

# Software Requirements Specification

for

<Project>

Prepared by <authors>

<Team #>

<date created>

## Revision History

Name	Date	Reason For Changes	Version

---

**전체가이드:** 시스템의 기능 (*functionality*)을 위주로 작성한다. 비기능적 (*non-functional*) 요소의 경우 (예: 10초내 응답 (성능), 사용자 친화적 *UI*), 해당 문서에서는 기능적 요소와 따로 구분하지는 않으나, 기능적 요소와 비교하여 낮은 우선순위를 갖는다)

---

## 1. Introduction

### 1.1 Purpose & System Scope

<본 시스템이 해결하려는 문제와 제공하는 가치를 서술 >

### 1.2. System Scope

<본 시스템이 제공할 주요 기능의 범위와 한계를 기술. System Request 문서를 참조>

주의: 주 질문은 “이 시스템이 어떤 기능들을 포함하는가, 무엇을 포함하지 않는가”, 즉 시스템의 경계 설정. 이와 더불어 이런 시스템의 범위에 따른 기대되는 사용 효과까지 큰 틀에서 기술 (시스템의 큰 그림과 포함 범위)

### 1.3. System Overveiw (선택)

<전체 시스템 구조/범위/상호작용을 요약한 Figure>

## 2. Overall Description

### 2.1 Product Perspective (제품 전망)

<본 시스템이 사용될 환경(맥락), 기존 시스템 또는 유사/관련 시스템과의 관계(예: 연동/대체/확장), 그리고 새로운 시스템이 필요한 이유(예: 기존 한계나 문제점) 서술>

### 2.2 Product Features (제품 기능)

<본 시스템이 제공해야 하는 주요 상위 수준 기능 카테고리 (Sytem Feature)를 한눈에 볼 수 있도록 요약. 세부 설명 및 Functional requirement들은 Section 3에서 등장>

### 2.3 User Characteristics (사용자 특성)

< 본 시스템의 주요 사용자 클래스(그룹)를 구분하여 요약. 각 사용자 그룹이 가지는 주요 역할/목표와 시스템 이용 특성을 간단히 설명. (자세한 상호작용은 Section 3의 System Features와 Use Cases에서 다룸). >

### 3. System Features (시스템 기능)

< 본 Section에서는 2.2 Product Features에서 요약한 각 System Feature를 상세히 기술한다. 각 Feature는 다음 요소들을 통해 기술된다:

- 기능 요구사항 (Functional Requirements)
  - (필요시) 일부 비기능 요구사항 (Non-functional)가 포함. FR과 구분 X (전체 가이드 참조)
- 유스케이스 다이어그램 (Use Case Diagram)
- 유스케이스 설명 (Use Case Description)

작성 지침: [Activity Diagram](#)

- 최소 3개의 System feature를 기술. 각 Feature는 독립된 Subsection (e.g., 3.1. “System\_faure1”)으로 구성
- Use-case diagram의 개수는 고정되어 있지 않으며, 팀 구성원과의 논의를 통해 적절한 수의 use-case diagram을 작성.
  - activity diagram의 복잡도와 use-case diagram의 개수 관계 고려하여 조절.
  - 예시: 전체 system feature에 대하여 high-level use case diagram 1개. 세부 카테고리 분리될 시, 각 카테고리에 대하여 low-level use-case diagram 작성.
- Activity diagram은 모든 use-case가 아니라 주요 use-case에 대해서만 작성.
- Use-case description의 경우, activity diagram과 유사하게 주요 use-case에 대해서만 작성. 이때 관련 use-case diagram과 activity diagram과 내용 일치에 주의
- 모든 다이어그램과 Use Case Description은 해당 Feature의 Functional Requirements와 논리적으로 연결.

가장 main이 되는  
Use Case와 연관된  
Activity Diagram만

#### 3.1 System Feature 1 (예) 건강 모니터링, 주치의 관리 기능, etc.)

##### 3.1.1 Functional Requirements

Appendix A 작성.

##### 3.1.2 Use case diagram(s) and Activity diagram(s)

- 시스템 경계(System Boundary)가 명확히 드러나도록 주요 유스케이스 다이어그램을 작성.
- 해당 유스케이스에 대응되는 액티비티 다이어그램을 작성 (지침 참조).
- 부록 B(Appendix B)를 활용하여 유스케이스 설명(Use Case Description) 작성.

**3.2    System Feature 2**

**3.3    System Feature 3 (and so on)**

...

< For writing>

## Appendix A: Functional requirements ->

Hierarchy*	Requirement ID	Requirement Description	Actors (can be more than one)	Priority (0 - 5)
XXX (A) (Manage appointments)	A_1	... (Create a new appointment)	... (patient)	... (5)
XXX (B)	B_1	...		

\* Hierarchy: major (high-level) use cases that compose a system feature.

\* A, B, etc: ID

\* Priority: 0 = least important. 5 = most important

## Appendix B: Use case description – 수업 슬라이드 참조

Use case description template (should be provided with the use case diagram)

Use case name	name of the use case to be described
Related requirements	Requirement ID
Primary actors	Main actor
Secondary actors (= stakeholders)	Do not need to include primary actors
Brief description	Briefly describe the use case
Pre-conditions	What needs to happen before the use case can be executed  e.g., appointment system: the patient information should be available before managing an appointment
Successful End Condition	What the system's condition should be if the use case executes successfully
Failed End Condition	What the system's condition should be if the use case fails to execute successfully

Trigger (external, internal, temporal)	What initiated this use case
Main flow (= normal flow)	Main steps
Subflows	Sub-steps (detail the main flows)
Alternate / Exceptional flows	alternate flow when exceptions happen