PCMJ

Progress in Computer and Mathematics Journal (PCMJ)
volume 3 [November, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

REAL-TIME PATIENT'S VITAL SIGN MONITORING USING ESP32

NUR FARAHDINNA ANUAR
College of Computing, Informatics and Mathematics
UiTM Melaka, Campus Jasin, Melaka
Farahdinna91@gmail.com

MOHAMMAD HAFIDZ RAHMAT*
College of Computing, Informatics and Mathematics
UiTM Melaka, Campus Jasin, Melaka

MOHD HAFIFI MOHD SUPIR
College of Computing, Informatics and Mathematics
UiTM Melaka, Campus Jasin, Melaka
hafifisupir@uitm.edu.my

Article Info

Abstract

The IoT technology in remote patient monitoring (RPM) allows healthcare enhancement through live vital sign monitoring. A real-time patient monitoring system based on ESP32 platform and ThingSpeak platform and Telegram bots and mobile application is introduced in this research. The system operates continuously to acquire heart rate data and SpO2 readings and temperature measurements after which the data gets sent to the cloud for both analysis and visualization. The mobile app shows current healthcare data immediately while Telegram bots send out immediate notice updates. The system operates effectively based on functionality and network performance verification results. Upcoming enhancements in the system involve GPS tracking capabilities along with improved encryption measures and extra sensors for expanding healthcare monitoring features.

Received: March 2025
Accepted: September 2025
Available Online: November 2025

Keywords: IoT, Remote Patient Monitoring, ESP32, ThingSpeak, Mobile Application, Telegram Bot

INTRODUCTION

The healthcare solution Remote Patient Monitoring (RPM) combines Internet of Things (IoT) technology to offer ongoing health monitoring services for patients. RPM technologies help medical staff manage chronic diseases more effectively while decreasing hospital patient volume through remote patient monitoring. IoT-based RPM systems establish a connection between medical sensors and microcontrollers and cloud platforms to acquire and analyze patient data in real time.

The project develops a genuine patient tracking system which utilizes ESP32 as a low-price wireless microcontroller. The system depends on vital sign sensors to monitor patient readings for heart rate and SpO2 and temperature measurements. Stored data goes to ThingSpeak for both analysis and transmission occurs through Telegram bot platforms.

This research has three essential aims to build a fully operational RPM system which integrates both mobile and cloud technologies followed by network and functionality testing stages. The system functions to enhance medical care accessibility through continuous patient monitoring parameters.

LITERATURE REVIEW

The analysis of existing research allows healthcare professionals to understand past studies regarding remote patient monitoring (RPM) and the creation of technical infrastructure in this field. The literature review enables teams to fill knowledge gaps and prove project importance through analysis of IoT-based healthcare, mobile application and cloud integration and network security research.

Remote Patient Monitoring (RPM)

RPM systems through IoT technology enable remote vital signal tracking that decreases hospital requirements for patients while improving healthcare operational effectiveness (Kadari et al., 2023). Healthcare providers receive timely intervention through wearable sensors which automatically transmit temperature as well as heart rate and blood oxygen level data continuously (Islam et al., 2023). Medical facilities utilizing effective RPM systems build better patient compliance while reducing healthcare expenses and delivering superior clinical results (Mukhopadhyay et al., 2024).

Mobile Application Integration

The crucial task of RPM relies on mobile applications because these systems show real-time data visualization as well as deliver remote accessibility to health records according to (Saleh et al., 2023). The healthcare application development uses Android Studio mainly because of its ability to work with numerous sensors together with its adaptable nature (Olabode et al., 2020). Timely alerts and enhanced health condition management by patients result from the deployment of mHealth applications (Koh et al., 2022).

ThingSpeak Cloud Integration

The cloud platform ThingSpeak functions as a platform to accumulate and study IoT-based patient health information according to (Irakomeye, 2022). Through the platform users gain immediate visibility of health metrics and telemedical access for medical professionals who need this for their decisions(Andrioaia et al., 2021). The data encryption protocols secure patient information transmission which ensures both privacy protection and data reliability (Pauzi & Hasan, 2020).

Security and Network Performance

Reliable network performance as well as data security needs to be established by IoT-based RPM systems. Bandwidth, latency, and reliability impact real-time monitoring efficiency (Mohd Takiyuddin et al., 2022). The combination of cloud communication with edge computing speeds up medical data processing which results in faster responses for critical medical situations (Hassan et al., 2020). Medical data encryption as a strategy helps limit unauthorized entry to patient information and maintains their privacy as reported by Vandebriel & Loveren (2019).

METHODOLOGY

The methodology section outlines the systematic approach used in developing the Remote Patient Monitoring System. It describes each phase of the project, including requirement analysis, system design, development, testing, and documentation. By following a structured methodology, the project ensures reliability, efficiency, and effectiveness in real-time health monitoring. The development of this system follows the Waterfall methodology, a structured approach that ensures systematic progress through each phase.

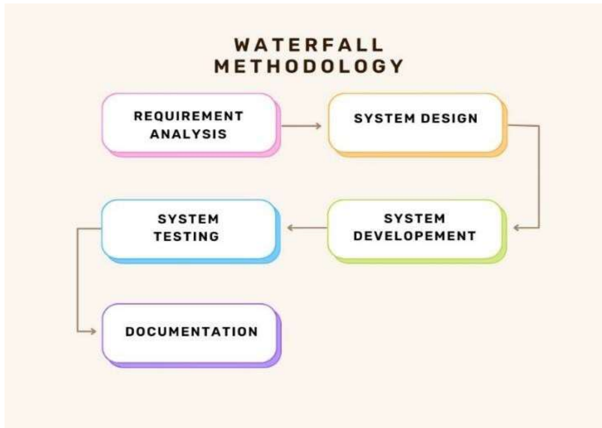
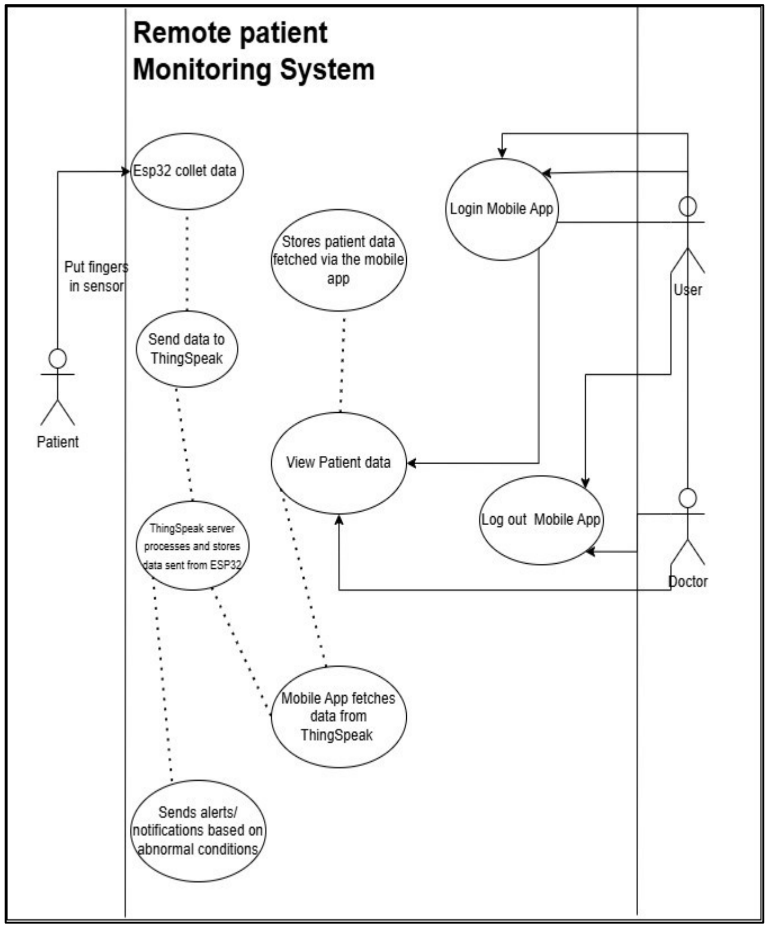


Figure 1 Methodology Framework

Requirement Analysis

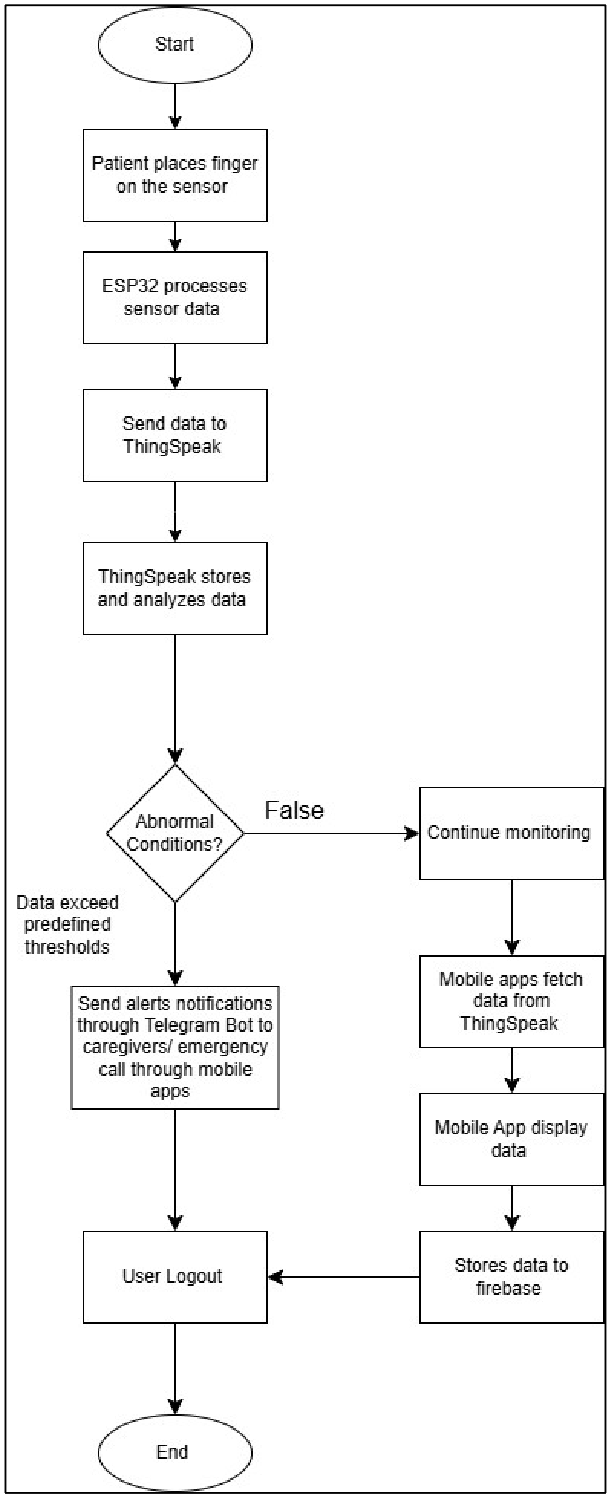
The requirements are developed based on a study of previous literature which consist of various research papers, journals, and technical documents and the study of the difficulties faced by the stakeholders such as the medical professionals. From these inputs, the project objectives were defined using the ESP32 microcontroller for real-time health data collection, Firebase for efficient storage and retrieval of data and a mobile application for monitoring health data through a smartphone. These components are combined in the solution, so the patient can track the vital signs of patients through the mobile app in real-time, and Firebase also takes care of data storage, making sure it is safe. This approach gives solutions to flexibility, healthcare issues checkup early enough and back up of information which are important to both patients and doctors. A use case diagram is a valuable tool in system development because it demonstratesthe interactions between users (actors) and the system. This diagram helps in establishing a system's functional needs by illustrating the different ways in which users interact with it to achieve certain objectives. The diagram displays interrelations between the system actors (Patient, User, and Doctor) while showcasing essential system operations.

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 3 [November, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj



System Design

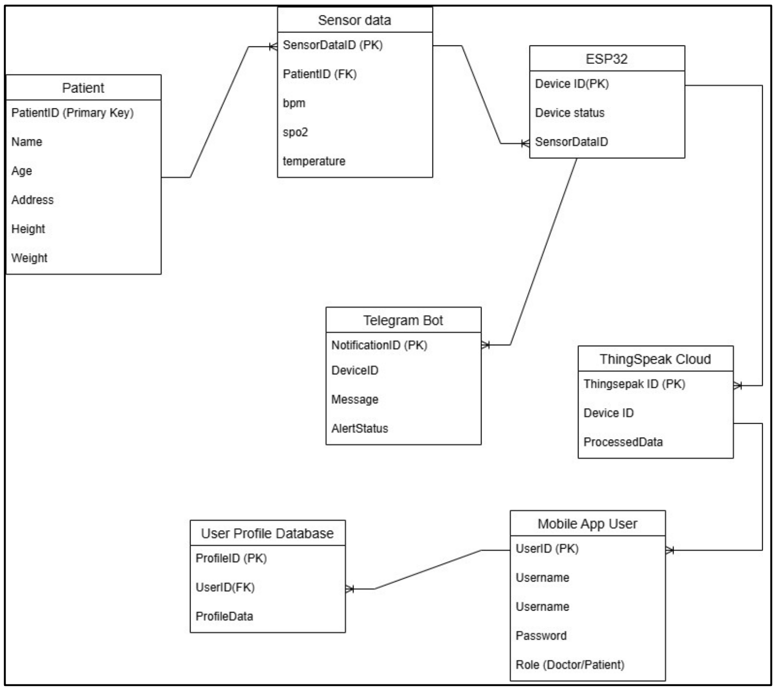
In this project, the design phase will have a flowchart of the system, Entity relationships diagram (ERD), User Interfaces, logical design. For the flowchart they will be one flowchart. For the ERD in this design phase about how the relationship between entities in remote patient monitoringsystems. At user interfaces show how the interfaces of the Mobile Application. Lastly,logical design shows the hardware will be use in this system. In system design flowchart provides an outline of how the system should work before the development process begins. Below show Figure 3.4 shows the flowchart of the system. The Remote Patient Monitoring System process appears as a sequential graphic format in your diagram. A patient initiates data collection through sensor contact with their finger. The ESP32 microcontroller functions as a processor of acquired vital signs data which it sends to the ThingSpeak IoT cloud platform. The IoT cloud platform ThingSpeak holds and processes the stored data for analytical purposes.



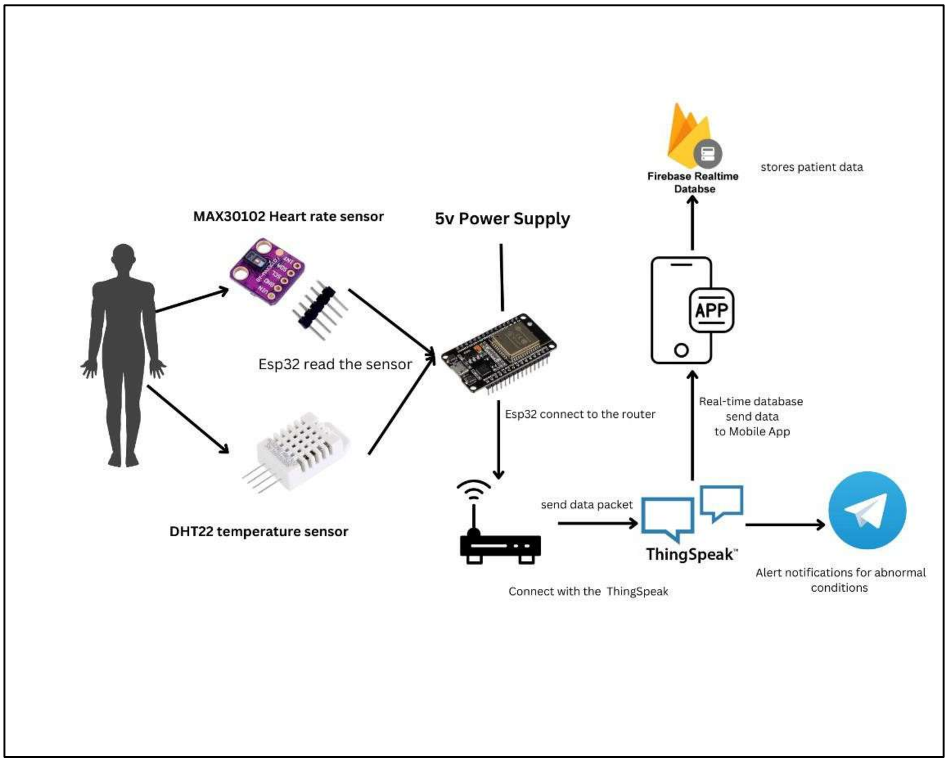
An Entity Relationship Diagram (ERD) for your Remote Patient Monitoring System demonstrates the connections between significant entities which the system includes. The Patient entity represents the diagram's central component while linking to different system

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 3 [November, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

components. Each patient maintains individual attributes including their identification number and personal and medical history details. All health data from patients goes into the Sensor Data entity that records vital measurements of heart rate alongside SpO2 level and temperature readings and timestamped data points. Hospital data entry incorporates one patient into multiple records through a one-to-many connection.



Next, the system architecture of a remote patient monitoring system using ESP32 consists of several interconnected components. The ESP32 microcontroller, connected to various sensors, captures real-time health data from patients, including heart rate, SpO2, and body temperature. This data is transmitted via Wi-Fi to ThingSpeak, which acts as the cloud backend. The mobile application provides an intuitive interface for caregivers and doctors, displaying live health metrics, sending notifications for critical conditions, and offering access to historical health records. This integrated system ensures continuous monitoring, timely alerts, and centralized data management to enhance patient care. The Remote Patient Monitoring System uses systematic integration of different components to supply live health tracking capabilities. The system employs MAX30102 heart rate sensors together with DHT22 temperature sensors to collect essential patient signs including heart rate and body temperature. The ESP32 microcontroller processes information from sensor interfaces it receives through Wi-Fi to transfer data to a connected router for communication.



RESULT AND DISCUSSION

The evaluation of ESP32-ThingSpeak connection performance together with mobile application-network stability was the main objective of the network testing. Ping test results demonstrate reliable data transmission since they present steady connectivity without losing packets. Testing demonstrated that connection times between ESP32 and ThingSpeak (192.168.0.128) reached 34ms as their minimum value while reaching 216ms as their maximum metric and averaging 84ms in total. Latency spikes occur sporadically while the system maintains a stable link status. A range of latencies from 47ms to 253ms at an average of 123ms was observed when the mobile application connected to the network through 192.168.0.117. Data retrieval operations experience less than substantial pauses represented through the average latency while maintaining a stable connection. The real-time transmission system demonstrates efficient operation through results which enable effective remote

Progress in Computer and Mathematics Journal (PCMJ)
volume 3 [November, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

monitoring of patient vital signs. The system requires upcoming enhancements to stabilize networks while minimizing latency variations for improved user satisfaction.

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>ping 192.168.0.128

Pinging 192.168.0.128 with 32 bytes of data:
Reply from 192.168.0.128: bytes=32 time=216ms TTL=64
Reply from 192.168.0.128: bytes=32 time=34ms TTL=64
Reply from 192.168.0.128: bytes=32 time=34ms TTL=64
Reply from 192.168.0.128: bytes=32 time=53ms TTL=64

Ping statistics for 192.168.0.128:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 34ms, Maximum = 216ms, Average = 84ms

C:\Users\User>
```

Figure 2 ESP32 ping connectivity

```
C:\Users\User>ping 192.168.0.117

Pinging 192.168.0.117 with 32 bytes of data:
Reply from 192.168.0.117: bytes=32 time=146ms TTL=64
Reply from 192.168.0.117: bytes=32 time=253ms TTL=64
Reply from 192.168.0.117: bytes=32 time=49ms TTL=64
Reply from 192.168.0.117: bytes=32 time=47ms TTL=64

Ping statistics for 192.168.0.117:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 47ms, Maximum = 253ms, Average = 123ms
```

Figure 3 Result ping mobile application

The mobile application profiling session evaluated performance metrics through analysis of CPU and memory use while the application communicated over the network. The profiler displays a network activity and transition report that detects operations on a Samsung SM-A750GN device. A user interface tap at 00:41.159 resulted in 118.32 milliseconds of application duration that caused CPU utilization to reach 100 percent temporarily probably because of data activity or graphical interface modifications. The application ran efficiently by using a consistent 97MB of memory throughout the session. Periods during which activities are stopped and destroyed by the system are logged in the session data because these state changes might affect network performance. Data demonstrates that the application works with network connections efficiently because resource consumption stays within normal thresholds. Advancing both CPU spike optimization and network request speed reduction will lead to better responsiveness together with enhanced performance.

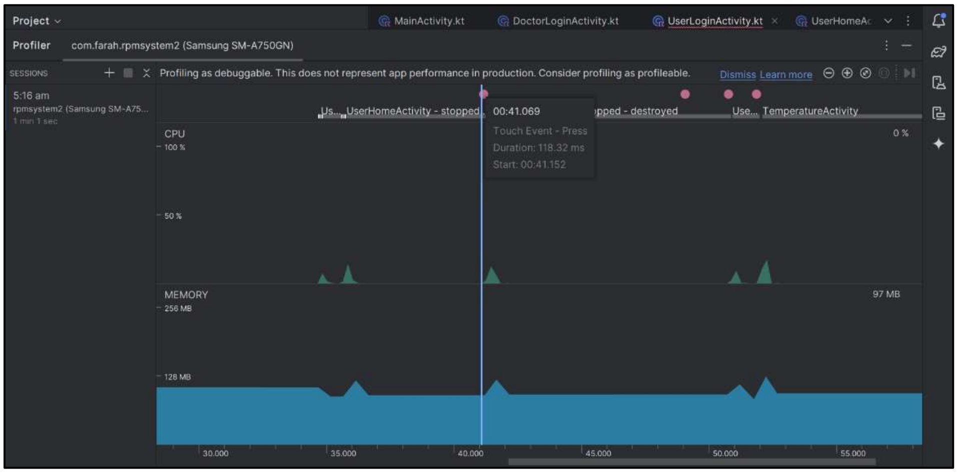


Figure 4 Profiling session of mobile application

CONCLUSION

Real-time Patient's Vital Sign Monitoring System using ESP32, Firebase and a mobile application does fulfill the objectives of this project by offering an actual time solution to monitor the patients with chronic diseases. Connecting ESP32 microcontroller with enhanced sensors enable the system to safely and effectively capture vital signs like the heart rate, SpO2, and body temperature and then upload the data onto Firebase for secure storage and recording. The use of this mobile application ensures a supplement to the system that makes it easier for doctors and caregivers to view patients' health in real-time. It is established that the system contributes to the early identification of essential health situations in addition to improving data accessibility, accuracy and centralization hence minimizing the dangers of constant split and inaccurate information from manual monitoring systems.

The impact of this project is therefore given by its potential to change the way healthcare is practiced by making remote monitoring possible and realizable. It enables patients with chronic diseases manage their own condition and allows doctors to manage their patient regardless of distance. Utilizing IoT technology, the proposed system improves efficiency and functionality of the overall health monitoring while reducing costs and risks associated with traditional health monitoring devices. The achievement of its performance parameters towards parameters such as data correctness, response time and usability suggest that it has potential

Progress in Computer and Mathematics Journal (PCMJ)
volume 3 [November, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

for use as a base on which future developments may be built. Not only does this project help an individual's health in particular but it also serves as the foundation for enabling these applications in telemedicine and worldwide health services.

REFERENCES

Andrioaia, D.-A., Culea, G., & Puiu, P.-G. (2021). ENVIRONMENTAL TEMPERATURE AND HUMIDITY MONITORING SYSTEM USING RASPBERRY PI 4 AND THINGSPEACK. In *Journal of Engineering Studies and Research* (Vol. 27, Issue 3).

Hassan, S. R., Ahmad, I., Ahmad, S., Alfaify, A., & Shafiq, M. (2020). Remote pain monitoring using fog computing for e-healthcare: An efficient architecture. *Sensors (Switzerland)*, *20*(22), 1–21. https://doi.org/10.3390/s20226574

Irakomeye. (2022). *IoMT BASED POSTOPERATIVE PATIENT MONITORING VITAL SIGNS USING WIRELESS BODY SENSOR IN BURUNDI: A CASE STUDY OF VAN NORMAN CLINIQUE* [Nelson Mandela]. https://doi.org/10.58694/20.500.12479/2135

Islam, M. R., Kabir, M. M., Mridha, M. F., Alfarhood, S., Safran, M., & Che, D. (2023). Deep Learning-Based IoT System for Remote Monitoring and Early Detection of Health Issues in Real-Time. *Sensors*, *23*(11). https://doi.org/10.3390/s23115204

Koh, K. S., Lim, H. S., Lim, J., Wei, Y., Minn, P. W., & Wong, J. (2022). Development and implementation of a national mobile health application: A case study from Brunei. *Journal of Global Health*, *12*. https://doi.org/10.7189/JOGH.12.03083

Mohd Takiyuddin, M. A., Abd Halim, I. H., Razak, T. R., & Mohd Fuzi, M. F. (2022). Remote Patient Monitoring System For Covid-19 Patient Using Wi-Fi-Based Pulse Oximeter Reading Sensor. *Journal of Computing Research and Innovation*, *7*(2), 101–110. https://doi.org/10.24191/jcrinn.v7i2.290

Mukhopadhyay, A., Remanidevi Devidas, A., Rangan, V. P., & Ramesh, M. V. (2024). A QoS-Aware IoT Edge Network for Mobile Telemedicine Enabling In-Transit Monitoring of Emergency Patients. *Future Internet*, *16*(2). https://doi.org/10.3390/fi16020052

Olabode, O., Daramola, O., & Akinbo, R. (2020). Mobile Application for Monitoring and Management of Out Patients. *International Journal of Scientific and Research Publications (IJSRP)*, *10*(3), p9931. https://doi.org/10.29322/ijsrp.10.03.2020.p9931

Pauzi, A. F., & Hasan, M. Z. (2020). Development of IoT Based Weather Reporting System. *IOP Conference Series: Materials Science and Engineering*, *917*(1). https://doi.org/10.1088/1757-899X/917/1/012032

Saleh, S., Cherradi, B., Gannour, O. El, Gouiza, N., & Bouattane, O. (2023). Healthcare monitoring system for automatic database management using mobile application in IoT environment. *Bulletin of Electrical Engineering and Informatics*, *12*(2), 1055–1068. https://doi.org/10.11591/eei.v12i2.4282