Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 6, No. 3, August 2021, Pp. 203-210

Implementation of oxygen saturation sensor data acquisition based on bluetooth low energy protocol

Muhammad Fahmi Ali Fikri*[1], Dany Primanita Kartikasari[2], Adhitya Bhawiyuga[3]
Brawijaya University, Indonesia[1,2,3]

Article Info

Keywords:
Internet of Things, Data Acquisition, Bluetooth Low Energy, Store and Forward, Data Redundancy

Article history:
Received: July 22, 2021
Accepted: August 18, 2021
Published: August 31, 2021

Cite:
Fikri, M. F. A., Kartikasari, D. P., & Bhawiyuga, A. (2021). Implementation of Oxygen Saturation Sensor Data Acquisition Based on Bluetooth Low Energy Protocol. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(3). https://doi.org/10.22219/kinetik.v6i3.1305

*Corresponding author
Muhammad Fahmi Ali Fikri
E-mail address:
mfahmialif@student.ub.ac.id

Abstract

The oxygen saturation sensor data acquisition system is implemented on the Android platform used to obtain sensor data from IoT devices that provide oxygen saturation data. We can use the Bluetooth Low Energy (BLE) protocol to acquire sensor data from IoT devices. This data acquisition aimed to process further data, which data will later send to the server. Bluetooth Low Energy (BLE) has an architecture consisting of sensors, gateways, and data centers. However, this architecture has weaknesses such as failure to send data to the data center because it is not connected to the internet network and data redundancy. The proposed solution to solve this problem is to create a system that can acquire sensor data using the Bluetooth Low Energy (BLE) protocol using a store and forward mechanism and checking data redundancy. The proposed system will be implemented using sensors from IoT devices, the gateway used is Android devices, and using the Bluetooth Low Energy protocol to acquire data from sensors. Then the system will send the data to the cloud or server. The test results give the results of the system is successfully implemented, and Android can connect IoT devices to the gateway with a maximum distance of 10 meters. Then, when the system stores, there is an increase in data of 4 kb for every minute. After that, there is no data redundancy in the system.

1. Introduction

With the development of the times, there have been many emerging technologies based on the Internet of Things (IoT). IoT is a physical object that allows for communication between objects to share information [1]. With the information obtained from sensors installed in IoT devices, IoT has several components, including identification, sensing, communication, computation, services, and semantics [1]. This research will focus more on the communication component, namely the use of data communication protocols.

Examples of the application of IoT can be found in various fields such as agriculture, smart city, smart home, and healthcare [1]. In IoT healthcare, one of the studies that have involved IoT healthcare is the application of monitoring systems, such as in Massimilino Donati's research [2] and analysis from Iuliu Alexandru Pap [3]. In this study, the sensors used were body temperature, blood pressure, and oxygen saturation sensors. Monitoring is an activity to collect data as a decision-maker [4]. Apart from being used for monitoring systems, IoT healthcare can also be used for hospital purposes, as in the research conducted by Aditi S. Patade [5].

IoT Healthcare used for the monitoring system helps monitor the patient's condition based on IoT devices [6]. When creating a system for patient monitoring, the system requires data obtained from IoT devices that produce data [7]. The process to get data from the IoT device is called data acquisition [3]. It is feasible to obtain sensor data and process it as needed [8]. Retrieval of data acquired in IoT Healthcare can be in real-time time series based on the system's needs [9]. Data retrieval in real-time time-series means that data will retrieve data directly with the actual time and with a predetermined time, such as data taken every one second [9]. The data that has been collected can also be sent to the cloud or server as an online data storage area [2].

One of the motivations of this research is the use of IoT devices in the health sector, which is applied to obtain one of the data from health parameters to measure patients infected with the COVID-19 virus. The COVID-19 virus is a respiratory disease caused by the respiratory syndrome coronavirus-2 (SARS-CoV-2), which attacks the respiratory system, causing sufferers to experience fever, cough, headache, and shortness of breath [10]. Patients with COVID-19 have abnormal health parameter values, such as body temperature, heart rate, respiration, and oxygen saturation [11].

The reason for choosing oxygen saturation data which is the focus of the research is also because of the motivation in this research. Oxygen saturation data is needed in determining whether or not a patient is infected with the COVID-19 virus. Not only with the required oxygen saturation data, but pulse rate also has a relationship with oxygen saturation as a determinant of patient status. Oxygen saturation and pulse rate data can be obtained from one IoT device which is also why the authors choose oxygen saturation data as the focus of the research. In this research, the IoT device used is Jumper JPD500G Pulse Oximeter.

IoT Healthcare that is used to acquire sensor data from IoT devices will require a media or data communication protocol and a gateway as a device for data acquisition [12]. One of the media that can be used is a cable connected from the IoT device to the gateway [3]. Then the use of ZigBee can also be used for data acquisition [13]. In addition, the Bluetooth Low Energy (BLE) protocol can also be used as a data acquisition communication protocol [2]. The Bluetooth Low Energy (BLE) protocol is suitable for monitoring systems because it is designed by considering the characteristics of low resource usage and small coverage area compared to other communication protocols [14].

Several researchers have carried out several studies related to utilizing IoT in sensor data acquisition. The research entitled "IoT-Based eHealth Data Acquisition System" by Iuliu Alexandru Pap explains the data acquisition system with sensors used is sensors obtained from IoT devices. The gateway used for data acquisition is the Raspberry Pi 3, and the data will be sent to the server with WebSocket in real-time [3]. Then in his research Massimiliano Donati entitled "A Telemedicine Service Platform Exploiting BT/BLE Wearable Sensors for Remote Monitoring of Chronic Patients". This study describes the acquisition of a system with data sources from IoT devices, and the gateway used is an android device [2]. Then also explained the use of Bluetooth as a communication protocol. Then the research conducted by Li Min with the title "Design of Multi-network Data Acquisition System Based on Cloud Platform" explains that data acquisition systems can be made possible by using several different types of networks, namely Bluetooth, Zigbee, and RF which then the data is sent to cloud or server [13].

Some of these studies show that BLE as a communication protocol in the monitoring system has an architecture consisting of sensors as data sources, gateways, and servers. However, the BLE infrastructure used in the study has weaknesses when sending data to the server or cloud. Suppose data is sent to a server or cloud that involves an internet connection. In that case, it will result in several obstacles, namely an internet connection that suddenly disconnects, so that data cannot be sent to the server [15]. And the data can also experience data redundancy or the presence of the same data as the data that has just been sent [16].

Handling the weakness of sending data to the server can be solved by doing store and forward [15]. When sending data to the server fails, the system will save the data to local storage, which can be called a store. Then when the internet connection is normal, the system will send the data to a server named forward. For data redundancy problems, the system can do it by checking the data on the server. If the data sent is on the server, the system will not send the data to the server [16].

From several existing problems and solutions, this research will focus on discussing data acquisition from the oxygen saturation sensor by utilizing the Bluetooth Low Energy (BLE) protocol, which considers the failure of sending data sent to the server. The sensors used are sensors from IoT devices, and the gateway used in this study is an android device. An android device is a smartphone with a Linux-based operating system that has dominated more than 75% of the total market share, and there are competitors from iOS, Windows Mobile OS, and Blackberry [17]. The android device is connected to the BLE of the IoT device for data acquisition so that system can send the sensor data to the cloud or server.

2. Research Method

The research method of this study begins with conducting a literature study to obtain all the information needed to conduct research. Then analyze the system requirements required to build a system that can solve the problems in research. The next step is to design a system based on system requirements. After that, carry out the implementation regarding the system design. After the system has been implemented, it is continued with system testing to test the results of the implemented system and obtain conclusions and suggestions from the research.

The system used to solve research problems is a system that can acquire data from the oxygen saturation sensor using the BLE protocol. An oxygen saturation sensor or pulse oximeter is a sensor that can be used to get data on oxygen saturation (SPO2) and user pulse (BPM) [3]. Then oxygen saturation is the level of blood oxygen concentration in the blood [18]. Then the system to be built will be implemented on android devices.

The system to be implemented will acquire data from IoT devices, and then the system will send the received data to the server, sending it to the server using an API. The system is equipped with Bluetooth auto-connect to the device last installed in the system, a store, and forward mechanism when there is no internet connection, and data redundancy checking so that data redundancy does not occur on the server. For the system topology of this system, referred to Figure 1.

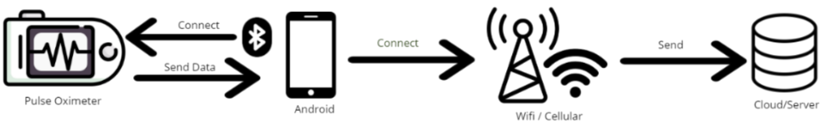

Figure 1. System Topology

The explanation of Figure 1 regarding the topology of this system is as follows, the topology of this system starts from connecting Android with Pulse Oximeter via the BLE protocol. Android will acquire the data, and Pulse Oximeter will send the data to Android. The data will be sent to the cloud by Android using a WiFi/Cellular network.

2.1 Hardware Design

The hardware used for the research is an IoT device that can provide oxygen saturation data and is equipped with Bluetooth. The device is a Pulse Oximeter. In this study, the IoT device used is the JPD500G Pulse Oximeter Jumper. The device can be seen in Figure 2.

Figure 2. Jumper JPD500G Pulse Oximeter

In Figure 2, we can see the IoT devices used in this study. The IoT device will provide the oxygen saturation data needed by the system by connecting the IoT device with Android.

2.2 Sensor Data Acquisition Method

Sensor data acquisition is implemented using the Bluetooth Low Energy (BLE) protocol. Bluetooth Low Energy (BLE) is a low-power communication protocol that has an essential role in IoT. BLE is required to meet QoS (Quality of Service) in its use, which is usually related to latency performance [19]. Because BLE has low power in its application, this BLE is suitable for use on IoT devices [20]. We can reference the communication architecture of the sensor data acquisition in Figure 3.

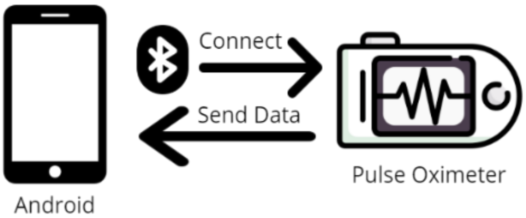

Figure 3. Sensor Data Acquisition Communication Architecture

Based on Figure 3, the data acquisition communication architecture involves Android and Pulse Oximeter connected with Bluetooth Low Energy. The system will implement this sensor data acquisition at the gateway, namely the android device using the Android Studio tools. Android studio is a tool used to create android-based applications. In its implementation, Android Studio uses the Java and Kotlin programming languages [21]. However, in this study, the programming language used in Kotlin has the advantage of writing less or shorter code than Java and has several modules to make it easier to implement [22].

A function or module is needed to operate GATT which is used to obtain sensor data using the BLE protocol. GATT is the format of the service and the characteristics and procedures of the interface of several attributes such as read, write, discovery, notification, and indication [23]. In GATT, there is a service that contains a collection of attributes or characteristics [23]. Because the data has notification characteristics, to perform data acquisition, it is necessary to enable notification to IoT devices to receive data from IoT devices and disable notifications to stop receiving data from IoT devices. We can reference the data structure in Table 1.

Table 1. Data Structure of Data Acquisition

No	Data	Data Type	Description
1.	Pulse (SPO2)	Double	Oxygen saturation data
2.	Bpm	Double	Pulse data

Cite: Fikri, M. F. A., Kartikasari, D. P., & Bhawiyuga, A. (2021). Implementation of Oxygen Saturation Sensor Data Acquisition Based on Bluetooth Low Energy Protocol. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(3). https://doi.org/10.22219/kinetik.v6i3.1305

2.3 Sending Data to Server

Sending data to the server is implemented using a prepared API. API or Application Programming Interface is an interface used to connect applications between platforms. API is not a database, but an API is a code that manages access points to the server so that it is possible to access the database through the API [24]. The API address refers to the pmeter.my.id website used as a server for data storage. To use the API in the android studio, we can use the AndroidNetworking module. AndroidNetworking is equipped with several conditions when the API is run, such as when the API is successful and when it fails to run. If the API is executed successfully, the API will continue on the success function by returning a response from the server. Meanwhile, if the API fails to run, then the API will continue to the error function by returning an error message. To see the communication architecture of data transmission, see Figure 4.

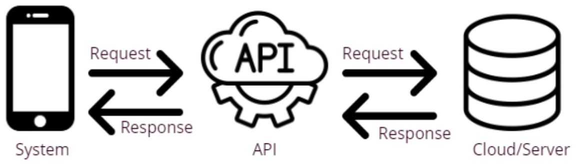

Figure 4. Data Delivery Communication Architecture to Server

Based on Figure 4, the flow of data transmission starts from the system requesting the API, and then the API will forward the request to the cloud/server. The database will respond to the system which is passed through the API. Then the API will provide a response to the system in the form of data delivery status. The data sent can be referred to in Table 2.

Table 2. Data Structure of Data Delivery

No	Data	Data Type	Description
1.	Officer	Varchar (100)	Data on the name of the officer who acquired the data
2.	Name	Varchar (100)	The name of the patient whose data was acquired
3.	Pulse	Double	Oxygen saturation data
4.	Bpm	Double	Pulse data
5.	Datetime	Datetime	Data retrieval time

2.4 Store and Forward Mechanism

Store and forward is implemented by designing a system that can store and forward when under certain conditions. If a problem occurs during data transmission, such as the smartphone does not have an internet network which causes the data to fail to be sent to the server, it will be stored or saved to local storage. When the internet connection is available again, the data will be forwarded or forwarded to the server. The limitation of this store and forward mechanism is that it is only limited when there is an error on the internet network caused by unavailable data packets or unavailable WiFi with a normal internet connection. The communication architecture of store and forward can be referred to in Figure 5.

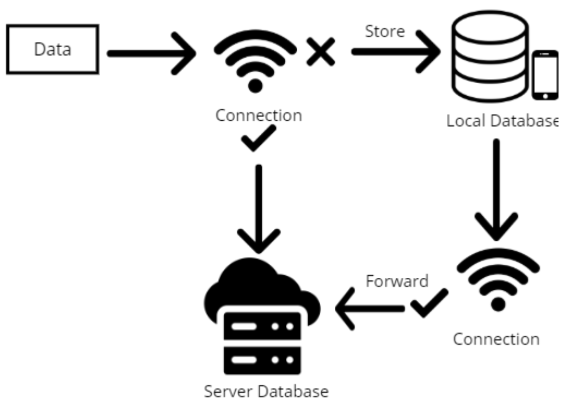

Figure 5. Store and Forward Communication Architecture

Based on Figure 5, the data is sent to the database server with the system checking the internet connection first. If the internet connection is unavailable or lost, the system will store the data in the local database. Then if the internet connection is available again, the data will be sent to the database server.

For the implementation of local data storage in the android studio, we can use SQLite. SQLite is a local datastore on the android studio that uses a local application database [25]. SQLite is like a database in general that can use select, insert, update and delete operations. The system will implement the forward mechanism by utilizing the android studio broadcast receiver, which is used to determine the condition of the internet connection. If the internet connection changes again to become available, then the system will send the data in the local database to the database server.

2.5 Data Redundancy Mechanism

Data redundancy is implemented with the system checking the data before the data is sent to the server. Checking is done by utilizing database operations to speed up data search results. The database operations used are SELECT and WHERE, because by using these operations, data will be obtained that has the same WHERE conditions as the parameters of the data to be stored on the server. Activity diagram for Data Redundancy can be reffered to in Figure 6.

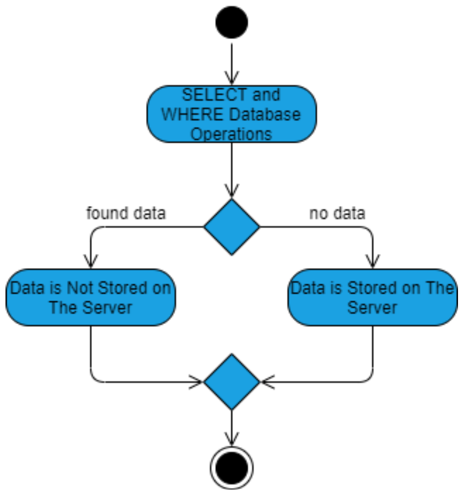

Figure 6. Data Redundancy Activity Diagram

Based on Figure 6, checking data redundancy starts from checking data redundancy with database operations. Database operations used are SELECT operations with additional WHERE to search data. The redundancy checking mechanism in this study will check all data values that will be stored on the server, so to check it can use SELECT and WHERE because it will issue results according to WHERE conditions, namely all parameters of the data to be stored in the database as shown in Table 2. When the same data is found during the search, the data will not be stored on the server. However, when the same data is not found, the data will be stored on the server. After checking data redundancy with the condition of data from local storage, the system will delete the data from local storage.

3. Results and Discussion

3.1 Data Acquisition Results

The data acquisition system using the BLE protocol has been successfully implemented on Android devices. To be able to find out the results of the data acquisition, a system test will be carried out aimed at checking the success of the data and the suitability of the data from the data acquisition results with the existing data on the IoT device. The test results can be referred to in Figure 7.

Figure 7. Test Results Display Sensor Data

Cite: Fikri, M. F. A., Kartikasari, D. P., & Bhawiyuga, A. (2021). Implementation of Oxygen Saturation Sensor Data Acquisition Based on Bluetooth Low Energy Protocol. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(3). https://doi.org/10.22219/kinetik.v6i3.1305

In Figure 7, it can be referred that the data displayed in the application has the same data as the data on the Bluetooth Jumper device, namely the SPO2 data has a value of 99 and BPM has a value of 98. The acquired data has a one-second delay for each data. With the similarity of the data, the test to display data from the Bluetooth device to the system or application was successfully carried out.

3.2 Performance Testing of Data Acquisition System Based on Distance

Testing the performance of the data acquisition system based on the distance of the gateway with the IoT device aims to determine the maximum distance between the IoT device and the gateway that can be connected. This test is carried out in an environment where there are connection obstacles such as a house wall. Testing is done by connecting the IoT device with the gateway first. Then perform a distance test starting from a distance of one meter until the IoT device is disconnected from the gateway. The test results can be referred to in Table 3.

Table 3. Results of Data Acquisition System Based on Distance

No.	Distance (meter)	Measurement RSSI (-dBM)	IoT Connected Status with Gateway	Data Acquisition
1.	1	63	Connected	Done successfully
2.	2	67	Connected	Done successfully
3.	3	74	Connected	Done successfully
4.	4	78	Connected	Done successfully
5.	5	88	Connected	Done successfully
6.	6	91	Connected	Done successfully
7.	7	94	Connected	Done successfully
8.	8	95	Connected	Done successfully
9.	9	97	Connected	Done successfully
10.	10	99	Connected	Done successfully
11.	11	-	Connection Lost	Failed

Based on Table 3, the results of testing the performance of the data acquisition system based on the yield distance at a distance of 1 meter – 10 meters, the system succeeded in acquiring data from the IoT device, but at a distance of 11 meters, the system failed to acquire the data because the IoT device and gateway were disconnected. So, from the results of this test, the maximum distance for the system to acquire data is 10 meters. These results are also influenced by the condition of the test environment, which has a network connection barrier, namely the presence of a house wall.

3.3 Store and Forward Mechanism Performance Testing

Testing the performance of the store and the forward mechanism is carried out to test the size of the data stored on the Android device while storing data. Then testing also tests the level of success of the forwarded data. The test starts by disconnecting the internet, then sending data to the system application. Then it records the size of the data in the local storage database for each time. The experiment was carried out ten times with new data for each trial but having the same data bit length. If data transmission is carried out with no internet available, the system will store data. The test was carried out every time using new data. The test results can be referred to in Table 4.

Table 4. Results of Store and Forward Performance Test

No.	Time (minutes)	Data size (kb)	Data Delivery Success
1.	1	16	100%
2.	2	20	100%
3.	3	24	100%
4.	4	28	100%
5.	5	32	100%
6.	6	36	100%
7.	7	40	100%
8.	8	44	100%
9.	9	48	100%
10.	10	52	100%

Based on Table 4, The table shows the results of store and forward testing at each time. The time column shows the length of time when recording data. From these results, there is a change in the data size, which is 4kb for one minute. So, it can be concluded, during one minute of data acquisition with the internet condition being disconnected and producing 60 data per minute. Each data can be calculated by dividing the 4kb value by 60 which is 66.67 bytes for each data.

3.4 Delayed Data Screening Test Sent to Server

Delayed data is data in local storage obtained from failed data transmission to server or, in other words, data from store process result. The tests carried out are to determine the similarity of data in local storage, with data sent from local storage data, used to check whether all the data was successfully sent to the server and check data redundancy. If there is data redundancy, the data in local storage with data sent from local storage will have differences. The test begins by disconnecting the internet connection to make the internet connection on the android device unavailable, then sending data. If data transmission is done without internet available, the system will perform storage. The test was carried out for 10 minutes by recording the results every minute. The results of the test can be referred to in Table 5.

Table 5. Results of Delayed Data Screening Test

No.	Time (minutes)	The similarity of Data Store and Forward	Data Redundancy
1.	1	100%	0%
2.	2	100%	0%
3.	3	100%	0%
4.	4	100%	0%
5.	5	100%	0%
6.	6	100%	0%
7.	7	100%	0%
8.	8	100%	0%
9.	9	100%	0%
10.	10	100%	0%

Based on Table 5, the delayed data filtering test results can be known at any time, there is no data redundancy, or the data redundancy level is 0%. Then the pending data with the data that has been successfully sent from the pending data is 100%, which means that all data is successfully sent in each test.

4. Conclusion

Based on the research that has been done, the implementation of data acquisition using Bluetooth Low Energy has been successfully implemented. So, the following conclusions are obtained based on the performance test of the data acquisition system, with the measurement parameter being the distance of the gateway with the IoT device. We can see that the maximum distance of the IoT device with the gateway can be connected at a distance of 10 meters. For a distance of more than 10 meters, the IoT device with the gateway will be disconnected. Then based on the performance testing of the store and forward mechanism, it can be seen that every minute of the experiment, there is an increase in the size of the data by 4kb. The process of data acquisition and data transmission is carried out every second, which means that it will produce 60 data for one minute. So, for each data, it can be concluded that the size is 66.67 bytes. Based on the pending data filtering test, it can be seen that previously pending data was due to internet connection disruptions, which temporarily stored the data in local storage. When the data was sent back to the server when the internet connection returned to normal, the data did not experience redundancy. Data has 100% similarity between pending data and successfully transmitted data.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communication Surveys & Tutorials,* vol. XVII, no. 4, p. 2347, 2015. https://doi.org/10.1109/COMST.2015.2444095

[2] M. Donati, A. Celli, A. Ruiu, S. Saponara and L. Fanucci, "A Telemedicine Service System Exploiting BT/BLE Wireless Sensors for Remote Management of Chronic Patients," in *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, Thessaloniki, Greece, 2019. https://doi.org/10.1109/MOCAST.2018.8376643

[3] I. A. Pap, S. Oniga, I. Orha and A. Alexan, "IoT-Based eHealth Data Acquisition System," in *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, Romania, 2018. https://doi.org/10.1109/AQTR.2018.8402711

[4] S. L. Groom, "Can We Measure Our Way Out Of Trouble? The Truth Behind Condition Monitoring," in *6th IET Conference on Railway Condition Monitoring (RCM 2014)*, Birmingham, 2014. https://doi.org/10.1049/cp.2014.1007

[5] A. S. Patade, H. P. Gandhi and N. Sharma, "IoT Solutions for Hospitals," in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, Bengaluru, India, 2019. https://doi.org/10.1109/COMSNETS.2019.8711425

[6] S. S. Mishra and A. Rasool, "IoT Health care Monitoring and Tracking: A Survey," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2019. https://doi.org/10.1109/ICOEI.2019.8862763

[7] A. Mumrez, H. Tariq, U. Ajmal and M. Abrar, "IOT- Based Framework for E-Health Monitoring System," in *2019 International Conference on Green and Human Information Technology (ICGHIT)*, Kuala Lumpur, Malaysia, 2019. https://doi.org/10.1109/ICGHIT.2019.00033

[8] C. Lee, T. Kim and S. J. Hyun, "A Data Acquisition Architecture for Healthcare Services in Mobile Sensor Networks," in *2016 International Conference on Big Data and Smart Computing (BigComp)*, Hong Kong, China, 2016. https://doi.org/10.1109/BIGCOMP.2016.7425966

[9] A. Bouslama, Y. Laaziz and A. Tali, "Scalable and Real-time time series Analytics: Telemedicine as use case," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Marrakech, Morocco, 2018. https://doi.org/10.1109/CIST.2018.8596544

[10] V. Chamola, V. Hassijavi, V. Gupta, and M. Guizani, "A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain and 5G in Managing its Impact," vol. 8, pp. 90225-90265, 2020. https://doi.org/10.1109/ACCESS.2020.2992341

[11] G. Casalino, G. Castellano and G. Zaza, "A mHealth Solution for Contact-Less Self-Monitoring of Blood Oxygen Saturation," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, Rennes, France, 2020. https://doi.org/10.1109/ISCC50000.2020.9219718

[12] J.V.Alamelu and A.Mythili, "Design of IoT based Generic Health Care System," in *2017 International Conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, Vellore, India, 2017. https://doi.org/10.1109/ICMDCS.2017.8211698

[13] L. Min, "Design of Multi-network Data Acquisition System Based on Cloud Platform," in *2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, Jishou, China, 2019. https://doi.org/10.1109/ICVRIS.2019.00033

[14] M. Choi, J. Han and I. Lee, "An Efficient Energy Monitoring Method Based on Bluetooth," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2016. https://doi.org/10.1109/ICCE.2016.7430647

[15] W. Sun, S. Yue, Q. Liu, X. Lin and W. Hu, "Bulk Data Transfer with Store-and-Forward in Wide Area Networks," in *2019 21st International Conference on Transparent Optical Networks (ICTON)*, Angers, France, 2019. https://doi.org/10.1109/ICTON.2019.8840553

[16] T. D. Nguyen, V. Nguyen, V.-N. Pham, L. N. T. Huynh, M. D. Hossain and E.-N. Huh, "Modeling Data Redundancy and Cost-Aware Task Allocation in MEC-Enabled Internet-of-Vehicles Applications," *IEEE Internet of Things Journal,* vol. VIII, no. 3, pp. 1-16, 2020. https://doi.org/10.1109/JIOT.2020.3015534

[17] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur and M. Conti, "Android Security: A Survey of Issues, Malware," *IEEE Communications Surveys & Tutorials,* vol. XVII, no. 2, pp. 1-27, 2015. https://doi.org/10.1109/COMST.2014.2386139

[18] J. Wan, Y. Zou, Y. Li and J. Wang, "Reflective type blood oxygen saturation detection system based on MAX30100," in *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, Shenzhen, China, 2017. https://doi.org/10.1109/SPAC.2017.8304350

[19] E. Park, M.-S. Lee, H.-S. Kim and S. Bahk, "AdaptaBLE: Adaptive control of data rate, transmission power, and connection interval in Bluetooth low energy," 2020. https://doi.org/10.1016/j.comnet.2020.107520

[20] M. O. A. Kalaa, W. Balid, N. Bitar and H. H. Refai, "Evaluating Bluetooth Low Energy in Realistic Wireless Environments," in *2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*, Porto, Portugal, 2016. https://doi.org/10.1109/CPSBench.2018.00007

[21] M. Nazar and Z. Zulfadli, "Usability Testing of Chemistry Dictionary (ChemDic) Developed on Android Studio," in *2017 International Conference on Electrical Engineering and Informatics (ICELTICs 2017)*, Banda Aceh, Indonesia, 2017. https://doi.org/10.1109/ICELTICS.2017.8253265

[22] D. Gotseva, Y. Tomov and P. Danov, "Comparative study Java vs Kotlin," in *2019 27th National Conference with International Participation (TELECOM)*, Sofia, Bulgaria, 2019. https://doi.org/10.1109/TELECOM48729.2019.8994896

[23] C. Yoon, H. Choi, J. Cho and Y. W. Kim, "CoAP over BLE-GATT for OCF," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea (South), 2017. https://doi.org/10.1109/ICTC.2017.8190936

[24] L. Moron-Rodr, ıguez, B. Longa-Chevarria and P. Shiguihara-Juarez, "Analysis of Image Transfer Mechanisms in a RESTful API client-server architecture and its application to lane detection," in *2017 IEEE International Conference on Aerospace and Signals (INCAS)*, Lima, Peru, 2017. https://doi.org/10.1109/INCAS.2017.8123497

[25] M. Liu and T. Mi, "Design and Implementation of Embedded Home Server Based on Sqlite in Smart Home System," in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, Changsha, China, 2017. https://doi.org/10.1109/ICISCE.2017.199