



답글형 게시판 설계 및 구현

커머스개발
우영선





목차



▶ 설계

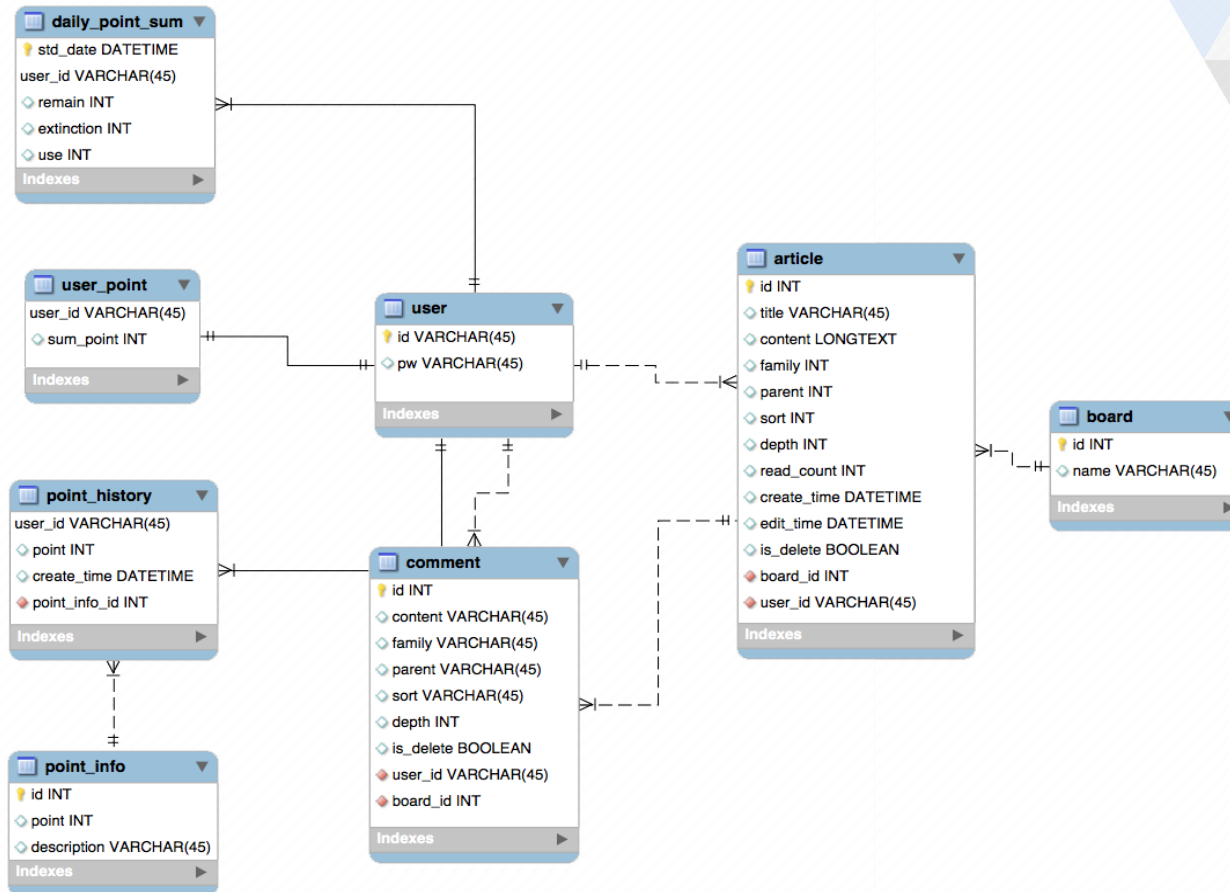
- E-R Diagram
- Class Diagram
- Usecase

▶ Review

- 중점구현항목&어려웠던점
- 추가개선요소

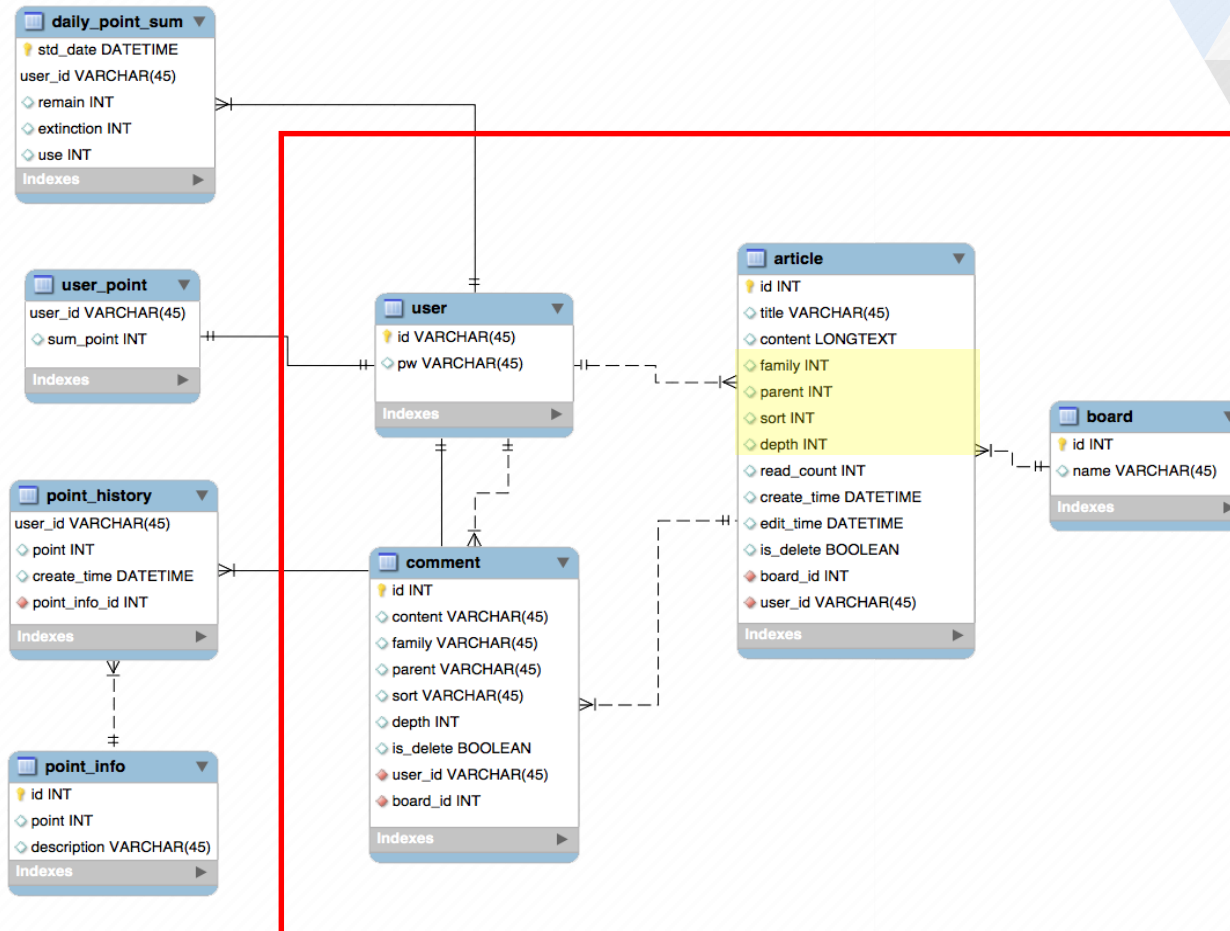


► E-R Diagram



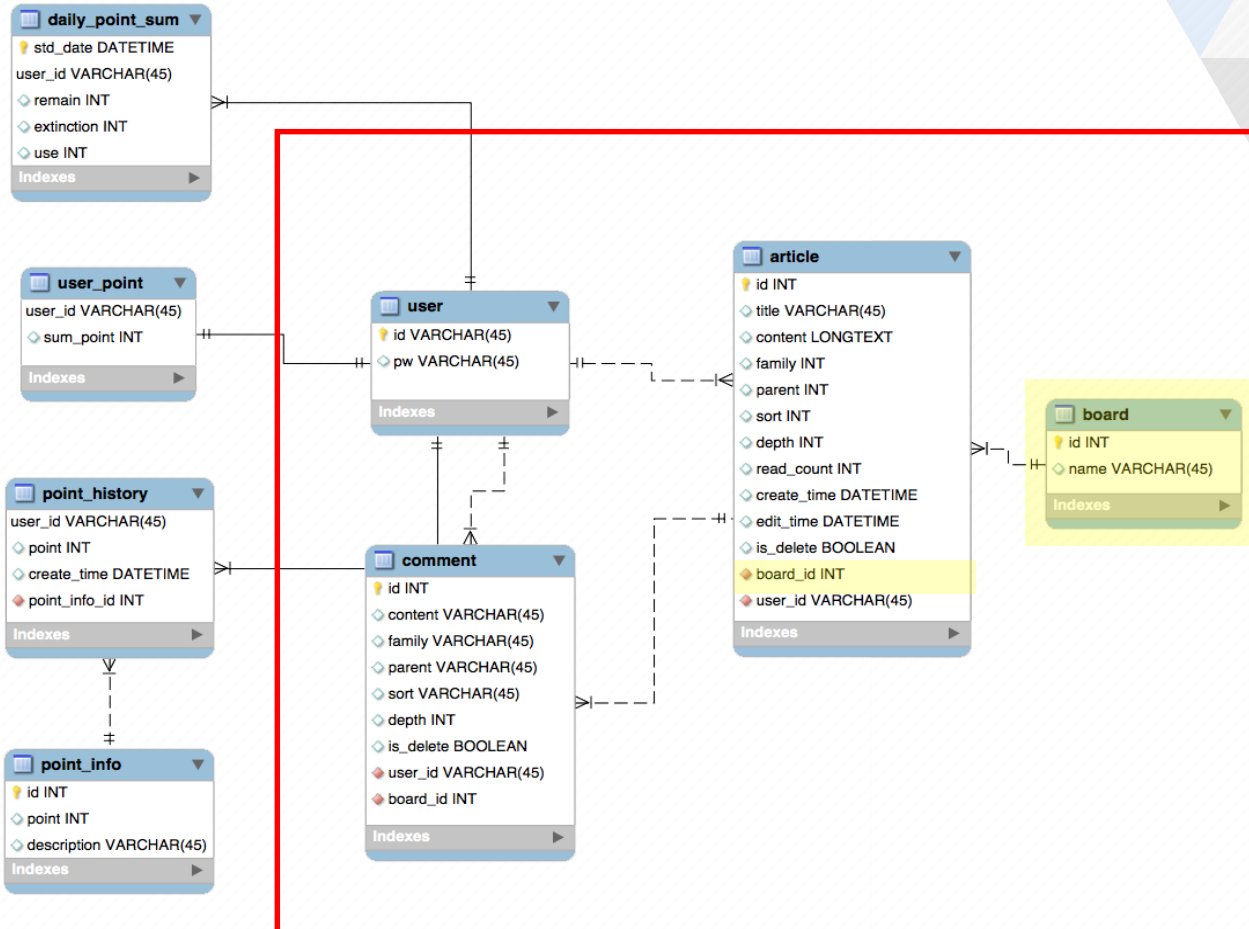
설계

▶ E-R Diagram

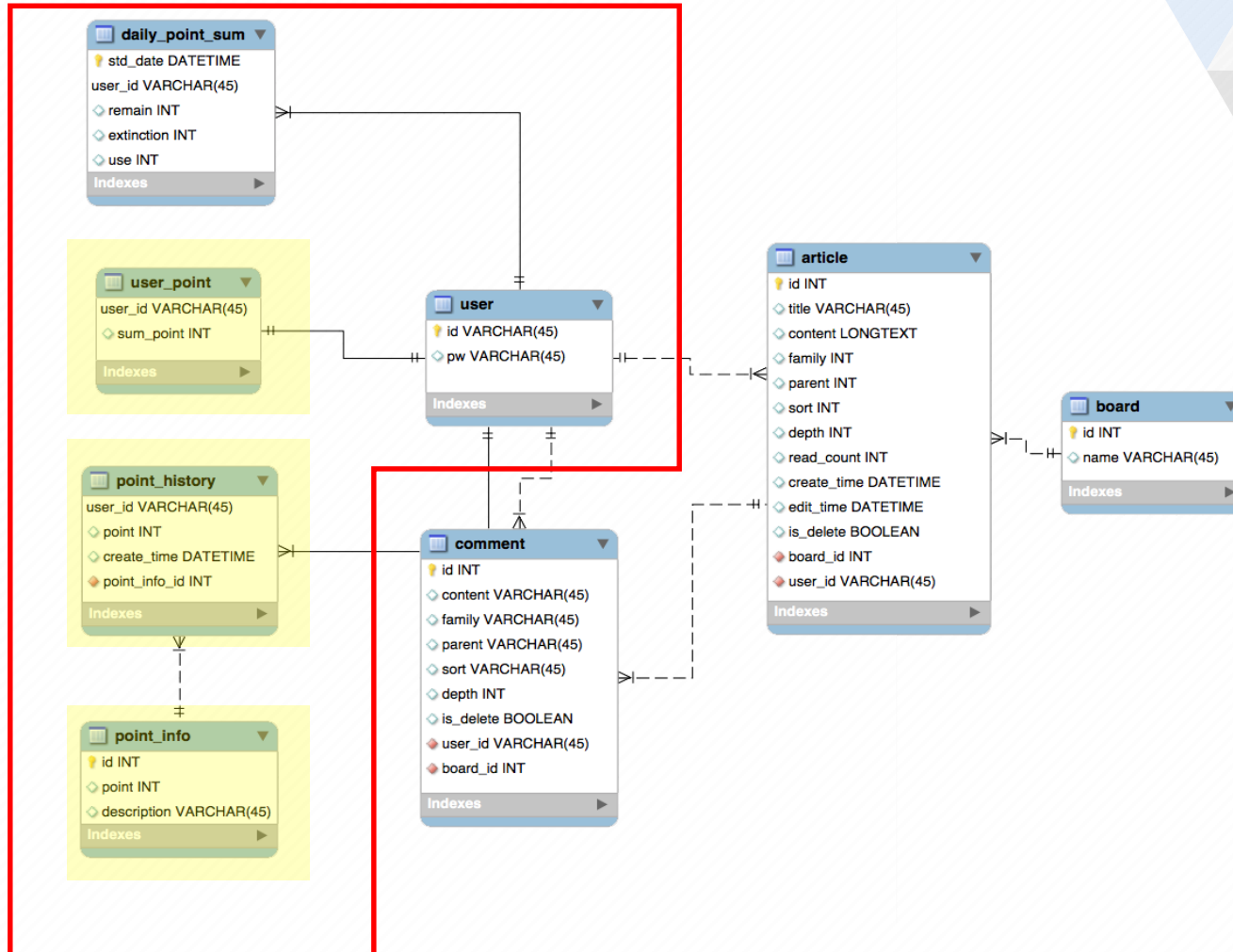


설계

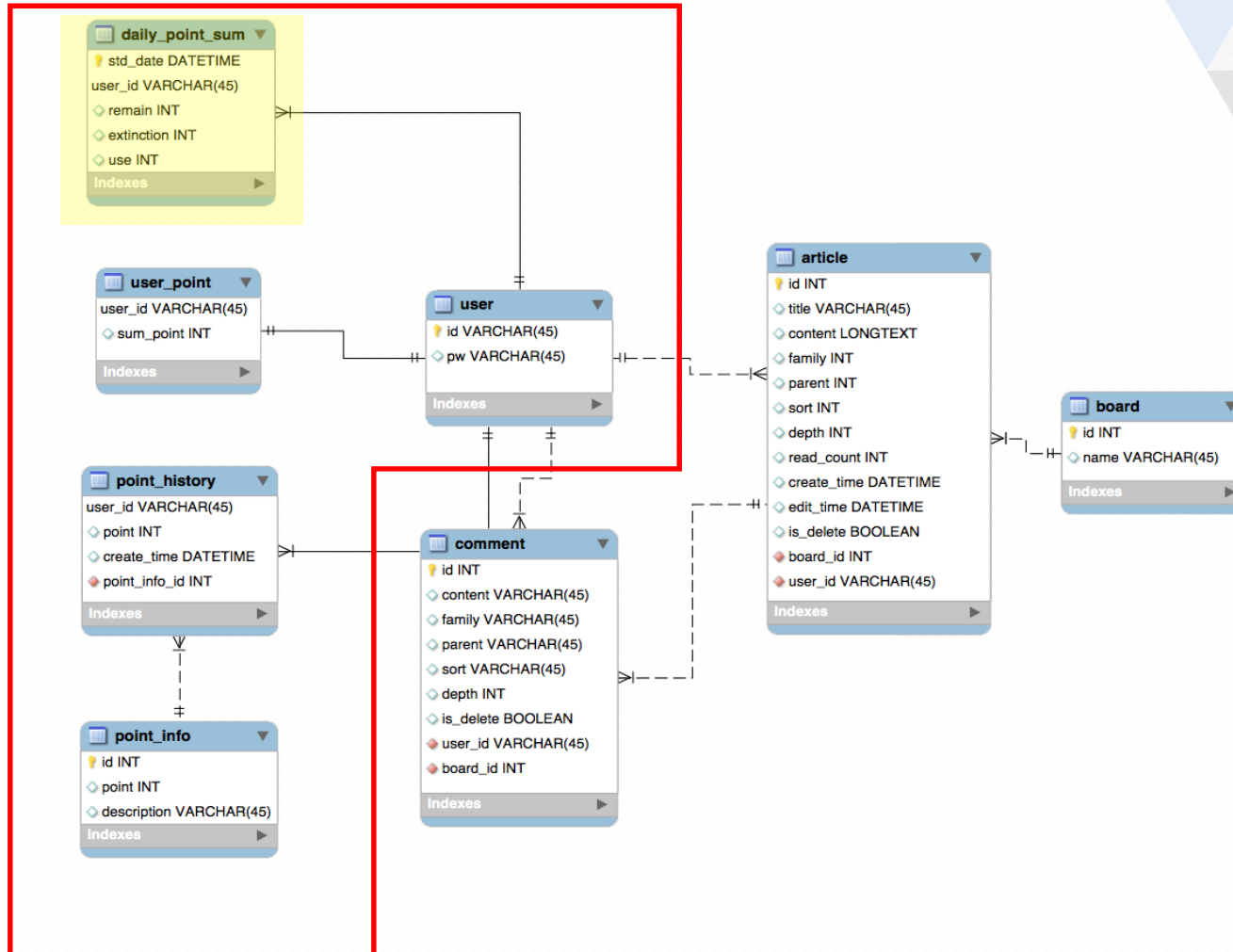
▶ E-R Diagram



► E-R Diagram

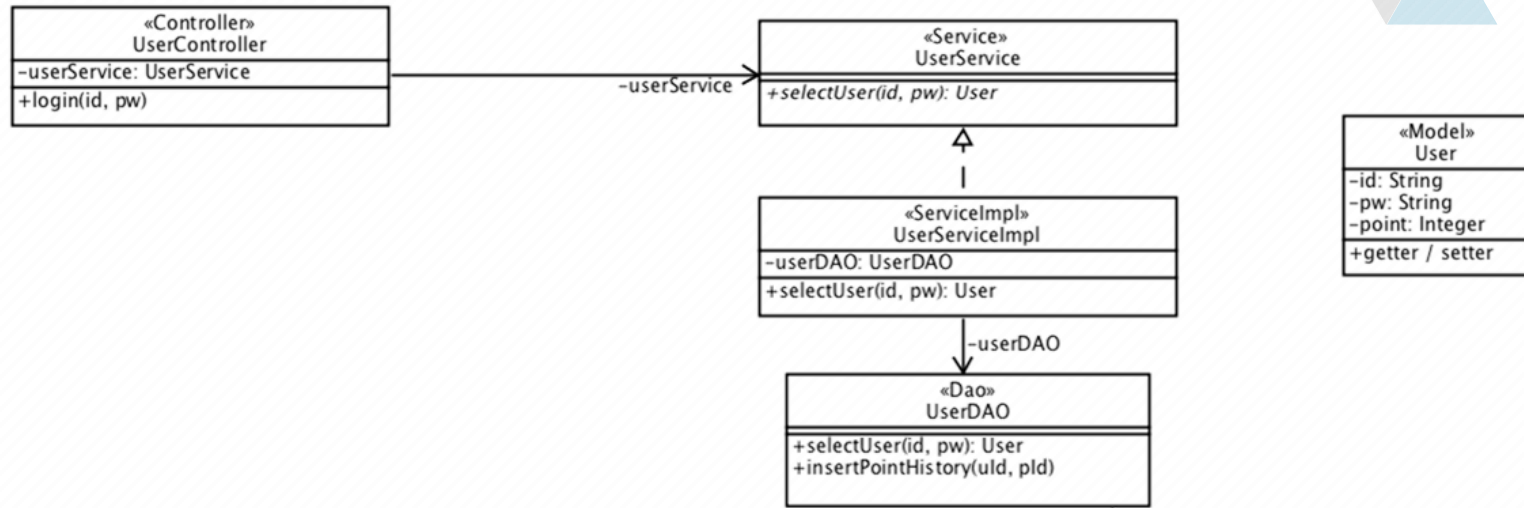


► E-R Diagram



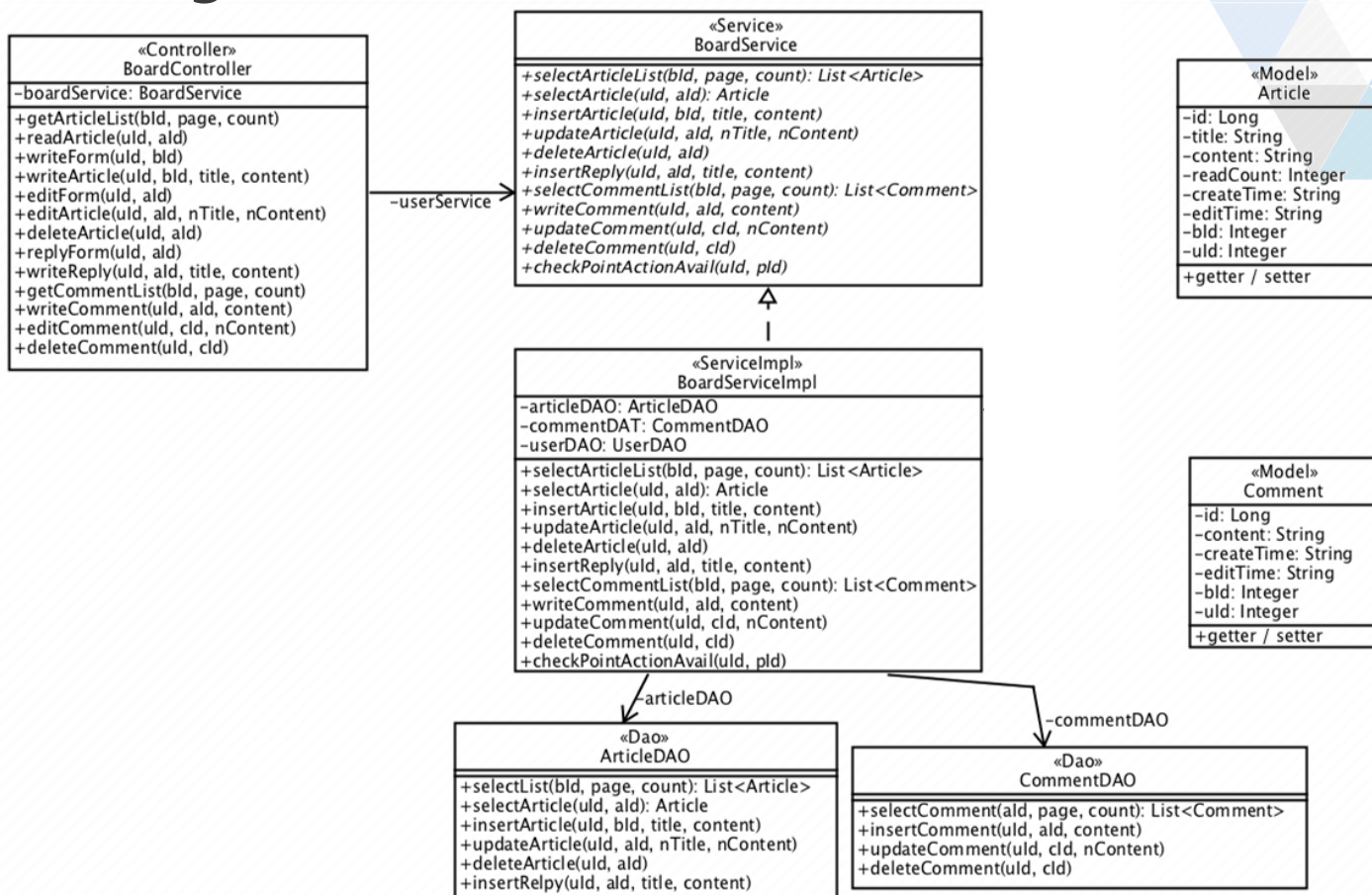
설계

▶ Class Diagram



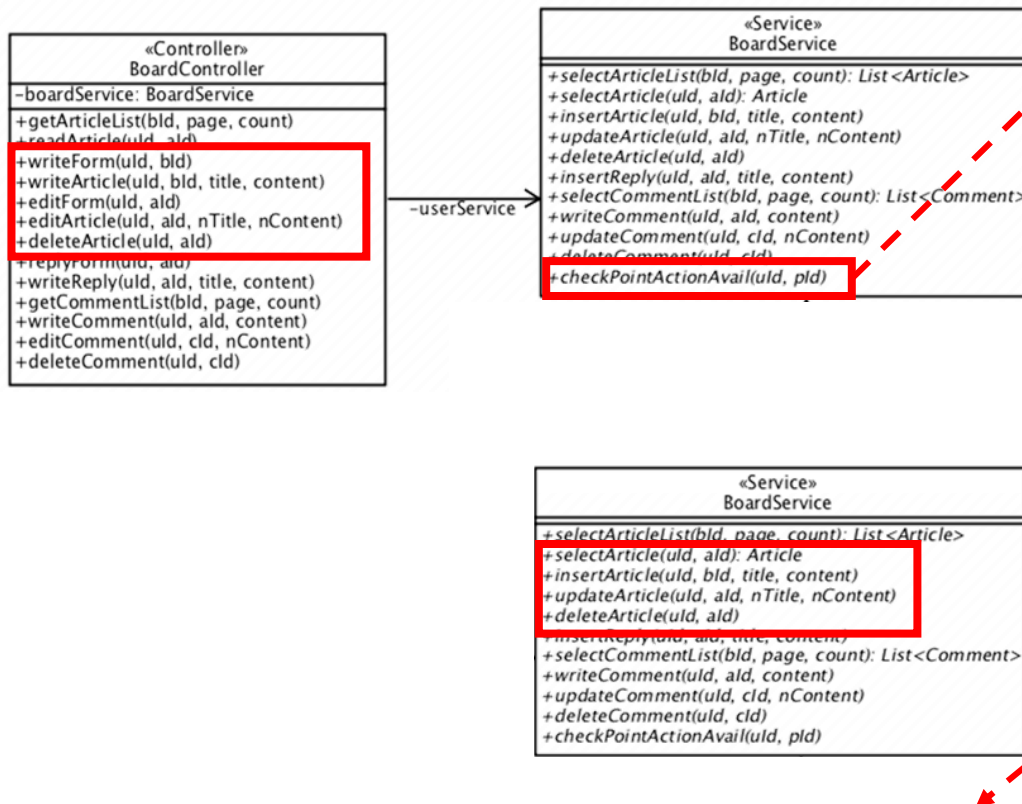
설계

▶ Class Diagram



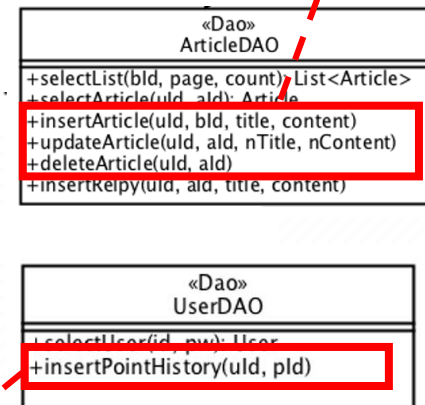
설계

▶ Class Diagram



미리 정의된 (1: 읽기/+10, 2:삭제/-10 ...)
포인트 아이디를 사용하여
유저가 가능한 액션인지 체크

게시글 저장/삭제/수정 실행
- Article 테이블 insert/update

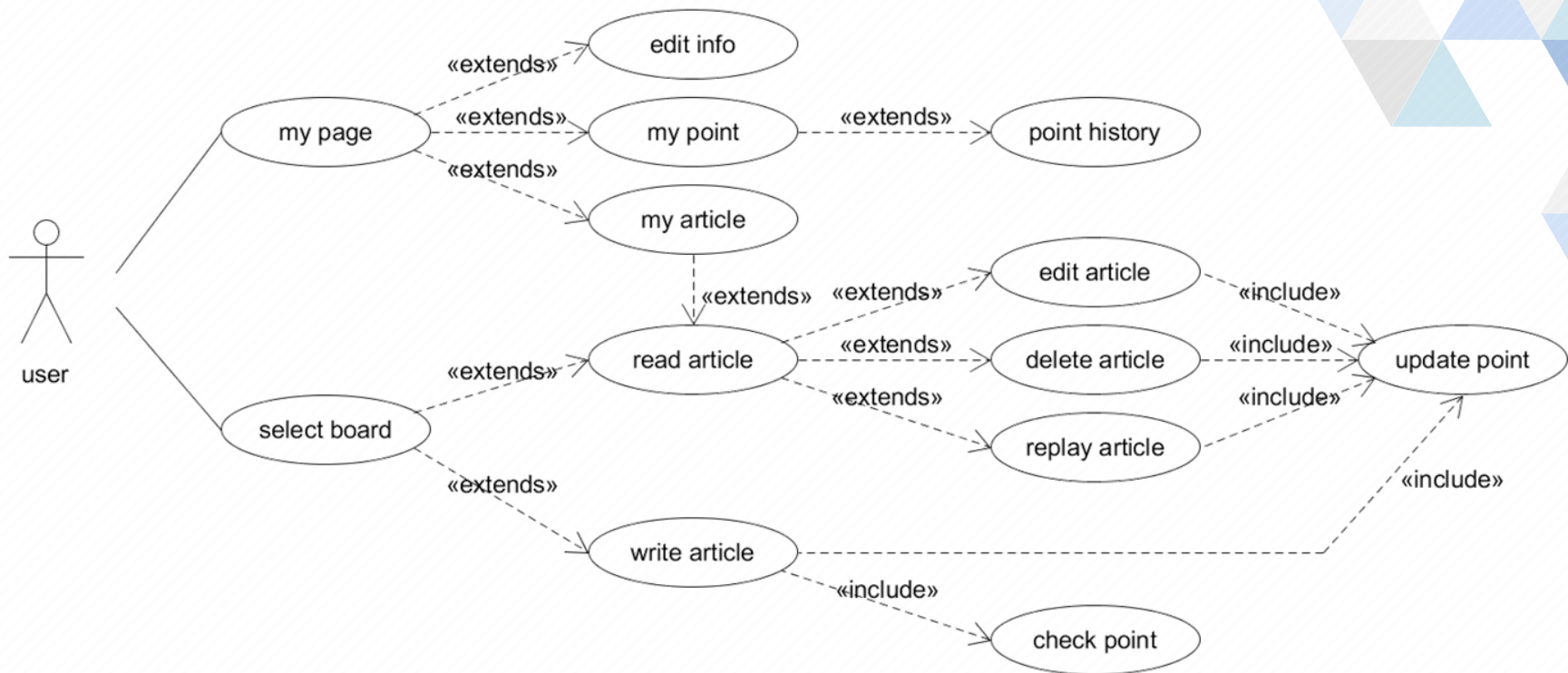


Point 갱신

- point_History 테이블에 정보 insert
- Trigger를 통해 user_point.point_sum(현재 유저포인트) Update

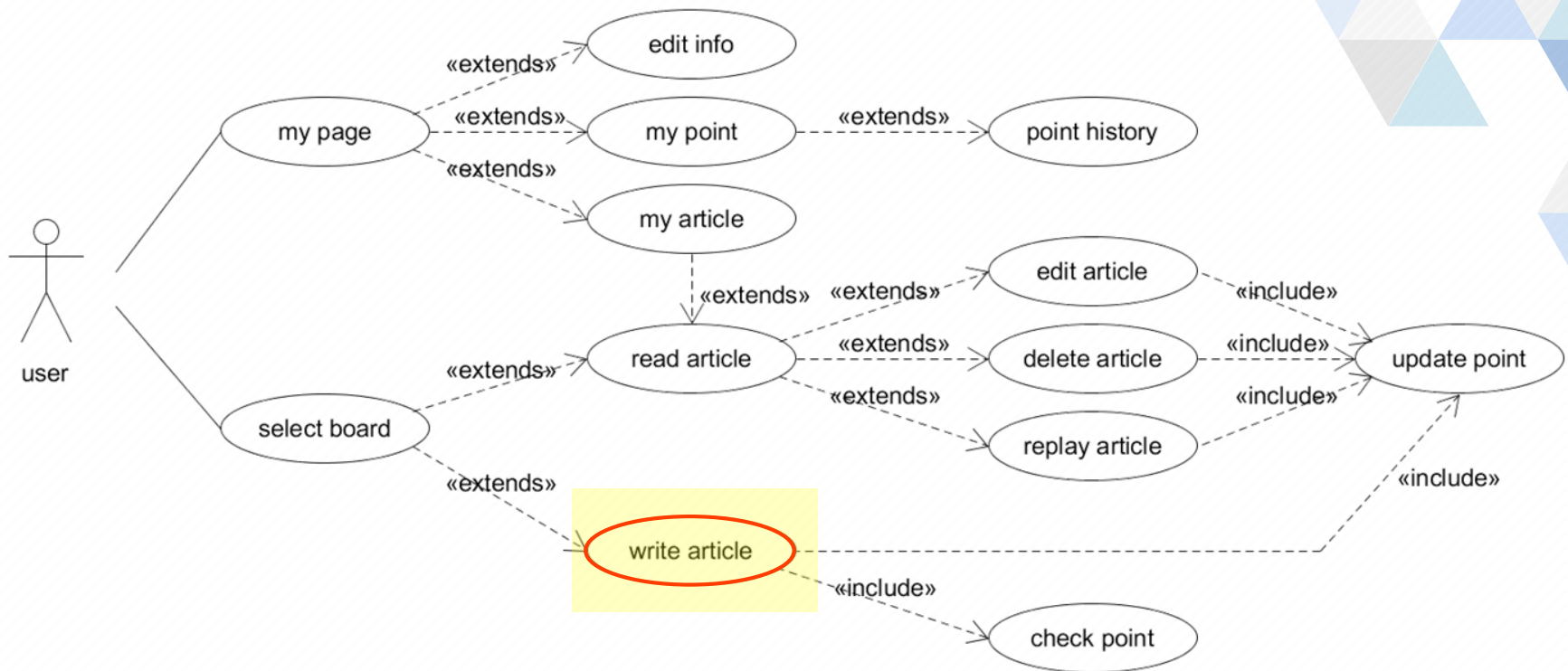
설계

► Usecase



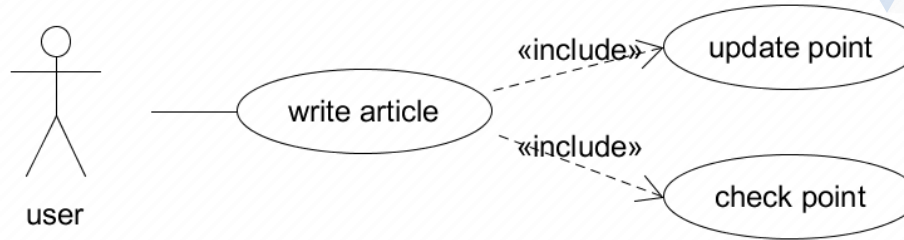
설계

► Usecase



설계

► Usecase



Use case name	Write Article : 게시글 작성하기
Participating actors	Initiated with Customers(고객)
Flow of events	<ol style="list-style-type: none"> 1. 사용자는 "글작성하기" 기능을 실행한다. 2. Board System은 사용자 Point를 체크한다. 3. 사용자는 글을 작성하고 저장한다. 4. Board System은 게시글을 저장하고 Point를 Update한다.
Entry condition	<ul style="list-style-type: none"> ▶ 사용자는 100이상의 Point를 가지고 있다. ▶ This usecase includes the checkpoint usecase.
Exit condition	<ul style="list-style-type: none"> ▶ 사용자는 저장한 게시글을 확인한다.
Quality requirements	<ul style="list-style-type: none"> ▶ Board System은 게시글등록, Point Update를 하나의 트랜잭션으로 관리한다.



Review

▶ 중점구현항목 & 어려웠던 점

- 답변형 게시글 순서

id	family	sort	depth	title
2	2	0	0	2번 글
4	2	1	1	└ 2번 글-2
5	2	2	2	└ 2번 글-2-1
3	2	3	1	└ 2번 글-1
1	1	0	0	1번 글

1. UPDATE article SET sort = sort+ 1

WHERE family = (원 글의 family) AND sort > (원 글의 sort)

2. INSERT INTO article VALUES

(ID, (원 글의 family), (원 글의 sort+1), (원 글의 depth+1), '제목')

Review

▶ 중점구현항목 & 어려웠던 점

- 답변형 / 페이징을 빠르게 처리하기 위한 인덱스 처리

```
1 EXPLAIN
2 SELECT id, family, sort
3 FROM article
4 WHERE is_delete = 'N' ORDER BY family, sort
5 LIMIT 3, 3;
```

Query Favorites ▾ Query History ▾ Run											
id	select_type	table	partit...	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	article	NULL	ref	board_sort	board_sort	1	const	4	100.00	Using where; Using index

Review

▶ 중점구현항목 & 어려웠던 점

- Point 기능 확장 / 관찰 / 실험을 위한 Daily Point 계산

daily_point_sum ▼	
std_date DATETIME	
user_id VARCHAR(45)	
remain INT	누적적립(+)
extinction INT	소멸대상(과거의+)
use INT	사용누적(-)
Indexes ▶	

"특정고객의 일정기간에 소멸된 포인트는 얼마인가?"

"소멸기간을 증감시켰을 경우 부담되어질 전체
포인트의 크기는 얼마인가?"



Review

▶ 추가 개선 요소

- **게시판 기능 강화**

- 계정관리 / 파일, 미디어 첨부 / 익명기능 등

- **포인트 / 요구사항 특성 최적화**

- 지식 관리 / 등업형 커뮤니티(카페) 성격으로 발전
- Point 사용/획득 처 증대
- Point 흐름, 사용, 소멸 등을 관찰/테스트 할 수 있는 환경 구축
: 로그 모니터링 시스템 등

