

Term Project



과 목 명	데이터베이스시스템
담 당 교 수	노서영
학 과	소프트웨어학과
학 년	3
학 번	2019038024
이 름	이동민

Term Project 목표

다음에 열거된 요구사항(requirement)를 충분히 숙지하여 Term Project 보고서를 제출하시오.

1. Database로 문제를 해결하기 위한 Mini-World Problem을 찾는다. (예: 도서관리)
2. 문제를 풀기위한 요구사항(requirements)들을 정리한다.
3. ER (Entity-Relationship) Diagram을 작성한다.
4. ER-Relational Mapping을 통해 Relation Schema를 생성한다.
5. Relation Schema에 대해 Normalization을 수행 하고 최소 3NF(3rd Normal Form) 형태가 되도록 한다. 1NF, 2NF로 Denormalization을 해야할 경우, 그 이유를 기술.
6. MySQL DBMS Database를 생성 한다.
7. 기본환경은 가상머신에 CentOS + MySQL이 설치되어야 한다.
8. 프로그램을 통해 데이터베이스에 접속하여 사용자가 요청을 수행하도록 한다. (예: Menu 기반)
9. 프로그램은 Network을 통해 데이터베이스에 접속해야 한다.
 - (a) Java, C#, C++, Python등 사용
 - (b) JDBC, ODBC등 프로그래밍 인터페이스 사용
 - (c) Web을 통한 접근 가능 (PHP)
10. 구현의 복잡도와 난이도에 따라 점수 차등 부여 함.
11. Term Project는 개별 프로젝트이며, 개인이 직접 프로그램을 작성해야 함.
12. 외부에서 검색된 프로젝트 사용 시 0 점 처리됨.
13. 프로젝트간 유사도를 검사하여, 중복도가 높은 프로그램들은 모두 0점 처리 됨.
14. 필요시 프로젝트 결과물은 Demonstration 해야할 수 있음.
15. 프로젝트 소스코드는 GitHub(<https://github.com/>)를 통해 관리되어야 함. 평가자가 Log기록 등을 검사함. (GitHub으로 관리되지 않는 프로젝트, 로그기록이 부실한 프로젝트 -50점)
16. eCampus(<https://ecampus.cbnu.ac.kr/>)에 TermProject-학번-이름.pdf 파일을 제출할것. GitHub URL 명시 할 것.

1. Database로 문제를 해결하기 위한 Mini-World Problem.

사용자가 원룸을 재계약시 회사/부동산 업체와의 계약으로 인해 생기는 불필요한 비용과 번거로움을 피하고자 함.
장기적으로 주거공간을 이용하는 데 있어 월세 납부가 요구되고, 까먹기 일상인데 월세 납부의 편의성을 증진 시키고자 함.

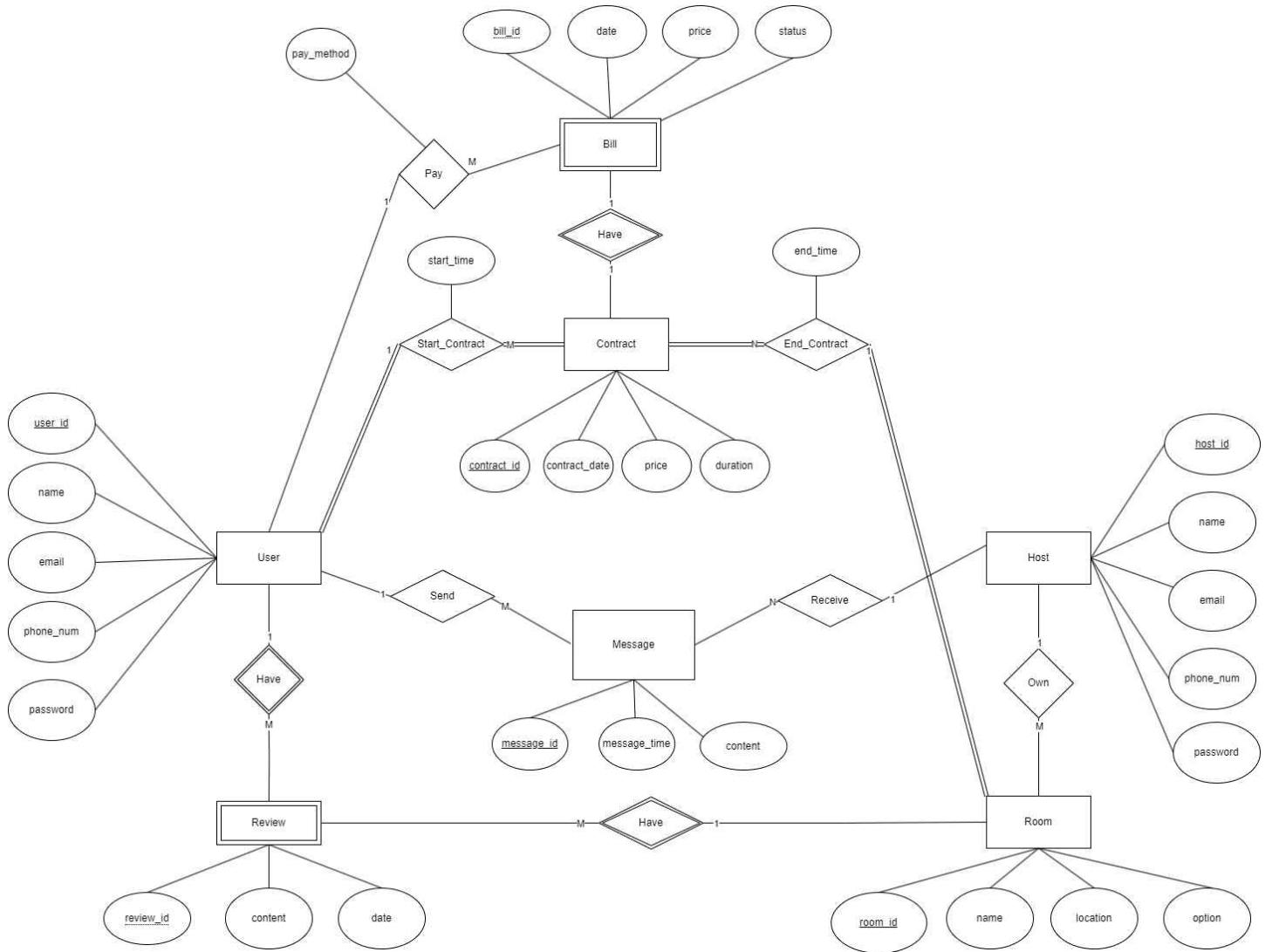
호텔이나 숙박업소와 같은 경우에는 후기가 많이 존재하나 원룸에는 후기가 존재하지 않아 원룸에도 후기가 있는 서비스를 만들고자 함.

후기를 보고 새로운 사용자가 회사/부동산 업체와의 계약 없이 전화로 이루어 질 수 있는 환경을 구축하고자 함.

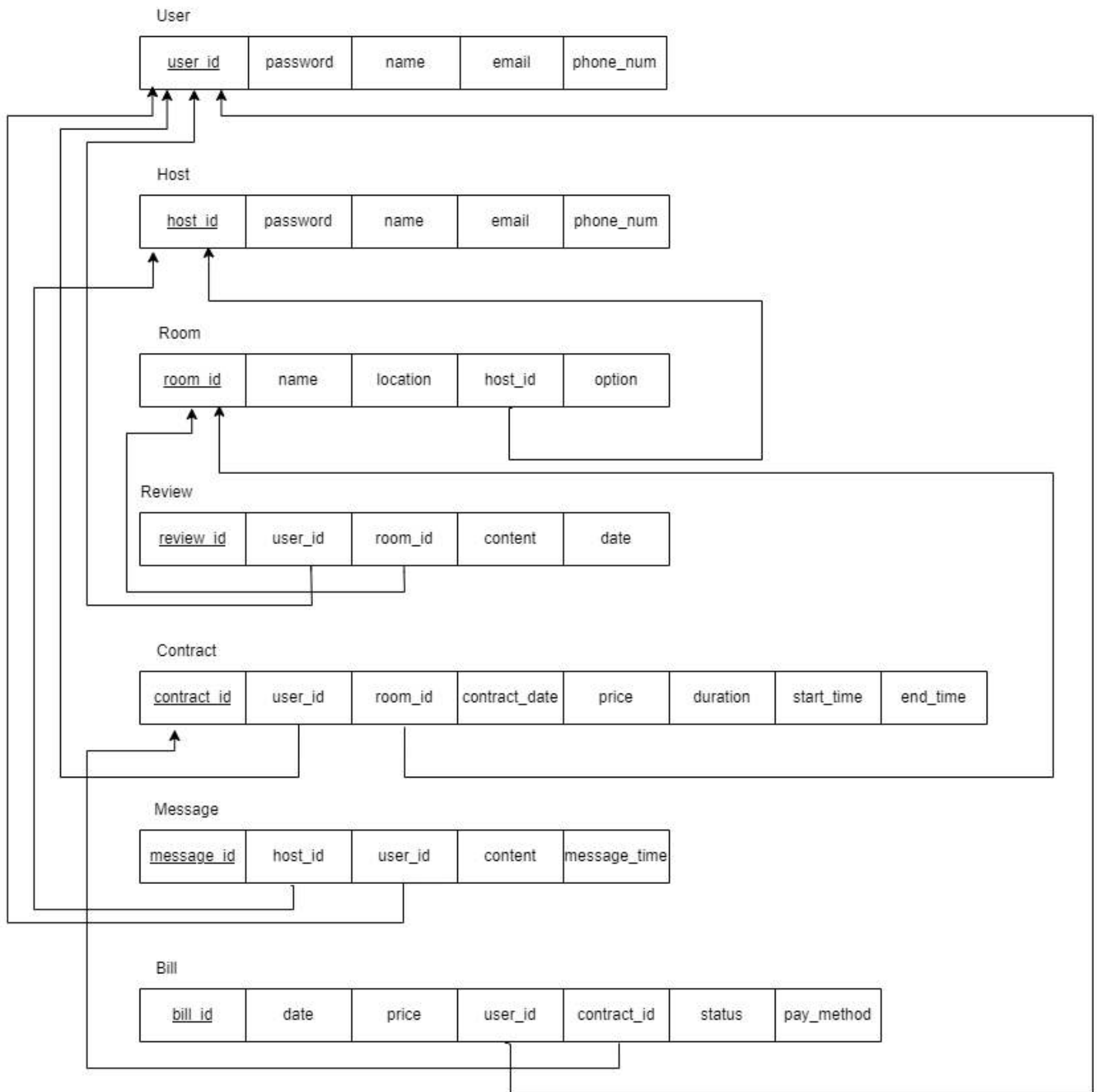
2. 문제를 풀기위한 요구사항(requirements).

1. 사용자는 아이디, 이름, 이메일, 휴대폰 번호, 비밀번호가 있어야 한다.
2. 사용자는 방을 계약 할 수 있고 계약시 계약에 대한 날짜, 가격, 기간 등에 대한 정보가 포함한다, 계약이 완료 되었을 경우에는 해당 계약에 대한 청구서를 만든다.
3. 계약을 통해 만들어진 청구서는 사용자가 청구서를 조회할 수 있고, 청구서에는 납부일자와 가격 정보, 납입상태를 포함한다.
4. 사용자는 요금 청구 시 청구서를 통해 자신이 지불 했던 방법에 대해 저장한다.
5. 사용자는 호스트에게 메시지를 줄 수 있으며, 호스트가 받은 메시지는 시간과 내용 정보를 포함한다.
6. 사용자는 자신이 거주했던 방에 대한 리뷰를 남길 수 있다, 해당 방에 대한 리뷰는 방에 종속된다.
7. 호스트는 아이디, 이름, 이메일, 휴대폰 번호, 비밀번호가 있어야 한다.
8. 호스트는 자신이 보유한 방에 대한 정보를 조회 할 수 있다, 방에는 방의 이름과 위치 정보를 포함한다.

3. ER (Entity-Relationship) Diagram.



4. ER-Relational Mapping을 통한 Relation Schema



5. Relation Schema에 대해 Normalization을 수행 하고 최소 3NF(3rd Normal Form) 형태가 되도록 한다. 1NF, 2NF로 Denormalization을 해야할 경우, 그 이유를 기술.

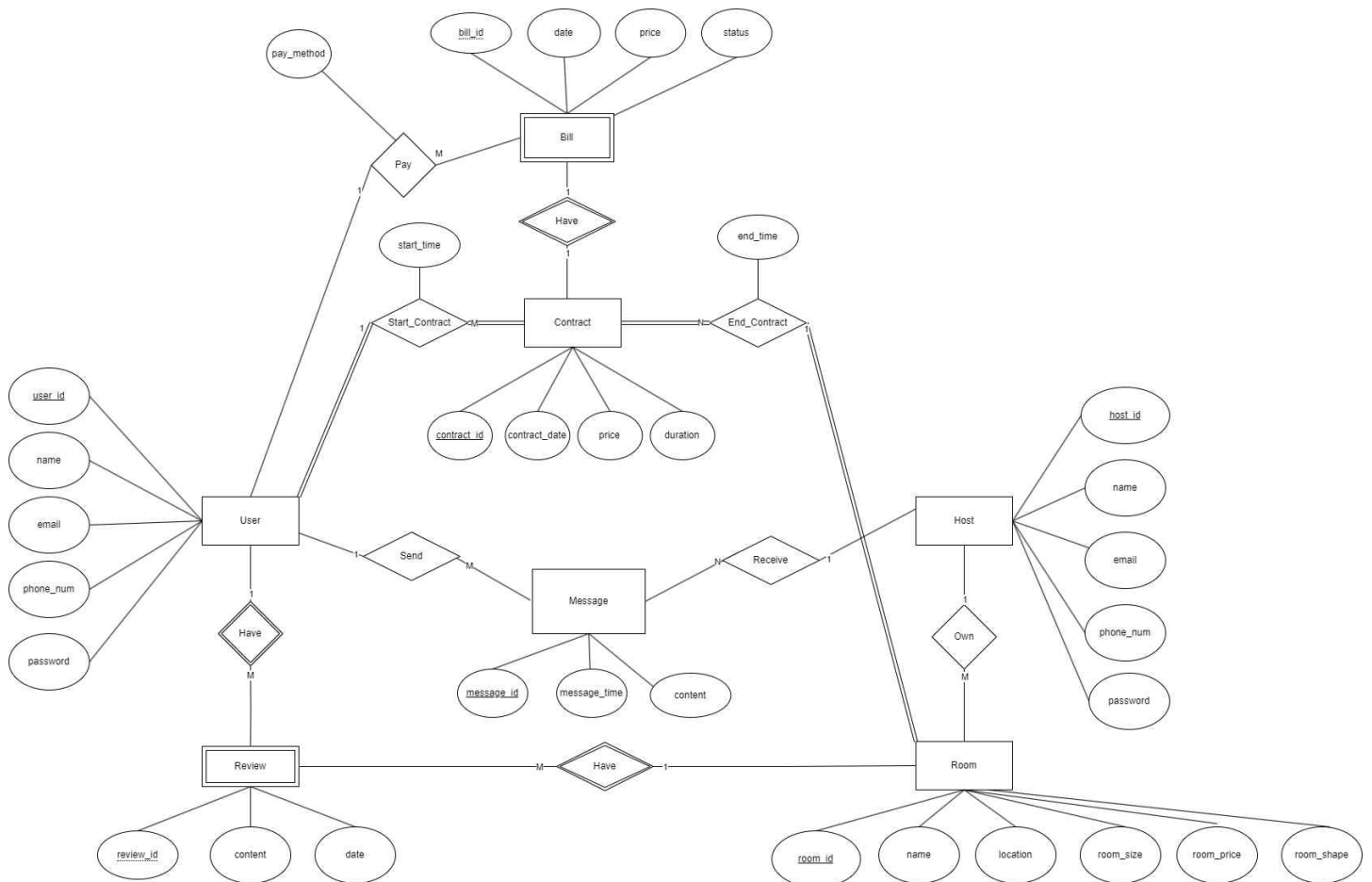
etc) 처음에 설계를 할 때 have의 관계를 고려하고 설계하여 나머지 2NF, 3NF는 만족하게 됨.

다른 건 문제가 없으나 Room의 Option은 다중값을 가짐

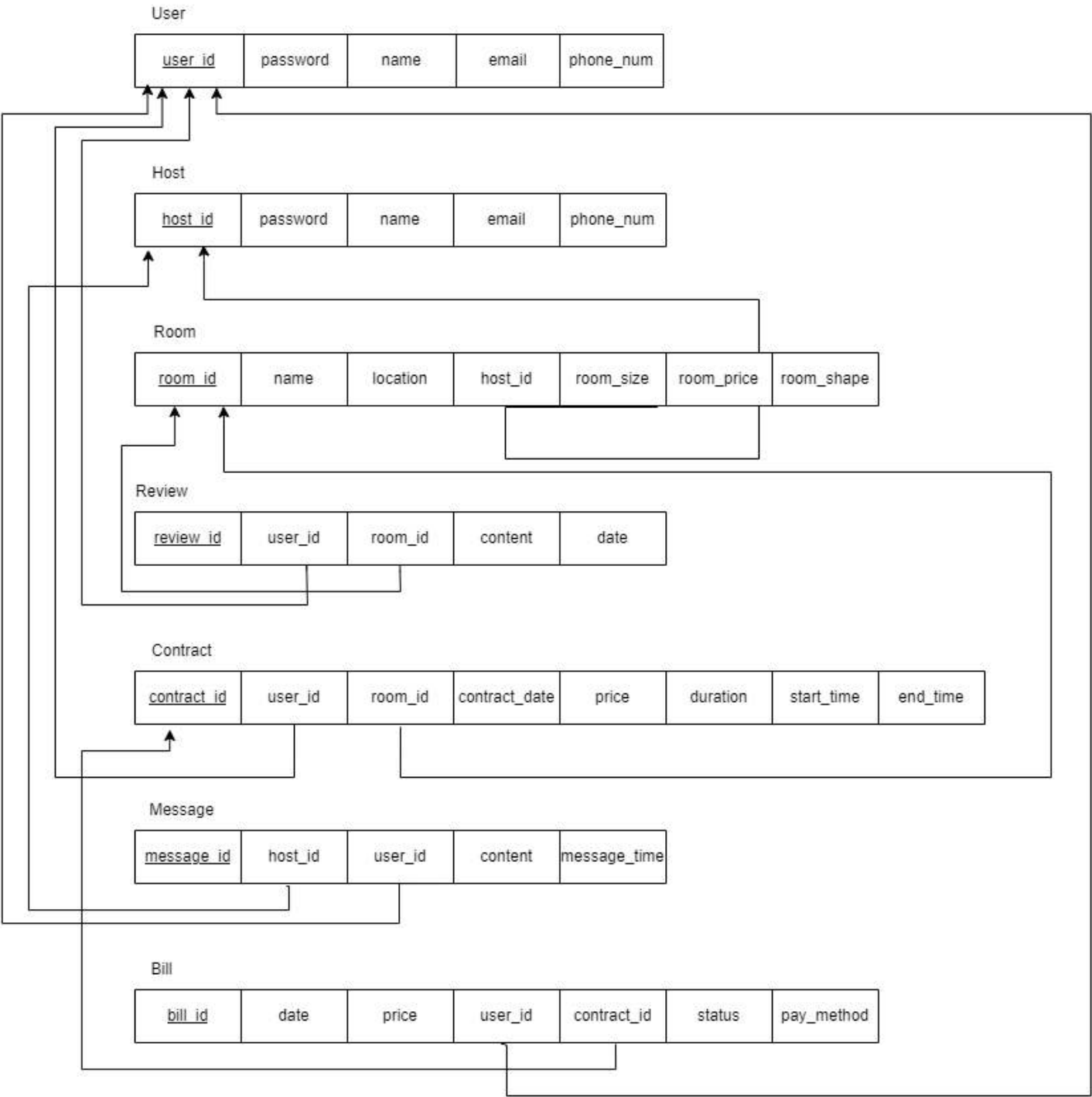
ex) 방의 크기, 방의 가격, 방의 형태 등등의 값을 가지므로

Option의 개수의 개별 요소로 나누어야 한다.

나머지 2NF 3NF의 형태는 만족하므로 최종적인 ERD는



최종 관계



6. MySQL DBMS Database를 생성.

스키마 생성

```
CREATE SCHEMA `Term_project` ;
```

테이블 생성

```
CREATE TABLE user (  
    user_id VARCHAR(100) PRIMARY KEY,  
    password VARCHAR(100) NOT NULL,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    phone_num VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE host (  
    host_id VARCHAR(100) PRIMARY KEY,  
    password VARCHAR(100) NOT NULL,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    phone_num VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE room (  
    room_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    location VARCHAR(100),  
    host_id VARCHAR(100) NOT NULL,  
    room_size INT NOT NULL,  
    room_price INT NOT NULL,  
    room_shape VARCHAR(50),  
    FOREIGN KEY (host_id) REFERENCES host(host_id)  
);
```

```
CREATE TABLE contract (  
    contract_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    room_id INT NOT NULL,  
    user_id VARCHAR(100) NOT NULL,  
    contract_date DATE NOT NULL,  
    price INT NOT NULL,  
    duration INT NOT NULL,  
    start_time DATETIME NOT NULL,  
    end_time DATETIME NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES user(user_id),  
    FOREIGN KEY (room_id) REFERENCES room(room_id)  
);
```

```
CREATE TABLE bill (  
    bill_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    contract_id INT NOT NULL,  
    room_id INT NOT NULL,  
    user_id VARCHAR(100) NOT NULL,  
    bill_date DATE NOT NULL,  
    bill_price INT NOT NULL,  
    FOREIGN KEY (contract_id) REFERENCES contract(contract_id),  
    FOREIGN KEY (room_id) REFERENCES room(room_id),  
    FOREIGN KEY (user_id) REFERENCES user(user_id)  
);
```



```

bill_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
date DATE NOT NULL,
price INT NOT NULL,
user_id VARCHAR(100) NOT NULL,
contract_id INT UNIQUE,
status VARCHAR(20) NOT NULL,
pay_method VARCHAR(50) NOT NULL,
FOREIGN KEY (user_id) REFERENCES user(user_id),
FOREIGN KEY (contract_id) REFERENCES contract(contract_id)
);

```

```

CREATE TABLE message (
message_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
user_id VARCHAR(100) NOT NULL,
host_id VARCHAR(100) NOT NULL,
content VARCHAR(1000) NOT NULL,
message_time DATETIME NOT NULL,
FOREIGN KEY (user_id) REFERENCES user(user_id),
FOREIGN KEY (host_id) REFERENCES host(host_id)
);

```

```

CREATE TABLE review (
review_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
user_id VARCHAR(100) NOT NULL,
room_id INT NOT NULL,
content TEXT NOT NULL,
date DATE NOT NULL,
FOREIGN KEY (user_id) REFERENCES user(user_id),
FOREIGN KEY (room_id) REFERENCES room(room_id)
);

```

추후 작업

Review에 대해 CASCADE옵션

```

ALTER TABLE `Term_project`.`review`
DROP FOREIGN KEY `review_ibfk_1`,
DROP FOREIGN KEY `review_ibfk_2`;
ALTER TABLE `Term_project`.`review`
ADD CONSTRAINT `review_ibfk_1`
FOREIGN KEY (`user_id`)
REFERENCES `Term_project`.`user` (`user_id`)
ON DELETE CASCADE
ON UPDATE CASCADE,
ADD CONSTRAINT `review_ibfk_2`
FOREIGN KEY (`room_id`)

```

```
REFERENCES `Term_project`.`room` (`room_id`)
ON DELETE CASCADE
ON UPDATE CASCADE;
```

Bill에 대해 CASCADE 옵션

```
ALTER TABLE `Term_project`.`bill`
DROP FOREIGN KEY `bill_ibfk_2`;
ALTER TABLE `Term_project`.`bill`
ADD CONSTRAINT `bill_ibfk_2`
FOREIGN KEY (`contract_id`)
REFERENCES `Term_project`.`contract` (`contract_id`)
ON DELETE CASCADE
ON UPDATE CASCADE;
```

7. 기본환경은 가상머신에 CentOS + MySQL이 설치되어야 한다.

기존 CentOS + Mysql 안에 새로운 databases를 만들어서 사용하였습니다.

```
root@database:~
Your MySQL connection id is 19
Server version: 5.7.43 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> showdatabases
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
for the right syntax to use near 'showdatabases' at line 1
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| Term_project |
| madang |
| mysql |
| performance_schema |
| sys |
| test |
+-----+
7 rows in set (0.00 sec)

mysql>
```

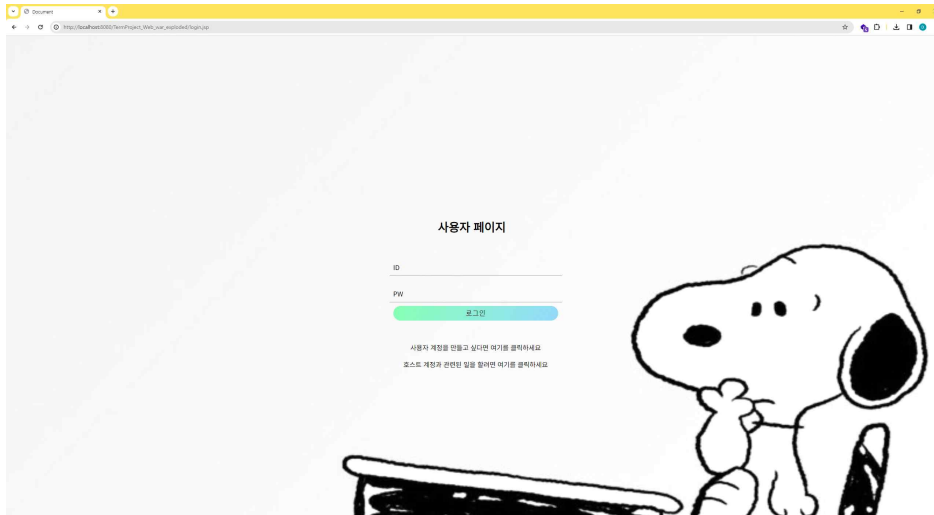
8. 결과물 시연 결과

사용언어 : JAVA(JSP)

사용자 기능

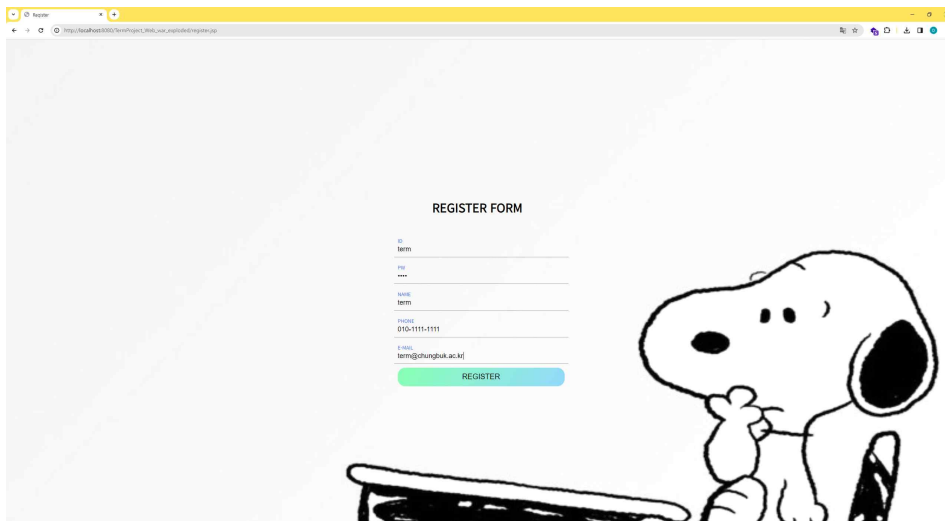
1.사용자 전용 로그인 페이지입니다.

로그인 하고 싶은 정보가 DB에 있을 경우에만 로그인이 가능합니다.



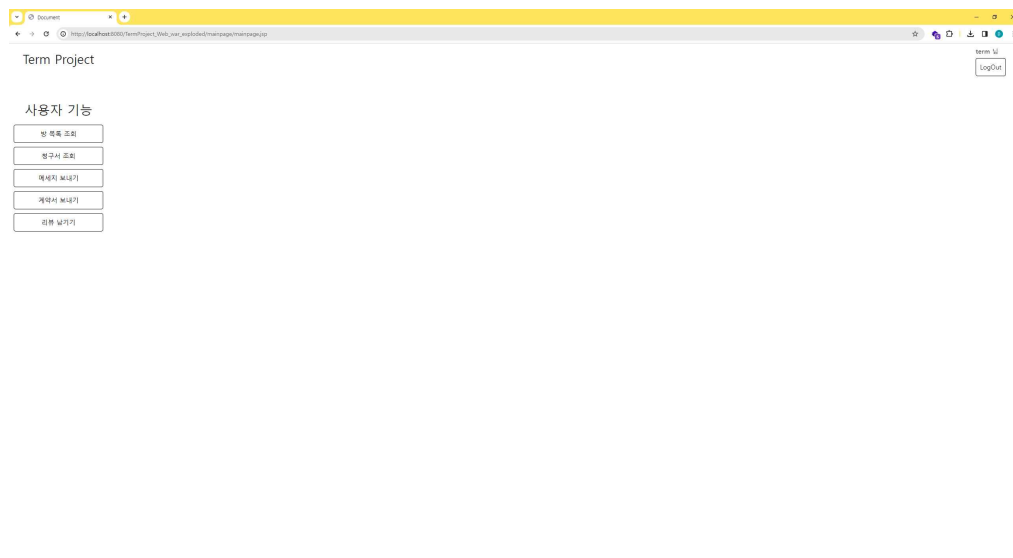
2.사용자 회원가입 페이지입니다.

ID가 겹치지 않을 경우 회원가입이 가능하게 설정하였습니다.



3. 사용자 메인화면 페이지입니다.

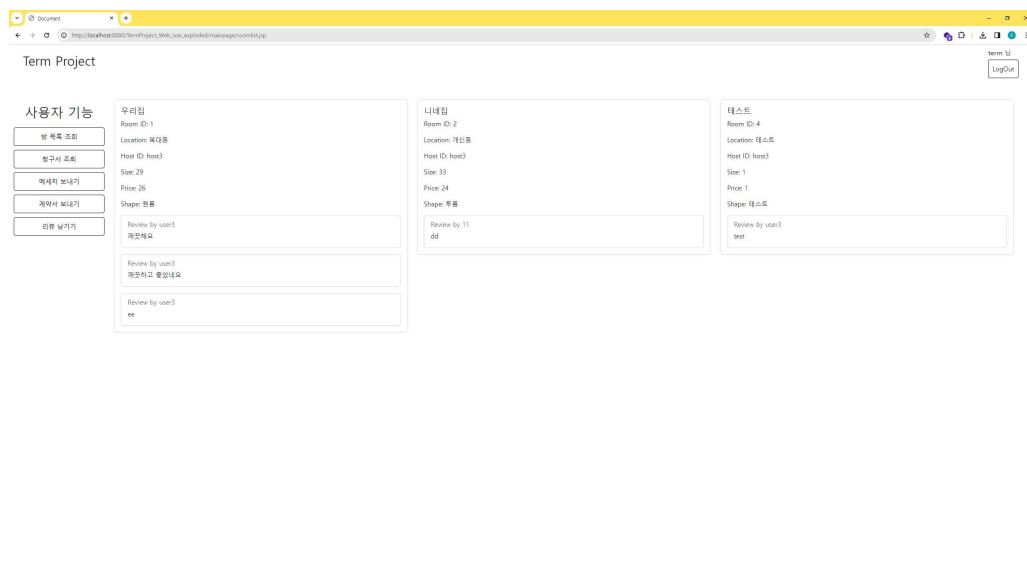
기본 템플릿으로 nav로 여러 기능을 사용할 수 있게 하였고 logout하면 로그인 화면으로 갈 수 있게 설정하였습니다.



4. 방 목록 조회

현재 Host들이 추가한 방의 목록을 확인 할 수 있습니다.

User들이 Review를 남길 경우 해당 방 밑에 보이도록 설정하였습니다.



5. 계약서 작성

해당 방을 확인하고 계약서를 작성할 수 있게 하였습니다.

The screenshot shows a web browser window with the URL `http://localhost:3030/TermProject_Web_war_exploded/mainpage/contractAnd.jsp`. The page title is "Term Project". On the left, there is a sidebar with the heading "사용자 기능" (User Functions) and five buttons: "방 목록 조회" (View Room List), "청구서 조회" (View Invoice), "메세지 보내기" (Send Message), "계약서 보내기" (Send Contract), and "리뷰 남기기" (Leave Review). The main content area is titled "계약서 작성" (Contract Creation) and contains several input fields: "계약일" (Contract Date) with value "2023-11-29", "방 번호" (Room Number) with value "4", "가격" (Price) with value "22", "기간" (Period) with value "30", "계약 시작 시간" (Contract Start Time) with value "2023-11-29 오후 10:14", and "계약 종료 시간" (Contract End Time) with value "2023-12-29 오후 10:15". At the bottom of the form is a blue button labeled "계약서 작성" (Create Contract). In the top right corner, there is a "term" logo and a "LogOut" button.

6. 만들어진 계약서에 대한 청구서 조회

계약시 청구서가 자동으로 생성됩니다.

The screenshot shows the same web browser window with the URL `http://localhost:3030/TermProject_Web_war_exploded/mainpage/bill.jsp`. The page title is "Term Project". The sidebar is identical to the previous screenshot. The main content area is titled "청구서 목록" (Invoice List) and displays a table with one row of invoice details: "Bill ID: 1", "Date: 2023-11-29", "Price: 22", "User ID: term", "Contract ID: 6", "Status: unpaid", and "Payment Method: credit". In the top right corner, there is a "term" logo and a "LogOut" button.

7. 사용자는 호스트에게 메시지를 전송 할 수 있습니다.

Term Project

term 님
Logout

사용자 기능

- 방 목록 조회
- 청구서 조회
- 메세지 보내기
- 계약서 보내기
- 리뷰 남기기

호스트에게 메세지 보내기

Host ID: host3

Message: 원선금 줄여주세요

Send Message

8. 사용자는 자신의 계약 목록을 불러와 해당 방에 머문 내역이 있다면 리뷰를 남길 수 있도록 하였습니다.

Term Project

term 님
Logout

사용자 기능

- 방 목록 조회
- 청구서 조회
- 메세지 보내기
- 계약서 보내기
- 리뷰 남기기

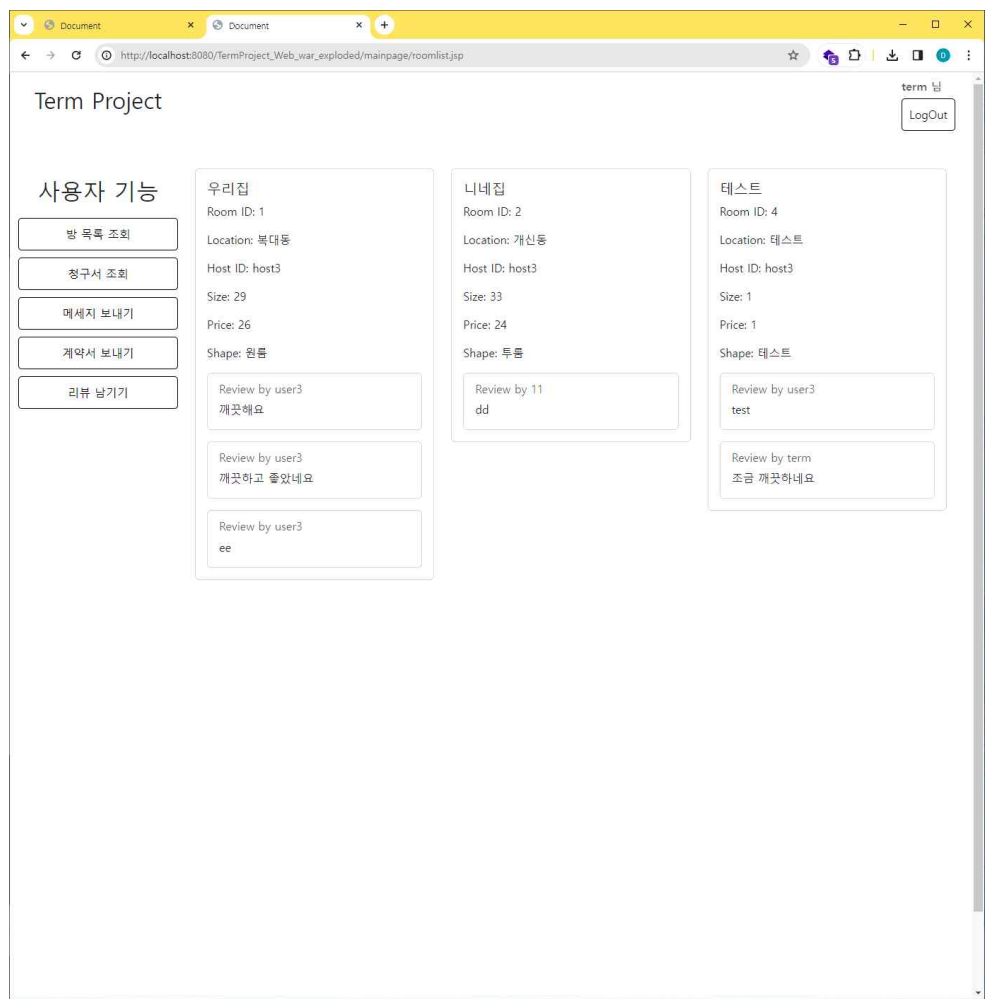
리뷰 작성

방 번호: 4

리뷰 내용: 조금 깨끗하네요

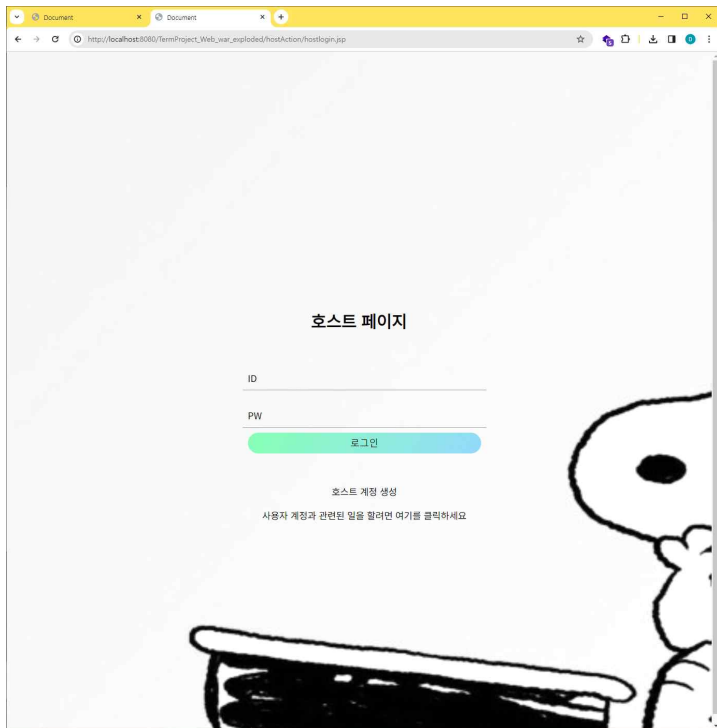
리뷰 작성

9. 리뷰를 작성할 경우 추가 된 것을 확인 할 수 있습니다.

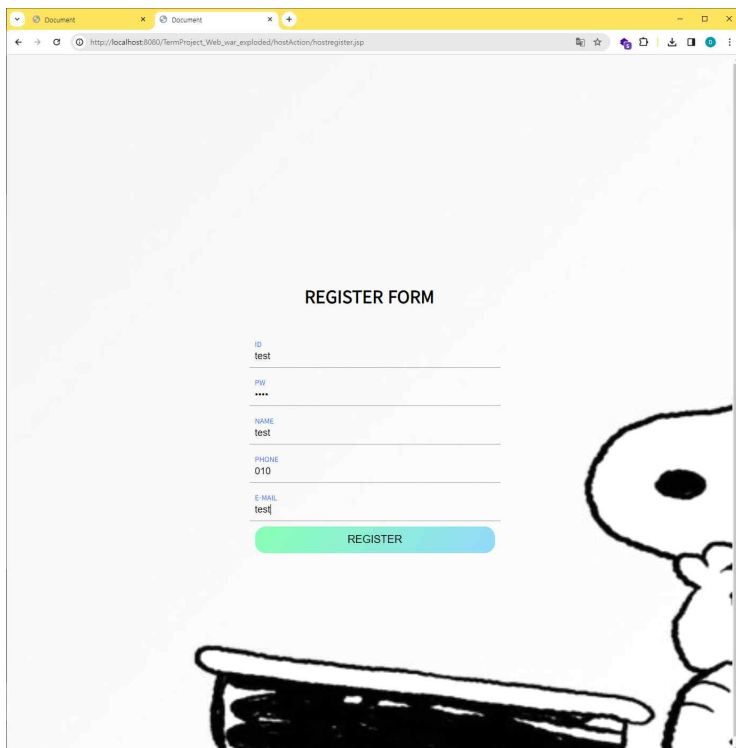


호스트 기능

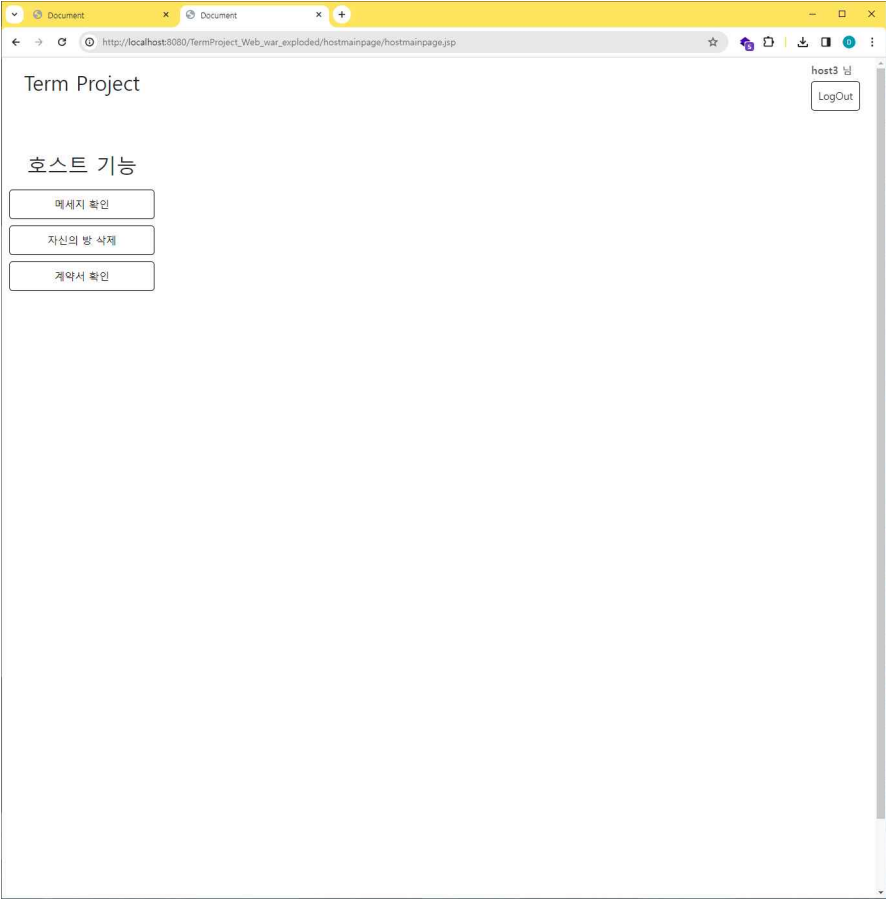
1. 사용자 페이지와 마찬가지로입니다. 이동 방법은 사용자 페이지에 있는 “호스트 계정과 관련한 일을 하려면 여기를 클릭하세요”를 누르면 됩니다.



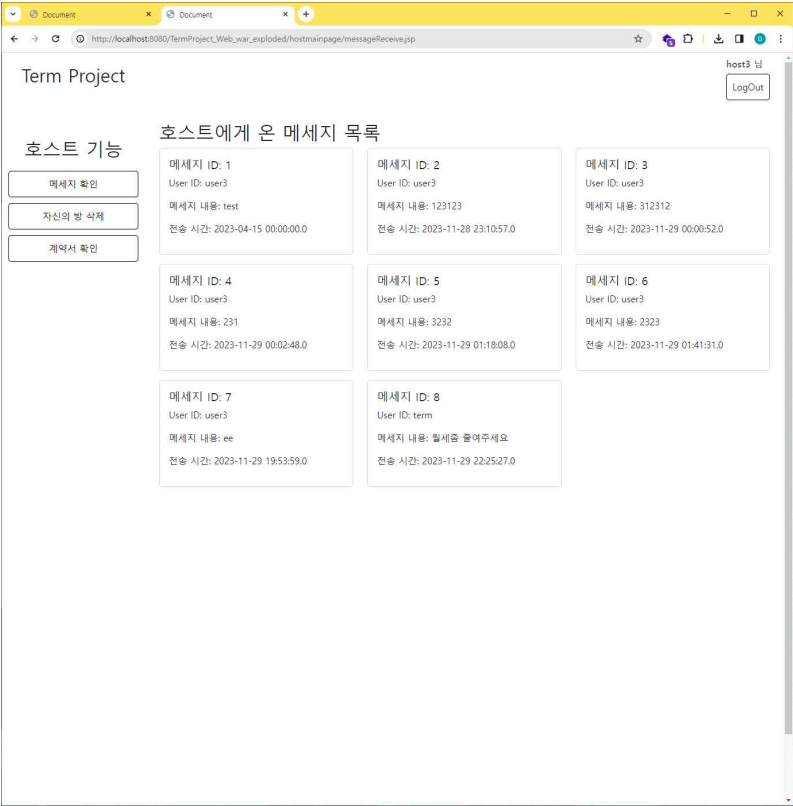
2. 호스트 회원 가입 기능입니다.



3. 호스트 페이지입니다. 앞서 설명한 사용자에서 항목만 조금 바뀌었습니다.

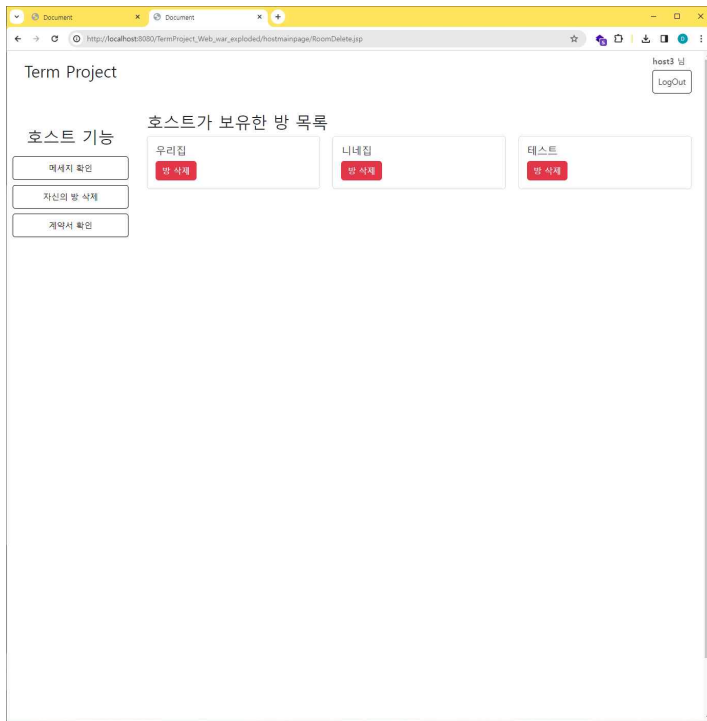


4. 사용자가 호스트에게 전송한 메시지 확인



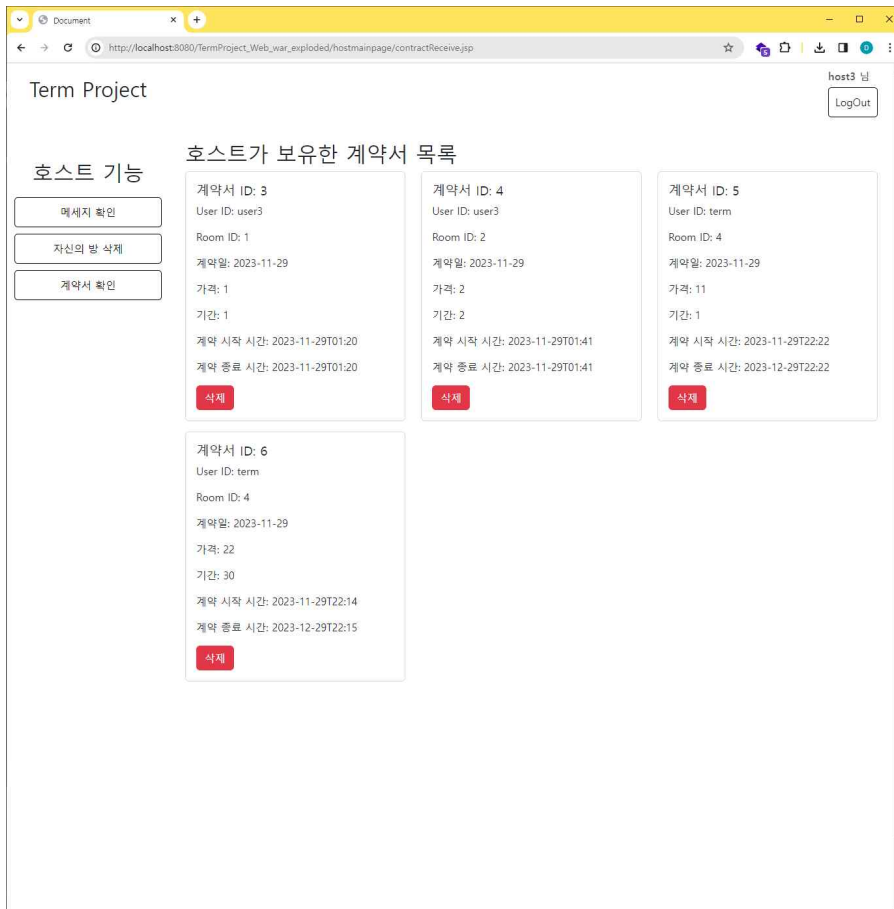
5. 보유한 방에 대해 삭제

방에 리뷰가 있더라도 지울 수 있게 하였습니다.



6. 계약서 목록 확인

호스트가 날이 겹치거나 혹은 마음에 들지 않는 계약에 대해 삭제 할 수 있게 하였습니다.



9. Github url

Github url : https://github.com/NKIWI0310/Database_TermProject