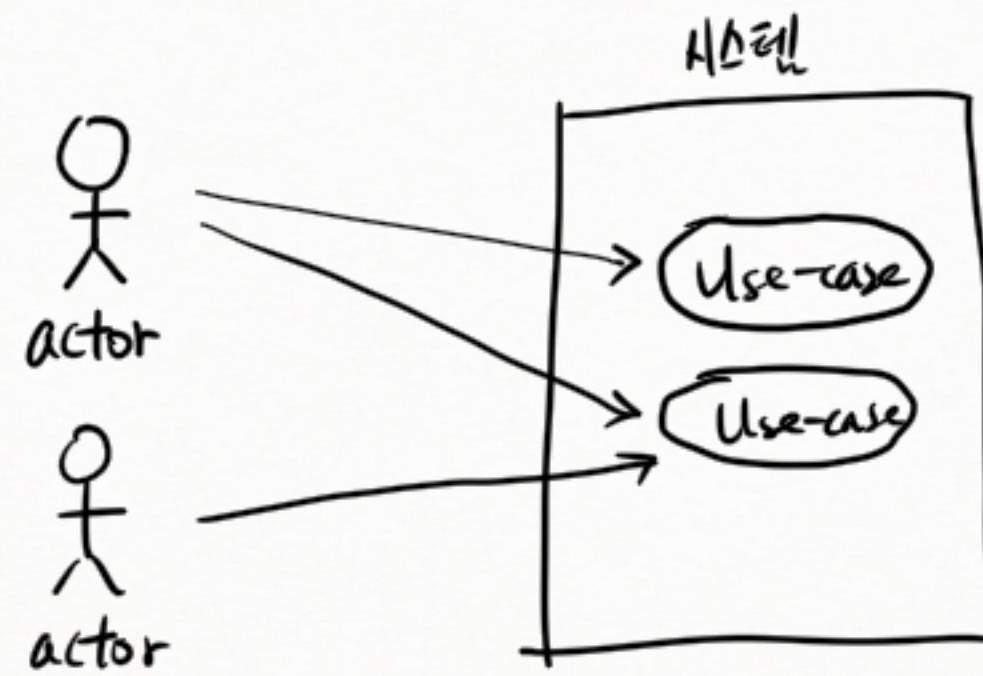
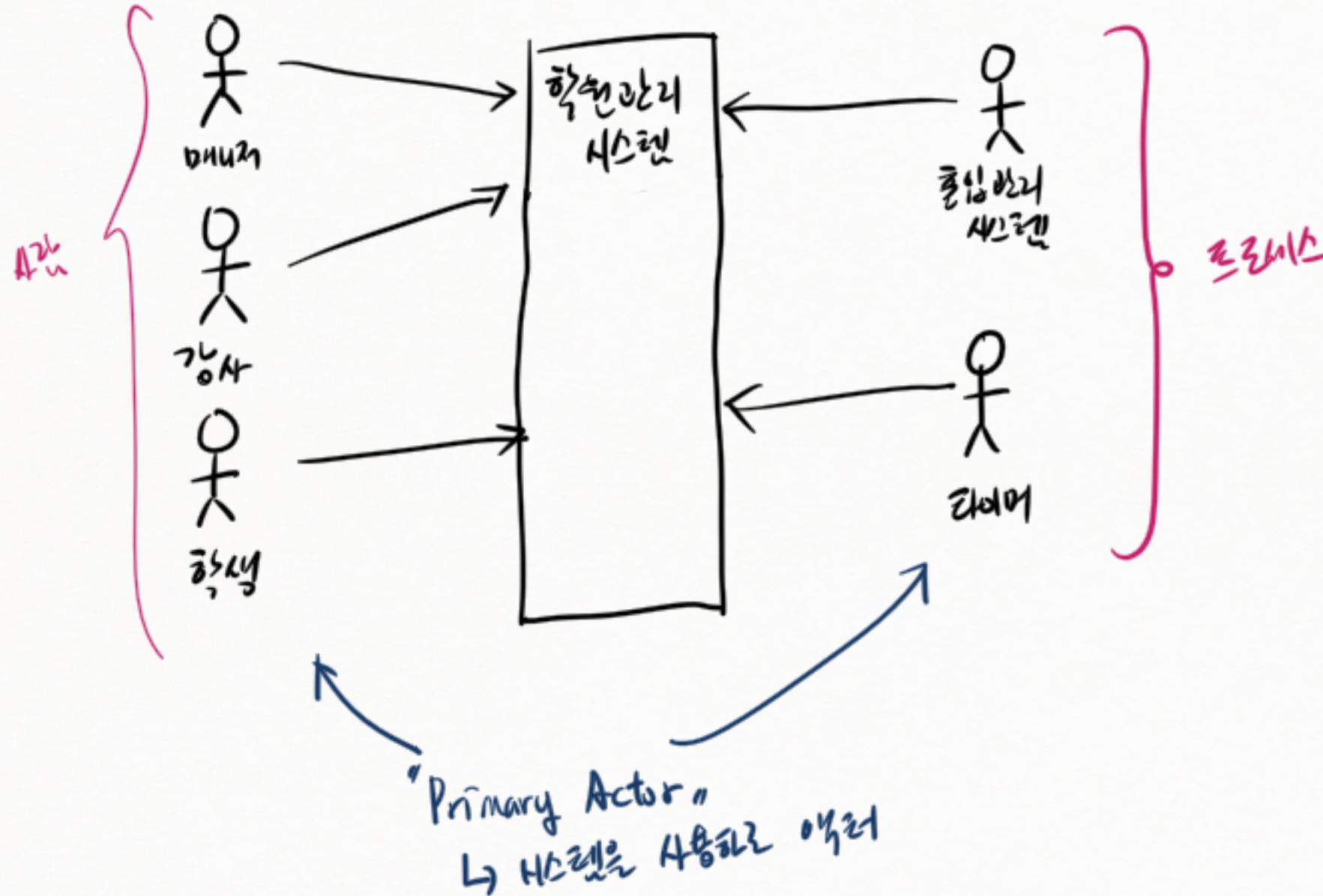


\* Use-case 모델링 - 퍼시시브 디자인  
↳ 개별화된 기능을 사용

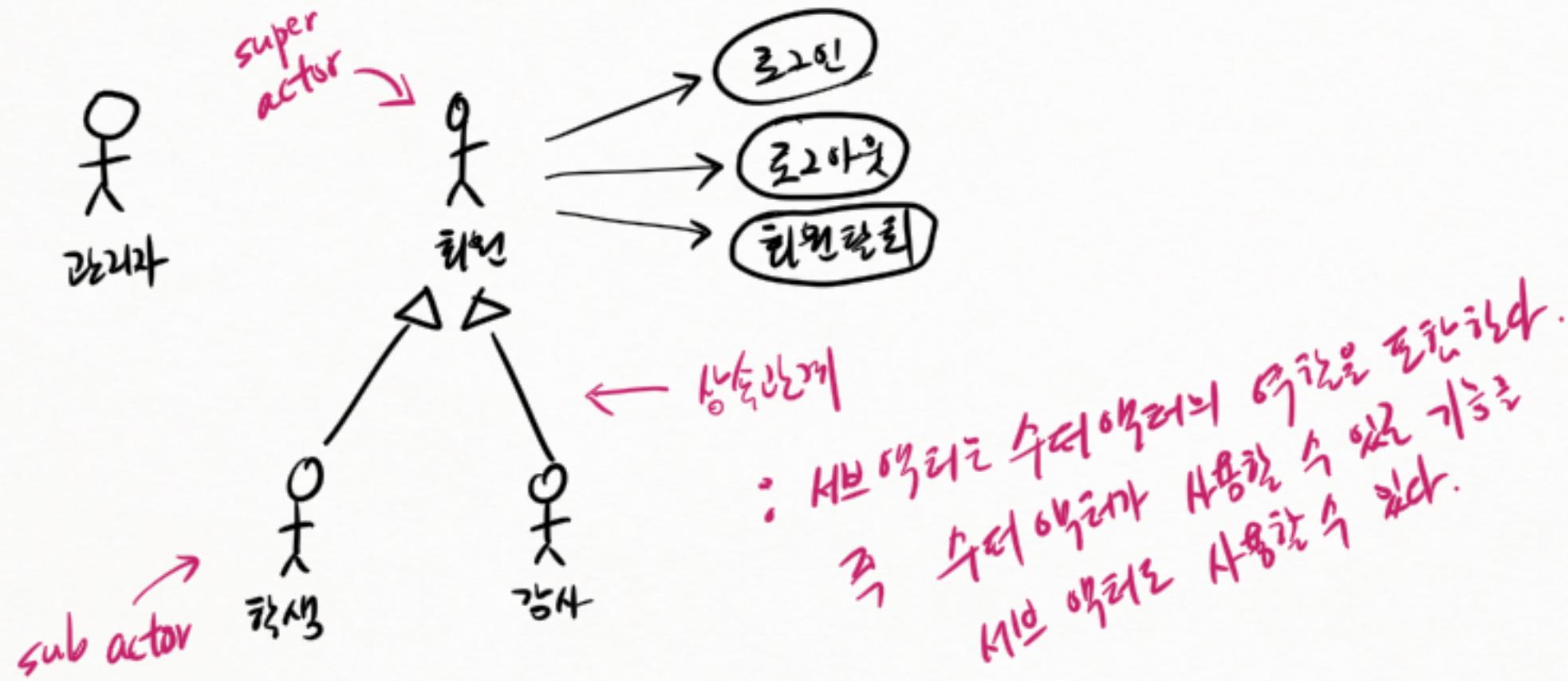


## \* Actor Hits

↳ 시스템을 사용하는 방법, (프로세스)를 가리킨다

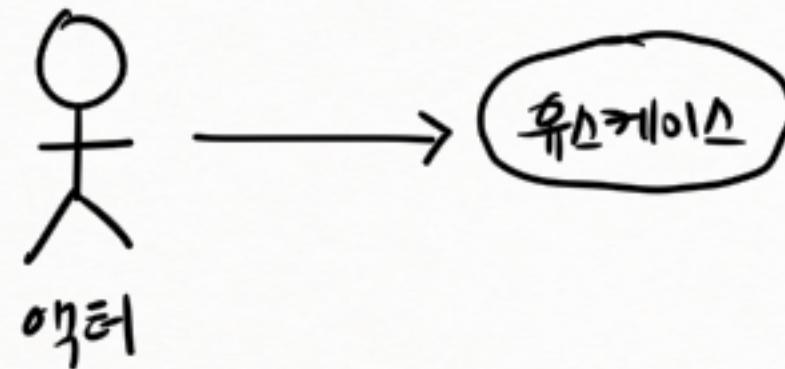


## \* Actor by XML



## \* Use-case 개념

↳ 액터가 시스템을 사용하여 처리하고자 하는 업무 (사용사례)



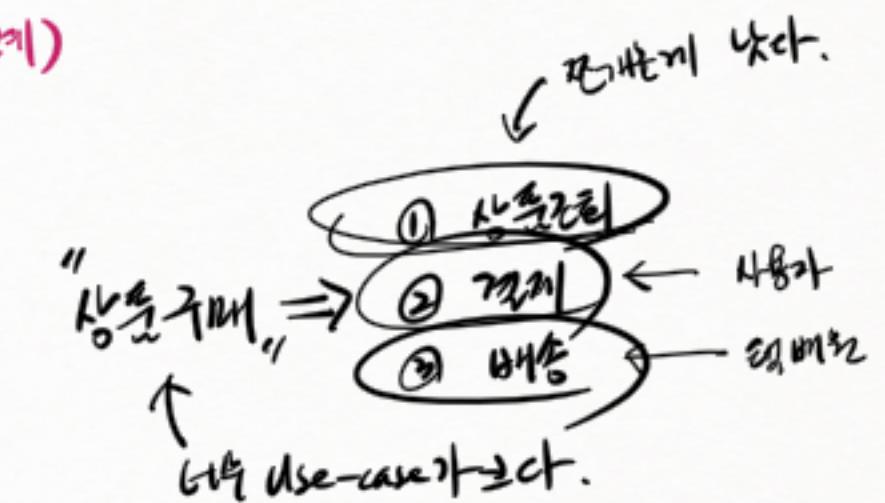
### \* Use-case 특징 지침서 (기준)

- ① 시스템을 통해 수행하는 업무
- ② 한 액터가 수행하는 업무
- ③ 한번에 한 단계에 수행하는 업무

같은 있는 업무

1회~4회 사이에 개발할 수 있는 규모로  
시스템의 기능을 조작할 수 있다.

프로젝트 관리가 용이한 규모로  
개발한 시스템의 기능을 정의할 수 있다.

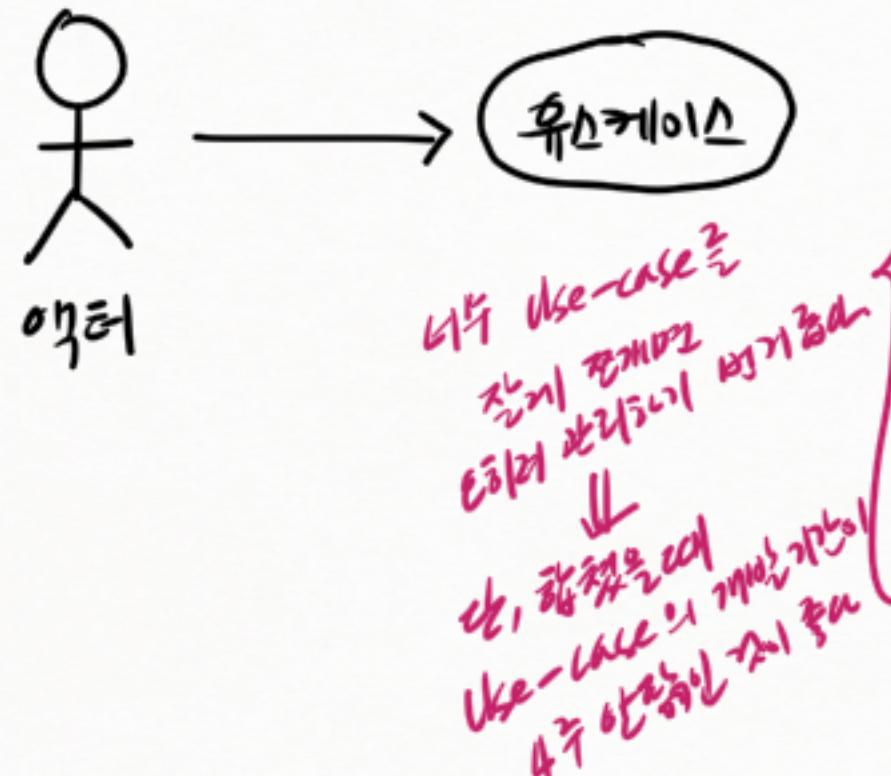


- ① 개시를 등록하기
- 개시를 목록처리하기
- 개시를 상세처리하기
- 개시를 변경하기
- 개시를 삭제하기

~~② 상품등록하기~~  
~~③ 상품판매하기~~ 업무가 아니다!  
그러면 Use-case가  
될 수 없다.

## \* Use-case Hier II

↳ 액터가 시스템을 사용하여 처리하고자 하는 업무 (사용사례)



## \* Use-case $U_1$

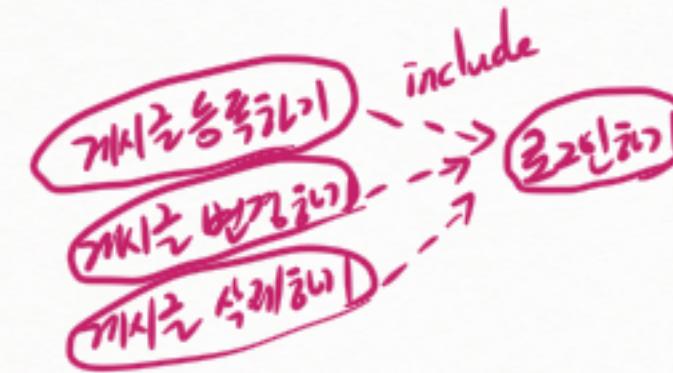
- ① 서로 관련된 Use-case 이중의 한 개의 Use-case를 통합할 수 있다



② CRUD Use-case 은 기본 Use-case 를 확장할 때

- 깨시를 등록하기
  - 깨시를 목록 조회하기
  - 깨시를 상세 조회하기
  - 깨시를 떤경하기
  - 깨시를 삭제하기

} ⇒ **깨시는 관리**

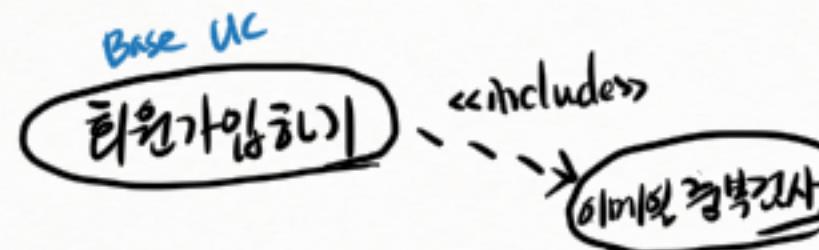


③ 여러 Use-case의 공동 사용 가능한 경의 별도의 Use-case로 뽑을 수 있다.

**로그인하기** ← 로그인하기는 비록 인수는 아니지만  
여러 Use-case에 포함된 책임이다.  
공통 기능으로 보리에게 개별화 시키기

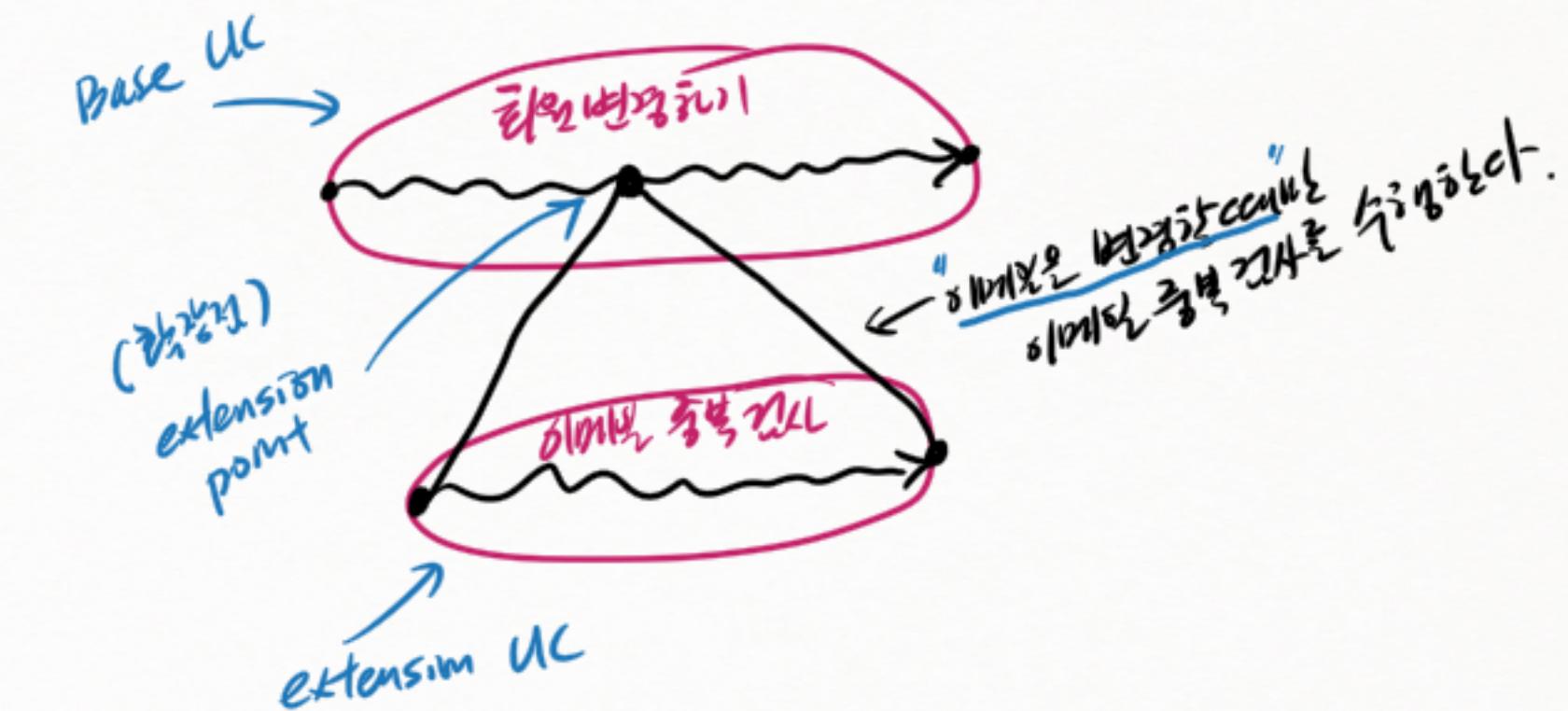
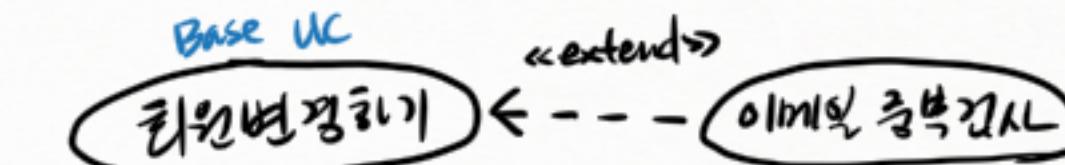
## \* Use-case Diagram

### ① 포함관계 (include) : 필수

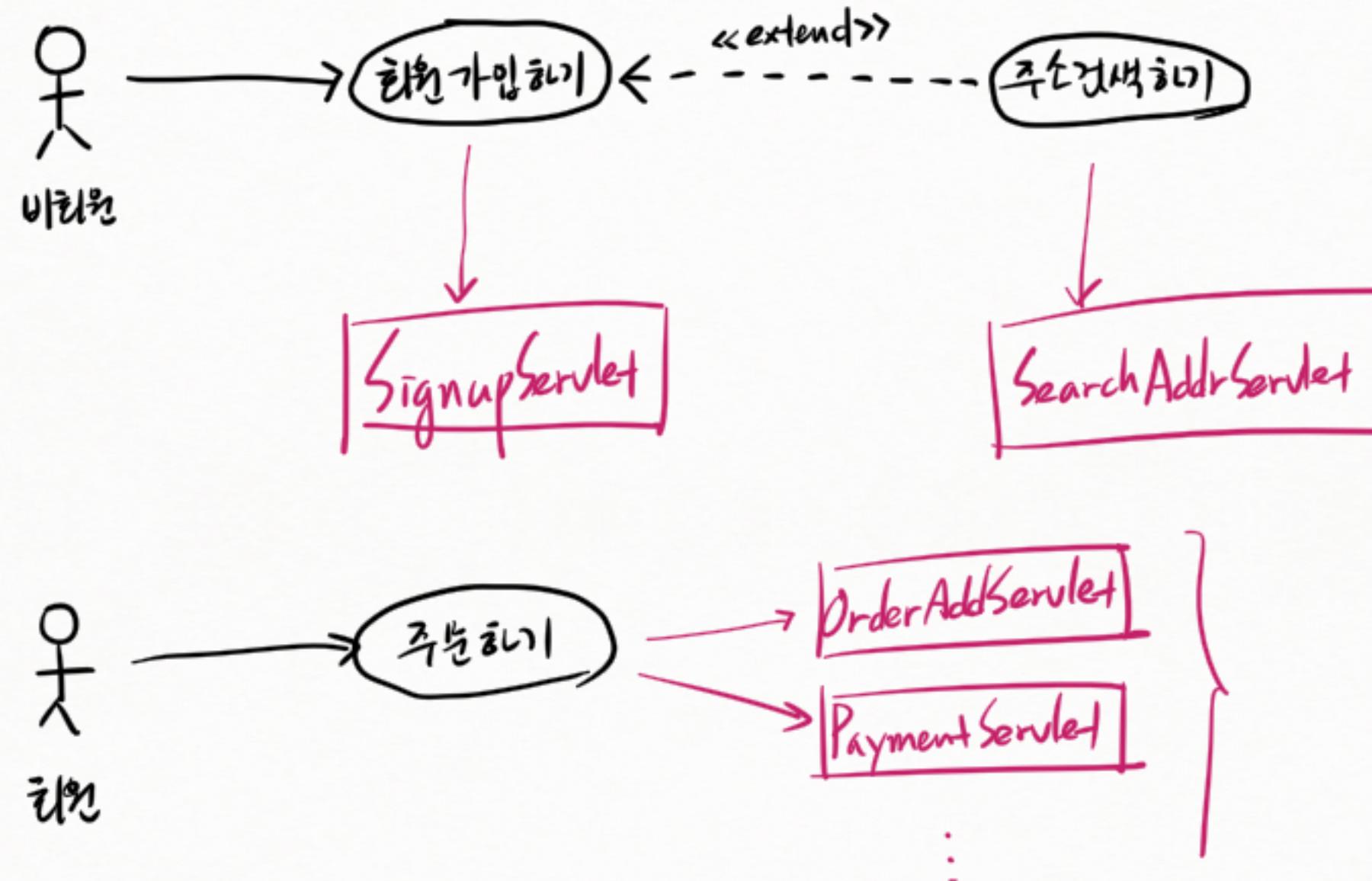


{ 포함 / 확장 관계를 명시적으로 표시하는  
방법이 있음! 하지만 그건 다른 거야!  
그리고 그건 많아도 관계를 표시하는 방법은 그거지 않자!

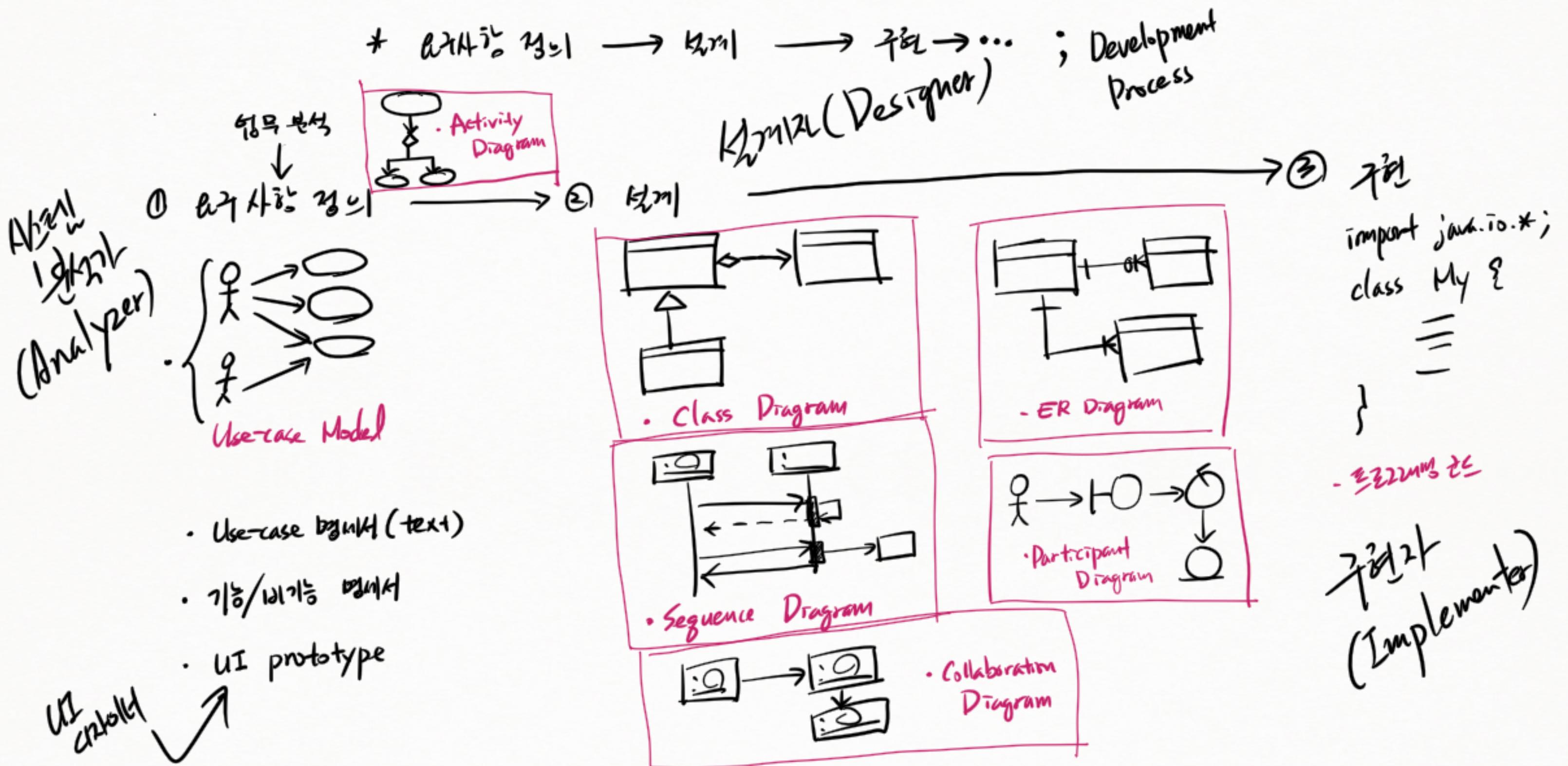
### ② 확장관계 (extend) : 선택



## \* Use-case 와 서블릿



\* 한 개의 Usecase는  
한 번 이상의 서블릿으로 구현된다.  
(컨트롤러)



## \* 프로젝트의 Use-case Modeling 흐름

