

5장 유즈 케이스 다이어그램



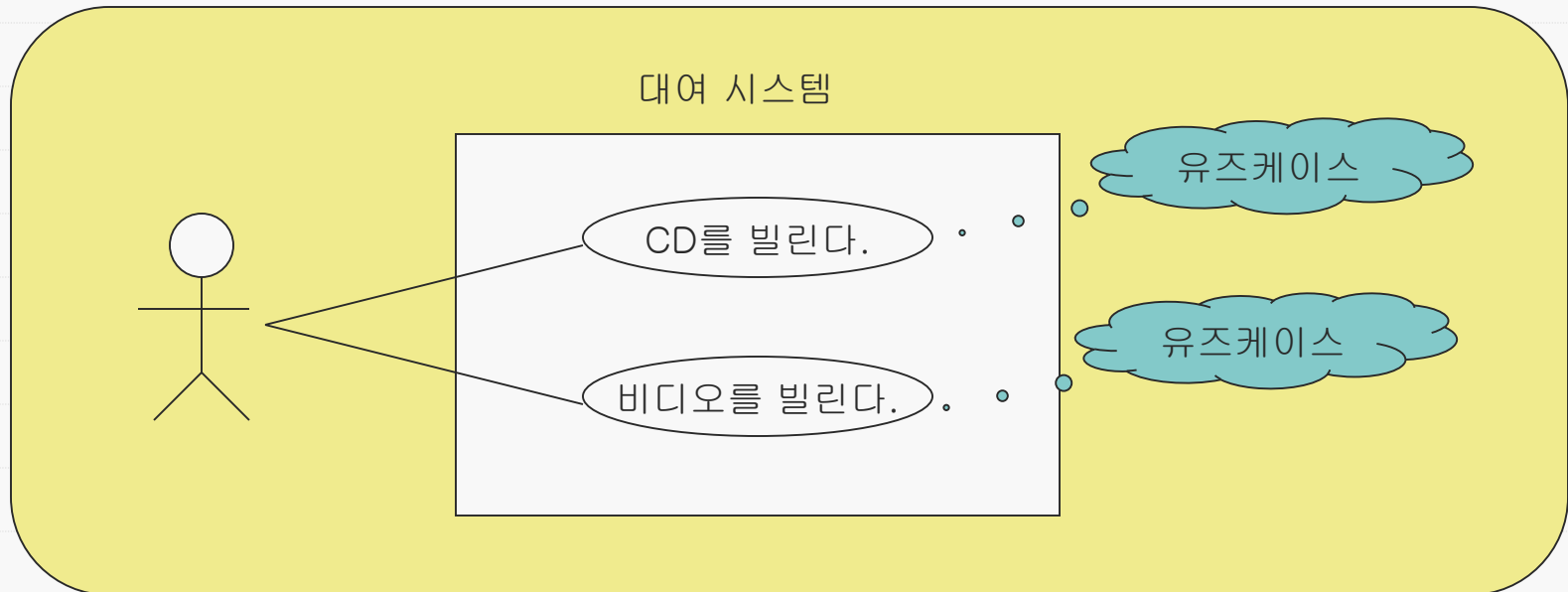
- 5.1 Usecase에 대하여
- 5.2 UseCase 모델링의 목적
- 5.3 Usecase 다이어그램의 요소와 표기법
- 5.4 UseCase 사이의 관계
- 5.5 액터와 유즈케이스의 추출법과 추출규칙
- 5.6 기타 특징들

5.1 Use case에 대하여

(1) 정의

어떤 일에 쓰느냐 하는 것을 의미

- UseCase란 사용의 사례로서 시스템의 사례들을 그려 놓은 것
- Usecase란 시스템의 외부에서 본 기능
- 개발자와 사용자와의 상호작용으로 표시한 것
- Usecase의 목적은 **시스템의 기능을 정의하는 것**(기능의 실현방법은 기술하지 않음)



5.1 use case에 대하여

(2) use case의 특징

- 유스케이스는 사용자 시각에 맞춘 분석.
어떤 시스템을 만드느냐를 사용자 입장에서 조망하는 것
유스케이스는 시스템 보다는 그것을 사용하는 인간, 즉 **사용자의 입장을 우선해서 시스템이 어떠해야 하는가를 알아보는 것**
- 유스케이스는 시스템의 행위를 결정하는 것
구체적으로는 시스템의 기능을 정의하고, 범위를 결정함으로써
시스템과 외부 환경 변수를 구분하고, 상호 관계를 정립하는 것

1. 의사 소통 수단이다.

- 의뢰인 혹은 사용자와 개발자는 시스템의 형태에 대해서 결정
- 이러한 결정을 위해서 UseCase 를 사용함

2. 구현이 독립적이다.

- Use Case 다이어그램을 그릴 때 **시스템의 내부적인 수행과정은 포함하지 않음.**
- 시스템의 기능을 알려주면 의뢰인의 이해를 더 감소시키는 영향을 미침

3. 테스트의 기본이다.

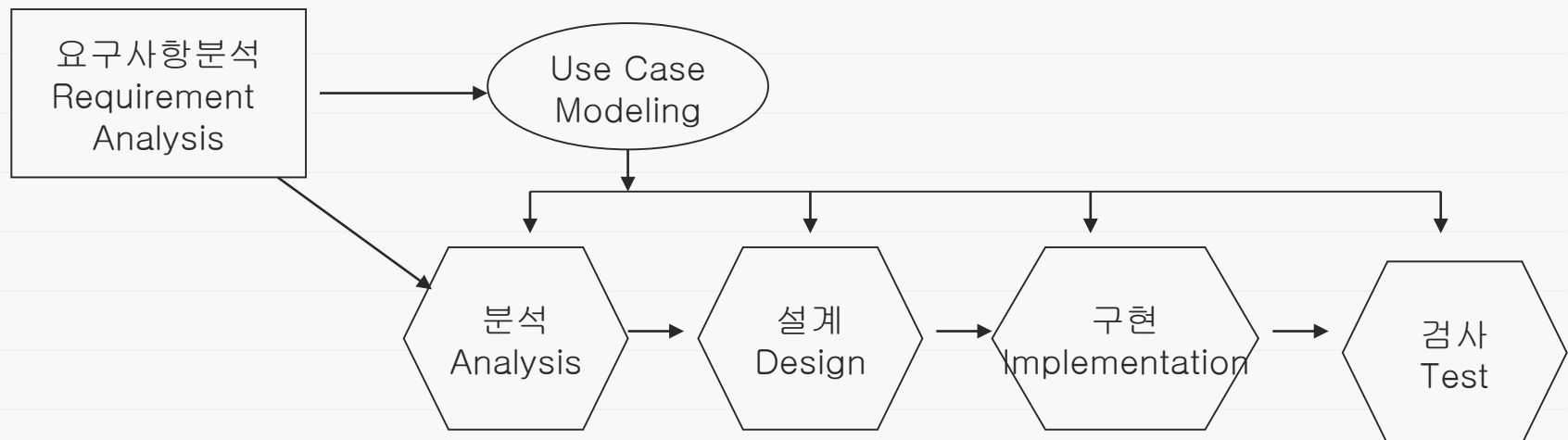
- 의뢰인과 개발자간의 합의로 인해 시스템의 기능이 확정됨
- Use Case 다이어그램은 앞으로 개발되는 중간결과물에 대한 테스트의 기본

유스케이스란 무엇인가

- 📌 유스케이스는 **시스템의 동작 하나를 기술한 것**임
 - 방금 시스템에 특정한 일을 시킨 사용자의 관점에서 작성하며, 사용자가 보낸 자극 '하나'에 대한 반응으로 시스템이 진행하는 '눈에 보이는' 이벤트들의 흐름을 포착한다.
 - 눈에 보이는 이벤트란 - 사용자가 볼 수 있는 이벤트
- 📌 유스케이스는 **사용자의 눈에 보이지 않는 동작을 전혀 기술하지 않고 시스템 안에 숨겨진 메커니즘도 다루지 않는다.**
 - 오직 사용자가 직접 볼 수 있는 것을 적어 놓을 뿐임

5.2 Use Case 모델링의 목적

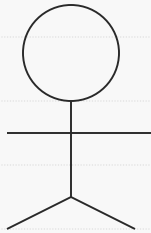
- 개발하려고 하는 **시스템의 기능적인 요구사항**(Functional Requirements)을 **최종 사용자와 개발자와의 합의하에 결정**하고 표현
- 시스템의 기능을 요구분석단계(Requirement Analysis Process)에서 명확하고 일관성있게 표현함으로써, **개발자들간의 의사소통의 수단**으로 이용
- 최종적으로 개발된 시스템의 기능을 초기의 요구사항과 비교, 검증하는데 이용
- 개발된 시스템을 구성하는 실제 클래스와 오퍼레이션들을 통해 기능적인 시스템의 확장과 유지보수의 용이성을 도모



(그림) Use Case 모델의 역할

5.3 use case 다이어그램의 요소와 표기법

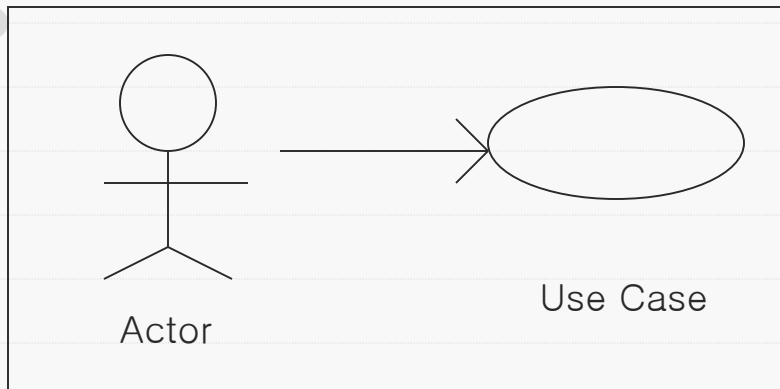
(1) 유즈 케이스 다이어그램에서 사용하는 주요 모델요소



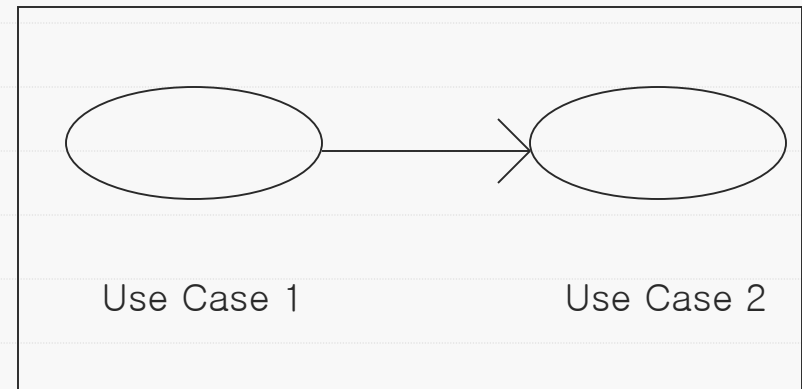
<UML에서 Actor 표기법>



<UML에서 Use Case 표기법>



연관 (Association) 관계



(그림) Use Case Diagram

5.3 use case 다이어그램의 요소와 표기법

(1) 유즈 케이스 다이어그램에서 사용하는 주요 모델요소

↳ 액터

- 액터는 시스템의 일부가 아니다.
- 시스템 사용자가 수행하는 역할을 나타낸다.
- 유즈 케이스를 기동하는 것은 액터가 된다.
- 시스템과 활발하게 정보교환을 하거나, 수동적으로 정보를 수취한다.
- 인간(시스템의 이용자), 하드웨어, 외부 시스템(개발대상 이외의 시스템)이 액터가 될 수 있다.
- 액터는 시스템과 상호작용을 하는 모든 것들을 나타냄
- 시스템을 사용하게 될 사람은 물론이고, 연관된 다른 시스템도 액터임

• 액터
 시스템을 동작하게 하는 사용자(동기부여자)
 시스템과 관계가 있는 타시스템

액터/유스케이스

유즈 케이스

- 유즈 케이스는 액터와 시스템 사이의 대화를 모델화한다.
- 유즈 케이스는 액터에 의해 개시되는 시스템의 기능(시스템 밖에서 본 동작)을 나타낸다.
- 시스템 사용자가 시스템을 이용하여 수행하는 단위업무의 하나를 추상화한 것
- 유스케이스 모델은 시스템과 액터와의 의사소통을 표현함
- 각각의 유스케이스는 시스템이 제공해야 하는 기능을 묘사하고, 이러한 유스케이스들이 시스템 전체의 기능을 나타냄.
- 하나의 유스케이스는 액터가 원하는 기능을 수행하기 위해 시스템이 수행하는 일련의 처리들의 연속임

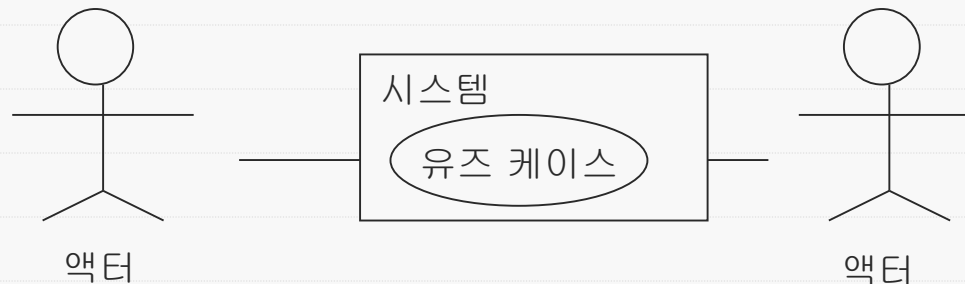
액터와 유스케이스의 관계

- 액터와 유스케이스와의 관계는 연관 (Association) 혹은 커뮤니케이션 연관 (Communicates Association)이라고 함.
- 이는 액터와 유스케이스간의 의사소통을 나타내기 때문임

5.3 use case 다이어그램

(2) Use Case Diagram의 표기와 의의

- 액터와 유즈케이스의 하단에는 각각 액터명과 유즈케이스명을 기록한다.
- 액터와 유즈케이스의 관계는 선의 연결로 표기한다.
- 유즈 케이스는 시스템의 내부, 액터는 외부로 포착할 수 있다.
- 유즈 케이스 다이어그램에 시스템 경계를 명시하는 경우는 유즈 케이스를 큰 장방형으로 둘러싸고, 장방형의 안쪽에 시스템명을 기술한다.
- 액터, 유즈 케이스 및 그러한 관련을 표현한 것을 유즈 케이스 모델이라 한다.
- Actor는 시스템과 상호작용하는 어떤 사람이나 어떤 것이다.
- 시스템과의 상호작용이란 시스템과 어떠한 정보의 교환을 한다는 의미이다.
- Actor는 시스템의 개인사용자가 아니라 하나의 역할을 나타낸다.
- Use Case는 시스템의 핵심적인 기능을 표현한 하나의 단위이다.

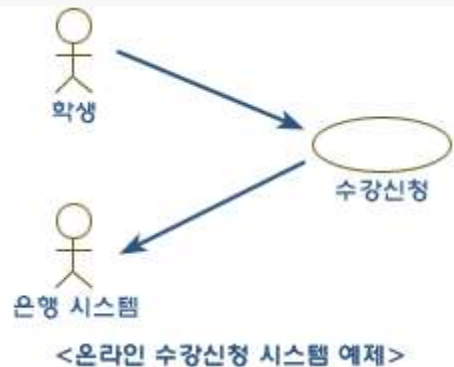


5.3 use case 다이어그램

(3) 도큐먼트에 대하여

- 📌 개요: 유즈 케이스의 목적을 몇 행으로 기술한다.
- 📌 event flow: 유즈 케이스가 언제, 어떻게 개시, 종료하는지, 액터와 유즈 케이스와의 상호작용에 대해 문장으로 기술한다.
- 📌 **시나리오**: 유즈 케이스의 실례를 문서로 기술한다. 각 유즈 케이스는 여러 시나리오의 집합으로서 표현된다.
 - **이벤트의 흐름을 실례(Instance)화하여 텍스트로 나타낸 것을 말한다.**

예



온라인으로 수강신청을 하는 시스템의 일부를 나타낸 것
 액터는 학생과 은행시스템이고,
 유스케이스는 수강신청
 화살표는 액터와 유스케이스간의 관계를 나타낸 것인데,
 단방향으로의 관계만 나타내고 있다.

학생이 웹에 접속해서 수강신청을 하면, 그 정보가 은행시스템에 입력이
 되는 과정을 나타내고 있다.

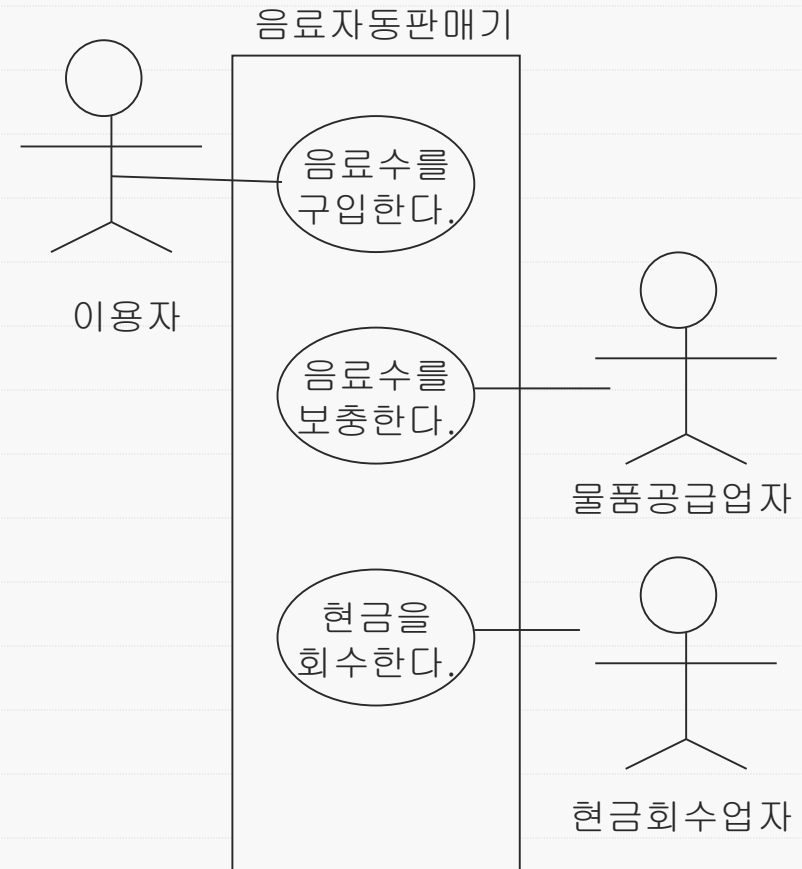
유스케이스 다이어그램으로 표현된 요구사항을 자연어로 표기해보자

| | | |
|-------------------------|---------------|---|
| 유스케이스 다이어그램 으로 표현 | | |
| 자연어로 표현 | 학생은 수강신청을 한다. | 수강신청을 통해 들어온 학생의 정보는 은행시스템에 입력이 된다. |

5.3 use case 다이어그램

(4) Use Case Diagram의 예

- ✚ 액터(Actor) : 이용자, 음료물품 공급자, 현금회수자
- ✚ Use Case : 음료수를 구입한다, 음료물품을 보충한다, 현금을 회수한다



<그림> 자동판매기의 유즈 케이스 모델

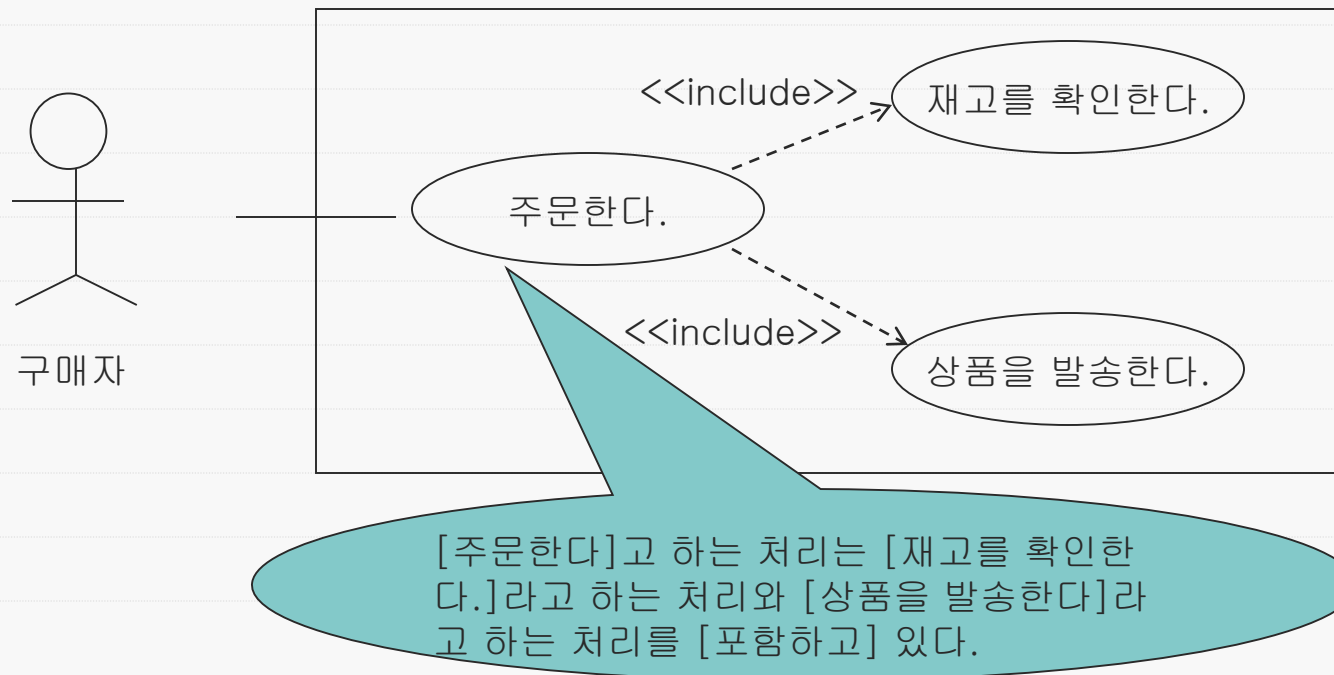
5.4 Use Case 사이의 관계

(1) Use Case간의 관계

📌 포함관계(Include)

기본 유스케이스에서 포함 유스케이스로 향하는 관계이다.

포함 유스케이스에 정의된 행위가 명시적으로 기본 유스케이스에 정의된 행위로 삽입되는 방법을 설명한다.



5.4 Use Case 사이의 관계

(1) Use Case간의 관계

확장관계(Extend)

- 유즈케이스의 변형(variation)을 늘릴 때에 확장 개념을 이용한다.
- 확장은 기저 유즈케이스로 확장점을 선언해서 이용한다.
- 기술은 의존관계와 관련하여 스테레오타입 <<extend>>를 부여해서 나타낸다.



[상품의 일부를 발송한다]고 하는 처리는
[상품을 발송한다]고 하는 처리가 근원이 되고 있다.

유스케이스간의 관계 - 포함/확장

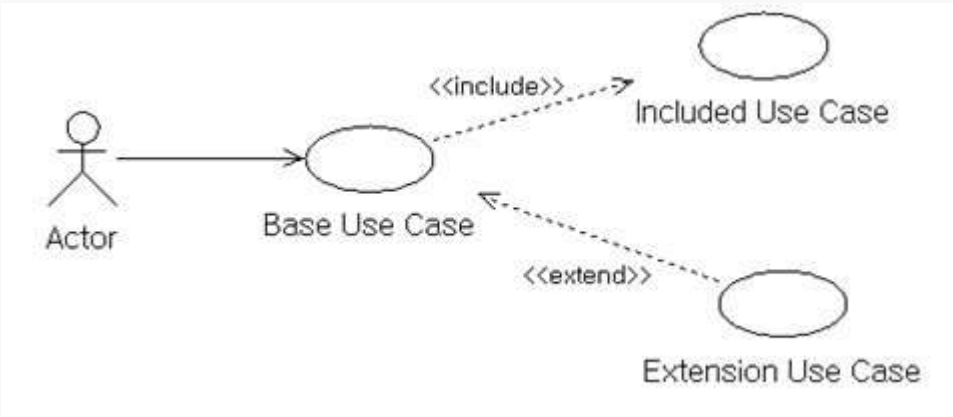
- [1] 포함(Inclusion)-여러 유스케이스들이 하나의 기능 조각을 공유할 때 이를 모든 유스케이스에 각각 집어 넣는 것보다는 이를 분리해 두고 필요한 유스케이스들이 이를 포함해서 사용하게 됨.
- 예) 인터넷 사이트에 접속해서 각종 서비스를 제공받기 전에 늘 수행하는 회원 인증과 같은 유스케이스

두 개 이상의 유스케이스를 작성하는 데
공통의 시나리오가 유스케이스들에 포함되어 있을 경우
공통부분을 분리해서 독립된 유스케이스로 작성
그리고 원래의 유스케이스와 공통부분에 해당하는 유스케이스 사이에 포함을 작성

- [2] 확장(Extension)-기본 유스케이스에서 **특정 조건이나 액터의 선택에 따라 발생하는 유스케이스**를 말함.
- 예) ATM에서 사용자의 메뉴 선택에 따라 달라지는 유스케이스의 경우나 긴급 상황시에 발생할 수 있는 유스케이스

포함 - 기본 유스케이스에서 다른 유스케이스를 반드시 포함하는 관계
반드시 사용한다는 의미 (기본 유스케이스 본연의 시나리오를 분리한 것)

확장 - 기본 유스케이스에서 다른 유스케이스를 **선택적으로 확장**되는 관계



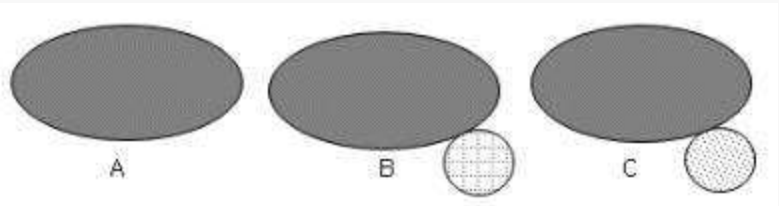
유스케이스와 액터와의 관계에서는 Association만 허용하고, 유스케이스 사이의 관계에서는 Dependency만 허용함
액터가 없는 기본 유스케이스는 존재할 수가 없게 됨

포함 유스케이스가 다른 유스케이스의 시나리오를 반드시 포함하는 것이라면, 그에 비하여 확장은 선택적으로 시나리오를 진행할 경우에 사용
기본 유스케이스에서 **특정 시점이 되면 어떤 내용이 수행될 수도 있고, 안될 수도 있을** 경우 그 수행되는 내용을 **확장 유스케이스**로 떼어 내어 모델링을 한다.

확장 유스케이스가 수행이 되든 안되든 기본 유스케이스는 본연의 시나리오가 공백이 없이 진행할 수 있어서 **본연의 기능을 발휘**할 경우에 <<extend>>를 사용
확장은 포함과 달리 기본 유스케이스 본연의 시나리오를 분리한 것이 아니라, 수행을 해도 되고 안 해도 되는 경우에 별도로 요구사항을 관리하기 위한 것임

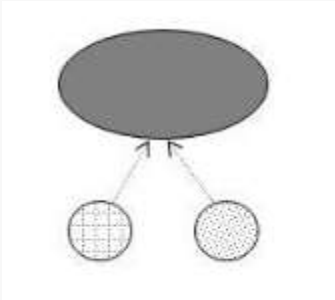
확장이 발생하는 과정

A를 작성한 상태에서 B를 작성
그런데 B가 A에 특정 부분을 더한 것과 같다는 것을 알게 됨
또 C를 작성하면서도 마찬가지로 사실을 발견



B, C를 A를 기준으로 해서 살펴본다
B, C에 더해지는 부분이 **특정한 조건이 되었을 때** A 시나리오의 **특정 부분에 추가**됨을 알게 됨

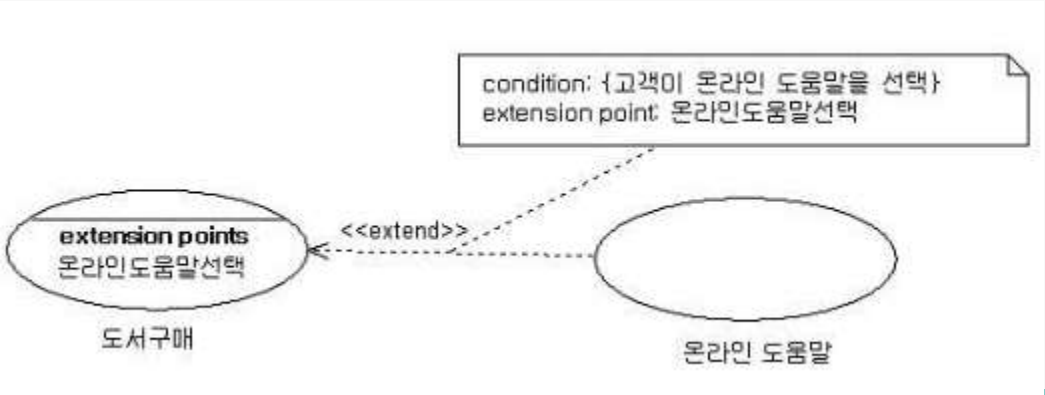
B, C에서 더해진 부분을 떼어 독립된 유스케이스로 작성
A를 확장되는(extended) 유스케이스, B, C를 확장하는(extending) 유스케이스라고 함



확장하는 유스케이스는
특정 조건이 만족될 때 수행되어야 하는 시나리오를
확장되는 유스케이스의 특정 지점에 추가해서 작성됨
이 특정 지점을 확장지점(extension point)이라고 함
다수의 확장하는 유스케이스가 있을 경우 확장지점들(extension points)이 관리되어야 함

확장지점들은 확장되는 유스케이스에 extension points 레이블 밑으로 작성됨

확장 조건과 확장 지점은 노트(note)에 condition과 extension point라는 레이블 다음에 작성
조건은 대괄호 안에 작성

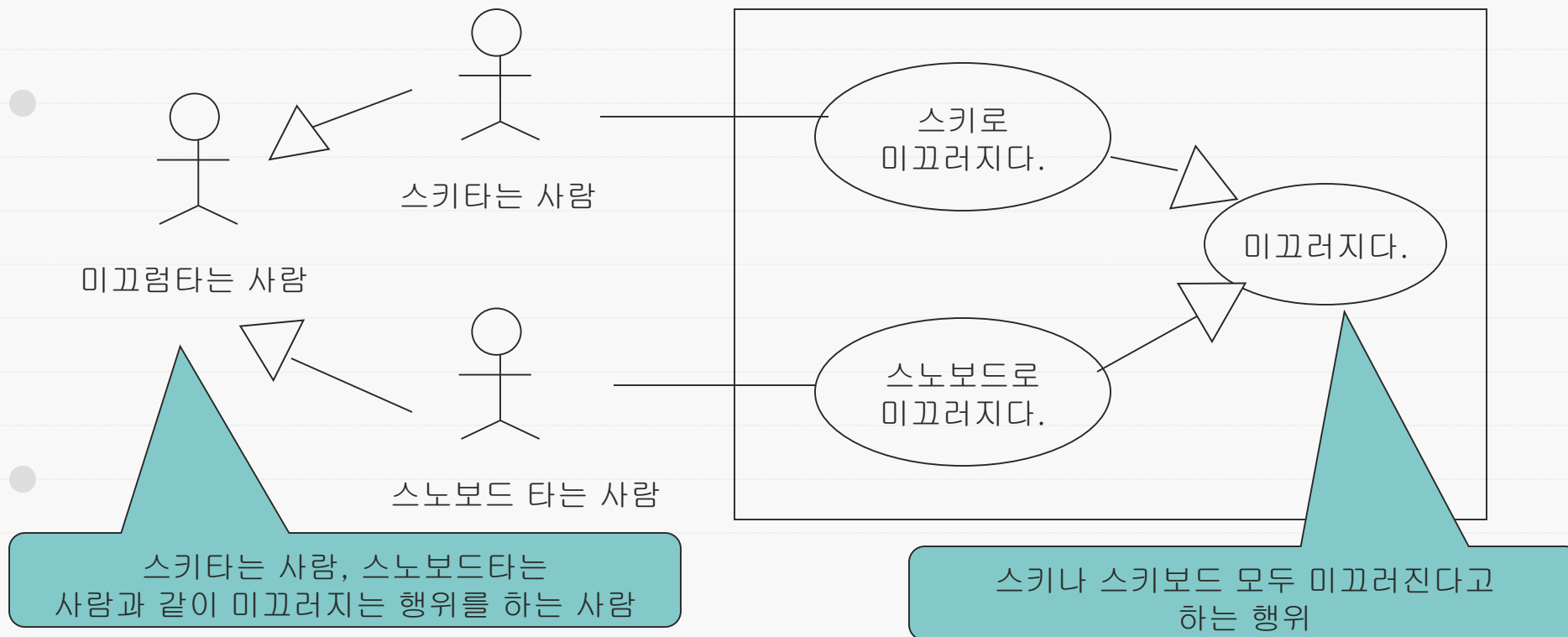


5.4 Use Case 사이의 관계

(1) Use Case간의 관계

📌 일반화관계(Generalization)

- 유즈케이스 다이어그램에서도 액터와 유즈케이스의 범용화를 행할 수가 있다.
- 동작의 변형(variation)을 늘리는 것도 가능해진다.



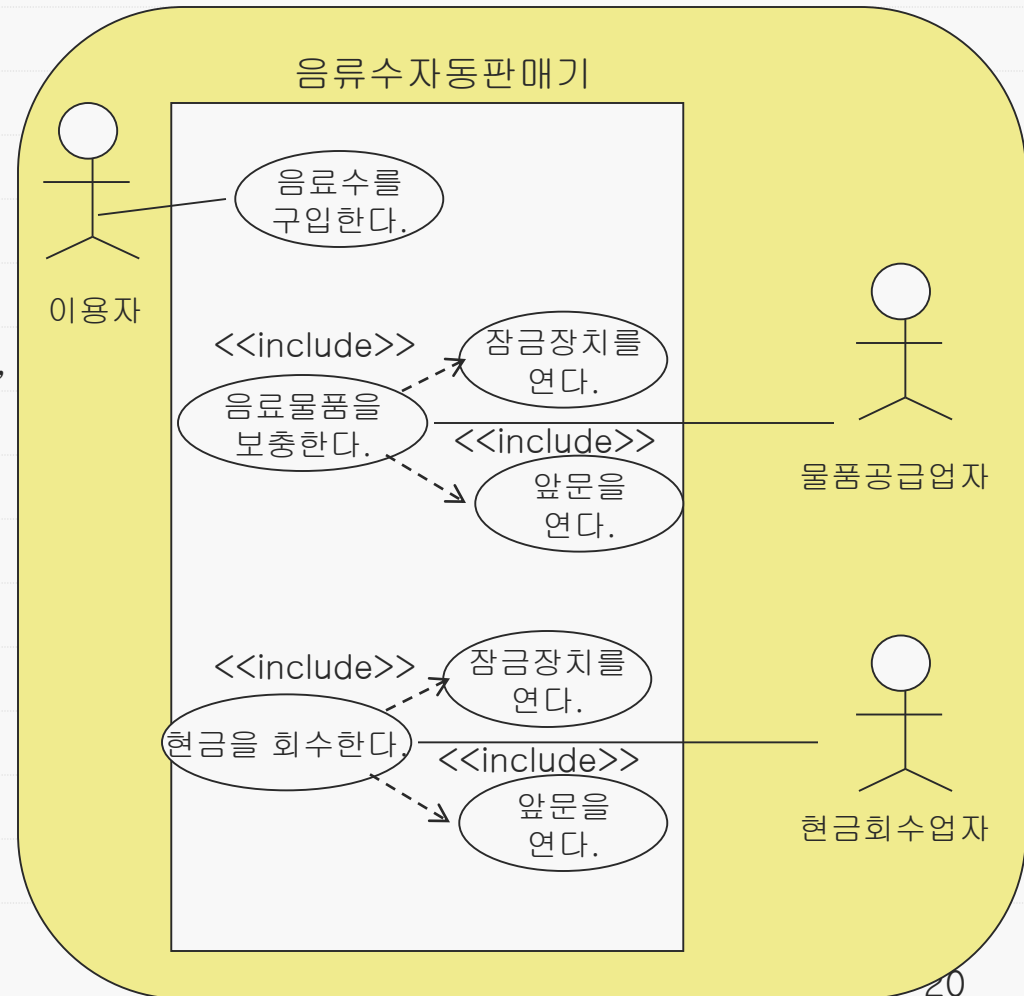
5.4 Use Case 사이의 관계

(2) Use Case의 관계 사용사례

포함(Include)

그림설명

- [음료물품을 보충하다]는 [잠금장치를 연다]를 포함하고 있다.
- [음료물품을 보충하다]는 포함하는 측이고, [잠금장치를 연다]는 포함되는 측이므로 독립하여 사용할 수 없다.



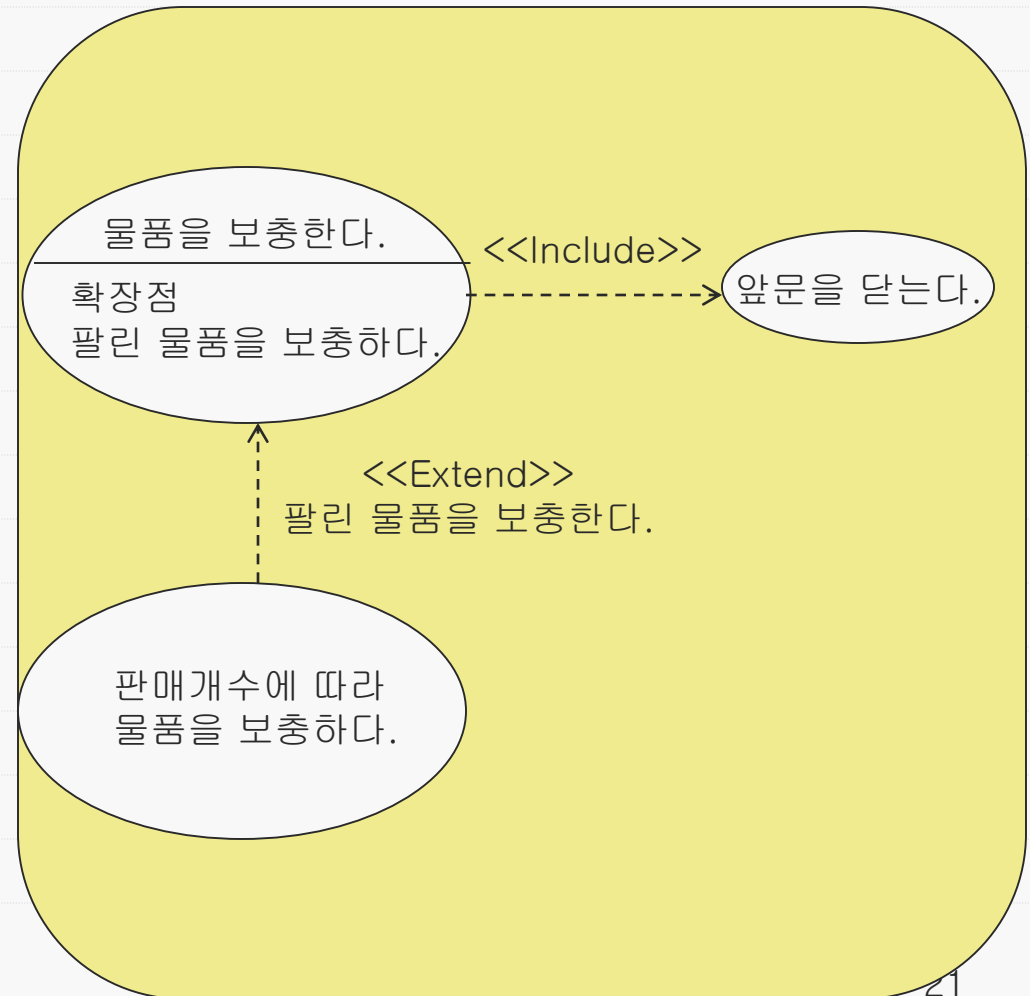
5.4 Use Case 사이의 관계

(2) Use Case의 관계 예

확장(Extend)

그림설명

- [물품을 보충하다.]은 판매개수에 따라 물품을 보충할 수 있으므로 이 관계를 확장이라고 할 수 있다.



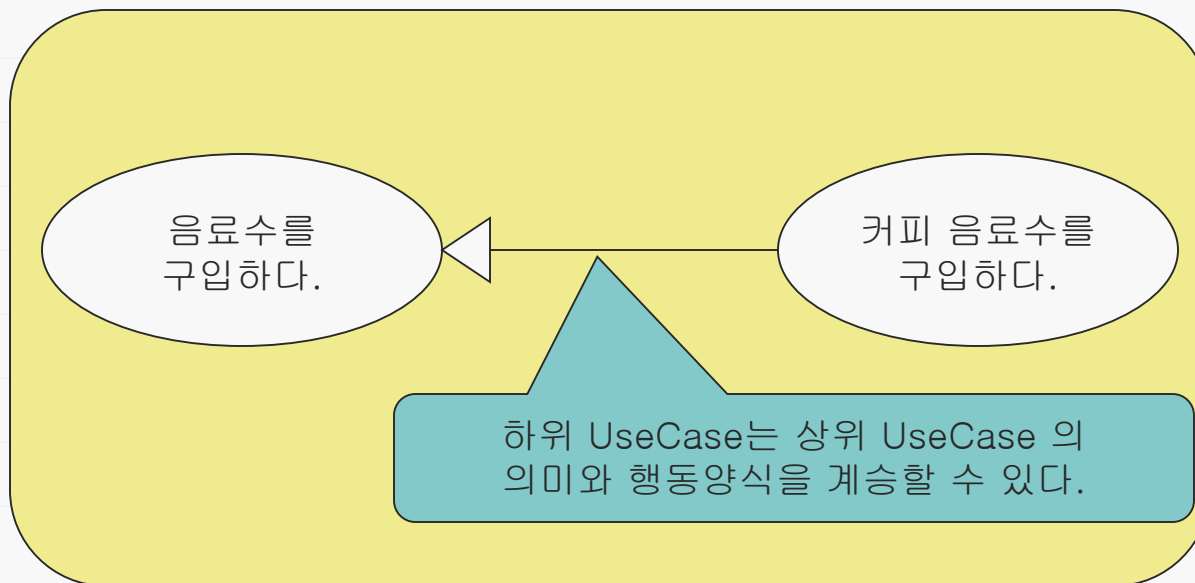
5.4 Use Case 사이의 관계

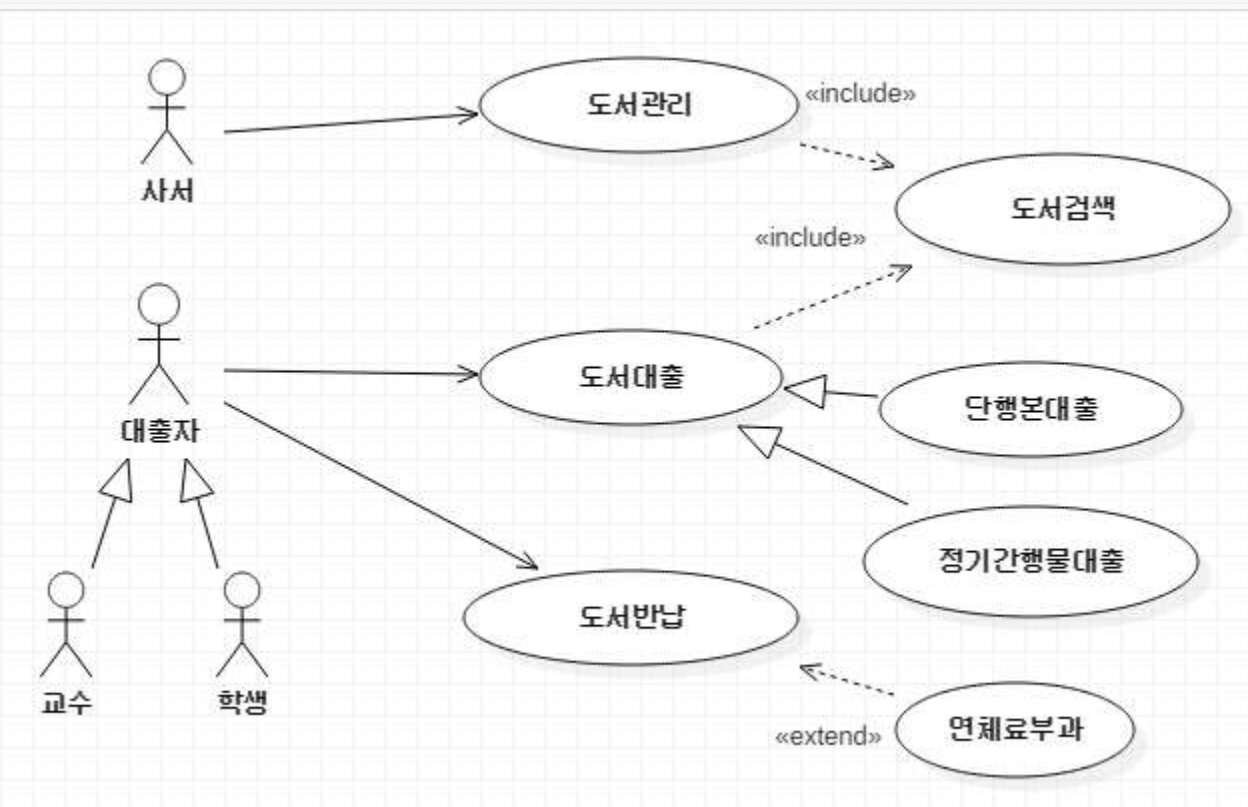
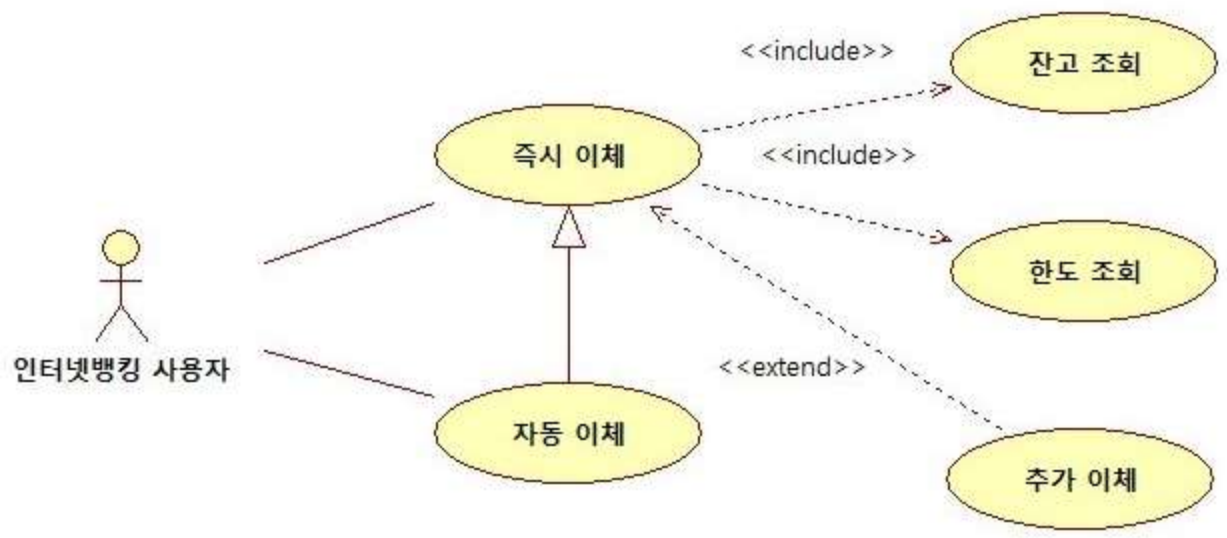
(2) Use Case의 관계 예

📌 일반화(Generalization)

그림설명

- [커피음료수를 구입하다.]는 유즈 케이스는 [음료수를 구입하다.]의 하위 유즈 케이스가 된다.
- 상위 케이스로부터 계승한 또 다른 [설탕을 추가하다.], [밀크를 섞는다]등의 독자 행동양식을 갖고 있다.





5.5 액터와 유즈 케이스의 추출법과 추출 규칙

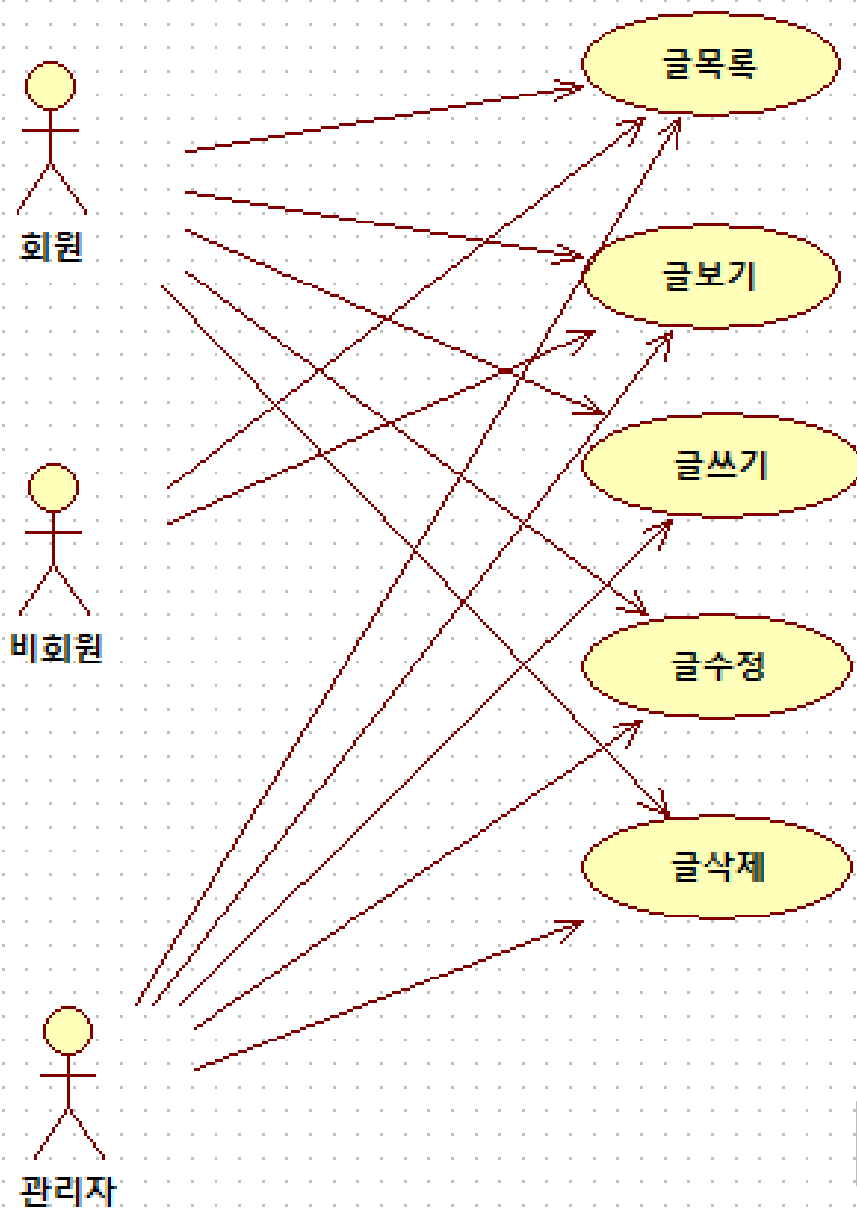
(1) Actor 추출법과 관련성 추출규칙

- ① 시스템의 주기능을 사용하는 사람은 누구인가?
- ② 누가 시스템으로부터 업무지원을 받는가?
- ③ 누가 시스템을 운영, 유지보수하는가?
- ④ 시스템과 정보를 교환하는 외부 시스템은 무엇인가?
- ⑤ 시스템이 내어놓은 결과를 누가 관심을 가지는가?
- ⑥ 한 사람이 여러가지 역할을 하는가?
- ⑦ 시스템이 기존의 시스템과 상호작용하는가?
- ⑧ 여러 사람이 동일한 역할을 하는가?

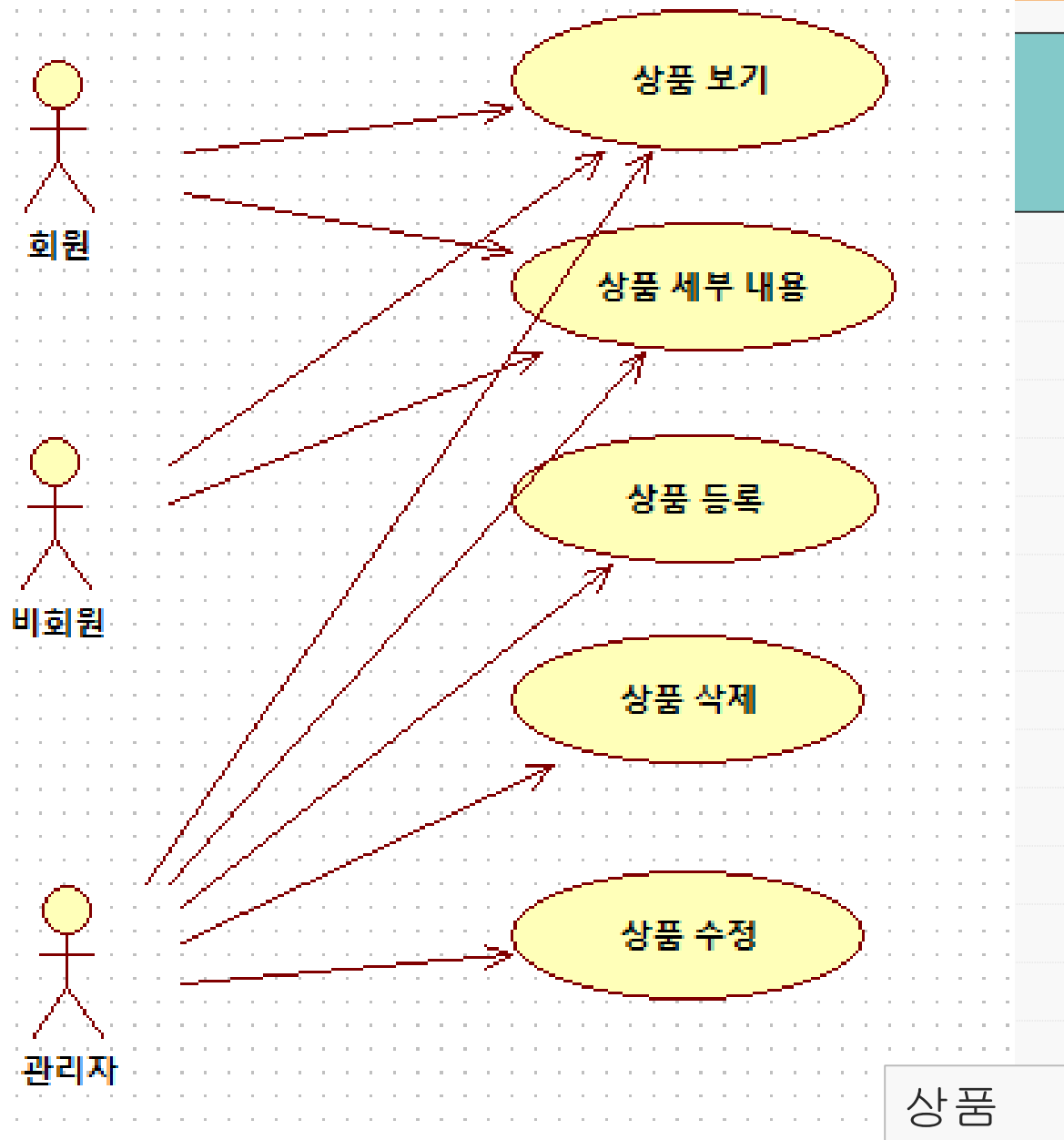
5.5 액터와 유즈케이스의 추출법과 추출규칙

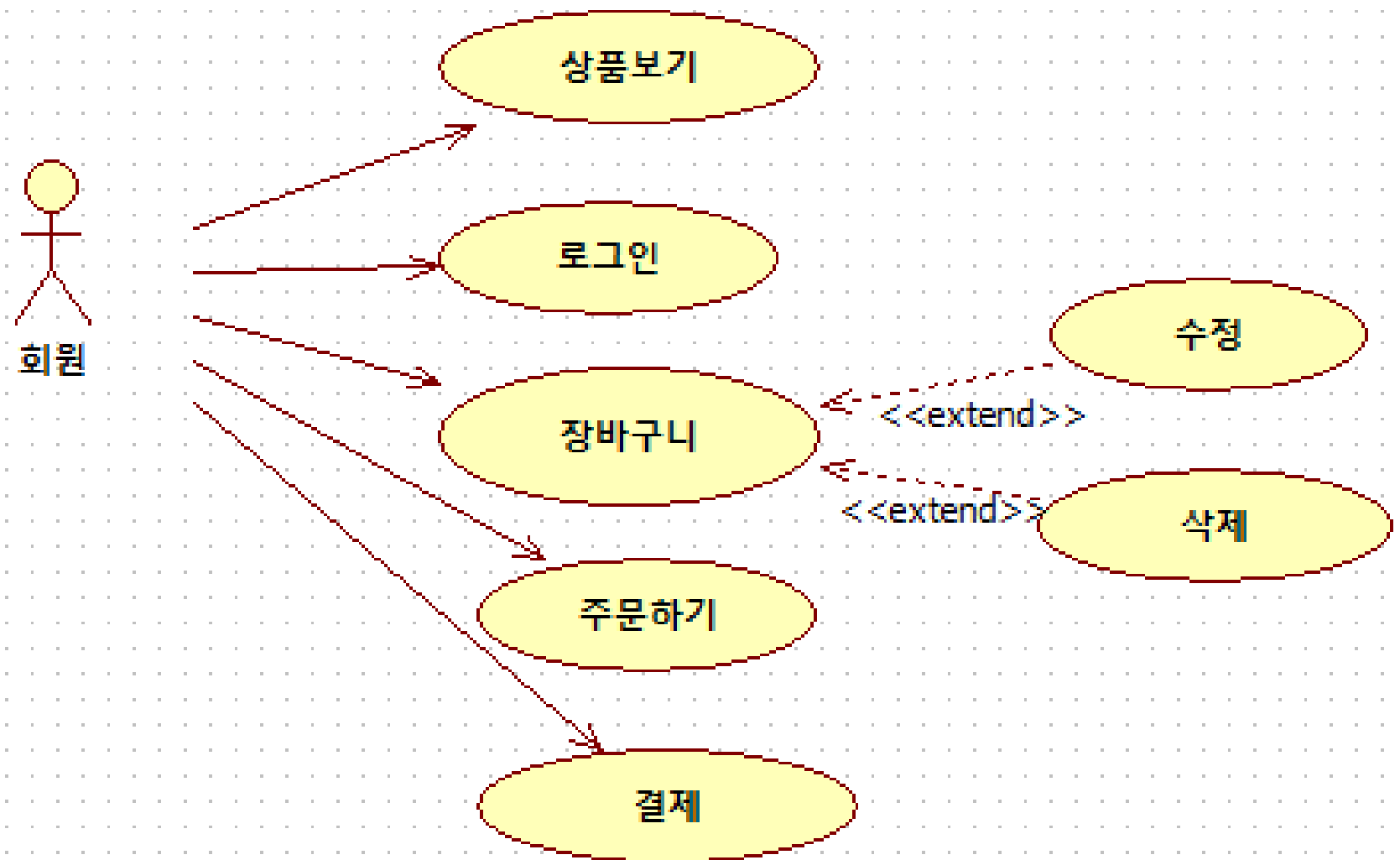
(2) Use Case 추출법과 후부 추출규칙

- ① Actor가 요구하는 시스템의 주요기능은 무엇인가?
- ② Actor가 시스템의 어떤 정보를 수정, 조회, 삭제, 저장하는가?
- ③ 시스템이 Actor에게 주는 어떤 Event가 있는가?
- ④ Actor가 시스템에 주는 어떠한 Event가 있는가?
- ⑤ 시스템의 입력과 출력으로 무엇이 필요한가?
- ⑥ 입력과 출력이 어디에서 오고 어디로 가는가?
- ⑦ 시스템의 구현에서 가장 문제가 되는 점은 무엇인가?

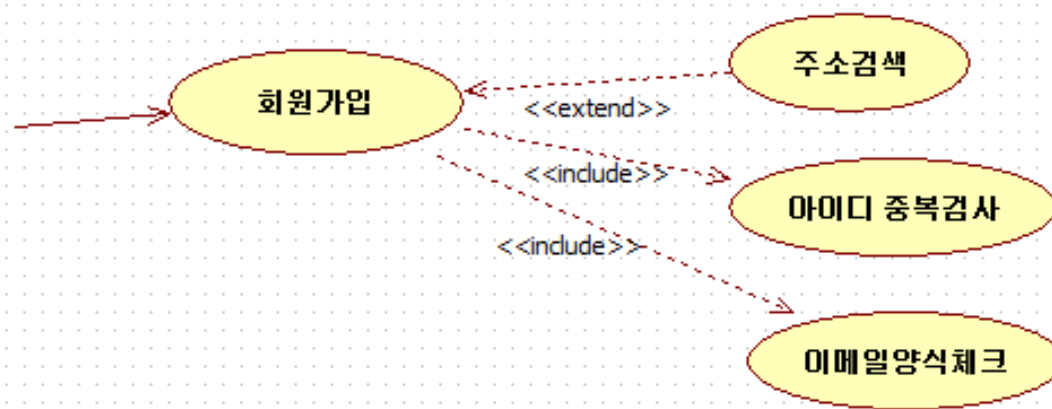


상품 이용후기

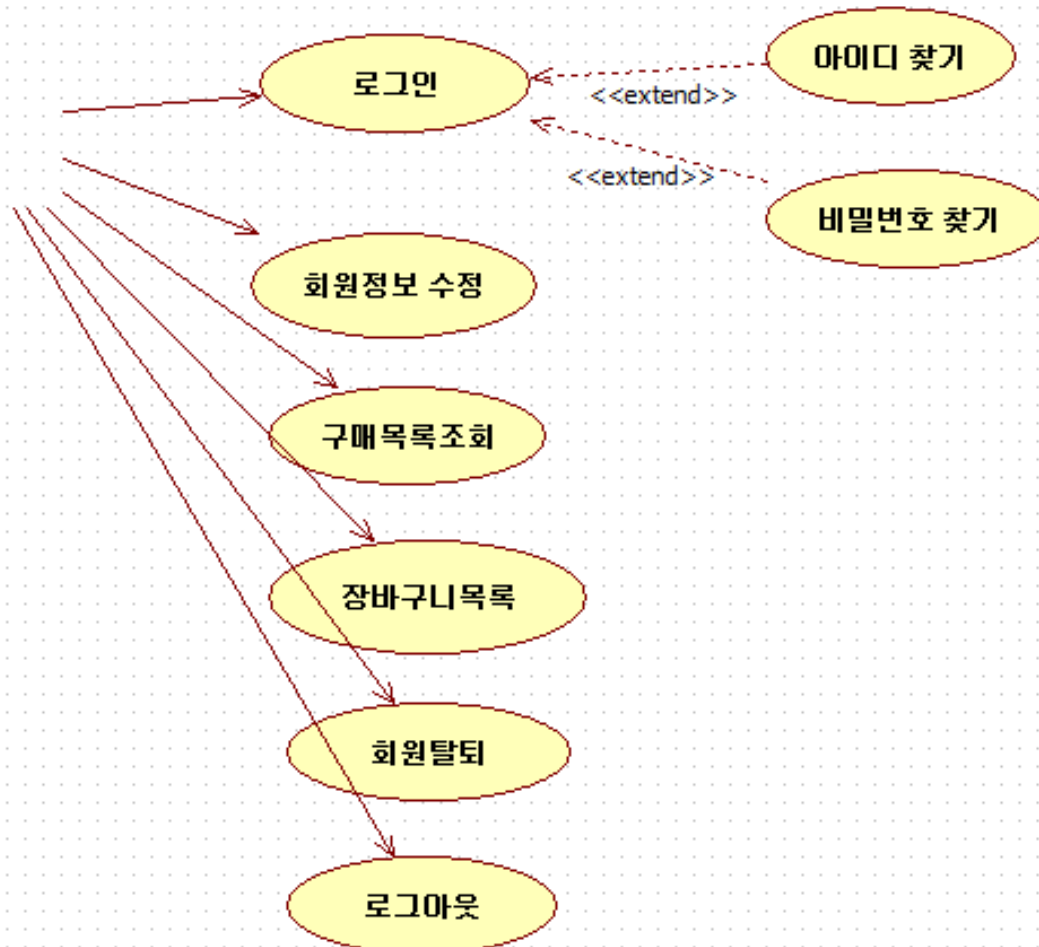




비회원



회원

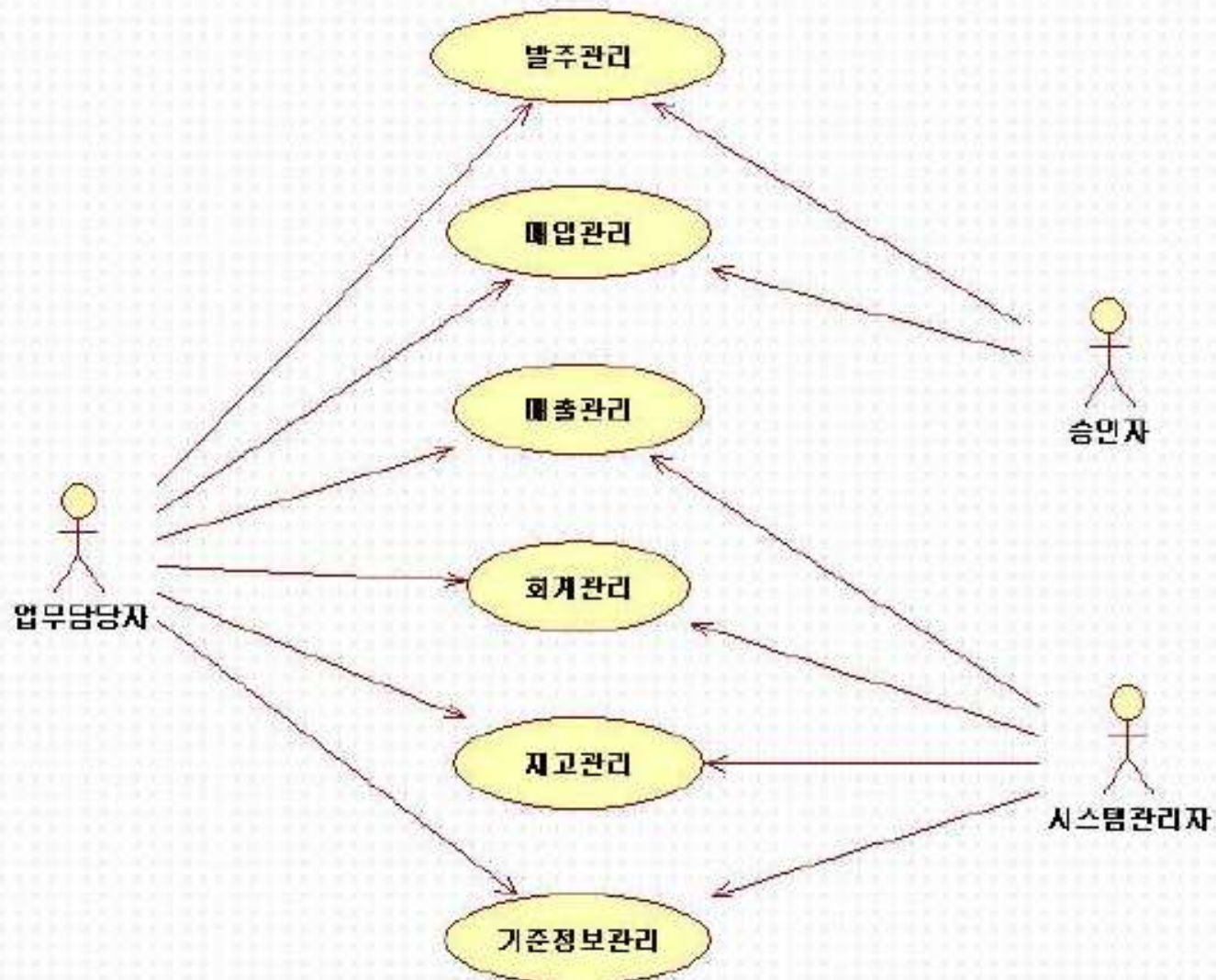


예 - 생산관리 시스템

📌 요구 사항

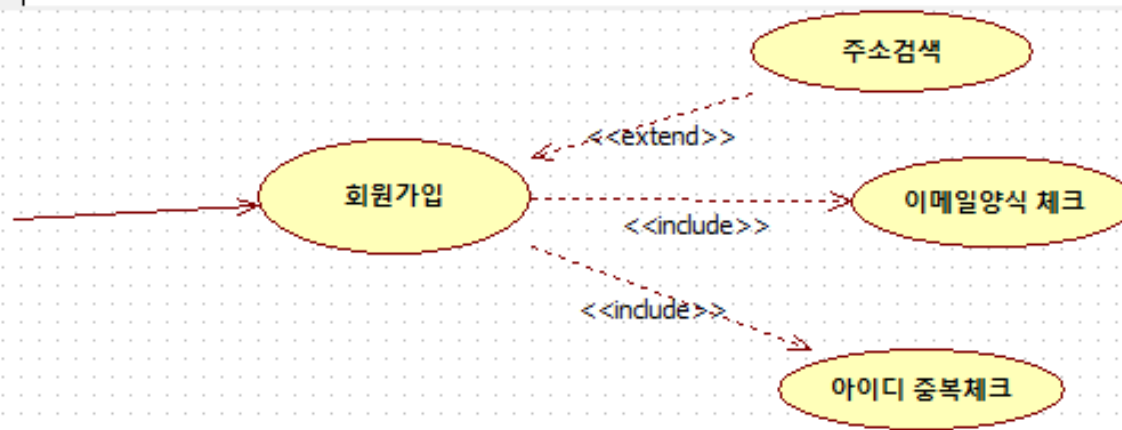
- 업무 담당자는 시스템에 접속하여 발주, 매입, 매출, 회계, 재고, 기준 정보 관리가 가능하다.
- 승인자는 발주와 매입에 대한 승인을 하며, 시스템 관리자는 매출, 회계, 재고, 기준정보에 대한 시스템 관리를 수행한다.
- 발주가 등록되면 발주를 위한 준비를 하여 재고 여부 조사 후 상품을 배송한다.

생산관리 시스템 유스케이스

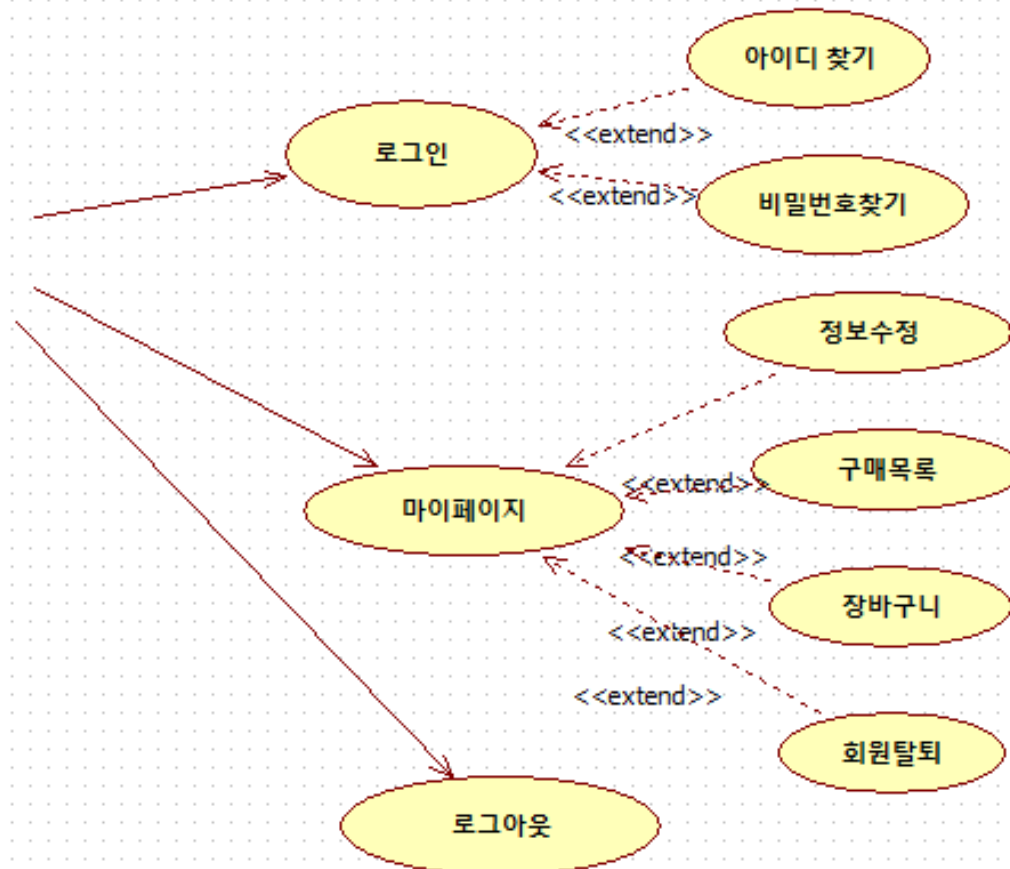




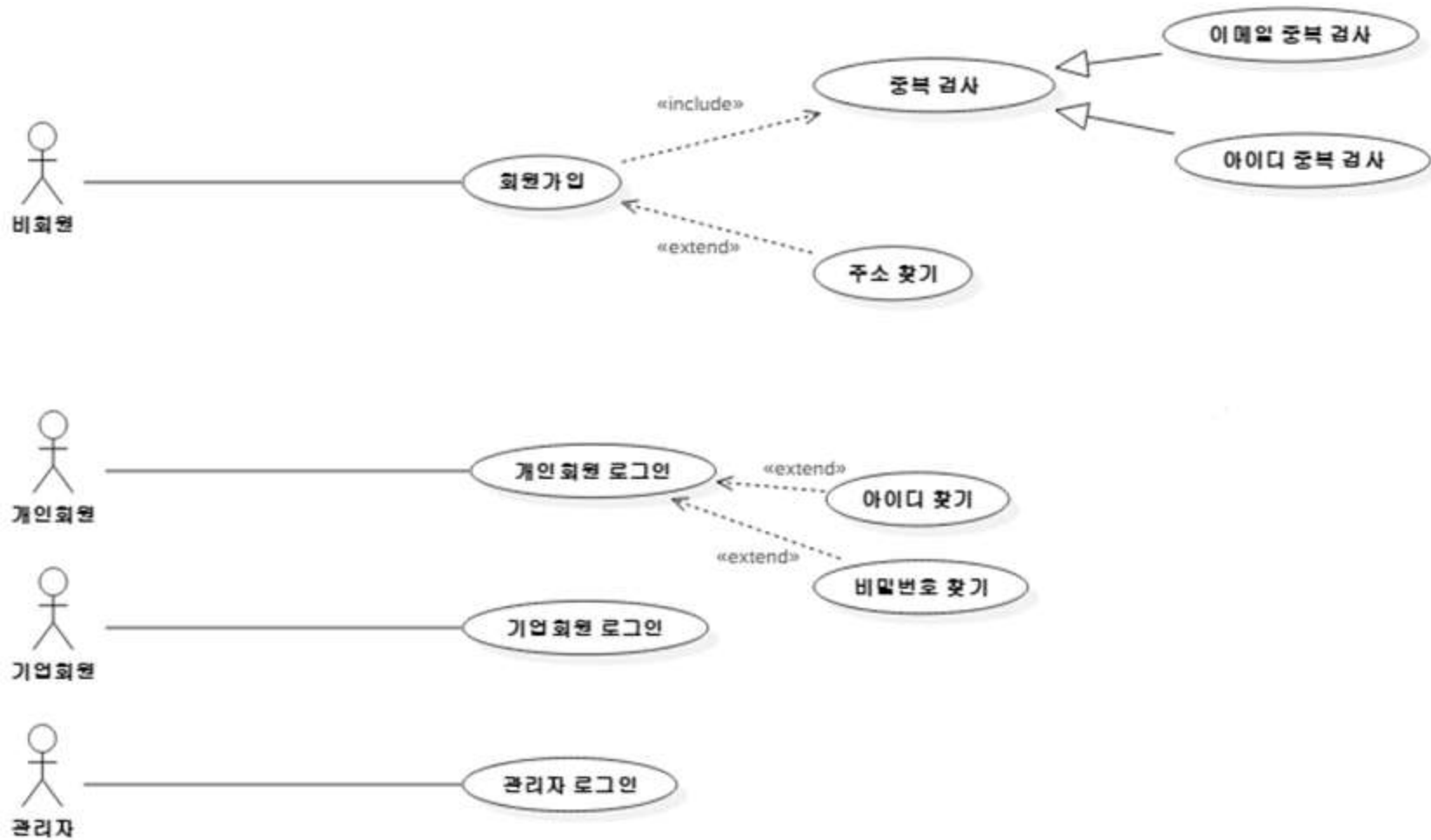
비회원



회원



시스템 연속



시스템 접속 해제

