

# RFID를 이용한 회원제 코인세탁소 이용 서비스

B489067 정민우  
B489071 정진욱

- 주제 선정
- 사용 기술 또는 이론
- 프로젝트 구성
- 결론



## 코인세탁소

- 이용자가 세탁물을 세탁기에 직접 투입한 후 수납기에 정해진 금액만큼의 돈을 넣고 이용하는 방식
- 과거부터 있었지만 최근 1인 가구의 비중이 증가하고 거주 형태가 과거와 다르게 많이 바뀌어 대량 세탁과 관리의 필요가 없는 코인세탁소 이용량이 증가함
- 일반적인 이용방법으로 환전기에서 500원 주화를 교환한 뒤 세탁기, 건조기, 세제 구입기 등을 이용함
- 셀프 세탁소는 1900년도에도 존재했을 만큼 오래되었지만 크게 변한 부분 없이 그대로 운영되고 있음
- 코인세탁소를 자주 이용하는 사람들의 공통된 불만에서 주제를 선정하게 됨

## 느낀 불편한 점과 개선점

- 세탁기가 모두 이용중일 경우 세탁물을 들고 헹굼 하는 경우, 자신의 세탁물이 언제 끝날지 모르는 경우가 잦음
- + 세탁기의 이용상황을 앱 상에서 알려주어 혼잡할 때를 피하게 하고 진행상황을 알려줌
- 이용 시 마다 현금을 동전으로 바꿔야 하고 카드 결제를 한다고 해도 기기별로 따로 결제해야 함
- + NFC를 이용한 회원카드를 만들어 충전 후 찍는 것으로 간편하게 사용할 수 있게 함
- + 편리한 이용 방법으로 회원을 유치하여 유실물 관리나 회원관리가 간편해짐
- 세탁이 끝난 후 세탁물의 분실위험이나 고가의 기기가 이용자가 아닌 제3인에게 고장나는 경우가 있음
- + 회원이 아닌 경우에는 이용할 수 없게 만들어 근본적으로 방지

## 주제 선정 동기



- 소요시간을 확인 후 시간에 맞게 다시 찾아 오거나 세탁소 안에서 기다리던 가장 기본적인 방법을 보완
- 세탁소 내에 카페 등 휴게공간을 마련함



- 한국에서 가장 많은 점포를 가진 기업
- 무인 접수 시스템과 세탁 경험에서 나온 노하우를 이용한 장점 내세움



- 키오스크를 이용하여 환전의 번거로움을 줄이고 회원의 휴대전화 번호로 회원 서비스를 행함
- 고정된 키오스크로는 이용요금 지불만 가능하고 세탁기 확인 기능이 존재하지 않음

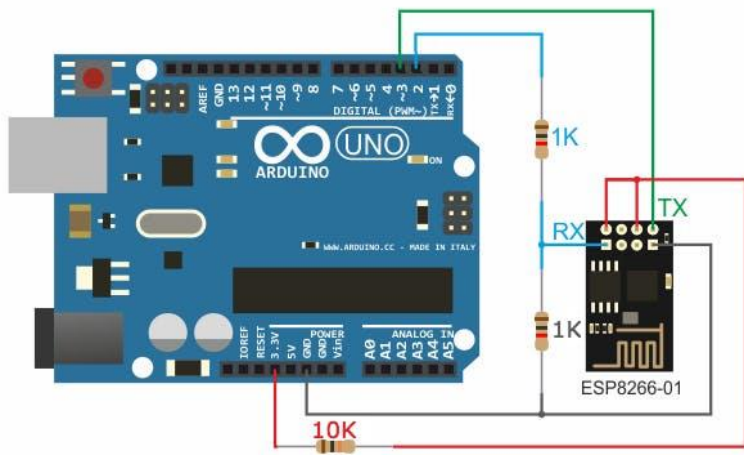
## 설계 목표

- 세탁기가 모두 이용중일 경우 세탁물을 들고 헹굼을 하는 경우, 자신의 세탁물이 언제 끝날지 모르는 경우가 잦음
  - + 세탁기의 이용상황을 웹으로 알려주어 혼잡할 때를 피하게 하고 진행상황을 알려줌
  - + 회원이 아닌 경우에는 이용할 수 없게 만들어 근본적으로 방지
- ↳
- 아두이노를 세탁기에 장착해 시간, 이용정보를 받아오게 하고 이더넷 실드를 이용하여 세탁소 와이파이와 연결함
  - 어플리케이션으로 이용자와 관리자가 언제 어디서든 세탁소 서비스에 접속할 수 있도록 구성
  - DB에는 기본정보와 연락처 등을 기록하여 이용회원을 관리하고 편리하게 사용할 수 있게 함

## 설계 목표

- 이용 시 마다 현금을 동전으로 바꿔야 하고 카드 결제를 한다고 해도 기기별로 따로 결제해야 함
  - + NFC을 이용한 회원카드를 만들어 충전 후 찍는 것으로 간편하게 사용할 수 있게 함
- ↳
- 비접촉으로 태그의 정보 값을 가져와 이용하는 RFID를 이용함  
세탁기에 달린 아두이노에서 모듈을 이용하여 회원임을 확인하고 작동
  - 지불방식을 현금지불에서 토큰을 이용한 본인 확인 후 충전된 금액을  
사용하도록 바꿈

- 아두이노와 서버 구성



NODE MCU

세탁기의 정보를 WIFI를 통해 전달하고 RFID 모듈 이용

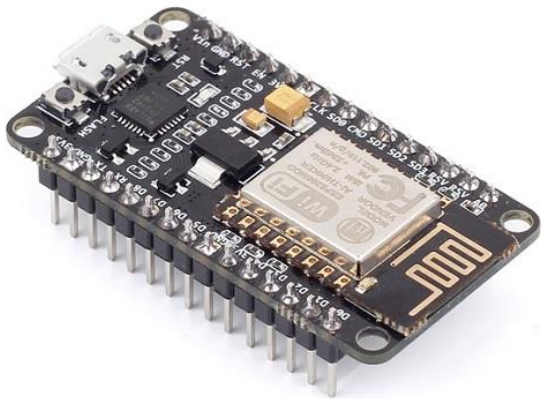


Firebase DB

회원정보와 세탁소 이용현황 등의 데이터



- 추가 모듈과 기술



### 이더넷 쉴드

아두이노에서 처리한  
정보들을 와이파이로 전송



### Firebase

### Firebase 데이터 베이스

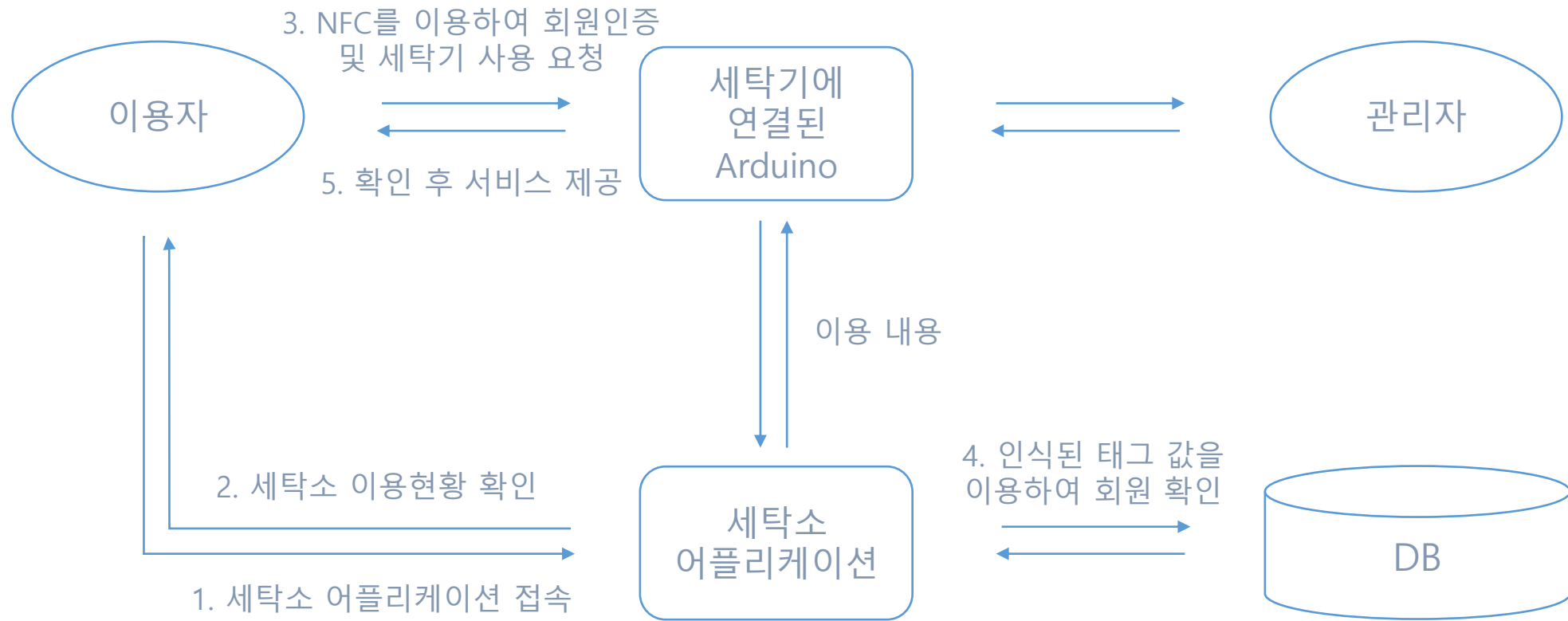
여러 플랫폼에서  
실시간 데이터베이스 구현



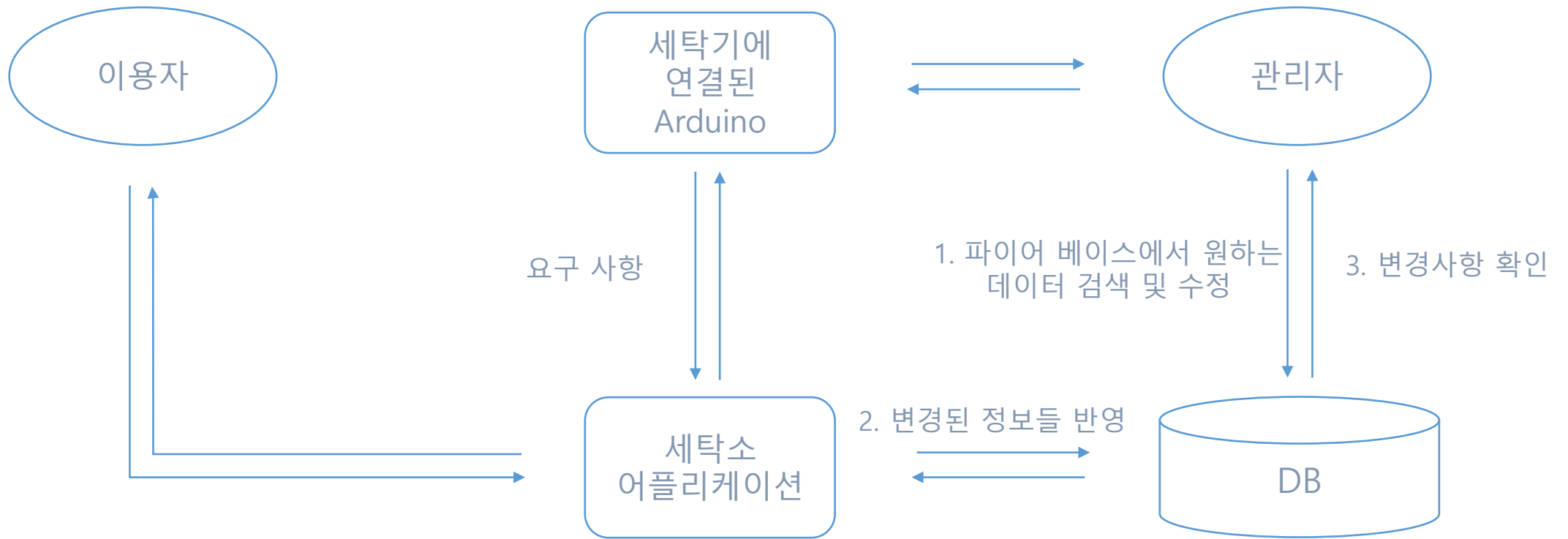
### RFID

비접촉방식으로 원거리 태그의  
정보를 읽음

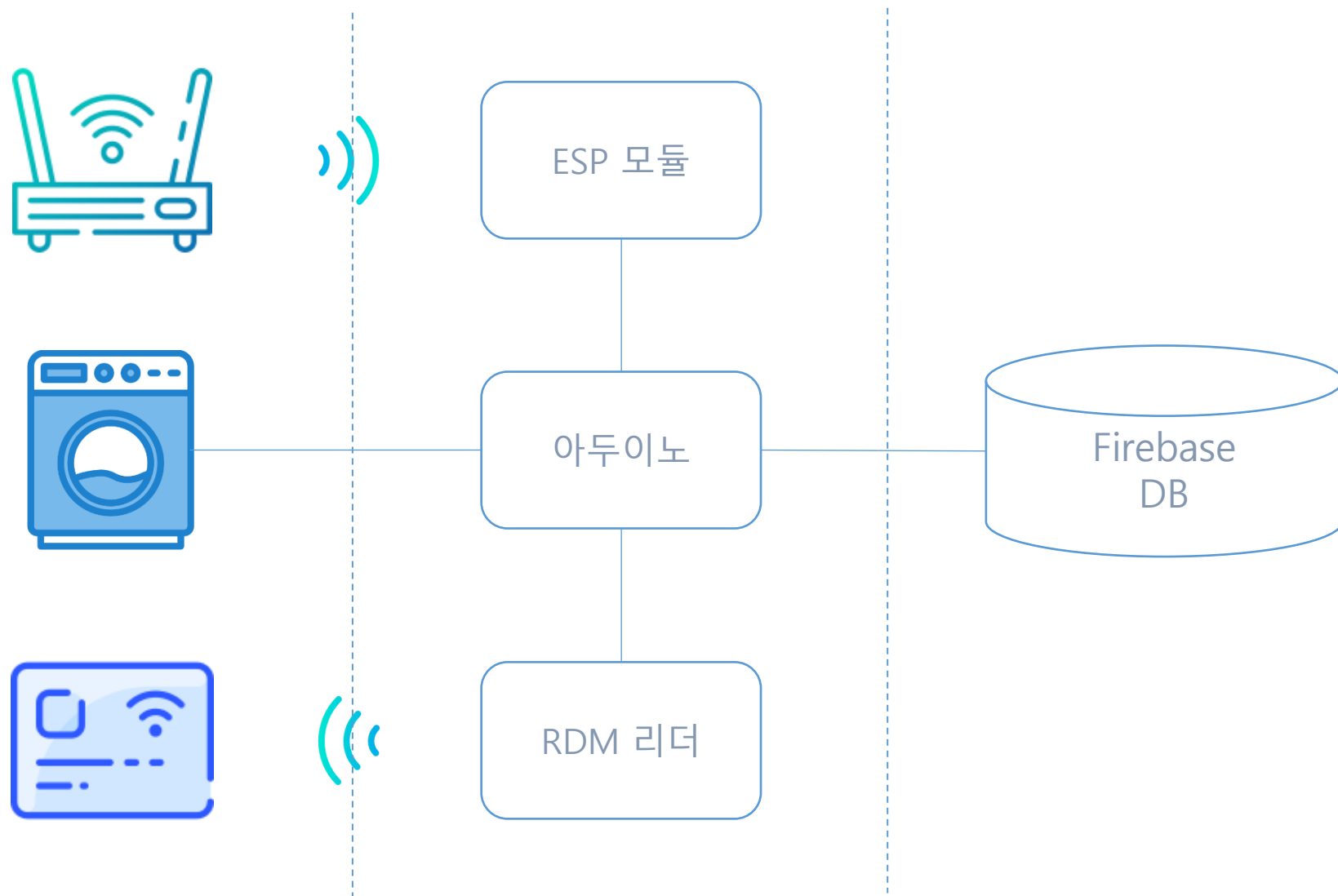
## 프로젝트 구성



## 프로젝트 구성



## 프로젝트 구성



## 세탁소 어플리케이션 화면

### 로그인

Email

---

password

---

LOG IN

RF\_TAG

회원 가입하기

### 회원 가입

이름

김몽몽

---

phone

010-1234-5678

---

Email

m@gmail.com

---

password 6자리이상

.....

passwordcheck 6자리이상

.....

REGISTER

### 회원 가입

이름

김몽몽

---

phone

010-1234-5678

---

Email

m@gmail.com

---

password 6자리이상

.....

passwordcheck 6자리이상

.....

회원가입에 성공하였습니다 태그를 등록해주세요

REGISTER

### 코인 세탁소

j@gamil.com님 23:3:16 이전

세탁기1 23:32:56  30 set	세탁기2  0 set	세탁기3 0:17:10  0 set
세탁기4  0 set	세탁기5  0 set	

## 아두이노에서 RFID 리딩과 가입

```
bool readTags()
{
    ok = -1;
    int tag_no;
    if(Read_state == false)
    {
        if (RFID.available() > 0)
        {
            delay(100);
            for (int i = 0 ; i < 14 ; i++)
            {
                datal = RFID.read();
                temptag[i] = newtag[i] = datal;

                Serial.print(newtag[i]);
                if(i<13)
                    Serial.print(",");
            }
        }
    }
}
```

- RFID 태그 값을 읽어와 newtag 배열에 저장
- 읽어온 태그 값을 출력하고 부족한 부분은  
    쉼표로 비워둠

```
if(MODE_GET == "REGISTER"){
    while(DB_GET_CNT++ < TOTAL_TAG)
    {
        Email_GET = Firebase.getString("Member/User"+String(DB_GET_CNT)+"/email");

        Serial.print("Email_GET:");Serial.println(Email_GET);
        if(Email_GET == "")
        {
            Serial.print("USER"); Serial.print(DB_GET_CNT);Serial.println(" EMPTY");

            Firebase.setString("TAG_INFO/USER_ORDER",String(DB_GET_CNT));
            Firebase.setString("TAG_INFO/TAG_READ",comfirmedTag);
            DB_GET_CNT_STORED = DB_GET_CNT;
            break;
        }
    }
}
```

- 파이어베이스의 이메일 값을 스캔하여 해당 유저의  
    등록 여부를 확인
- 해당 메일의 자리가 빈 값이면 메일 값을 받아오고  
    파이어베이스에 저장함

## 아두이노에서 RFID 태그로 회원인증

```
else if(MODE_GET == "RF_LOG")
{
    while(DB_GET_CNT++ < TOTAL_TAG)
    {
        TAG_GET = Firebase.getString("Member/User"+String(DB_GET_CNT)+"/tag");

        if(TAG_GET != "")
        {
            if(TAG_GET == confirmedTag)
            {
                String ID_GET = Firebase.getString("Member/User"+String(DB_GET_CNT)+"/email");

                Firebase.setString("LAUNDRY_INFO/ACCOUNT", ID_GET);
                Firebase.setString("TAG_INFO/MODE", "");

                Serial.print("ID_GET:"); Serial.println(ID_GET);
                Serial.print("DB_GET_CNT:"); Serial.println(DB_GET_CNT);
                Serial.print("confirmedTag:"); Serial.println(confirmedTag);

                break;
            }
        }
    }
}
```

- 파이어베이스에 유저 별 tag 값을 리딩
- Tag의 값이 존재하고 DB에 등록된 태그 값과 현재 읽은 태그 값이 일치할 시 해당 ID값을 저장
- 파이어베이스의 Account에 메일 값을 전송하고 받아온 정보들을 출력

## 어플리케이션에서 태그를 이용하여 로그인

```
loginTag.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) {
        Toast.makeText(MainActivity.this, "등록된 태그를 터치해주세요", Toast.LENGTH_LONG).show();
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference("TAG_INFO/MODE");
        myRef.setValue("RF_LOG");
    }
});
```

- TAG 로그인 버튼 클릭 시 회원등록을 위한 태그 접촉 메시지를 출력
- 파이어베이스 데이터베이스에서 이용하기 위한 인스턴스와 레퍼런스 생성
- 데이터베이스에서 이용할 MODE경로에 RF\_LOG 문자 저장



## 어플리케이션에서 회원가입

```
mResigettxt.setOnClickListener(new View.OnClickListener(){  
    @Override  
    public void onClick(View v) {  
        FirebaseDatabase database = FirebaseDatabase.getInstance();  
        DatabaseReference reference1 = database.getReference("TAG_INFO/");  
        reference1.child("MODE").setValue("REGISTER");  
  
        startActivity(new Intent(MainActivity.this, RegisterActivity.class));  
    }  
});
```

- 회원가입 버튼을 누른 후 파이어베이스에서 이용할 인스턴스와 레퍼런스 생성
- Intent 함수를 통해 register 액티비티 함수를 호출
- 인텐트를 실행 시 가입을 하기 위한 RegisterActivity로 이동

## 어플리케이션에서 세탁기 이용요청

```
setBtn1.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) {

        minSet1 = minEdt1.getText().toString().trim();
        tempMin = Integer.parseInt(minSet1) + threadMin;

        Hr1 = threadHr+(tempMin / 60);
        Min1 = tempMin % 60;
        Sec1 = threadSec;
        Hr1 = (Hr1>23?0:Hr1);

        DatabaseReference reference = database.getReference("LAUNDRY_INFO");
        reference.child("MC1").setValue(Hr1+":"+Min1+":"+Sec1);
    }
});
```

- 세탁기와 금액을 선택하여 세탁기를 이용
- 원하는 이용시간을 텍스트로 입력 받아 저장하고 세팅한 시간 값과 현재 값을 더해 저장
- 세탁기 이용정보를 보이기 위한 경로에 대한 참조를 생성 후 MC1 자식에 시간 값 저장

# 시연 영상

## 결론

- 주제를 선정할 때 이용 중 느낀 불편한 서비스 개선에 중심을 두고 선택하여 기술적인 측면에서 부족한 점이 많음
- 현재 서비스하고 있는 세탁소들이 행하지 않고 있거나 부분적으로 구현된 기술들을 이용하여 차별성을 가짐
- 상용화 시 이용자들이 가장 큰 변화를 느낄 수 있는 시간정보와 지불방법이 개선되어 접근성과 편의성이 증가됨

## 추가적인 사항

- 이용시간 예약이나 진행사항을 푸시알림으로 알려주거나 온라인에서 요금을 다양한 방법으로 결제할 수 있도록 앱 기능 추가
- 세탁기 이용 페이지에서 더 직관적으로 이용할 수 있도록 UI를 개선해야 함
- 무인 세탁소 특성 상 기기의 결함이나 고객의 클레임에 대한 대응이 늦기 때문에 오류코드를 통한 빠른 피드백과 관리자와의 대화 등의 기능이 필요

이상입니다