

개인 건강 정보 관리를 위한 통합 리파지토리 및 웹 기반 대시보드 UI 컴포넌트 설계 및 구현

전동철[†], 황희정^{**}

Design and Implementation of Web-based Dashboard UI Components and Integrated Repository for Personal Health Records Management

Dong-cheol Jeon[†], Heejoung Hwang^{**}

ABSTRACT

As life expectancy increases, chronic diseases become a problem, and health management in daily life becomes important. With the development of IT convergence technology, personal health records have been collected through various data sources, but they are saved separately by each medical services. Distributed storage of personal health records caused inconvenience about managing the user's health records. In this paper, we designed and implemented an integrated repository and web-based dashboard UI components to solve that inconvenience. The proposed method shows that users can manage their personal health records in the integrated repository effectively and show them through dashboard UI components.

Key words: Web, UI, u-Health, PHR, Vue, JSON, Design

1. 서 론

평균 수명이 증가함에 따라 고혈압, 관절염, 당뇨병과 같은 만성 질환이 문제가 되기 시작했다. 한국에서는 만성질환 중 하나인 고혈압으로 인한 뇌혈관과 심혈관질환 발생률은 각각 35%, 21%에 달하며, 이는 한국인 사망 원인의 2, 3위를 차지할 정도로 심각하다[1]. 이에 따라 만성 질환의 예방 및 관리에 대한 중요성이 높아지게 되고, 병원에서 받는 진단뿐만 아니라, 일상 생활에서 실천하는 건강 관리에 대해서도 관심이 높아지기 시작했다[2].

과거의 의료 서비스는 병원과 같은 의료 제공자

위주로 이루어졌으며, 환자나 일반인들은 자신들의 진료 기록과 같은 건강 기록 데이터에 대해서 접근하기 어려움이 있었다[3]. 최근에는 IT융복합기술과 클라우드 리파지토리 기술의 발전으로 인해 의료 기관과 연계된 의료 서비스나 모바일 및 웨어러블 디바이스 등의 다양한 경로를 통해서 저장된 개인 건강 기록(Personal Health Records, PHR) 관리 서비스가 화제가 되기 시작했다[4]. 하지만, 의료 기관에서 발생한 의료 기록은 각 기관별로 데이터 및 문서의 표현 형식에 차이가 존재하고, PHR 관리 서비스는 각자의 플랫폼으로만 수집되고 관리할 수 있어, 한 사람의 건강 데이터가 분산 저장되고, 이를 통합적으로

※ Corresponding Author : Heejoung Hwang, Address: (13120) Seongnamdaero 1342, Seongnam-si, Gyeonggi-do, Korea, TEL : +82-31-750-4758, E-mail : hwanghj@gachon.ac.kr

Receipt date : Oct. 23, 2019, Revision date : Nov. 20, 2019
Approval date : Nov. 15, 2019

[†] Dept. of IT Convergence Engineering, Gachon University (E-mail : aseah@naver.com)

^{**} Dept. of Computer Engineering, Gachon University

※ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2019-2017-0-06130) supervised by the IITP(Institute for Information & Communications Technology Promotion)

조회하는데 어려움이 존재했다. 이는 사람들이 새로운 의료 기관에서 의료 서비스를 제공 받을 때 기존 의료 기관의 진료 기록의 부재로 인하여 불필요한 중복 처치가 이루어지거나, 기존 의료 기관의 진료 기록을 직접 가지고 가야하는 불편함이 존재했다.

본 논문에서는 분산 저장으로 인해 발생하는 개인 건강 기록 조회 및 관리의 불편함을 해소하기 위해 통합 리파지토리 및 웹 기반의 대시보드 UI 컴포넌트를 설계 및 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구인 PHR 및 PHR 플랫폼의 대표 사례와 본 논문에서 대시보드 구현에 사용된 Vue에 대해서 소개한다. 3장에서는 통합 리파지토리 및 대시보드 UI 컴포넌트의 설계가 들어가고, 4장에서는 3장의 설계를 바탕으로 구현 및 실험을 진행한다. 마지막으로 5장에서는 결론으로 마무리를 짓는다.

2. 관련 연구

2.1 PHR

PHR은 개인으로부터 생성되는 개인 고유의 건강 기록으로, 생체 정보 등의 라이프로그, 진단 기록 및 유전체 정보 등을 모두 포함하고 있는 개인 건강과 관련된 정보를 말한다[5]. PHR은 의료 서비스를 제공자와 소비자 간의 파트너 관계를 형성해주는 결정적인 도구로 인식되고 있으며, PHR 시스템의 활용을 통해 중복 처치나 진료 과정의 감소 혹은 제거, 비용

과 시간의 절감이 가능할 것으로 기대되고 있다[6]. 최근 PHR을 활용한 서비스는 진료 기록, 운동량 정보, 라이프로그 기록, 유전체 분석 결과 등을 모두 포함하는 영역으로 확장되고 있으며, 웨어러블 디바이스 및 센서 기술 등 IT융복합 기술의 발전에 따라 다양한 헬스케어 디바이스와 연계 및 통합되어 다양한 서비스를 만들어내고 있다[7].

미국의 Fitbit, Withings는 2014년부터 피트니스에 최적화된 스마트 밴드 및 시계를 개발하여 운동량, 수면 패턴 등 기초적인 건강 데이터를 수집하기 시작했다[8-9]. Owlet은 영유아의 발에 양말처럼 신겨 심박수, 혈중 산소 농도, 체온, 수면 상태 등을 측정하는 기기인 유아 모니터를 개발했고, Cnet에서는 Hapi Fork를 개발해 음식 먹는 속도를 체크하여 사람의 손 떨림을 감지해 환자의 식습관을 관찰하고, 균형 잡힌 식습관을 가지도록 도와주는 서비스를 제공하고 있다[10-11]. 한국의 라이프시맨틱스는 IoT 디바이스 기반 중증 환자 재택 관리 서비스를 개발해 유의미한 임상적 검증을 마치고 2017년부터 서비스 상용화를 추진 중에 있다[12].

PHR Platform은 개인의 라이프로그, 진료 정보 및 유전체 정보 등을 포괄적으로 수집하고 관리할 수 있는 환경을 제공하는 플랫폼을 말한다. 대표적인 PHR Platform으로는 미국 Microsoft에서 개발한 Health Vault가 있다. Health Vault는 사용자 및 사용자 가족의 건강 정보를 수집하여 저장하고 사용 및 공유할 수 있으며, 이를 통해 건강 정보 관리를 할



Fig. 1. Monitoring system of Fitbit.

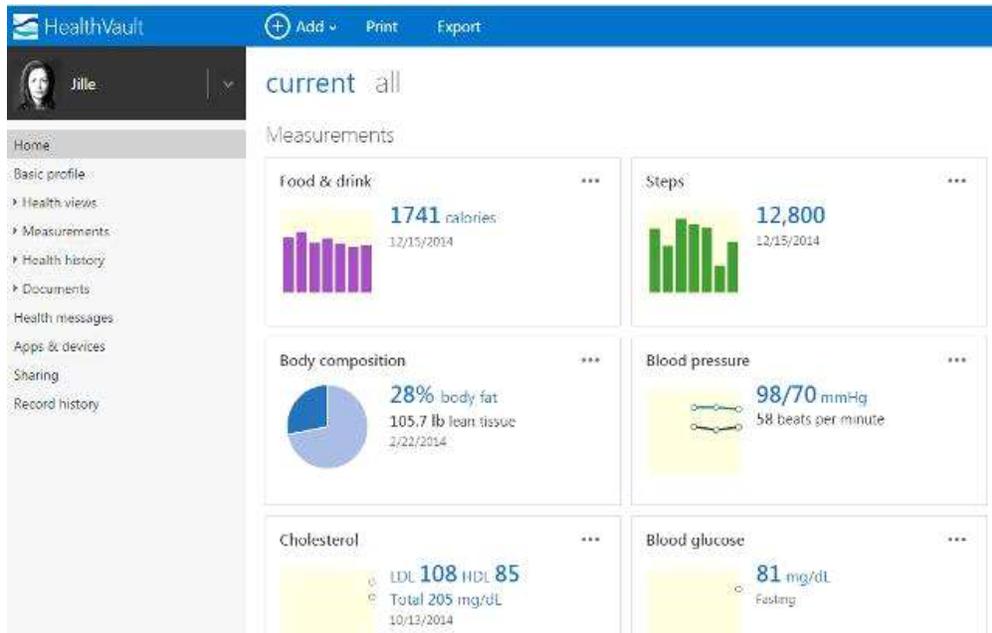


Fig. 2. Dashboard of Health Vault on desktop.

수 있는 플랫폼이다[13]. 병원과의 연계를 통해 사용자 개개인의 진료 기록을 통합 관리할 수 있는 환경을 제공하고, 사용자의 웰니스까지 관리할 수 있다는 점에서 많은 주목을 받고 있다. Health Vault의 장점은 다음과 같다.

- 가족의 건강 기록을 통합적으로 관리할 수 있다. 사용자 및 가족과 의사들이 건강 기록을 공유할 수 있으며, 사람뿐만 아니라, 애완동물의 기록까지 보관할 수 있다. 인터넷만 연결되면 어디에서나 접근이 가능하다.
- 다양한 일상 생활 속 건강 기록을 관리할 수 있다. 체중계로 측정된 몸무게 및 체지방 등의 데이터를 보내, 체중 관리에서 도움을 받을 수 있으며, 당뇨계로 측정된 혈당량 데이터를 저장해 식단을 조절하고, 의사가 필요한 경우 해당 기록을 활용할 수 있다.
- 계정과 연결된 어플리케이션을 통한 실시간 건강 모니터링이 가능하다. Health Vault는 전용 어플리케이션을 가지고 있는데, 혈압, 혈당, 활동량계 등에서 측정된 건강 정보 추가를 추가 및 관리하거나 실시간으로 모니터링 할 수 있다.

2.2 Vue

Vue는 Evan You가 개발한 자바스크립트 기반의

프레임워크로, 2014년 2월에 처음으로 릴리즈 되었다. 비슷한 프레임워크인 Angular.js, React.js에 비해서 인지도가 낮지만, 두 프레임워크의 장점을 결합한 만큼 최근에 많은 주목을 받고 있다.

Vue는 기존 소프트웨어 디자인 패턴인 MVC (Model, View, Controller) 기반이 아닌 MVVM(Model, View, View Model) 패턴 기반으로 디자인 되었으며, Vue는 그 중에서 View Model에 해당된다. View Model은 Model과 View 사이에 위치하고, Model에서 변경된 데이터를 View로 보내는 역할을 수행하는데, Vue는 이를 반응형으로 동작하게 한다. Vue의 특징은 다음과 같다.

- 데이터 바인딩과 화면 단위를 컴포넌트로 제공한다.
- Component 간 통신의 기본은 React.js의 단방향 데이터 흐름과 유사하다.
- 데이터 바인딩은 Angular.js와 동일한 양방향 데이터 바인딩을 지원한다.
- Virtual DOM을 사용하여 Virtual DOM이 변경될 때마다 실제 DOM과 비교해서 차이를 찾고 해당 부분만 업데이트한다. 따라서 실제 DOM만 사용하는 것에 비해서 빠른 성능을 보인다.
- React.js와 Angular.js에 비해서 가볍고 빠르다.

Vue의 라이프사이클은 크게 Create, Mount, Update, Destroy로 나뉘는데 각 특징은 다음과 같다 [14].

- Create : 가장 먼저 실행되며, 컴포넌트가 DOM에 추가되기 전으로, DOM에 접근할 수 없다.
- Mount : 컴포넌트가 DOM에 추가될 때 실행된다.
- Update : 컴포넌트가 재렌더링 되면 실행되는 데, 변경된 값으로 어떠한 작업을 해야할 때 유용하다.
- Destroy : 컴포넌트가 제거될 때 실행된다.

3. 통합 리파지토리 및 컴포넌트 설계

기존의 PHR 서비스 플랫폼에서는 각자 플랫폼에서 수집된 건강 기록을 해당 플랫폼 안에서만 저장할 수 있어, 한 사람의 건강 기록이 분산되어 관리되고 있었다. 또한 의료 기관의 경우에는 진료 기록이 공유되지 않아 이전의 진료 기록을 직접 찾거나, 중복 처치가 이루어지는 경우가 있었다. 사람들은 이로 인해 자신의 건강 관리에 어려움을 겪고 있었고, 이를 극복하기 위해 의료 및 개인 건강 기록을 한 저장소에 통합할 수 있는 리파지토리와 통합된 PHR을 통합해서 관리할 수 있는 대시보드가 필요하게 되었다. Fig. 3는 본 논문의 전체 프로세스 설계이다.

사용자는 자신의 웨어러블 디바이스나 병원의 진료를 통해서 건강 기록을 발생시킨다. 발생한 데이터는 건강 기록 변환 API를 통해서 통합 리파지토리의 데이터 구조에 맞게 변환된 후 리파지토리에 저장된다. 통합 리파지토리에서 사용자의 정보와 일치하는 건강 데이터를 검색한 후, 대시보드 API에 전송하고, 이는 분석 모듈을 통해서 전처리 후 대시보드에 표현된다.

3.1 통합 리파지토리 인터페이스

의료 및 개인 건강 기록은 다양한 경로를 통해서 수집이 되고, 서로 다른 데이터 표현 형식을 가지고 있다. 따라서 데이터가 수집되면 이를 하나의 형식으로 맞출 수 있도록 변환하는 기능이 필요하고, 이를 통합 리파지토리로 보낼 수 있어야 한다. 이러한 요구 사항을 처리하기 위해 본 논문에서는 Fig. 4와 같이 데이터를 통합해서 저장하고 대시보드로 보낼 수 있는 통합 리파지토리 모델을 설계했다.

다양한 플랫폼에서 건강 데이터가 발생되면, 이를 건강 데이터 수신 API를 통해서 통합 리파지토리

로 전달된다. 전달된 건강 데이터를 통합 리파지토리의 데이터 표현 형식에 맞게 변환시킨 후 사용자 정보와 건강 데이터를 매칭 시켜 데이터베이스에 저장한다. 웹 대시보드에서 건강 기록을 요청하면 요청 수신 모듈이 이를 받아서 요청 내용에서 사용자 정보를 건강 데이터 검색 모듈에 전달한다. 건강 데이터 검색 모듈은 데이터베이스에서 해당 사용자와 일치하는 건강 정보를 검색하고, 요청된 건강 데이터를 데이터 전송 API에 전달해준다. 데이터 전송 API는 해당 데이터를 웹 대시보드 시스템으로 요청에 대한 응답으로 보내준다.

의료 및 건강 데이터의 표현 형식은 플랫폼 별로 다양하게 구성이 되어있다. 이러한 요구사항에 맞추어 데이터 표현 형식에 제한이 없는 구조를 설계해야

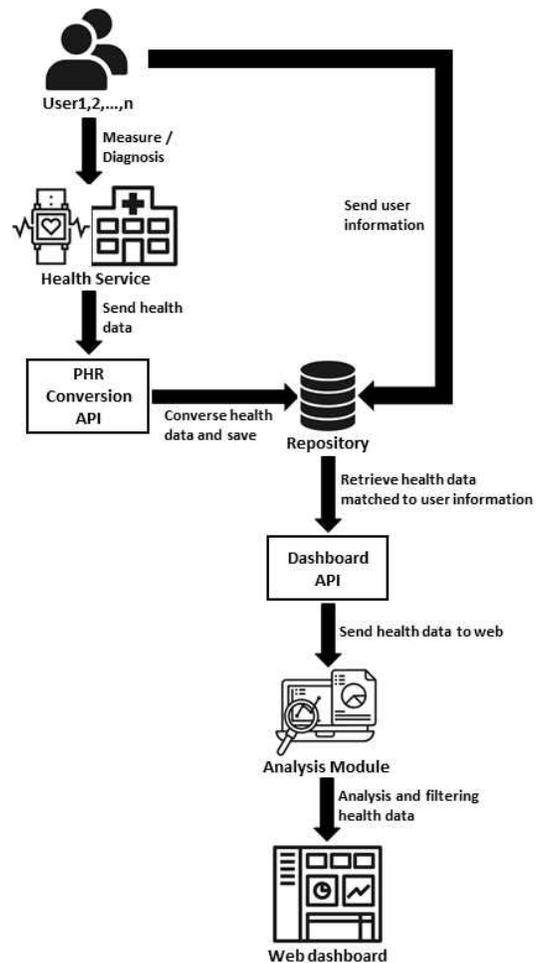


Fig. 3. Whole system process.

Table 1. Health data structure of unified repository

Name	Type	Description
_id	String	Unique key value of for that health data
uid	String	Id to identify the user in the integrated repository
did	String	Id of the department from which the health data originated
date	Date	Date when the health data is stored in the integrated repository
tags	Array	Meta-data for health data retrieval, containing keywords for actual data
data	Object	Actual health data. The representation format of data is comprised freely

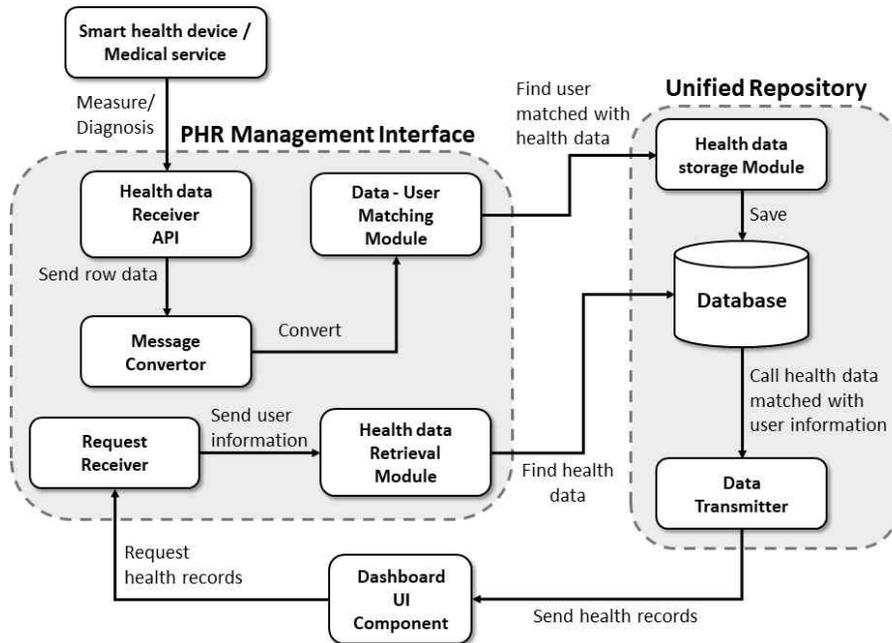


Fig. 4. System of unified repository.

한다. 따라서 본 논문에서는 비정형 데이터 구조인 JSON 형식을 사용하여 데이터 구조를 정의했으며, Table 1은 통합 리파지토리의 건강 데이터 구조 및 그에 대한 설명이다.

3.2 대시보드 UI 컴포넌트 설계

의료 및 건강 기록은 각 의료 서비스나 플랫폼을 통해서만 조회 및 관리가 가능하다. 사용자는 자신의 건강 기록을 조회하기 위해서 여러 플랫폼을 번갈아가며 조회해야 한다. 이는 사용자에게 불편함을 유발시키고, 이를 해결하기 위해서 통합 리파지토리를 통해 저장된 데이터를 효과적으로 보여줄 수 있는 대시보드가 필요하다. 본 논문에서는 리파지토리를 통해

통합된 건강 기록을 웹 기반의 대시보드를 통해서 효과적으로 조회할 수 있도록 했다. Fig. 5는 본 논문의 대시보드 UI의 전체 시스템 설계이다.

건강 기록 대시보드 UI 설계는 크게 Setting, Data Binder, Data Transfer, Dashboard UI로 구성되어 있다. Setting은 Vue 기반의 대시보드 UI 컴포넌트를 실행하기 위한 기본적인 설정을 담당하는데, 대시보드 UI 컴포넌트의 전체 환경 설정, 컴포넌트 구현에 사용되는 각종 라이브러리, 화면의 연결을 담당하는 라우터 등을 관리하는 역할을 수행한다.

Data Binder는 대시보드 UI 컴포넌트에게 통합 리파지토리의 건강 데이터를 연결해주는 역할을 수행한다. Axios 기반의 API Connector를 사용하여 건강

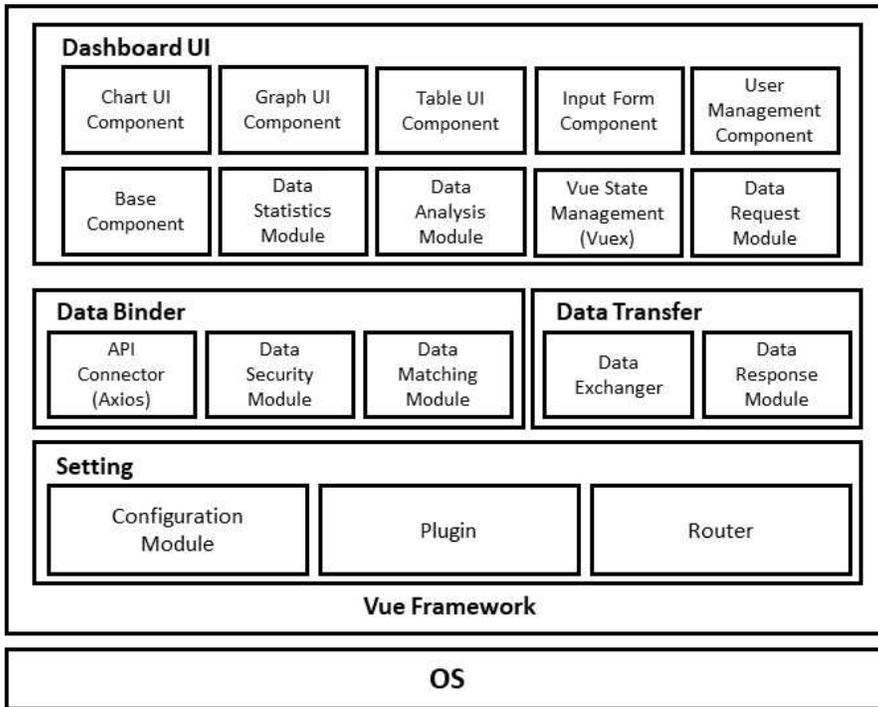


Fig. 5. Design of dashboard UI system.

데이터를 가져오고, Data Matching Module에서 사용자와 일치 여부를 판별한다. 건강 데이터는 사용자의 개인 정보를 담고 있어, Data Security Module을 통해 데이터를 암호화 시켜준다.

통합 리파지토리로부터 가져온 건강 데이터는 대시보드 UI에 표현하기 적합한 데이터의 형태가 아니기 때문에, Data Transfer를 거쳐 건강 데이터를 대시보드에 맞추어 변환해준다. Data Response Module은 대시보드 UI 컴포넌트에서 요청한 건강 데이터의 포맷을 받아 Data Exchanger로 보내주고, Data Exchanger는 해당 포맷에 맞추어 통합 리파지토리에서 가져온 건강 데이터를 변환한다. 변환된 건강 데이터는 Data Response Module을 통해 요청한 대시보드 UI 컴포넌트로 전달한다.

마지막으로 Dashboard UI는 건강 데이터를 실제 화면에 보여주기 위한 UI 컴포넌트들로 구성이 되어 있으며, 각종 UI 컴포넌트 및 데이터 요청 모듈, 상태 관리 모듈, 데이터 통계 모듈로 구성이 되어있다. Data Request Module은 실제 화면에 표현하기 위한 건강 데이터 요청 모듈로, 앞에서 설명한 Data Transfer의 Data Response Module로 데이터 요청을 보내

고, 변환된 데이터를 받아 개별 컴포넌트로 전송한다. Vuex를 사용한 Vue State Management 모듈은 Vue 기반으로 된 여러 컴포넌트들 간의 상하 관계 및 상태를 관리하기 위해 사용된다. Data Statistics 및 Analysis Module은 건강 데이터를 차트 및 표로 나타내기 위해 사용되는 통계 및 분석을 위한 모듈이다. Base Component는 대시보드 화면의 각 컴포넌트를 담기 위한 기본 컴포넌트로, 대시보드 전체의 레이아웃, 컴포넌트의 위치 및 구성 등을 담고 있다. 기타 UI 컴포넌트들은 Base Component 안에 들어가는 항목을 표현하기 위한 컴포넌트들이며, 차트, 표, 그래프 등으로 구성이 되어있다.

4. 구현 및 실험

4.1 통합 리파지토리 인터페이스 구현

본 논문에서는 3장의 통합 리파지토리 인터페이스 설계를 바탕으로 구현을 진행하였다. 서로 다른 건강 데이터 표현 형식의 한계를 극복하기 위해 비정형 데이터 구조를 기반으로 한 MongoDB를 통합 리파지토리로 사용하였으며, 대시보드 UI 컴포넌트 및

```

@PostMapping(consumes="application/json")
public ResponseEntity<ApiErrorMsg> insertDataMsg(@RequestBody
DataMsg dm, HttpServletRequest request) {
    DataMsg ret = datarepo.insert(dm);
    ApiErrorMsg msg;

    rxService.sendTotalCount();

    if(ret == null) {
        throw new DataHandleException("Save DataMsg");
    } else {
        msg = new ApiErrorMsg.Builder()
            .message("Saved Successfully")
            .status(HttpStatus.OK.value())
            .path(request.getRequestURI()).build();
    }
    logger.info("POST : insert : " + dm);
    return new ResponseEntity(msg, HttpStatus.OK);
}

```

Fig. 6. Source code about saving health data to unified repository.

외부 헬스 서비스와의 건강 데이터 연계를 위해 REST API를 사용하여 인터페이스를 구현하였다. Fig. 6는 외부 헬스 데이터로부터 건강 데이터를 받아서 통합 리파지토리에 저장하는 인터페이스에 관한 소스코드의 일부이다.

Fig. 7은 기존 Fitbit에서 발생된 심박수 데이터와 통합 리파지토리 인터페이스를 통해 변환되어 저장

된 심박수 데이터의 형태이다.

4.2 대시보드 UI 구현

개인 건강 정보 대시보드 UI 컴포넌트의 구현을 위해서는 컴포넌트 별로 관리하기 용이해야하고, 대량의 건강 데이터에 대응하여 화면이 빠르게 반응할 수 있어야 한다. 이를 위해 본 논문에서는 작은 단위로 분할해 유지보수와 재사용성이 높고, 다른 자바스크립트 기반 프레임워크에 비해 가볍고 빠른 성능을 낼 수 있는 Vue.js를 이용해 대시보드 UI 컴포넌트를 구현하였다. Vue 컴포넌트의 전반적인 소스코드의 구성은 Fig. 8과 같으며, 단일 Vue 컴포넌트 내부에서 HTML, CSS, JS를 작성해 컴포넌트 단위로 깔끔하고 쉽게 캡슐화할 수 있도록 하였다.

Fig. 9은 Vue 컴포넌트와 분석 및 통계 모듈을 담고 있는 대시보드 UI의 프로젝트 구조이다. views 디렉토리의 Dashboard.vue 파일은 각각의 UI 컴포넌트들을 배치하여 최종적으로 화면에 보여주는 대시보드 레이아웃에 대한 소스코드를 담고 있으며, components 디렉토리는 대시보드 화면에 대한 각각의 UI 컴포넌트 파일 및 설정 자바스크립트 파일을 담고 있다.

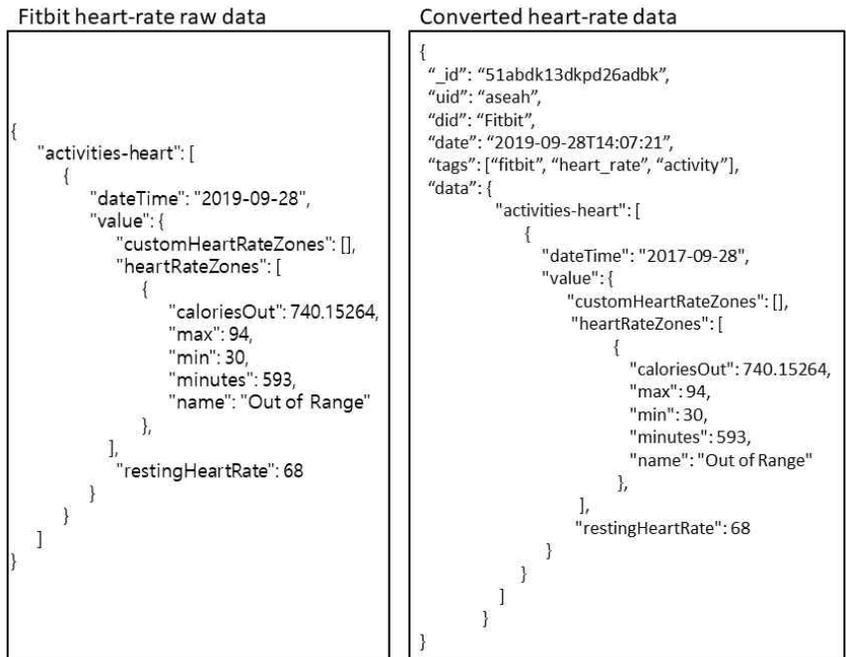


Fig. 7. Comparison between Fitbit's raw heart-rate data and converted heart-rate data.

```

<template>
    HTML code of component
</template>
<script>
    Import JavaScript file or library

    export default {
        Use Axios and take the health data from repository
        Process the health data and put the data in the variable
        Event Handling of the component
        Set the attribute of tags
    }
</script>
<style>
    Define the CSS of component
</style>
    
```

Fig. 8. Structure of source code in dashboard UI components.

4.3 실험

본 논문은 다양한 경로로부터의 건강 데이터 수집 상황을 고려하여 100명의 사용자가 4개의 의료 및 건강 기록 관련 서비스를 이용한다고 가정하고 가상 환경에서의 테스트베드를 구축하고 실험을 진행했다. Fig. 10은 전체 실험 환경에 대한 구성이다.

설계된 전체 실험 환경에서 개인 건강 정보 통합 리파지토리의 인터페이스에서 데이터 저장 및 조회를 위해 사용된 API는 5종류이며, 총 누적 API 요청 건수는 40,323 건이었다. 평균 정상 처리율은 약 98.61%였으며, 발생한 데이터의 종류는 Table 2와 같다.

```

PHR Unified Dashboard
├─ node_module
├─ src
│  └─ assets
│     └─ components
│        └─ Chart
│           ├── BarChart.js
│           ├── DounutChart.js
│           ├── LineChart.js
│           └─ HorizontalBar.js
│              └─ config.js
│                 ├── BaseInput.vue
│                 ├── BaseTable.vue
│                 ├── DashboardLayout.vue
│                 └─ SideNav.vue
│                    └─ views
│                       └─ Dashboard.vue
│                          ├── App.vue
│                          ├── config.js
│                          ├── main.js
│                          ├── router.js
│                          └─ store.js
│                             ├── babel.config.js
│                             ├── package.json
│                             └─ package-lock.json
    
```

Fig. 9. Project structure of dashboard UI component.

5종의 API 중에서 '/data/save/phr' API는 설계한 실험 환경에서 100명의 가상 사용자가 4개의 의료 및 건강 서비스를 이용한다고 가정했을 때 발생한 PHR 저장 API다. 각 사용자는 1개의 서비스별로 100개의 건강 데이터를 발생시켰으며, 발생 건수는 40,000건으로 제일 많았다. 대시보드 UI 컴포넌트의 건강 데이터 요청은 4가지로 구성이 되어있다. '/da-

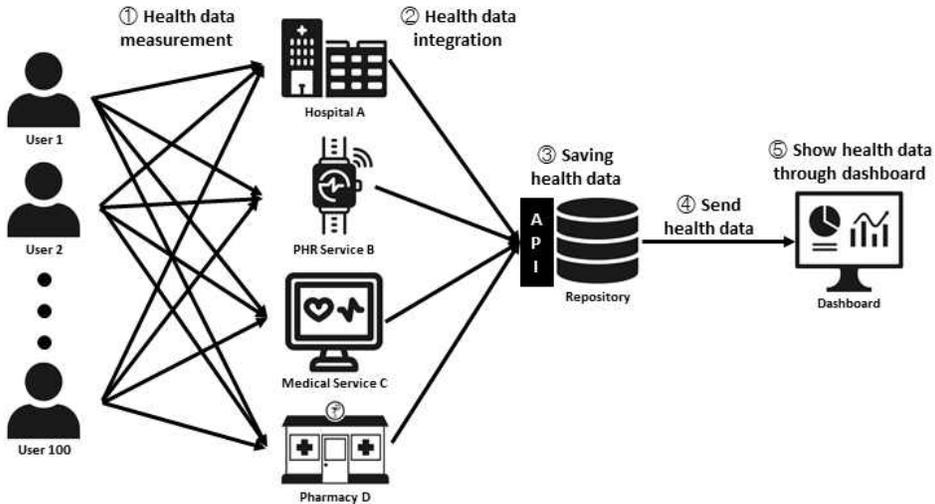


Fig. 10. Experiment configuration.

Table 2. Result of API test in experiment

Request	API URL	Total	Success	Error	Rate
Save	/data/save/phr	40,000	40,000	0	100%
Retrieval	/data/search/uid	100	100	0	100%
	/data/search/tags	120	113	7	94.16%
	/data/search/period	90	89	1	98.89%
	/data/search/services	13	13	0	100%

ta/search/uid' API는 사용자의 ID에 따른 건강 데이터 검색으로, 개별 사용자의 전체 건강 데이터를 조회하기 위해 사용된다. '/data/search/tags' API는 건강 데이터의 키워드에 따른 검색으로 키워드는 1개 부터 3개까지 늘려서 검색을 진행했다. 총 120건 중 7건의 에러가 발생하였는데, 이는 검색하는 과정에서 잘못된 키워드가 입력되어, 일치하는 건강 데이터를 찾지 못해서 발생한 것으로 확인되었다. '/data/search/period' API는 특정 기간 조건에 따른 건강 데이터 검색이다. 총 90건의 요청 중 1건의 에러가 발생하였는데, 데이터를 검색하는 과정에서 대시보드 UI 컴포넌트와 통합 리파지토리 간의 통신이 정상적으로 이루어지지 않아 발생한 에러였으며, 그 외 다른 API 요청은 성공적으로 처리했으며, 구현한 통합 리파지토리의 인터페이스가 실제 환경에서 사용되는데 문제가 없음을 확인했다.

Fig. 11은 설계에 따라 구현된 대시보드 UI 컴포넌트의 화면이다. 통합 리파지토리를 통해 수집된 건강 데이터를 바탕으로 사용자의 건강 데이터를 차트 및 그래프로 표현하며, 점이나 막대를 선택하면 자세한 수치 값을 확인할 수 있다. 또한 분석된 건강 데이터를 바탕으로 전반적인 건강 상태가 어떤지 확인할 수 있다.

Fig. 12는 사용자가 전체 건강 데이터를 조회할 수 있는 화면으로, 표 기반으로 자신의 건강 데이터를 확인할 수 있다. 또한 키워드, 특정 기간, 건강 데이터 발생 기관에 따라서 사용자가 원하는 건강 데이터를 조회할 수 있으며, 각 건강 데이터에 대해서 'View' 버튼을 눌렀을 때 세부 데이터를 조회할 수 있다.

기존 의료 및 건강 데이터 관련 서비스 플랫폼은 각각의 서비스 안에서만 데이터를 조회할 수 있었다.

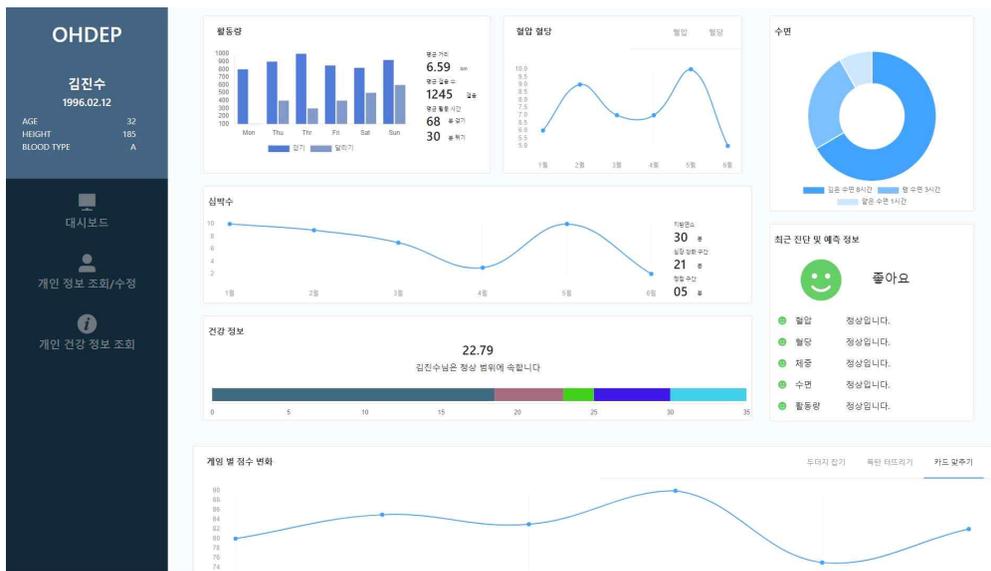


Fig. 11. Dashboard screen based on components.

DataMsg List-ALL

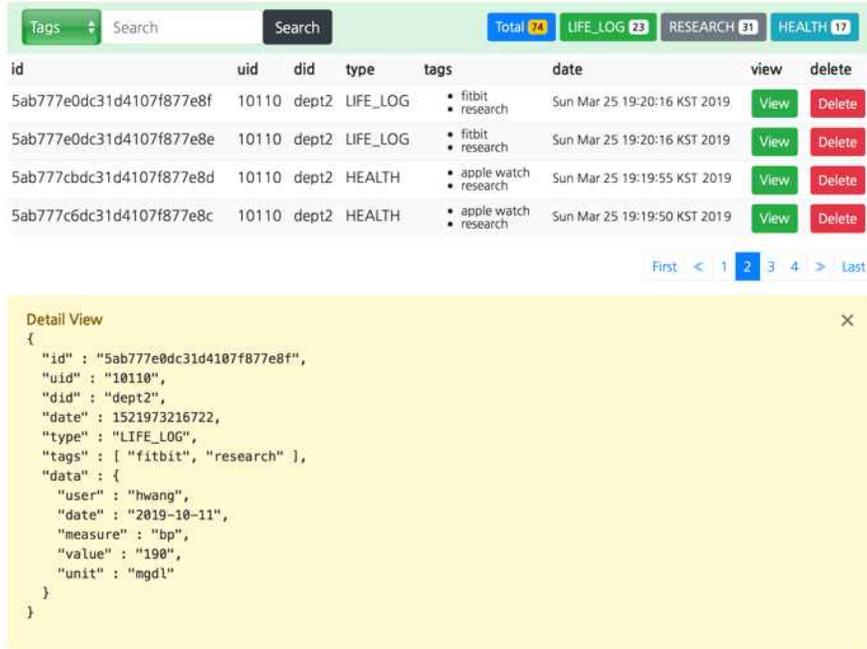


Fig. 12. User health data inquiry screen.

따라서 사용자들은 분산 저장된 자신의 건강 데이터를 조회하는데 불편함이 있었고, 이는 효율성을 떨어뜨렸다. 본 논문에서 연구한 건강 데이터 통합 리파지토리는 이러한 분산 저장된 개인 건강 기록을 통합 저장할 수 있도록 하고, 대시보드 UI 컴포넌트를 바탕으로 통합된 건강 기록을 조회할 수 있도록 가독성을 높이고, 그래프별로 자세한 수치를 볼 수 있도록 하였다. 이를 통해서 사용자는 자신의 건강이 현재 어떠한 상태인지 쉽게 파악할 수 있고, 어떤 부분을 집중적으로 관리해야 하는지를 알 수 있도록 했다. Table 3은 구현한 통합 리파지토리 및 대시보드 UI 컴포넌트와 웨어러블 디바이스 플랫폼인 Fitbit과 PHR Platform인 Health Vault 간의 기능에 대한 비

교이다.

본 연구 결과를 통해서 사용자의 건강 관련 서비스 이용에서 발생하는 모든 불편함을 해소할 수는 없었지만, 헬스 서비스의 건강 데이터 표현 형식의 차이를 극복하여 데이터를 통합 운용할 수 있는 통합 리파지토리와 이를 사용자에게 시각적으로 제공할 수 있는 대시보드 UI 컴포넌트를 통하여, 다른 건강 관련 서비스와 연계하여 사용자들이 자신의 건강 데이터를 효과적으로 조회할 수 있음을 검증하였다. 특히 특정 회사 서비스들에서 지원하지 않는 다양한 헬스 데이터 지원과 통계 및 분석 데이터 제공은 향후 다양한 건강관리 서비스에 활용될 수 있음을 확인할 수 있었다.

Table 3. Comparison between implemented dashboard UI and other health services

	Implemented dashboard	Fitbit	Health Vault
Dashboard UI	○	○	○
Mobile application	○	○	○
Connection with other health services	○	×	○
Supporting various health data format	○	×	×
Checking the detailed medical record	○	×	×
Statistics and analysis module	○	○	×

5. 결 론

최근 평균 수명의 증가로 인해 사람들의 건강 관리에 대한 관심이 높아지고, IT융복합기술이 발전하면서 다양한 건강 관리 서비스 플랫폼이 등장하였다. 하지만, 건강 관련 서비스 플랫폼은 각자의 서비스에서만 데이터를 관리하고 조회가 가능하여, 이로 인해 사람들은 의료 건강 관련 서비스를 이용하기 위해 각자의 서비스 플랫폼에 접근해서 건강 기록을 가져와야하는 불편함이 존재했다. 본 논문은 이러한 요구 사항에 맞추어 의료 및 건강 기록을 통합 관리할 수 있는 리파지토리를 설계하고 구현하여 개인 건강 기록을 한 번에 관리할 수 있게 하였으며, 대시보드 UI 컴포넌트를 통해서 통합 리파지토리에 저장된 개인 건강 기록을 사용자가 쉽게 조회할 수 있도록 하였다.

향후 연구 계획으로는 본 논문에서 구현한 통합 리파지토리와 대시보드 UI 컴포넌트를 바탕으로 개인 건강 기록 모니터링 시스템을 웹 및 모바일 기반으로 구현하는 것이며, 이를 통해 사람들이 자신의 건강 관리를 효율적으로 할 수 있을 것을 기대한다.

REFERENCE

- [1] E. Gil and H. Oh, "Testing a Middle-Range Theory of Self-Care of Chronic Illness: A Validation for Korean Adult Patients with Severe Hypertension," *Journal of Korean Academy of Nursing*, Vol. 48, No. 5, pp. 521-533, 2018.
- [2] J. Lee, G. Go, H. Kang, and H. Jung, "Web based Body Change Monitorinig System," *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 20, No. 3, pp. 615-620, 2016.
- [3] M. Yi and H. Hwang, "Design of Secure Personal Health Record Management Systems," *Journal of Korea Institute of Information Technology*, Vol. 13, No. 8, pp. 71-80, 2015.
- [4] D. Jeon, B.M. Lee, and H. Hwang, "Performance Analysis of RDBMS and MongoDB through YCSB in Medical Data Processing System Based HL7 FHIR," *Journal of Korea Multimedia Society*, Vol. 21, No. 8, pp. 934-941, 2018.
- [5] N. Archer, U. Fevrier-Thomas, C. Lokker, K. A. McKibbin, and S.E. Straus, "Personal Health Records: A Scoping Review," *Journal of the American Medical Informatics Association*, Vol. 18, No. 4, pp. 515-522, 2011.
- [6] T. Chen, C. Liu, T. Chen, C. Chen, J. Bau, and T. Lin, "Secure Dynamic Access Control Schema of PHR in Cloud Computing," *Journal of Medical Systems*, Vol. 36, No. 6, pp. 4005-4020, 2012.
- [7] R. Brooks, "e-Health Part 1: Current State of Play," *Australian Nursing Journal*, Vol. 20, No. 2, p. 20, 2012.
- [8] Fitbit Smart Band, <https://www.fitbit.com> (accessed September 28, 2019).
- [9] Withings Smart Watch, <https://www.healthmate.withings.com> (accessed September 27, 2019).
- [10] Owlet, www.owletcare.com (accessed September 27, 2019).
- [11] Hapi Fork, <https://www.hapilabs.com/product/hapifork> (accessed September 28, 2019).
- [12] Life Semantics, <https://www.lifeseantics.com> (accessed September 30, 2019).
- [13] Microsoft's Health Vault, international.healthvault.com (accessed October 01, 2019).
- [14] Vue Lifecycle, vuejs.org (accessed October 02, 2019).



전 동 철

2019년 가천대학교 컴퓨터공학과
(공학사)

2019년 ~ 현재 가천대학교 일반대
학원 IT융합공학과 컴퓨
터공학 석사과정

관심분야: Database, u-Health,
Mobile, Web



황 희 정

2000년 인하대학교 컴퓨터공학과
(공학석사)

2008년 인천대학교 컴퓨터공학과
(공학박사)

2000년 ~ 현재 가천대학교 IT대학
컴퓨터공학과

관심분야: Software Engineering, u-Health, Big Data,
Medical Informatics, Ubiquitous Computing