

기말 과제

(두 지점의 중간 장소를 알려주는 앱)

과목명: 모바일 프로그래밍

학과: 컴퓨터공학과

학번: 2017E7005

이름: 김건희



1. 개요

우리는 많은 약속을 하면서 살아간다. 만나서 밥을 같이 먹거나, 카페를 가는 것도 수 많은 약속 중 하나이다. 이러한 계획을 실행하기 위해서는 만나는 약속 장소가 중요한데, 이러한 약속 장소는 보통 특정인에게만 유리하게 정해진다. 이러한 점을 해결하기 위해 중간 지점을 구해주는 어플리케이션 ‘미드모여!’를 만들게 되었다. ‘미드모여!’는 게임에서 흔히 사용되는 말로, 모두 단합하여 좋은 결과를 이뤄내자는 뜻으로 사용된다. 또한 미드는 ‘중앙’이라는 뜻도 내포하고 있어 어플리케이션 이름을 이렇게 짓게 되었다.



2. 클래스 다이어그램



3. 소스코드(4p~22p 별도 첨부)

MapsActivity.java (메인 액티비티)

```
public class MapsActivity extends FragmentActivity implements
OnMapReadyCallback {
    private Gps gpsTracker; //gps 센서 이용.
    private GoogleMap mMap;
    private Geocoder geocoder; //위도, 경도와 주소 변환
    private Button search_button; //검색버튼
    private Button select_guide; //장소 선택 도우미 버튼
    private Button remove_button; //맵의 모든 마커 지우기 버튼
    private Button get_middle_button; //중간 장소 구하기 버튼
    private Button additional_menu; //부가 메뉴 버튼
    private EditText address; //검색 창
    private static int mark_count = 0; //지점의 개수 저장.
    private String[][] lat_long = new String[3][2]; //각 지점에 대한 위도, 경도
    정보
    private double lat1 = 0;
    private double lon1 = 0;
    private double lat2 = 0;
    private double lon2 = 0;
    private String Place_name; //즐거찾기 북마크 이름
    private static final int GPS_ENABLE_REQUEST_CODE = 2001;
    private static final int PERMISSIONS_REQUEST_CODE = 100;
    String[] REQUIRED_PERMISSIONS = {Manifest.permission.ACCESS_FINE_LOCATION,
    Manifest.permission.ACCESS_COARSE_LOCATION};
    PlaceDatabaseManager databaseManager; //데이터베이스
```



```

70 protected void onCreate(Bundle savedInstanceState) {
71     super.onCreate(savedInstanceState);
72     setContentView(R.layout.activity_maps);
73     address = (EditText) findViewById(R.id.address);
74     search_button = (Button) findViewById(R.id.search_button);
75     select_guide = (Button) findViewById(R.id.Load_data);
76     remove_button = (Button) findViewById(R.id.remove_button);
77     get_middle_button = (Button) findViewById(R.id.get_middle);
78     additional_menu = (Button) findViewById(R.id.additional_menu);
79
80     databaseManager = PlaceDatabaseManager.getInstance(this);
81
82     databaseManager.delete( whereClause: null, whereArgs: null);
83     if (!checkLocationServicesStatus()) { //안드로이드 구버전
84         showDialogForLocationServiceSetting();
85     } else {
86         checkRunTimePermission();
87     }
88
89     notice(); //시작 안내
90
91     // Obtain the SupportMapFragment and get notified when the map is ready to be used.
92     //구글 지도 불러오기
93     SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
94         .findFragmentById(R.id.map);
95     mapFragment.getMapAsync( onMapReadyCallback: this);
96 }
97
98 public void notice(){ //시작 안내문 출력
99     AlertDialog.Builder oDialog = new AlertDialog.Builder( context: this,
100         android.R.style.Theme_DeviceDefault_Light_Dialog_Alert);
101
102     String strHtml =
103         "이 어플리케이션은 입력한 '두 지점'의<b><font color='#ff0000'> 중간거리 </font></b>를 구해주는 APP입니다.<br>주소창"
104     Spanned oHtml;
105
106     if (android.os.Build.VERSION.SDK_INT < android.os.Build.VERSION_CODES.N) {
107         // noinspection deprecation
108         oHtml = Html.fromHtml(strHtml);
109     }
110     else
111     {
112         oHtml = Html.fromHtml(strHtml, Html.FROM_HTML_MODE_LEGACY);
113     }
114
115     oDialog.setTitle("안내")
116         .setMessage(oHtml)
117         .setPositiveButton( text: "ok", listener: null)
118         .setCancelable(false)
119         .show();
120 }

```



```

125 public void onRequestPermissionsResult(int permsRequestCode,
126                                     @NonNull String[] permissions,
127                                     @NonNull int[] grantResults) {
128     if (permsRequestCode == PERMISSIONS_REQUEST_CODE && grantResults.length == REQUIRED_PERMISSIONS.length) {
129         // 요청 코드가 PERMISSIONS_REQUEST_CODE 이고, 요청한 퍼미션 개수만큼 수신되었다면
130         boolean check_result = true;
131         // 모든 퍼미션을 허용했는지 체크합니다.
132         for (int result : grantResults) {
133             if (result != PackageManager.PERMISSION_GRANTED) {
134                 check_result = false;
135                 break;
136             }
137         }
138
139         if (check_result) {
140             // 위치 값을 가져올 수 있음
141             ;
142         } else {
143             // 거부한 퍼미션이 있다면 앱을 사용할 수 없는 이유를 설명해주고 앱을 종료합니다. 2 가지 경우가 있습니다.
144             if (ActivityCompat.shouldShowRequestPermissionRationale( activity: this, REQUIRED_PERMISSIONS[0])
145                 || ActivityCompat.shouldShowRequestPermissionRationale( activity: this, REQUIRED_PERMISSIONS[1])) {
146                 Toast.makeText( context: MapsActivity.this, text: "퍼미션이 거부되었습니다. 앱을 다시 실행하여 퍼미션을 허용해주세요.", Toast.LENGTH_SHORT).show();
147                 finish();
148             } else {
149                 Toast.makeText( context: MapsActivity.this, text: "퍼미션이 거부되었습니다. 설정(앱 정보)에서 퍼미션을 허용해야 합니다. ", Toast.LENGTH_SHORT).show();
150             }
151         }
152     }
153 }
154
155 void checkRunTimePermission() {
156     // 런타임 퍼미션 처리
157     // 1. 위치 퍼미션을 가지고 있는지 체크합니다.
158     int hasFineLocationPermission = ContextCompat.checkSelfPermission( context: MapsActivity.this,
159                             Manifest.permission.ACCESS_FINE_LOCATION);
160     int hasCoarseLocationPermission = ContextCompat.checkSelfPermission( context: MapsActivity.this,
161                             Manifest.permission.ACCESS_COARSE_LOCATION);
162
163     if (hasFineLocationPermission == PackageManager.PERMISSION_GRANTED &&
164         hasCoarseLocationPermission == PackageManager.PERMISSION_GRANTED) {
165         // 2. 이미 퍼미션을 가지고 있다면
166         // ( 안드로이드 6.0 이하 버전은 런타임 퍼미션이 필요했기 때문에 이미 허용된 걸로 인식합니다.)
167         // 3. 위치 값을 가져올 수 있음
168
169     } else { // 2. 퍼미션 요청을 허용한 적이 없다면 퍼미션 요청이 필요합니다. 2가지 경우(3-1, 4-1)가 있습니다.
170         // 3-1. 사용자가 퍼미션 거부를 한 적이 있는 경우에는
171         if (ActivityCompat.shouldShowRequestPermissionRationale( activity: MapsActivity.this, REQUIRED_PERMISSIONS[0])) {
172             // 3-2. 요청을 진행하기 전에 사용자에게 퍼미션이 필요한 이유를 설명해줄 필요가 있습니다.
173             Toast.makeText( context: MapsActivity.this, text: "이 앱을 실행하려면 위치 접근 권한이 필요합니다.", Toast.LENGTH_SHORT).show();
174             // 3-3. 사용자에게 퍼미션 요청을 합니다. 요청 결과는 onRequestPermissionsResult에서 수신됩니다.
175             ActivityCompat.requestPermissions( activity: MapsActivity.this, REQUIRED_PERMISSIONS,
176                                     PERMISSIONS_REQUEST_CODE);
177         } else {
178             // 4-1. 사용자가 퍼미션 거부를 한 적이 없는 경우에는 퍼미션 요청을 바로 합니다.
179             // 요청 결과는 onRequestPermissionsResult에서 수신됩니다.
180             ActivityCompat.requestPermissions( activity: MapsActivity.this, REQUIRED_PERMISSIONS,
181                                     PERMISSIONS_REQUEST_CODE);
182         }
183     }
184 }
185

```



```

196 public String getCurrentAddress(double latitude, double longitude) {
197     //지오코더... GPS를 주소로 변환
198     Geocoder geocoder = new Geocoder( context: this, Locale.getDefault());
199     List<Address> addresses;
200     try {
201         addresses = geocoder.getFromLocation(
202             latitude,
203             longitude,
204             maxResults: 7);
205     } catch (IOException ioException) {
206         //네트워크 문제
207         Toast.makeText( context: this, text: "지오코더 서비스 사용불가", Toast.LENGTH_SHORT).show();
208         return "지오코더 서비스 사용불가";
209     } catch (IllegalArgumentException illegalArgumentException) {}
210     Toast.makeText( context: this, text: "잘못된 GPS 좌표", Toast.LENGTH_SHORT).show();
211     return "잘못된 GPS 좌표";
212 }
213
214 if (addresses == null || addresses.size() == 0) {
215     Toast.makeText( context: this, text: "주소 미발견", Toast.LENGTH_SHORT).show();
216     return "주소 미발견";
217 }
218 Address address = addresses.get(0);
219 return address.getAddressLine( index: 0).toString();
220 }
221
222 //여기부터는 GPS 활성화를 위한 메소드들
223 private void showDialogForLocationServiceSetting() {
224     AlertDialog.Builder builder = new AlertDialog.Builder( context: MapsActivity.this);
225     builder.setTitle("위치 서비스 비활성화");
226     builder.setMessage("앱을 사용하기 위해서는 위치 서비스가 필요합니다.\n"
227         + "위치 설정을 수정하실래요?");
228     builder.setCancelable(true);
229     builder.setPositiveButton( text: "설정", new DialogInterface.OnClickListener() {
230         @Override
231         public void onClick(DialogInterface dialog, int id) {
232             Intent callGPSSettingIntent
233                 = new Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
234             startActivityForResult(callGPSSettingIntent, GPS_ENABLE_REQUEST_CODE);
235         }
236     });
237     builder.setNegativeButton( text: "취소", new DialogInterface.OnClickListener() {
238         @Override
239         public void onClick(DialogInterface dialog, int id) { dialog.cancel(); }
240     });
241     builder.create().show();
242 }

```



```

248 public boolean checkLocationServicesStatus() {
249     LocationManager locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
250
251     return locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)
252         || locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
253 }
254
255 @Override
256 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
257     super.onActivityResult(requestCode, resultCode, data);
258     switch (requestCode) {
259         case GPS_ENABLE_REQUEST_CODE:
260             //사용자가 GPS 활성화 시켰는지 검사
261             if (checkLocationServicesStatus()) {
262                 if (checkLocationServicesStatus()) {
263                     Log.d( tag: "###", msg: "onActivityResult : GPS 활성화 되었음");
264                     checkRunTimePermission();
265                     return;
266                 }
267             }
268             break;
269     }
270 }
271
272 public void onStop() { super.onStop(); }
273
274 public void onResume() { super.onResume(); }
275
276 @
277 public double getDistance(LatLng latLng1, LatLng latLng2) { // 두 지점의 거리를 구하는 모듈.
278     double distance = 0;
279
280     Location locationA = new Location( provider: "A");
281     locationA.setLatitude(latLng1.latitude);
282     locationA.setLongitude(latLng1.longitude);
283
284     Location locationB = new Location( provider: "B");
285     locationB.setLatitude(latLng2.latitude);
286     locationB.setLongitude(latLng2.longitude);
287
288     distance = locationA.distanceTo(locationB);
289
290     return distance;
291 }

```




```

297 public void get_Current_Gps() { //현재 위치를 구해 마킹하고, 알려주는 모듈.
298     gpsTracker = new Gps( context: MainActivity.this);
299     double latitude = gpsTracker.getLatitude();
300     double longitude = gpsTracker.getLongitude();
301
302     if (mark_count == 0) { //중간지점 구하고나서 맵 클리어.
303         mMap.clear();
304     } else if (mark_count == 2) {
305         Toast.makeText(getApplicationContext(), text: "이미 두 지점이 마킹되었습니다. 중간 지점을 구해주세요.", Toast.LENGTH_SHORT).show();
306         return;
307     }
308
309     lat_long[mark_count][0] = Double.toString(latitude);
310     lat_long[mark_count][1] = Double.toString(longitude);
311     String Current_address = getCurrentAddress(latitude, longitude);
312
313     LatLng Mypoint = new LatLng(latitude, longitude);
314     // 마커 생성
315     MarkerOptions mOptions2 = new MarkerOptions();
316     mOptions2.title("현재 GPS 주소");
317     mOptions2.snippet(Current_address);
318     mOptions2.position(Mypoint);
319     // 마커 추가
320
321     mMap.addMarker(mOptions2);
322     mark_count += 1;
323
324     // 해당 좌표로 화면 줌
325     mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(Mypoint, V: 17));
326     Toast.makeText(getApplicationContext(), text: "현재 위치는 \n" + Current_address + "입니다.", Toast.LENGTH_SHORT).show();
327 }
328
329 public void getPlaceData() { //즐거찾기를 불러오는 모듈. Load_data_Activity로 데이터베이스의 정보를 intent를 통해 전송시킬.
330     String[] columns = new String[]{"_id", "NAME", "LATITUDE", "LONGITUDE"};
331     ArrayList<String> list = new ArrayList<>();
332     Cursor cursor = databaseManager.query(columns, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
333
334     if (cursor != null) {
335         while (cursor.moveToNext()) {
336             System.out.println("getPlaceData 성공");
337             for (int j = 1; j < 4; j++) {
338                 list.add(cursor.getString(j));
339             }
340         }
341         System.out.println("인텐트전 ArrayList는 " + list.size());
342         Intent intent = new Intent(getApplicationContext(), Load_data_Activity.class);
343         intent.putExtra( name: "data", list);
344         startActivity(intent);
345     }
346 }

```



```

347 public boolean onKeyDown(int keyCode, KeyEvent event){ //종료 안내 모듈.
348     switch(keyCode){
349         case KeyEvent.KEYCODE_BACK:
350             String alertTitle = "약속장소 정하셨나요?";
351             String buttonMessage = "어플을 종료하시겠습니까?";
352             String buttonYes = "Yes";
353             String buttonNo = "No";
354
355             new AlertDialog.Builder( context: MapsActivity.this)
356                 .setTitle(alertTitle)
357                 .setMessage(buttonMessage)
358                 .setPositiveButton(buttonYes, new DialogInterface.OnClickListener() {
359
360                     @Override
361                     public void onClick(DialogInterface dialog, int which) {
362                         // TODO Auto-generated method stub
363                         moveTaskToBack( nonRoot: true);
364                         finish();
365                     }
366                 })
367                 .setNegativeButton(buttonNo, listener: null)
368                 .show();
369         }
370     return true;
371 }
372
373 public void onMapReady(final GoogleMap googleMap) { //맵과 관련된 모듈. 클릭 이벤트 정의.
374     mMap = googleMap;
375     geocoder = new Geocoder( context: this);
376
377     get_middle_button.setEnabled(false);
378     get_middle_button.setTextColor(Color.parseColor( colorString: "#808080"));
379     // 버튼 이벤트
380     search_button.setOnClickListener(new Button.OnClickListener() { //검색 버튼
381         @Override
382         public void onClick(View v) {
383             String str = address.getText().toString();
384             List<Address> addressList = null;
385
386             if (mark_count == 2) {
387                 Toast.makeText(getApplicationContext(), text: "이미 두 지점이 마킹되었습니다. 중간 지점을 구해주세요.", Toast.LENGTH_SHORT).show();
388                 return;
389             }
390
391             try {
392                 // editText에 입력한 텍스트(주소, 지역, 장소 등)을 지오 코딩을 이용해 변환
393                 addressList = geocoder.getFromLocationName(
394                     str, // 주소
395                     maxResults: 10); // 최대 검색 결과 개수
396             } catch (IOException e) {
397                 Toast.makeText(getApplicationContext(), text: "주소를 입력해주세요.", Toast.LENGTH_SHORT).show();
398                 return;
399             }
400
401             if (addressList.size() == 0) {
402                 Toast.makeText(getApplicationContext(), text: "검색결과가 없습니다.", Toast.LENGTH_SHORT).show();
403             }
404             else {
405                 System.out.println(addressList.get(0).toString());
406                 // 콤마를 기준으로 split
407                 String[] splitStr;
408                 int lat_num = 10;
409                 int lon_num = 12;

```

```

422 System.out.println(addressList.get(0).toString());
423 if (addressList.get(0).toString().contains("(")) {
424     lat_num += 1;
425     lon_num += 1;
426 }
427 // 콤마를 기준으로 split
428 splitStr = addressList.get(0).toString().split( regex: "," );
429 System.out.println(splitStr[0].length());
430 String search_address = splitStr[0].substring(splitStr[0].indexOf("\\") + 1, splitStr[0].length() - 1); // 주소
431 System.out.println(search_address);
432
433
434 String latitude = splitStr[lat_num].substring(splitStr[lat_num].indexOf("=") + 1); // 위도
435 String longitude = splitStr[lon_num].substring(splitStr[lon_num].indexOf("=") + 1); // 경도
436 lat_long[mark_count][0] = latitude;
437 lat_long[mark_count][1] = longitude;
438 System.out.println(latitude);
439 System.out.println(longitude);
440
441 // 좌표(위도, 경도) 생성
442 LatLng point = new LatLng(Double.parseDouble(latitude), Double.parseDouble(longitude));
443 // 마커 생성
444 MarkerOptions mOptions2 = new MarkerOptions();
445 mOptions2.title("검색 결과");
446 mOptions2.snippet(search_address);
447 mOptions2.position(point);
448 // 마커 추가
449 if (mark_count == 0) {
450     mMap.clear();
451 }
452
453 mMap.addMarker(mOptions2);
454 mark_count += 1;
455 // 해당 좌표로 화면 줌
456 mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(point, V: 17));
457
458 if (mark_count == 2) {
459     System.out.println("버튼활성화");
460     get_middle_button.setEnabled(true);
461     get_middle_button.setTextColor(Color.parseColor( colorString: "#000000"));
462 }
463 }
464 }

```



```

467 select_guide.setOnClickListner(new View.OnClickListener() { //장소 선택 도우미.
468     @Override
469     public void onClick(View v) {
470         PopupMenu p = new PopupMenu(getApplicationContext(), v);
471         getMenuInflater().inflate(R.menu.Load_option_menu, p.getMenu());
472
473         // 이벤트 처리
474         p.setOnMenuItemClickListener(
475             new PopupMenu.OnMenuItemClickListener() {
476                 public boolean onMenuItemClick(MenuItem item) {
477                     switch (item.getItemId()) {
478                         case R.id.current_gps: //현재 gps
479                             if (mark_count == 2) {
480                                 Toast.makeText(getApplicationContext(), text: "이미 두 지점이 마킹되었습니다. 중간 지점을 구해주세요.", Toast.LENGTH_SHORT).show();
481                             } else {
482                                 get_Current_Gps();
483                             }
484                             break;
485                         case R.id.load_favorite_place: //즐겨찾기 불러오기
486                             getPlaceData();
487                             break;
488                         case R.id.get_address_site: //도로명주소 사이트
489                             new Thread(new Runnable() {
490                                 @Override
491                                 public void run() {
492                                     Intent place_intent = new Intent(Intent.ACTION_VIEW);
493                                     Uri place_uri = Uri.parse("https://www.juso.go.kr/openIndexPage.do");
494                                     place_intent.setData(place_uri);
495                                     startActivity(place_intent);
496                                 }
497                             }).start();
498                             break;
499                     }
500                     return true;
501                 }
502             });
503         p.show(); // 메뉴를 띄우기
504         if (mark_count == 2) {
505             System.out.println("버튼활성화");
506             get_middle_button.setEnabled(true);
507             get_middle_button.setTextColor(Color.parseColor("colorString: "#000000"));
508         }
509     }

```



```

511 remove_button.setOnClickListener(new View.OnClickListener() { //맵의 모든 마커를 지움.
512     @Override
513     public void onClick(View v) {
514         mMap.clear();
515         mark_count = 0;
516
517         for (int i = 0; i < 3; i++) {
518             lat_long[i][0] = "";
519             lat_long[i][1] = "";
520         }
521     }
522 });
523 get_middle_button.setOnClickListener(new View.OnClickListener() { //중간 거리를 구함.
524     @Override
525     public void onClick(View v) {
526         if (mark_count == 0) {
527             Toast.makeText(getApplicationContext(), text: "마킹된 지점이 없습니다. 검색을 통해 마킹해주세요.", Toast.LENGTH_SHORT).show();
528         } else {
529             lat1 = Double.parseDouble(lat_long[0][0]);
530             lon1 = Double.parseDouble(lat_long[0][1]);
531             lat2 = Double.parseDouble(lat_long[1][0]);
532             lon2 = Double.parseDouble(lat_long[1][1]);
533
534             LatLng latlng1 = new LatLng(lat1, lon1);
535             LatLng latlng2 = new LatLng(lat2, lon2);
536
537             Toast.makeText(getApplicationContext(), text: "입력하신 두 지점의 거리는 대략" + Math.round(getDistance(latlng1, latlng2) / 1000) + "km 입니다.", Toast.LENGTH_SHORT).show();
538             LatLng middle_point = new LatLng( (lat1 + lat2) / 2, (lon1 + lon2) / 2);
539
540             Geocoder geocoder = new Geocoder(getApplicationContext(), Locale.getDefault());
541             List<Address> addresses;
542             try { //위도, 경도를 통해 주소를 구함.
543                 addresses = geocoder.getFromLocation(
544                     latitude: (lat1 + lat2) / 2,
545                     longitude: (lon1 + lon2) / 2,
546                     maxResults: 7);
547             } catch (IOException ioException) {
548                 //네트워크 문제
549                 return;
550             }
551
552             Address address = addresses.get(0);
553
554             System.out.println(middle_point);
555             MarkerOptions mOptions = new MarkerOptions();
556             mOptions.title("두 지점의 중간 지점\n");
557             mOptions.snippet(address.getAddressLine( index: 0).toString());
558             mOptions.position(middle_point);
559             mOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE));
560             mMap.addMarker(mOptions);
561             mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(middle_point, V: 15));
562
563             lat_long[mark_count][0] = Double.toString( d: (lat1 + lat2) / 2);
564             lat_long[mark_count][1] = Double.toString( d: (lon1 + lon2) / 2);
565
566             mark_count += 1;
567         }
568     }
569 });

```



```

onal_menu.setOnClickListener(new View.OnClickListener() { //부가 메뉴
@Override
public void onClick(View v) {
    PopupMenu p = new PopupMenu(getApplicationContext(), v);
    getMenuInflater().inflate(R.menu.popupmenu, p.getMenu());
    // 이벤트 처리
    p.setOnMenuItemClickListener(
        new PopupMenu.OnMenuItemClickListener() {
            public boolean onMenuItemClick(MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.place_data: //해당 지역 근처 식당 정보
                        new Thread(new Runnable() {
                            @Override
                            public void run() {
                                Intent place_intent = new Intent(Intent.ACTION_VIEW);
                                Uri place_uri = Uri.parse("https://www.google.co.kr/maps/search/%EB%A0%88%EC%8A%A4%ED%86%A0%EB%9E%91/@ " + lat_long[mark_count][0] + "," + lat_long[mark_count][1]);
                                place_intent.setData(place_uri);
                                startActivity(place_intent);
                            }
                        }).start();

                        break;
                    case R.id.save_favorite_place: //현재 지점을 즐겨찾기에 등록.
                        if (mark_count == 0) {
                            Toast.makeText(getApplicationContext(), "text: "먼저 검색을 통해 마킹을 해주세요.", Toast.LENGTH_SHORT).show();
                        } else {
                            Bundle extras = getIntent().getExtras();
                            AlertDialog.Builder alert = new AlertDialog.Builder(context: MapsActivity.this);

                            alert.setTitle("장소 이름을 정해주세요.");
                            alert.setMessage("장소 이름");

                            final EditText name = new EditText(context: MapsActivity.this);
                            alert.setView(name);
                        }
                    }
                }
            }
        }
    );
}
}

```



```

604 alert.setPositiveButton( text: "ok", new DialogInterface.OnClickListener() {
605     public void onClick(DialogInterface dialog, int whichButton) {
606         Place_name = name.getText().toString();
607         dialog.dismiss();
608
609         if (Place_name != null) {
610             ContentValues addRowValue = new ContentValues();
611
612             addRowValue.put("NAME", Place_name);
613             addRowValue.put("LATITUDE", lat_long[mark_count - 1][0]);
614             addRowValue.put("LONGITUDE", lat_long[mark_count - 1][1]);
615
616             databaseManager.insert(addRowValue);
617             Place_name = null;
618         }
619     }
620 });
621 alert.create();
622 alert.show();
623 }
624 break;
625 }
626 return true;
627 }
628 });
629 p.show(); // 메뉴를 띄우기
630 }
631 });
632
633 // Add a marker in Sydney and move the camera
634 LatLng seoul = new LatLng( v: 37.56, v1: 126.97);
635 mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(seoul, v: 10));
636 }
637 }

```



PlaceDatabaseManager.java (즐거찾기 DB)

```

9  public class PlaceDatabaseManager {
10     static final String DB_PLACE = "favorite_place.db"; //DB 이름
11     static final String TABLE_PLACE = "fa_pl_"; //Table 이름
12     static final int DB_VERSION = 1; //DB 버전
13
14     Context myContext = null;
15
16     private static PlaceDatabaseManager myDBManager = null;
17     private SQLiteDatabase mydatabase = null;
18
19     //PlaceDatabaseManager 싱글톤 패턴으로 구현
20     public static PlaceDatabaseManager getInstance(Context context)
21     {
22         if(myDBManager == null)
23         {
24             myDBManager = new PlaceDatabaseManager(context);
25         }
26
27         return myDBManager;
28     }
29     public Cursor query(String[] columns,
30                         String selection,
31                         String[] selectionArgs,
32                         String groupBy,
33                         String having,
34                         String orderBy)
35     {
36         return mydatabase.query(TABLE_PLACE,
37                                 columns,
38                                 selection,
39                                 selectionArgs,
40                                 groupBy,
41                                 having,
42                                 orderBy);
43     }
44     @private PlaceDatabaseManager(Context context) //데이터베이스 생성
45     {
46         myContext = context;
47
48         //DB Open
49         mydatabase = context.openOrCreateDatabase(DB_PLACE, context.MODE_PRIVATE, factory: null);
50
51         //Table 생성
52         mydatabase.execSQL("CREATE TABLE IF NOT EXISTS " + TABLE_PLACE +
53                             "(" + "_id INTEGER PRIMARY KEY AUTOINCREMENT," +
54                             "NAME TEXT," +
55                             "LATITUDE TEXT," +
56                             "LONGITUDE TEXT" +
57                             ");");
58     }
59     public long insert(ContentValues addRowValue){
60         return mydatabase.insert(TABLE_PLACE, nullColumnHack: null, addRowValue);
61     }
62     public int delete(String whereClause,
63                      String[] whereArgs)
64     {
65         return mydatabase.delete(TABLE_PLACE,
66                                 whereClause,
67                                 whereArgs);
68     }
69 }

```



Gps.java (gps 센서 이용 클래스)

```

18 public class Gps extends Service implements LocationListener {
19     private final Context mContext;
20     Location location;
21     double latitude;
22     double longitude;
23
24     private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10;
25     private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1;
26     protected LocationManager locationManager;
27
28     public Gps(Context context) {
29         this.mContext = context;
30         getLocation();
31     }
32
33
34     public Location getLocation() {
35         try {
36             locationManager = (LocationManager) mContext.getSystemService(LOCATION_SERVICE);
37
38             boolean isGPSEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
39             boolean isNetworkEnabled = locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
40
41             if (!isGPSEnabled && !isNetworkEnabled) {
42             } else {
43                 int hasFineLocationPermission = ContextCompat.checkSelfPermission(mContext,
44                     Manifest.permission.ACCESS_FINE_LOCATION);
45                 int hasCoarseLocationPermission = ContextCompat.checkSelfPermission(mContext,
46                     Manifest.permission.ACCESS_COARSE_LOCATION);
47
48                 if (hasFineLocationPermission == PackageManager.PERMISSION_GRANTED &&
49                     hasCoarseLocationPermission == PackageManager.PERMISSION_GRANTED) {
50                     ;
51                 } else
52                     return null;
53
54                 if (isNetworkEnabled) {
55                     locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, listener: this);
56                     if (locationManager != null)
57                     {
58                         location = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
59                         if (location != null)
60                         {
61                             latitude = location.getLatitude();
62                             longitude = location.getLongitude();
63                         }
64                     }
65                 }
66
67                 if (isGPSEnabled)
68                 {
69                     if (location == null)
70                     {
71                         locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, listener: this);
72                         if (locationManager != null)
73                         {
74                             location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
75                             if (location != null)
76                             {
77                                 latitude = location.getLatitude();
78                                 longitude = location.getLongitude();
79                             }
80                         }
81                     }
82                 }
83             }
84         } catch (Exception e)
85         {
86             Log.d( tag: "Gps", msg: ""+e.toString());
87         }
88
89         return location;
90     }
91 }

```

```
93     public double getLatitude()
94     {
95         if(location != null)
96         {
97             latitude = location.getLatitude();
98         }
99
100         return latitude;
101     }
102
103     public double getLongitude()
104     {
105         if(location != null)
106         {
107             longitude = location.getLongitude();
108         }
109
110         return longitude;
111     }
112
113     @Override
114     public void onLocationChanged(Location location)
115     {
116     }
117
118     @Override
119     public void onProviderDisabled(String provider)
120     {
121     }
122
123     @Override
124     public void onProviderEnabled(String provider)
125     {
126     }
127
128     @Override
129     public void onStatusChanged(String provider, int status, Bundle extras)
130     {
131     }
132
133     @Override
134     public IBinder onBind(Intent arg0) { return null; }
135
136     public void stopUsingGPS()
137     {
138         if(locationManager != null)
139         {
140             locationManager.removeUpdates( listener: Gps.this);
141         }
142     }
143
144 }
```



ListViewItem.java(리스트 뷰 아이템)

```

1 package com.example.myproject;
2
3 //리스트뷰 속의 아이템(2개의 텍스트{이름, 위도경도})
4 public class ListViewItem {
5     private String nameStr ;
6     private String lat_lonStr ;
7
8     public void setName(String name) { nameStr = name; }
11    public void setLat_lon(String lat_lon) { lat_lonStr = lat_lon; }
14
15    public String getName() { return this.nameStr ; }
18    public String getLat_lon() { return this.lat_lonStr ; }
21 }
22

```

ListAdapter.java(리스트 뷰 어댑터 클래스)

```

11 //리스트뷰 구현을 위한 어댑터 파일.
12
13 public class ListViewAdapter extends BaseAdapter {
14     // Adapter에 추가된 데이터를 저장하기 위한 ArrayList
15     private ArrayList<ListViewItem> listViewItemList = new ArrayList<>() ;
16
17     // ListViewAdapter의 생성자
18     public ListViewAdapter() {
19     }
20
21     // Adapter에 사용되는 데이터의 개수를 리턴. : 필수 구현
22     @Override
23     public int getCount() { return listViewItemList.size() ; }
26
27     // position에 위치한 데이터를 화면에 출력하는데 사용될 View를 리턴. : 필수 구현
28     @Override
29     public View getView(int position, View convertView, ViewGroup parent) {
30         final int pos = position;
31         final Context context = parent.getContext();
32
33         // "listview_item" Layout을 inflate하여 convertView 참조 획득.
34         if (convertView == null) {
35             LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
36             convertView = inflater.inflate(R.layout.data_view, parent, attachToRoot: false);
37         }
38
39         // 화면에 표시될 View(Layout이 inflate된)으로부터 위젯에 대한 참조 획득
40         TextView nameTextView = (TextView) convertView.findViewById(R.id.textView1) ;
41         TextView lat_lonTextView = (TextView) convertView.findViewById(R.id.textView2) ;
42
43         // Data Set(ListViewItem)에서 position에 위치한 데이터 참조 획득
44         ListViewItem listViewItem = listViewItemList.get(position);
45
46         // 아이템 내 각 위젯에 데이터 반영
47         nameTextView.setText(listViewItem.getName());
48         lat_lonTextView.setText(listViewItem.getLat_lon());
49
50         return convertView;
51     }

```



```

52
53 // 지정한 위치(position)에 있는 데이터와 관계된 아이템(row)의 ID를 리턴. : 필수 구현
54 @Override
55 public long getItemId(int position) { return position ; }
58
59 // 지정한 위치(position)에 있는 데이터 리턴 : 필수 구현
60 @Override
61 public Object getItem(int position) { return listViewItemList.get(position) ; }
64
65 // 아이템 데이터 추가를 위한 함수. 개발자가 원하는대로 작성 가능.
66 public void addItem(String Name, String Lat_Lon) {
67     ListViewItem item = new ListViewItem();
68
69     item.setName(Name);
70     item.setLat_lon(Lat_Lon);
71
72     listViewItemList.add(item);
73 }
74 }
75

```

Load data Activity.java(DB속 데이터 출력.)

```

22 //줄여찾기 보여주는 파일.
23 public class Load_data_Activity extends Activity {
24     String name="";
25     String lat_lon="";
26     ArrayList<String> list = new ArrayList<>();
27     ListView listview ;
28     ListViewAdapter adapter;
29
30     Geocoder geocoder = new Geocoder( context: this, Locale.getDefault());
31     List<Address> addresses;
32     String tmp[] = new String[] { };
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.Load_data);
36
37         Intent intent = getIntent();
38
39         // Adapter 생성
40         adapter = new ListViewAdapter() ;
41
42         list = intent.getStringArrayListExtra( name: "data");
43         listview = (ListView) findViewById(R.id.Listview);
44         listview.setAdapter(adapter);
45
46         //intent로 데이터베이스의 정보를 받아들여옴.
47         for(int i = 0; i < list.size(); i++){
48             if(i % 3 == 0){//name
49                 name = list.get(i);
50                 System.out.println("name은 " + name);
51             }
52             else if(i % 3 == 1){//lat
53                 lat_lon = list.get(i);
54                 System.out.println("lat은 " + lat_lon);
55             }
56             else if(i % 3 == 2){//lon
57                 lat_lon = lat_lon + " " + list.get(i);
58                 adapter.addItem(name, lat_lon);
59                 System.out.println("lat_lon은 " + lat_lon);
60             }
61         }
62     }
63 }

```



```

63 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() { //클릭시 클립보드에 해당 즐겨찾기의 주소 저장.
64     @Override
65     public void onItemClick(AdapterView parent, View v, int position, long id) {
66         ListViewItem item = (ListViewItem) parent.getItemAtPosition(position);
67         lat_lon = item.getLat_lon();
68         // get item
69         tmp = lat_lon.split( regex: " ");
70         double latitude = Double.parseDouble(tmp[0]);
71         double longitude = Double.parseDouble(tmp[1]);
72
73         try {
74             geocoder = new Geocoder(getApplicationContext(), Locale.getDefault());
75             addresses = geocoder.getFromLocation(
76                 latitude,
77                 longitude,
78                 maxResults: 7);
79         } catch (IOException e){
80             e.printStackTrace();
81         }
82         Address address = addresses.get(0);
83
84         ClipboardManager clipboard = (ClipboardManager) getSystemService(Context.CLIPBOARD_SERVICE);
85         ClipData clipData = ClipData.newPlainText( label: "label", address.getAddressLine( index: 0).toString());
86         clipboard.setPrimaryClip(clipData);
87         Toast.makeText(getApplicationContext(), text: "클립보드에 복사되었습니다.", Toast.LENGTH_SHORT).show();
88         // TODO : use item data.
89     }
90 });
91
92 listView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() { //롱클릭시 해당 즐겨찾기 삭제.
93     @Override
94     public boolean onItemLongClick(AdapterView<?> parent, View view,
95         int position, long id) {
96         PlaceDatabaseManager databaseManager = PlaceDatabaseManager.getInstance(getApplicationContext());
97         ListViewItem item = (ListViewItem) parent.getItemAtPosition(position);
98         String delete_name[] = {item.getName()};
99         databaseManager.delete( whereClause: "NAME = ?", delete_name);
100
101         adapter.notifyDataSetChanged();
102         Toast.makeText(getApplicationContext(), text: item.getName() + " 이(가) 즐겨찾기에서 삭제되었습니다.", Toast.LENGTH_SHORT).show();
103         // 이벤트 처리 종료 , 여기만 리스너 적용시키고 싶으면 true , 아니면 false
104         return true;
105     }
106 });
107
108
109
110 }
111 }
112

```



SplashActivity.java(시작 로딩 화면 구성)

```
1 package com.example.myproject;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6
7 //어플리케이션 처음 시작화면
8 public class SplashActivity extends Activity {
9     protected void onCreate(Bundle savedInstanceState){
10         super.onCreate(savedInstanceState);
11
12         try{
13             Thread.sleep( millis: 3000);
14         }catch (InterruptedException e){
15             e.printStackTrace();
16         }
17         startActivity(new Intent( packageContext: this, MapsActivity.class));
18         finish();
19     }
20 }
21
```



4. 부연 설명 및 실행 화면

```

Address
[addressLines=[0:"대한민국 서울특별시 회현동 힐튼호텔"],
  feature=힐튼호텔,
  admin=서울특별시,
  sub-admin=null,
  locality=서울특별시,
  thoroughfare=null,
  postalCode=null,
  countryCode=KR,
  countryName=대한민국,
  hasLatitude=true,
  latitude=37.556407,
  hasLongitude=true,
  longitude=126.976636,
  phone=null,
  url=null,
  extras=null]

```

Address 안의 addressLines, latitude, longitude가 해당 지점의 주소, 위도, 경도이고, 이 정보를 통해 중간 지점 구하기(연산), 주변 음식점 추천(위도와 경도 구글맵 연동, 앱 내에서 음식점 뿐만 아니라 호텔, 카페 등 추천), 데이터베이스 저장을 통한 즐겨 찾기 등의 기능 제공.

Address는 1) 직접 검색하거나, 2) gps센서를 통한 현재 위치 정보 수집을 통해 얻을 수 있음.





그림 1 어플리케이션 아이콘

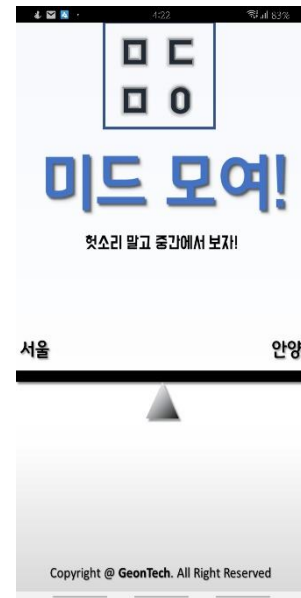


그림 2 어플리케이션 로딩 화면



그림 4 장소 선택 도우미

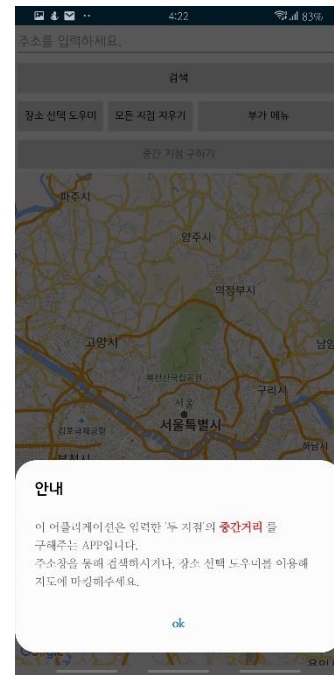


그림 3 어플리케이션 시작 화면





그림 5 검색 화면

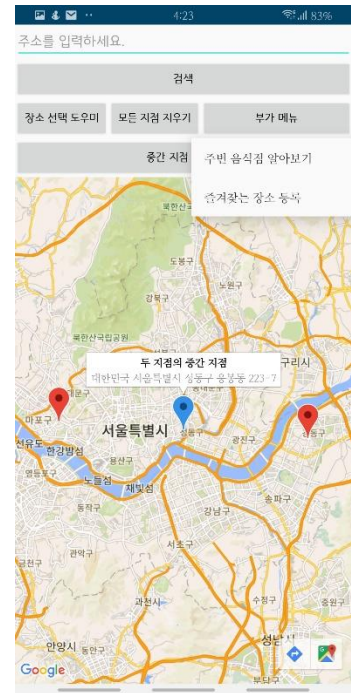
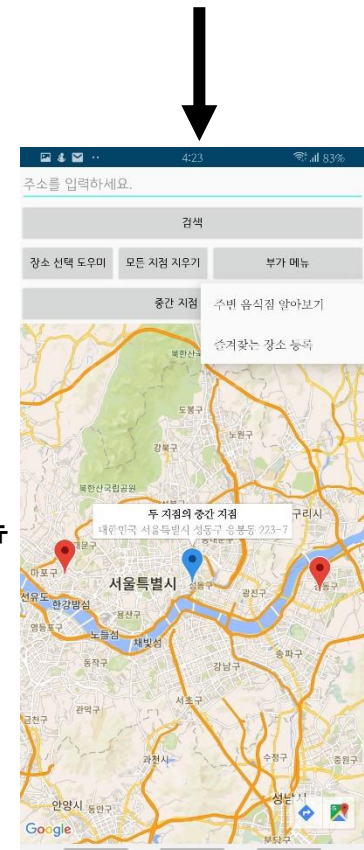


그림 6 중간 지점 계산



그림 8 음식점 추천(구글맵 연동)

그림 7
부가 메뉴

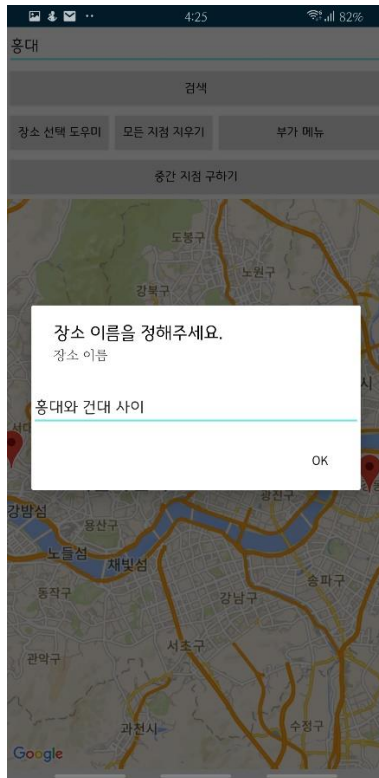


그림 9 내부DB(sqlite) 저장.

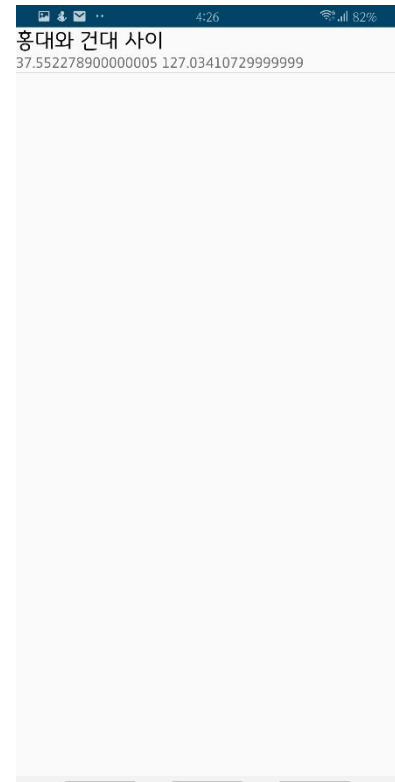


그림 10 내부DB 불러오기.

