

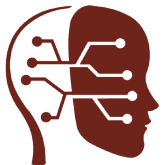
Software Engineering

Lecture 07: 요구분석 (Part 2)

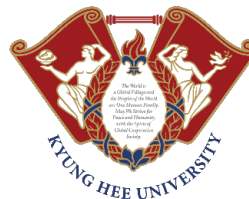
Professor: Jung Uk Kim

ju.kim@khu.ac.kr

Computer Science and Engineering, Kyung Hee University



Visual AI Lab.



요구분석의 절차 (Part 2에서..)

- 요구분석 절차

요구사항 도출

수집한 자료를 정리해 적절히 분류하여
개발에 필요한 요구사항을 도출

요구사항 분석

추출된 고객의 요구사항을 분석 기법을 이용하여
식별 가능한 문제들을 도출하고 요구사항 이해

문서화

도출한 요구사항을 요구분석명세서로 작성

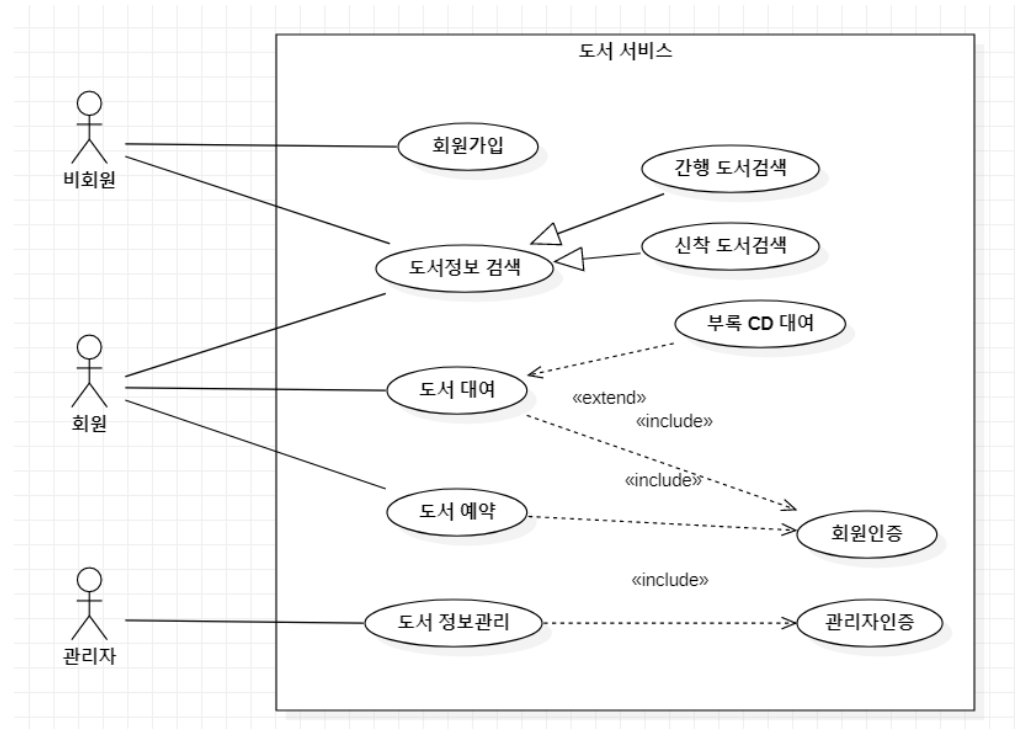
검증

사용자 요구가 정확히 기록이 되었는지, 빠뜨리지
않았는지 등 점검

(2) 요구사항 분석

• 유스케이스 (Use Case) 다이어그램

- 시스템 개발자와 고객 사이에 **요구를 이해하는 수단**이 됨
- **예외적인 케이스**를 개발자에게 주지시킴
- 대략적인 계획을 위하여 **기능의 수준을 파악**
- 구성
 - 시스템
 - 액터
 - 유스케이스 (Use case)
 - 관계

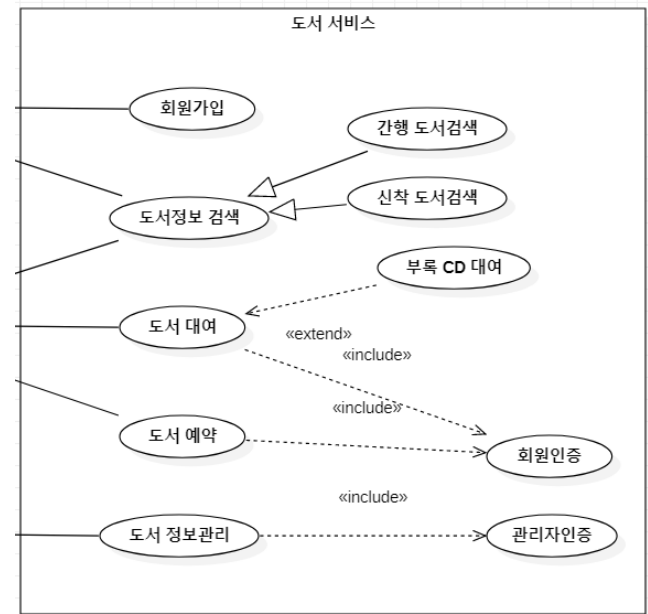


유스케이스 용어

- 시스템 (System)

- 만들고자 하는 프로그램
- 유스케이스를 둘러싼 사각형 틀 (안쪽 상단에 작성)

시스템 명칭



(예시)

유스케이스 용어

- **액터 (Actor)**

- 시스템을 **사용하는 사람**

- 예) 성적관리 사용 사례 → 성적과 관련 있는 기능을 사용하는 사람 (교수, 학생, 조교, 학사담당직원)

- 액터의 표기



교수



학생



조교



학사담당직원

액터명

- **주요 액터 (Primary Actor)** - 시스템의 왼쪽에 위치

- 행동을 하는 사람 (대부분의 액터가 여기에 해당)
 - 예) 학사관리시스템의 주요 액터: 시스템이 제공하는 기능을 사용하는 교수, 학생, 조교, 학사담당직원

- **시스템 액터 (System Actor)** - 시스템의 오른쪽에 위치

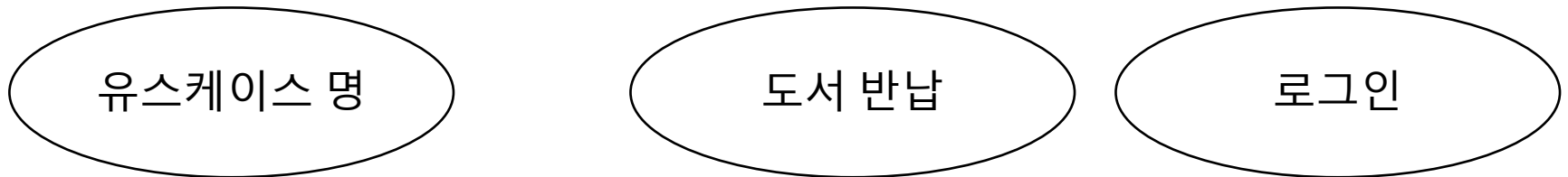
- 해당 프로그램 개발 범위에는 속하지 않지만 서로 연동되는 또다른 시스템

유스케이스 용어

- **유스케이스, 사용사례 (Use Case)**

- 사용자가 시스템을 통해 사용하고 싶은 기능
- 시스템이 액터에게 제공해야 하는 기능
- **시스템의 요구사항**
- 유스케이스 표기

(예시)



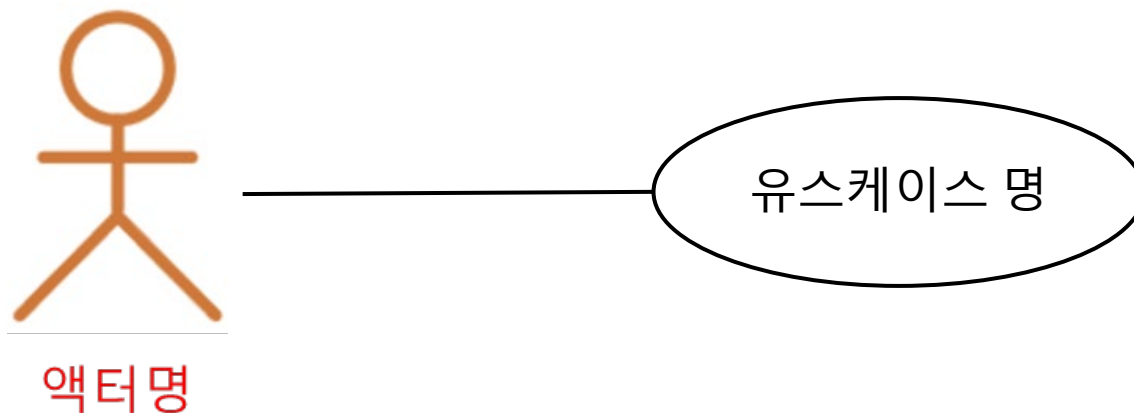
- **주의할 점**

- 유스케이스로 도출되지 않은 것은 시스템의 개발 범위에 포함할 수 없음
→ **유스케이스 도출 시 시스템 내 필요한 기능을 모두 도출해야함**

유스케이스 용어

• 관계 (Relationship)

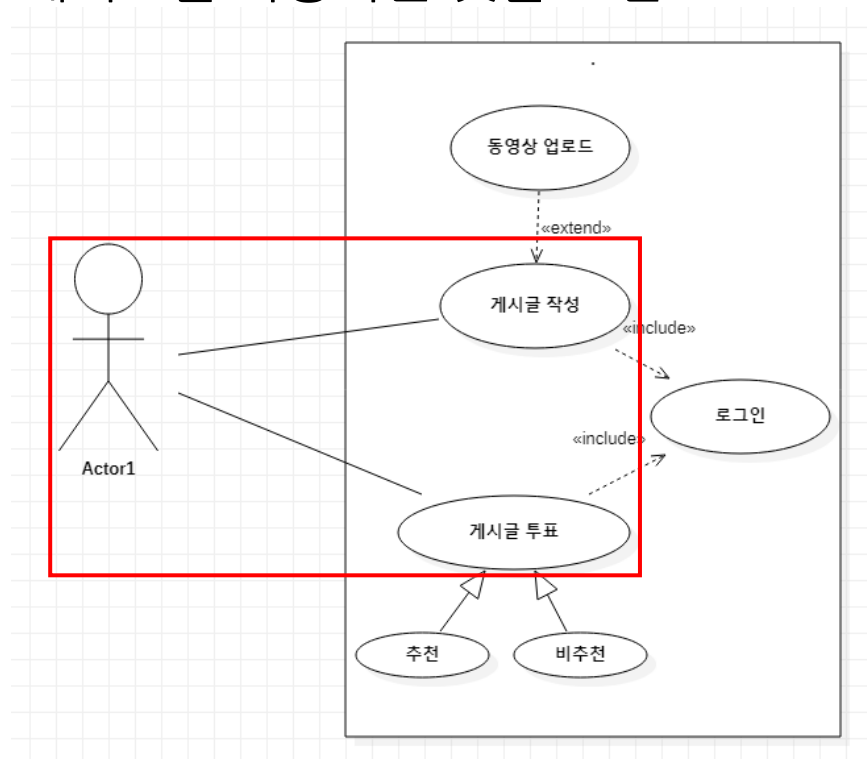
- 액터 및 유스케이스들 간의 의미있는 관계
- 관계의 종류
 - 연관 관계(Association)
 - 포함 관계(Include)
 - 확장 관계(Extend)
 - 일반화 관계 (Generalization)



유스케이스 용어

• 관계 (Relationship) - 연관 관계 (Association)

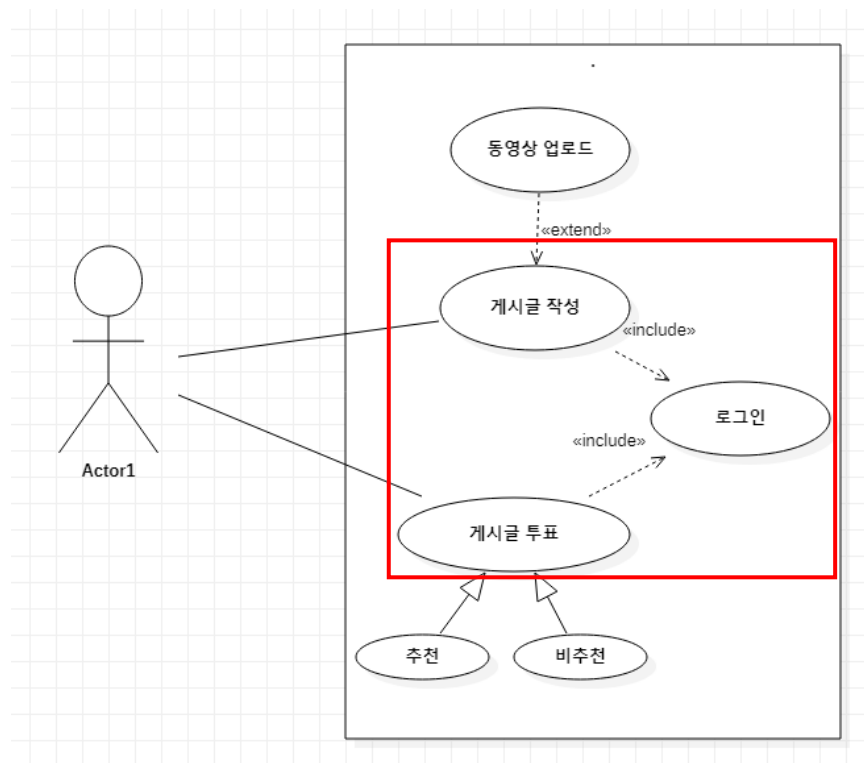
- 액터와 유스케이스 사이의 관계를 표현할 때 사용
 - 액터: 정보를 통보받거나 요구
 - 유스케이스: 정보를 제공
- 액터가 유스케이스를 사용하는 것을 표현



유스케이스 용어

• 관계 (Relationship) - 포함 관계 (Include)

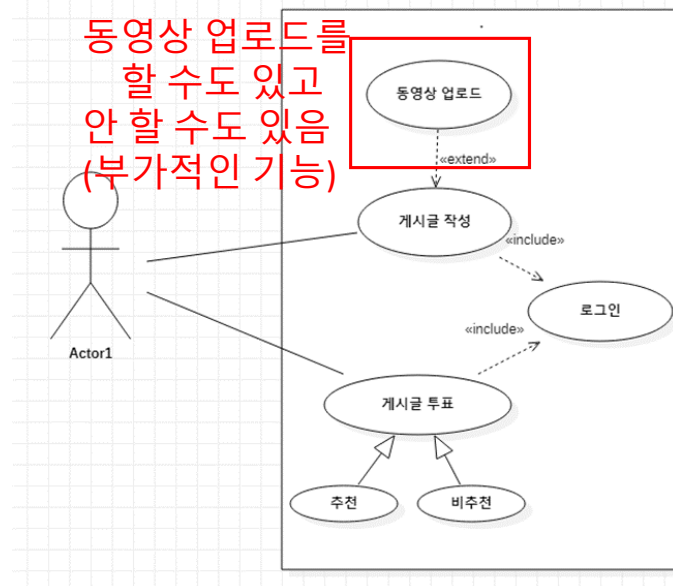
- 하나의 유스케이스를 수행할 때, 포함된 유스케이스가 반드시 수행되는 것
- 여러 유스케이스(예: 게시물 작성, 게시물 투표)에 나타난 공통적인 이벤트 흐름을 별도의 유스케이스로 표현 → 중복을 방지



유스케이스 용어

• 관계 (Relationship) - 확장 관계 (Extend)

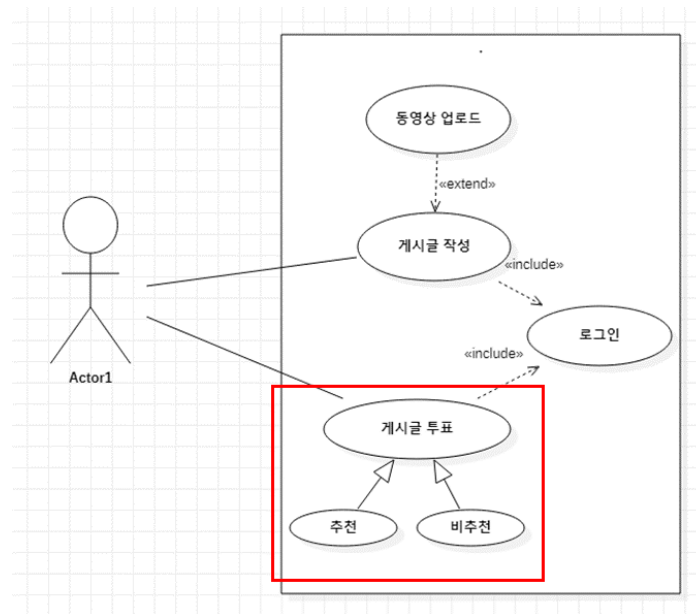
- 포함 관계 (Include)처럼 여러 유스케이스에 걸쳐 중복적으로 사용되지 않고 **특정 조건에서 한 유스케이스로만 확장되는 것**
 - 포함 관계: 여러 유스케이스에서 공통적으로 발견되는 시스템의 기능을 표현
 - 확장 관계: 한 유스케이스에서 추가되거나 확장된 기능을 표현
- 확장하는 유스케이스는 상위 유스케이스로부터 어떠한 **특정 조건에 의해 수행되는 것을 의미**



유스케이스 용어

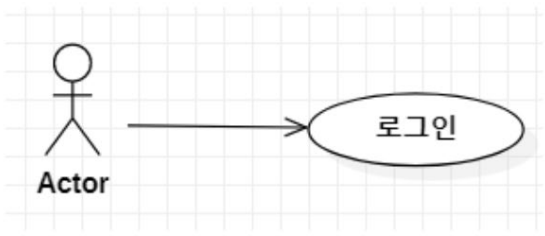
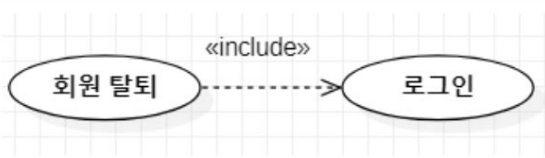
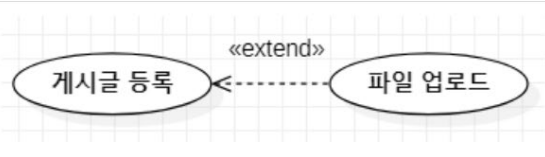
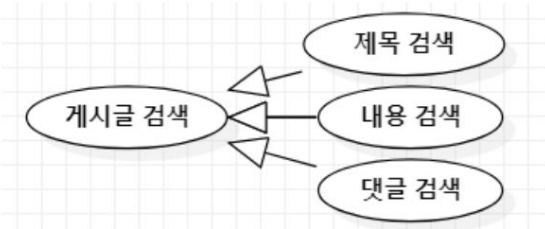
• 관계 (Relationship) - 일반화 관계 (Generalization)

- 액터와 액터간의 관계, 유스케이스와 유스케이스 사이의 정의 (상속)
- 특정 유스케이스들이 한 유스케이스의 특수화된 유스케이스라는 뜻
- 확장 관계와의 차이점
 - 확장 관계: **특정 조건이 충족될 때에만 확장되는 기능**
 - 일반화 관계: 하위 유스케이스는 상위 유스케이스에서 정의된 특징과 행동을 상속받아 공유 (하위 유스케이스는 항상 상위 유스케이스의 확장)



유스케이스 용어

• 관계 (Relationship)

유형	설명	관계 방향
연관관계	상호작용	 <p>A stick figure labeled 'Actor' is connected by a solid arrow to an oval labeled '로그인' (Login).</p>
포함관계	기능 포함&반드시 실행	 <p>An oval labeled '회원 탈퇴' (Logout) is connected by a dashed arrow labeled «include» to an oval labeled '로그인' (Login).</p>
확장관계	선택에 따른 관계	 <p>An oval labeled '게시글 등록' (Post Registration) is connected by a dashed arrow labeled «extend» to an oval labeled '파일 업로드' (File Upload).</p>
일반화관계	그룹핑	 <p>An oval labeled '게시글 검색' (Post Search) is connected by three solid arrows from three separate ovals: '제목 검색' (Title Search), '내용 검색' (Content Search), and '댓글 검색' (Reply Search).</p>

유스 케이스 다이어그램 작성 예시

- **유스케이스 다이어그램 작성법 & 작성 순서**

- **Step 1: 구성요소 정의**

- 유스케이스 다이어그램을 이루는 시스템, 액터, 유스케이스 정의

- **Step 2: 관계 정의**

- 액터와 액터 사이의 관계 정의 (일반화관계)
 - 액터와 유스케이스 사이의 관계 정의 (연관관계)
 - 유스케이스와 유스케이스 사이의 관계 정의 (포함관계, 확장관계)

- **Step 3: 유스케이스 구조화**

- 두 개 이상의 유스케이스 간 공통된 서비스를 추출해 일반화 (일반화 관계)

유스 케이스 다이어그램 작성 예시

• 유스케이스 다이어그램 작성 (예시)

• 액터를 찾기 위한 질문

- 누가 정보를 제공하고, 사용하고, 삭제하는가?
- 누가 또는 어떤 조직에서 개발될 시스템을 사용할 것인가?
- 누가 요구사항에 대해 관심을 가지고, 시스템이 만든 결과에 관심이 있는가?
- 개발될 시스템과 상호작용하는 하드웨어나 소프트웨어 시스템은 무엇인가?

• 유스케이스를 찾기 위한 질문

- 액터가 원하는 시스템 제공 기능은 무엇인가?
- 액터는 시스템에 어떤 정보를 생성, 수정, 조회, 삭제하고 싶어 하는가?
- 시스템이 어떤 기능을 제공하면 액터의 일상 작업이 효율적이고 편리해지는가?
- 모든 기능 요구사항들을 만족할 수 있도록 유스케이스가 모두 식별되었는가?

유스 케이스 다이어그램 작성 예시

• 유스케이스 다이어그램 작성 (예시)

액터



고객
(비회원)



회원



신용카드
인증회사

유스케이스

회원가입

상품주문

주문결제

상품검색

상품상세
조회

사용자
인증

장바구니
상품담기

장바구니
상품삭제

회원관리

회원조회

회원삭제

유스 케이스 다이어그램 작성 예시

• 유스케이스 다이어그램 작성 (예시)

액터



고객
(비회원)



회원



신용카드
인증회사

유스케이스

회원가입
(고객)

상품주문
(회원)

주문결제
(회원, 신용카드
인증회사)

상품검색
(고객)

상품상세
조회 (고객)

사용자
인증 (회원)

장바구니
상품담기
(회원)

장바구니
상품삭제
(회원)

회원관리
(회원)

회원조회
(회원)


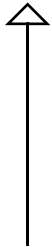
회원삭제
(회원)

유스 케이스 다이어그램 작성 예시

- 유스케이스 다이어그램 작성 (예시)



고객
(비회원)



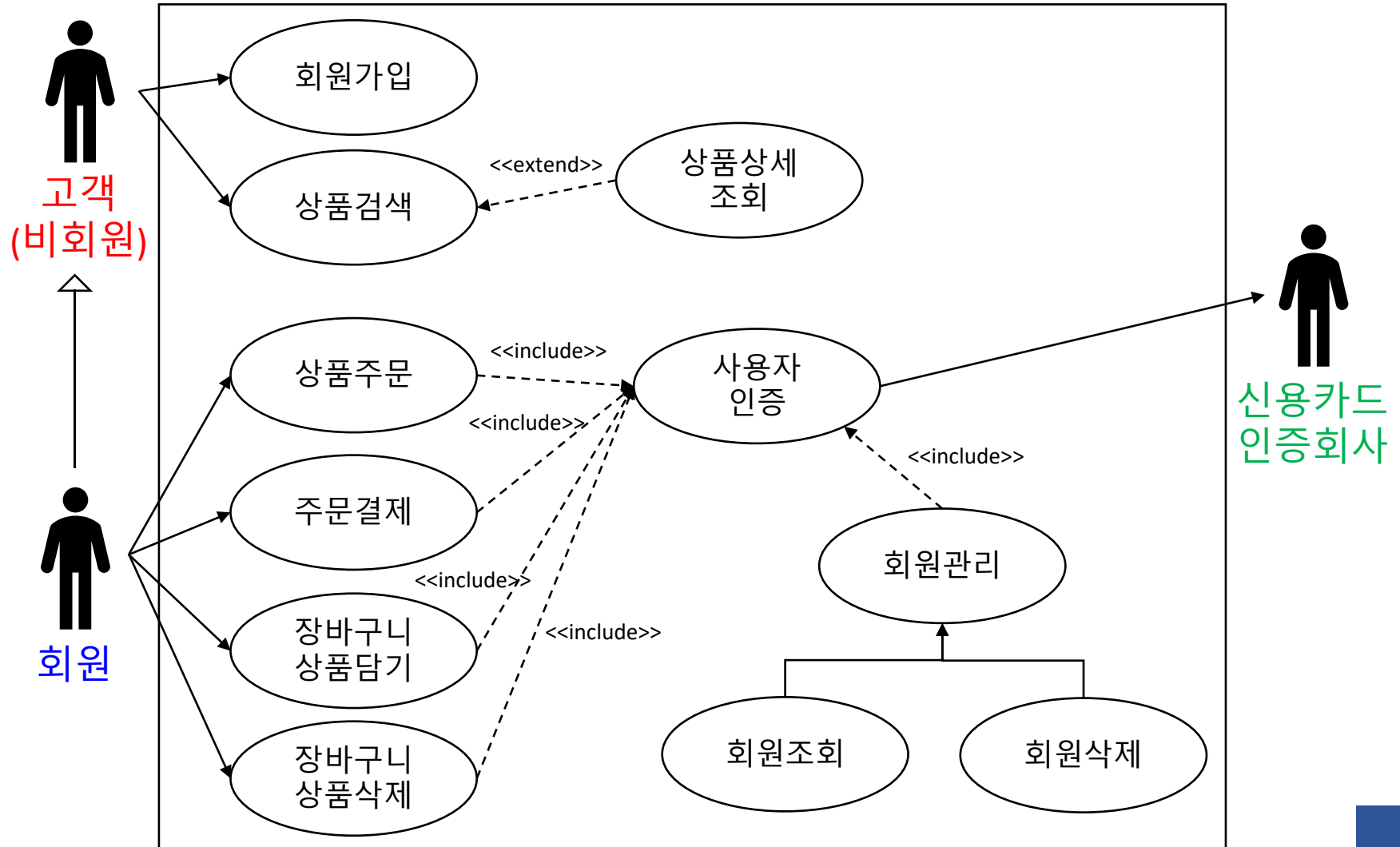
회원



신용카드
인증회사

유스 케이스 다이어그램 작성 예시

• 유스케이스 다이어그램 작성 (예시)



유스 케이스 다이어그램 작성 예시 (2nd)

- 요구사항 예시

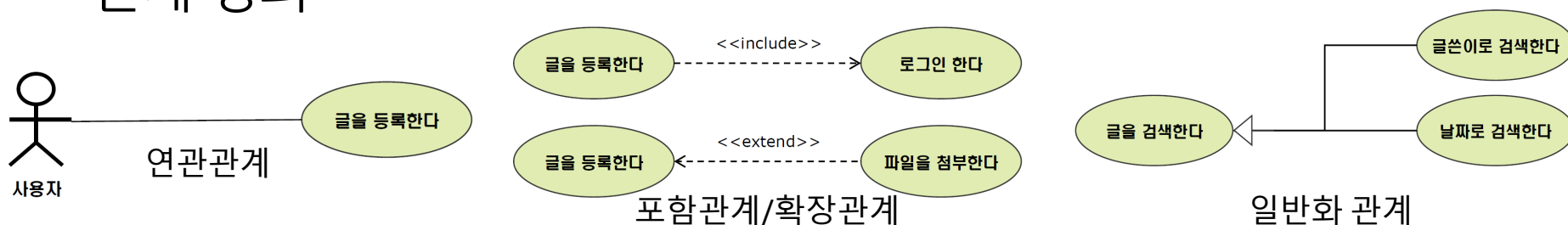
- **사용자 요구사항**

- : 글을 등록, 수정, 삭제할 수 있는 게시판을 개발한다.
(단, 관리자 모드는 개발하지 않는다)

- 글을 등록할 때에는 파일을 첨부할 수 있다.
 - 글을 조회하여 읽을 수 있다.
 - 등록된 글은 글쓴이 혹은 날짜 별로 검색할 수 있다.
 - 게시판의 모든 기능은 사용자 로그인 후에 사용할 수 있다.

유스 케이스 다이어그램 작성 예시 (2nd)

- 시스템: 게시판
- 액터: 사용자
- 유스케이스
 - 글을 등록, 글을 수정, 글을 삭제, 글을 조회, 글을 검색, 파일 첨부, 로그인
- 관계 정의



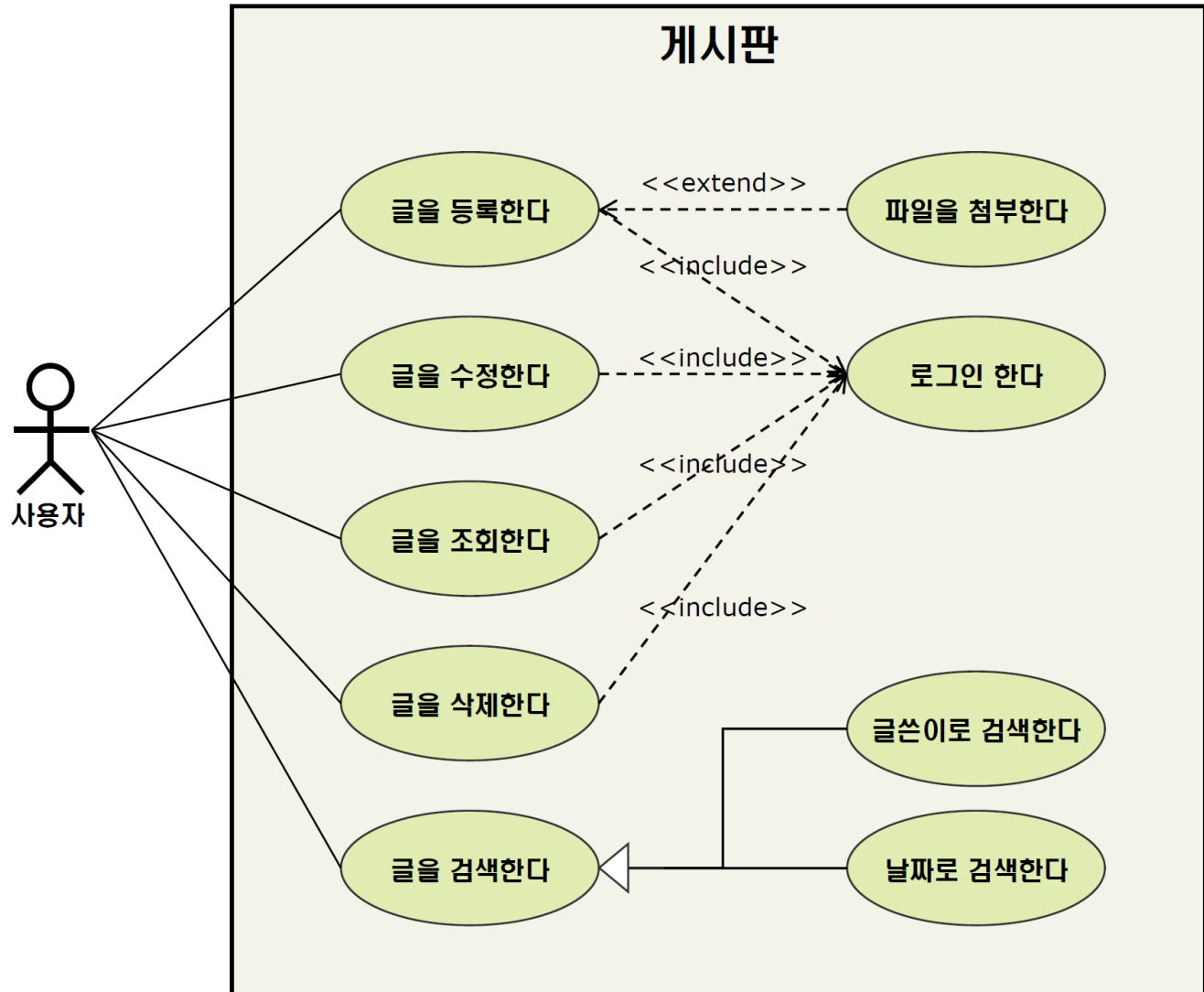
· 사용자 요구사항

: 글을 등록, 수정, 삭제할 수 있는 게시판을 개발한다.
(단, 관리자 모드는 개발하지 않는다)

- 글을 등록할 때에는 파일을 첨부할 수 있다.
- 글을 조회하여 읽을 수 있다.
- 등록된 글은 글쓴이 혹은 날짜 별로 검색할 수 있다.
- 게시판의 모든 기능은 사용자 로그인 후에 사용할 수 있다.

유스 케이스 다이어그램 작성 예시 (2nd)

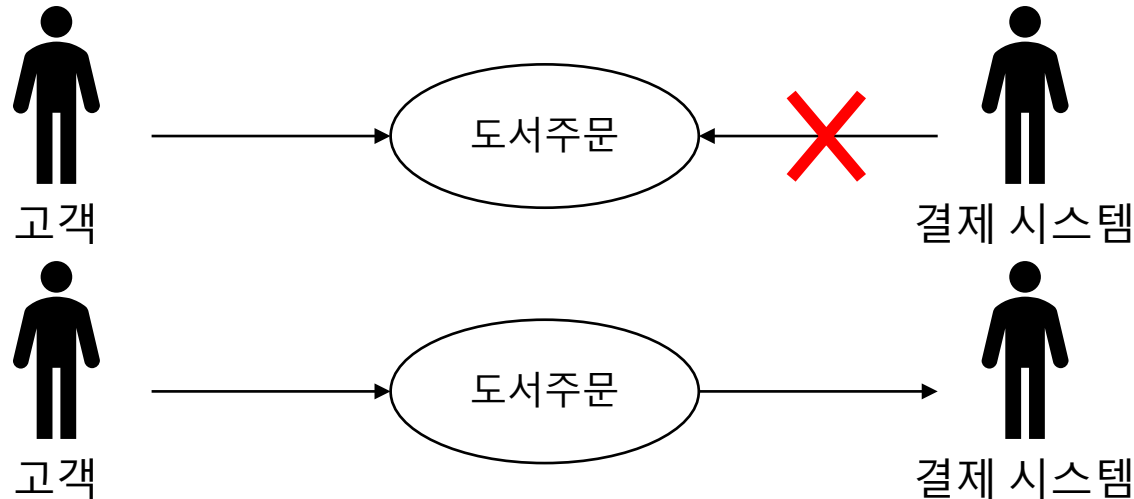
- 작성 결과



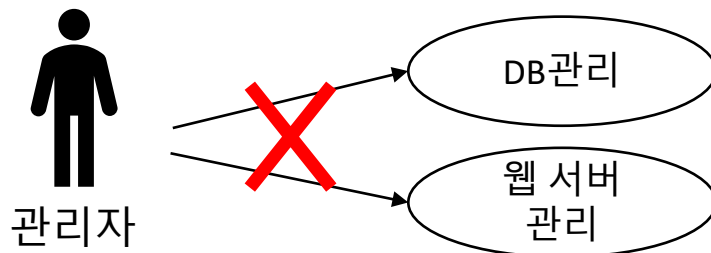
유스 케이스 다이어그램 점검사항

• 점검사항

- 유스케이스 간에는 연관관계(Association)를 정의할 수 없음
- 연관관계가 잘못된 것이 없는지 확인



- 잘못된 유스케이스가 추출된 것은 없는지 확인
(시스템을 이용하는 사람의 입장에서 추출 되어야 함)



DB관리/웹서버관리는 관리자에게 필요한 기능 설계하려고 하는 시스템 자체의 기능으로 존재할 필요가 없음
→ 해당 액터 및 유스케이스는 제거

(3) 요구사항의 문서화

- **요구분석명세서 (SRS, Software Requirements Specification)**
 - 분석한 요구사항을 모두 빠짐없이 작성한 문서
 - 주의 사항
 - 사용자가 쉽게 읽고 이해할 수 있도록 작성
 - 개발자가 설계와 코딩에 효과적으로 사용할 수 있도록 작성
 - 테스트 기준으로 사용할 수 있도록 **정량적으로 작성** (품질과 제약사항 등 비기능 요구사항을 정량적으로 명확히 작성)

(3) 요구사항의 문서화

• 요구분석명세서의 기본 항목

1. 소개

- 1.1 목적
- 1.2 범위
- 1.3 정의, 약어
- 1.4 참조
- 1.5 개요

2. 전반적 서술

2.1 제품 관점

- 시스템 인터페이스
- 사용자 인터페이스
- 하드웨어 인터페이스
- 소프트웨어 인터페이스
- 통신 인터페이스
- 메모리
- 운영

2.2 제품 기능

2.3 사용자 특성

2.4 제약 사항

- 규제 정책, 하드웨어 제약 사항, 다른 응용 프로그램과의 인터페이스, 병렬 수행, 감사 기능, 제어 기능, 신뢰성 요구 사항, 안전 및 보안 요구 사항

2.5 가정 및 의존성

2.6 요구 사항 할당

3. 구체적 요구 사항

3.1 외부 인터페이스

- 사용자 인터페이스, 하드웨어 인터페이스, 소프트웨어 인터페이스, 통신 인터페이스 기능

3.2 성능 요구 사항

3.3 로컬 DB 요구 사항

3.4 설계 제약 사항

3.5 소프트웨어 시스템 속성

- 신뢰성, 가용성, 보안성, 유지보수 용이성, 이식성

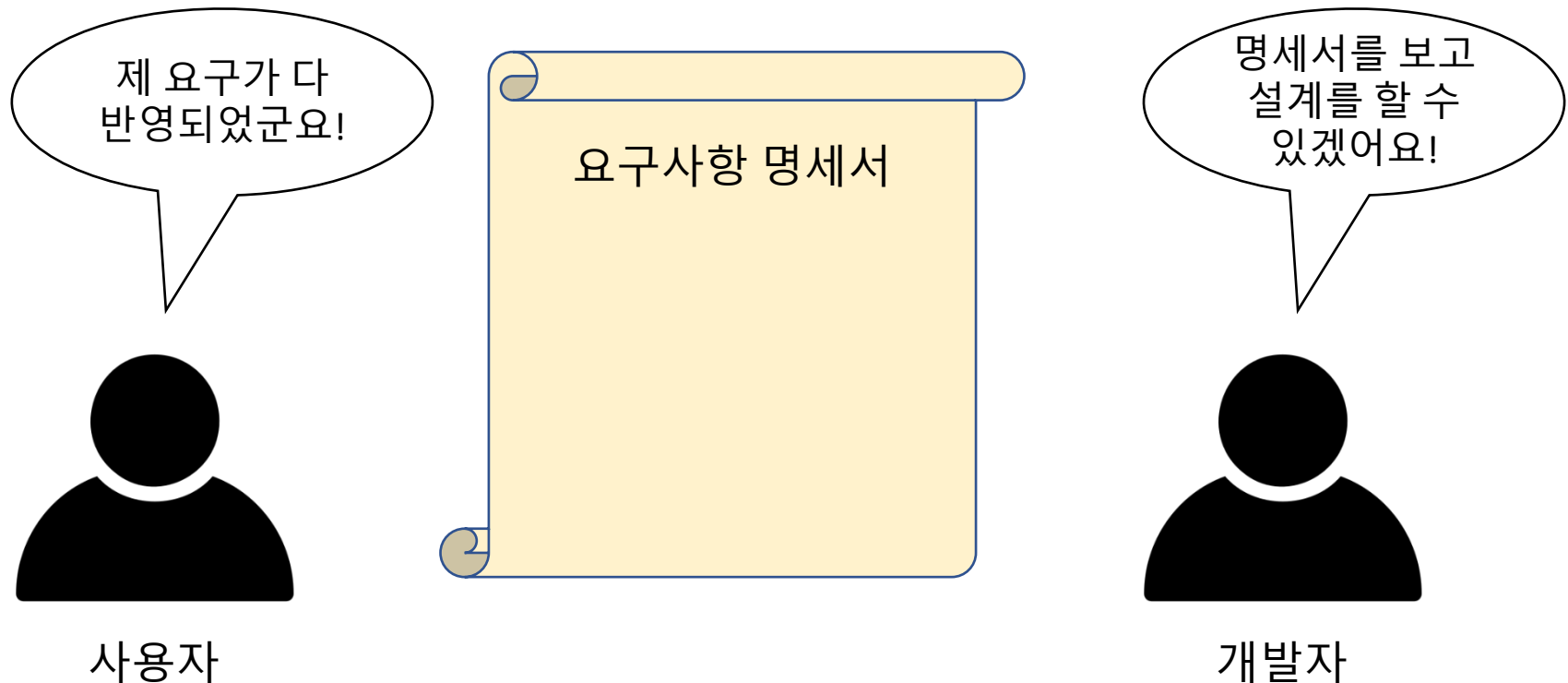
인덱스

부록

(4) 요구사항 검증

- 요구사항 검증

- 요구분석 명세서가 정확하고 안전하게 서술되었는지 검토하는 활동
- 사용자의 요구사항이 완전하게 서술되었는지 검증
- 요구분석 명세서가 설계 단계에 사용하기에 적합한지 확인



(4) 요구사항 검증

- 요구사항 검증

- **완전성**

- 사용자의 모든 요구 사항이 누락되지 않고 완전하게 반영되고 있는가?

- **일관성**

- 요구 사항이 서로 간에 모순되거나 충돌되는 점은 없는가, 산출물 또는 요구 사항의 내용이 일관성을 유지하고 있는가?

- **명확성**

- 서술된 명세서의 내용이 애매모호하지 않고 모든 참여자가 명확히 이해할 수 있는가?

- **기능성**

- 서술된 명세서가 “어떻게(How)”보다 “무엇을”에 관점을 두고 서술했는가?

(4) 요구사항 검증

- 요구사항 검증

- 검증 가능성

- 서술된 명세서의 내용이 사용자의 요구를 만족하는가?
 - 개발된 소프트웨어가 사용자가 요구하는 내용과 일치하는지를 검증할 수 있는가?

- 추적 가능성

- 사용자 요구 분석 명세서와 설계 사양서를 추적할 수 있는가?

- 변경 용이성

- 요구 분석 명세서의 내용을 변경하고자 할 때 쉽게 찾아 변경할 수 있도록 작성되었는가?

Questions?