HUFSmartKey

(블루투스 기반 스마트 열쇠)

2020. 06. 25

한국외국어대학교 정보통신공학과

인터넷 응용 4팀 HUFSmartKey

201802719 이유진, 201602560 이재성, 201703812 홍영빈, 201803282 정순인

제출일	2020, 06, 25	전공	정보통신공학과
과목	인터넷 응용	학번	201802719 201602560 201703812 201803282
담당교수	홍진표 교수님	이름	이유진 (팀장) 이재성 홍영빈 정순인

머리말

본 보고서는 Retrofit2, Django, 아두이노(Arduino), 라즈베리 파이(Raspberry Pi)등을 이용하여 제작한 HUFSmartKey라는 어플에 대해 소개한다. HUFSmartKey는 여러가지 장소에서 사용할 수 있는 열쇠 역할을 하는 어플인데 비콘(Beacon)의 블루투스 기능을 이용하여 구현하였다. 이 어플 하나로 공공 화장실, 상가, 현관문, 회사 등의 출입을 허가하는 열쇠 역할을 하는 것이다.

목차

1. 개요

본 장에서는 이 어플의 제작 목적, 이 앱을 만드는 데에 이용한 것들에 대한 설명, 개발 배경에 대해 기술하였고, 제품(앱)에 대한 소개와 이 제품을 사용하였을 때의 기대효과, 그리고 이 제품(앱)의 소스코드 등을 제공한다.

1.1. 개발 배경

기술이 발전하고 시대가 변함에 따라 사람들은 편리성을 추구하게 되었다. 현관문을 열 때, 또는 특정 용무를 보기 위한 장소에 진입하기 위해 우리는 일반적으로 비밀번호 혹은 카드 키 등을 통해 그 곳에 입장할 권한을 얻어서 들어가게 된다. 왜 어떤 곳은 비밀번호를 통해 들어가고, 또 어떤 곳은 카드 키를 통해 들어가는 것일까? 모든 장소를 카드 키를 통해 들어갈 수 있으면 더 편리하지 않을까? 라는 생각을 하게 되었고, 카드 키를 통해 출입 권한을 얻는 것이라면 하나의 어플이 여러 장소의 출입 권한을 가질 수는 없을까? 라는 생각을 하게 되었다. 우선, 지금까지는 비밀번호를 이용하여 출입할 수 있었던 장소를 카드 키를 통해 들어갈 수 있다는 것에서 우리는 편리성을 얻게 되었고, 여러 장소의 출입 권한을 갖기 위해서는 각 장소에 해당하는 키 또는 비밀번호를 갖고 있어야 했다면 이제는 하나의 어플로 여러 장소의 출입 권한을 가질 수 있기 때문에 휴대성도 좋아지게 되었다는 데에 의의가 있다.

1.2 용어 및 약어

용어 및 약어	풀이	비고
Retrofit2	REST API에 특화된 CRUD 방식의 서버연결을 편하게 구현할 수 있는 라이브러리이며, annotation 방식으로 구현이 쉽고, 가독성, 유지보수, 재사용성이 좋다는 장점이 있다. Retrofit은 크게 4가지 부분으로 나뉜다. 1. 네트워크 통신에 필요한 전반적인 설정을 관리하는 Retrofit 몸통 부분 2. 통신할 API의 Http Method를 정의하는 Service Interface 3. Request / Response 4. DTO (Data Transfer Object)	
Django rest API	Rest API는 Representational State Transfer라는 용어의 약자로 그대로 해석하면 대표적 상태 전송이라는 뜻이다. Rest API를 통해 Rest 서버는 API를 제공하고, 클라이언트는 사용자 인증이나 세션, 로그인정보 등을 직접 관리하는 구조로 각각의 역할이 확실히 구분되기 때문에 클라이언트와 서버에서 개발해야 할 내용이 명확해지고 서로간 의존성이 줄어들게 된다. REST API로 설계하기 위해서는 URI는 정보의 자원을 표현해야 하고 자원에 대한 행위는 HTTP 0Method (GET, POST, PUT, DELETE)로 표현한다. Django에서 프로젝트를 진행하게 되면 views에서 하나의 템플릿에 하나의 클래스 혹은 함수가 할당되어 실행되기 때문에 코드의 재사용성이 떨어지게 되는데 Django REST framework를 사용하게 되면 이러한 부분을 방지할 수 있다는 장점이 있다.	
Arduino	아두이노(Arduino)는 오픈소스를 기반으로 한 단일 보드 마이크로 컨트롤러이다. 아두이노는 다수의 스위치나 센서로부터 값을 받아들여, LED나 모터와 같은 외부 전자 장치들을 통제함으로써 환경과 상호작용이 가능한 물건을 만들어낼 수 있다. 또한 Adobe Flash, 프로세싱, Max/MSP와 같은 소프트웨어를 연동할 수 있다. 일반적으로 AVR프로그래밍이 WinAVR로 컴파일하여, ISP장치를 통해 업로드를 해야하는 등 번거로운 과정을 거쳐야 하는 데 비해, 아두이노는 컴파일된 펌웨어를 USB를 통해 업로드를 쉽게 할 수 있다. 또한 아두이노는 다른 모듈에 비해 비교적 저렴하고, 윈도우를 비롯해 맥 OSX, 리눅스와 같은 여러 OS를 모두 지원한다. 아두이노 보드의 회로도가 CCL에 따라 공개되어 있으므로, 누구나 직접 보드를 만들고 수정할 수 있다.	

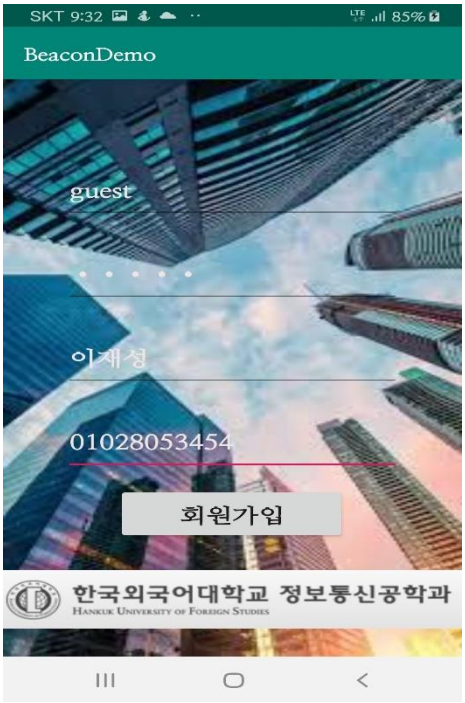
Beacon	비콘(Beacon)은 근거리에 있는 스마트 기기를 자동으로 인식하여 필요한 데이터를 전송할 수 있는 무선 통신 장치이다. 근거리 무선 통신인 NFC가 10츠 이내의 근거리에서만 작동하는 반면, 비콘은 최대 50m 거리에서 작동할 수 있다. 비콘 기술을 이용하면 쇼핑센터, 음식점, 박물관, 미술관, 영화관, 야구장 등을 방문한 고객의 스마트폰에 할인 쿠폰이나 상세 설명 등의 데이터를 전송할 수 있다.	
SQLite	SQLite는 별도의 서버 프로세스가 필요 없고 SQL 질의 언어의 비표준 변형을 사용하여 데이터베이스에 액세스할 수 있는 경량 디스크 기반 데이터베이스를 제공하는 C 라이브러리입니다. 일부 응용 프로그램은 내부 데이터 저장을 위해 SQLite를 사용할 수 있습니다. SQLite를 사용하여 응용 프로그램을 프로토타입한 다음 PostgreSQL 이나 Oracle과 같은 더 큰 데이터베이스로 코드를 이식할 수도 있습니다. SQLite는 킬로바이트의 데이터를 저장하든 수 기가바이트의 데이터를 저장하든 빠른 속도와 이식성, 안정성을 제공하도록 설계됐다. SQLite의 가장 큰 장점 중 하나는 거의 어디서나 실행 가능하다는 데 있다. SQLite는 윈도우, 맥OS, 리눅스, iOS, 안드로이드를 비롯한 다양한 플랫폼에 이식됐다. 특히 윈도우 사용자는 일반적인 Win32, UWP, WinRT, 닷넷용으로 사전 컴파일된 바이너리를 사용할 수 있다. 앱의 배포 대상이 무엇이든 그 중 대부분은 그 대상에 맞는 SQLite가 있거나 C 소스 코드를 해당 타겟으로 이식할 방법이 존재한다고 생각하면 된다.	

2. 제품(앱) 소개

2.1 사용법

2.1.1. Guest 권한으로 출입 시

(1) 회원가입

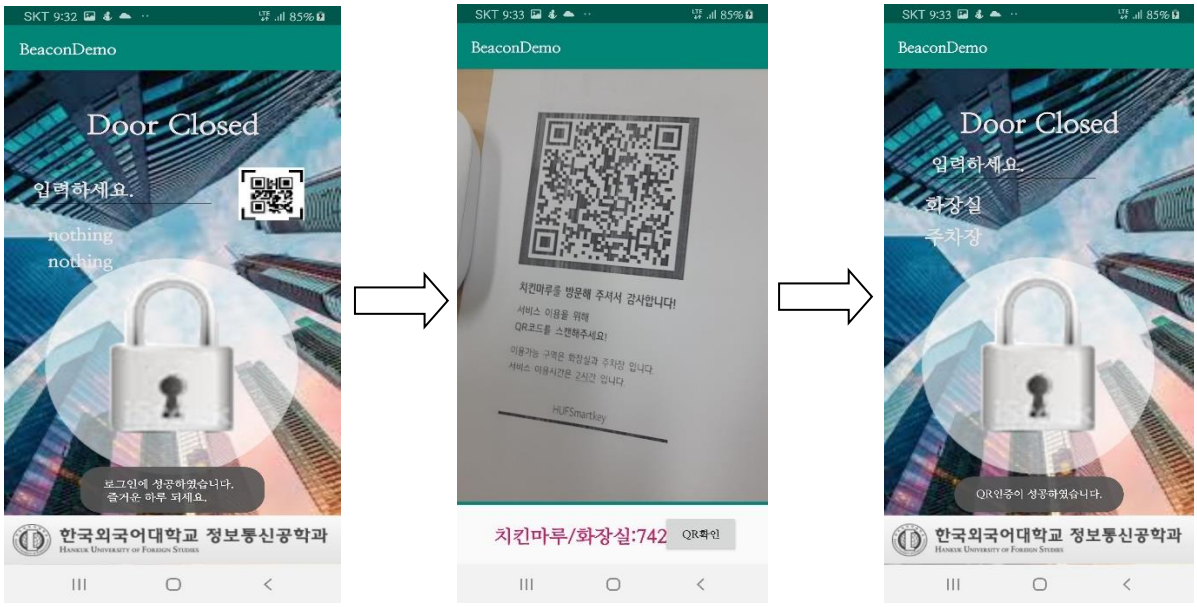


어플리케이션을 실행하게 되면 첫 화면에서 회원가입과 로그인 버튼이 있는데, HUFSmartKey를 이용하기 위해서는 우선 회원가입을 통해 사용자 등록을 해야 한다. 위 사진은 Guest 권한으로 출입을 하기 위해 회원가입을 하는 화면이고, 회원가입을 완료하면 완료 메시지와 함께 가입한 ID가 표시된다.

(2) 로그인

회원가입을 완료한 후 가입된 ID와 Password로 로그인을 한다.

(3) 잠금, QR코드 스캐너

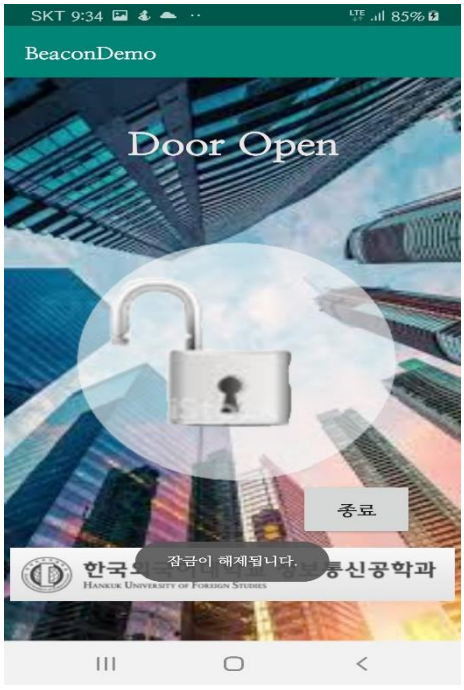


Guest의 경우 출입 가능 구역이 아직 없으므로 QR코드 스캔을 통해 출입 가능 구역을 획득한다. QR코드 스캐너를 통해 출입가능 구역을 확인하고 잠금 화면에서는 nothing으로 표시되었던 것이 화장실, 주차장으로 표시된다. 원하는 출입구역을 입력하면 자물쇠 버튼을 누를 수 있다.

(4) 서비스 시간 만료

QR코드를 스캔하고 2시간 이내에 자물쇠 버튼을 눌러야만 출입이 가능하다. 위 화면은 2시간이 지난 상태에서 자물쇠 버튼을 누른 경우이다.

(5) 잠금 해제



2.1.3에서 자물쇠 버튼을 누르게 되면 해당 출입구역의 Beacon을 스캔하고, 스캔이 완료되면 자물쇠 잠금이 해제된 화면이 나오고 LED가 켜지면서 출입문을 열 수 있다.

2.1.2 사업자 (소상공인) 권한으로 출입 시

(1) 회원가입

2.1.1 (1)과 동일한 방식이다.

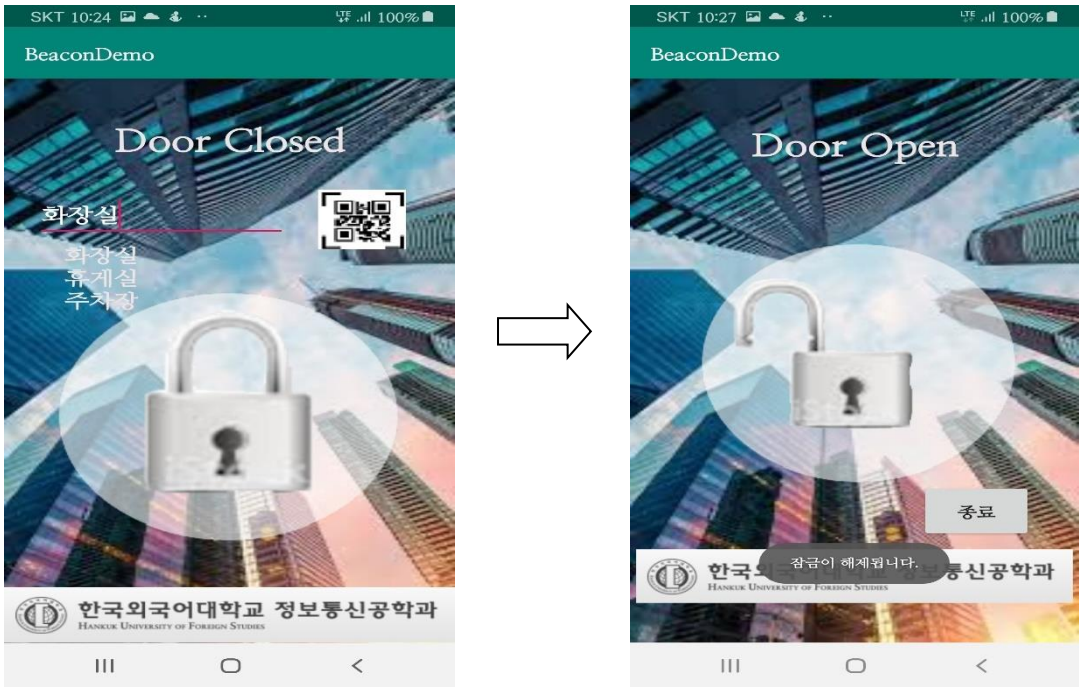
(2) 로그인

2.1.1 (2)과 동일한 방식이다.

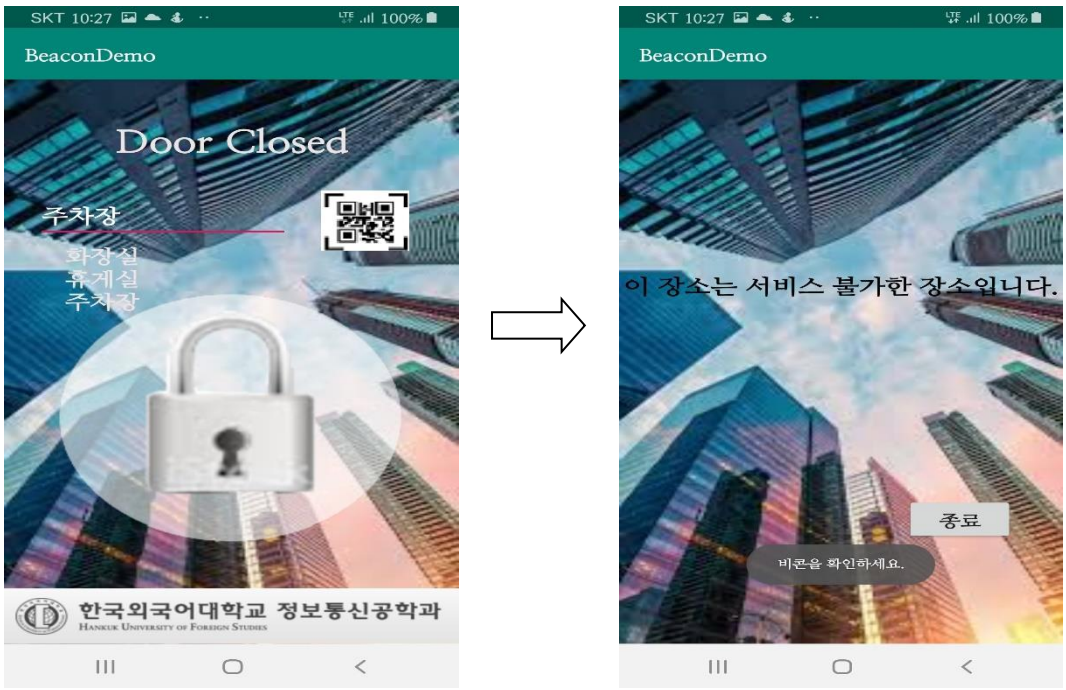
(3) 잠금

로그인을 하게 되면 Guest와 동일한 화면이 나오지만 사업자의 경우 미리 출입 가능 구역이 표시되어 있다. 그러므로, 사업자는 QR코드를 스캔할 필요가 없다.

2.2.4. 출입 구역 입력



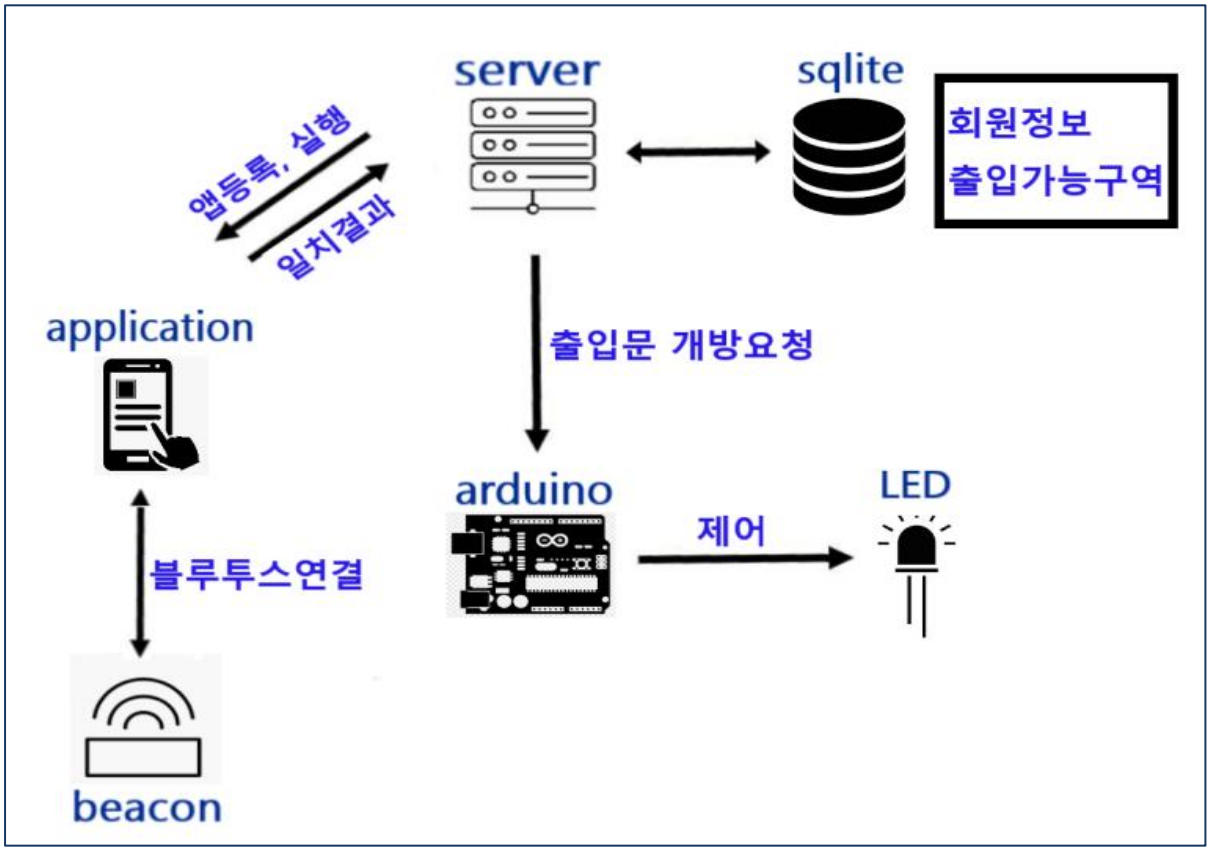
원하는 출입구역을 입력하고 자물쇠 버튼을 누르면 출입문이 열리는 것은 사용자가 Guest일 경우와 동일하다.



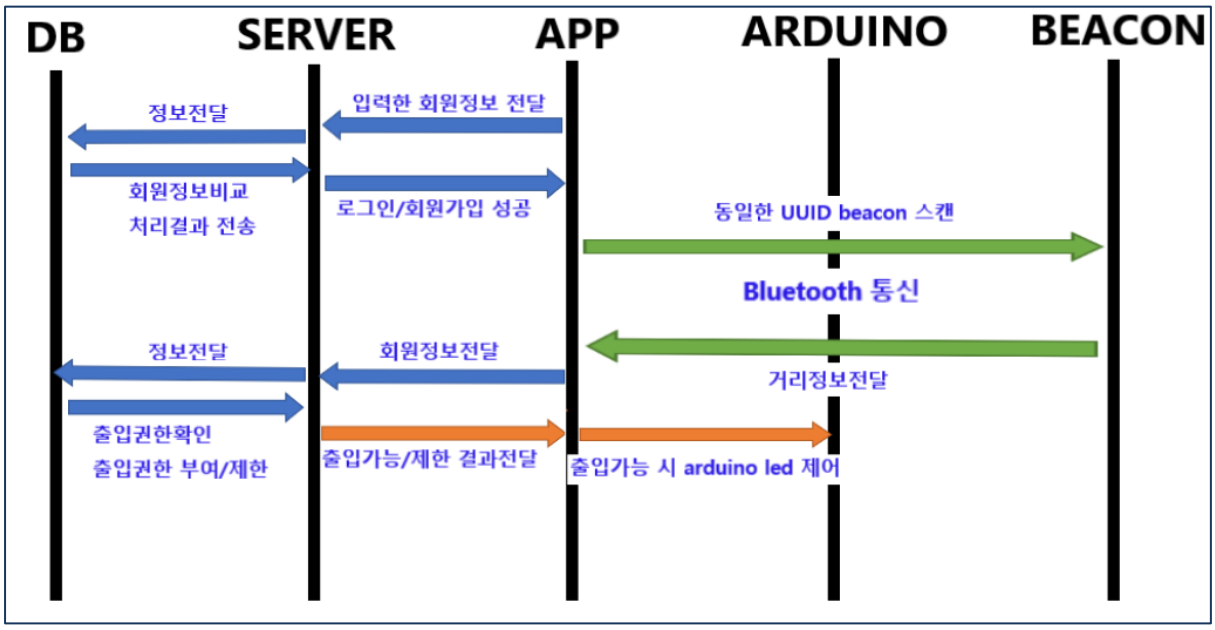
예를 들어, 현재 접근할 수 있는 구역은 화장실인데 주차장을 입력하게 되면 “비콘을 확인하세요”라는 메시지와 함께 출입이 제한된다.

2.2 구성도

2.2.1 제품 구성도



2.2.2 제품 흐름도



2.3 주요 적용 기술

S/W	1. SQLite 3 데이터베이스 구축
	2. Android app과 Beacon의 Bluetooth 통신
	3. Retrofit2 (Android app)와 Django Rest API (Server)를 이용한 HTTP통신
H/W	1. Arduino Kit의 LED를 이용한 출입문 개폐 확인
	2. Beacon을 이용한 출입 권한 제한

2.4 제품 개발 환경

구분		상세 내용
S/W 개발 환경	OS	Linux, Android
	개발 환경(IDE)	Android Studio, SQLite 3
	개발 언어	Python, Kotlin
H/W 구성 장비	디바이스	Arduino Uno, Beacon
	Actuator	LED
	통신	Serial, Bluetooth
	개발 언어	C

3. 기대효과 및 사업성

3.1 기대효과

기존에는 비밀번호로 입력해왔던 장소에도 어플로 권한만 얻으면 입장이 가능하기 때문에 이전에 비해 편리성이 증가하였고, 비밀번호만 알면 누구나 들어갈 수 있었던 이전과 달리 권한을 얻은 사람만 출입할 수 있기 때문에 보안성이 강화되었다. 또한 손을 쓸 수 없는 상황이나 위급한 상황 등에서 어플리케이션을 실행시켜 놓은 상태로 휴대폰만 소지하게 되면 출입문이 개방되기 때문에 예기치 못한 상황을 대비할 수 있다.

게다가 하나의 어플로 다양한 장소의 입장 권한을 얻을 수 있기 때문에 여러가지 열쇠 혹은 카드 키 등을 들고 다녔던 이전에 비해 휴대폰만 잘 소지하고 있다면 분실의 상황을 방지할 수 있다.

3.2 사업성

여러 장소의 열쇠 혹은 카드 키 등을 가지고 다닐 때에 비해 HUFSmartKey 어플리케이션만 있으면 되기 때문에 열쇠나 카드 키를 제작하는 비용을 절약할 수 있고, 분실을 하였을 때 추가 비용이 발생하는 것을 방지하여 줄 수 있다.

4. 참고 사이트

Title	URL
Android app 개발 (Kotlin)	https://developer.android.com/kotlin/
Arduino 보드 개발	https://www.arduino.cc/reference/en/#functions
Web Page 개발	https://github.com/StartBootstrap/startbootstrap-creative
개발 환경 및 네트워크 구축 (Port Forwarding)	https://yangarch.tistory.com/108
Arduino Serial 통신	https://pythonhosted.org/pyserial/pyserial_api.html
Django Rest API	https://velog.io/@yvvyoon/django-rest-framework-1
SQLite 3 데이터베이스 구축	https://docs.python.org/ko/3/library/sqlite3.html
Retrofit2 통신	https://square.github.io/retrofit/
Beacon Scanner 개발	https://altongmon.tistory.com/449
QR Scanner 개발	https://park-duck.tistory.com/109

5. 프로젝트 세부 추진 계획 및 세부 일정

프로젝트 기간		2020.06.03 ~ 2020.06.24			
구분	추진내용	프로젝트 기간			
		1주	2주	3주	4주
계획	아이디어회의, 역할분담				
분석	구체적인 자료조사				
개발	개발환경 구축				
	Android UI 구현 및 어플리케이션 개발				
	Django Rest API, db 구축				
	Rest API & Retrofit을 이용한 http 통신				
	Beacon scanner 프로그래밍				
	Arduino 제어 프로그래밍				
	테스트, 최종 보고서 작성				

6. 역할 분담

이름	학번	담당 역할
이유진	201802719	Django Rest API, SQLite 3 DB 구축, Beacon Scanner 개발
이재성	201602560	Android app 개발, Retrofit 통신, Beacon Scanner 개발 QR Scanner 개발
홍영빈	201703812	리눅스 및 원격 개발환경 구축, Arduino serial 통신, 홈페이지 제작
정순인	201803282	Android app UI 구현 및 개발

7. 소스 코드

Android Code – Retrofit

```
package com.zartre.app.beacondemo

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_login.*
import retrofit2.*
import retrofit2.converter.gson.GsonConverterFactory

class login : AppCompatActivity() {
    private val ALLOWED = "allowed_area"
    private val USERINFO = "UserID"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        // retrofit 사용
        var retrofit = Retrofit.Builder()
            .baseUrl("http://220.67.124.145:8080")
            .addConverterFactory(GsonConverterFactory.create())
            .build()

        var loginservice: forLoginService = retrofit.create(forLoginService::class.java)
// retrofit 객체 생성

        login_btn.setOnClickListener{
            var s_id = edit_ID.text.toString()
            var s_pw = edit_PW.text.toString()

            loginservice.requestLogin(s_id, s_pw).enqueue(object: Callback<forLogin> {
                override fun onFailure(call: retrofit2.Call<forLogin>, t: Throwable) {  //
실패할 때
```

```
        override fun onResponse(call: retrofit2.Call<forLogin>, response: Response<forLogin>)
{  // 서버에서 정상 응답이 올 때
                            if (response?.isSuccessful) {
                                var forLogin = response.body()
                                Log.d("LOGIN", "msg : " +  forLogin?.msg)
                                Log.d("LOGIN", "code : " +  forLogin?.code)
                                Log.d("LOGIN", "allowed_area : " +  forLogin?.allowed_area)
                                Toast.makeText(this@login, "로그인에 성공하였습니다. ₩n 즐거
운 하루 되세요.", Toast.LENGTH_SHORT).show()

                                var intent = Intent(applicationContext,
lockActivity::class.java).apply {
                                    putExtra(ALLOWED, forLogin?.allowed_area)
                                }
                                startActivity(intent)
                            }
                            else {
                                Toast.makeText(this@login, "로그인에 실패했습니다. ₩n 아이디
혹은 비밀번호를 확인하세요.", Toast.LENGTH_LONG).show()
                            }
                        }
                    })


                }
                signUp_btn.setOnClickListener {
                    var intent = Intent(applicationContext, signup::class.java)
                    startActivity(intent)
                }
            }
        }
```

Android Code – QR Scanner

```
package com.zartre.app.beacondemo

import android.Manifest
import android.content.Intent
import android.content.pm.PackageManager
import android.os.Bundle
import android.util.Log
import android.util.SparseArray
import android.view.SurfaceHolder
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.core.util.isNotEmpty
import com.google.android.gms.vision.CameraSource
import com.google.android.gms.vision.Detector
import com.google.android.gms.vision.barcode.Barcode
import com.google.android.gms.vision.barcode.BarcodeDetector
import kotlinx.android.synthetic.main.activity_qr_check.*
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import java.lang.Exception

class qr_check : AppCompatActivity() {
private val requestCodeCameraPermission = 1001
private lateinit var cameraSource: CameraSource
private lateinit var detector: BarcodeDetector

private val USERINFO = "UserID"
private val ID = "id"
private val ALLOWEDAREA = "Allowed area after QR"
```

```
        if(ContextCompat.checkSelfPermission(
                        this@qr_check,
                        Manifest.permission.CAMERA)                                    !=
PackageManager.PERMISSION_GRANTED
                ) {
                    askForCameraPermission()
                }
                else {
                    setupControls()
                }

                qr_check_btn.setOnClickListener {
                    var intent = Intent(applicationContext, lockActivity::class.java)
                    startActivity(intent)
                }

            }

            private fun setupControls() {
                detector = BarcodeDetector.Builder(this@qr_check).build()
                cameraSource = CameraSource.Builder(this@qr_check, detector)
                    .setAutoFocusEnabled(true)
                    .build()
                cameraSurfaceView.holder.addCallback(surfaceCallBack)
                detector.setProcessor(processor)
            }

            private fun askForCameraPermission() {
                ActivityCompat.requestPermissions(
                    this@qr_check,
                    arrayOf(Manifest.permission.CAMERA),
                    requestCodeCameraPermission
                )
            }
```

```
        override fun onRequestPermissionsResult(
                requestCode: Int,
                permissions: Array<out String>,
                grantResults: IntArray
            ) {
                super.onRequestPermissionsResult(requestCode, permissions, grantResults)
                if(requestCode == requestCodeCameraPermission &&
grantResults.isNotEmpty()) {
                    if(grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                        setupControls()
                    }
                    else {
                        Toast.makeText(applicationContext, "Permission Denied",
Toast.LENGTH_SHORT).show()
                    }
                }
            }

            private val surfaceCallBack = object : SurfaceHolder.Callback {
                override fun surfaceChanged(p0: SurfaceHolder?, p1: Int, p2: Int, p3: Int) {

                }

                override fun surfaceDestroyed(p0: SurfaceHolder?) {
                    cameraSource.stop()
                }
        override fun surfaceCreated(surfaceHolder: SurfaceHolder) {
                    try {
                        cameraSource.start(surfaceHolder)
                    } catch (exception: Exception) {
        Toast.makeText(applicationContext, "Something went wrong",
Toast.LENGTH_SHORT).show()
                    }
                }
            }

        private val processor = object : Detector.Processor<Barcode> {
```

```
        override fun receiveDetections(detections: Detector.Detections<Barcode>?) {
                    if(detections != null && detections.detectedItems.isNotEmpty()) {
                        val qrCodes: SparseArray<Barcode> = detections.detectedItems
                        val code = qrCodes.valueAt(0)

                        textScanResult.text = code.displayValue //= 치킨마루/화장실:74278bda-
b644-4520-8f0c-720eaf059935,주차장:1

                        var retrofit = Retrofit.Builder()
                            .baseUrl("http://220.67.124.145:8080")
                            .addConverterFactory(GsonConverterFactory.create())
                            .build()

                        var qrservice: forQRService = retrofit.create(forQRService::class.java)

                        var data = textScanResult.text.toString()
                        var output = data.split("/")
                        var store = output[0]
                        var area = output[1]

                        qrservice.requestQRScan(store, area).enqueue(object: Callback<forQR>
{
                            override fun onFailure(call: Call<forQR>, t: Throwable) {
                                var dialog = AlertDialog.Builder(this@qr_check)
                                dialog.setTitle("ERROR")
                                dialog.setMessage("서버와의 통신이 실패하였습니다.")
                                dialog.show()
                            }
        override fun onResponse(call: Call<forQR>, response: Response<forQR>){
                                if (response.isSuccessful) {
                                    var forQR = response.body()
                                    Log.d("QRCHECK", "msg(random guest id) : " +
forQR?.msg)
                                    Log.d("QRCHECK", "code : " + forQR?.code)
                                    Toast.makeText(this@qr_check, "QR인증이 성공하였습니
다.", Toast.LENGTH_SHORT).show()
```

```
        else {
                                        Toast.makeText(this@qr_check,  "QR인증이  실패하였습니
다.", Toast.LENGTH_SHORT).show()
                                    }
                                }
                            })
                        }
                        else {
                            textScanResult.text = ""
                        }
                    }
                }


        }
```

Android Code – Beacon Scanner

```
package com.zartre.app.beacondemo

import android.Manifest
import android.app.Dialog
import android.content.Context
import android.content.Intent
import android.content.pm.PackageManager
import android.graphics.Color
import android.graphics.drawable.ColorDrawable
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.RemoteException
import android.util.Log
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import kotlinx.android.synthetic.main.activity_lock.*
import kotlinx.android.synthetic.main.activity_scanner.*
import org.altbeacon.beacon.*
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import java.lang.IllegalArgumentException
import java.lang.reflect.Constructor
import kotlin.coroutines.coroutineContext

class ScannerActivity : AppCompatActivity(), BeaconConsumer {

    private val PERMISSION_REQUEST_COARSE_LOCATION = 1
    private val SCAN_PERIOD: Long = 150
    private val ENTER_AREA = 0.5
    private val LOG_TAG = "distance"
    private val EXTRA_BEACON = "beacon_uuid"
    private val ALLOWED = "allowed_area"
```

```
    private lateinit var beaconManager: BeaconManager
        private var beaconUuid: String = "74278BDA-B644-4520-8F0C-720EAF059935" //
default uuid to scan
        private var id: String = "id"
        private val ID = "id"
        override fun onCreate(savedInstanceState: Bundle?) {
            super.onCreate(savedInstanceState)
    //          setContentView(R.layout.activity_scanner)
            val uuid = intent.getStringExtra(EXTRA_BEACON)
            if (!uuid.isNullOrEmpty()) {
                beaconUuid = uuid // replace uuid with user input
            }
            val s_id = intent.getStringExtra(ID)
            id = s_id
    //          text_uuid.text = beaconUuid
            // Android M Permission check
            requestPerm()
            initBeaconManager()
    //          text_distance.text = "scanning"

    //          test_btn.setOnClickListener {
    //              var intent = Intent(applicationContext, unlock::class.java)
    //              startActivity(intent)
    //          }
    //          var intent = Intent(applicationContext, unlock::class.java)
    //          startActivity(intent)
        }

        private fun initBeaconManager() {
            beaconManager = BeaconManager.getInstanceForApplication(this)
            beaconManager.beaconParsers
                .add(
                    BeaconParser()
                        .setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24") //
iBeacon
                )
```

```
        else {
                        Log.d("MSG", "!!!!!!!!NO BEACON!!!!!!")
        //                  onPause()
                        Toast.makeText(this@ScannerActivity,  "접근이  허용되지  않았습니다.",
Toast.LENGTH_SHORT).show()
                        var intent = Intent(applicationContext, close::class.java)
                        startActivity(intent)
                    }
                }

                try {
                    val monRegion = Region("myMonitoringUniqueId",
Identifier.parse(beaconUuid), null, null)
                    val rangRegion = Region("myRangingUniqueId",
Identifier.parse(beaconUuid), null, null)
                    beaconManager.startMonitoringBeaconsInRegion(monRegion)
                    beaconManager.startRangingBeaconsInRegion(rangRegion)
                    beaconManager.addRangeNotifier(rangeNotifier)
                }
                catch (e: RemoteException) { }
                catch (il: IllegalArgumentException) {
                    Toast.makeText(applicationContext, "Invalid UUID",
Toast.LENGTH_LONG).show()
                    finish()
                }

            }
        }
```

Server Code

```
from django.shortcuts import render

from django.http import HttpResponse, JsonResponse

from django.views.decorators.csrf import csrf_exempt

from .models import ClientData

from .serializers import ClientDataSerializer

from rest_framework.parsers import JSONParser

from django.contrib.auth import authenticate

from django.views.decorators.csrf import csrf_exempt

from django.utils.decorators import method_decorator

import time

import sqlite3

import serial



# For arduino serial port

arduino = serial.Serial(

    port='/dev/ttyACM0',

    baudrate=9600,

)

print('Connected Serial Port is ' + arduino.portstr)
```

```
@csrf_exempt
def client_list(request, format=None): # app --(identification, password, phone_number, name)-
-> server
    if request.method == 'GET': # 전체조회
        query_set = ClientData.objects.all()
        serializer = ClientDataSerializer(query_set, many=True)
        return JsonResponse(serializer.data, safe=False)

    elif request.method == 'POST': # 회원가입_test완료
        identification = request.POST.get("identification", "")
        password = request.POST.get("password", "")
        phone_number = request.POST.get("phone_number", "")
        name = request.POST.get("name", "")

        print('identification = ' + identification + 'password = ' + password + 'phone_number
= ' + phone_number + 'name= ' + name) # 서버쪽 터미널에 띄움
        myuser = ClientData.objects.filter(identification=identification)

        if myuser: # db에 저장되어있으면 -> id중복
            print("duplicated id, signUp failed") # for server debugging
            return JsonResponse({'code':'400', 'msg':'duplicated id'}, status=400)
        else: # new client면 -> db저장
            form = ClientData(identification=identification, password=password,
phone_number=phone_number, name=name)
            form.save()
            print("signUp success") # for server debugging
            return JsonResponse({'code':'201', 'msg':'signup success'}, status=201) # app으로
보내는 msg
```

```
@method_decorator(csrf_exempt, name='dispatch')
def login(request, format=None): # app --(identification, password)--> server --(allowed_area)--> app
    if request.method == "GET":
        return render(request, 'client_data/login.html')

    elif request.method == 'POST':
        identification = request.POST.get("identification", "")
        password = request.POST.get("password", "")
        myuser = ClientData.objects.filter(identification=identification, password=password)

        print("identification = " + identification + " password" + password)

        if myuser:
            print("login success")

        obj = ClientData.objects.get(identification=identification)
        phone_number = obj.phone_number
        name = obj.name

        print("identification = " + identification + " password" + password)
        print("phone:" + phone_number + "name:" + name)

        if myuser:
            print("login success")
            try:
                con = sqlite3.connect("db.sqlite3")
                cursor = con.cursor()
                db = cursor.execute("SELECT allowed_area FROM small_business_businessdata WHERE phone_number='%s' AND name='%s'" %(phone_number, name)).fetchall()[0][0]
                return JsonResponse({'code':'201', 'msg':'login success', 'allowed_area' : db}, status=201)
            except IndexError: # business_data db에 없을 때 -> guest일때
                return JsonResponse({'code':'201', 'msg':'login success', 'allowed_area' : "nothing:nothing,nothing:nothing"}, status=201)
        else:
            print("login failed")
            return JsonResponse({'code':'400', 'msg':'login failed'}, status=400)

        # password 넘길때 암호화 필요. -> 추가하기
```

```
@csrf_exempt
def door_open(request, format=None): # app --(id, uuid)--> server
    if request.method == "GET":
        return render(request, 'client_data/login.html')
    if request.method == 'POST':
        id = request.POST.get("id", "")
        uuid = request.POST.get("uuid", "")

        print("<door_open> id = " + id + " uuid" + uuid)
        if(id == '0'): # 사업자 -> 바로 문 열어준다.
            # from .ctr_servo import run_servo
            # run_servo(1) # run servo Motor
            arduino.write([1])
            data = arduino.read()
            print(data)

            return JsonResponse({'code':'201', 'msg':'door open!'}, status=201)
        elif(int(id) > 0): # guest일 때 -> 현재시각과 service start 한 시간 비교
            now = round(time.time())
            print("now time is:" + str(now))
            con = sqlite3.connect("db.sqlite3")
            cursor = con.cursor()
            start_time = cursor.execute("SELECT start_time FROM small_business_businessdata
WHERE id='%d'" %(int(id))).fetchall()[0][0]
            if(now - int(start_time) >= 7200): # service time 이 두시간 이상일 때
                db = cursor.execute("DELETE FROM small_business_businessdata WHERE
id='%d'" %(int(id)))
                print("service time done")
                return JsonResponse({'code':'400', 'msg':'service time done'}, status=400)
            else:
                # from .ctr_servo import run_servo
                # run_servo(1) # run servo Motor
                arduino.write([1])
                data = arduino.read()
                print(data)
                return JsonResponse({'code':'201', 'msg':'door open!'}, status=201)
        else :
            return JsonResponse({'code':'400', 'msg':'door not open'}, status=400)
```

```
@csrf_exempt
def first_qr_scan(request, format=None): # app --(store, allowed_data)--> server --(id)--> app
    if request.method == 'POST':
        store = request.POST.get("store", "")
        allowed_area = request.POST.get("allowed_area", "")

        print("from app) store: " + store + ", allowed_area: " + allowed_area)

        start_time = str(round(time.time()))
        print("start time is : " + start_time)

        con = sqlite3.connect("db.sqlite3")
        cursor = con.cursor()
        db = cursor.execute("INSERT INTO small_business_businessdata (store, allowed_area, start_time) VALUES ('%s', '%s', '%s')" %(store, allowed_area, start_time))
        id = cursor.execute("SELECT id FROM small_business_businessdata WHERE start_time='%s'" %(start_time)).fetchall()[0][0]
        con.commit()
        con.close()
        print("db insert result id:" + str(id))

        if (id >= 0):
            return JsonResponse({'code':'201', 'msg': str(id)}, status=201)
        else:
            return JsonResponse({'code':'400', 'msg':'store into db as guest failed'}, status=400)
```

Pyserial Code

```
import serial

arduino = serial.Serial(
    port='/dev/ttyACM0',
    baudrate=9600,
)

print('Connected Serial Port is ' +  arduino.portstr)

while True:
    arduino.write([0])
    arduino.write([1])
    data = arduino.read()
    print(data)
```

Arduino Code

```
#include <Servo.h>
Servo servo;
int servoPin = 1;
int led = 2;

int data = 0; // for serial
int angle = 0; // for servo_motor

void setup() {
    servo.attach(servoPin);
    Serial.begin(9600);
    pinMode(led, OUTPUT);
}
```

```
void servo_open(){
   for(angle = 0; angle < 90; angle++){
     servo.write(angle);
   }
   for(angle = 90; angle > 0; angle--){
     servo.write(angle);
 }
}
*/
void blk(){
   digitalWrite(led,HIGH);
   delay(1000);
   digitalWrite(led,LOW);
   delay(1000);
}

void loop() {
   data = Serial.read();
   if(data==1){
            blk();
     blk();
   }
   data = 0;
   Serial.print(data);
}
```