고프트웨어공학 19조 2024 SE 화분반



이유관리제스템 절계 및 구현

5

6

# Software Engineering Team 19() { = 회억원, 유성훈, 장진우, 이승현 ; /

### 소프트웨어공학 01분반 TEAM 19

```
목차
프로젝트 개요;
요구 정의/분석
절계 및 데이터 모델링;
구현;
```

프로젝트 개요

l 1

0

\_\_

# 이유관리시으템

버그 혹은 요청

고프트웨어 개발 과정에서 이유 목록 및 내용을 체계적으로 관리하는 고프트웨어



프로젝트 진행 문제(이슈) 함께 공유 및 수정 => 안정적인 소프트웨어 개발

9 <u>프로젝트 개요</u> 10

이유관리시스템

이유관리제스템의 핵심적인 3기능

이유 등록

이유 추억

이유 해결

요구정의 역할 정의 ; 역할 이름 역할 역할 정의 Admin 계정 관리 계정의 역할 부여 및 관리 프로젝트 관리 이유 해결을 위한 담당까 배정 PL 이유에 대한 구정 및 보완 해결 OI유 해결 Dev 이유맹정 및 확인 이유 등록 및 해결 여부 확인 **Tester** 

요구사항

계정생성

이슈검색

이슈등록

코멘트추가

상세정보

담당자배정

상태변경

추가기능

٠

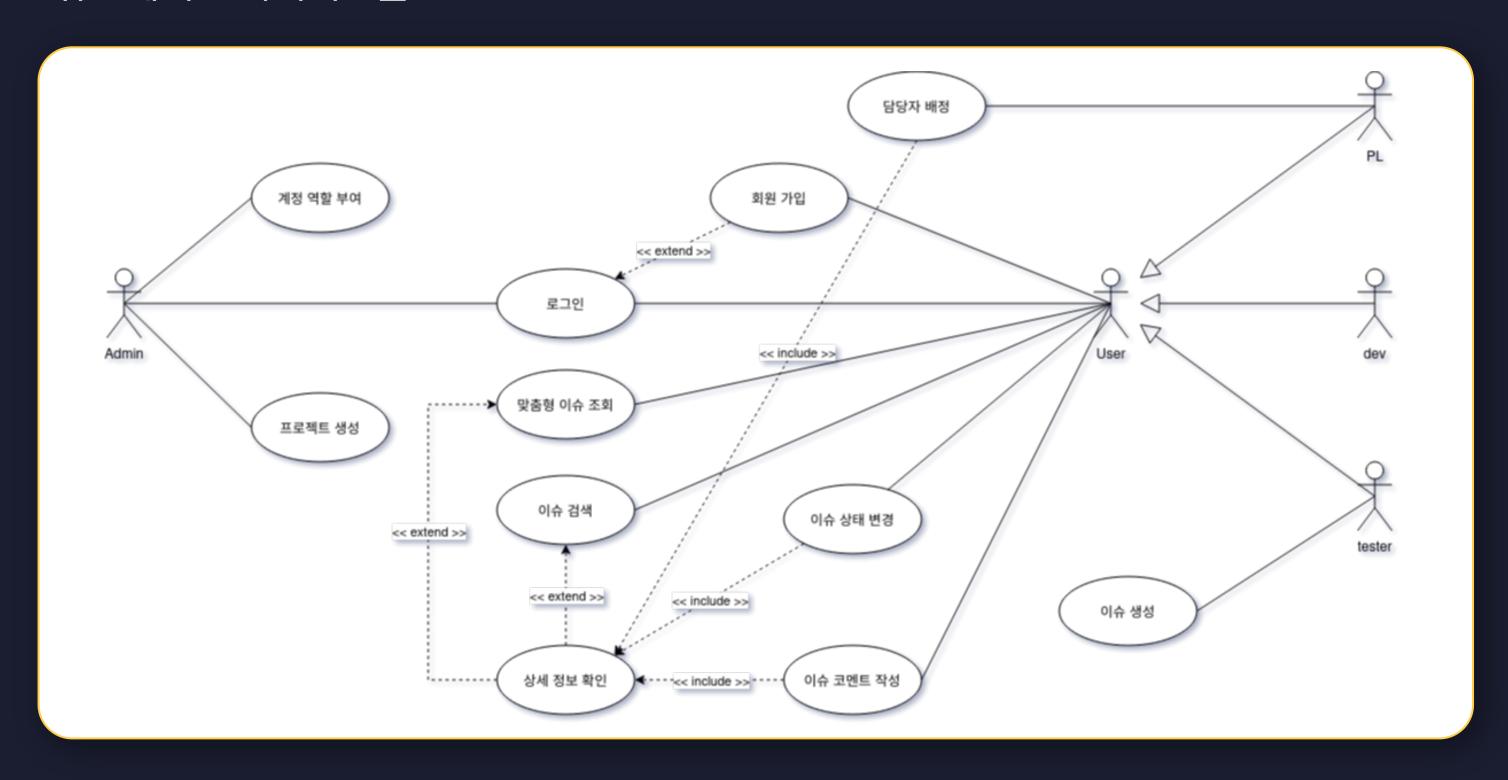
맞춤형 이슈 조회





つ1

### 유그게이그 다이어그램



유스케이스 명제

이슈 생성

지작 조건: tester계정으로 로그인 되어 있어야 함

1 Tester가 "이슈 생성"을 선택한다
2 시스템이 이슈생성 양식을 화면에 출력
3 Tester가 양식에 맞춰 내용 작성
4 시스템이 입력된 이슈 정보 저장, 이슈 상태 새로 지정

3A

제목을 작성하지 않은 상태로 생성버튼 눌렀을 경우 오류 메시지를 출력한다

Tester가 오류 메시지를 확인한 후 다시 입력한다

생성된 이슈가 시스템에 추가된다



### 유스케이스 명제

### 이슈 검색

시작 조건: User계정으로 로그인/ 1개 이상 이슈가 생성되어 있어야 한다

- User가 이슈검색 화면으로 이동한다
- User가 찾고자 하는 제목, 담당자, 상태 등의 검색어 를 입력한 뒤 '검색'버튼을 누른다
- 시스템이 검색어와 일치하는 이슈 조회하여 목록으로 화면 출력한다.
- User가 출력된 목록을 확인한다.

**Extension Point** 

상세정보 확인

검색어에 해당하는 이슈들이 화면에 출력된다

유스케이스 명제

맞춤형 조회

시작 조건: User계정으로 로그인

- User가 '나의 이슈 보기'를 클릭한다
- 시스템이 User의 역할과 관련한 이슈들을 조회하여 출력한다
- 시스템이 검색어와 일치하는 이슈 조회하여 목록으로 화면 출력한다.
- User가 출력된 목록을 확인한다. 4

**Extension Point** 

상세정보 확인

User와 관련한 이슈들이 화면에 출력된다

# 요구정의 유스케이스 명제 상세정보확인 시작 조건: User계정으로 로그인되어 있어야 한다 User가 이슈 하나 선택하여 클릭한다 2 시스템이 해당 이슈의 정보들을 조회하여 화면에 출력한다

3

이슈의 상세정보가 화면에 출력된다

User가 해당 정보를 확인한다

유스케이스 명제

이슈상태변경

시작 조건: User계정으로 로그인

- 1 (include(상세정보 확인)
- 2 User가 '상태변경' 버튼을 누른다
- 3 ( 시스템이 User의 정보 및 이슈의 상태, 담당자 정보를 비교한다
- 4 ( 시스템이 해당 이슈의 상태를 다음단계의 상태로 변경한다)

**3A** 

User가 담당자가 아닌 이슈의 상태 변경버튼을 누른다

시스템이 담당자 불일치 오류 메시지를 출력한다

User가 오류 메시지를 확인하고 유스케이스를 마친다

해당 이슈의 상태가 순차적으로 다음 단계의 상태로 변경/저장된다

요구정의 요구정의 유스케이스 명에 2 3 코멘트 작성

1 include(상세 정보 확인)

2 시스템이 코멘트 입력 양식을 화면에 출력한다

시작 조건: tester계정으로 로그인 되어 있어야 함

3 (User가 코멘트를 작성한 후, '작성'버튼을 누른다

시스템이 입력된 코멘트 정보를 저장한다

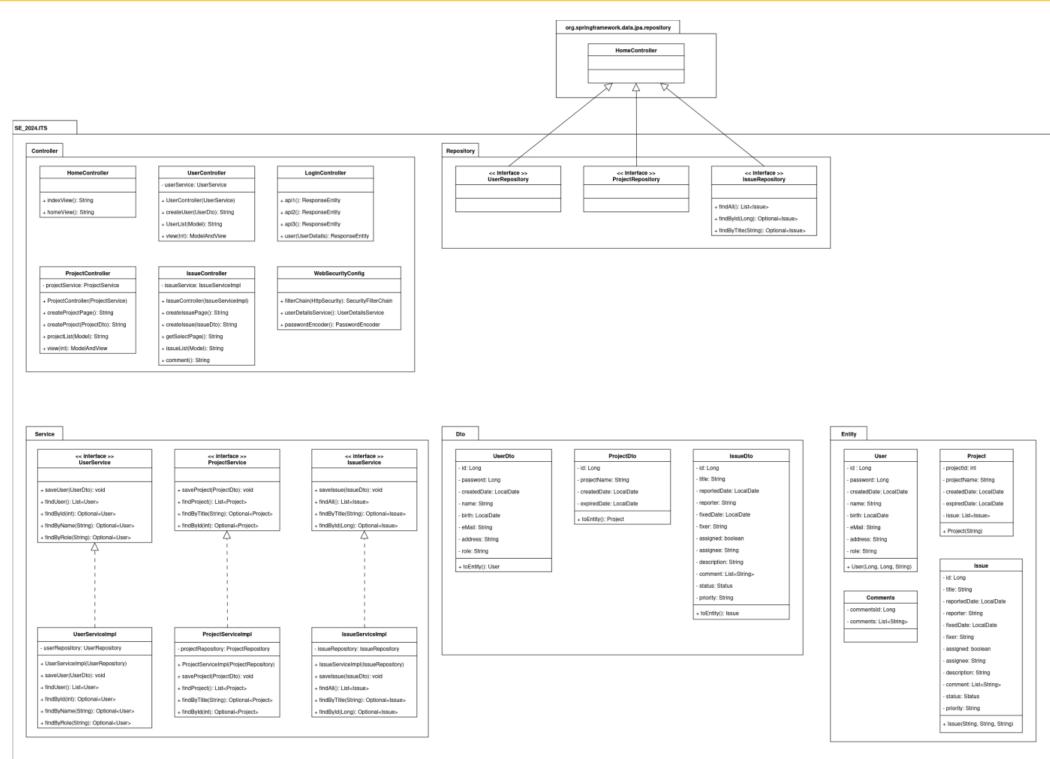
3A

User가 양식을 작성하지 않은 상태로 '작성'버튼을 눌렀을때 오류 메시지를 출력한다

> Tester가 오류 메시지를 확인한 후 코멘트를 다시 입력한다

해당 코멘트가 성공적으로 출력된다

9 교구분석 10 클래스 다이어그램 12



19

\_ `

23

24

24

25

26

## 요구분석

### 클래스 다이어그램

### Controller

#### HomeController

- + indexView(): String
- + homeView(): String

#### UserController

- userService: UserService
- + UserController(UserService)
- + createUser(UserDto): String
- + UserList(Model): String
- + view(int): ModelAndView

### LoginController

- + api1(): ResponseEntity
- + api2(): ResponseEntity
- + api3(): ResponseEntity
- + user(UserDetails): ResponseEntity

### ProjectController

- projectService: ProjectService
- + ProjectController(ProjectService)
- + createProjectPage(): String
- + createProject(ProjectDto): String
- + projectList(Model): String
- + view(int): ModelAndView

### IssueController

- issueService: IssueServiceImpl
- + IssueController(IssueServiceImpl)
- + createIssuePage(): String
- + createIssue(IssueDto): String
- + getSelectPage(): String
- + issueList(Model): String
- + comment(): String

### WebSecurityConfig

- + filterChain(HttpSecurity): SecurityFilterChain
- + userDetailsService(): UserDetailsService
- + passwordEncoder(): PasswordEncoder

## 요구분억

### 클래스 다이어그램

#### Entity

#### User

- id : Long
- password: Long
- createdDate: LocalDate
- name: String
- birth: LocalDate
- eMail: String
- address: String
- role: String
- + User(Long, Long, String)

#### Comments

- commentsId: Long
- comments: List<String>

#### Project

- projectId: int
- projectName: String
- createdDate: LocalDate
- expiredDate: LocalDate
- issue: List<Issue>
- + Project(String)

#### Issue

- id: Long
- title: String
- reportedDate: LocalDate
- reporter: String
- fixedDate: LocalDate
- fixer: String
- assigned: boolean
- assignee: String
- description: String
- comment: List<String>
- status: Status
- priority: String
- + Issue(String, String, String)

9







### 0

11

12

13

14

15

16

Τ (

TΩ

ΤŽ

20

21

22

**Z**5

2۲

24

25

20

### 클래스 다이어그램

#### Proje

- id: Long

Dto

- password: Long
- createdDate: LocalDate

UserDto

- name: String
- birth: LocalDate
- eMail: String
- address: String
- role: String
- + toEntity(): User

#### ProjectDto

- id: Long
- projectName: String
- createdDate: LocalDate
- expiredDate: LocalDate
- + toEntity(): Project

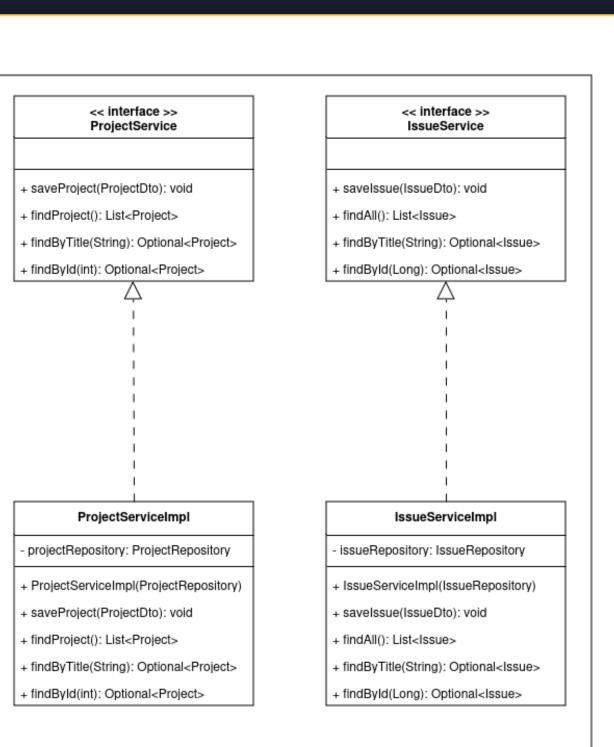
#### IssueDto

- id: Long
- title: String
- reportedDate: LocalDate
- reporter: String
- fixedDate: LocalDate
- fixer: String
- assigned: boolean
- assignee: String
- description: String
- comment: List<String>
- status: Status
- priority: String
- + toEntity(): Issue

9 교구분석

### 클래스 다이어그램

### Service << interface >> UserService + saveUser(UserDto): void + findUser(): List<User> + findByld(int): Optional<User> + findByName(String): Optional<User> + findByRole(String): Optional<User> UserServiceImpl userRepository: UserRepository + UserServiceImpl(UserRepository) + saveUser(UserDto): void + findUser(): List<User> + findByld(int): Optional<User> + findByName(String): Optional<User> + findByRole(String): Optional<User>



### 클래스 다이어그램

