

**SE 2024 봄학기 텀프로젝트**  
**이슈관리 시스템 개발**  
**프로젝트 최종 보고서**

**2024.06.03**

**TermProject Team 16**

**김재원(20201#72)**

**신나리(20212#43)**

**신형철(20204#93)**

**안지영(20212#38)**

**양희목(20220#67)**



프로젝트 요약	3
요구 정의 및 분석	3
설계	5
상세 구현	5
테스트 수행 내역	5
Github 활용 요약	5

프로젝트 요약  
요구 정의 및 분석

1. Use-case Diagram

2. Use-case spec

1. use case(bug report)

1. 테스터가 로그인한다.
2. 테스터는 이슈를 발행한다.
3. 테스터는 이슈의 우선 순위, Title, Description. Comments를 입력하고 reporter, reported date , 이슈상태 는 서버가 입력한다.

2. use case(bug resolve)

1. 테스터는 자신이 등록한 이슈를 검색한다.
2. 검색된 이슈를 열고 상태가 fixed인 이슈는 resolve상태로 바꿀 수 있다.

3. use case(admin project setting)

1. 관리자는 로그인한다.
2. 관리자는 프로젝트를 생성, 추가한다.
3. 관리자는 프로젝트의 이름을 입력한다.
4. 관리자는 프로젝트의 요약본을 입력한다.
5. 관리자는 프로젝트의 컴포넌트를 1개 이상 설정한다.
6. 관리자는 관리할 프로젝트의 저장소를 설정한다.

4. use case(project account add)

1. 관리자는 프로젝트를 선택한다.-
2. 그 프로젝트에 계정을 추가한다.
3. 각 계정에 권한을 추가한다.

5. use case(Log in)

1. 사용자는 아이디와 비밀번호를 입력한다.
2. 아이디와 비밀번호가 일치할 시 계정에 로그인한다.
3. 아이디와 비밀번호가 불일치할 시 에러 메시지가 뜨고, 재시도한다.

6. use case(이슈 통계)

1. 사용자가 로그인한다.
2. 사용자는 프로젝트를 선택하고 원하는 일/월을 선택한다.
3. 해당 기간에 발생한 이슈 발생 횟수 통계를 조회한다.

7. use case(이슈 배정-PL)

1. PL은 자신의 계정으로 로그인한다.
2. PL은 new 상태의 이슈를 검색한다.
3. 검색된 이슈 중 하나를 선택하여 내용(description, 코멘트)을 살펴본 뒤 개발자 중 하나를 선정해 배정한다.
4. PL이 자동배정 기능을 사용할 경우 시스템은 적합한 개발자를 배정한다.

8. use case(개발자 자동 배정- server)

1. 시스템이 해당 이슈의 정보를 바탕으로 가장 적합한 개발자를 선택하여 배정한다.

9. use case(issue fix - dev)

1. 개발자가 로그인한다.
2. 개발자는 검색 기능을 이용해 자신에게 할당된 이슈들을 확인한다.
3. 각 이슈에 대해 코멘트를 추가한다.
4. 해결 가능하면 상태를 fixed로 바꾼다.
5. 서버는 fixed된 이슈에 대해 fixer를 해당 개발자로 등록한다.

## 10. use case(browse)

1. 사용자가 로그인한다.
2. 사용자는 이슈를 조회할 프로젝트를 선택한다.
3. 프로젝트에 발생한 이슈 목록 중 이슈를 선택하고 조회한다.

## 11. use case(issue close - PL)

1. PL은 자신의 계정으로 로그인한다.
2. PL은 Resolved 상태로 된 이슈를 검색한다.
3. PL은 해당 이슈의 내용(코멘트)을 확인하고 이상이 없을 시 이슈의 상태를 Closed로 변경한다.

## 3. Domain Model

## 4. System Sequence Diagram

## 5. Operation Contract

## 설계

## 1. Class Diagram

## 2. Sequence Diagram

## 3. OOAD

## 4. MVC

## 상세 구현

## 테스트 수행 내역

## 1. JUnit

## Github 활용 요약

Use Case Name	Bug report
Level	User goal
Primary Actor	테스터
Sketeholders and Interests	빠르고 정확한 이슈 제보,
Preconditions	이슈를 등록하는 유저는 로그인 상태여야 한다.
Success Guarantee	새로운 이슈가 시스템에 등록되었다. 이슈 통계가 업데이트 되었다.
Main Success Scenario	1. 테스터가 이슈 관리 시스템에 로그인한다. 2. 해당 테스터가 속한 프로젝트 중 제보할 버그가 속한 프로젝트를 선택한다. 3. 새로운 이슈를 생성하고 해당 버그에 관련된 정보를 입력한다.
Extentions	Log in

Use Case Name	Issue resolve
Level	User Goal
Primary Actor	Developer
Sketeholders and Interests	
Preconditions	해당 개발자에게 배정된 이슈가 1개 이상 존재해야 한다.
Success Guarantee	해결된 이슈의 상태가 fixed로 변경된다. 해당 개발자에게 배정된 이슈의 개수가 줄어든다.
Main Success Scenario	1. 개발자가 개발자 계정으로 이슈 관리 시스템에 로그인한다. 2. 해당 개발자가 속한 프로젝트 중 하나를 선택하여 들어간다. 3. 개발자에게 배정된 이슈를 보고, 그 중 하나를 선택한다. 4. 선택된 이슈를 해결하고 관련 내용으로 코멘트를 남긴다.
Extentions	Leave comment, Log in

Use Case Name	Project initialize
Level	User Goal
Primary Actor	Administrator
Sketeholders and Interests	
Preconditions	
Success Guarantee	해당 설정으로 프로젝트가 생성된다.
Main Success Scenario	1. 관리자가 관리자 계정으로 로그인한다. 2. 프로젝트 생성 화면으로 넘어간다. 3. 프로젝트에 필요한 정보를 입력하고 저장한다.
Extentions	

Use Case Name	Project account add
Level	User goal
Primary Actor	Administrator
Sketeholders and Interests	
Preconditions	프로젝트가 1개 이상 생성되어 있어야 함.
Success Guarantee	해당 프로젝트에 지정된 권한을 가진 유저가 추가됨
Main Success Scenario	1. 관리자가 관리자 계정으로 로그인한다. 2. 유저를 추가할 프로젝트를 선택한다. 3. 추가할 유저의 정보를 입력하고 저장한다.
Extentions	Log in

Use Case Name	Log in
Level	User Goal
Primary Actor	All users
Sketeholders and Interests	정확한 정보를 입력했을 때 해당 계정으로 인증.
Preconditions	
Success Guarantee	유저에게 계정에 맞는 권한을 부여.
Main Success Scenario	1. 유저가 이슈 관리 시스템에 접속한다. 2. 유저의 정보를 입력받아 로그인을 시도한다. 3. 저장소에 저장된 정보와 입력한 정보가 일치한다. 4. 로그인에 성공하고 해당 유저에게 맞는 권한을 부여한다.
Alternation Scenario	3.a 유저의 ID와 PW가 일치하지 않는다. 4. 2번으로 돌아가 입력을 다시 받는다.
Extentions	

Use Case Name	Issue static
Level	User goal
Primary Actor	All users
Sketeholders and Interests	정확한 기록을 바탕으로 한 통계 정보 열람,
Preconditions	유저는 로그인이 완료된 상태여야 한다.
Success Guarantee	원하는 기간의 이슈 통계 기록이 화면에 출력된다.
Main Success Scenario	1. 유저는 로그인 하여 자신이 속한 프로젝트 중 통계를 보기 원하는 프로젝트에 들어간다. 2. 통계 기록을 원하는 기간을 입력한다. 3. 시스템에서 저장소에 기록된 정보를 바탕으로 해당 기간의 통계를 반환한다.

Alternative Scenario	2a. 입력된 기간에 아직 프로젝트가 생성되지 않았었다. 3. 기록이 존재하지 않는다고 안내하고 다시 2번으로 돌아가 새로운 입력을 받는다.
Extentions	Log in
Use Case Name	Issue assign
Level	User goal
Primary Actor	Project Leader(PL)
Sketeholders and Interests	적당한 개발자에게 적당한 이슈를 배정한다
Preconditions	PL의 계정으로 로그인 완료. 해당 프로젝트에 1개 이상의 배정되지 않은 이슈 존재. 해당 프로젝트에 1명 이상의 개발자 존재.
Success Guarantee	해당 이슈의 상태가 assigned로 변경된다. 배정 받은 개발자의 해결할 이슈가 한 개 늘어난다.
Main Success Scenario	1. PL이 자신의 계정으로 로그인 한다. 2. 자신이 관리자로서 있는 프로젝트 중 원하는 프로젝트에 들어간다. 3. 새롭게 등록된 이슈를 검색하여 하나를 선택한다. 4. 내용을 살펴본 뒤 해당 프로젝트에 속해있는 개발자 중 하나를 선택해 배정한다.
Alternative Scenario	4.A 자동 배정을 선택한다 5. 자동 배정 시스템이 개발자의 이력등을 고려하여 가장 적합한 개발자를 추천하여 전달한다. 6. PL이 해당 개발자로 배정한다.
Extentions	Auto developer recommendation
Use Case Name	Auto developer recommendation
Level	
Primary Actor	System
Sketeholders and Interests	알고리즘을 통해 최적의 개발자에게 이
Preconditions	해당 프로젝트에 1명 이상의 개발자 존재.
Success Guarantee	해당 이슈에 최적인 개발자 하나가 추천된다.
Main Success Scenario	1. 시스템에 이슈와 해당 프로젝트에 참여중인 개발자 목록이 주어진다. 2. 자체 알고리즘을 통해 가장 적합한 개발자 하나를 선정한다. 3. 해당 개발자를 추천한다.
Alternative Scenario	
Extentions	
Use Case Name	Issue fix



Level	User goal
Primary Actor	Developer
Sketeholders and Interests	
Preconditions	자신에게 1개 이상의 이슈 배정된 상태
Success Guarantee	자신에게 배정된 이슈 1개 감소. 해당 이슈의 상태가 fixed로 변경.
Main Success Scenario	1. 개발자가 자신의 계정으로 로그인 2. 자신이 속한 프로젝트 중 하나를 골라 입장. 3.
Alternative Scenario	
Extentions	

Use Case Name	
Level	
Primary Actor	
Sketeholders and Interests	
Preconditions	
Success Guarantee	
Main Success Scenario	
Alternative Scenario	
Extentions	

Use Case Name	
Level	
Primary Actor	
Sketeholders and Interests	
Preconditions	
Success Guarantee	
Main Success Scenario	
Alternative Scenario	
Extentions	