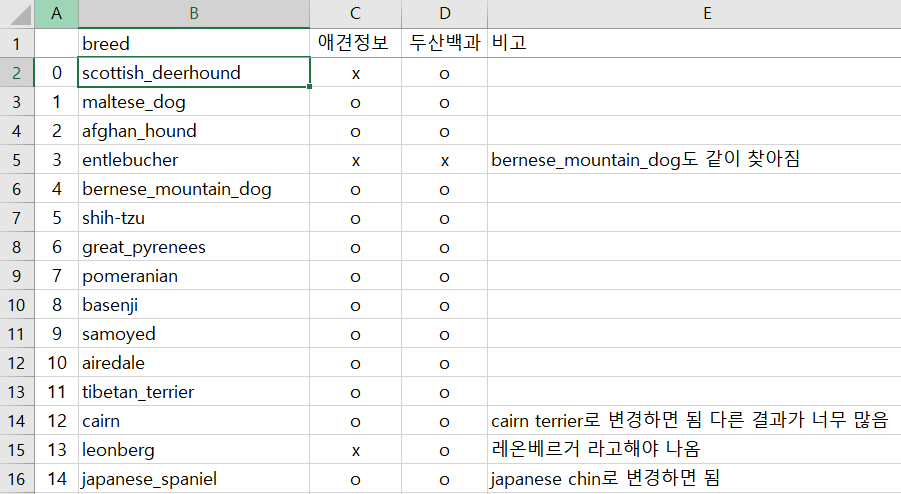
소설실 텀프로젝트 진행보고서 2

# ※ 21. 05. 16.까지의 진행 내용

## 1. API 기술 적용 및 어플리케이션 제작

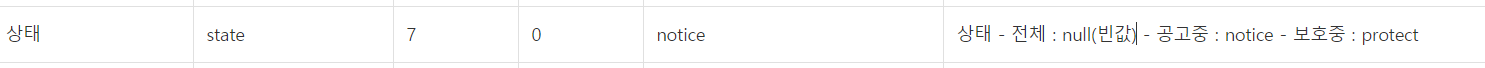
### 1. API 활용



*[그림 1] 모델 적용 후 도출되는 class 중 일부 모습*

네이버 검색을 통한 정보를 크롤링하는 것을 완료하였으나 모델 적용 후의 class명을 그대로 검색 시 결과가 안 나오는 경우가 있다는 것을 발견하였다. 사전에 문제를 해결하고자 어떤 class에서 문제가 생기는지 조사를 하였다.

그 결과 총 119가지의 견종에서 2종의 원하는 검색 결과가 나오지 않았으며 (**entlebucher, appenzeller**) 26종은 약간의 수정을 거치면 정상적으로 원하는 결과가 나오는 것을 확인하였다. 해당 2종에 대하여서는 한국애견연맹 견종 표준서의 결과를 발췌할 예정이다. 앞선 내용을 토대로, 견종의 정보를 읽어올 시 “요약”, “체고”, ”친화성” 등등의 정보를 속성으로 선정할 수 있다고 판단하였다.

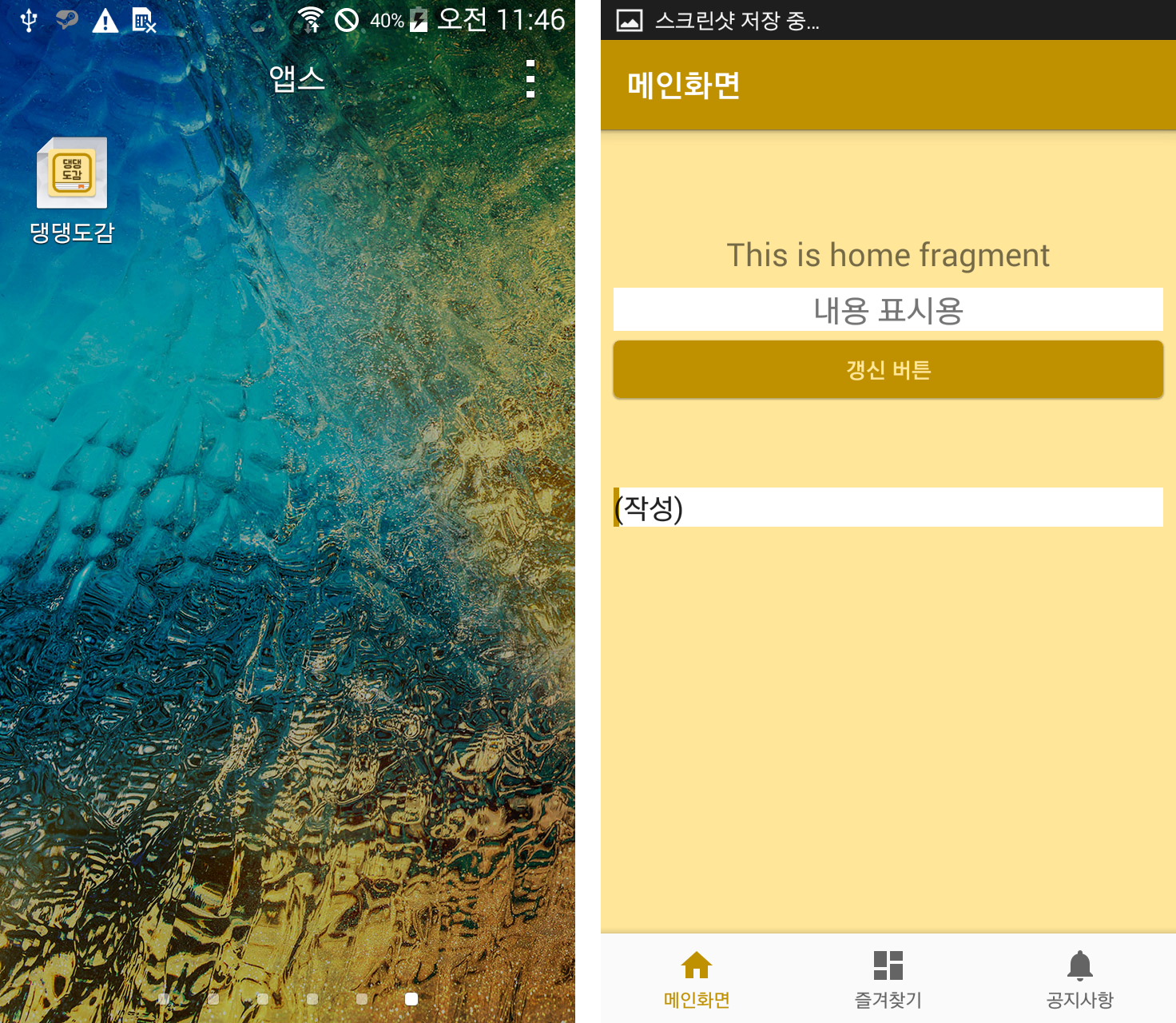


*[그림 2] 유기동물 조회 서비스에 대한 정보*

또한 유기동물 조회 서비스를 이용하여 특정 견종을 도출하지 않고 특정 기간동안 보호중인 견종을 출력하도록 계획하였다. 그 이유는 보호중인 특정 견종의 마리 수가 견종에 따라 많을 수도 혹은 적을 수도 있으나 대부분이 기타, 믹스견으로 등록 되어있는 경우가 많기 때문이다.

### 2. 어플리케이션 제작

우선, 조원들이 동시에 Application 개발을 수행해야 하므로, 이를 위해 git을 생성했다. (<https://github.com/acetyl-CoA/DaengDaengDogam>) 체계적으로 구조를 관리하는 것도 중요하지만, 여기에서는 여태까지의 개발 과정이나 자료 저장, 그리고 원활한 코드 교환을 위한 목적으로 사용하려고 한다. 따라서 통일된 양식을 추구하기는 힘들더라도, 주기적으로 개발하고 작업했던 자료, 코드들을 업로드 해서 원활한 자료 교환 및 코드 첨삭을 수행하고자 한다. 현재의 경우는 각자 구현해야 할 파트가 다르므로, 개별 branch를 생성해서 코드를 업로드 하고 있으나, 나중에 통합된 기능 구현시에는 이를 합병해서 하나의 어플로 만들어내고자 한다.



*[그림 3] ‘댕댕도감’ 어플리케이션의 간이 모습*

[그림 3]은 댕댕도감 어플 구현을 위해 간이적으로 메뉴를 설정한 모습이다. 사전에 만들어 놓은 아이콘을 메인어플에 적용하였고, 하단 **Navigation 메뉴** 바를 총 3가지 항목 (사진을 통한 견종 구분 기능, 즐겨찾기 기능, 구분 가능 견종 목록이나 개발 사항과 관련된 정보)으로 나누어서 각 페이지별로 기능을 구현할 예정이다.

## 2. Database 구조 모델링

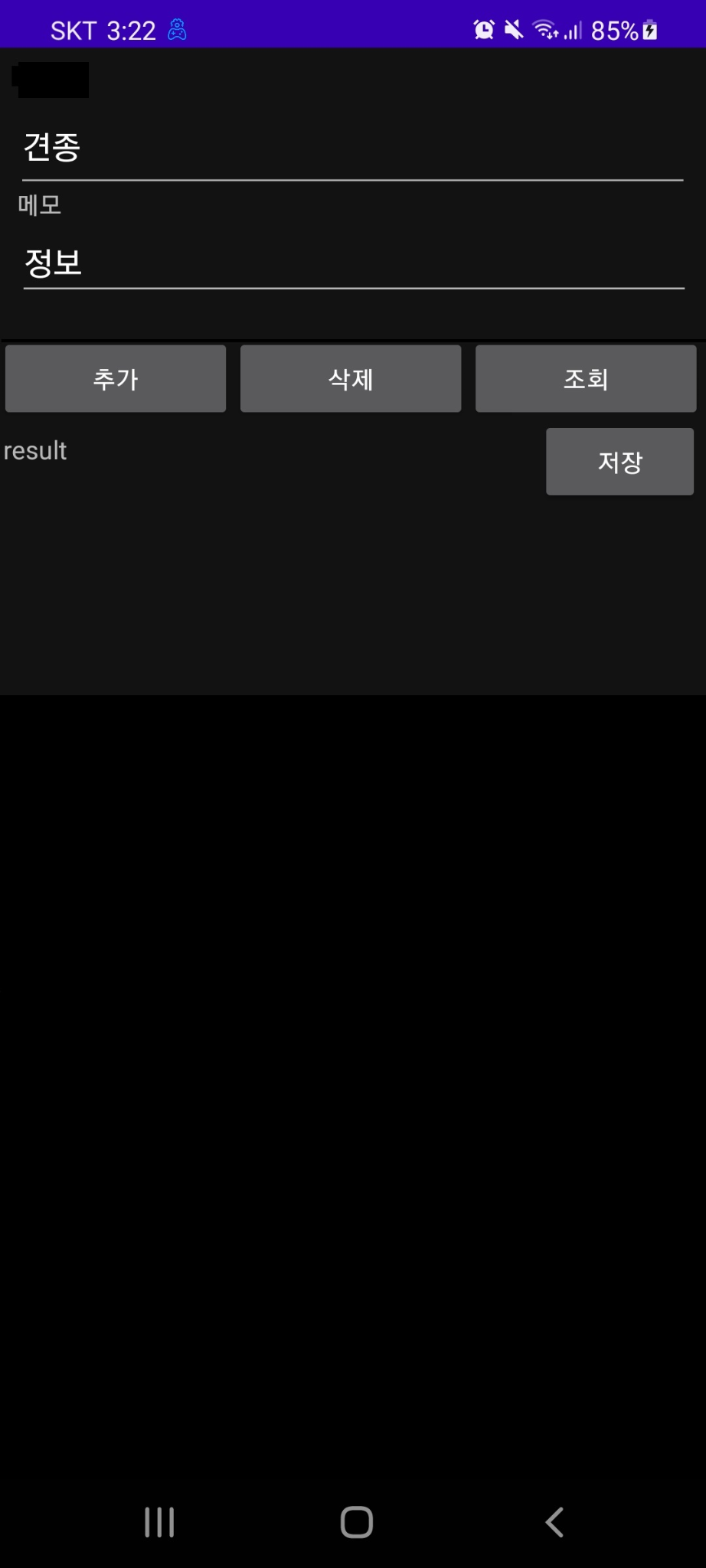
### 1. API 내용에서의 속성 선정

**SQLite를 사용**(**SQLiteDatabase**와 **SQLiteOpenHelper**를 사용)하여 데이터베이스를 생성하고 테이블을 구성하려 한다. 테이블에 레코드를 추가(**insert()**), 삭제(**delete()**)하는 기능과 메뉴에서 즐겨찾기를 눌렀을 때 테이블을 읽어오는 기능(**cursor()**)을 사용하기로 했다.

레코드가 들어온 순서대로 테이블에 추가가 되면 읽어올 때의 순서도 추가한 순서가 되는데, 테이블에서 견종에 대해 오름차순으로 정렬하는 것이 보기 좋다고 생각하여 테이블에 추가될 때마다 자동으로 정렬되도록 테이블 정렬 기능을 추가할 예정이다.

### 2. 간이적인 모델링

결과를 즉각적으로 확인할 수 있도록 기본 틀을 만들어 놓았다. 앞선 속성들을 아우르는 개체들을 다음과 같이 생각해볼 수 있겠다. [그림 4]는 기능을 틀로 만들어 놓은 예시이다.



*[그림 4] SQLite Database 기능*

## 3. 딥러닝 모델 적용

### 1. 학습된 모델 저장

기존에 학습하였던 모델 중에서 최종적으로 성능이 떨어지는 일반화된 모델 대신, (적어도 주어진 자료에 대한 구분은 가능하므로) dataset에 과적합 된 모델(Dataset에 대한 적중 예측률 약 80% 수준의 모델)을 사용하기로 결정하였다. colab 환경에서 해당 모델을 “**dog\_clsf\_model.h5**” 라는 명칭으로 저장하였고 최종적으로 이를 로컬 환경에 저장하였다. 모델의 크기가 크므로, git에 업로드 하는 대신 별도로 구글 드라이브에 해당 파일을 저장하였다. (https://drive.google.com/file/d/1mGM6-Gmm3JNa8CiqrO7FgVdrmAvnYupo/)

### 2. 딥러닝 모델 이식 방법 조사

앞선 .h5 형태로 저장한 모델을 어플리케이션에 연동하기 위해서 관련 기술을 확인해보았다. **TensorFlowLight**를 활용해서 만들어진 모델을 경량화 하고, 가벼운 모델을 통해서 어플 상에서의 예측 기능을 수행할 수 있다고 한다. 따라서, 현재 TensorFlowLight를 이용하는 방법을 배우고 있는 중이다.

# ※ 향후 개발 내용

## 1. 어플리케이션 제작

### 1. API 응용

유튜브 API를 활용한 영상 추천을 할 수 있도록 구현하고 크롤링한 정보를 토대로 레이아웃 구성을 할 예정이다.

### 2. Database 구현

SQLite Database를 이용하여 내부 저장소에 저장하고 불러온 뒤 다른 API(네이버 백과사전 검색, 유기동물 조회 서비스, 유튜브 동영상 추천)와 연동할 예정이다.

### 3. 딥러닝 모델 이식

TensorFlowLight의 활용 방식에 대한 조사가 끝난 뒤, 실제로 어플리케이션에 학습된 모델을 적용할 예정이다. 또한, 저장소 기능과의 연계를 위해서 입출력과 관련된 사항을 학습하고 조정할 생각이다.

### 4. 디자인 통일화 및 기능 통합

점차 개별적으로 구현했던 기능들을 메인 어플을 만들어가면서 통합해볼 예정이다. 이때 메인 어플리케이션의 디자인도 같이 통일화 할 계획이다.