**고객 요구사항 정의서**

팀장 : 문성욱

팀원 : 권성욱

윤성민

**변경 이력**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 버전 | 변경 내역 | 작성 | | 검토 | | 승인 | | |
| 일자 | 담당자 | 일자 | 담당자 | | 일자 | 담당자 |
| 1.0.0 | 요구사항 정의 초본 | 8.6 | 권성욱 |  |  | |  |  |
|  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  | |  |  |

목차

1. . 문서 개요 ………………………………………………………… 4
2. . 고객 요구사항 목록 ……………………………………………… 5
3. . 고객 요구사항 정의 ………………………………………………… 5
   1. 프로젝트 관리 ………………………………………………… 5
   2. 멤버 관리 ………………………………………………… 6
   3. 이슈 관리 ………………………………………………… 6
   4. 룰 관리 ………………………………………………… 7
   5. 코드 컨벤션 ………………………………………………… 7
   6. 코드 리뷰 ………………………………………………… 8
   7. 코드 템플 ………………………………………………… 8

# 문서 개요

본 문서는 고객 및 이해관계자들로부터 도출된 요구사항을 정의한다. 즉 고객 및 이해관계자들이 시스템으로부터 기대하는 특성인 요구사항을 추출하여 기록한다. 식별된 각 고객 요구사항에 대해서는 식별자, 요구사항 명, 우선 순위 그리고 요구사항의 유형을 기록한다.

본 문서의 주요 구성 항목은 다음과 같다.

* 고객 요구사항 목록: 식별된 고객 요구사항을 목록 형태로 나열한다. 각 요구사항에 대해서 식별자, 요구사항 명, 우선순위, 요구사항의 유형을 정의한다.
* 고객 요구사항 정의: 식별된 각 고객 요구사항에 대한 상세한 정의를 한다. 각 고객 요구사항에 대하여 요청자, 일자, 요약, 상세 내용 전제 조건을 정의한다.

# 고객 요구사항 목록

|  |  |  |  |
| --- | --- | --- | --- |
| 고객 요구사항 ID | 요구사항 명 | 우선  순위 | 유형 |
| CR-001 | 프로젝트 관리 | 1 | 기능 |
| CR-002 | 멤버 관리 | 1 | 기능 |
| CR-003 | 이슈 관리 | 2 | 기능 |
| CR-004 | 룰 관리 | 1 | 기능 |
| CR-005 | 코드 컨벤션 | 2 | 기능 |
| CR-006 | 코드 리뷰 | 2 | 기능 |
| CR-007 | 코드 템플릿 | 3 | 기능 |

# 고객 요구사항 정의

## CR-001 – 프로젝트 관리

|  |  |
| --- | --- |
| 항목 | 내용 |
| 상세 설명 | 개별 이용자 및 팀 단위 이용자(팀원, 팀장)들은 새로운 프로젝트를 생성하고 접근할 수 있으며, 새로이 생성되거나 기존의 프로젝트에 대하여 수정 및 삭제가 가능하다. |
| 선행조건 | * 로그인 : 이용자에 대한 신분 인증을 이루어진다. * 소스 파일 관리 : 프로젝트 하위 생성된 폴더 및 파일에 대하여 CRUD 작업이 가능하다. |
| 후행조건 | * 프로젝트 접근 : 프로젝트 및 하위 디렉토리로 읽기 작업이 가능하다. * 프로젝트 생성 : 이용자들은 사용하기에 앞서 프로젝트를 먼저 생성한다. * 프로젝트 삭제 : 프로젝트 단위로 삭제 작업 가능하다. * 프로젝트 수정 : 프로젝트 이름 및 하위 디렉토리의 수정 작업이 가능하다. |
| Test case | 개별 이용자 A는 개인의 프로젝트를 생성하여 읽기/쓰기/수정/삭제 작업을 하였다.  팀별 이용자 B는 팀장으로, 프로젝트에 대한 삭제 작업을 시행하였다.  팀별 이용자 C는 B의 팀원이기 때문에 프로젝트에 대한 삭제 작업을 실패하였다. |

## CR-002 – 멤버관리

|  |  |
| --- | --- |
| 항목 | 내용 |
| 상세 설명 | 개별 이용자 및 팀 단위 이용자(팀장)은 생성된 프로젝트에 대하여 개별 이용자들을 추가하여 프로젝트 관리 기능을 수행할 수 있다. |
| 선행조건 | - 로그인 : 이용자에 대한 신분 인증을 이루어진다.  - 프로젝트 관리 : 프로젝트에 대하여 CRUD 작업을 가능하다.  - 이메일 관리 : 개별 이용자는 이메일 시스템을 사용할 수 있다. |
| 후행조건 | - 멤버 추가 : 지정한 멤버에게 이메일을 통한 초대 발송을 한다.  - 멤버 삭제 : 기존의 멤버를 프로젝트에서 제외한다.  - 멤버 조회 : 프로젝트에 참여하고 있는 멤버에 대하여 조회 기능을 수행한다.  - 멤버 히스토리 : 과거 프로젝트를 진행한 적이 있는 멤버에 대하여 지난 이력을 조회할 수 있다. |
| Test case | - 개별 이용자 A는 자신의 프로젝트에 대하여 다른 개별 이용자와 함께 작업하기를 원하여 멤버를 추가하였다.  - 프로젝트 SW의 팀장 B는 멤버 조회 작업을 한 후 지정된 멤버를 삭제하였다.  - 프로젝트 SW의 팀원 C는 멤버 조회 작업을 한 후 지정된 멤버를 삭제 또는 추가하려 했으나 둘 경우 모두 실패하였다. |

## CR-003 – 이슈 관리

|  |  |
| --- | --- |
| 항목 | 내용 |
| 상세 설명 | - 개별 이용자 몇 팀별 이용자(팀장, 팀원)는 프로젝트 내부에 존재하는 이슈에 대하여 관리 작업을 시행한다. |
| 선행조건 | - 로그인 : 이용자에 대한 신분 인증을 이루어진다.  - 프로젝트 관리 : 프로젝트에 대하여 CRUD 작업을 가능하다. |
| 후행조건 | - 이슈 생성 : 프로젝트와 관련된 이슈에 대하여 생성할 수있다.  - 이슈 삭제 : 생성된 이슈에 대하여 더 이상 다룰 필요가 없을 시 삭제를 한다.  - 이슈 알림 : 새로이 수정된 룰에 대하여 개별 이용자 및 팀별 이용자에게 알림한다. |
| Test case | 개별 이용자는 코드 내부에 대한 집중적인 검토를 위해 이슈를 생성하였다.  팀별 이용자는 이슈에 대하여 검토한 뒤 팀의 동의를 얻어 이슈를 삭제하였다.  개인 이용자 및 팀장은 동의없이 이슈에 대하여 삭제를 할 수 있다. |

## CR-004 – 룰 관리

|  |  |
| --- | --- |
| 항목 | 내용 |
| 상세 설명 | - 개인 이용자 및 팀별 이용자(팀장)는 룰 필터를 위해 체크박스 형식을 통해 룰 관리를 시행한다. |
| 선행조건 | - 로그인 : 이용자에 대한 신분 인증을 이루어진다. |
| 후행조건 | - 룰 선택 : 지정한 룰 필터에 대하여 선택할 수 있다.  - 룰 해제 : 지정한 룰 필터에 대하여 선택 해제 할 수 있다.  - 룰 설정 : 선택된 룰들에 대하여 적용한다. |
| Test case | 개인 이용자는 지정한 룰 필터를 선택하여 룰 설정을 하였다.  팀장은 프로젝트에 대한 룰은 지정하고 프로젝트 룰을 설정하였다. |

## CR-005 – 코드 컨벤션

|  |  |
| --- | --- |
| 항목 | 내용 |
| 상세 설명 | 개별 이용자 및 팀별 이용자(팀장, 팀원)은 소스에 대하여 코드 컨벤션 기능을 수행한다. |
| 선행조건 | - 로그인 : 이용자에 대한 신분 인증을 이루어진다.  - 소스 파일 관리 : 프로젝트 하위 생성된 폴더 및 파일에 대하여 CRUD작업이 가능하다.  - 룰 관리 : 소스에 대한 룰 필터 작업이 가능하다.  - 이슈관리 : 프로젝트 내부에 존재하는 이슈에 대하여 관리한다.  - 로그 확인 : 코드 컨벤션 또는 코드 리뷰 진행과정에 있어서 나타나는 로그를 보여준다. |
| 후행조건 | - 코드 컨벤션 : 설정된 룰을 바탕으로 소스 파일 또는 프로젝트에 대하여 코드 스타일을 체크한다. |
| Test case | 개인 이용자는 룰을 수정한 뒤 코드 컨벤션 이후 스타일에 대한 지시사항을 로그로 확인하였다.  프로젝트의 팀원은 프로젝트 룰에 대하여 자신의 코드를 코드 컨벤션한 후 로그를 확인하고 내부 코드를 수정하였다. |

## CR-006 – 코드 리뷰

|  |  |
| --- | --- |
| 항목 | 내용 |
| 상세 설명 | 팀별 이용자(팀장, 팀원)는 팀 프로젝트에 대하여 코드를 지적하고 기록을 남길수 있고, 실시간 채팅을 통하여 코드에 대하여 검사를 수행할 수 있다. |
| 선행조건 | - 로그인 : 이용자에 대한 신분 인증을 이루어진다.  - 소스 파일 관리 : 프로젝트 하위 생성된 폴더 및 파일에 대하여 CRUD작업이 가능하다.  - 이슈관리 : 프로젝트 내부에 존재하는 이슈에 대하여 관리한다.  - 알람 :  - 로그 확인 : 코드 컨벤션 또는 코드 리뷰 진행과정에 있어서 나타나는 로그를 보여준다. |
| 후행조건 | - 실시간 채팅 : 실시간으로 팀원간에 채팅을 통하여 코드 리뷰를 수행한다.  - 코드 마킹 : 프로젝트 내부에서 업데이트 및 수정이 된 코드에 대하여 플래그(마킹)을 남겨둔다.  - 코드 알람 : 변경된 코드 내역이 존재할 시 팀원에게 알림을 발송한다. |
| Test case | 팀원 A는 수정이 예상되는 부분에 대하여 코드 마킹을 하고 코드를 업데이트 하였다.  팀원 B는 팀원 A가 남긴 코드 마킹을 확인하고 그 부분에 대한 검토 이후 추가로 코드 마킹을 하였다.  팀장은 코드 마킹 알람을 확인하고 수정 내역을 승인하였다. |

## CR-007 – 코드 템플릿

|  |  |
| --- | --- |
| 항목 | 내용 |
| 상세 설명 | 개별 이용자 및 팀별 이용자(팀장, 팀원)는 프로젝트 및 소스 파일 추가 시 원하는 주제에 대한 샘플 코드를 얻을 수 있다. |
| 선행조건 | - 로그인 : 이용자에 대한 신분 인증을 이루어진다.  - 소스 파일 관리 : 프로젝트 하위 생성된 폴더 및 파일에 대하여 CRUD작업이 가능하다.  - 룰 관리 : 소스에 대한 룰 필터 작업이 가능하다.  - 이슈관리 : 프로젝트 내부에 존재하는 이슈에 대하여 관리한다.  - 로그 확인 : 코드 컨벤션 또는 코드 리뷰 진행과정에 있어서 나타나는 로그를 보여준다. |
| 후행조건 | - 템플릿 선택 : 이용자가 원하는 주제에 대하여 선택한다.  - 텀플릿 조회 : 원하는 주제에 대한 검색을 시도한다.  - 템플릿 커밋 : 자신이 만든 템플릿에 대하여 공유할 수 있다.  - 템플릿 로드 : 지정한 템플릿에 대하여 새로운 소스 파일을 생성한다. |
| Test case | - 개별 이용자는 템플릿 조회를 한 뒤 지정 템플릿을 선택하여 새로운 파일을 생성하였다.  - 개별 이용자는 자신이 생성한 소스에 대하여 템플릿 커밋을 하였고 다른 이용자로부터 조회 가능하다. |

## CR-009 – New

|  |  |
| --- | --- |
| 항목 | 내용 |
| 상세 설명 |  |
| 선행조건 |  |
| 후행조건 |  |
| Test case |  |