Stakeholder Requirements

SK planet Interactive WEB Chart Library
프로젝트 Team E

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

Stakeholder Requirements
1. EXECUTIVE SUMMARY
2. GOAL/PROBLEM & REQUIREMENT
3. TECHNOLOGY ASSESSMENT
4. APPROACH
5. DEVELOPMENT ENVIRONMENT
6. ARCHITECTURE
7. BASIC SPEC
8. CURRENT STATUS
9. FURTHER PLAN
10. DIVISION AND ASSIGNMENT OF WORK
11. SCHEDULE

1. EXECUTIVE SUMMARY

조장 : 이재동
조원 : 유재민, 이현재

2. GOAL/PROBLEM & REQUIREMENT

현재 시중에서는 많은 차트 라이브러리들이 거래되고 있고 그중에서는 오픈소스로 공유되고 있는 것도 많다. 그러나 범용성과 개방성을 유지해야 하는 라이브러리의 특성상 차트 라이브러리가 특수한 기능을 가지기는 어렵다. 기능이 많을수록 라이브러리는 사용하고 이해하기에 어려워지기 때문이다. 따라서 시중에 라이브러리는 많지만 특별한 기능을 원하는 사람들이 찾고 있는 것들은 드물다. 헬스케어는 최근 가장 인기있는 분야 중 하나이다. 대부분의 스마트폰과 스마트워치 등의 기기에 필수적으로 생체정보 센서가 하나둘씩은 들어있다. 이제 사람들은 운동량, 수면주기, 심박수 등 스스로의 건강에 대한 정보를 스스로 확인하는 것을 당연시하게 되었다. 그러한 데이터들을 단순히 스마트폰 어플리케이션 뿐만 아니라 웹사이트에서 확인하고 싶어하는 사람들도 많다. 그러나 그러한 사람들을 위한 차트 라이브러리는 시중에 없다. 따라서 본 프로젝트는 헬스케어를 위해 특화된 라이브러리를 만드는 것을 목표로 한다. 그를 위해 범용성을 다소 희생할 수는 있겠지만 기본적인 차트 라이브러리의 기능은 모두 포함시킬 것이다.

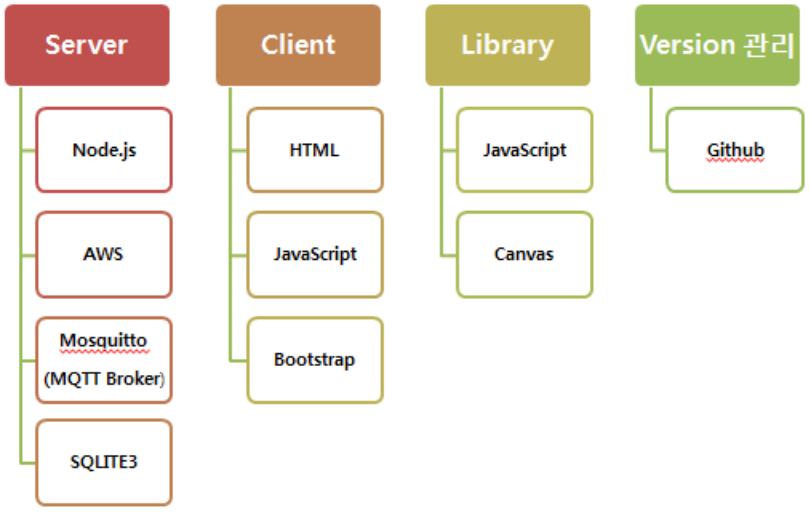
3. TECHNOLOGY ASSESSMENT

스마트케어에 대한 관심이 나날이 높아지고 있다. 현재 대부분의 휴대용 기기에는 신체정보를 감지할 수 있는 센서가 포함되어 있어서 제조사의 어플리케이션을 통해 정보를 전달한다. 그러나 그런 어플리케이션은 사용자의 특정한 요구를 반영하거나 사용자의 취향대로 최적화할 수 없다는 단점이 있다. 그런 어플리케이션은 재미로 사용하기는 쉽지만 실제로 스스로의 건강상태에 대한 실시간 모니터링이 필요한 사람들이 믿고 사용하기는 어렵다. 이에 따라 신체정보를 시각화하여 제공하고 위험상태를 경고해주는 서비스에 대한 수요가 늘고있지만 그런 서비스는 현재 거의 존재하지 않는다. 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. 따라서 본 프로젝트에서는 스마트케어 정보를 위한 웹차트 라이브러리를 만들어 웹페이지에 구현함으로서 사용자의 요구를 정확하게 반영한다. 또한 완성된 라이브러리를 자유롭게 배포함으로써 사람들이 자유롭게 수정하고 개선시킬 수 있도록 만든다.

4. APPROACH

본 프로젝트에서는 기본적으로 심박수, 체온, 수면주기에 대한 웹차트 라이브러리를 제공한다. 사용자는 제공된 함수를 통해 원하는 종류의 차트를 언제든지 추가할 수 있다. 심박수 및 체온 차트는 꺾은선 그래프를 통해 기본적인 데이터를 표시하고 최근 추세에 따라 위험상태와 정상상태를 구분하여 표시해 줄 것이다. 여기에는 사용자의 옵션에 따라 애니메이션, 색, 그림 등의 선택지를 제공한다. 위험상태가 오래 지속될 경우 사용자의 옵션에 따라 SMS 및 메일링 서비스를 통해 사용자에게 경고할 수 있고 원한다면 지정된 의사에게 연락할 수 있다. 여기에는 전문적인 통계 및 알고리즘이 사용된다. 수면주기 그래프는 특정 단위시간마다 받은 데이터를 통하여 수면시간을 깊은 수면, 얕은 수면 등으로 나누어 표시할 것이다. 이 기능들을 데모 웹페이지에 구현하여 실시간 차트로 표현한다.

5. DEVELOPMENT ENVIRONMENT



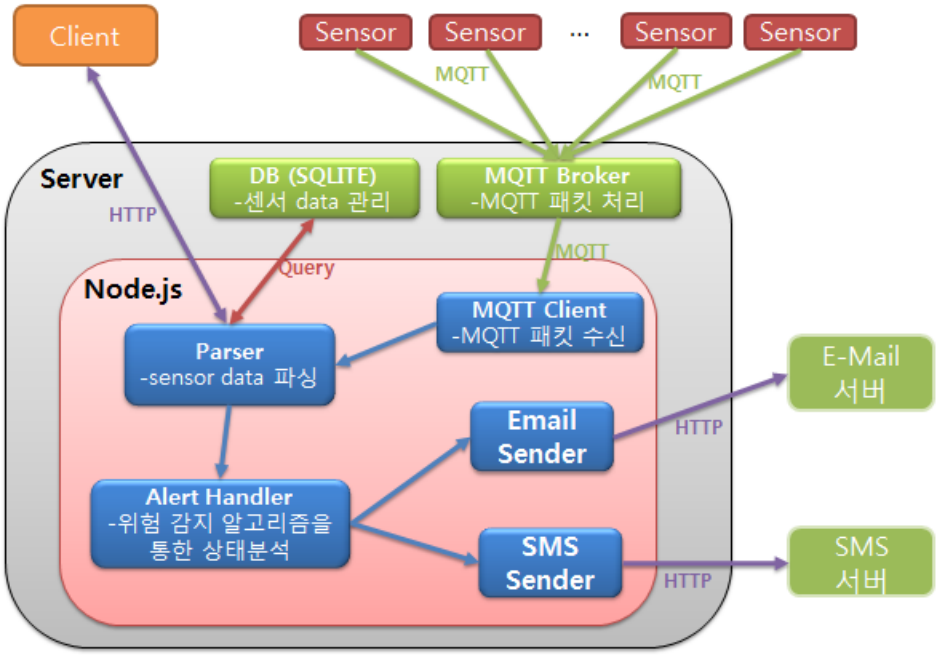
클라이언트 코드에는 INDEX.HTML과 JCHARTS.JS가 포함된다. INDEX.HTML은 웹페이지를 구성하는 HTML 파일이다. 실제로 배포될 때는 라이브러리에 포함되지 않겠지만 라이브러리의 기능을 점검하고 시연하기 위해서는 꼭 필요하다. INDEX.HTML에서 JCHARTS.JS를 불러와서 차트를 그린다. JCHARTS.JS는 실제 차트를 그리는 기능을 모두 포함하고 있다. 서버에 존재하는 JSON 파일을 불러와 파싱하여 CANVAS를 이용해 실제 그래프를 그려준다. HTML 파일을 통해 사용해야 하기 때문에 전체 파일이 HTML DOCUMENT의 WINDOW를 인자로 받는 함수로 이루어져 있다.

서버 코드는 MQTT.JS, SMS.JS, MAIL.JS, ALERT.JS로 이루어져 있다. MQTT.JS는 센서들로부터 데이터를 입력받고 그것을 저장하는 역할을 한다. 전체 데이터는 데이터를 받은 시각과 함께 MYSQL에 저장한다. 그리고 실제 그래프를 그리기 위해 필요한 부분적인 데이터는 JSON 파일을 통해 클라이언트에 넘겨준다. ALERT.JS 파일은 MQTT.JS 파일이 만든 데이타베이스를 사용하긴 하지만 MQTT.JS 파일과는 독립적으로 동작한다. 위험상태를 감지하기 위해 데이터베이스에서 최근 몇 분 ~ 몇 시간의 데이터를 읽어온 다음 사용자가 현재 위험상태에 처해 있는지 아닌지를 판단한다. 만약 위험상태에 처해있을 경우 SMS.JS와 MAIL.JS에 각각 신호를 보내어 SMS와 메일로 사용자에게 경고한다. SMS.JS와 MAIL.JS는 각각 SMS와 메일 전송을 담당하는 간단한 모듈이다.

기타 아키텍쳐에 포함된 것들로는 센서, SMS서버, 메일서버, 웹브라우저가 있다. 본 프로젝트는 유저입력이 아닌 센서데이터를 사용하기 때문에 센서가 필수적으로 필요하다. MQTT 프로토콜을 사용하기만 한다면 어떤 센서도 본 프로젝트를 위해 활용될 수 있다. 현재는 센서장비를 구하지 못했기 때문에 다른 서버에서 SHELL 파일을 이용해 보내는 가짜 데이터를 테스트에 사용한다. SMS서버는 CAFE24 호스팅을 통해 얻은 것이다. 해당 서버에 수신자, 발신자, SMS내용 등을 담아 HTTP POST REQUEST를 보내면 해당 서버가 대신 SMS메시지를 보내준다. 현재 메일은 GMAIL 계정을 통해 보내고 있기 때문에 메일서버는 구글서버이다. 기본적으로 범용성을 제공해야 하는 라이브러리이기 때문에 다양한 웹브라우저에 대해 차트가 잘 표현되어야 한다.

6. ARCHITECTURE

센서데이터를 실시간으로 서버에 전송한다. 다만 회사 측에서 아두이노 등의 지원이 불가하기 때문에 셸(Shell) 파일로 임의의 센서정보를 입력하여 가상의 센서처럼 작동시킬 것이다. 서버에 데이터를 넘겨줄 때에는 IoT 표준 프로토콜인 MQTT를 사용하여 전송한다. 이를 브로커에서 서버의 백엔드단에 전송하면 서버에서는 이를 웹 클라이언트가 읽을 수 있는 HTTP 형식으로 전송하여준다. 이를 클라이언트에서 라이브러리를 이용해 차트를 그려준다.



7. BASIC SPEC

일단 본 프로젝트는 기본적인 차트 라이브러리의 기능을 모두 지원해야 한다. 따라서 가장 핵심적인 차트인 꺾은선, 막대, 원 그래프를 필수적으로 포함해야 한다. 각 차트에 대해서는 유저 취향에 따라 어느 정도의 옵션화가 가능하게 한다. 필수적인 옵션으로는 그래프에 들어갈 데이터의 숫자, 그래프의 테마, 축의 값 표시 여부 등이 있다. 추가적인 옵션으로는 꺾은선 그래프의 interpolation 여부, 막대 그래프의 막대 굵기 등이 있다. 사용자 생체정보는 유저 입력이 아닌 센서 자체로부터 MQTT 프로토콜을 이용하여 받는다. 따라서 전송받은 데이터를 저장하고 클라이언트 측으로 넘겨줄 서버가 필요하다. 해당 서버에서는 데이타베이스(MYSQL)를 이용하여 전송된 사용자 생체정보를 저장하고 JSON 파일을 클라이언트에 넘겨줌으로써 차트에 필요한 데이터를 전달한다. 본 프로젝트는 서버를 구축하는 언어로 NODE.JS를 사용한다.

8. CURRENT STATUS

본 프로젝트를 진행하는 조원 세 명 모두 웹개발 경험이 전혀 없기 때문에 처음에는 웹에 대해 공부하는 것으로 시작했다. 먼저 HTML과 CSS를 공부하며 웹페이지를 만드는 데 익숙해진 다음 JAVASCRIPT 문법과 코딩 스타일을 익혔다. 그 다음에는 그래프를 그리는 데 필요한 HTML5의 CANVAS 라이브러리를 사용하는 방법을 공부했다. MQTT 프로토콜을 이용하여 센서와 통신해야 하므로 MQTT 프로토콜, MOSQUITTO, MQTT 브로커 연결 방법 등에 대해서도 공부해야 했고 기본적인 서버를 구축하기 위해 AWS 사용방법, NODE.JS를 이용한 서버구축 등을 익혔다.

현재 기본적인 클라이언트 및 서버의 구조가 잡힌 상태이다. 먼저 서버 측에서 가장 필수적인 모듈이며 센서와의 통신, 클라이언트로의 데이터 전달, MYSQL에의 데이터 저장을 맡는 MQTT.JS의 기본적인 기능이 완성되었다. 그 다음엔 위험상태 경고기능을 맡는 ALERT.JS와 사용자에게 경고해줄 때 필요한 SMS.JS, MAIL.JS가 각각 완성되었다. 클라이언트 측에서는 기본적인 BOOTSTRAP을 이용한 반응형 웹페이지가 만들어졌고 그래프 작성을 담당하는 JCHARTS.JS가 오류 없이 기본적인 세 가지 차트를 그릴 수 있게 되었다.

9. FURTHER PLAN

먼저 기본적인 형태로만 완성된 차트 라이브러리를 헬스케어에 특성화된 것으로 개선해야 한다. 심박수, 수면주기, 혈압에 대해 최적화된 디자인과 레이아웃을 찾아낸 다음 가장 직관적으로 정보를 전달할 수 있도록 옵션을 설정해야 한다. 그러면서도 유저의 취향에 맞게 몇 가지 디자인 선택지를 줄 수 있어야 한다. 그리고 본 프로젝트의 핵심적인 기능인 위험감지에 대해서도 알고리즘을 대폭 수정하여 더욱 정확하고 의미있는 경고기능이 되도록 한다. 본 프로젝트의 결과는 오픈소스로 배포하는 것이 목적이기 때문에 코드의 기능이 완성되었다고 하더라도 누구나 쉽게 이해하고 사용할 수 있도록 코드를 모듈화하고 개선하는 것도 빼놓지 않아야 한다. 차후 데이터부하가 심해질 경우 차트 그리기가 느려질 수 있기 때문에 MQTT 통신과 모듈간의 데이터 전달 과정을 최적화하여 서버부하를 최소로 만들어야 한다.

10. DIVISION AND ASSIGNMENT OF WORK

- 경고전송기능:유재민
- 프로토콜변환기능:유재민
- MQTT 보완:이현재
- 데모 shell 파일작성:이현재
- Web 디자인 :이재동
- 반응형웹구축 :이재동
- 심박수chart :유재민

- 체온chart :이현재
- 수면주기chart :이재동
- Chart 애니메이션 :유재민
- Chart 반응형기능추가 :이현재
- 위험감지알고리즘 :이재동

11. SCHEDULE

- 9월
  2주차: study 및 server 구축
  3주차: study 및 server 구축
  4주차: study 및 server 구축, web 디자인

- 10월
  1주차: web 디자인, MQTT 보완, 프로토콜변환기능
  2주차: 프로토콜변환기능
  3주차: 위험감지알고리즘
  4주차: 심박수 chart, 체온 chart, 수면주기 chart

- 11월
  1주차: 심박수 chart, 체온 chart, 수면주기 chart
  2주차: 심박수 chart, 체온 chart, 수면주기 chart
  3주차: 심박수 chart, 체온 chart, 수면주기 chart, chart 애니메이션, chart 반응형기능추가
  4주차: chart 애니메이션, chart 반응형기능추가.

- 12월
  1주차: chart 애니메이션, chart 반응형기능추가
  2주차: 경고전송기능, 반응형웹구축
  3주차: 데모 shell 파일작성

arch.png (20.54 KB) 이 현재, 2015/10/02 11:57
architecture.png (59.005 KB) 이 재동, 2015/11/05 22:42
dev_env.png (11.843 KB) 이 재동, 2015/11/05 22:45