

아키텍처 문서 리뷰 1차 피드백

📅 날짜	@2023년 1월 13일
👤 작성자	
☰ 안건	리뷰 아키텍처
👤 참석자	

아키텍처 페이지

- 아이콘이 크고 많다(가독성↓)
 - 시선이 Node Port쪽으로 많이 간다
 - 그러나 제일 중요한 건 서버 그룹이다
 - 서버 그룹은 일렬로 돼있고, db/메시지브로커는 넓은 공간에 있다
- ▼ 해결책: db/메시지브로커 부분을 줄이고, 서버그룹을 예쁘게 꾸미고, 서비스 부분의 아이콘 부분을 정리하기
- ingame info부분이 이해가 안된다 가독성 부분을 많이 고쳤으면 좋겠다
 - mysql이 각각 하나 붙을 이유가 없을 것 같다, redis도 마찬가지
 - infra structure가 모든 것을 지배하는 거 같다 이 부분을 아키텍처 정리를 했으면 좋겠다
 - ingame info부분은 서버4~5개를 말았는데 그중에 하나면 ok,
 - 그런데 이게 유의미한 정도로 팀프로젝트에 범위를 차지하면 아쉬울 것 같다
-

전체구조 페이지

- 캠프의 방향성이 개발, 데이터 교환, 연동, 성능, 안정성, 코드 quality 등 개발자로서 경험할 수 있는 부분을 추구하는데 infra structure의 비중이 높지 않을까하는 느낌이 든다
 - 실제로 구현해야하는 부분은 간단한 애들이 기능만 있다(chat server는 어려울 수 있는데)
 - stove를 클론한다하면 이런 아키텍처가 나올 수 없다
 - 이게 stove를 클론하는게 맞는지의 의문이다
-

도메인설계 페이지

- 다대다 채팅이 되면 난이도가 급상승한다
- 1대1 채팅이면 사실 쪽지다
- 기술적으로는 할 게 없다
- db로따지면 target id, message, datetime이 있으면 끝이다
- 그게 계속 번갈아가면 최신 거의 select가져오면 그게 1대1 채팅이다
- 다대다채팅은 댓글의 연속인 경우도 있다
- 유튜브, 트위치는 채팅이 아니라 댓글을 계속 달고 있는 것과 마찬가지로이다
- 어떤 채팅을 생각하는 건지?

→ 정현님: 우리는 카카오톡, 슬랙 같은 실시간 채팅을 생각했다

- 그러면 난이도가 높아진다
- 생각할게 많아진다
- 기술적으로 인증서비스는 구현다했고

- 인증은 조금만 더 확장시키면 될겁니다
- 그 상태에서 유저서버에서 조금 신경쓸 수 있는게 친구관계
- 팔로우, 피드형태를 생각하고있는건지

→ 서영님: 팔로우 제외, 친구, 피드 형태로 보여줄 것을 생각했다

- 포스트, 좋아요, 댓글, 유저가 1인프로젝트때 블로그나 인증을했으면 어느정도 했던거다
- 랭킹시스템은 redis사용하면 랭킹을 잡아준다
- 애네들은 사실 기술적으로 접근할게 거의 없다
- 기술적으로 접근할만한건 채팅, 머신러닝, 알림 정도이다
- 머신러닝 → 현성님
- 상우님, 정현님이 채팅이나 나머지 부분을 주로 담당할 것 같다
- 현성님이 해야할거: garbage director? data generator를 개발해야한다
- generate하고, generate데이터 뽑아와서 추천 알고리즘 쓰기
- 상우님, 정현님은 일반적인 기능을 빠르게 끝내고 채팅을 어떻게 가져갈건지 고민하기

DB구조

현성님 부분

- char_name, char_level 등 앞에 char을 붙여야하는가? 그냥 빼는게 나을거같다는 생각이 든다
- name이 길면 꺼내쓰기 어렵고 트래픽이 증가한다
- 타입이 정확했으면 좋겠다 String → Varchar 등으로
- int → small int, tiny int로 줄여도 될거 같다

- 장비 type은 그냥 type으로쓰기보단 equip_type으로 쓰는게 좋다
- type이 예약어로 설정된 게 많기때문에 장애가 날 수 있다
- db에는 camelcase안쓴다 isSet → is_set

상우님 부분

Q: 친구는 일방향인가요 양방향인가요?

A: 일단 친구요청보내면 수락받는식으로 했다

Q: 트위터 같은 팔로우 관계는 아닌가요?

A: 일단 친구로 생각했는데 확장할 생각이 있습니다

이거에 따라 friend의 구조가 바뀔 수 있다

- friend a와 b가 친구면 friend1에 a가 들어가고 friend2에는 b가 들어가서
 - 조회할때 두개의 필드를 조회해야한다
 - 아예 key자체를 이런 경우 두개를 합친 형태(a,b)로 가져갈 수 있다
 - 필드에는 친구 요청 시간, 친구 수락시간등이 들어간다
-
- authrotiy 경우에는 여러개의 권한을 가질땐 array형태로 넣으면 된다(이게 되는줄 몰랐다?)
 - 아니면 콤마형태로 넣고 분해해서 써도 되고
 - 이렇게 되면 유저테이블 가져올때 join해서 가져와야한다
 - array형태나 json형태로 가져와도 될 거 같다
-
- comment은 잘못된 설계이다
 - 대댓글을 달거면 parent_id같은거를 줘야한다

- linked list구조를 가져가면된다
- 여차하면 대댓글을 없애는 것도 좋다
- 이걸 구현해도 그 누구도 감탄하지 않는다
- 댓글 Varchar(100) → Varchar(255)정도로 늘리기?

서영님 부분

- 이름 camelcase적용하지 않고 snakecase로 바꾸기
- postcontent, posttitle 이런식으로 하지말고 content, title로 바꾸기
- timestamp와 datetime의 차이
- timestamp는 유닉스 타입 → 1970년부터 지금 이 시간까지 몇 밀리세컨드지났나 나타낸다
- datetime은 시간계산하는 것은 맞는데, utc타임도 넣을 수 있다
- 웬만하면 timestamp보단 datetime이 좋다
- 모든 db에 utc타임을 0으로 넣는다
- 중국, 한국에서 저장하면 시간이 local에 달려있기 때문이다
- postdate → postat으로 이름 바꿨으면 좋겠다
- content type varchar(255) → text필드로 가져가기
- 좋아요는 postlike, commentlike따로 관리하는게 낫다
- 게시글 하나에 좋아요가 10000개가 달리면 10000개의 컬럼이 생긴다 괜찮을까?
- 해결방법: like user id를 json형태로 넣기
- type을 json으로 하고 json데이터 속에 userid들을 넣으면 된다
- nosql을 쓰는 이유가 여기서 나온다

정현님 부분

- chatlist는 완전히 망했어요
- chatlist를 저렇게 하면 단독방이 100명이면 컬럼이 100개가 쌓인다
- 유저기반으로 채팅방을 가져가기

카카오톡의 채팅방목록생각하면

채팅방을 돌면서 너가 포함된 채팅방은 이거야 라고 select한게아니라

내가 유저기반으로 채팅방이 뭐뭐열려있는지 찾은거다

- chatroom 채팅방제목넣기
- chatroom 각자바꾸기로 하지말고 fix하기 복잡하니까

채팅은 그냥 다시 설계하기ㅠㅠ

업무분담

- 서영님 OK
- 현성님 MSA연동 다같이하는거니까 빼고 data generation넣기
- 상우님 약간 애매한디 딱보면 상우님이 애매하다 많을거같으면서도 필드자체는 신경쓸게생기면서도

그게 어떻게 되냐에따라 굉장히 빨리해놓고 혼자 놀 수 있을거같기도

내가 빨리하고 정현님을 도와줘야겠다

- 정현님 채팅이 어마어마할수도 있고 아닐수도있다 어마어마하게 키우면 굉장히 해볼만한 프로젝트가 될 거 같다

