

데이터 베이스 최종 보고서

09조

소프트웨어학과 201520882 천윤서

소프트웨어학과 201520993 정 욱

소프트웨어학과 201520998 정원중

소프트웨어학과 201521001 정찬호

1. 주제 설명

이번 프로젝트를 통해 동물 및 식물 서식지 탐색 서비스를 구현하였습니다. 이 서비스에는 두가지 기능이 있습니다. 첫 번째는 동물이나 식물의 서식지를 알려주는 것입니다. 이용자들은 동식물 관찰 및 채집, 그리고 사진 촬영을 하기위해 이 서비스를 이용할 수 있습니다. 두 번째는 검색한 동식물의 서식지와 가까운 휴양림을 추천해 주는 것입니다. 이 서비스를 통해 이용자들은 관찰, 수집, 채집 등을 하면서 가까운 휴양림에서 휴식을 취할 수 있습니다. 서비스 이용 타겟 층은 동식물에 관심이 많은 아이를 둔 부모님, 사진작가, 곤충 채집가 등이 되겠습니다.

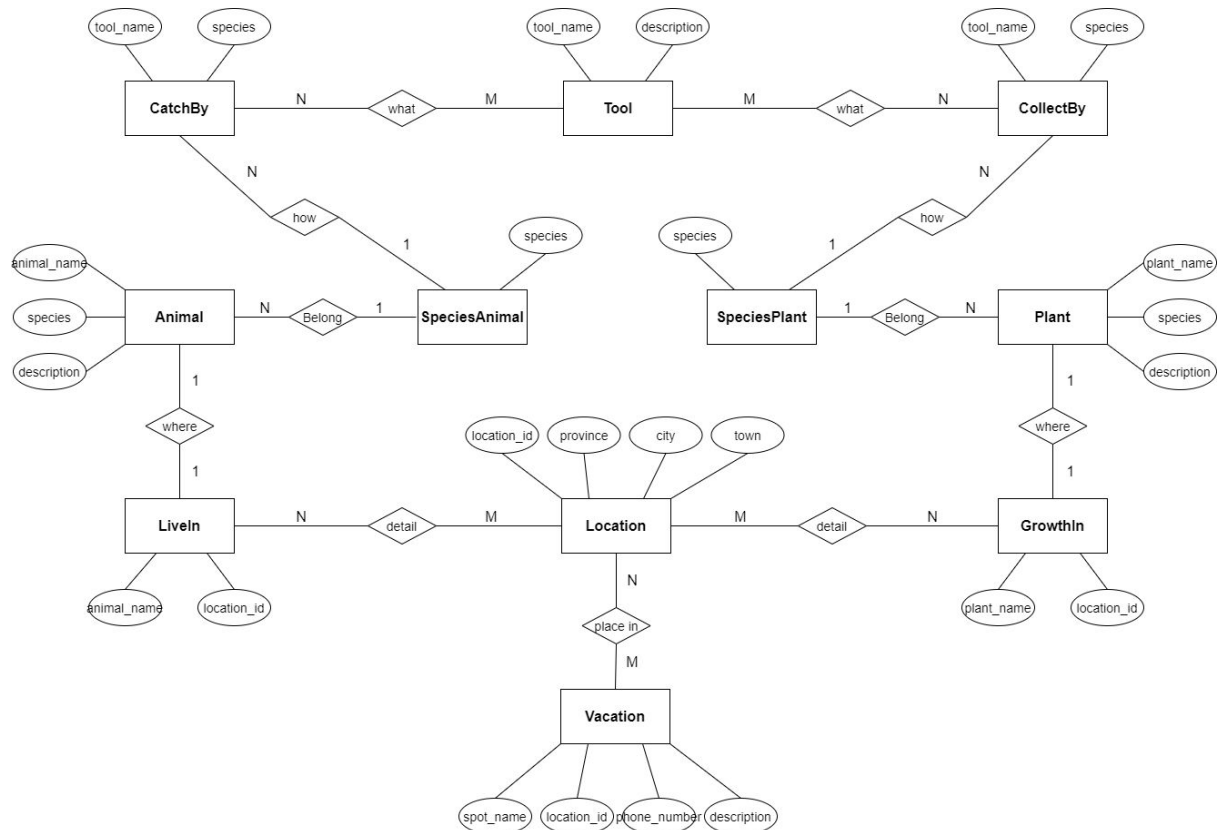
2. 주제 선정 배경

우리나라는 국토의 70% 이상이 산으로 이루어져 있습니다. 그렇기에 우리는 산속에서 다양한 종류의 동식물들과 곤충들을 만나볼 수 있습니다. 또한 산속에는 각 지역마다의 특별한 종이 존재하기도 합니다. 예를 들어 백서향이라는 식물은 희귀종이며 남부지방 해변 산기슭에만 난다고 합니다. 이렇게 산속에 있는 동식물들에게 흥미를 가진 분들은 직접 그 지역을 탐방해 사진을 찍고, 관찰을 하거나 채집을 하기도 합니다.

이전에는 각 지역에 있는 박물관 또는 체험관으로 견학을 가서 새로운 동식물들을 접했다면, 요즘은 정보의 접근성이 좋아 동식물에 관심이 있다면 검색을 통해 특정 생물들에 대한 이미지 또는 영상 등으로 접할 수 있습니다. 따라서 지역마다 어떠한 동식물 및 곤충이 있는지, 그리고 그 생물의 서식지와 가까운 휴양림은 어디에 있는지를 데이터베이스화 시킨다면 직접 그 지역을 가고자 할 때에도, 이에 대한 접근성이 한층 더 높아질 것이라고 보았습니다.

채집이나 관찰을 하러 갈때 필요한 도구들에대한 정보 또한 추가적으로 넣어준다면 도움이 될것이라고 보았습니다.

3. ER 다이어그램



[그림3-1:ER 다이어그램]

[그림3-1]은 ER 다이어그램을 나타냅니다.

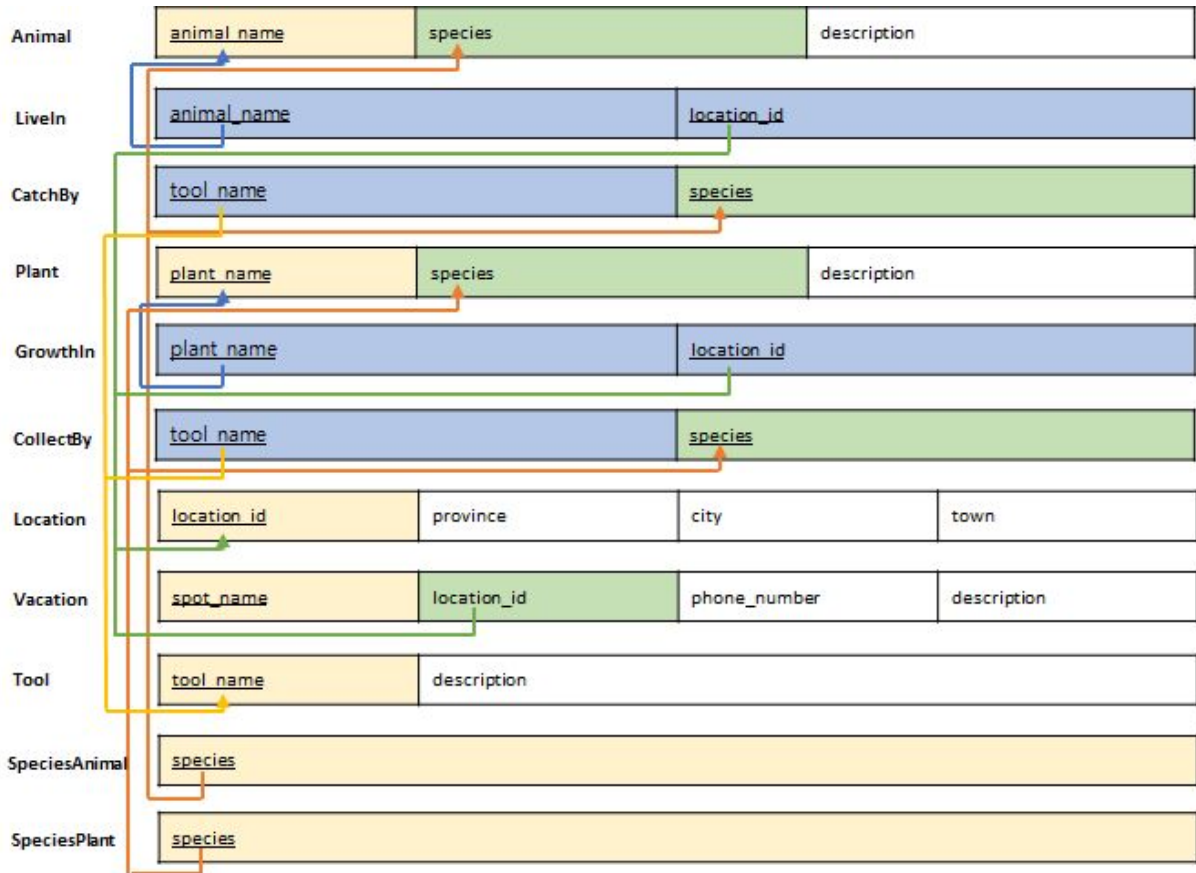
Animal은 SpeciesAnimal에 Belong이며 N:1 관계입니다. SpeciesAnimal는 Tool에 catch되며 SpeciesAnimal, Catchby, Tool의 관계는 1:N과 N:M입니다. Animal은 Location에 live하고 있으며 Animal, LiveIn, Location의 관계는 1:1과 N:M입니다.

Plant는 Animal과 비슷한 형태를 띄고 있습니다. Plant는 SpeciesPlant에 Belong이며 N:1의 관계입니다. SpeciesPlant는 Tool로 collect되며 SpeciesPlant, CollectBy, Tool의 관계는 1:N과 N:M입니다. Plant는 Location에서 Grow하고있으며, Plant, GrowthIn, Location의 관계는 1:1과 N:M입니다.

VacationSpot은 Location에 Place하고 있으며 N:M 관계를 가지고 있습니다.

이번 기말 프로젝트에서 동물과 식물은 두 종에 대해 흥미를 가지는 사람도 크게 다르고, 추가적인 특성들 또한 다른 모습을 많이 보입니다. 따라서 동물과 식물이 가진 관계와 각 속성들이 유사함에도 불구하고, 추후의 확장성을 고려해보았을 때 분리하는 것이 맞다고 생각했습니다. 그래서 이번 프로젝트(데모)에서 우리는 동물과 식물을 서로 분리하여 ER다이어그램을 작성하였습니다. 해당 ER 다이어그램을 통해서 데이터스키마를 생성했는데 이는 다음 장에서 서술합니다.

4. 데이터스키마



[그림4-1] 데이터 스키마

[그림3-1]의 데이터 스키마는 프로젝트 구현 시나리오에 따라서 작성되었습니다. 이에 대한 상세 설명은 아래와 같습니다.

Animal(동물): 동물에 관련된 정보를 지니고 있는 스키마입니다. 기본키는 동물 이름(animal_name)이며 외부키는 종(species)입니다. 나머지는 상세설명(description)을 가지고 있습니다.

LiveIn(동물 서식지): 동물의 서식지에 관련된 정보를 지니고 있는 스키마입니다. 동물 이름(animal_name)이 animal에서 서식지(location_id)는 location에서 직접 참조하고있습니다.

CatchBy(채집대상): 동물의 채집과 관련된 정보를 지니고 있는 스키마입니다. 도구 이름(tool_name)을 tool에서 직접 참조하며 외부키는 종(species)입니다.

Plant(식물): 식물에 관련된 정보를 지니고 있는 스키마입니다. 기본키는 식물 이름(plant_name)이며 외부키는 종(species)입니다. 나머지는 상세설명(description)을 가지고 있습니다.

GrowthIn(식물 서식지): 식물의 서식지에 관련된 정보를 지니고 있는 스키마입니다. 식물 이름(plant_name)이 plant에서 서식지(location_id)는 location에서 직접 참조하고있습니다.

CollectBy(채집대상): 식물의 채집과 관련된 정보를 지니고 있는 스키마입니다. 도구 이름(tool_name)을 tool에서 직접 참조하며 외부키는 종(species)입니다.

Location(지역): 지역과 관련된 정보를 나타내고 있는 스키마입니다. 기본키는 location_id입니다. 나머지는 도(Province) 시군구(City) 읍면동(Town)으로 이루어져있습니다.

Vacation(휴양림): 휴양림에 관한 정보를 가지고 있는 스키마입니다. 기본키는 휴양림 이름(spot_name)입니다. 나머지는 지역(location_id)과 전화번호(phone_number)와 상세설명(description)을 가지고 있습니다.

Tool(채집도구): 채집도구에 관련된 정보를 지니고 있는 스키마입니다. 기본키는 도구 이름(tool_name)입니다. 나머지는 상세설명(description)을 가지고 있습니다.

SpeciesAnimal(동물 종): 동물의 종에 관련된 정보를 지니고 있는 스키마입니다. 기본키는 종 이름(species)입니다.

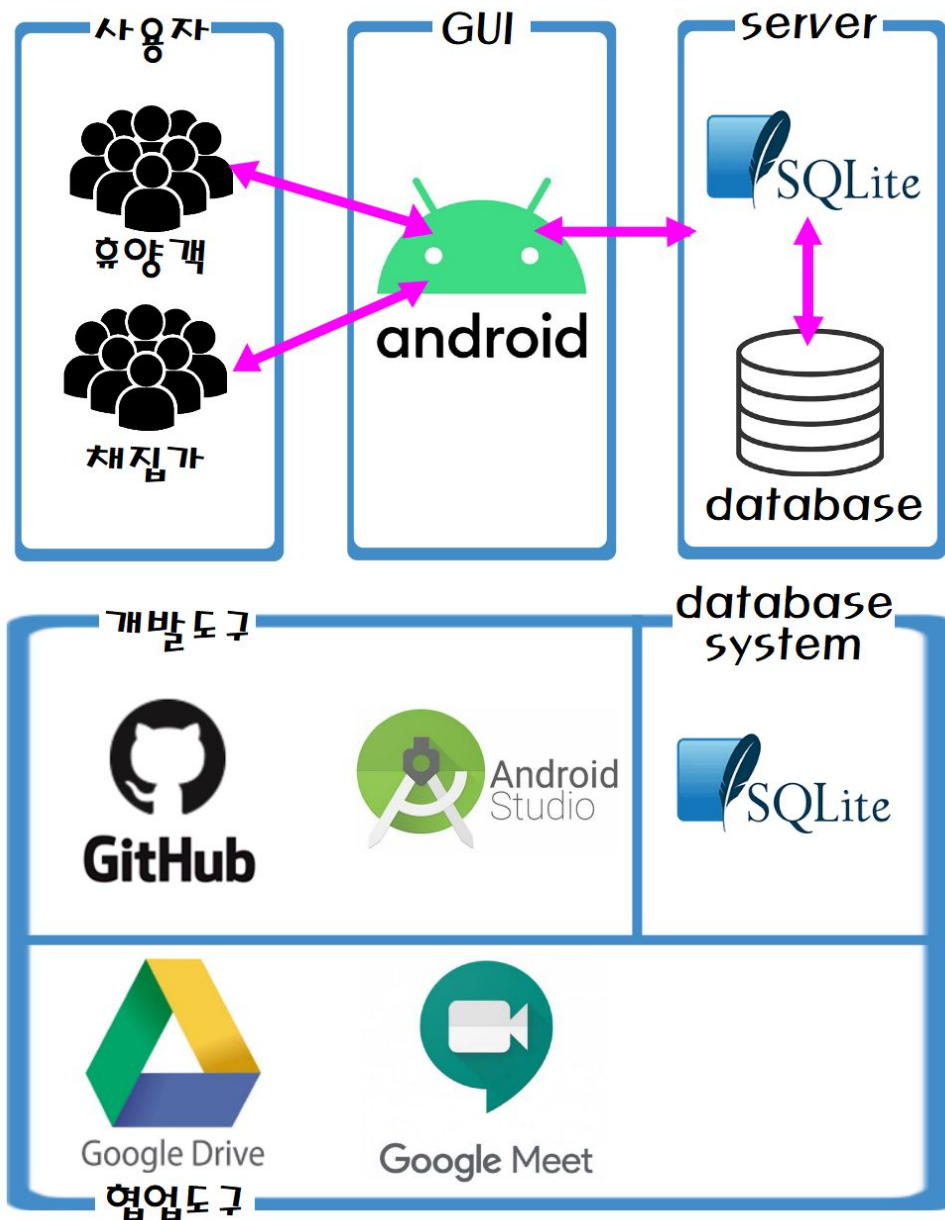
SpeciesPlant(식물 종): 식물의 종에 관련된 정보를 지니고 있는 스키마입니다. 기본키는 종 이름(species)입니다.

일부 테이블의 생성 SQL 문을 보자면, 아래 [표4-1]과 같습니다.

| | |
|--|--|
| <pre>CREATE TABLE "Animal" ("animal_name" TEXT NOT NULL, "species" TEXT, "description" TEXT, FOREIGN KEY("species") REFERENCES "SpeciesAnimal"("species"), PRIMARY KEY("animal_name")); CREATE TABLE "CatchBy" ("tool_name" TEXT, "species" TEXT, FOREIGN KEY("tool_name") REFERENCES "Tool"("tool_name"), FOREIGN KEY("species") REFERENCES "SpeciesAnimal"("species"), PRIMARY KEY("tool_name", "species")); CREATE TABLE "LiveIn" ("animal_name" TEXT, "location_id" INTEGER, PRIMARY KEY("location_id", "animal_name"), FOREIGN KEY("animal_name") REFERENCES "Animal"("animal_name"), FOREIGN KEY("location_id") REFERENCES "Location"("location_id"));</pre> | <pre>CREATE TABLE "Location" ("location_id" INTEGER NOT NULL, "province" TEXT, "town" TEXT, "city" TEXT, PRIMARY KEY("location_id")); CREATE TABLE "SpeciesAnimal" ("species" TEXT, PRIMARY KEY("species")); CREATE TABLE "Tool" ("tool_name" TEXT, "description" TEXT, PRIMARY KEY("tool_name")); CREATE TABLE "Vacation" ("spot_name" TEXT NOT NULL, "phone_number" TEXT, "description" TEXT, "location_id" INTEGER, PRIMARY KEY("spot_name"), FOREIGN KEY("location_id") REFERENCES "Location"("location_id"));</pre> |
|--|--|

[표4-1: 데이터베이스의 테이블 생성 명령어]

5. 사용 기술 스택



[그림5-1] 기술 스택

[그림5-1]의 기술 스택은 저희 프로젝트에서 진행할 때 사용되는 개발 기술들과 협업 도구들을 나타냅니다.

사용자(휴양가 및 채집가)는 안드로이드 앱 어플리케이션을 통해서 상호작용을 하며, 안드로이드 앱 어플리케이션은 SQLite의 데이터베이스를 기반으로 동작합니다.

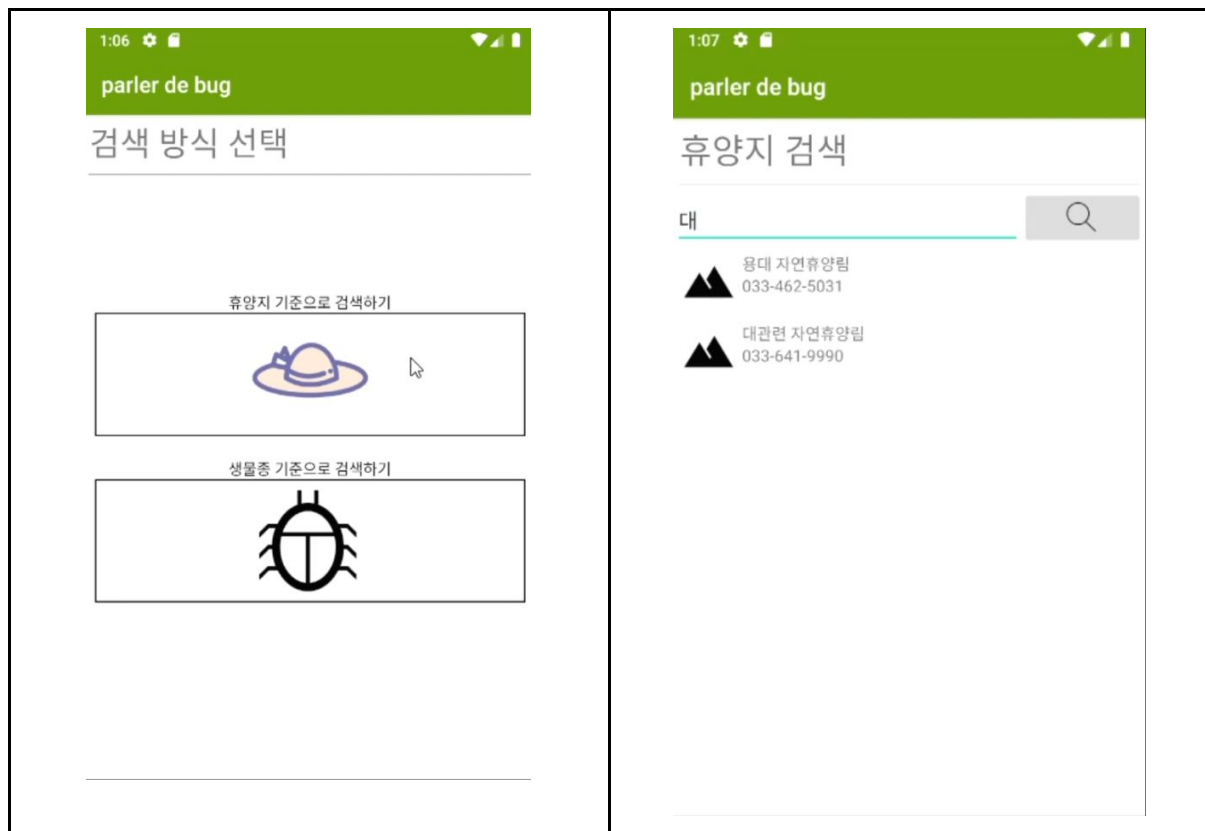
개발 도구로는 Android Studio를 사용하며 데이터베이스로는 SQLite, 버전 관리는 GitHub을 통해서 합니다. Google Meet과 클라우드를 통해서 회의와 문서관리를 진행합니다.

이 외에도 크롤링과 데이터 처리작업을 위해서 Python과 Selenium이 사용되었습니다.

6. 프로젝트의 독창성

국가생물종지식정보시스템(<http://www.nature.go.kr/main/Main.do>) 에서는 우리나라에 있는 모든 생물종을 확인을 할 수 있습니다. 학술적인 정보를 담고 있는 사이트 이지만 어디에 서식을 하고있는지 내용이 나오지 않아 불편하다는 단점이 있습니다. 또한 국립생물자원관의 생물지리정보(<https://species.nibr.go.kr/geo/html/index.do>) 에서는 해당 생물종이 어디에 사는지는 알 수 있지만 해당 지역에서 볼 수 있는 생물종에 대한 정보가 약간 미흡한 것으로 보였습니다. 저희 프로젝트는 관심있는 동식물의 정보를 알아보고 이를 직접 찾아가서 관찰하게 하는 것이 목적입니다. 직접 찾아가는 것이 부담스러울 뿐만 아니라, 체력적으로도 문제가 될 수 있어서, 이를 자연 휴양림 데이터와 연동을 지어보게 되었습니다. 이에따라 생물종의 정보와 지역정보, 그리고 자연 휴양림의 정보를 데이터베이스화 하여 저장을 하였고, 이를 가지고 자연 생물종과 휴양림을 연관지어 프로젝트를 진행하게되었습니다. 이것으로 생물종을 검색하여 그 생물종이 서식하는 자연 휴양림을 찾는 기능과 휴양림을 검색하여 해당 휴양림에서 볼 수 있는 생물종을 찾을 수 있게 구현하였습니다. 이외에도 채집 도구에 대한 정보를 넣음으로 해당 생물종을 채집 할 때 필요한 도구정보까지 알 수 있게 만들었습니다. 저희 조는 이러한 시스템이 지금까지 특별히 알려진 것이 없다고 생각하였고 그리하여 이것을 주제로 선정하였습니다.

7. 프로젝트의 실행 결과 및 시나리오



[그림 7-1: 메인화면 선택(좌)/휴양지 검색(우)]

시나리오는 중간 보고서에서 설명한 것과 거의 유사합니다. 먼저 사용자는 휴양지 기준으로 검색을 할지, 생물종 기준으로 검색할지 결정합니다. 이 경우에는 휴양지 기준으로

검색합니다. [그림7-1]의 오른쪽은 검색 키워드로 “대”를 주었을 때를 나타냅니다. 옹”대”와 “대”관련 휴양림이 검색되었습니다. 이때 사용한 sqlite의 쿼리는 아래와 같습니다.

```
select * from Vacation WHERE spot_name like "%대%"
```

그 다음에는 리스트에서 원하는 휴양지를 클릭해서 해당 휴양지에 관한 상세정보와 관련된 식물이나 동물의 리스트를 가져올 수 있습니다. 이 때 휴양지는 지역 정보를 location_id로만 가지고 있으므로, 쿼리가 또 필요합니다. 사용하는 쿼리는 아래와 같습니다.

```
SELECT * FROM Location WHERE location_id IN (SELECT location_id FROM Vacation WHERE spot_name="대관련 자연휴양림")
```

지역 정보를 이용해서 식물이나 동물의 리스트를 가져오는 쿼리는 아래와 같습니다.

```
SELECT * FROM PLANT WHERE plant_name IN (SELECT plant_name FROM GrowthIN WHERE location_id IN (SELECT location_id FROM Location WHERE province = "강원도" AND town = "강릉시" AND city = "견소동"))
SELECT * FROM Animal WHERE animal_name IN (SELECT animal_name FROM LiveIn WHERE location_id IN (SELECT location_id FROM Location WHERE province = "강원도" AND town = "강릉시" AND city = "강문동"))
```

| 휴양지 정보 | 생물 리스트 |
|---|---|
| <p>휴양지 이름</p> <p>대관련 자연휴양림</p> <p>전화번호</p> <p>033-641-9990</p> <p>지역</p> <p>강원도 강릉시 성산면</p> <p>상세설명</p> <p>1988년 전국 최초로 조성된 자연휴양림으로 50년~200년생 아름다운 소나무숲이 있는 자연휴양림입니다.</p> | <p>각다귀KUa 곤충류</p> <p>강변거저리 곤충류</p> <p>검은푸른뱀기나방 곤충류</p> <p>검정날개각다귀KUa 곤충류</p> <p>금오개미시늉거미 무척추동물류(곤충제외)</p> <p>다슬기 무척추동물류(곤충제외)</p> <p>동자개 어류</p> <p>실지렁이 무척추동물류(곤충제외)</p> <p>주름다슬기</p> |
| <p>상세 정보 조회</p> <p>서식 동물 정보</p> <p>서식 식물 정보</p> | <p>상세 정보 조회</p> <p>서식 동물 정보</p> <p>서식 식물 정보</p> |

[그림7-2: 휴양지 정보(좌)/서식 동물 리스트(우)]

[그림7-2]는 해당 정보를 이용해서 휴양지의 정보와 서식 동물 리스트를 가져온 어플리케이션의 화면을 보여줍니다.

왼쪽은 식물 '난'을 검색하는 것을 보여줍니다. 오른쪽은 이를 클릭했을 때 보여주는 식물 정보입니다. 각각에 쓰이는 쿼리는 다음과 같습니다.

```
select * from plant where plant_name like "%난%"
select * from Animal WHERE animal_name like "%파리%"
```

마찬가지로 식물이나 동물 또한 지역은 아이디 형태로 지니기에 따로 끌어오는 쿼리가 필요합니다.


```
SELECT * FROM Location WHERE location_id IN (SELECT location_id FROM GrowthIn
WHERE plant_name = "은대난초")
SELECT * FROM Location WHERE location_id IN (SELECT location_id FROM LiveIn WHERE
animal_name = "버섯뿔파리")
```

마지막으로 식물이나 동물 상세 정보 조회 후에 휴양지 리스트를 조회할 수 있고 채집에 쓰이는 도구를 볼 수 있습니다. 휴양지 리스트 조회에 쓰이는 쿼리는 아래와 같습니다.

```
SELECT * FROM Vacation WHERE location_id IN (SELECT location_id FROM GrowthIn
WHERE plant_name="은대난초")
SELECT * FROM Vacation WHERE location_id IN (SELECT location_id FROM LiveIn WHERE
animal_name="버섯뿔파리")
```

채집 도구를 보는데 사용하는 쿼리는 아래와 같습니다. 또한, [그림7-5]는 각각 휴양지 리스트 조회와 관련 채집 도구 조회를 보여줍니다.

```
SELECT * FROM Tool WHERE tool_name IN (SELECT tool_name FROM CollectBy WHERE
species IN (SELECT species FROM Plant WHERE plant_name ="은대난초"))
SELECT * FROM Tool WHERE tool_name IN (SELECT tool_name FROM CatchBy WHERE
species IN (SELECT species FROM Animal WHERE animal_name ="버섯뿔파리"))
```

| | |
|--|---|
| <div data-bbox="266 1288 531 1326">관련 휴양지 리스트</div> <div data-bbox="276 1344 451 1388">  <div data-bbox="331 1344 451 1388"> 대관령 자연휴양림 033-641-9990 </div> </div> <div data-bbox="485 1561 504 1588">  </div> <div data-bbox="293 1890 394 1912">상세 정보 조회</div> <div data-bbox="432 1890 549 1912">관련 리스트 조회</div> <div data-bbox="587 1890 689 1912">채집 도구 보기</div> | <div data-bbox="869 1283 1198 1321">생물종 채집 도구 리스트</div> <div data-bbox="879 1341 1182 1388">  <div data-bbox="935 1341 1182 1388"> 모종삽 작은 식물이나 모종, 버섯등을 캐는 도구 </div> </div> <div data-bbox="896 1883 997 1906">상세 정보 조회</div> <div data-bbox="1034 1883 1150 1906">관련 리스트 조회</div> <div data-bbox="1187 1883 1287 1906">채집 도구 보기</div> |
|--|---|

[그림7-5: 휴양지 리스트 조회(좌)/관련 채집 도구 조회(우)]

어플리케이션에서 사용한 튜플의 개수는 아래 [표 7-1]와 같습니다.

| Animal | CatchBy | CollectBy | GrowthIn | LiveIn | Location | Plant |
|--------|---------|-----------|----------|--------|----------|-------|
| 1,040 | 4 | 11 | 395 | 2,425 | 341 | 331 |

| SpeciesAnimal | SpeciesPlant | Tool | Vacation |
|---------------|--------------|------|----------|
| 9 | 13 | 5 | 12 |

[표 7-1] 사용한 튜플 개수

사용한 튜플은 4,586개이며, Tool, Vacation은 가상의 데이터를, 나머지는 국립생물자연관의 공식 홈페이지를 Selenium과 Python을 통해 크롤링하고, Python을 통해서 후처리를 하였습니다.

해당 튜플 데이터들과 후처리 코드들, 개발된 안드로이드 어플리케이션은 보고서의 분량이 넘는 관계로 대표 학생의 GitHub 링크를 통해서 공유합니다.

안드로이드와 sqlite로 개발된 메인 어플리케이션 코드:

https://github.com/chunyunseo/2020_AjouUniv_Database_09_parlerdebug_Main_Application

크롤링과 파싱을 하는데 사용된 파이썬 코드:

https://github.com/chunyunseo/2020_AjouUniv_Database_09_parlerdebug_Crawling-Processing