



Introduction to Software Engineering (SWE3002_42)

Smart Noise

SDS (Software Design Specification)

Team 7

2019311918 고남욱

2018310072 고은서

2017314545 손석규

2016314869 이승민

2018310478 허한율

목차

1. 머리말	7
1.1 독자층	7
1.2 문서 Scope	7
1.3 문서 목적	7
1.4 문서 구성	7
2. 서론	8
2.1 이 장에서는..	9
2.2 적용 다이어그램	9
2.2.1 UML	9
2.2.2 Use Case 다이어그램	9
2.2.3 Class 다이어그램	10
2.2.4 Sequence 다이어그램	10
2.2.5 Context 다이어그램	11
2.2.6 Entity Relationship 다이어그램	12
2.3 적용 툴	13
2.3.1 Lucid Chart	13
2.3.2 Miro	13
2.3.3 Fusion 360	13
2.4 프로젝트 범위	13
3. 시스템 컨텍스트 및 상호 작용	13
3.1 이 장에서는..	13
3.2 Context model	14
3.3 Use case	15
3.3.1 Use case diagram	15
3.3.2 Use case 예시	16
4. 전체 시스템 아키텍처	19
4.1 이 장에서는..	19
4.2 시스템 구성	20
5. 설계 모델	21
5.1 이 장에서는..	21
5.2 Overall Design model	22
5.2.1 Sequence 다이어그램	22
5.2.2 Data flow diagram	23
5.3 사용자 정보 등록 시스템	24
5.3.1 객체 클래스 식별	24
5.3.2 설계 모델	25
5.3.3 인터페이스 명세	27
5.4 사용자 데이터 관리 시스템	28
5.4.1 객체 클래스 식별	28
5.4.2 설계 모델	29
5.4.3 인터페이스 명세	30

5.5 응급 상황 미션 관리 시스템	31
5.5.1 객체 클래스 식별	31
5.5.2 설계 모델	32
5.4.3 인터페이스 명세	34
5.6 디바이스 연결 관리 시스템	35
5.6.1 객체 클래스 식별	35
5.6.2 설계 모델	36
5.6.3 인터페이스 명세	38
5.7 위험 소리 탐지 시스템	38
5.7.1 객체 클래스 식별	39
5.7.2 설계 모델	39
5.7.3 인터페이스 명세	41
5.8 위험 상황 알림 시스템	41
5.8.1 객체 클래스 식별	41
5.8.2 설계 모델	42
5.8.3 인터페이스 명세	43
6. 인터페이스 디자인	44
6.1 이 장에서는..	44
6.2 어플리케이션 인터페이스	44
6.3 스마트 디바이스 인터페이스	45
7. 프로토콜 디자인	46
7.1 이장에서는..	46
7.2 HyperText Transfer Protocol over TLS (HTTPS)	46
7.3 Bluetooth Protocol	47
7.4 인증	49
7.4.1 Register	49
7.4.2 로그인	50
7.5 사용자 프로필	50
7.5.1 사용자 프로필 설정	50
7.5.2 사용자 프로필 가져오기	52
7.6 데이터 가져오기	53
7.6.1 데이터 가져오기	53
7.6.2 데이터 분석하기	54
7.7 응급 상황 미션	55
7.7.1 응급 상황 미션 설정	55
7.7.2 응급 상황 미션 성공	55
7.7.3 응급 상황 미션 실패	56
7.7.4 119 호출 취소	56
7.8 디바이스 연결	57
7.8.1 디바이스 검색	57
7.8.2 디바이스 연결	58
7.8.3 디바이스 연결 끊기	59
8. 데이터베이스 디자인	60

8.1 이 장에서는..	60
8.2 ER Diagram	60
8.2.1 요소들	61
8.3 관계 계획	63
8.4 SQL DDL	64
8.4.1 User	64
8.4.2 Device	64
8.4.3 Breath_data	65
8.4.4 Emergency_contact	65
9. 테스트 계획	65
9.1 이 장에서는..	65
9.2 테스트 정책	65
9.2.1 개발 테스팅	65
9.2.2 배포 테스팅	66
9.2.3 사용자 테스팅	67
9.2.4 테스팅 케이스	67
10. 개발 계획	67
10.1 이 장에서는..	67
10.2 프론트엔드 환경	67
10.2.1 Adobe Photoshop (UI/UX 디자인)	67
10.2.2 Android Studio (Application)	68
10.3 백엔드 환경	68
10.3.1 Github (오픈 소스)	68
10.3.2 MySQL Database (DBMS)	69
10.3.3 Node.js (Server)	70
10.3.4 AWS EC2 (Server)	70
10.4 제약	71
10.5 가정 및 의존성	71
11. 추가 정보	71
11.1 시장 양산 방안	71
11.2 Traction / 매출 전략 및 비전	71
11.3 자금소요 및 성과창출 전략	72
11.4 문서 히스토리	73
12. 참조	74

그림 목록

[그림 1] Context model	14
[그림 2] Use case diagram	16
[그림 3] 전체 시스템 구성	21
[그림 4] Sequence 다이어그램	23
[그림 5] Data flow diagram	24
[그림 6] 사용자 정보 등록 시스템 - 설계 모델	27
[그림 7] 사용자 정보 등록 시스템 - 인터페이스 명세 29	
[그림 8] 사용자 데이터 관리 시스템 - 설계 모델	31
[그림 9] 사용자 데이터 관리 시스템 - 인터페이스 명세	32
[그림 10] 응급 상황 미션 관리 시스템 - 설계 모델	35
[그림 11] 응급 상황 미션 관리 시스템 - 인터페이스 명세	37
[그림 12] 디바이스 연결 관리 시스템 - 설계 모델	39
[그림 13] 디바이스 연결 관리 시스템 - 인터페이스 명세	41
[그림 14] 위험 소리 탐지 시스템 - 설계 모델	43
[그림 15] 위험 소리 탐지 시스템 - 인터페이스 명세	44
[그림 16] 위험 상황 알림 시스템 - 설계 모델	45
[그림 17] 위험 상황 알림 시스템 - 인터페이스 명세	46
[그림 18] 디바이스 프로토타입	48
[그림 19] 결합 시 디바이스의 모양	49
[그림 20] Https 프로토콜	50
[그림 21] 블루투스 스택 구조	51
[그림 22] 블루투스 프로토콜	52
[그림 23] ER Diagram	64
[그림 24] ER Diagram, 개체, User	65
[그림 25] ER Diagram, 개체, Device	66
[그림 26] ER Diagram, 개체, Breath_data	66
[그림 27] ER Diagram, 개체, Emergency_contact	67
[그림 28] 관계 계획	67
[그림 29] Adobe Photoshop 로고	71
[그림 30] Android Studio 로고 72	
[그림 31] GitHub 로고	72

[그림 32] MySQL 로고	73
[그림 33] Node.js 로고	74
[그림 34] AWS EC2 로고	74
[그림 35] 매출 전략 장표	
72	
[그림 36] 자금 소요 및 성과창출 전략 장표	
72	

표 목록

[표 1] Smart Noise 등록하는 사용자 Use case	17
[표 2] 자신의 호흡 정보를 확인하고 싶은 사용자 Use case	17
[표 3] Smart Noise Device를 등록하고 싶은 사용자 Use case	18
[표 4] 위험에 처한 사용자 및 보호자 Use case	19
[표 5] 위험에 처한 사용자 및 병원 Use case	19
[표 6] 딥러닝 모델 개발자 Use case	20
[표 7] 사용자 정보 등록 시스템 - 설계 모델	28
[표 8] 사용자 데이터 관리 시스템 - 설계 모델	31
[표 9] 응급 상황 미션 관리 시스템 - 설계 모델	35
[표 10] 디바이스 연결 관리 시스템 - 설계 모델	39
[표 11] 위험 소리 탐지 시스템 - 설계 모델	43
[표 12] 위험 상황 알림 시스템 - 설계 모델	46
[표 13] 어플리케이션 인터페이스	47
[표 14] Register request	53
[표 15] Register response	53
[표 16] 로그인 Request	54
[표 17] 로그인 Response	54
[표 18] 사용자 프로필 설정 Request - 1	54
[표 19] 사용자 프로필 설정 Request - 2	55
[표 20] 사용자 프로필 설정 Response	55
[표 21] 사용자 프로필 가져오기 Request	56
[표 22] 사용자 프로필 가져오기 Response	56
[표 23] 데이터 가져오기 Request	57
[표 24] 데이터 가져오기 Response	57
[표 25] 데이터 분석하기 Request	58
[표 26] 데이터 분석하기 Response	58

[표 27] 응급 상황 미션 설정 Request	59
[표 28] 응급 상황 미션 설정 Response	59
[표 29] 응급 상황 미션 성공 Request	59
[표 30] 응급 상황 미션 성공 Response	60
[표 31] 응급 상황 미션 실패 Request	60
[표 32] 응급 상황 미션 실패 Response	60
[표 33] 119 호출 취소 Request	61
[표 34] 119 호출 취소 Response	61
[표 35] 디바이스 검색 Request	61
[표 36] 디바이스 검색 Response	62
[표 37] 디바이스 연결 Request	62
[표 38] 디바이스 연결 Response	62
[표 39] 디바이스 연결 Request	63
[표 40] 디바이스 연결 Response	63
[표 41] 문서 히스토리	75

1. 머리말

이 장에서는 본 문서의 독자층과 문서의 범주 및 목적, 구조 등을 설명한다.

1.1 독자층

본 문서의 주요 독자층으로는 **Smart Noise**를 개발하는 개발자와 이 시스템의 소유자가 있으며, 이 시스템을 사용하고자 하는 **end-users**, 클라이언트 엔지니어, 소프트웨어 테스터와 같은 사람들이 본 문서의 독자가 될 수 있다.

1.2 문서 Scope

본 디자인 명세서는 가내의 위험 소리를 탐지하는 **Smart Noise**의 시스템 및 소프트웨어의 디자인 설계(System and software design)를 위해 쓰일 예정이다.

1.3 문서 목적

본 디자인 명세서는 **Smart Noise** 시스템의 전반적인 아키텍처와 설계 프로세스에 대한 내용을 담고 있다. 소프트웨어 설계와 구현은 실행 가능한 소프트웨어 시스템이 개발되는 소프트웨어 공학 프로세스 단계이며, 소프트웨어 설계는 고객의 요구사항을 기반으로 하여 소프트웨어 컴포넌트들과 그들 간의 관계를 식별하는 활동으로, 소프트웨어 구현에 있어서 필요한 단계이다. 따라서 본 문서는 소프트웨어 설계 프로세스를 묘사하여, 개발자로 하여금 **Smart Noise** 시스템의 아키텍처를 잘 이해하도록 하는 것이 이 문서의 목적이다. 또한 개발자들은 본 문서를 통해 이해한 설계의 세부사항을 바탕으로 시스템을 구현할 예정이다. 이 뿐만 아니라 본 시스템의 소유자와 이해관계자, 클라이언트 엔지니어, 소프트웨어 테스터 등이 본 문서의 독자가 될 수 있으며, 해당 문서를 통해 **Smart Noise** 시스템 설계에 대한 세부적인 사항들을 이해할 수 있다.

1.4 문서 구성

본 문서는 다음과 같이 구성된다.

1. 머리말: 본 문서의 독자층과 문서의 범주 및 목적, 구조 등을 설명한다.
2. 서문: 본 문서에서 적용한 다이어그램과 툴에 대한 설명을 다루고 있다.
3. 시스템 컨텍스트 및 상호 작용: **System Context Model**과 **Interactive Model (Use Case Model)**을 이용하여, 시스템과 환경 간 관계에 대한 상호보완적 관점을 제공한다.

4. 전체 시스템 아키텍처: 시스템을 구성하는 주요 컴포넌트들과 그들 간의 상호 작용을 식별한다.
5. 설계 모델: **Smart Noise**의 세부 시스템들의 설계 모델과 인터페이스 명세서를 제공한다.
6. 인터페이스 디자인: 프론트엔드의 인터페이스 디자인을 명세한다.
7. 프로토콜 디자인: 애플리케이션, 서버, 디바이스와 같은 하위 시스템간의 상호작용에서 사용되는 프로토콜의 구조에 대해 다룬다.
8. 데이터베이스 디자인: 시스템 데이터 구조와 이것이 어떻게 데이터베이스에 반영될지를 설명한다.
9. 테스트 계획: 테스트 계획에 대해 다룬다.
10. 개발 계획: **Smart Noise**을 개발하기 위한 계획을 명시한다.
11. 추가 정보: 본 디자인 명세서의 추가 정보를 다룬다.
12. 참조: 본 디자인 명세서를 작성할 때 참조한 정보들을 다룬다.

2. 서론

스마트 흡 서비스는 1인 가구가 많아진 현상을 반영한다. 대한민국 1인 가구는 2021년 10월 기준 40%를 넘어갔으며, 이 중에서 50% 이상은 3,569세대이다. 보건복지부 자료에 의하면 그중에서 20%는 무호흡 환자인데, 이들 중 제대로 치료를 받는 이들은 10% 내외라고 알려졌다. 때문에, 무호흡 환자에 대한 꾸준한 수요는 존재했지만, 접근 방식에서 방향성을 정의하지 못했다.

다른 질병에 비해서 쉽게 회복할 수 있는 질병으로 착각하는 사례도 많았다. 때문에, 이런 사회의 그릇된 과신은 더 많은 환자 양산의 원인이 될 수밖에 없다. 소음으로 평가받으며 사고 예방 가능한 불확실성을 정량화하지 못한 소리를 지금보다 이해하고 산업에 적용한다면, 소음 데이터 활용 시장 적합성 및 앞으로 발전 방향에 대한 예측은 한결 정확해질 수 있다고 판단한다.

무호흡과 코골이는 타인의 숙면 방해라는 숙제도 지녔지만, 당사자의 건강에 치명적인 문제를 일으킨다. 무호흡은 코골이와 동반되기에 치료가 필요하지만 명확한 대안이 부족하다. 그렇다고 버려두면 기억력과 인지 능력까지 떨어뜨릴 수 있는 위험을 지녔다. 참고로 한국은 2018년 수면 무호흡증에 대한 수면다원검사 및 양압기치료의 건강보험급여가 인정되면서 사업용 운전자뿐만 아니라 일반인들의 수면 무호흡 진단 및 치료의 접근성 및 이용 가능성이 개선되었다. 그럼에도, 코골이가 있는 경우 두통, 피로를 포함해 다양한 신경과 질환 및 증상 동반 숙제를 안고 있다. 또한, 적절한 양압기 사용에도 주간 졸림을 호소하는 수면 무호흡환자들이 각성 촉진 약제에만 의존하는 것이 현실인 점이 환자들에게 고민인 부분이다.

단순코골이 내지 수면 무호흡의 정도는 수면다원검사를 통해 정확하게 진단받을 수 있지만 결국 해법은 예방이다. 무호흡이 지속하면 급작스러운 사망으로 이어지는 경우도 더러 있다. 양압기를 활용한 치료는 중등도 이상의 수면 무호흡환자의 뇌-심혈관 질환의 예방에만 효과가 있다. 주간 졸림을 일부 개선하고 교통사고 발생률도 낮춰주지만, 수면 상태의 호전된 부분까지는 점검하지 못한다. 과거와 달리 수면 무호흡은 이제 유의미한 소음으로 분류된다. “유의미하다”는 뜻은 산업으로 분류 가능하다는 의미와 같다. 상기도의 무호흡 또는 부분적 폐쇄 탓에 공기의 흐름이 제한되고, 산소포화도가 감소하게 된다. 다시 말해, 수면 중 교감신경 증가는 혈압과 맥박의 상승을 초래했던 과거와 달리 근본적인 원인 제거가 가능하다. 과거에는 대안이 부족했기에, 수면의 질이 떨어지는 것은 물론 기상 후 두통, 주간 졸림, 피로, 병명을 알 수 없는 기분 저하, 집중력 저하 같은 다양한 증상에서 해법을 찾지 못했다. 단순 약물치료에 의존하는 부분이 잦은데, 대부분 증상이 만성화되면서 문제가 불거졌는데, 예방할 수 있는 시스템이 부족했다는 의미다.

따라서 스마트 노이즈는 무호흡, 깨짐 소리, 쿵 소리 등과 같은 위험한 상황을 소리로 탐지해내고, 이를 빠르게 보호자랑 119에 알림으로써 사용자를 빠르게 도울 수 있는 솔루션을 제공한다. 본 디자인 명세서에서는 스마트 노이즈를 구현하기 위한 설계 프로세스와 시스템 아키텍처를 UML 다이어그램, 도표 등으로 설명하고 있다.

2.1 이 장에서는..

이 장에서는 해당 문서에서 적용한 다이어그램과 툴에 대한 설명을 다루고 있다.

2.2 적용 다이어그램

2.2.1 UML

UML (Unified Modeling Language)은 프로그램 설계를 표현하기 위해 사용하는, 주로 그림으로 된 표기법을 의미한다. 객체지향 언어와 밀접한 관련이 있기에 객체지향 모델링 언어라고도 불린다. 소프트웨어 시스템 모델링에 사용될 수 있는 13가지 다이어그램 유형들의 집합을 포함하여, 더 일반적인 시스템 모델링을 위하여 SysML과 같은 변형이 제안되었다.

2.2.2 Use Case 다이어그램

아이바르 제이콥슨에 의해 개발된 모델링, 각 **Use Case**는 시스템과의 외부 상호 작용과 관련된 별개의 작업을 나타낸다. **Use Case** 모델링을 지원하기 위한 UML 다이어그램 유형은 UML의 일부가 되었다. **Use Case**는 사용자가 어떤 인터페이스를 통해 시스템에 무엇을 기대하는지에 대해 간단히 기술하며 시스템과의 외부 상호 작용과 관련된 별개의 작업을 나타낸다. **Use Case**는 간단하게 타원으로 표시된다. 유스케이스에서 막대 모양 표기법은 사람과의 상호 작용을 다루기 위해 개발되었다.

2.2.3 Class 다이어그램

시스템의 클래스들과 그들과의 연관 관계를 보여주는 객체지향 시스템 모델 개발이 필요할 때 클래스 다이어그램이 사용된다. 클래스는 한 가지 종류의 시스템 개체를 일반적으로 정의한 것으로 생각할 수 있다. 때문에 여기서의 연관은 클래스들 사이에 어떤 관계가 있다는 것을 표시하는 클래스 사이의 연결을 뜻한다. 결과적으로 각 클래스는 이와 연결된 클래스에 대해 어느 정도 알고 있어야 한다.

UML 클래스 다이어그램은 상세 수준을 다르게 표현할 수 있다. 모델을 개발할 때, 첫 단계는 보통 실세계를 관찰한다. 이후에는 필수 객체들을 식별하고 이들을 클래스로 표현하게 된다. 이들 다이어그램을 작성하는 가장 간단한 방법은 클래스 이름을 상자 안에 기입하는 것이다. 클래스들 사이에 선을 연결하여 연관의 존재를 표시할 수 있다.

2.2.4 Sequence 다이어그램

액터와 시스템의 객체들 간의 상호 작용을 모델링하기 위해 주로 사용된다. UML이 여러 다른 종류의 상호 작용이 모델링 될 수 있도록 시퀀스 다이어그램을 위한 풍부한 문법을 갖고 있기 때문이다. 시퀀스 다이어그램은 특정 유스케이스나 유스케이스 인스턴스에서 일어나는 상호 작용의 순서를 보여준다.

시퀀스 다이어그램을 코드 생성, 상세 문서화에 사용하지 않는다고 가정하자. 그렇다면, 모든 상호 작용을 다이어그램에 포함시킬 필요는 없다. 개발 프로세스 전반부에서 요구 공학과 상위 수준 설계를 지원하기 위한 시스템 모델을 개발한다고 가정하자. 구현 결정에 따라 달라지는 상호 작용이 많아지게 된다. 가령, 권한 검사를 위해 어떻게 사용자 식별자를 얻는지는 구현 시까지 미뤄질 수 있는 것 중 하나이다. 어떤 구현에서는 이것이 유저 객체와 상호 작용하는 것과 연관될 수 있다. 때문에 이 단계에서는 이런 부분들이 중요하지 않기 때문에 시퀀스 다이어그램에 포함시킬 필요가 없다.

2.2.5 Context 다이어그램

시스템 명세 초기 단계에서는 어떤 것이 개발되는 시스템의 일부이고, 어떤 것이 아닌지에 대해 시스템의 경계를 정해야 한다. 이것이 시스템 이해 당사자들과 작업하며, 어떤 기능이 시스템에 포함되어야 하고, 시스템 운영 환경에서 어떤 처리 과정과 어떤 오퍼레이션이 수행되어야 하는지 결정하는 것과 관련되어 있다. 비지니스 프로세스들 중 개발되는 소프트웨어에서 자동화된 자원이 구현되어야 하는 것, 다른 시스템에 의해 지원되어야 하는 것을 결정해야 한다. 기존 시스템의 기능과 중복되는 것을 살펴봐야 하며, 어디에 새로운 기능이 구현되어야 하는지 결정해야 한다. 시스템 비용, 시스템 요구사항과 설계를 이해하는 데 필요한 시간을 제한하기 위해 이와 같은 결정은 일찍 내려져야 한다.

어떤 경우에는 시스템과 그 환경 사이의 경계가 비교적 명확하다. 예를 들어 기존의 수작업 또는 전산 시스템을 자동화 된 시스템으로 대체하는 경우, 새로운 시스템의 환경은 보통

기존 시스템의 환경과 같다. 다른 경우에는 좀 더 융통성이 있어서 요구공학 프로세스 중에 시스템과 환경 사이의 경계를 구성하는 것이 무엇인지 결정한다.

컨텍스트 모델은 보통 여러 자동화 된 시스템들을 포함하는 환경을 보여준다. 다만, 환경에 포함된 시스템들과 명세해지려는 시스템 간 관계들의 유형을 보여주지는 않는다. 외부 시스템들은 시스템을 위해 데이터를 생성하거나 시스템에서 생성된 데이터를 소비할 수 있다. 외부 시스템들은 시스템과 데이터를 공유할 수 있고, 외부 시스템들은 직접 연결되거나 네트워크를 통하거나 연결되지 않을 수도 있다. 시스템들은 물리적으로 같은 장소에 있거나 다른 건물에 있을 수 있다. 이들 관계 모두는 정의하려는 시스템의 요구 사항과 설계에 영향을 줄 수 있으므로 고려되어야 한다. 그러므로 간단한 컨텍스트 모델은 비지니스 프로세스 모델 같은 다른 모델들과 같이 사용된다. 비지니스 프로세스 모델은 특정 소프트웨어 시스템이 사용되는 자동화된 프로세스들과 사람들을 설명한다.

2.2.6 Entity Relationship 다이어그램

데이터 모델링 분야에서 개체-관계 모델이란 구조화된 데이터에 대한 일련의 표현이다. "구조"화된 데이터를 저장하기 위해 데이터베이스를 쓴다. 이 데이터의 "구조" 및 그에 수반한 제약 조건들은 다양한 기법에 의해 설계될 수 있다. 그 기법 중 하나가 개체-관계 모델링(Entity-Relationship Modelling)이다. 줄여서 ERM이라고 한다. ERM 프로세스의 산출물을 가리켜 개체-관계 다이어그램(Entity-Relationship Diagram)이라 한다. 줄여서 ERD라 일컫는다. 데이터 모델링 과정은 데이터 모델을 그림으로 표현하기 위해 표시법을 필요로 한다. ERD는 개념적 데이터 모델 혹은 시맨틱 데이터 모델의 한 타입이다.

2.3 적용 툴

2.3.1 Lucid Chart

이 툴은 해당 문서에서 context diagram, use-case diagram, server-client architecture model 등의 다이어그램을 그릴 때 사용하였다.

2.3.2 Miro

이 툴은 해당 문서에서 ER Diagram, 관계 계획 등의 다이어그램을 그릴 때 사용하였다.

2.3.3 Fusion 360

Fusion 360은 설계, 엔지니어링, 전자 기기 및 제조 작업을 수행할 수 있는 모델링 플랫폼이다. 본 프로젝트에서 하드웨어 디바이스를 위한 모델링 프로토타입을 생성하기 위해 사용하였다.

2.4 프로젝트 범위

1인 가구가 많아진 현대 시대에, 무호흡 환자에 대한 스마트 헤어 서비스의 수요는 꾸준히 증가하였다. 본 프로젝트와 비슷한 기능을 가진 서비스를 비교해보았을 때, 무호흡 환자를 대상으로 배포된 구글 알람 서비스는 2539 직장인을 대상으로 운영되며 구글의 내부 기능과 디바이스의 연동에 초점을 두고 있지만 실질적인 활용도가 매우 떨어진다. 또 다른 서비스로 갤럭시 위치를 살펴보면, MZ 세대 수도권 여성은 우선 구매 대상으로 운영되며 범용성도 높지만, 기기 자체의 가격과 무호흡 인식 기능에 한계를 보인다.

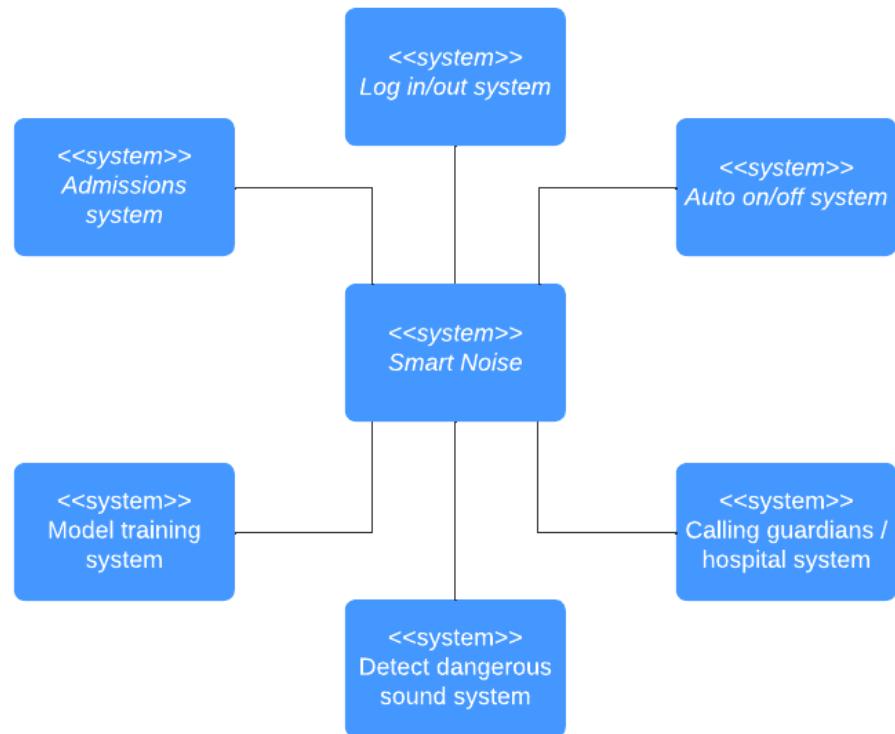
본 프로젝트는 수도권에 거주하는 3569 1인 가구를 타깃으로 시장에 접근하여 디바이스 앱 서비스를 추가로 제공하여 범용성을 높이고, 다양한 위험 신호를 분석하여 활용도를 높인다.

3. 시스템 컨텍스트 및 상호 작용

3.1 이 장에서는..

이 장에서는 **Smart Noise** 소프트웨어와 외부 환경 사이의 관계에 대한 이해를 발전시키기 위한 것이다. 이것은 요구되는 시스템 기능을 어떻게 제공하고 시스템을 그 환경과 통신하도록 어떻게 구조화하는지를 결정하는 데 핵심적이다. 또한 이것은 시스템의 경계를 확립하게 해주는데, 시스템의 경계는 설계 중인 시스템에 어떤 특징들이 구현되는지와 다른 관련된 시스템에 어떤 특징들이 있는지를 결정하는 데 도움이 된다. 이 장에서는 **System Context Model**과 **Interactive Model (Use Case Model)**을 이용하여, 시스템과 환경 간 관계에 대한 상호보완적 관점을 제공한다.

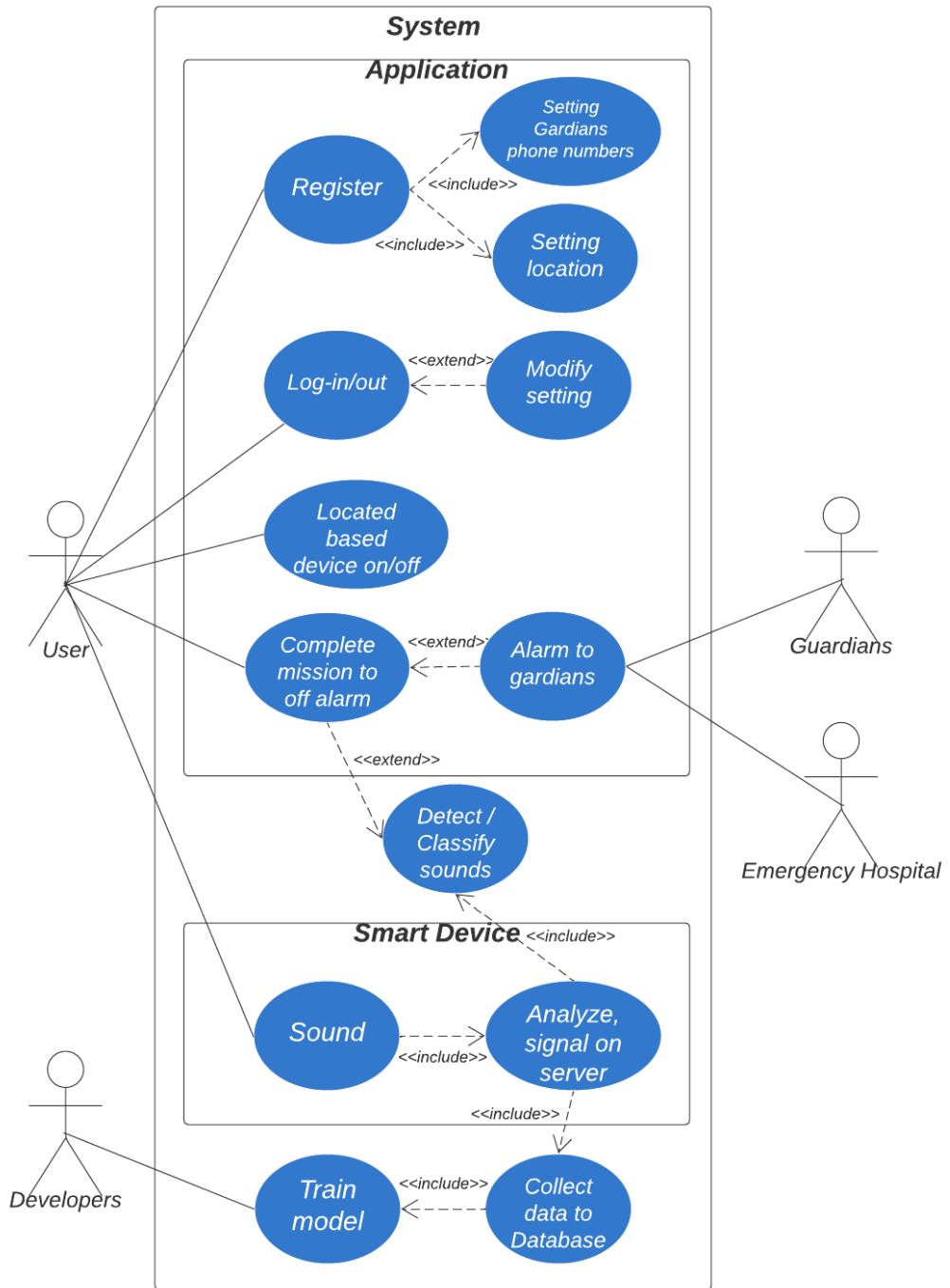
3.2 Context model



[그림 1] Context model

3.3 Use case

3.3.1 Use case diagram



[그림 2] Use case diagram

3.3.2 Use case 예시

① Smart Noise에 등록하는 사용자 Use case

[표 1] Smart Noise 등록하는 사용자 Use case

Actor	Smart Noise에 등록하는 사용자
Description	사용자는 Smart Noise 서비스에 등록하여 서비스를 이용할 수 있다.
Normal Course	<ol style="list-style-type: none"> 1. 사용자는 애플리케이션을 실행하여 등록 버튼을 누른다. 2. 사용자는 자신의 이름, 아이디, 비밀번호와 연락처를 작성하고 등록한다. <ol style="list-style-type: none"> 1. 전화번호 인증 절차를 진행한다. 2. 데이터베이스를 통해 아이디의 중복 여부를 확인한다. 3. 사용자 등록이 완료된다.
Precondition	N/A
Post Condition	사용자가 등록한 데이터는 암호화하여 안전하게 저장해야 한다.
Assumptions	사용자는 이전에 서비스에 등록한 적이 없어야 한다.

② 자신의 휴대 번호를 확인하고 싶은 사용자 Use case

[표 2] 자신의 휴대 번호를 확인하고 싶은 사용자 Use case

Actor	자신의 휴대 번호를 확인하고 싶은 사용자
Description	사용자는 애플리케이션에서 자신의 휴대 번호에 관한 정보를 확인할 수 있다.
Normal Course	<ol style="list-style-type: none"> 1. 사용자가 애플리케이션을 실행한다. 2. 애플리케이션은 서버로부터 전달받은 휴대 분석 데이터를 표시하며 유저는 이를 확인할 수 있다.
Precondition	서버로부터 받은 휴대 분석 결과 정보가 있어야 한다.

Post Condition	N/A
Assumptions	사용자는 정상적으로 로그인을 한 상태여야 한다.

③ Smart Noise Device를 등록하고 싶은 사용자 Use case

[표 3] Smart Noise Device를 등록하고 싶은 사용자 Use case

Actor	Smart Noise 디바이스를 등록하고 싶은 사용자
Description	사용자는 Smart Noise 디바이스를 블루투스를 통해 앱과 연동할 수 있다. 디바이스는 소리 신호를 측정하여 서버로 송신한다.
Normal Course	<ul style="list-style-type: none"> 1. 사용자가 애플리케이션을 실행한다. 2. 애플리케이션이 연동할 수 있는 디바이스들의 리스트가 나타난다. 3. 사용자는 연결 버튼을 눌러 원하는 Smart Noise 디바이스를 연결할 수 있다. 4. 사용자는 연결 해제 버튼을 눌러 원하는 Smart Noise 디바이스의 연결을 해제할 수 있다.
Precondition	블루투스 통신을 통해 연결 가능한 기기들을 확인할 수 있어야 한다.
Post Condition	기기 등록 정보를 애플리케이션에 저장해야 한다.
Assumptions	사용자에게 정상적으로 작동하는 Smart Noise 디바이스가 있어야 한다.

④ 위험에 처한 사용자 및 보호자 Use case

[표 4] 위험에 처한 사용자 및 보호자 Use case

Actor	위험에 처한 사용자와 그의 보호자
--------------	--------------------

Description	사용자의 상태가 위험하다고 판단됐을 시, 보호자는 이것에 대한 알림을 수신받게 된다.
Normal Course	<ol style="list-style-type: none"> 1. 위험 소음이 감지되어 애플리케이션을 통해 사용자에게 미션이 제공된다. 2. 사용자가 미션 수행에 실패한다. 3. 서버에서 보호자에게 알림을 전송하며, 보호자는 이를 확인할 수 있다.
Precondition	사용자의 위치 정보에 접근할 수 있어야 한다.
Post Condition	N/A
Assumptions	서버에 보호자의 정보가 등록되어 있어야 한다.

⑤ 위험에 처한 사용자 및 병원 Use case

[표 5] 위험에 처한 사용자 및 병원 Use case

Actor	위험에 처한 사용자, 병원
Description	사용자가 위험에 처했다고 판단됐을 시, 병원에 연락하여 구조를 요청한다
Normal Course	<ol style="list-style-type: none"> 1. 위험 소음이 감지되어 애플리케이션을 통해 사용자에게 미션이 제공된다. 2. 사용자가 미션 수행에 실패한다. 3. 서버에서 병원에 연락하여 119를 호출한다.
Precondition	사용자의 위치 정보에 접근할 수 있어야 한다.
Post Condition	사용자가 실수로 미션을 실패했을 경우를 대비하여, 구조 호출은 취소할 수 있어야 한다.
Assumptions	서버에 병원 정보가 등록되어 있어야 한다.

⑥ 딥러닝 모델 개발자 Use case

[표 6] 딥러닝 모델 개발자 Use case

Actor	딥러닝 모델 개발자
Description	위험 소리 탐지 및 분류 모델의 성능을 높이기 위해, 기존에 있던 모델을 새로 수집한 데이터셋으로 학습시키거나, 아예 새로운 구조의 알고리즘을 학습시킨다.
Normal Course	<ol style="list-style-type: none">1. 위험 소리로 탐지된 소리 중 예측값이 높은 데이터를 데이터베이스에 저장한다.2. 수집한 데이터를 학습 데이터로 해서 기존 모델을 더 학습시킨다.3. 수집한 데이터를 학습 데이터로 해서 다른 구조의 알고리즘으로 모델을 학습 시킨다.4. 테스트 데이터셋으로 학습한 모델의 성능을 평가한다.
Precondition	정제된 데이터여야 한다. 학습 데이터셋과 테스트 데이터셋은 중복이 없어야 한다.
Post Condition	학습된 모델의 성능은 98%(AUC) 이상이어야 한다.
Assumptions	학습이 가능할 정도로 충분한 데이터셋이 구축되어 있다고 가정한다.

4. 전체 시스템 아키텍처

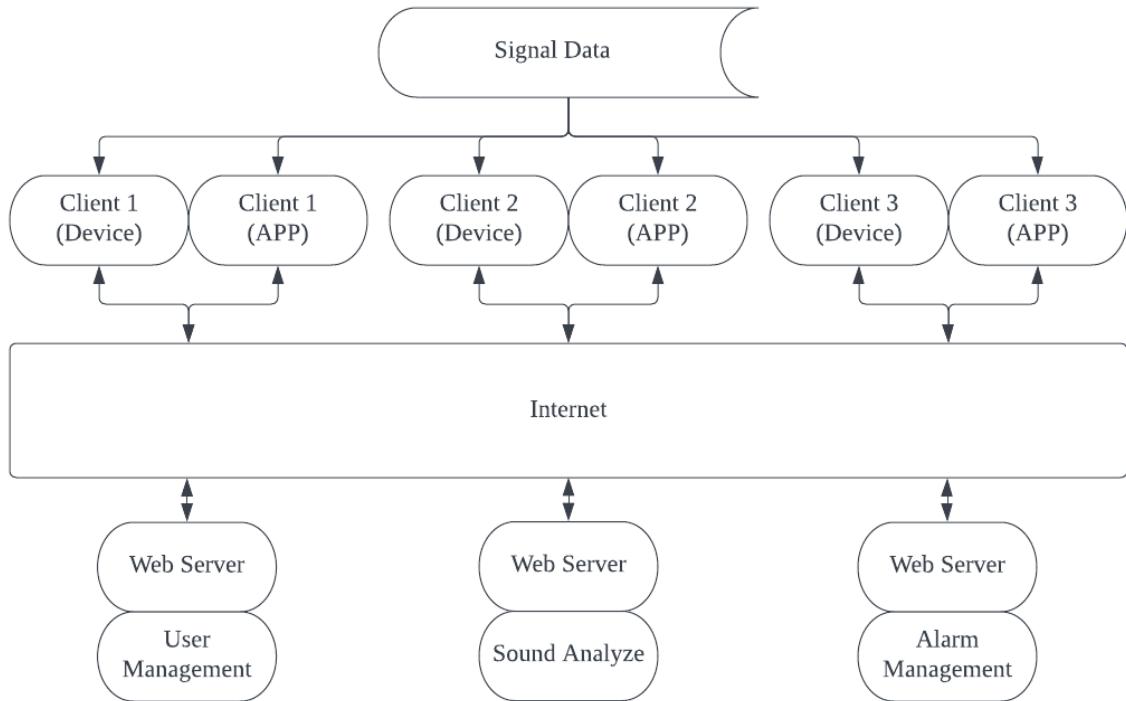
4.1 이 장에서는..

이 장에서는 3장에서 정의한 소프트웨어 시스템과 시스템의 환경 간 상호 작용 정보를 바탕으로, 시스템을 구성하는 주요 컴포넌트들과 그들 간의 상호 작용을 식별한다. 해당 문서에서는 **client-server** 아키텍처 패턴을 사용하여 **Smart Noise**의 아키텍처를 설명한다.

4.2 시스템 구성

Smart Noise는 실시간으로 가내의 위험 소리를 탐지해내는 시스템으로, 사용자가 위험 상황이 발생하였을 때 보호자 및 응급 의료 센터에 이를 빠르게 알릴 수 있는 시스템이다.

가내의 디바이스를 통해 소리를 수집하고, 이 소리 데이터를 서버로 보내야 하기 때문에, 서버-클라이언트 간 네트워크 연결이 중요하다. 그러므로 해당 문서는 서버-클라이언트 모델 형식으로 전체 시스템 아키텍처를 보여준다.



[그림 3] 전체 시스템의 아키텍처

Smart Noise의 서비스는 크게 3가지 서비스로 나눌 수 있는데, 첫 번째로 사용자 관리 서비스이다. 이 서비스는 사용자의 정보 데이터를 관리하며 해당 서버는 **admission**, 로그인/로그아웃, 디바이스 관리 등 사용자 정보와 관련된 서비스들을 제공한다. 이와 같은 서비스들은 주로 스마트폰 앱을 통해 제공되며, 앱에서 서버로 필요한 서비스를 요청한다.

두 번째로는 위험 소리를 탐지 및 분류하는 서비스이다. **Smart Noise Device**를 통해 수집한 소리는 딥러닝 기반의 탐지/분류 모델을 통해, 위험 소리인지 아닌지를 판별하게 된다. 이 때, 탐지/분류 모델은 서버 단에 있으며, **Smart Noise Device**는 서버에 위험 소리 판별 요청을 하게 된다. 추가 서비스로 만약 이 때 ‘호흡 곤란 소리’가 탐지되었다면, 서버는 **Smart Noise Device**에게 산소를 배출할 것을 요청한다.

세 번째는 위험 상황이 발생하였을 때, 이를 관리하는 시스템이다. 이 시스템을 담당하는 서버는 위험 상황이 발생했을 때, 조건에 따라 사용자의 보호자와 응급 의료 센터에 위험 상황을 알린다.

이처럼 서버-클라이언트 구조는 **Smart Noise**처럼 서비스들의 집합이 시스템을 이루고, 각 서비스가 서로 나누어진 서버에 전달되는 구조를 잘 설명한다. 이 구조는 네트워크 상에서

서버가 분산될 수 있으며, 일반적인 기능은 모든 서비스에서 구현될 필요가 없다는 장점을 가지고 있다. 하지만, 이 구조는 Dos 공격이나 서버 failure에 취약하고, 네트워크 상태에 따라 성능(ex - 속도)의 문제가 생길 수 있다.

5. 설계 모델

5.1 이 장에서는..

이 장에서는 Smart Noise의 설계 모델을 명시한다. 5.2장에서 시스템의 전반적인 디자인 모델을 Sequence diagram과 Data flow diagram으로 보여준다. 5.3장부터 5.8장까지는 Smart Noise의 세부 시스템의 정보를 다루는데, 다음과 같은 내용을 포함한다.

✓ 객체 클래스 식별

시스템의 핵심 객체들과 오퍼레이션을 식별하는 과정이다. 대략적인 시스템 기술로부터 초기에 식별된 객체 클래스, 속성, 오퍼레이션은 설계를 위한 시작점이 될 수 있다. 그 다음 응용 도메인 지식 또는 시나리오 분석으로부터의 정보는 초기 객체들을 정렬하고 확장하는데 사용될 수 있다. 또한 시스템 외부로의 개체를 표현하는 객체들뿐만 아니라 검색과 유효성 검사와 같이 일반적인 서비스를 제공하기 위해 사용되는 “구현 객체”도 설계할 수 있다.

✓ 설계 모델

설계 또는 시스템 모델은 시스템의 객체들과 객체 클래스들을 보여준다. 또한 이 모델은 개체들 간의 연관과 관계도 보여준다. 이 모델은 시스템 요구사항과 시스템 구현 사이의 연결이며, 프로그래머들이 구현 결정을 내릴 수 있도록 충분히 상세한 내용을 포함해야 한다. 설계를 작성하기 위하여 UML을 사용할 때 다음 두 종류의 설계 모델을 개발하였다.

- ① 구조 모델: 시스템의 정적 구조를 객체 클래스와 그들 간의 관계를 이용하여 기술한다.
- ② 동적 모델: 시스템의 동적 구조를 기술하고 시스템 객체 간에 예상되는 런타임 상호작용을 보여준다.

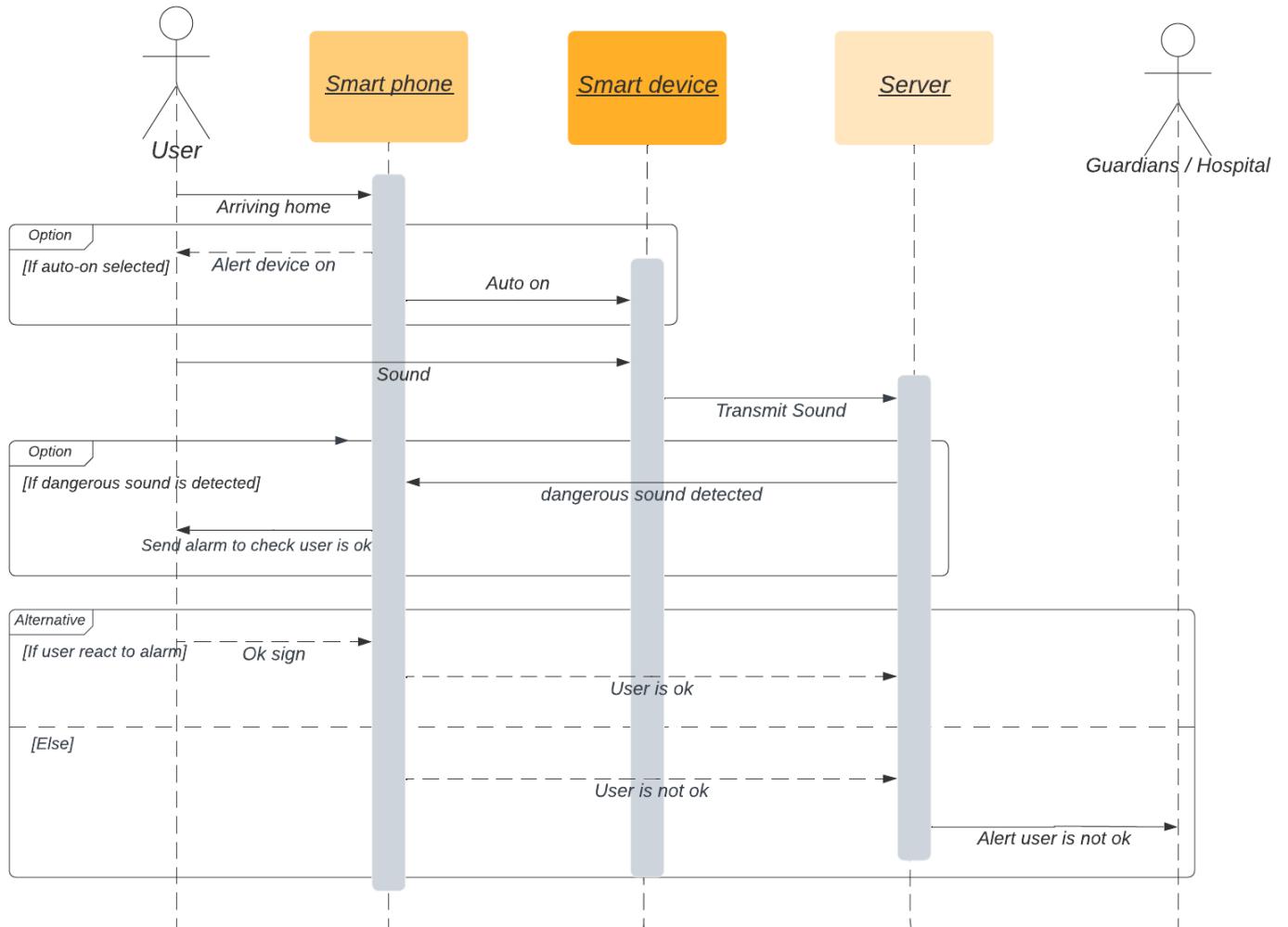
해당 문서에서는 구조 모델로 Class diagram, 동적 모델로 Sequence diagram를 사용하였다.

✓ 인터페이스 명세

설계 컴포넌트 간의 인터페이스 명세이다. 객체들과 서브 시스템들이 병행하여 설계될 수 있기 때문에 인터페이스를 명시할 필요가 있다. 인터페이스 설계는 객체의 인터페이스 또는 객체 그룹에 의해 제공되는 서비스들의 시그니처와 의미를 정의하는 것을 의미한다. 인터페이스는 클래스 다이어그램과 동일한 표기법을 사용하여 UML에서 명시될 수 있다.

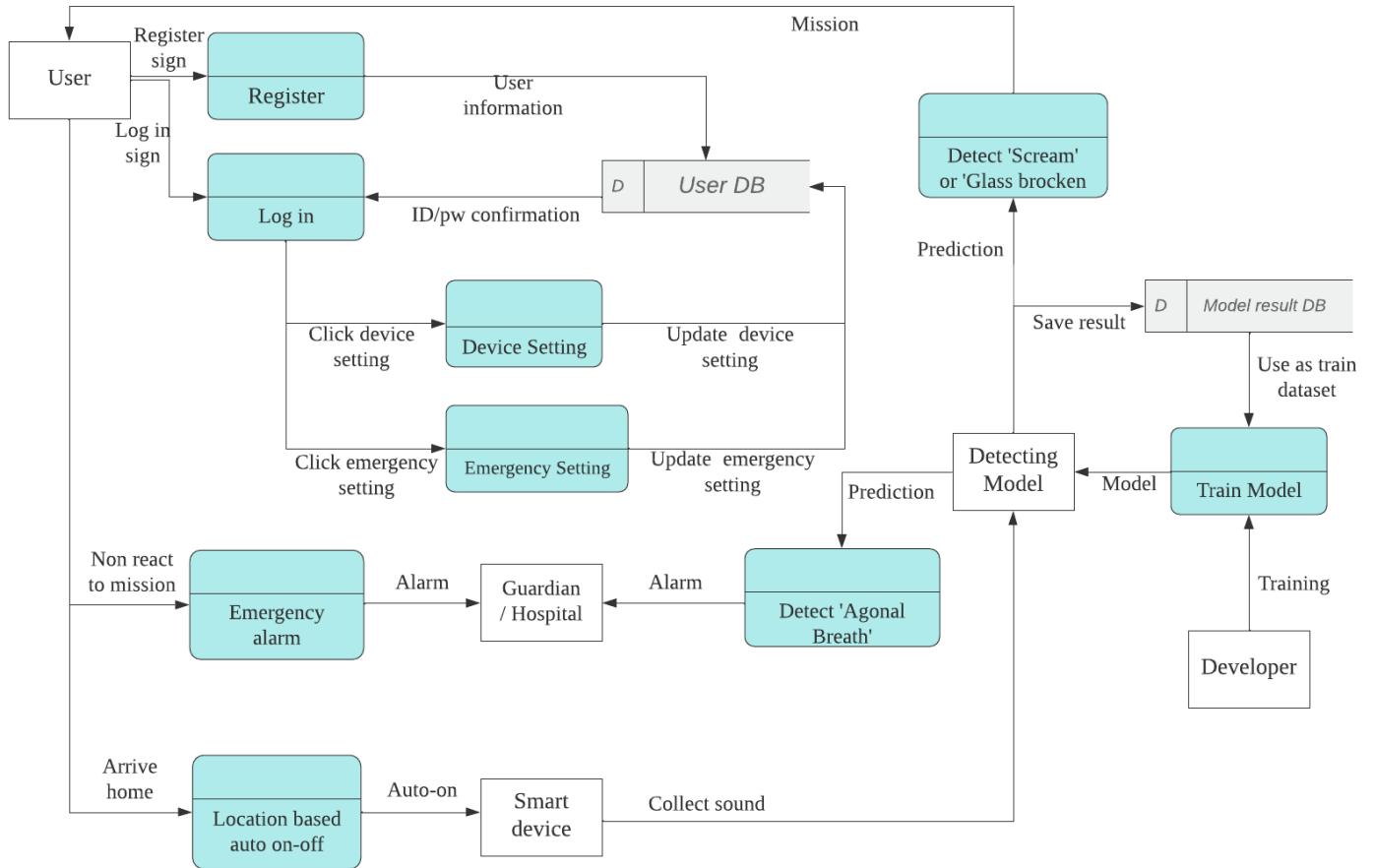
5.2 Overall Design model

5.2.1 Sequence 다이어그램



[그림 4] Sequence 다이어그램

5.2.2 Data flow diagram



[그림 5] Data flow diagram

5.3 사용자 정보 등록 시스템

해당 컴포넌트는 Smart Noise 서비스를 이용하는 사용자들의 정보를 다룬다. 사용자는 서비스를 사용하기 전 사용자 등록을 해야 하고, 사용자 등록이 되었을 때 로그인을 할 수 있으며 Smart Noise가 제공하는 서비스를 이용할 수 있다.

5.3.1 객체 클래스 식별

5.3.1.1 attribute

프로필 클래스는 다음과 같은 속성을 가지고 있다.

- 1) user_id: 사용자의 ID를 의미한다. 모든 사용자는 unique한 ID를 가지고 있다.
- 2) user_info: 사용자 정보를 의미한다.

프로필 클래스의 user_info는 다음과 같은 속성을 가지고 있다.

- 1) user_id: 사용자의 ID를 의미한다.
- 2) password: 사용자의 비밀번호를 의미한다.
- 3) name: 사용자의 이름을 의미한다.
- 4) email: 사용자의 이메일을 의미한다.
- 5) num: 사용자의 전화번호를 의미한다.
- 6) blood_type: 사용자의 혈액형을 의미한다.
- 7) allergies: 사용자가 앓고 있는 알레르기를 의미한다.
- 8) diseases: 사용자가 앓고 있는 병명을 의미한다.
- 9) emergency_contact: 사용자의 보호자를 의미한다.
- 10) location: 사용자의 위치를 의미한다.

EC 오브젝트는 다음과 같은 속성을 가지고 있다.

- 1) user_id: 사용자의 ID를 의미한다.
- 2) ec_name: 사용자의 보호자 이름을 의미한다.
- 3) ec_rel: 사용자와 보호자의 관계를 의미한다.
- 4) ec_num: 사용자의 보호자 전화번호를 의미한다.

디바이스 클래스는 다음과 같은 속성을 가지고 있다.

- 1) user_id : 사용자의 id를 의미한다.
- 2) record_time : 소리를 기록할 주기이며 사용자가 초기에 설정할 수 있다.
- 3) sensibility : 사용자가 분석할 소리의 종류에 따라 소리의 민감도를 설정할 수 있다.

5.3.1.2 Method

✓ 애플리케이션

프로필 클래스는 다음과 같은 메서드를 가지고 있다.

- 1) setProfile()
- 2) getInfo()
- 3) editInfo()
- 4) showInfo()
- 5) setLocation()
- 6) showLocation()

✓ 디바이스

디바이스 클래스는 다음과 같은 메서드를 가지고 있다.

- 1) set()

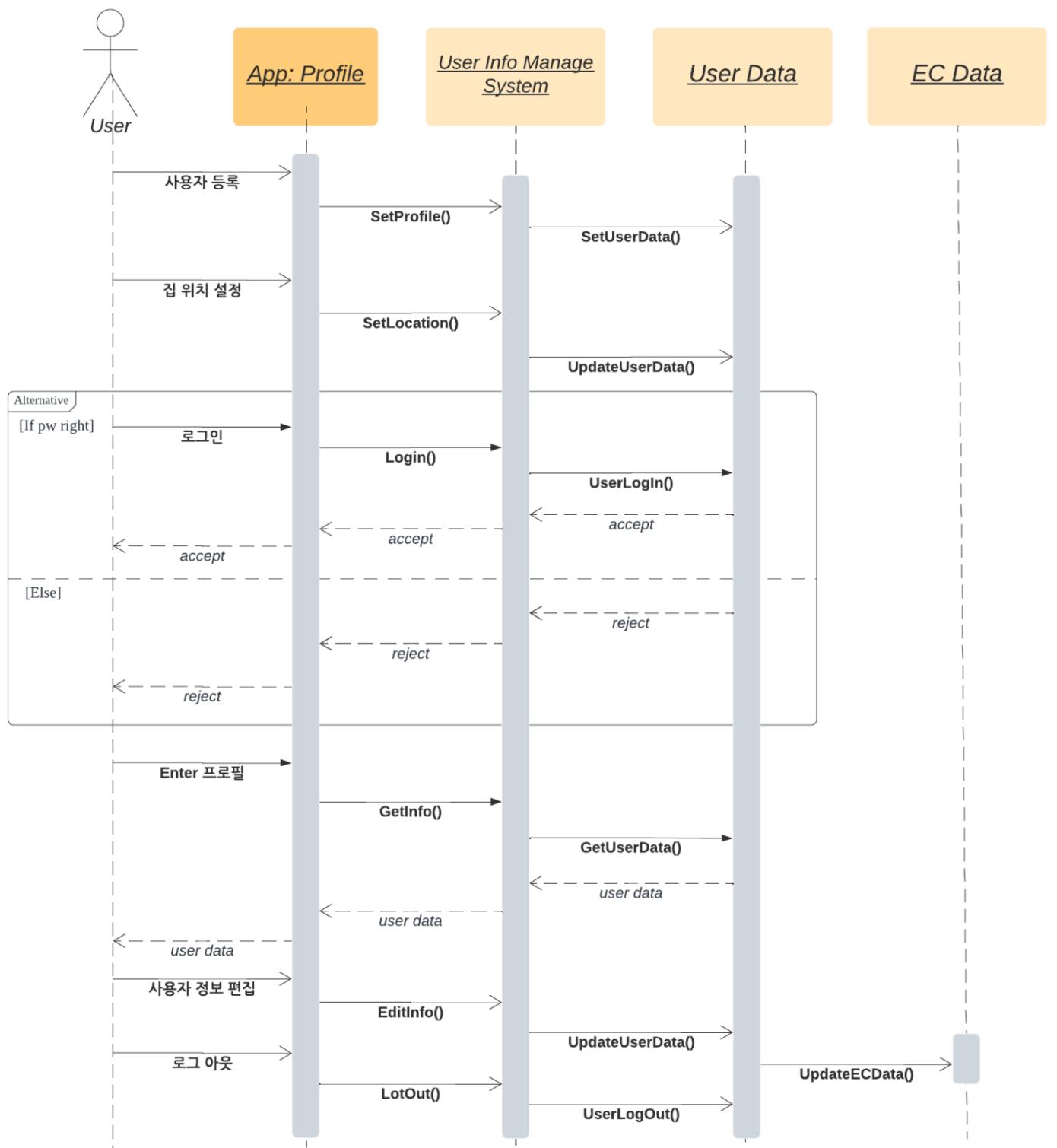
✓ 백엔드

사용자 정보 등록 시스템은 다음과 같은 메서드를 가지고 있다.

- 1) setUserData()
- 2) getUserData()
- 3) updateUserData()
- 4) updateECData()
- 5) userLogIn()
- 6) userLogOut()

5.3.2 설계 모델

사용자가 앱을 통해 사용자의 정보를 등록, 편집할 때와 로그인/로그아웃 할 때의 컴포넌트 간 상호작용을 **sequence model**로 순차적으로 보여준다.

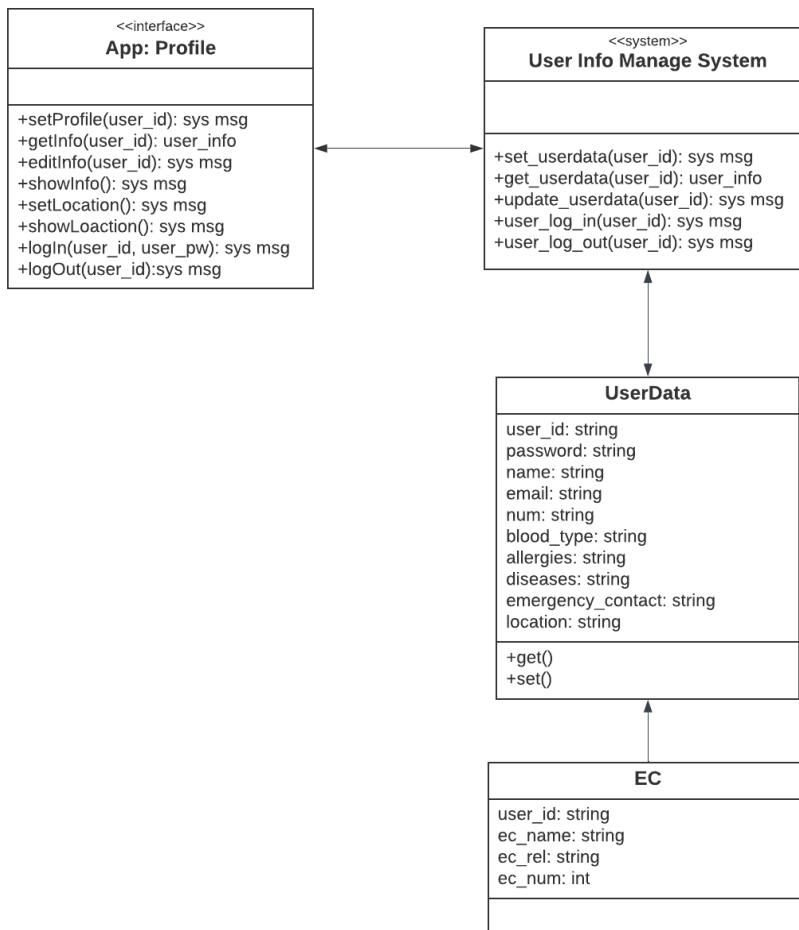


[그림 6] 사용자 정보 등록 시스템 - 설계 모델

[표 7] 사용자 정보 등록 시스템 - 설계 모델

서브 시스템	설명
사용자 정보 등록 / 편집	사용자의 정보가 SetProfile() , EditInfo() 함수를 통해 등록, 편집되었을 때, 이 정보는 SetUserData() , UpdateUserData() 함수를 통해 User Data 데이터 베이스에 반영된다.
사용자 정보 반환	사용자가 자신의 정보를 보고자 프로필 화면에 접근하면, GetInfo() 함수를 통해 시스템에 정보를 요청하고, 시스템은 GetUserData() 함수를 통해 User Data 데이터 베이스로부터 받은 사용자 정보를 반환한다.
로그인 / 로그아웃	사용자가 로그인, 로그 아웃을 할 것을 LogIn() , LogOut() 함수로 요청하면, 시스템은 UserLogIn() , UserLogOut() 함수를 통해 로그인, 로그아웃을 한다.

5.3.3 인터페이스 명세



[그림 7] 사용자 정보 등록 시스템 - 인터페이스 명세

5.4 사용자 데이터 관리 시스템

해당 컴포넌트는 **Smart Noise** 서비스를 이용하는 사용자들의 호흡 곤란 빈도수에 대한 정보를 다룬다. **Smart Noise** 디바이스와 스마트폰에서 감지된 소음 중 웹 서버가 호흡 곤란으로 감지한 횟수를 그래프로 표시한다. 또한, 호흡곤란이 감지된 빈도수에 따라 사용자의 상태를 **Normal** 또는 **Danger**로 표시한다.

5.4.1 객체 클래스 식별

5.4.1.1 attribute

데이터 클래스는 다음과 같은 속성을 가지고 있다.

- 1) `user_id`: 사용자의 ID를 의미한다.
- 2) `user_sensed`: 사용자 호흡 곤란 빈도수 데이터를 의미한다.

`sound_data` 클래스는 다음과 같은 속성을 가지고 있다.

- 1) `user_id`: 사용자의 ID를 의미한다.
- 2) `number_day`: 하루에 호흡곤란이 감지된 횟수를 의미한다.
- 3) `number_total`: 현재까지 호흡곤란이 감지된 총 횟수를 의미한다.
- 4) `number_avg`: 평균 호흡곤란 감지 횟수를 의미한다.

`nday` 클래스는 다음과 같은 속성을 가지고 있다.

- 1) `date`: 호흡곤란이 감지된 날짜를 의미한다.
- 2) `n_times`: 해당 날짜에 호흡 곤란이 감지된 횟수를 의미한다.

5.4.1.2 Method

✓ 애플리케이션

데이터 클래스는 다음과 같은 메서드를 가지고 있다.

- 1) `getData()`
- 2) `analyzeData()`
- 3) `showData()`
- 4) `showStatus()`

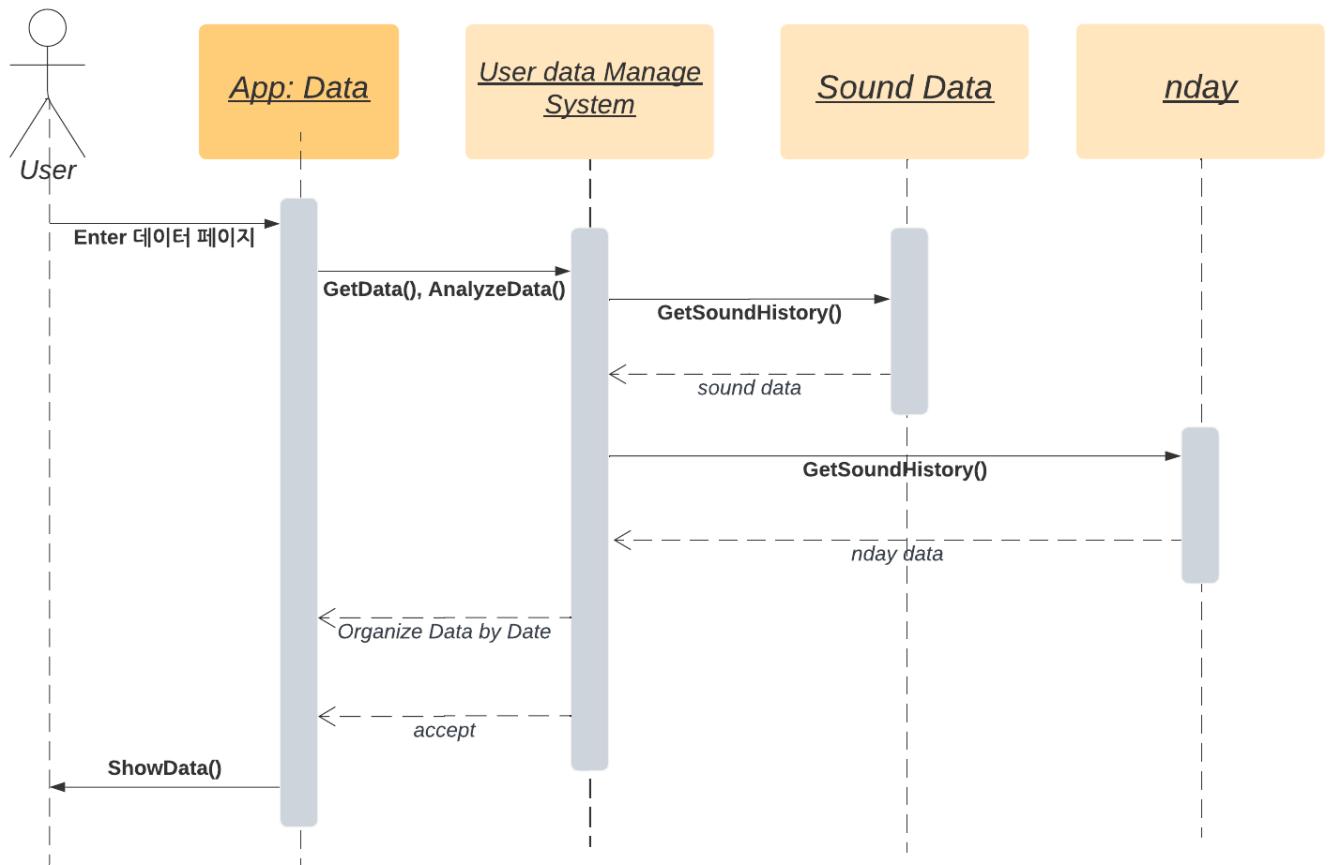
✓ 백엔드

사용자 데이터 관리 시스템은 다음과 같은 메서드를 가지고 있다.

- 1) `getSoundHistory()`

5.4.2 설계 모델

사용자가 앱을 통해 사용자의 데이터를 보고자 할 때의 컴포넌트 간 상호작용을 Sequence diagram으로 보여준다.

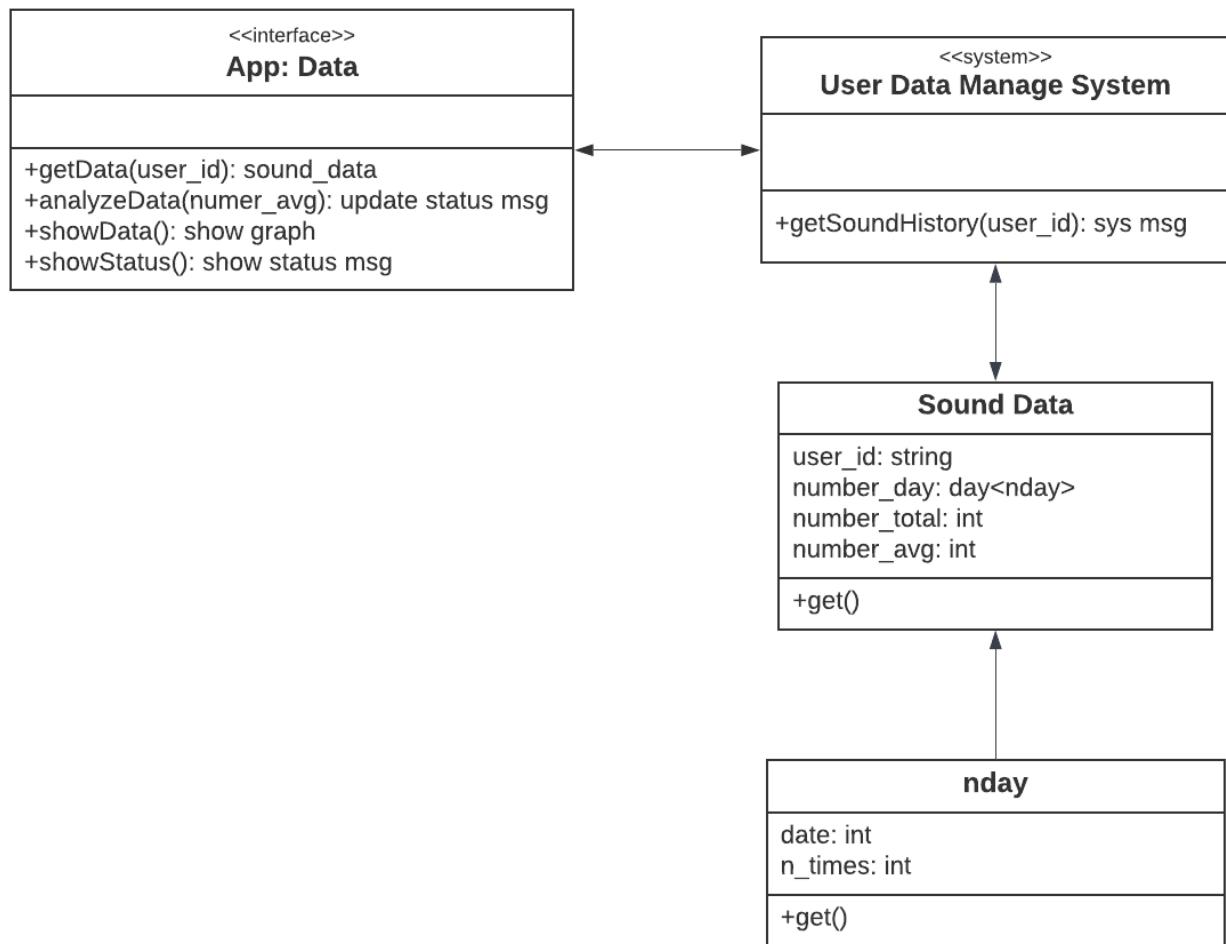


[그림 8] 사용자 데이터 관리 시스템 - 설계 모델

[표 8] 사용자 데이터 관리 시스템 - 설계 모델

서브 시스템	설명
데이터 분석 결과 반환	사용자가 자신의 소리 데이터 분석 결과를 보고자 데이터 페이지에 접근하면 GetData(), AnalyzeData() 함수를 통해 시스템에 정보를 요청한다. 시스템은 GetSoundHistory() 함수를 통해 Sound Data 데이터베이스로부터 소리 데이터를 가져온 후 이를 분석하여 반환한다.

5.4.3 인터페이스 명세



[그림 9] 사용자 데이터 관리 시스템 - 인터페이스 명세

5.5 응급 상황 미션 관리 시스템

해당 컴포넌트는 위험 소음이 감지되었을 때 사용자의 의식을 확인하기 위한 미션 제공 및 수행에 대해 다룬다.

5.5.1 객체 클래스 식별

5.5.1.1 attribute

응급상황 미션 클래스는 다음과 같은 속성을 가지고 있다.

- 1) user_id: 사용자의 ID를 의미한다.
- 2) mission: 사용자가 선택한 미션을 의미한다.

profile 클래스는 다음과 같은 속성을 가지고 있다.

- 1) user_id: 사용자의 ID를 의미한다. 모든 사용자는 다른 ID를 가지고 있다.
- 2) user_info: 사용자 정보를 의미한다.

프로필 클래스의 user_info는 다음과 같은 속성을 가지고 있다.

- 1) user_id: 사용자의 ID를 의미한다.
- 2) name: 사용자의 이름을 의미한다.
- 3) email: 사용자의 이메일을 의미한다.
- 4) num: 사용자의 전화번호를 의미한다.
- 5) blood_type: 사용자의 혈액형을 의미한다.
- 6) allergies: 사용자가 앓고 있는 알레르기를 의미한다.
- 7) diseases: 사용자가 앓고 있는 병명을 의미한다.
- 8) emergency_contact: 사용자의 보호자를 의미한다.
- 10) location: 사용자의 위치를 의미한다.

EC 오브젝트는 다음과 같은 속성을 가지고 있다.

- 1) user_id: 사용자의 ID를 의미한다.
- 2) ec_name: 사용자의 보호자 이름을 의미한다.
- 3) ec_rel: 사용자와 보호자의 관계를 의미한다.
- 4) ec_num: 사용자의 보호자 전화번호를 의미한다.

디바이스 클래스는 다음과 같은 속성을 가지고 있다.

- 1) chk : 알람을 울려야하는지를 판단하는 boolean 값이다.
- 2) alarm_fin : 알람을 끄는데 사용되는 boolean 값이다.

5.5.1.2 Method

✓ 애플리케이션

프로필 클래스는 다음과 같은 메서드를 가지고 있다.

- 1) setMission()
- 2) showSelected()
- 3) warningSig()
- 4) sendMission()
- 5) showMission()
- 6) successMission()
- 7) showSuccess()
- 8) failMission()
- 9) call119()
- 10) callGuardian()
- 11) showFail()
- 12) cancelCall()
- 13) popOff()

✓ 디바이스

디바이스 클래스는 다음과 같은 메서드를 가지고 있다.

- 1) make_alarm()

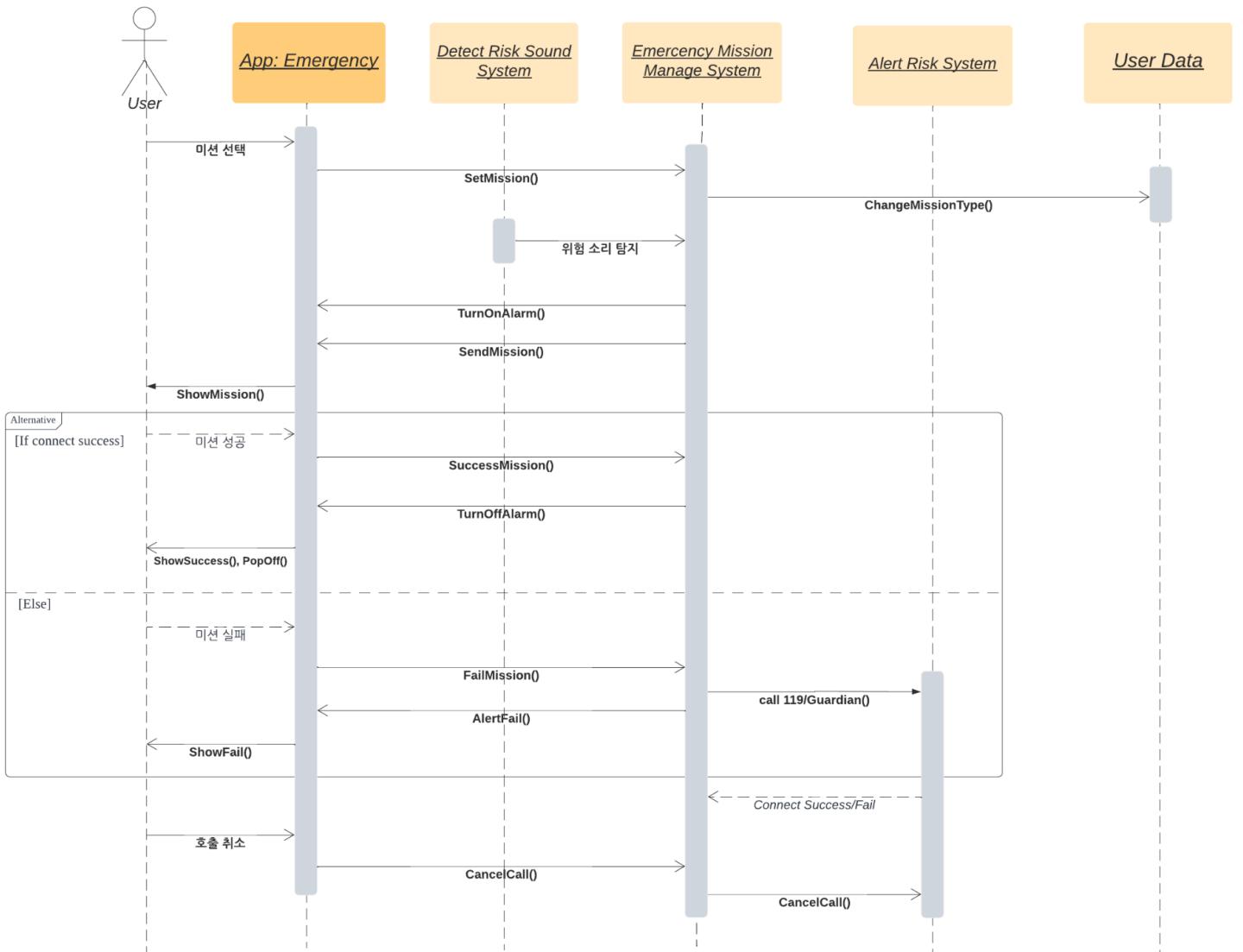
✓ 백엔드

응급 상황 미션 관리 시스템 클래스는 다음과 같은 메서드를 가지고 있다.

- 1) changeMissionType()
- 2) turnOnAlarm()
- 3) sendMission()
- 4) turnOffAlarm()
- 5) call119()
- 6) callGuardian()

5.5.2 설계 모델

사용자의 위험 상황이 감지된 후, 사용자의 반응을 확인할 때의 컴포넌트 간 상호 작용을 Sequence diagram으로 보여준다.



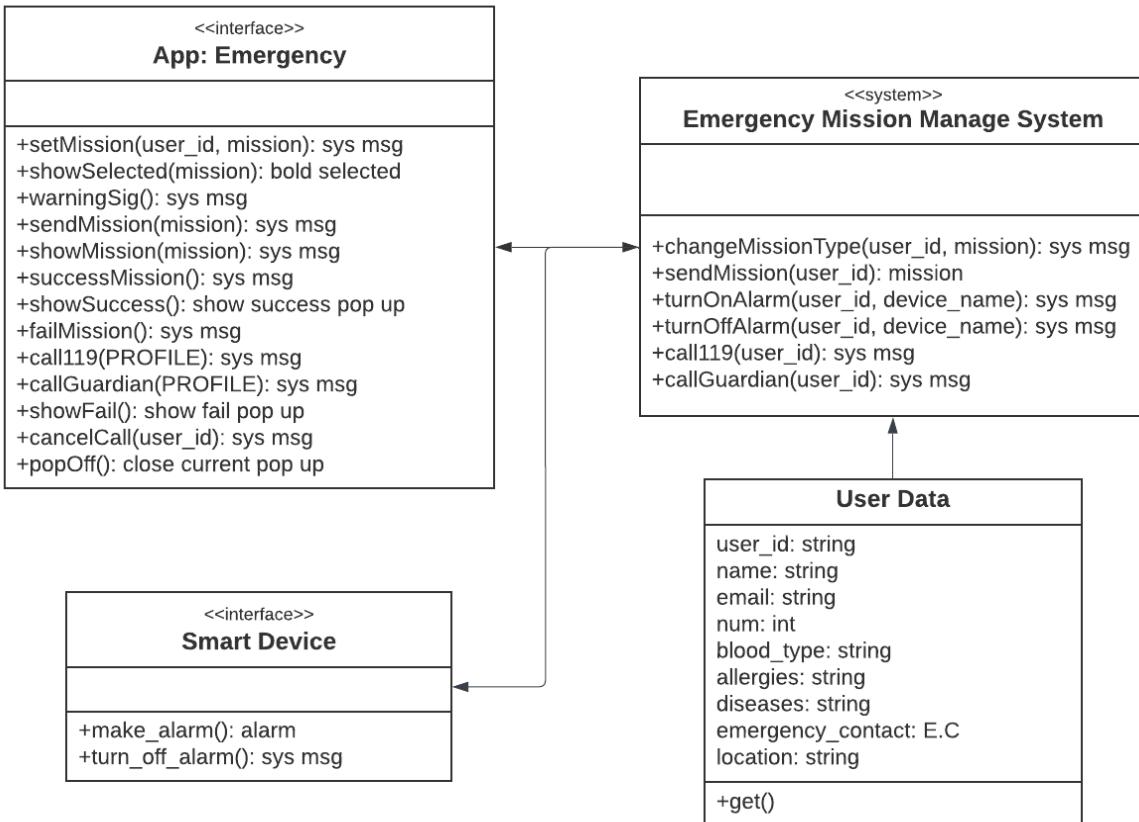
[그림 10] 응급 상황 미션 관리 시스템 - 설계 모델

[표 9] 응급 상황 미션 관리 시스템 - 설계 모델

서브 시스템	설명
미션 선택	사용자가 미션의 유형을 선택하면, SetMission()을 통해 시스템에 미션 유형을 변경할 것을 요청한다. 시스템은 ChangeMissionType()을 통해 User Data 데이터베이스에서 사용자 미션 정보를 변경한다.

미션	<p>위험 소리 탐지 시스템으로부터 사용자가 위험한 상황임이 탐지되었을 때, 사용자의 반응을 확인하기 위하여, TurnOnAlarm(), SendMission()를 통해 사용자에게 미션을 보낸다.</p> <ul style="list-style-type: none"> - 미션이 성공하였을 경우, SuccessMission()을 통해 시스템에 사용자 반응이 확인되었음을 알린다. 시스템은 TurnOffAlarm()을 통해 켜져 있는 알람을 끄도록 한다. - 미션이 실패하였을 경우, FailMission()을 통해 시스템에 사용자 반응이 없음을 알린다. 시스템은 Call119/Guardians()를 통해 119와 보호자에게 사용자의 위험 상황을 알린다.
호출 취소	<p>사용자가 피치 못한 상황으로 미션을 실패하였으나, 실제로 위험한 상황이 아닌 경우, 사용자는 호출 취소를 할 수 있다. 시스템은 CancelCall()을 통해 119와 보호자에게 위험 상황을 더 이상 알리지 않는다.</p>

5.4.3 인터페이스 명세



[그림 11] 응급 상황 미션 관리 시스템 - 인터페이스 명세

5.6 디바이스 연결 관리 시스템

해당 컴포넌트는 Smart Noise 디바이스와의 연결에 대해 다룬다.

5.6.1 객체 클래스 식별

5.6.1.1 attribute

디바이스 연결 클래스는 다음과 같은 속성을 가지고 있다.

- 1) user_id: 사용자의 ID를 의미한다.
- 2) device: 사용자가 사용하는 디바이스를 의미한다.

device 오브젝트는 다음과 같은 속성을 가지고 있다.

- 1) user_id: 사용자의 ID를 의미한다.
- 2) device_id: 디바이스의 ID를 의미한다.
- 3) connected_dev: 디바이스와 연결된 디바이스를 의미한다.

`conn_dev` 오브젝트는 다음과 같은 속성을 가지고 있다.

- 1) `user_id`: 사용자의 ID를 의미한다.
- 2) `conn_dev_id`: 연결된 디바이스의 ID를 의미한다.

5.6.1.2 Method

✓ 애플리케이션

프로필 클래스는 다음과 같은 메서드를 가지고 있다.

- 1) `showDevices()`
- 2) `connectDevice()`
- 3) `setConnected()`
- 4) `turnON()`
- 5) `establishConnection()`
- 6) `disconnectDevice()`
- 7) `setDisconnected()`
- 8) `turnOFF()`

✓ 디바이스

디바이스는 다음과 같은 메서드를 가지고 있다.

- 1) `bluetoothConnect()`

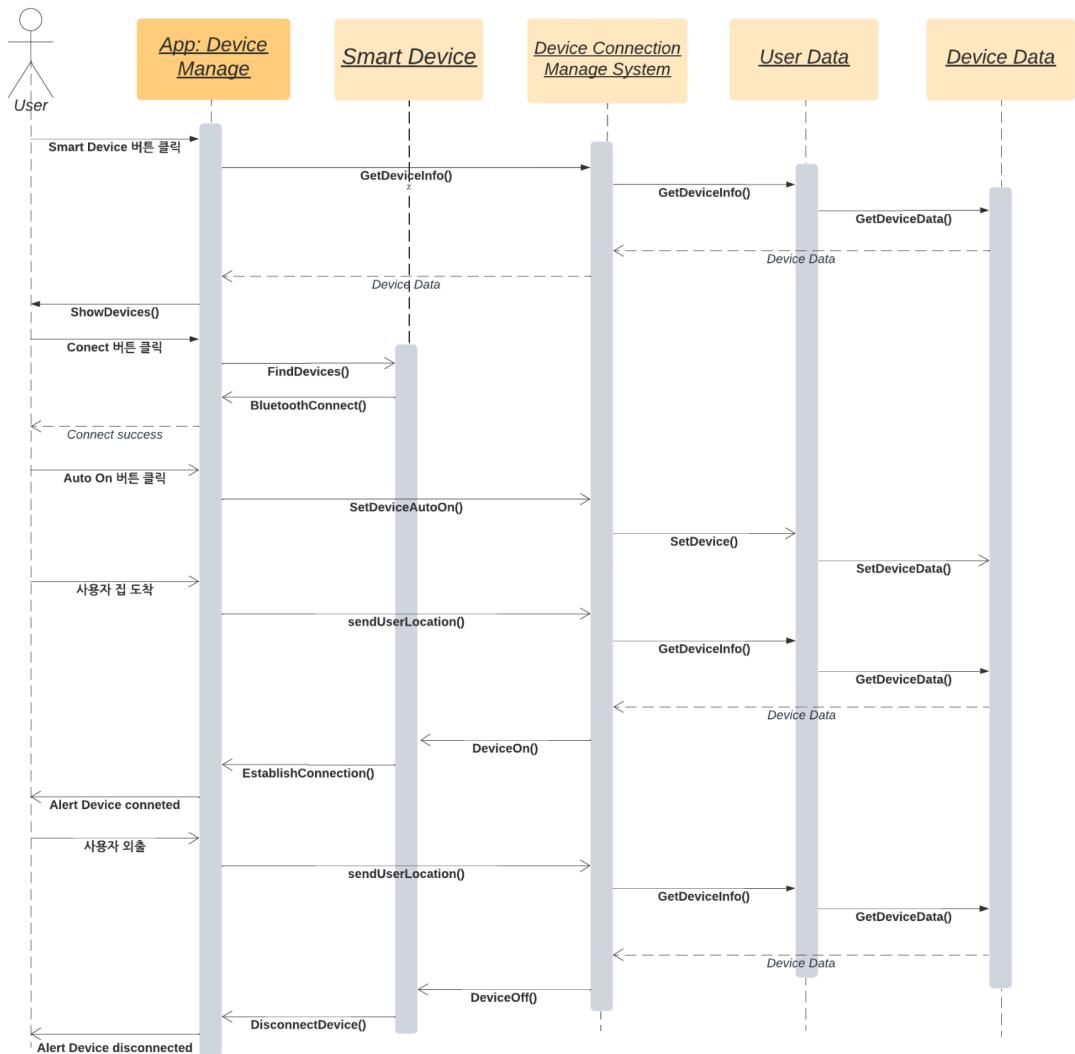
✓ 백엔드

디바이스 연결관리 시스템은 다음과 같은 메서드를 가지고 있다.

- 1) `getDeviceInfo()`
- 2) `AutoOnDevice()`
- 3) `sendUserLocation()`
- 4) `AlertDeviceOn()`

5.6.2 설계 모델

사용자가 스마트 디바이스를 앱에 연동할 때, 컴포넌트 간 상호작용을 Sequence Diagram으로 보여준다.



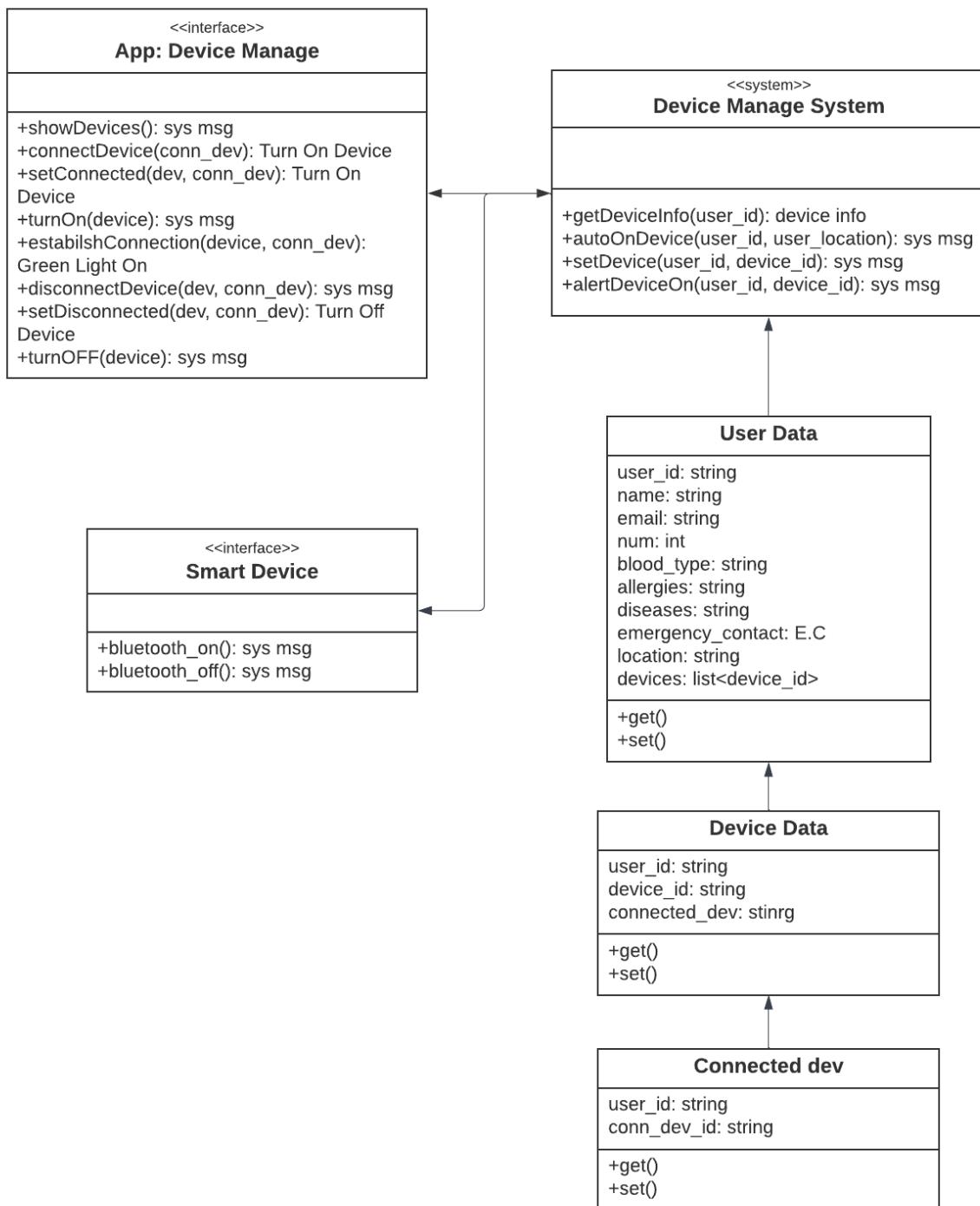
[그림 12] 디바이스 연결 관리 시스템 - 설계 모델

[표 10] 디바이스 연결 관리 시스템 - 설계 모델

서브 시스템	설명
디바이스 정보 확인	사용자가 현재 연결된 디바이스의 정보를 확인하려 하면, <code>GetDeviceInfo()</code> 를 통해 시스템에 정보를 요청한다. 시스템은 <code>GetDeviceInfo()</code> 를 통해, User Data 데이터베이스와 Device Data 데이터베이스로부터 디바이스 정보를 반환한다.
디바이스 연결	사용자가 디바이스를 앱에 연동하고자 한다면, <code>FindDevices()</code> 를 통해 주변에 연결 가능한 디바이스가 있는지 찾는다. 연결 가능한 디바이스가 있다면, <code>BluetoothConnect()</code> 를 통해 디바이스와 앱이 연동된다.

디바이스 자동 켜기 활성화	<p>사용자가 디바이스의 자동 켜기 기능을 활성화하길 원한다면, SetDeviceAutoOn()을 통해 시스템에 디바이스의 활성화를 요청한다.</p> <p>시스템은 SetDevice()를 통해 Device Data 데이터베이스로 디바이스의 활성화 정보를 변경할 것을 요청한다.</p>
디바이스 자동 켜기/ 끄기	<p>사용자가 집에 들어온다면, SendUserLocation()을 통해 시스템은 사용자가 집에 들어왔음을 인지한다. 시스템은 GetDeviceInfo()를 통해 자동 켜기 기능이 활성화된 디바이스의 정보를 얻고, DeviceOn()을 통해 해당 디바이스들을 자동으로 켜준다.</p> <p>사용자가 집에서 나왔다면, SendUserLocation()을 통해 시스템은 사용자가 집에서 나왔음을 인지한다. 시스템은 GetDeviceInfo()를 통해 자동 켜기 기능이 활성화된 디바이스의 정보를 얻고, DeviceOff()을 통해 해당 디바이스들을 자동으로 꺼준다.</p>

5.6.3 인터페이스 명세



[그림 13] 디바이스 연결 관리 시스템 - 인터페이스 명세

5.7 위험 소리 탐지 시스템

해당 컴포넌트는 가내에서 수집된 소리를 분석하여, 사용자에게 어떤 위험 상황이 발생하였는지 알 수 있도록 한다.

5.7.1 객체 클래스 식별

5.7.1.1 attribute

소리 데이터 클래스는 다음과 같은 속성을 가지고 있다.

- 1) **sound** : 실시간으로 전달할 소리를 나타낸다.
- 2) **record_time** : 신호를 탐지할 주기를 나타낸다.

5.7.1.2 Method

✓ 디바이스

디바이스 소리 감지 클래스는 다음과 같은 메소드를 가지고 있다.

- 1) **record_sound()** : 사용자의 소리를 기록하며 실시간으로 서버에 전송한다.
- 2) **send_sound()** : 사용자의 소리를 서버로 전송한다. **record_sound** 내부에서 사용된다.

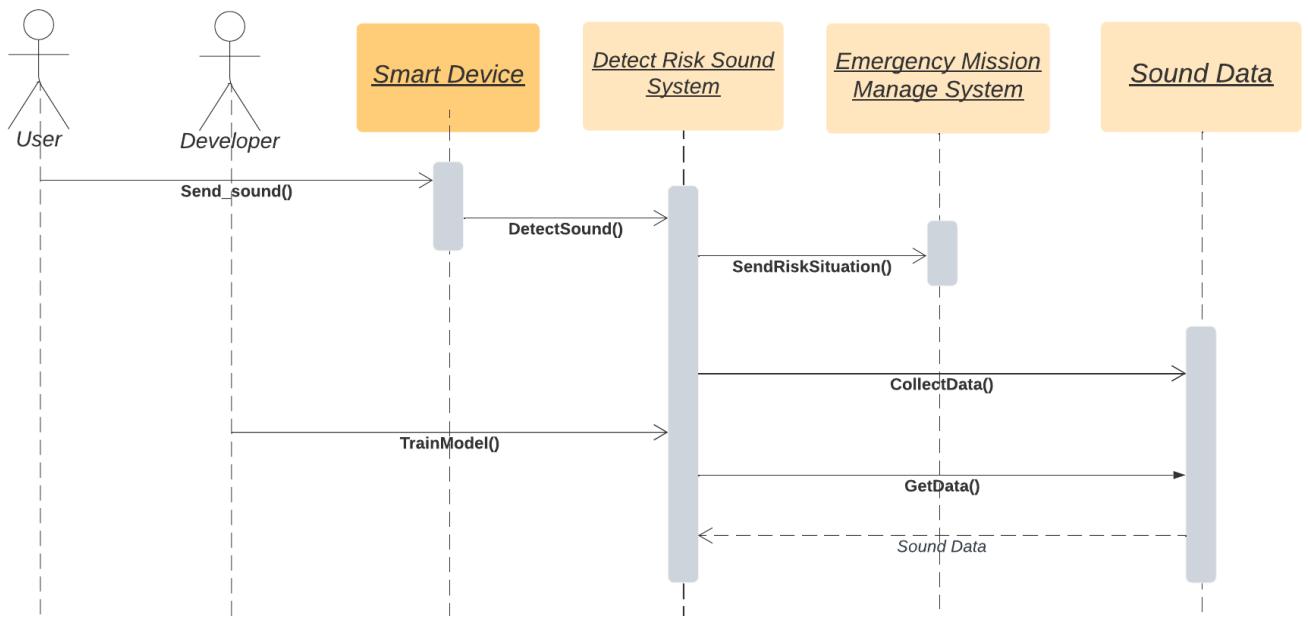
✓ 백엔드

위험 탐지 감지 시스템은 다음과 같은 메서드를 갖는다.

- 1) **detectSound()**
- 2) **trainModel()**
- 3) **sendRiskSituation()**

5.7.2 설계 모델

사용자 가내의 소리를 수집하고 소리를 분석하여 위험 상황인지를 탐지할 때, 컴포넌트 간 상호작용을 Sequence diagram으로 보여준다.

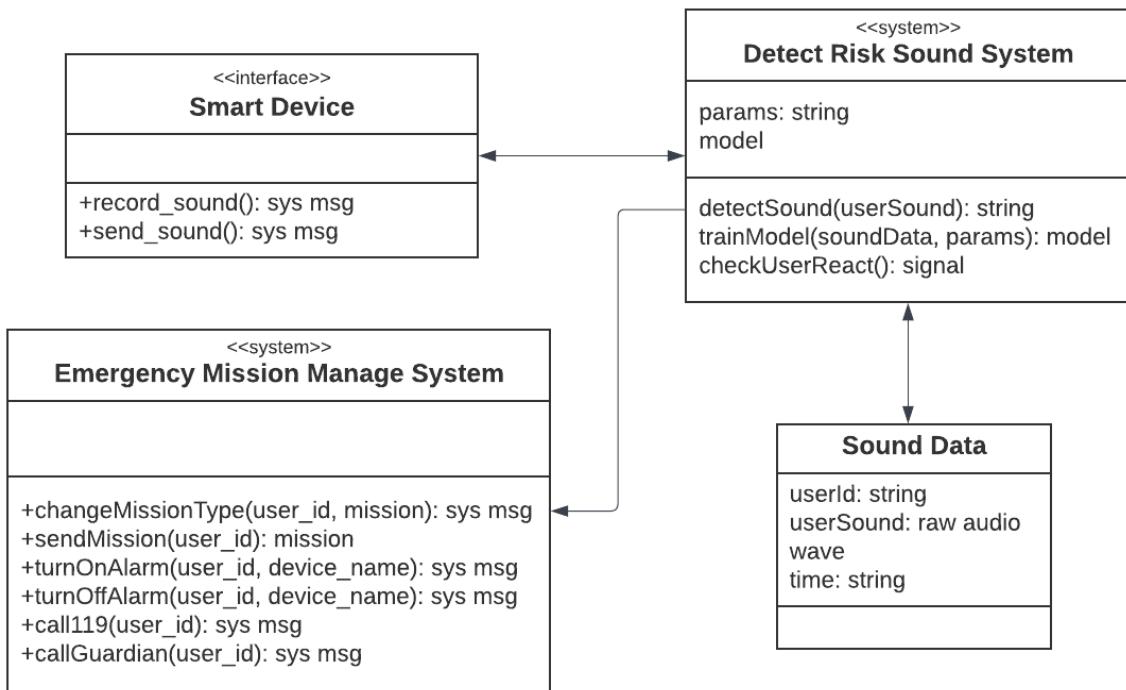


[그림 14] 위험 소리 탐지 시스템 - 설계 모델

[표 11] 위험 소리 탐지 시스템 - 설계 모델

서브 시스템	설명
위험 상황 탐지	사용자로부터 수집한 소리를 시스템으로 보내고, <code>DetectSound()</code> 를 통해 시스템이 소리를 분석할 것을 요청한다. 만약 위험 상황이 탐지 되었다면, 시스템은 <code>SendRiskSituation()</code> 을 통해 응급 상황 미션 관리 시스템에 위험 상황을 알린다. 또한 <code>CollectData()</code> 를 통해 Sound Data 데이터베이스에 소리 정보를 저장한다.
위험 탐지 모델 학습	개발자가 위험 탐지 모델을 학습하려할 때, <code>TrainModel()</code> 을 통해 시스템에서 모델을 학습할 것을 요청한다. 이 때 시스템은 Sound Data 데이터베이스로부터 Sound Data 를 받아 이를 학습 데이터로 사용한다. 개발자는 새로운 데이터로 모델을 학습시킴으로써, 더 좋은 성능의 탐지 모델을 개발할 수 있다.

5.7.3 인터페이스 명세



[그림 15] 위험 소리 탐지 시스템 - 인터페이스 명세

5.8 위험 상황 알림 시스템

해당 컴포넌트는 사용자에게 위험 상황이 생겼음이 확인되었을 때, 이를 사용자의 보호자나 가장 가까운 병원에 알리는 시스템이다.

5.8.1 객체 클래스 식별

5.8.1.1 attribute

사용자 데이터 클래스

- 1) `user_id`: 사용자의 고유 아이디이다.
- 2) `name`: 사용자의 이름이다.
- 3) `num`: 사용자의 번호이다.
- 4) `blood type`: 병원에 알릴 사용자의 혈액형 정보이다.
- 5) `allergies`: 병원에 알릴 사용자의 알러지 정보이다.
- 6) `diseases`: 병원에 알릴 사용자의 지병 정보이다.
- 7) `emergency_contact`: 연락을 취할 보호자의 연락처이다.
- 8) `location`: 사용자의 집 위치 정보이다.

5.8.1.2 Method

✓ 애플리케이션

보호자 및 응급실의 애플리케이션은 다음과 같은 메서드를 갖는다.

1) Alert()

✓ 백엔드

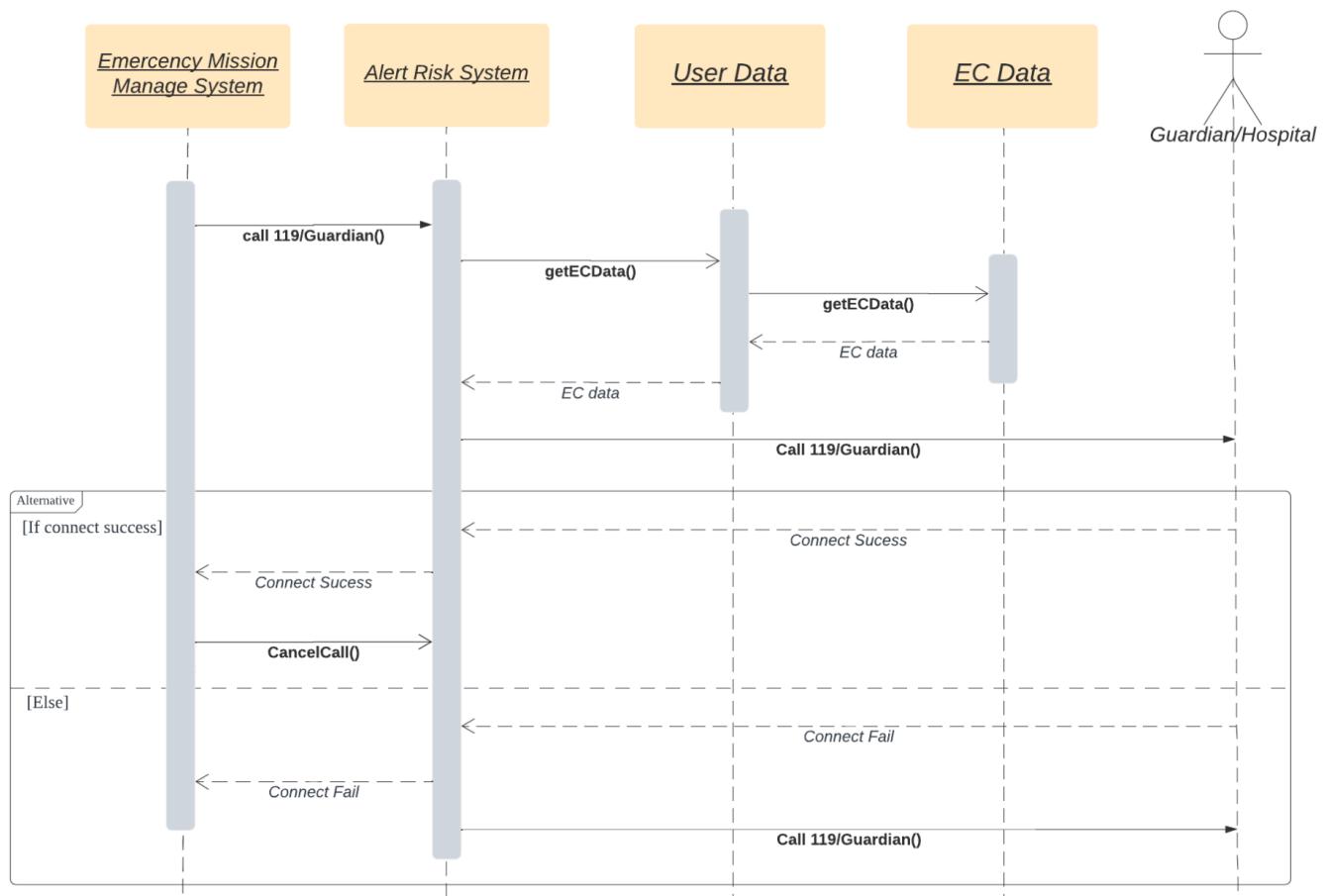
위험 상황 알림 시스템은 다음과 같은 메서드를 갖는다.

1) alertGuardians()

2) alertHospital()

5.8.2 설계 모델

사용자가 위험 상황에 처해있고, 사용자의 반응이 확인되지 않았을 때, 119와 보호자에게 연락을 취한다. 이 때 컴포넌트 간 상호작용을 Sequence diagram으로 보여준다.

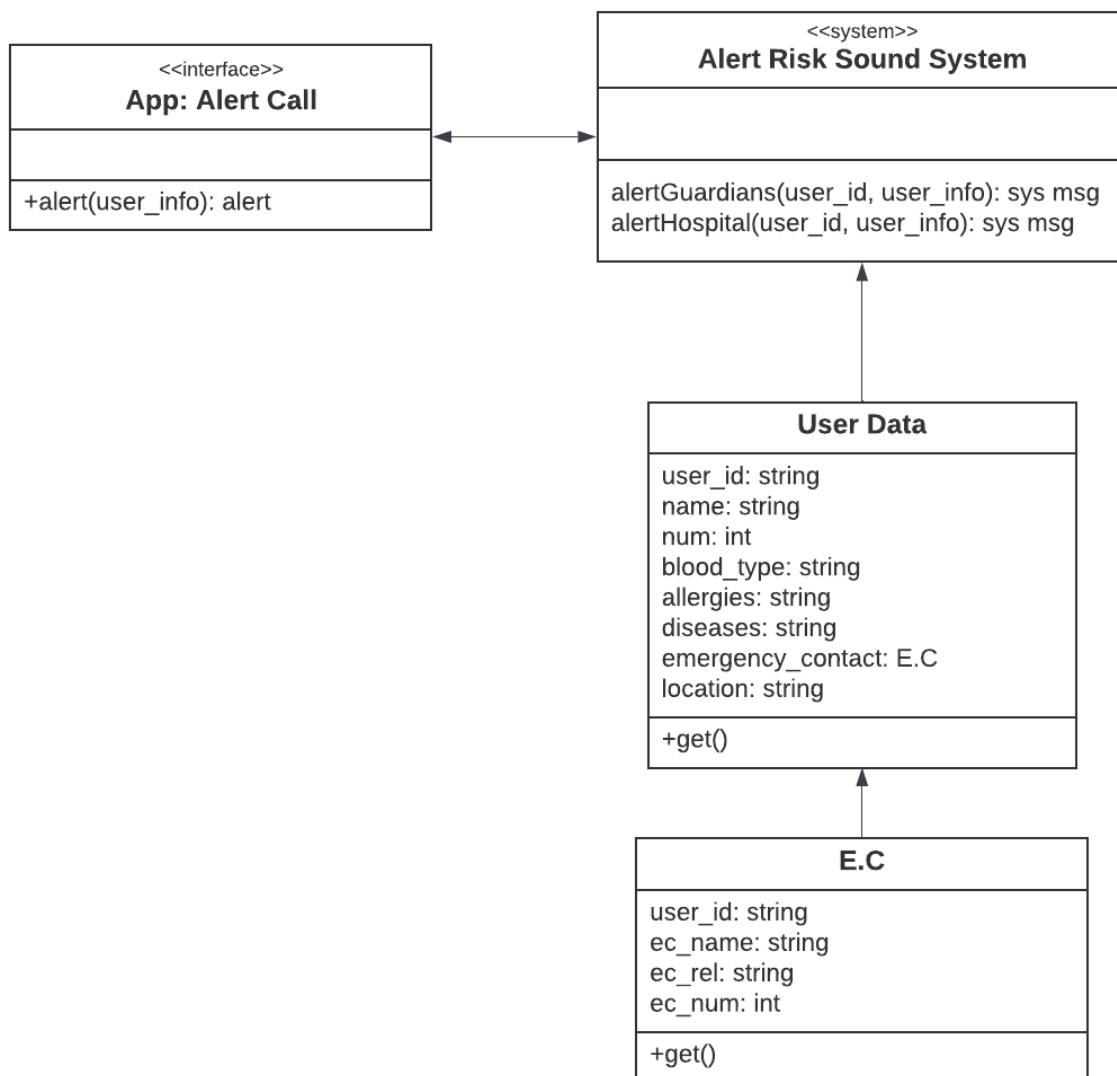


[그림 16] 위험 상황 알림 시스템 - 설계 모델

[표 12] 위험 상황 알림 시스템 - 설계 모델

서브 시스템	설명
119 / 보호자 알림	응급 상황 미션 관리 시스템으로부터 사용자가 위험 상황에 처해있음을 알았을 때, 시스템은 GetECData()를 통해 사용자의 보호자 연락처를 얻는다. 이 연락처로 Call 119/Guardian()을 통해 119와 보호자에게 사용자가 위험 상황에 처해있음을 알린다. 만약, 연락이 되지 않는다면, 일정 주기로 119와 보호자에게 연락을 취한다.

5.8.3 인터페이스 명세



[그림 17] 위험 상황 알림 시스템 - 인터페이스 명세

6. 인터페이스 디자인

6.1 이 장에서는..

이 장에서는 프론트엔드의 인터페이스 디자인을 명세한다.

6.2 어플리케이션 인터페이스

[표 13] 어플리케이션 인터페이스

로그인 화면	사용자 등록 화면	메인 화면	디바이스 연결 화면
응급 정보 수정 화면	미션 수행 화면	미션 성공 화면	미션 실패 화면

6.3 스마트 디바이스 인터페이스

디바이스의 프로토타입은 아래와 같다.



[그림 18] 디바이스 프로토타입

디바이스는 스피커 모양의 형태를 띠고 있으며 서버와의 통신을 위해 내부에 아두이노가 탑재된다.

추후 다양한 모듈과 결합할 수 있으며, 일례로 산소 발생기가 결합될 경우 디바이스의 모양은 다음과 같이 확장될 수 있다.



[그림 19] 결합 시 디바이스의 모양

7. 프로토콜 디자인

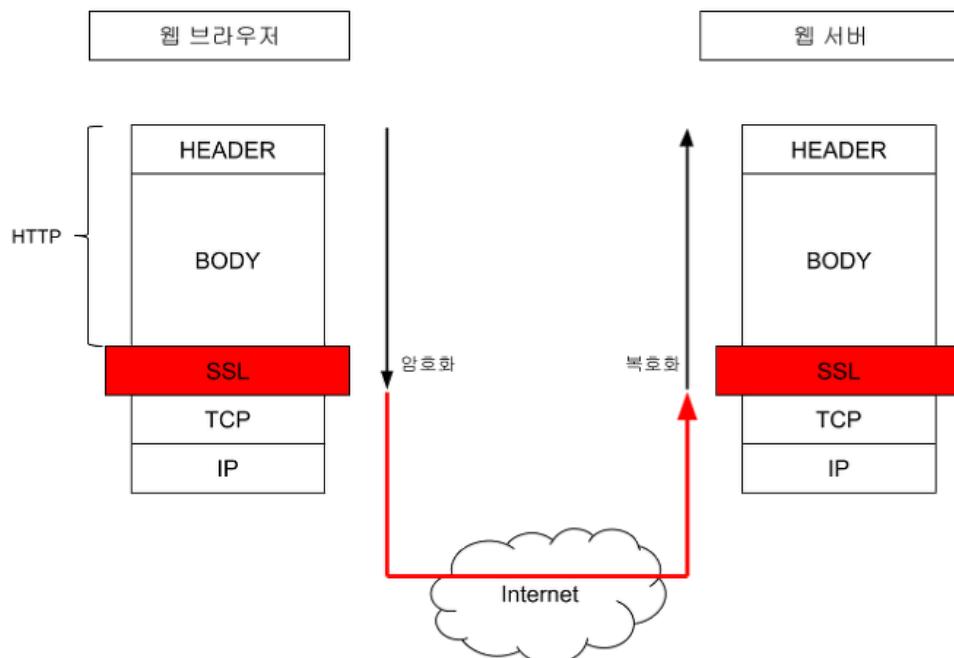
7.1 이장에서는..

7장 프로토콜 디자인에서는, 어플리케이션, 서버, 디바이스와 같은 하위 시스템간의 상호작용에서 사용되는 프로토콜의 구조에 대해 다룬다. 상호작용은 크게 어플리케이션과 디바이스, 어플리케이션과 서버간의 통신으로 이루어지지만 **client-server architecture**를 사용하므로 어플리케이션과 디바이스간의 상호작용은 서버를 거쳐서 이루어진다.

7.2 HyperText Transfer Protocol over TLS (HTTPS)

본 프로젝트에서 서버와의 통신과 인증, 암호화에는 HTTPS 프로토콜을 사용한다. HTTPS는 Hypertext Transfer Protocol over Transport Layer Security의 줄임말로, 일반 텍스트를 이용하는 대신 TLS 프로토콜을 사용해 세션 데이터를 암호화 한다. HTTP를 사용할 경우 사용자의 인증과정에서 사용되는 비밀번호가 일반 텍스트로 전송되어 위험에 노출될 수 있으므로, 사용자 계정 보호를 위해 스마트 노이즈 시스템에서는 HTTPS를 사용한다.

HTTPS 프로토콜은 Hand Shaking 과정에서 서버와 클라이언트 간에 세션키를 교환하며, 동작 과정은 다음과 같다.



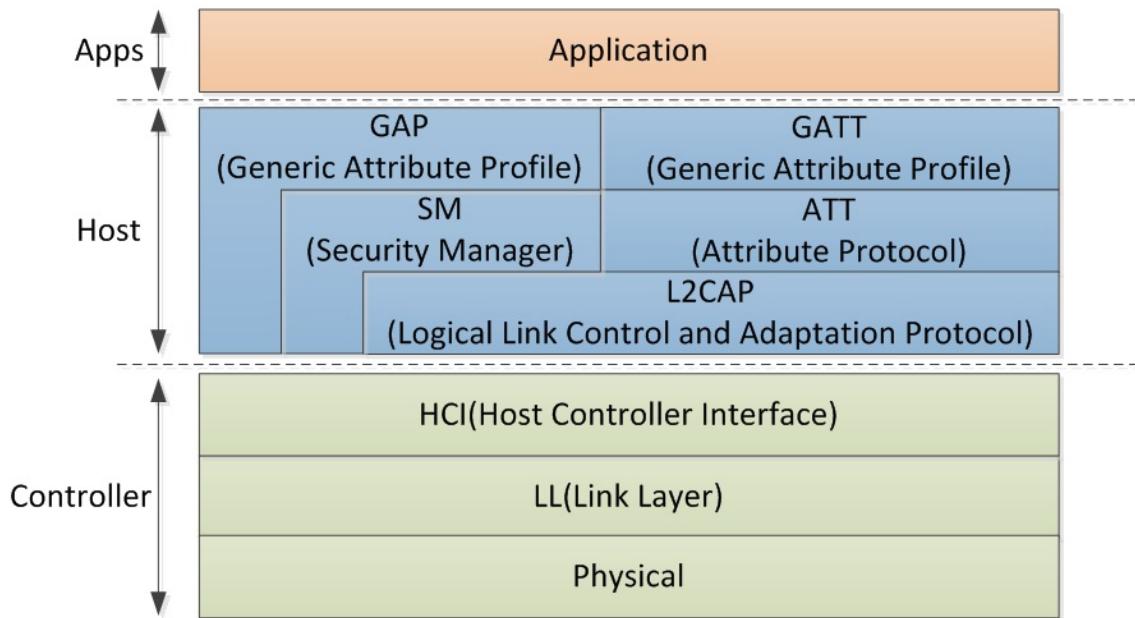
[그림 20] Https 프로토콜, 출처 : <https://sjh836.tistory.com/81>

- Step 1. 클라이언트가 서버로 연결을 시도한다.
- Step 2. 서버가 브라우저에게 공개키(인증서)를 넘겨준다.
- Step 3. 브라우저가 인증서의 유효성을 검사한 후 세션키를 발급한다.
- Step 4. 브라우저가 세션키를 보관하고 서버의 공개키로 세션키를 암호화해 전송한다.
- Step 5. 서버는 개인키로 암호화된 세션키를 복호화해 세션키를 얻는다.
- Step 6. 서버-클라이언트는 동일한 세션키를 공유함으로 데이터 전달 시에 세션키로 암호화/복호화를 진행한다.

7.3 Bluetooth Protocol

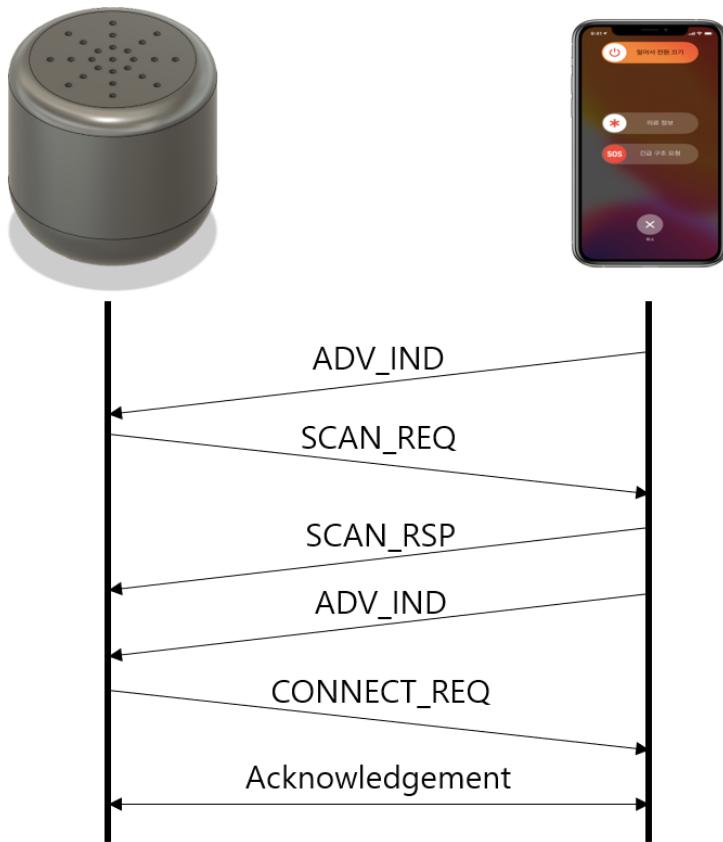
일반적으로 블루투스라 하면, BLE 이전의 스펙을 사용하는 Bluetooth Classic, 블루투스 클래식과 BLE를 모두 지원하는 Bluetooth Smart Ready, 마지막으로 BLE 연결만을 지원하는 Bluetooth Smart가 포함된다.

스마트 노이즈 시스템에서 어플리케이션과 디바이스는 소량의 데이터만을 송/수신하므로 블루투스 스마트를 사용한다.



[그림 21] 블루투스 스택 구조, 출처 : <http://intsain.com/w/?p=322>

Bluetooth 프로토콜은 앱과 디바이스의 연결에 사용되며 통신의 무선 신호를 도식화 하면 아래와 같다.



[그림 22] 블루투스 프로토콜

ADV_IND : 연결 가능한 undirected advertisement에 사용되며 패킷 안에 사용자가 원하는 데이터를 추가할 수 있다.

ADV_IND 전송 후 차례대로 **scan** 요청에 대한 응답을 받고 연결 요청을 받으면 연결이 성사된다.

7.4 인증

7.4.1 Register

- Request

[표 14] Register request

ATTRIBUTE	DETAIL	
Protocol	CHAP	
	Name	사용자의 이름
	User ID	사용자 ID

Request Body	Password	사용자의 비밀번호
	Email	사용자 이메일
	Phone Number	사용자 전화번호
	User_Info	사용자 정보

- Response

[표 15] Register response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 400 (Bad Request)
	HTTPS 404 (Not Found)
	HTTPS 500 (Server Error)
Success Response	Message (“등록 완료”)
Failure Response	Message (“등록 실패”)

7.4.2 로그인

- Request

[표 16] 로그인 Request

ATTRIBUTE	DETAIL	
Protocol	CHAP	
Request Body	User ID	사용자 ID
	Password	사용자의 비밀번호

- Response

[표 17] 로그인 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 400 (Bad Request)
	HTTPS 404 (Not Found)
	HTTPS 500 (Server Error)
Success Response	메인 화면으로 이동
Failure Response	Message (“로그인 실패”)

7.5 사용자 프로필

7.5.1 사용자 프로필 설정

- Request

[표 18] 사용자 프로필 설정 Request - 1

ATTRIBUTE	DETAIL	
Method	Set Profile	
URI	profile/user_info	
Parameter	Login Info	사용자 로그인 정보
	Name	사용자의 비밀번호
	Email	사용자 이메일
	Phone Number	사용자 전화번호
	Blood Type	사용자 혈액형
	Allergies	사용자 알레르기
	Diseases	사용자 질병
	Emergency Contact	사용자 보호자 정보

	Info	
--	------	--

[표 19] 사용자 프로필 설정 Request - 2

ATTRIBUTE	DETAIL	
Method	Edit Profile	
URI	profile/user_info	
Parameter	Blood Type	사용자 혈액형
	Allergies	사용자 알레르기
	Diseases	사용자 질병
	Emergency Contact Info	사용자 보호자 정보

- Response

[표 20] 사용자 프로필 설정 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 400 (Bad Request)
	HTTPS 403 (Forbidden)
	HTTPS 404 (Not Found)
	HTTPS 500 (Server Error)
Success Response	Message (“정보 등록 성공”)
Failure Response	Message (“정보 등록 실패”)

7.5.2 사용자 프로필 가져오기

- Request

[표 21] 사용자 프로필 가져오기 Request

ATTRIBUTE	DETAIL	
Method	Get Profile	
URI	profile/user_info	
Parameter	Name	사용자의 비밀번호
	Blood Type	사용자 혈액형
	Allergies	사용자 알레르기
	Diseases	사용자 질병
	Emergency Contact Info	사용자 보호자 정보

- Response

[표 22] 사용자 프로필 가져오기 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 400 (Bad Request)
	HTTPS 404 (Not Found)
	HTTPS 500 (Server Error)
Success Response	Message (“정보 가져오기 성공”)
Failure Response	Message (“정보 가져오기 실패”)

7.6 데이터 가져오기

7.6.1 데이터 가져오기

- Request

[표 23] 데이터 가져오기 Request

ATTRIBUTE	DETAIL

Method	GET Data	
URI	data/sound_data	
Parameter	Number per Day	하루에 호흡곤란이 감지된 횟수
	Total Number	호흡곤란이 감지된 총 횟수
	Average Number	하루에 평균적으로 호흡곤란이 감지되는 횟수

- Response

[표 24] 데이터 가져오기 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 404 (Not Found)
	HTTPS 500 (Server Error)
Success Response	Message (“정보 가져오기 성공”)
	그래프 보여주기
Failure Response	Message (“정보 가져오기 실패”)

7.6.2 데이터 분석하기

- Request

[표 25] 데이터 분석하기 Request

ATTRIBUTE	DETAIL	
Method	Analyze Data	
URI	data/sound_data	
	Number per Day	하루에 호흡곤란이 감지된 횟수

Parameter	Total Number	호흡곤란이 감지된 총 횟수
	Average Number	하루에 평균적으로 호흡곤란이 감지되는 횟수

- Response

[표 26] 데이터 분석하기 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 404 (Not Found)
	HTTPS 500 (Server Error)
Success Response	사용자 상태 표시
Failure Response	Message ("정보 가져오기 실패")

7.7 응급 상황 미션

7.7.1 응급 상황 미션 설정

- Request

[표 27] 응급 상황 미션 설정 Request

ATTRIBUTE	DETAIL	
Method	Set Mission	
URI	/emergency	
Parameter	Mission	사용자가 수행할 미션

- Response

[표 28] 응급 상황 미션 설정 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 500 (Server Error)
Success Response	Message (“선택 완료”)
	선택된 미션 강조 표시
Failure Response	Message (“서버 에러”)

7.7.2 응급 상황 미션 성공

- Request

[표 29] 응급 상황 미션 성공 Request

ATTRIBUTE	DETAIL
Method	Success Mission
URI	/emergency

- Response

[표 30] 응급 상황 미션 성공 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 500 (Server Error)
Success Response	미션 성공 팝업 창을 띠운다.
Failure Response	Message (“서버 에러”)

7.7.3 응급 상황 미션 실패

- Request

[표 31] 응급 상황 미션 실패 Request

ATTRIBUTE	DETAIL
Method	Fail Mission
URI	/emergency

- Response

[표 32] 응급 상황 미션 실패 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 500 (Server Error)
Success Response	미션 실패 팝업 창을 띠운다.
	119 호출 및 보호자 연락
Failure Response	Message (“서버 에러”)

7.7.4 119 호출 취소

- Request

[표 33] 119 호출 취소 Request

ATTRIBUTE	DETAIL	
Method	Cancel Call	
URI	/emergency	
Parameter	User Info	사용자 정보

- Response

[표 34] 119 호출 취소 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)

Failure Code	HTTPS 400 (Bad Request)
	HTTPS 500 (Server Error)
Success Response	119 호출 취소
	미션 실패 팝업 창 종료
Failure Response	Message (“취소 실패”)

7.8 디바이스 연결

7.8.1 디바이스 검색

- Request

[표 35] 디바이스 검색 Request

ATTRIBUTE	DETAIL	
Method	Get Devices	
URI	/connection	
Parameter	Device Name	연결 가능한 디바이스 이름
	Device ID	연결 가능한 디바이스 ID

- Response

[표 36] 디바이스 검색 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 404 (Not Found)
Success Response	연결 가능한 디바이스 리스트 띄우기
Failure Response	Message (“연결 가능한 디바이스 없음”)

7.8.2 디바이스 연결

- Request

[표 37] 디바이스 연결 Request

ATTRIBUTE	DETAIL	
Method	Connect Device	
URI	/connection	
Parameter	Device Name	연결 가능한 디바이스 이름
	Device ID	연결 가능한 디바이스 ID
	SmartPhone	사용자 스마트폰

- Response

[표 38] 디바이스 연결 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 400 (Bad Request)
	HTTPS 404 (Not Found)
	HTTPS 500 (Server Error)
Success Response	Message (“연결 성공”)
	Smart Device 섹션에 초록 불 띄우기
	Smart Device 기능 켜기
Failure Response	Message (“연결 실패”)

7.8.3 디바이스 연결 끊기

- Request

[표 39] 디바이스 연결 Request

ATTRIBUTE	DETAIL	
Method	Disconnect Device	
URI	/connection	
Parameter	Device Name	연결 가능한 디바이스 이름
	Device ID	연결 가능한 디바이스 ID
	SmartPhone	사용자 스마트폰

- Response

[표 40] 디바이스 연결 Response

ATTRIBUTE	DETAIL
Success Code	HTTPS 200 (OK)
Failure Code	HTTPS 400 (Bad Request)
	HTTPS 404 (Not Found)
	HTTPS 500 (Server Error)
Success Response	Message (“연결 해제 성공”)
	Smart Device 셕션에 빨간 불 띄우기
	Smart Device 기능 끄기
Failure Response	Message (“연결 해제 실패”)

8. 데이터베이스 디자인

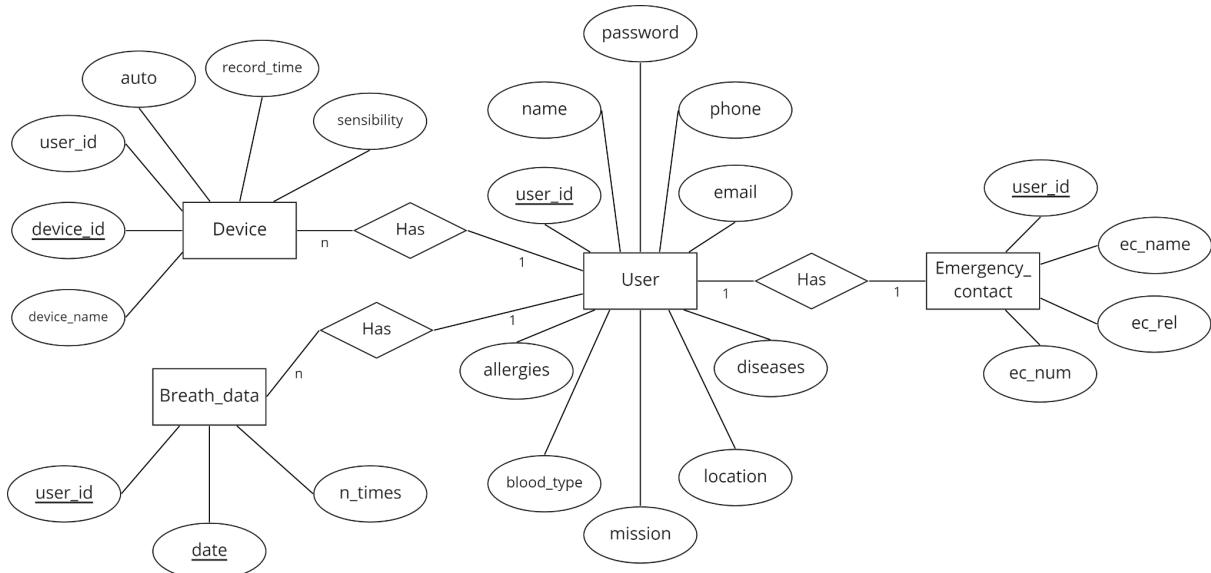
8.1 이 장에서는..

이 장에서는 시스템 데이터 구조와 이것이 어떻게 데이터베이스에 반영될지를 설명한다.

이는 ER-Diagram (Entity Relationship diagram), Relational Schema와 SQL DDL (Data Description Language)을 통해 설명될 것이다.

8.2 ER Diagram

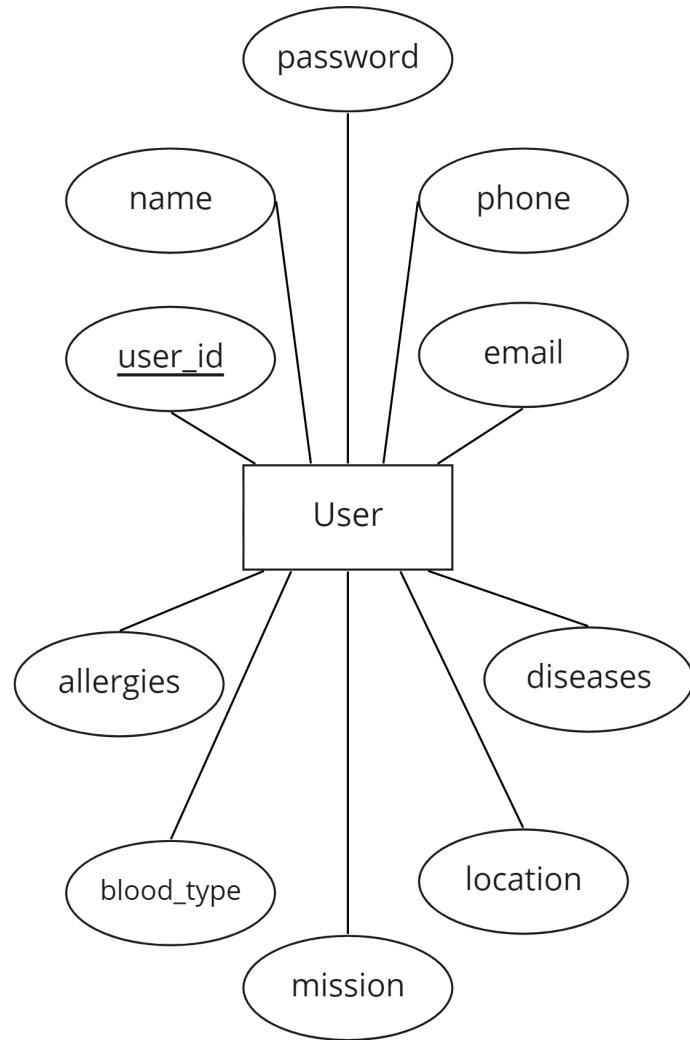
시스템은 4개의 개체: User, Device, Breath_data, Emergency_contact 들로 구성된다. 다이어그램에서 개체(Entity)는 사각형, 관계(Relationship)은 마름모, 속성(Attribute)은 타원으로 표현된다. 기본키(Primary key)는 밑줄로 표시된다.



[그림 23] ER Diagram

8.2.1 요소들

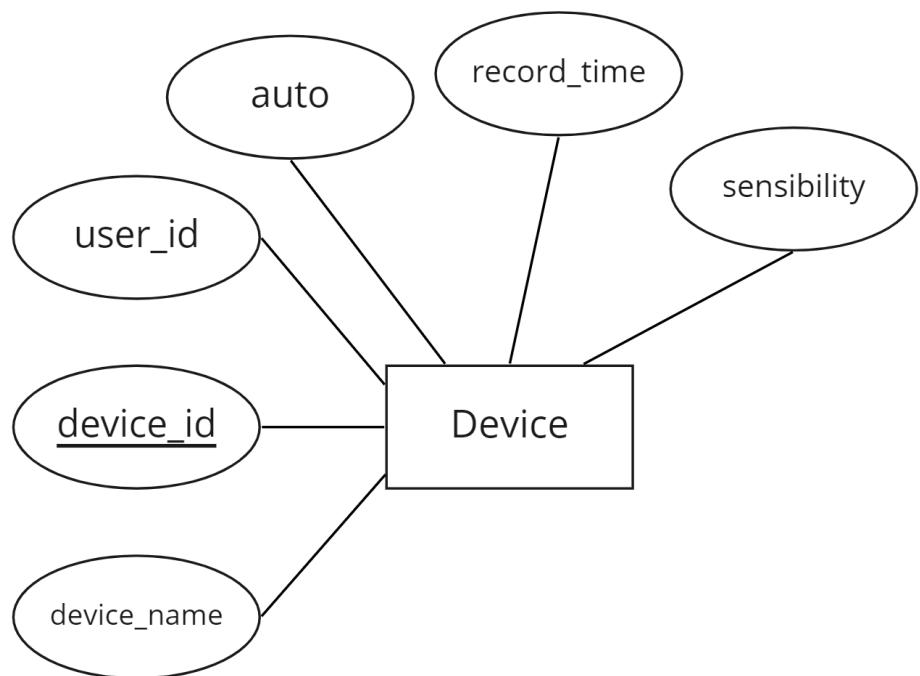
8.2.1.1 User



[그림 24] ER Diagram, 개체, User

User 개체는 Smart Noise 서비스를 이용하는 사용자를 나타낸다. 사용자의 계정 정보와 개인 정보가 저장될 것이다. `user_id`, `name`, `password`, `phone`, `email`, `allergies`, `blood_type`, `mission`, `location`, `diseases`를 포함한다. 기본키는 `user_id`이다.

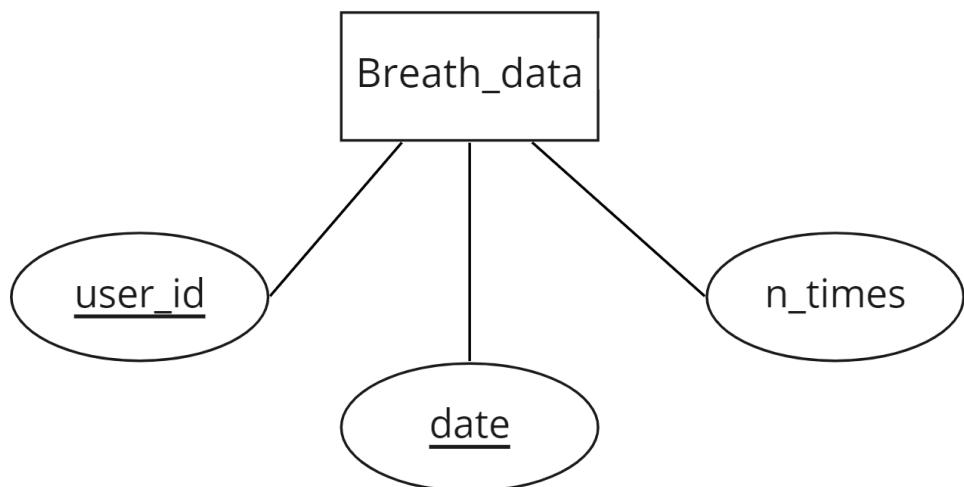
8.2.1.2 Device



[그림 25] ER Diagram, 개체, Device

Device 개체는 사용자가 등록한 디바이스의 정보와 설정을 나타낸다. user_id, device_id, device_name, auto, record_time, sensibility를 포함한다. 기본키는 device_id이다.

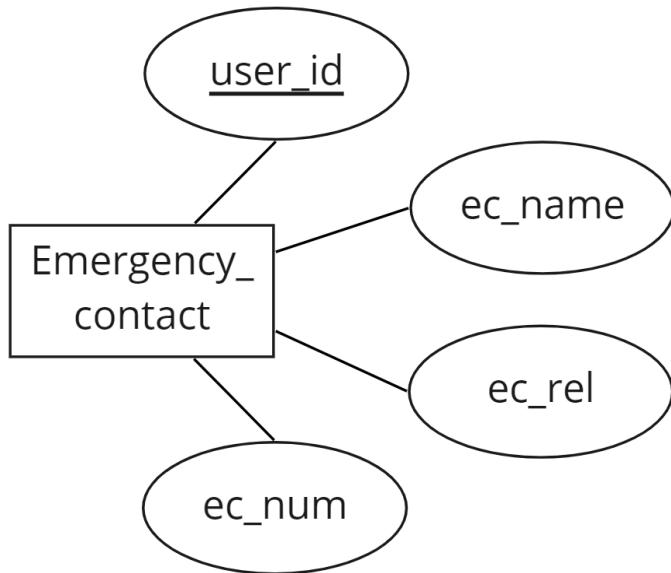
8.2.1.3 Breath_data



[그림 26] ER Diagram, 개체, Breath_data

`Breath_data` 개체는 호흡 곤란 감지 정보를 나타낸다. `user_id`, `date`, `n_times`을 포함한다. `user_id`와 `date`를 사용해 복합키를 구성한다. `n_times`은 해당 날짜에 호흡 곤란이 감지된 횟수이다.

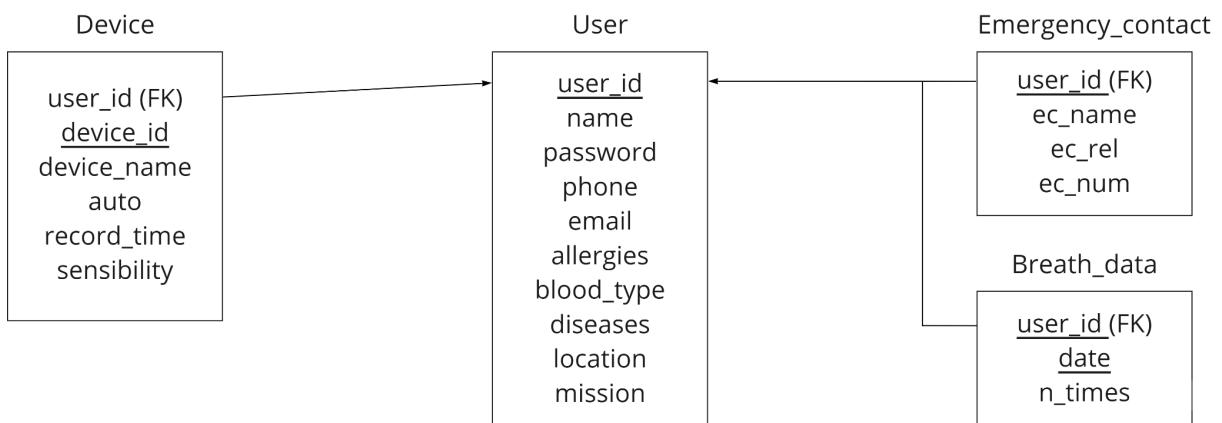
8.2.1.4 Emergency_contact



[그림 27] ER Diagram, 개체, `Emergency_contact`

`Emergency_contact` 개체는 위급 상황 시 알림을 전송할 보호자의 정보를 나타낸다. `user_id`, `ec_name`, `ec_rel`, `ec_num`을 포함한다. `ec_name`, `ec_rel`, `ec_num`은 각각 보호자의 이름, 사용자와의 관계, 전화번호이다. 기본키는 `user_id`이다.

8.3 관계 계획



[그림 28] 관계 계획

8.4 SQL DDL

8.4.1 User

```
Create TABLE User
(
    user_id VARCHAR NOT NULL
    name VARCHAR NOT NULL
    password VARCHAR NOT NULL
    phone VARCHAR NOT NULL
    email VARCHAR NOT NULL
    allergies VARCHAR NOT NULL
    blood_type VARCHAR NOT NULL
    diseases VARCHAR NOT NULL
    location VARCHAR NOT NULL
    mission VARCHAR NOT NULL
    PRIMARY KEY (user_id)
);
```

8.4.2 Device

```
Create TABLE Device
(
    user_id VARCHAR NOT NULL
    device_id VARCHAR NOT NULL
    device_name VARCHAR NOT NULL
    auto BOOLEAN NOT NULL
    record_time VARCHAR NOT NULL
    sensibility VARCHAR NOT NULL
    PRIMARY KEY (device_id)
    FOREIGN KEY (user_id) REFERENCES User(user_id)
);
```

8.4.3 Breath_data

```
Create TABLE User
(
    user_id VARCHAR NOT NULL
    date DATE NOT NULL
    n_times INT NOT NULL
    FOREIGN KEY (user_id) REFERENCES User(user_id)
);
```

8.4.4 Emergency_contact

```
Create TABLE User
(
    user_id VARCHAR NOT NULL
    ec_name VARCHAR NOT NULL
    ec_rel VARCHAR NOT NULL
    ec_num VARCHAR NOT NULL
    FOREIGN KEY (user_id) REFERENCES User(user_id)
);
```

9. 테스트 계획

9.1 이 장에서는..

이 장에서는 테스트 계획에 대해 다룬다. 테스팅은 크게 개발 테스팅, 배포 테스팅, 그리고 사용자 테스팅으로 나누어진다. 테스트를 통해 치명적인 오류들을 조기에 발견함으로써 소프트웨어의 안전성을 높일 수 있기 때문에 이러한 테스팅 과정은 매우 중요하다.

9.2 테스트 정책

9.2.1 개발 테스팅

개발 테스팅은 소프트웨어 개발 단계에서 개발자들에 의해 진행되는 테스트들이다. 개발 테스팅을 통해 프로그램의 품질과 예측성, 투명성 등의 개선 효과를 기대할 수 있다. 이

테스팅은 정적 코드 분석, 데이터 흐름 분석, 코드 리뷰, 유닛 테스팅의 과정으로 이루어진다. 이러한 테스팅 과정을 통해 소프트웨어가 성능, 신뢰성, 보안의 측면에서 원활하게 작동할 수 있도록 하여야 한다.

9.2.1.1 성능

지속적인 작동이 요구되는 특성상 높은 성능보다는 적은 자원을 효율적으로 사용하는 것이 중요하다. 비교적 적은 전력을 소모하면서 작동하여 배터리 소모량을 최소화하는 것이 좋기 때문이다. 또한, 소리 데이터와 분석 데이터를 주고받는 것이 중요한 만큼 네트워크 통신은 빠르게 이루어질 것이 요구된다. 통신이 빠르게 이루어져야 사용자가 분석 결과를 빠르게 확인할 수 있으며, 응급 상황 전달도 원활하게 이루어질 수 있다. 때문에 테스트를 통해 전력 소모량과 데이터 송수신 속도를 체크하고 개선해야 한다.

9.2.1.2 신뢰성

소리 데이터를 원활하게 수집하고 분석하기 위해서는 디바이스와 통신 과정이 안정적이고 지속적으로 작동할 필요가 있다. 그렇지 못하다면 사용자는 신뢰할 만한 분석 결과를 얻지 못해 서비스를 사용하는 의미가 퇴색된다. 또한 응급 상황 발생을 감지하고 이에 대응하여 보호자에게 알림을 보내기 위해서도 지속적, 안정적인 작동과 끊김 없는 네트워크는 필수적이다. 따라서 이러한 부분들은 개발 단계에서부터 지속적으로 테스트를 통해 문제를 감지하고 수정할 필요가 있다.

9.2.1.3 보안

스마트 노이즈는 서비스 특성상 민감한 정보들을 다루기 때문에 보안이 매우 중요하다. 사용자의 계정은 민감한 개인 정보들을 담고 있으며 서비스는 지속적으로 집 안에서 발생하는 소리를 수집·분석한다. 따라서 이러한 정보들을 통신 과정과 저장 과정에서 안전하게 암호화하고 전달할 수 있어야 한다.

9.2.2 배포 테스팅

배포 테스팅은 특정 배포 버전을 테스트하는 것이다. 개발 단계에서 진행하는 테스트로 모든 문제점을 찾아낼 수는 없다. 만약 배포 후에 치명적인 오류가 발생한다면 이는 심각한 문제가 될 수 있기 때문에 이를 미연에 방지할 필요가 있으며, 이것이 배포 테스팅의 목적이다. 배포 테스팅을 통해 소프트웨어의 배포 버전에서 발생할 수 있는 문제들을 조기에 발견, 수정 할 수 있으며, 이러한 과정들을 통해 사용자들에게 소프트웨어를 배포할 준비가 완료되었음을 확인할 수 있다.

9.2.3 사용자 테스팅

사용자 테스팅을 위해서는 현실적으로 발생 가능한 시나리오를 가정해야 한다. 스마트 노이즈는 특성상 집 안 환경에서 테스트를 진행할 필요가 있다. 디바이스와 애플리케이션을 세팅하고 디바이스 연동, 소리 데이터 수집과 분석이 정상적으로 이루어지고 있는지 테스트한다. 또한 응급 상황을 가정하여 응급 상황에서의 미션 제공과 보호자 연락 프로세스의 작동을 테스트할 필요가 있다.

9.2.4 테스팅 케이스

테스팅 케이스는 기능성, 퍼포먼스, 보안을 기본으로 고려하여 설정되어야 한다. 최대한 다양한 상황을 고려하여 테스트를 진행해야 예상치 못한 오류가 발생하는 것을 방지할 수 있다.

10. 개발 계획

10.1 이 장에서는..

이 장에서는 Smart Noise를 개발하기 위한 계획을 명시한다. 개발을 위해 필요한 프론트엔드, 백엔드 환경을 제시하고, 개발 제약 조건과 개발의 가정 및 의존성에 대한 내용을 제공한다.

10.2 프론트엔드 환경

10.2.1 Adobe Photoshop (UI/UX 디자인)



[그림 29] Adobe Photoshop 로고

포토샵은 이미지를 편집하기 위해 사용하는 편집툴이다. 스마트노이즈의 UI 와 UX를 디자인하기 위해 사용하고 있으며 향후 로고 디자인을 위해 사용할 예정이다.

10.2.2 Android Studio (Application)



[그림 30] Android Studio 로고

안드로이드 스튜디오는 안드로이드 전용 앱을 만들기 위해 사용되는 안드로이드 오픈 소스 프로젝트의 일부이다. 스마트 노이즈 앱을 개발하기 위해 안드로이드 스튜디오를 사용하고 있으며 안드로이드 뿐만 아니라 iOS에서도 사용 가능하도록 Flutter이라는 플러그인을 사용하고 있다.

10.3 백엔드 환경

10.3.1 Github (오픈 소스)



[그림 31] GitHub 로고

깃허브는 분산 버전 관리툴인 깃 저장소 호스팅을 지원하는 웹 서비스이다.

스마트노이즈 시스템은 크캐 앱 프론트 개발, 아두이노 개발, 딥러닝 알고리즘 개발 등이
독자적으로 이루어짐과 동시에 각 **component**들의 통합과 상호작용이 중요하므로 개발자들
사이의 즉각적인 피드백과 버전 관리, 빠른 통합이 가능한 깃허브를 사용한다.

10.3.2 MySQL Database (DBMS)



[그림 32] MySQL 로고

MySQL은 오픈 소스의 관계형 데이터베이스 관리 시스템이다. 스마트 노이즈 서비스에서는 사용자들의 계정이나 개인, 디바이스의 정보 등의 데이터를 저장 및 관리하기 위해 MySQL을 사용한다. 서버에서 사용자들의 데이터를 저장 및 관리함으로써 로그인, 동기화, 흐름 데이터 표시 등의 서비스를 제공할 수 있다.

10.3.3 Node.js (Server)



[그림 33] Node.js 로고

Node.js는 크롬 V8 엔진 위에서 실행되는 오픈 소스, 크로스 플랫폼, 백엔드 자바스크립트 런타임 환경이다. Node.js를 이용해 백엔드 서버를 구축하여 애플리케이션과 데이터베이스 간의 연동을 구현할 수 있다.

10.3.4 AWS EC2 (Server)



[그림 34] AWS EC2 로고

Amazon Elastic Compute Cloud(Amazon EC2)는 안전하고 크기 조정이 가능한 컴퓨팅 용량을 클라우드에서 제공하는 웹 서비스이다. EC2를 이용하여 서버 인스턴스를 생성함으로써 별도로 물리적 서버를 구축할 필요 없이 서비스를 제공할 수 있다. 또한 Auto Scaling 서비스를 통해 트래픽의 양에 따라 인스턴스를 증가/감소 시켜 트래픽의 변화에 유연하게 대응할 수 있다.

10.4 제약

스마트 노이즈는 이 문서에 기술된 내용을 기반으로 구현되며, 세부 사항은 사용자의 요구사항에 따라 변경될 수 있다. 다음은 **smart noise**의 제약사항이다.

- 한 기기는 한 사용자의 핸드폰에만 연동되며 최초 사용시 접근 권한을 허가받아야 한다.
- 디바이스는 최대 8시간동안 연속적으로 사용되므로 발열관리가 필수적이다.
- 가능한 성능을 개선하는 방향으로 개발해야 한다.
- 소스코드를 작성함에 있어, 차후의 **evolution**이나 유지보수를 위해 주석을 충분히 작성해야 한다.
- 불필요한 리소스의 소모를 막기 위해 가능한 최적화에 힘써야 한다.

10.5 가정 및 의존성

스마트 노이즈는 디바이스 최소 사양 2GB, CPU 속도 16MHz 이상, 프로세서 ATmega329p가 필요하며 앱의 경우 안드로이드 6.0 이상, iOS 9.0 이상이 필요하다. 웹 서버와 데이터 관리는 AWS와 MySQL에 의존하며 Amazon Cloud Server 환경은 32GB의 RAM, 1TB 이상의 SSD, 512개의 CUDA Cores, GPU Memory 2GB 이상을 가정한다.

11. 추가 정보

11.1 시장 양산 방안

- 평촌, 일산 등 1차 신도시 + 서울시 신속 통합 기획 단지 연계 예정, 인근 병원들과 제휴로 서비스 고도화, 양산 예정

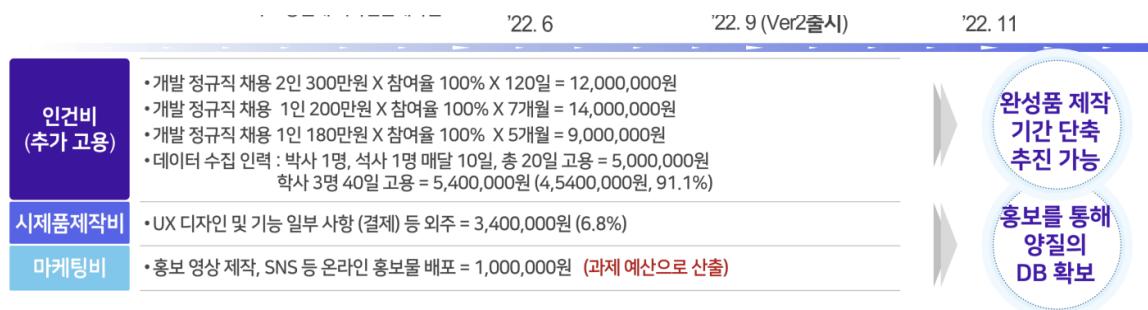
- ① 평촌 및 일산, 산본 등 1차 신도시로 확대: 인근 분당제생병원, CHA병원, 분당 서울대 병원과 협업, 6개월 동안 긴급 무호흡 환자 방문자수 추이 공유 예정
- ② 서울시 신속 통합 기획 개발 단지, 시공사 접촉: 1차 신도시 검증 지표 중심으로 LH, SH 포함 2군 건설사, 중심으로 1군 건설사까지 제휴 제안 예정

11.2 Traction / 매출 전략 및 비전

① 비즈니스 모델: 월 5천원 구독 방식, 연 50,000원 정도로 수익을 낼 예정으로, 서비스 가수요자 251명 대상으로 했을 때 재구매 여부 희망자가 94%였다. (자료 강의원 의학 전문대학원)

② 1인 가구 및 군부대까지: 스마트 흄 서비스 취약한 1인 가구 외, 지식 노동자 중심으로 육체 노동자까지, 군부대 및 소외 지역까지 서비스를 확장할 계획이다.

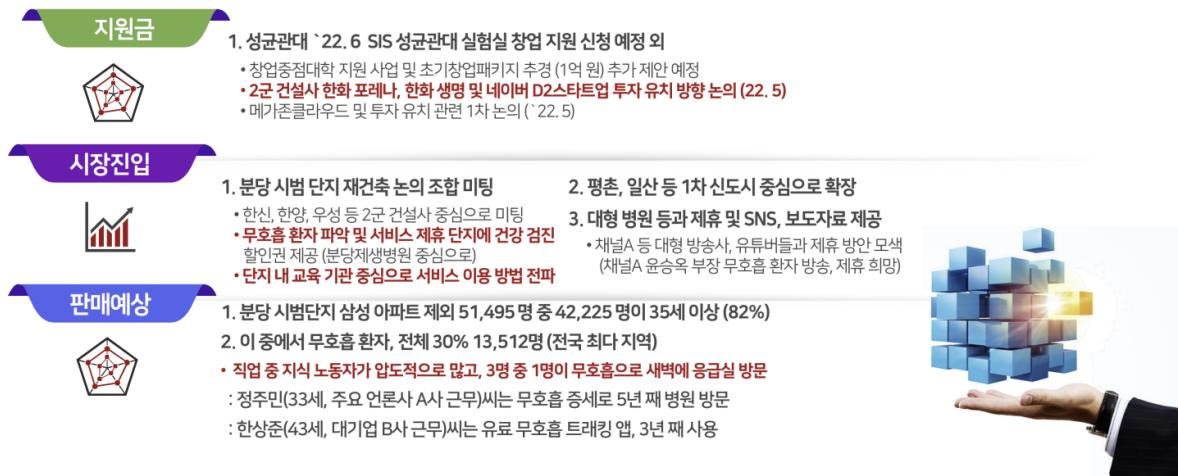
③ 국내 1인 이상 가구: 국내 22년 기준 1인 가구가 730만으로, 취약 계층은 30대 중반 이상으로 본다. 분당 및 재건축 지역 중심으로 주택, 빌라, 오피스텔 등으로 확장해 나갈 계획이다.



[그림 35] 매출 전략 장표

11.3 자금소요 및 성과창출 전략

22년 11월부터 매출 발생 예상하며, 분당 시범 단지 중심 과금 수요 가구 발생 추정한다.



[그림 36] 자금 소요 및 성과창출 전략 장표

11.4 문서 히스토리

[표 41] 문서 히스토리

날짜	내용	작성자
22.05.03	목차 정리	고은서
22.05.05	5.3, 5.4, 5.5, 5.6, 10.2 장 추가	이승민
22.05.05	2장 추가	고남욱
22.05.05	5.3, 5.5, 5.7, 7.1, 7.2, 7.3, 10.3 장 추가	허한울
22.05.06	7.4, 7.5, 7.6, 7.7, 7.8 장 추가	이승민

22.05.06	3.1, 3.2, 3.3, 5.7, 5.8장 추가	고은서
22.05.07	8.1, 8.2, 8.3, 8.4 장 추가	손석규
22.05.07	4.1, 4.2 장 추가	고은서
22.05.08	9.1, 9.2 장 추가	손석규
22.05.09	독자 재정비 및 초안 전체 피드백	고은서
22.05.10	5.3, 5.4, 5.5, 5.6 장 수정	이승민
22.05.11	4.2, 5.3, 5.5, 5.7 장 수정	허한울
22.05.11	1.1, 1.2, 1.3, 2.1, 2.2, 2.3 수정	고은서
22.05.12	8.2, 10.3 장 수정	손석규
22.05.12	2.3, 2.4 장 추가, 문서 포맷 수정	허한울
22.05.12	10.4, 10.5 장 추가	고은서
22.05.13	7.2 장 수정	허한울
22.05.13	5장 수정 및 표 추가	고은서
22.05.14	캡션 추가	손석규
22.05.14	문서 재검토 및 수정	고은서, 허한울
22.05.15	문서 재검토 및 수정	고남욱

12. 참조

1. 기사

주윤지, '스마트 스피커, 심장마비를 감지한다', MEDICAL Observer

<http://www.monews.co.kr/news/articleView.html?idxno=203613>

2019.07.19.에서 2022.04.23. 인출

이석원, 잠자는 아기 호흡, 스마트 스피커로 모니터링한다?, Tech Recipe

<https://techrecipe.co.kr/posts/11914>

2019.11.05.에서 2022.04.24. 인출

김희정, 안양시 어르신 "사람 살려" 소리 감지해 119 출동한다!, 내외뉴스통신

<http://www.nbnnews.co.kr/news/articleView.html?idxno=655919>

2022.02.09.에서 2022.04.22. 인출

"살려줘" 비명 듣고 보안업체 호출... AI스피커가 독거노인 돌보미 역할, 한국일보

<https://www.hankookilbo.com/News/Read/201907091662338206>

2019.07.09.에서 2022.04.25. 인출

최창현, AI 스피커의 놀라운 진화!... 비접촉식 심장 박동 모니터링 위한 세계 최초의 헬스케어 디바이스로 변신, 인공지능신문

<http://www.aitimes.kr/news/articleView.html?idxno=20509>

2021.03.12.에서 2022.04.26. 인출

-AI와 소리의 만남, 화장실 비명 감지하자 경찰 출동, ECONOMYChosun

http://economychosun.com/client/news/view.php?boardName=C00&t_num=13609813

2020.11.09.에서 2022.04.27. 인출

장희수, '심박수' 측정까지 가능... 날로 진화하는 AI 스피커, AI타임스

<http://www.aitimes.com/news/articleView.html?idxno=137212>

2021.03.10.에서 2022.04.28. 인출

이상철, 호흡음으로 수면 무호흡 진단 가능, 후생신보

<http://www.whosaeng.com/102420>

2018.06.21.에서 2022.04.29. 인출

박찬, 음성만 듣고 심장마비 예방....美 매요 클리닉, AI 음성 분석으로 심장 질환 감지, AI타임스

<http://www.aitimes.com/news/articleView.html?idxno=143775>

2022.04.04.에서 2022.04.30. 인출

한윤창, 사운드 AI 기술을 위한 비언어적 오디오 데이터 수집과 관리, 웹진

<https://webzine.aihub.or.kr/insight/vol07/content05.html>

2022.04.29. 인출

우영탁, [주목! 바이오벤처] "소리로 건강상태 알려주는 앱 만들것", 서울경제

<https://www.sedaily.com/NewsView/1VLKBW8VNU>

2019.07.04.에서 2022.04.26. 인출

손경호, 스마트홈 기기 대부분이 보안 위협에 무방비, zdnet korea

<https://zdnet.co.kr/view/?no=20150316162103>

2015.03.16.에서 2022.04.23. 인출

오다인, 늘어나는 애플리케이션, 백엔드 보안도 생각해야 한다, 보안뉴스

<https://www.boannews.com/media/view.asp?idx=55058>

2017.06.02.에서 2022.04.25. 인출

이주야, 미세먼지 기승에 산소발생기 관심 높아져, SOLAR TODAY

<http://www.solartodaymag.com/news/articleView.html?idxno=8016>

2019.02.05.에서 2022.04.27. 인출

김한식, 한국광기술원, AI 기반 실시간 소리 객체 인식·상황인지 솔루션 개발, 전자 신문

<https://m.etnews.com/20220412000114?obj=Tzo4OiJzdGRDbGFzcyl6Mjp7cz03OjyZWZlc mVyljtOO3M6NzoiZm9yd2FyZCI7czoxMzoid2ViIHRvIG1vYmlsZSI7fQ%3D%3D>

2022.04.12.에서 2022.04.24. 인출

황민규, SKT “AI 스피커로 독거 노인 23명 구해... 안전망 효과 입증”, 조선비즈

https://biz.chosun.com/site/data/html_dir/2020/05/20/2020052001956.html

2020.05.20.에서 2022.04.27. 인출

임솔, "침대 밑 센서나 웨어러블 착용은 불편" 무선 와이파이 기술로 편리하고 정확한 수면상태 측정 나선 에이슬립, 메디게이트뉴스

<https://m.medicatenews.com/news/2161288985>

2021.04.09.에서 2022.04.29. 인출

최윤정, 일상 속 모든 소리를 이해하는 ‘사운드 AI의 미래’, 제주의소리

<http://www.jesusori.net/news/articleView.html?idxno=334102>

2021.10.05.에서 2022.05.01. 인출

김동원, 김준모 카이스트 교수, 아이앤나와 영유아 울음소리 분석하는 AI 개발, Ai 타임스

<http://www.aitimes.com/news/articleView.html?idxno=143462>

2022.03.15.에서 2022.04.24. 인출

이준희, 스마트홈 핵심 서비스는 '헬스케어'...월 36만원 투자 의향, 전자신문

<https://m.etnews.com/20191104000339?obj=Tzo4OiJzdGRDbGFzcyl6Mjp7cz03OjyZWZlc mVyljtOO3M6NzoiZm9yd2FyZCI7czoxMzoid2ViIHRvIG1vYmlsZSI7fQ%3D%3D>

2019.11.04.에서 2022.04.25. 인출

오인규, 코로나 시대, 스마트 기기 홈 모니터링 시스템 확산, 의학신문

<http://www.bosa.co.kr/news/articleView.html?idxno=2165162>

2021.12.27.에서 2022.04.27. 인출

풀인, [풀인인사이트] "당신의 수면점수 A+인가요?" AI로 '꿀잠' 돋는 스타트업, 중앙일보

<https://www.joongang.co.kr/article/25007456#home>

2021.09.15.에서 2022.04.28. 인출

[헬스조선] 코골이는 어떻게 고혈압·당뇨병을 유발하나?

https://m.health.chosun.com/svc/news_view.html?contid=2021032202196

2021.03.23.에서 2022.04.28. 인출

원격 환자 모니터링 장치 놀라운 성장과 수요를 볼 수 있는 시장

<https://icibs.org/news/1169124/%EC%9B%90%EA%B2%A9-%ED%99%98%EC%9E%90-%EB%AA%A8%EB%8B%88%ED%84%B0%EB%A7%81-%EC%9E%A5%EC%B9%98-%EB%86%80%EB%9D%BC%EC%9A%B4-%EC%84%B1%EC%9E%A5%EA%B3%BC-%EC%88%98%EC%9A%94%EB%A5%BC-%EB%B3%BC-%EC%88%98/>

2021.04.26.에서 2022.04.28. 인출

'숙면은 보약'...왼쪽으로 누워 자고 운동은 3~4시간 전에 마쳐야

<https://m.hankookilbo.com/News/Read/A2022031912460001672>

2022.03.20.에서 2022.04.26. 인출

용인세브란스 정휘동 교수팀, CT로 수면무호흡증 검사 효과 예측

<https://www.docdocdoc.co.kr/news/articleView.html?idxno=2016746>

2021.11.19.에서 2022.04.26. 인출

[건강 올레길] 비만치료와 코골이 수면무호흡증 치료 함께 병행해야

<https://sports.donga.com/economy/article/all/20210901/109027923/1>

2021.09.01.에서 2022.04.26. 인출

[중앙 하와이] 수면 무호흡 진단 및 치료제 시장 점유율 2022 최고의 주요업체 분석, 지역 수익 분석, 공유 평가, 산업 확장 전략, 주요 부문, 비즈니스 성장 상태 및 예측 2029

<http://joonganghi.com/2022/04/29/%EC%88%98%EB%A9%B4-%EB%AC%B4%ED%98%BC%8%ED%9D%A1-%EC%A7%84%EB%8B%A8-%EB%B0%8F-%EC%B9%98%EB%A3%8C%EC%A0%9C-%EC%8B%9C%EC%9E%A5-%EC%A0%90%EC%9C%A0%EC%9C%A8-2022-%EC%B5%9C%EA%B3%A0%EC%9D%98-%EC%A3%BC/>

2021.04.29.에서 2022.04.30. 인출

App for Smart Speakers Uses AI to Detect Cardiac Arrest, MEDICAL DESIGN BRIEFS

<https://www.medicaldesignbriefs.com/component/content/article/mdb/insiders/mdb/stories/34716>

2019.06.24.에서 2022.04.29. 인출

2. 논문, 책

임희택, 지대범, 양회석 and 전삼현. (2018). MFCC와 DTW를 사용한 노인 낙상사고 인지 방안 연구. 한국IT정책경영학회 논문지, 10(1), 683-687.

박중오. (2021). 클라우드 서비스 기반 스마트 휴 환경에서 안전한 데이터 통신을 위한 메시지 통신 프로토콜 설계. 융합정보논문지, 11(7), 21-30.

Lee, S. B., Song, J., & Park, A. (2020). A Trend of Artificial Intelligence in the Healthcare. The Journal of the Korea Contents Association, 20(5), 448–456.
<https://doi.org/10.5392/JKCA.2020.20.05.448>

이기성, 조성원, 하태민, Ngo Luong Thanh Tra and Do Chi Thanh. (2022). 2차원 변환과 CNN 딥러닝 기반 음향 인식 시스템에 관한 연구. 스마트미디어저널, 11(1), 31-37.

Rhee, J. H., & Cha, S. H. (2021). A Long-term Monitoring Demonstration of Smart Home System for the Elderly. Journal of KIBIM, 11(3), 75–90.
<https://doi.org/10.13161/KIBIM.2021.11.3.075>

오원근 and 임동균. (2020). 딥러닝을 이용한 소리 분류 시 방해음의 영향 분석. 한국지식정보기술학회 논문지, 15(6), 973-981.

강성진. (2018). 스마트폰을 이용한 호흡 측정에 관한 연구. 반도체디스플레이기술학회지, 17(3), 108-112.

폐쇄성 수면 무호흡증의 병인 및 기전 (Pathogenesis and Mechanism of Obstructive Sleep Apnea), 2015, 최지호 (고려대학교 의과대학 이비인후-두경부외과학교실), 이승훈 (고려대학교 의과대학 이비인후-두경부외과학교실), 신철 (고려대학교 의과대학 호흡기내과학교실)

Baik, Ok Mi, Lee, Miyoung, Jang, Eunha, & Hong, Seokho. (2021). Smart Home Service Use among Older Adults Living Alone in Rural Areas. Health and Social Welfare Review, 41(4), 108–127. <https://doi.org/10.15709/HSWR.2021.41.4.108>

유광현. "딥러닝 기반의 비명소리 인식 및 구현." 국내석사학위논문 전남대학교, 2018. 광주

Chung, Yong-Joo. (2017). Scream Sound Detection Based on Universal Background Model Under Various Sound Environments. The Journal of the Korea Institute of Electronic Communication Sciences, 12(3), 485–492. <https://doi.org/10.13067/JKIECS.2017.12.3.485>

폐쇄성 수면무호흡증 환자에서 수면무호흡 정도, 수면 및 기분관련 척도, 수면중 활동도 간의 연관성 (The Correlation between Severity of Sleep Apnea, Sleep and Mood Related Scales, and Activity During Sleep in Obstructive Sleep Apnea Syndrome Patients) 2011년, 강현희 (가톨릭대학교 의과대학 성바오로병원 내과학교실), 강지영 (가톨릭대학교 의과대학 성바오로병원 내과학교실)

수면무호흡증과 수면변수가 인지기능에 미치는 영향과 우울증의 매개효과 (The Effects of Sleep Apnea and Variables on Cognitive Function and the Mediating Effect of Depression) 2017년, 이성훈 (연정 뇌기능 수면 연구소), 이나영 (연정 뇌기능 수면 연구소)

ZigBee를 이용한 수면 무호흡 검출 (Detection of Sleep Apnea Using ZigBee)

2006년, 김홍윤 (한서대학교 인터넷공학과); 이재용 (한서대학교 인터넷공학과)

[국내논문] 수면 무호흡증후군의 내과적 치료 (Medical Treatment of Sleep Apnea Syndrome), 1996, 문화식 (가톨릭대학교 의과대학 내과학교실), 최영미 (가톨릭대학교 의과대학 내과학교실)

소프트웨어공학 개론 [10판], Ian Sommerville, 한티 미디어 권기태 역.

Ha, T., Kang, S., & Cho, S. (2021). A Study on Hazardous Sound Detection Robust to Background Sound and Noise. Journal of Korea Multimedia Society, 24(12), 1606–1613.
<https://doi.org/10.9717/KMMS.2021.24.12.1606>

Wang J, Spicher N, Warnecke JM, Haggi M, Schwartze J, Deserno TM. Unobtrusive Health Monitoring in Private Spaces: The Smart Home. Sensors. 2021; 21(3):864.
<https://doi.org/10.3390/s21030864>

L. Abdoune and M. Fezari, "A sound database for health smart home," 2014 World Congress on Computer Applications and Information Systems (WCCAIS), 2014, pp. 1-5, doi: 10.1109/WCCAIS.2014.6916546.

Wang, A., Nguyen, D., Sridhar, A.R. et al. Using smart speakers to contactlessly monitor heart rhythms. Commun Biol 4, 319 (2021). <https://doi.org/10.1038/s42003-021-01824-9>

Jinwoo Kim, Kyungjun Min, Minhyuk Jung, Seokho Chi, Occupant behavior monitoring and emergency event detection in single-person households using deep learning-based sound recognition, Building and Environment, Volume 181, 2020, 107092, ISSN 0360-1323,

<https://doi.org/10.1016/j.buildenv.2020.107092>.

Dumpala, Sri Harsha & Alluri, K N R K. (2017). An Algorithm for Detection of Breath Sounds in Spontaneous Speech with Application to Speaker Recognition. 98-108.
10.1007/978-3-319-66429-3_9.

M. K. Nandwana, A. Ziae and J. H. L. Hansen, "Robust unsupervised detection of human screams in noisy acoustic environments," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 161-165, doi: 10.1109/ICASSP.2015.7177952.

O'Donovan R, Sezgin E, Bambach S, Butter E, Lin S. Detecting Screams From Home Audio Recordings to Identify Tantrums: Exploratory Study Using Transfer Machine Learning [published correction appears in JMIR Form Res. 2020 Jul 8;4(7):e21591]. JMIR Form Res. 2020;4(6):e18279. Published 2020 Jun 16. doi:10.2196/18279

Haider Mshali, Tayeb Lemlouma, Maria Moloney, Damien Magoni. A Survey on Health Monitoring Systems for Health Smart Homes. International Journal of Industrial Ergonomics, Elsevier, 2018, 66,

pp.26-56. ff10.1016/j.ergon.2018.02.002ff. fthal-01715576

Chan, J., Rea, T., Gollakota, S. et al. Contactless cardiac arrest detection using smart devices. npj Digit. Med. 2, 52 (2019). <https://doi.org/10.1038/s41746-019-0128-7>

3. 인터뷰

횡승식, 서울대학교 의과대학 예방의학 교실

강대희, 서울대학교 의과대학 예방의학 교실

박수경, 서울대학교 의과대학 예방의학 교실

신애선, 서울대학교 의과대학 예방의학 교실

박규남, 가톨릭대학교 의과대학 응급의학교실

최경호, 가톨릭대학교 의과대학 응급의학교실

이운정, 가톨릭대학교 의과대학 응급의학교실

윤준성, 가톨릭대학교 의과대학 응급의학교실

손미아, 강원대학교 의학전문대학원 예방의학 교실

이상아, 강원대학교 의학전문대학원 예방의학 교실

조희숙, 강원대학교 의학전문대학원 예방의학 교실

최지훈, 강원대학교 의학전문대학원 응급의료센터

강민성, 수원시 소방서 팀장

문상길, 서울시 종로구 소방서 팀원

정기성, 원광대학교 소방행정학과 교수

이승훈, <Daum-Kakao AI 플랫폼 팀 차장>

이태일, <現 스포츠투아이 대표, 前 NC 다이노스 대표, 前 네이버 이사>

금현창, <現 스포츠투아이 부사장, 前 네이버 이사 – 스포츠셀 팀장>

김민규, <네이버 AI 콘텐츠 기획팀 매니저>

유기준, <도봉구청 사회복지과 6급 공무원>

이지현, <삼성물산 건설부분, 래미안 기획 과장>

최수진, <GS 자이 건설부분 사업관리팀 대리>

박주희, <시티건설 건설기획부분 사업 관리팀 과장>

오윤재, <부동산 투자회사 리치 플레이인 이사>

김원, 중앙일보 산업부 기자

정도윤, 한국일보 산업부 기자