

안녕하세요. 23조 모아요 프로젝트를 시작하겠습니다.

1) 여러분은 '유산슬' 하면 무엇이 떠오르시나요? 중국 음식을 떠올리셨나요? 저는 요즘 인기 있는 트로트 가수가 떠오릅니다. 유재석씨의 가수 예명이죠.

이처럼, 관심 있는 분야는 사람마다 가지 각색이므로, 각자 원하는 게시물만을 쓱쓱 골라서 볼 수 있다면 참 좋겠죠?

1-1) 저희 모아요 프로젝트는

1-2) 메타 검색과 sns를 접목시켜 이를 가능하게 합니다.

접목시킬 sns는 인스타그램입니다. 최근 몇 년간 이용자수가 가장 많이 증가한 그야말로 '핫한' sns라고 볼 수 있지만, 저희가 생각한 인스타그램은 그리 편한 sns는 아니었기 때문입니다.

2)

2-1) 여러 개의 해시태그를

2-2) 동시에 검색할 수 없어, 원하는 게시물을 정확히 검색하기 어렵고

2-3) 고양이와

2-4) Cat처럼 같은 의미의 해시태그 이지만

2-5) 다른 검색결과를 출력하기 때문에, 여러 번 검색을 거쳐야 한다는 불편함이 있습니다. 검색된 게시물 가운데 유용한 정보이거나 관심 있는 분야의 게시물을 저장 하고 싶은 경우,

2-6) 계층 구조로 분류하여 저장하는 기능을

2-7) 제공하지 않고, 단순 1차적 분류만 가능하다는 문제점이 있습니다. 이는 컬렉션이 점차 늘어감에 따라 구분이 어려워지고 저장했던 게시물을 찾기 불편하게 합니다.

2-8) 또한 다른 사용자와 게시물을 공유할 때에도,

2-9) 한번에 다량의 게시물을 공유할 수 없어 일일이 공유해야 하는 번거로움이 있습니다. 이와 같은

2-10) 검색의 불편함

2-11) 저장의 불편함

2-12) 공유의 불편함을

2-13) 해소하고 양질의 서비스를 제공하고자

3) 저희 MoaYo 앱에서 구현하는 주요 기능 세가지는 다음과 같습니다.

3-1) Meta 검색

3-2) 계층 구조로 게시물 저장

3-3) 퍼블리싱. 입니다.

3-4) 각 기능에 대해 살펴보겠습니다.

4)

4-1) 먼저 메타 검색 기능입니다.

4-2)

4-3) 사용자는 직접 원하는 주제로 계층 구조의 카테고리를 생성합니다. 이 때, 만들어진 계층 구조 전체를 통틀어 하나의 '도감'이라고 합니다.

4-4) 도감에서 검색 결과를 받아보고 싶은 키워드를 선택합니다.

4-5) 예를 들어, 가수>트로트>유산슬 의 순서로 계층을 생성했고, 3차 계층인 유산슬을 선택했다면

4-6) 선택한 키워드에 직속 상위 레벨의 키워드들을 합하여

4-7) 해시태그를

4-8) 동시에 검색한 것과 같은 즉, AND연산의 결과를 보여주며

4-9) 사용자는 인스타그램으로부터 실시간으로 크롤링된 '트로트 가수인 유산슬' 에 해당하는 검색 결과를 얻게 됩니다.

4-10) 같은 유산슬이지만

4-11) 이번에는

4-12) '맛집' 도감의 유산슬을 선택한 경우입니다.

4-13) 검색 결과는

4-14) '일산의 유산슬 맛집' 에 해당하는 게시물입니다.

4-15) 또한 유산슬과

4-16) 류산슬처럼 각각의 태그에 대해 유의어를

4-17) OR연산으로 처리합니다.

4-18) 따라서 AND OR 연산의 검색 로직을 통해

4-19) 정교한 검색 결과를 받아볼 수 있습니다.

5) 다음은

5-1) 계층 구조로 게시물을 저장하는 기능입니다.

5-2)

5-3) 메타 검색을 통하여 얻은 '트로트 가수인 유산슬' 게시물 가운데

5-4) 원하는 게시물을 나의 도감으로 저장할 수 있습니다.

6)

6-1) 저장한 게시물을 한번에 공유할 수 있는 퍼블리싱 기능입니다.

6-2) 사용자가 직접 계층 구조를 만들고

6-3) 게시물을 저장한 하나의 도감을

6-4) 다른 사용자와 공유할 수 있습니다.

6-5) 사용자들은 마음에 드는 도감일 경우 '좋아요' 를 눌러 소통하며, 공유된 도감을 통해 특정 주제에 대한 게시물이 잘 분류되어 저장된 컬렉션을 볼 수 있는 것입니다.

7) 즉, MoaYo는 인스타그램의 여러 문제점을 보완하면서 더 나은 서비스를 제공하는 플랫폼입니다. 특히 한번에 하나의 해시태그만 검색이 가능하며, AND OR 연산을 거친 정확한 검색 결과를 받아볼 수 없다는 검색의 문제점에 주력했습니다. 또한 게시물 크롤링시 인스타그램에서 제공하는 오픈소스 API는 제공하는 정보량이 너무 적다는 점을 고려하여, 직접 메타 검색 로직을 구현함으로써 정확하고 많은 양의 정보를 얻을 수 있게 하고, 나아가 저장과 공유의 문제점을 보완하여 MoaYo만의 서비스를 제공하고자 합니다.

*****수정*****

8) 메타검색의 구현입니다.

8-1) 사용자에게 의해 도감 요소 중

8-2) 하나의 키워드가 선택되면,

`8-2)선택된 키워드부터 최상위 키워드까지의 경로를 객체 화하여 백엔드로 검색결과를 요청합니다. 이때, 키워드는 곧 태그 명입니다. 백엔드에서는 전달받은 각 태그명과,

8-3) TagsFinder에서 크롤링한 각 태그의 유의어를 제이슨 형태로 변환하여 검색 서버에 정보를 보냅니다. 이 작업은 Retrofit을 이용한 HTTP통신으로 합니다. 검색 서버는 전달받은 데이터를 기반으로

8-4) 인스타그램에서 게시물들을 크롤링하고, 태그와 연관 태그들의 교집합에 속하는 게시물들을 얻어냅니다. 크롤링과 메타 검색 로직이 단일 스레드로 처리하기엔 시간이 너무 오래 걸리는 문제가 있어 동시간대 작업에 능한

8-5) Go 언어의 장점을 살려

8-) 해결했습니다.

9) 계층 구조의 카테고리를 생성하고 게시물을 저장하는 기능의 구현입니다.

9-1) 사용자가 생성한 도감에 대한 정보는

9-2) DB테이블 세개로 나누어 저장합니다.

9-3) 도감 테이블에는 도감의 이름, 설명, 비밀번호, 아이디를 저장하고

9-4)

9-5) 카테고리 테이블에는 각 노드의 주제목, 레벨, 아이디, 직속 부모 계층의 아이디, 도감의 아이디를 저장합니다.

9-6)

9-6) 이렇게 저장된 도감을 이용하여

9-7) 게시물을 검색하고 이를 도감에 저장할 경우

9-7) 게시물 테이블에 저장됩니다.

9-8)

10) 도감 공유의 구현입니다.

10-1) 사용자는 자신이 만든 여러 도감 가운데

10-2) 공유할 도감을 선택합니다.

10-3) 도감의 내부는 이렇듯 계층 구조의 모습입니다.

10-4) 선택한 도감에 대해 백엔드로 공유를 요청합니다. 백엔드에서는 요청 받은 도감을 제이슨 형태로 변환하여 공유 서버에 보내고, 이 작업은 Retrofit을 이용한 HTTP통신으로 합니다. 공유 서버는 백엔드로부터 받은 데이터에 대해

10-5) 파싱 작업을 수행하여 ID값과 함께 DB에 저장하며, 저장된 값들은 바로 사용자들에게 공유 가능한 상태로 전환됩니다. 별도의 외장 서버를 설치할 필요가 없고 호환성 및 다양한 의존성을 가진

10-6) Spring boot를 이용하여 개발합니다. 언어는 자바, 데이터베이스는 MySQL을 기반으로 구성합니다.

11) 전체 시스템 구조입니다.

12) 구현 과정에서 변경된 사항입니다

백엔드에서 서버와 통신할 때, 코드를 간결화 시키고 속도를 향상시키기 위해 Square사의 오픈소스인 레트로핏을 사용하는 것으로 변경했습니다.

유의어 추출 시에 국립 국어원의 사전 API는 사전적 의미를 전달해줄 뿐 입력한 태그의 유의어를 주지 않기 때문에, TagsFinder에서 직접 크롤링하여 연관 유의어 3~5개를 추출하여 적용하는 것으로 변경했습니다.

데이터베이스에서는, 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 존재했습니다. 이에 따라 앱 내부의 데이터베이스인 SQLite를 사용하여 사용자가 입력한 태그 계층의 기록을 남겨 문제를 해결했습니다. 데이터베이스를 사용함에 따라 테이블의 구조에도 변경이 있었습니다.

또한 GraphQL EndPoint 캐싱을 위한 EndPoint값을 추가했습니다. 검색 서버에 요청하는 기존의 데이터 형식은 데이터 양이 많아질수록 속도가 저하됨을 발견하여, 이를 개선하기 위해 한정된 데이터를 먼저 응답 받고, 서버에서 검색을 멈춘 EndPoint 값을 받는 것으로 속도를 개선하기로 했습니다. 이에 따라 검색 서버의 Response Body에도 GraphQL EndPoint 값을 추가했습니다.

검색 서버에서는 인스타그램에서 제공하는 오픈소스 API가 제공하는 정보량이 너무 적다는 점을 고려하여 직접 구현하는 것으로 변경했습니다.

공유 서버에서는 게시물을 제이슨 파일로 저장하는 것에서 데이터 무결성을 고려하여 DB테이블에 저장하는 것으로 변경했습니다.

13) 시스템 구조로 비교하면 이와 같습니다.

14) 다음은 현재까지 구현 정도입니다.

14-1) 프론트엔드의 구현 정도입니다.

14-2) wireframe으로 설계한 7개의 액티비티 가운데 리사이클러뷰를 이용한 카드뷰 형식의 메인 액티비티, 프래그먼트를 전환하여 동적 페이지를 생성하는 나의 도감 관리 액티비티, expandable 리스트뷰로 도감의 계층 구조를 상세하게 볼 수 있는 도감 상세 액티비티, 메타 검색 결과를 보여주는 검색 결과 액티비티 총 네개의 액티비티를 구현했습니다.

14-3)

15) 백엔드의 구현 정도입니다.

15-1) 도감 관련 비즈니스 로직 및 스토리지 모듈의 구현을 완료했습니다.

15-2) 도감 테이블,

15-3) 카테고리 테이블의 구현 모습이며, 게시물을 저장하는 테이블은 구현 중에 있습니다. 또한 레트로핏을 이용한 서버와의 통신, 태그 유사어 크롤링 유틸, 데이터베이스 테이블 구현을 완료했습니다.

15-4) 태그 유사어 크롤링 결과입니다. 가장 연관된 태그로 판단되는 3~5개의 태그를 로직에 포함합니다.

16) 검색 서버의 구현 정도입니다.

16-1)

16-2) 인스타그램에서 제공하는 오픈소스 API는 제공하는 게시물 개수가 너무 적어서 직접 구현 하였습니다. 크롤링하면 게시물마다 게시물의 url, 커버사진 이미지 주소, 텍스트, 좋아요 개수를 가져옵니다. 해당 이미지는 인스타그램에 'daily'라는 태그로 검색한 결과를

16-1) 크롤링한 모습입니다.

16-3)

17) 현재까지 구현된 MoaYo 시연 동영상입니다.

18) 향후 계획입니다.

18-1) 와이어프레임으로 설계했던 7개의 액티비티 중 3개의 액티비티 구현을 앞두고 있습니다. 도감을 생성하는 액티비티인 BookFormActivity와 도감의 공유 기능을 담당하는 ShareMenuActivity, NewShareActivity 입니다. 백엔드에서는 변경 사항에서 언급했던 검색 서버와의 통신에서 시간을 줄이기 위한 비동기 처리를 앞두고 있고, 데이터베이스의 변경된 구조에 따라 매핑 데이터에 관련된 인터페이스를 구현할 예정입니다.

18-2) 검색서버는 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. 이에 따라 검색 서버는 향후 최적화 작업을 앞두고 있습니다. 최적화 완료 후에는 로컬 호스트에서 검색 서버를 구동하고 모바일 앱과 데이터 교환이 정상적으로 이루어지는지 테스트하고, AWS EC2 인스턴스를 생성하여 서버를 배포할 예정입니다.

현재 검색과 저장에 관련된 구현은 거의 완료된 상태이므로, 향후 공유 기능 개발에 주력할 예정입니다.

20) 마지막으로 MoaYo 프로젝트의 기대효과입니다. MoaYo앱 유저들이 각각의

20-1) 관심 분야를 도감으로 만들어 타 사용자와 공유하는 것에서 더 나아가,

20-2) 여러 분야의 전문가들이 MoaYo를 이용하여 인스타그램 게시물을 분류, 분석하는 플랫폼으로 활용함과 동시에, 이렇게 만들어진 전문가들의 도감을 일반 사용자들도 열람할 수 있는 데이터 공유의 장이 되고자 합니다.

21) 이것으로 23조의 발표를 마치겠습니다. 감사합니다.