


캡스톤 디자인 I 종합설계 프로젝트

프로젝트 명	모아요 (MoaYo)
팀 명	고리고리
문서 제목	중간보고서

Version	1.2
Date	2020-04-23

팀원	맹 산하 (팀장)
	강 길웅
	김 사라
	이 정현
	정 준권
지도교수	김 혁만

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	모아요 (MoaYo)	
	팀 명	고리고리	
	Confidential Restricted	Version 1.2	2020-04-23

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "모아요"를 수행하는 팀 "고리고리"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "고리고리"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	MoaYo-중간보고서.doc
원안작성자	맹산하, 이정현
수정작업자	맹산하, 강길웅, 김사라, 이정현, 정준권

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-04-20	맹산하	1.0	최초 작성	최초 작성
2020-04-22	정준권	1.1	문서 수정	전체 문서 내용 수정
2020-04-23	전체	1.2	문서 수정	전체 문서 내용 수정

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

내용

1. 프로젝트 목표	4
2. 수행 내용 및 중간결과	5
2.1. 계획서 상의 연구내용	5
2.2. 수행내용	8
2.2.1. 구조도 설계	8
2.2.2. 모바일 프론트엔드 진행상황	9
2.2.3. 모바일 백엔드 진행상황	13
2.2.4. 서버 진행상황	17
3. 수정된 연구내용 및 추진 방향	18
3.1. 수정사항	18
3.1.1. 모바일 백엔드 변경 사항	18
3.1.2. 검색 서버 변경 사항	21
3.1.3. 공유 서버 변경 사항	21
4. 향후 추진계획	22
4.1. 향후 계획의 세부 내용	22
4.1.1. 모바일 프론트엔드 향후 계획	22
4.1.2. 모바일 백엔드 향후 계획	23
4.1.3. 검색 서버 향후 계획	24
4.1.4. 공유 서버 향후 계획	25
5. 고충 및 건의사항	26
5.1. 개발분야 별 고충 및 건의사항	26

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	모아요 (MoaYo)	
	팀 명	고리고리	
	Confidential Restricted	Version 1.2	2020-04-23

1. 프로젝트 목표

프로젝트 ‘모아요’ (‘MoaYo’) 는 사용자에게 인스타그램 태그 검색에 대해 보다 정교한 검색 기능을 간편하게 제공하는 것을 목표로 한다.

이는 사용자가 직접 만든 카테고리, 즉 계층 (Hierarchy) 구조로 게시물을 검색하게 함으로써 이루어지는데, 계층 구조로 분류된 검색 결과는 단일 태그 검색에 비해 정확하고 체계적인 검색을 가능하게 한다.

계층구조 검색을 통해 최신 정보들을 실시간으로 가져오며, 이러한 카테고리 형태의 계층구조를 우리는 ‘도감’이라 정의한다.

즉, 사용자에게 의해 ‘도감’을 생성/관리하고, 검색하는 서비스를 하나의 플랫폼으로 제공하며, 타 사용자와 공유할 수 있는 퍼블리싱(Publishing) 기능까지 제공하는 것이 해당 프로젝트의 목표이다.

1. 검색 목표


- A. 사용자가 작성한 카테고리에서 메타 검색을 위한 유의어 크롤
- B. 카테고리화 얻어낸 유의어를 가지고 인스타그램 내부 게시물 크롤
- C. 얻어낸 게시물을 사용자에게 제공

2. 카테고리 목표

- A. 사용자가 입력한 태그를 통한 계층 구조의 도감 생성
- B. 생성된 도감의 수정 및 삭제
- C. 도감을 통해 검색된 게시물을 저장

3. 공유 목표

- A. 사용자의 도감을 공유 서버 내부에 저장
- B. 공유 서버 내부 정보를 사용자에게 제공

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

2. 수행 내용 및 중간결과

2.1. 계획서 상의 연구내용

크게 모바일 프론트엔드, 모바일 백엔드, 서버 세 가지로 구분한다.

모바일 프론트엔드는 앱의 UI/UX 및 이벤트 핸들링에 대한 처리와 사용자에게 제공할 인터페이스를 개발한다.

모바일 백엔드는 프론트엔드로 부터 받아온 데이터를 처리할 로직의 개발과 데이터의 저장 등 내부 로직에 대한 개발을 담당한다.

서버는 크게 검색 서버와 공유 서버 두가지로 분류한다.

1) 모바일 프론트엔드

모바일 프론트엔드는 위에 서술된 검색 목표, 카테고리 목표, 공유 목표에 해당하는 UI/UX에 관련된 작업을 실시한다. 또한, 사용자를 통해 발생하는 이벤트를 핸들링하고, 해당 이벤트를 통해 백엔드와 통신하는 것을 목표로 한다.


A. 검색 목표

- 사용자가 도감을 클릭하면 발생한다. 사용자가 원하는 계층에 해당하는 태그들을 JSON 형태로 변환해주고, 백엔드에 해당 내용을 전달하면서 게시물을 요청한다.
- 백엔드 로직을 통해 얻어낸 게시물을 Recycler view를 이용한 Card View형태로 제공한다.

B. 카테고리 목표

- 사용자가 입력한 태그를 통한 계층 구조의 도감 생성
 - 사용자가 태그들을 입력창을 통해 입력할 수 있고, 입력한 태그를 클릭하여 태그의 다음 계층으로 들어갈 수 있도록 함으로써 순차적으로 도감의 계층을 입력할 수 있도록 유도하는 것을 목표로 한다.
 - 사용자가 생성을 완료하면, 해당 태그들을 카테고리 형태를 가진 도감으로 생성하고 해당 내용을 백엔드에 전달하는 것을 목표로 한다.
- 생성된 도감의 수정 및 삭제
 - 생성된 도감을 수정하고 삭제할 수 있는 기능을 제공하는 것을 목표로 한다. 생성된 도감의 구조를 확장 패널을 사용한 트리 뷰 형태로 제공하며, 수정 및 삭제 버튼을 이용하여 수정 및 삭제할 수 있도록 하는 것을 목표로 한다.
 - 수정은 초기 생성을 위한 화면으로 전환되고, 해당 입력창에는 도감의 내용들이 채워져 있는 것을 목표로 한다.
 - 수정 및 삭제된 도감의 정보를 백엔드에 전달하는 것을 목표로 한다.
- 도감을 통해 검색된 게시물 저장
 - 도감의 각 계층에 관련된 게시물을 저장하는 것을 목표로 한다. 더블 클릭 등 여러 이벤트를 사용하여 저장할 수 있는 기능을 목표로 한다.
 - 저장된 게시물에 대한 정보를 백엔드에 전달하는 것을 목표로 한다.

C. 공유 목표

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

- i. 사용자의 도감을 서버 내부에 저장
 1. 사용자가 생성한 도감을 3X3의 Circle Image View를 통해 해당 도감의 대표 이미지와 설명을 보여주는 것을 목표로 한다.
 2. 사용자는 도감을 클릭하여 공유를 할 경우, 해당 도감의 정보를 작성할 수 있는 화면을 보여주고, 작성된 내용을 백엔드에 전달하는 것을 목표로 한다.
- ii. 공유 서버 내부의 정보를 사용자에게 저장
 1. 백엔드에서 전달받은 공유 서버의 도감들을 Recycler View를 이용하여 보여주고, Card View를 통해 해당 이미지를 보여주는 것을 목표로 한다.

2) 모바일 백엔드

모바일 백엔드는 위에 서술된 검색 목표, 카테고리 목표, 공유 목표에 해당하는 비즈니스 로직과 해당 데이터를 데이터베이스 및 서버에 요청을 보내는 작업을 실시한다. 비즈니스 로직은 프론트엔드 및 서버에서 받아온 데이터에 대한 유효성을 검사하고, 알맞은 형태로 변환시키는 것을 목표로 한다.

A. 검색 목표


- i. 사용자가 작성한 카테고리에서 메타 검색을 위한 유의어 크롤
 1. 작성한 카테고리에 해당하는 태그명을 태그와 관련된 유사어를 찾아주는 사이트를 크롤링하여 정보를 얻어내는 것을 목표로 한다.
- ii. 카테고리화 얻어낸 유의어를 가지고 인스타그램 내부 게시물 크롤
 1. 카테고리화 얻어낸 유의어를 JSON 형태로 변환하여 검색 서버에 정보를 보내는 것을 목표로 한다. 이 작업은 Retrofit을 이용한 HTTP통신하고, 쓰레드를 이용하여 비동기 처리하는 것을 목표로 한다.

B. 카테고리 목표

- i. 사용자가 입력한 태그를 통한 계층 구조의 도감 생성
 1. 사용자가 입력한 각각의 카테고리를 하나의 도감으로 합치고, 이를 도감과 카테고리 테이블을 나누어 저장하는 것을 목표로 한다.
- ii. 생성된 도감의 수정 및 삭제
 1. 사용자가 수정한 도감에 대해 데이터베이스의 데이터와 비교하여, 존재하는 경우 수정된 도감을 업데이트하는 것을 목표로 한다.
 2. 사용자가 삭제한 도감에 대해 데이터베이스의 데이터와 비교하여, 존재하는 경우 해당 도감을 삭제하는 것을 목표로 한다.
- iii. 도감을 통해 검색된 게시물 저장
 1. 사용자가 저장을 원하는 게시물에 대해 해당 게시물의 정보와 관련된 도감, 카테고리의 정보를 담은 테이블에 저장하는 것을 목표로 한다.

C. 공유 목표

- i. 사용자의 도감을 공유 서버 내부에 저장
 1. 사용자가 공유를 요청한 도감에 대해 해당 도감을 JSON형태로 변환하여 공유 서버에 보내는 것을 목표로 한다. 이 작업은 Retrofit을 이용한 HTTP 통신을 수행하고, 쓰레드를 이용하여 비동기 처리하는 것을 목표로 한다.
- ii. 공유 서버 내부 정보를 사용자에게 제공
 1. 공유 서버에서 받아온 정보를 일치하는 형태로 변환시켜서 프론트엔드에 전달하는 것을 목표로 한다. 이 작업은 Retrofit을 이용한 HTTP 통신을 수행하고, 쓰레드를 이용하여 비동기 처리하는 것을 목표로 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	모아요 (MoaYo)	
	팀 명	고리고리	
	Confidential Restricted	Version 1.2	2020-04-23

3) 서버

서버는 크게 두종류의 API를 제공하는 것을 목표로 한다.

인스타그램 크롤링 및 메타 검색을 수행하는 검색 서버와 사용자 간 도감을 공유하는 공유 서버를 두고 운용한다.

1. 검색서버

인스타그램에서 제공하는 API는 한번에 얻을 수 있는 게시물의 갯수가 33개로 한정되어 있고, 검색어도 하나밖에 넣을 수 없기 때문에 정교한 검색 결과를 도출하기 위해서는 인스타그램 크롤러가 필요하다. 단, 실시간으로 검색 결과를 보여주기 위해 따로 데이터베이스는 두지 않는다.

인스타그램 게시물들을 크롤링하고 메타 검색 로직을 적용하여 정확하고 체계적으로 분류된 검색 결과를 모바일 백엔드에 제공한다. 요청은 계층 별 검색어와 해당 검색어의 동의어 목록이 담긴 http request이다. 데이터의 교환은 JSON형식을 기본으로 하며 필요한 경우 Header에 값을 포함시켜 보낸다. 모바일 백엔드에 API를 제공하는 것이 목적이므로 모바일 앱과 독립되도록 한다.

검색 서버는 모바일 백엔드에 검색의 기능을 제공하는 단일 목표를 가진다.

A. 검색 목표

- I. 카테고리화 얻어낸 유의어를 가지고 인스타그램 내부 게시물 크롤
 1. 모바일 백엔드가 검색 서버로 검색할 태그들과 태그들의 연관 태그를 전달하면, 해당 데이터를 기반으로 인스타그램에서 게시물들을 크롤링하고, 태그와 연관 태그들의 교집합에 속하는 게시물들을 얻어내 모바일 백엔드에게 제공함을 목표로 한다.


2. 공유 서버

공유서버는 모바일 백엔드에서 보낸 공유 요청 및 공유된 도감 요청에 대해 서버 내부 DB를 이용하여 데이터를 저장 및 제공함을 목표로 한다. 모바일 백엔드로부터 받은 데이터를 가공하여 저장하고, 저장된 데이터를 JSON의 형태로 재 가공하여 백엔드에게 제공한다.

공유서버는 공유의 기능만을 가지고 있기 때문에 공유 목표에 대한 세부 목표만을 가진다.

A. 공유 목표

- I. 사용자의 도감을 공유 서버 내부에 저장
 1. 백엔드가 공유서버 API를 통해 보낸 JSON 데이터를 파싱하여 엔티티 객체에 매핑해 이용 가능하도록 만드는 것을 목표로 한다.
 2. 매핑된 엔티티 객체 데이터를 공유서버와 연결된 DB에 저장함을 목표

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

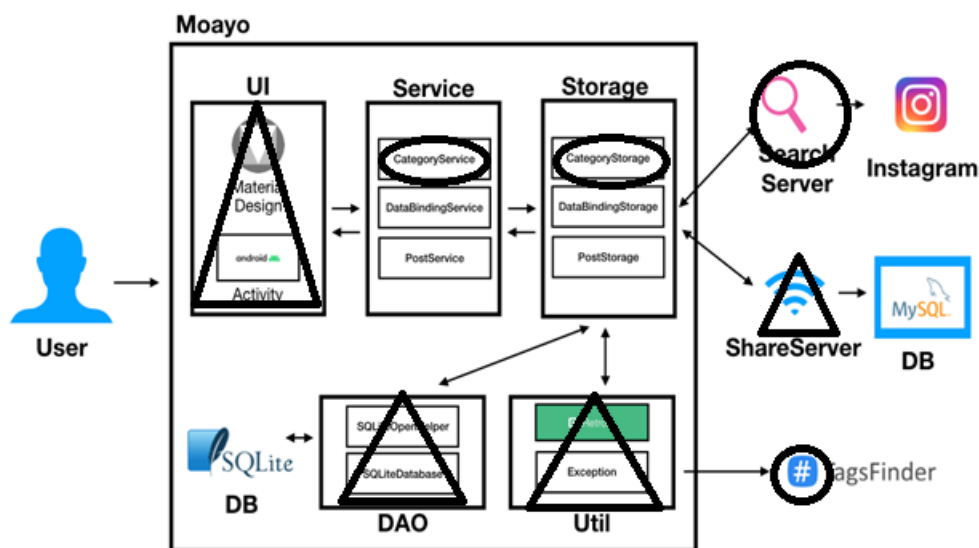
로 한다.

II. 공유서버 내부 정보를 사용자에게 제공


1. 백엔드가 공유서버 API로 공유된 도감 목록을 요청하면, 요청 바디에 연관 태그를 분석하고 이에 대한 데이터를 파싱하여 내부 로직을 처리 할수 있도록 함을 목표로 한다.
2. 공유서버와 연결된 DB로부터 공유된 도감 정보들을 가져와 모바일 백엔드에게 전달함을 목표로 한다.
3. 백엔드가 특정 도감에 대한 공유 요청을 보내면 해당 도감에 대한 정보를 DB에서 가져와 제공함을 목표로 한다.

2.2. 수행내용

2.2.1.구조도 설계

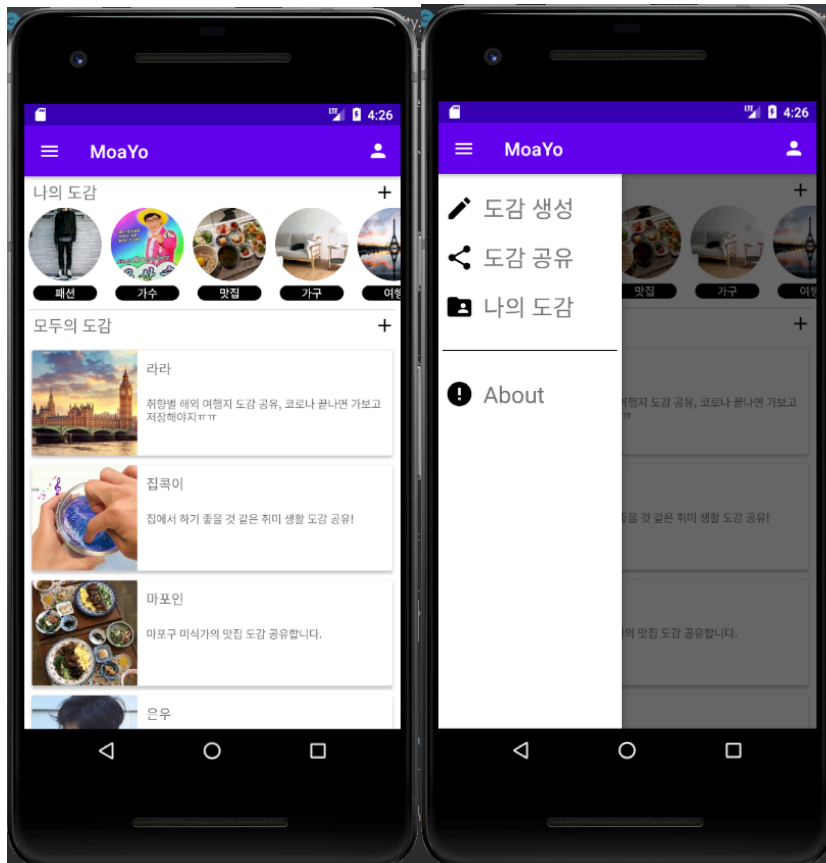


- A. 현재 구현 완료된 부분은 ○표시를 하였고, 현재 구현을 하고 있는 부분을 △ 표시하여 구분하였다. 표시가 없는 부분은 추후, 구현을 할 예정이다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

2.2.2.모바일 프론트엔드 진행상황

1. 프레젠테이션 UX/UI 및 이벤트 처리 구현



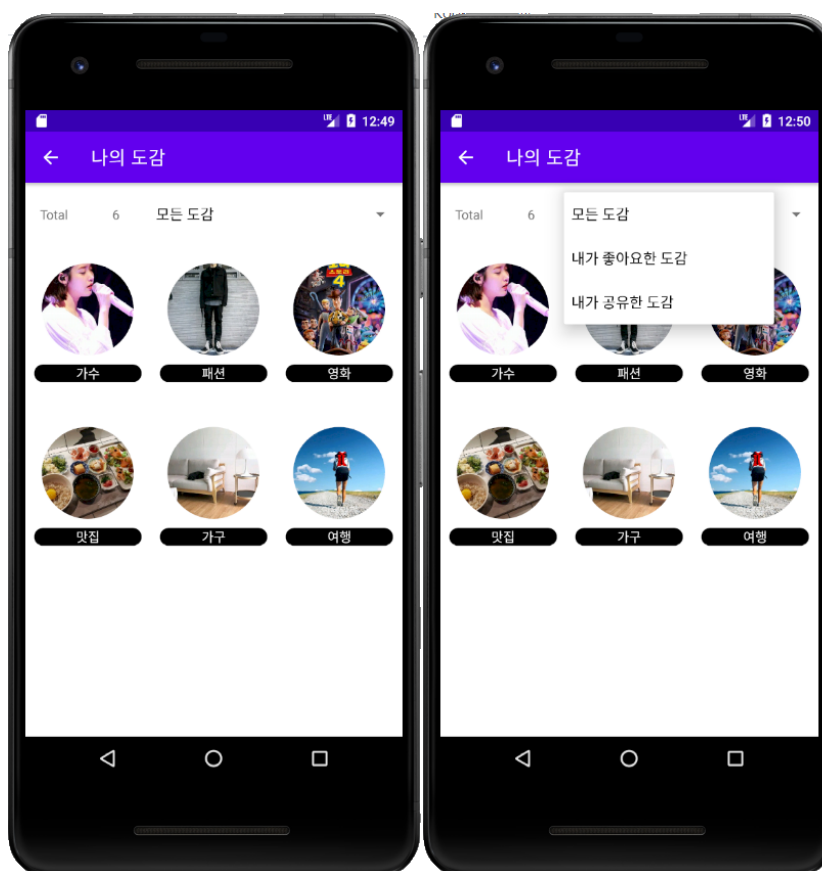
a. 메인 화면(MainActivity)

- i. 어플리케이션을 시작했을 때, 사용자에게 처음 접근되는 시작 화면.
 해당 화면에서는 나의 도감 리스트와 추천되는 타 사용자의 공유도감 리스트가 RecyclerView를 통해 제공되는데, 각 Adapter를 통해 View에 데이터를 할당을 처리하였다.
 현재 추천되는 공유도감은 미 구현 단계이므로 예시 데이터만을 보여준다.
- ii. 상단 톨 바의 메뉴 버튼을 클릭 시, 도감 생성, 도감 공유, 나의 도감 세가지 메뉴에 접근 할 수 있는 Drawer가 펼쳐진다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

현재 나의 도감 버튼만이 Intent를 통해 연결되어 있으며, 나머지 메뉴에 대한 연결은 구현되어 있지 않다.

- iii. 또한 툴바의 우측 유저모양의 아이콘을 통해 나의 도감에 바로 접근이 가능하다.



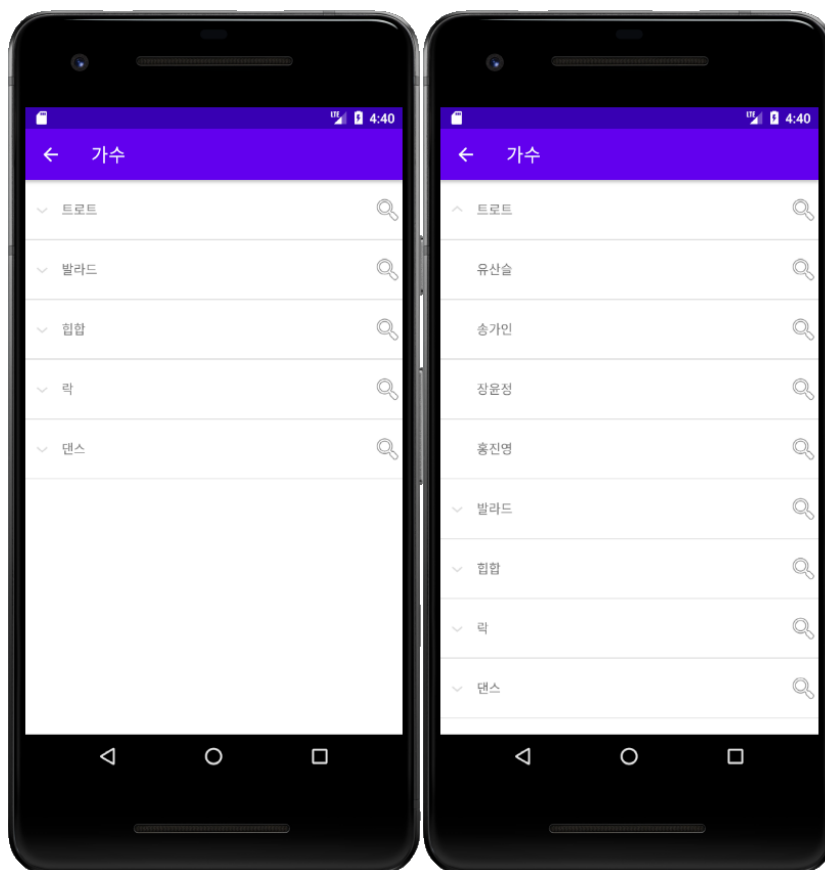
b. 나의 도감 화면(BookDetailActivity)

- i. 해당 액티비티에서는 백엔드에게 해당 사용자의 도감 데이터를 받아오는 함수를 작성되어야 한다. (현재 백엔드와의 통신은 구현되지 않았으므로, 더미 데이터를 생성하는 함수를 호출하도록 구현하였다.)
- ii. 나의 도감화면 상단에는 유저가 가진 도감의 총 개수가 TextView를 통해 표시되며,우

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서	
		프로젝트 명	모아요 (MoaYo)
		팀 명	고리고리
		Confidential Restricted	Version 1.2 2020-04-23

측엔 SpinnerView를 통해 도감의 정렬기준을 선택할 수 있는 드롭다운 메뉴를 제공한다. 정렬 기준은 모든 도감, 좋아요한 도감, 공유한 도감 세가지로 구분된다. 현재는 정렬에 필요한 기준(좋아요, 공유)이 구현되지 않았으므로 작동되지 않는다.

- iii. (i) 단에서 가져온 사용자의 도감데이터는 ViewPager를 통해 제공된다. 도감 개수에 따라 여러 Fragment가 동적으로 생성되어 Pager에 표시되며, 하나의 Fragment 당 9개의 도감이 바둑판 형태로 제공되도록 구현하였다.



c. 도감 상세 화면(BookDetailActivity)

- i. 해당 액티비티에는 선택된 도감에 대한 계층 구조가 화일시스템(TreeView)의 형태로 보여지도록 ExpandableListView(확장패널)를 사용하여 구현하였다.

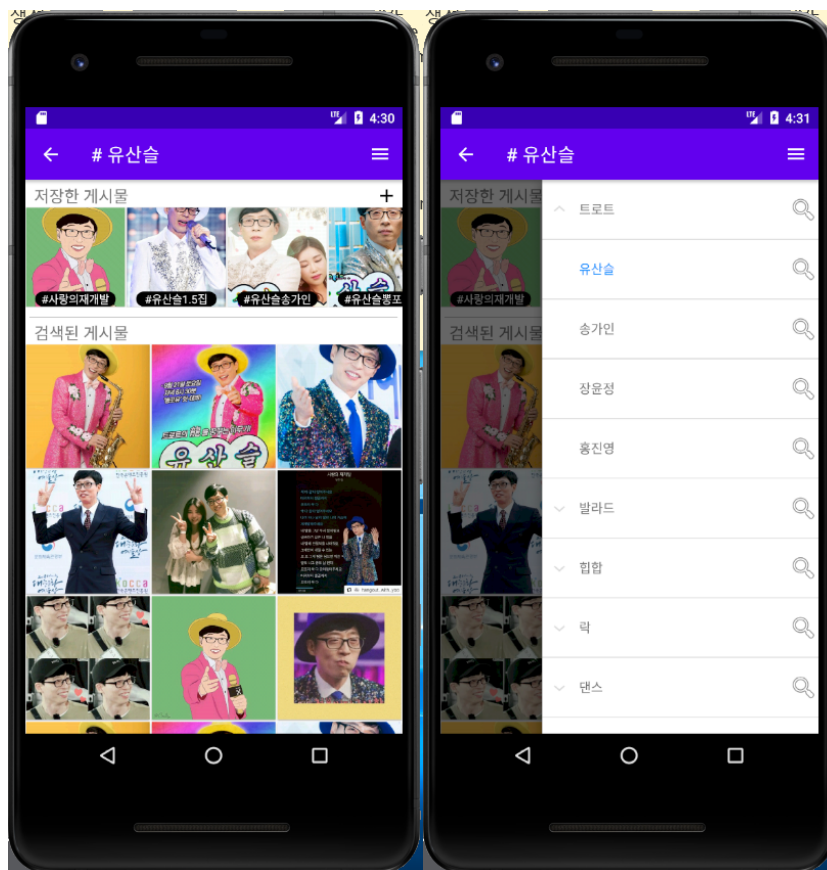
ExpandableAdapter의 생성자에 도감 데이터(CategoryNode 객체의 root node)를 매개

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서	
		프로젝트 명	모아요 (MoaYo)
		팀 명	고리고리
		Confidential Restricted	Version 1.2 2020-04-23

변수로 전달하여 데이터가 담긴 view를 생성하며, 각 노드에 담긴 lowLayer들은 하위 패널로 생성되어, 패널 확장시에만 표시되도록 구현하였다.


- ii. 각 패널은 onClickEvent를 통해 확장/축소를 실행하며, 확장 시, 하위패널이 존재한다면 펼쳐진다.

모든 패널 우측에는 해당 키워드의 검색을 위한 ImageButton(search)이 제공되며, 클릭 시 해당 패널 데이터(객체)를 검색 결과화면으로 전달한다.



d. 검색 결과 화면(ResultActivity)

- i. 해당 액티비티에서는 선택된 키워드로부터 검색된 게시물과, 그 게시물 중 사용자에게 의해 저장된 게시물이 RecyclerView로 제공된다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

저장된 게시물은 가로뷰의 형태이며, 우측 상단 Plus 버튼(ImageButton)을 통해 한번에 모든 저장 게시물을 확인할 수 있도록 하였다. (아직 저장게시물 리스트 화면은 구현되지 않음)

- ii. 상단 톨 바에는 검색된 키워드 명이 title로 표시되며, 메뉴 버튼이 존재한다.
메뉴 버튼을 클릭 시, 도감 계층구조가 ExpandableListView로 보여지며, 현재 검색된 키워드는 구분되도록 다른 색으로 표시되도록 제공한다.
이 메뉴(Drawer)에서는 도감 상세화면과 동일하게 우측 검색버튼을 통해 바로 다른 키워드에 대한 검색이 가능하도록 구현되었다.

2.2.3.모바일 백엔드 진행상황

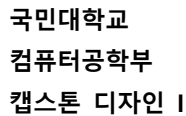
1. 도감 관련 비즈니스 로직 및 스토리지 모듈 구현

A. 도감 생성

- i. 사용자가 카테고리를 입력하면 해당 카테고리를 비즈니스 로직에서 Null값이 들어있는지 유효성을 검사한다. 검사 후, 데이터베이스에 저장하기 위한 데이터로 변경하기 위한 스토리지 모듈에 보내진다. 해당 모듈에서는 카테고리를 각각의 테이블에 저장하기 위한 매핑 데이터로 변환 시키고, Dao를 통해서 데이터베이스에 저장한다.

co_id	co_title	co_description	co_password
필터	필터	필터	필터
1	sample dogam	this is sample...	1234

도감의 기본적인 정보를 도감 테이블에 저장한다.



2020-04-23

co_id	co_title	co_parent	co_level	co_dogamld
필드	필드	필드	필드	필드
1	L1	1	1	1
2	L2_0	1	2	1
3	L3+0	2	3	1
4	L3+1	2	3	1
5	L3+2	2	3	1
6	L3+3	2	3	1
7	L3+4	2	3	1
8	L2_1	1	2	1
9	L3+0	8	3	1
10	L3+1	8	3	1
11	L3+2	8	3	1
12	L3+3	8	3	1
13	L3+4	8	3	1
14	L2_2	1	2	1
15	L3+0	14	3	1
16	L3+1	14	3	1
17	L3+2	14	3	1
18	L3+3	14	3	1
19	L3+4	14	3	1

사용자가 입력한 각 계층의 카테고리 노드를 데이터베이스에 저장한다. Parent는 자신의 상위 계층의 id를 저장한다. 각 카테고리 노드를 레벨로 구분하여 데이터베이스에서 정보를 파싱할 경우, 쉽게 변환이 가능하다.


[illegible]

- ii. 생성이 완료된 경우, 해당 카테고리의 내용을 JSON 형태로 출력한다.

B. 도감 찾기

- i. 생성된 카테고리를 찾고자 할 때, 두 가지의 방법을 사용할 수 있도록 메소드를 구현하였다. 도감의 ID를 통해서 찾는 방법과 도감의 Title을 이용해서 찾는 방법이다. 각 노드를 찾는 것은 사용자가 접근할 수 없는 권한이기 때문에 구현하지 않았다.

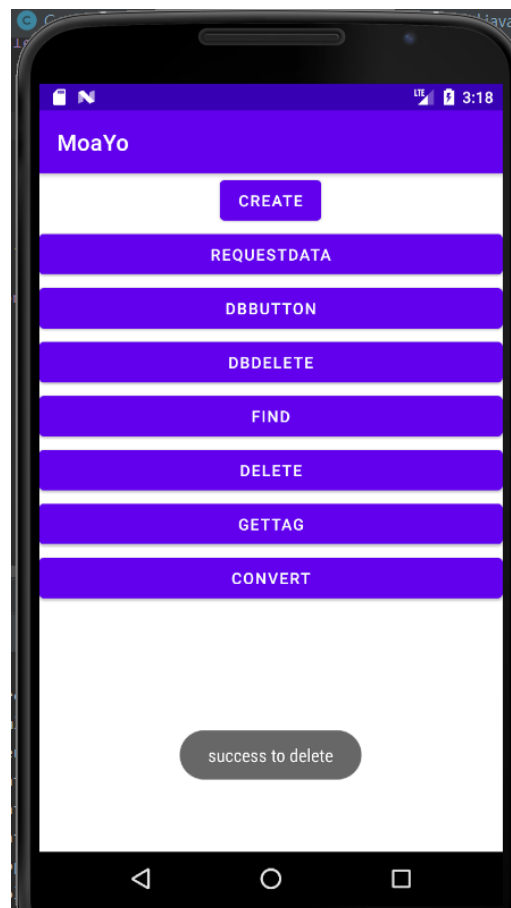
[illegible]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	모아요 (MoaYo)	
	팀 명	고리고리	
	Confidential Restricted	Version 1.2	2020-04-23

- ii. 사용자가 원하는 도감의 정보를 데이터베이스에서 찾은 경우, JSON 형태로 출력한다.

C. 도감 삭제


- i. 사용자가 입력한 ID에 존재하는 도감이 존재할 경우, 해당 도감을 삭제한다. 도감의 ID만 있으면 해당 ID를 외래키로 받고 있는 다른 테이블의 데이터도 같이 지워지는 Cascade방식을 사용한다.



- ii. 도감의 삭제가 완료된 경우, 삭제가 완료되었다는 토스트 메시지를 출력한다.

2. Retrofit을 이용한 서버와의 통신 구현

- A. 검색 서버, 공유 서버와 통신을 하기 위해 HTTP 통신을 사용하고, 이를 위해 Square 사의 Retrofit2를 이용한다. 검색 서버, 공유 서버를 위한 서비스 인터페이

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모야요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

스를 따로 만들고, 해당 서버의 API에 통신할 메서드를 구현하였다. 또한 직접 통신을 할 Retrofit Client는 싱글톤 기법을 사용하여 충돌이 일어나지 않도록 한다.

- B. 현재 EC2 서버의 배포가 완료되지 않아 직접적인 통신은 불가능한 상태이지만, 타 API를 이용하여 연결이 되는 것을 확인하였다.

3. 태그 유사어 크롤링 유틸 구현

- A. 유사어 크롤링은 사용자가 카테고리 생성을 완료하고 난 후, 스토리지 모듈에서 데이터를 각각의 테이블에 저장하기 전에 사용한다.

```
I/System.out: #데일리 #일상 #daily #몰아요 #소통 #맛집 #골카 #골스타그램 #ootd #selfie #캠피 #인샷 #탈로우 #instagood #좋아요번사 #오오리니 #f4f #follow #instadaily #인신 #데일리룩 #올스타그램 #일상스타그램
#인샷하면맛있음 #올반 #selca #instagram #여행 #fff #인스타그램
I/art: Do partial code cache collection, code=47KB, data=55KB
After code cache collection, code=44KB, data=54KB
Increasing code cache capacity to 256KB
```


- i. 위 데이터는 '데일리'라는 태그의 유사어를 크롤링하여 가져온 것이다.

4. 데이터베이스 테이블 구현

- A. 도감과 게시물 저장에 위하여 어플리케이션 내부에 SQLite를 이용하여 데이터베이스에 데이터를 저장한다. 도감에는 도감의 정보, 각 계층의 노드의 정보, 노드의 게시물의 정보가 들어있기 때문에, 도감을 3개의 테이블로 나누어 저장한다.

tb_category			CREATE TABLE tb_category(co_id integer primary key autoincrement, co_title text not null, co_parent integer, co_level integer not null)
co_id	integer		"co_id" integer PRIMARY KEY AUTOINCREMENT
co_title	text		"co_title" text NOT NULL
co_parent	integer		"co_parent" integer
co_level	integer		"co_level" integer NOT NULL
co_dogamId	integer		"co_dogamId" integer NOT NULL
tb_content			CREATE TABLE tb_content(co_id integer primary key autoincrement, co_parent integer, co_level integer, co_url text, co_imgUrl text, co_info text, co_hashTag text)
co_id	integer		"co_id" integer PRIMARY KEY AUTOINCREMENT
co_parent	integer		"co_parent" integer
co_level	integer		"co_level" integer
co_url	text		"co_url" text
co_imgUrl	text		"co_imgUrl" text
co_info	text		"co_info" text
co_hashTag	text		"co_hashTag" text
tb_dogamlist			CREATE TABLE tb_dogamlist(co_id integer primary key autoincrement, co_title text, co_description text, co_password text)
co_id	integer		"co_id" integer PRIMARY KEY AUTOINCREMENT
co_title	text		"co_title" text
co_description	text		"co_description" text
co_password	text		"co_password" text

- i. 카테고리 테이블은 각 계층의 노드를 저장하는 테이블로, 각 노드의 ID, Title, 노드의 직속 부모 노드의 ID, 노드의 Level, 노드의 DogamId를 인덱스로 구성되었다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

ii. 게시물 테이블은 각 게시물을 저장하는 테이블로 각 게시물의 ID, 게시물을 가지고 있는 카테고리 노드의 ID, 게시물의 Level, 게시물의 실제 정보를 가지고 있는 URL, 게시물의 실제 이미지를 가지고 있는 imgUrl, 게시물의 간략한 정보를 나타내는 info, 게시물의 태그를 가지고 있는 hashtag 인덱스로 구성되었다.

iii. 도감 테이블은 도감을 저장하는 테이블로, 도감의 ID, 도감의 Title, 도감의 설명을 나타내는 Description, Password 인덱스로 구성되었다.

2.2.4. 서버 진행상황

I. 검색서버

모바일 백엔드로부터 발생한 검색 요청에 대한 기능적 처리 구현 완료. JSON 형태로 요청 받은 검색어에 맞게 인스타그램에서 실시간으로 크롤링을 수행하고, 크롤링을 통해 얻은 게시물 데이터들에 메타 검색 로직을 적용하여 계층 별로 분류한 결과를 JSON 형태로 제공한다.

A. 인스타그램 게시물

카테고리의 태그 및 태그에 대한 관련유의어를 JSON 형태로 입력받아 파싱한 뒤 파싱한 데이터들을 가지고 인스타그램 게시물들을 크롤링 수행하고, 메타 검색 로직을 적용시킨 결과를 JSON에 담아 반환한다. 메타 검색 로직의 수행은 다음과 같다.

1. 각 계층의 태그에 대해 인스타그램에서 크롤링을 수행한다.
2. 상위 계층 게시물 기준, 하위 계층의 크롤링된 게시물과 비교해 동일한 ID값을 가진 게시물은 하위 계층에 넣고 일치하지 않으면 상위 계층에 둔다. 하위 계층에서 일치하지 않는 게시물은 버린다.
3. 위와 같은 방식의 AND 연산을 유의어에도 적용시켜 값을 도출한다. 해당 작업은 스텝으로 실시한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	모아요 (MoaYo)	
	팀 명	고리고리	
	Confidential Restricted	Version 1.2	2020-04-23

위 작업은 수행 시간 길기 때문에 멀티 스레딩으로 작업하여 수행시간을 단축 시켰으며, 계층간 도출되는 게시물은 나누어 실행하여 제공되기 까지의 시간을 단축시키도록 구현하였다.

II. 공유 서버

A. 도감 공유 요청

모바일 백엔드로부터 도감 공유 요청이 들어오면 해당 요청에 대한 데이터를 파싱하여 공유 서버 데이터베이스에 저장한다. 현재 파싱하여 저장하는 로직까지 완료 되었으나, 데이터베이스 변경으로 수정 예정.

B. 공유된 도감 리스트 요청

데이터베이스에 접근하여 도감들에 대한 ID, Title, Description, Writer 정보를 가져오는 로직은 구현완료. 해당 정보를 JSON으로 변환하는 로직이 구현 진행중.

C. 공유된 도감 세부 정보 요청

데이터베이스에 접근하여 도감에 대한 상세 데이터를 매핑하는 로직까지 구현. 해당 정보를 JSON으로 변환하는 로직 진행중.


3. 수정된 연구내용 및 추진 방향

3.1. 수정사항

3.1.1.모바일 백엔드 변경 사항

1. 서버와 통신하는 방법 변경

- A. 서버와 통신하는 방법 중 하나로 HTTP 통신을 사용하고 있다. 이 통신의 방법으로 HttpURLConnection 클래스를 사용하여 값을 요청하고 응답받는 방식을 사용

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	모아요 (MoaYo)	
	팀 명	고리고리	
	Confidential Restricted	Version 1.2	2020-04-23

했지만, 해당 기능은 객체를 JSON 형태로 변환하는 과정에서 많은 시간을 소모시키고, 직관적이지만, 코드의 길이가 길어지고, 단순 노가다를 해야 하는 부분이 존재한다.

B. 따라서 통신하는 방식을 Square사의 오픈소스인 Retrofit2를 사용하기로 변경함. 아래는 Retrofit을 사용할 때의 기대 효과이다.

- i. 모든 서버 콜에 대해서 짧고 간결한 코드로 대응이 가능
- ii. OkHttp 기반 – 기존보다 속도가 더 빨라진다.
- iii. JSON 파싱을 Gson 파싱을 사용하기 때문에 자동으로 파싱을 해줌으로써, 코드를 간결화시킬 수 있다.
- iv. 콜백은 메인 스레드에서 돌고 통신부는 알아서 스레드에서 돌아와서 별도의 비동기 처리를 구현할 필요가 없다. 코드 어디서든 호출이 가능하다는 장점이 있음.

C. 해당 Retrofit의 Gson 파싱을 위한 Response 데이터는 미리 클래스화 시켜야 하기 때문에, 따로 클래스를 새로 적용한다.

2. 유의어 추출시 사용하는 API의 변경 → TagFinder에서 크롤링한 결과 적용

- A. 기존 확인한 API는 국립국어원의 사전 API를 사용하지만, 해당 API는 사전적 의미를 전달해줄 뿐, 입력한 태그의 유의어를 주지 않는다.
- B. 따라서 TagFinder 사이트에서 직접 크롤링하여 연관 유의어를 3~5개 추출하는 것을 적용한다.

3. 카테고리에 대한 게시물 데이터 저장 가능

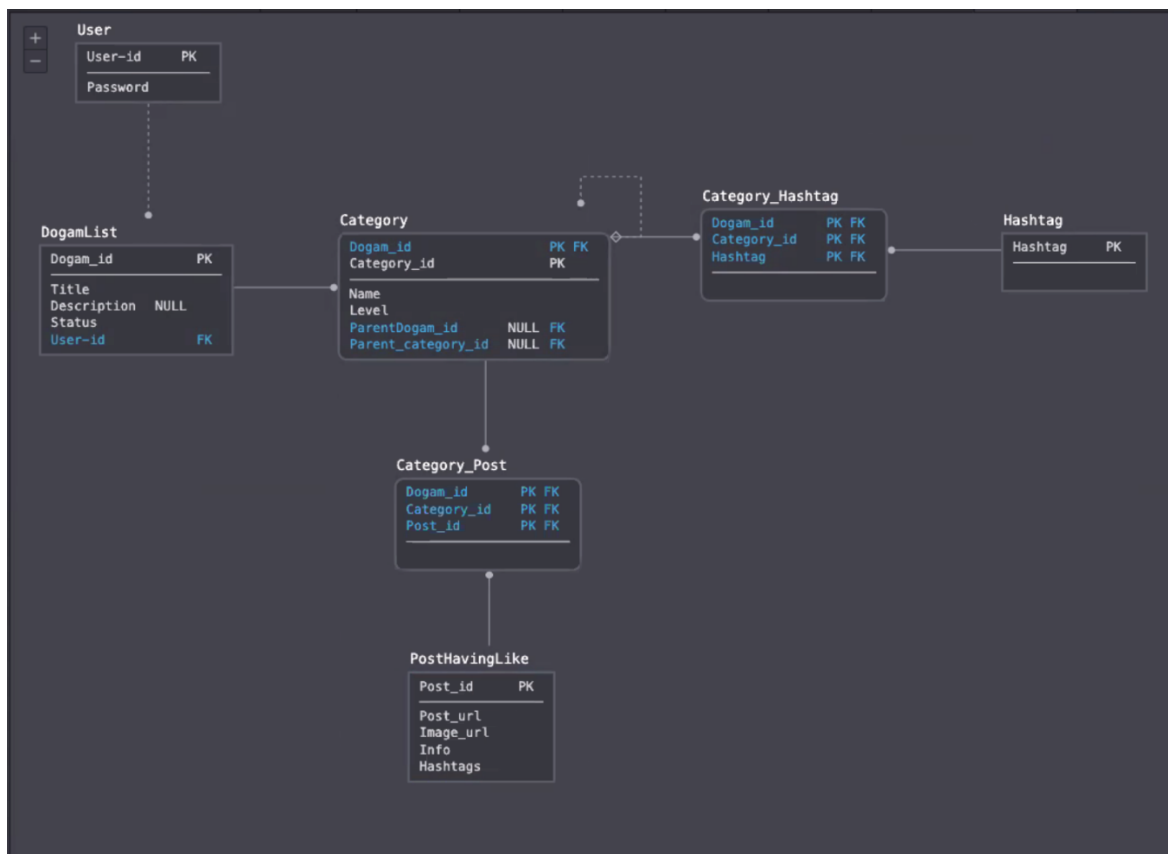
- A. 초기 구현 당시 단일 태그에 대한 문제점을 인식하고, 이를 해결하기 위한 복합 태그 연산 작업을 했지만, 작업에 대한 데이터를 저장하지 않아서 사용자가 계속 입력해야 하는 문제점이 존재

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

- B. 어플리케이션 내부의 데이터베이스를 사용하여 사용자가 입력한 태그 계층의 기록을 남김 – SQLite를 사용하기로 한다.

4. 앱 데이터베이스 테이블 변경


- A. 기존의 데이터베이스 테이블은 카테고리, 도감, 게시물 3개의 테이블이 존재. 하지만 데이터의 무결성을 위해 테이블의 1:N, N:M 관계를 모색하였고, 이를 통해 추가된 테이블이 생겼다. 아래는 추가된 테이블과 그 관계도이다.



- B. 카테고리과 게시물을 연결하는 카테고리-게시물 테이블, 해시태그 테이블, 카테고리과 해시태그를 연결하는 카테고리-해시태그 테이블이 추가로 생성되었다.

5. GraphQL EndPoint 캐싱을 위한 EndPoint 값 추가

- A. 검색 서버에 요청하는 기존의 데이터 형식은 각 계층의 해시태그와 유의어를 JSON 형태로 보내는 것이었다. 하지만 데이터의 양이 많아질수록, 속도가 저하되

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

는 것을 발견하였다. 속도의 개선을 위해서 한정된 데이터를 먼저 응답받고, 서버에서 검색을 멈춘 EndPoint 값을 받는 것으로 속도를 개선하기로 했다.

- B. 이후, 사용자가 새로고침, 또는 스크롤을 끝까지 내릴 경우, 다시 요청을 보낼 때, 해당 EndPoint의 값도 같이 보낸다.

3.1.2.검색 서버 변경 사항


게시물 크롤링 시 인스타그램에서 제공하는 오픈소스 API를 쓰려다가 제공하는 정보량이 너무 적어서 안쓰기로 함

앱 백엔드에서 Request Body가 수정됨에 따라 Response Body에 GraphQL 엔드포인트 값 추가

3.1.3.공유 서버 변경 사항

json파일로 저장 -> 데이터 무결성 때문에 DB 테이블에 저장

앱 내부 DB 테이블을 변경함에 따라 공유 서버 DB 테이블도 바뀜

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

4. 향후 추진계획

4.1. 향후 계획의 세부 내용

4.1.1. 모바일 프론트엔드 향후 계획

A. 카테고리 계층 구조 생성/관리 화면 UX/UI

1. 도감을 생성하고, 관리하는 품을 파일시스템 구조(Tree View) 형태의 화면으로 제공한다. 각 계층 레벨 별로 Fragment를 분할하고, 사용자가 자유롭게 화면을 전환할 수 있도록 구현 예정.

B. 검색 게시물 저장

1. 검색된 게시물 리스트 중 사용자가 영구적으로 보관하고 확인하고 싶은 게시물을 저장 기능을 통해 보관한다. 저장된 게시물은 해당 태그의 검색결과 화면 상단에 RecyclerView(horizontal)로 제공되며, 모든 저장된 게시물을 모아 확인할 수 있는 메뉴 또한 추가적으로 구현할 예정.

C. 스크롤링을 통한 동적 데이터 요청


1. 사용자에게 짧은 지연시간 내 요청된 검색게시물 데이터를 제공하기 위해, 분산된 데이터를 백엔드로부터 응답 받도록 한다. 검색 결과 화면에서 게시물 리스트를 일정 스크롤링 이벤트가 발생 할 때마다, 백엔드에게 데이터를 동적으로 받아와 화면에 출력하는 기능을 구현할 예정.

D. 도감 공유화면 UX/UI

1. 사용자가 가진 도감을 공유할 수 있는 품과 타 사용자들의 공유된 도감들을 볼 수 있는 화면을 구현할 예정.

E. 디자인

1. 사용자에게 제공 될 전반적인 어플리케이션의 디자인을 보완/수정한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	모아요 (MoaYo)	
	팀 명	고리고리	
	Confidential Restricted	Version 1.2	2020-04-23

AdobeXD tool을 통해 사전에 만들어진 디자인 화일을 각 View에 알맞게 적용하며, 액티비티 내 요소들의 위치, 크기 등 구조적인 부분에 있어 완성도 있는 디자인으로 보완할 예정.


4.1.2.모바일 백엔드 향후 계획

A. 검색 서버의 요청 데이터 비동기 처리 및 위치 기억

1. 검색 서버에서 데이터를 처리하기 까지의 시간이 많이 걸리기 때문에, 쓰레드를 이용해서 비동기 처리하는 것을 목표로 한다. 또한, 많은 데이터를 가져오게 되면, 데이터 처리를 위한 시간이 오래 걸리기 때문에, 한정된 양의 데이터만 우선적으로 받아 해당 데이터를 프론트엔드에 리턴하고, 추가적으로 사용자가 데이터를 요청한 경우, 서버에서 검색한 EndPoint에서부터 읽을 수 있도록 하기 위한 EndPoint 값도 같이 보내는 것을 목표로 한다.
2. 검색 서버에 요청할 데이터는 Util 모듈의 Converter 모듈을 사용하여 JSON 형태로 변환한다. 또한 응답받는 데이터는 ResponseForm이라는 클래스를 통해서 받는 것을 목표로 한다. - 검색 서버에서 요청받은 데이터를 저장할 필요가 없기 때문에 해당 비즈니스 로직은 스토리지 모듈에 연관되지 않는다.

B. 공유 서버 관련 비즈니스 로직 및 스토리지 구현

1. 타 사용자가 공유한 도감을 볼 수 있도록 하기 위한 데이터 추출을 위한 서버와 연결을 Retrofit을 이용하여 연결하도록 한다.
2. 사용자가 자신의 도감을 타 사용자에게 공유할 경우, 해당 도감을 공유 서버에 요청하는 작업을 구현하는 것을 목표로 한다.
3. 공유 서버와의 통신에서 저장할 필요가 없기 때문에 스토리지 모듈은 연관되지 않는다. 해당 서브시스템을 구현하기 위한 아키텍처 구조는 Client-Server 아키텍처 구조를 따르는 것으로 한다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

C. 데이터베이스 변경 사항 적용

1. 수정한 데이터베이스의 테이블에 맞게 기존의 엔티티 양식 변경
2. 수정한 데이터베이스의 테이블에 맞게 스토리지 모듈에 해시태그, N대M 관계를 위한 매핑 데이터에 관련된 인터페이스를 구현한다.

4.1.3. 검색 서버 향후 계획

A. 최적화


현재는 3~4천 개 정도의 게시물을 한번에 크롤링하여 메타 검색 로직을 적용한 결과를 제공하는데, 이로 인해 긴 지연 시간과 404 에러 때문에 크롤링이 원활하지 않은 문제가 발생한다.

이 문제는 pagination 값을 요청, 응답 구조체에 캐싱함으로써 해결 가능하다.

즉, 15번 크롤링할 것을 3번씩 5번으로 나누어 크롤링함으로써 지연 시간을 낮출 수 있고 필요할 때마다 검색 서버에 요청하면 되므로 한번에 보여줄 수 있는 게시물 갯수가 한정적인 문제도 해결된다.

또한 크롤링 시 종종 404 에러가 발생하는데, 이 또한 pagination 값을 캐싱함으로써 404 에러가 발생하면 다시 해당 pagination부터 크롤링을 재시도하여 에러가 발생해도 크롤링이 중단되는 오류를 방지할 수 있다. 404 에러는 해당 pagination에 재요청하면 해결되는 문제이기 때문이다.

단, 이 때 계속 404 에러가 발생한다면 인스타그램의 난독화 매커니즘이 발동하는 시간에 물린 것으로 간주하고 처음부터 다시 크롤링한다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서	
		프로젝트 명	모아요 (MoaYo)
		팀 명	고리고리
		Confidential Restricted	Version 1.2 2020-04-23

B. 모바일 앱과의 데이터 교환 테스트

최적화 완료 후 로컬 호스트에서 검색 서버를 구동하고 모바일 앱과 데이터 교환이 정상적으로 이루어지는 지 테스트한다.

C. EC2 배포

테스트까지 완료되면 AWS EC2 인스턴스를 생성하고 서버를 배포한다.

4.1.4.공유 서버 향후 계획

A. 변경 사항 적용

모바일 백엔드 내부 데이터베이스 테이블 변경에 따라 공유서버의 데이터베이스 테이블도 해당 사항을 반영하여 변경하여 적용한다. 데이터베이스 변경에 따라 내부 로직들도 수정 및 추가를 실시한다.

▪ 변경 내용


해시태그 테이블,유저 ID 테이블 추가. Content 테이블명 PostHavingLike로 변경. Category와 PostHavingLike, Category와 HashTag테이블 사이의 관계가 n:m의 관계로 성립됨에 따라 중간 테이블 추가.

B. 기능 구현

현재 공유된 도감 정보 요청과 도감 정보 세부요청에 대한 구현이 진행중이다. 공유서버에서 필요한 기능을 모두 구현한다.

1. 도감 리스트 정보 요청

모바일 백엔드에서 Header값에 검색하고자 하는 태그를 실어 GET방식으로 요청하면, 해당 정보를 파싱하여 공유서버 DB에서 검색하여 도감에 대한 ID, Title, Description, Writer,

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

Date 값을 Json의 형태로 반환한다.

2. 도감 정보 세부 요청

모바일 백엔드에서 도감에 대한 ID값을 POST로 요청하면, 해당 값을 가진 도감 정보와 카테고리 정보, 포스트 정보를 공유서버 DB에서 검색하여 반환한다.

C. 디버깅

모바일 백엔드와 테스트 하기 이전, Postman 을 이용하여 자체 테스트를 실시한다.

D. 모바일 백엔드와 교환 테스트

기능 구현 완료 이후 로컬 서버에서 모바일 백엔드와 통신 테스트 실시한다.

E. AWS배포


테스트가 완료 되었다면, AWS EC2를 이용하여 서버를 배포한다. 검색서버와 다른 EC2를 이용하되, VPC,Subnet 는 같은 곳을 이용한다.

5. 고충 및 건의사항

5.1. 개발분야 별 고충 및 건의사항

a) 동적 검색결과 데이터 요청.

서버를 통해 전달받은 검색 결과 게시물들을 한번에 모두 가져오기에는 오랜 지연시간이 걸리며, 사용자에게 따라 원하는 것보다 과대한 데이터를 받아올 수 있다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
		프로젝트 명	모아요 (MoaYo)	
		팀 명	고리고리	
		Confidential Restricted	Version 1.2	2020-04-23

때문에 인스타그램, 페이스북과 같이 RecyclerView 리스트를 일정길이 만큼 스크롤링 할 때마다, 데이터를 분할하여 받아와 사용자에게 제공하려 한다.

i. **어떠한 이벤트를 기준으로 동적으로 데이터 요청을 해야 하는가.**

> 스크롤링 이벤트를 감지하여 데이터를 호출해야 할지, 사용자가 화면의 최하단에 도달하는 것을 감지하여야 할지에 대한 고려사항이 존재한다.

ii. **동적으로 받아온 데이터를 RecyclerView에 실시간으로 출력.**

> 현재 Adapter를 통해 데이터를 담은 RecyclerView 내부의 요소들을 생성하여 화면에 표시하고 있다. Refresh 없이 사용자에게 부드러운 응답이 가능할 지에 대한 고려사항이 존재한다.

b) 비동기 처리 문제

> 현재 검색 서버에서 데이터를 요청하는 로직은 레트로핏을 이용한 비동기처리 작업을 한다. 하지만 메인 액티비티에서 함수를 호출하여 동기로 처리하는데 해당 함수는 비동기처리로 데이터를 가져온다. 이 부분에서 비동기 처리한 데이터를 받을때까지 리턴을 하지 않는 방법을 사용하고 있는데. 이 경우 비동기 처리를 하는 의미가 없어지게 된다. 이 부분에서 데이터를 요청하고 기다리는 동안 프론트엔드에서는 다른 작업을 시킬 수 있는 방법을 강구중이다.