

# 회의실예약시스템

우리가아마도여기서이름이제일길겠조

조원 : 2013105088 장효원

2013105080 이지원

2013105087 장호택

2013105084 임영규

## **프로젝트 설명**

최근 우리나라에서는 전국적인 대학간 협력 연구 과제가 활발히 이루어지고 있다. 예를 들어 ITRC 소프트웨어 센터에는 열 개의 대학이 참여하고 있으며 그 중 일곱 개의 대학은 서울에, 나머지 세 대학은 각각 대전, 대구와 전주에 위치하고 있다. 이러한 전국 단위의 협력 연구에서는 크고 작은 회의와 세미나가 빈번히 개최되는데, 매 번 회의 장소를 선정하기가 쉽지 않다. 회의실을 알아보기 위해서는 특정 도시에서 특정 일자에 사용 가능한 회의실을 찾아봐야 하는데, 현재 이는 호텔, 사립 강의실, 기차역 회의실 등 여러 군데에 문의를 한 뒤에야 알 수 있다. 이러한 불편함을 해소하고자 우리는 전국에 있는 회의실을 검색하고 예약까지 할 수 있는 회의실 예약 시스템을 제작하고자 한다.

## **비지니스 요구 사항**

1. 등록된 사용자만이 시스템을 사용할 수 있다.
2. 기업 사용자는 회의실에 대한 정보 – 위치와 수용 인원 수, 대여 비용과 날짜별 가용 회의실 수 등을 시스템에 등록할 수 있다.
3. 등록된 사용자는 시스템을 사용하여 예약 가능한 회의실을 검색할 수 있다.
4. 검색한 일자와 도시에서 가용한 회의실이 없는 경우 시스템은 차선택을 제시한다.

## 기존 시스템에 대한 고찰

### 1. 기능 부족의 문제

기존 시스템에 구현되어 있는 기능은 회의실 등록, 등록된 회의실 목록 조회, 기업사용자용 회원가입, 회의실 예약, 전체 회의실 목록 조회, 로그인으로서 회의실 삭제, 일반사용자용 회원가입, 회의실 예약 취소 등등이 없어 원래 계획했던 것보다 시스템의 기능의 수가 부족하였다.

### 2. 기능 오류의 문제

기존 시스템에 구현되어 있는 기능 중 회의실 예약을 하면 예약현황이 테이블에 바로 반영되지 않고 프로그램을 종료한 뒤 다시 실행해야 반영이 되는 문제가 발생하였다.

### 3. 유지 보수의 문제

기존 시스템의 코드는 매우 복잡하였다. 기능을 구분하지 않고 하드코딩을 하였기 때문에, GUI에서 데이터를 처리하고, 서버에서 회의실 처리를 하는 등 기능들이 많이 섞여있으며, 하나의 메서드 안에 여러 기능이 들어있어 코드의 내용을 식별하기 어려웠다. 그리고 과도한 클래스의 연결 관계로 인해 복잡도가 아주 높았다. SonarQube에서 SQALE Rating은 B, Technical Debt Ratio는 10.1%, Duplications는 6.8%, Issues는 789를 평가받아서 시스템의 구조를 대폭 개선해야만 했다.

SQALE Rating

B

Technical Debt Ratio

10.1%

Duplications

6.8%

Lines Blocks Files

336 17 8

Issues

789

🚫 Blocker	4	
🔴 Critical	32	■
🔴 Major	449	■■■■■■■■■■
🟢 Minor	285	■■■■■■■■■■
🟢 Info	19	■

## 과제 목표 및 세부 목표

1. Re-engineering	소프트웨어의 구현이란 단지 디자인을 코드로 번역하는 과정이다. 하지만 기존 시스템의 디자인이 너무 복잡하여서 새로운 팀원들이 기존의 시스템을 분석하고 디자인을 발전시키는데 너무 많은 비용이 들 것으로 예상되어 디자인의 구조를 새롭게 만들고 시스템을 새롭게 작성하게 되었다.
2. 기능 추가	기존의 시스템에 구현할 예정이었으나 구현하지 못한 회의실 예약 취소, 로그아웃, 일반사용자용 회원가입, 회의실 예약 현황 조회와 새롭게 구현하기로 한 회의실 정보를 텍스트 파일로 다운받기 추가적으로 구현하기로 하였다.
3. 기능 오류 개선	Re-engineering을 하면서 코드를 새롭게 작성하면서 이전에 발생했던 똑같은 오류가 나지 않도록 하였다.
4. 코드의 분류화	앞서 언급했듯이 기존의 코드가 너무 하드코딩이 되어 있어서, Re-engineering을 하면서 코드를 알맞게 분류하고, 하드코딩을 최대한 줄여서 유지보수가 용이하게 하였다.
5. 기능의 서브모듈화	기능들을 한 곳에 모아서 구현하지 않고 잘게 쪼개어 서브모듈화를 시켜서 유지보수가 용이하게 하였다.

## 추진 전략

1. 커뮤니케이션을 최대화	팀원 모두가 지금까지 프로젝트를 각자 해오면서 느낀 공통된 불편함은 서로 커뮤니케이션이 잘 되지 않아 프로젝트가 망했던 점이였다. 이러한 점이 이번에는 발생하지 않도록 서로 커뮤니케이션을 최대화 할 수 있도록 Trello를 활용하여 서로가 전하고 싶은 바를 확실히 하였다.
2. Git 최대한 활용	조장을 제외한 팀원들이 git을 사용해보지 않아서 git에서 많은 문제가 발생하였다. 프로젝트 초반에는 push, pull 개념도 확실하지 않았고, conflict가 자주 발생하여 reset을 여러 번 하였다. 그래서 일주일동안 집중적으로 git에 대해서 학습하게 되었고, git으로 인한 문제는 발생하지 않았다.
3. 디자인 패턴 학습	기존 시스템이 하드 코딩되었던 주요 이유가 구현하려는 부분을 어떻게 해야할지에 대해 고민 없이 생각난대로 구현하여서 발생한 문제였다. 이러한 점을 해결하기 위해서 디자인 패턴을 최대한 적용시켜서 하드코딩이 되지 않도록 하였다.

## 추진내용

### 기존 프로그램의 MRD

1. 기업사용자는 회의실에 대한 정보를 시스템에 등록할 수 있다.
2. 기업사용자는 예약한 회의실을 시스템에서 삭제할 수 있다.
3. 일반사용자는 시스템을 사용하여 예약 가능한 회의실을 검색할 수 있다.
4. 일반사용자는 회의실을 예약할 수 있다.
5. 일반사용자는 예약한 회의실을 취소할 수 있다.
6. 검색한 일자와 도시에서 가용한 회의실이 없는 경우 시스템은 차선책을 제시한다.
7. 시스템에 기업사용자를 등록한다.
8. 시스템에 일반사용자를 등록한다.
9. 사용자는 로그인하여 시스템을 사용할 수 있다.
10. 등록된 사용자만이 시스템을 사용할 수 있다.
11. 일반사용자는 권한을 가진다.
12. 기업사용자는 권한을 가진다.
13. 기업사용자만이 예약자 정보를 볼 수 있다.
14. 일반사용자는 선택한 회의실의 예약여부를 볼 수 있다.
15. 일반사용자는 예약한 회의실을 조회할 수 있다.
16. 기업사용자는 등록된 회의실을 조회할 수 있다.
17. 시스템은 화폐단위를 한국화폐단위(W)로 쓴다.

### 개선된 MRD

1. 기업사용자는 회의실에 대한 정보를 시스템에 등록할 수 있다.
2. 일반사용자는 시스템을 사용하여 예약 가능한 회의실을 검색할 수 있다.
3. 일반사용자는 회의실을 예약할 수 있다.
4. 시스템은 일반사용자가 검색한 일자와 도시에서 가용한 회의실이 없는 경우, 차선책을 제시한다.
5. 사용자는 시스템에 회원 등록할 수 있다.
6. 등록된 사용자만이 시스템을 사용할 수 있다
7. 사용자는 로그인하여 시스템을 사용할 수 있다.
8. 사용자는 로그아웃 할 수 있다.
9. 일반사용자는 선택한 회의실의 예약현황을 볼 수 있다.
10. 일반사용자는 예약한 회의실을 조회할 수 있다.
11. 기업사용자는 등록된 회의실을 조회할 수 있다.
12. 기업사용자는 등록된 회의실을 삭제할 수 있다.
13. 모든 사용자는 회의실 정보를 텍스트 파일로 다운로드 받을 수 있다.
14. 시스템은 화폐단위를 한국화폐단위(W)로 쓴다.

개선된 MRD로 만들어진 SRS
1. 기업사용자는 회의실에 대한 정보를 시스템에 등록할 수 있다.
<p>1.1 시스템은 회의실 이름, 회의실 위치, 수용가능 인원, 대여 비용의 관련 정보를 필수적으로 제공 받아야 한다.</p> <p>1.1.a 회의실 위치는 대한민국 내의 특별시, 광역시, 특별자치시, 시로 한정한다.</p> <p>1.1.b 수용가능 인원은 최소 두 명에서 최대 만 명까지 입력 받아야 한다.</p> <p>1.1.c 대여 비용은 최소 만 원으로 천 원 단위로 증가시킬 수 있다.</p> <p>1.1.d 하나의 조건이라도 충족되지 않으면 에러 메시지를 출력한다.</p> <p>1.1.e 시스템은 회의실 이름을 한 자 이상 스무 자 이하의 문자열 형태로 입력 받는다.</p> <p>1.2 시스템은 부대 시설 관련 정보를 문자열 형태로 부가적으로 제공받을 수 있다.</p> <p>1.2.a 시스템은 부대 시설 관련 정보를 최대 500자까지 입력 받을 수 있다.</p>
2. 일반사용자는 시스템을 사용하여 예약 가능한 회의실을 검색할 수 있다.
<p>2.1 시스템은 검색과 관련된 정보(위치, 날짜, 수용가능 인원)를 제공 받으면 이와 관련된 검색 결과를 보여줄 수 있다.</p> <p>2.1.a 회의실 위치는 대한민국 내의 특별시, 광역시, 특별자치시, 시 내에서 입력 받아야 한다.</p> <p>2.1.b 시스템은 수용가능 인원을 최소 두 명 이상 입력 받아야 한다.</p> <p>2.1.c 수용가능 인원은 사용자가 입력한 인원 이상일 경우 모두 출력한다.</p> <p>2.2 시스템은 대여 비용이 적은 순으로 결과를 출력한다.</p> <p>2.2.a 대여 비용이 같은 회의실이 존재할 경우 먼저 등록된 회의실을 우선으로 한다.</p> <p>2.3 시스템은 제공받지 않은 정보는 모든 범위로 간주하고 검색 결과를 보여준다.</p> <p>2.3.a 위치를 검색 받지 못한 경우 해당 날짜의 수용가능 인원 이상의 모든 회의실을 출력한다.</p> <p>2.3.b 날짜를 검색 받지 못한 경우 해당 위치의 수용가능 인원 이상의 모든 회의실을 출력한다.</p> <p>2.3.c 수용가능 인원을 검색 받지 못한 경우 해당 날짜와 위치의 모든 회의실을 출력한다.</p>
3. 일반사용자는 회의실을 예약할 수 있다.
<p>3.1 일반사용자는 원하는 회의실을 검색한 후, 출력된 회의실 중 하나를 선택해 이 기능을 실행할 수 있다.</p> <p>3.2 시스템은 사용자가 예약하고자 하는 날짜를 제공 받아야 한다.</p> <p>3.2.a 해당 날짜가 이미 예약되어 있는 경우 이미 예약된 날짜라는 메시지를 출력한다.</p> <p>3.3 시스템은 하나의 일반사용자에 대해 회의실을 최대 3개까지 예약 받을 수 있다.</p> <p>3.3.a 이미 3개의 회의실을 예약한 사용자가 이 기능을 실행할 경우 더 이상 예약할 수 없다는 메시지를 출력한다.</p>
4. 시스템은 일반사용자가 검색한 일자와 도시에서 가용한 회의실이 없는 경우, 차선책을 제시한다
<p>4.1 사용자가 입력한 수용 인원을 충족하는 회의실이 없는 경우에는 차선책을 제시하지 않는다.</p>

4.2 사용자는 해당 일자의 다른 도시의 회의실을 차선책으로 추천 받는다.
4.3 시스템은 해당 일자의 다른 도시가 없을 경우 차선책이 없다는 메시지를 출력한다.
5. 사용자는 시스템에 회원 등록할 수 있다.
<p>5.1 시스템은 기업사용자를 등록 받을 수 있다.</p> <p>5.1.a 기업사용자를 등록 받을 경우에 시스템은 기업명, 아이디, 패스워드, 전화번호, E-mail 을 필수적으로 제공 받아야 한다.</p> <p>5.2 시스템은 일반사용자를 등록 받을 수 있다.</p> <p>5.2.a 일반사용자를 등록 받을 경우에 시스템은 아이디, 패스워드, 전화번호, E-mail 을 필수적으로 제공 받아야 한다.</p> <p>5.3 시스템은 지정된 양식에 맞춰 사용자 정보를 입력 받아야 한다.</p> <p>5.3.a 시스템은 기업명을 한 자 이상 스무 자 이하의 문자열로 입력 받는다.</p> <p>5.3.b 시스템은 아이디를 다섯 자 이상 스무 자 이하의 문자열로 입력 받는다.</p> <p>5.3.c 시스템은 패스워드를 다섯 자 이상 스무 자 이하의 문자열로 입력 받는다.</p> <p>5.3.d 시스템은 전화번호를 스무 자 이하의 문자열로 입력 받는다.</p> <p>5.3.e 시스템은 E-mail을 마흔 자 이하의 문자열로 입력 받는다.</p> <p>5.4 시스템은 이미 존재하는 아이디를 등록 받을 수 없다.</p> <p>5.4.a 이미 존재하는 아이디를 입력 받았을 경우 오류 메시지를 출력한다.</p>
6. 등록된 사용자만이 시스템을 사용할 수 있다.
6.1 시스템은 등록되지 않은 사용자에게 회원가입 기능만을 제공한다.
7. 사용자는 로그인하여 시스템을 사용할 수 있다.
<p>7.1 시스템은 등록된 사용자에게만 로그인을 허용할 수 있다.</p> <p>7.2 시스템은 등록되지 않은 사용자가 로그인을 요청할 경우 오류 메시지를 출력한다.</p> <p>7.2.a 등록되지 않은 아이디를 입력 받았을 경우 등록되지 않은 사용자라는 메시지를 출력한다.</p> <p>7.2.b 등록된 아이디를 입력 받았지만 잘못된 비밀번호를 입력 받았을 경우 비밀번호가 잘못되었</p> <p>다는 메시지를 출력한다.</p> <p>7.3 시스템에 등록된 정보에 따라 기업사용자와 일반사용자로 분류한다.</p>
8. 사용자는 로그아웃 할 수 있다.
<p>8.1 사용자는 로그인한 후 어느 상태에서나 로그아웃 할 수 있다.</p> <p>8.1.a 사용자가 양식을 작성하는 도중 로그아웃을 요청한다면 작성하고 있던 양식의 정보는 저장되지 않는다.</p> <p>8.2 사용자가 로그아웃을 요청하면 시스템은 사용자와 연결을 끊는다.</p>
9. 일반사용자는 선택한 회의실의 예약현황을 볼 수 있다.
<p>9.1 일반사용자는 원하는 회의실을 검색한 후, 출력된 회의실 중 하나를 선택해 이 기능을 실행할 수 있다.</p> <p>9.2 시스템은 선택된 회의실의 예약가능 날짜, 이미 예약된 날짜를 출력한다.</p> <p>9.2.a 날짜 순으로 출력하되 이미 예약된 날짜는 붉은색으로 출력하고 이미 예약되었다는 문구를 덧붙인다.</p>

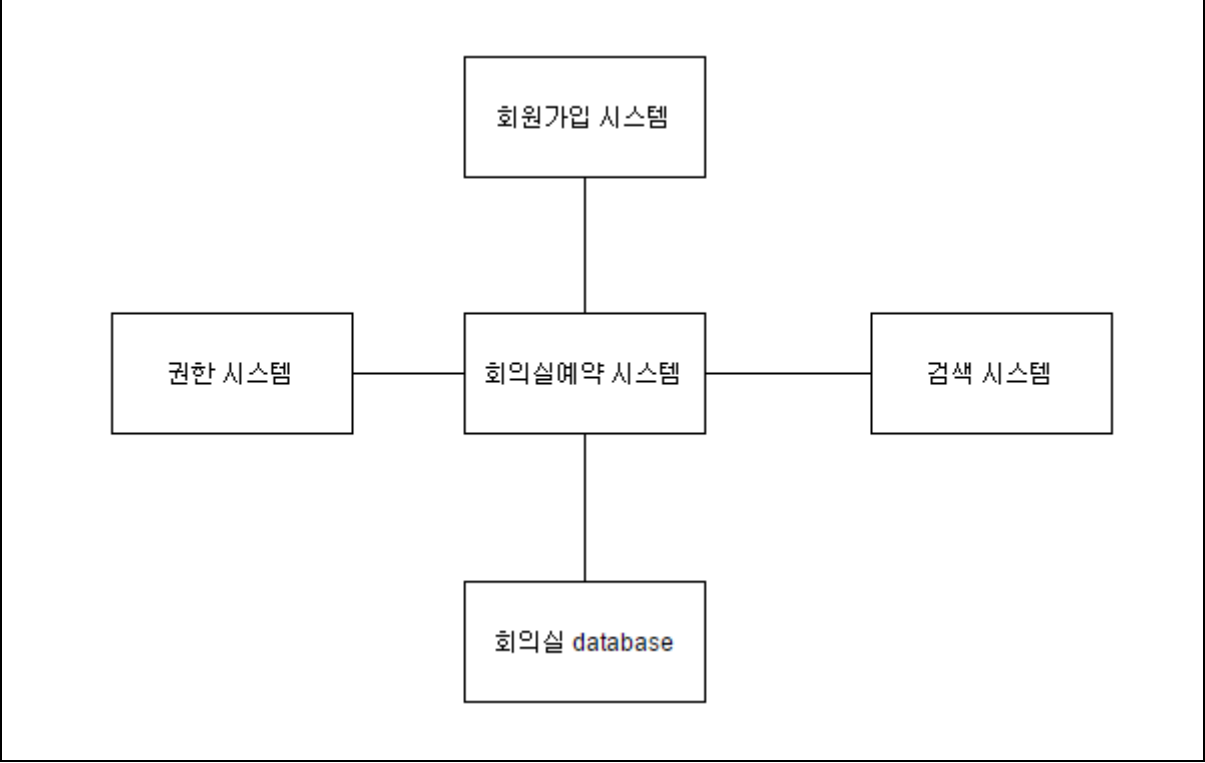
9.3 일반사용자는 이미 예약된 날짜에 예약한 사용자에게 관한 정보는 제공받을 수 없다.
10. 일반사용자는 예약한 회의실을 조회할 수 있다.
<p>10.1 시스템은 일반사용자가 예약한 회의실이 하나도 없더라도 이 기능을 실행할 수 있다.</p> <p>10.1.a 예약한 회의실이 하나도 없는 사용자가 이 기능을 실행할 경우 빈 리스트를 출력한다.</p> <p>10.2 사용자는 예약한 회의실을 날짜순으로 제공받는다.</p> <p>10.2.a 시스템은 가장 빠른 날짜가 맨 위로 오도록 리스트를 출력해야 한다.</p> <p>10.3 사용자는 회의실 이름, 위치, 등록된 기업명, 전화번호, 예약 날짜를 제공받는다.</p> <p>10.3.a 시스템은 회의실 이름, 위치, 등록된 기업명, 전화번호, 예약 날짜의 순으로 정보를 출력한다.</p>
11. 기업사용자는 등록된 회의실을 조회할 수 있다.
<p>11.1 기업사용자는 등록된 회의실을 순서대로 제공받는다.</p> <p>11.1.a 기업사용자는 먼저 등록된 회의실이 더 위로 오도록 정보를 제공받는다.</p> <p>11.2 기업사용자는 등록된 회의실을 예약한 일반사용자들의 정보(아이디, 전화번호, E-mail, 예약날짜)를 제공받는다.</p> <p>11.2.a 기업사용자는 예약한 사용자 정보를 예약날짜 순서대로 제공받는다.</p>
12. 기업사용자는 등록된 회의실을 삭제할 수 있다.
<p>12.1 기업사용자는 회의실 조회한 후, 출력된 회의실 중 하나를 선택해 이 기능을 실행할 수 있다.</p> <p>12.2 시스템은 기업사용자가 등록된 회의실이 적어도 1개 이상 존재할 때 이 기능을 실행할 수 있다.</p> <p>12.3 이 기능을 실행하면 시스템에서 해당 회의실 정보가 사라진다.</p>
13. 모든 사용자는 회의실 정보를 텍스트 파일로 다운로드 받을 수 있다.
<p>13.1 일반사용자는 원하는 회의실을 검색한 후, 출력된 회의실 중 하나를 선택해 이 기능을 실행할 수 있다.</p> <p>13.2 기업사용자는 회의실을 조회한 후, 출력된 회의실 중 하나를 선택해 이 기능을 실행할 수 있다.</p> <p>13.3 회의실 정보는 회의실 이름, 기업명, 회의실 위치, 수용가능인원, 대여 비용 순으로 출력된다.</p> <p>13.4 파일은 해당시스템의 최상위 폴더 내에 저장된다.</p>
14. 시스템은 화폐단위를 한국화폐단위(₩)로 쓴다.
14.1 시스템은 화폐단위가 출력되는 모든 곳에 단위를 명시한다.



Context Diagram & Architecture Design
Architecture Design
<b>1. MVC 패턴</b> <p>A. MVC가 현재 시스템에 적합한가? <b>NO.</b></p> <p>B. 적합하지 않은 이유는 무엇인가?</p> <p>i. MVC패턴은 비즈니스 영역의 상태 정보를 처리하는 Model과 비즈니스 영역의 대한 결과 화면을 담당하는 View와 사용자의 요청 값에 처리 및 흐름 제어를 담당하는 Controller로 구성된다. 이러한 패턴은 개별 사용자가 시스템에 접근하는 방식으로는 적합하다. <u>하지만, 전체적으로 보았을 때 회의실 예약 시스템은 여러 사람들이 동시다발적으로 사용할 수 있어야 하고 사용 할 데이터가 synchronize되어 입력되어야 하는데, MVC 패턴은 이러한 입력을 처리하는데 적절하지 않다.</u></p>
<b>2. Layered 아키텍처</b> <p>A. Layered 아키텍처가 현재 시스템에 적합한가? <b>NO.</b></p> <p>B. 적합하지 않은 이유는 무엇인가?</p> <p>i. Layered 아키텍처는 크기가 큰 프로젝트에서 레이어를 여러 가지로 나누어 플랫폼에 독립적이고, 보안을 중요시 할 때 유용한 방식이다. 하지만 회의실 예약 시스템은 기능이 수직적이 아니라 같은 레벨에 있는 사용자들이 시스템을 이용하는 등 <u>수평적인 기능들을 많이 가지고 있어 레이어를 여러 가지로 나누는 것이 불필요하다.</u></p>
<b>3. Repository 아키텍처</b> <p>A. Repository 아키텍처가 현재 시스템에 적합한가? <b>NO.</b></p> <p>B. 적합하지 않은 이유는 무엇인가?</p> <p>i. Repository 아키텍처는 모든 데이터가 중앙 repository에서 관리하고 데이터는 오직 repository를 통해서만 접근할 수 있는 데이터를 중요시하는 모델이다. 이러한 모델은 매우 많은 양의 데이터가 생성되어 오랜 기간 동안 시스템에 저장되어야 할 때 적합한 모델이다. 이 모델을 사용하면 각 요소가 독립적 이어서 각 요소가 다른 요소의 존재에 대해 알 필요가 없다. 하지만 우리가 개발하고 있는 회의실 예약 시스템은 <u>추가로 응용 프로그램을 붙일 일도 잘 없고, 데이터의 크기도 크지 않다. 또한 데이터 Evolution이 어렵기 때문에 적합하지 않다.</u></p>
<b>4. Client-Server 아키텍처 (적합한 모델)</b> <p>A. Client-Server 아키텍처가 현재 시스템에 적합한가? <b>YES.</b></p> <p>B. 적합한 이유는 무엇인가?</p> <p>i. Client-Server 아키텍처는 시스템의 각 기능들이 서비스로 구성되고, 각 서비스가 분리된 서버에서 전달되는 방식이다. Client들은 이러한 서비스들의 사용자이고, 이러한 서비스를 사용하기 위해 서버에 접속한다. 이 모델은 여러 장소</p>

에서 접속할 필요가 있는 공유되는 데이터베이스 데이터를 사용할 때 사용한다. 우리가 개발하는 시스템은 공유되는 데이터를 사용하므로 이러한 모델이 적합하다. 또한, 다중 사용자가 동시다발적으로 접속하기에 편리하고, 또한 데이터를 Synchronize하기에 적합하므로 좋다.

**Context Diagram**



Test Case
-----------

고유번호	TS - 1
Requirement	사용자는 로그인한 후 어느 상태에서나 로그아웃 할 수 있다.
Input	없음
Test Procedure	사용자가 양식을 작성하는 도중 로그아웃을 요청한다.
Expected output	작성하고 있던 양식의 정보는 저장되지 않고 로그아웃된다.

고유번호	TS - 2
Requirement	시스템은 회의실 이름을 한 자 이상 스무 자 이하의 문자열 형태로 입력 받는다.
Input	회의실 이름('ReserveConferenceRoom'), 회의실 위치('대구광역시'), 수용가능 인원(100), 대여 비용(100000), 부대 시설 관련 정보(NULL)
Test Procedure	기업 사용자는 21자의 회의실 이름을 시스템에 입력하고 나머지 정보는 입력 가능 범위 내의 input으로 입력한다.
Expected output	팝업 창을 통해 '회의실 이름은 한 자 이상 스무 자 이하의 문자열만 가능합니다' 라는 에러 메시지를 출력한다.

고유번호	TS - 3
Requirement	시스템은 부대 시설 관련 정보를 최대 500자까지 입력 받을 수 있다.
Input	회의실 이름('ConferenceRoom'), 회의실 위치('대구광역시'), 수용가능 인원(100), 대여 비용(100000), 부대 시설 관련 정보('a'를 300번 입력)
Test Procedure	기업사용자는 시스템에 부대 시설 관련 정보를 300자 입력하고 나머지 정보는 입력 가능 범위 내의 input을 입력한다.
Expected output	시스템에 부대 시설 관련 정보가 저장된다.

고유번호	TS - 4
Requirement	시스템은 검색과 관련된 정보(위치, 날짜, 수용가능 인원)를 제공 받으면 이와 관련된 검색 결과를 보여줄 수 있다.
Input	회의실 위치('울진군'), 날짜(2015.5.9), 수용가능 인원(100)
Test Procedure	'울진군'을 입력한 후 날짜와 수용가능 인원은 입력 가능 범위 내의 회의실을 검색한다.
Expected output	'회의실 위치는 대한민국 내의 특별자치시, 특별시, 광역시 내에서 입력해야 합니다' 라는 에러 메시지를 출력한 후 다시 검색 화면으로 되

	돌아간다..
--	--------

고유번호	TS - 5
Requirement	시스템은 대여 비용이 적은 순으로 결과를 출력한다.
Input	회의실 이름('A'), 수용가능 인원('100명'), 대여 비용('50만원') 회의실 이름('B'), 수용가능 인원('100명'), 대여 비용('60만원')
Test Procedure	회의실 위치와 수용가능 인원, 대여 비용을 입력한 후 검색을 한다.
Expected output	대여 비용이 낮은 A를 B보다 먼저 출력한다.

고유번호	TS - 6
Requirement	시스템은 검색과 관련된 정보(위치, 날짜, 수용가능 인원)를 제공 받으면 이와 관련된 검색 결과를 보여줄 수 있다.
Input	회의실 위치('대구광역시'), 수용가능 인원(100), 날짜(2015.4.10)
Test Procedure	Input의 조건에 맞는 회의실을 리스트에 추가해 놓은 상태에서 회의실 위치와 수용가능 인원, 대여 비용을 입력한 후 검색을 한다.
Expected output	2015년 4월 10일에 대구 광역시에서 100명을 수용할 수 있는 모든 회의실을 보여준다.

고유번호	TS - 7
Requirement	시스템이 일반사용자로부터 날짜를 검색 받지 못한 경우 해당 위치의 수용가능 인원 이상의 모든 회의실을 출력한다.
Input	회의실 위치('대구광역시'), 수용인원(100)
Test Procedure	Input의 조건에 맞는 회의실을 리스트에 추가해 놓은 상태에서 회의실 이름과 위치와 대여 비용을 입력한 후 검색을 한다.
Expected output	대구광역시의 100명 이상 수용 가능한 모든 회의실을 출력한다.

고유번호	TS - 8
Requirement	시스템이 일반사용자로부터 수용가능 인원을 검색 받지 못한 경우 해당 날짜와 위치의 모든 회의실을 출력한다
Input	회의실 위치('대구광역시'), 날짜(2015.4.10)
Test Procedure	Input의 조건에 맞는 회의실을 리스트에 추가해 놓은 상태에서 회의실 이름과 위치와 대여 비용을 입력한 후 검색을 한다.
Expected output	2015년 4월 10일에 대구광역시에 위치한 모든 회의실을 출력한다..

고유번호	TS - 9
Requirement	시스템은 하나의 일반사용자에 대해 회의실을 최대 3개까지 예약 받을 수 있다.
Input	4개의 회의실 예약
Test Procedure	이미 3개의 회의실을 예약한 사용자가 하나의 회의실을 더 예약한다.
Expected output	팝업 창을 통해 '최대 예약 가능한 회의실을 3개입니다' 라는 메시지를 출력한다.

고유번호	TS - 10
Requirement	시스템은 기업사용자를 등록 받을 수 있다.
Input	기업명('GGulCompany'), 아이디('GGulid'), 비밀번호('GGulpasswd'), 전화번호('053-950-6659'), E-mail('GGulcompany@GGul.co.kr')
Test Procedure	기업 사용자로 등록을 하고자 하는 유저는 기업명(현재 유저 리스트에 같은 기업명이 없도록 한다.), 아이디(현재 유저 리스트에 같은 아이디가 없도록 한다.), 비밀번호, 전화번호, E-mail을 입력한다.
Expected output	기업명, 아이디, 비밀번호, 전화번호, E-mail이 시스템에 저장되며 기업 사용자로 등록된다.

고유번호	TS - 11
Requirement	시스템은 일반사용자를 등록 받을 수 있다.
Input	아이디('GGulid'), 비밀번호('GGulpasswd'), 전화번호('053-950-6659'), E-mail('GGulcompany@GGul.co.kr')
Test Procedure	일반 사용자로 등록을 하고자 하는 유저는 아이디(현재 유저 리스트에 같은 아이디가 없도록 한다.), 비밀번호, 전화번호, E-mail을 입력한다.
Expected output	아이디, 비밀번호, 전화번호, E-mail이 시스템에 저장되며 일반사용자로 등록된다.

고유번호	TS - 12
Requirement	시스템은 이미 존재하는 아이디를 등록 받을 수 없다.
Input	이미 등록된 아이디('GGulid') 비밀번호('GGulpasswd'), 전화번호('053-950-6659'), E-mail('GGulcompany@GGul.co.kr')
Test Procedure	회원가입 시 이미 시스템에 등록된 아이디('GGulid')를 입력한다.

Expected output	시스템은 '이미 등록된 아이디입니다'라는 오류 메시지를 출력한다.
-----------------	--------------------------------------

고유번호	TS - 13
Requirement	시스템은 등록되지 않은 사용자에게 회원가입 기능만을 제공한다.
Input	회원가입을 하지 않은 비등록 사용자
Test Procedure	비등록 사용자가 회의실 예약 프로그램을 사용하기 위해 시스템에 접속한다.
Expected output	시스템은 비등록 사용자에게 회원가입 기능만을 제공한다.

고유번호	TS - 14
Requirement	등록되지 않은 아이디를 입력 받았을 경우 등록되지 않은 사용자라는 메시지를 출력한다.
Input	ID('GG'), PASSOWRD('GG123')
Test Procedure	비등록 사용자가 로그인을 하기 위해 시스템에 등록되지 않은 ID와 PASSWORD를 입력한 후 로그인 버튼을 누른다.
Expected output	팝업 창을 통해 '등록되지 않은 사용자' 라는 오류 메시지를 출력한다.

고유번호	TS - 15
Requirement	등록된 아이디를 입력 받았지만 잘못된 비밀번호를 입력 받았을 경우 비밀번호가 잘못 되었다는 메시지를 출력한다.
Input	ID('GG'), Password('GG456')
Test Procedure	시스템에 ID가 'GG'이고 비밀번호가 'GG123'인 회원이 존재하는 상태에서 비등록 사용자가 로그인을 하기 위해 시스템에 ID와 PASSWORD를 입력한 후 로그인 버튼을 누른다.
Expected output	팝업 창을 통해 '비밀번호가 잘못되었습니다' 라는 오류 메시지를 출력한다.

고유번호	TS - 16
Requirement	일반사용자는 원하는 회의실을 검색한 후, 출력된 회의실 중 하나를 선택해 회의실 현황을 볼 수 있다.
Input	회의실 검색 후 출력된 회의실 중 맨 위의 회의실
Test Procedure	회의실 목록 중 하나의 CheckBox 버튼을 선택해서 회의실 예약 현황을 실행한다.

Expected output	날짜 순으로 회의실의 예약 현황을 출력하며 이미 예약된 날짜는 붉은색으로 이미 예약되었다는 문구를 출력한다.
-----------------	--

고유번호	TS - 17
Requirement	시스템은 일반사용자가 예약한 회의실이 하나도 없더라도 예약한 회의실을 조회할 수 있다.
Input	예약한 회의실이 없음
Test Procedure	일반 사용자가 예약한 회의실을 조회한다.
Expected output	시스템은 빈 리스트를 출력한다.

고유번호	TS - 18
Requirement	사용자는 예약한 회의실을 날짜순으로 제공받는다.
Input	회의실 이름('A'), 회의실 위치('대구광역시'), 수용가능 인원(100), 대여비용(100000), 예약 날짜 (2015.5.8), 부대 시설 관련 정보(NULL) 회의실 이름('B'), 회의실 위치('대구광역시'), 수용가능 인원(100), 대여비용(100000), 예약 날짜 (2015.5.4), 부대 시설 관련 정보(NULL)
Test Procedure	일반 사용자가 예약한 회의실을 조회한다.
Expected output	시스템은 예약 날짜가 먼저인 B를 먼저 출력하고 A를 출력한다.

고유번호	TS - 19
Requirement	사용자는 회의실 이름, 위치, 등록된 기업명, 전화번호, 예약 날짜를 제공받는다.
Input	회의실 이름('A'), 회의실 위치('대구광역시'), 등록기업('SDcompany') 수용가능 인원(100), 대여 비용(100000), 예약 날짜 (2015.5.8), 부대 시설 관련 정보(NULL) 회의실 이름('B'), 회의실 위치('대구광역시'), 등록기업('RCcompany'), 수용가능 인원(100), 대여 비용(100000), 예약 날짜 (2015.5.4), 부대 시설 관련 정보(NULL)
Test Procedure	일반 사용자가 예약한 회의실을 조회한다.
Expected output	시스템은 회의실 이름, 위치, 등록된 기업명, 전화번호, 예약 날짜의 순으로 정보를 출력한다.

고유번호	TS - 20
Requirement	기업사용자는 회의실 조회 시 회의실을 등록한 순서대로 제공받는다.

Input	회의실 이름('A'), 회의실 위치('대구광역시'), 수용가능 인원(100), 대여 비용(100000), 예약 날짜 (2015.5.8), 부대 시설 관련 정보(NULL) 회의실 이름('B'), 회의실 위치('대구광역시'), 수용가능 인원(100), 대여 비용(100000), 예약 날짜 (2015.5.4), 부대 시설 관련 정보(NULL)
Test Procedure	기업사용자는 등록한 회의실을 조회한다.
Expected output	시스템은 A를 먼저 등록 받았으므로 A를 먼저 출력하고 B를 출력한다.

고유번호	TS - 21
Requirement	기업사용자는 등록한 회의실을 예약한 일반사용자들의 정보(아이디, 전화번호, E-mail, 예약날짜)를 제공받는다.
Input	ID('A'), 전화번호('123-456-7890'), E-mail('abc@google.com') ID('B'), 전화번호('123-789-7890'), E-mail('bdw@google.com')
Test Procedure	기업사용자는 등록한 회의실을 조회한 후 예약한 일반사용자들의 정보를 조회한다..
Expected output	기업 사용자는 예약한 사용자 정보(아이디, 전화번호, E-mail, 예약날짜)를 순서대로 제공받는다.

고유번호	TS - 22
Requirement	기업사용자는 회의실 조회한 후, 출력된 회의실 중 하나를 선택해 회의실을 삭제할 수 있다.
Input	회의실 이름('A'), 회의실 위치('대구광역시'), 수용가능 인원(100), 대여 비용(100000), 예약 날짜 (2015.5.8), 부대 시설 관련 정보(NULL) 회의실 이름('B'), 회의실 위치('대구광역시'), 수용가능 인원(100), 대여 비용(100000), 예약 날짜 (2015.5.4), 부대 시설 관련 정보(NULL) 중 A.
Test Procedure	기업 사용자는 회의실 하나를 선택해서 삭제 버튼을 누른다.
Expected output	선택된 회의실(A)은 등록이 취소되며 리스트에서 사라진다.

고유번호	TS - 23
Requirement	시스템은 기업사용자가 등록한 회의실이 적어도 1개 이상 존재할 때 회의실 삭제 기능을 실행할 수 있다.
Input	등록된 회의실이 없음.
Test Procedure	등록된 회의실이 없는 상태에서 회의실 삭제 버튼을 누른다.

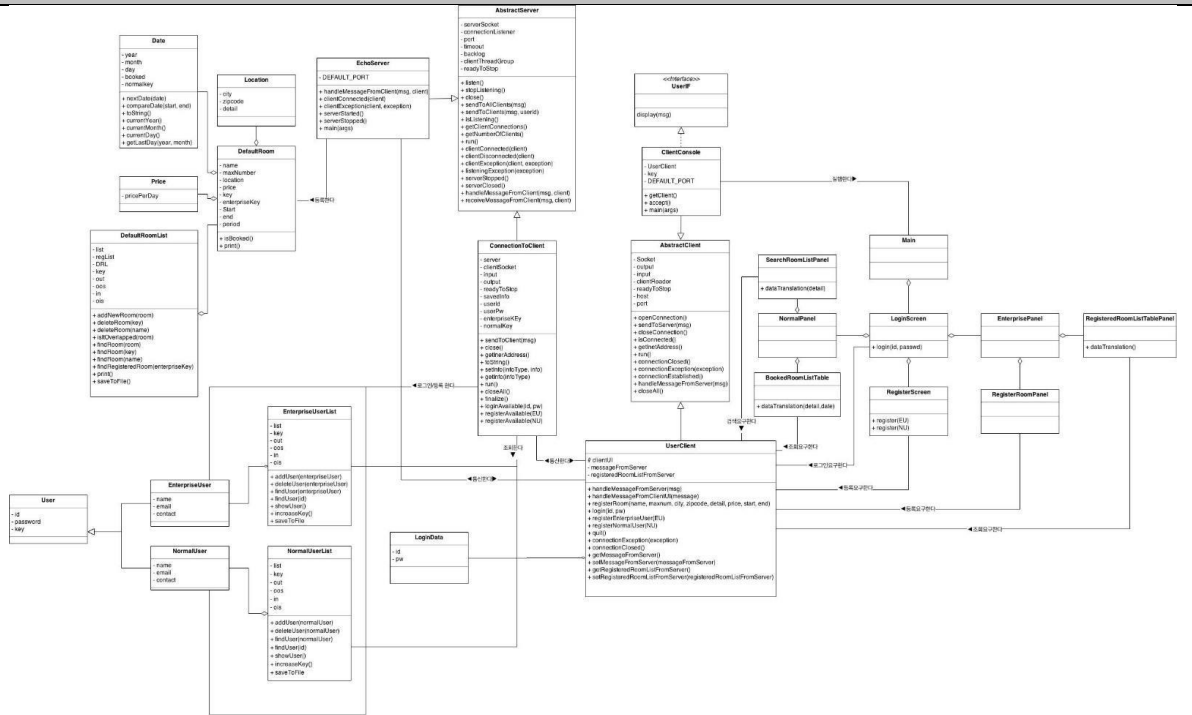


Expected output	“등록된 회의실이 없다”는 오류 메시지를 출력한다.
-----------------	------------------------------

고유번호	TS - 24
Requirement	일반사용자는 원하는 회의실을 검색한 후, 출력된 회의실 중 하나를 선택해 회의실 정보를 텍스트 파일로 다운받을 수 있다.
Input	출력된 회의실 중 첫 번째 회의실 선택
Test Procedure	일반 사용자는 회의실을 선택하지 않고 텍스트 파일로 다운받는다.
Expected output	텍스트 파일이 해당 시스템 최상위 폴더에 저장되며 파일 안에는 회의실 정보가 입력되어 있지 않는다.

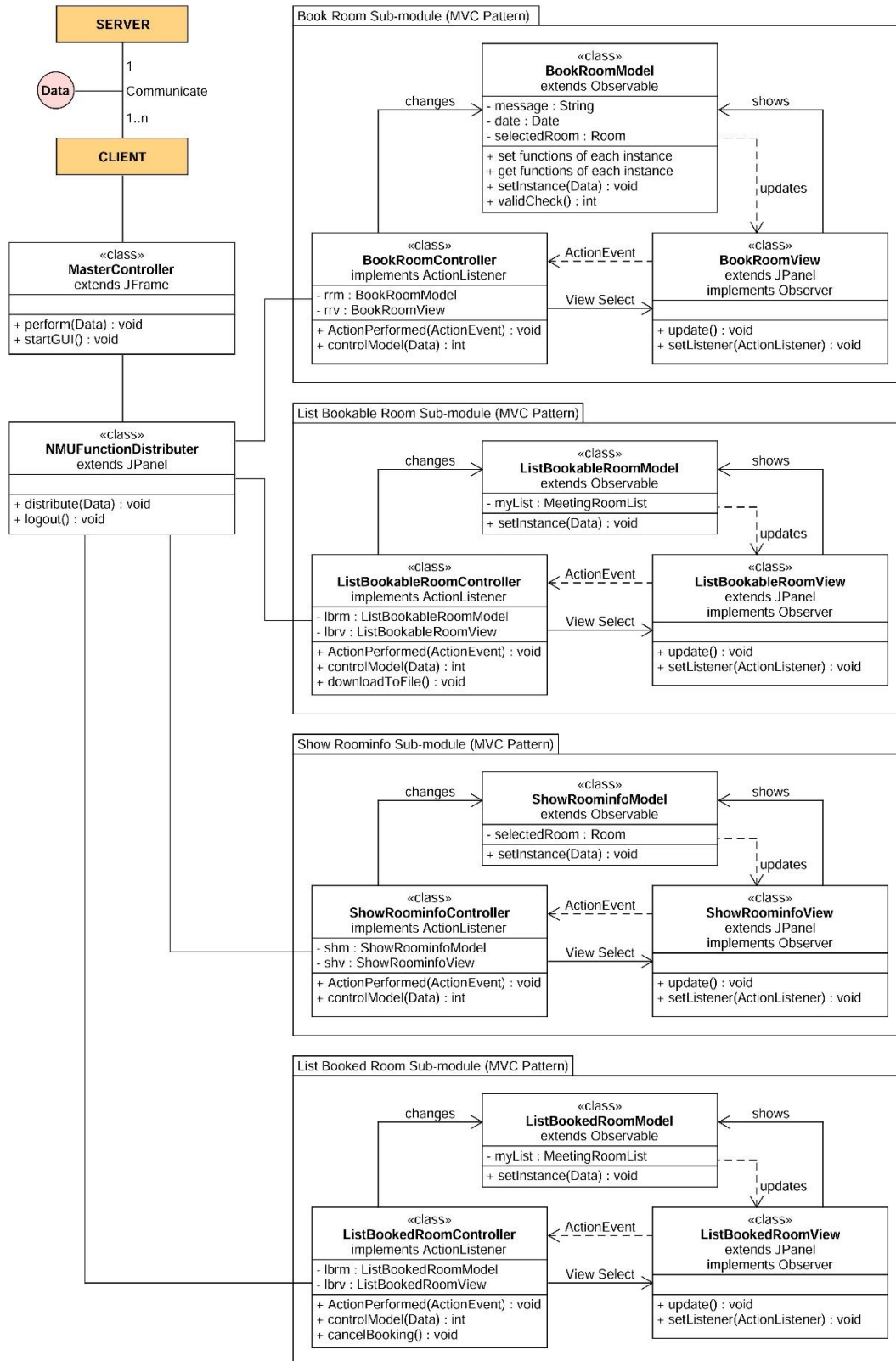
고유번호	TS - 25
Requirement	기업사용자는 회의실을 조회한 후, 출력된 회의실 중 하나를 선택해 회의실 정보를 텍스트 파일로 다운받을 수 있다.
Input	등록된 회의실 중 첫 번째 회의실 선택
Test Procedure	기업 사용자는 회의실을 선택하여 텍스트 파일로 다운받는다.
Expected output	텍스트 파일이 해당 시스템 최상위 폴더에 저장되며 선택한 회의실 정보가 파일 안에 출력된다.

## 기존 클래스 다이어그램

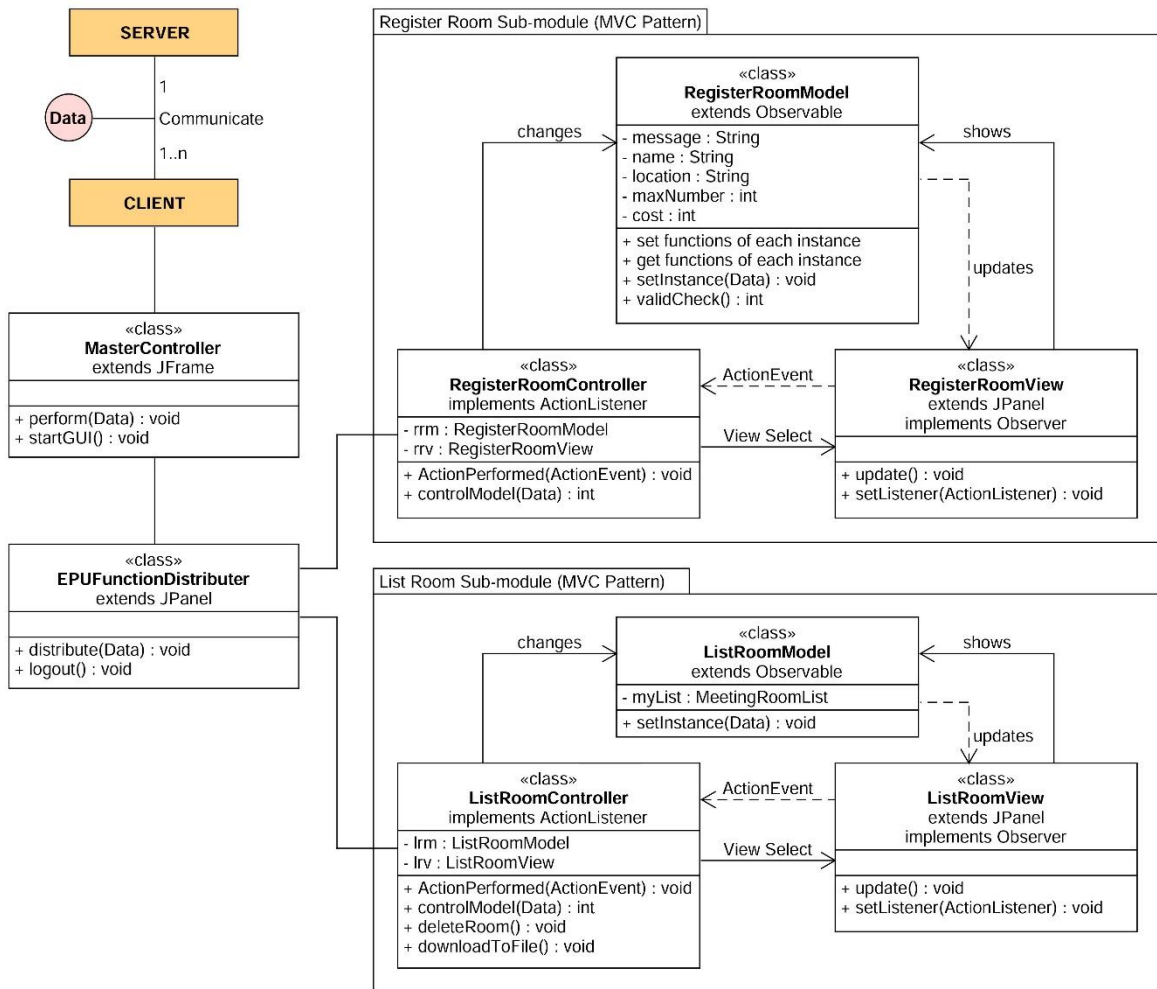


## 개선된 다이어그램

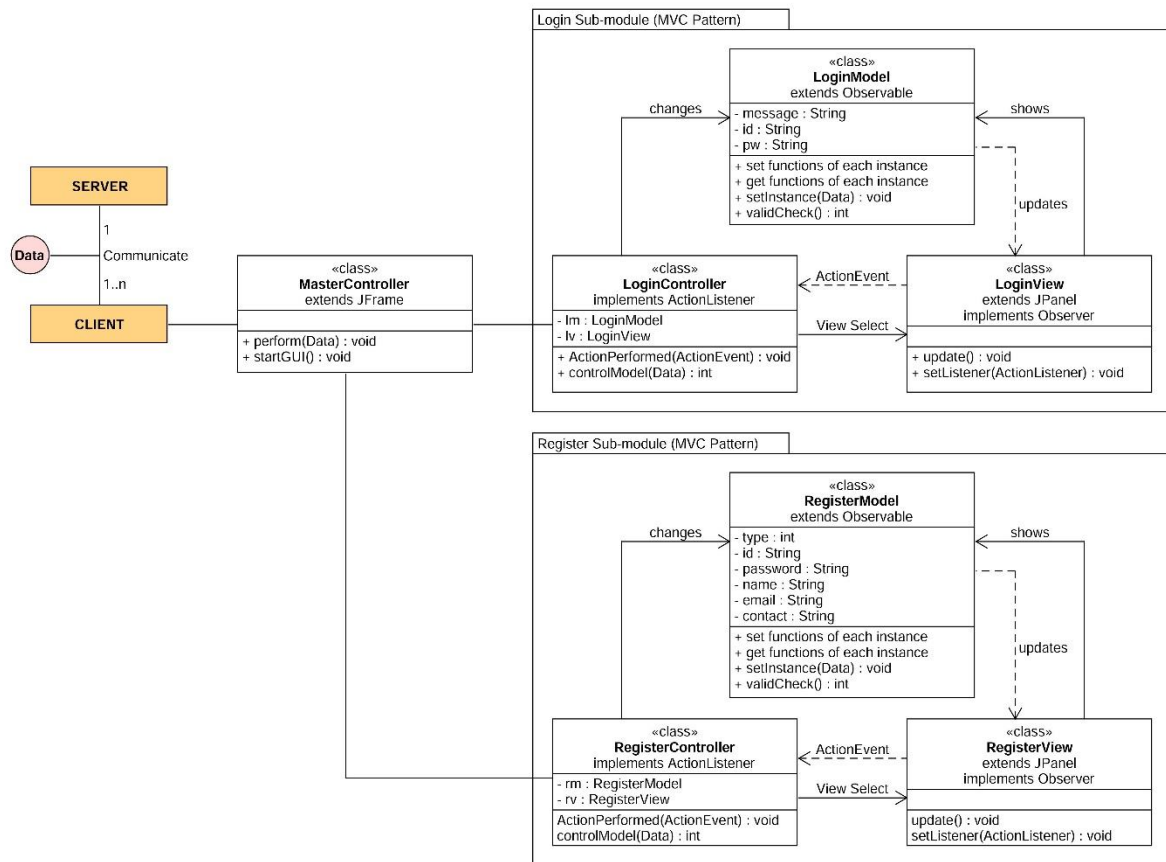
### 1. Normal User Class Diagram



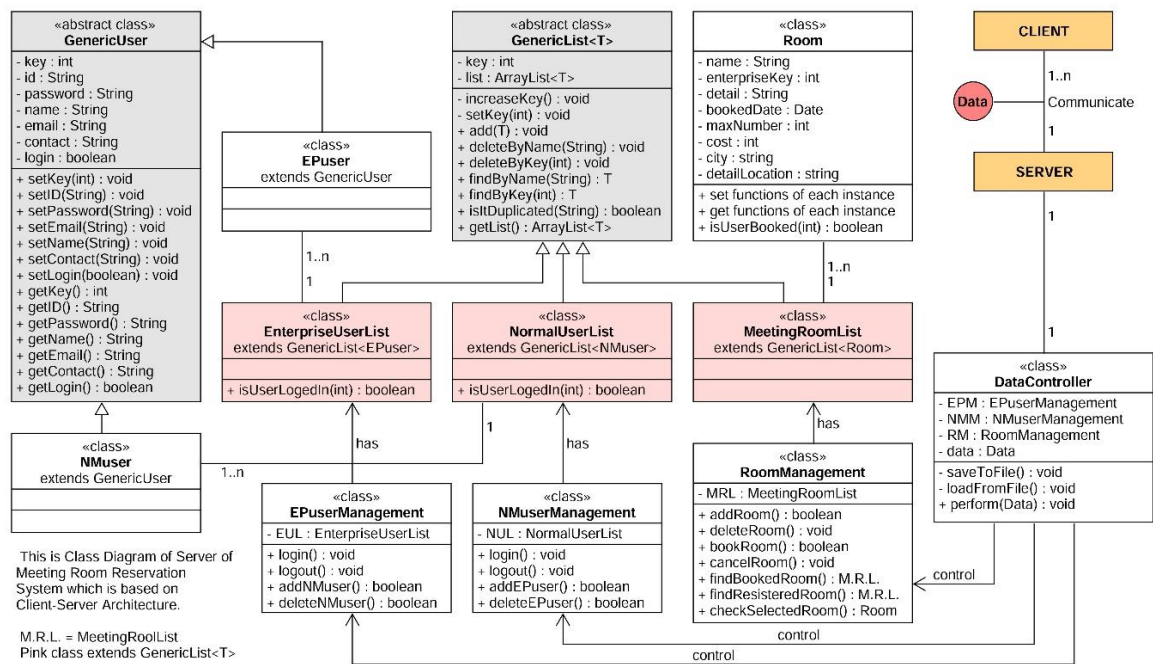
## 2. Enterprise User Class Diagram



### 3. Client Register, Login Class Diagram



### 4. Server Class Diagram



transmissionData 값
0. 회원가입 요청
1. 회원가입 승낙
2. 회원가입 취소 요청
3. 회원가입 취소 승낙
4. 일반사용자 회원정보 전송
5. 기업사용자 회원정보 전송
6. 중복
7. 성공
10. 일반사용자 로그인
11. 기업사용자 로그인
12. 일반사용자 로그인 성공
13. 기업사용자 로그인 성공
14. 로그인 실패
20. 회의실 등록
21. 중복
22. 성공
30. 등록한 회의실 목록
31. 성공
40. 회의실 삭제
41. 성공
50. 회의실 예약
51. 성공
52. 예약 가능 수 초과로 실패
53. 이미 예약된 날짜라서 실패
60. 예약 가능한 회의실 목록
61. 성공
70. 회의실 정보 조회
71. 성공
80. 예약한 회의실 목록
81. 성공
82. 예약 날짜 확인 요청
83. 성공
90. 예약 취소
91. 성공
100. 로그 아웃 요청
101. 성공

## 성과

프로젝트 팀들 중 Lines of code 당 issues의 비율이 가장 낮다.

QG	NAME ▲	LAST ANALYSIS	LOC	ISSUES
★	SE2015-89s	2015/06/13	3,695 ↗	830 ↗
★	SE2015-best	2015/06/16	1,575	442
★	SE2015-electronics	04:10	637	262
★	SE2015-haribo	2015/06/03	4,842	1,176
★	SE2015-k2nm	2015/06/06	1,322	432
★	SE2015-killer	2015/06/12	3,913 ↗	1,235 ↗
★	SE2015-maybe	2015/06/12	3,553 ↗	477 ↘
★	SE2015-me	2015/06/14	3,209 ↗	498 ↘

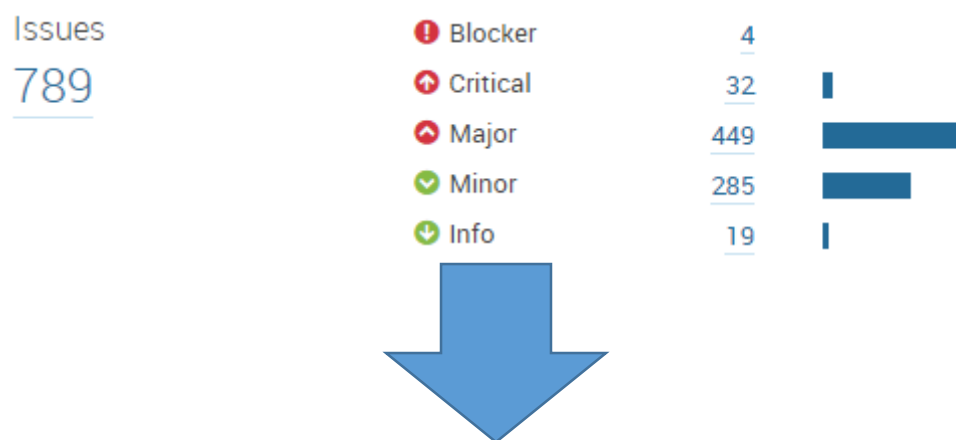
SQALE Rating이 B에서 A로 상승하였다.



Technical Debt Ratio가 10.1%에서 3.3%로 감소하였다.



Issues의 수가 789에서 477로 감소하였다.



Issues

477 ↘

🚫 Blocker 2

🚫 Critical 18 ↘

🚫 Major 230

✅ Minor 202 ↘

✅ Info 25 ↗

Duplications가 6.8%에서 5.6%로 감소하였다.

Duplications

6.8%

Lines

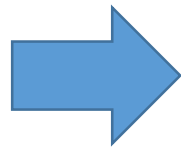
336

Blocks

17

Files

8



Duplications

5.6%

Lines

311 ↘

Blocks

19

Files

13 ↗



## 고찰

### 1. 완성도

기존 시스템은 기능이 원하는 대로 돌아가지 않는 오류와 구현하지 못한 기능들이 몇 가지 있었지만, Re-engineering을 진행하면서 디자인에 공을 들여서 디자인의 퀄리티가 많이 상승하여 에러를 찾기가 수월하여 에러를 대부분 해결하였으며, 계획한 기능등을 모두 구현하였습니다.

### 2. 질적 수준

이번 프로젝트를 진행하면서 디자인에 매우 신경을 썼습니다. 그리고 구현에 있어서 디자인 패턴에 대해 많이 알아보고 이를 적용시키려고 했습니다. 그래서 iterator, observer, MVC 등 여러가지 패턴을 적용하여 시스템을 올바르게 개발하려고 노력하였습니다. 그의 결과로 시스템을 평가하는 SonarQube에서 LOC당 Issues의 비율이 프로젝트 팀들 중 제일 우수하게 나올 수 있었고, SQALE Rating이 B에서 A로 증가하였으며, Technical Debt Ratio도 10.1%에서 3.3%로 줄였습니다. Issues도 789에서 477으로 대폭 감소하였으며, Duplications는 6.8%에서 5.6%로 감소하여 질적 수준이 대폭 향상된 것을 눈으로 확인할 수 있습니다.

### 3. 어려운 점과 극복방법

구현 중에 디자인했던대로 구현해놓았지만 나중에 그것 때문에 문제가 생긴 부분이 있었다. 한 사용자가 같은 회의실을 다른 날짜에 예약한 후 조회/예약취소에서 문제가 있었지만 새로운 int값을 추가함으로써 극복하였습니다. 소켓 통신상에서 발생하는 에러들 중 원인 파악이 힘든 에러가 있었지만 설계가 잘 되어 있어서 구조를 잘 알고 있으니 금방 찾아낼 수 있었습니다. 팀으로 작업하면서 git으로인해 상당부분 고생을 하였습니다. 혼자서 git을 쓸 때와 팀으로 git으로 쓸 때가 너무 달랐습니다. 그래서 git에 대해서 좀 더 공부하는 시간을 가진 뒤 그 이후로는 git으로 인한 문제를 대폭 줄이고 팀 개발의 장점을 최대한 끌어올렸습니다. 그리고 무엇보다 프로젝트를 진행하면서 팀원들이 외적인 요인으로 인해 심적으로 많이 힘들었습니다. 이번학기에 팀에서 세명이나 집에서 초상이 나기도 하였고, 제일 힘들다고 하는 3학년 1학기였지만, 결국 프로젝트는 잘 해결되고 팀원들간의 신뢰도도 더욱 상승하였습니다. 여느 팀 프로젝트처럼 끝나면 서로 팀원들간의 불신만 쌓이고 인간관계만 안 좋아졌지만 이번은 다음에 다시 작업하고 싶은 생각이 드는 팀이었습니다.