JIKJI팀 중간발표

김정현 박종하 이한솔 김영현

목차

JIKJI

팀 소개

직지

BE
FE

BE
FE

BE
FE

BE
FE

JIKJI

Instagram

선정 이유

- 다양한 서비스들이 혼합되어 있어 MSA 구조로 개발하기 유리
- 심플한 UI => 백엔드 개발에 집중
- 참고 자료가 많아서 개발이 용이
- 대용량 트래픽을 가진 서비스의 서버 아키텍처를 구현해보면서 대형 서비스에서 서버 아키텍처를 선정, 구현하는 원리를 알아보고 싶음
- 여러 서비스들을 함께 개발하면서 협업에 대한 이해도 증진 및 소프트 스킬 습득 가능

팀 목표

전문적인 현업 개발 프로세스를 통해 완성도 높은 팀 프로젝트 경험

=> 지속 성장 가능한 개발자 되기

+

체계적인 협업 경험을 통해 소프트 스킬 습득

개인 목표

김정현

- 개발자의 공부 방법, 문제 해결 방식을 경험해 나만의 문제 해결 루틴을 찾고 레퍼런스 프로젝트로 활용할 수 있도록 만들기
- 버전 관리 및 문서 관리를 체계적으로 경험하고 팀을 위해 기록하는 습관 만들기

김영현

- git 관리 습관 형성
- MSA 직접 설계와 구현을 통한 MSA 철학 이해
- 기능 명세, 개발 과정, 문제 해결 과정 정리

이한솔

- MSA를 구현하고 배포하며 통신하는 방법 이해
- 클린코드 작성 원칙, 객체 지향 원칙 이해
- 개발일지 작성 기록하는 습관 형성

박종하

- 스프링을 활용한 백엔드 구현 능력 향상
- 개발 이슈 및 해결 과정 문서화
- MSA 아키텍쳐를 이해하고 구현

일정 관리 – Workflow

일일 스크럼 회의

시간 : 매일 아침 10시

장소 : 슬랙 허들

회의 안건 :

- 전날 스크럼 회의에서 세운 목표사항 완료 여부 체크
- 새로운 일일 목표 공유
- 문제 상황 공유

주간 스프린트 회의

시간 : 금요일 저녁 8시

장소 : 슬랙 허들

회의 안건 :

- 스프린트 진행 상황 점검
- 마일스톤 환기
- 문제 상황 공유

일정 관리 – Workflow

팀빌딩

주기적으로 팀의 목표와 개인의 목표를 되돌아보는 시간 가지기

협업을 위해 필요할 경우 회의를 주최하고 되도록 참석하기

개발 상황에 대해 솔직하게 이야기하고 서로 신뢰하기

그라운드룰

회의-슬랙, 노션-자료공유

만장일치로 의사 결정하기

네이버 자바 코딩 컨벤션 활용

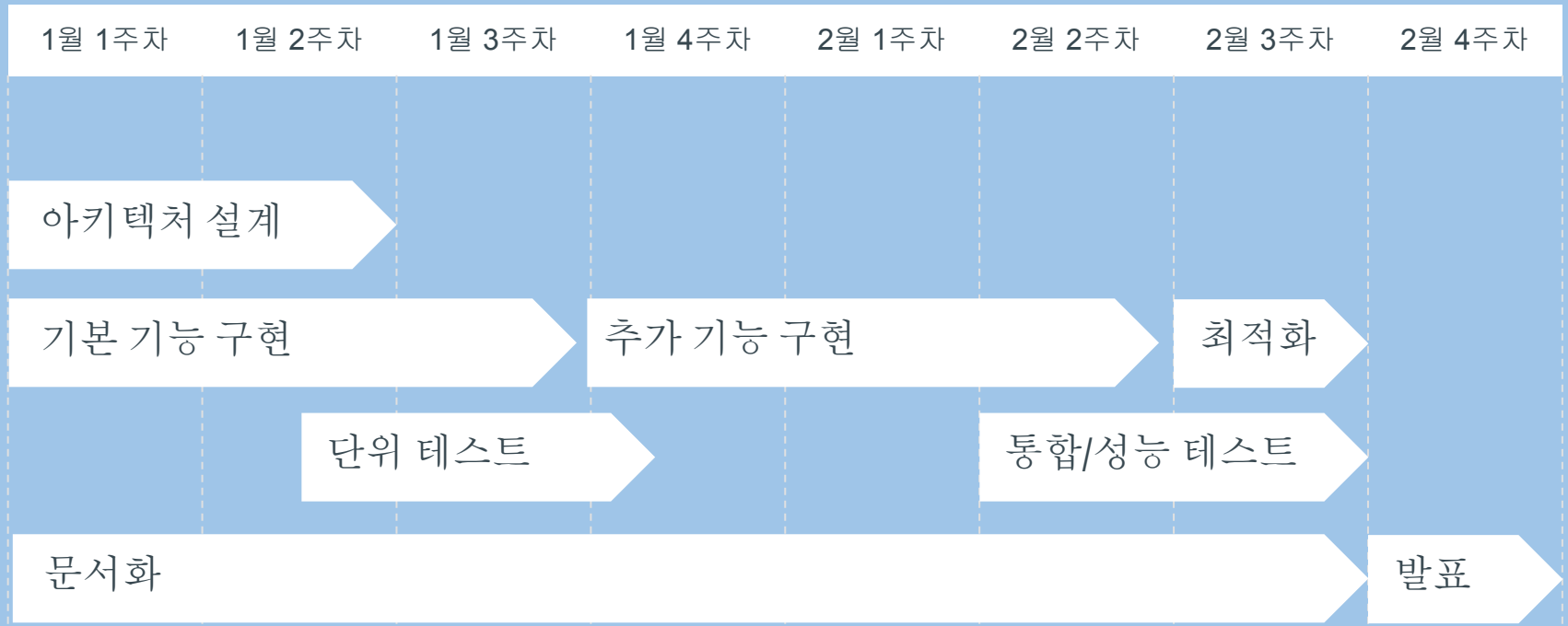
결석 많이 하지 않기

리뷰시 서로 존중하기

Milestone

1. SNS 기본 기능 구현 - Auth, Content, Feed, Media, Follower 서버
2. 추가 기능 구현 - Notification, Search, Chat, Story 서버
3. 테스트 및 성능 최적화

개발 계획


1월 1주차
1월 2주차
1월 3주차
1월 4주차
2월 1주차
2월 2주차
2월 3주차
2월 4주차
아키텍처 설계
기본 기능 구현
추가 기능 구현
최적화
단위 테스트
통합/성능 테스트
문서화
발표

JIKJI

Auth Server(User)

- [~1/3] 엔티티 설계 및 요구사항 분석
- [~1/6] API 명세 및 구조 설계
- [~1/13] React 학습 및 API 개발
- [~1/17] 프론트엔드, 백엔드 연동
- [~1/18] 이슈 처리 및 부가 기능 설계
- [~2/23] 코드 리팩토링
- [~2/23] API 수정

개발 계획

Graph(Follow) Server

- [~1/4] Graph DB 사용법 학습
- [~1/6] 요구사항 및 엔티티 설계
- [~1/12] Neo4j Repository 활용 Server 개발
- [~1/13] API 명세서 및 개발 문서 작성
- [~1/18] React.js 학습 및 프론트엔드 개발
- [~1/20] 프론트엔드, 백엔드 연동 완료

Story Server

- [~1/27] 요구사항 및 엔티티 설계
- [~2/3] Spring data jpa 활용해서 Server 개발
- [~2/6] API 명세서 및 개발 문서 작성
- [~2/9] 프론트엔드 개발
- [~2/13] 프론트엔드, 백엔드 연동 완료

개발 계획

Content Server

- [~1/9] Content 도메인 요구사항 분석
- [1/10 ~ 12] 아키텍처, DB, API 설계
- [~1/14] CQRS 및 MSA 패턴 공부
- [~1/16] API 구현
- [1/17] 기능 테스트
- [~1/19] react 컴포넌트 구현

Search Server

- [~1/9] Search 도메인 요구사항 분석
- [~1/20] Elasticsearch 공부
- [1/24] 요구사항 및 설계 재 점검
- [1/24 ~ 26] 아키텍처, DB, API 설계

개발 계획

Media Server

- [~1/4] 요구 사항 및 엔티티 설계
- [~1/6] AWS S3 개념 및 이론 학습
- [~1/12] S3 활용하여 이미지 업로드 구현
- [~1/15] API 명세서 및 개발 문서 작성

Comment Server

- [~1/6] 요구사항 및 엔티티 설계
- [~1/13] Comment Server 개발
- [~1/15] API 명세서 및 개발 문서 작성
- [~1/18] 프론트엔드, 백엔드 연동 완료
- [~1/21] React.js 학습 및 프론트엔드 개발

Feed Server

- [~1/6] 요구사항 및 엔티티 설계
- [~1/12] Feed 성능 높일 수 있는 알고리즘과 DB 관련 이론 학습
- [1/20 -] 중간 문서 발표 이후 구체화

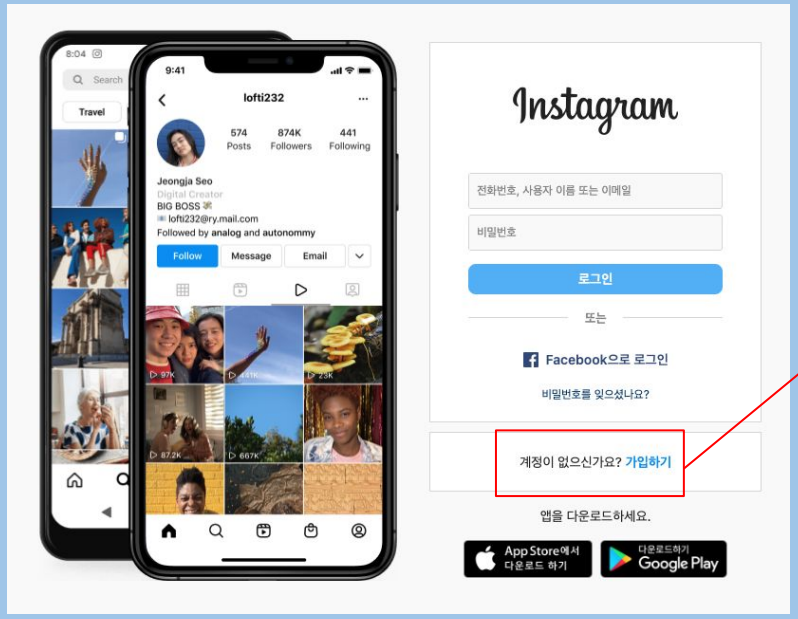
개발 진행 상황

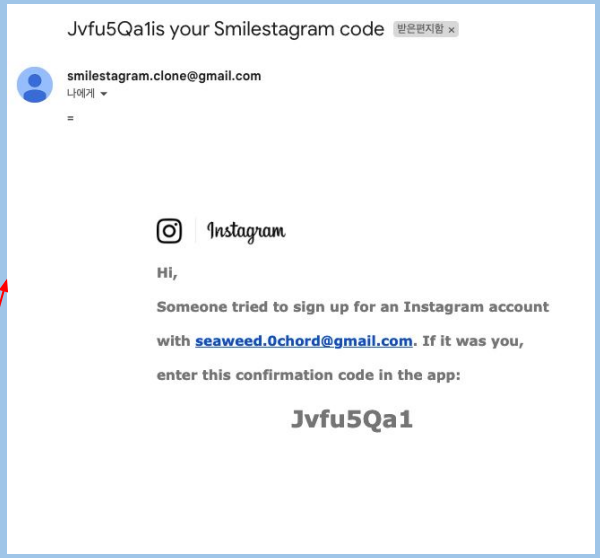
👩‍💻BE	🖥️FE
요구사항 작성	요구사항 작성
아키텍쳐 설계	UI 컴포넌트 구현
API 명세 작성	
DB 설계	
기본 기능 개발중	

업무 분담

Auth(User) 김영현	Media/Comment 박종하	Follower 김정현
Content/Hashtag 이한솔	Feed 박종하, 김정현	Notification 김정현, 김영현
Search 이한솔	Chat 이한솔, 김영현	Story 김정현, 박종하

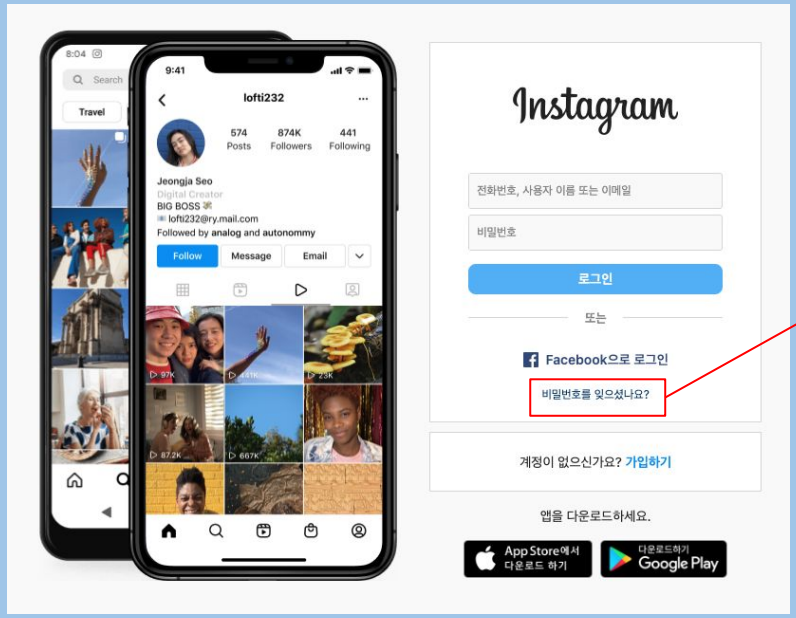
스토리보드 – Auth

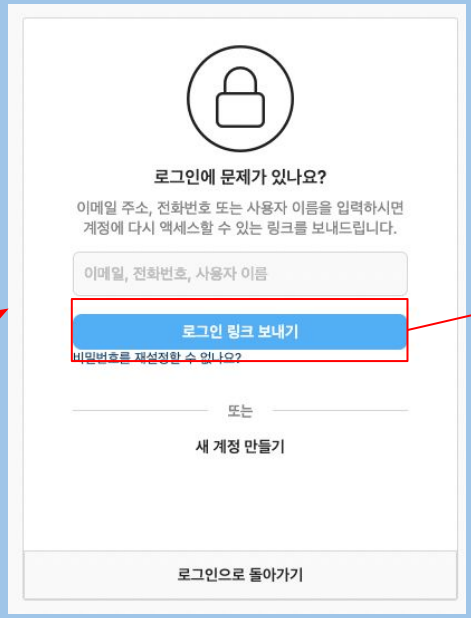


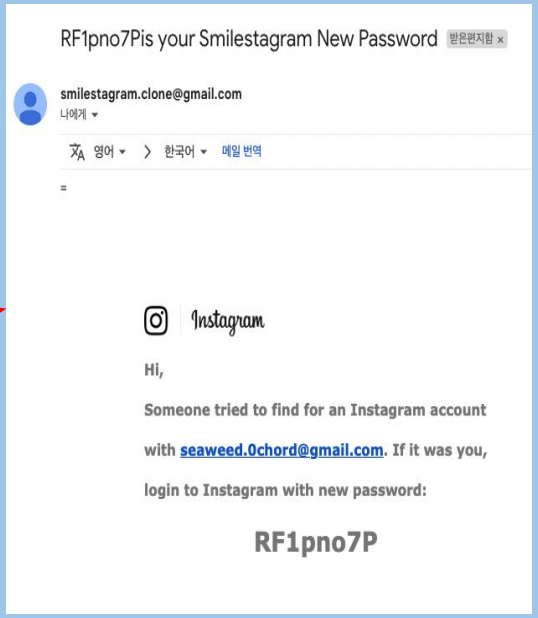


JIKJI

스토리보드 – Auth

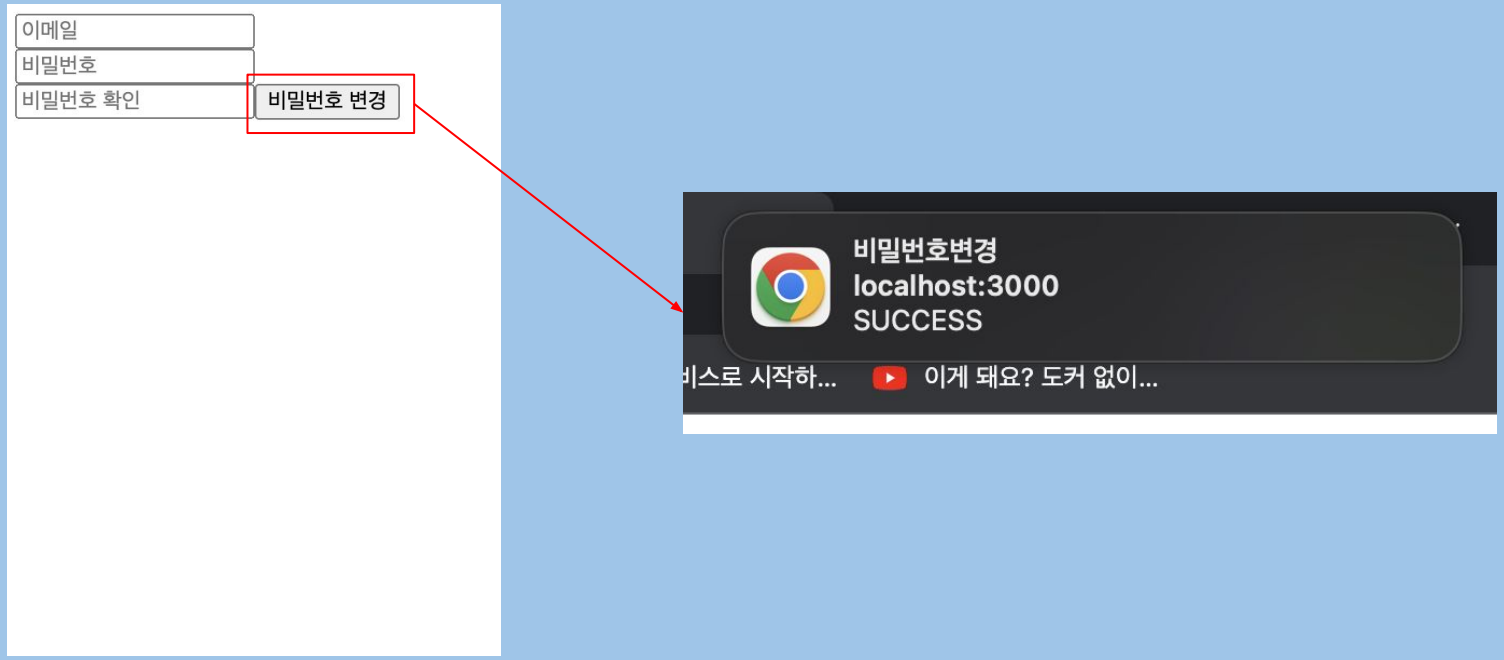




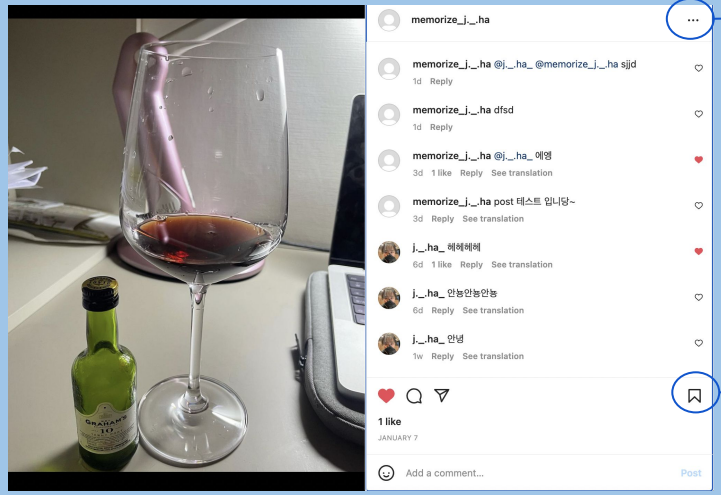


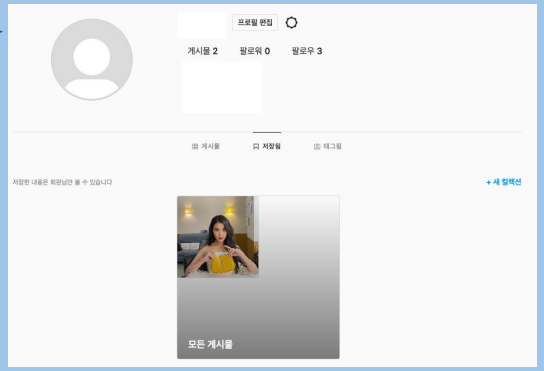
JIKJI

스토리보드 – Auth


이메일
비밀번호
비밀번호 확인
비밀번호 변경
비밀번호변경
localhost:3000
SUCCESS
이게 돼요? 도커 없이...

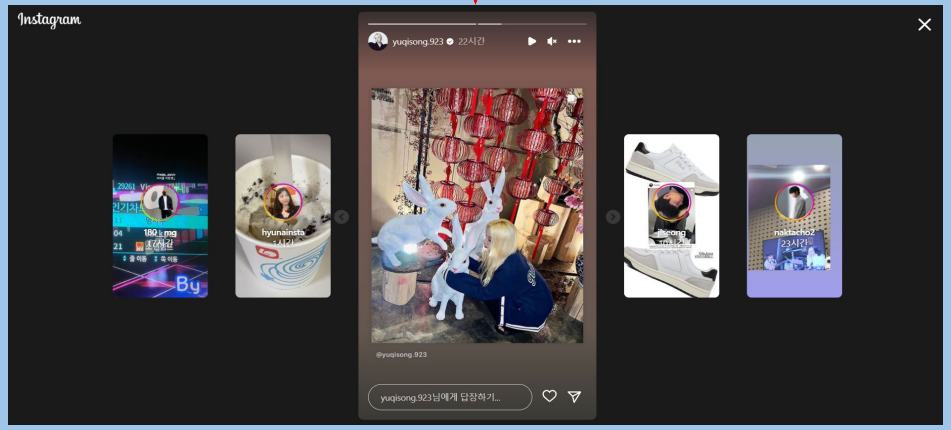
스토리보드 – Post, Comment





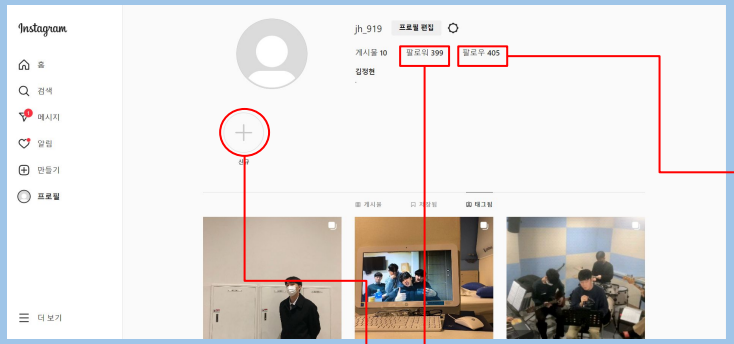
JIKJI

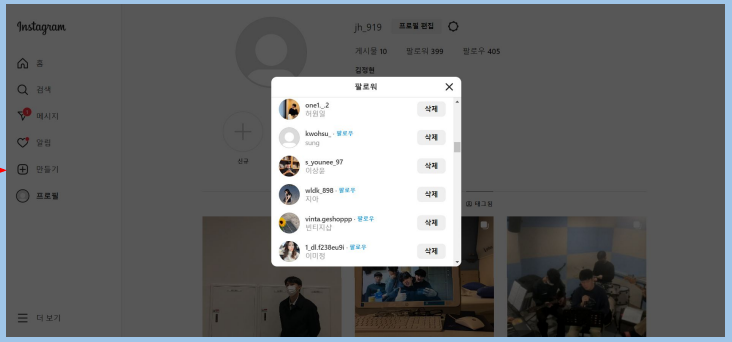
스토리보드 – 피드, 스토리

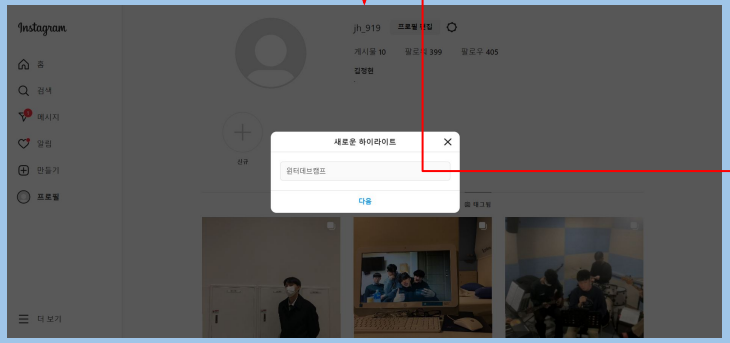


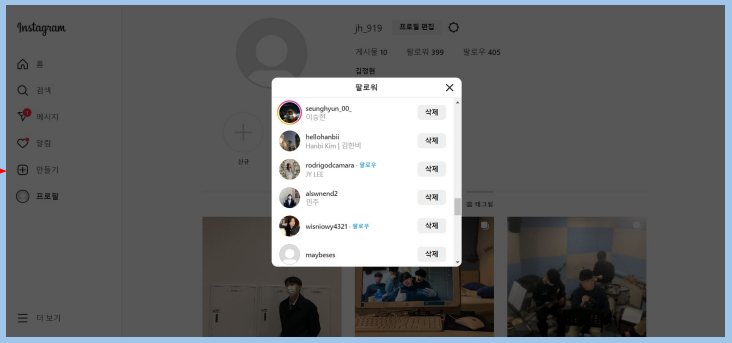
JIKJI

스토리보드 – 프로필, 팔로우









JIKJI

기술 스택

Language

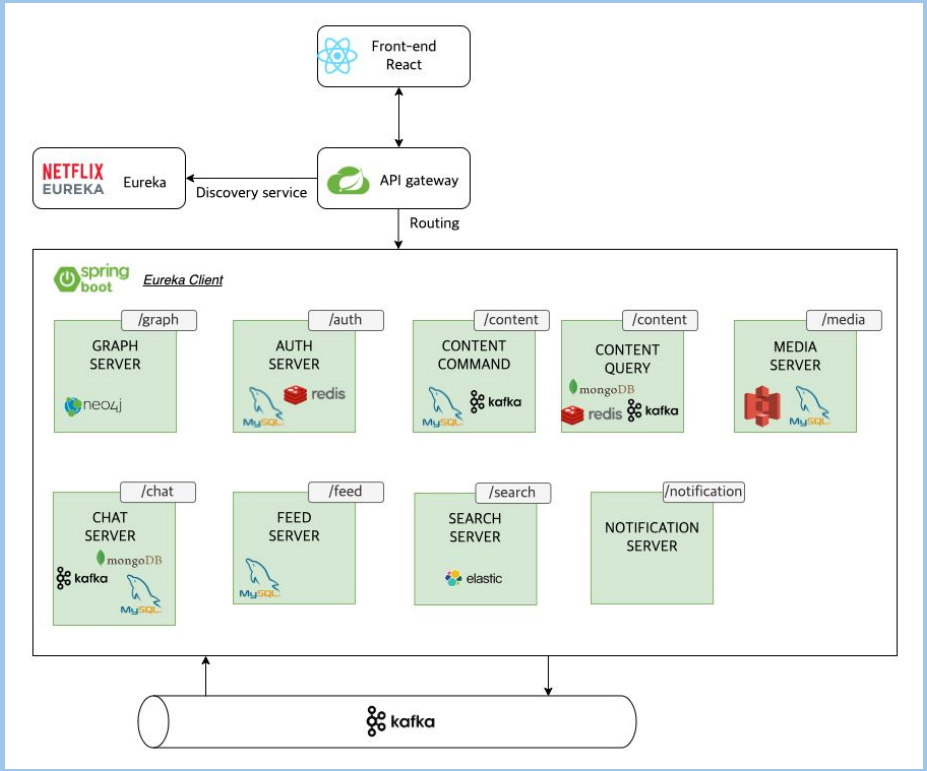
Framework

* react 17

Database

DevOps

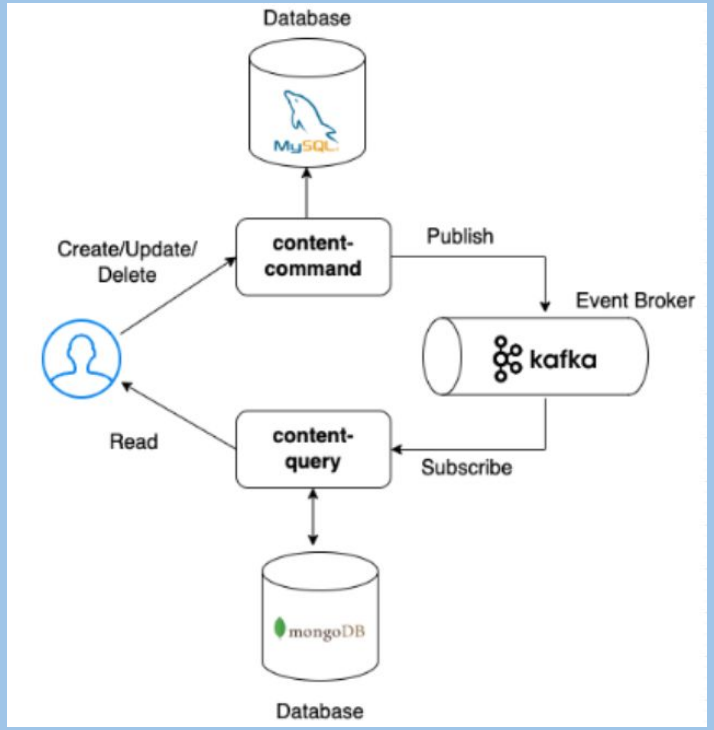
서비스 구조


Front-end
React
NETFLIX
EUREKA
Eureka
Discovery service
API gateway
Routing
spring boot
Eureka Client
/graph
GRAPH SERVER
neo4j
/auth
AUTH SERVER
redis
MySQL
/content
CONTENT COMMAND
kafka
MySQL
/content
CONTENT QUERY
mongoDB
redis
kafka
/media
MEDIA SERVER
MySQL
/chat
CHAT SERVER
mongoDB
kafka
MySQL
/feed
FEED SERVER
MySQL
/search
SEARCH SERVER
elastic
/notification
NOTIFICATION SERVER
kafka

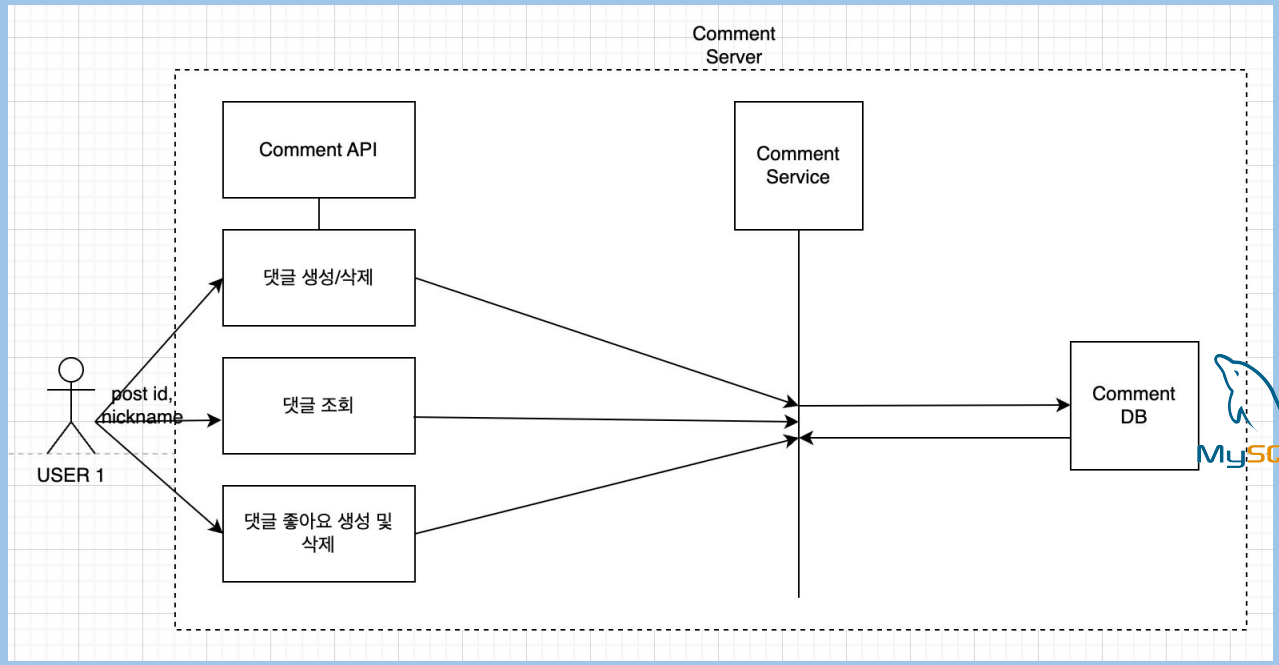
서비스 구조

Content

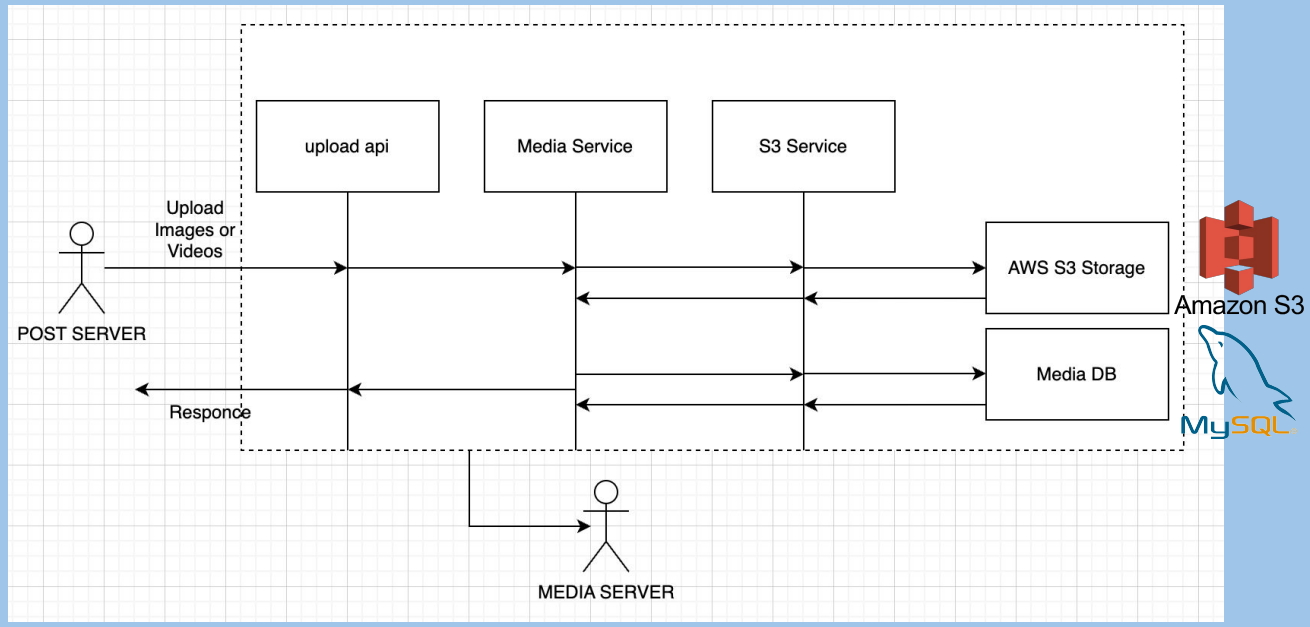
CQRS 패턴 적용



Comment

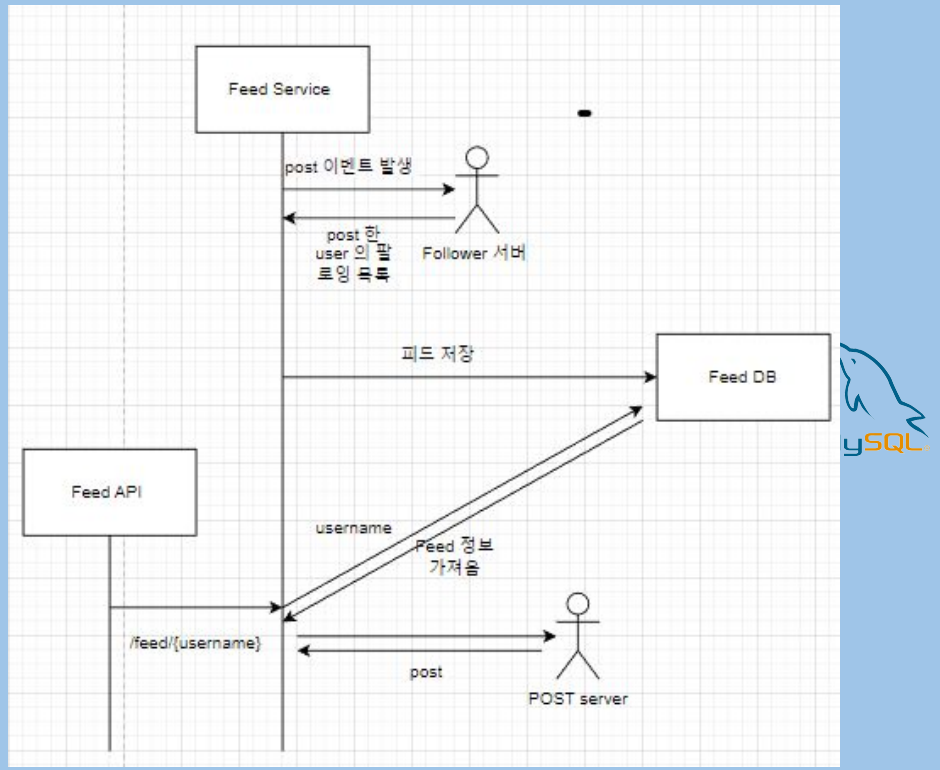

Comment
Server
Comment API
Comment
Service
댓글 생성/삭제
댓글 조회
댓글 좋아요 생성 및
삭제
post id,
nickname
USER 1
Comment
DB
MySQL

Media

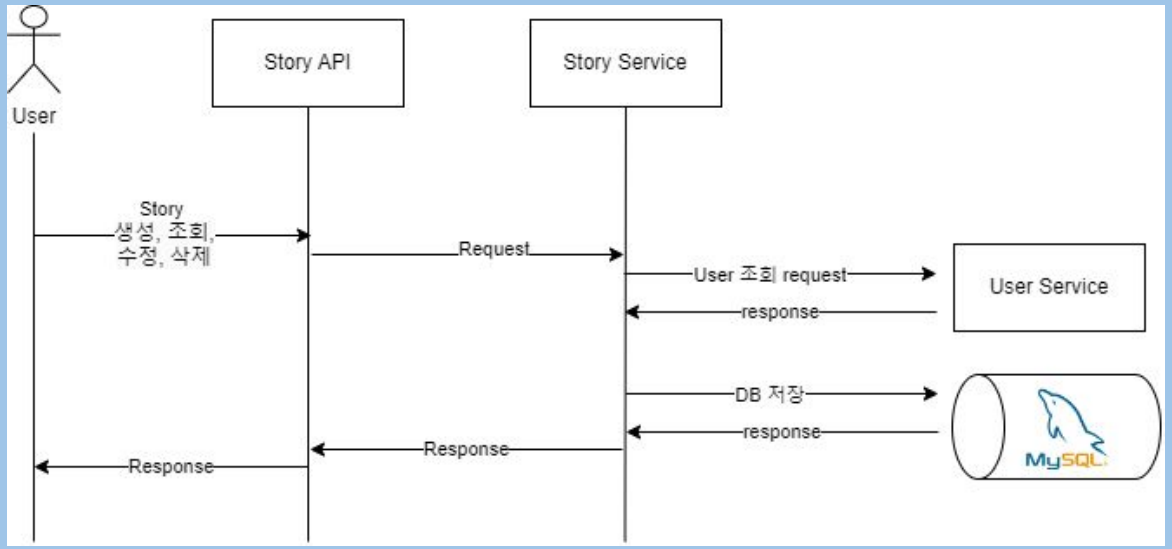

upload api
Media Service
S3 Service
Upload
Images or
Videos
POST SERVER
AWS S3 Storage
Amazon S3
Media DB
MySQL
Responce
MEDIA SERVER

서비스 구조

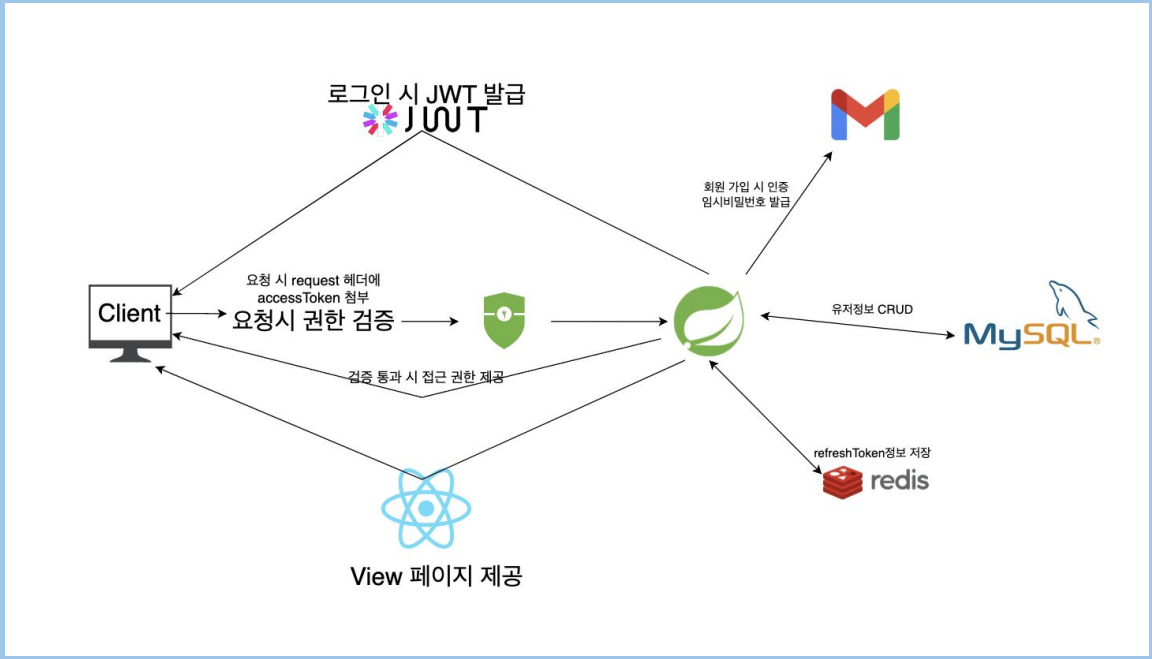
Feed


Feed Service
post 이벤트 발생
post 한
user 의 팔
로잉 목록
Follower 서버
피드 저장
Feed DB
Feed API
username
Feed 정보
가져옴
/feed/{username}
post
POST server

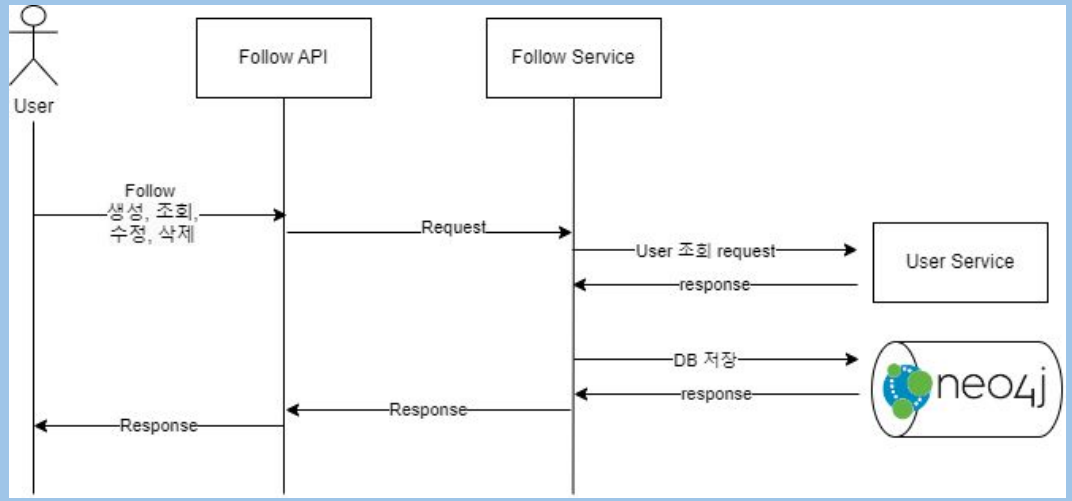
Story


User
Story API
Story Service
Story
생성, 조회,
수정, 삭제
Request
User 조회 request
response
User Service
DB 저장
response
Response
Response
MySQL

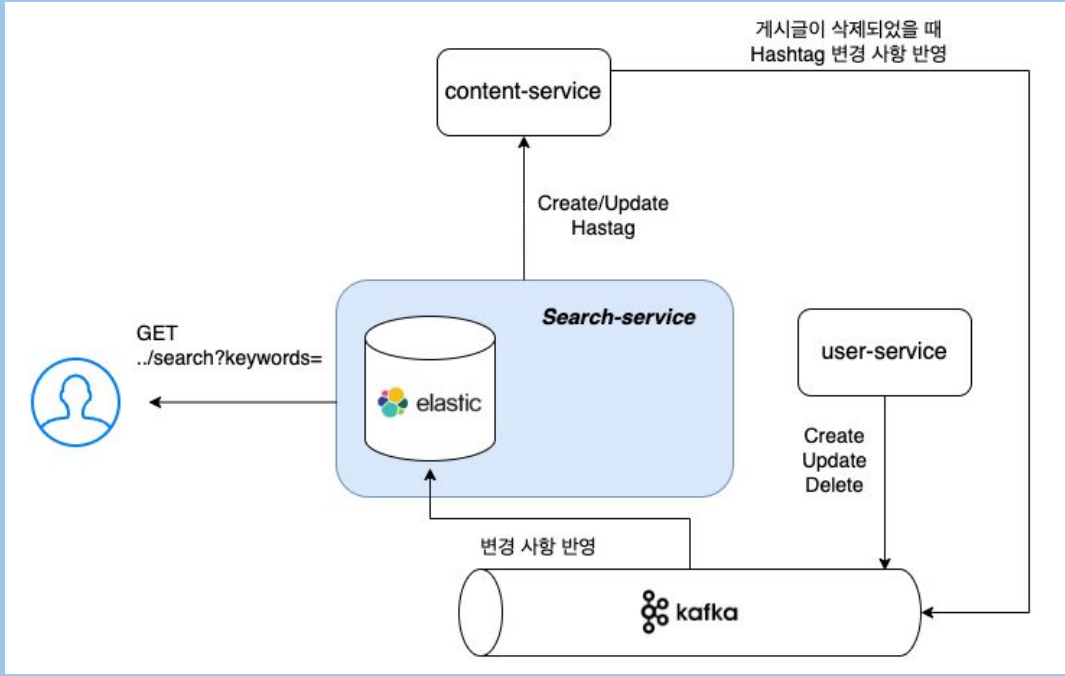
Auth(User)


로그인 시 JWT 발급
JWT
회원 가입 시 인증
임시비밀번호 발급
요청 시 request 헤더에
accessToken 첨부
Client
요청시 권한 검증
유저정보 CRUD
MySQL
검증 통과 시 접근 권한 제공
refreshToken정보 저장
redis
View 페이지 제공

Follow


User
Follow API
Follow Service
Follow
생성, 조회,
수정, 삭제
Request
User 조회 request
response
User Service
DB 저장
response
neo4j
Response
Response

Search


게시글이 삭제되었을 때
Hashtag 변경 사항 반영
content-service
Create/Update
Hastag
Search-service
GET
../search?keywords=
elastic
user-service
Create
Update
Delete
변경 사항 반영
kafka

DB 구조

Contents

Contents Entity

Aa 이름	속성	type	Description
id	PK	bigint	
user_id	NOT NULL	bigint	
text		varchar(3000)	게시글 본문
image_url	NOT NULL	text	첨부된 사진 url list
likes		int	default = 0
visible_likes		bit	deafult = true
visible_comments		bit	default = true
created_at		datetime(6)	
modified_at		datetime(6)	

command

Contents Document

Aa 이름	속성	type	description
id	ObjectId	String	
content_id	unique	Long	
user_id		Long	
text		String	
image_urls		List	url(String), order(int), user_id(Long)
likes		int	
visible_likes		boolean	
visible_comments		boolean	
created_at		String	
modified_at		String	

query

JIKJI

Bookmark

Bookmark Entity

Aa 이름	type	속성
id	bigint	PK
user_id	bigint	index NOT NULL
content_id	bigint	index NOT NULL

MySQL command

Bookmark Document

Aa column	type	속성
id	ObjectId	ObjectId
comment_count	int	
likes_count	int	
content_id	Long	index
user_id	Long	index

mongoDB query

Contents-like

Content_likes Entity

Aa 이름	type	속성
id	bigint	PK
user_id	bigint	index NOT NULL
content_id	bigint	index NOT NULL

command

Content_likes Document

Aa column	type	속성
id	String	ObjectId
user_id	Long	index
content_id	Long	index

query

DB 구조

Comments

Comment Entity

Aa 이름	속성	type	description
id	PK	Long	comment id
user_id	FK	Long	user 의 id
post_id	FK	Long	post 의 id
created_at		TIMESTAMP	생성한 날짜
Likes		Int	좋아요 수

Comment_mention Entity

Aa 이름	속성	type	description
id	PK	Long	comment mention id
receiver_id	FK NOT NULL	Long	receiver 의 id
sender_id	FK NOT NULL	Long	sender의 id
created_at		TIMESTAMP	created 된 날짜
comment_id	FK NOT NULL	Long	comment 의 id

Comment_likes Entity

Aa 이름	속성	type	description
id	PK	Long	comment mention id
user_id	FK NOT NULL	Long	user의 id
comment_id	FK NOT NULL	Long	comment 의 id

JIKJI

Feed

Feed Entity

Aa 이름	속성	type	description
id	PK	LONG	feed의 id
userId	FK	LONG	유저의 id
postId	FK	LONG	post id
createdAt		TIMESTAMP	생성된 날짜
username		NOT NULL	유저 이름

Media

Media Entity

Aa 이름	속성	type	Description
id	PK	LONG	미디어의 ID
created_at		TIMESTAMP	미디어 create 날짜
url	NOT NULL	String	미디어의 URL
type	NOT NULL	ENUM	IMAGE, VIDEO 타입
user_id	NOT NULL FK	LONG	

DB 구조

Stories

Story Entity

표 +

Aa 이름	속성	type	Description
id	PK	LONG	story의 id
user_id	not null FK	LONG	스토리를 올린 사용자의 id
image_url	not null	String	gcp or aws storage 업로드된 링크
created_at	not null	TIMESTAMP	스토리가 생성된 날짜, 시간
origin	FK	LONG	공유한 스토리일 경우 원본 사용자의 id

Story-likes Entity

표 +

Aa 이름	속성	type	Description
id	PK	LONG	id
story_id	not null FK	LONG	좋아요가 달린 스토리의 id
likes_user_id	not null FK	LONG	좋아요를 누른 유저의 id

Story-usertags Entity

표 +

Aa 이름	속성	type	Description
id	PK	LONG	id
story_id	not null FK	LONG	태그가 달린 스토리의 id
user_id	not null FK	LONG	스토리가 태그한 유저의 id
location_x	not null	int	스토리에서 태그 위치의 x축 값
location_y	not null	int	스토리에서 태그 위치의 y축 값

DB 구조

Auth

User Entity

Aa 이름	속성	type	Description
id	PK	Long	
email	not null	String	이메일 형식만 가능 중복안됨
password	not null	String	특수문자 포함 8~20자만 가능
nickname	not null	String	다른 닉네임과 중복 안됨
phone_number	not null	String	전화번호 형식만 가능
email_auth	not null	Boolean	default=false, 이메일 인증
created_at	not null	Date	생성일
user_profile		String	default=회색 바탕
updated_at	not null	Date	업데이트 시행일
login_at	not null	Date	로그인 접속일

Mail_auth Entity

Aa 이름	속성	type	Description
id	PK not null	Long	
user_email	not null	String	
code	not null	String	이메일 인증 번호

Refresh_token Entity

Aa 이름	속성	type	Description
refresh_token_id	PK not null	Long	
user_email	not null	String	
refresh_token	not null	String	refresh_token 정보

User_roles Entity

Aa 이름	속성	type	Description
user_email	PK not null	Long	
roles	not null	List	유저 역할

JIKJI

Follow

Follow Entity

Aa 이름	속성	type	Description
id	PK	LONG	follow의 id
follower_id	not null	LONG	follow하는 유저의 id
followed_id	not null	LONG	follow당하는 유저의 id
created_at	not null	TIMESTAMP	follow가 생성된 날짜, 시간

Search

표

Aa 이름	태그	참고
id	KEY	
user_id	string	사용자 아이디
username	string	

표

Aa 이름	태그
id	KEY
hashtag_name	string

QnA