 use case 1

l   Admin이 project1을 추가함. 또한 PL계정, 개발자 계정, 테스터 계정을 추가함.

l   tester1이 이슈를 하나 생성하고, 해당 이슈의 코멘트를 하나 추가함. 이후 시스템은 해당 이슈의 상태를 new로 변경하고, reporter의 값은 tester1으로 저장함.

l   PL1은 모든 이슈들을 브라우즈 함. 이후 new 상태의 이슈들을 검색하고, 위에서 tester1이 reporter로 되어 있는 이슈에 대해 dev1을 담당자(assignee)로 지정하며, 코멘트에 적절한 메시지를 추가함. 이후 시스템은 해당 이슈의 상태를 assigned로 변경함

l   dev1은 이슈 검색 기능으로 자신에게 assign 되어 있는 이슈들을 브라우즈 함. 이슈들에 대해 남겨진 코멘트를 포함하여 다양한 정보를 확인하고, 코드를 고친 후 (ITS에는 없는 기능임. 타 시스템을 이용해서 한다고 가정), 코멘트를 추가한 후 해당 이슈를 fixed로 바꿈. 이렇게 되면 해당 이슈의 fixer는 dev1으로 등록됨.

l   tester1은 자신이 report하였던 이슈 중에 fixed된 이슈들을 검색하고, 적절히 고쳐진 이슈들을 선택하여 “resolved” 상태로 바꿈.

l   PL1은 resolved 상태인 이슈들을 선택해서 closed로 바꿈.

l   optional 기능 (추가 배점): PL2는 현재 new 상태인 이슈들을 브라우즈하고, 상세 정보 보기 기능을 통해 특정 이슈의 상세한 내용과 기존 코멘트들을 살펴봄. 이때 시스템은 fixed된 이슈들의 이력을 이용해서 가정 적절한 개발자를 추천해줌. (예를 들어 “best candidate: dev2, dev5, dev1 등으로”, 가장 가능성이 높은 후보 3명을 순서대로 추천함)

use case 2

-       관리자가 프로젝트 기본 설정(이슈 필드) 입력

-       관리자는 프로젝트가 저장된 관리소를 설정한다.

use case

-       관리자가 계정을 생성하고 계정마다 권한을 부여한다.(ex 티켓 생성, 수정, 로그 접근, 등)

-

use case 3

-       PL은 들어온 티켓을 개발자에게 할당한다.

-       해결되지 않은 티켓이 있으면 다른 개발자에게 할당한다.

-       resolve된 이슈는 closed로 바꿈(저장)

use case 4

-       유저는 이슈(이슈, 코멘트로 이루어짐)를 등록한다.

-       자신이 등록한 이슈에 한해 티켓을 검색할 수 있다.

-       fixed 이슈는 resolve상태로 바꿀 수 있다.

-       서버는 티켓이 등록된 날짜를 자동으로 기록한다.

use case 5

-       개발자는 자신에게 할당된 티켓을 브라우즈할 수 있다.

-       티켓의 상태를 fixed로 바꿀 수 있다.

-       서버는 해당 티켓의 fixer를 해당 개발자(티켓의 상태를 fixed로 바꾼)로 등록한다.

use case 6

-       사용자는 서버에 로그인 할 수 있다.

use case(bug report)

-       테스터가 로그인한다.

-       태스터는 이슈을 발행한다.

-       테스터는 이슈의 우선 순위, Title, Description. Comments를 입력하고 reporter, reported date , 이슈상태  는 서버가 입력한다.

use case(bug resolve)

-       테스터는 자신이 등록한 이슈를 검색한다.

-       검색된 이슈를 열고 상태가 fixed인 이슈는 resolve상태로 바꿀 수 있다.

use case(admin project setting)

-       관리자는 로그인한다.

-       관리자는 프로젝트를 생성, 추가한다.

-       관리자는 프로젝트의 이름을 입력한다.

-       관리자는 프로젝트의 요약본을 입력한다.

-       관리자는 프로젝트의 컴포넌트를 1개 이상 설정한다.

-       관리자는 관리할 프로젝트의 저장소를 설정한다.

use case(project account add)

-       관리자는 프로젝트를 선택한다.-

-       그 프로젝트에 계정을 추가한다.

-       각 계정에 권한을 추가한다.

use case(Log in)

* 사용자는 아이디와 비밀번호를 입력한다.
* 아이디와 비밀번호가 일치할 시 계정에 로그인한다.
* 아이디와 비밀번호가 불일치할 시 에러 메세지가 뜨고, 재시도한다.

use case(이슈 통계)

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

use case(이슈 배정-PL)

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

use case(개발자 자동 배정- server)

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

use case(issue fix - dev)

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

use case(browse)

* 사용자가 로그인한다.
* 사용자는 이슈를 조회할 프로젝트를 선택한다.
* 프로젝트에 발생한 이슈 목록 중 이슈를 선택하고 조회한다.

use case(issue close - PL)

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