

유스케이스 기반의 요구사항 분석

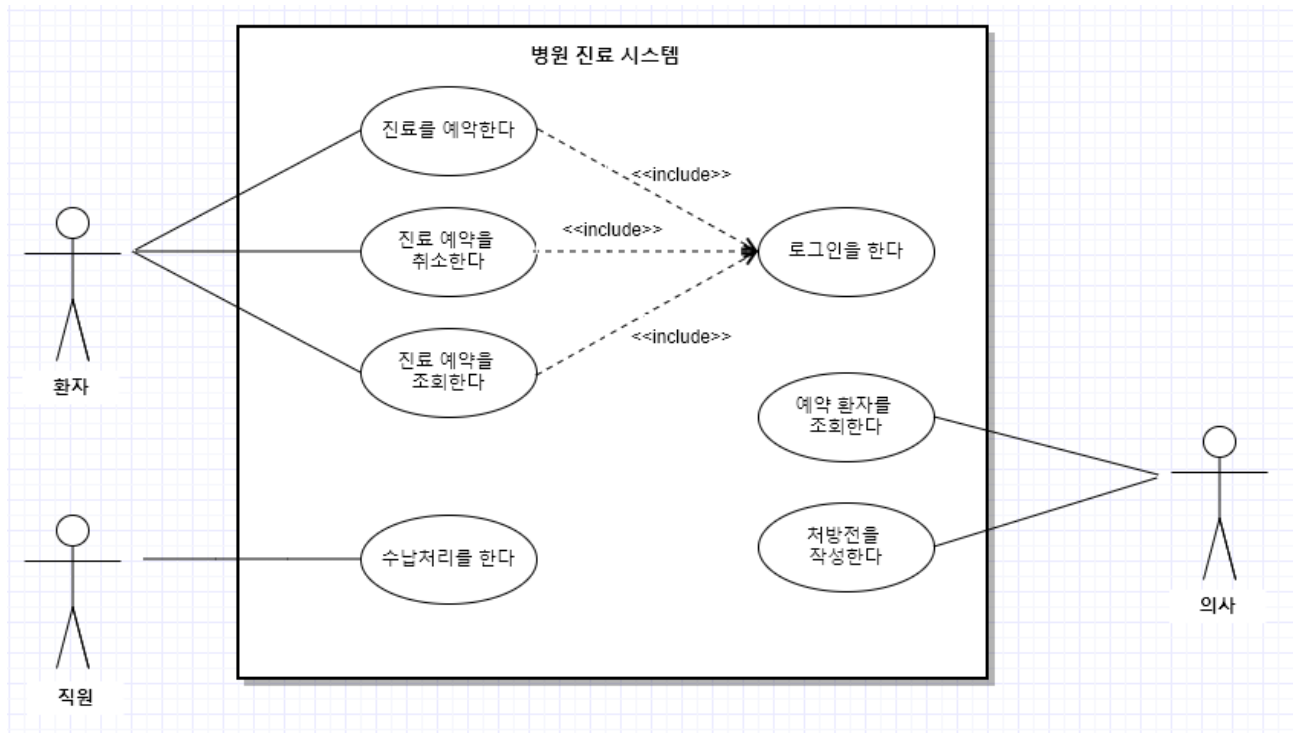
유스케이스(UseCase)는 말 그대로 풀어보면 "사용 사례"로서 시스템이 어떻게 사용될지에 대해 표준화된 규칙에 따라 다이어그램으로 표현한 것이다. 고객과 개발자는 표준화된 규칙만 이해하고 있다면, 유스케이스 다이어그램(UseCase Diagram)을 통해 시스템의 기능이나 상호 작용 등을 쉽게 이해할 수 있다. 유스케이스를 이용한 요구사항 분석 방법은 통일된 규칙으로 시스템을 표현하고, 이를 통해 고객 및 사용자, 개발자의 다양한 관점을 찾아내어 그 간격을 조정하는 것을 말한다. 이러한 유스케이스 기반의 요구사항 분석은 시스템을 가시화시켜 주는 모델링 기법인 UML(Unified Modeling Language)을 이용하게 된다.

1. UML(Unified Modeling Language)

UML은 요구사항의 추출부터 분석, 명세를 통하여 생성되는 산출물들을 표준화된 다이어그램으로 연결 시킴으로써, 고객 및 사용자, 개발팀의 다양한 관점을 다룰 수 있는 효과적인 도구이다. 이러한 도구를 통해 모든 참여자들 간의 효율적인 의사소통을 돕고 요구사항의 누락사항이나 불일치를 지적할 수 있다.

2. 유스케이스 다이어그램

유스케이스 다이어그램(UseCase Diagram)은 사용자 관점에서 시스템의 서비스와 기능 및 그와 관련된 외부 요소를 보여주는 다이어그램이다. 고객과 개발자는 이 다이어그램을 보며 요구사항에 대한 의견을 조율할 수 있다.

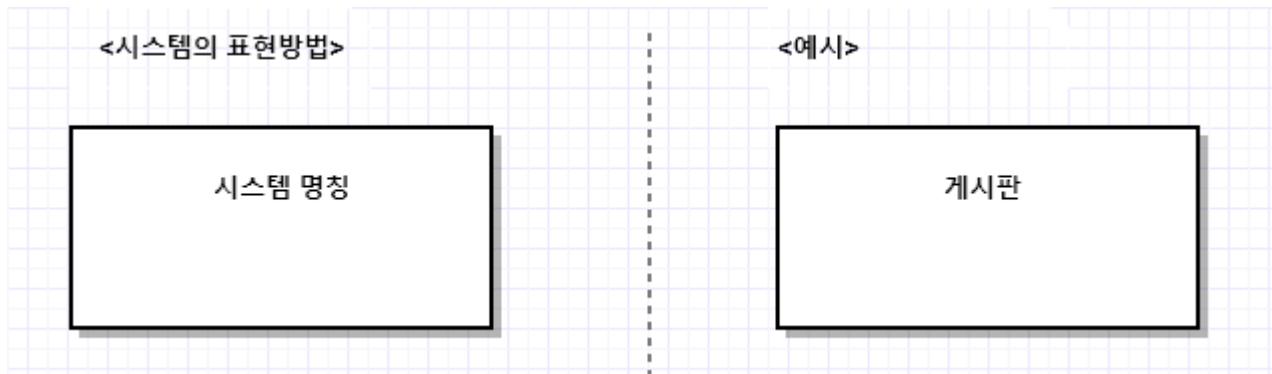


(1) 구성 요소와 표기법

유스케이스 다이어그램을 구성하는 4가지 요소인 시스템(System), 액터(actor), 유스케이스(UseCase), 관계(Relationship)의 의미와 표기법을 알아보자.

- 시스템(System)

시스템이란 만들고자 하는 시스템의 범위를 뜻한다. **시스템을 표현하기** 위해서는 유스케이스를 둘러싼 사각형의 틀을 그리고, 시스템 명칭을 사각형 안쪽 상단에 기술한다.



- 액터(actor)

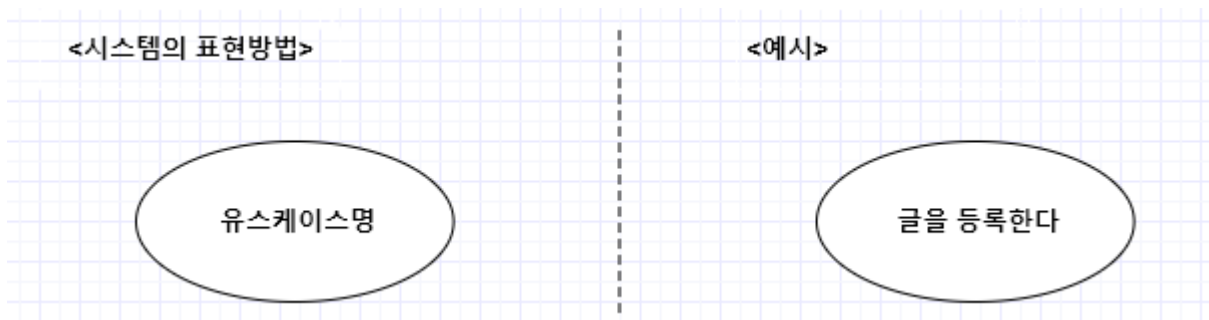
액터는 시스템의 외부에 있으면서 시스템과 상호 작용을 하는 **사람 또는 다른 시스템**을 말한다.



액터는 원과 선을 조합한 형태의 사람 모양으로 표현하고, 그 위 또는 아래에 액터명을 표시한다. 액터명을 정할 때에는 액터의 역할을 반영한다. 예를 들면 관리자, 사용자 등이 액터명으로 사용될 수 있다.

- 유스케이스(UseCase)

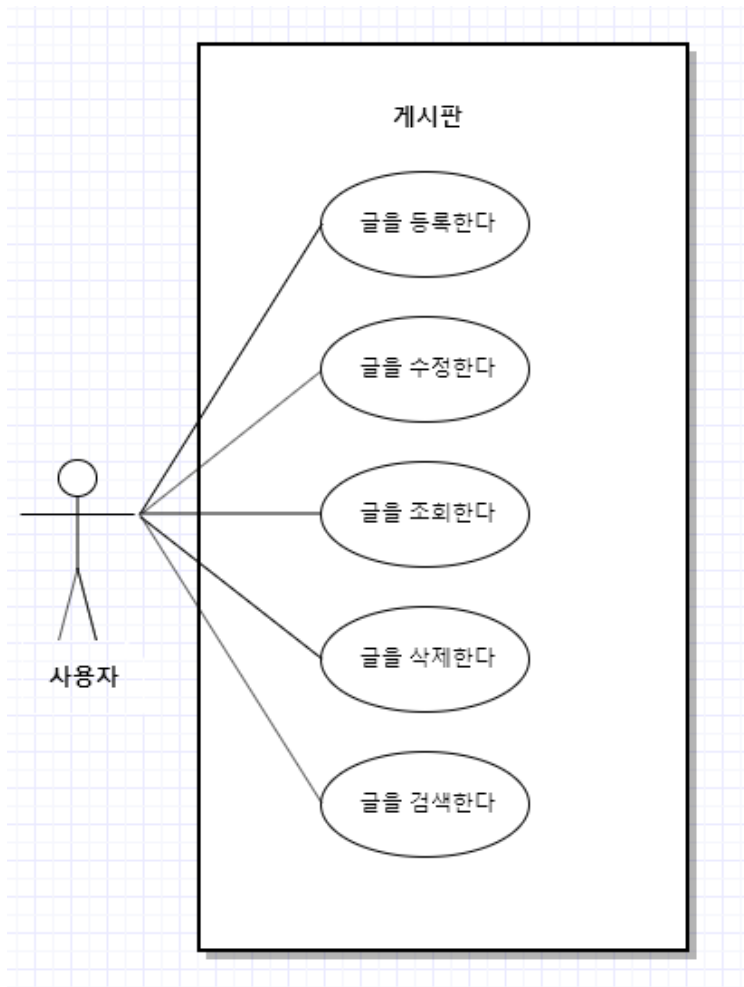
유스케이스(UseCase)는 시스템이 액터에게 제공해야 하는 기능의 집합이다. 따라서 **유스케이스는 시스템의 요구사항을 보여준다**. 유스케이스는 타원으로 표시하고, 그 안쪽이나 아래쪽에 유스케이스명을 기술한다. 유스케이스명은 "글을 등록한다", "글을 조회한다"와 같이 동사로 표현하여, 유스케이스 하나가 개발될 기능 하나와 연결될 수 있도록 한다.



- 관계(Relationship)

시스템의 요구사항을 액터와 유스케이스 간에 상호 작용이 존재함을 나타낸다. 예를 들어, "사용자"라는

액터가 "글을 등록한다"라는 유스케이스와 상호작용하는 것은 연관관계로 나타낼 수 있는데 이때 액터와 유스케이스 사이에 실선을 연결하여 표현한다.



- 포함관계(include)

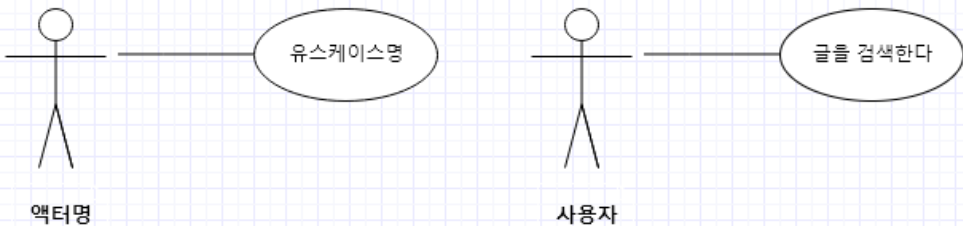
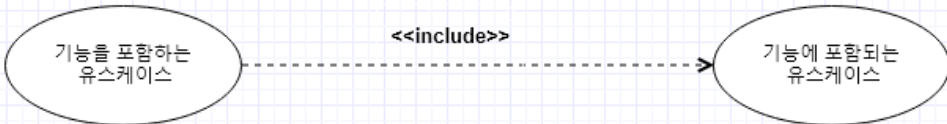
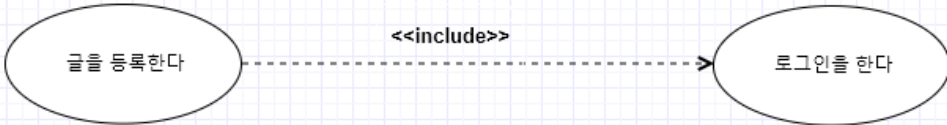
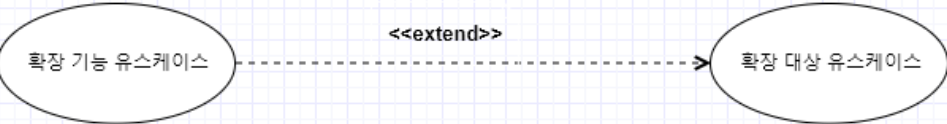
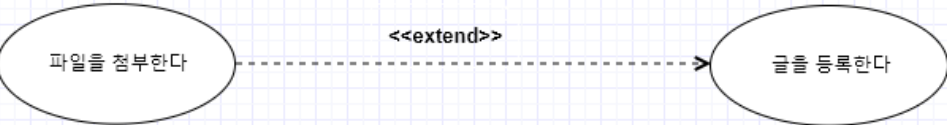
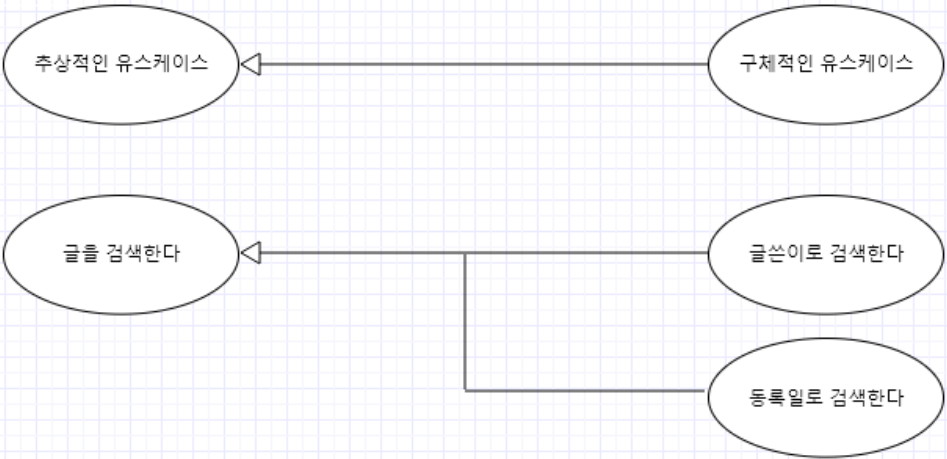
포함관계는 하나의 유스케이스가 다른 유스케이스의 실행을 전제로 할 때 형성되는 관계이다. 포함되는 유스케이스는 포함하는 유스케이스를 실행하기 위해 반드시 실행되어야 한다. 예를 들어 "글을 등록한다"라는 유스케이스를 실행하기 위해서는 "로그인을 한다"라는 유스케이스가 먼저 실행되어야 하므로 포함관계로 나타낼 수 있다. '포함하는 유스케이스'에서 '포함되는 유스케이스' 방향으로 화살표를 점선으로 연결하여 표현하고 <<include>>라고 표기한다.

- 확장관계(extend)

확장관계는 확장 기능(extending) 유스케이스와 확장 대상(extended) 유스케이스 사이에 형성되는 관계이다. 확장 대상 유스케이스를 수행할 때에 특정 조건에 따라 확장 기능 유스케이스를 수행하기도 하는 경우 적용된다. 예를 들어 "글을 등록한다"라는 유스케이스는 일반적으로 실행되지만, "파일을 첨부한다"라는 유스케이스는 글을 등록하는 경우 중 첨부할 파일이 있는 조건의 경우에 실행되지만, 그렇지 않은 경우 실행되지 않기도 하므로 확장관계로 나타낼 수 있다. "글을 등록한다"라는 유스케이스는 '확장 대상 유스케이스'가 되고 "파일을 첨부한다"라는 유스케이스는 '확장 기능 유스케이스'가 된다. '확장 기능 유스케이스'에서 '확장 대상 유스케이스' 방향으로 화살표를 점선으로 연결하여 표현하고 <<extend>>라고 표기한다.

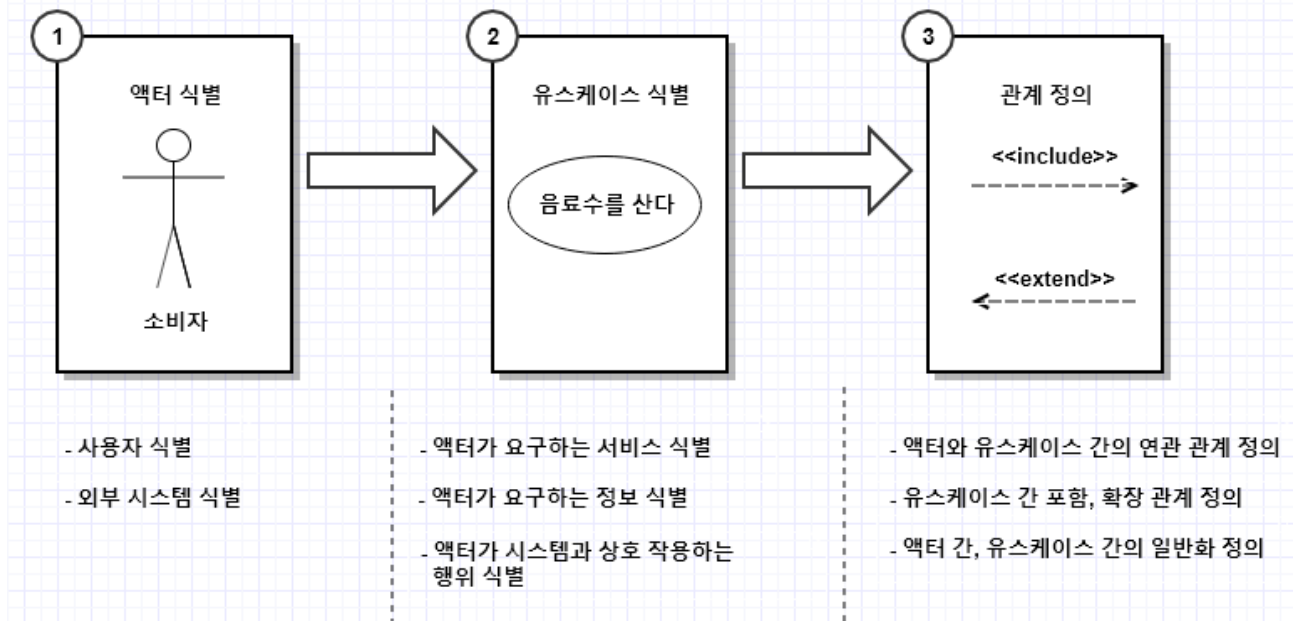
- 일반화관계(Generalization)

일반화관계는 유사한 유스케이스들 또는 액터들을 모아 그들을 추상화(주어진 문제나 시스템을 중요하고 관계 있는 부분만 분리해 내어 간결하고 이해하기 쉽게 만드는 작업.)한 유스케이스 또는 액터와 연결시켜 그룹핑(Grouping)함으로써 이해도를 높이기 위한 관계이다. 예를 들어 "글쓰기로 검색한다"와 "등록일로 검색한다"라는 유스케이스가 있는 경우, 독립적으로 존재하는 것보다 "글을 검색한다"라는 추상적인 유스케이스를 만들어 연결시킴으로써 이해도를 높인다. '구체적인 유스케이스'에서 '추상적인 유스케이스' 방향으로 끝부분이 삼각형의 테두리로 표현된 화살표를 실선으로 연결하여 표현한다.

관계		표기법	
연관관계			
의존 관계	포함관계		
			
	확장관계		
			
일반화관계			

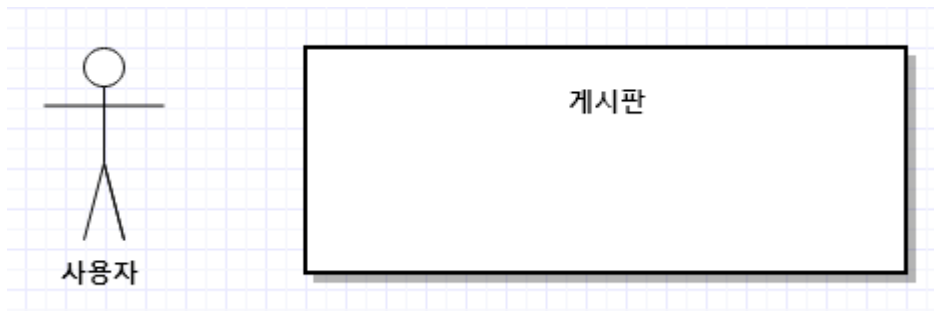
(2) 유스케이스 다이어그램 작성

개발하고자 하는 시스템을 유스케이스 다이어그램으로 표현하는 것을 유스케이스 모델링(UseCase Modeling)이라고 한다.



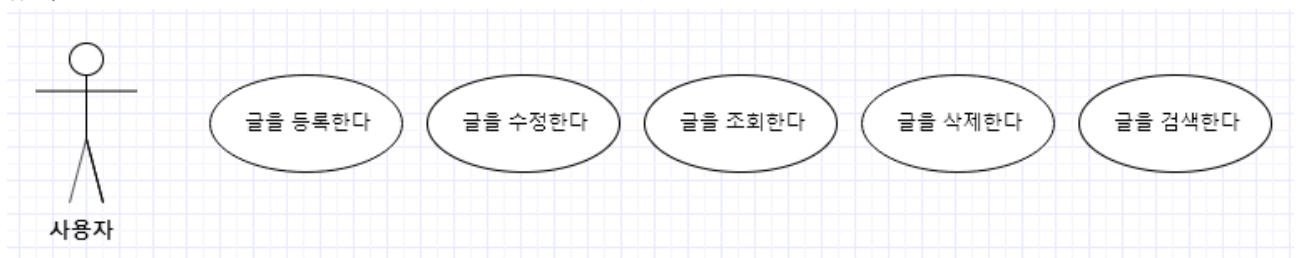
먼저 시스템과 상호작용하는 액터를 식별하고, 액터가 시스템에게 요구하는 유스케이스를 식별한 후, 액터와 유스케이스, 유스케이스와 유스케이스의 관계 등을 정의하여 유스케이스 다이어그램을 작성한다.

• 액터 식별



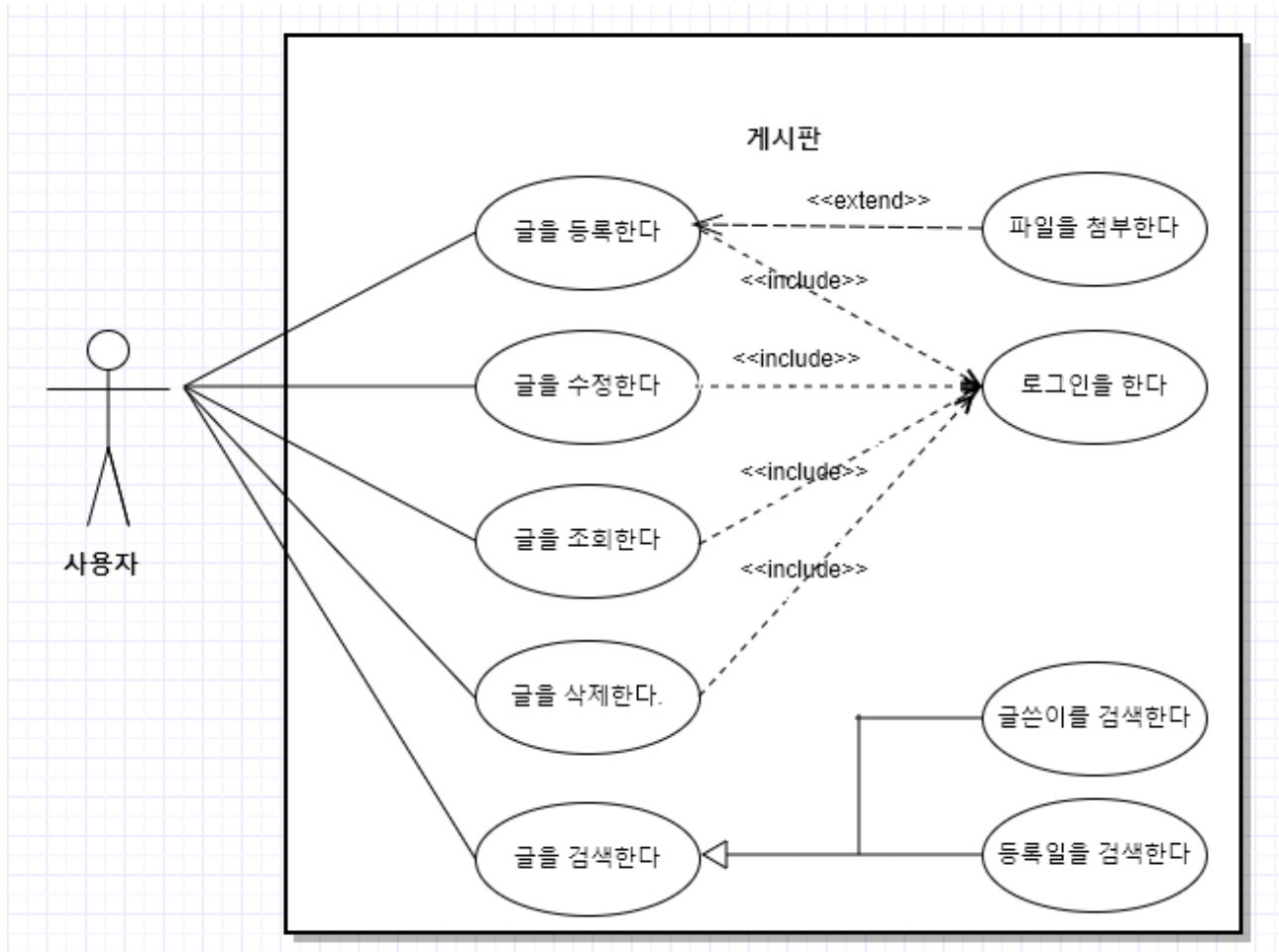
• 유스케이스 식별

식별된 액터가 게시판을 통해 하고자하는 기능, 즉 유스케이스를 식별해 보자. 사용자는 게시판을 통해 글을 등록, 수정, 삭제, 조회 등의 작업을 하고자 한다. 그러므로 유스케이스는 다음과 같이 식별될 수 있다.



- 관계 정의

액터와 유스케이스가 식별되었으므로, 이제 이들 사이의 관계를 살펴보자.



(4) 유스케이스 다이어그램 예제 2 (인터넷 서점 쇼핑몰)

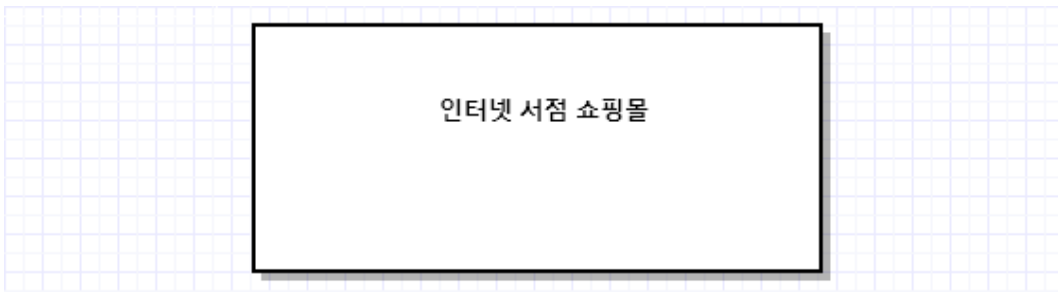
- 사용자 요구사항

: 인터넷 상에서 도서를 구매할 수 있는 쇼핑몰을 개발한다.

- 고객은 회원등록과 로그인 과정을 거쳐 쇼핑몰을 이용할 수 있다.
- 회원등록을 할 때에는 반드시 실명확인을 한다.
- 도서를 구입할 때에는 현금결제 혹은 카드결제를 할 수 있다.
- 카드결제는 외부 시스템을 사용한다
- 구입한 도서의 배송상황을 조회할 수 있다.
- 직원은 고객이 요청하는 상담을 처리할 수 있다.

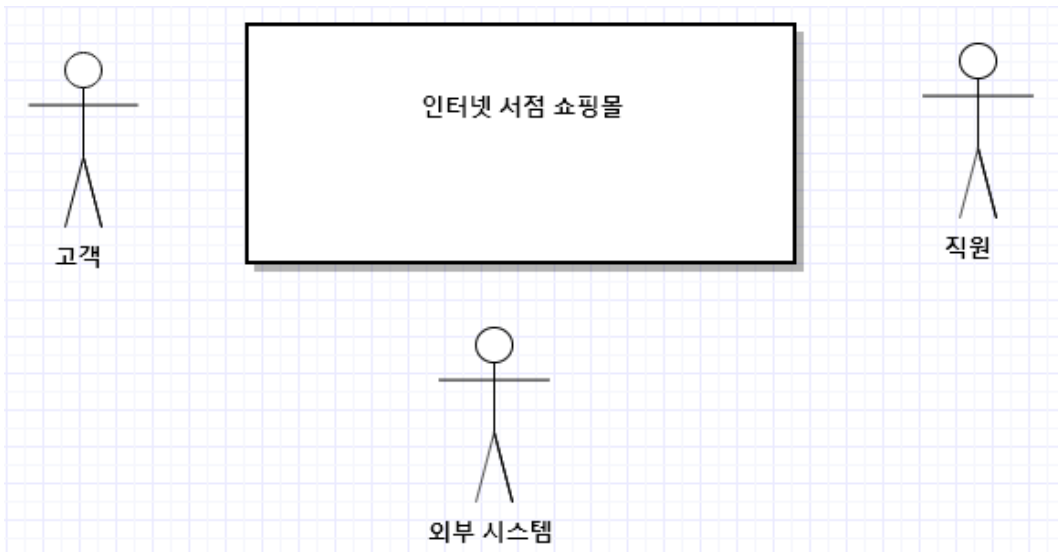
- 시스템 식별

먼저 개발할 시스템을 식별해 보면 그 대상은 '인터넷 서점 쇼핑몰'이 된다. 인터넷 서점 쇼핑몰 시스템은 다음과 같이 표현할 수 있다.



• 액터 식별

시스템인 '인터넷 서점 쇼핑몰'과 상호 작용하는 액터를 식별해 보자. 책을 사는 역할을 하는 고객, 고객 관리의 역할을 하는 직원, 그리고 전자결제 처리의 역할을 하는 전자결제 시스템이 인터넷 서점 쇼핑몰의 시스템과 상호 작용을 한다.

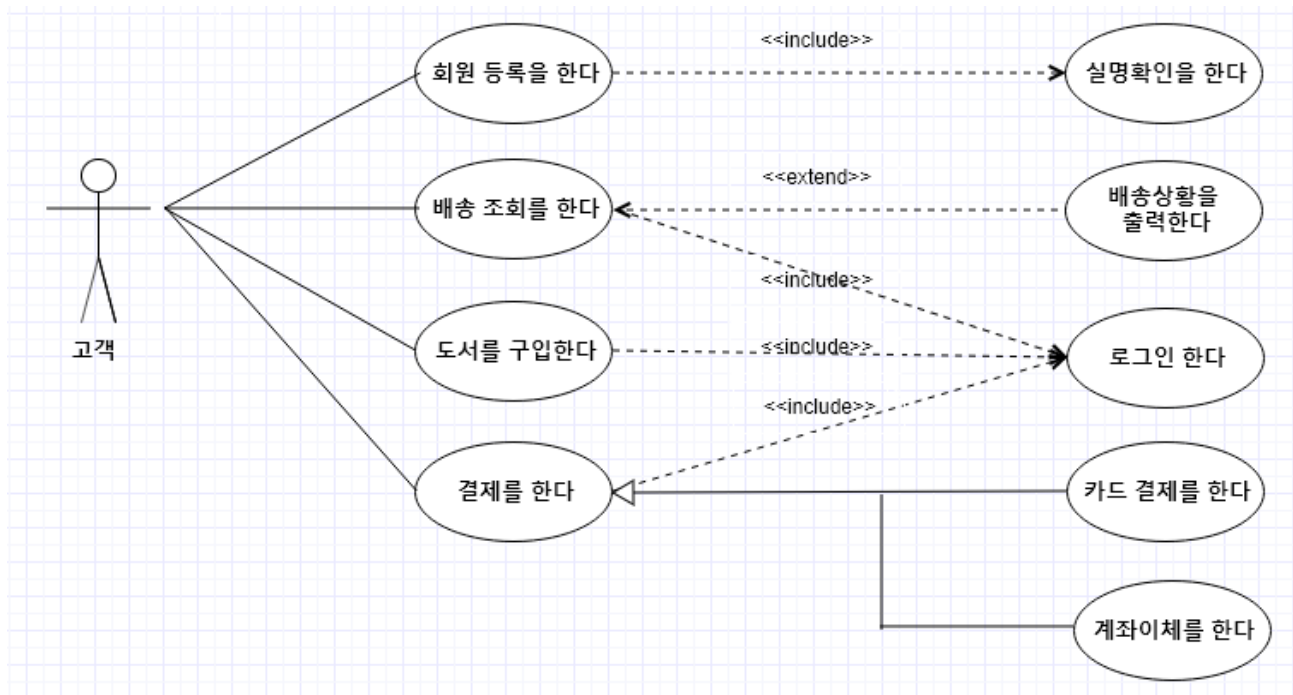


• 유스케이스 식별

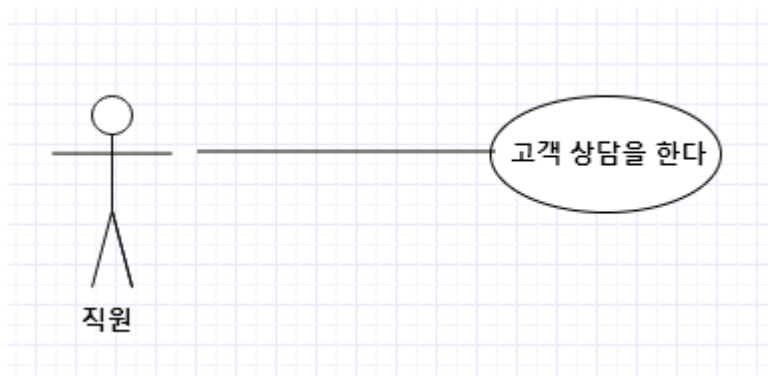


• 관계 정의

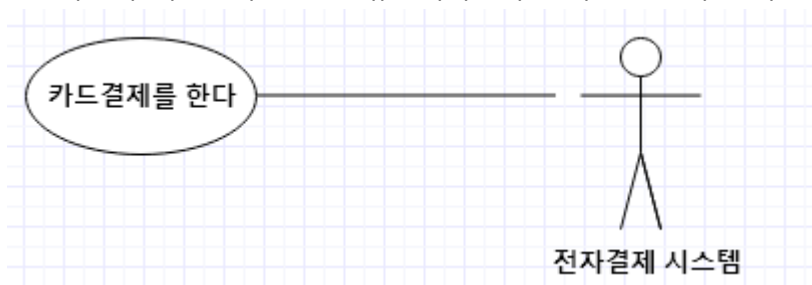
- 고객과 관련된 유스케이스의 관계를 표현하면 다음과 같다.



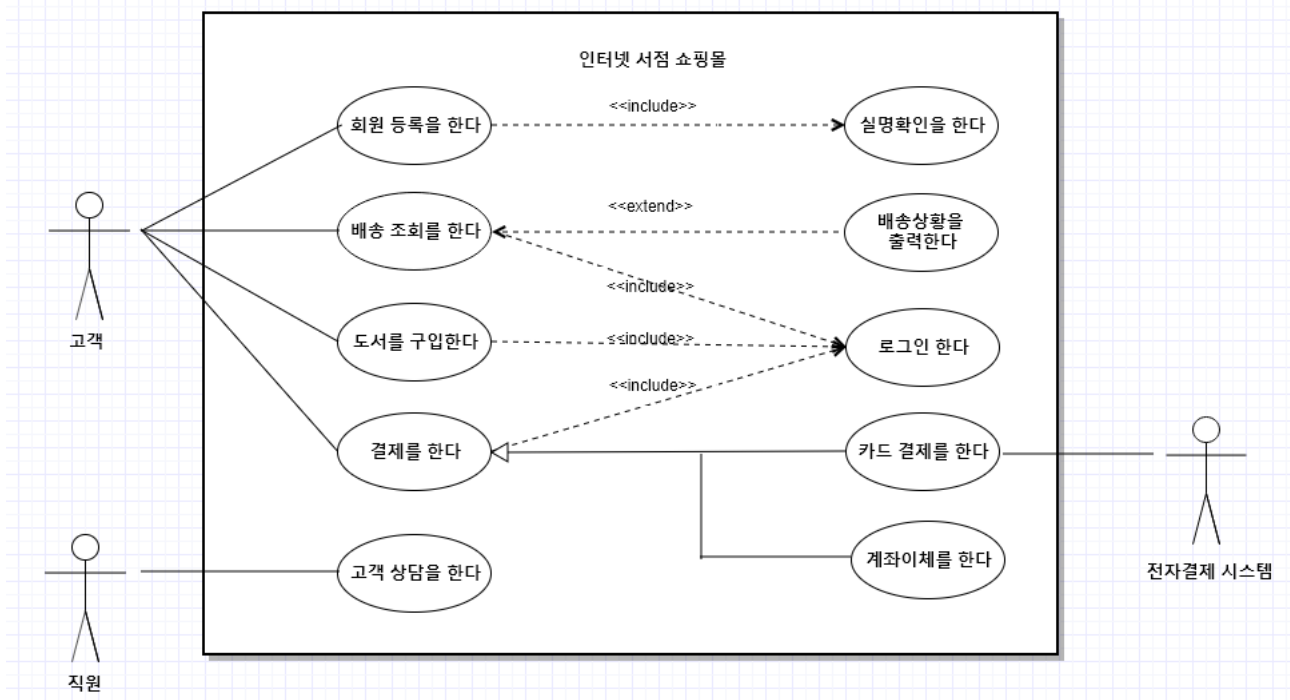
- 직원과 관련된 유스케이스의 관계를 표현하면 다음과 같다.



- 전자결제 시스템과 관련된 유스케이스의 관계를 표현하면 다음과 같다.



이와 같은 과정을 통해 유스케이스 다이어그램을 완성히 보면 다음과 같다.



(5) 유스케이스 기술서 작성

유스케이스 다이어그램에서 유스케이스는 시스템이 수행해야 할 기능이 무엇인지에 대해서만 표현한다. 이를 보완하고자 각 유스케이스에 대해서 해당 유스케이스가 어떻게 수행되는지를 정의한 유스케이스 기술서를 작성하게 된다. 유스케이스 기술서는 각 유스케이스의 기능을 명확히 정의하고 작업의 흐름 및 제약 사항등을 정의한다.

일반적으로 다음과 같은 항목을 기술서에 작성한다.

- 유스케이스명
- 액터명
- 유스케이스 개요 및 설명
- 사전 및 사후 조건
- 작업 흐름
 - 정상 흐름(Normal Flow): 해당 유스케이스가 정상적으로 수행되는 흐름.
 - 대안 흐름(Alternative Flow): 유스케이스 내의 작업 흐름이 수행되는 중에 특정 시점에서 여러 가지 선택적인 흐름으로 나뉘어 질 경우에 발생하는 흐름.
 - 예외 흐름(Exception Flow): 유스케이스 내의 작업 흐름이 수행되는 중에 발생할 수 있는 예외상황이나 오류를 표현하는 흐름.
- 시나리오: 각 시나리오는 유스케이스의 특정한 예를 나타냄.

다음은 앞서 살펴본 유스케이스 다이어그램 작성 예제 중, 사용자가 게시판에 글을 등록하는 유스케이스를 기술서에 정의한 예이다

- 유스케이스명: 글을 등록한다.
- 액터명: 사용자
- 유스케이스 개요 및 설명
 - 사용자는 원하는 글을 게시판을 등록한다.
- 사전 조건: 로그인한다.
- 작업 흐름
 - 정상 흐름(Normal Flow)
 1. 사용자는 글쓰기 버튼을 클릭한다
 2. 시스템은 글쓰기 화면을 실행한다.
 3. 사용자는 글쓰기 화면에서 원하는 글을 작성한다.
 4. 사용자는 등록버튼을 클릭한다
 5. 시스템은 글을 데이터베이스 게시판 테이블에 저장한다.
 - 대안 흐름(Alternative Flow):
 1. 정상흐름4에서 등록 취소버튼을 선택할 경우, 시스템을 게시판 목록 조회 화면을 보여준다
 - 예외 흐름(Exception Flow)
 1. 정상흐름3에서 글쓰기 화면에서 글을 쓰지 않고 등록 버튼을 클릭할 경우, 시스템은 "내용을 입력해 주세요"라는 경고 메시지를 보여준다.