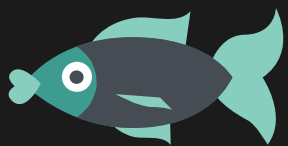


원격 어항 관리 시스템  
Remote Fishbowl Management System  
FishBerry

2014  
2014  
2014



# 목차

- 1. 종합 설계 개요 (p.3-8)
- 2. 관련 연구 및 사례 (p.9-12)
- 3. 시스템 수행 시나리오 (p.13-17)
- 4. 시스템 구성도 (p.18-20)
- 5. 시스템 모듈 상세 설계 (p.21-54)
- 6. 개발 환경 및 개발 방법 (p.55-65)
- 7. 종합설계 수행 일정 (p.66)
- 8. 데모 환경 설계(p.67)
- 9. 업무 분담, 종합설계일정(p.68-69)
- 10. 필요기술 및 참고문헌, GitHub (p.70-71)



# 1. 종합 설계 개요

## 지적 사항

번호	지난 발표에서의 지적 사항	지적 사항에 대한 답변	관련 페이지
1	기능이 추가되면(산소량 조절) 좋은 작품이 되리라 보임	<ul style="list-style-type: none"> <li>산소량 조절의 경우 용존산소량 센서의 가격이 비싸고,(약 20만원) 산소 공급은 여과기가 대신 해주기 때문에 고려X</li> <li>다른 추가기능으로 생각한 것               <ol style="list-style-type: none"> <li>온도, pH, 먹이급여날짜, 환수날짜 등의 데이터를 그래프로 시각화해서 보여주고, 데이터를 종합하여 언제 환수하는게 좋을지 추천하는 기능</li> <li>라즈베리 관련 리소스를 사용자에게 알려주어 기기오류원인을 찾아주고, 버튼클릭으로 사용자로 하여금 간단히 해결할 수 있게 하는 기능</li> </ol> </li> </ul>	p.4,5
2	하드웨어 설계를 잘할것, 다른 방식으로 주제를 살리는 거나 어항관리의 목표를 명확히 할 것	어항 관리의 목표: 어항 관리에 대한 사용자의 부담을 줄여주고, 외출상황에서도 원격으로 어항상태를 확인하고 관리할 수 있도록 도와주는 것	p.7



# 1. 종합 설계 개요

## 추가 기능-환수 날짜 추천

SaveTime	Temperature	PH
2019/4/13/23/41	24.2	8.30
2019/4/14/14/38	23.5	7.40
2019/4/14/15/8	23.5	7.42
2019/4/14/15/38	23.7	7.30
2019/4/14/16/8	24.2	7.47
2019/4/14/16/38	24.3	7.51
2019/4/14/17/8	24.3	7.57
2019/4/14/17/38	24.1	7.90
2019/4/14/18/8	24.1	7.83
2019/4/14/18/38	24.3	7.78
2019/4/14/19/8	24.1	7.71
2019/4/14/19/38	24.2	7.64
2019/4/14/20/8	24.1	7.63
2019/4/14/20/38	24.2	7.59
2019/4/14/21/8	24.1	7.56
2019/4/14/21/38	24.2	7.45
2019/4/14/22/8	24.2	7.54

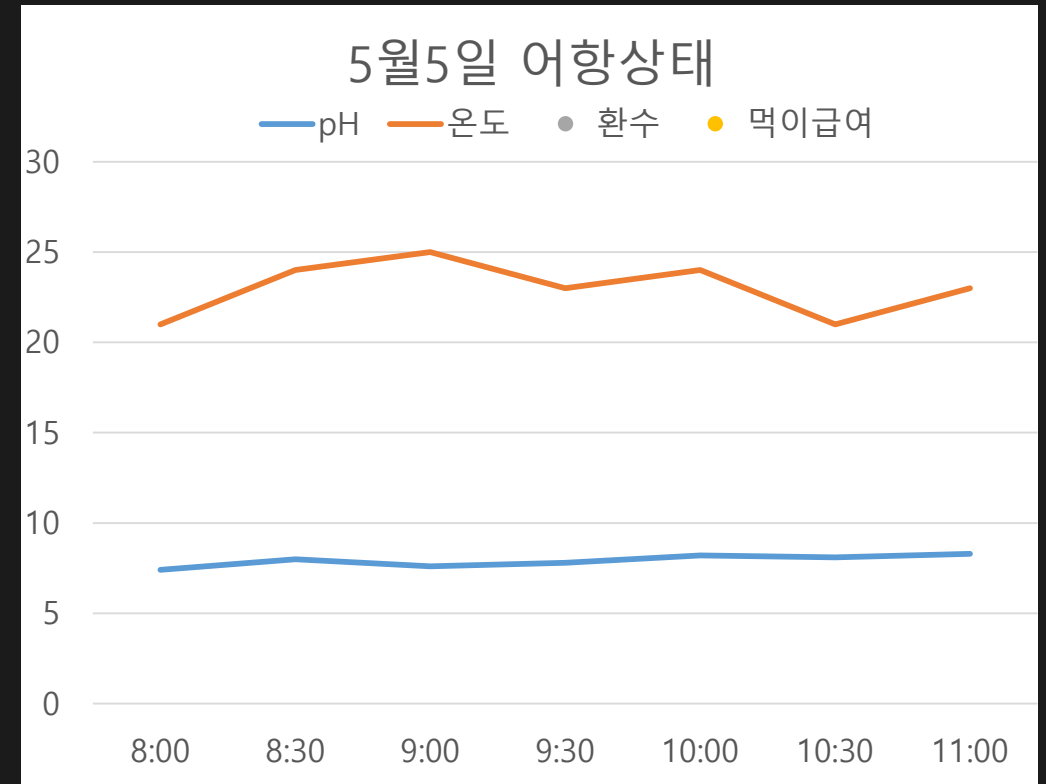
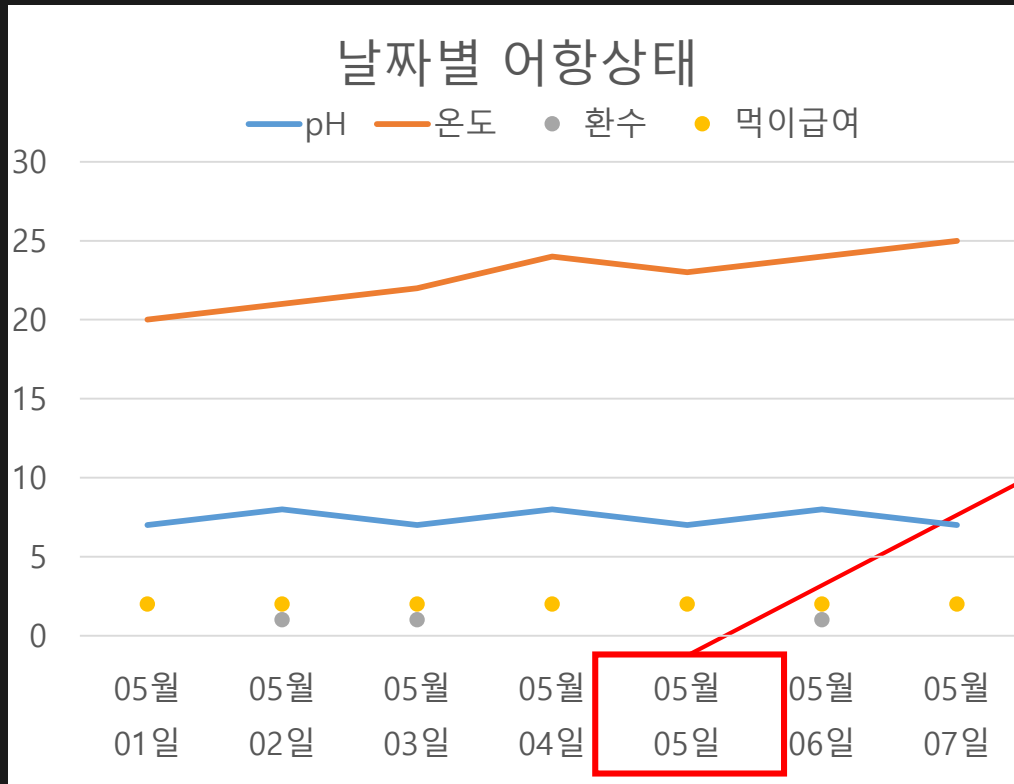
2019/4/17/9/6	23.9	7.90
2019/4/17/9/36	24.5	7.97
2019/4/17/10/6	24.6	8.04
2019/4/17/10/36	24.8	8.10
2019/4/17/11/6	24.7	8.12
2019/4/17/11/36	24.8	8.19
2019/4/17/12/6	24.8	8.24
2019/4/17/12/59	25.0	8.26
2019/4/17/13/29	24.6	8.38
2019/4/17/13/59	24.8	8.33
2019/4/17/14/29	24.6	8.31
2019/4/17/14/59	24.5	8.28
2019/4/17/15/29	24.5	8.28
2019/4/17/15/59	24.3	8.26
2019/4/17/16/29	24.3	8.26
2019/4/17/16/59	24.3	8.26
2019/4/17/17/29	24.3	8.24
2019/4/17/17/59	24.3	8.21
2019/4/17/18/29	24.2	8.22
2019/4/17/18/59	24.4	8.21

1. 30분 단위로 pH값을 저장
2. 일주일의 평균pH를 구함
3. 이후 평균pH에서 크게 차이 나는 날짜에 환수를 추천



# 1. 종합 설계 개요

## 추가 기능-시각화





# 1. 종합 설계 개요

## 개발 배경

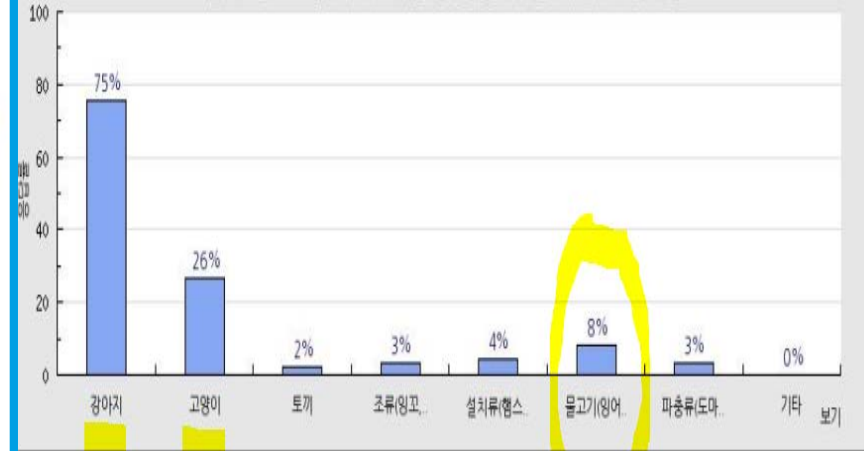
- ✓ 물고기 : 개, 고양이에 다음으로 가장 많이 키우는 반려동물 3위
- ✓ 개인사정으로 인해 장기간 어항을 돌보지 못하는 상황 대비
- ✓ 분양, 판매 등 재테크 수단으로도 이용되고 있음
- ✓ 키우기 까다로운 어종을 개인의 부주의로 인해 잃게 되는 사례
- ✓ 매번 번거롭게 어항의 상태를 확인해야 하는 불편함
- ✓ 어항의 상태를 자동으로 파악해 사용자에게 알려줄 필요성

## <반려동물 양육상 어려운 점>

(복수응답, 단위:%)



[반려동물 보유자] 현재 키우고 있는 반려동물은 무엇입니까? (복수응답 가능)





# 1. 종합 설계 개요

## 개발 목표

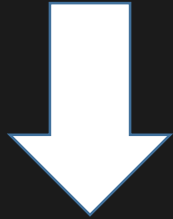
- ✓ 어항의 수질 관리를 위해 필요한 수온과 오염도를 체크해 위험상황에 LED표시와 사용자 스마트폰으로 알림
- ✓ 사용자가 먹이를 직접 줄 수 없는 상황에 먹이를 자동으로 줄 수 있는 자동먹이지급 기능
- ✓ 사용자의 편의를 위한 자동 부분 환수(어항 물갈이) 기능
- ✓ 어항 스트리밍을 통해 원격으로 어항 상태를 수시로 확인 가능
- ✓ 위 기능들을 안드로이드 앱을 통해 제어 가능



# 1. 종합 설계 개요

## 개발 효과

- ✓ 사용자들에게 다양한 편의 기능을 제공
- ✓ 어항 환경에 대한 실시간 모니터링 시스템 제공
- ✓ 어항을 원격으로 관리할 수 있게 됨으로써 사용자들이 자리를 비울 수 없는 불편함 해소
- ✓ 초보자에게 물고기를 쉽게 기를 수 있는 환경 제공



사용자 편의성 향상





## 2. 관련 연구 및 사례

### ■ FiSHBiT



- ✓ 스마트폰 앱을 통해 어항 상태 확인 가능
- ✓ 히터, 펌프, LED 원격으로 제어 가능
- ✓ 작은 기기를 어항 안에 넣어 작동하기 때문에 외관 상 깔끔
- ✓ 국내 발매가 안됐기 때문에 해외배송 해야 함
- ✓ 비싼 가격 (하위 버전-\$425 / 상위 버전-\$600)



## 2. 관련 연구 및 사례

### ■ LED 미니 인테리어 무소음 스마트 어항



- ✓ 오픈마켓에서 판매중인 LED스마트 어항
- ✓ 달력,온도 체크,단순LED가능
- ✓ 어항 크기가 정해져 있어서 다양한 환경 제공 불가능
- ✓ 스마트폰으로 어항 관리 불가능



## 2. 관련 연구 및 사례

### ■ 블루네로



- ✓ 해외마켓에서 판매중인 스마트 어항
- ✓ 스트리밍, 온도/오염도 체크, 스마트폰 제어 가능
- ✓ 다양한 기능에 비해 어항관리 이외의 필요 없는 기능(ex 스마트폰 충전)들이 존재함
- ✓ 가격은 899달러로 상당히 높음
- ✓ 크기는 60L, 100L로 소형사이즈가 없음

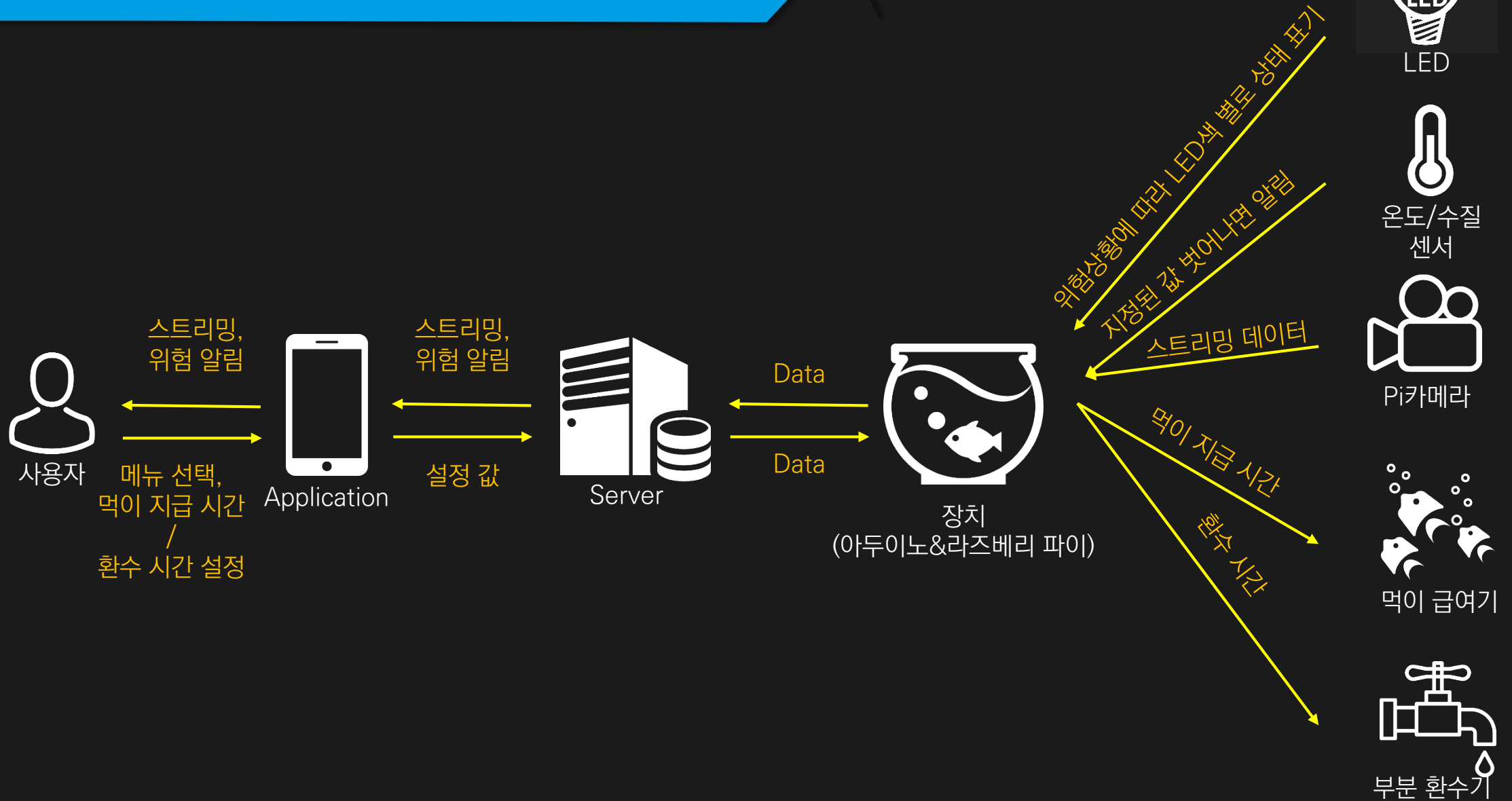


## 2. 관련 연구 및 사례

사례명	온도, 오염도 측정	부분 환수	자동 먹이급여	영상 스트리밍	스마트폰 관리	LED 상태표기
블루네로	O	X	O	O	O	X
FiSHBiT	O	X	X	X	O	X
스마트 어항	온도만	X	X	X	X	X
FishBerry	O	O	O	O	O	O



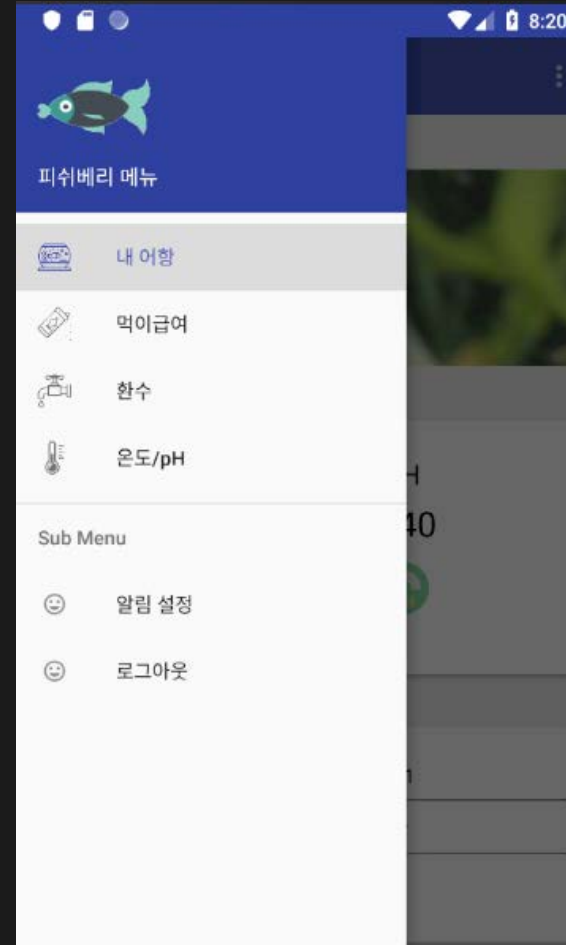
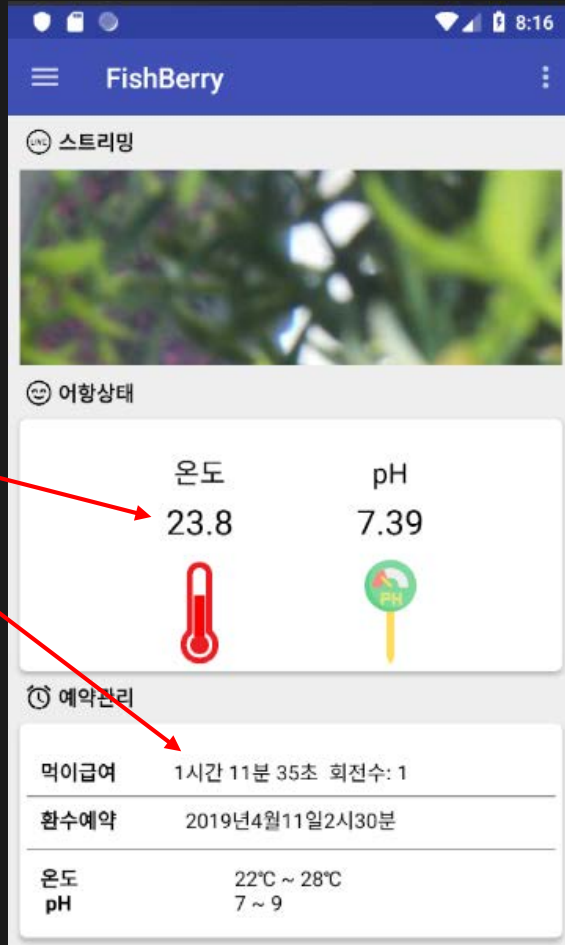
### 3. 시스템 수행 시나리오





### 3. 시스템 수행 시나리오

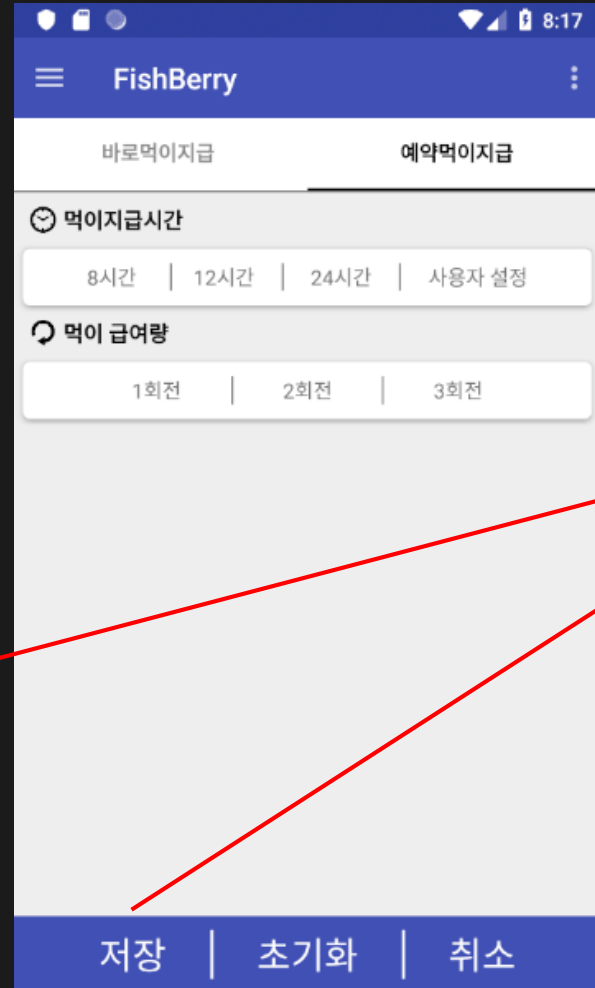
#### App-메인 및 메뉴





### 3. 시스템 수행 시나리오

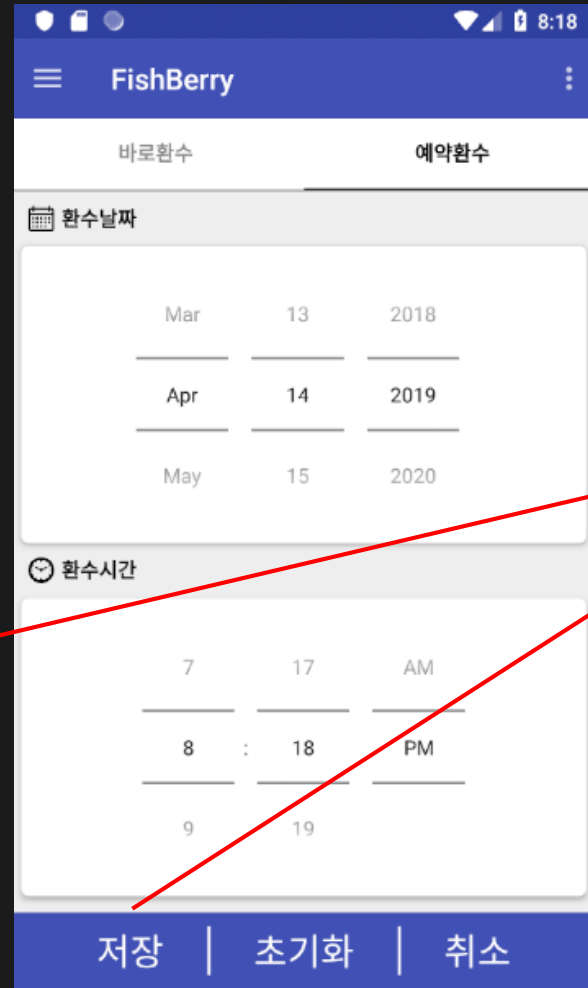
#### App-먹이급여





### 3. 시스템 수행 시나리오

#### App-환수







### 3. 시스템 수행 시나리오

#### App-경고범위 설정

FishBerry

수온 경고범위

\_\_\_ °C ~ \_\_\_ °C

pH 경고범위

\_\_\_ ~ \_\_\_

저장 | 초기화 | 취소



알림 켜기/끄기

온도 경고 받기

수질 경고 받기

온도 경고 주기 시간 55 초

수질 경고 주기 시간 22 초

저장 | 초기화 | 취소

Android 100% 8:43

Sun, Apr 14

FishBerry

온도 경고!

설정한 온도를 벗어났습니다! / 현재 온도 : 24.0

Google Play Store · 4d

Google Play Protect is turned on

This feature protects your device from harmful apps

FishBerry

FishBerry is running

Tap for more information or to stop the app.

Android System

Virtual SD card

For transferring photos and media

Android System

USB debugging connected

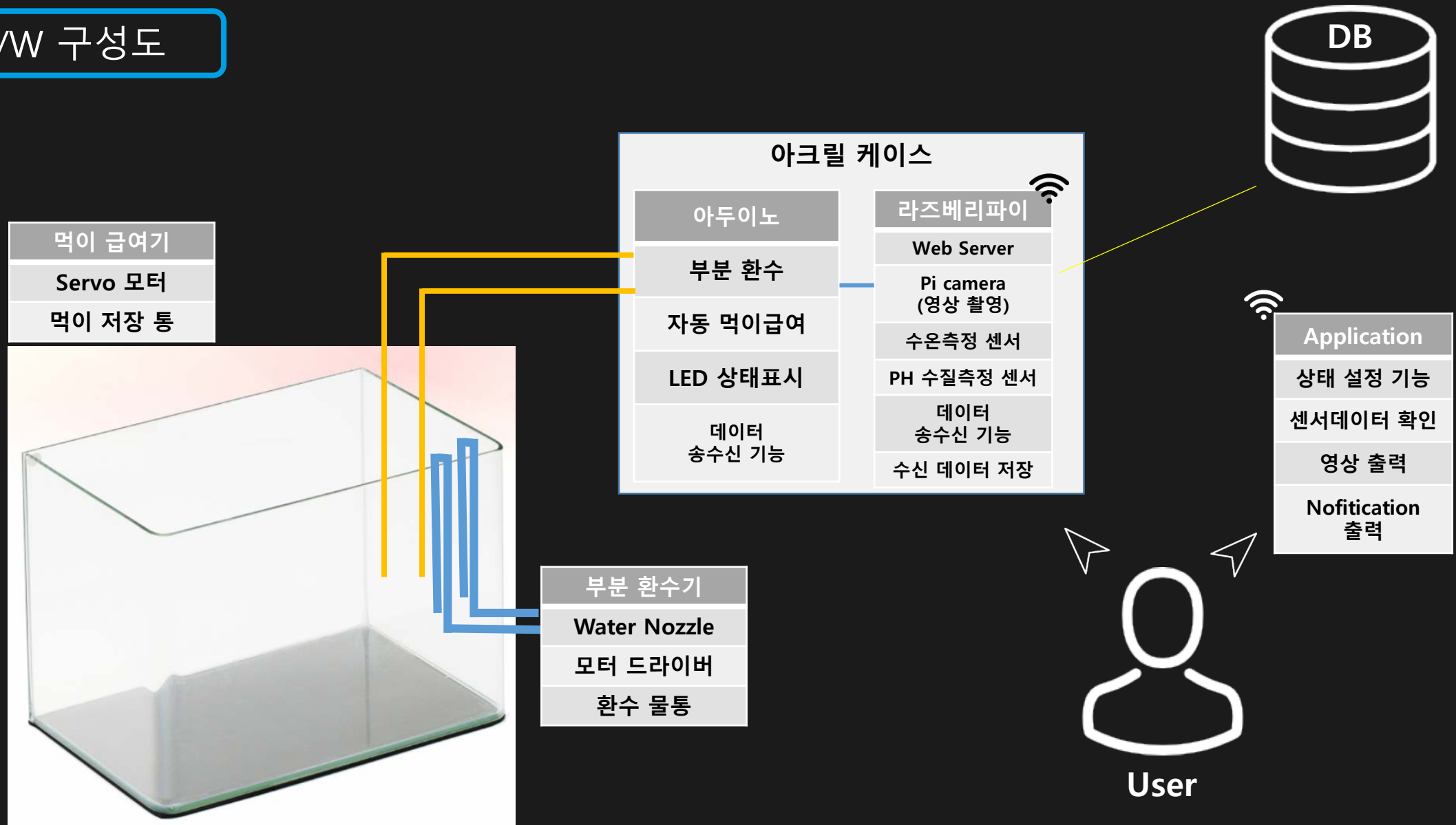
Tap to disable USB debugging.

Android System · USB charging this device



## 4. 시스템 구성도

### H/W 구성도

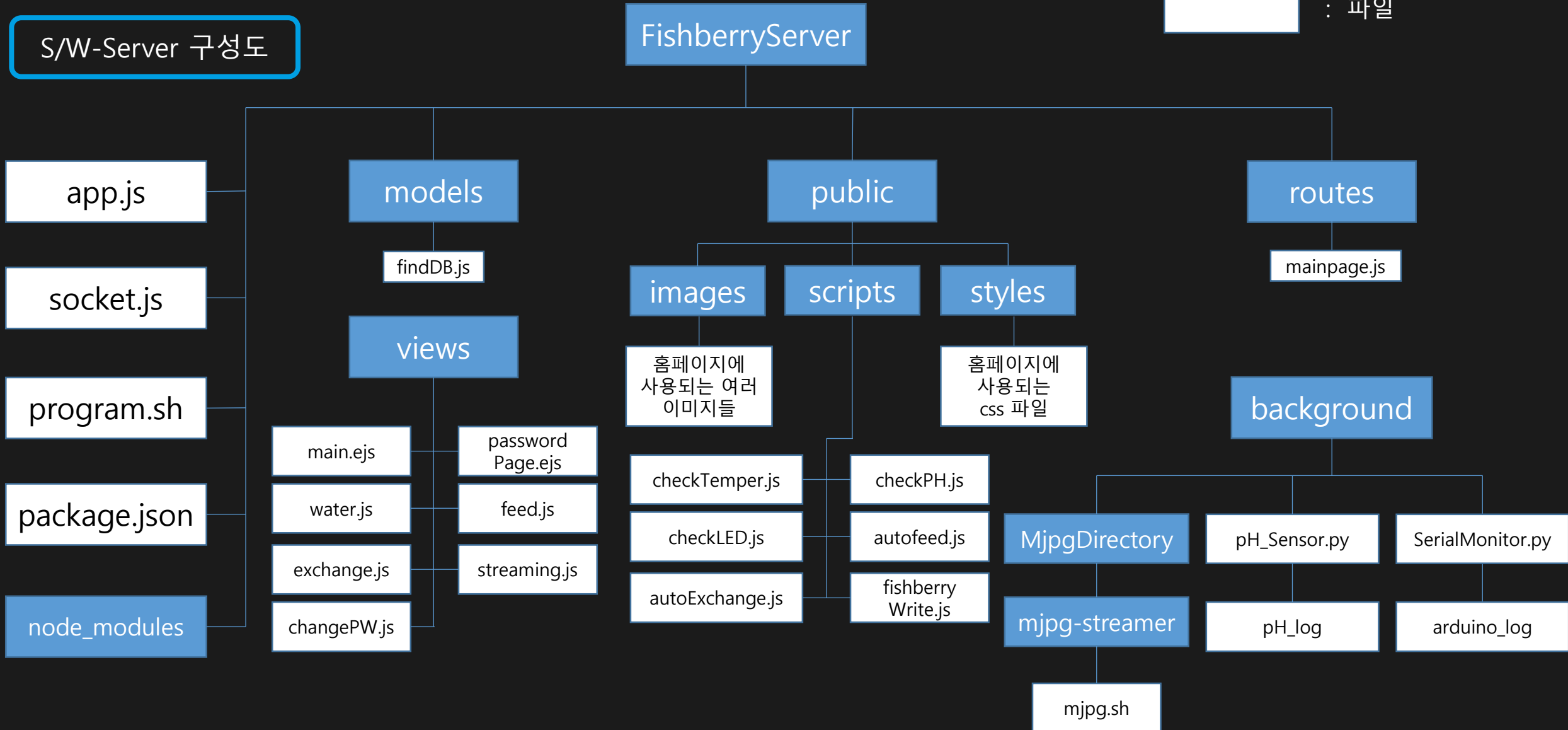




## 4. 시스템 구성도

■ : 디렉토리  
□ : 파일

S/W-Server 구성도

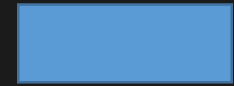




## 4. 시스템 구성도

### S/W-App 구성도

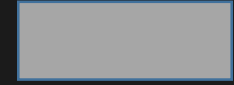
FishberryApp



: 패키지



: 클래스



: 인터페이스

BaseActivity

IntentData

MainActivity

RegisterActivity

ManualActivity

SettingActivity

SplashActivity

DB

DBHelper

DBElement

Feed

FeedFragActivity

FeedFrag  
Presenter

FeedFrag  
PresenterImpl

FeedNow  
Fragment

FeedReserve  
Fragment

FeedUser  
SettingTimer

Login

LoginActivity

Login  
Presenter

Login  
PresenterImpl

Notification  
Service

BootReceiver

Notification  
Service

RestartService

Notification  
Setting

Notification  
SettingService

Notification  
Setting  
Presenter

Notification  
Setting  
PresenterImpl

Streaming

Streaming  
Activity

Streaming  
Presenter

Streaming  
PresenterImpl

FullStreaming  
Activity

Temperature

Temperature  
Activity

Temperature  
Presenter

Temperature  
PresenterImpl

Water

WaterFrag  
Activity

WaterFrag  
Presenter

WaterFrag  
PresenterImpl

WaterNow  
Fragment

WaterReserve  
Fragment



## 5. 시스템 모듈 상세설계

### 서버 Database

#### TemperSetting

min 최소수온경고값  
max 최대수온경고값

#### TemperValue

Tvalue 현재 수온값

#### PHSetting

min 최소수질경고값  
max 최대수질경고값

#### PHValue

Pvalue 현재 수질값

#### FeedSetting

timer 먹이지급타이머  
circle 먹이지급 회전량  
save\_time 사용자가  
저장한 지급 반복시간

#### ExchangeSetting

exTimer1 환수 전반부 남은시간  
exTimer2 환수 후반부 남은시간  
exTime\_save 환수 예약시간  
isChanged 환수시작 플래그  
totalPercent 환수 총 진행률

#### DataRepository

Temperature 어느 시점의 수온값  
PH 어느 시점의 수질값  
SaveTime 30분 간격 측정 시간값

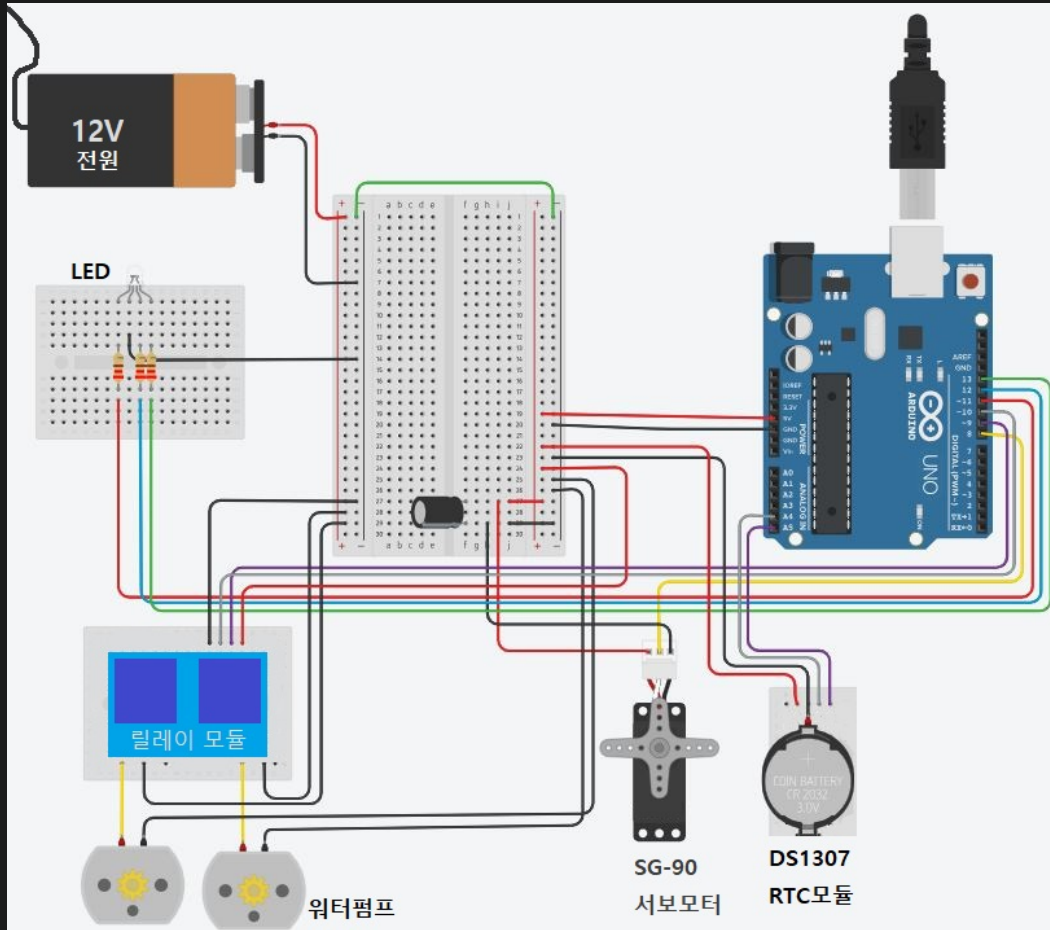
#### passwordSetting

password 접속 비밀번호

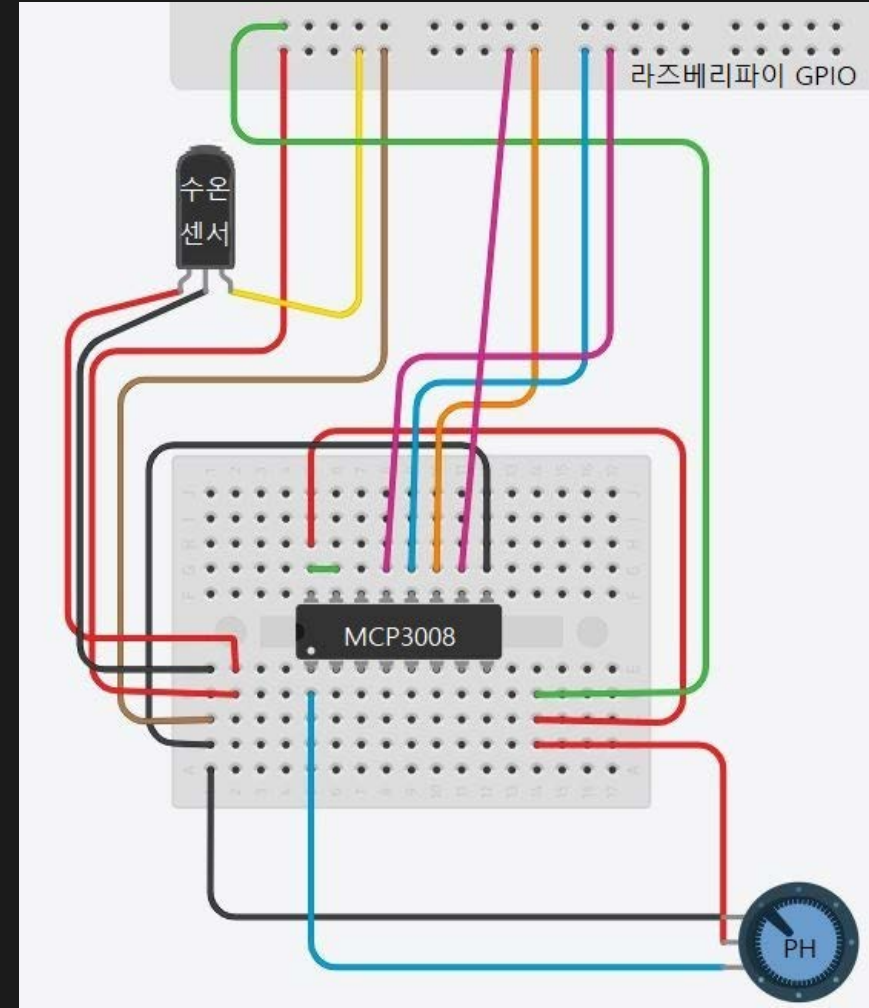


## 5. 시스템 모듈 상세설계

### 회로도



아두이노



라즈베리파이



## 5. 시스템 모듈 상세설계

### DB 관련 모듈 – Server

/models/findDB.js

형식	function insertTemper(min, max)	형식	function insertTvalue(Tvalue)
설명	사용자가 지정한 경고온도 저장	설명	어느 시점의 현재 수온을 저장
형식	function insertPH(min, max)	형식	function insertPvalue(Tvalue)
설명	사용자가 지정한 경고 수질 값 저장	설명	어느 시점의 현재 수질 값을 저장
형식	function insertFeed(timer, circle, save_time)	형식	function insertExchange(t1, t2, save_time)
설명	사용자가 지정한 먹이지급 시 반복시간 값 및 반복회전량을 저장	설명	사용자가 지정한 부분환수 예약시간을 저장
형식	function confirmPassword(password)	함수선언	const db = require('./models/findDB');
설명	사용자가 입력한 비밀번호와 DB에 저장된 비밀번호 비교	사용예시	db.insertTemper(min, max);



## 5. 시스템 모듈 상세설계

### 수온 측정 모듈 - Server(1/6)

#### startCheckTemper 클래스

##### 기능

- 어항의 내부 수온을 측정
- 측정한 온도는 경고수온이 비교하는 대상 값

##### 다루는 정보

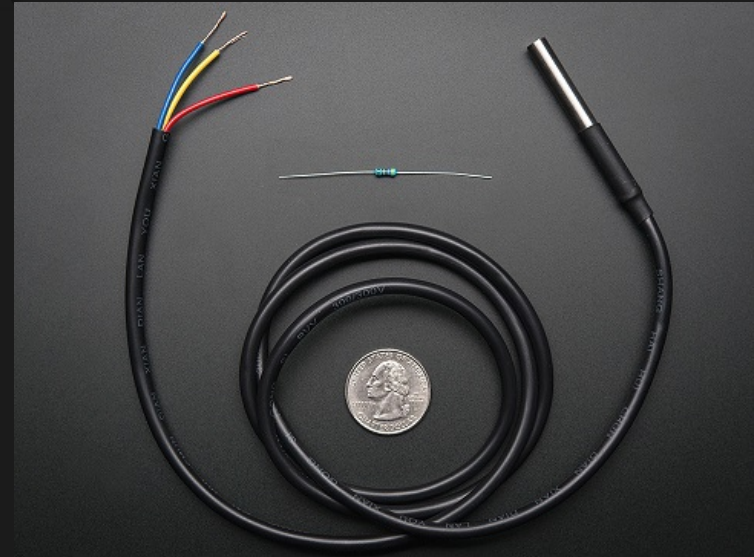
fs모듈을 통해 센서 시스템으로 접근한다.

temperature : 측정한 수온값의 저장변수

temperFullData : 센서가 측정한 전체 데이터. 이 값에서 수온을 추출해서 저장한다.

##### 수온 측정값을 App에 반영하는 시점

setInterval 메서드를 통해 5초에 한번씩 DB에 측정 값을 저장하고, 저장한 값을 App에 socket.io를 통해서 전송한다.



#### 😊 어항상태

온도  
→ 22.9



pH  
8.32







## 5. 시스템 모듈 상세설계

### 수온 측정 모듈 - Server(1/6)

#### startCheckTemper()

형식	function startCheckTemper(fs)
리턴 값	없음
설명	센서로부터 수온값을 받아와서 DB에 저장
예시	checkTemper.startCheckTemper(fs);

#### startCheckTemper() - 데이터 구조

데이터	자료형	설명
fs	FileSystem	파일에 접근을 위해 사용하는 모듈
temperature	float	수온 측정 저장값 (first + '.' + second)
temperFullData	String	센서의 전체 데이터
first	String	Data중 수온값의 정수값 자리
second	String	Data중 수온값의 소수값 자리



## 5. 시스템 모듈 상세설계

### 수질 측정 모듈 - Server(2/6)

#### startCheckPH 클래스

##### 기능

- 어항의 내부 수질을 측정
- 측정한 pH는 경고수질이 비교하는 대상 값

##### 다루는 정보

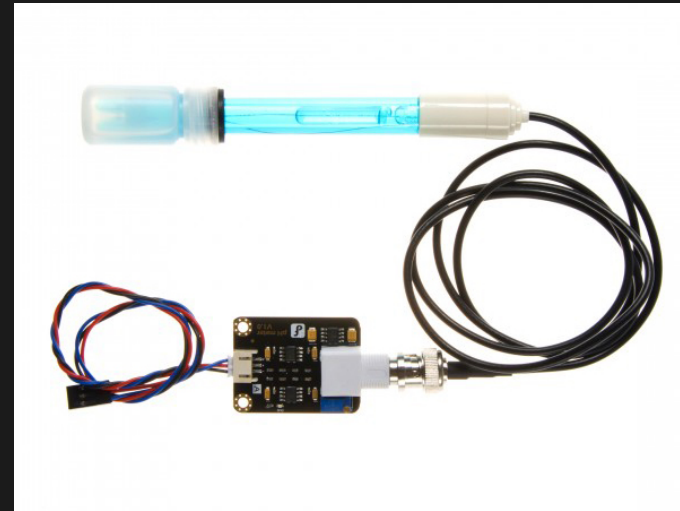
fs모듈을 통해 센서 시스템으로 접근한다.

phValue : 측정한 수온값의 저장변수

phFullData : 센서가 측정한 전체 데이터. 이 값에서 수질을 추출해서 저장한다.

##### 수질 측정값을 App에 반영하는 시점

setInterval 메서드를 통해 5초에 한번씩 DB에 측정 값을 저장하고, 저장한 값을 App에 socket.io를 통해서 전송한다.



#### 😊 어항상태

온도

22.9



pH

→ 8.32





## 5. 시스템 모듈 상세설계

### 수질 측정 모듈 - Server(2/6)

#### startCheckPH()

형식	function startCheckPHfs)
리턴 값	없음
설명	센서로부터 수질값을 받아와서 DB에 저장
예시	checkPH.startCheckPH(fs);

#### startCheckPH() – 데이터 구조

데이터	자료형	설명
fs	FileSystem	파일에 접근을 위해 사용하는 모듈
phValue	float	수질 측정 저장값
phFullData	String	센서의 전체 데이터



## 5. 시스템 모듈 상세설계

### LED 경고 모듈 – Server(3/6)

#### startCheckLED 클래스

##### 기능

- 수온과 수질의 변화에 따른 경고를 LED로 표시
- 정상, 수온경고, 수질경고, 수온&수질 경고

##### 다루는 정보

측정된 수온, 수질 값들을 변수로 받아온다.  
받아온 측정값과 최소경고온도, 최대경고온도 / 최소경고수질값,  
최대경고수질값을 비교하여 각 상황에 맞는 LED를 점등한다.

##### 경고 LED값을 아두이노에 반영하는 시점

setInterval 메서드를 통해 5초에 한번씩 아두이노로 경고 메시지를 보낸다. 아두이노는 메시지에 반응하여 LED를 점등하게 된다.





## 5. 시스템 모듈 상세설계

### LED 경고 모듈 – Server(3/6)

#### startCheckLED()

형식	function startCheckLED()
리턴 값	없음
설명	센서값과 최소/최대 경고수치를 비교하여 아두이노로 경고메세지를 보낸다.
예시	checkLED(tty, fs, temperature, pHValue, maxTemper, minTemper, maxPH, minPH);

#### startCheckLED() – 데이터 구조

데이터	자료형	설명
fs	FileSystem	파일에 접근을 위해 사용하는 모듈
tty	String	아두이노 연결포트
temperature	float	수온 측정값
pHValue	float	수질 측정값
maxTemper	float	최대경고수온
minTemper	float	최소경고수온
maxPH	float	최대경고수질
minPH	float	최소경고수질



## 5. 시스템 모듈 상세설계

### LED 경고 모듈과 RTC 측정 모듈 – Arduino

#### setLED()

형식	void setLED(int mode)
리턴 값	없음
설명	서버로부터 경고 메시지를 받은 후 상황에 따라 각 LED를 점등한다.
예시	<pre>if(nodeData == "TempWN") { setLED(1); }</pre>

#### getTime()

형식	void getTime()
리턴 값	없음
설명	아두이노의 RTC 시간을 측정
예시	<pre>loop() 내에 getTime()을 넣으면 시리얼 모니터에 시간을 찍는다.</pre>



## 5. 시스템 모듈 상세설계

### 자동 먹이지급 모듈 – Server(4/6)

#### startAutoFeed 클래스

##### 기능

- 사용자가 지정한 시간(ex 8시간)을 반복하여 자동으로 어항에 먹이를 지급하게 해준다.

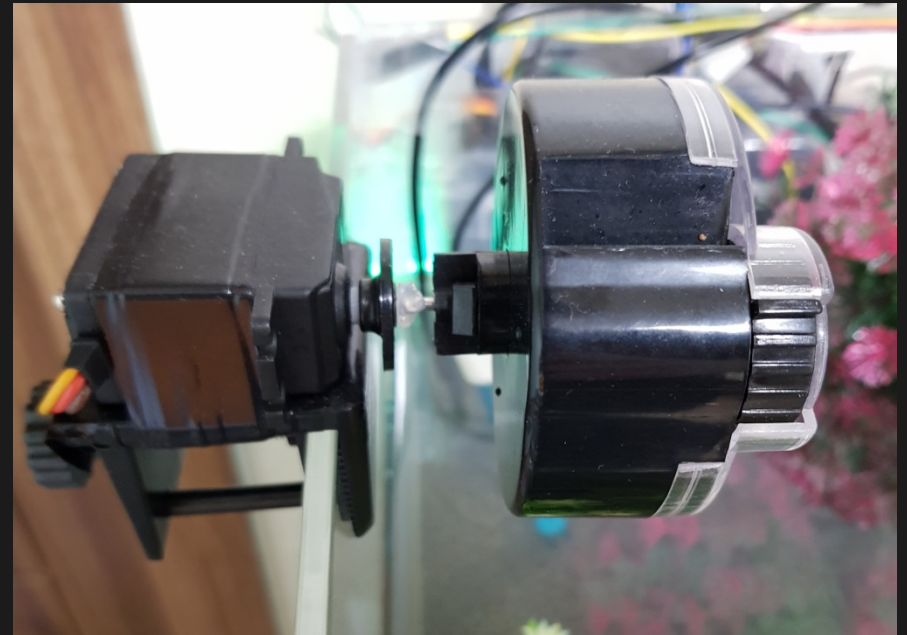
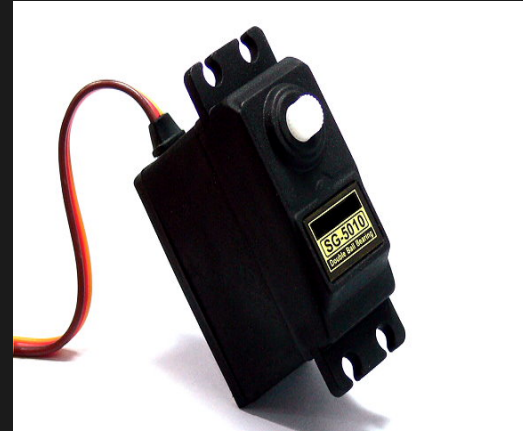
##### 다루는 정보

servoTimer : DB로부터 타이머를 받아와서 1초에 한번씩 타이머 값을 1 빼준 후 다시 DB에 update한다.

servoCircle : 먹이지급 시 먹이통이 몇 번 회전할지에 대한 값이다.

##### 먹이지급 신호를 아두이노에 반영하는 시점

setInterval 메서드를 통해 1초에 한번씩 타이머가 감소하여, 타이머가 0이 되면 아두이노로 먹이지급 메시지를 보내게 된다.





## 5. 시스템 모듈 상세설계

### 자동 먹이지급 모듈 – Server(4/6)

#### startAutoFeed()

형식	function startAutoFeed()
리턴 값	없음
설명	1초에 한번 지정한 타이머가 감소하며, 타이머가 0이 되면 자동으로 먹이지급
예시	autoFeed(connection, tty, fs);

#### startAutoFeed() – 데이터 구조

데이터	자료형	설명
fs	FileSystem	파일에 접근을 위해 사용하는 모듈
tty	String	아두이노 연결포트
connection	mysql.createConnection	mysql에 접속하기 위한 connection
servoTimer	int	먹이지급 타이머
servoCircle	int	먹이지급 반복량
getHour	int	타이머가 millisecond형식이라 hour형식으로 변환
getMinute	int	타이머를 minute으로 변환
getSecond	int	타이머를 second으로 변환





## 5. 시스템 모듈 상세설계

### 자동 부분환수 모듈 – Server(5/6)

#### startAutoExchange 클래스

##### 기능

- 어항의 부분환수를 자동으로 해줘서 환수의 번거로움을 해결한다.
- 즉시 자동환수 및 예약시간 부분환수 기능이 있다.

##### 다루는 정보

totalPercent : 부분환수의 총 진행률. 0%~100%로 진행함을 표시  
isChanged : 부분환수를 진행하고 싶으면 true로 값을 변경해주며,  
일시정지 혹은 환수완료 시 값을 false로 바꿔준다.

##### 부분환수 신호를 아두이노에 반영하는 시점

아두이노 RTC와 서버에 저장한 사용자의 예약시간을 비교하여,  
지정한 예약시간이 되면 서버에서 아두이노로 환수시작 메시지를 보낸다.  
또는 즉시 부분환수 진행을 위해 App에서 버튼을 클릭하면 아두이노로 환수시작 메시지를 보낸다.





## 5. 시스템 모듈 상세설계

### 자동 부분환수 모듈 – Server(5/6)

#### startAutoExchange()

형식	function startAutoExchange()
리턴 값	없음
설명	1초에 한번 아두이노의 RTC 시간과 사용자가 지정한 예약시간을 비교하여 설정한 시간이 되면 자동으로 부분환수를 진행.
예시	autoExchange(connection, tty, fs);

#### startAutoExchange() – 데이터 구조

데이터	자료형	설명
fs	FileSystem	파일에 접근을 위해 사용하는 모듈
tty	String	아두이노 연결포트
exTime_save	String	사용자의 환수 예약시간
isChanged	boolean	환수를 진행하려면 값을 true로 변경
waterTimer1	int	환수의 전반부 진행률
waterTimer2	int	환수의 후반부 진행률
arduinoTime	String	아두이노 RTC 시간값
totalPercent	int	환수의 총 진행률. 이 값을 App으로 전송한다



## 5. 시스템 모듈 상세설계

### 공통 파일시스템 모듈 – Server(6/6)

#### fishberryWrite.input()

형식	function input()
리턴 값	에러 시 로그기록, 성공 시 그대로 진행
설명	서버에서 아두이노의 파일시스템으로 접근하여 메시지를 전송한다.
예시	fishberryWrite.input(fd, data, msg);

#### fishberryWrite.input() – 데이터 구조

데이터	자료형	설명
fd	File Descriptor	아두이노 파일 시스템의 디스크립터
data	String	아두이노로 보내는 메시지 데이터
msg	String	아두이노로 메시지를 보낸 후 서버의 log표시용 String 데이터



## 5. 시스템 모듈 상세설계

### 부분환수, 먹이지급 모듈 – Arduino

#### operateMotor()

형식	void operateMotor(char sw)
리턴 값	없음
설명	서버의 메시지에 따라 수중모터가 동작
예시	<pre>if(nodeData == "StartIN") { operateMotor('a'); }</pre>

#### operateServo()

형식	void operateServo(int circle)
리턴 값	없음
설명	서버의 메시지에 따라 먹이지급을 위한 서보모터가 동작
예시	<pre>if(nodeData == "StartServo1") { operateServo('1'); }</pre>



## 5. 시스템 모듈 상세설계

### 로그인 모듈 - App(1/6)

#### Login 모듈

##### 기능

- 체크박스 활성화 이후 어플 재기동 시, 로그인 여부 없이 로그인 가능.
- 비밀번호 확인 후, 웹 서버의 비밀번호와 일치한다면, 로그인.

##### 다루는 정보

웹 서버와의 통신 소켓 : Socket(io.socket.client) socket

IP, 비밀번호 : EditText ipEdit, EditText pwEdit

DBElement (DB의 IP, IP 기억 여부, 알림 설정 등 사용자의 정보를 담고 있는 내부 DB의 데이터 클래스)

IntentData : 사용자의 소켓과 IP주소를 저장하는 싱글톤 클래스

checkBox : 사용자의 IP 기억 여부 체크박스

context : 서비스 실행 및 intent 실행을 위한 context 변수

##### 로그인을 하는 시점


아래 중 하나의 조건을 완료 시


1. 로그인 버튼 클릭 시, 해당 IP에 부합되는 비밀번호 적합
2. 체크박스 적용 -> 로그인 성공 -> 어플 재기동

로그인

IP : fishberry.iptime.org

비밀번호 : \_\_\_\_\_

 ☐ IP 기억



 로그인

v1.0.0

로그인

IP : fishberry.iptime.org

비밀번호 : \_\_\_\_\_

  로그인

v1.0.0



## 5. 시스템 모듈 상세설계

### 로그인 모듈 - App(1/6)

rememberIP()	
형식	public void rememberIP(DBElement dbElement, Socket socket, IntentData intentData, Context context)
리턴 값	없음
설명	'IP 기억' 체크박스의 활성화 여부 판단 후, 자동 로그인 여부 결정.
예시	loginPresenterimpl.rememberIP(dbElement, socket, intentData, this);

rememberIP() - 데이터 구조		
데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓
dbElement	DBElement	안드로이드 내부 DB의 데이터 객체
intentData	IntentData	소켓정보와 IP를 담기 위한 싱글톤 클래스 객체
context	Context	서비스 실행 및 intent 실행을 위한 context



## 5. 시스템 모듈 상세설계

### 로그인 모듈 - App(1/6)

#### connectServer()

<b>형식</b>	public void connectServer(EditText ipEdit, EditText pwEdit, CheckBox checkBox, Context context)
<b>리턴 값</b>	없음
<b>설명</b>	사용자가 입력한 IP와 비밀번호를 이용해 로그인 시도.
<b>예시</b>	loginPresenterimpl.connectServer(ipEdit, pwEdit, checkBox, this);

#### connectServer() – 데이터 구조

데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓
ipEdit	EditText	IP의 정보가 있는 EditText
pwEdit	EditText	비밀번호의 정보가 있는 EditText
checkBox	CheckBox	내부 DB에 'IP 기억' 확인 여부를 저장하기 위한 CheckBox
context	Context	intent 실행을 위한 context



## 5. 시스템 모듈 상세설계

### 먹이 급여 모듈 - App(2/6)

#### Feed 모듈

##### 기능

- 어항의 먹이 급여기를 1회 바로 회전
- 사용자가 선택한 먹이 시간과 먹이 회전 수를 서버에 저장.

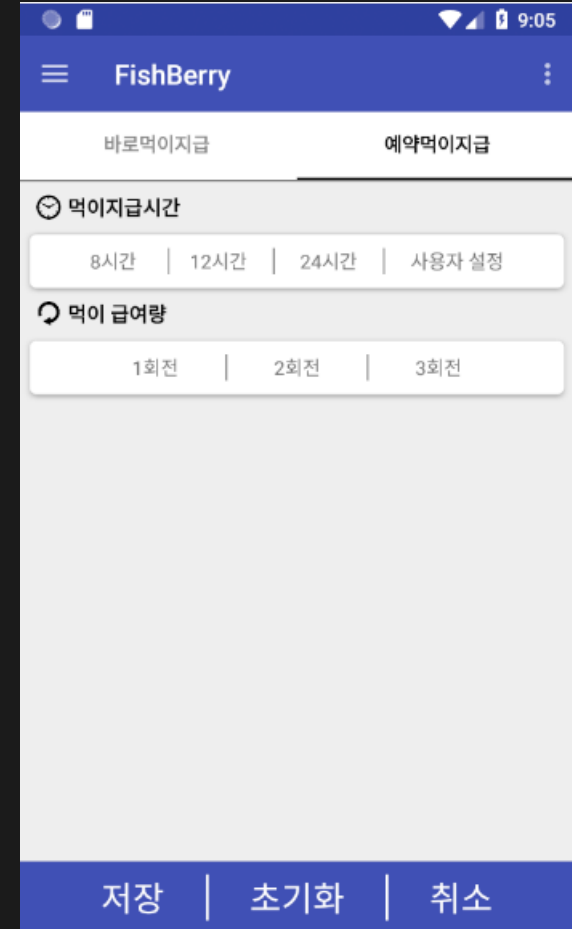
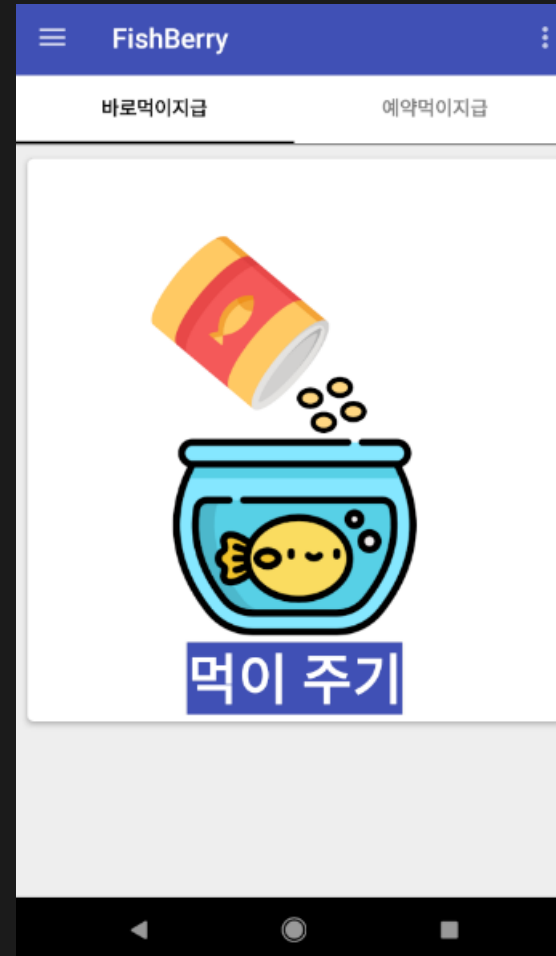
##### 다루는 정보

웹 서버와의 통신 소켓 : Socket(io.socket.client) socket  
'예약먹이지급' - 먹이지급시간 : int timerFeed  
'예약먹이지급' - '먹이 급여량' : int circleFeed

##### 정보 전달을 반영하는 시점

아래 두 조건 만족 시

1. 웹 서버가 현재 socket.io 모듈로 listen하고 있는 시점
2. App에서 '먹이 주기' 혹은 '저장' 버튼을 눌렀을 때







## 5. 시스템 모듈 상세설계

### 먹이 급여 모듈 - App(2/6)

#### startFeed()

형식	public void startFeed(Socket socket)
리턴 값	없음
설명	서버로 먹이 급여기 1회 회전 명령을 전달
예시	feedFragPresenterImpl.startFeed(socket)

#### saveFeed()

형식	public void saveFeed(Socket socket, int timerFeed, int circleFeed)
리턴 값	없음
설명	서버로 먹이 급여 시간과 먹이 급여기 회전 수를 서버로 전달
예시	feedFragPresenterimpl.saveFeed(socket, timerFeed, circleFeed);

#### startFeed() – 데이터 구조

데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓

#### saveFeed() – 데이터 구조

데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓
timerFeed	int	먹이 급여 시간
circleFeed	Int	먹이 급여기 회전 수



## 5. 시스템 모듈 상세설계

### 알림 모듈 - App(3/6)

#### Notification 모듈

##### 기능

- 알림 여부 설정
- 온도 경고 여부 설정 및 온도 경고 알림 주기 시간 설정
- 수질 경고 여부 설정 및 수질 경고 알림 주기 시간 설정
- 알림 설정 저장.

##### 다루는 정보

watchElement : 알림 여부

temperLoopTime : 온도 경고 알림 주기 시간

pHLoopTime : 수질 경고 알림 주기 시간

##### 알림 정보를 저장하는 시점

아래 두 조건을 모두 만족 시

1. 사용자가 모든 정보를 정확히 적었을 경우
2. 웹 서버가 socket.io 모듈로 통신이 가능한 경우

The image shows a mobile application interface for the Notification module. At the top, there's a status bar with signal, Wi-Fi, and battery icons, and the time 9:50. The app has a blue header bar. Below it, there are five settings rows, each with a label and a toggle switch or a numeric input field. The first three rows have toggle switches that are currently turned on (red). The last two rows have numeric input fields set to 0. At the bottom, there is a blue bar with three buttons: '저장' (Save), '초기화' (Reset), and '취소' (Cancel).

설정 항목	현재 상태
알림 켜기/끄기	켜기 (ON)
온도 경고 받기	켜기 (ON)
수질 경고 받기	켜기 (ON)
온도 경고 주기 시간	0 초
수질 경고 주기 시간	0 초

저장 | 초기화 | 취소



## 5. 시스템 모듈 상세설계

### 알림 모듈 - App(3/6)

#### updateNotification()

형식	public void updateNotification(int watchElement, int temperLoopTime, int pHLoopTime)
리턴 값	없음
설명	사용자가 입력한 알림 여부와 온도, 수질의 알림 주기 시간 저장.
예시	dbHelper.updateNotification(0, 0, 0);

#### updateNotification() – 데이터 구조

데이터	자료형	설명
watchElement	Int	알림 여부 결정
temperLoopTime	Int	온도 경고 알림 주기 시간
pHLoopTime	Int	수질 경고 알림 주기 시간



## 5. 시스템 모듈 상세설계

### 스트리밍 모듈 - App(4/6)

#### Streaming 모듈

##### 기능

- 남은 먹이 급여 시간을 서버에서 수신.
- 남은 환수 예약 시간을 서버에서 수신.
- 현재 온도와 pH를 서버에서 수신.
- 파이카메라에서 웹으로 출력하는 사진을 웹뷰로 출력.

##### 다루는 정보

웹 서버와의 통신 소켓 : Socket(io.socket.client) socket

webView : StreamingActivity의 웹뷰 객체

예약관리 - 먹이급여 시간 : TextView feedTimer

예약관리 - 환수예약 시간 : TextView waterTimer

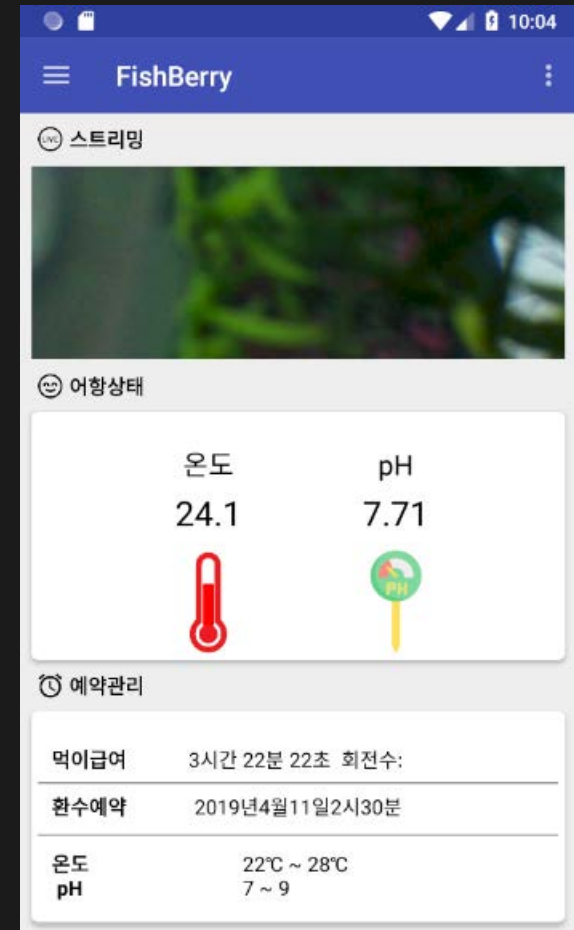
어항상태 - 온도 : TextView tempValue

어항상태 - pH : TextView pHValue

##### 어항 상태 정보를 받아오는 시점

아래 하나의 조건을 만족 시

1. 웹 서버가 socket.io 모듈로 통신이 가능한 경우





## 5. 시스템 모듈 상세설계

### 스트리밍 모듈 - App(4/6)

#### reqTimerFeed()

형식	public void reqTimerFeed(Socket socket)
리턴 값	없음
설명	남은 먹이 급여 시간을 서버에서 수신
예시	streamingPresenterimpl.reqTimerFeed( <i>socket</i> );

#### reqTimerFeed() – 데이터 구조

데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓



## 5. 시스템 모듈 상세설계

### 스트리밍 모듈 - App(4/6)

reqTimerWater()	
형식	public void reqTimerWater(Socket socket)
리턴 값	없음
설명	남은 환수 예약 시간을 서버에서 수신
예시	streamingPresenterimpl.reqTimerWater( <i>socket</i> );

reqTimerWater() – 데이터 구조		
데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓



## 5. 시스템 모듈 상세설계

### 스트리밍 모듈 - App(4/6)

reqTemperPHMsg()	
형식	public void reqTemperPHMsg(Socket socket)
리턴 값	없음
설명	현재 온도와 pH 수치를 서버에서 수신
예시	streamingPresenterimpl.reqTemperPHMsg( <i>socket</i> );

reqTemperPHMsg() – 데이터 구조		
데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓



## 5. 시스템 모듈 상세설계

### 스트리밍 모듈 - App(4/6)

#### startWebView()

형식	public void startWebView(Socket socket)
리턴 값	없음
설명	웹 서버에 출력되는 사진을 웹뷰로 스트리밍.
예시	streamingPresenterimpl.startWebView( <i>socket</i> );

#### startWebView() – 데이터 구조

데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓
webView	WebView	Url 정보를 저장하고 출력하기 위한 WebView 객체





## 5. 시스템 모듈 상세설계

### 온도/ph 설정 모듈 - App(5/6)

#### saveTemperPH 모듈

##### 기능

- 사용자가 설정한 경고 온도 범위와 경고 수질 범위를 설정 및 웹 서버에 저장.

##### 다루는 정보

웹 서버와의 통신 소켓 : Socket(io.socket.client) socket  
minTemper : 최소 온도의 정보가 저장되어 있는 EditText  
maxTemper : 최대 온도의 정보가 저장되어 있는 EditText  
minPH : 최소 수질의 정보가 저장되어 있는 EditText  
maxPH : 최대 수질의 정보가 저장되어 있는 EditText

##### 경고 온도와 pH의 범위를 저장하는 시점

아래 두 조건을 모두 만족 시

1. 사용자가 모든 정보를 정확히 적었을 경우
2. 웹 서버가 socket.io 모듈로 통신이 가능한 경우

The screenshot shows the 'FishBerry' app interface. At the top, there's a blue header with the app name and a menu icon. Below the header, there are two sections for setting warning ranges. The first section is '수온 경고범위' (Water Temperature Warning Range) with a text input field showing a range from '\_\_\_ °C' to '\_\_\_ °C'. The second section is 'pH 경고범위' (pH Warning Range) with a text input field showing a range from '\_\_\_' to '\_\_\_'. At the bottom, there's a blue bar with three buttons: '저장' (Save), '초기화' (Reset), and '취소' (Cancel).



## 5. 시스템 모듈 상세설계

### 온도/ph 설정 모듈 - App(5/6)

#### saveTemperPH()

<b>형식</b>	public void saveTemperPH(Socket socket, EditText minTemper, EditText maxTemper, EditText minPH, EditText maxPH)
<b>리턴 값</b>	없음
<b>설명</b>	웹 서버에 최소 최대 온도, 최소 최대 수질 값을 저장.
<b>예시</b>	temperaturePresenterimpl.saveTemperPH(socket, minTemper, maxTemper, minPH, maxPH);

#### saveTemperPH() – 데이터 구조

데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓
minTemper	EditText	최소 온도가 저장되어 있는 EditText
maxTemper	EditText	최대 온도가 저장되어 있는 EditText
minPH	EditText	최소 수질이 저장되어 있는 EditText
maxPH	EditText	최대 수질이 저장되어 있는 EditText



## 5. 시스템 모듈 상세설계

### 환수 모듈 - App(6/6)

#### Water 모듈

##### 기능

- 현 시각부로 환수를 실행 혹은 재실행
- 현 시각부로 환수를 일시 정지
- 사용자가 원하는 시간에 환수 할 수 있도록 웹 서버에 환수 시각 저장.

##### 다루는 정보

웹 서버와의 통신 소켓 : Socket(io.socket.client) socket

count : 현재 환수 진척도

'바로환수' - '환수 시작' 버튼 : Button btnStartWaterNow

'바로환수' - '일시 정지' 버튼 : Button btnPauseWaterNow

waterReserveFragment : WaterReserveFragment 클래스 안의 시간을 담당하는 변수에 접근하기 위한 객체

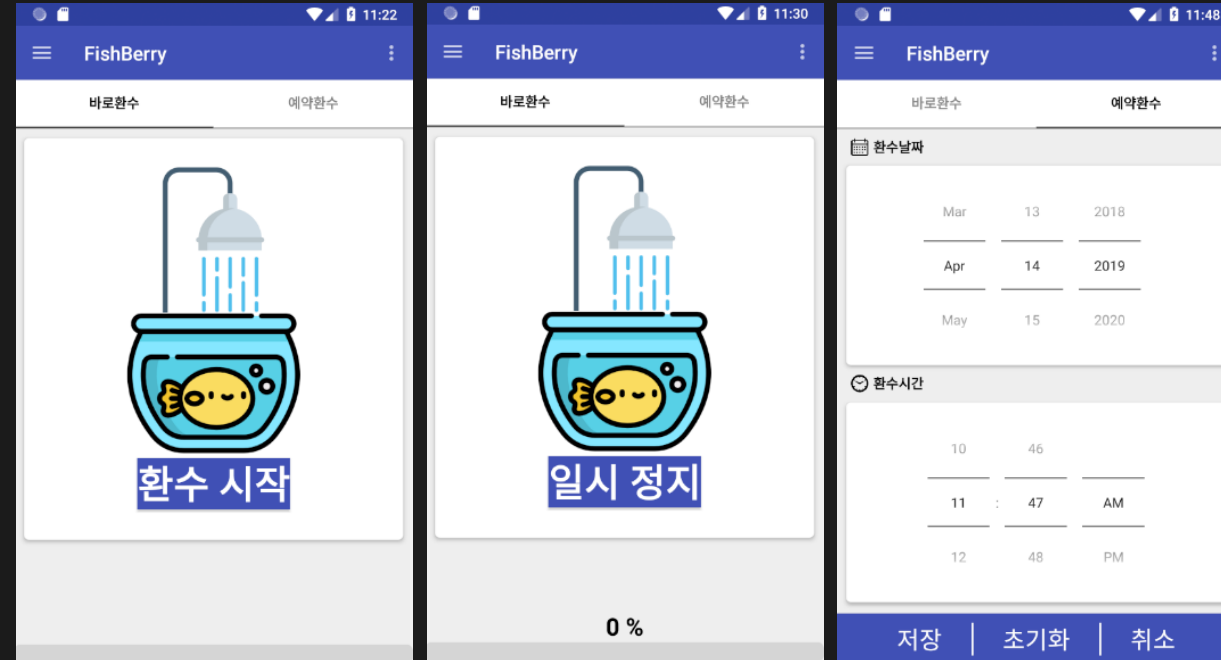
'예약환수' - '환수날짜' : DatePicker datePickerWater

'예약환수' - '환수시간' : TimePicker timePickerWater

##### 환수 실행, 정지, 날짜 저장을 하는 시점

아래 두 조건을 만족 시

1. 웹 서버가 socket.io 모듈로 통신이 가능한 상태일 경우
2. App이 socket.io 모듈로 웹 서버와 통신이 가능한 상태일 경우





## 5. 시스템 모듈 상세설계

### 환수 모듈 - App(6/6)

#### reqChangeWater()

형식	public void reqChangeWater(Socket socket, int count)
리턴 값	없음
설명	웹 서버에게 환수 시작/재시작 요청
예시	waterFragPresenterImpl.reqChangeWater(socket, count);

#### reqChangeWater() – 데이터 구조

데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓
count	int	현재 환수 진척도



## 5. 시스템 모듈 상세설계

### 환수 모듈 - App(6/6)

pauseChangeWater()	
형식	public void pauseChangeWater(Socket socket)
리턴 값	없음
설명	웹 서버에게 환수 일시정지 요청
예시	waterFragPresenterImpl.pauseChangeWater(socket);

pauseChangeWater() – 데이터 구조		
데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓



## 5. 시스템 모듈 상세설계

### 환수 모듈 - App(6/6)

#### saveReserveWater()

형식	public void saveReserveWater(Socket socket, WaterReserveFragment waterReserveFragment)
리턴 값	없음
설명	웹 서버에게 환수 예약 시간 전달
예시	waterFragPresenterImpl.saveReserveWater(socket, waterReserveFragment);

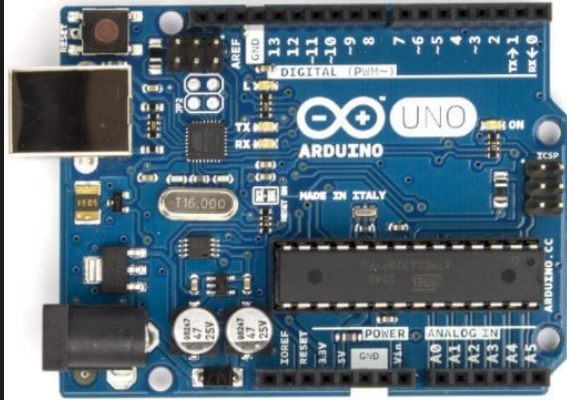
#### saveReserveWater() – 데이터 구조

데이터	자료형	설명
socket	Socket(io.socket.client)	웹 서버에 접근하기 위한 웹 소켓
waterReserveFragment	WaterReserveFragment	WaterReserveFragment 클래스 내의 시간 변수들에 접근을 하기 위한 객체



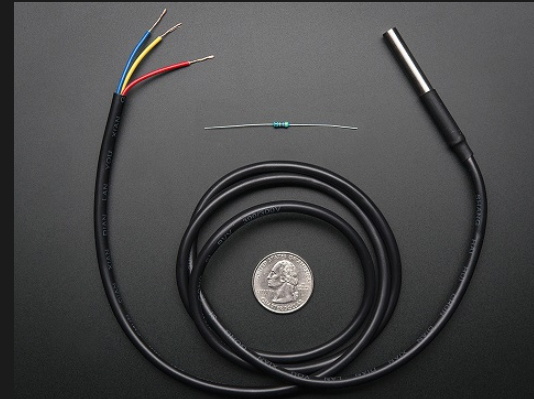
## 6. 개발 환경 및 개발 방법

### 하드웨어 개발 환경



아두이노 우노

- 아두이노 보드
- 플래시 메모리 : 32KB
- 전압 : 5V
- 디지털 입출력 핀 : 14개
- 아날로그 입력 핀 : 6개



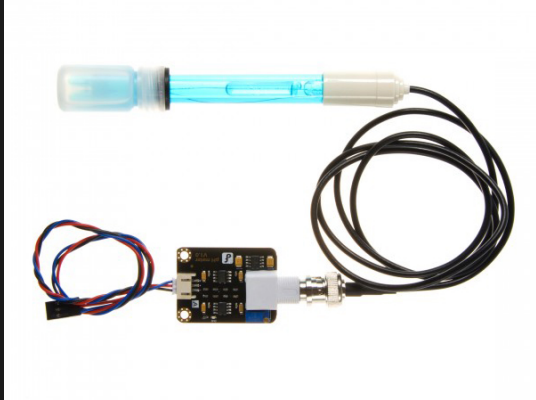
DS18B20

- 아두이노 방수 온도센서
- 스테인리스 스틸 튜브
- 케이블 길이 : 81cm
- 케이블 지름 : 4mm



## 6. 개발 환경 및 개발 방법

### 하드웨어 개발 환경



SEN0161

- 아두이노 pH 수질 센서
- 전압 : 5V
- 규격 : 43mm x 32mm
- 측정 온도 : 0~60도

#### - 부연설명

- ✓ 어종마다 적정 pH가 존재
- ✓ pH 6 ↓ : 생물학적 여과 기능 억제
- ✓ pH 높으면 암모니아 독성 증가
- ✓ 보통 관상어의 적정 pH=6~8.2



사각형 어항

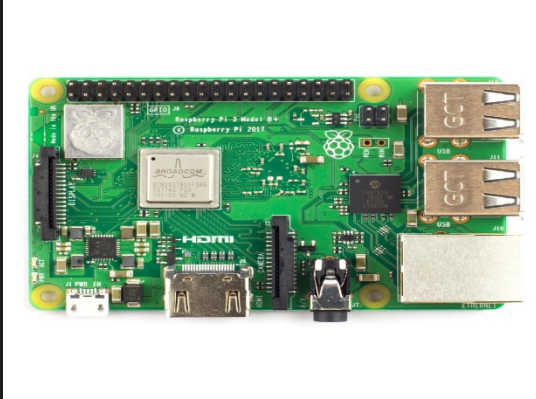
- 재질 : 아크릴케이스
- 어항 규격 : 41cm x 25cm x 25cm





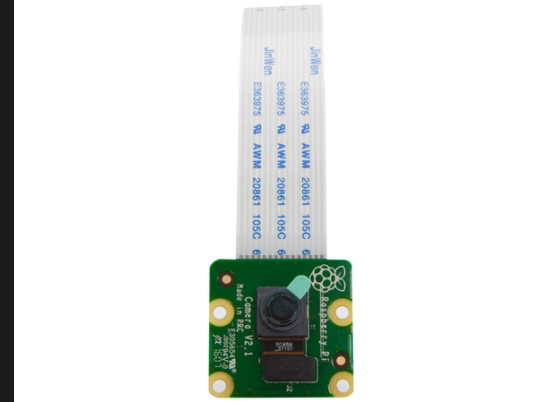
## 6. 개발 환경 및 개발 방법

### 하드웨어 개발 환경



라즈베리파이 3 B+

- 라즈베리파이 보드
- CPU : 1.4GHz ARM Cortex-A53 MP4
- 메모리 : 1GB LPDDR2
- 네트워크 : 10/100/1000 Mbps 기가비트 이더넷  
Wi-Fi 내장 802.11b/g/n/ac Dual-Band  
블루투스 4.2



RPI 8MP CAMERA BOARD

- 파이 카메라
- 해상도 : 3280 x 2464 pixels
- 화질 : 1080p 30Frame, 720p 60Frame



## 6. 개발 환경 및 개발 방법

### 하드웨어 개발 환경



JT-180A 수중 펌프

- 전압 : DC 4.5 ~ 12V
- 무게 : 95g
- 사용 가능 온도 : -20 ~ 90C
- 데시벨 : 35dB 이하



아두이노 워터펌프  
실리콘호스

- 직경 : 8mm



## 6. 개발 환경 및 개발 방법

### 하드웨어 개발 환경



SG5010 서보 모터

- 무게 : 39g
- Stall torque : 5.2kg/cm
- 속도 : 60°/0.2sec
- 크기 : 40 x 20 x 38(mm)



USB 2.0 A-B M-M Type Cable

- 아두이노-라즈베리 파이 통신
- 통신 속도 : (1.5 ~ 480Mbps)



## 6. 개발 환경 및 개발 방법

### 소프트웨어 개발 환경



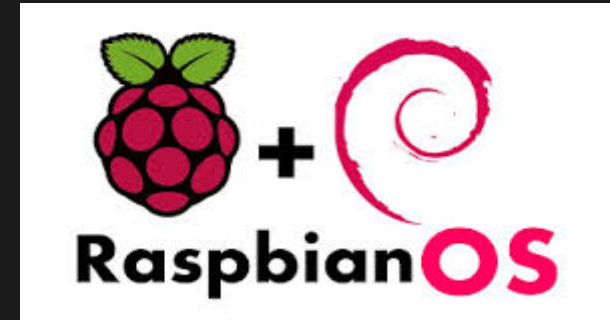
안드로이드 스튜디오

- 개발사 : 구글
- 사용 개발 언어 : Java
- 라이선스 : Apache 2.0
- 사용 버전 : 3.0.1



아두이노 IDE

- 개발사 : 아두이노 소프트웨어
- 사용 개발 언어 : C
- 라이선스 : GPL
- 사용 버전 : 1.8.7



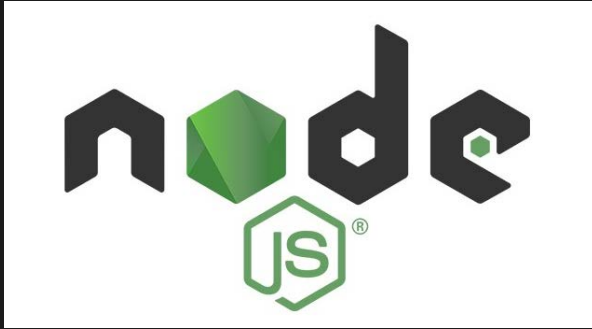
Raspbian OS

- 개발사 : 라즈베리 파이 재단
- 업데이트 방식 : APT
- 라이선스 : GPL
- 사용 버전 : 4.14



## 6. 개발 환경 및 개발 방법

### 소프트웨어 개발 환경



Node.js

- 개발사 : Node JS Foundation
- 라이선스 : MIT
- 사용 버전 : 10.13
- 서버 종류 : 웹
- 서버 사이드 스크립트 언어 : JavaScript



MySQL

- 개발사 : MySQL AB
- 라이선스 : GPL
- 사용 버전 : 10.1.23



Python

- 개발사 : Python SW Foundation
- 라이선스 : Python SW Foundtion Licence
- 사용 버전 : 2.7.13



## 6. 개발 환경 및 개발 방법

### 개발 방법

#### 아두이노

- 아두이노는 센서 값 측정, 부분 환수, 먹이 급여 기능 수행
- '아두이노 우노'와 '아두이노 IDE'를 이용하여 아두이노 시스템 개발
- **아두이노-라즈베리 파이 통신 방법** : 아두이노와 라즈베리 파이를 USB케이블로 연결하고, 라즈베리 파이에 아두이노IDE를 설치하여 Serial Port를 입력하여 **Serial 통신**

#### 라즈베리파이

- 라즈베리 파이는 모바일에서 입력한 사용자의 요구를 아두이노에 전송하는 기능과 아두이노의 센서 측정 값을 사용자에게 전달하는 기능, Pi Camera를 이용해 어항 실시간 스트리밍 기능을 수행
- 라즈베리 파이에 PC에서 putty로 원격접속하여 vim 또는 emacs를 이용하여 **위 기능을 수행하는 코드를 Node.js로 작성**
- **스트리밍 기능**은 라즈비언OS에 **mjpg-streamer와 mjpg-streamer 컴파일을 위한 라이브러리/헤더파일**을 다운로드 하고, mjpg-streamer를 자동실행하는 스크립트를 작성하여 Pi Camera가 찍고 있는 영상을 웹으로 출력하고, 모바일에서는 웹에 출력된 영상을 받아온다.



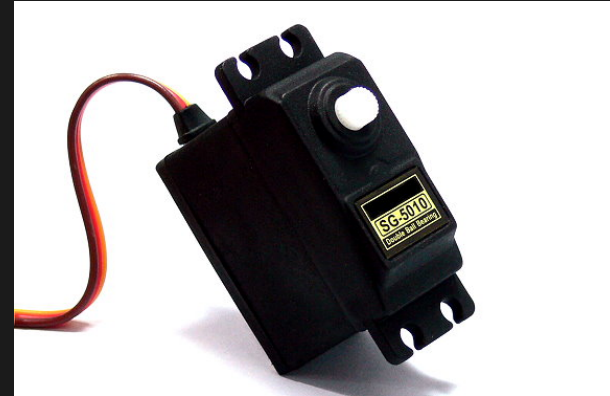
## 6. 개발 환경 및 개발 방법

### 개발 방법

#### 먹이 급여기



<먹이급여기 예시>



<SG5010 서보 모터>

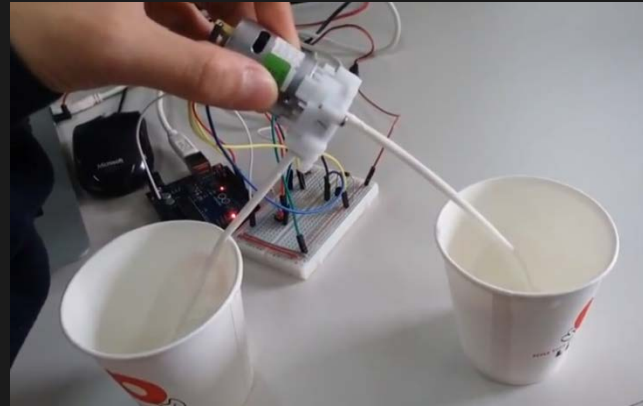
- SG-5010 서보모터와 먹이통을 연결하여 구성
- 아두이노와 서보모터를 연결하여 아두이노가 조작
- 서보 모터가 돌아가면 먹이통이 돌아가게 되면서 먹이통의 문이 아래로 향하게 될 때 먹이가 지급됨



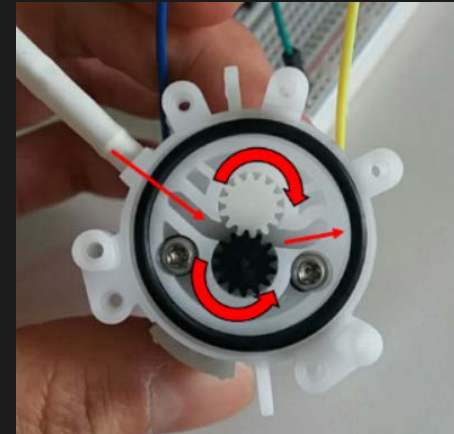
## 6. 개발 환경 및 개발 방법

### 개발 방법

#### 부분 환수기



<부분환수기 예시>



<워터펌프 단면>

- R385 워터펌프에 튜브를 연결하여 구성(자체 제작)
- 아두이노와 워터펌프를 연결하여 아두이노가 조작
- 워터펌프 단면에서 기어가 맞물려 돌아가면서 화살표 방향의 공간으로 물이 배출([유압의 원리](#))





## 6. 개발 환경 및 개발 방법

### 개발 방법

#### 웹 서버

- 라즈비안OS에 Node.js를 이용하여 웹 서비스
- 파이카메라의 이미지 자료를 연속적으로 출력하여 영상화하는 웹 스트리밍 페이지 구성

#### 데이터베이스

- 라즈비안OS에 MySQL 서비스로 DB 구축 및 웹 서버와 연동
- DB에 입력되는 정보들로는 사용자가 모바일에서 입력한 위험수치(온도,pH)설정 값, 환수 시간/먹이급여 시간이 있으며, 아두이노 센서가 측정한 어항의 온도, pH 센서 측정 값 등이 포함

#### 안드로이드

- 안드로이드 스튜디오와 AVD, Java를 이용하여 사용자가 안드로이드 폰을 통해 장치에 설정 값(온도, pH, 환수 시간, 먹이 급여 시간)을 넣을 수 있는 화면, 센서의 측정값을 보여주는 화면, 어항 스트리밍을 볼 수 있는 화면 구성



## 7. 종합설계 수행 일정

구현 현황: 96%

+  $\alpha$  (추가기능)

하드웨어(아두이노, 어항)	95%	서버(라즈베리파이)	100%	App	92%
워터펌프 제작 및 연동	100%	Node.js 서버구축	100%	사용자 로그인	100%
먹이급여기 제작 및 연동	100%	방수온도센서 연동	100%	수온 및 수질측정 관리	100%
RTC 모듈 이용	100%	pH 수질측정 센서 연동	100%	자동 먹이급여 관리	100%
아두이노 제어프로그램 코딩	100%	Database 설계 및 연동	100%	부분환수 관리	100%
라즈베리파이와 연동	100%	파이카메라 설정 및 사용	100%	어항 스트리밍	100%
어항에 하드웨어 부착	80%	웹 로그인	100%	알리미	100%
어항 디자인	80%	홈페이지 페이지 구성&디자인	100%	어플 디자인	100%
		사용자관리	100%	사용자관리	30%



## 8. 데모 환경 설계

### 준비물

- 어항
- 안드로이드 갤럭시S8
- PC(웹)
- 무선 공유기

### 데모 순서

- 어항 스트리밍 출력
- 센서로부터 값을 받아와 안드로이드 폰에 실시간 출력
- 먹이급여 액티비티에서 급여 버튼을 눌러 먹이지급기 작동 (시간 여유될 시 예약까지)
- 환수 액티비티에서 환수 버튼을 눌러 워터펌프 작동 (시간 여유될 시 예약까지)
- 위험온도와 위험pH를 조작하여 위험알림 Notification, LED색상 확인
- 웹에서 위 기능을 동일하게 수행



## 9. 업무 분담

	XXX	yyy	zzz
환수 장치	✓ 모터&Pi Server 연동	✓ 환수 장치 제작 ✓ 모터&Pi Server 연동	✓ 환수 장치 제작 ✓ Pi Server&안드로이드 연동
먹이 급여기	✓ 먹이 급여 장치 제작 ✓ 모터&Pi Server 연동	✓ 먹이 급여 장치 제작 ✓ 모터&Pi Server 연동	✓ Pi Server&안드로이드 연동
아두이노&센서&라즈베리 파이	✓ 아두이노 센서Data를 아두이노에 입력	✓ 아두이노 센서Data를 아두이노에 입력	✓ 아두이노 센서 Data를 라즈베리 파이에 입력
Pi Server	✓ 아두이노, 라즈베리 파이의 Data를 Pi Server로 가져오기 ✓ Pi Server에서 Data를 웹 브라우저로 출력	✓ 아두이노, 라즈베리 파이의 Data를 Pi Server로 가져오기 ✓ Pi Server에서 Data를 웹 브라우저로 출력	✓ Pi Server의 Data를 안드로이드로 출력
DB	✓ DB스키마 설계 ✓ 저장된 Data 웹 연동	✓ DB에 Pi Server Data 저장 ✓ 저장된 Data 웹 연동	✓ DB에 Pi Server Data 저장 ✓ 저장된 Data 안드로이드 연동



## 9. 종합설계 수행 일정

일정	12월	1월	2월	3월	4월	5월	6월	7~9월
자료조사 & 요구사항분석								
시스템 설계								
어항제작 및 IOT 모듈 제어프로그램 개발								
App 설계 및 제작								
서버구축 및 DB 설계								
장치와 서버 & App 연동								
데모 & 디버깅								
최종테스트 및 보고서 작성								



## 11. 필요 기술 및 참고 문헌

### ✓ 라즈베리파이 관련 서적

- 사물인터넷 라즈베리파이 : CCTV
- 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리파이
- 사물인터넷을 품은 라즈베리파이

### ✓ 안드로이드 관련 서적

- 명품 안드로이드 프로그래밍

### ✓ 아두이노 관련 서적

- 익스플로링 아두이노

### ✓ DB 관련 서적

- 데이터베이스 배움터

### ✓ [안드로이드-라즈베리파이] 스마트 어항 블로그

- <http://blog.naver.com/PostView.nhn?blogId=korin57&logNo=220765400422>

### ✓ 라즈베리 파이와 아두이노 연결

- <http://www.makewith.co/page/project/1009/story/2412/>

### ✓ 라즈베리 파이 파이카메라 활용-웹 스트리밍

- <http://www.rasplay.org/?p=7174>



# 12. GitHub

## ✓ GitHub 주소

- <https://github.com/Fishberry/RemoteFishbowl>

## ✓ GitHub ID

- XXX
- yyy
- zzz

The screenshot displays the GitHub repository page for 'Fishberry / RemoteFishbowl'. The repository has 243 commits, 1 branch, 0 releases, 3 contributors, and is licensed under MIT. The 'Code' tab is selected, showing a list of files and their commit history. The files listed are:

File	Commit Message	Time Ago
0201_메인서버_및_APP	소켓만 재업로드	21 hours ago
김병갑	Update README.md	19 days ago
신제우	Update TEST.txt	3 minutes ago
자료들	Github 구조변경	11 days ago
종합설계_보고서	Github 구조변경	11 days ago
차해운	Github 구조변경	11 days ago
LICENSE	Initial commit	4 months ago
README.md	Update README.md	2 months ago



THANK YOU FOR YOUR ATTENTION