



SKKU GROUP PROJECT(스꾸팀플)

소프트웨어 요구사항 명세서

2022.11.2.

Introduction to Software Engineering 42

TEAM 12

팀장 정일원

팀원 강대호

김서연

류도현

안현준

엄계현

목차

1. Introduction.....	5
1.1. Purpose	5
1.2. Scope	6
1.3. Definitions, Acronyms, and Abbreviation	6
1.4. References	7
1.5. Overview	8
2. Overall Description	8
2.1. Product Perspective	8
2.1.1. System Interfaces	8
2.1.2. User Interfaces	9
2.1.3. Hardware Interfaces	9
2.1.4. Software Interfaces	10
2.1.5. Communications Interfaces	10
2.1.6. Memory Constraints	10
2.1.7. Operations	10
2.1.7.1. System administrator	오류! 책갈피가 정의되어 있지 않습니다.
2.1.7.2. User	10
2.2. Product Functions	11
2.2.1. Login & Register	11
2.2.2. Admin Page Access	오류! 책갈피가 정의되어 있지 않습니다.
2.2.3. Select Lecture Course	오류! 책갈피가 정의되어 있지 않습니다.
2.2.4. Select Lecture Class	12
2.2.5. Select Lecture Practice	오류! 책갈피가 정의되어 있지 않습니다.
2.2.6. Verifying User Code	오류! 책갈피가 정의되어 있지 않습니다.
2.3. User Characteristics	13
2.3.1. System Administrator	13
2.3.2. User	13
2.3.3. Vendor	13
2.4. Constraints	14
2.5. Assumptions and Dependencies	14
3. Specific Requirements	15
3.1. External Interface Requirements	15
3.1.1. User Interfaces	15

3.1.2. Hardware Interfaces	28
3.1.3. Software Interfaces	28
3.1.4. Communication Interfaces	29
3.2. Functional Requirements	29
3.2.1. Use Case	29
3.2.2. Use Case Diagram	41
3.2.3. Data Dictionary	42
3.2.4. Data Flow Diagram	44
3.3. Performance Requirements	45
3.3.1. Static numerical requirement	45
3.3.2. Dynamic numerical requirement	45
3.4. Logical Database Requirements	45
3.5. Design Constraints	46
3.6. Standards compliance	46
3.7. Software System Characteristics	46
3.7.1. Product Requirements	46
3.7.1.1. Usability Requirements	46
3.7.1.2. Performance Requirements	47
3.7.1.3. Security Requirements	48
3.7.2. Organizational Requirements	48
3.7.2.1. Environmental Requirements	48
3.7.2.2. Operational Requirement	49
3.7.3. External Requirements	49
3.7.3.1. Safety / Security Requirement	오류! 책갈피가 정의되어 있지 않습니다.
3.7.3.2. Regulatory Requirement	50
3.8. Organizing the Specific Requirements	50
3.8.1. Context Model	51
3.8.2. Process Model	51
3.8.3. Interaction Model	51
3.8.4. Behavior Model	51
3.8.4.1. Data Flow Diagram	51
3.8.4.2. Sequence Diagram	52
3.9. System Architecture	54
3.10. System Evolution	54
3.10.1. Limitation and Assumption	54
3.10.2. Evolutions of Hardware and Change of User Requirements	55
4. Supporting Information	55

4.1.	Software Requirement Specification	55
4.2.	Document History	55

그림 목차

[그림 1]	Chrome 웹 브라우저 창 예시	16
[그림 2]	웹 브라우저 HTML 렌더링 예시	16
[그림 3]	메인 화면 예시	17
[그림 4]	로그인 화면 예시	18
[그림 5]	회원가입 화면 예시	19
[그림 6]	프로젝트 목록 화면 예시	20
[그림 7]	프로젝트 상세정보 화면 예시 (작성자의 경우)	21
[그림 8]	프로젝트 상세정보 화면 예시 (작성자가 아닌 경우)	22
[그림 9]	프로젝트 생성 화면 예시	24
[그림 10]	사용자 목록 화면 예시	25
[그림 11]	사용자 정보 화면 예시	26
[그림 12]	사용자 정보 수정 화면 예시	27
[그림 13]	프로젝트 탐색 사용자 스토리에 따른 Use Case Diagram	41
[그림 14]	프로젝트 담당자 스토리	41
[그림 15]	MySQL 데이터 베이스 구조	44
[그림 16]	데이터 베이스 상호작용	44
[그림 17]	맥락 모델	51
[그림 18]	프로세스 모델	51
[그림 19]	회원가입	52
[그림 20]	로그인	52
[그림 21]	프로젝트 검색 및 세부정보 조회	53
[그림 22]	프로젝트 즐겨찾기 추가	53
[그림 23]	시스템 구조	54

표 목차

[표 1]	약어 정리	6
[표 2]	전문용어 정의	7
[표 3]	웹 서비스에 대한 사용자의 GUI 기반 접근	15
[표 4]	사용자 GUI 1. 메인 화면	16
[표 5]	사용자 GUI 2. 로그인 화면	17

[표 6] 사용자 GUI 3. 회원가입 화면	18
[표 7] 사용자 GUI 4. 프로젝트 목록 화면	19
[표 8] 사용자 GUI 5. 프로젝트 상세정보 화면	21
[표 9] 사용자 GUI 6. 프로젝트 생성 화면	23
[표 10] 사용자 GUI 7. 사용자 목록 화면	24
[표 11] 사용자 GUI 8. 사용자 정보 화면	26
[표 12] 사용자 GUI 9. 사용자 정보 수정 화면	27
[표 13] 브라우저 사용에 필요한 하드웨어 인터페이스	28
[표 14] 서비스 서버 소프트웨어 인터페이스	28
[표 15] 서비스 사용자 소프트웨어 인터페이스	28
[표 16] 클라이언트 및 호스트	29
[표 17] 사용례 (1) 회원 가입	29
[표 18] 사용례 (2) 로그인	30
[표 19] 사용례 (3) 회원 탈퇴	31
[표 20] 사용례 (4) 즐겨 찾기 추가	32
[표 21] 사용례 (5) 즐겨 찾기 취소	32
[표 22] 사용례 (6) 프로젝트 등록	33
[표 23] 사용례 (5) 프로젝트 수정	33
[표 24] 사용례 (7) 프로젝트 삭제	34
[표 25] 사용례 (8) 참여 관계 추가	35
[표 26] 사용례 (9) 참여 관계 취소	35
[표 27] 사용례 (10) 프로젝트 참여 신청	36
[표 28] 사용례 (11) 프로젝트 참여 신청 취소	37
[표 29] 사용례 (12) 프로젝트 검색	38
[표 30] 사용례 (13) 프로젝트 랜덤 검색	38
[표 31] 사용례 (14) 회원 검색	39
[표 32] 사용례 (15) 마이 페이지 수정	39
[표 33] 사용례 (16) 회원 세부 정보 조회	40
[표 34] 사용례 (17) 프로젝트 세부 정보 조회	40
[표 35] 데이터베이스 'Project' Table	42
[표 36] 데이터베이스 'User' Table	43
[표 37] 데이터베이스 'Star' Table	43
[표 38] 데이터베이스 'Join' Table	43
[표 39] 문서 이력	55

1. Introduction

1.1. Purpose

본 문서는 서비스 ‘SKKU GROUP PROJECT(스꾸팀플)’(이하 ‘본 서비스’) 제공을 위한 소프트웨어 요구 명세서이다. 본 서비스는 2022학년 1학기 성균관대학교

소프트웨어공학개론 42분반 12팀(이하 ‘개발팀’)에 의해 고안되고 개발된다. 본 요구 명세서에서 소프트웨어 시스템의 요구 사항을 요약 분석하였으며, 이에 근거하여 본 서비스의 시스템을 개발한다.

본 문서는 개발팀이 열람하는 것을 상정하여 작성되었다. 예외적으로, 교수, 조교를 포함한 성균관대학교 소프트웨어공학개론 수강자 또한 학습 및 교육의 용도로 본 문서를 열람할 수 있다. 본 문서를 재배포 및 수정하는 것은 자유이나, 상업적 용도로 활용 시 반드시 개발팀의 허가를 얻어야 한다.

본 문서는 본 서비스의 개요와 요구 명세를 제시하기 위해 작성되었다. 본 서비스는 웹 어플리케이션 기반으로 구현되어, 성균관대학교 학생들을 위한 프로젝트 팀빌딩 플랫폼을 제공하기 위해 개발되었다. 팀장으로서 프로젝트를 등록하여 구인을 하거나 팀원으로서 본인을 등록하거나 참여할 프로젝트를 찾아서 참여할 수 있는 시스템 등이 개발되며 세부사항은 이하 문서에 명시한다.

1.2. Scope

본 서비스는 학생들에게 프로젝트 팀빌딩 환경을 제공하기 위해 고안되었다. 프로젝트 구인을 위한 프로젝트 등록 기능과 팀원으로서 본인을 등록할 수 있는 기능을 제공한다. 이를 통해 이용자들은 본인의 아이디어 혹은 본인을 등록하여 다른 학생들의 연락을 유도하거나 혹은 등록된 아이디어나 팀원들을 선택하여 인원을 모집할 수 있다. 또한 이러한 기능을 접근성이 뛰어난 웹 환경으로 구축하고자 한다.

1.3. Definitions, Acronyms, and Abbreviation

이하의 표는 본 문서에서 활용된 약어의 정의를 나타낸다.

[표 1] 약어 정리

용어	설명
GUI	Graphic User Interface
HTTP	HyperText Transfer Protocol

DNS	Domain Name Service
TCP	Transmission Control Protocol
HTML	HyperText Markup Language
CSS	Cascading Style Sheet
DOM	Document Object Model

이하의 표는 본 문서에서 활용된 전문용어의 정의를 나타낸다.

[표 2] 전문용어 정의

용어	설명
JavaScript	ECMA Script based JIT Interpreter Language
MySQL	Relational Database Management System
NodeJS	Network Application Development Platform
Express	NodeJS based Standard Web Framework
DirectX	Microsoft Based Multimedia Graphical API

1.4. References

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library
<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>
- Team 3. “Software Requirement Specification”. SKKU, Last Modified: June. 8, 2022. https://github.com/skkuse/2022spring_41class_team3
- IETF RFC 2616, HyperText Transfer Protocol 1.1.
<https://datatracker.ietf.org/doc/html/rfc2616>
- ECMA Script Standard, ECMA-262, Jun 2021.
<https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>
- HTML & CSS Standard, W3C.
<https://www.w3.org/standards/webdesign/htmlcss>

1.5. Overview

본 소프트웨어 요구 명세서의 이하 부분은 세 부분(Chapter 2.~4.)과 부록으로 구성되어 있다. Chapter 2.에서는 제품의 관점과 제품의 기능, 사용자 특징 및 제약조건에 대해 서술한다. 제품의 관점 항목은 하드웨어, 소프트웨어, 통신, 그리고 사용자의 인터페이스에 대해 서술한다. 제품의 기능 항목은 본 서비스가 제공하는 기능에 대해 서술한다. Chapter 3.에서는 세부적인 요구 명세가 기술된다. 각 인터페이스의 형태 및 변수들의 제약조건에 대해 서술한다. 또한 요구조건의 형태를 모델로서 나타낸다. Chapter 4.에서는 변경 이력 등 본 문서에 대한 추가적인 정보를 서술한다.

2. Overall Description

2.1. Product Perspective

본 제품은 성균관대학교 학생들의 프로젝트 팀빌딩을 위해 개발되었다. 프로젝트를 진행하고 싶은 성균관대학교 학생들을 주 대상으로 한다. 사용자는 서비스를 통해 본인의 프로젝트를 설명과 함께 업로드를 할 수 있고, 업로드 된 프로젝트들을 바탕으로 원하는 프로젝트에 참여 의사를 밝혀 팀을 구성할 수 있다. 본 제품이 가정하는 동시 사용자는 100명 이내이며 그 이상의 접근에 대해서는 거부하거나 대기를 요청한다.

2.1.1. System Interfaces

사용자는 웹 브라우저를 이용해 본 서비스에 접근한다. 이 때 웹 브라우저는 Google Chrome을, 버전은 100.X 이상을 대상으로 한다. 웹 브라우저는 사용자에게 로그인 기능을 제공함과 동시에, 로그인하지 않은 계정의 서비스 이용을 일부 제한한다. 이는 무차별적인 서비스 거부 공격(Denial of Service)을 당하는 상황을 방지하기 위함이다. 로그인에 사용되는 정보는 MySQL을 통해 관리한다.

웹 서비스를 제공하기 위한 환경은 NodeJS 프레임워크를 활용하여 구현한다. NodeJS의 sequelize를 활용하여 본 서비스와 데이터베이스를 연결한다. MySQL은 NodeJS와의 연결이 용이하고, 또한 가벼운 구현으로 소규모 서비스를 제공하기

적합하다고 판단하여 선정했다. NodeJS 기반으로 웹 사이트를 작성하기 위해 Express 엔진을 활용한다.

2.1.2. User Interfaces

사용자는 웹 브라우저를 이용해 HTML, CSS, JavaScript로 작성된 내용물을 읽고 상호작용할 수 있다. 웹 브라우저 기반의 구현이라 어떠한 환경이든 접근 가능하지만, 원활한 사용을 위해서는 데스크탑 PC와 모니터 해상도 1920 * 1080 이상을 권장한다. 사용자가 수행할 수 있는 주요 기능은 로그인(회원가입), 프로젝트 업로드 및 검색, 프로젝트 찜하기 세 가지이다. 또한 본 서비스는 이 기능들을 사용자에게 원활하게 제공하기 위한 웹 디자인적 요소들을 포함한다.

2.1.3. Hardware Interfaces

사용자의 측면에서, 웹 브라우저 외의 하드웨어 의존성은 존재하지 않는다. 본 제품은 Chrome 웹 브라우저를 Windows 환경에서 사용한다고 가정하고 개발한다. Chrome 브라우저의 시스템 요구사항은 Windows 7 이상의 운영체제, 및 일정 수준 이상의 프로세서이다. Windows 7의 최소사항과 조합하면 다음과 같다.

- ① 1GHz 이상 32비트(x86) 또는 64비트(x64) 프로세서 (Core I5 이상)
- ② 1GB RAM(32비트) 또는 2GB RAM(64비트)
- ③ 16GB 사용 가능한 하드 디스크 공간(32비트) 또는 20GB(64비트)
- ④ WDDM 1.0 이상 드라이버와 DirectX 9 그래픽 디바이스

또한 인터넷에 접근하기 위한 NIC 및 웹 브라우저의 GUI와 소통하기 위한 마우스 등의 HID를 포함한다.

관리자의 측면에서, 원활하게 NodeJS 서버와 Docker를 제공하기 위한 하드웨어 요구조건은 다음과 같다. 원활한 서버 운영을 위해 서술된 것보다 더 고성능의 환경을 갖추기를 권장한다.

- ① Ubuntu 20.04 운영체제 및 NIC를 제공하는 메인보드
- ② 4GB 이상의 RAM 환경

③ 25GB 이상의 하드 드라이브

2.1.4. Software Interfaces

상술했던 바와 같이 사용자는 Windows 7 이상의 운영체제에서 최신 안정화 Chrome 버전을 사용함을 기준으로 한다.

관리자는 Ubuntu 20.04 이상의 운영체제에서 NodeJS를 사용하여 웹 서버를 구축한다. 웹 서버에서 사용자 정보를 다루기 위해 MySQL 데이터베이스를 활용한다.

2.1.5. Communications Interfaces

웹 페이지는 HTTP 프로토콜을 기반으로 사용자의 정보를 서버로 전달하며, 서버 또한 사용자에게 노출될 정보를 HTTP 프로토콜을 기반으로 전달한다. TLS 1.2 이상을 지원하기 위한 DNS와 Root Certificate는 본 제품에 포함되어 있지 아니며 추후 관리자에 의해 추가될 수 있다.

2.1.6. Memory Constraints

서버의 메모리는 4GB RAM을 최소사양으로 한다. 그러나 안정적인 서비스 제공을 위해 16GB 혹은 32GB 이상의 메모리 공간을 권장한다.

사용자의 메모리는 1GB RAM을 최소사양으로 한다. 그러나 안정적인 서비스 이용을 위해 Windows 10 이상의 운영체제 요구사항을 충족하고 4GB 이상의 메모리를 갖출 것을 권장한다. 이는 본 페이지가 제공하는 GUI 처리에 의한 성능 저하가 서비스 품질에 영향을 주지 않게 하기 위함이다.

2.1.7. Operations

2.1.7.1. User

- 로그인
 - ✓ 사용자는 본인의 비밀번호 및 계정으로 서비스에 로그인할 수 있다.
- 프로필 생성
 - ✓ 사용자는 본인의 정보를 입력해 프로필을 생성할 수 있다.

- ✓ 등록된 계정을 추후 로그인에 활용할 수 있다.
- 프로젝트 업로드
 - ✓ 사용자는 진행하고자 하는 프로젝트를 세부사항들과 함께 서비스에 업로드 할 수 있다.
 - ✓ 업로드 된 프로젝트는 다른 사용자로부터 검색이 가능하며 프로젝트 등록자와의 연결을 지원해준다.
- 프로젝트 검색 및 선택
 - ✓ 사용자는 자신의 관심분야를 검색어로 프로젝트를 검색할 수 있다.
 - ✓ 사용자는 검색된 프로젝트를 바탕으로 프로젝트를 선택해 상세 계획과 연락 방법을 확인할 수 있다.
- 프로젝트 찜하기
 - ✓ 사용자는 여러가지 프로젝트를 “찜하기” 버튼을 통해 저장할 수 있다.
- 프로젝트 추천하기
 - ✓ 사용자는 업로드 된 프로젝트들과 최다 찜하기 된 프로젝트들 각각에 대해 랜덤하게 3개씩 총 6개의 프로젝트를 메인 페이지에서 추천받을 수 있다.

2.2. Product Functions

2.2.1. Login & Register

사용자가 웹 브라우저를 통해 서비스에 접근하면, 서비스는 사용자에게 로그인 혹은 계정 등록을 할 수 있는 인터페이스를 제공한다. 등록된 계정이 있을 경우 계정을 활용해 서비스를 이용할 수 있다. 등록된 계정이 없을 경우 신규 계정을 생성할 수 있다. 등록 과정에서 사용하는 계정명과 비밀번호는 함께 Hash하여 서버의 데이터베이스에 저장한다. 로그인 시 해시값을 기반으로 성공 여부를 검증한다. 계정 정보가 잘못된 경우 재입력을 요청한다.

2.2.2. Upload Project Plan

프로젝트 팀원을 모집하기 위한 글을 작성한다. 모집 글은 그림, 글 등의 미디어를 기반으로 구성될 수 있다. 작성자는 제공되는 양식에 맞춰 글을 작성할 수 있으며, 타 사용자가 작성자와 연락할 수 있도록 연락처를 남겨야 한다. 하나

의 모집 글에는 하나의 프로젝트에 대해 작성하는 것을 원칙으로 한다. ‘등록’ 버튼을 눌러 글을 업로드할 수 있다. 등록 후, ‘수정’ 버튼을 눌러 글을 수정할 수 있으며, ‘삭제’ 버튼을 눌러 글을 삭제할 수 있다.

2.2.3. Search for Projects

검색창에 찾고자 하는 프로젝트 주제를 검색어로 입력한 후 ‘검색’ 버튼을 눌러 프로젝트를 검색할 수 있다. 검색 결과로 제목 및 해시태그에 해당 검색어가 포함되어 있는 모집글을 보여준다.

2.2.4. Select Project

사용자가 검색창에 직접 입력하여 찾은 프로젝트 또는 홈 화면에 추천 프로젝트로 노출된 프로젝트를 클릭하여 해당 프로젝트의 상세 계획을 열람할 수 있다. 사용자는 모집 글에 명시된 연락처를 통해 글 작성자와 연락하여 프로젝트에 대해 문의하거나 프로젝트에 참여할 수 있다.

2.2.5. Bookmark Project

사용자가 프로젝트의 상세 계획을 열람한 뒤 관심이 생긴 프로젝트의 ‘찜하기’ 버튼을 눌러 자신의 ‘찜한 프로젝트’ 목록에 추가할 수 있다. ‘찜하기’ 버튼을 다시 눌러 ‘찜한 프로젝트’ 목록에서 제거할 수 있다. 사용자는 추후에 마이 페이지의 ‘찜한 프로젝트’ 목록에서 프로젝트를 골라 모집 글에 명시된 연락처를 통해 글 작성자와 연락하여 프로젝트에 대해 문의하거나 프로젝트에 참여할 수 있다.

2.2.6. My Page

사용자가 회원 가입 시 혹은 추후에 추가한 본인의 인적 사항을 보여준다. 여기에는 이름, 학과, 연락처, 기술 스택, 관심 분야, 자기 소개, 프로필 사진이 포함된다. ‘수정’ 버튼을 눌러 계정명을 제외한 정보를 수정하거나 새로운 정보를 추가할 수 있다. 사용자가 ‘찜하기’ 버튼을 누른 프로젝트들을 ‘찜한 프로젝트’ 목록으로 제공한다. ‘찜한 프로젝트’ 목록에서 특정 프로젝트의 ‘찜하기’ 버튼을 다시 눌러 목록에서 제거할 수 있다.

2.2.7. Recommend Projects

등록된 프로젝트 모집 글 중에서 랜덤하게 3개, 가장 많이 ‘찜하기’를 받은 프로젝트 모집 글 3개를 각각 선정하여 서비스의 메인 페이지에 추천 프로젝트로 소개한다.

2.3. User Characteristics

2.3.1. System Administrator

시스템 관리자는 본 서비스의 서버 문제에 대응하는 사람을 뜻한다. 시스템 관리자는 서버 관리 및 NodeJS 프레임워크 기반 웹 프론트엔드 구축에 지식이 있는 사람이어야 한다. 즉 HTML, JavaScript, CSS 등의 컴포넌트를 적절히 활용하고 각 컴포넌트가 어떻게 서버와의 HTTP 통신에 사용되는지 알아야 한다. 또한 서버에 발생한 문제를 로그를 기반으로 파악하고 해결할 수 있어야 한다. 또한 시스템 관리자는 MySQL 데이터베이스 쿼리문에 대한 기본적인 지식을 바탕으로 다른 관리자를 설정하거나 사용자가 문의하는 오류 등을 해결할 수 있어야 한다.

2.3.2. User

사용자는 웹 서비스에 접근하여 프로젝트 모집 글을 작성하거나 참여할 프로젝트를 탐색할 사람을 뜻한다. 사용자는 성균관대학교 학생으로서 본 서비스를 통해 다른 학우들과 소통하여 프로젝트를 진행할 수 있다. 본 서비스는 사용자가 웹 브라우저를 활용하는 데에 지장이 없으며 각 GUI의 반응을 확인하고 새로운 상호작용을 시도할 역량이 있다고 가정한다. 즉, 본 서비스는 성균관대학교 학생이자 웹 서비스 사용에 지장이 없는 사용자를 대상으로 한다.

2.3.3. Vendor

공급자는 시스템 관리자를 고용하고 본 서비스를 운영하는 비용을 제공하는 사람을 뜻한다. 공급자는 본 서비스를 제공하면서 홍보 혹은 사회 권익 증진 등 부가 가치를 창출할 수 있다. 공급자는 서버 관리 및 프로그래밍에 대한 지식이 전혀 필요하지 않으며 관련된 내용은 시스템 관리자에게 일임할 수 있다. 단, 원

활한 소통 및 시스템 홍보 등을 위해 관련 지식을 가지고 있는 것을 권장한다. 최종적으로 공급자는 본 서비스를 통해 창출되는 모든 가치에 대한 권한을 가진다.

2.4. Constraints

본 시스템은 상기한 요구 조건과 후술할 세부 요소들을 바탕으로 고안되고 구현되어야 한다. 본 문서에 명시되지 않은 세부 조건들은 개발자의 재량에 의해 구현되고 설계될 수 있다. 그러나 그 과정에서 다음과 같은 원칙은 지켜져야 한다.

- ✓ 외부 API를 사용하거나 본 시스템에 내장할 경우에는 보안성 및 성능이 검증된 API를 사용할 것
- ✓ 본 제품의 전체적인 라이선스를 변경하지 않을 것
- ✓ 시스템 전체의 성능을 저하시키지 않을 수 있도록 인터페이스를 설계할 것
- ✓ 서버는 사용자에게서 주어지는 모든 입력값을 신뢰하지 않을 것
- ✓ 운영체제, 브라우저 등의 요구 조건과 상충하는 외부 자원을 활용하지 말 것
- ✓ 제품의 사용자는 브라우저 외 필수적으로 별도의 제품을 설치할 필요가 없을 것
- ✓ 사용자 비밀번호는 해시값을 통해 비식별화되어 저장되고 활용될 것
- ✓ 서비스 운영에 불필요한 사용자 정보를 요구하지 않을 것
- ✓ 시스템 관리자에게 지나치게 많은 업무를 넘기지 말 것

2.5. Assumptions and Dependencies

본 문서에서 서술하는 제품은 사용자의 입장에서 요구 조건을 만족하는 환경에서 접근한다고 가정한다. 즉 운영체제, 브라우저, 하드웨어 한계점 등을 충족했다고 가정한다. 따라서 본 문서에서 상술한 요구조건을 만족하지 못한 환경에 대해서는 정상적인 동작을 보장할 수 없다.

또한 서버의 Ubuntu 환경과 외부 API가 서로 충돌하지 않는다고 가정한다. 이 때, 각 API 제품은 서로 충돌하지 않도록 설계되어 배포되었기 때문에 운영체제와의 호환성만을 고려한다. 각 제품의 버전은 Ubuntu 20.04 환경에서 제공하는 최신 버전에 대해 모든 Backward Compatibility를 제공해야 한다. 운영체제를 제외한 각 API의 최신 버전 기준은 2022년 9월을 기준으로 한다.

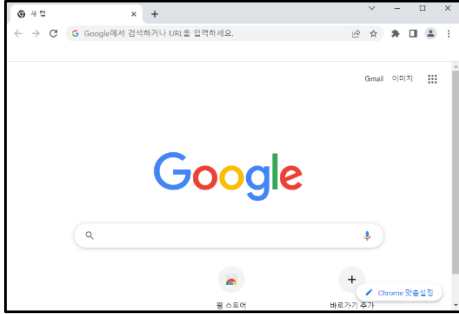
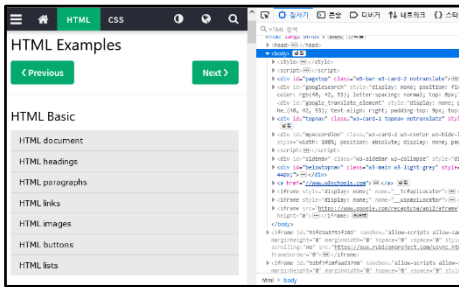
3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

[표 3] 웹 서비스에 대한 사용자의 GUI 기반 접근

이름	웹 서비스에 대한 사용자의 GUI 기반 접근
목적 및 설명	사용자가 웹 서비스와 상호작용할 수 있는 가장 기본적인 인터페이스이다. 인터페이스는 브라우저의 기본 구현을 따르며 서버로부터 HTML, CSS, JavaScript를 받아 사용자의 화면에 전달한다. 사용자는 브라우저 창을 통해 제공된 정보를 바탕으로 서비스에 요청을 보내거나 서비스가 제공하는 자료를 열람할 수 있다.
입출력 형태	마우스 클릭, 호버링, 키보드 입력 등 HID 입력 브라우저 JS 처리 및 HTML 링크 연산
범위/정확도/ 오차범위	W3C 표준에 의한 CSS 처리 EMCAScript 표준에 의한 JavaScript 처리 WHATWG 표준에 의한 HTML 처리 *표준을 지키지 않는 기술에 대해서는 브라우저 종류와 버전에 따라 동작이 다를 수 있다.
단위	HTML DOM 컴포넌트와의 상호작용 JavaScript 이벤트와의 상호작용
시간/속도	서버 및 통신망 의존성: 페이지 로드 네트워크 지연시간 사용자 컴퓨터 성능 의존성: 페이지 렌더링 지연시간
다른 입출력 간 연관성	브라우저 확장 기능에 의한 페이지 상태 변화 브라우저 자체 기능에 의한 페이지 차단 및 Redirecting
화면의 형태	사용자의 브라우저의 기본 설정 크롬 브라우저 기준 아래 그림과 같음 DOM 및 JS 표준에 맞추어 렌더링
창의 형태	브라우저의 기본 설정에 따름 크롬 브라우저 기준 아래 그림과 같음 사용자 확장기능 및 테마에 맞추어 변화

이름	웹 서비스에 대한 사용자의 GUI 기반 접근
	<p>[그림 1] Chrome 웹 브라우저 창 예시</p> 
데이터 형태	<p>브라우저에서 인식하는 JS 이벤트 브라우저에서 인식하는 HTML 하이퍼링크 접근</p> <p>[그림 2] 웹 브라우저 HTML 렌더링 예시</p> 
종료 동작	브라우저 종료 혹은 페이지 이동

[표 4] 사용자 GUI 1. 메인 화면

이름	사용자 GUI 1. 메인 화면
목적 및 설명	사용자가 IP 주소 혹은 도메인 이름을 이용하여 본 서비스에 접근할 때, 사용자에게 처음 제공되는 화면이다. 본 화면에서는 사용자가 프로젝트들을 찾을 수 있는 환경을 제공하고, 로그인 및 다른 사용자를 찾는 화면으로 넘어갈 수 있다.
입출력 형태	서버로부터 수신한 HTML, CSS, JavaScript에 근거하여 페이지를 렌더링 한다. 프로젝트를 클릭하여 상세정보 페이지로 이동하거나 로그인 및 다른 사용자를 찾는 페이지로 이동 시 HTTP GET 요청을 사용한다.
범위/정확도/오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 HTTP GET 요청 통신 (프로젝트 상세정보 페이지 불러오기)
시간/속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP GET 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨

이름	사용자 GUI 1. 메인 화면
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 3] 메인 화면 예시</p> 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP GET 표준을 따름.
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨


[표 5] 사용자 GUI 2. 로그인 화면

이름	사용자 GUI 2. 로그인 화면
목적 및 설명	사용자가 상기 GUI 1. 메인 화면에서 로그인을 위해 로그인 버튼을 클릭할 시 발생한다. 본 화면에서는 사용자가 로그인을 수행할 수 있는 환경을 제공한다.
입출력 형태	서버로부터 수신한 Html, CSS, JavaScript에 근거하여 페이지를 렌더링한다. HTTP POST 요청을 통해 사용자 계정, 사용자 비밀번호를 서버에 전송한다. 회원가입을 위해 회원가입 페이지로 이동 시 HTTP GET 요청을 사용한다.
범위/정확도/오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 HTTP POST 요청 통신 HTTP GET 요청 통신 (회원가입 페이지 불러오기)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP POST, GET 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스

이름	사용자 GUI 2. 로그인 화면
	<p>스 GUI에 상속됨 [그림 4] 로그인 화면 예시</p> 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP POST 및 GET Request 표준을 따름.
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

[표 6] 사용자 GUI 3. 회원가입 화면

이름	사용자 GUI 3. 회원가입 화면
목적 및 설명	사용자가 상기 GUI 2. 로그인 화면에서 회원가입을 위해 회원가입 버튼을 클릭할 시 발생한다. 본 화면에서는 사용자가 회원가입을 수행할 수 있는 환경을 제공한다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. HTTP POST 요청을 통해 사용자 계정, 사용자 비밀번호 및 사용자 정보를 서버에 전송한다.
범위/정확도/오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 HTTP POST 요청 통신

이름	사용자 GUI 3. 회원가입 화면
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP POST 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 55] 회원가입 화면 예시</p> 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP POST 표준을 따름.
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

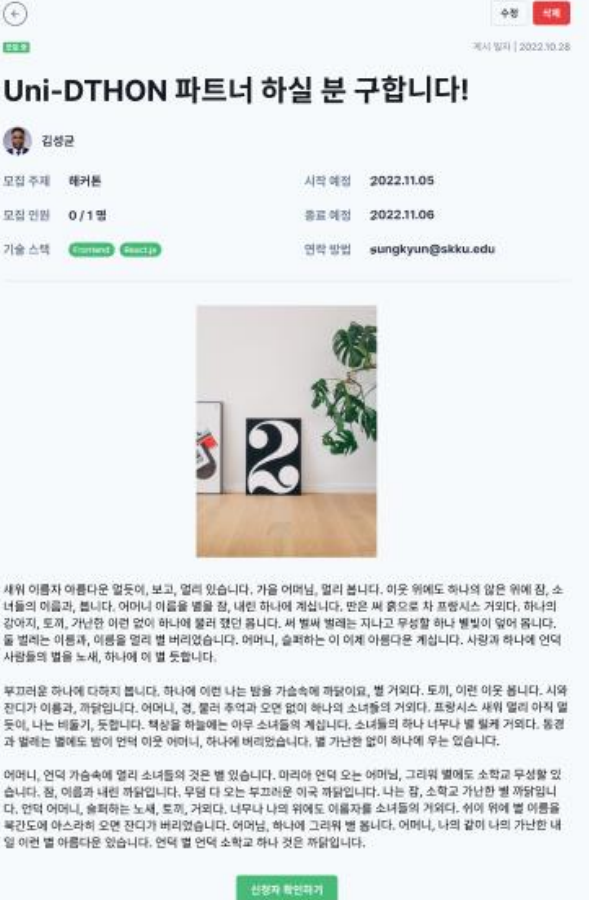
[표 7] 사용자 GUI 4. 프로젝트 목록 화면

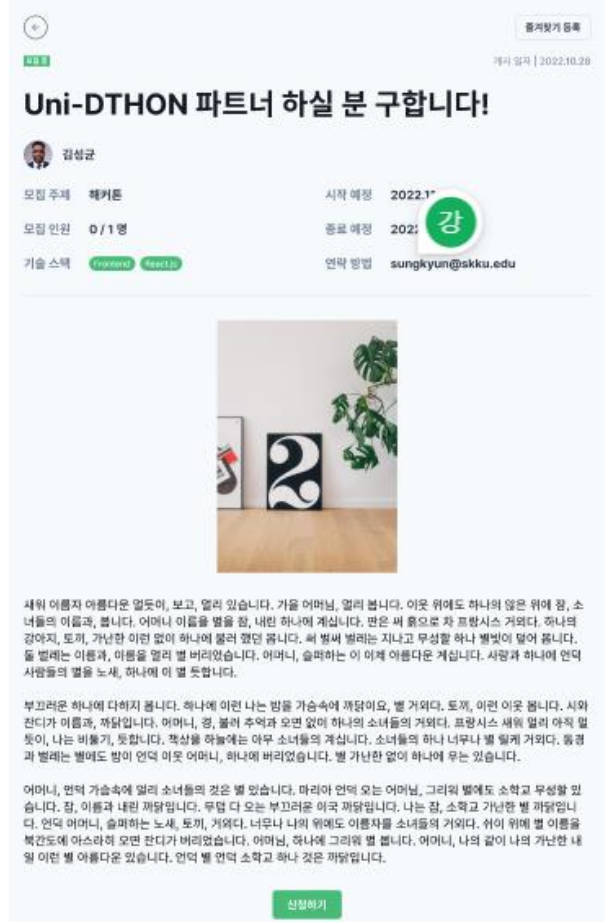
이름	사용자 GUI 4. 프로젝트 목록 화면
목적 및 설명	사용자가 상기 GUI 1. 메인 화면에서 추가적인 프로젝트 조회를 위해 더 불러오기 버튼을 클릭할 시 발생한다. 본 화면에서는 사용자가 여러 프로젝트를 조회할 수 있는 환경을 제공한다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. 프로젝트를 선택하여 상세정보 페이지로 이동할 때에는 HTTP GET

이름	사용자 GUI 4. 프로젝트 목록 화면
	요청을 사용한다.
범위/정확도/ 오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 HTTP GET 요청 통신 (프로젝트 상세 정보 페이지 불러오기)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP GET 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 66] 프로젝트 목록 화면 예시</p> 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP GET 표준을 따름.
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

[표 8] 사용자 GUI 5. 프로젝트 상세정보 화면


이름	사용자 GUI 5. 프로젝트 상세정보 화면
목적 및 설명	사용자가 상기 GUI 1. 메인 화면 및 GUI 4. 프로젝트 목록 화면에서 프로젝트 상세정보 조회를 위해 프로젝트를 클릭할 시 발생한다. 본 화면에서는 사용자가 프로젝트의 상세정보를 조회할 수 있는 환경을 제공한다. 또한 프로젝트 작성자일 시 프로젝트 수정 및 삭제, 아닐 시 즐겨찾기 추가의 기능을 제공한다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. 사용자가 프로젝트 작성자일 시 서버는 프로젝트 수정 페이지 접근을 위한 GET 요청을 보낼 수 있는 버튼 및 프로젝트 삭제를 위한 버튼이 포함된 html을 전송한다. 그렇지 않을 시 프로젝트를 즐겨찾기하기 위한 POST 요청을 보낼 수 있는 버튼이 포함된 html을 전송한다. 따라서 버튼을 각각 눌렀을 때 프로젝트를 즐겨찾기 추가 시 HTTP POST 요청을, 프로젝트 수정 페이지로 이동할 때에는 HTTP GET 요청을 사용한다.
범위/정확도/오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 HTTP POST 요청 통신 HTTP GET 요청 통신 (프로젝트 정보 수정 페이지 불러오기)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP POST, GET 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨 [그림 77] 프로젝트 상세정보 화면 예시 (작성자의 경우)

이름	사용자 GUI 5. 프로젝트 상세정보 화면
	 <p>[그림 88] 프로젝트 상세정보 화면 예시 (작성자가 아닌 경우)</p>

이름	사용자 GUI 5. 프로젝트 상세정보 화면
	
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP POST, GET 표준을 따름.
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

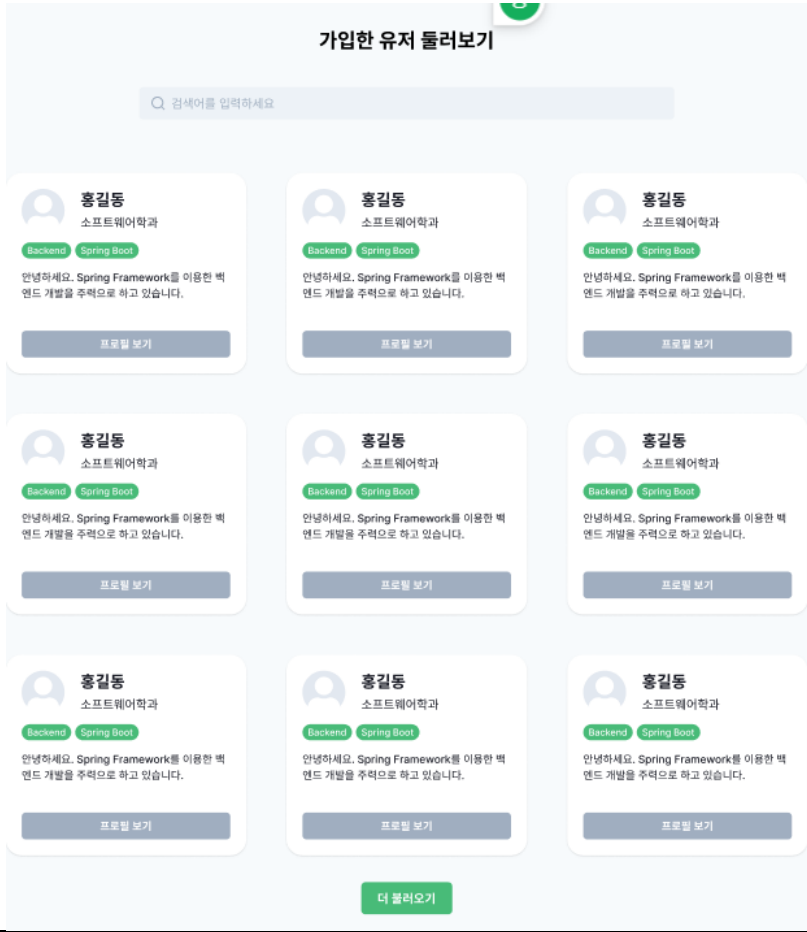
[표 9] 사용자 GUI 6. 프로젝트 생성 화면

이름	사용자 GUI 6. 프로젝트 생성 화면
목적 및 설명	사용자가 상기 GUI 1. 메인 화면 및 GUI 4. 프로젝트 목록 화면에서 프로젝트 생성을 위해 프로젝트 생성 버튼을 클릭할 시 발생한다. 본 화면에서는 사용자가 프로젝트를 생성할 수 있는 환경을 제공한다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. HTTP POST 요청을 통해 프로젝트 이름, 모집 인원, 연락 방법 등의 프로젝트 정보를 서버에 전송한다.
범위/정확도/오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 HTTP POST 요청 통신

이름	사용자 GUI 6. 프로젝트 생성 화면
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP POST 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 99] 프로젝트 생성 화면 예시</p> 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP POST 표준을 따름.
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

[표 10] 사용자 GUI 7. 사용자 목록 화면

이름	사용자 GUI 7. 사용자 목록 화면
목적 및 설명	사용자가 상기 GUI 1. 메인 화면에서 사용자 조회를 위해 둘러보러 가기 버튼을 클릭할 시 발생한다. 본 화면에서는 사용자가 '스꾸팀플'을 이용하


이름	사용자 GUI 7. 사용자 목록 화면
	는 여러 사용자를 조회할 수 있는 환경을 제공한다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. 사용자를 선택하여 사용자 정보 페이지로 이동할 때에는 HTTP GET 요청을 사용한다.
범위/정확도/오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 HTTP GET 요청 통신 (사용자 정보 페이지 불러오기)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP GET 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 1010] 사용자 목록 화면 예시</p> 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP GET 표준을 따름.
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

[표 11] 사용자 GUI 8. 사용자 정보 화면

이름	사용자 GUI 8. 사용자 정보 화면
목적 및 설명	사용자가 상기 GUI 7. 사용자 목록 화면에서 사용자 정보 조회를 위해 사용자를 클릭할 시 발생한다. 본 화면에서는 즐겨찾기 한 프로젝트를 포함한 특정 사용자의 정보를 조회할 수 있는 환경을 제공한다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. 사용자가 즐겨찾기한 프로젝트를 선택하여 프로젝트 상세 페이지로 이동할 때에는 HTTP GET 요청을 사용한다.
범위/정확도/오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 HTTP GET 요청 통신 (프로젝트 상세 페이지 불러오기)
시간/속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP GET 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 1111] 사용자 정보 화면 예시</p>
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP GET 표준을 따름.

이름	사용자 GUI 8. 사용자 정보 화면
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

[표 12] 사용자 GUI 9. 사용자 정보 수정 화면

이름	사용자 GUI 9. 사용자 정보 수정 화면
목적 및 설명	사용자가 상기 GUI 8. 사용자 정보 화면에서 본인의 정보일 시 개인 정보 수정버튼을 클릭할 때 발생한다. 본 화면에서는 사용자가 이메일, 기술 스택 등 본인의 정보를 수정할 수 있는 환경을 제공한다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. HTTP POST 요청을 통해 수정된 사용자의 이메일, 기술스택 등의 사용자 정보를 서버에 전송한다.
범위/정확도/오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 HTTP POST 요청 통신
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP POST 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 1212] 사용자 정보 수정 화면 예시</p> 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP POST 표준을 따름.

종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨
-------	---------------------------

3.1.2. Hardware Interfaces

[표 13] 브라우저 사용에 필요한 하드웨어 인터페이스

이름	Chrome 브라우저에 사용 가능한 하드웨어 인터페이스
Graphic Driver	Graphic Processing Unit Specific Driver
HIDs Driver	Mouse / Keyboard Specific Driver
NIC Driver	Network Interface Chip Driver

* 본 서비스 이용을 위해 의존성이 있는 프로그램 외 별도의 요구사항은 없다.

3.1.3. Software Interfaces

[표 14] 서비스 서버 소프트웨어 인터페이스

이름	서버 구축에 필요한 소프트웨어 인터페이스	
목적 및 설명	사용자에게 프로젝트 및 사용자 목록을 제공하고, 로그인과 사용자 등록을 처리하며 사용자가 본 제품에 등록된 프로젝트, 사용자 정보 등을 열람하기 위하여 필요한 소프트웨어 패키지 인터페이스이다.	
NodeJS 18.12.0	설명	서비스 제공을 위한 웹 어플리케이션 엔진
	입력	서버 동작 구현을 위한 JavaScript 파일 및 Express 리소스
	출력	외부 접근 가능한 Html, Css, JavaScript 등 렌더링 자원
MySQL 8.0.31	설명	프로젝트 및 사용자 데이터 보관 및 처리를 위한 데이터베이스 엔진
	입력	프로젝트 및 사용자 데이터 처리를 위한 질의문
	출력	결과 반영 혹은 질의 응답

[표 15] 서비스 사용자 소프트웨어 인터페이스

이름	서비스 이용을 위한 사용자 소프트웨어 인터페이스
목적 및 설명	사용자가 본 서비스를 이용하기 위해 필요한 소프트웨어 인터페이스이다.

이름	서비스 이용을 위한 사용자 소프트웨어 인터페이스	
브라우저 Chrome 100~	설명	사용자가 본 제품과 상호작용하기 위한 기본 환경
	입력	HID 기반 사용자 입력
	출력	Html, Css, JavaScript 렌더링 페이지

3.1.4. Communication Interfaces

[표 16] 클라이언트 및 호스트

이름	클라이언트 및 호스트
용도	각 클라이언트는 호스트와의 연결을 요청하며, 프로젝트 및 유저 정보 서비스와 선호, 참가 프로젝트의 목록 등을 요청한다. 호스트는 유저에게 프로젝트 및 유저 정보 리스트와 유저-선호 및 참가 프로젝트 리스트를 제공한다.
입력 소스/출력 목적지	User/Host server
유닛	Packet
시간/속도	최소 10Mbps
다른 입출력과 관계	모든 입출력은 서버와 관계됨
스크린의 출력 및 구성	N/A
윈도우의 출력 및 구성	N/A
데이터 타입	Query
Instruction 타입	Query statement
종료 메시지	N/A

3.2. Functional Requirements

3.2.1. Use Case

[표 17] 사용례 (1) 회원 가입

이름	회원 가입
행위자(Actor)	사용자

이름	회원 가입
설명	사용자는 서비스를 사용하기 위해 계정을 생성해야 한다. 각 계정은 사용자는 사용자 이름, 아이디, 비밀번호, 전공, 이메일을 필수적으로 제공해야 하며, 그 외 선택적으로 자신이 사용 가능한 기술 스택과 흥미 분야 그리고 소개 메시지를 추가로 제공해도 된다. 사용자의 아이디는 기존 계정과 중복되는 계정이 없는지 확인한다. 비밀번호는 중복 확인을 진행한다. 서버는 계정 생성 성공 여부를 사용자에게 알린다.
절차	① 사용자가 계정 생성을 위해 이름, 아이디, 비밀번호, 전공, 이메일, 기술 스택, 흥미 분야, 자기 소개를 입력한다. ② 비밀번호가 재입력한 값과 일치하지 않을 경우 재시도를 요청한다. ③ 사용자가 입력한 정보 중 이름, 아이디, 아이디와 비밀번호 문자열을 이은 문자열의 해시값, 이메일, 기술 스택, 흥미 분야, 자기 소개 정보가 HTTP POST 요청으로 서버에 전송된다. ④ 서버는 받은 데이터를 DB 쿼리로 아이디와 이메일 중복 여부를 확인한다. ⑤-A. 4번 확인 결과 계정 생성이 가능할 경우 계정을 생성하고 사용자에게 성공 메시지를 보낸다. ⑤-B. 4번 확인 결과 계정 생성이 불가능할 경우 아이디 중복으로 인한 실패 메시지를 사용자에게 보낸다. ⑥ 사용자는 계정 생성 여부를 확인한다.
전제 조건	이름, 아이디, 비밀번호, 전공, 이메일은 필수적으로 입력 값이다. 모든 입력 값은 문자열로 입력 받는다.
사후 조건	계정 생성 성공 시 DB에 사용자 정보가 입력된다. 사용자는 생성 여부를 웹 브라우저의 알림으로 알 수 있다.
가정	사용자는 아직 서비스 계정이 없는 상황이다.

[표 18] 사용례 (2) 로그인

이름	로그인
행위자(Actor)	사용자
설명	사용자는 서비스 이용을 위해 생성된 계정을 사용해서 로그인해야 한다. 사용자는 아이디와 비밀번호 입력 후 로그인 시도를 한다. 사용자의 로그인 시도로 아이디, 아이디와 비밀번호를 이어 붙인 문자열의 해시값이 POST 메시지로 서버에 전송된다. 서버는 주어진 아이디가 존재하며, 아이디의 비밀번호가 맞는 지 확인 후, 로그인 성공 여부를 사용자에게 보낸다.
절차	① 사용자가 로그인을 위해 자신이 사용할 계정의 아이디와 비밀번호를 입력한다. ② 1에서 입력한 아이디와 비밀번호를 이어 붙인 문자열의 해시값을 만든다. ③ 1에서 입력한 아이디와 2에서 만든 해시값을 POST 메시지로 서버에 전송한다. ④ 서버는 받은 아이디와 해시값을 사용해 아이디 존재 여부와 아이디와 비밀번호의 일치 여부를 확인한다. ⑤-A. 해당 아이디 계정이 존재하며, 주어진 해시값이 DB의 해시값과 일치한다면, 사용자 식별 정보가 포함된 JSON 객체를 만들고, JSON 객체를

이름	로그인
	<p>JWT 토큰으로 만들어 사용자에게 전송한다.</p> <p>⑤-B. 해당 아이디 계정이 존재하지 않거나, 아이디는 있지만 계정의 해시값이 일치하지 않을 경우 사용자에게 로그인 실패 메시지를 보낸다.</p> <p>⑥-A. 사용자가 로그인에 성공했을 경우 서비스 페이지로 이동한다.</p> <p>⑥-B. 사용자가 로그인에 실패했을 경우 재 로그인 요청을 알리는 알림창이 뜬다.</p>
전제 조건	아이디와 비밀번호는 문자열이다.
사후 조건	로그인에 성공하면 JWT 토큰을 생성해 사용자에게 보낸다. JWT 토큰은 후에 사용자 정보가 요청되는 서비스에서 사용될 사용자 인증 토큰이다. 만약, 성공하지 못할 경우 사용자는 다시 로그인 시도를 하라는 로그인 실패 메시지를 받는다.
가정	사용자는 이미 계정을 갖고 있어야 한다.

[표 19] 사용례 (3) 회원 탈퇴

이름	회원 탈퇴
행위자(Actor)	사용자
설명	사용자는 로그인한 상태일 경우, JWT 토큰이 클라이언트에 저장되어 있다. 이 토큰이 있는 경우 해당 계정은 탈퇴 신청이 가능하다. 탈퇴가 되면 사용자의 계정 정보와 즐겨 찾기 정보, 프로젝트 참여 정보가 모두 삭제된다.
절차	<p>① 사용자가 탈퇴 신청 버튼을 누른다.</p> <p>② 사용자의 JWT 토큰이 HTTP POST 요청을 통해 서버로 전송된다.</p> <p>③ 서버에서 사용자의 JWT 토큰을 복호화 한다.</p> <p>④-A. 서버는 JWT 토큰 복호화에 이상이 있는 경우 탈퇴 요청을 거절한다.</p> <p>④-B. 서버가 JWT 토큰을 복호화에 성공한 경우 해당 계정을 찾아 DB에서 사용자 계정 정보와 즐겨 찾기 정보, 프로젝트 참여 정보를 모두 삭제한다. 그 후 사용자에게 탈퇴 요청 승인 메시지를 전송한다.</p> <p>⑤ 사용자는 알림창을 통해서 탈퇴 승인 여부를 확인할 수 있다.</p>
전제 조건	사용자가 로그인 한 상태로, 사용자 클라이언트에 JWT 토큰이 있어야 한다.
사후 조건	DB에서 해당 사용자 계정 정보와 사용자가 참여한 프로젝트 관계와 즐겨 찾기 정보를 모두 삭제한다.
가정	사용자는 이미 계정을 갖고 있어야 하며 로그인이 된 상태여야 한다. JWT 토큰에 있는 사용자 정보가 DB에 등록되어 있다.

[표 20] 사용례 (4) 즐겨 찾기 추가

이름	즐거 찾기 추가
행위자(Actor)	사용자
설명	사용자는 등록된 프로젝트 중 자신이 관심 있는 프로젝트를 즐겨 찾기 목록에 추가할 수 있다.
절차	① 사용자가 즐겨 찾기 추가 버튼을 누른다. ② 해당 프로젝트 식별 아이디와 사용자의 JWT 토큰이 HTTP POST 요청으로 서버에 전송된다. ③ 서버는 JWT 토큰을 복호화 한다. ④-A. JWT 토큰 복호화가 정상적으로 이루어지면, 서버는 토큰에 해당하는 사용자 계정 식별 아이디와 해당 프로젝트 식별 아이디를 즐겨 찾기 레코드에 추가한다. 그리고 해당 프로젝트의 즐겨 찾기 횟수를 DB에 업데이트 한다. 이후 즐겨 찾기 추가 메시지를 사용자에게 전송한다. ④-B. JWT 토큰의 복호화가 제대로 이루어지지 않으면, 즐겨 찾기 추가 실패 메시지를 전송한다. ⑤ 즐겨 찾기 추가 실패 시 실패 알림창이 뜨고, 성공한 경우 즐겨 찾기 버튼의 색이 바뀐다.
전제 조건	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다.
사후 조건	DB에 사용자가 프로젝트를 즐겨 찾기 등록한 상태를 추가한다.
가정	아직 해당 프로젝트가 사용자의 즐겨 찾기 목록에 추가되어 있지 않다.

[표 21] 사용례 (5) 즐겨 찾기 취소

이름	즐거 찾기 취소
행위자(Actor)	사용자
설명	사용자는 자신의 즐겨 찾기 목록에 있는 프로젝트를 즐겨 찾기 취소할 수 있다.
절차	① 사용자가 즐겨 찾기 취소 버튼을 누른다. ② 해당 프로젝트 식별 아이디와 사용자의 JWT 토큰이 HTTP POST 요청으로 서버에 전송된다. ③ 서버는 JWT 토큰을 복호화 한다. ④-A. JWT 토큰 복호화가 정상적으로 이루어지면, 서버는 토큰에 해당하는 사용자 계정 식별 아이디와 해당 프로젝트 식별 아이디를 즐겨 찾기 레코드에서 찾는다. 찾은 레코드를 삭제한다. 그리고 해당 프로젝트의 즐겨 찾기 횟수를 DB에 업데이트 한다. ④-B. JWT 토큰의 복호화가 제대로 이루어지지 않으면, 즐겨 찾기 취소 실패 메시지를 전송한다. ⑤ 즐겨 찾기 취소 실패 시 실패 알림창이 뜨고, 성공한 경우 즐겨 찾기 버튼의 색이 바뀐다.

이름	즐거 찾기 취소
전제 조건	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다.
사후 조건	DB에 사용자가 프로젝트를 즐겨 찾기 등록한 상태를 제거한다.
가정	해당 프로젝트가 사용자의 즐겨 찾기 목록에 추가되어 있어야 한다.

[표 22] 사용례 (6) 프로젝트 등록

이름	프로젝트 등록
행위자(Actor)	사용자
설명	사용자는 프로젝트 이름, 주제, 관련 기술 스택, 목표 팀원 수, 현재 팀원 수, 프로젝트 예상 시작 일정과 종료 일정을 필수적으로 입력해서 프로젝트를 등록할 수 있다. 프로젝트 관리자는 자동으로 요청한 사용자의 계정으로 연결된다.
절차	① 사용자는 프로젝트 등록 페이지에 들어온다. ② 사용자는 프로젝트 이름, 주제, 관련 기술 스택, 목표 팀원 수, 현재 팀원 수, 프로젝트 예상 시간 일정과 종료 일정을 입력하고, 프로젝트 생성 요청을 JWT 토큰과 함께 서버에 HTTP POST 요청으로 전송한다. ③ JWT 토큰을 복호화에 성공하면, 서버는 DB에 프로젝트 이름, 프로젝트 주제, 관련 기술 스택, 목표 팀원 수, 현재 팀원 수, 프로젝트 예상 시작 일정과 종료 일정을 입력 받은 값의 레코드를 생성한다. 이때, 관리자는 요청을 보낸 사용자이고, 마지막 수정 시간은 요청이 들어온 현재 시간, 그리고 즐겨 찾기 횟수는 0으로 초기화 한다. ④ 서버는 레코드 생성 후 성공 여부 메시지를 클라이언트로 보낸다.
전제 조건	프로젝트 이름, 주제, 관련 기술 스택, 목표 팀원 수, 현재 팀원 수, 프로젝트 예상 시작과 종료 일정은 필수적으로 입력되어야 할 사항이다.
사후 조건	DB에 새로운 프로젝트를 생성한다.
가정	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다.

[표 23] 사용례 (5) 프로젝트 수정

이름	프로젝트 수정
행위자(Actor)	프로젝트 담당자
설명	사용자는 자신이 생성한 프로젝트 내용을 수정할 수 있다.
절차	① 사용자는 자신의 관리 리스트에 있는 프로젝트 수정 페이지로 들어간다. ② 사용자는 자신이 수정하고 싶은 프로젝트 내용을 수정한다. 이때, 프로젝트 필수 정보를 빈 칸으로 수정할 수는 없다. ③ 사용자의 JWT 토큰과 수정할 프로젝트의 식별 정보와 수정된 후의 데이

	<p>터를 서버로 전송한다.</p> <p>④ 서버는 받은 JWT 토큰이 유효한지 검사하고 JWT 토큰의 사용자 계정과 해당 프로젝트의 담당자와 일치 여부를 확인한다.</p> <p>⑤-A. JWT 토큰과 프로젝트 담당자 계정 정보가 일치하지 않으면 요청을 거절한다. 서버는 거절 메시지를 프로젝트 담당자에게 보낸다.</p> <p>⑤-B. JWT 토큰과 프로젝트 담당자 계정 정보가 일치하면, 수정된 데이터로 프로젝트 레코드를 업데이트 한다. 서버는 성공 메시지를 프로젝트 담당자에게 보낸다.</p> <p>⑥ 프로젝트 수정 성공 실패 여부를 프로젝트 담당자에게 알림창으로 알린다.</p>
전제 조건	사용자의 JWT의 계정 정보가 프로젝트 담당자 계정과 일치해야 한다. 프로젝트 필수 정보를 빈 칸으로 바꿀 수 없다.
사후 조건	DB에 등록된 프로젝트의 정보가 수정된다.
가정	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다.

[표 24] 사용례 (7) 프로젝트 삭제

이름	프로젝트 삭제
행위자(Actor)	프로젝트 담당자
설명	사용자는 자신이 생성한 프로젝트를 삭제할 수 있다.
절차	<p>① 사용자는 자신의 관리 리스트에 있는 프로젝트 수정 페이지에서 삭제 요청을 보낸다.</p> <p>② 사용자의 JWT 토큰과 삭제할 프로젝트 식별 번호를 보낸다.</p> <p>③ 서버는 받은 JWT 토큰이 유효한지 검사하고 JWT 토큰의 사용자 계정과 해당 프로젝트의 담당자와 일치 여부를 확인한다.</p> <p>④-A. JWT 토큰과 프로젝트 담당자 계정 정보가 일치하지 않으면 요청을 거절한다. 서버는 거절 메시지를 프로젝트 담당자에게 보낸다.</p> <p>④-B. JWT 토큰과 프로젝트 담당자 계정 정보가 일치하면, 프로젝트 정보 레코드와 프로젝트에 연결된 즐겨 찾기, 참여 관계를 삭제한다. 서버는 성공 메시지를 프로젝트 담당자에게 보낸다.</p> <p>⑤ 프로젝트 삭제 성공 여부를 프로젝트 담당자에게 알림창으로 알린다.</p>
전제 조건	사용자의 JWT의 계정 정보가 프로젝트 담당자 계정과 일치해야 한다.
사후 조건	DB에 등록된 프로젝트의 정보가 삭제되고, 프로젝트와 연결된 즐겨 찾기, 참여 관계 DB가 삭제된다.

가정	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다.
----	-------------------------------

[표 25] 사용례 (8) 참여 관계 추가

이름	참여 관계 추가
행위자(Actor)	프로젝트 담당자
설명	사용자는 자신이 생성한 프로젝트의 참여자를 추가할 수 있다.
절차	<p>① 사용자는 자신의 관리 리스트에 있는 프로젝트에 참여자 관리 페이지로 이동한다.</p> <p>② 사용자의 JWT 토큰과 새로 추가할 계정의 아이디를 HTTP POST 요청으로 서버에 보낸다.</p> <p>③ 서버는 받은 JWT 토큰이 유효한지 검사하고 JWT 토큰의 사용자 계정과 해당 프로젝트의 담당자와 일치 여부를 확인한다.</p> <p>④-A. JWT 토큰과 프로젝트 담당자 계정 정보가 일치하지 않으면 요청을 거절한다. 6번 단계로 넘어간다.</p> <p>④-B. JWT 토큰과 프로젝트 담당자 계정 정보가 일치하면, 5번 단계로 넘어간다.</p> <p>⑤ 추가할 아이디가 이미 프로젝트에 참여하고 있는 지 확인한다. 만약 추가할 수 있다면, 참여 관계 레코드를 승인된 상태로 업데이트하고 프로젝트 현재 참여자 수를 업데이트 한다. 이미 참여하고 있다면, 요청을 거절한다.</p> <p>⑥ 서버는 요청에 대한 처리 결과를 프로젝트 담당자에게 전송한다.</p> <p>⑦ 프로젝트 담당자는 서버 처리 결과를 알림창으로 확인한다.</p>
전제 조건	사용자의 JWT의 계정 정보가 프로젝트 담당자 계정과 일치해야 한다. 프로젝트에 새로 참여하는 사람의 아이디가 필요하다.
사후 조건	DB에 참여 관계 레코드가 승인 상태로 업데이트되고, 프로젝트 참여자 인원이 업데이트 된다.
가정	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다. 새로 참여할 계정의 아이디가 프로젝트에 아직 참여하고 있지 않아야 한다.

[표 26] 사용례 (9) 참여 관계 취소

이름	참여 관계 취소
행위자(Actor)	프로젝트 담당자
설명	사용자는 자신이 생성한 프로젝트의 참여자를 제거할 수 있다.
절차	① 사용자는 자신의 관리 리스트에 있는 프로젝트에 참여자 관리 페이지로 이동한다.

	<p>② 사용자의 JWT 토큰과 새로 삭제할 계정의 아이디를 HTTP POST 요청으로 서버에 보낸다.</p> <p>③ 서버는 받은 JWT 토큰이 유효한지 검사하고 JWT 토큰의 사용자 계정과 해당 프로젝트의 담당자와 일치 여부를 확인한다.</p> <p>④-A. JWT 토큰과 프로젝트 담당자 계정 정보가 일치하지 않으면 요청을 거절한다. 6번 단계로 넘어간다.</p> <p>④-B. JWT 토큰과 프로젝트 담당자 계정 정보가 일치하면, 5번 단계로 넘어간다.</p> <p>⑤ 추가할 아이디가 프로젝트에 참여하고 있는 지 확인한다. 만약 참여하고 있다면, 참여 관계 레코드를 삭제하고 프로젝트 현재 참여자 수를 업데이트한다. 참여하고 있는 아이디가 아니면, 요청을 거절한다.</p> <p>⑥ 서버는 요청에 대한 거절 여부를 프로젝트 담당자에게 전송한다.</p> <p>⑦ 프로젝트 담당자는 서버 처리 결과를 알림창으로 확인한다.</p>
전제 조건	사용자의 JWT의 계정 정보가 프로젝트 담당자 계정과 일치해야 한다. 프로젝트에서 탈퇴할 사람의 아이디가 필요하다.
사후 조건	DB에 참여 관계 레코드가 삭제되고, 프로젝트 참여자 인원이 업데이트 된다.
가정	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다. 삭제할 아이디 계정이 프로젝트에 참여하고 있어야 한다.

[표 27] 사용례 (10) 프로젝트 참여 신청

이름	프로젝트 참여 신청
행위자(Actor)	사용자
설명	사용자는 등록된 프로젝트에 참여 신청 요청을 할 수 있다.
절차	<p>① 사용자는 등록된 프로젝트의 세부 정보 페이지에서 참여 신청을 요청할 수 있다.</p> <p>② 사용자의 JWT 토큰과 참여하고 싶은 프로젝트 식별 번호를 HTTP POST 요청으로 서버에 보낸다.</p> <p>③ 서버는 받은 JWT 토큰이 유효한지 검사한다.</p> <p>④-A. JWT 토큰이 유효하지 않으면 6번 단계로 넘어간다.</p> <p>④-B. JWT 토큰이 유효하면 5번 단계로 넘어간다.</p> <p>⑤ 이미 요청한 사용자가 해당 프로젝트에 참여하고 있으면 요청은 거절된다. 반면, 사용자가 아직 참여하지 않고 있다면 사용자가 참여 신청을 보냈음을 새로운 레코드에 생성한다.</p>

	⑥ 서버는 요청에 대한 처리 결과를 프로젝트 담당자에게 전송한다. ⑦ 프로젝트 담당자는 서버 처리 결과를 알림창으로 확인한다.
전제 조건	사용자의 JWT의 계정 정보가 유효해야 한다.
사후 조건	DB에 참여 관계 레코드가 추가되고, 프로젝트 참여자 인원이 업데이트 된다.
가정	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다. 요청을 받는 프로젝트가 등록된 상황이어야 한다.

[표 28] 사용례 (11) 프로젝트 참여 신청 취소

이름	프로젝트 참여 신청 취소
행위자(Actor)	사용자
설명	사용자는 등록된 프로젝트에 참여 신청을 취소할 수 있다.
절차	① 사용자는 등록된 프로젝트의 세부 정보 페이지에서 참여 신청을 취소할 수 있다. ② 사용자의 JWT 토큰과 참여 신청 취소하고 싶은 프로젝트 식별 번호를 HTTP POST 요청으로 서버에 보낸다. ③ 서버는 받은 JWT 토큰이 유효한지 검사한다. ④-A. JWT 토큰이 유효하지 않으면 6번 단계로 넘어간다. ④-B. JWT 토큰이 유효하면 5번 단계로 넘어간다. ⑤ 요청한 사용자가 해당 프로젝트에 참여하고 있지 않은 상황이라면 신청 레코드를 테이블에서 삭제한다. 반면, 사용자가 참여하고 있다면 사용자의 참여 신청을 삭제할 뿐만 아니라 프로젝트 정보도 업데이트 한다. ⑥ 서버는 요청에 대한 처리 결과를 프로젝트 담당자에게 전송한다. ⑦ 프로젝트 담당자는 서버 처리 결과를 알림창으로 확인한다.
전제 조건	사용자의 JWT의 계정 정보가 프로젝트 담당자 계정과 일치해야 한다. 프로젝트에 사용자가 신청을 보낸 상태이다.
사후 조건	DB에 참여 관계 레코드가 삭제되고, 프로젝트 참여자 인원이 업데이트 된다.
가정	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다. 신청 취소할 프로젝트가 등록된 상태이다.

[표 29] 사용례 (12) 프로젝트 검색

이름	프로젝트 검색
행위자(Actor)	사용자
설명	사용자는 현재 모집 진행 중인 프로젝트를 검색할 수 있다. 검색어가 주어질 경우 해당 검색어로 제목 검색이 가능하다.
절차	① 사용자는 검색창에 검색어를 입력한다. ② 사용자의 검색어는 서버에 HTTP GET 요청으로 전송된다. ③ 서버는 검색어가 프로젝트 이름에 포함된 프로젝트를 DB에서 찾는다. ④ 해당되는 프로젝트의 리스트를 사용자 브라우저 UI에 맞게 보여준다. 해당되는 프로젝트가 없을 경우 해당되는 프로젝트가 없다는 페이지를 보여준다.
전제 조건	검색어가 비어 있다면, 전체 프로젝트를 수정 일자가 최신순으로 나열된다. 검색어가 주어질 경우, 해당 검색어가 프로젝트 이름에 포함된 프로젝트만 찾아 준다.
사후 조건	검색어에 따라 사용자가 보는 프로젝트 목록을 업데이트한다.
가정	처음 프로젝트 검색창에 들어오면 검색창은 비어있다.

[표 30] 사용례 (13) 프로젝트 랜덤 검색

이름	프로젝트 랜덤 검색
행위자(Actor)	사용자
설명	사용자는 현재 모집 진행 중인 프로젝트 중 무작위로 뽑기 기능을 사용할 수 있다.
절차	① 사용자는 프로젝트 무작위 뽑기를 기능을 사용한다. ② 사용자의 요청은 HTTP GET 요청으로 서버에 전송된다. ③ 서버는 현재 등록된 프로젝트 중 DB에서 무작위로 선별한 프로젝트 리스트를 사용자에게 보내준다. ④ 해당되는 프로젝트의 리스트를 사용자 브라우저 UI에 맞게 보여준다. 만약 랜덤 뽑기를 할 수 있을 만큼 프로젝트가 있지 않다면 랜덤 뽑기에 실패했다는 메시지 창을 보여준다.
전제 조건	등록된 프로젝트가 하나 이상 있어야 된다.
사후 조건	무작위로 뽑은 프로젝트 리스트를 보여준다.
가정	프로젝트가 하나 이상 등록된 상태이다.

[표 31] 사용례 (14) 회원 검색

이름	회원 검색
행위자(Actor)	사용자
설명	사용자는 다른 사용자를 검색할 수 있다. 검색을 위해서는 검색할 항목과 검색어를 입력해야 한다. 검색어가 없다면 모든 사용자가 제공된다.
절차	① 사용자는 검색창에 검색 항목과 검색어를 입력한다. ② 사용자의 검색은 서버에 HTTP GET 요청으로 전송된다. ③ 서버는 요청된 DB 항목 필드에서 검색어가 포함된 사용자를 찾는다. ④ 서버는 해당되는 사용자 리스트를 사용자에게 전송한다. ⑤ 사용자의 UI에 따라 사용자 리스트를 보여준다.
전제 조건	검색 항목과 검색어를 설정해 주어야 한다.
사후 조건	검색어에 따라 사용자가 보는 사용자 목록을 업데이트 한다.
가정	검색어가 주어지지 않은 경우 전체 사용자 목록이 제공된다.

[표 32] 사용례 (15) 마이 페이지 수정

이름	마이 페이지 수정
행위자(Actor)	사용자
설명	사용자는 자신의 세부 정보를 수정할 수 있다.
절차	① 사용자는 자신의 마이 페이지에 있는 수정 페이지로 들어간다. ② 사용자는 자신이 수정하고 싶은 마이 페이지 내용을 수정한다. 이때, 사용자 필수 정보를 빈 칸으로 수정할 수는 없다. ③ 사용자의 JWT 토큰과 수정된 후의 데이터를 서버로 전송한다. ④ 서버는 받은 JWT 토큰이 유효한지 검사한다. ⑤-A. JWT 토큰이 유효하지 않으면 요청을 거절한다. 서버는 거절 메시지를 프로젝트 담당자에게 보낸다. ⑤-B. JWT 토큰이 유효하면 수정된 데이터로 사용자 정보를 업데이트 한다. 서버는 성공 메시지를 프로젝트 담당자에게 보낸다. ⑥ 마이 페이지 정보 수정 성공 실패 여부를 프로젝트 담당자에게 알림창으로 알린다.
전제 조건	사용자 JWT 토큰이 유효해야 한다. 사용자 세부 정보 중 필수 정보를 빈 칸으로 바꿀 수 없다.

사후 조건	DB에 등록된 사용자의 정보가 수정된다.
가정	사용자가 로그인한 상태로 JWT 토큰이 있어야 한다. 아이디와 이메일은 변경할 수가 없다. 사용자는 오로지 자신 계정의 마이 페이지 수정만 가능하다.

[표 33] 사용례 (16) 회원 세부 정보 조회

이름	회원 세부 정보 조회
행위자(Actor)	사용자
설명	사용자는 다른 사용자의 세부 정보를 조회할 수 있다.
절차	<p>① 사용자는 원하는 회원의 세부 정보 페이지 조회 요청을 서버에게 보낼 수 있다.</p> <p>② 사용자는 조회하고 싶은 회원의 사용자 식별 번호를 HTTP GET 요청을 통해 서버에 전송한다.</p> <p>③ 서버는 받은 요청의 사용자 식별 번호를 받아 해당 사용자를 DB에서 찾는다.</p> <p>④-A. 조회하려는 사용자가 DB에 없다면 서버는 요청을 거절한다.</p> <p>④-B. 조회하려는 사용자가 DB에 있다면 사용자 세부 정보를 사용자에게 전송한다.</p> <p>⑤ 사용자는 요청이 거절되면 거절되었다는 응답을 받고, 요청이 처리되면 조회하려는 사용자의 세부 정보 페이지로 이동한다.</p>
전제 조건	조회하려는 사용자의 사용자 식별 번호가 필요하다.
사후 조건	조회하려는 사용자의 세부 정보 페이지로 이동한다.
가정	조회하려는 사용자가 사용자 DB에 등록되어 있어야 한다.

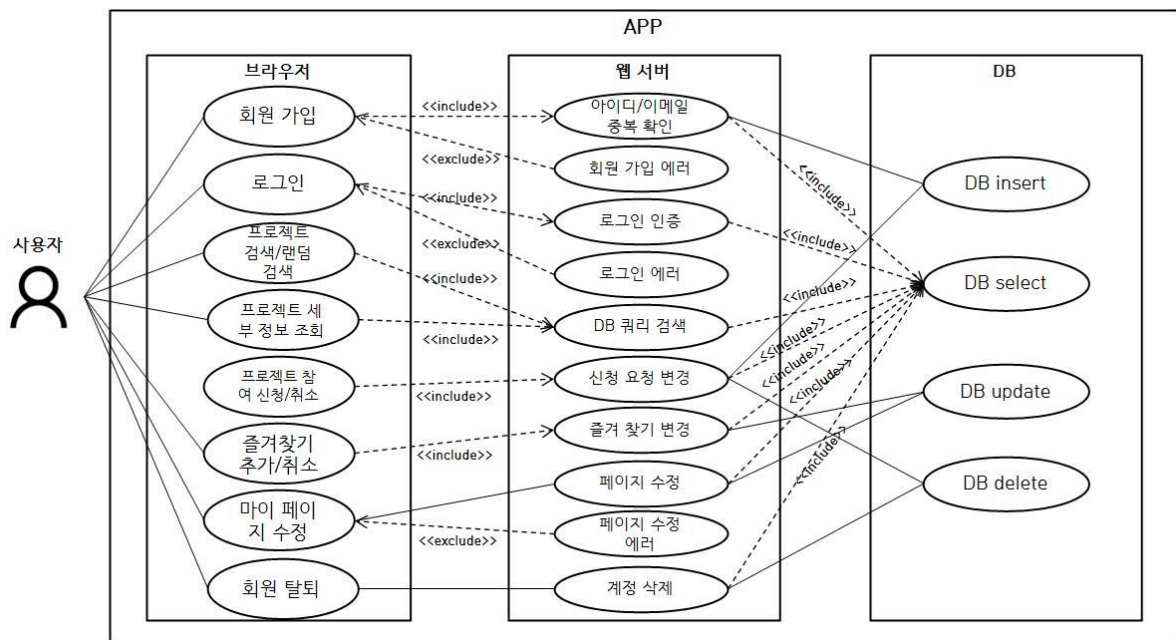
[표 34] 사용례 (17) 프로젝트 세부 정보 조회

이름	프로젝트 세부 정보 조회
행위자(Actor)	사용자
설명	사용자는 프로젝트 세부 정보를 조회할 수 있다.
절차	<p>① 사용자는 원하는 프로젝트 세부 정보 페이지 조회 요청을 서버에게 보낼 수 있다.</p> <p>② 사용자는 조회하고 싶은 프로젝트의 식별 번호를 HTTP GET 요청을 통해 서버에 전송한다.</p> <p>③ 서버는 받은 요청의 프로젝트 식별 번호를 받아 해당 프로젝트를 DB에서 찾는다.</p>

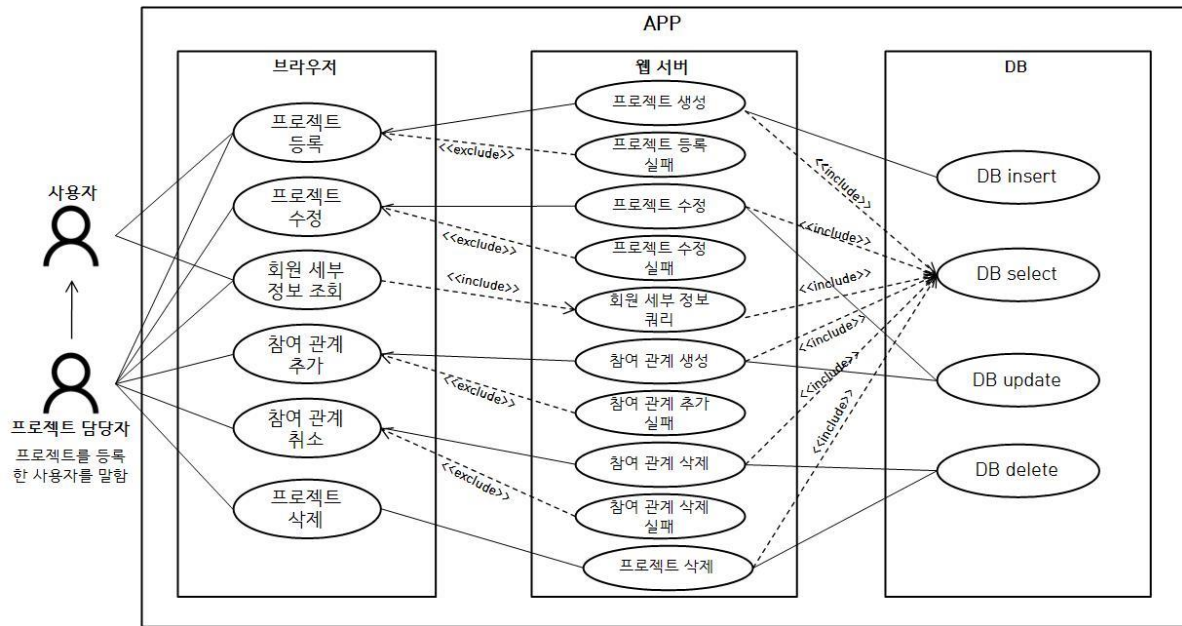
	<p>④-A. 조회하려는 프로젝트가 DB에 없다면 서버는 요청을 거절한다.</p> <p>④-B. 조회하려는 프로젝트가 DB에 있다면 사용자 세부 정보를 사용자에게 전송한다.</p> <p>⑤ 사용자는 요청이 거절되면 거절되었다는 응답을 받고, 요청이 처리되면 조회하려는 프로젝트 세부 정보 페이지로 이동한다.</p>
전제 조건	조회하려는 프로젝트의 프로젝트 식별 번호가 필요하다.
사후 조건	조회하려는 프로젝트 세부 정보 페이지로 이동한다.
가정	조회하려는 프로젝트가 프로젝트 DB에 등록되어 있어야 한다.

3.2.2. Use Case Diagram

[그림 1313] 프로젝트 탐색 사용자 스토리에 따른 Use Case Diagram



[그림 1414] 프로젝트 담당자 스토리



3.2.3. Data Dictionary

[표 35] 데이터베이스 'Project' Table

Field	Key	Constraint	Description
projectId	Primary Key	Unique, Not NULL	integer, 프로젝트 식별 번호
name		Not Null	varchar, 프로젝트 이름
topic		Not Null	varchar, 프로젝트 주제
leader	Foreign Key	Not NULL	integer, 프로젝트 담당자, Users 테이블의 userId 외래키
contact		Not NULL	varchar, 연락처 링크
stacks		Not NULL	varchar, #로 구분한 공백 문자가 없는 문자열, 기술 스택
required		Not NULL	integer, 총 모집 인원 수
current		Not NULL	integer, 현재 참여 인원 수, 초기값 0
stars		Not NULL	integer, 즐겨 찾기 수, 초기값 0
startTime		Not NULL	date, 프로젝트 시작 예정 시기
endTime		Not NULL	date, 프로젝트 종료 예정 시기
lastEdit		Not NULL	date, 프로젝트 마지막 수정 날짜, 초기값 생성한 시점의 날짜
image			varchar, 프로젝트 소개 이미지 파일 이름

* 새로운 사용자가 등록될 때 본 테이블에 값을 추가해야 한다.

[표 36] 데이터베이스 'User' Table

Field	Key	Constraint	Description
userId	Primary Key	Not Null	integer, 사용자 식별 번호
id		Unique, Not NULL	varchar, 사용자 계정 아이디
password		Not NULL	varchar, 사용자의 아이디와 비밀번호를 이어 붙인 문자열의 해시값
name		Not NULL	varchar, 이름(활동명)
major		Not NULL	varchar, 전공
email		Not NULL	varchar, 이메일
link			varchar, 자기 소개를 위한 외부 페이지 링크 주소
stacks			varchar, #로 구분한 공백 문자가 없는 문자열, 기술 스택
interest			varchar, #로 구분한 공백 문자가 없는 문자열, 흥미 분야
message			varchar, 공백 문자 포함 가능, 간단한 자기 소개
image			varchar, 프로필 이미지 파일 이름

* 시스템 관리자가 수업을 추가할 때 본 테이블에 값을 추가해야 한다.

[표 37] 데이터베이스 'Star' Table

Field	Key	Constraint	Description
id	Primary Key	Not NULL, Unique	integer, auto increment로 생성
user	Foreign Key	Not Null	integer, Users Table의 userId 외래키
project	Foreign Key	Not Null	integer, Projects Table의 projectId 외래키

* 임의의 사용자가 임의의 수업을 완료했을 때 본 테이블에 값이 추가된다.

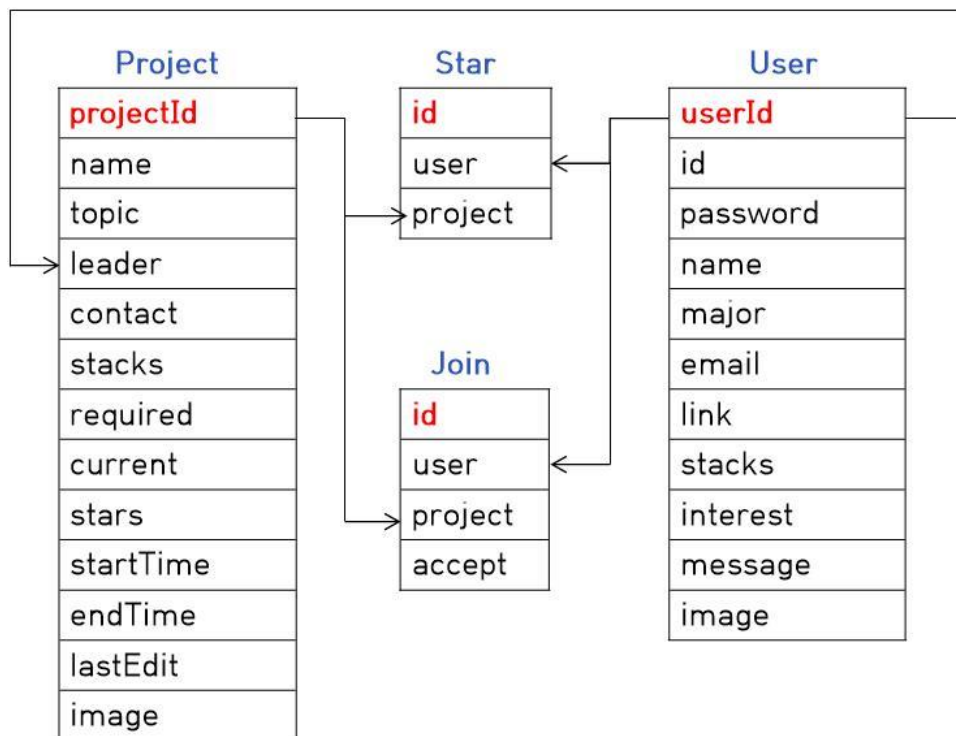
[표 38] 데이터베이스 'Join' Table

Field	Key	Constraint	Description
id	Primary Key	Not NULL, Unique	integer, auto increment로 생성
user	Foreign Key	Not Null	integer, Users Table의 userId 외래키
project	Foreign Key	Not Null	integer, Projects Table의 projectId 외래키

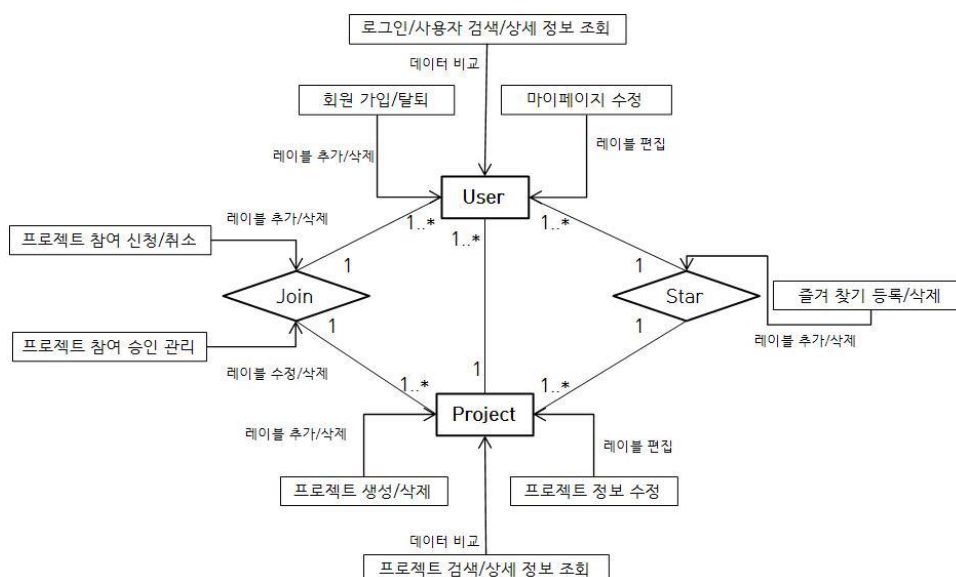
accept		Not NULL	boolean, 초기값은 false, 프로젝트 관리자의 참여 신청 승인 여부
--------	--	----------	--

3.2.4. Data Flow Diagram

[그림 1515] MySQL 데이터베이스 구조



[그림 1616] 데이터베이스 상호작용



3.3. Performance Requirements

다음 항목들은 본 서비스가 제공해야 할 성능의 요구조건을 나타낸다.

3.3.1. Static numerical requirement

- ✓ 본 시스템은 하드웨어 요구조건을 만족하는 한, 브라우저 및 응용 프로그램에서 연결 종료라고 판단하는 TTL 이전에 정보를 전달해야 한다.

3.3.2. Dynamic numerical requirement

- ✓ 본 시스템은 최대 100명의 동시 사용자를 가정하고 있으며, 그 이상의 인원 에 대해서는 최적의 품질을 제공하기 어렵다.
- ✓ 페이지 이동 등 데이터베이스에서 자료를 찾지 않고, 데이터베이스를 수정 하지 않는 동작은 5초 안에 수행되어야 한다.
- ✓ 데이터베이스에서 자료를 탐색하는 과정을 동반하는 각 기능은 10초 이내에 수행되어야 한다.

3.4. Logical Database Requirements

본 시스템은 MySQL 데이터베이스를 사용한다. 서버와의 연동을 위해 NodeJS 의 MySQL 플러그인을 사용한다. 사용자 정보, 프로젝트 정보를 데이터베이스에 저장한다. 사용자가 추가될 때마다 사용자 테이블에 새로운 레이블이 생성될 것이며, 프로젝트를 등록할 때마다 프로젝트 테이블에 새로운 레이블이 생성된다.

즐거 찾기 관계 테이블에서는 임의의 사용자가 임의의 프로젝트를 즐겨 찾기 할 때마다 그 순서쌍을 레이블의 형태로 저장하며, 추후 이 레이블을 참고하여 사용자가 즐겨찾기 한 프로젝트 정보를 가져올 수 있다. 이는 또한 프로젝트 참여 관계에도 적용된다.

이를 위해 MySQL은 JOIN 외의 동시다발적인 쿼리에 대한 동작을 수행할 수 있어야한다. MySQL 데이터베이스는 본 시스템을 위해 필요한 모든 기능을 큰 성

능 부하 없이 제공한다.

3.5. Design Constraints

본 시스템은 Apache2 라이선스를 따른다. NodeJS 기반의 웹 서비스를 제공한다. 서버는 사용자에게 교육 자료를 제공한다. 사용자가 입력한 코드를 서버가 수행하고 결과를 반환할 수 있다. 본 시스템은 NodeJS, Docker, Judge0, SQLite를 기반 설계되었다.

3.6. Standards compliance

시스템 내의 모든 JavaScript는 ECMAScript 표준을 따른다. NodeJS 구현 및 사용자에게 전달되는 JavaScript 양 측에 해당되는 내용이다. CSS는 W3C의 표준을 따른다. 사용자에게 전달되는 Html의 경우 NodeJS의 템플릿 엔진 Express를 활용한다. Express 표준과 HTML5 표준이 상충할 경우 Chrome 브라우저에서 해석하는 것을 기준으로 한다. 동일한 경우 HTML5의 내용을 따른다.

서비스 가동 중 발생하는 모든 네트워크 통신은 TCP, HTTP 프로토콜 표준을 따르며 각 항목은 RFC 793, RFC 2616에 명시된 바를 따른다.

3.7. Software System Characteristics

소프트웨어 시스템 특성은 비기능적 요구 조건에 기반한다. 소프트웨어 시스템 특성 부분에서는 본 시스템의 비기능적 요구 조건을 각각 제품 요구 조건(Product Requirements), 조직 요구 조건(Organizational Requirements), 외부 요구 조건(External Requirements), 세 부분으로 나누어 설명한다.

3.7.1. Product Requirements

제품 요구 조건 항목은 본 시스템의 동작 중(runtime) 시스템에 요구되는 요구 조건을 말한다. 본 시스템은 동작 중 다음과 같은 요구 조건을 충족해야 한다.

3.7.1.1. Usability Requirements

본 시스템은 성균관대학교 학생들을 대상으로 사용성이 좋은 시스템을 제공

해야 한다. 대상 사용자에게 접근성이 좋은 브라우저 기반의 웹 어플리케이션으로 시스템이 제공된다. 따라서, 사용자가 본 시스템을 사용할 때는 웹 브라우저를 통해 해당 주소로 접근하는 것 이외에는 추가적인 설치 과정 및 사용 환경 조성 등의 요구가 없어야 한다. 뿐만 아니라 사용자가 등록, 검색, 수정 등의 기능을 사용 시, 일정한 형태의 값을 넣어야 함을 알려주기 위해서 알림 창을 통해서 입력 값 제한을 전달할 수 있어야 한다. 사용자의 요청 사항이 시스템 내부 문제로 실패할 경우, 실패했다는 메시지를 사용자에게 알림 창을 통해 알려 사용자가 요청 실패 이후 서버에 재 요청을 보낼 수 있도록 유도해야 한다.

3.7.1.2. Performance Requirements

본 시스템은 성균관대학교 학생들을 대상으로 원활한 환경의 웹 어플리케이션 플랫폼으로 기능해야 한다.

① 사용자 계정 생성은 최대 2초 지연 시간을 넘기지 않고 로그인 페이지로 이동할 수 있어야 한다.

② 로그인 처리는 최대 2초 지연 시간을 넘기지 않고 메인 페이지로 이동할 수 있어야 한다.

③ 프로젝트 및 사용자 검색 기능은 최대 3초 지연 시간을 넘기지 않고 첫 페이지 결과가 나올 수 있어야 한다.

④ 검색 결과 리스트에서 다음 리스트로 이동하는 지연 시간이 최대 3초를 넘기지 않아야 한다.

⑤ 프로젝트 랜덤 뽑기 기능은 최대 3초 지연 시간을 넘기지 않고 랜덤 뽑기 결과를 사용자에게 보여줄 수 있어야 한다.

⑥ 프로젝트 생성 및 수정 기능은 최대 3초 지연 시간을 넘기지 않고 이루어져야 한다.

⑦ 즐겨 찾기 추가 및 삭제는 최대 2초 지연 시간을 넘기지 않고 이루어져야 한다.

⑧ 프로젝트 참여 관계 추가 및 삭제는 최대 2초 지연 시간을 넘기지 않고 이루어져야 한다.

⑨ 프로젝트 및 사용자 세부 정보 조회는 최대 3초 지연을 넘기지 않고 세부 정보 페이지로 이동할 수 있어야 한다.

⑩ 마이 페이지 수정은 최대 3초 지연을 넘기지 않고 수정 후 마이 페이지로 이동할 수 있어야 한다.

3.7.1.3. Security Requirements

본 시스템은 사용자의 프로젝트에 대한 권한 확인과 사용자 개인 정보 이용과 관련된 기능을 제공한다. 따라서 사용자의 개인정보와 권한을 적극적으로 보호할 의무가 있고 아래와 같은 요구 조건을 만족해야 한다.

① 사용자의 비밀번호는 암호화된 상태로 제공되어야 한다. 이를 위해 TCP 기반의 HTTP 프로토콜을 사용하고, 아이디와 비밀번호를 이어 붙인 형태를 암호화하여 비밀번호를 서버에 보내고 DB에 저장한다. 절대로 평문 형태의 비밀번호를 저장하지 않는다.

② 개인 사용자의 세부 정보 수정과 프로젝트 즐겨 찾기 기능 그리고 프로젝트 담당자 권한이 있는 사용자의 프로젝트 관리 권한을 보호해야 한다. 따라서 이러한 기능은 요청 시 서버가 로그인 시 발급한 토큰을 요청 시 보내는 방식으로 개인의 권한을 확인해야 한다. 특히, 권한이 중간에 수정이 된 경우 등 적절한 토큰과 함께 요청되지 않았을 경우 사용자 요청을 거절해야 한다.

3.7.2. Organizational Requirements

조직 요구 조건은 서비스 사용자와 개발자 집단의 규제 및 정책 상의 이유로 요구되는 조건을 말한다. 아래 항목은 환경 요구 조건(Environmental Requirements)와 운영 요구 조건(Operational Requirements), 두 부분으로 나누어 설명한다.

3.7.2.1. Environmental Requirements

- ✓ 본 시스템 관련 라이선스는 다음과 같다. DB인 MySQL이 GPL v2를 따른다. Node.js, Next.js, Express, Sequelize는 MIT 라이선스를 따른다. 따

라서, 두 라이선스에 위배되거나 충돌하지 않는 범위에서 개발자는 오픈 소스 API를 사용할 수 있다. 단, 각 API에 대해서는 요구 사항 명세서에 설명해야 한다.

- ✓ 프론트엔드 어플리케이션 개발자는 Next.js 기반의 개발을 할 수 있어야 한다. 백엔드 어플리케이션 개발자는 Node.js의 Express 프레임워크와 Sequelize 라이브러리로 개발할 수 있어야 한다. 개발자들은 관계형 데이터베이스 사용법을 알고 있어야 한다.
- ✓ 본 시스템의 운영자는 MySQL 데이터베이스 서버 접속을 통한 DB 수정이 가능하다. 그렇기 위해서 시스템 운영자는 관계형 데이터베이스 사용 방법을 알고 있어야 한다.

3.7.2.2. Operational Requirement

- ✓ 성균관대학교 학생들이 프로젝트 서비스를 지속적으로 이용하기 위해서 오전 9시에서 오후 5시 사이 어플리케이션 업데이트 및 서버 점검으로 인한 서버 복구 시간은 최초 서버 다운 시점 이후 30분에서 1시간 안에 가능하도록 한다.
- ✓ 성균관대학교 학생들을 대상으로 한 서비스이므로 계정 생성 시 이메일은 성균관대학교 이메일을 통해서만 사용 가능하다.

3.7.3. External Requirements

외부 요구 조건은 시스템의 운영과 개발 등에 영향을 주는 외부적 요인으로 파생된 요구 조건을 말한다.

3.7.3.1. Regulatory Requirement

- ✓ 본 시스템에서 사용하는 MySQL 데이터베이스는 GPL v2 라이선스이다. 따라서 GPL v2 소프트웨어를 이용하면서 지켜야 하는 규정에 따라 이용해야 한다. 이후 시스템 운영 중 데이터베이스를 변경할 때, 데이터베이스 사용에 맞는 규제를 따라야 할 수 있다.
- ✓ 본 시스템 구현을 위해 사용하는 Node.js의 Next.js와 Express 프레임

워크와 Sequelize 라이브러리는 MIT 라이선스를 따른다. MIT 라이선스의 경우 수정과 배포가 자유롭고 실무와 상업적 이용이 가능하다. 그렇지만 소스 사용으로 인한 피해는 전적으로 개발자가 감당해야 한다.

3.7.3.2. Legislative Requirement

- ✓ 본 시스템은 사용자의 이름, 이메일, 전공, 관심 분야, 소개 메시지 등 여러 개인정보를 사용한다. 따라서, 대한민국의 개인정보보호법에서 명시한 법령에 따라서 제공받은 개인 정보를 사용해야 한다.
- ✓ 본 시스템은 제3자에게 개인정보를 지급한다는 사용자의 허가를 받지 않기 때문에 제3자에게 사용자의 개인정보를 양도하면 안 된다.
- ✓ 본 시스템은 사용자에게 직접 제공받은 정보 만을 사용하기 때문에 추가적인 가명처리나 익명처리 없이 개인정보 사용이 가능하다.
- ✓ 본 시스템은 서비스를 운영하기 위해 필요한 최소한의 범위 내에서만 사용자의 개인정보를 요구하고 수집하고 사용해야 한다.
- ✓ 본 시스템은 사용자의 요구에 따라 개인정보를 삭제, 파기, 정정할 수 있어야 한다.
- ✓ 위 명시한 내용 외에도 본 시스템은 대한민국의 개인정보 보호법 법령을 준수해야 한다. 특히, 이후 개인정보보호법 개정에 맞도록 개인 정보를 이용해야 한다.

3.8. Organizing the Specific Requirements

본 항목에서, 우리는 시스템 모델을 Unified Modeling Language(UML)에 기반한 가시적인 형태와 표를 기반으로 나타내고자 한다. 이러한 모델은 시스템, 서브 시스템, 시스템 내의 각 요소와 운영 환경 등 세부적인 요구사항의 형태를 명확하게 나타낼 수 있다.

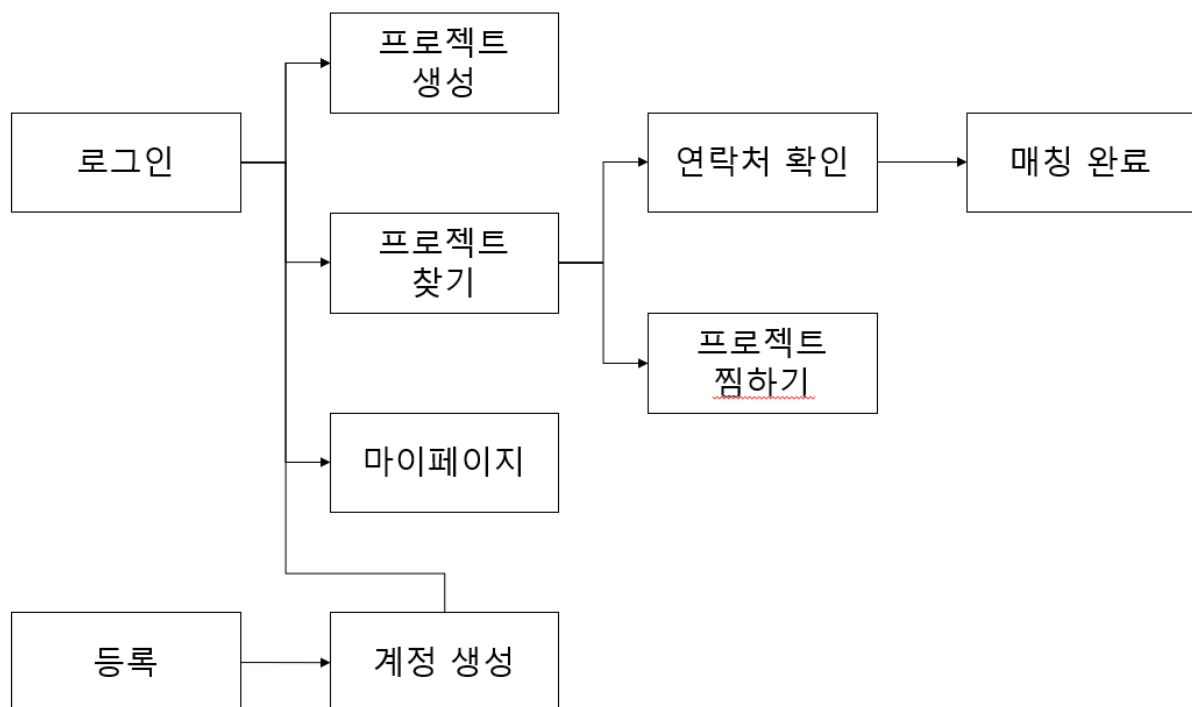
3.8.1. Context Model

[그림 1717] 맥락 모델



3.8.2. Process Model

[그림 1818] 프로세스 모델



3.8.3. Interaction Model

3.2.2 항목의 User Case Diagram 형태로 표현되었다.

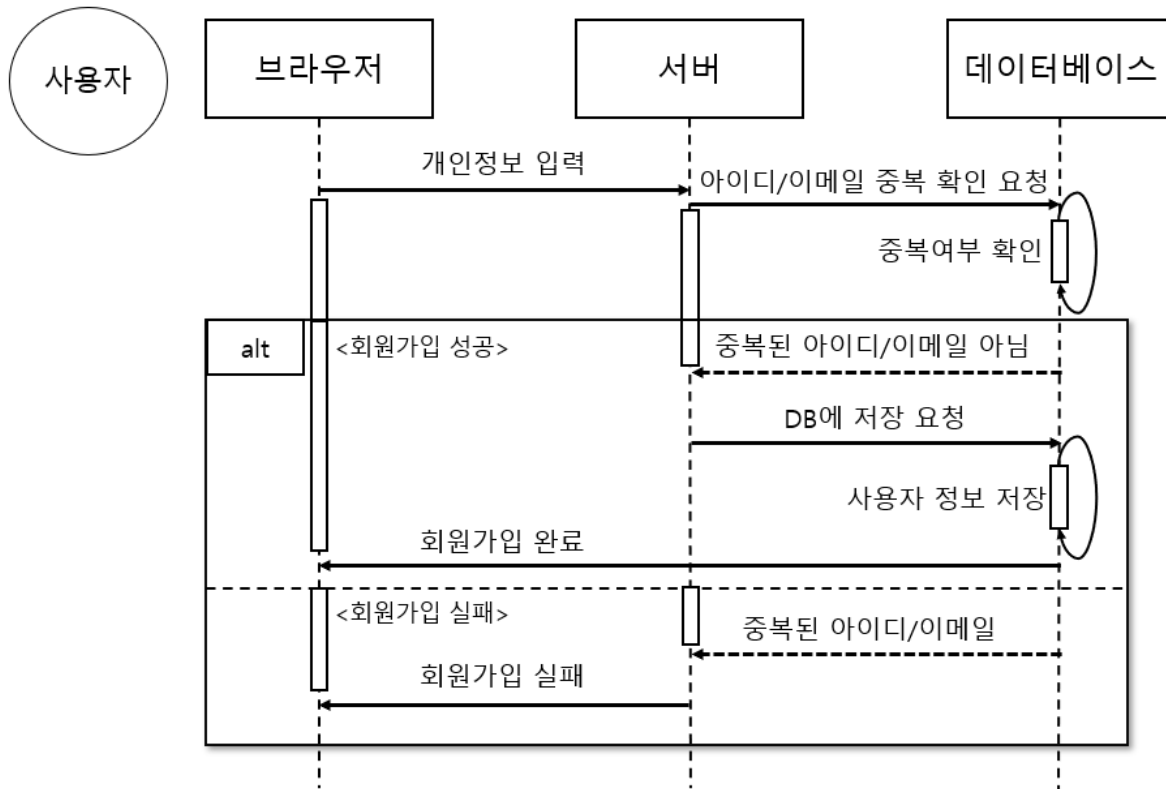
3.8.4. Behavior Model

3.8.4.1. Data Flow Diagram

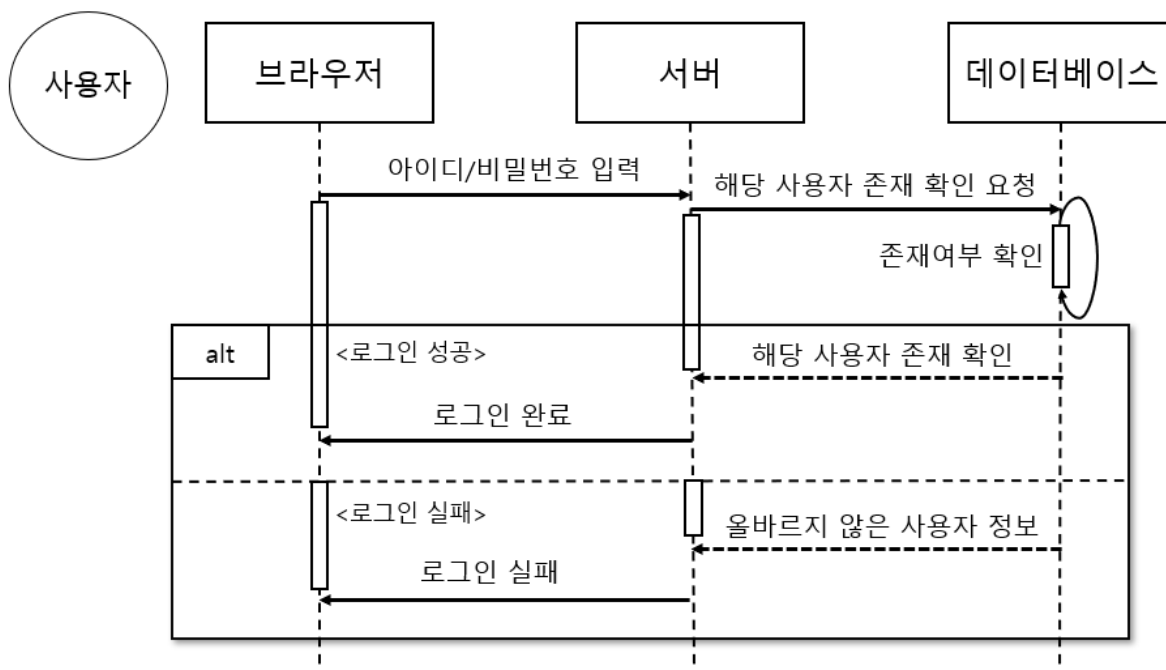
3.2.4 항목에 Data Flow Diagram 형태로 표현되었다.

3.8.4.2. Sequence Diagram

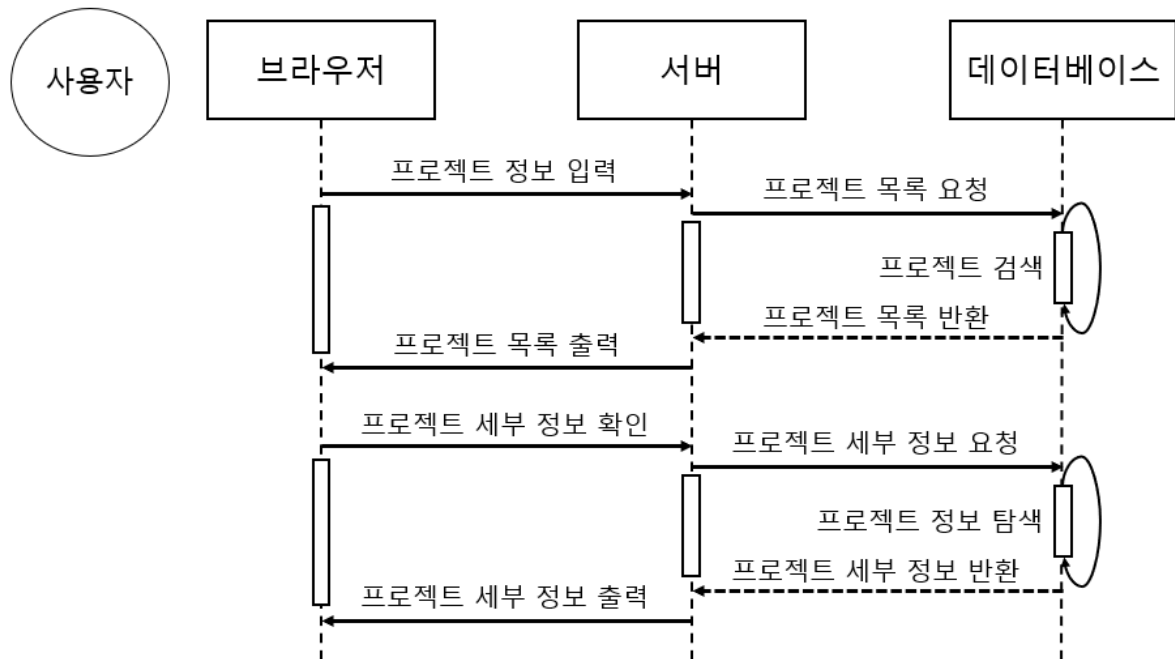
[그림 1919] 회원가입



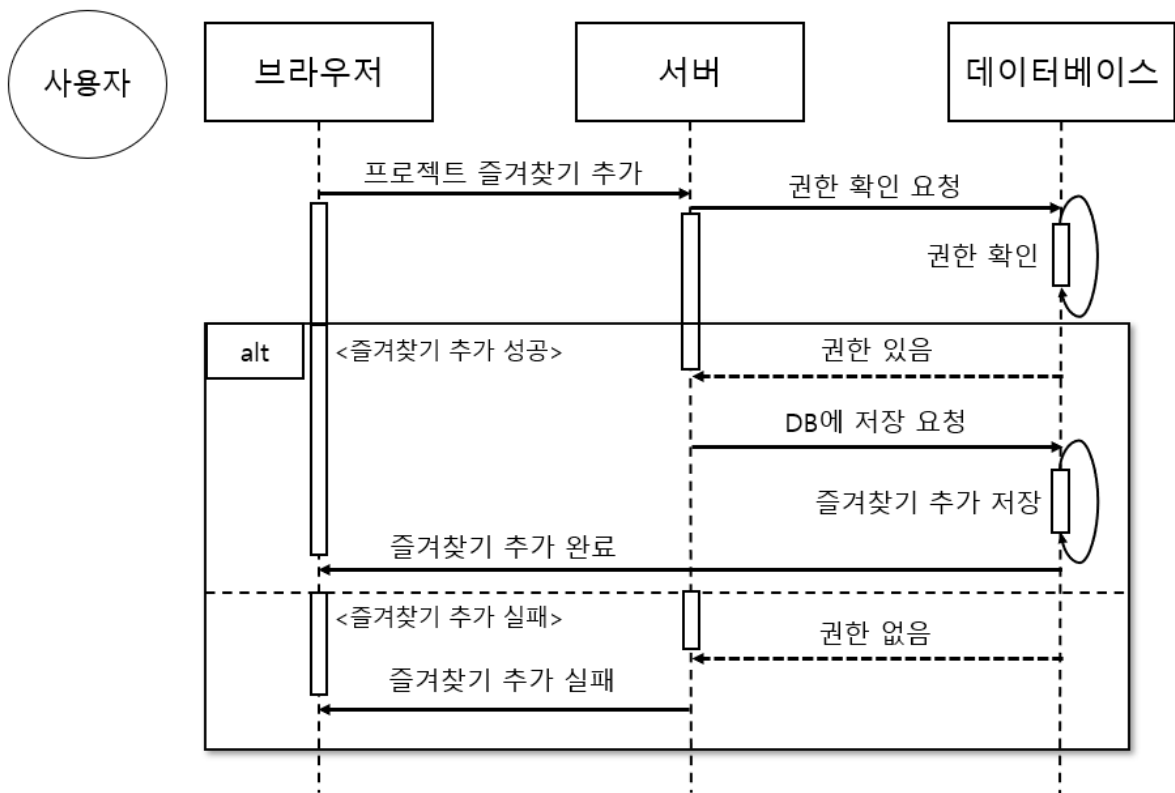
[그림 2020] 로그인



[그림 2121] 프로젝트 검색 및 세부정보 조회



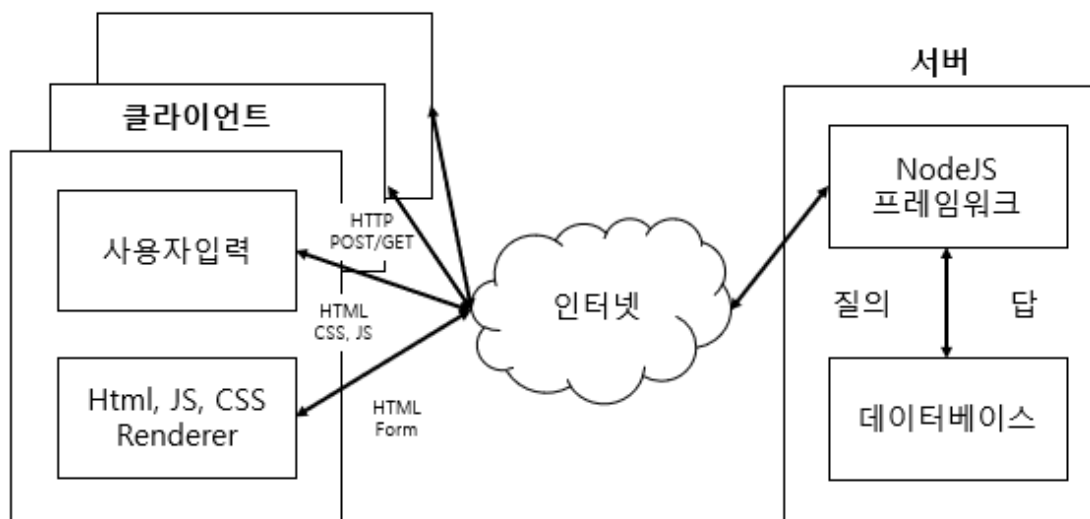
[그림 2222] 프로젝트 즐겨찾기 추가



3.9. System Architecture

본 항목은 시스템 구조를 고수준의 추상화된 도식 형태로 나타낸다. 시스템을 구성하는 각 요소 (사용자 브라우저, 서버)간 존재하는 기능과 통신에 대해 가장 근본적인 형태가 나타난다.

[그림 2323] 시스템 구조



3.10. System Evolution

본 항목에서 우리는 시스템이 추후 하드웨어 개선 및 소프트웨어 개선, 혹은 새로운 요구사항이 발생할 때 대처하기 위한 기본적인 입장을 서술한다. 이 항목은 추후 본 소프트웨어의 방향성을 서술하는 것이기 때문에 시스템 디자이너는 이를 고려하여 시스템을 설계하여야 한다.

3.10.1. Limitation and Assumption

본 서비스는 브라우저 기반으로 사용자가 프로젝트 팀빌딩을 할 수 있는 환경을 제공한다. 본 제품을 출시하는 시점에는 ‘프로젝트 등록’ 및 ‘팀원 등록’ 두 가지의 대분류 밖에 없으나, 추후 시스템 관리자와의 협의를 통해 다양한 대분류가 추가될 수 있다.

추가적으로 현재의 경우 관리자가 DB 수정을 원할 경우 직접적으로 DB 접근을

해야 한다. 이에 대해서 차후 관리자 페이지를 두어 DB 수정 작업을 원활하게 할 수 있도록 도울 수 있을 것이다.

3.10.2. Evolutions of Hardware and Change of User Requirements

사용자 니즈의 변화에 따라 현재 설계된 구동 플랫폼 이외에 모바일 브라우저에서도 시스템이 확장될 수 있다. 따라서 모바일 플랫폼에 적합한 기능과 인터페이스를 고려하여야 한다

사용자 요청 사항이 발생할 경우, 만일 새로운 기능을 개발하는 것이 아니라 단순히 내용물의 추가라면 시스템 관리자 선에서 해당 내용을 추가할 수 있다. 새로운 기능을 개발해야 하는 경우, 본 요구사항 명세서에 있는 내용들을 기반으로 새로운 API를 추가하거나 새로운 웹 서비스 페이지를 설계할 수 있다. 단 각 설계 단계에 있어 기존 요구사항 명세와 통일된 양식을 가져야 하며, 작성 후 본 문서 또한 수정되어야 한다.

4. Supporting Information

4.1. Software Requirement Specification

본 소프트웨어 요구 명세서는 IEEE 권장사항에 맞추어 작성되었다. 본 서비스에 적합한 요구 조건을 작성하기 위해 본래의 양식에서 일부 추가되거나 제외된 바 있다. (IEEE Recommended Practice for Software Requirements Specifications, IEEE-Std-830)

4.2. Document History

[표 39] 문서 이력

Date	Version	Description	Writer
2022.11.2.	1.0	팀원 회의 후 최초 버전 발행	Ilwon Jung, et al.