

종합설계 프로젝트 최종 보고서

Smart Box

한성대학교

IT응용시스템공학과

1594038 주 혜원

1694029 이 민정

1694036 이 진솔

지도 교수 : 김 승천 교수님

차 례

- I . 프로젝트 개요
- II . 시스템 구성
- III . 모듈별 동작 원리
- IV . 모듈 설계
- V . 어플리케이션 설계
- VI . 제작내용 (회로도, 소스코드)
- VII . 결과물 설명
- VIII . 종합설계프로젝트 소감문

I. 프로젝트 개요

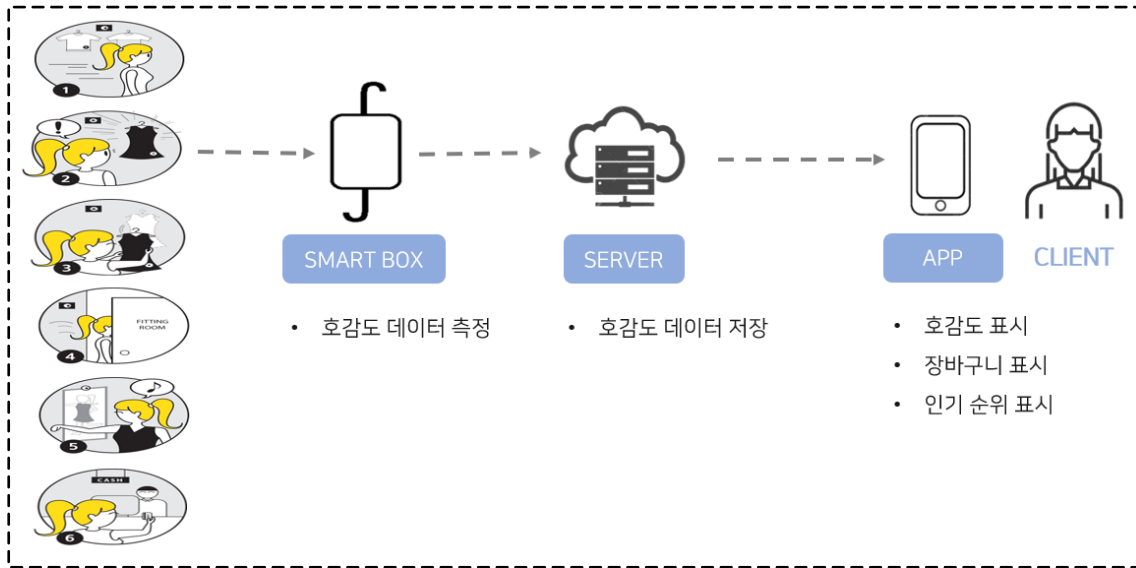
판매량만 측정하던 기존 상용브랜드의 평가 시스템에는 한계가 있고 메가트랜드 변화로 현대인의 소비패턴의 변화와 유행의 빠른 변화로 인해 빠른 구매패턴 분석이 필요하며, 마케팅 방식 또한 상품, 가격, 유통, 촉진의 4P에서 고객가치, 고객측의 비용, 편리성, 소통의 4C로 변화함에 따라 위 사항들을 아우를수 있는 소비자중심의 신속한 서비스가 필요하다 판단했습니다.

소비자를 중심으로 생각하다 보니 소비자 행동에 초점이 맞춰졌습니다. 소비자 행동이란 소비자가 제품의 구매 과정 전반에 걸쳐서 내리는 모든 의사결정입니다. 소비자가 자신의 문제해결을 위해 중요하게 고려하는 가치를 분석하고, 그러한 고객가치를 얻기 위한 제품을 생산하여 소비자의 니즈를 충족시키는 소비자 중심적 사고의 마케팅을 위해서는 필수적으로 제품의 차별화된 속성과 기능을 통해 소비자의 기능적 니즈를 완벽히 충족시켜줄 뿐만 아니라 소비자의 상징적, 체험적 니즈 역시 동시에 만족시킬 수 있어야합니다.

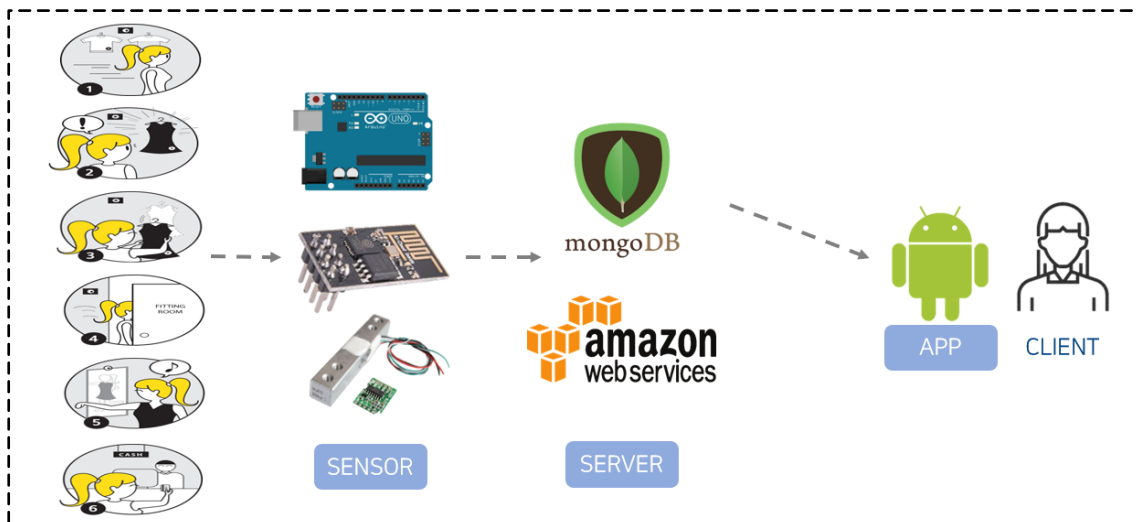
이와 관련된 이론으로는 AISAS이론이 있습니다. AISAS이론은 소비자가 어떤 물건이나 서비스를 구매하기까지의 지각과 행동단계를 나타내는 이론으로 주의, 흥미, 검색, 구매, 공유 5가지의 단계를 가집니다.

저희가 만들고자하는 스마트박스를 통해 고객이 어떤 제품에 주의를 하고, 얼마나 흥미를 가지는지 알 수 있고, 다양한 사용자들의 행동을 토대로 검색을 돕고 제품의 구매까지 이어지며, 자동으로 사용자의 구매 행동이 공유되어 사용자들과 제품 마케팅에 도움을 줄 수 있게 하고 싶었습니다. 저희 팀은 매장에 자주 방문하여 옷을 구매하고, 유행에 민감하며, 지금보다 더 편리하게 쇼핑을 하고 싶어 하는 2050 소비자들을 타겟 군으로 설정하여 이들의 니즈를 충족시키는 스마트 매장 시스템을 구축하는 것이 목적입니다.

II . 시스템 구성



<그림 1 . 시스템 구성도>



<그림 2 . 세부적인 시스템 구성도>

위 그래프는 전체적인 시스템 아키텍처입니다. 먼저 매장 내 고객의 행동유형을 아두이노 UNO에 연결한 로드셀 무게 센서를 활용하여 분석합니다. 행동유형에 따른 무게 값들을 측 받아온 뒤 짜여진 알고리즘을 통해 가공된 정보로 바뀝니다. 행동유형의 기준으로는 크게 옷을 건드려봄, 들어봄, 입어봄 세 가지로 행동유형을 구분합니다. 구분된 행동유형은 와이파이 센서를 통해 몽고디비와 Amazon Web Service로 구축된 서버로 들어가게 됩니다. 안드로이드 어플리케이션에는 서버에 저장된 정보를 통해 고객들에게 유용한 정보를 제공합니다. 매장 내에 있는 제품들의 인기순위를 보여주고 바로 장바구니에 담을 수 있습니다. 또한 Smart Box에 붙여있는 QR코드를 인식해서 장바구니에 담을 수도 있습니다. 스마트 박스에 QR코드를 이용해서 담은 물품들을 리스트 형식으로 볼 수 있습니다.

Ⅲ . 모듈별 동작 원리

모듈	설명	사진
Arduino Uno	ESP8266과 Load Cell을 연결하여 Smart Box의 전반적인 제어를 담당	
ESP8266	WIFI 모듈. Load Cell을 통해 측정된 정보를 서버로 전달하는 역할	
HX711 Load Cell	옷걸이와 옷걸이에 걸린 옷의 무게를 측정하여 짜여진 알고리즘을 통해 옷의 건드려짐, 옷의 들어짐, 옷의 입혀짐을 측정	
Android	Smart Box를 통해 얻은 정보를 활용하여 고객들에게 도움이 되는 어플리케이션 제작 주요기능1. 매장별 상품 인기순위 확인 주요기능2. QR코드 인식하여 장바구니에 담기	
MongoDB	MongoDB와 Amazon Web Service를 활용하여 Smart Box의 서버를 구축함	

IV . 모듈 설계

Smart Box 초기 모델	
Smart Box 최종 모델	Smart Box 최종 모델 내부 모습

Smart Box의 초기 모델 설계 당시에는 실력이 부족하다 보니 하나의 행동유형을 측정할 때 마다 각기 다른 센서들을 사용하였습니다. 하지만 자꾸 측정할 때 오류가 나고, 값도 원하는 대로 잘 나오지 않았습니다. 그래서 센서들을 교체도 해보고, 센서 위치도 바꿔보고 하였는데도 결과가 좋아지지 않았습니다. 그리고 초기의 방법으로 계속 진행을 하게 되면 옷걸이 단가가 너무 올라가고, 소모하는 전력량이 많아서 충전을 해주기 너무 어려우며 신발이나 다른 잡화들 말고 오직 의류에 한해서만 사용이 가능하다는 단점이 있었습니다. 이러한 단점들을 극복할 방안이 무엇일까 고민해보다가, 무게센서 하나만 사용해서 이 모든 것을 다 해결할 수 있는 법을 고안해내었습니다. 무게센서 하나만 사용해서 모듈화를 진행하였고, 이로 인해 충전 문제와 의류에 한해서만 사용이 가능하다는 단점을 극복하였습니다. 또한 충전은 5V 건전지를 갈아 끼워주면 해결되는 간단한 문제였습니다. 그리고 모듈화를 하게 되면 어떻게 모듈을 위치하느냐에 따라 옷걸이처럼 걸면 의류, 뒤집어서 무엇인가를 올려서 측정을 하게 되면 신발이나 모자 등 다양하게 사용 가능하도록 아크릴을 이용하여 제작을 하였습니다.

V. 어플리케이션 설계



왼쪽의 화면은 앱의 첫 화면입니다.

사용자가 회원가입을 할 수 있는 버튼이 있으며, 회원가입이 완료된 사용자는 바로 로그인을 할 수 있도록 구현되어 있습니다.



왼쪽의 화면은 앱의 회원가입 화면입니다.

사용자는 이름과 로그인에 필요한 ID, PASSWORD를 작성하게 됩니다. 사용자가 입력한 값을 그대로 서버에 보냅니다. 서버에서 값을 암호화 하여 저장합니다.



왼쪽의 화면은 스마트 박스를 통해서 얻은 호감도 값을 이용하여 매장에 걸려있는 옷들을 호감도가 높은 순서대로 띄운 화면입니다. 사용자들은 이러한 리스트를 이용하여 매장 내에서 쇼핑을 할 수 있습니다.



왼쪽의 화면은 장바구니에 옷을 담은 화면입니다. 이러한 다이얼로그를 통해 장바구니에 옷을 담을 수 있습니다.

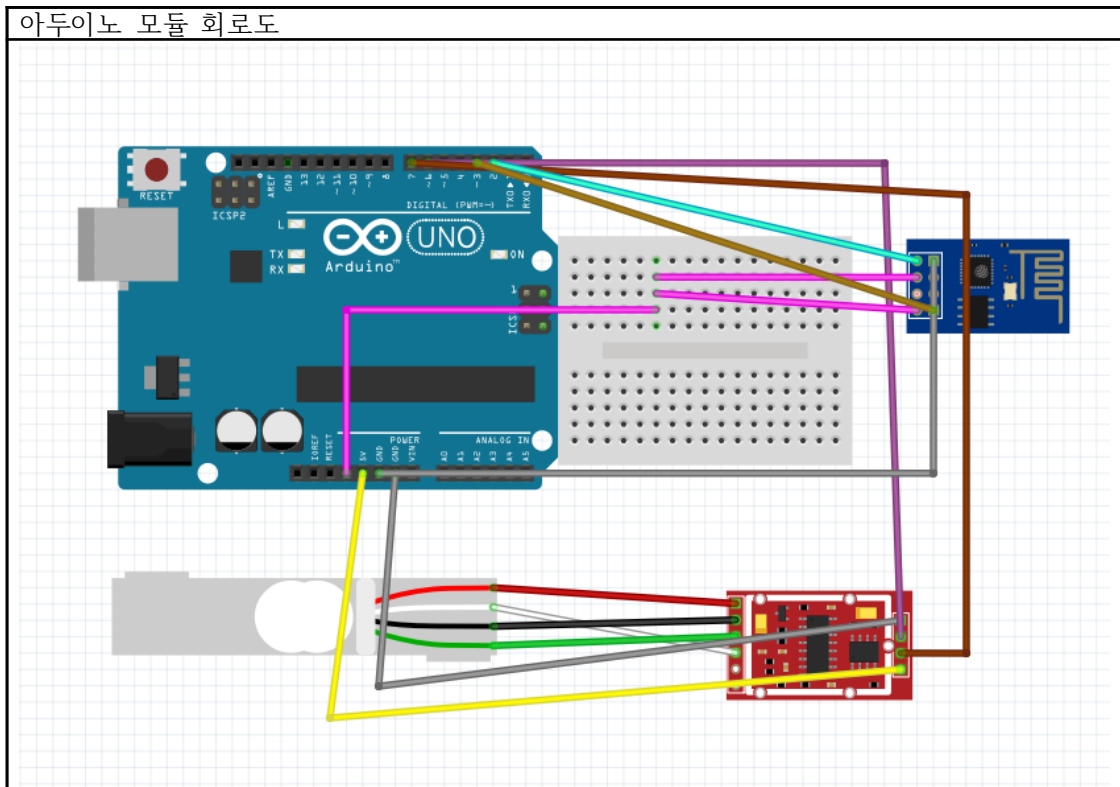


왼쪽의 화면은 장바구니에 옷이 담긴 화면입니다. 장바구니에 담길 때는 옷의 사진과 호감도 값이 함께 담기게 됩니다.

구매하기 버튼을 누르면 구매하는 액티비티로 이동합니다.

VI . 제작내용 (회로도, 소스코드)

아두이노 모듈 회로도



아두이노 모듈 소스코드

```
// 호감도 잘 합산되고 들어봄, 입어봄이 추가될때만! thingspeak에 올라감(터치 --)
// 시작하고 4초이내에 걸어야하는 문제 / 음..
// 1번 옷걸이
#include "HX711.h" // 초하빨검
#include <Wire.h>
#include <SoftwareSerial.h>

#define calibration_factor - 7050.0
#define DOUT 6 // dt
#define CLK 7 // sck
HX711 scale(DOUT, CLK);
int count = 0;
double w = 0;
double pre_w = 0;
int touch_count = 0;
//int touch_count = -1;
int up_count = 0;
int put_count = 0;
int pre_up = 0;
int pre_put = 0;
```

```
int favorite = 0;
boolean timeSet = false; //타이머가 실행되는지 여부
boolean dataSet = false; //데이터가 들어왔는지 여부

unsigned long startTime = 0;
unsigned long finishTime = 0;
unsigned long middleTime = 0;
unsigned long dataTime = 0;

//String apiKey = "7JPQA4FCBZBO158T"; // hanger_test

//8EQ0ZZTUZZA6C97Y minjung_test

SoftwareSerial esp8266(2, 3); // RX/TX 설정, serial 객체생성

boolean test = false; //업로드할때 무게 0점

void setup()
{
    Serial.begin(9600);
    scale.set_scale(calibration_factor);
    scale.tare();

    esp8266.begin(9600); //소프트웨어 시리얼 시작
    esp8266.println("AT+RST"); //ESP8266 Reset
}

void loop()
{
    if(!test){ //test가 false일때
        Serial.println(" Test Start ");
        delay(4000);
        test =true;
    }
    delay(1500);
    unsigned long currentTime = millis();
    w = scale.get_units();

    Serial.print("무게 = ");
```

```

Serial.println(w);

// 터치
if(w - pre_w > 1 || w - pre_w < -1){
    pre_w = w;
    touch_counter();
}

// 들렸을때
if(-1<w && w<=1){
    //if(w<=1){// 시간 측정 시작
    if(startTime==0){ // 시간 측정 시작
        if(!timeSet){
            startTime = currentTime;
        }
        timeSet = true;
        Serial.print("time Start==");
        Serial.println(startTime/1000);
    }
}

if(w>=1&&startTime!=0){

    if(w>4){ // 옷+옷걸이 걸린 무게
        timeSet = false;
        finishTime = currentTime;
        up_counter();//들어봄
    }
    else if(3<w<4){ // 옷걸이 무게만 걸린거
        timeSet = false;
        middleTime = currentTime;
        put_counter();//입어봄
    }
    else{
        Serial.println("No if");
        delay(1500);
    }
}

```

```

        checkTime(dataSet);
    }//loop끝남

void checkTime(boolean dataSet){
    // delay(1500);
    if(dataSet){

        if(!timeSet){
            // delay(1500);

            //실제 데이터 쓰는부분    finishTime, middleTime 이계 0이 아닌 경우에 dataTime을
            계산해
            //    dataTime = finishTime - startTime;
            //////////////////////////////////////

            favorite = touch_count + (up_count*5) + (put_count*15) ;

            ///초기화
            //    dataTime = 0;
            finishTime = 0;
            middleTime = 0;
            startTime = 0;
            timeSet = false;
            dataSet = false;

            Serial.print("터치함==");
            Serial.println(touch_count);
            Serial.print("들어봤다==");
            Serial.println(up_count);
            Serial.print("입어봤다==");
            Serial.println(put_count);
            Serial.print("호감도==");
            Serial.println(favorite);

            if(up_count - pre_up >= 1 || put_count - pre_put >= 1){
                pre_up = up_count;
                pre_put = put_count;

                //와이파이코드
                String cmd = "AT+CIPSTART=\"TCP\", \"";

```

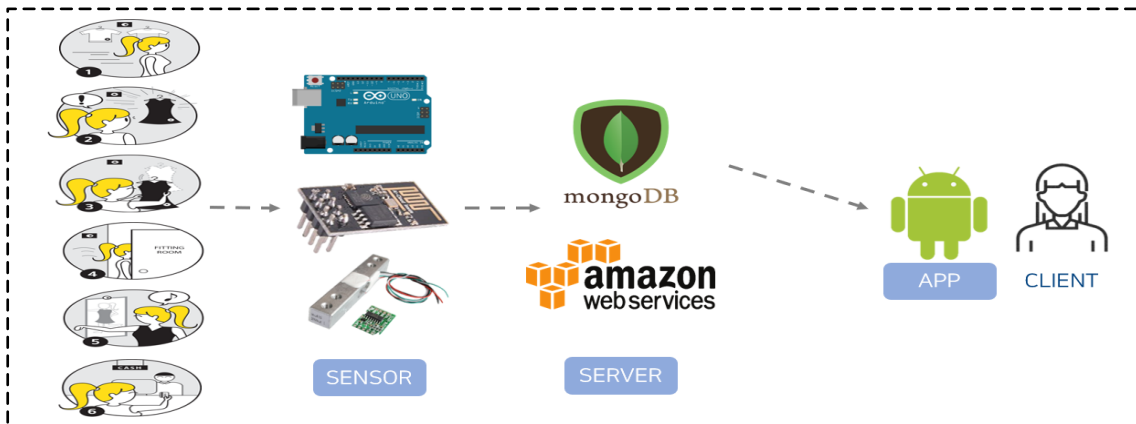
```
cmd += "52.78.132.239"; // server 접속 IP
cmd += "\",3000";          // 접속 포트, 3000
esp8266.println(cmd);
if(esp8266.find("Error")){
    Serial.println("AT+CIPSTART error");
    return;
}
// GET 방식으로 보내기 위한 String, Data 설정
String getStr = "GET /list/relist?name=";
getStr += "myj";
getStr += "&favorite=";
getStr += String(favorite);
getStr += "&touch=";
getStr += String(touch_count);
getStr += "&shake=";
getStr += String(up_count);
getStr += "&tryon=";
getStr += String(put_count);
getStr += "\r\n\r\n";
// Send Data
cmd = "AT+CIPSEND=";
cmd += String(getStr.length());
esp8266.println(cmd);
if(esp8266.find(">")) {
    esp8266.print(getStr);
    Serial.println("@send!!");
}
else {
    esp8266.println("AT+CIPCLOSE");
    Serial.println("AT+CIPCLOSE"); // alert user
}
delay(1000); // Thingspeak 최소 업로드 간격 15초를 맞추
```

VII . 결과물 설명

Smart Box를 구현하기 위해 Nodejs 서버, 아두이노를 활용한 Smart Box, 어플리케이션으로 서비스 플랫폼을 구성하였으며, restful 방식으로 통신하고, 각 파트에 대한 설명은 다음과 같습니다.

1. Server

- 서버는 AWS를 활용하여 환경에 유연하게 적응할 수 있게 구현하였고, NodeJS를 통해 서버를 구축 하였습니다. 서버는 그림3처럼 아두이노를 활용한 Smart Box와 사용자 어플리케이션간의 통신을 담당합니다.



<그림 3 . 세부적인 시스템 구성도>

데이터를 축적하기 위해 mongoDB를 활용하였으며, 주로 회원가입을 위한 사용자의 데이터를 저장하거나, Smart Box에서 측정한 데이터 값을 축적하는데 사용하였습니다. 사용자의 데이터를 저장하고, Smart Box에서 측정한 데이터를 저장한 mongoDB의 스키마는 그림4와 같습니다.

```
const cartSchema = new Schema({
  user_id: {
    type: String,
    required: true,
  },
  cart_list: [{
    list_name: {
      type: String,
      default: null,
    },
  }],
},
{
  versionKey: false,
})
module.exports = mongoose.model('cart', cartSchema)
```

```
const listSchema = new Schema({
  list_name: {
    type: String,
    required: true,
  },
  list_favorite: {
    type: Number,
    required: true,
  },
  list_touch: {
    type: Number,
    required: true,
  },
  list_shake: {
    type: Number,
    required: true,
  },
  list_tryon: {
    type: Number,
    required: true,
  },
  list_picture: {
    type: String,
    required: true,
  },
}, {
```

```
const userSchema = new Schema({
  user_id: {
    type: String,
    required: true,
  },
  user_pw: {
    type: String,
    required: true,
  },
  user_name: {
    type: String,
    required: true,
  }
}, {
  versionKey : false
})
```

<그림 4 . 스키마>

사용자의 데이터를 저장하며 보안적인 측면을 강화하기 위해, hash암호화와, salt를 사용하여 사용자의 데이터의 보안성을 높였습니다. 또한 측정한 데이터의 오류를 줄이기 위해 측정한 값이 너무 커지게 되면 측정값을 버리고 다음 측정값을 기다리게 구현하여 오류를 줄이는 방법을 사용하였습니다.

그 외에 사용자에게 상품리스트를 제공하고, 사용자가 장바구니에 상품을 추가하는 기능을 구현하기 위해 위와 같이 스키마를 구성했습니다.

또한 상품리스트를 사용자에게 순차적으로 제공하기 위해 오름차순으로 정렬하여 사용자에게 제공하였습니다.

2. APPLICATION

어플리케이션은 안드로이드 스튜디오를 활용하여 구현하였습니다.

초기화면은 로그인 화면입니다. 회원가입을 통해 회원정보를 생성하고 로그인을 합니다.

서버와 restful 방식을 사용하여 http통신을 하여 회원정보를 post방식으로 전송하여 서버에 저장을 시키고, 로그인을 하게 되면 서버에서 로그인 정보가 있는지 확인한 후 결과값을 get방식으로 받아와서 로그인이 되거나, 회원정보가 맞지 않는다고 재로그인을 요구합니다.

로그인을 하게 되면 메인화면으로 액티비티가 전환됩니다. 메인화면에는 실시간으로 측정된 호감도를 바탕으로 인기순위가 나타나며 화면을 내리면 새로고침이 됩니다. 인기순위에 있는 제품을 선택하면 장바구니에 담을 것인지 아닌지 다이얼로그가 띄워지게 됩니다.

인기순위 밑에 있는 QR코드 버튼을 누르게되면 QR인식 화면이 나타나며 제대로 인식 되면 인식된 제품이 장바구니에 담기게 됩니다. 장바구니 버튼을 클릭하면 소비자가 장바구니에 다아왔던 제품들의 목록이 뜨게 됩니다. 장바구니 화면으로 넘어가는 순간 post방식으로 회원정보를 보내면 서버쪽에서 회원정보에 맞는 장바구니정보와 회원정보를 get 방식으로 받아옵니다. 받아온 정보를 리스트뷰 형식으로 사용자에게 제공해줍니다. 추후에는 장바구니에서 결제를 할 수 있도록 구현할 계획입니다.

3. Hardware

하드웨어는 보편적으로 사용 되고 있는 아두이노 우노 보드를 사용하여 제작하였습니다. 기본 오픈 소스가 많아서 스케치하는데 용이했습니다. 구현하고자 하는 기능은 옷과 옷걸이의 무게를 측정하는 것이었습니다. 아두이노 무게센서인 로드셀을 사용하여 구현하였습니다. 고객의 행동유형을 분석하는 알고리즘을 제작하였습니다. 맨 처음에는 아무것도 안 걸려있는 상태에서 시작하고, 옷이 걸려있는 옷걸이가 걸리게 되면 작동하는 조건을 걸었습니다. (전무게)-(현무게)=1 이하이면 “옷을 건드려봄”으로 측정합니다. “옷을 들어봄”과 “옷을 입어봄”은 같은 조건문 안에서 돌게 됩니다. 옷걸이와 옷이 같이 들린 뒤에 옷걸이의 무게만 다시 측정될 시에는 “옷을 입어봄”으로 측정이 되고 옷걸이와 옷의 무게가 다시 측정되면 “옷을 들어봄”으로 측정이 됩니다. 이 알고리즘을 생각해내는데는 얼마 안 걸렸지만 코드로 구현하는데에는 많은 시간이 들었습니다. 생각하는데로 잘 작동하지 않았고, 로드셀 무게센서의 오류도 큰 문제가 있었습니다. 로드셀 무게센서는 수평이 잘 맞아야만 값이 제대로 측정되는데, 초기에 구상했던 모듈은 수평이 잘 맞지 않아 값이 제멋대로 튀어 평균값을 잡아서 하기 힘들었습니다. 그래서 최종적으로 프로젝트를 완성해 나갈 때는 아크릴을 통해 수평을 완벽히 맞춰주는 모듈을 제작하였습니다.

1. 재학중 취득한 기초지식의 활용내용

4학년 1학기 수업중 웹앱 이란 수업에서 nodejs를 배워서 기본적인 nodejs 사용법을 알고 있었습니다. 따라서 종합설계프로젝트에서 서버를 구축할 수 있었고, 어플리케이션과의 통신하는 부분을 구축할 수 있었습니다.

2. 재학중 취득한 실험지식의 활용내용

웹앱 수업에서 실습한 nodejs 서버 구축하는 법을 사용하여 이번 종합설계프로젝트에서 서버구현을 조금 더 쉽게 할 수 있었습니다. 또한 오픈소스 HW수업에서 아두이노를 다뤄본 경험과 안드로이드 프로그래밍 수업을 들어 안드로이드 제작에 도움이 크게 되었습니다.

3. 본 프로젝트 수행과정에서의 설계 능력 향상 내용

이번 프로젝트를 통해 리눅스로 원격 접속하여 서버를 구축하는 법을 알게 되었고, 회원가입한 회원의 정보를 암호화하는 방법에 고민하고 암호화하는 방법을 알게 되었습니다. 또한 아두이노와 로드셀센서를 능숙하게 사용할 수 있게 되었습니다. 또한 어플리케이션을 제작하면서 한번도 새로고침이라는 기능을 구현해본적이 없었는데 졸업프로젝트를 진행하며 새로고침 기능을 구현할 수 있었습니다.

4. 본 프로젝트 수행과정에서의 문제 해결 내용

mongoDB에서 오름차순으로 데이터를 정렬하여 어플리케이션에 전달해주는 일이 잘 해결되지 않았습니다. 검색과, 다른 사람들이 비슷한 문제를 해결한 내용들을 참고하며 해결하였고, 버전과, 새로 바뀐 mongoDB 메소드를 사용하여 해결하였습니다.

또한 esp8266을 통해 nodejs서버와 통신하는 방법에서 서버측에서는 header부분으로 코드를 작성하였고, 아두이노쪽에서는 get일때는 param방식, post방식일때는 body방식으로 진행하여 오류가 있었으나 이것이 통신방법의 문제임을 깨닫고 get일때는 param post일때는 body방식으로 통일하여 통신문제를 해결하였습니다.

새로고침 기능을 구현하는데 있어서 메인화면을 불러오는 속도가 새로고침 됐을 때의 전송 속도와 달라서 제대로 구현되지 않았는데, 여러 시행착오를 거쳐서 메인화면을 불러오는 코드의 위치를 수정하여 오류를 해결하였습니다.

5. 본 프로젝트 수행과정에서의 실무 능력 향상 내용

이번 프로젝트를 진행하며 nodejs 와 mongodb의 사용에 능숙해졌고, 보안에 조금 더 신경 쓸 수 있었습니다. 또한 아두이노와 센서사용에 능숙해졌고 통신방법에 대해 보다 능숙해졌습니다. 안드로이드를 통해 어플리케이션을 제작하며 여러 라이브러리를 능숙하게 다룰 수 있게 되었고, JAVA를 사용하는 것 또한 익숙해졌습니다.

VIII . 종합설계프로젝트 소감문

한 학기 동안 종합설계프로젝트를 진행하면서 느낀 점이 많았습니다. 첫 번째, 분업화된 시스템을 간접적으로 느낄 수 있었습니다. 1차적으로 하드웨어와 소프트웨어 파트를 나누고, 백엔드와 프론트엔드를 나누어서 작업을 진행하였습니다. 이번 종합설계프로젝트에서는 앱은 주혜원 학생, 서버는 이진솔 학생, 하드웨어는 이민정 학생이 맡아서 진행하였습니다.

저희는 하루에 시간을 정해두고 서로 협동하여 프로젝트를 진행하였습니다. 시간을 정해두고 프로젝트를 진행한 이유는 서로 시간을 미루며 늘어지는 것을 방지하기 위함입니다.

하지만 실제로 개발에 들어간 이후로는 팀원들 모두가 부족한 코딩실력을 매우기 위해서 수업시간을 제외한 모든 시간을 밤낮을 가리지 않고 프로젝트에 쏟아 부어야했습니다..

시작을 할 때 Restful API를 워드 프로세서를 이용하여 작업하여 작업을 진행하는 동안 경로 문제가 생기지 않게 하였습니다. 파트별로 소통을 위해서 주별 보고서 제출과 매달 조별 회의를 통하여 아이디어의 부족한 부분을 충족시키려고 노력했습니다. 매달 조별회의에서는 서로의 코드를 이해하는데에 집중하였습니다.

처음 언어를 정할 때에는 수업 시간에 한 번씩 경험해보았던 언어(C언어, JAVA, JavaScript)를 사용하여 프로젝트를 진행하였습니다. 팀원 모두가 코딩에 익숙하지 않았고 졸업 프로젝트 내내 구현을 하는데에 어려움을 많이 느꼈습니다. 코딩 효율을 높이기 위해서 구글을 통한 검색으로 문제를 해결하는 능력을 많이 길렀습니다. 기본적인 문법을 활용하고, 반복적으로 작업하여 효율을 높였습니다.

프로젝트를 진행하는 내내 팀 내에서 즐거운 분위기를 유지하려고 노력하였으나, 프로젝트가 뜻대로 술술 진행되지 않으면서 서로의 탓을 할 때도 있었습니다. 저희 팀원들은 그럴때마다 저녁에 맛있는 것을 시켜먹으면서 대화를 통해 해결하려고 노력하였습니다. 각자 맡은 부분을 하다가 막히는 부분이 생길 때마다 서로 같이 해결하는 방법도 찾아주고, 선배들에게 적극적으로 물어보는 방법을 통해 주변에서도 많은 도움을 주어서 프로젝트를 마무리하는데 큰 도움이 되었습니다.

학교를 다니며 처음으로 진행하는 장기 프로젝트여서 미숙한 부분이 많았고 완성을 할 수 있을지에 대한 걱정이 컸었는데, 교수님과 매주 미팅을 통해 어렵고 부족한 부분들을 하나씩 채워나갔고, 수상부분에서는 아쉽게 순위권에 들어가지 못하였지만 저희 팀원들의 생각보다 훨씬 높은 목표치를 구현했고 구현 결과 또한 만족도 높게 끝내서 아주 뿌듯하게 한학기 마무리 할 수 있었습니다.