Poco Poco

친화적인 온라인 영상 템플릿 판매 플랫폼

영상 템플릿 판매 및 수주

VPLAY

제작 날짜
2025.03.24 (월)

Sales and Orders
Video Templates|

TABLE OF **CONTENTS**

01 팀 소개 및 프로젝트 개요
02 프로젝트 개발 배경
03 벤치마킹 사이트와 차별화
04 프로젝트 산출물
05 코드 리뷰
06 시연

CHAPTER 1

팀 소개 및 프로젝트 주요 목표

팀 구성원 소개

팀장 : 권태혁

구성원

- 안효빈
- 손동준
- 이상윤

01 사용자 친화적인 템플릿 마켓플레이스 구축

02 안정적인 결제 및 라이선스 관리 시스템 개발

03 크리에이터를 위한 판매 및 정산 시스템 제공

04 마케팅 최적화로 트래픽 유입 극대화

CHAPTER 2 - 1

개발배경
국내 영상 템플릿 마켓 부재

대부분의 템플릿 플랫폼은 글로벌 서비스를 기반으로 하고 있어,
언어 장벽과 결제 시스템의 제약,
그리고 현지화된 콘텐츠 부족 등으로 인해 국내 사용자들에게는 불편하다

국내 영상 템플릿 마켓의 부재, 크리에이터들의 불편 증가

국내에서 영상 템플릿을 판매하는 전문 플랫폼이 없어....

조선일보

해외 결제 의존… 국내 창작자들의 수익 창출 기회 제한

외화 결제, 높은 수수료, 환전 과정에서 발생하는 불이익 등으로...

동아일보

CHAPTER 2 - 2

개발배경
국내 소규모 창작자와 기업들의 영상 제작 수요 증가

국내 시장에서는 소규모 창작자나 중소기업, 개인 사업자들이
간단하고 빠르게 영상을 제작하고자 하는 니즈가 꾸준히 증가하고 있음에도,
이들이 손쉽게 활용할 수 있는 로컬화된 영상 템플릿 서비스는
거의 전무한 실정이다

앱애니, "한국인 동영상 편집앱 지출 2년새 3배 증가"

한국 스마트폰 이용자의 동영상 플레이어 및 편집앱 지출이 2년새 3배 이상 증가...

뉴스웍스

'동영상 콘텐츠 수요 폭증'… 콘텐츠 제작 중개 플랫폼들 성장전망↑

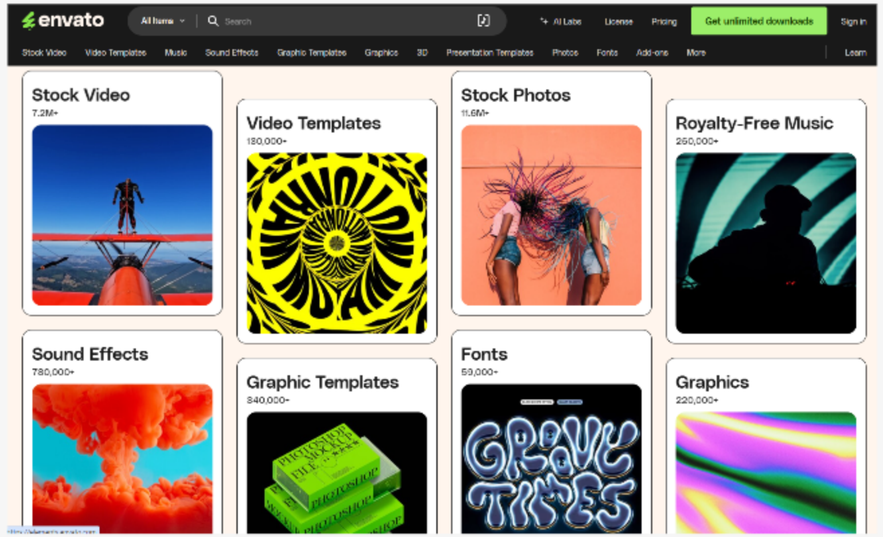
유튜브 등 동영상 노출 매체들 많아지면서
제작 수요 급증...

THE STOCK

CHAPTER 3 - 1

밴치 마킹 사이트 및 내용

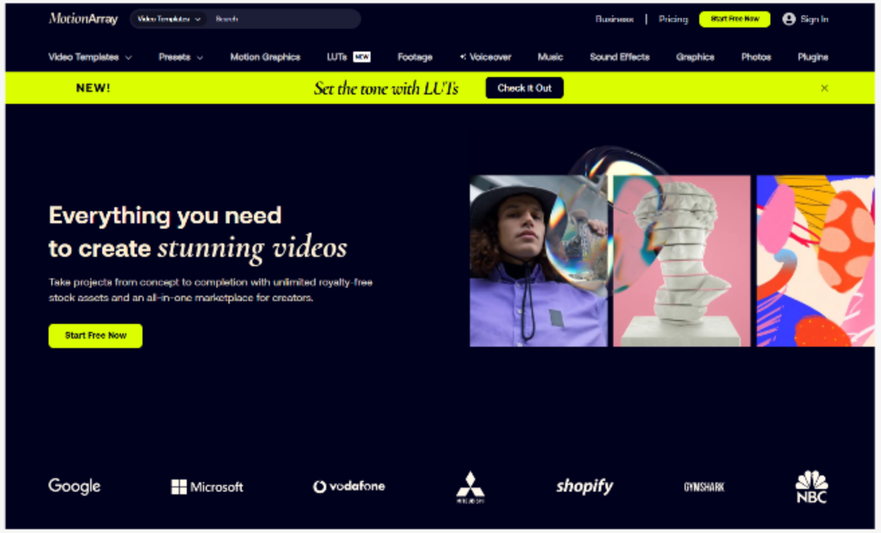
01



ENVATO

대표적인 영상 템플릿 판매 사이트
stock 비디오도 판매

02



MOTION ARRAY

영상 템플릿 판매 사이트로
모션그래픽이 위주

CHAPTER 3-2

차별화 전략

국내 이용자들이 쉽게 활용할 수 있도록
커뮤니케이션 기능, 한국어 지원, 튜토리얼 제공,
그리고 영상 제작 수주 시스템을 갖춘 영상 템플릿 플랫폼 구축.

01 유저 간 커뮤니케이션 기능 추가

구매자와 판매자가 직접 소통할 수 있도록 댓글, 리뷰, Q&A 게시판 등의 기능을 제공하여
피드백과 의견 교환이 원활하게 이루어지도록 함.

02 영어를 못하는 국내 이용자를 위한 사이트

해외 플랫폼의 불편함을 해소하기 위해 한국어 지원과 직관적인 UI를 제공하여
국내 이용자들이 언어 장벽 없이 편리하게 이용할 수 있도록 함

03 템플릿 사용법에 대한 튜토리얼 제공

초보자들도 쉽게 활용할 수 있도록 영상 템플릿 사용법을 안내하는
가이드 및 영상 튜토리얼을 제공하여 접근성을 높임

04 영상 제작 수주

영상 템플릿을 구매하는 것뿐만 아니라, 맞춤형 영상 제작을 원하는 고객이
크리에이터에게 직접 의뢰할 수 있는 기능을 추가하여 수익 모델을 다변화함

프로젝트 산출물

CHAPTER 4

요구 사항 정의서

대분류	요구사항 명	요구사항 내용
회원, 관리자		
공통 기능	로고	- 클릭시 메인 페이지로 이동
	sns	- 인스타, 틱톡, 유튜브 등 sns 버튼과 클릭시 해당 페이지로 이동
	이용약관 - 개인정보 처리방침	- 이용약관 버튼과 클릭시 해당 페이지로 이동
회원 관리	회원가입	- 아이디는 5자 이상 15자 미만으로 작성, 중복 확인 - 닉네임은 2자 이상 8자 미만으로 작성, 중복 확인 - 이름을 작성하는 칸 - 비밀번호는 비밀번호는 8~15자 사이의 영어숫자특수문자로 구성, 비밀번호는 '*'로 노출 - 생년월일을 입력할 수 있는 기능으로 입력 시 생년월일이 그대로 노출 - 전화번호를 입력할 수 잇는 기능으로 입력 시 전화번호가 그대로 노출, 중복 확인 - 비밀번호 힌트 유형을 선택할 수 있게 해준다 - 비밀번호 답변을 입력할 수 있는 기능으로, 입력 시 답변이 그대로 노출 - 개인약관 동의 체크박스, 동의하면 회원가입 성공, 안하면 회원가입안되게 - SNS 로그인 지원 (Google, Facebook, 카카오, 네이버)
	로그인	- 이메일 및 비밀번호 입력 후 가입 가능 - SNS 로그인 지원 (Google, Facebook, 카카오, 네이버) - 소셜 로그인 기능 추가 - 아이디 및 비밀번호 입력 후 로그인 - 아이디 입력 시 아이디가 그대로 노출 - 비밀번호 입력시 비밀번호가 '*'로 노출(비밀번호 암호화 기능 추가) - 로그인버튼 클릭 시 위에 있는 아이디와 비밀번호가 사용자 정보와 일치할 시 로그인 되며 메인페이지로 이동 - 로그인버튼 클릭 시 위에 있는 아이디와 비밀번호가 사용자 정보와 일치하지 않을 때 경고 문구 - 로그인 유지 기능 제공
	아이디 찾기	- 이름, 이메일 입력 - 이름과 이메일이 일치할 때 인증번호 이메일로 제공 - 인증번호 일치시 아이디 제공
	비밀번호 찾기	- 이름, 아이디, 이메일 입력 - 이름, 아이디, 이메일이 일치할 때 인증번호 이메일로 제공 - 인증번호 일치시 새비밀번호 입력 후 적용
	로그아웃	- 로그아웃 버튼 클릭 시 세션 종료
마이페이지	내 정보	- 사용자가 본인의 정보를 확인하고 수정 할 수 있음(아이디 제외)
	문의 내역	- 사용자가 지금까지 한 문의내용을 답변과 함께 한 번에 볼 수 있음 - 문의한 내용에 대한 답변 상태를 표시
	장바구니 내역	- 사용자가 장바구니 버튼을 누르면 영상에 대한 내용이 장바구니에 담김
	업로드 내역	- 해당 사용자가 업로드한 프로젝트 템플릿 및 영상 내역 확인
	쪽지 내역	- 사용자들과 주고 받은 쪽지 내역
	구독 내역	- 현재 내 구독상태 확인가능 - 구독료 지불 내역 제공
	즐겨찾기 내역	- 즐겨찾기한 프로젝트 및 게시물 제공

CHAPTER 4

요구 사항 정의서

결제 및 라이선스 관리	상품 결제	- 상품 구매시 현금으로 결제
	라이선스 관리	- 콘텐츠별 라이선스 유형 설정 가능 - 다운로드 시 라이선스 문서 자동 발급
소셜 및 커뮤니티	이벤트	- 진행 중인 이벤트 목록 및 상세 페이지 제공 - 첫 구매시 할인 이벤트
	SNS 연동	- SNS 공유 버튼 제공 (Instagram, TikTok, YouTube 등)
커뮤니케이션	문의하기	- 사용자가 각종 버그나, 요청내용을 관리자한테 쪽지로 문의 가능
	댓글	- 게시판에 댓글을 달고 , 수정, 삭제 가능
	쪽지	- 사용자끼리 쪽지를 주고 받을 수 있음
컨텐츠 다운로드	다운로드	- 비회원일시 샘플만 다운 가능하고 , 회원가입 후 결제시 원본 동영상 다운 가능
	업로드	- 회원이 판매하고자 하는 영상을 업로드 - 해당 템플릿 사용 영상 업로드(선택)
	미리보기	- 영상다운 전에 업로더가 올린 미리보기영상으로 확인 가능
	장바구니	- 한번에 하나의 상품만 결제 하는게 아닌 여러 상품을 담아 한 번에 결제 가능
	즐겨찾기	- 사용자가 즐겨찾기를 누르면 마이페이지에서 바로 즐겨찾기했던 영상게시판으로 이동 가능
관리자		
회원 정보 관리	삭제 여부	- status 'Y'일 경우 현재 사용 회원, 'N'일 경우 사용 중지 회원
	관리자 여부	- admin이 Y 일경우 관리자 N일 경우 일반 사용자
문의 관리	문의 답변	- 문의에 대한 답변을 댓글로 달 수 있음
	문의 삭제 여부	- 문의 내용을 삭제 할 수 있음
게시물 관리	게시물 삭제 여부	- 게시글을 삭제 가능
	댓글 삭제 여부	- 댓글을 삭제 가능

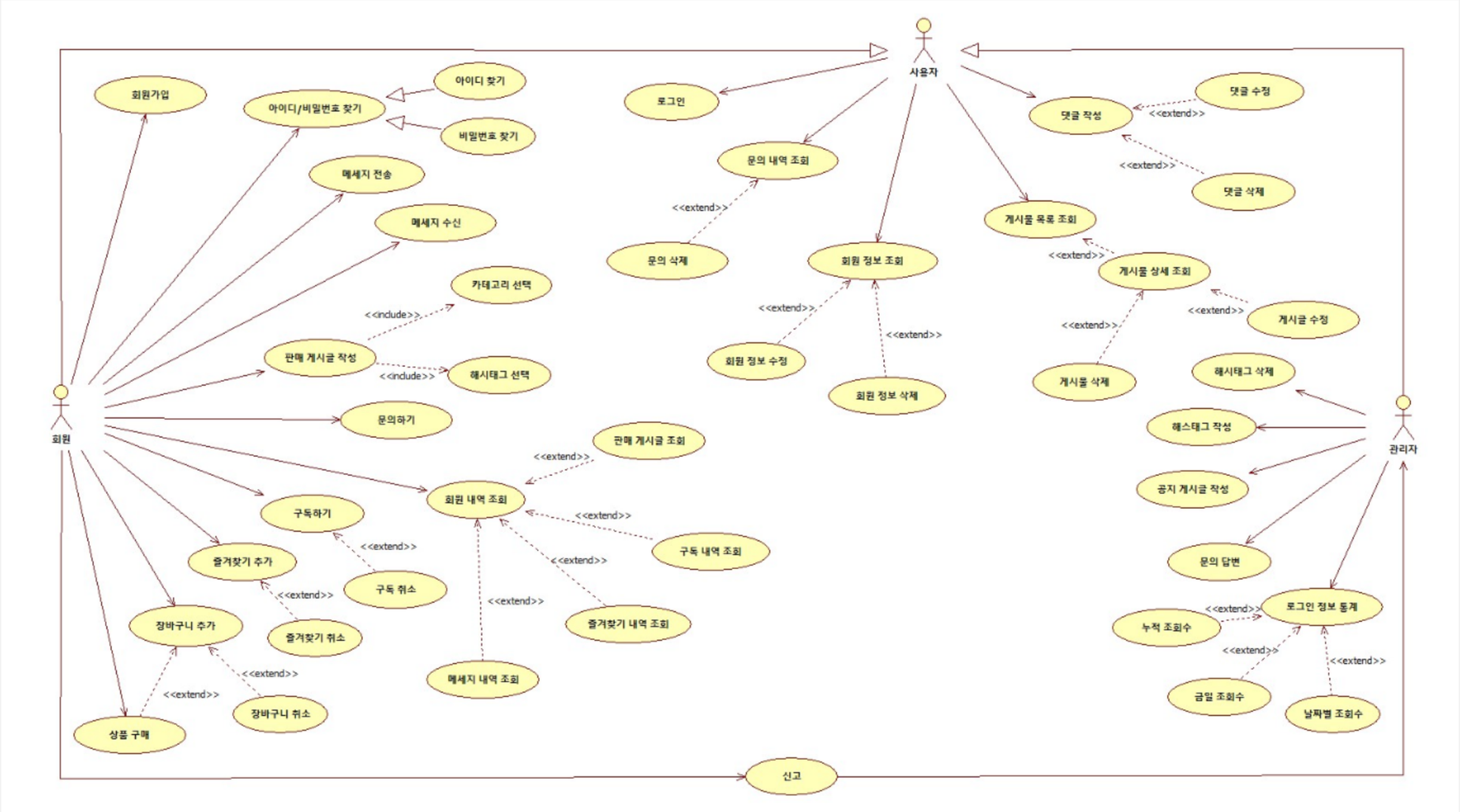
CHAPTER 4

단위업무 정의서

회원, 관리	세부 없음	
공통 기능	로고	- 클릭시 메인 페이지로 이동
공통 기능	sns	- 인스타, 틱톡, 유튜브 등 sns 버튼과 클릭시 해당 페이지로 이동
공통 기능	이용약관	- 이용약관 버튼과 클릭시 해당 페이지로 이동
회원 관리	회원가입	- 아이디는 5자 이상 15자 미만으로 작성, 중복 확인- 닉네임은 2자 이상 8자 미만으로 작성, 중복 확인- 이름을 작성하는 칸- 비밀번호는 비밀번호는 8~15자 사이의 영어숫자특수문자로 구성, 비밀번호는 '*'로 노출- 생년월일을
회원 관리	로그인	- 이메일 및 비밀번호 입력 후 가입 가능 - SNS 로그인 지원 (Google, Facebook, 카카오, 네이버)- 소셜 로그인 기능 추가- 아이디 및 비밀번호 입력 후 로그인- 아이디 입력 시 아이디가 그대로 노출- 비밀번호 입력시 비밀번호가
회원 관리	아이디 찾	- 이름, 이메일 입력- 이름과 이메일이 일치할 때 인증번호 이메일로 제공- 인증번호 일치시 아이디 제공
회원 관리	비밀번호	- 이름, 아이디, 이메일 입력- 이름, 아이디, 이메일이 일치할 때 인증번호 이메일로 제공- 인증번호 일치시 새비밀번호 입력 후 적용
회원 관리	로그아웃	- 로그아웃 버튼 클릭 시 세션 종료
회원 정보	내 정보	- 사용자가 본인의 정보를 확인하고 수정 할 수 있음(아이디 제외)
회원 정보	문의 내역	- 사용자가 지금까지 한 문의내용을 답변과 함께 한 번에 볼 수 있음- 문의한 내용에 대한 답변 상태를 표시
회원 정보	장바구니	- 사용자가 장바구니 버튼을 누르면 영상에 대한 내용이 장바구니에 담김
회원 정보	업로드 내	- 해당 사용자가 업로드한 프로젝트 템플릿 및 영상 내역 확인
회원 정보	쪽지 내역	- 사용자들과 주고 받은 쪽지 내역
회원 정보	구독 내역	- 현재 내 구독상태 확인가능- 구독료 지불 내역 제공
회원 정보	즐겨찾기	- 즐겨찾기한 프로젝트 및 게시물 제공
결제 및 리	상품 결제	- 상품 구매시 현금으로 결제
결제 및 리	라이선스	- 콘텐츠별 라이선스 유형 설정 가능 - 다운로드 시 라이선스 문서 자동 발급
소셜 및 커	이벤트	- 진행 중인 이벤트 목록 및 상세 페이지 제공- 첫 구매시 할인 이벤트
소셜 및 커	SNS 연동	- SNS 공유 버튼 제공 (Instagram, TikTok, YouTube 등)
커뮤니케이	문의하기	- 사용자가 각종 버그나, 요청내용을 관리자한테 쪽지로 문의 가능
커뮤니케이	쪽지	- 사용자끼리 쪽지를 주고 받을 수 있음
컨텐츠 다	다운로드	- 비회원일시 샘플만 다운 가능하고 , 회원가입 후 결제시 원본 동영상 다운 가능
컨텐츠 다	업로드	- 회원이 판매하고자 하는 영상을 업로드- 해당 템플릿 사용 영상 업로드(선택)
컨텐츠 다	미리보기	- 영상다운 전에 업로더가 올린 미리보기영상으로 확인 가능
컨텐츠 다	장바구니	- 한번에 하나의 상품만 결제 하는게 아닌 여러 상품을 담아 한 번에 결제 가능
컨텐츠 다	즐겨찾기	- 사용자가 즐겨찾기를 누르면 마이페이지에서 바로 즐겨찾기했던 영상게시판으로 이동 가능
관리자	세부 없음	
회원 정보	삭제 여부	- status 'Y'일 경우 현재 사용 회원, 'N'일 경우 사용 중지 회원
회원 정보	관리자 여	- admin이 Y 일경우 관리자 N일 경우 일반 사용자
문의 관리	문의 답변	- 문의에 대한 답변을 댓글로 달 수 있음
문의 관리	문의 삭제	- 문의 내용을 삭제 할 수 있음

CHAPTER 5

유스케이스 다이어그램

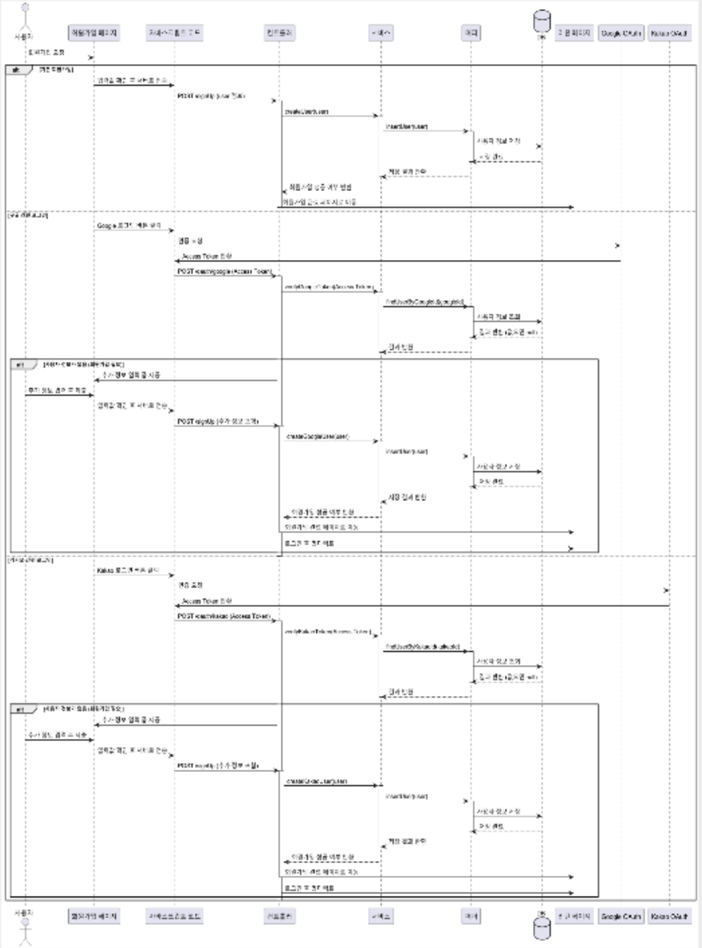


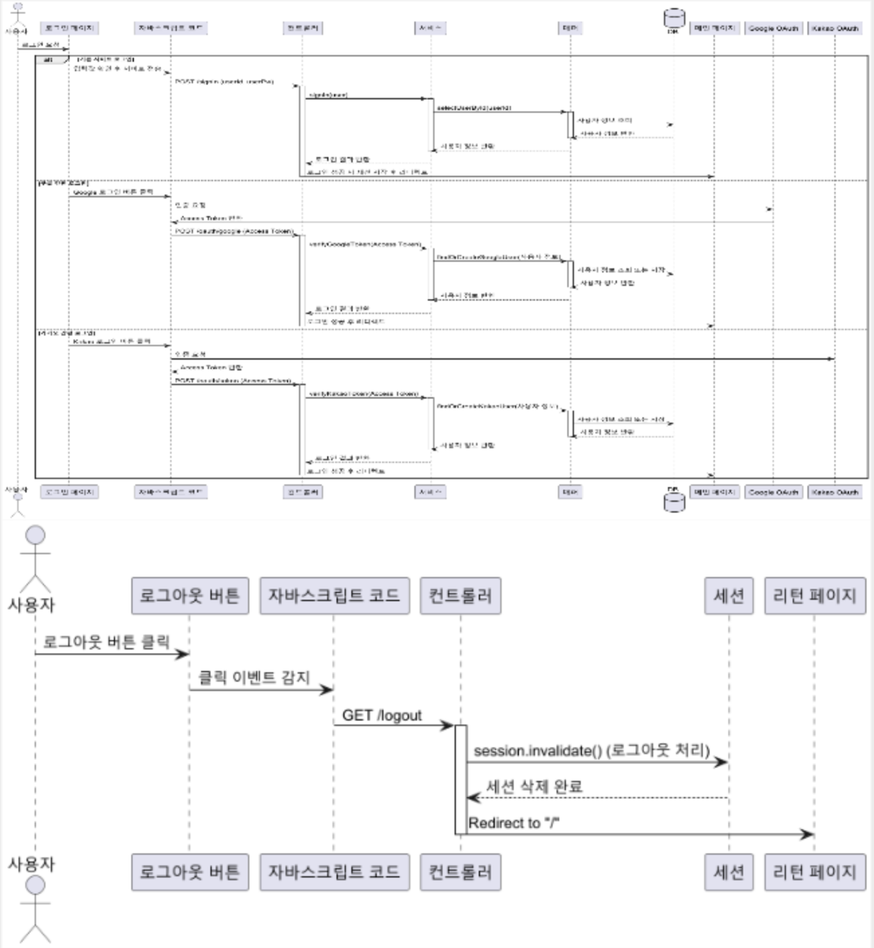
CHAPTER 5

클래스 다이어그램

CHAPTER 6-1

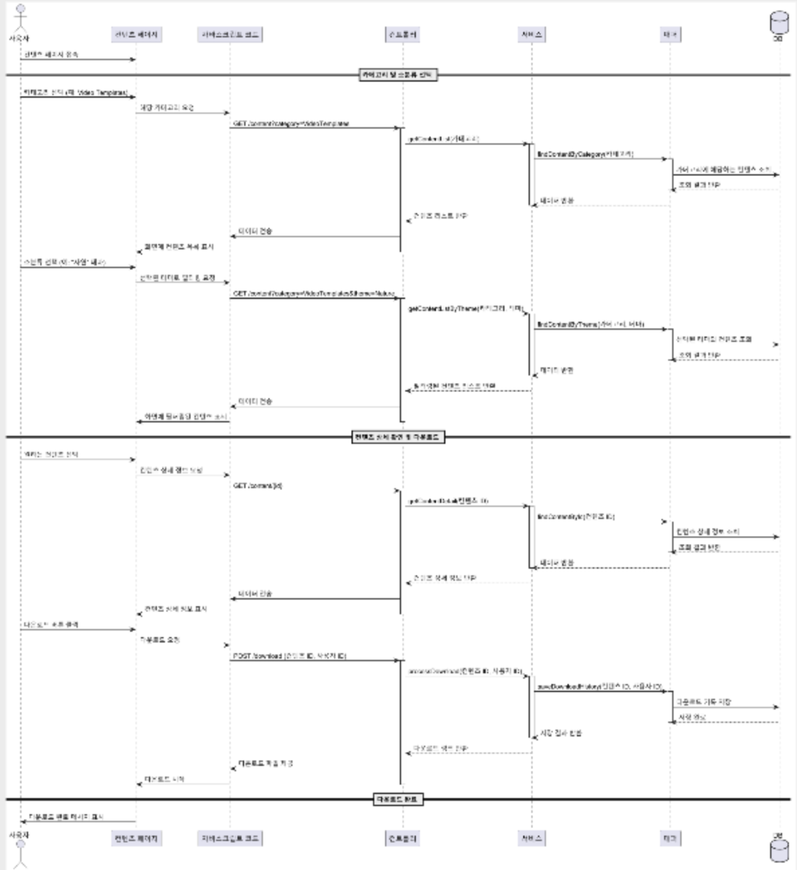
시퀀스 다이어그램

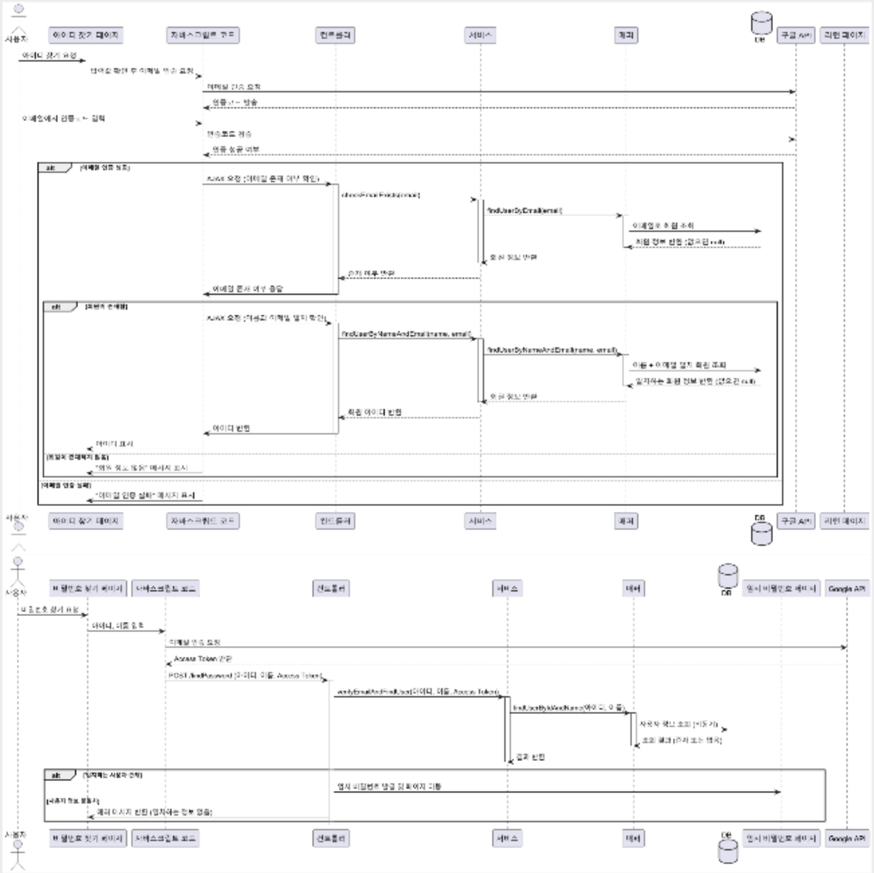




CHAPTER 6-2

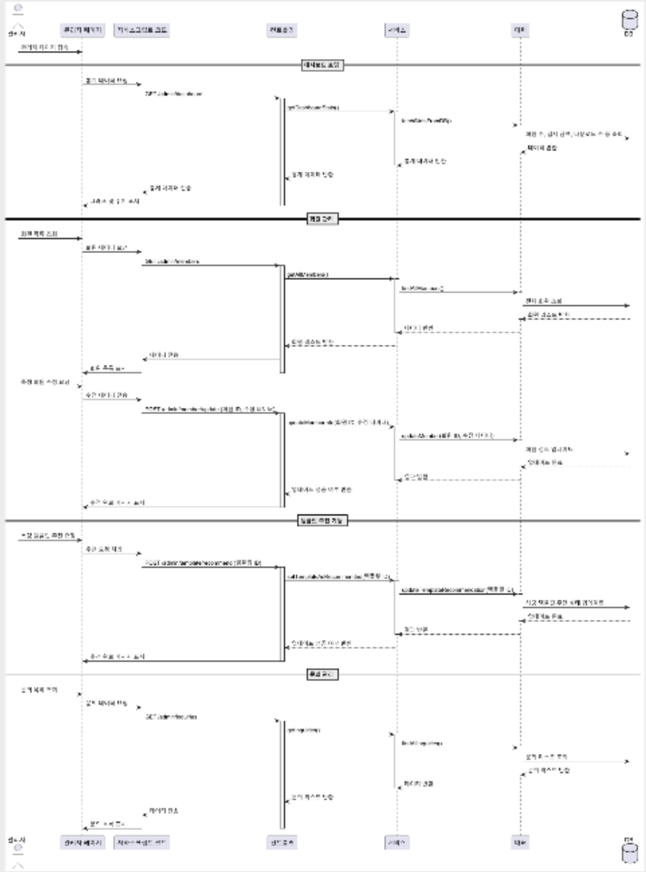
시퀀스 다이어그램

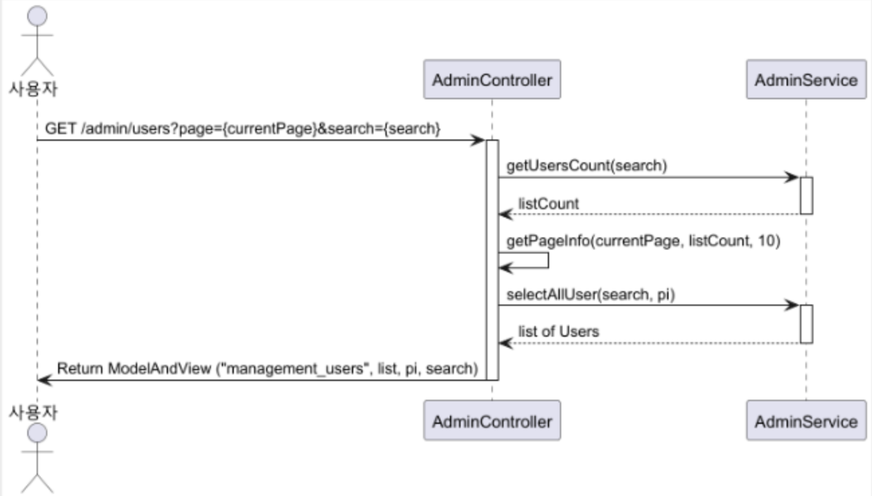




CHAPTER 6-3

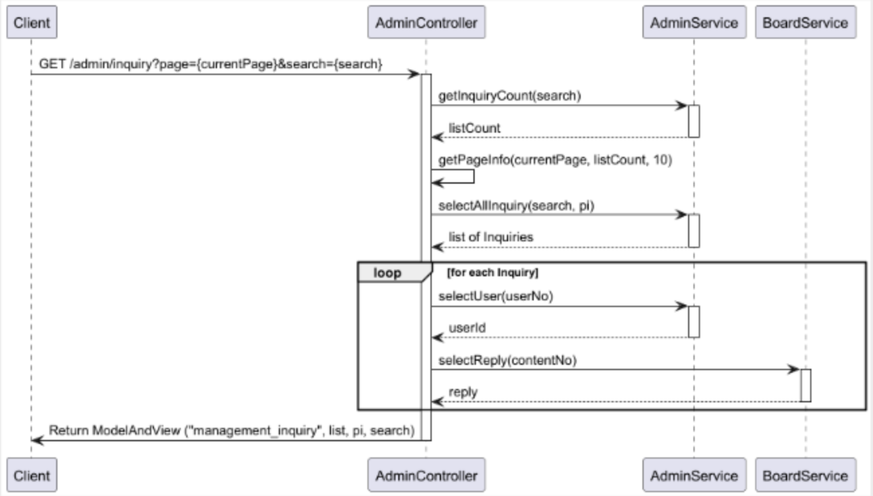
시퀀스 다이어그램

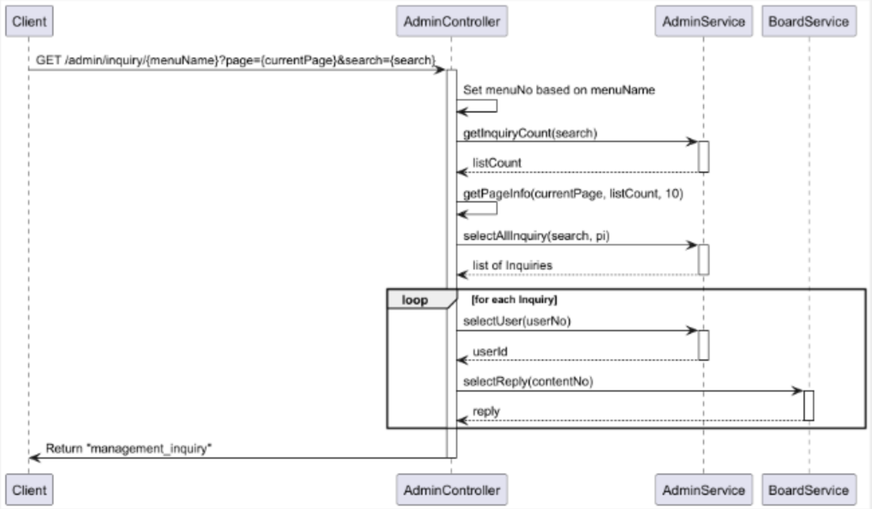




CHAPTER 6-4

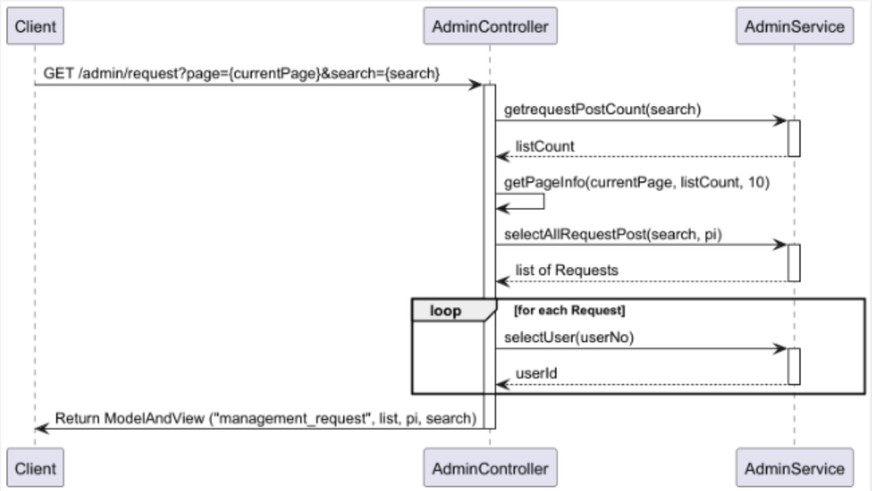
시퀀스 다이어그램

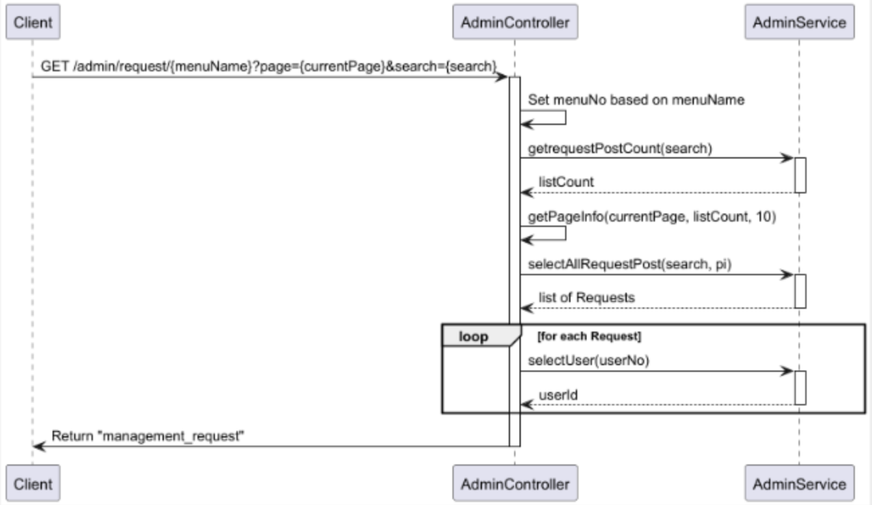




CHAPTER 6-5

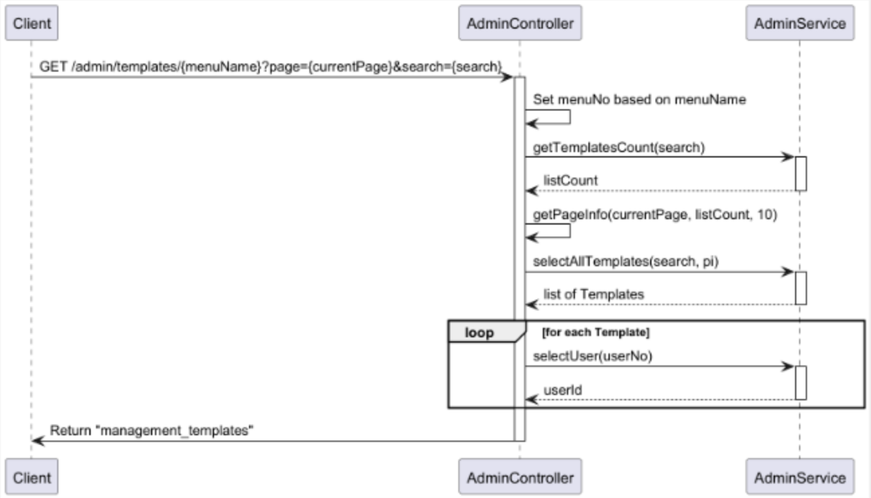
시퀀스 다이어그램

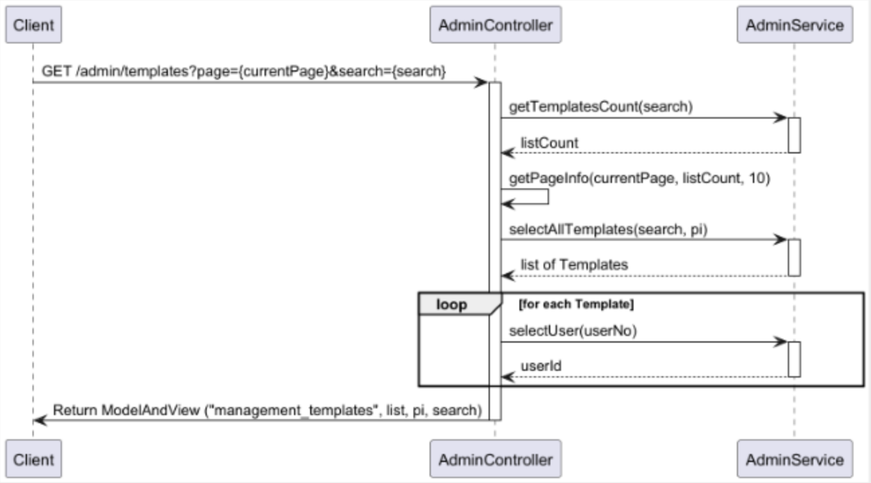




CHAPTER 6-6

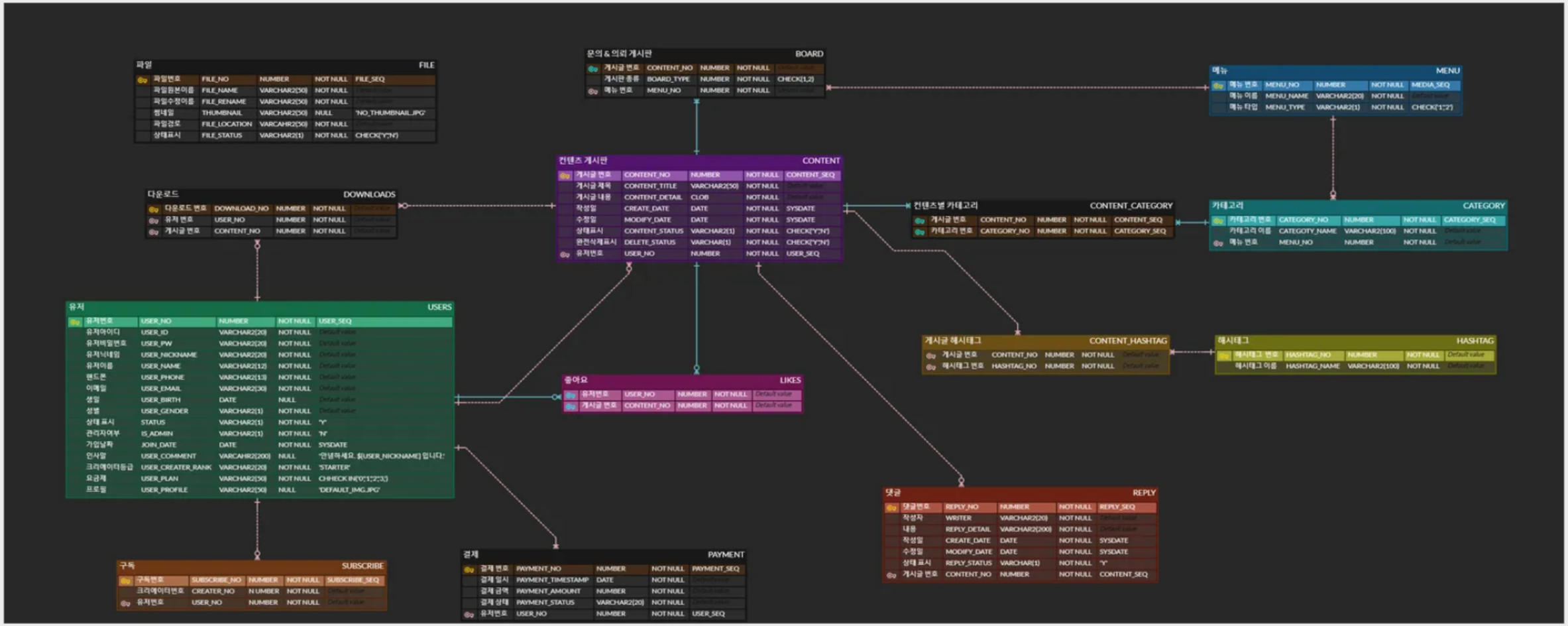
시퀀스 다이어그램





CHAPTER 7

ERD



CHAPTER 8-1

테이블 정의서

유저 / USERS						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
유저 번호	USER_NO	NUMBER	NOT NULL	YES	NO	
유저 아이디	USER_ID	VARCHAR2(20)	NOT NULL	NO	NO	
유저 비밀번호	USER_PW	VARCHAR2(20)	NOT NULL	NO	NO	
유저 닉네임	USER_NICKNAME	VARCHAR2(20)	NOT NULL	NO	NO	
유저 이름	USER_NAME	VARCHAR2(20)	NOT NULL	NO	NO	
핸드폰	USER_PHONE	VARCHAR2(13)	NOT NULL	NO	NO	
이메일	USER_EMAIL	VARCHAR2(30)	NOT NULL	NO	NO	
생일	USER_BIRTH	DATE	NULL	NO	NO	
성별	USER_GENDER	VARCHAR2(1)	NOT NULL	NO	NO	
상태표시	STATUS	VARCHAR2(1)	NOT NULL	NO	NO	
관리자 여부	IS_ADMIN	VARCHAR2(1)	NOT NULL	NO	NO	
가입 날짜	JOIN_DATE	DATE	NOT NULL	NO	NO	
인사말	USER_COMMENT	VARCHAR2(200)	NULL	NO	NO	
크리에이터 등급	USER_CREATER_RANK	VARCHAR2(20)	NOT NULL	NO	NO	
요금제	USER_PLAN	VARCHAR2(50)	NOT NULL	NO	NO	
프로필	USER_PROFILE	VARCHAR2(50)	NULL	NO	NO	

CHAPTER 8-2

테이블 정의서

다운로드 / DOWNLOADS						
Attribute Name	**Column Name**	**Data type**	**Null Option**	**Is PK**	**Is FK**	**Parent table**
다운로드 번호	DOWNLOAD_NO	NUMBER	NOT NULL	YES	NO	
유저 번호	USER_NO	NUMBER	NOT NULL	NO	YES	USER
게시글 번호	CONTENTS_NO	NUMBER	NOT NULL	NO	YES	CONTENT

구독 / SUBSCRIBE						
Attribute Name	**Column Name**	**Data type**	**Null Option**	**Is PK**	**Is FK**	**Parent table**
구독 번호	SUBSCRIBE_NO	NUMBER	NOT NULL	YES	NO	
크리에이터 번호	CREATER_NO	NUMBER	NOT NULL	NO	NO	
유저 번호	USER_NO	NUMBER	NOT NULL	NO	YES	USER

CHAPTER 8-3

테이블 정의서

컨텐츠 게시판 / CONTENT						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
게시글 번호	CONTENT_NO	NUMBER	NOT NULL	YES	NO	
게시글 제목	CONTENT_TITLE	VARCHAR2(50)	NOT NULL	NO	NO	
게시글 내용	CONTENT_DETAIL	CLOB	NOT NULL	NO	NO	
작성일	CREATE_DATE	DATE	NOT NULL	NO	NO	
수정일	MODIFY_DATE	DATE	NOT NULL	NO	NO	
상태표시	CONTENT_STATUS	VARCHAR2(1)	NOT NULL	NO	NO	
완전삭제 표시	DELETE_STATUS	VARCHAR2(1)	NOT NULL	NO	NO	
유저 번호	USER_NO	NUMBER	NOT NULL	NO	YEST	USER

문의 & 의뢰 게시판 / BOARD						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
게시글 번호	CONENT_NO	NUMBER	NOT NULL	YES	NO	
게시판 종류	BOARD_TYPE	NUMBER	NOT NULL	NO	NO	
메뉴 번호	MENU_NO	NUMBER	NOT NULL	NO	YES	MENU

CHAPTER 8-4

테이블 정의서

메뉴 / MENU						
Attribute Name	**Column Name**	**Data type**	**Null Option**	**Is PK**	**Is FK**	**Parent table**
메뉴 번호	MENU_NO	NUMBER	NOT NULL	YES	NO	
메뉴 이름	MENU_NAME	VARCHAR2(20)	NOT NULL	NO	NO	
메뉴 타입	MENU_TYPE	VARCHAR2(1)	NOT NULL	NO	NO	

카테고리 / CATEGORY						
Attribute Name	**Column Name**	**Data type**	**Null Option**	**Is PK**	**Is FK**	**Parent table**
카테고리 번호	CATEGORY_NO	NUMBER	NOT NULL	YES	NO	
카테고리 이름	CATEGORY_NAME	VARCHAR2(100)	NOT NULL	NO	NO	
메뉴 번호	MENU_NO	NUMBER	NOT NULL	NO	YES	MENU

CHAPTER 8-5

테이블 정의서

컨텐츠 별 카테고리 / CONTENT_CATEGORY						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
게시글 번호	CONTENT_NO	NUMBER	NOT NULL	YES	NO	CONTENT
카테고리 번호	CATEGOTY_NO	NUMBER	NOT NULL	YES	NO	CATEGORY

해시태그 / HASHTAG						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
해시태그 번호	HASHTAG_NO	NUMBER	NOT NULL	YES	NO	
해시태그 이름	HASHTAG_NAME	VARCHAR2(100)	NOT NULL	NO	NO	

게시글 해시태그 / CONTENT_HASHTAG						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
게시글 번호	CONTENT_NO	NUMBER	NOT NULL	NO	YES	CONTENT
해시태그 번호	HASHTAG_NO	NUMBER	NOT NULL	NO	YES	HASHTAG

CHAPTER 8-6

테이블 정의서

댓글 / REPLY						
Attribute Name	**Column Name**	**Data type**	**Null Option**	**Is PK**	**Is FK**	**Parent table**
댓글 번호	REPLY_NO	NUMBER	NOT NULL	YES	NO	
작성자	WRITER	VARCHAR2(20)	NOT NULL	NO	NO	
내용	REPLY_DETAIL	VARCHAR2(200)	NOT NULL	NO	NO	
작성일	CREATE_DATE	DATE	NOT NULL	NO	NO	
수정일	MODIFY_DATE	DATE	NOT NULL	NO	NO	
상태표시	REPLY_STATUS	VARCAR2(1)	NOT NULL	NO	NO	
게시글 번호	CONTENT_NO	NUMBER	NOT NULL	NO	YES	CONTENT

좋아요 / LIKES						
Attribute Name	**Column Name**	**Data type**	**Null Option**	**Is PK**	**Is FK**	**Parent table**
유저 번호	USER_NO	NUMBER	NOT NULL	NO	NO	USER
게시글 번호	CONTENT_NO	NUMBER	NOT NULL	NO	NO	CONTENT

CHAPTER 8-7

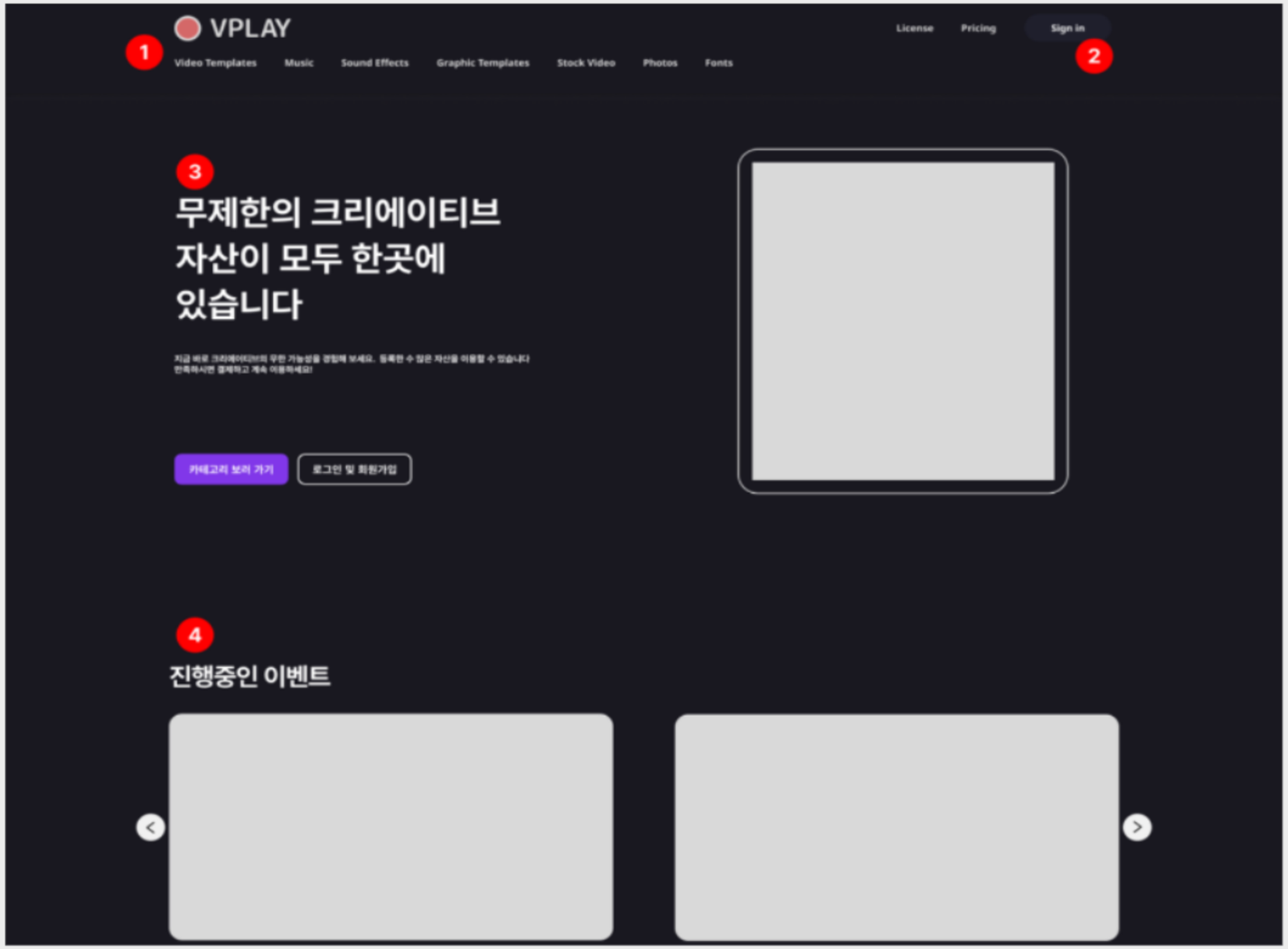
테이블 정의서

결제 / PAYMENT						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
결제 번호	PAYMENT_NO	NUMBER	NOTNULL	YES	NO	
결제 일시	PAYMENT_TIMESTAMP	DATE	NOTNULL	NO	NO	
결제 금액	PAYMENT_AMOUNT	NUMBER	NOTNULL	NO	NO	
결제 상태	PAYMENT_STATUS	VARCHAR2(20)	NOTNULL	NO	NO	
유저 번호	USER_NO	NUMBER	NOTNULL	NO	YES	USER

파일 / FILE						
Attribute Name	Column Name	Data type	Null Option	Is PK	Is FK	Parent table
파일번호	FILE_NO	NUMBER	NOTNULL	YES	NO	
파일 원본 이름	FILE_NAME	VARCHAR2(50)	NOTNULL	NO	NO	
파일 수정 이름	FILE_RENAME	VARCHAR2(50)	NOTNULL	NO	NO	
썸네일	THUMBNAIL	VARCHAR2(50)	Null	NO	NO	
파일 경로	FILE_LOCATION	VARCHAR2(50)	NOTNULL	NO	NO	
상태표시	FILE_STATUS	VARCHAR2(1)	NOTNULL	NO	NO	

CHAPTER 9

스토리보드 - 메인 페이지



1. 로고 : 누르면 홈으로 index로 이동
   카테고리 : 누르면 해당 카테고리로 이동

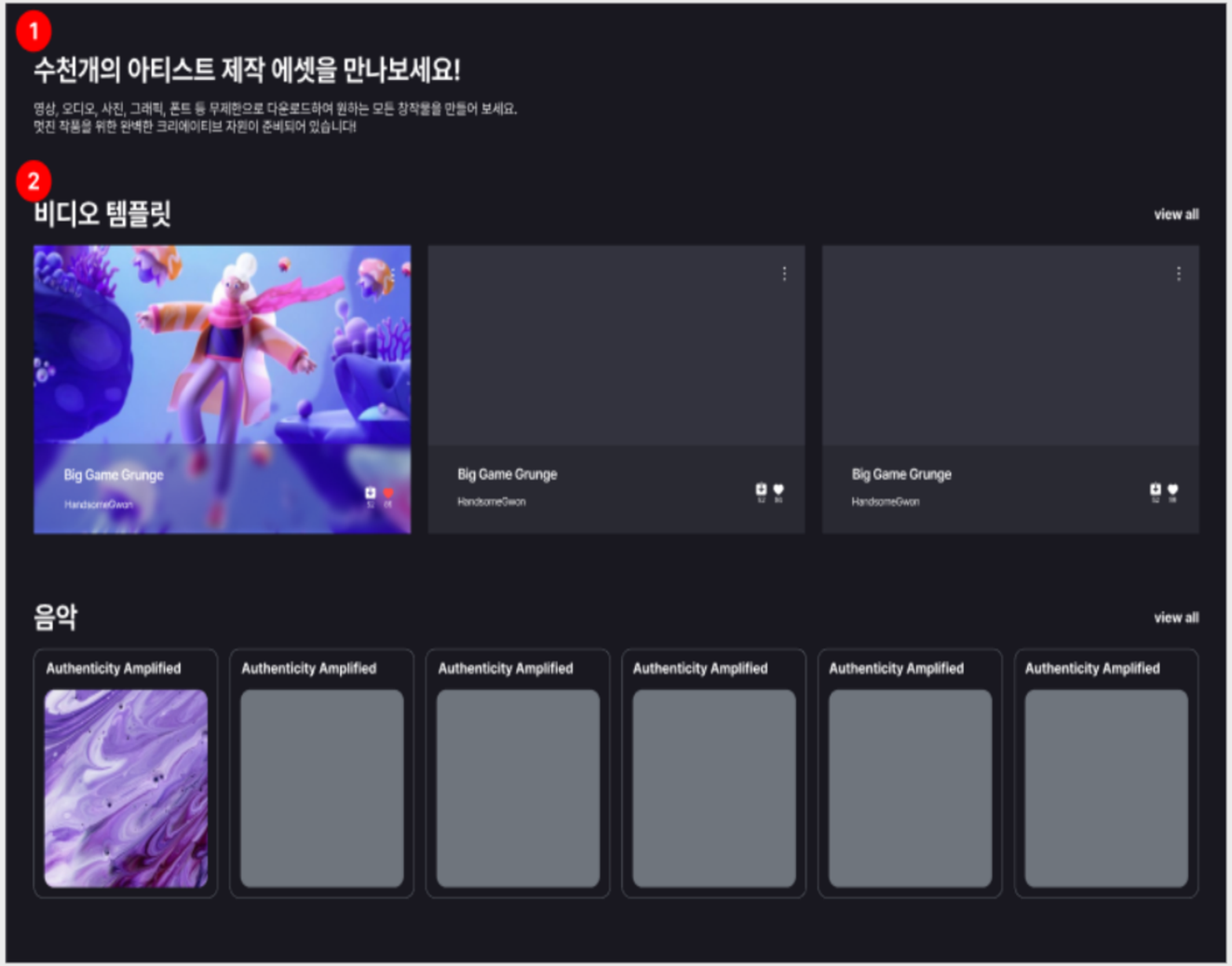
2. 로그인 : 누르면 로그인 화면으로 이동
   Lincense, Prcing : 누르면 해당 화면으로 이동

3. 카테고리 별 간단한 설명
   카테고리 보러 가기 클릭 시 해당 카테고리로 이동
   로그인 및 회원가입 클릭 시 로그인 화면으로 이동

4. 현재 진행 중인 이벤트 카테고리 및 해시태그
   클릭 시 해당 이벤트와 관련된 게시글로 이동

CHAPTER 9

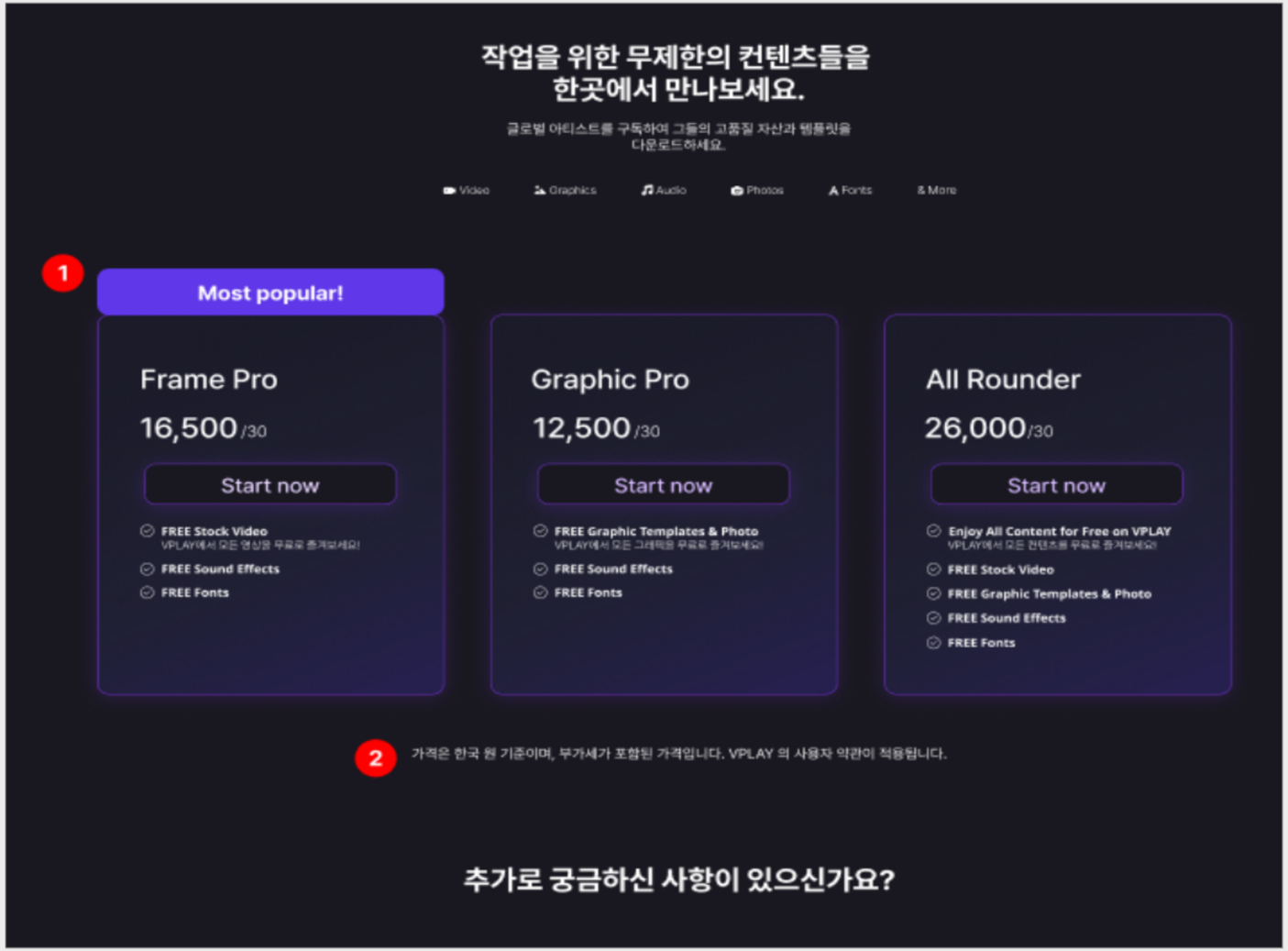
스토리보드 - ALL MENU



1. 전체 카테고리 리스트의 설명이 나온다
2. 각 카테고리 별 리스트들이 나온다

CHAPTER 9

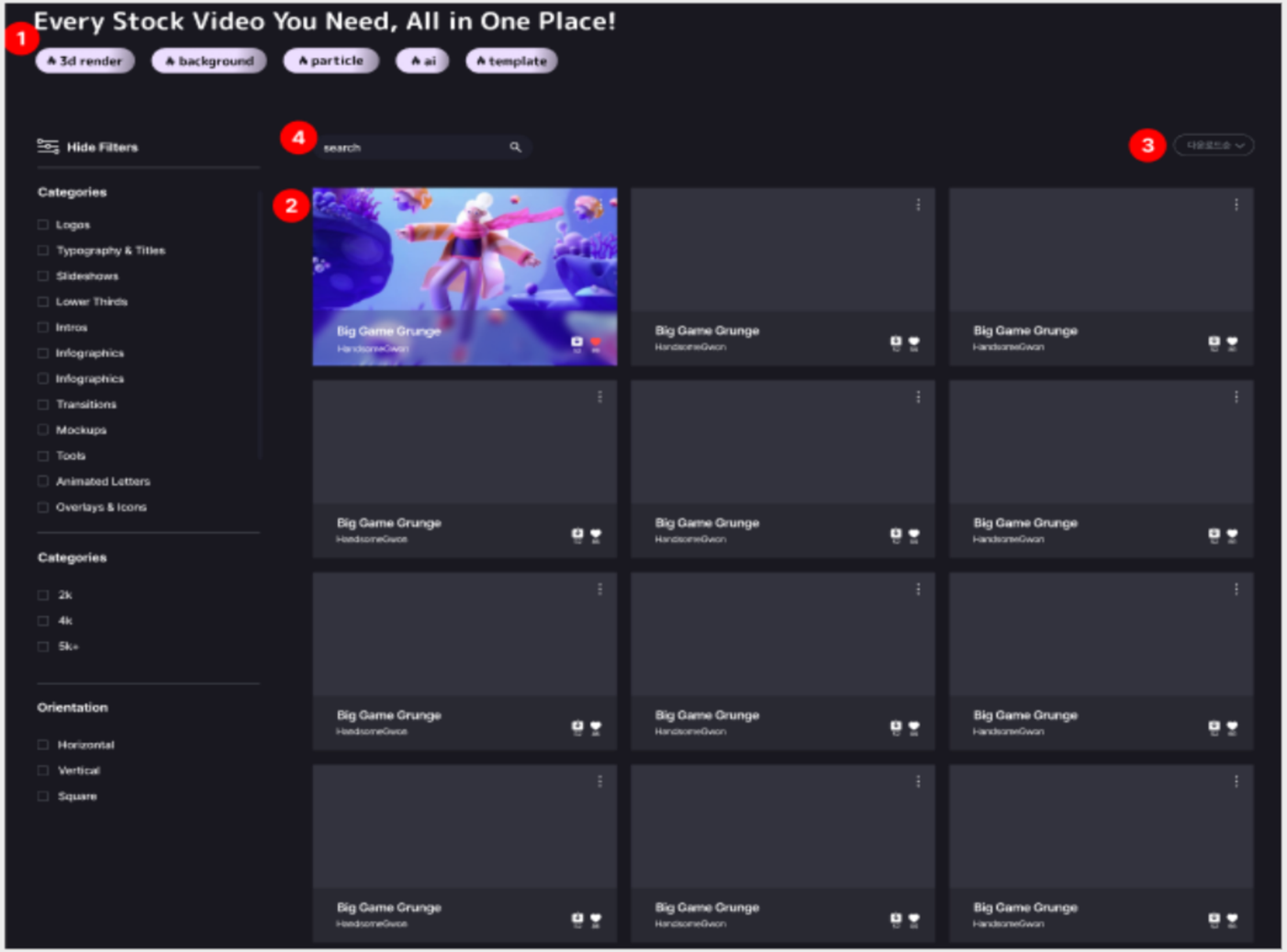
스토리보드 - 결제 페이지



1. 요금제를 고르고 결제를 진행한다.
   1) start now를 누르면 결제창으로 이동한다
2. 결제에 대한 안내문구

CHAPTER 9

스토리보드 - 템플릿 리스트

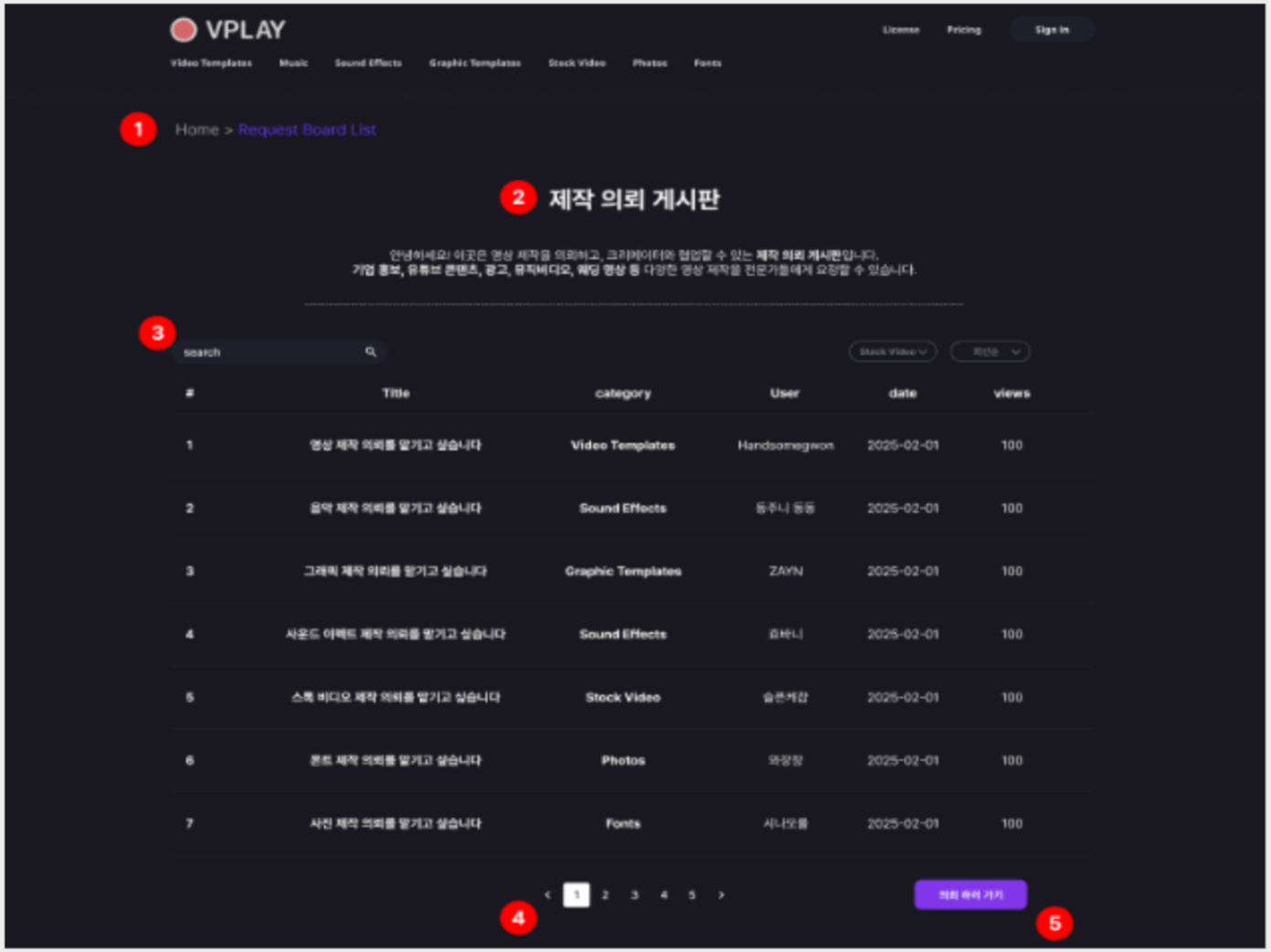


Poco Poco

1. 현재 가장 핫한 태그 리스트 들이다.
   1)클릭 시 선택한 태그가 포함된 항목의
   컨텐츠들이 나열된다.
2. 클릭한 카테고리에 맞춰 컨텐츠들이 리스트로 나온다.
3. 조회순,다운로드순,좋아요 순으로 선택하면
   그에 맞춰서 정렬 가능하다.
4. 검색창으로 검색 하면 해당 검색어에 따른 글들이 나온다.

CHAPTER 9

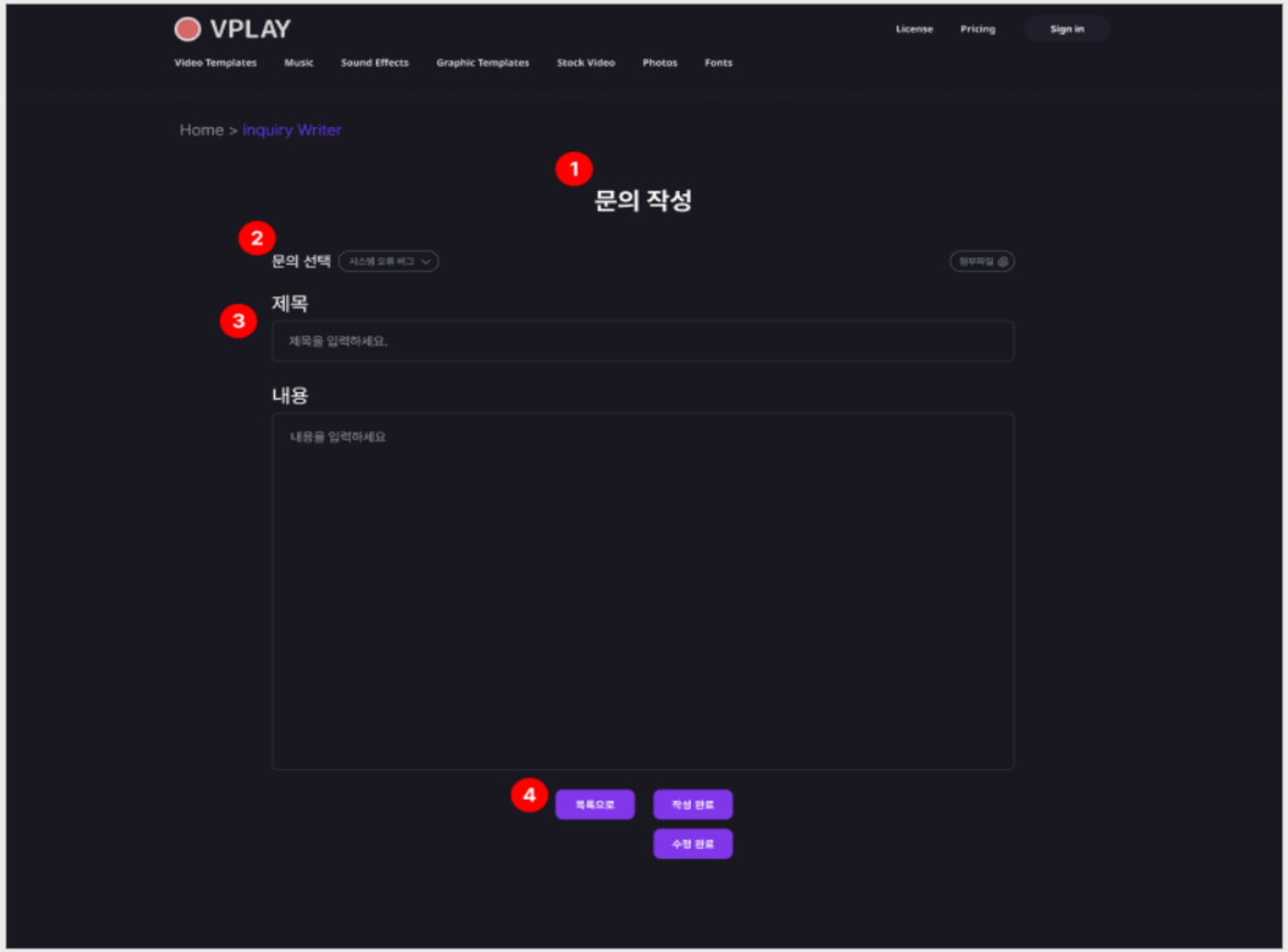
스토리보드 - 제작의뢰 게시판



1. 현재 어느 게시판에 있는 지 명시
2. 제작 의뢰 게시판이라는 걸 대제목으로 명시해주고 밑에 간단한 설명이 적혀져 있다
3. 해당 게시글이 나열 되어 있다
   1) 검색기능으로 검색을 하면 검색어에 해당된 게시글이 보여진다
   2) 게시글의 형식은 글 번호, 제목, 카테고리, 글쓴이, 만든 날짜, 조회수가 적혀져있다
   3) 오른쪽 상단에는 카테고리 별 분류와 최신순, 오래된 순으로 나눠져있다
4. 현재 게시글의 페이징을 할 수 있는 번호
5. 의뢰 하러 가기 버튼으로 클릭 시 의뢰하기 게시글로 이동한다

CHAPTER 9

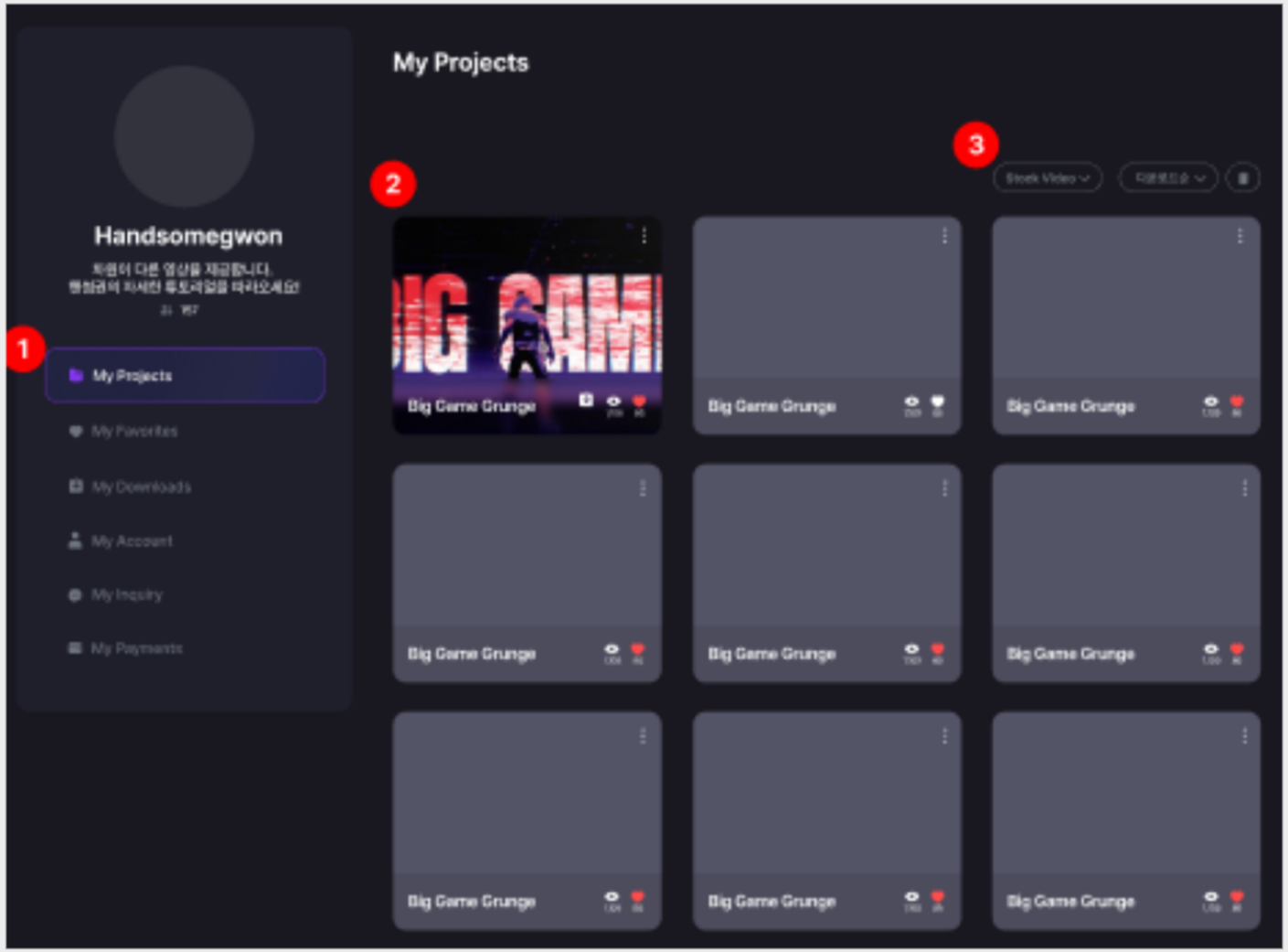
스토리보드 - 문의 게시판



1. 현재 어느 게시판에 있는 지 명시
2. 문의 선택 및 첨부파일
   1) 문의 선택을 클릭 시 해당 카테고리 관련이 나온다
   2) 첨부파일 클릭 시 첨부파일을 올릴 수 있다
3. 제목과 내용을 입력
4. 목록으로, 작성완료, 수정완료 버튼
   1) 목록으로 클릭 시 그 전 리스트로 돌아간다
   2) 작성/수정 완료 클릭 시 해당 게시글 작성/수정

CHAPTER 9

스토리보드 - 마이페이지



1. 프로필과 마이페이지 카테고리
   1)내 사진과 이름, 간단한 자기소개
   2)마이페이지 카테고리 중 하나 선택
2. 내가 올린 프로젝트들이 한 눈에 보인다
   1)썸네일과 제목이 보인다
   2)조회수와 좋아요가 보인다
   3)좋아요 클릭 시 해당 글이 좋아요가 된다
   4)해당 프로젝트 클릭 시 해당 프로젝트 상세페이지로 이동
   5)...클릭 시 삭제하기 버튼이 뜬다
3. 미디어 타입 선택 및 해당 글 상태 선택
4. 휴지통으로 삭제한 글들을 볼 수 있다

CHAPTER 9

스토리보드 - 관리자

1.대시보드버튼을 눌러 대시보드 창을 불러온다.
2. pie그래프를 통해 전체 요금제에 대한 비율을 시각화해서 보여준다.
3. 전체 회원수를 보여준다.
4. 이번달 총 결제 금액을 보여준다.
5. 전체 다운로드 수에 대한 통계를 카테고리별로 막대그래프를 통해 보여준다.
6. 카테고리 별 다운로드 수를 보여준다
7. 트래픽에 대한 자료를 시각화해서 30일간의 데이터를 1일단위로 보여준다

CHAPTER 9

스토리보드 - 관리자

1. 회원이 문의한 내용 보기
   1) 문의 종류와 해당 회원이 올린 첨부파일 확인 버튼이 있다
   2) 제목과 내용이 작성되어 있다
2. 답변 작성하기
3. 작성 버튼으로 답변 작성 후 작성 버튼 클릭 시
   해당 회원한테 답변이 보내진다

CHAPTER 9

코드 리뷰

```
<div class="social-buttons">
    <a href="https://kauth.kakao.com/oauth/authorize?client_id=ffd6b91df4ad805e542c6a8a450195b3&redirect_uri=http://localhost:8080/oauth/kakao&response_type=code&prompt=login" style="width: 50%;"> <img cla
    </a>
    ka href="https://accounts.google.com/o/oauth2/auth?client_id=613310099838-fvnikhprevm9v3po3m8afjbuffjjl2vj.apps.googleusercontent.com&redirect_uri=http://localhost:8080/oauth/google&response_type=code&
```

```
@Controller
@RequestMapping("/oauth")
@RequiredArgsConstructor
public class KakaoAurhController {
    private final UsersService uService;
    private final GoogleOAuthConfig googleOAuthConfig;
    private final String KAKAO_CLIENT_ID = "ffd6b91df4ad805e542c6a8a450195b3";
    private final String KAKAO_REDIRECT_URI = "http://localhost:8080/oauth/kakao";
    private final String KAKAO_TOKEN_URL = "https://kauth.kakao.com/oauth/token";
    private final String KAKAO_USER_INFO_URL = "https://kapi.kakao.com/v2/user/me";



    private final String GOOGLE_REDIRECT_URI = "http://localhost:8080/oauth/google";
    private final String GOOGLE_TOKEN_URL = "https://oauth2.googleapis.com/token";
    private final String GOOGLE_USER_INFO_URL = "https://www.googleapis.com/oauth2/v2/userinfo";
```

CHAPTER 9

코드 리뷰

```java
@GetMapping("/kakao")
public String kakaoLogin(@RequestParam("code") String code, HttpSession session) {
    String accessToken = getAccessToken(code, KAKAO_TOKEN_URL, KAKAO_CLIENT_ID, KAKAO_REDIRECT_URI,null);
    if (accessToken == null) throw new UsersException("카카오 토큰 못 가져옴 ㅋㅋ");

    Map<String, Object> userInfo = getUserInfo(accessToken, KAKAO_USER_INFO_URL);
    if (userInfo == null) throw new UsersException("카카오 사용자 정보 못 가져옴 ㅋㅋ");

    String kakaoId = userInfo.get("id").toString();
    String nickname = userInfo.get("nickname") != null ? userInfo.get("nickname").toString() : "닉네임 없음";

    Users result = uService.existUsers(kakaoId);

    if (result != null) {
        session.setAttribute("loginUser", result);
        return "redirect:/";
    } else {
        Users users = new Users();
        users.setKakaoId(kakaoId);
        users.setUserId(null);
        users.setLoginType("K");
        users.setUserNickname(nickname);
        session.setAttribute("kakaoUser", users);
        return "redirect:/users/signUp";
    }
}
```

CHAPTER 9

코드 리뷰

```java
@GetMapping("/google")
public String googleLogin(@RequestParam("code") String code, HttpSession session) {
     String CLIENT_ID = googleOAuthConfig.getClientId();
     String CLIENT_SECRET = googleOAuthConfig.getClientSecret();
    String accessToken = getAccessToken(code, GOOGLE_TOKEN_URL, CLIENT_ID, GOOGLE_REDIRECT_URI,CLIENT_SECRET);
    if (accessToken == null) throw new UsersException("구글 토큰 못 가져옴 ㅋㅋ");

    Map<String, Object> userInfo = getUserInfo(accessToken, GOOGLE_USER_INFO_URL);
    if (userInfo == null) throw new UsersException("구글 사용자 정보 못 가져옴 ㅋㅋ");

    String googleId = userInfo.get("id").toString();
    String email = userInfo.get("email") != null ? userInfo.get("email").toString() : "이메일 없음";

    Users result = uService.existGoogleUsers(googleId);

    if (result != null) {
        session.setAttribute("loginUser", result);
        return "redirect:/";
    } else {
        Users users = new Users();
        users.setGoogleId(googleId);
        users.setUserId(null);
        users.setLoginType("G");
        users.setUserEmail(email);
        session.setAttribute("googleUser", users);
        return "redirect:/users/signUp";
    }
}
```

CHAPTER 9

코드 리뷰

```java
private String getAccessToken(String code, String tokenUrl, String clientId, String redirectUri,String clientSecret) {
    RestTemplate restTemplate = new RestTemplate();
    HttpHeaders headers = new HttpHeaders();
    headers.setContentType(MediaType.APPLICATION_FORM_URLENCODED);

    MultiValueMap<String, String> params = new LinkedMultiValueMap<>();
    params.add("grant_type", "authorization_code");
    params.add("client_id", clientId);
    params.add("redirect_uri", redirectUri);
    params.add("code", code);

    if (clientSecret != null) {
        params.add("client_secret", clientSecret);
    }


    HttpEntity<MultiValueMap<String, String>> request = new HttpEntity<>(params, headers);
    ResponseEntity<Map> response = restTemplate.exchange(tokenUrl, HttpMethod.POST, request, Map.class);
    Map<String, Object> responseBody = response.getBody();

    return responseBody != null ? (String) responseBody.get("access_token") : null;
}
```

```java
private Map<String, Object> getUserInfo(String accessToken, String userInfoUrl) {
    RestTemplate restTemplate = new RestTemplate();
    HttpHeaders headers = new HttpHeaders();
    headers.set("Authorization", "Bearer " + accessToken);

    HttpEntity<?> request = new HttpEntity<>(headers);
    ResponseEntity<Map> response = restTemplate.exchange(userInfoUrl, HttpMethod.GET, request, Map.class);
    return response.getBody();
}
```

CHAPTER 9

코드 리뷰

```
<div class="date" th:text="${i.createDate}"></div>
<div class="title" th:text="${i.contentTitle}"></div>
<div class="actions">
    <span class="status-badge"
        th:text="${i.reply != null ? '답변 완료' : '답변 대기'}"
        th:classappend="${i.reply != null ? 'completed' : ''}"> </span>
    <div class="options-wrapper">
        <button class="options-btn">⋮</button>
```

CHAPTER 9

코드 리뷰

```
<div class="button-group">
    <button type="button" class="button" onclick="javascript:history.back();">목록으로</button>
    <a th:if="${session.loginUser.userNo == inquiry.userNo
            and (reply == null or #strings.isEmpty(reply.replyDetail))}"
    th:href="@{/board/inquiryUpdate(contentNo=${inquiry.contentNo})}" class="button">수정 하기</a>
</div>
```

```
<div class="form-group">
    <textarea class="reply-textarea"
    th:attr="readonly=${session.loginUser.isAdmin != 'Y' ? 'readonly' : null}"
    th:style="${session.loginUser.isAdmin == 'Y'
        and (reply == null or #strings.isEmpty(reply.replyDetail))} ? 'cursor: text' : 'pointer-events: none; cursor: default'"
    th:text="${reply != null ? reply.replyDetail : ''}"
    th:placeholder="${(reply != null and #strings.isEmpty(reply.replyDetail))
    and (session.loginUser.isAdmin == 'Y') ? '답변 내용을 입력해주세요' : '답변이 아직 등록되지 않았습니다.'}"></textarea>
</div>
<div class="reply-button-group" th:if="${session.loginUser.isAdmin == 'Y' and (reply == null or #strings.isEmpty(reply.replyDetail))}">
    <button type="button" class="button" onclick="submitReply()">답변 등록</button>
</div>
```

CHAPTER 9

코드 리뷰

```
$.ajax({
    url : '/admin/insertReply',
    method : 'post',
    data : {
        contentNo : contentNo,
        replyDetail : replyContent
    },
    success : function(response) {
        if (response == 1) {
            alert('답변이 성공적으로 등록되었습니다');
            location.reload();
        } else {
            alert('답변 등록에 실패했습니다.');
        }
    },
    error : function(xhr, status, error) {
        console.error('오류가 발생했습니다. 다시 시도해주세요.', status, error);
    }
});
```

CHAPTER 9

코드 리뷰

```java
@PostMapping("insertReply")
public String insertReply(@RequestParam(value = "page", defaultValue = "1") int currentPage, @ModelAttribute Reply reply,
        HttpSession session, Model model, @RequestParam("contentNo") String contentNo) {
    Users loginUser = (Users) session.getAttribute("loginUser");
    if (loginUser != null) {
        reply.setWriter(loginUser.getUserNickname());
        reply.setContentNo(contentNo);
    }
    int result = bService.insertReply(reply);
    if (result > 0) {
        ArrayList<Reply> replyList = bService.selectReplyList(Integer.parseInt(contentNo));
        model.addAttribute("replyList", replyList);
        return "redirect:/board/" + contentNo + "/" + currentPage;

    } else {
        throw new BoardException("댓글 등록에 실패하였습니다.");
    }

}
```

CHAPTER 9

코드 리뷰

```
<div class="search-bar">
    <select class="filter-dropdown" name="menuName" id="menuName">
        <option value="All">All</option>
        <option value="video-Templates" th:selected="${menuName == 'video-Templates'}">video Templates</option>
        <option value="Graphic-Templates" th:selected="${menuName == 'Graphic-Templates'}">Graphic Templates</option>
        <option value="Stock-Video" th:selected="${menuName == 'Stock-Video'}">Stock Video</option>
        <option value="Photos" th:selected="${menuName == 'Photos'}">Photos</option>
        <option value="Music" th:selected="${menuName == 'Music'}">Music</option>
        <option value="Sound-Effects" th:selected="${menuName == 'Sound-Effects'}">Sound Effects</option>
        <option value="Fonts" th:selected="${menuName == 'Fonts'}">Fonts</option>
    </select>
    <form>
        <input type="search" name="search" placeholder="search" th:value="${searchValue}">
    </form>
    <button class="create-button" th:if="${session.loginUser}" onclick="location.href = '/board/writeRequest'">의뢰 하러 가기</button>
</div>
```

CHAPTER 9

코드 리뷰

```
<tbody>
    <tr class="request-list" th:each="c : ${list}">
        <td>[[${c.contentNo}]]</td>
        <td>[[${c.contentTitle}]]</td>
        <td>[[${c.menuName}]]</td>
        <td>[[${c.userNickName}]]</td>
        <td>[[${c.modifyDate}]]</td>
        <td>[[${c.views}]]</td>
    </tr>
    <tr th:if="${#lists.isEmpty(list)}">
        <td colspan="6" style="...">검색 결과가 없습니다.</td>
    </tr>
</tbody>
```

CHAPTER 9

코드 리뷰

```
<div class="pagination">
    <ul>
        <!-- 현재 페이지가 2 이상이면 이전 페이지로 이동하는 링크 보여줌 -->
        <th:block th:if="${pi.currentPage > 1}">
            <li class="prev chevron-left">
                <a th:href="@{${Loc}(page=${pi.currentPage-1}, menuName=${menuName}, search=${searchValue})}" aria-label="Previous"> < </a>
            </li>
        </th:block>
        <!-- 현재 페이지가 1 이하이면 비활성화된 이전 버튼을 보여줌 -->
        <th:block th:if="${pi.currentPage <=1}">
            <li class="prev chevron-left">
                <a aria-label="Previous"> < </a><!-- href 없음: 비활성화 상태 -->
            </li>
        </th:block>
        <!-- 숫자 페이지 버튼 영역 -->
        <!-- pi.startPage부터 pi.endPage까지 반복해서 페이지 번호 링크 생성 -->
        <li th:each="p : ${#numbers.sequence(pi.startPage, pi.endPage)}">
            <a th:href="@{${Loc}(page=${p}, menuName=${menuName}, search=${searchValue})}" th:text="${p}" th:class="${p == pi.currentPage ? 'active' : ''}"> </a>
        </li><!-- 현재 페이지에는 'active' 클래스 부여됨 -->
```

CHAPTER 9

코드 리뷰

```
            <!-- 다음 페이지 버튼 (비활성/활성 분기) -->
            <!-- 현재 페이지가 최대 페이지보다 작을 때: 다음 페이지 링크 활성화 -->
            <th:block th:if="${pi.currentPage < pi.maxPage}">
                <li class="next chevron-right">
                    <a th:href="@{${Loc}(page=${pi.currentPage+1}, menuName=${menuName}, search=${searchValue})}" aria-label="Next"> > </a>
                </li>
            </th:block>
            <!-- 마지막 페이지인 경우: 다음 버튼 비활성 -->
            <th:block th:if="${pi.currentPage >= pi.maxPage}">
                <li class="next chevron-right">
                    <a aria-label="Next"> > </a>
                </li>
            </th:block>
        </ul>
    </div>
</div>
```

CHAPTER 9

코드 리뷰

```
document.getElementById("menuName").addEventListener('change', function() {
    const menuName = this.value;
    console.log(menuName);
    if(menuName == "All"){
        location.href= "/board/request_list";
    }else{
        location.href= "/board/request_list/" + menuName;
    }
});
```

CHAPTER 9

코드 리뷰

```java
@GetMapping("request_list")  효바니 +3
public ModelAndView joinrequestPost(@RequestParam(value = "page", defaultValue = "1") int currentPage,
                                    @RequestParam(value = "search", required = false) String search,
                                    ModelAndView mv) {
    HashMap<String, String> map = new HashMap<>();
    map.put("search", search);
    int listCount = bService.getrequestPostCount(map);
    PageInfo pi = Pagination.getPageInfo(currentPage, listCount, boardLimit: 10);
    ArrayList<Content> list = bService.selectAllRequestPost(map, pi);
    for (Content c : list) {
        c.setUserNickName(bService.selectUser(c.getUserNo()));
    }
    mv.addObject( attributeName: "list", list)
            .addObject( attributeName: "pi", pi)
            .addObject( attributeName: "searchValue", search);
    mv.setViewName("request_list");
    return mv;
}
```

CHAPTER 9

코드 리뷰

```java
@GetMapping("request_list/{menuName}")  GRAM₩dltkd +2 *
public String filterRequestList(@RequestParam(value = "search", required = false) String search,
                                @RequestParam(value = "page", defaultValue = "1") int currentPage,
                                @PathVariable("menuName") String menuName, Model model) {
    HashMap<String, String> map = new HashMap<>();
    map.put("search", search);
    switch (menuName) {
    case "video-Templates":
        map.put("menuNo", "1");break;
    case "Graphic-Templates":
        map.put("menuNo", "4");break;
    case "Stock-Video":
        map.put("menuNo", "5");break;
    case "Photos":
        map.put("menuNo", "6");break;
    case "Music":
        map.put("menuNo", "2");break;
    case "Sound-Effects":
        map.put("menuNo", "3");break;
    case "Fonts":
        map.put("menuNo", "7");break;
    }
    int listCount = bService.getrequestPostCount(map);
    PageInfo pi = Pagination.getPageInfo(currentPage, listCount, boardLimit: 10);
    ArrayList<Content> list = bService.selectAllRequestPost(map, pi);
    for (Content c : list) {
        c.setUserNickName(bService.selectUser(c.getUserNo()));
    }
    model.addAttribute( attributeName: "list", list)
            .addAttribute( attributeName: "menuName", menuName)
            .addAttribute( attributeName: "Loc",  attributeValue: "/board/request_list/" + menuName)
            .addAttribute( attributeName: "pi", pi).addAttribute( attributeName: "searchValue", search);
    return "request_list";
}
```

CHAPTER 9

코드 리뷰

```
<select id="selectAllRequestPost">
    select *
    from content
    join board using(content_no)
    join menu using(menu_no)
    where board_type = 2 and content_status = 'Y'
    <if test="menuNo != null">
        and menu_no = #{menuNo}
    </if>
    <if test="search != null and search != ''">
        and (content_title like '%${search}%' or content_detail like '%${search}%')
    </if>
    order by content_no desc
</select>
```

```
<select id="selectUser">
    select user_nickname
    from users
    where user_no = #{userNo}
</select>
```

CHAPTER 9

코드 리뷰

```
//버킷용
implementation 'software.amazon.awssdk:s3:2.17.68'
```

```
cloudflare.r2.public-url=https://pub-91253393bf234b79b30a739e5ba17bae.r2.dev
aws.s3.bucket-name=poco-vplay-bucket
aws.credentials.access-key=f849f0f671f9bd48e94c41e2e07eef72
aws.credentials.secret-key=e0f8dafd340fddb125f3d8c9b50583ec3a00a8b0da3c5b6600ddeecb72f4490f
aws.s3.endpoint=https://d23533183859af70aa3bc72fd8e26aa8.r2.cloudflarestorage.com
```

CHAPTER 9

코드 리뷰

```
public R2Service(
        @Value("${aws.credentials.access-key}") String accessKey,
        @Value("${aws.credentials.secret-key}") String secretKey,
        @Value("${aws.s3.bucket-name}") String bucketName,
        @Value("${cloudflare.r2.public-url}") String publicUrl,
        @Value("${aws.s3.endpoint}") String endpoint) {
```

CHAPTER 9

코드 리뷰

```
public String uploadFile(MultipartFile file) throws IOException {
    String fileName = UUID.randomUUID() + "_" + file.getOriginalFilename();

    s3Client.putObject(PutObjectRequest.builder()
                    .bucket(bucketName)
                    .key(fileName)
                    .build(),
            RequestBody.fromBytes(file.getBytes()));

      return publicUrl + "/" + fileName;

}
```

CHAPTER 9

코드 리뷰

```
public boolean deleteFile(String fileName) {
    try {
        DeleteObjectRequest deleteRequest = DeleteObjectRequest.builder()
                .bucket(bucketName)
                .key(fileName)
                .build();

        s3Client.deleteObject(deleteRequest);
        System.out.println("✅ 파일 삭제 성공: " + fileName);
        return true;
    } catch (SdkException e) {
        System.err.println("❌ 파일 삭제 실패: " + fileName);
        e.printStackTrace();
        return false;
    }
}
```

CHAPTER 9

코드 리뷰

```java
public InputStream downloadFileStream(String fileName) {
    GetObjectRequest getObjectRequest = GetObjectRequest.builder()
            .bucket(bucketName)
            .key(fileName)
            .build();

    try {
        InputStream inputStream = s3Client.getObject(getObjectRequest);
        System.out.println("✅ 파일 스트리밍 시작: " + fileName);
        return inputStream;

    } catch (SdkException e) {
        System.err.println("❌ 파일 스트리밍 실패: " + fileName);
        e.printStackTrace();
        return null;
    }
}
```

CHAPTER 9

코드 리뷰

```java
@GetMapping("download/{fileName}/{contentNo}/{userNo}")
public ResponseEntity<StreamingResponseBody> downloadFile(@PathVariable("fileName") String fileNa

    System.out.println(fileName);

    InputStream fileStream = r2Service.downloadFileStream(fileName);

    if(fileStream == null) {
        throw new BoardException("파일 다운로드 중 오류 발생");
    }

    int count = bService.checkDownload(contentNo, userNo);
    if(count == 0) {
        int result = bService.downloadRecord(contentNo, userNo);
    }

    StreamingResponseBody responseBody = outputStream -> {
        byte[] buffer = new byte[8192];
        int bytesRead;
        while ((bytesRead = fileStream.read(buffer)) != -1) {
            outputStream.write(buffer, 0, bytesRead);
        }
        fileStream.close();
    };

    return ResponseEntity.ok()
            .header(HttpHeaders.CONTENT_DISPOSITION, "attachment; filename=\"" + fileName + "\"")
            .contentType(MediaType.APPLICATION_OCTET_STREAM)
            .body(responseBody);
}
```

CHAPTER 10

시연