IoT-based Health Monitoring System for Intensive Care Units

Adriana Collaguazo[a], Rebeca Estrada[b] and Nestor Arreaga
Escuela Superior Politcnica del Litoral, ESPOL,
Electrical Engineering and Computer Science and Technology Information Center,
Campus Gustavo Galindo Km. 30.5 Via Perimetral, P.O. Box 09-01-5863, Guayaquil, Ecuador

Keywords: Mobile App, Intensive Care Unit, Smart Environment, IoT Health Application.

Abstract: Due to the critical condition and unique treatments required by ICU's patients, their vital signs should be continuously and reliably monitored. In this paper, we present a system to address this problem including the design considerations. Our system has three components: i) a mobile application to present the collected data from the hospitals server, ii) a database server with patients data and iii) a transactional server that manages and keeps the connection between the other two components. The mobile application implements a vital signs monitor for each patient including an underlying emergency notification system. After several simulations, we were able to identify that the proposed solution takes around 3 seconds to generate the notification after an emergency alarm occurs. Moreover, the application uses an average of 2.11 Kb/s of traffic data and a maximum of 250Mb of RAM. This system is a feature that will help to track the ICU patients status, allowing doctors and ICU managers to work outside the unit and to determine when their presence is required. Results show that relevant and accurate notifications can effectively reduce the time response in cases of emergency and consequently increases the likelihood of the patients' recovery.

1 INTRODUCTION

The Intensive Care Unit (ICU) is a specialized area of a hospital that provides care to patients with severe or life-threatening illnesses and injuries. This area requires constant and close supervision from life support equipment and medication to ensure normal bodily functions (National Health Service,). A continuous physician supervision is recommended for patients under these conditions, however, treating physicians cannot remain in the ICU unit all the time to take care of these patients. Instead, resident doctors remain in the area to constantly monitor each patient. In the event of an emergency, resident physicians should contact the treating physicians, either by using a communication devices (e.g., phone or beeper) or by personally searching for them, leading to higher response time in emergency cases.

This paper proposes the design and implementation of an automated alert system to tackle the above problem. For instance, a case study was carried out in the ICU of Guayaquil Hospital, in Ecuador. In this hospital, the ICU unit has 21 places for critical care, of which 7 places are for isolated patients, and serves approximately 15 patients (Ministerio de Salud, 2018). The proposed system have several components, such as: 1) a mobile application to present the collected data from the hospitals server, 2) a database server with patients data and 3) a transactional server that manages and keeps the connection between the other two components.

The mobile application shows the vital signs of patients and each doctor can login into the application, then, they can see the area with several windows and the sectors of the ICU and their respective beds. Accelerometer sensors are used to monitor the patients' movement. In the event of any strange movement, the physicians can get an idea of patients' position through the report patients' physical status. Moreover, alerts are automatically generated to notify doctors when a patient's vital signs are outside the established range. This type of emergency is displayed in the application by changing the buttons of the sector, the bed and the parameter to a red color.

This work implements a solution that is able to:

- Report constantly the vital signs of patients by means of non traditional systems.
- In case of emergency, alert are sent to the respec-

[a] https://orcid.org/0000-0002-0707-0226
[b] https://orcid.org/0000-0003-3957-9294

Collaguazo, A., Estrada, R. and Arreaga, N.
IoT-based Health Monitoring System for Intensive Care Units.
DOI: 10.5220/0011339800003286
In *Proceedings of the 19th International Conference on Wireless Networks and Mobile Systems (WINSYS 2022)*, pages 101-106
ISBN: 978-989-758-592-0; ISSN: 2184-948X

tive staff in a hierarchical manner.

- Reduce the current average response time.

This paper is organized as follows: Section 2 describes the prior investigation of the equipment and materials used to design the system. In section 3 we presents the proposed system with the connection between the different modules. Section IV shows the results of the application tests. Finally, section 5 concludes our research work.

2 RELATED WORK

Researchers of the preliminary studies have managed the real-time data over networks to be available to clinicians in the Intensive Care Unit (ICU) anywhere on the web with appropriate software and privileges. Considering the IoT applications covers smart environments in domains such as emergency, health care, and user interaction(Dr. Ovidiu Vermesan et al., 2017).

According Thibaud (Thibaud et al., 2018), identifies that in the near future (within 5 years) several theoretical or pilot projects will tackle different issues such as better integration of pervasive healthcare services with general health care services in a local database environment that ensures data availability and that performs intelligent processes to deliver quicker preliminary information without compromising the energy-efficiency of sensor networks.

In (Lamberti and Montrucchio, 2003), the medical staff equipped with Personal Digital Assistant (PDA) devices both inside and outside the hospital have mobile access to the electronic patient's clinical record. Using a framework for ubiquitous monitoring in an ICU, by the bedside monitoring network, on secure wireless communication channels. Although, the validation of the proposed framework effectiveness is fundamental, and requires the design of the software modules needed.

The system called ADSA (Automatic Detection of risk Situations and Alert) (Ahouandjinou et al., 2016) is based on a hybrid architecture for a visual patient monitoring system using a multi-camera system and collaborative medical sensors network was developed. Although this proposal proved to enable personalization of treatment and management options targeted particularly to the specific circumstances and needs of the individual, it requires the use of cameras to support the decision process which is not possible considered the financial constraints faced by the hospital used as a case of study.

In (Gupta et al., 2016) the IoT-based health monitoring system can provide support in Intensive Care Units(ICU) using an INTEL GALILEO 2ND. This system contains a live graph of the patients heart rate and the temperature is being monitored. However, the system could have been developed in a mobile application for facilitating the access of the users.

In (Chiuchisan et al., 2014) propose the architecture of a health care system for Intensive Care Unit (ICU) through of bedside monitors to monitor and record multiple physiological parameters of patients; Microsoft XBOX Kinect to monitor the movement of the patients; and sensor board for monitoring of environmental parameters such as temperature, humidity, atmospheric pressure. The system is part of a more complex system in development and will be improved by adding new types of sensors like pressure, body weight.

In (Pickering et al., 2018) using the Continuous Time Markov Chain (CTMC) in the develop a bottleneck analysis method to identify opportunities for improvement in the rounding process in an Intensive Care Units at Mayo Clinic. The workflow redesign needs further investigation as the ideal balance between patient care activities and education is still largely unknown and is likely to vary depending on the circumstances.

In (Mahmud et al., 2018) present a Fog-based IoT-Healthcare solution structure and explore the integration of Cloud-Fog services in inter-operable Healthcare solutions extended upon the traditional Cloud-based structure. The mobility of the user and edge-centric affinity of the applications should be handled together by the Fog cluster for better performance.

3 SYSTEM DESIGN

In this section, we present the design of the proposed system taking into account the users requirements to better asset their needs. Based on this requirements a top-level architecture is proposed. In addition, the back end server components are described as well as the application and node design.

3.1 User Requirements

From a user perspective, collecting and sharing health information of patients must satisfy at least the following properties:

- Data accuracy: A margin of error should be established for the data measuring. In this particular case, health measures must be very precise since the slightest difference can have a serious or unwanted impact.

- Availability: The system is used by different users who have different access roles and requirements, and therefore, must be available to all users anytime, anywhere.
- Performance: The system should offer a satisfactory performance to end users. The response time in all procedures should be as low as possible.
- Data privacy: The privacy of health data must be protected for all users, especially those under critical monitoring. The system should provide a proven mechanism to ensure privacy.

3.2 System Architecture

Figure 1 presents the top-level architecture of the proposed ICU automation system. The architecture consists of a Sink Node that is connected to the back-end server using the Local Area Network (LAN). The existing infrastructure provides a WiFi physical layer to connect the smart devices or phones to the local network. The intranet is also connected to this WiFi network. Thus, these devices provide the primary human machine interface for the nurses, doctors and other staff. The server has its own database and application servers and MySQL is used as the Data Base Management System (DBMS) engine while Apache provides the application server functionality. The application is Web-enabled, which requires only a standard browser at the client side.

The first module is composed of the sensors deployed on the beds. These sensors are connected to a gateway designed with an Arduino UNO and an Ethernet Shield, connected directly to the hospital's internal network where the data is transmitted and stored in the MySQL database. In addition, each patient has a monitor connected directly to the hospital network, which provides their vital signs and stores this information in a proprietary database. The second module consists of a PHP-based server based and a MySQL server. The server contains several scripts to perform search, add or update information in the database when a request is received from the mobile application. The third module is a mobile application that allows physicians to visualize patient data.

3.3 Sink Node

The sink node consists of an accelerometer sensor and a wireless transceiver, connected to an Arduino Nano, on the other side each node is connected to BeneView T8 vital signs monitor. The motion sensor MMA7361 is used to detect a fall from the bed. This sensor allows us to measure the acceleration of 3 axes (x, y, z), and it was calibrated correctly at the time of use. Then we connected it to the nano Arduino, together with a nrf24101 transceiver that allows the connection between the nodes and the sink node (Table 1). The Arduino nano is programmed so that when the severity of an axis changes abruptly, the Arduino will register that the person is falling and send a signal to the sink node to notify the emergency.

Table 1: Connections between Arduino and transceivers.

Transmitter		**Receiver**	
nrf24101	Arduino nano	nrf24101	Arduino UNO
VCC	3.3V	VCC	3.3V
GND	GND	GND	GND
CSN	10	CSN	8
CE	9	CE	7
MOSI	11	MOSI	11
SCK	13	SCK	13
MISO	12	MISO	12

The sink node was built over an Arduino UNO, to which an ethernet shield was added to be able to communicate with the server. When the Arduino UNO receives the signal that the patient fell, he connects with the server and sends him a request with the number of the bed to generate an alarm in the database which the cell phone will notify the doctors. This module should be improved with equipment with higher capacity and durability, which can support long periods, such as a Beaglebone or a Raspberry.

3.4 Transactional Server

The transactional server consists of several PHP scripts to perform the requests generated by the users of the mobile application. These scripts receive a request from the application to search information from the database and return the requested query in a specific format, which is transferred through JSON messages because of their simplicity and the speed to transfer the data. To prevent the leakage of sensitive information or problems such as SQL injection, the data received in each request was sent separately and parameterized, and no script contains queries that could reveal data about database structures. Moreover, as an extra security measure, the patient's information and vital signs are not stored in the device running the mobile application.

3.5 Database Server

The management of MySQL database was carried out by means of PHPMyAdmin software tool. With this tool, we were able to maintain an up-to-date copy of

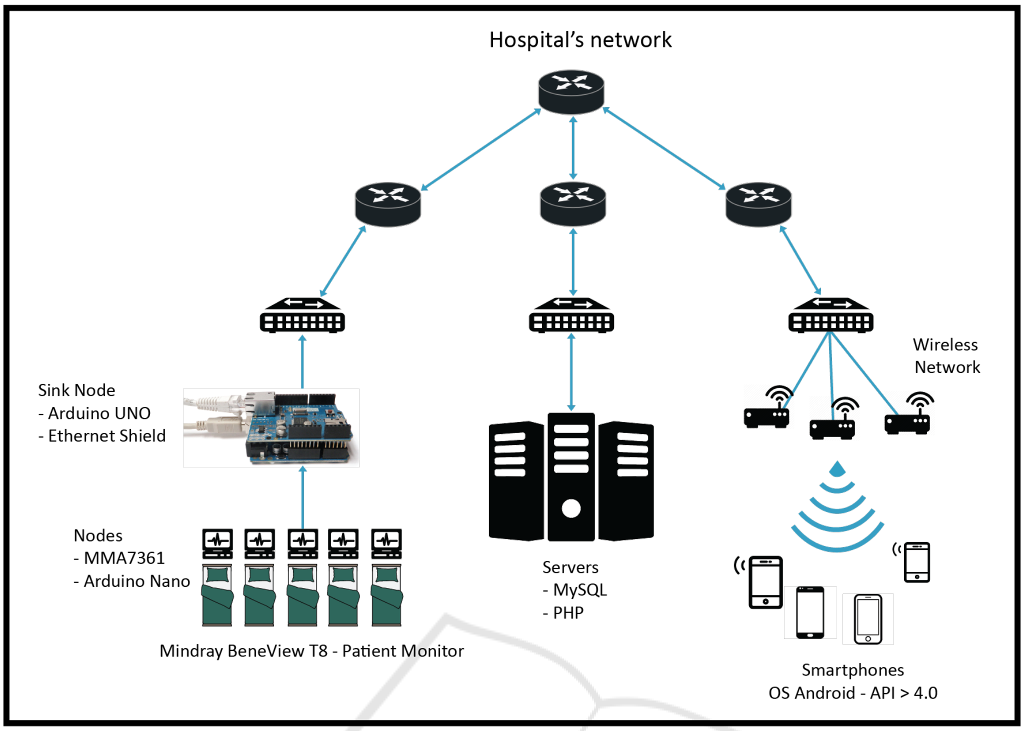

Figure 1: Top-level system architecture of the UCI system.

the database structure and its contents from a hospital vendor's proprietary equipment. Thus, we will not affect the integrity of the actual data of the patients in the intensive care unit. The patient's name, location, blood pressure, heart rate, temperature and oxygenation are some of the information stored as the patient data. This data can also be consulted through the mobile application.

3.6 Mobile Application

The mobile application was developed using Android Studio for phones with Android operating system because it is a prevalent mobile operating system in Ecuador. This IDE has available libraries that allow direct interaction with mobile functions and offers a suitable graphical environment for view design.

The application generates POST requirements to communicate with the PHP server. Figure 2 illustrates different views that the mobile application has. The first is used to allow the user can connect to the system using their identification number and password. The second view shows the four sectors of the ICU area and by selecting any of these, the numbered beds in the sector is shown. For each bed, the app shows the patient's name when there is a patient using the bed otherwise it shows "NO DATA". By choosing the patient, a parameter view is presented, where the values of the patient's vital signs in real time.

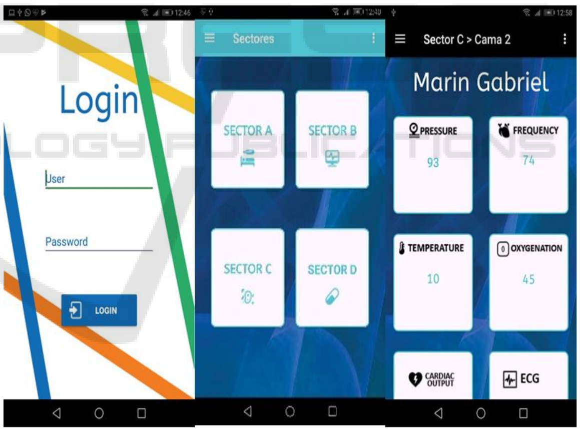

Figure 2: Mobile Application Interfaces.

To facilitate the data analysis, the user can choose any of the parameters and a real-time graph of this parameter is shown. In addition, the user can choose to forecast the trend of the parameter, which is done using the 3rd order moving average of the parameter. To obtain this data, the system generates requests to the server every 0.5 seconds sending the corresponding information of the selected parameters, bed and section.

The mobile application has a persistent service that keeps the medical staff aware of the patient's status. This service sends a request every 3 seconds and thus be able to issue timely emergency alerts.

When an event is identified, a vibrating and sound alert notification is generated. Such notifications may not be silenced unless the user voluntarily logs out or the emergency is resolved.

For the generation of alarms, a hierarchical classification is implemented according to users' role as follows:

- Physician on shift: instant alert generation.
- Sector supervisor: 5 minutes after the emergency alert is generated.
- Area supervisor: 10 minutes after the emergency alert is generated.

To keep control of notifications, the user's role is saved in the internal memory of the mobile phone while his session is active.

4 RESULTS ANALYSIS

Once the system was fully developed, we perform several tests to validate the functionality of each component. The tool used for testing was Android Profile, which is part of the Android Studio software package () [7]. This tool allows the developer to determine the usage of CPU, memory, power and data. Each test was performed fifteen times and the results were averaged. The first test consisted of measuring the total time between the generation of an emergency in the database and the notification in the mobile application to determine the average time to generate emergency notifications.

Each sample time period began with the activation of the emergency trigger and ended with the received pop-up notification. For better precision, a digital trigger was used to start the tests. Also, the trigger system and the alarm were installed on different networks and separated by a distance of 9.1 miles.

During the initialization period, the average response took around 5.5 seconds with standard deviation of 1.5s. With the system in steady state, the mean response time was 3 seconds with a standard deviation of 0.5s, as shown in Fig 3

The second test was used to measure mobile data consumption and to estimate the maximum bandwidth consumption during a 20-second period while keeping the real-time parameters window open, as it is the most resource-consuming window of the application due to the constant checking of the different vital signs of the patient.

Fig. 4 shows the data rate used for transmission while performing the parameter analysis in real-time since it is the most demanding activity of the mobile application. Each test was run for a duration of 20 seconds, and the average data rate was 2.11 Kbps.

The third test allows us to measure the bandwidth consumption of the notification generation service to determine the minimum bandwidth consumption of the mobile application when it is in the background. This test consists of having the alarm system running in background mode considering a time 20-seconds interval. In this case, the average data rate was 0.51 Kbps, as shown in Fig. 5

The purpose of the paper was to analyze the amount of resources required on the mobile phone to run the application in a controlled environment. However, it is important to remark that the used sensor was a low cost sensor, which means that it is device with a time-consuming maintenance. In addition, the accelerometer starts to overheat after a few hours, which might affect the response time and the accuracy. This should be taken into account when implementing a sustainable solution for a real environment.

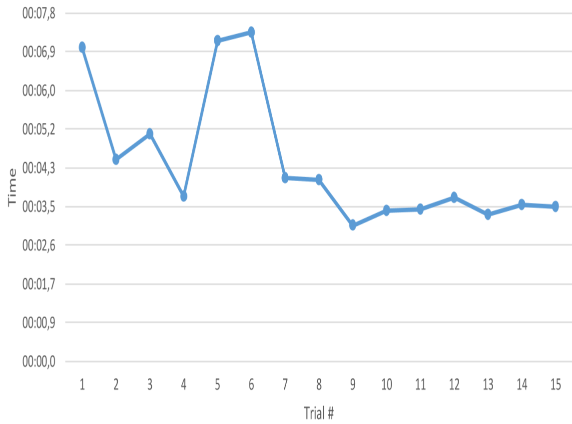

Figure 3: Tests of the time interval between the patient's fall and the notification of the alert.

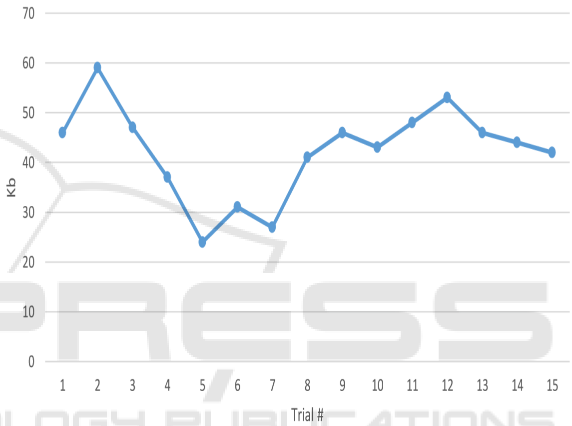

Figure 4: Data Rate for the functionality of parameter analysis in real-time.

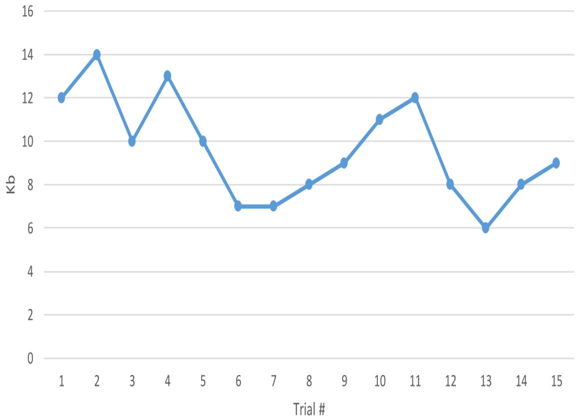

Figure 5: Data Rate required for notification generation service.

5 CONCLUSIONS

In this paper, we presented an innovative system that enables the medical staff to check and analyze the vital signs of ICU's patients from distance, having a constant control of them. Pop-up notifications alert the response team if one or more patients have a significant change of vital signs. This allows the medical team to react quickly and apply the appropriate treatment. The database access table helps the hospital management team to control and protect patient data. This also helps the system provide a hierarchically controlled alert. The resources required for mobile application allows it to be run in low-mid range and higher phones. The application design facilitates the training period its users, which helps the users' work environment. As future work, we plan to make the system more functional, allowing to establish relationships between several parameters and obtain more accurate information about the patient's condition. Patients' data from the system could be analyzed with the machine learning algorithm to predict their status and reduce the generation of emergency notifications to the cases where the presence of the main doctor is really necessary.

REFERENCES

Ahouandjinou, A., Assogba, K., and Motamed, C. (2016). Smart and pervasive ICU based-IoT for improving intensive health care. In *2016 International Conference on Bio-engineering for Smart Technologies (BioSMART)*, pages 1–4. IEEE.

Chiuchisan, I., Costin, H. N., and Geman, O. (2014). Adopting the internet of things technologies in health care systems. In *EPE 2014 - Proceedings of the 2014 International Conference and Exposition on Electrical and Power Engineering*, pages 532–535. Institute of Electrical and Electronics Engineers Inc.

Dr. Ovidiu Vermesan, SINTEF, N., Friess, D. P., and EU, B. (2017). *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*.

Gupta, P., Agrawal, D., Chhabra, J., and Dhir, P. K. (2016). IoT based smart healthcare kit. In *2016 International Conference on Computational Techniques in Information and Communication Technologies, ICCTICT 2016 - Proceedings*, pages 237–242. IEEE.

Lamberti, F. and Montrucchio, B. (2003). Ubiquitous real-time monitoring of critical-care patients in intensive care units. In *Proceedings of the IEEE/EMBS Region 8 International Conference on Information Technology Applications in Biomedicine, ITAB*, volume 2003-Janua, pages 318–321.

Mahmud, R., Koch, F. L., and Buyya, R. (2018). Cloud-Fog Interoperability in IoT-enabled Healthcare Solutions. In *Proceedings of the 19th International Conference on Distributed Computing and Networking - ICDCN '18*, pages 1–10, New York, New York, USA. ACM Press.

Ministerio de Salud, E. (2018). Rendición de cuentas 2018.

National Health Service. Intensive care - NHS.

Pickering, B., Li, J., Dong, Y., Gajic, O., and Lee, H. K. (2018). Bottleneck Analysis to Improve Multidisciplinary Rounding Process in Intensive Care Units at Mayo Clinic. *IEEE Robotics and Automation Letters*, 3(3):2678–2685.

Thibaud, M., Chi, H., Zhou, W., and Piramuthu, S. (2018). Internet of Things (IoT) in high-risk Environment, Health and Safety (EHS) industries: A comprehensive review. *Decision Support Systems*, 108:79–95.