

# 오픈 소스

국제대학 모바일시스템공학과

32132579 김성진

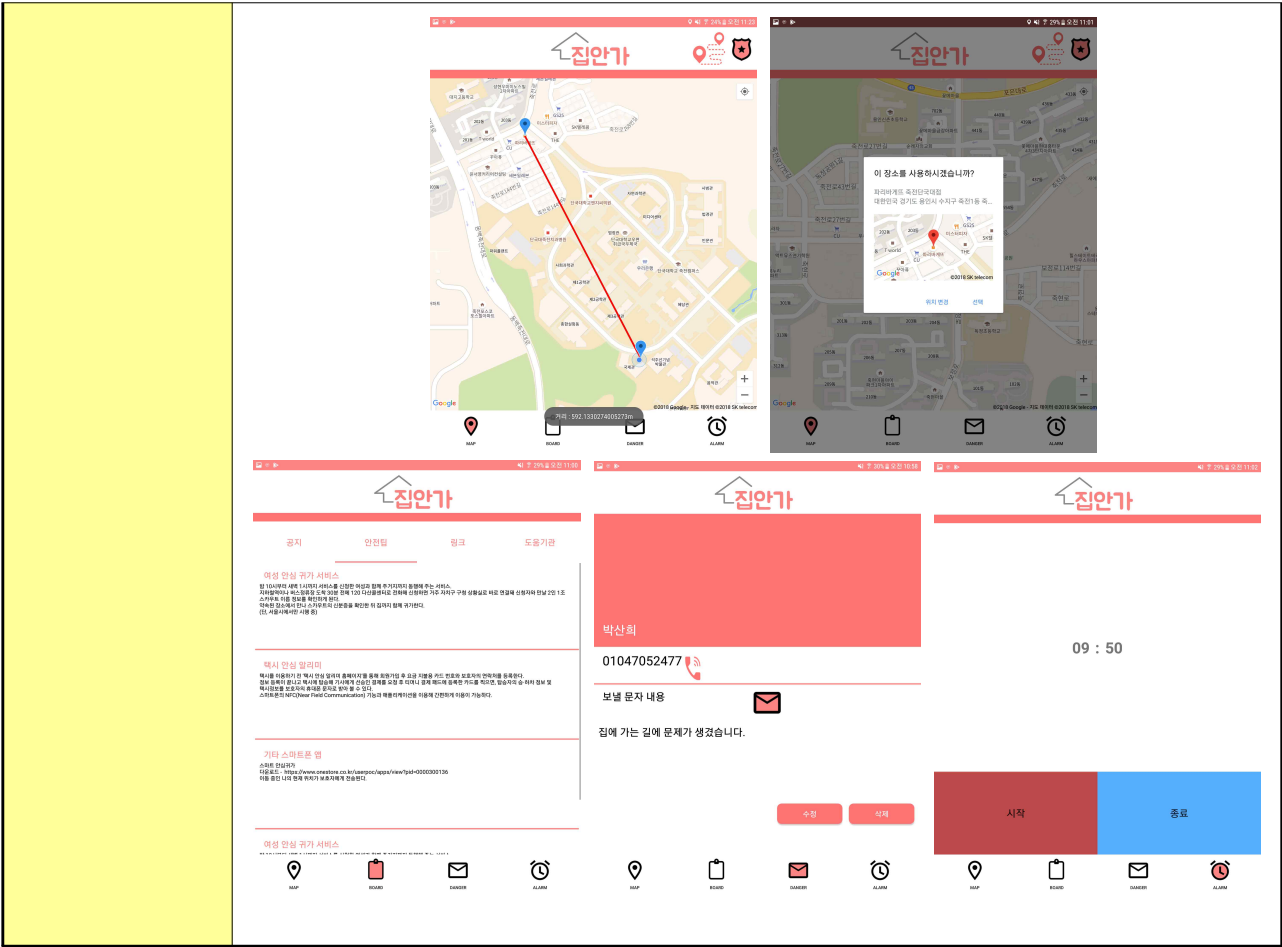
32143153 이대훈

32161620 박산희

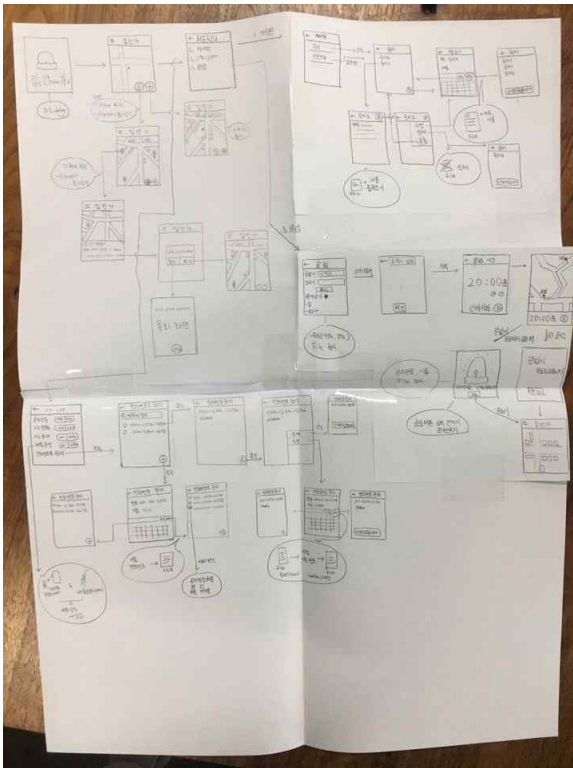
32164816 표지영

# 1. 프로젝트 요약문(연구목표 및 최종결과물)

구 분	내 용
<p>과제개요</p>	<p>최근 사회문제로 부상한 귀갓길 범죄를 예방하는 데 도움이 되는 각종 정보를 제공하고, 긴급 상황 발생 시에 사용자가 즉각 대처를 할 수 있도록 하는 기능들을 제공하는 스마트폰 어플리케이션을 개발함. 사용자의 현재 위치로부터 가까운 경찰서 검색과 목적지까지의 거리와 시간 계산, 타이머 설정을 하고 버튼기능으로 위급상황에서 보낼 수 있는 긴급문자 전송과 전화걸기 기능을 구현함.</p> <p>“집안가: 집에 안전하게 가자“</p> <p>9/18.</p>
<p>최종 목표</p>	<p>어플리케이션 개발결과물로 다음과 같은 기능을 사용자에게 제공함.</p> <ul style="list-style-type: none"> <li>- GPS를 사용한 지도상의 현재 위치 확인과 근처 경찰서 및 지구대 위치 표시 (클릭 시 세부정보 확인: 주소, 전화번호)</li> <li>- 지도에서 주변의 원하는 장소를 선택하여 직선거리(m) 자동 계산, 1m당 1초로 소요시간을 계산해 타이머 설정</li> <li>- 타이머가 설정된 시간 내에 종료가 눌러지지 않을 시 알람 울림</li> <li>- 전화번호부 관리 (DB) - 이름, 번호 저장, 메시지내용</li> <li>- 저장된 연락처로 전화 걸기, 메시지 전송</li> <li>- 게시판의 공지, 안전팁, 관련기사, 도움기관 정보 확인</li> </ul>



## 2. 과제 세부 설명



### 1. 메인 화면: 지도

어플 시작 후 첫 화면에 지도를 불러오고 내 위치를 표시함. 버튼들을 통해 지도 상에서의 기능들을 실행할 수 있도록 함.

#### ▶ 내위치

어플 다운 후 맨 처음 실행 시에 GPS로 기기의 현재 위치를 불러올 수 있도록 설정(허용)함.

#### ▶ 가까운 경찰서 및 지구대

현재 위치 기준 가까운 경찰서들의 위치 마크를 한 번에 띄우기

마크를 클릭하여 주소, 전화번호 표시

#### ▶ 길찾기

목적지의 이름을 검색하거나 마크 설정된 위치를 클릭하여 목적지 선택.

내위치로부터 직선 거리(m)로 계산함

### 2. 메뉴: 기타 기능

4가지 대표 기능을 4가지 아이콘으로 표시해 각각의 페이지로 구성

#### A. 게시판

##### ▶ 공지 / 안전팁 / 관련기사 / 도움기관

범죄 발생의 주요 대상인 여성을 위한 여성안심귀가 서비스, 여성긴급전화, 성폭력상담 및 지원센터에 대한 정보를 우선적으로 제공

범죄 관련 최근 기사와 다른 사용 가능한 어플리케이션에 대한 정보 제공

## B. 전화번호부관리

- ▶ 데이터베이스에 연락처 내용(이름, 전화번호) 저장
- ▶ 연락 기능: 전화 걸기, 문자 보내기
- ▶ 버튼 동작: 기기의 음량버튼을 3초간 누르면 문자 전송

## C. 알람

- ▶ 타이머 화면 구성

분, 초 (00:00) 표시, 시작 / 종료 버튼

- ▶ 타이머 시간 설정

지도-길찾기-직선거리에서 거리에 따른 소요시간 산출

소요시간(1m당 1초)이 분과 초로 환산되어 화면에 표시

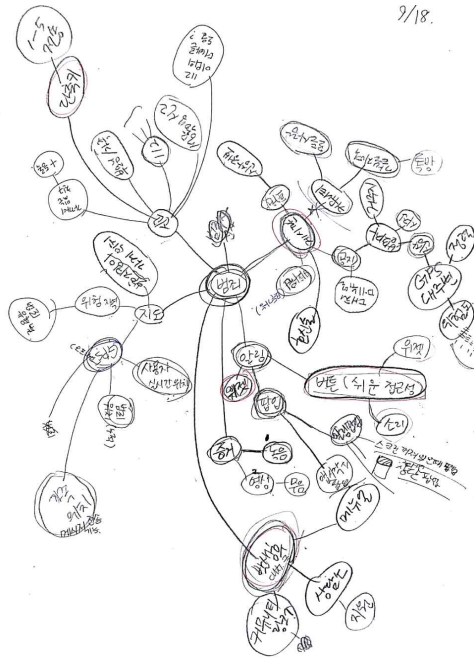
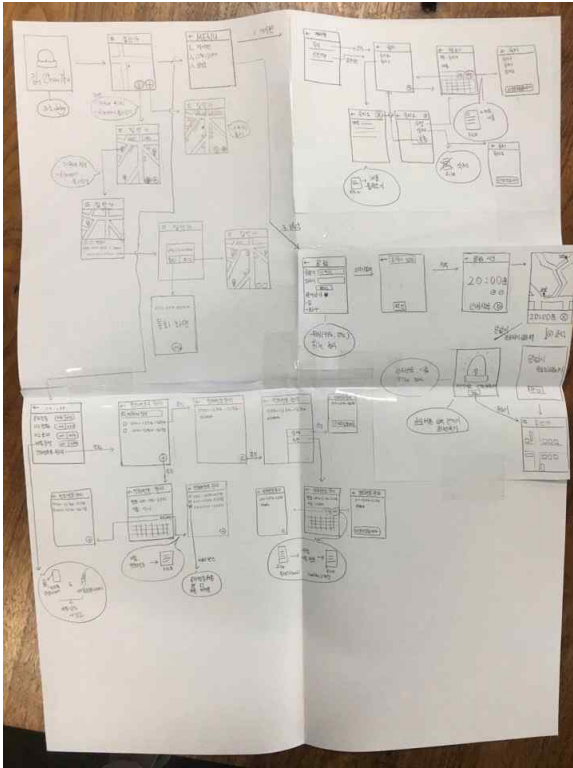
시작 버튼 누르면 카운트 시작되고 시간 내에 종료버튼이 눌리지 않을 시에 알람

## D. 지구대

(메인 화면인 지도로 넘어가는 것으로 대체)

### 3. 과제 기능별 설명

#### 1) 전체 클래스 다이어그램



1. 처음 로고 화면이 실행
2. 지도 화면이 가장 먼저 실행 이후 아래의 버튼 4개로 큰 기능을 구분 (지도, 게시판, 문자 전송, 알람)
3. 지도에서 주변 경찰서 위치 검색 및 목적지에 대한 검색
4. 게시판에 데이터 베이스를 사용하여 정보들 기입 (공지, 안전 팁, 기사, 도움 기관)
5. 문자 전송 문자 내용 설정 및 전화번호부 등록으로 문자 갈 사람들 지정
6. 서비스 작업을 통해 음량 조절 버튼 3초간 유지시 전화번호부에 기록된 번호로 문자 전송
7. 지도에서 계산한 현재위치와 목적지간의 거리계산을 통해 알람의 시간이 자동 설정
8. 알람이 시작되고 종료를 시키지 않은 상태로 알람이 끝난다면 전화번호부에 있는 사람들에게 문자 전송
9. 다른 화면에서도 알람이 계속 유지

## 2) 세부 클래스 분류 및 진행상황

[illegible]

알람 시간 설정	계획									100
	진도									
게시판 내용 구성	계획									링크 연결 x 80
	진도									
다이얼로그	계획									100
	진도									
전화 걸기	계획									100
	진도									
문자전송	계획									100
	진도									
아이콘 및 디자인 통합	계획									100
	진도									



### 3) 세부 클래스 참조 문서

<중복되는 속성 및 메서드는 다시 설명하지 않음>

#### 1. act1\_sh

설명 : 문자 전송 내용 설정 및 전화번호부 관리

속성

이름	설명
check_opt1	checkbox에서 check를 감지하는데 사용되는 변수
check_opt2	
check_opt3	
check_opt4	
txt_opt1	각 checkbox에 해당하는 text 값
txt_opt2	
txt_opt3	
btn_cancel_sms	문자내용을 저장하지 않을 때 사용하는 버튼
btn_store_sms	문자내용을 정하고 저장할 때 사용하는 버튼
editTextSMS	문자내용 직접 입력할 때 입력받는 edittext
store_sms_text	각 checkbox를 선택했을 때 값이 저장되는 변수
check4	직접입력한 값인지 확인
helper	DB table 객체
db	SQLiteDatabase 인스턴스 받아오는 변수
sql	DB 쿼리문 update 으로 입력
args	DB에 넣을 array 작성
nextIntent	다음에 올 Activity 입력
imgbtnmap	map 이미지 버튼, map Activity로 넘어감
btnmap	map 버튼, map Activity로 넘어감
imgbtnboard	게시판 이미지 버튼, board Activity로 넘어감
btnboard	게시판 버튼, board Activity로 넘어감
imagealarm	알람 이미지 버튼, alarm Activity로 넘어감
btalarm	알람 버튼, alarm Activity로 넘어감
mapintent	MapsActivity 입력
boardintent	board_notice 입력
alarmintent	Main_dh 입력

메서드

이름	설명
findViewById	layout에 설정된 view들을 가져옴.
setChecked	다른 checkbox 체크 해제
Toast.makeText	토스트 메시지를 띄움
arrayOf	배열 생성
execSQL	넣을 값을 넘겨주어 table을 update한다.
startActivity	다른 Activity로 이동
overridePendingTransition	애니메이션 없애기 위해 사용함.
setOnClickListener	버튼 클릭 이벤트를 처리

example

```
var check_opt1 = findViewById<CheckBox>(R.id.check_opt1)
check_opt3.setChecked(false);
Toast.makeText(applicationContext, "update " + store_sms_text, Toast.LENGTH_LONG).show()
db.execSQL(sql,args)
startActivity(nextIntent)
```

```
overridePendingTransition(0, 0)
btn_store_sms.setOnClickListener
```

## 2. act2\_1\_sh

설명 : 전화번호부 연락처 추가 및 목록 확인

속성

이름	설명
imgbtn_back	뒤로가기 이미지 버튼, 전 Activity로 돌아감
backintent	이전의 Activity 입력
btn_add_sh	연락처 추가 버튼
idx	DB phone table 에서 select 할 때 index
nameData	DB phone table 에서 select 할 때 이름 data
numberData	DB phone table 에서 select 할 때 번호 data
relateData	DB phone table 에서 select 할 때 관계 data
c	select 쿼리를 실행할 때 결과를 받을 cursor
txt_show_name	DB로부터 받아서 화면에 띄울 이름
txt_show_num	DB로부터 받아서 화면에 띄울 번호
idx_pos	해당 index값을 반환
nameData_pos	해당 index값에 들어있는 name data 반환
numberData_pos	해당 index값에 들어있는 number data 반환
relateData_pos	해당 index값에 들어있는 관계 data 반환
relate_var	DB로부터 받아서 화면에 띄울 관계
friend	친구 버튼 생성할 layout
button_friend	DB로부터 받아서 받은 개수만큼 친구 버튼 생성
next	intent 전환 시 같이 보낼 data
family	가족 버튼 생성할 layout
button_family	DB로부터 받아서 받은 개수만큼 가족 버튼 생성
acquaintance	지인 버튼 생성할 layout
button_acq	DB로부터 받아서 받은 개수만큼 지인 버튼 생성
relative	친척 버튼 생성할 layout
button_rel	DB로부터 받아서 받은 개수만큼 친척 버튼 생성

메서드

이름	설명
DBphone	DB phone table
writableDatabase	SQLiteDatabase 인스턴스를 받아오게 하는 메소드
rawQuery	SELECT 구문을 직접 실행
moveToNext	다음 레코드로 커서를 이동
getColumnIndex	특정 필드의 인덱스값을 반환
getInt	Int 형으로 값을 받음
getString	String 형으로 값을 받음
LayoutParams	코드에서 레이아웃 파라미터 설정
getDrawable	코드에서 button 생성 시 drawable 설정
putExtra	Activity간 데이터 전송
addView	layout에 버튼 띄우기
setPadding	코드에서 button 에 padding 값 주기
close	DB 닫기

example

```
db.rawQuery(sql, null)
c.getColumnIndex("idx")
```

```

c.getInt(idx_pos)
LinearLayout.LayoutParams(
    LinearLayout.LayoutParams.MATCH_PARENT,
    LinearLayout.LayoutParams.MATCH_PARENT
)
nextIntent.putExtra("phonenum", next as String)
db.close()

```

### 3. act2\_2\_sh

설명 : 전화번호부 내용 기입

속성

이름	설명
input_relate_store	spinner로부터 관계에 대한 값 저장
s	spinner 변수
intent	객체를 지정
getName	intent될 때 전달 받은 변수
getNum	
getRelate	
getIdx	
check	
input_name_store	edit text로 입력받은 이름 값
input_num_store	edit text로 입력받은 번호 값
btn_numcancel_sh	번호 저장하지 않을 때 버튼
btn_numstore_sh	번호 저장할 때 클릭하는 버튼
name	edit text로 입력받은 이름 값
number	edit text로 입력받은 번호 값
relate	spinner로부터 입력받은 관계 값
mod	DB에 저장할 array(update)
name_arg	DB에 저장할 array(insert)

메서드

이름	설명
OnItemSelectedListener	스피너 클릭했을 경우
onItemSelected	스피너 항목 클릭 했을 때
getItemAtPosition	아이템을 선택했을 때 선택한 값
onNothingSelected	아무것도 선택 안했을 때
getIntent	객체를 받아옴
getStringExtra	String형으로 변수를 받음
getIntExtra	Int형으로 변수를 받음
setText	textView에 text를 설정
getText	text를 받아옴

example

```

s.onItemSelectedListener
var intent = getIntent()

```

### 4. act3\_sh

설명 : 전화번호부 수정 및 삭제 가능, 통화 기능 및 보낼 문자 내용 확인

## 속성

이름	설명
textData	문자내용 DB로부터 받아서 저장
txt_text_send	textData 받아서 문자로 전송할 때 내용
getidx	index를 받아옴
btn_del_sh	번호 삭제 버튼
btn_modify_sh	번호 수정 버튼
go_idx	intent시 보낼 인자 값
builder	알림창 상속 받아 만든 클래스
sql2	DB로부터 삭제할 때 사용되는 쿼리문
remove	삭제하는 행
btn_call	전화 걸 때 사용하는 버튼
tel	전화번호 저장하는 변수
count	sms table 읽어올 때 index1만 읽도록 설정

## 메서드

이름	설명
AlertDialog.Builder	알림창 띄우기
layoutInflater.inflate	layout을 view의 형태로 반환
DBsms	DB sms table
setData	전화 걸 때 전환되는 intent

## example

```
val builder = AlertDialog.Builder(this)
val dialogView = layoutInflater.inflate(R.layout.dialog1, null)
```

## 5. board\_help

설명 : 도움기관에 대한 게시판

## 속성

이름	설명
txt_tip_title1	게시판 도움기관 항목 제목
txt_tip_title2	
txt_tip_title3	
txt_tip_title4	
txt_tip_title5	
txt_tip_title6	
txt_tip_con1	게시판 제목에 따른 내용
txt_tip_con2	
txt_tip_con3	
txt_tip_con4	
txt_tip_con5	
txt_tip_con6	
btn_board_tap1_sh	게시판 항목 공지 버튼
btn_board_tap2_sh	게시판 항목 안전팁 버튼
btn_board_tap3_sh	게시판 항목 기사 버튼
imgbtndanger	SMS 이미지 버튼, SMS activity로 넘어감
btndanger	SMS 버튼, SMS activity로 넘어감
dangerintent	SMS activity 입력

## 메서드 : 중복

## 6. board\_link

설명 : 기사에 대한 게시판

속성

이름	설명
idx	board table에서 받아오는 인덱스
cateData	board table에서 받아오는 카테고리
titleData	board table에서 받아오는 제목
contentData	board table에서 받아오는 내용
dateData	board table에서 받아오는 날짜
txt_show_cate	DB에서 받은 카테고리 출력
txt_show_title	DB에서 받은 제목 출력
txt_show_content	DB에서 받은 내용 출력
idx_pos	DB 해당 index로부터 받아온 data
cateData_pos	
titleData_pos	
dateData_pos	
contentData_pos	
linearLayout	코드에서 버튼 넣는데 사용한 레이아웃 id
button	코드에서 넣는 버튼
btn_board_tap4_sh	게시판 항목 도움기관 버튼

메서드

이름	설명
setBackgroundColor	배경색 설정
getDrawable	drawable 파일을 받아옴

example

```
button.setBackgroundColor(0xFFFFFFFF)
```

```
button.background = ContextCompat.getDrawable(this, R.drawable.short_bottom)
```

## 7. board\_notice

설명 : 공지에 대한 게시판

속성

이름	설명
btn_notice_sh	공지 내용 버튼

메서드 : 중복

## 8. board\_tip

설명 : 안전 팁에 대한 게시판

속성

이름	설명
txt_tip_title1	안전 팁 제목
txt_tip_title2	
txt_tip_title3	
txt_tip_title4	
txt_tip_con1	안전 팁 내용
txt_tip_con2	
txt_tip_con3	
txt_tip_con4	

메서드 : 중복

#### 9. DBboard

설명 : 게시판 데이터 베이스 테이블 생성

속성

이름	설명
sql	쿼리문 작성

메서드

이름	설명
execSQL	테이블과 색인 등 생성
onUpgrade	SQLiteDatabase 인스턴스와 스키마 버전 받음

#### 10. DBphone

설명 : 전화번호 데이터 베이스 테이블 생성

속성

이름	설명
sql	쿼리문 작성

메서드

이름	설명
execSQL	테이블과 색인 등 생성
onUpgrade	SQLiteDatabase 인스턴스와 스키마 버전 받음

#### 11. DBsms

설명 : 문자내용 데이터 베이스 테이블 생성

속성

이름	설명
sql	쿼리문 작성

메서드

이름	설명
execSQL	테이블과 색인 등 생성
onUpgrade	SQLiteDatabase 인스턴스와 스키마 버전 받음

## 12. detail\_text

설명 : 게시판 데이터 베이스로부터 가져온 내용 표출

속성

이름	설명
txt_title_detail_sh	게시글 타이틀
txt_content_detail_sh	게시글 내용
txt_date	게시글 작성 날짜

메서드 : 중복

## 13. detail\_text2

설명 : 게시판 데이터 베이스로부터 가져온 내용 표출

속성

이름	설명
txt_title_detail_sh	게시글 타이틀
txt_content_detail_sh	게시글 내용
txt_date	게시글 작성 날짜

메서드 : 중복

## 14. dh4

설명 : 알람이 종료되었을 때 문자 전송 및 화면에 문자 전송 표현

속성

이름	설명
btn4_dh	알람 종료 버튼

메서드 : 중복

## 15. First\_logo

설명 : App을 처음 실행했을 때 처음 2초가 Logo가 나온 후 다음 activity로 이동

속성

이름	설명
start	permission이 끝났는지에 대한 확인
FHandler	handler 선언
SHandler	
time	logo화면을 몇 초간 유지할지에 대한 변수
permissionCheck	각각의 시스템 권한에 대한 체크

메서드

이름	설명
FirstHandler	permission체크가 완료될 때까지 계속 수행
SecondHandler	permission체크가 완료된 후 logo 화면 수행 후 다음 Activity로 넘어감
sendEmptyMessage	handler에 메시지를 전달
checkSelfPermission	시스템 접근 권한 처리
requestPermissions	런타임에 권한 요청
sendEmptyMessageDelayed	설정 delay를 가지고 지속적으로 handler를 호출
removeMessages	handler를 종료

example

```
FHandler = FirstHandler()
```

```
FHandler?.sendEmptyMessage(0)
```

```
val permissionCheck =
```

```
ContextCompat.checkSelfPermission(this@First_logo,android.Manifest.permission.ACCESS_FINE_LOCATION)
```

```
ActivityCompat.requestPermissions(
    this@First_logo,
    arrayOf(
        Manifest.permission.SEND_SMS,
        Manifest.permission.CALL_PHONE,
        android.Manifest.permission.ACCESS_FINE_LOCATION
    ),
    1
)
```

```
SHandler?.sendEmptyMessageDelayed(0,1000)
```

```
FHandler?.removeMessages(0)
```

## 16. Main\_dh

설명 : 알람 기능

속성



이름	설명
dHandler	handler 선언
pHandler	
time	현재위치에서 목적지까지 걸리는 시간
tick	pHandler의 조건을 위한 변수
get_distance	Mapsactivity에서 거리 값을 받아옴
txt_time	알람 시간을 나타내는 textview
btn_start	알람 시작 버튼
btn_end	알람 종료 버튼
img_clock	알람 진행시 imageview
showdg	dialogview를 저장
btn3_dh	알람을 종료시켰을 때 dialog의 확인 버튼
hour	알람의 시간
minute	알람의 분
second	알람의 초
smsManager	데이터, 텍스트 및 SMS 메시지 전송과 같은 SMS 작업을 관리

#### 메서드

이름	설명
getDoubleExtra	Double 형으로 값을 반환
toInt	int 형으로 변환
DisplayHandler	남은 시간을 보여주는 handler
ImageHandler	알람이 실행 중 이미지를 계속 바꿔주는 handler
padStart	두자리 숫자로 표현 없는 경우 '00'으로 표시
setImageResource	imageview를 다른 drawable 파일로 변환

#### example

```
var get_distance = intent.getDoubleExtra("distance_map",0.0)
txt_time.text = minute.toString().padStart(2, '0')
img_clock.setImageResource(R.drawable.walk1)
```

#### 17. MainService

설명 : 서비스로 키 이벤트를 사용하여 볼륨 키를 3초 이상 유지시 문자 전송, Background에서도 활용 가능

#### 속성

이름	설명
mediaSession	미디어 버튼과의 상호 작용을 허용하는 변수
flag	아래 버튼키 감지시 true
handler	handler 선언
time	1초당 1씩 증가함
myVolumeProvider	볼륨 버튼

#### 메서드

이름	설명
MediaSessionCompat	미디어 버튼 및 전송 컨트롤과의 상호 작용을 허용
setFlags	flag 설정
setPlaybackState	현재 상태 업데이트
VolumeProviderCompat	세션의 볼륨 조정 또는 조정 요청을 처리
onAdjustVolume	볼륨 제어에 대한 액세스
postDelayed	주어진 시간만큼 delay 시킴
setPlaybackToRemote	볼륨 버튼 이벤트를 수신
onBind	사용 안함
onDestroy	서비스 종료

example

```
mediaSession!!.setFlags(MediaSessionCompat.FLAG_HANDLES_MEDIA_BUTTONS
MediaSessionCompat.FLAG_HANDLES_TRANSPORT_CONTROLS)
```

or

## 18. MapsActivity

설명 : 지도 화면에서 내 위치 표시, 목적지 찾기 및 현재 위치와의 거리 계산, 주변 경찰서 안내  
속성

이름	설명
mMap	GoogleMap 객체를 받아 옴
fusedLocationClient	위치정보 또는 위치 업데이트 요청
lastLocation	app을 마지막으로 사용했을 때의 위치 저장
currentLatLng	현재 위치 저장
next_dis	현재 위치와 목적지 사이의 거리를 저장
locationCallback	장치 위치가 변경되었거나 더 이상 확인할 수 없는 경우 FusedLocationProviderApi에서 알림 수신
locationRequest	적당한 간격으로 변경된 사용자 위치를 저장
locationUpdateState	현재 위치를 update할지를 설정
police1	경찰서 위도 경도 설정
police2	
police3	
police4	
btn_dh7	목적지 검색 버튼
btn_dh8	주변 경찰서 표시 버튼
a	스위치가 켜져 있을 경우, a를 0으로 설정하여 경찰서 위치에 마커를 띄움. 스위치가 꺼져 있을 경우, a를 1으로 설정하여 모든 마커를 없앴.
pref	getSharedPreferences()함수를 사용하여 만들어진 파일을 저장하는 변수
editor	경찰서위치 표시에 대한 switch의 상태를 유지하기 위해 SharedPreferences에 이 값을 저장
sw	SharedPreferences에서 switch의 상태를 읽어 옴
dialogswitch	레이아웃을 다이얼로그창으로 변환
isFirst	SharedPreferences에서 값을 읽어 옴

이름	설명
arg_notice1	게시판 공지 항목 내용 입력
arg_notice2	
arg_notice3	
arg_notice4	
arg_notice5	
arg_notice6	
arg_notice7	
arg_notice8	
arg_notice9	
arg_notice10	
arg_notice11	
arg_notice12	
arg_link1	게시판 기사 항목 내용 입력
arg_link2	
arg_link3	
arg_link4	
arg_link5	
arg_link6	
arg_link7	
builder_location	알림창 상속 받아 만든 클래스
policeLocation	경찰서 위치 알림창
dialogCancel	dialog 취소 버튼
dialogCall	dialog 전화 버튼
display1	dialog 화면 입력
size1	다이얼로그 창 사이즈
window1	다이얼로그 창

이름	설명
x1	다이얼로그 창의 가로 길이
y1	다이얼로그 창의 세로 길이
tel	전화번호 입력
client	사용자 위치를 설정
task	지도에 사용자 위치 표시
place	목적지 저장
addressText	목적지 주소
distance1	현재 위치와 목적지간의 거리
REQUEST_CHECK_SETTINGS	request 요청에 대한 확인 처리
PLACE_PICKER_REQUEST	place_picker 요청에 대한 확인 처리
markerOptions	마커의 여러 속성(위치, 이름 등)들을 이용할 수 있게 하는 변수
titleStr	marker의 위치를 string으로 저장
geocoder	Geocoder api 사용
addresses	geocoder로 받은 목적지의 주소
address	주소 값 저장함

## 메서드

이름	설명
getMapAsync	google map api를 받아옴
loadPlacePicker	google place api를 사용하여 목적지 검색
getFusedLocationProviderClient	각종 지역 정보를 받아옴

이름	설명
setOnCheckedChangeListener	버튼 클릭 이벤트를 처리함
setPositiveButton	“확인”에 대해 응답하는 버튼
addMarker	지도에 마커 표시
moveCamera	화면을 위치로 이동시킴
animateCamera	화면이 이동할 때 animation 효과 발생
putInt	int 형으로 값을 입력
commit	editor를 저장함
clear	map 초기화
setNegativeButton	“취소”에 대해 응답하는 버튼
show	다이얼로그 창을 띄어 줌
createLocationRequest	실시간 위치를 특정 시간 간격으로 요청하고 받아 옴
CheckAppFirstExecute2	설치 후 맨 처음 한번만 실행함
getSharedPreferences	파일 형식으로 값들을 저장하여 앱을 삭제하지 않는 이상 값들을 유지
getBoolean	boolean 형태로 받음
onInfoWindowClick	마커를 클릭 시 나타나는 정보창을 클릭했을 때 실행
getWindowManager	화면의 dp를 구하기 위해 사용
Point	두 정수 좌표의 값
getSize	size 값을 받음
getWindow	화면 창을 가져옴
setData	저장한 값을 입력함
startLocationUpdates	내 위치 업데이트
placeMarkerOnMap	지도에 마커 표시
addPolyline	두 지점 사이에 직선으로 표시
SphericalUtil.computeDistanceBetween	두 지점 사이의 거리 구하기
removeLocationUpdates	현재 위치 업데이트 정지
setUpMap	현재 위치를 업데이트
LatLng	위도, 경도 표시
getFromLocation	위도, 경도로 실제 주소 변환 함

example

```
var distance1 = SphericalUtil.computeDistanceBetween(currentLatLng, place.latLng)
mMap.addPolyline(PolylineOptions().add(currentLatLng).add(place.latLng).width(8f).color(Color.RED))
mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(LatLng(37.3097,127.1094), 13.8f))
mMap.moveCamera(CameraUpdateFactory.newLatLng(LatLng(37.3097,127.1094)))
mMap.addMarker(MarkerOptions().position(police2).title("용인서부경찰서 수지지구대").snippet("주소 및 전화번호 보기"))
```

#### 19. sh\_notice1

설명 : 게시판 공지 이미지 파일




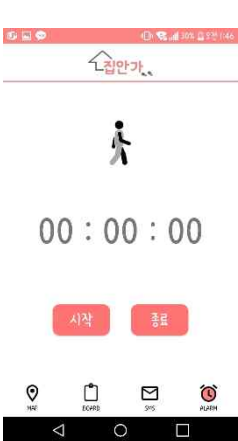
속성

이름	설명
imgbtn_back	뒤로가기 버튼
backintent	뒤로가기 intent

메서드 : 중복

## 4. 프로그램 구성

### 1) 개요

① map	② board	③ sms
 <ul style="list-style-type: none"> <li>- 실시간 내 위치 알 수 있음.</li> <li>- 단국대학교 주변 경찰서 위치 및 정보를 알 수 있음.</li> <li>- 도착지를 설정하여 현재 내 위치에서 도착지까지의 일직선 거리를 표시할 수 있음.</li> </ul>	 <ul style="list-style-type: none"> <li>- 공지: 어플 사용 방법 및 업데이트 정보 알 수 있음.</li> <li>- 안전팁: 집에 안전하게 들어가도록 도와주는 여러 서비스들을 설명.</li> <li>- 기사: 범죄관련 기사들 확인 가능.</li> <li>- 도움기관: 범죄 발생 후, 상담 등의 서비스 지원하는 기관 설명.</li> </ul>	 <ul style="list-style-type: none"> <li>- 전화번호부 관리: 연락처 추가 및 수정, 삭제/ 관계(친구, 가족 등)에 따른 연락처 분류 가능.</li> <li>- 볼륨 키 3초 이상 길게 눌리거나 alarm(④번)의 시간이 모두 경과한 경우, 보낼 문자 선택 가능. (수정된 사항, 전화번호부 관리에서 확인 가능.)</li> </ul>
④ alarm		
	<ul style="list-style-type: none"> <li>- map(①번)에서 계산한 거리 값(내 위치 ~ 도착지)을 받아 옴.</li> <li>- 시작 누를 시, 1m당 1초로 계산하여 받아 온 거리 값에 맞게 카운트다운 시작</li> <li>- 카운트다운이 끝나기 전에 집에 도착한 경우, 종료버튼을 누르면 알람 종료됨.</li> <li>- 집 도착 전, 카운트다운이 끝난 경우, 저장한 전화번호로 문자 전송. (③번에서 연락처 및 문자내용 설정.)</li> </ul>	

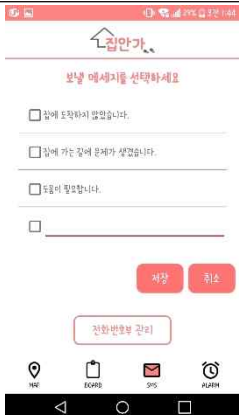
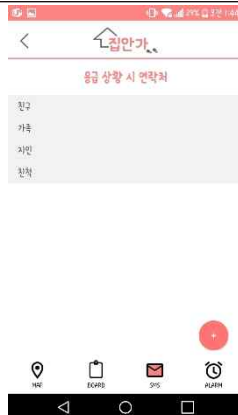




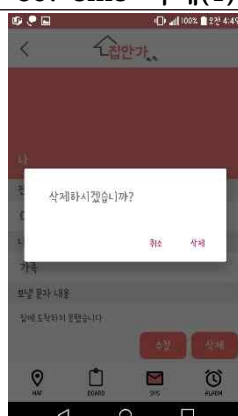

## 2) 사용 방법



1. SMS 전송 permission	2. 전화 발신 permission	3. 위치 서비스 permission
 <p>application 설치 시, 문자 전송에 대한 permission.</p>	 <p>문자전송에 관한 permission 이후, 통화에 대한 permission.</p>	 <p>마지막으로, 위치 정보 사용을 위한 permission.</p>
4. 로딩 화면	5. 첫 화면-map	6. map-내 위치 버튼
 <p>Application 로딩 화면</p>	 <p>application실행 시, 첫 화면에 지도 및 현재 내 위치가 뜸.</p>	 <p>오른쪽 위 버튼/ 실시간 내 위치가 뜨며 이 버튼을 계속 누르지 않아도 자동으로 업데이트됨.</p>
7. map-경찰서 버튼	8. map-경찰서 스위치	9. map-경찰서 위치
 <p>왼쪽 위 방패 모양 버튼/ 주변 경찰서 및 지구대 위치를 보여주게 하는 다이얼로그 창이 뜸.</p>	 <p>스위치를 켜. (경찰서 및 지구대의 마커를 사라지게 하고 싶을 경우, 스위치 off)</p>	 <p>단국대학교 주변 경찰서 및 지구대 위치가 마커로 표시됨.</p>

<p><b>10. map-마커 클릭</b></p>  <p>마커 클릭 시, 경찰서에 대한 정보(이름) 볼 수 있음.</p>	<p><b>11. map-정보창 클릭</b></p>  <p>정보창 클릭 시, 경찰서 및 지구대에 대한 정보(이름, 주소, 전화번호) 확인 가능.</p>	<p><b>12. map-경찰서로의 전화</b></p>  <p>전화 버튼 클릭하면 해당 경찰서 및 지구대 전화번호로 전화 가능.</p>
<p><b>13. map-네비게이션 버튼</b></p>  <p>지도 화면에서 왼쪽 위 버튼 클릭 시, 도착지를 설정할 수 있는 화면으로 넘어감.</p>	<p><b>14. map-도착지 설정</b></p>  <p>도착지 검색. 지도를 움직여서 도착지를 설정할 수도 있음.</p>	<p><b>15. map-도착지 선택</b></p>  <p>위치 선택 또는 변경.</p>
<p><b>16. map-일직선 거리</b></p>  <p>현재 내 위치에서 도착지(15번에서 설정)까지의 일직선 거리(m)가 계산됨.</p>	<p><b>17. alarm(1)</b></p>  <p>alarm버튼 클릭 시, 16번에서 계산된 거리 값을 가져옴.</p>	<p><b>18. alarm(2)</b></p>  <p>시작 버튼 클릭하면 도착지까지 걸릴 시간(1m당 1초)을 계산하고 카운트다운 시작.</p>

<div data-bbox="177 190 467 226" data-label="Section-Header"> <h2>19. alarm-종료 버튼</h2> </div> <div data-bbox="231 228 469 645" data-label="Image"> </div> <div data-bbox="162 676 537 795" data-label="Text"> <p>종료 버튼을 누르면 알람이 종료되었다는 문구와 함께 다이얼로그 창이 뜬다.</p> </div>	<div data-bbox="569 190 935 226" data-label="Section-Header"> <h2>20. alarm-모든 시간 경과</h2> </div> <div data-bbox="632 228 871 645" data-label="Image"> </div> <div data-bbox="569 656 935 817" data-label="Text"> <p>종료버튼을 누르기 전에 시간이 모두 지난 경우, 저장한 전화번호(전화번호 저장방법 29번부터 설명)로 문자가 감.</p> </div>	<div data-bbox="1023 190 1286 226" data-label="Section-Header"> <h2>21. 게시판-공지(1)</h2> </div> <div data-bbox="1031 228 1275 645" data-label="Image"> </div> <div data-bbox="965 676 1345 795" data-label="Text"> <p>공지: 업데이트, 사용 방법 등 여러 공지사항을 확인할 수 있음.</p> </div>
<div data-bbox="221 817 478 853" data-label="Section-Header"> <h2>22. 게시판-공지(2)</h2> </div> <div data-bbox="231 855 469 1272" data-label="Image"> </div> <div data-bbox="183 1323 513 1359" data-label="Text"> <p>필독: 어플 사용 방법 설명</p> </div>	<div data-bbox="624 817 873 853" data-label="Section-Header"> <h2>23. 게시판-공지(3)</h2> </div> <div data-bbox="632 855 871 1272" data-label="Image"> </div> <div data-bbox="569 1283 935 1359" data-label="Text"> <p>업데이트 클릭 시, 업데이트된 내용들 확인 가능.</p> </div>	<div data-bbox="1023 817 1286 853" data-label="Section-Header"> <h2>24. 게시판-안전팁</h2> </div> <div data-bbox="1031 855 1275 1272" data-label="Image"> </div> <div data-bbox="965 1283 1345 1400" data-label="Text"> <p>안전팁: 안전 귀가를 위해 사용할 수 있는 여러 서비스들 소개.</p> </div>
<div data-bbox="221 1400 478 1435" data-label="Section-Header"> <h2>25. 게시판-기사(1)</h2> </div> <div data-bbox="231 1438 469 1854" data-label="Image"> </div> <div data-bbox="162 1910 537 1946" data-label="Text"> <p>기사: 범죄관련 기사 확인가능.</p> </div>	<div data-bbox="624 1400 873 1435" data-label="Section-Header"> <h2>26. 게시판-기사(2)</h2> </div> <div data-bbox="632 1438 871 1854" data-label="Image"> </div> <div data-bbox="569 1890 935 1966" data-label="Text"> <p>기사제목 클릭하면 기사 내용 확인가능.</p> </div>	<div data-bbox="1023 1400 1286 1435" data-label="Section-Header"> <h2>27. 게시판-도움기관</h2> </div> <div data-bbox="1031 1438 1275 1854" data-label="Image"> </div> <div data-bbox="965 1868 1345 1986" data-label="Text"> <p>도움기관: 범죄 발생 후, 상담 등 서비스 제공하는 센터 확인 가능</p> </div>



<p><b>28. sms</b></p>  <p>보낼 메시지 및 연락처 관리.</p>	<p><b>29. sms-전화번호 관리</b></p>  <p>저장된 연락처가 친구, 가족, 지인, 친척으로 분류됨.</p>	<p><b>30. sms-전화번호 추가(1)</b></p>  <p>'+'버튼 클릭 시, 연락처를 추가할 수 있고 관계를 설정할 수 있음.</p>
<p><b>31. sms-전화번호 추가(2)</b></p>  <p>선택한 관계에 맞게 연락처가 분류되어 저장됨.</p>	<p><b>32. sms-연락처 클릭</b></p>  <p>연락처를 클릭하면 저장한 연락처에 대한 정보를 볼 수 있으며 전화 버튼을 누르면 전화 가능.</p>	<p><b>33. sms-수정(1)</b></p>  <p>연락처에 대한 정보 수정 가능.</p>
<p><b>34. sms-수정(2)</b></p>  <p>수정된 내용 반영하여 저장됨.</p>	<p><b>35. sms-삭제(1)</b></p>  <p>연락처 삭제 가능.</p>	<p><b>36. sms-삭제(2)</b></p> 

37. sms-문자내용 변경(1)	38. sms-문자내용 변경(2)	39. sms-문자내용 변경(3)
 <p>볼륨 키 3초이상 누르거나 alarm 시간이 모두 지났을 경우, 보낼 문자 설정. (기본으로 '집에 도착하지 않았습니다.' 설정되어 있음.)</p>	 <p>전화번호부 관리 클릭하여 연락처 클릭하면 문자내용이 변경된 것 확인 가능.</p>	 <p>문자 내용 직접 입력 가능.</p>
40. sms-문자내용 변경(4)		
 <p>변경된 문자 내용 확인 가능.</p>		

## 5. 필수 적용 사항

### 1. 서비스

사용자가 또 다른 애플리케이션으로 전환하더라도 백그라운드에서 계속해서 실행됨.

사용자 인터페이스를 제공하지 않음.

- activity가 종료된 상태에서도 동작하기 위해서 만들어짐.
- manifest에서 service로 선언해주어야 함.
- intent 로 시작함.
- startService(), bindService()를 이용할 수 있고, 이번 어플에서는 startService() 사용함.
  1. startService() : 시작
    - 시작된 서비스는 한 작업을 수행하고 결과를 호출자에게 반환하지 않음.
    - 한 번 시작되고 나면 백그라운드에서 무기한으로 실행
  2. bindService() : 바인드
    - 바인드된 서비스는 클라이언트-서버 인터페이스를 제공하여 구성 요소가 서비스와 상호작용 (서비스가 서버 역할).
      - 결과를 가져올 수도 있음(액티비티는 서비스에 어떠한 요청을 할 수 있고, 서비스로부터 어떠한 결과를 받을 수 있음).
      - 이와 같은 작업을 여러 프로세스에 걸쳐 프로세스 간 통신(IPC)으로 수행할 수도 있음.
- 볼륨 아래 버튼 키 이벤트를 감지하는데 사용 함.

### 2. 문자전송

어플에 권한을 주어서 사용자가 권한 허용을 해야 함.

- smsManager class 사용
- 데이터, 텍스트 및 SMS 메시지 전송과 같은 SMS 작업을 관리함.
- getDefault ()
  - : 기본 구독 ID와 연결된 SmsManager를 가져옴.
  - : 정적 메소드를 호출하여 이 객체를 가져옴.
- sendtextMessage

(String destinationAddress, String scAddress, String text,

PendingIntent sentIntent, PendingIntent deliveryIntent)

: parameter

1. phone number : 메시지 보낼 번호
2. scAddress : 현재 기본 SMSC를 사용하는 서비스 센터 주소 또는 null.
3. String text : 보낼 문구
4. sentIntent : NULL이 아닌 경우 PendingIntent는 메시지가 성공적으로 전송되거나 실패 할 때 브로드 캐스팅됨.
5. deliveryIntent : NULL이 아닌 경우 PendingIntent는 메시지가 수신자에게 전달 될 때 브로드 캐스팅됨.

### 3. 액티비티

- Activity는 일종의 애플리케이션 구성 요소.
- 사용자가 전화 걸기, 사진 찍기, 이메일 보내기 또는 지도 보기 등의 일을 하기 위해 상호작용할 수 있는 화면을 제공함.
- 액티비티 마다 창이 하나씩 주어져 액티비티에 사용자 인터페이스를 끌어올 수 있음.
- 액티비티는 stack 구조로 쌓이게 됨(LIFO 구조).
  - : manifest에서 no history 옵션을 주면 stack에 쌓이지 않음.
- 액티비티를 생성하려면 Activity의 서브클래스(또는 이의 기존 서브클래스)를 생성함.
- onCreate()
  - : 시스템은 액티비티를 생성할 때 onCreate()을 호출
  - : setContentView()를 호출해서 xml과 연결해 주어야 액티비티의 사용자 인터페이스 레이아웃을 정의할 수 있음.

### 4. intent + 다음 activity로 값 넘겨주기 put extra

- 메시지 객체.
- 앱 구성요소(액티비티, 서비스, 콘텐츠 제공자, 브로드캐스트 수신기)로부터 작업을 요청할 수 있음.
- 사용은 크게 3가지로 나뉨.
  1. 액티비티 시작
    - : 액티비티의 새 인스턴스를 시작하려면 intent를 startActivity()로 전달한다.
    - : 이때 putExtra 로 인자 값 전달이 가능하고, getDoubleExtra로 인자 값을 전달 받을 수 있음 (key 값으로 구별).
  2. 서비스 시작
    - : 서비스를 시작 할 때 Intent를 startService()에 전달함.
    - : 서비스가 클라이언트-서버 인터페이스로 디자인된 경우, 다른 구성 요소로부터 서비스에 바인드하려면 Intent를 bindService()에 전달함.
  3. 브로드캐스트 전달
    - : 브로드캐스트는 모든 앱이 수신할 수 있는 메시지.
    - : 시스템은 여러 시스템 이벤트에 대해 다양한 브로드캐스트를 전달함.

## 5. SQLite Database

- SQLiteDatabase에는 SQL 명령을 작성, 삭제, 실행하고 기타 일반적인 데이터베이스 관리 작업을 수행하는 메소드가 존재함.
- 데이터베이스 이름은 응용 프로그램 내에서 고유해야 함.

### - onCreate() 메소드

: SQLiteDatabase를 넘겨 받으며, 테이블을 생성하고 초기 데이터를 추가할 수 있음.

### - onUpgrade() 메소드

: SQLiteDatabase 인스턴스를 넘겨 받으며, 현재 스키마 버전과 최신 스키마 버전 번호도 받음.

### - SQLiteOpenHelper

: 먼저 인스턴스를 하나 생성.

: 하려는 작업이 읽기 전용인지의 여부에 따라

getReadableDatabase()나 getWritableDatabase() 메소드를 호출.

### [테이블 준비]

- sql 작성 : table 생성, column 항목 작성.
- 테이블과 색인 등을 생성하려면 원하는 DDL 구문을 준비해서 SQLiteDatabase 인스턴스의 execSQL() 메소드를 호출함.

### [데이터 추가, insert]

- getWritableDatabase() 메소드를 통해 권한 받아옴.
- 대상이 되는 테이블 이름, column 별 들어갈 data를 넣어줌.

### [데이터 업데이트, update]

- getWritableDatabase() 메소드를 통해 권한 받아옴.
- 값을 변경할 대상을 한정 지으려면 WHERE 구문과 함께 WHERE 조건에 해당하는 값 역시 넘겨줌.
- WHERE 구문에 물음표로 지정된 위치에 각자의 값이 배치돼 처리된다.
- 대상 테이블 이름과 변경할 값이 들어 있는 ContentValues 객체를 넘겨줌.

### [데이터 삭제, delete]

- delete() 메소드 역시 테이블 이름과 WHERE 구문을 사용.
- 변경할 값을 지정하지 않는다는 점만 제외하면 update()와 동일한 방법으로 동작함.

### [데이터 불러오기, select]

- .rawQuery() 메소드에 SELECT 구문을 넘겨 줌.
- 결과로 받아오는 값은 Cursor 인스턴스.
- Cursor를 사용하면 여러 건의 결과를 하나씩 받아오면서 처리할 수 있음.

## 6. 전화 걸기

전화 연결을 위한 permission 필요. 사용자가 권한을 허용해야 함.

- Manifest에 permission을 세팅.
- xml파일에 있는 전화번호를 tel이라는 변수에 string으로 저장

- intent를 이용한 전화 걸기
  - : intent는 application들 간에 정보를 전달하고 호출하는 역할
  - : startActivity()함수를 이용해서 호출됨.
  - : 전화걸기 application을 호출하기 위해 ACTION\_CALL이라는 intent 객체를 사용.
- Uri.parse(uriString): 인코딩된 uriString을 구문 분석하는 URI 객체를 만듦.
  - : 이 application에서는 전화번호가 uriString 부분에 들어감.
- URI : 자원 식별자라는 뜻으로 하위개념으로 URL이 있음.

## 7. 다이얼로그

- 레이아웃 위에 띄우는 창.
  - 보통 프로그램 종료나 메시지를 알릴 때 간단하게 쓸 수 있는 기본 창.
  - AlertDialog.Builder()함수를 통해 builder 객체 생성.
- 이 객체의 메소드들을 호출해서 속성을 지정할 수 있음.
- 예) 확인 버튼(setPositiveButton("확인"){ ... }) 클릭 또는  
취소 버튼(setNegativeButton("취소"){ ... }) 클릭 시 코드 실행
- LayoutInflater : 레이아웃 XML파일을 View객체로 만들기 위해 사용.
    - : 첫 번째 parameter : 레이아웃 XML파일
    - : 두 번째 parameter : root를 null로 설정하여 android:layout\_xxxx 값들을 무시.

## 8. 지도

### 1) api 키

- application에서 Google지도 서버에 액세스하기 위해서 API키 필요.
- api key 받는 방법: Google Maps Activity로 파일을 생성 후, res/values/google\_maps\_api.xml.파일에서 api 키를 얻을 수 있는 인터넷 주소를 얻을 수 있음. 키 생성 후 키 값을 이파일의 YOUR\_KEY\_HERE에 입력.

### 2) build.gradle파일에서 필요한 dependency 추가

- Google 지도 및 위치 서비스 API를 application에 공개하기 위해 Google Play Services dependency 추가(implementation 'com.google.android.gms:play-services-maps:16.0.0').
- 위치 서비스를 이용하기 위해서 Google Maps location dependency 추가(implementation 'com.google.android.gms:play-services-location:11.8.0').

### 3) 구글맵 객체 불러 오기 : onMapReady 함수

### 4) 위치서비스에 대한 permission

- 사용자에게 대한 상세한 위치를 얻기 위해 AndroidManifest.xml.파일에 ACCESS\_FINE\_LOCATION permission 추가.
- MapsActivity.kt에서 위치 permission에 대한 코드가 담긴 setUpMap()함수 사용.

### 5) 현재 위치 받아오기

- Google Play Services 위치 APIs에서 사용자의 장치의 마지막 위치를 요청하여 사용 가능.
- setUpMap()함수에서 사용자의 가장 최근 위치를 가져오고 이 위치로 카메라를 옮겨가게 함(지도에 파

란색 점으로 표시됨).

- 위치 업데이트: `createLocationRequest()` 함수에서 업데이트를 할 간격을 지정하여 위치를 요청.

#### 6) 마커 및 정보창

- `setOnMarkerClickListener()`, `setOnWindowClickListener()`를 통해 경찰서 및 지구대 정보를 보기 위해 사용.

#### 7) geocoding

- Geocoder api를 사용하여 위치의 실제 주소로 변형할 수 있음.
- 변형한 주소의 값을 string 형으로 받아와 지도에 마커를 표시함.
- 위의 google map에서 사용한 api 키를 사용하여 google console에서 api 제한사항에 Geocoding API를 입력함.
- google place api를 사용함.
- google console에서 api 제한사항에 google place API를 입력함.
- Geocoder를 통해 변형된 실제 주소를 통해 지도 내에서 검색함.
- 검색을 완료하면 `addmarker`를 사용하여 검색한 위치에 마커로 표시함.

### 9. 핸들러

- Handler를 사용하면 스레드의 `MessageQueue`와 관련된 `Message` 및 `Runnable` 객체를 보내고 처리할 수 있음.
- 각 Handler 인스턴스는 단일 스레드 및 해당 스레드의 메시지 큐와 연관됨.
- 새로운 Handler를 만들면, 그 핸들러를 생성하고 있는 스레드의 스레드 / 메시지 큐에 바인딩 됨.
- 그 시점부터 메시지와 실행 파일을 해당 메시지 큐에 전달하고 메시지에서 나오는 대로 실행함.
- 즉, 핸들러란 메인스레드와 서브스레드 간에 Handler를 통해 메시지를 전달하여 메시지 큐에 저장하는 방식의 통신임.

- `Handler.sendMessage(Message msg)`

: 핸들러의 메시지 전달 함수

: 메시지 큐는 FIFO방식.

\*프로젝트에서의 사용 : timer로 사용함.

- Handler를 사용함.
- 지도 Activity에서 거리를 구하고 그 값을 넘겨 받아옴.
- 거리 1m마다 1초로 계산함.
- 계산한 값들을 시간, 분, 초에 맞게 입력.
- 시작 버튼을 누르면 handler를 1초마다 호출하도록 설정함.
- 1초마다 신호를 받으면 구한 시간 값에서 1초씩 뺌.
- 시간이 0이 되었을 경우 문자를 전송하고 전송 완료 layout으로 넘어감.
- 종료 버튼을 누르면 handler를 종료시킴.
- 시간 위의 이미지 파일을 handler를 사용하여 변환해줌.
- 0.1초마다 이미지가 바뀌도록 함.

## 10. 미디어 버튼 사용

- `MediaSessionCompat` class 사용
  - : 미디어 컨트롤러, 볼륨 키, 미디어 버튼 및 전송 컨트롤과의 상호 작용을 허용함.
  - : 앱이 미디어 재생 정보를 게시하거나 미디어 키를 처리하려고 할 때 `MediaSession`을 만들어야 함.
  - : 세션이 생성되면 세션 소유자는 세션 토큰을 다른 프로세스에 전달하여 `MediaControllerCompat`을 만들어 세션과 상호 작용할 수 있게 함.
- `MediaSessionCompat`
  - : 지정된 미디어 버튼 수신기 (구성 요소 이름 및 보류 중인 의도)를 사용하여 새 세션 만들.

## 11. 처음 실행 시 한 번만 코드 실행하도록 하기

- `import android.content.SharedPreferences`
- `SharedPreferences` 사용함.
- 간단한 값 저장에 DB를 사용하기에는 복잡하기 때문에 `SharedPreferences` 사용.
- 보통 초기 설정값이나 자동로그인 여부 등 간단한 값을 저장하기 위해 사용.
- 어플리케이션에 파일 형태로 데이터를 저장.
  - ex. `data/data/패키지명/shared_prefs/SharedPreferences이름.xml` 위치에 저장.
- 어플리케이션이 삭제되기 전까지 보존됨.
- 인스턴스 얻기
  1. `getPreferences(int mode)`
    - : 하나의 액티비티에서만 사용하는 `SharedPreferences`를 생성함.
  2. `getSharedPreferences(String name, int mode)`
    - : 특정 이름을 가진 `SharedPreferences`를 생성함.
    - : application 전체에서 사용.
- 저장할 수 있는 data type
  - : `Boolean`, `Integer`, `Float`, `Long`, `String`
- 함수를 실행하고, `SharedPreferences`을 이용하여 한번 실행했다는 값을 넣어줌.
- 어플 실행 시 미다 실행했다는 값을 읽어오면 더 이상 그 함수는 호출하지 않는 방식.



## 12. spinner

- res>values 에 xml파일 만들기.
- xml에 사용할 array 만들어서 저장.
- item 은 spinner에 들어갈 항목들을 나타 냄.
- kt 파일에서 onCreate에서 spinner를 초기화 시켜줌.
- arrayadapter를 만들어서 spinner와 연결함.
- setSelected를 사용하여 이벤트 리스너를 걸어줌.

## 13. 디자인

- background
  - : xml 파일 만들어서 사용
  - : png, jpg 파일 사용( 이미지 파일명에 대문자,괄호,- 등 특정문자 사용 불가능, 사용 시 R class오류 발생)
  - : RGB 사용하여 색상 넣기
- spinner
  - : res의 value 에 array로 항목 작성
  - : 항목을 주어 사용자가 선택하도록 함.
- checkbox
  - : 사용자가 원하는 항목 선택하도록 함.
- editText / textView
  - : text를 나타냄.
  - : editText 는 사용자가 입력할 수 있어서 입력값을 받고 kt 파일에서 사용 가능함.

## 14. kt파일에서 button 생성하기

- linear layout 에 id값을 줌.
- 해당하는 layout에 button을 생성함.
- button 생성 시 layoutParams를 사용하여 속성값을 설정함(padding, textsize, background 등).
- addview 메소드를 통해 해당 layout에 버튼을 추가함.

## 15. switch

getSharedPreferences(String name, int mode)함수를 이용

- 첫 번째 parameter에서 입력한 이름의 SharedPreferences 파일을 생성.
- 두 번째 parameter mode는 읽고 쓰기 권한과 관련.
  - 1) getInt : key와 디폴트값을 파라미터로 하여 호출하여 값을 읽음.
  - 2) putInt : editor 객체를 통해 원하는 값을 key, value 형태로 입력하고 commit()를 통하여 데이터 저장.
- getSharedPreferences를 이용하여 스위치가 켜져 있는 것을 유지하게 함  
(value값이 1이면 스위치가 켜져 있으며 0일 경우 스위치가 꺼져 있음).