



SKKU GROUP PROJECT(스꾸팀플)

테스트 플랜 요구사항 명세서

2022.11.13.

Introduction to Software Engineering 42

TEAM 12

팀장 정일원

팀원 강대호

김서연

류도현

안현준

엄계현

목차

1. Introduction.....	10
1.1. Purpose	10
1.2. Scope	10
1.3. Definitions, Acronyms, and Abbreviation	11
1.3.1. Terms and Definitions	11
1.3.2. Acronyms and Abbreviations	12
1.4. References	13
1.5. Overview	13
2. Approach	13
2.1. Test methods	14
2.1.1. Software Unit Test Methods.....	14
2.1.1.1. Objective.....	14
2.1.1.2. Function List.....	14
2.1.2. Software Interface Test Methods.....	25
2.1.2.1. Objective.....	25
2.1.2.2. Function List.....	25
2.1.3. Testing Tools	26
2.1.3.1. Postman 프로그램(Free Version)	26
2.1.3.2. Chrome 웹 브라우저	27
2.1.3.3. MySQL Workbench(8.0 Community Edition)	28
2.1.3.4. AWS S3 콘솔	28
2.1.3.5. Notion(노션)	29
3. Software Unit Test.....	30
3.1. 회원가입	30
3.1.1. 시나리오.....	30
3.1.2. 일반적인 성공 케이스.....	30
3.1.3. 필수 정보가 부족한 경우.....	30
3.1.4. 성균관대학교 이메일이 아닌 경우	31
3.1.5. 입력한 아이디가 사용 중인 경우.....	31
3.1.6. 입력한 이메일이 이미 사용 중인 경우.....	31
3.2. 로그인	31
3.2.1. 시나리오.....	31

3.2.2. 일반적인 성공 케이스.....	32
3.2.3. 아이디와 비밀번호 중 하나라도 입력되지 않은 경우.....	32
3.2.4. 사용자 아이디가 등록되지 않은 경우.....	32
3.2.5. 비밀번호가 틀린 경우.....	33
3.3. 사용자 리스트 조회	33
3.3.1. 시나리오.....	33
3.3.2. 일반적인 성공 케이스(검색어가 있을 때).....	33
3.3.3. 일반적인 성공 케이스(검색어가 없을 때).....	33
3.3.4. 페이지 사이즈나 페이지 순서 번호가 없는 경우	34
3.3.5. 결과가 빈 리스트인 경우(검색어가 있을 때).....	34
3.3.6. 페이지 범위가 잘못된 경우(검색어가 있을 때).....	34
3.3.7. 결과가 빈 리스트인 경우(검색어가 없을 때).....	35
3.3.8. 페이지 범위가 잘못된 경우(검색어가 없을 때).....	35
3.4. 내 정보 페이지 조회	35
3.4.1. 시나리오.....	35
3.4.2. 일반적인 성공 케이스.....	35
3.4.3. 사용자 인증에 실패한 케이스.....	36
3.5. 사용자 세부 정보 조회	36
3.5.1. 시나리오.....	36
3.5.2. 일반적인 성공 케이스.....	36
3.5.3. 찾으려는 사용자가 데이터베이스에 없을 때	37
3.6. 내 정보 수정	37
3.6.1. 시나리오.....	37
3.6.2. 일반적인 성공 케이스.....	37
3.6.3. 아무것도 수정한 것이 없을 때.....	38
3.6.4. 사용자 인증에 실패한 경우.....	38
3.6.5. 변경하려는 이메일이 성균관대학교 이메일이 아닌 경우	38
3.7. 회원 탈퇴	39
3.7.1. 시나리오.....	39
3.7.2. 일반적인 성공 케이스(사용자 프로필 이미지가 기본 이미지가 아닐 경우)...39	
3.7.3. 일반적인 성공 케이스(사용자 프로필 이미지가 기본 이미지일 경우).....39	
3.7.4. 사용자 인증에 실패한 경우.....	40
3.8. 프로젝트 리스트 조회	40

3.8.1. 시나리오.....	40
3.8.2. 일반적인 성공 케이스(인증은 없고 검색어가 있을 경우, 정렬 옵션 (Favorite.desc))	40
3.8.3. 일반적인 성공 케이스(인증은 없고 검색어가 없을 경우, 정렬 옵션 (Favorite.asc))	41
3.8.4. 일반적인 성공 케이스(사용자가 작성한 것만 조회하는 옵션)	41
3.8.5. 일반적인 성공 케이스(사용자가 즐겨 찾기 등록한 것만 조회하는 옵션)	41
3.8.6. 일반적인 성공 케이스(사용자가 참여 신청한 것만 조회하는 옵션)	42
3.8.7. 페이지 사이즈나 페이지 순서 번호가 없는 경우	42
3.8.8. 조회 결과 전체 리스트가 비어 있는 경우	42
3.8.9. 페이지 범위가 실제 가능 페이지 범위를 벗어난 경우	42
3.9. 프로젝트 등록	43
3.9.1. 시나리오	43
3.9.2. 일반적인 성공 케이스	43
3.9.3. 필수 입력 사항이 주어지지 않은 경우	43
3.9.4. 사용자 인증이 되지 않은 경우	44
3.10. 프로젝트 랜덤 조회	44
3.10.1. 시나리오	44
3.10.2. 일반적인 성공 케이스	44
3.10.3. 성공 케이스(랜덤 뽑기할 프로젝트 수가 음수인 경우)	45
3.10.4. 성공 케이스(랜덤 뽑기할 프로젝트 수가 3보다 큰 경우)	45
3.10.5. 성공 케이스(전체 프로젝트 개수가 뽑기 조회 개수보다 작은 경우)	45
3.11. 프로젝트 세부 정보 조회	45
3.11.1. 시나리오	45
3.11.2. 일반적인 성공 케이스(사용자 인증 안 된 경우)	46
3.11.3. 일반적인 성공 케이스(사용자 인증된 경우, 해당 프로젝트가 즐겨 찾기로 등록된 경우)	46
3.11.4. 일반적인 성공 케이스(사용자 인증된 경우, 해당 프로젝트에 신청서를 제 출한 경우)	47
3.12. 프로젝트 삭제	47
3.12.1. 시나리오	47
3.12.2. 일반적인 성공 케이스	47
3.12.3. 사용자 인증에 실패한 경우	48
3.12.4. 사용자가 프로젝트 관리 권한이 없는 경우	48

3.13. 프로젝트 정보 수정	48
3.13.1. 시나리오	48
3.13.2. 일반적인 성공 케이스(전부 변경하는 경우).....	49
3.13.3. 사용자 인증에 실패한 경우.....	49
3.13.4. 사용자가 프로젝트 수정 권한이 없는 경우.....	50
3.14. 프로젝트 즐겨 찾기 리스트에 추가	50
3.14.1. 시나리오	50
3.14.2. 일반적인 성공 케이스	50
3.14.3. 사용자 인증에 실패한 경우.....	51
3.14.4. 중복된 즐겨 찾기 추가 요청인 경우.....	51
3.15. 프로젝트 즐겨찾기 리스트에서 삭제	51
3.15.1. 시나리오	51
3.15.2. 일반적인 성공 케이스	52
3.15.3. 사용자 인증에 실패한 경우.....	52
3.15.4. 중복된 즐겨 찾기 삭제 요청인 경우.....	52
3.16. 프로젝트 참여 신청자 리스트 조회	53
3.16.1. 시나리오	53
3.16.2. 일반적인 성공 케이스	53
3.16.3. 프로젝트 고유 번호가 주어지지 않은 경우.....	53
3.16.4. 사용자 인증에 실패한 경우.....	54
3.16.5. 사용자가 관리자 권한이 없는 경우.....	54
3.16.6. 페이지 사이즈나 페이지 순서 번호가 주어지지 않은 경우.....	54
3.16.7. 프로젝트 참여 신청자가 아무도 없을 때.....	54
3.17. 프로젝트 참여 수락/거절 여부 변경	55
3.17.1. 시나리오	55
3.17.2. 일반적인 성공 케이스(신청 보류 옵션).....	56
3.17.3. 일반적인 성공 케이스(신청 승인 옵션).....	56
3.17.4. 일반적인 성공 케이스(신청 거절 옵션).....	56
3.17.5. 필수 입력 사항이 부족한 경우.....	57
3.17.6. 사용자 인증에 실패한 경우.....	57
3.17.7. 사용자에게 프로젝트 관리 권한이 없는 경우.....	57
3.17.8. 처리할 수 없는 옵션이 주어진 경우.....	58
3.18. 프로젝트 참여 신청하기	58
3.18.1. 시나리오	58

3.18.2.	일반적인 성공 케이스	58
3.18.3.	프로젝트 고유 번호가 주어지지 않은 경우	59
3.18.4.	사용자 인증에 실패한 경우	59
3.18.5.	이미 프로젝트 참여 신청서를 등록한 경우	59
3.19.	프로젝트 참여 신청 취소하기	60
3.19.1.	시나리오	60
3.19.2.	일반적인 성공 케이스	60
3.19.3.	사용자 인증에 실패한 경우	60
3.19.4.	프로젝트에 제출한 참여 신청서가 없는 경우	61
4.	Software Interface Test.....	61
4.1.	MySQL(데이터베이스 Sequelize 라이브러리 연동 검증)	62
4.1.1.	MySQL 데이터베이스 테이블 스키마 생성	62
4.1.1.1.	사용할 빈 데이터베이스가 생성된 경우	62
4.1.1.2.	사용할 데이터베이스가 생성되지 않은 경우	62
4.1.1.3.	이전에 사용한 데이터베이스가 생성되어 있는 경우	63
4.1.2.	MySQL 데이터베이스 정보 조회	63
4.1.2.1.	조건을 충족하는 정보가 있을 경우	63
4.1.2.2.	조건을 충족하는 정보가 없을 경우	64
4.1.3.	MySQL 데이터베이스 정보 삽입	64
4.1.3.1.	삽입된 정보가 제한 크기를 넘는 경우	64
4.1.3.2.	정상적인 경우	65
4.1.4.	MySQL 데이터베이스 정보 수정	65
4.1.4.1.	수정될 정보가 제한 크기를 넘는 경우	65
4.1.4.2.	정상적인 경우	66
4.2.	브라우저 쿠키 설정	66
4.2.1.	쿠키 생성	66
4.2.1.1.	쿠키 키를 적지 않은 경우	66
4.2.1.2.	정상적인 경우	67
4.2.2.	쿠키 삭제하기	67
4.2.2.1.	쿠키 키를 적지 않은 경우	67
4.2.2.2.	정상적인 경우	68
4.2.3.	쿠키 사용하기	68
4.2.3.1.	쿠키 유효 기간이 지난 경우	68

4.2.3.2. 정상적인 경우	68
4.3. 업로드 이미지 관리	69
4.3.1. 이미지 파일 생성	69
4.3.1.1. 이미지 크기 제한보다 큰 경우	69
4.3.1.2. 이미지 확장자가 지정된 확장자가 아닌 경우	70
4.3.1.3. 정상적인 경우	70
4.3.2. 이미지 파일 불러오기	70
4.3.2.1. 해당 경로에 이미지가 없는 경우	70
4.3.2.2. 정상적인 경우	71
4.3.3. 이미지 파일 삭제	71
4.3.3.1. 해당 경로에 이미지가 없는 경우	71
4.3.3.2. 정상적인 경우	72
5. Supporting Information	72
5.1. Test Environment	72
5.2. Document History	73

그림 목차

[그림 1] Postman 프로그램	26
[그림 2] Chrome Cookie-Editor	27
[그림 3] WySQL Workbench	28
[그림 4] AWS S3	28
[그림 5] Notion	29

표 목차

[표 1] 용어 정리	11
[표 2] 약어 정의	12
[표 3] 회원가입 - 성공	30
[표 4] 회원가입 - 필수정보 부족	30
[표 5] 회원가입 - 이메일 오류	31
[표 6] 회원가입 - 아이디 중복	31
[표 7] 회원가입 - 이메일 중복	31
[표 8] 로그인 - 성공	32

[표 9] 로그인 - 입력값 부족.....	32
[표 10] 로그인 - 미등록 아이디	32
[표 11] 로그인 - 비밀번호 오류	33
[표 12] 사용자 리스트 조회 - 성공	33
[표 13] 사용자 리스트 조회(검색어 미포함) - 성공	33
[표 14] 사용자 리스트 조회 - 입력값 부족	34
[표 15] 사용자 리스트 조회 - 빈 리스트	34
[표 16] 사용자 리스트 조회 - 페이지 범위 오류	34
[표 17] 사용자 리스트 조회(검색어 미포함) - 빈 리스트	35
[표 18] 사용자 리스트 조회(검색어 미포함) - 페이지 범위 오류	35
[표 19] 본인 정보 조회 - 성공	35
[표 20] 본인 정보 조회 - 사용자 인증 실패	36
[표 21] 사용자 정보 조회 - 성공	36
[표 22] 사용자 정보 조회 - 미등록 사용자	37
[표 23] 본인 정보 수정 - 성공	37
[표 24] 본인 정보 수정 - 수정 내용 없음	38
[표 25] 본인 정보 수정 - 사용자 인증 실패	38
[표 26] 본인 정보 수정 - 이메일 오류	38
[표 27] 회원 탈퇴 - 성공	39
[표 28] 회원 탈퇴 (기본 이미지) - 성공	39
[표 29] 회원 탈퇴 - 사용자 인증 실패	40
[표 30] 프로젝트 리스트 조회 - 성공	40
[표 31] 프로젝트 리스트 조회 (검색어 미포함) - 성공	41
[표 32] 사용자 프로젝트 리스트 조회 - 성공	41
[표 33] 사용자 즐겨찾기 리스트 조회 - 성공	41
[표 34] 사용자 신청 프로젝트 리스트 조회 - 성공	42
[표 35] 프로젝트 리스트 조회 - 입력값 부족	42
[표 36] 프로젝트 리스트 조회 - 빈 리스트	42
[표 37] 프로젝트 리스트 조회 - 잘못된 페이지 접근	42
[표 38] 프로젝트 등록 - 성공	43
[표 39] 프로젝트 등록 - 필수정보 부족	43
[표 40] 프로젝트 등록 - 사용자 인증 실패	44
[표 41] 프로젝트 랜덤 조회 - 성공	44
[표 42] 프로젝트 랜덤 조회 (0개 미만) - 성공	45
[표 43] 프로젝트 랜덤 조회 (4개 이상) - 성공	45
[표 44] 전체 프로젝트 개수 이상의 프로젝트 랜덤 조회 - 성공	45
[표 45] 프로젝트 정보 조회 (사용자 인증 없음) - 성공	46
[표 46] 즐겨찾는 프로젝트 정보 조회 - 성공	46
[표 47] 신청한 프로젝트 정보 조회 - 성공	47
[표 48] 프로젝트 삭제 - 성공	47
[표 49] 프로젝트 삭제 - 사용자 인증 실패	48
[표 50] 프로젝트 삭제 - 권한 없음	48
[표 51] 프로젝트 수정 - 성공	49

[표 52]	프로젝트 수정 - 사용자 인증 실패	49
[표 53]	프로젝트 수정 - 권한 없음	50
[표 54]	프로젝트 즐겨찾기 - 성공	50
[표 55]	프로젝트 즐겨찾기 - 사용자 인증 실패	51
[표 56]	프로젝트 즐겨찾기 - 중복된 요청	51
[표 57]	프로젝트 즐겨찾기 삭제 - 성공	52
[표 58]	프로젝트 즐겨찾기 삭제 - 사용자 인증 실패	52
[표 59]	프로젝트 즐겨찾기 삭제 - 중복된 요청	52
[표 60]	프로젝트 신청자 리스트 조회 - 성공	53
[표 61]	프로젝트 신청자 리스트 조회 - 입력값 부족 (프로젝트 번호)	53
[표 62]	프로젝트 신청자 리스트 조회 - 사용자 인증 실패	54
[표 63]	프로젝트 신청자 리스트 조회 - 권한 없음	54
[표 64]	프로젝트 신청자 리스트 조회 - 입력값 부족 (페이지)	54
[표 65]	프로젝트 신청자 리스트 조회 - 신청자 없음	54
[표 66]	프로젝트 신청서 보류 - 성공	56
[표 67]	프로젝트 신청서 승인 - 성공	56
[표 68]	프로젝트 신청서 거절 - 성공	56
[표 69]	프로젝트 신청서 거절 - 입력값 부족	57
[표 70]	프로젝트 신청서 거절 - 사용자 인증 실패	57
[표 71]	프로젝트 신청서 거절 - 권한 없음	57
[표 72]	프로젝트 신청서 - 승인, 보류, 거절 외의 접근	58
[표 73]	프로젝트 참여 신청 - 성공	58
[표 74]	프로젝트 참여 신청 - 입력값 부족	59
[표 75]	프로젝트 참여 신청 - 사용자 인증 실패	59
[표 76]	프로젝트 참여 신청 - 중복된 신청	59
[표 77]	프로젝트 참여 신청 취소 - 성공	60
[표 78]	프로젝트 참여 신청 취소 - 사용자 인증 실패	60
[표 79]	프로젝트 참여 신청 취소 - 신청서 없음	61
[표 80]	DB 테이블 스키마 생성 - 성공	62
[표 81]	DB 테이블 스키마 생성 - 실패	62
[표 82]	DB 테이블 스키마 생성 (기존 사용 DB 존재) - 성공	63
[표 83]	DB 정보 조회 - 성공	63
[표 84]	DB 정보 조회 - 정보 없음	64
[표 85]	DB 정보 삽입 - 제한 크기 초과	64
[표 86]	DB 정보 삽입 - 성공	65
[표 87]	DB 정보 수정 - 제한 크기 초과	65
[표 88]	DB 정보 수정 - 성공	66
[표 89]	쿠키 생성 - 쿠키 키 없음	66
[표 90]	쿠키 생성 - 성공	67
[표 91]	쿠키 삭제 - 쿠키 키 없음	67
[표 92]	쿠키 삭제 - 성공	68
[표 93]	쿠키 사용 - 쿠키 유효 기간 초과	68
[표 94]	쿠키 사용 - 성공	68

[표 95] 이미지 파일 생성 - 크기 제한 초과	69
[표 96] 이미지 파일 생성 - 확장자 오류	70
[표 97] 이미지 파일 생성 - 성공	70
[표 98] 이미지 파일 요청 - 이미지 없음	70
[표 99] 이미지 파일 요청 - 성공	71
[표 100] 이미지 파일 삭제 - 이미지 없음	71
[표 101] 이미지 파일 삭제 - 성공	72
[표 102] 웹브라우저 테스트 환경	72
[표 103] DB 테스트 환경	72
[표 104] 문서 이력	73

1. Introduction

Introduction 장은 본 Test Plan 문서의 목적, 문서의 범위와 테스트의 범위, 문서를 작성하며 사용한 낱어, 참조한 문서, Test Plan 문서의 전체적인 내용에 대해 짧막하게 설명한다.

1.1. Purpose

본 Test Plan 문서는 기본적으로 Software Requirement Specification에서 계획한 테스트를 구체적으로 작성하기 위한 문서이다. Test Plan 문서는 테스트 진행 중 테스트 방향성과 각 기능의 목적에 맞는 테스트 케이스 생성을 통해 더 효율적인 테스트와 팀원들 사이 원활한 테스트 분업을 돕는다. Test Plan 문서를 기본으로 체계적인 테스트 단계 진행과 의사소통이 가능하다.

1.2. Scope

본 Test Plan 문서에는 우선 SKKU Group Project(스꾸 팀플)에서 테스트를 진행할 때 필요한 기본적인 정보를 제공해 준다. 이 정보에는 테스트할 구현한 기능들과 기존 소프트웨어 시스템과의 연동 기능, 테스트를 위해 사용할 소프트웨어 시스템, 각 테스트 케이스 발생 시나리오, 각 테스트 케이스의 입력 값과 출력 값이 포함된다. 테스트는 크게 Software Unit Test와 Software Interface Test 두 가지로 나누어 진행한다. Software Unit Test의 경우 구현한 시스템을 이루는 각 함수 단위로 함수 내에서 branch coverage를 높이는 방향으로 진행한

다. Software Interface Test의 경우 기존 소프트웨어 시스템과 구현한 시스템 사이 간 연동의 원활함을 검증하기 위한 방향으로 진행한다.

1.3. Definitions, Acronyms, and Abbreviation

1.3.1. Terms and Definitions

이하의 표는 본 문서에서 활용된 용어의 정의를 나타낸다.

[표 1] 용어 정리

용어	설명
사용자(User)	본 시스템을 사용하는 모든 사용자를 말한다.
관리자 (Administrator)	본 시스템 사용 중 프로젝트를 등록한 사람은 자신이 등록한 프로젝트에 한하여 관리자가 된다.
프로젝트 (project)	프로젝트란 본 시스템에서 사람들이 등록한 글을 말한다.
데이터베이스 (database)	데이터베이스란 여러 데이터가 저장되어 운영되는 시스템을 말한다.
스키마 (schema)	데이터베이스에 포함된 데이터를 설명하는 속성과 데이터의 제약 조건을 포함한다.
HTTP 프로토콜	웹 상에서 정보를 주고 받을 수 있는 프로토콜이다.
쿼리(query)	요청 질의를 의미한다. 일반적으로 HTTP 요청에서 쿼리는 URL에 표시되는 질의를 말한다.
파라미터 (parameter)	HTTP 요청에서 두 개의 값 사이에서 특정한 관계를 만들어 주는 변수이다.
바디(body)	HTTP 요청 시 URL에 표시되지 않는 입력 값이다.
헤더(header)	HTTP 요청에서 특수한 meta 정보를 포함하고 있는 부분이다.
쿠키(cookie)	HTTP 헤더에 포함되는 부분으로 웹사이트 접속 시 접속자의 개인 장치에 다운로드 되고 브라우저에 저장되는 작은 텍스트 파일이다. 사용자 장치 인식과 이용 내역, 인증을 위해 사용된다.

토큰(token)	웹 상에서 특정한 목적으로 사용 가능한 일종의 권한을 나타내는 증표이다.
웹 서버(web server)	웹 어플리케이션 서비스를 제공하는 서버를 말한다.
브라우저(browser)	웹 페이지를 다운로드하고 웹 페이지의 코드를 실행해서 사용자들이 웹 어플리케이션을 쉽게 사용할 수 있도록 도와주는 소프트웨어 프로그램이다.
소프트웨어(Software)	코드 뿐만이 아니라 소프트웨어 개발 전반에 관련된 문서 등을 모두 포함하는 개념이다.
Unit Test	각 기능마다 개별 기능 검증을 위한 테스트이다.
Interface Test	각 개별 기능을 통합했을 때, 기능 들 사이의 연결을 검증하기 위한 테스트이다.
브랜치(branch)	코드 중 조건으로 논리가 분기 되는 지점이다.
코드 커버리지(code coverage)	소프트웨어의 테스트를 논할 때 얼마나 테스트가 충분한지 나타내는 개념으로 얼마나 많은 코드가 실행이 되었는지 나타내는 개념이다.
브랜치 커버리지(branch coverage)	코드 커버리지 지표 중 하나로, 얼마나 많은 분기 문이 실행 되었는지 나타내는 개념이다.
라이브러리(library)	소프트웨어 프로그램 구현 간에 사용하는 특정 부분 기능을 수행하도록 제작된 코드를 말한다.

1.3.2. Acronyms and Abbreviations

[표 2] 약어 정의

용어	설명
API	Application Programming Interface
HTTP	HyperText Transfer Protocol
UI	User Interface
DB	DataBase
URL	Uniform Resource Locator

1.4. References

본 Test Plan 문서를 작성하며 참고한 자료는 아래와 같다.

- Test plan document format
- Ian Sommerville, Software Engineering
- IEEE Std 830-1998

1.5. Overview

Test Plan 문서는 해당 장인 Introduction을 포함해 총 5개의 장으로 구성된다. 2장, 3장, 4장은 전부 Test Case에 관한 장이다. 2장은 테스트할 기능과 도구를 설명한다. 2장에서는 Software Unit Testing 대상 기능의 의사 코드를 통해 기능의 작동 이해를 돕고, Software Interface Testing 대상 기능에 대해서 소개하며, 테스트 과정에서 사용할 도구들을 설명한다. 3장과 4장은 각각 Software Unit Test, Software Interface Test에 대해 기술한다. 3장과 4장은 각 테스트 케이스가 어떠한 시나리오와 경우를 고려해서 제작된 것에 대한 구체적인 내용으로 이루어져 있다. 3장과 4장에 각 테스트 케이스는 Input과 예상되는 Output, 두 가지로 이루어진다. 5장에서는 수정이 생길 때 마다 수정 이력에 대해 기록해둔다.

2. Approach

본 장에서는 전반적인 테스트 접근에 대해서 다룬다. 테스트는 Software Unit Testing과 Software Interface Testing 둘로 나누어서 진행한다. Software Unit Testing은 직접 구현한 기능이 올바르게 구현이 된 것인지 검증하기 위한 Testing 입니다. Software Interface Testing은 팀이 직접 구현하지 않은 소프트웨어 시스템과 연결이 원활하게 이루어지는지 확인하는 것에 초점을 맞춘다.

본 장에서는 각 Software Unit Testing과 Software Interface Testing에서 입력 값

과 출력 값, 시나리오 설정 등 테스트 진행 방향에 대한 이해를 돕기 위해 각 기능에 대한 의사 코드를 제공하고, 테스트 간 사용할 도구에 대해서 이야기한다. 각 테스트 케이스의 자세한 시나리오, 입력 값, 출력 값은 뒤에 3장과 4장에서 다룬다.

2.1. Test methods

2.1.1. Software Unit Test Methods

2.1.1.1. Objective

Software Unit Test는 구현한 시스템의 개별 기능이 의도한 방향으로 작동하는 지, 각 기능 단독으로 검증하는 단계이다. 하나의 기능에 대해서 여러 테스트 케이스를 통해 Code Coverage를 높이는 방향으로 진행할 계획이다. 이 부분과 관련한 테스트 케이스는 3장에서 다룬다. 아래는 이를 검증하기 위한 대표적인 방법이다.

- 정상적인 상황에서 의도한 대로 작동하는지 검사하기
- 입력 형식에 맞지 않는 데이터 넣어서 오류가 발생하는지 체크하기
- 같은 요청을 연속해서 여러 번 보내기 등이 존재한다.

2.1.1.2. Function List

본 부분에서는 각 기능들의 실제 구현 코드를 바탕으로 각 코드마다 브랜치를 표시하기 위한 간단한 의사 코드를 작성해 두었다. 아래에 있는 각 기능들의 의사 코드의 Coverage를 높이는 방향으로 3장에서 Test Case를 설정한다.

1) 회원가입

```
if (필수로 주어져야 할 정보가 없는 경우){  
    return "입력값이 부족합니다"  
}  
  
if (입력한 이메일이 성균관대학교 이메일이 아닌 경우){
```

```
        return "성균관대학교 이메일만 사용가능 합니다"
    }

    if (입력한 아이디가 사용 중인 경우){
        return "해당 아이디가 사용 중입니다"
    }

    if (입력한 이메일이 사용 중인 경우){
        return "해당 이메일은 사용 중입니다"
    }

    return "회원 가입 완료"
```

2) 로그인

```
    if (유저 아이디가 없음) or (유저 비밀번호가 없음){
        return "입력값이 부족합니다"
    }

    if (유저 아이디가 존재하지 않은 경우){
        return "아이디, 비밀번호를 다시 확인해주세요."
    }

    else{
        if (주어진 아이디에 패스워드가 틀림){
            return "아이디, 비밀번호를 다시 확인해주세요."
        }

        else{
            return "로그인 성공"
        }
    }
}
```

3) 사용자 리스트 조회

```
    if (페이지 사이즈가 없는 경우 or 페이지 순서가 없는 경우){
        return "입력값이 부족합니다"
    }

    if (검색어가 있는 경우){
```

```
    if (검색 결과가 없는 경우){  
        return "빈 리스트"  
    }  
    if(페이지 범위가 실제 가능 페이지 범위 밖일 때){  
        return "잘못된 접근입니다"  
    }  
    return "사용자 리스트"  
}  
else {  
    if (사용자가 아무도 없을 때){  
        return "빈 리스트"  
    }  
    if (페이지 범위가 실제 가능 페이지 범위 밖일 때){  
        return "잘못된 접근입니다"  
    }  
    return "사용자 리스트"  
}
```

4) 내 정보 조회

```
if (인증 토큰에 일치하는 사용자가 데이터베이스에 없을 때){  
    return "존재하지 않는 사용자입니다"  
}  
  
return "쿠키에 저장된 토큰 정보에 일치하는 사람의 정보"
```

5) 사용자 세부 정보 조회

```
if (조회하려는 사용자가 데이터베이스에 없을 때){  
    return "존재하지 않는 사용자입니다"  
}  
  
return "쿠키에 저장된 토큰 정보에 일치하는 사람의 정보"
```

6) 내 정보 수정

```
if (아무것도 수정한 것이 없을 때){  
    return "입력값이 부족합니다"  
}
```



```
if (인증 토큰에 일치하는 사용자가 데이터베이스에 없을 때){
    return "존재하지 않는 사용자입니다"
}
else {
    if (이미지를 변경한 경우){
        이미지 변경
    }
    if (이름을 변경한 경우){
        이름 변경
    }
    if (전공을 변경한 경우){
        전공 변경
    }
    if (이메일 변경한 경우){
        if (학교 이메일이 아닌 경우){
            return "성균관대학교 이메일만 사용 가능합니다"
        }
        이메일 변경
    }
    if (개인 페이지 링크를 변경한 경우){
        개인 페이지 링크 변경
    }
    if (개인 소개 메시지를 변경한 경우){
        개인 소개 메시지 변경
    }
    if (기술 스택을 변경한 경우){
        기술 스택 변경
    }

    return "정보 수정에 완료했습니다"
}
```

7) 회원 탈퇴

```
if (인증 토큰에 일치하는 사용자가 데이터베이스에 없을 때){
    return "존재하지 않는 사용자입니다"
}
else {
    if (사용자 이미지가 기본 이미지가 아닐 경우){
```

사용자 프로필 이미지 삭제

```

    }
    return "탈퇴 완료"
}

```

8) 프로젝트 리스트 조회

```

if (페이지 사이즈가 없는 경우 or 페이지 순서가 없는 경우){
    return "입력값이 부족합니다"
}

```

```

if(요청보낸 사용자가 인증이 된 경우){
    if (사용자가 작성한 것만 조회 옵션){
        사용자 작성한 것만 조회 옵션 처리
    }
    else if(사용자가 즐겨 찾기한 것 조회 옵션){
        사용자 즐겨 찾기한 것만 조회 옵션 처리
    }
    else if(사용자가 참여 신청한 것만 조회 옵션){
        사용자 참여 신청한 것만 조회 옵션 처리
    }
}

```

```

if (정렬 옵션이 주어진 경우){
    if (즐거 찾기 등록 횟수 내림차순 옵션){
        정렬 옵션 처리
    }
    else if(즐거 찾기 등록 횟수 오름차순 옵션){
        정렬 옵션 처리
    }
}

```

```

if (사용자가 작성한 것만 조회 옵션){
    if (검색 결과가 없는 경우){
        return "빈 리스트"
    }
    if (페이지 범위가 실제 가능 페이지 범위 밖일 때){
        return "잘못된 접근입니다"
    }
}

```

```
    }
    return "사용자 리스트"
}
else if (사용자가 즐겨 찾기한 것만 조회 옵션){
    if (사용자가 아무도 없을 때){
        return "빈 리스트"
    }
    if (페이지 범위가 실제 가능 페이지 범위 밖일 때){
        return "잘못된 접근입니다"
    }
    return "사용자 리스트"
}
else if (사용자가 참여 신청한 것만 조회 옵션){
    if (검색 결과가 없는 경우){
        return "빈 리스트"
    }
    if (페이지 범위가 실제 가능 페이지 범위 밖일 때){
        return "잘못된 접근입니다"
    }
    return "사용자 리스트"
}
else if (검색어가 있는 경우){
    if (검색 결과가 없는 경우){
        return "빈 리스트"
    }
    if (페이지 범위가 실제 가능 페이지 범위 밖일 때){
        return "잘못된 접근입니다"
    }
    return "사용자 리스트"
}
else {
    if (사용자가 아무도 없을 때){
        return "빈 리스트"
    }
    if (페이지 범위가 실제 가능 페이지 범위 밖일 때){
        return "잘못된 접근입니다"
    }
}
```

```
    return "사용자 리스트"
```

```
}
```

9) 프로젝트 등록

```
if (필수로 주어져야 할 정보가 없는 경우){
```

```
    return "입력값이 부족합니다"
```

```
}
```

```
if (등록 요청한 사용자가 데이터베이스에 등록되지 않음){
```

```
    return "사용자가 존재하지 않습니다"
```

```
}
```

```
else {
```

```
    return "프로젝트 등록 완료"
```

```
}
```

10) 프로젝트 랜덤 조회

```
if (랜덤 뽑기할 프로젝트 수가 음수 일때){
```

```
    뽑기 조회 개수 1로 설정
```

```
}
```

```
else if(랜덤 뽑기할 프로젝트 수가 3보다 클 때){
```

```
    뽑기 조회 개수 3으로 설정
```

```
}
```

```
if (전체 프로젝트의 개수가 뽑기 조회 개수보다 작을 때){
```

```
    뽑기 조회 개수를 전체 프로젝트 개수로 설정
```

```
}
```

```
return "뽑기 조회 개수만큼 프로젝트 랜덤 뽑기 리스트 생성"
```

11) 프로젝트 세부 정보 조회

```
if(요청받은 사용자가 인증이 된 경우){
```

```
    유저 인증 옵션 활성화
```

```
}
```

```
프로젝트 상세 정보 처리
```

```
if (사용자 인증 성공 여부){
    if (인증된 사용자의 해당 프로젝트 즐겨 찾기 추가 여부){
        옵션 처리
    }
    if (인증된 사용자의 해당 프로젝트 참여 신청 제출 여부){
        옵션 처리
    }
}

return "프로젝트 삭제 성공"
```

12) 프로젝트 삭제

```
if (인증 토큰에 일치하는 사용자가 데이터베이스에 없을 때){
    return "존재하지 않는 사용자입니다"
}

if (사용자가 프로젝트 삭제 권한이 없을 때){
    return "권한이 없습니다"
}
else {
    return "프로젝트 삭제 성공"
}
```

13) 프로젝트 정보 수정

```
if (인증 토큰에 일치하는 사용자가 데이터베이스에 없을 때){
    return "존재하지 않는 사용자입니다"
}

if (사용자가 프로젝트 수정 권한이 없을 때){
    return "권한이 없습니다"
}
else {
    if (프로젝트 이름을 변경할 경우){
        프로젝트 이름 변경
    }
    if (프로젝트 주제를 변경할 경우){
        프로젝트 주제 변경
    }
}
```

```
    }  
    if (프로젝트 모집 인원 변경할 경우){  
        프로젝트 모집 인원 변경  
    }  
    if (프로젝트 시작 시기를 변경할 경우){  
        프로젝트 시작 시기 변경  
    }  
    if (프로젝트 종료 시기를 변경할 경우){  
        프로젝트 종료 시기 변경  
    }  
    if (프로젝트 연락 수단 변경할 경우){  
        프로젝트 연락 수단 변경  
    }  
    if (프로젝트 기술 스택 변경할 경우){  
        프로젝트 기술 스택 변경  
    }  
    if (프로젝트 소개 메시지를 변경할 경우){  
        프로젝트 소개 메시지 변경  
    }  
  
    return "프로젝트 삭제 성공"  
}
```

14) 프로젝트 즐겨 찾기 리스트에 추가

```
if (사용자 인증에 실패했을 때){  
    return "사용자 인증에 실패했습니다"  
}  
  
if (이미 해당 프로젝트가 즐겨 찾기 리스트에 있는 경우){  
    return "이미 즐겨찾기 리스트에 있습니다"  
}  
  
return "즐거 찾기 리스트에 추가 완료했습니다"
```

15) 프로젝트 즐겨 찾기 리스트에서 삭제

```
if (사용자 인증에 실패했을 때){  
    return "사용자 인증에 실패했습니다"
```

```
}

if (이미 해당 프로젝트가 즐겨 찾기 리스트에 없는 경우){
    return "이미 즐겨찾기 리스트에 없습니다"
}

return "즐거 찾기 리스트에서 삭제 완료했습니다"
```

16) 프로젝트의 참여 신청자 리스트 조회

```
if(프로젝트 고유 번호가 주어지지 않은 경우){
    return "입력값이 부족합니다"
}

if (데이터 베이스에 없는 사용자가 요청을 보냈을 때){
    return "사용자가 존재하지 않습니다"
}

if (요청을 보낸 사용자가 프로젝트 관리자가 아닐 때){
    return "해당 프로젝트 관리 권한이 없습니다"
}

if (페이지 사이즈가 없는 경우 or 페이지 순서가 없는 경우){
    return "입력값이 부족합니다"
}

if (사용자가 아무도 없을 때){
    return "빈 리스트"
}

if (페이지 범위가 실제 가능 페이지 범위 밖일 때){
    return "잘못된 접근입니다"
}

return "사용자 리스트"
```

17) 프로젝트 참여 신청하기(프로젝트 참여자)

```
if(프로젝트 고유 번호가 주어지지 않은 경우){
    return "입력값이 부족합니다"
```

```
}

if (데이터 베이스에 없는 사용자가 요청을 보냈을 때){
    return "사용자가 존재하지 않습니다"
}

if (이미 신청 리스트에 등록된 프로젝트인 경우){
    return "이미 신청서를 제출한 프로젝트 입니다"
}
else {
    return "프로젝트 참여 신청서를 보냈습니다"
}
```

18) 프로젝트 참여 신청 수락/거절 여부 변경(프로젝트 등록자)

```
if(프로젝트 고유 번호가 없는 경우 or 참여 수락/거절 여부가 없는 경우){
    return "입력값이 부족합니다"
}

if (요청을 보낸 사용자가 데이터베이스에 존재하지 않을 때){
    return "사용자가 존재하지 않습니다"
}

if (요청을 보낸 사용자가 프로젝트 관리자가 아닐 때){
    return "해당 프로젝트 관리 권한이 없습니다"
}

if (신청 보류 옵션){
    return "신청서를 보류했습니다"
}
else if (신청 승인 옵션){
    return "신청서를 승인했습니다"
}
else if (신청 거절 옵션){
    return "신청서를 거절했습니다"
}
else {
    return "처리할 수 없는 옵션입니다"
}
```



```

    }

```

19) 프로젝트 참여 신청 취소하기(프로젝트 참여자)

```

    if(데이터 베이스에 없는 사용자가 요청을 보냈을 때){
        return "사용자가 존재하지 않습니다"
    }

    if (제출한 프로젝트 참여 신청서가 없을 때){
        return "제출한 프로젝트 참여 신청서가 없습니다"
    }
    else{
        return "프로젝트 참여 신청을 취소했습니다"
    }

```

2.1.2. Software Interface Test Methods

2.1.2.1. Objective

Software Interface Test는 Integration Test의 한 부분이다. 구현한 시스템 외에 사용한 브라우저나 데이터베이스 관리 시스템, 데이터 등과 구현한 시스템 사이 인터페이스의 오류를 찾고 작동이 원활하게 이루어지는 지를 검증하기 위한 테스트이다. 이 부분과 관련한 테스트 케이스는 4장에서 다룬다. 아래는 이를 검증하기 위한 대표적인 방법이다.

- 시스템 간 연결 테스트하기
- 예외 데이터로 요청 보내기
- 무리한 상황 설정하기 등이 존재한다.

2.1.2.2. Function List

1) MySQL (Sequelize 라이브러리 연동)

- Sequelize 라이브러리 제공 함수를 통한 MySQL 데이터베이스 Schema 생성
- Sequelize 라이브러리 제공 함수를 통한 MySQL 쿼리 전송 테스트

2) 브라우저 쿠키 설정

- 브라우저 쿠키 생성
- 브라우저 쿠키 삭제
- 브라우저 쿠키 사용

3) 업로드 이미지 파일 관리(Multer 라이브러리, AWS S3 연동)

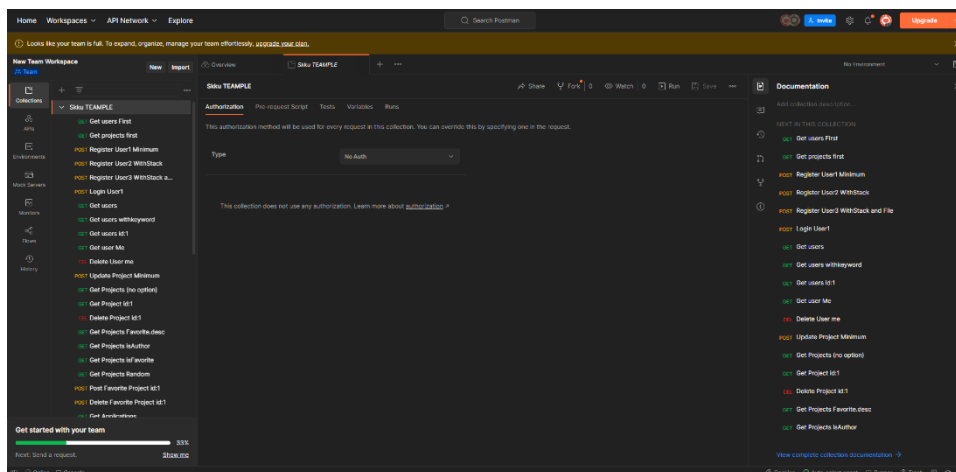
- 이미지 파일 저장
- 이미지 파일 불러오기
- 이미지 파일 삭제하기

2.1.3. Testing Tools

본 장에서는 테스트를 진행하는 과정에서 사용할 도구에 대해 정리했다. 아래의 도구들을 사용해서 테스트를 진행한다.

2.1.3.1. Postman 프로그램(Free Version)

[그림 1] Postman 프로그램

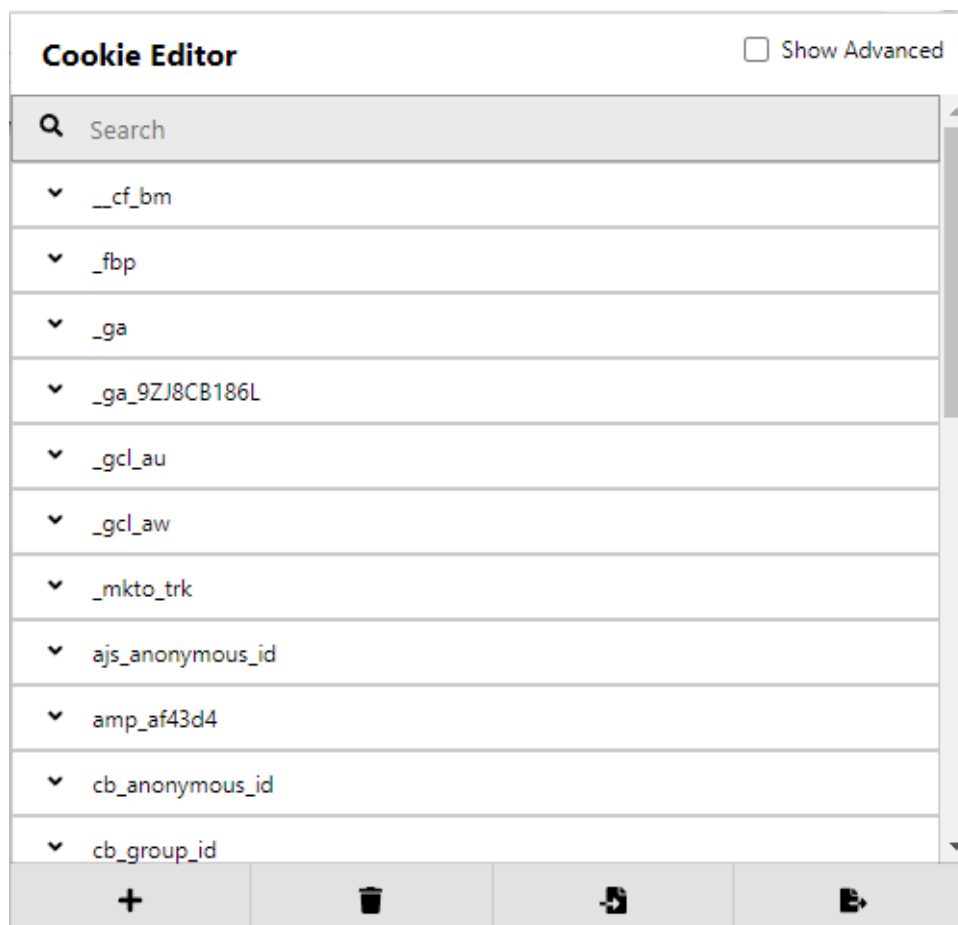


Postman은 웹 서버 API 테스트 어플리케이션이다. 위는 Postman에서 제공하는 UI 사진이다. Postman은 팀원들 간 공유된 작업 공간에서 테스트 진행이 가능하도록 서비스를 제공하고 있다. Postman을 이용하면 각각의 테스트 케이스를 저장해서 팀원들 간 현재 테스트 진행 상황을 공유할 수 있으며, 진행한 테스트 케

이스를 문서화하는 기능도 제공하고 있다. 주로 Software Unit Test에 사용될 예정이다.

2.1.3.2. Chrome 웹 브라우저

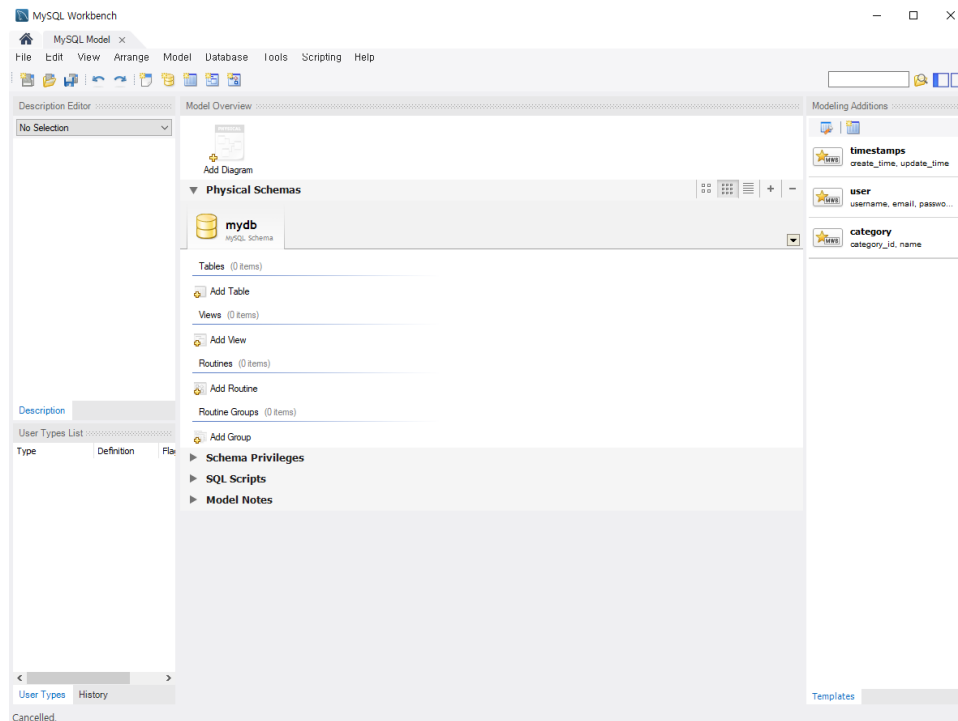
[그림 2] Chrome Cookie-Editor



구현한 UI를 통해서 테스트를 할 경우에는 Chrome 웹 브라우저를 사용한다. Chrome 브라우저에서 제공하는 개발자 도구를 통해 쿠키 생성, 삭제 등을 알아볼 수 있다. 개발자 도구 내 제공하는 기능을 통해서 전송 받은 웹 서버 응답도 확인 가능하다. 특히, 쿠키 생성, 삭제를 검증하기 위해서 Chrome 브라우저에서 설치할 수 있는 Cookie-Editor 개발자 도구를 설치해서 사용할 예정이다. 주로 Software Interface Test에 사용될 예정이다.

2.1.3.3. MySQL Workbench(8.0 Community Edition)

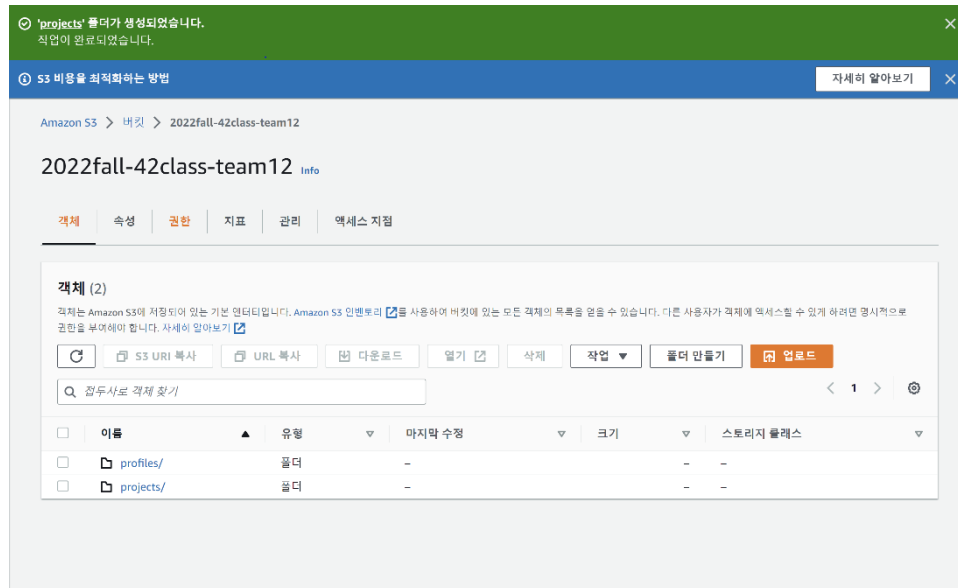
[그림 3] MySQL Workbench



MySQL Workbench 프로그램은 전체적인 데이터베이스 변화를 관찰하기 위해서 사용될 도구이다. MySQL에서 제공하는 Command Line 프로그램을 이용해서 데이터베이스를 관찰하는 것 보다는 더 직관적인 UI를 사용할 수 있다.

2.1.3.4. AWS S3 콘솔

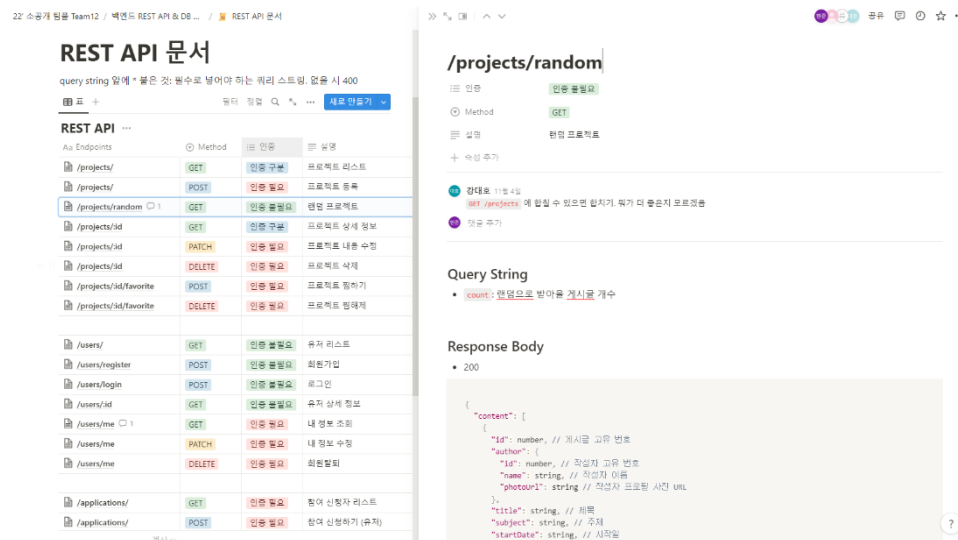
[그림 4] AWS S3



본 테스트를 진행하면서 AWS S3 저장소에 이미지가 제대로 업로드가 되어 저장되고, 삭제가 되는지 확인하기 위해서 AWS 서비스에서 제공하는 콘솔 창을 사용한다.

2.1.3.5. Notion(노션)

[그림 5] Notion



본 테스트를 진행하면서 전체적인 테스트 케이스를 공유하기 위해서 노션을 사용한다. 위 사진은 노션을 통해 작성한 REST API 문서이다. 노션에서 제공하는 체크리스트, 리스트 등의 기능은 테스트 진행 상황 공유를 위해 사용된다. 노션을 통해 팀원들 간 실시간 프로젝트 테스트 계획 공유 및 수정이 가능하다.

3. Software Unit Test

3.1. 회원가입

3.1.1. 시나리오

1. 회원 가입 페이지 이동 버튼을 누른다.
2. 회원 가입에 필요한 정보를 입력한다.
3. 회원 가입 완료 버튼을 누른다.
4. 회원 가입 결과를 받는다.

3.1.2. 일반적인 성공 케이스

[표 3] 회원가입 - 성공

(Success case)	
Input	body: id, password, email(skku.edu or g.skku.edu), department, name
Output message	회원가입되었습니다.

3.1.3. 필수 정보가 부족한 경우

[표 4] 회원가입 - 필수정보 부족

(Failure case)	
Input	body: id, password, email(skku.edu or g.skku.edu), department , name
Output message	회원가입되었습니다.

3.1.4. 성균관대학교 이메일이 아닌 경우

[표 5] 회원가입 - 이메일 오류

(Failure case)	
Input	body: id, password, email(skku.edu or g.skku.edu), department, name
Output message	성균관대학교 이메일만 사용 가능합니다!!

3.1.5. 입력한 아이디가 사용 중인 경우

[표 6] 회원가입 - 아이디 중복

(Failure case)	
Input	body: id(이미 사용중), password, email(skku.edu or g.skku.edu), department, name
Output message	중복된 아이디가 사용 중입니다.

3.1.6. 입력한 이메일이 이미 사용 중인 경우

[표 7] 회원가입 - 이메일 중복

(Failure case)	
Input	body: id, password, email(skku.edu or g.skku.edu / 이미 사용중), department, name
Output message	이미 사용 중인 이메일 입니다.

3.2. 로그인

3.2.1. 시나리오

1. 로그인 페이지로 이동

2. 아이디와 비밀번호를 입력
3. 로그인 버튼 클릭
4. 로그인 결과

3.2.2. 일반적인 성공 케이스

[표 8] 로그인 - 성공

(Success case)	
Input	body: id, password
Output message	로그인 성공!!

3.2.3. 아이디와 비밀번호 중 하나라도 입력되지 않은 경우

[표 9] 로그인 - 입력값 부족

(Failure case)	
Input	body: id, password
Output message	입력 값이 부족합니다.

3.2.4. 사용자 아이디가 등록되지 않은 경우

[표 10] 로그인 - 미등록 아이디

(Failure case)	
Input	body: id(등록되지 않음), password
Output message	아이디, 비밀번호를 다시 확인해주세요..

3.2.5. 비밀번호가 틀린 경우

[표 11] 로그인 - 비밀번호 오류

(Failure case)	
Input	body: id, password(올바르지 않음)
Output message	아이디, 비밀번호를 다시 확인해주세요.

3.3. 사용자 리스트 조회

3.3.1. 시나리오

1. 사용자는 다른 사용자 리스트 페이지로 이동한다.
2. 다른 사용자를 검색하기 위해 검색어를 입력한다.
3. 원하는 내용으로 검색된 사용자 리스트를 본다.

3.3.2. 일반적인 성공 케이스(검색어가 있을 때)

[표 12] 사용자 리스트 조회 - 성공

(Success case)	
Input	query: pageSize, pageNumber, keyword
Output message	사용자 리스트 리턴

3.3.3. 일반적인 성공 케이스(검색어가 없을 때)

[표 13] 사용자 리스트 조회(검색어 미포함) - 성공

(Success case)	
Input	query: pageSize, pageNumber, keyword

Output message	사용자 리스트 리턴
----------------	------------

3.3.4. 페이지 사이즈나 페이지 순서 번호가 없는 경우

[표 14] 사용자 리스트 조회 - 입력값 부족

(Failure case)	
Input	query: pageSize , pageNumber
Output message	입력 값이 부족합니다.

3.3.5. 결과가 빈 리스트인 경우(검색어가 있을 때)

[표 15] 사용자 리스트 조회 - 빈 리스트

(Failure case)	
Input	query: pageSize, pageNumber, keyword(해당 하는 요소가 없음)
Output message	빈 리스트 반환

3.3.6. 페이지 범위가 잘못된 경우(검색어가 있을 때)

[표 16] 사용자 리스트 조회 - 페이지 범위 오류

(Failure case)	
Input	query: pageSize, pageNumber, keyword(존재하지 않는 키워드)
Output message	잘못된 접근입니다

3.3.7. 결과가 빈 리스트인 경우(검색어가 없을 때)

[표 17] 사용자 리스트 조회(검색어 미포함) - 빈 리스트

(Failure case)	
Input	query: pageSize, pageNumber (이전 레코드가 없음)
Output message	빈 리스트 반환

3.3.8. 페이지 범위가 잘못된 경우(검색어가 없을 때)

[표 18] 사용자 리스트 조회(검색어 미포함) - 페이지 범위 오류

(Failure case)	
Input	query: pageSize(9999), pageNumber
Output message	잘못된 접근입니다

3.4. 내 정보 페이지 조회

3.4.1. 시나리오

1. 사용자가 로그인에 성공한다.
2. 로그인에 성공한 사용자가 내 정보 페이지로 이동한다.
3. 조회 결과를 기다린다.

3.4.2. 일반적인 성공 케이스

[표 19] 본인 정보 조회 - 성공

(Success case)	
Input	cookie:유효한 아이디로 생성된 토큰

Output message	유저 정보 리턴(userId, name department, introduction, skills, email, personalLink, photoUrl)
----------------	--

3.4.3. 사용자 인증에 실패한 케이스

[표 20] 본인 정보 조회 - 사용자 인증 실패

(Failure case)	
Input	cookie:유효하지 않은 아이디로 생성된 토큰
Output message	사용자가 존재하지 않습니다.

3.5. 사용자 세부 정보 조회

3.5.1. 시나리오

1. 사용자는 검색한 사용자 리스트에서 한 사람이 더 궁금하다.
2. 궁금한 사람에 대해서 더 알아보고 싶은 사용자는 알아보고 싶은 계정의 세부 정보 페이지로 이동 버튼을 클릭한다.
3. 알고 싶은 다른 사용자의 세부 정보 페이지로 이동한다.

3.5.2. 일반적인 성공 케이스

[표 21] 사용자 정보 조회 - 성공

(Success case)	
Input	path param:userid
Output message	해당 유저 정보 리턴(userId, name department, introduction, skills, email, personalLink, photoUrl)

3.5.3. 찾으려는 사용자가 데이터베이스에 없을 때

[표 22] 사용자 정보 조회 - 미등록 사용자

(Failure case)	
Input	path param:userid(유효하지 않음)
Output message	존재하지 않는 유저입니다.

3.6. 내 정보 수정

3.6.1. 시나리오

1. 사용자는 내 정보 페이지에 들어왔다.
2. 사용자는 계정을 생성한 후 바뀐 개인 정보를 바꾸고 싶다.
3. 사용자는 원하는 활동 이름, 프로필 이미지, 전공, 개인 페이지 링크, 개인 소개 메시지, 개인 기술 스택 등으로 다시 입력한다.
4. 자신이 원하는 내용으로 개인 프로필을 업데이트 한다.

3.6.2. 일반적인 성공 케이스

[표 23] 본인 정보 수정 - 성공

(Success case)	
Input	cookie:유효한 아이디로 생성된 토큰 body: file id password name department email(skku.edu or g.skku.edu) personalLink skills introduction
Output message	유저 정보 리턴

3.6.3. 아무것도 수정한 것이 없을 때

[표 24] 본인 정보 수정 - 수정 내용 없음

(Failure case)	
Input	cookie:유효한 아이디로 생성된 토큰 body: file id password name department email personalLink skills introduction
Output message	입력 값이 부족합니다.

3.6.4. 사용자 인증에 실패한 경우

[표 25] 본인 정보 수정 - 사용자 인증 실패

(Failure case)	
Input	cookie:유효하지 않은 아이디로 생성된 토큰 body: file id password name department email personalLink skills introduction
Output message	사용자가 존재하지 않습니다.

3.6.5. 변경하려는 이메일이 성균관대학교 이메일이 아닌 경우

[표 26] 본인 정보 수정 - 이메일 오류

(Failure case)	
Input	cookie:유효한 아이디로 생성된 토큰 body: file id password name department email(skku.edu or g.skku.edu) personalLink skills introduction
Output message	성균관대학교 이메일만 사용 가능합니다!!

3.7. 회원 탈퇴

3.7.1. 시나리오

1. 사용자는 이제 더 이상 본 시스템의 서비스를 사용하고 싶은 생각이 없다.
2. 사용자는 개인 페이지에 들어간다
3. 회원 탈퇴 버튼을 누른다.
4. 회원 탈퇴가 이루어진다.

3.7.2. 일반적인 성공 케이스(사용자 프로필 이미지가 기본 이미지가 아닐 경우)

[표 27] 회원 탈퇴 - 성공

(Success case)	
Input	cookie:유효한 아이디(프로필 이미지가 기본이미지가 아님)로 생성된 토큰
Output message	회원탈퇴가 완료되었습니다.

3.7.3. 일반적인 성공 케이스(사용자 프로필 이미지가 기본 이미지일 경우)

[표 28] 회원 탈퇴 (기본 이미지) - 성공

(Success case)	
Input	cookie:유효한 아이디(프로필 이미지 기본이미지)로 생성된 토큰
Output message	회원탈퇴가 완료되었습니다.

3.7.4. 사용자 인증에 실패한 경우

[표 29] 회원 탈퇴 - 사용자 인증 실패

(Failure case)	
Input	cookie:유효하지 않은 아이디로 생성된 토큰
Output message	사용자가 존재하지 않습니다.

3.8. 프로젝트 리스트 조회

3.8.1. 시나리오

- 1번 시나리오
 1. 사용자는 참여할 만한 프로젝트를 찾고 있다.
 2. 검색 창에 검색어를 통해 검색도 할 수 있다.
 3. 리스트를 조회한 사용자는 최신 생성 순으로 볼 수 있을 뿐만 아니라 즐겨 찾기 등록(내림차순, 오름차순 선택 가능) 수로 정렬해서 볼 수도 있다.
- 2번 시나리오
 1. 사용자는 내 정보 페이지에 진입한다.
 2. 내 정보 페이지에 내가 작성한 프로젝트, 내가 즐겨 찾기 등록한 프로젝트, 내가 참여 신청을 보낸 프로젝트 리스트가 있다.

3.8.2. 일반적인 성공 케이스(인증은 없고 검색어가 있을 경우, 정렬 옵션(Favorite.desc))

[표 30] 프로젝트 리스트 조회 - 성공

(Success case)	
Input	query: pageSize, pageNumber, keyword, sortBy(favorite.desc)

Output message	프로젝트 리스트 리턴(즐거찾기 수 내림차순)
----------------	--------------------------

3.8.3. 일반적인 성공 케이스(인증은 없고 검색어가 없을 경우, 정렬 옵션(Favorite.asc))

[표 31] 프로젝트 리스트 조회 (검색어 미포함) - 성공

(Success case)	
Input	query: pageSize, pageNumber, sortBy(favorite.asc)
Output message	프로젝트 리스트 리턴(즐거찾기 수 오름차순)

3.8.4. 일반적인 성공 케이스(사용자가 작성한 것만 조회하는 옵션)

[표 32] 사용자 프로젝트 리스트 조회 - 성공

(Success case)	
Input	cookie: 유효한 아이디로 생성된 토큰 query: pageSize, pageNumber, keyword, isAuthor
Output message	사용자가 작성한 프로젝트 리스트 리턴

3.8.5. 일반적인 성공 케이스(사용자가 즐겨 찾기 등록한 것만 조회하는 옵션)

[표 33] 사용자 즐겨찾기 리스트 조회 - 성공

(Success case)	
Input	cookie: 유효한 아이디로 생성된 토큰 query: pageSize, pageNumber, keyword, isFavorite
Output message	사용자가 즐겨찾기 등록한 프로젝트 리스트 리턴

3.8.6. 일반적인 성공 케이스(사용자가 참여 신청한 것만 조회하는 옵션)

[표 34] 사용자 신청 프로젝트 리스트 조회 - 성공

(Success case)	
Input	cookie: 유효한 아이디로 생성된 토큰 query: pageSize, pageNumber, keyword, isApplied
Output message	사용자가 신청한 프로젝트 리스트 리턴

3.8.7. 페이지 사이즈나 페이지 순서 번호가 없는 경우

[표 35] 프로젝트 리스트 조회 - 입력값 부족

(Failure case)	
Input	query: pageSize , pageNumber, keyword
Output message	입력 값이 부족합니다.

3.8.8. 조회 결과 전체 리스트가 비어 있는 경우

[표 36] 프로젝트 리스트 조회 - 빈 리스트

(Failure case)	
Input	query: pageSize, pageNumber, keyword(존재하지 않음)
Output message	빈 리스트 반환

3.8.9. 페이지 범위가 실제 가능 페이지 범위를 벗어난 경우

[표 37] 프로젝트 리스트 조회 - 잘못된 페이지 접근

(Failure case)	
Input	query: pageSize(9999), pageNumber, keyword
Output message	잘못된 접근입니다

3.9. 프로젝트 등록

3.9.1. 시나리오

1. 사용자는 기발한 아이디어가 떠올랐다.
2. 사용자는 같이 아이디어를 구현해 갈 팀원이 필요하다.
3. 사용자는 팀원을 찾기 위해 자신이 하고 싶은 프로젝트를 등록하고 싶다.
4. 로그인한 사용자는 프로젝트 생성 페이지로 이동한다.
5. 프로젝트 세부 정보 사항을 작성하고 글을 등록한다.

3.9.2. 일반적인 성공 케이스

[표 38] 프로젝트 등록 - 성공

(Success case)	
Input	cookie: 유효한 아이디로 생성된 토큰 body: title, subject, startDate, endDate
Output message	프로젝트를 성공적으로 등록했습니다.

3.9.3. 필수 입력 사항이 주어지지 않은 경우

[표 39] 프로젝트 등록 - 필수정보 부족

(Failure case)	
Input	cookie: 유효한 아이디로 생성된 토큰 body: title , subject, startDate, endDate
Output message	입력 값이 부족합니다.

3.9.4. 사용자 인증이 되지 않은 경우

[표 40] 프로젝트 등록 - 사용자 인증 실패

(Failure case)	
Input	cookie: 유효하지 않은 아이디로 생성된 토큰 body: title, subject, startDate, endDate
Output message	사용자가 존재하지 않습니다.

3.10. 프로젝트 랜덤 조회

3.10.1. 시나리오

1. 사용자는 무엇인가 하고 싶긴 한데 어떻게 프로젝트를 알아봐야 하는지 감이 안온다.
2. 사용자는 누군가 무작위로 자신에게 프로젝트를 제시해주면 좋겠다.
3. 사용자는 본 시스템에서 제공하는 랜덤 프로젝트 리스트를 통해 참여할 프로젝트를 찾는다.

3.10.2. 일반적인 성공 케이스

[표 41] 프로젝트 랜덤 조회 - 성공

(Success case)	
Input	query: count
Output message	count에 해당하는 랜덤 프로젝트 리스트 리턴

3.10.3. 성공 케이스(랜덤 뽑기할 프로젝트 수가 음수인 경우)

[표 42] 프로젝트 랜덤 조회 (0 개 미만) - 성공

(Success case)	
Input	query: count(-1)
Output message	1개의 랜덤 프로젝트 리스트 리턴

3.10.4. 성공 케이스(랜덤 뽑기할 프로젝트 수가 3 보다 큰 경우)

[표 43] 프로젝트 랜덤 조회 (4 개 이상) - 성공

(Success case)	
Input	query: count(4)
Output message	3개의 랜덤 프로젝트 리스트 리턴

3.10.5. 성공 케이스(전체 프로젝트 개수가 뽑기 조회 개수보다 작은 경우)

[표 44] 전체 프로젝트 개수 이상의 프로젝트 랜덤 조회 - 성공

(Success case)	
Input	query: count(3) / 전체 프로젝트 개수 2개
Output message	전체 프로젝트 리스트 리턴

3.11. 프로젝트 세부 정보 조회

3.11.1. 시나리오

1. 사용자는 드디어 관심 있는 프로젝트를 찾았다.
2. 관심 있는 프로젝트를 더 자세하게 알아보기 위해서 사용자는 프로젝트 세

부 조회 버튼을 누른다.

3. 프로젝트 세부 정보 페이지로 이동한다.

3.11.2. 일반적인 성공 케이스(사용자 인증 안 된 경우)

[표 45] 프로젝트 정보 조회 (사용자 인증 없음) - 성공

(Success case)	
Input	path param: projectid
Output message	프로젝트 정보 리턴(id, title, subject, startdate, enddate, contact, content, skills, photoUrl, lastEdited, capacity, approvalCount, favoriteCount, applicationCount, author)

3.11.3. 일반적인 성공 케이스(사용자 인증된 경우, 해당 프로젝트가 즐겨찾기로 등록된 경우)

[표 46] 즐겨찾는 프로젝트 정보 조회 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 즐겨찾기 등록)로 생성된 토큰 path param:projectid
Output message	프로젝트 정보 리턴(id, title, subject, startdate, enddate, contact, content, skills, photoUrl, lastEdited, capacity, approvalCount, favoriteCount, applicationCount, author, isFavorite(true), userApplication(false))

3.11.4. 일반적인 성공 케이스(사용자 인증된 경우, 해당 프로젝트에 신청서를 제출한 경우)

[표 47] 신청한 프로젝트 정보 조회 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 신청서 제출)로 생성된 토큰 path param:projectid
Output message	프로젝트 정보 리턴(id, title, subject, startdate, enddate, contact, content, skills, photoUrl, lastEdited, capacity, approvalCount, favoriteCount, applicationCount, isFavorite(false), userApplication(true))

3.12. 프로젝트 삭제

3.12.1. 시나리오

1. 프로젝트 글을 올렸던 관리자는 이제 더 이상 프로젝트 구인을 하고 싶은 생각이 없다.
2. 프로젝트 관리자는 글을 삭제하기 위해 로그인한다.
3. 프로젝트 관리자는 내 정보 페이지로 이동해 자신이 등록한 글 중 지우고 싶은 글의 세부 정보 페이지로 이동한다.
4. 프로젝트 관리자는 프로젝트 삭제 버튼을 누른다.

3.12.2. 일반적인 성공 케이스

[표 48] 프로젝트 삭제 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트 수정 권한

	있음)로 생성된 토큰 path param: projectid
Output message	프로젝트가 삭제되었습니다.

3.12.3. 사용자 인증에 실패한 경우

[표 49] 프로젝트 삭제 - 사용자 인증 실패

(Failure case)	
Input	cookie: 유효하지 않은 아이디로 생성된 토큰 path param: projectid
Output message	사용자가 존재하지 않습니다.

3.12.4. 사용자가 프로젝트 관리 권한이 없는 경우

[표 50] 프로젝트 삭제 - 권한 없음

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트 수정 권한이 없음)로 생성된 토큰 path param: projectid
Output message	프로젝트 수정 권한이 없습니다.

3.13. 프로젝트 정보 수정

3.13.1. 시나리오

1. 프로젝트 구성원을 모집하던 관리자는 프로젝트 관련 사항 변동 뉴스를 접하게 된다.

2. 변경된 내용을 프로젝트 정보에 반영하고 싶은 관리자는 로그인 후 프로젝트 수정 페이지로 들어간다.
3. 변경하고 싶은 내용을 작성한 관리자는 프로젝트 정보 수정 버튼을 누른다.

3.13.2. 일반적인 성공 케이스(전부 변경하는 경우)

[표 51] 프로젝트 수정 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트 수정 권한 있음)로 생성된 토큰 path param: projectid body: file title subject capacity startDate endDate contact skills content
Output message	정보를 수정했습니다.

3.13.3. 사용자 인증에 실패한 경우

[표 52] 프로젝트 수정 - 사용자 인증 실패

(Failure case)	
Input	cookie: 유효하지 않은 아이디로 생성된 토큰 path param: projectid body: file title subject capacity startDate endDate contact skills content
Output message	사용자가 존재하지 않습니다.

3.13.4. 사용자가 프로젝트 수정 권한이 없는 경우

[표 53] 프로젝트 수정 - 권한 없음

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트 수정 권한 없음)로 생성된 토큰 path param: projectid body: file title subject capacity startDate endDate contact skills content
Output message	프로젝트 수정 권한이 없습니다.

3.14. 프로젝트 즐겨 찾기 리스트에 추가

3.14.1. 시나리오

1. 로그인 한 후 관심 있는 프로젝트를 찾던 사용자가 흥미로운 프로젝트를 발견했다.
2. 사용자는 자신이 관심 있는 프로젝트를 표시하고 싶다.
3. 사용자는 해당 프로젝트를 즐겨 찾기 리스트에 추가하기 위해 즐겨 찾기 추가 버튼을 누른다.

3.14.2. 일반적인 성공 케이스

[표 54] 프로젝트 즐겨찾기 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 즐겨 찾기 신청하지 않음)로 생성된 토큰 path param: projectid

Output message	즐거찾기에 등록되었습니다.
----------------	----------------

3.14.3. 사용자 인증에 실패한 경우

[표 55] 프로젝트 즐거찾기 - 사용자 인증 실패

(Failure case)	
Input	cookie: 유효하지 않은 아이디로 생성된 토큰 path param: projectid
Output message	사용자가 존재하지 않습니다.

3.14.4. 중복된 즐거 찾기 추가 요청인 경우

[표 56] 프로젝트 즐거찾기 - 중복된 요청

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 즐거 찾기 신청하였음)로 생성된 토큰 path param: projectid
Output message	이미 즐거찾기된 프로젝트 입니다.

3.15. 프로젝트 즐거찾기 리스트에서 삭제

3.15.1. 시나리오

- 로그인 한 사용자는 즐거찾기 리스트에서 특정 프로젝트를 제거하고 싶다.
- 사용자는 즐거 찾기 리스트에서 제거하고 싶은 프로젝트를 즐거 찾기 리스트에서

삭제하기 위해 즐겨 찾기에서 삭제 버튼을 누른다.

3.15.2. 일반적인 성공 케이스

[표 57] 프로젝트 즐겨찾기 삭제 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 즐겨 찾기 신청함)로 생성된 토큰 path param: projectid
Output message	즐거찾기에서 삭제했습니다.

3.15.3. 사용자 인증에 실패한 경우

[표 58] 프로젝트 즐겨찾기 삭제 - 사용자 인증 실패

(Failure case)	
Input	cookie: 유효하지 않은 아이디로 생성된 토큰 path param: projectid
Output message	사용자가 존재하지 않습니다.

3.15.4. 중복된 즐겨 찾기 삭제 요청인 경우

[표 59] 프로젝트 즐겨찾기 삭제 - 중복된 요청

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 즐겨 찾기 신청하지 않음)로 생성된 토큰 path param: projectid
Output message	이미 프로젝트가 즐겨찾기 상태가 아닙니다.

3.16. 프로젝트 참여 신청자 리스트 조회

3.16.1. 시나리오

1. 관리자는 지금까지 프로젝트 참여 신청을 작성한 사람들을 보고 싶다.
2. 로그인 한 관리자는 작성한 글 리스트로 들어가서 참여 신청 현황을 보고 싶은 글 세부 정보 페이지로 들어간다.
3. 관리자는 신청자 명단 확인 버튼을 누른다.

3.16.2. 일반적인 성공 케이스

[표 60] 프로젝트 신청자 리스트 조회 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트의 권한이 있음)로 생성된 토큰 query: pageSize, pageNumber, projectid
Output message	해당 프로젝트의 참여 신청 리스트 리턴

3.16.3. 프로젝트 고유 번호가 주어지지 않은 경우

[표 61] 프로젝트 신청자 리스트 조회 - 입력값 부족 (프로젝트 번호)

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트의 권한이 있음)로 생성된 토큰 query: pageSize, pageNumber, projectid
Output message	입력 값이 부족합니다.

3.16.4. 사용자 인증에 실패한 경우

[표 62] 프로젝트 신청자 리스트 조회 - 사용자 인증 실패

(Failure case)	
Input	cookie: 유효하지 않은 아이디로 생성된 토큰 query: pageSize, pageNumber, projectid
Output message	사용자가 존재하지 않습니다.

3.16.5. 사용자가 관리자 권한이 없는 경우

[표 63] 프로젝트 신청자 리스트 조회 - 권한 없음

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트의 권한이 없음)로 생성된 토큰 query: pageSize, pageNumber, projectid
Output message	해당 프로젝트 관리 권한이 없습니다.

3.16.6. 페이지 사이즈나 페이지 순서 번호가 주어지지 않은 경우

[표 64] 프로젝트 신청자 리스트 조회 - 입력값 부족 (페이지)

(Failure case)	
Input	query: pageSize , pageNumber, projectid
Output message	입력 값이 부족합니다.

3.16.7. 프로젝트 참여 신청자가 아무도 없을 때

[표 65] 프로젝트 신청자 리스트 조회 - 신청자 없음

(Failure case)	
----------------	--

Input	cookie: 유효한 아이디(해당 프로젝트의 권한이 있음)로 생성된 토큰 query: pageSize, pageNumber, projectid(신청이 없는 프로젝트)
Output message	빈 리스트 반환

3.17. 프로젝트 참여 수락/거절 여부 변경

3.17.1. 시나리오

- 1번 시나리오

1. 관리자는 제출된 신청자의 신청을 허락해서 같이 일하려고 한다.
2. 관리자는 로그인 후 자신이 등록한 프로젝트 신청자 리스트 페이지로 이동한다.
3. 관리자는 신청자 리스트 페이지에 있는 신청자 중 한 명의 신청을 수락하기 위해서 승인 버튼을 누른다.

- 2번 시나리오

1. 관리자는 승인했던 신청을 다시 보류 상태로 돌리고 싶다.
2. 관리자는 로그인 후 자신이 등록한 프로젝트 신청자 리스트 페이지로 이동한다.
3. 관리자는 신청자 리스트 페이지에 있는 신청자 중 한 명의 신청을 보류하기 위해서 보류 버튼을 누른다.

- 3번 시나리오

1. 관리자는 신청한 사람의 요청을 거절하고 싶다.
2. 관리자는 로그인 후 자신이 등록한 프로젝트 신청자 리스트 페이지로 이동한다.

3. 관리자는 신청자 리스트 페이지에 있는 신청자 중 한 명의 신청을 거절하기 위해 거절하기 버튼을 누른다.

3.17.2. 일반적인 성공 케이스(신청 보류 옵션)

[표 66] 프로젝트 신청서 보류 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트의 권한이 있음)로 생성된 토큰 path param: applicationid body : projectId, status(waiting)
Output message	신청서를 보류했습니다.

3.17.3. 일반적인 성공 케이스(신청 승인 옵션)

[표 67] 프로젝트 신청서 승인 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트의 권한이 있음)로 생성된 토큰 path param: applicationid body : projectId, status(approved)
Output message	신청서를 승인했습니다.

3.17.4. 일반적인 성공 케이스(신청 거절 옵션)

[표 68] 프로젝트 신청서 거절 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트의 권한이

	있음)로 생성된 토큰 path param: applicationid body : projectId, status(rejected)
Output message	신청서를 거절했습니다.

3.17.5. 필수 입력 사항이 부족한 경우

[표 69] 프로젝트 신청서 거절 - 입력값 부족

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트의 권한이 있음)로 생성된 토큰 path param: applicationid body : projectId, status(rejected)
Output message	입력 값이 부족합니다.

3.17.6. 사용자 인증에 실패한 경우

[표 70] 프로젝트 신청서 거절 - 사용자 인증 실패

(Failure case)	
Input	cookie: 유효하지 않은 아이디로 생성된 토큰 path param: applicationid body : projectId, status(rejected)
Output message	사용자가 존재하지 않습니다.

3.17.7. 사용자에게 프로젝트 관리 권한이 없는 경우

[표 71] 프로젝트 신청서 거절 - 권한 없음

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트의 권한이

	없음)로 생성된 토큰 path param: applicationid body : projectId, status(rejected)
Output message	해당 프로젝트 관리자 권한이 없습니다.

3.17.8. 처리할 수 없는 옵션이 주어진 경우

[표 72] 프로젝트 신청서 - 승인, 보류, 거절 외의 접근

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트의 권한이 있음)로 생성된 토큰 path param: applicationid body : projectId, status(reject +wating+ rejected)
Output message	잘못된 접근입니다.

3.18. 프로젝트 참여 신청하기

3.18.1. 시나리오

1. 사용자는 자신이 특정 프로젝트에 참여하고 싶다.
2. 사용자는 프로젝트 참여 신청을 제출하고 싶다.
3. 사용자는 로그인 한 상태에서 프로젝트 세부 정보 확인 페이지로 이동한다.
4. 사용자는 프로젝트 참여 신청 버튼을 누른다.

3.18.2. 일반적인 성공 케이스

[표 73] 프로젝트 참여 신청 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 참여 신청하지

	않음)로 생성된 토큰 path param: projectid
Output message	프로젝트 신청서를 보냈습니다.

3.18.3. 프로젝트 고유 번호가 주어지지 않은 경우

[표 74] 프로젝트 참여 신청 - 입력값 부족

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 참여 신청하지 않음)로 생성된 토큰 body: projectId
Output message	입력 값이 부족합니다.

3.18.4. 사용자 인증에 실패한 경우

[표 75] 프로젝트 참여 신청 - 사용자 인증 실패

(Failure case)	
Input	cookie: 유효하지 않은 아이디로 생성된 토큰 body: projectId
Output message	사용자가 존재하지 않습니다.

3.18.5. 이미 프로젝트 참여 신청서를 등록한 경우

[표 76] 프로젝트 참여 신청 - 중복된 신청

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 이미 참여 신청함)로 생성된 토큰 body: projectId
Output message	이미 신청서를 제출한 프로젝트 입니다.

3.19. 프로젝트 참여 신청 취소하기

3.19.1. 시나리오

1. 사용자는 알고 보니 일정 상의 문제로 신청서를 냈던 프로젝트에 참여할 수 없을 거 같다.
2. 사용자는 프로젝트 참여 신청을 취소하고 싶다.
3. 사용자는 로그인 한 상태에서 프로젝트 세부 정보 확인 페이지로 이동한다.
4. 사용자는 프로젝트 참여 신청 취소 버튼을 누른다.

3.19.2. 일반적인 성공 케이스

[표 77] 프로젝트 참여 신청 취소 - 성공

(Success case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 참여 신청함)로 생성된 토큰 path param: applicationid
Output message	참여 신청을 취소했습니다.

3.19.3. 사용자 인증에 실패한 경우

[표 78] 프로젝트 참여 신청 취소 - 사용자 인증 실패

(Failure case)	
Input	cookie: 유효하지 않은 아이디로 생성된 토큰 path param: applicationid

Output message	사용자가 존재하지 않습니다.
----------------	-----------------

3.19.4. 프로젝트에 제출한 참여 신청서가 없는 경우

[표 79] 프로젝트 참여 신청 취소 - 신청서 없음

(Failure case)	
Input	cookie: 유효한 아이디(해당 프로젝트에 참여 신청하지 않음)로 생성된 토큰 path param: applicationid(유효하지 않은 식별번호)
Output message	존재하지 않은 신청서 입니다.

4. Software Interface Test

본 장은 Software Interface Test에 대해 다룬다. 본 장에서 다루는 테스트는 직접 구현한 시스템과 연결되어 작동하는 다른 시스템과의 연결과 의도한 작동이 이루어지는 지 검증하는 목적이 있다. 구현한 시스템은 데이터 저장을 위해 Node.js의 Sequelize 라이브러리를 사용해서 MySQL에 테이블을 생성하고 쿼리를 보낸다. 이를 확인하기 위해서는 MySQL 데이터베이스가 의도한 대로 잘 바뀌는 지 검증해야 한다. 구현한 시스템은 사용자 인증을 위해 로그인 시 발행한 토큰을 쿠키에 저장한다. 이를 위해 브라우저 내 쿠키가 제대로 저장되고 삭제되는 지를 확인해야 한다. 마지막으로 구현한 시스템은 사용자들이 업로드한 이미지를 저장하고 해당 이미지를 사용할 수 있어야 한다. 이 기능이 원활하게 작동하는 지도 검증한다.

4.1. MySQL(데이터베이스 Sequelize 라이브러리 연동 검증)

본 기능을 검증하기 위해서 MySQL Workbench와 구현한 시스템의 Sequelize 라이브러리를 사용합니다. 본 기능을 검증하기 위해서 데이터베이스 스키마 생성, MySQL 데이터베이스 정보 조회, 정보 삽입, 정보 수정과 관련된 테스트를 검증합니다.

4.1.1. MySQL 데이터베이스 테이블 스키마 생성

본 테스트는 데이터베이스와 구현 어플리케이션 간 연동을 검증할 목적의 테스트입니다.

데이터 베이스 테이블 스키마 생성과 관련된 테스트 케이스는 두 가지가 있습니다. 첫째, 정상적으로 테이블 스키마가 생성되도록 미리 설정한 이름의 데이터베이스가 생성되어 있는 경우와 그렇지 않은 경우, 그리고 생성은 되어있지만 이전에 사용하던 데이터베이스와 새로운 데이터베이스의 테이블 스키마가 다른 경우로 나누어 진행합니다.

4.1.1.1. 사용할 빈 데이터베이스가 생성된 경우

[표 80] DB 테이블 스키마 생성 - 성공

Input	데이터베이스 테이블 스키마 정보
Output message	데이터베이스 테이블 생성 성공 메시지

4.1.1.2. 사용할 데이터베이스가 생성되지 않은 경우

[표 81] DB 테이블 스키마 생성 - 실패

Input	데이터베이스 테이블 스키마 정보
Output message	데이터베이스 테이블 생성 실패 메시지

4.1.1.3. 이전에 사용한 데이터베이스가 생성되어 있는 경우

[표 82] DB 테이블 스키마 생성 (기존 사용 DB 존재) - 성공

Input	데이터베이스 테이블 스키마 정보
Output message	데이터베이스 테이블 생성 성공 메시지, 기존 테이블 스키마 삭제되고 새로운 테이블 스키마 생성

4.1.2. MySQL 데이터베이스 정보 조회

본 테스트는 데이터베이스와 구현 어플리케이션 간 연동을 검증할 목적의 테스트입니다.

데이터 베이스 테이블 정보 조회와 관련해서는 MySQL과 Sequelize가 잘 연동되어 데이터가 원활하게 조회가 되는 지를 테스트합니다. 조건에 맞는 데이터가 없는 경우와 조건을 충족하는 정보가 있는 경우로 나누어 정보 조회 테스트를 진행합니다.

4.1.2.1. 조건을 충족하는 정보가 있을 경우

[표 83] DB 정보 조회 - 성공

Input	조건을 충족하는 데이터베이스 내 데이터에 대한 데이터
-------	-------------------------------

	요청
Output message	요청에 맞는 데이터 반환, 5ms 안에 완료

4.1.2.2. 조건을 충족하는 정보가 없을 경우

[표 84] DB 정보 조회 - 정보 없음

Input	데이터베이스에 있는 모든 정보가 충족하지 못하는 데이터 요청
Output message	빈 메시지 반환, 5ms 안에 완료

4.1.3. MySQL 데이터베이스 정보 삽입

본 테스트는 데이터베이스와 구현 어플리케이션 간 연동을 검증할 목적의 테스트입니다.

데이터 베이스 테이블 정보 삽입과 관련해서는 MySQL과 Sequelize가 잘 연동되어 데이터가 원활하게 삽입이 되는 지를 테스트합니다. 스키마 상에 등록된 데이터 크기를 넘어가는 입력을 받을 때와 제한된 크기를 넘지 않는 정상적인 상황으로 나누어 테스트 케이스를 진행합니다.

4.1.3.1. 삽입된 정보가 제한 크기를 넘는 경우

[표 85] DB 정보 삽입 - 제한 크기 초과

Input	삽입하는 데이터의 속성 중 데이터베이스 속성의 제한 크기보다 큰 값을 삽입하는 요청
-------	---

Output message	데이터 삽입 요청에 대한 에러 메시지 반환, 3ms 안에 완료
----------------	---------------------------------------

4.1.3.2. 정상적인 경우

[표 86] DB 정보 삽입 - 성공

Input	유효한 크기와 범위의 데이터 삽입 요청
Output message	데이터 삽입 요청 성공 메시지, 3ms 안에 완료

4.1.4. MySQL 데이터베이스 정보 수정

본 테스트는 데이터베이스와 구현 어플리케이션 간 연동을 검증할 목적의 테스트입니다.

데이터 베이스 테이블 정보 수정과 관련해서는 MySQL과 Sequelize가 잘 연동되어 데이터가 원활하게 수정이 되는 지를 테스트합니다. 스키마 상에 등록된 데이터 크기를 넘어가는 입력을 받을 때와 제한된 크기를 넘지 않는 정상적인 상황으로 나누어 테스트 케이스를 진행합니다.

4.1.4.1. 수정될 정보가 제한 크기를 넘는 경우

[표 87] DB 정보 수정 - 제한 크기 초과

Input	새롭게 수정되어 적용되는 데이터의 속성 중 데이터베이스 속성의 제한 크기보다 큰 값이 있는 요청
-------	--

Output message	데이터 수정 요청에 대한 에러 메시지와 수정 실패, 3ms 안에 완료
----------------	---

4.1.4.2. 정상적인 경우

[표 88] DB 정보 수정 - 성공

Input	유효한 크기와 범위의 데이터로 수정 요청
Output message	데이터 수정 요청 성공 메시지, 3ms 안에 완료

4.2. 브라우저 쿠키 설정

본 기능을 검증하기 위해서 구현한 시스템에서 사용한 쿠키 관리 라이브러리가 실제로 브라우저의 쿠키에 잘 적용되는 지를 테스트합니다. 본 기능을 검증하기 위해서 구글 Chrome 브라우저에서 제공하는 Cookie-Editor 개발자 도구 프로그램을 사용합니다.

4.2.1. 쿠키 생성

본 테스트의 목적은 로그인 시 발행되는 토큰을 쿠키를 통해 저장하는 구현 방식을 검증하기 위함이다. 구현한 시스템에서 사용하는 쿠키 관리 라이브러리를 통해 브라우저가 쿠키를 제대로 생성하는 지 검증한다.

4.2.1.1. 쿠키 키를 적지 않은 경우

[표 89] 쿠키 생성 - 쿠키 키 없음

Input	쿠키 키 없이 쿠키 생성 요청
Output message	쿠키가 생성되지 않고 에러 메시지 반환

4.2.1.2. 정상적인 경우

[표 90] 쿠키 생성 - 성공

Input	키와 값으로 정상적인 쿠키 생성 요청
Output message	주어진 키 이름의 쿠키가 생성되고 쿠키의 값은 주어진 값으로 설정, 3ms 안에 완료

4.2.2. 쿠키 삭제하기

본 테스트 목적은 다음과 같다. 로그아웃 구현을 위해 새로운 웹 서버 기능을 따로 구현하지 않고 쿠키에 저장된 쿠키를 삭제하는 방식을 사용했기 때문에 이 기능이 제대로 작동하는 지 검증할 필요가 있다. 따라서 구현한 시스템에서 사용하는 쿠키 관리 라이브러리를 통해 브라우저가 쿠키를 제대로 삭제하는 지 검증한다.

4.2.2.1. 쿠키 키를 적지 않은 경우

[표 91] 쿠키 삭제 - 쿠키 키 없음

Input	쿠키 키 없이 쿠키 삭제 요청
Output message	에러 메시지와 쿠키 삭제 실패

4.2.2.2. 정상적인 경우

[표 92] 쿠키 삭제 - 성공

Input	키를 명시한 정상적인 쿠키 삭제 요청
Output message	해당 키 이름을 가진 쿠키 삭제

4.2.3. 쿠키 사용하기

본 테스트의 목적은 쿠키가 유효 시간이 지난 경우에 사용이 제한되는지 검증함에 있다. 로그인 후 쿠키를 통해 사용자 인증이 이루어지기 때문에 장기간 쿠키 유효 시간이 지나지 않고 있다면 사용자 계정 보안 상 문제가 될 수 있다.

4.2.3.1. 쿠키 유효 기간이 지난 경우

[표 93] 쿠키 사용 - 쿠키 유효 기간 초과

Input	유효 기간 지난 쿠키 사용에 대한 요청
Output message	쿠키 유효 기간 경과 에러 메시지

4.2.3.2. 정상적인 경우

[표 94] 쿠키 사용 - 성공

Input	유효 기간이 경과 되지 않은 쿠키 사용에 대한 요청
-------	------------------------------

Output message	데이터 수정 요청 성쿠키 키에 대응되는 쿠키 값 반환, 3ms 안에 완료 공 메시지, 3ms 안에 완료
----------------	--

4.3. 업로드 이미지 관리

본 기능을 검증하기 위해서 구현한 시스템에서 Node.js 라이브러리 중 multer 라이브러리를 사용해서 저장한 이미지 파일이 제대로 저장이 되고, 링크로 사용이 잘 이루어지는 지 검증합니다. 더불어 이미지 저장이 AWS S3와 연동이 잘 이루어지는지 확인합니다. 이 부분을 검증하기 위해서 Postman 프로그램을 이용해서 이미지 전송을 확인하고, 실제 프로그램 실행을 통해서 프론트엔드에서 이미지 사용에 문제가 없는지 확인합니다.

4.3.1. 이미지 파일 생성

본 테스트의 목적은 시스템에서 업로드 요청한 파일을 저장해서 사용자가 프로필 이미지나 프로젝트 이미지를 변경할 수 있도록 하기 위한 이미지 저장 필요성에 있다. 이미지 파일이 multer 라이브러리를 통해서 정상적으로 AWS S3 저장소에 저장되는지 확인한다.

4.3.1.1. 이미지 크기 제한보다 큰 경우

[표 95] 이미지 파일 생성 - 크기 제한 초과

Input	이미지 제한 크기보다 큰 이미지 업로드 요청
Output message	이미지 저장 안되고 반환 값 없음

4.3.1.2. 이미지 확장자가 지정된 확장자가 아닌 경우

[표 96] 이미지 파일 생성 - 확장자 오류

Input	이미지 확장자가 지정된 확장자가 아닌 업로드 요청
Output message	이미지 저장 안되고 반환 값 없음

4.3.1.3. 정상적인 경우

[표 97] 이미지 파일 생성 - 성공

Input	정상적인 이미지 업로드 요청
Output message	이미지가 지정된 경로에 저장되고 저장된 이미지 파일 정보를 반환, 2초 안에 진행되어야 함

4.3.2. 이미지 파일 불러오기

본 테스트의 목적은 사용자가 업로드한 이미지가 사용자가 다른 페이지를 방문했을 때, 해당 업로드 요청에 따라 이미지가 바뀌기 위함에 있다. 따라서 저장된 이미지 파일이 지정된 경로에서 불러올 수 있는지 확인한다. AWS S3 콘솔을 통해서 확인한다.

4.3.2.1. 해당 경로에 이미지가 없는 경우

[표 98] 이미지 파일 요청 - 이미지 없음

Input	해당 경로에 이미지 파일이 없는 이미지 파일 불러오기 요청
Output message	오류 메시지 반환

4.3.2.2. 정상적인 경우

[표 99] 이미지 파일 요청 - 성공

Input	해당 경로에 이미지 파일이 있는 이미지 파일 불러오기 요청
Output message	이미지 파일 경로와 이름을 반환, 3ms 안에 진행되어야 함

4.3.3. 이미지 파일 삭제

본 테스트의 목적은 사용자가 프로젝트를 삭제하거나 프로필 사진을 바꾸거나 회원 탈퇴를 할 경우, 이미지 파일은 저장 공간 확보와 프라이버시 보호를 위해 사용되지 않는 사진을 삭제해야 하는 기능에 있다. AWS S3 콘솔을 통해서 확인한다.

4.3.3.1. 해당 경로에 이미지가 없는 경우

[표 100] 이미지 파일 삭제 - 이미지 없음

Input	해당 경로에 이미지 파일이 없는 이미지 삭제 요청
Output message	이미지 삭제 실패 에러 메시지 반환

4.3.3.2. 정상적인 경우

[표 101] 이미지 파일 삭제 - 성공

Input	정상적인 이미지 삭제 요청
Output message	해당 이미지 삭제

5. Supporting Information

5.1. Test Environment

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)

- Web browser test environment

[표 102] 웹브라우저 테스트 환경

Support Level 1 (browsers)	- Windows 10: Edge, Chrome(latest), Firefox(latest), Safari(latest) - Mac OS X: Chrome(latest), Firefox(latest), Safari(latest) - Linus Ubuntu: Chrome(latest), Firefox(latest)
Support Level 2	Windows10: IE 11, Chrome(latest), Firefox(latest), Safari(latest)
Support Level 3	Anything else

- Database test environment

[표 103] DB 테스트 환경

Support Level 1 (database)	- MySQL: 8.0.31 Community Edition or higher version
Support Level 2	- MySQL: 8.X Community Edition
Support Level 3	- MySQL: Any version

5.2. Document History

[표 104] 문서 이력

Date	Version	Description	Writer
2022.11.2.	1.0	팀원 회의 후 최초 버전 발행	Ilwon Jung, et al.
2022.11.11	1.1	2, 4 장 작성, 1 장 수정, 5.1 수정, 3 Software Unit Test 틀 작성	Hyunjun Ahn
2022.11.11	1.2	전체 문서 작성 완료	Ilwon Jung, Hyunjun Ahn
2022.11.12	2.0	최종 검토 완료	Ilwon Jung, et al.