International Journal of Internet, Broadcasting and Communication Vol.14 No.1 122-128 (2022)
http://dx.doi.org/10.7236/IJIBC.2022.14.1.122

IJIBC 22-1-15

IoT notification system for marine emergencies

Dong-Hwan Gong

Assistant Professor, Department of Computer Engineering, Hansei University, Korea
armyvision@hansei.ac.kr

Abstract

Minimization of human casualties in disaster situations is of paramount importance. In particular, if a marine disaster occurs, it can be directly connected to human casualties, so prompt action is needed. In the event of a marine disaster, the route and location of movement should be identified and life tubes should be used to float on the water. This paper designs and proposes an emergency IoT notification system that can quickly rescue drowning people. The maritime emergency IoT notification system consists of four main types. First, an emergency IoT device that detects the expansion of the life tube and delivers location and situation information to the emergency IoT notification server. Second, an emergency IoT web server that manages emergency information and provides notification. Third, a database server that stores and manages emergency IoT notification information. And finally, an emergency notification app that can receive and respond to emergency notification information. The emergency IoT device consists of a TPMS(Tube Pressure Monitoring System) device that checks the pressure value of the TPMS in real time and sends it to the IoT device, and an IoT device that sends the rescuer's voice information and emergency information to the emergency IoT server. Emergency information is delivered using the MQTT(Message Queuing Telemetry Transport) protocol, and voice information is delivered to the IoT server as HTTP FormData.

Keywords: *IoT Server, Life tube, IoT Device, TPMS, Notification service, Emergency notification.*

1. Introduction

Every year, deaths occur due to swimming accidents, and the causes of fatal accidents are mainly negligence in safety and inexperience in swimming. According to a press release from the Ministry of Public Administration and Security for 2021, there was 158 water-related deaths over the five years from 2016 to 2020, most deaths occurred in the summer. The causes of death in water play include inexperienced swimming (28.5%), negligence in safety (27.2%), drinking and swimming (17.1%), rapids/waves (17.1%), and tube abalone (8.9%). In case of such an accident, if the location information and status information of the survivor can be known for rescue, it is possible to respond quickly[7-8]. Rescuers can use this information to respond more efficiently, appropriately and quickly. In the event of a marine disaster, it is necessary to determine the movement route and location, and to float on the water using a life tube, etc[3]. This paper develops and proposes an emergency IoT notification system that can quickly rescue people from drowning.

Manuscript Received: December. 5, 2021 / Revised: December. 8, 2021 / Accepted: December. 13, 2021
Corresponding Author: armyvision@hansei.ac.kr
Tel: +82-31-450-5340, Fax: +82-31-450-5172
Assistant Professor, Department of Computer Engineering, Hansei University, Korea

The emergency IoT notification system is mainly composed of four types. It consists of first emergency IoT device, second emergency IoT web server, third database server, and fourth emergency notification app[4-5]. An emergency IoT device is a portable, waterproof device that can be dropped using a drone or carried around by a user when a distress occurs. A portable IoT device allows the user to pull the device lever in an emergency to operate the life tube and device. In the IoT device dropped by a drone, the life tube expands when it detects moisture and the device operates automatically. The activated emergency IoT device transmits voice information and emergency information to the emergency IoT server[6]. When emergency information arrives at the emergency IoT server, the emergency IoT server sends an emergency notification message to the rescuer and allows the rescuer to respond. Emergency information includes GPS information of the survivor and information such as voice, time, tube pressure, and device ID.

2. Architecture of the proposal system

The emergency IoT device consists of a TPMS(Tube Pressure Monitoring System) device combined with a life tube, and an IoT device. The IoT device delivers emergency situation information and voice information to the IoT server when pressure is detected by manually or automatically operating the life tube. Emergency information includes device ID, shock value, voltage, pressure, time, GPS location, device status information, and user's voice information for less than 10 seconds. The IoT server consists of an MQTT(Message Queuing Telemetry Transport) broker that receives IoT devices, a Firebase Cloud Messaging (FCM) system that sends notification messages, and an emergency IoT web server that subscribes to MQTT messages, sends notification messages to FCM, and manages emergency situations[10]. MySQL is used for database and storage, and AWS S3 cloud is used for voice information[1-2]. The emergency notification app, which is provided to rescuers for emergency response and rescue of survivors, is executed by receiving a notification message sent from FCM or directly executed by rescuers. The emergency notification app provides emergency status and location information to rescuers so that they can quickly rescue the survivors. This app is developed as PWA (Progressive Web Apps) for scalability and real-time updates[9]. Figure 1 shows the overall structure of an emergency IoT notification system.

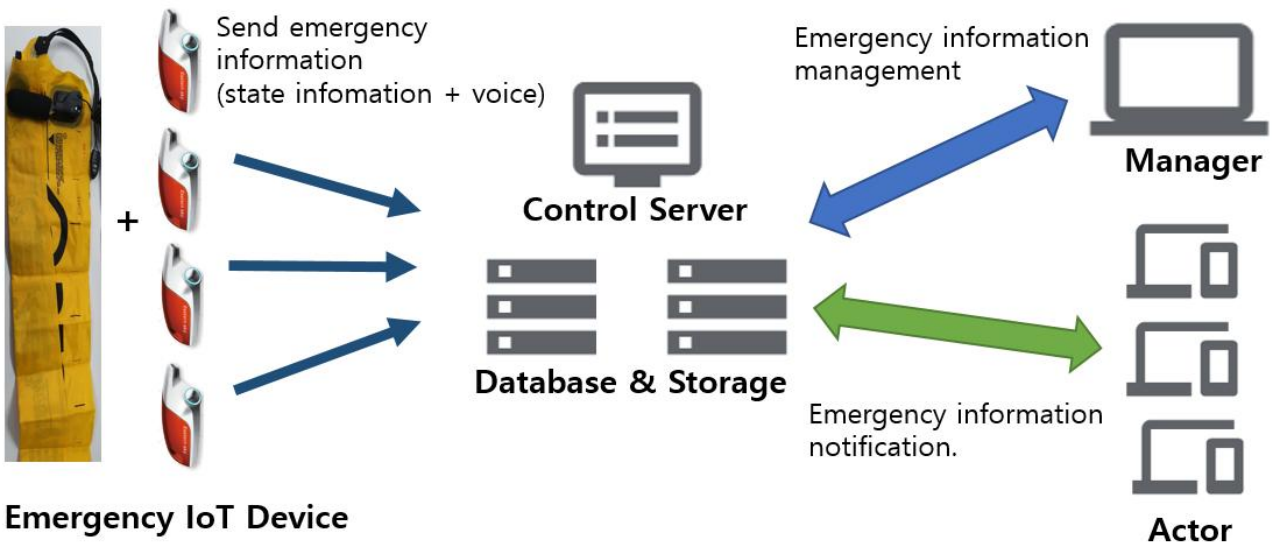

Figure 1. Overall structure of an emergency IoT notification system

Figure 2 is a state diagram of the IoT device. When the TPMS device is charged, it always waits in the Sleep state, and when the pressure of the connected life tube is detected, it switches to the BLE(Bluetooth) Active state and transmits the state information of the life tube to the IoT device through Bluetooth. The IoT device waits in Sleep state when charging, and when it receives the first Bluetooth information from the TPMS device, it enters the Slow Active state, sends emergency information to the IoT server slowly, and when the pressure of the TPMS device changes or the user presses the button, it enters the Fast Active state to quickly transmit emergency information and voice information to the IoT server.

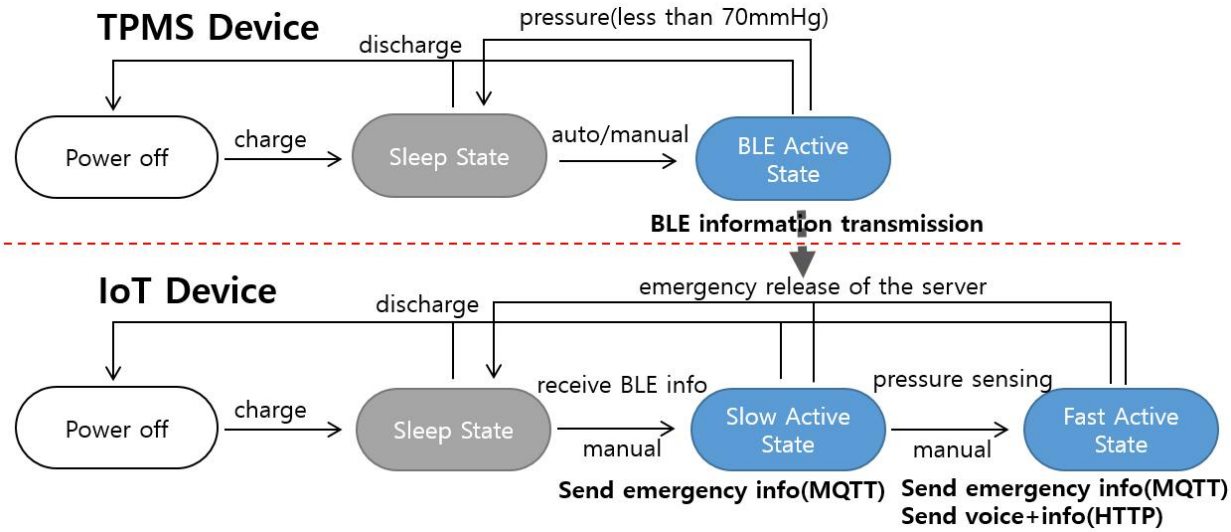

Figure 2. State diagram of the IoT device

Figure 3 shows the design of the IoT server database model. all_devices_info stores all device information produced. Register the device by barcode recognition or manual input by the administrator. device_info indicates a device registered with a management organization such as a police station or a fire station, and police_agency_info indicates a registered management organization. police_member_info stores information about rescuers receiving emergency notification information, and emergency_info is a table for storing all state information about emergency situations. emergency_now_table fetches and stores emergency information generated from IoT devices from the MQTT broker, and the information stored here is transmitted to emergency notification apps of all rescuers through the FCM system. When the response to the emergency situation is finished, the rescuer or manager releases the emergency situation, and the device information in which the emergency situation is released is deleted from emergency_now_table, and changed information is stored in emergency_info. emergency_sound is a table for storing emergency voice information generated in the IoT device.

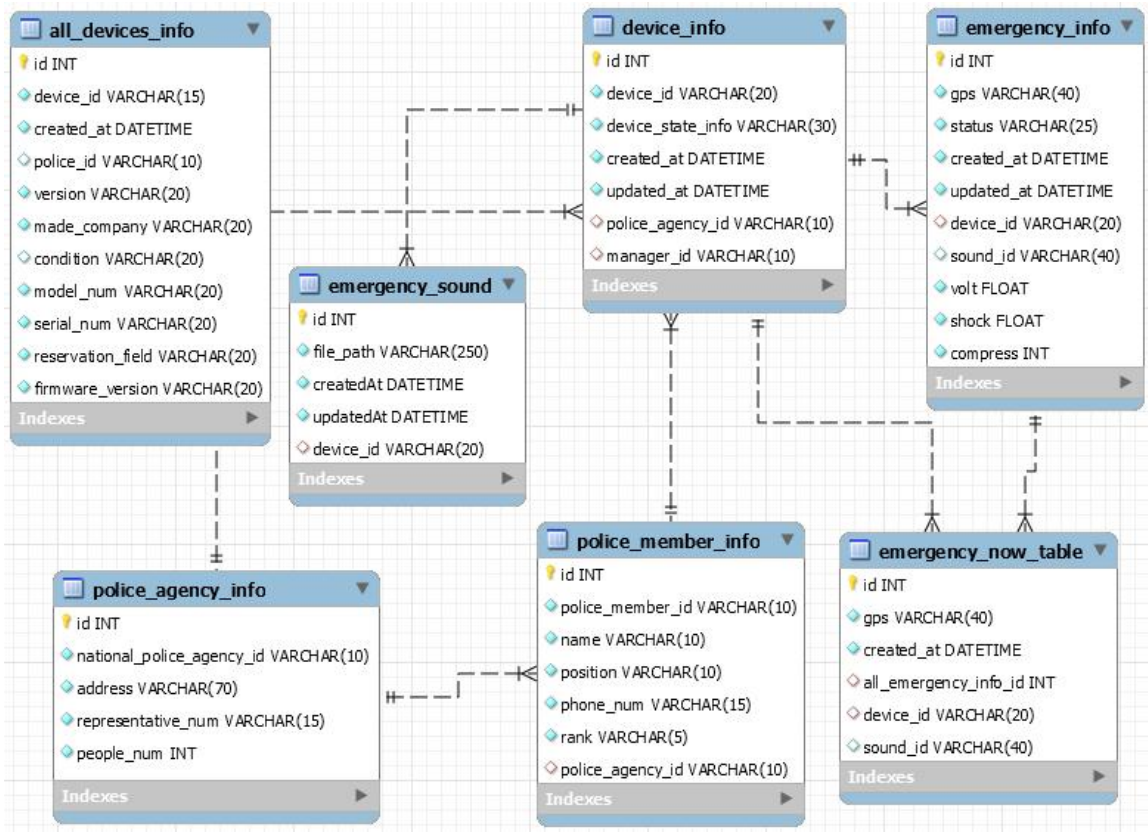

Figure 3. Design of the IoT server database model

3. Implementation of the proposal system

3.1 Hardware Implementation

The emergency IoT device consists of a TPMS device that detects the pressure of the life tube and an IoT device that transmits emergency information and voice information to the IoT server. The life tube connected to the TPMS device automatically expands by injecting carbon dioxide into the tube from the carbon dioxide generating cylinder when moisture is detected or the user pulls the lever. When the life tube is inflated, the TPMS device detects the inflation and transmits the pressure value of the life tube to the IoT device through Bluetooth. The IoT device transmits emergency information and voice information to the IoT server through LTE. Emergency information is transmitted to the MQTT broker, and voice information is transmitted to the IoT server as HTTP multipart/formed-data. Figure 4 is an implementation of an emergency notification IoT device, voice is recorded in less than 10 seconds and transmitted to the IoT server, and GPS location information, time, and pressure information are periodically transmitted according to the device status.

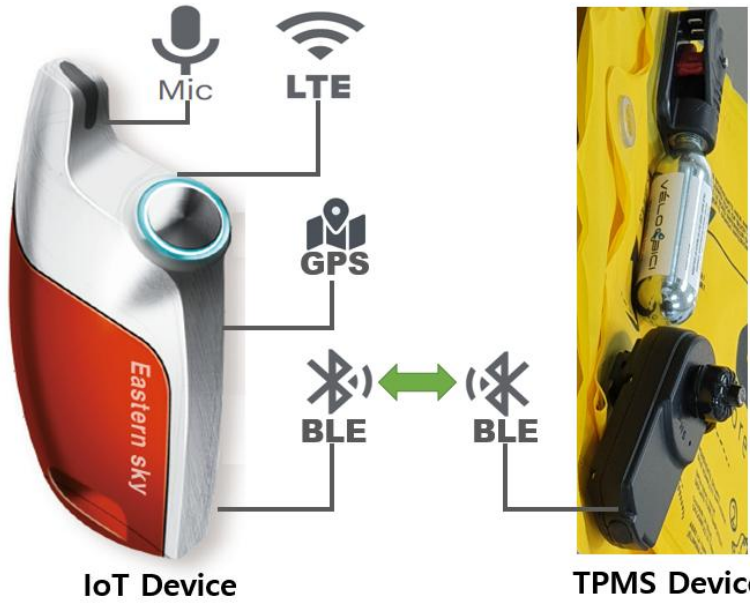

Figure 4. Implementation of an emergency notification IoT device

3.2 Software Implementation

Figure 5 shows the simulated result of emergency notification of an actual IoT device. It retrieves emergency information from the MQTT broker, registers it as a table, and sends notification information to the emergency IoT notification app client to notify rescuers of emergency information. Figure 5 is the result of testing for one week using a simulator to test location information, status information, occurrence time, device ID, and voice information of each device. Emergency status information cannot be released from the device to prevent a user's mistake, and only the administrator of the emergency IoT server can be changed.

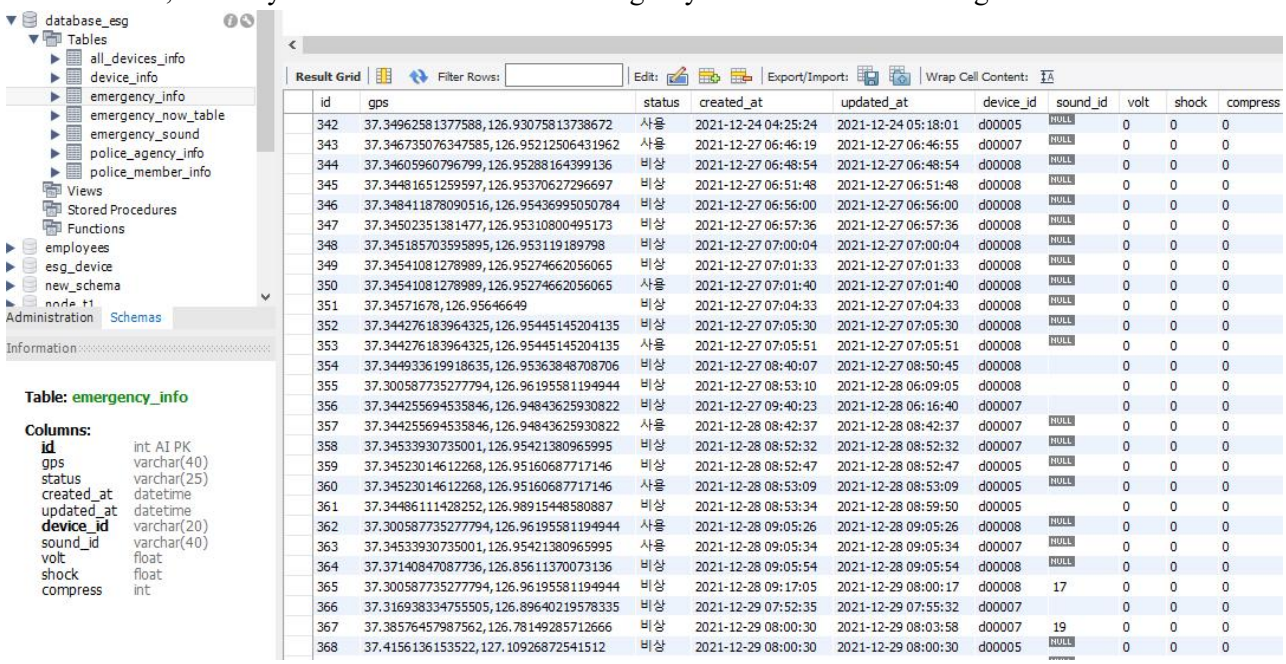

Figure 5. Simulated result of emergency notification of an actual IoT device

Figure 6 shows the location of the emergency situation on the map as a part of the administrator monitoring screen and the notification app screen that implements the emergency IoT notification system. If several emergency situations occur at once, emergency information can be viewed in full screen or by searching in a list. If there is a voice message sent by a person in dangers, it can be listened to and delivered to the rescuer, or the rescuer can directly click the listen button to listen. In case of emergency notification message and device emergency status, emergency status can be canceled only with administrator authority, and single release and multiple inquiry release are possible. The emergency notification app was implemented based on PWA and was developed to run on the web, app, and multiple platforms and to have excellent program extensibility.

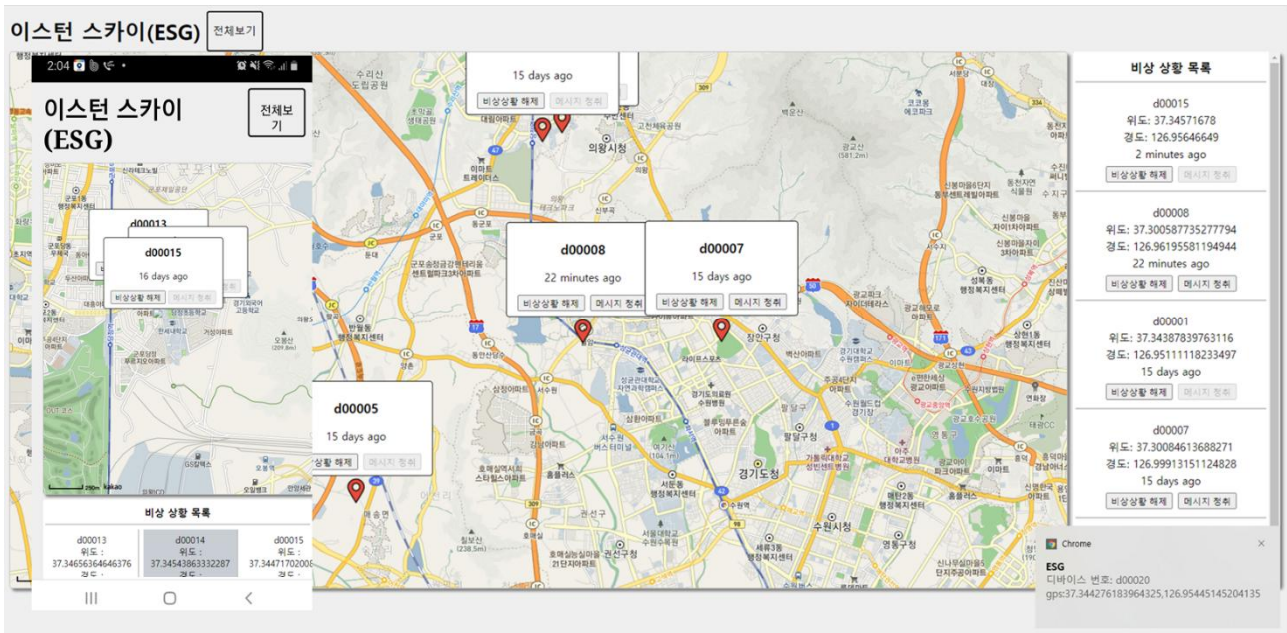

Figure 6. Location of the emergency situation on the map

4. Conclusion

Unfortunately, many people lose their lives every year in water accidents. The main causes of death are inexperienced swimming and negligence in safety. In the event of such an accident, if the status information and location information of the survivors are known, appropriate measures can be taken. This information is important because people around you or rescuers can help you quickly. In addition, it is possible to determine the movement route and location in a disaster situation, thereby minimizing human casualties. For this reasons, in this paper, we proposed and implemented an emergency IoT notification system to quickly rescue people from drowning in a disaster situation.

The hardware of the design system consists of a IoT device and an TPMS device combined with a life tube, and the two devices communicate through Bluetooth. The TPMS device of the life tube transmits information such as whether the person in distress is holding the life tube to the IoT device, and the IoT device sends emergency information to the IoT server. The software of the design system consists of a server that manages emergency information, a database that stores it, and an app that receives emergency information notifications. The system of this paper is thought to be able to provide various information that can minimize human casualties and help the victims quickly in case of emergency.

References

[1] Kyubaek Kim, “Integrated Service Framework for Naval Ship using REST Architecture”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC). Vol. 17, No. 4, pp. 1-10, Aug. 2017.
DOI : https://doi.org/10.7236/JIIBC.2017.17.4.1

[2] Seokjin Im and Heejoung Hwang, “Design and Development of Framework for Wireless Data Broadcast of XML-based CCR Documents”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC). Vol. 15, No. 5, pp. 169-175, Oct. 2015.
DOI : http://dx.doi.org/10.7236/JIIBC.2015.15.5.169

[3] Jongmin Eun, Jaekon Oh and Jeongjoon Kim, “Group Management System based on Apache Web Server and Android App”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC). Vol. 18, No. 2, pp. 141-147, Apr. 2018.
DOI : https://doi.org/10.7236/JIIBC.2018.18.2.141

[4] Myungkyu Yi and Heejoung Hwang, “A Secure Personal Health Record System for Handling of Emergency Situations”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC). Vol. 16, No. 5, pp. 117-123, Oct. 2016.
DOI : http://dx.doi.org/10.7236/JIIBC.2016.16.5.117

[5] Hotae Lee and Byoungchul Kim, “Implementation of Maritime Telemedicine System Using Android”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC). Vol. 18, No. 6, pp. 221-228, Dec. 2018.
DOI : https://doi.org/10.7236/JIIBC.2018.18.6.221

[6] Hyokyung Bahn, “Management Technique of Energy-Efficient Cache and Memory for Mobile IoT Devices”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC). Vol. 21, No. 2, pp. 27-32, Apr. 2021.
DOI : https://doi.org/10.7236/JIIBC.2021.21.2.27

[7] Huhmyung Hoi and Shungjung Shin, “A Study on the Energy Saving House System Using IOT Technology”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC). Vol. 20, No. 6, pp.109-113, Dec. 2020.
DOI : https://doi.org/10.7236/JIIBC.2020.20.6.109

[8] Jihun Lim, Gyusik Yang, Sunghun Jung and Dongkook Park, “Lifejacket-Integrated Antenna for Search and Rescue System”, J. Navig. Port Res. Vol. 38, No. 4, pp. 367-371, Aug. 2014.
DOI : http://dx.doi.org/10.5394/KINPR.2014.38.4.367

[9] FCM, https://en.wikipedia.org/wiki/Firebase_Cloud_Messaging

[10] PWA, https://en.wikipedia.org/wiki/Progressive_web_application