

프로젝트 최종보고서

금오도우미

1. 서론

*배경설명

- 단체나 기관에서 장애인들을 위한 시설을 구축하고 도움을 주려해도 작은 부분에서 장애인들이 느끼는 불편함이 있을 꺼라 생각하였다, 자연스럽게 이들을 도와줄 수 있는 서비스를 생각하게 되었다.

*개발동기

- 기존 교내 국가근로 도우미 학생의 시간제약의 이유로 도와주지 못하는 맹점을 해결하기 위해 개발을 시작하였다.
기존 국가근로 도우미와는 다르게 정해진 학생이 아닌 어플을 설치하고, 교내에 있는 학생이라면 누구나 도와줄 수 있는 시스템 구축해보고자 한다.

*개발 목표

- 스마트폰 App을 통하여 장애학우와 도우미 학우를 연결해주는 것을 기본 목표로 한다.
도움을 필요로 하는 학생이 위치, 도움 내용을 전송하면 도우미 학생의 핸드폰으로 알람이 가게 되고 그것을 확인한 도우미 학생이 요청을 확인한 뒤 찾아가 도움을 주는 형식으로 시스템을 구현하고자 한다.

*관련기술 현황

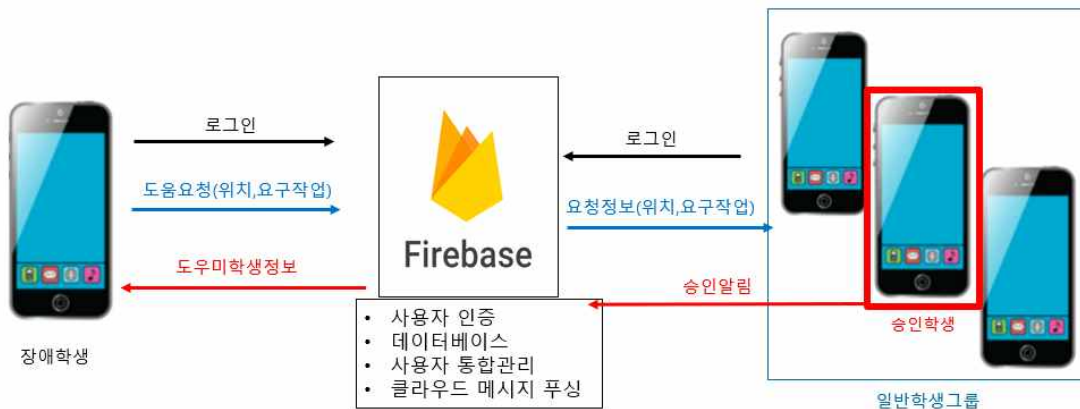
- App Store, Google Play Store등을 조사해본결과 우리가 만들고자하는 일정범위, 지정된 단체를 대상으로 하여 도움을 요청하고 도움을 주는 서비스는 없는 것으로 확인되었다.

2. 작품 개요

<설계구성요소 및 설계제한 요소>

설계 구성요소						설계 제한요소						
목표 설정	합성	분석	구현/제작	시험/평가	결과 도출	성능	규격/표준	경제성	미학	신뢰성	안정성/내구성	환경
○		○	○	○	○		○			○		○

- 전체 시스템 USE-CASE



2.1. 설계 구성요소

- 목표 설정

- 네트워크를 이용한 스마트폰 간 통신으로 장애학생과 일반학생 간 푸쉬 알림을 통한 도움 호출 및 도움 호출 수신

- 분석

- 장애인들을 위한 편의시설이 많이 보급되고 보편화 되는 추세이나 아직도 위생시설이나 안내 시설 등 일부 시설은 보급률이 떨어지는 편이다. 또, 이미 설치된 편의시설이라고 하더라도 해당 시설들 또한 장애인들이 혼자서 사용하기에는 불편함이 따른다는 장애인들의 조사 결과가 있다. 해당 결과에 따라 장애인들은 누군가의 도움이 필요한 필연적인 상황이 있다.

또한, 교내에는 근로 형태의 학습 도우미 학생이 존재하나 학습 도우미의 경우 장애 학생의 시간표에 맞추어 항상 상주하여야 한다는 점 및 그리고 다양한 형태의 도움에 학습 도우미가 모든 형태의 도움을 해결해줄 수 없다는 한계점이 있다.

- 구현/제작

- 어플리케이션 제작 : Android Studio
- Back-End 및 DB : FireBase
- 테스트 환경 : 안드로이드 휴대폰 및 ADB 가상 환경

- 시험/평가

- 다양한 네트워크 환경에서 장애학생-일반학생 간 상호 통신 및 푸쉬가 잘 이루어지는 지 확인한다.
- 상호 간 푸시 알림 및 내역 전송이 원활하게 DB에 저장되는 지 확인한다.

- 결과 도출

- 어플리케이션 현장시연 및 Android Studio 내의 Virtual Machine을 이용한 시연 영상

2.2 설계 제한요소

- 규격/표준

- 일반학생 뿐만 아니라 신체가 불편한 장애학생들 또한 원활하게 사용할 수 있는 형태의 어플리케이션 제작
- 장애 학생들을 식별하여 장애 학생들에게만 도움을 요청할 수 있는 권한을 부여

- 신뢰성

- 일반 학생 및 장애 학생 간의 도움 요청 및 푸시 알림이 누락 없이 원활하게 전송

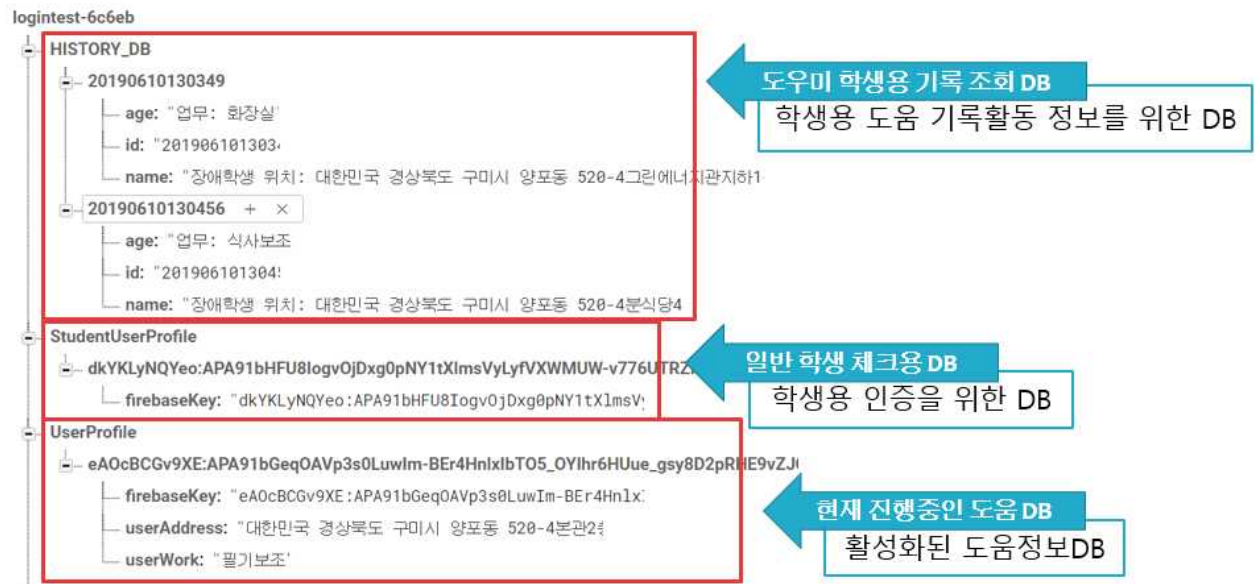
- 환경

- 교내에서만 제공하는 서비스로서 교내를 제외한 장소에서는 푸시 알림 혹은 도움 요청이 뜨지 않도록 제한
-

3. 개발 내용

- 상세 설계 내역

1. Database 구조



3가지 RealTime Database로 구성되며 일반 학생용 도움 기록정보(히스토리)기능을 위한 HISTORY_DB, 장애학생의 도움정보를 일반학생세션 에게만 알림 푸시를 보내기위한 StudentUserProfile, 현재 활성화된 도움정보를 알려주는 UserProfile로 데이터베이스가 구성됨

2. 사용자 등록(회원가입) 및 인증 부분구현



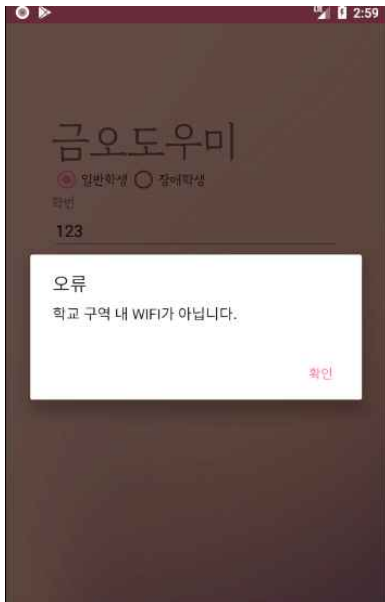
<Firebase 사용자 인증(Authentication)>

<회원가입 화면>

<로그인 화면>

일반학생의 경우 회원가입시 학번에 “@student.com” 장애학생의 경우 회원가입시 학번에 “@disabled.com”을 붙여 각각 등록 및 인증 시스템을 관리함. 해당 도메인문구(문자열)을 통해 관리하여 일반학생이 장애학생으로 장애학생이 일반학생으로 로그인 하지 못하게 구현함.

3. Wifi 구역망 확인

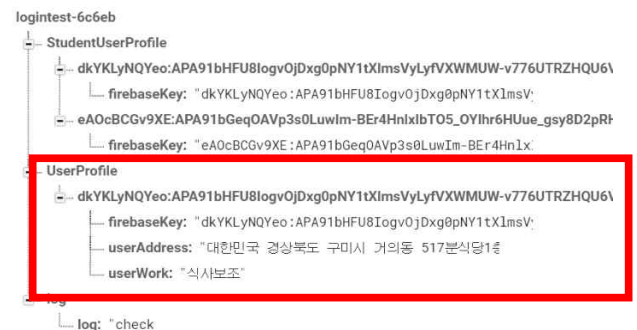
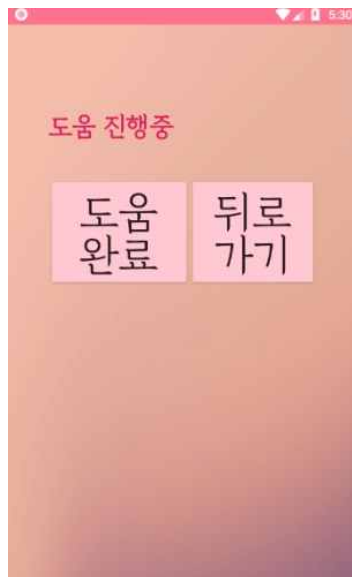
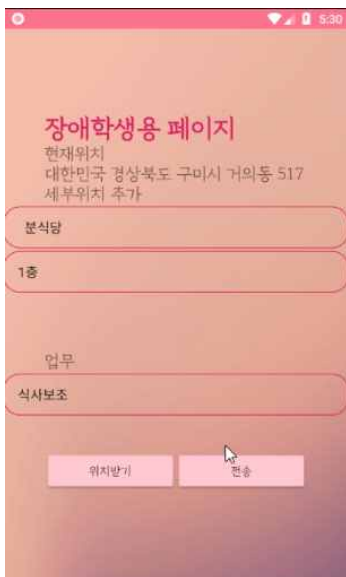


안드로이드SDK의 WifiManager라이브러리의 SSID값(교내의 경우 “kumoh-web”, 또는 “kumoh_mobile”)을 받아와 로그인시 SSID값이 특정 SSID 인지 검증 단계를 거침

해당 검증 실패시 오류 안내 문구를 통해 사용자에게 알려줌

<특정 Wifi 구역이 아닐시 오류안내>

4. 장애학생 페이지 구현



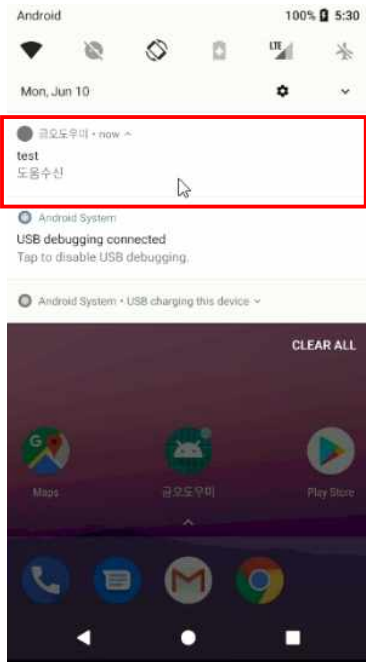
<도움요청시 RealTime Database값 저장>

<도움 요청 페이지>
(장애학생 페이지 초기화면)

<도움 진행중 페이지>

장애학생페이지는 현재 기본위치는 안드로이드 디바이스에서 위,경도를 받아와 해당 위,경도를 Geocoder 모듈을 통해 좌표에서 한글 주소로 변환 하여 기본 위치 출력 추가적으로 장애학생이 학교내 건물명과 층수 그리고 필요한 업무명을 선택하여 해당 정보를 Firebase의 RealTime Database에 저장함

5. 일반학생의 안드로이드 메시지 푸시 수신 구현

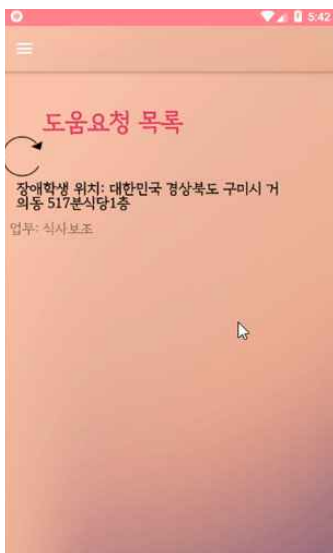


FCM(Firebase Cloud Messaging)을 활용하여 파이어 베이스 프로젝트를 키 값으로 하여 일반학생세션 데이터베이스를 참조하여 일반학생 기기를 접근함

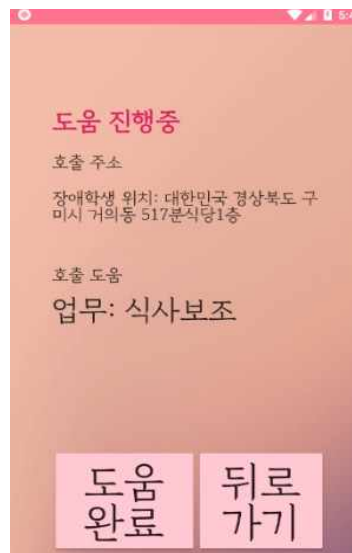
해당 푸시를 위해선 HTTP 통신이 필요한데(REST API) 안드로이드 SDK 의 Okhttp와 라이브러리를 통해 HTTP요청 및 전송을 POST방식으로 전송하며 Gson 라이브러리를 통해 객체들을 JSON 형식으로 바꾸어 메시지를 푸시함

<장애 학생이 도움요청시 메시지 푸시 결과>

6. 일반 학생용 페이지 구현



<도움 요청목록 화면>
(일반학생 페이지 초기화면)



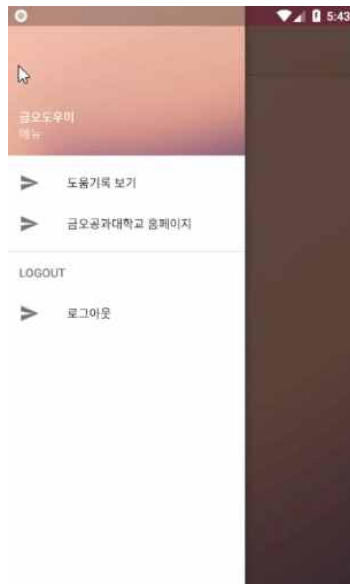
<도움 진행중 페이지>



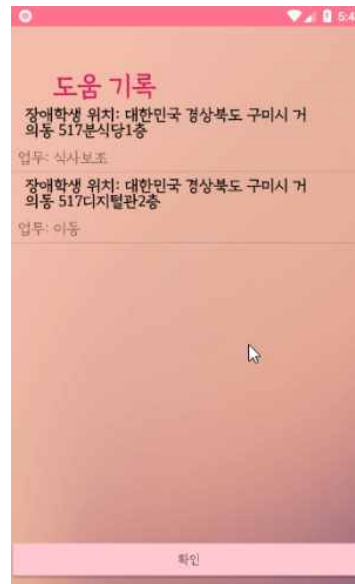
<도움 완료시 RealTime Database 상태>

일반학생은 로그인시 도움 요청목록이 보여짐. 해당 목록은 Android SDK에서 FirebaseDatabase를 참고하여 커스텀 리스트로 실시간으로 보여지게 됨. 해당 항목을 누르면 바로 도움이 진행되며 일반학생과 장애학생간의 도움이 끝난다면(두 명 모두 도움 완료를 누른다면) Firebase의 RealTime Database는 현재 진행 중인 도움 목록을 삭제하고 해당 데이터를 일반 학생의 도움 기록용 데이터베이스에 해당 값을 저장함.

7. 일반학생용 도움 기록 페이지



<네비게이션 뷰 화면>



<도움 기록페이지 화면>

안드로이드 좌측 메뉴창 인터페이스인 네비게이션뷰를 통해 각 항목은 도움기록, 학교페이지, 로그아웃 페이지를 구현함.

도움기록페이지는 커스텀리스트로 출력하며 해당 목록은 Android SDK에서 FirebaseDatabase를 참고하여 실시간으로 보여 지게 됨.

4. 결 론

- 개발 내용 및 실험 결과 요약

Firestore의 실시간 데이터베이스와 Firebase 클라우드 메시지인 FCM을 이용한 장애학우 도우미 앱으로 장애 학생과 도우미 학생을 앱을 통해 장애 학생들의 학교생활 처우 개선

- 향후 개선 과제

도우미 학생들에 대한 보상안 구상

악용방지를 위한 조건 추가 및 이용 약관 추가

추가적인 통신환경(비콘 인증, 블루투스인증) 구축을 통한 도움 인증 연락

- 기술적, 사회적, 경제적 파급 효과 및 기대 효과

본 어플리케이션은 현 상태에서는 금오공과대학교에 한정하여 사용이 가능하지만 조금 더 구체적인 시스템을 구축하여 관공서나 대형시설에서 사용할 수 있게 된다면 이를 담당하는 인력의 업무 부담 또한 줄어들것으로 생각되며, 합리적인 보상이 제공된다면 어플리케이션 이용률이 늘어나 장애인들이 일상생활에서 겪는 어려움이 일정부분 해소 될 것이라 생각됨

5. 자체 평가서

5.1 자체 분석과 평가

팀장 : 윤예준	<p>안드로이드 어플리케이션 특성상 여러 가지 권한과 각각의 라이브러리버전으로 인한 gradle정보(빌드정보)에서 문제가 많아 이를 해결하는 문제에 시간이 많이 소요되었다.</p> <p>초기 프로젝트에는 비콘을 통한 통신을 구상 하였으나 쌍방향 통신에서 어려움이 많아 Wifi인증을 통한 통신으로 구현을 하였다.</p> <p>데이터베이스부분에서 개발 초기에는 관계형 데이터베이스인 MySQL을 통해 구현을 하였는데 실시간으로 값이 처리되고 각각의 값들이 key 값과 value 값으로 저장할 수 있는 Firebase의 실시간 데이터베이스로 프로젝트를 작성하여 원할 하게 수행 할 수 있었다. 푸시알림기능에서도 예전 푸시알림 모듈인 GCM이 현재의 FCM으로 완전히 서비스가 바뀌어 참고할 대상이 적어져 어려움이 많았다.</p> <p>전체적인 어플리케이션은 현재 단순히 도우미와 장애학생간의 요청 수신 어플리케이션에 지나지 않지만 보상제도등의 기능 장애인친화적인 UI들을 만든다면 더 나은 어플리케이션이 될 거라 예상이 된다.</p>
팀원 : 김동한	<p>MySQL같은 DB가 아닌 FireBase를 사용한건 좋은 선택이라고 생각 한다</p> <p>구글이 안드로이드 표준을 기존 플러그인에서 FireBase로 넘어가는 시점에서 해당 기능을 사용하여 구현한 것만으로도 의미 있는 프로젝트였다고 생각하며 향후 어플리케이션개발에 있어 유용하게 사용될 것이라고 생각한다.</p> <p>조금 더 디테일한 세부기능이 구현되었다면 더 좋았을 것이라는 생각을 하게 된다.</p>
팀원 : 임지범	<p>수행 과정 중에는 관련 프로젝트 경험이 많지 않아 미흡한 점이 많았으나 관련 정보나 API 등을 찾아 활용하면서 비교적 수월하게 수행할 수 있었다고 생각한다. 결과물은 완벽하지는 않지만 Firebase와 Firebase Cloud Messaging 을 이용하여 핵심적인 기능은 일부 구현하였으므로 조금 더 시간을 갖고 다듬는다면 완성도 있는 어플리케이션이 될 것이라고 생각한다.</p>
팀원 : 이동현	<p>프로젝트를 진행하면서 모르는 부분도 많고 팀원들끼리 의견 조율이 힘들었지만 많은 대화와 회의를 통하여 의견을 맞춰가는 과정에서 팀 프로젝트에 대한 효율성을 느끼면서 팀 프로젝트에 대하여 많은 공부가 되고 개발 과정 중에서 평상시에 경험이 많지 않은 분야여서 부족한 부분이 많았지만 정보 수집과 회의를 통하여 관련 API를 잘 찾아서 활용하여 프로젝트 결과물을 냈습니다. 완벽한 개발은 아니지만 Firebase와 Firebase Cloud Messaging 을 통하여 어플리케이션 완성도를 높였습니다. 각자 모르는 부분이 많아서 시간이 많이 부족하여 완벽한 결과물은 내지 못하였지만 팀 프로젝트를 통하여 팀의 중요성 효율성 실용성 많은 것을 배웠습니다.</p>

5.2 팀원 역할 및 기여도

팀원	역할	기여도
팀장 : 윤예준	어플리케이션 핵심 기능 구현 및 프로젝트 진행	35%
팀원 : 김동한	부가기능 구현 및 UI 디자인	22%
팀원 : 임지범	부가기능 구현 및 UI 디자인	22%
팀원 : 이동현	부가기능 구현 및 서류, 회의록 담당	21%

6. 부록

소스코드 핵심 부분

● GPS 위치받기 권한 문제 해결 관련

```
1 // 기본 위치정보 + 사용자 추가위치 정보 불러옴
2 sendbt_location.setOnClickListener(new Button.OnClickListener() {
3     public void onClick(View v) {
4         if (ActivityCompat.checkSelfPermission(DisableMain.this, Manifest.permission.ACCESS_FINE_LOCATION)
5             != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(DisableMain.this,
6             Manifest.permission.ACCESS_COARSE_LOCATION)
7             != PackageManager.PERMISSION_GRANTED) {
8             if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
9                 requestPermissions(new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 1); //위치권한 탐색 허용 관련 내용
10            }
11            return;
12        }
13        Location location = lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
14        longitude = location.getLongitude();
15        latitude = location.getLatitude();
16        getLocation(latitude, longitude);
17
18        String cut = addressString;
19        int idx1 = cut.indexOf("W");
20        String cut1 = cut.substring(idx1 + 1);
21        int idx2 = cut1.indexOf("W");
22        String cut2 = cut1.substring(0, idx2);
23        addressString = cut2;
24        // 위치 테스트(콘솔)
25        System.out.println(cut1);
26        // System.out.println(cut2);
27        address.setText("" + cut2);
28    }
29 });
```

● OkHttp를 통한 FCM 앱 푸시 기능 구현

```
1 // Gson okhttp 라이브러리 필요
2 // JSON방식으로 데이터 푸시 전송
3 // POSTMAN 체크
4 public void sendServerToFirebase(String studentToken) {
5     Gson gson = new Gson();
6     String userName = FirebaseAuth.getInstance().getCurrentUser().getDisplayName();
7     NotificationModel notificationModel = new NotificationModel();
8     notificationModel.to = studentToken;
9     notificationModel.notification.title = "test";
10    notificationModel.notification.text = "도움수신";
11
12    okhttp3.RequestBody requestBody = RequestBody.create(MediaType.parse("application/json; charset=utf8"),
13        gson.toJson(notificationModel));
14
15    okhttp3.Request request = new Request.Builder()
16        .header("Content-Type", "application/json")
17        .addHeader("Authorization", "key=파이어베이스 프로젝트 세션 키값 부분")
18        .url("https://fcm.googleapis.com/fcm/send")
19        .post(requestBody)
20        .build();
21    okhttp3.OkHttpClient okHttpClient = new OkHttpClient();
22    okHttpClient.newCall(request).enqueue(new Callback() {
23        @Override
24        public void onFailure(Call call, IOException e) {
25
26        }
27
28        @Override
29        public void onResponse(Call call, Response response) throws IOException {
30
31        }
32    });
33 }
```

Colored by Color Scripter

개발일정

(수행 기간 : 2019년 4월 1일 ~ 2019년 6월 30일)				
작품 제작 내용	3월	4월	5월	6월
프로젝트 수행 계획 수립				
통신연구 및 라이브러리 조사				
어플리케이션 UI 디자인				
어플리케이션 구현				
최종 결과 확인				

참고문헌

[1] 안드로이드 API Firebase 개발문서
<https://firebase.google.com/docs/android/setup?hl=ko>

[2] OkHttp 라이브러리
<https://square.github.io/okhttp/>

[3] Gson 라이브러리
<https://github.com/google/gson>
