목차

[1. 개요 3](#_Toc104467923)

[2. Use Case 3](#_Toc104467924)

[2.1. Use Case 명세서 3](#_Toc104467925)

[2.2. Use Case Diagram 4](#_Toc104467926)

[3. Entity Relationship Diagram 4](#_Toc104467927)

[4. Class Diagram 5](#_Toc104467928)

[4.1. BJDApiController 5](#_Toc104467929)

[4.2. BoardController 6](#_Toc104467930)

[4.3. HomeRestController 6](#_Toc104467931)

[4.4. MarketRestController 7](#_Toc104467932)

[4.5. NoticeController 8](#_Toc104467933)

[4.6. StatisticsRestController 9](#_Toc104467934)

[4.7. ExceptionRestControllerAdvice, FileDownLoadView 9](#_Toc104467935)

[4.8. UserInfoController 10](#_Toc104467936)

[5. 화면 설계 10](#_Toc104467937)

[6. 부록 11](#_Toc104467938)

[6.1. 로그인 동작 시퀀스 다이어그램 11](#_Toc104467939)

[6.2. 인가 과정 시퀀스 다이어그램 11](#_Toc104467940)

[6.3. Use Case Diagram 12](#_Toc104467941)

# 개요

본 문서는 SSAFY 7기 1학기 마지막 관통프로젝트 Happy house의 설계서입니다. Use Case 명세, ERD, 클래스 다이어그램, 화면 설계에 대한 내용을 포함하고 있습니다.

# Use Case

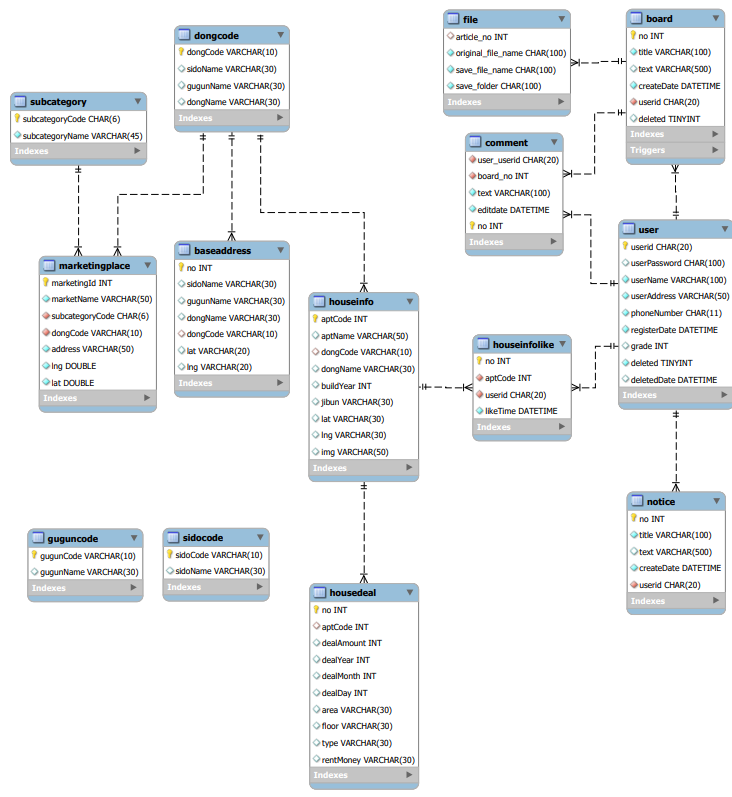
# Use Case 명세서

|  |  |  |
| --- | --- | --- |
| 유즈케이스 이름 | 설명 | 사전조건 |
| 1.로그인 | 사용자가 ID와 Password를 이용해 로그인한다. | X |
| 2.회원가입 | 사용자가 ID,Password,이름,주소,전화번호 정보를 입력하여 회원 가입을 한다. | X |
| 3.회원정보 수정 | 사용자는 이름, 주소, 휴대폰 정보를 변경할 수 있다. | 1.로그인 |
| 4.회원정보 탈퇴 | 사용자가 회원 탈퇴한다. | 1.회원탈퇴 |
| 5.회원정보 상세보기 | 사용자가 자신의 정보를 볼 수 있다. | 1.로그인 |
| 6.지역별 아파트 매물검색 | 사용자가 시/도,구/군,동을 선택하여 해당 지역의 아파트 매물 정보를 본다. 지도에 해당 아파트의 위치를 확인할 수 있고, 리스트 또한 확인할 수 있다. | 1.로그인 |
| 7.매물 상세정보 조회 | 사용자가 원하는 매물에 대한 상세정보를 확인할 수 있다. | 1.로그인 |
| 8.관심 매물 등록 | 사용자가 원하는 매물에 대해 관심 등록을 할 수 있다. | 1.로그인 |
| 9.관심 매물 등록 취소 | 사용자가 관심등록한 매물을 취소할 수 있다. | 1.로그인 |
| 10.관심매물 모아보기 | 사용자가 관심 등록한 매물에 대해서 조회할 수 있다. | 1.로그인 |
| 11. 상권 정보 확인 | 사용자가 매물별 상세정보 조회를 했을 경우, 해당 위치에 대한 상권 정보를 확인할 수 있다. | 1.로그인,2.매물 상세정보 조회 |
| 12. 최고 거래액 순 차트 조회 | 사용자가 최고 거래액을 기준 상위 10건에 대해 보여지는 차트를 확인할 수 있다. | X |
| 13. 관심 매물 등록 수 순 차트 조회 | 사용자가 관심 등록이 높은 순으로 상위 10건에 대해 차트를 확인할 수 있다. | X |
| 14. 거래량 순 차트 조회 | 사용자가 거래량 기준으로 상위 10건에 대해 보여지는 차트를 확인할 수 있다. | X |
| 15. 대기환경지수 조회 | 사용자가 시/도별 대기 환경 지수에 대해 알 수 있다. | X |
| 16. 공지사항 등록 | 관리자 권한(Admin)을 가진 사용자가 공지사항을 등록할 수 있다. | 1.로그인(관리자 권한) |
| 17. 공지사항 수정 | 관리자 권한(Admin)을 가진 사용자가 자신이 등록한 공지사항을 수정할 수 있다. | 1.로그인(관리자 권한) |
| 18. 공지사항 삭제 | 관리자 권한을 가진 사용자가 자신이 등록한 공지 사항을 삭제한다. | 1.로그인 (관리자 권한) |
| 19.공지 사항 상세보기 | 사용자가 공지상을 확인할 수 있다. | X |
| 20. 게시물 등록 | 사용자가 제목,내용,파일 요소들이 포함된 게시글을 등록할 수 있다. | 1.로그인 |
| 21. 게시물 상세보기 | 사용자는 게시글 검색으로 나온 결과에 대해 상세보기를 할 수 있다. 게시글에 업로드 되어 있는 파일을 다운로드 할 수 있다. | 1.로그인 |
| 22. 게시물 수정 | 사용자는 자신의 게시글을 수정할 수 있다. | 1.로그인 |
| 23. 게시물 삭제 | 사용자는 자신의 게시글을 삭제할 수 있다. | 1.로그인 |
| 24. 게시판 댓글 등록 | 사용자는 게시글에 자신의 댓글을 작성할 수 잇다. | 1.로그인 |
| 25. 게시판 댓글 삭제 | 사용자는 게시글에 등록한 자신의 댓글을 삭제할 수 있다. | 1.로그인 |
| 26.게시글 검색 | 사용자는 게시글 검색 시 , 게시물 번호,제목,유저 이름을 통해 검색을 할 수 있다. | 1.로그인 |

# Use Case Diagram

기능 별로 분류하여 구분해 놓았다. (게시물, 공지사항), (회원), (통계, 대기 정보), (매물)로 분류하였고 이는 부록에서 볼 수 있다.

# Entity Relationship Diagram

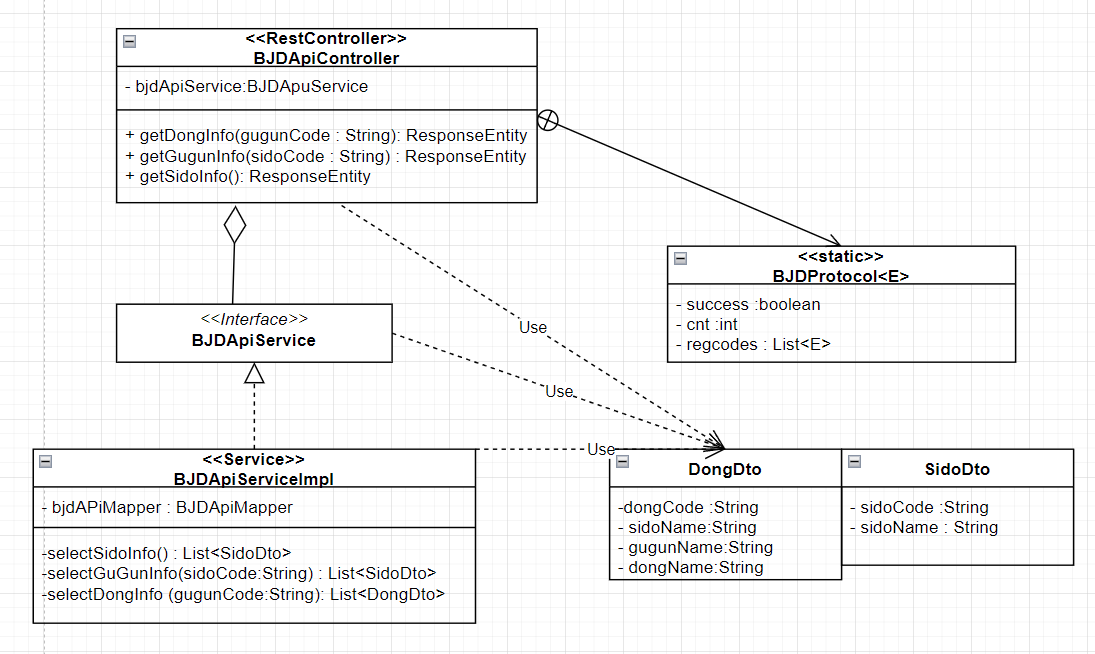


회원 탈퇴 된 유저는 deleted가 1이된다. 그리고 탈퇴 된 날짜가 deletedDate에 기록된다. Grade가 0이면 관리자, 1이면 일반 사용자이다. Comment(댓글)는 board와 user의 Primary key를 참조하고 있다. 또한 하나의 게시글에 여러 개의 댓글을 쓸 수 있음으로 따로 Primary key(no)를 가진다. houseInfolike는 사용자가 관심 등록한 아파트 매물을 기록하기 위해 사용된다. 하나의 houseInfo 레코드에는 여러 개(OneToMany)의 houseDeal(거래 내역)이 있다. Marketingplace, houseInfo는 dongcode 테이블의 주 키를 참조하고 있다. Baseaddress table은 사용되지 않지만 확장을 이해 놔두었다. Marketingplace에서 subcategory table을 참조하는데 이는 업종 명 code에 따라 업종명을 얻기 위해서이다. Guguncode와 sidocode 테이블은 법정동을 제공하기 위해 사용된다.

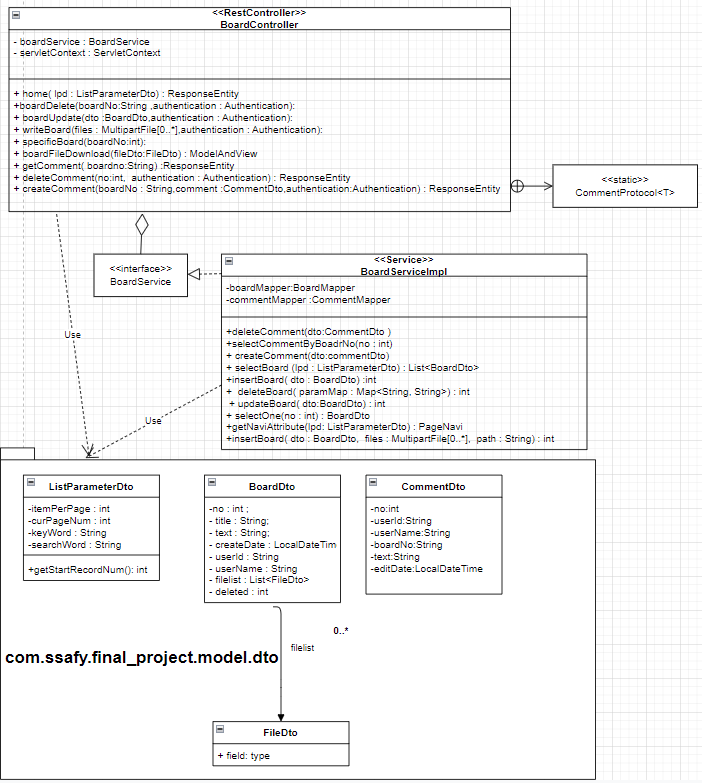
# Class Diagram

SSAFY 1학기 마지막 관통 프로젝트의 클래스 다이어그램을 설명하기 위해서 작성되었습니다. 사용자의 요청을 받는 컨트롤러 중심적으로 기술되어있으며, Throws Exception, Mapper등이 생략되어 있습니다.

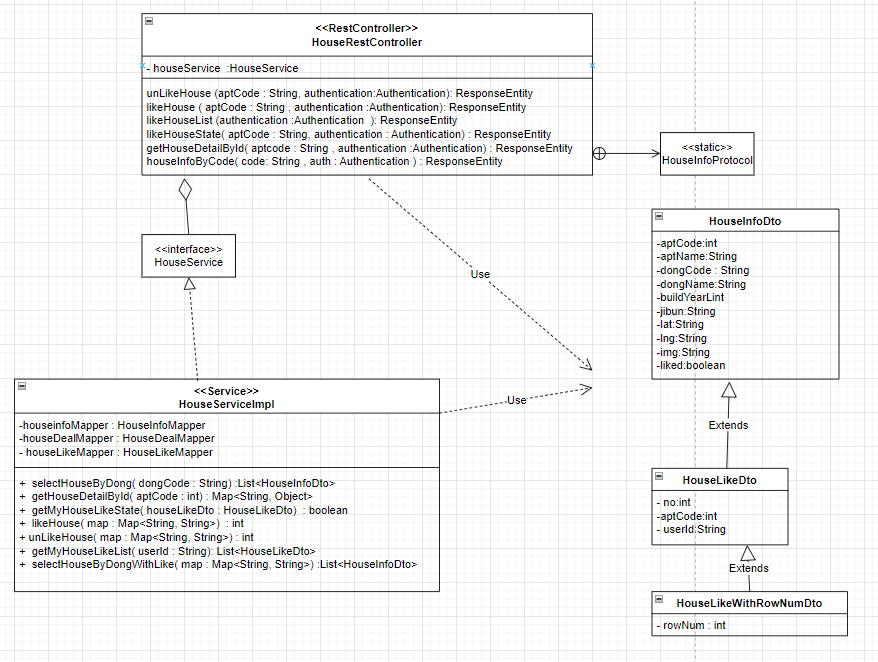
# BJDApiController

  
BJDApiController 는 시,구/군,구 코드 검색을 위해 사용됩니다. getSidoInfo 에서 요청을 받을 시 시의 이름과 code를 반환합니다. getGugunInfo의 경우 시코드를 받아 그 하위의 구/군 이름과 코드를 반화하고 DongCode는 gugunCode를 받아 같은 처리를 진행합니다. Service 스테레오 타입의 경우 Mybatis를 통해 Mysql에 접근하여 데이터를 얻어옵니다. BJDProtocol은 사용자에게 정형화된 데이터를 반환하기 위해 사용됩니다. 위 클래스들은 모두 DongDto와 SidoDto를 사용하게 되는데, DongDto의 경우 GuGunDto를 만들지 않고 처리하기 위해 필드에 구/군에 대한 필드가 들어가 있습니다.

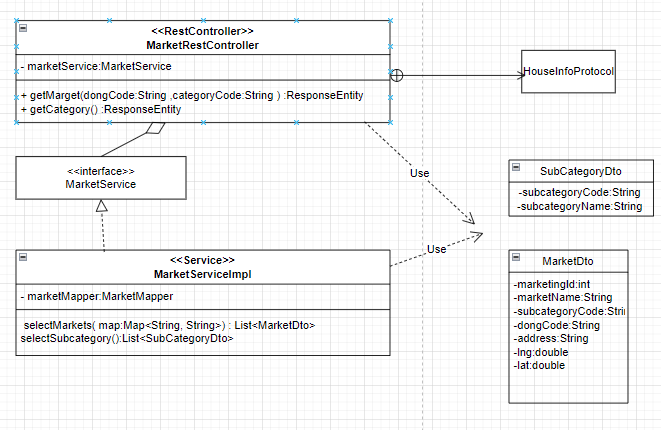
# BoardController

  
BoardController는 게시판 이용시 사용하게 되는 RestController 입니다. home 메서드의 경우 ListParameterDto를 받아 게시판 리스트를 반환합니다. ListParameterDto는 페이징을 위하여 구현된 DTO입니다. 그 외에도 게시판 삭제, 게시판 수정, 파일 업로드를 포함한 게시글 작성, 게시판 상세보기, 게시판의 댓글 가져오기, 삭제를 지원합니다. CommentProtocol은 댓글 리스트를 응답하는데 사용됩니다. BoardDto의 경우 FileDto를 리스트로 가집니다. 이는 하나의 게시판에 여러 개의 파일을 등록할 수 있음을 알 수 있습니다.

# HomeRestController

  
HouseRestController는 본 프로젝트의 핵심인 아파트 매물 정보 제공에 이용됩니다. 사용자가 보내준 코드에 맞게 아파트 매물 정보를 반환하며 해당 아파트 상세보기, 관심 설정하기,해제하기, 관심을 설정한 아파트 리스트 가져오기를 지원합니다. DTO의 경우 특이한 점을 볼 수 있습니다. HouseInfoDto의 경우 아파트의 정보를 나타내기 위해 사용되었는데, 개발 중 변경이 생겨 liked필드를 추가하였습니다. HouseLikeDto의 경우 사용자가 관심 설정한 아파트 내역을 볼 때 사용되며, HouseLikeWithRowNumDto의 경우 통계 정보 이용에 사용됩니다.

# MarketRestController

  
MarketRestController의 경우 상권 정보를 읽어오기 위해 사용된다. donCode를 받아 해당 동에 있는 상권정보를 넘겨준다. subcategoryCode의 경우 required= false이며, 넘겨 줄 시 해당 업종의 정보만 넘겨준다. getCategory의 경우 요청 시 업종의 정보를 넘겨준다.

# NoticeController

테이블이(가) 표시된 사진

자동 생성된 설명  
NoticeController의 경우 공지사항 관련 기능을 제공한다. 공지사항 작성,삭제,수정,조회를 할 수 있다.

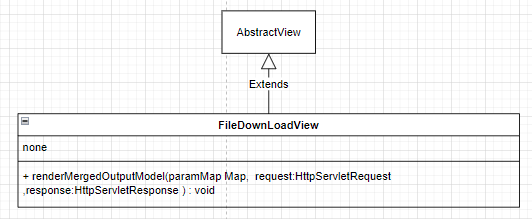
# StatisticsRestController

텍스트, 지도, 하늘, 실내이(가) 표시된 사진

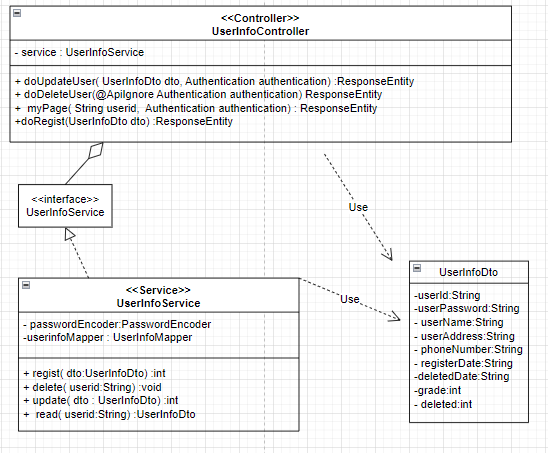
자동 생성된 설명  
statisticsRestController는 통계 정보를 반환하는데 사용된다. 아파트 거래가 상위 10, 관심 아파트 상위 10, 거래 횟수 최대 상위 10을 응답한다. 각 주제마다 다른 DTO를 사용한다. DustService는 공공 API에서 대기 환경 정보를 받아와서 컨트롤러에 넘겨주는 역할을 한다. 시도별 대기환경 정보는 500건 조회로 매우 적은 편이다. 따라서 DustUtilProxy을 사용하여 같은 시,도이고 같은 시각(분 이하 제외) 이면 저장되어 있는 대기환경 정보를 반환한다.

# ExceptionRestControllerAdvice, FileDownLoadView

텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명  
  
ExceptionRestControllerAdvice는 RestController하위에서 발생하는 오류에 대응하기 위한 기능을한다. FileDownLoadView의 경우, 사용자의 파일다운로드를 처리해가 위해 사용된다. BoardController에서 FileDownLoadView로 요청을 넘겨주면 response 헤더를 조작해서 서버 컴퓨터에 저장되어 있는 파일을 넘겨준다.

# UserInfoController

  
userInfoController는 회원가입, 사용자 조회, 탈퇴, 수정에 이용된다. UserInfoService는 회원가입 시 받은 password를 암호화하여 저장하기 위해 passwordEnoder를 필드로 가지고 있다. 탈퇴 시 실제로 삭제하는 것이 아니라 값만 바꾼다. Login의 경우 spring security 쪽에서 처리한다.

# 화면 설계

화면 설계에는 ovenapp을 사용하였다. 초기 디자인은 [Main (1 of 11) (ovenapp.io)](https://ovenapp.io/view/iAQens7qJMXmDC7AzycvF4HFAgto9Zmh/uYyJI)에서 확인할 수 있다.

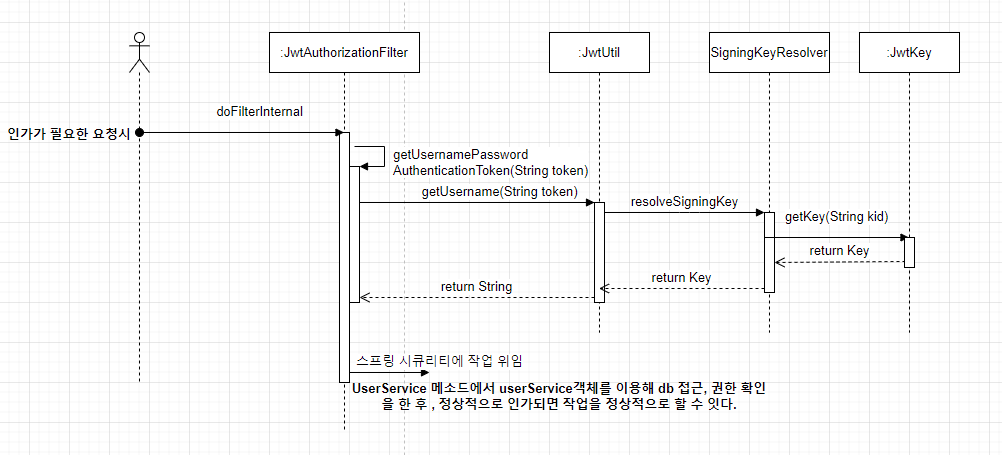
# 부록

# 로그인 동작 시퀀스 다이어그램

테이블이(가) 표시된 사진

자동 생성된 설명  
사용자가 로그인을 요청할 시, spring security 필터에 추가한 JwtAuthenticationFileter 메서드가 호출된다. 1.해당 메서드에서는 요청으로부터 사용자 id와 password를 읽어 스프링 security에서 사용할 수 있는 객체(authentication)를 생성한 후 등록한다. 2.userDetailService는 authentication에서 id를 읽어와 데이터 베이스에서 사용자 레코드를 읽어오고 UserDetail객체를 생성하여 반환한다. 그러면 spring security에서는 사용자 요청에서 넘어온 password를 암호화하고 데이터 베이스에서 넘어온 password 와 일치하는지 검사한다. 인증이 성공 시 id와 key를 통해 토큰을 생성하고 사용자에게 넘겨준다.

# 인가 과정 시퀀스 다이어그램

  
위 내용은 success케이스에 대해서만 표현되어 있다. JWT 토큰으로부터 userId를 가져와서 데이터 베이스에서 검색, 존재 여부를 확인하고 데이터 베이스에 저장된 기록에 의해 권한을 부여한다.

# Use Case Diagram

Figure 1게시글, 공지사항 관련

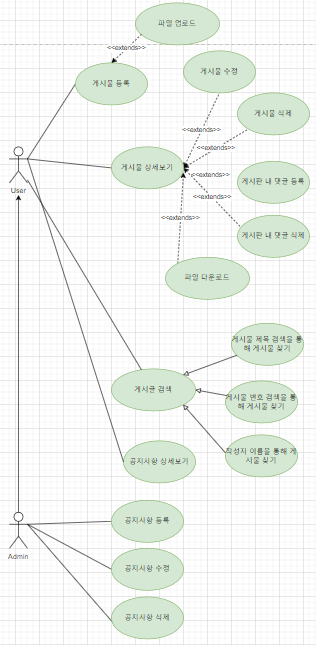


Figure 2 아파트 매물 관련

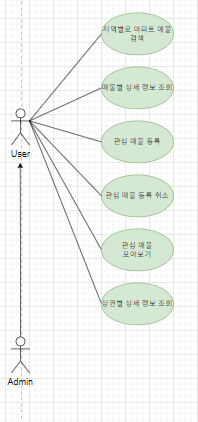


Figure 3 통계, 대기 환경정보 관련

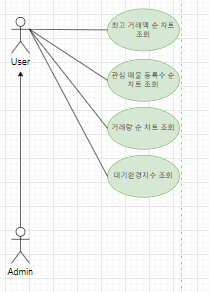
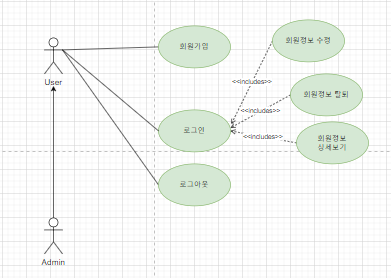


Figure 4 회원 관련



# 전체 클래스 다이어그램

