

나만의 일기장

“세 칸 일기”

01

주제 선정 이유

02

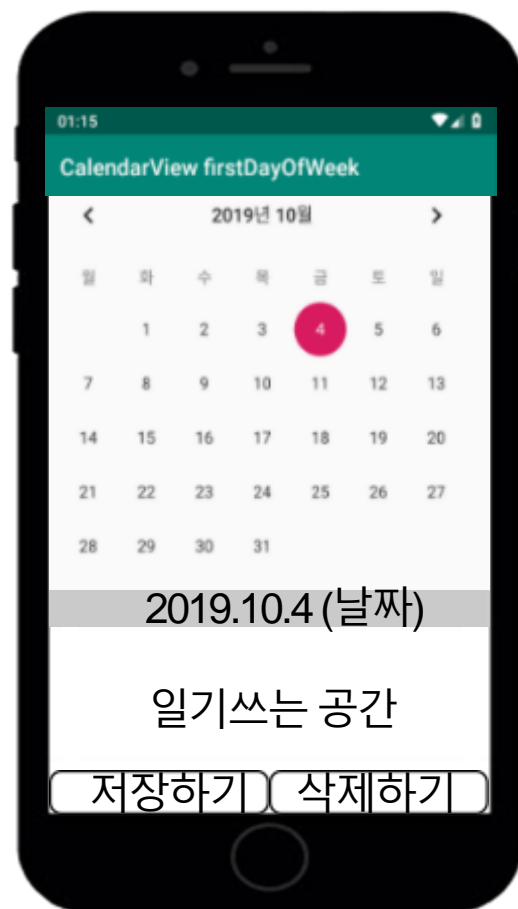
세 칸 일기 구성

03

세 칸 일기 기능

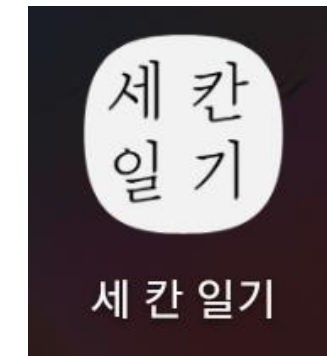
- **일기쓰는 스타일에 맞는 일기장 X**
- **일기쓰는 스타일**
 - 매일 쓰지 않음
 - 길게 쓰는 건 귀찮음
 - 간단하게 오늘 하루를 요약

초기 구상안



VS

세 칸 일기



- 쓰고 싶을 때 일기쓰기

- 일기를 쓴 날만 목록으로 보여줌

=> 일기를 다시 보려고 할 때 접근성 증가

01

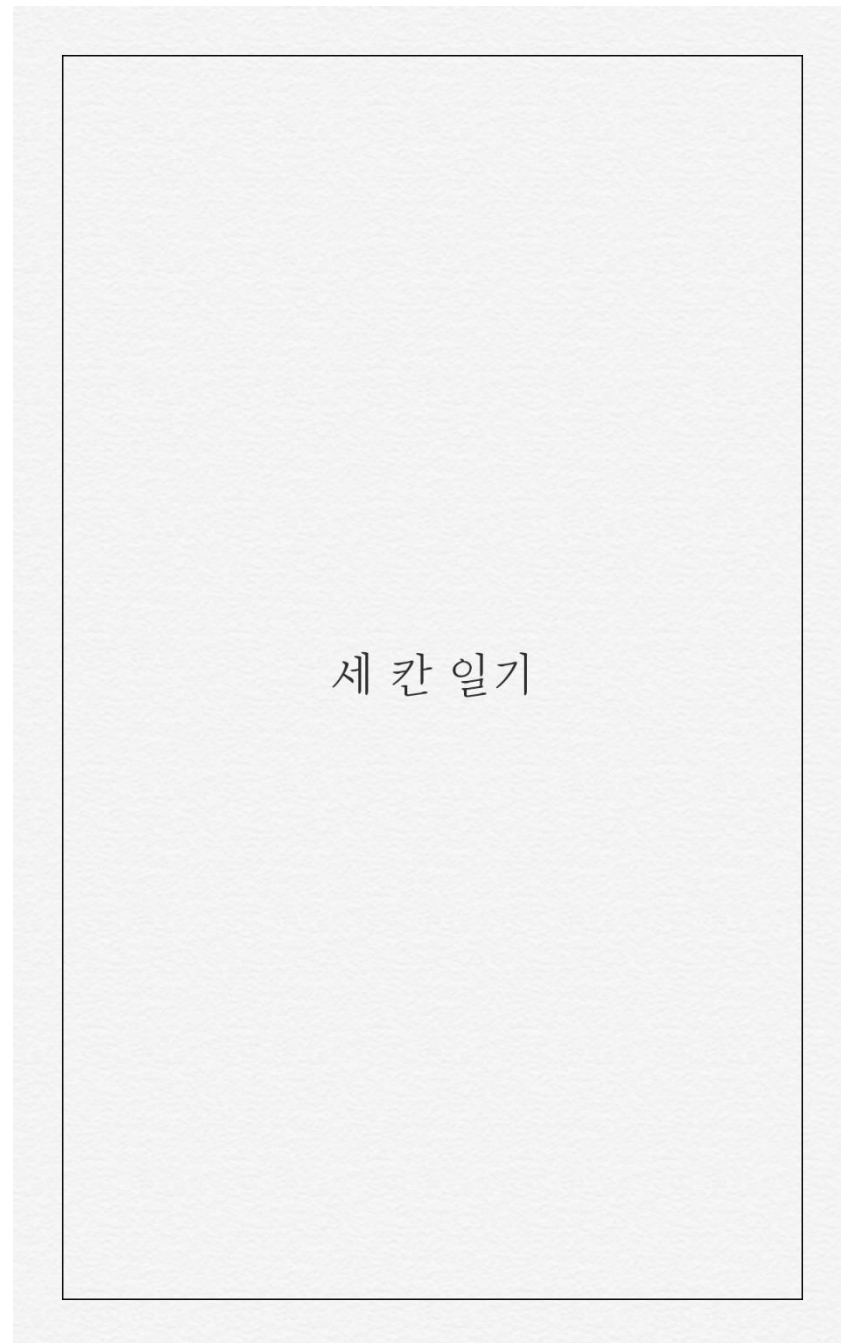
주제 선정 이유

02

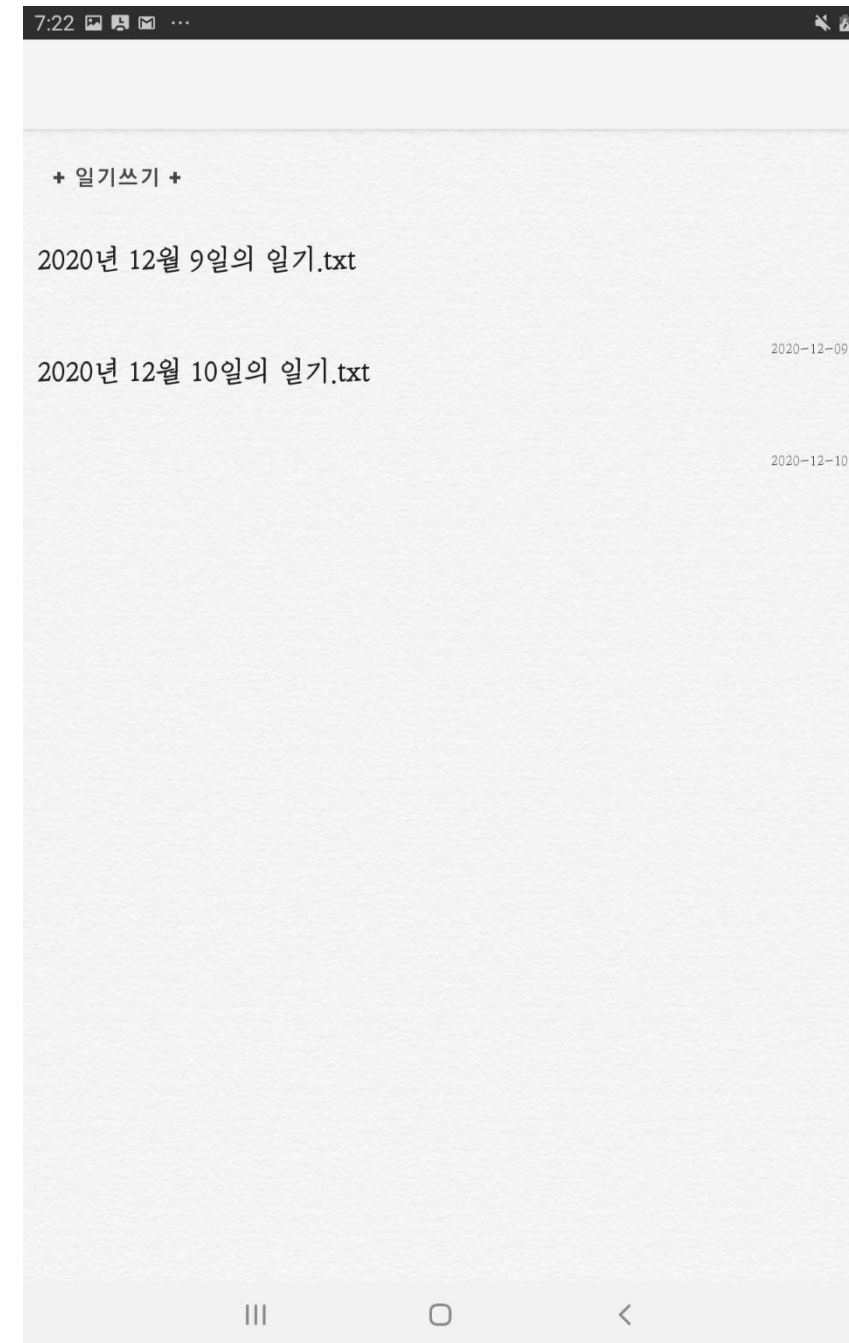
세 칸 일기 구성

03

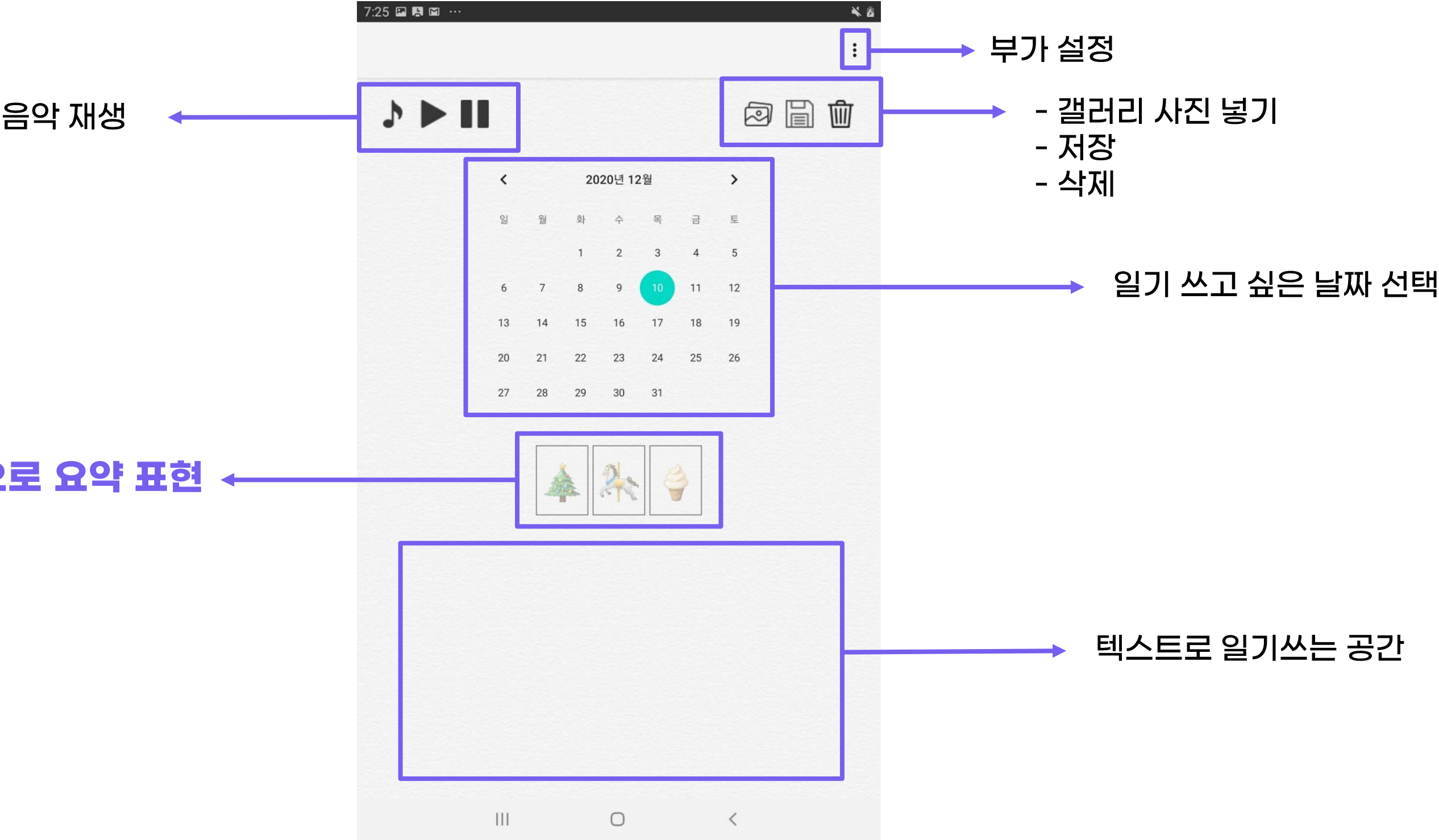
세 칸 일기 기능



시작 화면
(activity_intro.xml)



일기 목록
(activity_main.xml, items.xml)



일기 쓰는 화면
(activity_write.xml)

01

주제 선정 이유

02

세 칸 일기 구성

03

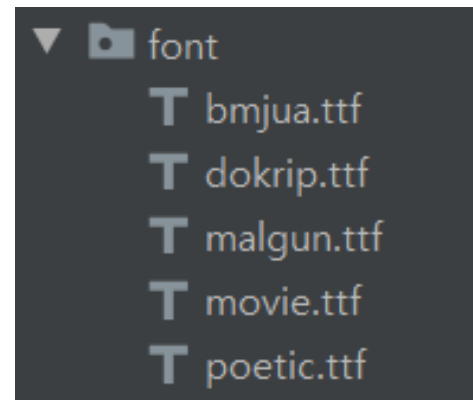
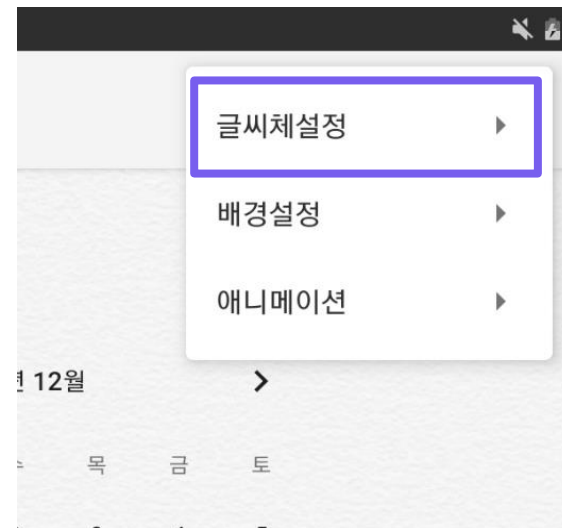
세 칸 일기 기능

글씨체 설정

- Part3 Menu 활용

onCreateOptionsMenu()

onOptionsItemSelected() 재정의



```
case ID_FONT_POETIC:
    Typeface typeface1 = getResources().getFont(R.font.poetic);
    d_write.setTypeface(typeface1);
    return true;
```

시인과 나

2020년 12월 10일

맑은 고딕

2020년 12월 10일

Yoon 독립

2020년 12월 10일

주아체

2020년 12월 10일

영화자막체

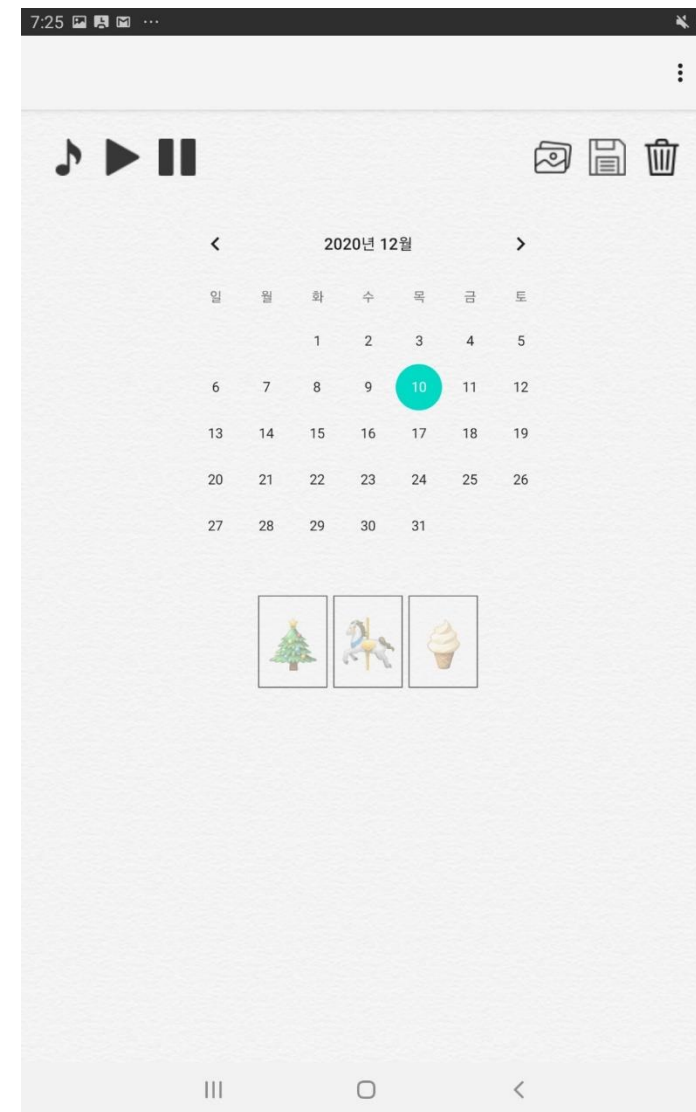
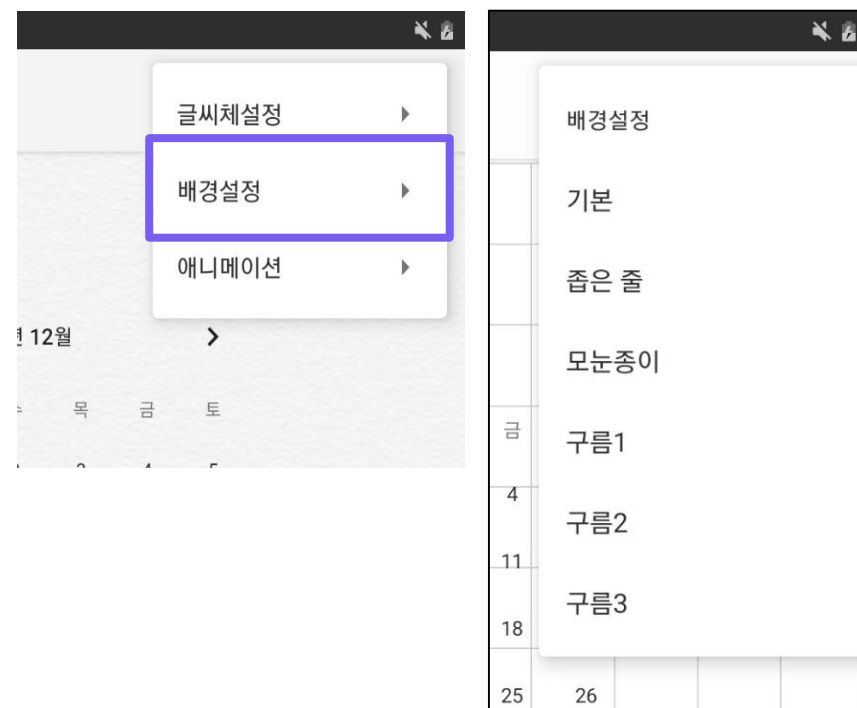
2020년 12월 10일

일기장 배경 설정

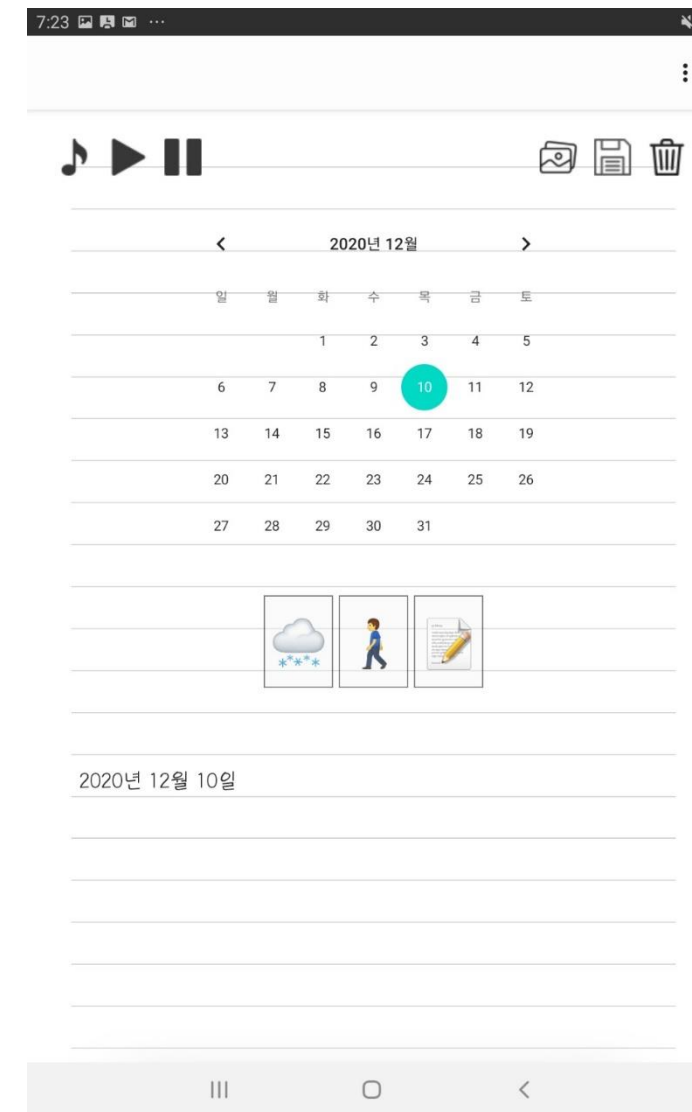
- Part3 Menu 활용

onCreateOptionsMenu()

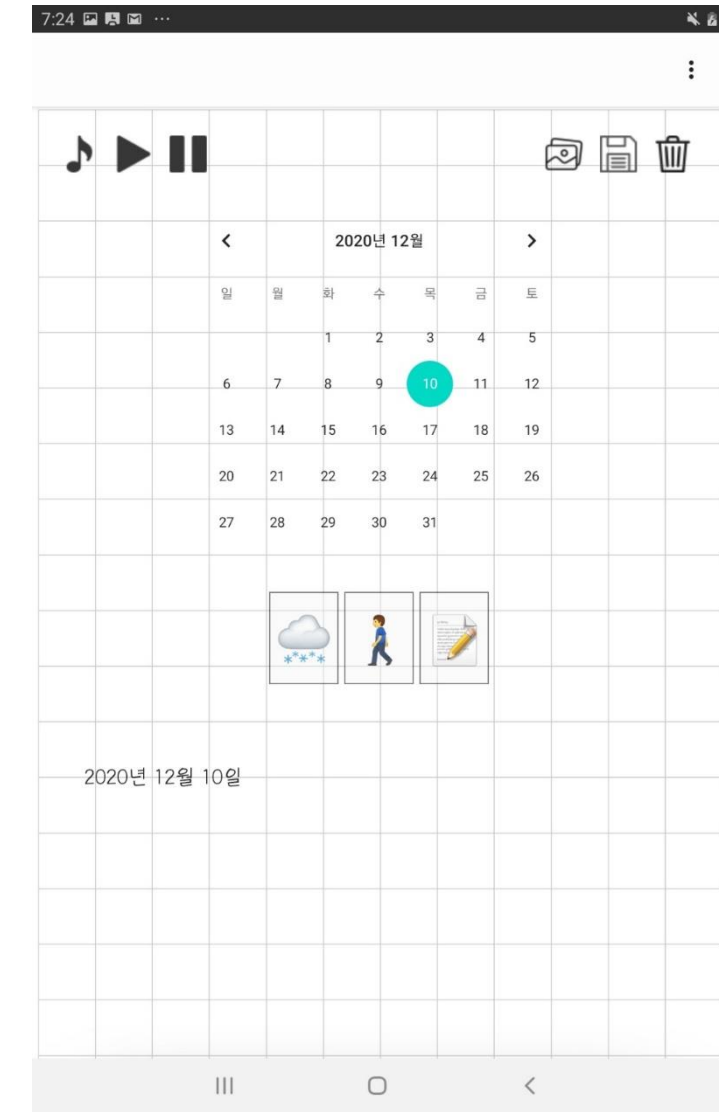
onOptionsItemSelected() 재정의



기본



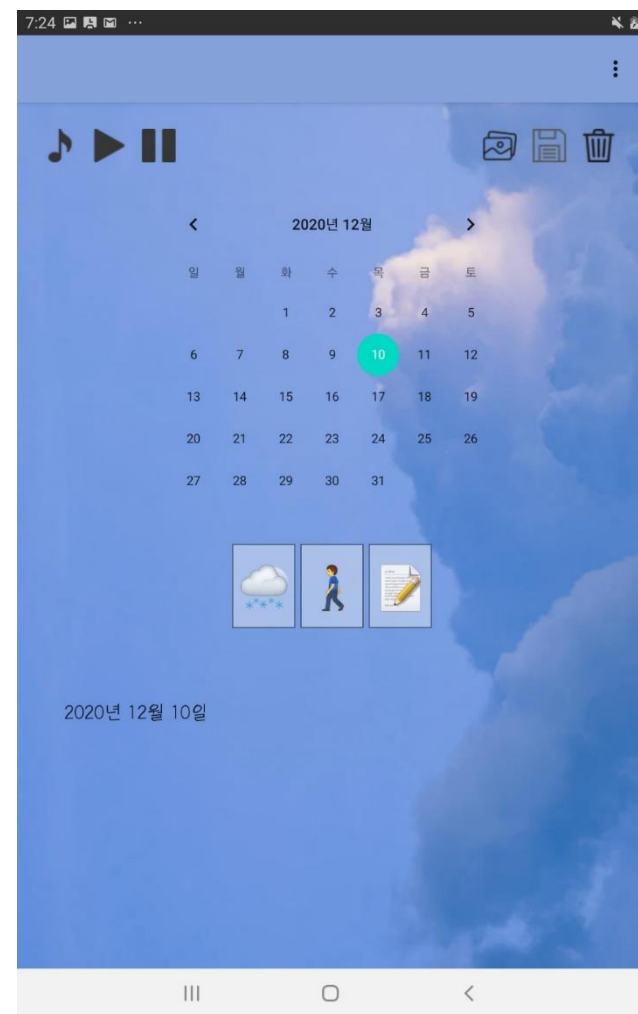
좁은 줄



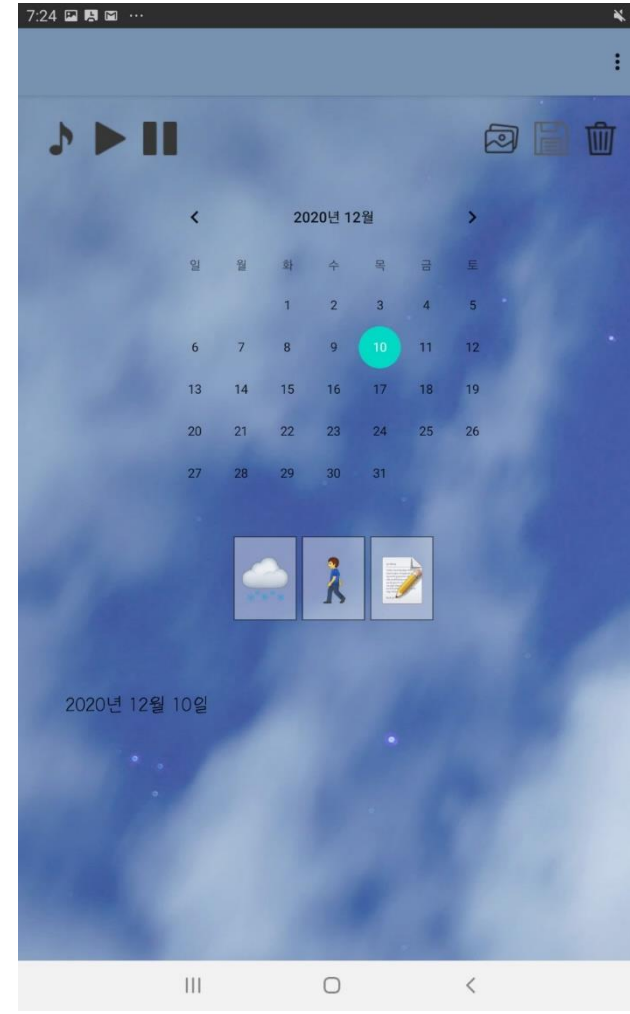
모눈종이

일기장 배경 설정

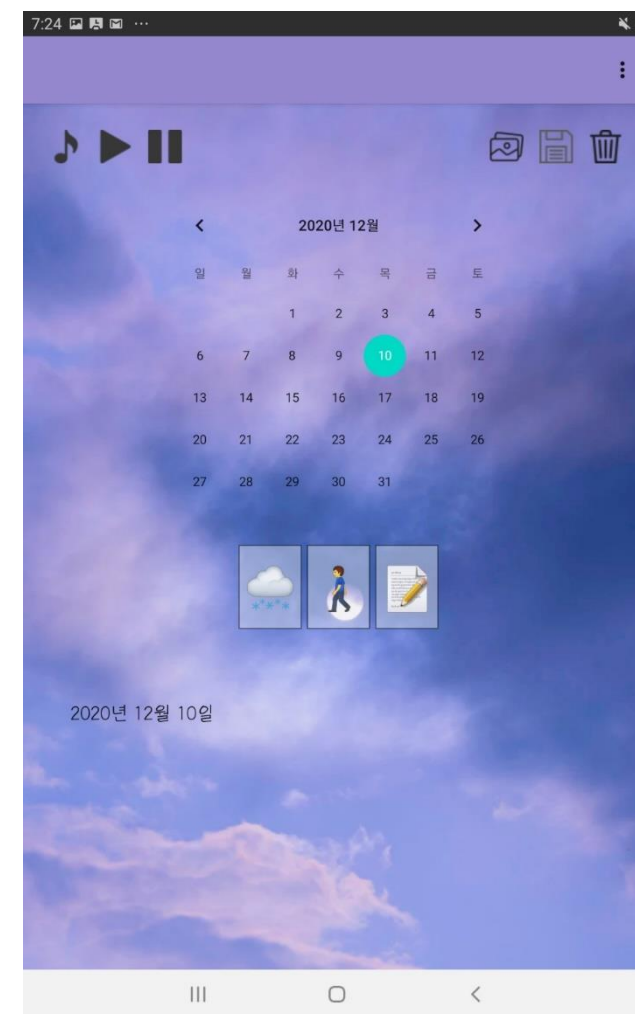
- Part3 Menu 활용



구름1



구름2

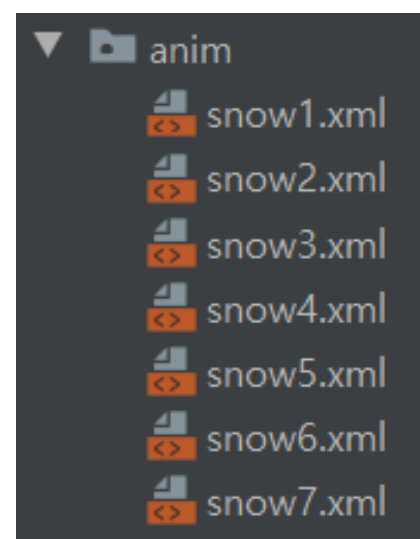
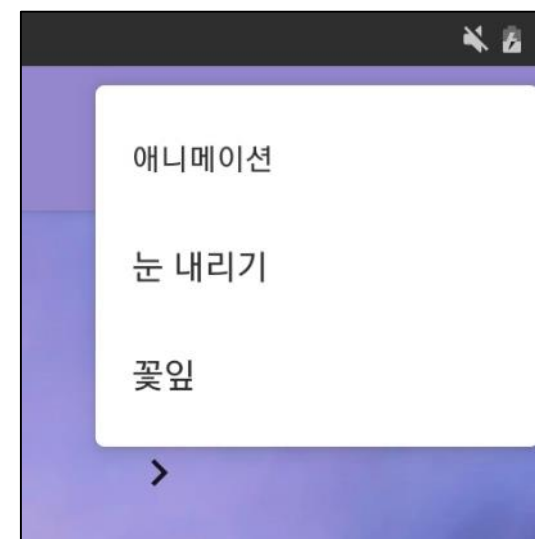


구름3

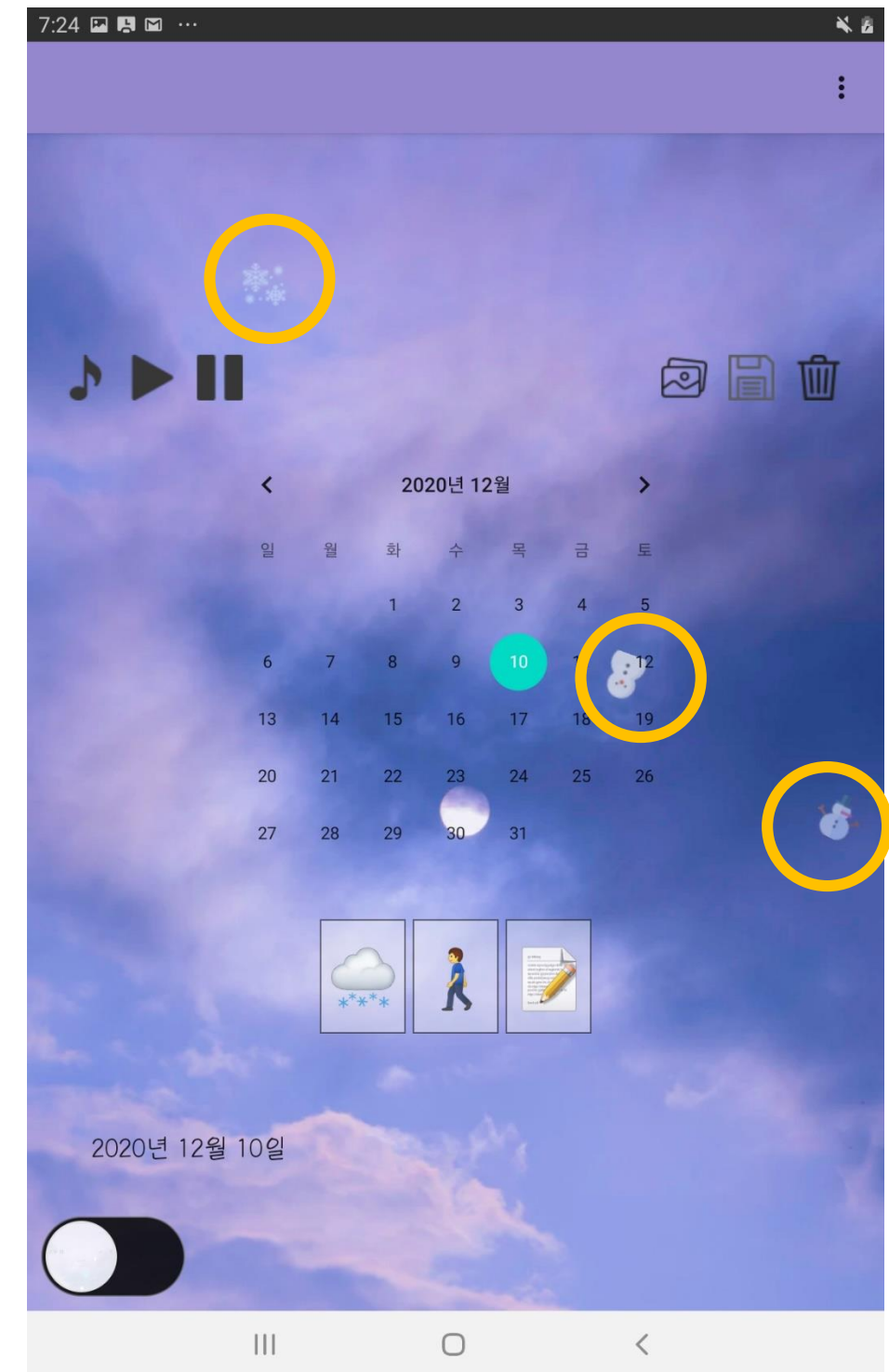
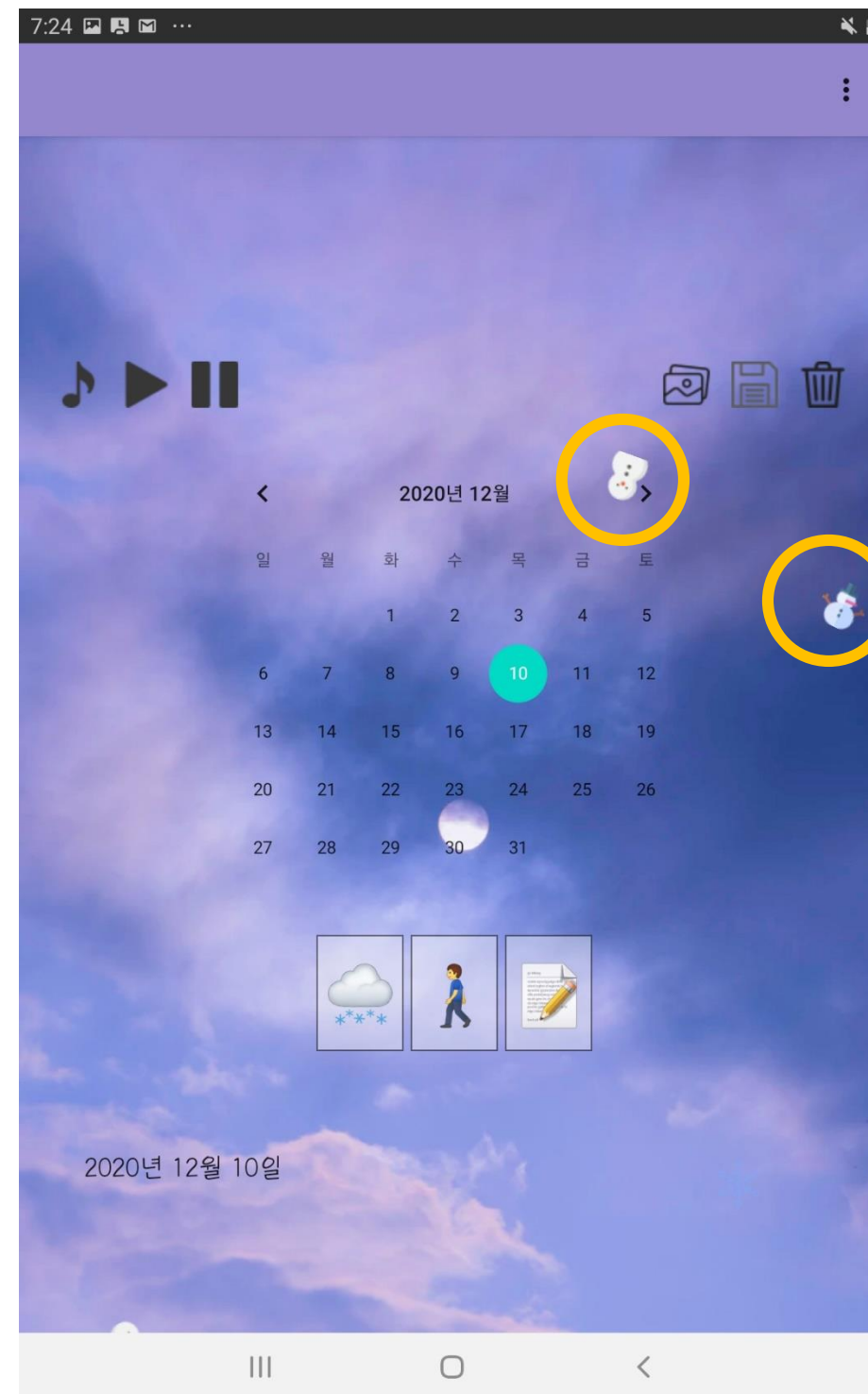
```
case ID_BG_SKY1:  
    linearLayout.setBackgroundResource(R.drawable.sky);  
    getSupportActionBar().setBackgroundDrawable(new ColorDrawable(0xFF84A1DA));  
    return true;
```

애니메이션 효과

- Part5 animation 활용

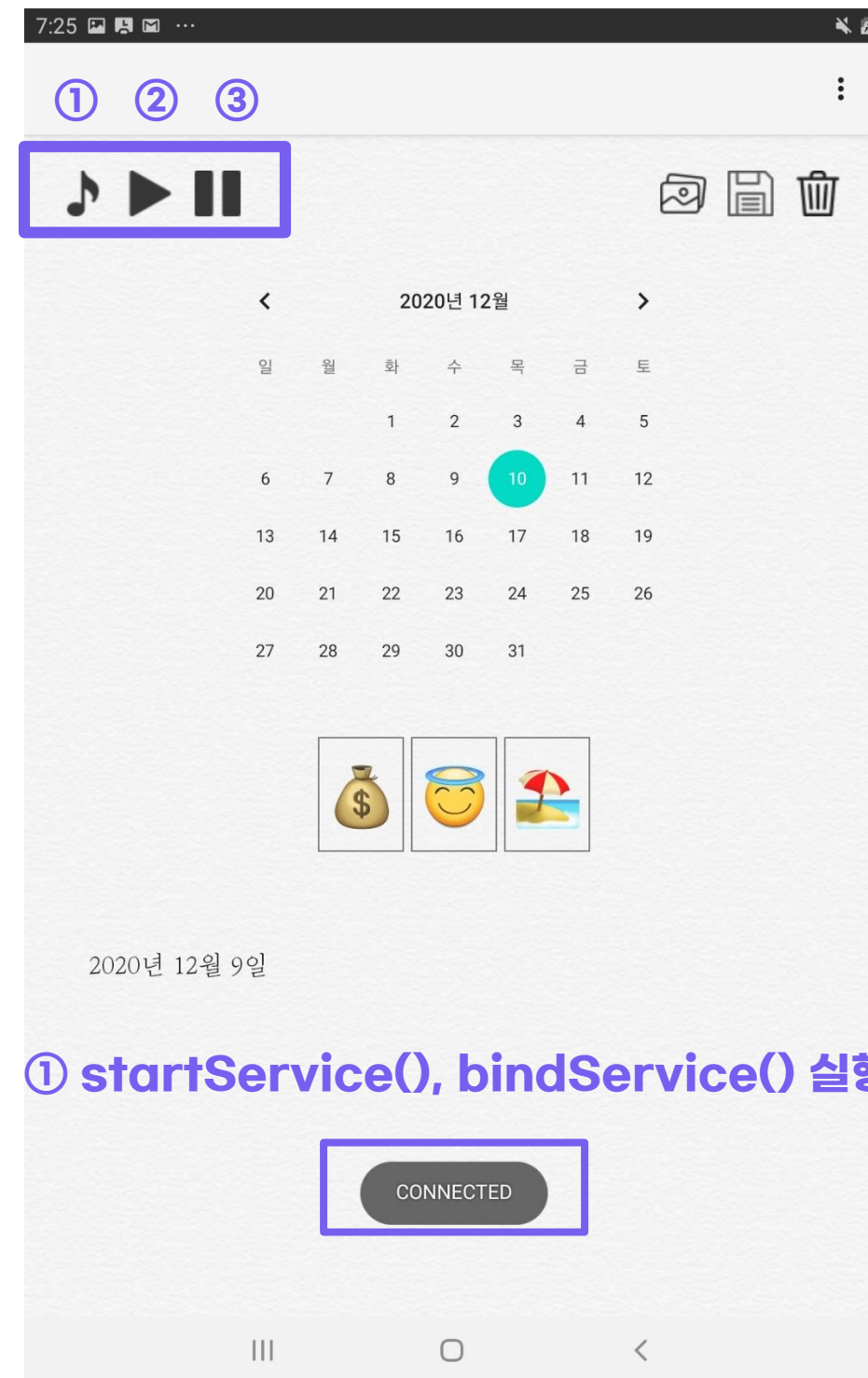


눈 내리기



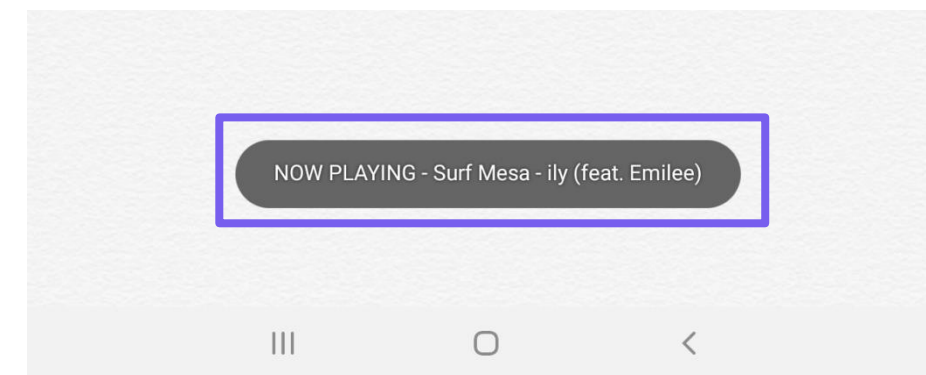
배경 음악 재생

- Part7 Service 적용



① startService(), bindService() 실행

② play() 메소드 실행 & Toast 메시지



③ pause 메소드 실행 & Toast 메시지

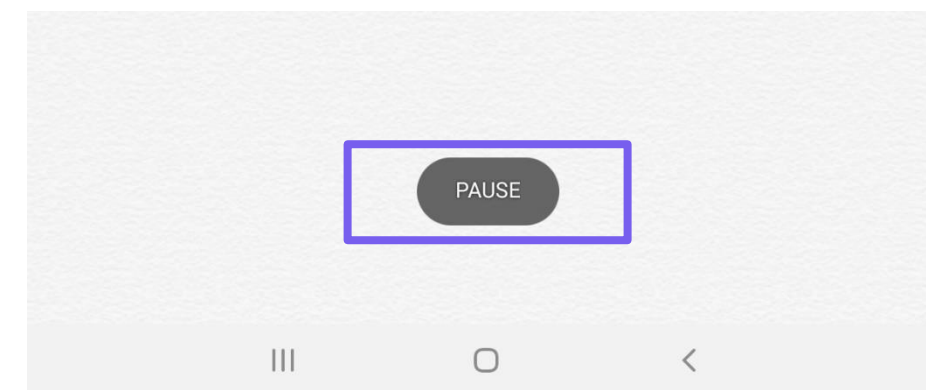
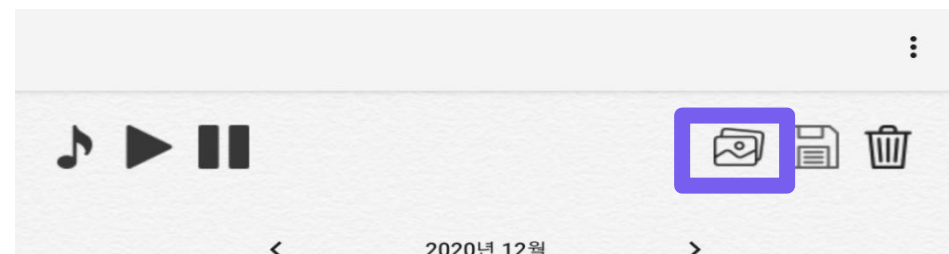


사진 넣기

- Part7 ContentProvider 활용

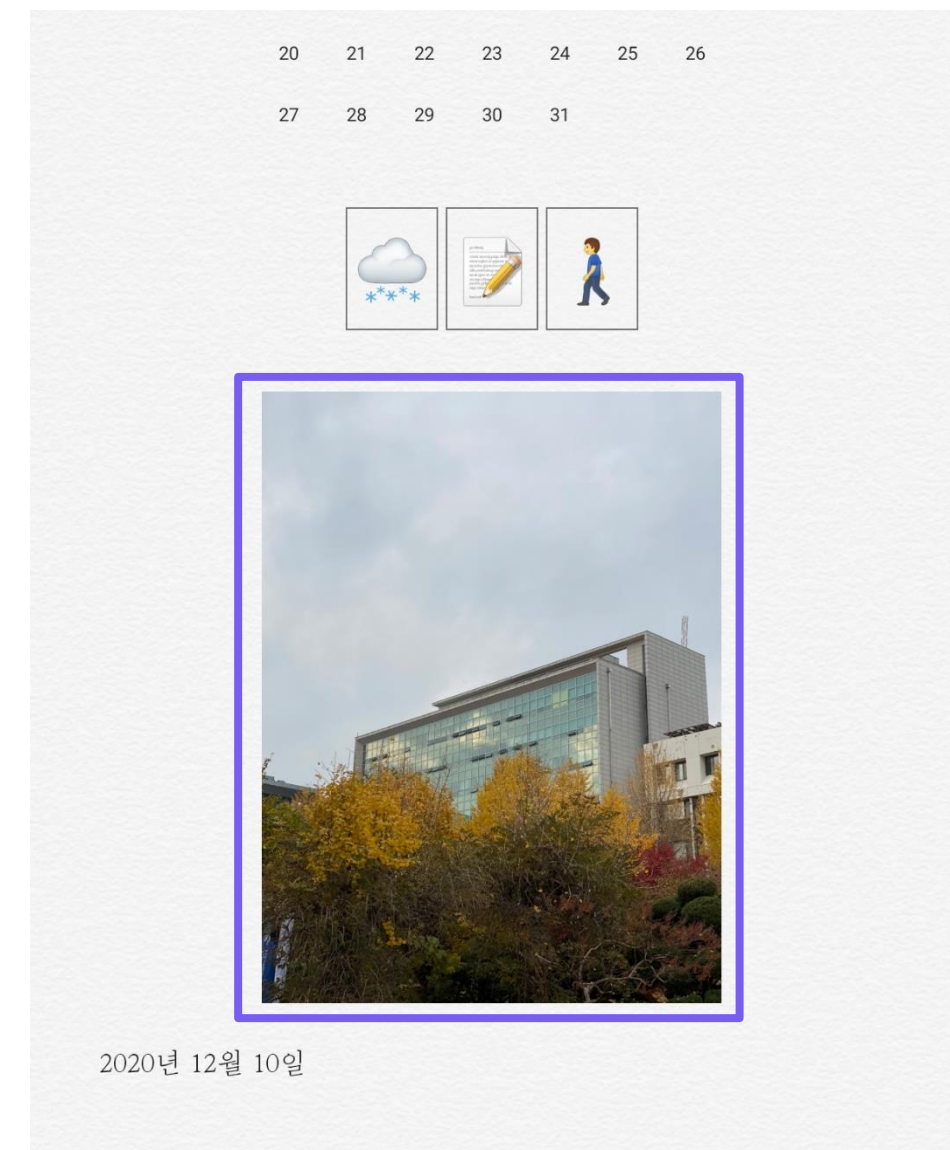
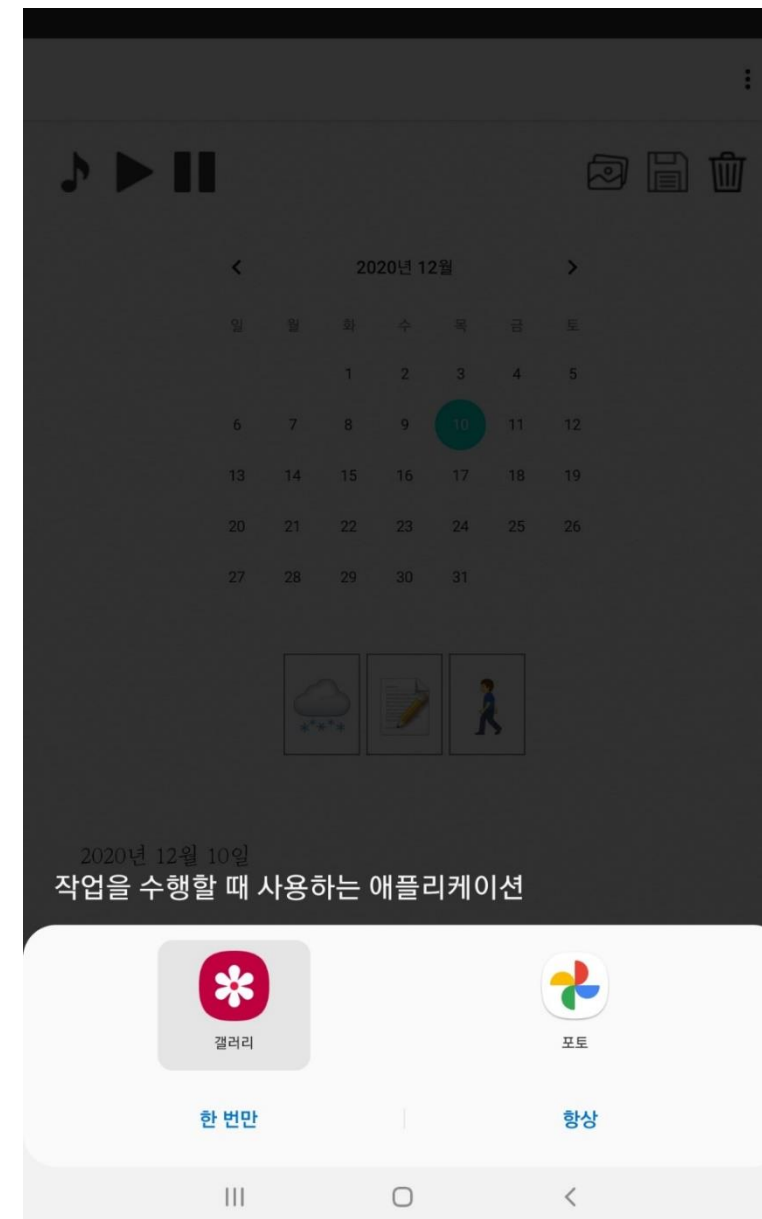
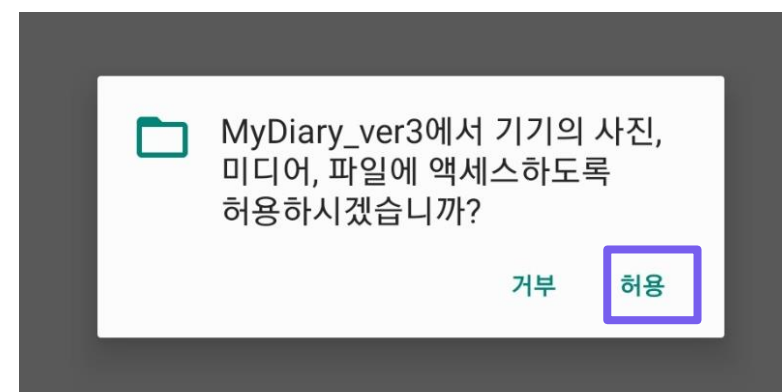
1. getRealPathURI()에서 이미지의 경로를 찾음
2. sendPicture()에서 경로를 이용해 사진을 비트맵으로 변환

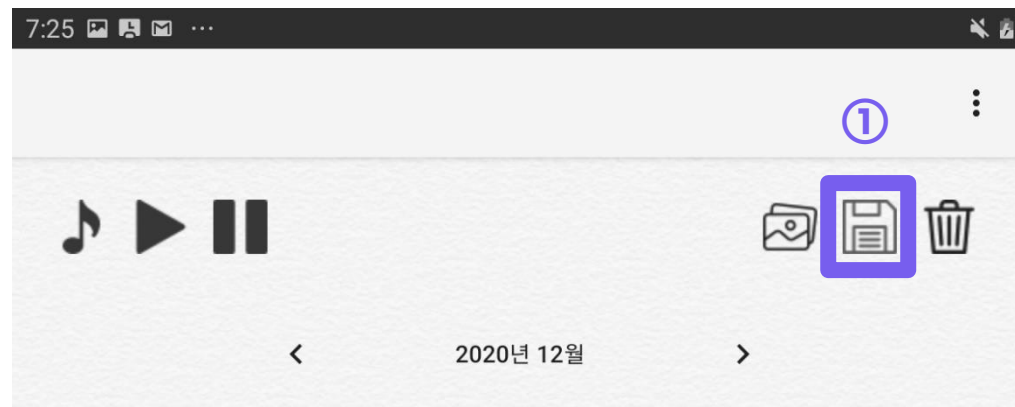


권한 설정

- 외부 저장소 읽기/쓰기

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```





① 디스크 버튼 클릭

```
save.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        saveDiary();
        GoToMain();
    }
});
```

```
private void GoToMain(){
    finish();
}
```

② 파일에 write한 뒤 데이터 베이스에 저장

```
boolean newFile = true;

try {
    fos = new FileOutputStream(getDiaryFile());
    fos.write(c_emoji1.getBytes());
    fos.write(c_emoji2.getBytes());
    fos.write(c_emoji3.getBytes());
    fos.write(content.getBytes());
    fos.close();

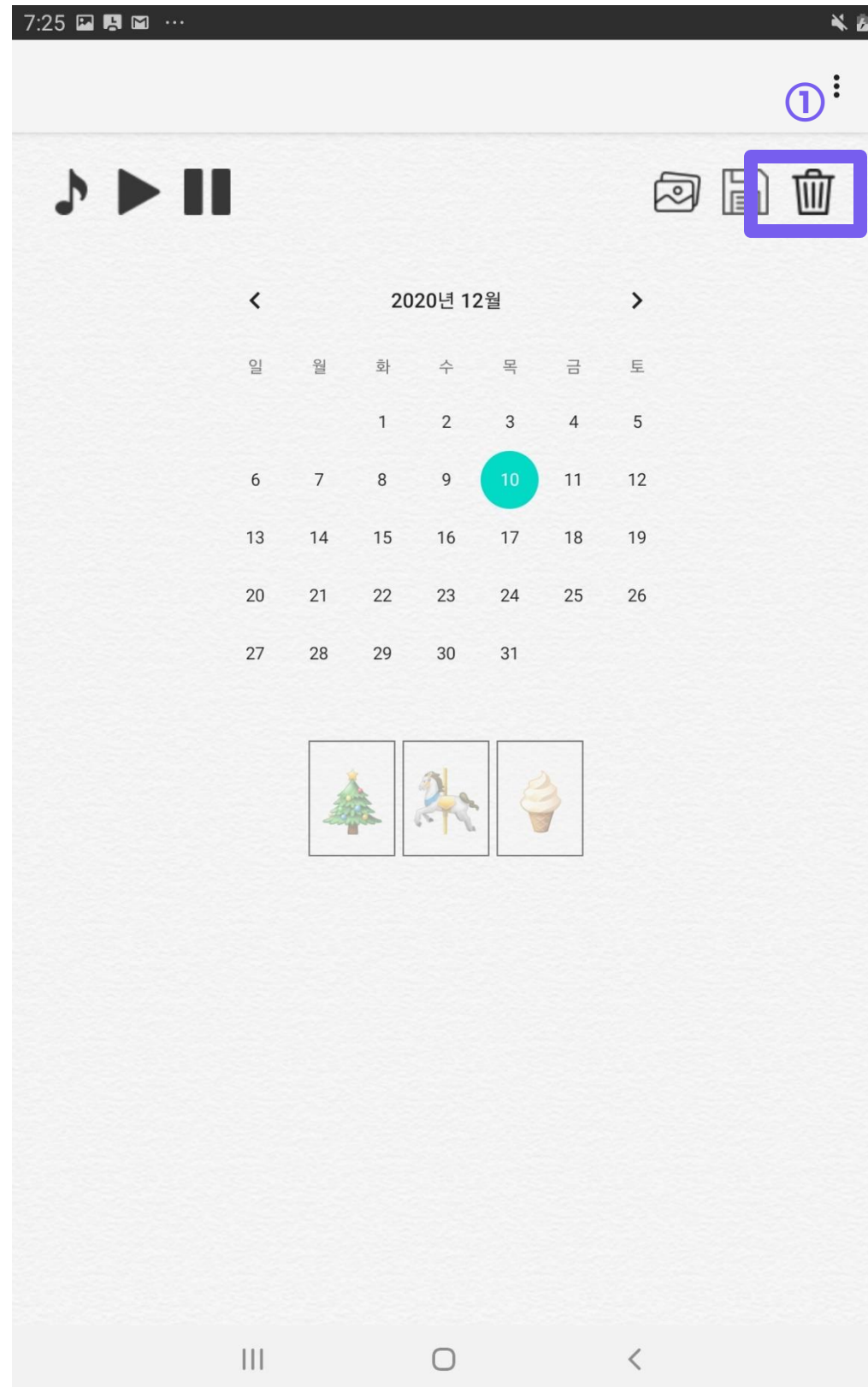
    if (newFile){
        diaryDB.saveDiary(diaryFile);
    }
}
```

③ 각 컬럼에 데이터를 put한 뒤 데이터베이스에 삽입

```
public void saveDiary(File file){
    if (file.exists() == false){
        return;
    }

    SQLiteDatabase db = getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(DiaryEntry.COLUMN_NAME_FILENAME, file.getName());
    values.put(DiaryEntry.COLUMN_NAME_DATE, getDate());
    values.put(DiaryEntry.COLUMN_NAME_IMAGE, ImagePath);

    db.insert(DiaryEntry.TABLE_NAME, nullColumnHack: null, values);
}
```

① 휴지통 버튼 클릭

```
delete.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        deleteDiary();  
        GoToMain();  
    }  
});
```

② 일기 파일을 가져와서 데이터베이스에서 삭제, 파일 삭제

```
private void deleteDiary(){  
    File diaryFile = getDiaryFile();  
  
    if (diaryFile.exists()){  
        diaryDB.removeDiary(diaryFile);  
        diaryFile.delete();  
    }  
}
```



감사합니다