MSI Journal of Multidisciplinary Research (MSIJMR)

Frequency:- Monthly Published by MSI Publishers
ISSN:- 3049-0839 (Online)
Journal Link:- https://msipublishers.com/msijmr/
Volume:- 2, *Issue:-* 2 (February-2025)

Article History
Received on :- 05-02-2025
Accepted on :- 10-02-2025
Published on :- 13-02-2025

Total Page:- 01-15 **DOI:- 10.5281/zenodo.14850575**

Design and Creation of Remote Temperature Measure System with An Esp32 to Evaluate Patient Health

By

Kouassi Bonaventure Ago[1], Babatounde Raoul Ferry Aiwala[2], Probus Manhougbe A. Farel KIKI[3], Thakar Saloni[4], Sylvain Meinrad Donkeng Voumo[5*]

Research Scholar, Department of Computer Engineering Gokul Global University, Sidhpur (Patan), India.[1]
Ecole de Génie électrique et Informatique, Université catholique de l'Afrique de l'Ouest, Cotonou, Benin.[2]
Doctor of Engineering, University of Abomey-Calavi/UAC. Polytechnic School of Abomey-Calavi (EPAC), Benin.[3]
Assistant Professor, computer Engineering Department, Gokul Global University, Sidhpur (Patan), India[4]
Research Scholar, Department of Environmental Sciences, Gokul Global University, Sidhpur (Patan), India.[5*]

Abstract:- Remote patient monitoring has become increasingly crucial in modern healthcare, particularly for patients who require constant observation but cannot remain in medical facilities. This project focuses on the development of a remote medical monitoring system using the ESP32 microcontroller for real-time health evaluation. The system is designed to monitor vital signs and body temperature transmitting the data via Wi-Fi or Bluetooth to a mobile or web application for continuous patient evaluation. The core objective is to create an affordable and efficient monitoring solution that can be deployed in rural and remote areas where access to healthcare is limited. The system provides real-time alerts in case of abnormal readings, enabling timely medical interventions. This research highlights the system's design, implementation, performance, and potential impact on healthcare delivery, especially for elderly patients or those with chronic conditions. The project shows promising results regarding system accuracy, cost-effectiveness, and user satisfaction.

Keywords: Remote patient monitoring, ESP32, IoT healthcare, real-time alerts, telemedicine.

Introduction

Healthcare systems worldwide face increasing demands due to ageing populations, rising numbers of chronic disease patients, and challenges in accessing healthcare in remote regions. Remote patient monitoring (RPM) systems have emerged as effective tools to manage healthcare needs by allowing continuous observation of a patient's vital signs outside of traditional clinical settings. RPM systems are part of a larger trend towards the Internet of Things (IoT), where interconnected devices collect, analyze, and transmit data to improve decision-making and service delivery (Ravi et al., 2022). IoT-based health monitoring solutions have the potential to reduce the burden on hospitals, enhance early diagnosis, and improve healthcare accessibility, particularly in underserved areas (Chatterjee & Armenta, 2021). The ESP32 microcontroller, with its low-cost, low-power Wi-Fi and Bluetooth capabilities, is an ideal platform for developing affordable health monitoring systems. Its versatility and ease of programming make it a popular choice for building IoT applications, including medical systems that require real-time data transmission and alerts (Zhu et al., 2021). This study aims to develop a remote monitoring system based on the ESP32, leveraging its IoT potential to track patient health and alert healthcare providers of any abnormalities. Such systems are particularly beneficial in rural or underserved areas, where access to healthcare facilities is limited, and continuous monitoring can save lives. Traditional patient monitoring methods rely on periodic visits to healthcare facilities, which may be insufficient for patients with chronic conditions or those recovering from surgery. These approaches are also resource-intensive, requiring staff and infrastructure that can be costly and unavailable in many parts of the world. As a result, patients in rural or remote regions often struggle to receive timely medical care, leading to severe health complications. Furthermore, the necessity for continuous monitoring has been underscored by recent global health crises, such as the COVID-19 pandemic, which overwhelmed healthcare systems worldwide (Giannini et al., 2020). The development of a cost-effective, reliable, and easy-to-use remote medical monitoring system can help address these challenges. By integrating IoT technologies, such systems reduce the burden on hospitals, allow patients to stay at home, and ensure that healthcare providers can monitor vital signs remotely. However, issues such as data security, integration with existing healthcare frameworks, and affordability must be carefully addressed to ensure successful deployment and adoption (Hussain et al., 2021). This study aims to develop a medical monitoring system to track vital parameters, particularly temperature, using IoT technology, provide remote monitoring capabilities to ensure real-time data transmission, and address privacy and security concerns in the transmission and storage of sensitive health data. Despite the advances in IoT-based healthcare solutions, several gaps remain unaddressed. Many existing systems rely on manual data collection or intermittent monitoring, which can miss critical changes in a patient's condition (Sharma et al., 2022). Current IoT-based health monitoring systems often focus on specific health conditions without offering a complete, integrated solution. Security concerns regarding patient data in IoT applications remain under-addressed in many systems, increasing the risk of data breaches and privacy violations (Sicari et al., 2015). The motivations for this study include increasing healthcare costs, as rising healthcare expenses necessitate cost-effective solutions that reduce hospital visits and enable remote monitoring. The global elderly population is growing, requiring continuous monitoring for chronic conditions to allow timely interventions and reduce hospitalizations. Traditional patient monitoring provides intermittent data, which may miss critical health changes. A real-time monitoring system ensures better patient care and immediate responses in emergencies. The rapid development of IoT technology and affordable microcontrollers, such as the ESP32, makes reliable, low-cost healthcare monitoring solutions feasible. Remote medical monitoring enhances healthcare access for patients in remote areas, ensuring consistent monitoring and reducing the need for frequent in-person visits. By providing real-time data and early detection of potential health issues, these systems enable quicker interventions, improving patient outcomes and potentially saving lives.

Review of Literature

The reviewed literature explores the development and implementation of Internet of Things (IoT)-based health monitoring systems, predominantly leveraging the ESP32 microcontroller (Chakraborty & Aithal, 2023; Foltynek et al., 2019; Ghosh et al., 2018; Li, 2023; Naresh et al., 2024; Rusimamto et al., 2021). These systems aim to enhance patient care, facilitate remote monitoring, and personalize healthcare through the integration of diverse sensors, communication protocols, and data processing techniques. Several key themes emerge:

Remote Monitoring and Real-time Alerts: A significant focus is on real-time data acquisition and transmission, enabling remote monitoring of vital signs such as heart rate, SpO2, temperature, and even saline levels (Ghosh et al., 2018; Li, 2023; Naresh et al., 2024; Rusimamto et al., 2021). This allows for timely interventions and reduces the need for constant direct supervision.

Sensor Integration: The systems employ a wide array of sensors, including load cells (for saline levels), pulse oximeters (MAX30100, MAX30102), temperature sensors (DS18B20, MLX90614), and even fingerprint recognition (Ghosh et al., 2018; Naresh et al., 2024; Rusimamto et al., 2021).

Communication Protocols: MQTT and MQTT-S are frequently utilized for reliable message delivery, especially in healthcare settings (Chakraborty & Aithal, 2023; Ghosh et al., 2018; Foltynek et al., 2019; Li, 2023). BLE is also employed for short-range communication with wearable devices (Li, 2023).

Data Processing and Analysis: Some systems incorporate machine learning algorithms, such as Random Forest, for predictive analytics and personalized healthcare recommendations (Naresh et al., 2024).

Web and Mobile Interfaces: Web servers and Android applications are used to display health data, making it accessible to caregivers and patients remotely (Rusimamto *et al.*, 2021).

Security Concerns: Several papers acknowledge the importance of data privacy and security, particularly when dealing with sensitive medical information. Encryption techniques like AES are proposed as solutions (Al-Mashhadani & Shujaa, 2022; Naresh et al., 2024).

Scalability and Centralized Control: Some systems aim to manage multiple IoT devices from a centralized platform, making them applicable to various settings like smart homes and hospitals (Chakraborty & Aithal, 2023).

Cost-Effectiveness: The use of low-cost hardware like ESP32 is emphasized as a way to make these systems accessible and affordable (Chakraborty & Aithal, 2023; Foltynek et al., 2019; Ghosh et al., 2018).

Table 1: ESP32-Based Health Monitoring Systems

Authors	Microcontroller	Sensors	Communication	Data Processing	Advantages	Disadvantages
Ghosh et al. (2018)	ESP32	Load Cell	MQTT-S	N/A	Real-time alerts, remote monitoring	Internet dependency, security risks
Naresh et al. (2024)	ESP32	MAX30102, Pulse sensor, Fingerprint sensor	N/A	Random Forest	Predictive analytics, personalized healthcare	Data privacy concerns
Chakraborty & Aithal (2023)	ESP32	N/A	MQTT	N/A	Centralized control, scalability	Internet dependency, security concerns
Hridhya (2023)	ESP32	MAX30100 (Pulse Oximeter), DS18B20 (Temperature Sensor)	N/A	N/A	Easy-to-access web server	Internet dependency, security concerns
Li (2023)	ESP32	Smart Pillow, Wristwatch, Pulse Oximeter	BLE; MQTT	N/A	Real-time alerts for elderly care	Notification latency
Foltynek et al. (2019)	ESP32	Vibration Sensor	MQTT	N/A	Cost-effective, optimized performance	MQTT complexity
Al-Mashhadani & Shujaa (2022)	ESP32	N/A	N/A	Advanced Encryption Standard	Enhanced IoT security	Computational overhead

				(AES) Encryption		
Rusimamto et al. (2021)	Arduino Pro Mini, ESP32 Camera	MLX90614 (Infrared Temperature Sensor)	N/A	N/A	High accuracy	Outdoor sensitivity

N/A: In this context, **N/A** means "Not Applicable" or "Not Available; **MQTT:** Message Queuing Telemetry Transport; **BLE**: Bluetooth Low Energy; **MQTT-S:** Message Queuing Telemetry Transport for Sensor Networks

Research and Methodology

Hardware study and web platform design

Presentation of the system

The system, which provides local medical monitoring, is an embedded system mounted around a specialized board for Internet of Things (IoT)-oriented applications. The sensor used in our system's electronic assembly is specialized for measuring body temperature. The program embedded in this assembly extracts data relating to the health status of the connected patient, and presents this data stream via a local or remote web server to doctors and other healthcare professionals. The aim is to improve patient supervision and medical care. Our system can be used to monitor body temperature daily.

ESP32 development board

The ESP32, developed by Expressive, is a low-cost development board dedicated to the Internet of Things (IoT) and embedded applications. It's a system-on-a-chip (SoC) with Wifi and Bluetooth wireless communications. The chip is Dual Core. It offers two 32-bit processing units operating at a frequency of 160 Mhz. This module uses the WIFI 802.11 b/g/n communication protocol, which enables the internal microcontroller to connect to a WIFI network and make simple TCP/IP connections, or to be detectable by other devices using the same WIFI protocol.

ESP32 module features

RAM: 520 KB; Flash 64 M Bytes; Power supply 2.2 V - 3.6 V; Consumption 80 mA Average; Temperature -40°C to 125°C; Wifi 802.11b/g/n and Bluetooth 4.2 BR/EDR + BLE

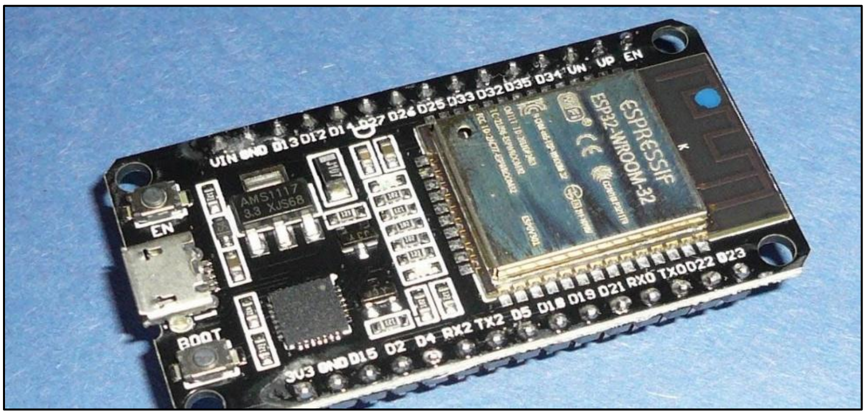


Figure 1: ESP32 Card

ESP32 board pinout

The ESP32 module has a total of 32 I/Os, 26 digital and 18 analog. The touch screen can also be used to create a human-machine interface.

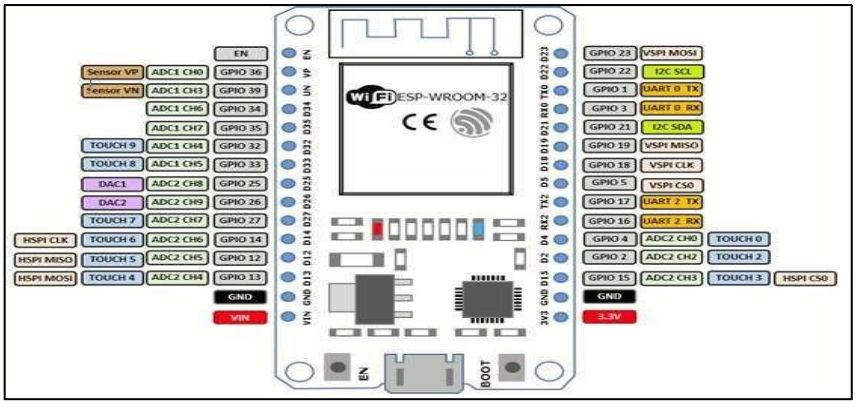


Figure 2: Pin assignment for ESP3 modules

Sensors

A transducer is a device for sampling information that, from a physical quantity, produces another physical quantity of a different nature (usually electrical), which is an image of the quantity sampled and can be used to indicate a measurement. The purpose of a sensor is to transform the physical input property and convert it into an electrical signal compatible with electronic circuits. Sensors are electronic devices that measure a physical quality such as light or temperature and convert it into a voltage.

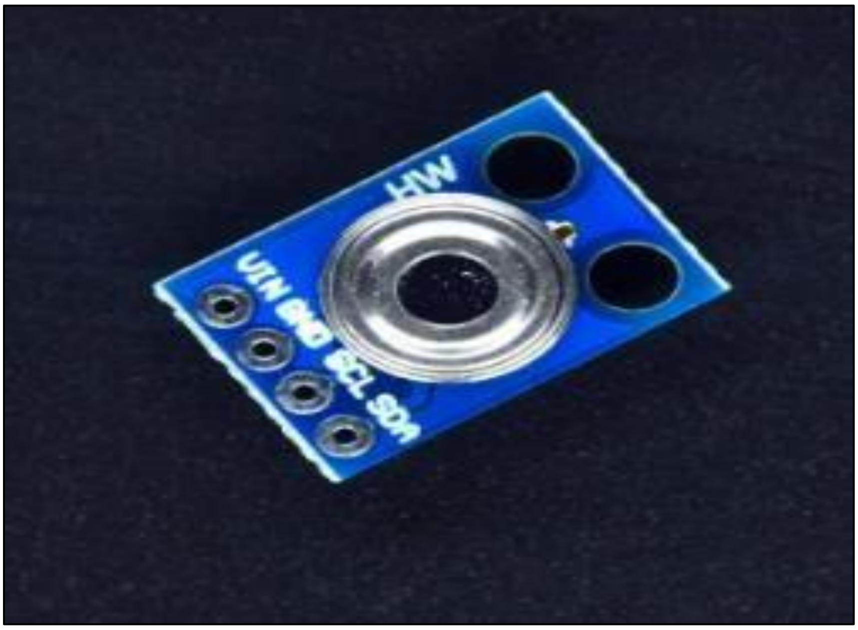


Figure 3: Temperature sensor without infrared contact MLX90614 [38].

Pin description

VIN: 1.8-5V main power input terminal.
SCL: clock connected to I2C bus.
SDA: data connected to the I2C bus.
GND: ground wire.

Other accessories

There are several pieces of equipment in this category (such as Oled screens, electronic wires, test boards, etc.) that can be attached to the ESP32 to facilitate control. Here are a few examples:

Oled screen SSD1306 128*64

This display module features a monochrome OLED screen with a resolution of 128 pixels by 64 pixels. It measures 0.96” diagonally. This OLED display uses the SSD1306 OLED driver IC. SSD1306 OLED displays have three types of communication interface:

6800 8-bit parallel interfaces

3 or 4-wire SPI
- I2C

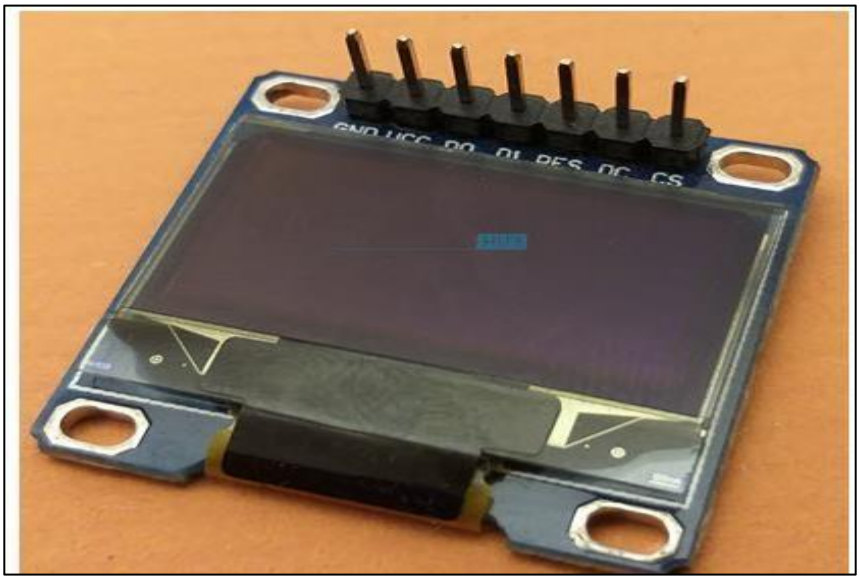


Figure 4: Ecran Oled SSD1306 128*64.

Electronic wires

Electrical wires are components used to transmit signals or electrical energy.

Figure 5: electrical wires.

Test plate

A test plate is extremely useful for solderless electronic assembly, particularly in conjunction with microcontroller boards such as Arduino. So, it's vital to know how to use them and understand the principles involved. It's not very complicated, but it's important to understand. The test plate is used with stras, pieces of copper wire. Single stranded (this is important) of different sizes and lengths. The ends of the stras must be about 1 cm bare.

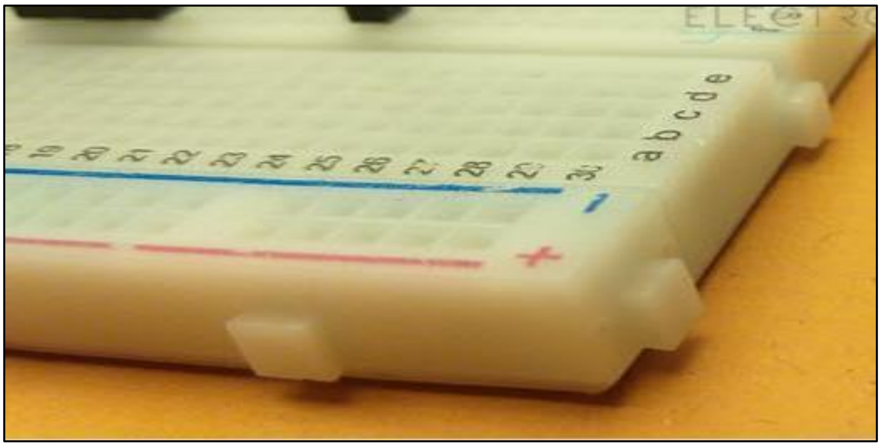


Figure 6: The test plate

The electric battery

It's an electrochemical device that generates electricity by converting chemical energy into electrical energy through a redox reaction.

Figure 7: the electric battery

Synoptic diagram

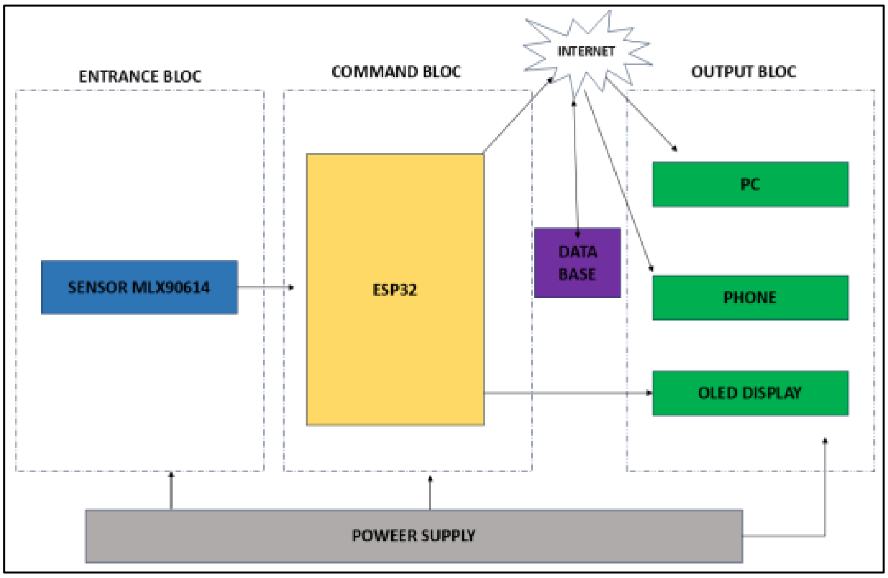


Figure 8: synoptic diagram

Operating principles

The medical monitoring system consists of an electronic device and a web platform. First of all, you need to register on the web platform and have your account activated and have a profile.

Depending on your profile:

• If you are a caregiver, you can add a device to the platform;

• If you are a nurse, you can add a device and register a patient on the platform;

• If you are a doctor, you can add a device, register a patient, associate a device with a patient, dissociate a device and consult a patient's results on the platform;

• If you are an administrator, you can add a device, register a patient, associate a device with a patient, disassociate a device, view a patient's results, activate a user account and assign a profile to a user on the platform;

The electronic device is made up of electronic circuits. The most important element here is the temperature (MLX90614 sensor). The system is activated by an ON/OFF switch. As soon as it is switched on, the MLX90614 sensor supplies the processing board with temperature information. This information will be displayed on the oled screen, and will also be sent to the database for later consultation by the user.

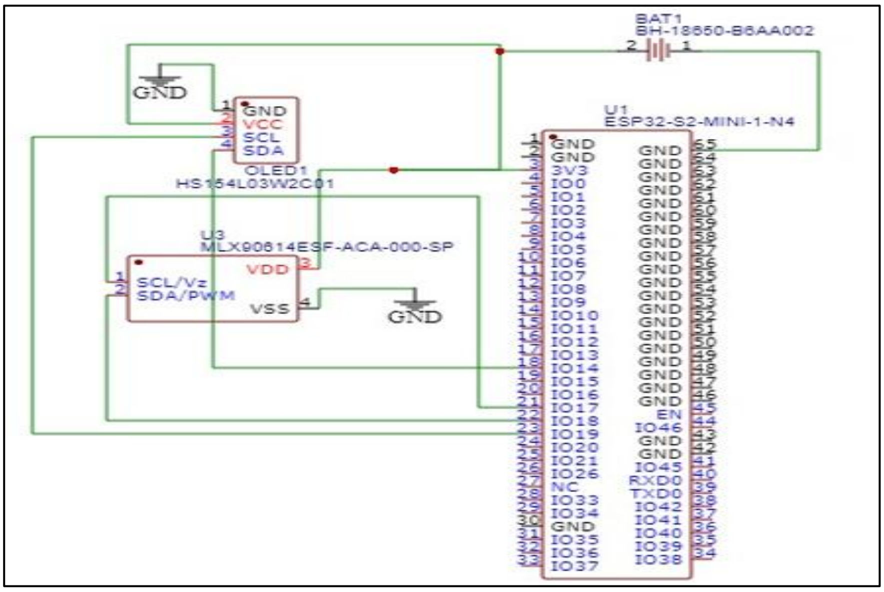


Figure 9: Electrical diagram

Electrical diagram

Designing a Web Platform

Functional requirements

The platform will enable users to: register; register a patient; add advice; associate a device with a patient; dissociate a device; view patient data.

Non-functional requirements

Non-functional requirements are objectives linked to system performance and environmental constraints. The platform must be ergonomic; Fast processing; Ensure confidentiality; Remain available; Responsivity

Implementation and Design

This phase in the software development process enables us to clearly define the system's operations (how it works) in order to facilitate its realization. We chose the UML modeling language and created use case, class and sequence diagrams.

Use case diagram

This diagram shows the functional behaviour of the platform. The main actors in the system are users, page administrators and platform administrators.

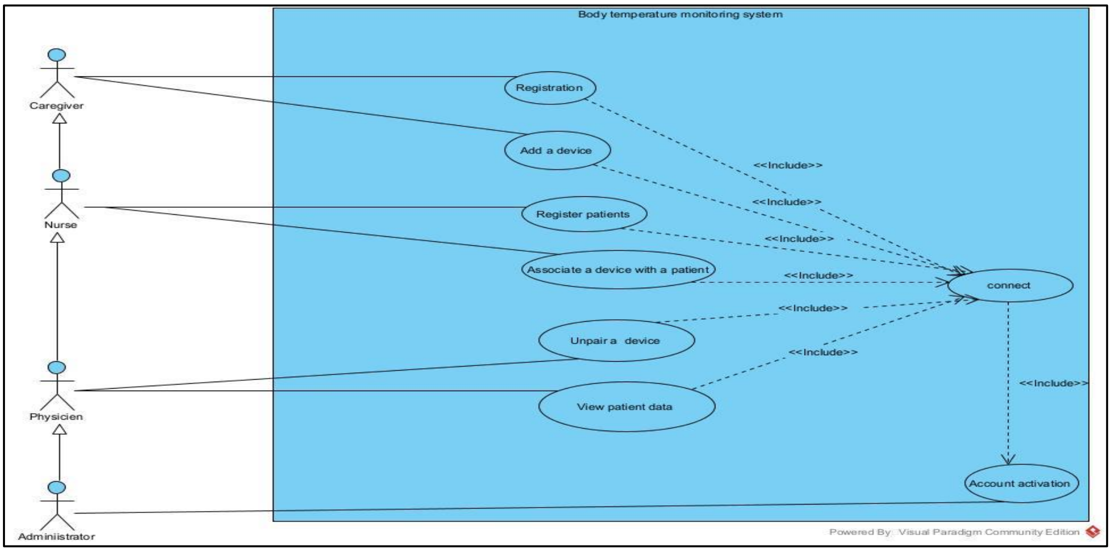


Figure 10: Use case diagram

Class diagram

This diagram shows the various classes and the relationships between them.

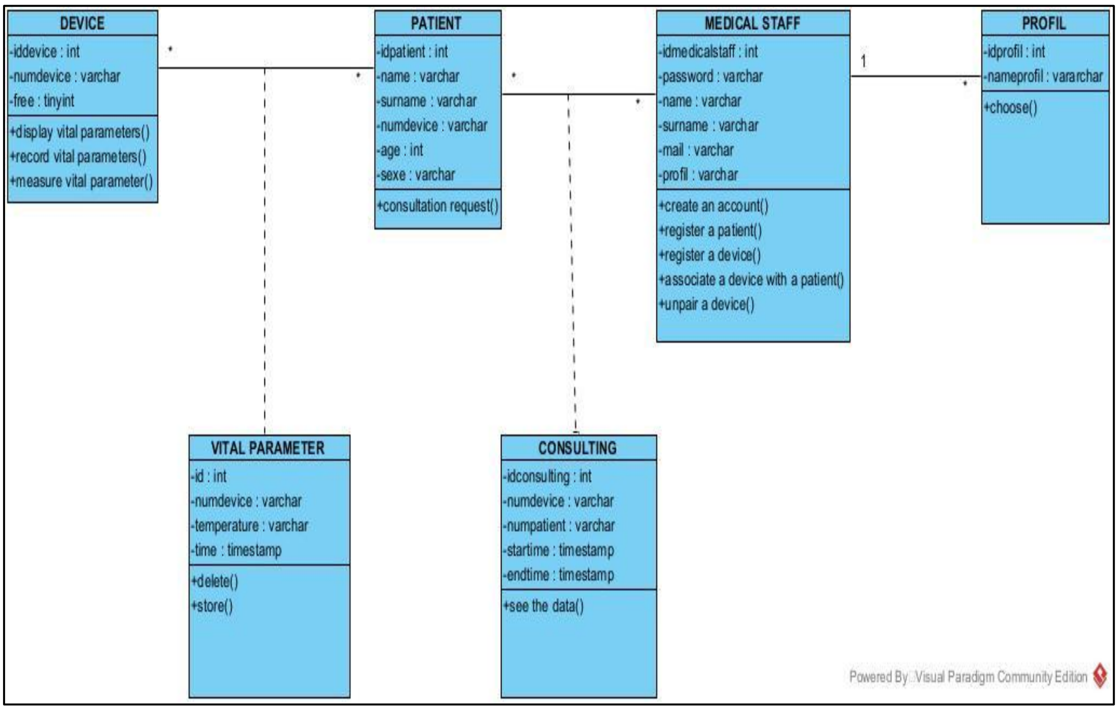


Figure 11: Class diagram

Sequence diagram

In this section, we describe the interactions between objects using sequence diagrams. To do this, we have chosen two use cases to illustrate: registering on the platform and creating a publication.

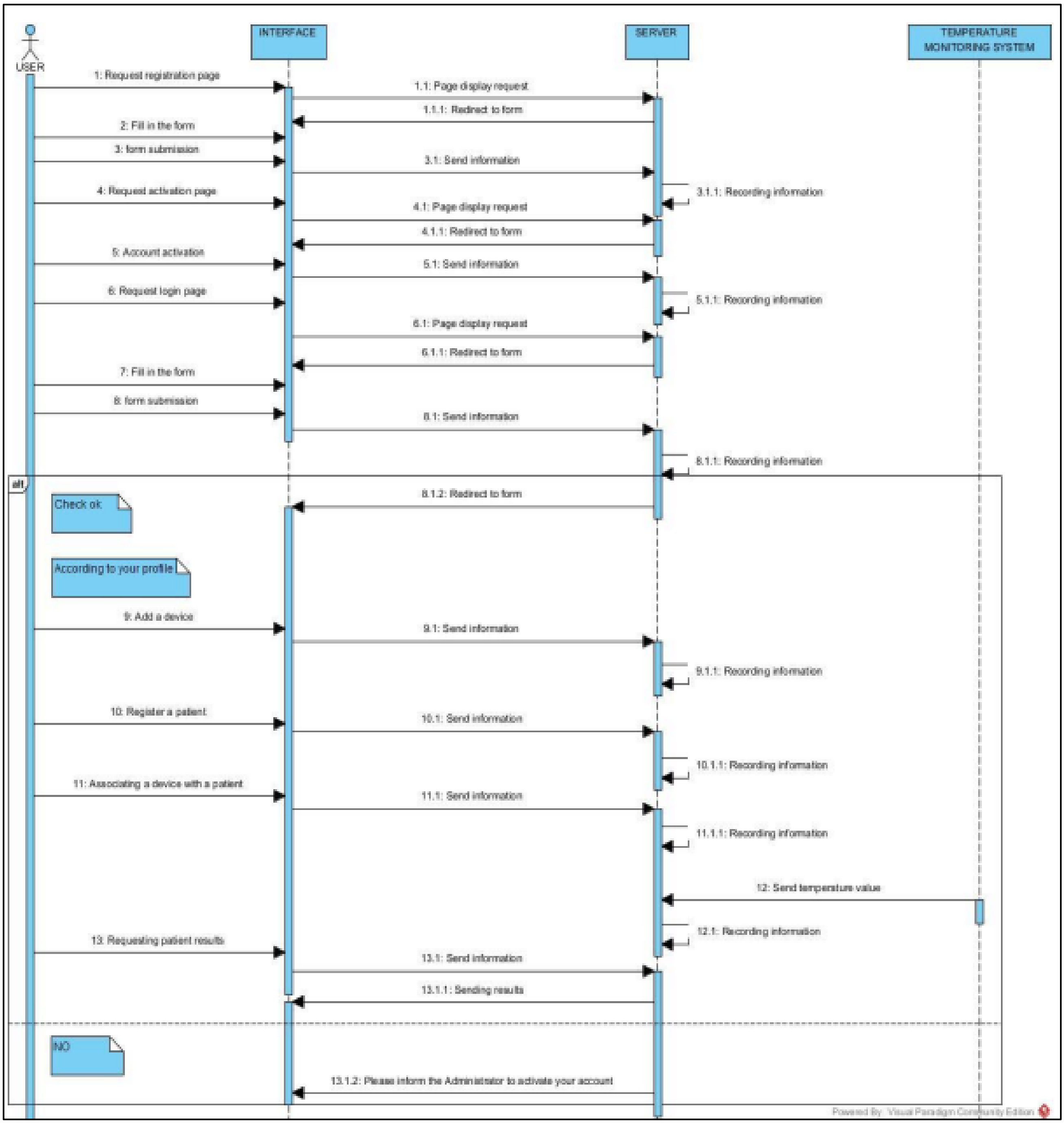


Figure 12: Sequence diagram

Presentation used

The Arduino IDE: Arduino development is simple: - the Arduino language is based on C/C++, with Arduino-specific functions and libraries (input/output management); the Arduino board is connected to the PC and the program transferred to the board. The programming software for Arduino modules is a cross-platform Java application (running on any operating system), acting as a code editor and compiler. It can transfer the program via serial link (RS232, Bluetooth or USB, depending on the module). The software is very easy to use, and there are very good, well-done tutorials with explanations in French. Numerous examples are provided. The example files are really well documented and allow you to code very complicated operations without too much effort. The libraries provided make it easy to use complex components in just a few clear lines.

Visual Studio Code IDE: Visual Studio Code is an extensible code editor developed by Microsoft for Windows, Linux and macOS. Features include debugging support, syntax highlighting, intelligent code completion, snippets, code refactoring and integrated Git. Users can modify the theme, keyboard shortcuts, preferences and install extensions that add further functionality.

WampServer: WampServer is a WAMP (Windows Apache MySQL PHP) Web development platform, enabling PHP scripts to be run locally (without having to connect to an external server). WampServer is not software per se, but an environment comprising three servers (Apache, MySQL and MariaDB), a script interpreter (PHP) and phpMyAdmin for Web administration of MySQL databases.

Fritzing: is an open-source PCB design program that lets you design circuits entirely graphically.

A database management system (DBMS) is a software system used to store, manipulate or manage, and share data in a database, guaranteeing the quality, permanence and confidentiality of information, while concealing the complexity of operations.

Programming language

PHP: PHP is a popular general-purpose scripting language particularly suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the world's most popular websites. It's an easy language to master, and speeds up programming thanks to its simple syntax. It's portable; you can adopt object-oriented or procedural programming styles; it supports a large number of databases and its documentation is well supplied. However, PHP has no debugger.

HTML5 and CSS3: HTML5 is the markup language for Web pages. CSS3 is used to manage the presentation of HTML documents. These two technologies are essential for web development.

JavaScript: JavaScript (often abbreviated to "JS") is a lightweight, object-oriented scripting language, primarily known as the scripting language for web pages. Along with HTML and CSS, JavaScript is sometimes considered one of the core technologies of the Web [30].

Bootstrap: Bootstrap is a front-end framework containing HTML, CSS and JavaScript extensions. With a responsive grid system, it enables the creation of responsive sites. It is one of the most widely used front-end frameworks in the world.

jQuery: jQuery is an open-source, cross-platform JavaScript library created to facilitate the writing of client-side scripts in the HTML code of web pages. It is currently the most widely used front-end framework in the world. We opted for jQuery because it simplifies the use of JavaScript and is widely available on the Internet.

MySQL: MySQL is a relational database management system (RDBMS). It is one of the most widely used database management systems in the world, both for the general public (mainly web applications) and for professionals, in competition with Oracle, PostgreSQL and Microsoft SQL Server. MySQL's advantages are many: it's open source and free, its performance is excellent, it's multi-threaded and multi-user, and it's designed to work perfectly with PHP, the most widely used server-side web programming language. However, it is not recommended for large-scale projects.

Results

Platform interface presentation

Home page

This is the page that is accessible to everyone and gives access to the registration and login pages. Below is also the form we need to complete to register on the platform.

pages. Below is also the form we need to complete to register on the platform.

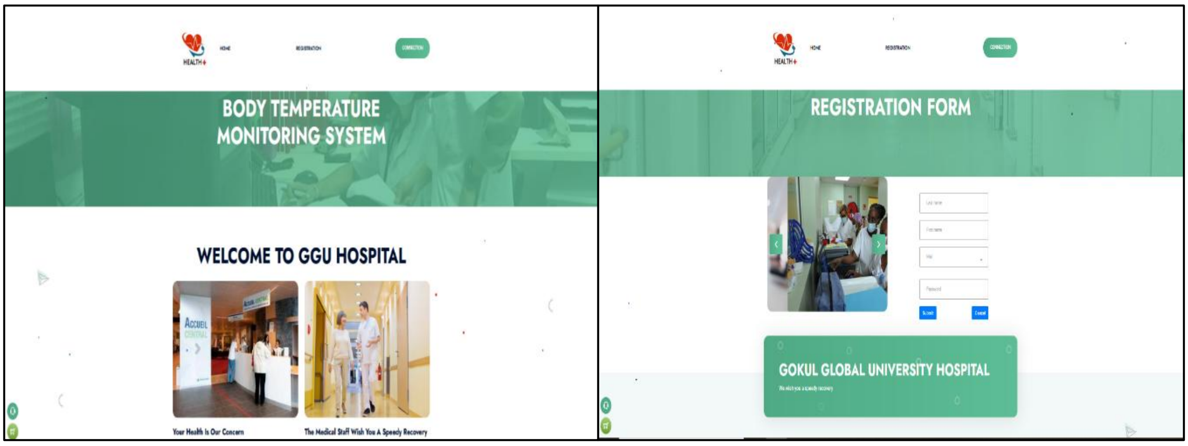


Figure 13: Home page and registration page.

Login page and Add device page

The login page (Figure) allows platform users to authenticate themselves. It is the entry page for full access to the functionalities we offer. If the user is not a member, he can click on the "registration" menu to register and tell the administrator to activate his account. Members can also reset their password (if they've forgotten it) by clicking on the "forgot password" link. (Figure) lets you add devices to the platform.

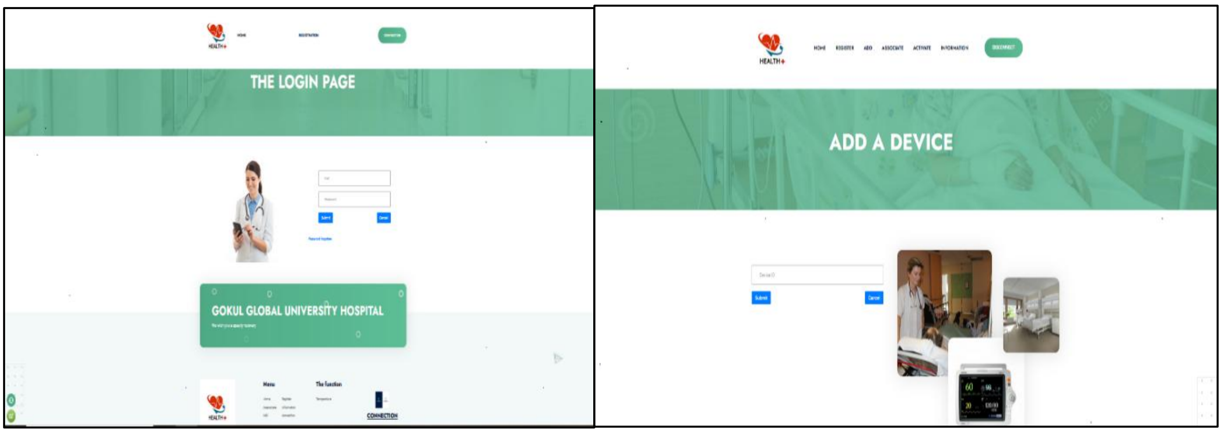
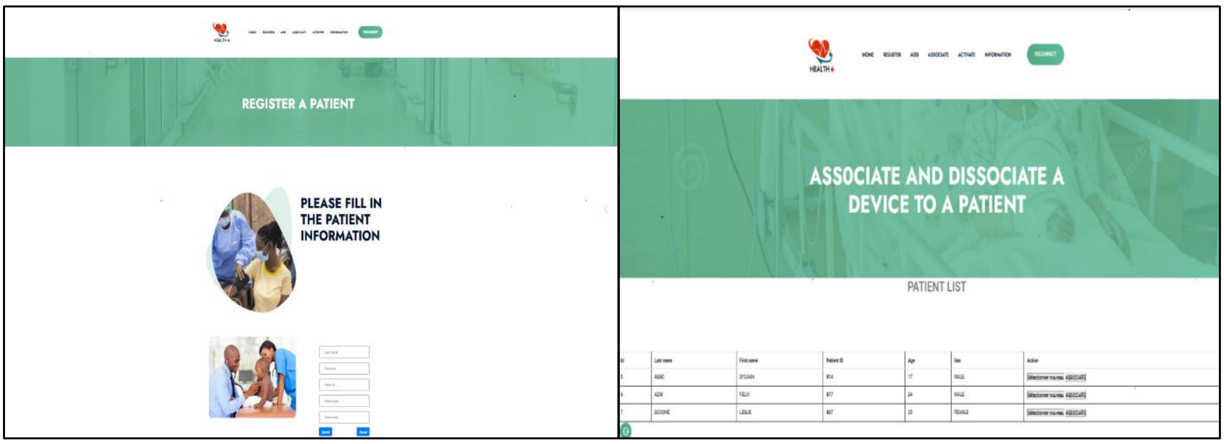


Figure 14: Login Page and Add device page

Patient registration page and Page associates a device

The page enables patient registration on the platform. This page allows you to associate a device with a patient, and also to dissociate a device, then a consult link is available to view the results



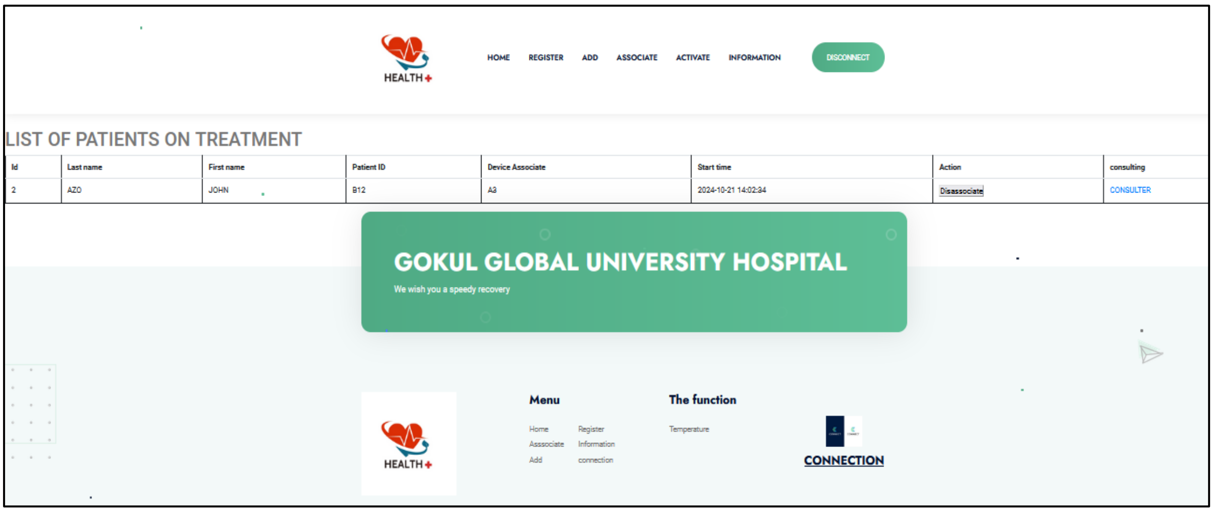


Figure 15: Patient registration page and patient associated device

Information page

This page displays patients whose treatment has been completed, and a consult link is available to view the results.

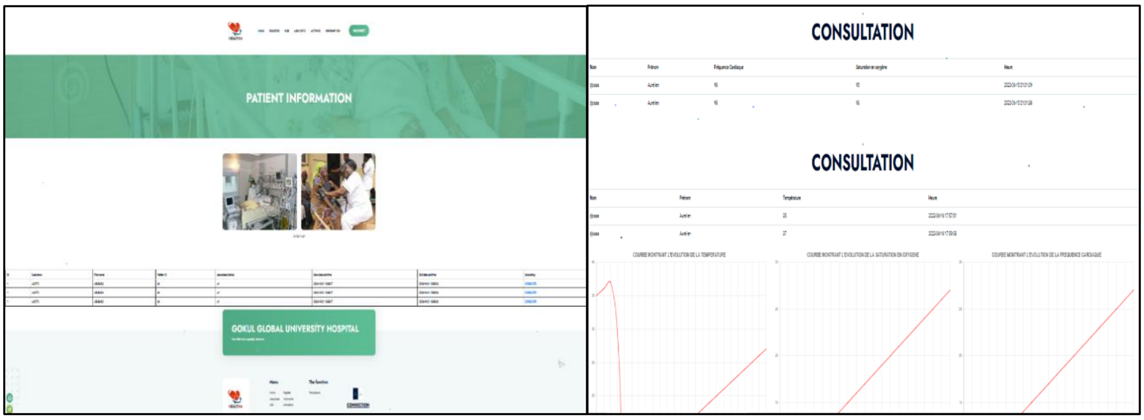


Figure 16 : Information page and consulting page

Activate page

This page allows administrators to activate and deactivate user accounts and assign profiles to users.

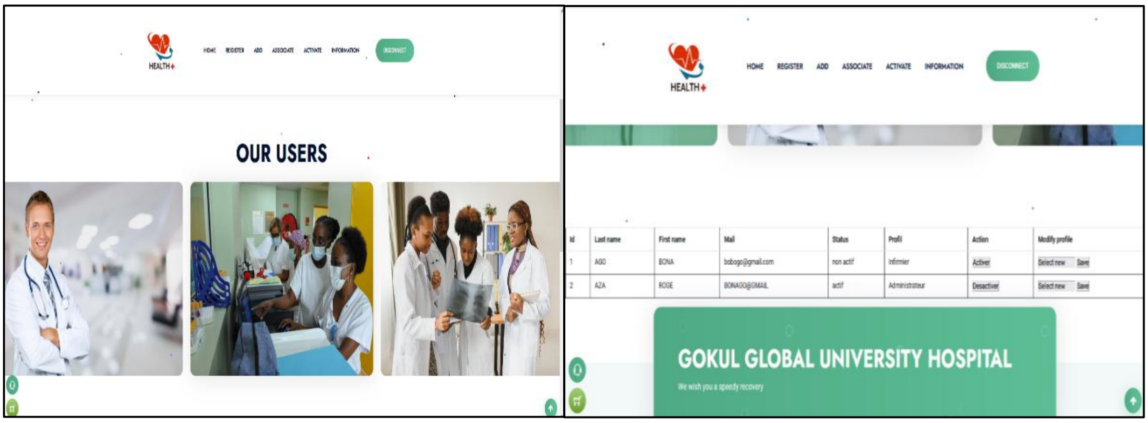


Figure 17 : Activate page

Discussion

This study presents the design and development of a remote temperature monitoring system using the ESP32 microcontroller. The system aims to address the growing need for accessible and affordable healthcare solutions, particularly in remote and underserved areas. As highlighted by Ravì et al. (2022) and Chatterjee &

Armenta (2021), remote patient monitoring (RPM) systems are a crucial component of the evolving landscape of Internet of Things (IoT)-enabled healthcare. The use of the ESP32, as emphasized by Zhu et al. (2021), provides a cost-effective and versatile platform for such applications due to its integrated Wi-Fi and Bluetooth capabilities.

The developed system focuses on real-time temperature monitoring, a vital sign often indicative of a patient's health status. This aligns with the broader trend in IoT-based health monitoring, as observed in the reviewed literature (Ghosh et al., 2018; Li, 2023; Naresh et al., 2024; Rusimamto et al., 2021), where real-time data acquisition and transmission are prioritized for timely interventions. The system's design incorporates the MLX90614 non-contact temperature sensor, enabling comfortable and hygienic temperature measurement. The data collected is then transmitted wirelessly, making it accessible to healthcare professionals via a web platform. This remote accessibility is a key advantage, echoing the benefits of remote monitoring discussed by Ghosh et al. (2018) and Hridhya (2023), enabling continuous patient evaluation without requiring physical presence.

The web platform, designed with user-friendliness in mind, allows for patient registration, device management, and data visualization. The use case, class, and sequence diagrams provide a clear representation of the platform's functionality and user interactions. The platform's features, including user authentication, patient data management, and device association/dissociation, are essential for a practical and secure RPM system. While the current system focuses on temperature monitoring, the modular design allows for future integration of other vital signs, such as heart rate and SpO2, as explored by Naresh et al. (2024) and Li (2023), creating a more comprehensive health monitoring solution.

A crucial consideration in any IoT-based healthcare system is data security. While this study acknowledges the importance of data privacy, the discussion of specific security measures is limited. As pointed out by Al-Mashhadani & Shujaa (2022) and Naresh et al. (2024), robust security measures, such as encryption and secure data storage, are paramount to protect sensitive patient information. Future iterations of this system should prioritize the implementation of such security protocols to ensure patient confidentiality and data integrity.

Furthermore, the study acknowledges the challenges related to internet dependency, a common limitation of many IoT-based systems (Ghosh et al., 2018; Chakraborty & Aithal, 2023; Hridhya, 2023). In remote areas with limited internet connectivity, the reliability of the system could be compromised. Exploring alternative communication methods, such as satellite communication or local data storage with periodic synchronization, could enhance the system's robustness in such environments.

Despite these limitations, the developed system demonstrates the potential of ESP32-based solutions for remote temperature monitoring. The focus on affordability and accessibility makes it particularly relevant for underserved communities. The system's real-time monitoring capabilities and alert system can significantly improve patient care, especially for elderly individuals and those with chronic conditions. Future research should focus on addressing the identified limitations, including enhancing security measures, exploring alternative communication strategies, and integrating additional vital signs to create a more comprehensive and reliable RPM system.

Conclusion

This study successfully designed and implemented a remote temperature monitoring system using the ESP32 microcontroller. The system offers a cost-effective and accessible solution for real-time patient temperature monitoring, particularly beneficial for individuals in remote areas and those requiring continuous observation. The developed web platform provides a user-friendly interface for healthcare professionals to access and manage patient data. The system's potential to improve healthcare delivery, especially for elderly patients and those with chronic conditions, is significant. While challenges related to data security and internet dependency remain, this research provides a solid foundation for future development. Future work will focus on enhancing security measures, exploring alternative communication strategies, integrating additional vital signs, and conducting clinical trials to evaluate the system's effectiveness in real-world healthcare settings. We believe that this system contributes to the growing field of IoT-enabled healthcare and has the potential to make a positive impact on patient care and healthcare accessibility.

Acknowledgement

The authors would like to express their sincere gratitude to Gokul Global University for providing the necessary resources and support for this research project. We also acknowledge the valuable contributions of Dr. Bhargav P. Majmundar and Saloni Thakar from computer Engineering Department at Gokul Global University for their mentorship.

Reference

1. Hobbi, A. H., Abdelmajid, I., & BEMMOUSSAT, C. E. (2020). *Conception et réalisation d'un prototype pour la surveillance à distance en temps réel de l'insuffisance cardiaque à base d'Arduino* (Doctoral dissertation).
2. Abderrahim, M. (2020, January). Capteurs connectés et intelligence artificielle pour la conception d'un prototype du poulailler domestique intelligent. Université Aboubakr Belkaïd-Tlemcen Faculté de Technologie

Département de Génie Electrique et Electronique (GEE).
3. Al-Mashhadani, M., & Shujaa, M. (2022). IoT security using AES encryption technology based ESP32 platform. *Int. Arab J. Inf. Technol.*, *19*(2), 214-223.
4. Benghozi, P. J., Bureau, S., & Massit-Folléa, F. (2015). *L'Internet des objets/The Internet of Things: Quels enjeux pour l'Europe?/What Challenges for Europe?*. Les Editions de la MSH.
5. Chakraborty, S., & Aithal, P. S. (2023). Let Us Create Multiple IoT Device Controller Using AWS, ESP32 And C. *International journal of applied engineering and management letters (IJAEML)*, *7*(2), 27-34.
6. Challal, Y. (2012). *Sécurité de l'Internet des Objets: vers une approche cognitive et systémique* (Doctoral dissertation, Université de Technologie de Compiègne).
7. Ghosh, D., Agrawal, A., Prakash, N., & Goyal, P. (2018, September). Smart saline level monitoring system using ESP32 and MQTT-S. In *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)* (pp. 1-5). IEEE.
8. Foltýnek, P., Babiuch, M., & Šuránek, P. (2019). Measurement and data processing from Internet of Things modules by dual-core application using ESP32 board. *Measurement and Control*, *52*(7-8), 970-984.
9. Ghosh, D., Agrawal, A., Prakash, N., & Goyal, P. (2018, September). Smart saline level monitoring system using ESP32 and MQTT-S. In *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)* (pp. 1-5). IEEE.
10. Amani, M. I. M. I. Suivi médicale à distance dans l'internet des objets.
11. Bobi, A. (2023). *Détection de nœuds centraux et de communautés dans les réseaux sociaux à deux modes* (Doctoral dissertation, Université du Québec en Outaouais).
12. Hridhya, A. P. (2023). IOT based health monitoring using ESP32 web server along with Android application. *International Journal of Emerging Technologies and Innovative Research, ISSN*, 2349-5162.
13. Li, S. (2023, August). IoT Healthcare System based on ESP32 for Smart Home. In *2023 IEEE International Conference on Mechatronics and Automation (ICMA)* (pp. 1768-1772). IEEE.
14. Naresh, S. R., Mariya, X. B., Ilamathi, S., & Tejus, J. L. (2024). An ESP32-Based Healthcare Monitoring System Integrating Multiple Sensors Using Integromat and Random Forest Algorithm. International Journal of New Innovations in Engineering and Technology, 24(1), 900–906.
15. Patel, K. K., Patel, S. M., & Scholar, P. (2016). Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, *6*(5).
16. Rusimamto, P. W., Endryansyah, L. A., Harimurti, R., & Anistyasari, Y. (2021). Implementation of arduino pro mini and ESP32 cam for temperature monitoring on automatic thermogun IoT-based. *Indones. J. Electr. Eng. Comput. Sci*, *23*(3), 1366-1375.
17. Lay, K. S., Li, L., & Okutsu, M. (2022). High altitude balloon testing of Arduino and environmental sensors for CubeSat prototype. *HardwareX*, *12*, e00329.
18. ElectronicsHub. (n.d.). Retrieved April 15, 2022, from https://www.electronicshub.org/esp32-oled-display/
19. Electronique-mag.com. (n.d.). Retrieved June 1, 2022, from http://www.electronique-mag.com/article255.html
20. Remini, B. (2022). SAHARA DUST NOURISHES THE SOIL-ROLE OF FLOODS IN THE FERTILIZATION OF PALM GROVES. *LARHYSS Journal P-ISSN 1112-3680/E-ISSN 2521-9782*, (51), 161-185.
21. bin Uzayr, S. (2022). *Bootstrap: The Ultimate Guide*. CRC Press.
22. Chidambaram, N., & Mazrae, P. R. (2022, May). Bot detection in GitHub repositories. In *Proceedings of the 19th International Conference on Mining Software Repositories* (pp. 726-728).
23. JQuery. (n.d.). Retrieved March 15, 2022, from https://fr.wikipedia.org/wiki/JQuery
24. Attache, I. N. E., & Badaoui, B. (2024). *Digitalizing the bank credit process through an Agile Project methodology* (Doctoral dissertation, Higher School Of Management and Digital Economy).
25. Maciuchová, N. (2024). Designing effective LinkedIn page for Company X: case Andament Oy.
26. MySQL. (n.d.). Retrieved April 8, 2022, from https://fr.wikipedia.org/wiki/MySQL
27. Sayers, E. W., Beck, J., Bolton, E. E., Brister, J. R., Chan, J., Comeau, D. C., ... & Sherry, S. T. (2024). Database resources of the national center for biotechnology information. *Nucleic Acids Research*, *52*(D1), D33.
28. Berge, F., Bechler, A., & Chardy, M. (2023, May). A Mixed Integer Programming Approach for 5G Spectrum Auction Strategy Optimization. In *ICC 2023-IEEE International Conference on Communications* (pp. 4846-4852). IEEE.
29. bin Uzayr, S. (2022). *PHP: The Ultimate Guide*. CRC Press.
30. Sqli-digital-experience.com. (n.d.). Retrieved February 14, 2024, from https://www.sqli-digital-experience.com/blog-fr/
31. Dizon, J. P., & Sonza, R. (2023). Project Monitoring with Decision Support System for the University Infrastructure Development

Office. *The Quest: Journal of Multidisciplinary Research and Development*, *2*(1).
32. Gyrard, A., & Boudaoud, K. (2022). Interdisciplinary iot and emotion knowledge graph-based recommendation system to boost mental health. *Applied Sciences*, *12*(19), 9712.
33. Automaton Sense. (n.d.). Retrieved February 14, 2024, from https://www.automationsense.com
34. Price, M. J. (2021). *C# 10 and. NET 6–Modern Cross-Platform Development: Build apps, websites, and services with ASP. NET Core 6, Blazor, and EF Core 6 using Visual Studio 2022 and Visual Studio Code*. Packt Publishing Ltd.
35. *Developer.mozilla.org*. (n.d.). Retrieved September 10, 2021, from https://developer.mozilla.org/fr/docs/Web/JavaScript