their ID matching the user tab. In case the caregiver decides not to add pictures to the dependents the markers will all look the same. In the future, name initials can be used instead. For now, only one house with multiple users is considered and it is not possible to edit the house plant through the application, as these are not essential to the proof of concept. However, these functionalities can easily be added in the future.

Regarding the outdoor map container, it was built using the Google Maps library available for React. This already includes buttons for zooming in and out of the map and a full-screen button. The library easily allows adding markers for each user by providing their latitude and longitude, as can be seen in Figure 48.

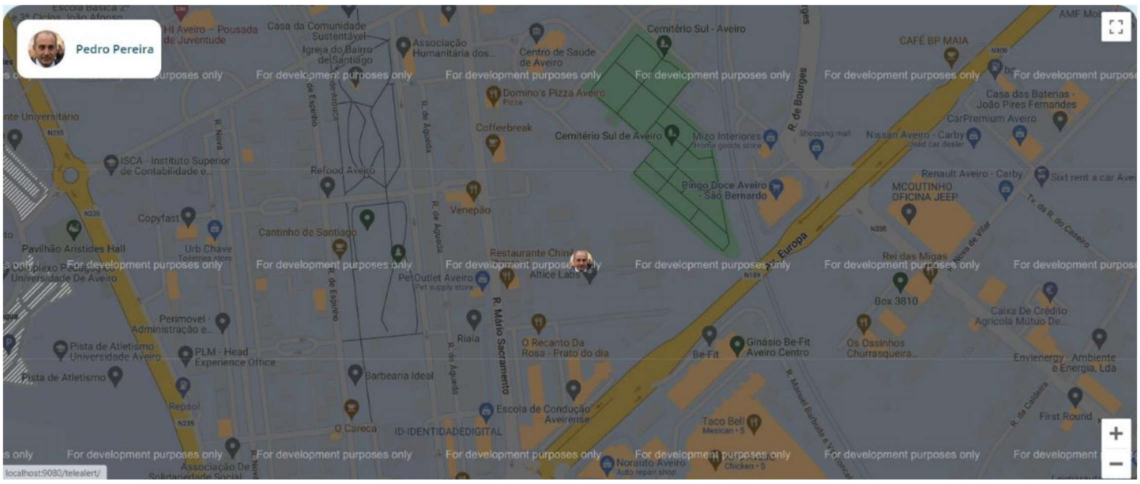


Figure 48: Outdoor Map Component

It is relevant to mention that the map will not always have those watermarks on it. For them to disappear the SmartAL team will have to provide a valid Google API key to the component.

Moving on to the devices part of the application, and as mentioned in the specification chapter, the devices are organized separately by sections.

Figure 49: Indoor (top) and Outdoor (bottom) Device Sections

As can be seen in Figure 49Figure 49, the indoor sections are room-related, and the outdoor sections are dependent-related. The difference in structure is that the dependent-related section has an image or placeholder to identify the user to it. Additionally, when the device has no room, it will be put in a section called *Sem divisão* (without division). This will not happen to outdoor devices as they always have an associated user.

Within each section, there will be a list of device cards containing the most pertinent information for each device: name, icon, and the most relevant fields.

Figure 50: Indoor (left) and Outdoor (right) Device Card

In Figure 50, it is easy to verify that there is little difference between the indoor and outdoor device cards. The indoor sensors card will have the following possible values: lamps and plugs either *ligado* (on) or *desligado* (off); motion or contact sensors *alarme (alarm)* or *normal* (normal); gateways will be *conetado* (connected) or *disconetado* (disconnected). There are still no icons in the storybook for the indoor devices so a placeholder is being used. Regarding the outdoor devices, the battery was chosen to be used as additional information. Nevertheless, more information can be added to each card as needed.

In addition, two other components were developed by the SmartAL team that were coupled to the TeleAlert's frontend. The first one is a popup that gives access to both the account information and the logout button as seen in Figure 52.

Figure 52: Account Popup

The second is the notification popup. This popup lists the most recent alerts. This way the caregiver has quick access to the most important notifications and can act on them if needed. Additionally, this popup will also have a button that gives access to older notifications: alert history. In the future, the team can also add a feedback button to these notifications (Figure 51).

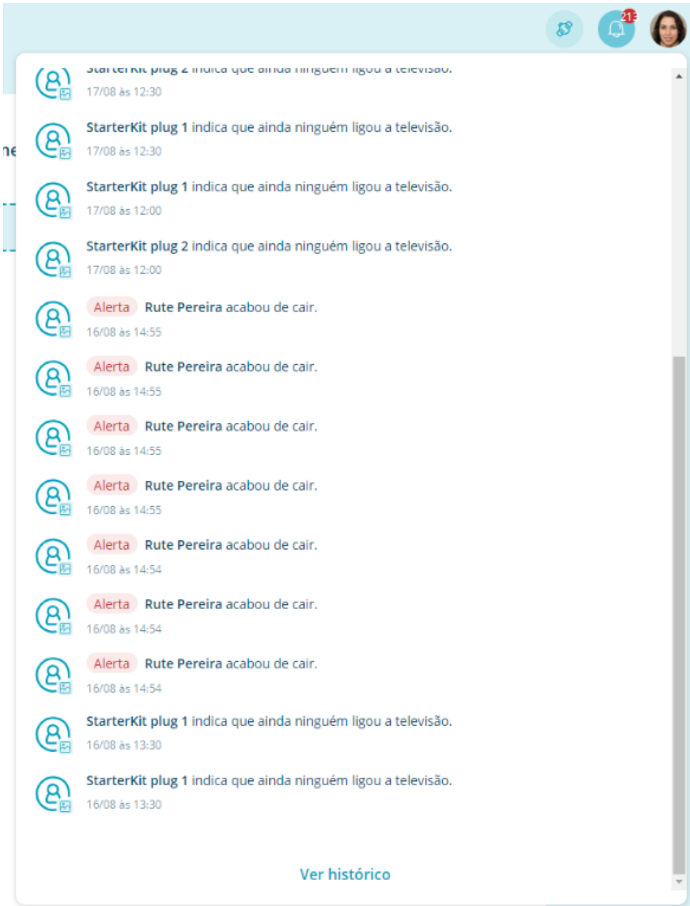


Figure 51: Alerts Popup

6.5.3 User Flow

Having covered the key components constituting the TeleAlert frontend, the subsequent focus will be on the usability flow that creates a comprehensive and user-friendly application. In the following subsection, a closer examination of the already described web pages and their significance will be presented in a more demo-like manner to better show how the screens are connected.

Like many other web applications, upon accessing the TeleAlert App, users are greeted with a login screen where they can securely authenticate their accounts (Figure 53). The login process ensures that only authorized personnel can access the information displayed in the application.

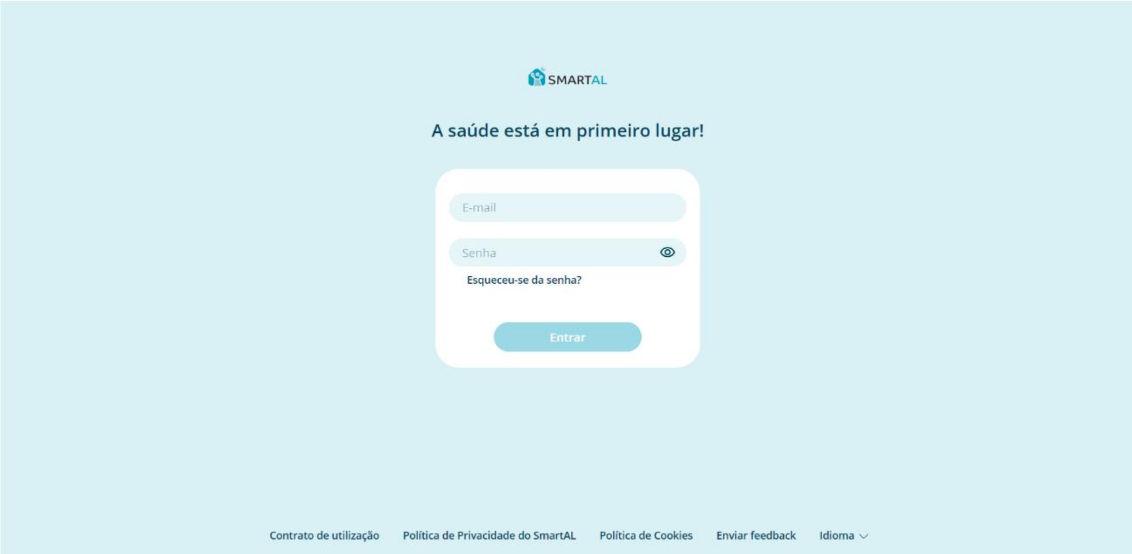


Figure 53: TeleAlert Login Screen

After logging in, if the caregiver never added a plant image, the plant form is presented as part of the indoor location screen, since this is the first screen shown after the user opens the app.

The caregiver should then provide a plant image. After filling in the form (Figure 54), the user should be able to submit it and see immediately their dependents represented on top of the house plant (Figure 55).

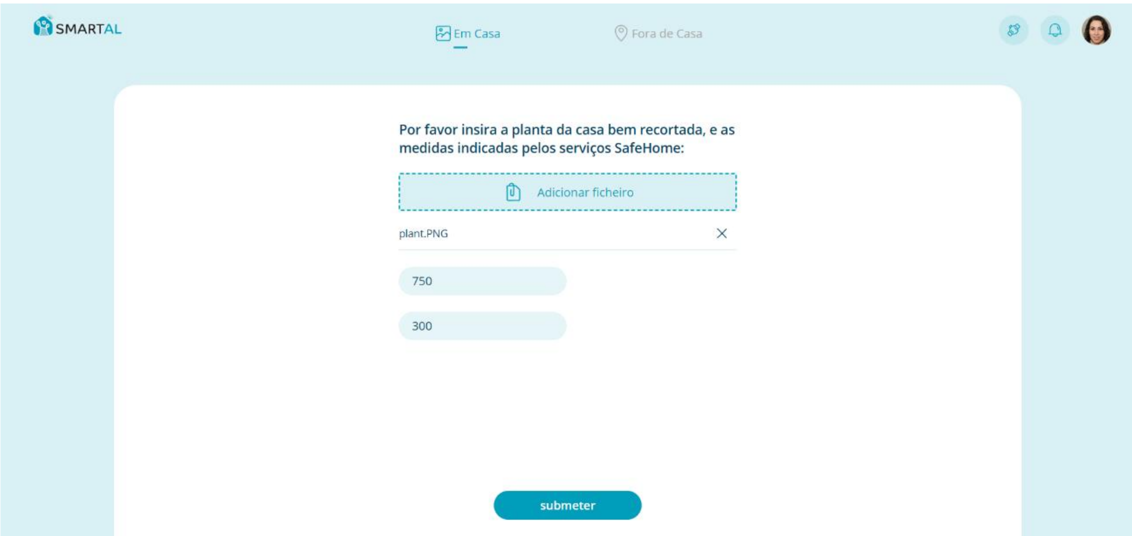


Figure 54: Plant Form Screen

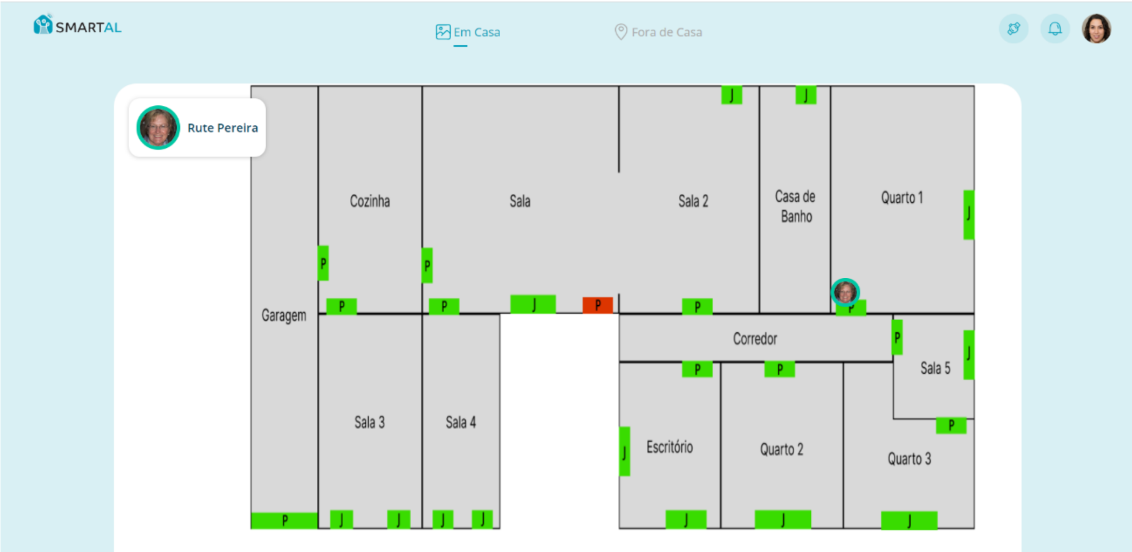


Figure 55: Indoor Location Screen

If the dependent moves outside, the caregiver can immediately switch to the outdoor tab to better accompany them (Figure 56).

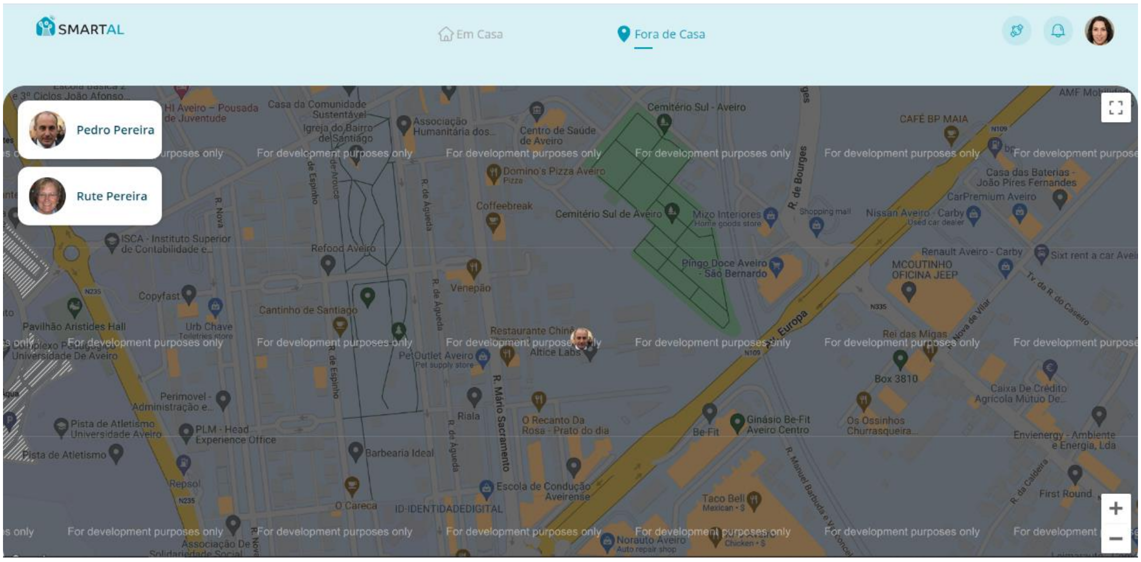


Figure 56: Outdoor Location Screen

In the meantime, the caregiver might want to make sure everything is okay with the dependents' devices both inside and outside the house. So, the device button on the top corner will allow the user to change the view to the device screen.

Figure 57: Indoor Device Screen

The first screen will contain the indoor devices (Figure 57). On this screen, the user will be able to know the status of the devices that are configured inside the house using the existing SmartHome app. The user can then switch to the outdoor tab.

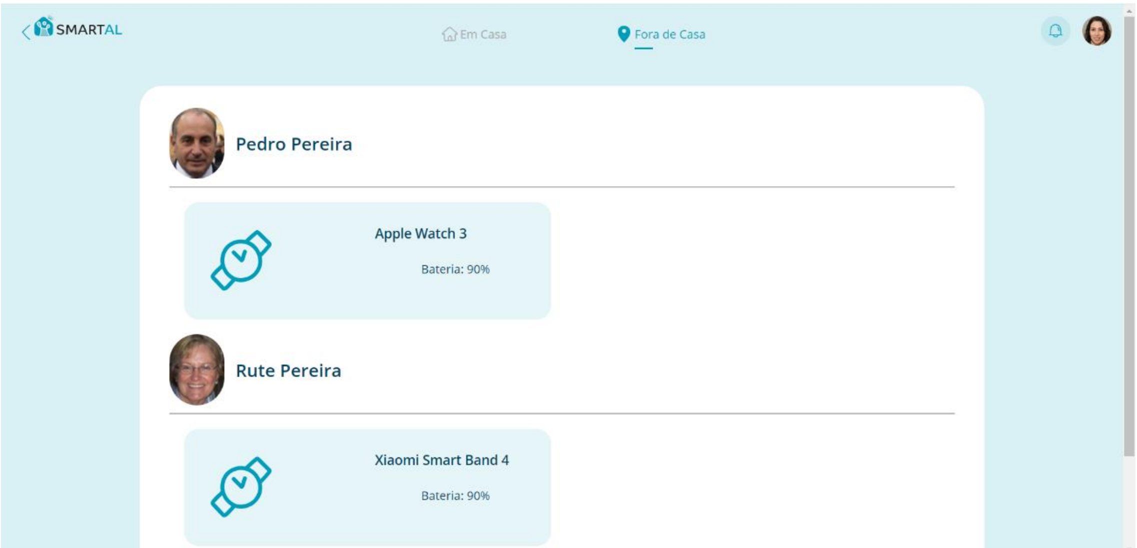


Figure 58: Outdoor Device Screen

In this screen (Figure 58), the caregiver can see the devices associated with which dependent and if they are charged. This allows the caregiver to contact the dependent reminding them to charge the devices needed.

Next, users might want to change data in their profiles. In that case, they can use the account pop (Figure 52) and by clicking on the first button *Gerir Conta* (Manage Account) the caregiver can change their account settings.

In here (Figure 59) the user can modify information such as name, phone number, and user image. The user can also change tabs.

Figure 59: Account Info Screen

In the next screen (Figure 60), the caregiver can change their password by providing their previous password and inserting the new one twice.

Figure 60: Password Change Screen

Finally, the user can access the notification popup (Figure 51) through the bell button in the top right corner of the screen. In case the caregiver needs to see older notifications, a button is available to allow access to the notification history screen (Figure 61).

Figure 61: Alert History

The caregiver can now scroll through the notifications as they see fit. The user can finally log out back into the login screen through the account popup (Figure 52).

6.6 Integration

System integration is a pivotal stage in the software development lifecycle, where individual pieces of the frontend and backend come together to form a cohesive and functional platform. This phase involves assembling various components, modules, and services, and interconnecting them to create a complete system. Before this stage, the frontend and backend sections are independently developed, each focusing on their specific functionalities and design aspects.

However, it is during the integration phase that these disparate parts are integrated and synchronized to deliver a unified software solution. This section describes the details of the system integration process by exploring the connections needed for the TeleAlert to function as intended. There are three main connections between the frontend and the backend: endpoints, real-time updates, and alerts.

6.6.1 Endpoints

The endpoints that compose the TeleAlert backend have already been covered earlier in the chapter, in subsection 6.4.6 Endpoints. However, the frontend cannot just call backend endpoints without receiving additional data from other microservices, for example, information about the users; the dependent's names and images are managed by the Entities microservice.

The Api-Gateway helps to forward the requests to the correct microservice (refer back to the Specification chapter, Architecture section for more information). This service receives URL requests formatted in the style **api/<service name>/<endpoint url>** and redirects them to the server that is provided in the link.

In this manner, the frontend can access any service it needs to get all the information it requires to function.

6.6.2 Real-time Updates

The procedure described in the previous subsection is used to get information when the web pages are first loaded. However, this information needs to be refreshed every time new data reaches the collectors.

Frontend updates should be automated, and mechanisms like polling[42] are not ideal due to its heavy load and stress on the Api-Gateway. Therefore, a more efficient and lightweight approach to handle these updates automatically is needed.

The Communication service was created to be responsible for real-time event handling and redirecting backend events to the correct client (refer back to section 5.6 Architecture for more information). This service collects messages from the RabbitMQ bus in specific routing keys. Taking into account that four TeleAlert screens need real-time updates, four routing keys in the exchange **home.exchange** were defined.

The communication code was altered by the SmartAL team to be able to receive those messages from the new routing keys and detect to which account they belong. The frontend was configured to support sockets to receive these messages. The next table presents a list of the names of the routing keys and which server sends information to that topic.

Table 16: Real-time Update Routing Keys

Routing key	Sender Service	Equivalent endpoint
home.devices	home	/device/account
home.location	indoor	/location/account
home.outdoor.devices	outdoor	/device/account
home.outdoor.location	outdoor	/location/account

As can be seen in Table 16, the information sent through these routing keys is the same as the one returned on the backend endpoints. To be able to visualize this information and not have the user wait for an update, the application still needs to call the endpoints available.

6.6.3 Alerts

The last form of connection between the frontend and the backend is through alerts. Similarly, to the Communication server, the Notification server also collects and distributes information from the Message bus (refer back to section 5.6 Architecture for more information). The main difference is that the Notification server sends messages to the Microfrontend-Common instead of just redirecting the received information it builds the notification and collects the additional data needed from the Entities server.

The alerts for TeleAlert are all generated by the Rule Server as a result of checking the preconfigured rules. However, the server does not generate the text of the alerts. It just constructs objects to send to the Notification server that uses them to generate a specific notification.

All the alert objects have a base structure composed of an account ID, a type, a severity, and a time stamp indicating when it was generated. Each type is sent to a different routing key in the **home.exchange** with additional info as is shown in the next table.

[42] check information status as part of a repeated cycle of requests.

Table 17: Alerts

Type	Description	Routing Key	Additional data
Walking	The dependent is walking differently than normal	home.notification.walking	Dependent id, walking manner, duration
Fall	The dependent fell indoors	home.notification.fall	Dependent id
Unexpected activity	There is indoor activity when it should not	home.notification.unexpectedActivity	Device name and type
Routine	The dependent is not following their usual routine	home.notification.routine	Device name and type, an activity that changed
Fall Outside	The dependent fell outside	home.notification.fallOutside	Dependent id
GPS connection	The device's GPS connection is down	home.notification.gpsConnections	Device name and type
Battery	The device's battery is low	home.notification.battery	Device name and type
Out of Fence	The dependent left a fence	home.notification.outOfFence	Dependent id
Removed Device	Dependent removed device	home.notification.removed	Dependent ID, device name and type
SOS	Dependent requested SOS	home.notification.sos	Dependent id

The Notification server then collects the information provided and turns it into structured notifications to be visualized in the micro frontend. In the future new notifications can be added for other rules created.

6.7 Summary

In summary, this chapter highlighted key aspects crucial to developing the TeleAlert solution. It covered deployment, version control, backend development, frontend development, and integration. Each element plays a vital role in ensuring a robust and efficient system.

The deployment process using Docker was essential for the seamless development of the microservices, while Git version control ensured code integrity and collaborative development. These wise choices had a positive impact on this project since many changes needed to be made by the SmartAL team to accommodate new requirements. Independent backend and frontend development phases were critical in delivering the expected functionality and user experience, respectively. Integration tied everything together, allowing different components and services to work together and communicate in different manners to reach the desired result.

This chapter's insights provide valuable guidance for future enhancements, contributing to a better understanding of the work developed during this dissertation.

7 Testing and Validation

The preceding chapter detailed the implementation process of the proposed TeleAlert system, elucidating the methodologies, and decisions undertaken to realize the theoretical concepts into a tangible reality. However, the success of a software solution hinges not only on its development but also on its ability to perform reliably and consistently in real-world scenarios. This chapter delves into the critical phase of testing and validating the solution.

7.1 Relevance

Testing and validation form the bedrock of software engineering, ensuring that the developed system meets the stipulated requirements and follows its intended purpose. It involves a structured approach to identifying defects, anomalies, and discrepancies within the software, while also verifying its alignment with the initial design specifications. Moreover, validation endeavors to ascertain that the system is aptly suited to fulfill its intended role within its operational environment.

Furthermore, validation extends beyond the realm of functionality to encompass the broader context of user satisfaction, system reliability, and adherence to industry standards. In the real world, it is important to perform a holistic evaluation of the system's viability and effectiveness in addressing the identified problem statement. The validation process not only corroborates the efficacy of the developed system but also validates the underlying assumptions and theoretical constructs that underpin its design.

As this is an academic project focused mainly on backend development, the full testing and validation process affirming the correctness, reliability, robustness, and effectiveness of the implemented system will not be performed. The present chapter will solely examine the alignment of the system with the projected objectives and will ascertain the validity of the TeleAlert system as a proof of concept.

7.2 Testing Process

As a first approach, manual testing was chosen to assess the TeleAlert system. It provided a practical and efficient means to evaluate the core functionalities of the system and gauge its performance under controlled scenarios. In this initial stage, no detailed functional test specification, covering the functional requirements was requested because the pre-existing microservices the system interacts with are still evolving to cover new requirements (e.g., internal fields of responses could change and invalidate specifications). But as the system evolved to a more integrated solution, end-to-end functional testing was also performed in more detail.

Note that the creation of a test catalog and automated testing is advisable in the future (if the system turns into a product) since only a full test case specification covering all the detailed functional aspects (including UX and UI testing) can ensure the overall quality of the service. Nevertheless, given the proof-of-concept nature of the TeleAlert system, the emphasis during this phase was on testing the fundamental features and ensuring they function as intended. As the system was developed, the manual testing process involved running several scenarios and carefully observing the system's behavior to ensure it worked correctly. While manual testing (without formal testing specification) introduces an element of subjectivity and potential human error, the team's direct involvement in system development mitigated these concerns. The iterative development process also allowed continuous adjustment and refinement of the system based on insights gained from manual testing.

During the development of the Collectors, the focus was mainly on testing message reception and storage, as well as ensuring the correct functioning of endpoints. Updates sent to the rule server and frontend were closely monitored. Manipulating the messages entering the collectors' queues facilitated the assessment of collector behavior across various scenarios. Deviations from the expected behavior were meticulously investigated, and code modifications were implemented to address those scenarios.

Concerning the Rule Engine service, testing revolved around interactions with the collectors (endpoints) and notification generation, but the primary focus was on the rules themselves. Complex rules underwent extensive testing in several scenarios to ensure they functioned as expected. The testing process included scenarios where rules were anticipated to trigger and others where they were not.

Regarding the frontend, testing was centered on verifying that information received from the backend services was accurately displayed. Endpoint calls were tested to ensure seamless integration with the frontend. Additionally, messages were sent to the collectors to confirm that the end-to-end system appropriately registered updates across all the application screens. To close the manual testing process, the proper generation of notifications was tested and their correct display within the system was validated.

7.2 Validation Process

Having realized manual testing, the focus now turns to system validation—a crucial phase that involves further assessment of the TeleAlert system's overall effectiveness and alignment with the end-user expectations. While manual testing mainly focused on functional evaluation, the validation process delved into asserting the system's performance within a specific real-world scenario. By subjecting the system to realistic use cases, the project demonstrates the status of the system as a valid proof of concept.

This section outlines the validation process applied to the TeleAlert system and provides comprehensive insights into the used scenarios. It will detail the real-world settings, methodologies, and results obtained during the validation process for each component. By dissecting the validation scenarios involving both the **Indoor** component (i.e., Indoor + Home service) and **Outdoor** component (i.e., Outdoor Service), a nuanced understanding of the TeleAlert system's applicability within different contexts will be elucidated.

7.2.1 Indoor Component

The Indoor Component of the system encompasses both the Indoor and Home services, along with the associated rules. The devices connected respectively to the SAFEHOME and SmartHome external systems were installed in the residence of a team member’s grandmother since the project’s main target audience is informal caregivers and their associated elderly dependents.

The objective of the initial validation phase was to verify if TeleAlert’s indoor component was collecting information as it should and passing that information through the preconfigured rules. It is important, however, to understand the operational constraints of the real scenario under analysis.

Table 18: Indoor Scenario Constraints

N	Constraint	Origin
1	Accounts in SmartAL and accounts created in external systems are not yet integrated/mapped. So, there is currently no way of mapping into a SmartAL account, residences/houses sent by the SAFEHOME and SmartHome systems to the TeleAlert service.	SmartAL, SmartHome, SAFEHOME
2	For the same reason, there is currently no way of mapping the dependents' data sent by the external systems to a specific SmartAL account.	SmartAL, SmartHome, SAFEHOME
3	The SAFEHOME team only had enough sensors to be installed in one room of the house.	SAFEHOME
4	The SAFEHOME team did not reach the objective of uniquely distinguishing steps from different people in the room. So, it will always send information to the same preconfigured user. (Id: 00000000-0000-0000-0000-000000000000).	SAFEHOME
5	The indoor sensors were in the house for a short trial period (approximately a month).	SAFEHOME

Table 18 summarizes the constraints of the indoor scenario used to validate TeleAlert. To overcome these limitations all the collected information was mapped into a single SmartAL user account that represented an informal caregiver. Additionally, the information sent by the SAFEHOME service was interpreted as being from the whole house, and as the dependent lived alone it was possible to ignore the problem of distinguishing between multiple dependents' steps. None of these simplifications cause incongruencies among services as the indoor location data is only used for display purposes and there is no additional usage that contradicts information received from the SmartHome devices. Furthermore, there are no rules that use information sent by both services at the same time, so triggers will not be affected by mismatched information (refer back to the Implementation chapter for more information on the rules implemented).

Given the presented constraints, the scenario will be further explained. The following image shows the house plant provided by the SAFEHOME team.

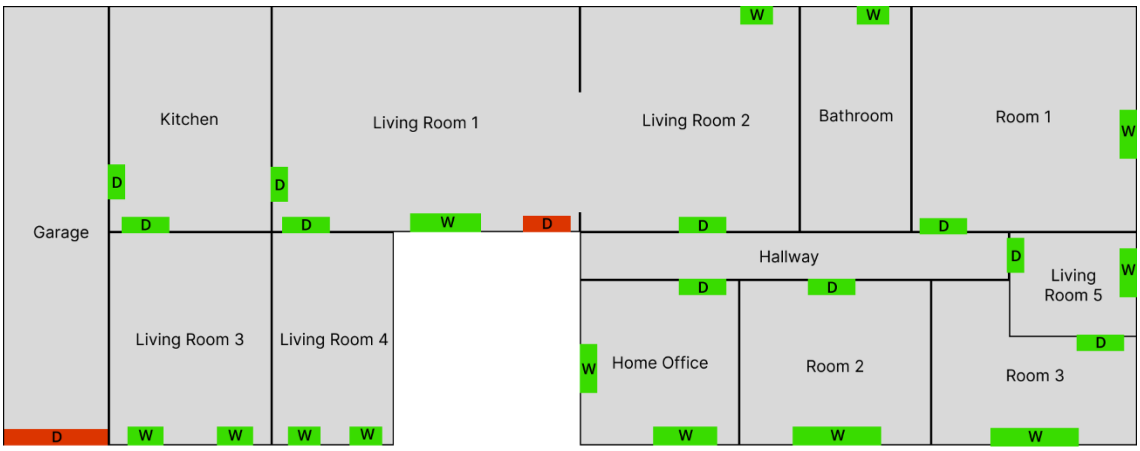
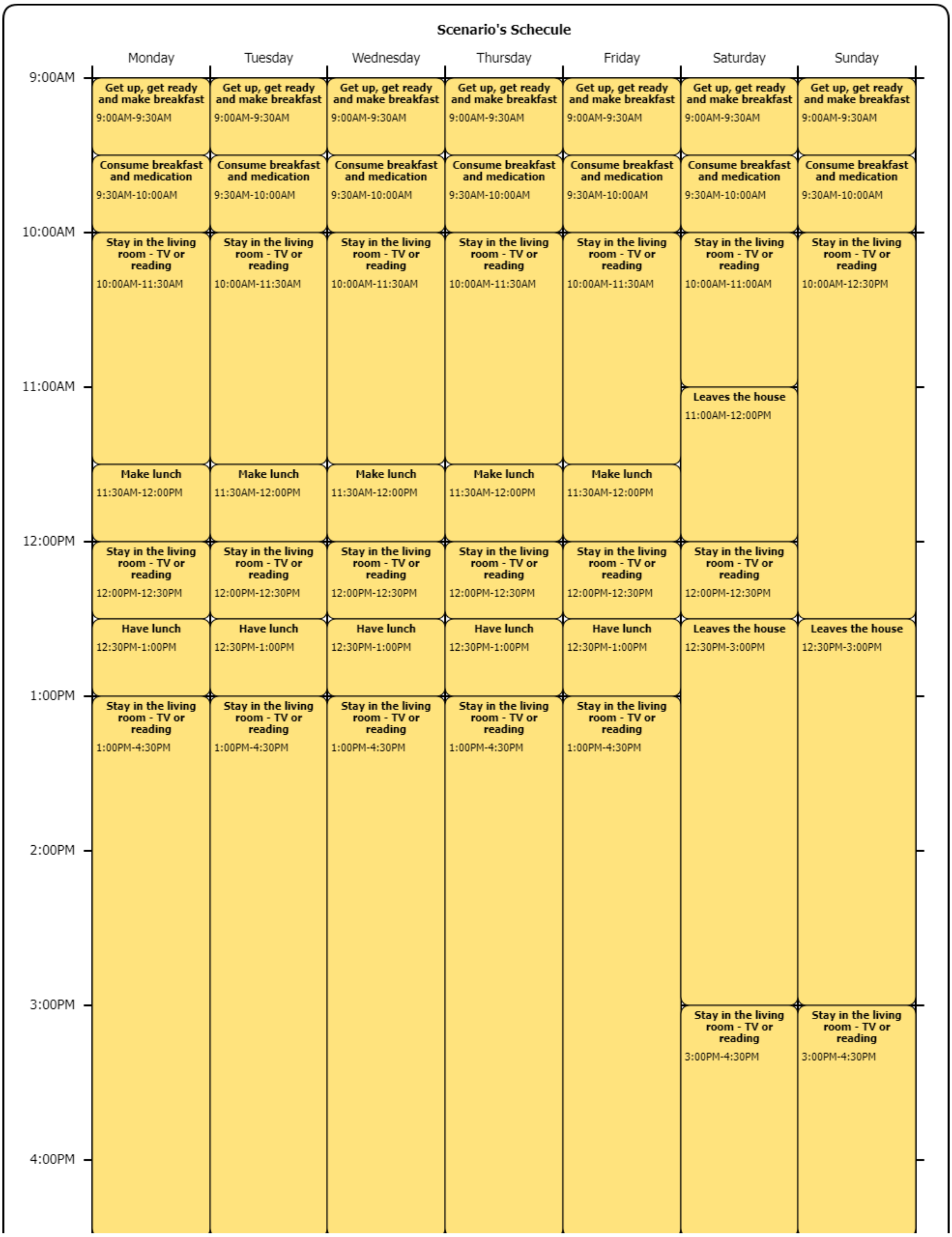


Figure 62: Scenario's House Plant

In addition, the following weekly schedule of the grandmother was provided:



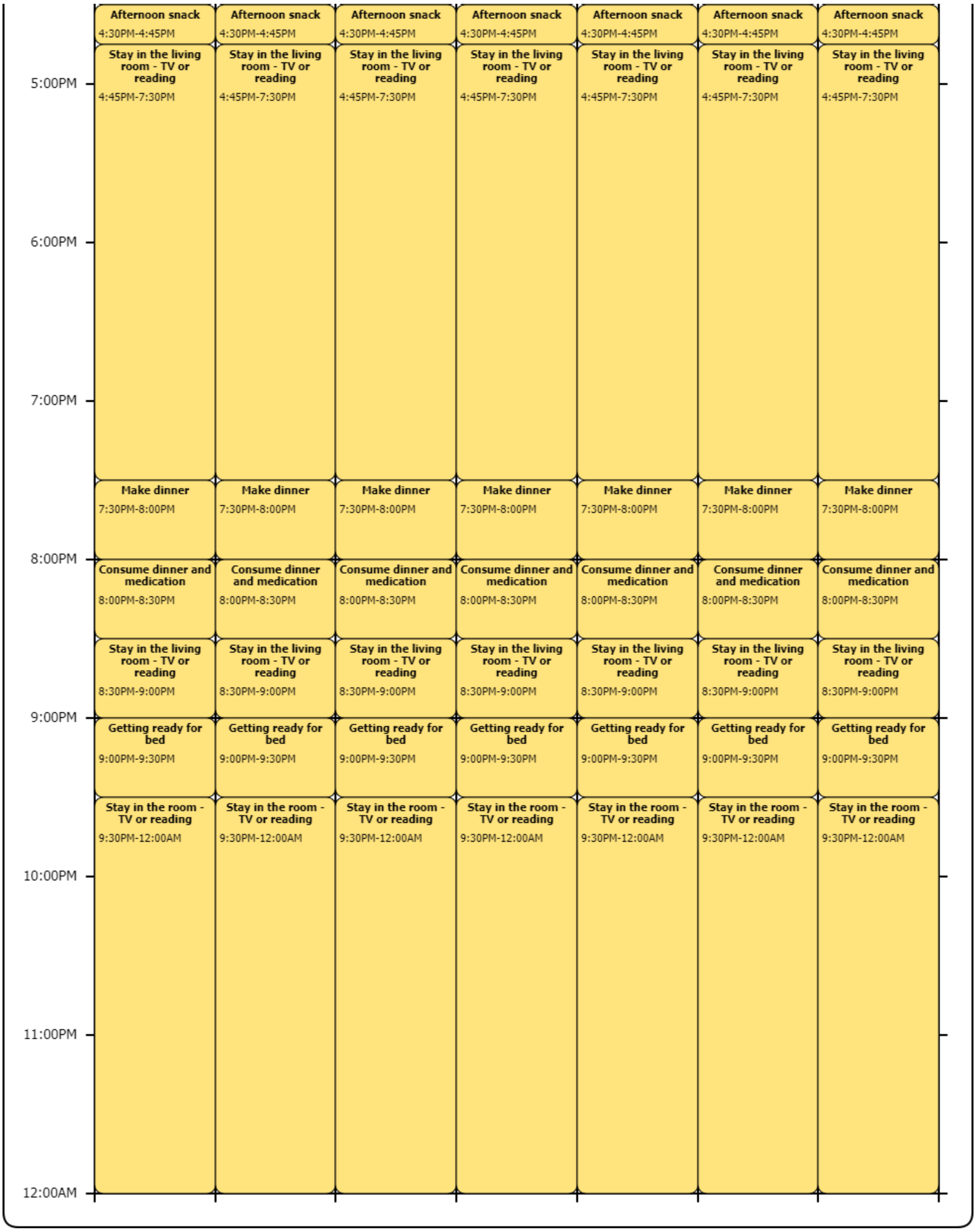


Figure 63: Dependent Schedule for the Scenario

Figure 63 details what is the expected weekly schedule of the dependent in care during the validation period. Along with the provided house plant, this helped decide where to place the indoor devices to better detect if there were deviations from the usual routine. Along with four indoor location sensors that were placed and tuned by the SAFEHOME team, SmartHome off-the-shelf smart devices were also distributed around the house: one home gateway, one light bulb, one plug, two contact sensors, and three motion sensors (as shown in the following picture).

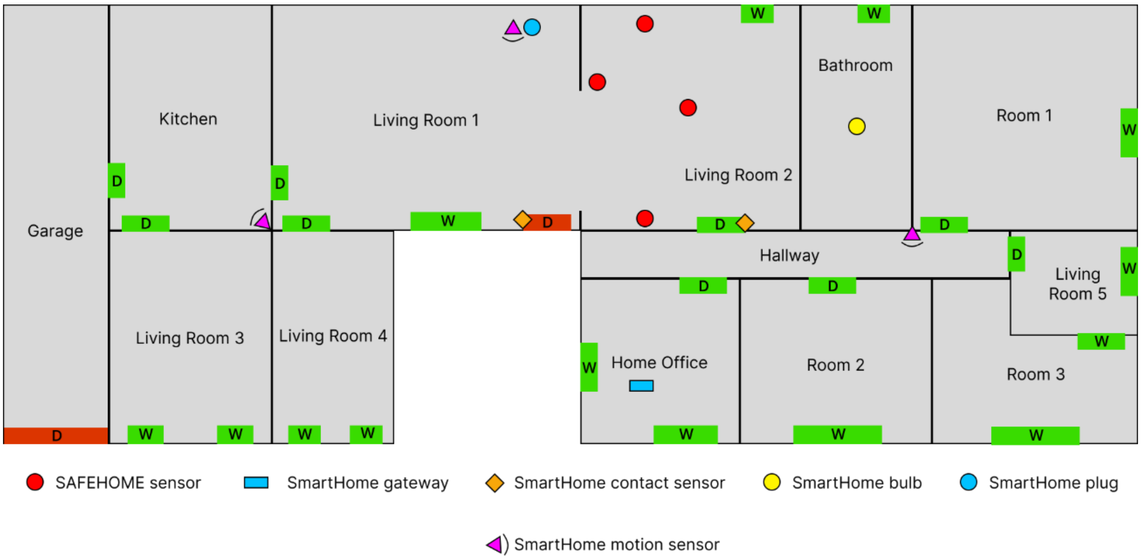


Figure 64: Indoor Device Positions

As can be seen in Figure 64, indoor sensors were placed by the SAFEHOME team in the living room 2. Currently, as already mentioned, indoor positioning is used for display purposes only. So, preconfigured rules are applied to falls, walking speed, and intensity of step.

Concerning the SmartHome devices, the gateway needed to be placed in a central room of the house - the home office – to collect the information from both the contact and motion sensors. The plug was connected to the TV to monitor usage and the light bulb was used to indirectly keep track of bathroom trips. Regarding the contact sensors, one was placed in the front door to detect if/when the dependent leaves the house and the other was placed in the hallway door to detect abnormal situations, since the grandmother informed that she was always closing that door before going to sleep, opening it back when she goes for breakfast in the morning. Finally, the motion sensors were placed in different rooms to monitor specific situations: in the kitchen to survey meals and medication consumption, in the main living room where the dependent usually reads and watches TV, and in the hallway to check for any bedroom leaves as the dependent sleeps in room 1 and wants to keep her privacy in the room.

The rules defined for data coming from the Home collector were adjusted to fit the placement of the devices and the dependent's schedule. Rules for routine checking and unexpected activity were adapted to fit the real scenario. In summary, two types of rules need to be adapted: the routine rules attest to the lack of activity and the unexpected activity rules attest to the presence of activity. The other types of rules are fixed and are valid for any scenario. The conditions the rules need to verify are detailed in the next table.

Table 19: Routine Conditions for Rules

Type	Device	Start Time	End Time	Days of the Week	Severity
Unexpected	Hallway Motion Sensor	00h00	09h00	All	Medium
Routine	Hallway Motion Sensor	09h00	09h45	All	High
Routine	Hallway Contact Sensor	09h30	10h10	All	High
Routine	Kitchen Motion Sensor	09h30	10h30	All	High
Routine	Living Room TV Plug	10h00	11h30	All	Low

Routine	Living Room Motion Sensor	10h00	11h30	All	Low
Routine	Kitchen Motion Sensor	11h30	12h00	All	Medium
Routine	Living Room TV Plug	12h00	12h30	All	Low
Routine	Living Room Motion Sensor	12h00	12h30	All	Low
Routine	Kitchen Motion Sensor	12h30	13h00	All	Medium
Routine	Living Room TV Plug	13h00	16h20	Weekdays	Low
Routine	Living Room Motion Sensor	13h00	16h20	All	Low
Routine	Kitchen Motion Sensor	16h20	17h00	Weekdays	Low
Routine	Living Room TV Plug	16h45	19h30	Weekdays	Low
Routine	Living Room Motion Sensor	16h45	19h30	All	Low
Routine	Kitchen Motion Sensor	19h30	20h10	All	Medium
Routine	Kitchen Motion Sensor	20h00	20h40	All	High
Routine	Living Room TV Plug	20h40	21h30	All	Low
Routine	Living Room Motion Sensor	20h40	21h30	All	Low
Routine	Bathroom Bulb	21h00	21h30	All	Low
Routine	Bathroom Bulb	21h20	21h40	All	High
Routine	Hallway Contact Sensor	21h00	23h00	All	High
Routine	Hallway Motion Sensor	21h30	22h00	All	Medium
Routine	Entrance Contact Sensor	11h00	11h20	Saturday	Medium
Unexpected	Any Sensor	11h20	11h50	Saturday	Medium
Routine	Living Room TV Plug	12h00	12h30	Saturday	Low
Routine	Living Room Motion Sensor	12h00	12h30	Saturday	Low
Routine	Entrance Contact Sensor	12h40	13h10	Saturday	Medium
Unexpected	Any Sensor	12h50	14h30	Saturday	Medium
Routine	Living Room TV Plug	15h00	16h30	Saturday	Low
Routine	Living Room Motion Sensor	15h00	16h30	Saturday	Low
Routine	Living Room TV Plug	10h00	12h30	Sunday	Low
Routine	Living Room Motion Sensor	10h00	12h30	Sunday	Low
Routine	Entrance Contact Sensor	12h40	13h10	Sunday	Low
Unexpected	Any Sensor	12h50	14h30	Sunday	Medium
Routine	Living Room TV Plug	15h00	16h30	Sunday	Low
Routine	Living Room Motion Sensor	15h00	16h30	Sunday	Low

In Table 19, it is represented the dependent's routine through conditions without having an exponential number of rules. More rules and devices could be added but these seem to be enough to cover the dependent's routine. This is a personalized routine table and, as such, it is not reusable for other users and therefore not scalable. To overcome this problem, a configuration interface can be provided in the future, with typical conditions where time can be edited and changed according to the user. Also, the option of creating new conditions can be added. Furthermore, this is a typical case study for AI/ML algorithms that can learn and adjust the time according to the needs of each user.

After mounting the trial scenario and collecting all the data, the next stage is to make a comprehensive analysis of the results obtained.

Regarding the collectors involved in this validation phase - Home and Indoor - they behaved as expected collecting two weeks' worth of data related to both indoor position and indoor device status respectively. The whole history of the user and related devices was registered as expected.

The Indoor collector's position values consistently adhered to expectations, falling within the designated 4x3 meters area. However, an anomaly arose with the recorded walking speed values. Due to the SAFEHOME system's incorporation of walking direction, negative values were transmitted, and this affected the rules regarding walking speed. Intensity values, encompassing a range from zero to fifteen, aligned with predetermined expectations. However, a noteworthy issue emerged regarding fall detection. Instances of false positives occurred, stemming from the SAFEHOME's AI model that was not trained in the dependent's house as it should have been. Consequently, the system registered falls when none were reported by the dependent as the intensities registered in the house were higher than in the place where the model was trained.

This particular collector holds responsibility for rules governing falls and the dependent's walking behavior. Consequently, any inaccuracies in the data unavoidably impact the notifications generated. During the validation scenario, notifications linked to falls emerged as the most prevalent. While no actual falls transpired, the system efficiently generated notifications in response to fall-indicative updates. Therefore, it can be affirmed that if no falls are communicated, no notifications are triggered, showcasing TeleAlert's expected behavior in this aspect. Nevertheless, to address the problem created by the indoor system, TeleAlert now incorporates a validation step to detect erroneous fall triggers - updates indicating falls with an intensity below 0.2 (the highest value to be considered a step) will be stored without classifying them as falls.

In addition, adjustments were necessary in the walking behavior rules. The presence of negative values in the velocity field notably influenced the calculation of the baseline velocity. To address this, the solution was to retain the direction for future rules, while altering the baseline calculation to utilize the absolute value of speed. Also, recalibrations were required for the intensity and velocity baseline calculation ranges. Adaptations were introduced to account for speed values ranging from zero to five meters per second, and intensity values spanning from 0 to 0.2. Notably, rule modifications were implemented to accommodate smaller differences from the baseline intensity, as the shifts were so slight that they evaded rule detection.

Looking at the collector's functionality it worked as expected. Along with the devices from the scenario, other device updates being tested by the SmartHome team were detected by the system, and their whole history was registered.

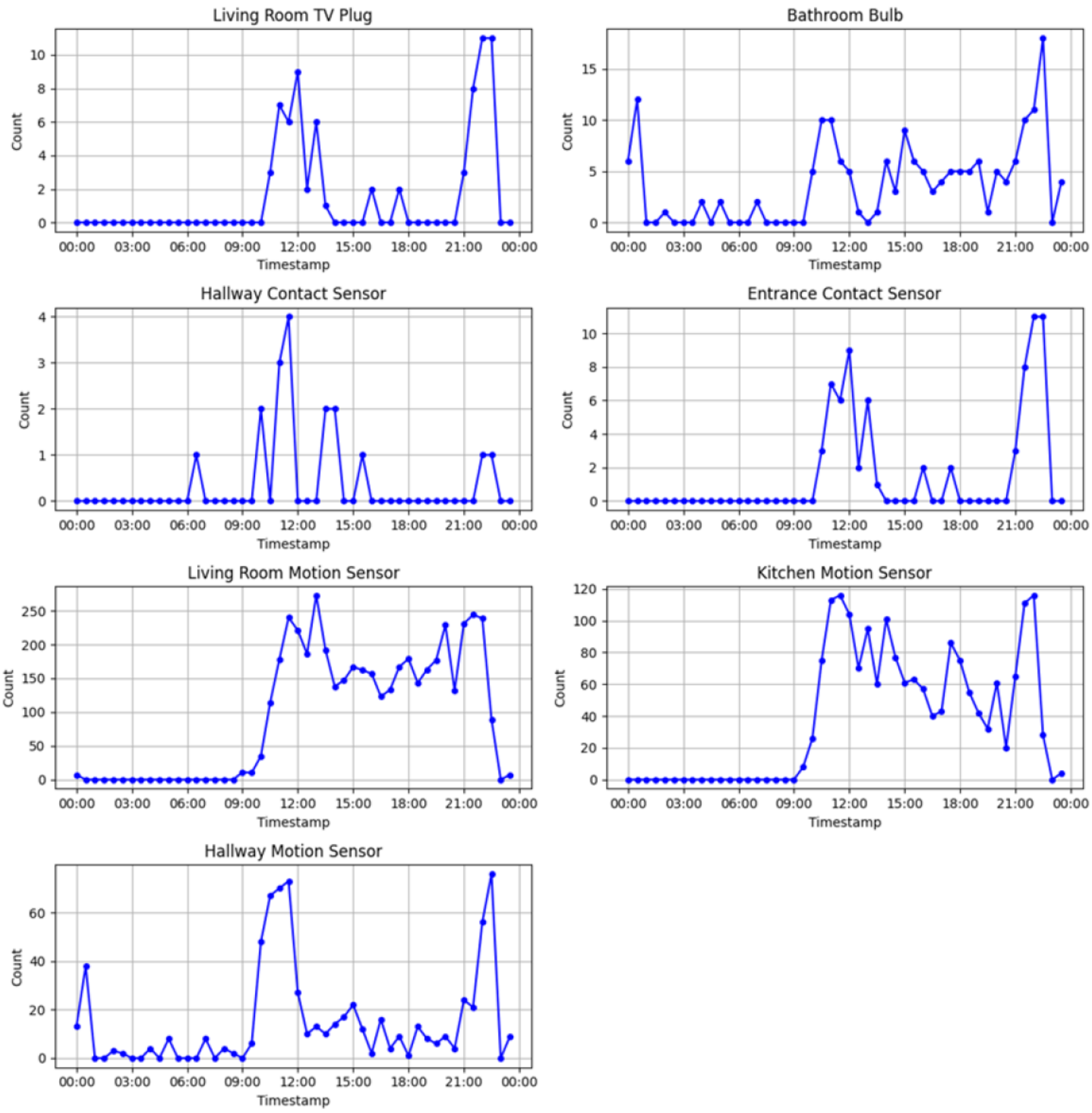


Figure 65: Home Device Update Graphs

Figure 65 shows the number of updates sent by each device during the whole month in 30-minute intervals during the day.

Having a closer look at the results, the living room plug placed on the TV was rarely turned off during the day, so the related routine rules were triggered many times. This probably happened because of the way the plug was set too close to the floor and harder for the dependent to reach. This means that the plug positioning should be rethought in future scenarios.

Regarding the contact sensors, the entrance sensor behaved as expected. While not having as many updates as the other sensors because the dependent only leaves during the weekend, it proved valuable in showing that the dependent mostly leaves the house on those days after 11 a.m. and usually returns before 10 p.m. The hallway sensor data, however, was not as useful. Only after the scenario ended, did the dependent inform that the sensor fell in the first few days. The rules related to it would also turn out to be irrelevant since there seemed to be no pattern in the times the dependent opened the hallway door, so they would trigger very frequently.

The bulb placed in the bathroom was used throughout the day but mostly at night around the hours the dependent was getting ready for bed. This meant that the rules proved useful to detect if the dependent was late to perform that task. Additionally, rules for unexpected activity would be activated if needed. The data collected showed that the dependent often went to the bathroom during the night. If this situation becomes too frequent the caregiver can contact the dependent to check if everything is ok.

Lastly, the motion sensors proved to be the most useful to monitor the routine. While the other devices need a specific action to be performed by the dependent like opening a door or turning on a light, the motion sensors only need the dependent to enter to rooms. The living room sensor proved more useful than the TV plug. It showed constant activity during the day and was not restricted to detecting TV usage, proving that the dependent stayed in the living room most of her time even when she was not watching TV. The hallway sensor proved very useful as well. It mostly activated when the dependent woke up and went to bed proving very valuable to monitor waking up and going to bedtimes. The kitchen sensor seemed to only activate a bit later than expected and that caused the routine rules to activate if the medicine was taken later in the morning. It also showed that the dependent visited the kitchen very frequently during the day, so the rules were not triggered, describing accurately the several daily kitchen visits for meals.

Overall, the rules seem to describe the routine closely to reality. Ignoring the hallway contact sensor and the plug, all the other sensors activated rules as expected. Nevertheless, the rules should be adjusted to be a bit more flexible time-wise allowing small variations in the routine to not trigger them so often.

7.2.2 Outdoor Component

The outdoor component only contains the OGTW that feeds the outdoor collector. This service was ready to be used during the trial, but the smart device was not. The chosen watch is a new acquisition by Altice Labs and is still being tuned along with the provider. So, it was not possible to collect real-time data from the dependent of the previous scenario, but it was still tested later. During some weeks, I wore the smartwatch to make sure the outdoor collector worked as expected. Since this collector did not attest to any routine-related rules but only to situations surrounding outdoor activity, this fallback solution was good enough for rule validation.

The outdoor collector behaved as expected, and every update that came from the watch was registered to the database. In addition to the device I was using, it was also possible to collect data from another device used by a team member and register its updates. However, the outdoor system that fed the collector had some problems receiving the coordinates of the watch; it always sent the default coordinates (latitude and longitude 0). Nevertheless, this did not affect the validation of the collector since it must register coordinates independently of its values. Furthermore, the coordinates are only used for display purposes, so not for rule validation.

Another unexpected situation happened with the device removal. Even though the device was removed multiple times during the validation period none of these situations were detected by the watch and therefore, no notification was ever generated for that scenario. Other situations were also not possible to validate: a) out-of-fence scenarios could not be tested as the outdoor system that fed the collector still does not support that function; b) the GPS signal was also never lost and as such, no notification was ever generated; c) it was not possible to generate an SOS alarm as the team is not receiving them from the watch yet. For these situations, manual testing results had to suffice.

What was operational to test, were the status changes such as, low battery and fall detection. Therefore, they were all registered in the collector and the rules that evaluated these situations were able to generate the corresponding alerts.

Please, note that the outdoor component was considered from the beginning as “nice-to-have” due to the time frame of the dissertation. Nevertheless, it was fully implemented and manually tested and validated. So, it is expected to work properly as soon as the communication problems of the watch are solved.

7.3 Summary

In this chapter, the TeleAlert system underwent manual testing and validation, cementing its status as a proof of concept aligning with the project's objectives. Through manual testing, the core components were evaluated, uncovering bugs, and facilitating ongoing improvements. The subsequent validation phase focused on a real-world scenario within the confines of a single home, offering insights into system functionality under genuine conditions.

The validation process exposed both successes and areas for refinement. Notably, the indoor component of the system showcased expected behavior regarding position values and notifications for falls. However, challenges arose in walking speed measurements, leading to adjustments in baseline calculations and validation criteria.

While certain aspects required adaptation, the results proved TeleAlert's capability to operate as a proof of concept. The validation process not only confirmed its functionality but also provided valuable insights for refining the system's performance and enhancing its real-world applicability.

As the system evolves, integration of automated testing and other methodologies promises to bolster comprehensive coverage, repeatability, and scalability, serving as a bridge to future phases of development and deployment. This chapter thus stands as a pivotal juncture, where the TeleAlert system's journey from implementation to validation underscores its potential to address real-world needs effectively.

8 Conclusion

This concluding chapter undertakes the pivotal task of synthesizing the multifaceted components explored throughout the work. This closing chapter will succinctly include the final results from the project, shedding light on the successful development and implementation of the telelocation and alert transversal software system, aptly named TeleAlert.

This system represents a big step in the right direction of a functional non-clinical telemonitoring system, enabling efficient telelocation and timely alerts with various external systems. Moreover, it will address the challenges and difficulties encountered during the research and development process, offering insights into the technical and logistical considerations that were navigated to bring TeleAlert to fruition.

Delving into the realm of prospective scholarly endeavors, this last chapter will outline potential avenues for future research, extending the trajectory of the current study's insights and inviting further exploration of the software's capabilities and potential refinements. Ultimately, the conclusion will draw together these threads to offer a reflective assessment, emphasizing the significance of TeleAlert's findings and contributions within the academic and practical landscape of modern technological advancements.

8.1 Final Status

To determine the overall success of the project, it is crucial to refer to the defined threshold of success outlined in Chapter 2 Work Plan. Upon reflection by the team, it can be concluded that the project met the threshold of success for the following reasons:

- The system was developed according to the must-have and should-have requirements.
- The system respected all non-functional requirements.
- The system was tested through manual testing. The developed application was successfully deployed in the cloud to be validated. The indoor component was validated in a realistic scenario of use. The outdoor component was still validated but in a less focused scenario.
- The results from the scenarios were explained and analyzed.

The internship culminated in successfully creating a proof of concept that demonstrates its practical viability. This achievement involved the development of data collectors and a rule server, which transmitted information to the frontend for visualization, thereby assisting caregivers in making informed decisions. The application not only meets all the must-have stipulated requirements but also incorporates additional features categorized as should have.

While the developed application effectively addresses the immediate project goals and objectives, it is acknowledged that more extensive testing could have further fortified its credibility. Furthermore, numerous opportunities for future improvements and refinements were identified. These opportunities encompass the exploration of novel functionalities, the enhancement of the user interface, and the introduction of a more comprehensive notification service.

The internship has successfully laid the foundation for subsequent development phases, and the pinpointed areas for refinement. It presents a clearly defined trajectory for augmenting the application's capabilities and elevating the overall user experience.

8.2 Problems

Every project has difficulties and challenges to overcome to reach the final results. This project was no different. This section will cover the main difficulties the dissertation faced:

- Initial Scope: It took some time to understand the scope of the work being proposed leading to some misguided research and development.
- Software and Hardware: Considering the microservice nature of SmartAL, when it came to Docker local deployment on the computer provided by the organization, it did not have the capabilities to run all the services needed. Even after adding RAM extension the computer was slow to build and deploy containers.
- Administration privileges: It was requested from the beginning that the provided account beginning should have admin privileges but there was a delay. When the permanent privileges were finally conceded, everything was already installed using temporary privileges. This caused additional delays in installations and configurations.
- Communication: Even though communication with the team was frequent, not all members had a full understanding of the project since each one of them focused on developing one or two microservices according to their role on the team and their technological knowledge. Additionally, communication with the other teams from the external systems that sent information to the collectors caused delays in development. This was overcome on a case-by-case basis and gave an idea of how a real organization works. Overall project coordination is very important for development.
- Deployment: Many existing services were needed for TeleAlert to function as intended and the configuration on the local machine was a difficult and slow process. The complexity of the service structure was initially hard to understand and caused problems setting up the services TeleAlert needed to communicate with.

8.3 Future Work

TeleAlert stands as a work in progress, with significant strides made, but the ultimate goal of achieving the desired final product still lies ahead. This section summarizes the forthcoming updates and improvements that can propel TeleAlert closer to its envisioned state. By addressing the current limitations and exploring innovative solutions this dissertation aims to propose potential trajectories that can guide TeleAlert's evolution and maturation into a more robust and effective system.

First, a solution needs to be found for mapping accounts from different external systems into the SmartAL accounts, and therefore mapping the collected information to the respective account. In this PoC, information was mapped to the same example account provided by the SmartAL team. Forms can be developed as a way of adding dependents, devices, and homes to a specific caregiver account. Furthermore, the frontend currently only supports one house at a time for indoor information, so a restructuring should be made to accept houses of multiple dependents and multiple houses for the same dependent. Also, the current indoor map solution does not account for user errors. New solutions for the indoor map should be explored to make the experience more reliable and user-friendly.

Next, some rules are specific to a given scenario and there is currently no way of changing these rules. A component that supports rule adjustment should be created to allow the caregivers to set their own rules while keeping them saved and associated specifically with their accounts. New rule types can also be created to use more data received and other rules interlacing information from more than one collector. Machine Learning algorithms could also be researched and added to the rule engine server to allow for automatic rule adjustment. To ease the process, some user feedback mechanisms can be incorporated in the web app allowing the caregiver to provide some kind of rating to the generated alerts. The Machine Learning algorithm should then be able to use that information to adjust the conditions that generate the alerts.

Currently, only the alert history is being used in the interface. Devices and location history can also be added to enrich the UI. Regarding the location, a path can be drawn on the map to show the last user locations. The device cards can also become clickable to open a popup that shows past device information.

Optimization of code should also be considered to ensure the performance and scalability of software applications. Well-optimized code can significantly enhance the efficiency and responsiveness of an application, allowing it to handle higher workloads and scale more effectively.

There is still a long way to go until TeleAlert is considered a viable product for the market. Further testing and validation are necessary for this system. Manual testing is very limited and biased as it is usually performed by the developer. Automated endpoint testing should be considered to achieve bigger code coverage. Frontend automated testing should also be considered to test edge cases that were not covered by manual testing. Furthermore, other scenarios should be found to validate the system before releasing it. This will help further enhancements and adjustments to real users' needs.

Also, usability testing should be considered as it evaluates the effectiveness, efficiency, and overall user-friendliness of TeleAlert. Its main goal is to identify problems the user might face when using the application and gather feedback to improve design and user experience.

Finally, performance and scalability testing should also be performed to ensure the system functions properly under a bigger volume of requests and heavier loads. Reliability and resilience are requirements this kind of critical system needs. These attributes should also be tested to make sure the system does not fail unexpectedly. Mechanisms of fault tolerance and detection should be explored.

8.4 Final Thoughts

In the face of its challenges, the internship has emerged as a success, affirming the viability of the TeleAlert system as proof of concept. Moreover, the system is now well-positioned for further research and the impending future tasks that lie ahead. From a technical standpoint, the internship has proven to be an invaluable source of knowledge.

My understanding of software development has undergone a significant deepening, underscoring the paramount importance of conceptual service and architecture design. Additionally, a long-standing aspiration of mine has been realized through achieving proficiency in React, thereby expanding my skill set in front-end development.

Despite working independently on the project, being integrated into an enterprise team has provided me with valuable insights into the mechanics of collaborative and effective communication team skills. The feedback I have received, both from team members and external parties, has played an important role in my professional growth. This constructive feedback has not only honed my technical expertise but has also contributed substantially to my personal development.

Taken as a whole, this internship has been a remarkable journey of learning, progress, and noteworthy accomplishments. It has provided me with hands-on experience within a professional context, fostering a deeper understanding of teamwork, technical execution, and the iterative nature of software development. I hold deep gratitude for this opportunity and eagerly look forward to applying the knowledge and skills I have acquired during this internship to future endeavors.

References

[1] G. Eysenbach, "What is e-health?," 18 06 2001. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1761894/. [Accessed 01 12 2022].

[2] EUROPEAN COMMISSION, "Market study on telemedicine," EUROPEAN COMMISSION, Brussels, 2018.

[3] S. Frade and H. Rodrigues, "Benefits, Challenges and Impact of Teleconsultation –A Literature Review," MEDINFO, Porto, 2013.

[4] N. Persaud, G. Moffitt, K. Kravette and A. DeRoss, "Usability and Benefits of a Telehealth Pilot for Older Adults: Via a Service-Learning Technology Partnership," Communications of the IIMA, New York, 2013.

[5] L. T. Machadeiro, "Dispositivos e sistemas de Telemonitorização: Panorama atual e tendências futuras," Universidade da Beira Interior, Covilhã, Portugal, 2017.

[6] United Nations Department of Economic and Social Affairs, "World Population Prospects," United Nations Department of Economic and Social Affairs, New York, 2022.

[7] M. Roser, H. Ritchie and E. Ortiz-Ospina, "Life Expectancy," Our World in Data, 10 2019. [Online]. Available: https://ourworldindata.org/life-expectancy. [Accessed 23 11 2022].

[8] G. C. Brown, "Living Too Long," EMBO Reports, 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4328740/. [Accessed 03 11 2022].

[9] Z. Li, Z. Zhang, Y. Ren, Y. Wang, J. Fang, H. Yue, S. Ma and F. Guan, "Aging and age-related diseases: from mechanisms to therapeutic strategies," Springer Nature B.V., Berlim, 2021.

[10] M. J. Divo, C. H. Martinez and D. M. Mannino, "Ageing and the epidemiology of multimorbidity," European Respiratory Journal, 30 09 2014. [Online]. Available: https://erj.ersjournals.com/content/44/4/1055. [Accessed 03 11 2022].

[11] K. H. Howell, J. K. Coffey, G. M. Fosco, K. Kracke and K. S. Nelson, "e-Publications@Marquette," 01 01 2016. [Online]. Available: https://epublications.marquette.edu/cgi/viewcontent.cgi?article=1268&context=psych_fac. [Accessed 04 11 2022].

[12] altice labs, "eHealth Telemedicine Teleconsultation & Telediagnosis System – Medigraf," altice labs, [Online]. Available: https://www.alticelabs.com/products/ehealth-telemedicine/. [Accessed 01 12 2022].

[13] altice labs, "eHealth Smart Assisted Living ICT platform for telemonitoring and teleconsultation – SmartAL," altice labs, [Online]. Available: https://www.alticelabs.com/products/ehealth-smart-assisted-living/. [Accessed 01 12 2022].

[14] M. Cary and A. Darkins, Telemedicine and Telehealth: Principles, Policies, Performances and Pitfalls, 2000.

[15] H. Shirzadfar and F. Lotfi, "The evolution and transformation of telemedicine," International Journal of Biosensors & Bioelectronics, Baharestan, 2017.

[16] C. "What are B2B and B2C? Definition, examples, differences," moovago, 20 01 2022. [Online]. Available: https://moovago.com/en/blog-en/prospecting/b2b-and-b2c/. [Accessed 17 11 2022].

[17] "What is B2B2C?," LifeRay, [Online]. Available: https://www.liferay.com/blog/customer-experience/what-is-b2b2c. [Accessed 17 11 2022].

[18] "B2B vs B2C Sales: Four Key Differences and Their Importance," Configure One, [Online]. Available: https://www.configureone.com/company-news/b2b-vs-b2c-sales/. [Accessed 17 11 2022].

[19] M. Waida, "B2B vs. B2C Marketing: What is the Difference?," Wrike, 19 08 2021. [Online]. Available: https://www.wrike.com/blog/b2b-vs-b2c-marketing-differences/#What-is-the-difference-between-B-B-and-B-C-marketing. [Accessed 17 11 2022].

[20] B. Dean, "Uber Statistics 2022: How Many People Ride with Uber?," backlinko, 23 03 2021. [Online]. Available: https://backlinko.com/uber-users. [Accessed 17 11 2022].

[21] Hakee, "How do On-Demand apps shape our lifestyle," haktos, 09 11 2020. [Online]. Available: https://medium.com/haktos/how-do-on-demand-apps-shape-our-lifestyle-7ed5ee384fd4. [Accessed 18 11 2022].

[22] A. Džakula, D. Relić and P. Michelutti, "Health workforce shortage – doing the right things or doing things right?," Croatian Medical Journal, 2022.

[23] Mobifilia, "Uberization of Healthcare," Mobifilia, 26 09 2019. [Online]. Available: https://www.mobifilia.com/healthcare-uberization/. [Accessed 18 11 2022].

[24] N. Merchant, "IoT Technologies Explained: History, Examples, Risks & Future," Vision of Humanity, [Online]. Available: https://www.visionofhumanity.org/what-is-the-internet-of-things/. [Accessed 19 11 2022].

[25] G. J. Carnaz and V. Nogueira, "An Overview of IoT and Healthcare," Repositório Ciêntifico da Universide de Évora, Évora, 2017.

[26] Digiteum, "Benefits of Using Internet of Things (IoT) in the Healthcare Industry," Digiteum, 26 06 2020. [Online]. Available: https://www.digiteum.com/iot-benefits-healthcare-industry/. [Accessed 23 07 2023].

[27] M. Thomas, "IoT in Healthcare: 15 Examples of Internet of Things Healthcare Devices and Technology," built in, 07 11 2022. [Online]. Available: https://builtin.com/internet-things/iot-in-healthcare. [Accessed 19 11 2022].

[28] ordr, "10 INTERNET OF THINGS (IOT) HEALTHCARE EXAMPLES," ordr,

[Online]. Available: https://ordr.net/article/iot-healthcare-examples/. [Accessed 19 11 2022].

[29] Y. Arora, "GPS & ITS APPLICATIONS," International Journal of Latest Trends in Engineering and Technology, 2017.

[30] C. Hoffman, "How GPS Actually Works," How-TO Geek, 23 09 2016. [Online]. Available: https://www.howtogeek.com/137862/htg-explains-how-gps-actually-works/. [Accessed 25 07 2023].

[31] GPS.GOV, "Surveying & Mapping," GPS.GOV, 10 02 2022. [Online]. Available: https://www.gps.gov/applications/survey/. [Accessed 25 07 2023].

[32] CBO, "The Global Positioning System for Military Users: Current Modernization Plans and Alternatives," CBO, 2011.

[33] S. N. Khan, "Integration of GIS with Remote Sensing and GPS for Disaster Mitigation," University of Gujrat, Gujrat, 2015.

[34] S. Paiva and C. Abreu, "Low Cost GPS Tracking for the Elderly and Alzheimer Patients," Viana do Castelo Polytechnic Institute, Viana do Castelo, 2012.

[35] V. N. V. V. P. Ranjith H D, "Ambulance Guidance System for Medical Emergency," Global Scientific Journal, Mangalore, 2021.

[36] M. Chelly and N. Samama, "New techniques for indoor positioning, combining deterministic and estimation methods," in *European Navigation Conference - Global Navigation Satellite Systems*, Naples, 2009.

[37] MOKO Smart, "What is IPS? – An Introduction to Indoor Positioning System," MOKO Smart, 26 05 2022. [Online]. Available: https://www.mokosmart.com/introduction-to-indoor-positioning-system/. [Accessed 21 11 2022].

[38] I. Perišić, I. Kekeljević and D. Venus, "Model Driven Dynamic Resources Finder (MDD_RF)," University of Novi Sad, Faculty of Technical Science, Novi Sad, 2014.

[39] G. E. Violletas, T. L. Theodorou and C. K. Georgiadis, "NetArgus: An SNMP Monitor & Wi-Fi Positioning, 3-tier Application Suite," in *Fifth International Conference on Wireless and Mobile Communications*, Cannes/La Bocca, 2009.

[40] A. W. Reza and T. K. Geok , "Apple Inc. (AAPL) iBeacon with Micromapping can revolutionize retail [REPORT]," SpringerLink, 2008.

[41] Z. Jie, L. HongLi and Tanjian, "Research on ranging accuracy based on RSSI of wireless sensor network," IEEE, 2010.

[42] F. Alkhawaja, M. A. Jaradat and L. Romdhane, "Techniques of Indoor Positioning Systems (IPS): A Survey," in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, 2019.

[43] M. Y. Umair, K. V. Ramana and Y. Dongkai, "An enhanced K-Nearest Neighbor algorithm for indoor positioning systems in a WLAN," IEEE, Beijing, 2014.

[44] R. Jérémi, B. Karell and F. Cyril, "Ble Based Indoor Positioning System and Minimal Zone Searching Algorithm (MZS) Applied to Visitor Trajectories within a Museum," MDPI, 31 05 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/13/6107/htm. [Accessed 22 11 2022].

[45] H. Nurminen, A. Ristimaki, S. Ali-Löytty and R. Piché, "https://www.researchgate.net/publication/303012657_Particle_filter_and_smoother_for_indoor_localization," Research Gate, 2013.

[46] L. T. Machadeiro, P. D. Gaspar and M. Castelo-Branco, "Telemonitoring Devices and Systems: Current Status and Future Trends," University of Beira Interior, Covilhã, 2017.

[47] The Victorian Quality Council, "A guide to using data for health care quality improvement," The Victorian Quality Council, 2008.

[48] S. Meystre, "The Current State of Telemonitoring: A Comment on the Literature," Mary Ann Liebert, Inc., New York, 2005.

[49] M. J. Rodrigues, O. Postolache and F. Cercas, "Physiological and Behavior Monitoring Systems for Smart Healthcare Environments: A Review," MDPI, Lisbon, 2020.

[50] F. Kariuki, "The Top 13 Benefits of Remote Patient Monitoring," Health Recovery Solutions, [Online]. Available: https://www.healthrecoverysolutions.com/blog/the-top-13-benefits-of-remote-patient-monitoring. [Accessed 23 07 2023].

[51] Aetonix, "Top 4 Challenges Of Monitoring Patients Remotely," Aetonix, [Online]. Available: https://aetonix.com/care-pathways/top-4-challenges-of-monitoring-patients-remotely/. [Accessed 24 11 2022].

[52] M. Khalifa, "Clinical Decision Support: Strategies for Success," Procedia Computer Science, Jeddah, 2014.

[53] M. Khalifa, "Clinical Decision Support: Strategies for Success," Procedia Computer Science, Jeddah.

[54] V. R. Chifu, C. B. Pop, D. Demjen, R. Socaci, D. Todea, M. Antal, T. Cioara, I. Anghel and C. Antal, "Identifying and Monitoring the Daily Routine of Seniors Living at Home," 27 01 2022. [Online]. Available: https://www.researchgate.net/publication/358172059_Identifying_and_Monitoring_the_Daily_Routine_of_Seniors_Living_at_Home/citations. [Accessed 26 11 2022].

[55] A. M. Khalili, A.-H. Soliman, M. Asaduzzaman and A. Griffiths, "Wi-Fi Sensing: Applications and Challenges," School of Creative Arts and Engineering, Staffordshire Universit, Staffordshire, 2019.

[56] D. Shain, "Top Advantages and Disadvantages of Microservices: Biggest Benefits of Microservices and Key Use Cases," Camunda, 09 02 2023. [Online]. Available: https://camunda.com/blog/2023/02/benefits-of-microservices-advantages-disadvantages/. [Accessed 20 08 2023].

[57] J. Ellingwood, "The benefits of PostgreSQL," Prisma's Data Guide, [Online].

Available: https://www.prisma.io/dataguide/postgresql/benefits-of-postgresql. [Accessed 28 07 2023].

[58] W. Akolo, "Which Is Superior: Node.js vs. Spring Boot?," hostadvice., 13 April 2023. [Online]. Available: https://hostadvice.com/blog/web-hosting/node-js/node-js-vs-spring-boot/. [Accessed 28 07 2023].

[59] R. Katiyar, "What is Rule-Engine?," Medium, 13 10 2019. [Online]. Available: https://medium.com/@er.rameshkatiyar/what-is-rule-engine-86ea759ad97d. [Accessed 28 07 2023].

[60] L. Rosa, P. G. Alves, T. J. Cruz and P. Simões, "A Comparative Study of Correlation Engines for Security Event Management," University of Coimbra, Coimbra, 2015.

[61] Zymr, "Using Drools vs. Esper Rules," Zymr, 04 01 2023. [Online]. Available: https://www.zymr.com/blog/using-drools-vs-esper-rules. [Accessed 29 07 2023].

[62] S. Modan, "The benefits of ReactJS and reasons to choose it for your project," peerbits, 09 January 2023. [Online]. Available: https://www.peerbits.com/blog/reasons-to-choose-reactjs-for-your-web-development-project.html. [Accessed 29 07 2023].

[63] Paty, "5 Reasons to use TypeScript for your Projects," Clickit, 09 01 2023. [Online]. Available: https://www.clickittech.com/developer/why-use-typescript/#h-why-is-typescript-better-than-javascript. [Accessed 29 07 2023].

[64] Stakify, "What is Storybook and Why Developers Should Use It," Stakify, 26 05 2023. [Online]. Available: https://stackify.com/what-is-storybook-and-why-developers-should-use-it/. [Accessed 20 08 2023].

[65] I. M. Ibrahim, O. F. Nonyelum and I. R. Saidu, "ITERATIVE AND INCREMENTAL DEVELOPMENT ANALYSIS STUDY OF VOCATIONAL CAREER INFORMATION SYSTEMS," International Journal of Software Engineering & Applications, 2020.

[66] git, "git --fast-version-control," git, [Online]. Available: https://git-scm.com/. [Accessed 20 08 2023].

[67] L. Gupta, "REST API URI Naming Conventions and Best Practices," REST API Tutorial, 30 07 2023. [Online]. Available: https://restfulapi.net/resource-naming/. [Accessed 02 08 2023].

[68] M. F. "A Brief History of Digital Health," medium.com, 05 02 2021. [Online]. Available: https://medium.com/that-medic-network/a-brief-history-of-digital-health-b238f1f5883c. [Accessed 14 11 2022].