

# 소프트웨어공학

유스케이스 시나리오 작성 PART 2

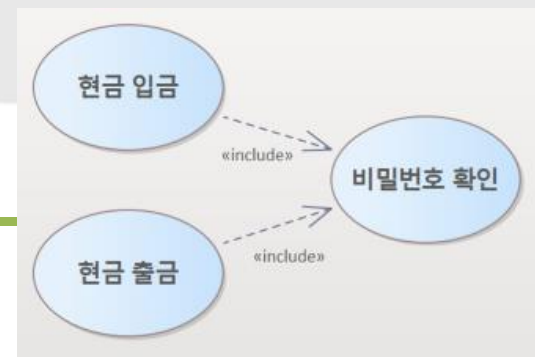


# 유스케이스 시나리오 작성 필요성

- 개발된 소프트웨어 시스템이 잘 사용되지 않는다면,

그 이유는 사용자가 필요로 하는 사용자의 요구사항을 적절하게 고려하지 않았기 때문입니다.

- 그럼 사용자의 요구사항을 적절하게 고려해서 좋은 소프트웨어를 만들 수 있는 방법은? 사용자가 개발될 시스템에 대해 잘 이해, 사용자와 많은 의사소통
- 근데 앞서 작성한 유스케이스 다이어그램은 각 기능의 이름과 관계 외에 자세한 설명이나 정보는 포함되어 있지 않습니다.
- 즉, 유스케이스 다이어그램이 '전체적인 기능 요구사항'은 시각적으로 잘 표현하고 있지만 구체적인 동작은 알 수가 없습니다. 따라서 유스케이스 다이어그램이 완성되면 여러 이해관계자들이 이해하기 쉽도록 각각의 유스케이스에 대한 자세한 시나리오를 유스케이스 별로 각각 작성해 줘야 합니다.



## 유스케이스 시나리오 작성 필요성

- 이 유스케이스 시나리오는 ‘자연어로 세부적인 요구사항의 각 단계를 표현해주는 방법’으로 사람들은 추상적인 기술보다는 실제의 예를 들어서 설명할 때 더 쉽게 이해할 수 있습니다.
- 예를 들어 이런 시나리오 형태로 ‘자신이 실제 만들어질 시스템과 어떻게 대화할 수 있는지를 알려주면 더 잘 이해할 수 있다’는 건데요, 그런 점에 유스케이스 시나리오는 아주 좋은 방법 중의 하나

- 또한 이 시나리오는 상세한 사용 예를 이용해 개괄적인 요구사항에 좀 더 상세한 사항을 추가할 때도 유용합니다.

1. 학생이 수강신청하기 메뉴를 요청함으로써 이 Use Case는 시작한다.
2. 학생은 “조회하기” 버튼을 클릭하여 수강 정보를 요청한다. (E5)
3. 금번학기에 개설된 강의 정보(기신청학점수, 학기, 강좌번호, 강좌 이름, 담당학과, 학점, 강좌 설명, 신청가능여부, 수강신청여부)를 출력한다. (A1, E1)
4. 학생이 수강신청 하고자 하는 수강과목을 선택한다. (E2,E3,E4)
5. 선택된 강의정보를 “수강신청하기” 버튼을 클릭하여 신청한다. (E3,E5)
6. “\*건[성공] \*건[실패] 로 수강과목 신청이 처리되었습니다.”라는 메시지를 출력한다.
7. 신청된 수강과목 목록을 출력한다.

# 유스케이스 시나리오 작성 (유스케이스 별 내부 모델링)

- 유스케이스 시나리오 예시

- 모든 이해관계자들이 서로 소통할 수 있게 해주는 도구로  
일상의 언어 즉 자연어로  
세부적인 요구사항의 각 단계를 표현
- 작성할 때는 실제 사용자가 사용하는 것과 같은 상황을 단계적으로 기술  
즉, 해당 기능 요구 사항의 절차(과정)을 기술
- 예를 들어서 '실제 수강신청을 하려고 하는 학생 입장에서  
실제 수강신청하는 과정을 있는 그대로, 단계별로 기술'

- 개요

- 학생이 금번학기에 개설된 강좌 중 수강하고자 하는 과목을 신청한다.

- 흐름

- 기본 흐름 (Basic Flow)

1. 학생이 수강신청하기 메뉴를 요청함으로써 이 Use Case는 시작한다.
2. 학생은 “조회하기” 버튼을 클릭하여 수강 정보를 요청한다.(E5)
3. 금번학기에 개설된 강의 정보(기신청학점수, 학기, 강좌번호, 강좌 이름, 담당학과, 학점, 강좌 설명, 신청가능여부, 수강신청여부)를 출력한다. (A1, E1)
4. 학생이 수강신청 하고자 하는 수강과목을 선택한다. (E2,E3,E4)
5. 선택된 강의정보를 “수강신청하기” 버튼을 클릭하여 신청한다. (E3,E5)
6. “\*건[성공] \*건[실패] 로 수강과목 신청이 처리되었습니다.”라는 메시지를 출력한다.
7. 신청된 수강과목 목록을 출력한다.

- 대안 흐름(Alternative Flow)

- A1. 수강 신청 기간이 아닌 경우, “수강신청 기간이 아닙니다“ 라는 메시지를 출력한다.

# 유스케이스 시나리오 작성 (유스케이스 별 내부 모델링)

- 유스케이스 시나리오(기술서, 명세서, 설명서) 작성

- ☞ 유스케이스 별로 각각 작성, 유스케이스 다이어그램을 보완, 해당 기능의 상세 흐름 나타냄

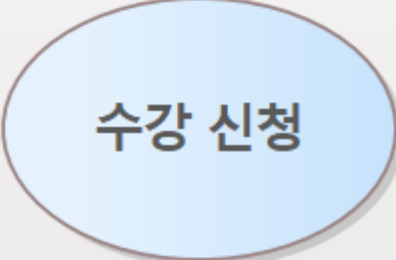
- 액터와 유스케이스 간의 상호작용을 기반으로 (사용자 입장에서) 순차적 시나리오 형식으로 기술한다.
  - 시스템이 해당 기능을 수행하기 위해 어떤 절차를 거치는지, 그 과정에서 주고 받는 데이터 등 파악 가능
  - 시스템 개발 전체 단계에서 다양한 이해관계자들과 관련되므로 이해성 및 가독성을 높일 수 있도록 작성
  - 요구사항 분석 단계에서 작성하기 때문에 (사용자 관점에서 기술하며) 액터와 시스템이 상호작용하는 과정을 구체적으로 묘사한다.
  - 시나리오 문장은 ‘액터’ 또는 ‘시스템’으로 시작해서 명사 뒤에 행위를 나타내는 동사를 기술하여 ‘주어- 목적어- 동사’의 일관된 문장으로 작성하는 것이 좋다.

- . ‘액터는 ~을 한다’ 형식 (예: 고객은 조회하고자 하는 카테고리 선택한다. (입력))

- . ‘시스템은 ~을 한다’ 형식 (예: 시스템은 선택된 카테고리의 하위 카테고리를 출력한다. (출력))

# 유스케이스 시나리오 작성

- 유스케이스 기본 정보 (정형화된 방법은 없으나 보편적으로 다음과 같은 내용을 포함)
  - 유스케이스 이름
  - 유스케이스를 수행하는 행위자(액터, 사용자)
  - 유스케이스 개요
  - 상태(Status) : 유스케이스 작성 진행 단계를 기술 (예: 작성중)
  - 사전 조건(Preconditions) : 유스케이스의 기본 흐름이 올바르게 동작되기 위하여 유스케이스를 시작하기 전에 만족해야하는 조건을 제시  
(예: 학생은 재학중이고 로그인 되어있어야 한다)
  - 사후 조건(Postconditions) : 유스케이스 종료 이후 정상적인 시스템 상태를 기술  
(예: 수강 신청된 강좌가 스케줄로 저장된다.)
  - 이벤트 흐름(기본 흐름, 대안 흐름, 예외 흐름) ➡ 핵심적인 부분(다음 페이지)



수강 신청



# 유스케이스 시나리오 작성

수강 신청

예금 인출

상품 결제

- 유스케이스 시나리오 흐름

**1. 기본 흐름(Basic Flows)**: 기본 흐름을 구성하는 각 단계의 수행 주체가 액터인지 시스템인지가 명확하게 구분, 유스케이스를 시작해서 종료할 때까지 정상적으로 발생하는 상호작용의 흐름을 계층화, 구조화하고 번호를 부여하여 단계별로(순차적으로) 기술, 조건문 반복문 등 포함 가능

**2. 대안 흐름(Alternative Flows)**: 유스케이스의 정상적인 흐름에 대한 대안을 제시하는 것으로 각 단계에서 조건이 false일 때 기본 흐름으로 분기 (필수는 아님), 기본 흐름과 연계되어 발생 특정 시점에 여러가지 선택적인 흐름으로 나뉘어질 경우 발생하는 흐름

☞ 번호는 기본 흐름 번호와 함께 1a, A1 등의 형식으로 작성

**3. 예외 흐름(Exceptional Flows)**: 유스케이스 각 단계에서 비정상적인 상황(오류나 예외적인 사건)이 발생할 때 기술 (필수는 아님), 기본 흐름과 연계되어 발생

☞ 번호는 기본 흐름 번호와 함께 1b, E1 등의 형식으로 작성

※ 기본흐름과 대체 흐름의 2가지 단계로 나누기도 한다.

# 유스케이스 흐름 작성 예시

## 유스케이스 흐름 작성 예시

### 1. 기본 흐름

1.1 사용자는 ~을 한다. (입력) -> 구체적으로 작성

1.2 시스템은 ~을 한다. (출력) -> 구체적으로 작성

1.3 사용자는 ~을 한다. (A1)

1.4 시스템은 ~을 한다. (출력)

1.5 사용자는 ~을 한다. (입력)

1.6 시스템은 ~을 한다. (E1)

:

### 2. 대안 흐름

A1. ~일 경우에는, 시스템이 ~을 한다.

### 3. 예외 흐름

E1. ~일 경우에는, 시스템이 ~을 한다.



## 예제) '현금 출금' 유스케이스 시나리오 (여기부터 PART 3에서 설명)

- (ATM 기기에서 돈을 찾는) '현금 출금' 유스케이스 시나리오 (쉽게 작성하고, 쉽게 이해할 수 있는 기술서)

- **유스케이스명** : 현금 출금
- **액터명** : 사용자
- **개요**
  - 고객이 ATM으로부터 자신의 계좌에 입금되어 있는 현금을 출금한다.
- **상태(Status)** : 작성완료
- **사전 조건(Preconditions)**
  - 사용자가 은행 카드를 소지하고 있어야 한다.
- **사후 조건(Postconditions)**
  - 사용자가 카드를 돌려받는다.
  - 은행 계좌에 출금액이 반영된다. (잔액 조정)

## 예제) (고객이 ATM으로부터 자신의 계좌에 입금되어 있는 현금을 출금한다.)

- 내가 ATM 기기 앞에서 서서 실제 돈을 찾는 것과 같은 상황을 기술, 해당 기능 요구 사항의 절차(과정)를 기술

### · 흐름

#### - 기본 흐름(Basic Flows)

1. 사용자는 ATM 기계에 카드를 넣는다.
2. ATM은 카드 정보를 읽는다.
3. ATM은 사용자에게 초기 화면을 출력한다. (3a)
4. 사용자는 ATM 기계의 인출 버튼을 클릭한다.
5. ATM은 인출 화면을 출력한다.
6. 사용자는 ATM에 인출금액을 입력하고 확인 버튼을 클릭한다.
7. ATM은 사용자에게 돈을 내어준다. (7a) (7b)

#### - 대안 흐름(Alternative Flows)

- 3a. ATM이 읽은 카드가 문제가 있는 경우, ATM은 사용자에게 '사용할 수 없는 카드'라는 사실을 알리고 유스케이스를 종료한다.
- 7a. 통장 잔액이 부족한 경우, 사용자에게 잔액 부족 사실을 알리고 5를 다시 수행한다.

#### - 예외 흐름(Exceptional Flows)

- 7b. ATM 기기에 돈이 부족한 경우, 사용자에게 시스템 에러 화면을 띄우고 유스케이스를 종료한다.



 **T h a n k      y o u**

## **TECHNOLOGY**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Velit ex  
plicabo ipsum, labore sed tempora ratione asperiores des  
cenderat bore sed tempora rati jgert one bore sed tem!