



구독서비스 추천, 비교 웹서비스(LET ME SUB)

완 료 보 고 서

2021. 10. 31

가천대학교
컴퓨터공학과

팀 전문성

팀장 ■ 조영우(201736042)
팀원 ■ 유민상(201739426)
팀원 ■ 안주현(201939623)

문서 정보

본 문서는 LET ME SUB 개발에 대한 완료 보고서입니다.

버 전	1.0
작성일	2021.10.31
상 태	<input checked="" type="checkbox"/> 완료 <input type="checkbox"/> 진행 중 <input type="checkbox"/> 초안

버전	변경한 사람	변경한 날짜	버전업 변경(또는 추가)내용
0.1	유민상	2021.09.26.	프로젝트 개요 작성
0.2	안주현	2021.09.26.	서비스 개요작성
0.3	조영우	2021.09.28.	요구사항정의 작성
0.4	안주현	2021.10.25.	화면(UI 설계) 작성, 데이터 베이스 설계 작성
0.5	유민상	2021.10.27.	소스디렉터리 구조
0.55	유민상	2021.10.27.	시스템 아키텍처 기능설계 작성
0.6	안주현	2021.10.27.	DB수정
0.7	조영우	2021.10.27.	UML다이어그램 작성
0.75	조영우	2021.10.28.	기능설계 추가
0.8	유민상	2021.10.28.	시스템 아키텍처 , 기능설계 , 개발환경 추가
0.85	안주현	2021.10.28	서비스 개요 수정, 메뉴 구조도 설명추가
0.9	유민상	2021.10.31	서비스 개요 ,기능설계 추가 , 운영환경 수정
1.0	조영우	2021.10.31	서비스 기능, 요구사항 수정, 추천알고리즘 기술

목 차

1. 프로젝트 개요		
1.1 개발목표와 범위	4
1.2 개발일정/산출물	9
1.3 개발조직/역할	9
2. 서비스기능 정의		
2.1 서비스 개요	10
2.2 요구기능 정의	12
3. 프로세스(기능) 설계		
3.1 시스템 아키텍처	18
3.2 기능 설계(Sequence Diagram)	19
3.3 추천 알고리즘 기능설계	29
3.4 UML 다이어그램 설계	33
4. 화면(UI)설계		
4.1 스토리보드 설계	40
4.2 UI 설계	41
5. 데이터베이스 설계		
5.1 데이터 베이스 구조	50
5.2 논리적 DB설계 (테이블명세서)	51
5.3 물리적 DB설계 (SQL스크립트)	52
6. 환경구성		
6.1 개발환경	53
6.2 운영환경	53
6.3 소스 디렉터리 구조	54

1, 프로젝트 개요

1.1 개발목표와 범위

배경 및 필요성



[그림 1.1.1 국내 구독 경제 시장 규모]

1) 구독 경제의 엄청난 성장

구독 경제는 일정 기간마다 비용을 지불하고 물건이나 서비스를 받는 개념입니다. 구독 경제는 수십 년 전부터 우유나 신문을 구독하던 시대부터 존재했던 오래된 개념입니다. 하지만 최근 대두된 구독경제는 일상 전반을 바꾸어 놨다는 점에서 차이점이 있습니다. IT 기술의 발전, 배송 시스템의 고도화, 결제방식의 혁신 등의 힘입어 한층 더 진화했고 코로나 19로 비대면 거래와 서비스가 확대되면서 구독경제는 엄청난 속도로 성장을 하고 있습니다.



[그림 1.1.2 많은 구독 서비스들]

2) 구독 서비스 비교 추천 플랫폼의 부재

현재 국내외를 비롯한 수많은 기업과 플랫폼에서 구독 서비스를 운영하고 있습니다. 항상 이런 구독 서비스들을 이용할 때마다 구독 서비스들을 검색하고 비교 및 추천을 해주는 플랫폼이 있었으면 편리하고 유용할 거라고 생각했습니다. 그러나 현재 국내에는 구독 서비스를 검색, 비교, 추천 해주는 기능을 하는 서비스가 없어서 이러한 서비스들을 저희가 만들어 보려고 합니다.

개발 목표

1) 구독 서비스 카테고리화 조회 기능

- 구독 서비스들을 카테고리별로 분류한 후 카테고리별로 구분해 조회하는 기능

2) 구독 서비스 추천 기능

- 구독서비스 추천 알고리즘을 개발해 사용자에게 적합한 구독 서비스를 추천해주는 기능

3) 구독 서비스 비교 기능

- 비슷한 구독서비스들 끼리 비교해 각각 장단점, 사용자들의 평점 등을 보여주는 기능

4) 사용자의 구독 서비스 평가 기능

- 사용자가 구독서비스를 평가하고 평가한 내용을 조회하는 기능

개발 범위



[그림 1.1.3 개발 범위]

프론트엔드

1) 회원가입

- 회원가입 화면 UI/UX 구현
- 회원가입 폼 작성 시 서버에 AJAX 통신으로 회원 데이터 전송

2) 로그인

- 로그인 화면 UI/UX 구현
- 로그인 폼 작성 후 서버에 AJAX 통신으로 로그인 정보 전송
- 서버에서의 Response 값 확인 후 로그인 성공시 브라우저에 로그인 쿠키정보를 생성

3) 인덱스

- 인덱스 화면 UI/UX 구현
- 서버에 구독서비스 정보를 카테고리 별로 요청한 후 Response 값을 이용해 카테고리 별로 인덱스 화면에 구독 서비스 정보를 보여줌
- 추천 기능으로 이동 하는 버튼 구현
- 비교 버튼을 누르고 구독 서비스 2개를 선택 후 비교 서비스로 선택한 구독 서비스 정보와 함께 이동하는 기능

4) 상세 페이지

- 상세 페이지 UI / UX 구현
- 인덱스 화면이나 추천 기능 에서 받아온 구독 서비스 정보를 이용해 서버에 해당 구독 서비스의 상세 정보를 요청한다.
- 받아온 정보를 이용해 구독 서비스 사진, 설명, 평점, 리뷰를 작성 할 수 있는 기능을 제공

5) 비교 페이지

- 비교페이지 UI
- 받아온 두 구독 서비스 정보를 서버에 전송해, 해당 구독 서비스에 대한 상세정보를 서버에 요청하여 가져온 후 비교 페이지 생성

6) 추천 기능

- 추천 기능 UI
- 구독 추천 알고리즘에 정보를 입력하기 위해 사용자에게 질문
- 질문한 사용자의 답변 정보를 서버에 보내고 알고리즘을 이용해 추천된 서비스를 보여준다.

백엔드

- 1) SpringBoot를 사용해 웹 서버 구현
- 2) Tomcat 9.0 사용
- 3) controller를 이용해 html 렌더링
- 4) 클라이언트와 통신해 필요한 로직 처리
- 5) Aws Ec2를 사용해서 배포
- 6) AWS RDS의 MySql을 사용 MyBatis 라이브러리를 사용해 DB와 연결

1.2 개발일정/산출물(마일스톤)

		1주	2주	3주	4주	5주	6주	7주	8주
분석 설계	제안서	작성							
	설계서			작성					
구현 시험	프로그래밍			프론트엔드					
						백엔드			
정리 완료	완료발표-1차								
	산출물 정리								

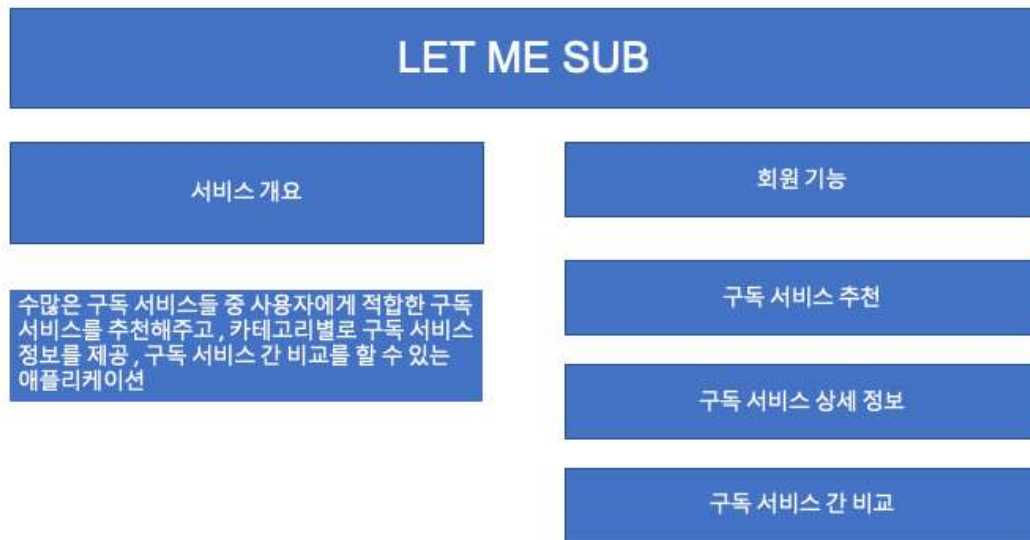
		9주	10주	11주	12주
분석 설계	설계서				
구현 시험	프로그래밍	프론트엔드			
		백엔드		배포	
	테스트		TEST		
정리 완료	완료발표-1차			발표준비	
	산출물 정리			서류보완	

1.3 개발조직/역할

역할	이름	역할(개발기능)
팀장	조영우	<ul style="list-style-type: none"> - 프로젝트 관리(일정, 성과) - 데이터베이스 설계 - 웹 개발 - 유지 및 보수
팀원	유민상	<ul style="list-style-type: none"> - 서버 구축 - 웹 개발 - 유지 및 보수
팀원	안주현	<ul style="list-style-type: none"> - 웹 개발 - 유지 및 보수 - 프로젝트 문서화

2. 서비스기능 정의

2.1 서비스 개요



[그림 2.1.1 LET ME SUB 서비스 개요]

국내의 수많은 구독 서비스들이 존재하고 있고, 다양한 서비스를 제공하고 있습니다. LET ME SUB은 구독 서비스 추천을 위해 사용자에게 다양한 질문을 하고 사용자가 원하는 가장 적합한 구독 서비스를 추천해 줍니다. 또한 카테고리별로 구독 서비스를 정리해 사용자가 원하는 분야의 구독 서비스들의 정보를 리스트로 제공하고 상세한 정보와 평점을 보여줍니다. 마지막으로 사용자가 원하는 두 구독 서비스를 비교하는 페이지를 제공하고 다른 사용자들은 비교하는 두 구독 서비스 중 어떤 걸 더 선호하는지 알려줍니다.

1) 회원기능

- 사용자는 제공하는 서비스를 사용하기 위해서 회원가입과 로그인을 한다.

2) 구독 서비스 상세 조회

- 사용자가 메인화면에 출력된 구독 서비스를 클릭하면 상세페이지로 이동한다.
- 사용자는 상세 페이지에서 해당 구독 서비스의 설명, 리뷰를 볼 수 있고, 리뷰를 작성할 수 있다.
- 서버는 사용자의 요청에 따라 데이터베이스와 통신하여 작성된 리뷰를 보여주고 사용자에게 의해 작성된 리뷰는 데이터베이스에 입력된다.
- 사용자가 메인화면에서 추천하기 버튼을 클릭하면 추천페이지로 이동한다.

3) 구독 서비스 추천

- 사용자는 추천페이지에서 여러 요소의 질문에 따라 개인의 선호도를 선택하고 적합한 구독 서비스를 추천받는다.
- 서버는 사용자가 선택한 선호도를 가중치 알고리즘에 의해 계산하여 가장 적합한 구독 서비스를 추천 해준다.

4) 구독 서비스 비교

- 사용자가 메인화면에서 나열된 구독 서비스 중 두 가지를 선택하고 비교하기 버튼을 클릭하면 비교하기 페이지로 넘어간다.
- 사용자는 비교 페이지에서 각 구독 서비스의 설명과 UI를 볼 수 있고, 두 구독 서비스의 6가지 요소 중 강점을 육각형 형태로 볼 수 있다.
- 사용자가 비교 페이지에서 둘 중 하나를 선택하도록 해서 데이터베이스에 저장하고 선택한 결과를 반영하여, 유저 선호도를 볼 수 있다.

2.2 요구기능 정의

프론트엔드

번호	주요기능명	상세기능명	개발담당
1	사용자 인증 기능	회원가입	유민상
2		로그인	유민상
3	서비스 추천 기능	사용자 선호도 입력	조영우
4		추천 서비스 출력	조영우
5		추천 서비스 화면 구성	조영우
6	서비스간 UI 비교 기능	화면 넘기기	조영우
7		육각형 별점 비교	조영우
8		사용자 선호 서비스 입력	조영우
9		전체 사용자 선호 서비스 출력	조영우
10		비교 서비스 화면 구성	조영우
11	상세정보 보기 기능	육각형 별점 출력	안주현
12		리뷰 보기	안주현
13		리뷰 작성	안주현
14		상세정보 화면 구성	안주현

사용자 인증 기능

1) 회원가입

- 해당 서비스에 가입한 적이 없다면 회원가입을 해야 한다.
- 사용자는 생성할 계정의 아이디, 비밀번호, 이름, 이메일을 입력해야 한다.

2) 로그인

- 가입된 계정이 있으면 아이디와 비밀번호를 입력하고 로그인을 한다.
- 네이버 아이디로 로그인할 수 있다.
- 해당 서비스에 가입한 적이 없다면 회원가입을 해야 한다.
- 사용자는 생성할 계정의 아이디, 비밀번호, 이름, 이메일을 입력해야 한다.

서비스 추천기능

1) 사용자 선호도 입력

- 사용자의 선호도를 조사한다.
- 예를 들어 ‘한국 영화를 좋아하시나요?’ 라고 물어보고 1~7까지 버튼으로 질문의 선호도를 받는다.

2) 추천 서비스 출력

- 사용자의 성향을 분석하여 적절한 서비스를 추천한다.
- 추천 서비스 화면 구성
- 가독성이 좋고, 깔끔한 UI를 구성해야 한다.

서비스 비교기능

1) 화면 넘기기

-UI의 화면을 넘길 수 있는 좌, 우 화살표를 출력하여 사용자가 클릭 시 화면을 전환할 수 있게 구성한다.

2) 육각형 도형 비교

-기능별로 육각형의 각 꼭짓점을 구성하여, 더 좋은 기능일수록 짝 찬 육각형인 도형을 출력하여, 두 서비스간 기능비교를 쉽게 한다.

3) 사용자 선호 서비스 입력

-사용자는 비교한 두 서비스 중 더 선호하는 서비스를 선택할 수 있다.

-선택한 후에는 '97%(더 선호하는 비율)의 사용자가 넷플릭스(서비스명)를 더 선호합니다!' 라고 출력된다.

4) 비교 서비스 화면 구성

-가독성이 좋고, 깔끔한 UI를 구성해야 한다.

상세정보 보기 기능

1) 육각형 별점 출력

-기능별로 육각형의 각 꼭짓점을 구성하여, 더 좋은 기능일수록 짝 찬 육각형인 도형을 볼 수 있다.

2) 리뷰 보기

-사용자가 작성한 리뷰를 볼 수 있다.

3) 리뷰 작성

-로그인을 한 사용자는 서비스에 대한 리뷰를 작성할 수 있다.

4) 상세정보 화면 구성

-가독성이 좋고, 깔끔한 UI를 구성해야 한다.

백엔드

번호	주요기능명	상세기능명	개발담당
1	사용자 인증 기능	회원가입	유민상
2		로그인	유민상
3	회원 정보 관리	회원 정보 데이터 입력	유민상
4		회원 정보 데이터 조회	유민상
5		회원 정보 데이터 수정	유민상
6		회원 정보 데이터 삭제	유민상
7	서비스 정보 관리	서비스 정보 데이터 입력	안주현
8		서비스 정보 데이터 조회	안주현
9		서비스 정보 데이터 수정	안주현
10		서비스 정보 데이터 삭제	안주현
11	리뷰 정보 관리	리뷰 정보 데이터 입력	조영우
12		리뷰 정보 데이터 조회	조영우
13		리뷰 정보 데이터 수정	조영우
14		리뷰 정보 데이터 삭제	조영우

사용자 인증 기능

1) 회원가입

- 해당 서비스에 가입한 적이 없다면 회원가입을 해야 한다.
- 사용자는 생성할 계정의 아이디, 비밀번호, 이름, 이메일을 입력해야 한다.
- 서버는 사용자의 정보를 저장하고, 비밀번호는 암호화하여 저장해야한다.

2) 로그인

- 가입된 계정이 있으면 아이디와 비밀번호를 입력하고 로그인을 한다.
- 네이버 아이디로 로그인할 수 있다.
- 해당 서비스에 가입한 적이 없다면 회원가입을 해야 한다.
- 서버는 사용자가 입력한 정보를 토대로 로그인을 수행해야한다.

회원 정보 관리

1) 회원 정보 데이터 입력

서버는 데이터베이스에서 사용자의 정보를 입력할 수 있다.

2) 회원 정보 데이터 조회

서버는 데이터베이스에서 원하는 사용자의 정보를 조회할 수 있다.
사용자의 이메일 조회와 같은 기능을 수행할 수 있다.

3) 회원 정보 데이터 수정

서버는 데이터베이스에서 원하는 사용자의 정보를 수정할 수 있다.
사용자 전화번호 수정과 같은 기능을 수행할 수 있다.

4) 회원 정보 데이터 삭제

서버는 데이터베이스에서 원하는 사용자의 정보를 삭제할 수 있다.
사용자 탈퇴와 같은 기능을 수행할 수 있다.

서비스 정보 관리

1) 서비스 정보 데이터 입력

서버는 데이터베이스에서 원하는 서비스의 정보를 입력할 수 있다.

2) 서비스 정보 데이터 조회

서버는 데이터베이스에서 원하는 서비스의 정보를 조회할 수 있다.
'카테고리 = ott' 와 같은 정보를 조회할 수 있다.

3) 서비스 정보 데이터 수정

서버는 데이터베이스에서 원하는 서비스의 정보를 수정할 수 있다.
바뀐 서비스에 대해서 서비스를 수정할 수 있다.

4) 서비스 정보 데이터 삭제

서버는 데이터베이스에서 원하는 서비스의 정보를 삭제할 수 있다.
만료된 서비스에 대해서 데이터를 삭제할 수 있다.

리뷰 정보 관리

1) 리뷰 정보 데이터 입력

서버는 데이터베이스에서 리뷰를 입력할 수 있다.

2) 리뷰 정보 데이터 조회

서버는 데이터베이스에서 원하는 리뷰를 조회할 수 있다.
'좋음'이라는 단어가 들어간 리뷰를 조회할 수 있다.

3) 리뷰 정보 데이터 수정

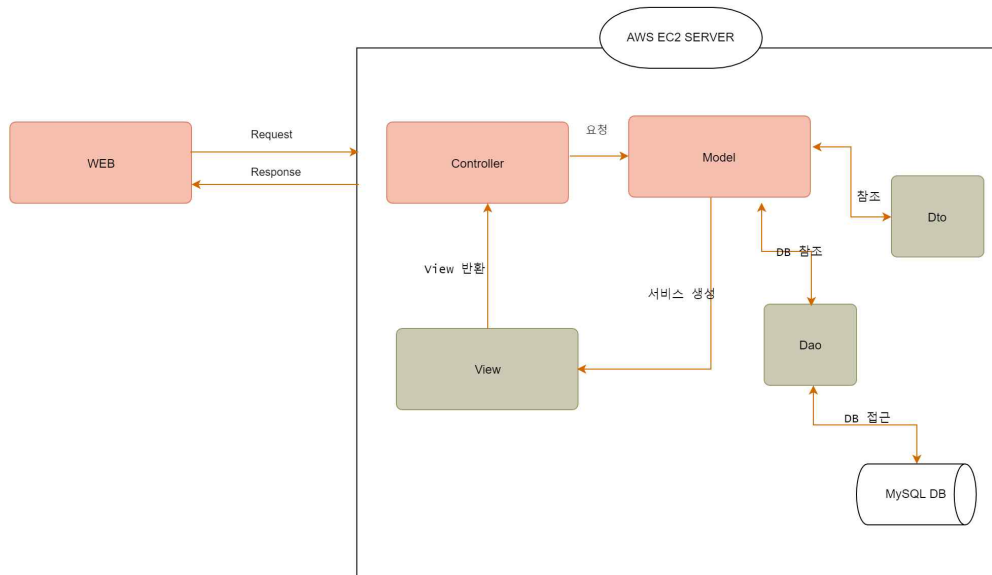
서버는 데이터베이스에서 원하는 리뷰를 수정할 수 있다.
사용자의 개인정보가 포함된 리뷰를 수정할 수 있다.

4) 리뷰 정보 데이터 삭제

서버는 데이터베이스에서 원하는 리뷰를 삭제할 수 있다.
욕설이 들어간 리뷰를 삭제할 수 있다.

3. 프로세스(기능) 설계

3.1 시스템 아키텍처



[그림 3.1.1 시스템 아키텍처]

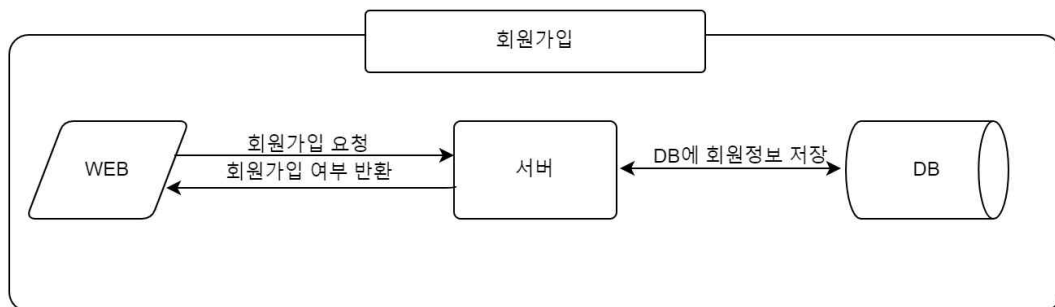
사용 기술들

- 1) **JAVA** : Python의 Flask 나 JavaScript의 Node.js 같은 기술들이 비교되었습니다. Java는 불편한 점이 많지만 취업시장 에서 가장 많이 요구되는 언어이고 JVM에 대한 경험을 무시할 수 없기 때문에 Java를 선택하게 되었습니다.
- 2) **Spring Boot** : JVM 기반 널리 쓰는 웹 프레임워크이면 이미 지원해주는 기능이 많기 때문에 보일러 플레이트 코드 작성을 줄이고 서비스 개발에 집중을 할 수 있었습니다.
- 3) **MySQL** : NoSQL 같은 기술 사용도 검토되었지만, 전통적인 관계형 데이터 베이스를 이용해서 서비스를 구현하고 싶어서 MySQL을 사용하였습니다.
- 4) **BootStrap** : 디자인이 좋고 반응형 웹을 지원하는 서비스를 구현하기 위해서 자체제작은 어렵다고 생각되어 BootStrap템플릿을 활용해 프론트엔드를 구현했습니다.
- 5) **Jquery** : React 나 Angular 같은 최신 프론트엔드 기술 도입을 검토했습니다. 그러나 조원중에서 해당 기술을 가진 사람이 없어 공부를 하면서 도입을 해야 했고 스프링부트 프레임워크도 처음부터 공부를 하고있는 상황이라 다소 Legacy한 Jquery를 사용 했습니다.
- 6) **AWS** : 웹 서비스 배포를 위해 가장 유명한 클라우드 플랫폼 AWS 기술을 활용해서 배포를 했습니다. 저렴하게 도메인(letmesub.shop) 을 구해서 EC2를 이용해 웹서비스 배포를 하였습니다.

3.2 기능 설계

회원 기능

1) 회원가입

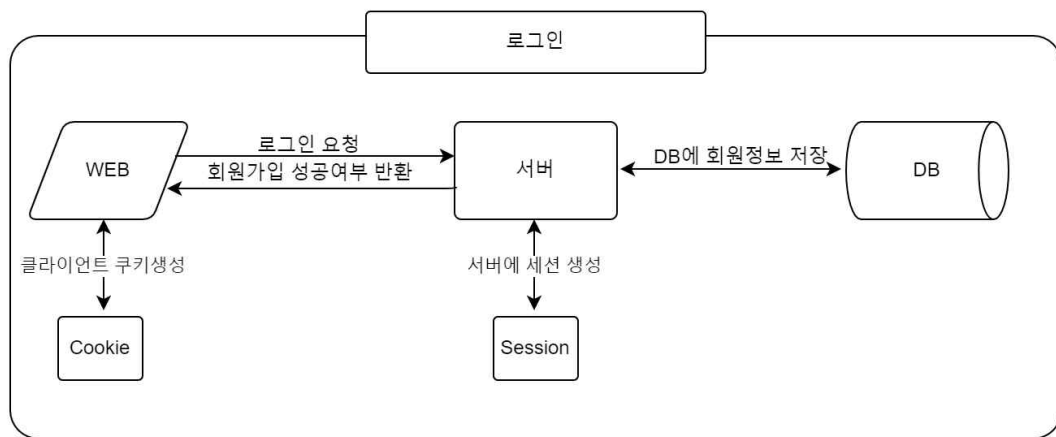


[그림 3.2.1 회원가입 기능 설계]

웹에서 회원가입 정보를 입력 후 회원가입 버튼을 누를 시 아이디, 비밀번호, 이메일 데이터를 Ajax 통신을 이용해서 서버로 전송한다.

서버에서는 받아온 데이터의 값을 회원 테이블에 적합한 값인지 검증한 후 DB에 저장한다. 저장이 성공한 경우 다시 클라이언트로 성공 여부를 반환한다.

2) 로그인

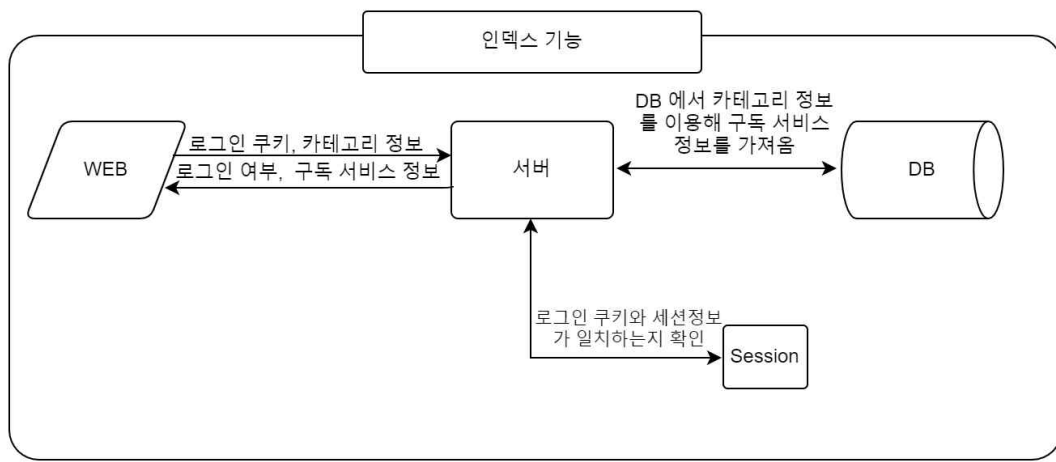


[그림 3.2.2 로그인 기능 설계]

웹에서 로그인 정보를 입력 후 로그인 버튼을 누를시, 아이디와 비밀번호 정보를 Ajax 통신을 통해 서버로 전송한다. 서버에서 DB회원 테이블의 정보를 접근해 해당 아이디와 비밀번호가 존재하는지 검증한다. 만약 존재한다면 서버에 회원의 세션을 만들고 클라이언트로 로그인 성공여부를 반환한다. 클라이언트가 로그인 성공값을 받으면 브라우저에 로그인 쿠키를 생성한다.

구독 서비스 카테고리별로 조회

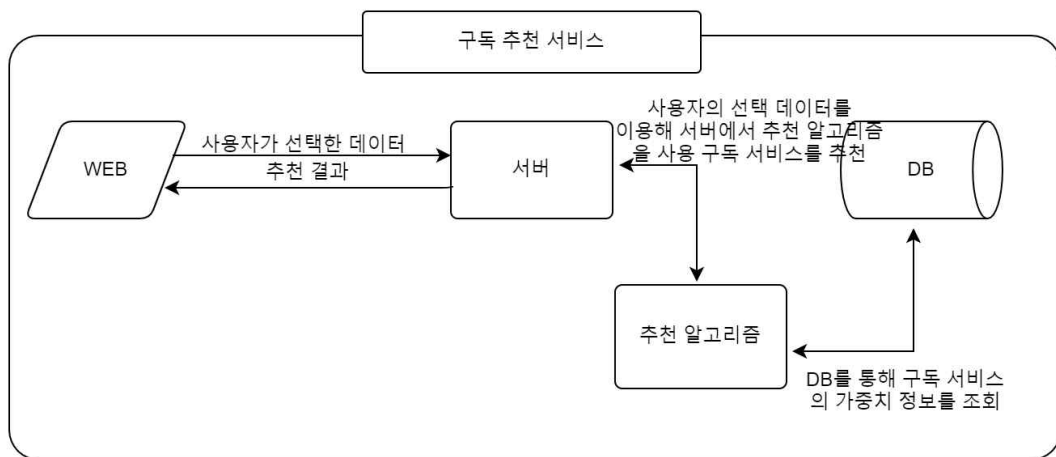
1) 인덱스 기능



[그림 3.2.3 인덱스 기능 설계]

인덱스에서는 로그인 여부 확인과 현재 조회하기 원하는 카테고리 정보 데이터를 서버에 Ajax 통신을 이용해 request 한다. 서버에서는 세션 과의 로그인 여부를 확인하고 클라이언트로 로그인 여부 정보를 다시 response해 인덱스 화면의 로그인 여부를 확인한다. 그리고 카테고리 정보를 이용해 DB로 접근해 요청한 카테고리에 해당하는 구독 서비스들의 데이터를 Dto 에 담고 Json 형식으로 변환한 후 다시 브라우저로 response한다. 클라이언트에서는 받아온 json 정보를 이용해 구독 서비스들을 조회할 수 있게 한다.

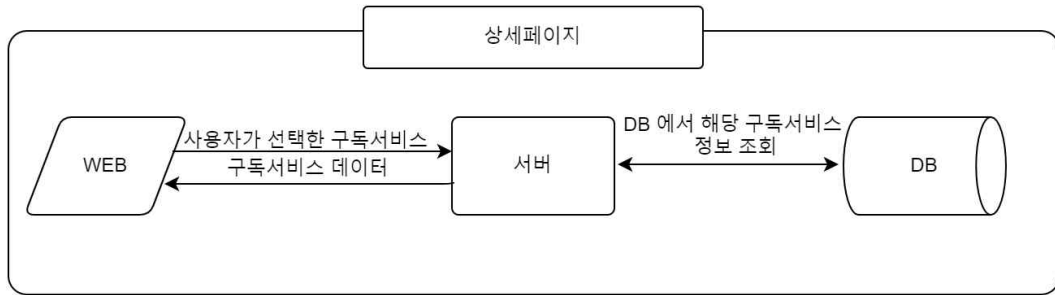
구독 서비스 추천 기능



[그림 3.2.4 구독 추천 서비스 설계]

인덱스에서 원하는 카테고리를 선택한 후, 추천페이지로 가는 링크를 클릭한다. 서버에는 GET방식으로 카테고리의 정보가 전송되고, 이에 맞추어 각 카테고리에 부합하는 적절한 질문이 사용자에게 출력된다. 사용자는 여러 번의 질문에 1~7까지의 가중치를 선택해서 서버에 전송하고, 마지막 질문이 끝나면 서버는 추천 알고리즘에 의해 사용자에게 가장 적절한 서비스를 출력한다. 이때 서버에서 브라우저로 렌더링하는 방식은 타임리프를 사용한다.

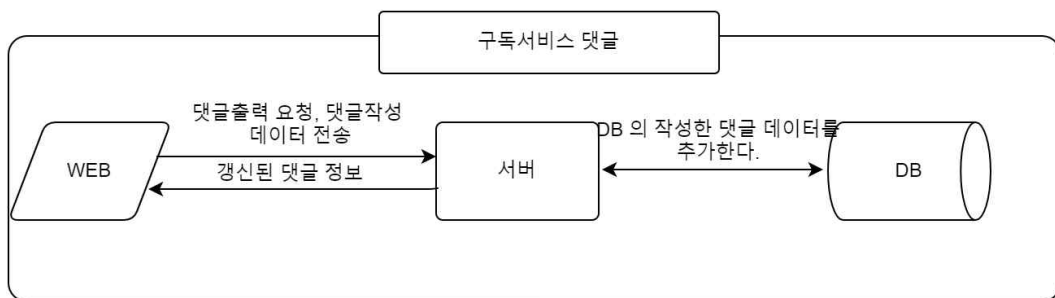
구독 서비스 상세 페이지



[그림 3.2.5 상세 페이지 설계]

인덱스 페이지에서 출력된 구독 서비스를 클릭시 서버로 클릭한 구독 서비스 이름 정보를 GET방식으로 서버로 보낸다. 서버에서 받은 구독 서비스의 이름을 이용하여 이미지, 상세설명 등을 데이터베이스에서 DTO에 담아 가져와서 브라우저에 렌더링한다. 이때 서버에서 브라우저로 렌더링하는 방식은 타임리프를 사용한다.

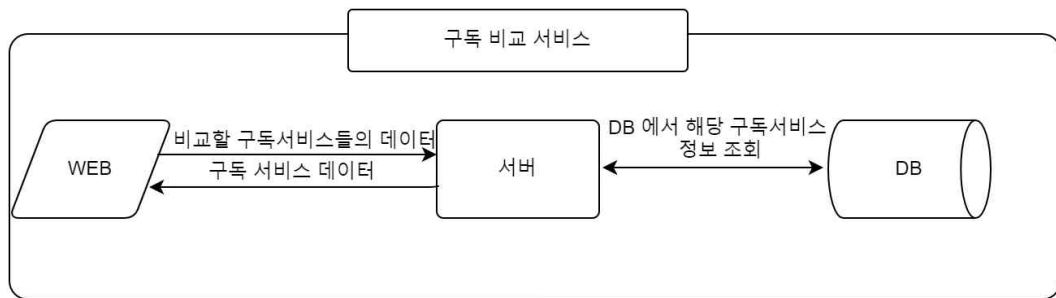
구독 서비스 댓글



[그림 3.2.6 구독서비스 댓글 설계]

인덱스에서 구독 서비스를 클릭해 상세 페이지로 이동시 서버로 해당 구독 서비스의 댓글 정보 출력을 요청한다. DB에서 구독 서비스에 해당하는 댓글 정보를 가져와 출력을 한다. 댓글을 입력하고 댓글입력 버튼을 누를시 댓글작성에 필요한 아이디 댓글 내용 등의 데이터를 서버에 전송한다. 서버에서는 해당 데이터를 DB의 댓글 테이블에 추가를 한 후 갱신된 댓글 정보를 다시 클라이언트로 보내준다.

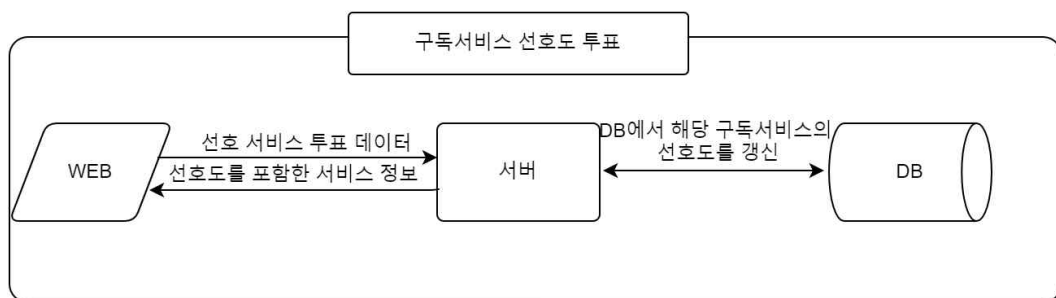
구독 서비스 비교 페이지



[그림 3.2.7 구독 비교 서비스 설계]

인덱스에서 비교하기 버튼을 누르면 비교하고 싶은 구독 서비스들을 선택 할 수 있는 체크박스가 생긴다. 2개를 선택해 비교하기를 누르면 서버로 비교를 할 구독 서비스 이름 두 개를 GET방식으로 보낸다. 서버에서 DB에 접속해 비교할 구독 서비스 정보를 조회하고 서비스 두 개의 UI, 그리고 자체적으로 평가한 평가도를 별점형태로 사용자에게 제공하여 보기 쉽게 비교 할 수 있게 한다. 이때 서버에서 브라우저로 렌더링하는 방식은 타임리프를 사용한다.

구독 서비스 선호도 투표



[그림 3.2.8 구독서비스 선호도 투표 설계]

비교하기 버튼을 통해 이동한 페이지에서, 하단에 선호하는 서비스를 누르면, DB에 내가 선호하는 서비스의 선호도가 올라가는 입력을 하고, DB에서 비교하고 있는 서비스의 전체 사용자 선호도를 가져와서, 선호율을 사용자에게 보여준다. 이때 서버에서 브라우저로 렌더링하는 방식은 타임리프를 사용한다.

AWS를 사용한 배포

1)

Name	인스턴스 ID	인스턴스 유형	가용 영역	인스턴스 상태	상태 검사	경보 상태	퍼블릭 DNS(IPv4)	IPv4 퍼블릭
	i-0c02e2e2ed16727ba	t2.micro	ap-northeast-2a	running	2/2개 검사 ...	없음	ec2-3-36-50-229.ap-no...	3.36.50.22

[그림 3.2.9 AWS EC2 인스턴스 생성]

웹 서버를 사용하기 위해 AWS EC2의 프리티어를 이용해 EC2를 생성

2)

DB 식별자	역할	엔진	리전 및 AZ	크기	상태
minsang	인스턴스	MySQL Community	ap-northeast-2b	db.t2.micro	사용 가능

[그림 3.2.10 AWS RDS를 사용해 데이터베이스 생성]

데이터베이스를 사용하기 위해 AWS RDS를 사용해 MySQL 데이터베이스 인스턴스를 생성

3)

```
ubuntu@ip-172-31-4-247: ~
ubuntu@ip-172-31-4-247:~$ git clone https://github.com/LetMeSub/LetMeSub.git
```

[그림 3.2.11 EC2 인스턴스에 git에 올려둔 프로젝트를 clone한다.]

4)

```
ubuntu@ip-172-31-4-247: ~/LetMeSub
ubuntu@ip-172-31-4-247:~$ cd LetMeSub/
ubuntu@ip-172-31-4-247:~/LetMeSub$ ls
LICENSE build build.gradle gradle gradlew gradlew.bat settings.gradle src
ubuntu@ip-172-31-4-247:~/LetMeSub$ ./gradlew build
```

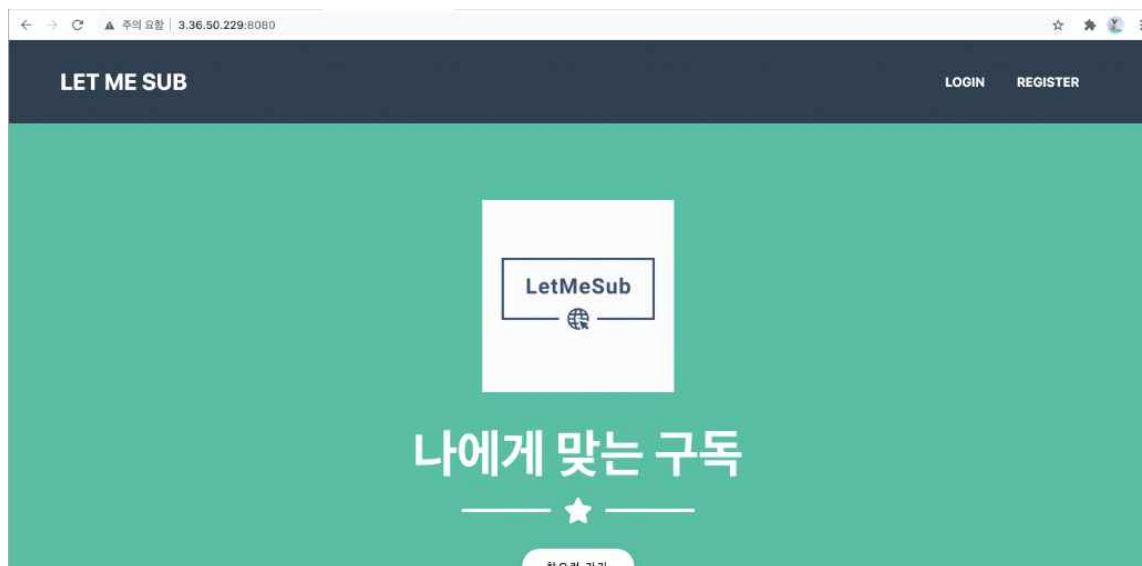
[그림 3.2.12 가져온 프로젝트를 gradle을 이용해 빌드를 진행한다.]

5)

[illegible]

[그림 3.2.13 EC2에서 빌드한 프로젝트를 java를 이용해 실행한다.]

6)



[그림 3.2.14 EC2의 Public IP를 이용해 웹 서비스가 잘 작동되는지 확인한다.]

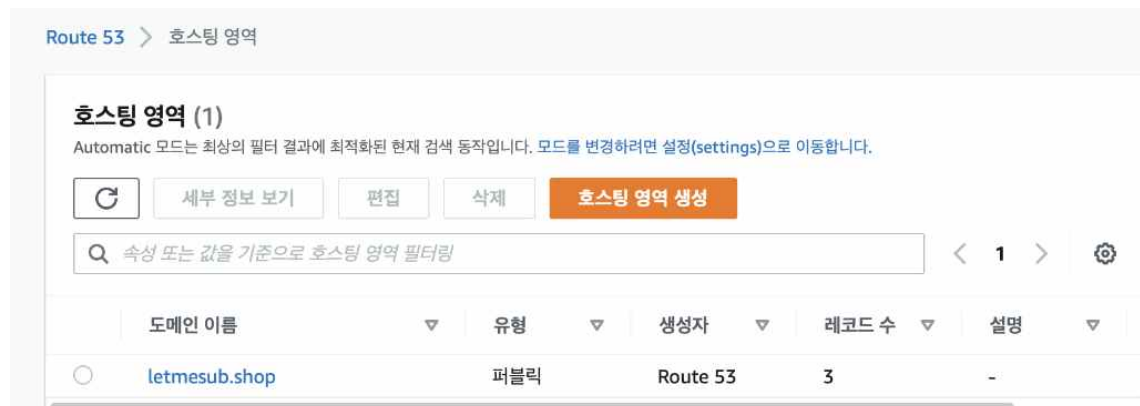
7)

DNS 설정

DNS 설정	도메인 연결	포워딩	웹 파킹	모바일 파킹
<input type="checkbox"/>	도메인 명 ^			
<input checked="" type="checkbox"/>	letmesub.shop	🔍	A	

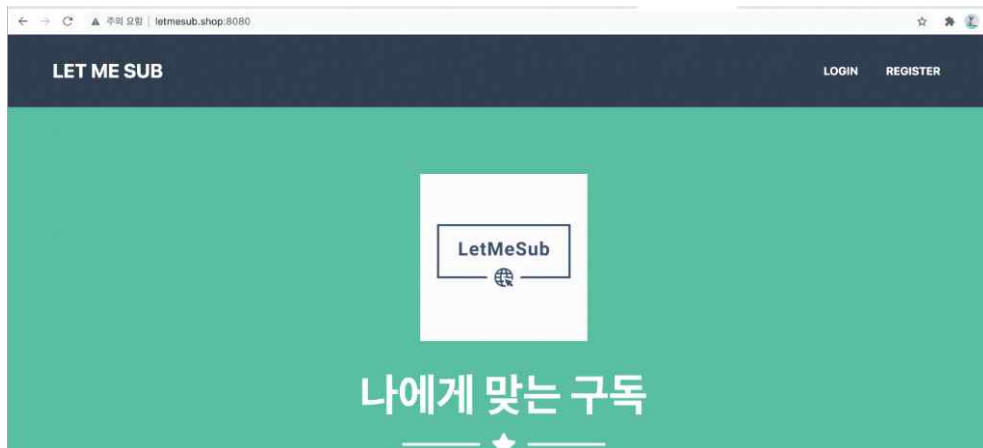
[그림 3.2.15 DNS를 사용하기 위해 가비아에서 도메인을 구입한다.]

8)



[그림 3.2.16 Route 53을 이용해 도메인을 호스팅하고 네임서버를 연결함]

9)



[그림 3.2.17 DNS(letmesub.shop)를 이용해 웹 서비스 접속]

10)

```
ubuntu@ip-172-31-4-247:~/LetMeSub/build/libs$ nohup java -jar LetMeSub-0.0.1-SNAPSHOT.jar
nohup: ignoring input and appending output to 'nohup.out'
>>ubuntu@ip-172-31-4-247:~/LetMeSub/build/libs$ >>
```

[그림 3.2.18 nohup 명령어를 사용해 서버 프로그램이 백그라운드로 돌아가게 실행함]

3.3 추천 알고리즘 기능설계

1. 공통 질문을 정한다. 이때 대답은 둘 중 하나를 선택하게 한다.

1) 사람들이 평가하는 나는?

-따뜻한 사람

-유능한 사람

2) 자유시간이 있을 때 나는?

-혼자 아무지게 놀아야지~

-누구랑 놀까?

3) 일을 할 때 나는?

-한땀한땀,,, 꼼꼼히,,,

-유연하게,,, 자유롭게,,,

4) 나와 의견이 다를 때 나는?

-나의 옳음을 증명해야겠어!!

-상처받지 않게 조심,, 조심,,

5) 내방이 정리되지 않은 상태?

-스트레스 받는다,, 치운다...!

-이런 거에 스트레스 따위 받지 않아!

6) 학교에서 상을 받은 당신, 무슨상?

-토론왕 김XX

-봉사왕 이XX

2. 카테고리별 가중치를 정할 질문을 정함

ex) ott 서비스

1. 한국영화를 좋아하시나요?
2. 영화 마니아이신가요?
3. 마블영화는 어떤가요?
4. 애니메이션을 선호 하시나요?
5. 다큐멘터리는 어떤가요?
6. 예능은 어떤가요?

3. 구독서비스별로 각 카테고리에 맞는 카테고리별 질문에 대한 가중치를 평가 (단, 가중치는 1~7사이의 값으로 한다)

ex) 'ott'카테고리의 'netflix'서비스

1) 한국영화를 좋아하시나요?

1번 질문의 가중치 : 5

2) 영화 마니아이신가요?

1번 질문의 가중치 : 6

3) 마블영화는 어떤가요?

1번 질문의 가중치 : 3

4) 애니메이션을 선호 하시나요?

1번 질문의 가중치 : 5

5) 다큐멘터리는 어떤가요?

1번 질문의 가중치 : 4

6) 예능은 어떤가요?

1번 질문의 가중치 : 4

4. 카테고리별로 공통질문의 답변에 따라 부여할, 카테고리별 질문의 가중치를 부여한다.
(공통질문의 답변당 카테고리별 질문 두 개에 '2'씩 가중치를 부여한다.)

ex) 공통질문의 답변으로 가중치를 부여할 'ott'카테고리의 번호

1) 사람들이 평가하는 나는?

-따뜻한 사람

카테고리별 질문 1, 5번에 '2'씩 가중치 합산

-유능한 사람

카테고리별 질문 3, 6번에 '2'씩 가중치 합산

2) 자유시간이 있을 때 나는?

-혼자 아무지게 놀아야지~

카테고리별 질문 2, 6번에 '2'씩 가중치 합산

-누구랑 놀까?

카테고리별 질문 1, 3번에 '2'씩 가중치 합산

3) 일을 할 때 나는?

-한땀한땀... 꼼꼼히...

카테고리별 질문 2, 5번에 '2'씩 가중치 합산

-유연하게... 자유롭게...

카테고리별 질문 1, 4번에 '2'씩 가중치 합산

(이하 생략)

5. 공통질문의 답변으로 인해 부여된 가중치 + 유저가 선택한 가중치를 계산한다.

예를 들어 공통질문으로 인해서 부여된 카테고리별 가중치를 합산했을 때, 각 질문별 가중치가 4,4,2,5,2,5이다. 그리고 카테고리별 질문에서 유저가 선택한 각 질문별 가중치가 4,2,1,5,6,3이다. 그리고 같은 질문끼리 가중치를 합산하면 8,6,3,10,8,8이다.

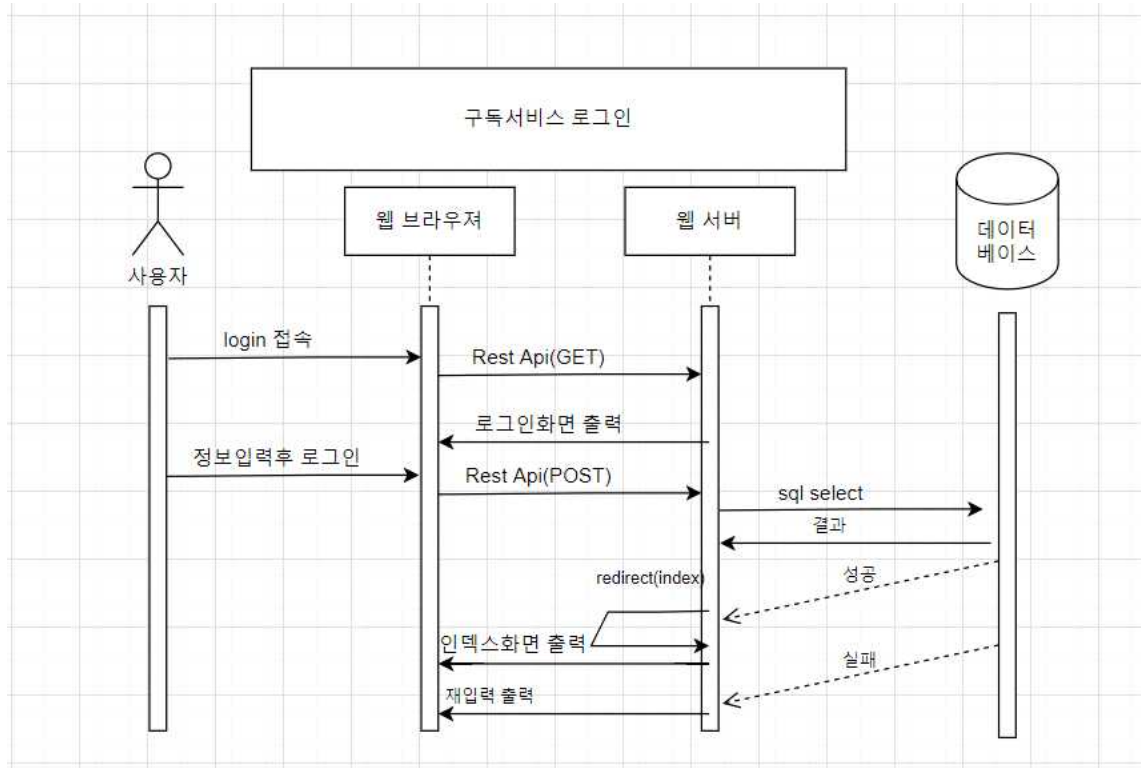
최종 가중치를 가지고, 선택한 카테고리내의 서비스의 가중치와 연산해서, 비교하면 유저에게 추천할 서비스를 정할 수 있다.

만약 'netflix'서비스의 카테고리별 질문의 평가점수가 1,1,1,1,1,6이고, 'watcha'의 카테고리별 질문의 평가점수가 1,1,6,1,1,1 이다. 그러면 같은 열에 있는 점수끼리 곱하고, 곱한후 나온 숫자들을 더한 점수를 최종 점수로 한다.

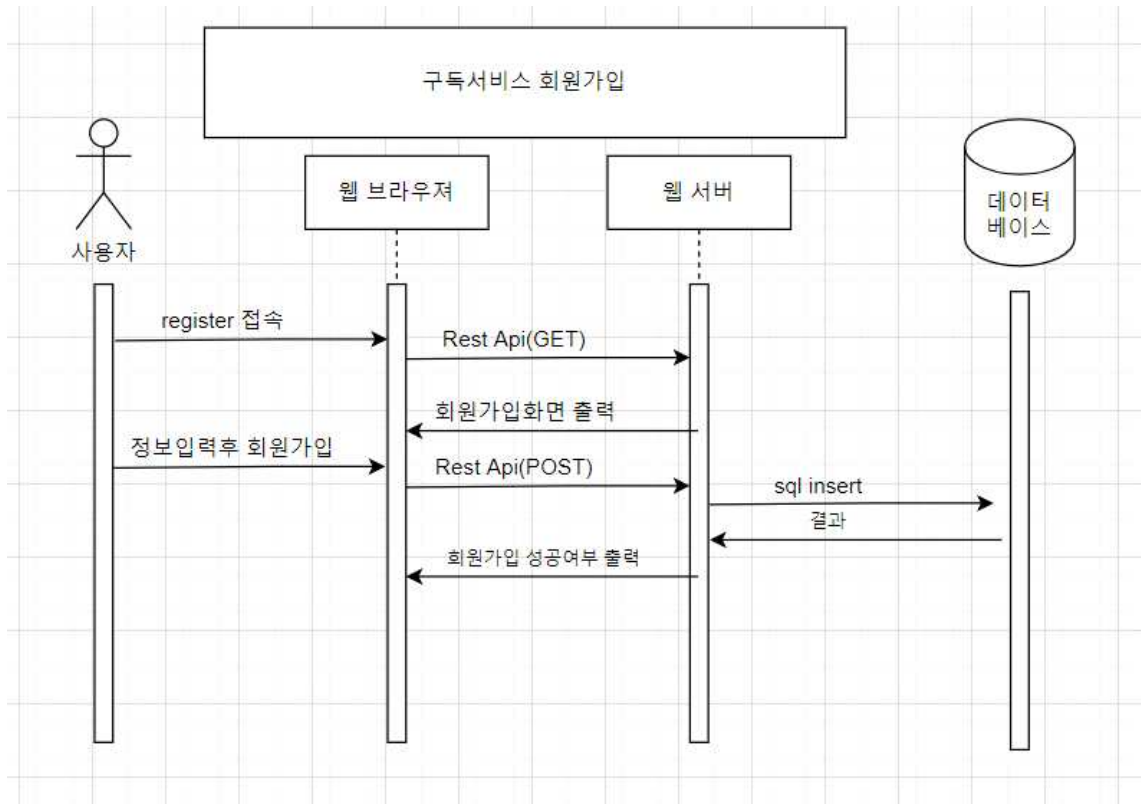
'netflix'의 경우 최종점수는 83이 된다. 그리고 'watcha'의 경우 최종점수는 58이 되게 된다. 'netflix'는 여섯 번째 질문에 높은 평가를 받았고 'watcha'는 세 번째 질문에 높은 평가를 받았는데, 유저가 더 중요하게 생각하는 평가는 여섯 번째 질문이었기 때문에 'netflix'가 더 좋은 점수가 나올 수 있었다.

위와 같은 방식으로 카테고리내의 모든 서비스를 비교하여, 최종점수가 가장 높은 서비스를 유저에게 추천한다.

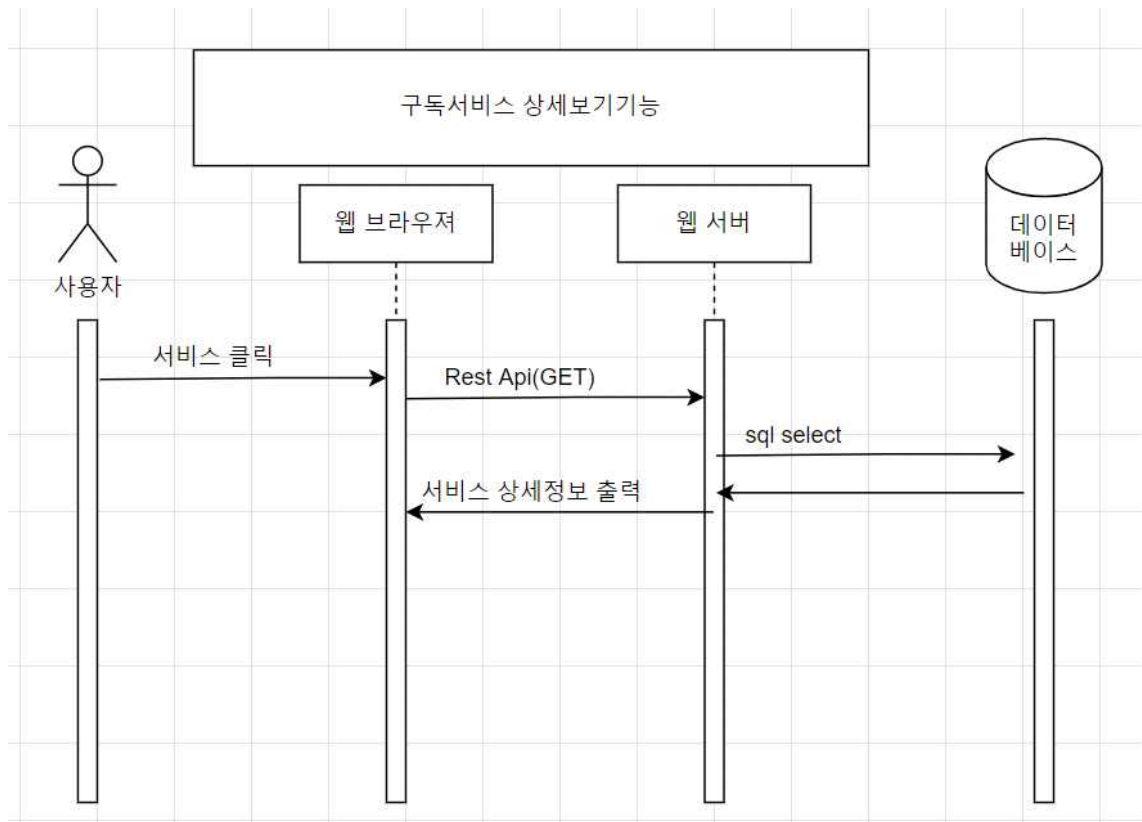
3.4 UML 다이어그램 설계



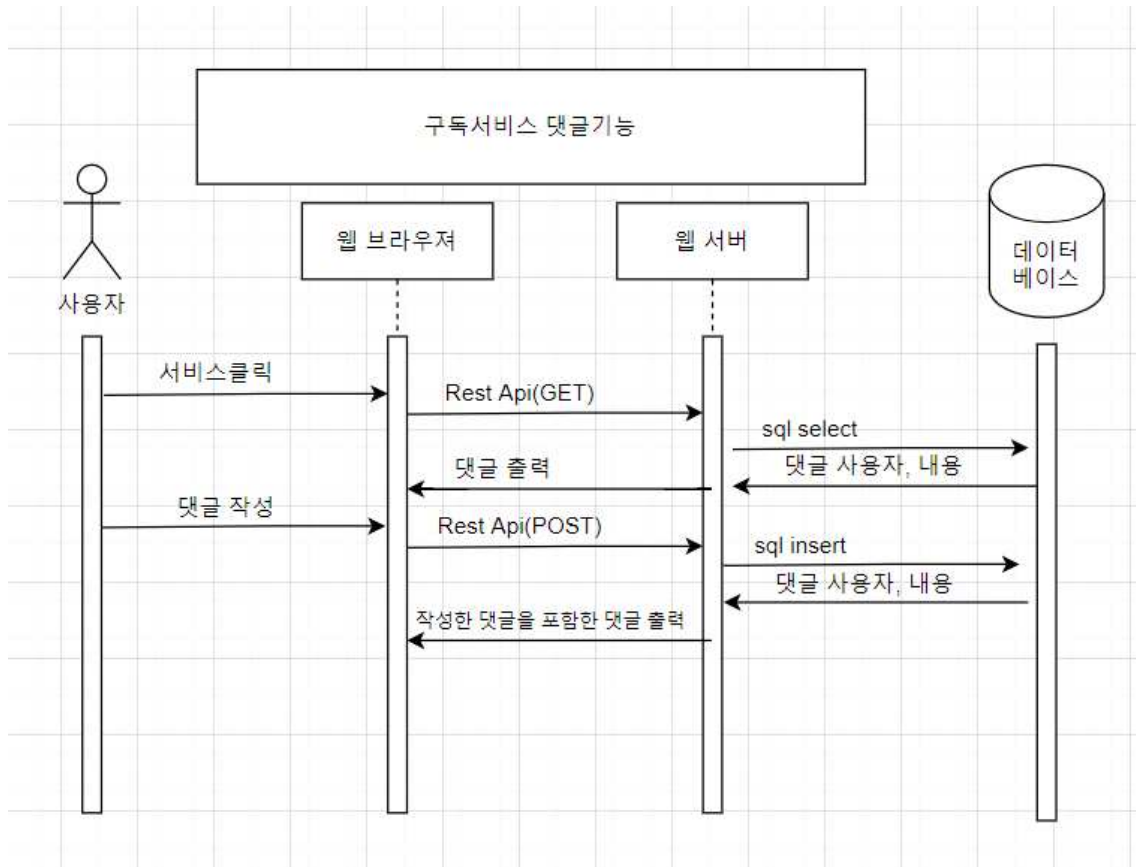
[그림 3.4.1 구독서비스 로그인]



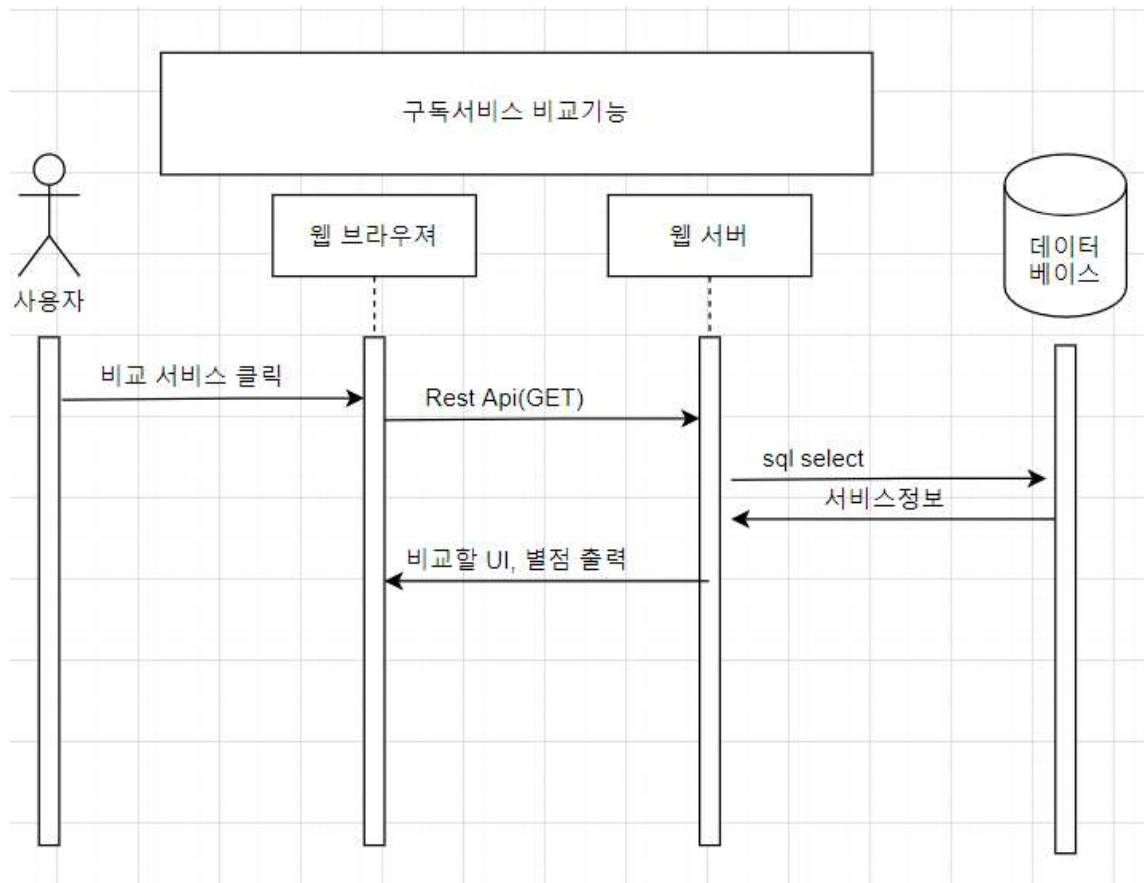
[그림 3.4.2 구독서비스 회원가입]



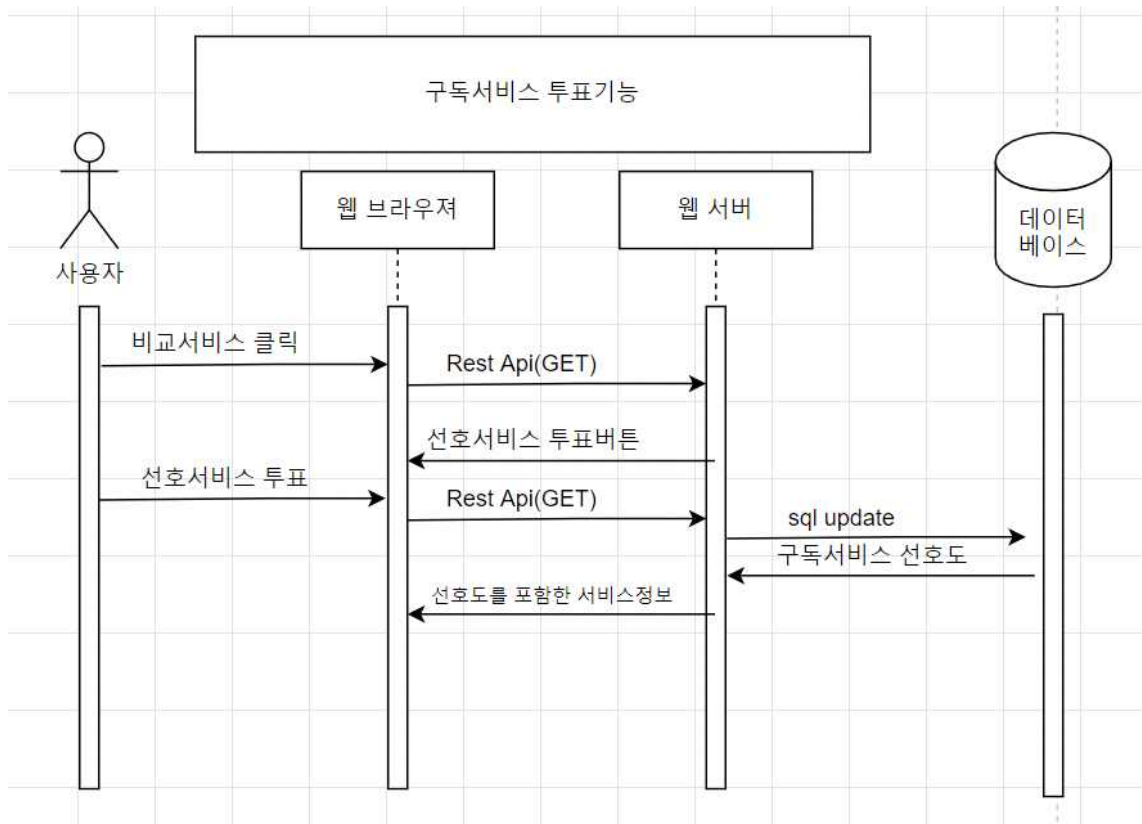
[그림 3.4.3 구독서비스 상세보기]



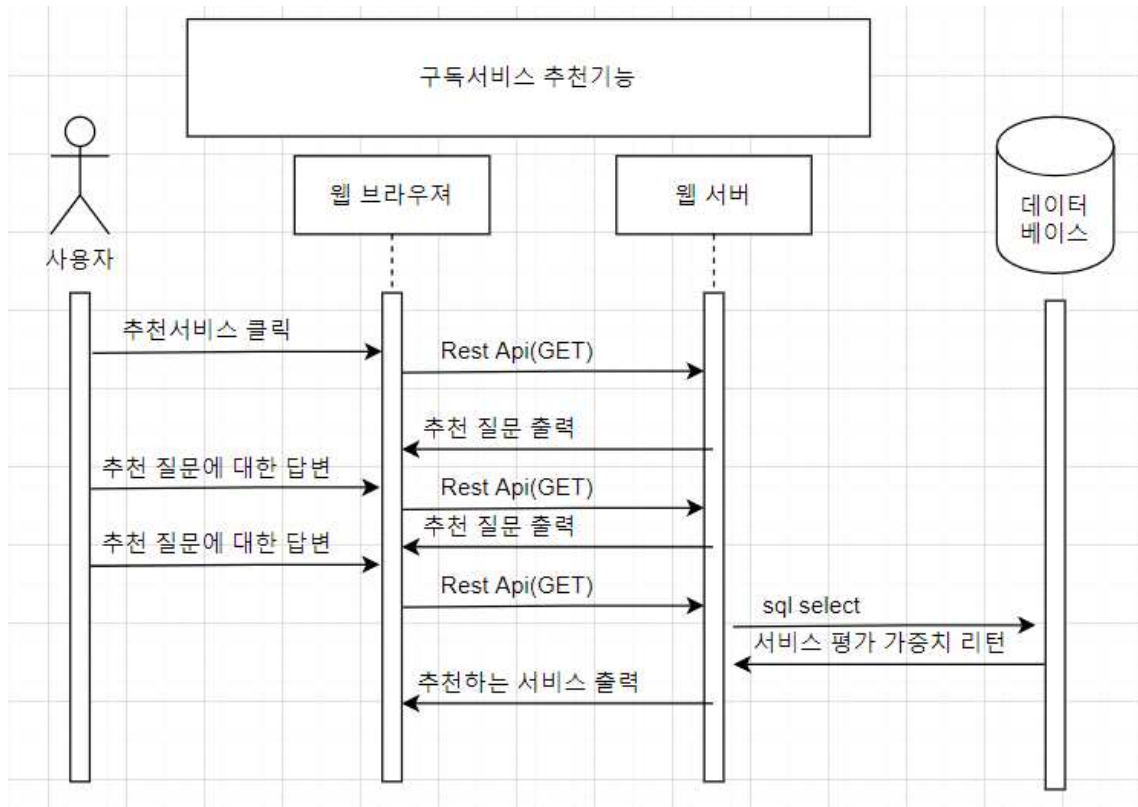
[그림 3.4.4 구독서비스 댓글]



[그림 3.4.5 구독서비스 비교]



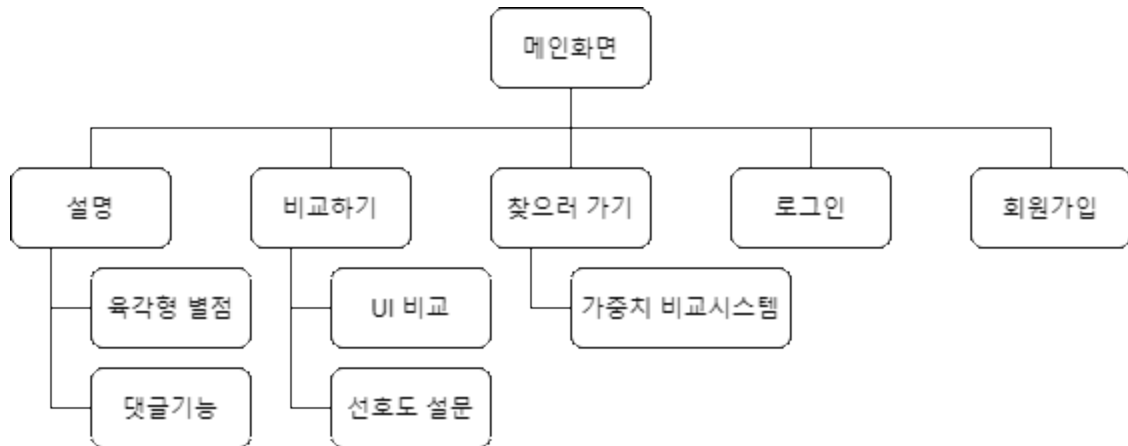
[그림 3.4.6 구독서비스 선호도 투표]



[그림 3.4.7 구독서비스 추천]

4. 화면(UI) 설계

4.1 스토리보드(메뉴) 구성



[그림 4.1.1 일반 사용자 메뉴구조도]

1) 설명

메인페이지에서 구독 서비스의 로고를 클릭하면 설명과 육각형 별점을 볼 수 있고, 댓글기능을 사용할 수 있다.

2) 비교하기

메인페이지에서 두 가지 구독 서비스를 선택 후 ‘비교하기’ 버튼을 클릭하면 UI비교가 가능하고, 두 가지 구독 서비스의 육각형 별점과 설명을 통해 선호도를 고를 수 있고, 이를 선택하면 통계가 나온다.

3) 찾으러가기

메인페이지에서 ‘찾으러 가기’ 버튼을 클릭하면 6가지 공통질문과 6가지 카테고리별 질문에 답변을 하고 이를 통해 사용자가 선호할만한 구독서비스를 추천받는다.

4) 로그인

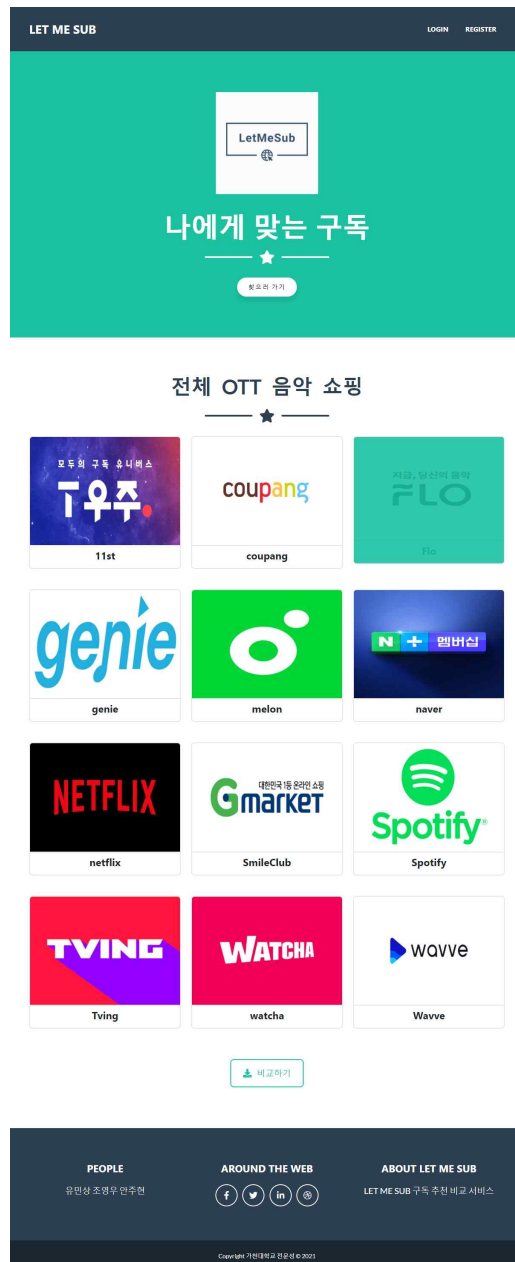
사용자가 로그인 할 수 있다.

5) 회원가입

사용자가 회원가입을 할 수 있다.

4.2 UI 설계

메인 페이지



[그림 4.2.1 메인 페이지]

1) LET ME SUB 홈페이지에 들어오면 나타나는 화면이다.

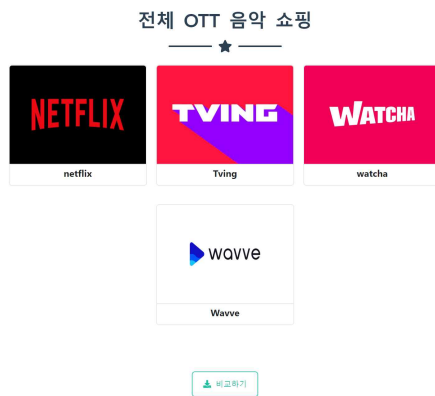
2) 오른쪽 상단위에 LOGIN과 REGISTER를 통해 회원가입이 가능하며, 로그인이 가능하다.

3) 나의 구독 밑에 '찾으러 가기' 버튼을 누르면 추천페이지로 넘어간다.

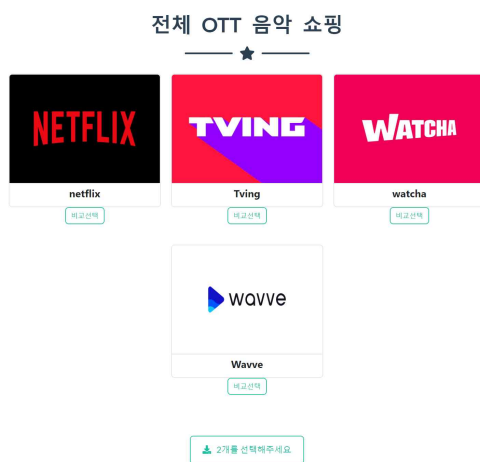
4) 카테고리를 클릭하면 화면이 카테고리 별로 구독 서비스를 볼 수 있다.

5) '비교하기' 버튼을 누르면 두 가지 구독 서비스를 선택해서 비교 페이지로 넘어간다.- 카테고리 OTT 선택

6) 구독 서비스 로고를 두 번 클릭하면 구독 서비스 설명 페이지로 넘어간다.



[그림 4.2.2 OTT 선택]

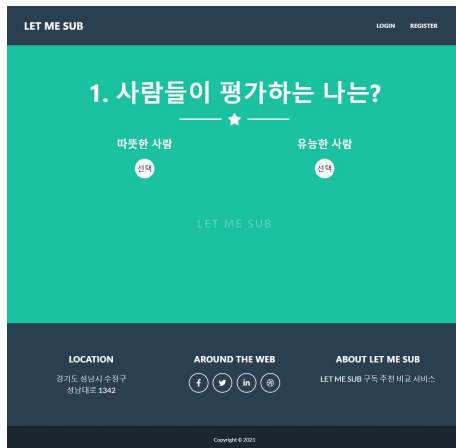


[그림 4.2.3 비교하기 버튼 선택]

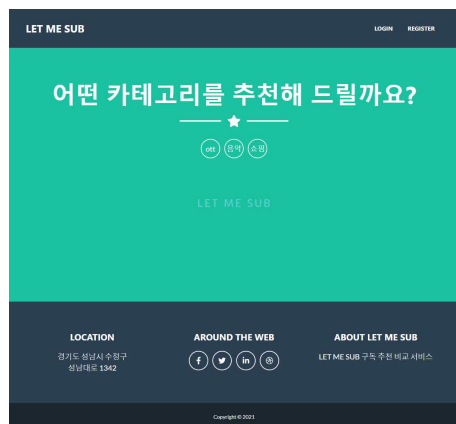
1) 메인 페이지의 OTT를 선택하면 OTT 카테고리만 보여준다.

1) 비교하기 버튼을 클릭하면 선택창이 활성화 되어 2가지 구독서비스를 선택할 수 있다.

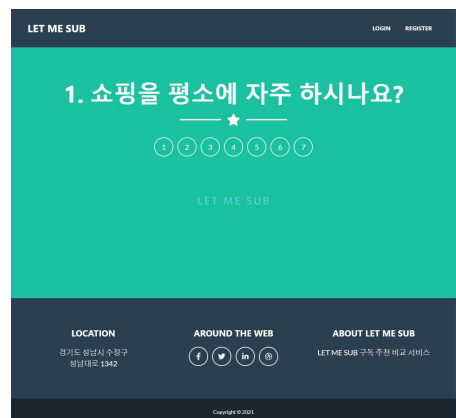
추천 페이지



[그림 4.2.2 추천페이지]



[그림 4.2.3 추천 카테고리 선택]

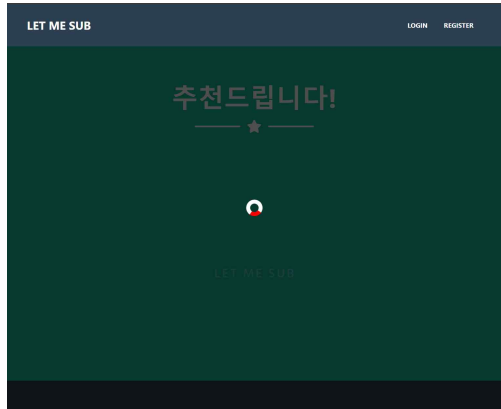


[그림 4.2.4 카테고리별 질문]

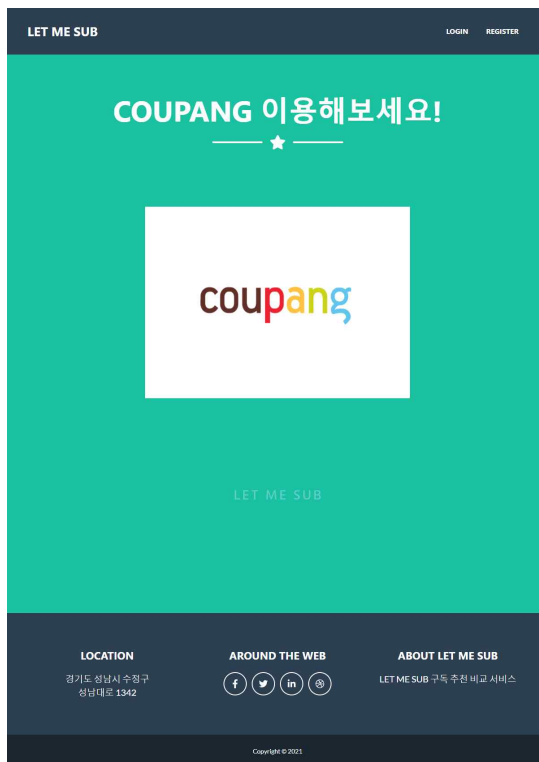
1) ‘찾으러 가기’ 버튼을 클릭하면 추천 페이지로 넘어온다.

2) 추천페이지에서 6가지 공통 질문에 답변하면 카테고리를 선택하는 질문이 나온다.

3) 카테고리를 선택하면 카테고리에 따른 질문이 나온다.



[그림 4.2.5 답변 완료]

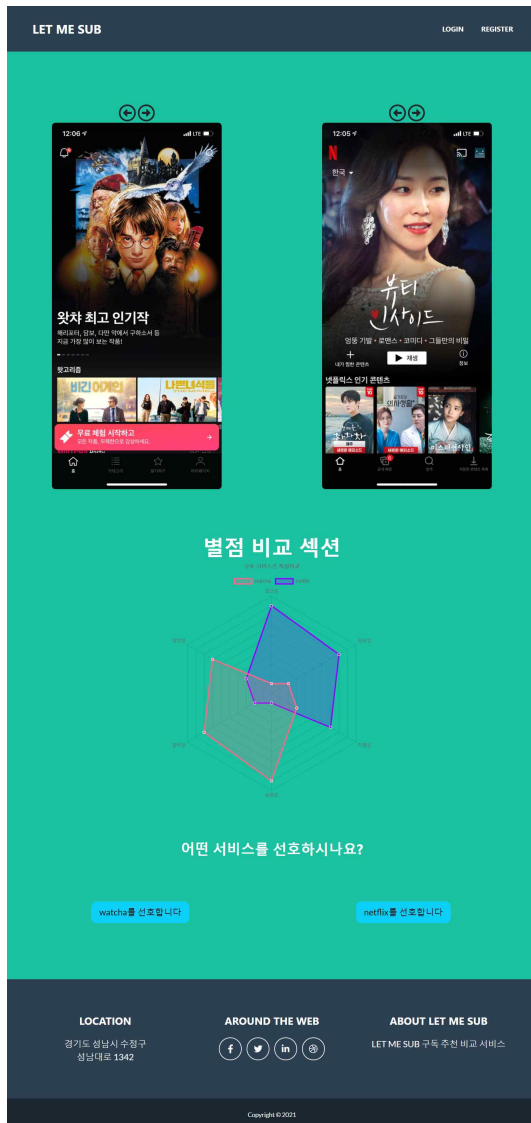


[그림 4.2.6 답변 결과]

4) 답변이 완료되면 잠시 후 설문 결과가 나온다.

5) 답변에 따라 추천 알고리즘에 의해 알맞은 구독 서비스를 추천해준다.

비교 페이지

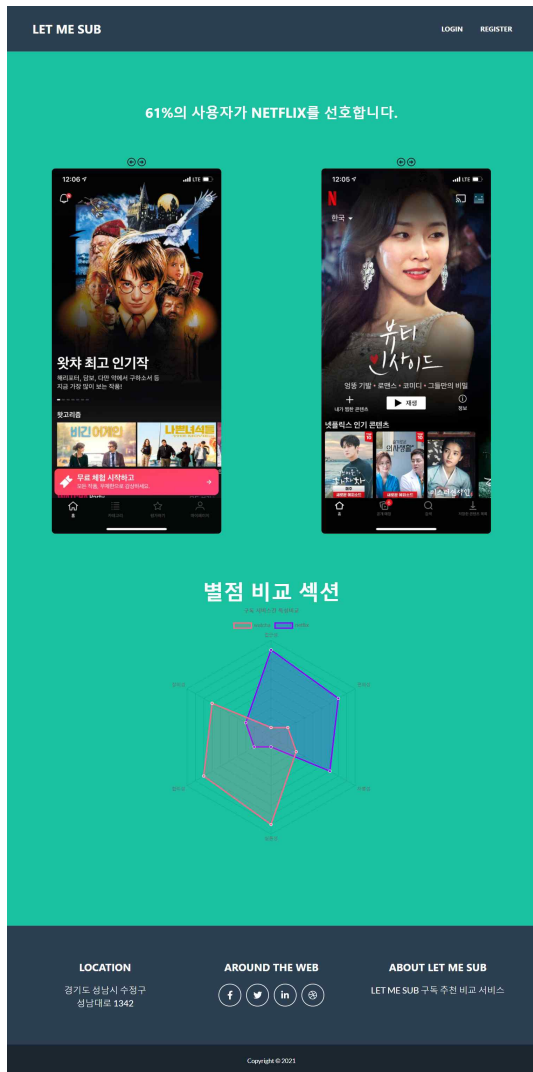


[그림 4.2.7 비교 페이지]

1) [그림 4.2.3]에서 두 개의 구독 서비스를 고르고 ‘두개를 선택해 주세요’ 버튼을 누르면 나타나는 비교 페이지다.

2) 두 개의 구독 서비스의 UI가 표시되고 육각형 모양의 별점이 겹쳐져서 비교하기 쉽게 보여준다.

3) 하단에 설문 어떤 구독 서비스를 선호하는지 설문 버튼을 두어 선택 시 선호도를 보여준다.



[그림 4.2.8 선호도 결과]

4) 'netfliex를 선호합니다' 버튼을 누르면 선호도 결과를 보여준다.

설명 페이지



[그림 4.2.9 설명 페이지]

1) 메인 페이지에서 'NETFLIX' 로고를 누르면 설명 페이지로 넘어간다.

2) 육각형 모양의 별점이 표시되고, 설명이 적혀있다.

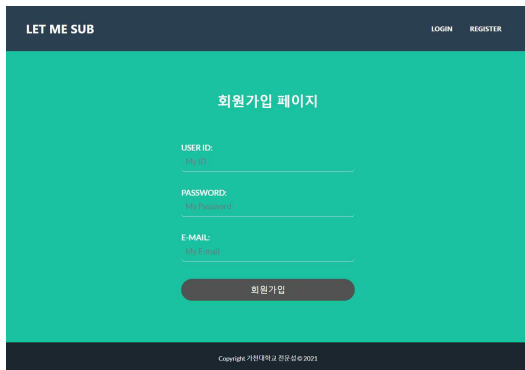
3) 하단에 리뷰를 남길 수 있는 댓글창이 있다.



[그림 4.2.10 댓글 입력]

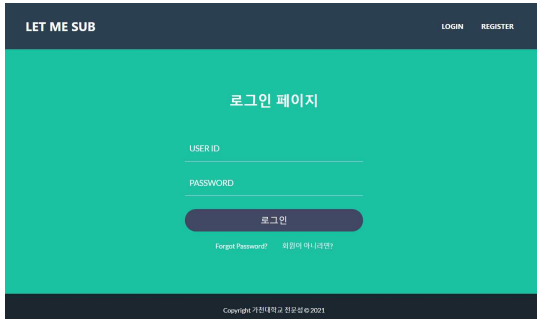
4) 리뷰를 작성하고 ‘댓글 작성하기’ 버튼을 누르면 댓글이 작성된다.

회원 가입 페이지

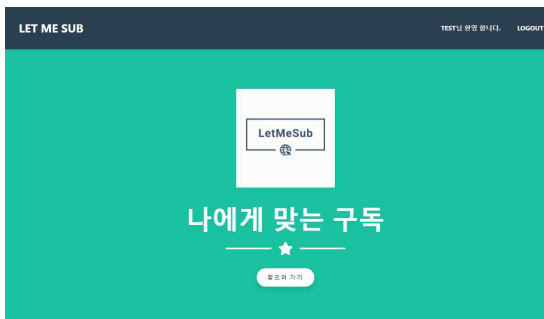


[그림 4.2.11 회원 가입 페이지]

로그인 페이지



[그림 4.2.12 로그인 페이지]



전체 OTT 음악 쇼핑

[그림 4.2.13 로그인 완료]

1) 메인페이지 상단에 'REGISTER' 버튼을 클릭하면 회원가입 페이지로 넘어간다.

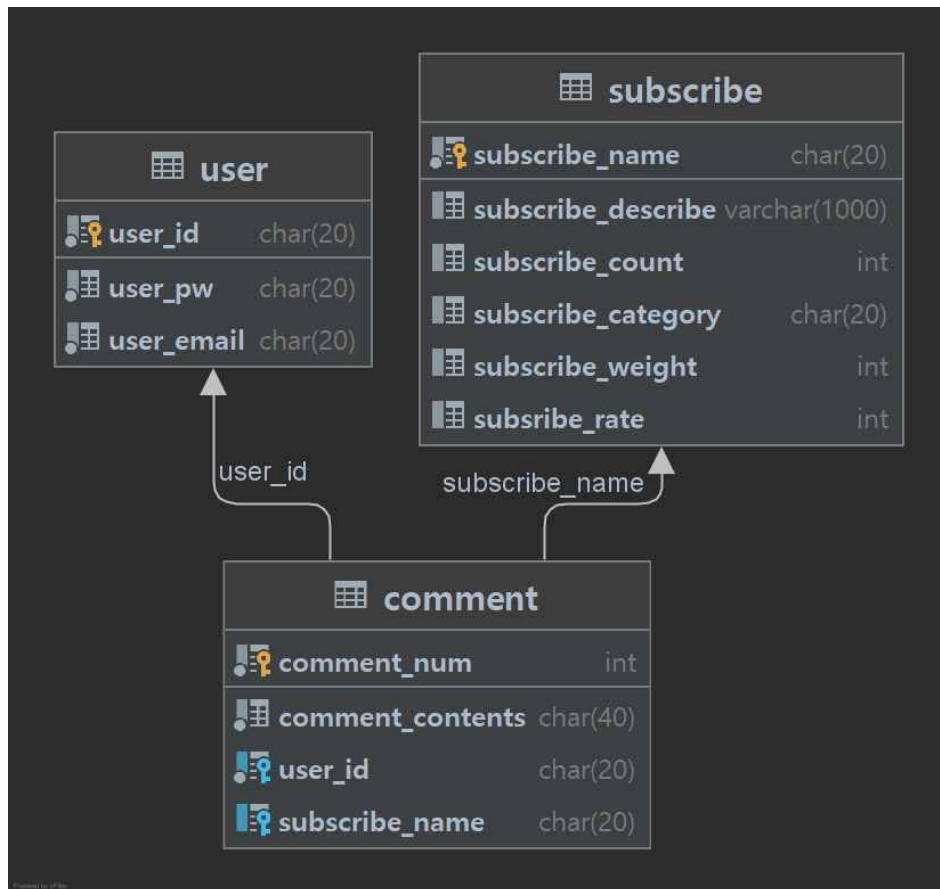
2) 유저 아이디와 패스워드, 이메일을 입력하면 회원가입이 완료된다.

1) 메인페이지 상단에 'LOGIN' 버튼을 클릭하면 로그인 페이지로 넘어간다.

2) 로그인이 완료되면 오른쪽 상단의 버튼이 '[유저 아이디]님 환영합니다'와 'LOGOUT' 버튼으로 바뀐다.

5. 데이터 베이스 설계

5.1 데이터 베이스 구조



[그림 5.1.1 데이터 베이스 구조]

5.2 논리적 DB설계 (테이블명세서)

테이블명			subscribe			
NO	Attribute	Data Type	NN	PK	FK	Description
1	subscribe_name	char(20)	O	O		구독서비스 이름
2	subscribe_describe	varchar(1000)				구독서비스 설명
3	subscribe_count	int				선호 카운트
4	subscribe_category	char(20)				카테고리
5	subscribe_weight	int				가중치
6	subscribe_rate	int				육각형 별점

테이블명			comment			
NO	Attribute	Data Type	NN	PK	FK	Description
1	comment_num	int	O	O		댓글 순서
2	comment_contents	char(40)	O			댓글 내용
3	user_id	char(20)	O			작성자
4	subscribe_name	char(20)				구독서비스 이름

테이블명			user			
NO	Attribute	Data Type	NN	PK	FK	Description
1	user_id	char(20)	O	O		유저 아이디
2	user_pw	char(20)	O			유저 비밀번호
3	user_email	char(20)				유저 이메일

5.3 물리적 DB설계 (SQL스크립트)

테이블명	comment
	<pre>CREATE TABLE `letmesub`.`subscribe` (`subscribe_name` char(20) NOT NULL, `subscribe_describe` varchar(1000) DEFAULT NULL, `subscribe_count` int DEFAULT NULL, `subscribe_category` char(20) DEFAULT NULL, `subscribe_weight` int DEFAULT NULL, `subscribe_rate` int DEFAULT NULL, PRIMARY KEY (`subscribe_name`)) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE=utf8mb4_0900_ai_ci;</pre>

테이블명	comment
	<pre>CREATE TABLE `letmesub`.`comment` (`comment_num` int NOT NULL AUTO INCREMENT, `comment_contents` char(40) NOT NULL, `user_id` char(20) NOT NULL, `subscribe_name` char(20) DEFAULT NULL, PRIMARY KEY (`comment_num`), KEY `user_id` (`user_id`), KEY `subscribe_name` (`subscribe_name`), CONSTRAINT `comment_ibfk_1` FOREIGN KEY(`user_id`) REFERENCES `user` (`user_id`), CONSTRAINT `comment_ibfk_2` FOREIGN KEY(`subscribe_name`) REFERENCES `subscribe` (`subscribe_name`)) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE=utf8mb4_0900_ai_ci;</pre>

테이블명	user
	<pre>CREATE TABLE `letmesub`.`user` (`user_id` char(20) NOT NULL, `user_pw` char(20) NOT NULL, `user_email` char(20) NOT NULL, PRIMARY KEY (`user_id`)) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4 COLLATE=utf8mb4_0900_ai_ci;</pre>

6. 환경구성

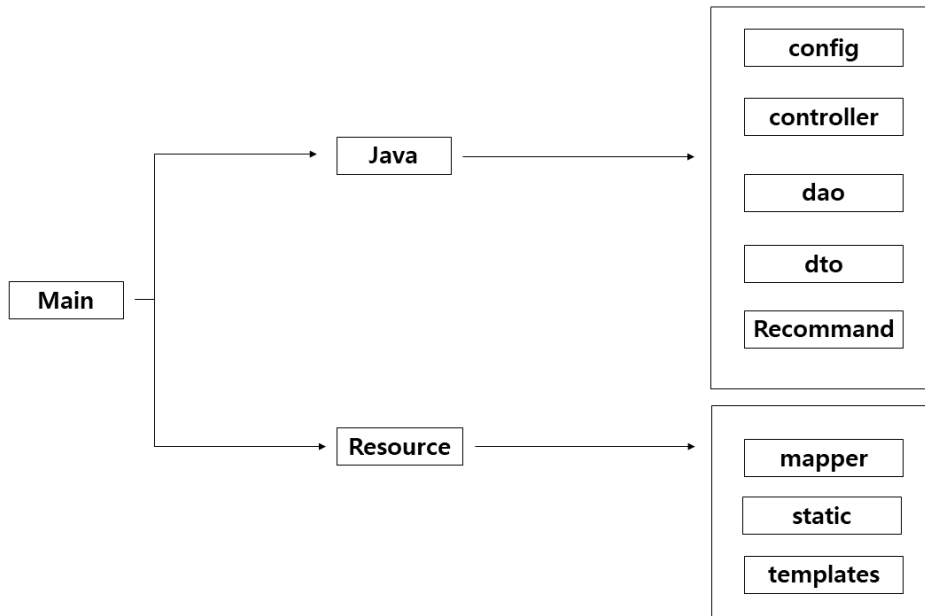
6.1 개발환경

구분	개발환경	규격
개발툴	편집툴	IntelliJ IDEA 2021.2.1
	프레임워크	Spring Boot
	언어	Java, JavaScript
	형상 관리	Git : https://github.com/LetMeSub/LetMeSub
	사용할 미들웨어(모듈)	Spring-Mysql-connector Spring-session-data-redis Spring-jdbc Spring-lombok Spring-thymleaf Spring-mybatis fasterxml.jackson Tomcat Server Gradle JQuery Bootstrap
서버	OS	Mac
	DB	MySQL Database
	Web	Chrome
서버 하드웨어	Apple M1 Macbook	

6.2 운영환경

구분	개발환경	규격	비고
서버 컴퓨터 AWS EC2	OS	Linux Ubuntu 20.04	
	CPU	t2.micro	
	RAM	1Gb	
	DB	AWS RDS(MySQL)	

6.3 소스디렉터리 구조



[그림 6.3.1 소스 디렉터리 구조]

Java : 서버 관련 디렉토리

- 1) **config** : DB와 연결을 위해 세팅을 하는 클래스들
- 2) **controller** : 서비스 처리를 위한 컨트롤러들
- 3) **dao** : mybatis 라이브러리를 이용해 DB와 매핑을 하기 위한 인터페이스들
- 4) **dto** : 서비스 구현을 위한 DataObjects

Recommand : 추천 기능 구현을 하기위한 클래스들

- 1) **Resource** : html, css ,javascript 등 프론트엔드 구현을 위한 디렉터리
- 2) **mapper** : mybatis 라이브러리를 이용해 mysql과 연결을 위해 필요한 xml
- 3) **static** : 이미지파일, css파일, 자바스크립트등 프론트엔드를 구현하기 위한 디렉터리
- 4) **templates** : 렌더링을 수행할 html 파일들