

- Wireless-network project •

Smart Aquarium

Index

- ▶ 주제
- ▶ 라즈베리파이
- ▶ 아두이노
- ▶ 안드로이드 앱 개발
- ▶ 정리

주제

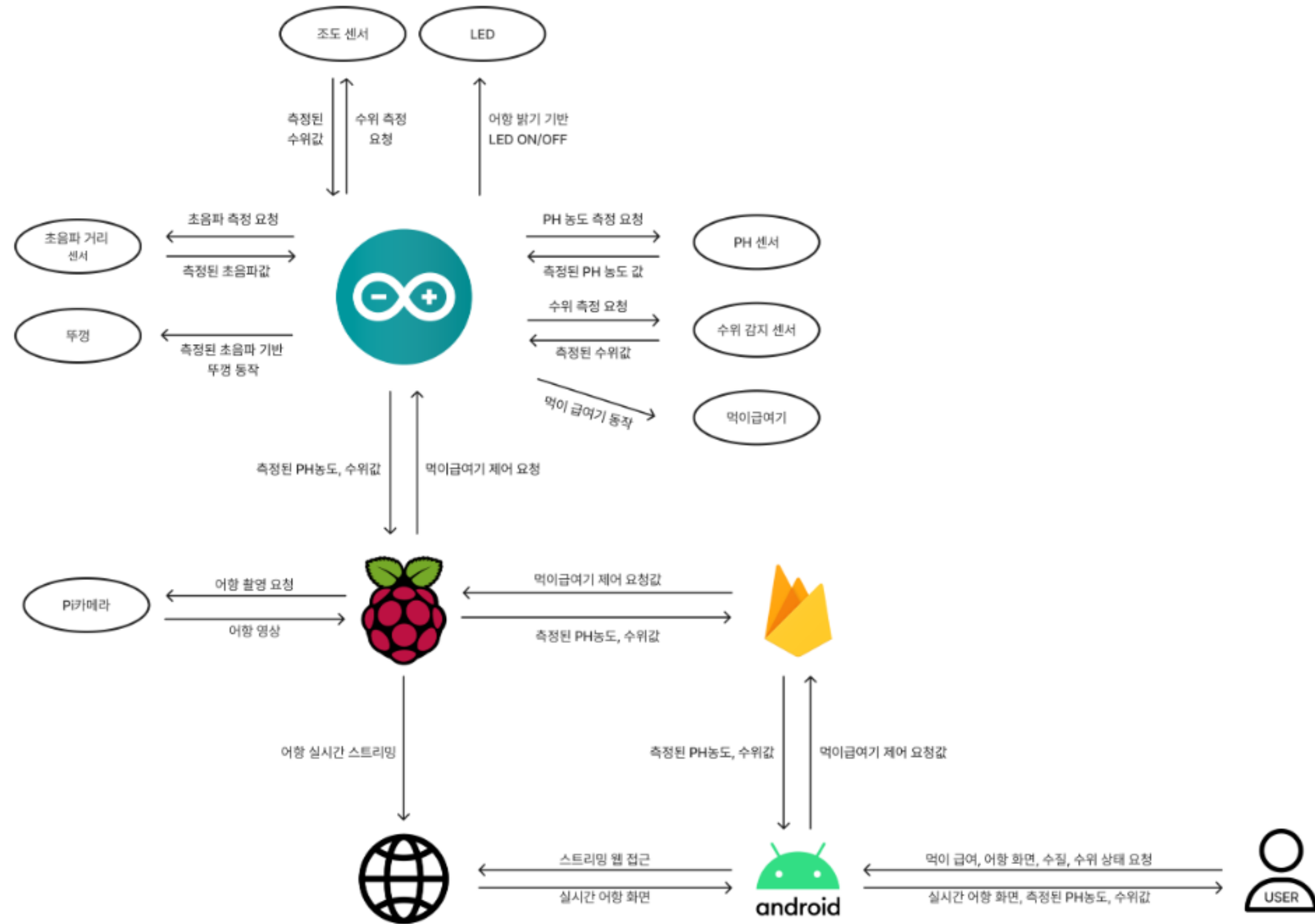
주제

1. 기술과 자연의 결합: 어항은 자연과 생물을 관찰하고 즐기는 공간이지만, 기술을 통해 더 나은 환경을 조성할 수 있습니다. 이번 프로젝트는 물고기의 건강과 행복을 돕는 동시에, 기술을 활용해 이를 관리하고 모니터링합니다

2. IoT 및 스마트 기기의 활용: IoT 기술을 어항에 적용한다는 것은 실제로 우리 주변에 있는 물고기를 돌보는 데 더 많은 편의성과 혁신을 가져다줄 수 있습니다. 수위, 수질, 먹이 공급 등을 원격으로 제어할 수 있다는 점에서 기존과는 다른 기술력을 보여줄 수 있을 것입니다.



시스템 흐름도



라즈베리파이

파이어베이스 - 라즈베리파이 연동

```
1  import serial
2  import firebase_admin
3  from firebase_admin import credentials, db
4
5  # Firebase 초기화
6  cred = credentials.Certificate("/home/pi/project2/firebase/smartbowl-b05d9-firebase-adminsdk-ldgsu-b9626cd4c3.json")
7  firebase_admin.initialize_app(cred, {'databaseURL': 'https://smartbowl-b05d9-default-rtdb.firebaseio.com/'})
8  ref = db.reference('/data')
9
10 # 시리얼 포트 설정
11 ser = serial.Serial('/dev/ttyACM0', 9600)
12
```

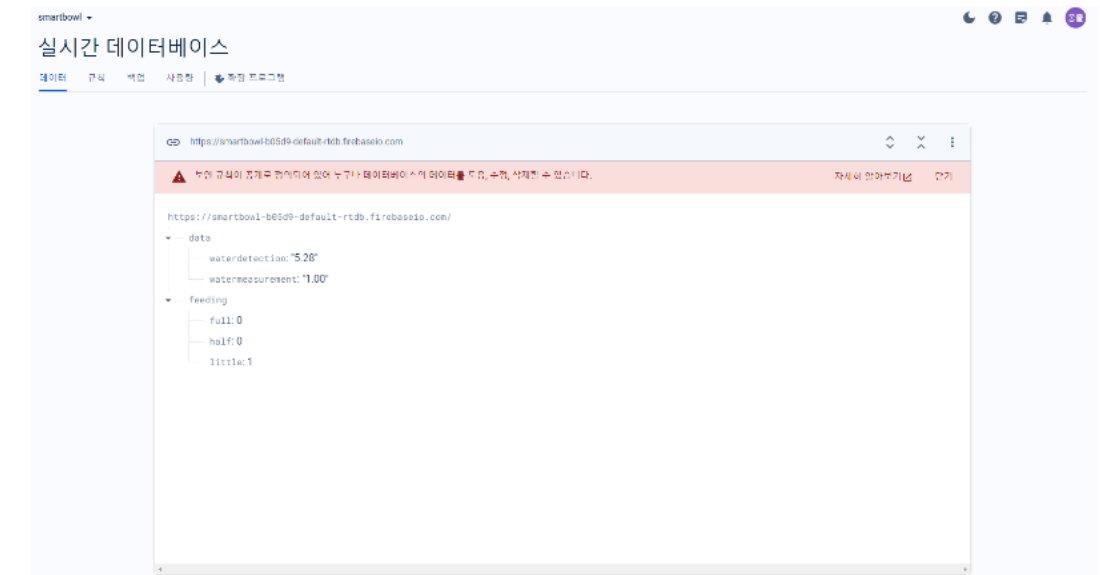
통합된 아두이노와 라즈베리 파이 연결

```
131
132     String pHValue1 = String(pHValue);
133     String LevelValue = String(value2);
134
135     Serial.print(pHValue1 + ",");
136     Serial.println(LevelValue);
137
138     delay(1000); // 1초 동안 대기
139
140
```

[아두이노에서 보내기]

```
10 # 시리얼 포트 설정
11 ser = serial.Serial('/dev/ttyACM0', 9600)
12
13 while True:
14     # 아두이노에서 값 읽어오기
15     arduino_data = ser.readline().decode('utf-8').strip()
16
17     values = arduino_data.split(',')
18     print(values)
19
20     # pH 값과 수위 값 추출
21     ph_value = values[0]
22     water_level = values[1]
23
24     # Firebase에 데이터 업로드
25     data = {
26         'waterdetection': ph_value,
27         'watermeasurement': water_level
28     }
29
30     ref.set(data)
31
```

[라즈베리파이에서 받아오고]



[파이어베이스에 저장]

실시간 수질, 수위 값

🔥 Firebase

🏠 프로젝트 개요

⚙️

프로젝트 바로가기

📺 Realtime Database

제품 카테고리

빌드

출시 및 모니터링

애널리틱스

참여

모든 제품

탐색 메뉴 맞춤설정

이제 탐색 메뉴를 맞춤설정하여 콘솔 환경에 집중할 수 있습니다.

Spark업그레이드

smartbowl

실시간 데이터베이스

데이터규칙백업사용량확장 프로그램

결제 사기나 피싱과 같은 악용으로부터 Realtime Database 리소스를 보호하세요. 앱 체크 구성

https://smartbowl-b05d9-default-rtdb.firebaseio.com

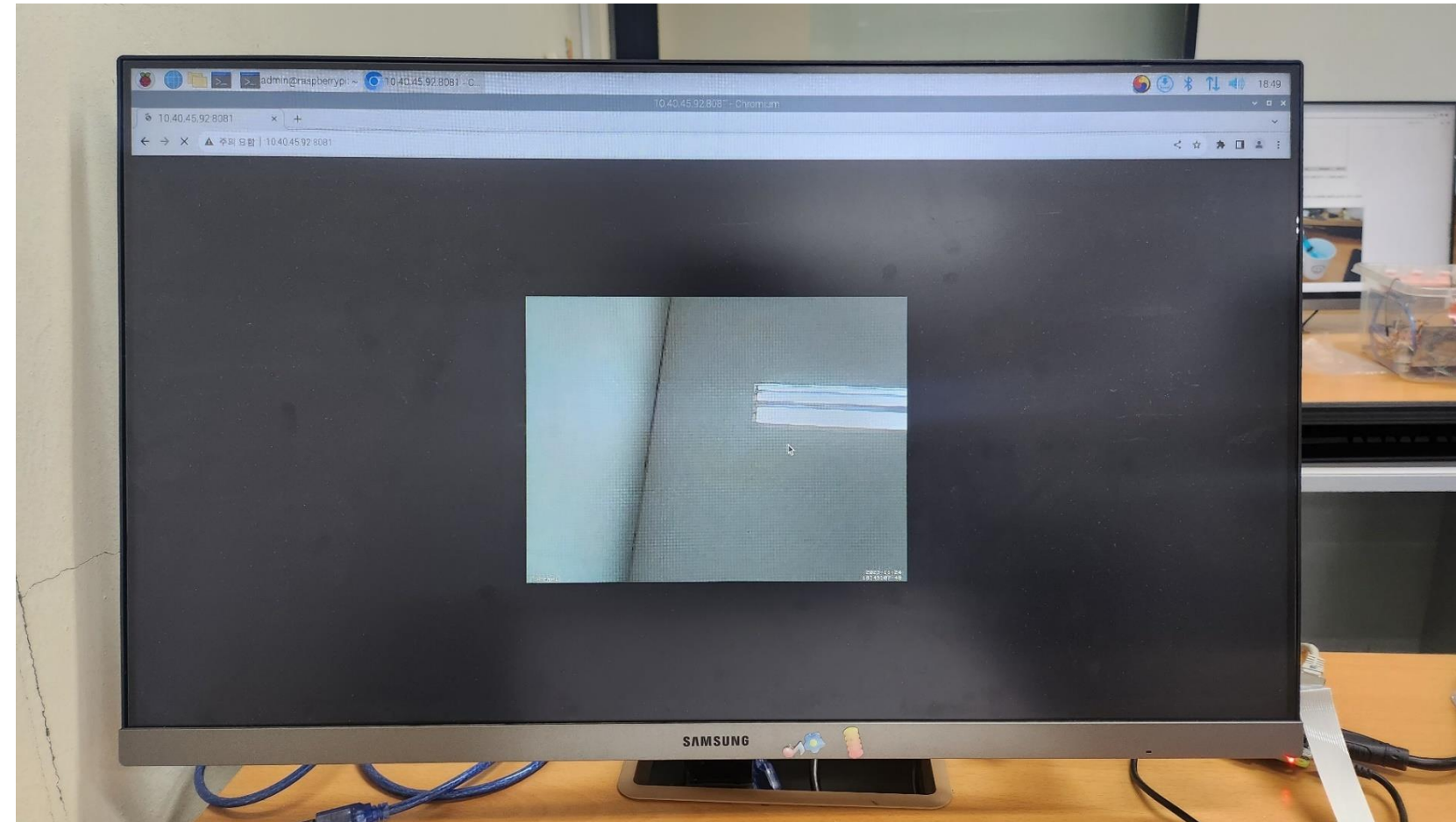
⚠️ 보안 규칙이 공개로 정의되어 있어 누구나 데이터베이스의 데이터를 도용, 수정, 삭제할 수 있습니다. 자세히 알아보기닫기

https://smartbowl-b05d9-default-rtdb.firebaseio.com/

- data
 - waterdetection: "3.85"
 - watermeasurement: "224"
- feeding

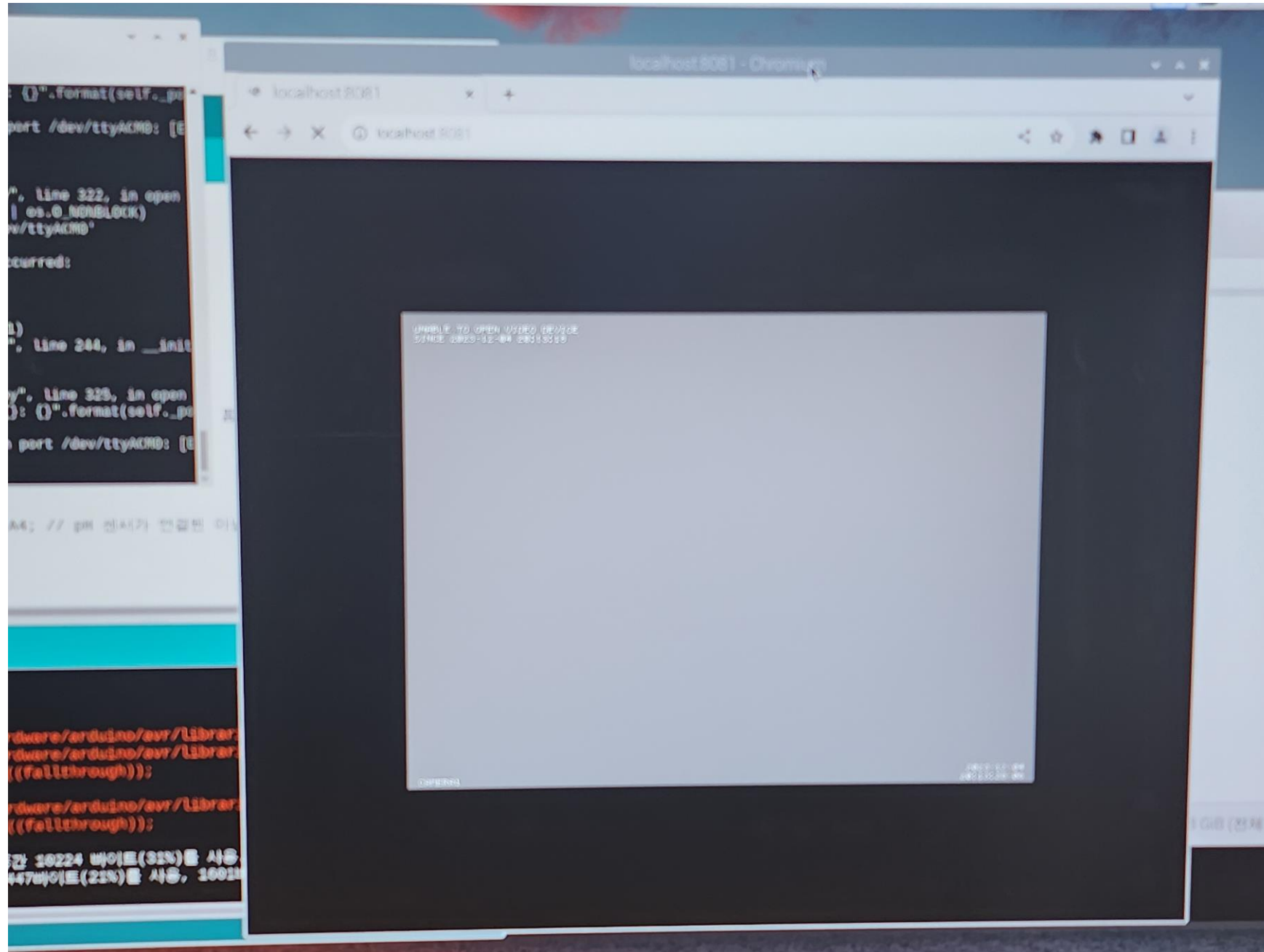
실시간 수조 스트리밍 (motion - Android)

- 라즈베리파이에 파이카메라를 연결하여 모션을 통해 웹 서버에서 스트리밍 영상 출력
- 출력된 영상은 안드로이드 앱을 통해 웹 서버에 접속하여 사용자에게 실시간으로 촬영 중인 카메라 화면을 보여줌



(첫 테스트 당시 정상 작동 확인)

실시간 수조 스트리밍 (motion - Android)



(두 번째 테스트 카메라 오류 발견)

```
user@raspberrypi:~ $ raspistill -o test.jpg
mmal: mmal_vc_component_enable: failed to enable component: ENOSPC
mmal: camera component couldn't be enabled
mmal: main: Failed to create camera component
mmal: Failed to run camera app. Please check for firmware updates
```

라즈베리파이가 카메라를 인식하고 웹 서버에 보내는 것은
가능했지만
카메라로 촬영해보면 아무것도 안 뜨고 오류가 뜸
[시도해 본 오류 해결 방법]

OS 업그레이드, 업데이트 -> 같은 오류
GPU 메모리 늘리기 -> 변화 없음
카메라 센서, 라즈베리파이 새로 구운 후 다시 설치 -> 인식조차 안
되는 기기들이 너무 많음

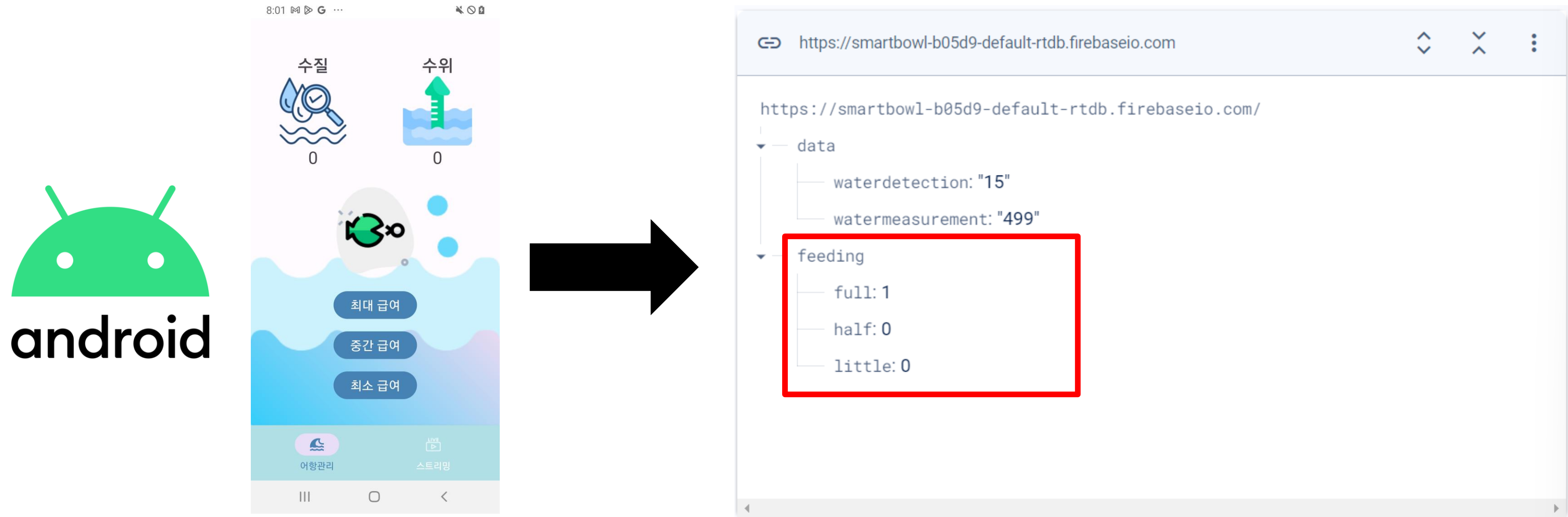
먹이 급여기 제어 (Firebase – Raspberry Pi – Arduino)

- 안드로이드 앱에서 먹이 급여기 제어 버튼을 누르면 데이터베이스에 값이 들어옴
- 들어온 값은 라즈베리파이를 통해 받고, 그 값을 아두이노로 보내 서보 모터를 제어함



(라즈베리파이 – 안드로이드 급여 요청 데이터 전달 완료)

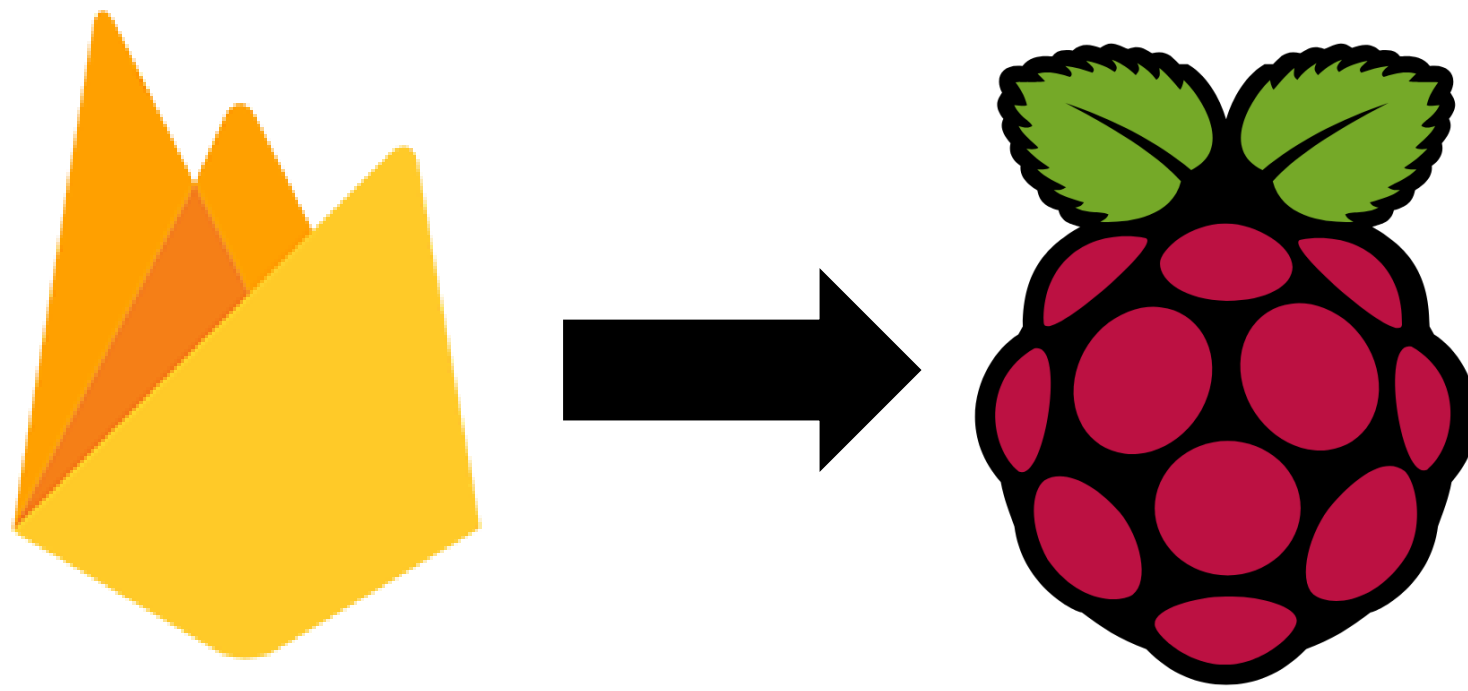
먹이 급여기 제어 (Firebase – Raspberry Pi – Arduino)



먹이 급여 버튼 (최대, 중간, 최소)

- feeding 필드에 각각 급여 양에 따라 full, half, little key에 기본값 0을 저장
- 모바일에서 사용자가 먹이 급여 제어 버튼을 누르면 해당되는 key의 값에 1이 저장됨

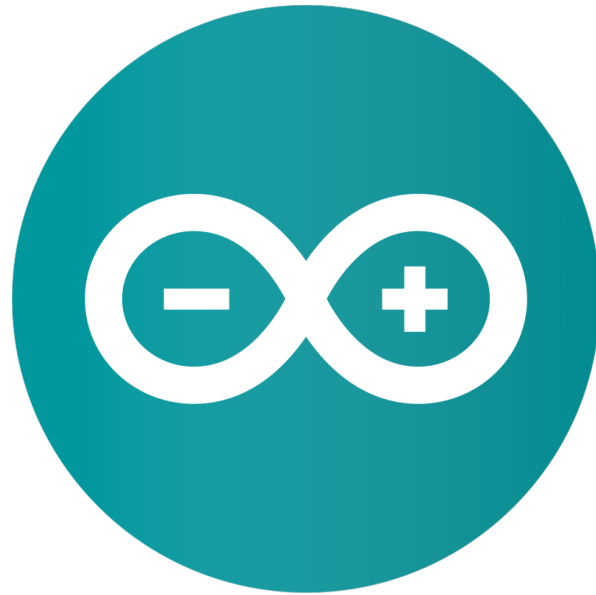
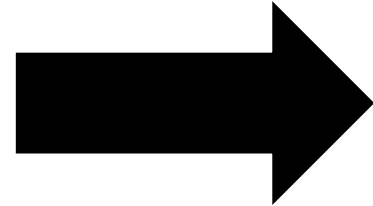
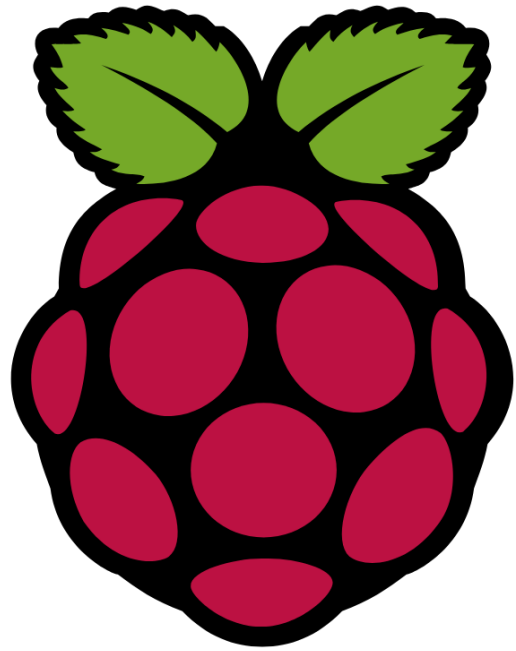
먹이 급여기 제어 (Firebase – Raspberry Pi – Arduino)



```
1  import requests
2  import time
3  import serial
4
5  firebase_url = 'https://smartbowl-b05d9.firebaseio.com/feeding.json'
6  arduino_serial_port = '/dev/ttyACM0'
7
8  ser = serial.Serial(arduino_serial_port, 9600, timeout=1)
9
10 while True:
11     try:
12         response = requests.get(firebase_url)
13         data = response.json()
14
15         if data:
16             for key, value in data.items():
17                 if value == 1:
18                     if key == "little":
19                         ser.write("L".encode('utf-8'))
20                         print("Little")
21                     elif key == "half":
22                         ser.write("H".encode('utf-8'))
23                         print("Half")
24                     elif key == "full":
25                         ser.write("F".encode('utf-8'))
26                         print("Full")
27
28                 data[key] = 0
29                 requests.patch(firebase_url, json={key: 0})
30
31     except Exception as e:
32         print(f"Error: {e}")
```

- 라즈베리파이에서 데이터베이스의 각 key값을 검사하여 1이 저장된 키를 도출함
- 도출해낸 키에 따라 해당되는 제어값을 아두이노로 전달함

실시간 수조 스트리밍 (motion - Android)

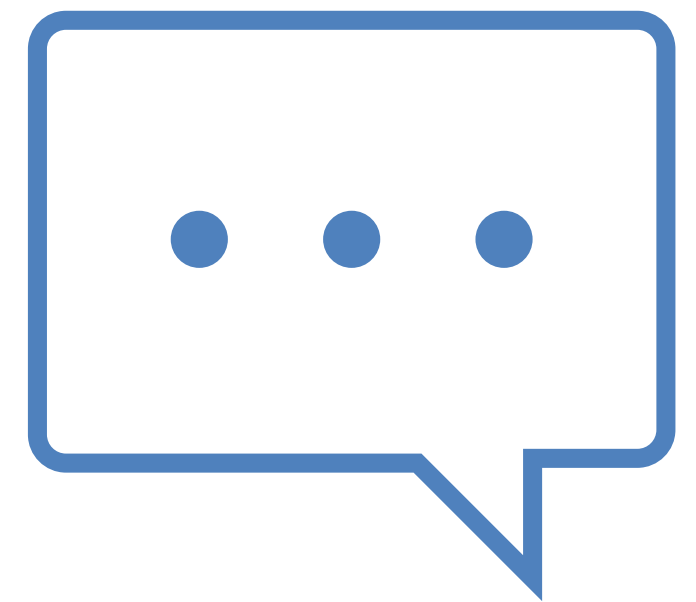


```
1 //아두이노 먹이급여기 부분 수정
2 //라즈베리파이에서 보낸 문자를 읽고 해당 레벨에 맞게 급여
3
4 if (Serial.available() > 0) {
5     char receivedChar = Serial.read();
6     if (receivedChar == 'L') {
7         Pos = 30;
8         myservo.write(Pos);
9         delay(1000);
10        Pos = 0;
11        myservo.write(Pos);
12    }
13    else if (receivedChar == 'H') {
14        Pos = 60;
15        myservo.write(Pos);
16        delay(1000);
17        Pos = 0;
18        myservo.write(Pos);
19    }
20    else if (receivedChar == 'F') {
21        Pos = 90;
22        myservo.write(Pos);
23        delay(1000);
24        Pos = 0;
25        myservo.write(Pos);
26    }
27 }
```

- 전달받은 제어값에 따라 먹이 급여기를 동작 시키는 서보모터를 제어함

회고

- 주제 선정에 어려움을 겪어 프로젝트 시작이 늦었던 점이 아쉬움
- 끝내 해결하지 못 한 카메라 오류. 시간이 더 많았으면...
- 실습 시간에 못 해봤던 부분들을 스스로 학습해볼 수 있었던 기회



아두이노

먹이 급여기, 천장 개폐, 수위 조절

먹이 급여기

```
if(digitalRead(LittlePin) == HIGH){ //버튼이 눌렸는지 확인
  Pos = 30; //서보모터 각도 지정
  myservo.write(Pos); //서보모터 제어
  delay(1000);
  Pos = 0; //서보모터 각도 초기값으로 지정
  myservo.write(Pos);
}
```

천장 개폐

```
if(distance < 15){ //거리가 15cm 미만일때
  myservo.attach(ServoPin);
  Pos = 90;
  myservo.write(Pos); // 서보모터 제어
  delay(1000);
  myservo.detach();
}
else{
  myservo.attach(ServoPin);
  Pos = 0;
  myservo.write(Pos);
  delay(1000);
  myservo.detach();
}
```

먹이 급여기, 천장 개폐, 수위 조절

수위 조절

```
if(val >= 500){  
  Serial.println("물 넘침 위험");  
  delay(5000);  
}  
else if(val <= 100) {  
  Serial.println("물 부족 위험");  
  delay(5000);  
}
```

조명제어, 수질 관리

LED 조명제어

```
void setup(){
  pinMode(4,OUTPUT);           //핀 4번을 출력으로
  Serial.begin(9600);          //시리얼통신설정
}

void loop(){

  int a = analogRead(A0);      //data변수에 조도 센서 측정값 저장

  delay(200);
  if(a>300)digitalWrite(4,LOW); //a가 300이 넘으면 4번핀에 LOW값
  else digitalWrite(4,HIGH);    //a가 300이 안넘으면 4번핀에 HIGH값

  Serial.println(a);           //측정값에 따라 나온 결과값 출력
}
```

수질 관리

```
// 아날로그 값을 전압으로 변환
voltage = (rawValue / 1024.0) * 5.0;

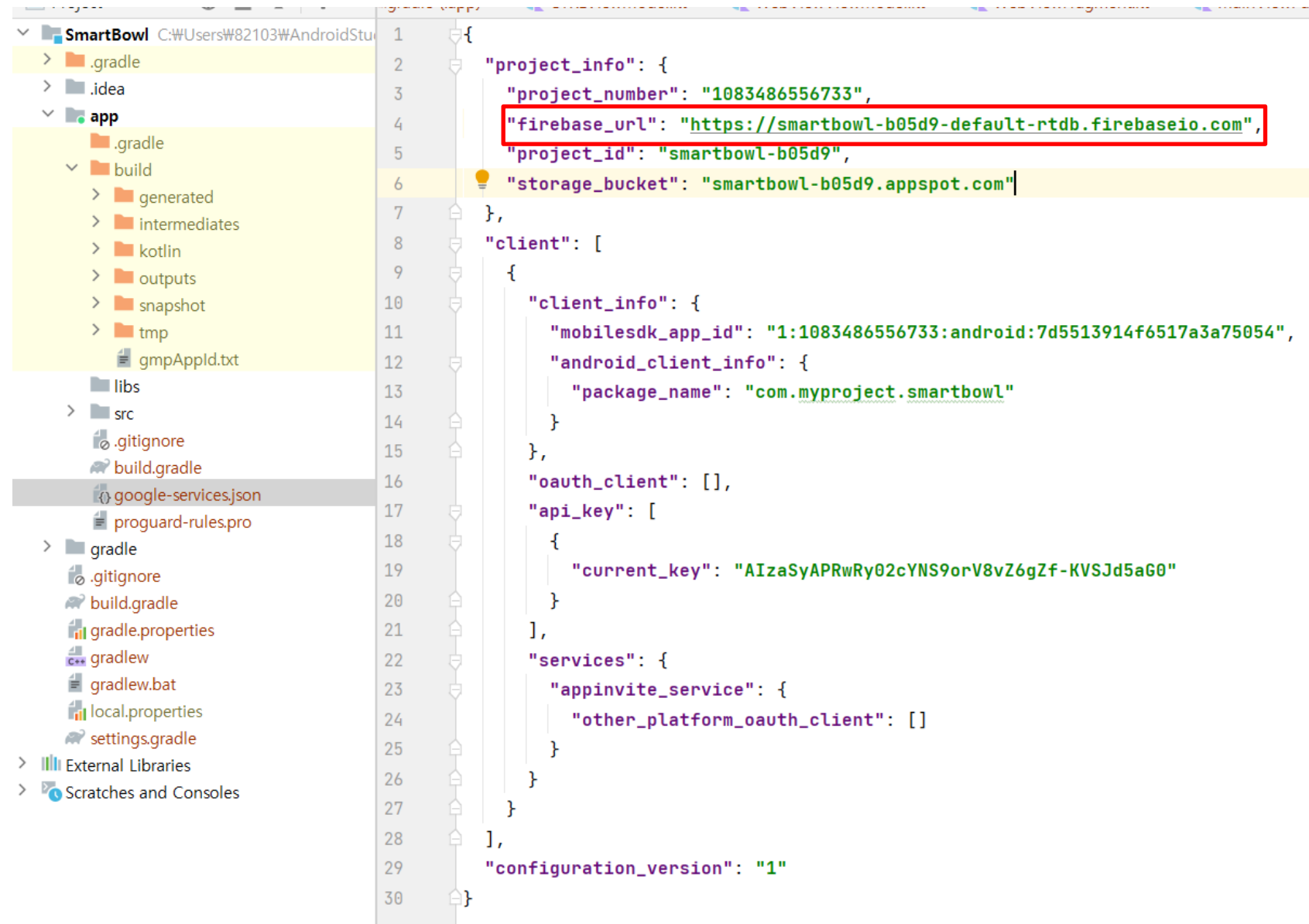
// 전압을 pH 값으로 변환 (이 부분은 pH 센서의 특정 데이터시트에 따라 달라질 수 있음)
pHValue = pHValue = 7 - (voltage - VOffset);

// 결과를 시리얼 모니터에 출력
Serial.print("Raw Value: ");
Serial.print(rawValue);
Serial.print("\tVoltage: ");
Serial.print(voltage);
Serial.print("\tpH Value: ");
Serial.println(pHValue);
```

안드로이드

안드로이드 앱 개발

안드로이드 – 파이어베이스 연동



● 파이어베이스의 정보를 담고있는 JSON 파일

● **firebase_url**: Firebase Realtime Database의 URL
이 URL을 통해 데이터베이스에 액세스할 수 있음

● **"firebase_url"**: "https://smartbowl-b05d9-default-rtdb.firebaseio.com/"

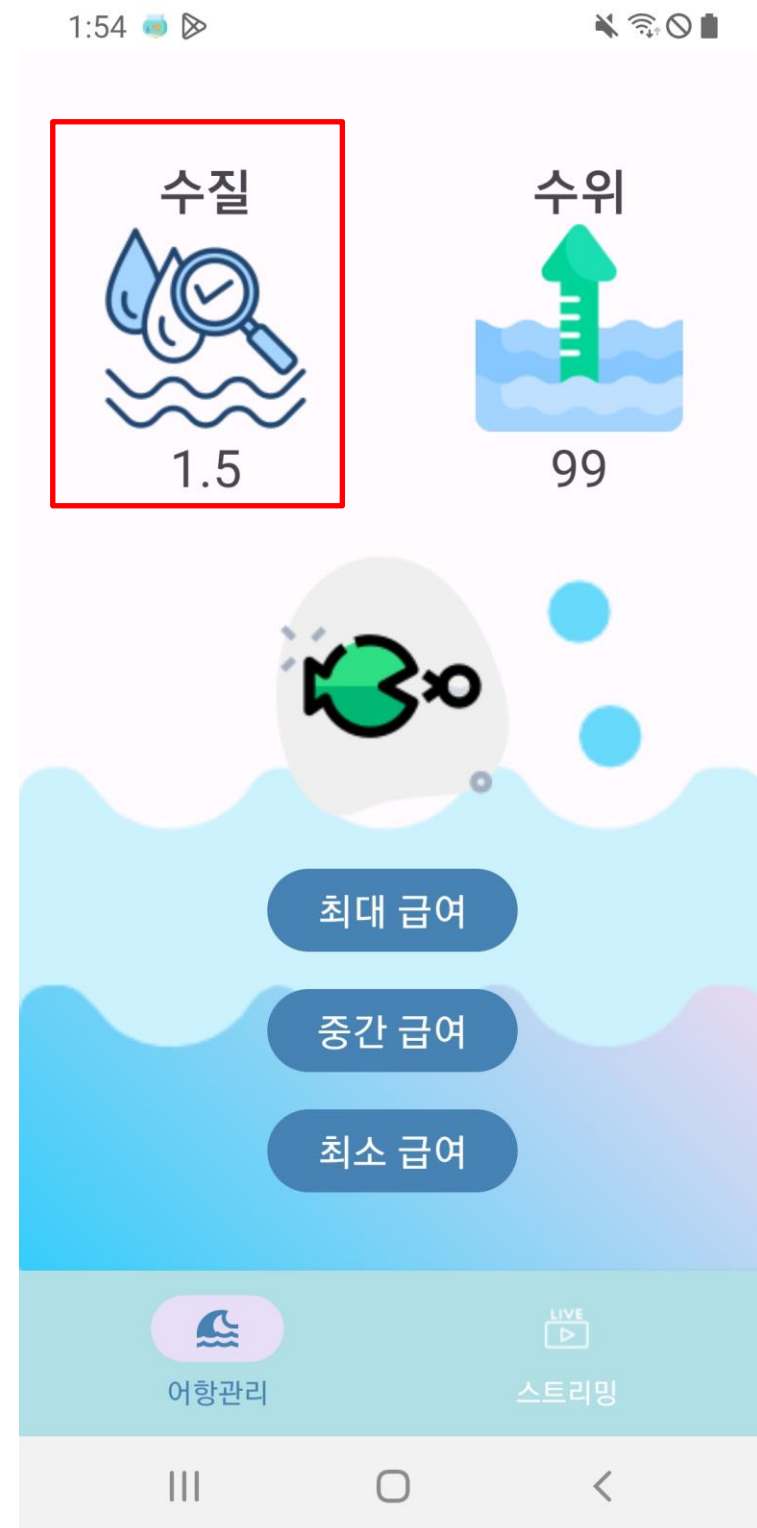
안드로이드 앱 개발

안드로이드 - 파이어베이스 연동

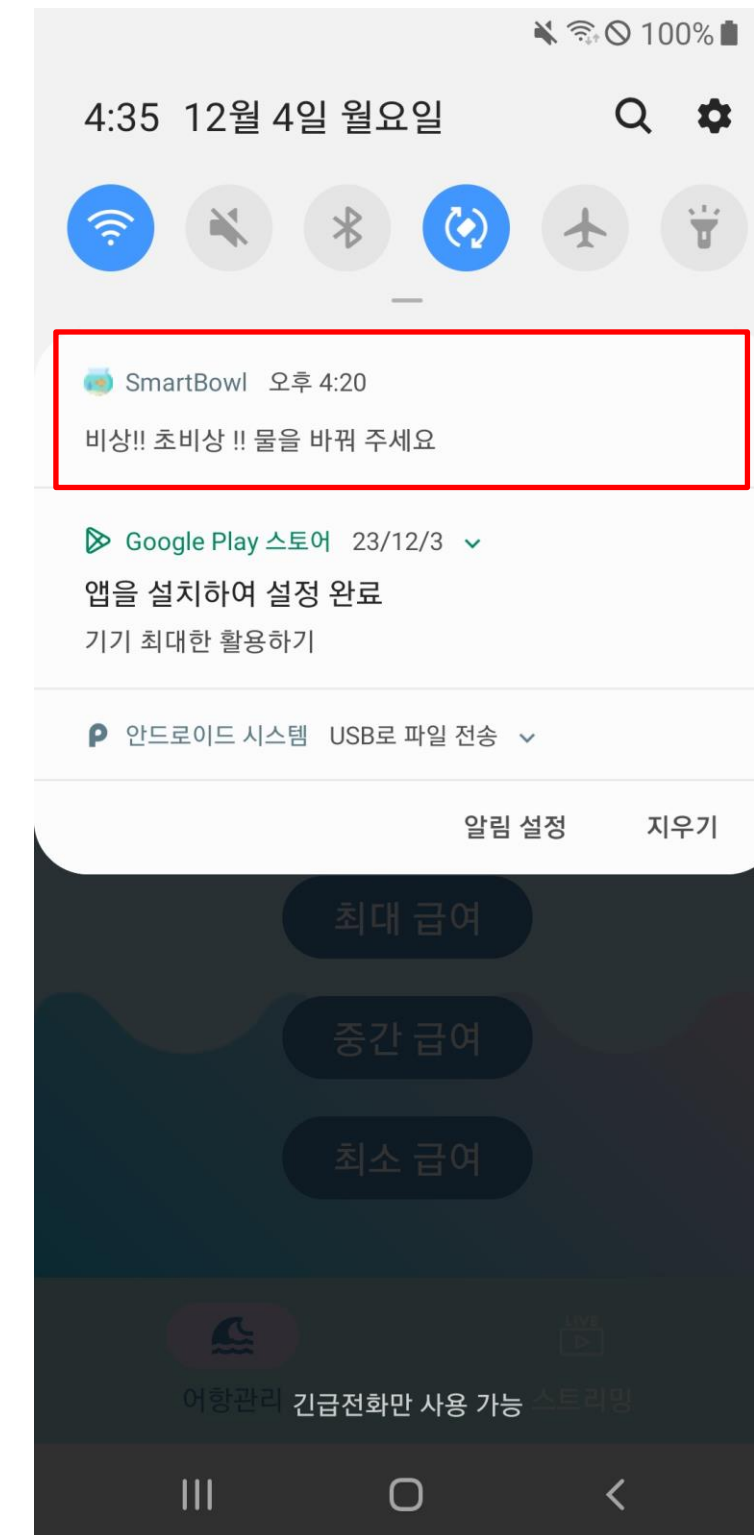
```
//FireBase  
implementation platform('com.google.firebase:firebase-bom:32.0.0')  
implementation 'com.google.firebase:firebase-database-ktx:20.3.0'  
implementation 'com.google.firebase:firebase-firestore-ktx:24.9.1'  
implementation 'com.google.firebase:firebase-storage-ktx:20.3.0'  
implementation 'com.firebaseui:firebase-ui-storage:8.0.1'
```

- 라이브러리나 모듈을 프로젝트에 추가하는 코드

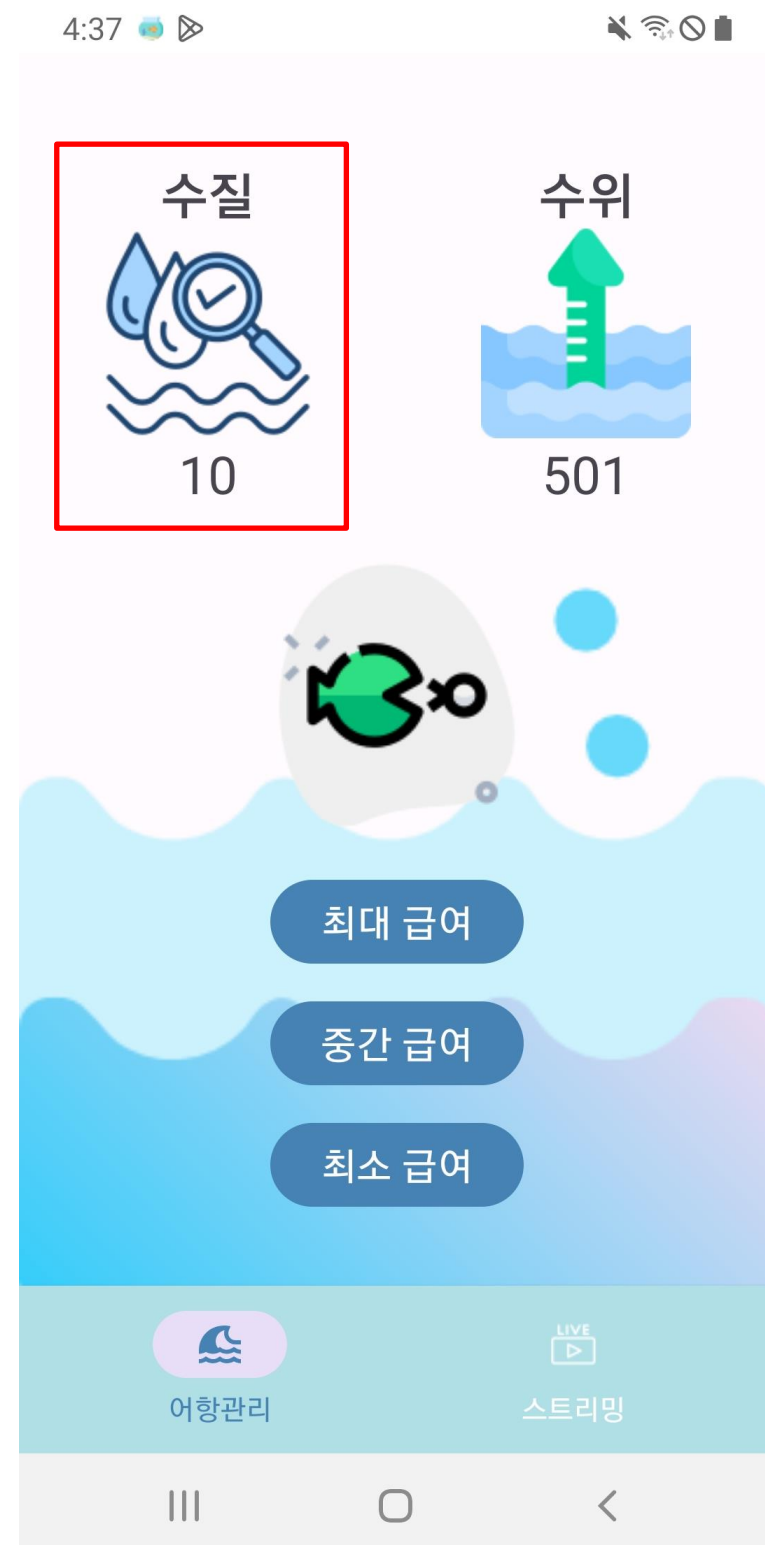
안드로이드 앱 개발 - 팝업 알람(수질)



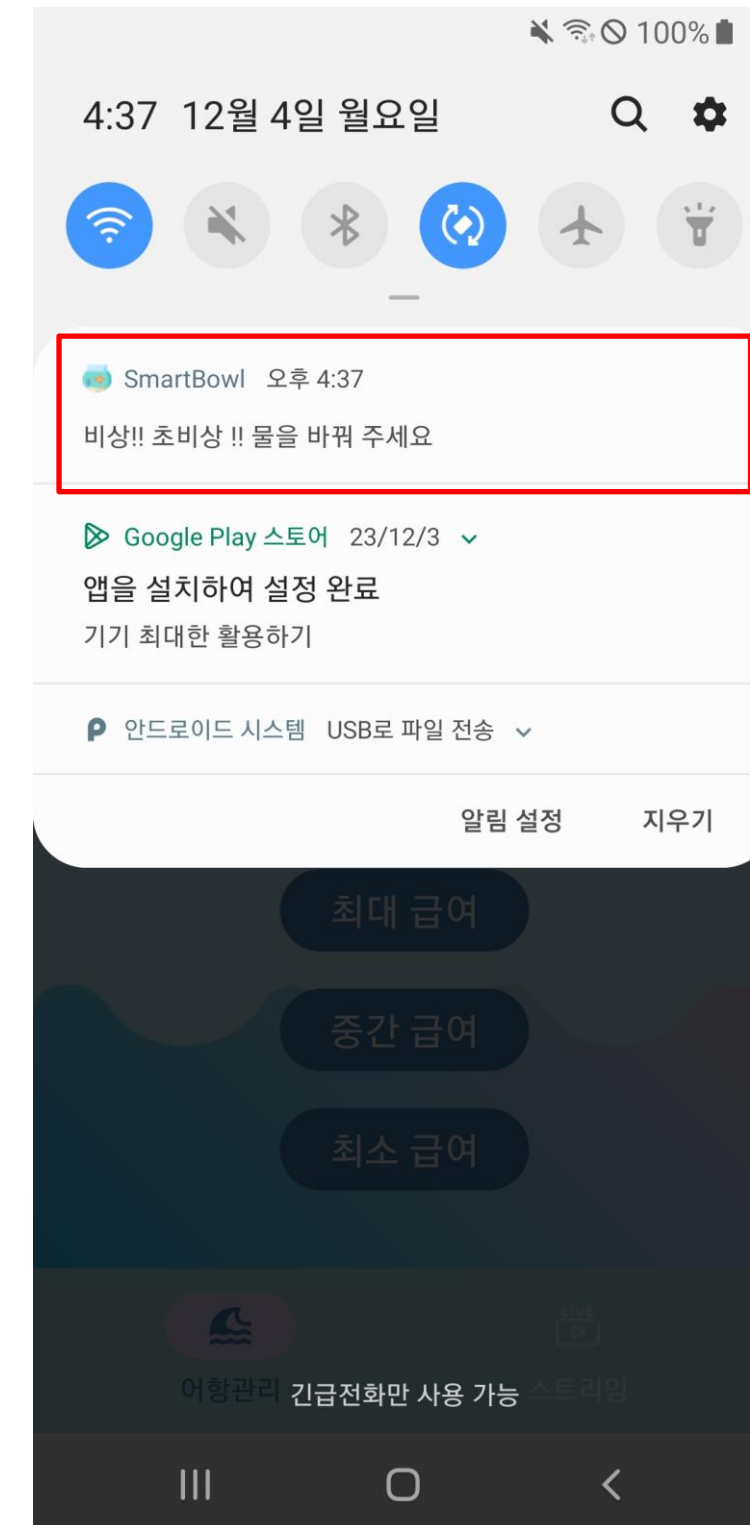
수질이 5 미만일 때
(너무 낮을 때)



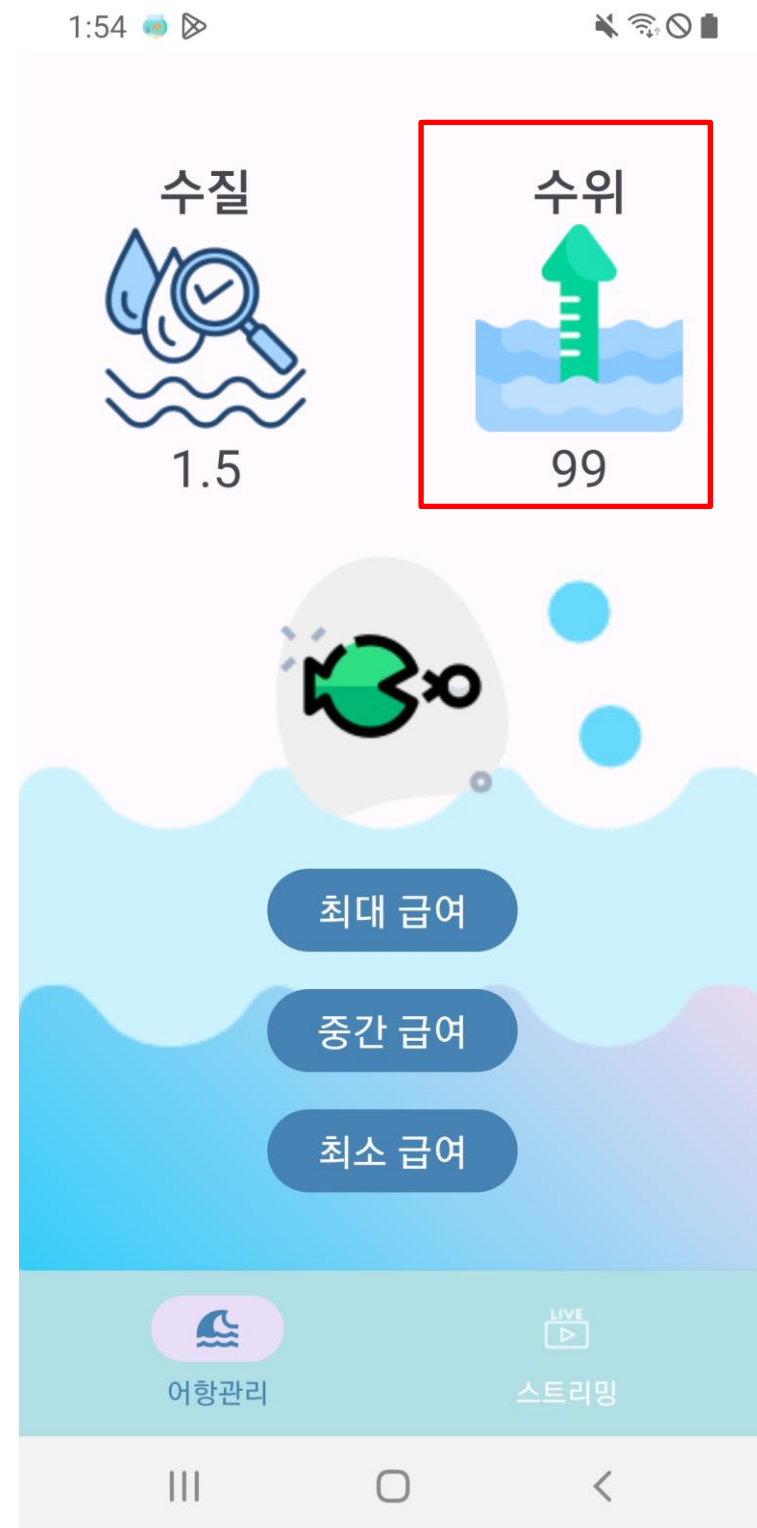
안드로이드 앱 개발 - 팝업 알람(수질)



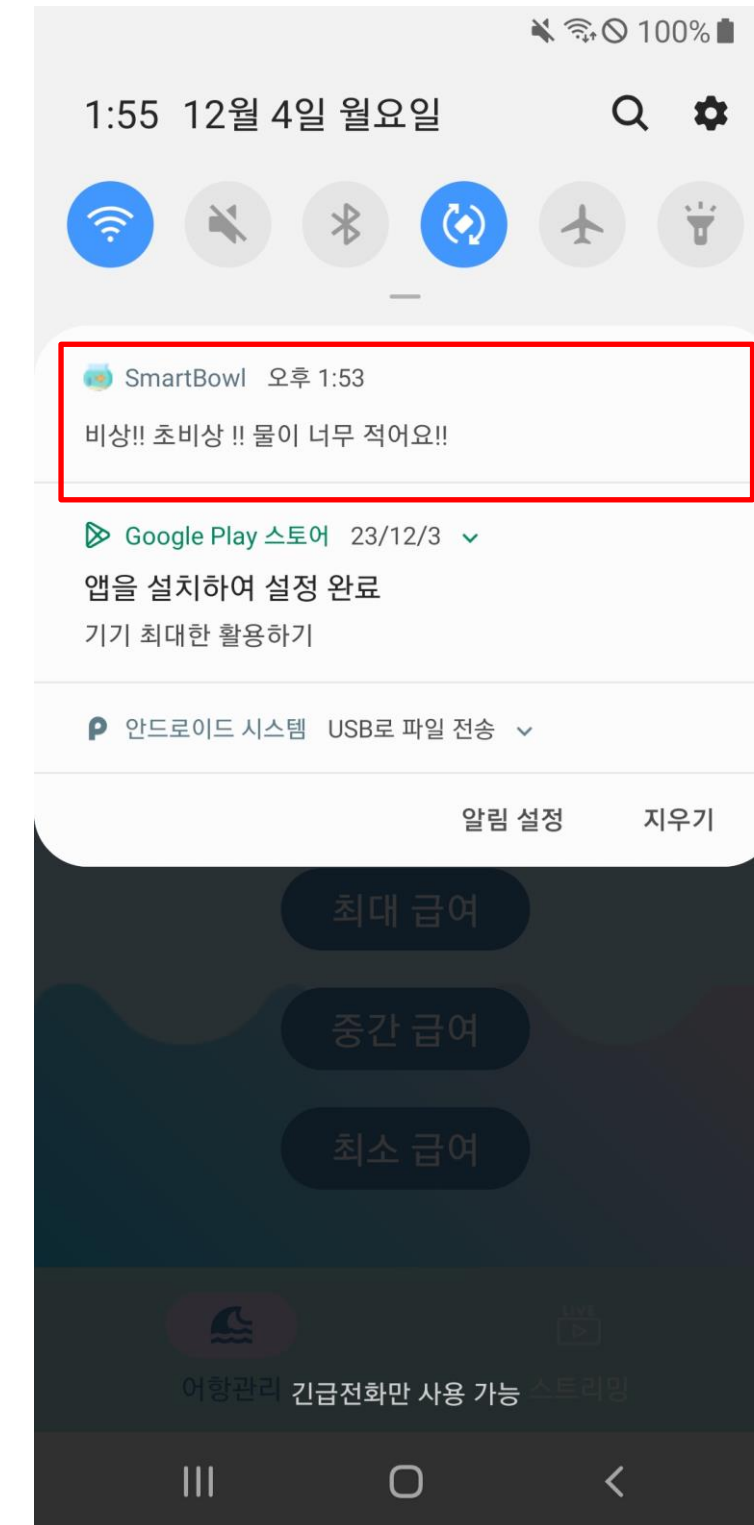
수질이 9 초과일 때
(너무 높을 때)



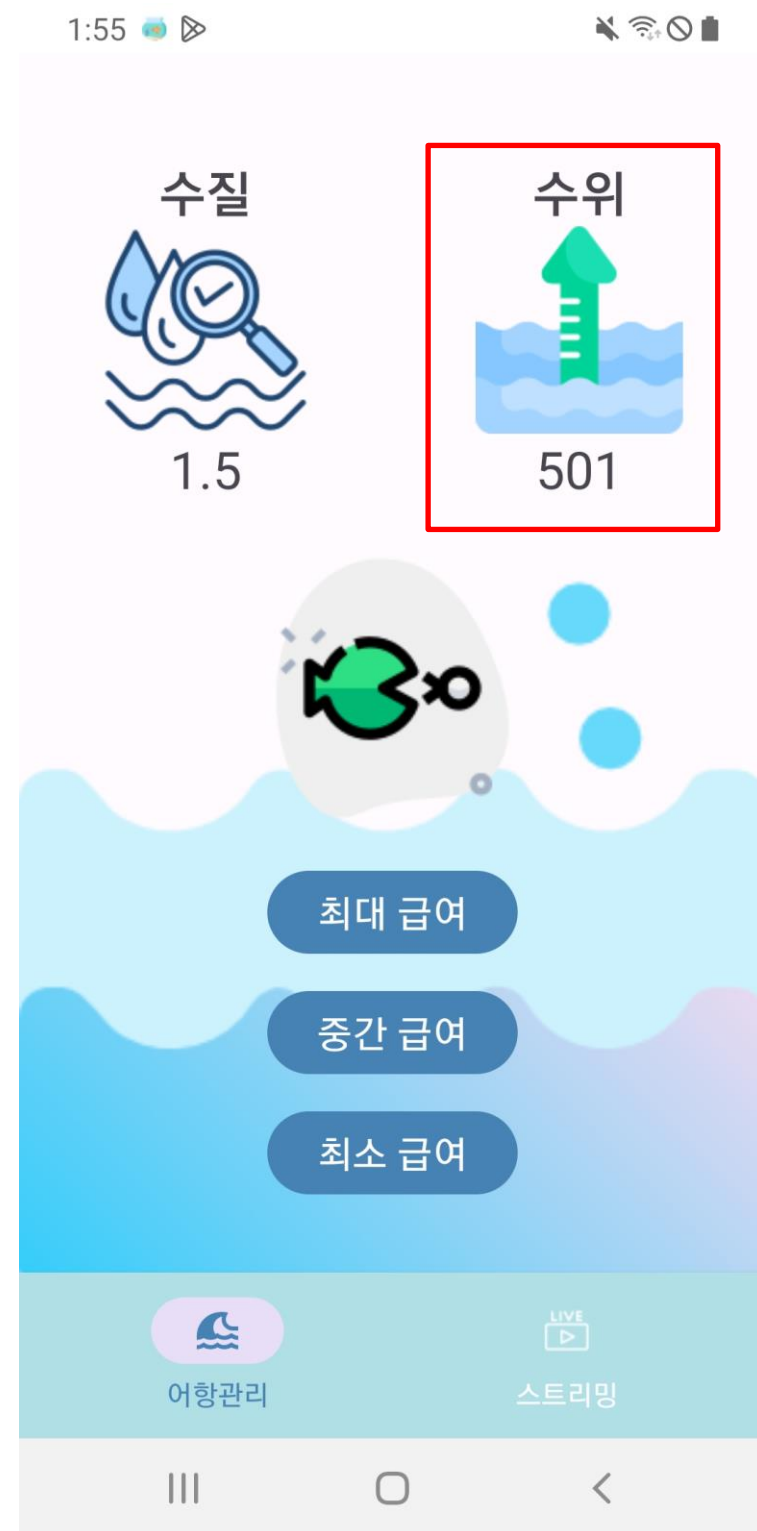
안드로이드 앱 개발 - 팝업 알람(수위)



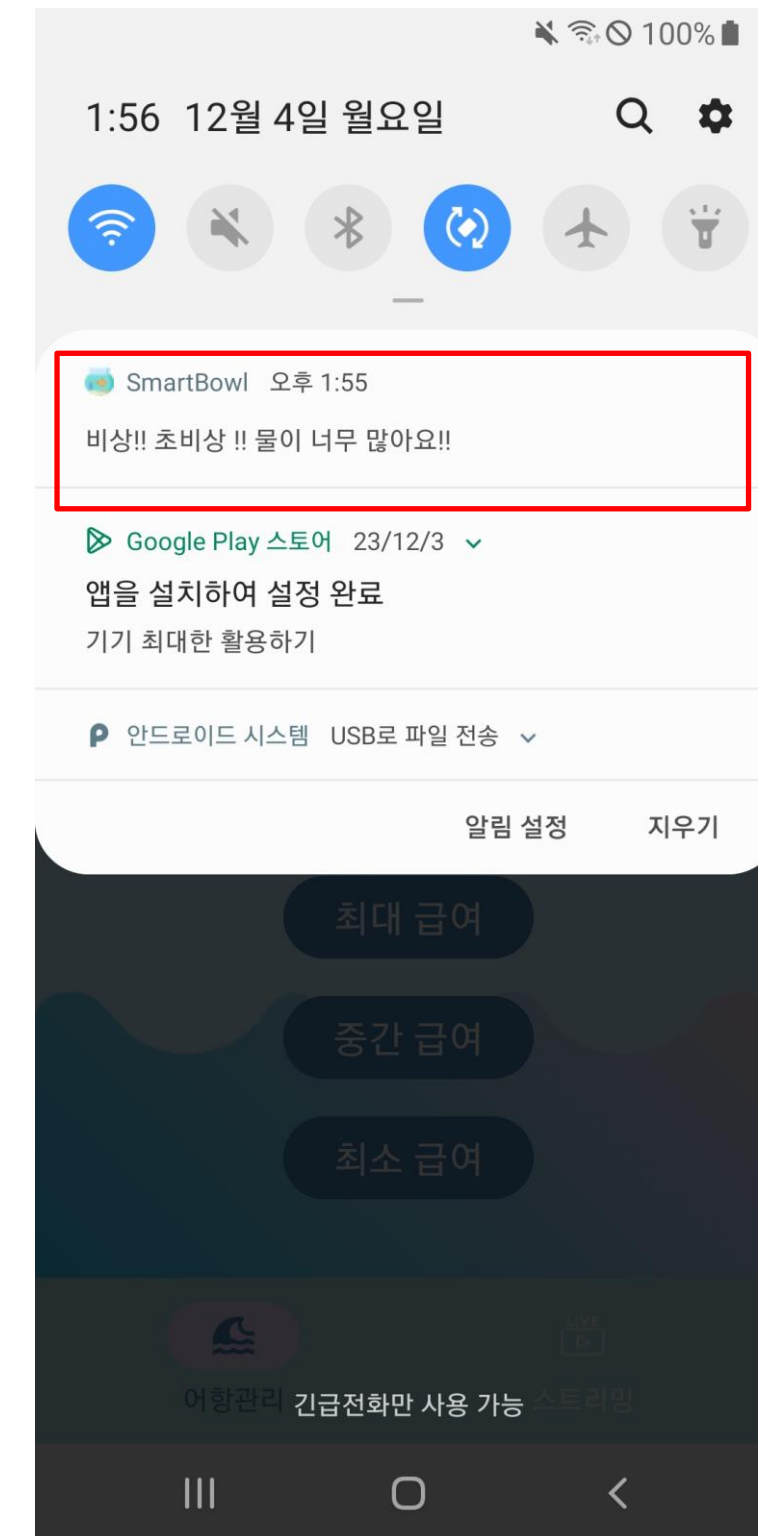
수위가 100 미만일 때
(너무 낮을 때)



안드로이드 앱 개발 - 팝업 알람(수위)



수위가 500 초과일 때
(너무 높을 때)



안드로이드 앱 개발 - 팝업 알람(수위)



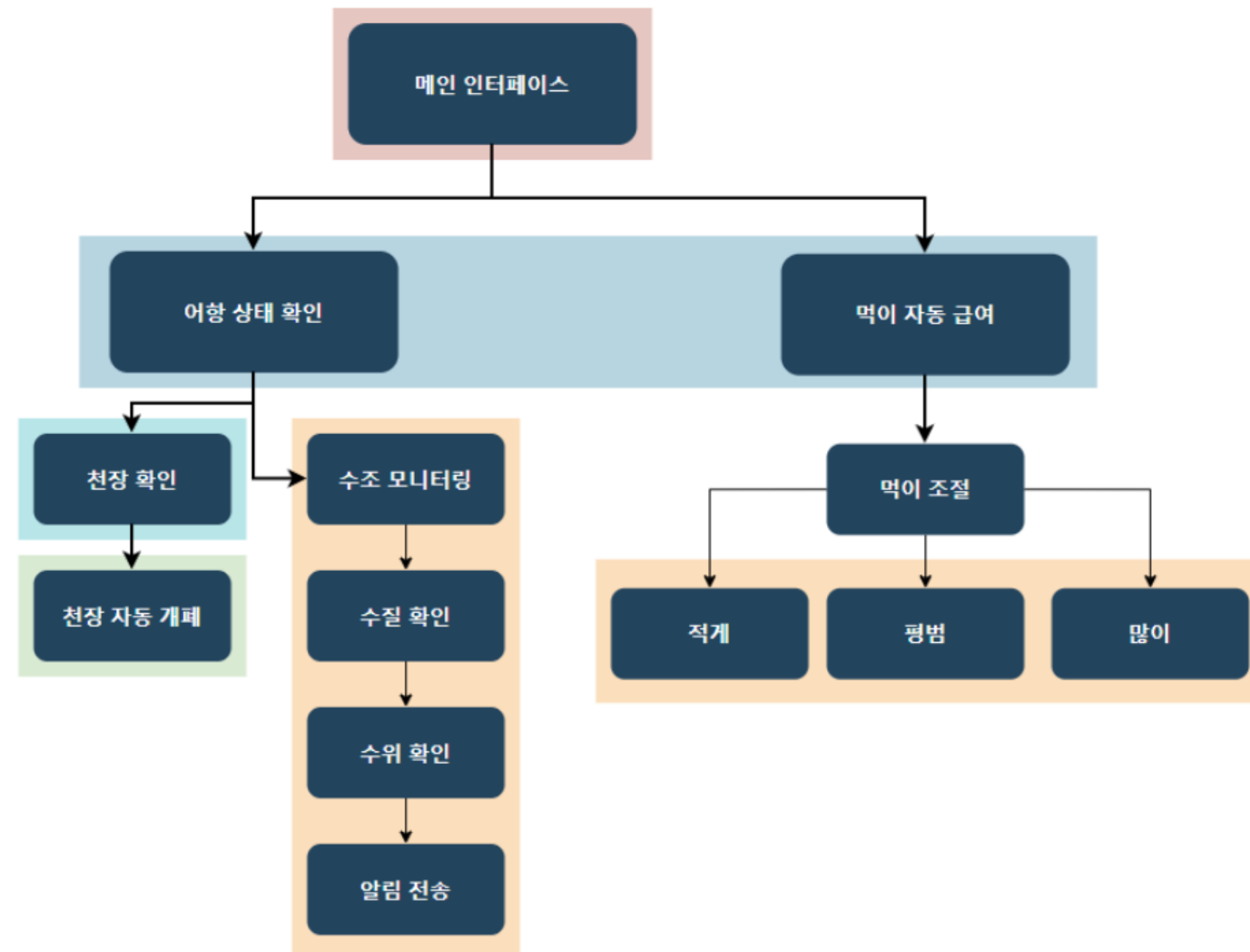
중간급여 Click시



```
└─ feeding
    └─ full: 0
        └─ half: 1
            └─ little: 0
```

정리

구조도



감사합니다.