

캡스톤디자인II 중간보고서

제 목	국문		노션 연동 블로그 서비스 구축	
	영문		Building of Notion-linked blog service	
진행 상황	중요 마일스톤	UI/UX 설계 노션 연동, 테스트 기반 개발 - 공통 요구사항 분석 API명세서 작성 도메인 분석 - 프론트엔드 Component 설계 UI/UX 분석 및 설계 상태관리 설계 SSR - 백엔드 DB테이블 설계 ERD 및 클래스 다이어그램 작성 비즈니스 로직 및 테스트 코드 작성		
	진행상황	완료 - 공통 요구사항 분석 API명세서 작성 도메인 분석 - 프론트엔드 Component 설계 UI/UX 분석 및 설계 상태관리 설계 SSR - 백엔드 DB테이블 설계 ERD 및 클래스 다이어그램 작성 진행중 - 프론트엔드 Component 설계 - 백엔드 비즈니스 로직 및 테스트 코드 작성		
산출물	요구사항 정의서(별첨 1), 중간보고서(별첨 2)			
팀 구성원	학년	학 번	이 름	연락처(전화번호/이메일)
	4	20197121	정지용	010-8621-2963 / 20197121@edu.hanbat.ac.kr
	4	20181605	홍현표	010-9934-3834 / 20181605@edu.hanbat.ac.kr
컴퓨터공학과와 프로젝트 관리규정에 따라 다음과 같이 요구사항 정의서와 중간보고서를 제출합니다				
2023 년 4 월 25 일				
책임자 : (인)				
지도교수 : (인)				

프로젝트명 : 노션 연동 블로그 서비스 구축
캡스톤 디자인II, 중간보고서

Version 1.0

개발 팀원 명(팀리더) : 정지용
홍현표

대표 연락처 : 010-8621-2963
e-mail : 20197121@edu.hanbat.ac.kr

프로젝트명 : 노션 연동 블로그 서비스 구축

소프트웨어 요구사항 정의서

Version 1.0

개발 팀원 명(팀리더):정지용

홍현표

대표 연락처:010-8621-2963

e-mail: 20197121@edu.hanbat.ac.kr

목차

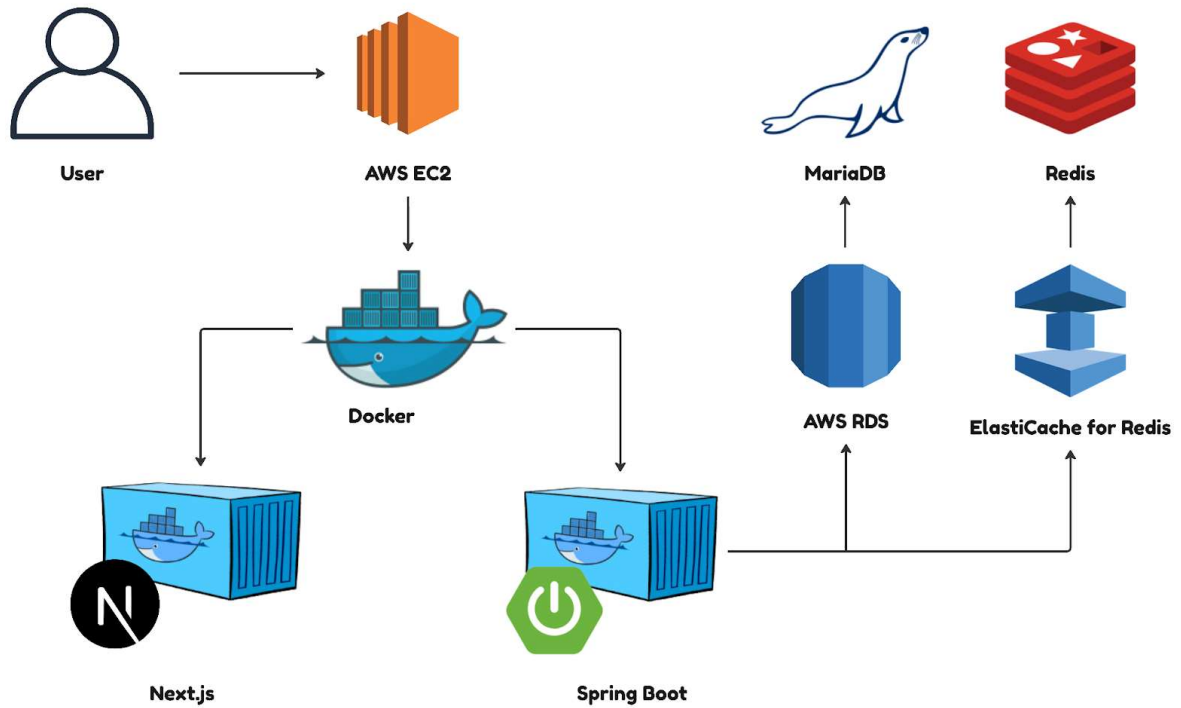
1. 개요
2. 시스템 장비 구성요구사항
3. 기능 요구사항
4. 성능 요구사항
5. 인터페이스 요구사항
6. 데이터 요구사항
7. 테스트 요구사항
8. 보안 요구사항
9. 품질 요구사항
10. 제약 사항
11. 프로젝트 관리 요구사항

1. 시스템 개요

1) 목표

개발자를 위한 노션 연동 블로그 서비스 구축

2) 시스템 구성도



2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001		
요구사항 명칭		IaaS		
요구사항 분류		시스템 장비 요구사항	응락수준	필수
요구사항 상세 설명	정의	백엔드 장비		
	세부 내용	-장비 품목 : Amazon EC2 -장비 수량 : 2 -장비 기능 : 프론트, 백엔드 서버 실행		

요구사항 고유번호		ECR-002		
요구사항 명칭		DB관리 소프트웨어		
요구사항 분류		시스템 장비 요구사항	응락수준	필수
요구사항 상세 설명	정의	DB 관리 소프트웨어(RDS)		
	세부 내용	-장비 품목 : Amazon RDS -장비 수량 : 1 -장비 기능 : DB저장 및 DB관리 - 장비 성능 및 특징: Amazon에서 제공하는 데이터베이스로 관리 프로그램으로 간편하게 DB를 설정 가능하며 원하는 경우 Custom으로 설정 또한 가능하다.		

요구사항 고유번호		ECR-003		
요구사항 명칭		DB관리 소프트웨어		
요구사항 분류		시스템 장비 요구사항	응락수준	필수
요구사항 상세 설명	정의	DB 관리 소프트웨어(Redis)		
	세부 내용	-장비 품목 : Amazon ElastiCache for Redis -장비 수량 : 1 -장비 기능 : DB저장 및 DB관리 - 장비 성능 및 특징: Amazon에서 제공하는 Redis 관리 프로그램으로 메모리 기반 DBMS이다.		

3. 기능 요구사항

요구사항 고유번호		SFR-001		
요구사항 명칭		회원 관련 기능		
요구사항 분류		기능 요구사항	응락 수준	필수
요구사항 상세설명	정의	- 회원 가입, 회원 수정, 회원 탈퇴 등 회원 관리와 관련된 기능		
	세부 내용	<ul style="list-style-type: none"> - 블로그를 서비스를 사용하기 위해 정상적인 회원가입 및 로그인 - 회원가입시 이메일 혹은 OAuth2를 이용한 회원가입 지원 - 변경사항이 있을 경우 정상적인 회원 정보 변경 - 서비스를 더 이상 이용 하고 싶지 않을 경우 정상적인 회원 탈퇴 		

요구사항 고유번호		SFR-002		
요구사항 명칭		포스트 관련 기능		
요구사항 분류		기능 요구사항	응락 수준	필수
요구사항 상세설명	정의	- 포스트 생성, 수정 등 포스트 기능 및 댓글, 시리즈 관련 기능		
	세부 내용	<ul style="list-style-type: none"> - 포스트는 Markdown을 이용하여 작성 - 작성 완료 한 포스트는 정상적으로 업로드 - 작성된 포스트 중 변경 사항이 있을 경우 수정 기능 지원 - 포스트에 해시태그를 이용하여 검색에 용이하도록 함 - 자유롭게 포스트를 삭제 - 포스트에 댓글 기능 지원 - 댓글은 대댓글까지 허용 - 댓글 수정 기능 - 댓글 삭제 기능 - 포스트를 사용자가 원하는 분류에 따라 시리즈로 묶어보기 기능 지원 - 다른 사용자들이 작성한 포스트에 댓글을 남김 - 다른 사용자들이 작성한 포스트에 좋아요(공감) 표시 가능 - 작성한 포스트들을 시리즈로 묶어서 관리 - 시리즈 생성 및 삭제, 수정 지원 		

요구사항 고유번호		SFR-003		
요구사항 명칭		노선과 연동		
요구사항 분류		기능 요구사항	응락 수준	필수
요구사항 상세설명	정의	- 노선에서 글을 불러오거나 작성한 글을 노선에 저장		
	세부 내용	<ul style="list-style-type: none"> - 노선에서 작성한 글을 블로그에 연동 - 블로그에서 작성한 글을 노선에 저장 - 서로 연동된 글은 한쪽이 바뀔시 양쪽 다 수정 		

4. 성능 요구사항

요구사항 고유번호		PER-001		
요구사항 명칭		일반 서비스 반응 시간		
요구사항 분류		성능	응락 수준	필수
요구사항 상세설명	정의	- General Service Render Time		
	세부 내용	<ul style="list-style-type: none"> - 너무 빠른 응답은 사용자 인지 불가 - 평균응답시간 최대 0.5~1초 - 부하 발생시 2초 미만 		

요구사항 고유번호		PER-002		
요구사항 명칭		보안 관련 서비스 반응 시간		
요구사항 분류		성능	응락 수준	필수
요구사항 상세설명	정의	- Security Service Render Time		
	세부 내용	<ul style="list-style-type: none"> - 보안과 관련된 결과물의 응답시간이 빠르게 출력될 경우 오히려 반감을 사며, 신뢰도를 떨어뜨림 - 너무 오랜 기다림은 더 큰 불편함 발생시킴 - 평균응답시간 최대 3초 미만 - 부하 발생시 4초 미만 		

5. 인터페이스 요구사항

요구사항 고유번호		SIR-001		
요구사항 명칭		서비스 사용 필수 서비스		
요구사항 분류		인터페이스	응락 수준	필수
요구사항 상세설명	정의	- 메인 페이지 헤더		
	세부 내용	<ul style="list-style-type: none"> - 홈페이지로 접근 할 수있는 로고 버튼 - 눈의 피로와 부담을 덜어주는 야간모드 - 손 쉽게 글찾기, 글쓰기 가능 - 회원정보 		
설계화면				

6. 데이터 요구사항

요구사항 고유번호		DAR-001		
요구사항 명칭		데이터 저장 방법		
요구사항 분류		데이터	응락 수준	필수
요구사항 상세설명		<ul style="list-style-type: none"> - 데이터는 MariaDB에 저장한다. - 저장 및 보존은 Amazon RDS에서 실시한다. - DB의 구조는 ERD를 기반으로 작성하며 수정시 ERD와 같이 수정한다. - 사용자 개인정보 중 민감사항은 저장하지 않으며 비밀번호처럼 꼭 필요한 경우해싱하여 저장한다. 		

요구사항 고유번호	DAR-002		
요구사항 명칭	인증 코드 저장 방법		
요구사항 분류	데이터	응락 수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - Email로 회원가입시 필요한 인증코드는 Redis에 저장한다. - 저장 및 보존은 AmazonElasticCacheForRedis에서 실시한다. - 데이터는 5분간 저장하며, 만료 후 삭제한다. 		

7. 테스트 요구사항

요구사항 고유번호		TER-001		
요구사항 명칭		테스트 방안		
요구사항 분류		테스트	응락 수준	필수
요구사항 상세설명		<ul style="list-style-type: none"> - 시스템은 제공되기로 한 요구사항을 모두 제공한다. - 직접 작성한 메서드에 대해 단위 테스트를 적용해 개발 단계에서 가능한 많은 오류 탐지 및 제거 - 기능 구현 정확성은 사용자가 직접 테스트 수행 기간에 테스트를 수행함으로써 평가함 		

8. 보안 요구사항

인증 - 이메일, OAuth2를 활용한 회원가입을 통한 사용자 인증

인가 - 역할 기반 접근 통제 적용

패스워드 암호화 - BCrypt 해시 알고리즘 적용

요구사항 고유번호		SER-001		
요구사항 명칭		응용 보안		
요구사항 분류		보안	응락수준	필수
요구사항 상세 설명		세부 내용	<ul style="list-style-type: none"> - 일반 사용자는 직접 DB에 접근할 수 없으며, 응용시스템과 DB 간 데이터 교환 시 계정 기반 인증을 거쳐야 한다. - 사용자 정보, 패스워드 등은 소스코드에 직접 하드코딩하지 않음 	

9. 품질 요구사항

요구사항 고유번호		QUR-001		
요구사항 명칭		결함 발생률		
요구사항 분류		품질	응락 수준	필수
요구사항 상세설명	정의	품질관리(기술 관점)		
	세부 내용	<ul style="list-style-type: none"> - 시험 운영 기간 동안 발견된 결함 수와 결함의 지속 시간을 측정하며, 결함 발생률이 5% 이상인 경우, 서비스 오픈 기간을 연장해야 함 - 시험 운영 기간 동안 결함 지속 시간의 최대 한계 값은 3시간 이하여야 함 		

10. 제약 사항

요구사항 고유번호		COR-001		
요구사항 명칭		시스템 개발 제약사항		
요구사항 분류		제약사항	응락수준	필수
요구사항 상세 설명	정의	시스템 개발		
	세부 내용	- 프레임워크: Spring Boot, Spring Security, Hibernate, Next.js - 언어: Java 11, JavaScript(es6+)		

요구사항 고유번호		COR-002		
요구사항 명칭		표준 제약사항		
요구사항 분류		제약사항	응락수준	필수
요구사항 상세 설명	정의	표준		
	세부 내용	- 국내 사용률 5% 이상 브라우저 지원		

11. 프로젝트 관리 요구사항

1) 프로젝트 수행 조직

형상관리는 git으로 하며, git-flow를 통해 개발 워크플로우를 구성한다.

그리고 github를 활용해 코드를 공유하고 협업한다.

Pull Request를 활용해 작업 내용을 파악하고 팀원 간 코드 리뷰를 진행하여 더 효율적인 코드를 작성할 수 있도록 하며 코드 병합시에 SquashMerge를 시행하여 커밋 로그의 가독성을 향상시킨다.

정지용: 백엔드 및 프론트엔드(팀원 1명 하차로 인한 업무 증가) / 개발 리더

홍현표: 백엔드

2) 프로젝트 일정계획

	9월	10월	11월	12월
소프트웨어 개발				
테스트				
배포 및 운영				
결과 보고서 작성				

소프트웨어 개발

- 프론트엔드

1. 댓글 및 시리즈, 해시태그 기능 개발

- 백엔드

1. OAuth2 로그인 기능 개발

2. Notion 연동 기능 수정

3. 기본기능 개발 - 해쉬태그 기능 개발

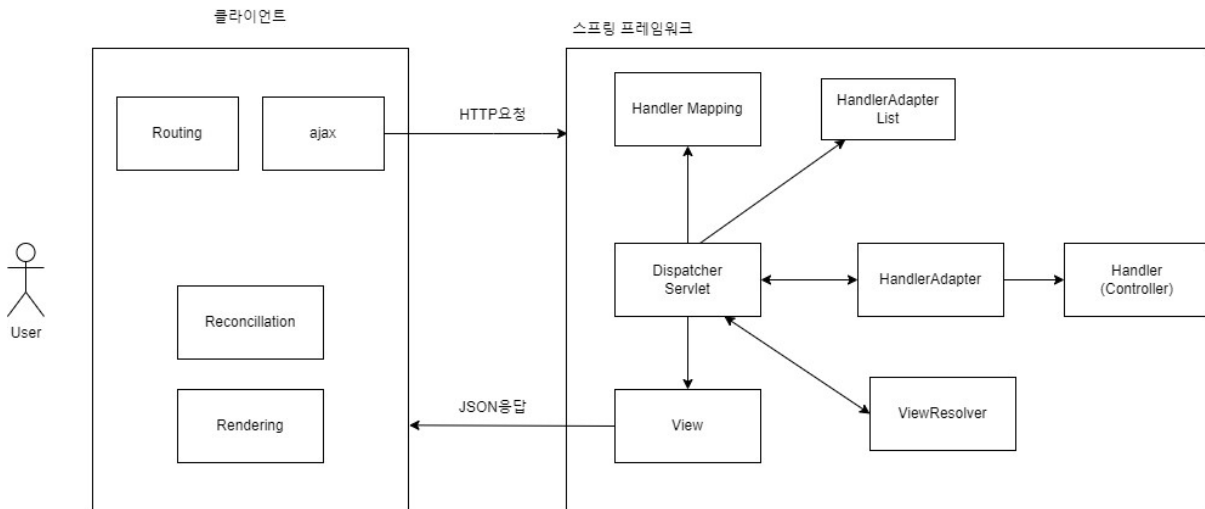
테스트: 단위 테스트 및 E2E 테스트

배포 및 운영: AWS EC2 기반 K8S 배포

중간보고서

1. 요구사항 정의서에 명시된 기능에 대하여 현재까지 분석, 설계, 구현(소스코드 작성) 및 테스트한 내용을 기술하시오.

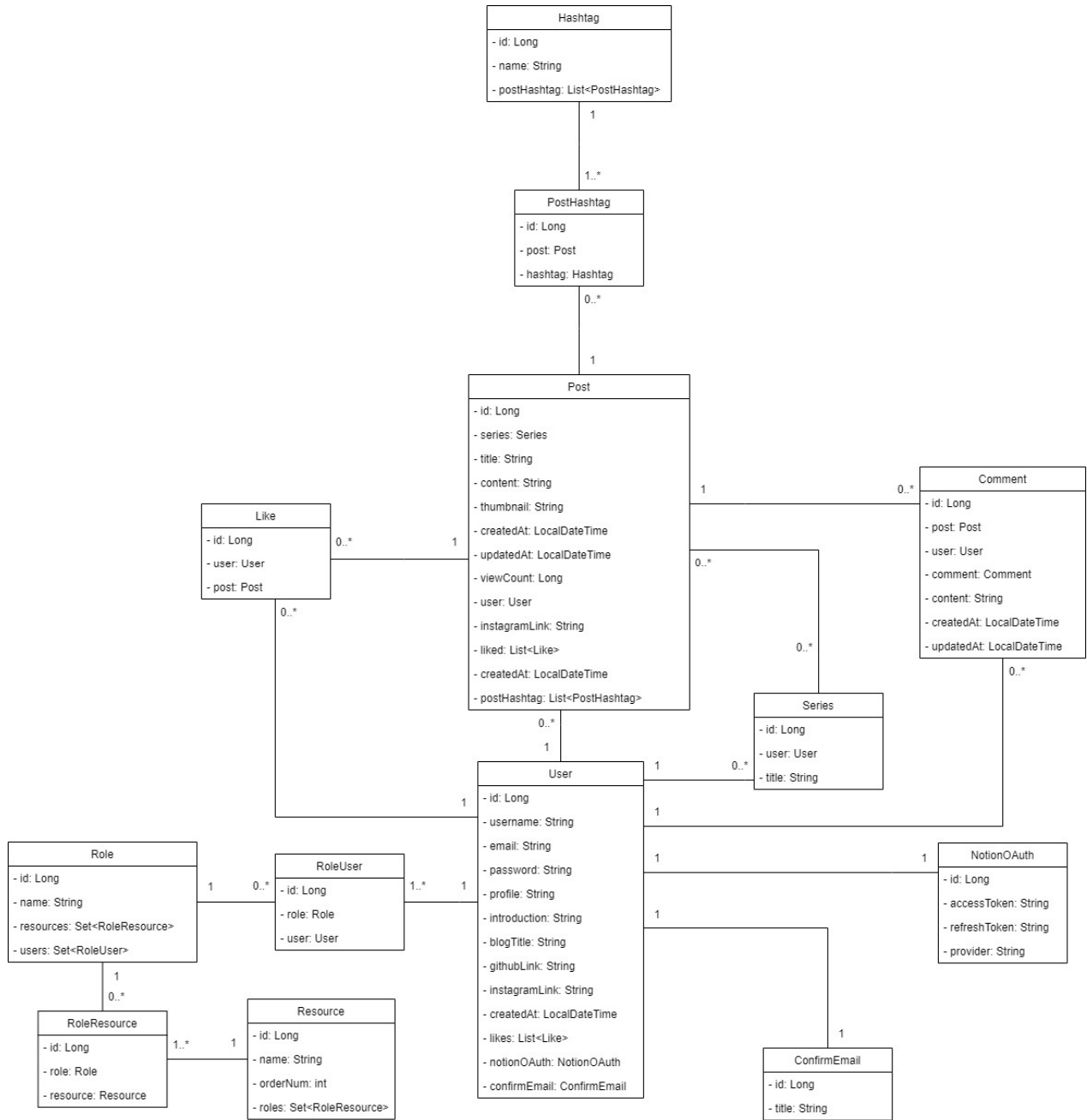
- 소프트웨어 아키텍처



- 기능별 상세 요구사항(또는 유스케이스)

Level1	Level2	Level3	Level4
사용자 관리	사용자 가입	이메일로 가입	X
		OAuth2로 가입	Google OAuth2 Github OAuth2
	사용자 정보 조회	유저 정보 조회	X
		유저 정보 수정	X
		유저 탈퇴	X
게시물 관리	게시글 관리	게시글 조회	전체 조회 특정 게시물 조회
		게시글 작성	해시태그 첨부
		게시글 수정	X
		게시글 삭제	X
	댓글 관리	댓글 조회	X
		댓글 작성	댓글에 대한 대댓글 가능
		댓글 수정	X
		댓글 삭제	X
	게시글 분류	시리즈 기능을 통한 게시글 통합	시리즈 생성
			시리즈 추가
			시리즈 수정
			시리즈 삭제

- 설계 모델(클래스 다이어그램, 클래스 및 모듈 명세서)



The diagram illustrates a database schema for a Notion-like application. It consists of several tables and their relationships:

- 테이블 (Tables):**
 - hashtagPK**: Fields: hashtagName, name, createdAt, NOT NULL.
 - posts**: Fields: postPK, title, content, createdAt, updatedAt, viewCount, NOT NULL.
 - comments**: Fields: commentPK, comment, userPK, createdAt, updatedAt, NOT NULL.
 - likes**: Fields: likePK, like, userPK, createdAt, NOT NULL.
 - resources**: Fields: resourcePK, resourceName, url, createdAt, NOT NULL.
 - confirm_email**: Fields: email, createdAt, NOT NULL.
- 포스트, 해시태그 (Post, Hashtag):**
 - post_hashtagPK**: Fields: postPK, hashtagPK, NOT NULL.
 - hashtag**: Fields: hashtagPK, name, NOT NULL.
- 포스트 (Post):**
 - postPK**: Fields: title, content, createdAt, updatedAt, viewCount, NOT NULL.
 - series**: Fields: seriesPK, seriesName, createdAt, NOT NULL.
 - tags**: Fields: tagPK, tag, userPK, createdAt, NOT NULL.
- 댓글 (Comments):**
 - commentPK**: Fields: comment, userPK, createdAt, updatedAt, NOT NULL.
 - commentPK**: Fields: comment, userPK, createdAt, updatedAt, NOT NULL.
- 좋아요 (Likes):**
 - likePK**: Fields: like, userPK, createdAt, NOT NULL.
- 게시판 (Board):**
 - boardPK**: Fields: boardName, boardType, createdAt, NOT NULL.
 - boardPK**: Fields: boardName, boardType, createdAt, NOT NULL.
- 유저 권한 (User Permissions):**
 - userPK**: Fields: userPK, role, NOT NULL.
 - role**: Fields: rolePK, roleName, NOT NULL.
- 자원 (Resources):**
 - resourcePK**: Fields: resourceName, url, createdAt, NOT NULL.
 - resourcePK**: Fields: resourceName, url, createdAt, NOT NULL.
- 확인 (Check):**
 - confirm_email**: Fields: email, createdAt, NOT NULL.
 - confirm_email**: Fields: email, createdAt, NOT NULL.

Relationships are indicated by arrows between the tables, showing foreign key constraints and one-to-many relationships.

- 문제해결을 위해 적용한 방법(또는 기법) 결과, (문제점, 해결방안)

- 팀원의 책임 및 역할 수행에 대한 결과, (문제점, 해결방안)

- 프로젝트 일정계획에 맞추지 못한 경우의 문제점, 해결 방안

- 16 -