electronics

MDPI

Article

Identification and Visualization of a Patient's Medical Record via Mobile Devices without an Internet Connection

Sergio Laso [1,*], Daniel Flores-Martin [2], Juan Luis Herrera [2], Jaime Galán-Jiménez [2] and Javier Berrocal [2]

[1] Global Process and Product Improvement S.L., 10003 Cáceres, Spain
[2] Departamento de Ingeniería de Sistemas Informáticos y Telemáticos, Universidad de Extremadura, 10003 Cáceres, Spain
* Correspondence: slasom@unex.es

Abstract: Nowadays, people's medical records are crucial when it comes to providing treatments, discovering pathologies, or keeping track of health status. Advances in technology have allowed these records to be increasingly digitized, to the point that they can be consulted by specialists from anywhere. This also allows people to report their health status, allergies, or treatments. However, knowing a person's medical history is a delicate and complex process, and accessing these digitized data is not always possible due to problems with network connectivity or physical communication with the person. In this work, we propose a solution for medical staff to obtain a patient's medical history through facial recognition, and without the need for an internet connection. This proposal is based on the development of an architecture that allows connecting nearby mobile devices of doctors and patients without an internet connection to obtain the desired information using facial recognition as an authentication method. The architecture has been validated with the development of a mobile application that, by focusing on the patient's face with the camera of the doctor's mobile device, makes it possible to obtain the patient's medical information. With this proposal, it is possible to obtain the medical history of a person in dangerous situations or where the connectivity is limited or non-existent, in a simple and fast way.

Keywords: mobile application; nearby connection; facial recognition; personal healthcare

Citation: Laso, S.; Flores-Martin, D.; Herrera, J.L.; Galan-Jiménez, J.; Berrocal, J. Identification and Visualization of a Patient's Medical Record via Mobile Devices without an Internet Connection. *Electronics* **2023**, *12*, 75. https://doi.org/10.3390/electronics12010075

Academic Editors: Fernando Reinaldo Ribeiro, José Metrôlho and Rogério Dionísio

Received: 30 November 2022
Revised: 16 December 2022
Accepted: 19 December 2022
Published: 25 December 2022

1. Introduction

In recent years, medicine has been advancing toward telemedicine and artificial-intelligence-assisted healthcare [1,2]. In environments such as the internet of medical things, it is not unusual to find applications that make use of different wearable medical devices that track health-related metrics, such as heartbeat or blood pressure [2]. These applications process these data to manage and control the user's health, enabling users to care for their own health and allowing them to know their health status better. Nonetheless, the information generated through the use of these applications is not only big in size, but also very sensitive in nature, and data-protection regulations, such as the EU GDPR consider that biomedical data must be especially protected [3]. Therefore, patients are required to give explicit consent to allow others, such as doctors or healthcare institutions, to obtain or process these data [3].

While this requirement for explicit consent prevents the leakage of sensitive private information, it may also lead to problems in some situations. For example, in emergency situations where the patient is unable to provide consent due to his/her health status (e.g., fainting due to low sugar level or a work accident), it would not be possible to have access to the patient's health data. Such situations can be even more problematic if the emergency occurs in a location where the internet connection is of poor quality, or where there is no connection at all, as it would prevent consulting the data from the cloud, even if the consultant is provided with consent.

The problem to be solved is, therefore, twofold. On the one hand, storing a user's biomedical data exclusively in the cloud makes access to the information completely dependent on the availability and quality of an internet connection [4], which is an acceptable assumption for urban or hyperconnected environments, but may be complicated to hold in rural or natural areas [5]. On the other hand, provided that biomedical data can be accessed during an emergency, it is necessary to obtain explicit consent from the patient, which may prove complicated if they are not conscious or are otherwise unable to communicate. Current works on biomedical data access [6–10] do not focus on solving this two-faced problem.

In this paper, we propose a mobile application in which patients can provide their health data to certain medical roles through facial recognition. In the case of an emergency, a doctor previously authenticated to the system can obtain the data from the patient's phone, using facial detection and recognition of the patient, securing the information so that only the people that have consent to access the information are able to do so. Furthermore, the biomedical data are stored on the patient's mobile device, allowing them to be transmitted to the doctor that has been granted access without requiring an internet connection. The concrete contributions of this work are as follows:

- The authentication through facial recognition in emergency situations.
- The in-device storage of health information in end mobile devices.
- The access to information using direct, peer-to-peer connections without the need for internet access.

The remainder of this paper is as follows: Section 2 presents the motivations for the development of the mobile application. Its requirements, design, and evaluation are described in Section 3. Section 4 presents related works. Finally, Section 5 concludes the paper and presents some future works.

2. Motivations

Chronic diseases are affections characterized by their long duration and slow progression. Usually, they appear due to genetic, physiologic, environmental, and behavioral factors. Currently, the most common chronic diseases in the world are cardiovascular diseases (e.g., cardiac attacks and cerebrovascular accidents), cancer, chronic respiratory diseases (e.g., chronic obstructive pulmonary disease and asthma), and diabetes [11]. These diseases are especially harmful to low- and middle-income countries, where 77% of the deaths due to these diseases occur [11]. Although they are usually associated with the elderly, 15 million deaths attributed to chronic diseases are of people between 30 and 69 years old [11]. Over 85% of these *premature* deaths are in low- and middle-income countries [11]. Some key behavioral factors that contribute to the risk of suffering chronic diseases are tobacco consumption, physical inactivity, unhealthy diets, and consumption of alcohol [11].

The socioeconomic impact of these diseases is also relevant. According to the National Center for Chronic Disease Prevention and Health Promotion of the United States (NCCDPHP), 90% of the nation's USD 4.1 trillion in annual healthcare expenditures are for people with chronic and mental health conditions [12]. For example, heart disease and stroke cost the healthcare system of the USD 216 billion per year, and cause USD 147 billion in lost productivity on the job [12]. The cost of diagnosed diabetes, in medical costs and lost productivity, was estimated to be USD 327 billion in 2017 [12]. Similar socioeconomic costs exist in the US due to other chronic diseases such as obesity (USD 173 billion per year), arthritis (USD 303.5 billion per year), or Alzheimer's disease (USD 305 billion per year, projected to become USD 1.1 trillion by 2050).

In parallel to this situation of incommunicable and chronic diseases, the capacity of mobile devices has been increasing in recent years. Current mobile devices often have sensors able to measure the user's activity, such as accelerometers, gyroscope sensors, temperature sensors, or GPS-enabled location sensors [13]. Furthermore, companion devices, such as wearable smart watches or bands, are now also able to sense and track health metrics, such as the user's sleep schedule, heart rate, physical activity, or stress

level [14]. These metrics allow users to track and keep a record of their health status, a functionality that can be useful for the general public (e.g., automated tracking of the distance the user walks daily, and sleep schedule recommendations), and is especially interesting for patients with chronic or non-communicable diseases. By making use of these devices or systems, it could be possible to monitor and detect the health status of patients.

The potential of these mobile devices to help patients with chronic and non-communicable diseases has brought interest to the personal healthcare paradigm [15]. This paradigm is focused on bringing together healthcare and technological advances in mobile devices as a means to mitigate the cost of these diseases. Personal healthcare is an enabler for the prevention, anticipated diagnosis, and treatment of diseases, with a special focus on chronic diseases [15]. Through the use of personal healthcare applications, it is possible for doctors to keep a record and track the status of chronic patients without the need for constant revisions, making use of the mobile devices of the patient, for example, by communicating to multiple medical devices that the person may have and reporting the metrics they have obtained, such as measures of the blood sugar level, his/her body mass index, or his/her SpO2 [15]. Furthermore, these analyses are not only available to doctors, and the patients themselves can also obtain them to perform self-care activities. For example, some applications may be to give the patient a simplified summary of his/her health status, diagnose some diseases such as obesity, or give them recommendations to prevent or treat these conditions, such as recommending healthy diets or giving them adjusted physical activity objectives.

However, if we look at the personal healthcare paradigm from a privacy perspective, there are some key challenges that must be considered and addressed: healthcare-related information is especially sensitive, and should not be freely transmitted, stored, or processed [3]. Hence, we must determine and analyze the storage media and processing systems for personal healthcare applications, as well as whether they require data transmission. The traditional cloud computing approach, in which the doctor and the patient are given just a client or *front-end* for the application, and all the processing and storage is performed in a cloud-hosted *back-end*, is not ideal for this regard, as it requires constant data transmission between the patient's device and the cloud, and has data storage and processing in the remote data center [4]. Furthermore, even if the user explicitly consents to his/her healthcare data being processed, stored, and transmitted, the accessibility of the system would rely on the patient's internet connection, which may become an issue if an emergency occurs in a location with an unstable connection.

In contrast, if the patient's device performed the processing and storage of healthcare information, it would be maintained in a more private environment under the complete control of the patient. Furthermore, whenever a doctor would require to access the patient's data, they could explicitly give consent to such access, guaranteeing that the information is only released under this consent. To realize this vision, the people as a service (PeaaS) paradigm provides a technological foundation [16]. PeaaS proposes that users should be able to store and process their sociological profiles, which may include healthcare information, directly within their devices, ensuring they are in complete control of their information. These sociological profiles can be offered to other devices and applications as a service, reversing the roles with respect to the cloud approach: rather than having the patients' devices be clients to the cloud-hosted application services, the applications become clients that obtain their information from the patients' services [17]. The access control of the service can be directly managed by the patient, ensuring that only the information they desire is shared, and only with the people that they desire to share it with, following the data privacy regulations and allowing them to expressively limit which information is accessed [18]. PeaaS is not limited to service consumption through the internet; other opportunistic or direct connections can be used to access the services, enabling an on-premises doctor to consume the relevant healthcare information, even if there is no internet connection [16,18,19].

However, in emergency cases, users may not be in a situation where they can provide explicit consent at the moment it is requested, for instance, if they have fainted, or they are not in a condition to operate the smartphone. For these cases, it would be useful to allow the user to register some medical roles to whom they consent to access their data without a need for their interaction. In these cases, the medical personnel could have a system that could only be operated by medical professionals and would allow them to access the patient's information. However, it is likely that people are at the place of the emergency, which would require the medic's system to identify which phone it should request the information to. Due to the characteristics of these emergencies, it is desirable to determine from which phone information should be obtained in a fast, user-friendly manner, and to work without an internet connection. Facial recognition is thus a suitable method: if the recognition is also performed in the device itself, the patient can register themselves easily, and similarly, the medical personnel would only need to take a photo of the patient to identify the device where information should be accessed, aside from not requiring an internet connection. Thus, motivated by the personal healthcare and PeaaS paradigms, and the characteristics of emergency situations, we propose the nearby medical records provider (NEMREP) system to store and share a patient's medical information, directly from device to device without the need for an internet connection, using facial recognition.

3. Nearby Medical Records Provider (NEMREP) System

This section presents the NEMREP system. This system allows patients to give consent to access their information through facial detection and recognition so that, in case of emergencies, it can be consulted by any doctor. This information is stored locally on patients' devices and will be transmitted to the doctor without the need for an internet connection. The system is divided into two parts. One part is focused on the application to be used by the doctors and the other part is focused on the use by the patients. The following is a brief description of the functionalities of each part.

- **Doctor**: The Doctor section is the simplest. First, the doctor will identify himself with the authentication method of his/her health system (membership number or ID). Once logged in, it includes an option to start analyzing people. This option will open the camera of the mobile device and when it detects a patient's face, it will draw a square shape around it (facial detection), indicating some basic orientation features (position of the eyes, central point of the image with a number that will identify it and a level of happiness of the person analyzed). The application will connect with nearby devices that have the application used by the patients and if the photograph matches any of them, the doctor will be able to see their medical record and browse it.
- **Patient**: The Patient section is made up of several data entry fields, which form his/her personal and medical profiles. They will have to establish a profile photo where they will appear exclusively, which will have the goal of correct communication with the doctor when being analyzed, as a comparison will be made between the photo that the doctor analyzes and the patient's profile photo. It will have the option of entering personal data, medical data, and medical treatments that you are following. It also keeps a record of the identification number of the doctors who have analyzed them in order to detect misuse.

In this section, Section 3.1 explains the modules of which the system architecture is composed. Section 3.2 describes the workflow of the system. Section 3.3 presents an implementation of the system showing a real use case. Finally, in Section 3.4, an evaluation of different aspects of the developed system is carried out.

3.1. Architecture

This section explains the architecture of the system. On the one hand, the mobile components are those hardware components of the device that are used to perform some action (taking a photograph, entering profile data by typing on the screen, etc.). On the other hand, the application managers are the modules in charge of managing the results

obtained from the mobile components or from other managers. Figure 1 shows a general scheme of the architecture, where we can see which modules each of the parts (doctor and patient) has. Each of the modules is described below.

- **Camera**: It is the mobile phone component that is responsible for taking photographs so that the faces can be detected later.
- **Screen**: View, create and edit a user's stored or to-be-stored information via the mobile screen.
- **Communication**: This manager is responsible for the connection and exchange of data between the two parties. It can exchange image-type data (to recognize a patient's face) or a stream of bytes (the patient's history).
- **Face Detection**: It is responsible for detecting faces in the photographs taken by the camera component and applying the filter so that there is only one face in the photograph.
- **Face Recognition**: This manager is responsible for checking that the photograph of the face received from the doctor matches the patient's face.
- **Patient Data**: It is responsible for managing the storage of patient data (profile picture, personal information, medical data, etc.).

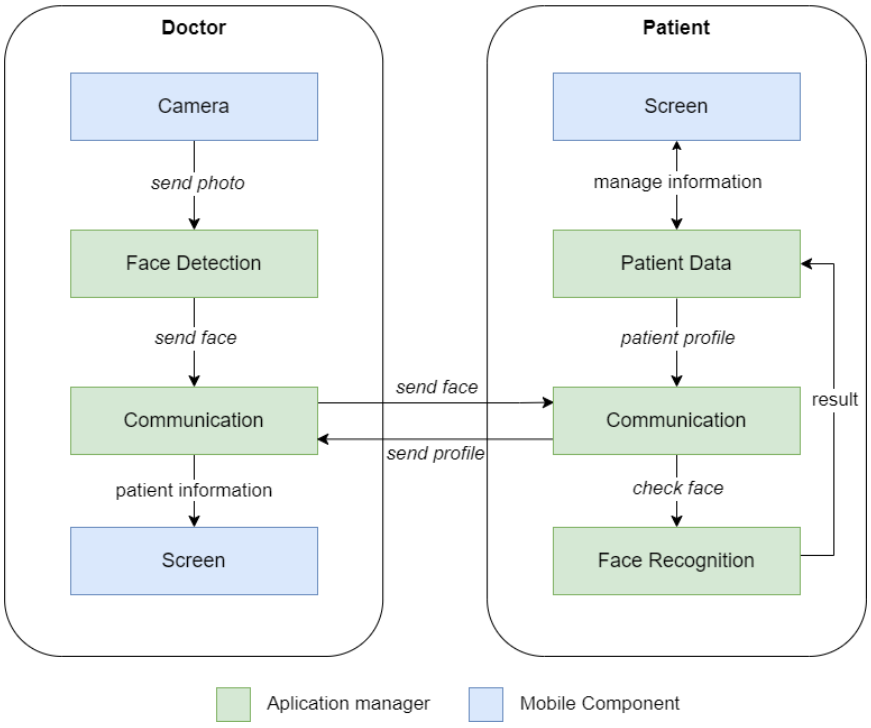


Figure 1. Architecture.

The following subsections explain in detail the functioning and the technologies used in the application managers.

3.1.1. Face Detection

The Face Detection manager is in charge of detecting the different faces that appear in a photo/video and then sends it to the face recognition manager to identify the patients. For this purpose, we used Google's ML Kit [20].

Google's ML Kit provides a framework for finding objects in photos and videos. It includes detectors, which locate and describe visual objects in images or video frames, and an event-driven API that tracks the position of those objects in the video. In our case, we used the face detection feature. In addition to facial detection in a photo, it allows for facial tracking. The camera of the mobile device itself can implement the real-time face detection of one or several faces and keep track of them. In addition, it can display various

parameters in real-time, such as the position of the eyes, the center point of the face, the level of happiness of a person, etc.

In the app, the doctor, when going to obtain a patient record, opens the camera on his/her mobile device from the app. The camera identifies and tracks any faces it detects, and the doctor is able to view them, as well as see some additional data that could help him or her, such as those mentioned above. In addition, this system allows us to know if several faces have been detected and discard the result, as the doctor only wants to consult the profile of a patient to access his/her information.

3.1.2. Face Recognition

The Face Recognition manager is in charge of analyzing and comparing the face detected by the Face Detection manager and comparing it with the faces of the closest patients through the Communication manager. For this purpose, the OpenCV dlib-android library [21] was used.

Dlib is a C++ toolkit containing machine learning algorithms and open-source tools. One of the sets of algorithms it implements is based on image processing, including tools for face detection and recognition, based on OpenCV. There is an abstraction of this library in Java, which facilitates its use in other types of projects, such as Android, and allows easy integration into them.

Concerning its working, as it is a library that works with machine learning, first a model that recognizes faces must be trained to be able to compare it with other different ones and carry out a correct recognition. This process does not require an internet connection. Moreover, the training of the model is performed locally so it does not depend on external identifiers or variables, and it exclusively uses information that is always kept in the user's device, improving the privacy of the application. Moreover, the image used to evaluate the recognition is ephemeral: it is neither stored on the doctor's device, nor in the patient's device. In terms of privacy, this implies that users aside from the patient can access, at most, an ephemeral facial image that will be automatically discarded from the device.

In the application, the patient will set up a profile picture, and it is checked if there is a face in it. If so, an internal graph is generated and trained with this image. On the other hand, the doctor, when analyzing a patient, will check that a face has been detected with the Face Detection manager and will send the analyzed image to the devices of nearby patients through the *Communication* manager. Each one of them, with the *Face Recognition* manager, will check that image with its trained graph and, in the case of the user that matches its profile picture, it will send its information to the doctor who requested it.

3.1.3. Communication

The *Communication* manager is responsible for communication from the doctor's device to the devices of close patients. For this purpose, we have chosen the Nearby [22] by Google. Nearby is a platform for discovering and communicating with nearby devices without an Internet connection. It facilitates the discovery of nearby devices and communication with them. It consists of three different APIs: Nearby Messages, Fast Pair, and Nearby Connections. Google Nearby was chosen over other protocols because of its native integration with smartphones. This facilitates both its implementation for developers and its use by users. In addition, being Google's own service, it has a high level of power optimization and the developer guide offers fairly complete documentation to cope with resource usage and thus avoid battery drainage [23].

On the one hand, Nearby Messages consist of publishing and subscribing to small messages between nearby devices connected to the internet. On the other hand, Fast Pair provides an API with facilities to enable pairing between devices via Bluetooth.

In our case, we used the remaining Nearby Connections API. Nearby Connections is a peer-to-peer network API that allows applications implementing this technology to announce, discover, connect and exchange data with nearby devices in real time, regardless of network connectivity. Examples of use are local multiplayer games, multi-screen games,

or offline file transfers. In terms of security, the Nearby protocol encrypts the information sent between the peers [22]. Hence, information sent through Nearby is protected in terms of confidentiality and integrity, due to this encryption.

The API uses a combination of Bluetooth, BLE, and Wi-Fi access points, taking advantage of the strengths of each and avoiding their respective weaknesses. This is a great advantage, as an internet connection is not necessary, as in some situations, emergencies can occur where internet connectivity is very low.

The use of the API is somewhat similar to a publish–subscriber service, where there is a discoverer or subscriber who watches for events to be published by a publisher or advertiser. In the case of Nearby Connections, there are two phases: pre-connection and post-connection.

To ensure effective communication in all situations, including emergencies, the patient's profile is monitored at all times so that if he or she is unable to use the mobile phone and is analyzed, the correct exchange of data can take place. The chosen communication system allows data to be exchanged without the need for a network connection. For this purpose, both parts/profiles of the application will have an options or configuration section of the application, where the user can select whether they want to be visible to other users or not.

The doctor, as the discoverer, will start looking for advertisers listening to receive data, which will be all the patients around him. To all of them, he will send the image he has just analyzed by the Face Detection manager, first initiating a connection and thus sending the image with Nearby Connections. This is when the patient application, internally and in the background, will compare its trained graph with the image received by the *Facial Recognition* manager. In case of the image matches, this user will send his/her profile data back to the doctor with whom he established the connection.

3.1.4. Patient Data

The Patient Data manager handles the processing of patient data to store it locally on the device. As this information is fully stored locally, the patient is in full control of this data, in terms of storage. This approach empowers the patient's privacy, as the data are exclusively stored in his/her device. To store the data, we propose the use of the Room persistence library [24], which provides an abstraction layer on top of SQLite to allow more robust access while taking advantage of all SQLite functionalities [25]. The following is a list of the different fields that the patient can include:

- Personal information that may be included: first name, surname, birth date, emergency telephone number, home address, and personal details.
- Medical information that can be included:
  - Medical history: Blood type, list of intolerances, operations, allergies, and additional medical data.
  - Medication: Information about medication for cardiovascular diseases and diabetes can be added. In addition, measurement values for both diseases can be entered and displayed as an evolution in a line graph.

Furthermore, it is important to note that, when the patient data are sent over to the doctor making use of the communication module, the doctor's device does not store the information: it is exclusively sent as an ephemeral message. This fact further improves the patient's privacy, as NEMREP allows doctors to query for patient data, but not to store it, and any query to a patient's record must be authorized through facial recognition.

3.2. Workflow

This section explains how the workflow of the application works, i.e., the flow of the normal operation of the application on both sides from the start of the application until the physician receives a patient profile. Figure 2 shows the workflow of the application and is detailed below.

- On the one hand, the doctors' application will access the main screen, where they configure their visibility while waiting to analyze a patient and establish a connection with nearby users and obtain the desired profile.
- On the other hand, the patients' application will also access the main screen to set their profile picture to be trained by the machine learning model, their personal and medical data, and also set their visibility. After meeting these requirements, they wait to find a doctor to connect with.
- When the doctor wants to obtain a patient's profile, he will access the mobile component camera through the application to take a picture of the patient's face. Once done, the picture is sent to the Face Detection manager to perform face detection. Once it detects a face correctly, it sends the image to nearby patients via the Communication manager. They check the image received matches their profile picture through the Facial Recognition manager. If the facial recognition is successful, the user obtains their information through the Patient Data manager and sends it back to the doctor with whom they have made the connection so that they can view it. Otherwise, the doctor would return to the point of taking a picture with the camera to test with another patient, because the analyzed patient does not have the application installed, does not have the connection active, or the profile picture is not well taken.

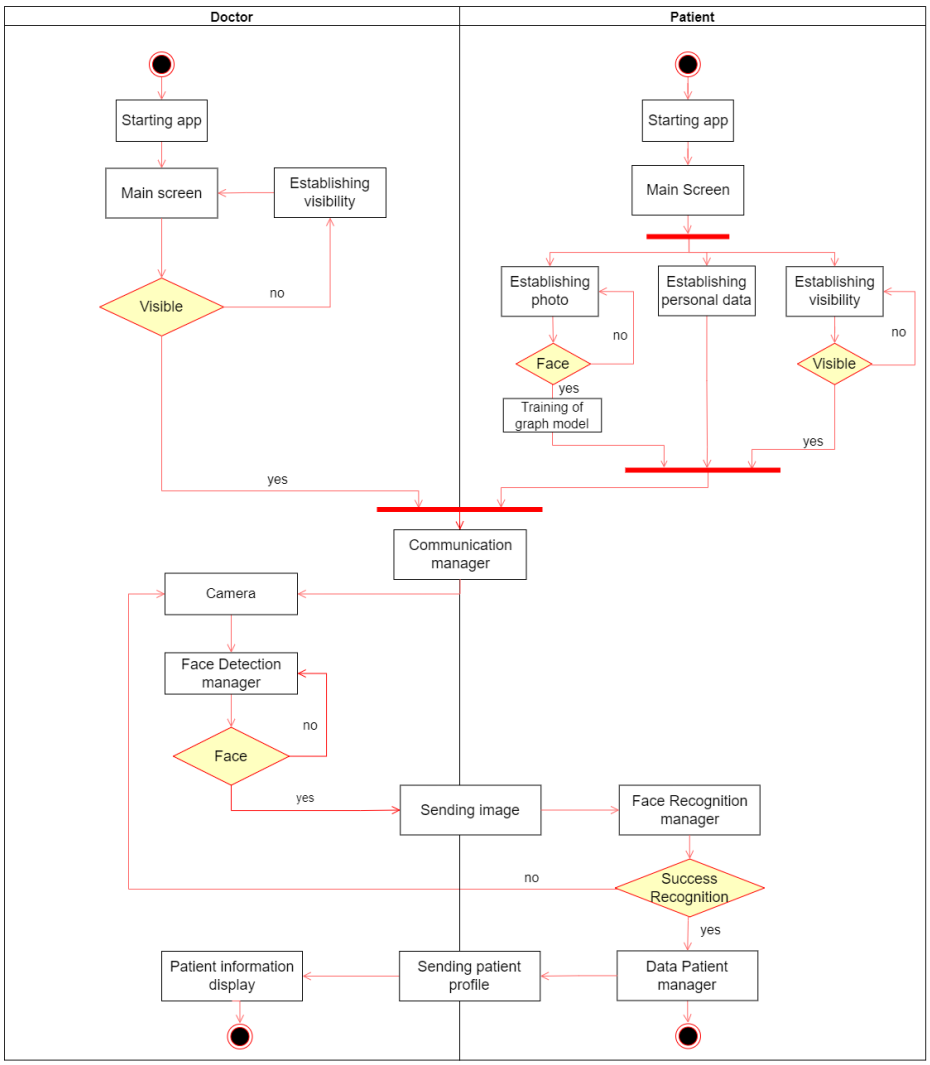
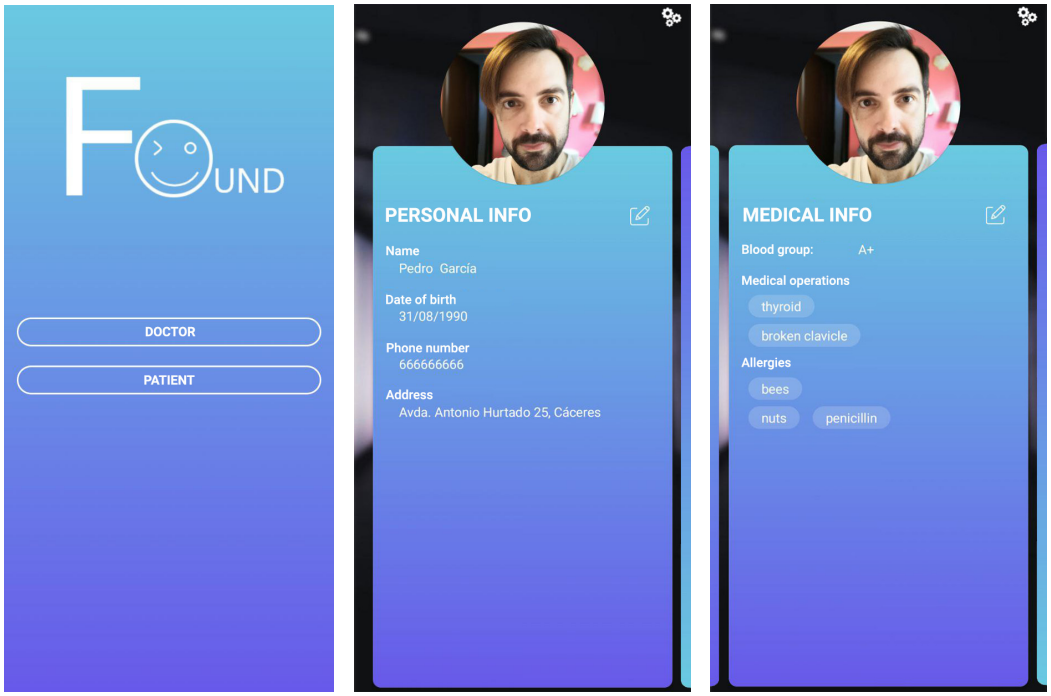


Figure 2. Workflow.

3.3. Case Study: Mountain Emergency

This section implements a case study (the images of the person were generated with AI by using DALL-E 2 (https://openai.com/dall-e-2/, accessed on 17 October 2022)) for an Android mobile application. This application is called *Found* and is based on the proposed architecture. Found can be used by both profiles: doctor and patient. Note that we have proposed this *"two-in-one"* solution to optimize resources and leverage the same application for both profiles. Figure 3a shows the initial screen of the application where each type of user can choose their profile type. Note that this is a prototype, so each user will choose the corresponding profile type.

The case study is described as follows: a group of cyclists goes on a route to the mountains, as they do every weekend. The internet connection on the route is limited and intermittent almost all the way. As a preventive measure, the cyclists have the Found application installed. In this way, in the event of any emergency and if it is necessary for the emergency services to arrive, they can provide their personal and medical details via the app, even if there is no internet connection. Previously, they had to access it and enter all their personal and medical data, including their profile picture, for correct identification. Figure 3b,c refer to the profile picture and medical information, respectively, of one of the cyclists called Pedro. By scrolling sideways they access the different sections and with the edit button, they can modify the necessary fields to enter their data.



(**a**) Initial interface (**b**) Patient personal info (**c**) Patient medical info

Figure 3. APP main interfaces.

During the route, Pedro falls off his bike, and because of this, his companions who were behind him also fall with him, causing him greater damage. Pedro is left unconscious due to the multiple blows of his companions. The companions are not seriously injured, but Pedro is, so the emergency services are called to the scene of the accident. When the emergency personnel arrive at the scene of the accident, they quickly attend to Pedro and ask his colleagues for medical information about him in order to treat him correctly. However, his colleagues do not know his allergies, medical history, treatments, etc. As Pedro is unconscious, they cannot show him their information. In addition, the unstable internet connection means that the staff cannot access the hospital's database. However, thanks to Found, the healthcare staff can access the application to obtain Pedro's medical

information and act accordingly. By opening the application with the doctor profile, the healthcare staff directly access the camera within the application, as shown in Figure 4. The healthcare worker takes a picture of Pedro's face and this is scanned by the application. Automatically, when the application detects Pedro's face, it is sent to all nearby devices (in this case Pedro's and his colleagues') that are present at that moment. The devices analyze the detected face and compare it with the one on his profile. As the image received corresponds to Pedro's profile image, it will be Pedro's device that will send his entire profile to the healthcare staff. The healthcare staff would see this information in the same way as the patient, as shown in Figure 3c, but without being able to edit it. In this way, the healthcare staff will be able to act correctly and treat him, taking into account his medical information.

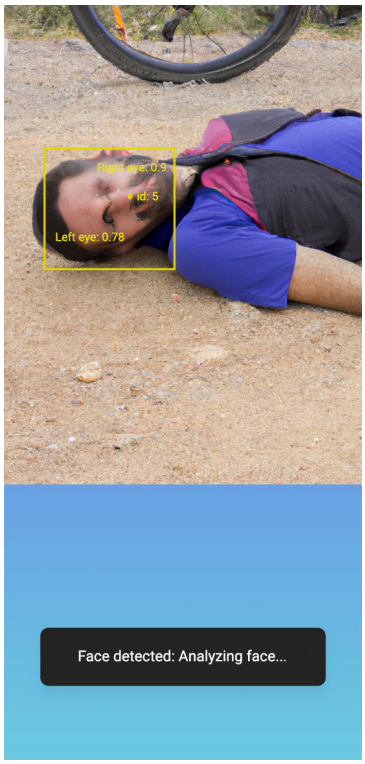


Figure 4. Doctor's interface for face detection.

3.4. Evaluation

One of the main aspects that users take into account is the resource consumption that the application will have on their device [26]. Another important aspect is the quality of experience (QoE) they offer, i.e., a user's satisfaction with the functionality provided [27]. QoE evaluates both the quality of service (QoS) [28], quantitative measures of the performance of functionality, and the user experience (UX), the interaction between the end user and the functionality. This section presents the evaluation of application resource consumption and QoS analysis, focusing on performance in the main functionality of the application, the analysis of a patient to obtain his/her personal, and medical information.

For the evaluation, we used four Android devices. These devices are a Xiaomi Mi 9, a Xiaomi Mi9T, Xiaomi Mi Mix 2, and a OnePlus 9. To determine resource consumption, battery size and memory consumption were measured. The battery consumption measurements were obtained using Batterystats (https://developer.android.com/topic/performance/power/setup-battery-historian, accessed on 19 October 2022). Batterystats is an Android framework tool that collects device battery data in the background. To analyze the battery consumption provided by Batterystats, we used another Android tool called Battery Historian (https://developer.android.com/topic/performance/power/battery-historian, accessed on 19 October 2022). To analyze memory consumption, we

used the Profiler tool (https://developer.android.com/studio/profile/network-profiler, accessed on 19 October 2022). This tool is included in the Android Studio IDE (https://developer.android.com/studio, accessed on 19 October 2022).

To evaluate the performance, we took into account the following parameters:

- Face Detection: is the time it takes to detect the face after taking the picture with the device's camera.
- Send Photo: measures the time it takes to send the photograph with the detected face to the patients' devices and is received by them.
- Face recognition and sending profile: captures the time from when patients' devices evaluate the photo received through facial recognition to check the match with their profile picture and obtain their profile to send to the doctor.

The test scenario was as follows: one of the devices acted as the doctor and the other three as the patient. The test consisted of launching 60 tests, always capturing the same face. However, we wanted to add other patient devices to simulate a real scenario where there would be several patient devices as in the case study proposed above. The duration of the test according to the results obtained from the battery consumption was about 35 min (*Foreground* parameter in Tables 1 and 2). During this time, we measured both resource consumption and performance parameters. The results obtained are presented below.

Firstly, Tables 1 and 2 show the results of the battery consumption of the doctor and the patient devices analyzed, respectively. The results of the doctor's device consumption are good, and despite having a high use of the camera (20 min) and the facial detector, we consider that the percentage of battery consumed is not very high (8.98%). The same occurs with the patient's device consumption, where the results shown are favorable (5.95%), and the stress on the device due to the use of the facial recognition algorithm and access to the database does not have a significant impact on battery consumption.

Table 1. Doctor's device battery consumption.

Application	es.spilab.unex.facetrackernear
Device estimated power use	8.98%
Foreground	10 times over 35 m 49 s 578 ms
CPU user time	24 m 12 s 237 ms
Camera	60 times for a total duration of 20 m 51 s 989 ms

Table 2. Patient's device battery consumption.

Application	es.spilab.unex.facetrackernear
Device estimated power use	5.52%
Foreground	7 times over 36 m 25 s 742 ms
CPU user time	18 m 35 s 147 ms

Secondly, Figures 5 and 6 show an extract of the memory consumption of both profiles. The x-axis shows the execution time in seconds and the y-axis, the amount of memory consumed in Mb. Figure 5 shows an excerpt of the memory consumption of the doctor device. As we can see in the figure, when the doctor's device is not performing any analysis is when the consumption band remains horizontally stable. The average consumption is around 200 Mb. However, when he performs an analysis, requiring the use of the camera and the face detector, this can be seen in the peaks shown in the figure, implying an increase in consumption up to 285 Mb. Figure 6 shows the memory consumption extract of the patient's device. When the application is waiting to receive the photograph, the memory consumption remains above 190 Mb. However, when it receives a photograph and has to run facial recognition and obtain the data to send it, the consumption increases to about

220 Mb as shown in the figure, and a "hill" is produced, which is the time frame that executes the face recognition. Concerning the memory consumption of both profiles, we can conclude that the consumption is not too high; currently, most devices on the market have at least 6 Gb of RAM, as is the case with the devices we have used. In the worst case, which is when the doctor uses the camera to take the picture and analyze the face (285 Mb), it means the consumption of only 4.75% of the RAM memory of the device, a very low percentage.

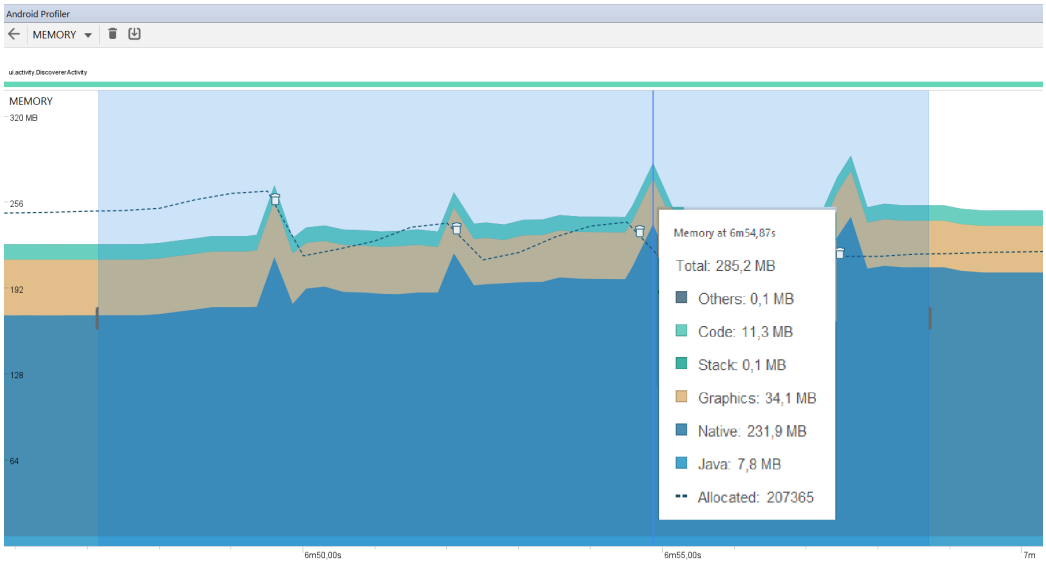


Figure 5. Doctor's device memory consumption.

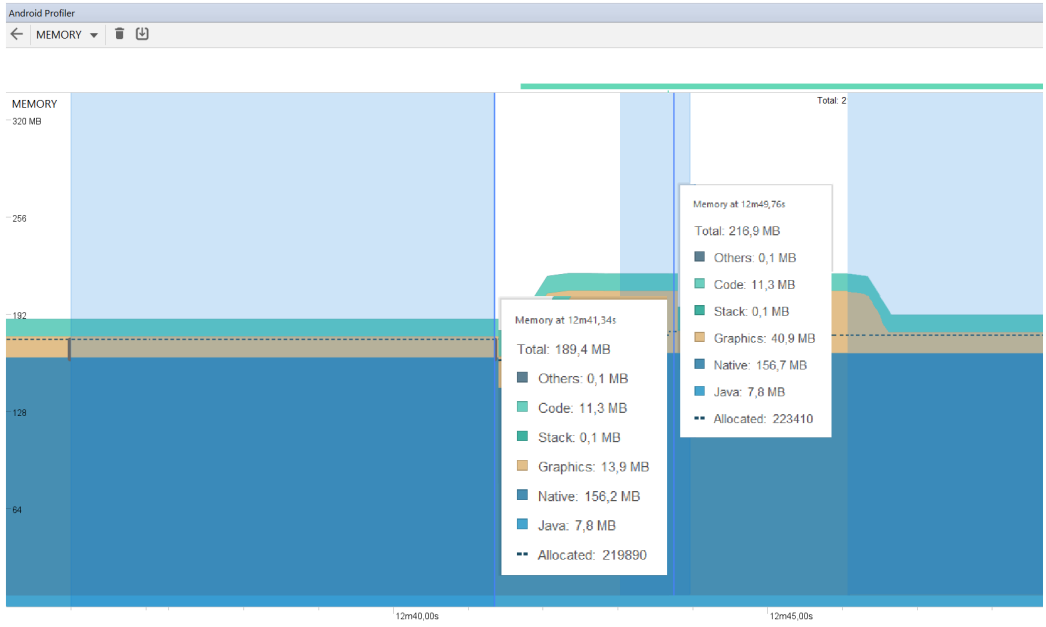


Figure 6. Patient's device memory consumption.

Thirdly, we show the results obtained from the performance parameters. Figure 7 shows a box-plot diagram with all the parameters described for the evaluation. For the parameter Send Photo, the variability is very small, obtaining values between 30/50 ms, remaining constant in all cases, and being the most stable of the three. Regarding the parameter Face Detection, it has a little more deviation (800/1800 ms) with some outliers. However, the median is in the center of the box so the distribution is symmetrical. Finally,

the parameter Facial recognition and sending profile is the process that takes by far the longest and has the most deviation, being in the range of 3400 ms and 6500 ms. Therefore, if there is a bottleneck, we would have it in the part of facial recognition and data collection.

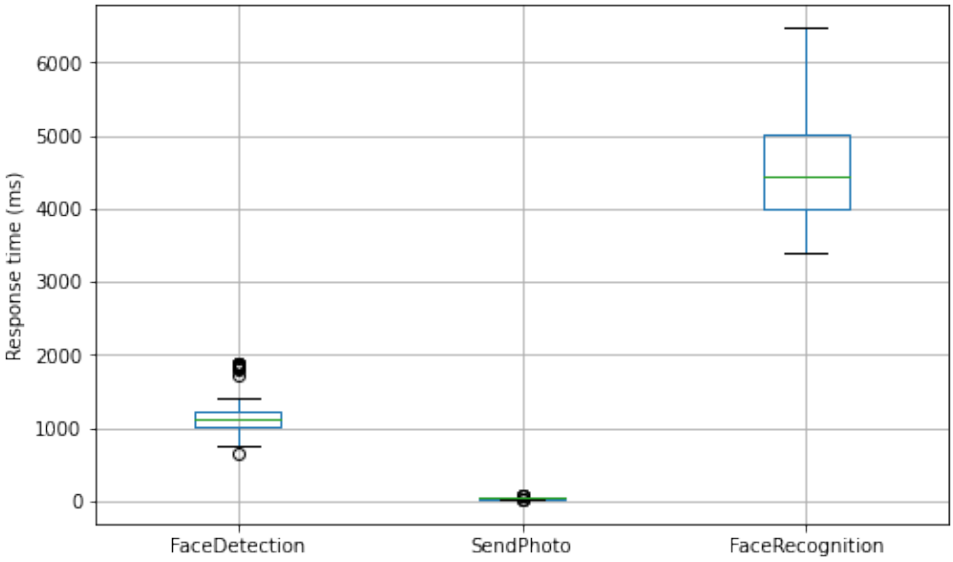


Figure 7. Performance parameters box-plot diagram.

Finally, we also show the average execution time of the entire process, from the moment the patient's photograph is taken until the patient's profile is received by the physician, i.e., the sum total of the three evaluation parameters in each execution. Figure 8 shows the box-plot diagram with this information. It can be seen that the average execution time for the 60 tests performed is between 4220 ms and 7772 ms, with an average of 5795.95 ms. Considering that the deviation caused by the main bottleneck is in the facial recognition component, the global response times suppose acceptable results, considering that these are environments where connectivity is limited and data availability is critical. Thus, it could be interesting to optimize this aspect in the future.

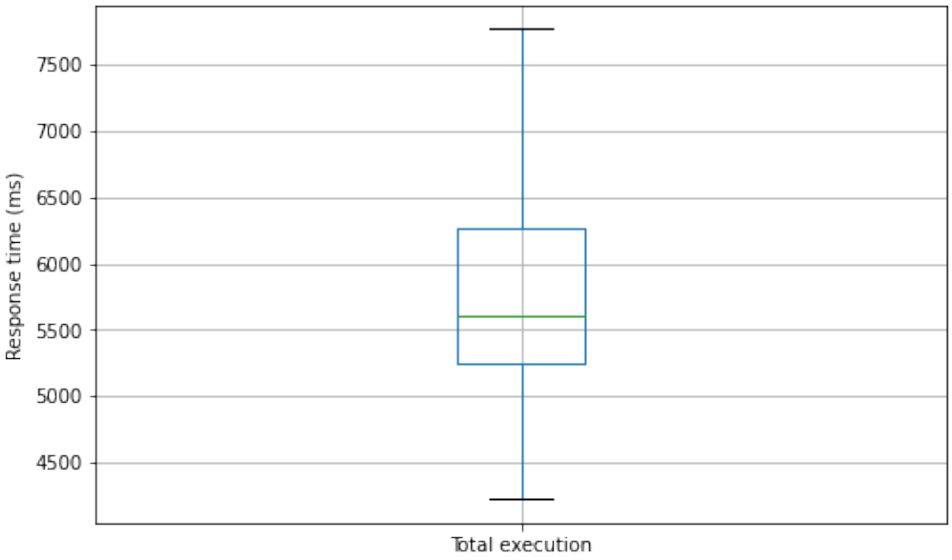


Figure 8. Total execution time.

4. Related Work

The treatment of patients' medical records has always been a sensitive issue. This is mainly because it is sensitive personal data and therefore its access and treatment entail certain risks. In addition, the digitization of this information has meant that medical institutions do not always have up-to-date patient information, or it is not consistent between different medical centers. Thus, it becomes problematic for patients and healthcare providers to keep track of medical records. Numerous solutions have tried to facilitate the visualization and sharing of this data, so that consistency of information is maintained regardless of devices or environment.

One of the most well-known solutions is health passport worldwide (HPW) [6]. HPW is presented as a digital health solution that aims to unify all clinical information on individuals. This information includes diagnoses, X-rays, scans, prescriptions, blood tests, dentistry, etc. All of this is securely unified in the cloud and can be accessed at any time by medical specialists through the invocation of a multitude of services. In addition, HPW allows users to decide what information to share and with whom, ensuring that the information will only be available to those people or providers to whom access has been authorized. The fact that its access is cloud-based guarantees its accessibility from virtually anywhere; however, in environments where the connection is limited or non-existent, the information may not be shared. That is why the solution presented in this work allows the exchange of clinical information through local protocols without the need for an internet connection.

In [7], an application is developed that allows access to a user's medical history. Users can manage their information and share it with different healthcare providers. Among their personal information such as height, weight, or body mass index, the application can record values, such as blood pressure, heart rate, or oxygen saturation. This information is stored in a cloud database so that it can be accessed from anywhere. Once again, an internet connection is essential to be able to consult and share medical records. This solution is similar to our proposal in that it considers that people are the owners of their medical history and only they can decide with whom to share it. However, the main difference is the requirement of internet connectivity for this information exchange, which in environments with limited or no connectivity would mean that the information may not be accessed.

Regarding the method of information exchange, in [8], it is proposed an open-source solution to read people's medical history through NFC tags. This proposal makes use of a mobile application to read patients' NFC tags to consult their history. This NFC tag stores all the patient's information, such as personal information (first name, last name, date of birth, etc.) and medical information (blood type, medical conditions, etc.). In addition, this proposal has the advantage that no internet connectivity is needed to read the NFC tag. However, two main differences with our proposal are detected. On the one hand, the information is stored in an NFC tag and not in the mobile device itself, which means carrying an additional device in addition to the mobile device. On the other hand, although it is specified that this tag will be read by health experts, anyone could read it and access the person's medical history. Privacy and data sharing are important issues when it comes to mobile devices. Because of COVID-19, Peng et al. [29] developed a mobile system for contact tracking via NFC that does not require special permissions. This work is interesting from a privacy point of view, by performing location through a decentralized system. Again, the power of smartphones is highlighted to execute different types of algorithms, in this case, people localization. In this sense, in [30], a decentralized blockchain-based architecture is proposed to register an authenticated data structure of user contact records. This allows for ensuring the integrity of user data by preventing users from changing their information locally on their devices. Also, in [31] a system called BlockShare based on blockchain is proposed to preserve user information. This system is based on a decentralized data system that allows verification of the exchange of data between devices in a secure way.

In addition, in [9] they bet on the security of medical data in cloud servers. The authors propose a secure method of data storage and exchange that consists of an encryption algorithm to protect data security and privacy. The main security feature used is that patients use a trusted device such as a cell phone to secure access to their data, thus also allowing them to decide with whom they are shared. In addition, the encryption algorithm can be run directly on the patients' trusted device making it easy to use. The main advantage of this work lies in the security mechanisms implemented, which are based on selective encryption algorithms in combination with data fragmentation and dispersion. One of the important differences with our proposal is that the data are stored on external servers and not locally on the patient's device, which may lead to the most frequent problems of availability and accessibility to these data in environments with limited connectivity. Additionally, thanks to the power of smartphones, these can be used to make operations even when no internet connection is available. In [32], these devices are used to preprocess images streaming video streams in order to improve the smoothness of live broadcasts. The preprocessing technique also helps to effectively reduce the delay of the live broadcast compared to approaches that do not preprocess videos.

Finally, a model for the transmission of health-related data is proposed in [10]. The authors emphasize the security and privacy of patient data, and how information confidentiality should be maintained. To describe security, IoT networks, and information exchange patterns, this proposal provides an architecture that can be adopted to realize IoT applications based on people's medical records, safeguarding their information, and using secure communication protocols. It also introduces a mobile application where patient information can be stored and can also be used to authenticate the patient in medical centers. However, the consistency of data from different sources is not specified, nor is the possibility of taking advantage of the architecture for environments with limited connectivity, which are two important aspects considered in our proposal. Additionally, in [33] addressed the shared challenges and open issues surrounding electronic health record (EHR) storage. This approach considers mobile devices integrating with cloud computing to facilitate medical data exchanges between patients and healthcare providers. The authors propose a system using the Ethereum blockchain in a real data exchange scenario in a mobile application with Amazon cloud computing. The results show that this solution provides an effective mechanism for reliable data exchange in mobile clouds while preserving sensitive health information from potential threats. However, for the system to function properly, Internet connectivity is required, which means that on lathes with limited connectivity this system cannot be properly leveraged. In addition, in [34] a solution based on W3C Thing Description is proposed for the definition of people's information. This solution uses mobile devices to store personal information so that only they are the owners of their data and decide with whom to share it. This work is interesting, first, for storing patients' medical information following a common data format so that it can be easily used by different applications and, second, for providing a mechanism to safeguard users' privacy. In addition, this work uses the MQTT protocol to exchange this information with nearby devices. Although it is an alternative to be considered, its use requires an internet connection so that in rural environments or those with limited connectivity, its use could be reduced.

These works reflect the importance of digitizing people's medical information. Security is one of the most critical parts when dealing with this kind of information, but also the way to share it becomes complex, first, because the privacy of individuals must be kept and, second, network connectivity must be taken into account to ensure its availability. Likewise, our proposal offers a solution focused on the mechanisms for storing and sharing medical information, and whose main difference with the works analyzed is that the data do not travel to third-party servers and also internet connectivity is not necessary for its transfer to other nearby devices.

5. Conclusions and Future Work

The digitization of people's medical records involves ensuring the security, integrity, and consistency of the information. It must be ensured that regardless of the environment, this history can be shared by patients and viewed by healthcare experts.

In this work, NEMREP has been presented for the identification of a patient's medical history through mobile devices and without the need for an Internet connection. The solution presented is based on a mobile application that, through facial recognition, can identify the patient and request their medical history. This is done by identifying nearby devices in the environment, so internet connectivity is not necessary. Furthermore, patients are the sole owners of their data and can decide how and with whom they share it. This is a breakthrough, especially for environments where internet connectivity is limited or patients are not available to share their records manually. It also means that all medical centers receive the same information, which is stored on the patient's device, thus ensuring consistency of information.

The application was tested to check its efficiency and that it meets the needs detected. Even so, we consider that there are some aspects to which we are devoting effort as future work. Among these aspects is the speed of patient history exchange, which although it is within acceptable values, can be improved by using more specific image detection algorithms for facial recognition. We are also considering the option of bringing the patient's history to other wearable devices, such as smart bracelets so that a mobile device does not have to be available. Another important aspect is to improve security and privacy aspects to avoid misuse by both doctors and patients. For example, limit the use of recognition or rejection by the patient if his/her vital signs are stable and have a login through identification (passport or another type of ID) for patients. We expect to improve the integrity of patient data in the future, by enforcing the data in the medical record to be digitally signed by a doctor (e.g., using a public key infrastructure [35]) or with measurements verified through remote attestation [36]. Work will continue on expanding the types of medical data entered about the patient to include data from other health-connected devices that monitor heart rate, glucose level, or blood oxygen level. This avoids the manual entry of patient data, as well as obtaining the periodically updated medical history through the appropriate security protocols of each healthcare institution. Regarding the evaluation, we are working on testing in real environments with several devices where we can get feedback from both doctors and patients. Finally, we are working on improving the user interface to make it more intuitive and user-friendly, as well as having the option of bringing the application to other operating systems, such as iOS.

Author Contributions: Conceptualization, S.L.; Methodology, D.F.-M.; Software, S.L. and J.L.H.; Validation, S.L., J.L.H. and J.B.; Formal analysis, J.G.-J.; Investigation, D.F.-M., J.L.H. and J.B.; Resources, D.F.-M.; Writing—review & editing, J.G.-J. and J.B.; Project administration, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially funded by grant DIN2020-011586, funded by MCIN/AEI/10.13039/501100011033 and by the European Union "Next GenerationEU /PRTR", by the Ministry of Science, Innovation and Universities (projects RTI20 18-094591-B-I00, TED2021-130913B-I00, PDC2022-133465-I00, and by grant FPU17/02251), by the 4IE+ project (0499-4IE-PLUS-4-E) funded by the Interreg V-A Spain-Portugal (POCTEP) 2014-2020 program, by the Regional Ministry of Economy, Science and Digital Agenda of the Regional Government of Extremadura (GR21133, IB18030) and the European Regional Development Fund, and by the Valhondo Calaff Institution.

Data Availability Statement: The data used were obtained from tests conducted with real users in a controlled environment. They are available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dwivedi, R.; Mehrotra, D.; Chandra, S. Potential of Internet of Medical Things (IoMT) applications in building a smart healthcare system: A systematic review. *J. Oral Biol. Craniofacial Res.* **2021**, *12*, 302–318. [CrossRef] [PubMed]
2. Greco, L.; Percannella, G.; Ritrovato, P.; Tortorella, F.; Vento, M. Trends in IoT based solutions for health care: Moving AI to the edge. *Pattern Recognit. Lett.* **2020**, *135*, 346–353. [CrossRef] [PubMed]
3. Union, E. Official Journal of the European Union. EUR-Lex-32016R0679-EUR-Lex. 2016. Available online: https://eur-lex.europa.eu/eli/reg/2016/679/oj (accessed on 4 October 2022)
4. Mell, P.; Grance, T. The NIST definition of cloud computing. *Cloud Comput. Gov. Backgr. Benefits Risks* **2011**, *800*, 171–173. [CrossRef]
5. Jesús-Azabal, M.; Herrera, J.L.; Laso, S.; Galán-Jiménez, J. OPPNets and rural areas: An opportunistic solution for remote communications. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 8883501. [CrossRef]
6. HPW.health. Health Passport Worldwide | HPW.health. Available online: https://www.hpw.health/ (accessed on 14 October 2022).
7. Mahmud, M.T.; Soroni, F.; Khan, M.M. Development of a Mobile Application for Patient's Medical Record and History. In Proceedings of the 2021 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 10–13 May 2021; pp. 0081–0085.
8. Martín, A.Q.; Lantada, A.D. An open source medical passport based on an Android mobile application and near-field communication. *SoftwareX* **2020**, *11*, 100492. [CrossRef]
9. Qiu, H.; Qiu, M.; Liu, M.; Memmi, G. Secure health data sharing for medical cyber-physical systems for the healthcare 4.0. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2499–2505. [CrossRef] [PubMed]
10. Kalpally, A.T.; Vijayakumar, K. Privacy and security framework for health care systems in IoT: originating at architecture through application. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–11. [CrossRef]
11. World Health Organization. Non-Communicable Diseases. 2022. Available online: https://www.who.int/en/news-room/fact-sheets/detail/noncommunicable-diseases (accessed on 14 October 2022).
12. National Center for Chronic Disease Prevention and Health Promotion. Health and Economic Costs of Chronic Diseases. 2022. Available online: https://www.cdc.gov/chronicdisease/about/costs/index.htm (accessed on 23 November 2022).
13. Flores-Martin, D.; Laso, S.; Berrocal, J.; Murillo, J.M. Contigo: Monitoring People's Activity App for Anomalies Detection. In *International Workshop on Gerontechnology*; Springer: Cham, Switzerland, 2022; pp. 3–14.
14. Xiaomi Inc. Mi Smart Band 6. 2022. Available online: https://www.mi.com/en/product/mi-smart-band-6 (accessed on 24 November 2022).
15. Traver, V.; Faubel, R. Personal health: The new paradigm to make sustainable the health care system. In Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies, Valencia, Spain, 20–23 January 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 3–11.
16. Guillen, J.; Miranda, J.; Berrocal, J.; Garcia-Alonso, J.; Murillo, J.M.; Canal, C. People as a service: A mobile-centric model for providing collective sociological profiles. *IEEE Softw.* **2013**, *31*, 48–53. [CrossRef]
17. Laso, S.; Berrocal, J.; García-Alonso, J.; Canal, C.; Manuel Murillo, J. Human microservices: A framework for turning humans into service providers. *Softw. Pract. Exp.* **2021**, *51*, 1910–1935. [CrossRef]
18. Herrera, J.L.; Chen, H.Y.; Berrocal, J.; Murillo, J.M.; Julien, C. Context-aware privacy-preserving access control for mobile computing. *Pervasive Mob. Comput.* **2022**, *87*, 101725. [CrossRef]
19. Flores-Martin, D.; Rojo, J.; Moguel, E.; Berrocal, J.; Murillo, J.M. Smart nursing homes: Self-management architecture based on iot and machine learning for rural areas. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 8874988 . [CrossRef]
20. Google. Google's ML Kit. 2022. Available online: http://dlib.net/ (accessed on 22 November 2022).
21. DLib C++ Library. 2022. Available online: https://developers.google.com/ml-kit/vision/face-detection (accessed on 22 November 2022).
22. Google. Nearby. 2022. Available online: https://developers.google.com/nearby/ (accessed on 22 November 2022).
23. Google. Message API Consumption. 2022. Available online: https://developers.google.com/nearby/developer-guidelines (accessed on 22 November 2022).
24. Google. Room Persistence Library. 2022. Available online: https://developer.android.com/training/data-storage/room (accessed on 22 November 2022).
25. SQLite Database Engine. 2022. Available online: https://www.sqlite.org (accessed on 23 November 2022).
26. Pramanik, P.K.D.; Sinhababu, N.; Mukherjee, B.; Padmanaban, S.; Maity, A.; Upadhyaya, B.K.; Holm-Nielsen, J.B.; Choudhury, P. Power consumption analysis, measurement, management, and issues: A state-of-the-art review of smartphone battery and energy usage. *IEEE Access* **2019**, *7*, 182113–182172. [CrossRef]
27. Laso, S.; Berrocal, J.; Fernández, P.; Ruiz-Cortés, A.; Murillo, J.M. Perses: A framework for the continuous evaluation of the QoS of distributed mobile applications. *Pervasive Mob. Comput.* **2022**, *84*, 101627. [CrossRef]
28. Kim, H.J.; Lee, D.H.; Lee, J.M.; Lee, K.H.; Lyu, W.; Choi, S.G. The QoE Evaluation Method through the QoS-QoE Correlation Model. In Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management, Gyeongju, Republic of Korea, 2–4 September 2008; Volume 2, pp. 719–725. [CrossRef]
29. Peng, Z.; Huang, J.; Wang, H.; Wang, S.; Chu, X.; Zhang, X.; Chen, L.; Huang, X.; Fu, X.; Guo, Y.; et al. BU-trace: A permissionless mobile system for privacy-preserving intelligent contact tracing. In Proceedings of the International Conference on Database Systems for Advanced Applications, Taipei, Taiwan, 11–14 April 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 381–397.

30. Peng, Z.; Xu, C.; Wang, H.; Huang, J.; Xu, J.; Chu, X. P2b-trace: Privacy-preserving blockchain-based contact tracing to combat pandemics. In Proceedings of the 2021 International Conference on Management of Data, Xi'an, Chian, 20–25 June 2021; pp. 2389–2393.
31. Peng, Z.; Xu, J.; Hu, H.; Chen, L.; Kong, H. BlockShare: A Blockchain empowered system for privacy-preserving verifiable data sharing. *Bull. IEEE Comput. Soc. Tech. Comm. Data Eng* **2022**, *1*, 14–24.
32. Li, J.; Peng, Z.; Xiao, B. Smartphone-assisted smooth live video broadcast on wearable cameras. In Proceedings of the 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), Beijing, China, 20–21 June 2016; pp. 1–6.
33. Nguyen, D.C.; Pathirana, P.N.; Ding, M.; Seneviratne, A. Blockchain for secure ehrs sharing of mobile cloud based e-health systems. *IEEE Access* **2019**, *7*, 66792–66806. [CrossRef]
34. Flores-Martin, D.; Berrocal, J.; García-Alonso, J.; Murillo, J.M. Extending w3c thing description to provide support for interactions of things in real-time. In Proceedings of the International Conference on Web Engineering, Helsinki, Finland, 9–12 June 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 30–41.
35. Weise, J. Public key infrastructure overview. *Sun Blueprints Online August* **2001**, 1–27.
36. Jain, L.; Vyas, J. Security analysis of remote attestation. *CS259 Proj. Rep.* **2008**, 1–8.