

웃자

웃는 상황 자동 녹음 어플
20172634 이민희

Table of Contents

01



프로젝트 개요 / 목표

02



Use case / UML

03



pseudo code / 중요 부분

04



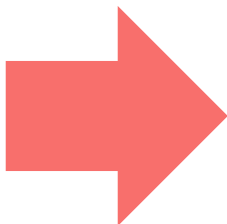
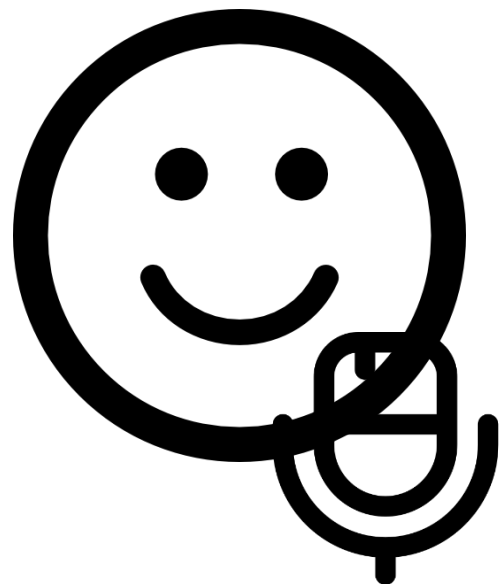
트러블 슈팅 / 결론

05



데모 / QnA

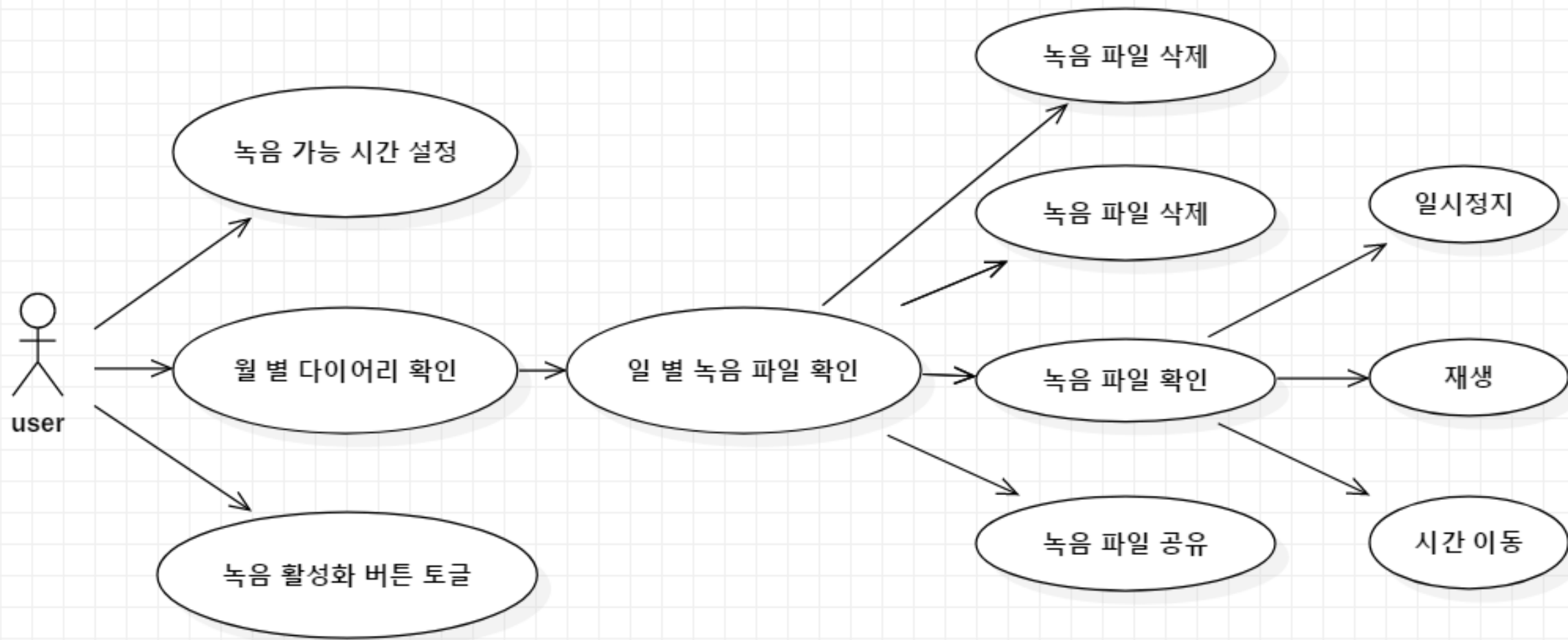
프로젝트 개요



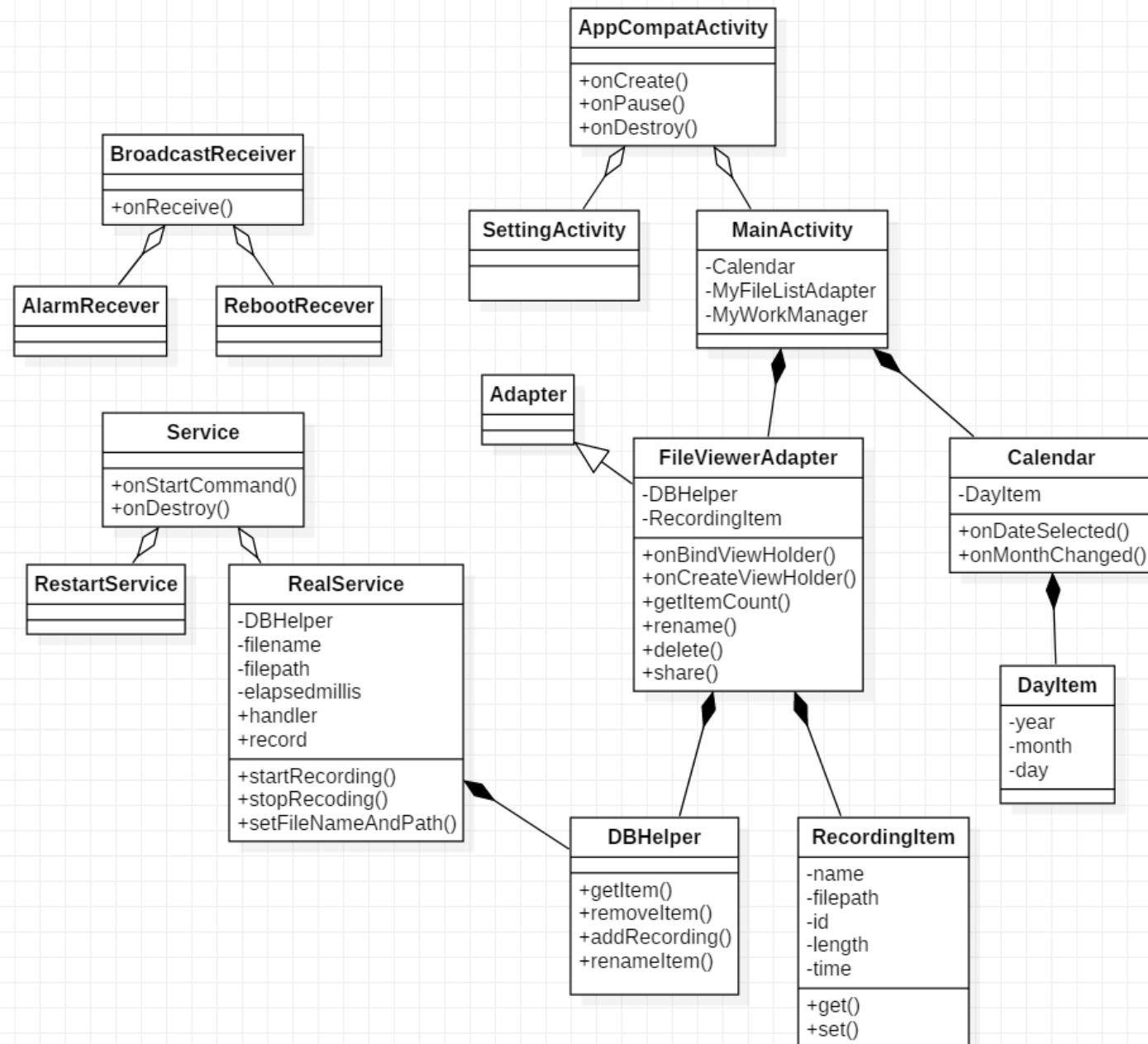
웃음 인식을 통해
웃는 상황 자동 녹음 어플

웃는 상황 손 쉽게 기록
작가, 방송인, 개그맨 등
아이디어 창출
하이라이트 추출

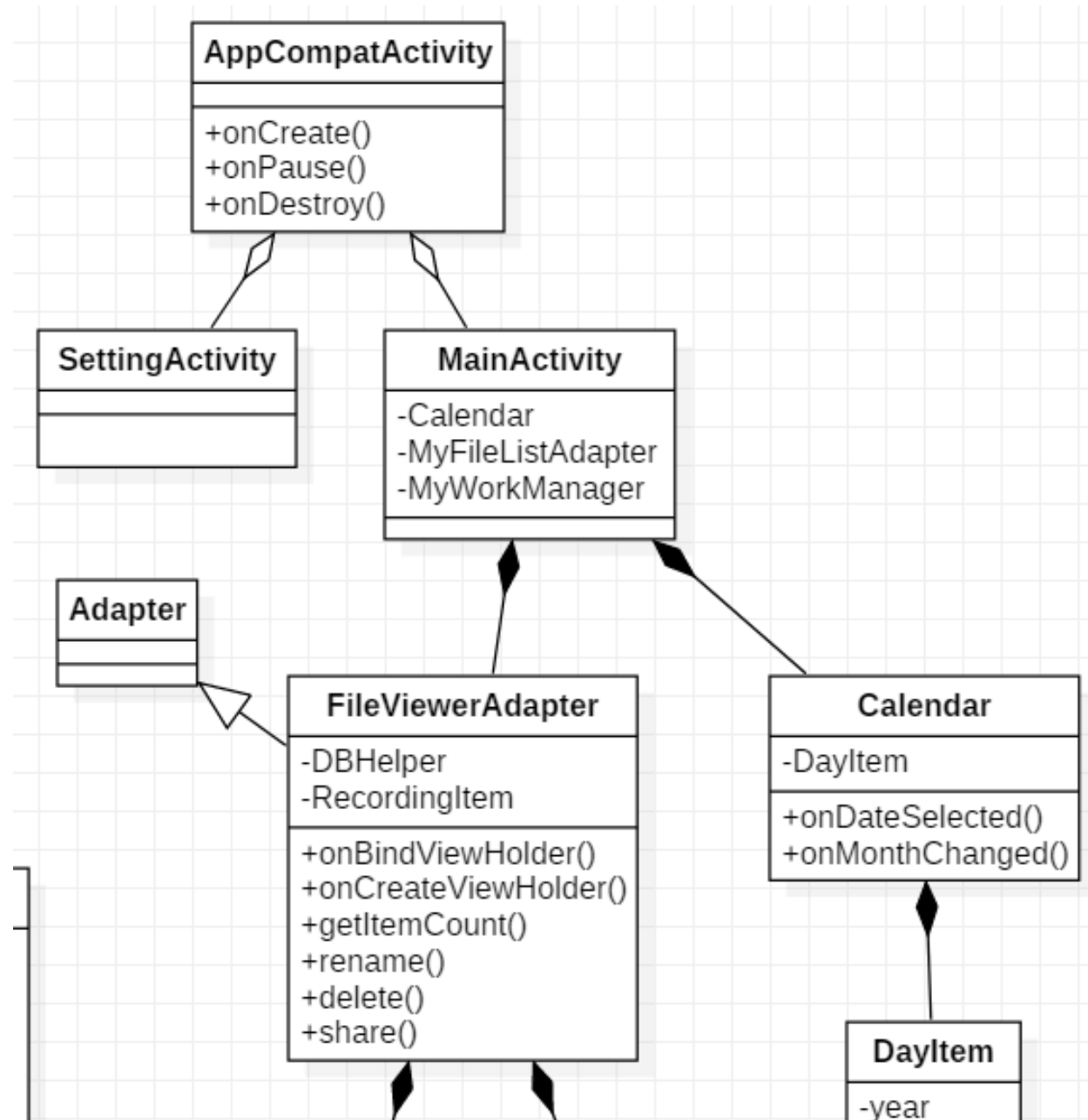
Use case



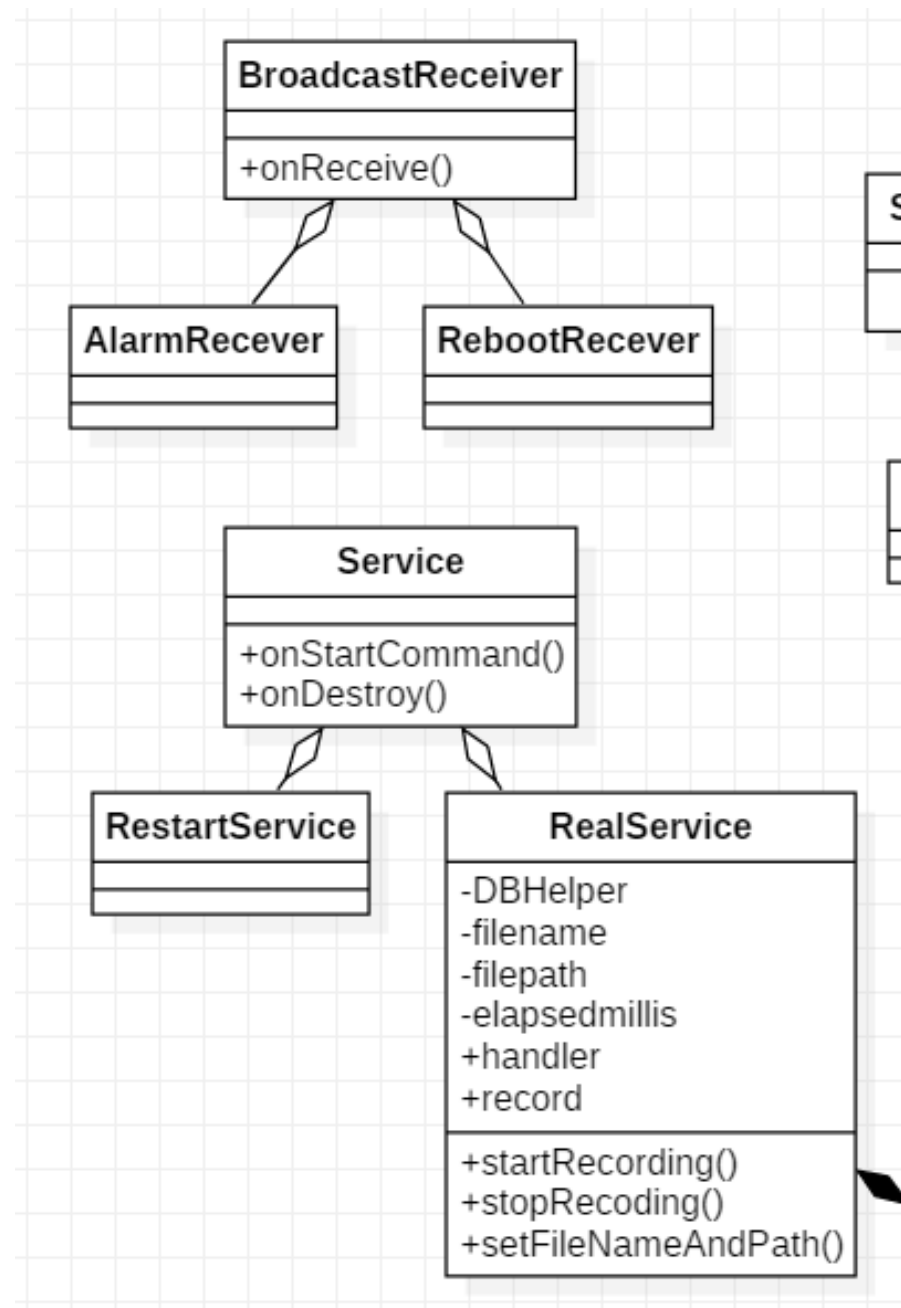
UML



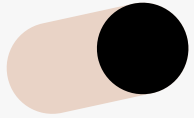
UML



UML



중요 기능



스위치로 쉽게 녹음 기능을 활성화 및 비활성화

앱이 실행 중이지 않더라도 계속 기능 진행

녹음된 파일을 리스트 뷰로 띄워서
재생, 이름변경, 삭제 등의 기능 제공



pseudo code

MainActivity.java : 어플리케이션 main에 나타날 화면

```
fun onCreate()  
    set SettingActivity  
    processing calendar  
    set switch  
    processing service  
    set recyclerView
```

```
fun onPause()  
    IF isSettingAct  
        //  
    ELSE  
        IF isActivate  
            startService()  
        ELSE  
            storeState
```

pseudo code

RealService.java : 백그라운드에서 계속 돌아갈 서비스

```
fun onStartCommand()
```

```
    WHILE TRUE
```

```
        RecordingTime1()
```

```
        RecordingTime2()
```

```
fun RecordingTime()
```

```
    curTime
```

```
    startRecording()
```

```
    WHILE TRUE
```

```
        IF now - curTime > 10
```

```
            break
```

```
    stopRecording()
```

트러블슈팅

데이터 셋 수집, 모델 구축 시간

Animals	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Birds (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw

1. 핸드폰이 자주 들을 수 있는 소리가 아닌, 특정 상황, 악기, 동물 등 접하기 쉽지 않은 데이터 셋이 많아 높은 정확도가 나오지 않을 것 같다는 생각
2. 앱 개발 시간을 고려하였을 때 데이터 셋 수집, 데이터 전/후처리, 모델 구축 등에 사용할 시간이 부족할 것이라 생각

➡ 모델 사용 부분을 제외하고 어플 기능 구현에 집중

트러블슈팅

WorkManager

일정 시간마다 해야 할 일을
순서대로 넣어 놓고
처리를 하려고 시도

할 일이라고 보내 놓고
뒤에 일은 신경 쓰지 않음

효율적인 리소스 사용이지
엄청난 사용량 개선 X



backgroundService

시도 하는 것이 아닌
일정 시간동안 꼭 일어나야 함

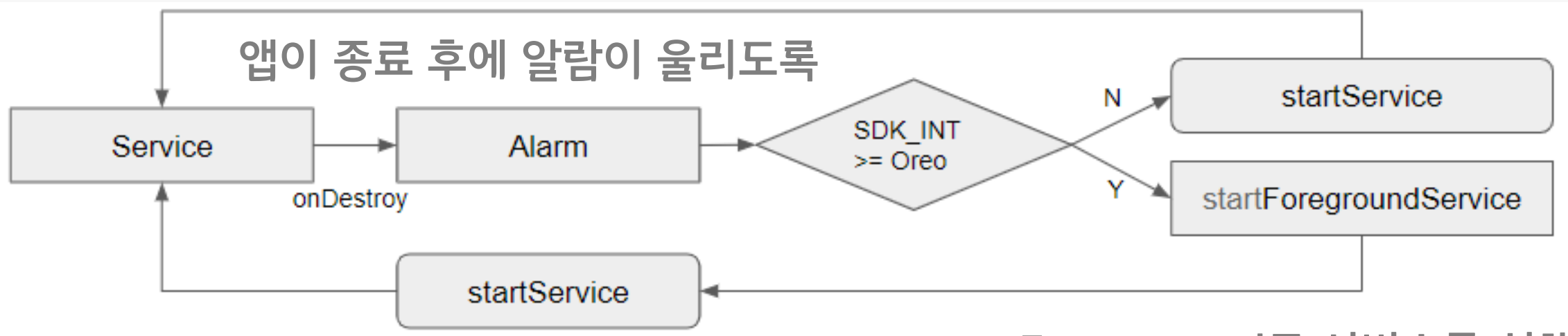
녹음을 시작하고 끝내는 것은
꼭 순서대로 일어나야 하므로
끝나는 시간을 인식하는 것이 중요

실제 메신저 앱들도
이런 방법으로 구현이 되어있음

트러블슈팅

최신 버전에서 이것을 금지

backgroundService



Foreground로 서비스를 실행 후
백 그라운드로 서비스를 실행

트러블슈팅

1. 프로젝트 이슈

앱을종료하거나 프로세스에서 앱을 죽이면 서비스가 재시작 되어 정상적으로 돌아가나

앱을 종료한후(BACK 키 눌러 finish) 프로세스에서 모두지우기 누르는 경우 서비스가 중단되고 다시 시작되지 않습니다. 해결책이 있을까요?

좋은 지적이십니다.

해보지 않았지만, 앱을 종료하면(`onDestroy`) 타이머를 이용해서 서비스를 실행하듯이,

백스페이스로 앱이 종료되면(`onBackPressed`) 타이머를 이용해서 서비스를 실행하면 될 것 같습니다.

제가 나중에 시간 될 때 코드를 작성해 보겠지만, 먼저 해보시고 코드를 공유해주셔도 좋을 것 같습니다.

트러블슈팅

2. 액티비티 생명주기

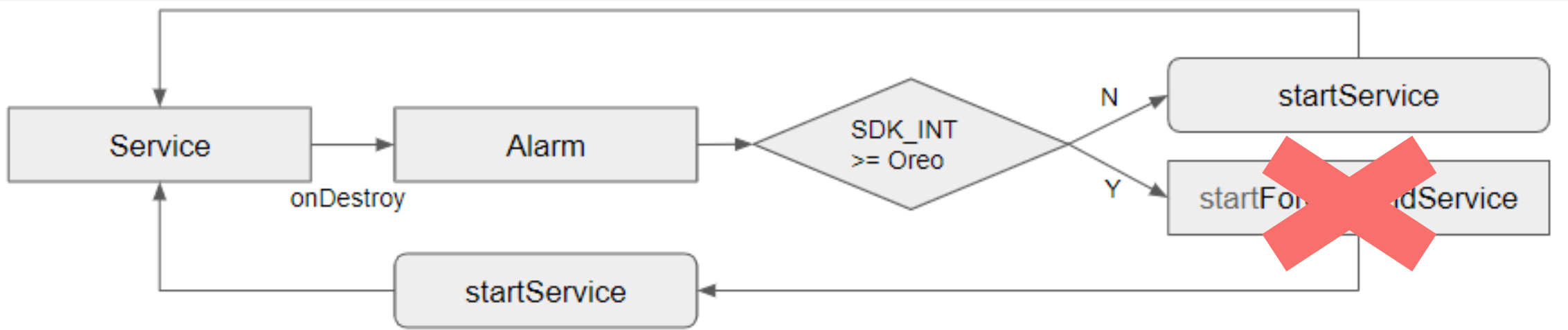


Mair
Activity

Setting
Activity

트러블슈팅

3. 서비스 종료 불가능



트러블슈팅

1. 3.

```
@Override
public void onBackPressed() {
    new AlertDialog.Builder( context: this)
        .setIcon(android.R.drawable.ic_dialog_alert)
        .setTitle("Closing Activity")
        .setMessage("Are you sure you want to close this activity?")
        .setPositiveButton( text: "Yes",
            {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    finish();
                }
            })
        .setNegativeButton( text: "No", listener: null)
        .show();
}
```

```
@Override
public void onBackPressed() {
    new AlertDialog.Builder( context: this)
        .setIcon(android.R.drawable.ic_dialog_alert)
        .setTitle("Closing Activity")
        .setMessage("Are you sure you want to close this activity?")
        .setPositiveButton( text: "Yes", new DialogInterface.OnClickListener()
        {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                android.os.Process.killProcess(android.os.Process.myPid());
            }
        })
        .setNegativeButton( text: "No", listener: null)
        .show();
}
```

트러블슈팅

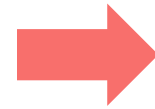
Toggle 기능 구현

```
@Override
protected void onStop() {
    super.onStop();

    if (isActive){
        if (RealService.serviceIntent==null) {
            serviceIntent = new Intent( packageContext: this, RealService.class);
            serviceIntent.putExtra( name: "isActive", isActive);
            startService(serviceIntent);
        } else {
            serviceIntent = RealService.serviceIntent;//getInstance().getApplication();
            Toast.makeText(getApplicationContext(), text: "already", Toast.LENGTH_LONG).show();
        }
    }else{
        SharedPreferences pref = getSharedPreferences( name: "pref", MODE_PRIVATE);
        SharedPreferences.Editor editor = pref.edit();
        editor.putBoolean( s: "isActive", isActive);
        editor.commit();
        android.os.Process.killProcess(android.os.Process.myPid());
    }
}
```

트러블 슈팅

onStop()



onPause()

```

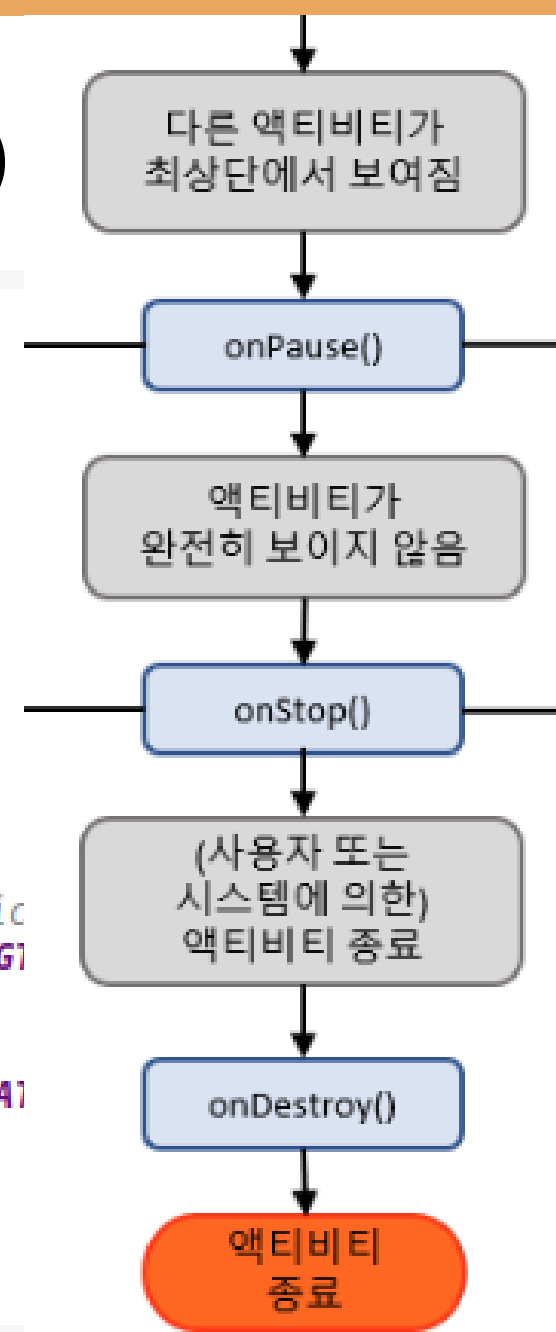
@Override
protected void onPause() {
    super.onPause();

    if (isSettingAct) {
        Log.d("나", "세팅페이지간거구나");
    } else {
        if (isActivate) {
            if (RealService.serviceIntent == null) {
                serviceIntent = new Intent(packageContext, this, RealService.class);
                serviceIntent.putExtra("isActive", isActivate);
                startService(serviceIntent);
            } else {
                serviceIntent = RealService.serviceIntent; // getInstance().getApplication
                Toast.makeText(getApplicationContext(), "already", Toast.LENGTH_SHORT);
            }
        } else {
            SharedPreferences pref = getSharedPreferences("pref", MODE_PRIVATE);
            SharedPreferences.Editor editor = pref.edit();
            editor.putBoolean("isActive", isActivate);
            editor.commit();
            android.os.Process.killProcess(android.os.Process.myPid());
        }
    }
}

```

SettingActivity에 가린 상태인가?

활성화 할 상태인가?



결론

현재 코드에는 모델 처리를 해서 웃음 인식이 되는 기능은 없지만

값 손실이 있는 mp4 확장자가 아닌
wav 확장자로 저장하도록 작성하였으므로
음성 인식 모델을 구축한다면 쉽게 적용시킬 수 있을 것이다

연결이 되어있지 않거나, 구현되지 않은 부분이 있으나

원하는 시간에 앱이 실행되지 않아도 지속적으로 녹음을 하는 앱은
지속성이나 경향성을 보는 연구나 데이터 수집에 도움이 될 것이다

데모

Q&A