

Hello, GrandMa

Created by Isabel Martinez Isabel
from Noun Project

| | | | | |
|------|---|---------|---------------|----------------------|
| 팀명 | 망고스틴 | | | |
| 담당교수 | 강희중 교수님 | | | |
| 팀원 | 이름 | 학번 | 전화번호 | 이메일 |
| | 김종우● | 1392012 | 010-7732-7511 | borama7511@naver.com |
| | 김민우 | 1292004 | 010-5436-2296 | iamminwooz@naver.com |
| | 사현빈 | 1392018 | 010-2883-1833 | smbill@nate.com |
| | 이유진 | 1433078 | 010-7175-0454 | sigma_95@naver.com |
| 홈페이지 | https://github.com/JJongwoo/CapstoneDesign2018 | | | |

1. 프로젝트 수행 목적

1-1.프로젝트 정의

1-2.아이디어 선정 배경

1-3.프로젝트 가치

1-4.개발을 통해 배울 점

2. 프로젝트 결과물 개요

2-1.프로젝트 결과

2-1-1. 프로젝트 상세

2-1-2. 프로젝트 결과물 그림

2-1-3. 기존에 있었던 것과의 차별성(외출감지, 조도감지)

2-2.프로젝트 구조

2-3.제약 조건

2-4.관련 기술

2-5.개발 도구

2-6.이용한 자료

3. 제출할 실적물

4. 프로젝트 수행 추진 체계 및 일정

4-1. 조직도

4-2. 마일스톤 및 일정

4-3. 프로젝트 주 단위 수행 일정

5. 참고 자료

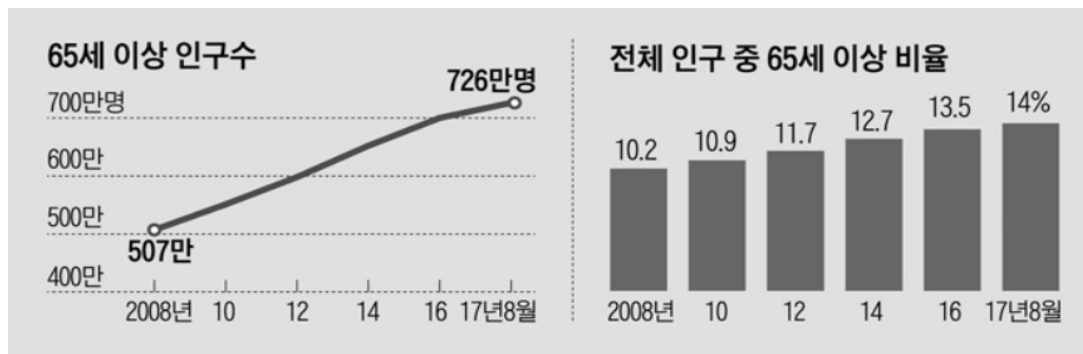
1. 프로젝트 수행 목적

1-1. 프로젝트 정의

독거노인을 감지하여 보호자 혹은 보호기관에 Data를 지속적으로 전송하여 실시간 보호 서비스를 Web과 App으로 제공하는 프로젝트이다.

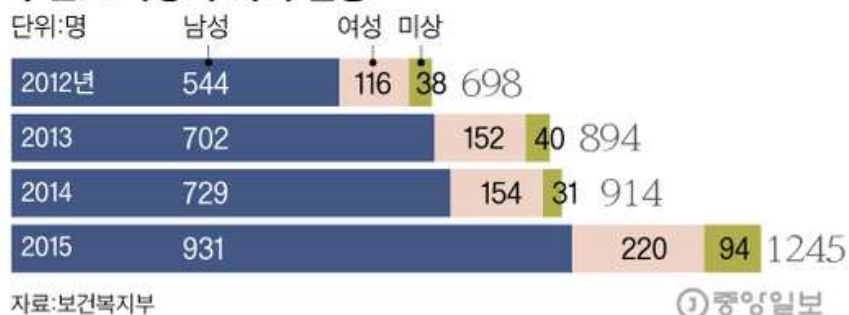
1-2. 아이디어 선정 배경

UN의 기준에 따르면 고령사회는 한 국가의 65세 이상 인구 비율이 7%를 넘으면 고령화사회, 14%를 넘어서면 고령사회이다. 행정안전부 자료에 따르면 2017년 8월을 기점으로 우리나라는 65세이상 인구비율이 14%를 넘어서면서 고령사회에 진입했다.



노인 인구가 증가함에 따라 정부 및 지방 자치 단체는 여러가지 독거노인 관리 서비스를 제공하고 있지만 예산 및 인력의 한계로 효과가 미비한 수준이다. 이에 따라 발생하는 여러가지 문제들 중 우리는 독거노인 분들의 무연고 사망, 우리가 흔히 알고있는 고독사 문제에 초점을 맞췄다.

무연고 사망자 처리 현황



위 표에 따르면 5년만에 고독사 비율이 123.5% 증가 했고, 보건복지부 자료에 따르면 최근 4년간 65세 이상 노인의 고독사는 2013년 464명에서 2017년 835명으로 증가율은 80%에 육박한다고 한다. 그래서 사물인터넷의 센서를 이용해 독거노인에게 특이사항이 발생했을 때 실시간으로 Web과 App으로 확인할 수 있다면 도움이 될 수 있다고 생각해 프로젝트를 진행하게 됐다.

1-3. 프로젝트 가치

이 프로젝트를 수행함으로써 손쉽게 독거노인 분들을 보호할 수 있는 시스템이 갖춰 지면 정부의 부족한 인력 및 예산 문제를 효율적으로 해결할 수 있을 것 이라고 생각한다.

1-4. 개발을 통해 배울 점

2. 프로젝트 결과물 개요

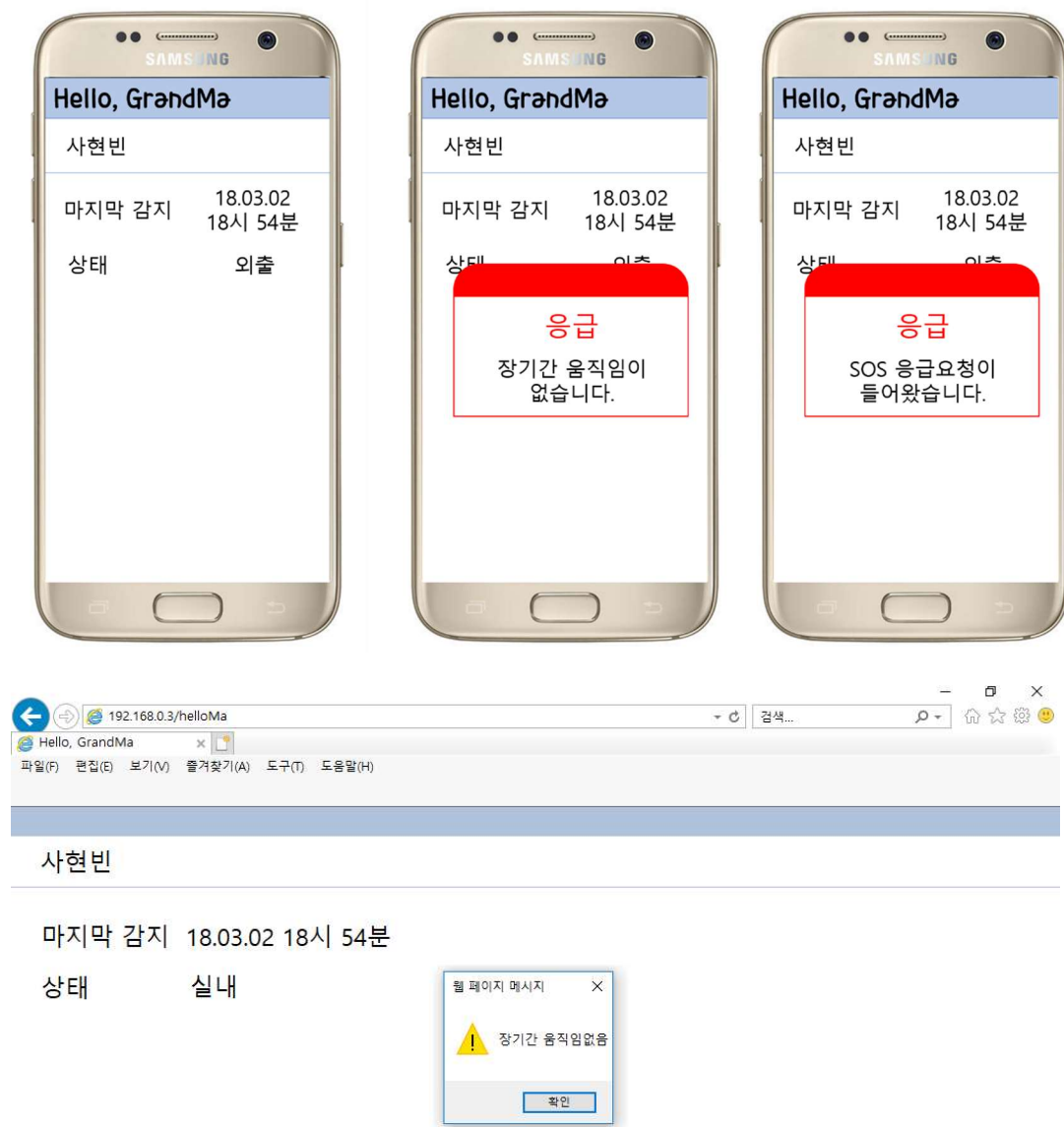
2-1. 프로젝트 결과

2-1-1. 프로젝트 상세

Machine을 웹서버에 등록 후 집에 설치하면 보호서비스가 시작되며 Machine은 독거노인의 외출, 활동량 등을 감지하며 서버에 Data를 전송한다. 이 Data들은 서버에 일단위로 저장되며, 장시간동안 움직임이 없거나 갑작스런 충격 등 특이사항 발생 시 웹사이트에 등록된 관리인에게 알림을 보낸다. 또한 독거노인 본인이 몸에 이상을 감지하면 SOS버튼을 통해 알림을 보낼 수 있다.

서버에 저장된 데이터들은 분석에 사용되어 특이사항 및 상태전환을 판단하며 서비스 가입자들은 분석된 결과와 특이사항을 일단위로 확인할 수 있다.

2-1-2. 프로젝트 결과물 화면

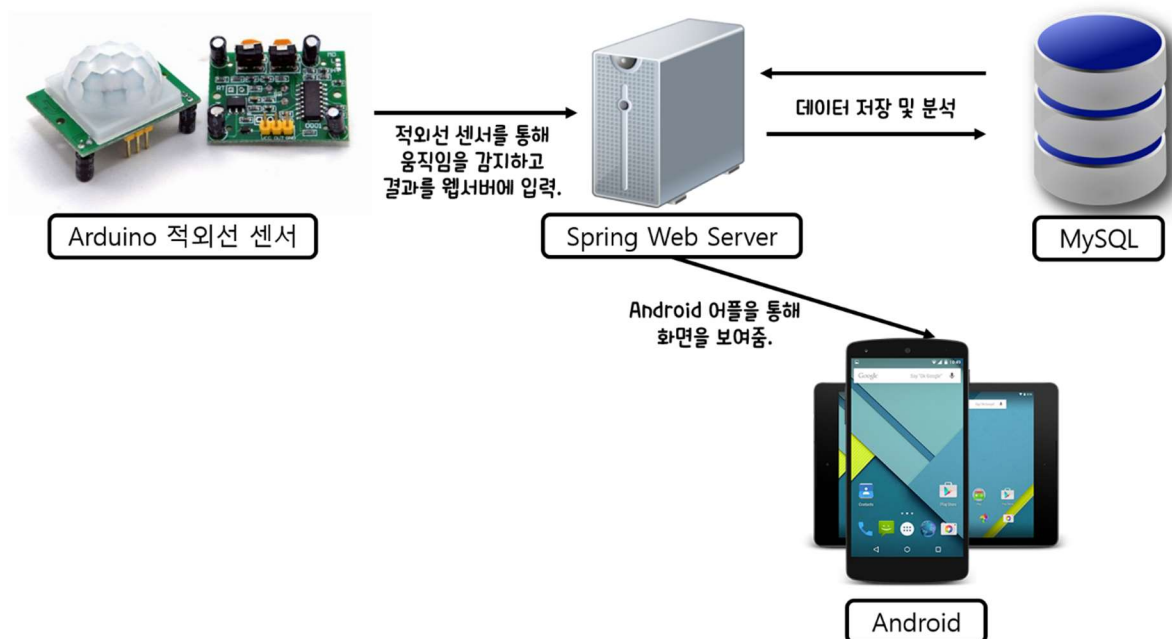


관리자 측에서 보게 되는 화면이며 App뿐만 아니라 웹 브라우저로도 확인할 수 있다.

2-1-3 기존에 있었던 것과의 차별성

기존에 있던 프로젝트들은 단지 노인분들의 죽음을 감지하는 기능만 있어 다른 특이사항들은 감지하지 못하는 단점이 있었다. 또한 외출 시 카드를 찍는다거나 건강체크를 위해 착용해야 하는 장비가 있어 실생활에 불편함을 주거나 실용적이지 못했다. 그래서 최대한 실생활에 지장을 주지 않도록 여러가지 센서의 정보를 종합적으로 분석하여 특이사항 및 상태전환(외출, 취침)을 알릴 수 있도록 하고 새로운 기능으로 SOS버튼을 추가하여 위급상황시 버튼 하나로 긴급호출을 할 수 있도록 하여 사고 예방의 효과까지 기대할 수 있다.

2-2. 프로젝트 구조



Arduino Machine은 센서로 24시간 감지를 하며 WIFI를 이용해 Web Server와 통신을 유지한다. Server로 전송된 Data를 DB에 저장 및 분석하여 특이사항이나 상태전환 발생시 서비스 가입자에게 알림을 전송한다. 서비스 가입자는 Web Browser 및 App을 통해 이를 확인할 수 있다.

2-3. 제약 조건

- 1) 경제성 : 각 가구마다 Machine을 설치해야 하므로 구매비용이 있지만 Machine비용이 크게 비싸지 않다.
- 2) 윤리성 : 노인분들의 생활을 감지하는 것이 사생활침해로 받아 들여질 수 있다.
- 3) 신뢰성 : 실세계에서 일어날 수 있는 무한한 경우의 상황들을 모두 대처하기 어려울 수 있다. 하지만 여러가지 센서모듈을 이용한 종합적인 정보처리로 다양한 상황을 대처 할 수 있도록 한다.
- 4) 미학 : 집에 설치하는 Machine은 최대한 거주자에게 자연스럽게 받아들여져야 하며 관리자가 보는 화면은 간단 명료하게 필요한 것을 볼 수 있게 디자인 한다.

2-4. 관련 기술

1) Spring Framework

자바 플랫폼을 위한 오픈 소스 애플리케이션 프레임워크로서 간단히 스프링(spring)이라고 한다. 동적인 웹 사이트 개발을 위한 여러 가지 서비스를 제공한다. EJB 기반으로 개발을 하지 않고 POJO(Plain Old Java object) 기반으로 개발을 하더라도 가볍고, 제어가 가능한 상호 관련이 적은, AOP(Aspect Oriented Programming. 관점지향 프로그래밍)을 지원하고, 컨테이너를 통해 라이프사이클을 관리하고, XML 기반으로 컴포넌트를 개발할 수 있도록 지원해주는 프레임워크를 말한다.

2) JAVA

객체지향프로그래밍 언어로서 C/C++에 비해 간략하고 쉬우며 네트워크 기능의 구현이 용이하기 때문에, 인터넷 환경에서 가장 활발히 사용되는 프로그래밍 언어이다. 자바 프로그램은 운영체제의 종류에 관계없이 대부분의 시스템에서 실행 가능하다. 자바는 널리 사용되는 프로그래밍

언어로 인기를 얻고 있으며, 특히 웹 어플리케이션 개발 언어로 가장 많이 사용되고 있다.

3) Bootstrap

부트스트랩은 동적인 웹 사이트 및 웹 응용 개발을 위한 프론트엔드 프레임워크로 입력 창, 버튼, 네비게이션 및 기타 구성물, 각종 레이아웃 등을 HTML 및 CSS 기반의 디자인 템플릿으로 제공하며 추가적인 자바스크립트 확장들도 포함한다. 모듈화 되어 있으며 LESS(CSS의 기능을 보완해주는 전처리기가자 스타일시트 언어)의 다양한 요소를 구현하고 있다. 웹 브라우저의 종류 및 크기에 따라 디자인 요소가 자동으로 정렬되는 그리드 레이아웃을 표준 설계로 사용하며 동일한 하나의 웹 페이지를 수정 없이 데스크탑, 태블릿, 스마트폰에서 모두 볼 수 있도록 지원한다.

4) JSON

JSON(JavaScript Object Notation)은 속성-값 쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다. 비동기 브라우저/서버 통신 (AJAX)을 위해, 넓게는 XML(AJAX가 사용)을 대체하는 주요 데이터 포맷이다. 특히, 인터넷에서 자료를 주고 받을 때 그 자료를 표현하는 방법으로 알려져 있다. 본래는 자바스크립트 언어로부터 파생되어 자바스크립트의 구문 형식을 따르지만 언어 독립형 데이터 포맷이다. 즉, 프로그래밍 언어나 플랫폼에 독립적이므로, 구문 분석 및 JSON 데이터 생성을 위한 코드는 C, C++, C#, 자바, 자바스크립트, 펄, 파이썬 등 수많은 프로그래밍 언어에서 쉽게 이용할 수 있다.

5) IoT

사물인터넷이란 인터넷을 기반으로 모든 사물을 연결하여 사람과 사물, 사물과 사물 간에 정보를 상호 소통하는 지능형 기술 및 서비스를 말하며 '아이오티(IoT : Internet of Things)'라 약칭되기도 한다. 원래 사물인터넷 서비스

는 기업 간(B2B) 서비스에서 출발하였으나, 이후 일반 소비자형(B2C) 서비스로 발전하면서 B2C 중심의 스마트 헬스케어 분야 등 다양한 사물인터넷 서비스가 등장하게 되었다.

6) Android

안드로이드는 구글과 핸드폰 업체들이 연합하여 개발한 개방형 모바일 운영체제이며 리눅스(Linux) 2.6 커널을 기반으로 강력한 운영체제(OS ; operating system)와 포괄적 라이브러리 세트, 풍부한 멀티미디어 사용자 인터페이스, 폰 애플리케이션 등을 제공한다. 안드로이드는 기반 기술인 '소스 코드'를 모두 공개함으로써 누구라도 이를 이용하여 소프트웨어와 기기를 만들어 판매할 수 있도록 하였다.

2-5 개발 도구

1) Eclipse

컴퓨터 프로그래밍에서 사용되는 통합 개발 환경(IDE)으로 자바 IDE 중 가장 널리 사용된다. 기본 워크스페이스 및 커스텀 환경을 위한 확장 가능한 플러그인 시스템을 포함하고 있다. 이클립스는 플러그인으로 다양한 기능을 통합할 수 있도록 설계되어 있으며, 확장성이 매우 높고 플러그인에 따라 PHP, Perl, C+++, Ruby, Python 등의 여러 언어에 대응할 수 있다. 또한 자바 디버그 인터페이스를 이용한 그래픽 디버거(Debugger)가 포함되어 있으며, 버전 관리 시스템인 C++VS, Subversion, git 등을 이용하여 소스 코드 관리를 할 수 있다.

2) Android Studio

Android Studio는 Android의 공식 IDE이다. Android용으로 제작되어 개발을 가속화하고 모든 Android기기용으로 App을 빌드하도록 도와준다. 또한 풍부한 코드편집, 디버깅, 테스트 및 프로파일링 도구를 비

롯한 맞춤형 도구를 Android개발자에게 제공한다.

3) Apache Tomcat

아파치 톰캣(Apache Tomcat)은 아파치 소프트웨어 재단에서 개발된 서블릿 컨테이너(또는 웹 컨테이너)만 있는 웹 애플리케이션 서버이다. 톰캣은 웹 서버와 연동하여 실행할 수 있는 자바 환경을 제공하여 자바 서버 페이지(JSP)와 자바 서블릿이 실행할 수 있는 환경을 제공하고 있다. 톰캣은 관리 툴을 통해 설정을 변경할 수 있지만, XML 파일을 편집하여 설정할 수도 있다. 그리고, 톰캣은 HTTP 서버도 자체 내장하기도 한다.

4) MySQL

표준 데이터베이스 질의 언어인 구조화 질의 언어(SQL: Structured Query Language)를 사용하는 공개 소스의 관계형 데이터베이스 관리 시스템(RDBMS). 매우 빠르고, 유연하며, 사용하기 쉬운 특징이 있다. 다중 사용자, 다중 스레드(thread)를 지원하고, C, C++, 에펠(Eiffel), 자바, 펄, PHP, 파이썬(Python) 스크립트 등을 위한 응용 프로그램 인터페이스(API)를 제공한다. 유닉스나 리눅스, 윈도우 운영 체제 등에서 사용할 수 있다. 램프(LAMP), 즉 리눅스 운영 체제와 아파치(Apache) 서버 프로그램, MySQL, PHP 스크립트 언어 구성은 상호 연동이 잘되면서도 공개 소스(오픈 소스)로 개발되는 무료 프로그램이어서 홈 페이지나 쇼핑몰 등 일반적인 웹 개발에 널리 이용되고 있다.

5) Arduino

Arduino는 물리적인 세계를 감지하고 제어할 수 있는 인터랙티브 객체들과 디지털 장치를 만들기 위한 도구로, 간단한 마이크로컨트롤러

(Microcontroller) 보드를 기반으로 한 오픈 소스 컴퓨팅 플랫폼과 소프트웨어 개발 환경을 말한다. 아두이노는 다양한 스위치나 센서로부터 입력 값을 받아들여 LED나 모터와 같은 전자 장치들로 출력을 제어함으로써 환경과 상호작용이 가능한 물건을 만들어 낼 수 있다. 또한 아두이노는 회로가 오픈소스로 공개되어 있으므로 누구나 직접 보드를 만들고 수정할 수 있다.

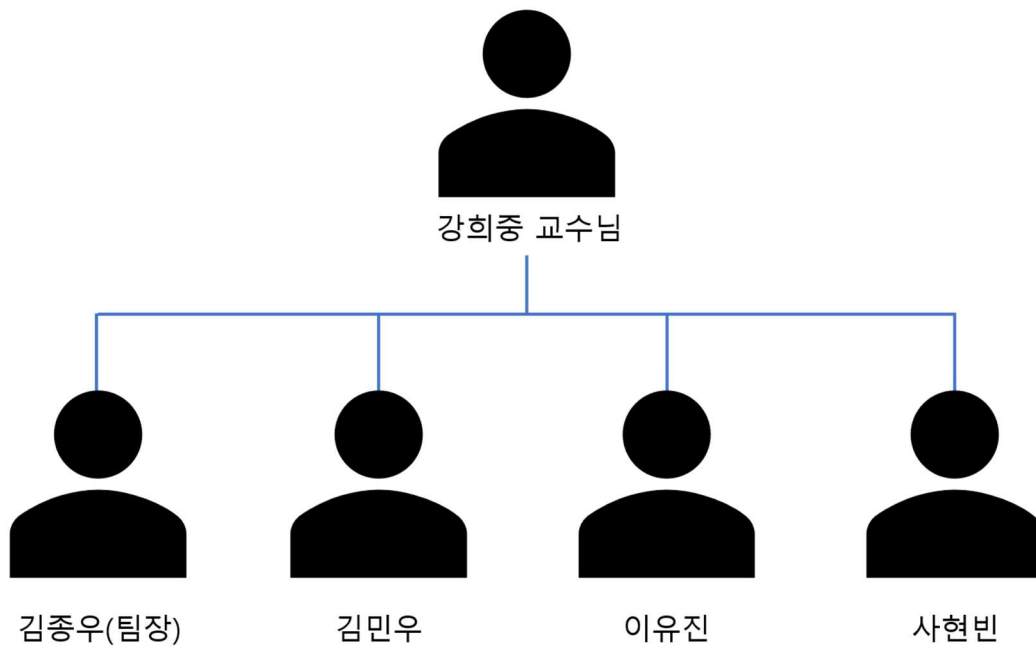
2-6 이용한 자료

3. 제출할 실적물

- 1) 프로젝트 소스/문서
- 2) 매뉴얼
- 3) Arduino Machine

4. 프로젝트 수행 추진 체계 및 일정

4-1. 조직도



4-2. 마일스톤 및 일정

| 날 짜 | 일 정 | 주요 산출물 |
|-----------------|---------------------|-----------------|
| ~3.9 (1, 2주차) | 팀 편성 및 프로젝트 확정 | |
| ~3.16 (3주차) | 프로젝트 제안서 작성 | 제안서 |
| ~3.29 (4, 5주차) | 요구사항 분석 및 프로젝트 설계 | 요구사항 명세서, 설계서 |
| ~4.25 (6~9주차) | 설계에 따른 개발 및 중간데모 준비 | 데모 파일(감지 및 전송) |
| ~5.16 (10~12주차) | 테스트 및 최종데모 준비 | 최종데모, 테스트 결과 |
| ~6.1 (13~14주차) | 현장 공개 발표, 전시 | |
| ~6.15 | 프로젝트 최종 제출 자료 준비 | Cross평가서, 포트폴리오 |

4-3. 프로젝트 주 단위 수행 일정

| | | | | | | | | | |
|--|----|--|-----|--|-----|--|-----|--|-----|
| | 공통 | | 김종우 | | 김민우 | | 이유진 | | 사현빈 |
|--|----|--|-----|--|-----|--|-----|--|-----|

| 단계 | 활동 | 3월 | | | 4월 | | | | 5월 | | | | 6월 | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 |

| | | 구분 | | | | | | 기간 | | | | | | | |
|-----|----------------------------|----|----|----|----|----|----|----|----|----|-----|-----|-----|--|--|
| | | 1월 | 2월 | 3월 | 4월 | 5월 | 6월 | 7월 | 8월 | 9월 | 10월 | 11월 | 12월 | | |
| 계획 | 아이디어 선정 | | | | | | | | | | | | | | |
| | 자료조사 | | | | | | | | | | | | | | |
| | 제안서 작성/수정 | | | | | | | | | | | | | | |
| 학습 | Android | | | | | | | | | | | | | | |
| | Arduino | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | Spring Framework | | | | | | | | | | | | | | |
| | DataBase | | | | | | | | | | | | | | |
| 설계 | Arduino | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | Server 및 Web 설계 | | | | | | | | | | | | | | |
| | DataBase | | | | | | | | | | | | | | |
| | Android | | | | | | | | | | | | | | |
| | UI | | | | | | | | | | | | | | |
| 구현 | Arduino 센서 작동 및 신호 수신 | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | WebServer(DB) 및 WebPage | | | | | | | | | | | | | | |
| | WebPage를 App으로 이식 | | | | | | | | | | | | | | |
| | Arduino 움직임 분석 알고리즘 | | | | | | | | | | | | | | |
| 테스트 | 기초 신호 수신 시험 / 중간데모 | | | | | | | | | | | | | | |
| | 움직임 분석 알고리 즘 테스트 / 최종데모 | | | | | | | | | | | | | | |
| 종료 | Cross평가 / 포트폴리 오 | | | | | | | | | | | | | | |

5. 참고자료

네이버 지식 백과 - terms.naver.com

위키 피디아 - ko.wikipedia.org