



유스케이스 다이어그램

1. 유스케이스 다이어그램의 구성요소와 표현

- 유스케이스 다이어그램의 사용 목적: 사용자의 요구사항 분석
 - 누가 시스템을 사용할 것인가
 - 시스템은 사용자를 위해 무엇을 해야 하는가
 - 사용자와 상호작용하기 위해 시스템이 제공해야 할 인터페이스는 무엇인가
- **액터: 개발할 시스템 외부의 존재**
- **유스케이스: 시스템 내부에 해당되는 단위 기능**

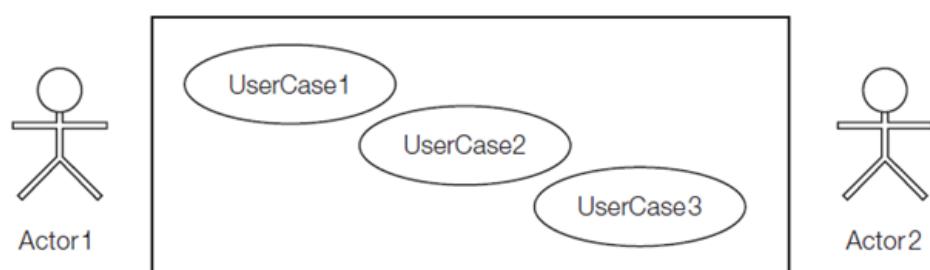


그림 3-1 시스템 경계와 개발 대상
사각 창은 시스템 경계이고, 경계 내부에 위치한 UserCase1, 2, 3은
구축할 시스템의 기능. 시스템 경계의 외부에 위치한 Actor1, 2는
외부 존재로 개발 대상에 미포함

- 사용자 관점에서 시스템을 모델링하는 것이 유스케이스의 역할임
- 유스케이스는 시스템에 대한 시나리오의 집합으로 볼 수 있음
 - 각 시나리오에는 발생되는 이벤트의 흐름이 나타나 있음
 - 각 이벤트의 흐름은 사람, 시스템, 하드웨어 혹은 시간의 흐름에 의해 시작됨
 - 이벤트 흐름을 시작하는 객체가 액터임, 이 흐름은 해당 액터 혹은 다른 액터에게 어떤 의미가 생기는 결과를 제공함

- 유스케이스 다이어그램에는 다양한 관계가 존재함
 - 포함include 관계: 다른 유스케이스에서 기존 유스케이스를 재사용할 수 있음
 - 확장extend 관계: 기존 유스케이스에 진행 단계를 추가하여 새로운 유스케이스를 만들어내는 관계
 - 일반화 관계

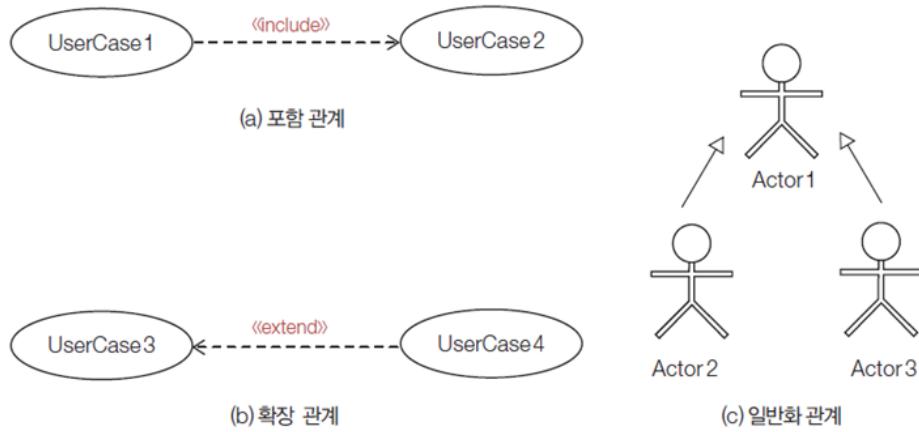


그림 3-3 유스케이스 다이어그램에서 사용되는 관계들

2. 유스케이스 다이어그램의 관계

1) 액터와 유스케이스 사이의 관계

- 액터와 유스케이스 사이에는 연관 관계(—로 표시)가 존재함
- 액터: 사용자, 시스템 유지/관리자, 외부 시스템

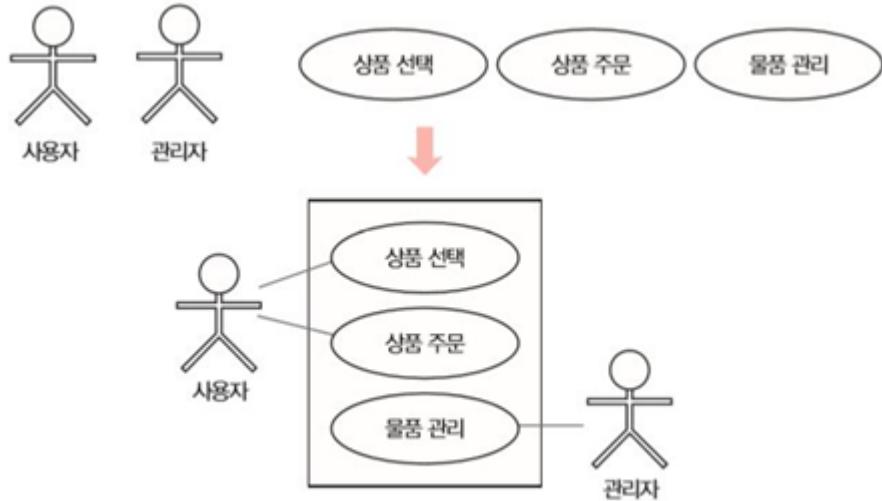


그림 3-5 액터와 유스케이스의 연관 관계

2) 유스케이스 사이의 관계

- 유스케이스 간의 관계는 의존 관계(- - ->로 표시)이며, 스테레오타입으로 그 관계가 명확히 표현됨
- **포함 관계**
 - 하나의 유스케이스를 수행할 때, 같은 기능이 있는 다른 유스케이스가 반드시 수행 되는 관계
 - 예) 고객이 자판기에 동전을 투입하면 금액이 자동으로 표시됨
 - 여러 유스케이스에서 공통으로 발견되는 시스템의 기능을 표현
 - 여러 유스케이스에 나타난 공통적인 이벤트 흐름을 별도의 유스케이스로 표현함으로써 이벤트 흐름의 중복을 방지할 뿐 아니라 시스템의 공통 기능을 추출하여 효율적으로 개발할 수 있음

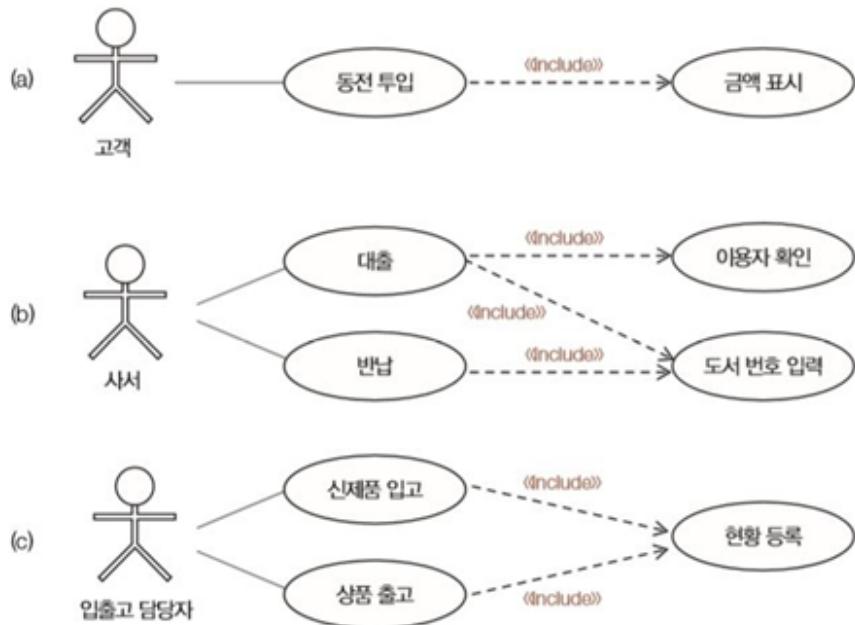


그림 3-6 유스케이스 간 포함 관계 예

• 확장 관계

- 여러 유스케이스에 걸쳐 중복되지 않고, 특정 조건에서 한 유스케이스로만 확장되는 것
- 상위 유스케이스로부터 어떤 특정한 조건에 의해 수행됨
- 한 유스케이스에서 추가되거나 확장되는 기능을 표현하기 위해서 사용
 - 기본 유스케이스를 수정하지 않고 새로운 요구사항을 추가로 표현하고자 할 때 사용함
 - 확장 유스케이스는 기본 유스케이스의 속성을 변경하여 수행할 수 있지만, 기존 유스케이스는 확장 유스케이스의 속성을 변경할 수 없음
 - 화살표 방향 주의(포함 관계와 반대); 기준 ← — 확장

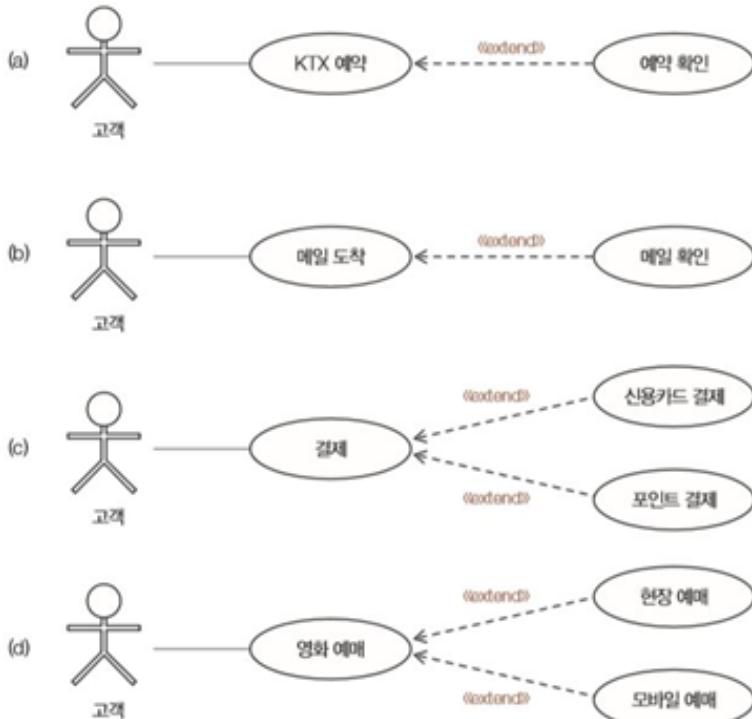


그림 3-7 확장 유스케이스

- 포함 관계 vs 확장 관계

- 포함 관계: 이벤트 흐름이 포함 유스케이스로 항상 분기됨
 - 대부분 포함 유스케이스의 수행 결과에 따라서 기준 유스케이스의 이벤트 흐름이 영향을 받음
- 확장 관계: 기준 유스케이스에 기술된 조건이 만족되는 경우에만 확장 유스케이스의 이벤트 흐름으로 분기됨
 - 기준 유스케이스 이후의 이벤트 흐름은 확장 유스케이스의 수행 결과에 의존하지 않음

3) 액터 사이의 관계

- 일반화 관계

- 액터들이 유스케이스와 중복하여 관계를 나타나면 액터들을 통합하여 일반화 관계로 표현할 수 있음
 - 액터들 간의 유스케이스 차이는 별도 문서에서 기술하면 됨
- 추상적인 액터와 좀 더 구체적인 액터 사이에 관계를 맺어줌

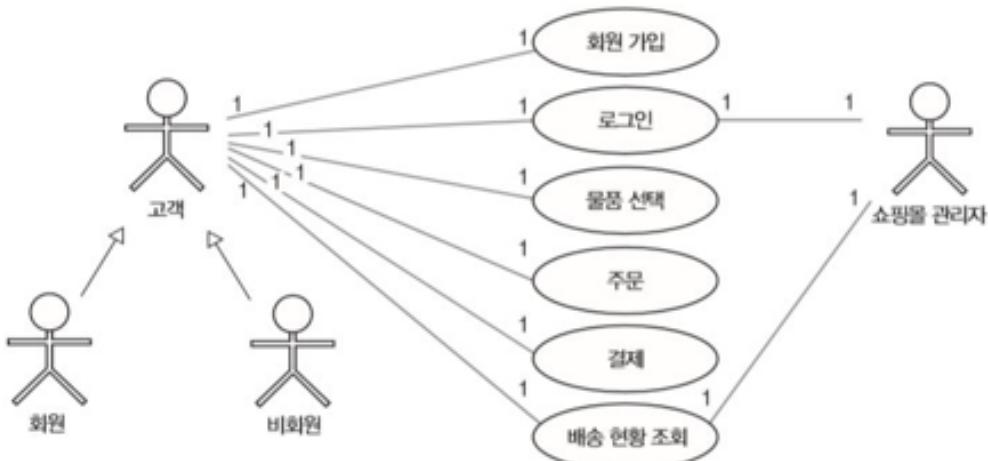


그림 3-13 일반화 관계 적용 전과 후 비교

- 중복 관계

- 유스케이스 모델링할 때 유스케이스 이벤트 흐름에서 중복된 부분이 발생할 수 있음. 유스케이스 사이에 포함 관계를 설정하여 해결함

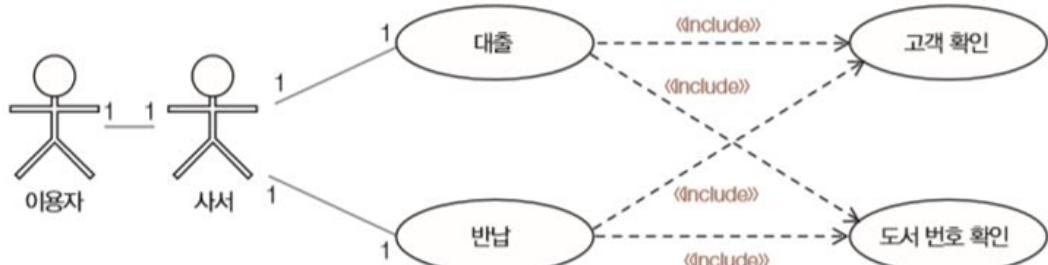


그림 3-15 중복이 제거된 도서 관리 시스템의 유스케이스 다이어그램

3. 유스케이스 다이어그램의 단계별 모델링

1) 시스템 상황 분석

- 시스템 상황을 분석하여 문제 기술서를 작성함
- 시스템의 제목을 결정한다고 생각하면 됨

2) 액터 식별

- 행위자와 그들의 책임을 확인함
- 액터는 다음과 같은 질문으로 찾을 수 있음

- 시스템의 주요 기능을 사용하는 사람은 누구인가
- 시스템을 지원하기 위해 필요한 사람은 누구인가
- 시스템을 유지하고 관리하는 사람은 누구인가
- 시스템에 필요한 하드웨어 장치는 무엇인가
- 시스템과 상호작용하는 다른 시스템은 무엇인가
- 시스템의 처리 결과에 연결되는 사람 또는 사물은 무엇인가
- 액터는 외부에서 시스템에 접근할 수 있는 사람이나 시스템과 관련된 외부 시스템을 의미함
 - 액터의 명칭으로 특정 사람의 이름보다는 역할을 의미하는 이름을 사용함
 - 예) 고객, 관리자, 시스템 등

3) 유스케이스 식별

- 액터 관점에서 시스템의 기능을 확인함
- 유스케이스는 개발을 위한 시스템의 기능을 의미함. 즉, 시스템을 수행하는 일련의 행위임
- 이는 시스템에서 제공해야하는 독립적인 기능을 말하며, 이 기능은 외부 시스템과 상호 작용하는 행위를 하는 기능만 가리킴
- 유스케이스는 행위 자체만 표현할 뿐이며, 이때 행위 과정은 기술할 필요가 없음
 - 행동에 대한 구체적인 수행 방법은 규정하지 않음
- 유스케이스를 식별하려면 먼저 개발할 시스템이 기능 요구 명세서를 살펴봐야 함. 액터를 중심으로 시스템에 원하는 기능이 무엇인지를 찾아 이를 유스케이스로 추출함
- 기능을 나타내도록 동사형 명사를 사용
 - 예) 성적 등록, 성적 조회 등

4) 유스케이스 다이어그램 작성

- 액터와 유스케이스, 유스케이스와 유스케이스의 관계를 설정해 표현함
- 액터와 정보를 주고받는 유스케이스를 찾아 연관 관계를 설정함
- 유스케이스는 의존 관계로 표현되며, <<include>>, <<extend>> 스테레오타입을 그 관계가 명확하게 표현됨
 - <<include>>: 하나의 유스케이스를 수행할 때 같은 기능을 가진 또 하나의 유스케이스를 반복적으로 반드시 수행해야 할 경우

- <<extend>>: 여러 유스케이스에 걸쳐 중복 사용되지 않고, 특정 조건에서 한 유스케이스로만 확장되는 것
- 액터의 일반화 관계를 찾는다

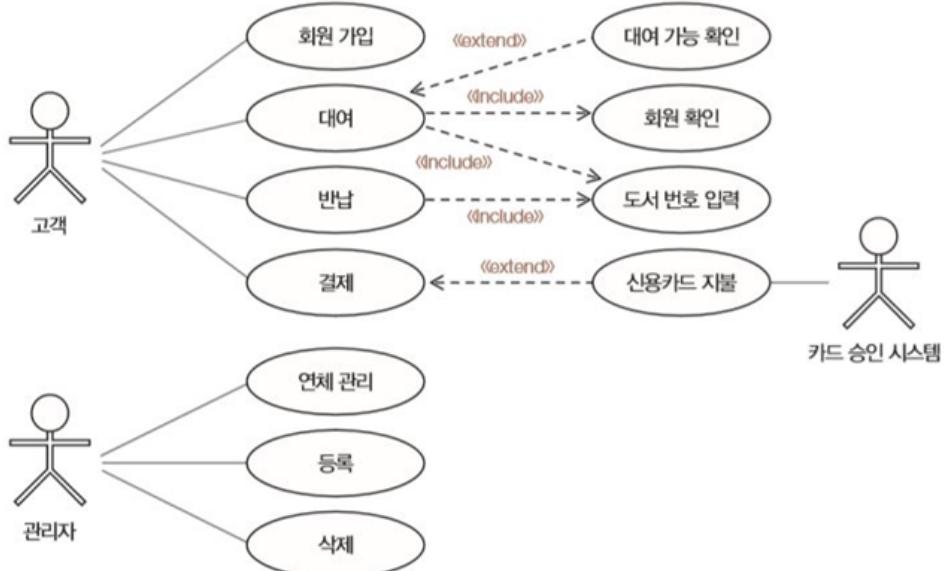


그림 3-21 깨비책방 관리 시스템의 유스케이스 다이어그램

5) 유스케이스 명세서 작성

- 유스케이스 다이어그램에서 유스케이스는 시스템이 수행해야 할 기능이 무엇인지만 표현하고, 유스케이스 내부에 대한 기술은 유스케이스 명세서로 기술함
 - 유스케이스는 시스템의 기능을 표현
 - 유스케이스 명세서는 각 유스케이스가 수행되는 방법을 표현
- 유스케이스명, 액터명 및 개요를 기술함
- 사전 및 사후 조건과 제약사항들을 식별함
- 작업(정상, 대치, 예외) 흐름과 시나리오를 도출함
- 유스케이스 흐름에서 포함이나 확장 유스케이스로 구조화함

□ 유스케이스명 : 회원 가입
□ 액터명 : 고객(비회원)
□ 유스케이스 개요 : 고객이 깨비책방 관리 시스템을 사용하기 위해 회원 가입을 하는 유스케이스다.
□ 사전 조건 : 회원으로 가입되어 있지 않은 상태여야 한다.
□ 이벤트 흐름
- 정상 흐름
① 기존 가입 회원인지 이름과 전화번호를 검색하여 확인한다(시스템).
② 회원 가입을 요청한다(액터).
③ 회원 약관을 보여준다(시스템).
④ 회원 약관에 동의한다(액터).
⑤ 회원 정보 입력 항목을 보여준다(시스템).
⑥ 회원 정보, 항목(이름, 전화번호)을 입력하고 등록을 요청한다(액터).
⑦ 입력된 정보를 확인한다(시스템).
⑧ 회원 정보를 저장, 등록한다(시스템).
- 선택 흐름
▶ 이미 가입된 회원인 경우 “이미 가입된 회원입니다” 메시지를 보여준다.
▶ 회원 약관에 동의하지 않을 경우 약관 동의하에 회원 가입 가능 오류 메시지를 보여주고 동의를 요청한다.
▶ 회원 정보 입력 항목 중 입력하지 않은 항목이 있을 경우, 오류 메시지를 띄우고 재입력을 요청한다.
▶ 등록 번호의 형식이 틀린 경우 메시지를 보여주고 재입력을 요청한다.

6) 유스케이스 실체화

- 유스케이스 실체화는 도출된 유스케이스를 구현 시스템의 구성 요소로 구체화하는 작업임
- 즉, 명세서 중심의 유스케이스를 구현 시스템의 논리적 구성 요소인 클래스로 식별하고 그 통신 관계를 파악함
- 이를 위해 순차 다이어그램(이벤트 흐름)과 활동 다이어그램(화면 흐름)이 사용됨
- 유스케이스 다이어그램은 시스템 외부의 뷰를 표현하고, 실체화를 위해 작성하는 다이어그램은 유스케이스들이 어떻게 실현되는지를 객체들 간 메시지 흐름의 상호작용으로 설명함으로써 실체화된다고 볼 수 있음
- 요구사항 정의 활동의 산출물
 - 유스케이스 모델, 유스케이스 명세서, 이벤트 흐름 모델, 화면 흐름 모델 등 —> 요구사항 모델

표 3-6 요구 사항 정의 활동의 산출물

산출물	UML 다이어그램	필수 여부
요구 사항 모델	유스케이스 모델	유스케이스 다이어그램 필수
	유스케이스 명세서	이용 안 함 필수
	이벤트 흐름 모델	순차 다이어그램 선택
	화면 흐름 모델	활동 다이어그램 선택