

소프트웨어 요구 사항 v1.0.0

2023년 8월 4일

Blog

남궁범
류형환
유진수
이슬기
이승빈
최승현

1. 목차

목차	2
1.0 소개	3
1.1. 목적	3
1.2. 프로젝트 범위	3
1.3. 용어집	3
1.4. 문서 개요	4
2.0 전체 설명	5
2.1 시스템 환경	5
2.2 사용사례	6
2.2.1 Viewer의 사용 사례	6
2.2.1.1 사용 사례 : 글 조회	6
2.2.1.2 사용 사례 : 글 검색	7
2.2.1.3 사용 사례 : 회원가입	7
2.2.2 User의 사용 사례	8
2.2.2.1 사용 사례: 로그인	8
2.2.2.2 사용 사례: 글 작성	8
2.2.2.3 사용 사례: 글 수정	9
2.2.2.4 사용 사례: 댓글 작성	9
2.2.2.5 사용 사례: 댓글 수정	9
2.2.2.6 사용 사례: 프로필 설정	10
2.2.3 Admin의 사용 사례	10
2.2.3.1 사용 사례: 글 관리	11
2.2.3.2 사용 사례: 댓글 관리	11
2.2.3.3 사용 사례: 사용자 관리	11
3.0 기능적 요구사항	12
3.1 메인 페이지	12
3.2 회원가입 기능	12
3.3 로그인 기능	12

3.4 게시물 작성 기능	12
3.5 게시물 목록 기능	13
3.6 게시물 수정 기능	13
3.7 게시물 상세보기 기능	13
3.8 게시물 검색 기능	13
3.9 게시물 삭제 기능	13
3.10 회원 관리 기능	14
3.11 댓글 기능	14
4.0 비기능적 요구사항	15
4.1 보안 요구사항	15
4.2 성능 요구사항	15
4.3 사용성 요구사항	15
4.4 확장성 요구사항	15
4.5 번역 기능 요구사항	15
4.6 코드 품질 테스트링 요구사항	15
4.7 배포 요구사항	15
논리 구조 요구사항	16

2. 소개

2.1. 목적

이 문서의 목적은 사용중인 블로그에 대한 자세한 설명을 제공합니다. 블로그의 목적과 특징, 블로그에서 수행할 수 있는 작업, 블로그를 이용하기 위해 필요한 최소한의 제약 조건에 대해 설명합니다. 이 문서는 블로그를 이용하는 이용자와 개발자 모두를 위해 작성되었습니다.

2.2. 프로젝트 범위

이 프로젝트는 회원가입 후 로그인한 사용자들의 다양한 게시글 작성을 위해 **Django Rest Framework** 를 통해 개발되어진 블로그입니다. 가입한 사용자들에게 글 작성 및 상세페이지 보기 기능이 제공되어지며, 댓글 및 해시태그 입력 또한 로그인한 사용자에게 권한이 주어지도록 설계되었습니다. 이 블로그는 이용자들끼리 다양한 주제를 가지고 유익한 정보들을 공유하여 다양한 정보를 얻고 나눌 수 있는 유익한 장으로 사용되어질 수 있습니다. 또한 댓글을 통해 해당 게시글에 대한 다양한 의견을 나눌 수 있고, 해시태그를 통해 원하는 정보를 쉽게 검색할 수 있습니다.

2.3. 용어집

용어	정의
블로그	이 프로젝트를 가리키는 용어. 다양한 게시글을 작성하고, 해시태그 작성 및 댓글 작성이 가능합니다.
Actor	이 프로젝트를 사용하는 모든 사람들을 의미합니다.
사용자(User)	이 프로젝트에 가입하여 제한적인 기능들을 사용할 수 있는 사람들을 의미합니다.
관리자(Admin)	이 블로그를 이용하는 사용자들을 관리하며, 블로그

	내에서 이용되는 서비스가 원활하게 동작할 수 있도록 관리하는 권한이 있는 사람을 의미합니다.
방문자(Viewer)	블로그 글을 읽기 위해 사이트를 방문한 모든 사람을 의미합니다.
회원가입	블로그를 사용하기 위한 방문자들이 본인의 정보를 입력하여 블로그 내에서 다양한 게시물, 해시태그, 댓글을 작성하기 위한 계정을 등록하는 기능입니다.
로그인	이 블로그를 사용하기 위한 사용자들이 회원가입을 통해 작성한 유저아이디와 비밀번호를 입력하여 블로그를 이용할 수 있도록 만들어진 기능입니다.
게시글	블로그에 로그인한 사용자들이 다양한 주제를 가지고 정보를 공유하기 위해 작성하는 글, 이미지, 파일 등을 브라우저를 통해 볼 수 있습니다.
댓글	작성된 1개의 게시물에 로그인을 한 다양한 사용자들이 의견을 제시하거나 공감하는 글을 작성할 수 있는 기능입니다.
해시태그	작성된 게시글의 큰 특징을 단어별로 게시물 하단에 추가하여 이후에 사용자들이 검색어를 통해 해당 게시글을 쉽게 검색할 수 있도록 하는 기능입니다.

2.4. 문서 개요

이 문서의 다음장에서 블로그의 기능적 요구사항 및 비기능적 요구사항에 대한 설명이 제공되어집니다. 기능적 요구사항 및 비기능적 요구사항은 주로 개발자를 위해 작성되었으며, 이 블로그의 기능에 대한 세부사항을 기술적인 용어로 설명합니다.

3. 전체 설명

3.1. 시스템 환경

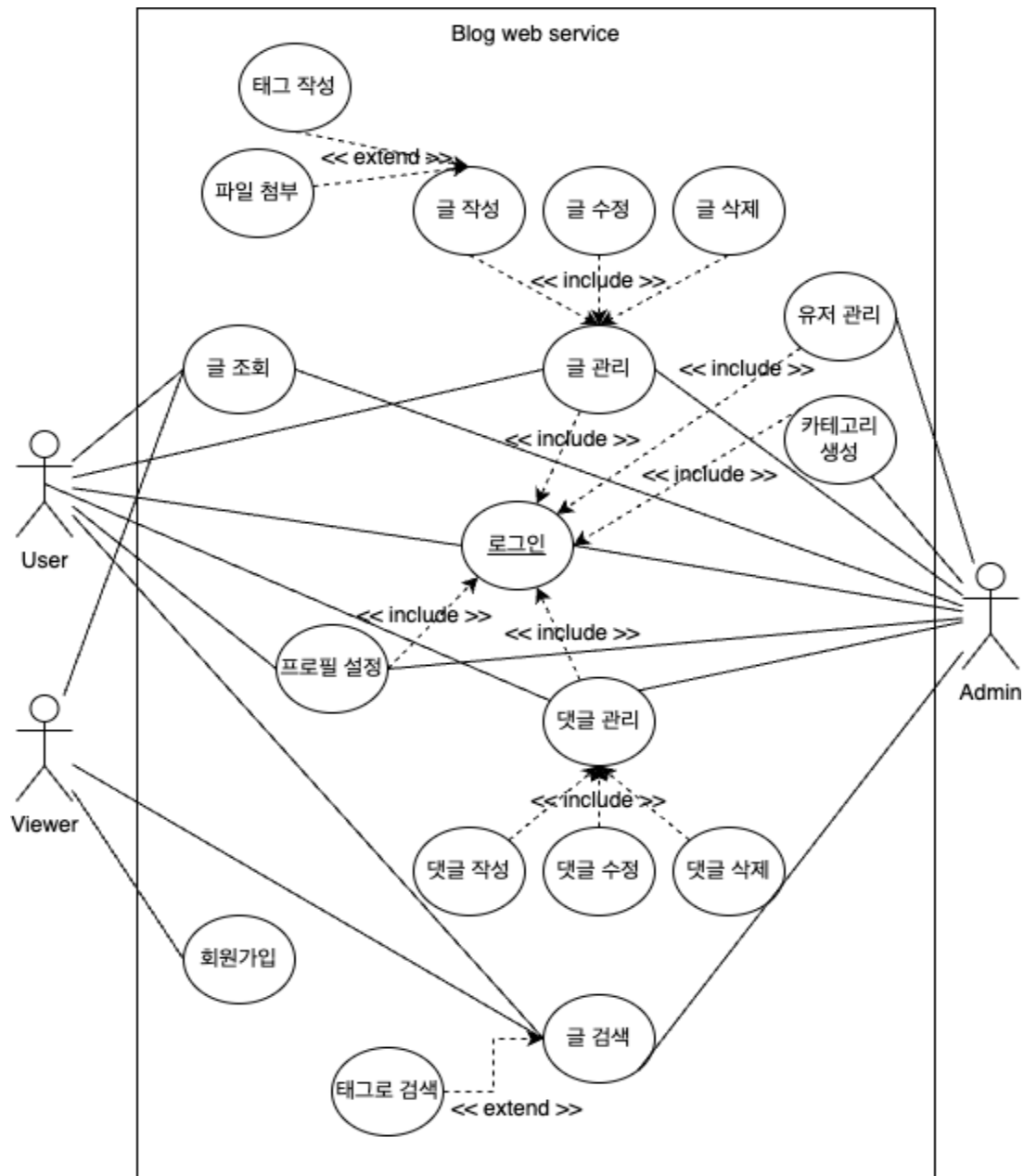


그림1 - 시스템 환경

이 블로그에는 세명의 **Actor**가 있습니다. **User**는 회원가입 후 로그인을 한 사용자를 의미하며 글 조회, 글 관리, 로그인, 댓글관리, 프로필 설정, 글 검색 기능을 사용할

수 있습니다. **Viewer**는 회원가입 전 블로그를 처음 방문한 방문자를 의미하며 글 조회, 글 검색, 회원가입 기능을 사용할 수 있습니다. **Admin**은 이 블로그를 관리하는 관리자를 의미하며, 유저관리, 카테고리 생성, 글 관리, 글 조회, 로그인, 프로필 설정, 댓글관리, 글 검색 등의 기능을 사용할 수 있습니다.

3.2. 사용사례

3.2.1. Viewer의 사용 사례

애플리케이션에 접속하여 블로그 글을 읽기, 글 검색, 회원가입 을 할 수 있는 기본적인 권한을 가진 사용자입니다.

3.2.1.1. 사용 사례 : 글 조회

다이아그램 :



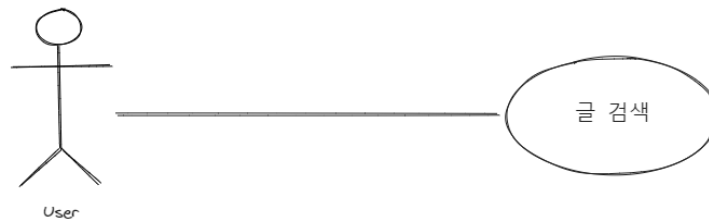
간단한 설명 : 사용자는 게시글 목록 페이지 또는 메인 페이지에서 모든 게시글을 볼 수 있습니다.

초기 단계별 설명:

1. 사용자는 메인 페이지 또는 게시글 목록 페이지에 접속합니다.
2. 시스템은 모든 게시글을 최신순 또는 인기순 등으로 정렬하여 사용자에게 표시합니다.
3. 사용자는 게시글의 제목, 작성자, 작성일 등 기본 정보를 확인할 수 있습니다.
4. 사용자는 원하는 게시글을 선택하여 상세보기 페이지로 이동할 수 있습니다.
5. 상세보기 페이지에서는 게시글의 전체 내용과 작성자 정보를 볼 수 있습니다.

3.2.1.2. 사용 사례 : 글 검색

다이어그램:



간단한 설명: 사용자는 키워드를 사용하여 게시글을 검색할 수 있습니다.

초기 단계별 설명:

1. 사용자는 메인 페이지 또는 글 목록 페이지에서 검색 기능을 찾아 접근합니다.
2. 사용자는 검색 입력 필드에 원하는 키워드를 입력합니다.
3. 사용자는 검색 버튼을 클릭하여 검색을 시작합니다.
4. 시스템은 입력된 키워드와 관련된 게시글을 찾아서 검색 결과를 표시합니다.
5. 사용자는 검색 결과를 확인하고 원하는 게시글을 선택하여 상세보기 페이지로 이동할 수 있습니다.

3.2.1.3. 사용 사례 : 회원가입

다이어그램:



간단한 설명: 사용자는 회원가입 페이지를 통해 새로운 계정을 생성할 수 있습니다.

초기 단계별 설명:

1. 사용자는 회원가입 페이지에 접속합니다.
2. 사용자는 필요한 정보인 사용자 이름, 이메일, 비밀번호 등을 입력합니다.

3. 사용자는 입력한 정보가 정확한지 확인하고, 필수 정보를 모두 입력한 후 가입 버튼을 클릭합니다.
4. 시스템은 입력된 정보를 검증하고, 올바른 정보 라면 새로운 계정을 생성합니다.
5. 사용자는 가입 성공 메시지를 확인하고 로그인 페이지로 이동하여 새로운 계정으로 로그인할 수 있습니다.

3.2.2. User의 사용 사례

- 뷰어의 권한을 포함하며, 추가적으로 로그인, 글 작성 및 관리, 댓글 작성 및 관리, 프로필 설정 기능을 사용할 수 있습니다.

3.2.2.1. 사용 사례: 로그인

간단한 설명: 사용자는 자신의 계정으로 로그인하여 애플리케이션의 기능을 사용할 수 있습니다.

초기 단계별 설명:

1. 사용자는 로그인 페이지에 접속합니다.
2. 사용자는 등록된 이메일과 비밀번호를 입력합니다.
3. 사용자는 로그인 버튼을 클릭하여 로그인 시도를 합니다.
4. 시스템은 사용자의 입력 정보를 검증하고, 올바른 정보라면 인증된 세션을 생성합니다.
5. 사용자는 성공적으로 로그인되어 애플리케이션의 기능을 사용할 수 있습니다.

3.2.2.2. 사용 사례: 글 작성

간단한 설명: 로그인한 사용자는 게시글 작성 페이지를 통해 새로운 글을 작성할 수 있습니다.

초기 단계별 설명:

1. 사용자는 로그인합니다.
2. 로그인한 사용자는 글 작성 페이지에 접속합니다.

3. 사용자는 게시글 제목과 내용을 입력합니다.
4. 사용자는 글 작성 버튼을 클릭하여 새로운 글을 작성합니다.
5. 시스템은 작성된 글을 저장하고, 사용자의 글 목록에 추가합니다.
6. 사용자는 작성한 글을 성공적으로 등록한 것을 확인할 수 있습니다.

3.2.2.3. 사용 사례: 글 수정

간단한 설명: 로그인한 사용자는 자신이 작성한 글을 수정할 수 있습니다.

초기 단계별 설명:

1. 사용자는 로그인합니다.
2. 로그인한 사용자는 글 목록 페이지에서 수정할 글을 선택합니다.
3. 사용자는 글 상세보기 페이지로 이동하여 수정 버튼을 클릭합니다.
4. 사용자는 글의 제목과 내용을 수정합니다.
5. 사용자는 수정한 내용을 저장하기 위해 수정 완료 버튼을 클릭합니다.
6. 시스템은 수정된 글 내용을 업데이트하고, 사용자의 글 목록에 반영합니다.
7. 사용자는 글이 성공적으로 수정되었음을 확인할 수 있습니다.

3.2.2.4. 사용 사례: 댓글 작성

간단한 설명: 로그인한 사용자는 게시글 상세보기 페이지에서 댓글을 작성할 수 있습니다.

초기 단계별 설명:

1. 사용자는 로그인합니다.
2. 로그인한 사용자는 특정 게시글을 선택하여 게시글 상세보기 페이지로 이동합니다.
3. 사용자는 댓글 작성을 위해 댓글 입력 필드에 내용을 입력합니다.
4. 사용자는 작성한 댓글을 저장하기 위해 댓글 작성 버튼을 클릭합니다.
5. 시스템은 작성한 댓글을 저장하고, 해당 게시글의 댓글 목록에 추가합니다.
6. 사용자는 작성한 댓글이 성공적으로 등록된 것을 확인할 수 있습니다.

3.2.2.5. 사용 사례: 댓글 수정

간단한 설명: 로그인한 사용자는 자신이 작성한 댓글을 수정할 수 있습니다.

초기 단계별 설명:

1. 사용자는 로그인합니다.
2. 로그인한 사용자는 댓글이 달린 게시글을 선택하여 게시글 상세보기 페이지로 이동합니다.
3. 사용자는 자신이 작성한 댓글을 수정하기 위해 댓글 영역에서 수정 버튼을 클릭합니다.
4. 시스템은 선택한 댓글의 내용을 수정할 수 있는 수정 입력 필드를 제공합니다.
5. 사용자는 수정 입력 필드에 변경할 내용을 입력합니다.
6. 사용자는 수정한 내용을 저장하기 위해 수정 완료 버튼을 클릭합니다.
7. 시스템은 수정된 댓글 내용을 업데이트하고, 해당 게시글의 댓글 목록에 반영합니다.
8. 사용자는 댓글이 성공적으로 수정되었음을 확인할 수 있습니다.

3.2.2.6. 사용 사례: 프로필 설정

간단한 설명: 사용자는 프로필 설정 페이지를 통해 자신의 정보를 관리할 수 있습니다.

초기 단계별 설명:

1. 사용자는 로그인합니다.
2. 로그인한 사용자는 프로필 설정 페이지로 이동합니다.
3. 사용자는 프로필 설정 페이지에서 이름, 이메일, 비밀번호 등의 정보를 수정합니다.
4. 사용자는 수정한 정보를 저장하기 위해 변경 완료 버튼을 클릭합니다.
5. 시스템은 수정된 정보를 업데이트하고, 사용자의 프로필 정보에 반영합니다.
6. 사용자는 프로필 정보가 성공적으로 변경된 것을 확인할 수 있습니다.

3.2.3. Admin의 사용 사례

모든 기능에 대한 접근 권한을 가진 사용자 역할로, 블로그 글 및 댓글 관리, 사용자 관리 등의 기능을 사용할 수 있습니다.

2.2.3.1 사용 사례: 글 관리

간단한 설명: 어드민 페이지에서 작성된 댓글을 관리할 수 있습니다.

초기 단계별 설명:

1. 사용자는 어드민 페이지에 로그인합니다.
2. 사용자는 어드민 페이지에서 작성된 글들을 확인할 수 있습니다.
3. 사용자는 댓글의 수정 버튼을 클릭하여 글을 수정할 수 있습니다.
4. 사용자는 댓글의 삭제 버튼을 클릭하여 글을 삭제할 수 있습니다.

3.2.3.2. 2 사용 사례: 댓글 관리

간단한 설명: 어드민 페이지에서 작성된 댓글을 관리할 수 있습니다.

초기 단계별 설명:

1. 사용자는 어드민 페이지에 로그인합니다.
2. 사용자는 어드민 페이지에서 작성된 댓글들을 확인할 수 있습니다.
3. 사용자는 댓글의 수정 버튼을 클릭하여 댓글을 수정할 수 있습니다.
4. 사용자는 댓글의 삭제 버튼을 클릭하여 댓글을 삭제할 수 있습니다.

3.2.3.3. 사용 사례: 사용자 관리

간단한 설명: 관리자는 사용자 계정을 생성, 수정, 삭제할 수 있습니다.

초기 단계별 설명:

1. 사용자는 어드민 페이지에 로그인합니다.
2. 사용자는 어드민 페이지에서 유저들을 확인할 수 있습니다.
3. 사용자는 댓글의 수정 버튼을 클릭하여 유저 정보를 수정할 수 있습니다.
4. 사용자는 댓글의 삭제 버튼을 클릭하여 유저를 삭제할 수 있습니다.

4. 기능적 요구사항

4.1. 메인 페이지

설명:

- 사용자가 애플리케이션의 메인 페이지에 접속할 수 있어야 합니다.

요구사항:

1. 메인 페이지는 애플리케이션의 소개 및 주요 기능(회원가입, 로그인, 게시판)을 제공할 수 있습니다.

4.2. 회원가입 기능

설명:

- 사용자는 회원가입 페이지를 통해 새로운 계정을 생성할 수 있어야 합니다.

요구사항:

1. 사용자 이름, 이메일, 비밀번호 등 필수적인 정보를 입력하여 회원가입이 가능해야 합니다.
2. 사용자 이름과 이메일은 중복되지 않도록 고유해야 합니다.
3. 올바른 이메일 형식을 준수해야 합니다.
4. 비밀번호는 최소 길이를 만족하고, 보안을 위해 암호화되어야 합니다.

4.3. 로그인 기능

설명:

- 사용자는 로그인 페이지를 통해 애플리케이션에 인증된 세션을 생성할 수 있어야 합니다.

요구사항:

1. 등록된 유저이름과 비밀번호를 입력하여 로그인이 가능해야 합니다.
2. 올바른 유저이름과 비밀번호를 확인해야 합니다.
3. 올바른 로그인 정보로만 세션 인증이 이루어져야 합니다.

4.4. 게시물 작성 기능

설명:

- 로그인한 사용자는 게시물 작성 페이지를 통해 새로운 글을 작성할 수 있어야 합니다.

요구사항:

1. 게시물 제목, 내용 등 필수적인 정보를 입력하여 작성이 가능해야 합니다.
2. 작성한 게시물은 로그인한 사용자에게 연결되어야 합니다.

4.5. 게시물 목록 기능

설명:

- 사용자는 게시물 목록 페이지에서 모든 게시물을 볼 수 있어야 합니다.

요구사항:

1. 게시물 목록은 최신순 또는 인기순 등으로 정렬 가능해야 합니다.
2. 게시물의 제목, 작성자, 작성일 등 기본 정보를 표시해야 합니다.

4.6. 게시물 수정 기능

설명:

- 로그인한 사용자는 자신이 작성한 게시물을 수정할 수 있어야 합니다.

요구사항:

1. 수정할 게시물을 선택하고 제목과 내용을 수정할 수 있어야 합니다.

4.7. 게시물 상세보기 기능

설명:

- 사용자는 게시물 목록 페이지에서 특정 게시물을 클릭하여 상세보기 페이지로 이동할 수 있어야 합니다.

요구사항:

1. 게시물 상세보기 페이지에서는 게시물의 전체 내용과 작성자 정보를 볼 수 있어야 합니다.

4.8. 게시물 검색 기능

설명:

- 사용자는 키워드를 사용하여 게시물을 검색할 수 있어야 합니다.

요구사항:

1. 검색 키워드를 입력하고, 검색 버튼을 눌러 검색이 가능해야 합니다.
2. 검색 결과는 키워드와 관련된 게시물 목록을 보여주어야 합니다.

4.9. 게시물 삭제 기능

설명:

- 로그인한 사용자는 자신이 작성한 게시물을 삭제할 수 있어야 합니다.

요구사항:

1. 삭제할 게시물을 선택하고, 확인 절차를 거친 후 게시물이 삭제되어야 합니다.

4.10. 회원 관리 기능

설명:

- 관리자는 사용자 계정을 생성, 수정, 삭제할 수 있어야 합니다.

요구사항:

1. 관리자 계정으로 로그인하여 사용자 관리 페이지를 통해 유저를 관리할 수 있어야 합니다.
2. 사용자 계정 생성 시 사용자 이름, 이메일, 비밀번호 등 필요한 정보를 입력할 수 있어야 합니다.
3. 관리자는 다른 사용자의 계정 정보를 수정하거나 삭제할 수 있어야 합니다.

4.11. 댓글 기능

설명:

- 로그인한 사용자는 게시글 상세보기 페이지에서 댓글을 작성, 수정, 삭제할 수 있어야 합니다.

요구사항:

1. 댓글 작성 기능: 댓글 입력 필드에 내용을 입력하고, 작성 버튼을 눌러 댓글을 작성할 수 있어야 합니다.
2. 댓글 수정 기능: 사용자는 자신이 작성한 댓글을 수정할 수 있어야 합니다.
3. 댓글 삭제 기능: 사용자는 자신이 작성한 댓글을 삭제할 수 있어야 합니다.

5. 비기능적 요구사항

5.1. 보안 요구사항

- 회원가입 시 비밀번호는 안전한 해시 알고리즘을 사용하여 암호화되어 저장되어야 합니다.
- 로그인 시에는 안전한 인증 프로토콜과 보안 접속(HTTPS)을 사용하여 사용자 인증이 이루어져야 합니다.
- 사용자의 개인 정보와 게시글에 접근하는 권한은 적절히 설정되어야 하며, 인가되지 않은 사용자로부터의 접근을 차단해야 합니다.
- 개인 정보와 민감한 데이터는 암호화되어 저장되어야 하며, 데이터베이스 백업 및 보안 조치가 필요합니다.

5.2. 성능 요구사항

- 웹 페이지 로딩 속도는 빠르도록 최적화되어야 합니다. 캐싱, 이미지 최적화 등을 고려하여 사용자 경험을 향상시켜야 합니다.
- 게시글 작성, 수정, 삭제, 조회 등의 기능은 빠르게 응답되어야 하며, 서버의 응답 시간 최소화가 필요합니다.
- 검색 기능은 빠르게 결과를 제공해야 하며, 검색 인덱싱 등을 활용하여 높은 성능을 유지해야 합니다.

5.3. 사용성 요구사항

- 사용자 친화적인 UI/UX가 제공되어야 합니다. 직관적이고 사용하기 쉬운 인터페이스를 디자인해야 합니다.
- 모바일 기기에서도 원활하게 사용할 수 있도록 반응형 웹 디자인을 구현해야 합니다.
- 에러 메시지와 안내 문구 등은 사용자가 이해하기 쉽도록 구성되어야 합니다.

5.4. 확장성 요구사항

- 향후 애플리케이션의 확장을 고려하여 유연하고 확장 가능한 아키텍처로 구현되어야 합니다.
- 새로운 기능 추가 또는 기존 기능 수정 시 코드 변경 범위를 최소화하여 유지보수 용이성을 높여야 합니다.
- 서버 부하 분산과 수평적 확장을 고려하여 시스템이 확장 가능하도록 설계되어야 합니다.

5.5. 번역 기능 요구사항

- 애플리케이션은 한국어(kr)와 영어(en)를 지원하는 언어 번역 기능을 제공해야 합니다.
- 사용자가 언어를 선택할 수 있도록 언어 변경 옵션을 제공해야 합니다.

5.6. 코드 품질 테스트 요구사항

- 코드 품질 향상을 위해 정적 코드 분석, 코드 리뷰 등의 절차를 도입하여 코드 품질을 확보해야 합니다.
- 단위 테스트와 통합 테스트를 구현하여 코드의 기능적인 정확성을 검증해야 합니다.

5.7. 배포 요구사항

- 애플리케이션을 배포할 경우, AWS Lightsail 또는 유사한 클라우드 환경을 활용하여 운영해야 합니다.
- 클라우드 환경에서는 사용자 수에 따라 유동적으로 리소스를 조정할 수 있어야 합니다.
- 배포 후에도 안정적으로 운영되도록 모니터링과 로그 분석 등을 통해 실시간으로 시스템 상태를 체크해야 합니다.

6. 논리 구조 요구사항

