

## **1. 서론**

- 1.1) 프로젝트 개요
- 1.2) 구현 동기 및 배경

## **2. 본론**

- 2.1) 어플리케이션 소개
- 2.2) 어플리케이션 구성
- 2.3) 구현
  - 2.3.1) 사용자 인터페이스
  - 2.3.2) 동작 과정
  - 2.3.3) 데이터베이스 설계
- 2.4) 테스트
  - 2.4.1) 기기별 테스트

## **3. 결론**

- 3.1) 어플리케이션 한계 및 개선사항

## 1. 서론

우선 앱을 소개하기에 앞서서, 본 앱을 고안하게 된 동기와 배경을 살펴보겠습니다.

본론에서는 서론에서의 내용을 바탕으로 본 앱(뚜두뚜두)의 주요 기능과 동작과정을 설명하고 소프트웨어적으로 어떻게 구현했는지 소개하겠습니다. 이어서 본 앱의 기기별 테스트를 통해 실제로 어떤 환경에서 어떻게 작동이 되는지 테스트 결과에 대해 서술하겠습니다.

결론에서는 본 앱을 통한 한계점, 기대효과를, 마지막으로 이번 프로젝트를 진행하면서 느낀점에 대해 피력하며 마치도록 하겠습니다.

### 1.1. 프로젝트 개요

내부 저장 공간(SQLite)를 이용한 Todo-list(메모) 어플리케이션입니다.

할 일을 메모(작성), 수정, 삭제, 검색 할 수 있는 메모장 어플리케이션입니다.

### 1.2. 구현 동기 및 배경

내부 저장 공간을 활용하여 "입력", "수정", "삭제", "검색"을 할 수 있는 실생활에 유용한 어플리케이션(이하 "앱")을 개발이 필요했습니다. 성별과 나이, 시간과 장소에 구애 받지 않고 누구나 사용할 수 있는 어플에 대한 고민 끝에 메모장 앱을 개발하고자 하였으며, 프로젝트 끝에 나온 결과물이 본 앱 "뚜두뚜두"입니다.

## 2. 본론

위의 내용을 통해서 개발한 앱을 깔끔하고 귀여운 느낌을 살려 개발해야겠다고 판단했습니다.

또, 사용자에게 메모장과 동일한 기능을 구현하고자 하였고, 도움말 기능을 통해 앱을 사용하는데 도움이 될 수 있는 서비스 기능을 제공하여 사용자에게 보다 다양한 편의를 제공할 수 있도록 하였습니다.

### 2.1 어플리케이션 소개

#### ✓ 어플명 : 뚜두뚜두

Todo-List를 목적으로 한 앱으로 영어 발음 Todo를 한국어로 귀여움을 더해 발음 하여 뚜두뚜두라고 칭했습니다.

#### ✓ 도움말 :: 기능 도움

앱 메인 화면 우측 하단 메뉴를 누를 경우, 앱의 기본적인 사용 설명서가 적혀 사용자의 앱 기능 숙지를 도와줍니다. 해당 도움말은 다른 버튼과 달리 하얀색 버튼으로 디자인 되어 있어, 다른 기능

보다 눈에 띄 수 있게끔 되어 있어, 구별이 용이하며 쉽게 접근 할 수 있습니다.

✓ **입력 :: 메모 추가**

메모장 앱이라면 할 일을 입력 할 수 있어야하는 필수적 기능입니다. 앱의 메인 화면에서 우측 하단의 메뉴를 눌렀을 때, 추가 버튼을 통해 할 일을 입력하여 저장 할 수 있습니다.

✓ **수정 :: 메모 수정**

앱의 메인화면 우측 하단 메뉴를 누른 뒤, 입력을 통해 저장된 메모(데이터)를 "꼭" 눌러 내용을 수정 할 수 있습니다.

✓ **완료 :: 완료 한 메모 혹은 삭제 할 메모**

입력 된 메모 중 완료한 메모를 구분 할 수 있습니다. 사용자는 입력한 메모중 완료한 해당 항목을 1회 터치(누름)으로써, 해당 항목의 글씨가 바뀌서 **취소선**이 그여집니다

✓ **삭제 :: 메모 삭제**

앱의 메인화면 우측 하단 메뉴를 누른 뒤, 삭제를 할 수 있습니다. 단, **취소선**이 있는 **완료된 항목**만 삭제 할 수 있습니다.

✓ **검색 :: 메모 검색**

앱의 메인화면 우측 하단 메뉴를 누른 뒤, "돋보기" 모양의 조회 버튼을 누르면 조회 화면으로 이동됩니다. 이동된 조회 화면에서 포함된 "단어" 혹은 "문장"을 통해서 검색 후 조회 할 수 있습니다. **삭제한 항목은 조회가 불가능하며, "스위치" ON/OFF 여하에 따라 미완료된 항목만을 조회하거나, 완료된 항목을 조회 검색** 할 수 있습니다.

## 2.2. 어플리케이션 구성

앱은 다음과 같이 크게 5개의 화면으로 구성됩니다. 실행(로딩)화면을 넣음으로써 본 앱의 심심함을 없앴습니다.

**메인 화면부**에서 사용자가 입력한 항목들을 받아오며 출력해주는 역할을 합니다. 더불어 플로팅액션버튼을 통해 버튼으로 인해 화면이 가려지는 답답함을 최소화 시켰습니다.

**입력화면 부**에서 캘린더 하단에 카드뷰 형식의 디자인을 더해 내용을 입력 하는 곳임을 직관적으로 알 수 있게끔 하였습니다.

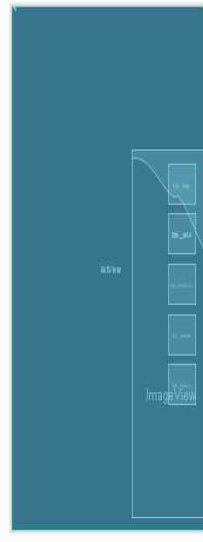
**조회(검색) 화면부**에서는 스위치 버튼의 ON/OFF를 통해 완료된 항목에 대한 **검색 조건을 설정** 할 수 있습니다. 예를들어 스위치가 ON인 상태에서 리스트에 있는 단어 혹은 문장을 검색할 경우 완료된(취소선이 그여져있는) 항목은 조회 되지 않습니다.

마지막으로, **도움말 화면**은 사용자가 어떤 기능을 어떻게 사용하는지에 대한 설명이 포함되어 있습니다. 사용자는 기능 사용법을 도움말 화면을 통해서 언제든지 확인 할 수 있습니다. 이어서 각각의 구현방법, 인터페이스, 데이터베이스 설계에 대해 세부하게 살펴보겠습니다.

실행 화면



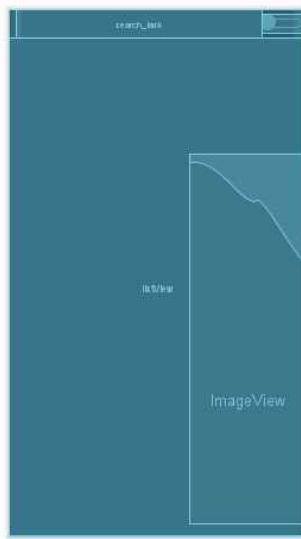
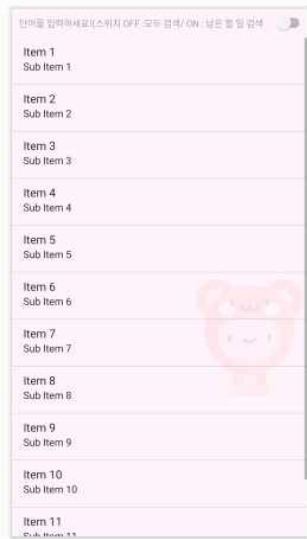
메인 화면



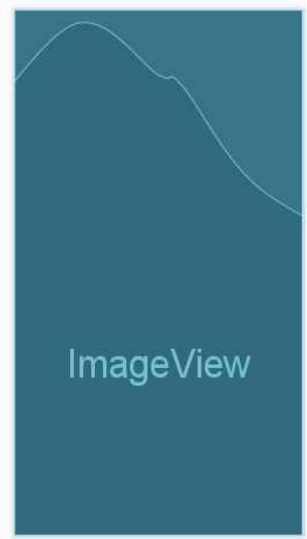
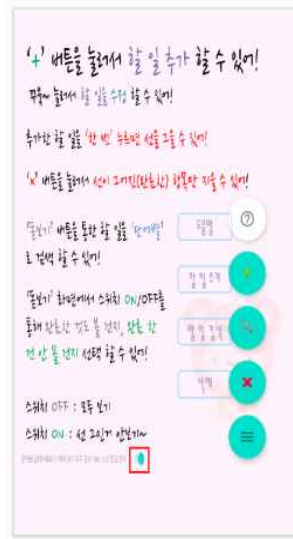
입력 화면



조회(검색) 화면



도움말 화면

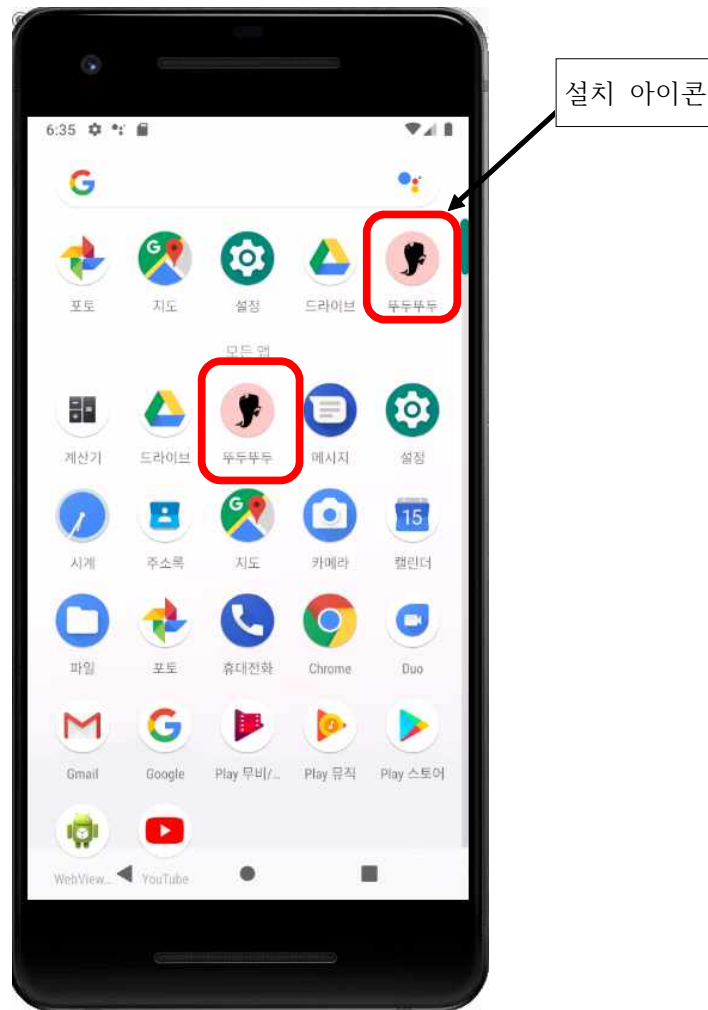


### 2.3. 구현

구현에서는 각 화면의 사용자 인터페이스와 동작 과정, 데이터베이스 설계 등을 상세하게 설명하겠습니다.

### 2.3.1. 사용자 인터페이스

우선 앱 설치시 스마트폰 내부에 설치되어 있는 아이콘입니다.



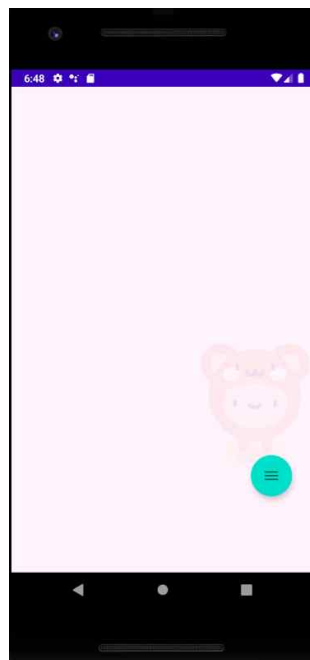
[사진 1 : 앱 설치 시 나타나는 실행 아이콘]

핑크색 배경(background)를 기반으로 고래가 노트에 메모를 하고 있는 아이콘입니다. 벡터이미지를 통해서 제작을 하는 것이 올바른 방식이나, 본 앱 개발과정에서는 직접 포토샵을 이용하여 비트맵 (레스터)이미지를 통해 제작하였습니다. 거부감 없이 딱딱하지 않게 구상하여 제작했기 때문에, 아이콘에서 사용자에게 **친근한 인상**을 심을 수 있습니다. 다음 장은 **실행화면**입니다.



[사진 2 : 앱 실행 시 나타나는 이미지]

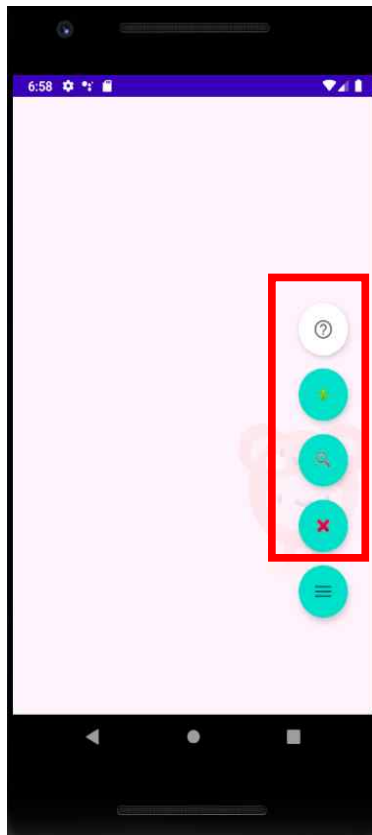
핑크색 배경을 메인으로 설치 아이콘과 동일한 모습의 메모하는 고래입니다. 핑크색 바다 속에서 일상을 메모하는 고래를 이미지로 형상화한 그림입니다. 아이콘 실행 시 바로 메인화면으로 넘어가는 과정에서 앱의 초기 세팅시간을 심심하지 않게, 또 갑작스레 메인으로 실행되는 괴리감을 줄이기 위해 스플래쉬를 통한 로딩이미지를 삽입하였습니다. 마찬가지로 포토샵을 사용한 비트맵 이미지입니다.



[사진 3 : 앱 실행 후 나타나는 메인 화면]

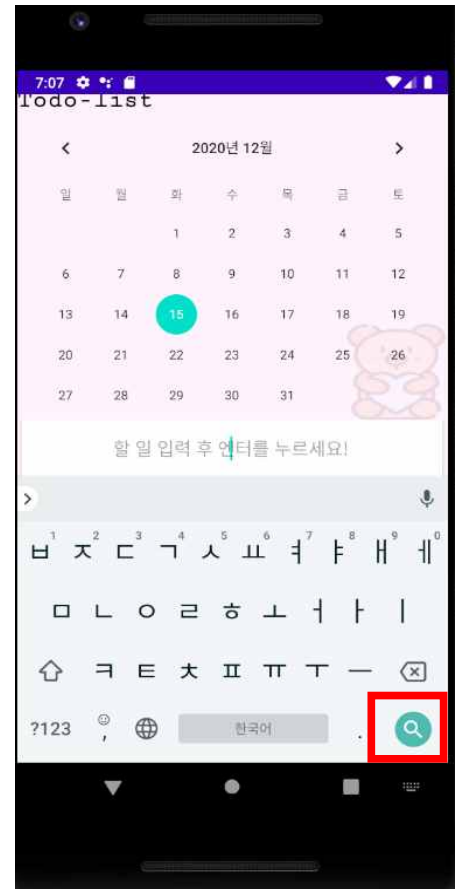
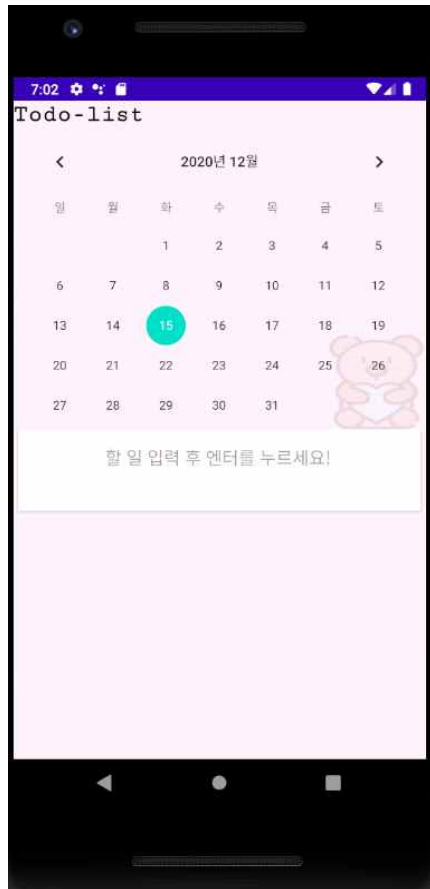
아이콘과 실행화면이 약간 연한 핑크색이었다면, 조금 더 색이 빠진 핑크색 혹은 **연한 보라색**으로 배경(background)상을 잡았습니다. 똑같은 색상의 컬러감을 도입하는 것 보다는 약간의 색상이 바뀌는 것이 조금 **더 좋은 컬러감**을 형성 할 수 있게 컬러의 "**톤인톤**" 매칭을 하여 비슷한 톤의 구성을 적용하였습니다.

**메인화면**에서는 심플한 화면으로 우측 하단의 플로팅 액션 버튼으로 구현한 **메뉴 버튼**이 있습니다. 사용법을 숙지하지 못 한 처음 접하는 사용자들도 직관적으로 메뉴라는 것을 알 수 있게끔 아무것도 존재하지 않는 화면 우측하단에 버튼을 형성하였습니다. 기존 버튼을 활용하지 않고 플로팅 액션 버튼 기능을 활용한 이유는 **본 앱('뚜두뚜두')의 궁극적 목적인 메모관리를** 함에 있어 적합하다고 생각하였기 때문입니다. 플로팅 액션 버튼을 활용하면 화면과 붙어 있어 자리를 차지하는 형태가 아닌, 화면과 약간 공간이 떨어져있는 느낌을 줄 수 있어 입력한 메모(데이터베이스에 저장 되어 있는 데이터)에 방해되지 않습니다. 또한 플로팅 액션 버튼을 통해 터치할 경우 다른 추가적인 작업 "입력", "삭제", "도움말" 기능을 이용 할 수 있습니다. 이어서 플로팅 액션 메뉴 버튼을 누를 경우의 메인 화면입니다.



[사진 4 : 메뉴 버튼을 누른 후 나타나는 메인 화면]

앞서 설명한 기본 메뉴 버튼이 있던 것과 달리 **애니메이션을 통해 메뉴 버튼 위로 4개의 버튼이 추가적으로 생성**됩니다. 사용자가 직관적으로 판단 할 수 있게, 추가, 검색, 삭제 등을 **알기 쉬운 벡터 이미지로 삽입**하였으며, 위로부터 도움말, 입력, 조회, 삭제입니다. 도움말 버튼(흰색 물음표 플로팅 액션 버튼)의 경우 다른 버튼과 달리 쉽게 눈에 띄고 구별 될 수 있게 버튼의 색상을 흰색으로 구현하였습니다. 다음은 **"+"** (입력버튼)을 터치하였을 경우 나오는 화면 구성입니다.



[사진 5 : 입력 버튼을 누른 후 나타나는 입력 화면]

[사진 6 : 할 일을 입력하는 화면]

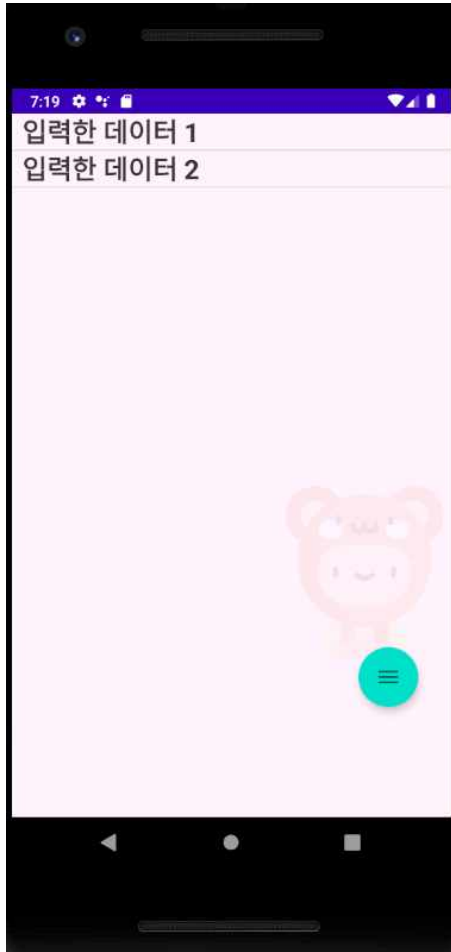
#### 사진 5 : 입력 버튼을 누른 후 나타나는 입력 화면

메인화면에서 '+' (입력) 버튼을 누르면 위와 같은 화면으로 전환이 됩니다. 캘린더를 통해 당일의 날짜를 확인 할 수 있으며, 캘린더 하단 중앙부에 카드뷰 형태의 텍스트 입력란으로 직관적으로 이란에 입력을 할 수 있는 곳임을 알 수 있게 구성하였습니다. 카드뷰 형태의 텍스트 입력란은 배경색상을 기준으로 "톤인톤" 색상을 통해 자연스러운 매칭을 하였습니다.

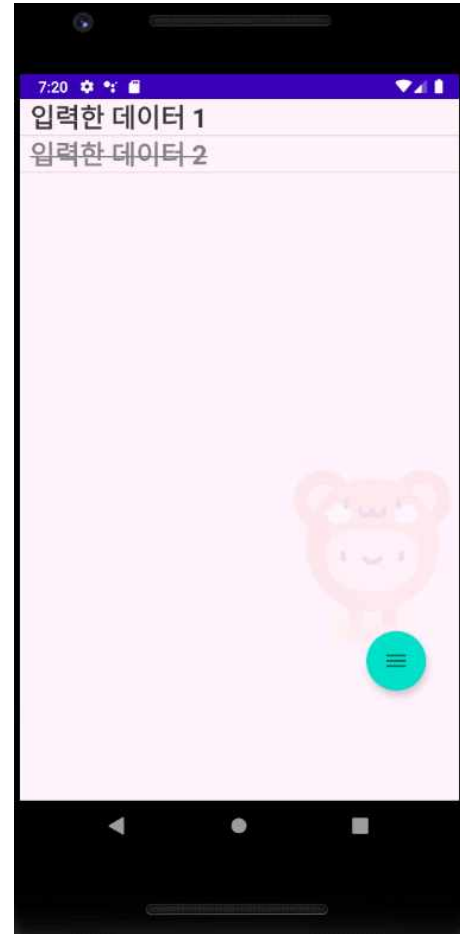
#### 사진 6 : 할 일을 입력하는 화면

입력할 텍스트(메모)를 작성하고 추가적인 버튼을 통한 저장 없이, 기본 키보드에 삽입되어 있는 엔터 혹은 입력 버튼(위 안드로이드 스튜디오 에뮬레이터에서는 돋보기 모양)을 통해 바로 저장 할 수 있게 구현하였습니다. 사용자는 텍스트를 입력하고 또 한 번 버튼을 통해 저장을 해야 하는 번거로움 없이 키보드에 내장 되어 있는 엔터 버튼을 통해 즉시 저장 할 수 있습니다. 저장이 완료되면 메인 화면으로 넘어가게 되면서 입력한 메모가 나타나게 됩니다.





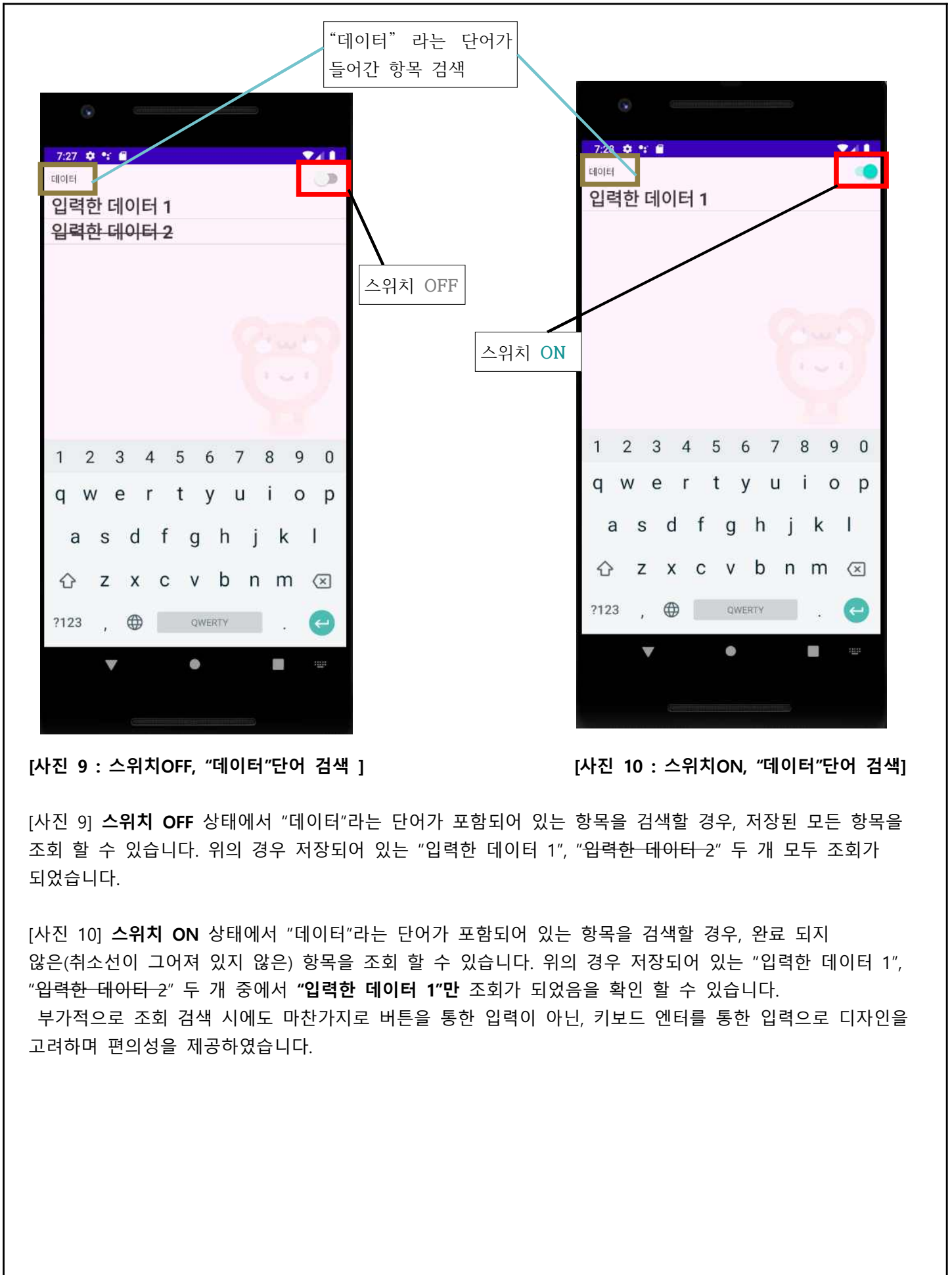
[사진 7 : 입력한 데이터가 저장되어 출력된 메인 화면]



[사진 8 : 해당 항목을 1회 터치한 화면]

#### 사진 7 : 입력한 데이터가 저장되어 출력된 메인 화면

입력화면을 통해 메모(데이터) 2개가 저장된 화면입니다. 해당 [사진7] 상태에서 **완료한 항목** 혹은 **삭제 하고 싶은 항목을 가볍게 1회 터치**할 경우 **[사진 8]**과 같이 글자의 색상이 회색으로 바뀌며 취소선이 글자위에 나타납니다.



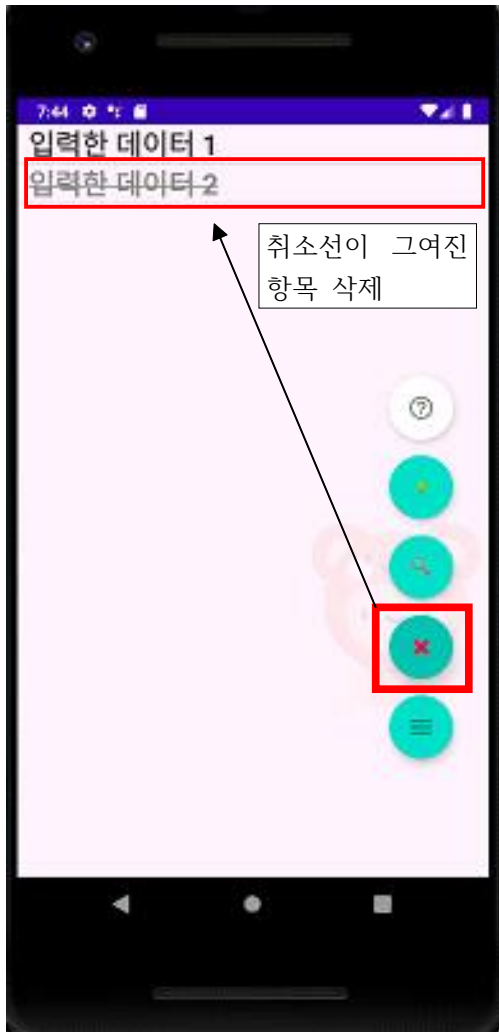
[사진 9 : 스위치OFF, “데이터”단어 검색 ]

[사진 10 : 스위치ON, “데이터”단어 검색]

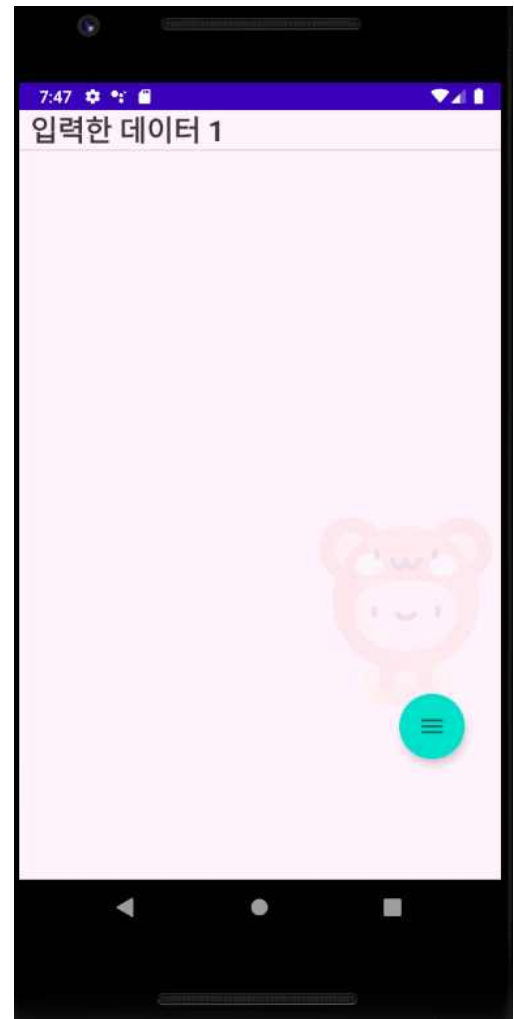
[사진 9] **스위치 OFF** 상태에서 “데이터”라는 단어가 포함되어 있는 항목을 검색할 경우, 저장된 모든 항목을 조회 할 수 있습니다. 위의 경우 저장되어 있는 “입력한 데이터 1”, “입력한 데이터 2” 두 개 모두 조회가 되었습니다.

[사진 10] **스위치 ON** 상태에서 “데이터”라는 단어가 포함되어 있는 항목을 검색할 경우, 완료 되지 않은(취소선이 그어져 있지 않은) 항목을 조회 할 수 있습니다. 위의 경우 저장되어 있는 “입력한 데이터 1”, “입력한 데이터 2” 두 개 중에서 **“입력한 데이터 1”만** 조회가 되었음을 확인 할 수 있습니다.

부가적으로 조회 검색 시에도 마찬가지로 버튼을 통한 입력이 아닌, 키보드 엔터를 통한 입력으로 디자인을 고려하며 편의성을 제공하였습니다.



[사진 11 : 항목 삭제]



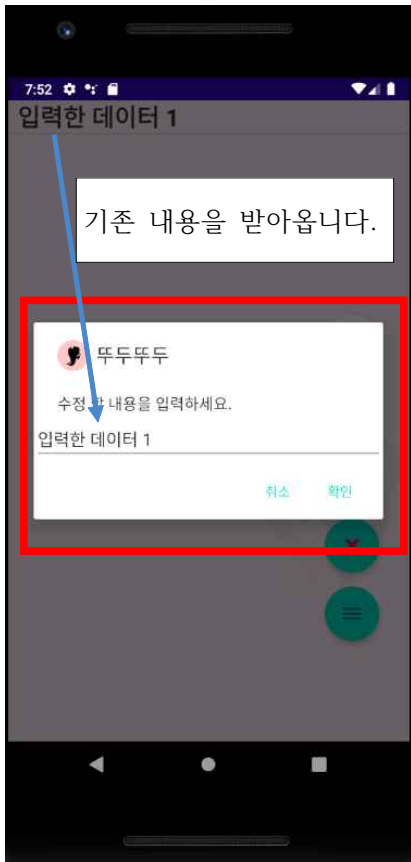
[사진 12 : 항목이 삭제 된 모습]

사진 11 : 항목 삭제

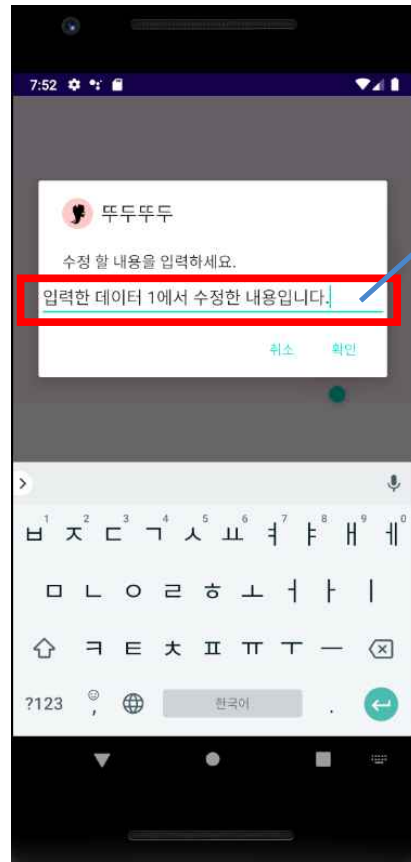
메뉴를 누르면 나오는 삭제 'X' 버튼을 누르면 취소선이 그여 있는 해당 항목이 삭제됩니다. 단, 취소선이 그여 있지 않는 항목은 삭제가 되지 않습니다..

사진 12 : 항목이 삭제 된 모습

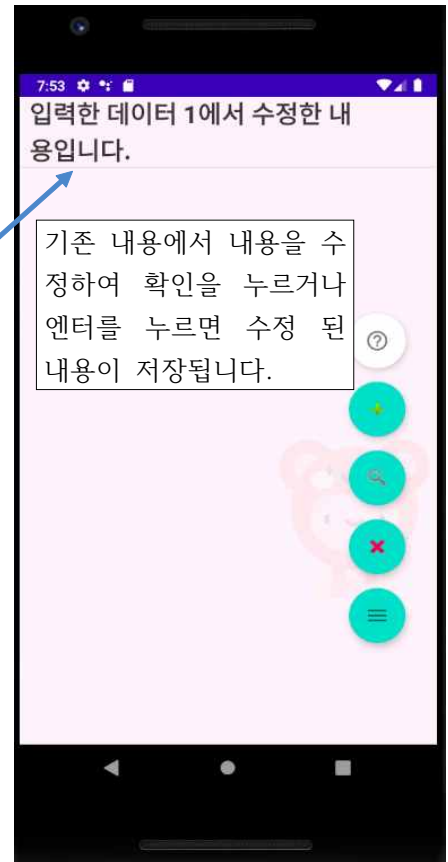
[사진 11]에서 “입력한-데이터-2”를 삭제하여 나타난 모습입니다. 기존 입력 된 항목은 사라졌으며, 완료되지 않은(취소선이 그여 있지 않은) 항목이 남아 있습니다. 다음 장에서 수정 인터페이스를 설명 드리겠습니다.



[사진 13 : 항목 수정 ]



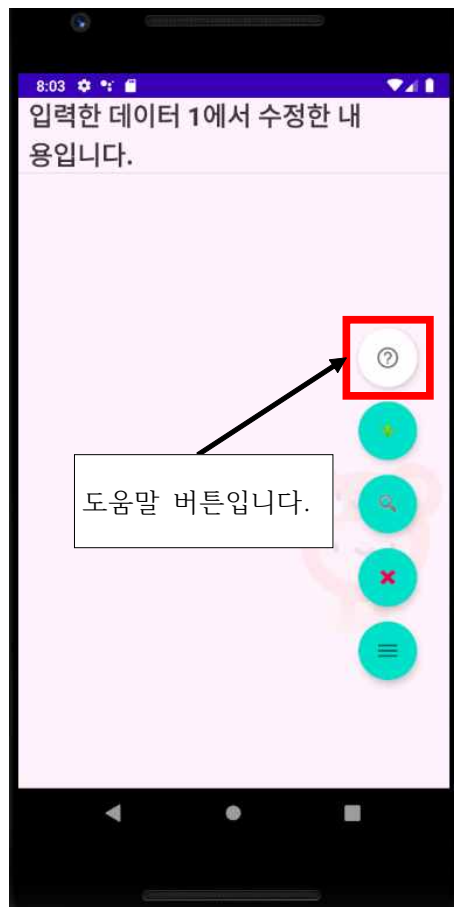
[사진 14 : 항목 수정 될 내용 ]



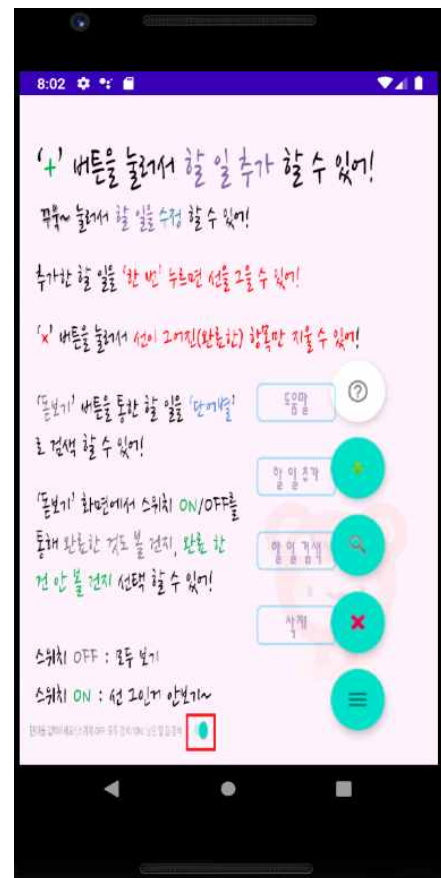
[사진 15 : 항목 수정 된 내용 ]

## 수정

기존에 입력했던 항목을 **꼭 누르면** 내용을 수정 할 수 있는 팝업 창이 나타납니다. 해당 팝업창은 기존에 입력 했던 내용을 그대로 받아오며 사용자가 필요한 내용을 첨삭할 수 있습니다. 위 예시 이미지 [사진13] 및 [사진 14]는 "입력한 데이터 1" 이라는 항목을 **꼭 눌러서** 나타난 팝업 창에 "입력한 데이터 1에서 수정한 내용입니다" 라는 문구로 수정 하여 확인을 눌렀습니다. 이후 [사진 15]와 같이 팝업 창에서 수정 했던 내용으로 갱신되어 저장이 되었음을 확인 할 수 있었습니다.



[사진 16 : 도움말 버튼 ]

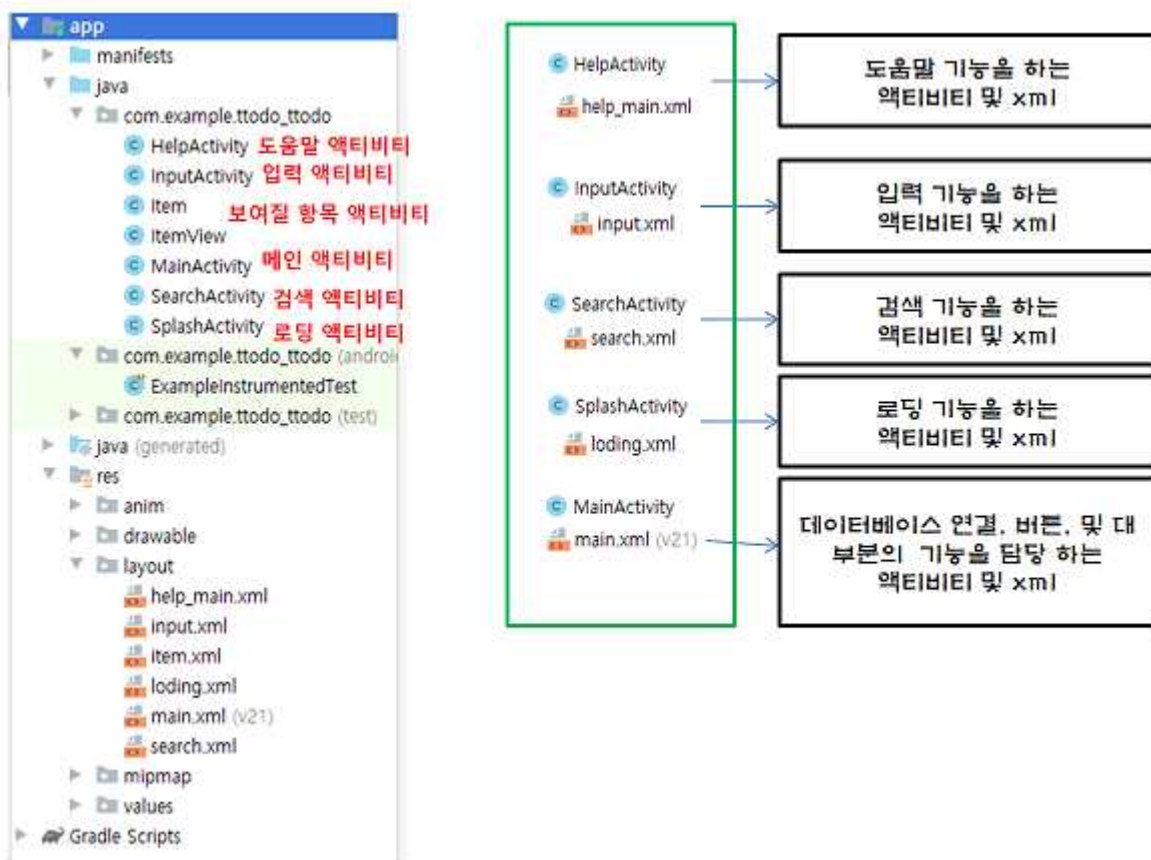


[사진 17 : 도움말 화면 ]

## 도움말

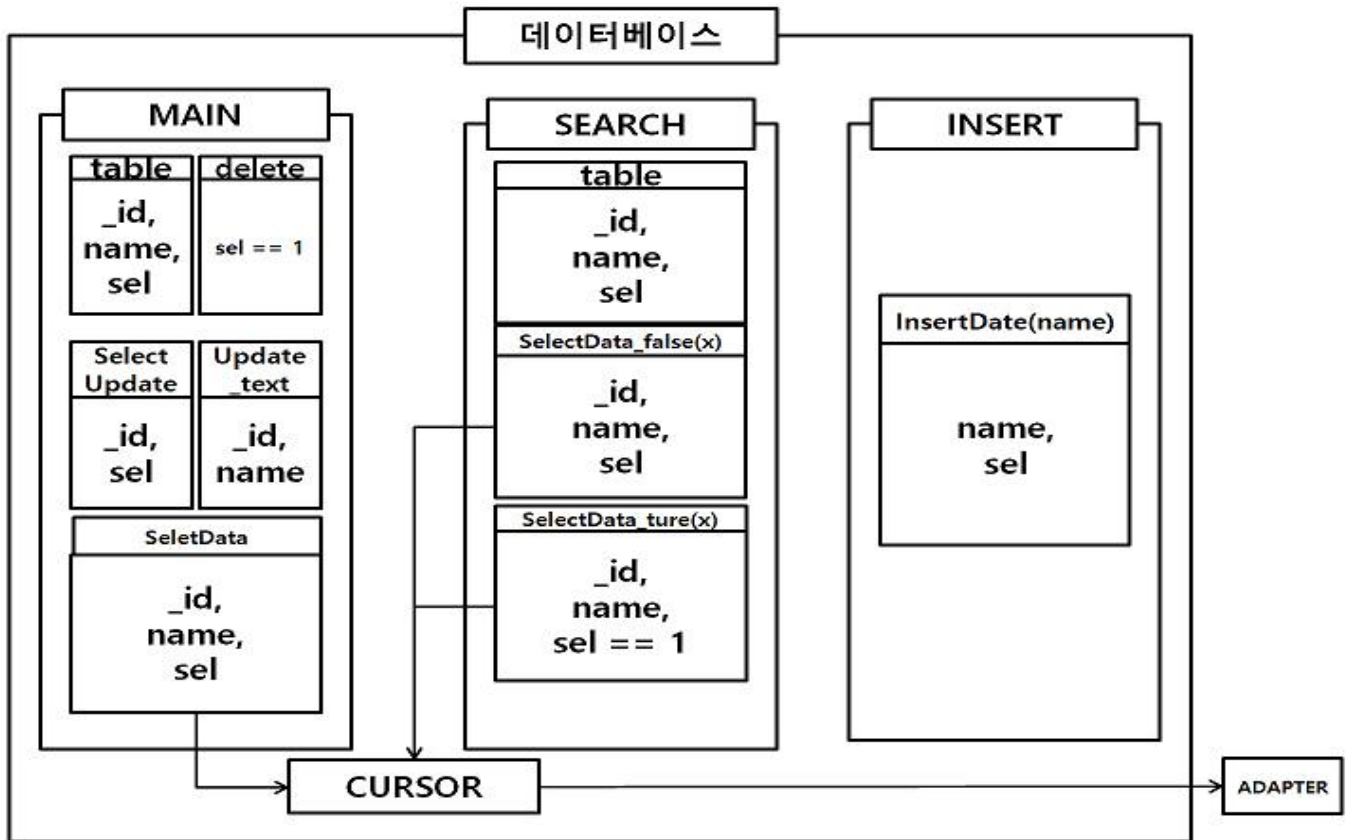
사용자는 메인 메뉴의 도움말 플로팅 액션 버튼을 클릭하여 **본 앱의 사용 설명서를 확인** 할 수 있습니다. **입력, 수정, 저장, 삭제**등 본 앱을 사용하는데 필요한 모든 정보가 포함되어 있습니다. 본 도움말은 사용자 필요시 언제든지 버튼을 눌러 확인할 수 있습니다.

### 2.3.2) 동작 과정



- ✓ HelpActivity와 help\_main.xml은 위에서 설명했던 것처럼 도움말 화면을 담당하는 파일입니다.
- ✓ InputActivity와 input.xml은 입력기능을 담당하는 화면이 있는 파일입니다.
- ✓ SearchActivity와 search.xml은 검색(조회)를 담당하는 화면이 있는 파일입니다.
- ✓ SplashActivity와 loading.xml은 앱 실행시 로딩(스플래쉬)를 담당하는 파일입니다.
- ✓ MainActivity와 main.xml은 기본적인 데이터베이스가 구현되어 있으며, 버튼의 기능, 애니메이션, 리스트와 어댑터등 연결을 담당하는 메인 파일입니다.
- ✓ Item, ItemView는 부가적 파일로, 저장되는 항목의 디자인을 담당하는 파일입니다.

### 2.3.3) 데이터베이스 설계입니다.



데이터베이스 동작 과정입니다.

**MAIN**에서 테이블에 `_id`, `name`, `sel`을 담습니다.

**delete** : 삭제를 하기 위해서는 `sel`값이 1이 되어야합니다. `sel == 1` 조건을 만족하는 데이터를 삭제합니다.  
데이터베이스에 저장 되어 있는 데이터 중 `sel`이 1인 데이터를 삭제합니다.

**SelectUpdate** : `_id`, `sel`을 받아서, 해당 항목을 조사하며, `sel`에 1을 담거나 0을 담습니다.  
해당 `_id`(KEY)값을 받아 키에 해당하는 값에 `sel` 값을 갱신합니다.

**Update\_text** : `_id`, `name`을 받아서, `name`의 값을 수정시킵니다(다른 값 교체)  
`_id`(KEY)값을 받아 키 값에 해당하는 곳에 `name`을 새로 넣어 `name`을 갱신합니다.

**SelectData** : `_id`, `name`, `sel`을 받아, Cursor에 보낸 뒤 해당 값을 검사하여 ADAPTER에 보내 출력합니다.

**SEARCH**에서 테이블에 `_id`, `name`, `sel`을 담습니다.

**SelectDate\_false(x)** : `_id`, `name`, `sel` 값을 받아 `x`에 값을 모두 들고 옵니다.  
`sel` 값에 상관 없이 받아서 커서에 담아 어댑터로 출력합니다.

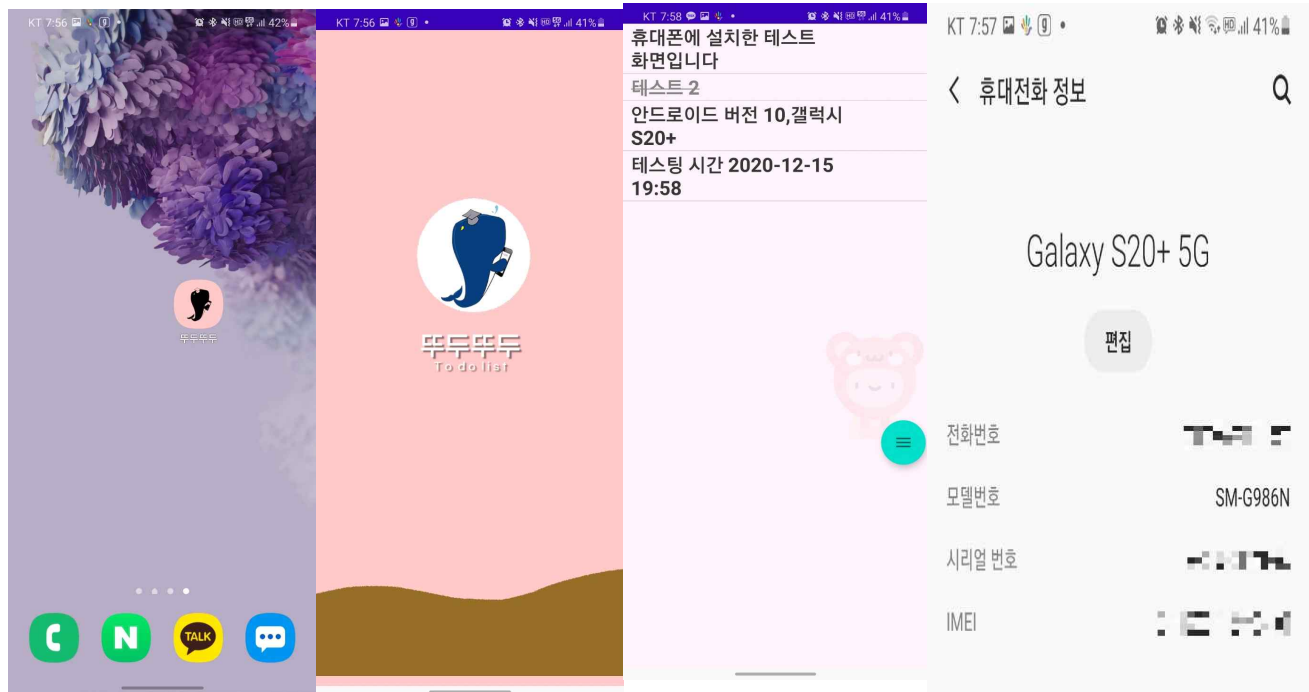
**SelectDate\_true(x)** : `_id`, `name`, `sel`을 받지만, `sel == 1`을 만족하는 값을 받아옵니다.  
`sel`의 값이 1인 경우의 데이터만을 커서에 담아 어댑터로 보내 출력합니다.

**INSERT**에서 **InsertDate(name)**에 `name`, `sel`을 받아 데이터베이스에 담습니다.

## 2.4. 테스트

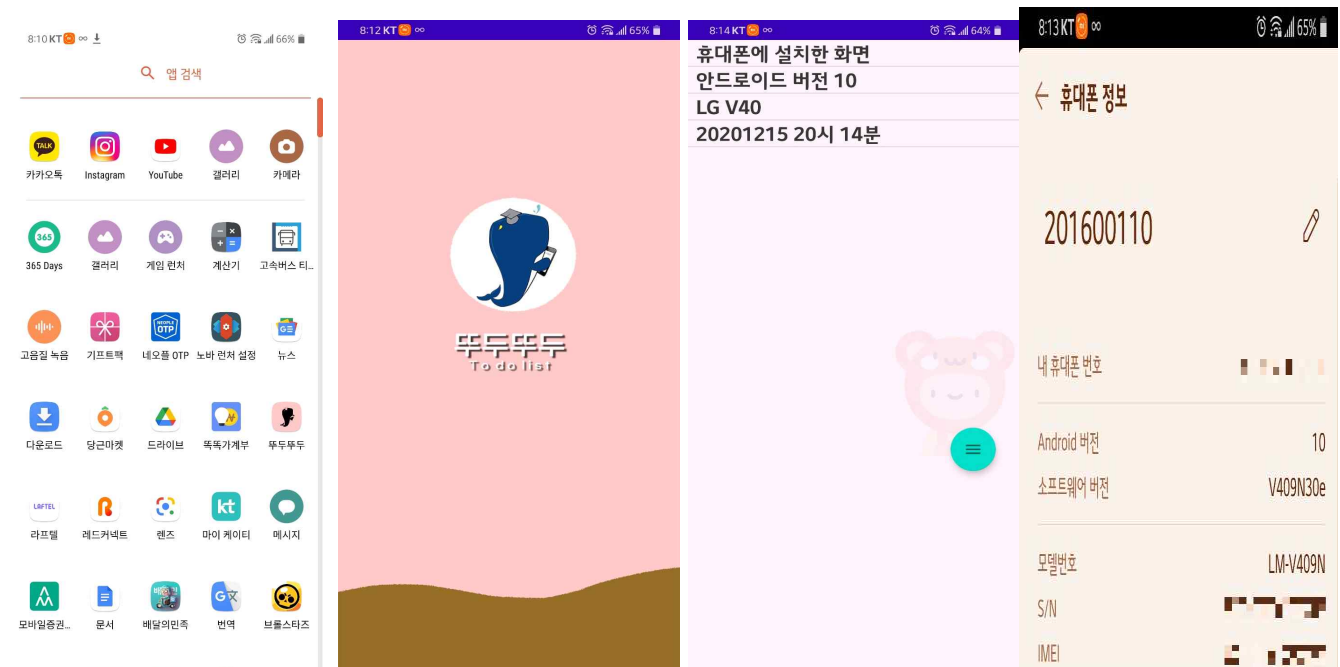
본 앱의 테스트는 APK파일을 통해 설치한, **안드로이드 기기별 실제 사용 테스트**입니다. 다만, 모든 테스트 과정을 보고서 내에 담는데에 어려움이 있어, 기기별 테스트과정의 일부만 본 내용에 명시하겠습니다.

### <삼성 갤럭시S20+ 안드로이드 버전 10>



[정상적으로 작동]

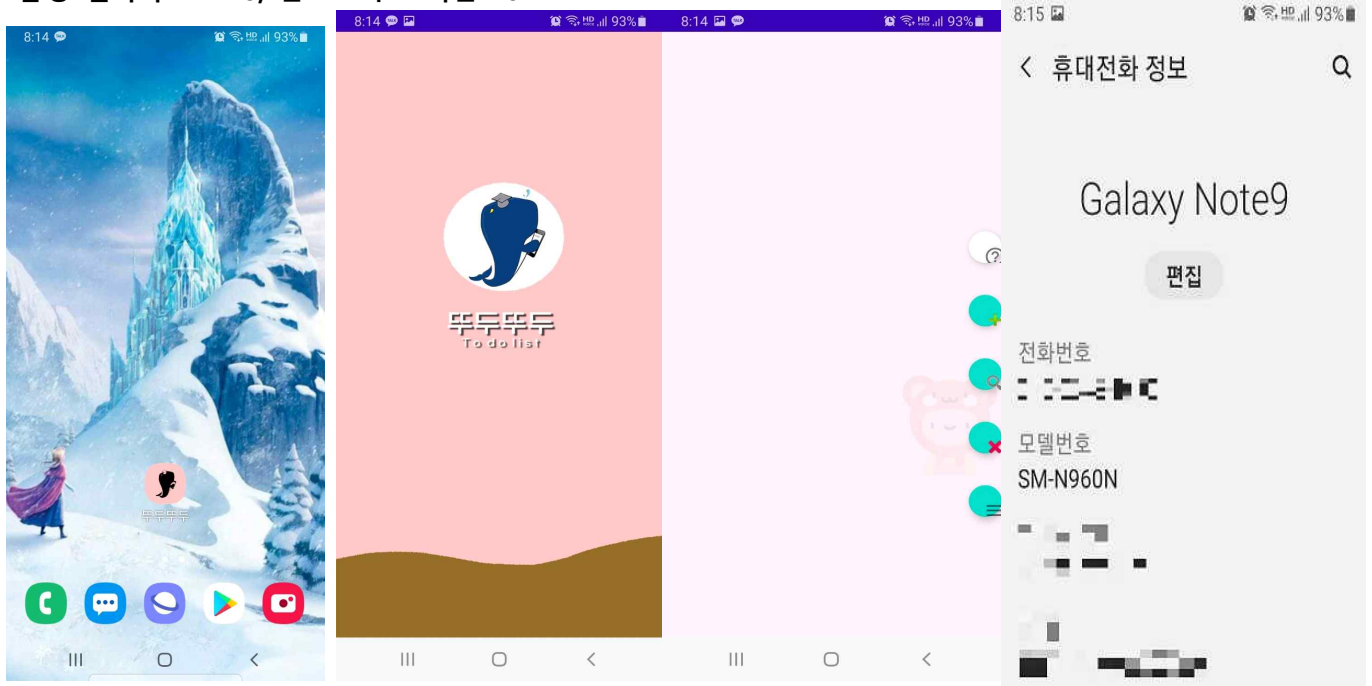
### <LG V40, 안드로이드 버전 10>



[정상적으로 작동]

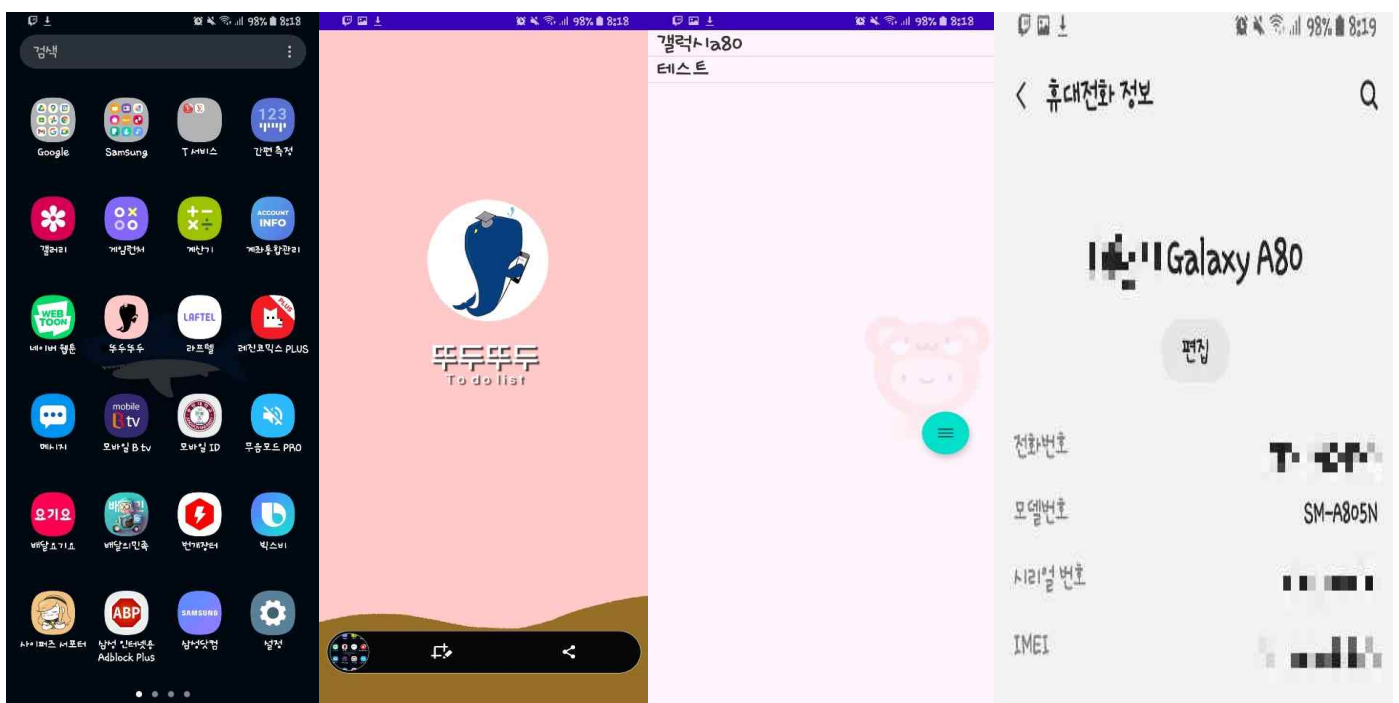


<삼성 갤럭시 노트 9, 안드로이드 버전 10>



[갤럭시 노트9 버그 :: 플로팅 액션 버튼 메뉴 구성의 이미지와 버튼이 맞지 않는 상황]

<삼성 갤럭시 A80, 안드로이드 버전10>



[정상적으로 작동]

### 3. 결론

마치면서, 본 앱("뚜두뚜두")의 한계 및 개선사항을 설명하고 느낀 점을 작성하겠습니다.

#### 3.1. 어플리케이션 한계 및 개선사항

본 앱은 사용자의 할 일 메모 즉, Todo-List를 목적으로 만들어진 SQLite 내부 데이터베이스를 활용한 앱입니다. 기본적으로는 입력, 수정, 삭제, 기능의 기본적인 모든 기능을 할 수 있으나, 사용자가 입력한 데이터의 날짜 데이터는 받아오지 못하여, 입력 화면에 달력은 기기 자체의 상단바 스크롤을 이용하여 달력을 확인 하지 않아도 화면내에서 현재의 날짜를 확인 할 수 있는 용도로만 사용이 가능합니다. 한 번의 불편함을 줄여주긴 하나, 날짜의 데이터베이스와는 연동하지 못 하였기 때문에, 언제 입력한 메모인지는 확인 할 수 없습니다.

또한, 스플래쉬(로딩) 이미지와, 런처 아이콘, 백그라운드의 이미지뷰 속 삽입 된 이미지는 벡터 이미지가 아닌, 포토샵을 이용하여 만든 비트맵(래스터) 이미지입니다. 테스팅에서는 화질이 깨지거나 하는 문제는 없었으나, 벡터 이미지를 활용 할 경우 조금 더 안정적이고 선명한 이미지를 넣을 수 있습니다.

마지막으로, 다양한 실제 스마트폰 기기를 통해 APK파일로 어플을 다운로드하여 테스팅을 진행하면서, 갤럭시 노트9에서 플로팅 액션 버튼 속 벡터 이미지 아이콘의 위치가 한쪽으로 쏠리는 현상이 발생했습니다. 앱을 만드는 안드로이드 스튜디오 환경은 개발자의 휴대폰 환경에 맞춰 최적화 되게끔 개발을 했습니다. 더 다양한 기기에서 문제 없이 해상도 별로 최적화를 한다면, 위 테스팅에서와 같은 문제점을 해결 할 수 있을 것이라 판단됩니다.

#### 3.2. 프로젝트를 진행하며 느낀점

SQLite를 활용한 "입력", "수정", "삭제", "검색" 기능은 시중에 널리 사용되는 앱들이 제공 해야하는 필수불가결적 요건입니다. 지금은 SQLite를 활용하여 내부 저장공간을 활용하였지만, 어떤 데이터베이스를 사용하던간에 모든 앱이 제공해야 하는 필수적인 요건은 아니나, 기본적으로 대부분의 어플이 사용자의 편의를 위해 제공하여야 하는 내용임은 분명합니다.

앱 개발을 함에 있어, 어떻게 하면 조금 더 사용자에게 편의를 줄 수 있을지, 어떻게 하면 디자인을 더 예쁘게 할 수 있을지 등과 같은 사용자 관점에서 더욱 생각을 깊게 할 수 있는 계기가 되었습니다. 더불어 같은 기능을 하는 타 앱보다 본 "뚜두뚜두"를 사용하며 어떤 기능이 있다면 더 좋을까? 와 같이 앱을 더 보완 할 수 있는 요소와 차별성 등에 대해 생각을 하며 다음 개발에는 이보다 발전한 앱을 만들 수 있으리란 자신과 만들고 싶다는 목표가 생겼습니다.