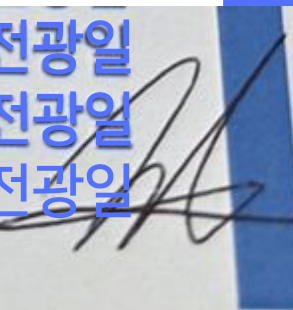


스마트글래스를 활용한 AR 피트니스 보조 서비스

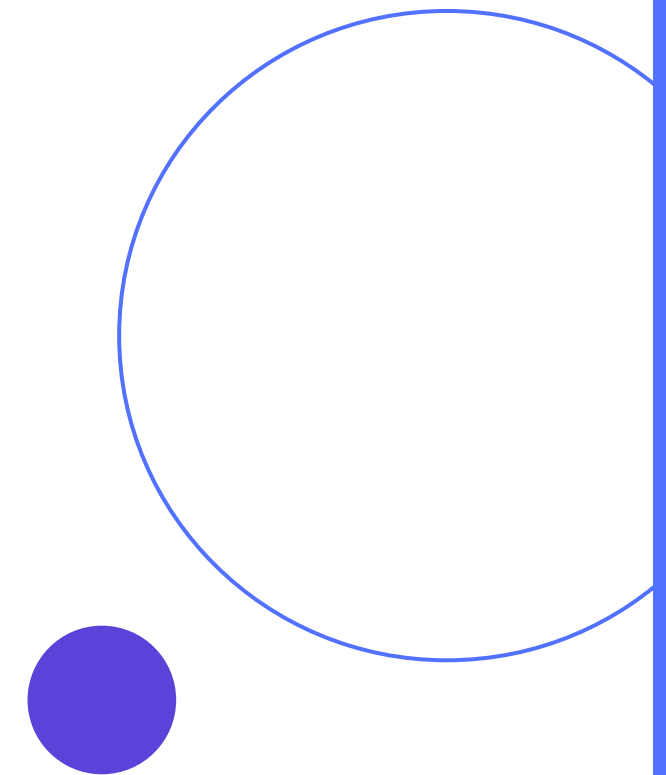
AR FITNESS ASSISTANCE SERVICE
WITH SMART GLASSES

2021152027 이예권 지도교수 전광일
2022150005 김동일 지도교수 전광일
2019150028 양동균 지도교수 전광일



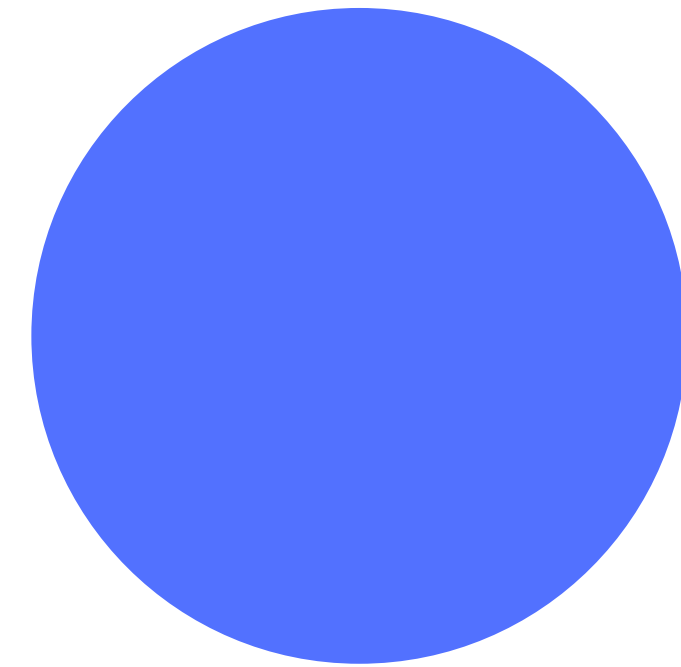
목차

- 종합 설계 개요
- 시스템 수행 시나리오
- 시스템 구성도
- 시스템 모듈 상세 설계
- 업무 분담
- 종합 설계 수행 일정
- 필요기술 및 참고문헌
- 데모 환경 설계



지난 발표에서의 지적 사항

1. 운동 1~2개 타겟팅
2. 기존 영상에 대한 각도 및 스마트폰 2개를 사용해서 동시에 출력
3. 시나리오와 구성도의 괴리감



지적 사항에 대한 답변

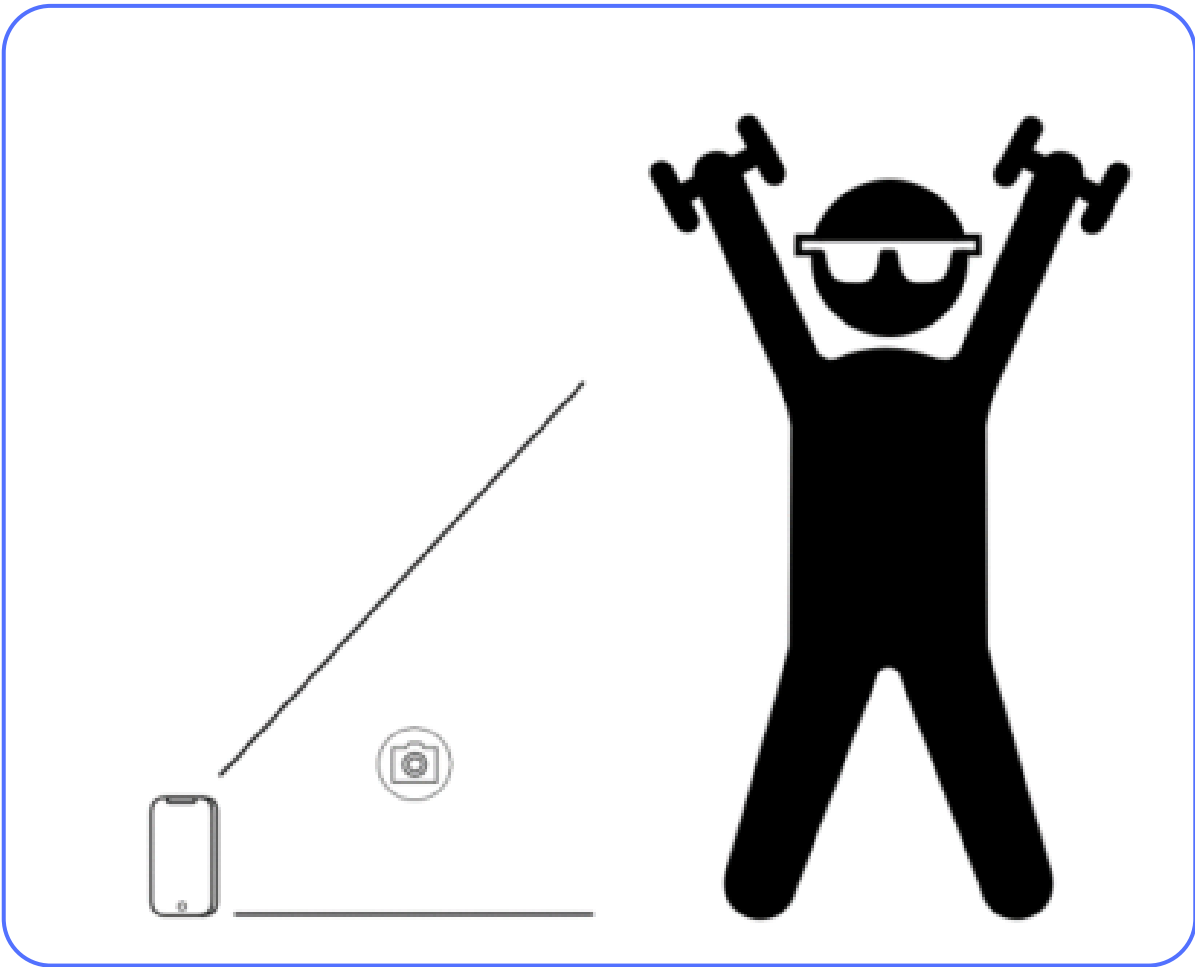
1. 푸쉬업, 플랭크, 런지로 운동 타겟팅
2. 스마트폰 2개 사용해서 다방면 각도로 진행
3. 시스템 수행 시나리오 수정



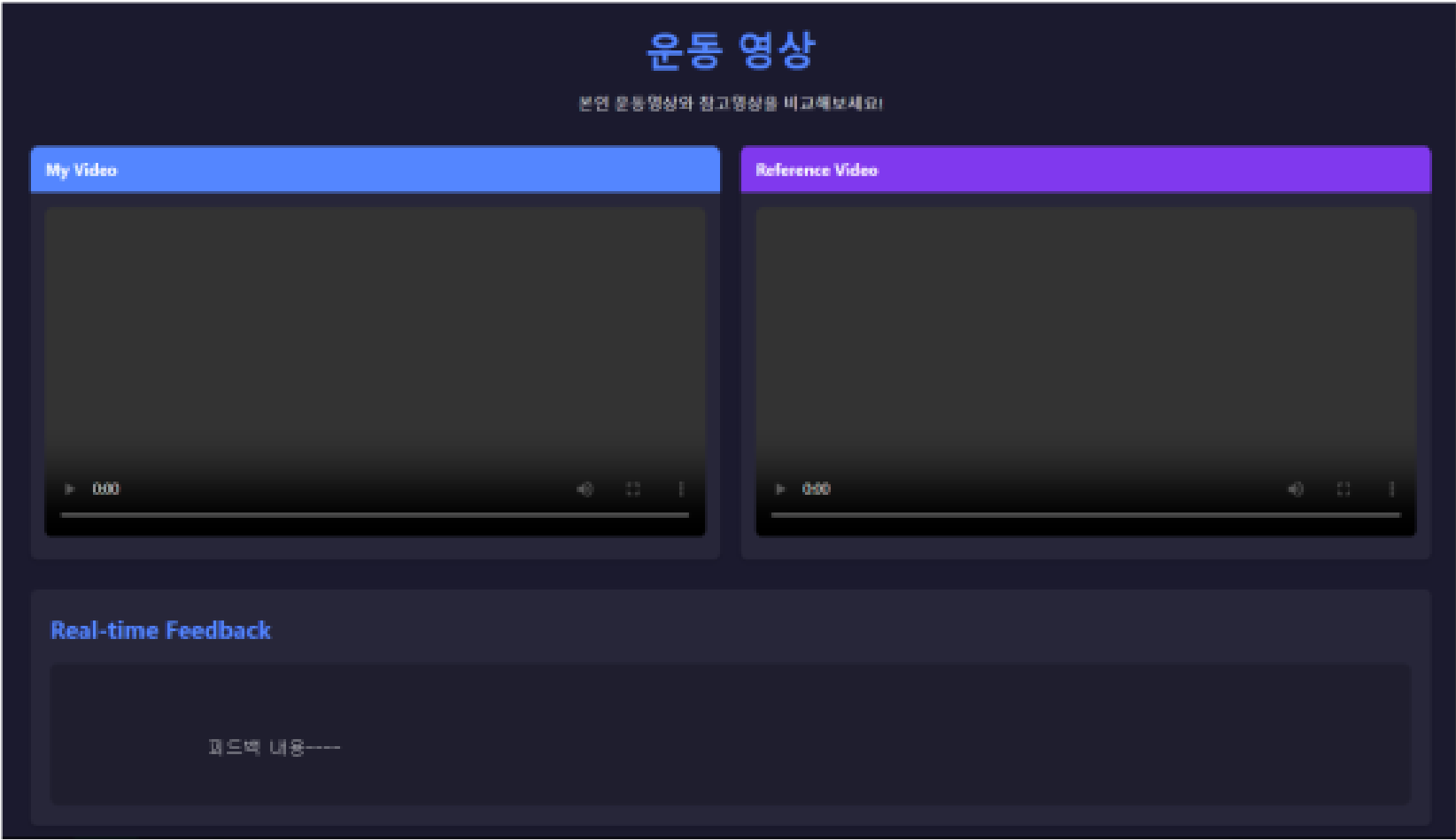
연구 개발 목표

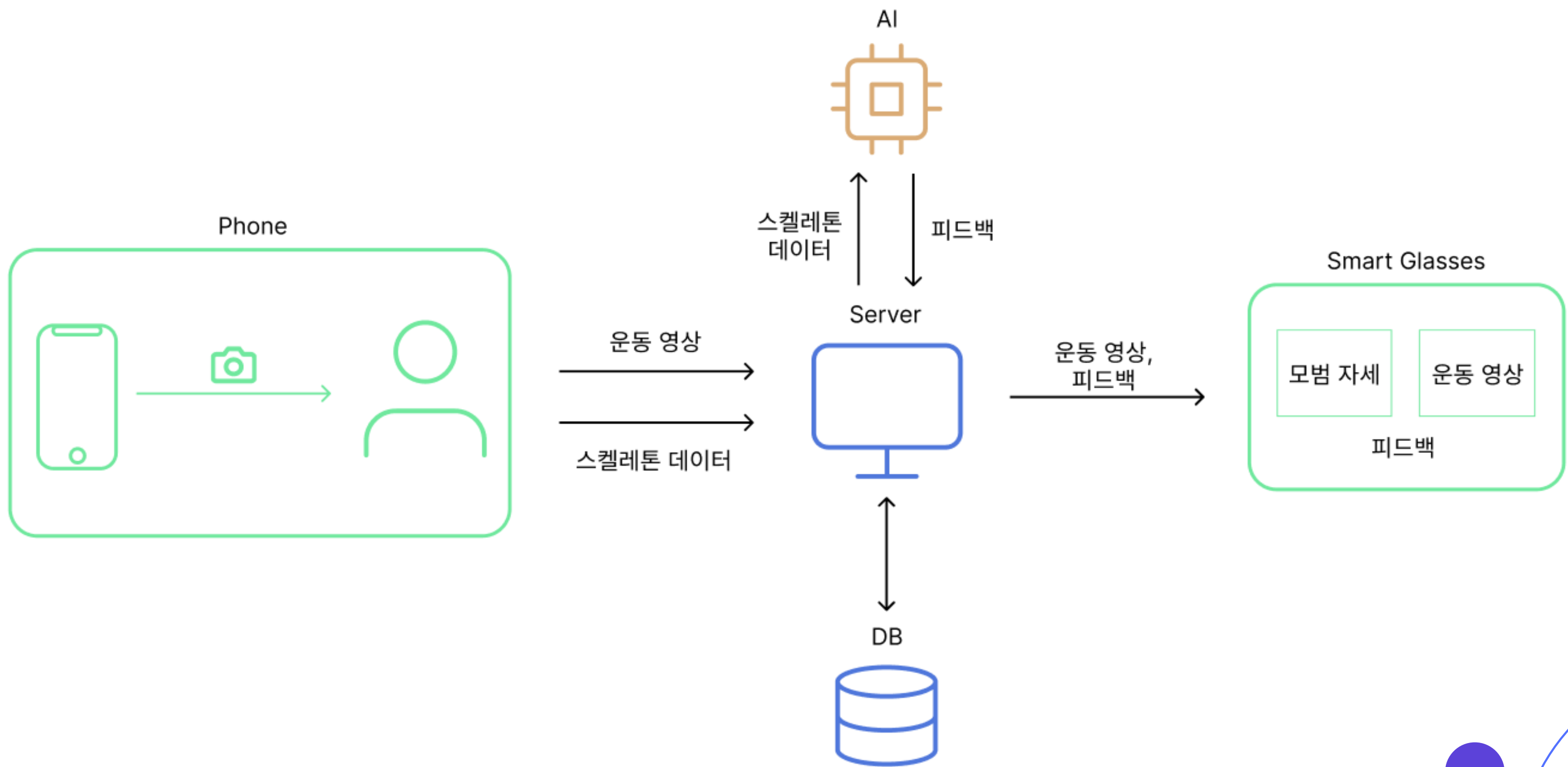
올바른 운동 자세 정보를 제공하고, 스마트글래스를 활용한 운동 자세 교정으로
피드백을 제공하는 디바이스 및 소프트웨어를 개발

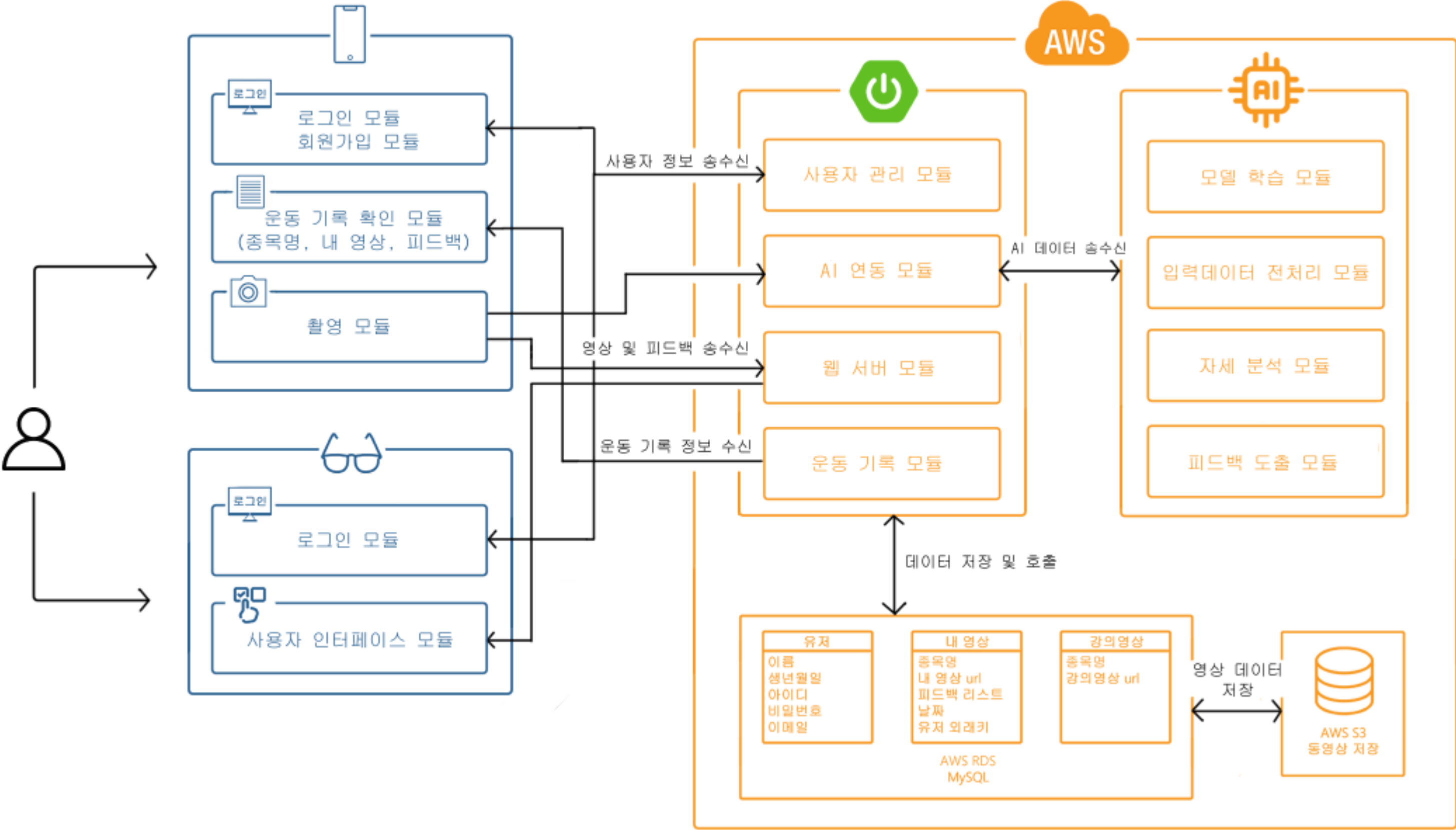
운동 중인 모습 촬영



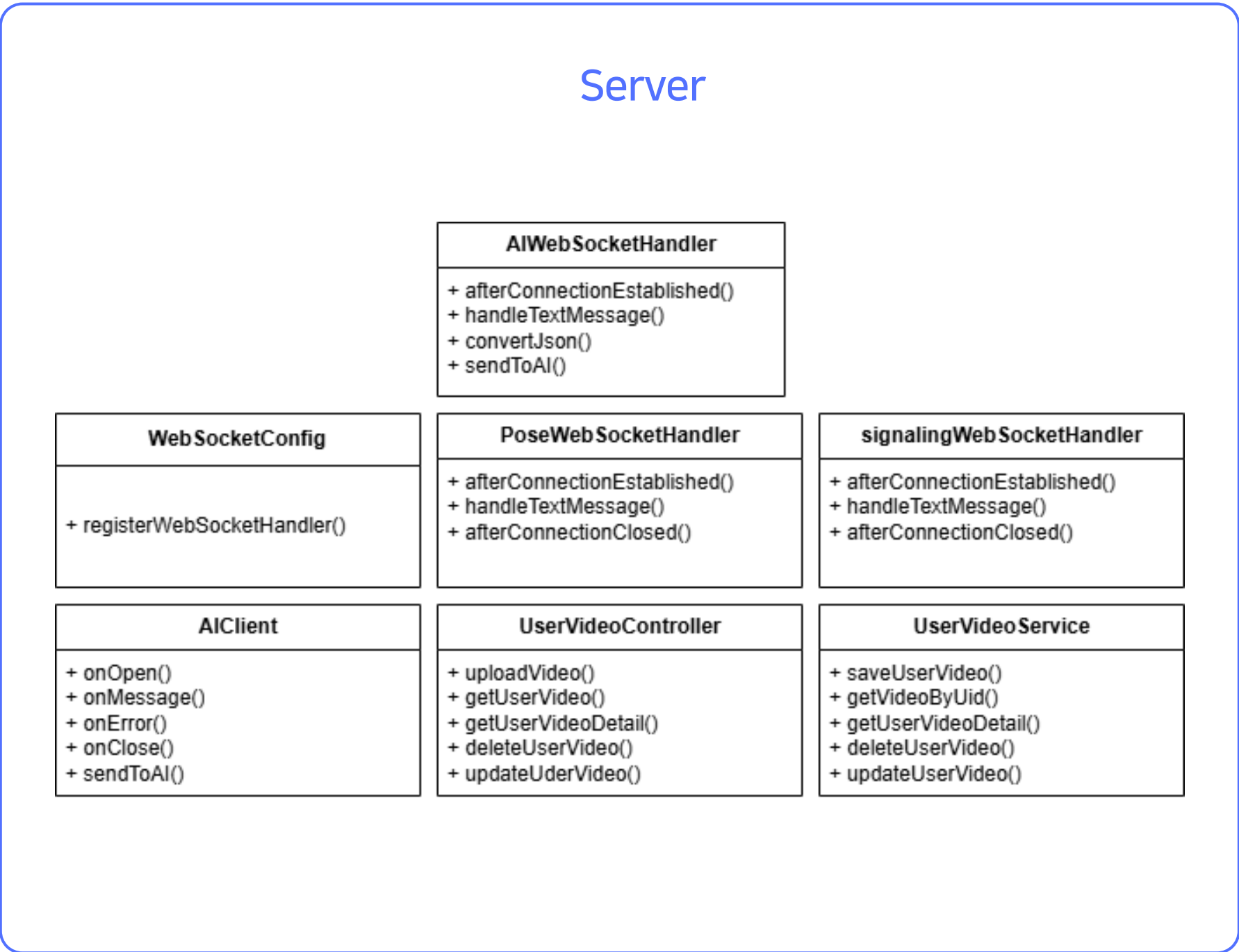
스마트글래스 화면





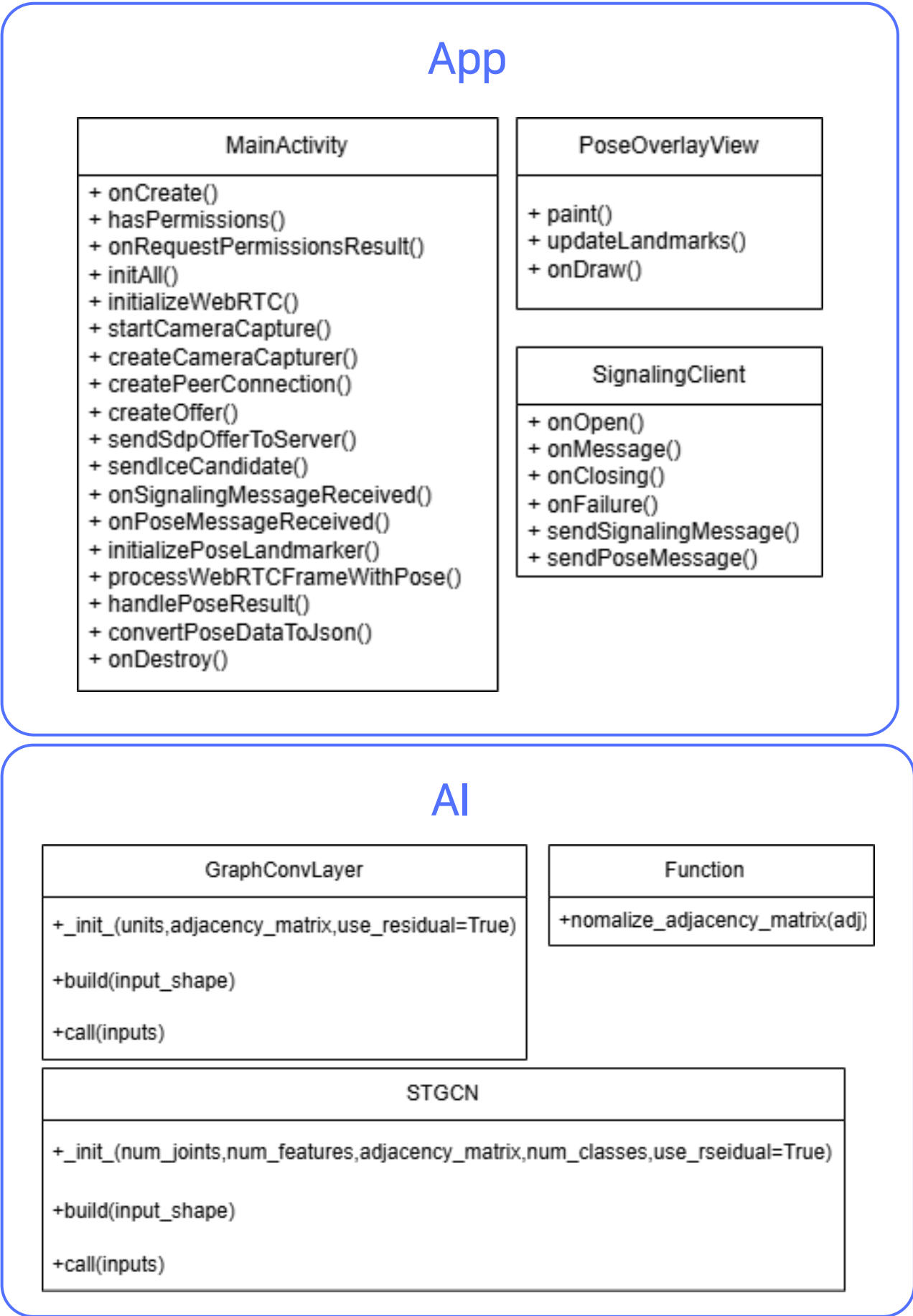


s/w모듈



운동 영상
←
←
스켈레톤 데이터

피드백
←
→
스켈레톤 데이터



시그널링 모듈 (Signaling Module)

```
class SignalingClient(
    // 시그널링 서버에서 오는 메시지를 받을 콜백
    private val signalingListener: (String) -> Unit,
    // 포즈 서버에서 오는 메시지를 받을 콜백
    private val poseListener: (String) -> Unit
) {

    private val client = OkHttpClient()

    // 시그널링 서버용 WebSocket
    private val requestSignaling = Request.Builder()
        .url("ws://15.165.138.10:8080/signaling")
        .build()

    init {
        // 시그널링용
        signalingWebSocket = client.newWebSocket(requestSignaling, object : WebSocketListener() {
            override fun onOpen(webSocket: WebSocket, response: Response) {
                // 연결 성공 시
            }
            override fun onMessage(webSocket: WebSocket, text: String) {
                // 시그널링 메시지를 전달
                signalingListener(text)
            }
            override fun onMessage(webSocket: WebSocket, bytes: ByteString) {
                signalingListener(bytes.utf8())
            }
            override fun onClosing(webSocket: WebSocket, code: Int, reason: String) {
                webSocket.close( code: 1000, reason: null)
            }
            override fun onFailure(webSocket: WebSocket, t: Throwable, response: Response?) {
                t.printStackTrace()
            }
        })
    }

    // 시그널링 서버로 메시지 전송 (offer, answer, candidate 등)
    fun sendSignalingMessage(message: String) {
        signalingWebSocket.send(message)
    }
}
```

개요

- WebRTC 연결을 위해 필요한 Offer/Answer/ICE Candidate 정보를 주고받는 핵심 모듈
- WebSocket(ws://.../signaling)을 통해 실시간 메시지 전달

제공 기능

1. OFFER/ANSWER 교환 기능
 - 안드로이드 클라이언트가 만들어낸 SDP Offer를 서버(WebSocket)로 전송 → 상대 피어와 공유
 - 서버에서 전달받은 SDP Answer를 다시 클라이언트에 반영
2. ICE Candidate 교환 기능
 - P2P 연결을 완성하기 위해 필요한 ICE 후보 정보(내부 IP, 외부 IP 등)를 주고받는 역할
3. WebSocket 실시간 통신
 - SignalingClient를 통해 서버와 실시간으로 메시지를 주고받으므로, REST API보다 빠르게 신호를 교환
4. 멀티 엔드포인트 지원(예: /signaling, /pose)
 - 같은 클래스에서 시그널링용 WebSocket, 포즈용 WebSocket을 나눠 연결할 수도 있음

AI 통신 모듈

```
public AIClient(){ ① Yang Donggyun
    //AI서버와 연결 설정
    try{
        WebSocketContainer container = ContainerProvider.getWebSocketContainer();
        aiSession = container.connectToServer( ② this, new URI(AI_URL));
        log.info("AI 서버 연결 성공!");
    }catch (Exception e){
        log.error("AI websocket 연결 실패!",e);
    }
}
```

```
// 사용할 관절 (필터링)
private final List<String> Points = Arrays.asList( ① usage
    "Point_0", "Point_2", "Point_5", "Point_7", "Point_8",
    "Point_11", "Point_12", "Point_13", "Point_14", "Point_15",
    "Point_16", "Point_17", "Point_18", "Point_21", "Point_23",
    "Point_25", "Point_26", "Point_28", "Point_30", "Point_31", "Point_32"
);
```

```
// JSON 변환 로직
private Map<String, Object> convertJson(Map<String, Object> rawData) {...
```

개요

- 클라이언트(안드로이드) 에서 전송하는 포즈 데이터를 수신
- 필요한 관절(Point)을 필터링하여 AI 서버에 전달할 JSON으로 변환

제공 기능

1. AI서버와 연결 기능
 - WEBSOCKET을 사용하여 AI서버와 직접 연결
2. 관절(랜드마크) 필터링
 - Points 리스트를 기준으로 필요한 관절만 추출
 - 불필요한 데이터는 제거해 AI 입력 형식에 맞춤

AI 통신 모듈

```
public void sendToAI(String json)throws IOException{
    if(aiSession != null && aiSession.isOpen()){
        aiSession.getBasicRemote().sendText(json);
        log.info("AI 모델로 데이터 전송: {}",json);
    }else{
        log.error("AI Socket 세션 닫혀 있음.");
    }
}
```

```
@OnMessage no usages Yang Donggyun
public void onMessage(String message){
    log.info("AI 응답 수신: {}",message);
    try{
        for(WebSocketSession session : clientSessions.values()){
            if(session!= null && session.isOpen()){
                session.sendMessage(new TextMessage(message));
                log.info("클라이언트로 전달 완료");
            }
        }
    } catch (IOException e) {
        log.error("전달 중 오류 발생",e);
    }
}
```

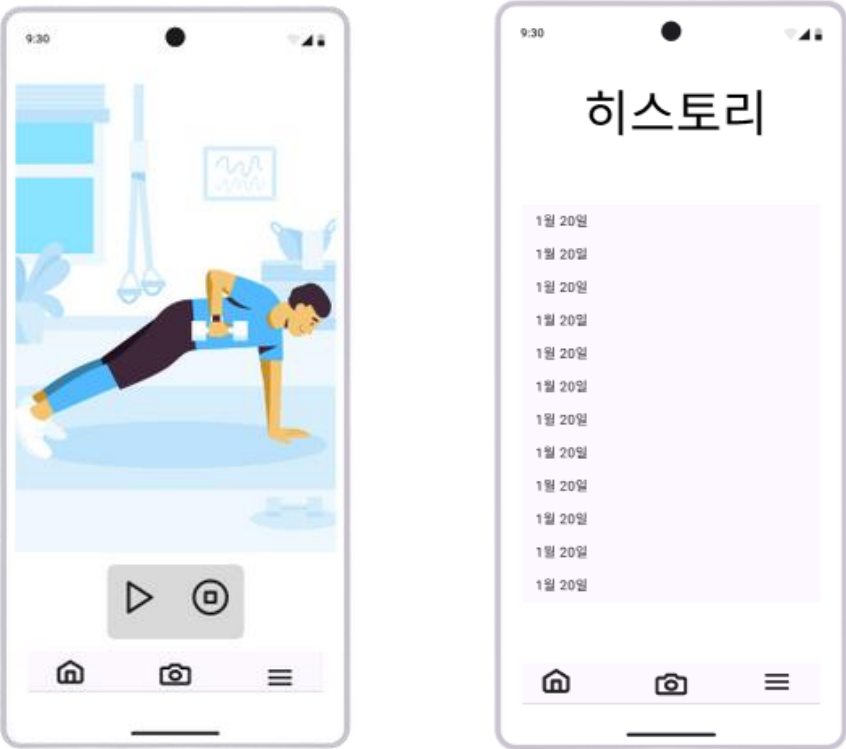
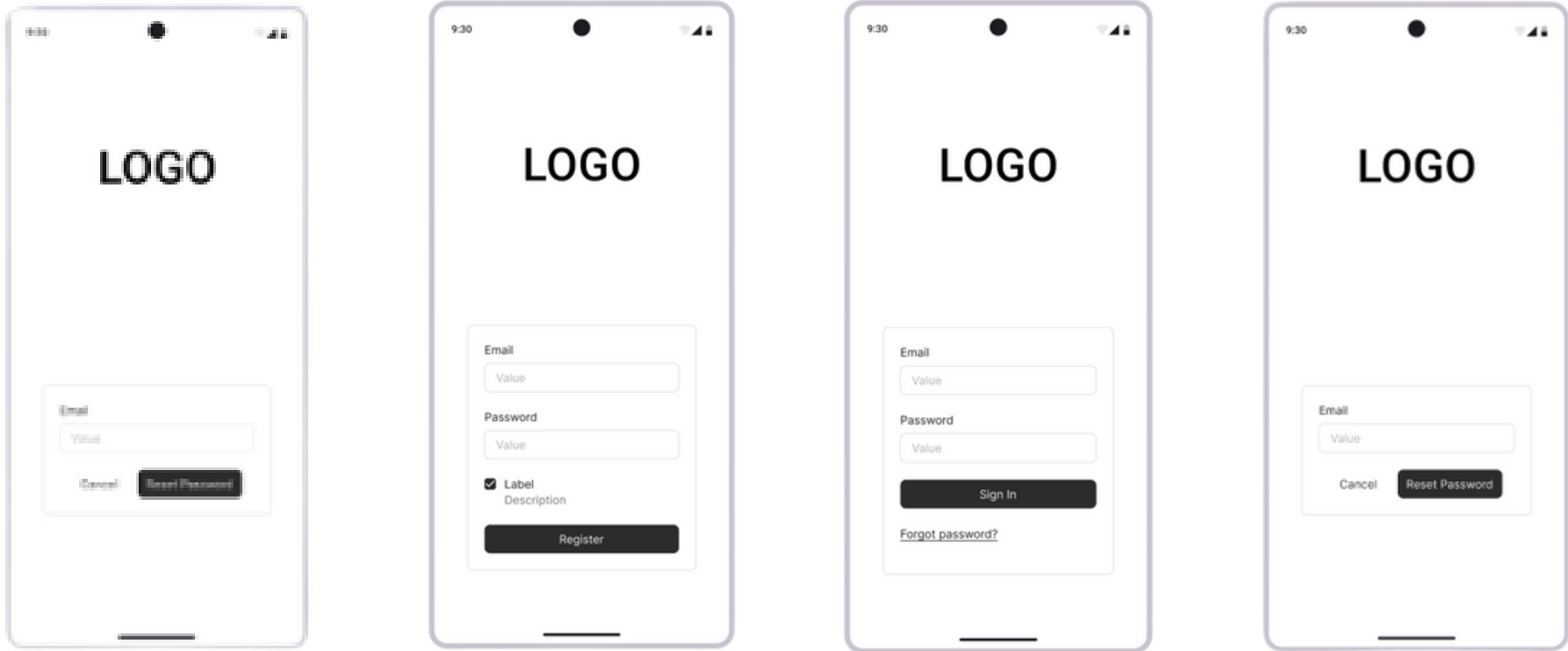
개요

- 필터링된 JSON데이터 AI서버로 전달
- AI서버로부터 응답이 오면 클라이언트에게 피드백 전달

제공 기능

1. AI 서버로 데이터 전송 기능
 - WEBSOCKET을 사용하여 AI서버와 실시간 데이터 교환 수행
2. AI 모델 응답 수신 및 클라이언트 전달 기능
 - AI 서버의 분석 결과를 클라이언트로 자동 전달

ui



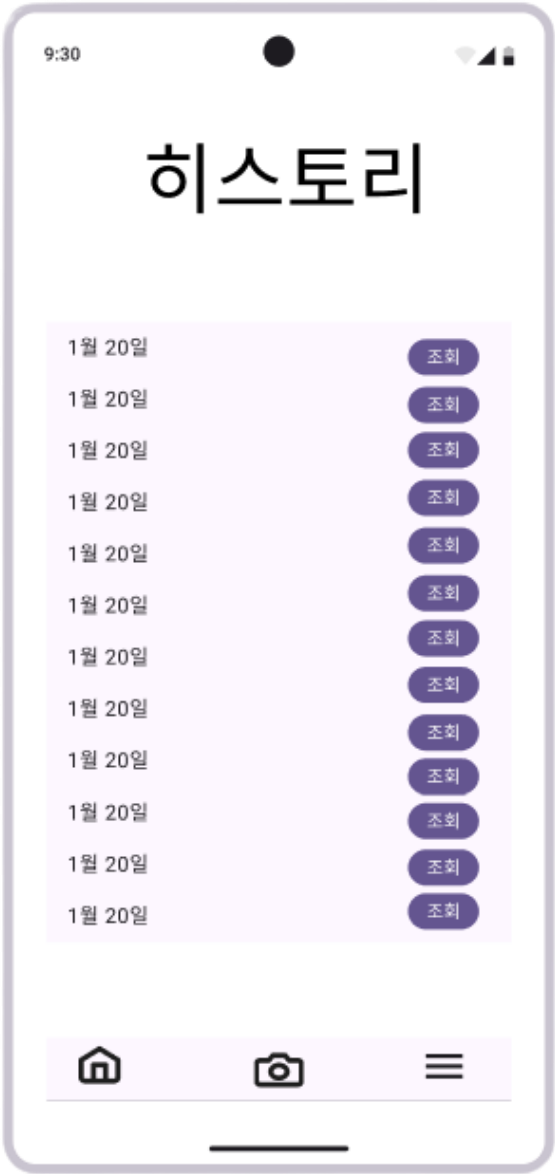
API

getUserVideo(uid)

특정 사용자(UID)의 운동 기록(영상 목록)을 조회

getUserVideoDetail(vid)

특정 비디오(vid)에 대한 상세 정보 조회



uploadVideo()

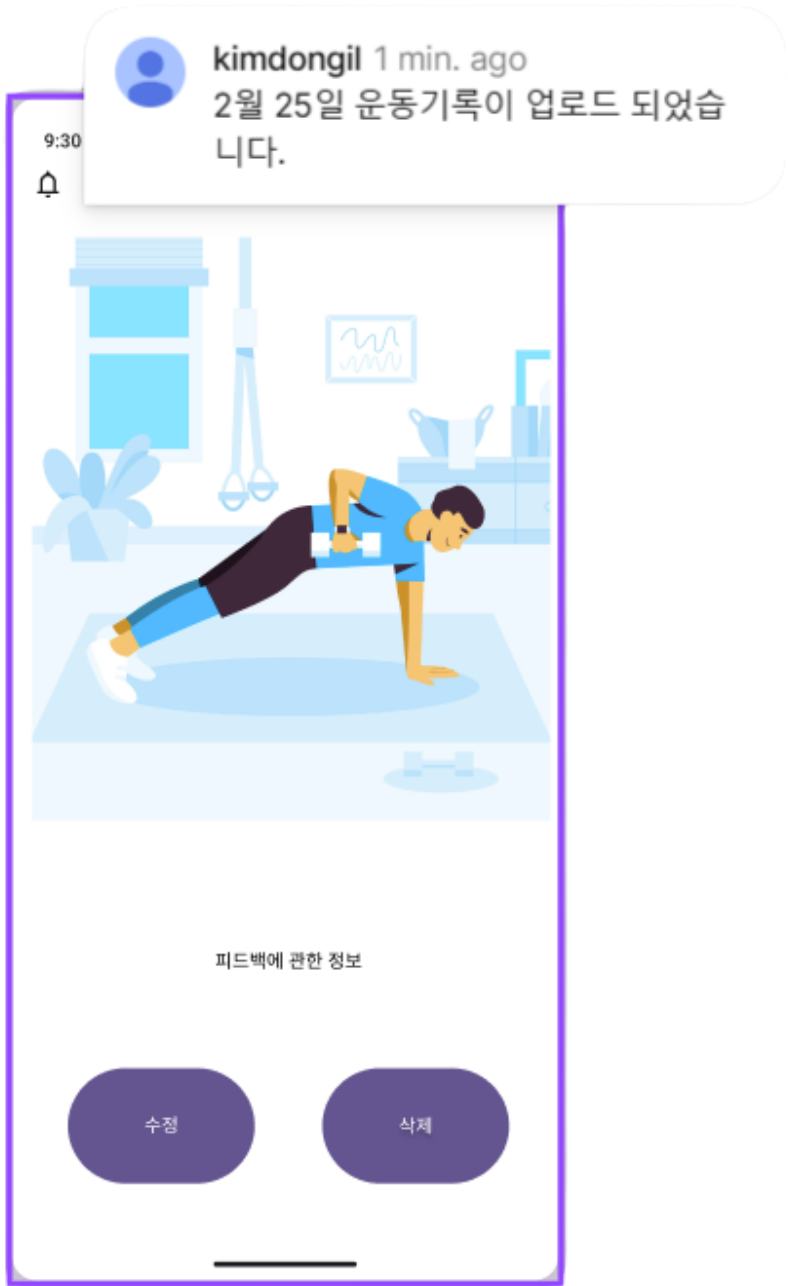
운동 기록을 업로드하는 API

deleteUserVideo(vid, video)

특정 비디오를 삭제

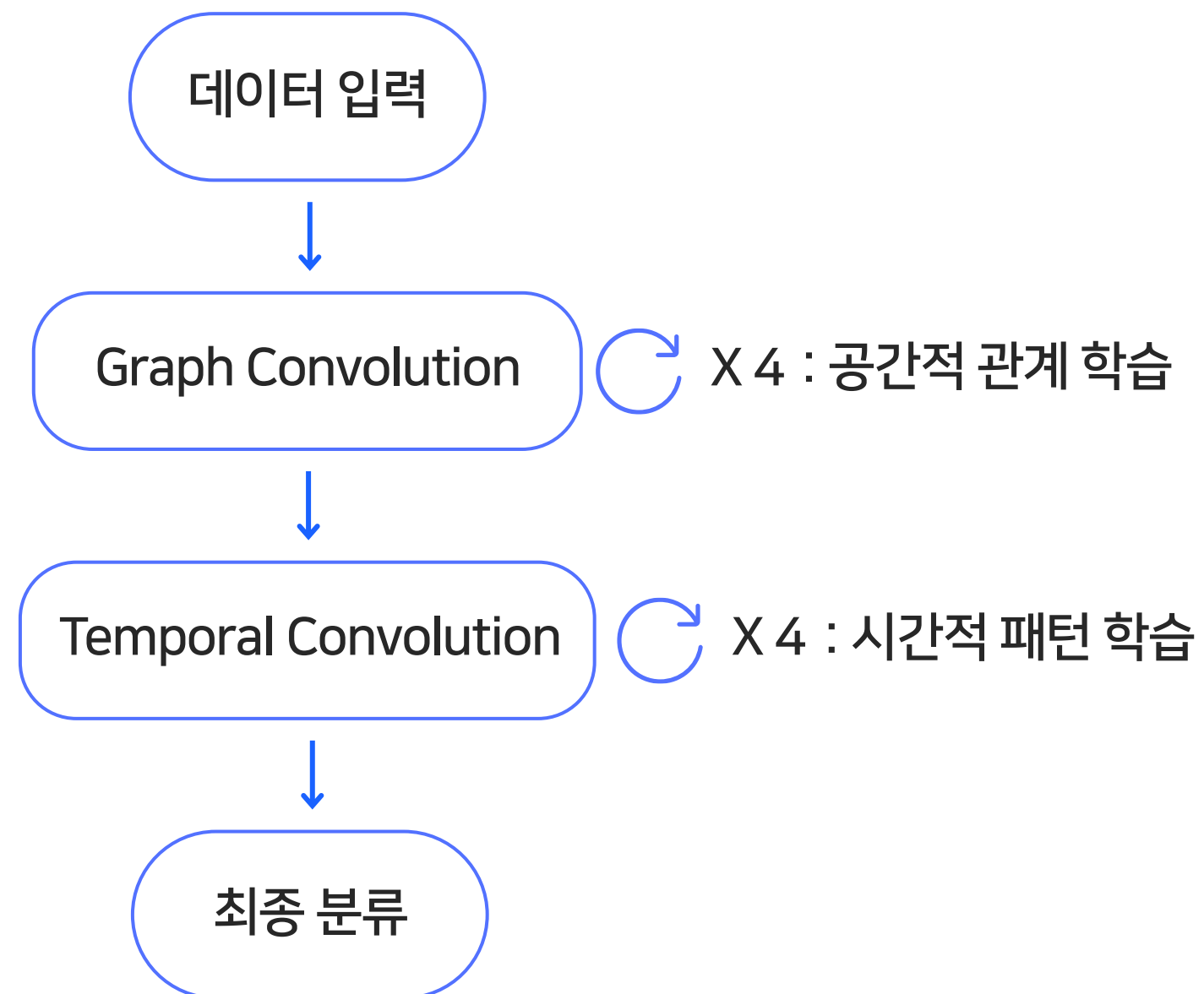
updateUserVideo(vid, video)

특정 비디오 정보를 수정



알고리즘

ST-GCN 구조



피드백 제공

if 운동 자세 True:

“자세 올바름” 제공

if 운동 자세 False:

잘못된 신체부위 탐색:
각 신체부위의 좌표를
비교하여 이상 감지

알고리즘

양끝단 프레임 제거:

비운동 활동을 고려하여 양끝단 데이터 필터링

기준 초과 프레임 비율 설정:

이상치 및 순간의 프레임 때문에 판정 오류가 생기지 않도록 설정

척추 중립 여부:

척추가 중립적인 상태를
유지하는지 확인

```
spine_X, spine_Y  
= upper_back - lower_back  
  
spine_angle =  
np.arctan2(spine_X, spine_Y)  
* (180 / np.pi)  
  
avg_spine_angle =  
np.mean(spine_angle)
```

가슴 이동 여부:

가슴이 충분히 아래로
내려가는지 확인

```
min_height =  
np.percentile(chest_Y, 10)  
max_height =  
np.percentile(chest_Y, 90)  
  
movement_range =  
max_height - min_height
```

손 위치 정상 여부:

손의 위치가 가슴과
일직선상에 있는지 확인

```
hand_alignment =  
np.abs(hand - chest)  
  
avg_hand_alignment =  
np.mean  
    (hand_alignment)
```

머리 정렬 여부:

머리가 바르게
정렬되어 있는지 확인

```
head_angle =  
np.arctan2(head, back)  
* (180 / np.pi)  
  
head_alignment =  
np.abs(head_angle -  
    spine_angle)
```

졸업작품 GitHub 주소

[HTTPS://GITHUB.COM/2025TUKCOMCD/BUCKET](https://github.com/2025TUKCOMCD/BUCKET)

팀원별 GitHub ID

팀장: 이예권

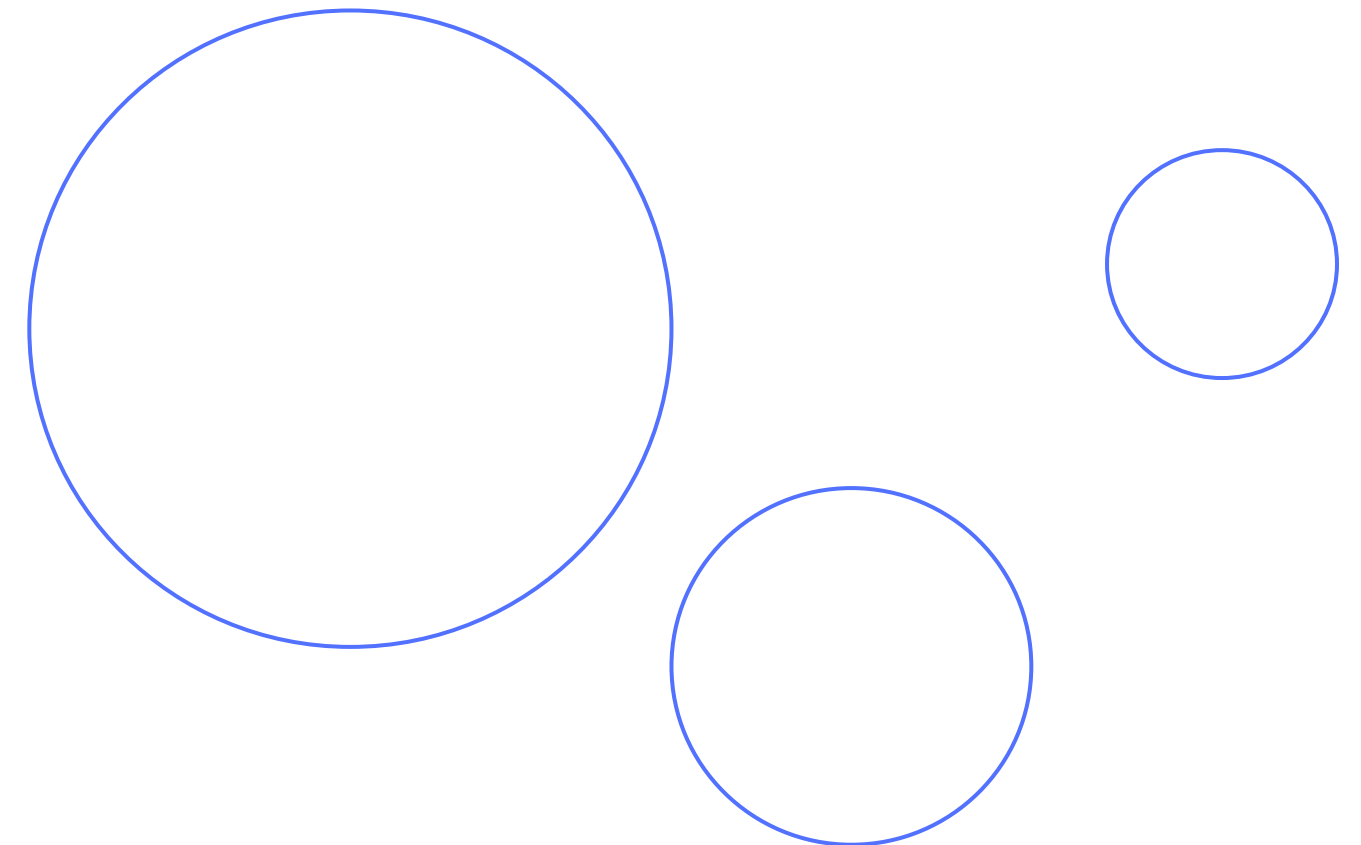
ID: [@Baaunn](#)

팀원: 김동일

ID: [@kimdongiil](#)

팀원: 양동균

ID: [@ydg0724](#)

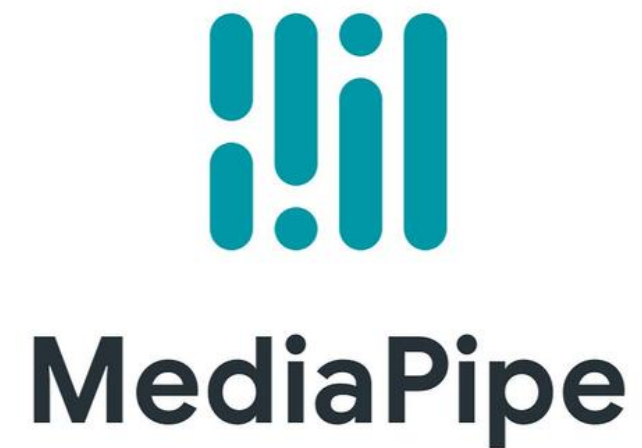


<div>이예권</div> <div>2021152027</div> <div>Baaunn</div>	<div>김동일</div> <div>2022150005</div> <div>kimdongiil</div>	<div>양동균</div> <div>2019150028</div> <div>ydg0724</div>
<div>DB 설계 및 관리</div> <div>자세 판단 기능</div> <div>딥러닝 모델 개발</div>	<div>백엔드 개발</div>	
	<div>스마트 폰 앱 개발</div>	<div>AR 개발</div>

[illegible]

Mediapipe

사용자의 움직임을
스켈레톤 데이터로 전환



Smart glasses(Xreal)

AR화면 구현을 위한
스마트글래스 활용



AWS Cloud Server

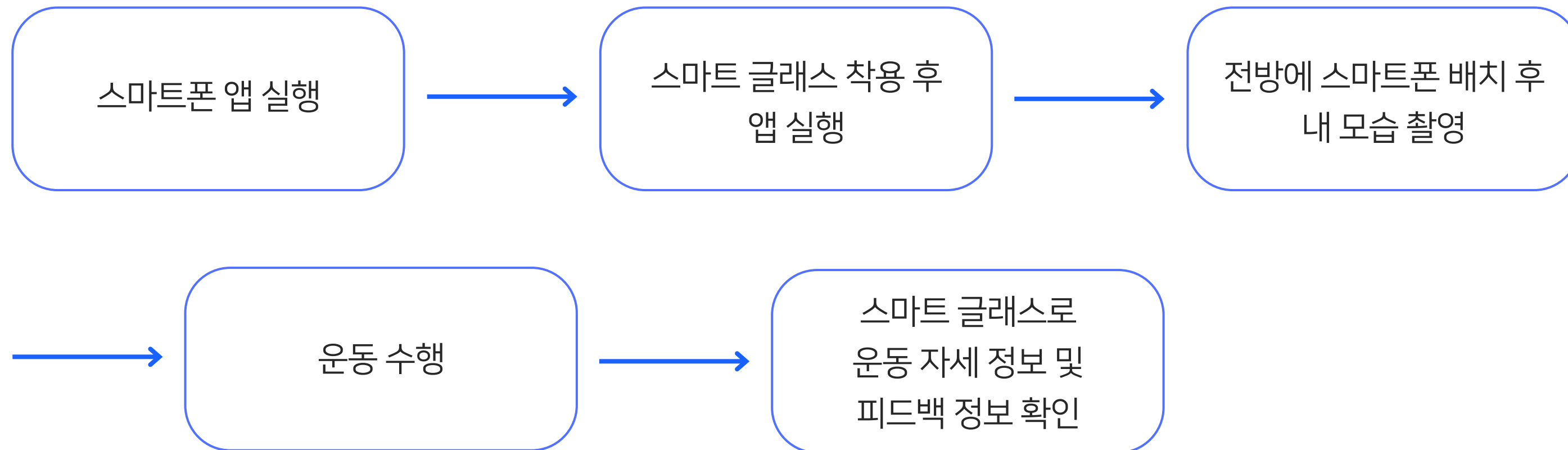
서버 기능 처리와 데이터 관리를
제공하는 AWS 클라우드



데모 환경 설계

주변에 스마트폰을 배치하고 운동할 수 있는 공간
사전 녹화 진행

데모 진행 순서

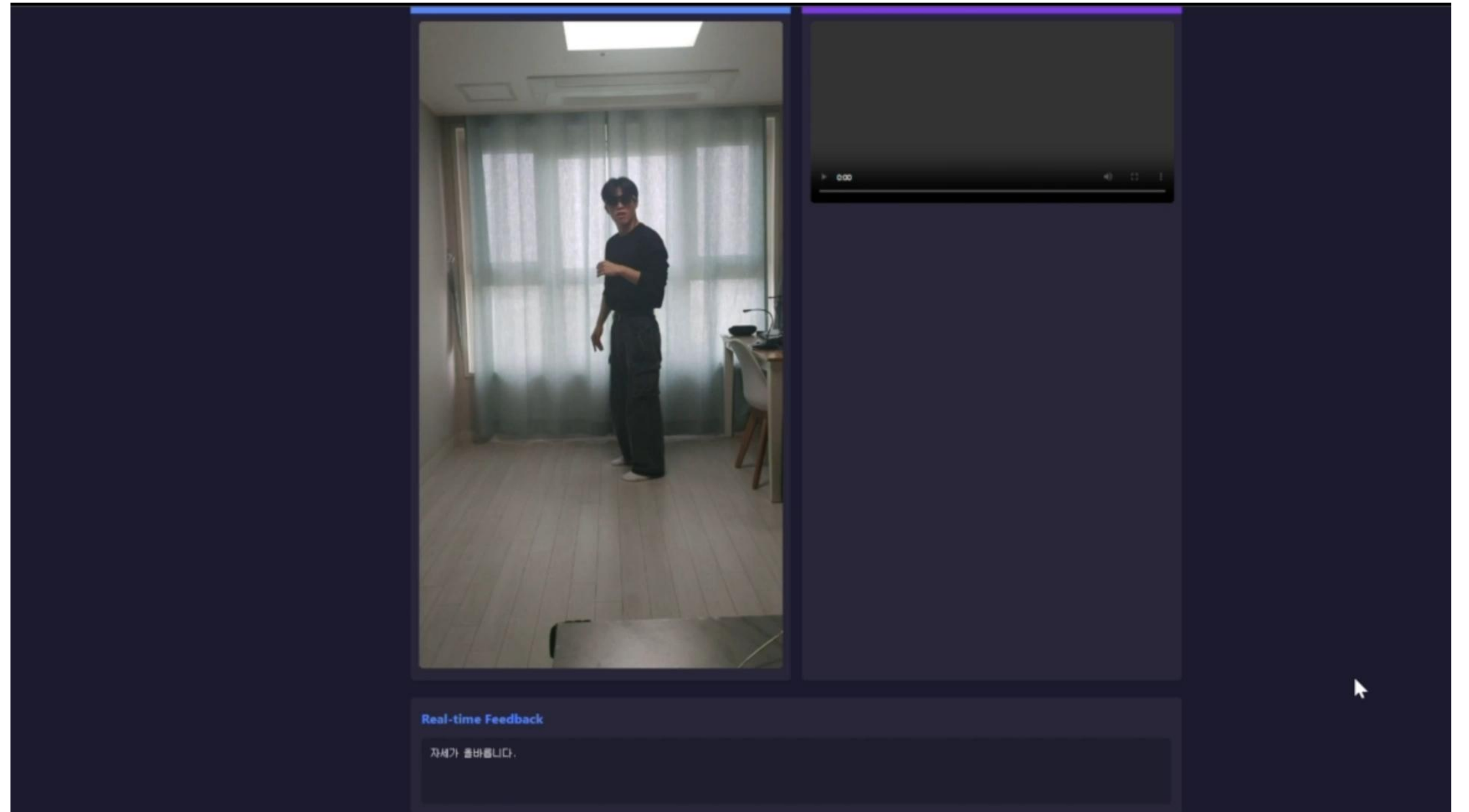


데모

데모 환경 사진



데모 동영상



감사합니다