**Sw 코드 인스펙션**

1. 논의 대상
2. 전체 인스펙션 범위 지정
3. 품질관리 중에서 우선순위 결정
4. 인스펙션 범위 정하기
5. 현재 나와있는 인스펙션 툴 종류
6. 웹 앱과 앱 구성
7. 웹 앱 이외의 앱 만들지 여부
8. 웹 앱 UI 구성
9. 사용자가 올린 파일들에 대한 관리
10. 올린 파일들에 대한 접근 범위 : 프로젝터끼리 허용할지
11. 기존의 프로젝터끼리 연결하는 방식 : svn, git
12. 서버관리

* 서버 관리 방식 : 직접? 클라우드?

1. 시나리오
2. 파트 나누기
3. 토의진행
4. **인스펙션 범위 지정**
5. 품질관리 중에서 우선순위 결정
6. 인스펙션 범위 정하기
7. **기존에 있는 인스펙션 툴 종류**
8. **웹 앱과 앱 구성**

3.1 앱 만들지 여부

* 일단 웹 앱에 대한 정확한 그림이 필요할 것이고 웹 앱에 중점을 두고 개발하는 것이 0순위일듯하다.
  1. UI

|  |  |
| --- | --- |
| **페이지** | **내용** |
| 메인 | 웹 앱이름 게시, 로그인, 회원가입 |
| 사용자 개별 | 인스펙션 도구, 업로드, 다운로드, 팀구성, 팀 폴더 |
| 인스펙션 시작 | 품질관리 분류, 인스펙션 범위 지정, 작업 시작, 작업 중지, 업로드 |
| 인스펙션 결과 | (선택한 품질관리에 대한) 결과 – 시각적 요소, 다운로드, 공유 |
| 작업 결과 공유 | 공유 범위 지정 |

* + 1. UI 분류 및 구성

* 1. 고려사항
* 로그인 하지 않더라도 인스펙션 도구를 제공할 것인가?

1. **사용자가 올린 파일들에 대한 관리**
   1. 접근 가능 대상

* 개인
* 같은 프로젝터끼리 허용할지
  1. 기존의 프로젝터끼리 연결하는 방식 : svn, git

1. **서버관리** 
   1. 서버 관리 방식

* 직접?
* 클라우드?

1. **시나리오**

6.1. 사용자측면

- 웹에 접속한 뒤 로그인을 진행

- 사용자는 신이 원하는 품질관리 분류를 선택( 원하는 단계에 대한)한다.

- ( 룰에 대한 개별 적용 여부 )

- 인스펙션을 진행할 프로젝트 또는 소스코드를 업로드한다.

- 인스펙션 결과에 대한 페이지가 출력된다.

- 적용할 룰들을 선택하거나 양식에 맞게 업로드한 룰들을 프로젝트 및 소스 코드에 적용을 실행한다.

- 작업이 완료된 파일들을 내려받거나 또는 공유한다.

6.2 서버 측면

- 사용자가 로그인을 시도한다.

- DB로부터 회원을 조회한 뒤 로그인을 허용한다.

- 회원 기본 페이지를 출력한다.

6.3 파일 업로드

- 사용자가 올린 프로젝트 또는 소스코드가 웹에 업로드 된다.

- 웹으로부터 해당 파일들을 서버가 내려받는다.

- 서버 내부에서 해당 코드에 대한 인스펙션을 진행한다.

- 작업이 완료된 인스펙션 출력 결과물만 웹으로 반환한다.

- 웹에 페이지가 출력된다.

6.4 파일 인스펙션 처리

- 서버는 사용자로부터 받은 파일들에 대하여 품질관리를 진행한다.

- 웹에 인스펙션에 대한 결과물을 출력한 뒤 사용자가 원하는 수정작업 단계를 진행한다

: 룰 적용

- 서버는 사용자가 적용시킨 룰 값에 대하여 서버 내부에서 해당 프로젝트 또는 소스 코드에 대하여 룰적용을 진행한다.

- 룰 적용이 이상이 없을 경우, 웹에 성공 메시지를 출력한다.

- 사용자가 내려받기를 원할 경우 작업 파일을 사용자에게 넘겨준다. 공유를 원할 경우 서버 내부 저장소에 저장한다.

( 공유관련 문제 )

1. **파트 나누기**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 파트 | 기능 | 스터디파티 |
| 전성욱 |  |  | 0 |
| 권성욱 |  |  | 1 |
| 박문수 |  |  | 1 |
| 전효진 |  |  | 0 |
| 박슬기 |  |  | 1 |