
오픈소스 소프트웨어 프로젝트 제안서

2023.04.12.

동국대학교  3팀 MAC
dongguk university

2017112288 민한결

2019112567 안상연

2019112453 최필환

서론

1. 개요

본 보고서는 오픈소스 소프트웨어 프로젝트 수업의 팀 프로젝트 제안서이다. 팀원 토의 결과 선정한 프로젝트인 ‘동국대 LMS 협업 파일시스템 구축’에 대한 요구사항 분석 및 프로젝트 설계, 수행계획 수립을 진행하였다. 요구사항 분석에서는 현행 시스템 분석, 사례 조사를 수행하여 요구사항 명세서를 작성하였다. 프로젝트 설계에서는 구현에 필요한 오픈소스 선정, 사용자 인터페이스 설계, DB 설계를 진행하였다. 수행계획 수립에서는 역할, 개발 환경, 협업 규칙을 선정하고 개발 일정을 수립하였다.

2. 참여자 명단

분 야	학 번	전공	학 년	이 름	역할
백엔드	2017112288	건설환경공학	4	민한결	팀장
	2019112453	산업시스템공학	3	최필환	팀원
프론트엔드	2019112567	멀티미디어공학	4	안상연	팀원

3. 제안서 보고 수행 일정

내 용	3월 4주	5주	4월 1주	4월 2주
주제 선정	■			
요구사항 분석		■		
UI 설계		■		
DB 설계		■		
수행계획 수립			■	
PPT 작성				■
발표연습				■
제안서 보고				■

<목 차>

제 1 장 프로젝트 개요

1. 프로젝트명	1
2. 프로젝트 목적	1
3. 기대효과	1
4. 프로젝트 라이선스	1

제 2 장 요구사항 분석

1. 요구사항 수집	2
2. 요구사항 명세서	6

제 3 장 프로젝트 설계

1. 오픈소스 선정 및 소개	7
2. 사용자 인터페이스 설계	11
3. DB 설계	13

제 4 장 프로젝트 수행 계획

1. 업무 분장	14
2. 개발 환경	14
3. 협업 규칙	15
4. 개발 일정	16

제 1 장 프로젝트 개요

1. 프로젝트명

동국대학교 LMS 내 협업 파일시스템 구축

2. 프로젝트 목적

동국대학교 학습관리시스템(LMS)인 ‘동국대학교 E-class’ 상에 팀 프로젝트를 위한 파일 스토리지 시스템을 구축하여, 팀 프로젝트 과정에서 생성된 파일의 이력 관리가 가능하게 한다.



[그림 1] 동국대학교 E-class UI

3. 기대효과

기존 카카오톡이나 클라우드 스토리지를 사용하여 팀 프로젝트 파일을 공유 시, 작성자, 수정 내용 파악이 어렵다. 또한 이전 버전으로 복구를 원할 시 복구하고자 하는 파일을 찾기 어렵고 기존 파일에 덮어썼을 때 복구가 불가능하다. 파일 스토리지 시스템을 구축하여 업로드일, 업로드자, 수정 메시지를 DB에 저장하면 이러한 문제점들을 해결하고 협업 능력을 올릴 수 있을 것이다.

4. 프로젝트 라이선스

오픈소스 프로젝트의 라이선스는 프로젝트에 사용된 컴포넌트 및 라이브러리의 라이선스에 따라 결정된다. 개발 환경과 사용되는 오픈소스의 라이선스를 분석하여 GPL v2 라이선스를 선정하였다. GPL v2는 오픈소스를 사용, 수정, 배포할 수 있으나, 수정된 소스 코드에도 동일한 GPL v2 라이선스를 적용해야 한다. 또한 소프트웨어를 상업적으로 이용하는 경우, 소스 코드를 공개해야 한다.

제 2 장 요구사항 분석

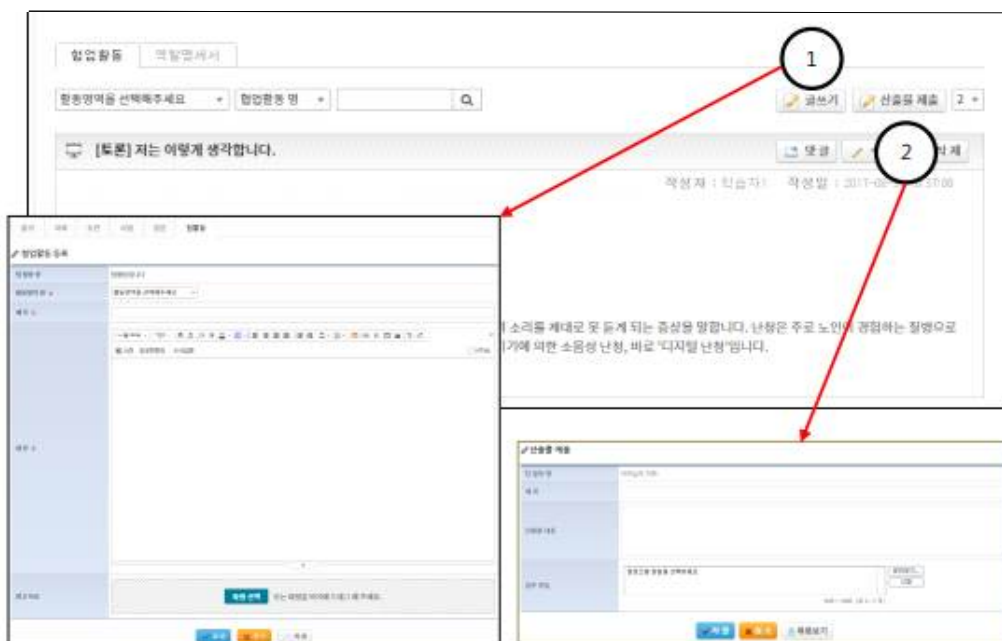
소프트웨어 기능과 제약조건의 명확화를 위해 현행 시스템 분석과 기존 사례 분석을 통해 요구사항을 수집하였다. 이를 통해 통해 파일 스토리지 시스템, 팀 구성 기능, 팀 내 공지사항 기능이 필요함을 파악하고 그에 따른 요구사항 명세서를 작성하였다.

1. 요구사항 수집

개발할 소프트웨어의 기능과 제약조건 등을 명확히 하기 위해 현행 시스템 분석과 기존 사례 분석을 통한 요구사항 수집을 시행하였다.

1.1 현행 시스템 분석

동국대 이클래스 상에 기존 팀 활동 영역을 분석하여 기능과 개선점을 파악하였다. 기존 팀 활동 영역은 [그림 2]와 같다. 기존 팀 활동 영역의 기능은 게시판 구조에서 글을 쓰고 댓글을 달거나, 파일을 업로드하는 기능과 팀장이 대표로 과제를 제출하는 기능으로 구성되어 있다. 또한 팀 구성을 교수 및 조교 권한으로만 가능하다는 제약이 있다. 협업에 필요한 파일시스템 기능과 학생들이 자율적으로 팀을 구성하는 기능을 도입하면 팀 활동 영역의 활용도를 높일 수 있을 것이다.



[그림 2] 동국대 E-class 팀 활동 영역 UI

1.2 기존 사례 분석

기존 팀 프로젝트에 진행에 주로 활용되는 플랫폼을 분석하여 개선점과 참고할 점을 정리하였다.

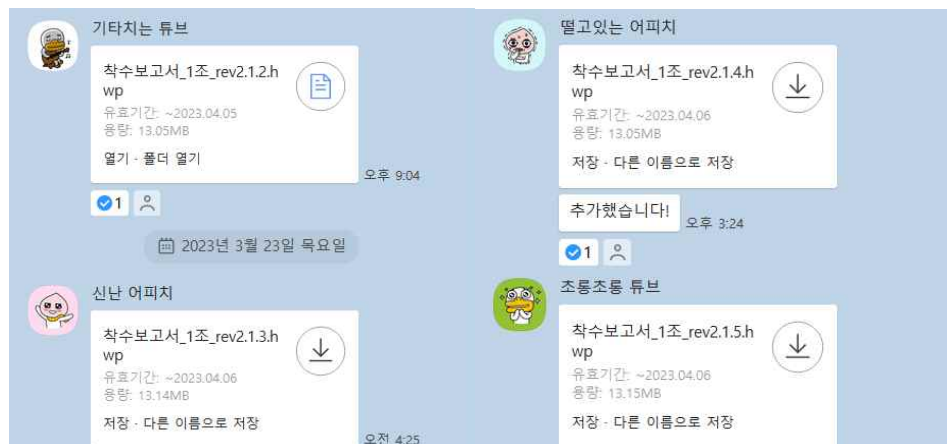
가. 카카오톡

- 기능

사람들과 메시지를 주고받을 수 있는 모바일 메신저 앱으로 기본 대화 기능과 더불어 파일, 영상, 사진 등을 첨부하는 기능, 보이스톡이라 불리는 전화 기능, 송금 기능 등 여러 가지 다양한 기능을 제공한다.

- 사용사례

한국에서는 메신저 앱 중 이용률이 압도적이다. 팀원 대부분이 이미 해당 서비스를 이용 중이고 익숙하다는 점에서 협업 시에도 카카오톡을 이용하는 경우가 많다. 하지만 협업을 위한 용도로 나온 서비스가 아니므로 타 협업툴에 비해 프로젝트 진행 과정이나 팀원들 간 업무 파악이 어려운 편이다.



[그림 3] 카카오톡 파일 공유 기능 사용 예시

- 개선점

카카오톡을 협업툴로 보았을 때 협업에서 중요한, 프로젝트 진행 상황을 정리해 보여주는 기능이나 자료(파일) 정리 기능이 없다는 단점이 있다. 또한 팀 구성 및 정보 공유에 팀원의 개인정보를 제공해야 하는 단점이 있다.

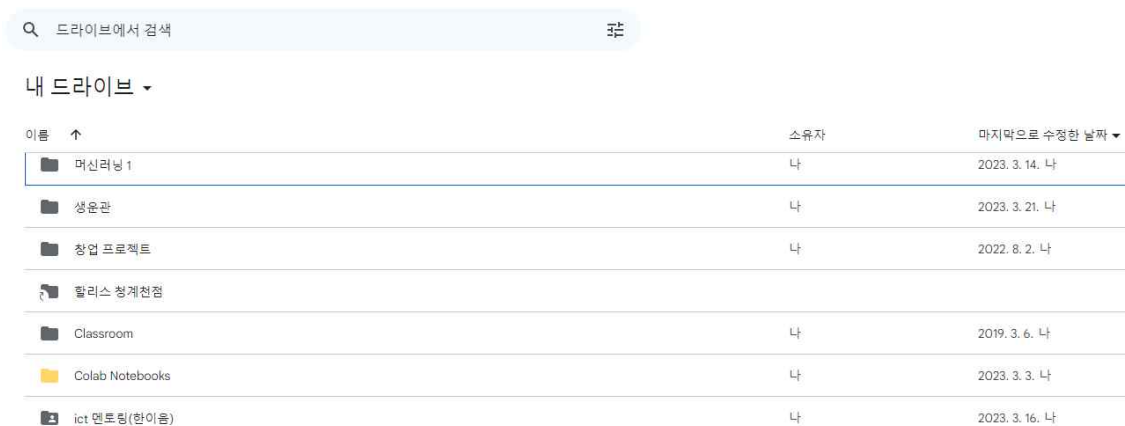
나. 구글 드라이브

- 기능

구글에서 제공하는 클라우드 기반 협업 도구이자 파일 저장/공유 서비스이다. 구글 드라이브는 구글 독스 에디터 오피스 제품군인 구글 문서 도구, 구글 시트, 구글 슬라이드와 연동되므로 문서, 스프레드시트, 프레젠테이션, 드로잉, 폼 등의 공동 편집이 가능하다.

- 사용사례

구글 드라이브에 파일 및 폴더를 업로드한 화면이다. 구글 드라이브 상에서 공동 편집이 가능하므로 수정한 날짜와 수정자가 명시된다. 또한, 로컬과 연결하여 동기화를 시킬 수도 있다.



이름	소유자	마지막으로 수정한 날짜
머신러닝 1	나	2023. 3. 14. 나
생윤관	나	2023. 3. 21. 나
창업 프로젝트	나	2022. 8. 2. 나
칼리스 청계천점		
Classroom	나	2019. 3. 6. 나
Colab Notebooks	나	2023. 3. 3. 나
ict 멘토링(한이름)	나	2023. 3. 16. 나

[그림 4] 구글 드라이브 사용 예시

- 개선점

구글 드라이브의 경우 이전 버전에 대한 관리는 30일간만 이루어진다. 학기 특성상 3~4달의 기간 동안 파일을 생성하고, 복구할 일이 많은 것을 고려하여 더 장기적인 버전관리가 지원되어야 한다. 또한, 공동 편집을 위해서는 구성원들의 이메일을 알아야 팀 구성이 가능하다. 이처럼 불필요하게 개인정보를 제공해야 한다.

다. Git&Github

- 기능

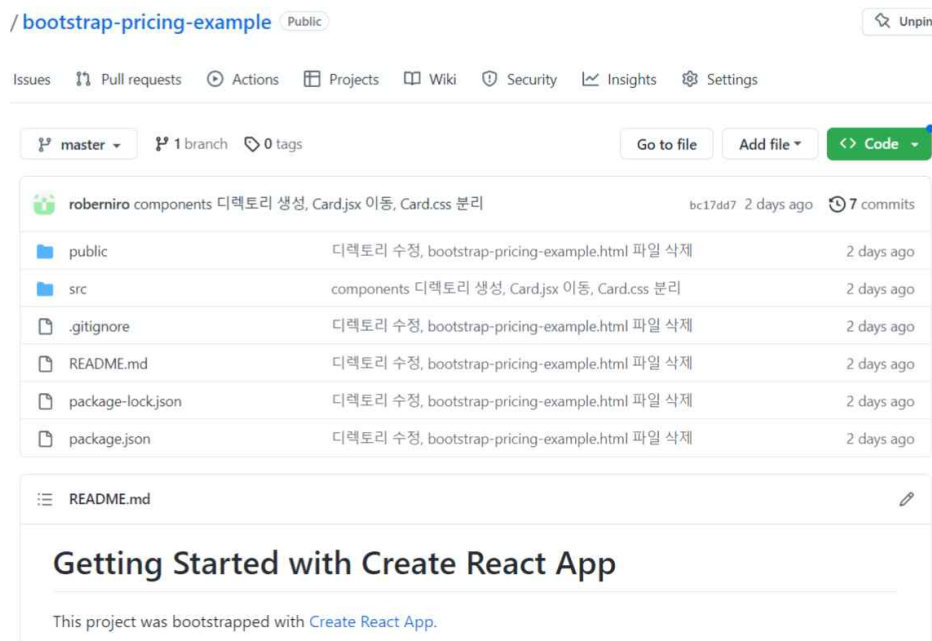
Git은 소스코드 분산 버전관리 시스템이며 Github는 Git 원격 저장소를 호스팅하는 웹 서비스이다. Git에서 사용하는 주요 명령어와 그 기능은 [표 1]과 같다.

[표 1] Git 명령어 및 기능

명령어	기능
add	이력 관리 대상 파일을 스테이징함
commit	스테이징한 파일을 로컬 리포지토리에 저장함
push	커밋한 내용을 원격 리포지토리에 업로드함
pull	원격 리포지토리에 저장한 내용을 로컬로 가져와 병합함
fetch	로컬 저장소에 원격 저장소의 변경사항을 확인해 가져옴
merge	분기된 커밋을 병합함

- 사용사례

비주얼 스튜디오 코드로 작성한 React 프로젝트를 Github 원격 리포지토리에 업로드한 화면 예시는 [그림 5]와 같다. 커밋 메시지를 통해 변경 내용을 명시하고 업로드한 시간이 기록되어 소스코드 이력 관리가 편리하다.



[그림 5] 깃허브 원격 저장소 사용 예시

– 개선점

기능이 소스코드 관리에 특화되어 있고, push 전 로컬 저장소에 커밋시 CLI에서 명령어를 입력하거나 별도의 IDE나 'Sourcetree'를 통해서만 GUI로 사용할 수 있는 점을 개선할 필요가 있다.

– 참고할 점

커밋시 커밋 메시지를 남길 수 있고, 해시를 커밋 id로 하여 자식 커밋은 부모 커밋의 해시 값을 저장한다. 이를 통해 수정 내용을 파악하기 쉽고, 파일의 변경 이력을 찾기 쉽게 한다.

1.3 결론

현행 시스템 분석과 사례 분석을 통해 수집한 요구사항은 [표 2]와 같다.

[표 2] 요구사항 수집 결과

분류	대상	문제점	요구사항
현행 시스템	E-class	파일 이력 관리▼	파일 스토리지 기능
		팀 구성 자율성▼	팀 구성 기능
기존 사례	카카오톡	파일 이력 관리▼	파일 스토리지 기능
		프라이버시 보호▼	팀 구성/공지사항 기능
	구글 드라이브	파일 이력 보존 기간▼	파일 스토리지 기능
	깃&깃허브	비전공자 접근성▼	파일 스토리지 기능

2. 요구사항 명세서

현행시스템 분석과 기존 사례 분석을 통해 파일 스토리지 시스템, 팀 구성 기능, 팀 내 공지사항 기능이 필요함을 파악하였다. 신규 시스템의 요구사항을 정리하여 [표 3]로 명세하였다.

[표 3] 요구사항 명세서

사용자	메뉴	필요 기능	기능 설명
학생	팀 활동	팀 구성	수강생 조회하여 팀 프로젝트 초대 발송한다.
		업로드	파일과 관련 정보를 업로드할 수 있다.
		다운로드	파일을 다운로드할 수 있다.
		삭제	특정 파일 및 전체 파일 삭제 기능이다.
		버전관리	이전 버전 조회 및 다운로드 가능하다.
		공지사항	팀 내의 공지사항을 등록하고 조회한다.
	메인 페이지	로그인/로그아웃	학번, 비밀번호로 접속 가능하게 한다.

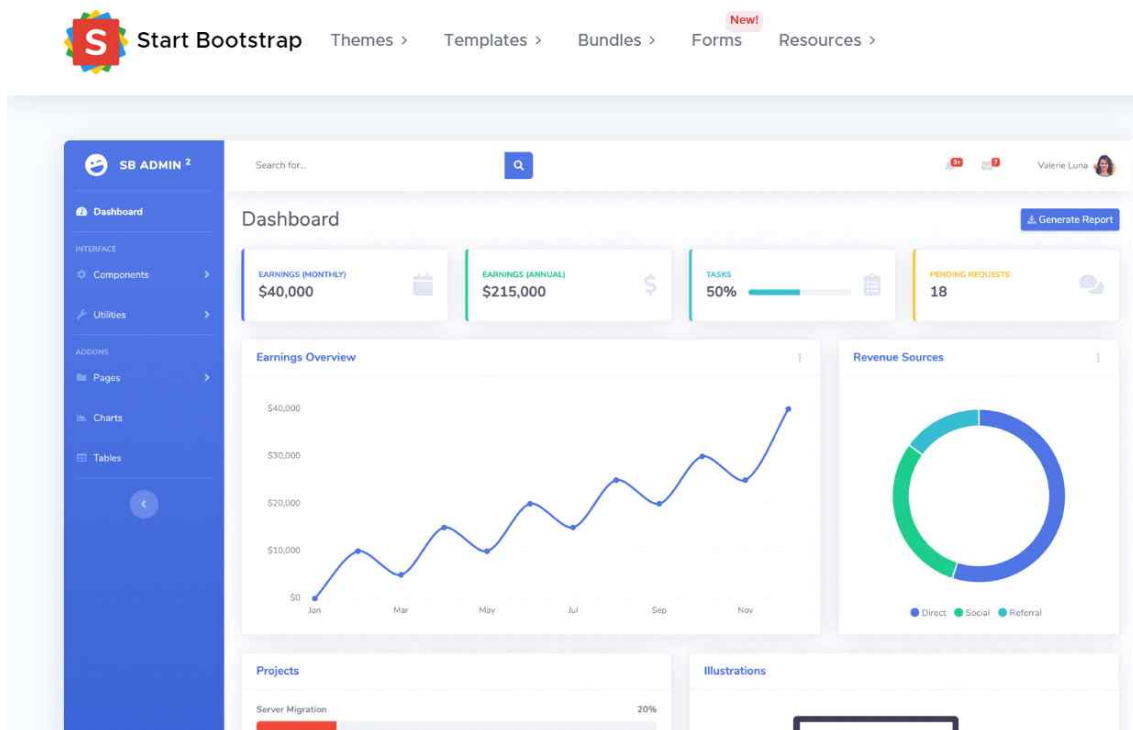
제 3 장 프로젝트 설계

앞서 분석한 요구사항을 기반으로 팀 활동 페이지 내 파일 관리 시스템 구축에 필요한 오픈소스 선정, 사용자 인터페이스 설계, DB설계를 진행하였다.

1. 오픈소스 선정 및 소개

1.1 프론트엔드

- Github: <https://github.com/startbootstrap/startbootstrap-sb-admin-2>
- 오픈소스 소개: 부트스트랩 템플릿을 무료로 제공하는 Start Bootstrap 사이트를 이용하여 관리자 대시보드 테마의 템플릿 오픈소스를 사용할 예정이다.



[그림 6] 프론트엔드 템플릿 예시

- 활용방안: 로그인 페이지는 웰컴 페이지 구현에 활용 가능하다. 파일 목록, 부가 서비스 제공을 위한 섹션 등은 팀 활동 페이지 구현에 활용할 수 있다.
- 라이선스 : MIT License

1.2 백엔드

가. 업로드

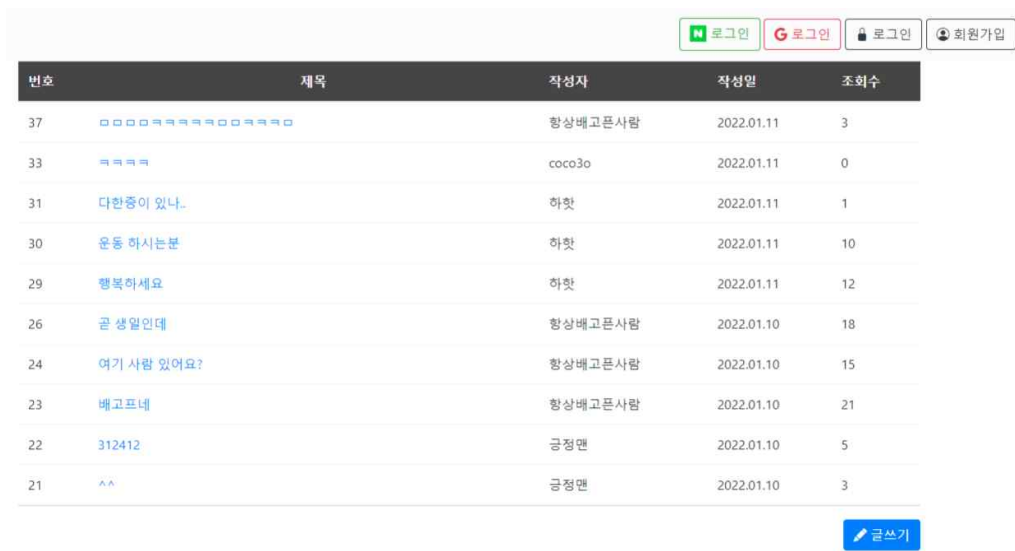
- 깃허브: <https://github.com/seungh0/spring-cloud-aws-s3>
- 오픈소스 소개: 본 오픈소스는 AWS s3를 이용하여 파일 업로드를 구현했다.
- 활용방안: 현재 오픈소스는 파일 업로드에 대한 부분만 구현이 되어 있다. 따라서 본 코드를 참고하여 폴더 및 파일의 업로드도 구현하려고 한다. 파일시스템 특성에 맞추어 다운로드, 삭제도 구현한다. 기존 이클래스와 다르게 버전관리가 가능한 이클래스를 만들 것이기 때문에, s3에서 이전 버전에 대한 파일을 가져올 로직도 구현한다.
- 라이선스: 프로젝트 라이선스가 명시되어 있지 않으나 사용된 라이선스는 [표 4]와 같다. 이를 바탕으로 해당 오픈소스의 라이선스는 GPL v2 라이선스로 판단할 수 있다.

[표 4] 파일 업로드 오픈소스 사용 라이선스

분야	SW	라이선스
언어	JAVA(Corretto JDK11)	GPL v2 라이선스
프레임워크	Spring Boot	아파치 라이선스 2.0
빌드도구	Gradle	아파치 라이선스 2.0

나. 로그인

- 깃허브: <https://github.com/hojunnnnnn/board>
- 오픈소스 소개: Spring Boot로 구현한 게시판 프로젝트로 Security 회원가입 및 로그인, OAuth 2.0 구글, 네이버 로그인, 회원 정보 수정, 회원가입 시 유효성 검사 및 중복 검사 기능, 게시판 CRUD 기능, 댓글 CRUD 기능이 있다.



번호	제목	작성자	작성일	조회수
37	oooooooooooooooooooo	항상배고픈사람	2022.01.11	3
33	ㅋㅋㅋㅋ	coco3o	2022.01.11	0
31	다한증이 있냐..	하핫	2022.01.11	1
30	운동 하시는분	하핫	2022.01.11	10
29	행복하세요	하핫	2022.01.11	12
26	곧 생일인데	항상배고픈사람	2022.01.10	18
24	여기 사람 있어요?	항상배고픈사람	2022.01.10	15
23	배고프네	항상배고픈사람	2022.01.10	21
22	312412	금정맨	2022.01.10	5
21	^^	금정맨	2022.01.10	3

[그림 7] 게시판 오픈소스 UI

- 활용 방안: 메인 페이지의 로그인/로그아웃 기능을 구현하는 데 Security 회원가입 및 로그인 기능을 활용할 수 있다.
- 라이선스: 프로젝트 라이선스가 명시되어 있지 않으나 사용된 라이선스는 [표 5]와 같다. 이를 바탕으로 해당 오픈소스의 라이선스는 GPL v2 라이선스로 판단할 수 있다.

[표 5] 게시판 오픈소스 사용 라이선스

분야	SW	라이선스
언어	JAVA(Corretto JDK11)	GPL v2 라이선스
프레임워크	Spring Boot	아파치 라이선스 2.0
DB	MySQL	GPL v2 라이선스
빌드도구	Gradle	아파치 라이선스 2.0

다. 공지사항

- 깃허브: <https://github.com/rgl-za/ToDo>
- 오픈소스 소개: Bootstrap기반 프론트엔드 템플릿에 Spring Boot를 이용해 백엔드를 구현한 Todo-List로 할 일에 대한 CRUD가 가능하다.

Todo App

No.	Todo item	status	Actions
0	안녕하세요	N	<div style="display: inline-block; background-color: #dc3545; color: white; padding: 5px 10px; margin-right: 5px;">Delete</div> <div style="display: inline-block; background-color: #28a745; color: white; padding: 5px 10px;">Finished</div>
1	안녕	N	<div style="display: inline-block; background-color: #dc3545; color: white; padding: 5px 10px; margin-right: 5px;">Delete</div> <div style="display: inline-block; background-color: #28a745; color: white; padding: 5px 10px;">Finished</div>

[그림 8] Todo-List 오픈소스 UI

- 활용 방안: 팀 활동 페이지의 프로젝트 공지사항 기능 구현에 활용할 수 있다.
- 라이선스: 프로젝트 라이선스가 명시되어 있지 않으나 사용된 라이선스는 [표 6]와 같다. 이를 바탕으로 해당 오픈소스의 라이선스는 GPL v2 라이선스로 판단할 수 있다.

[표 6] Todo-List 사용 라이선스

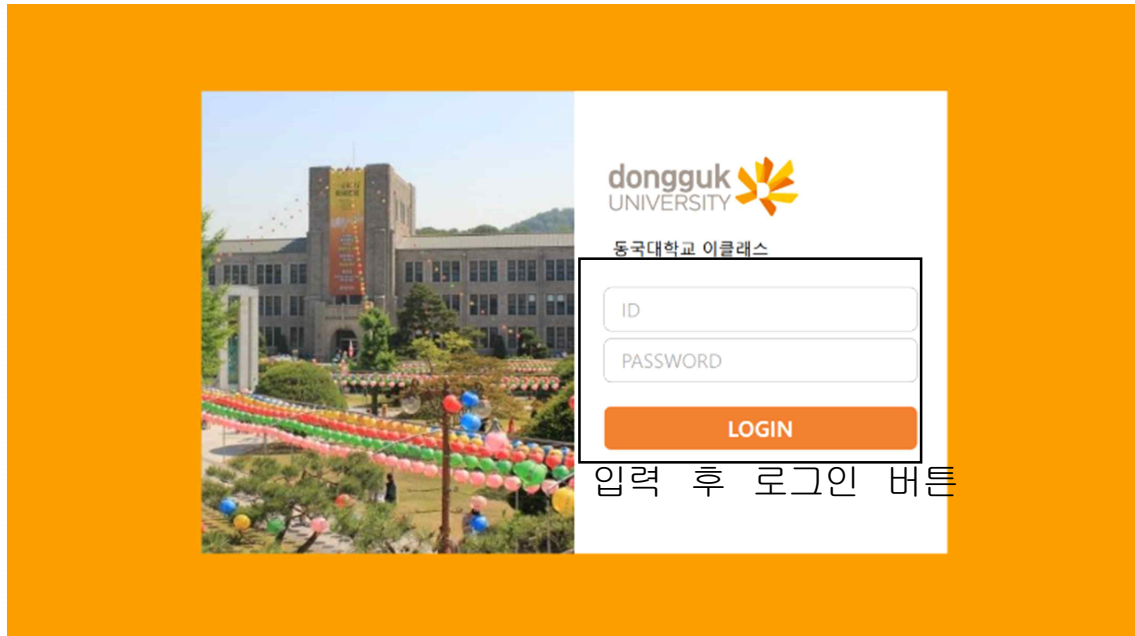
분야	SW	라이선스
언어	JAVA(Corretto JDK11)	GPL v2 라이선스
프레임워크	Spring Boot	아파치 라이선스 2.0
DB	MySQL	GPL v2 라이선스
빌드도구	Gradle	아파치 라이선스 2.0

2. 사용자 인터페이스 설계

앞서 정의한 요구사항을 기반으로 사용자 인터페이스를 설계하였다.

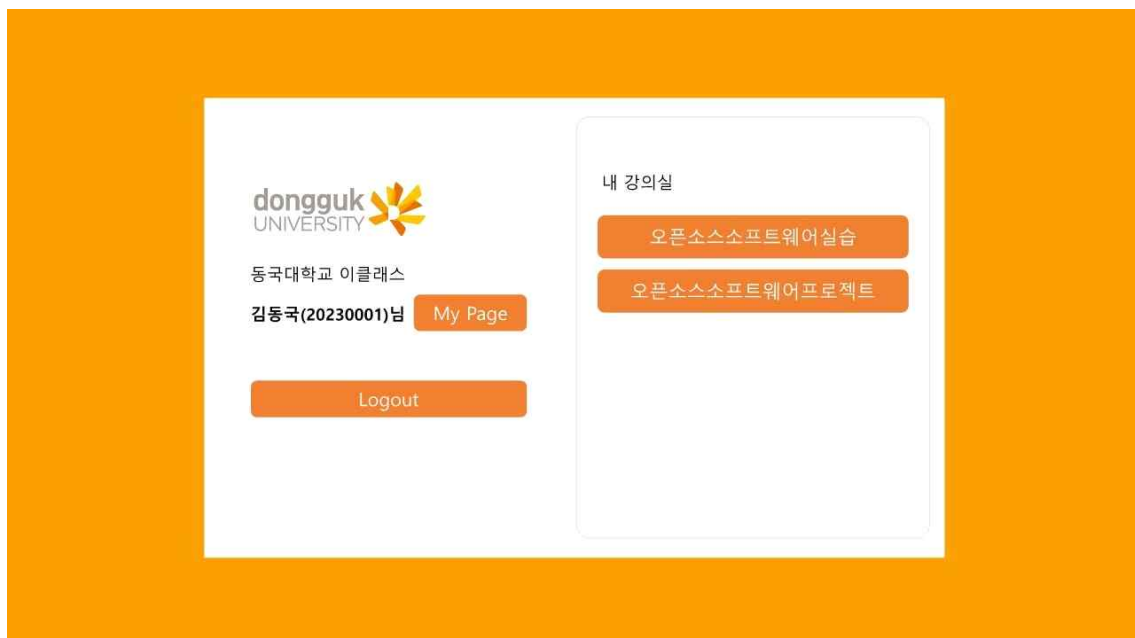
2.1 웰컴 페이지

웰컴 페이지로 로그인 화면을 구상하였다.



2.2 과목 선택 페이지

로그인한 사용자가 과목을 선택할 수 있다.



제 3 장 프로젝트 설계

2.3 팀 활동 페이지

파일 저장, 다운로드, 버전관리 버튼이 있는 파일 스토리지가 있다. 추가로 팀 구성 정보 확인 및 팀원 초대가 가능한 섹션과 공지사항을 등록하고 확인할 수 있는 섹션이 있다.



3. DB 설계

요구사항 명세와 사용자 인터페이스 설계를 바탕으로 필요한 테이블과 속성을 정의해보았다. 속성과 테이블 매핑 관계 관점에서 설명하겠다.

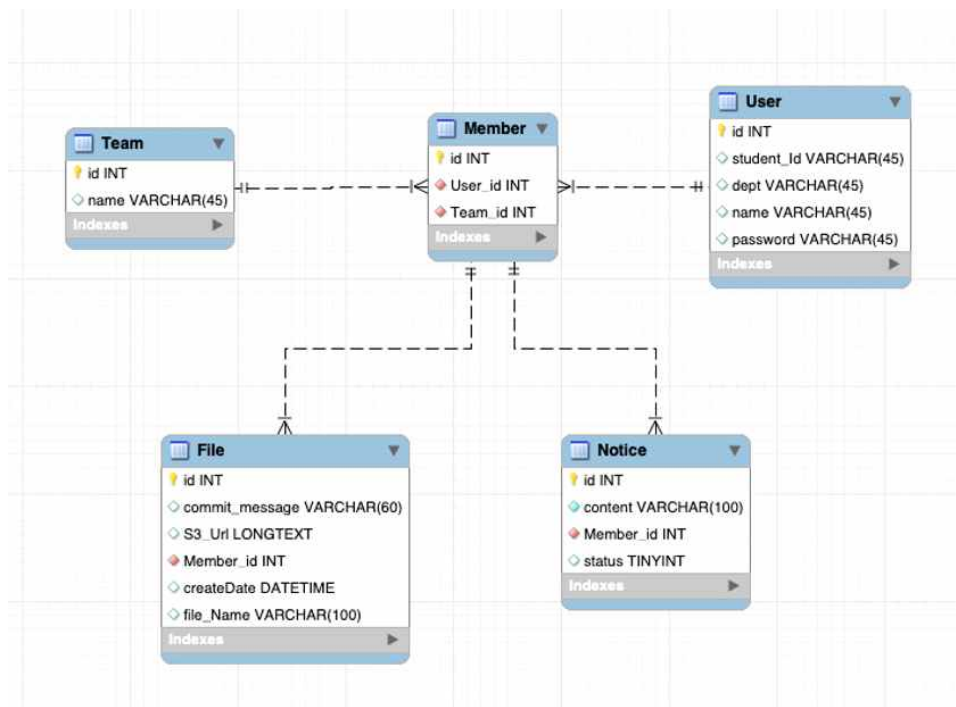
3.1 속성

- User: 로그인에 필요한 id, password를 속성으로 가지고 있고, 학생 정보인 dept(학과)와 name도 가지고 있다.
- Team: 팀 구성에 필요한 릴레이션으로 name(팀이름)을 가지고 있다.
- Member: User와 Team의 중간 매핑 테이블로 각각의 테이블인 User와 Team의 FK를 가지고 있다.
- File: 버전관리를 위한 S3_url과 commit_message가 있다. 외래키로 Member_id를 가지고 있다.
- Notice: 공지사항으로 수행 여부를 판단하는 status가 있다.

3.2 테이블 매핑 관계

- User, Team은 다대다 관계이므로, 중간테이블은 Member를 통해 일대다로 만들어준다.
- File과 Notice는 중간테이블인 Member 테이블과 다대일 관계이다.

3.3 ER Diagram



제 4 장 프로젝트 수행 계획

프로젝트 수행을 위한 업무분장과 협업 규칙을 작성하고, 개발 일정을 수립하였다.

1. 업무분장

이름	역할
민한결	로그인, 팀 구성, 공지 사항, 버전관리 기능 구현
안상연	웰컴 페이지, 팀 활동 페이지 화면 구성
최필환	파일 업로드, 다운로드, 삭제 구현

2. 개발 환경

개발 환경을 SW/HW로 나누어 [표 7], [표 8]로 정리하였다.

[표 7] SW 환경

분야	세부 분야	SW	라이선스
운영체제		Ubuntu 22.04.1 LTS	GPL
백엔드	언어	JAVA(Corretto JDK11)	GPL v2 라이선스
	프레임워크	Spring Boot 2.7.10	아파치 라이선스 2.0
	IDE	IntelliJ	아파치 라이선스 2.0
	DB	Maria DB	GPL v2 라이선스
	빌드도구	Gradle	아파치 라이선스 2.0
프론트엔드	언어	HTML, CSS, JavaScript	-
	디자인시스템	BootStrap	MIT 라이선스
	IDE	Visual Studio Code	MIT 라이선스
스토리지	클라우드	AWS	AWS 라이선스

[표 8] HW 환경

팀원	프로세서	메모리
민한결	Intel® Core™ i5-1240P Processor 12M Cache, up to 4.40 GHz	16GB
안상연	Intel® Core™ i5-8265U CPU @ 1.60GHz 1.80 GHz	8GB
최필환	6/8코어 Apple Firestorm 0.6 ~ 3.23 GHz + 2코어 Apple Icestorm 0.6 ~ 2.06 GHz	16GB

3. 협업 규칙

팀원과 협의를 통해 아래와 같은 협업 규칙을 정하였다.

- CSID 조직상의 원격 저장소를 clone
- 각자 local에서 수정 후, add, commit 및 push 후 Pull Request
- 기능 단위의 branch 만들어 작업 수행
- 코드 리뷰 후 문제가 없으면 merge
- commit 단위는 하나 이상의 의미 가지지 않음
- commit 양식

[날짜, 이름] (type) 설명

타입	설명
(Add) 기능 및 코드	기능 및 코드 추가
(Mod) 기능 및 코드	정상 작동 기능 및 코드 수정
(Fix) 코드	오류가 있는 코드 수정
(Delete) 기능 및 코드	필요 없는 기능 및 코드 삭제
(Docs) 문서	문서 업로드 및 수정

예시) [23.04.03. Hankyul] (Add) todo

- 정기회의: 매 수업 시간 후 30분

4. 개발 일정

분류	내용	담당	4월 4주	5주	5월 1주	2주	3주	4주	6월 1주	2주	3주
오픈소스 분석	로그인 기능 분석	민한결									
	프론트 템플릿 분석	안상연									
	파일 저장 기능 분석	최필환									
웰컴 페이지	로그인 기능	민한결									
	화면 구현	안상연									
인프라 설정	AWS 환경 설정 / 버킷 설정	최필환									
팀 활동	팀 구성 / 공지사항	민한결									
	화면 구현	안상연									
	파일 업로드, 다운로드, 삭제	최필환									
	파일 버전관리	민한결(정)									
	화면 구현	안상연									
	파일 버전관리	최필환(부)									
최종 수정 및 발표		모두									