FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

VitalCoViD Platform: web-based telehealth system for real-time monitoring of COVID-19 patients at home using wearable health devices

Diogo Mota Pinto

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: João Paulo Trigueiros da Silva Cunha

Co-Supervisor: Duarte Dias

September 29, 2021

VitalCoViD Platform: web-based telehealth system for real-time monitoring of COVID-19 patients at home using wearable health devices

Diogo Mota Pinto

Mestrado Integrado em Engenharia Informática e Computação

September 29, 2021

Abstract

The Coronavirus Disease 2019 (COVID-19) pandemic has generated a high demand for remote monitoring solutions, considering that most healthcare systems worldwide are suffering severe problems related to a shortage of health professionals. In addition, patients infected (or presumed to be infected) with COVID-19 are required to self-isolate.

For the COVID-19 patients quarantined at their homes, their medical follow-up is solely based on daily phone calls, relying on the patient's symptom and measurement descriptions. Remote health monitoring relates to monitoring patients outside the traditional hospital environment, usually in their homes. It allows monitoring patients' well-being and health conditions that require medical surveillance, such as COVID-19.

At INESC TEC, Biomedical Research And INnovation (BRAIN) Laboratory from the Center for Biomedical Engineering Research has been developing wearable health technologies to monitor physiological variables in real-time based on innovative devices connected in an Internet-of-Things (IoT) system. This API-based IoT system has already been deployed, presenting a high scalability level to any ambulatory monitoring scenario. Moreover, it can be easily adapted for COVID-19 patients' home monitoring, sharing their physiological status with the clinicians.

In this Master Thesis, it is presented a potential application of the IoT in healthcare by designing and implementing a web-based and IoT-based monitoring system with a user-friendly interface for clinicians to monitor their patients at home in real-time.

VitalCovid is a web-based telehealth system for real-time monitoring of COVID-19. The system aims to assist clinicians in their patient follow-up by providing all the necessary data to make a reliable assessment of the patients' health status. The system features a mobile application and a web-based medical dashboard. The patient-oriented mobile application functions as a data aggregator by receiving the vital signs from the Bluetooth wearable devices and broadcasting them through the internet. The dashboard provides the clinicians reliable data of each patient, facilitating their medical follow-up without the need for phone calls. VitalCovid has shown promising results. In interviews with healthcare professionals, the consensus was that this system could replace the need to follow up via phone calls.

Keywords: Remote monitoring, Vital Signs, COVID-19, mHealth, Web Development, Software Engineering

Resumo

A pandemia de COVID-19 gerou uma necessidade sem precedentes de monitorização remota de pacientes, considerando que maior parte dos sistemas de saúde mundiais sofreram problemas graves devido à escassez de profissionais de saúde. Para além disso, pacientes infectados (ou sobre suspeita) com COVID-19 devem isolar-se. O seguimento médico dos doentes com COVID-19 em quarentena baseou-se apenas em chamadas telefónicas diárias, dependendo das descrições dos sintomas relatados pelos próprios doentes ou cuidadores. A monitorização remota de saúde está relacionada com a monitorização de indivíduos fora do ambiente hospitalar, geralmente nas suas casas. Permite observar o bem-estar dos pacientes e condições de saúde que requerem vigilância médica, como é o caso do COVID-19.

No INESC TEC, o laboratório BRAIN do centro de Investigação de Engenharia de Biomedicina tem vindo a desenvolver tecnologias *wearable* para a saúde para monitorizar variáveis fisiológicas em tempo real baseadas em dispositivos portáteis inovadores ligados a um sistema IoT. Este sistema web já foi implementado e apresentando um alto nível de escalabilidade para qualquer cenário de monitorização ambulatória, sendo fácil de adaptar para acompanhamento de pacientes domiciliários com COVID-19, partilhando com os profissionais de saúde, em tempo real, o seu estado fisiológico. Nesta tese de mestrado, é apresentada a potencial aplicação de IoT em cuidados de saúde, desenvolvendo e implementado um sistema web de monitorização com uma interface intuitiva para os profissionais de saúde acompanharem os seus pacientes confinados em tempo real.

VitalCovid é um sistema web de *telehealth* para monitorização em tempo real de pacientes com COVID-19. O sistema pretende ajudar profissionais de saúde no seguimento dos seus pacientes, dando acesso a todos os dados necessários para uma avaliação fiável do estado de saúde dos mesmos. Este sistema utiliza uma aplicação móvel e um dashboard médico. A aplicação orientada a pacientes funciona como um agregador de dados, recebendo os sinais vitais recolhidos pelos dispositvos Bluetooth e enviando-os através da internet. O dashboard médico fornece aos profissionais de saúde dados fidedignos de cada paciente, facilitando o seu acompanhamento médico, eliminando a necessidade de realização de chamadas telefónicas. O VitalCovid mostra resultados promissores. Em entrevistas com profissionais de saúde, a opinião geral foi de que este sistema poderia substituir o acompanhamento por chamadas telefónicas.

Keywords: Monitorização Remota, Sinais Vitais, COVID-19, mHealth, Desenvolvimento Web, Engenharia de Software

Acknowledgements

First, I would like to thank my family for their outstanding support; they are the main ones responsible for where I am today. A special thanks to my friends who always motivated me to be the best version of myself and showed me how rewarding work can be. To the friends university has given me, thank you for the support both is in life and throughout these years. Also, with great regard, thank you to the ones who have helped throughout the dissertation, with special thanks to those who kept me company day after day.

A very special thank you to my supervisor Dr. João Paulo Cunha. Thank you for granting me this opportunity; it has been a blast to work with such a talented and heartwarming team. To my co-supervisor, Duarte Filipe Dias, who has been tireless, thank you for all your support throughout the dissertation and thank you for all the things you taught me.

Finally, I would like to thank the FEUP community for giving me the proper tools to become a great person and professional.

Diogo Mota Pinto

"More is lost by indecision than wrong decision."

Tony Soprano

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
BMI	Body Mass Index
BRAIN	Biomedical Research And INnovation
CHUSJ	Centro Hospitalar Universitário de São João
COPD	Chronic Obstructive Pulmonary Disease
COVID-19	Coronavirus Disease 2019
DGS	Direção Geral de Saúde
EHR	Electronic Health Records
FR	First Responder
HTTPS	Hypertext Transfer Protocol Secure
ICU	Intensive Care Units
IoT	Internet of Things
JEE	Java Enterprise Edition
NHS	National Health Systems
ORM	Object–relational mapping
REST	Representational State Transfer
SARS-CoV-2	Severe Acute Respiratory Syndrome Coronavirus 2
UI	User Interface
VR	Vital Responder
WHDs	Wearable Health Devices
WHO	World Health Organization

Chapter 1

Introduction

The COVID-19 pandemic was provoked by the Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). The virus was identified in December 2019 in Wuhan, China, and only later in January 2020 was it constituted as a Public Health Emergency of International Concern by The World Health Organization (WHO) [4].

SARS-CoV-2 pneumonia clinical spectrum is highly variable, ranging from almost none to seriously ill cases, and demands early detection and monitoring.

Measures were put into place to prevent contamination in Hospitals and other clinical environments, leading to a reduction of personal for routine care. Recommended preventive measures include continuously monitoring and self-isolating people exposed or symptomatic. Since the patients have to self-isolate, it has generated an unprecedented need to remotely and effectively monitor patients' well-being and health conditions.

IoT development brings several new applications, including in smart healthcare. Remote health monitoring is a healthcare delivery method that employs the latest advancements in information technology to collect patient data outside the classical healthcare environment [5]. The most effective remote patient monitoring system relies on consumer-friendly solutions, leveraging technology in a way that makes patients feel satisfied with helping manage their health. Not only are patients incentivized to engage with their health, but healthcare professionals are also better fitted to understand and manage their patients' health situations.

This thesis presents a novel solution for clinicians to monitor patients at home, a web-based telehealth system for real-time monitoring of COVID-19 patients at home using wearable health devices (WHDs).

1.1 Context

The COVID-19 pandemic has generated a demand for remote patient monitoring due to the severe shortage of healthcare professionals, and medical equipment [6]. In most countries, individuals infected or suspected of being infected with the COVID-19 virus are required to self-isolate at home to contain the spread of the illness. Currently, the medical follow-up for COVID-19 patients is based on phone calls. This process is very time-consuming and depends on the patients' measurements and symptoms description. Therefore there is a strong need for alternative medical solutions such as remote patient monitoring [6, 7]. Remote health monitoring is a type of medical monitoring that operates outside the hospital setting. It allows healthcare professionals to monitor patients remotely and provide the necessary care. At INESC TEC, BRAIN Laboratory has been developing remote monitoring solutions that are scalable and easy to integrate with third-party systems [3, 8]. These remote monitoring solutions could be easily adapted and integrated into a platform for COVID-19 patients home monitoring.

1.2 Motivation

IoT is promising for developing remote healthcare monitoring systems [9]. This technology is rapidly evolving, meaning that it will become easier to implement and maintain, enabling new opportunities for a broader range of activities to benefit from it [10].

The COVID-19 pandemic has generated an unprecedented need for remote monitoring due to shortages regarding medical equipment and health professionals since the healthcare systems are overloaded [**?**]. As a result, the medical follow-up is solely based on phone calls taking more time and effort from the healthcare professionals. On the other hand, remote monitoring of COVID-19 patients at home enables monitoring patients' well-being and health conditions [11] efficiently and smartly without wasting as many resources. Therefore, IoT and remote monitoring is a promising combination that may improve the infrastructures already in place in an efficient and time-saving manner and possibly without a substantial change in the medical routine.

Although this dissertation is written in the context of COVID-19, the research scope is not limited to telehealth solutions for COVID-19 only. The thesis should also consider remote health monitoring for other pathologies to understand its implementation in different contexts, benefits, and limitations.

1.3 Objectives

This dissertation aims to design and implement a monitoring system that uses wearable devices, mobile Apps, and user-friendly web interfaces for clinicians to monitor COVID-19 patients at home. To do this, the first step is to collect healthcare professionals' requirements and understand how such a system can be as most user-friendly as possible for both clinicians and patients. Once this preliminary phase is finished, a system architecture will be designed with a front-end and back-end to be furthered implemented conjointly with a database and a connection with the existing API for data collection. The last phase will be dedicated to evaluating the developed telehealth monitoring system's usability in real scenarios.

1.4 Thesis Contribution

The BRAIN Laboratory of INESC TEC has been developing WHDs for several years, assisting people in monitoring their vital signs providing accurate data to clinicians [3, 8]. This API-based IoT system has already been deployed, presenting a high level of scalability to any ambulatory monitoring scenario, being easy to adapt for CoViD-19 patients home monitoring, sharing in real-time their physiological status with the clinicians at the hospital. The project to be developed will be using the work previously done by the BRAIN Laboratory of INESC TEC and adapt it to fit its demanding requirements.

The INESC TEC project, WeSENSS, was first developed to create a novel wearable monitoring solution for hazardous professionals and has improved ever since. The WeSENSS aims to provide services to manage and review information of teams and individuals during monitoring sessions. Since it is scalable and easy to integrate with third-party systems, it can significantly benefit the VitalCovid platform, removing the need to implement a module for managing this type of data.

Although the VitalCovid platform focuses on monitoring COVID-19 patients, such a system can be easily adapted to different contexts, permitting the measurement of other vital signs and the addition of other features. The platform could provide clinicians a novel approach to remote monitor all kinds of patients outside the hospital environment.

Chapter 2

State of the Art

This chapter describes the state of the art of telehealth monitoring systems in the IoT. This first Section represents the research context and explains why this research is essential to understanding the study's main aspects. It is vital to understand what remote health diagnosis and monitoring options other countries have applied to understand what gaps need to be addressed. In Section 2.1 is explored the state-of-the-art telehealth monitoring systems in the IoT. The last Section 2.2 presents a conclusion regarding this chapter's findings.

COVID-19 is an infectious disease provoked by the SARS-CoV-2 virus [12]. It was first was identified in December 2019 in Wuhan, China, and it has since spread, leading to a global pandemic. Although symptoms may vary, they usually include breathing difficulty, cough, fever, fatigue, loss of smell, and flavor. Symptoms usually take between one to fourteen days to manifest after being in contact with the virus. Most people exhibit mild symptoms, while some develop severe to acute symptoms and may be admitted to a hospital regarding the disease's progression [13]. More than a third of the population infected with the virus is asymptomatic, but they can still spread it [14]. Infected people can spread the virus up to two days before showing any symptoms and remain infectious for up to two weeks in severe cases.

Preventive measures include social distancing and quarantining. In addition, self-isolation has been an effective measure to prevent the virus in cases where the person might be infected or tested positive for the virus. Health agencies have guidelines for proper isolation. In addition, several governments have issued a quarantine for entire populations to reduce the number of new infections.

2.1 Research studies and existing solutions

Remote health monitoring uses digital devices to collect medical and health-related data from individuals in one location. It transmits that information via secure channels to another site later analyzed by a healthcare professional for assessment and recommendations.

Monitoring devices can collect a wide range of health data, such as weight, blood sugar, blood oxygen levels, blood pressure, respiratory rate, heart rate, and electrocardiograms. Monitoring applications can also help people be healthy, allow elderly and disabled individuals to live at home longer, avoid moving into nursing facilities, and reducing hospital admissions, therefore improving quality of life while saving costs [1].

It is impossible not to talk about m-Health when discussing remote monitoring. Mobile health consists of utilizing mobile devices to aggregate health data and send it to the Internet [15]. The devices generate data by measuring the patient's vitals and send it to designated servers. The data can provide sensitive information about the patient's condition, leading to a diagnosis, treatment, or hospitalization in a medical emergency.

Telehealth technologies are potential ways to make better use of healthcare resources. Telehealth is not a new concept. Most hospitals have partly or fully implemented a telehealth system subset. However, the majority of telehealth-related projects have a small and restricted scope [2]. The Internet provides a chance to make telehealth programs more accessible, allowing third-party developers to contribute to its improvement and growth. Telehealth employs technological applications ranging from biosensors to cloud solutions using artificial intelligence (AI). Adopting AI solutions can improve health outcomes while also benefiting the stakeholders [16]. Advancements in technology bring improvements and affordability to wearables. Combining wearable technology with AI delivers reliable and valuable data to clinicians, enhancing the diagnosis process and treatment prescription [2].

Throughout the years, several solutions have been researched and implemented with the most distinct architectures.

A research team in China implemented a WeChat-based remote monitoring model to evaluate its effectiveness in addressing the challenges of monitoring COVID-19 patients outside the clinical environment [17]. The team utilized the WeChat mobile application to communicate between the medical teams and the COVID-19 patients. WeChat is a multi-purpose mobile application developed by a Chinese company, and its main features are instant messaging, and social media [18]. The monitoring process begins when a patient enrolls in the telehealth platform. The first step is to fill in a form to provide the clinicians with medical and personal information crucial for the medical assessment. Then, the patient undergoes a quarantine control evaluation. The multidisciplinary team has a set of criteria to evaluate the severity of each patient's condition. Regarding the control evaluation, the team must decide if the patient should be quarantined at home under medical observation or if the patient should be admitted to the hospital. While using this system, the patients must update their health status daily by filling the team's form. Additionally, any clinician can instantly access the information regarding the patients' conditions. For each patient, the observational attention is adjusted based on the disease's progressions. The monitoring results provided insights into the benefits of early intervention. The monitoring teams were able to understand better the disease's progressions preventing delayed hospitalizations.

Another research team in Italy implemented a telemedicine system for remote management of chronic patients[19] . The system aims to monitor patients with Chronic Obstructive Pulmonary

Disease (COPD). The system proposed by the team was composed of several monitoring kits distributed at patients' homes and a server with a database installed in the provider's facility. These sub-systems allow for bi-directional communication via the Internet, exploiting several communication technologies. The system's client-server architecture is displayed in Figure 2.1.

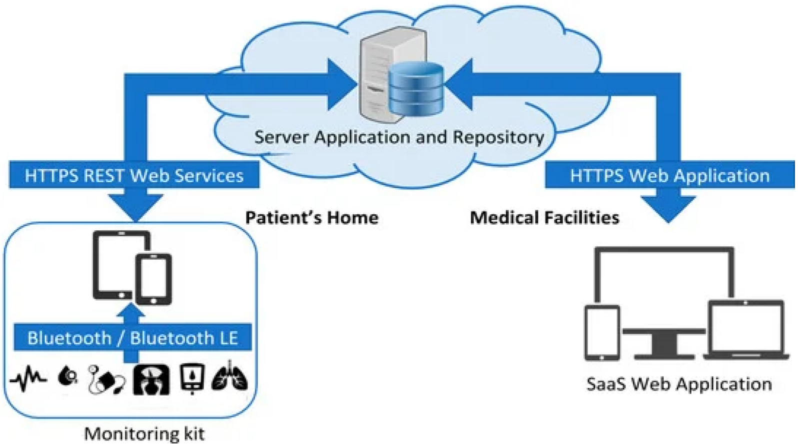

Figure 2.1: The architecture of the Italian remote management system of chronic patients [1].

The server represents the centralized node of the system. The server uses Representational State Transfer (REST) web services to access resources, upload data, and receive configuration updates remotely. The server also runs specific algorithms for data analysis to evaluate the patient's condition and to raise alarms in specific circumstances. The packets transferred by the server and the other services follow a set of XML tags. The web-based application allows healthcare professionals to visualize and manage the patient's data by using web browsers. The repository is a database and handles the patient's electronic health records (EHR). The database stores the data gathered by the wearable devices, the data entered by the healthcare professionals during the remote monitoring phase, and the data created by the algorithms. Authenticity, Integrity, and Security are ensured by the use of Hypertext Transfer Protocol Secure (HTTPS) protocol, which is utilized for reliable and secure communications over the network, and user authentication. The architecture was implemented with Java Enterprise Edition (JEE) and Spring Web and Security Frameworks. The database is a relational database implemented in MySQL. The system allows clinicians to schedule monitoring activities of specific vital signs. Simultaneously the care team could set notifications in case some alarms are generated during the activities. There are specific interfaces where the clinician can monitor the patient's vital signs in real-time with charts and other detailed information. There is also a feature for clinicians to specify each patient's rules when an alarm should be raised. The clinicians can define thresholds for specific vital signs that should not be surpassed.

All of the systems mentioned above take advantage of IoT. IoT plays a significant role in telehealth remote monitoring, considering it allows bidirectional communications, providing clinicians with data in real-time through the network. Therefore, it safe to say that telehealth relies highly on IoT to function correctly and seamlessly.

2.1.1 COVID-19 monitoring/follow-up solutions

There has been an increasing interest in developing or adapting current technologies to assist healthcare with the COVID-19 pandemic. In addition, the pandemic has generated a need for telehealth systems.

Portugal is an early adopter of telehealth. Many initiatives have been adopted and developed since the '90s. For a long time, real-time remote consultations, remote screenings in dermatology, and remote monitoring for chronic diseases [20]. These systems allow video chats and clinical document sharing, such as images and lab test results. Users can access the online citizen area to consult their health records, prescriptions, and schedule medical appointments. Furthermore, the MySNS mobile app is a tool that allows the user to consult news from the National Health System (NHS), consult health information, provide a list and map health institutions. The broad usage of these telehealth tools is a clear sign of their widespread adoption. It is safe to say that there have been significant advances in adopting telehealth systems in Portugal, and those advances have been gradually changing and improving the country's NHS. A group of Portuguese researchers developed a remote monitoring system for COVID-19 patients, eCoVig [21]. This monitoring system uses a proprietary Bluetooth device to measure the patients' vital signs, such as temperature, heart rate, and oxygen saturation. The Bluetooth device sends the data to the mobile device. The data is displayed on the eCovig mobile application and sent to the cloud, providing the monitoring results to the clinicians in real-time. As previously stated, the medical follow-up in Portugal is solely based on daily phone calls. According to the researchers, this system promises to replace this weak interaction. The main modules of this system are the mobile application, the patient dashboard, the management dashboard dedicated to the clinicians, and a back-end to process the data. Patients rely on the e-CoVig mobile application to collect the monitoring data and send that data to the back-end as soon as the mobile device is connected to the Internet. The application also allows defining reminders to hint the patient to do the monitoring activities regularly. There is also a web-based dashboard with analytics for the patients to check their monitoring report in further detail. Additionally, clinicians have a similar dashboard at their disposal that allows them to evaluate their patients' health status and receive alerts when anomalies regarding the patient's vital signs are detected. Researchers concluded that the preliminary tests done in a nursing home showed positive results. According to them, the residents who participated in the study adhered well to the program. Nurses also reported that sharing the medical data with clinicians was more accessible than ever.

In Canada, many initiatives have been arising to provide telehealth alternatives since the pandemic began. Toronto's University Health Network has modified a telehealth system for COVID-19 patients in isolation [22]. The system is composed of two cameras. A camera is faced at the patients and is used to observe them. In contrast, the other camera is aimed to monitor medical devices that measure vital signs, such as oxygen saturation and blood pressure. Healthcare professionals can communicate with the patient through audio communication devices. If the vital signs fall to critical levels, physical intervention is initiated. As the COVID-19 pandemic progressed, a significant share of in-person medical appointments were replaced by teleconsultations.

In the United Kingdom, Sensyne Health developed a web-based application called CVm-Health, to support individuals to self-monitor their health during the pandemic. The application allows users to provide information regarding their symptoms, vital signs, and medication intake. The Institute of Biomedical Engineering, University of Oxford, has been evaluating a concept of a virtual high-dependency unit to monitor patients remotely using WHDs [23]. The system is being used in isolation rooms and allows healthcare professionals to track the COVID-19 patients' condition from outside of the rooms. It uses WHDs to measure vital signs such as skin temperature, oxygen saturation, respiratory rate, and heart rate. The data is then sent through the hospital's Wi-Fi to a medical dashboard. The Imperial College of London is trialing a wearable sensor for remote health monitoring in partnership with NHS organizations [24]. The wearable device measures the temperature, respiratory rate, and heart rate every couple of minutes. The data collected by the wearables is later processed by an algorithm, which warns the medical staff if it detects signs of deterioration.

2.1.2 Clinician's Opinion on Remote Monitoring

A research team in Australia researched whether self-monitoring via telehealth equipment could reduce hospitalizations for people with COPD [1]. The researchers split a randomized group of patients into two groups. One group received telehealth equipment to monitor their vital signs. In contrast, the others did not receive anything. The vital signs recorded by the equipment were sent through a smartphone to a secure Web site where they were monitored by a healthcare professional. According to the research, the telehealth group had fewer hospital admissions in comparison to the control group. In addition, researchers noticed that telehomecare is becoming a viable solution as the population ages to aid people living with chronic illnesses. Participants were incentivized to participate in their health care by measuring their vital signs and answering questions related to their health daily. All this data was transmitted via smartphone to a secure Web site where they were monitored daily. The platform was responsible for alerting the nurse if a deviation from their typical vital signs was detected. The research team concluded that the telehealth group had fewer hospital admissions and spent fewer days in the hospital than the control group, reducing the overall costs. Participants found the telehealth monitoring beneficial because it helped them identify if they were getting sick earlier, improving their recovery process due to seeking treatment before their condition became critical. In general, participants were satisfied with telehealth monitoring, describing that they had more control over their condition.

Another group of researchers [2] focused on describing the concept of telemedicine and its future. According to them, the ability to remotely monitor infected patients proves to be an essential tool in managing the virus's spread, reducing the number of hospitalization drastically. Therefore it alleviates the healthcare system. Telehealth is a helpful way to contain the COVID-19 virus's spread and provide primary health care to patients. Wearable devices can measure vital signs and raise alarms when the patient's vitals start to deteriorate. In this article, it is predicted that telehealth solutions will transform the health sector. There have been enormous expectations for telemedicine to grow in the following years, as seen in Figure 2.2. The development of COVID-specific applications, including real-time remote monitoring systems, has been accelerated by the spread of the virus and the quest for means to contain it.

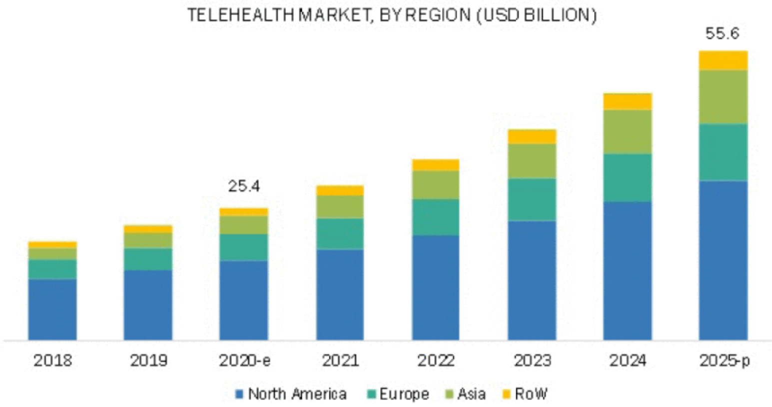

Figure 2.2: The global telehealth market expected growth [2].

Researchers concluded that the time for telehealth has arrived, and online business will continue to grow. Telehealth is projected to become a standard tool in healthcare. However, telehealth does not replace the "seeing of the doctor" part. It augments the interaction mode that already exists, allowing a full spectrum of information to be exchanged between patient and doctor.

2.1.3 Previous Work: The VitalResponder® platform

Concerning the dissertation topic, this research [3] is particularly relevant since it will be the development starting point. The VitalResponder® (VR) platform is a wearable monitoring solution composed of on-body and body-area digital sensors developed by INESC TEC, BRAIN group.

The VR aims to collect individuals' vital signs and surroundings using an on-body and body-area sensing network. Its goal is to provide First Responders (FR) reliable real-time indicators if exposed to unhealthy environments and evaluate their stress level. Therefore, supporting FR decision-makers with reliable data improving the team's management.

Each FR bears a VR-Unit Application running on mobile devices. Its goal is to collect data from the sensors connected through Bluetooth and send it to a VR-DataCollector. Then VR-DataCollector is responsible for sending the data to the network to be consumed by other gateways and stored in the VR-Data services. The system's architecture is shown in Figure 2.3.

In the absence of an internet connection, the VR-Unit data can be sent to the network once it is back online. This approach is used to combat VR's limitations in harsh environments where internet connection is not available. Every location where there is an ongoing operation has a VR-DataCollector node. This node is used to gather the nearby VR-Units' data and send it to the network. All the data transmitted by the node is stored in the VR-Data, where it can later consume

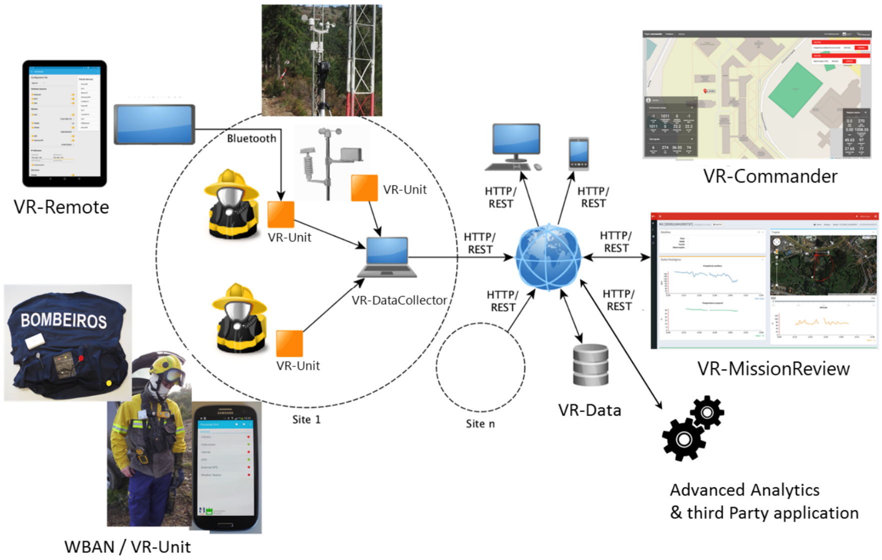

Figure 2.3: VR architecture [3]

through a REST API. This API allows external entities to retrieve information regarding the missions. Additionally, the VR-DataCollector API can be used by applications with visualization tools and can also integrate dashboards to monitor current or previous missions. The VR-DataCollector has evolved to this day, and now it is called WeSENSS. The WeSENSS presents a high level of scalability, therefore it should be an excellent addition to the VitalCovid platform removing the need to implement a new system to manage the patients' vital signs. Like the WeSENSS, the VR-Unit, now WeSensor is a mobile application that functions as an aggregator. The mobile application receives the data from the WHDs and sends it to the internet. This application can be easily adapted to fill the requirements for COVID-19 remote monitoring. Throughout the research, [3] it was also developed a remarkable wearable device. The VitalSticker is a WHDs that measures vital signs such as temperature, heart rate and can also acquire medical-grade quality ECG. Since the temperature and heart rate can be key indicators of a patient's health status, this device could potentially be used in addition to an oximeter to measure all the required vitals for monitoring COVID-19 patients.

2.2 Conclusion

In conclusion, nowadays, IoT systems are more accessible than ever to the general population, expecting to grow in the following years. Researchers believe its adoption may bring many benefits to health care. Therefore the number of solutions for remote monitoring is ever-growing such as its research. Many solutions have been developed, and the most promising ones use stacks similar to the one proposed in this dissertation.

The systems reviewed were very effective in reducing the number of patients admitted to the hospital, and the main factor to the success was patient participation. The more willing a patient was to participate, the better results it has shown. Thus, participation should be considered a challenge, and the system should be as user-friendly as possible to encourage the subjects to adopt it. The system's goal proposed herein is to monitor patients at home while their condition is not yet life-threatening. Therefore, the vital signs to be taken into account should indicate whether the patient should be hospitalized. During our findings, vital signs such as body temperature, oxygen saturation, and heart rate are considered a must-have for COVID-19 medical follow-up. In contrast, respiratory rate and blood pressure are helpful in severe cases but not essential for the context of the problem. Security should also be taken into account since the system deals with sensitive information.

Chapter 3

Proposed Solution

3.1 Problem and proposed solution

As previously mentioned, the COVID-19 pandemic has generated an unprecedented need for remote patient monitoring, considering many NHS are facing shortages related to clinicians, healthcare equipment, and ICU. Additionally, in most countries, patients who test positive for COVID-19 are required to self-isolate at home. Their follow-up is solely based on phone calls, relying on the patients' or carers' symptom measurement descriptions.

Based on the literature review, a web-based telehealth approach seems to be the most common candidate for our proposed system. Studies have shown that remote telehealth monitoring improved the patients' quality of life, resulted in fewer hospital admissions, and reduced overall costs. Fewer hospital admissions combined with the ability to free clinicians that otherwise could perform other tasks rather than making calls for the patients' follow-up may be a significantly efficient method to prevent shortages.

Nowadays, clinicians in Portugal have at their disposal a system, TraceCovid-19, provided by the Portuguese government to monitor and manage COVID-19 patients. Even though the system helps support healthcare professionals to follow-up and manage COVID-19 patients, it does not have any automatic features. Due to a lack of automatic features, healthcare professionals are dependent on phone calls to gather necessary information from patients. Adding extra features, such as access to the patients' vital signs in real-time, could greatly benefit the monitoring process.

There is a vast potential to make use of IoT telehealth, wearable, and mobile systems to support clinicians by developing a clinician-oriented web application that receives physiological data from a patient-oriented mobile application. The mobile application can run on an Android mobile device and functions as a data aggregator taking advantage of WHDs that broadcast the patient's vital signs via Bluetooth to the mobile device. Providing all the necessary information for the clinician to perform the health assessment task could completely replace the need to contact the patient via phone call. Therefore, the VitalCovid platform could be an excellent add-on to TraceCovid, improving the clinicians' workflow by simplifying the most time-consuming task.

The following sections are dedicated to the discussion of the proposed system in detail.

3.2 Development Objectives

The goal is to develop a remote monitoring system for clinicians to monitor COVID-19 patients outside of the classic hospital environment. The system is composed of a mobile application running on an Android mobile device dedicated to the patients and a web-based dashboard dedicated to healthcare professionals.

First and foremost, it's essential to split our goal into four main objectives.

3.2.1 Develop a dedicated server.

Develop a server connected to a database to store patients' personal information and additional relevant data, such as assessment forms with the patients' symptoms. This server should have an API to interact with the other system modules.

3.2.2 Improve the WeSensor mobile application.

The application needs to be improved in two different domains. The first one consists of adapting the WeSensor mobile application to communicate with a Bluetooth Oximeter via Bluetooth. The second one is to integrate a new API and create new menus.

After examination, it is believed that knowing the patient's oxygen saturation value is extremely useful to determine the disease's progression. Therefore it is crucial that our platform can read and display this information. Since commercial Bluetooth Oximeter with open APIs were not found, the first approach is to reverse engineer a Bluetooth Oximeter to understand how the application can receive and read the values measured by such a device. The second step is to add a field on the application dashboard so the patient can visualize the oxygen saturation values in real-time. Finally, the last step is to send the data to the server.

The second significant improvement is to create new menus for the patients to report their symptoms and integrate the API to send the assessment form to the VitalCovid backend through this new API.

3.2.3 Create clinician-oriented dashboard.

This goal consists of developing a dedicated web-based dashboard according to the requirements gathered in interviews with clinicians. This dashboard aims to allow healthcare professionals to monitor their patients remotely. It should communicate directly with the dedicated server via an API. The features available in the dashboard will be discussed in detail in the following sections.

3.2.4 Implement a scheduling and notifications system.

The last objective is to implement a scheduling system where the clinician can set a monitoring schedule. The patient will receive notifications accordingly as a reminder to perform those monitoring activities.

3.3 Major Constraints

The web-based dashboard should be easy to use and intuitive even for the less tech-savvy users. Since we are dealing with sensitive personal information, it's essential to take data privacy into account. During the dissertation, sensitive patients' data will not be collected without their consent. The data will be safely sent and stored. It will be sent through a VPN and stored in a database held by INESC TEC. The integration between the mobile application and the oximeter should be flexible to integrate other devices of this kind.

3.4 Project Scope

The scope consists of a web-based dashboard for clinicians, a mobile application for the patients, and a server that handles and stores data.

The mobile application is an adaptation from the WeSensor App used on the VitalResponder project. New features need to be implemented, such as forms for medical assessment, additional vital signs necessary for the COVID-19 follow-up and push notifications to remind patients to fill the forms and measure their vital signs.

A web-based telehealth monitoring system for clinicians to remote monitoring their COVID-19 patients. This system should be a dashboard with features for visualizing the patient's data. The dashboard should communicate with a newly implemented backend via a GraphQL API.

The server is expected to communicate with the WeSENSS backend via a REST API. It should also be connected to a database to store data ensuring availability, integrity, and confidentiality.

3.5 Requirements Analysis

Two meetings were held with healthcare professionals to understand the requirements for remote monitoring of patients with COVID-19.

The first interview was with Dr. Margarida Tavares, infectious disease specialist at the University Hospital Center of São João (CHUSJ), a reference hospital in the fight against this pandemic. Her experience took place mainly during the first wave of CoVID-19 in Portugal when there was still no specialized system for registering and monitoring patients with this pathology. After this first wave, the follow-up of patients at home was carried out by family physicians using the Direção Geral da Saúde (DGS) system TraceCOVID. In Dr. Margarida's experience regarding patient follow-up, respiratory rate, heart rate, blood pressure, temperature, and oxygen saturation are critical signals that allow a clinician to notice any significant anomaly. These signals do not need to be monitored continuously but for some time (30-60 min) several times a day to have a daily overview of their evolution. Furthermore, it is essential to understand the patient's Body Mass Index (BMI), calculated using the patient's height and weight. At the moment the meeting took place, the patients monitored by Dr. Margarida are those who are hospitalized at the CHUSJ. In some occasional cases, some patients are discharged early due to the high clinical occupation.

In the second meeting, we talked with Dr. André Santiago, who works closely with the Trace-Covid Platform.

In contrast to the previous session, since we were interviewing someone whose task is to monitor patients through the TraceCovid platform, the goal of this meeting was to understand its limitations and features and get input on how we could improve the monitoring experience as a whole. According to Dr. André, the most important vital sign to measure and analyze is body temperature since it varies for all patients, making it the most reliable indicator of the disease progression. The DGS states that a patient can be discharged after ten days if the patient has been apyretic for the last two days of this period. Getting reliable data on the temperature is helpful to decide whether to discharge the patient since, in real-life scenarios, the measurements can be unreliable, and the patient is not always trustworthy. The temperature measurements should be periodic. Since the fever has 12 hours intervals, we concluded that two measures a day would be optimal. The second most reliable vital sign is oxygen saturation given that it is a good indicator of whether the patient should be hospitalized or not. Values below 94% are considered to be alarming and should be taken into consideration. Heart rate is a good sign to be considered if the patient has any chronic disease associated. Other significant symptoms for the doctor to assess the progression of the disease are the lack of smell and taste. These could be filled on a form by the patient if the clinician requests it.

The patient's state should be easily accessible to the clinician, which could be achieved by having pre-defined thresholds for the vital signs. It was discussed that vitals such as temperature and oxygen saturation should have pre-defined thresholds. A clinician should define the thresholds for vitals such as heart rate and respiratory rate.

It was suggested that the clinician should fill a form with some information regarding the patient's medical history (associated diseases) and personal data (height, weight, gender, age, etc.) when a patient is registered in the system. While a patient is being monitored, the patient should receive periodical tasks to fill in assessment forms to provide the clinician with better feedback on the disease progression.

3.6 User Stories

Based on the previous section, we can group the user stories into two classes:

- Admin - stories related to the admin.
- Clinician - stories related to the clinician.

- Patient - stories related to the patient.

The high-priority user stories are the ones fundamental to the main scope of the project. The medium-priority user stories are not essential, but they fit the project's scope and enhance the monitoring process and medical assessments. Ultimately, the low-priority user stories are features that improve the flow of the monitoring process but have no impact on the medical evaluation.

Every user story table is ordered by priority, and the features will be implemented in that exact order. Thus, every high to medium priority feature should be implemented throughout the dissertation, and the low priority features will be implemented once the main scope has been achieved.

ID	Title	Priority
US101	Register clinicians	High

Table 3.1: Admin User Stories.

ID	Title	Priority
US201	Pair the mobile application with a Bluetooth oximeter	High
US202	Read vital signs	High
US203	Send vital signs view	High
US204	Fill assessment form	Medium
US205	Report vital signs manually	Medium
US206	Receive notifications to from the clinician	Low
US206	Report medication intake	Low
US207	Send messages to the clinician	Low

Table 3.2: Patient User Stories.

ID	Title	Priority
US301	Authenticate in the medical dashboard	High
US302	Create patients	High
US303	Check patient's profile	High
US304	Check patient's vital signs data	High
US305	Check patient's assessment form data	Medium
US306	Set patient's thresholds for alarms	Medium
US307	Get notified if a patient's condition is critical.	Medium
US308	Notify patients to perform their monitoring activities	Low
US309	Receive pop-up notifications	Low
US310	Group patients by family/housing	Low
US311	Reply the patient with automated messages	Low

Table 3.3: Clinician User Stories.

Chapter 4

Software Design

This chapter describes the software design process. Software design is the process of planning and specifying software solutions to satisfy established requirements by following recognized design standards and principles. This chapter covers the problem-solving and planning of the web-based telehealth monitoring system.

4.1 System Overview

VitalCovid is designed to be an add-on to the TraceCOVID-19 platform. Clinicians use the TraceCOVID-19 platform to assess the COVID-19 patients' well-being. During an assessment task, the clinician contacts a single patient and then updates the patient's information according to the patient's responses. VitalCovid's goal is to reduce the duration of the task by minimizing the amount of information needed during the call or even replacing it.

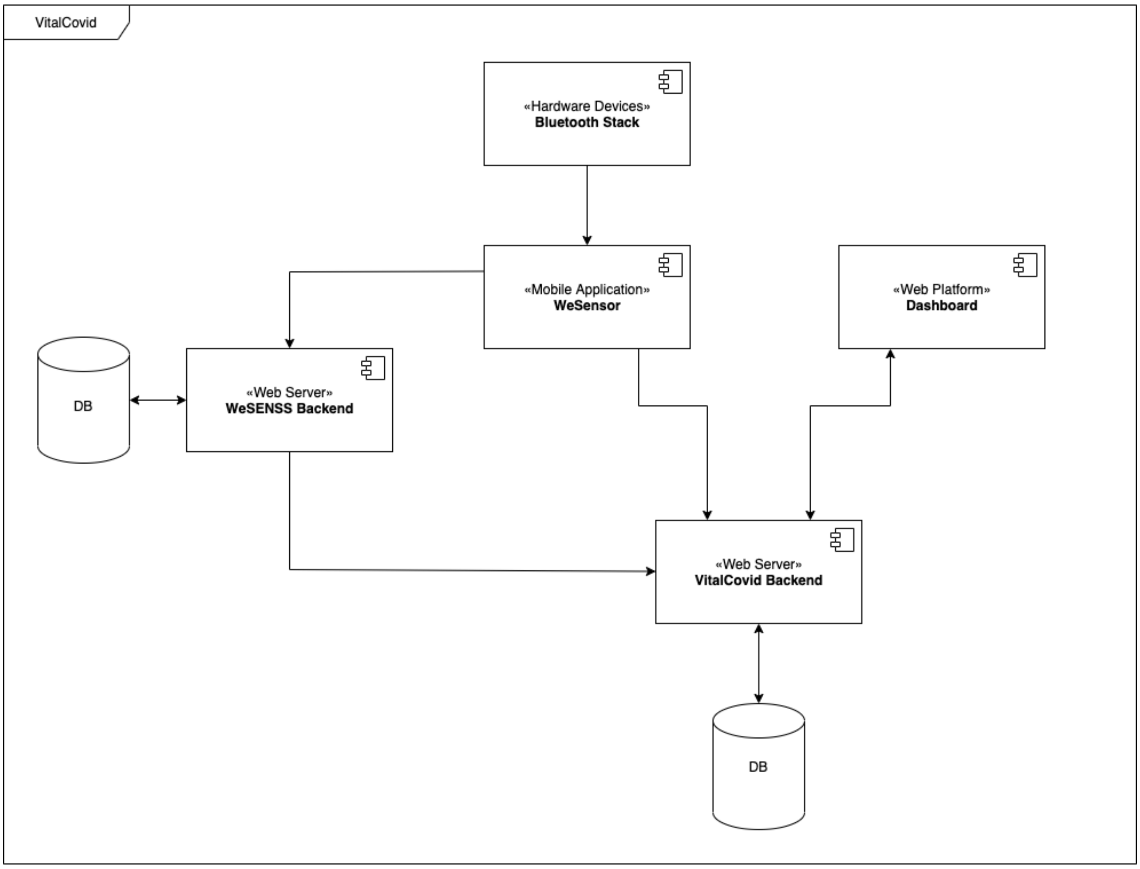

Figure 4.1: VitalCovid's system diagram overview.

Diagram 4.1 presents an overview of how the components interconnect in the VitalCovid system. The BRAIN Laboratory of INESC TEC developed the WeSENSS backend. Despite being critical for the functioning of the VitalCovid, this module did not suffer any changes throughout the project's development.

The Bluetooth stack relates to the predefined WHDs used to measure the patients' vital signs. These devices are connected to the mobile device via Bluetooth.

The mobile application is an adaptation from the WeSensor Application, developed by the BRAIN Laboratory. The previously mention application aggregates the vital signs data from the Bluetooth devices connected to a patient and sends them to the WeSENSS backend. Two more views will be added to the application, one view for the patient to report the symptoms and another view for the patients to manually report the vital signs measurements in cases where no Bluetooth devices are available. The symptoms form is sent to the VitalCovid backend, where the data is processed.

The VitalCovid dashboard is a web application, and it was developed in the context of this dissertation. The VitalCovid dashboard is a tool for clinicians to manage and monitor their patients.

The VitalCovid backend handles data storage and business logic. It communicates with all the endpoints through APIs.

The VitalCovid database stores all the information so it can be easily accessed, managed, and updated.

We will have a deeper look into each model in the following sections

4.2 System Architecture

The following diagram describes the system's architecture:

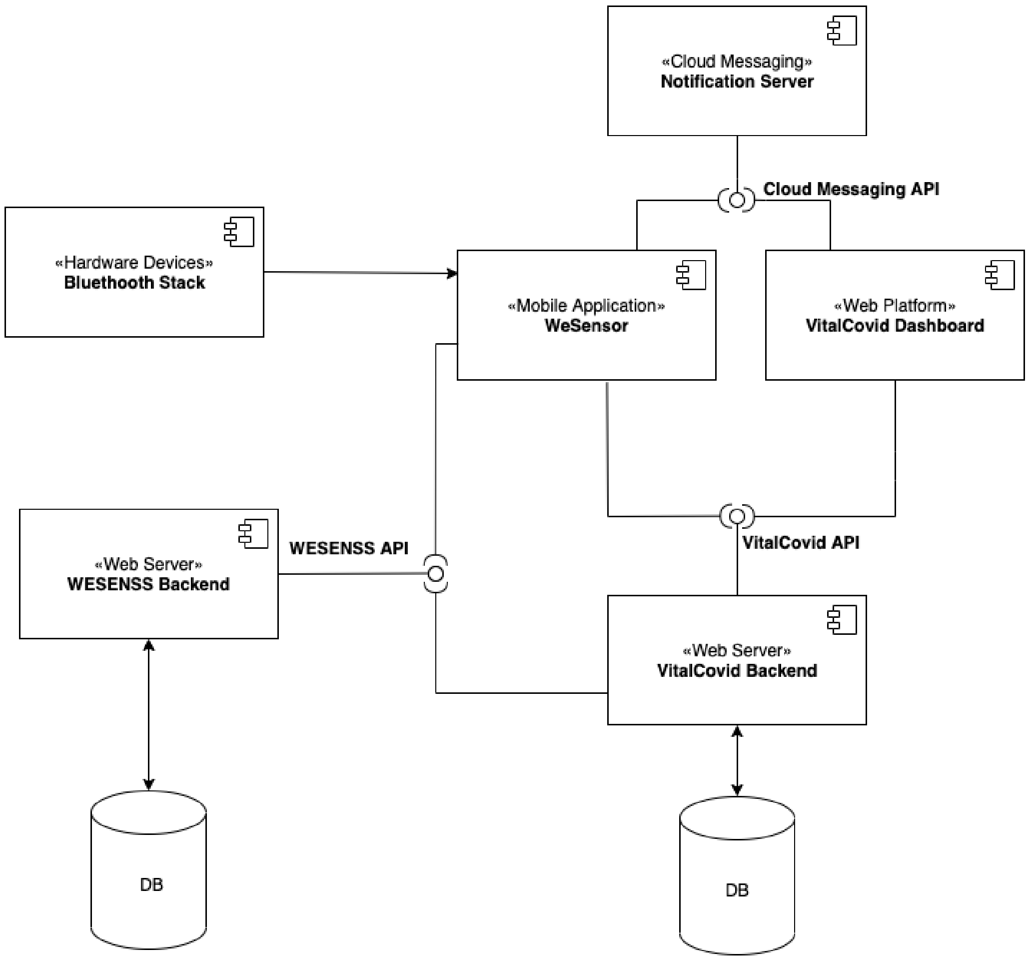

Figure 4.2: VitalCovid's Architecture.

VitalCovid's architecture is composed of several components, each one with a distinct role. The system will employ many different technologies, but all the components developed in this dissertation use technologies based on JavaScript. The web-based dashboard will be developed using Next.js. This React frontend development web framework allows developers to create robust web applications quickly and supports TypeScript, a superset of JavaScript. The dedicated VitalCovid backend will be developed using Express, a minimal Node.js web application framework, and some other JavaScript libraries to build a GraphQL server. Using the same programming language throughout the stack improves the development process. The communication between the clients

(WeSensor mobile application and VitalCovid frontend) and the server will be performed using GraphQL. GraphQL is a query language for APIs and a server-side runtime for executing queries using a type system.

The mobile application will receive the data from the hardware devices via Bluetooth and send it to the WeSENSS backend, where the information will be processed and stored. When a clinician opens a patient's profile, the VitalCovid backend queries the vital signs data from the WeSENSS backend through a REST API and sends it to the VitalCovid dashboard.

The VitalCovid backend interacts with a PostgreSQL database with the assistance of an Object-relational mapping (ORM). We will be using the Prisma ORM due to its integration with GraphQL libraries.

The WeSensor application was already taking advantage of Google's Cloud Firestore service to handle the notifications. Therefore this service should be integrated into the VitalCovid dashboard to send notifications directly to mobile applications when required.

The figure 4.3 below provides a visual representation of where the technologies previously state are going to be employed.

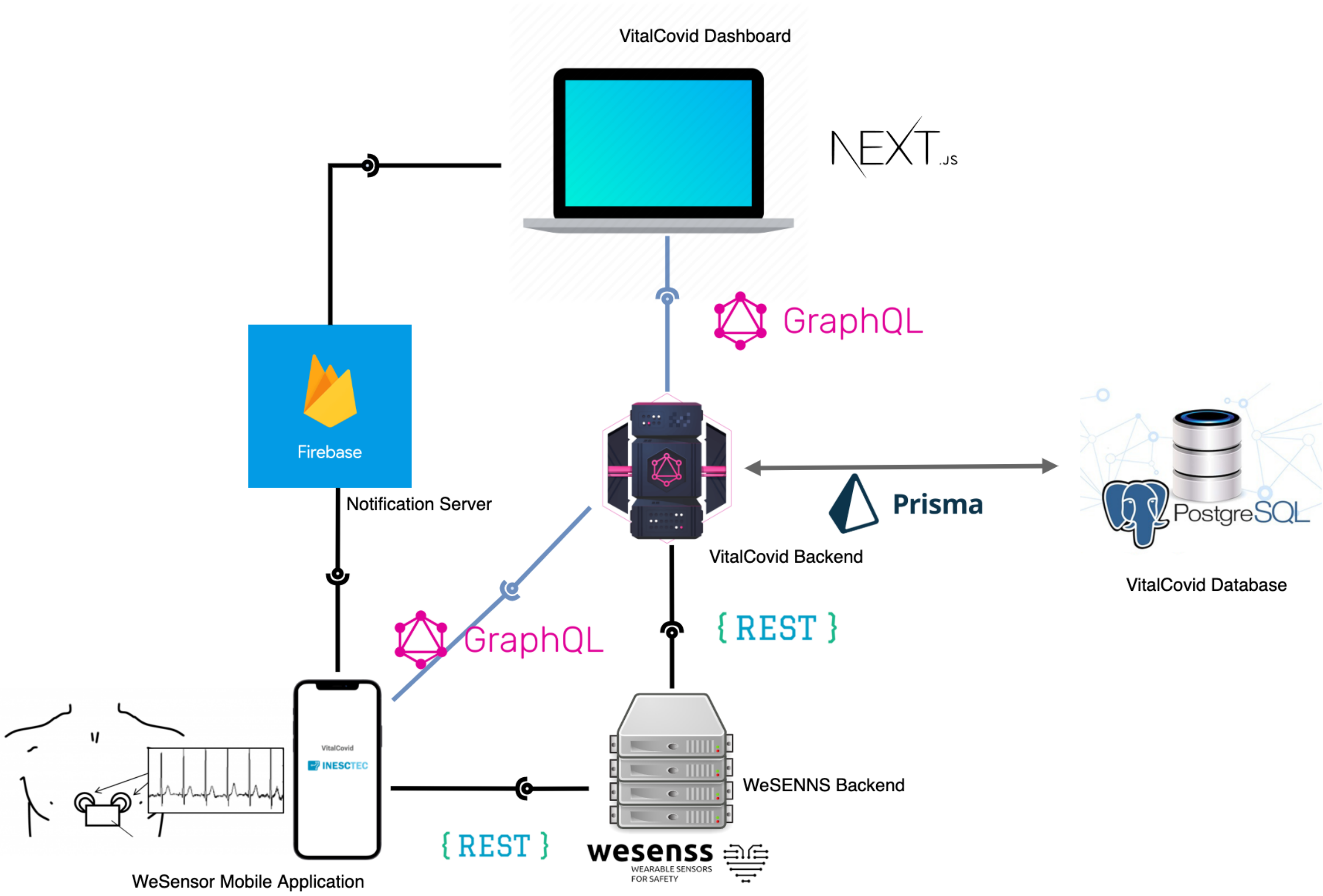

Figure 4.3: System's Technologies.

4.3 System Users Description

The table below describes the distinct user classes of the system and their main interactions. This description aims to explain what each type of user should achieve and their place in the system.

Name	Description
Patient	The patient is the central user of the system and uses the mobile application to perform the monitoring activities required. Should be able to perform the monitoring activities with the Bluetooth devices and fill the assessment forms.
Clinician	The clinician is accountable for zero or more patients. Should be able to access the dashboard and handle the patients, receive notifications regarding the patient's health status, define thresholds for each patient's vital signs, and request monitoring activities. A clinician can be a Doctor or a Nurse.
Admin	Manages the clinicians, can register clinicians in the system.

Table 4.1: User classes and Characteristics

4.4 Domain Model

The following diagram describes the conceptual model of the system. This diagram aims to conceptualize the architecture 4.4. Its utility is in formulating and expressing the functions, characteristics, and structure of the system. [25]

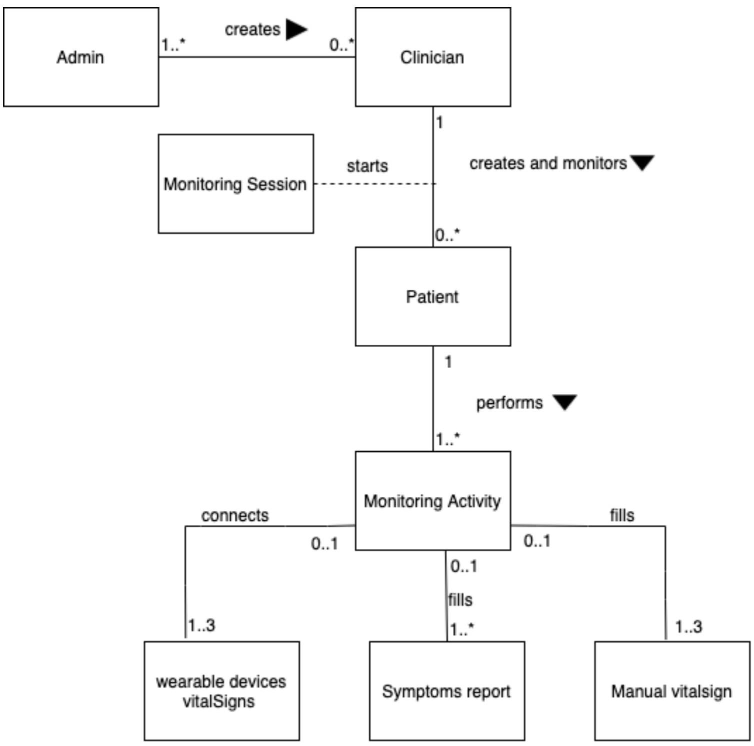

Figure 4.4: Conceptual Model Diagram

The system should be implemented in a Hospital where an admin is responsible for managing the clinicians' accounts. Then, the clinicians can register the patients infected with COVID-19 and start the monitoring session of each one. Monitoring activity is usually a two steps activity. The first step is mandatory, while the second one is optional but recommended.

The first step consists of measuring the vital signs. The patient should always send the vital signs to the system and has two methods of doing that. First, suppose the patient has the predefined Bluetooth devices available. In that case, the patient should pair them to a mobile device and use the WeSensor mobile application to collect and send the vital signs. Otherwise, if the patient does not have one or more of the predefined Bluetooth devices, the patient can measure the vital signs with regular devices and enter the values manually in the mobile application manual report screen.

The second step is to fill the assessment form. This step optional but highly recommended. This step aims to provide a list of the patient's symptoms so that the clinician can better assess the patient's health status.

4.5 Use cases

The use case diagram summarizes details of the system and the users within. The following diagram 4.5 describes the interactions between the main actors and the system:

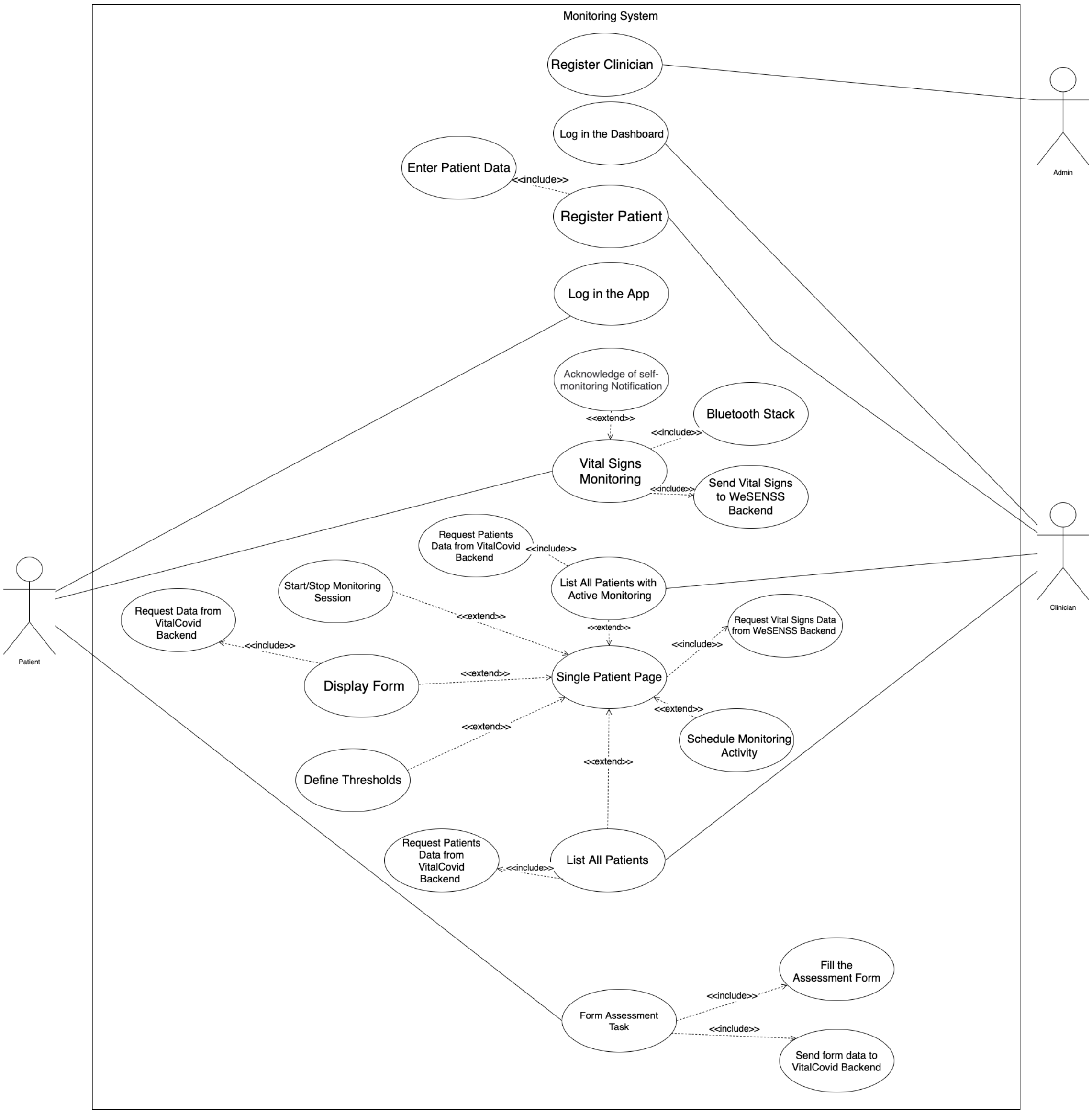

Figure 4.5: Use Case Diagram.

4.6 System Features

The primary actors of the system are the clinician and the patient, and the system operates around them. The following features need to be implemented in order to accomplish a well thought remote monitoring system:

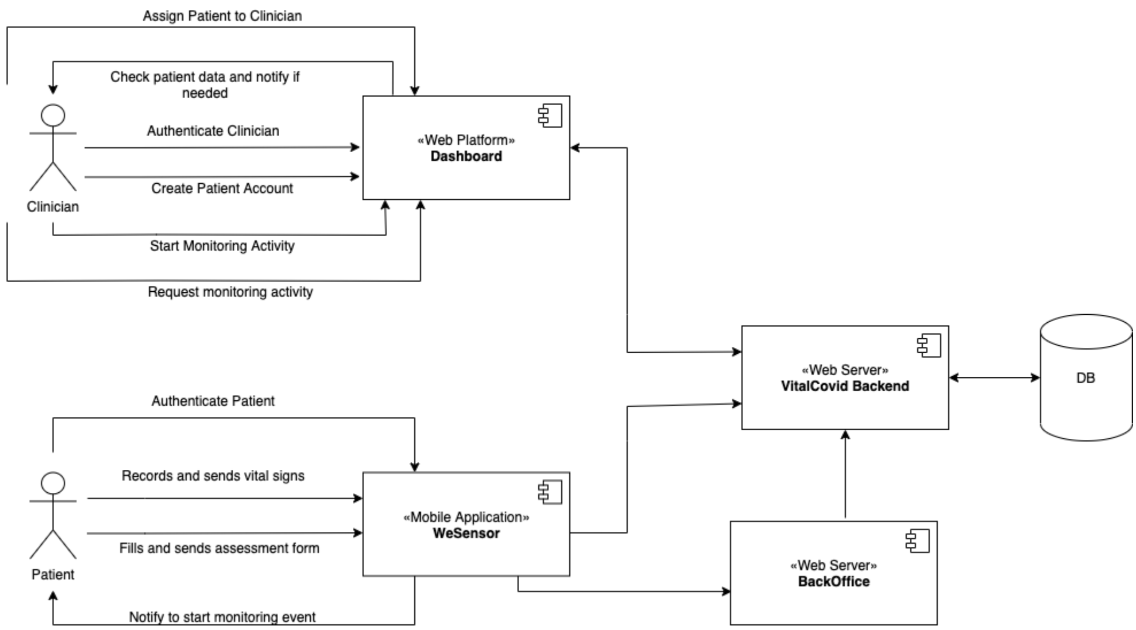

Figure 4.6: System Features

1. Authenticate clinicians in the dashboard application
2. Allow clinicians to register patients.
3. Allow clinicians to access their patient's profiles.
4. Authenticate patients in the mobile application.
5. Remind patients to do their monitoring activities.
6. Record and send the patient's vital signs.
7. Record and send data from the patient's assessment forms.
8. Display the recorded vital signs.
9. Display the patient's basic information and assessment forms.
10. Notify clinicians on the state of their patients.
11. Allow setting thresholds for their patient's vital signs.
12. Request the patient to perform a monitoring activity.

First and foremost, the process starts with the creation of a clinician account, this type of account only be created by an admin.

Then the clinician can log in to the medical dashboard. After logging in, the clinician is redirected to the overview page. On this page, he can get an overview of all the patients associated with him and is also able to register new patients. While registering new patients, the clinician must fill in basic information regarding the patient's health.

Once the patient is registered, the subject can log in to the mobile application to start the monitoring process. When the clinician activates the monitoring session for a specific patient, the patient will receive notifications to do monitoring events. To start the monitoring activity, the patient should tap the "Start" button on the overview screen to begin sending the data to the WeSENSS backend. The patient should also fill in a form with the symptoms so the clinician can make a better assessment of the patient health status.

On the dashboard landing page, the clinician has a list of all the associated patients. In this list, the clinician can overview their health status, age, and the last time they did a monitoring event. Clicking on a specific patient redirects the clinician o that patient-specific page. On that page, the clinician has access to essential information for the medical assessment, such as medical-related personal information, the results of the assessment forms, and the results of the monitoring activities.

Chapter 5

Software Development

This chapter describes the software development process. There will be a description of the development of each component illustrated in System Overview. As well, a more comprehensive look into the decisions behind the choice of technologies used to implement each component.

During the dissertation, the VitalCovid platform was implemented to assist clinicians in monitoring their COVID-19 patients outside the classic hospital environment. The system is composed of five major components as seen in figure 4.1 and four of them were fully or partially developed to comprise the system.

The mobile application was adapted from a previous project created and maintained by the BRAIN group. Furthermore, the VitalCovid dashboard, backend, and database were built to fit the context of the dissertation.

5.1 Backend

The server is one of the most critical components of the system. The server needed to be compliant with REST APIs while being a GraphQL server. Accordingly, in this section, the decisions made to build a well-implemented and scalable GraphQL server will be discussed.

5.1.1 Guidelines

The backend package follows the folder structure in figure 5.1.

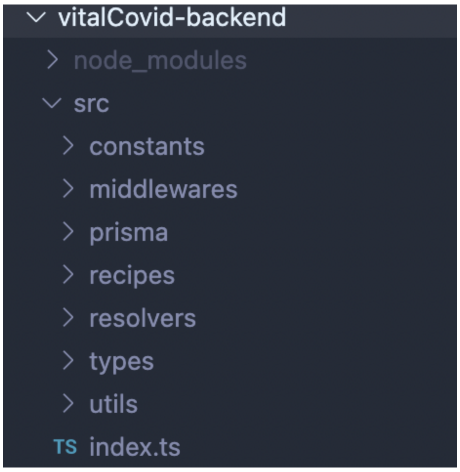

Figure 5.1: backend package code structure

The entry point is the index file, and then the code is split between seven main folders.

- constants - this folder relates to the system's constants, such as the WeSENSS API endpoint.
- middlewares - this folder relates to the middlewares. The middleware functions provide access to the following components: the request object, the response object, and the next middleware function. For instance, the AuthMiddleware is used to validate the authentication token from the request headers.
- prisma - this folder relates to the Prisma generated files and schema.
- recipes - relates to the type-graphql models.
- recipes - relates to the type-graphql models used by the resolvers.
- types - this folder relates to the TypeScript types.
- utils - this folers contains utility functions.

5.1.2 Implementation

5.1.2.1 Frameworks and Technologies

Since the frontend was going to be built using Typescript, keeping the same language is advantageous to the workflow — many libraries required for the project work exceptionally well with Typescript, taking full advantage of type annotations. In addition, having type annotations helps prevent many bugs, about 15% according to Adrian Colyer [26], and improves the developer experience.

Prisma is an open-source ORM. ORM is a method that allows query and manipulation from a database utilizing an object-oriented paradigm [27]. Prisma is a library that implements these techniques. In addition, there are many libraries like TypeGraphQL that work well with Prisma to develop a GraphQL Server using TypeScript.

Prisma has a feature called Prisma Schema, allowing developers to specify their application models in a data modeling language [1]. With the assistance of the Prisma Migrate tool, it is possible to migrate the schema, generating the SQL files in the relational database desired, and create the database. TypeGraphQL provides integration with Prisma [2], and it generates the type classes and resolver based on the schema defined.

5.1.2.2 Development

The first step was to build a server using Express.js and apply a middleware, Apollo Server, to generate a GraphQL server endpoint. GraphQL uses a schema to describe the shape of the data graph. The schema defines a hierarchy of types with fields and specifies which queries and mutations are available for clients to perform.

TypeGraphQL allows the developers to create queries, mutations, and field resolvers. The typegraphql-Prisma package automatically creates type classes and resolvers based on the Prisma Schema. Therefore, only custom resolvers with rich interactions, such as authentication resolvers and those who communicate with the WeSENSS, were manually implemented. Many of the resolvers, such as simple queries, were automatically generated. Accordingly, four crucial resolvers were manually created:

- PatientResolver - handles many of the operations related to the patient, such as the whole process of creating a patient.
- AuthResolver - handles the clinician and admin authentication.
- VitalsResolver - handles the actions related to the WeSENSS backend, such as assigning a patient to an event and fetching and processing the patient's vital signs.

[1] What is Prisma?, https://www.prisma.io/docs/concepts/overview/what-is-prisma,Last accessed on 2021-08-6

[2] TypeGraphQL & Prisma 2 integration, https://github.com/MichalLytek/typegraphql-prisma#readme,Last accessed on 2021-08-6

Prisma is responsible for all the interactions between the server and the database. The following figure 5.2 is a visual representation of all these technologies stack on each other and the flow of the interactions.

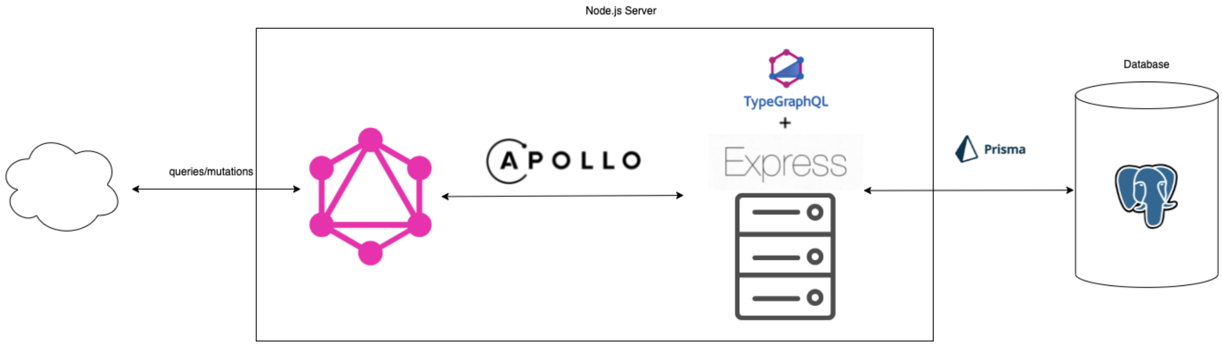

Figure 5.2: Backend Technology Architecture.

5.1.3 Security

Prisma Client mitigates the risk of SQL injection preventing attackers from modifying or deleting data from the database.

JWT validation is used to verify the incoming request preventing unauthenticated users from accessing sensible data.

5.2 Dashboard

This section describes the development process of the VitalCovid dashboard. First, the decisions behind the technologies adopted will be discussed, followed by a glance into the guidelines utilized to make the project as easy to understand as possible for newcomers. Lastly, there will be a deeper look into the implementation of the dashboard.

5.2.1 Technologies and Guidelines

The dashboard was developed using novel technologies that have shown remarkable results and are used by prominent companies, such as Hulu [28], Netflix, and Uber [3].

Next.js is an open-source React framework created by Vercel's CEO, Guilhermo Rauch, and backed up by Google, which has largely contributed to its growth [29]. It is a production-ready framework that empowers developers to create JAMstack web applications. Finally, Next.js is a framework often recommended for developers who want to create React applications, providing a layer of abstraction that fills some of React's gaps. It has many built-in features to improve

[3] Showcase, https://nextjs.org/showcase, Last accessed on 2021-05-6

the developer's experience, such as an intuitive page-based routing system, built-in sass support, TypeScript support, and much more [4].

React is an open-source JavaScript library created by Jordan Walke at Facebook [30] to create interactive user interfaces (UI). Nonetheless, it has many applications besides the web. For example, React can be used in combination with ReactDOM [5] to create single-page web applications; React Native it a React-based framework to develop hybrid mobile and desktop applications[6].

React's primary goal is to reduce bugs and achieves this through the use of components. React is declarative. It updates and renders just the right components when the data changes. Each component is a self-contained piece of code that defines a part of the UI. The composition of these components together creates a complete UI. React utilizes JSX [7] syntax, and it's an extension to JavaScript that allows nest components together like in HTML.

TypeScript is an open-source language developed by Microsoft which builds on JavaScript [8]. It adds static type definitions, allowing it to validate the code. TypeScript code is compiled into JavaScript. By using type-checking, it is possible to catch bugs effortlessly, preventing many runtime errors from occurring.

Tailwind CSS is a CSS framework. It was used to ensure a consistent design throughout the whole frontend, making the platform as user-friendly as possible for clinicians.

The frontend uses GraphQL to query data from the GraphQL server. Even though this topic will be discussed in section Backend, two fundamental concepts need to be defined. In GraphQL, it is possible two to perform two basic operations:

- Queries - fetch data just like a GET request in REST APIs.
- Mutations - change data just like a DELETE or a POST in REST APIs.

---

[4]Create a Next.js App, https://nextjs.org/learn/basics/create-nextjs-app?utm_source=next-site&utm_medium=nav-cta&utm_campaign=next-websit,Last accessed on 2021-05-6

[5]ReactDOM, https://reactjs.org/docs/react-dom.html,Last accessed on 2021-05-6

[6]React Native Learn once, write anywhere., https://reactnative.dev/,Last accessed on 2021-05-6

[7]Introducing JSX, https://reactjs.org/docs/introducing-jsx.html,Last accessed on 2021-05-6

[8]What is TypeScript?, https://www.typescriptlang.org/,Last accessed on 2021-06-6

5.2.1.1 Guidelines

Consistency is very important, so files are organized logically to be seen as a map to the project. Thus, allowing newcomers to quickly understand the project without looking at the documentation in depth. The image below illustrates the file organization of the dashboard package:

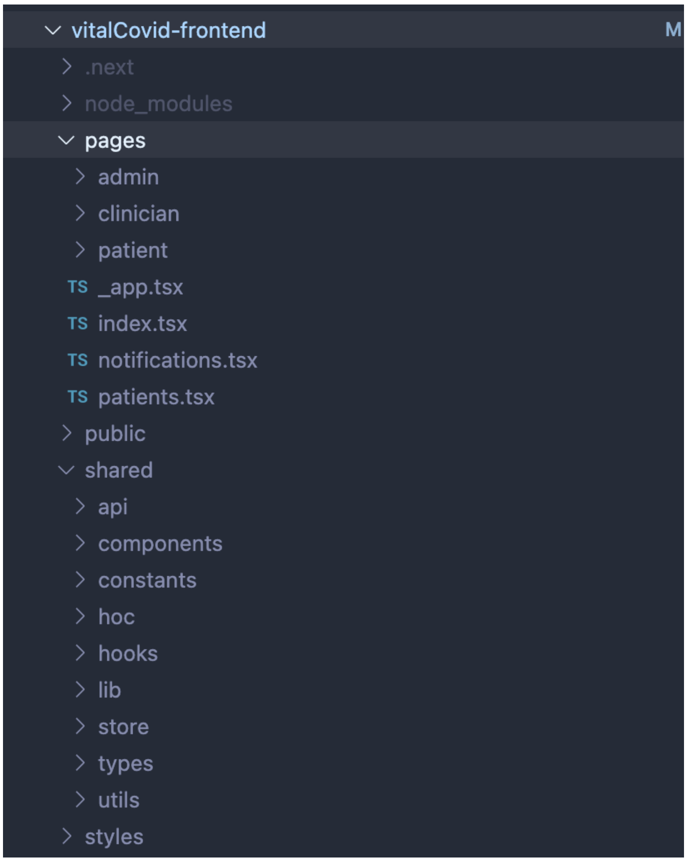

Figure 5.3: VitalCovid package code structure.

The package is organized according to the Next.js pagination guidelines [31]. Unlike other frameworks, React is not very strict when it comes to code conventions, or file organization [32].

The code is split into four main folders:

- pages - Following the next.js pagination rules, a page is a react component file in the directory. For example, the file *patients.tsx* will be accessible at */patients* route. There are also pages with dynamic routes; for example, *pages/patient/[uuid].tsx* will be available at */patient/1ecf6ca1-fb84-423f-a2b8-11ba3b28e2a2*.

- public - Next.js can serve static files under a folder called public in the root directory [33].

- shared - This folder relates to pieces of code that will be shared across the codebase; for example, the folder *components* contain React components that will be shared among multiple pages and other components.

- styles - Relates to the global styles.

The distinct separation and organization improve the navigation within the project. Some concepts present in the shared folder will be described in the following sections.

5.2.2 Implementation

This section describes the implementation of the dashboard and the decision made.

5.2.2.1 Authentication

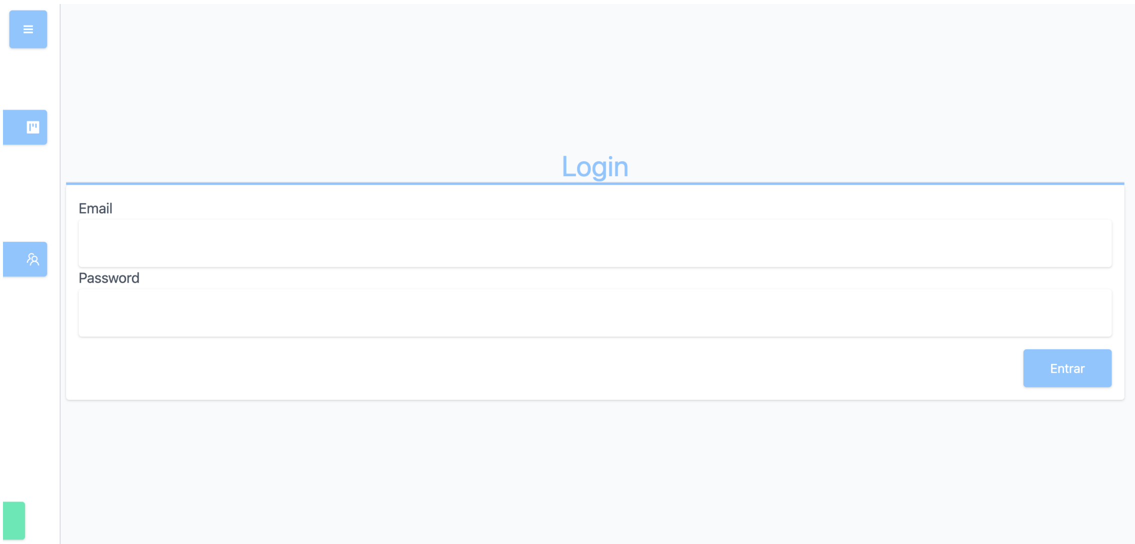

Figure 5.4: VitalCovid Login Page.

To maintain confidentiality, only authenticated clinicians can use the dashboard. A login page was created for the clinician to enter the required credentials. When the clinician submits the login form, the following mutation is executed:

```
mutation login($email: String!, $password: String!) {
  login(email: $email, password: $password) {
    token
  }
}
```

The mutation receives two arguments to be validated in the backend, email and password, and if the login is successful, it returns a token. This token is a JSON Web Token (JWT). A JWT is a secure way to send information between two entities as a JSON object, it can be verified and trusted [34]. The token is signed in the backend and sent to the frontend. When the token is received, it is stored in local storage. This token is also required to execute some queries since some are protected. The token is sent in the request headers to the backend in every request, where a middleware validates it.

Since some routes should be protected from unauthenticated users, a Higher-Order Component was implemented to do just that. A HOC is a function that receives as an argument a component and returns a new one. It is a technique for reusing component logic [35].

```
const withAuth = (WrappedComponent: any) => {
  return (props: any) => {
    const router = useRouter();
    const { data, refetch } = useQuery<IClinicianRequest>(GET_ME);
    const setUser = useAuthStore((state) => state.setUser);

    useEffect(() => {
      const verify = async () => {
        const accessToken = localStorage.getItem("auth-token");

        if (!accessToken) {
          router.push("/clinician/login");
        } else {
          refetch();

          if (data) {
            if (data.getMe.success) {
              setUser(data.getMe.clinician);
            } else {
              router.push("/clinician/login");
            }
          }
        }
      };
      verify();
    }, [data]);

    return <WrappedComponent {...props} />;
  };
};

export default withAuth;
```

This HOC executes a query to validate the token in the backend. If the token is validated, the user can access the page; else, the user is redirected to the login page. Finally, if the user chooses to log out, the token is deleted from the local storage.

A state-management solution was adopted to avoid Prop Drilling [36]. Since a global state is not required, a valid alternative was to use Zustand, a small, fast and scaleable state-management solution. Some components need some data regarding the clinician. If the login is successful, the clinician's information is saved in a store to be consumed by other components.

5.2.2.2 Landing Page

After a successful login, the clinician gets redirect to the landing page as seen in figure 5.5.

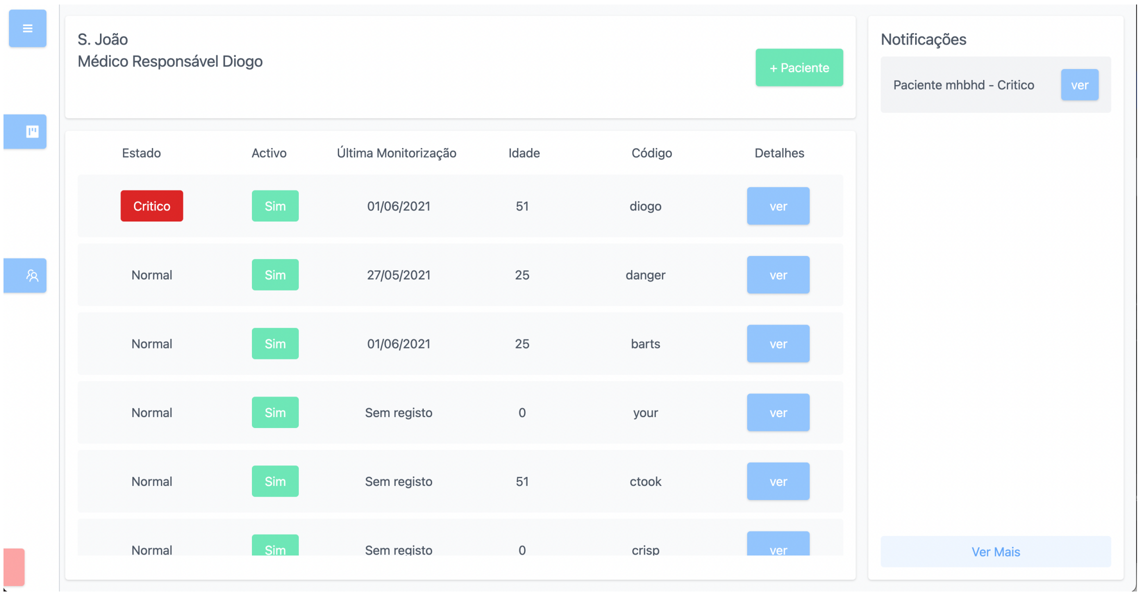

Figure 5.5: VitalCovid Landing Page.

On this page, the clinician has an overview of their activity. The clinician has a list of all the patients he is accountable for and a notifications section to quickly look at changes regarding the patient's health status. The list gives an overview of the most important data regarding the patients. It is possible to check the patients' health status, if they are under observation, the last time they performed a monitoring activity, and their age. Furthermore, the notification section gives an overview of the most recent notifications. The clinician is notified whenever the health status of one of his patients becomes critical. Each notification shows the patient's code. The notification is created when a patient performs a monitoring activity, further details in section 5.4.2.1. The clinician can click the "*ver*" button to inspect the patient's profile page.

To list the patients, the following query is executed:

```
query findManyPatients($clinicianId: Int!) {
  findManyPatient(where: { clinicianId: { equals: $clinicianId } }) {
    id
    birthdate
    status
    uuid
    activeMonitoring
    user {
      username
    }
    Form(take: 1, orderBy: { createdAt: desc }) {
      createdAt
```

```
        }
      }
    }
```

The *clinicianId* is fetched from the useAuthStore and passed as an argument to the query. In GraphQL, it is possible to request the data precisely needed, making the requests more efficient, especially on slow network connections, as might be the case in a health center.

5.2.2.3 Profile Page

The patient profile is where the clinician has access to the results of the monitoring events.

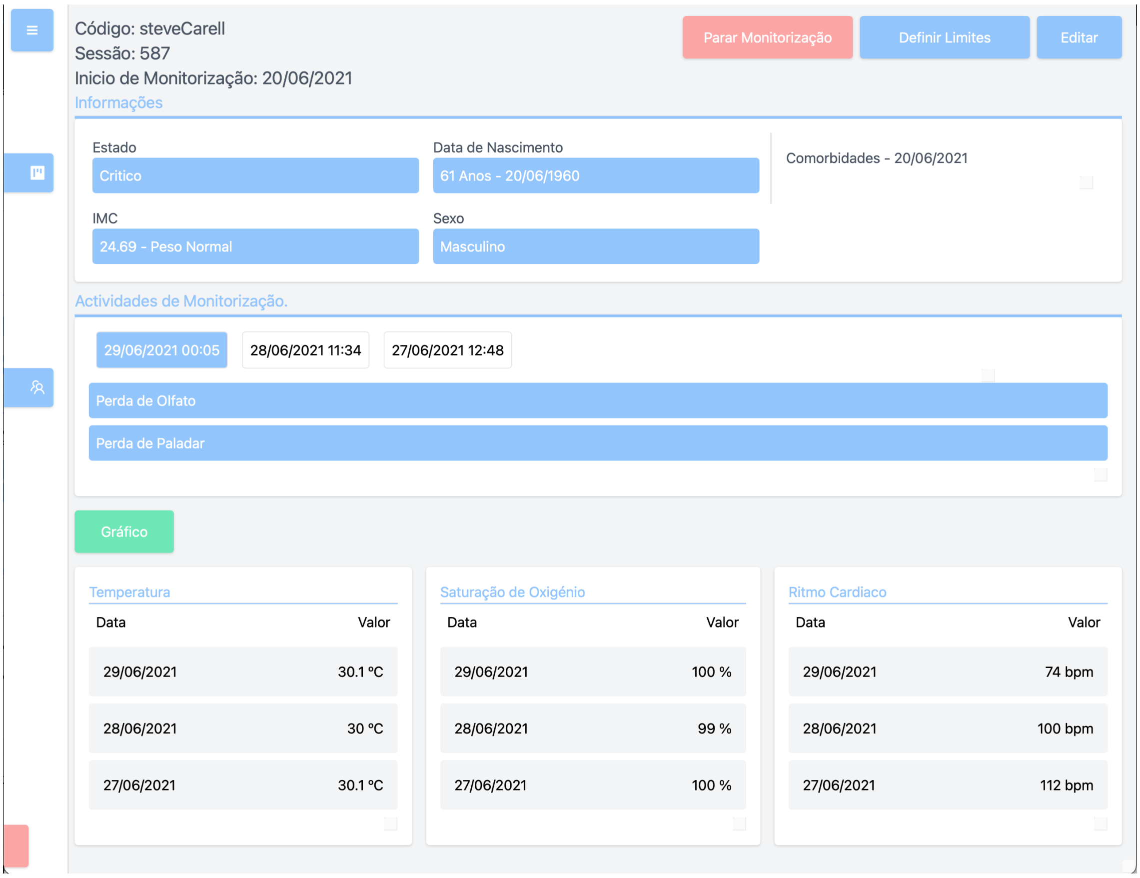

Figure 5.6: Patient Profile Page

The page is divided into two main sections. The first section has some basic static information about the patient and their conditions. The second section concerns the monitoring activities.

The monitoring activities section is divided into two main areas, the assessment form area, and the vital sign area. The assessment form area displays the forms filled by the patient over the monitoring period. The vital sign section shows the patient's vital sign values collected throughout the monitoring section. The clinician can decide between viewing the vitals on a list or a chart.

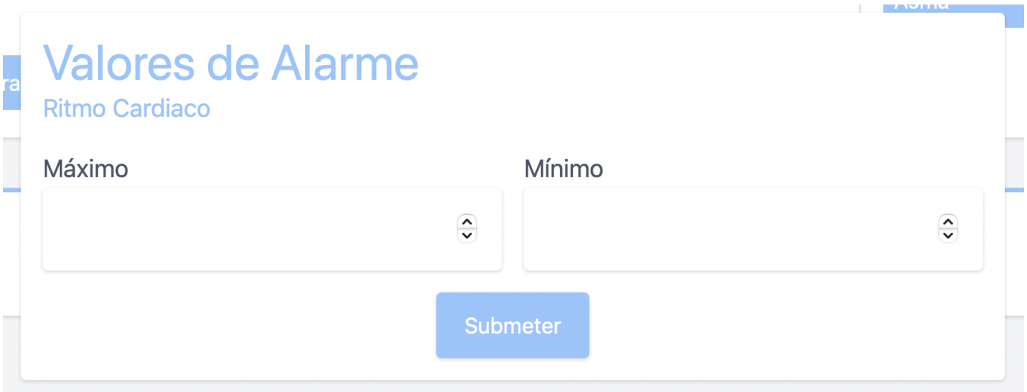

Figure 5.7: Thresholds Modal

The clinician can also use a modal to select the patient's heart rate threshold values on this page. If those values are surpassed, the clinician is notified.

Each patient has a UUID associated with preventing brute force attackers from cycling through every patient profile. That UUID is used in the URL to identify the patient. The patient's profile URL looks as follows:

- https://[domain]/patient/[UUID]

5.2.2.4 Patients Page

This page has a list of all the COVID-19 patients registered in the health center. The clinician can access this list and search for any patient by username. This list, just like the list on the landing page, can be ordered by any one of the headers.

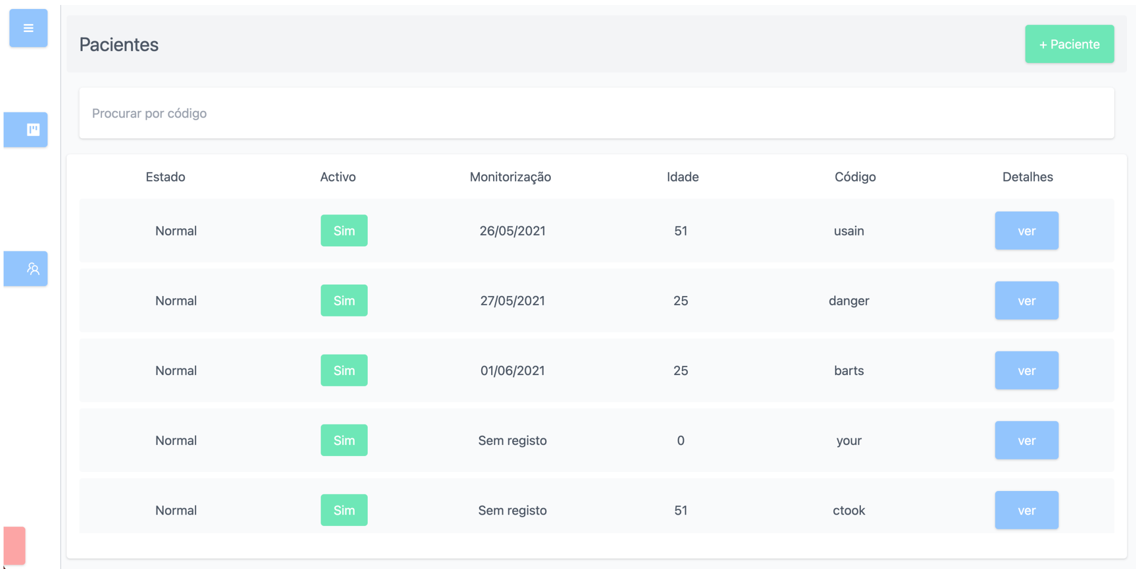

Figure 5.8: Patients Page

5.2.2.5 Patient Registration Page

The patient registration process is composed of two pages. The first page is where the clinician enters some basic patient information, and the second page is where the clinician selects the co-morbidities the patient might have.

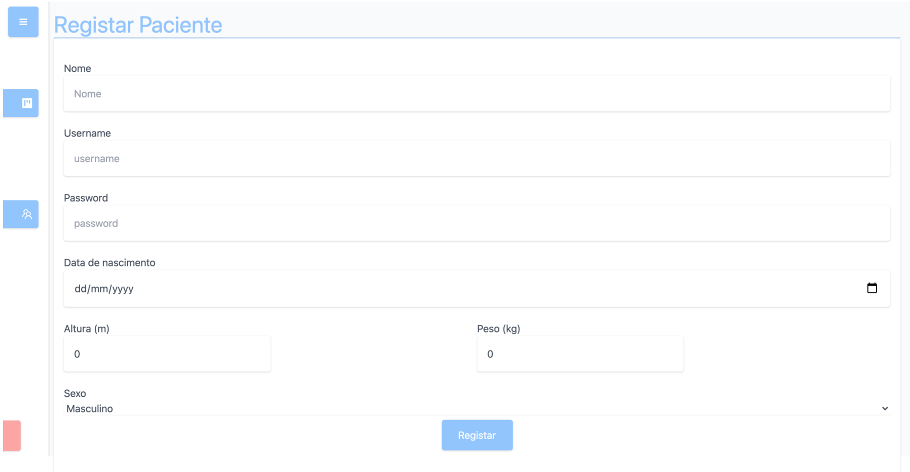

Figure 5.9: Register Patient Page

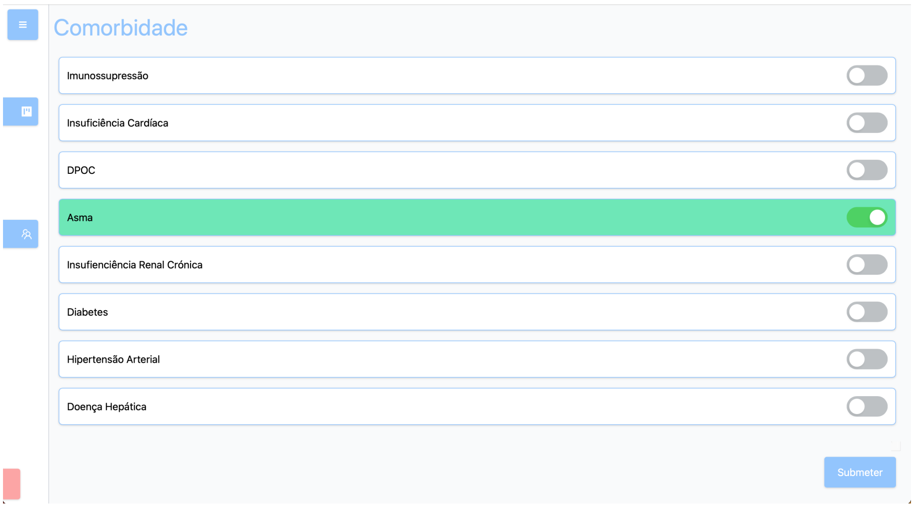

Figure 5.10: Comorbidities Page

5.2.2.6 Admin Login

Admin accounts will be given to the health centers. The admins have to authenticate to start registering clinicians. After successful authentication, the admin gets redirected to the clinician registration page.

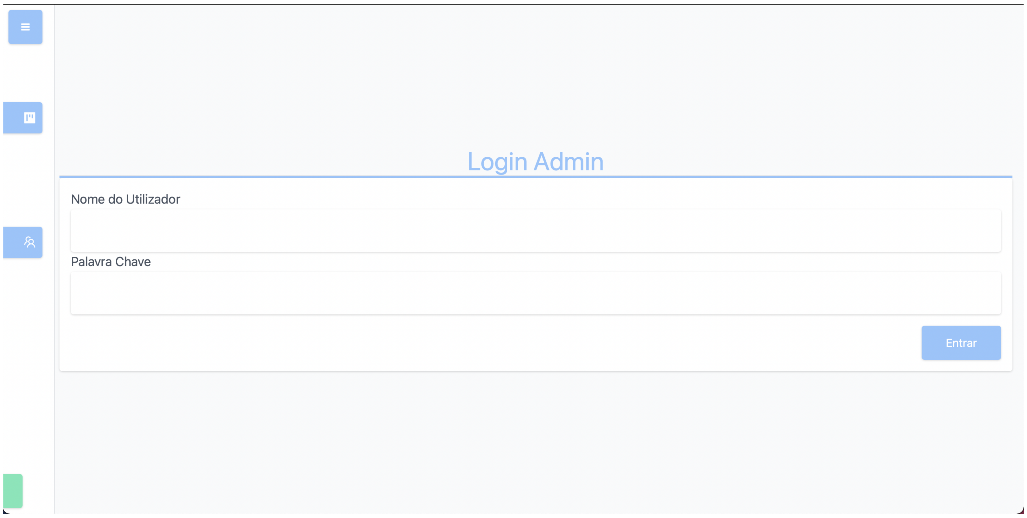

Figure 5.11: Registe Patient Page

5.2.2.7 Clinician Registration Page

To register a clinician, an admin has to enter their basic information.

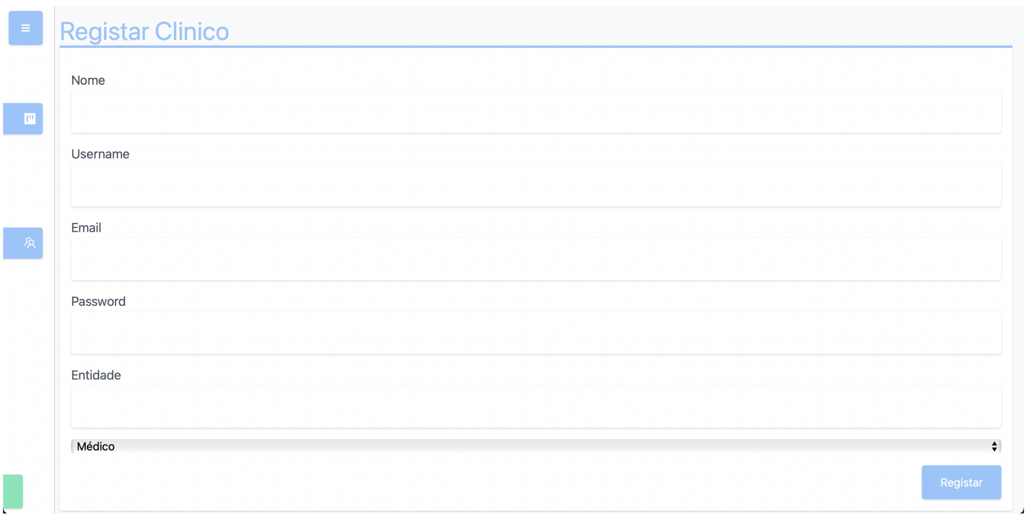

Figure 5.12: Comorbidities Page

5.3 Database

The project uses a PostgreSQL relational database due to its extensibility, standards compliance, and scalability. Prisma is used to manage the database, and it is responsible for the communication between the database and the server.

The conceptual data model of the system looks as in figure 5.13.

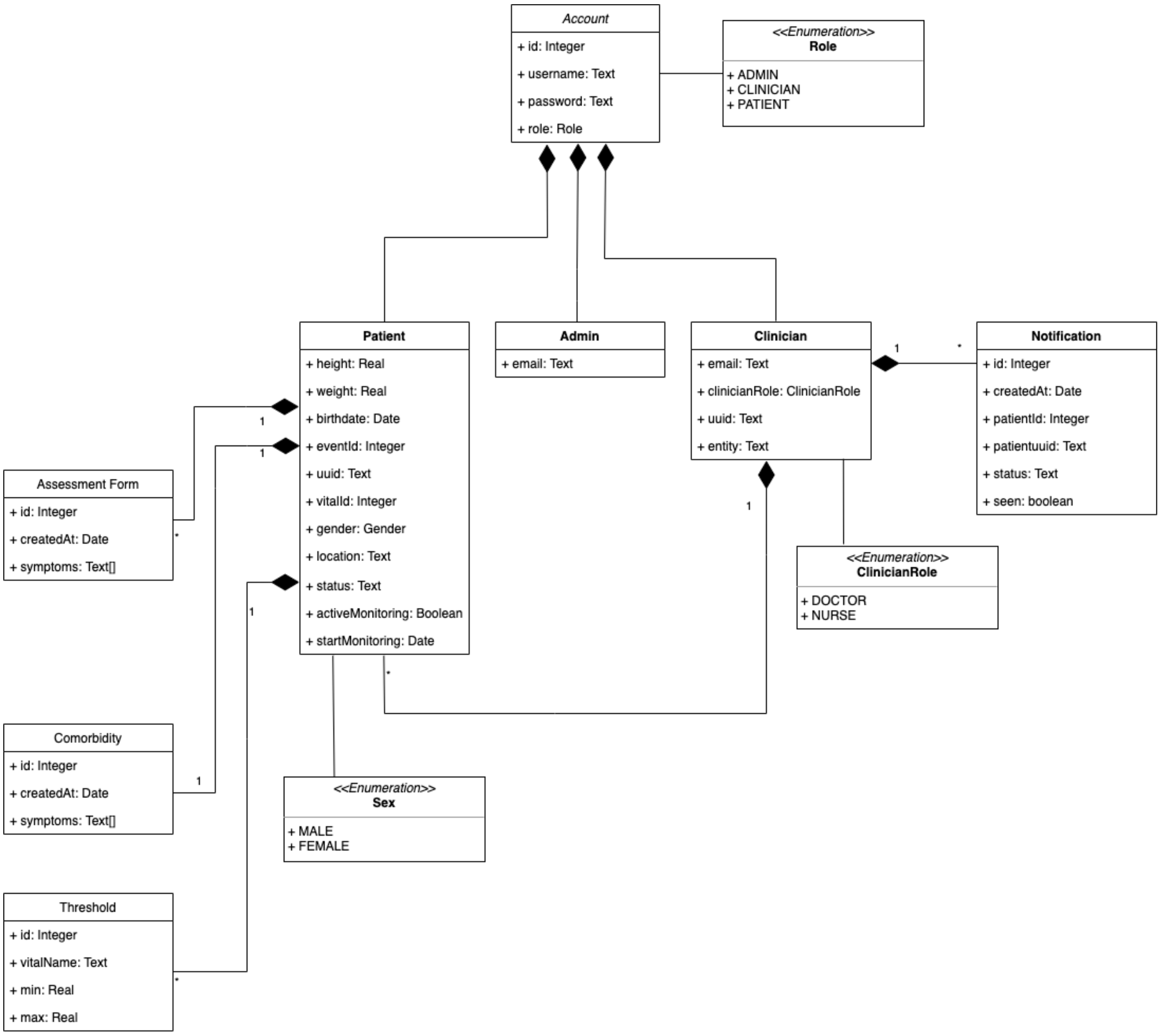

Figure 5.13: Conceptual Data Model

This diagram is an abstract model representation of the system data. It depicts the relations between the data items in the system and emphasizes how the data should be organized.

The following table describes each class in detail:

Name	Description
User	This class represents the system's users.
Clinician	This class represents the clinician. A clinician can be a Doctor or a Nurse. Zero or more patients are associated with the clinician who registered them, and the entity represents the health center where the clinician works.
Admin	This class represents the admin. This user's task is only to register clinicians.
Patient	This class represents the patient. The patient is associated with precisely one clinician. Height and weight are used to calculate the BMI, UUID is utilized to protect the system from brute force attacks. When the patient starts the monitoring session, the Date is stored for ordering purposes. There is also a boolean to verify is the patient is under observation or not. The status represents the patient health status. Finally, the eventId is used to query the vital signs from the WeSENSS backend.
Notification	One clinician can have zero or more notifications associated. Each notification holds the patient's health status and some information to connect the notification to the patient.
Comorbidity	The class lists all the comorbidities associated with a patient.
Assessment Form	The assessment form is the self-assessment form that the patient fills in when performing monitoring activities. This form lists all the symptoms the patient has at the time of completing the monitoring activity.
Threshold	The clinician can create alarm values for specific vital signs. Those values are stored in this class. Furthermore, the server uses the information in this class to verify if the vital signs are within the thresholds established by the clinician or not.

Table 5.1: Class Table.

5.4 Mobile Application

The mobile application collects the patients' symptoms and vital signs and sends them to the respective backends. The application was already built to work with the WeSENSS backend and the wearables developed by the BRAIN team. Therefore there was a need to adapt it to this new use case. There were three main requirements to be met:

Make the application able to pair with a Bluetooth oximeter to read the oxygen saturation values and send them to the WeSENSS backend with the remainder of the vital signs.

Develop an assessment form for the patients to fill it and send the results to the VitalCovid backend.

Implement a notification service where the patient receives request notifications created by the clinician to perform the monitoring activities.

5.4.1 Features added to the WeSensor application.

First and foremost, it was necessary to integrate a Bluetooth oximeter as part of the Bluetooth stack to gather oxygen saturation value. After a thorough search, we were able to find a blog article describing how to reverse engineer a Bluetooth oximeter [37]. The article explained how to parse the incoming packages and convert the incoming byte array into human-interpretable values. However, of course, the first step was to implement the service into the mobile application. According to the article, the Bluetooth oximeters from Shanghai Berry Electronic Technology use the BCI Protocol. The protocol is simple and should support at least the Berry Bluetooth oximeter BM1000 and BM2000 models. Thankfully, the article provided a Github repository with plenty of information on how to achieve that.

Firstly the application needed to be programmed for pairing with a new type of device. The application was already prepared to be paired with two kinds of devices, the *VitalStickers*, and *AmbiUnits*, so a minor code update was needed to receive an *Oximeter*. The name and type of device needed to be added to the list of trusted devices since the application only supports known devices. The oximeter service was implemented based on the implementation of the other services and adapting it to the particular device. In the VitalSticker Activity, a field was implemented to display the oxygen saturation values in real-time.

The Bluetooth pulse oximeter used was the Berry BM1000C. The measurement range is from 35% to 100%, and the uncertainty is 2%.

As a result of the interviews with clinicians, the need to collect the patients' symptoms was understood. Beforehand the application needs to communicate with the backend GraphQL API. Therefore, a GraphQL client needed to be implemented to send the assessment form data. Apollo Android is a GraphQL client that generates Java models from GraphQL queries [38]. These models give a type-safe API to work with GraphQL servers, such as the VitalCovid backend. This list should be filled by the patient during each monitoring activity and sent to the clinician to assess the patient's symptoms.

A report screen should be added to the application for the patient to enter the vital signs measurements manually. If the patient does not have Bluetooth devices at their disposal, the patient can measure the vital signs on regular devices and use this screen to report them.

The notifications service was a low-priority feature, and being that the project was quite ambitious, there was not sufficient time to implement it. However, this feature is part of the future work and will be implemented using Firebase Cloud Messaging since the application already uses Firebase.

5.4.2 Application Screens

5.4.2.1 Main Screen

The mobile application should be provided to the patient with the Bluetooth devices configured in the application's settings and the patient already authenticated. The lading screen is the one in the following figure 5.14.

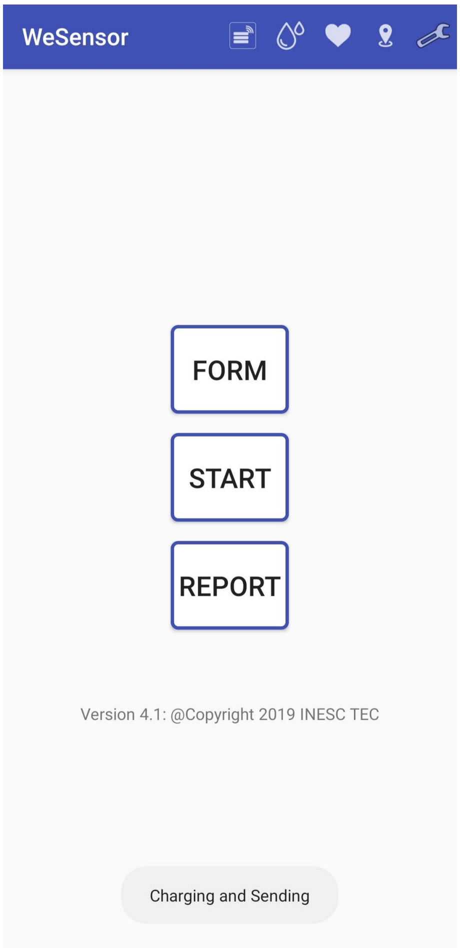

Figure 5.14: Application Landing View

The figure 5.14 represents the main view where the patient can navigate to every relevant screen. By pressing the Form button, the patient can navigate to the assessment form view.

When the patient taps the *Stop* button to stop sending the Bluetooth data, a query is executed for the backend to analyze the monitoring activity data. Suppose the average value for each vital sign is above or under the threshold defined for that specific vital sign. In that case, a notification is generated, and it will be displayed when the clinician accesses the dashboard.

5.4.2.2 Assesment Form Screen

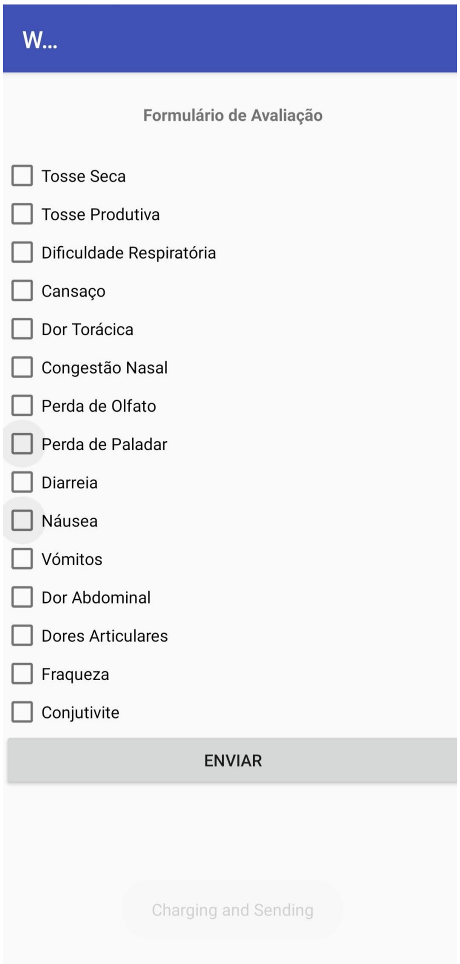

Figure 5.15: Assessment Form View

The view in figure 5.15 aims to collect the patient's symptoms and sends them to the Vital-Covid backend. The patient selects the symptoms he might have and presses the "*Enviar*" button to submit the form.

5.4.2.3 Vital Signs Screen

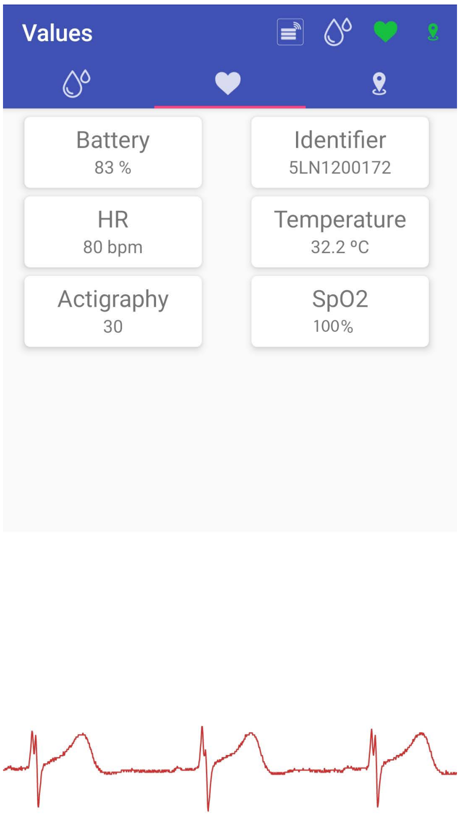

Figure 5.16: Vital Signs View

The view in figure 5.16 is dedicated to the vital signs. After the patient presses the *Start* button in figure 5.14, if the pairing with the Bluetooth devices is successful, they can see their vital signs in real-time.

The previous version of the mobile application already displayed every vital sign present in figure 5.16 but oxygen saturation. In the dashboard, the clinician has access to oxygen saturation, temperature, and heart rate. However, since the application is intended to be adapted to other contexts besides COVID-19, vital signs like actigraphy and electrocardiogram were kept for future integrations. The vital signs displayed in this view are sent to the WeSENSS backend to be processed and stored. The clinician can access the information of every monitoring event in the VitalCovid dashboard.

5.4.2.4 Vital Signs Report Screen

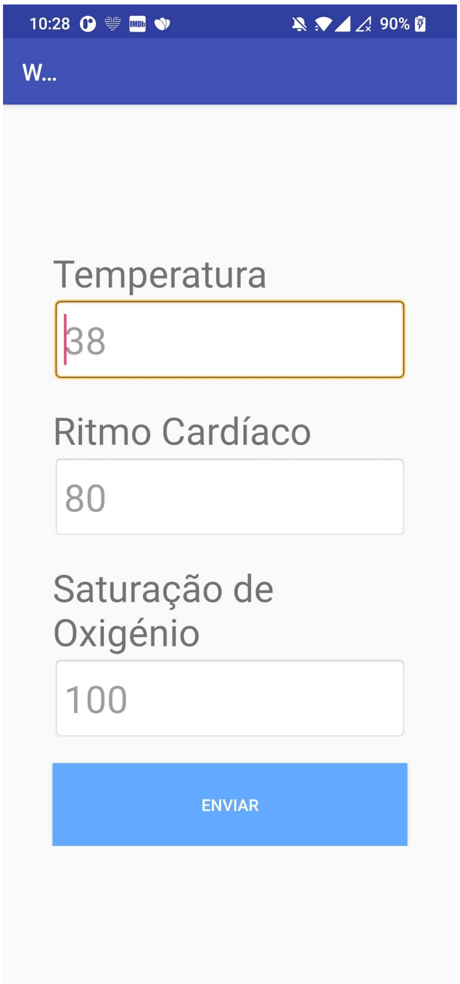

Figure 5.17: Vital Signs Report Screen

If the patients do not have the necessary Bluetooth devices to measure the vital signs, they can use regular devices and use the view in figure 5.17 to report their vitals. The patient can choose which fields to fill and press the button "*Enviar*" to submit the values, and the data is sent to the WeSENSS backend.

Chapter 6

Results

Even though VitalCovid is just a proof-of-concept, the consensus from the doctors' interviews was that this platform could significantly impact the remote monitoring of COVID-19 patients.

There was a final meeting with a doctor who worked closely with the TraceCovid platform, Dr. André Santiago. The meeting was dedicated to showing the VitalCovid system's potential and further discussing it and how it could impact the future of remote monitoring. According to Dr. André Santiago, the proof-of-concept might be able to achieve what was proposed. Even though the first and last phone calls are still needed to discharge the patient, the platform could replace the phone calls in between by providing all the necessary information. The system was regarded as very user-friendly and capable of monitoring other types of patients; the interviewed doctor even declared that he would like this tool to help him in his daily monitoring work. Lastly, Dr. André Santiago also made a few remarks regarding the features he would like to see in future system iterations. Those will be further discussed in the future work section.

Another meeting was held with two medicine students, Gonçalo Pinto and Gonçalo Neto, who worked closely with TraceCovid. The group belongs to the School of Medicine of the University of Minho. Furthermore, at the end of 2020, due to the overwhelming number of people infected with COVID-19, these students were designated to monitor assignments. Following VitalCovid's demonstration, they commented that the system is straightforward and intuitive and could lighten the doctor's workload. They said that the system could be especially useful in separating patients in critical condition from asymptomatic patients. In contrast, with TraceCovid, it is necessary to make phone calls to evaluate the patient's status since doctors do not have the information beforehand. The group's opinion was that the platform is a great proof-of-concept that could improve the overall monitoring process; they also said it could be used in other contexts, mainly monitoring chronic patients.

The following YouTube playlist can overview all features present in the system; the use cases are covered throughout the videos: https://youtu.be/YL-f0SIEdgg.

6.1 User experience evaluation

A Google Forms questionary was created for the respondents to fill at the end of each meeting to get feedback regarding the system's usability. After the meeting with Dr. André Santiago, the questionary was sent and promptly filled by the clinician later. However, since the meeting with the students was in an informal setting, the questionary was filled verbally with the assistance of the interviewer. The first section consists of two Yes or No questions(the first two) and multiple-choice questions, ranging from one to five (1-5), one meaning extremely useless and five meaning extremely useful. The second section consists of questions with open-ended questions. The questionary contained the following questions:

1. Multiple questions

    - Could VitalCovid replace the phone calls performed during the TraceCovid tasks?
    - Could VitalCovid reduce the time it takes to perform each task?
    - Is the design attractive?
    - Could VitalCovid improve the current remote monitoring process?
    - Could VitalCovid be used to monitor patients with different illnesses?
    - Would you like to have access to this platform?

2. Open-ended questions

    - What are the advantages of such a platform not only for remote monitoring of patients with COVID-19 but also for other illnesses?
    - What features should be added to the platform?

The goal of multiple questions was to assess the system usability with quantitive values to understand what could be improved regarding the user's point of view. Nonetheless, the open-ended questions were to understand how the system could be adapted to other scenarios outside of the COVID-19 context and what features could be implemented to achieve that.

In the multiple questions section, each question got the maximum grade (5 out of 5), thus indicating the respondents regarded the system as very useful and attractive.

In the open-ended questions, Dr. André Santiago commented that more and more is said about the possibility of home care. In this sense, this tool would be helpful in remote assessment and monitoring of this type of patient. Moreover, the two students noticed that human interaction, such as satisfying the patients' doubts and fears, is crucial in medicine and its code of ethics. Therefore, for the system to be widely adopted by the community, outside of exceptional pandemic scenarios, it should have some messaging or additional communication features. The questionary is displayed in the figure 6.1.

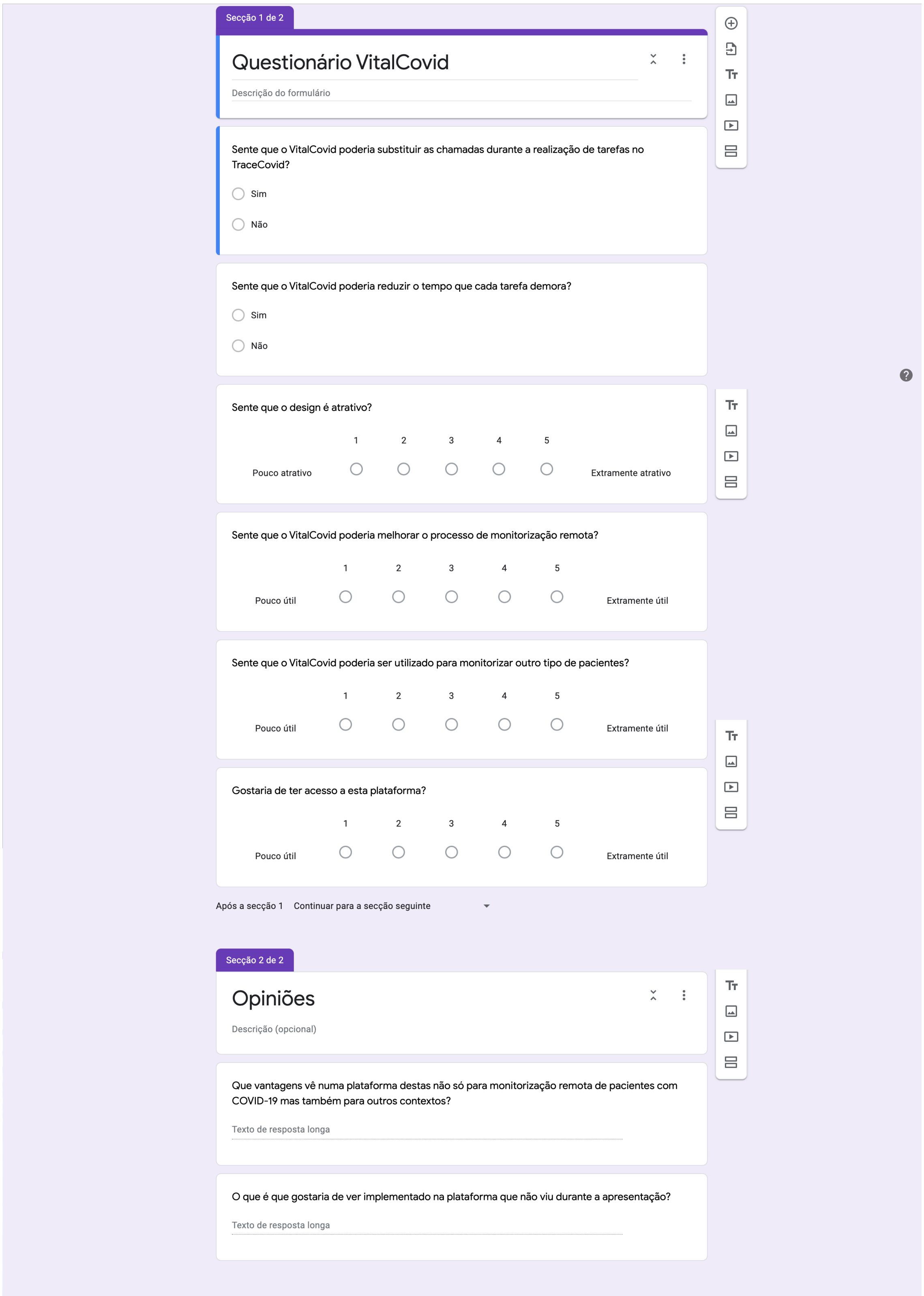

Figure 6.1: Questionary

Chapter 7

Conclusions and Future Work

The COVID-19 pandemic has generated an unprecedented need for remote patient monitoring. VitalCovid is a robust proof-of-concept and could be regarded as a starting point for further research of a remote monitoring solution that allows clinicians to monitor their patients outside the traditional hospital environment. During the making of the dissertation, a solid proof-of-concept has been developed to aid clinicians in their monitoring activities.

VitalCovid might be a great addition to the system proposed by the Portuguese Government, the TraceCovid, reducing the time of each monitoring activity since it relied on phone calls with the patients. The system can provide clinicians all the information needed to reduce or even replace these phone calls.

Even though VitalCovid was developed for the specific use case of monitoring COVID-19 patients, it is flexible and scalable. With a few upgrades, it can fit a broader range of use cases.

7.1 Future Work

Everything from the main scope was fully implemented. Nevertheless, the low-priority user stories were not implemented due to the lack of time. The features in line for future work are the following:

- Internationalization, since the platform is to be used by Portuguese patients and clinicians, only one language is available.
- Notifications Service to let the clinician remind the patient to perform the monitoring activities.
- Allow more than two Bluetooth devices to connect at once; at the moment, the mobile application can only connect to one or two devices at a time, but a third or more devices might be needed to collect other types of information the clinician might need.

- Order patients by household. In later interviews with clinicians, they reported that this feature might be helpful. Sometimes clinicians need to monitor more than one person by family, and it is faster if the patients are grouped.
- Automated responses. Clinicians also reported that patients might want to contact them to inquire about their vital signs values or other disturbances. A messaging service where clinicians can use automated responses could solve this issue.
- Report medication intake might also be a helpful feature since some medications can influence the vital signs. For example, antipyretics can reduce fever.

References

[1] Kristen Miguel, Joanna Smith, and Gill Lewin. Telehealth remote monitoring for community-dwelling older adults with chronic obstructive pulmonary disease. *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*, 19, 06 2013. doi: 10.1089/tmj.2012.0244.

[2] J. R. Barr, D. D'Auria, and F. Persia. Telemedicine, homecare in the era of covid-19 beyond. In *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*, pages 48–51, 2020. doi: 10.1109/AI4I49448.2020.00017.

[3] J. P. Cunha, S. Rodrigues, D. Dias, Pedro Brandão, A. Aguiar, I. Oliveira, J. M. Fernandes, Catarina Maia, Ana Rita Tedim, A. Barros, Orangel Azuaje, E. Soares, and F. Torre. Vital-responder®: wearable wireless platform for vitals and body-area environment monitoring of first response teams. 2019.

[4] WHO. Covid-19 public health emergency of international concern (pheic) global research and innovation forum. https://www.who.int/publications/m/item/covid-19-public-health-emergency-of-international-concern-(pheic)-global-research-and-innovation-forum, Last accessed on 2021-22-06.

[5] What is telehealth? what is remote patient monitoring? how are they different? https://news.careinnovations.com/blog/what-is-telehealth-what-is-remote-patient-monitoring-how-are-they-different Last accessed on 2021-21-06.

[6] Carlo Massaroni, Andrea Nicolò, Emiliano Schena, and Massimo Sacchetti. Remote respiratory monitoring in the time of covid-19. *Frontiers in Physiology*, 11:635, 2020. ISSN 1664-042X. doi: 10.3389/fphys.2020.00635. URL https://www.frontiersin.org/article/10.3389/fphys.2020.00635.

[7] Wendy Glauser. Proposed protocol to keep COVID-19 out of hospitals. *CMAJ*, 192(10), 2020. ISSN 1488-2329. doi: https://doi.org/10.1503/cmaj.1095852.

[8] Duarte Dias and João Paulo Silva Cunha. Wearable health devices—vital sign monitoring, systems and technologies. *Sensors*, 18(8), 2018. ISSN 1424-8220. doi: 10.3390/s18082414. URL https://www.mdpi.com/1424-8220/18/8/2414.

[9] Hoa Nguyen, Farhaan Mirza, Muhammad Asif Naeem, and Minh Nguyen. A review on iot healthcare monitoring applications and a vision for transforming sensor data into real-time clinical feedback. pages 257–262, 04 2017. doi: 10.1109/CSCWD.2017.8066704.

[10] Abhishek Khanna and Sanmeet Kaur. Evolution of internet of things (iot) and its significant impact in the field of precision agriculture. *Computers and Electronics in Agriculture*, 157:218 – 231, 2019. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.

2018.12.039. URL http://www.sciencedirect.com/science/article/pii/S0168169918316417.

[11] Joachim A Behar, Chengyu Liu, Kevin Kotzen, Kenta Tsutsui, Valentina D A Corino, Janmajay Singh, Marco A F Pimentel, Philip Warrick, Sebastian Zaunseder, Fernando Andreotti, and et al. Remote health diagnosis and monitoring in the time of covid-19. *Physiological Measurement*, 41(10):10TR01, Nov 2020. ISSN 1361-6579. doi: 10.1088/1361-6579/abba0a. URL http://dx.doi.org/10.1088/1361-6579/abba0a.

[12] World Health Organisation. Coronavirus, 2021. https://www.who.int/health-topics/coronavirus#tab=tab_1, Last accessed on 2021-25-01.

[13] CDC. Interim clinical guidance for management of patients with confirmed coronavirus disease, 2021. https://www.cdc.gov/coronavirus/2019-ncov/hcp/clinical-guidance-management-patients.html, Last accessed on 2021-25-01.

[14] Topol E. J. Oran, D. P. The proportion of sars-cov-2 infections that are asymptomatic. *Annals of Internal Medicine*. doi: 10.7326/M20-6976. URL https://doi.org/10.7326/M20-6976. PMID: 33481642.

[15] Sultan H. Almotiri, Murtaza A. Khan, and Mohammed A. Alghamdi. Mobile health (m-health) system in the context of iot. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 39–42, 2016. doi: 10.1109/W-FiCloud.2016.24.

[16] Craig Kuziemsky, Anthony Maeder, Oommen John, Shashi Gogia, Arindam Basu, Sushil Meher, and Márcia Ito. Role of artificial intelligence within the telehealth domain: Official 2019 yearbook contribution by the members of imia telehealth working group. *Yearbook of Medical Informatics*, 28, 04 2019. doi: 10.1055/s-0039-1677897.

[17] Hui Xu, Sufang Huang, Chun Qiu, Shangkun Liu, Juan Deng, Bo Jiao, Xi Tan, Ling Ai, Yaru Xiao, Mirko Belliato, and Li Yan. Monitoring and management of home-quarantined patients with covid-19 using a wechat-based telemedicine system: Retrospective cohort study. *J Med Internet Res*, 22(7):e19514, Jul 2020. ISSN 1438-8871. doi: 10.2196/19514. URL https://www.jmir.org/2020/7/e19514.

[18] How wechat became china's app for everything, 2017. https://www.fastcompany.com/3065255/china-wechat-tencent-red-envelopes-and-social-money, Last accessed on 2021-05-13.

[19] Massimiliano Donati, Alessio Celli, Alessio Ruiu, Sergio Saponara, and Luca Fanucci. A telemedicine service system exploiting bt/ble wireless sensors for remote management of chronic patients. *Technologies*, 7(1), 2019. ISSN 2227-7080. doi: 10.3390/technologies7010013. URL https://www.mdpi.com/2227-7080/7/1/13.

[20] SNS. O percurso da telessaúde em portugal, 2017. http://www.cnts.min-saude.pt/2017/03/28/211/, Last accessed on 2021-25-01.

[21] Afonso Raposo, Luis Marques, Rafael Correia, Francisco Melo, João Valente, Telmo Pereira, Luis Brás Rosário, Filipe Froes, João Sanches, and Hugo Plácido da Silva. e-covig: A novel mhealth system for remote monitoring of symptoms in covid-19. *Sensors*, 21(10), 2021. ISSN 1424-8220. doi: 10.3390/s21103397. URL https://www.mdpi.com/1424-8220/21/10/3397.

[22] University Health Network. ‘we needed to find a different way to protect our patients and staff’, 2020. https://www.uhn.ca/corporate/News/Pages/We_needed_to_find_a_different_way_to_protect_our_patients_and_staff.aspx, Last accessed on 2021-25-01.

[23] Erin Lyons. John radcliffe coronavirus patients monitored with wearable tech, 2020. https://www.oxfordmail.co.uk/news/18361997.john-radcliffe-coronavirus-patients-monitored-wearable-tech/, Last accessed on 2021-26-01.

[24] Justine Alford. Wearable sensor trialled for remote covid-19 monitoring, 2020. https://www.imperial.ac.uk/news/196973/wearable-sensor-trialled-remote-covid-19-monitoring/, Last accessed on 2021-26-01.

[25] Anand Kumar, Kesav Vithal Nori, Swaminathan Natarajan, and Doji Samson Lokku. Chapter 10 - value matrix: From value to quality and architecture. In Ivan Mistrik, Rami Bahsoon, Rick Kazman, and Yuanyuan Zhang, editors, *Economics-Driven Software Architecture*, pages 205–240. Morgan Kaufmann, Boston, 2014. ISBN 978-0-12-410464-8. doi: https://doi.org/10.1016/B978-0-12-410464-8.00010-6. URL https://www.sciencedirect.com/science/article/pii/B9780124104648000106.

[26] To type or not to type: quantifying detectable bugs in javascript. https://blog.acolyer.org/2017/09/19/to-type-or-not-to-type-quantifying-detectable-bugs-in-javascript/, Last accessed on 2021-08-6.

[27] What is an orm, how does it work, and how should i use one? https://stackoverflow.com/questions/1279613/what-is-an-orm-how-does-it-work-and-how-should-i-use-one, Last accessed on 2021-08-6.

[28] Hulu case study, . https://nextjs.org/case-studies/hulu, Last accessed on 2021-05-6.

[29] How next.js aims to simplify front-end development, . https://www.techrepublic.com/article/how-next-js-aims-to-simplify-front-end-development/, Last accessed on 2021-05-6.

[30] The history of react.js on a timeline, . https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/, Last accessed on 2021-05-6.

[31] Pages, . https://nextjs.org/docs/basic-features/pages, Last accessed on 2021-05-6.

[32] Getting started with react. https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started, Last accessed on 2021-06-6.

[33] Hulu case study, . https://nextjs.org/docs/basic-features/static-file-serving, Last accessed on 2021-05-6.

[34] Introduction to json web tokens. https://jwt.io/introduction, Last accessed on 2021-05-6.

[35] High-order components. https://reactjs.org/docs/higher-order-components.html, Last accessed on 2021-05-6.

[36] Prop drilling. https://kentcdodds.com/blog/prop-drilling, Last accessed on 2021-05-6.

[37] Reverse engineering a bluetooth low energy oximeter. https://thejeshgn.com/2020/08/05/reverse-engineering-a-bluetooth-low-energy-oximeter/, Last accessed on 2021-09-6.

[38] Apollo android. https://www.apollographql.com/docs/android/#apollo-android, Last accessed on 2021-09-6.