

System Requirements

SK Planet Interactive Web Chart Library
프로젝트 Team B

컴퓨터공학부 2012-11264 이재동
컴퓨터공학부 2011-11268 이현재
컴퓨터공학부 2011-11259 유재민

System Requirements
1. INTRODUCTION
2. SYSTEM OVERVIEW
3. SYSTEM FEATURES
4. EXTERNAL SYSTEM FEATURES
5. OTHER CONSIDERATIONS

1. INTRODUCTION

최근 들어 스마트케어에 대한 관심이 나날이 높아지고 있다. 오래전부터 다양한 신체정보를 감지할 수 있는 센서가 출시되었고, 심지어 근래 들어 출시되는 여러 휴대용 기기에 이를 위한 센서가 포함하고 있다. 세계적으로는 고령화와 건강에 대한 관심 증가로 Healthcare 관련 IoT분야가 크게 성장할 것으로 전망되고 있다. Healthcare 분야의 IoT는 2020년 까지 7억7400만개 정도가 연결될 것으로 예상되고 있다. 이러한 하드웨어의 발전과 더불어 휴대용 기기 제조사에서는 자신들이 만든 센서와 연동되는 어플리케이션을 출시하여, 건강에 관심이 많은 사용자들의 수요를 충족해왔다. 하지만 조금만 살펴봐도 이러한 어플리케이션은 진정한 의미에서의 스마트케어와는 거리가 멀다는 것을 알 수 있다. 실제로 아이폰이나 갤럭시 등에 탑재되어있는 스마트케어 관련 어플리케이션을 사용해보면 단순히 현재 정보와 간단한 통계를 제공 해줄 뿐이다. 따라서 해당 프로젝트에서는 진정한 스마트케어를 가능하게 하는 Front-end와 Back-end를 아우르는 프레임워크를 제시 및 구현하고, 센서 정보를 시각화 해주는 스마트케어 전용 chart library를 구현한다.

해당 프레임워크를 웹 기반으로 설정한 이유는 웹 환경은 핸드폰 어플리케이션에 비해 다양한 환경에서 사용이 가능할 뿐만 아니라, 스마트케어 정보를 웹에서 시각적으로 표현하기 위한 tool이 전무하기 때문이다. 인터넷에 배포되고 있는 오픈소스 웹차트 라이브러리를 통해 스마트케어 데이터를 시각화할 수는 있지만 그런 라이브러리에도 부족한 점이 많다. 신체정보의 특징에 따라 직관적으로 정보를 전달하거나 위험상태 및 정상상태를 구분할 수 없고 스마트케어 정보라는 특수성에 기인하여 차트를 시각화할 수도 없다. 이러한 현 상황에서 본 프로젝트를 통해 만드는 스마트케어 전용 chart library 및 전체적인 프레임워크는 미래의 스마트케어 시장에 유의미한 기여를 할 수 있을 것으로 생각된다.

2. SYSTEM OVERVIEW

시스템은 Sensor data를 MQTT를 통해 입력받는 INPUT 파트, 실시간으로 입력되는 input data를 이용해 웹에 차트로 나타낼 수 있도록 제공되는 LIBRARY파트, 만들어진 차트를 시각화하여 웹에 표현해주는 WEB파트 크게 3가지 파트로 구성된다. INPUT파트는 스마트기기로 부터 sensing되는 data를 MQTT 프로토콜을 통해 server에 전송해준다. 본 프로젝트에서는 실제 health data를 제공하는 스마트기기의 역할을 simulation하기 위해 shell 파일로 작성하여 실시간으로 data를 보내주는 것으로 한다. LIBRARY파트는 본 프로젝트의 핵심 파트이며, 직접 범용성을 가지는 스마트케어 전용 웹 차트 라이브러리를 구현하는 것이다. 라이브러리는 최소 3가지 이상의 차트를 만들 수 있도록 해야하며, 여러 웹 브라우저에서 사용 가능해야 한다. 기본적으로 제공되는 3가지 차트는 심박수 차트, 수면주기 차트, 체온 차트이다. 각 차트는 실시간으로 입력되는 data에 반응해야 하며, 이용자의 편의를 위해 애니메이션 기능 등을 제공한다. 또한 실시간으로 입력되는 data값을 관찰하며 위험 감지 알고리즘을 이용해 건강상태의 위험, 보통 등을 판별할 수 있으며, 위험 상황시 이용자에게 경고, 알림 기능이 제공된다. WEB파트는 기본적으로 LIBRARY를 이용해 만든 스마트케어 전용 chart를 사용자가 볼 수 있도록 해준다. 다양한 스마트기기로 접속해도 chart가 온전하게 보일 수 있도록 반응형 웹으로 구현한다.

3. SYSTEM FEATURES

MQTT는 IoT 디바이스와 서버 사이의 통신에 사용하는 프로토콜이다. SERVER는 MQTT 프로토콜을 이용하여 IoT 디바이스와 통신한다. IoT 디바이스에게 받은 데이터를 다른 형식으로 변환하여 라이브러리에 전달한다. 라이브러리에서 위험 상태 경고 요청이 오면 SMS 서버와 통신하여 사용자에게 SMS를 전송한다. 주기적으로 데이터를 정리하여 사용자의 이메일로 전송한다. 데이터베이스(MySQL)에 체계적으로 데이터를 정리한다. LIBRARY는 최소한 세 가지 기능의 차트를 포함한다. (심박수, 체온, 수면주기) 최소한 두세 가지의 브라우저와 사이트에서 동작한다. 몇몇 차트는 애니메이션, 사용자 입력에 대한 동적 반응 등을 포함한다. 모든 차트는 내부적으로 정의된 위험 상태 판단 알고리즘을 가지고 있으며 위험 상태에 다다랐다고 판단될 경우 서버에 요청을 보낸다. 주기적으로 데이터를 서버에 보내 데이터베이스 저장 및 백업을 요청한다. 모든 차트는 최소한 두세 가지의 커스터마이징 기능을 포함한다. (사진, 애니메이션, 위치 조정 등) WEB은 다양한 디바이스에서 동작할 수 있도록 브라우저 크기에 대한 동적 템플릿 변경을 지원한다. 라이브러리의 특징을 잘 보여줄 수 있도록 간단한 모양을 지닌다.

4. EXTERNAL SYSTEM FEATURES

다양한 센서로부터 MQTT 프로토콜을 이용하여 데이터를 넘겨받는다. 같은 프로토콜을 사용하지만 하면 센서의 종류나 개수에는 제한이 없다. 단지 PUBLISH하고 SUBSCRIBE하는 주제와 데이터의 형식만 같으면 된다. 서버가 MQTT 프로토콜을 이용하여 데이터를 넘겨받으면 그것을 알아서 클라이언트에 넘겨주고 클라이언트는 그 데이터를 이용하여 차트를 그린다. 기본적으로 라이브러리이기 때문에 사용할 때 여러 가지 옵션을 줄 수 있다. 옵션은 크게 두 가지 방식으로 줄 수 있다. 하나는 CONFIG.JSON 파일을 이용하여 서버측 기능에 대한 옵션을 제공하는 것이고 다른 하나는 HTML 파일 안에서 차트를 그릴 때 옵션으로 여러 가지 인자를 제공하는 것이다. CONFIG.JSON 파일에서는 위험상태감지 알고리즘의 사용여부나 변형된 알고리즘의 사용여부, SMS의 전송여부, 메일의 전송여부 등을 설정할 수 있다. HTML 파일에서는 같은 차트에 대해서도 색, 그래레이션 여부, 선의 굵기, 축의 값 표시 여부 등을 설정할 수 있다.

센서로부터 받은 데이터를 시간순서로 잘 정리해야 위험상태를 성공적으로 감지할 수 있다. 따라서 본 프로젝트에서는 MYSQL 데이터베이스를 이용하여 모든 데이터를 데이터 타입, 데이터 값, 데이터를 받은 시간에 따라서 저장한다. 라이브러리를 통해서가 아니라도 해당 데이터베이스에는 MYSQL을 이용하여 접근할 수 있다. 차트는 웹 브라우저를 통해 확인할 수 있다. 최소한 CHROME, FIREFOX, SAFARI의 세 가지 브라우저에서 정상적으로 작동하게끔 만드는 것이 목표이다. INTERNET EXPLORER에서의 동작 여부는 현재 선택사항이다. 브라우저의 창 크기에 따라 반응형으로 차트의 배치나 레이아웃이 달라지게 하는 기능 또한 현재는 선택사항이다.

5. OTHER CONSIDERATIONS

Web Client는 최신 브라우저에서 동작하고 사용자의 입력에 동적으로 반응해야 한다. Web Chart Library는 애니메이션을 포함하고 다양한 종류의 차트를 제공하고 여러 웹사이트에서 재사용이 가능한 구조로 개발되어 있어야 한다. IoT에서는 MQTT 프로토콜을 올바르게 이해하고 이를 올바르게 사용하고 웹 페이지가 실시간으로 데이터를 받아서 보여주어야 한다. Software Engineering 분야에서는 코드 계획, 데모, 리뷰, 회고의 과정을 올바르게 수행하며 개발을 진행해야 하고 팀이 정한 룰을 잘 지키고 문제점을 꾸준히 개선해야 하고 Sprint별로 업무를 나누고 매번 데모와 계획을 수립해야 한다.