

## 종합설계 프로젝트 수행 보고서

프로젝트명	원격 제어가 가능한 신생아 건강 스마트 모빌 (Remotely Controllable Smart Mobil for newborn baby)
팀번호	S2-4
문서제목	수행계획서( ) 2차발표 중간보고서( ) 3차발표 중간보고서( ) 4차발표 중간보고서( ) 최종결과보고서( O )

2020.11.21

팀원 : 박현욱 (팀장)

김진엽

신용원

지도교수 : 교수 공기석(인)

## 문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2019.01.22.	박현욱(팀장)	1.0	수행계획서	최초작성
2020.02.22.	박현욱(팀장)	1.0	2차발표 수행보고서	최초작성
2020.04.30.	박현욱(팀장)	1.0	3차발표 수행보고서	최초작성
2020.06.26.	박현욱(팀장)	1.0	4차발표 수행보고서	최초작성
2020.11.21.	박현욱(팀장)	1.0	5차발표 수행보고서	최초작성

## 문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (4월)	학기말발표 (6월)	최종발표 (10월)
포함 내용	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I
	II. 본론 (1~3)	II. 본론 (1~4)	II. 본론 (1~5)	II. 본론 (1~7)	II
	참고자료	참고자료	참고자료	참고자료	III

이 문서는 한국산업기술대학교 컴퓨터공학부의 “종합설계” 교과목에서  
 프로젝트 “원격 제어가 가능한 신생아 건강 스마트 모빌”을 수행하는  
 예시 S2-4, 박현욱, 김진엽, 신용원이 작성한 것으로  
 사용하기 위해서는 팀원들의 허락이 필요합니다.

# 목 차

## I. 서론

1. 작품선정 배경 및 필요성 .....	4
2. 기존 연구/기술동향 분석 .....	4
3. 개발 목표 .....	6
4. 팀 역할 분담 .....	7
5. 개발 일정 .....	8
6. 개발 환경 .....	8

## II. 본론

1. 개발 내용 .....	9
2. 문제 및 해결방안 .....	11
3. 시험시나리오 .....	12
4. 상세 설계 .....	13
5. Prototype 구현 .....	22
6. 시험/ 테스트 결과 .....	38
7. Coding & DEMO .....	44

## III. 결론

1. 연구 결과 .....	55
2. 작품제작 소요재료 목록 .....	57

참고자료 .....	58
------------	----

## I. 서론

### 1. 작품선정 배경 및 필요성

신생아는 출생 전 모체의 일정한 체온 환경에서 지내다가 출생과 함께 외부 온도와 환경에 급격히 노출된다. 아동간호학회지에서 발표한 ‘신생아 감각자극에 관한 국내 연구 논문 분석에 따르면, 인생 초기에 태아가 겪는 환경과 경험은 성장 발달의 모태로 작용하기 때문에 영아기의 적절한 환경적 요인이 중요하다고 이야기한다<sup>[1]</sup>. 마찬가지로 한국아동간호학회에서 발표한 논문 ‘고위험 신생아의 저체온증 현황 및 관련 요인’에서도 신생아 주변 환경에 대한 중요성을 언급한다<sup>[2]</sup>. 논문에 따르면 전체 영아 중 최소 23%에서 최대 85%까지 저체온증이 보고되고 있다. 이처럼 주변 환경은 신생아의 건강에 직·간접적 영향을 미친다. 신생아에게 쾌적한 환경을 제공하기 위해, 신생아들과 가장 밀접한 장난감인 모빌을 응용한 ‘스마트 모빌’을 개발하여 주변 환경 데이터를 효과적으로 수집한다.

본 작품은 신생아들의 감각 자극 발달에 큰 영향을 미치는 장난감 모빌이다. 아이와 가장 가까운 위치에 있는 장난감인 만큼, 아이에게 크고 작은 영향을 미치는 환경의 데이터를 수집하고 전송한다. 부모는 온도, 습도, 미세먼지, 체온 데이터 등을 제공받음으로써 신생아에게 쾌적한 환경이 제공되고 있는가를 확인할 수 있다. 또한 모빌 동작 원격 제어, 음악 재생 기능을 사용하여 신생아의 인지 능력, 운동 능력 등 감각 자극 발달을 이루어 낼 수 있다. 추가적으로 실시간 동영상, 육아 정보, 수유등과 같은 기능을 제공하여 부모에게 편의성을 제공하는 활용도 높은 모빌로써 기능한다.

### 2. 기존 연구/기술동향 분석

사물인터넷(IoT, Internet of Things)은 사물에 센서를 부착해 실시간으로 데이터를 인터넷으로 주고받는 기술이나 환경을 일컫는다. 블루투스나 근거리무선통신(NFC), 센서데이터, 네트워크가 사용되어 사물을 조작하거나, 제어하는 기술이다. 현재 세탁기, 냉장고, 정수기와 같은 대부분의 가전제품에서 활용되고 있고, 글래스, 워치 같은 웨어러블 기기에도 활용되고 있다. 현재 신생아를 위한 장난감인 모빌에도 IoT기술이 사용되고 있다. 현재 가장 많이 사용되는 2가지 스마트 모빌은 다음과 같다.

기존의 유사제품으로는 [그림1]선플라워 스마트 모빌이 있다. 카메라를 통해 애플리케이션에서 스트리밍 영상을 확인할 수 있는 기능을 구현하였다. 신생아의 감각자극 발달을 위한 5가지의 멜로디를 선택할 수 있도록 구현하였으며 LED 기능,

모터를 이용한 모빌 회전이 가능하다.



[ 그림 1 : 썬플라워 스마트 모빌, (주)코니스 ]

두 번째 유사제품은 [그림2]큐티 뽀띠 스마트 오르골이다. 이 제품은 단순히 멜로디 재생, 선택 기능만을 가지고 있다. 현재 상용화된 스마트 모빌들을 살펴본 결과 본 작품에서 개발하고자 하는 온습도 센서, 미세먼지센서를 활용한 데이터를 제공하는 사례는 없었다.



[ 그림 2 : 큐티 뽀띠 스마트 오르골, HUILIDUO ]

썬플라워 스마트 모빌과 큐티뽀띠 스마트 오르골을 분석한 결과, 본 작품에서 개발하고자 하는 스마트 모빌과는 다음 [표1]과 같은 차이점이 존재하였다. 개발하고자 하는 ‘원격 제어가 가능한 신생아 건강 스마트 모빌’은 타 제품과 비교하였을

때, 온·습도 및 미세먼지 농도 데이터 측정이 가능하다는 점과 비접촉식 온도센서를 이용하여 신생아의 체온을 측정할 수 있다는 점에서 차이점이 존재하였다.

[ 표 1 : 기존 제품과 개발 대상 작품 기능 비교 ]

제품	카메라	온습도	수유등	음악 재생	미세먼지 농도	비접촉식 온도센서	모빌 동작제어
[그림1]썬플라워 스마트모빌	○	×	○	○	×	×	○
[그림2]큐티쁘띠 스마트오르골 스마트 모빌	×	×	×	○	×	×	○
(개발 대상 작품)	○	○	○	○	○	○	○

### 3. 개발 목표

모빌에 라즈베리파이, 아두이노, 모듈들을 부착하여 애플리케이션을 통해 각 모듈들을 제어하는 것을 개발 목표로 한다. 라즈베리파이와 아두이노는 시리얼 통신을 이용해 데이터를 송·수신하며, 라즈베리파이에서 웹 서버를 통해 애플리케이션으로 데이터를 송·수신한다. 모듈들을 통해 구현할 기능들은 다음과 같다.

첫 번째는 스트리밍 기능이다. 모빌에 라즈베리파이 전용 카메라가 부착된다. UV 4L 영상 스트리밍 전용 서버를 구축하여 영상 스트리밍 서비스가 가능하도록 한다. 애플리케이션을 통해서 사용자(부모)가 실시간 스트리밍 영상을 확인할 수 있다. 회사에서 일하는 부모님, 혹은 다른 지역에 사는 할머니, 할아버지들도 모빌에 부착된 카메라를 통해 아이의 모습을 실시간으로 볼 수 있다.

두 번째는 음악 재생 기능이다. 아이의 감각자극 발달을 위한 다양한 멜로디를 선택해서 재생할 수 있도록 한다. 기본적으로 상황에 따라 15가지 정도의 멜로디를 선택하여 재생할 수 있도록 구축할 예정이다.

세 번째는 조명 기능이다. LED 모듈은 전등으로써 기능한다. LED 모듈은 상황에 따라 수유등, 무드등 등으로 기능할 수 있어 사용자의 편의성을 높인다.

네 번째는 온·습도센서와 미세먼지 센서를 이용하여 환경 데이터 값을 애플리케이션을 통해 확인할 수 있도록 구현한다. 사용자(부모)는 애플리케이션을 통해 데이터 값을 확인할 수 있기 때문에 신생아에게 쾌적한 환경을 제공하는 데에 도움을 얻을 수 있다.

다섯 번째는 애플리케이션을 통해 모빌을 원격제어함으로써 부모가 원거리에서도 아이의 관심을 유도할 수 있고, 아이의 감각자극 발달에 도움을 줄 수 있다. On/off 기능과 방향제어, 속도 제어 기능을 구현한다.

여섯 번째는 애플리케이션은 사용자(부모)에게 뛰어난 편의성을 제공하도록 한다. 애플리케이션 내부에서 각 모듈을 통해 수집된 주변 환경 데이터의 통계치를 계산하여 적정 환경과 환경에 대한 권장량을 정보를 제공한다. 또한 신생아를 키우는데 필요한 다양한 육아정보 및 유의사항을 제공한다.

#### 4. 팀 역할 분담

[표2]의 내용은 팀 역할 분담 내용이다. 박현욱은 통신데이터 수집/전송, 음악출력, 통신, 웹 서버 구축, 웹 전반 구현을 담당한다. 김진엽은 온습도, 미세먼지 데이터 수집/전송, 비접촉식온도센서, 애플리케이션제작, 데이터베이스 구축을 담당한다. 마지막으로 신용원은 카메라 모듈, 스트리밍 서버 구축, LED와 서버모터 제어, 앱 프론트엔드를 담당한다.

[ 표 2 : 역할분담 ]

박현욱	<ul style="list-style-type: none"> <li>• 아두이노-라즈베리파이 통신데이터 수집/전송</li> <li>• 음악출력</li> <li>• 라즈베리파이-앱 통신</li> <li>• nodejs웹 서버 구축</li> <li>• 프레임설계</li> <li>• 웹 백엔드구현</li> </ul>
김진엽	<ul style="list-style-type: none"> <li>• 온습도데이터 수집/전송</li> <li>• 미세먼지데이터 수집/전송</li> <li>• 비접촉식온도센서</li> <li>• 애플리케이션제작</li> <li>• 프레임설계</li> <li>• 데이터베이스구축</li> </ul>
신용원	<ul style="list-style-type: none"> <li>• 카메라모듈 스트리밍</li> <li>• UV4L영상 스트리밍 서버 구축</li> <li>• LED 원격제어</li> <li>• 서버모터원격제어</li> <li>• 앱프론트엔드구현</li> <li>• 프레임제작</li> </ul>

## 5. 개발 일정

[표3]은 계획단계부터 최종모델링 단계까지의 추진 계획이다. 1월 말까지는 하드웨어의 구현을 완료하고 더불어 소프트웨어 개발을 4월까지 마무리한다. 이후 테스트와 유지보수 등을 진행한다.

[ 표 3 : 수행 일정 ]

	추진 사항	12	1	2	3	4	5	6	7	8	9
계획단계	개발 과정 토의 코딩 규칙 확립										
요구분석단계	요구사항 분석 자료 조사										
설계단계	하드웨어 설계/구현 (아두이노, 라즈베리파이)										
구현단계	소프트웨어 설계/구현 (앱, 서버, DB)										
테스트단계	장치, 서버, App 연동 간이 모델링										
유지보수단계	환경 제약 처리, 디버깅										
최종모델링	최종 테스트 데모 시연 준비										

## 6. 개발 환경

개발 환경으로는, [표4]와 같이 크게 7가지로 구분할 수 있다. 라즈베리파이는 라즈베리 OS와 VNC viewer을 사용한다. ubuntu 16.04로 교차개발환경을 구축하여 개발한다. 아두이노 통합개발환경으로 아두이노 IDE 1.8.10을 사용하고, 안드로이드 스튜디오 3.4.1과 데이터베이스 환경으로 MySQL Workbench 8.0 CE를 사용한다. 여러 시리얼통신 데이터를 보여줄 서버로 node.js 12.13.0을 사용하고, 실시간 동영상 스트리밍을 제공할 서버로 UV4L을 사용한다

[ 표 4 : 개발 환경 ]

라즈베리파이	라즈비안 OS, NC-Viewer
ARDUINO	아두이노 IDE 1.8.10
Ubuntu	VmWare 가상머신, Ubuntu 16.04 (교차 개발환경)
Android Studio	Android Studio 3.4.1
My SQL	MySQL Workbench 8.0 CE
Node.js	v12.13.0
UV4L	UCV Driver, XScreen Driver, MJPEG Stream Driver



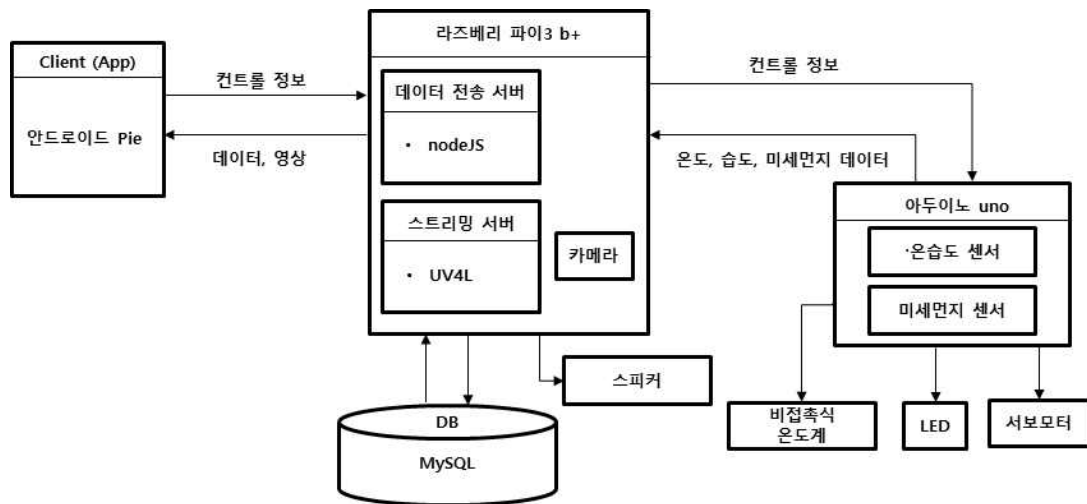
## II. 본론

### 1. 개발 내용

각 SoC보드와 애플리케이션은 다음과 같은 통신 방법을 이용한다. 아두이노와 라즈베리파이는 시리얼통신을 통해 데이터를 송·수신한다. 라즈베리파이와 애플리케이션은 node.js서버를 통해서 데이터를 송·수신한다. 스트리밍 영상은 라즈베리파이에 구축 가능 한 UV4L서버를 이용하여 영상 데이터를 전달한다.

아두이노에 사용될 온습도 센서, 서보 모터, 비접촉식 온도센서, 먼지센서, LED 모듈은 아두이노 IDE를 통해 개발한다. 라즈베리파이에서는 Java Script를 이용하여 node.js 서버를 구축한다. node.js 서버를 이용해 데이터를 송·수신한다. 애플리케이션은 안드로이드 스튜디오, 타겟 디바이스로는 갤럭시 S8을 이용해 개발한다. 아두이노에서 미세먼지, 온습도 데이터, 비접촉식 온도 센서를 이용한 온도 데이터를 시리얼 통신을 통해 전달한다. 전달받은 데이터를 node.js 웹서버를 이용해 애플리케이션에 전달한다. 환경 데이터는 이처럼 아두이노에서 애플리케이션까지 시리얼 통신, 웹 서버를 통해 전달된다. 컨트롤 정보인 음악, LED, 서보모터에 대한 동작 정보를 애플리케이션에서 node.js 서버를 통해 라즈베리파이에 전달한다. 라즈베리파이에 전달된 동작 정보는 시리얼통신을 통해 아두이노에게 전달하게 된다. 스트리밍 영상은 UV4L서버를 통해 라즈베리파이에서 애플리케이션으로 전달된다.

[그림 3]은 시스템 구성도이다. SoC보드는 아두이노 Uno와 라즈베리파이 3 B+를 사용한다. 아두이노 Uno 보드에는 온습도 센서, 미세먼지 센서, 비접촉식 온도계, LED, 서보모터 모듈이 동작 가능하도록 구현한다. 라즈베리파이 3 B+에는 카메라, 스피커 모듈이 동작 가능하도록 구현한다. 아두이노 Uno와 라즈베리 3 B+는 시리얼 통신을 통해 컨트롤 정보와 온도, 습도, 미세먼지 데이터를 송·수신한다. 라즈베리 3B+에 송신된 데이터들은 MySQL을 사용한 DB에 저장된다. 라즈베리파이를 통해 데이터베이스에 저장된 데이터들은 node.js로 구현한 웹 서버를 통해 애플리케이션에 실시간 전송된다. 스트리밍 서비스는 UV4L 전용 서버를 별개 구축하도록 하여 카메라 모듈을 통해 애플리케이션에 실시간 송신한다.



[ 그림 3 : 시스템 구성도 ]

[그림 4]는 하드웨어 구성도이다. 아두이노 Uno 보드에는 모빌 주변 주요 환경 데이터를 수집할 수 있는 온습도 센서, 서보 모터, 비접촉식 온도센서, 먼지 센서, 상황별 전등 기능을 위한 LED 모듈이 동작 가능하도록 구현한다. 라즈베리 파이에 는 실시간 영상 스트리밍을 위한 카메라 모듈, 음악 재생을 통한 스피커를 동작 가능하도록 구현한다.



[ 그림 4 : 하드웨어 구성도 ]

## 2. 문제 및 해결방안

개발 단계에서 예상되는 문제로는 다음과 같은 문제가 존재한다. 탑재되는 모듈 수로 인한 보드 전력 공급 문제, 영상 스트리밍 서버 전송 지연, 다수의 시리얼 데이터 혼란, 외관 완성도이다.

제시하는 해결 방안은 다음과 같다. 보드 전력 공급 문제에 대하여, 현재 센서

전력 문제가 일어날 수 있는 모듈로는 서보모터와 LED 모듈이 존재하기 때문에 외부전력(4.8~7.2V 배터리)을 공급하여 해결하도록 한다.

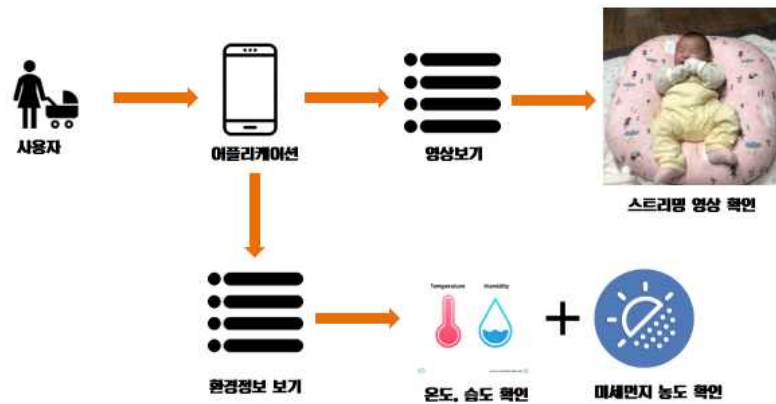
영상 스트리밍 서버 전송 지연 문제는 무선 LAN (와이파이) 환경에서 스트리밍 속도가 느려질 가능성이 존재한다. 전송 지연이 심할 경우, 유선으로 연결 하거나 라즈베리파이에서 응용 가능한 타 스트리밍 서비스(ex.G스트리머)로 대체하여 해결한다.

시리얼 데이터 전송 문제는 전송되는 시리얼 데이터 값을 특정 문자로 구분하는 것으로 해결한다. 센서별 데이터는 괄호로, 센서 데이터 내의 세부 구분은 쉼표로, 종결은 빗금으로 구분함으로써 송신되는 디바이스에서 명확한 구분이 가능 하도록 한다.

작품의 외관, 즉 모델링의 완성도 또한 높아야 할 것이다. 작품의 보드, 센서, 점퍼선 등 적나라하게 드러나는 것들을, 3D 프린터와 Eva 폼을 사용함으로써 외관을 보완하여 완성도와 제품성을 향상시킨다. 또한 자칫 신생아에게 위험을 초래할 수 있는 무거운 SoC보드와 같은 것들은 모빌의 하단에 배치함으로써 안정성을 향상시킨다.

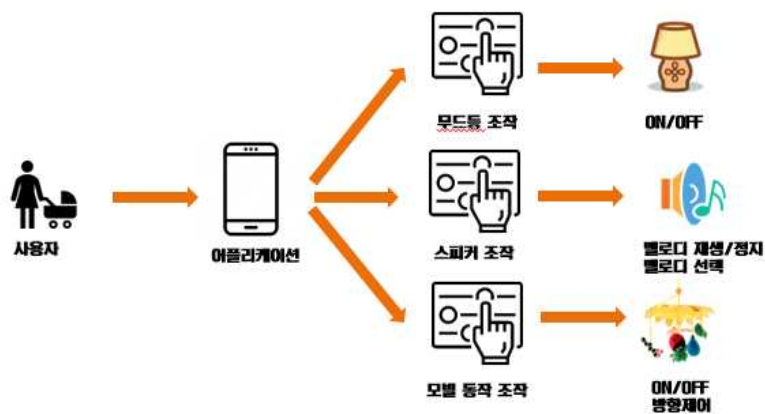
### 3. 시험 시나리오

[그림 5]는 영상 스트리밍과 온·습도, 미세먼지 농도 확인 시나리오이다. 사용자(부모)는 실시간으로 신생아의 상태를 확인하기 위해 애플리케이션을 이용하여 영상 보기 모드를 선택한다. 사용자는 라즈베리 파이의 카메라 모듈, UV4L서버를 통해 실시간으로 신생아의 모습을 모니터링 할 수 있다. 온·습도, 미세먼지 농도 확인을 위해 환경 정보 보기 모드를 선택하면 실시간으로 모빌에 장착된 아두이노 모듈로부터 주변의 온·습도, 미세먼지 농도 데이터와 적정 권장 데이터를 제공받을 수 있다.



[ 그림 5 : 시스템 수행 시나리오, 영상 스트리밍 & 환경정보 ]

[그림 6]은 LED, 스피커, 모터 모듈 동작 제어 시나리오이다. 사용자는 애플리케이션을 통해 무드등 조작 메뉴를 선택하여 신생아가 수면에 든 경우나 수유 중인 상황에 맞춰 LED 모듈의 색과 밝기를 조절할 수 있다. 또한, 신생아의 충분한 흥미 유도를 위하여 음악 재생 메뉴를 통해 상황에 맞는 멜로디를 아이에게 제공할 수 있다. 사용자는 모빌 동작 제어 메뉴를 통해 원거리에서도 모빌의 회전 방향, 속도 등을 제어하여 아이가 우는 등의 유사 상황 시에 유연한 대처가 가능하다.



[ 그림 6 : 시스템 수행 시나리오, 모듈 동작 제어 ]

[그림 7]은 비접촉식 온도 센서와 육아 정보에 대한 시나리오이다. 사용자는 애플리케이션의 온도 보기 메뉴를 통해 모바일에 부착된 비접촉식 온도계를 이용하여 신생아의 온도를 실시간으로 검사할 수 있다. 또한, 애플리케이션 내에서 자체적으로 제공하는 유용한 신생아 육아 정보를 ‘육아 정보 보기’ 메뉴를 통해 접근할 수 있다.



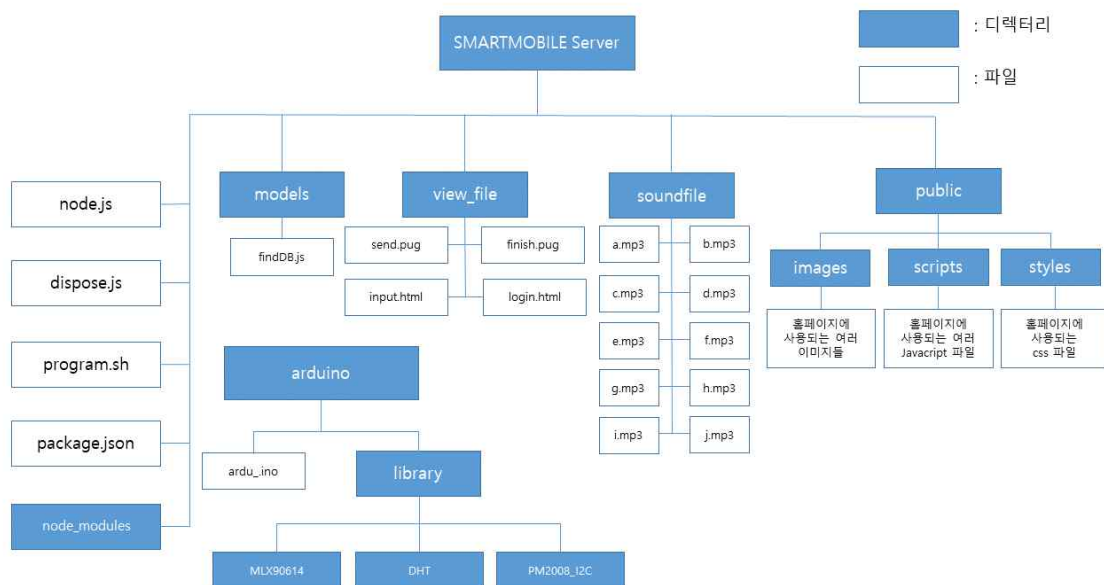
[ 그림 7 : 시스템 수행 시나리오, 비접촉식 온도센서&육아정보 ]

종합하면, 생후 12개월 이내의 아기를 둔 부모(사용자)가 본 제품을 사용한다. 사용자는 애플리케이션을 통해 모바일에서 수집되는 모든 데이터를 얻을 수 있고, 부착된 기기를 제어할 수 있다. 모바일에 부착된 카메라 모듈을 통해 실시간으로 스트리밍되는 동영상을 시청할 수 있고, 온습도 센서와 미세먼지 센서에서 받아온 데이터를 통해 애플리케이션에서 온습도, 미세먼지 수치와 통계 그래프를 확인한다. 또한, 비접촉식 온도 센서를 사용하여 아기의 체온을 검사할 수 있고, 추가로, 육아 정보를 확인할 수 있으며 필요시에 무드등과 스피커, 모바일의 동작을 애플리케이션으로 원격 제어할 수 있다.

#### 4. 상세 설계

[그림 8]은 스마트 모빌 서버와 아두이노의 모듈 상세 설계 표이다. 라즈베리파이와 아두이노로 구성된다.

아두이노에서는 시리얼 포트를 통해 데이터를 전송하는 기능을 하는 ardu.ino 파일과 센서별 라이브러리로 나누어진다. 라즈베리 파이에에는 node.js 서버 실행에 필요한 모듈들을 포함하는 node\_modules, 웹 통신을 담당하는 node.js 파일, 장치의 동작을 담당하는 dispose.js 파일이 있다. models 디렉터리에는 데이터베이스를 사용을 위한 findDB.js가 존재하고, view\_file에는 node.js 웹서버에서 사용하는 화면을 구성하는 html, pug 파일이 배치되어 있다. soundfile에는 스피커에서 지원할 다양한 동요 파일이 mp3 형태로 저장되어 있다. public에는 images, scripts, styles 디렉터리에 홈페이지에서 사용되는 이미지, css, javascript 파일들이 배치되어 있다.

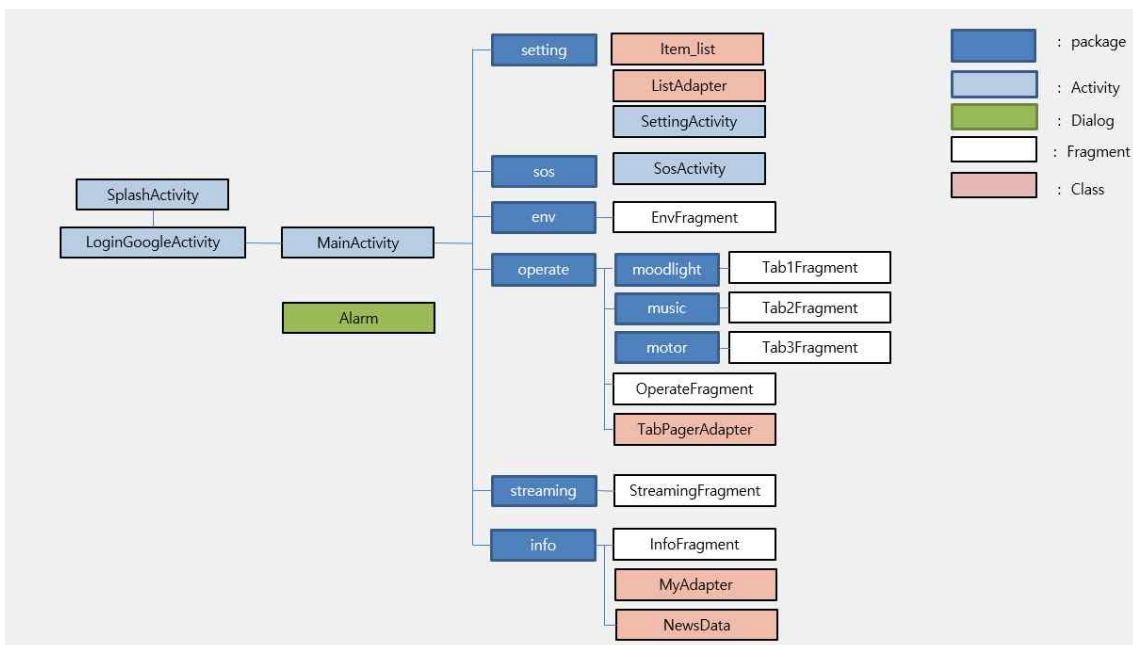


[ 그림 8 : 아두이노, 서버 구성도 ]

[그림 9]는 본 프로젝트 애플리케이션의 모듈 상세 설계 구성도이다. 애플리케이션은 MainActivity에서 뷰를 관리한다. 현재 기능 단위로 나누어진 5개의 패키지를 통해 세부 동작을 관리한다.

login은 로그인을 담당하는 패키지로써, 애플리케이션 실행시 권한을 담당하는 FirstAuthActivity, 로그인을 담당하는 LoginActivity, 사용자 가입을 담당하는 RegisterActivity, 로그인 정보를 기억하는 SaveSharedPreferences로 구성된다. env는 환경정보를 담당하는 패키지이다. 해당 프래그먼트인 EnvFragment는 주변 미세먼지, 온도, 습도, 당월 데이터의 차트를 제공한다. operate는 모빌 주변장치와 모빌 동작을 제어를 담당하는 패키지로써, OperateFragment와 TabPagerAdapter는 탭레이아웃을 정의하고, Tab1Fragment, Tab2Fragment, Tab3Fragment 각각은 무드등(LED), 음악(스피커), 모빌 동작(모터) 제어를 담당한다. streaming 패키지는 StreamingFragment를 가지고 있으며 아이의 실시간 동영상 스트리밍을 담당한다. 마지막으로 info 패키지는 InfoFragment를 가지고 있으며 육아정보를 제공한다.

MainActivity의 DrawbleLayout을 통해 '환경정보', '무드등/음악/모빌동작', '동영상', '육아정보' 각 항목을 기능별로 접근할 수 있다. 오른쪽 위의 option을 통해 앱 설정 기능을 담당하는 SettingActivity로 화면전환을 할 수 있다.



[ 그림 9 : 애플리케이션 구성도 ]

[표 5]는 아두이노 모듈 상세 설계 내용이다. 아두이노 보드에 사용되는 센서는 총 4가지로, 이는 미세먼지 농도 측정 센서, 온습도 측정 센서, 비접촉식 온도센서, 소리감지 센서로 이루어진다.

미세먼지 농도 측정 센서에서 사용되는 함수 CheckDust()는 미세먼지 농도 측정 값을 String형 문자열 변수 \_dust로 반환한다. 온습도 측정 센서에서 사용되는 함수 CheckTempHumidity()는 온습도 측정 데이터를 String형 문자열 변수 \_temphum으로 반환한다. 비접촉식 온도 센서에서 사용되는 함수 CheckBodyTemp()는 신생아로부터 측정된 체온 데이터를 String형 문자열 변수 \_bodytemp로 반환한다. 소리감지 센서에서 사용되는 함수 CheckSound()는 센서로부터 측정된 주변 소음 데시벨 값을 int형 변수 \_sound로 리턴한다.

[ 표 5 : 아두이노 모듈 상세 설계 ]

아두이노 모듈 상세 설계		
동작센서	함수명	리턴값
미세먼지 농도 측정 센서	string CheckDust()	String형 문자열 _dust
온습도 측정 센서	string CheckTempHumidity()	String형 문자열 _temphum
비접촉식 온도센서	string CheckBodyTemp()	String형 문자열 _bodytemp
소리 감지센서	int CheckSound()	int형 _sound

[표 6]은 라즈베리파이 모듈의 상세 설계이다. 라즈베리파이에서 사용되는 센서는 총 3가지로, 이는 LED와 DC모터, 스피커로 구성된다.

LedOn(int rgb, int time)은 매개변수로 불빛의 색상을 결정할 RGB값과 점등을 유지할 시간을 입력받는다. LedOff()는 LED를 불빛을 끄는 역할을 수행한다. DC모터의 SpinMotor(int direction, int speed, int time) 메소드는 매개변수로 모터의 회전 방향과 속도, 동작 시간을 입력받는다. StopMotor() 메소드는 모터의 회전 동작을 정지하는 역할을 수행한다. 스피커의 MusicOn(int musicnum)과 SoundOn(music num)은 각각 음악과 알람을 재생하는 역할을 하며 매개변수로는 음악의 재생목록 번호를 입력받는다. MusicOff()와 SoundOff()메소드는 스피커로 동작을 정지하는 기능을 수행한다.



[ 표 6 : 라즈베리파이 모듈 상세 설계 ]

라즈베리파이 모듈 상세 설계	
동작센서	함수명
LED	fuction LedOn(int rgb, int time) function LedOff()
DC모터	fuction SpinMotor(int direction, int speed, int time) fuction StopMotor()
스피커	fuction MusicOn(int musicnum) fuction MusicOff() void SoundOn(int musicnum) void SoundOff()

[표 7]은 애플리케이션 주요 함수 설계이다. getDust(String dustString)메서드는 아두이노로부터 전달된 미세먼지 농도 측정 값을 getDust(String dustString)을 통해 String형 문자열 dust로 반환하는 역할을 한다. getTempHum(String THString) 메서드는 아두이노로부터 전달된 온습도 측정 값을 getTempHum(String THString)을 통해 String형 문자열 temphum으로 반환하는 역할을 한다.

마찬가지로 getTempBody(String bodyTempString)메서드는 비접촉식 온도센서 측정 값을 getTempBody(String bodyTempString)을 통해 String형 문자열 tempbody으로 반환하는 역할을 한다. getInfo()메서드는 애플리케이션에서 활용될 주요 함수이다. 아두이노와 라즈베리파이로부터 데이터를 수신받은 웹사이트의 데이터 내용을 getInfo()메서드를 통해 파싱한다. getInfo()함수를 통해 파싱해온 뒤, 상기의 함수들을 호출하여 문자열을 반환한다.

[ 표 7 : 애플리케이션 함수 상세설계 ]

애플리케이션 함수 상세설계		
동작 기준	함수명	리턴값
미세먼지 농도 측정 센서	string getDust(String dustString)	string형 문자열 dust
온습도 측정 센서	string getTempHum(String THString)	string형 문자열 temphum
비접촉식 온도센서	string getTempBody(String bodyTempString)	string형 문자열 tempbody
웹 파싱	void getInfo()	-

[표 8]은 데이터베이스에서 활용될 테이블의 상세 설계이다. 테이블은 총 6개로 구성된다. 각각의 테이블은 애플리케이션으로부터 전달받은 연관된 센서 데이터를 저장·관리한다. THValue 테이블은 온습도 측정 데이터와 연관된 테이블이다. 튜플은 날짜, 시간, 현재 온도, 현재 습도로 이루어진다.

DustValue 테이블은 미세먼지 농도 측정 데이터와 연관된 테이블이다. DustValue 테이블의 튜플은 날짜, 시간, 현재 미세먼지 농도로 이루어진다. NonTouchValue 테이블은 비접촉식 온도센서로부터 측정되는 데이터와 연관된 테이블이다. NonTouchValue 테이블의 튜플은 날짜, 시간, 신생아 체온 값으로 이루어진다. SoundCheck 테이블은 소리감지 센서로부터 측정된 데시벨 데이터와 연관된 테이블이다. 해당 테이블의 튜플은 측정순서, 날짜, 시간, 측정 소음 데시벨로 이루어진다. PlayMusic 테이블은 모빌에서 재생되는 음악의 정보와 연관된 테이블이다. PlayMusic 테이블의 튜플은 재생목록 내 해당 음악의 번호와 해당하는 음악의 제목으로 이루어진다. BabyInfo 테이블은 애플리케이션에서 확인하는 육아 정보와 연관된 테이블이다. 해당 테이블의 튜플은 정보 카테고리, 정보 목록 번호, 육아 정보로 이루어진다.

[ 표 8 : 데이터베이스 테이블 상세 설계 ]

Database 모듈 상세 설계	
테이블 명	컬럼
THValue	date(날짜), time(시간), temp(현재온도), hum(현재습도)
DustValue	date(날짜), time(시간), dust(현재 미세먼지 농도)
NontouchValue	date(날짜), time(시간), bodytemp(신생아 체온)
SoundCheck	index(측정순서), date(날짜), time(시간), sounddb(측정 소음 데시벨)
PlayMusic	musicnum(음악 재생목록 번호), musicname(음악 제목)
BabyInfo	category(정보 카테고리), infonum(정보 목록 번호), infocontent(육아정보)

[표 9]는 애플리케이션 내에서 연동 데이터베이스를 활용할 수 있는 함수들이다. 동일 센서에서 활용되는 메소드들은 하나의 클래스에서 작성된다.

온습도 측정 센서와 관련된 메소드는 startCheckTH 클래스에서 작성된다. void InsertTemp(int date, int time, int temp) 메소드는 날짜, 시간, 온도를 매개변수로 입력받아 데이터베이스 THValue 테이블에 값을 저장한다. String SearchPastTemp(int date)와 String SearchSeasonTemp(int season)은 각각 과거의 온도 내역 및 특정 기간의 온도를 출력한다.

미세먼지 농도 측정 센서와 관련된 메소드는 startCheckDust 클래스에서 작성된다. void InsertDust(int date, int time, int dust) 메소드는 날짜, 시간, 미세먼지 농도를 매개변수로 입력받아 데이터베이스 DustValue 테이블에 값을 저장한다. String SearchPastDust(int date)와 String SearchSeasonDust(int season)은 각각 과거의 미세먼지 농도 측정 내역 및 특정 기간의 미세먼지 농도를 출력한다.

비접촉식 온도 센서와 관련된 메소드는 startCheckBT 클래스에서 작성된다. void InsertBodyTemp(int date, int time, int body\_temp, int body\_state) 메소드는 날짜, 시간, 체온, 체온 상태를 매개변수로 입력받아 데이터베이스 NontouchValue 테이블에 값을 저장한다. String SearchPastBodyTemp(int date)와 String SearchSeasonBodyTemp(int season)은 각각 과거의 온도 내역 및 특정 기간의 온도를 출력한다. 또한 체온값을 매개변수로 하는 String StateBodyTemp(it body\_temp) 메소드를 통해 체온 이상 지수를 출력하고 이를 그래프에서 활용할 수 있도록 한다.

소리감지 센서와 관련된 메소드는 startChecksound 클래스에서 작성된다. int InsertSound(int date, int time, int sound) 메소드는 날짜, 시간, 측정된 데시벨을 매개변수로 입력받아 데이터베이스 SoundCheck 테이블에 값을 저장한다. Void AlarmCheck(int last\_sound\_index) 메서드를 통해 리턴된 값으로 알람 작동 여부를 체크하도록 한다. 해당 메서드는 int alarm\_state를 리턴한다.

스피커와 관련된 메소드는 startCheckSpeaker 클래스에서 작성된다. string MusicInfo(int musicnum) 메소드는 재생목록 번호를 매개변수로 입력받아 데이터베이스 PlayMusic 테이블에 값을 저장한다.

육아정보와 관련된 메소드는 startCheckBabyInfo 클래스에서 작성된다. string BabyInfo()는 카테고리 및 콘텐츠, 특정 번호를 매개변수로 입력받아 데이터베이스 BabyInfo 테이블에 값을 저장한다.

[ 표 9 : 데이터베이스 관련 애플리케이션 함수 상세설계 ]

애플리케이션 함수		
센서 명	클래스명	형식
온습도 측정 센서	startCheckTH	void InsertTemp(int date, int time, int temp) String SearchPastTemp(int date) String SearchsSeasonTemp(int season)
미세먼지 농도 측정 센서	startCheckDust	void InsertDust(int date, int time, int dust) String SearchPastDust(int date) String SearchSeasonDust(int season)
비접촉식 온도 센서	startCheckBT	void InsertBodyTemp (int date, int time, int body_temp, int body_state) String SearchPastBodyTemp(int date) String SearchSeasonBodyTemp(int season) String StateBodyTemp(int body_temp)
소리 감지 센서	startCheckSound	int InsertSound(int date, int time, int sound) ←last_sound_index 리턴 void AlarmCheck(int last_sound_index) ←리턴된 값으로 알람작동여부 검사 ←int alarm_state 리턴
스피커	startCheckSpeaker	string MusicInfo(int musicnum)
육아정보	startCheckBabyInfo	string BabyInfo(int category, int content) string BabyInfo(int category, int infonum, int content)

[표 10]은 웹 서버 상세 설계이다. main.js의 메소드는 다음과 같다. app.listen(3000,())는 node.js 서버를 실행하는 메소드 이다. Express 모듈의 함수이고, 첫 번째 매개변수에 3000은 포트 번호를 의미한다. app.get("/", (req, res)) 메소드는 URL의 /로 접근했을 때의 실행되어지는 함수이다. 이때 /hello 메소드로 리다이렉트 한다. serialPort.open(function ())은 시리얼 포트를 통해 들어오는 데이터를 입력받기 위한 open 함수이다.

serialPort.on('data',function(data)는 해당 시점부터 시리얼 포트의 데이터를 data로 읽어오는 것을 시작하는 메소드이다. 해당 데이터들을 읽어와 정제해서 웹에 배포하는 형태로 구현하였다. app.post("/data", function(req, res))는 포스트

값으로 입력되는 데이터를 처리하는 함수이다. 앱, 웹으로부터 포트 값을 입력받고 동작을 수행하기 위해 해당하는 메소드를 호출한다.

dispose.js의 메소드들이다. function selectdevice(device)는 포트 데이터에 해당하는 디바이스를 선택하는 메소드이다. function ledpower(onoff)는 led의 on, off동작을 제어하는 메소드이고, function ledcolor(color) led의 색깔을 결정한다. function speakerpower(onoff) 메소드는 스피커의 on, off동작을 제어하는 메소드이고, function speakermusic(music)는 재생할 음악을 결정하는 메소드이다. function moterpower(onoff)은 서보모터의 on,off동작을 제어하는 메소드이고, function moterdirection(direction)는 모터의 방향, function moterspeed(speed) 모터의 속력을 결정하는 메소드이다.

[ 표 10 : 웹 서버 상세설계 ]

파일명	함수이름
main.js	<pre> app.listen(3000, () ) app.get("/", (req, res)) serialPort.open(function () ) serialPort.on('data',function(data) app.post("/data", function(req, res)) </pre>
dispose.js	function selectdevice(device)
	function ledpower(onoff) function ledcolor(color)
	function speakerpower(onoff) function speakermusic(music)
	function moterpower(onoff) function moterdirection(direction) function moterspeed(speed)

## 5. Prototype 구현

[표 11]은 프로토타입 구현 현황에 대해 요약된 표이다. 전체적인 진행 상황은 68.5%이며, 아두이노는 73%, 라즈베리파이 63%, 웹 서버 100%, 애플리케이션 핵심기능 85%, 애플리케이션 추가기능 80%, 프레임 10%이다. 대부분 원활히 진행되었으며 진행되지 못한 항목에 대한 원인으로는 모듈의 추가 구매, 프레임 설계 등이 있다.

[ 표 11 : 프로토타입 구현 현황 ]

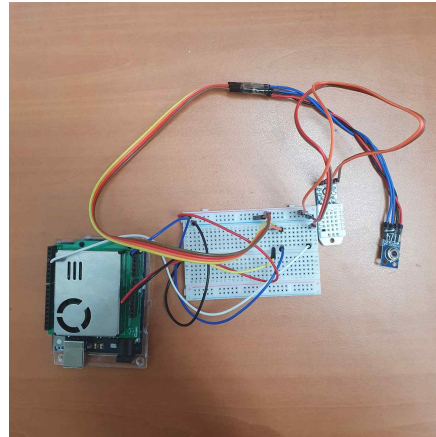
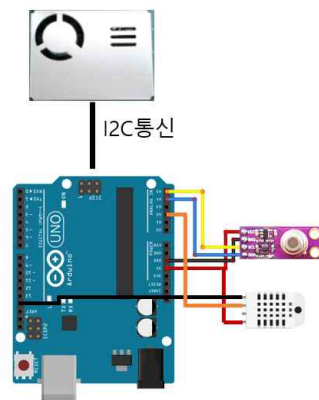
플랫폼		진도
아두이노		73%
라즈베리 파이		63%
웹 서버		100%
애플리케이션	핵심기능	85%
애플리케이션	추가기능	80%
프레임 제작		10%
전체진행률		68.5%

[표 12]는 아두이노 보드와 모듈 구현 현황에 대한 표이다. 핵심 기능인 온습도, 미세먼지 정보 수신의 경우, 센서 모듈과 보드 정상적으로 연결되어 데이터 수신을 명확히 할 수 있었다. 비접촉식 온도 센서의 경우 모듈 특성상 납땜이 필요하나, 현재 납땜을 하지 못하여 간혹 비정상적 데이터가 수집되는 경우가 발생한다. 이 문제는 차후 납땜을 통해 해결하거나, 모듈 자체적인 문제라고 판단되면 비접촉식 온도센서 모듈을 교체할 예정이다. 부가기능으로 설계하였던 소리감지 센서 모듈은 구매를 완료한 상황이다.

[ 표 12 : 아두이노 모듈 구현 현황 ]

플랫폼	기능분류	세부모듈	진도		문제점	해결방안
아두이노	핵심기능	온습도센서	100%	73%	연결 및 데이터 전송 완료	없음
		미세먼지센서	100%		연결 및 데이터 전송 완료	없음
		비접촉식 온도센서	90%		접촉불량으로 인한 데이터전송 오류 문제	납땜 예정
	부가기능	소리감지 센서	0%		미구현	소리 감지 센서 구매 완료

[그림 10]은 아두이노 회로도 와 실물 사진이다. 센서 모듈 테스트 단계에 있으며 현재 회로도를 기반으로 프레임 제작, 모델링 단계에서 보완할 예정이다.



[ 그림 10 : 아두이노 회로도와 실물 ]

[그림 11]은 아두이노 모듈 테스트 사진이다. 센서 모듈들이 수집한 환경정보 데이터를 Node.js서버로 전송한 모습이다. 1번째 행에 전송된 데이터는 미세먼지 데이터로, PM10(미세먼지) 농도이며, PM2.5(초미세먼지)농도이다. 2번째 행에 전송된 데이터는 온습도 데이터이다. 3번째 행에 전송된 데이터는 비접촉식 온도센서의 값이다. 테스트 결과, 아두이노 보드에 연결된 센서 모듈로부터 수집된 테스트 환경 정보 데이터는 미세먼지 농도  $23\mu\text{g}/\text{m}$ , 초미세먼지 농도  $21\mu\text{g}/\text{m}$ , 온도  $24.4^{\circ}\text{C}$ , 습도 59.8%, 비접촉식 온도센서로 측정된 값은  $26.31^{\circ}\text{C}$ 였다.



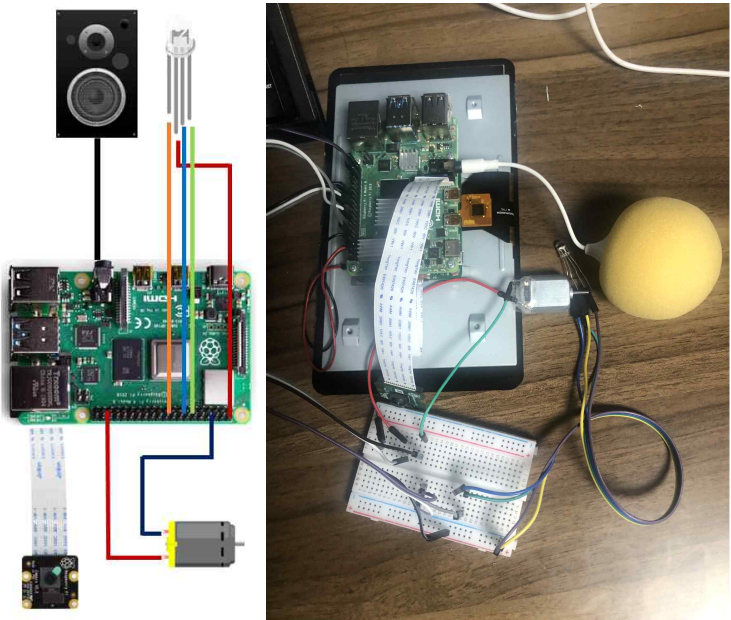
[ 그림 11 : 아두이노 모듈 테스트 ]

[표 13]은 라즈베리파이 보드와 모듈 구현 내용이다. UV4L 서버의 애로사항으로는 와이파이 연결 시 지속적으로 와이파이가 끊기거나, 프레임 저하 현상이 있었다. 또한 UV4L 서버 특성상 외부 IP에서 해당 IP로의 접근이 불가능한 어려움이 있었다. 현재 와이파이 문제는 유선 랜이나 핸드폰 핫스팟 및 USB 테더링을 이용하여 해결하였고, 프레임 저하 문제는 기존 사용하던 라즈베리 파이 3B+ 모델에서 새로 출시된 4B 모델로 보드 버전 업그레이드를 하여 해결하였다. 외부 IP 접근 불가의 경우 포트포워딩을 이용하여 해결하였다.

[ 표 13 : 라즈베리파이 모듈 구현 현황 ]

플랫폼	기능분류	세부모듈	진도		문제점	해결방안
라즈베리 파이	핵심기능	DC모터	0%	63%	미구현	모터 드라이버 구매 완료
		스피커	100%		연결 및 데이터 전송 완료	없음
		UV4L서버	100%		프레임 드랍 문제	라즈베리 파이 버전 업그레이드
	부가기능	LED 무드등	50%		일부 구현	줄 LED 교체 채구매 완료

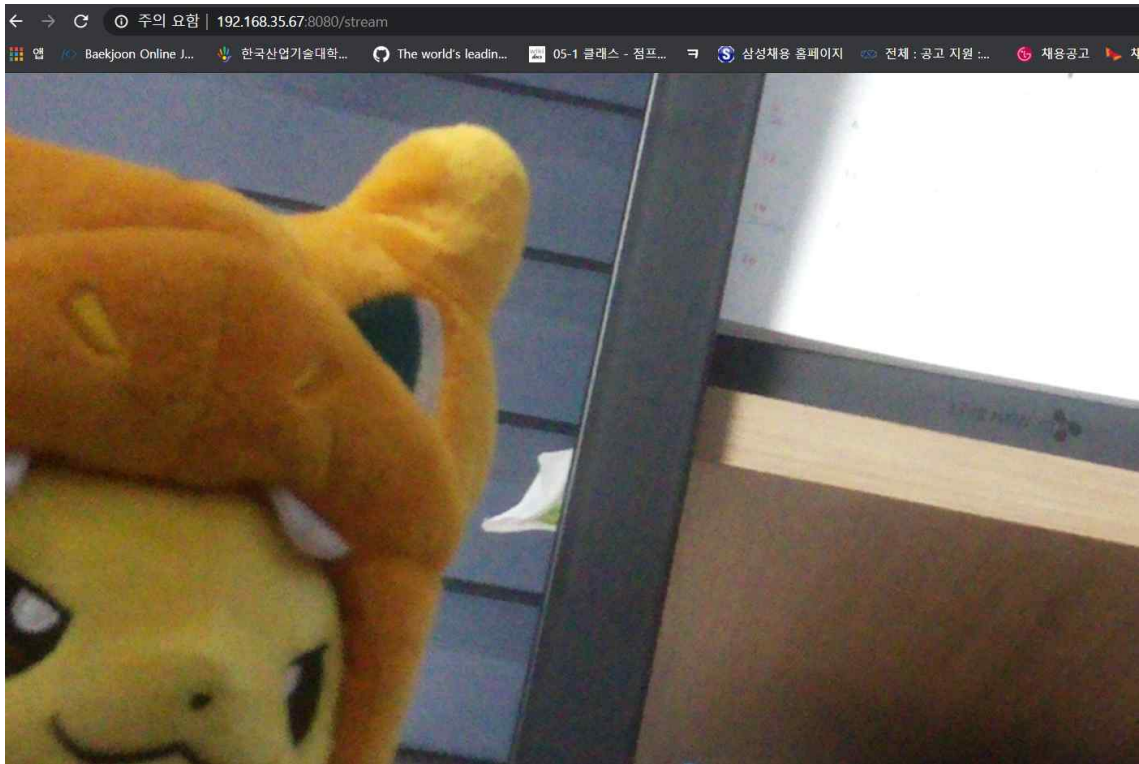
[그림 12]은 라즈베리파이 회로도와 실물 사진이다. 아두이노와 마찬가지로 센서 모듈 테스트 단계에 있으며, 회로도를 기반으로 프레임 제작, 모델링 단계에서 보완할 예정이다. 라즈베리파이에서는 아두이노와의 시리얼통신, 스피커, led, dc모터, pi 카메라를 제어한다.



[ 그림 12 : 라즈베리파이 회로도와 실물 ]



[그림 13]은 라즈베리파이 UV4L 서버 테스트 모습이다. 라즈베리 파이에 부착된 카메라 모듈을 통해 UV4L 전용 서버로 영상 데이터를 전송한다. 해당 서버에서 전송되고 있는 영상을 애플리케이션을 통해 사용자에게 전달한다.



[ 그림 13 : 라즈베리파이 UV4L 서버 ]

[그림 14]는 데이터 베이스 환경정보 데이터의 속성값이다. 라즈베리파이에서 환경정보 데이터를 서버에 전달함과 동시에 1분단위로 데이터베이스에도 전달한다. 애플리케이션에서는 환경데이터를 실시간으로 전달받으며, 전달받는 시점에서 서버에 접속한다. 그래프나 이전 기록을 확인할 때에는 데이터베이스 환경 정보에 접근하게 된다. 데이터베이스 개발에는 Mariadb를 사용하였다.

	Field	Type	Null	Key	Default	Extra
▶	env_no	int(11)	NO	PRI	NULL	auto_increment
	time	varchar(50)	NO		NULL	
	tempandhum	varchar(30)	NO		NULL	
	temperature	varchar(30)	NO		NULL	
	dust	varchar(30)	NO		NULL	

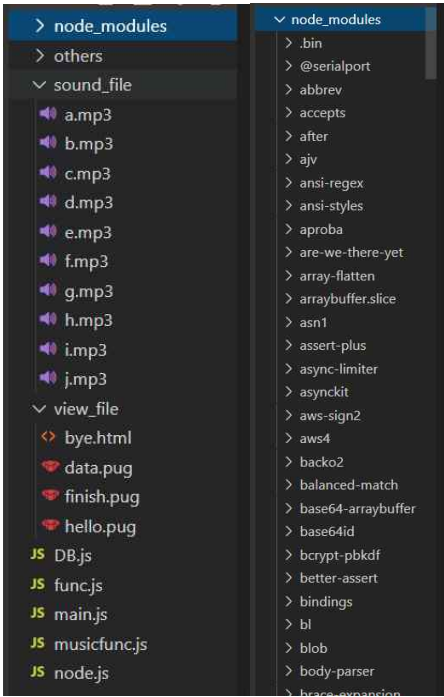
[ 그림 14 : 환경정보 테이블의 속성값 ]

[표 14]는 Node.js 웹 서버 구현 내용에 대한 설명이다. 통신의 용도로 사용하기 위해서 개발하였다. 현재 환경데이터들을 전달하는 부분과 애플리케이션에서 모빌의 동작을 제어하기 위한 정보를 실시간으로 주고받는데 사용한다. 서버 배포 시에 다른 npm 라이브러리들과 버전 호환이 맞지 않는 문제가 발생하였으나 실행이 되지 않는 오류를 node 버전을 낮추면서 해결하였다.

[ 표 14 : 웹 서버 구현 현황 ]

플랫폼	기능분류	세부모듈	진도		문제점	해결방안
웹 서버	핵심기능	백엔드	100%	100%	데이터 전송 서버 구현 데이터 출력	node버전 변경

[그림 15]는 Node.js 웹 서버 개발 조직도이다. 시리얼통신, 서버배포는 main.js에서 이루어지며 데이터베이스, 장치제어와 관련된 작업들은 외부의 함수를 호출하는 형태이다. 데이터베이스와 관련된 작업은 DB.js에서 이루어진다. 스피커, led, 모터와 같은 장치들의 제어는 func.js안에서 함수단위로 구현하였다. 음악파일은 sound\_file 내부에 보관한다. 웹서버의 뷰는 view\_file에 보관이 되며 각 기능에 따라 html과 pug형식 2가지를 사용한다.



[ 그림 15 : 웹 서버 개발 조직도 ]

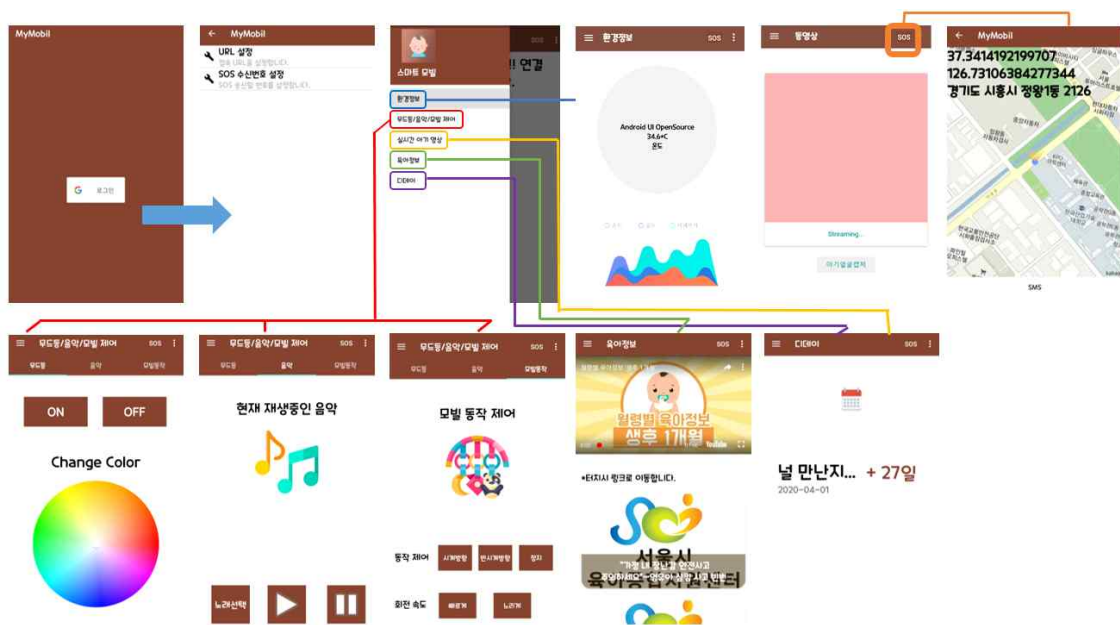
[표 15]는 애플리케이션 구현 내용에 대한 설명이다. 애플리케이션의 경우 핵심 기능은 모터, 무드등 기능을 제외한 모든 기능을 구현 완료한 상황이며, 해당 두 기능들의 레이아웃 프론트엔드 작업은 완료되었다. 차후에, 구매한 모터, LED 모듈을 통해 구현 할 예정이다. 앱 디자인은 세부 디자인을 제외한 전체적인 UI/UX 구현은 완료하였다.

추가 기능은 총 5가지의 기능을 구현하였으며, ‘구글 auth 활용 로그인’, ‘SOS’, ‘D-day’, ‘아기 얼굴 캡처’, ‘동화책 읽어주기’ 기능이다. 현재 동화책 읽어주기 기능을 제외한 4가지의 추가 기능의 오류들은 모두 해결하여 구현 완료하였고, ‘동화책 읽어주기’ 기능은 차후 개발 예정이다.

[ 표 15 : 애플리케이션 구현 현황 ]

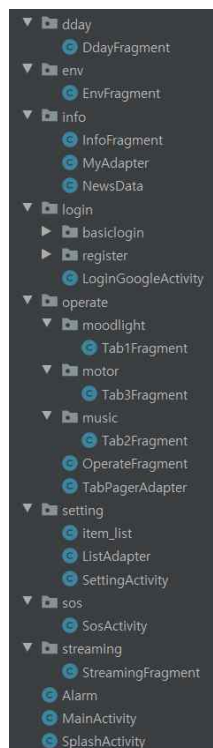
플랫폼	기능분류	세부모듈	진도		문제점	해결방안
애플리케이션	핵심기능	백엔드	80%	85%	모터/무드등 제어 미구현	모터 드라이버/LED 모듈 구매 후 재구현
		프론트엔드	90%		세부 디자인 미구현	추가 기능 구현내용에 따른 레이아웃 디자인 추가
	추가기능	구글 계정 로그인	100%	80%	API Exception 발생	인증서 코드 일치를 통해 해결
		SOS 기능	100%		API 오류 발생	키 값을 일치시켜 해결
		D-day 기능	100%		데이터 입출력 완료	없음
		아기 얼굴 캡처 기능	100%		데이터 출력 완료	없음
		동화책 읽어주기 기능	0%		미구현	차후 개발 예정

[그림 16]은 애플리케이션의 프로토타입이다. 초기 SplashActivity와 GoogleLoginActivity를 시작으로 애플리케이션을 실행하고, 좌측의 drawer navigation 메뉴를 통해 세부 기능으로 접근할 수 있다. 우측의 'SOS' 옵션과 'setting' 옵션이 존재하며, 'setting' 옵션을 통해 애플리케이션의 URL, sos 수신번호 설정, sos 문구 설정을 할 수 있다.



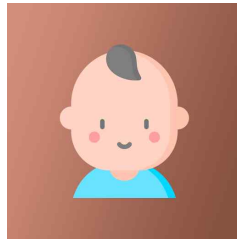
[ 그림 16 : 애플리케이션 프로토타입 ]

[그림 17]은 애플리케이션 개발 구성도이다. 안드로이드 애플리케이션은 크게 login, sos, setting, env, operate, info, dday 패키지와 MainActivity로 이루어져 있다. 세부 java 파일 구성은 아래 그림과 같다.



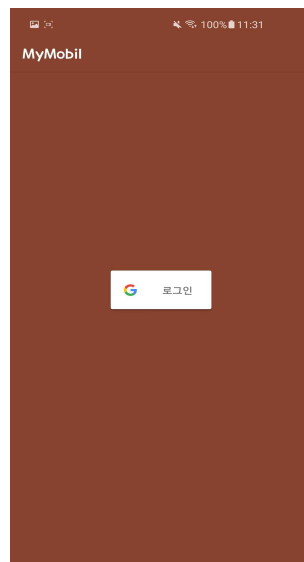
[ 그림 17 : 애플리케이션 개발 구성도 ]

[그림 18]~ [그림 26]는 애플리케이션 구현 결과에 대한 설명이다. [그림 18]은 스마트 모빌 애플리케이션의 실행 아이콘이다. 정사각형 형태의 512px크기를 가진다. 애플리케이션의 디자인은 전체적으로 안정적이고 편안한 느낌을 줄 수 있도록 브라운 계열의 색조합을 기반으로 하였다. Android 8.0 (API 26)부터 적용되는 적응형 런처 아이콘을 통해 상이한 기기에서도 아이콘을 마스킹하여 렌더링할 수 있도록 한다.



[ 그림 18 : 애플리케이션 아이콘 ]

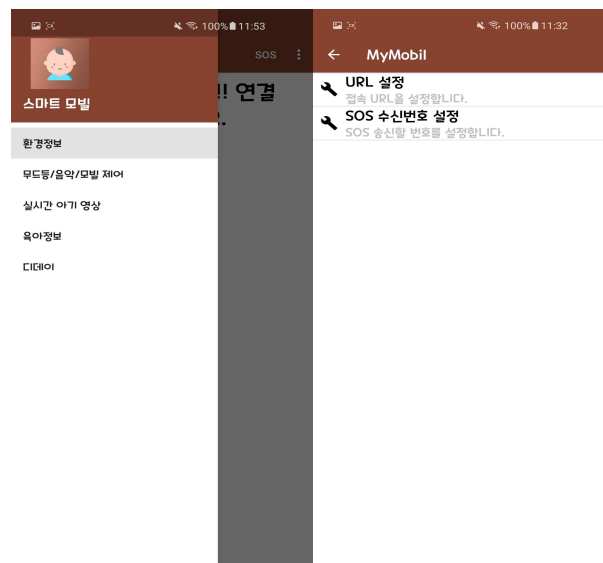
[그림 19]는 스마트 모빌 애플리케이션의 첫 실행 화면이다. 앱 실행 시, 먼저 Splash Activity와 LoginGoogleActivity를 실행하여 사용자 권한 획득과 구글 auth 획득을 수행한다. firebase Authentication 기능을 사용하여 구글 인증으로 로그인을 대체하였다. 사용자 인증 완료 시, UpdateUI() 함수에서 account를 확인 후 MainActivity로 intent 전환한다. 본 소스 코드에서 에러가 났던 'ApiException'은 구글과 firebase, android studio의 고유 SHA-1 인증서 코드를 일치 시켜서 해결하였다.



[ 그림 19 : 로그인 액티비티(Application) ]

[그림 20]은 로그인 이후 처음으로 출력되는 환경정보 MainActivity 화면이다. MainActivity에서는 Drawer Layout과 Navigation View, 우측 상단의 옵션 메뉴를 초기화 한다. Drawer Navigation의 메뉴는 총 5개로, ‘환경정보’, ‘무드등/음악/모빌 제어’, ‘실시간 야기 영상’, ‘육아정보’, ‘디데이’로 이루어져 있다. 모두 공통적으로 진입 초기에 서버 URL을 검사한다.

이는 SettingActivity에서 dialog를 통해 SharedPreferences 내부 저장소에 설정할 수 있으며, 설정한 URL이, URLUtil 라이브러리의 isValidUrl 함수를 통해 유효한 URL임이 검증이 되어야 코드가 실행되며, 유효하지 않은 URL일 시 SettingActivity로 intent 전환된다.



[ 그림 20 : 메뉴, 세팅(Application) ]

[그림 21]은 환경 정보를 제공하는 Fragment이다. 먼저 ‘환경정보’ 기능을 제공하는 EnvFragment로, 현재 보고 출력중인 Frame이 교체된다. while 무한 루프 안에서 1초의 쓰레드 sleep으로 1초 간격으로 서버에서 파싱을 해오는 getInfo() 함수를 호출 한다. getInfo() 함수에서는 서버의 URL을 Jsoup 라이브러리를 통해 파싱해 오고, 이를 select 메소드를 사용하여 태그 별로 나누어진 값들을 가져온다. 그 후 미세먼지, 온도, 체온 데이터를 각각 getDust(), getTempHum(), getTempBody() 함수를 호출하여 데이터를 가공하여 UI를 업데이트한다. (연결 실패 시 ‘connection fail’ 메시지 출력)

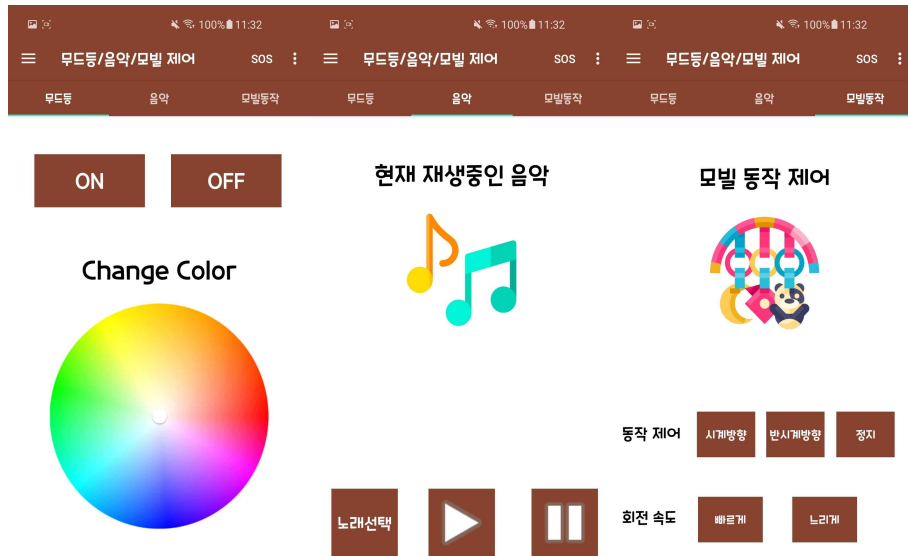


[ 그림 21 : 환경정보 메뉴(Application) ]

[그림 22]는 무드등/음악/모빌제어 탭으로 구성된 액티비티이다. 2번째 Drawer Navigation 메뉴의 ‘무드등/음악/모빌 제어’ 는 총 3개의 Fragment로 View Pager 형식으로 구성되어 있다. 무드등 제어 역할의 Tab1Fragment, 음악 제어 역할의 Tab2Fragment, 모터 제어 역할의 Tab3Fragment이다. 이들은 OperateFragment와 TabPagerAdapter를 통해 초기화된다.

무드등 제어에서는 ‘ON’, ‘OFF’, ColorPicker를 통해 켜, 끄, 색상 선택 기능을 제어할 수 있다. postInfo() 함수에서 Jsoup 라이브러리로 ‘http://xxx.xxx.xx.x:3000’ 이러한 형식의 url에서 ‘/data’ 링크에 ‘data’와 보내고자하는 값(on, off, color값)을 POST 한다.

음악 제어와 모터 제어 역시 같은 방식으로 동작한다. 그러나 음악 선택이 10가지 종류로 매우 많기 때문에, 데이터바인딩을 사용하여 위젯을 연결해주었다. (FragmentOperateTab2Binding binding을 사용)



[ 그림 22 : 무드등, 음악, 모빌동작 뷰페이지(Application) ]

[그림 23]은 원격 동영상 스트리밍을 통해 아이의 얼굴을 확인하는 액티비티이다. 라즈베리 파이에 장착된 카메라 모듈에서 수집된 영상은 UV4L 서버를 통해 웹으로 전송되며 특정 웹에 전송된 영상을 파싱함으로써 애플리케이션 내에서 바로 확인이 가능하도록 구현하였다.

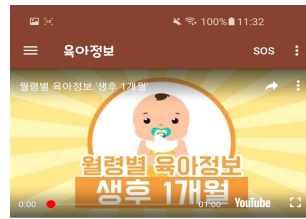
해당 기능은 StreamingFragment에서 동작한다. URL 포트에 8080을, 뒤에 '/stream'을 concat하여 웹뷰에 출력한다. 본 스트리밍 영상은 라즈베리파이 UV4L 서버에서 송출되며, 기존 프레임이 현저하게 끊기던 문제는 라즈베리파이 3B+에서 4B+로 변경하면서 해결하였다. 그리고, 얼굴 캡처 버튼 터치 시, 현재 스트리밍 화면을 ' /DCIM/Screenshots/폴더에 ' MyMobil\_Streaming.jpg' 파일로 저장한다.





[ 그림 23 : 동영상 스트리밍(Application) ]

[그림 24]는 신생아 육아에 필요한 육아정보를 제공하는 액티비티이다. 상단에는 유용한 육아 영상을 바로 시청할 수 있도록 제공한다. 메뉴의 ‘육아정보’는 info 패키지인 infoFragment, MyAdapter, NewsData에서 동작한다. 최상단에는 prnd YouTubePlayer API를 사용하여 육아정보 영상을 출력한다. 하단에는, 크게 Recycler View로 뉴스항목들을 Card View로 나타내었다. 항목당 이미지와 텍스트가 들어가는 데, 이미지는 facebook fresco를 사용하여 url 이미지를 출력하였고, 자연스러운 UI를 위해 Relative Layout을 사용하였다. 내부적으로, InfoFragment에서 seoul.childcare.go.kr(서울시 육아종합지원센터)에서 육아 관련 뉴스의 제목과 URL을 파싱해온다. 각 항목을 터치 시 해당 URL로 이동하는 기능을 제공한다.

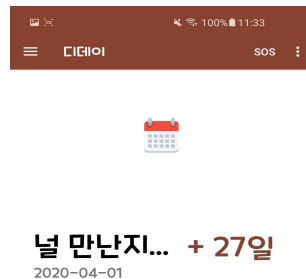


\*터치시 링크로 이동합니다.



[ 그림 24 : 육아정보(Application) ]

[그림 25]는 아이의 출산일을 기념하기 위해 만든 애플리케이션 내 부가 기능이다. 메뉴의 ‘디데이’ 항목 터치 시, DdayFragment에서 동작한다. 기존에 설정한 날짜가 내부저장소에 있는지 검사 후에, 있으면 ddayCalculate() 함수를 통해 날짜를 계산하여 해당 날짜로부터 얼마나 지났는지를 출력한다. 없으면 날짜를 선택하라는 문구를 출력한다. 캘린더 터치 시 날짜를 선택할 수 있다.



[ 그림 25 : 디데이(Application) ]

[그림 26]은 위급 상황 대처를 위해 만든 애플리케이션 내 부가 기능이다. 우측 상단의 옵션 메뉴의 SOS 터치 시, SosActivity로 전환한다. 해당 기능은 아이를 키우는 부모가 위급상황 발생 시 설정해둔 번호로 현재 위치를 전송하는 역할을 수행한다. 먼저 위치 권한과 sms 권한을 확인한다. 그 후 현재 위치를 지도상에 위도, 경도, 주소와 함께 표시한다. sms 버튼 터치 시 위치를 전송한다. 위치 기능은 Kakao Map API를 사용하였다. API 오류는 해시키와 네이티브앱 키를 Kakao developer 사이트의 키와 맞추어 해결하였다.



[ 그림 26 : SOS(Application) ]

[표 16]은 프레임 설계 내용에 대한 설명이다. 프레임은 기존의 시제품을 기반으로 하여 플라스틱 파이프, 우드락, 하드보드지, 시트지, 아이 인형 등을 이용하여 프레임을 설계 할 예정이다. 기존의 시제품 모빌은 3단 모빌이며, 스탠드형 모빌이기에 데모 발표에 용이할 것으로 생각된다. 프레임 제작에 필요한 시제품 모빌은 구매 완료한 상황이며, 제작에 필요한 제품들을 구입 후 모델링 할 예정이다.

[ 표 16 : 프레임 설계 현황 ]

플랫폼	기능분류	세부모듈	진도		문제점	해결방안
프레임 제작	핵심기능	-	10%	10%	프레임 제작 물품 미구매	프레임 제작에 필요한 물품 구매 예정

[그림 27]은 실제 모델링에 사용될 3단 스탠드형 모빌 시제품이다. 해당 시제품은 스탠드 부분이 2단 분리가 가능 하여 모듈을 배치하기에 용이하다. 모빌대 부분은 원형 파이프의 형태를 하고 있어 점퍼선을 삽입함으로써 깔끔한 모델링이 가능

하다.



[ 그림 27 : 시제품 실물 ]

프로토타입 구현 결과, 전체진행률이 68.5%정도이나 프레임 설계 및 데모 현장에서 직면할 수 있는 돌발적 문제가 예상되므로 이에 대한 철저한 준비가 필요하다. 세부사항으로는 아두이노에서는 ‘비접촉식 온도 센서’의 접촉 불량으로 인한 전송 데이터 오류 문제를 모듈 납땜을 통해 해결하고, ‘DC모터’를 통한 모빌 원격제어 기능을 구현하는 것이 중요하다. DC모터의 경우 구현에 필요한 모터 드라이버를 주문하였다. ‘LED 무드등’의 경우 모듈에 따라 구현 방식과 회로 연결 방식 및 색 표현의 차이가 존재하므로 이를 고려하여 충분히 제어 가능한 모듈을 선정하고 구현하는 것이 중요하다. 결론적으로, 주요 핵심 기능은 1가지(모빌동작 원격 제어)를 제외하고는 현재 구현 되었으며 추가기능은 2가지(소리 감지, 동화책 읽어주기)가 남은 상황이다. 모터를 제외한 하단 프레임 설계 부분을 진행하며 모터 구현 이후 상단 프레임을 설계, 모듈과 보드를 모델화 시켰을 때 애플리케이션과 동시에 원활히 작동되도록 프레임을 제작하는 과정이 중요할 것이다.

[표 17]은 3차발표 이후 일정 계획표이다. 개발단계에서 장치와 서버, 애플리케이션의 연동 이상 여부를 점검하며 개발하였기 때문에 5-6월 테스트 단계에서는 데모 시연 시 발생할 수 있는 wi-fi문제, 모듈 에러 문제 등을 점검하도록 하고 모델링과 프레임 설계에 집중한다. 이후 6월 중순-7월까지의 데모 시연 및 제품화를 가정하였을 때 발생할 수 있는 환경 제약 처리와 디버깅을 수행한다. 마지막으로 7월 중순-9월까지의 테스트 단계에서 설계되었던 프레임을 제품화하고 원활한 데모 수행을 위해 모듈 점검 및 테스트를 수행한다.

[ 표 17 : 3차발표 이후 일정 계획표 ]

	추진 사항	5	6	7	8	9
테스트단계	모델링 / 프레임 설계 장치, 서버, App 연동					
유지보수단계	환경 제약 처리, 디버깅					
최종모델링	최종 테스트 데모 시연 준비					

## 6. 시험/테스트 결과

[표 18]은 원격 제어가 가능한 신생아 건강 스마트 모빌의 테스트 단계 전체 구현 현황이다. 보완 사항으로는 라즈베리파이 카메라를 통한 영상 스트리밍 기능에서의 픽셀 깨짐 현상의 원인을 분석하고 문제를 해결해야 하며, 추가 기능으로 설계된 동화책 읽어주기 기능을 구현중에 있다. 외부 프레임의 경우 하단 프레임이 완성되었으며 센서 모듈별 프레임을 설계하고 전체적으로 표면을 다듬고 정돈하는 단계이다.

[ 표 18 : 스마트 모빌 구현 현황 ]

플랫폼		모듈	진도	보완 사항
아두이노		온습도 센서	100%	없음
		미세먼지 센서	100%	없음
		비접촉식 온도 센서	100%	없음
		LED	50%	구현 중
라즈베리 파이		DC모터	100%	없음
		스피커	100%	없음
		카메라	90%	픽셀 깨짐 현상 테스트 필요
웹 서버			100%	없음
애플리케이션	핵심기능	환경데이터 구현	100%	없음
		음악 재생	100%	없음
		모빌 원격제어	90%	모빌 회전 속도 낮추어 구현
		실시간 영상 스트리밍	90%	픽셀 깨짐 현상 테스트 필요
	부가기능	SOS	100%	없음
		육아정보	100%	없음
		D-day	100%	없음
추가기능		동화책 읽어주기	50%	구현 중
프레임 제작			60%	모듈별 프레임 설계 필요
전체진행률			90%	

[표 19]는 원격 제어가 가능한 신생아 스마트 모빌의 모듈별 테스트 결과이다. 아두이노와 라즈베리 파이에 부착된 각 모듈별로 10가지 항목에 대해 테스트를 진행하였다. 10가지 항목은 정상 동작하는가, 회로상의 문제가 없는가, 센서 자체에는 문제가 없는가, 정상 데이터가 송수신되는가, node.js 서버에서 데이터가 정상적으로 송수신되는가, 애플리케이션에서 데이터가 정상적으로 수신, 출력되는가, 구동 중 동작이 정지되지 않는가에 대해 평가하였다. 아두이노에 부착된 센서는 온도 센서, 미세먼지 센서, 비접촉식 온도 센서이며 라즈베리 파이에 부착된 센서는 DC 모터, 스피커, 카메라(라즈베리파이)이다.

[ 표 19 : 모듈별 테스트 결과 ]

○ : 정상동작 △ : 수정 완료 × : 수정 필요						
평가 항목	아두이노			라즈베리파이		
	온습도 센서	미세먼지 센서	비접촉식 온도센서	DC 모터	스피커	카메라
1. 정상 동작 여부	○	○	○	○	○	○
2. 회로상의 문제 여부	○	○	×	△	○	△
3. 센서 자체의 문제 여부	○	○	○	○	○	△
4. 정상 데이터 송신 여부	○	△	○	○	○	○
5. 정상 데이터가 수신 여부	○	△	○	○	○	○
6. node.js 서버 데이터 수신	○	○	○	○	○	○
7. node.js 서버 데이터 송신	○	○	○	○	○	○
8. 애플리케이션 데이터 수신	○	○	○	○	○	○
9. 애플리케이션 데이터 출력	○	○	○	○	○	○
10. 구동 중 동작 정지 테스트	○	○	○	○	○	○

[표 20]은 온습도 센서와 스피커에 대한 테스트 결과 표이다. 온습도 센서(DHT11)의 경우 평가 항목 1~10에 대하여 정상 동작하는 것을 확인할 수 있었다. 라즈베리 파이에 부착된 스피커의 경우에도 마찬가지로 평가 항목 1~10에 대하여 정상 동작하는 것을 확인할 수 있었다.

[ 표 20 : 온습도 센서와 스피커 테스트 결과 ]

○ : 정상동작 △ : 수정 완료 × : 수정 필요

평가 항목 \ 모듈 이름	아두이노	라즈베리파이	문제점
	온습도 센서	스피커	
1. 정상 동작 여부	○	○	없음
2. 회로상의 문제 여부	○	○	
3. 센서 자체의 문제 여부	○	○	
4. 정상 데이터 송신 여부	○	○	
5. 정상 데이터가 수신 여부	○	○	
6. node.js 서버 데이터 수신	○	○	
7. node.js 서버 데이터 송신	○	○	
8. 애플리케이션 데이터 수신	○	○	
9. 애플리케이션 데이터 출력	○	○	
10. 구동 중 동작 정지 테스트	○	○	



[ 표 21 ]은 아두이노에 부착된 미세먼지 센서와 비접촉식 온도센서에 대한 테스트 결과 표이다. 미세먼지 센서(PM 2008M)는 테스트 도중 비정상적인 데이터값이 송수신되는 것이 확인되었다. 데이터가 송수신 과정에서는 문제가 발생하지 않았으나 데이터값이 유효하지 않은 값이 전송되었다. 미세먼지 센서(PM2008M)의 회로를 전체적으로 재 점검하여 해결할 수 있었다.

비접촉식 온도센서(MLX90614)는 테스트 과정에서 회로상의 문제가 존재함을 확인할 수 있었다. 비접촉식 온도센서로 사용되는 MLX90614 모듈의 경우 점퍼 케이블을 연결하는 부분이 원형 구멍에 납땜을 해야 하는 구조로 되어있다. 하지만 모듈 보드에 존재하는 원형 구멍 사이의 간격이 매우 좁고, 현재 납땜에 사용되는 Paste용액의 부재로 인해 차후 정밀한 납땜을 통해 수정해야 한다.

[ 표 21 : 미세먼지 센서와 비접촉식 센서, DC모터, 카메라 테스트 결과 ]

○ : 정상동작 △ : 수정 완료 × : 수정 필요			
평가 항목	아두이노		문제점
	미세먼지 센서	비접촉식 온도센서	
1. 정상 동작 여부	○	○	없음
2. 회로상의 문제 여부	○	×	비접촉식 온도센서 납땜 필요
3. 센서 자체의 문제 여부	○	○	없음
4. 정상 데이터 송신 여부	△	○	미세먼지 센서 회로 재점검을 통해 해결
5. 정상 데이터가 수신 여부	△	○	미세먼지 센서 회로 재점검을 통해 해결
6. node.js 서버 데이터 수신	○	○	없음
7. node.js 서버 데이터 송신	○	○	없음
8. 애플리케이션 데이터 수신	○	○	없음
9. 애플리케이션 데이터 출력	○	○	없음
10. 구동 중 동작 정지 테스트	○	○	없음

[표 22]는 라즈베리파이에 부착된 DC모터와 카메라 모듈에 대한 테스트 결과 표이다. DC 모터는 비접촉식 온도센서와 마찬가지로 회로상의 문제가 있었으나 DC모터는 납땜해야 할 부분의 간격이 넓기 때문에 납땜을 통해 문제를 해결할 수 있었다. 또한 DC 모터는 모빌의 상단에 부착되는 모듈이므로 여러 개의 점퍼 케이블을 연결해야 했다. 점퍼 케이블을 연결하는 과정에서 접촉문제가 발생했으나 절연테이프를 이용하여 해결하였다.

라즈베리 파이 전용 카메라 모듈은 스트리밍 도중 영상의 픽셀 깨짐 현상이 발생하였다. 픽셀 깨짐 현상은 불규칙적으로 나타났다. 정상적인 스트리밍 영상이 애플리케이션을 통해 출력되는 도중 불규칙적으로 일부분 픽셀이 깨지는 영상이 전송되는 경우가 발생했다. 해당 문제의 경우 카메라 모듈을 교체하여 재 테스트 과정을 거친 뒤, 라즈베리파이 카메라 케이블(RPI Camera cable)과 라즈베리파이 카메라 모듈 중 문제점이 있는 부분을 파악하여 교체하도록 한다.

[ 표 22 : DC모터와 카메라 테스트 결과 ]

○ : 정상동작 △ : 수정 완료 × : 수정 필요			
평가 항목	라즈베리파이		문제점
	DC모터	카메라	
1. 정상 동작 여부	○	○	없음
2. 회로상의 문제 여부	△	△	DC모터 : 납땜을 통해 해결 카메라 : 픽셀 깨짐 현상 발생
3. 센서 자체의 문제 여부	○	△	카메라 : 픽셀 깨짐 현상 발생 재 테스트 필요
4. 정상 데이터 송신 여부	○	○	없음
5. 정상 데이터가 수신 여부	○	○	없음
6. node.js 서버 데이터 수신	○	○	없음
7. node.js 서버 데이터 송신	○	○	없음
8. 애플리케이션 데이터 수신	○	○	없음
9. 애플리케이션 데이터 출력	○	○	없음
10. 구동 중 동작 정지 테스트	○	○	없음

[표 23]은 원격 제어가 가능한 신생아 스마트모빌의 전체 테스트 결과이다. 전체 테스트는 강도(Stress) 테스트와 성능(Performance) 테스트의 목적을 가진다. 테스트 도중 동작을 방해하는 몇 가지 문제가 발생했다.

첫 번째 문제는 라즈베리파이 3B+ 보드에서 데모 환경에서 사용할 모바일 핫스팟 Wi-Fi의 끊김 현상이 발생하는 것이었다. 라즈베리 파이를 기존에 사용하던 3B+에서 4B로 업그레이드 하였음에도 지속적으로 끊김 현상이 발생하였다. 해당 문제는 무선 랜 카드(USB dongle)를 USB 3.0 포트에 연결하여 해결할 수 있었다.

두 번째 문제는 아두이노와 라즈베리 파이 간 시리얼 통신 과정에서 발생했다. 라즈베리 파이에서 동작하는 node.js 서버는 자바스크립트 언어로 작성된 서버이다. 자바스크립트가 싱글 스레드 기반 언어이다 보니 아두이노에서 빠른 속도로 전송되는 실시간으로 환경정보 데이터로 인해 서버의 부하가 일어났다. 해당 문제는 아두이노 보드에서 node.js 서버로 전송되는 환경정보 데이터의 전송 속도를 느리게 조정함으로써 해결하였다.

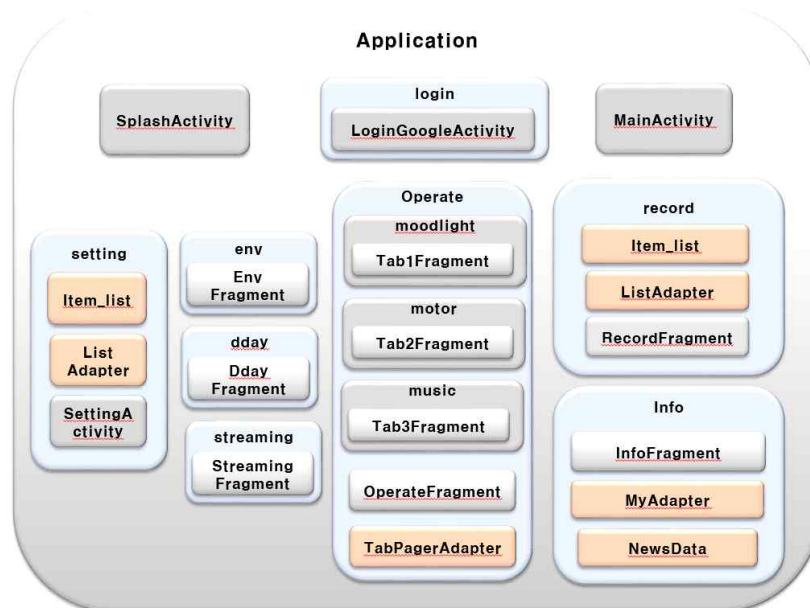
[ 표 23 : 전체 테스트 결과 ]

○ : 문제없음 △ : 문제수정 × : 문제발생					
평가 항목		모듈 이름	테스트 결과	문제점	해결방안
강도 테스트	1. 과다정보 송신 테스트		△	아두이노에서 과다정보 송신	전송 속도를 늦춤
	2. 과다정보 수신 테스트		△	node.js 서버에서 과다정보 수신	아두이노에서 전송 속도를 늦춤
	3. 시스템 과부하 테스트		○	문제 없음	문제 없음
성능 테스트	1. 응답 시간 속도 테스트		○	문제 없음	문제 없음
	2. 특정 시간 내 처리 테스트		○	문제 없음	문제 없음
	3. 사용자 불편함 테스트		○	문제 없음	문제 없음
기능 테스트	1. Wi-fi 끊김 문제		△	Wi-fi 불규칙적 끊김 문제	무선 랜카드를 이용해 문제를 해결

## 7. Coding&Demo

### 7.1 Coding

[그림 28]은 애플리케이션 구성도이다. Splash Activity에서 최초 권한을 획득한 후 LoginGoogle Activity를 이용해 로그인을 한다. 환경정보와 디데이 기능, 스트리밍 기능은 별개의 Fragment를 통해 작동하고 무드등, 모터, 음악 재생은 TabFragment를 통해 사용자에게 제공된다. 육아정보는 Info Fragment를 통해 사용자에게 제공된다.

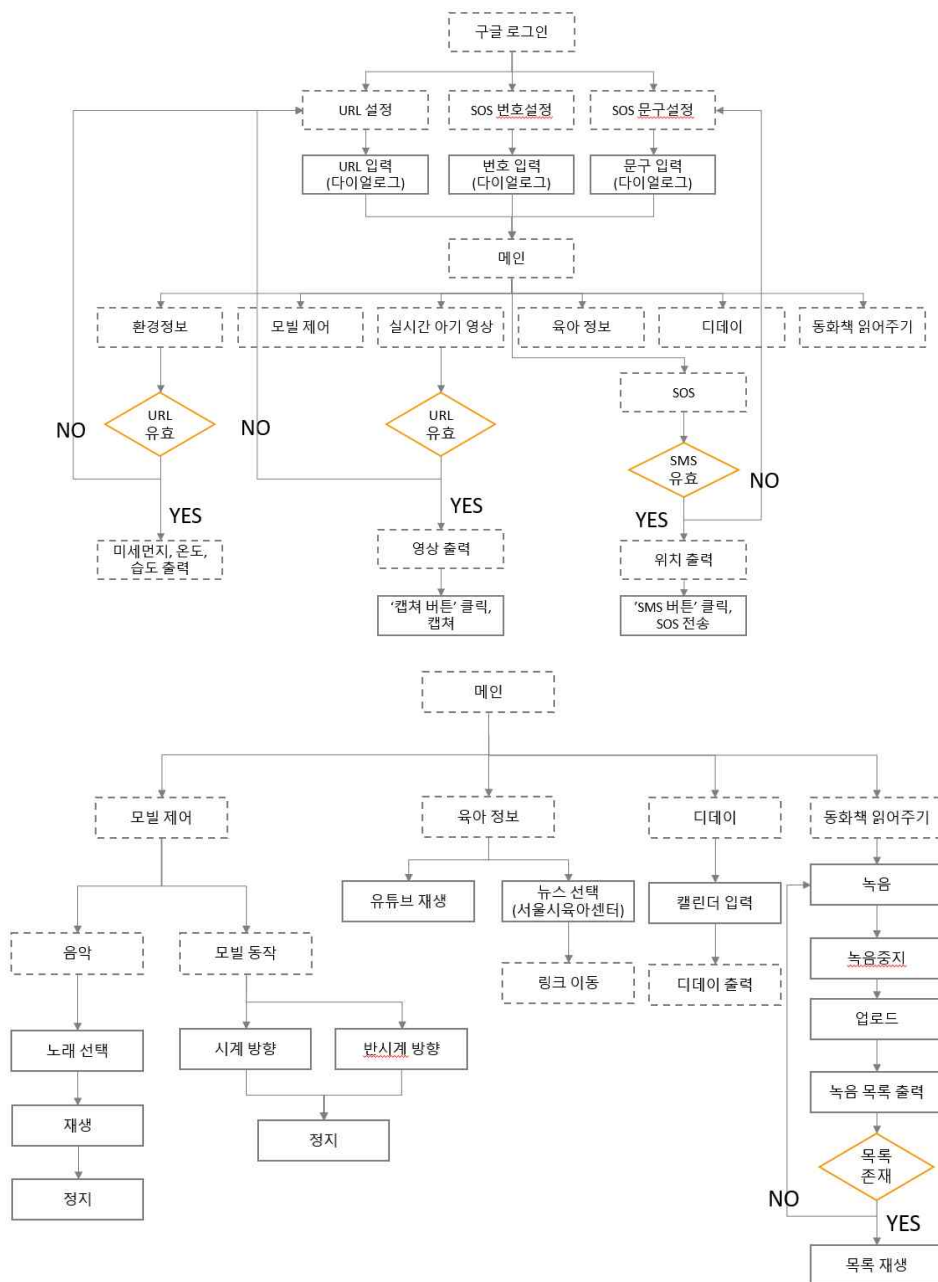


[ 그림 28 : 애플리케이션 구성도 ]

[그림 29]는 애플리케이션 흐름도이다. 먼저, 구글 로그인을 하고 URL, SOS번호, SOS문구를 각 다이얼로그를 통해 입력한다. 입력을 마친 후에 첫 화면으로 환경정보를 출력한다. 환경정보, 실시간 아기 영상 페이지는 URL이 유효하지 않을 시에 다시 설정 페이지로 인텐트한다. URL이 유효할 시에 환경정보는 미세먼지, 온도, 습도를 출력하고, 실시간 아기영상은 아기영상을 출력하고, 캡처를 할 수 있게 된다. SOS 페이지는 SMS 번호가 유효한지 확인하고 유효할 시에 위치를 출력하고, SMS 전송을 할 수 있게 된다. 모빌 제어 페이지는 음악과 모빌 동작 탭으로 이루어져 있으며 음악 탭에서는 노래를 선택하여 재생 및 정지를 할 수 있다. 모빌 동작 탭

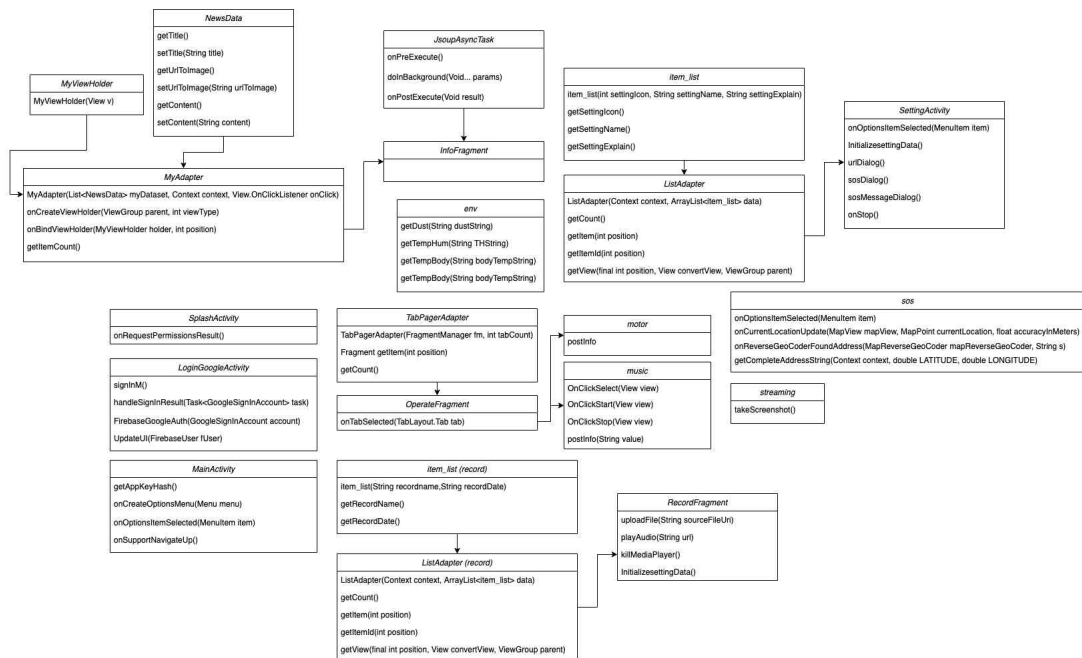
에서는 모빌을 시계 방향과 반시계 방향으로 동작시킬 수 있으며, 정지도 가능하다. 육아 정보 페이지에서는 육아 정보 유튜브를 재생할 수 있고, 실시간 뉴스 리스트를 터치하여 서울시육아센터 링크로 이동할 수 있다.

디데이 페이지에서는 캘린더에서 날짜를 선택하여 디데이를 자동 계산하여 볼 수 있다. 동화책 읽어주기 페이지에서는 부모가 목소리를 녹음하여 서버에 업로드하고, 전에 녹음한 목록 중 선택하여 모빌을 통해 아이에게 목소리를 들려줄 수 있다.



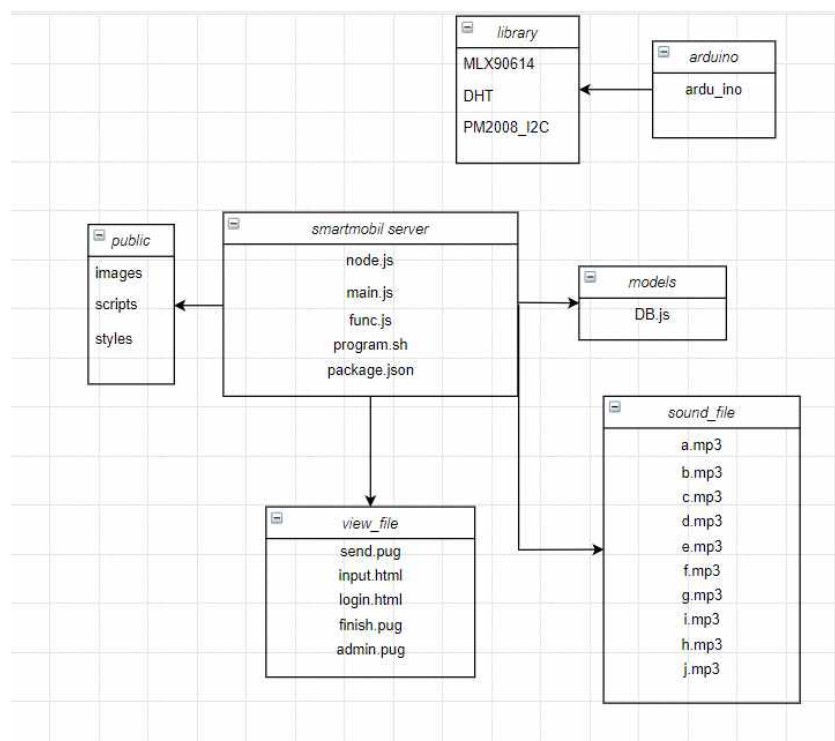
[ 그림 29 : 애플리케이션 흐름도 ]

[그림 30]은 애플리케이션 클래스 다이어그램이다. 클래스 및 함수를 다이어그램으로 나타내었다. 모빌 동작 클래스의 TabPagerAdapter는 OperateFragment를 통해 모터, 음악 탭을 초기화하고 드래그해서 페이지 전환을 가능하게 한다. item\_list와 ListAdapter는 RecordFragment의 리스트뷰를 나타내어주는 클래스이다. 마찬가지로, NewsData는 getter, setter, MyAdapter는 뷰홀더의 역할을 하여 InfoFragment의 리스트뷰를 나타내어주는 클래스들이다. InfoFragment는 JsoupAsyncTask를 통해 웹 파싱을 한다. SettingActivity도 마찬가지로, item\_list와 LisAdapter를 통해 설정 리스트뷰를 나타낸다. 나머지 클래스들은 단일 클래스로 각 함수들을 포함하여 역할을 수행한다.



[ 그림 30 : 애플리케이션 클래스 다이어그램 ]

[그림 31]은 node.js 웹 서버 다이어그램이다. smartmobil server 에 node.js, main.js, func.js 등이 포함된다. 웹의 프론트 디자인과 관련한 파일들이 public에 포함이 된다. 데이터와 관련된 파일은 models 디렉토리에 포함된다. 온습도 값을 데이터베이스에 입력하고 호출하는 부분을 담당한다. 웹페이지의 구성 화면은 view\_file에 포함된다. 각각의 값들과 페이지를 구성한다. sound\_file에는 구성하는 음악파일들이 포함된다. 모빌 동작, 스피커 동작 시에 해당 파일들이 실행된다. 아두이노에는 ardu\_ino가 포함되고 필요한 라이브러리들은 라이브러리 파일에 포함된다.



[ 그림 31 : node.js 웹 서버 다이어그램 ]

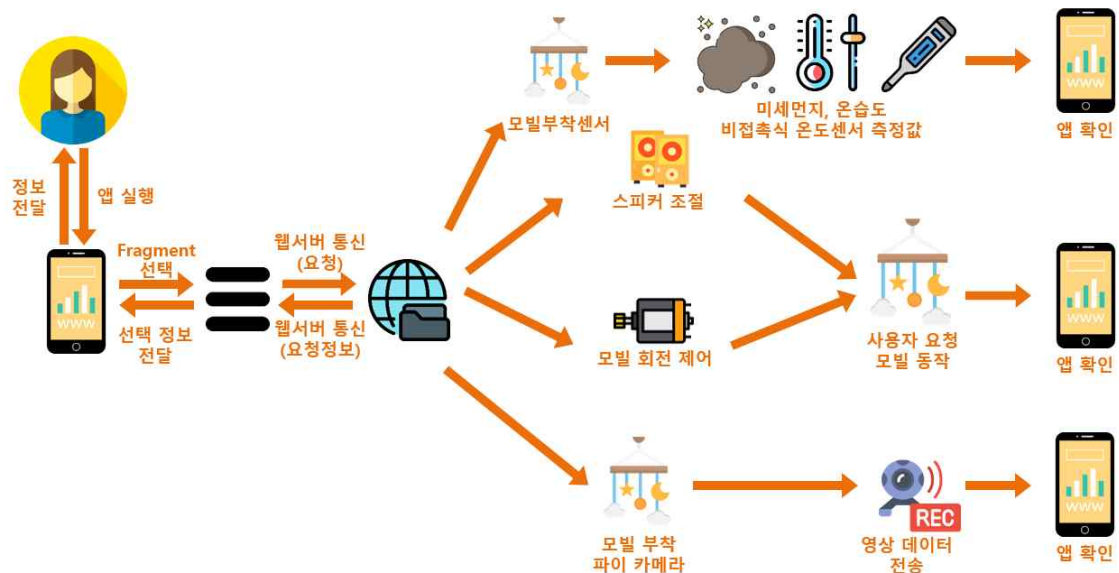
## 7.2 Demo

[그림 32]는 시나리오의 전체 흐름도이다. 사용자는 MyMobil 애플리케이션을 통해 기능에 해당하는 Fragment를 선택한다. 애플리케이션은 웹 서버와 통신하여 정보를 요청하고, 웹 서버는 해당하는 정보를 애플리케이션에게 제공한다. 사용자는 선택한 정보를 애플리케이션을 통해 제공받을 수 있다.

핵심 기능은 크게 세 가지이다. 첫 번째로는 모빌에 부착된 센서를 통해 신생아 주변 환경 데이터를 수집하여 사용자에게 제공하는 것이다. 환경데이터의 정보로는 미세먼지, 온도와 습도, 비접촉식 온도센서를 통한 체온 데이터가 있다. 사용자는 위와 같은 데이터들을 애플리케이션을 통해 확인하고, 신생아 주변 환경을 쾌적하게 유지하는 데 도움을 얻을 수 있다.

두 번째로는 애플리케이션을 이용하여 모빌에 부착된 스피커에서 음악을 재생하고 모빌의 회전 동작을 제어하는 기능이다. 사용자는 원격에서 애플리케이션을 이용해 모빌을 제어할 수 있다.

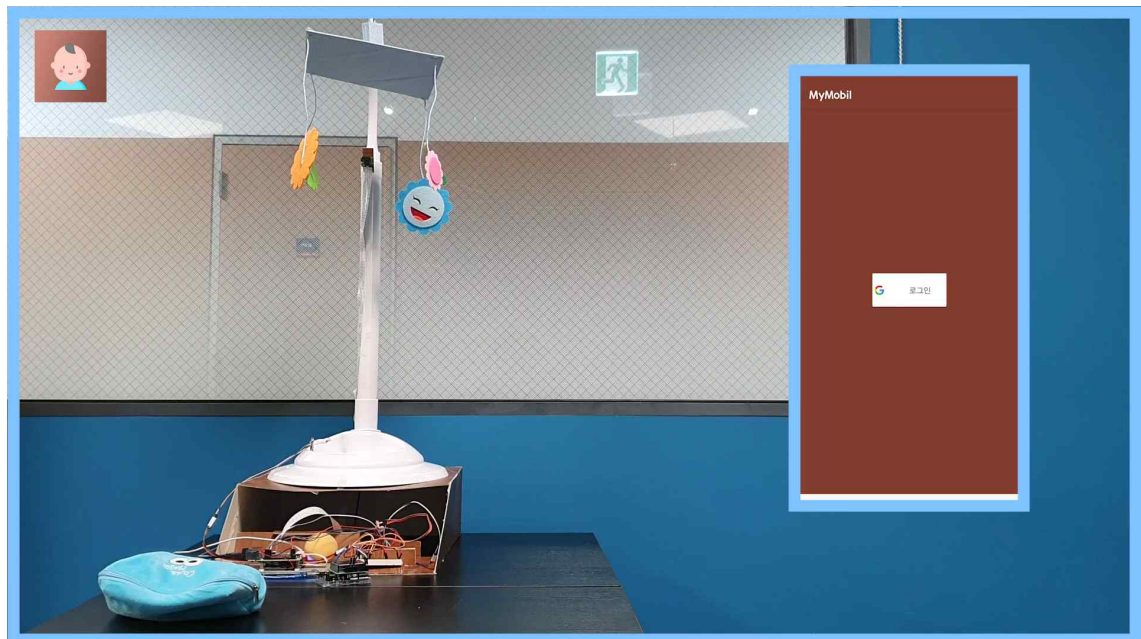
세 번째로는 라즈베리파이 카메라를 이용한 실시간 영상 스트리밍 기능이다. 모빌을 원격 동작 하는 도중, 사용자가 신생아의 안전에 유의할 수 있도록 애플리케이션을 이용해 실시간으로 아이의 모습을 확인할 수 있다.



[ 그림 32 : 시나리오 흐름도 ]



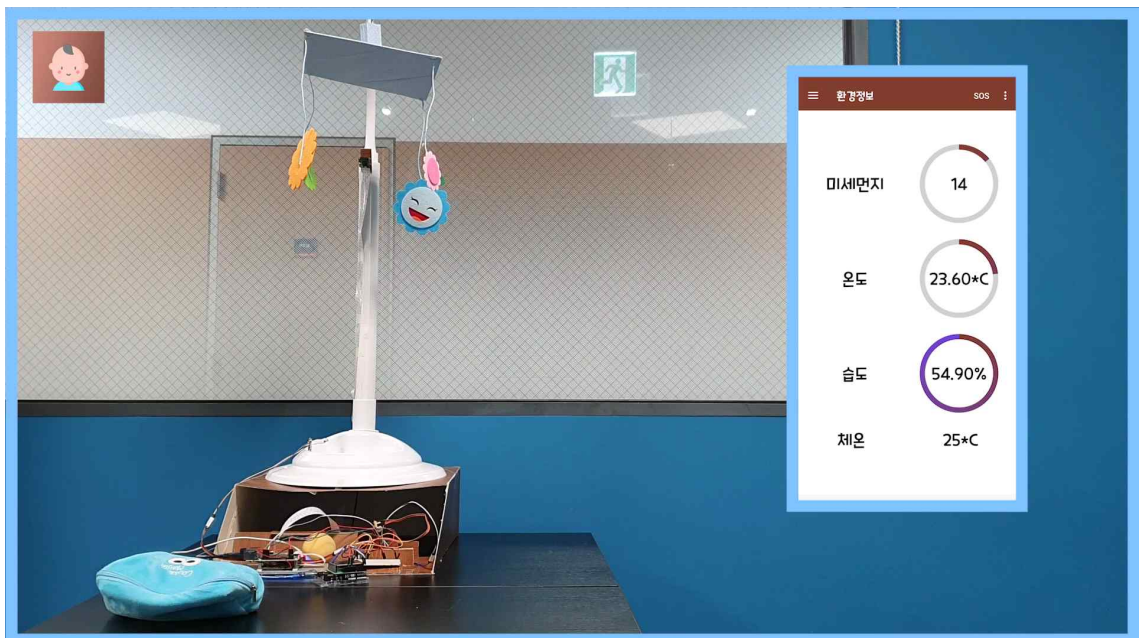
[그림 33] ~ [그림 42]은 데모 시연에 관한 시나리오이다. [그림 33]는 초기 My Mobil 애플리케이션을 실행시켰을 때, 구글 로그인이 실행되는 화면이다. [그림 34]은 라즈베리파이와 동일한 URL을 설정하는 화면이다. 이후 [그림 35]처럼 환경 정보 데이터를 수집해 사용자에게 제공한다. [그림 36]와 [그림 37]처럼 애플리케이션을 통해 모빌에 부착된 스피커와 모터를 제어하여 음악 재생 기능과 모빌 동작 제어 기능을 제공한다. [그림 38]은 라즈베리파이 카메라를 통해 스트리밍을 제공한다. [그림 39]는 각각 육아 정보, [그림 40]은 디데이 기능이다. [그림 41]의 SOS 번호와 문구를 설정하여 [그림 42]과 같이 지도 API를 통해 설정된 번호와 설정된 문구로 SOS 메시지를 전송한다.



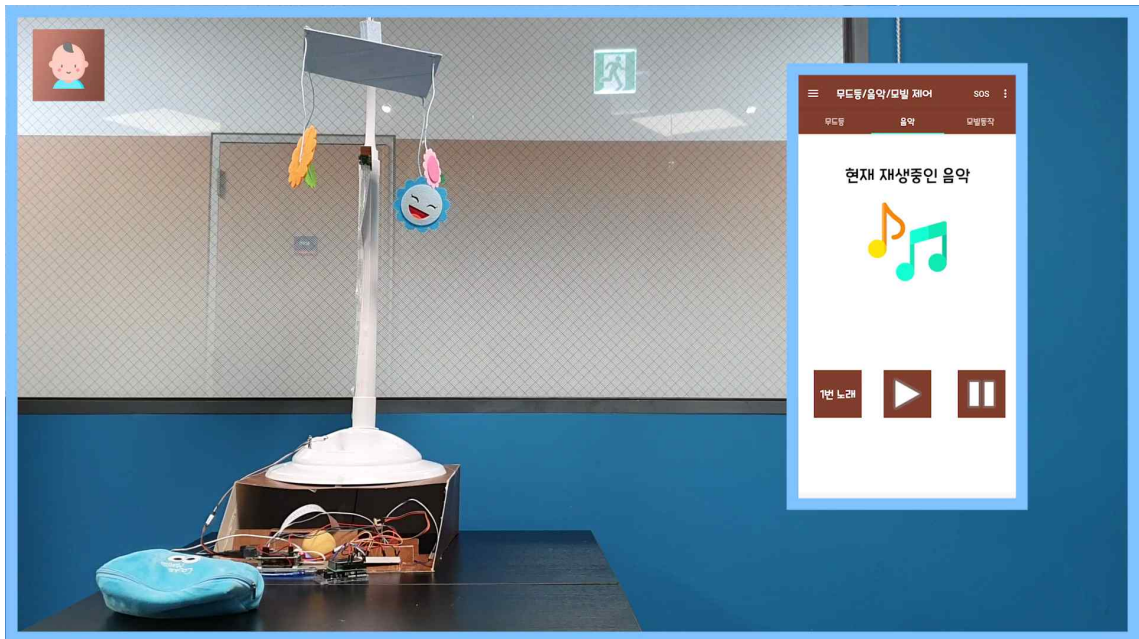
[ 그림 33 : 초기 애플리케이션 구글 로그인 실행 ]



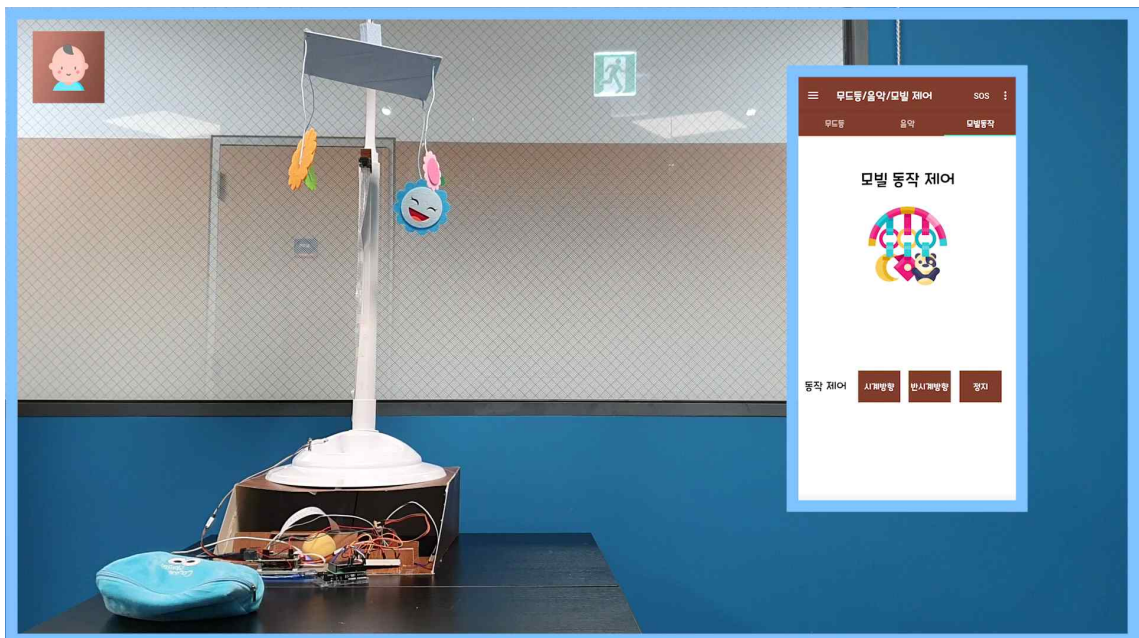
[ 그림 34 : 접속 URL 설정 ]



[ 그림 35 : 환경정보 데이터 표기 ]

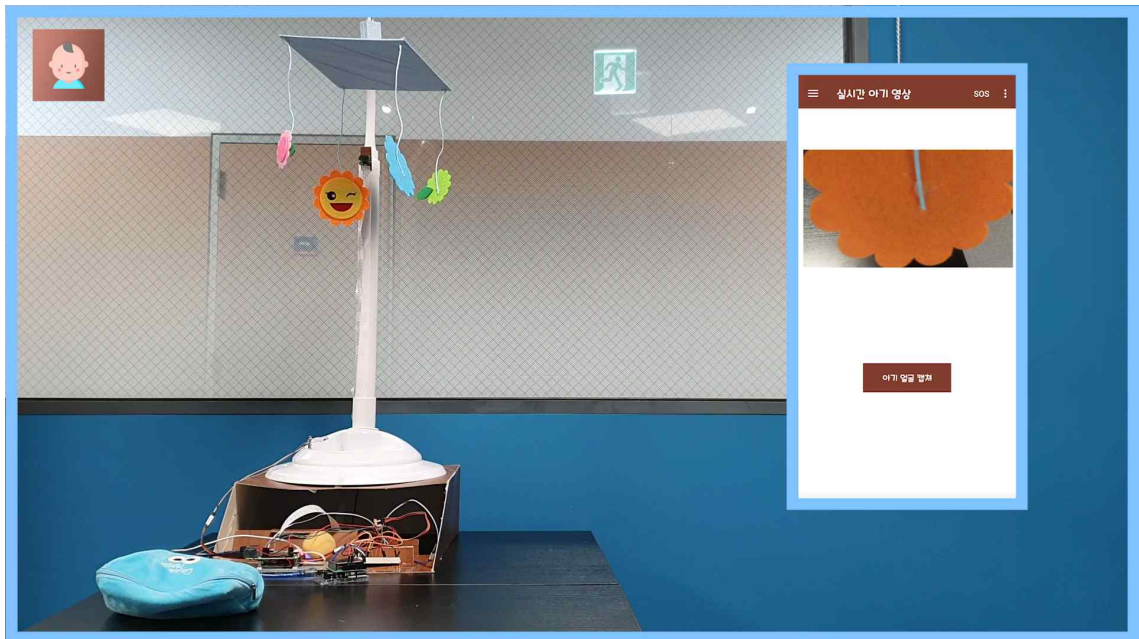


[ 그림 36 : 음악 재생 ]

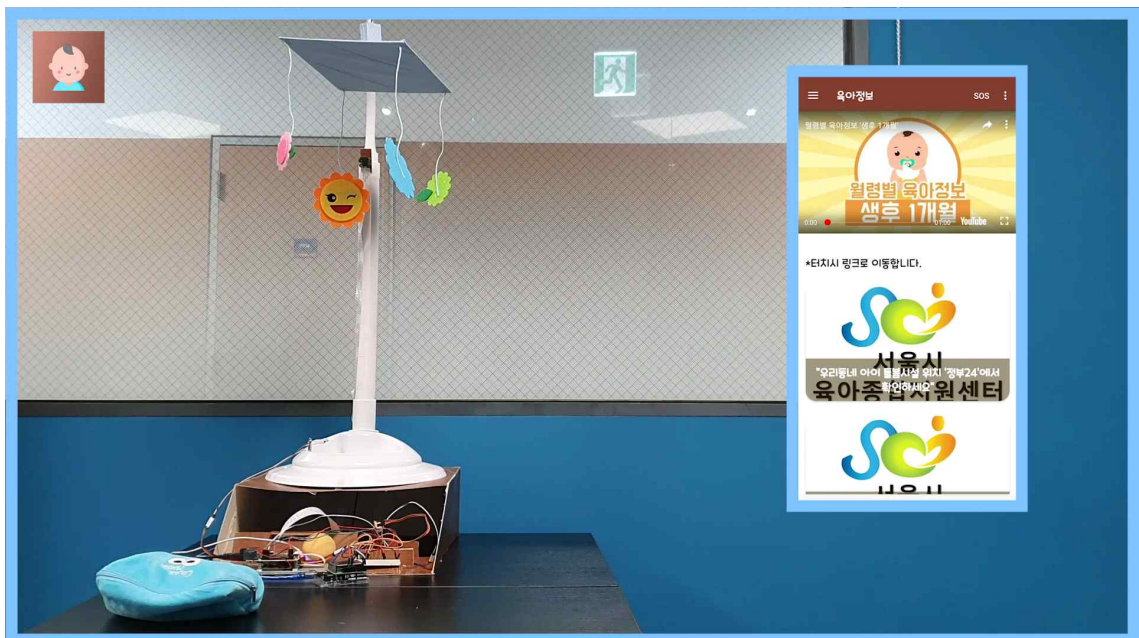


[ 그림 37 : 모빌 동작 제어 ]

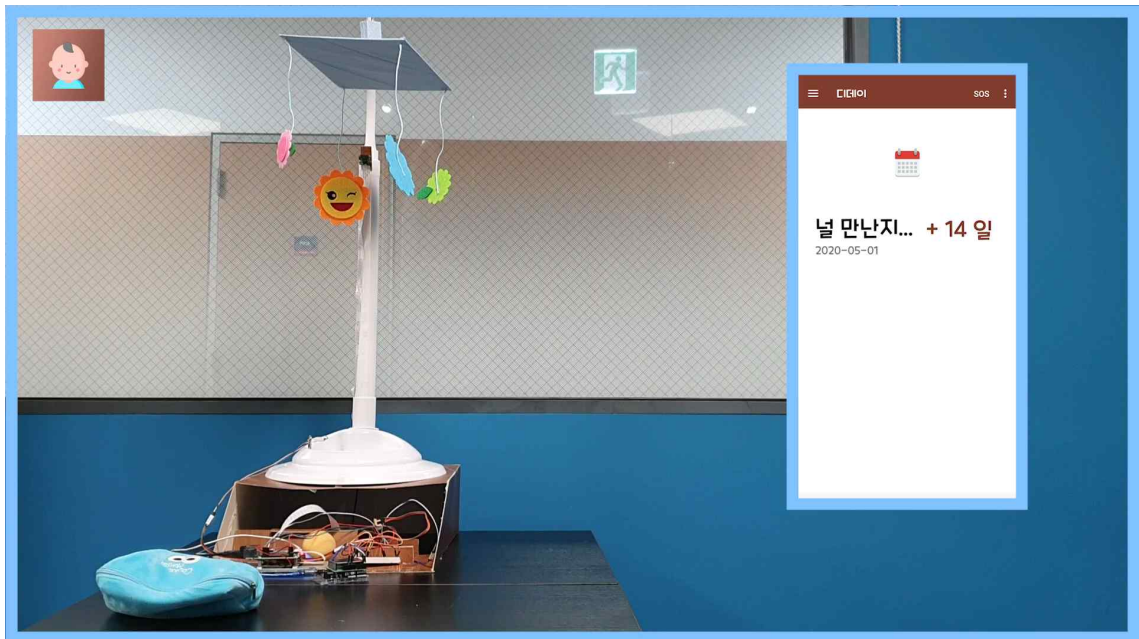




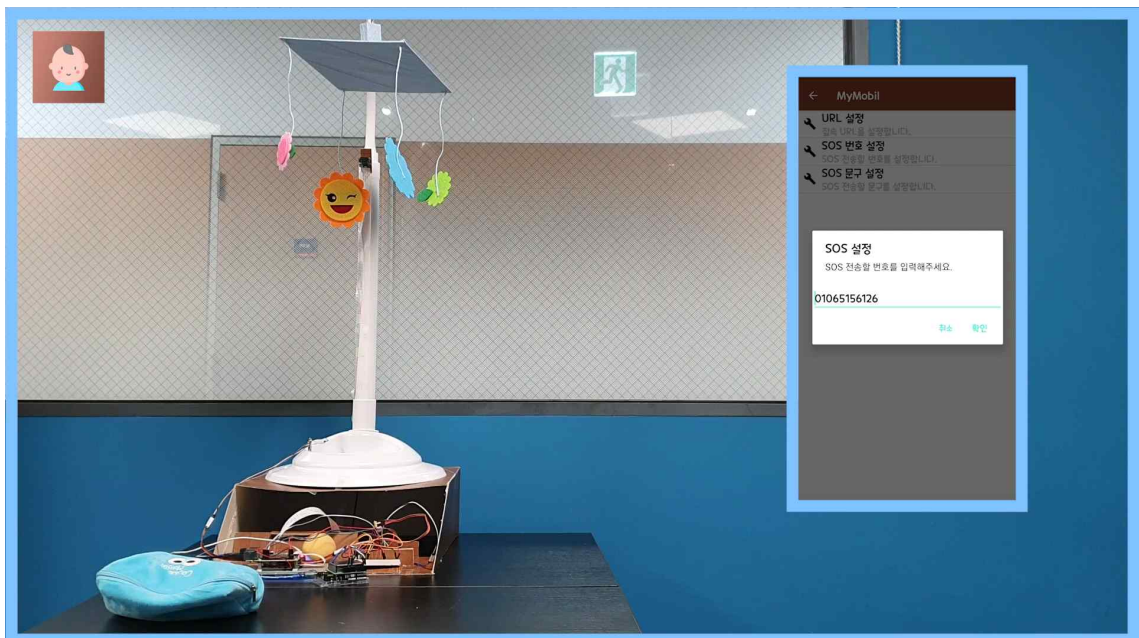
[ 그림 38 : 실시간 아기 영상 ]



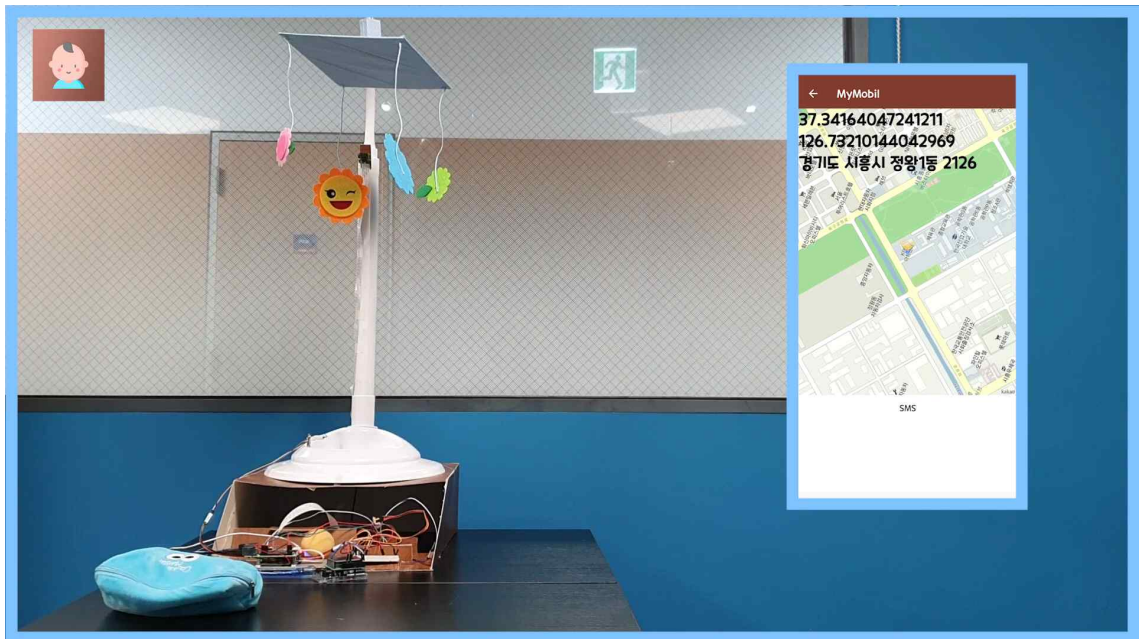
[ 그림 39 : 육아 정보 ]



[ 그림 40 : 다테이 ]



[ 그림 41 : SOS 번호, 문구 설정 ]



[ 그림 42 : SOS 기능 ]

### Ⅲ. 결론

#### 1. 연구결과

모듈의 정확한 데이터 수집과 애플리케이션의 올바른 동작 수행을 검증하기 위하여 미세먼지, 온습도, 체온 데이터를 측정하였고, 애플리케이션의 사용자 검증을 시행하였다.

표 24는 모듈 테스트의 결과이다. 과도하게 많은 양의 데이터가 웹 통신을 통해서 송수신되는 것을 확인하였고, 전송 속도를 낮추며 문제를 해결하였다. 시스템 과부하, 응답 시간 속도 테스트, 프로세스 테스트, 사용자 불편 테스트를 진행하였다.

추가로 라즈베리 파이에서 Wi-Fi가 자주 끊기는 현상이 발생하였으나 무선 LAN 카드를 사용하여 문제를 해결하였다.

[ 표 24 : 전체 테스트 결과 ]

○ : No problem × : Problem occurrence				
Evaluation Item		Test Result	Problem	Solutions
Test	Excessive information transmission test	×	Excessive information transmission	Slow down the transmission speed
	Excessive information reception test	×	Receive excessive information from the node.js server	Slow down the transmission speed on Arduino
	System overload test	○	No problem	No problem
	Response time speed test	○	No problem	No problem
	Process test within specific time	○	No problem	No problem
	User discomfort test	○	No problem	No problem
	Wi-fi disconnection problem	×	Wi-fi Irregular stuttering problem	Using a wireless LAN card solve a problem

표 25는 애플리케이션에서 모빌로의 데이터 전송시간을 장소별로 3가지 기능을 각각 10회씩 측정하여 도출한 평균 데이터이다. 측정 결과, 집, 카페, 학교 순서로 최대 0.5초의 전송시간의 차이가 존재하였다. 이는 사용자가 불편함을 느끼지 않는 범위 내에서 시스템이 정상 동작하는 것으로 판단된다.

[ 표 25 : 전체 테스트 결과 ]

Location	School	Cafe	House	(Seconds)
Measurement Data	1.5	1.2	1.0	
Melody Play	1.3	1.2	1.1	
Motor Operation	1.2	1.4	1.2	

현재 시중에 판매되고 있는 육아 편의성 증진을 위한 스마트 모빌 제품들은 영아 주변 환경 데이터를 제공하지 않는다. 또, 애플리케이션과 연동하여 모빌의 동작을 제어하거나, 애플리케이션을 통한 부가 기능을 제공하지 않는다.

신생아 건강을 위한 스마트 모빌은 영아를 위한 원격 제어 모빌 시스템은 기존 제품들의 단점을 보완하여 온도, 습도, 미세먼지 데이터를 제공한다. 그리고 애플리케이션을 통해 모빌 동작을 원격으로 제어할 수 있다. 또한, 부모의 목소리를 녹음하고 재생함으로써 신생아에게 심리적 안정을 주는 새로운 기능을 추가하고 육아 캘린더, SOS 기능과 같은 육아 편의성을 증진하는 부가 기능들을 구현하여 사용자는 쉽고 다양한 방식으로 육아에 도움을 받을 수 있다.

라즈베리 파이에서 고질적으로 발생하는 무선 랜 연결 끊김 문제의 경우, 유선 랜을 사용하거나 무선 랜 카드를 사용함으로써 해결할 수 있다. 애플리케이션에서 녹음한 파일이 라즈베리 파이 서버에서 재생이 되지 않는 문제는, 안드로이드 녹음 미디어 파일 포맷을 웹 서버에서 재생할 수 있게 해주는 서버 라이브러리를 사용하거나, 안드로이드에서 wav 파일로 녹음을 할 수 있게 해주는 OmRecorder API를 사용함으로써 해결할 수 있을 것으로 판단된다.



## 2. 작품제작 소요재료 목록

항목	세부 항목	용도	개수
SoC Board	Arduino UNO R3	H/W 개발용 임베디드 보드	1EA
	Raspberry Pi 4	H/W 개발용 임베디드 보드	1EA
	Raspberry Pi 4 Monitor	H/W 개발용 모니터	1EA
Sensor/Module	DHT22	온습도 측정 센서	1EA
	PM 2008M	미세먼지 측정 센서	1EA
	MLX 90614	비접촉식 온도 센서	1EA
	Camera Module V2, 8M	실시간 스트리밍 카메라 모듈	1EA
	SHZ-MT001	모빌 회전 동작 제어 DC 모터	1EA
	RGB LED	무드등 기능 LED	1EA
	Bread Board	H/W 개발용 브레드 보드	2EA
	점퍼선	H/W 개발용 점퍼선	2EA
Frame	시제품 모빌	기본 프레임	1EA
	나무 상자	(주문제작) 모빌 하단 프레임 제작	1EA
	실	모빌 프레임 제작	4EA
	인형	모빌 프레임 제작	4EA
	포맥스 판	모빌 프레임 제작 센서 프레임 제작	1EA

## 참고자료 .....

### 신생아 관련 논문

[1] 이군자&이명희, 신생아 감각자극에 관한 국내 연구 논문 분석, 아동간호학회지, vol.11, no.9, p.332 - 333, July. 2002.

[2] 안영미&손 민&김남희&강나래&강승연&정은미, 고위험신생아의 저체온증 현황 및 관련요인, 아동간호학회지, vol.23, no.4, pp.505-514, October. 2017.