|  |
| --- |
| **가천대학교** |
| 아두이노 센서 모듈을 사용한 스마트 어항 서비스 |

**프로젝트**

**완료보고서**

**2022.12.13**

1팀

컴퓨터공학과 201735908 권민혁

컴퓨터공학과 201735963 박재균

컴퓨터공학과 201736018 이호왕

컴퓨터공학과 202035223 장정우



**목 차**

1. 프로젝트 개요 3

[2.](#_Toc500175999) 프로그램 구성 5

[3.](#_Toc500176000) 프로젝트 개발결과 8

4. 프로그램의 소스파일 구조 27

5. 프로젝트 완료 후기 31

6. 소스코드 별첨 35

# 프로젝트 개요

* **프로젝트 명**

|  |
| --- |
| 『 **아두이노 센서 모듈을 사용한 스마트 어항 서비스**』개발 |

* **프로젝트 요약**

|  |
| --- |
| 우리 조는 IOT 서비스 중, 장기간 집을 비우는 현대인을 위해 제공해 줄 수 있는 서비스가 무엇이 있을까? 하는 고민을 하게 되었다. 추가로, 다양한 센서를 사용하는 아두이노 서비스를 구현하고 싶다는 생각을 하였다. 요즘 반려어를 키우는 가정이 늘어나고 있는 추세와, 반려어를 잘 키우기 위해서는 여러 요인들이 잘 갖추어 져야 한다는 점을 고려하여 스마트 어항 시스템을 개발하기로 결정하였다.  반려어를 위해 온도, 수위, 탁도, PH를 적정하게 맞추어야 한다는 것을 알게 되었고, 위 네 가지 센서를 활용하기로 결정했다. 또한, 출근 혹은 외출을 하게 되면 한 나절 이상 자리를 비울 수 있다는 점을 고려하여 자동으로 먹이를 급여할 수 있도록 3D 프린터로 먹이 급여기를 제작하였다.  위 요인들은, 어종에 따라 다른 값을 설정해주어야 한다는 점을 고려하여 사용자가 본인이 보유한 어종에 맞게 값을 설정 해줄 수 있어야 한다고 생각하였다. 따라서, 사용자가 편하게 접근할 수 있는 APP 서비스를 제공하기로 하였다.  사용자가 희망한 센싱값에서 벗어나면 APP을 확인하여 상태를 점검할 수 있게끔 Firebase Cloud Messaging 서비스를 이용해 푸시 알림을 주기로 하였고, 각 어항마다 다른 값들을 저장해 주기 위해 Raspberri pi에 서버와 데이터베이스를 설치하여 서비스를 관리하기로 하였다.  기획부터 완료발표까지 총 4주라는 짧은 기간 내에 사용자를 고려한 서비스를 만든다는 과정이 쉽지는 않았지만, 날밤을 지세우며 서비스 완성을 위해 노력을 다했고, 팀원들과 협업하는 방법을 배우며 각자의 기술 스택에 대한 견문을 넓힐 수 있었기에 상당히 의미있는 P-실무 프로젝트 였다. |

* **팀구성**

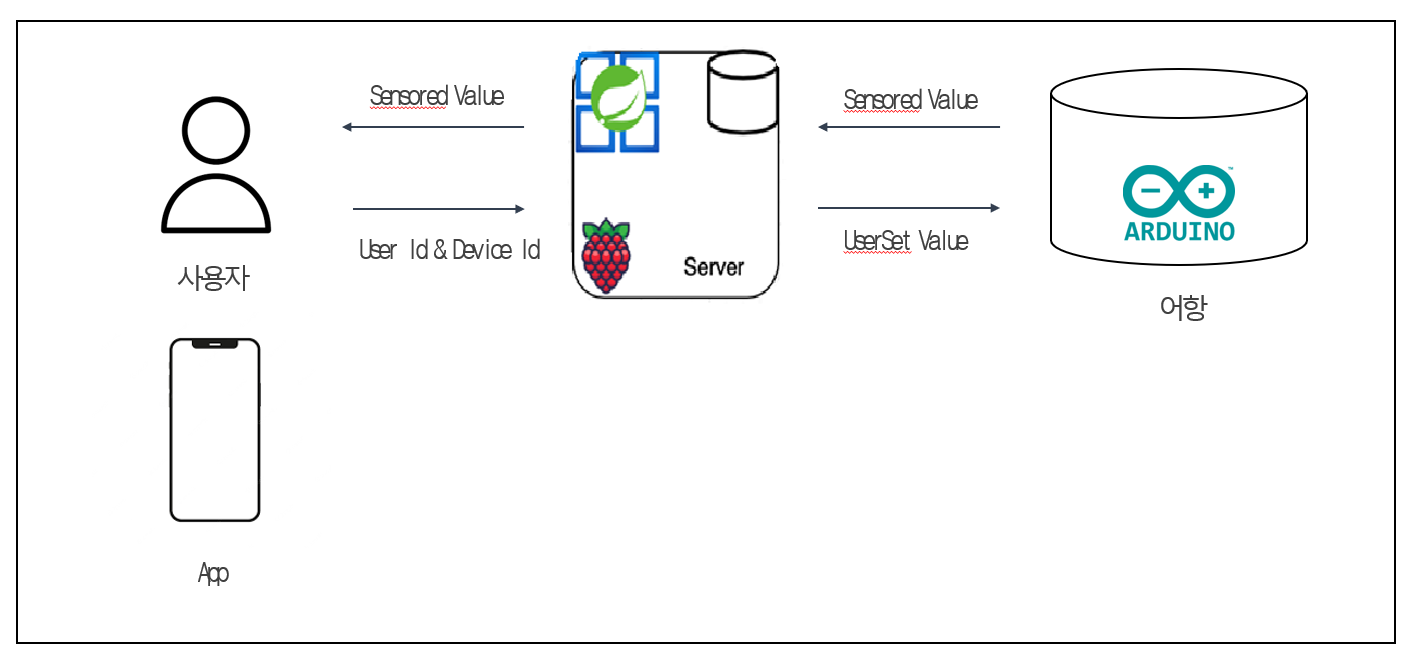
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 역할 | 학번 | 이름 | email | 비고 |
| App, Raspberri pi | 201735908 | **권민혁** | danny9kwon@gmail.com | 팀장 |
| Arduino 설계 및 구현 | 201735963 | **박재균** | pjg9905@naver.com | 팀원 |
| DB&Push Server 구현 | 201736018 | **이호왕** | thrusum123@gmail.com | 팀원 |
| AI 연동, Server 구현 | 202035223 | **장정우** | jw1010110@gmail.com | 팀원 |

* **프로젝트 과정**

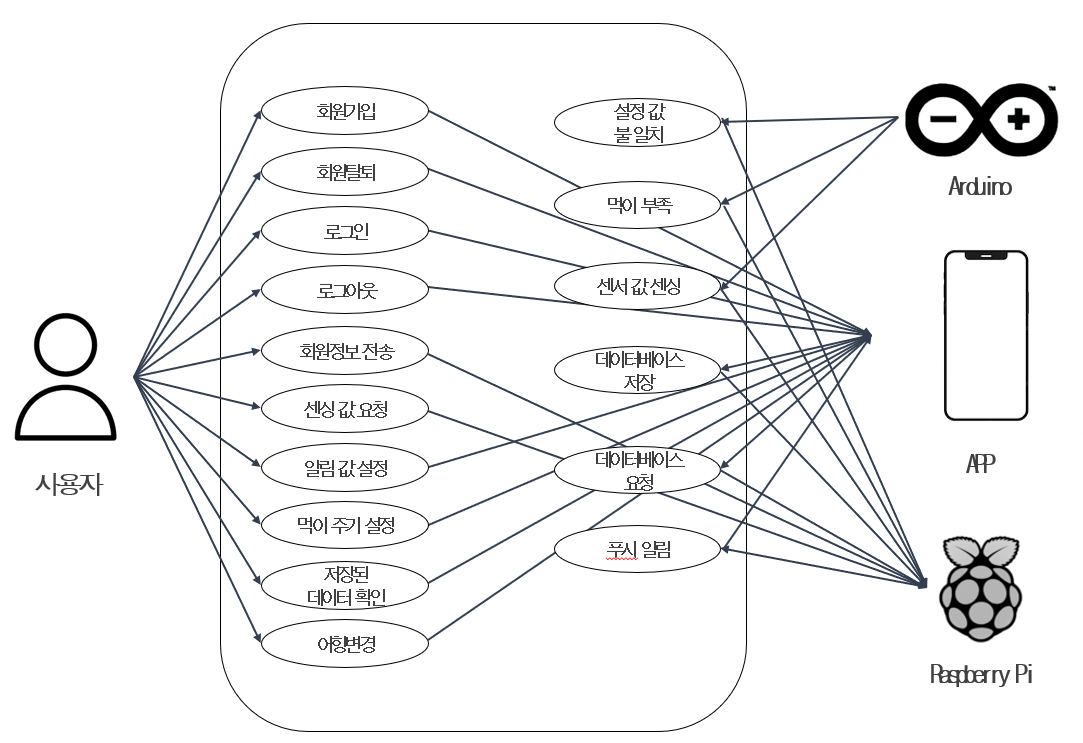
|  |
| --- |
| 텍스트, 사람, 노트북, 실내이(가) 표시된 사진  자동 생성된 설명 |
| 텍스트, 사람, 노트북, 실내이(가) 표시된 사진  자동 생성된 설명 |
| 텍스트, 사람, 노트북, 실내이(가) 표시된 사진  자동 생성된 설명 |

# 프로그램 구성

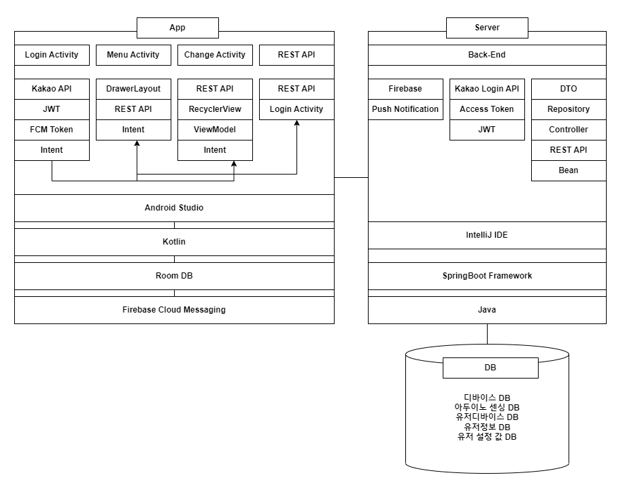
* **사용자 시나리오 구성도**



* **사용자 유스케이스 다이어그램**



* **시스템 아키텍쳐**



* **운영 실행환경**

|  |  |
| --- | --- |
| 실행환경 | 실행절차 및 필요사항 |
|
| PIS Server  실행환경 | 1.ssh로 서버접속 ………… 외부: ssh ubuntu@222.238.106.88 -p 9450  내부: ssh ubuntu@192.168.0.52  2.로그인 …………………………… 아이디: ubuntu, 암호: ubuntu  3.서버 디렉터리 이동  …………$ cd 다운로드/smartfishbowl/Smartfishbowl/SERVER/fishbowl/build/libs  4.서버 실행 ………………………… $ java -jar fishbowl-0.0.1-SNAPSHOT.jar  5.실행 정상  … Started FishbowlApplication in 35.824 seconds (JVM running for 38.96) **출력** |
| APP  실행환경 | 1.github접속 …………………………… https://github.com/Danny-Kwon/apks.git  2.apk 파일 다운로드 ………………………………………… SmartFishBowl.v.1.0.apk  3.앱 계정 만들기 …………………………………………………………… 카카오 로그인  4.어항 등록하기 ………………………………………… 어항 ID 입력(테스트 ID: 123)  5.실행 정상 ……………………………………………… 현재 어항 센서 값들 보기 가능 |

* **개발내용**

|  |  |  |  |
| --- | --- | --- | --- |
| 개발대상 | 주요기능 | 개발결과 | |
| 개발자 | 완료율 |
| Android Native APP | 회원가입 기능, 로그인 기능, 어항 추가 기능, 어항 삭제 기능, 어항 변경 기능, 현재 센서 값 확인 기능, 희망 온도 설정 기능, 희망 수위 설정 기능, 희망 PH 설정 기능, 희망 탁도 설정 기능, 자동 먹이 급여 시간 설정 기능, 로그아웃 기능, 회원 탈퇴 기능 | 권민혁 | 100% |
| Back-end | Spring Security 보안 로직 설정,Spring Security Custom filter를 통한 인증 기능, Jpa 쿼리 메서드, 자동 회원가입 기능, 로그인 시 jwt 토큰 발급 기능, 추가된 어항 저장 기능, 설정된 온도 저장 기능, 설정된 수위 저장 기능, 설정된 ph 저장 기능, 설정된 탁도 저장 기능, 자동 먹이 급여 시간 저장 기능, 회원 탈퇴 시 관련 정보 삭제 기능 | 장정우 | 100% |
| Back-end | DB구현, 서버와 아두이노 간 웹 소켓 통신 기능, firebase cloud messaging 기능 | 이호왕 | 100% |
| Arduino | 아두이노 회로도 작성 및 조립, 각 센서 측정 기능, 서보모터를 이용한 먹이 주기 기능, 방열 패드와 쿨러를 이용한 온도 조절 기능 | 박재균 | 100% |

# 프로젝트 개발 결과

**◾ 앱 화면(LoginActivity - 로그인 전)**

****

**◾ 앱 화면(LoginActivity – 카카오 로그인)**

**텍스트이(가) 표시된 사진

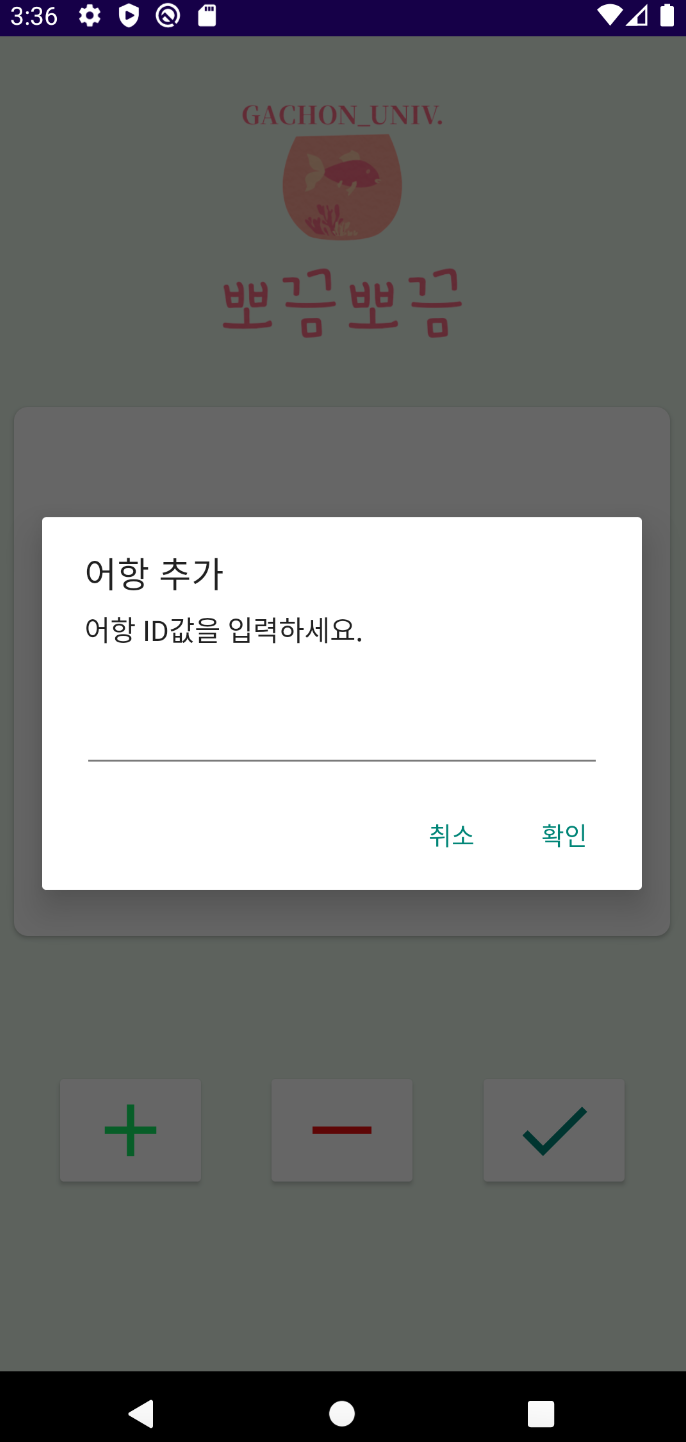
자동 생성된 설명**

**◾ 앱 화면(ChangeActivity – 어항 추가, 제거 및 선택 화면)**

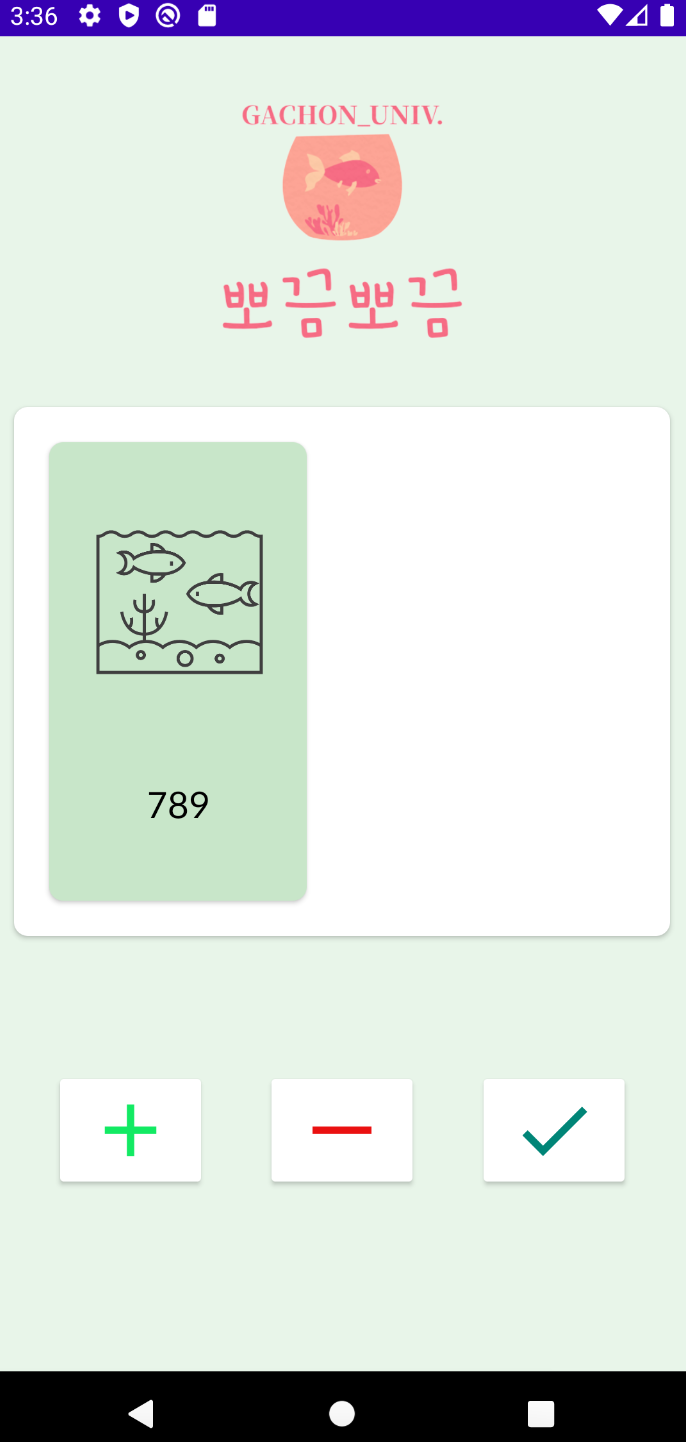
**광장이(가) 표시된 사진

자동 생성된 설명**

**◾ 앱 화면(ChangeActivity – 어항 추가)**

****

**◾ 앱 화면(ChangeActivity – 어항 추가 완료)**

****

**◾ 앱 화면(MenuActivity - 현재 센싱값 확인 및 희망 값 설정 화면)**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**◾ 앱 화면(MenuActivity - 적정 온도 수정)**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**◾ 앱 화면(MenuActivity - 적정 수위 수정)**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**◾ 앱 화면(MenuActivity - 적정 PH 수정)**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**◾ 앱 화면(MenuActivity - 적정 탁도 수정)**

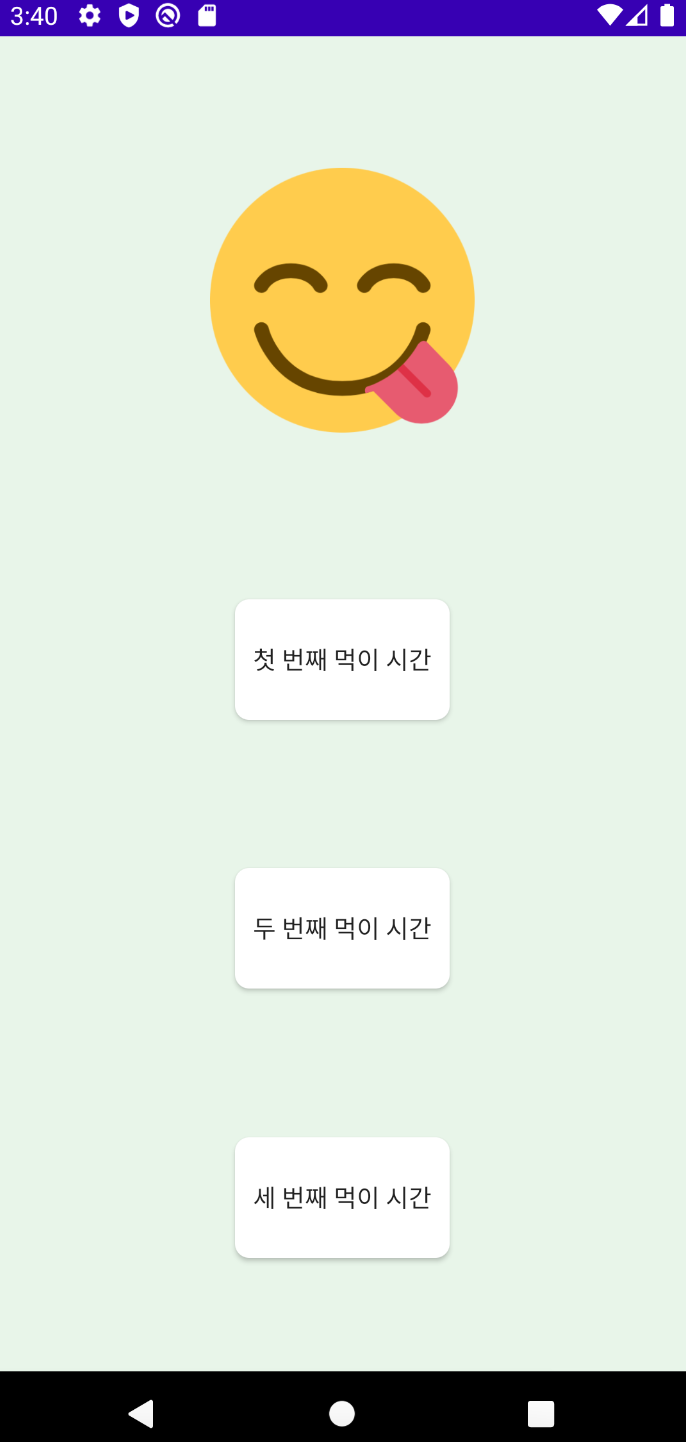
**텍스트이(가) 표시된 사진

자동 생성된 설명**

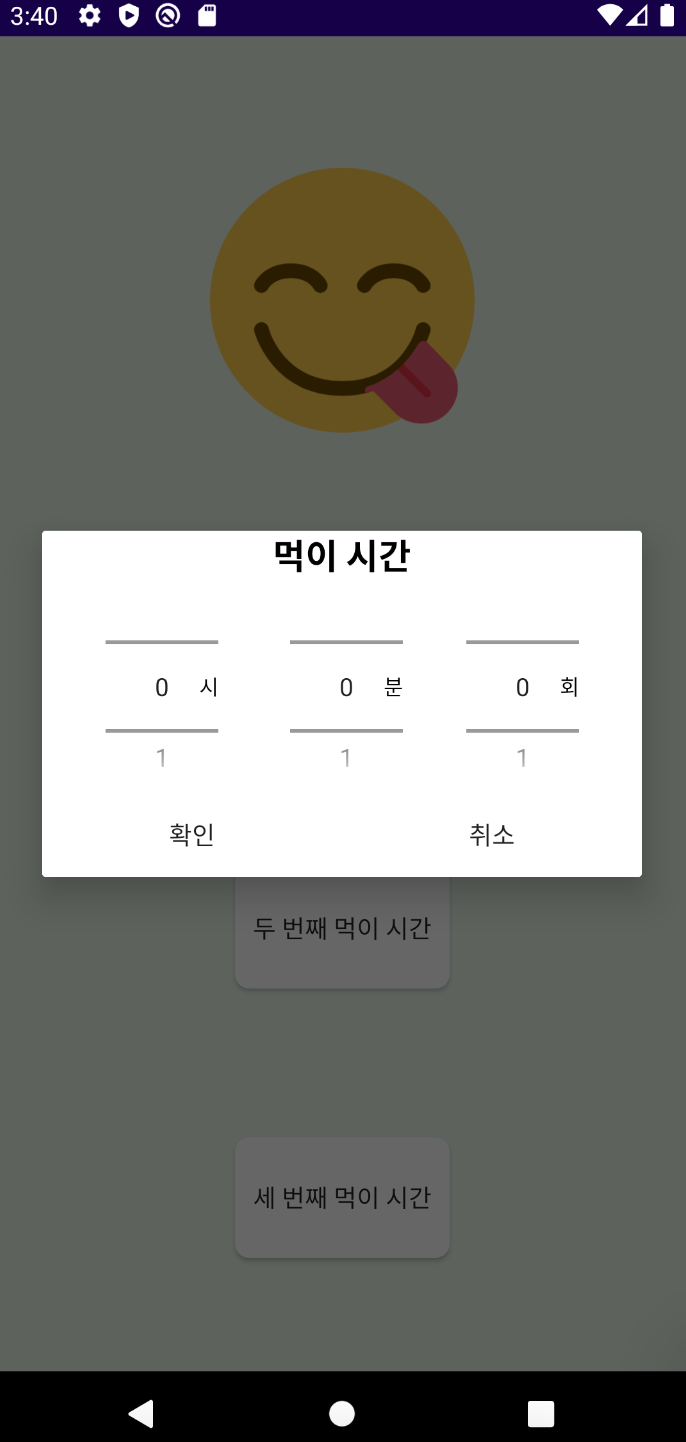
**◾ 앱 화면(MenuActivity - Drawer 열린 후 화면)**

****

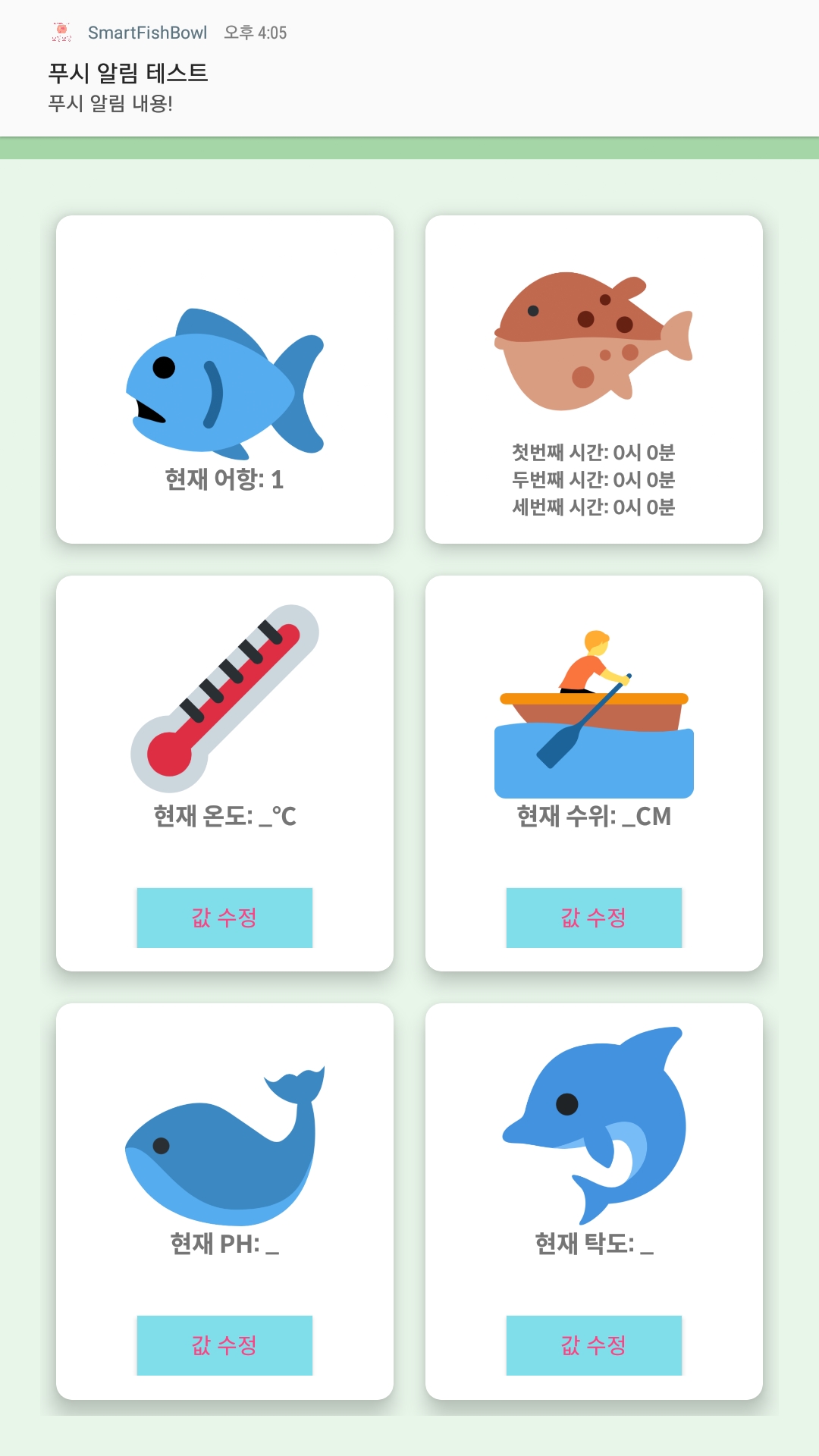
**◾ 앱 화면(TimeActivity - 자동 먹이 급여시간 설정)**

****

**◾ 앱 화면(TimeActivity - 먹이 시간 및 횟수 설정 넘버 picker화면)**

****

**◾ 앱 화면(FCM 수신 화면 - 포그라운드)**

****

**◾ 앱 화면(FCM 수신 화면 - 백그라운드)**

**텍스트, 전자기기, 파란색이(가) 표시된 사진

자동 생성된 설명**

**◾ 앱 화면(FCM 수신 화면 - 알림 센터)**



**◾ 서버 화면(SSH 접속 화면)**

텍스트이(가) 표시된 사진

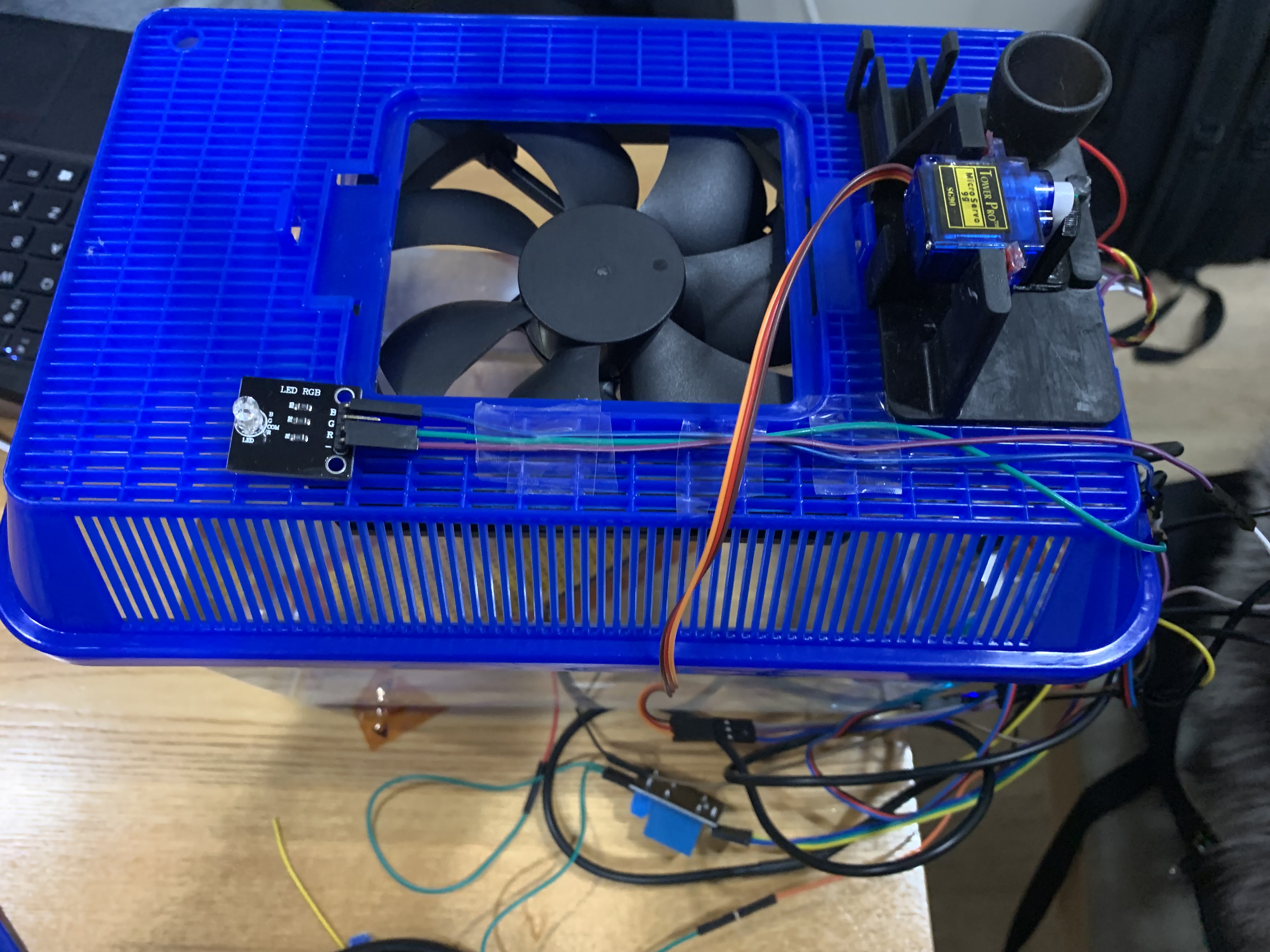
자동 생성된 설명

**◾ 어항 사진(측면)**

실내, 바닥이(가) 표시된 사진

자동 생성된 설명

**◾ 어항 사진(상단)**



# 프로그램의 소스파일 구조

* **Android APP 소스 구조 및 파일**

|  |  |  |
| --- | --- | --- |
| 디렉터리 | 소스파일명 | 소스파일설명 |
| SmartFishBowl/app/src/main | | |
| java/com/examples/smartfishbowl | GlobalApplication | 카카오 App key 저장 |
| java/com/examples/smartfishbowl | LoginActivity | 카카오 로그인 및 FCM 토큰 수신 기능 |
| java/com/examples/smartfishbowl | ChangeActivity | 어항 등록, 삭제 및 변경 기능 |
| java/com/examples/smartfishbowl | MenuActivity | 센싱값 수신, 세팅값 전송 기능 |
| java/com/examples/smartfishbowl | TimeActivity | 먹이 자동 급여 시간 설정 기능 |
| java/com/examples/smartfishbowl  /work | MyWorker | FCM 백그라운드 작업 기능 |
| java/com/examples/smartfishbowl  /viewmodel | BowlViewModel | 어항 뷰 리사이클러뷰 Room DB 함수 사용 설정 |
| java/com/examples/smartfishbowl  /sharedpreferences | PreferenceUtil.kt | SharedPreference 자료형 별 값 저장법 |
| java/com/examples/smartfishbowl  /firebasemessageservice | MyFirebaseMessagingService.kt | 앱 상태 별 FCM data payload 수신 기능 |
| java/com/examples/smartfishbowl/  database | BowlData | RoomDB 테이블 |
| java/com/examples/smartfishbowl  /database | BowlDao | RoomDB 쿼리 함수 |
| java/com/examples/smartfishbowl  /database | BowlDatabase | RoomDB 호출 클래스 |
| java/com/examples/smartfishbowl  /api | APIS | Retrofit2 를 이용한 REST API 인터페이스 |
| java/com/examples/smartfishbowl  /api | Datas.kt | Json format 데이터 클래스 |
| java/com/examples/smartfishbowl  /adapter | BowlAdapter | 어항 뷰 리사이클러뷰 어댑터 |
| res | drawable | 드로어블 이미지 폴더 |
| res | font | 앱 내 글꼴 |
| res | layout | 앱 화면 레이아웃 |
| res | menu | drawer 메뉴 아이템 레이아웃 |
| res | mipmap | 앱 로고 |
| res | values | 색상, 문자열, 앱 테마 |

* **Server 소스 구조 및 파일**

|  |  |  |
| --- | --- | --- |
| 디렉토리 | 소스파일명 | 소스파일설명 |
| fishbowl/src/main/java/com/gachon | | |
| fishbowl/config/firebase | FirebaseConfig | firebase 서버와 연결하기 위해 설정해둔 파일 |
| fishbowl/config/security | SecurityConfig | jwt 토큰 필터 등록, uri 권한을 설정해둔 파일 |
| fishbowl/config/websocket | WebSocketConfig | 아두이노와 통신할 소켓 uri을 설정해둔 파일 |
| fishbowl/controller | AndroidController | 앱과 통신하기 위한 API를 정의해둔 파일 |
| fishbowl/dto | FeedTimeDto | POST 통신으로 먹이 시간/횟수 값을 앱에서 받기 위한 DTO |
| fishbowl/dto | FireabseDto | POST 통신으로 firebase 토큰 값을 앱에서 받기 위한 DTO |
| fishbowl/dto | LoginDto | POST 통신으로 사용자 이메일와 비밀번호를 앱에서 받기 위한 DTO |
| fishbowl/dto | SensingDto | POST 통신으로 알림 희망 값을 앱에서 받기 위한 DTO |
| fishbowl/dto | TokenDto | POST 통신으로 카카오 Access token 값을 앱에서 받기 위한 DTO |
| fishbowl/dto | UserDto | 카카오 resource 서버에서 사용자 이메일, 패스워드, 이름을 받기 위한 DTO |
| fishbowl/entity/role | Role | 유저와 admin을 분류하기 위한 객체 |
| fishbowl/entity | DeviceId | 디바이스 아이디를 저장하는 테이블 |
| fishbowl/entity | Sensing | 아두이노에서 센싱된 값을 저장하는 테이블 |
| fishbowl/entity | UserDevice | 디바이스와 사용자 아이디를 매핑하기 위한 테이블 |
| fishbowl/entity | UserId | 유저 정보를 저장하는 테이블 |
| fishbowl/entity | UserSet | 유저가 설정한 알림 희망 값을 저장하는 테이블 |
| fishbowl/entity | UserSetFoodTime | 유저가 설정한 먹이 시간/횟수를 저장하는 테이블 |
| fishbowl /jwt | JwtFilter | jwt 토큰을 활용해 request를 필터링하는 파일 |
| fishbowl /jwt | TokenProvider | jwt 토큰 생성 및 검증을 하는 파일 |
| fishbowl/repository | DeviceIdRepository | DeviceId 테이블 관련 Jpa 메서드 정의한 파일 |
| fishbowl/repository | SensingRepository | Sensing 테이블 관련 Jpa 메서드 정의한 파일 |
| fishbowl/repository | UserDeviceRepository | UserDevice 테이블 관련 Jpa 메서드 정의한 파일 |
| fishbowl/repository | UserSetFoodTimeRepository | UserSetFoodTime 테이블 관련 Jpa 메서드 정의한 파일 |
| fishbowl/repository | UserSetRepository | userSet 테이블 관련 Jpa 메서드 정의한 파일 |
| fishbowl/service/socket | ArduinoSocketService | 아두이노와 소켓 통신을 위해 정의한 파일 |
| fishbowl/service | DeviceIdService | deviceIdRepository에 정의된 Jpa메서드를 활용한 기기 검색 구현 파일 |
| fishbowl/service | FirebaseService | 앱에 보낼 푸시 알림을 정의한 파일 |
| fishbowl/service | SensingService | sensingRepository에 정의된 Jpa 메서드를 활용한 센싱 데이터 검색 구현 파일 |
| fishbowl/service | UserDeviceService | userDevice 테이블과 관련된 검색, 검증, 저장 메소드를 정의한 파일 |
| fishbowl/service | UserIdService | userId 테이블과 관련 메소드를 정의한 파일 |
| fishbowl/service | UserSetFoodTimeService | : userSetFoodTime 테이블과 관련된 검색, 검증, 저장 메소드를 정의한 파일 |
| fishbowl/service | UserSetService | userSet 테이블과 관련된 검색, 저장, 삭제 메소드를 정의한 파일 |

* **Arduino 소스 구조 및 파일**

|  |  |  |
| --- | --- | --- |
| 디렉터리 | 파일명 | 소스파일설명 |
| Arduino/fishbowl | fishbowl.ino | 아두이노와 연결된 기기들을 제어하는 소스파일 |
| Arduino/libraries/ArduinoJson | ArduinoJson.h | 아두이노에서 JSON 형식의 데이터를 파싱하는 라이브러리 |
| Arduino/libraries/ESP8266-Websocket | WebSocketClient.h | 웹 소켓에서 클라이언트로 데이터를 주고받는 라이브러리 |
| Arduino/libraries/DallasTemperature | DallasTemperature.h | 온도센서를 제어하는 라이브러리 |
| Arduino/libraries/ESP32\_Servo | ESP32\_Servo.h | 서보모터를 제어하는 라이브러리 |
| Arduino/libraries/NTPClient | NTPClient.h | 현재 시간을 받아오는 라이브러리 |
| Arduino/libraries/OneWire | OneWire.h | 1-Wire 통신을 가능하게 해주는 라이브러리 |

# 프로젝트 완료후기

* **권민혁**

|  |  |
| --- | --- |
| 사람, 벽, 실내, 소년이(가) 표시된 사진  자동 생성된 설명 | * **나의 소감**   종합 프로젝트를 끝내고 바로 P-실무 프로젝트를 시작하게 되었다. 종합 프로젝트는 입학 후, 학교에서 한 첫 프로젝트 였기에 아쉬운 점도 있었고 보완해야할 점들도 있었다. 그때 아쉬웠던 점은, 주제 선정에 있어 실현이 어려운 목표를 잡았다는 점이었다. 기획할때 주제에 넣은 단어 때문에, 원했던 기능을 구현하는데 성공했음에도 불구하고 발목이 잡혀 아쉬운 평가를 듣기도 했었다. 그에 따라, 이번 프로젝트에서는 좀 더 접근이 쉽고 원하는 주제 그대로를 구현할 수 있는 스마트 어항을 개발하게 되었다. IOT 서비스 구현인 만큼, APP이란 플랫폼이 주가 아니었지만 사용자가 이 서비스를 어떻게 사용하는 것이 가장 편할까를 생각하며 개발해보게 되었다. 10주간 개발기간으로 시간을 쓸 수 있었던 종합프로젝트와 비교해서, 기획~유지보수 및 최종 발표 까지 4주안에 모든것을 끝내야 하기 때문에 시간적 압박도 받았다. 그러나 실무에서 서비스들을 개발하려면 더한 극한의 상황에서도 사용자가 원하는 소프트웨어를 만들어야 한다는 것을 알기 때문에 상당히 의미있는 시간을 보낸 것 같다.  팀원으로 참여했던 지난 프로젝트와 달리 이번에는 팀장으로 프로젝트를 진행하게 되었는데, 모든 팀원들이 자신이 맡은 부분을 해내기 위해 밤을 세우며 고생해 준것에 감사한 마음이 들었다. |
| * **P 실무 프로젝트를 통해 얻은 것**   그간 앱 개발을 하면서, 기능을 만들고 완료하는 것도 버겁다는 생각을 하곤 했었다. 그러나 이번에는 비동기 방식으로 프로그램을 처리하고, FCM 을 백그라운드에서도 수신할 수 있도록 작업 처리하는 등 Native App이 가질 수 있는 좋은 성능을 최대한 활용할 수 있도록 노력하였고, 현재 기술 스택을 발전할 수 있었던 좋은 계기가 되었다.  앱 개발에만 전념하였던 이전 프로젝트와 다르게, Raspberri pi에 우분투를 설치하여 스프링부트로 구축한 서버 배포 및 포트포워딩을 해보게 되었다. 서버가 필요한 앱 서비스를 관리함에 있어 리눅스를 사용하는 것은 필수인데, 이번 기회를 통해 실무에서 활용할 지식을 얻게 되었다.  또한, 각자 다른 4명의 팀원과 함께 서비스를 완성하기 위해 팀장으로써 어떻게 해야하는가를 생각해 볼 수 있어서 좋았다. | |

* **장정우**

|  |  |
| --- | --- |
| 사람, 서있는, 가장이(가) 표시된 사진  자동 생성된 설명 | * **나의 소감**   P실무 프로젝트를 진행하며 IOT Service를 개발했다. 주제가 IOT인 만큼 Arduino, Back-End, App 각각을 담당하는 팀원들과 개발 속도를 맞춰 진행하는 점이 힘들었다. 특히 App과 Arduino와 데이터를 주고받는 과정에서 사용되는 통신 프로토콜을 결정하고 인증 과정을 구현하는 부분에서 내가 담당한 Back-End 뿐 아니라 팀원의 포지션에 대한 정보도 조금은 알아야 했기 때문에 고생을 했던 것 같다. 그리고 만약 다음 팀 프로젝트 기회가 생기면 CI/CD 파이프라인을 구축해 배포 환경을 개선해보고 싶다는 생각을 했다. 이번 p실무 프로젝트에서는 로컬 환경에서 빌드하여 Githup에 push하고 배포를 담당한 팀원이 jar파일을 pull 하여 직접 배포하는 방식으로 진행되었다. 하여 재배포 과정에서 서비스가 중단되는 일이 몇 번 발생하였고 Test Code를 제대로 통과하지 않은 코드가 배포될 뻔한적도 있었기 때문이다. 다음 프로젝트에선 Cubernetes 환경에서 개발하여 Jenkins를 통해 CI/CD를 구축해보고 싶다. |
| * **P 실무 프로젝트를 통해 얻은 것**   프로젝트를 진행하며 학부 과정 내에서 얻은 지식들을 팀원들과 함께 응용해볼 수 있었다. 나아가 이론으로 공부했던 것들을 직접 구현해볼 수 있었으며, 팀 단위의 공동 프로젝트 경험을 얻을 수 있었다. 마지막으로 혼자 토이 프로젝트를 진행할 때와는 다르게, 여러 팀원들의 피트백을 통해 새로운 인사이트를 얻어갈 수 있었다. | |

* **이호왕**

|  |  |
| --- | --- |
|  | * **나의 소감**   10주동안 종합프로젝트를 진행하고 바로 4주동안 p실무프로젝트를 진행해보니 상대적으로 시간에 많이 쫓겨서 프로젝트를 진행했던 거 같다. 이번 프로젝트에서는 평소에 공부하지 않았던 서버 쪽 코딩 역할을 맡았는데 이번 학기 때 수강하였던 서버프로그래밍 강의에서 배운 내용을 생각하면서 진행했음에도 불구하고 생각보다 큰 어려움이 있었다. 그래서 혼자서 해결 못하는 부분은 팀원과의 소통을 통해서 해결을 하였고 이 과정에서 팀원들과 더 가까워질 수 있는 경험이었다고 생각한다. 짧은 시간동안 프로젝트를 진행하면서 어렵고 힘든 점도 많았지만 프로젝트를 하나씩 끝날 때마다 많은 지식을 몸으로 터득한 것 같아서 뿌듯하고 프로젝트는 나의 성장을 위해서 꼭 필요한 작업이라고 생각하게 됐다. |
| * **P 실무 프로젝트를 통해 얻은 것**   p실무프로젝트를 통해서 평소에 사용하지 않았던 툴과 프로그래밍 언어를 사용하였다. 평소에는 c 기반 언어를 사용해서 공부를 진행했지만 이번 프로젝트에서는 java 언어를 기반으로 spring boot 프레임 워크를 사용하였기 때문에 java 언어의 편의성과 프로젝트를 진행함에 있어서 좋은 언어라는 것을 깨닫았다. 그리고 firebase cloud messaging 기능을 다루면서 앱의 기능에 대한 내용 또한 약간이나마 접할 수 있었고 DB를 설계하는 과정에서 DB를 설계하기 위해서는 프로젝트의 전반적인 모든 내용을 알고 있어야 하며 잘못된 설계를 하였을 때 시스템에 큰 문제가 생긴다는 것을 알게 되었다. 처음에 DB를 설계할 때 구현을 하고 만약 틀린 점이 있다면 고치면 된다는 막연한 생각을 가지고 임했으나 프로젝트 진행 중간에 DB의 내용을 수정하는 일이 생겼는데 이때 DB 관련 API들을 모두 수정해야하는 사실을 깨닫고 DB를 잘못 설계했을 때 프로젝트 진행에 큰 문제가 생긴다는 점을 알았다. 이번 프로젝트로 자신의 무지함에 대해 많이 느꼈고 아직 배워야할 부분들이 많다는 것을 알았으니 앞으로는 더 여러가지에 대해 공부를 해봐야겠다. | |

* **박재균**

|  |  |
| --- | --- |
|  | * **나의 소감**   다른 팀원들과는 달리 종합 프로젝트를 경험해보지 않고 P-실무프로젝트를 하게 됐는데 본격적인 프로젝트는 처음이다 보니 모르고 미숙한 부분이 많았다. 특히 아두이노 자체를 처음 접해봐서 시작하는 단계에서 많이 헤매기도 했다. 그렇지만 아두이노에 익숙해질수록 문제가 생기거나 어떻게 진행해야 할지 생각이 나지 않을 때 헤쳐 나가는 방법을 빠르게 찾을 수 있었다. 짧은 시간인 4주동안 진행해야 하는 프로젝트이기 때문에 혼자서 해결하기 어려운 부분들은 팀원들과 소통을 통해 진행할 수 있었다. 어려움을 함께 헤쳐 나가준 팀원들에게 고마움을 전하고 싶다. |
| * **P 실무 프로젝트를 통해 얻은 것**   P-실무프로젝트를 통해 처음 접하게 된 아두이노에 대해 이해할 수 있었고 아두이노를 이용한 많은 IoT 기기들을 개발할 수 있다는 것을 깨달을 수 있었다. 아두이노와 사용할 장치들의 회로도를 작성하고 아두이노와 장치들을 연결하는 방법과 아두이노 IDE를 이용한 개발 방법을 배울 수 있었다. 아두이노를 만진다는 것이 많이 까다롭고 정교한 작업들과 사전 지식이 필요하다는 것을 배웠기 때문에 다음에 아두이노를 하게 된다면 지금보다 수월한 진행이 가능할 것 같다.  이번 프로젝트를 통해 앞으로도 진행해야 하는 프로젝트들을 어떤 방식으로 진행하면 좋을지 생각해보게 되었고 아직 하지 못한 종합 프로젝트를 대비해서 미숙한 부분들을 보완해야 하겠다는 결심을 하게 되었다. | |

# 소스코드(별첨)

개발한 소스코드를 다음의 폴더구조(샘플과 같은 스타일)에 와 같이 저장하고 완료보고서와 함께 제출합니다.

* 등록된 모든 데이터베이스

**P\_실무프로젝트--+- Team\_1\_뽀끔뽀끔- |- App – *files***

**| |- Server – *files***

**| |- Arduino – *files***

**|**

**+- Database --- test.mv.db**

**|**

**+ 완료보고서(docx) – *1팀-완료보고서v1.0-권민혁-장정우-이호왕-박재균.docx***

**+ 완료발표자료(pptx) – *1팀-완료발표v1.0-권민혁-장정우-이호왕-박재균.pptx***

***Github Link :*** [***https://github.com/Danny-Kwon/SmartFishBowl.git***](https://github.com/Danny-Kwon/SmartFishBowl.git)

**\* 완료발표자료는 ppt 파일로 작성하되, 제시된 템플릿이나 형식은 없으나,**

**각 팀마다 완성도 있는 템플릿을 스스로 만들어서 작성할 것. (발표효과 극대화)**