

# Hand Sanitizer Manage System

팀 명	GakGak				
프로젝트명	Hand Sanitizer Manage System				
신청자 (팀장)	성명	유재현			
	생년월일	981029	소속	세종대학교	
	휴대전화	01024295483	이메일	slade12307@gmail.com	
구성원	성명	생년월일	소속	휴대전화	이메일
	최민지	980805	세종대학교	01083989020	gracemj805@gmail.com
	구세화	980528	세종대학교	01076865911	qlslrnlldyal5@naver.com
2021년 1월 25일					

**Team: GakGak**

구세화, 유재현, 최민지

## Contents

1. Motivation .....	3
1.1. Background.....	3
1.2. Problems.....	3
1.3. Solutions .....	3
2. System Overview.....	4
2.1. Key Functions .....	4
2.2. Expectations .....	4
3. Use Case .....	4
4. Procedures .....	6
4.1. User Registration.....	6
4.2. Resource Update.....	7
5. Implementation.....	8
5.1. Resource Structure .....	8
5.1.1. Present Resource Structure.....	8
5.1.2. Future Resource Structure .....	9
6 Demonstration.....	10
6.1.1. IoT Device .....	10
6.1.2. Web App Server Database .....	12
6.1.3. Login.....	13
6.1.4. Sign Up.....	15
6.1.5. Hand Sanitizer List .....	16
6.1.6. DashBoard.....	20
6.1.7. Data Update.....	23

## 1. Motivation

### 1.1. Background

1. 2020년 1월 국내 첫 코로나 바이러스 확진 환자 발생
2. 감염병의 확산이 심화되고, 방역 및 예방의 필요성 증가
  - 누적 확진 환자는 현재까지 계속 늘어나는 추세이며, 확산 속도를 늦추려면 적절한 방역과 예방이 필요하다.
  - 간편하고 용이하게 손을 소독할 수 있는 손소독제 비치, 마스크 착용, 주기적으로 손 씻기 등 예방방법이 존재한다.
3. 학교나 공공기관 등 다양한 장소에 손소독제 비치
  - 개인이나 단체에서 코로나 바이러스 예방을 위해 다양한 장소에 손소독제 비치
4. 코로나 바이러스가 장기화됨에 따라 손소독제의 지속적인 관리 필요
  - 손소독제는 소모품이기 때문에 관리자의 지속적인 점검 및 교체가 필요하다.

### 1.2. Problems

1. 관리자는 잔량 파악을 위해 직접 가서 눈으로 확인하는 수밖에 없다.
2. 손소독제는 여러 장소에 분산되어 있기 때문에 보다 효율적인 점검 및 관리 시스템이 필요하다.

### 1.3. Solutions

<1.2> 에서 언급한 문제점을 해결하기 위해서 IoT 기기를 이용하여 손소독제의 잔량을 원격으로 확인하고 시각화 하는 서비스를 고안하였다.

- 손소독제에 IoT 센서(Load Cell)를 부착하여 잔량 측정
- 관리자가 직접 이동할 필요 없이 원격으로 손소독제 잔량을 확인할 수 있다.
- 잔량을 시각화를 함으로서 효율적인 관리가 가능하다.
- 손소독제 잔량이 일정 용량 미만일 경우 경고 표시를 출력함으로써 손소독제가 고갈 되었지만 미처 확인하지 못하는 상황 방지
- 관리자의 필요없는 동선이 줄어들어 관리자의 코로나 감염 가능성 감소

## 2. System Overview

이 시스템은 손 소독제 관리 시스템의 데모버전을 목적으로 진행되었다.

### 2.1. Key Functions

1. 이 시스템에서는 라즈베리 파이를 이용한 IoT 센서를 통해 손소독제의 잔량 데이터를 서버로 전송하게 된다.
2. 서버에 존재하는 이 잔량 데이터는 Web 또는 App을 통하여 확인이 가능하다.
3. 만일, 손소독제의 잔량이 일정 용량 미만일 경우 Web 또는 App에서 관리자가 경고 표시를 확인할 수 있고, 서버의 데이터들을 활용한, 시간별 변화량과 같은 시각적 데이터를 확인할 수 있다.

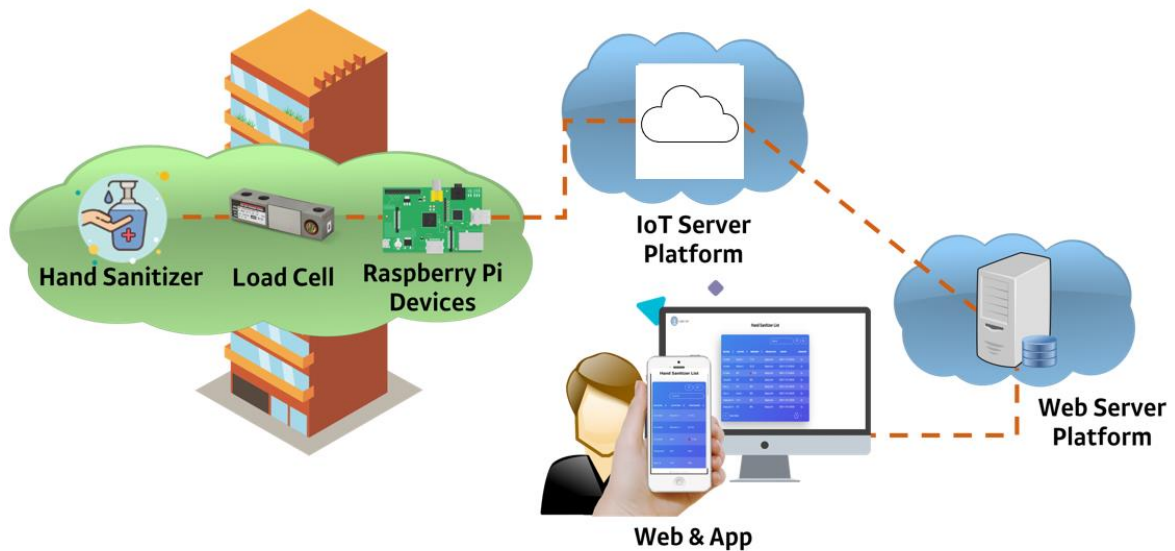
### 2.2. Expectations

1. 관리자가 아직 손소독제가 충분히 남아있는 위치에 방문하거나, 모두 고갈될 때까지 미처 확인하지 못하는 등의 비효율적인 관리를 방지할 수 있다.
2. 관리자가 직접, 원격으로 실시간 데이터로 잔량을 확인하며 효율적인 관리가 가능하다.
3. 손소독제의 시간별 잔량 데이터들을 활용하여 혼잡도를 파악하여 용량의 변화가 필요한 지를 체크하는 등 간접적인 효과 또한 얻을 수 있다.

## 3. Use Case

서비스 및 사용자 관점에서의 Use Case를 설명하고자 한다. 현재 손소독제 관리 현황과 이 프로젝트에서의 지향 방향을 일상에서 살펴보면 다음과 같은 설명이 가능하다:

- 손소독제는 건물이나 층, 강의실 등 여러 장소에 분산되어 위치하고 있다.
- 현재 손소독제 관리에서 손소독제들을 각 건물의 관리자가 직접 확인하며 관리하고 있다.
- 이 시스템 데모버전에서 목표로 한 IoT 서비스, 손소독제 관리 서비스를 이용하여 관리자는 원격으로도 손소독제 잔량 확인이 가능할 것이다.



[그림 1] 손소독제 관리 시스템의 Use Case Overview

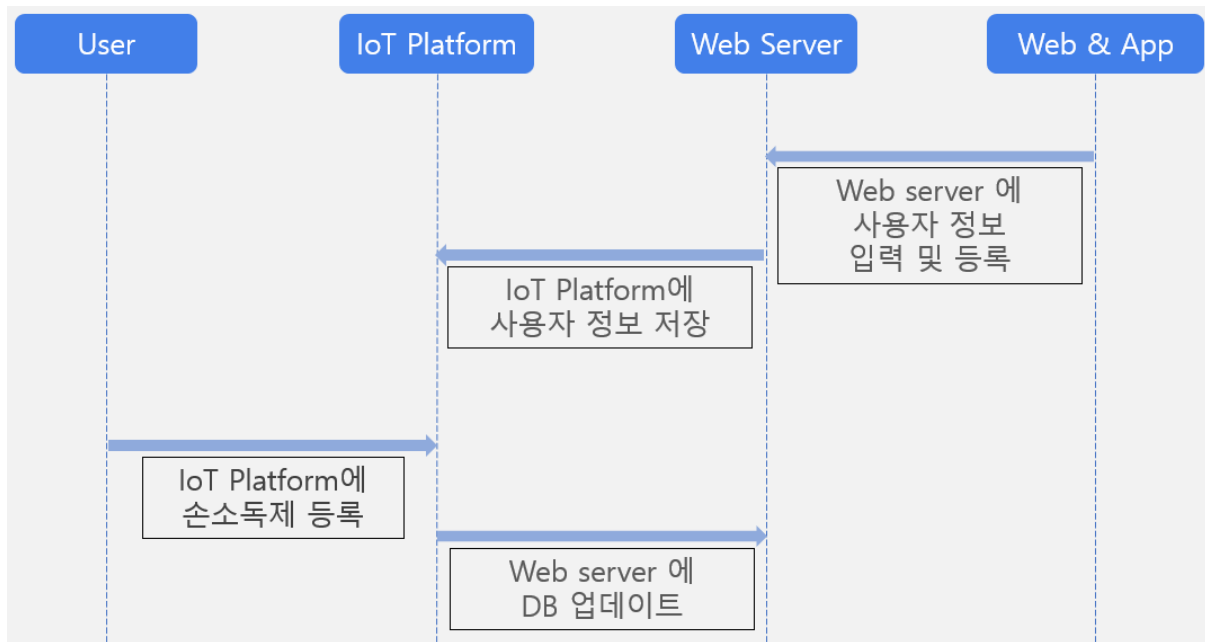
위 [그림 1]은 손소독제 관리 시스템의 Use Case 개요를 나타내며, 주요 구성 요소는 다음과 같다.

1. 본 Use Case에 보이는 손소독제는 여러 장소에 분산되어 위치하고 있으며, 각각의 손소독제는 Load Cell 센서를 통한 무게 측정이 가능하다.
2. Raspberry Pi Device는 Load Cell 센서와 연결되고, 손소독제와 IoT Server Platform 사이에 위치하여 HTTP를 통한 통신 기능을 제공한다.
3. Web Server Platform은 손소독제 정보, 관리자 정보를 저장하는 데이터베이스를 포함하며, IoT Server Platform에서 데이터를 불러와 Web이나 App을 통해 접속한 관리자에 해당하는 데이터와 페이지를 출력하는데 활용된다.
4. Web이나 App은 다음과 같은 기능들을 통해 원격 손소독제 관리 시스템 역할을 한다.
  - 관리자 등록 및 로그인 기능
  - 관리자가 관리하고 있는 손소독제에 해당하는 데이터들 리스트화
  - 건물 명, 위치, 시각 등의 조건 검색을 통한 리스트화 기능
  - 일정 용량 미만의 손소독제 잔량에 대한 경고 표시 기능
  - DASHBOARD를 통한 데이터의 시각화 기능

## 4. Procedures

Use Case를 통해 소개된 각 단계에 관한 메시지 흐름을 설명한다. <4.1> 장에서는 사용자를 등록하는 과정을, <4.2> 장에서는 리소스 업데이트 과정을 설명하도록 한다.

### 4.1. User Registration

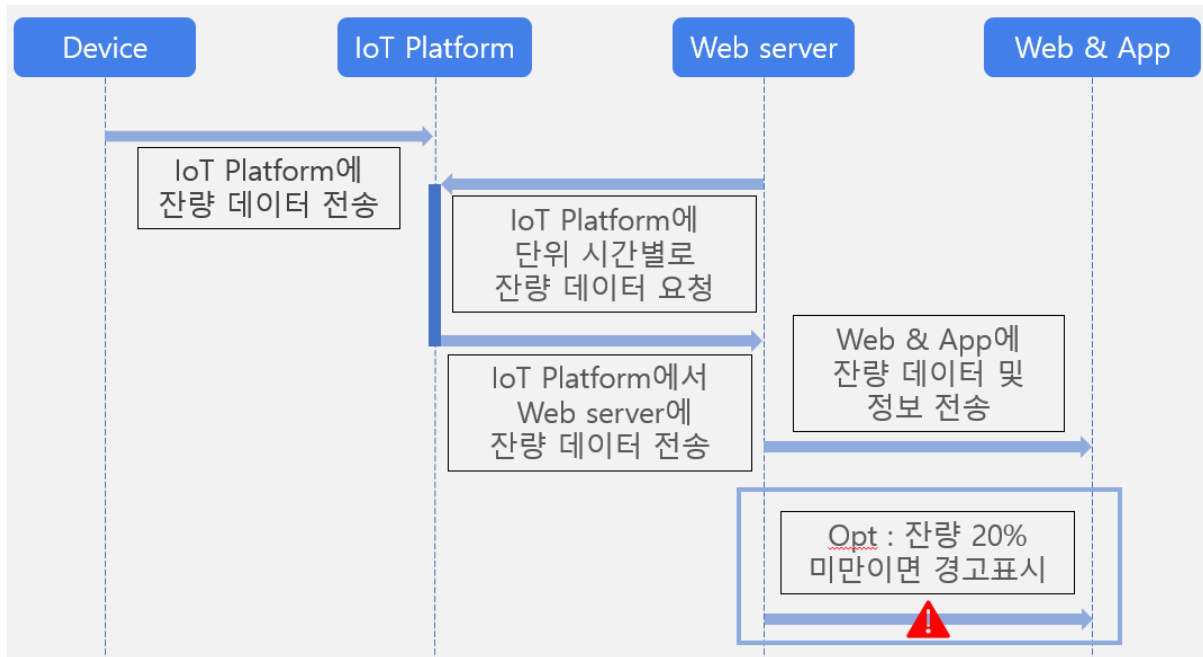


[그림 2] User Registration Procedure

사용자 등록과 관한 메시지 흐름은 [그림 3]에 보이며 다음 순서로 진행된다:

- 1)-1 웹 서버에 사용자 정보 입력 및 등록
- 1)-2 다음 웹 서버는 IoT 플랫폼에 사용자 정보를 저장한다.
- 2) oneM2M 서비스 플랫폼 (IN-CSE)에 손 소독제 잔량 측정 디바이스 (ASN-AE)를 등록
- 3) 웹 서버 (IN-AE)에 데이터 베이스를 업데이트

## 4.2. Resource Update



[그림 3] Resource Update Procedure

손 소독제 잔량 측정 디바이스로부터 전송된 무게 데이터를 웹 서버에 저장해 사용자에게 보이기 위한 과정이다. [그림 4]에서 보이며 순서는 다음과 같다.

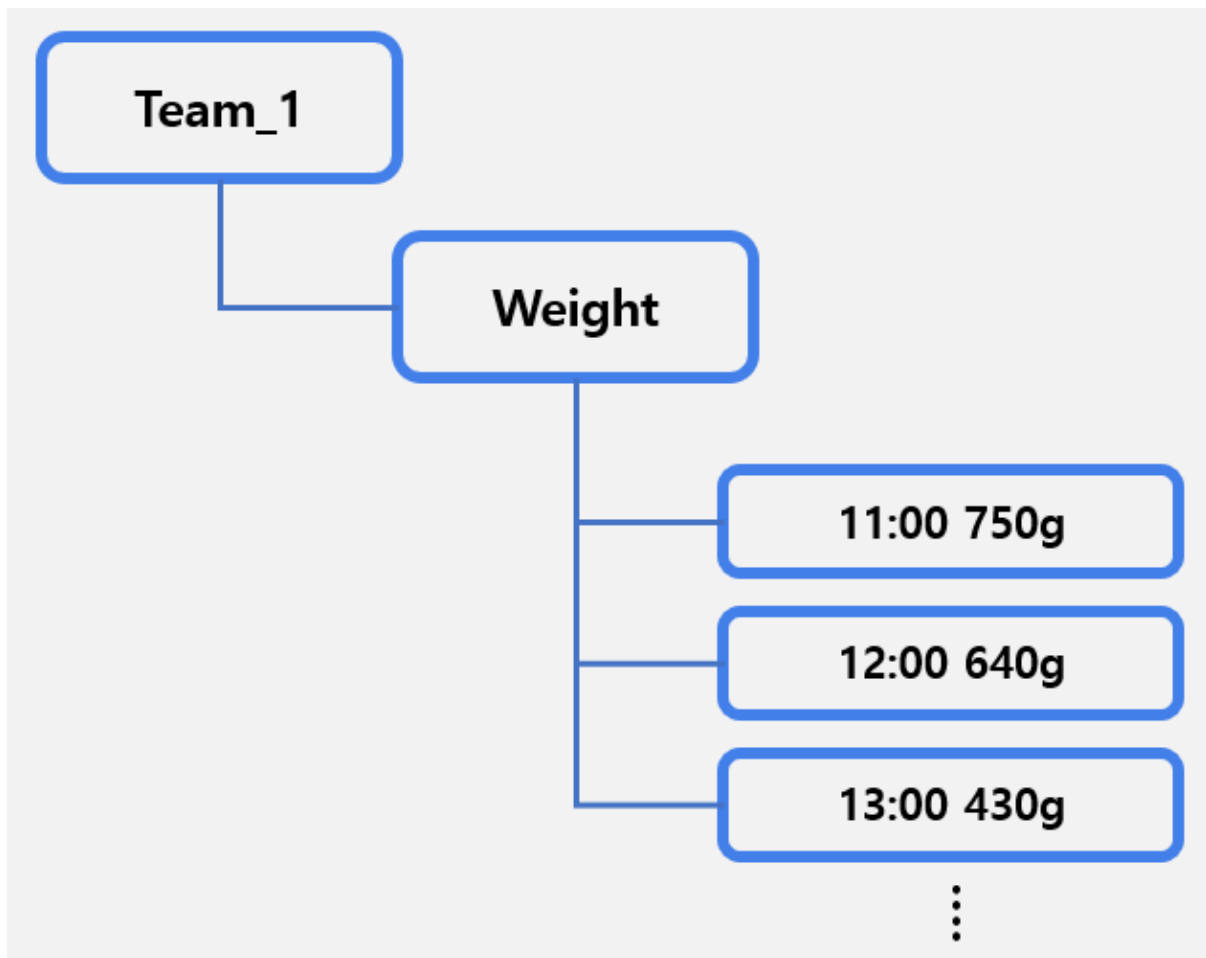
- 1) 손 소독제 잔량 측정 디바이스 (ASN-AE)는 단위 시간인 한 시간마다 oneM2M 서비스 플랫폼 (IN-CSE)에 측정된 무게 데이터를 보낸다.
- 2)-1 웹 서버 (IN-AE)는 oneM2M 서비스 플랫폼 (IN-CSE)에 단위 시간 별로 잔량 데이터를 요청한다.
- 2)-2 IN-CSE는 요청에 대한 응답으로 손 소독제 잔량 측정 디바이스 (ASN-AE)에서 전송 받은 데이터를 IN-AE로 전송한다.
- 3) 웹 서버 (IN-AE)는 Web&App에 손 소독제와 관련한 정보인 잔량 데이터, 측정 시간, 특정 시간 내 변화량 등을 보낸다.
- 4) IN-AE에서 Web&App으로 송신하는 데이터에 만약 잔량 데이터가 필터링 되는 기준인 20% 미만이 된다면, 그 기준에 따라 경고 표시를 포함해서 응답한다.

## 5. Implementation

### 5.1. Resource Structure

메시지 흐름과 Use Case를 구현하기 위해 필요한 리소스들에 관해 설명한다. 현재의 리소스(데모) 구조와 상용화 이후 리소스 구조에 대해 차례로 설명한다.

#### 5.1.1. Present Resource Structure



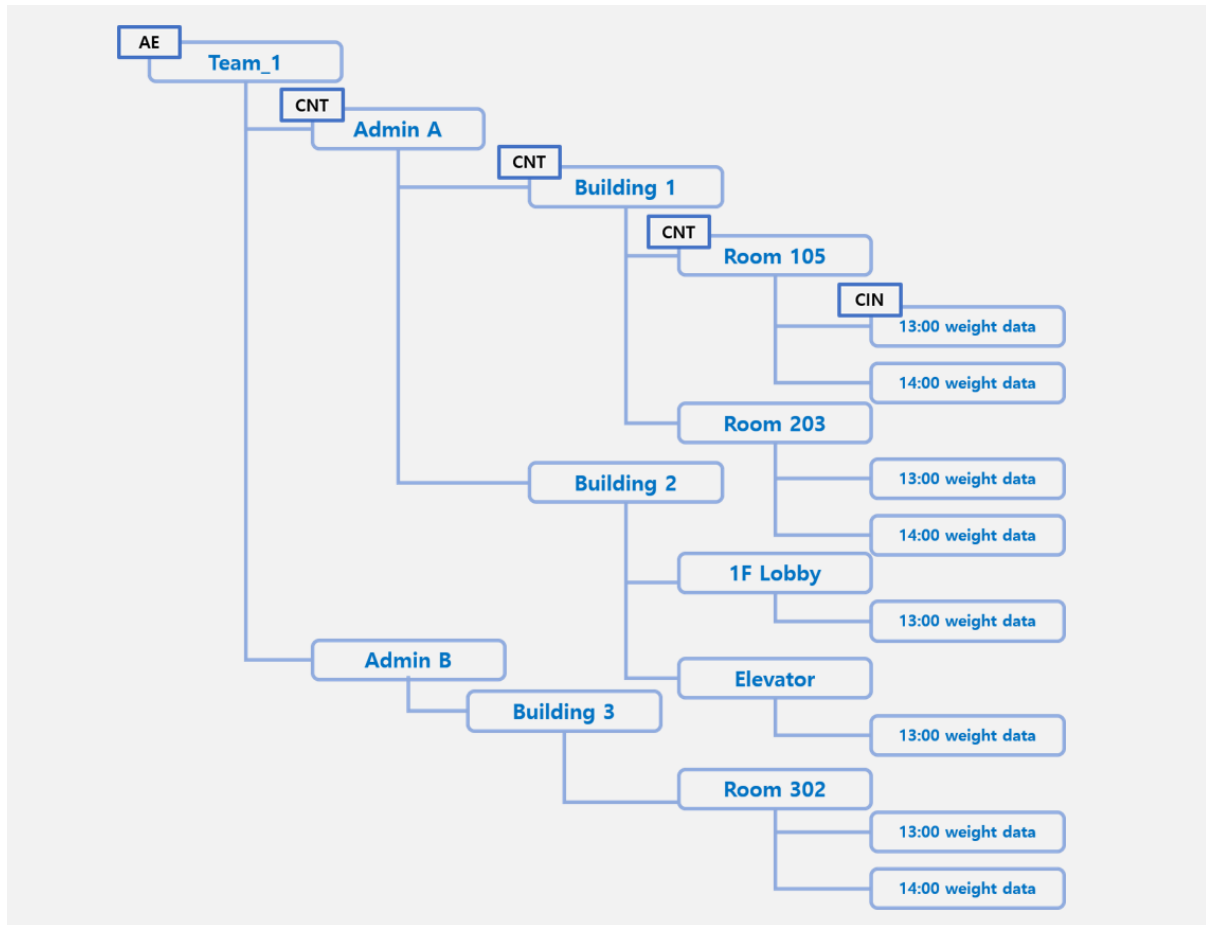
[그림 4] Present Resource Structure

현재의 리소스 구조 트리는 [그림 5]에 표시된 Team\_1이라는 이름으로 시작한다.

- Team\_1 AE하단에 Weight라 이름 지어진 container의 여러 contentInstance를 포함한다.
- Weight라는 이름의 container는 측정시간과 잔량 데이터에 해당하는 contentInstance를 포함한다.



### 5.1.2. Future Resource Structure



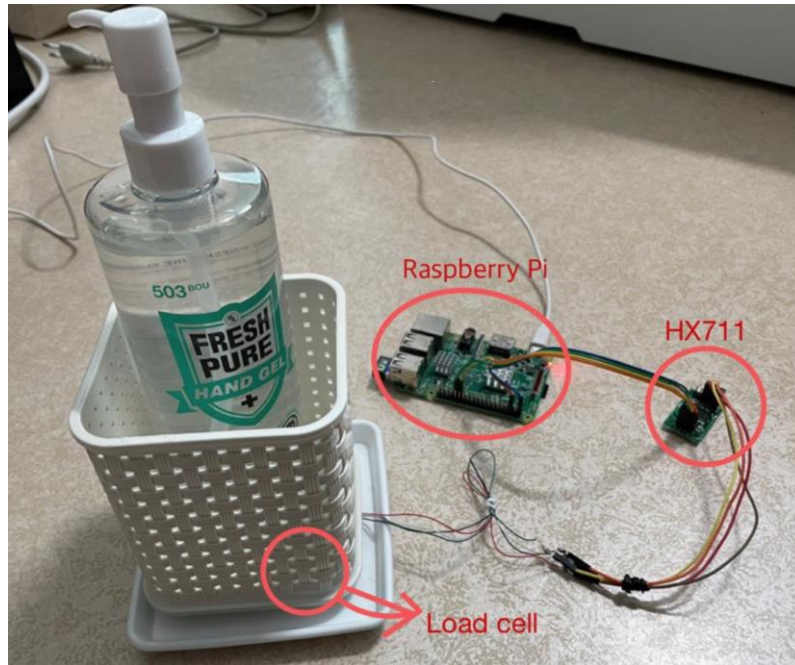
[그림 5] Future Resource Structure

상용화 이후의 리소스 구조 트리는 [그림 6]에 표시된 Team\_1 이라는 AE로 시작한다. Team\_1의 리소스 트리는 다음과 같은 하위 리소스로 구성된다.

- Admin A 라는 이름의 container는 Building에 맞게 이름 지어진 container의 하위 리소스로 손 소독제 위치에 따라 이름 지어진 container를 포함한다.
- 손 소독제 위치에 따라 이름 지어진 container는 측정 시간과 잔량 데이터에 해당하는 Content Instance를 포함한다.
- Admin B 라는 이름의 container는 Building에 맞게 이름 지어진 container의 하위 리소스로 손 소독제 위치에 따라 이름 지어진 container를 포함한다.
- 손 소독제 위치에 따라 이름 지어진 container는 측정 시간과 잔량 데이터에 해당하는 Content Instance를 포함한다.

## 6 Demonstration

### 6.1.1. IoT Device



[그림 6] 기기 구성

디바이스를 설계하는 데에는 Raspberry Pi 보드를 사용하였다. HX711 증폭센서와 로드셀은 납땜을 통해 연결하였고, 이를 Raspberry Pi 와 범퍼 케이블로 연결시켜 주었다. 로드셀 센서는 받침 판과 손 소독제 보관 통 사이에 부착하여 무게가 일정하게 측정되도록 고정시켜 주었다.

```

45  hx = HX711(5, 6)
46  hx.set_reading_format("MSB", "MSB")
47  hx.set_reference_unit(referenceUnit)
48  c=0
49  total=0
50  hx.reset()
51  hx.tare()
52
53  url = "http://203.250.148.89:7579/Mobius/Team_1/Weight"
54
55  while True:
56      try:
57          val = hx.get_weight(5)
58          val = int((float(val)-500)/45)
59          print(val)
60          hx.power_down()
61          hx.power_up()
62          time.sleep(0.1)
63      c=c+1
64
65      if((c%3600)==0):
66          payload="{\n      \"m2m:cin\": {\n      \"con\": \"\""+format(int(val),".0f")+"\"\\n    }\n}"
67          headers={
68              'Accept': 'application/json',
69              'X-M2M-RI': '12345',
70              'X-M2M-Origin': 'Duribeonx2',
71              'Content-Type': 'application/vnd.onem2m-res+json; ty=4'
72          }
73          response = requests.request("POST",url,headers=headers,data=payload)
74          print(response.text.encode('utf8'))
75
76          remainder = val
77
78          print("hi ",total," finish")
79          sys.exit()
80
81          print("Uploaded Weight Data!")
82
83  except (KeyboardInterrupt, SystemExit):
84      cleanAndExit()
85

```

[그림 7] 손소독제 무게 측정 &amp; 모비우스 서버(IoT서버)로 데이터 전송

로드셀 센서를 통해 무게는 실시간으로 계속 측정된다. 이때 1시간 간격으로 모비우스 서버로 HTTP 통신을 통해 현재의 손 소독제의 무게 데이터를 업로드 한다.

```

pi@raspberrypi:~/hand_sanitizer $ python hs_weight.py
Tare done! Add weight now...
-11
-9
-11
-10
-12
-12
1907
2162
2156
2155
2157
2159
2154
2149
2151
{"m2m:cin":{"rn":"4-20201203072928378ZYKx","ty":4,"pi":"HkQsV6Bwtv","ri":"SyQgJ4fUoD","ct":"20201203T072928","et":"20231203T072928","lt":"20201203T072928","st":32,"cs":4,"con
": "2151","cr":"Duribeonx2"}}
Uploaded Weight Data!
2149

```

[그림 8] 무게 측정 &amp; 모비우스로 데이터 전송

무게를 측정하고, 모비우스 서버로 무게데이터를 전송하는 것을 확인할 수 있다.

## 6.1.2. Web App Server Database

```

+-----+
| Tables_in_gakgak |
+-----+
| HS_admin          |
| HS_list           |
+-----+

```

[그림 9] Database Table

서버 내에서 MySQL을 이용해 데이터베이스를 구축하였다. 데이터베이스 내에는 HS\_admin, HS\_list 테이블이 구성되어 있다.

## (1) 관리자의 계정 정보를 저장하는 HS\_admin 테이블

Field	Type	Collation	Null	Key	Default	Extra	Privileges
USERID	varchar(20)	latin1_swedish_ci	NO	PRI	NULL		select,insert,update,references
PASSWD	varchar(20)	latin1_swedish_ci	NO		NULL		select,insert,update,references
EMAIL	varchar(50)	latin1_swedish_ci	YES		NULL		select,insert,update,references
USERNAME	varchar(40)	latin1_swedish_ci	NO		NULL		select,insert,update,references

[그림 10] HS\_admin Table

- USERID (varchar(20)) : Primary Key, 사용자(손 소독제 관리자)의 아이디
- PASSWD (varchar(20)) : 사용자가 로그인 시 사용하는 패스워드
- EMAIL (varchar(50)) : 사용자의 이메일 정보, 선택 입력 사항
- USERNAME (varchar(40)) : 사용자의 이름

## (2) 손 소독제의 위치, 잔량, 최근 업데이트 시간 등의 정보를 저장하는 HS\_list 테이블

Field	Type	Collation	Null	Key	Default	Extra	Privileges
BUILDING	varchar(20)	latin1_swedish_ci	NO		NULL		select,insert,update,references
LOCATION	varchar(20)	latin1_swedish_ci	NO		NULL		select,insert,update,references
REMAINDER	double	NULL	YES		NULL		select,insert,update,references
INIT_WEIGHT	double	NULL	YES		NULL		select,insert,update,references
ORGAN	varchar(40)	latin1_swedish_ci	YES		NULL		select,insert,update,references
CHKDATE	datetime	NULL	YES		CURRENT_TIMESTAMP		select,insert,update,references
USERID	varchar(20)	latin1_swedish_ci	NO	MUL	NULL		select,insert,update,references
chg	int(11)	NULL	YES		0		select,insert,update,references

[그림 11] HS\_list Table

- BUILDING (VARCHAR(20)) : 손 소독제가 위치한 건물명
- LOCATION (VARCHAR(20)) : 건물 내에서 손 소독제가 위치한 장소
- REMAINDER (double) : 손 소독제의 잔량(무게 데이터)
- INIT\_WEIGHT (double) : 손 소독제가 최대 잔량일 때의 무게
- ORGAN (VARCHAR(40)) : 손 소독제 위치에 해당하는 관리 기관이나 대학 명 등
- CHKDATE (datetime) : 잔량 정보를 업데이트한 시간

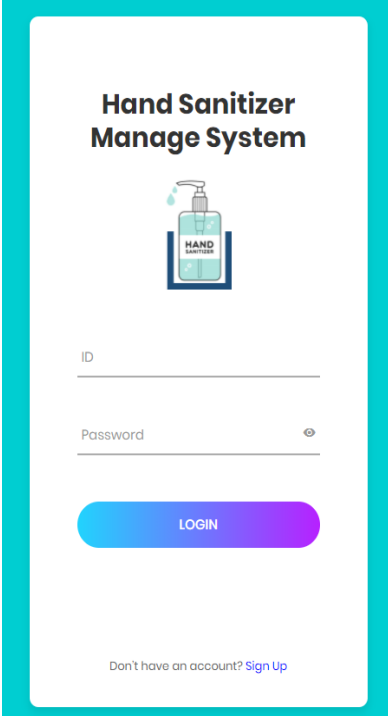
- USERID VARCHAR(20) : Foreign Key, HS\_admin 테이블의 USERID 를 참조한다

AI center	446	1414	2141	Sejong Univ	2020-12-10 12:00:00	duri	1
AI center	446	1300	2141	Sejong Univ	2020-12-10 13:00:00	duri	1
AI center	446	1100	2141	Sejong Univ	2020-12-10 14:00:00	duri	1
AI center	446	900	2141	Sejong Univ	2020-12-10 15:00:00	duri	1
AI center	446	880	2141	Sejong Univ	2020-12-10 16:00:00	duri	1
AI center	446	800	2141	Sejong Univ	2020-12-10 17:00:00	duri	1
AI center	446	600	2141	Sejong Univ	2020-12-10 18:00:00	duri	1
AI center	446	550	2141	Sejong Univ	2020-12-10 19:00:00	duri	1
AI center	446	490	2141	Sejong Univ	2020-12-10 20:00:00	duri	1
AI center	446	420	2141	Sejong Univ	2020-12-10 21:00:00	duri	1

[그림 12] HS\_list 업데이트

HS\_list 테이블 저장 값 예시로 1시간 마다 잔량 정보가 업데이트 되는 것을 확인할 수 있다.

### 6.1.3. Login



The login page features a teal border and a white background. At the top, it displays the title 'Hand Sanitizer Manage System' in bold black text, followed by an illustration of a hand sanitizer bottle. Below this, there are two input fields: 'ID' and 'Password', each with a light gray placeholder text. The 'Password' field includes a toggle icon (an eye) to the right. A prominent blue-to-purple gradient button labeled 'LOGIN' is positioned below the input fields. At the bottom, a small link reads 'Don't have an account? Sign Up'.

[그림 13] 로그인 페이지

관리자는 아이디와 비밀번호를 입력하여 로그인할 수 있다. 신규 관리자일 경우 하단 Signup 버튼을 통해 회원가입을 할 수 있다.

```

session_start();
$conn = mysqli_connect("localhost","root","system","gakgak") or die("fail");
$id=$_POST['userID'];
$password=$_POST['password'];
//아이디가 있는지 검사
$query = "SELECT * FROM HS_admin WHERE USERID='$id'";
$result = $conn->query($query);
if($result->num_rows>0){
    $row=mysqli_fetch_assoc($result);
    if($row['PASSWORD']==$password){
        $_SESSION['id']=$id;
        if(isset($_SESSION['id'])){
            echo "<script>
                alert('로그인 되었습니다.');//로그인 되었습니다를 띄우고
                //location.href='../HTML/full_list.php';
            </script>";

            Header("Location:../HTML/full_list.php");
        }
        else{
            echo "session fail"; //세션 아이디가 정의되지 않았음
        }
    }
    else{ // ID와 세트인 PASSWORD가 입력된 비밀번호와 같지 않음
        echo "<script>
            alert('아이디 혹은 비밀번호가 잘못되었습니다.');//비밀번호와 아이디가 맞지 않음
            location.href='../HTML/login.html' //이전 페이지로 돌아가기
        </script>";
    }
}
else{
    echo "접속 오류";
}

```

[그림 14] login\_action.php

관리자가 입력한 아이디와 비밀번호를 서버의 데이터베이스와 비교하여 로그인 기능을 구현하였다. 로그인이 성공적으로 진행될 시, 세션에 관리자 id가 등록된다.

#### 6.1.4. Sign Up

The image shows a 'CREATE ACCOUNT' form. It has five input fields: 'Your Name', 'Your ID', 'Password' (with a toggle icon), 'Repeat your password', and 'Your Email'. Below these fields is a blue 'SIGN UP' button. At the bottom, there is a link that says 'Have already an account ? [Login here](#)'.

[그림 15] 회원가입 페이지

관리자는 이름, 아이디, 비밀번호, 이메일을 입력하여 회원가입을 진행할 수 있다. 관리자가 계정이 있을 경우, 하단 Login here 버튼을 통해 로그인 화면으로 이동할 수 있다. 올바른 비밀번호를 입력하였는지 한 번 더 검사하는 Password Check기능과 이메일 양식이 올바른 지 검사하는 Email Check기능을 구현하였다.

```
$connect = mysqli_connect("localhost", "root", "system","gakgak") or die("fl");
$userid=$_POST['userID'];
$password=$_POST['password'];
$email=$_POST['email'];
$username=$_POST['username'];

$query = "INSERT INTO HS_admin values ('$userid','$password','$email','$username')";
$result = $connect->query($query);
if($result) {
    echo ' <script>
        alert("가입 되었습니다.");
        location.href="../HTML/login.html";
    </script>';
}
else{
    echo ' <script>
        alert("fail");
        location.href="../HTML/register.html";
    </script>';
}

mysqli_close($connect);
```

[그림 16] register\_action.php

서버의 데이터베이스에 접근하여 사용자가 입력한 아이디, 비밀번호, 이메일, 사용자의 이름 값을 추가한다. 성공적으로 회원가입이 완료될 시엔 로그인 화면으로 이동하고, 오류가 발생했을 경우, 회원가입 화면 창이 새로고침 된다.

### 6.1.5. Hand Sanitizer List



#### Hand Sanitizer List

BUILDING	LOCATION	REMAINDER	ORGANIZATION	CHKDATE	DASHBOARD
Chung Mu	207	▲18.6%	Sejong Univ	2020-11-29 12:00:00	📊
Gun Ja	510	▲9.3%	Sejong Univ	2020-11-29 12:00:00	📊
Gun Ja	room A	20.46%	Sejong Univ	2020-11-29 12:00:00	📊
Guang Gae To	1119	▲19.99%	Sejong Univ	2020-11-29 12:00:00	📊
Guang Gae To	210	20.46%	Sejong Univ	2020-11-29 12:00:00	📊
AI center	200	20.92%	Sejong Univ	2020-11-30 10:00:00	📊
AI center	200	20.92%	Sejong Univ	2020-11-30 10:00:00	📊
AI center	1010	▲4.65%	Sejong Univ	2020-11-29 07:00:00	📊

[그림 17] Hand Sanitizer List

관리자는 앞선 회원가입과 로그인 과정을 거쳐 해당 서비스에 접속함으로써 스스로에게 할당되어 있는 손소독제를 확인하게 된다.



[그림 18] User ID

해당 웹페이지의 왼쪽 상단을 확인하면 위와 같이 로그인한 관리자의 ID를 확인할 수 있다.



```

<div class="wrapper">
  <div class="logo-container">
    <div class="logo">
      
    </div>
    <div class="brand">
      Login:
      <?php
        session_start();
        echo $_SESSION['id'];
        $sess_id = $_SESSION['id'];
      <?>
    </div>
  </div>
</div>

```

[그림 19] full\_list.php

관리자의 ID는 앞서 로그인 성공 직후 유지된 세션에 등록된 ID를 불러옴으로써 위와 같이 표기할 수 있었다.

세션 ID는 어떤 관리자가 로그인 했는지를 표시하는 데에 사용되었을 뿐만 아니라, 데이터베이스로부터 관리자가 담당하고 있는 손소독제들을 불러들이는 데에도 사용되었다. 데이터베이스로부터 불러들인 손소독제 데이터는 위와 같이 리스트화하여 사용자가 한 눈에 현황을 파악할 수 있도록 구현하였다.

```

<table id="fresh-table" class="table">
  <thead>
    <th data-field="name" data-sortable="true">building</th>
    <th data-field="country" data-sortable="true">location</th>
    <th data-field="salary" data-sortable="true">remainder</th>
    <th data-field="city">Organization</th>
    <th data-field="time">CHKDATE</th>
    <th data-field="actions" data-formatter="operateFormatter" data-events="operateEvents">Dashboard</th>
  </thead>
  <?php
    $connect = mysqli_connect("localhost","root","system","gakgak") or die("fail");
    //query = "SELECT * FROM HS_list WHERE USERID='$sess_id' ORDER BY CHKDATE DESC";
    $query = "select * from HS_list where (LOCATION, CHKDATE) in (select LOCATION, max(CHKDATE) as date_time from HS_list where userid='$sess_id' group by LOCATION order by date_time desc)";
    $result = $connect->query($query);
    echo "<tbody>";
    while ($row = mysqli_fetch_array($result)){
      $remain = $row['REMAINDER'] / $row['INIT_WEIGHT'] * 100;
      echo "<tr>";
      echo "<td>".$row['BUILDING']. "</td>";
      echo "<td>".$row['LOCATION']. "</td>";
      //echo "<td>".$row['REMAINDER']. "</td>";
      if ($remain < 20)
        echo "<td><i class='fa fa-exclamation-triangle' style='color:red'></i>".round($remain,2). "</td>";
      else
        echo "<td>".round($remain,2). "</td>";
      echo "<td>".$row['ORGAN']. "</td>";
      echo "<td>".$row['CHKDATE']. "</td>";
      echo "<td></td>";
      echo "</tr>";
    }
    echo "</tbody>";
  <?>
</table>

```

[그림 20] full\_list.php

데이터는 화면에서 확인 가능하시피 테이블 형태로 불러들였고, 앞서 지정한 DB의 HS\_list 테이블의 규칙에 따라 BUILDING, LOCATION, REMAINDER, ORGANIZATION, CHKDATE를 확인할 수 있게 하였다.

```

select * from HS_list where (LOCATION, CHKDATE) in (select LOCATION, max(CHKDATE) as date_time
from HS_list where userid='$sess_id' group by LOCATION order by date_time desc);

```

위의 쿼리문으로 관리자가 관리하는 손소독제에 대한 정보들을 불러들이도록 하였다.

## (1) REMAINDER 데이터

BUILDING, LOCATION, ORGANIZATION, CHKDATE 데이터의 경우에는 모두 데이터베이스에 저장되어 있는 데이터 그대로를 불러왔다는 점에서 이들이 동일하다고 볼 수 있다. 하지만 손소독제의 퍼센티지를 나타내는 REMAINDER 데이터의 경우, 데이터베이스에 저장되어 있는 REMAINDER 데이터를 그대로 가져다 쓰게 되면 서비스가 사용자에게 제공하고자 하는 목표를 달성하지 못할 가능성이 크다.



```
$remain = $row['REMAINDER'] / $row['INIT_WEIGHT'] * 100;
```

따라서 위와 같이 REMAINDER 데이터를 INIT\_WEIGHT 데이터로 나눈 값에 100을 곱함으로써 전체 손소독제 용량 중 실제 채워져 있는 양이 얼마나 되는지 상대적인 데이터를 나타낼 수 있도록 하였다.

```
if ($remain <=20)
    echo "<td><i class='fa fa-exclamation-triangle' style='color:red'></i>".round($remain,2)."%</td>";
else
    echo "<td>".round($remain,2)."%</td>";
```

[그림 21] full\_list.php

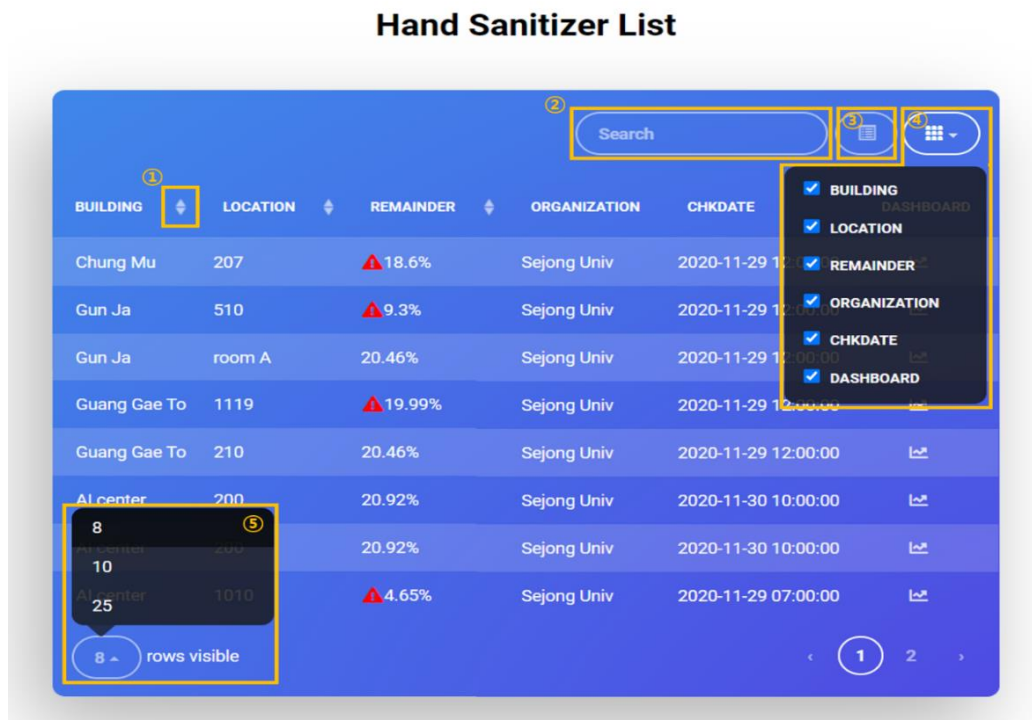
이렇게 계산한 값을 기반으로 만약 잔량이 20%이하일 경우, 빨간색 경고 표시와 함께 데이터를 나타내도록 하였고, 그렇지 않을 경우에는 퍼센트 데이터만을 출력하도록 하였다. 모든 퍼센트 데이터는 소수점 둘째 자리까지만 표기하였다.

BUILDING	LOCATION	REMAINDER	ORGANIZATION	CHKDATE	DASHBOARD
Chung Mu	207	 18.6%	Sejong Univ	2020-11-29 12:00:00	

[그림 22] REMAINDER 데이터 표시

그 결과 REMAINDER 데이터가 나타내는 결과는 위와 같다.

## (2) Hand Sanitizer List 기능



[그림 23] Hand Sanitizer List 기능

사용자가 관리하는 손소독제 리스트를 보여주는 이 페이지는 데이터를 보여주는 방법에 따라 다양한 기능들을 제공하고 있다.

```
$(function () {
    $table.bootstrapTable({
        classes: 'table table-hover table-striped',
        toolbar: '#toolbar',

        search: true,
        //showRefresh: true,
        showToggle: true,
        showColumns: true,
        pagination: true,
        striped: true,
        sortable: true,
        pageSize: 8,
        pageList: [8, 10, 25, 50, 100],

        formatShowingRows: function (pageFrom, pageTo, totalRows) {
            return ''
        },
        formatRecordsPerPage: function (pageNumber) {
            return pageNumber + ' rows visible'
        }
    })

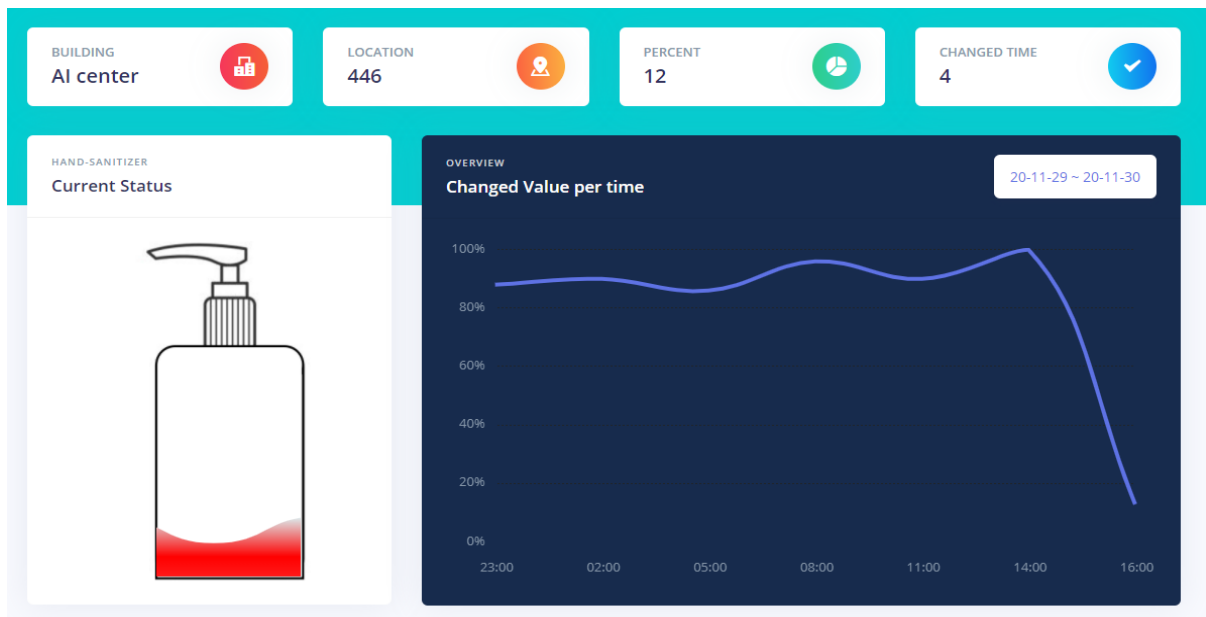
    $alertBtn.click(function () {
        alert('You pressed on Alert')
    })
})
```

[그림 24] full\_list.php

①에서 ⑤까지의 기능들은 모두 부트스트랩의 테이블을 사용하여 기능을 구현하였다.

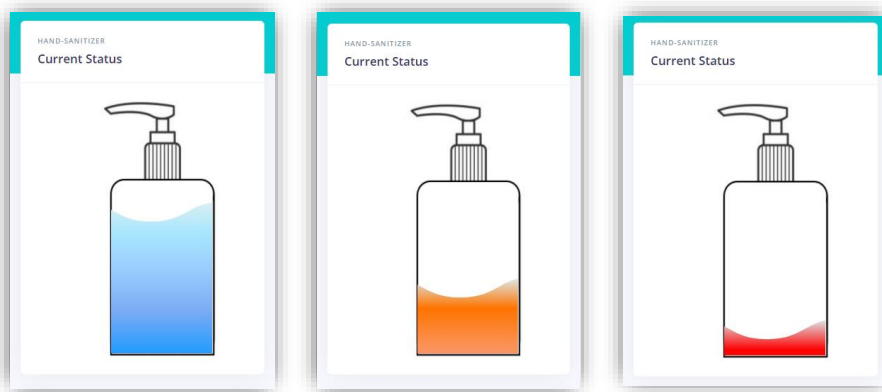
- **기능 ①**은 화살표를 클릭함에 따라 오름차순 혹은 내림차순으로 데이터를 볼 수 있도록 하여 관리자가 보고 싶은 데이터들을 필드마다 모아서 볼 수 있도록 기능을 구현하였다.
- **기능 ②**는 Search 기능으로, 해당 테이블에서 볼 수 있는 모든 데이터들 중 원하는 데이터를 빠르게 찾고자 할 때 사용자가 검색할 수 있도록 해당 기능을 구현하였다.
- **기능 ③**은 데이터를 리스트 형태 혹은 카드 뷰 형태, 두 가지 형태로 볼 수 있도록 기능을 구현하였다. 두 형태 모두 보여주는 데이터는 동일하지만, 이 서비스의 취지와 사용자의 사용 목적을 생각해본다면 default인 리스트 형태로 데이터를 확인하는 것이 더욱 적합할 것으로 예상된다.
- **기능 ④**는 보여지는 column을 체크하는 기능으로, 해당 리스트에서 보고 싶은 column과 보고싶지 않은 column을 사용자가 직접 선택함으로써 데이터를 선별적으로 확인할 수 있도록 구현하였다.
- **기능 ⑤**는 볼 수 있는 줄 수를 8줄, 10줄, 25줄 중 하나를 택함으로써 한 화면에 얼마만큼 많은 데이터를 볼 수 있게 할 것인지 사용자가 선택 가능하도록 구현하였다.

#### 6.1.6. DashBoard



[그림 25] DashBoard

관리자가 손소독제의 다양한 정보를 시각적으로 확인할 수 있는 DASHBOARD 화면이다.



[그림 26] 잔여량별 손소독제 시각화

잔여량별 손 소독제의 높이와 색깔을 다르게 표현하여 관리자가 시각적으로 손소독제 잔량을 빠르게 파악할 수 있다.

```
<svg class="sea" style="transform: translate(-240px);" version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" width="143px" height="340px" viewBox="30 30 170 360" style="enable-background: new 0 0 800 400;"> <!-- # -90~ 190-->

<style>
<![CDATA[
.st3{fill:url(#SVGID_3);}
]]>
</style>

<g id="LightWaves" class="LightWaves">
<g>
<linearGradient id="SVGID_3_" gradientUnits="userSpaceOnUse" x1="400.0005" y1="600" x2="400.0005" y2="214.3545">
<?php
if($per>70){
echo '
<stop offset="0" style="stop-color:#008cff"/>
<stop offset="0.1643" style="stop-color:#008cff"/>
<stop offset="0.3574" style="stop-color:#209bff"/>
<stop offset="0.5431" style="stop-color:#7cacf5"/>
<stop offset="0.7168" style="stop-color:#90ceff"/>
<stop offset="0.874" style="stop-color:#a9e7ff"/>
<stop offset="1" style="stop-color:#dceff1"/>;
}
else if(20<$per){
echo '
<stop offset="0" style="stop-color:#008cff"/>
<stop offset="0.1643" style="stop-color:#008cff"/>
<stop offset="0.3574" style="stop-color:#f3c78e"/>
<stop offset="0.5431" style="stop-color:#fc8f8f"/>
<stop offset="0.7168" style="stop-color:#fa945a"/>
<stop offset="0.874" style="stop-color:#ff7300"/>
<stop offset="1" style="stop-color:#dceff1"/>;
}
else{
echo '
<stop offset="0" style="stop-color:#008cff"/>
<stop offset="0.1643" style="stop-color:#008cff"/>
<stop offset="0.3574" style="stop-color:#fd7575"/>
<stop offset="0.5431" style="stop-color:#fc8f8f"/>
<stop offset="0.7168" style="stop-color:#ff4242"/>
<stop offset="0.874" style="stop-color:#ff0000"/>
<stop offset="1" style="stop-color:#dceff1"/>;
}
?>
</linearGradient>
<path class="st3" d="M750,9,229.8c-14.8,7.9,28.7,15.4,57.2,15.4c-28.5,0,42.4,7.5,57.2,15.4c-15.2,8.2,30.9,16.6,62.1,16.6
s-46.9,8.4,62.1,16.6c-14.8,7.9,28.7,15.4,57.2,15.4c-28.5,0,42.4,7.5,57.2,15.4c-15.2,8.2,30.9,16.6,62.1,16.6
c-31.2,0,46.9,8.4,62.1,16.6c-14.8,7.9,28.7,15.4,57.2,15.4c-28.5,0,42.4,7.5,57.2,15.4c-15.2,8.2,30.9,16.6,62.1,16.6
c-31.2,0,46.9,8.4,62.1,16.6c-14.8,7.9,28.7,15.4,57.2,15.4c-28.5,0,42.4,7.5,57.2,15.4c-15.2,8.2,30.9,16.6,62.1,16.6
C782.1,246.4,766.1,237.9,750,229.8z"/>
</g>
</g>
</svg>
```

[그림 27] dash.php

손소독제 잔여량에 맞추어 높낮이를 조절하였다. 손소독제 잔여량이 70% 이상일 땐 파란색, 20% 이상일땐 주황색, 20% 미만일 때에는 빨간색으로 표시하였다.

```

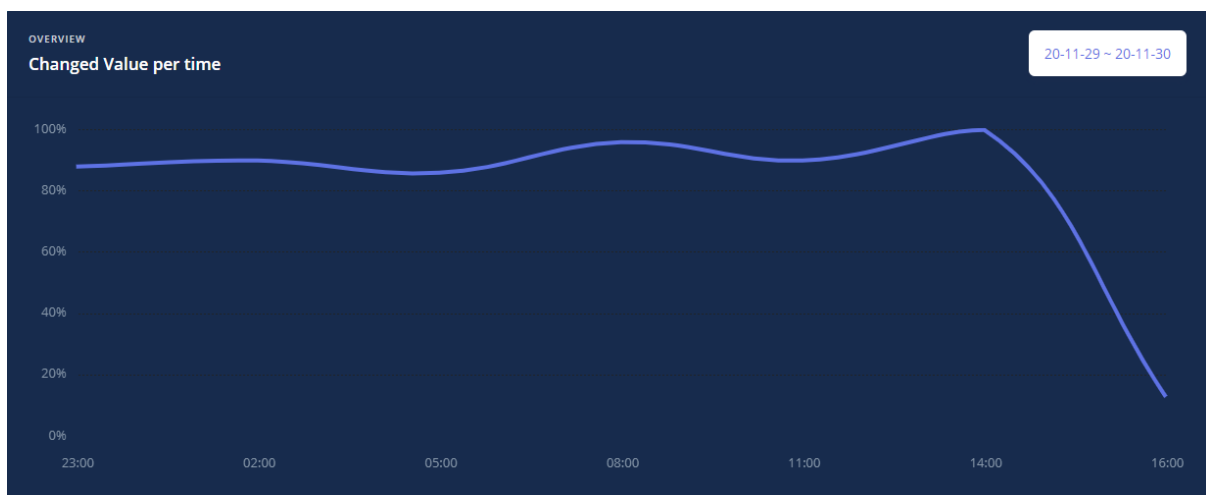
.LightWaves {
  animation: lightwaves 4s infinite;
  position: relative;
}

@keyframes lightwaves {
  0%,100% { transform: translate(0,0); }
  25%    { transform: translate(5px,5px); transform: scale(1.05); }
  50%    { transform: translate(25px, 5px); }
  75%    { transform: translate(12px,10px); transform: scale(1.05); }
}

```

[그림 28] dash.php

손소독제를 svg 를 이용하여 표현하였고 출렁임을 표현하여 잔여량에 대해 보다 쉽게 파악할 수 있도록 구현하였다.



[그림 29] 잔여량별 그래프

최신 18 시간의 데이터를 가져와 3 시간 간격으로 출력한다. X 축은 시간, Y 축은 잔여량을 나타내며 오른쪽 상단 하얀색 박스는 데이터의 날짜를 나타낸다. 그래프의 상승 혹은 하락의 정도를 통해 소모속도를 시작적으로 파악할 수 있다.

## 6.1.7. Data Update

```
url = "http://203.250.148.89:7579/Mobius/Team_1/Weight/la"
sys.stdout = open('./weight.txt', 'w')
headers = {
    'Accept': "application/json",
    'X-M2M-RI': "12345",
    'X-M2M-Origin': "Duribeonx2",
}

response = requests.request("GET", url, headers=headers)
```

[그림 30] get\_data.py

```
f=open('weight.txt','r')
x=f.read()
res=eval(x)
input_data=res['m2m:cin']['con']
sys.stdout = open('./weight2.txt', 'w')
remainder=int(input_data)
conn = pymysql.connect(host='127.0.0.1',user='duri',password='gakgak',charset='utf8')
curs = conn.cursor()
sql = "USE gakgak;"
curs.execute(sql)
sql = "SELECT * FROM HS_list WHERE LOCATION=446 ORDER BY CHKDATE DESC LIMIT 1;"
curs.execute(sql)
result = curs.fetchall()
for row in result:
    p_remainder = row[2]
    p_init_weight = row[3]
    p_chg = row[7]
    if (remainder < p_remainder+1000):
        chg = p_chg
        init_weight = p_init_weight
    else:
        chg = p_chg +1
        init_weight = remainder

sql = 'INSERT INTO HS_list (BUILDING, LOCATION, REMAINDER,INIT_WEIGHT,ORGAN,CHKDATE, USERID,chg) VALUES ("AI center","446", %s,%s,
"Sejong Univ",%s,"duri",%s);'
now = datetime.datetime.now();
curs.execute(sql,(remainder,init_weight,now,chg))
conn.commit()
conn.close()
```

[그림 31] final.py

IoT 플랫폼으로부터 최근 손소독제 잔여량을 받아온 뒤, 서버 데이터베이스에 업데이트한다. 이때, 최근 손소독제 잔여량이 이전의 손소독제 잔여량보다 300g 이상 증가할 경우, 손소독제를 새로 교체하였다고 판단하여 교체횟수를 증가하고, 새로 바꾼 손소독제가 가득할 경우의 무게를 추가적으로 업데이트한다.