

2019년 정보통신기기과 8기 프로젝트 포트폴리오

밥 줘, 시바



마이스터고
MEISTER SCHOOL

과 목	정보통신기기
소 속	정보통신기기과 3학년 1반 10번
제 출 일 시	2019년 11월 11일 월요일
이 름	윤수화
담 당 교 사	김진섭



바른 인성을 갖춘 산업수요 맞춤형 글로벌 기술인재 육성
인천전자마이스터고등학교
<http://intec.icehs.kr>



Contents

I . Introduction

1. 작품 동기	1
2. 작품 계획	1

II . Project

1. 나의 역할	2
2. 관련 전공	2
3. 실무 경험	3
4. 실무 결과	42

III. Reflection

1. 느낀점	43
2. 시사점	43

IV. Activity

1. 활동사진	44
---------------	----

※ 참고문헌	46
--------------	----





Chart

<표 I -1> 작품 계획	1
<표 I -2> 나의 역할	2



[그림 I -1] 라즈베리파이 설정창	1
[사진 I -2] Localisation Options	2
[사진 I -3] 한글 깨짐 현상	2
[사진 I -4] vnc와 ssh 활성화	2
[사진 I -5] 네트워크 연결	2
[사진 I -6] 인터넷 공유	2
[사진 I -7] ssh 로그인	2
[사진 I -8] 라즈베리파이 화면	2

I. Introduction

1 작품 동기

현재 우리나라 인구 중 1000만 명이 반려동물을 키우고 있다. 그렇기에 반려동물, 특히 반려견은 우리 사회에서 중요한 가족의 역할이 되었다. 하지만 직장인의 경우 집을 비우는 시간이 많기 때문에 반려견이 혼자 있을 때가 대부분이다. 반려견이 혼자 오래 있으면 스트레스를 받게 되고, 분리불안에 걸릴 확률이 높아진다. 이러한 문제를 해결하기 위해 집에 혼자 있는 반려견을 놀아주고 관리해 줄 스마트 통합 플랫폼을 만들면 좋겠다고 생각이 들어 제작하게 되었다.

2 작품 계획

<표 I-1> 작품계획

내 용	4월	5월	6월	7월	8월	9월	10월	11월
작품 계획								
부품 조사								
H/W								
S/W								
외형 제작								
H/W 와 S/W 연동								
문제점 수정								
PPT 제작								
보고서 작성								

II. Project

1 나의 역할

밥 쥐, 시바는 어플을 통해 반려견을 지켜보거나 원할 때 간식을 던져주어 즐거움을 줄 수 있다. 나는 어플에서 어떤 동작에 대한 버튼을 클릭할 때 오는 명령어를 판단하여 알맞은 동작을 수행하거나 원하는 데이터 값을 받아 다시 어플로 보내주는 서버와 안드로이드의 마켓 부분을 맡았다.

<표 II-1> 나의 역할

내 용	4월	5월	6월	7월	8월	9월	10월	11월
DB 구상								
PORT 정리								
알고리즘 구상								
안드로이드와 소켓통신								
본체 raspberry와 소켓통신								
smart market과 통신								
하드웨어 마무리 작업								
android 마켓 구현								
외관 꾸미기								

2 관련 전공

1. node.js

node.js는 이벤트 기반 비동식 방식의 고성능 네트워크 서버이다. Javascript를 사용하여 서버와 클라이언트 모두 개발 가능해서 혼자서 하나의 어플리케이션을 개발 할 수 있다. 싱글스레드이기 때문에 하나의 작업 자체가 시간이 많이 걸리면 전체 시스템의 성능이 낮아질 수 있어 주의해야 한다.

2. 라즈베리파이

라즈베리파이는 마이크로프로세서, 메모리, 입출력 연결단자 등을 하나의 회로로 구성한 초소형 컴퓨터라 할 수 있다. 저소득층이나 저개발국가에 거주하는 청소년 및 학생들의 인터넷 및 컴퓨터 교육을 위해 만들어졌다. 다양한 운영체제를 지원하며 가장 널리 사용되는 OS는 라즈비안이다. 강력한 그래픽 기능을 가지고 있어 Full HD 해상도의 동영상에 대한 처리가 가능하다.

3. android studio

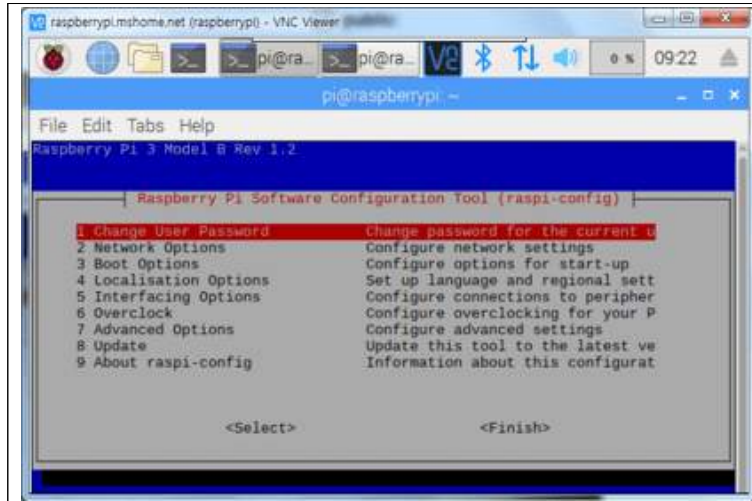
지금까지 안드로이드 개발자들은 이클립스에서 안드로이드 ADT를 플러그인하여 어플리케이션을 개발하였지만 android studio에서는 더 특화된 기능을 추가로 제공한다. 유연성이 좋은 Gradle 기반의 프로젝트 빌드 시스템을 사용하며 하나의 프로젝트 코드로 여러 가지 안드로이드 장치용 어플리케이션을 구현할 수 있다. 구글 클라우드 플랫폼을 자체적으로 지원하여 구글 클라우드 메시징/앱 엔진과 쉽게 통합할 수 있다.

3 실무 경험

1. 라즈베리파이

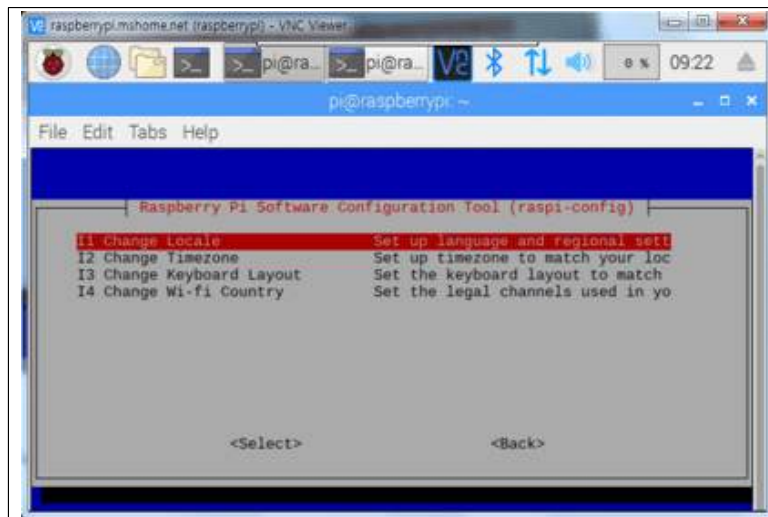
가. 초기설정

- 1) 설치가 완료된 SD를 라즈베리파이에 꽂고 키보드와 마우스를 연결한다.
- 2) 커맨드창을 열고 `sudo raspi-config`를 입력하여 다음과 같은 설정창을 띄워준다.



[그림 II-1] 라즈베리파이 설정창

- 3) Localisation Options를 선택하면 지역정보, 시간정보, 키보드, wifi를 설정할 수 있다.



[그림 II-2] Localisation Options

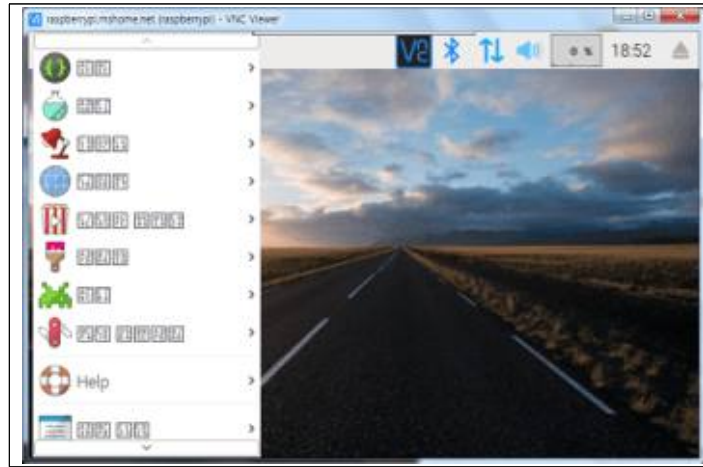
지역 : Change Locale에서 en_GB(UTF8), en_US(UTF8), ko_KR(UTF8), ko_KR(EUC)을 선택하고 기본으로는 ko_KR(UTF8)을 선택해준다.

시간 : Change Timezone에서 Asia 선택 후 Seoul로 지정해준다.

키보드 : Change Keyboard Layout에서 Generic 105-key(Intl) PC => Other => Korean => Korean-Korean(101-104 key compatible) => The default for the keyboard layout => No compose Key => No를 차례대로 선택해준다.

wifi : Change Wi-fi Country에서 country는 USA로 설정해 주어야 wifi 연결이 가능하다.

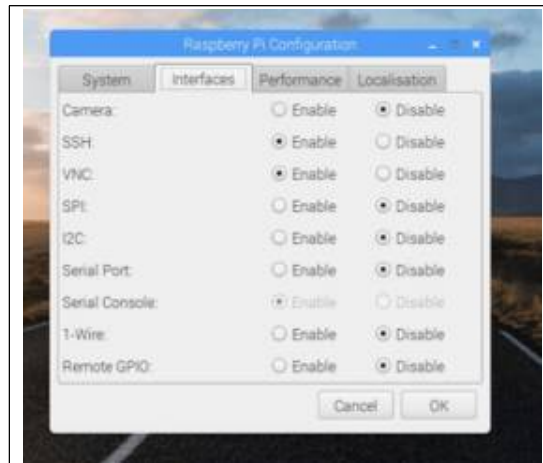
- 4) 커맨드 창을 열고 `sudo apt-get update`를 입력하여 업데이트를 한다.
- 5) `sudo apt-get upgrade`를 입력하여 업그레이드를 한 후 `sudo reboot`으로 재시작해준다.
- 6) 다음과 같이 한글이 깨져 보이는 경우 커맨드 창에 `sudo apt-get install ibus => sudo apt-get install ibus-hangul => sudo apt-get install fonts-unfonts-core`를 차례로 입력하고 재시작하면 한글이 제대로 나타난다.



[그림 II-3] 한글 깨짐 현상

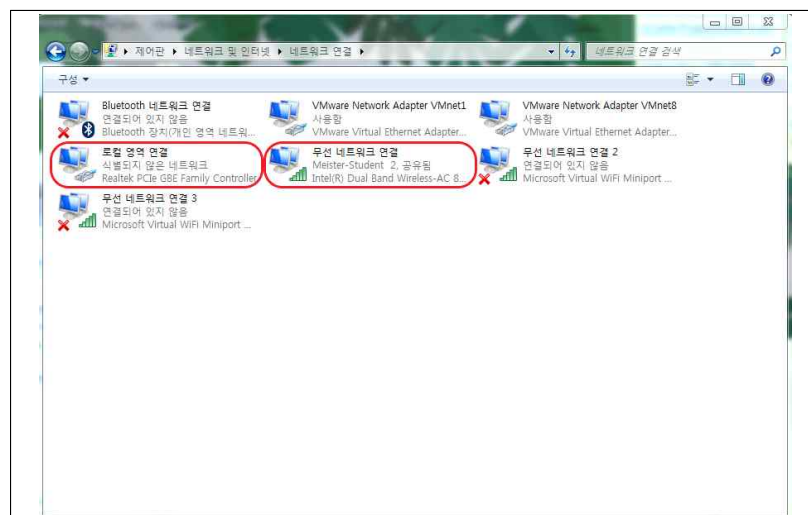
나. 원격접속

- 1) putty와 vnc viewer를 설치한다.
- 2) ssh 접속과 vnc를 사용하기 위해 라즈베리파이 설정에서 ssh와 vnc를 활성화시켜준다.



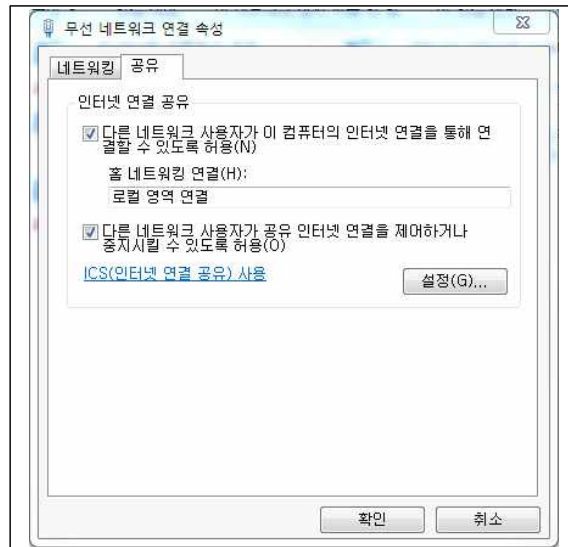
[그림 II-4] vnc와 ssh 활성화

- 3) 제어판 => 네트워크 및 인터넷 => 네트워크 및 공유센터 => 어댑터 설정 변경 에 들어가면 라즈베리파이와 연결된 “로컬 영역 연결”과 인터넷에 연결되어 있는 “무선 네트워크 연결”이 나타난다.



[그림 II-5] 네트워크 연결

- 4) “무선 네트워크 연결”의 속성에 들어가 팝업창의 2번째 “공유”탭에서 인터넷 공유를 활성화한다.

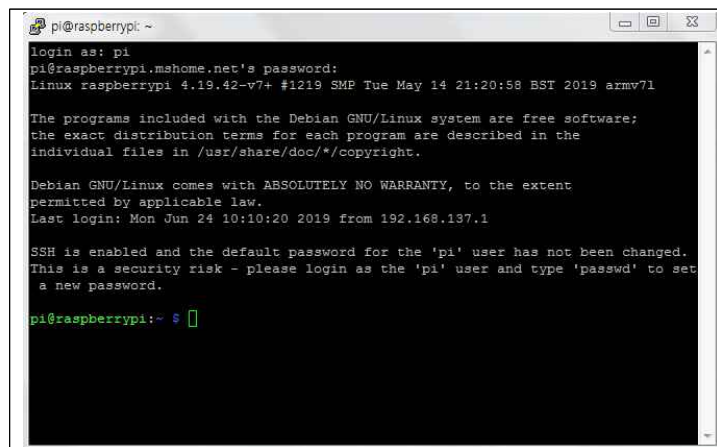


[그림 II-6] 인터넷 공유

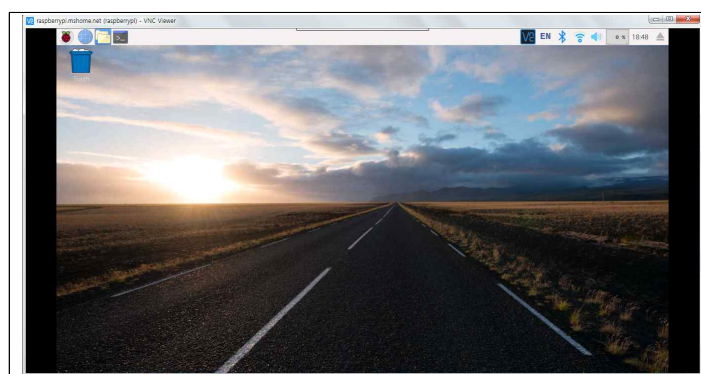
5) putty를 실행하여 Host Name에 도메인 주소인 raspberrypi.mshome.net을 넣어준다.

* mshome.net은 Windows IOS(Internet Connection Share)에서 제공하는 DNS이다.

6) putty를 통해 ssh로 로그인한 후 vnc viewer를 실행하여 raspberrypi.mshome.net을 입력하면 라즈베리 파이의 메인화면이 나오는 것을 확인할 수 있다.



[그림 II-7] ssh 로그인



[그림 II-8] 라즈베리파이 화면

다. node.js 설치

node.js를 이용하여 서버를 구축하기 위해 라즈베리파이에 node.js를 설치해준다.

1) node.js를 최신버전으로 설치하기 위해 다음과 같이 입력한다.

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash-
```

2) sudo apt-get install -y nodejs를 입력하여 node.js를 설치한다.

3) 설치가 완료되었으면 node -v를 입력하여 정상적으로 설치되었는지 버전을 확인한다.

4) 그 후 새 폴더를 생성해 node.js 프로젝트를 생성한다.

=> mkdir 폴더명

=> cd 폴더명

=> npm init

* npm(Node Package Manager)은 다양한 기능을 더해주는 모듈들을 패키지로 만들어 관리해준다.

라. MariaDB 설치

mariadb는 오픈 소스 관계형 데이터베이스 관리 시스템으로, mysql을 모태로 개발되었다. 최신 버전의 라즈비안에서 기본적으로 mysql을 설치하려고 하면 mariadb가 설치된다.

1) sudo apt-get install mysql-server mysql-client를 입력하여 server 및 client를 설치한다.

2) sudo service mysql status를 통해 정상적으로 동작하는지 확인한다.

* 정상적일 경우 active(running) 표시됨

```

pi@raspberrypi:~ $ sudo service mysql status
● mariadb.service - MariaDB 10.1.38 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
   Active: active (running) since Mon 2019-06-24 20:32:56 KST; 11h ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 647 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START
   Process: 620 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCC
   Process: 491 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR=
   Process: 480 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_
   Process: 459 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run
   Main PID: 577 (mysqld)
    Status: "Taking your SQL requests now..."
   CGroup: /system.slice/mariadb.service
           └─577 /usr/sbin/mysqld

Jun 24 20:32:51 raspberrypi systemd[1]: Starting MariaDB 10.1.38 database server
Jun 24 20:32:53 raspberrypi mysqld[577]: 2019-06-24 20:32:53 1996070704 [Note] /
Jun 24 20:32:56 raspberrypi systemd[1]: Started MariaDB 10.1.38 database server.
lines 1-18/18 (END)
    
```

[그림 II-10] mariaDB 설치 확인

2. DataBase

```

MariaDB [mydog]> select * from feed;
+-----+-----+-----+
| name                                     | price | grade |
+-----+-----+-----+
| 주식권 쇠고기, 100g, 24개 일            | 27,300 | 4      |
| 양고기 해건 사프, 5.2kg                 | 59,600 | 1      |
| 양마지 사프 연어 레시피, 3kg            | 15,500 | 3      |
| 유기농 닭 양마지 사프, 4kg              | 48,000 | 3      |
| 피부에 좋은 사프, 2.2kg                | 28,800 | 5      |
| 치킨 & 야채 슈퍼 프리미엄, 7.2kg        | 27,400 | 2      |
| 전연령 오리 버네 해건 사프, 6.4kg       | 34,000 | 5      |
| 양마지 사프 빙나는 피요맛, 2kg          | 18,400 | 4      |
| 생선 건강관 관절 해건 사프, 3.9kg        | 28,800 | 4      |
| 유기농 6Free 연어, 6kg                 | 37,000 | 2      |
| 그레인프리 스톱브리트 어덜트, 2kg      | 27,200 | 4      |
| 플리스틱 네츄럴바이스, 2.27kg           | 22,900 | 5      |
| 미니 인포어 어덜트, 8.7kg               | 67,900 | 2      |
| 말랑말랑 소프트발 양고기, 3kg           | 33,600 | 5      |
| 프리바이오틱스 해건 사프               | 14,900 | 1      |
| 어덜트 스톱브리트, 11.3kg               | 99,000 | 3      |
| 유리아 피피 자우, 14.5kg                | 36,500 | 3      |
| 가스트로 인데스티널 로우렛, 1.5kg       | 23,600 | 2      |
| 말스코 워프 양마지 사프, 2kg            | 17,100 | 5      |
| 플삭 플리스틱 연어, 5.2kg               | 59,000 | 2      |
+-----+-----+-----+
20 rows in set (0.01 sec)

MariaDB [mydog]>
    
```

[그림 II-11] 마켓 사료 database

```

pi@raspberrypi: ~
MariaDB [mydog]> select * from snack;
+-----+-----+-----+
| name                                     | price | grade |
+-----+-----+-----+
| 저녁에 요리버거 100g                   | 450   | 3     |
| 파트유산균 혼합 12개입                 | 3,600 | 4     |
| 청알 요리가 3면어 180g                | 4,900 | 4     |
| 연달힐 린트함 12개입                  | 1,300 | 2     |
| 카리얼 연달힐 6개입                   | 1,800 | 5     |
| 친구 굼프치                             | 3,500 | 1     |
| 네이저 소고기 류프 300g               | 2,200 | 3     |
| 와우와우 혼합간식 150g                | 1,800 | 1     |
| 치킨추펄고퀄 5개입                   | 1,100 | 4     |
| 클라겐 핏터스틱 30개입                | 5,200 | 3     |
| 도기쓰리 저키 숯불감바맛 120g         | 2,000 | 3     |
| 해쓰프렌드 치킨 소시지 8개입           | 1,000 | 4     |
| 인견우유 아이젯밀크 180ml             | 2,500 | 2     |
| 도그맨 고구마 벨곶케익 100g           | 1,800 | 5     |
| 미스 죽 연어 슬라이스 300g            | 2,800 | 4     |
| 로비도비 치즈 소세지대장 11개입       | 1,300 | 5     |
| ENF 밀크스틱 27개입                   | 4,300 | 5     |
| 리틀렛 비타 합습 비스켓 100g          | 1,000 | 1     |
| 구르스 밀크곶 5개입                   | 900   | 4     |
| 아들아들 문득한 소고기 120g           | 4,700 | 3     |
+-----+-----+-----+
20 rows in set (0.00 sec)

MariaDB [mydog]>

```

[그림 II-12] 마켓 간식 database

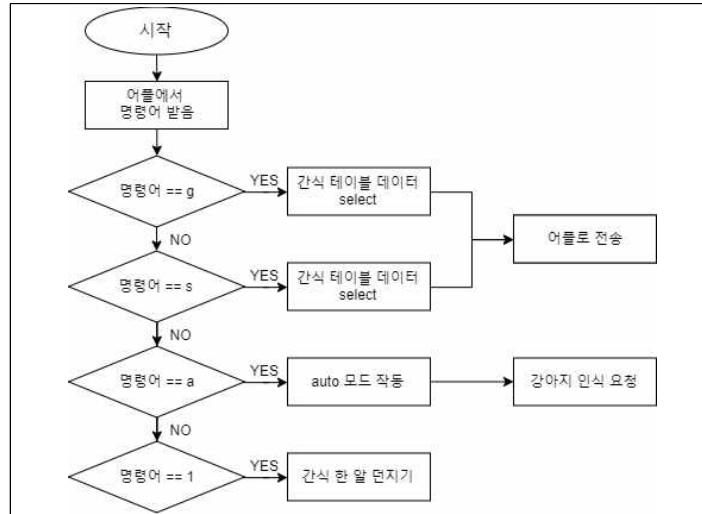
```

pi@raspberrypi: ~
MariaDB [mydog]> select * from user
-> ;
+-----+
| date                |
+-----+
| 2019-07-05 12:26:07 |
| 2019-07-05 12:26:11 |
| 2019-07-05 12:26:23 |
| 2019-10-17 19:11:01 |
| 2019-10-17 19:11:03 |
| 2019-10-17 19:11:05 |
| 2019-10-17 19:11:06 |
| 2019-10-23 15:31:23 |
| 2019-10-23 15:55:23 |
| 2019-10-23 16:10:28 |
| 2019-10-23 16:11:02 |
| 2019-10-23 16:26:42 |
| 2019-10-23 16:27:00 |
| 2019-10-23 17:00:20 |
| 2019-10-23 17:00:45 |
| 2019-10-23 17:01:02 |
| 2019-10-24 20:20:11 |
| 2019-10-24 20:20:34 |
| 2019-10-24 20:27:05 |
| 2019-10-24 20:27:46 |
| 2019-10-24 20:32:52 |
| 2019-10-24 20:33:47 |
| 2019-10-24 20:39:08 |
| 2019-10-24 20:39:26 |
| 2019-10-24 20:39:41 |
+-----+
31 rows in set (0.00 sec)

```

[그림 II-13] 간식 준 시간 database

3. 서버



[그림 II-14] 순서도

가. server.js

```

var net = require('net');
var app = require('./app_server.js');
var que = require('./query.js');
var dog = require('./dog_server.js');

var moment = require('moment');

function time(){
    //시간을 나타내기 위한 사용자 정의 함수
    return '['+moment().format("YYYY-MM-DD, HH:mm:ss")+']: ';
}
exports.time = time;

// Raspberry 서버 생성
var rb_server = net.createServer(function(socket){
    console.log(time()+"Pi connected");
    exports.appsend = function(data){
        console.log(time()+"AppServer --> Pi : "+ data);
        app_rb_data = data;
        socket.write(app_rb_data);
    }
    exports.dogsend = function(data){
        console.log(time()+"DogServer --> Pi : "+data);
        dog_rb_data = data;
        socket.write(dog_rb_data);
    }
    socket.on('data', function(data){
        console.log(time()+"Pi --> Server : " + data);
        if(data == "d"){
            dog.send(data);
        }
        else if(data == "auto on"){
            app.send(data);
        }
    })
});
    
```

//소켓 서버를 위한 모듈

//Pi 연결

//send 함수가 호출되면

//app에서 Pi로 보내는 데이터 출력

//데이터 옮겨담기

//rb에 데이터 전송

//개 인식 후 Pi에 보내기 위한 사용자 정의 함수

//Pi가 데이터를 보냈을 때

//데이터가 d이면

//dog server의 함수 사용

//데이터가 auto on이면

//app server의 함수 사용

```

    })
    socket.on('end', function(data){
        console.log(time()+"Pi disconnected.");
    });

    socket.on('error', function(err){
        console.log(time()+err);
    });
})

rb_server.listen(8403, function(){
    console.log('listening on Pi(8403)...');
})

```

//소켓 연결 종료시 콜백함수 실행
// 콘솔창에 연결이 종료되었다고 출력

//소켓 연결에 에러 발생 시 콜백함수 실행
// 에러를 콘솔창에 출력

// 포트 8403 열기

나. app_server.js

```

var net = require('net');
var que = require('./query.js');
rb = require('./server.js');

var moment = require('moment');

// 앱 서버 생성
var app_server = net.createServer(function(socket){
    console.log(rb.time()+"App connected");

    exports.send = function(data){
        console.log(rb.time()+"Server --> App : " + data);
        rb_dog_data = data
        socket.write(rb_dog_data);
    }

    socket.setEncoding("utf8");
    socket.on('data', function(data){
        console.log(rb.time()+"App --> AppServer:" + data);
        if(data == "s"){
            que.select('feed', function(db){
                console.log(db);
                socket.write(db);
            })
        }else if(data == "g"){
            que.select('snack', function(db){
                console.log(db);
                socket.write(db);
            })
        }else{
            try{
                rb.append(data);
                if(data=="1"){

```

// 소켓 서버를 위한 모듈

//어플 연결

//어플로 데이터 보내기 위한 사용자 정의 함수

//데이터 UTF-8로 인코딩
//데이터를 받았을때

//사료 데이터(테이블)를 요청하는 명령어가 오면
//select문 실행

//app에 데이터값 전송

//간식 데이터(테이블)를 요청하는 명령어가 오면
//select문 실행

//app에 데이터값 전송

//그 외 명령어가 오면

//server.js의 send 함수로 data 전송

```

        socket.write("Shooting");
        que.insert_time() //시간 저장
    }else
        socket.write(data); //전송이 되었음을 app에 알림
    }catch(err){ //에러가 나면, 연결이 안되는 것이다
        console.error(rb.time()+err)
        socket.write("Pi POWER OFF") //클라이언트가 Off되어있음으로 판단하고 넘긴다
    }
}
});

socket.on('end', function(data){ // 소켓 연결 종료 시 콜백함수 실행
    console.log(rb.time()+"App disconnected."); // 콘솔창에 연결이 종료되었다고 출력
});

socket.on('error', function(err){ // 소켓 연결에 에러 발생 시 콜백함수 실행
    console.log(rb.time()+err); // 에러를 콘솔창에 출력
});
})

app_server.listen(8402, function(){ //포트 8402 열기
    console.log('listening on App(8402)...');
})

```

다. query.js

```

var net = require('net'); //소켓 서버를 위한 모듈
var rb = require('./server.js');
var app = require('./app_server.js');
var mysql = require("mysql");

var dog = mysql.createConnection({ //mariaDB에 연결
    host: "localhost",
    user: "root",
    password: "1234",
    port: "3306",
    database: "mydog"
});

module.exports.select = function(name, callback){ //select문 실행을 위한 사용자 정의 함수
    console.log('SELECT * FROM ' + name);
    var selectsql = "SELECT * FROM " + name;
    dog.query(selectsql, function(err, result){ //쿼리문 실행
        if(!err){ // 쿼리문이 에러가 없다면
            var db = JSON.stringify(result); // 결과값을 JSON 형태의 문자열로 변환
            callback(db); // 연결된 소켓으로 데이터 전송
        } else{ // 에러가 있다면
            callback("no data!"); //데이터가 없다고 전송
        }
    })
}

```

```

    });
}

module.exports.insert_time = function(){
    //시간 저장을 위한 사용자 정의 함수
    var in_feed = "INSERT INTO user (date) values (default)"; //먹이 모터 작동시간 저장
    dog.query(in_feed, function(err){
        //쿼리문 실행
        if(err)
            console.log(err);
    })
}

```

4. 안드로이드 (마켓)

가. pet_MarketActivity.java

```

package com.example.shiba;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;

import android.support.v7.app.AppCompatActivity;

import java.util.ArrayList;

public class pet_MarketActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.pet_activity_market);

    }

    public void onClick(View v) {
        ArrayList cart_name_list = getIntent().getStringArrayListExtra("cart_name_list");
        ArrayList cart_count_list = getIntent().getStringArrayListExtra("cart_count_list");
        ArrayList cart_price_list = getIntent().getStringArrayListExtra("cart_price_list");

        switch (v.getId()) {
            case R.id.picture_button: //+버튼을 눌렀을 때
                break;

            case R.id.gansick_btn1: //간식버튼을 눌렀을 때
                Intent i = new Intent(pet_MarketActivity.this, pet_GansickActivity.class);
                i.putExtra("cart_name_list", cart_name_list); //putextra로 간식 액티비티에 넘김
                i.putExtra("cart_price_list", cart_price_list);
            }
        }
    }
}

```

```

        i.putExtra("cart_count_list", cart_count_list);
        startActivity(i); //간식 액티비티로 넘어감
        finish(); //화면 종료
        break;

    case R.id.saryo_btn1: //사료버튼을 눌렀을 때
        Intent i1 = new Intent(pet_MarketActivity.this, pet_SaryoActivity.class);
        i1.putExtra("cart_name_list", cart_name_list); //putextra로 사료 액티비티에 넘김
        i1.putExtra("cart_price_list", cart_price_list);
        i1.putExtra("cart_count_list", cart_count_list);
        startActivity(i1); //사료 액티비티로 넘어감
        finish(); //화면 종료
        break;
    }
}

public void onBackPressed(){ //Back 버튼을 누르면
    Intent intent = new Intent(pet_MarketActivity.this, pet_MainActivity.class);
    startActivity(intent); //메인 화면으로 넘어감
    finish(); //화면 종료
}

```

나. pet_activity_market.xml

```

package com.example.shiba;

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".pet_GansickActivity"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/pet_market_2">

    <android.support.constraint.Guideline // 사료 버튼 위치 조정
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/left"
        app:layout_constraintGuide_percent="0.30"
        android:orientation="vertical" />

    <android.support.constraint.Guideline // 간식 버튼 위치 조정
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/right"
        app:layout_constraintGuide_percent="0.70"
        android:orientation="vertical" />

    <android.support.constraint.Guideline //버튼 높이 위치 조정
        android:layout_width="wrap_content"

```



```

        android:layout_height="wrap_content"
        android:id="@+id/h"
        app:layout_constraintGuide_percent="0.66"
        android:orientation="horizontal" />

<ImageButton                //사료 버튼
    android:onClick="onClick"
    android:id="@+id/saryo_btn1"
    android:background="#00000000"
    android:layout_width="180dp"
    android:layout_height="210dp"
    app:layout_constraintRight_toRightOf="@id/left"
    app:layout_constraintLeft_toLeftOf="@id/left"
    app:layout_constraintTop_toTopOf="@id/h"
    app:layout_constraintBottom_toBottomOf="@id/h"
    android:scaleType="fitXY"
    app:srcCompat="@drawable/pet_sa_ryo_button" />

<ImageButton                //간식 버튼
    android:onClick="onClick"
    android:id="@+id/gansick_btn1"
    android:background="#00000000"
    android:scaleType="fitXY"
    android:layout_width="180dp"
    android:layout_height="210dp"
    app:layout_constraintRight_toRightOf="@id/right"
    app:layout_constraintLeft_toLeftOf="@id/right"
    app:layout_constraintTop_toTopOf="@id/h"
    app:layout_constraintBottom_toBottomOf="@id/h"
    app:srcCompat="@drawable/pet_gan_sick_button" />

</android.support.constraint.ConstraintLayout>

```

다. pet_Gansick_Activity.java

```

package com.example.shiba;

package com.example.shiba;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.TextView;

```

```

import android.widget.Toast;

import android.support.v7.app.AppCompatActivity;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.net.InetSocketAddress;
import java.nio.ByteBuffer;
import java.nio.channels.SocketChannel;
import java.nio.charset.Charset;
import java.util.ArrayList;

import static android.os.StrictMode.setThreadPolicy;

public class pet_GansickActivity extends AppCompatActivity {

    TextView textV, select_v, set_price_v;
    Button bt_plus;
    Button bt_minus;
    ImageButton bt_cart;
    ArrayList<String> nameList, priceList, gradeList;
    String price, item, total_price, temp;
    int count;
    int order;
    final Context context = this;

    SocketChannel socketChannel = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.pet_activity_gansick);

        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        setThreadPolicy(policy);

        textV = (TextView)findViewById(R.id.counting_textline);
        select_v = (TextView)findViewById(R.id.select_gansick);
        set_price_v = (TextView)findViewById(R.id.set_price_v);
        bt_plus = (Button)findViewById(R.id.plus);
        bt_minus = (Button)findViewById(R.id.minus);
        bt_cart = (ImageButton)findViewById(R.id.cart);
        LinearLayout linear = (LinearLayout) findViewById(R.id.Left_layout);
    
```

```

count=1;
textV.setText(""+count);

nameList = new ArrayList<>(); //ArrayList 정의
priceList = new ArrayList<>();
gradeList = new ArrayList<>();

socketConnect("192.168.0.53", 8402); //소켓 연결
sendData("g"); //간식 테이블 요청

Button btn[] = new Button[nameList.size()]; //이름 배열의 길이만큼
for(int i=0; i<btn.length; i++) { //버튼 추가
    btn[i] = new Button(this);
    btn[i].setText(nameList.get(i));
    btn[i].setId(i);
    linear.addView(btn[i]);

    final int finalI = i;
    btn[i].setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            select_v.setText(nameList.get(finalI));
            set_price_v.setText(priceList.get(finalI).replaceAll(",",""));
            order = finalI;
            count = 1;
            textV.setText("" + count);
        }
    });
}

select_v.setText(nameList.get(0)); //첫 버튼은 index 0번째
set_price_v.setText(priceList.get(0).replaceAll(",",""));
}

public void onClick(final View v) {
    price = priceList.get(order).replaceAll(",","");
    switch (v.getId()){
        case R.id.plus:
            count++;

            if(count > 10){ //열 개 이상 사려고 하면
                count = 10;
                Toast.makeText(this, "한 번에 10개 이상 구매할 수 없어요..",
                    Toast.LENGTH_SHORT).show();
            }
            textV.setText(""+count); //그대로 넣으면 문자열로 인식 X, 그래서 큰따옴표 붙여줌
            set_price_v.setText(""+Integer.parseInt(price)*count);
            break;

        case R.id.minus:
            count--;
    }
}

```

```

        if(count < 1){ //한 개 이하로 사려고 하면
            Toast.makeText(this, "최소 하나 이상 구입해주세요!", Toast.LENGTH_SHORT).show();
            count = 1;
        }

        textV.setText(""+count);
        set_price_v.setText("" + Integer.parseInt(price)*count);
        break;

    case R.id.cart:
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {

            @Override
            public void run() {

                item = select_v.getText().toString();
                total_price = set_price_v.getText().toString();
                temp = String.valueOf(count);

                ArrayList<String> cart_name_list =
                getIntent().getStringArrayListExtra("cart_name_list");
                ArrayList<String> cart_count_list =
                getIntent().getStringArrayListExtra("cart_count_list");
                ArrayList<String> cart_price_list =
                getIntent().getStringArrayListExtra("cart_price_list");

                cart_name_list.add(item);
                cart_count_list.add(temp);
                cart_price_list.add(total_price);

                //장바구니 화면으로 이동
                Intent intent = new Intent(pet_GansickActivity.this, pet_CartActivity.class);
                intent.putExtra("cart_name_list", cart_name_list); //putExtra로 데이터 전송
                intent.putExtra("cart_price_list", cart_price_list);
                intent.putExtra("cart_count_list", cart_count_list);

                startActivity(intent);
                finish(); //화면 종료
            }
        }, 500); //0.5초 후 카트액티비티로 넘어감
    }
}

public void sendData(String data){ //데이터 발신
    Log.d("Send", data);

    try {
        ByteBuffer buffer = null;
        Charset charset = Charset.forName("UTF-8");
    }
}

```

```

        buffer = charset.encode(data);

        socketChannel.write(buffer);

        buffer = ByteBuffer.allocate(15000);
        int len = socketChannel.read(buffer);
        buffer.flip();

        String recive = charset.decode(buffer).toString();
        ReceiveData(recive);

    } catch (Exception e){
        e.printStackTrace();
    }
}

public void socketConnect(String host, int port){    //소켓 연결
    try{
        socketChannel = SocketChannel.open();
        socketChannel.configureBlocking(true);
        socketChannel.connect(new InetSocketAddress(host, port));

    } catch (Exception e){
        e.printStackTrace();
    }
}

@Override
public void finish() {    //종료
    if(socketChannel.isOpen()){
        try {
            socketChannel.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    super.finish();
}

public void ReceiveData(String recive){    //데이터를 받으면

    JSONArray jsonArray = null;
    try {
        jsonArray = new JSONArray(recive);
        for(int i = 0; i<javascriptArray.length(); i++){    //데이터 list에 추가
            JSONObject jsonObject = jsonArray.getJSONObject(i);
            String name = jsonObject.getString("name");
            String price = jsonObject.getString("price");
            String grade = jsonObject.getString("grade");
            nameList.add(name);
            priceList.add(price);
        }
    }
}

```

```

        gradeList.add(grade);
    }
} catch (JSONException e) {
    e.printStackTrace();
}
}
public void onBackPressed(){
    //마켓 화면 이동
    Intent intent = new Intent(pet_GansickActivity.this, pet_MarketActivity.class);
    startActivity(intent);
    finish();
}
}

```

라. pet_activity_gansick.xml

```

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".pet_GansickActivity"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffffff">

    <android.support.constraint.Guideline
        android:id="@+id/guideline50per"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.57" />

    <ScrollView
        android:id="@+id/scrollView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@drawable/pet_pet_pattern1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/guideline50per"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:id="@+id/Left_layout"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="20dp"
            android:scrollbarStyle="insideInset">

```

```

        </LinearLayout>
    </ScrollView>

    <LinearLayout
        android:background="#ffffff"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@+id/guideline50per"
        app:layout_constraintTop_toTopOf="parent"
        android:id="@+id/Right_layout"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0"
            android:background="@drawable/pet_early_buy"
            android:gravity="center"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:padding="7dp">

                <TextView
                    android:id="@+id/textView5"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_margin="10dp"
                    android:text="선택 상품"
                    android:textColor="#000000"/>

                <TextView
                    android:id="@+id/select_gansick"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:textColor="#000000"
                    android:layout_gravity="center"

```

```

        android:gravity="center"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:orientation="horizontal"
    android:padding="7dp">

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:text="가격"
        android:textColor="#000000" />

    <TextView
        android:id="@+id/set_price_v"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textColor="#000000"
        android:layout_gravity="center"/>

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="원"
        android:textColor="#000000"
        android:layout_gravity="center"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:orientation="horizontal"
    android:padding="7dp">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="1"

```



```

        android:text="수량 :"
        android:textColor="#000000"
        android:layout_gravity="center"/>

<Button
    android:onClick="onClick"
    android:id="@+id/minus"
    android:layout_width="12dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.5"
    android:background="@drawable/pet_button"
    android:text="-"
    android:layout_gravity="center"/>

<TextView
    android:id="@+id/counting_textline"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:textAlignment="center"
    android:textColor="#000000"
    android:textSize="20dp"
    android:onClick="onClick"
    android:layout_gravity="center" />

<Button
    android:onClick="onClick"
    android:id="@+id/plus"
    android:layout_width="12dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.5"
    android:background="@drawable/pet_button"
    android:text="+"
    android:layout_gravity="center"/>
</LinearLayout>

<ImageButton
    android:onClick="onClick"
    android:id="@+id/cart"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_marginTop="10dp"
    android:background="#00000000"
    android:scaleType="fitXY"
    app:srcCompat="@drawable/pet_cart" />
</LinearLayout>
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```



[그림Ⅲ-2] android market 간식

마. pet_SaryoActivity.java

```
package com.example.shiba;
```

```
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
```

```
import java.io.IOException;
import java.net.InetSocketAddress;
import java.nio.ByteBuffer;
import java.nio.channels.SocketChannel;
import java.nio.charset.Charset;
import java.util.ArrayList;
```

```
import static android.os.StrictMode.setThreadPolicy;
```

```
public class pet_SaryoActivity extends AppCompatActivity {
    TextView textV2, select_v, set_price_v;
    Button bt_plus, bt_minus;
```

```

ImageButton bt_cart;
String price, item, total_price, temp;
int count;
int order;

final Context context = this;
ArrayList<String> nameList, priceList, gradeList;

SocketChannel socketChannel = null;
@Override
protected void onCreate(Bundle savedInstanceState) {
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
    super.onCreate(savedInstanceState);
    setContentView(R.layout.pet_activity_saryo);

    StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
    setThreadPolicy(policy);

    textV2 = (TextView)findViewById(R.id.counting_textline);
    select_v = (TextView)findViewById(R.id.select_saryo);
    set_price_v = (TextView)findViewById(R.id.set_price_v);
    bt_plus = (Button)findViewById(R.id.plus);
    bt_minus = (Button)findViewById(R.id.minus);
    bt_cart=(ImageButton)findViewById(R.id.cart);
    LinearLayout linear = (LinearLayout) findViewById(R.id.Left_layout);

    count=1;
    textV2.setText(""+count);

    nameList = new ArrayList<>();
    priceList = new ArrayList<>();
    gradeList = new ArrayList<>();

    socketConnect("192.168.0.53", 8402);    //소켓 연결
    sendData("s");    //사료 테이블 요청

    Button btn[] = new Button[nameList.size()]; //버튼 추가
    for(int i=0; i<btn.length; i++) {
        btn[i] = new Button(this);
        btn[i].setText(nameList.get(i));
        btn[i].setId(i);
        linear.addView(btn[i]);

        final int finalI = i;
        btn[i].setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                select_v.setText(nameList.get(finalI));
                set_price_v.setText(priceList.get(finalI).replaceAll(",",""));
            }
        });
    }
}

```

```

        order = finalI;
        count = 1;
        textV2.setText("" + count);
    }
});
}
select_v.setText(nameList.get(0));
set_price_v.setText(priceList.get(0).replaceAll(",", ""));
}

public void onClick(final View v) {
    price = priceList.get(order).replaceAll(",", "");
    switch (v.getId()){
        case R.id.plus:
            count++;

            if(count > 10){
                count = 10;
                Toast.makeText(this, "한 번에 10개 이상 구매할 수 없어요..",
                    Toast.LENGTH_SHORT).show();
            }
            textV2.setText(""+count); //그대로 넣으면 문자열로 인식 X 그래서 큰따옴표 붙여줌
            set_price_v.setText("" + Integer.parseInt(price)*count);
            break;

        case R.id.minus:
            count--;

            if(count < 1){
                Toast.makeText(this, "최소 하나 이상 구입해주세요!", Toast.LENGTH_SHORT).show();
                count = 1;
            }

            textV2.setText(""+count);
            set_price_v.setText("" + Integer.parseInt(price)*count);
            break;

        case R.id.cart:
            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {

                @Override
                public void run() {

                    item = select_v.getText().toString();
                    total_price = set_price_v.getText().toString();
                    temp = String.valueOf(count);

                    ArrayList<String> cart_name_list =
                        getIntent().getStringArrayListExtra("cart_name_list");

```

```

        ArrayList cart_count_list =
getIntent().getStringArrayListExtra("cart_count_list");
        ArrayList cart_price_list =
getIntent().getStringArrayListExtra("cart_price_list");

        cart_name_list.add(item);
        cart_count_list.add(temp);
        cart_price_list.add(total_price);

        Intent intent = new Intent(pet_SaryoActivity.this, pet_CartActivity.class);
        intent.putExtra("cart_name_list", cart_name_list); //putextra로 cart액티비티에
넘김
        intent.putExtra("cart_price_list", cart_price_list);
        intent.putExtra("cart_count_list", cart_count_list);

        startActivity(intent);
        finish(); //화면 종료
    }
    }, 500); //0.5초 후 카트액티비티로 넘어감
}
}

public void sendData(String data){ //데이터 발신
    Log.d("Send", data);

    try {
        ByteBuffer buffer = null;
        Charset charset = Charset.forName("UTF-8");
        buffer = charset.encode(data);

        socketChannel.write(buffer);

        buffer = ByteBuffer.allocate(15000);
        int len = socketChannel.read(buffer);
        buffer.flip();

        String recive = charset.decode(buffer).toString();
        ReceiveData(recive);

    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void socketConnect(String host, int port){
    //연결 부분
    try{
        socketChannel = SocketChannel.open();
        socketChannel.configureBlocking(true);
        socketChannel.connect(new InetSocketAddress(host, port));
    }
}

```

```

        } catch (Exception e){
            e.printStackTrace();
        }
    }

    @Override
    public void finish() {
        if(socketChannel.isOpen()){
            try {
                socketChannel.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        super.finish();
    }

    public void ReceiveData(String recive){    //데이터를 받으면

        JSONArray jsonArray = null;
        try {
            jsonArray = new JSONArray(recive);
            for(int i = 0; i<javascriptArray.length(); i++){
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                String name = jsonObject.getString("name");
                String price = jsonObject.getString("price");
                String grade = jsonObject.getString("grade");
                nameList.add(name);
                priceList.add(price);
                gradeList.add(grade);
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    public void onBackPressed(){
        Intent intent = new Intent(pet_SaryoActivity.this, pet_MarketActivity.class);
        startActivity(intent); //마켓으로 이동
        finish();
    }
}

```

바. pet_activity_saryo.xml

```

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".pet_SaryoActivity"
    android:layout_width="match_parent"

```

```
android:layout_height="match_parent"
android:background="#ffffff">
```

```
<android.support.constraint.Guideline
    android:id="@+id/guideline50per"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.57" />
```

```
<ScrollView
    android:id="@+id/scrollView"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:background="@drawable/pet_pattern3"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline50per"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
```

```
<LinearLayout
    android:id="@+id/Left_layout"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="20dp"
    android:scrollbarStyle="insideInset">
```

```
</LinearLayout>
```

```
</ScrollView>
```

```
<LinearLayout
    android:background="#ffffff"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/guideline50per"
    app:layout_constraintTop_toTopOf="parent"
    android:id="@+id/Right_layout"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0"
    android:background="@drawable/pet_early_buy"
    android:gravity="center"
    android:orientation="vertical">
```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:orientation="vertical">
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="7dp">

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="선택 상품"
        android:textColor="#000000"/>

    <TextView
        android:id="@+id/select_saryo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textColor="#000000"
        android:layout_gravity="center"
        android:gravity="center"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:orientation="horizontal"
    android:padding="7dp">

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:text="가격"
        android:textColor="#000000" />

    <TextView
        android:id="@+id/set_price_v"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```



```

        android:layout_weight="1"
        android:textColor="#000000"
        android:layout_gravity="center"/>

<TextView
    android:id="@+id/textView9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="원"
    android:textColor="#000000"
    android:layout_gravity="center"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:orientation="horizontal"
    android:padding="7dp">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:text="수량 :"
        android:textColor="#000000"
        android:layout_gravity="center"/>

    <Button
        android:onClick="onClick"
        android:id="@+id/minus"
        android:layout_width="12dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.5"
        android:background="@drawable/pet_button"
        android:text="-"
        android:layout_gravity="center"/>

    <TextView
        android:id="@+id/counting_textline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textAlignment="center"
        android:textColor="#000000"
        android:textSize="20dp"
        android:onClick="onClick"

```

```

        android:layout_gravity="center" />

<Button
    android:onClick="onClick"
    android:id="@+id/plus"
    android:layout_width="12dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.5"
    android:background="@drawable/pet_button"
    android:text="+"
    android:layout_gravity="center"/>
</LinearLayout>

<ImageButton
    android:onClick="onClick"
    android:id="@+id/cart"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_marginTop="10dp"
    android:background="#00000000"
    android:scaleType="fitXY"
    app:srcCompat="@drawable/pet_cart" />
</LinearLayout>
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```



[그림 III-1] android market 사료

사. pet_CartActivity.java

```

package com.example.shiba;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

```

```

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.WindowManager;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;
import java.net.InetSocketAddress;
import java.nio.ByteBuffer;
import java.nio.channels.SocketChannel;
import java.nio.charset.Charset;
import java.util.ArrayList;

public class pet_CartActivity extends AppCompatActivity {

    TextView num, total_price;
    Button buy_button;
    EditText pin_num, name;
    ListView cart_list;
    ArrayList<String> cart_items;
    Context context = this;
    Integer total = 0;

    SocketChannel socketChannel = null;

    ArrayList<add_Item> itemArrayList = new ArrayList<>();
    private ListviewAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.pet_activity_mycart);

        buy_button = (Button)findViewById(R.id.Buy);
        total_price = (TextView)findViewById(R.id.total_price);
        pin_num = (EditText)findViewById(R.id.pin_num);
        name = (EditText)findViewById(R.id.name);
        cart_list = (ListView)findViewById(R.id.cart_list);

        ArrayList cart_name_list = getIntent().getStringArrayListExtra("cart_name_list");
        ArrayList cart_count_list = getIntent().getStringArrayListExtra("cart_count_list");
    }
}

```

```

ArrayList cart_price_list = getIntent().getStringArrayListExtra("cart_price_list");

for(int i = 0; i < cart_name_list.size(); i++){
    String cart_name = cart_name_list.get(i).toString();
    String cart_price = cart_price_list.get(i).toString();
    String cart_count = cart_count_list.get(i).toString();
    add_Item a = new add_Item(cart_name, cart_price, cart_count);
    itemArrayList.add(a);

    total = total + Integer.parseInt(cart_price);
}

adapter = new ListViewAdapter(this, itemArrayList); //Adapter 설정
cart_list.setAdapter(adapter);

total_price.setText(total.toString());

socketConnect("192.168.0.3", 8205);    //7조 소켓 연결

buy_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        final String send_name = name.getText().toString();
        final String send_pin = pin_num.getText().toString();
        final String send_price = total_price.getText().toString();

        if(send_name.length() == 0){
            Toast.makeText(context, "구매자 이름을 확인해주세요.",
Toast.LENGTH_LONG).show();
        } else if (send_pin.length() == 0){
            Toast.makeText(context, "핀 번호를 확인해주세요.", Toast.LENGTH_LONG).show();
        }else {

            AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(context);

            // 제목 셋팅
            alertDialogBuilder.setTitle("※주의※");

            // AlertDialog 셋팅
            alertDialogBuilder
                .setMessage("정말 구매하시겠습니까?")
                .setCancelable(false)
                .setPositiveButton("확인",
                    new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {

                            String data = "dog " + send_pin + " " +
Integer.parseInt(send_price) + " " + send_name;
                            sendData("Shiba connected!");
                        }
                    }
                )
            .setNegativeButton("취소", null);

            alertDialogBuilder.create().show();
        }
    }
});

```

```

        sendData(data);

    }

    })

    .setNegativeButton("취소",
        new DialogInterface.OnClickListener() {
            public void onClick(
                DialogInterface dialog, int id) {
                // 다이얼로그를 취소한다
                dialog.cancel();
            }
        });

    // 다이얼로그 생성
    AlertDialog alertDialog = alertDialogBuilder.create();

    // 다이얼로그 보여주기
    alertDialog.show();
}

});
}

public void sendData(String data) {    //데이터 발신
    Log.d("Send", data);

    try {
        ByteBuffer buffer = null;
        Charset charset = Charset.forName("UTF-8");
        buffer = charset.encode(data);

        socketChannel.write(buffer);

        buffer = ByteBuffer.allocate(15000);
        int len = socketChannel.read(buffer);
        buffer.flip();

        String recive = charset.decode(buffer).toString();
        receiveData(recive);

    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void socketConnect(String host, int port) {
    //연결 부분
    try {
        socketChannel = SocketChannel.open();
    }
}

```

```

        socketChannel.configureBlocking(true);
        socketChannel.connect(new InetSocketAddress(host, port));

    } catch (Exception e) {
        e.printStackTrace();
    }
}

@Override
public void finish() {
    if (socketChannel.isOpen()) {
        try {
            socketChannel.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    super.finish();
}

public void receiveData(String receive){    //확인 데이터를 받으면
    Log.d("receive", receive);
    Intent intent = new Intent(pet_CartActivity.this, pet_MainActivity.class);
    switch (receive){
        case "nm":
            Toast.makeText(this, "잔액 부족!", Toast.LENGTH_SHORT).show();
        case "ok":
            Toast.makeText(this, "결제 완료!", Toast.LENGTH_SHORT).show();
            startActivity(intent); //결제 완료 시 화면 전환
            finish();
        case "fail":
            Toast.makeText(this, "전송 에러, 다시 입력!", Toast.LENGTH_SHORT).show();
    }
}

}

//리스트 뷰 설정
private class ListviewAdapter extends BaseAdapter {
    Context context;
    ArrayList<add_Item> itemlist = new ArrayList<>();

    public ListviewAdapter(Context context, ArrayList<add_Item> itemlist) {
        this.context = context;
        this.itemlist = itemlist;
    }

    @Override
    public int getCount() {
        return itemlist.size();
    }
}

```

```

    }

    @Override
    public Object getItem(int i) {
        return itemlist.get(i);
    }

    @Override
    public long getItemId(int i) {
        return i;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        if(convertView == null) convertView = new ListviewItem(context);
        ((ListviewItem)convertView).setData(itemlist.get(position));
        return convertView;
    }
}

//리스트뷰 속 아이템 설정
private class ListviewItem extends LinearLayout {
    TextView list_select_name, list_select_price, list_select_count;

    public ListviewItem (Context context){
        super(context);
        init(context);
    }

    private void init(Context context) {
        View view = LayoutInflater.from(context).inflate(R.layout.pet_list_item, this);
        list_select_name = (TextView)findViewById(R.id.select_name);
        list_select_price = (TextView)findViewById(R.id.select_price);
        list_select_count = (TextView)findViewById(R.id.select_count);
    }

    public void setData(add_Item one) {

        list_select_name.setText(one.getName());
        list_select_price.setText(one.getPrice());
        list_select_count.setText(one.getCount());
    }
}

private class add_Item {
    private String name, price, count;

    public add_Item(String name, String price, String count) {
        this.name = name;
        this.price = price;
    }
}

```

```

        this.count = count;
    }

    public String getName() {
        return name;
    }

    public String getPrice() {
        return price;
    }

    public String getCount() {
        return count;
    }
}

public void onBackPressed(){
    ArrayList cart_name_list = getIntent().getStringArrayListExtra("cart_name_list");
    ArrayList cart_count_list = getIntent().getStringArrayListExtra("cart_count_list");
    ArrayList cart_price_list = getIntent().getStringArrayListExtra("cart_price_list");

    Intent intent2 = new Intent(pet_CartActivity.this, pet_MarketActivity.class);
    intent2.putExtra("cart_name_list", cart_name_list);
    intent2.putExtra("cart_price_list", cart_price_list);
    intent2.putExtra("cart_count_list", cart_count_list);

    startActivity(intent2);    //마켓 화면으로 전환
    finish();
}
}

```

아}. pet_activity_cart.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.constraint.ConstraintLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        app:layout_constraintEnd_toEndOf="@+id/guideline_mycart"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

```



```

<ScrollView
    android:background="@drawable/pet_pattern2"
    android:id="@+id/scrollView2"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    tools:ignore="MissingConstraints">

    <LinearLayout
        android:id="@+id/Left_layout2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="10dp">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="52dp"
            android:text="MY CART"
            android:textColor="#6258C4"
            android:textSize="30dp"
            android:textStyle="bold" />

        <ListView
            android:id="@+id/cart_list"
            android:layout_width="match_parent"
            android:layout_height="282dp"
            android:background="#00B68282" />
    </LinearLayout>
</ScrollView>
</android.support.constraint.ConstraintLayout>

<android.support.constraint.Guideline
    android:id="@+id/guideline_mycart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.62243503" />

<LinearLayout
    android:id="@+id/Right_layout2"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@id/guideline_mycart">

    <LinearLayout
        android:layout_width="match_parent"

```

```

android:layout_height="match_parent"
android:background="#6258C4"
android:gravity="center"
android:orientation="vertical"
android:textStyle="bold">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:orientation="horizontal">

```

```

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="결체"
        android:textSize="30dp"
        android:textStyle="bold" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp">

```

```

    <TextView
        android:layout_width="1dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="이름 : "
        android:textStyle="bold" />

```

```

    <EditText
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:hint="이름을 입력하세요."
        android:textSize="10dp"
        android:textStyle="bold" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp">

```

```

<TextView
    android:layout_width="28dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="배 송지 : "
    android:textStyle="bold" />

<TextView
    android:id="@+id/home"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:text="인천전자마이스터고등학교"
    android:textSize="10dp"
    android:textStyle="bold" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp">

    <TextView
        android:layout_width="4dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="은행 : "
        android:textStyle="bold" />

    <TextView
        android:id="@+id/bank"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:text="카카오뱅크"
        android:textSize="15dp"
        android:textStyle="bold" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp">

    <TextView
        android:layout_width="46dp"

```

```

        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="핀 번호 : "
        android:textStyle="bold" />

```

```

<EditText
    android:id="@+id/pin_num"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:hint="핀 번호(6자리)를 입력하세요."
    android:textSize="10dp"
    android:textStyle="bold" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="5dp">

```

```

    <TextView
        android:layout_width="23dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:gravity="center"
        android:text="총 금액"
        android:textSize="20dp"
        android:textStyle="bold" />

```

```

    <TextView
        android:id="@+id/total_price"
        android:layout_width="82dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:gravity="center"
        android:textSize="20dp" />

```

```

    <TextView
        android:layout_width="6dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:gravity="center"
        android:text="원"
        android:textSize="20dp"

```

```

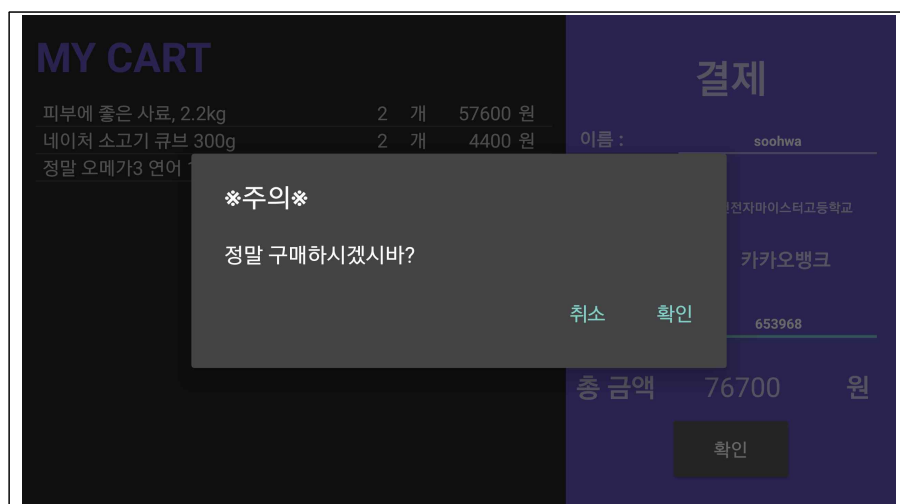
        android:textStyle="bold" />
</LinearLayout>

<Button
    android:id="@+id/Buy"
    android:layout_width="wrap_content"
    android:layout_height="45dp"
    android:layout_gravity="center"
    android:layout_marginTop="5dp"
    android:text="확인" />
</LinearLayout>
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```



[그림 III-1] cart 화면



[그림 III-1] cart 구매 화면

4 실무 결과

서버를 하면서 통신에 대해 많이 알게 되었다. 처음에는 웹서버와 헷갈려 헤매기도 하였지만 그 덕에 웹서버에 대해서도 조금이나마 알게 되는 시간이었다. 통신을 하며 여러 오류들을 만나면서 어떤 점에 대해 주의를 해야 하고 어떤 것이 중요한지 차차 깨닫게 되었다. 서버 라즈베리파이와 본체용 라즈베리파이를 통신하고 나니 마켓 서버에 클라이언트의 입장으로 데이터를 주고 받는 것도 어렵지 않게 느껴져 금방 성공할 수 있었다.

III. Reflection

1 느낀점

매년 프로젝트를 하면서 내가 발전되고 있다는 느낌이 든다. 선생님의 말씀대로만 행동해왔던 1학년 때의 나, 조원들과 상의해가면서 우리가 생각하는 데로 실행했지만 아직은 서툴렀던 2학년의 내가 생각한다. 하지만 이제는 기능에 대해 생각하면서 어떻게 구현해나갈 것인지, 구현 가능한 기능인지 머릿속으로 정리도 하고 부족한 부분은 인터넷, 선생님, 친구 등 여러 도움을 받으며 최대한 스스로 성공시키려 노력했다. 이번 프로젝트를 진행하면서 배운 점들이 많은 것 같아 뿌듯하다. 서버를 담당하면서 초반에는 무엇을 해야 하는지, 어디부터 시작해야 하는지 몰라서 잘 하는 친구의 소스를 받아 이해하면서 따라했지만 어느새 내가 다른 친구를 도와주고 있는 모습을 발견할 수 있었다. 전공뿐만 아니라 조원들과의 수많은 회의를 하면서 상대방의 의견을 존중하고 조율해나가는 방법 또한 배웠다고 할 수 있을 것 같다. 하드웨어에는 내 손이 닿은 곳을 찾기 힘들지만 서버와 안드로이드에서 많이 찾을 수 있기 때문에 아쉬운 마음은 없다. 중간엔 하루 종일 코딩만 하다보니까 계속 앉아 있어 몸이 답답한 감이 없지 않아 있어서 차라리 내가 하드웨어를 하고 싶다는 생각이 들기도 했다. 그때 포맥스를 자르러 갔는데 하나 자르는 데도 힘이 굉장히 많이 들고 쉽지 않았다. 역시 뭐든지 쉬워 보인다고 만만하게 보면 안 되는 것 같다. 우리 조원을 포함해서 다른 조 친구들에게 전부 수고했다고 한마디 해주고 싶다. 항상 프로젝트를 할 때는 원하는 대로 되지 않아 스트레스를 받기도 하고 2학기가 되면 대부분의 수업 시간에 프로젝트를 하다 보니 지쳐있을 때가 많지만 그만큼 보상받는 것이 많다고 생각하기 때문에 몇 년 후 지금을 되돌아볼 때면 좋은 경험으로 남아있을 것 같다.

2 시사점

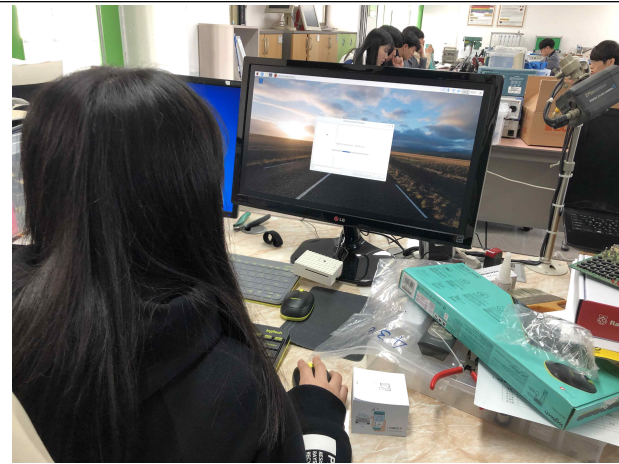
프로젝트를 준비하는 중간에 조원 한 명을 제외한 나 포함 세 명이 현장실습을 나가게 되어 혼자 남은 성진이가 대부분의 역할을 맡게 되었다. 역할 분담이 제대로 이루어지지 않은 것이다. 현장실습을 나가더라도 각자 자기의 파트는 수행했어야 했지만 그렇게 하지 못해 성진이에게 미안한 마음이 들었다.

본체를 회전시키기 위해 스텝모터를 사용하였다. 모터의 축에 본체부분을 붙인 것이기 때문에 스텝모터 하나로 모든 것을 지탱해야 했다. 또 웹캠과 간식을 던져주기 위한 부품들이 한쪽으로 몰리게 되어 회전을 할 때마다 무게중심이 잘 맞지 않아 한쪽으로 기울어지곤 했다. 회전할 때마다 약간씩 흔들려 조마조마했다. 이 점에 대해서 굉장히 아쉬운 마음이 크다. 스텝모터와 본체를 조금 더 단단하게 고정시켰다면 흔들림이 많이 줄지 않았을까 생각이 들었다.

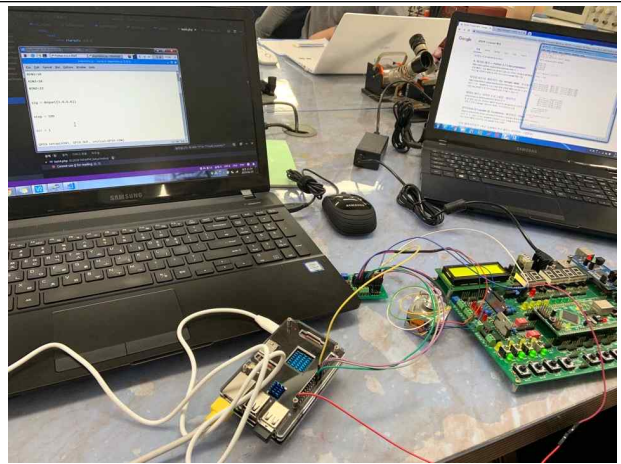
프로젝트를 계획하면서 초반에 생각한 기능들이 많이 있는데 중간에 여러번 바뀌게 되었다. 처음 생각할 때는 뭐든지 구현 가능할 것이라고 생각하여 계획했지만 지금 돌아보니 너무 마음만 앞선 것 같다. 크게 생각하고 점점 줄여나가는 것이 아닌 작게 생각하고 하나씩 늘여갔어야 했다.

IV. Activity

1 활동사진



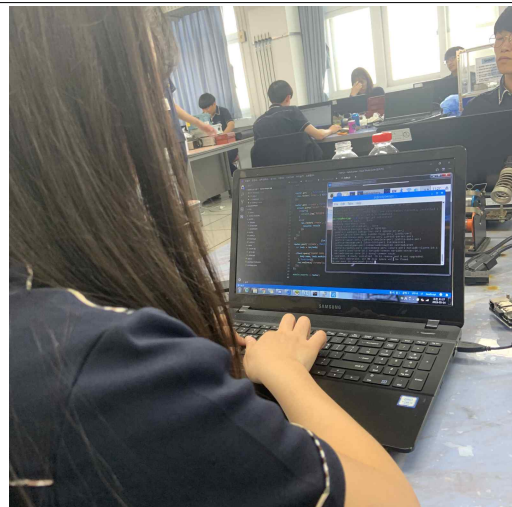
[그림 IV-1] 라즈베리파이 초기 설정



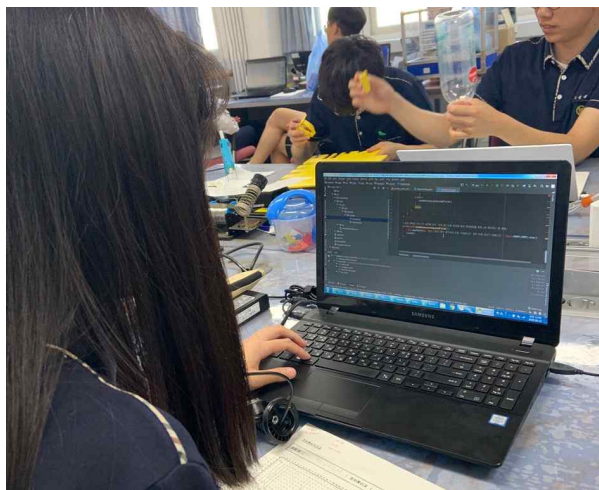
[그림 IV-2] 스텝모터 구동



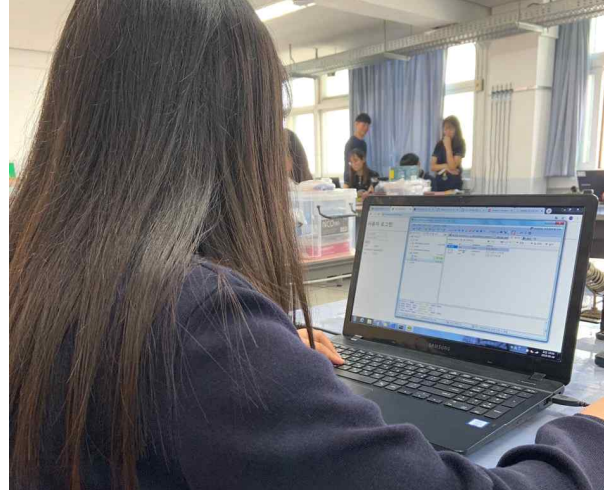
[그림 IV-3] 중간 발표



[그림 V-4] 밥 쥐, 시바 서버 구축



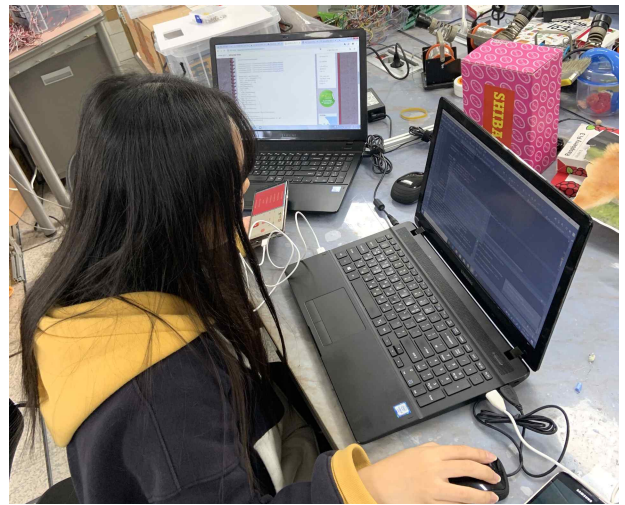
[그림 IV-5] 밥 쥐, 시바 서버 구축



[그림 IV-6] DB 설정 및 정리



[그림Ⅳ-7] 구조물 수평 맞추기



[그림Ⅳ-8] 안드로이드 마켓 부분 구현

참고문헌

- 1) Jade_, Fall in IT, “node.js 특징”, <https://ithub.tistory.com/32>, 2019-10-29
- 2) sblogforyou, 21세기 4차산업 르네상스 :b, “라즈베리파이란”,
<https://21innovation.tistory.com/58>, 2019-10-29
- 3) 비타민씨, 비타민씨의 포만감성라이프, “안드로이드 스튜디오 특징”,
<http://blog.naver.com/PostView.nhn?blogId=vitaminsense&logNo=220680127586>, 2019-10-29
- 4) 형준, public:, “라즈베리파이 초기 설정”, <https://m.blog.naver.com/hyungjun212/221217407894>
- 5) 심해펭귄, 심해펭귄의 심해도서관, “node.js 소켓통신”,
<https://m.blog.naver.com/PostView.nhn?blogId=platinasnow&logNo=220024486295&proxyReferer=https%3A%2F%2Fwww.google.com%2F>