

서비스 명세서

**CPR 통합 보조 서비스**

**Push of Life**

B303

구고운, 김수민, 김예운, 라송연, 윤채영, 최봉준

삼성 청년 SW 아카데미, November 2024

---

본 개발 과정을 성공적으로 마칠 수 있도록 지도해 주신 서성수 컨설턴트님께 감사의 말씀을 드립니다. 또한, 설무아 강사님께 낙상 감지 센서와 관련하여 귀중한 조언과 도움을 주신 것에 깊이 감사드립니다.

아낌없는 지도를 해주신 최진학 코치님과 정휘원 코치님, 그리고 생활 전반에 세심한 도움을 주신 이순희 교육 프로님께도 감사드립니다.

본 개발이 이루어질 수 있도록 도와주신 모든 분들께 감사의 말씀을 드립니다.

---

## Abstract

본 서비스는 Wear OS 기반 웨어러블 디바이스를 활용하여 CPR 가이드와 응급 상황 통신 시스템을 통합한 새로운 응급 대응 솔루션을 제안한다. CPR 가이드는 센서 데이터를 기반으로 압박 깊이, 빈도, 팔 각도를 실시간으로 분석하며, 이를 통해 사용자가 CPR을 올바르게 수행할 수 있도록 지속적인 피드백을 제공한다. 가속도계와 중력 데이터를 활용한 3D 벡터 계산과 저주파 필터, 칼만 필터 등의 기법을 적용하여 CPR의 정확도를 향상시키고, 음성 및 시각적 안내를 실시간으로 제공한다.

응급 상황 통신 시스템은 위치에서 응급 상황이 감지되면 서버로 정보를 전송하고, 환자 주변 300m 이내의 사용자들에게 실시간 알림을 보낸다. Redis의 GEO Spatial 쿼리와 MongoDB 기반의 FCM 토큰 관리를 통해 데이터 처리 속도를 최적화하였으며, 알림을 받은 헬퍼는 BLE 스캔을 통해 환자와 연결된다. BLE 매칭이 완료되면 헬퍼의 위치에서 자동으로 CPR 가이드가 활성화되며, 실시간 피드백을 제공한다.

응급 상황에서 신속한 통신과 정확한 CPR 피드백을 동시에 제공함으로써 응급 대응의 효과성을 높이고, 병원 외 심정지 상황에서 생존율을 극대화할 가능성을 제시한다.

# Contents

<b>Abstract</b>	<b>3</b>
<b>1 기획 배경</b>	<b>10</b>
1.1 기획 의도 . . . . .	10
1.2 기존 서비스 분석 . . . . .	13
1.2.1 하트 히어로 . . . . .	13
1.2.2 CPR 서포터즈 . . . . .	13
1.2.3 HeartiSense . . . . .	13
1.2.4 CREDO . . . . .	14
1.2.5 결론 . . . . .	14
1.3 인터뷰 요약 . . . . .	15
1.3.1 환자 기저질환 및 사전 정보 제공의 유용성 . . . . .	15
1.3.2 중복 신고 접수의 영향 . . . . .	15
1.3.3 CPR의 중요성과 필요성 . . . . .	15
1.3.4 일반인의 CPR 인식과 시행 능력 . . . . .	15
1.3.5 정확한 CPR 시행과 결과의 차이 . . . . .	16
1.3.6 CPR 도우미 앱의 기대 효과 . . . . .	16
1.3.7 일반 시민의 CPR 시행 실수 . . . . .	16
1.3.8 CPR 성공을 위한 핵심 팁 . . . . .	16
1.4 서비스 목표 . . . . .	17
1.5 주요 기능 . . . . .	18
1.5.1 응급상황 감지 기능 . . . . .	19
1.5.2 AED 위치 제공 기능 . . . . .	19
1.5.3 주변 유저 알림 기능 . . . . .	19
1.5.4 헬퍼의 CPR 가이드 자동화 기능 . . . . .	19
1.5.5 CPR 피드백 및 실시간 가이드 . . . . .	19
1.5.6 구급대 자동 신고 기능 . . . . .	20
<b>2 응급상황 감지</b>	<b>21</b>
2.1 서비스 개요 . . . . .	21
2.2 센서 . . . . .	22
2.2.1 심박수 센서(Sensor.TYPE_HEART_RATE) . . . . .	22
2.2.2 중력 센서(Sensor.TYPE_GRAVITY) . . . . .	22
2.2.3 선형 가속도 센서(Sensor.TYPE_LINEAR_ACCELERATION) . . . . .	23
2.3 서비스 플로우 . . . . .	24
2.3.1 위치 응급상황 감지 . . . . .	24
2.3.2 모바일 응급상황 감지 . . . . .	25

<b>3 구조 요청</b>	<b>27</b>
3.1 서비스 개요 . . . . .	27
3.2 모바일 - 모바일 통신 : FCM . . . . .	28
3.2.1 Firebase Cloud Messaging(FCM) . . . . .	28
3.2.2 성능 최적화 - Spring Webflux & Kafka . . . . .	29
3.2.3 성능 최적화 - Redis & MongoDB . . . . .	30
3.2.4 성능 평가 . . . . .	31
3.3 모바일 - 모바일 통신 : BLE . . . . .	32
3.3.1 Bluetooth Low Energy(BLE) . . . . .	32
3.3.2 Bluetooth Classic vs BLE . . . . .	33
3.4 위치 - 모바일 통신 . . . . .	34
3.4.1 MessageClient . . . . .	34
3.4.2 프로젝트 내 사용된 알림 채널 . . . . .	34
<b>4 손목 각도 피드백 시스템</b>	<b>35</b>
4.1 서비스 개요 . . . . .	35
4.2 이론적 배경 . . . . .	36
4.2.1 중력 가속도와 손목 방향 벡터 . . . . .	36
4.2.2 3D 벡터 내적과 각도 계산 . . . . .	36
4.2.3 각도 측정을 통한 피드백 시스템 . . . . .	36
4.3 구현 방법 . . . . .	37
4.3.1 시스템 설계 . . . . .	37
4.3.2 데이터 필터링 . . . . .	37
4.3.3 실시간 피드백 . . . . .	37
4.4 결과 . . . . .	38
4.4.1 환경 및 과정 . . . . .	38
4.4.2 성능 평가 . . . . .	38
<b>5 압박 깊이 및 빈도 피드백 시스템</b>	<b>39</b>
5.1 서비스 개요 . . . . .	39
5.2 이론적 배경 . . . . .	41
5.2.1 선형 가속도와 중력 가속도의 역할 . . . . .	41
5.2.2 중력 보정을 통한 수직 선형 가속도 계산 . . . . .	41
5.3 구현 방법 . . . . .	42
5.3.1 데이터 수집 및 필터링 . . . . .	42
5.3.2 가속도 적분을 통한 깊이 계산 . . . . .	42
5.3.3 깊이 추적을 통한 빈도 측정 . . . . .	42
5.4 결과 . . . . .	43
5.4.1 실험 환경 및 과정 . . . . .	43
5.4.2 성능 평가 . . . . .	43
<b>6 기대 효과</b>	<b>44</b>
6.1 삼성헬스와의 협력 가능성 . . . . .	44
6.1.1 지역 응급 네트워크 형성 . . . . .	45
6.1.2 골든타임 확보에 기여 . . . . .	45
6.1.3 안전한 의료 환경 제공 . . . . .	45
6.2 생존율 향상 . . . . .	46
6.2.1 자동화된 CPR 가이드 . . . . .	46

6.2.2	긴급 알림을 통한 지연 시간 최소화 . . . . .	46
<b>A</b>	<b>화면 정의서</b>	<b>47</b>
A.1	스마트폰 화면 . . . . .	48
A.2	위치 화면 . . . . .	54
<b>B</b>	<b>CPR 피드백 시스템을 위해 사용된 필터</b>	<b>59</b>
B.1	저주파 통과 필터 (Low-pass Filter) . . . . .	59
B.2	칼만 필터 (Kalman Filter) . . . . .	59
B.2.1	업데이트 단계 . . . . .	60
B.3	이동 평균 필터 . . . . .	62
B.4	적분 필터 (Trapezoidal Integration) . . . . .	62
<b>C</b>	<b>GeoSpatial &amp; GeoHash</b>	<b>63</b>
C.1	GeoSpatial . . . . .	64
C.1.1	GEOADD . . . . .	64
C.1.2	GEODIST . . . . .	64
C.1.3	GEORADIUS . . . . .	64
C.1.4	GEORADIUSBYMEMBER . . . . .	64
C.2	GeoHash . . . . .	65
C.2.1	GeoHash의 동작 원리 . . . . .	65
C.2.2	GeoHash의 장점 . . . . .	65
<b>D</b>	<b>MongoDB Sharding</b>	<b>67</b>
D.1	Sharding . . . . .	67
D.1.1	Mongos . . . . .	67
D.1.2	Config Server . . . . .	68
D.1.3	Shard . . . . .	68

# List of Tables

1.1	기존 서비스 비교 분석	14
3.1	알림 기능 성능 비교	31
6.1	CPR 시행 여부에 따른 생존율 및 회복률 비교	46

# List of Figures

1.1	급성심장정지 발생 추이[13]	10
1.2	일반인 심폐소생술 시행률(2012~2022) 및 시행 여부에 따른 생존율[13]	11
1.3	응급상황 도움요청 서비스 플로우	17
1.4	서비스 기능 및 실제 작동 화면	18
2.1	응급상황 감지 서비스 개요	21
2.2	심박수 이상 감지	22
2.3	낙상 감지	23
2.4	위치 응급상황 감지 서비스	24
2.5	다양한 경우에서의 서비스 플로우	26
3.1	모바일 통신 전체 플로우	28
3.2	성능 최적화 : webflux & Kafka	29
3.3	성능 최적화 - Redis & MongoDB	30
3.4	Bluetooth Classic vs BLE	32
4.1	손목 각도 피드백 시스템	35
5.1	올바른 CPR 수행시의 깊이 및 가속도 그래프	40
5.2	올바르지 못한 CPR 수행시의 깊이 및 가속도 그래프	40
6.1	삼성헬스와의 협력시 기대효과	44
A.1	회원가입 화면	48
A.2	로그인 화면	48
A.3	정보추가 화면	49
A.4	정보추가 확인 화면	49
A.5	메인 화면(위치가 없는 경우)	50
A.6	메인 화면	50
A.7	메인 화면(주변에 응급환자가 있을 경우)	51
A.8	위치 가이드 화면	51
A.9	CPR 가이드 화면	52
A.10	AED 위치 화면	52
A.11	AED 위치 추가정보 화면	53
A.12	응급환자 및 AED 위치 화면	53
A.13	메인 화면 - 1	54
A.14	메인 화면 - 2	54
A.15	신고 화면	55
A.16	낙상 감지 화면	55
A.17	CPR 가이드 헬퍼 화면	56

A.18 CPR 가이드 환자 화면	57
A.19 응급환자 및 AED 위치 화면	58
B.1 가능한 Q,R에 대한 완전 탐색을 통해 시각화한 그래프	61
B.2 시스템 안정성 모니터링 이미지	61
B.3 filtered depth, raw depth 비교 그래프	62
C.1 Redis GeoSpatial[4]	63
C.2 GeoHash 동작 원리	65
D.1 Sharding Structure	68

# Chapter 1

## 기획 배경

### 1.1 기획 의도

‘Push Of Life(օ]하 POL)’는 Wear OS 및 Mobile 센서를 기반으로 한 응급 상황 자동 감지 및 신속 구조 지원 서비스로, 응급 대응의 체계적 자동화를 목표로 기획되었다. 본 서비스는 사용자가 착용한 디바이스를 통해 낙상, 심박수 변화 등의 데이터를 실시간으로 감지하여 응급 상황을 판단하고, 이에 대한 즉각적이고 효율적인 대응을 제공한다.

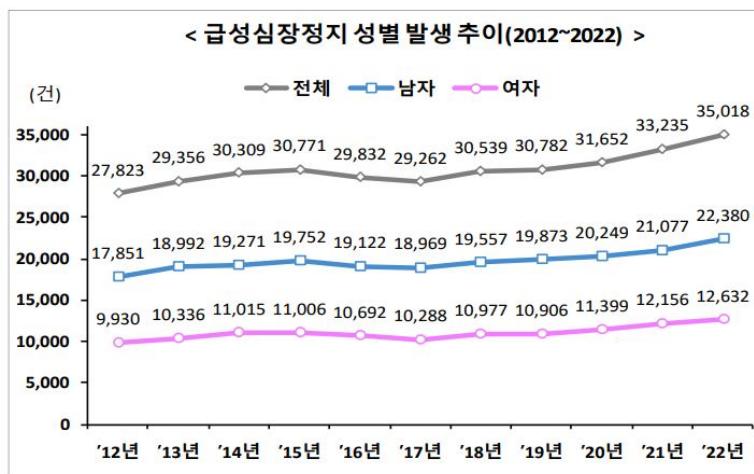


Figure 1.1: 급성심장정지 발생 추이[13]

서비스의 주요 의도는 기술적 인프라를 활용하여 기존 응급 구조 시스템의 한계를 보완하는

데 있다. POL은 응급 상황 발생 시 사용자의 사전 정보를 기반으로 자동 신고를 수행하며, 주변 사용자에게 알림을 전송하여 신속한 구조를 지원한다. 또한 구조사가 환자와 가까워지면 근거리 통신을 통해 구조사와 환자의 디바이스를 연결하고, 구조사의 동작을 감지하여 적절한 CPR 피드백을 제공함으로써 응급 처치의 효과성을 극대화한다.

현대 사회는 급격한 고령화와 이에 따른 질병 및 사고 발생률 증가로 인해 심정지와 같은 응급 상황의 중요성이 점차 부각되고 있다[12]. 2022년 119구급대 통계에 따르면, 급성 심정지 환자는 35,018명(인구 10만 명당 68.3명)으로 매년 증가하고 있으며, 특히 남성(63.9%)에서 높은 비율을 차지하며 연령이 증가할수록 발생 빈도가 높아지는 경향을 보이고 있다. 급성 심정지 환자의 약 72.2%는 60세 이상의 고령 인구로, 대한민국 고령화 사회에서는 응급 상황 발생 위험도와 대응 필요성이 날로 중요해지고 있다. 특히 공공장소에서 발생하는 심정지 사례는 전체의 약 16.4%(5,713건)를 차지하여, 사회적 안전망과 공공 응급 구조 체계의 강화가 필수적임을 시사한다.

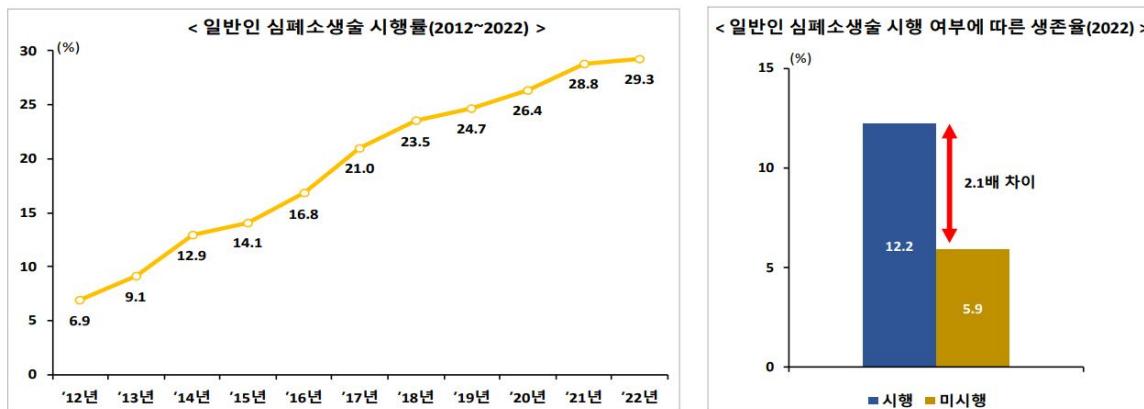


Figure 1.2: 일반인 심폐소생술 시행률(2012~2022) 및 시행 여부에 따른 생존율[13]

또한, 일반인의 심폐소생술(CPR) 시행률은 꾸준히 증가하는 추세다. 질병관리청의 조사에 따르면, 2022년 기준 일반인 심폐소생술 시행률은 29.3%로 전년도 대비 0.5%p 증가했으며, 이는 2012년 6.9%에서 10년 만에 4배 이상 증가한 수치이다. 일반인이 심폐소생술을 시행한 경우 생존율은 12.2%로, 미시행 시 생존율(5.9%)보다 약 2.1배 높아 CPR의 중요성과 효과가 입증되고 있다. 하지만 시행률은 여전히 낮은 수준에 머물러 있으며, 교육 및 기술적 지원의 부족으로 인해 CPR 수행의 한계[9]가 존재한다.

POL 서비스는 이러한 배경에서 출발하였다. 본 서비스는 고령화 사회에서 발생하는 응급 상황

증가와 일반인 CPR 참여율 제고의 필요성을 해결하고자, 낙상 및 심박수 감지를 통한 응급 상황 자동 감지, CPR 피드백 제공, 신속한 구조 지원 등을 통해 응급 대응의 효과성을 극대화하려는 목표를 가지고 있다. 또한 사용자와 구조자의 편의성을 증대시키며, 비전문가도 효과적으로 응급 처치에 기여[1]할 수 있도록 설계되었다.

## 1.2 기존 서비스 분석

기존 응급 구조 및 CPR 관련 서비스들은 각각의 강점을 가지고 있지만, 실시간 긴급 대처와 구조자의 편의성을 위한 통합적 지원 면에서는 한계가 있다. 이를 구체적으로 살펴보면 다음과 같다.

### 1.2.1 하트 히어로

먼저, 하트 히어로는 스마트폰의 센서를 활용해 CPR의 압박 깊이와 속도를 실시간으로 피드백하는 서비스다. 사용자는 스마트폰을 가슴에 올려놓고 CPR을 수행하며, 센서를 통해 압박 깊이와 속도가 적절한지 확인할 수 있다. 또한, 응급 상황 발생 시 근처 AED(자동심장충격기) 위치를 지도 형태로 제공하고, 환자와 문자 및 전화로 소통할 수 있는 커뮤니케이션 기능도 포함하고 있다. 하지만 스마트폰의 물리적 센서 의존성으로 인해 사용 환경이 제한적이고, 실제 구조 상황에서 효율적으로 활용하기 어려운 단점이 있다.

### 1.2.2 CPR 서포터즈

CPR 서포터즈는 AED 위치 정보를 제공하는 데 특화된 서비스다. 사용자의 현재 위치를 기반으로 가장 가까운 AED 위치를 계산하고 최단거리 도보 경로를 안내해 AED를 신속히 확보할 수 있도록 돕는다. 하지만 AED 위치 안내에만 초점이 맞춰져 있어, CPR 피드백이나 응급 상황에서의 실시간 지원 기능은 제공하지 않아 구조자의 활동 범위가 제한적이라는 한계가 있다.

### 1.2.3 HeartiSense

HeartiSense는 CPR 훈련용 마네킹과 연동해 압박 깊이와 속도 등에 대한 정밀한 피드백을 제공한다. 특히, CPR 기술을 학습하거나 훈련하는 데 최적화된 서비스로, 정확한 피드백과 데이터를 통해 훈련 효과를 극대화할 수 있다. 하지만 실제 응급 상황에서 활용하기에는 한계가 있으며, 마네킹 의존적인 특성으로 인해 실질적인 구조 활동보다는 CPR 기술 훈련에 초점이 맞춰져 있다는 단점이 있다.

### 1.2.4 CREDO

CREDO는 스마트 워치를 활용한 CPR 지원 기능을 제공하며, CPR 압박 리듬과 강도를 확인할 수 있도록 메트로놈 리듬과 가속도 그래프를 제공한다. 또한, CPR 진행 데이터를 모바일 어플리케이션과 연동해 기록을 확인할 수 있는 기능도 지원한다. 하지만 의료기기로 인증된 워치를 판매하는 구조로 인해 사용자 범위가 제한적이며, 제공되는 CPR 피드백의 다양성과 상세성이 부족하다는 한계가 있다. 특히, 헬퍼와 환자 간 실시간 협력이나 응급 상황에서의 자동화 지원 기능이 미흡하다는 점이 단점으로 지적된다.

	실시간 피드백	AED 위치 제공	응급 알림	실전 사용 가능성
하트 히어로	O	O	O	O
CPR 서포터즈	X	O	X	△
HeartiSense	O	X	X	X
CREDO	O	X	X	△

Table 1.1: 기존 서비스 비교 분석

### 1.2.5 결론

이처럼 기존 서비스들은 각자의 특정 기능에는 강점이 있지만, 응급 상황에서 즉각적으로 대처할 수 있는 실시간 자동화 대응이나 헬퍼와 환자 간 정보 연계 면에서는 부족하다. 대부분 훈련용 기능이나 특정 데이터 제공(예: AED 위치)에만 치중돼 있어, 골든타임 내 구조 요청과 CPR 피드백을 통합적으로 제공하기에는 한계가 있다. 특히, 헬퍼와 환자 간 상호작용과 실시간 협력을 지원하는 기능이 미흡해 효과적인 응급 구조를 위한 완전한 솔루션으로 보기는 어렵다.

따라서, 이러한 한계를 보완하고 실시간 대응 및 자동화된 응급 지원 기능을 갖춘 새로운 서비스가 필요하다. 이를 통해 사용자들이 응급 상황에서도 신속하고 효과적으로 대처할 수 있는 환경을 제공해야 한다.

## 1.3 인터뷰 요약

본 프로젝트에선 CPR 가이드 앱의 실질적 경험과 교육 현장에서의 관점을 탐구하기 위해 반구조화된 인터뷰를 실시하였다. 인터뷰는 강진호 소방대원과 충청남도 안전체험관의 교육관 두 명을 대상으로 진행되었다.

### 1.3.1 환자 기저질환 및 사전 정보 제공의 유용성

환자의 기저질환, 복용 약물, 병력, 치료받는 병원 정보 등 사전 정보는 현장 활동에 큰 도움이 된다. 특히 환자의 의식이 없거나 지남력이 떨어지는 경우, 사전 정보는 구급대원들이 정확한 처치와 병원 이송을 계획하는 데 필수적이다. 현장에서 환자의 정보를 파악하는 데 한계가 있기 때문에 사전에 간결한 문장 형태로 환자의 인적사항(이름, 생년월일, 연락처)과 과거력 등이 제공된다면 구급활동의 효율성을 크게 높일 수 있다.

### 1.3.2 중복 신고 접수의 영향

중복 신고 접수는 문제가 되지 않는다. 119 상황실에서 여러 신고 접수를 하나의 출동 건으로 통합해 구급대를 출동시키기 때문에 중복 접수로 인한 혼선은 없다. 대규모 사고 시에도 다수의 신고를 효율적으로 관리할 수 있다.

### 1.3.3 CPR의 중요성과 필요성

심정지 상황에서 CPR은 생존율과 예후를 결정짓는 핵심 요소이다. 심장이 멈춘 뒤 4분 이내에 뇌에 혈액이 공급되지 않으면 뇌사 상태로 이어질 수 있으므로, 즉각적인 CPR이 필수적이다. CPR은 생명을 구하는 첫 단계로, 빠르고 정확한 시행이 중요하다.

### 1.3.4 일반인의 CPR 인식과 시행 능력

일반인들은 CPR의 중요성과 필요성을 기본적으로 알고 있지만, 실제로 정확히 시행하는 데는 어려움을 겪는다. 군 복무 경험이나 학교 교육 등을 통해 CPR을 접한 경우가 많지만, 실전에서 정확한 압박 위치와 깊이를 유지하는 데 미숙한 경우가 많다. CPR 상황을 정확히 인지하지 못하거나, 불규칙 호흡이 있는 환자에게 CPR을 시행하지 않는 등의 실수도 빈번히 발생한다.

### 1.3.5 정확한 CPR 시행과 결과의 차이

CPR을 정확히 시행할 경우, 소생률과 뇌기능 회복률이 크게 향상된다. 반대로 부정확한 압박 위치나 깊이로 시행하면 생존 가능성이 떨어진다. 정확한 CPR 시행을 위해서는 압박 위치, 깊이, 속도, 자세 등을 유지하며 ”High-Quality CPR”을 수행해야 한다.

### 1.3.6 CPR 도우미 앱의 기대 효과

CPR 가이드 앱은 단 한 사람의 생명이라도 구할 수 있다면 의미 있는 성과로 볼 수 있다. 앱이 정확한 압박 리듬(분당 100-120회), 압박 깊이(성인 5-6cm 기준)를 실시간으로 안내하고, 가장 가까운 AED 위치를 제공한다면 CPR 시행의 성공률을 높일 수 있다. 진동과 소리로 압박 리듬을 제공하거나 음성 피드백으로 실시간 조언을 제공하는 기능은 특히 효과적일 것이다.

### 1.3.7 일반 시민의 CPR 시행 실수

일반 시민들이 가장 자주 하는 실수는 CPR 상황을 정확히 인지하지 못하거나, 불규칙 호흡이 있는 환자를 심정지로 오인하지 못하는 것이다. 시행 중에는 압박 위치와 깊이가 부정확한 경우가 많아 소생률에 부정적인 영향을 미친다. 이러한 실수를 줄이기 위해 정확한 가이드를 제공하는 기술적 지원이 필요하다.

### 1.3.8 CPR 성공을 위한 핵심 팁

CPR 성공의 핵심은 ”생존 사슬”에 따라 유기적으로 진행되는 구조 절차를 준수하는 것이다. 특히 압박 위치와 깊이, 속도, Hands-Off Time을 최소화하는 등 ”High-Quality CPR”을 유지해야 한다. 심폐소생술 관련 자세한 정보는 대한심폐소생협회 등에서 제공하는 자료를 통해 학습할 수 있다.

## 1.4 서비스 목표

스마트 워치와 모바일 기술을 기반으로 긴급 상황에서 신속하고 효과적인 구조와 생명 구호를 지원하는 응급 처치 어플리케이션이다. 낙상, 심박수 이상, 심정지와 같은 상황을 자동으로 감지하고, 이를 주변 사용자 및 헬퍼에게 알림으로 전달한다. 또한, AED 위치 안내, 실시간 CPR 피드백, 환자 상태 모니터링 기능을 통해 골든타임 내에 신속하고 정확한 응급 구조를 가능하게 한다.

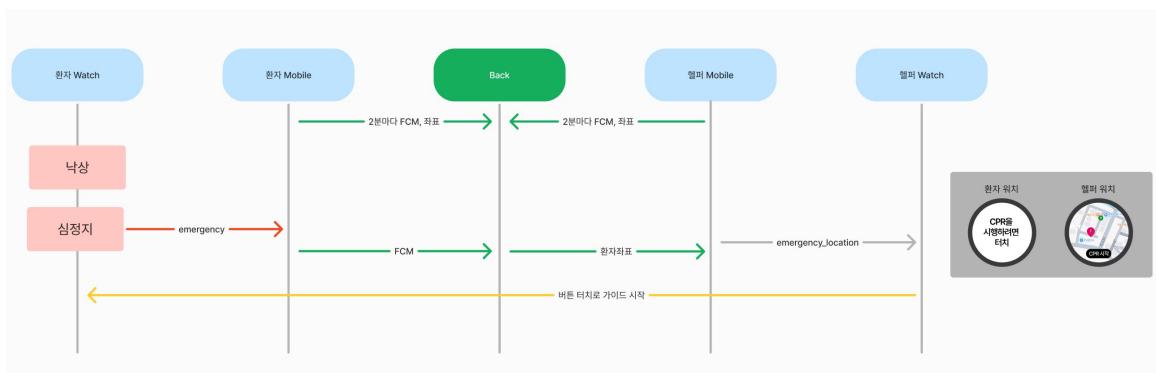


Figure 1.3: 응급상황 도움요청 서비스 플로우

## 1.5 주요 기능

본 서비스는 스마트 워치와 모바일 어플리케이션을 활용하여 긴급 상황 감지, AED 위치 제공, CPR 가이드 및 실시간 피드백, 구급대 자동 신고 기능 등을 통합적으로 제공한다. 이를 통해 응급 상황에서 발생할 수 있는 다양한 문제를 자동화된 방식으로 해결하고, 사용자와 헬퍼 간의 협력을 극대화한다.



Figure 1.4: 서비스 기능 및 실제 작동 화면

### 1.5.1 응급상황 감지 기능

응급상황 감지 기능은 심박수가 위험 수준 이하로 감소하거나 낙상이 감지될 경우 워치에서 경고음(사이렌)을 발생시키고, 3초간 반응이 없으면 등록된 응급 연락처와 구급대에 자동으로 알림을 발송한다. 모바일에서도 낙상 감지 기능을 활성화할 수 있으며, 모바일 기기에서는 30초 동안 반응이 없을 경우 자동으로 신고가 이루어진다.

### 1.5.2 AED 위치 제공 기능

AED 위치 제공 기능은 응급 상황 발생 시 환자와 가장 가까운 AED(자동심장충격기) 위치를 워치와 모바일 화면에서 지도 형태로 제공한다. 이를 통해 헬퍼가 AED를 신속히 확보할 수 있도록 지원하며, 구조 과정에서의 효율성을 높인다.

### 1.5.3 주변 유저 알림 기능

주변 유저 알림 기능은 낙상이나 심박수 이상이 감지되었을 때 주변의 헬퍼들에게 즉시 긴급 알림을 발송한다. 헬퍼는 알림을 클릭하면 응급 환자의 위치와 주변 AED 위치를 워치 및 모바일 화면에서 시각적으로 확인할 수 있다. 이 기능은 헬퍼와 환자를 빠르게 연결하여 구조 활동이 지연되지 않도록 돋는다.

### 1.5.4 헬퍼의 CPR 가이드 자동화 기능

헬퍼의 CPR 가이드 자동화 기능은 헬퍼가 환자의 위치 화면을 누르면 BLE 통신[5]을 통해 CPR 가이드가 자동으로 헬퍼의 워치에 실행되도록 설계되었다. 또한, 헬퍼가 환자를 발견한 순간, 워치 타일을 이용해 간편하게 어플리케이션에 접근할 수 있어 추가적인 시간 소모 없이 즉각적인 CPR 지원을 시작할 수 있다.

### 1.5.5 CPR 피드백 및 실시간 가이드

CPR 피드백 및 실시간 가이드는 CPR의 각도, 빈도, 깊이를 분석하여 상세한 피드백을 제공한다. 헬퍼의 워치에서는 가속도 그래프를 통해 CPR의 정확도를 시각적으로 확인할 수 있으며, 환자의 워치에서는 실시간 심박수를 모니터링하여 CPR 진행 상황을 지원한다. 이를 통해 헬퍼가 적절한 CPR 리듬과 강도를 유지할 수 있도록 실시간으로 도움을 준다.

### 1.5.6 구급대 자동 신고 기능

구급대 자동 신고 기능은 낙상이나 심박수 이상이 감지될 경우, 모바일 앱에 미리 입력된 환자의 이름, 나이, 병력, 약물 알레르기 등의 정보를 구급대에 자동으로 전송한다. 헬퍼가 신고할 경우에도 신고자의 위치를 기반으로 구급대가 즉시 출동할 수 있도록 지원한다.

이와 같이 본 서비스는 긴급 상황에서 발생할 수 있는 모든 프로세스를 자동화하고 통합하여 환자의 생존 가능성을 극대화할 수 있는 종합적인 응급 지원 솔루션을 제공한다.

# Chapter 2

## 응급상황 감지

### 2.1 서비스 개요

본 서비스는 스마트 워치와 모바일 기기의 센서를 활용하여 응급상황을 실시간으로 감지하고, 긴급 상황에 신속히 대응하는 기능을 제공한다. 워치에서는 심박수에 이상이 감지되거나 낙상이 발생하면 심박수 데이터를 분석하여 긴급성을 확인하고 경고음을 울린다. 모바일에서는 30초 동안 사이렌을 울리며 사용자 반응을 기다리고, 반응이 없을 경우 자동으로 구조 요청을 발송한다. 이러한 시스템은 낙상이나 심박수 이상으로 인해 의식이 없거나 도움이 필요한 상황에서 구조 요청이 지연되지 않도록 설계되었다.

**Foreground Service** 기반,



Figure 2.1: 응급상황 감지 서비스 개요

## 2.2 센서

응급상황 감지 서비스는 스마트 워치와 모바일 기기에 내장된 다양한 센서를 활용[6]하여 사용자의 움직임과 생체 신호를 실시간으로 모니터링한다. 주요 센서로는 심박수 센서(Sensor.TYPE\_HEART\_RATE), 중력 센서(Sensor.TYPE\_GRAVITY), 선형 가속도 센서(Sensor.TYPE\_LINEAR\_ACCELERATION)가 있으며, 이 센서들은 사용자 상태를 종합적으로 분석하여 낙상과 심박수 이상을 감지하는 핵심적인 역할을 수행한다.

### 2.2.1 심박수 센서(Sensor.TYPE\_HEART\_RATE)

심박수 센서는 사용자의 심박수를 실시간으로 측정한다. 서비스는 0.5초마다 심박수를 확인하며, 심박수가 비정상적으로 낮은 범위(5~30 bpm)에 해당하면 긴급 상황으로 판단한다. 심박수는 사용자의 상태를 평가하는 데 중요한 지표로 활용되며, 낙상 여부와 결합하여 정확한 응급상황 감지를 지원한다.

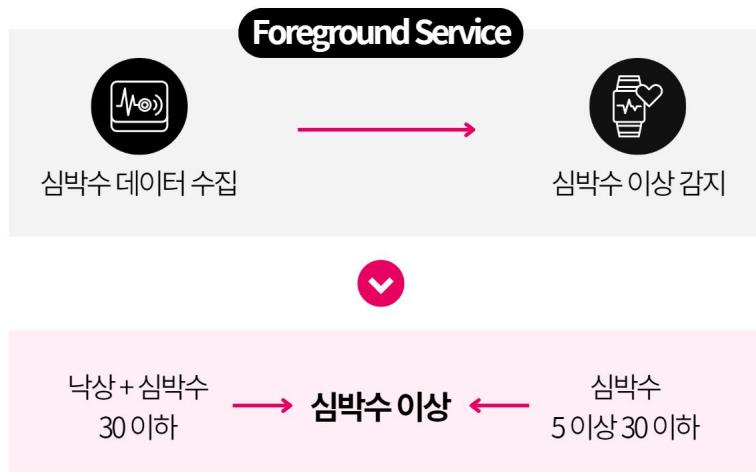


Figure 2.2: 심박수 이상 감지

### 2.2.2 중력 센서(Sensor.TYPE\_GRAVITY)

중력 센서는 사용자의 자세 변화를 감지한다. 정상적인 상태와 낙상 시 자세 변화에서 나타나는 중력 벡터의 차이를 분석하여 낙상 여부를 판단한다.

### 2.2.3 선형 가속도 센서(Sensor.TYPE\_LINEAR\_ACCELERATION)

사용자의 움직임으로 발생하는 가속도 변화를 측정한다. 낙상이 발생하면 일반적으로 큰 가속도 변화가 감지되며, 이를 기반으로 충격 정도를 계산할 수 있다[11]. 선형 가속도 센서의 데이터와 중력 벡터를 결합하여 중력 방향으로 가해지는 충격 성분을 분석함으로써 신뢰도 높은 낙상 감지를 수행한다.

이 센서들은 백그라운드 환경에서도 저전력으로 동작하며, 데이터를 지속적으로 수집하여 실시간 응급 상황 감지가 가능하도록 지원한다.

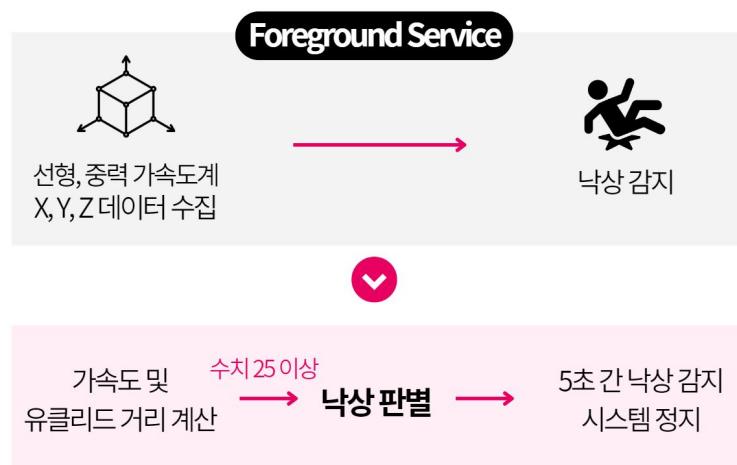


Figure 2.3: 낙상 감지

## 2.3 서비스 플로우

응급상황 감지 서비스는 스마트 워치와 모바일 기기에서 백그라운드 서비스로 실행되며, 사용자의 움직임과 심박수 데이터를 실시간으로 모니터링한다. 센서로부터 데이터를 지속적으로 수집하고 분석하여 낙상 여부와 심박수 이상 여부를 판단하며, 이상 상황이 감지될 경우 포그라운드 서비스로 전환되어 즉각적인 알림과 구조 요청 프로세스를 실행한다.

백그라운드 단계에서는 심박수 센서, 중력 센서, 선형 가속도 센서를 활용하여 사용자의 상태를 모니터링한다. 심박수 센서는 0.5초마다 데이터를 측정하며, 심박수가 530 bpm 사이에 해당하면 심박수 이상으로 판단한다. 중력 센서와 선형 가속도 센서는 낙상을 감지하기 위해 데이터를 분석하며, 중력 방향의 충격이 임계값(예:  $25m/s^2$ )을 초과하면 낙상으로 판단한다.

응급상황이 감지되면 사용자와 주변 환경에 긴급 상황을 알리고, 구조 요청을 진행한다.



Figure 2.4: 워치 응급상황 감지 서비스

### 2.3.1 워치 응급상황 감지

스마트 워치에서는 심박수 이상이나 낙상이 감지되었을 때 경고음을 울려 주변의 주의를 끌고, 사용자의 상태를 지속적으로 모니터링한다. 심박수 센서를 통해 수집된 데이터는 심박수 이상 여부를 실시간으로 평가하며, 이상이 계속될 경우 긴급 상황으로 판단된다.

워치는 응급상황임을 모바일 기기로 전송하여 연동 프로세스를 시작한다. 워치에서는 사이렌을 울리고, 모바일에서는 구조요청을 보낸다.

### 2.3.2 모바일 응급상황 감지

모바일 기기에서의 응급상황감지는 위치와 독립적으로 작동하며, 낙상을 감지했을 때 30초 동안 사이렌을 울린다. 사이렌은 사용자에게 경고음을 제공하며, 사용자의 반응을 기다린다. 30초 내에 사용자가 반응하지 않을 경우, 모바일 앱은 등록된 응급 연락처와 구급대에 자동으로 구조 요청을 발송한다.

구조 요청 시 사용자의 위치 정보, 심박수 데이터, 낙상 감지 시간 등의 정보가 포함되며, 이는 구조 요청의 정확성을 높이고 신속한 대응을 가능하게 한다.

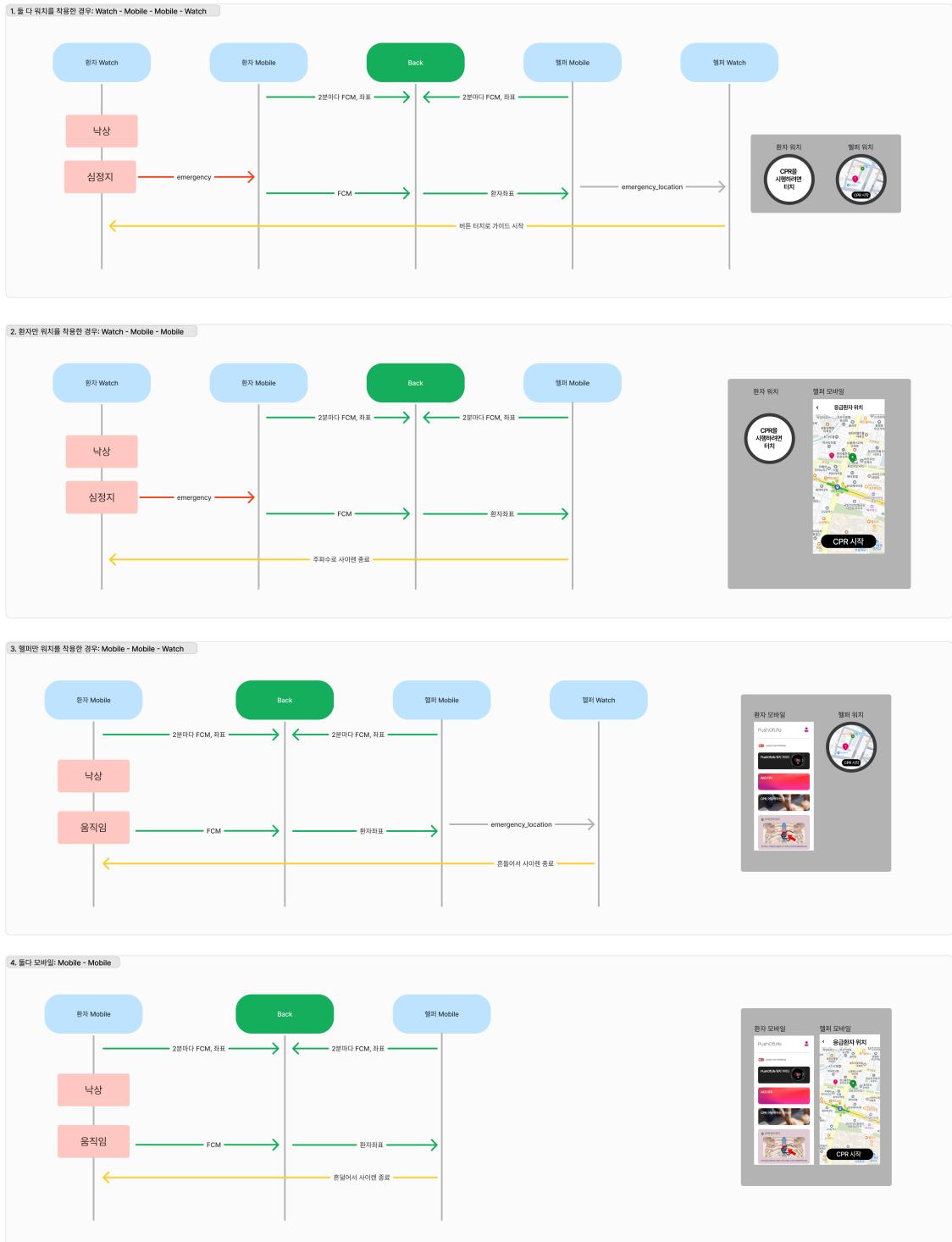


Figure 2.5: 다양한 경우에서의 서비스 플로우

# Chapter 3

## 구조 요청

### 3.1 서비스 개요

본 서비스는 응급 상황을 신속하게 감지하고 이를 바탕으로 빠르게 근처 유저들에게 알림을 전송하여 응급처치를 유도하는 기능을 제공한다. 이를 통해 사용자는 신속하게 도움을 받을 수 있다.

환자가 긴급 상황에 처했을 때, 환자의 모바일 기기가 서버로 요청을 보내고, 서버는 이를 수신하여 근처의 유저들에게 알림을 전송하는 방식으로 작동한다. 이 시스템은 Firebase Cloud Messaging(FCM)과 Bluetooth Low Energy(BLE), Redis와 MongoDB 등을 활용해 성능 최적화 및 안정성을 확보하였다.

## 3.2 모바일 - 모바일 통신 : FCM

환자가 긴급 상황에 처했을 때, 환자의 모바일 기기는 서버에 요청을 보내고, 서버는 이를 수신하여 Firebase Cloud Messaging(FCM)을 통해 근처의 유저들에게 알림을 전송한다. 이 시스템은 신속하고 안정적인 알림 전송을 목표로 하며, FCM을 사용한 이유와 이를 통한 최적화 방법은 다음과 같다.

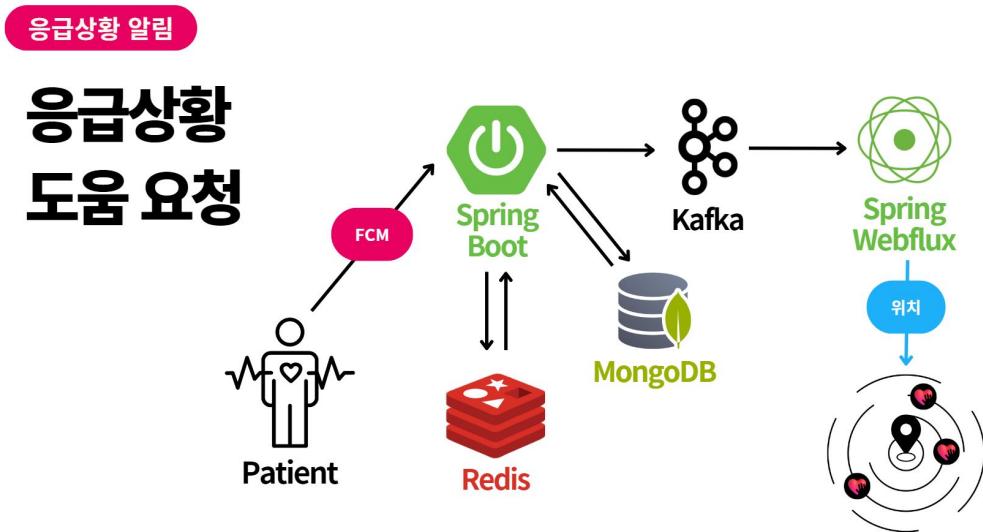


Figure 3.1: 모바일 통신 전체 플로우

### 3.2.1 Firebase Cloud Messaging(FCM)

FCM[10]은 구글이 제공하는 클라우드 기반 푸시 알림 서비스로, Android와 iOS 기기 모두에서 실시간 알림을 전송할 수 있다. FCM은 무료로 제공되며, 서버에서 클라이언트 애플리케이션으로 푸시 알림을 보낼 수 있는 기능을 제공한다. FCM은 대규모 사용자에게 동시에 알림을 전송할 수 있으며, 요청 당 최대 4KB의 데이터까지 전송할 수 있다. 본 서비스에서는 환자 기기의 긴급 상황 요청을 받아, FCM을 통해 근처 유저들에게 실시간 알림을 전송하는 데 사용되었다.

FCM을 선택한 이유는 여러 가지가 있다. 첫째, 간편한 구현과 높은 신뢰성이다. 서버사이드 이벤트나 소켓 방식도 가능하지만, 이러한 방식은 지속적인 연결 유지를 필요로 하며, 상태 관리가 어려워 확장성 및 성능에 한계가 있을 수 있다. 반면, FCM은 클라우드 기반으로 메시징을 처리하며, 리소스 관리가 용이하고, 대규모 시스템에 적합하다. 또한, FCM은 서버 간 연결을 최소화하여

성능을 최적화할 수 있기 때문에 긴급 상황에서 유저에게 빠르게 알림을 전달할 수 있다. 단, FCM을 단순히 Spring Boot 서버 내에서 유저에게 순차적으로 전송하는 방식으로 구현할 경우, 동기 작업으로 인한 시간 지연이 발생하게 된다. 다음은 이러한 알림 서비스의 속도를 개선하기 위한 방법론들이다.

### 3.2.2 성능 최적화 - Spring Webflux & Kafka

알림 서비스의 전송 속도를 개선하기 위한 방법 중 하나로 Spring Webflux[2] 서버로 notification 기능을 분리하는 것을 선택하였다. Spring Webflux는 논블로킹 I/O를 지원하는 비동기 웹 프레임워크로, 높은 성능과 확장성을 제공한다. 이를 통해 notification 전송 기능의 성능을 최적화했다. 기존의 동기 방식으로 서버가 처리되는 동안 다른 요청들이 대기해야 했던 것과 달리, Webflux는 논블로킹 방식으로 병렬 처리를 통해 더 많은 요청을 동시에 처리할 수 있다.

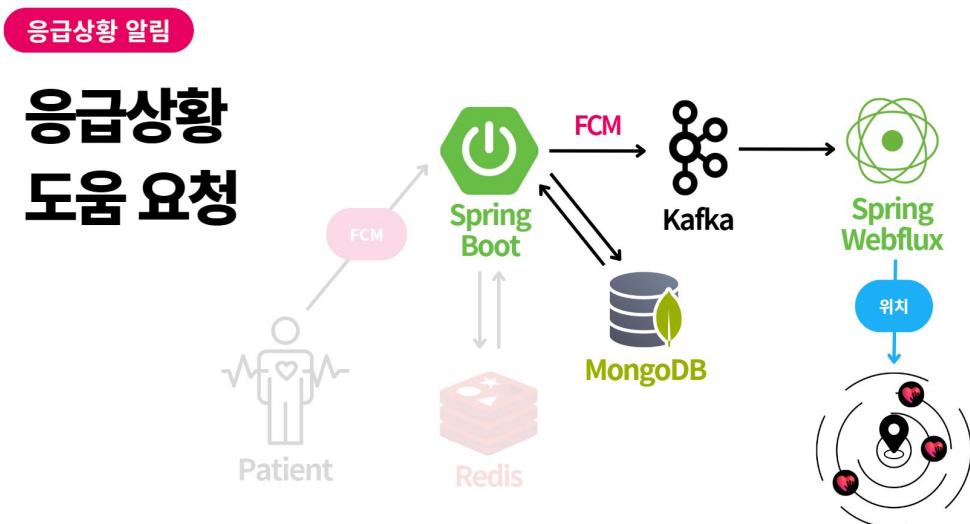


Figure 3.2: 성능 최적화 : webflux & Kafka

또한, Apache Kafka를 사용하여 서버 간 통신을 진행했다. Kafka는 분산 메시지 큐 시스템으로, 서버 간 데이터 전송을 처리할 때 HTTP 요청 방식에 비해 빠르고 안정적이다. Kafka의 고속 데이터 처리와 내결함성 덕분에 시스템의 확장성과 신뢰성을 높일 수 있었다. HTTP 방식은 동기적으로 요청을 처리하므로 병목 현상이 발생할 수 있지만, Kafka는 비동기적으로 메시지를

처리하고, 큐에 쌓인 메시지를 차례대로 처리할 수 있어 효율적인 데이터 처리와 성능 향상을 도모할 수 있다.

### 3.2.3 성능 최적화 - Redis & MongoDB

FCM의 전송 속도 외에도 얼마나 효율적으로 환자 주위의 헬퍼 유저를 찾을 것인가에 대한 최적화 역시 진행하였다. 이를 위해 Redis의 geospatial 기능을 활용해 빠르게 주위 유저를 찾는 것이 가능했다. Redis는 in-memory DB로 매우 빠른 데이터 접근 속도를 제공하며, 특히 위치 기반 서비스에 강점을 가진다. Redis는 유저들의 위치 정보를 GeoHash로 저장하고, 이를 바탕으로 가까운 유저를 빠르게 조회할 수 있다. 이에 대한 상세한 기술은 Appendix C에 기술하였다.

Redis는 메모리 기반으로 운영되기 때문에 대용량 데이터를 처리할 때 제한이 있다. 예를 들어, FCM 토큰과 같은 긴 정보를 Redis에 저장하는 것은 메모리 사용량에 부담을 줄 수 있다. 따라서 MongoDB를 사용하여 Redis에 저장된 간단한 키 값과 유저 위치 정보만 Redis에 저장하고, FCM 키와 같은 방대한 정보는 MongoDB에 저장했다. MongoDB는 기존 RDBMS에 비해 레플리카 셋과 데이터 색인을 쉽게 사용하는 것이 가능하며, 이를 통해 짧은 개발 시간 내에 확장성 및 안정성을 보장하는 기술을 구현할 수 있게 한다. 자세한 사항은 Appendix D를 통해 명시하였다.

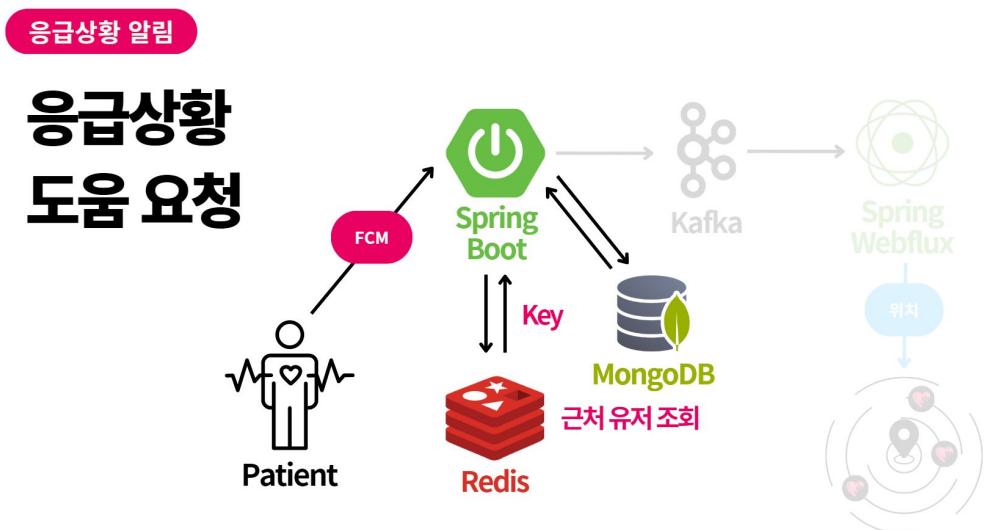


Figure 3.3: 성능 최적화 - Redis & MongoDB

이러한 MongoDB로의 분리는 단순히 안정성과 확장성만을 제공하는 것이 아닌 Redis내에 유저 식별 정보를 저장하지 않는 것으로, 설령 Redis DB의 내용이 탈취당하더라도 바로 유저를 판별할 수 없다는 점에서 보안성을 제공하는 역할도 존재한다.

### 3.2.4 성능 평가

성능 개선 전, 단순 FCM 알림 발송 시 200건의 알림 발송에 약 94초가 소요되었으며, 개선 후에는 54초로 단축되었다. Apache Kafka가 상대적으로 느린 메시지 큐 시스템인 만큼 성능 개선의 폭은 예상보다 크지 않았지만, 응급 서비스라는 특성을 고려할 때 정보 유실 방지와 높은 신뢰성 역시 우선시되었기에, Rabbit MQ와 같은 더 빠른 메시지 큐를 사용하는 대신 Apache Kafka를 채택하기로 결정하였다.

	성능 개선 전	성능 개선 후
200건 기준(s)	94 s	54 s
비교율(%)	100 %	57 %

Table 3.1: 알림 기능 성능 비교

### 3.3 모바일 - 모바일 통신 : BLE

본 서비스는 Bluetooth Low Energy (BLE) 기술을 사용하여 헬퍼가 환자의 위치를 통해 CPR 가이드를 시작할 때, 환자의 모바일 기기가 헬퍼의 모바일 기기를 스캔하고 자동으로 CPR 피드백 시스템을 트리거하는 기능이 존재한다.

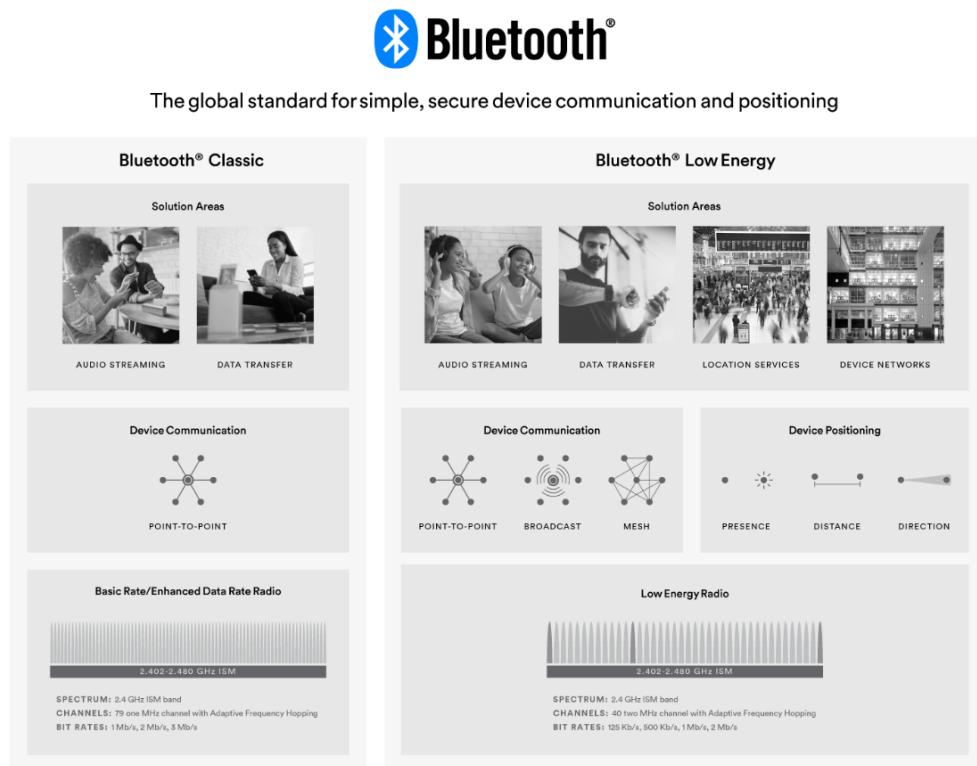


Figure 3.4: Bluetooth Classic vs BLE

#### 3.3.1 Bluetooth Low Energy(BLE)

Bluetooth Low Energy (BLE)는 저전력 소모로 짧은 거리에서 데이터를 전송하는 무선 통신 기술이다. BLE는 배터리 수명을 중요하게 여기는 웨어러블 기기에서 주로 사용된다. BLE는 전력 소비가 적어 장시간 사용이 가능하다. BLE는 Bluetooth Classic보다 최대 10배 더 적은 전력을 소비하며, 기기의 연결 시간이 짧고 빠른 데이터 전송을 제공한다. 해당 프로젝트에선 배터리 소모 최적화 및 피드백 트리거 전송을 위해 BLE를 채택하였다.

### 3.3.2 Bluetooth Classic vs BLE

Bluetooth Classic은 더 높은 데이터 전송 속도와 넓은 범위의 연결을 제공하지만, 전력 소비가 많아 배터리 수명이 짧다는 단점이 있다. 반면 BLE는 저전력 소모 덕분에 웨어러블 기기나 모바일 기기에서 장시간 작동이 가능하다. 이 서비스에서는 전력 소비를 최소화하고 실시간 통신이 중요한 요구사항이므로 BLE를 선택했다. BLE는 헬퍼와 환자 간의 실시간 연결에서 효율적으로 작동하며, CPR 피드백을 신속하게 제공할 수 있다.

## 3.4 위치 - 모바일 통신

본 서비스는 위치와 모바일 간의 같은 인터페이스와 정보를 공유해야하는 경우가 많다. 또한 모바일로 통신을 받았을 때 위치에서 자동으로 지도가 켜지는 등의 기능이 존재하므로, 사용자의 위치와 모바일간의 실시간 통신을 필요로 한다. 본 서비스는 이 기능을 MessageClient를 활용하여 구현하였다.

### 3.4.1 MessageClient

MessageClient는 위치와 모바일 간의 메시지 전송을 담당하는 클라이언트 시스템이다. 이 시스템은 실시간으로 데이터를 주고받을 수 있도록 설계되었으며, 알림 채널 분리를 통해 특정 정보가 들어왔을 때 위치와 모바일의 행동 기작을 조절하는 것이 가능하다.

### 3.4.2 프로젝트 내 사용된 알림 채널

프로젝트 내에선 크게 네 개의 알림채널이 사용 되었다. emergency\_alert, emergency\_location, feedback\_start, cpr\_start 이 그 네 가지이다.

\* **cpr\_start** : 헬퍼가 환자의 위치 화면을 탭하여 CPR을 시작했을 때, 환자의 위치에서 환자의 모바일로 BLE 통신을 시작하게 하는 채널

\* **feedback\_start** : 헬퍼의 모바일이 BLE 통신을 통한 트리거를 받았을 때 헬퍼의 위치에 피드백 기능이 작동하도록 통신을 보내는 채널

\* **emergency\_alert** : 환자의 위치에서 낙상 감지가 되거나, 심정지 위험이 감지된 경우, 모바일로 해당 사항을 전송하는 채널

\* **emergency\_location** : 헬퍼의 모바일이 구조 요청을 받았을 때, 위치에서 맵 렌더링을 위해 모바일로부터 환자의 위치 정보를 받는 채널

# Chapter 4

## 손목 각도 피드백 시스템

### 4.1 서비스 개요

심폐소생술(CPR)은 생명을 구하는 응급 처치 기술로, 정확한 자세와 적절한 압박 각도는 효과적인 CPR 수행에 필수적이다. CPR 중 팔이 지면과 수직(약 90도)으로 유지될 때 흉부 압박의 효율이 높아지므로, 팔 각도를 유지하는 것이 중요하다. 본 프로젝트(PushOfLife)는 Galaxy Watch와 같은 웨어러블 장치의 중력 센서와 선형 가속도 센서를 활용하여 CPR 시 팔 각도를 실시간으로 모니터링하고 피드백을 제공하는 시스템을 제안한다.

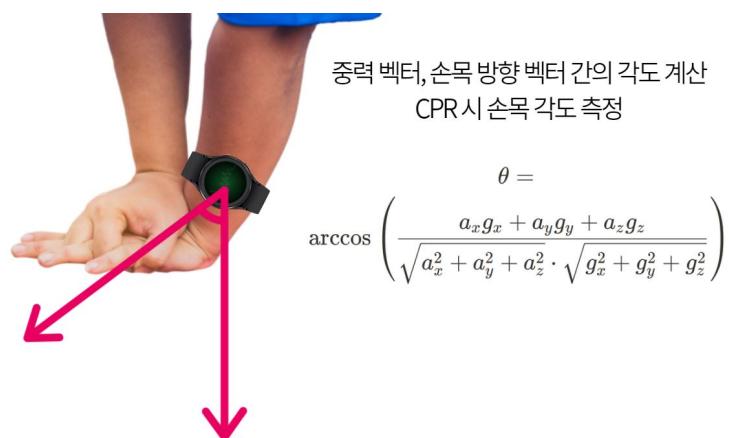


Figure 4.1: 손목 각도 피드백 시스템

## 4.2 이론적 배경

### 4.2.1 중력 가속도와 손목 방향 벡터

중력 센서는 장치가 지면과 이루는 방향을 측정하는 데 유용하며, 이 벡터 값은 중력 방향과 동일하다고 가정할 수 있다. CPR을 할 때 손목의 방향을 정확히 알기 위해서는 이 중력 벡터를 기준으로 한 손목 방향 벡터가 필요하다.

### 4.2.2 3D 벡터 내적과 각도 계산

팔의 각도를 측정하기 위해 3D 공간의 두 벡터, 즉 중력 벡터  $\vec{g}$ 와 선형 가속도 벡터  $\vec{a}$  사이의 각도를 계산한다. 벡터 내적을 이용하여 이 각도는 다음과 같이 구할 수 있다.

$$\cos \theta = \frac{\vec{a} \cdot \vec{g}}{|\vec{a}| |\vec{g}|} \quad (4.1)$$

여기서,  $\vec{a} = (a_x, a_y, a_z)$ 는 가속도 벡터이며,  $\vec{g} = (g_x, g_y, g_z)$ 는 중력 벡터이며,  $\vec{a} \cdot \vec{g} = a_x g_x + a_y g_y + a_z g_z$ 는 벡터 내적을 의미한다.  $|\vec{a}|$ 와  $|\vec{g}|$ 는 각각 벡터의 크기이며, 이는 각 축 값의 제곱합의 제곱근으로 계산된다.

$$|\vec{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (4.2)$$

$$|\vec{g}| = \sqrt{g_x^2 + g_y^2 + g_z^2} \quad (4.3)$$

따라서, 두 벡터의 각도는 다음과 같이 계산된다.

$$\theta = \arccos \left( \frac{a_x g_x + a_y g_y + a_z g_z}{\sqrt{a_x^2 + a_y^2 + a_z^2} \cdot \sqrt{g_x^2 + g_y^2 + g_z^2}} \right) \quad (4.4)$$

### 4.2.3 각도 측정을 통한 피드백 시스템

계산된 각도를 기반으로 CPR 수행 시 적절한 각도인지 실시간으로 평가할 수 있다. 예를 들어, 가 75도에서 105도 사이에 있을 때 적정한 각도로 간주하고, 그 외의 경우에는 사용자에게 피드백을 제공한다. 이를 통해 CPR 수행자의 자세가 유지되도록 돋는 효과를 기대할 수 있다.

## 4.3 구현 방법

### 4.3.1 시스템 설계

본 프로젝트에서는 Android Wear OS의 Galaxy Watch를 사용하여 중력 센서와 가속도 센서 데이터를 수집하고 분석하였다. 센서 값은 GravitySensor와 LinearAccelerationSensor 클래스를 통해 수집되며, 각각 중력 벡터와 가속도 벡터로 활용된다. 수집된 벡터 데이터를 CPRSensor 클래스로 전달하여 손목의 각도를 계산하고, 이를 통해 CPR에 적합한 피드백을 제공한다.

### 4.3.2 데이터 필터링

센서 데이터의 노이즈를 줄이기 위해 저주파 필터를 사용하였고, 보다 안정적인 각도 측정을 위해 칼만 필터(Kalman Filter)를 적용하여 수신된 신호를 보정하였다. 칼만 필터에 대한 자세한 설명은 Appendix B.CPR 피드백 시스템을 위해 사용된 필터 부분에 존재한다.

### 4.3.3 실시간 피드백

계산된 각도가 적절하지 않을 경우, 사용자에게 ”팔을 90도로 세워주세요”와 같은 음성 피드백을 제공한다. 이 시스템은 Galaxy Watch의 TTS(Text-to-Speech) 엔진을 이용하여 실시간으로 음성 피드백을 제공하도록 구현되었다.

## 4.4 결과

### 4.4.1 환경 및 과정

CPR 수행자의 팔 각도를 지속적으로 모니터링하며, 각도가 적절한지 여부를 측정하였다. 실험 대상자들은 다양한 각도에서 팔의 움직임을 수행하였으며, 시스템은 이 데이터를 바탕으로 각도 피드백을 실시간으로 제공하였다.

### 4.4.2 성능 평가

CPR을 처음 접하는 사용자들도 올바른 팔 각도를 유지하는데 도움이 되는 것으로 확인되었다. 특히 팔 각도가 적절하지 않은 경우에 대한 피드백은 응답 시간은 5초 이상으로, 실시간 피드백 시스템으로서의 유용성을 입증하였다.

# Chapter 5

## 압박 깊이 및 빈도 피드백 시스템

### 5.1 서비스 개요

심폐소생술(CPR)은 생명을 구하는 데 필수적인 기술로, 정확한 압박 깊이와 빈도는 심장과 폐에 충분한 혈액 순환을 유도하는 데 중요하다. CPR 수행 중 흉부를 5cm 깊이로 압박하는 것이 권장되며, 빈도는 1분에 100~120회가 적절하다. 이를 정확히 준수하기 위해 압박 깊이 및 빈도에 대한 실시간 피드백이 유용하다. 본 연구에서는 Wear OS를 탑재한 갤럭시 워치의 선형 가속도와 중력 가속도 센서를 사용하여 압박 깊이 및 빈도를 계산하는 방법을 제안한다.

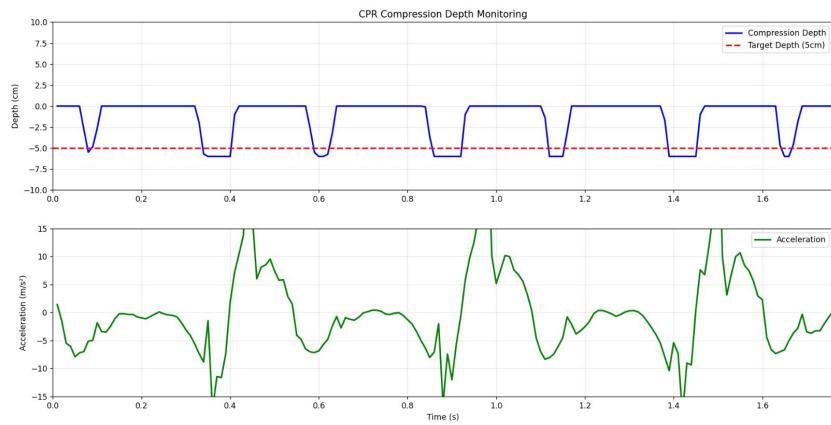


Figure 5.1: 올바른 CPR 수행시의 깊이 및 가속도 그래프

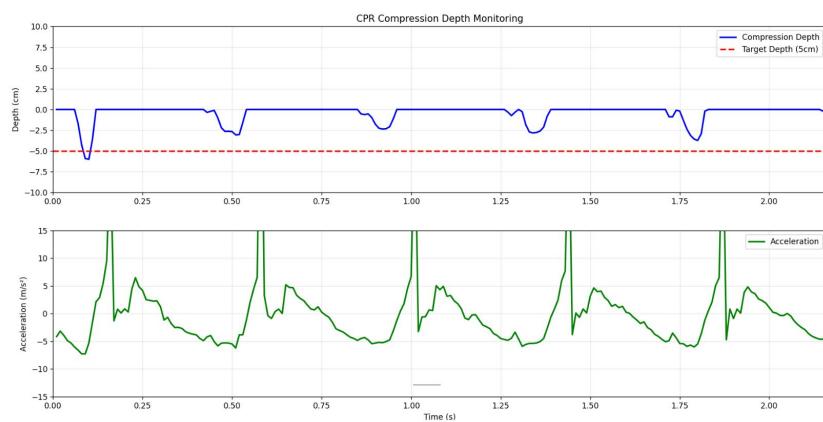


Figure 5.2: 올바르지 못한 CPR 수행시의 깊이 및 가속도 그래프

## 5.2 이론적 배경

### 5.2.1 선형 가속도와 중력 가속도의 역할

해당 서비스에서는 중력 센서와 선형 가속도 센서를 활용하여 선형 가속도와 중력 가속도를 측정하게 된다. 중력 센서는 위치의 수직 방향을 기준으로 중력 벡터  $\vec{g} = (g_x, g_y, g_z)$ 를 제공한다. 이는 CPR 압박 중 위치의 방향성을 정의하는 데 중요한 역할을 한다. 또한 선형 가속도 센서  $\vec{a} = (a_x, a_y, a_z)$ 는 중력을 제외한 운동에 따른 가속도만을 측정하여, 압박 시 위치가 수직 방향으로 움직이는 정도를 판단할 수 있게 한다.

### 5.2.2 중력 보정을 통한 수직 선형 가속도 계산

압박 깊이, 빈도를 계산하기 위해서는 선형 가속도의 수직 성분을 중력 벡터에 투영해야 한다. 이때 중력 보정된 수직 가속도  $a_{\text{vertical}}$ 은 다음과 같이 계산된다:

$$a_{\text{vertical}} = \frac{\vec{a} \cdot \vec{g}}{|\vec{g}|} \quad (5.1)$$

여기서,  $-\vec{a} \cdot \vec{g} = a_x g_x + a_y g_y + a_z g_z$ 는 선형 가속도 벡터와 중력 벡터 간의 내적이며,  $|\vec{g}| = \sqrt{g_x^2 + g_y^2 + g_z^2}$ 는 중력 벡터의 크기이다. 이를 통해 수직 방향 가속도  $a_{\text{vertical}}$ 을 계산하고, 이를 사용하여 압박 깊이를 유추할 수 있다.

## 5.3 구현 방법

### 5.3.1 데이터 수집 및 필터링

본 연구에서는 갤럭시 워치의 중력 센서와 선형 가속도 센서를 활용하여 CPR 수행 시 워치가 움직이는 수직 가속도를 실시간으로 측정하였다. 센서 데이터를 수집한 후, 안정적이고 정확한 압박 깊이, 빈도를 얻기 위해 저주파 필터(Low-pass Filter), 칼만 필터(Kalman Filter), 이동 평균 필터(Moving Average Filter), 적분 필터(Integration Filter)를 적용하였다. 이를 통해 노이즈를 제거하고 신뢰도 높은 데이터를 확보하였다.

### 5.3.2 가속도 적분을 통한 깊이 계산

수직 방향 가속도  $a_{\text{vertical}}$ 를 시간에 대해 적분하여 속도  $v(t)$ 를 구하고, 다시 적분하여 변위(압박 깊이)  $d(t)$ 를 구하는 과정을 통해 압박 깊이를 계산한다.

속도 계산 식은 다음과 같다.

$$v(t) = \int a_{\text{vertical}} dt \quad (5.2)$$

변위(압박 깊이) 계산 식은 다음과 같다.

$$d(t) = \int v(t) dt \quad (5.3)$$

적분 필터를 통해 압박 깊이를 구하며, 기타 필터를 적용하여 실시간 데이터의 신뢰성을 높였다.

### 5.3.3 깊이 추적을 통한 빈도 측정

실시간으로 계산되는 깊이를 통해 깊이 추적을 통해 얻음 사용자의 움직임을 토대로 압박 구간에 대한 시간차를 구하였다.

$$\Delta t = t_c - t_p$$

## 5.4 결과

### 5.4.1 실험 환경 및 과정

압박 깊이를 측정하기 위해, CPR 수행자의 팔이 수직으로 움직일 때 선형 가속도와 중력 가속도를 수집하였다. 각 압박 동작에 대해 수직 가속도를 적분하여 면위를 계산하였으며, 압박 깊이가 5cm 이상일 때 정확한 CPR 압박으로 간주하였다. 압박 깊이가 1cm를 충족하였을 경우를 기준으로 다음 압박 깊이가 1cm가 도달할 때 까지의 시간을 측정하여 0.5(s) ~ 0.6(s)라면 적절한 빈도로 간주하였다.

### 5.4.2 성능 평가

압박 깊이 피드백 시스템은 90% 이상의 정확도로 CPR 압박이 적절한 깊이인지 실시간으로 평가하였으며, 압박 깊이가 5cm 이하로 측정되면 ”더 깊게 압박하세요”와 같은 피드백을 제공하여 CPR 정확성을 개선하는 데 도움을 주었다. 빈도 피드백 시스템은 90% 이상의 정확도를 갖는 시간차로 CPR 압박이 가해지지 못하면 “빠릅니다.” 혹은 “느립니다.”와 같은 피드백을 제공한다. 여러 필터를 적용한 결과, 압박 깊이에 대한 노이즈를 줄이고 안정적인 피드백을 제공하는 데 유효한 성능을 보였다.

# Chapter 6

## 기대 효과

### 6.1 삼성헬스와의 협력 가능성

본 서비스는 삼성헬스와의 연동 가능성을 통해 사용자 개인의 건강 데이터를 응급 구조에 효과적으로 활용할 수 있는 기반을 제공한다. 사용자가 삼성헬스에 기록한 병력, 복용 약물, 알레르기 정보, 심박수 및 활동 데이터 등을 연동함으로써, 응급 상황 발생 시 헬퍼와 구급대가 환자의 상태를 신속히 파악할 수 있다. 이러한 데이터는 헬퍼와 의료진이 맞춤형 응급 조치를 수행하는데 있어 중요한 정보를 제공하며, 구조 활동의 정확성과 효율성을 높이는 데 기여한다.



Figure 6.1: 삼성헬스와의 협력시 기대효과

### 6.1.1 지역 응급 네트워크 형성

삼성헬스의 방대한 사용자 기반은 지역 응급 네트워크 형성에 있어 핵심적인 역할을 할 수 있다. 삼성헬스를 사용하는 많은 사람들이 참여함으로써, 응급 상황 발생 시 주변 헬퍼와 구급대가 신속하게 연결될 가능성이 높아진다. 특히, 심박수 이상이나 낙상 데이터가 실시간으로 공유되면, 구조 요청과 AED 위치 안내가 더욱 효율적으로 이루어진다.

삼성헬스 이용자 수의 규모는 지역 단위에서 응급 상황 데이터를 공유하고 분석할 수 있는 잠재력을 제공한다. 이를 통해 응급 구조 요청이 빠르게 이루어지고, 골든타임 내 구조 성공률을 높이는 데 기여할 것으로 기대된다. 이러한 네트워크는 사용자 기반이 클수록 효과가 커져, 지역 사회 전반의 응급 대응력을 강화할 수 있다.

### 6.1.2 골든타임 확보에 기여

예를 들어, 심장질환 병력이 있는 환자가 심정지 상황에 처했을 경우, 삼성헬스에서 제공받은 데이터를 바탕으로 즉시 AED 사용이 권장될 수 있다. 또한, 약물 알레르기 정보는 구급대가 응급 처치를 시행할 때 안전성을 확보하는 데 중요한 역할을 한다. 이러한 맞춤형 데이터 활용은 구조 과정에서 발생할 수 있는 시행착오를 줄이고, 응급 상황에서 골든타임을 확보하는 데 결정적인 도움을 준다.

### 6.1.3 안전한 의료 환경 제공

나아가, 건강 데이터의 연동은 단순히 응급 구조에서 끝나는 것이 아니라, 사후 관리와 예방적 조치를 위한 데이터 환경 조성에도 기여할 수 있다. 구조 이후 환자와 의료진은 응급 상황에서의 데이터를 검토하고, 이를 바탕으로 재발 가능성을 줄이기 위한 건강 관리 방안을 설계할 수 있다. 이러한 데이터 기반 접근은 장기적인 건강 관리를 지원하며, 사용자에게 더욱 안전한 의료 환경을 제공할 것으로 기대된다.

## 6.2 생존율 향상

병원 도착 전 심폐소생술(CPR)의 시행 여부는 환자의 생존율과 뇌기능 회복률에 중대한 영향을 미친다. 기존 연구에 따르면, 일반인이 병원 도착 전에 CPR을 시행한 경우 생존율은 14.0%로, 시행하지 않은 경우의 8.2%에 비해 약 1.7배 높은 수치를 보였다. 또한, 뇌기능 회복률은 CPR을 시행한 경우 10.0%로, 시행하지 않은 경우의 4.6%보다 약 2.2배 높은 결과를 나타냈다. 이러한 데이터는 CPR의 시행이 생존 가능성과 신경학적 회복 가능성을 모두 크게 향상시킬 수 있음을 시사한다.

	CPR 시행한 경우	CPR 시행하지 않은 경우
생존율	14.0 %	8.2 %
뇌 기능 회복률	10.0 %	4.6 %

Table 6.1: CPR 시행 여부에 따른 생존율 및 회복률 비교

### 6.2.1 자동화된 CPR 가이드

본 서비스는 자동화된 CPR 가이드와 실시간 피드백 기능을 통해 일반인도 신속하고 올바르게 CPR을 수행할 수 있도록 지원한다. CPR의 각도, 빈도, 깊이에 대한 실시간 피드백과 메트로놈 리듬 안내는 구조자가 적절한 속도와 강도로 CPR을 지속할 수 있도록 돋는다. 이러한 기능은 구조의 질을 높이는 동시에, 구조자가 CPR을 시행하는 데 필요한 심리적 부담을 완화한다.

### 6.2.2 긴급 알림을 통한 지연 시간 최소화

또한, 낙상 및 심박수 이상 감지 시 주변 사용자에게 긴급 알림을 전송하여 CPR 시행의 지연 시간을 최소화한다. 헬퍼가 현장에 도착하면 환자의 실시간 심박수 및 상태 데이터를 공유받아 보다 정확하고 효과적인 응급 처치를 수행할 수 있다. 이는 일반인의 CPR 시행률을 높이고, 병원 도착 전에 구조가 이루어질 가능성을 크게 증가시킨다.

결과적으로, 본 서비스는 병원 도착 전 CPR 시행률을 높이고, 이를 통해 환자의 생존율과 뇌기능 회복률을 개선하는 데 크게 기여할 것으로 기대된다. 이는 골든타임 내 구조의 효과를 극대화하고, 응급 구조 품질을 향상시키며, 더 많은 생명을 구하는 데 중추적인 역할을 할 것이다.

# Appendix A

## 화면 정의서

본 화면 정의서는 스마트폰과 워치 디바이스에서 사용자 인터페이스(UI)와 사용자 경험(UX) 요소들을 설계한 문서이다. 시스템의 각 화면에 대한 상세한 설명과 함께, 관련된 화면 자료가 포함되어 있다.

본 장은 스마트폰 화면과 워치 화면을 나누어져 있다. 각 화면은 해당 기능을 제공하기 위한 주요 인터페이스를 포함하고 있다. 스마트폰 화면에서는 사용자 등록, 로그인, 정보 추가, 메인 화면과 같은 기본적인 상호작용을 다루며, 응급 상황에 대한 정보 제공 및 CPR, AED 가이드 등의 중요한 기능이 구현되어 있다. 워치 화면에서는 응급환자 신고, 낙상 감지, CPR 가이드 등의 기능을 간결하게 보여준다.

## A.1 스마트폰 화면



제목		회원가입 화면
		설명
1		전화번호 입력
2		비밀번호 입력
3		비밀번호 확인 입력
4		회원가입 버튼

Figure A.1: 회원가입 화면



제목		로그인 화면
		설명
1		전화번호 입력
2		비밀번호 입력
3		로그인 버튼

Figure A.2: 로그인 화면



Figure A.3: 정보추가 화면

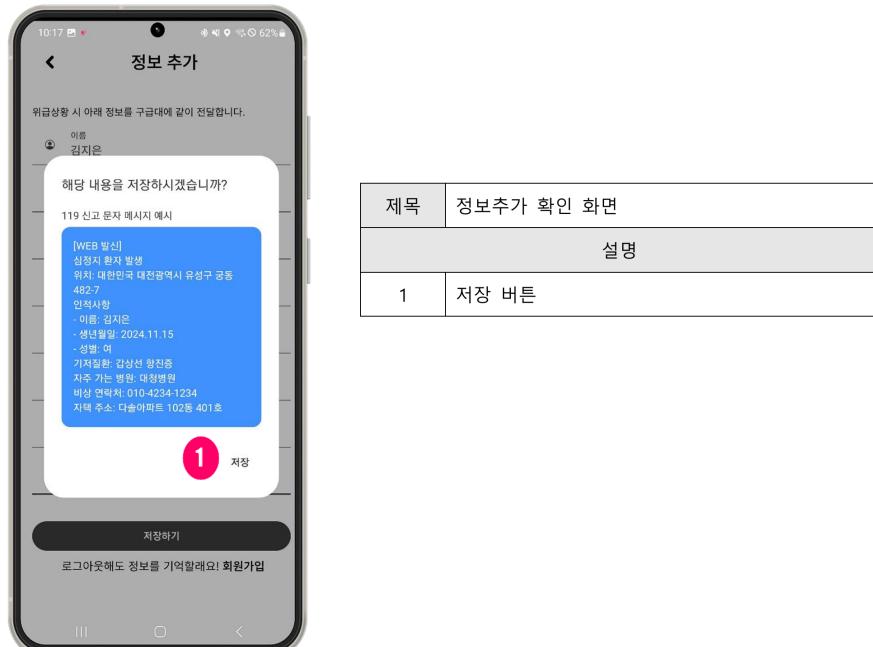
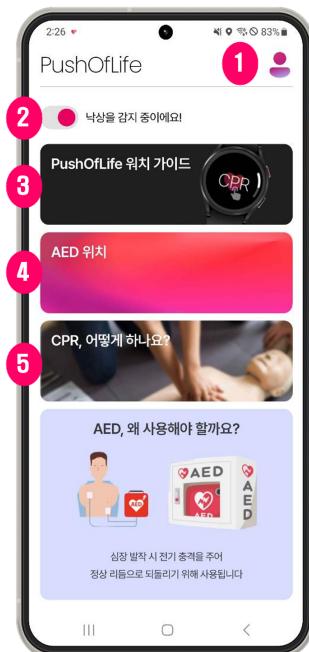


Figure A.4: 정보추가 확인 화면



제목	메인 화면(워치 없을 경우)
설명	
1	추가 정보 입력 이동 버튼
2	낙상 감지 토글
3	위치 가이드 이동 버튼
4	AED 위치 이동 버튼
5	CPR가이드 이동 버튼

Figure A.5: 메인 화면(워치가 없는 경우)



제목	메인 화면
설명	
1	추가 정보 입력 이동 버튼
2	워치 가이드 이동 버튼
3	AED 위치 이동 버튼
4	CPR가이드 이동 버튼

Figure A.6: 메인 화면



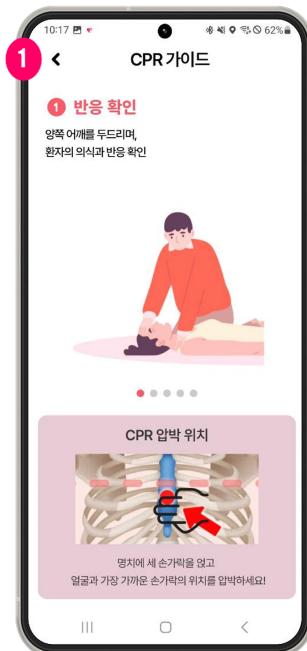
제목	메인 화면(주변에 응급환자가 있을 경우)
설명	
1	추가 정보 입력 이동 버튼
2	응급환자 및 AED위치 이동 버튼
3	위치 가이드 이동 버튼
4	AED 위치 이동 버튼
5	CPR가이드 이동 버튼

Figure A.7: 메인 화면(주변에 응급환자가 있을 경우)



제목	위치 가이드 화면
설명	
1	메인화면 이동 버튼

Figure A.8: 위치 가이드 화면



제목	CPR 가이드 화면
설명	
1	메인화면 이동 버튼

Figure A.9: CPR 가이드 화면



제목	AED 위치 화면
설명	
1	현재 위치
2	AED 위치 버튼
3	현재 위치로 이동 버튼

Figure A.10: AED 위치 화면

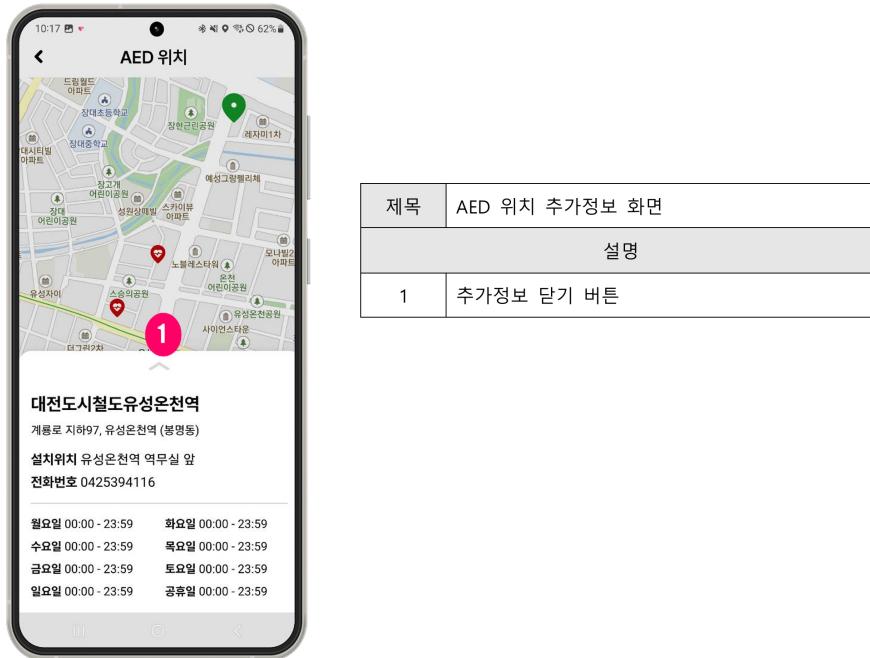


Figure A.11: AED 위치 추가정보 화면

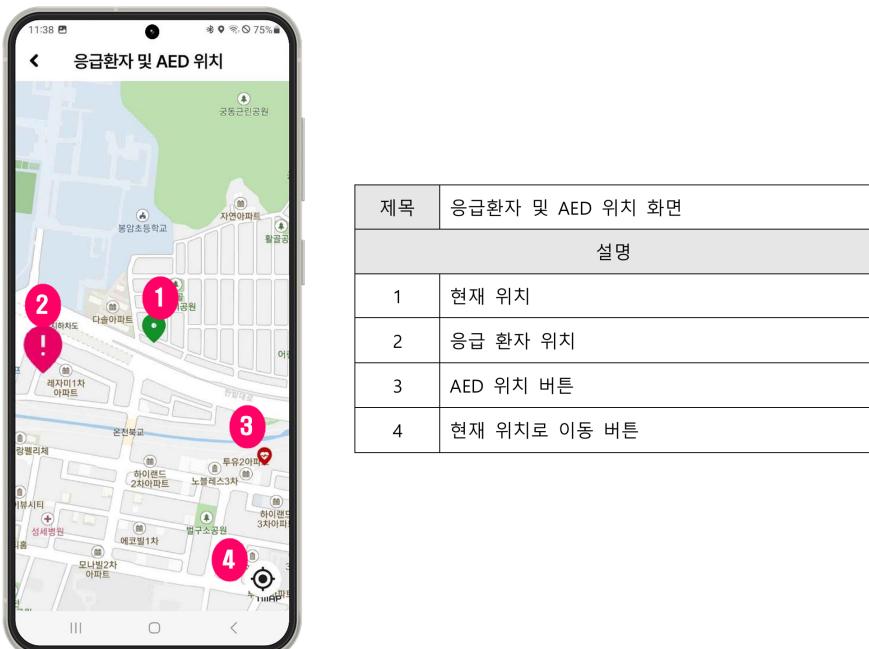


Figure A.12: 응급환자 및 AED 위치 화면

## A.2 워치 화면



제목	메인 화면
설명	
1	CPR 가이드 시작 버튼(환자 발견 시)

Figure A.13: 메인 화면 - 1



제목	메인 화면
설명	
1	CPR 연습 시작 버튼
2	AED 위치 이동 버튼

Figure A.14: 메인 화면 - 2



Figure A.15: 신고 화면



제목	낙상 감지 화면
설명	
1	환자 CPR가이드 화면으로 이동
2	메인화면으로 이동

Figure A.16: 낙상 감지 화면



제목	CPR 가이드 헬퍼 화면
설명	
1	CPR 피드백에 따라 색 변경
2	메인화면으로 이동

Figure A.17: CPR 가이드 헬퍼 화면



제목	CPR 가이드 환자 화면
설명	
1	심박수 측정 화면
2	메인화면으로 이동

Figure A.18: CPR 가이드 환자 화면



제목	응급환자 및 AED 위치화면
설명	
1	현재 위치
2	응급환자 위치
3	AED 위치

Figure A.19: 응급환자 및 AED 위치 화면

## Appendix B

# CPR 피드백 시스템을 위해 사용된 필터

### B.1 저주파 통과 필터 (Low-pass Filter)

가속도 데이터를 필터링해 고주파 노이즈를 제거하고, 안정적인 신호를 얻기 위해 사용한다. 예를 들어, 가속도 데이터를 처리할 때 갑작스러운 진동이나 외부 충격에 의해 발생한 노이즈를 줄이는 데 유용하다. 가속도 데이터에서 중력 성분을 추출하거나, 특정 주파수 이하의 신호만 남기고 싶은 경우에 효과적이다. [3]

$$filteredAcc = rawAcc + (1 - \alpha) filteredAcc \quad (\text{B.1})$$

### B.2 칼만 필터 (Kalman Filter)

예측 단계에서는 이전 상태와 시스템 모델을 바탕으로 현재 상태를 예측한다. 주로 주어진 시스템 모델을 사용하여 상태와 공분산을 예측한다. 상태 전이 행렬  $F$ 는 시스템이  $dt$  동안 어떻게 변화하는지를 나타내는 행렬, 예를 들어, 위치와 속도 상태가 주어졌을 때,  $F$ 는 다음과 같이 정의된다. [8]

$$F = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \quad (\text{B.2})$$

예측된 상태는 현재 상태와 상태 전이 행렬  $F$ 의 곱으로 계산된다.

$$predictedState = F state \quad (\text{B.3})$$

이는 이전 상태에서 현재 상태로 변화할 때 시스템의 동작을 반영하는 계산이다. 예측 공분산  $P$ 는 상태 추정의 불확실성을 나타내며, 예측 과정에서도 업데이트 된다. 예측 공분산은 다음과 같이 계산된다

$$\text{predictedP} = F \cdot P \cdot F^T + Q \quad (\text{B.4})$$

여기서  $Q$ 는 프로세스 노이즈 공분산 행렬로, 시스템의 예측에 대한 불확실성을 반영한다. 이 계산을 통해 예측된 상태의 불확실성을 조정한다.

### B.2.1 업데이트 단계

업데이트 단계에서는 측정된 값을 바탕으로 예측된 상태를 보정한다. 이를 통해 추정의 정확성을 높일 수 있다. 관측 모델  $H$ 는 측정 가능한 상태와 예측된 상태 간의 관계를 나타내는 행렬이다. 여기서는 단순히 위치와 속도를 측정한다고 가정하므로, 다음과 같다.

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{B.5})$$

예측 오차 공분산  $S$ 는 측정과 예측된 상태 간의 불확실성을 나타내며, 칼만 이들을 계산하기 위한 중요한 요소이다. 다음과 같이 계산된다.

$$\text{predictedP} = F \cdot P \cdot F^T + Q \quad (\text{B.6})$$

여기서  $R$ 은 측정 노이즈 공분산 행렬로, 측정 데이터에 포함된 노이즈의 불확실성을 나타낸다. 칼만 이득  $K$ 는 예측된 상태를 측정 데이터로 얼마나 보정할지 결정하는 요소이다. 칼만 이득은 다음과 같이 계산된다.

$$K = \text{predictedP} \cdot S^{-1} \quad (\text{B.7})$$

$$K = P_{\text{pred}} H^T (H P_{\text{pred}} H^T + R)^{-1} \quad (\text{B.8})$$

칼만 이득이 높으면 측정 데이터의 영향이 커지고, 낮으면 예측 상태의 영향이 커지게 된다. 이는 시스템의 노이즈 특성에 따라 달라진다. 다음 그림B.1은 파라미터  $Q, R$ 을 구하기 위한 완전 탐색의

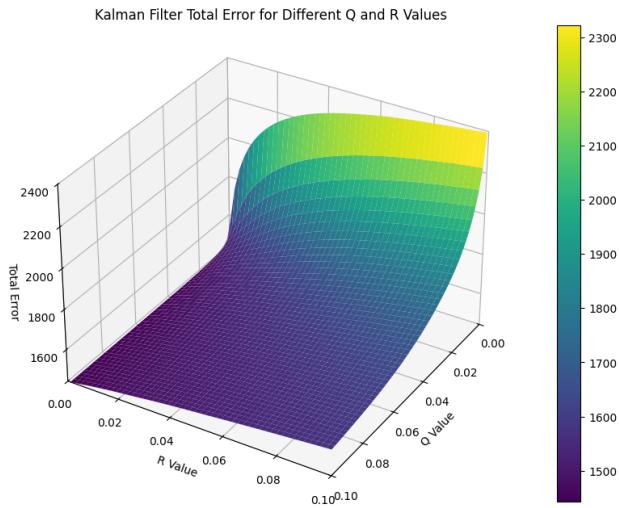


Figure B.1: 가능한 Q,R에 대한 완전 탐색을 통해 시각화한 그래프

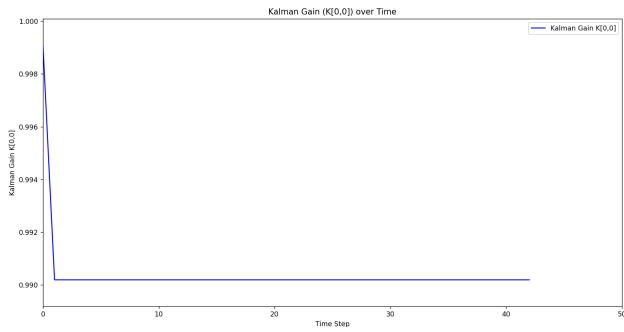


Figure B.2: 시스템 안정성 모니터링 이미지

시각화 자료이다.

선정된 파라미터 행렬을 통해 시스템이 안정적으로 가동되고 있는지에 대한 모니터링을 그림B.2과 같이 진행할 수 있다.

추가적으로, 보정된 상태는 예측된 상태에 칼만 이득과 측정 오차(측정값 - 예측값)를 곱하여 더해줌으로써 계산된다.

$$\text{state} = \text{predictedState} + K \cdot (\text{measurement} - H \cdot \text{predictedState}) \quad (\text{B.9})$$

마지막으로, 보정된 공분산  $P$ 를 업데이트하여 새로운 상태 추정의 불확실성을 조정할 수 있다.

$$P = (I - K \cdot H) \cdot \text{predictedP} \quad (\text{B.10})$$

여기서  $I$ 는 단위 행렬로, 업데이트된 공분산은 상태 추정이 얼마나 신뢰할 만한지를 나타낸다.

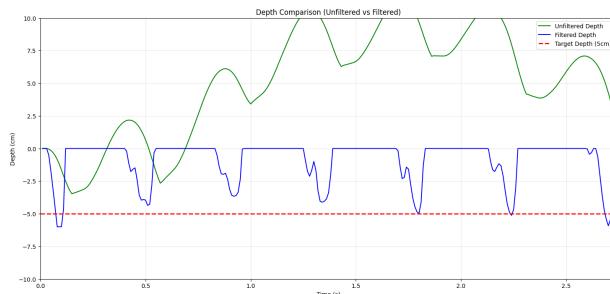


Figure B.3: filtered depth, raw depth 비교 그래프

이를 통해 다음 예측 단계에서 더 정확한 상태 예측이 가능해진다.

### B.3 이동 평균 필터

센서 데이터의 노이즈를 줄이고 안정적인 추세를 제공하기 위해 사용된다. 이 필터는 일정 개수의 최근 데이터의 평균을 계산하여 갑작스러운 변화나 불규칙한 변동을 완화하며, 실시간으로 간단하고 빠르게 적용할 수 있다. 이를 통해 CPR 압박 깊이와 같은 중요한 데이터에서 신뢰할 수 있는 값을 지속적으로 얻는 데 유용하다. [7]

$$\text{filteredDepth} = \frac{1}{N} \sum_{i=1}^N d_i \quad (\text{B.11})$$

### B.4 적분 필터 (Trapezoidal Integration)

주어진 가속도 데이터를 적분해 속도를 구하고, 속도를 적분해 위치를 구한다. 이때 단순하게 데이터를 적분할 경우 누적 오차가 커질 수 있어, 트라페조이드 방법(사다리꼴 적분법)과 같은 적분 방식을 적용해 정확도를 높일 수 있다.

$$v = v_{\text{previous}} + \frac{a_{\text{previous}} + a_{\text{current}}}{2} \cdot \Delta t \quad (\text{B.12})$$

## Appendix C

# GeoSpatial & GeoHash

본 장에서는 Redis의 GeoSpatial 기능과 GeoHash 기술에 대해 논의하고, 이를 통해 공간 데이터를 처리하고 관리하는 방법에 대한 설명을 진행한다. Redis는 인메모리 데이터베이스로, 빠른 읽기 및 쓰기 성능을 제공하며, 최근에는 GeoSpatial 데이터 처리에 대한 지원을 강화하고 있다. GeoSpatial 기능은 위치 기반 서비스, 거리 계산 및 범위 검색 등에 유용하며, GeoHash는 공간 데이터의 효율적인 저장과 검색을 위해 활용되는 기술이다.

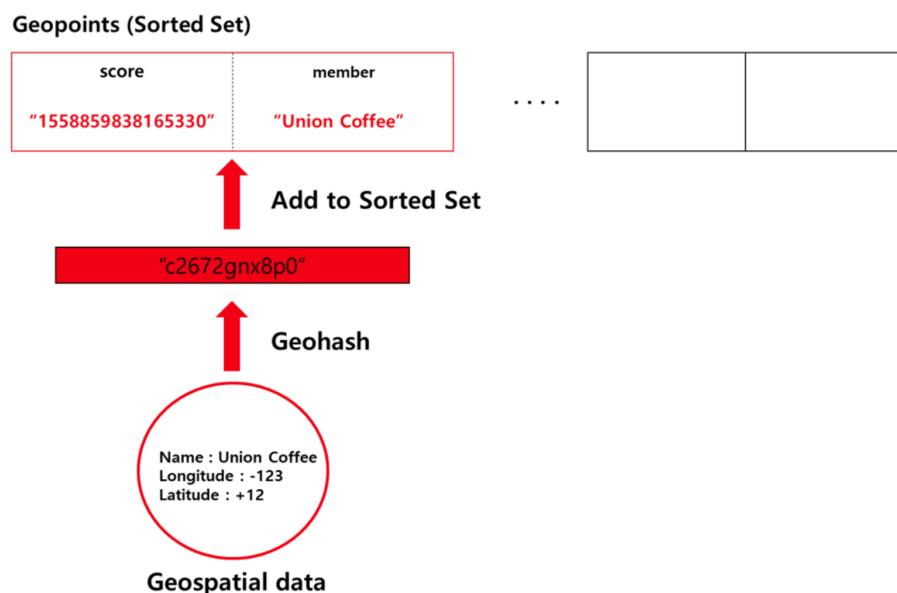


Figure C.1: Redis GeoSpatial[4]

## C.1 GeoSpatial

Redis의 GeoSpatial 기능은 공간 데이터를 저장하고 처리하기 위해 몇 가지 명령어를 제공한다. Redis에서 GeoSpatial 데이터는 지리적 좌표(위도와 경도)로 표현되며, 주로 GEOADD, GEODIST, GEORADIUS, GEORADIUSBYMEMBER와 같은 명령어를 사용하여 데이터를 다룬다. GeoSpatial 데이터는 Redis의 기본 데이터 구조인 Sorted Set을 기반으로 저장된다. 이 구조는 각 지리적 좌표를 위치 값과 함께 저장하고, 이를 빠르게 검색하고 거리 계산을 할 수 있도록 돋는다.

### C.1.1 GEOADD

GEOADD 명령어는 하나 이상의 좌표를 Redis에 추가하는 데 사용된다. 좌표는 위도, 경도와 함께 사용자 정의 키에 저장된다. 예를 들어, 특정 도시나 장소의 위치 정보를 저장할 때 유용하다. 본 서비스에서 사용자의 위치를 저장하기 위해 해당 기능을 사용하였다.

### C.1.2 GEODIST

GEODIST 명령어는 두 지점 사이의 거리를 계산하는 데 사용된다. 거리 단위는 미터, 킬로미터, 마일, 피트로 지정할 수 있다.

### C.1.3 GEORADIUS

GEORADIUS 명령어는 특정 지점으로부터 일정 반경 내에 있는 모든 지리적 위치를 검색할 수 있다. 이 명령은 특정 좌표를 중심으로 일정 범위 내의 모든 좌표를 효율적으로 찾아낼 수 있도록 한다.

### C.1.4 GEORADIUSBYMEMBER

GEORADIUSBYMEMBER 명령어는 특정 유저로부터 일정 반경 내에 있는 모든 지리적 위치를 검색할 수 있다. 이 명령은 특정 유저를 중심으로 일정 범위 내의 모든 좌표를 효율적으로 찾아낼 수 있도록 하기에 서비스 내에서 환자 주위의 헬퍼를 찾는 방식으로 채택하였다.

## C.2 GeoHash

GeoHash는 공간을 일정한 크기의 격자로 나누어 공간을 해시 값으로 표현하는 알고리즘이다. 이 알고리즘은 공간 데이터를 2D 좌표를 1D 문자열로 변환하여 저장하는 방식으로, 주로 공간 데이터베이스에서 효율적인 검색을 위해 사용된다. GeoHash는 각 좌표를 고정 길이의 문자열로 변환하여, 인덱싱 및 검색을 빠르게 처리할 수 있도록 돋는다. GeoHash에 대한 실습을 진행하고 싶다면, 다음 사이트(<https://geohash.softeng.co/>)를 추천한다.

### C.2.1 GeoHash의 동작 원리

GeoHash는 좌표(위도, 경도)를 이진수로 변환한 후, 이들을 교차 방식으로 결합하여 문자열로 변환한다. 각 문자열의 길이는 지정된 정밀도에 따라 달라지며, 정밀도가 높을수록 더 작은 구역을 나타낸다.



Figure C.2: GeoHash 동작 원리

### C.2.2 GeoHash의 장점

\* **효율성** : GeoHash는 공간 데이터를 1D 문자열로 변환하여 저장하고, 이를 빠르게 검색할 수 있다.

\* **인접성** : GeoHash는 근처의 좌표들이 비슷한 해시 값을 가지므로, 인접한 좌표들을 쉽게

찾을 수 있다.

\* 범위 검색 : GeoHash는 범위 검색을 효율적으로 수행할 수 있어, 지정된 범위 내의 좌표를 빠르게 찾을 수 있다.

# Appendix D

## MongoDB Sharding

MongoDB는 높은 성능과 확장성으로 널리 사용되는 NoSQL 데이터베이스 관리 시스템이다. 그러나 데이터베이스의 크기가 증가함에 따라 단일 서버에 데이터가 모두 저장될 경우 성능 저하와 확장성 문제를 겪을 수 있다. 이를 해결하기 위해 MongoDB는 샤딩(Sharding) 기술을 사용하여 데이터를 여러 서버에 분산시킨다. 샤딩은 데이터의 수평적 분할을 통해 데이터베이스의 처리 성능을 향상시키고, 대규모 데이터를 효율적으로 처리할 수 있게 만든다. 이 장에서는 MongoDB의 샤딩에 대해 설명한다.

### D.1 Sharding

샤딩은 데이터베이스의 데이터를 여러 서버에 분배하는 기술로, 대규모 데이터베이스의 확장성 문제를 해결한다. MongoDB에서 샤딩은 데이터를 여러 ”샤드(Shard)”로 나누고 각 샤드는 독립적인 MongoDB 서버로 구성된다. 각 샤드는 데이터의 일부를 저장하며, 전체 데이터를 여러 샤드에 분산시켜 처리 성능을 향상시킨다. MongoDB의 샤딩은 다음과 같은 주요 요소로 구성된다.

#### D.1.1 Mongos

Mongos는 클라이언트 애플리케이션과 샤드 서버 사이의 라우터 역할을 하는 컴포넌트를 의미한다. 본 서비스 내에서는 Spring Boot 서버와 샤드 서버 사이의 라우터 역할을 담당하였다. Mongos는 클라이언트의 요청을 받아 적절한 샤드로 요청을 전달한다.

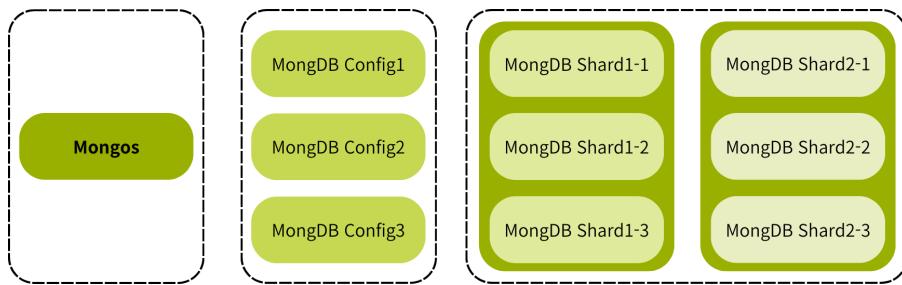


Figure D.1: Sharding Structure

### D.1.2 Config Server

샤딩된 데이터의 메타데이터를 저장하는 서버이다. Config 서버는 샤드의 위치와 데이터를 어디에 저장할지에 대한 정보를 관리한다. 본 서비스에선 Config Server를 세개의 Config Server Replica Set으로 구성하여, 설령 Master의 Config Server가 Shutdown 되더라도, Slave가 그 일을 이어받을 수 있도록 구성하였다.

### D.1.3 Shard

데이터를 저장하는 실제 서버를 의미한다. 각 샤드는 데이터의 부분 집합을 포함하고 있으며, 샤드마다 독립적인 MongoDB 인스턴스가 실행된다. 본 서비스에서는 3개의 Replica Set으로 구성된 Shard Server 2개를 활용하여 Shard를 구성하였다.

## 참고 문헌

- [1] Misuk An, Youngmee Kim, and Won-Kyung Cho. Effect of smart devices on the quality of cpr training: A systematic review. *Resuscitation*, 144:145–156, 2019.
- [2] Marten Deinum, Daniel Rubio, and Josh Long. Spring webflux. In *Spring 6 Recipes: A Problem-Solution Approach to Spring Framework*, pages 205–240. Springer, 2023.
- [3] Jonatan Fridolfsson, Mats Börjesson, Christoph Buck, Örjan Ekblom, Elin Ekblom-Bak, Monica Hunsberger, Lauren Lissner, and Daniel Arvidsson. Effects of frequency filtering on intensity and noise in accelerometer-based physical activity measurements. *Sensors*, 19(9):2186, 2019.
- [4] Wonyong Jang. [redis] geospatial 자료구조, 2021. [Redis] Geospatial 자료구조.
- [5] Elke Mackensen, Matthias Lai, and Thomas M Wendt. Bluetooth low energy (ble) based wireless sensors. In *SENSORS, 2012 IEEE*, pages 1–4. IEEE, 2012.
- [6] Greg Milette and Adam Stroud. *Professional Android sensor programming*. John Wiley & Sons, 2012.
- [7] Vishakha Pandey and VK Giri. High frequency noise removal from ecg using moving average filters. In *2016 International conference on emerging trends in electrical electronics & sustainable energy systems (ICETEESES)*, pages 191–195. IEEE, 2016.
- [8] Sangheon Park, Sunjin Yu, Joongrock Kim, Sungjin Kim, and Sangyoun Lee. 3d hand tracking using kalman filter in depth space. *EURASIP Journal on Advances in Signal Processing*, 2012:1–18, 2012.

- [9] Keun Jeong Song and Dong Jin Oh. Current status of cpr in korea. *The Korean journal of medicine*, 73(1):4–10, 2007.
- [10] Neha Srivastava, Uma Shree, Nupa Ram Chauhan, and Dinesh Kumar Tiwari. Firebase cloud messaging (android). *International Journal of Innovative Research in Science, Engineering and Technology,(An ISO 3297: 2007 Certified Organization)*, 6, 2017.
- [11] 류정탁. 3 축 가속도센서와 기울기 센서를 이용한 낙상감지시스템 개발. 한국산업정보학회 논문지, 18(4):19–24, 2013.
- [12] 이윤지, 이초롱, 임연희, 조민희, 조연경, 조준희, 진주실, 김진아, 안성아, and 김은희. 대학생 심폐소생술 교육의 중요성 및 교육 실태에 관한 연구. *한국임상보건과학회지*, 1(1):47–61, 2013.
- [13] 질병관리청. 2022 급성 심장정지 조사 통계, 2024. 질병관리청, 2022년 급성 심장정지 통계 보고서.