

TCG RPG HEX

커뮤니티 사이트

Document Version	Ver. 20221010.0
Last Update	2022-10-10
Organization	휴먼교육센터 4조
Author	조인국, 정경섭
GitHub	https://github.com/humanbest/hex
Prototype	http://game.rubisco.kro.kr
Requirements Specification	https://docs.google.com/spreadsheets/d/1OPSx4iUnJ-PtS7t8Rowr5nPP1VSDcEoyxUdw2kQup98/edit?usp=sharing

Document History

Ver.	Date	Reason for modification	Detail of modification	Page	Author
ver. 20220823.0	2022-08-23	웹부분을 따로 문서 작성		All	조인국
ver. 20220823.0	2022-08-24	UI시나리오 파트 수정	UI시나리오 가시성 보완	8,9	양호열
ver. 20221006.0	2022-10-06	레이아웃 시나리오 작성	레이아웃 시나리오 작성	9-10	조인국
ver. 20221008.0	2022-10-08	회원 시나리오 작성	회원 시나리오 작성	11-14	조인국
ver. 20221008.0	2022-10-08	커뮤니티 시나리오 작성	커뮤니티 시나리오 작성	15-18	조인국
ver. 20221010.0	2022-10-10	개발문서 작성	개발문서 작성	23-38	조인국

Outline

구분	내용
기획의도	TCG RPG HEX 게임에 접속하기 위한 엔트리와 유저간 커뮤니케이션을 위한 커뮤니티 공간을 제공하기위한 웹어플리케이션 사이트 제작
요구사항	<ul style="list-style-type: none">• 회원가입 기능• 로그인 기능• 커뮤니티(보드, 댓글) 기능
개발환경	<ul style="list-style-type: none">• backend : 우분투 20.04 LTS / JAVA JDK 17 / Spring framework 2.7.2• frontend : Thymeleaf / Tailwind CSS / Javascript• DB : MariaDB

역할분배

이름	역할
조인국	<p>프로젝트 전반적인 흐름을 관리</p> <ul style="list-style-type: none">- 서버 환경 구축 및 도메인 연결- 아키텍처 설계- DB 구축 및 Entity 작성 : Member, Board, Category, Group, Comments- 레이아웃 디자인- 커뮤니티 페이지 디자인- 회원 페이지 디자인- 커뮤니티 CRUD 기능- 게시글 및 댓글 페이지징 기능- 댓글 REST 기능- 로그인 유지 기능
정경섭	<p>시큐리티 구축</p> <ul style="list-style-type: none">- 관리자 레이아웃 디자인- 관리자 페이지 디자인- 로그인 기능 : 인증 및 인가 시스템 구현- Group 기능 : 회원의 접근 등급을 지정

목차

AI 시나리오	7
유스케이스 다이어그램	8
사이트맵	9
ER 다이어그램	10
UI 시나리오	12
레이아웃	13
로그인 페이지	15
회원 페이지	16

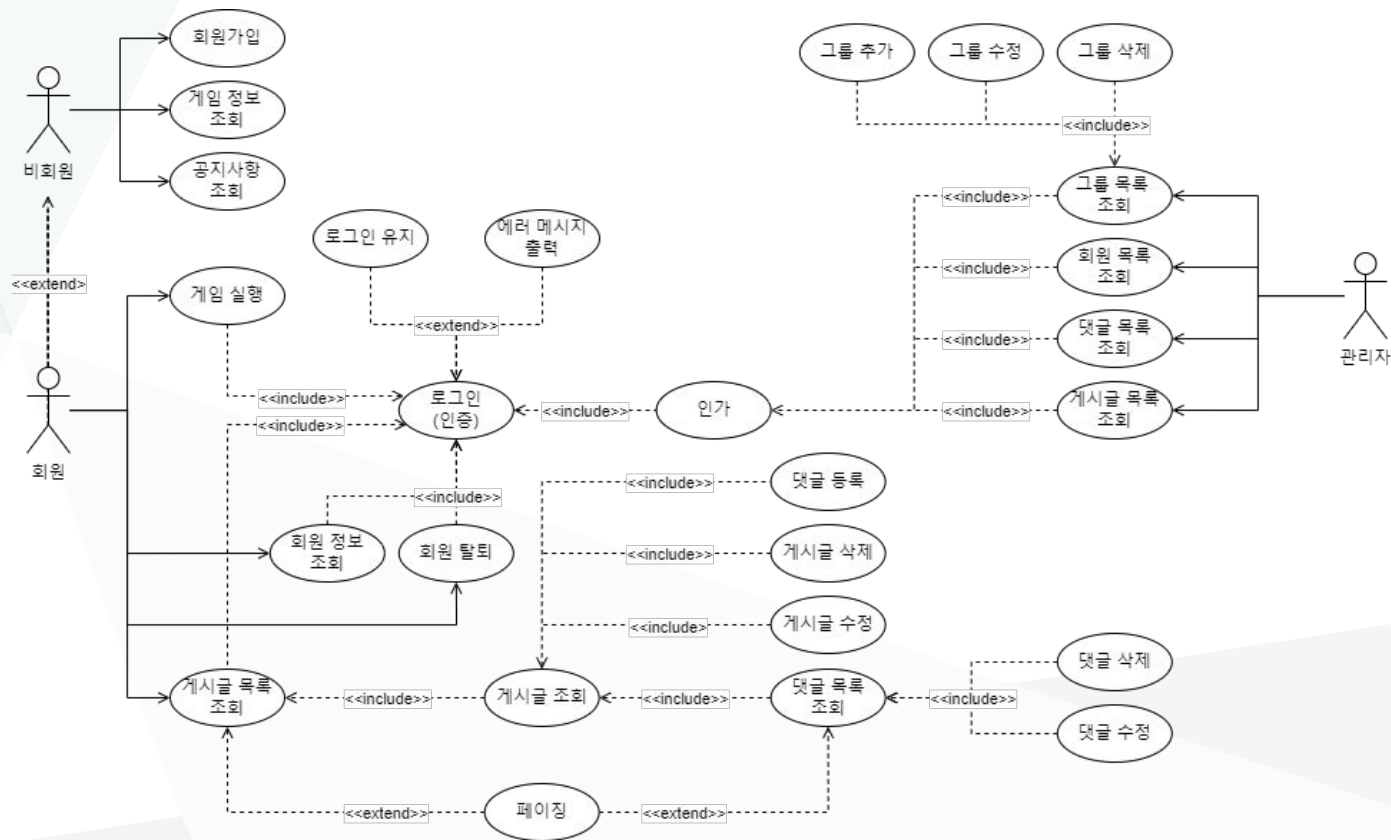


커뮤니티 게시판	19
Admin 게시판	23
개발문서	26
아키텍처	27
Entity	32
Repository.....	34
Service	35
Controller	37
인증 / 인가 시스템	40
개선사항	42

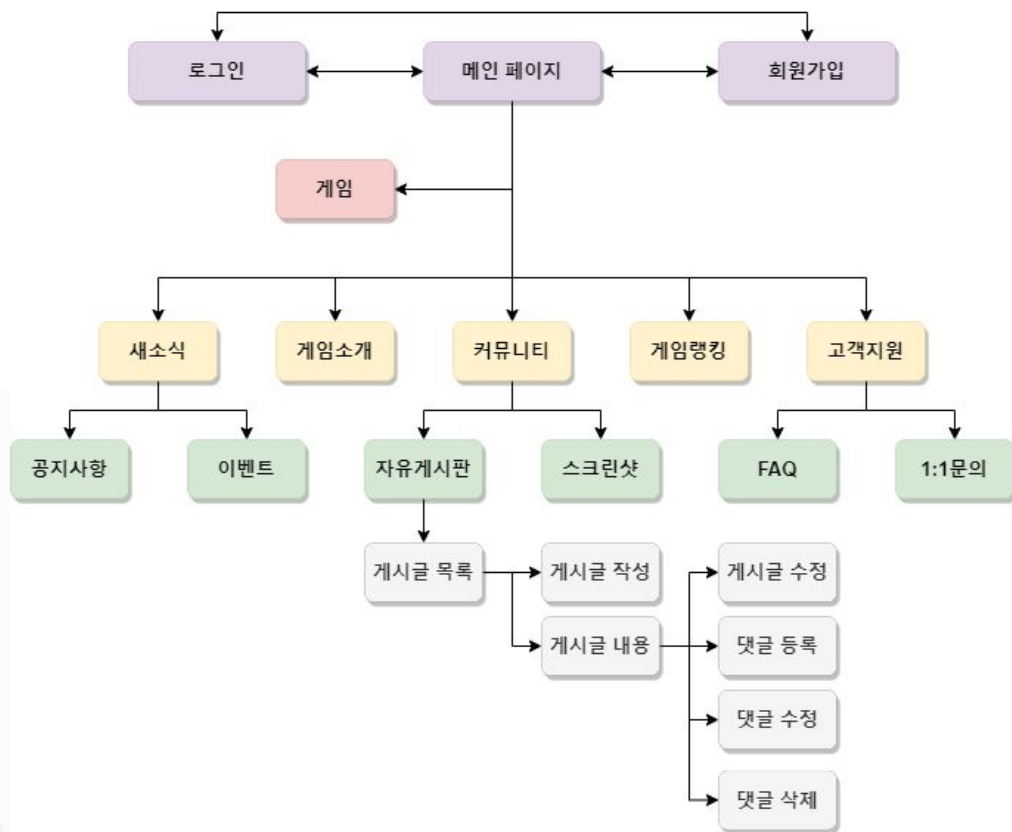
The background features several overlapping geometric shapes. A large teal triangle is at the top left. A grey triangle is at the bottom left. A teal triangle is on the left side, pointing right. A grey triangle is at the bottom left, pointing up. A teal triangle is at the bottom left, pointing up. A grey triangle is at the bottom left, pointing up.

AI 시나리오

AI 시나리오 - 유스케이스 다이어그램

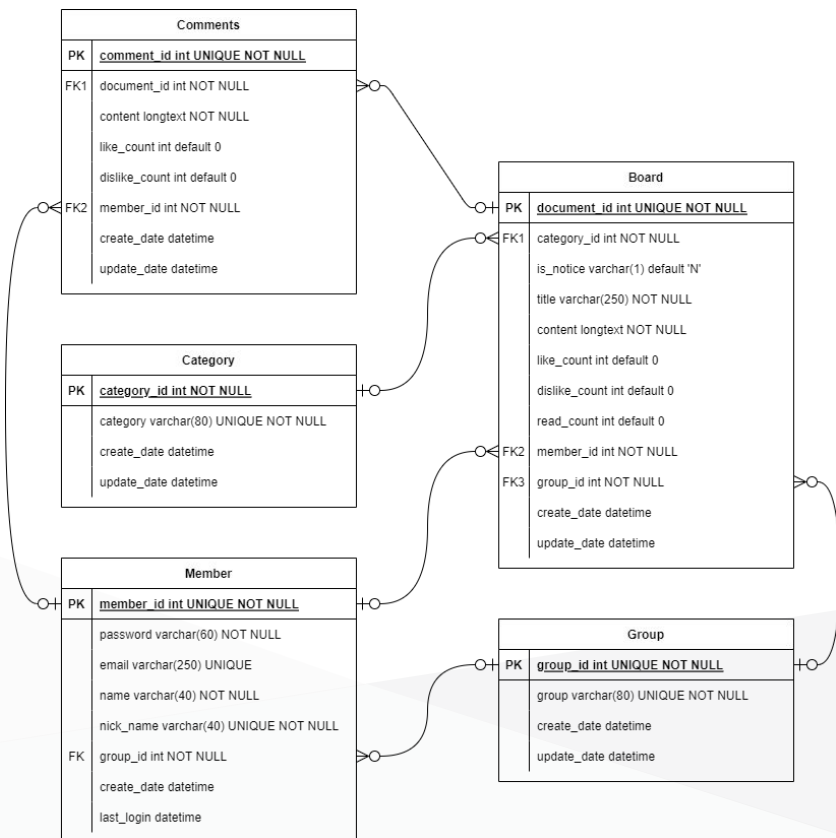


AI 시나리오 - 사이트맵



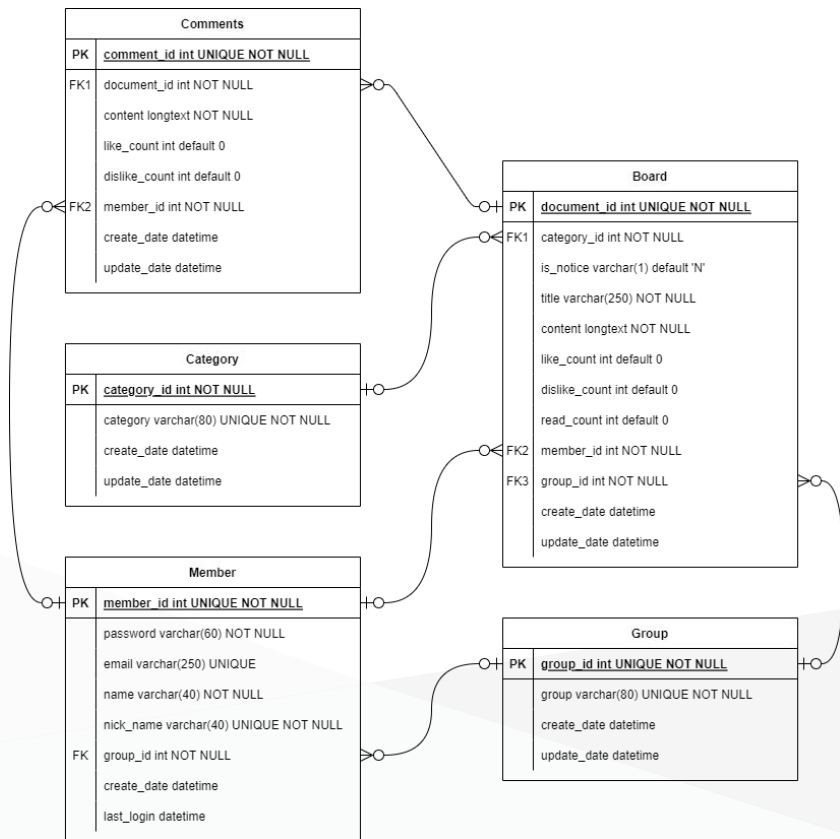
AI 시나리오 - ER 다이어그램을 통한 요구사항 분석

구분	내용
Board	커뮤니티 게시글을 저장하기 위한 테이블입니다. 카테고리, 회원, 그룹과 N:1의 연관관계를 가지며, 댓글과는 1:N 연관관계를 가집니다.
Category	게시글의 카테고리를 정의하여 저장하기 위한 테이블입니다. 게시글과 1:N 연관관계를 가집니다.
Comments	게시글에 대한 댓글을 저장하기 위한 테이블입니다. 회원, 게시글과 N:1 연관관계를 가집니다.



AI 시나리오 - ER 다이어그램을 통한 요구사항 분석

구분	내용
Member	<p>회원 정보를 저장하기 위한 테이블입니다.</p> <p>게시글, 댓글과는 1:N 연관관계를 가집니다.</p>
Group	<p>회원 등급을 정의하여 저장하기 위한 테이블입니다.</p> <p>회원, 게시글과 1:N 연관관계를 가집니다.</p>

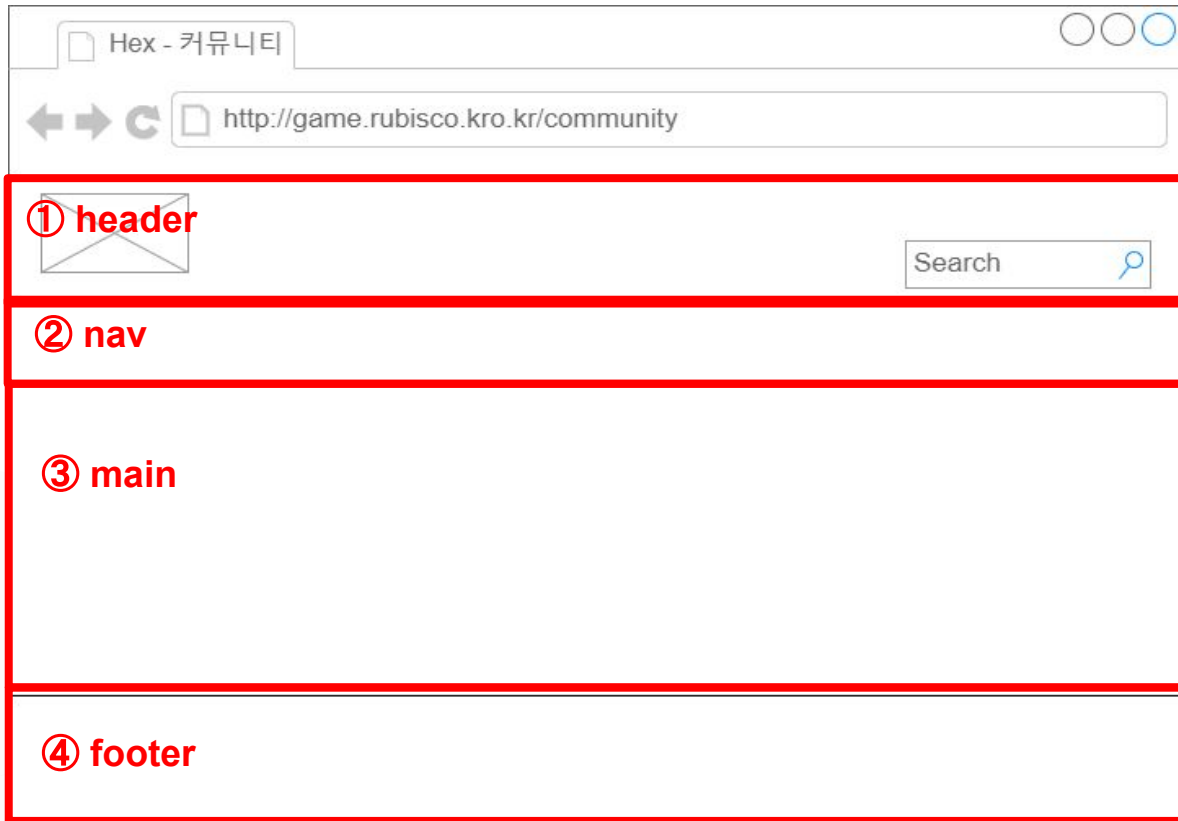


The background features several overlapping geometric shapes. A large teal triangle points downwards from the top left. A horizontal teal bar is positioned to the left of the text. Below this, a grey triangle points upwards from the bottom left. A thin, light teal line runs diagonally across the page. The text 'UI 시나리오' is centered in a dark blue font.

UI 시나리오

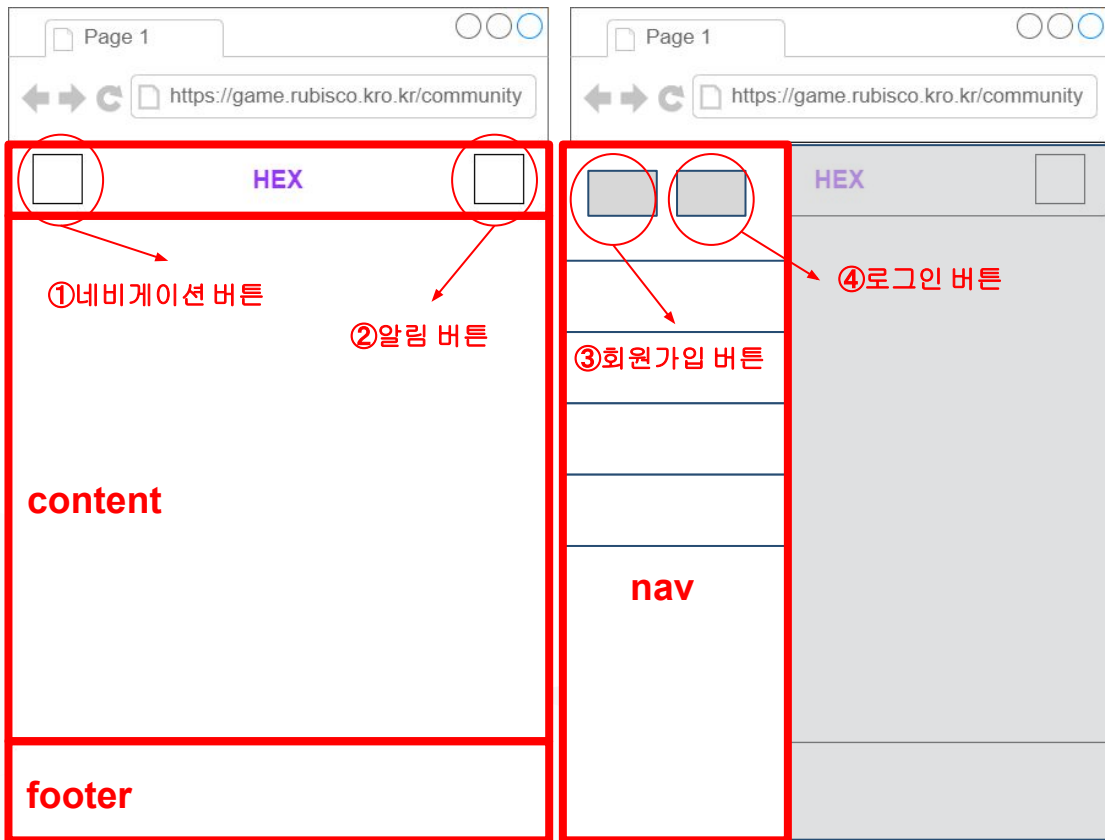
레이아웃 - 타임리프의 레이아웃 모듈 및 fragment 기능을 통해 개발

구분	내용
① header	로고, 네비게이션, 서치박스 등을 포함한 고정된 메인 헤더 영역입니다.
② nav	각 서비스 페이지에 접속가능한 고정된 메인 네비게이션 영역입니다.
③ main	관점별로 해당 영역을 작성하며, 서비스의 UI를 나타냅니다.
④ footer	사이트맵, 연락처, 운영방침, 저작권 등의 링크를 포함한 고정된 메인 푸터 입니다.



레이아웃 - 반응형 레이아웃으로 작성 (모바일 환경의 경우)

구분	내용
① 네비게이션 버튼	해당 버튼을 누르면 숨겨진 네비게이션 창이 열립니다.
② 알림 버튼	해당 유저가 받은 메시지를 확인할 수 있습니다.
③ 회원가입 버튼	회원가입 페이지로 이동합니다.
④ 로그인 버튼	로그인 페이지로 이동합니다.



로그인 - 로그인 페이지

구분	내용
접속 경로 : /auth?act=login	
① 로그인폼	로그인 요청을 위한 폼을 작성합니다.
② 아이디 / 비밀번호 인풋박스	아이디와 비밀번호를 입력하는 인풋박스 입니다.
③ 로그인 유지 체크박스	로그인을 일정기간 유지할 수 있도록 하는 기능의 사용여부를 체크합니다.
④ 암호 찾기 버튼	암호찾기 페이지로 이동합니다.
⑤ 로그인 버튼	로그인폼을 서버에 전송하여 로그인 요청을 합니다.
⑥ 회원가입 버튼	회원가입 페이지로 이동합니다.

The screenshot shows a web browser window titled 'Hex - 로그인' with the URL 'https://game.rubisco.kro.kr/auth?act=login'. The page layout includes a header with the text 'header HEX', a navigation bar with links 'nav', '새소식', '게임소개', '커뮤니티', '게임랭킹', and '고객지원'. The main content area is titled '로그인' and contains a login form. The form includes input fields for 'ID' and 'password', a checkbox for '로그인 상태 유지' (Login status maintain), a link for '암호를 잊으셨나요?' (Forgot your password?), a '로그인' (Login) button, and a '회원가입' (Sign up) button. Red numbered circles 1 through 6 are overlaid on the page to indicate specific elements: 1 points to the main login form area, 2 points to the ID input field, 3 points to the password input field, 4 points to the '로그인 상태 유지' checkbox, 5 points to the '로그인' button, and 6 points to the '회원가입' button.

회원 - 회원가입 페이지

구분	내용
접속 경로 : /member?act=signUp	
① 회원가입폼	회원가입을 위한 폼을 작성합니다.
② 사이트 이용정보 섹션	사이트에서 인증 및 인가를 위한 아이디와 비밀번호를 입력하는 텍스트 박스입니다.
③ 개인정보 섹션	이름, 닉네임, 이메일 등 개인정보를 입력하는 텍스트 박스 입니다.
④ 회원가입 버튼	회원가입 요청을 합니다.

The screenshot shows a web browser window with the URL `https://game.rubisco.kro.kr/member?act=signUp`. The page has a header with the text "header HEX" and a navigation bar with links: "nav", "새소식", "게임소개", "커뮤니티", "게임랭킹", and "고객지원". The main content area is titled "회원가입" (Sign Up) and contains two sections: "사이트 이용정보 입력" (Site Usage Information Input) and "개인정보 입력" (Personal Information Input). The "사이트 이용정보 입력" section includes an "아이디" (ID) field and a "비밀번호" (Password) field with a "비밀번호 확인" (Confirm Password) field. The "개인정보 입력" section includes "이름" (Name), "닉네임" (Nickname), and "이메일" (Email) fields. A "회원가입" (Sign Up) button is located at the bottom of the form. Red annotations are present: a red box around the entire sign-up form, a red circle around the "회원가입" button, and red numbers 1 through 4 indicating the flow of the form.

회원 - 회원정보 페이지

구분	내용
접속 경로 : /member	
① 회원정보 섹션	회원정보를 확인할 수 있습니다.
② 회원정보 폼	회원정보를 수정할 수 있습니다.
③ 수정 버튼	회원정보의 수정을 요청합니다.
④ 비밀번호 변경 버튼	비밀번호 변경 페이지로 이동합니다.
⑤ 회원탈퇴 버튼	회원탈퇴를 할 수 있습니다.

Hex - 회원정보

https://game.rubisco.kro.kr/member

header HEX

nav 새소식 게임소개 커뮤니티 게임랭킹 고객센터

회원정보

①

②

ID

이름

닉네임

이메일

③ 수정 ④ 비밀번호 변경 ⑤ 탈퇴

회원 - 비밀번호 변경 페이지

구분	내용
접속 경로 : /member?act=updatePassword	
① 비밀번호 변경 폼	비밀번호 변경 데이터를 입력하는 폼입니다.
② 에러 메시지 영역	서버로부터 에러메시지를 받으면 나타납니다.
③ 비밀번호 변경 인풋 박스	비밀번호를 변경 데이터가 입력되는 인풋박스 입니다.
④ 수정 버튼	비밀번호 수정을 요청합니다.

①

비밀번호 변경

②

새 비밀번호가 일치하지 않습니다.

아이디

test

③

현재 비밀번호

새 비밀번호

새 비밀번호 확인

④

수정

커뮤니티 게시판 - 게시물 리스트 페이지

구분	내용
접속 경로 : /community 파라미터 : page - 게시물 목록의 페이지 size - 한 페이지에 출력되는 게시물 목록 수	
① 게시글 목록 테이블	<p>게시글의 목록을 테이블 형태로 나타냅니다. 기본값으로 10개의 목록을 나타냅니다.</p> <p>게시글 제목에 hover하면 툴팁으로 요약된 내용이 보여집니다.</p>
② 글작성 버튼	글작성 페이지로 이동합니다.
③ 페이지 네비게이션	<p>페이지를 나타내는 네비게이션 영역입니다.</p> <p>한 페이지에 10개의 페이지 링크를 표시합니다.</p>

The screenshot shows a web browser window with the URL `https://game.rubisco.kro.kr/community`. The page title is "Hex - 게시물 목록". The page layout includes a header with the word "HEX" and a "header" label. Below the header is a navigation bar with links: "nav", "새소식", "게임소개", "커뮤니티", "게임랭킹", and "고객지원". The main content area contains a table of posts with columns: "카테고리", "제목", "닉네임", "등록일", and "조회수". A red box labeled "①" highlights the table. To the right of the table is a "글작성" button, labeled "②". At the bottom, there is a pagination control with links: "<<", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", and ">>". A red box labeled "③" highlights the pagination control.

커뮤니티 게시판 - 게시글 작성 / 수정 페이지

구분	내용
접속 경로	<ul style="list-style-type: none"> - 작성 : /community?act=write - 수정 : /community/{documentId}?act=update
① 게시글 작성 폼	<p>게시글을 작성하기 위한 form 입니다.</p> <p>카테고리를 선택하고, 제목, 내용을 입력한 후 등록버튼을 누르면 게시글이 등록됩니다.</p>
② 공지사항 체크박스	해당 게시글을 공지사항으로 등록합니다. Admin 그룹만 사용가능합니다.
③ 뒤로가기 버튼	게시글 목록 페이지로 이동합니다.
④ 등록/수정 버튼	페이지를 등록 또는 수정합니다.

Hex - 게시글 작성

https://game.rubisco.kro.kr/community?act=write

header HEX

nav 새소식 게임소개 커뮤니티 게임랭킹 고객지원

①

카테고리 제목

내용

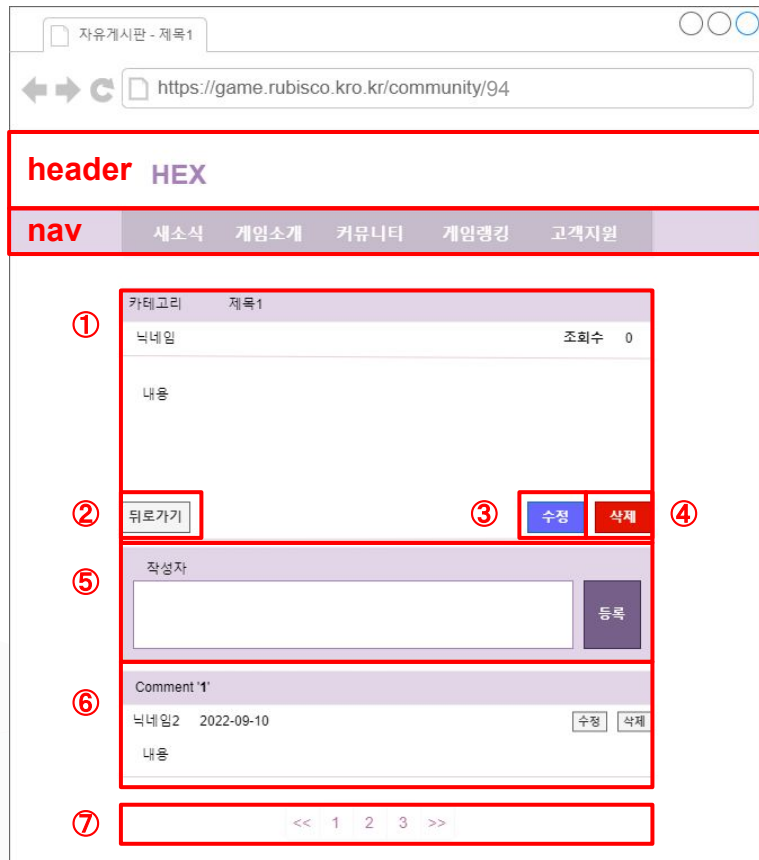
② 공지사항 ☐

③ 뒤로가기

④ 등록

커뮤니티 게시판 - 게시물 페이지

구분	내용
<p>접속 경로 : /community/ {documentId}</p> <p>파라미터 : page - 게시물에 대한 댓글의 페이지</p> <p>size - 한 페이지에 출력되는 댓글 목록 수</p>	
① 메인 콘텐츠 영역	<p>게시글 정보를 출력하는 콘텐츠 영역입니다. 카테고리, 제목, 닉네임, 조회수, 내용이 출력됩니다.</p> <p>해당 영역은 다시 3개의 섹션으로 구분됩니다.</p>
② 뒤로가기 버튼	게시글 목록 페이지로 이동합니다.
③ 수정 버튼	<p>게시글 수정 페이지로 이동합니다.</p> <p>Admin 그룹 또는 게시물 작성자만 호출 가능합니다.</p>



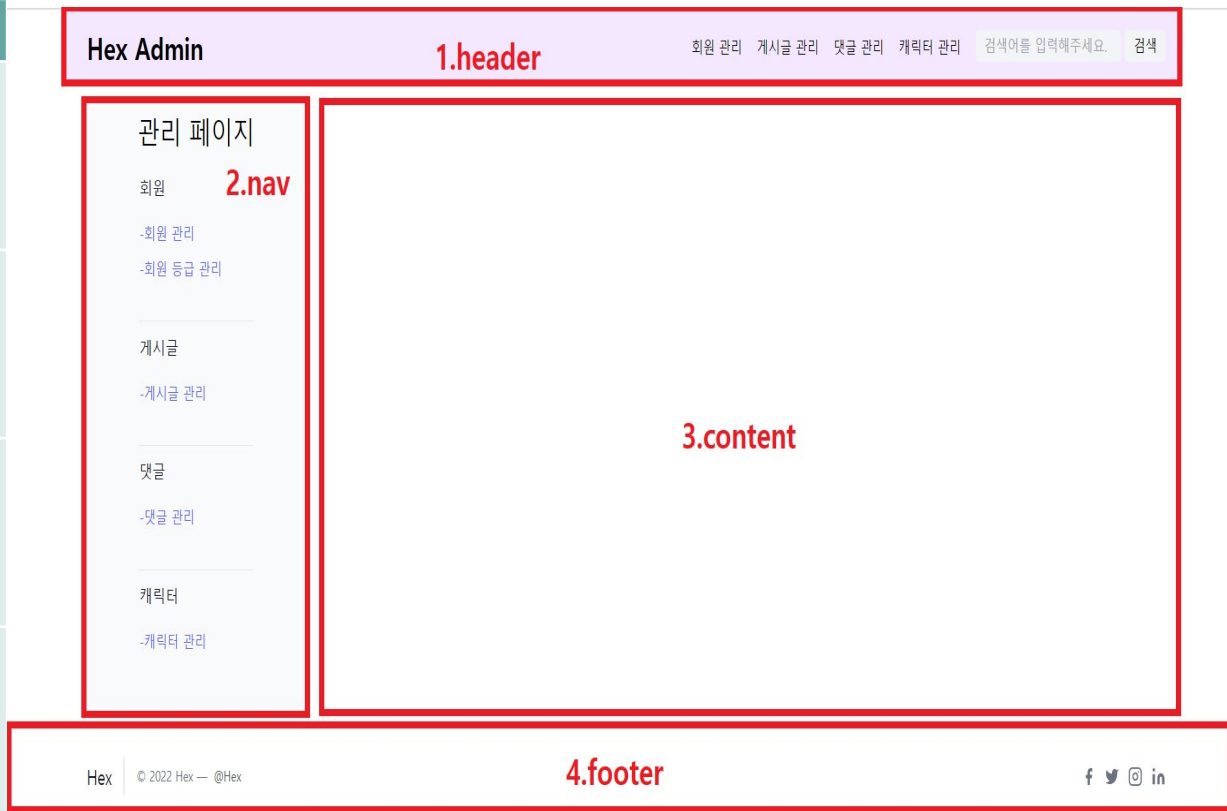
커뮤니티 게시판 - 게시물 페이지

구분	내용
④ 삭제 버튼	게시글을 삭제합니다. Admin 그룹 또는 게시물 작성자만 호출 가능합니다.
⑤ 댓글 작성 섹션	내용을 입력하고 등록 버튼을 누르면 해당 게시물에 댓글을 등록할 수 있습니다.
⑥ 댓글 리스트 섹션	게시글에 대한 댓글의 목록을 나타내는 영역입니다. 댓글에 hover를 하면 댓글을 수정하거나 삭제할 수 있는 메뉴버튼이 보여집니다. 메뉴 버튼은 Admin 그룹 또는 게시물 작성자만 호출 가능합니다.
⑦ 댓글 페이지 네비게이션	댓글의 페이지를 나타내는 네비게이션 영역입니다. 한 페이지에 10개의 페이지 링크를 표시합니다.

The screenshot shows a web browser window with the address bar displaying `https://game.rubisco.kro.kr/community/94`. The page layout includes a header with the title '자유게시판 - 제목1', a navigation bar with links like '새소식', '게임소개', '커뮤니티', '게임랭킹', and '고객지원', and a main content area. The main content area contains a post form and a comment list. Numbered annotations (1-7) highlight specific UI elements: 1 points to the post category and title; 2 points to the '뒤로가기' button; 3 points to the '수정' button; 4 points to the '삭제' button; 5 points to the '작성자' field and '등록' button; 6 points to the comment list header and a comment entry; 7 points to the pagination controls at the bottom.

Admin 레이아웃 - 레이아웃 및 fragment를 통해 페이지 제작

구분	내용
① header	로고, 페이지이동, 서치박스 등을 포함한 고정된 메인 헤더 영역입니다.
② nav	각 관리 페이지에 접속가능한 고정된 메인 네비게이션 영역입니다.
③ content	관리를 위한 테이블이 나오는 영역입니다.
④ footer	고정된 메인 푸터 입니다.



Admin 게시판 - 게시물 페이지

구분	내용
① 게시물 개 수 출력	총 게시물 수, 검색한 게시물 수를 표시합니다.
② 검색 기능 구현	제목, 작성자를 기준으로 키워드를 모두 검색합니다.
③ 게시판 세션	작성된 게시판의 모든 view를 반환합니다. 제목을 누르면 상세보기가 가능합니다.
④ 게시글 선택 가능	해당 게시글을 선택하여 삭제할수 있습니다. All을 누르면 모두 선택합니다.
⑤ 페이지 표시 및 페이지 이동	Next, Previous를 통해 페이지를 이동합니다. 현재 페이지와 총 페이지를 나타냅니다.

Hex Admin

[회원 관리](#)
[게시글 관리](#)
[댓글 관리](#)
[캐릭터 관리](#)
[검색어를 입력해주세요.](#)
[검색](#)

관리 페이지

[회원](#)
[-회원 관리](#)
[-회원 등급 관리](#)

[게시글](#)
[-게시글 관리](#)

[댓글](#)
[-댓글 관리](#)

[캐릭터](#)
[-캐릭터 관리](#)

① 현재 총 969개의 게시물이 존재합니다.

② 검색 내용 선택

검색

All <input type="checkbox"/>	카테고리	제목	작성자	날짜
<input type="checkbox"/>	1	테스트971	vhdlssm0731	2022-10-06T07:46:23.727051
<input type="checkbox"/>	1	테스트970	vhdlssm0731	2022-10-06T07:46:23.722790
<input type="checkbox"/>	1	테스트969	vhdlssm0731	2022-10-06T07:46:23.718030
<input type="checkbox"/>	1	테스트968	vhdlssm0731	2022-10-06T07:46:23.713565
<input type="checkbox"/>	1	테스트967	vhdlssm0731	2022-10-06T07:46:23.709184
<input type="checkbox"/>	1	테스트966	vhdlssm0731	2022-10-06T07:46:23.704738
<input type="checkbox"/>	1	테스트965	vhdlssm0731	2022-10-06T07:46:23.700274
<input type="checkbox"/>	1	테스트964	vhdlssm0731	2022-10-06T07:46:23.695498
<input type="checkbox"/>	1	테스트963	vhdlssm0731	2022-10-06T07:46:23.690908
<input type="checkbox"/>	1	테스트962	vhdlssm0731	2022-10-06T07:46:23.686367

⑤ 현재 페이지 1 / 총 페이지 97

페이지 이동 [Previous/ Next]

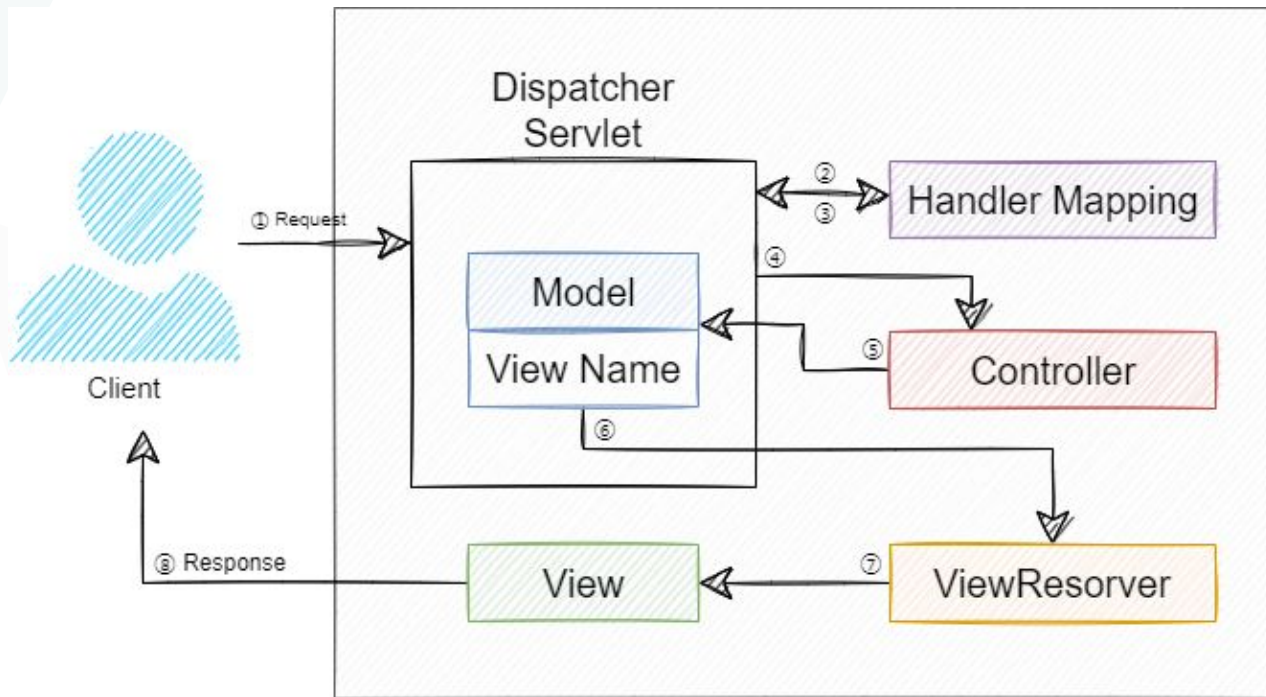
삭제

The background features several overlapping geometric shapes. A large teal triangle is at the top left. A grey triangle is at the bottom left. A teal triangle is on the left side, pointing right. A grey triangle is at the bottom left, pointing up. A teal triangle is at the bottom left, pointing up. A grey triangle is at the bottom left, pointing up.

개발문서

아키텍처 - 스프링 MVC 패턴 적용

스프링 프레임워크를 통해 MVC 패턴을 적용하여 Model, View, Controller로 관점을 나누어 개발합니다.



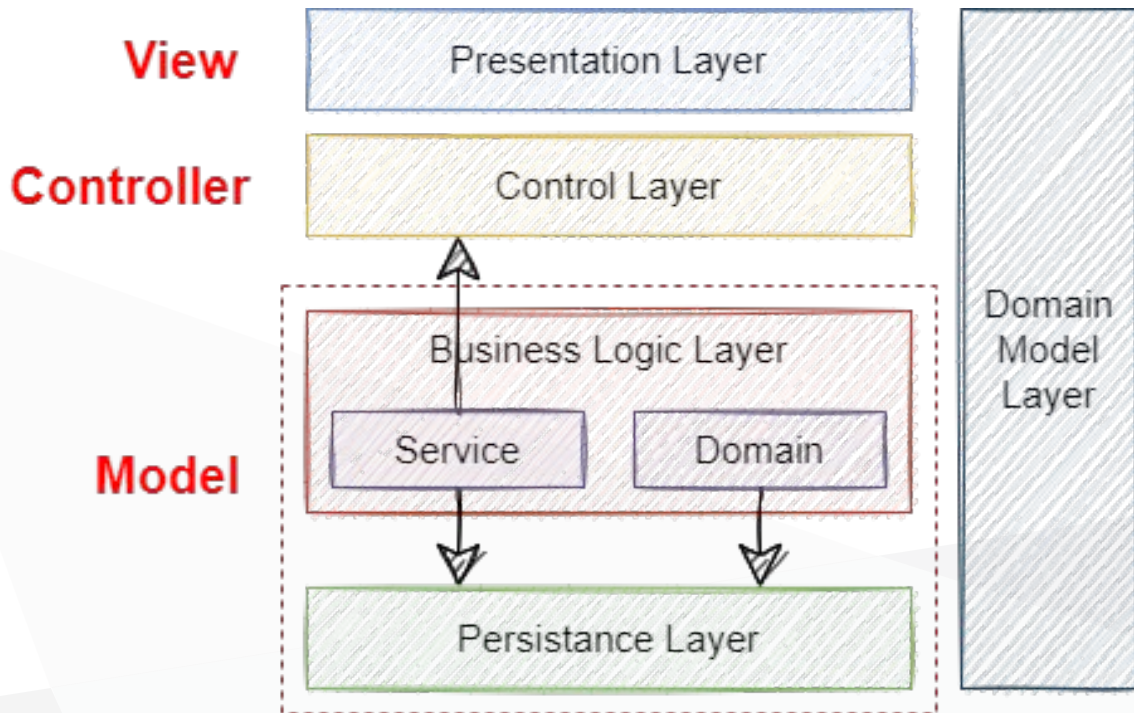
아키텍처 - 스프링 MVC 패턴 적용

스프링 프레임워크를 통해 MVC 패턴을 적용하여 Model, View, Controller로 관점을 나누어 개발합니다.

1. 유저의 Request를 Dispatcher Servlet에서 받습니다.
2. Dispatcher Servlet은 Handler를 통해 유저의 Request를 Controller에 매핑합니다.
3. Controller는 Request에 해당하는 Service를 수행하고 ModelAndView를 리턴합니다.
4. ModelAndView는 ViewResolver에 의해 파싱(Parsing)되어 View를 생성합니다.
5. 생성된 View는 Request를 보낸 유저에게 Response 됩니다.

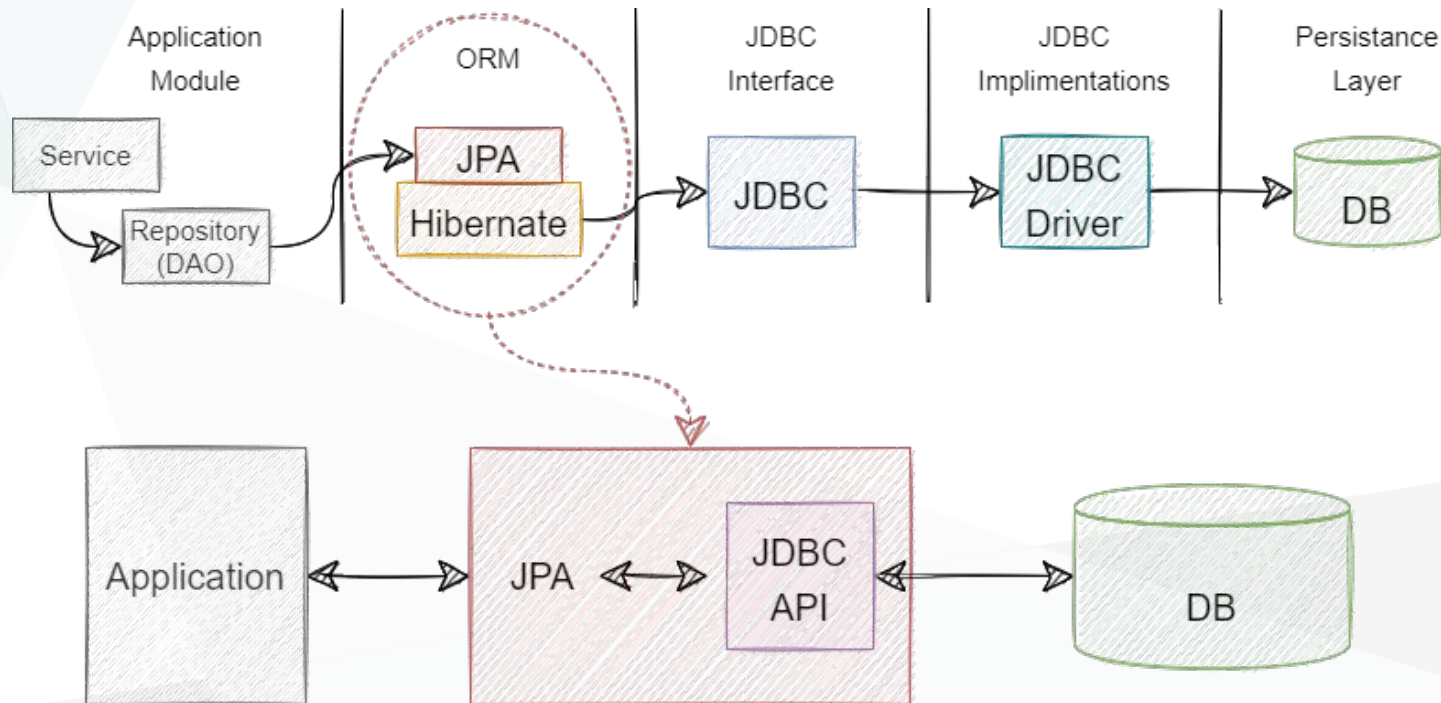
아키텍처 - 스프링 MVC 패턴 적용

MVC 패턴에서 모델부분을 세분화하여 Service, Domain(Entity), Persistence(Repository)로 구분합니다.



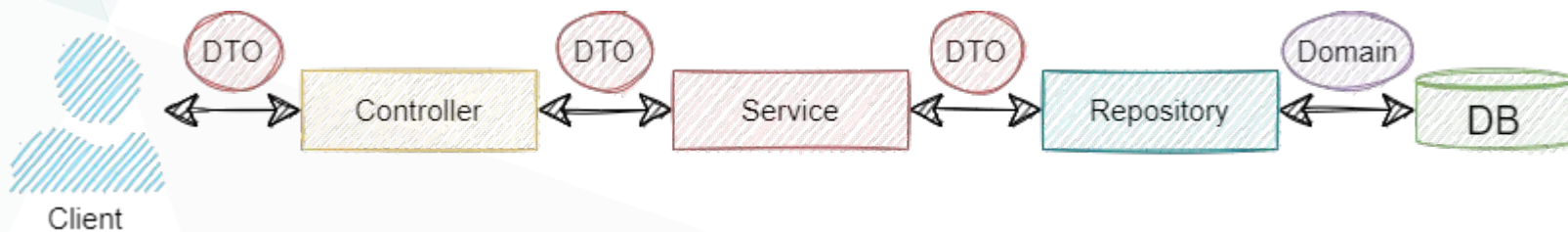
아키텍처 - JPA를 통해 repository 개발

DB와 연동되는 repository(Data Access Object; DAO)는 JPA를 통해 개발합니다.

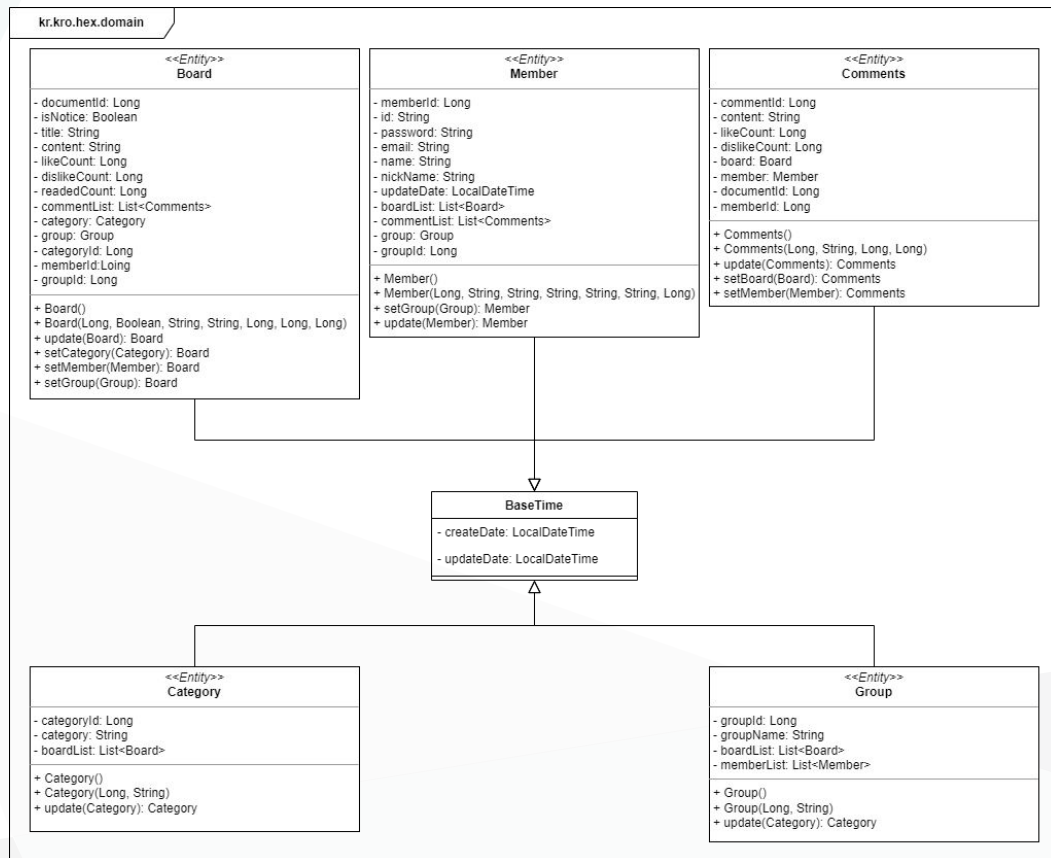


아키텍처 - JPA를 통해 repository 개발

Entity 클래스에 빌더패턴을 적용하여 DTO(Data Transfer Object)대신 사용했으며, 상황에 따라서 DTO를 따로 개발합니다.



Entity - 클래스 다이어그램



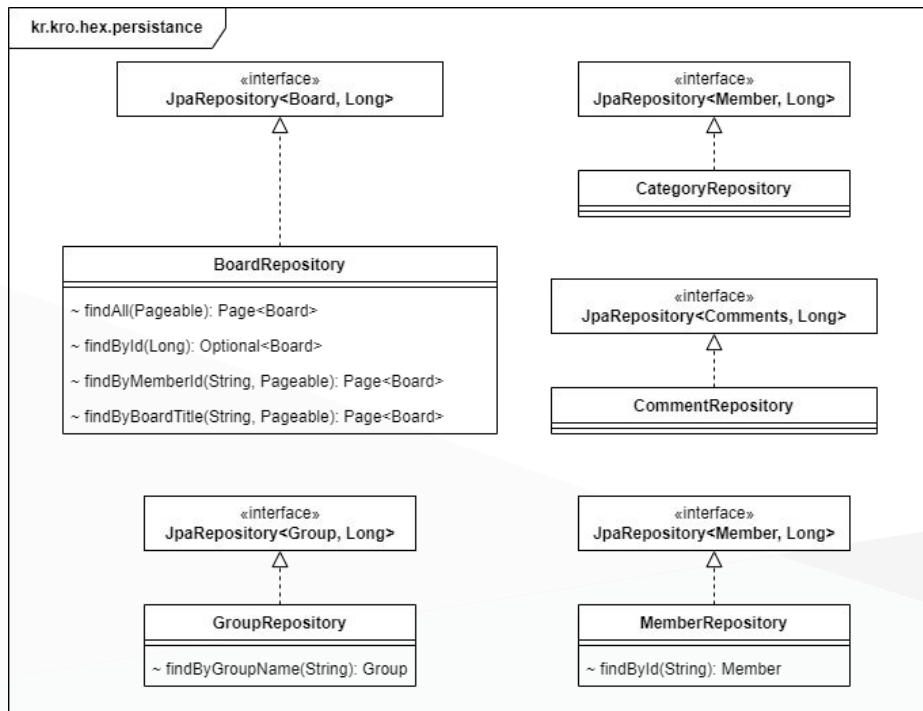
Entity – JPA의 어노테이션을 사용하여 작성

JPA 어노테이션으로 Entity 클래스를 정의하여 개발합니다.

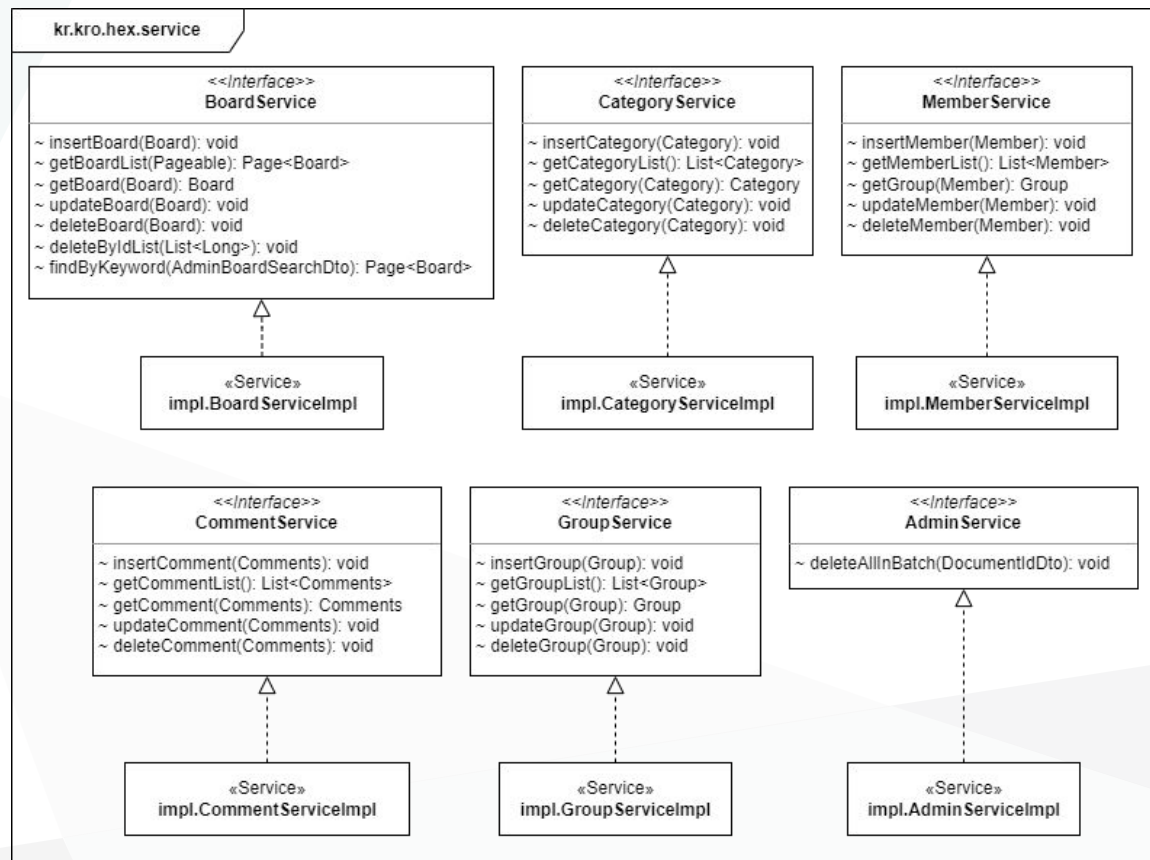
구분	내용
@Entity	Entity 클래스임을 선언합니다. 테이블과 매핑됩니다.
@Id	해당 속성이 Entity의 기본키(PK)임을 선언합니다. 테이블의 PK와 매핑됩니다.
@GeneratedValue	기본키를 생성하는 전략을 설정합니다. (기본값 auto)
@Column	테이블의 column과 매핑됩니다.
@ColumnDefault	테이블 스키마의 기본값을 설정합니다. DDL 쿼리문에서 유효합니다.
@OneToMany	1:N 연관관계 매핑을 설정합니다.
@ManyToOne	N:1 연관관계 매핑을 설정합니다.
@BatchSize	1:N 연관관계 매핑에서 하나의 쿼리문에 매핑되는 최대 entity 수를 설정합니다.
@JoinColumn	연관관계 매핑에서 외래키(FK)로 사용할 column을 설정합니다. 연관관계의 주인이 됩니다.
@Transient	해당 속성을 Entity 매핑에서 제외합니다.
@Lob	Large Object의 약자로, 용량이 큰 데이터를 저장하기 위해 설정합니다.

Repository - 클래스 다이어그램

스프링 부트의 Spring Data JPA를 통해 repository를 개발합니다. JpaRepository를 상속받아 구현합니다.



Service — 클래스 다이어그램



Service – @Service 어노테이션으로 서비스 클래스 선언

@Service 어노테이션을 붙여 서비스 클래스임을 선언하고 비즈니스 로직을 구현합니다.

1. @RequiredArgsConstructor 어노테이션을 붙여서 Repository 의존성을 생성자 주입합니다.
2. @Transactional 어노테이션을 붙여서 트랜잭션 처리를 합니다.
3. Entity 클래스를 DTO로 사용하여 비즈니스 로직을 구현합니다.

Controller — @Controller 어노테이션으로 컨트롤러 클래스 선언

@Controller 어노테이션을 붙여 컨트롤러 클래스임을 선언하고 유저의 요청에 대하여 서비스와 뷰를 연결합니다.

1. @RequiredArgsConstructor 어노테이션을 붙여서 Service 의존성을 생성자 주입합니다.
2. @RequestMapping 어노테이션을 붙여서 유저의 요청을 매핑합니다.
3. @RequestMapping는 @GetMapping, @PostMapping, @PutMapping, @PatchMapping, @DeleteMapping으로 세분화할 수 있습니다.
4. @Controller 대신 @RestController로 컨트롤러를 선언하면 @ResponseBody 어노테이션이 붙이지 않더라도 View 대신에 객체 자체를 Response 할 수 있습니다.

Controller — Request Mapping

클래스	메소드	기능	URI
BoardController	getBoardListView	게시글 목록 조회	GET /community
	getBoardView	게시글 조회	GET /community/ {documentId}
	insertBoardView	게시글 등록 페이지	GET /community?act=write
	updateBoardView	게시글 수정 페이지	GET /community?act=update
	insertBoard	게시글 등록 요청	POST /community
	updateBoard	게시글 수정 요청	PATCH /community -d documentId= {documentId}
	deleteBoard	게시글 삭제 요청	DELETE /community -d documentId= {documentId}
CommentController	insertComment	댓글 등록 요청	POST /comment -d documentId= {documentId}
	updateComment	댓글 수정 요청	PATCH /comment -d commentId= {commentId}
	deleteComment	댓글 삭제 요청	DELETE /comment -d commentId= {commentId}
CommentRestController	deleteCommentResponse	댓글 삭제 요청 (REST)	DELETE /api/comment -d commentId= {commentId}
PageController	getGameView	게임 페이지	GET /hex

Controller — Request Mapping

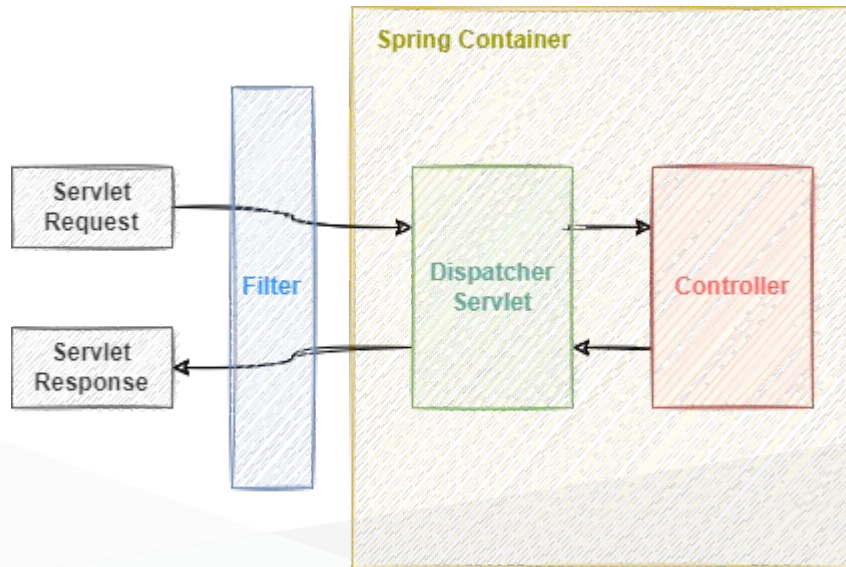
클래스	메소드	기능	URI
MemberController	getSignUpView	회원가입 페이지	GET /member?act=signUp
	insertMember	회원가입 요청	POST /member
	getMemberInfoView	회원정보 페이지	GET /member
	updateMember	회원정보 수정 요청	PATCH /member
	deleteMember	회원탈퇴 요청	DELETE /member

Login - 인증 / 인가 시스템 구현

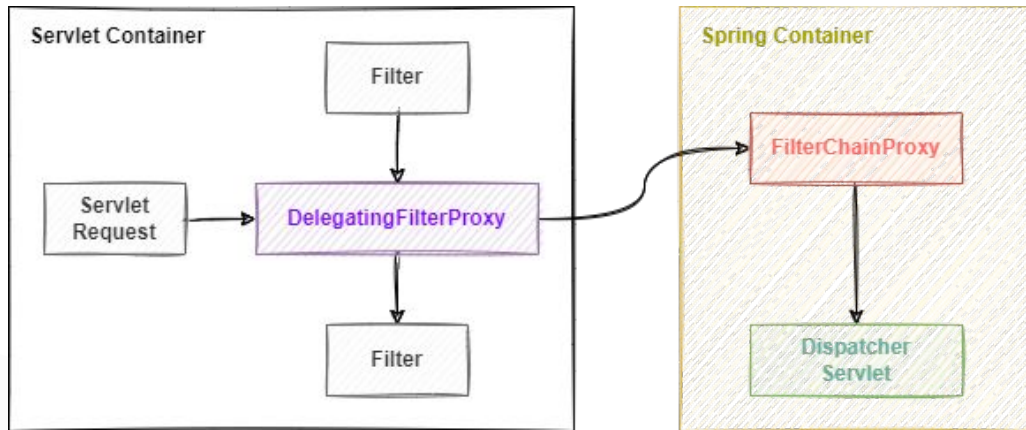
Spring Security를 통해 인증(Authentication)과 인가(Authorization) 시스템을 구현합니다.

유저의 Request는 DispatcherServlet에 오기전에 Servlet Filter를 거칩니다.

Servlet Filter는 Servlet Container에서 생성되기 때문에 Spring Bean을 주입할 수 없습니다.



Login - 인증 / 인가 시스템 구현



Spring Security에서는 **DelegatingFilterProxy**를 Servlet Container의 **Servlet Filter**로 등록하고 유저의 Request를 Spring Container의 **FilterChainProxy**에 위임합니다.

FilterChainProxy는 Spring Bean으로 등록된 일련의 **SecurityFilterChain**을 수행합니다. 그러므로 **Security**에서 제공하는 **Filter**를 통해 **SecurityFilterChain**을 개발하고 Bean으로 등록하면 **Login** 기능을 구현할 수 있습니다.

개선사항

1. 잘못된 요청에 대한 예외처리가 부족합니다. 잘못된 요청이 오면 에러 페이지로 **redirect**할 예정이며, **REST**의 경우 적절한 메시지를 **Response** 하여 예외 처리를 할 예정입니다.
2. **CSRF** 방어코드가 필요합니다. 요청을 보내는 폼마다 **CSRF** 토큰 코드를 넣어서 **CSRF** 방어처리를 할 예정입니다.
3. 회원가입시 유효성 검사가 필요합니다. **front**에서 1차적으로 유효성 검사를 하도록 스크립트를 작성하고, 컨트롤러에서 다시 한번 유효성 검사를 하여 에러 메시지를 전송하도록 할 예정입니다.
4. 회원가입시 메일 인증이 필요합니다. 암호 분실시 인증된 메일을 통해 임시 비밀번호를 보내고 비밀번호 변경하도록 시스템을 구축할 예정입니다.
5. 게임과 연동이 필요합니다. 로그인을 통해 게임 캐릭터를 생성할 수 있으며, **RDB**를 통해 저장하기 보다는 **NoSQL**을 통해 게임 캐릭터를 저장하고 **Member**와 연관관계를 맺을 예정입니다.
6. 검색기능이 부실합니다. 헤더에 서치 인풋 박스가 있지만 구현하지 않았습니다. 인가 권한에 따라 다르게 검색되도록 구현할 예정입니다.