

다수의 이종 기기로부터 수집된 데이터의 실시간 관리 시스템 구현

이 정 진*, 박 형 곤°

Implementation of the Real-Time Management System for Data Collected from Multiple Heterogeneous Devices

Jungjin Lee*, Hyunggon Park°

요 약

본 논문은 위치 정보를 제공할 수 있는 다수의 기기로부터 전달되는 위치 데이터와 사용자로부터 발생한 건강 데이터를 실시간으로 수집, 전달 및 저장할 수 있는 시스템을 구현하였다. 구현한 시스템은 비콘 기기로부터 위치 데이터를 사용자 기기에 실시간으로 수집하고, 스마트워치로부터 건강 데이터를 수집하여 사용자 기기 내부에 저장할 수 있다. 앱 간 데이터 공유 영역을 설정하여 2개 이상의 애플리케이션에서 하나의 파일에 데이터를 저장하고 읽어오는 것이 가능하다. 사용자 기기와 서버 간 통신은 소켓이 연결된 이후에 모든 데이터의 전송이 안정적으로 이루어질 수 있도록 소켓 통신으로 데이터 통신 기능을 구현하였고, 이에 따라 클라이언트가 서버와 접속이 되면 서버나 클라이언트에서 접속을 해제할 때까지는 계속해서 접속을 유지할 수 있다. 따라서 서버와 클라이언트가 한번 연결되면 계속 같은 소켓으로 통신하므로 서버 부하를 줄여 원활한 데이터 전송이 가능하다. 또한 소켓 통신은 이벤트 기반으로 동작하기 때문에 서버에서 데이터베이스 생성과 이벤트를 추가하면 다른 기기에 대한 데이터 전송도 가능하다. 마지막으로 사용자 기기에 데이터를 불러오는 방식에 따라 발생할 수 있는 데이터의 중복적 저장 문제를 해결하기 위하여 중복 데이터를 제거하기 위한 알고리즘을 제안하고 구현하여 실제 시스템에서 발생하는 데이터 중복성 문제를 해결하였다.

키워드 : 위치 기반 서비스, 이종기기, 비콘, 스마트워치, 실시간, 데이터 관리 시스템

Key Words : location based services, heterogeneous devices, beacon, smart watch, real-time, data management system

ABSTRACT

In this paper, we implemented a system that can collect, transmit, and store location data delivered from multiple location-aware devices and health data generated from users in real-time. The implemented system includes functions to store data inside the user devices in real-time, where the data can be location of beacon devices and health data from smart watches. By setting the data sharing area between applications, two or more applications can save and read the data in a single file. The communication between user devices and

* 본 연구는 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(2021-0-00739)과 한국연구재단의 지원(NRF-2020R1A2B5B01002528)을 받아 수행되었습니다.

• First Author : Ewha Womans University Department of Electronic and Electrical Engineering, jungjin.lee@ewhain.net, 정희원

° Corresponding Author : Ewha Womans University Department of Electronic and Electrical Engineering, Graduate Program in Smart Factory, hyunggon.park@ewha.ac.kr, 종신회원

논문번호 : 202111-312-0-SE, Received October 29, 2021; Revised January 24, 2022; Accepted March 9, 2022

servers has been implemented as a data communication function based on the socket communication protocol to ensure stable transmission once the socket is connected, and as a result, the connection can be maintained until the server or client disconnects. Therefore, once the server and the client are connected, they continue to communicate through the same socket, thereby reducing server load and smoothly transmitting data. Since socket communication operates based on events, in addition, data transmission from other devices can also be enabled by creating a database and adds an event. Finally, we proposed an algorithm that can solve the data duplication problem in the actual system implementation by removing duplicate data.

I. 서 론

스마트폰이 대중화되고 사물 인터넷 환경이 보편화 되면서 다양한 서비스에 대한 사회적·상업적 관심이 크게 높아지고 있다. 그림 1에서 볼 수 있듯이 전 세계 사물 인터넷 시장 규모는 2020년 기준 약 320억 달러이며, 연평균 성장률이 꾸준히 상승하여 2026년에는 약 920억 달러로 성장할 것으로 예상된다. 사물 인터넷 기기 도입 증가, 네트워크 인프라 확산, 위치 기반 서비스 수요 증가로 제조, 운송, 에너지 시설 등 관련 산업 및 서비스가 급성장하고 있다¹⁾.

사물 인터넷 기반 서비스 중 하나인 위치 기반 서비스(LBS, Location Based Service)는 이동통신망의 기지국 셀 정보, GPS(Global Positioning System), 블루투스, Wi-Fi와 같은 다양한 무선통신 기술을 이용하여 사용자의 위치 측정, 필요에 맞는 가장 적절한 맞춤형 서비스 제공을 목적으로 한다²⁾. 이와 더불어 스마트워치와 같은 웨어러블 디바이스가 대중화되고 널리 사용되기 시작하면서³⁾ 웨어러블 디바이스에 포함된 다양한 센서로부터 측정되고 수집된 사용자의 움직임 데이터, 신체 활동 데이터, 생체 데이터 등을 분석하여 개인 맞춤형 서비스 제공도 가능해졌다⁴⁾.

이와 같은 데이터를 분석을 통한 프로세스 효율화

는 산업 전반을 통하여 이루어지고 있으며, 이에 따라 데이터 수집과 저장에 대한 중요성도 높아지고 있다⁵⁾. 스마트 공장은 제품 생산과정에 다양한 정보통신 기술을 이용하여 제품을 생산하는 지능형 공장으로서 생산 설비에 인공지능을 이용하여 현장 자동화에 활용되고 있다⁶⁾. 컨트롤러, 센서 등을 통해 생산 설비를 제어하고 생산 설비로부터 발생한 상태·가동 데이터를 수집하고 분석함으로써 자동 제어를 실현 및 적용하고 있다⁷⁾. 또한, 작업자들의 경우 상시 관리 감독을 하지 않으면서도 위치 기반 시스템을 이용하여 공장 설비에 문제가 생길 경우 모바일 디바이스 또는 웨어러블 디바이스 등을 통하여 알림 또는 경고를 받는 것이 가능하다⁸⁾.

건설 현장에서는 다양한 사고 위험요소들이 존재하고, 이러한 위험요소들이 실제 위험 상황에 이르지 않도록 모니터링과 작업자 관리에 데이터 기반 프로세스가 활용되고 있다. 작업 환경에서 사고를 줄이기 위해서 카메라나 센서, 작업자의 상태를 모니터링 할 수 있는 데이터 및 작업자의 생체 데이터 등을 수집하고 분석하여 사고가 발생하기 전에 경보를 줄 수 있는 시스템이 필요하며 이에 대한 연구가 활발히 이루어지고 있다⁹⁾.

의료 현장에서는 끊임 없는 의료서비스를 제공하기 위해서 간호사들이 교대로 업무를 수행하는 근무체제를 활용하고 있다. 교대 근무는 생체리듬에 역행하는 생활을 하기 때문에 근무자들은 여러 가지 정신적·신체적으로 어려움¹⁰⁻¹²⁾을 겪고 있다. 따라서 업무 공간별, 교대 근무에 따른 시간대별 활동량을 분석하고 분석 결과를 바탕으로 적절한 업무량을 배분하는데 적용하고자 하는 움직임을 보이고 있다¹³⁾.

최근 대형병원을 중심으로 빠르게 확산되는 스마트 병원은 ICT기술을 의료에 적용하여 환자의 안전관리, 진단 및 치료 등 다양한 분야에서 의료 서비스를 제공하고 있으며¹⁴⁾, 국내 통신사들은 사물 인터넷 기반 웨어러블 위치확인 장비 등을 활용하여 실시간 개인 맞춤형 헬스케어 서비스를 제공하고 있다.

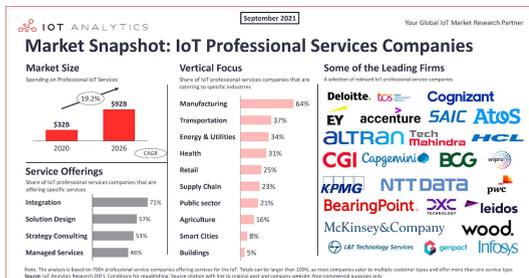


그림 1. 사물 인터넷 시장 규모(IoT Analytics Research 2021)
Fig. 1. Internet of Things market size and related industries (IoT Analytics Research 2021)

이와 같이 다양한 산업현장에서 데이터를 기반으로 머신러닝, 인공지능과 같은 데이터 분석·추론 기술을 활용하는 노력이 매우 활발하게 이루어지고 있다. 이때, 머신러닝 또는 인공지능 기술의 활용 성능에 양질의 데이터가 분석 모델의 성능 향상에 직접적인 영향을 미치는 중요한 요소이다¹⁵⁾. 따라서 다양한 이중적 디바이스로부터 측정되고 생성되는 데이터를 안정적으로 수집하고 저장하며 이를 전달할 수 있는 시스템 구축은 매우 중요하다.

본 논문에서는 위치 정보를 제공할 수 있는 다수의 기기로부터 전달되는 데이터와 사용자로부터 발생 및 수집된 데이터를 실시간으로 수집, 전달 및 저장할 수 있는 시스템을 다음과 같이 제안한다. 첫째, 사용자 기기에 위치 데이터와 건강 데이터를 실시간으로 수집하여 기기 내부에 저장한다. 둘째, 사용자 기기와 서버 간 통신 서비스를 소켓 통신으로 구축하고 수집한 데이터를 전달한다. 수집한 데이터는 데이터베이스의 열의 개수만큼 전달 되어야하기 때문에 소켓이 연결된 이후에 모든 데이터의 전송이 안정적으로 이루어져야 한다. 따라서 소켓 통신으로 통신을 구현한다. 셋째, 데이터의 중복을 방지한다. 본 논문에서는 위치 데이터와 건강 데이터를 전송하며 두 데이터가 사용자 기기에 저장된 방식이 다르기 때문에 데이터를 불러오는 방법에 따라 다른 방식으로 중복 데이터를 제거한다.

본 논문은 다음의 순서와 같다. II장에서는 제안하는 시스템의 전체적인 구조를 설명하고, III장에서는 시스템 구현 내용을 기술한다. IV장에서는 구현 결과를 제시하고, V장에서는 최종 결론을 맺는다.

II. 제안하는 시스템 개요

제안하는 시스템은 그림 2와 같이 데이터 생성 기기, 사용자 기기(클라이언트), 서버, 데이터베이스 4가지로 요소로 구성되어 있다. 각 요소들은 위치 데이터와 건강 데이터 생성 기능, 사용자 기기에 수집 데이터 저장 기능, 클라이언트와 서버의 통신 기능, 서버의 데이터베이스에 데이터 저장 기능을 수행한다.

실내의 위치를 파악하는데 필요한 데이터는 다수의 비콘과 같은 이중적 디바이스로부터 생성 및 제공되며, 사용자의 다양한 활동과 생체 상태를 파악하는데 필요한 데이터는 스마트워치와 같은 웨어러블 디바이스로부터 수집된다. 본 논문에서는 실내에 고정된 위치에 설치된 비콘으로부터 생성된 데이터를 수집하는데, 비콘은 블루투스를 사용해 주변에 있는 기기에 신

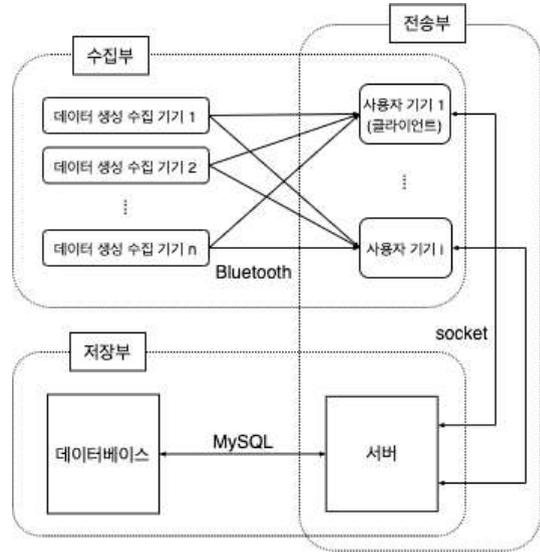


그림 2. 시스템 개요
Fig. 2. System overview

호를 전달하는 장치로 근처에 있는 비콘의 신호 정보를 통해 현재 위치에 대한 정보를 파악할 수 있다. 사용자의 활동 및 생체 상태를 위한 건강 데이터는 스마트워치로부터 수집된다.

위와 같은 데이터는 사용자 기기에 저장된다. 사용자 기기는 근처에 있는 비콘 신호를 검색하고 발생한 신호 정보를 기기 내부의 데이터베이스에 저장한다. 본 논문에서는 내부 데이터베이스로 SQLite를 사용하는데, 이는 스마트폰과 같은 기기 모바일 기기에서 사용하기 적절한 데이터베이스이다. 건강 데이터는 스마트워치로 수집하며 사용자 기기와 페어링 되어 자동으로 기기 내부에 저장된다.

사용자 기기에 저장된 데이터를 서버로 전달하기 위한 클라이언트와 서버 간 통신은 소켓 통신을 기반으로 이루어진다. 일반적으로 널리 활용되는 HTTP 통신은 데이터를 전송할 때마다 연결을 요청하기 때문에 부하가 발생할 수 있는 반면, 소켓 통신은 서버와 클라이언트가 특정 포트를 통해 연결을 유지하고 있어서 다량의 데이터를 전송하기에 적합한 통신 방법이다. 또한, 연결 종료하기 전까지 연결을 유지하기 때문에 서버에서 클라이언트로도 요청을 보낼 수 있다.

본 논문에서는 제안하는 시스템 구현을 위해 socket.io 라이브러리를 활용한 소켓 통신을 적용했다. 그러나 최근 IoT 기기들의 효과적인 통신을 위해 개발된 CoAP, MQTT, XMPP, AMQP 등 IoT 메시지 전송 프로토콜도 사용 가능하며¹⁶⁾, 시스템에 적합한

통신 방식을 채택하여 사용할 수 있다.

서버로 전달된 데이터는 서버 데이터베이스에 저장된다. 서버에서는 전송받은 위치 데이터와 건강 데이터를 미리 만들어둔 MySQL 데이터베이스에 저장한다. 데이터베이스는 비콘 데이터를 저장하는 테이블과 건강 데이터를 저장하는 테이블로 구성되어 있다.

위에서 설명한 다양한 시스템 구성 요소에 대한 구체적인 설명은 III장에서 기술한다.

III. 시스템 구현

본 장에서는 그림 3과 같이 제안하는 시스템에서 데이터를 생성하여 사용자 기기에 저장하는 수집부, 사용자 기기로부터 서버에 데이터를 전송하는 전송부, 서버의 데이터베이스에 저장하는 저장부로 나누어 설명한다.

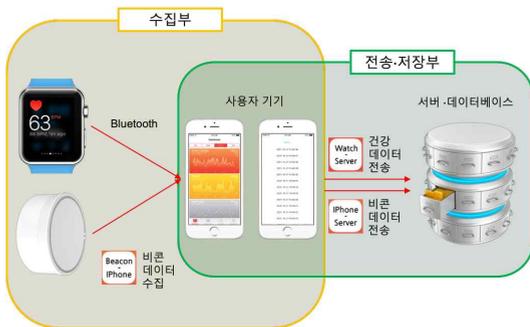


그림 3. 실제 시스템 구성도
Fig. 3. Actual system configuration

3.1 비콘 기반 위치 정보 데이터 수집 및 전송

3.1.1 수집부

그림 4는 비콘의 수집부로 총 3단계의 과정으로 구성되어 있다.

1단계: 애플리케이션을 시작하면 데이터베이스를 가져오고, 데이터를 저장할 테이블이 없다면 테이블을 생성하여 수집한 데이터를 저장할 수 있는 상태를 만든다. 데이터베이스의 테이블은 현재 시각, 수집장치의 정보를 알 수 있는 기기명, 비콘 식별을 위한 Major, Minor 값, 근접도, 정확도, RSSI(Received Signal Strength Indicator) 값 총 7개의 열로 구성되어 있다.

2단계: 비콘 스캐닝(scanning)으로 사전에 지정된 UUID 값으로 최근접 3대의 비콘 신호를 수신한다.

3단계: 미리 생성한 테이블에 수집한 데이터를 삽

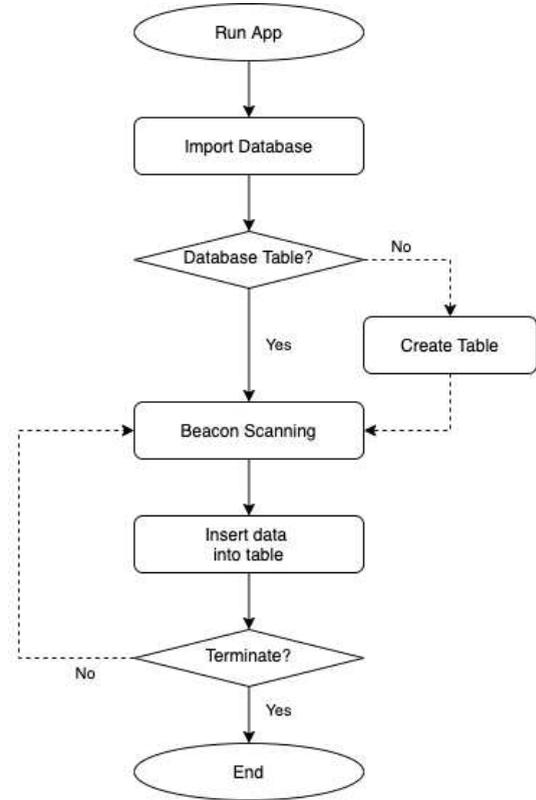


그림 4. 비콘 데이터 수집 다이어그램
Fig. 4. Diagram of beacon data collection

입한다.

애플리케이션을 종료하기 전까지 2, 3단계의 과정을 반복 수행한다.

3.1.2 전송-저장부

그림 5는 수집한 비콘 데이터를 서버로 전송하기 위한 과정을 나타낸다. 서버에 데이터를 전송하기 위한 방법으로 소켓 통신을 이용한다. 소켓 통신의 가장 큰 특징은 실시간, 양방향, 이벤트 기반의 통신 방법이다. 비콘 데이터는 1초에 1개의 데이터 셋이 발생하기 때문에 장시간 수집을 할 경우 많은 양의 데이터가 발생하게 된다. 다량의 데이터를 실시간으로 전송하면서, 서버와 클라이언트의 양방향 통신이 필요하기 때문에 소켓 통신을 사용했다.

총 5단계의 과정으로 데이터를 서버에 저장하도록 시스템을 구축하였다.

1단계: 클라이언트와 서버 간 데이터를 전송하는 통로로 소켓 연결을 한다. 비콘 데이터를 전송하기 위해서 서버의 3000번 포트를 이용한다.

2단계: 소켓이 연결되면 클라이언트인 아이폰에 미

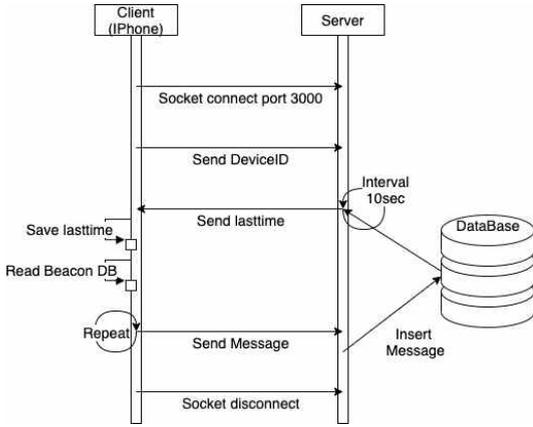


그림 5. 비콘 데이터 전송 다이어그램
Fig. 5. Diagram of beacon data transmission

리 설정된 기기 명을 전송한다. 기기명은 추후 서버의 데이터베이스에 쿼리 요청을 위해 필요한 변수로 사용된다.

3단계: 서버에서는 현재 소켓에 연결된 기기명 조건 하에 저장된 데이터 중 가장 최신의 시각을 10초마다 클라이언트로 전송하며, 클라이언트에서는 수신한 데이터를 lasttime 변수에 저장한다.

4단계: 클라이언트에서는 저장된 비콘 테이블을 한 행씩 읽고 서버로 Message 이벤트로 데이터를 전송한다. 데이터 전송한 행에 대해서는 테이블에서 제거한다. 서버에서는 Message 이벤트로 전송받은 데이터를 미리 만들어둔 데이터베이스에 삽입한다. 이 때 전송하는 데이터는 사용자 기기에 대한 정보를 포함하여 시각, Major, Minor, 근접도, 정확도 RSSI 총 7개의 데이터를 전송한다. 마지막 행의 데이터 전송까지 4단계를 반복한다.

5단계: 애플리케이션 UI 상에서 Button을 누르면 lasttime과 Message의 타임스탬프를 비교하고, 이 값이 일치하는 경우에 소켓의 연결을 종료한다. 애플리케이션을 종료하면 클라이언트의 기기 안에는 빈 테이블만 존재하게 된다.

3.1.3 데이터 중복 방지 처리

비콘 데이터를 전송할 때, 애플리케이션 화면의 Button을 통해 시간을 비교함으로써 데이터 전송이 완료됨을 확인한다. 데이터 전송이 완료된 것을 확인하면 아이폰에 저장된 데이터베이스 테이블을 내용을 지워줌으로써 데이터가 중복으로 전송되지 않게 해준다. Button을 누르는 과정을 무시한 경우 사용자 기기에서 데이터베이스 테이블 내용이 지워지지 않게 되

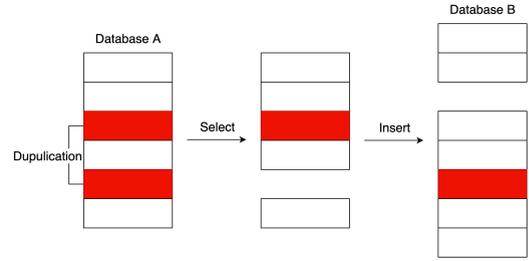


그림 6. 데이터 중복 제거 방법
Fig. 6. Method of removing duplicated data

므로 다시 애플리케이션을 실행하면 서버로 데이터가 다시 전송되어 결과적으로 중복 데이터가 저장되게 된다. 따라서 서버 단에서 중복 데이터를 처리하기 위해 다음과 같은 작업을 수행한다. 그림 6과 같이 서버에서는 중복으로 저장된 데이터를 데이터베이스 A에서 제거하기 위해 데이터베이스 B를 생성해둔다. 주기적으로 중복되지 않은 데이터를 데이터베이스 A에서 선택된 데이터는 데이터베이스 B로 이동한다. 그리고 기존의 데이터베이스 테이블의 내용은 제거한다. 데이터베이스 A는 전송받은 데이터를 모으는 임시 저장소이며, 데이터베이스 B는 최종적으로 데이터를 저장하는 곳이다.

ETL(추출, 변환, 적재) 프로세스에서는 임시 데이터베이스에서 중복데이터를 처리하고 최종 데이터베이스에 적재하는 방식을 사용한다¹⁷⁾. 본 논문에서는 ETL 프로세스 방식을 이용, 임시 데이터베이스에서 중복데이터를 제거한 후 최종 데이터베이스에 적재하는 방식을 사용한다. 이는 저장된 전체 데이터를 처리하는 것이 아닌 새롭게 저장된 데이터만 처리하기 때문에 전체 데이터를 처리하는 것에 비해 훨씬 적은 시간이 소요되어 효율적으로 데이터의 중복을 방지할 수 있다.

3.2 건강 데이터 수집 및 전송

아이폰에서 사용자 건강 데이터에 접근하기 위해 헬스킷(HealthKit)을 사용한다. 헬스킷은 iOS 운영체제를 사용하는 기기에서 사용자의 건강 및 피트니스 데이터에 접근하고 공유가 가능하게 하는 프레임워크이다. 헬스킷은 애플리케이션 간 데이터를 공유하도록 설계되어 있기 때문에 개발자는 언제든지 원하는 데이터를 읽어 올 수 있다. 헬스킷 자체적으로 다양한 데이터 유형과 단위를 제공하지만 사용자 정의 데이터 유형이나 단위는 사용할 수 없다. 헬스킷에서 제공하는 데이터 유형으로는 특성(Characteristic), 샘플

(Sample), 운동(Workout), 소스(Source) 등이 있다. 특정 데이터에는 생일, 혈액형, 성별, 피부 타입 등이 있고, 샘플 데이터는 특정 시점에 저장된 기록이며, 운동 데이터는 피트니스 및 운동 활동 중 저장된 기록이다. 마지막으로 소스 데이터는 각 샘플을 저장한 기기에 관한 정보를 나타낸다.

웨어러블 디바이스 중 하나인 애플워치로 수집한 건강 데이터는 아이폰과 페어링 되어서 데이터가 자동으로 동기화 된다. 본 논문에서 수집하려는 건강 데이터는 샘플 데이터에서 원하는 시간의 데이터를 쿼리 요청하여 얻을 수 있다.

3.2.1 전송-저장부

그림 7은 클라이언트에서 수집한 샘플 데이터에서 원하는 시간 영역의 데이터를 서버로 전송하는 과정이다.

1단계: 클라이언트와 서버 간 데이터를 전송하는 통로로 4000번 포트를 통해 소켓 연결을 한다. 샘플 데이터는 가져올 건강 데이터 타입, 시간, 단위를 지정해주면 쿼리 요청을 통해 원하는 데이터를 가져올 수 있다.

2단계: 건강 데이터 타입별로 불러와서 서버로 전송한다. 건강 데이터 타입은 총 5 종류로 심박 수(heartRate), 스텝 수(stepCount), 걷기 및 달리기 거리(distanceWalkingRunning), 휴식 에너지(basalEnergyBurned), 활동 에너지(activeEnergyBurned)를 순차적으로 읽고 서버로 데이터를 전송한다. 데이터 타입 별로 이벤트를 구분해

서 사용하기 때문에 서버의 원하는 데이터베이스에 삽입하게 된다.

애플리케이션을 종료하면 소켓 통신이 종료된다.

3.2.2 데이터 중복 방지 처리

건강 데이터는 쿼리로 원하는 시간의 샘플 데이터를 불러온다. 이때 시작시간을 정해 두지 않으면 기존에 전송되었던 데이터도 함께 전송되게 된다. 이미 전송 완료한 데이터는 다시 전송할 필요가 없기 때문에 마지막으로 전송한 데이터의 시간을 기기가 저장해 둘 필요가 있다. 또한 애플리케이션이 종료되어도 저장된 값이 유효해야하며 다시 실행 했을 경우 저장된 값을 불러올 수 있어야한다. 따라서 마지막으로 전송한 데이터의 시간을 UserDefaults에 Key-Value 형태로 저장하여 이용함으로써 데이터를 중복으로 요청하지 않도록 한다.

먼저 UserDefaults에 해당하는 시각을 저장할 변수를 만들어 준다. 이 값은 맨 처음 애플리케이션을 실행할 경우에만 시행하게 된다. 건강 데이터의 타입별로 독립적인 시간으로 데이터가 수집되며, 수집 데이터에는 StartDate와 EndDate 변수가 존재한다. 즉, StartDate부터 EndDate에 발생한 데이터를 의미한다. 애플리케이션을 종료한 뒤에 뒤이어 데이터를 전송하기 위해서는 EndDate의 값이 필요하다. 따라서 건강 데이터를 한 줄씩 전송하면서 지정된 Key의 값으로 EndDate 값으로 업데이트함으로써 데이터를 중복으로 요청하지 않도록 한다.

IV. 구현 결과

본 논문에서는 위치 데이터 수집을 위해서 iBeacon E7 모델의 비콘을 사용했으며, 건강 데이터는 애플워치 SE2를 사용하여 수집했다. 수집한 데이터는 사용자 기기인 아이폰 SE2에 저장했다. Xcode로 아이폰 애플리케이션을 구현하였다. 서버는 Node.js를 활용하여 클라이언트와 데이터베이스 연결을 각각 구현하였다.

4.1 비콘 데이터

4.1.1 수집부

시스템 동작을 확인하기 위해서 비콘 7대를 다른 공간에 배치하고, 애플워치를 착용한 뒤 동작을 테스트했다. 비콘 수집 애플리케이션을 실행하면 그림 8과 같은 화면이 나타난다. 사용자의 아이폰을 기준으로 주변에 있는 비콘에 대한 위치 데이터의 획득이 가능

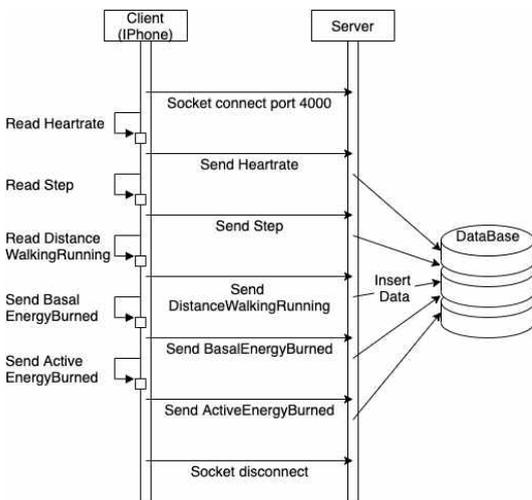


그림 7. 건강 데이터 전송 다이어그램
Fig. 7. Diagram of health data transmission

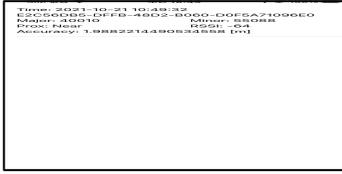


그림 8. 비콘 데이터 수집 애플리케이션 실행 화면
Fig. 8. Execution of beacon data collection application

하다. 화면에는 시각, 비콘 식별자인 UUID, Major, Minor값이 차례로 표기되며 위치에 관한 정보인 근접도, 정확도, RSSI 값이 표시된다.

4.1.2 전송-저장부

비콘 데이터를 저장하기 위해 먼저 서버의 데이터베이스에 그림 9와 같이 테이블을 구성하였다. 비콘 데이터를 전송-저장하기 위한 애플리케이션을 실행하면 그림 10과 같은 화면을 확인할 수 있다. 화면 UI는 Button과 그 아래 가져온 비콘 데이터의 시각정보가 표시되도록 구성하였다. 애플리케이션이 실행되면 먼

COLUMN_NAME	DATA_TYPE
Timestamp	timestamp
DeviceID	text
major	int
minor	int
prox	text
acc	double
rssi	int

그림 9. 비콘 데이터베이스 테이블 스키마
Fig. 9. Schema of beacon database table

Button
2021-10-21 10:48:36
2021-10-21 10:48:37
2021-10-21 10:48:38
2021-10-21 10:48:39
2021-10-21 10:48:40
2021-10-21 10:49:18
2021-10-21 10:49:19
2021-10-21 10:49:20
2021-10-21 10:49:21
2021-10-21 10:49:22
2021-10-21 10:49:23

그림 10. 비콘 데이터 전송 애플리케이션 실행 화면
Fig. 10. Execution of beacon data transmission application

저 서버와 소켓 연결을 시도한다. 아이폰에 저장된 비콘 데이터베이스에 저장된 테이블을 한 줄씩 불러오면서 서버에 그림 11의 구조로 데이터를 전송하였고, 전송된 데이터는 그림 12와 같이 서버의 로그로 확인할 수 있다. 서버에서는 이벤트가 일어나면 콜백 함수로 전송받은 데이터를 그림 13과 같이 지정된 데이터베이스에 자동으로 삽입한다.

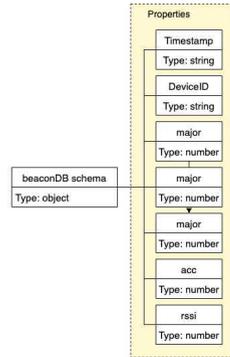


그림 11. 비콘 데이터 구조
Fig. 11. Beacon data structure

```

index Logs
index > {
index >   prox: 'Far',
index >   DeviceID: 'nurse_iPhone3',
index >   minor: 55450,
index >   Timestamp: '2021-10-21 10:49:37',
index >   rssi: -68,
index >   acc: 2.40427955517536,
index >   major: 40010
index > }
index > {
index >   prox: 'Near',
index >   DeviceID: 'nurse_iPhone3',
index >   minor: 55088,
index >   Timestamp: '2021-10-21 10:49:38',
index >   rssi: -66,
index >   acc: 2.45589025031197,
index >   major: 40010
index > }
    
```

그림 12. 서버에서 전송 받은 비콘 데이터 예시
Fig. 12. Example of beacon data sent to the server

	Timestamp	DeviceID	major	minor	prox	acc	rssi
<input type="checkbox"/>	2021-10-21 10:49:21	nurse_iPhone3	40010	55088	Near	1.94215756404386	-64
<input type="checkbox"/>	2021-10-21 10:49:22	nurse_iPhone3	40010	55450	Unknown		-1
<input type="checkbox"/>	2021-10-21 10:49:23	nurse_iPhone3	40010	55088	Near	1.91839102125547	-64
<input type="checkbox"/>	2021-10-21 10:49:24	nurse_iPhone3	40010	55088	Near	1.91213328690223	-64
<input type="checkbox"/>	2021-10-21 10:49:25	nurse_iPhone3	40010	55088	Near	1.90771947550997	-64
<input type="checkbox"/>	2021-10-21 10:49:26	nurse_iPhone3	40010	55088	Near	2.02603257401399	-66
<input type="checkbox"/>	2021-10-21 10:49:27	nurse_iPhone3	40010	55088	Near	1.99308064735327	-64
<input type="checkbox"/>	2021-10-21 10:49:28	nurse_iPhone3	40010	55088	Near	2.02559065579276	-65
<input type="checkbox"/>	2021-10-21 10:49:29	nurse_iPhone3	40010	55088	Near	1.99965874151511	-64
<input type="checkbox"/>	2021-10-21 10:49:30	nurse_iPhone3	40010	55088	Near	2.02400980904718	-65
<input type="checkbox"/>	2021-10-21 10:49:31	nurse_iPhone3	40010	55088	Near	2.00371265770218	-64
<input type="checkbox"/>	2021-10-21 10:49:32	nurse_iPhone3	40010	55088	Near	1.98822144905346	-64
<input type="checkbox"/>	2021-10-21 10:49:33	nurse_iPhone3	40010	55091	Unknown		-1
<input type="checkbox"/>	2021-10-21 10:49:34	nurse_iPhone3	40010	55088	Near	2.07701169836175	-67
<input type="checkbox"/>	2021-10-21 10:49:35	nurse_iPhone3	40010	55088	Near	2.17037598168084	-67
<input type="checkbox"/>	2021-10-21 10:49:36	nurse_iPhone3	40010	55449	Unknown		-1
<input type="checkbox"/>	2021-10-21 10:49:37	nurse_iPhone3	40010	95450	Far	2.40427955517536	-68
<input type="checkbox"/>	2021-10-21 10:49:38	nurse_iPhone3	40010	55088	Near	2.45589025031197	-66

그림 13. 서버 데이터베이스 예시 (비콘 테이블)
Fig. 13. Example of server database (beacon table)



그림 14. 전송 완료 확인 알림 창
Fig. 14. Notifications for data transmission completion

소켓 연결이 된 서버는 주기적으로 데이터베이스에 접근하여 현재 연결된 기기로부터 전송받은 시간 데이터 중 가장 마지막 시각을 아이폰으로 전송했을 때 문에 아이폰에서는 데이터베이스에 저장된 시각과 비교하여 값이 다를 경우 그림 14-(a)의 알림 창을, 값이 같은 경우 그림 14-(b)의 알림 창을 띄워줌으로써 전송 완료 상태를 확인할 수 있다. 전송 완료료를 확인하면 아이폰과 서버 간의 소켓 연결을 종료했다.

4.1.3 데이터 중복 방지 처리

클라이언트 단에서 데이터 중복을 방지하기 위해 전송 완료 후 사용자 기기에 저장된 데이터베이스 내용을 삭제한다. 서버에서는 그림 15와 같이 지정된 시각에 중복되지 않은 데이터만을 선택해서 다른 테이블로 옮긴다. 그리고 현재 테이블의 데이터는 삭제해주는 작업을 실행한다.

본 실험에서 사용자 기기에 10분간 수집·저장된 1776개의 데이터를 서버로 전송하고자 하며, 전송업의 일시 종료와 재시작으로 인하여 데이터도 전송 중

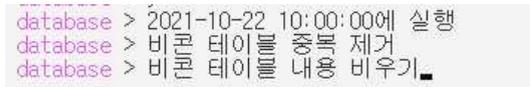


그림 15. 비콘 테이블 중복 제거 로그
Fig. 15. Log of duplicated data removal from beacon table

Timestamp	DeviceID	major	minor	prox	acc	rssl
2022-01-21 11:25:23	nurse_phone4	40010	55449	Near	1.1364636638572	-60
2022-01-21 11:25:23	nurse_phone4	40010	55100	Near	1.295486650488	-61
2022-01-21 11:25:23	nurse_phone4	40010	55088	Near	2.154434690398	-61
2022-01-21 11:25:23	nurse_phone4	40010	55449	Near	1.1364636638572	-60
2022-01-21 11:25:23	nurse_phone4	40010	55449	Near	1.1364636638572	-60
2022-01-21 11:25:23	nurse_phone4	40010	55100	Near	1.295486650488	-61
2022-01-21 11:25:23	nurse_phone4	40010	55088	Near	2.154434690398	-61
2022-01-21 11:25:23	nurse_phone4	40010	55449	Near	1.1364636638572	-60
2022-01-21 11:25:23	nurse_phone4	40010	55100	Near	1.295486650488	-61
2022-01-21 11:25:23	nurse_phone4	40010	55088	Near	2.154434690398	-61
2022-01-21 11:25:24	nurse_phone4	40010	55100	Near	1.396962608728	-64
2022-01-21 11:25:24	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:24	nurse_phone4	40010	55449	Near	1.4253905950665	-62
2022-01-21 11:25:24	nurse_phone4	40010	55100	Near	1.61428790231431	-63
2022-01-21 11:25:24	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:24	nurse_phone4	40010	55449	Near	1.4253905950665	-62
2022-01-21 11:25:24	nurse_phone4	40010	55100	Near	1.61428790231431	-63
2022-01-21 11:25:24	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:24	nurse_phone4	40010	55449	Near	1.4253905950665	-62
2022-01-21 11:25:24	nurse_phone4	40010	55100	Near	1.61428790231431	-63
2022-01-21 11:25:24	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:24	nurse_phone4	40010	55449	Near	1.4253905950665	-62
2022-01-21 11:25:24	nurse_phone4	40010	55100	Near	1.61428790231431	-63
2022-01-21 11:25:24	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:24	nurse_phone4	40010	55449	Near	1.4253905950665	-62
2022-01-21 11:25:24	nurse_phone4	40010	55100	Near	1.61428790231431	-63
2022-01-21 11:25:24	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:24	nurse_phone4	40010	55449	Near	1.4253905950665	-62
2022-01-21 11:25:24	nurse_phone4	40010	55100	Near	1.61428790231431	-63
2022-01-21 11:25:24	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:24	nurse_phone4	40010	55449	Near	1.4253905950665	-62
2022-01-21 11:25:24	nurse_phone4	40010	55100	Near	1.61428790231431	-63
2022-01-21 11:25:24	nurse_phone4	40010	55088	Far	1.939692608728	-64

그림 16. 데이터 중복 제거 전
Fig. 16. Before removing duplicate data

Timestamp	DeviceID	major	minor	prox	acc	rssl
2022-01-21 11:25:23	nurse_phone4	40010	55088	Near	2.154434690398	-61
2022-01-21 11:25:23	nurse_phone4	40010	55100	Near	1.295486650488	-61
2022-01-21 11:25:23	nurse_phone4	40010	55449	Near	1.1364636638572	-60
2022-01-21 11:25:24	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:24	nurse_phone4	40010	55100	Near	1.61428790231431	-63
2022-01-21 11:25:24	nurse_phone4	40010	55449	Near	1.4253905950665	-62
2022-01-21 11:25:25	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:25	nurse_phone4	40010	55100	Near	1.7432864466385	-64
2022-01-21 11:25:25	nurse_phone4	40010	55449	Near	1.53843648494	-63
2022-01-21 11:25:25	nurse_phone4	40010	55088	Far	1.939692608728	-64
2022-01-21 11:25:25	nurse_phone4	40010	55100	Near	1.7432864466385	-64
2022-01-21 11:25:25	nurse_phone4	40010	55449	Near	1.53843648494	-63
2022-01-21 11:25:26	nurse_phone4	40010	55088	Far	1.4336794009549	-61
2022-01-21 11:25:26	nurse_phone4	40010	55088	Near	1.722283927994	-63
2022-01-21 11:25:26	nurse_phone4	40010	55088	Near	1.74866393584257	-63
2022-01-21 11:25:26	nurse_phone4	40010	55100	Near	1.7143444888409	-63
2022-01-21 11:25:26	nurse_phone4	40010	55449	Near	1.4336794009549	-61
2022-01-21 11:25:26	nurse_phone4	40010	55088	Near	1.722283927994	-63
2022-01-21 11:25:26	nurse_phone4	40010	55088	Near	1.74866393584257	-63
2022-01-21 11:25:26	nurse_phone4	40010	55100	Near	1.7143444888409	-63
2022-01-21 11:25:26	nurse_phone4	40010	55449	Near	1.4336794009549	-61
2022-01-21 11:25:27	nurse_phone4	40010	55449	Near	1.382685447472	-61
2022-01-21 11:25:28	nurse_phone4	40010	55088	Near	1.76670728819681	-64
2022-01-21 11:25:28	nurse_phone4	40010	55100	Near	1.799992622247	-63
2022-01-21 11:25:28	nurse_phone4	40010	55449	Near	1.5445646691704	-62
2022-01-21 11:25:29	nurse_phone4	40010	55088	Near	1.7974737818014	-64
2022-01-21 11:25:29	nurse_phone4	40010	55100	Near	1.5724048824876	-64
2022-01-21 11:25:29	nurse_phone4	40010	55449	Near	1.70969899194	-68
2022-01-21 11:25:30	nurse_phone4	40010	55088	Near	1.28039004343	-68
2022-01-21 11:25:30	nurse_phone4	40010	55100	Near	1.504384344262	-61
2022-01-21 11:25:30	nurse_phone4	40010	55449	Near	1.882598004056	-66
2022-01-21 11:25:31	nurse_phone4	40010	55088	Far	1.4607979789	-58
2022-01-21 11:25:31	nurse_phone4	40010	55088	Near	1.3364242398	-57
2022-01-21 11:25:31	nurse_phone4	40010	55100	Near	1.4203346488021	-60
2022-01-21 11:25:32	nurse_phone4	40010	55088	Far	1.400384344262	-61
2022-01-21 11:25:32	nurse_phone4	40010	55089	Near	1.0513312686	-57
2022-01-21 11:25:32	nurse_phone4	40010	55100	Near	1.393698692787	-61

그림 17. 데이터 중복 제거 후
Fig. 17. After removing duplicate data

단과 재전송 과정을 거치게 된다. 이로 인하여 데이터의 중복 저장에 발생되며 이는 그림 16에서 확인할 수 있다. 그림 16의 결과에서는 전송된 데이터가 3번씩 중복되어 데이터베이스에 저장된 것을 확인할 수 있다. 하지만 본 논문에서 제안한 중복 방지 처리 방식을 적용하면 그림 17과 같이 중복된 데이터가 없이 저장되는 것을 확인할 수 있다.

4.2 건강 데이터

4.2.1 수집부

애플워치와 아이폰을 페어링 하면 주기적으로 수집된 건강 데이터는 애플워치에서 아이폰의 건강 애플리케이션으로 전송된다. 그림 18과 같이 수집된 데이터를 간단하게 확인할 수 있다.



그림 18. 건강 애플리케이션
Fig. 18. Health application

4.2.2 전송-저장부

전송할 건강 데이터는 총 5개이므로 그림 19와 같이 5개의 테이블을 만든다. 애플리케이션을 실행하면 그림 20과 같은 화면이 나타나며 데이터가 표시되면 전송할 데이터가 있다는 것을 의미한다. 가져올 건강 데이터는 UserDefaults에 저장된 시각 이후의 건강 데이터를 가져와 서버로 그림 21과 같은 구조로 전송한다. 서버에서 전송받은 데이터는 그림 22와 같이 로그 기록을 통해 확인할 수 있다.

서버에서는 이벤트 발생 후 콜백 함수로 데이터베이스에 삽입하도록 한다. 이때 건강 데이터의 타입이 5가지이지만 각 타입별로 발생 이벤트 명이 다르기

COLUMN_NAME	DATA_TYPE	COLUMN_NAME	DATA_TYPE
StartDate	timestamp	StartDate	timestamp
EndDate	timestamp	EndDate	timestamp
Energy	text	Energy	text
ID	text	ID	text

(a) 휴식 에너지

COLUMN_NAME	DATA_TYPE	COLUMN_NAME	DATA_TYPE
StartDate	timestamp	StartDate	timestamp
EndDate	timestamp	EndDate	timestamp
Energy	text	Energy	text
ID	text	ID	text

(b) 활동 에너지

COLUMN_NAME	DATA_TYPE	COLUMN_NAME	DATA_TYPE
StartDate	timestamp	StartDate	timestamp
Heartrate	text	Distance	text
ID	text	ID	text

(c) 심박 수

COLUMN_NAME	DATA_TYPE	COLUMN_NAME	DATA_TYPE
StartDate	timestamp	StartDate	timestamp
EndDate	timestamp	Distance	text
Step	text	ID	text
ID	text		

(d) 걷기 + 달리기 거리

COLUMN_NAME	DATA_TYPE
StartDate	timestamp
EndDate	timestamp
Step	text
ID	text

(e) 스텝 수

그림 19. 건강 데이터베이스 테이블 스키마
Fig. 19. Schema of health database table



그림 20. 건강 데이터 전송 애플리케이션 실행화면
Fig. 20. Execution of health data transmission application

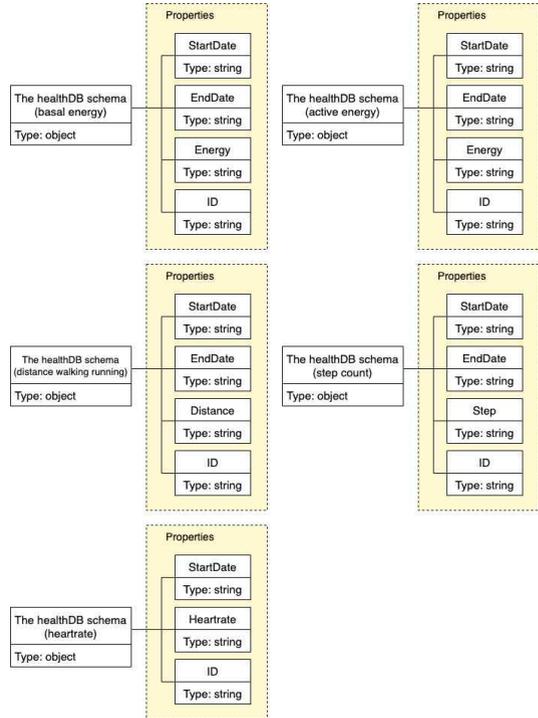


그림 21. 건강 데이터 구조
Fig. 21. Health data structure

```

index2 Logs
index2 > {
index2 >   StartDate: '2021-10-21 13:30:07',
index2 >   Energy: '1.036',
index2 >   ID: 'nurse의 Apple Watch3',
index2 >   EndDate: '2021-10-21 13:31:09'
index2 > }
index2 > {
index2 >   StartDate: '2021-10-21 13:31:09',
index2 >   Energy: '0.756',
index2 >   ID: 'nurse의 Apple Watch3',
index2 >   EndDate: '2021-10-21 13:31:40'
index2 > }
    
```

그림 22. 서버에서 전송 받은 건강 데이터 예시
Fig. 22. Example of health data transmitted to server

때문에 데이터를 원하는 데이터베이스의 테이블을 선택하여 삽입한다.

4.2.3 데이터 중복 처리

건강 데이터는 데이터 중복을 피하고 원하는 시각 영역의 데이터를 가져오기 위해 UserDefaults에 그림 23과 같이 값을 저장한 것을 확인할 수 있다. 데이터 전송이 끝나면 UserDefaults의 값이 그림 24와 같이 업데이트된 것을 확인할 수 있다. 애플리케이션을 다시 실행할 경우 기존에 저장된 값을 기준으로 데이터를 요청하게 되므로 데이터를 중복으로 요청하는 것을 방지할 수 있다.

Key	Type	Value
√ Root	Dictionary	(5 items)
LastSaveActive	String	2021-10-20 16:36:58
LastSaveHeart	String	2021-10-20 16:33:38
LastSaveStep	String	2021-10-20 17:50:58
LastSaveRun	String	2021-10-20 17:50:58
LastSaveBasal	String	2021-10-20 16:36:58

그림 23. 건강 데이터 전송 전 UserDefaults에 저장된 값
Fig. 23. Health data stored in UserDefaults (before transmission)

Key	Type	Value
√ Root	Dictionary	(5 items)
LastSaveActive	String	2021-10-21 16:09:42
LastSaveHeart	String	2021-10-21 15:18:13
LastSaveStep	String	2021-10-21 16:09:17
LastSaveRun	String	2021-10-21 16:09:17
LastSaveBasal	String	2021-10-21 16:34:14

그림 24. 건강 데이터 전송 후 UserDefaults에 저장된 값
Fig. 24. Health data stored in UserDefaults (after transmission)

V. 결 론

본 논문은 위치 정보를 제공할 수 있는 다수의 기기로부터 전달되는 위치 데이터와 사용자로부터 발생한 건강 데이터를 실시간으로 수집, 전달 및 저장할 수 있는 시스템을 구현하였다. 사용자 기기에 비콘의 위치 데이터와 애플워치의 건강 데이터를 수집하여 저장하였고, 수집된 데이터의 개수가 많아도 안정적으로 서버로 전달할 수 있음을 저장된 데이터베이스를 통해 확인했으며 중복 데이터의 발생을 방지하고 제거함으로써 데이터 관리 시스템을 구축하였다. 따라서 추후 데이터 기반의 다양한 서비스에 제한한 시스템으로 수집 및 저장된 데이터를 사용할 수 있을 것으로 기대한다.

References

[1] D. Gil, A. Ferrández, H. Mora-Mora, and J. Peral, "Internet of things: A review of surveys based on context aware intelligent services," *Sensors*, vol. 16, no. 7, 2016.

[2] Y. C. Pu and P. C. You, "Indoor positioning system based on BLE location fingerprinting with classification approach," *Appl. Math. Modelling*, vol. 62, pp. 654-663, 2018.

[3] L. Goasduff, *Gartner says global end-user spending on wearable devices to total \$52*

billion in 2020, Gartner, Retrieved Oct. 19, 2021, <https://www.gartner.com/en/newsroom/press-releases/2019-10-30-gartner-says-global-end-user-spending-on-wearable-dev>.

[4] J. W. Lockhart, T. Pulickal, and G. M. Weiss, "Applications of mobile activity recognition," in *Proc. 2012 ACM Conf. Ubiquitous Comput.*, pp. 1054-1058, 2012.

[5] D. Bae, H. S. Park, and K. Oh, "Big data trends and policy implications," *ICT & Media Policy*, vol. 25, no. 10, pp. 37-74, 2013.

[6] F. Shrouf, J. Ordieres, and G. Miragliotta, "Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm," in *2014 IEEE Int. Conf. Ind. Eng. and Eng. Manag.*, pp. 697-701, 2014.

[7] J. J. Oh, S. J. Choi, and J. S. Kim, "Development of multiple wireless communication controller for smart factory construction," *J. Korean Inst. Electr. and Electron. Mater. Eng.*, vol. 30, no. 9, pp. 602-608, 2017.

[8] H. Kim, J. H. Park, K. Y. Kim, and Y. K. Park, "Smart factory based on ICT technology," *Electron. and Telecommun. Trends*, vol. 29, no. 5, pp. 62-71, 2014.

[9] S. Im, C. S. Park, H. Kim, S. Kim, D.-Y. Lee, and K. Park, "Smart construction technology research program(part 3) integrated smart safety management technology," *J. KSCE*, vol. 68, no. 8, pp. 44-51, 2020.

[10] K. J. Vener, S. Szabo, and J. G. Moore, "The effect of shift work on gastrointestinal (GI) function: a review," *Chronobiologia*, vol. 16, no. 4, pp. 421-439, 1989.

[11] A. J. Scott and J. LaDou, "Shiftwork: Effects on sleep and health with recommendations for medical surveillance and screening," *Occupational Med. (Philadelphia, Pa.)*, vol. 5, no. 2, pp. 273-299, 1990.

[12] T. Uehata and N. Sasakawa, "The fatigue and maternity disturbances of night workwomen," *J. Human Ergology*, vol. 11, no. Supplement, pp. 465-474, 1982.

[13] W. Shin, W. H. Lee, and Y. Kim, "A study on estimating the consuming time required for the work performed by police officers, especially the detectives, based on 8 tasks in 4 field," *The Korean Assoc. Police Sci. Rev.*, vol. 22, no. 3, pp. 117-138, 2020.

[14] H. Park and Y. J. Cho, "Overview and prospects of patient centered-smart hospitals," *J. Digital Convergence*, vol. 19 no. 7, pp. 419-426, 2021.

[15] N. Gupta, S. Mujumdar, H. Patel, S. Masuda, N. Panwar, S. Bandyopadhyay, S. Mehta, S. Guttula, S. Afzal, R. Sharma Mittal, and V. Munigala, "Data quality for machine learning tasks," in *Proc. 27th ACM SIGKDD Conf. Knowledge Discovery & Data Mining*, pp. 4040-4041, 2021.

[16] C. Sharma and N. K. Gondhi, "Communication protocol stack for constrained IoT systems," in *2018 3rd Int. Conf. Internet of Things: Smart Innovation and Usages*, pp. 1-6, Feb. 2018.

[17] A. A. Yulianto, "Extract transform load (ETL) process in distributed database academic data warehouse," *APTİKOM J. Comput. Sci. and Inf. Technol.*, vol. 4, no. 2, pp. 61-68, 2019.

이 정 진 (Jungjin Lee)



2014년 2월 : 이화여자대학교 전자공학과 졸업
 2019년 8월 : 이화여자대학교 전자전기공학과 석사
 2020년 2월~2022년 12월 : 이화여자대학교 전자전기공학과 연구원

<관심분야> 머신러닝 기반 네트워크 데이터 분류
 [ORCID:0000-0002-2422-1153]

박 형 곤 (Hyunggon Park)



2004년 2월 : 포항공과대학교 전자전기공학과 졸업
 2006년 3월 : University of California, Los Angeles (UCLA) M.S.
 2008년 12월 : University of California, Los Angeles (UCLA) Ph.D.

2010년~현재 : 이화여자대학교 전자전기공학과 정교수
 <관심분야> 멀티에이전트 네트워크 시스템, 머신러닝 기반 분산적 의사 결정 전략, 게임이론 기반 네트워크 분산적 자원 관리, 네트워크 코딩
 [ORCID:0000-0002-5079-1504]