**SE 2024 봄학기 텀프로젝트:**

**이슈관리 시스템 개발**

# 1. 프로젝트 개요

이슈관리 시스템

객체지향 분석, 설계 (문서화할 것!)

기능 확장(기본적인 기능만 하는 결과물 만들고 확장) / 설계 개선 활동(마찬가지) - JUnit 유닛으로 테스트 (테스트케이스, 설계 변경 등 과정을 문서화)

인터페이스 후보 : Jswing, spring boot ()

가산점 기능 -> 기능 확장으로 구현 가능

# 2. 역할분담

인터페이스 구현

* GUI (1) **함형우**
* 웹 (1) **김규리**

내부 비지니스 로직 구현 (1명 중심 구현 - 작은 이슈단위 추가 분배, 설계는 다 같이) **정지현** 이정인 김규리 함형우

DB 관리 (1) **이정인**

# 3. Inception

큰 requirement 정리

1. 계정 관리
   1. 계정 추가 (회원가입)
      1. 아이디 중복 검사
      2. 아이디 비번 조건
   2. 로그인 (아이디, 비번 확인)
   3. 로그아웃
   4. 아이디/비밀번호 찾기, 아이디/비밀번호 변경
2. 이슈 브라우즈 및 검색(issueList class 느낌)
   1. assignee 값, 이슈 상태, reporter 값 기준 검색
   2. title, reporter, date(자동 스탬프 기능 - junit 테스트를 위해 메소드로 빼두기), assignee, fixer(픽서는 리졸브된 순간), 우선순위, 이슈상태, 이슈처리상태, 이슈 종류,
   3. desc, 코멘트의 경우, 세부 사항 페이지에서 확인 가능
3. project 등록
   1. title, description, admin(총책임자 - 처음에 project를 등록하는 사람 )
4. project browse
   1. title admin, 내 권한
5. project 관리
   1. distributor 등록(only admin만 가능)
   2. distributor 권한 (PLayer ,developer, tester)
6. 이슈등록
   1. 상위 필드 - non null, null 필드 구분할 것
   2. 시간 자동 스탬프
7. 이슈 코멘트 추가
8. field : timestamp, 작성자, text
9. 이슈 상세 확인
   1. 상위 필드, 코멘트 확인 가능 (시간순 히스토리도 확인해야함)
   2. issue’s description
10. 이슈 상태 변경
    1. 변경 가능 필드 구분할 것
    2. 변경 시 타임 스탬프 기능
11. 이슈통계 (총체적 이슈관리)
    1. 이슈 등록 시간 (연월일, 시간(낮밤))
    2. 리포터 (단위시간 동안)
    3. 트랜드(코멘트 숫자, fixer,이슈 종류, 주요 키워드)
12. 가산점 - 자동추천
    1. 이슈 상세 - assignee 추천
    2. 다중의 인터페이스 방식으로 구현(모델 분리 필수, 다중의 인터페이스 지원을 위한 설계 문서화 필수, 잘 동작하는거 보이기)
13. 기타

# 4. 보고서

1. 요구정의 및 분석
   1. 유스케이스 다이어그램 (전체 유스케이스 모두, 단 includes 관계 1개 이상 포함, extends 관계 1개 이상 포함)
   2. **유스케이스 명세 (10개 이상의 유스케이스에 대해)**
   3. **도메인 모델 (Domain Model)**
   4. SSD (System Sequence Diagram), Operation Contract 등을 이용하여 요구 정의 및 분석 결과를 표현할 것. 텍스트 형태의 설명도 포함해야 함
   5. glossary (용어집)
2. 설계
   1. 클래스 다이어그램과 시퀀스 다이어그램은 해당 그림에 나타난 주요 구성요소들 예를 들어 클래스, 오브젝트, 메시지 등에 대한 설명이 포함되어야 함
   2. OOAD/GRASP 패턴/설계 원칙의 적용 부분 및 이유에 대해 설명
   3. 이외에도 적절한 UML 다이어그램들을 이용하여 다양한 설계 결과를 표현할 것
3. 구현
   1. 주요 기능들을 스크린 샷 등을 통해 표현
   2. 위의 내용을 설명하기 위해 다양한 UML 다이어그램의 활용이 추천됨 (클래스 다이어그램, 시퀀스 다이어그램 등)
4. 테스트 수행 내역
   1. 각 테스트 케이스에 대해 테스트 케이스를 만든 목적 등에 대한 설명이 포함되어야 함. 기존 기능들이 잘 동작하는 것도 보여야 하며, 특히 새롭게 추가한 기능의 경우 각 추가 기능들이 잘 동작한다는 것을 효과적으로 보여주어야 함.
   2. JUnit을 활용했다면 테스트 코드도 함께 제시
   3. 테스트 케이스 적용 결과 (스크린 샷)
5. GitHub 프로젝트 활용 요약
6. 팀의 github 프로젝트 주소
7. 프로젝트 progress history 스크린샷
8. 팀원별 기여를 잘 나타낼 수 있는 각종 자료
9. 비디오클립

6. Risk requirement(개발 우선 순위)

1. 프로젝트 생성 구현
2. issue 관련 구현
3. 계정 관련(로그인, 로그아웃 등등)

7. UX 분석

모든 사용자는 user 계정을 생성한다

로그인한 user는 자신에게 주어진 플젝 리스트 + 플젝 생성이 보인다

-프로젝트 리스트 테이블에는 title, admin, user의 권한 filed를 가지고 있다.

user는 프로젝트를 생성할 수 있다. (이때 생성한 user = admin)

- 프로젝트 생성은 title, description을 요구한다(둘다 필수)

각 프로젝트의 admin은 PL, tester, developer를 추가할 수 있음(id 로 추가)

각 프로젝트는 이슈 리스트를 가진다

* 이슈 리스트 테이블은 title, 이슈종류, 우선순위, 이슈상태, reporter, assignee, fixer, date로 구성
* assignee, fixer는 null일 수 있다.(데이터상 null X, 걍 지정 X인거임)

각 프로젝트의 tester는 새로운 이슈를 생성할 수 있다.

* 이슈 생성 시, 이슈상태는 new로 고정
* date는 타임스탬프로 고정
* reporter는 user로 고정
* title, 우선순위, 이슈종류, 코멘트는 필수로 작성
* 이슈의 description은 optional

유저는 이슈 하위에 comment를 생성할 수 있다. 이슈를 생성하거나, 이슈상태를 수정할 경우, 코멘트를 필수로 작성한다.

* 코멘트는 text를 필수로 작성
* reporter는 user로 고정, date는 타임스탬프로 고정

이슈는 하위 필드와 코멘트 리스트, 수정버튼을 가진다.

**이슈 수정 시, 이슈 상태가 변경된다.**

PL은 이슈 상태가 new, resolved, reopened인 이슈만 수정할 수 있다.

* 이슈 상태는 new, reopened -> assigned / resolved -> closed 로 고정
* 이슈 상태가 new, reopened일 경우, developer 중 한 명을 assignee로 필수로 선택
* 코멘트는 필수로 작성
* date는 타임스탬프로 고정

developer는 자신이 assignee로 지정되고, asigned 상태인 이슈만 수정할 수 있다.

* 코멘트는 필수로 작성
* date는 타임스탬프로 고정,
* fixer는 본인(dev)으로 고정
* 이슈 상태는 fixed로 고정

tester는 자신이 reporter이고 이슈 상태가 fixed인 경우에만 수정할 수 있다.

* 이슈 상태는 reopened, resolved 중 하나로 필수로 선택한다
* 코멘트를 필수로 작성
* date는 타임스탬프로 고정

# 8. 계획

5/12 (일)

- 유스케이스 명세(이정인, 함형우), 도메인 모델(함형우), 클래스 명세(정지현)

5/14 (화)

- UI 설계 (김규리)

5/17 (금)

5/21 (화)

5/24 (금)

5/28 (화)

시작

* 최대 지연 시 보고서 작성 시작 시기

끝

* 최대 지연 시 다이어그램, 시각자료 정리, 토대 완성 시기

5/31 (금)

끝

* 최대 지연 시 보고서 완성

6/2 (일) - 최종