

데이터베이스 프로그래밍 최종보고서

02분반 6팀 덕방이(부동산 중개)

20160925 컴퓨터학과 강은영

20160966 컴퓨터학과 송혜진

20160971 컴퓨터학과 여지은

(2018년 12월 23일)

목 차

1. 시스템 개요 및 특징
2. 핵심기능 및 세부기능
3. 요구사항 분석 결과(Use Case)
4. 시스템 구성도
5. UI 설계
6. 데이터베이스 설계
7. 클래스 설계
8. Request URI 정의 및 처리 흐름
9. 구현방법
10. 결론 및 소감
11. 업무 분담

1. 시스템 개요 및 특징

1) 개발 배경

매물을 구할 때, 직접 부동산에 가기에는 부담을 느끼지만 미리 정보를 찾아보고 싶은 사람들이 많다. 하지만 학업과 알바에 바쁜 대학생들과 직장과 야근에 바쁜 직장인들은 직접 매물을 찾아보기 힘들다. 이렇게 바쁜 현대인들에게는 언제 어디서든 나에게 맞는 매물을 찾을 수 있는 주거정보 플랫폼이 필요하다. 또 공인 중개업자들에게는 기존 부동산 서비스와 다른 웹을 이용한 새로운 방식의 서비스 도입이 필요하다.

2) 특징 및 장점

- 바쁜 현대인들의 생활에 맞춘 주거정보 플랫폼
- 중개사무소에 가기 전에 원하는 정보를 미리 확인 가능
- 매물을 확인, 비교하는 시간 절약
- 원하는 매물을 찜하고 사용자가 찜 한 매물과 유사한 매물을 추천

유사한 사례로는 직방과 다방이 대표적이다.



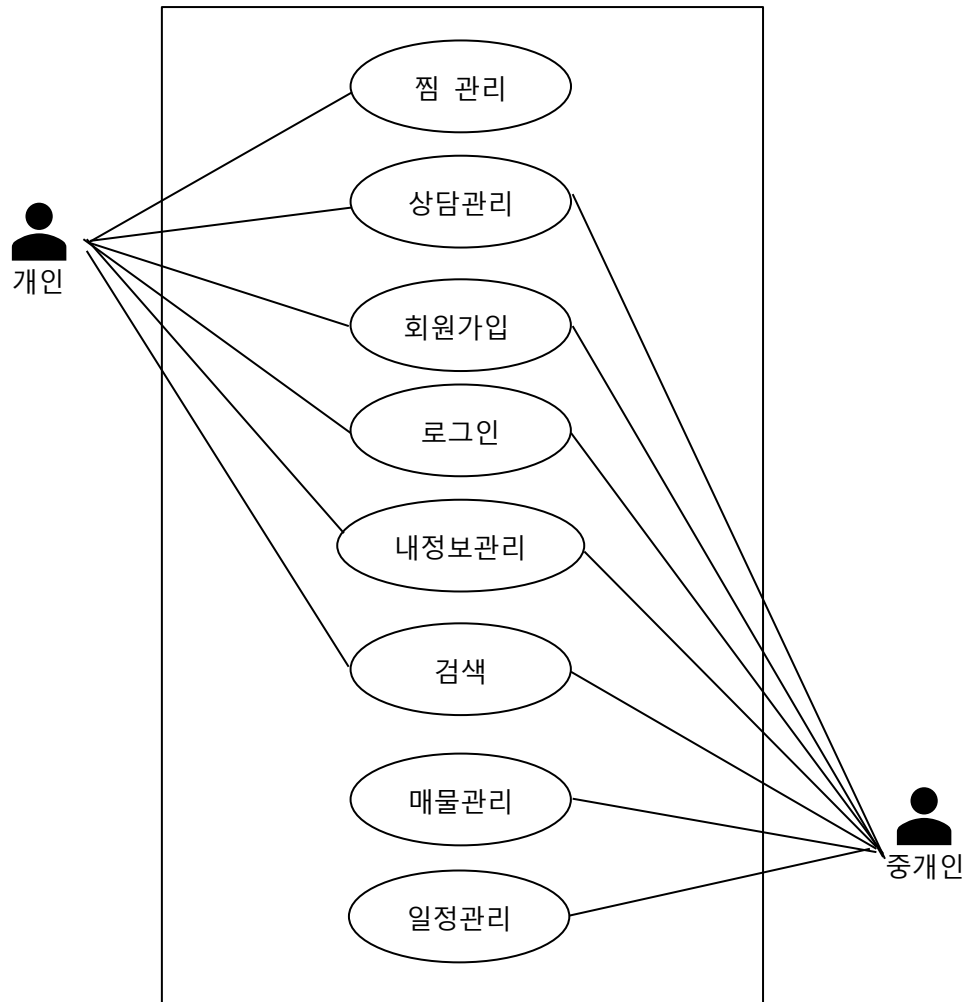
2. 핵심기능 및 세부기능

핵심기능으로는 검색, 필터링, 매물 찜 하기, 유사한 매물 추천, 상담예약이 있고 세부 기능으로는 회원가입, 로그인/로그아웃, 내정보관리, 매물 관리가 있다

- 1) 회원가입 & 로그인/로그아웃: 개인과 중개인으로 가입하고 개인과 중개인이 로그인/아웃한다.
- 2) 내 정보 관리 : 로그인 후 내 정보 창에서 개인정보를 확인하고 수정한다.
- 3) 매물 관리 : 중개인은 매물을 등록하고 수정, 삭제 한다.
- 4) 검색 : 사용자가 지역명을 가지고 매물을 검색한다.
- 5) 필터링 : 원하는 물건을 필터링해서 검색할 수 있다. 필터링 조건에는 물건 형태, 주소, 가격, 평수, 옵션, 중개인, 중개사무소명 등이 있다.
- 6) 매물 찜 하기 : 여러 물건들을 둘러보다가 마음에 드는 물건을 발견하면 찜 하기 버튼을 눌러서 그 물건을 찜 한다
- 7) 유사한 매물 추천 : 자신이 찜한 매물을 바탕으로 유사한 매물을 추천해준다.
- 8) 상담 예약 기능 : 원하는 물건의 상세 페이지에 있는 상담 예약 버튼을 눌러 중개업자가 가능한 날짜와 개인이 가능한 날짜를 고려해 상담 예약한다.

3. 요구사항 분석 결과

1) Use Case



액터	의미
개인	회원으로 매물을 검색하고 중개인을 찾는 사람
중개인	회원으로 매물을 관리하는 사람

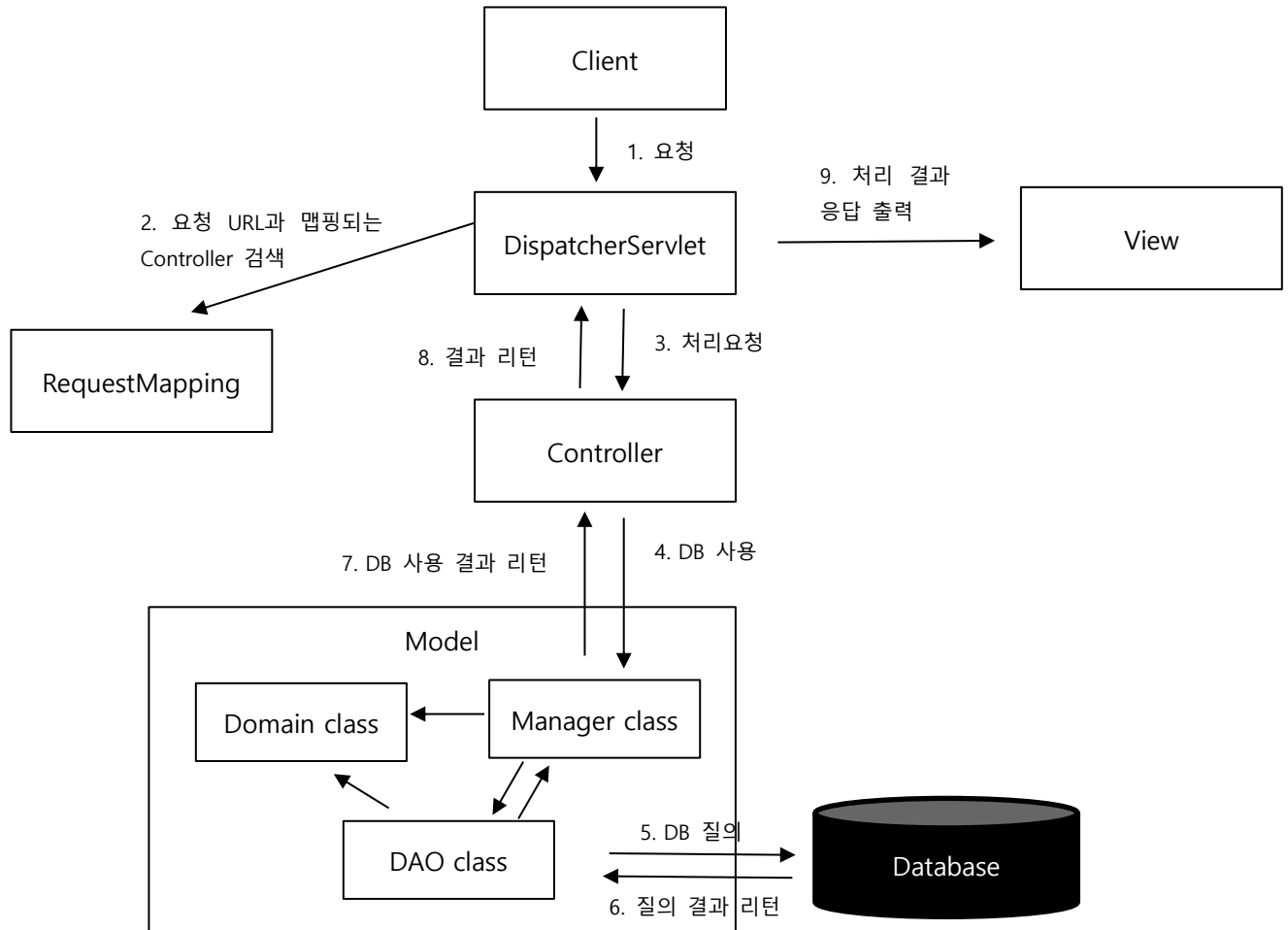
Use case	의미
로그인	사용자는 내정보관리, 찜, 상담예약, 매물검색을 하려면 반드시 시스템에 로그인해야 한다.
내정보관리	자신의 정보를 확인, 수정 할 수 있다.
찜 관리	개인이 원하는 매물을 찜 하거나 필요 없는 매물의 찜을 삭제할 수 있다.
상담관리	개인이 중개사의 상담가능 날짜를 보고 상담하고 싶은 날짜를 선택하여 중개사와 상담날짜를 예약한다.

매물검색	검색한 검색기준에 의해 매물을 찾을 수 있다.
매물관리	중개인은 자신의 매물을 등록, 수정, 삭제 할 수 있다.

2) 요구사항 정의

요구사항 명	세부 요구사항
회원가입	개인(이름, 이메일, 전화번호, 주소, 비밀번호) / 중개업자(중개사무소명, 중개사 등록번호, 사업자 등록번호, 중개사무소 주소, 중개사 대표 명, 중개사 번호, 이름, 이메일, 비밀번호)
로그인	이메일, 비밀번호
물건 등록	1. 물건을 게시판에 올린다. 2. (필수 입력 정보)주소, 방 종류, 거래종류, 가격, 건물 층수 - 건물 층수, 해당 층수, 건물크기 - 공급면적, 전용면적 / (추가 정보)관리비, 주차 여부, 엘리베이터, 난방종류, 반려동물, 입주 가능일, 옵션항목
물건 수정	물건 정보를 수정하고 수정 버튼을 누른다.
물건 삭제	물건 정보 리스트 옆에 있는 삭제 버튼을 누른다.
물건 검색	1. 검색할 물건의 주소를 입력 받는다. 2. (선택 사항)매물 종류, 방 종류, 보증금, 월세 등으로 검색 범위를 좁힌다.
프로필	사진, 이름, 중개사무소명, 중개사무소 주소, 중개사 번호
마이페이지	회원정보, 회원정보 수정, 찜 한 방 정보, 상담 예약 정보를 보여준다.
방 찜 하기	원하는 물건에 찜 하기 버튼을 눌러서 방을 찜 한다.(마이페이지에 반영됨)
상담 예약	1. 상담이 가능한 날짜를 보여준다. 2. 제시된 날짜 중 원하는 날짜를 고른다.(마이페이지에 반영됨)
효율성	시간 효율성: 3 초 이내 결과 계산, 5 초 이내 결과 화면 출력
매물	id, 매물 종류, 주소, 거래 종류(월, 전세 등), 면적, 입주일, 계약기간, 특징, 사진, 등록인 ID(중개인), 예약
회원	id, pw, 성별, 이메일, 핸드폰번호, 중개/개인 구분 id
찜	사용자 id, 매물 id, 등록 날짜

4. 시스템 구성도



시스템 MVC구조 Model2로 구성하였다. MVC구조의 각 컴포넌트의 역할은 who, when, what으로 Controller는 클라이언트의 요청을 받았을 때, 그 요청에 대해 실제 업무를 수행하는 모델 컴포넌트를 호출한다. 또한 클라이언트가 보낸 데이터가 있다면, 모델에 전달하기 쉽게 데이터를 가공한다. 모델이 업무를 마치면 그 결과를 뷰에게 전달한다. Model은 컨트롤러가 호출할 때, 요청에 맞는 역할을 수행하고 비즈니스 로직을 구현하는 영역으로 DB에 간접하고 데이터를 추출하거나 저장, 삭제, 업데이트, 변환 등이 작업을 수행한다. View는 컨트롤러로부터 받은 모델의 결과값을 가지고 사용자에게 출력할 화면을 만드는 일을 하여 화면에 표시한다.

1. Client가 특정 URL로 요청을 보내면 2. Dispatcher Servlet이 이 요청을 RequestMapping으로 넘기고 해당 요청에 맞는 Controller를 찾아 Controller를 리턴 한다. 3. Dispatcher Servlet이 해당 Controller에게 Client에서 받은 값들을 넘기면 Controller가 4~8 Business Logic를 수행하는 Service를 요청하고 결과를 받는다. 9. Dispatcher Servlet는 결과에 맞는 View 출력한다.

5. UI 설계



메인(로그인 전)

로그인

메인(로그인 후)



검색한 결과창



내정보창



상세 매물 창

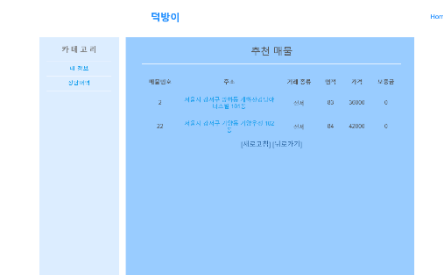
(검색결과창에서 매물 주소를 클릭)



중개인의 매물 관리 창



내 매물창



유사매물추천창

덕방이

Home

환복록

jothang@naver.com 정보

Logout

상담예약	
일련	일시
2018-11-01	O
2018-11-05	O
2018-11-06	O
2018-11-08	O
2018-11-10	O
2018-11-11	O
2018-11-11	O
2018-11-16	O
2018-11-19	O
2018-11-22	O
2018-11-24	O
2018-11-30	O

<예약 전부>

상담 내용

상담예약

상담예약창

덕방이

Home

내 덕방

hokosaki@naver.com 정보

Logout

카테고리

내 정보

상담내역

스케줄 추가

예약 스케줄 관리하는 작업입니다. 가능한 일 또는 불가능 일의 날짜를 추가해주세요.
YYYY-MM-DD로 월/일/년 입력해주세요 ex)2018-08-21

년

월

일

추가

2018-01-17

2018-01-08

2019-01-05

중개인의 스케줄 관리 창

덕방이

Home

환복록

jothang@naver.com 정보

Logout

카테고리

내 정보

상담내역

상담내역

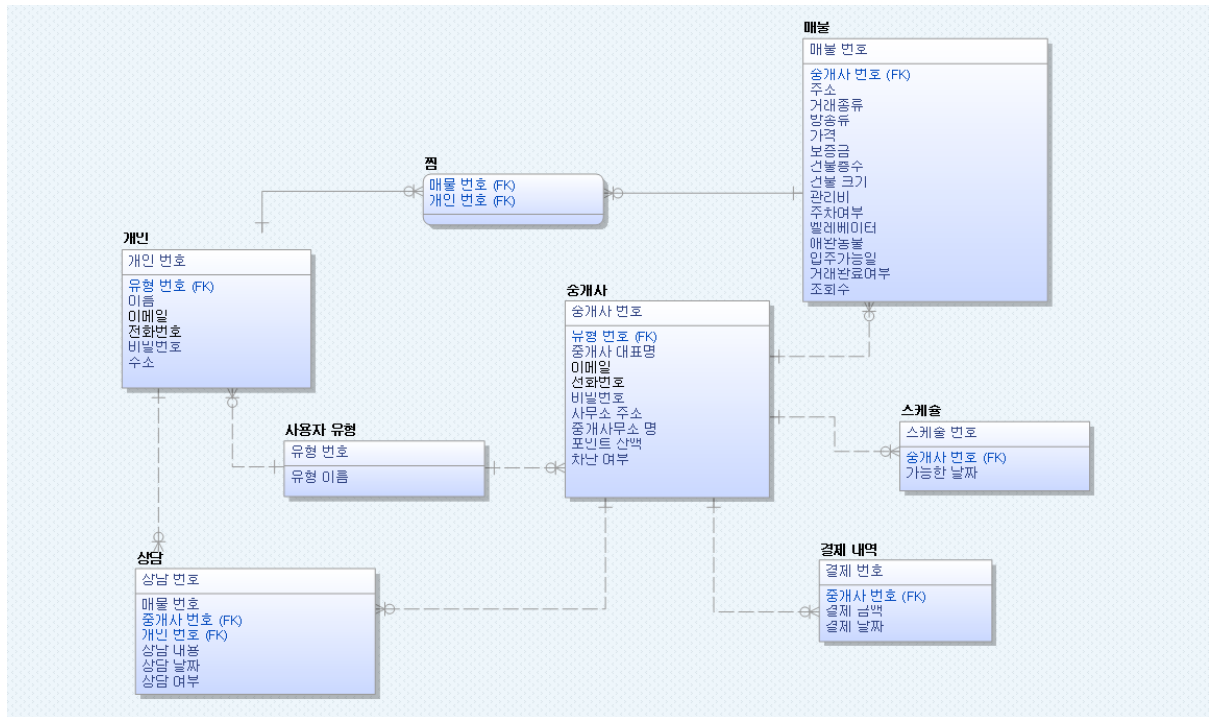
상담번호	중개인	상담일자	상담분류	서지 여부	
3	7호준	3	2018-10-08	FALSE	상세
19	나연자	6	2018-11-18	FALSE	상세
20	나연자	7	2018-12-29	FALSE	상세
21	나연자	6	2018-12-29	FALSE	상세
22	나연자	7	2019-01-02	FALSE	상세
18	김혜경	1	2018-11-14	FALSE	상세

상담내역 창

5. 데이터베이스 설계

1. 논리적 모델링

1) ER다이어그램



- ➔ 사용자 유형 : 가입시 회원의 유형을 지정하는 TABLE
- ➔ 개인 : 사용자 유형 중 '개인' 회원인 사용자의 정보를 담은 TABLE
- ➔ 중개사 : 매물을 등록하고 관리하는, 사용자 유형 중 '중개사' 회원인 사용자의 정보를 담은 TABLE
- ➔ 점 : 사용자가 나중에 다시 보고 싶고, 마음에 드는 매물을 점 하여 저장하는 TABLE
- ➔ 상품 : 개인과 중개사가 상담한 정보를 담은 TABLE
- ➔ 매물 : 중개사가 개인에게 매물 정보를 제공할 때 필요한 매물에 대한 매물 정보 TABLE
- ➔ 스케줄 : 중개사가 개인과 상담 할 수 있는 날짜에 대한 정보를 담은 TABLE
- ➔ 결제 내역 : 중개사가 매물 등록 시에 사용한 포인트에 대한 정보를 담은 TABLE

2) 객체 및 속성 정의

1. 사용자 유형 : 가입시 회원의 유형을 지정하는 TABLE

유형 번호(PK)	가입한 회원의 유형을 식별할 수 있는 번호
유형 이름	가입한 회원의 유형 이름 (ex)개인, 중개인)

2. 개인 : 사용자 중에서 '개인' 회원인 사용자의 정보를 담은 TABLE

개인 번호(PK)	개인을 식별할 수 있는 번호
유형 번호(FK)	가입한 회원의 유형을 식별할 수 있는 번호
이름	개인 회원의 이름
이메일	개인 회원의 이메일
전화번호	개인 회원의 전화번호
비밀번호	개인 회원의 비밀번호
주소	개인 회원의 주소

3. 중개사 : 매물을 등록하고 관리하는 중개사와 관련된 정보 TABLE

중개사 번호(PK)	중개사를 식별할 수 있는 번호
유형 번호(FK)	가입한 회원의 유형을 식별할 수 있는 번호
중개사 대표 명	중개사무소를 대표하는 대표자의 이름
이메일	로그인 시 아이디 역할을 하는 중개사의 이메일 주소
전화번호	회원가입 한 중개사 당사자의 전화번호
비밀번호	로그인 시 입력할 비밀번호
중개사무소 주소	중개사가 일하는 중개사무소의 주소
중개사무소 명	중개사가 일하는 중개사무소의 이름
중개사 등록번호	자격증을 소지한 중개사임을 증명하는 번호 중개사 등록번호 형식을 만들어서 형식을 벗어나면 입력 불가
포인트 잔액	중개사가 매물 등록 시 이용하는 포인트의 잔액
사업자 등록번호	사업자 등록을 정상적으로 했음을 증명하는 번호 중개사 등록번호 형식을 만들어서 형식을 벗어나면 입력 불가
차단 여부	허위 매물을 올려서 개인에게 신고를 받고, 운영자에 의해 로그인을 차단당했는지 정보

4. 매물 : 중개인이 개인에게 매물 정보를 제공할 때 필요한 매물에 대한 매물 정보 TABLE

매물 번호(PK)	매물을 식별할 수 있는 번호
중개사 번호(FK)	매물을 등록한 중개사를 식별하는 번호
주소	누구든 식별 할 수 있는 상세한 매물의 주소
거래 종류	월세, 전세, 매매 등 매물의 거래 종류
방 종류	아파트, 빌라, 원룸 등 거래할 매물의 건물 종류
가격	등록한 매물의 가격
보증금	거래 종류가 월세 일 경우, 매물에 대한 보증금
건물 층수	등록한 매물의 층수

건물 크기	등록한 매물의 평수
관리비	거주 시 납부해야 하는 관리비
주차여부	매물의 추가 속성으로, 주차가 가능한지의 여부
엘레베이터 유무	건물 내에 엘리베이터 존재 여부
애완동물	건물 내에서 애완동물의 입주가능 여부
입주 가능일	매물거래 후 개인 회원이 입주 가능한 날짜
거래완료여부	매물의 거래가 완료되었는지 여부를 검사
조회 수	개인회원이 각 매물을 조회(클릭)한 횟수

5. 상담 : 개인과 중개인이 상담한 정보를 담은 TABLE

상담 번호(PK)	상담 정보를 식별할 수 있는 번호
중개사 번호(FK)	중개사를 식별할 수 있는 번호
개인 번호(FK)	개인을 식별할 수 있는 번호
매물 번호	각각의 매물을 식별할 수 있는 번호
상담 내용	개인이 중개인에게 상담 신청한 내용
상담 날짜	개인이 중개인에게 상담 신청한 날짜
상담 여부	개인과 중개인의 상담여부를 나타내는 정보

6. 찜 : 사용자가 나중에 다시 보고 싶고 마음에 드는 매물을 찜 하여 저장하는 TABLE

매물 번호(FK)	각 매물을 식별할 수 있는 번호
개인 번호(FK)	각각의 개인을 식별할 수 있는 번호

7. 결제 내역 : 중개인이 매물 등록시에 사용한 포인트 정보를 담은 TABLE

결제 번호(PK)	사용한 포인트 정보를 식별할 수 있는 번호
중개사 번호(FK)	중개사를 식별할 수 있는 번호
결제 금액	사용한 포인트 금액
결제 날짜	포인트를 사용한 날짜

8. 스케줄 : 중개사가 개인과 상담 할 수 있는 날짜가 들어있는 TABLE

스케줄 번호(PK)	스케줄을 식별할 수 있는 번호
중개사 번호(FK)	중개사를 식별할 수 있는 번호
가능한 날짜	중개사가 가능한 날짜 정보

2) 관계 정의

1. 사용자 유형 -----> 중개사

: 사용자 유형 하나에 중개사 여러 명이 등록되어 일 대 다 관계가 된다. (1:N)
사용자 유형으로 회원을 중개사로 판단한다.

2. 사용자 유형 -----> 개인

: 사용자 유형 하나에 개인 여러 명이 등록되어 일 대 다 관계가 된다. (1:N)
사용자 유형으로 회원을 개인으로 판단한다.

3. 개인 —————> 찜

: 개인 한명이 찜을 여러 개 할 수 있어서 일 대 다 관계가 된다. (1:N)
개인이 찜 목록에 등록을 한다.

4. 개인 -----> 후기

: 개인이 후기를 여러 개 작성 할 수 있어서 일 대 다 관계가 된다. (1:N)
개인이 후기를 작성한다.

5. 개인 -----> 신고

: 개인이 여러 번 신고를 할 수 있어서 일 대 다 관계가 된다. (1:N)
개인이 허위매물로 판단되는 매물을 신고한다.

6. 개인 -----> 상담

: 개인이 상담을 여러 번 신청 할 수 있어서 일 대 다 관계가 된다. (1:N)
개인이 상담을 신청한다.

7. 중개사 -----> 매물

: 중개사 한명이 매물을 여러 개 등록 할 수 있어 일 대 다 관계가 된다. (1:N)
중개사가 매물정보를 입력하여 등록한다.

8. 중개사 -----> 상담

: 중개사가 상담 신청을 여러 번 받을 수 있어서 일 대 다 관계가 된다. (1:N)
중개사는 상담 신청을 받는다.

9. 중개사 -----> 결제 내역

: 중개사가 결제를 여러 번 할 수 있어서 일 대 다 관계가 된다. (1:N)
중개사는 결제 후 결제내역을 확인할 수 있다.

10. 매물 —————> 찜

: 매물이 찜에 여러 번 등록 될 수 있어서 일 대 다 관계가 된다. (1:N)

찜에 매물이 등록 된다.

11. 매물 -----> 후기

: 매물 하나당 후기가 여러 개 등록 될 수 있어서 일 대 다 관계가 된다. (1:N)
매물에 후기가 등록된다.

12. 매물 -----> 신고

: 매물이 신고를 여러 번 받을 수 있어서 일 대 다 관계가 된다. (1:N)
매물이 허위매물로 신고 된다.

2. 물리적 모델링

1) DDL 스크립트

CREATE TABLE CUSTOMER

```
(
    C_NO                INTEGER NOT NULL ,
    USER_NO             INTEGER NOT NULL ,
    C_NAME              VARCHAR2(20) NOT NULL ,
    C_EMAIL              VARCHAR2(50) NOT NULL  CONSTRAINT  email_2069064763 CHECK
(C_EMAIL LIKE '%@%'),
    C_PHONE              VARCHAR2(11) NOT NULL ,
    C_PWD                VARCHAR2(30) NOT NULL  CONSTRAINT  pwd_443703394 CHECK
( LENGTH(C_PWD) BETWEEN 8 AND 15
),
    C_ADDR               VARCHAR2(150) NOT NULL
);
```

➔ 개인회원에 대한 정보를 테이블로 정의

CREATE UNIQUE INDEX CUSTOMER_PK ON CUSTOMER

(C_NO ASC);

ALTER TABLE CUSTOMER

ADD CONSTRAINT CUSTOMER_PK PRIMARY KEY (C_NO);

CREATE UNIQUE INDEX C_UNIQUE ON CUSTOMER

(C_EMAIL ASC,C_PHONE ASC);

ALTER TABLE CUSTOMER

ADD CONSTRAINT C_UNIQUE UNIQUE (C_EMAIL,C_PHONE);

➔ 개인회원의 C_NO 는 기본 키이며, 개인을 구별하는 번호이기에 중복될 수 없다. C_EMAIL와 C_PHONE은 개인의 아이디와 전화번호로, 고유한 값이기 때문에 중복될 수 없음.

```

CREATE TABLE AGENT
(
    A_NO                INTEGER NOT NULL ,
    USER_NO            INTEGER NOT NULL ,
    A_NAME              VARCHAR2(20) NOT NULL ,
    A_EMAIL              VARCHAR2(50) NOT NULL  CONSTRAINT  email_927943092 CHECK
(A_EMAIL LIKE '%@%'),
    A_PHONE              VARCHAR2(11) NOT NULL ,
    A_PWD                VARCHAR2(30) NOT NULL  CONSTRAINT  pwd_854256048 CHECK
( LENGTH(A_PWD) BETWEEN 8 AND 15
),
    A_ADDR              VARCHAR2(150) NOT NULL ,
    A_OFFICE            VARCHAR2(20) NOT NULL ,
    A_POINT              INTEGER DEFAULT 0  NOT NULL ,
    A_BAN                VARCHAR2(10) DEFAULT 'FALSE' NOT NULL
);

```

➔ 중개인 회원에 대한 정보를 테이블로 정의

```

CREATE UNIQUE INDEX AGENT_PK ON AGENT
(A_NO  ASC);
ALTER TABLE AGENT
    ADD CONSTRAINT  AGENT_PK PRIMARY KEY (A_NO);
CREATE UNIQUE INDEX A_UNIQUE ON AGENT
(A_EMAIL  ASC,A_PHONE  ASC);
ALTER TABLE AGENT
ADD CONSTRAINT  A_UNIQUE UNIQUE (A_EMAIL,A_PHONE);

```

➔ 중개인 회원의 A_NO 는 기본 키이며, 중개인을 구별하는 번호이기에 중복될 수 없다.
A_EMAIL와 A_PHONE은 중개인의 아이디와 전화번호로, 고유한 값이기 때문에 중복 될 수 없음.

```

CREATE TABLE COUNSEL
(
    COUNSEL_NO          INTEGER NOT NULL ,
    ITEM_NO             INTEGER NOT NULL ,
    A_NO                INTEGER NOT NULL ,
    C_NO                INTEGER NOT NULL ,
    COUNSEL_DETAIL       VARCHAR2(200) NOT NULL  CONSTRAINT  detail_417441695 CHECK
(LENGTH(COUNSEL_DETAIL) <= 100),

```

```
COUNSEL_DATE          DATE NOT NULL ,
COUNSEL_TF             VARCHAR2(10) DEFAULT 'FALSE' NOT NULL
);
```

➔ 개인이 중개인에게 상담요청을 하기위해 사용되는 테이블.

```
CREATE UNIQUE INDEX COUNSEL_PK ON COUNSEL
(COUNSEL_NO ASC);
ALTER TABLE COUNSEL
```

```
ADD CONSTRAINT COUNSEL_PK PRIMARY KEY (COUNSEL_NO);
```

➔ COUNSEL_NO는 각 상담을 구별하는 번호로 기본 키이고 중복될 수 없다.

```
CREATE TABLE PAY_LOG
```

```
(
    PAY_NO              INTEGER NOT NULL ,
    A_NO                INTEGER NOT NULL ,
    PAY_PRICE           INTEGER NOT NULL ,
    PAY_DATE            DATE NOT NULL
);
```

➔ 중개인이 매물을 등록할 시의 포인트와 사용기록을 보여주기 위한 테이블

```
CREATE UNIQUE INDEX PAY_LOG_PK ON PAY_LOG
(PAY_NO ASC);
ALTER TABLE PAY_LOG
```

```
ADD CONSTRAINT PAY_LOG_PK PRIMARY KEY (PAY_NO);
```

➔ PAY_NO는 각 결제내역을 구별할 수 있는 번호로 기본 키이며 중복될 수 없다.

```
CREATE TABLE SCHEDULE
```

```
(
    SCHEDULE_NO         INTEGER NOT NULL ,
    A_NO                INTEGER NOT NULL ,
    AVAILABLE_DATE       DATE NOT NULL
);
```

➔ 중개인이 사용자에게 상담가능한 날짜를 알려주기 위한 테이블

```
CREATE UNIQUE INDEX SCHEDULE_PK ON SCHEDULE
(SCHEDULE_NO ASC);
```

```
ALTER TABLE SCHEDULE
```

```
ADD CONSTRAINT SCHEDULE_PK PRIMARY KEY (SCHEDULE_NO);
```

```
ALTER TABLE SCHEDULE
```

```
ADD CONSTRAINT SCHEDULE_UNIQUE UNIQUE(A_NO, AVAILABLE_DATE);
```

- ➔ SCHEDULE_NO 는 각 스케줄을 구별 할 수 있는 기본키로 중복될 수 없으며 각 중개인 한명에게 스케줄은 중복될 수 없다.

```
CREATE TABLE ITEM
```

```
(
    ITEM_NO            INTEGER NOT NULL ,
    A_NO              INTEGER NOT NULL ,
    ITEM_ADDR          VARCHAR2(150) NOT NULL ,
    ITEM_DEAL_TYPE      VARCHAR2(10) NOT NULL ,
    ITEM_TYPE           VARCHAR2(12) NOT NULL  CONSTRAINT  ITEM_TYPE_1937124118
CHECK (ITEM_TYPE IN ('원룸', '아파트', '빌라', '오피스텔')),
    ITEM_COST          INTEGER NOT NULL ,
    ITEM_DEPOSIT        INTEGER NULL ,
    ITEM_LAYER          INTEGER NOT NULL ,
    ITEM_SIZE           INTEGER NOT NULL ,
    ITEM_MANAGE_COST    INTEGER NOT NULL ,
    ITEM_PARK_TF        VARCHAR2(10) DEFAULT 'FALSE' NOT NULL ,
    ITEM_ELE_TF         VARCHAR2(10) DEFAULT 'FALSE' NOT NULL ,
    ITEM_PET_TF         VARCHAR2(10) DEFAULT 'FALSE' NOT NULL ,
    ITEM_AVAIL_DATE     DATE NOT NULL ,
    ITEM_CLOSE_TF       VARCHAR2(10) DEFAULT 'FALSE' NOT NULL ,
    ITEM_HITS           INTEGER DEFAULT 0 NOT NULL
);
```

- ➔ 중개인이 등록하는 매물에 대한 정보를 나타내는 테이블

```
CREATE UNIQUE INDEX ITEM_PK ON ITEM
```

```
(ITEM_NO ASC);
```

```
ALTER TABLE ITEM
```

```
ADD CONSTRAINT ITEM_PK PRIMARY KEY (ITEM_NO);
```

- ➔ ITEM_NO는 각 매물을 구별하는 기본 키이며 중복될 수 없다.

```
CREATE TABLE BASKET
```

```
(
    ITEM_NO            INTEGER NOT NULL ,
    C_NO              INTEGER NOT NULL
);
```

- ➔ 개인회원이 찜목록을 구현하기 위한 테이블

```
CREATE UNIQUE INDEX BASKET_PK ON BASKET
```

```
(ITEM_NO ASC,C_NO ASC);
```

```
ALTER TABLE BASKET
```

```
ADD CONSTRAINT BASKET_PK PRIMARY KEY (ITEM_NO,C_NO);
```

➔ ITEM_NO, C_NO는 각 짐을 구별하는 기본 키이므로 중복될 수 없다.

```
CREATE TABLE USER_TYPE
```

```
(
```

```
USER_NO INTEGER NOT NULL ,
```

```
USER_NAME VARCHAR2(10) NOT NULL
```

```
);
```

➔ 개인회원과 중개인회원을 구별하기 위해 사용하는 테이블

```
CREATE UNIQUE INDEX USER_TYPE_PK ON USER_TYPE
```

```
(USER_NO ASC);
```

```
ALTER TABLE USER_TYPE
```

```
ADD CONSTRAINT USER_TYPE_PK PRIMARY KEY (USER_NO);
```

➔ USER_NO 는 회원종류를 구별하는 기본 키이므로 중복될 수 없음.

```
ALTER TABLE CUSTOMER
```

```
ADD (CONSTRAINT USER_TYPE_TO_CUSTOMER_FK FOREIGN KEY (USER_NO) REFERENCES  
USER_TYPE (USER_NO));
```

```
ALTER TABLE AGENT
```

```
ADD (CONSTRAINT USER_TYPE_TO_AGENT_FK FOREIGN KEY (USER_NO) REFERENCES USER_TYPE  
(USER_NO));
```

```
ALTER TABLE COUNSEL
```

```
ADD (CONSTRAINT CUSTOMER_TO_COUNSEL_FK FOREIGN KEY (C_NO) REFERENCES CUSTOMER  
(C_NO));
```

```
ALTER TABLE COUNSEL
```

```
ADD (CONSTRAINT AGENT_TO_COUNSEL_FK FOREIGN KEY (A_NO) REFERENCES AGENT (A_NO));
```

```
ALTER TABLE PAY_LOG
```

```
ADD (CONSTRAINT AGENT_TO_PAY_LOG_FK FOREIGN KEY (A_NO) REFERENCES AGENT (A_NO));
```

```
ALTER TABLE SCHEDULE
```

```
ADD (CONSTRAINT AGENT_TO_SCHEDULE_FK FOREIGN KEY (A_NO) REFERENCES AGENT
```


(A_NO));

ALTER TABLE ITEM

ADD (CONSTRAINT AGENT_TO_ITEM_FK FOREIGN KEY (A_NO) REFERENCES AGENT (A_NO));

ALTER TABLE BASKET

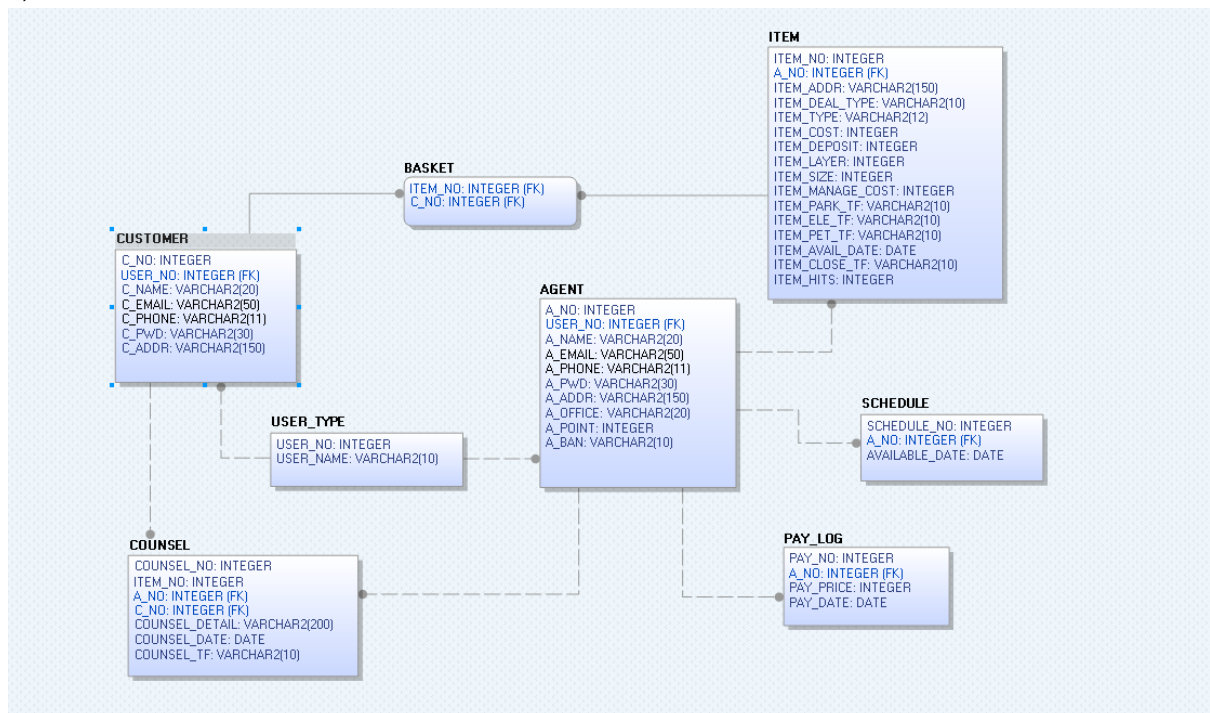
ADD (CONSTRAINT CUSTOMER_TO_BASKET_FK FOREIGN KEY (C_NO) REFERENCES CUSTOMER (C_NO));

ALTER TABLE BASKET

ADD (CONSTRAINT ITEM_TO_BASKET_FK FOREIGN KEY (ITEM_NO) REFERENCES ITEM (ITEM_NO));

➔ 각 외래 키 참조에 대한 정의

2) 스키마 다이어그램



7. 클래스 설계

1) Domain : Java객체들 사이의 메소드 호출에 필요한 데이터들을 포함하는 객체

Domain	설명
Agent	데이터베이스의 AGENT테이블에 대응되는 클래스
Basket	데이터베이스의 BASKET테이블에 대응되는 클래스
Counsel	데이터베이스의 COUNSEL테이블에 대응되는 클래스
Customer	데이터베이스의 CUSTMOER테이블에 대응되는 클래스
Item	데이터베이스의 ITEM테이블에 대응되는 클래스
User	Customer과 Agent 클래스의 부모 클래스로 두 클래스의 공통요소를 가지는 클래스
Schedule	데이터베이스의 SCHEDULE테이블에 대응되는 클래스
Point	데이터베이스의 PAY_LOG테이블에 대응되는 클래스

2) DAO : 데이터 저장소를 공통의 인터페이스를 통해 이용할 수 있도록 해주는 구성요소

DAO	설명
CounselDAO	데이터베이스의 COUNSEL테이블이나 상담관련 테이블에 접근하기 위한 DAO
ItemDAO	데이터베이스의 ITEM테이블이나 매물관련 테이블에 접근하기 위한 DAO
UserDAO	데이터베이스의 USER테이블이나 사용자관련 테이블에 접근하기 위한 DAO
WishDAO	데이터베이스의 WISH테이블이나 찜 관련 테이블에 접근하기 위한 DAO
PointDAO	데이터베이스의 PAY_LOG테이블이나 결제내역관련 테이블에 접근하기 위한 DAO

3) Controller : 사용자와 Model, View 사이의 실행 흐름을 제어하는 클래스

기능	Controller	설명
	Controller	모든 Controller의 부모 클래스로 모든 컨트롤러들이 구현해야 할 execute() method를 선언한 인터페이스
	DispatcherServlet	모든 요청을 받은 Front Controller 클래스
	ForwardController	전달받은 URL로 Forward하는 클래스
	RequestMapping	URI에 따라 컨트롤러를 결정해주는 클래스
사용자	LoginController	Request의 전달 방식에 따라 LoginForm.jsp로 이동할지 UserManager를 통해 DB에 사용자를 확인하고 로그인을 실행하는 컨트롤러
	LogoutController	세션을 삭제하고 무효화 시키는 컨트롤러
	RegisterUserController	가입하려는 사용자가 개인인지 중개인인지에 따라 다르

		게 객체를 만들어 UserManager를 통해 사용자를 추가하는 컨트롤러
	UpdateUserController	개인과 중개인이 자신의 정보를 확인후 주소, 비밀번호, 전화번호를 수정할 수 있도록 해주는 컨트롤러
	UserInfoController	개인과 중개인이 각각의 개인정보를 확인 할 수 있도록 사용자정보를 출력해주는 컨트롤러
상담	ReservationCounsel	Request의 전달방식에 따라 counselForm.sp로 이동할지 CounselManager를 통해 DB에 상담을 추가할지 결정하여 실행하는 컨트롤러
	CounselDetailController	상담 내역페이지에서 개인이 상담신청시 작성했던 상세내용을 출력해주는 컨트롤러
	CounselListController	개인이 중개인에게 상담을 신청 시 개인과 중개인에게 각각의 상담내역을 출력해주는 컨트롤러
	CreateScheduleController	중개인이 개인회원에게 상담을 제공해 줄 수 있는 날짜를 등록할 수 있도록 해주는 컨트롤러
	SheduleListController	중개인이 등록한 상담제공 가능한 날짜를 출력해주는 컨트롤러
매물	DeleteltemController	선택된 매물 테이블에 담겨있는 중개인과 매물을 올린 중개인이 같은 지 확인하고 같으면 매물을 삭제하는 컨트롤러
	InsertItemController	중개인이 보유하고 있는 포인트가 10 포인트 이상이면 매물을 등록하고 중개인의 포인트를 10 포인트 차감하고 pay_log 테이블에 정보를 기록하고, 10 포인트 미만이면 exception 메시지를 등록하는 컨트롤러
	ItemSessionUtils	메인 페이지에서 입력된 주소 값으로 item 테이블에서 주소와 부분 일치하는 데이터를 검색하여 화면에 보여주는 컨트롤러, 검색 페이지에서 필터링 할 값을 선택해서 검색하면 필터링 로직을 돌리고 그 결과로 나온 리스트를 이용해 화면에 출력하는 컨트롤러
	ListItemController	폼에서 수정한 매물 값을 전달받아서 해당 매물을 수정하고 그 결과 매물을 다시 /item/updateFrom.jsp에 보여주는 컨트롤러
	UpdateItemController	Session에 있는 user 객체에서 중개사 번호를 얻은 후, 해당 중개사가 등록한 매물 리스트를 검색하여 /item/usersItem.jsp에 보여주는 컨트롤러
	UserItemController	중개사가 등록한 매물 리스트에서 주소를 선택하면 Session에 있는 user 객체에서 중개사 번호를 얻은 후, 해당 중개사가 등록한 매물에 대한 상세 정보를

		<p>검색하여 /item/usersItemView.jsp에 보여주는 컨트롤러,</p> <p>검색 페이지의 매물 리스트에서 주소를 선택하면 매물에 대한 상세 정보와 매물을 등록한 중개사에 대한 상세 정보를 검색하여 /item/view.jsp에 보여주는 컨트롤러</p>
	ViewItemController	<p>선택된 매물 테이블에 담겨있는 중개인과 매물을 올린 중개인이 같은 지 확인하고 같으면 매물을 삭제하는 컨트롤러</p>
포인트	ListPayLogController	<p>Session에 있는 user 객체에서 중개사 번호를 얻은 후, 그 번호를 이용해 pay_log 테이블의 기록을 검색해 나온 포인트에 관한 상세 정보 결과 리스트를 /point/pointList.jsp에 보여주는 컨트롤러</p>
찜	CountBasketController	<p>매물 검색을 하는 개인이 매물상세페이지를 보고 마음에 들어 찜을 눌렀을 경우 찜 목록에 원하는 매물을 추가하는 컨트롤러.</p>
	CreateWishController	<p>개인이 찜 했던 매물들의 데이터를 토대로 추천 로직을 이용해 비슷한 매물을 추천해주는 컨트롤러.</p>
	RecommandController	<p>개인이 찜 했던 목록에서 원하는 번호의 매물을 삭제하는 컨트롤러.</p>
	RemoveWishController	<p>개인이 찜 했던 목록을 볼 수 있도록 출력하는 컨트롤러.</p>
	WishListController	<p>매물 검색을 하는 개인이 매물상세페이지를 보고 마음에 들어 찜을 눌렀을 경우 찜 목록에 원하는 매물을 추가하는 컨트롤러.</p>

4) View : 사용자 요청 처리 결과에 대한 출력 및 UI 담당하는 JSP

기능	JSP	설명
	index.jsp	메인 기능(화면)을 위한 요청이 실행
사용자	main.jsp	메인 화면
	loginForm.jsp	로그인 화면
	registerForm.jsp	회원가입 입력 화면
	changPw.jsp	개인이 선택한 상담에 해당하는 상담 상세 화면
	counselDetail.jsp	개인의 경우 상담신청한 내역화면. 중개인의 경우 개인이 신청한 상담내역화면.
	counselList.jsp	개인이 찜 한 매물들을 토대로 추천된 매물 확인화면
	recommend.jsp	중개인이 자신이 상담을 제공 할 수 있는 날짜를 등록하여 개인에게 보여질 수 있도록 하거나 자신이 등록한 날짜를 출력하는 화면.

	scheduleList.jsp	개인과 중개인에 따라 자신의 정보를 확인 할 수 있도록 출력하고, 회원정보 변경을 눌러 수정 또한 가능하게 만든 화면
	userInfo.jsp	개인이 마음에 드는 매물을 찜 했던 리스트를 출력하고, 매물번호를 입력 후 매물삭제 버튼을 눌러 매물을 삭제 할 수 있는 화면
	wishList.jsp	개인이 선택한 상담에 해당하는 상담 상세 화면
매물	counselForm.jsp	상담 예약 정보 입력 화면
	insertForm.jsp	등록할 매물 정보 입력 화면
	list.jsp	메인 페이지나 검색 필터를 이용해 나온 매물 리스트 화면
	updateForm.jsp	수정할 매물 정보 입력 화면
	usersItem.jsp	중개사가 등록한 매물 리스트 화면
	usersItem.jsp	중개사가 등록한 매물에 대한 상세 정보 화면
	view.jsp	메인 페이지나 검색 필터를 이용해 나온 매물에 대한 상세 정보와 해당 매물을 등록한 중개사에 대한 상세 정보 화면
포인트	pointList.jsp	중개사가 매물을 등록할 때 기록된 포인트에 대한 상세 정보 화면

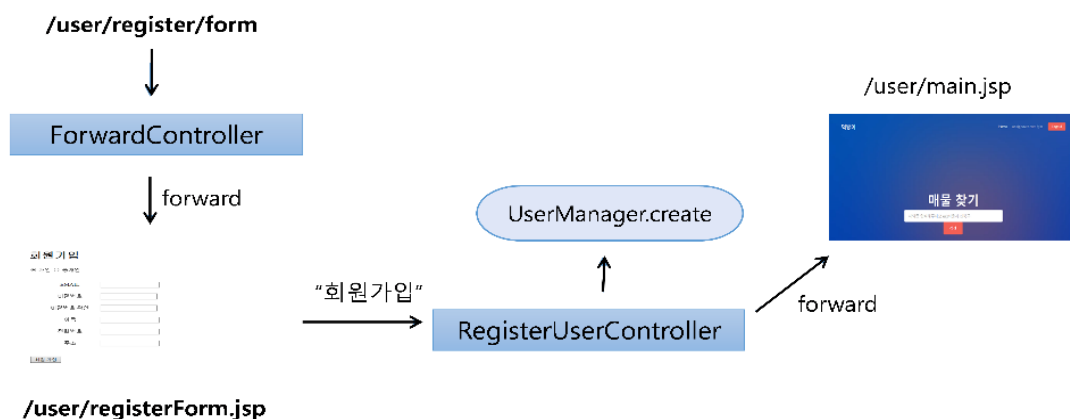
5) Manager : Contoller에서 Model에 접근하기 위해 사용하는 인터페이스를 제공하는 클래스

Manager	설명
UserManager	개인과 중개인에 관련하여 필요한 Controller와 DAO를 연결해주는 클래스
CounselManager	상담기능, 스케줄 관련하여 CounselDAO를 이용해 Controller와 연결해주는 매니저 클래스.
ItemManager	매물 등록, 수정, 삭제, 검색과 관련하여 ItemDAO와 관련된 Controller들을 연결해주는 Manager Class
PaylogManager	중개사가 매물을 등록할 때 차감되는 포인트와 관련하여 PointDAO와 관련된 Controller들을 연결해주는 Manager Class
WishManager	개인의 찜, 추천매물과 관련하여 WishDAO를 이용해 Controller와 연결해주는 매니저 클래스.

8. Request URI 정의 및 처리 흐름

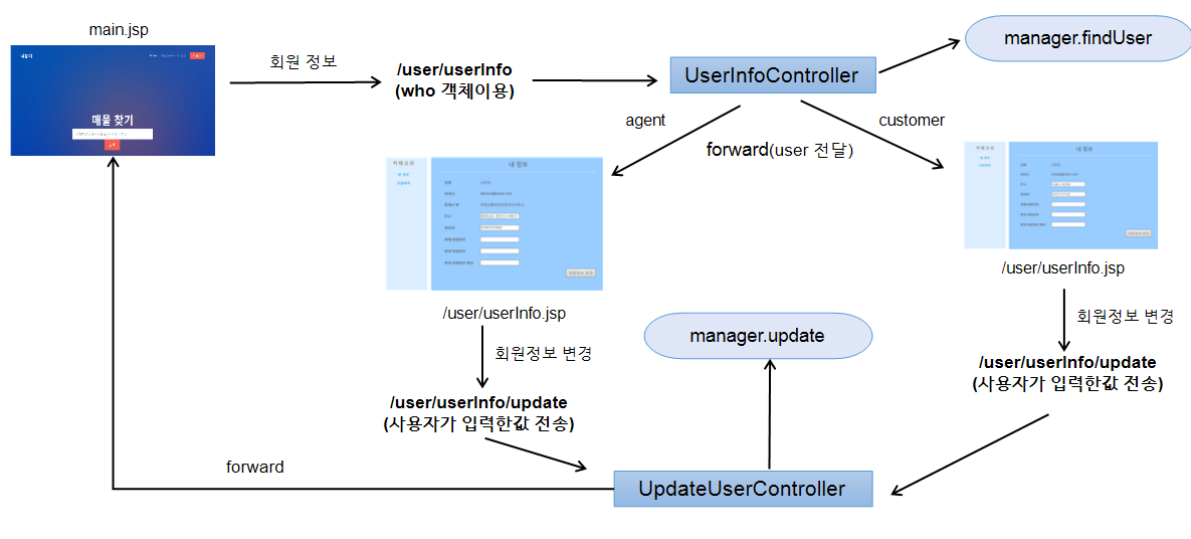
1) Request URI 정의

기능	URI	설명
회원가입 로그인 로그아웃	/	index.jsp로 이동한다.
	/user/main	메인 페이지인 /user/main.jsp를 보여준다.
	/user/login	LoginController를 통해 로그인 입력창으로 이동하거나 로그인을 한다.
	/user/logout	LogoutController를 통해 로그아웃을 한다.
	/user/register/form	회원가입 폼인 /user/register/form.jsp를 보여준다
	/user/register	RegisterUserController를 통해 회원가입을 한다.
상담 내 정보 유사매물추천	/user/counselList	CounselListController를 통해 상담 내역을 보여준다.
	/user/counselDetail	CounselDetailController를 통해 작성자가 작성한 상담 내용을 보여준다.
	/user/wish	WishListController를 통해 일반 사용자에게 찜 목록을 보여준다.
	user/recommand	RecommandController를 통해 일반 사용자에게 찜목록에 등록된 매물을 토대로 다른 매물을 추천해준다
	/user/userInfo	UserInfoController를 통해 사용자의 개인 정보를 출력한다.
	/user/userInfo/update	UpdateUserController를 통해 개인 정보를 수정한다.
	/item/counsel/form	ReservationCounsel를 통해 상담예약창으로 이동하거나 상담을 예약한다.
매물검색 상세 내용 수정 삭제	/item/list	ListItemController를 통해 매물 리스트를 보여준다.
	/item/view	ViewItemController를 통해 매물의 상세 정보와 그 매물을 올린 중개인의 정보를 보여준다.
	/item/update	UpdateItemController를 통해 매물의 정보를 수정한다.
	/item/delete	DeleteItemController를 통해 매물을 삭제한다.
	/item/insert/form	매물 추가 폼인 /item/insertForm.jsp를 보여준다.
	/item/insert	InsertItemController를 통해 매물을 추가한다.
	/item/counsel/form	상담 폼인/item/counselForm.jsp를 보여준다.



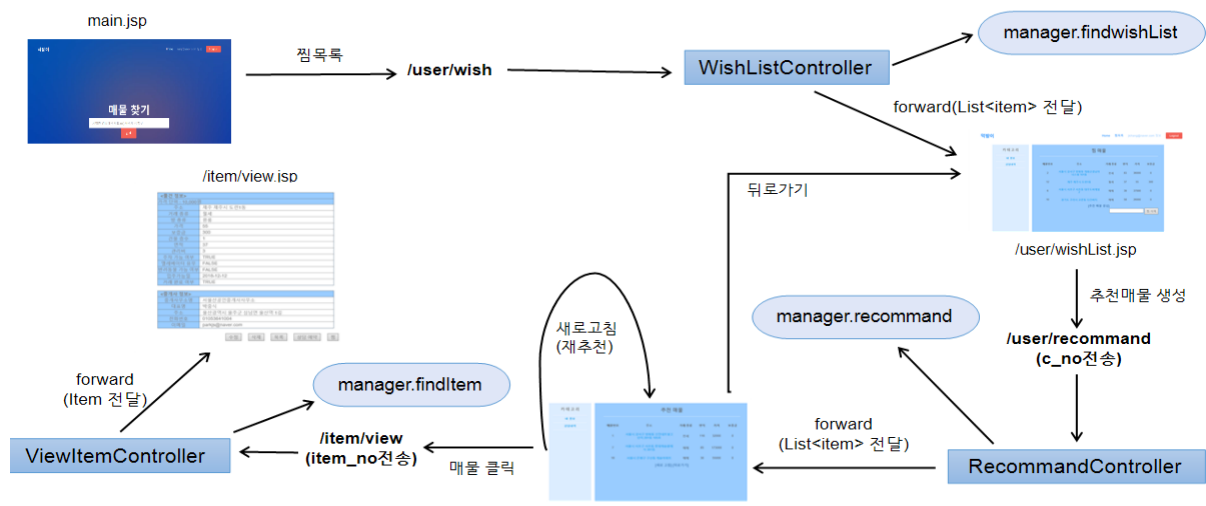
3. 사용자 정보 관리

메인에서 회원정보를 누를 시 /user/userInfo의 uri로 매핑 된 UserInfoController 에 의해 UserManager의 findUser를 이용하여 회원정보를 출력한다. 여기서 따로 넘겨주는 값은 없으며 UserInfoController에서 Session의 "who"객체를 이용하여 사용자 정보를 검색해낸다. who의 type에 따라 출력되는 사용자의 정보가 달리 나온다. userInfo.jsp에서 회원정보 변경을 클릭할 시 사용자가 입력한 비밀번호와 DB에 저장되어 있는 비밀번호를 비교하여 본인이 수정했음을 확인한다. 그 외에 사용자가 입력한 변경 전화번호, 주소, 비밀번호를 UpdateUserController로 전송하여 Manager의 update메소드를 이용하여 DB의 사용자 정보를 변경하고 메인 페이지로 이동하게 된다.



4. 찜 관리, 추천매물

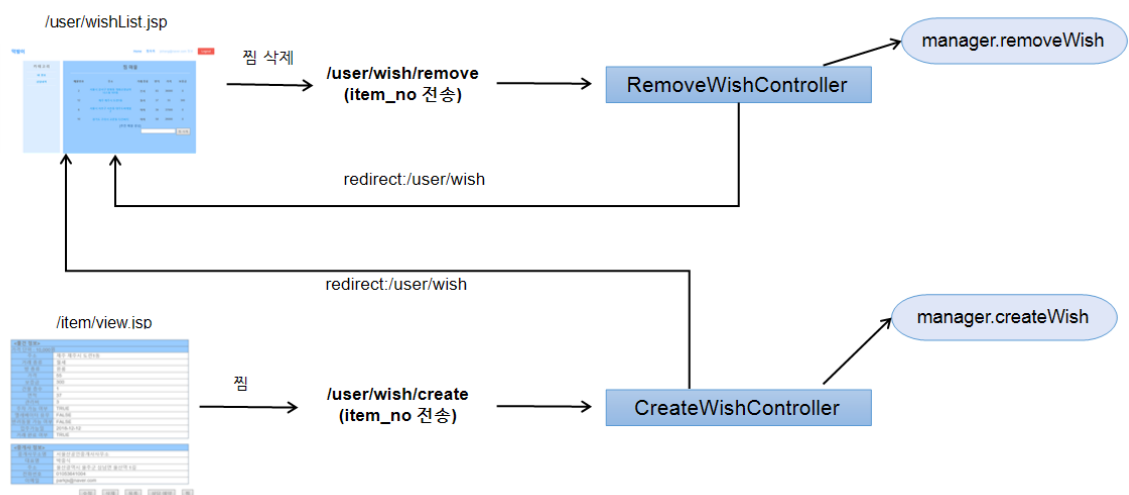
메인에서 찜목록을 누를 시 /user/wish uri로 매핑 된 WishListController에 의해 WishManager의 findWishList 메소드를 이용하여 List<item>을 /user/wishList.jsp로 forward 해 현재 로그인한 회원이 찜목록에 추가한 리스트를 출력한다. (Session의 "who"객체를 이용하여 현재 로그인한 회원 번호를 구별) 찜 목록에서 [추천목록생성]을 누를 시 RecommendController의 recommend 메소드를 이용해 recommend.jsp로 이동하면서 추천매물 로직에 따른 추천 매물을 출력한다 /user/recommend.jsp에서 새로 고침을 누를 시 위와 같은 검색 로직을 다시 실행하여 재검색한후 다시 리스트를 출력하여 보여준다. 혹은 매물의 주소를 클릭 시 ViewItemController의 findItem 메소드를 이용하여 /item/view.jsp로 이동하여 매물에 대한 상세정보를 볼 수 있도록 한다.



5. 찜 삭제, 추가

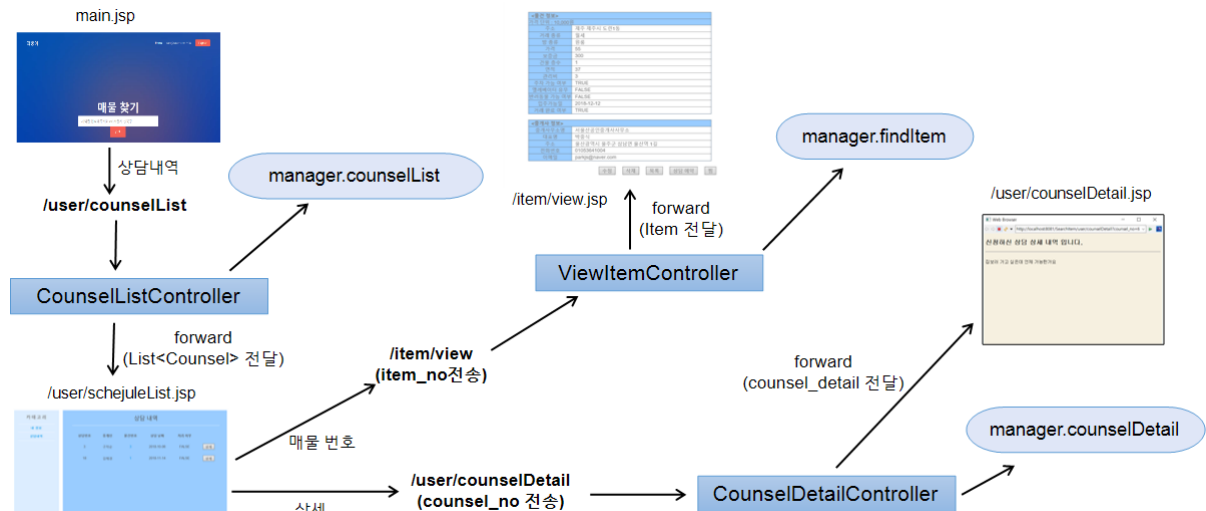
`/user/wishList.jsp`에서 입력창에 삭제한 매물번호를 입력 후 찜 삭제를 누를 경우 `/user/wish/remove`의 uri로 매핑 된 `RemoveWishController`에 의해 전달받은 `item_no`를 이용하여 `removeWish`메소드를 이용하여 찜 등록 된 매물을 삭제한 후 다시 `wishList.jsp`로 `redirect`하여 삭제된 결과를 보여주게 된다.

사용자가 매물의 상세페이지에서 마음에 들어 찜을 누르게 되면 `/user/wish/create` 의 uri로 매핑 된 `CreateWishController`에 의해 `item_no`를 전달받아 `createWish` 메소드를 이용해 찜목록에 매물을 생성한다. 단, 사용자의 찜 매물이 6개가 넘어가게 될 경우에는 등록되지 않도록 한다. (다시 삭제 후 등록가능) 생성된 매물을 확인 할 수 있도록 `redirect`하여 다시 `wishList.jsp`로 이동하게 된다.



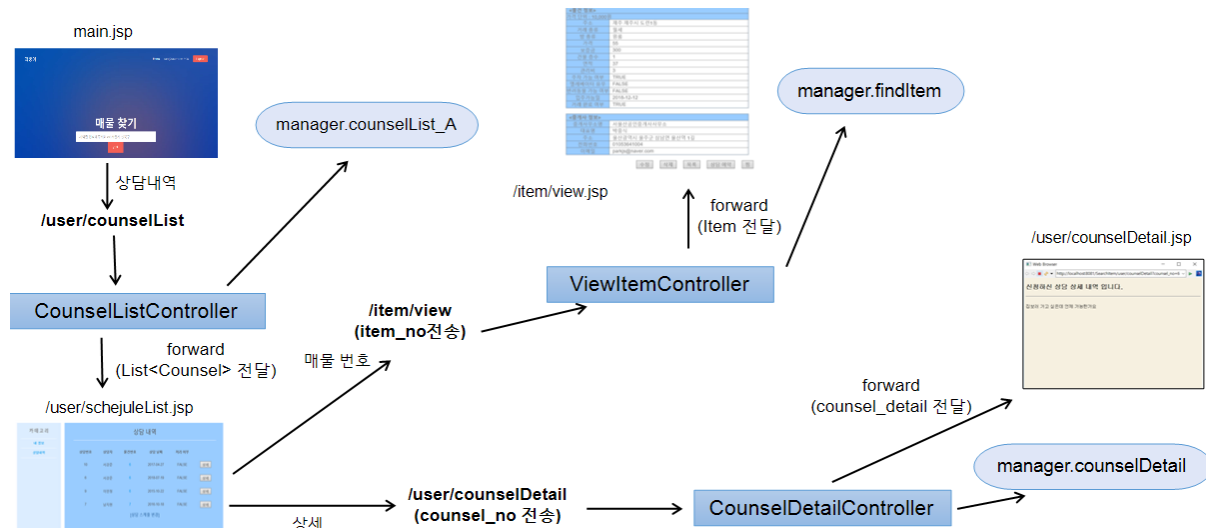
6. 상담내역 확인(개인회원)

메인에서 개인회원이 상담 내역을 누를 경우 `/user/counselList` uri에 매핑된 `CounselListController`에 의해 `CounselList`메소드를 사용해 중개인의 이름이 포함된 상담내역 리스트를 `/user/scheduleList.jsp`에서 출력한다. `scheduleList.jsp`에서 매물번호를 누를 시 `ViewItemController`의 `findItem` 메소드를 사용해 `/item/view.jsp`로 이동하여 매물에 대한 상세페이지를 보여주게 된다. `scheduleList.jsp`에서 상세를 누를 경우 `CounselDetailController`의 `counselDetail` 메소드에 의해 `counsel_detail`을 forward로 전달하여 상담의 상세내용을 보여주게 된다.



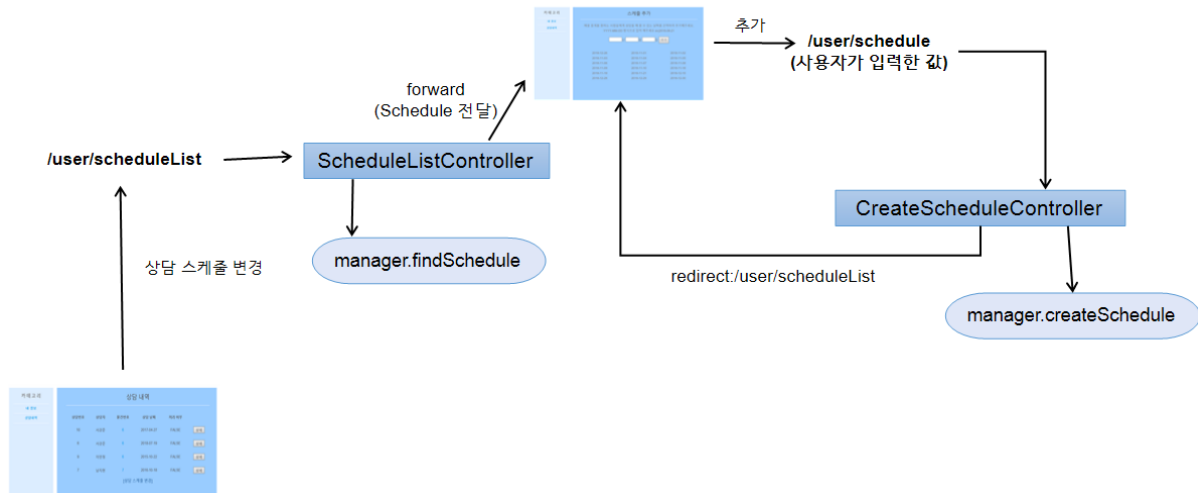
7. 상담내역 확인(중개인)

메인에서 중개인 회원이 상담내역을 누를 경우 개인과 동일한 컨트롤러에서 `CounselList_A`메소드를 이용해 상담요청한 개인의 이름이 포함된 상담내역 리스트를 `/user/scheduleList.jsp`에서 출력한다. 개인과 동일하게 매물 번호를 누를 시 해당하는 번호의 매물 상세 페이지를 출력한다. 상세를 누를 시에도 개인과 동일하게 매물번호에 해당되는 상담의 상세내용을 출력하게 된다.

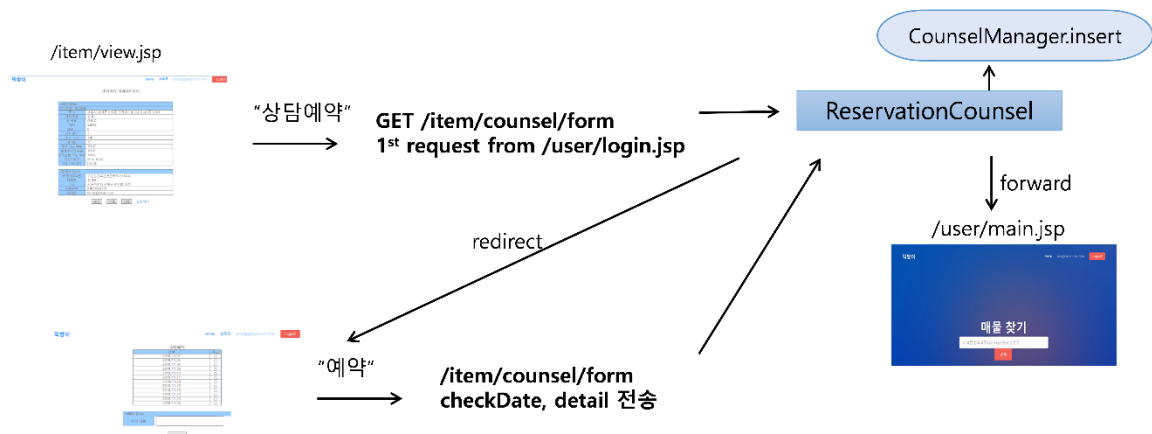


9. 상담 스케줄 출력, 추가 (중개인)

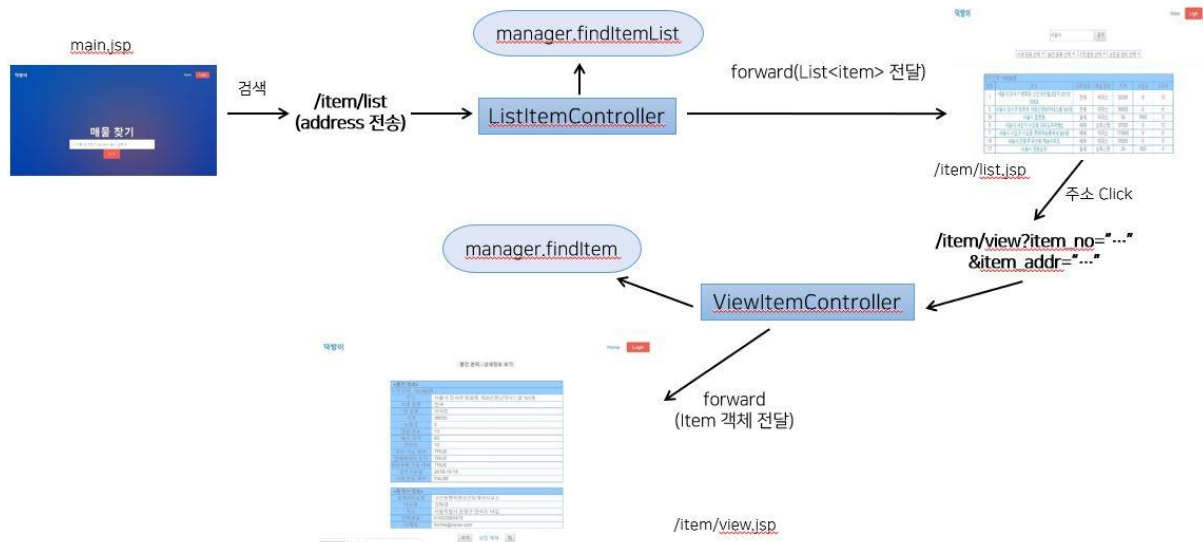
중개인의 경우 `/user/scheduleList.jsp`에서 상담 스케줄변경 이라는 하이퍼링크를 클릭할 수 있게 된다. 이 버튼을 누를 경우 `/user/scheduleList` uri에 매핑된 `ScheduleController`의 `findSchedule` 메소드를 사용해 자신이 등록한 개인회원에게 상담을 제공할 수 있는 날짜를 출력한다. 여기서 상담 가능한 날짜를 추가할 수 있다. 상담가능한 날짜를 YYYY-MM-DD 형식으로 작성한 후 추가를 누를 시 `createSchedule` 메소드에 의해 DB에 상담가능한 날짜가 추가되고 `/user/scheduleList.jsp`로 redirect되어 추가한 날짜를 포함한 상담가능한 스케줄을 표시하게 된다.



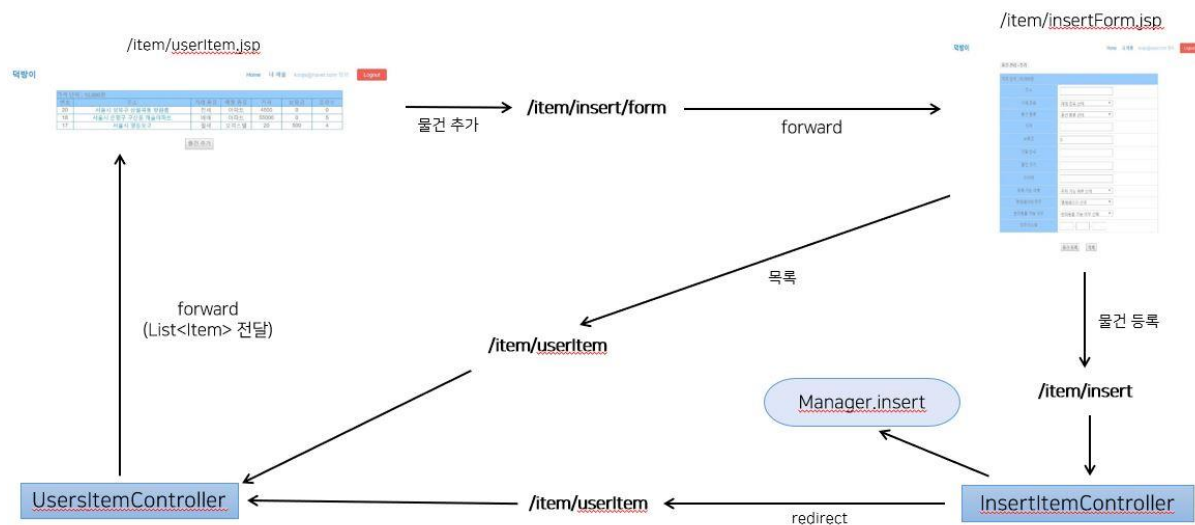
8. 상담예약: `view.jsp`(매물상세정보창)에서 상담예약버튼을 누르면 `ReservationCounsel`(컨트롤러)은 GET방식인 request임을 인지하고 상담예약창인 `counselForm.jsp`로 redirect한다. 상담을 예약할 날짜와 세부내용을 입력하고 예약버튼을 누르면 `ReservationCounsel`는 POST방식으로 온 request인 것 확인하고 `checkDate`와 `detail`를 받아 `CounselManager`를 통해 DB에 새로운 상담을 추가한다. 성공하면 메인화면으로 forward한다.



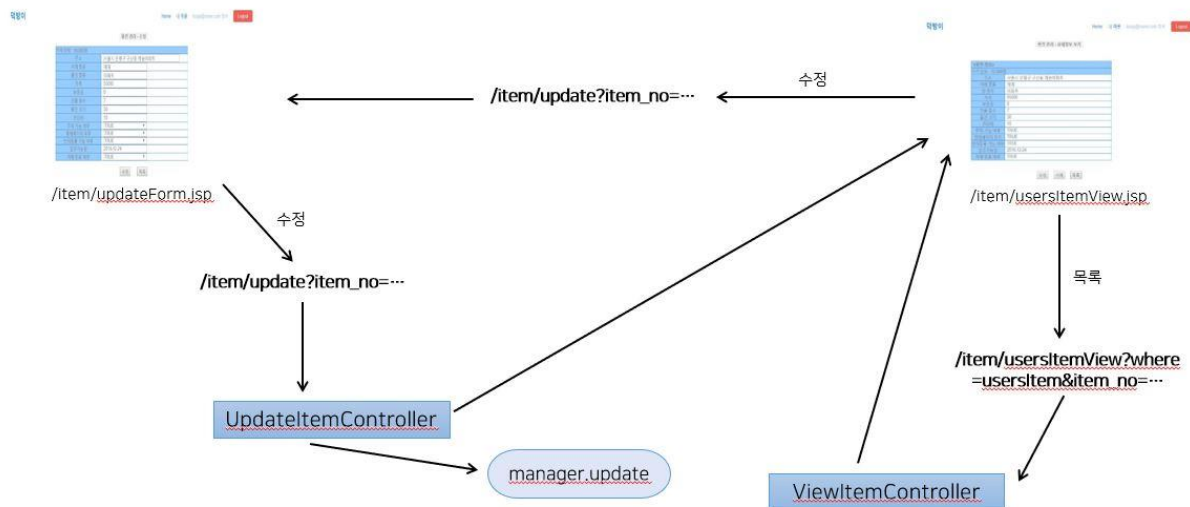
9. 매물 검색 : 메인 페이지의 검색 창에 검색할 주소를 적는다. -> ListItemController에서 입력한 주소가 포함된 매물 리스트들을 찾아서 /item/list.jsp에 전달한다. -> 거래 종류, 매물 종류, 가격 범위, 보증금 범위를 선택하여 조건을 걸어서 검색할 수 있다. -> 주소를 클릭하면 선택한 주소와 매물 번호가 ViewItemController 에 전달된다. -> ViewItemController에서는 전달받은 주소와 매물 번호로 매물 정보와 그 매물을 올린 중개인에 대한 정보를 검색해 /item/view.jsp에 전달한다. -> 전달 받은 매물과 중개인에 대한 정보를 보여준다. -> 상담 예약 버튼을 눌러 중개인과 해당 매물에 대해 상담을 신청할 수 있다.



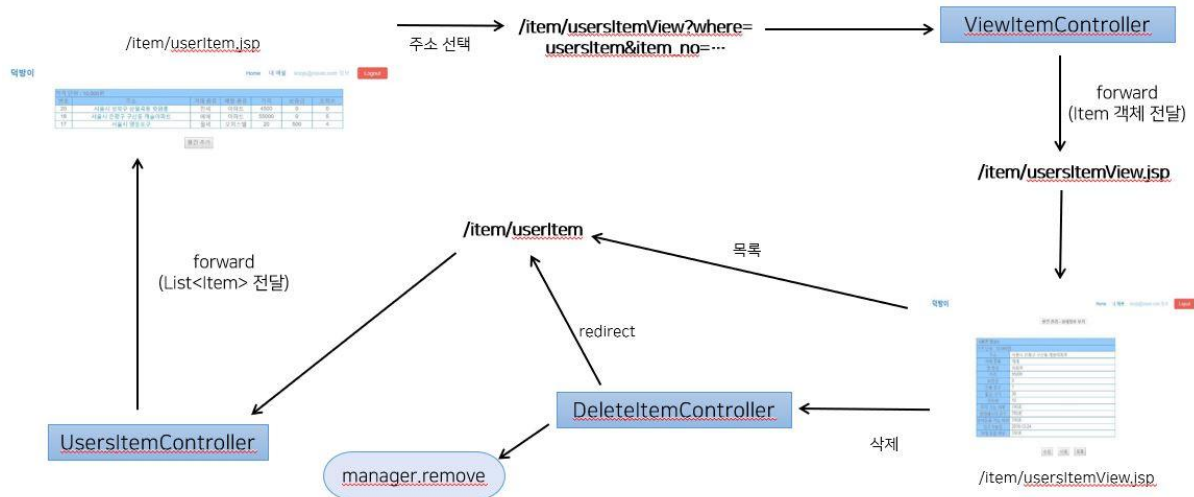
10. 중개사의 내 매물 검색 & 추가 : 중개사로 로그인 하고 상단 네비게이션 바의 '내 매물'을 누르면 중개사가 올린 매물 리스트를 보여준다. -> '물건 추가' 버튼을 눌러 원하는 매물의 정보를 적는다. -> '물건 등록' 버튼을 누르면 InsertItemController에 적은 정보가 request로 넘어간다. -> InsertItemController에서 새로 입력한 정보로 매물을 추가한다 -> ListItemController에서 전체 매물을 검색하고 결과로 나온 매물 리스트를 /item/list.jsp로 전달한다. -> 전달받은 매물 리스트를 이용해 매물을 보여준다.



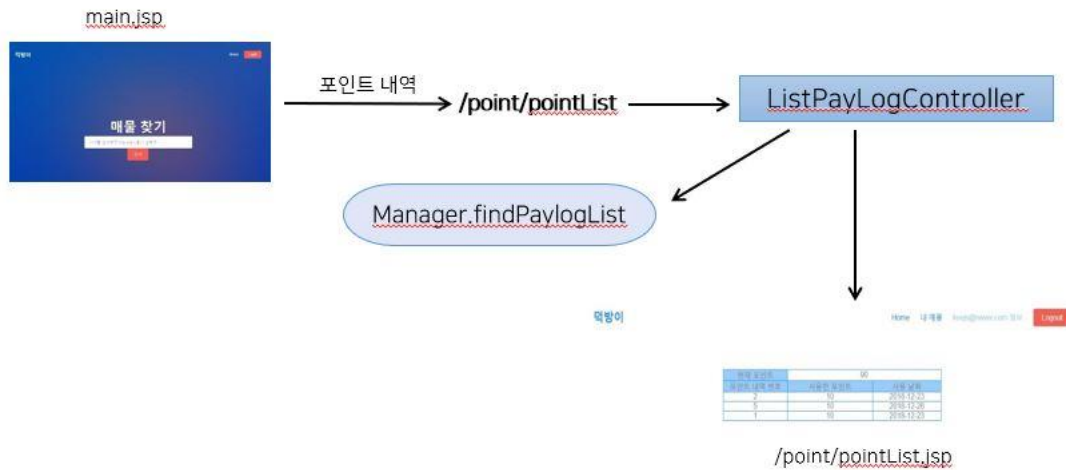
11. 매물 수정 : 중개인으로 로그인 하고 상단 네비게이션 바의 '내 매물'을 누르면 중개인이 올린 매물 리스트를 보여준다. -> 물건의 주소를 누르면 해당 매물 번호와 상태 정보를 ViewItemController에 전달한다. -> 전달 받은 매물 번호와 상태 정보로 검색하여 매물 정보를 /item/usersItemView.jsp에 전달한다. -> 전달 받은 매물 정보를 이용해 매물에 대한 정보를 보여 준다. -> '수정' 버튼을 누르면 UpdateItemController로 수정할 매물 번호가 전달된다. -> 전달받은 매물 번호에 해당하는 매물 정보를 검색해서 /item/update.jsp에 보여준다. -> 매물 정보를 수정하고 '수정' 버튼을 누르면 UpdateItemController로 수정한 매물 정보가 request로 전달된다. -> 전달된 매물 정보로 해당 매물의 정보를 수정하고 수정된 정보를 이용해 다시 /item/update 화면에 보여준다.



12. 매물 삭제 : 중개인으로 로그인 하고 상단 네비게이션 바의 '내 매물'을 누르면 중개인이 올린 매물 리스트를 보여준다. -> 물건의 주소를 누르면 해당 매물 번호와 상태 정보를 ViewItemController에 전달한다. -> 전달 받은 매물 번호와 상태 정보로 검색하여 매물 정보를 /item/usersItemView.jsp에 전달한다. -> 전달 받은 매물 정보를 이용해 매물에 대한 정보를 보여 준다. -> '삭제' 버튼을 누르면 DeleteItemController로 삭제할 매물 번호가 전달된다. -> 전달받은 매물 정보를 이용해 해당 매물을 삭제하고 UsersItemController로 이동한다. -> session에 있는 user 정보를 이용해 중개사가 올린 매물을 다시 검색하고 결과 리스트를 /item/usersItem.jsp로 전달한다. -> 전달받은 리스트를 이용해 화면에 정보를 보여준다.



13. 포인트 : 중개인으로 로그인 하고 상단 네비게이션 바의 '포인트 내역'을 누른다. -> `ListPayLogController`에서 session에 있는 user 정보를 이용해 중개사의 번호를 얻어내고, 해당 중개사 번호를 이용해 중개인이 매물을 등록할 때 기록되었던 `pay_log` 테이블을 검색하고, 그 결과 리스트를 `/point/pointList.jsp`로 전달한다. -> 전달 받은 리스트를 이용해 화면에 정보를 보여준다.



9. 구현방법

1) 개발 환경과 개발 기술/라이브러리

Erwin 프로그램을 사용하여 논리적, 물리적 모델링을 하여 데이터 베이스를 설계하였다. Oracle SQL Developer를 이용하여 데이터베이스를 구축하고 Eclipse 프로그램에서 Dynamic Web Project로 프로젝트를 진행하였다.

기술로는 Presentation Layer는 HTML/CSS, bootstrap, JavaScript, JSP을 사용, Business Layer는 Java를 사용, Persistence Layer에서는 SQL를 사용했다. 라이브러리는 JRW System Library와 Web App Libraries를 이용했다.

2) 비즈니스 로직

이 프로젝트에서는 두가지 비즈니스 로직을 사용했다.

첫번째 알고리즘은 매물 검색 알고리즘이다. 사용자가 검색 조건 등을 이용하여 매물을 아래의 로직으로 검색할 수 있다.

1.검색 값이 있을 경우

1.1 거래 종류를 선택할 경우

1.1.1 '월세' 선택

1.1.1.1 물건 종류를 선택할 경우

1.1.1.1.1 '아파트' 선택

1.1.1.1.1.1 가격 범위를 선택할 경우

1.1.1.1.1.1.1 ' ~ 30' 선택

[1]1.1.1.1.1.1.1.1 보증금 범위를 선택할 경우

1.1.1.1.1.1.1.1.1 ' ~ 5000' 선택

1.1.1.1.1.1.1.1.2 ' ~ 10000' 선택

1.1.1.1.1.1.1.1.3 ' ~ 15000' 선택

1.1.1.1.1.1.1.1.4 '15001 ~ ' 선택

1.1.1.1.1.1.1.2 보증금 범위를 선택하지 않을 경우

1.1.1.1.1.1.2 ' ~ 50' 선택

[1]

1.1.1.1.1.1.3 ' ~ 70' 선택

[1]

1.1.1.1.1.1.4 ' ~ 90' 선택

[1]

1.1.1.1.1.1.5 '91 ~ ' 선택

[1]

1.1.1.1.1.2 가격 범위를 선택하지 않을 경우

[1]

1.1.1.1.2 '오피스텔' 선택

1.1.1.1.2.1 가격 범위를 선택할 경우

1.1.1.1.2.1.1 ' ~ 30' 선택

[1]

1.1.1.1.2.1.2 ' ~ 50' 선택

[1]

1.1.1.1.2.1.3 ' ~ 70' 선택

[1]

1.1.1.1.2.1.4 ' ~ 90' 선택

[1]

1.1.1.1.2.1.5 '91 ~ ' 선택

[1]

1.1.1.1.2.2 가격 범위를 선택하지 않을 경우

[1]

1.1.1.1.3 '원룸' 선택

1.1.1.1.3.1 가격 범위를 선택할 경우

1.1.1.1.3.1.1 ' ~ 30' 선택

[1]

1.1.1.1.3.1.2 ' ~ 50' 선택

[1]

1.1.1.1.3.1.3 ' ~ 70' 선택

[1]

1.1.1.1.3.1.4 ' ~ 90' 선택

[1]

1.1.1.1.3.1.5 '91 ~ ' 선택

[1]

1.1.1.1.3.2 가격 범위를 선택하지 않을 경우

[1]

1.1.1.2 물건 종류를 선택하지 않을 경우

1.1.1.2.1.1 가격 범위를 선택할 경우

1.1.1.2.1.1.1 ' ~ 30' 선택

[1]

1.1.1.2.1.1.2 ' ~ 50' 선택

[1]

1.1.1.2.1.1.3 ' ~ 70' 선택

[1]

1.1.1.2.1.1.4 ' ~ 90' 선택

[1]

1.1.1.2.1.1.5 '91 ~ ' 선택

[1]

1.1.1.2.1.2 가격 범위를 선택하지 않을 경우

[1]

1.1.2 '전세' 선택

1.1.2.1 물건 종류를 선택한 경우

1.1.2.1.1 '아파트' 선택

[2]1.1.2.1.1.1 가격 범위를 선택한 경우

1.1.2.1.1.1.1 ' ~ 10000' 선택

1.1.2.1.1.1.2 ' ~ 20000' 선택

1.1.2.1.1.1.3 ' ~ 30000' 선택

1.1.2.1.1.1.4 ' ~ 40000' 선택

1.1.2.1.1.1.5 ' ~ 50000' 선택

1.1.2.1.1.1.6 '50001 ~ ' 선택

1.1.2.1.1.2 가격 범위를 선택하지 않을 경우

1.1.2.1.2 '오피스텔' 선택

[2]

1.1.2.1.3 '원룸' 선택

[2]

1.1.2.2 물건 종류를 선택하지 않을 경우

[2]

1.1.3 '매매' 선택

1.1.3.1 물건 종류를 선택할 경우

1.1.3.1.1 '아파트' 선택

[3]1.1.3.1.1.1 가격 범위를 선택한 경우

1.1.3.1.1.1.1 ' ~ 50000' 선택

1.1.3.1.1.1.2 ' ~ 100000' 선택

1.1.3.1.1.1.3 ' ~ 150000' 선택

1.1.3.1.1.1.4 '150001 ~ ' 선택

1.1.3.1.1.2 가격 범위를 선택하지 않을 경우

1.1.3.1.2 '오피스텔' 선택

[3]

1.1.3.1.3 '원룸' 선택

[3]

1.1.3.2 물건 종류를 선택하지 않을 경우

[3]

1.2 거래 종류를 선택하지 않을 경우

1.2.1 물건 종류를 선택할 경우

1.2.1.1 '아파트' 선택

1.2.1.2 '오피스텔' 선택

1.2.1.3 '원룸' 선택

1.2.2 물건 종류를 선택하지 않을 경우

2. 검색 값이 없을 경우

2.1 거래 종류를 선택할 경우

2.1.1 '월세' 선택

2.1.1.1 물건 종류를 선택할 경우

2.1.1.1.1 '아파트' 선택

2.1.1.1.1.1 가격 범위를 선택할 경우

2.1.1.1.1.1.1 ' ~ 30' 선택

[1]2.1.1.1.1.1.1 보증금 범위를 선택할 경우

2.1.1.1.1.1.1.1 ' ~ 5000' 선택

2.1.1.1.1.1.1.2 ' ~ 10000' 선택

2.1.1.1.1.1.1.3 ' ~ 15000' 선택

2.1.1.1.1.1.1.4 '15001 ~ ' 선택

2.1.1.1.1.1.2 보증금 범위를 선택하지 않을 경우

2.1.1.1.1.1.2 ' ~ 50' 선택

[1]

2.1.1.1.1.1.3 ' ~ 70' 선택

[1]

2.1.1.1.1.1.4 ' ~ 90' 선택

[1]

2.1.1.1.1.1.5 '91 ~ ' 선택

[1]

2.1.1.1.1.2 가격 범위를 선택하지 않을 경우

[1]

2.1.1.1.2 '오피스텔' 선택

2.1.1.1.2.1 가격 범위를 선택할 경우

2.1.1.1.2.1.1 ' ~ 30' 선택

[1]

2.1.1.1.2.1.2 ' ~ 50' 선택

[1]

2.1.1.1.2.1.3 ' ~ 70' 선택

[1]

2.1.1.1.2.1.4 ' ~ 90' 선택

[1]

2.1.1.1.2.1.5 '91 ~ ' 선택

[1]

2.1.1.1.2.2 가격 범위를 선택하지 않을 경우

[1]

2.1.1.1.3 '원룸' 선택

2.1.1.1.3.1 가격 범위를 선택할 경우

2.1.1.1.3.1.1 ' ~ 30' 선택

[1]

2.1.1.1.3.1.2 ' ~ 50' 선택

[1]

2.1.1.1.3.1.3 ' ~ 70' 선택

[1]

2.1.1.1.3.1.4 ' ~ 90' 선택

[1]

2.1.1.1.3.1.5 '91 ~ ' 선택

[1]

2.1.1.1.3.2 가격 범위를 선택하지 않을 경우

[1]

2.1.1.2 물건 종류를 선택하지 않을 경우

2.1.1.2.1.1 가격 범위를 선택할 경우

2.1.1.2.1.1.1 ' ~ 30' 선택

[1]

2.1.1.2.1.1.2 ' ~ 50' 선택

[1]

2.1.1.2.1.1.3 ' ~ 70' 선택

[1]

2.1.1.2.1.1.4 ' ~ 90' 선택

[1]

2.1.1.2.1.1.5 '91 ~ ' 선택

2.1.1.2.1.2 가격 범위를 선택하지 않을 경우

2.1.1.2.1.2.1 ' ~ 30' 선택

[1]

2.1.1.2.1.2.2 ' ~ 50' 선택

[1]

2.1.1.2.1.2.3 ' ~ 70' 선택

[1]

2.1.1.2.1.2.4 ' ~ 90' 선택

[1]

2.1.1.2.1.2.5 '91 ~ ' 선택

[1]

2.1.2 '전세' 선택

2.1.2.1 물건 종류를 선택한 경우

2.1.2.1.1 '아파트' 선택

[2]2.1.2.1.1.1 가격 범위를 선택한 경우

2.1.2.1.1.1.1 ' ~ 10000' 선택

- 2.1.2.1.1.1.2 ' ~ 20000' 선택
- 2.1.2.1.1.1.3 ' ~ 30000' 선택
- 2.1.2.1.1.1.4 ' ~ 40000' 선택
- 2.1.2.1.1.1.5 ' ~ 50000' 선택
- 2.1.2.1.1.1.6 '50001 ~ ' 선택
- 2.1.2.1.1.2 가격 범위를 선택하지 않을 경우
- 2.1.2.1.2 '오피스텔' 선택
 - [2]
 - 2.1.2.1.3 '원룸' 선택
 - [2]
 - 2.1.2.2 물건 종류를 선택하지 않을 경우
 - [2]
- 2.1.3 '매매' 선택
 - 2.1.3.1 물건 종류를 선택할 경우
 - 2.1.3.1.1 '아파트' 선택
 - [3]2.1.3.1.1.1 가격 범위를 선택한 경우
 - 2.1.3.1.1.1.1 ' ~ 50000' 선택
 - 2.1.3.1.1.1.2 ' ~ 100000' 선택
 - 2.1.3.1.1.1.3 ' ~ 150000' 선택
 - 2.1.3.1.1.1.4 '150001 ~ ' 선택
 - 2.1.3.1.1.2 가격 범위를 선택하지 않을 경우
 - 2.1.3.1.2 '오피스텔' 선택
 - [3]
 - 2.1.3.1.3 '원룸' 선택
 - [3]
 - 2.1.3.2 물건 종류를 선택하지 않을 경우
 - [3]
- 2.2 거래 종류를 선택하지 않을 경우
 - 2.2.1 물건 종류를 선택할 경우
 - 2.2.1.1 '아파트' 선택
 - 2.2.1.2 '오피스텔' 선택
 - 2.2.1.3 '원룸' 선택
 - 2.2.2 물건 종류를 선택하지 않을 경우

두번째 알고리즘은 유사매물 추천 알고리즘이다. 개인이 찜 하기 기능을 이용해 찜 한 매물을 아래의 로직으로 유사매물을 추천해준다. 현재 개인회원의 찜목록에 등록 되어있는 매물의 주소 앞글자(~시,~도) 까지를 구별, 만약 같은 ~시가 있다면 그에 해당하는 ~시 까지의 매물, 혹은 겹치는 매물이 없다면 랜덤 한 ~시의 매물, 그리고 가장 많은 거래종류(월,전세,매매)의 매물, 혹은 같은 거래종류의 매물 개수가 같을 경우에는 같은 거래종류의 매물 중 랜덤 한 거래종류의 매물 ,

그리고 선택된 거래종류가 월세 일시 보증금+-30%, 가격+- 30%, 매물의 면적 +- 30% 조건에 해당하는 가격을 가진 매물, 혹은 선택된 거래종류가 전세,매매 일시, 가격+-30%, 매물의 면적 +- 30% 조건에 해당하는 가격의 매물을 출력한다. 선택된 매물이 5개가 넘을 경우 랜덤하게 5개만 선택하여 출력한다.

1. 찜 한 내용이 있을 경우

- 1.1 찜 한 매물들의 방 종류(아파트,빌라,원룸)로 전체 매물에서 나눈다.
- 1.2 찜 한 매물들의 주소(구)로 매물을 나눈다
- 1.3 찜 한 매물들의 거래종류(매매,전세,월세)로 나눈다
- 1.4 찜 한 매물들의 가격으로 나눈다.
 - 1.4.1 월세의 경우, 월세와 보증금을 따로 평균내서 $\text{평균값} \pm (10\% * \text{평균값})$ 의 범위 사용
 - 1.4.2 전세 또는 매매의 경우, 금액을 평균을 내고 $\text{평균값} \pm (10\% * \text{평균값})$ 의 범위 사용
- 1.5 유사매물로 나온 매물들을 유사추천 테이블에 저장한다.
- 1.6 유사매물 중 5개를 랜덤으로 보여준다.
 - 1.6.1 유사매물의 개수가 5개 이하이면 '활성화 버튼' 동작하지 않도록 한다.
 - 1.6.2 유사매물의 개수가 5개 초과이면 '활성화 버튼' 동작하도록 한다.
 - 1.6.2.1 '활성화 버튼'을 누를 경우 유사매물추천 테이블에서 랜덤 5개를 보여준다.

2. 찜 한 내용이 없을 경우

- 2.1 유사매물 추천 창이 없고 새로 고침 아이콘 비활성화
- 2.2 매물을 찜 하라는 설명

10. 결론 및 소감

1) 결론

직접 부동산에 가지 않아도 인터넷으로 원하는 정보를 미리 검색해 볼 수 있고, 물건을 찜하거나 상담예약을 할 수 있는 부동산 웹사이트이다. 사용자 (개인, 중개인)에게 필요한 기능(검색, 찜 관리, 매물관리)들을 메인 페이지에서 바로 들어갈 수 있도록 간단하고 사용하기 쉽게 만들었다. 또한 검색과 유사매물추천에서 우리만의 알고리즘을 만들었다. Java, JavaScript, HTML/CSS, SQL, JSP, Bootstrap을 이용했다. 이 프로젝트를 마치면서 우리가 아쉬웠던 점은 우리가 예상하지 못한 사용자의 요청이나 경로로 들어오는 것에 대한 제어가 부족했고 모든 화면에 포함되는 nav 코드를 하나의 파일로 만들지 못하고 계속 반복했고 UI디자인의 완성도가 높지않는 점이다.

2) 소감

강은영 : 이 수업에서 무엇을 먼저 계획하고 어떻게 단계를 나아가야하는 지 배웠고 앞으로 다른 프로젝트를 진행한다면 수업에서 배운 것들이 많은 도움이 될 것이다. 또한 프로젝트를 팀으로 하면서 역할 분담을 하고 의견을 조율하고 그것을 하나로 합치는데 오랜 시간이 걸리고 어려운 일이라는 것을 느꼈다. 그 과정에서 어떤 식으로 풀어갈지 고민하고 함께 해결하면서 스스로 더 발전했다고 생각한다. 처음에 기획했던 대로 기능과 다른 것들을 구현하지는 못한 점이 아쉬웠지만 기획부터 개발까지 모든 과정에 참여하면서 어떤 부분에서 어려움이 있었고 어떤 방식이 더 나은지 찾아보고 공부하게 되는 프로젝트였다.

여지은 : 우선 학교내에서 교외활동을 하지 않고도 팀 프로젝트를 진행하면서 실제개발을 경험해보는 기회가 생겨서 좋았으며 매주 수업에 맞추어 프로젝트를 단계적으로 진행 함으로써 그 주에 배운 것을 토대로 큰 어려움없이 개발이 가능했다는 점이 좋았다. 일주일에 최소 두번씩은 만나면서 회의하고 각자 맡은 일을 완수해야 하는 것이 귀찮고 힘들기도 했지만 책임감을 가지고 하나씩 기능들을 완성해가고 결과를 확인해보면서 성취감과 보람을 느낄 수 있었다. 또한 팀을 이뤄 활동하는 것에 팀워크와 분업이 얼마나 중요한지 알게 되었고 앞으로 있을 팀 프로젝트에서는 더 잘할 수 있을 것이라고 생각한다.

송혜진 : 기획할 프로젝트의 아이디어를 생각하는 것부터 시작하여 화면 디자인과 코드 구현까지 해봄으로써 웹에 대한 이해를 높이고 실전 경험을 얻을 수 있었던 팀 프로젝트였다. 팀 프로젝트를 하면서 팀원들과 의견을 조율하고, 같이 코드를 구현하는 법과 bootstrap을 이용하는 법 등 다양한 것을 배웠고 그 결과로 나온 프로젝트는 꽤 괜찮은 웹이라고 생각한다. 3달 정도의 팀 프로젝트 기간이 주어졌고, 매 주 수업에 배운 것을 기반으로 프로젝트를 구현해 나간다는 점은 좋았다. 하지만 실제 구현한 기간은 한 달 남짓한 시간이어서 구현할 수 있는 기간이 짧았다는 점은 아쉬웠다. 이번 과목에서 MVC 구조를 바탕으로 웹을 구현해봤으니, 4학년 때는 Spring관련 과목을 수강해서 Spring을 이용해 좀 더 간편하게 웹을 구현하는 방법을 배우고 싶다.

11. 역할분담

강은영(팀장) – 일정관리, 수행일지 작성, 자료 종합/정리/업로드, 교수님께 질문 및 피드백
매물관련 데이터베이스 설계 및 구축, 로그인/로그아웃, 회원가입, 상담예약 기능에 해당하는 model, view, controller 설계 및 구현

송혜진(팀원) – 회원,개인,중개사 관련 데이터베이스 설계 및 구축, 매물 등록/수정/삭제/검색/필터링, 포인트 차감, 검색 기능에 해당하는 model, view, controller 구현, 네비게이션 바 관련 코드 구현, 교수님께 질문 및 피드백

여지은(팀원) – 찜, 상담관련 데이터 베이스 설계 및 구축, 찜목록관리(추가,삭제,출력), 추천매물(생성,출력), 사용자정보관리(조회,수정), 상담관리(상담내역출력, 중개인 스케줄 입력) 기능에 해당하는 model, view, controller 구현