




컴퓨터과학 종합설계: 상세설계보고서

기술개발 과 제	스마트 기기를 활용한 배드민턴 분석 프로그램 (영문 Badminton Analysis Program Using Smart Devices)				
과제팀 이름	곡곡		지도교수	김민호 교수님	
개발기간	2023년 9월 ~ 2023년 12월 (총 4개월)				
개발소요비용	총 액	648 (천원)	학교부담금	381천원	
			과제팀부담금	267천원	
과제팀 구성원	이름	강송모	신도영	임재욱	최지웅
	사진				
	학번	2018920001	2018920028	2018920052	2018920057
	연락처	010-9991-2002	010-6779-6906	010-4037-7572	010-5564-7151

컴퓨터과학종합설계 과제를 성실히 수행하고자 상세설계 보고서를 제출합니다.

2023년 11월 30일

과제 수행자1 : 강송모 
과제 수행자2 : 신도영 (인)
과제 수행자3 : 임재욱 
과제 수행자4 : 최지웅 

지도교수 : 김민호 교수님 (인)

서울시립대학교 컴퓨터과학부 귀중

1. 개발 과제의 요약

국내에서 배드민턴을 치는 인구는 약 320만명으로, 대표적 생활체육 종목으로 자리잡고 있다. 또한 아마추어 선수가 많아 다른 스포츠와 달리 지역 단위의 대회 또한 다수 활성화되어 있다. 이처럼 배드민턴은 입문하기 쉽고 대중적이지만, 바른 자세가 중요한 스포츠이기도 하다. 빠르고 복합적인 움직임을 요구하는 배드민턴 특성 상, 아마추어 수준에서 자신의 자세를 피드백 받을 수 있는 방법은 상급자의 레슨 등으로 한정적이다. 이러한 방법을 접하지 않은 이들의 잘못된 자세 습관은 부상을 유발하고, 실력 향상에 걸림돌이 된다. 전문가 수준에서는 다양한 센서 데이터 수집을 통한 분석이 시도되고 있지만, 이는 추가적인 전문 장비를 요구하기 때문에 아마추어 수준에서 기술 접근성이 떨어진다. 서비스 '콕콕'은 사용자의 스윙 및 경기 데이터를 분석해 실력 수준을 측정하고, 실력 향상을 위한 정보를 제시한다. 이를 통해 생활체육으로 배드민턴을 즐기는 이들의 기술 접근성을 높이고, 건강한 체육 활동에 도움이 되고자 한다.

2. 개념설계안

가. 스윙 분석 기능

- 사용자의 하이클리어 스윙 과정 영상과 IMU 데이터를 수집
- 수집한 데이터를 분석해 정량적인 점수로 변환
- 사용자의 스윙 장단점을 분석해 피드백을 제공
- 기록되어 있는 스윙들의 점수 추이를 그래프로 표현해 사용자의 실력 변화를 시각화

나. 경기 기록 기능

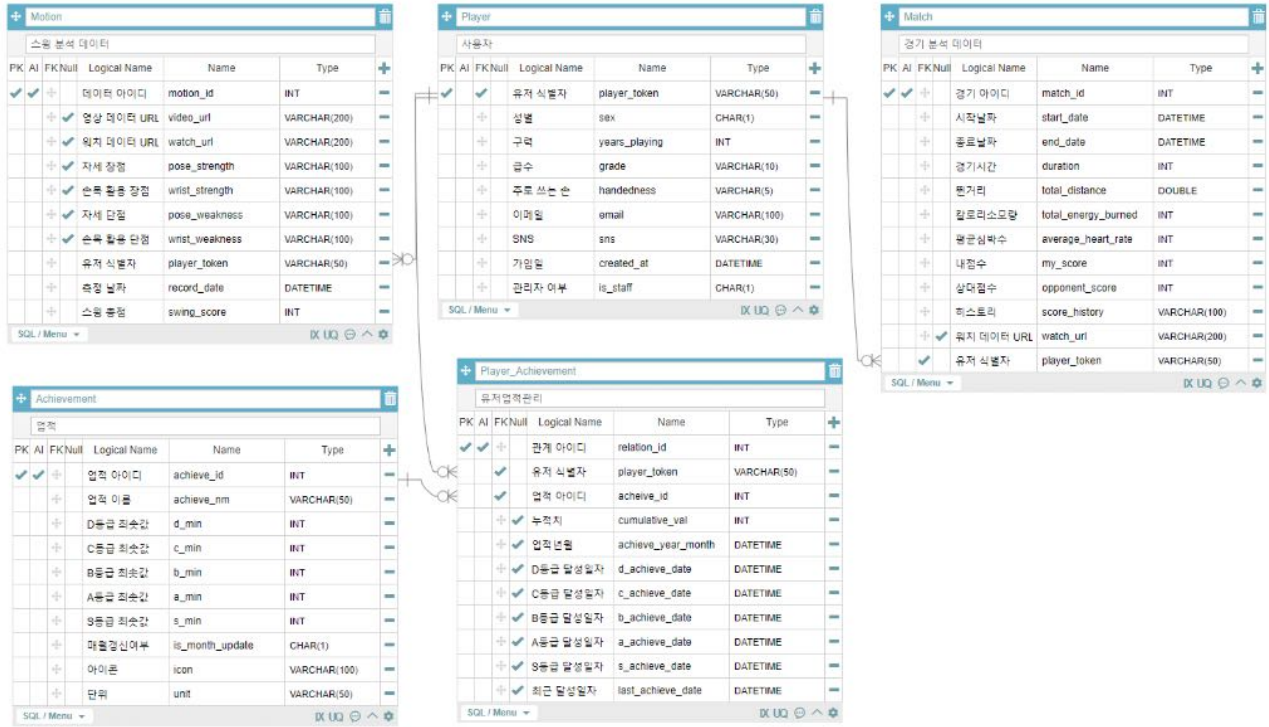
- 사용자의 배드민턴 경기 동안의 IMU 데이터와 운동 데이터를 수집
- 수집한 데이터를 분석해 스윙이 일어난 시점과 스윙의 종류를 구분하고 평가
- 경기가 끝나면 운동 데이터, 스윙 횟수 등을 시각적으로 표현

다. 요약 페이지

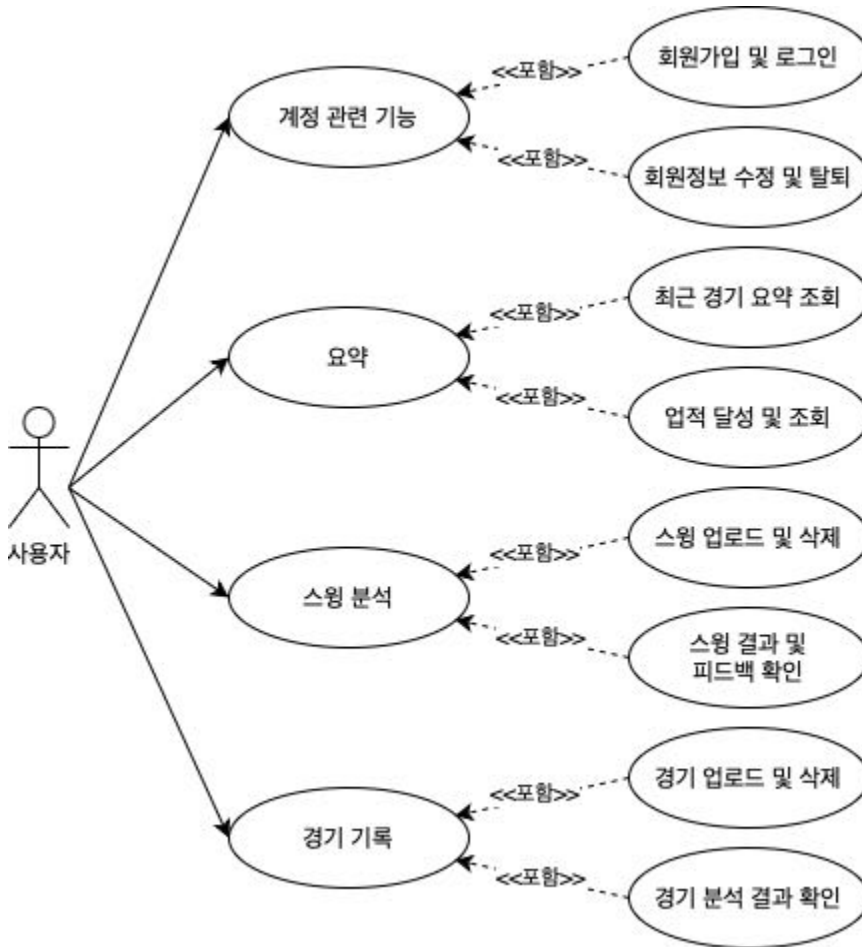
- 사용자의 한 달 동안의 운동 기록을 시각적으로 표시
- 사용자의 운동 기록을 업적으로 변환해 운동에 대한 지속적인 동기 부여

3. 소프트웨어 설계

◇ 데이터베이스 ERD 설계



◇ 유즈케이스 다이어그램



• 계정 관련 기능

• 회원가입 및 로그인

사용자가 소셜 계정으로 서비스에 로그인하면서 시작된다.

시스템은 Firebase로부터 발급된 토큰을 식별 및 검증하고, 토큰이 일치하면 회원가입 또는 로그인 페이지로 넘어간다. 사용자가 처음 로그인했다면 본인의 기본정보를 입력하고, 이후 앱의 기능을 사용할 수 있다.

• 회원정보 수정 및 탈퇴

사용자가 앱의 '마이 페이지' 탭을 열면서 시작된다. 시스템은 사용자의 기본 정보를 표시하고, 다양한 기능을 제공한다. 시스템은 사용자가 다음 사항 중 하나를 선택하도록 한다.

- 로그아웃
- 기본 정보 수정
- 캐시 삭제
- 데이터 삭제
- 회원 탈퇴

• 요약

• 최근 경기 요약 조회

사용자가 앱을 열어 요약 탭을 열면서 시작된다. 시스템은 사용자의 최근 한 달간의 경기를 분석하여 다음 사항에 대한 통계 정보를 그래프와 함께 보여준다.

- 경기 수
- 경기 시간
- 스윙 종류와 횟수
- 승률
- 이동 거리

- 업적 달성 및 조회

요약 탭에서 최근 달성 업적과 전체 업적이 표시된다. 사용자가 경기를 기록하면 시스템은 경기 통계를 바탕으로 사용자의 업적 달성 정보를 자동으로 갱신한다. 사용자는 업적의 달성 여부와 달성 날짜를 확인할 수 있다.

- 스윙 분석

- 스윙 업로드 및 삭제

사용자가 스윙 분석 탭의 스윙 업로드를 눌러 스윙을 기록할 수 있다. 사용자는 절차에 따라 스마트폰과 스마트 워치에서 스윙 분석 기능을 수행하고, 서버에 스윙을 업로드 할 수 있다.

Alternate flow : 업로드 된 스윙이 분석 불가능할 경우 시스템은 사용자에게 오류 내용을 표시
사용자가 업로드한 스윙 분석 결과를 삭제할 수 있다.

- 스윙 결과 및 피드백 확인

사용자가 스윙 분석 탭의 스윙을 선택하여 결과를 확인할 수 있다. 시스템은 사용자에게 스윙 구간(임팩트 전, 임팩트 시점, 임팩트 후)별 점수와 장단점을 그래프와 함께 표시하고, 영상과 자세에 대한 장단점을 표시한다.

- 경기 기록

- 경기 업로드 및 삭제

사용자가 스마트 워치에서 경기 기록을 선택하면서 시작된다. 시스템은 사용자의 손목 움직임을 기록하면서 경기 중의 운동 정보, 점수판 등을 표시하고, 사용자는 경기 중 그것을 제어할 수 있다. 경기가 종료되면 사용자는 기록 종료를 선택하여 서버에 경기를 업로드한다.

Alternate flow : 스마트 워치가 인터넷에 연결되어 있지 않으면 기기 내부에 경기 기록을 임시 저장하고, 추후 인터넷에 연결되면 자동으로 업로드한다.

사용자가 업로드한 경기 분석 결과를 삭제할 수 있다.

- 경기 분석 결과 확인

사용자가 경기 분석 탭의 경기를 선택하여 결과를 확인할 수 있다. 시스템은 사용자에게 운동 정보와 경기 통계, 스윙 종류별 점수와 횟수 등을 그래프와 함께 표시한다.

◇ 클래스 다이어그램

1. iOS



ModelData

앱 내의 전체 데이터를 가지고, 변경 사항을 UI에 즉각 반영한다.

AuthenticationManager

회원가입과 로그인, 로그아웃 등의 기능을 담당한다. APIManager의 기능들을 사용하여 서버와 통신하고, 받은 사용자 데이터를 ModelData에 전달한다.

SwingRecordManagerPhone

'스윙 분석' 기능을 위한 두 개의 클래스 중 하나이다. WCSSession을 이용하여 watchOS와 상호작용하여 녹화 상태를 통제하고, 녹화중인 화면을 watchOS로 전송하며, 데이터를 DataManager를 사용하여 저장하거나 불러오고, APIManager를 이용하여 서버에 업로드한다.

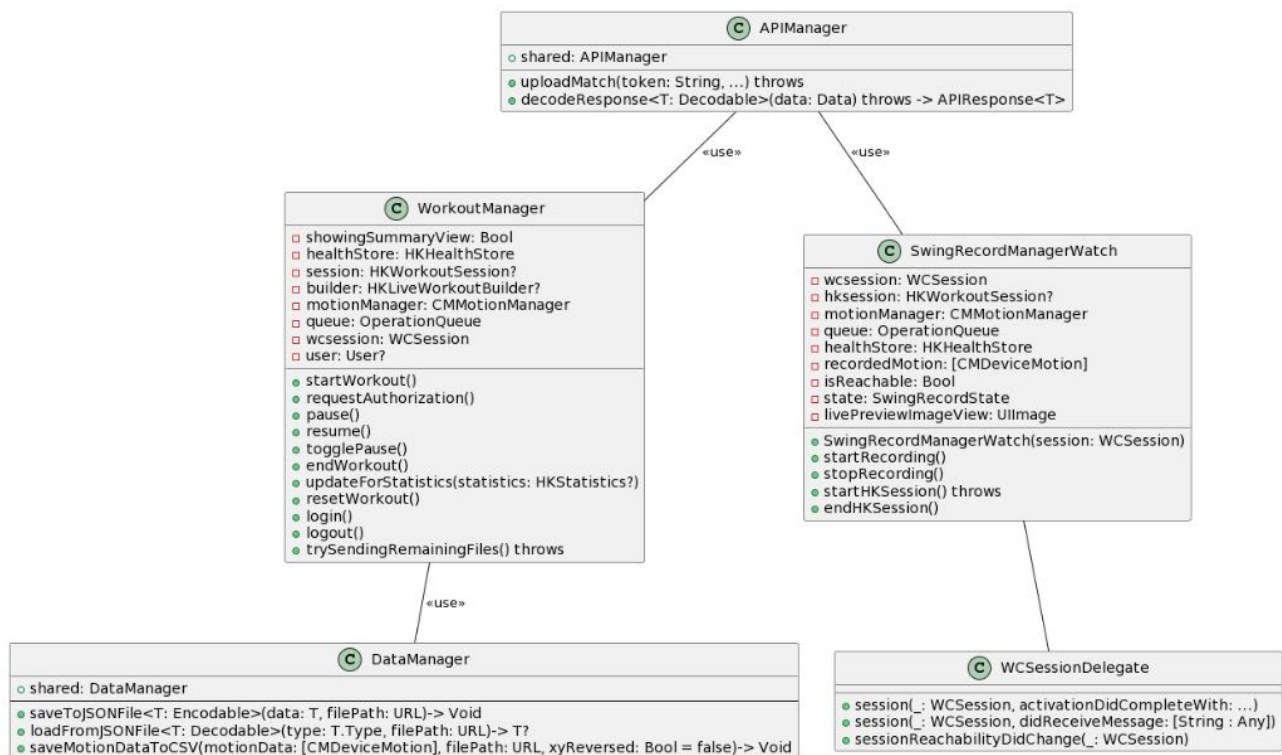
APIManager

실제 서버와 상호작용하는 클래스이다. 모든 서버 관련 기능은 이 클래스를 통해 이루어진다.

DataManager

파일 처리를 위한 클래스이다. 파일 인코딩과 디코딩은 이 클래스를 통해 이루어진다.

2. watchOS



WorkoutManager

'경기 기록' 기능을 위한 클래스이다. HKWorkoutSession을 이용하여 사용자의 운동 상태를 모니터링하고 기록하며, CMMotionManager클래스를 이용하여 사용자의 손목 움직임을 기록한다.

SwingRecordManagerWatch

'스윙 분석' 기능을 위한 두 개의 클래스 중 하나이다. iOS와 제어 신호를 주고받으며 손목 움직임을 기록하고, 녹화된 손목 움직임을 iOS로 전송한다. iOS로부터 녹화되고 있는 영상 스냅샷을 전송받아 화면에 표시한다.

WCSessionDelegate

실제 iOS와 상호작용하는 위임자 클래스이다. iOS로부터 받은 신호를 처리하고, 연결 상태를 체크한다.

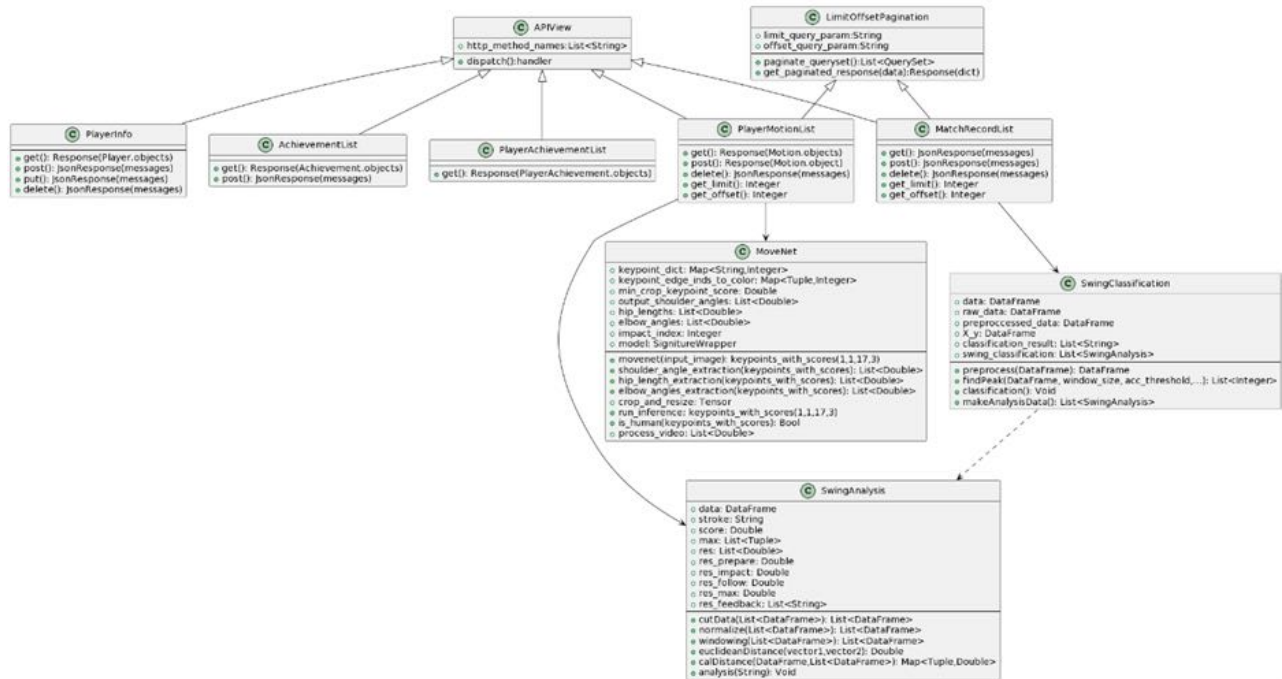
APIManager

실제 서버와 상호작용하는 클래스이다. 모든 서버 관련 기능은 이 클래스를 통해 이루어진다.

DataManager

파일 처리를 위한 클래스이다. 파일 인코딩과 디코딩은 이 클래스를 통해 이루어진다.

3. django 서버



APIView

GET,POST와 같은 Method가 포함된 HTTP 요청을 처리하기 위한 Class Based View이다.

PlayerInfo,AchievementList, PlayerAchievementList

각각 계정, 업적, 유저 개개인 별 업적 관련 유즈케이스를 구현하기 위한 클래스이다.

LimitOffsetPagination

GET 요청 시 제시된 limit, offsetparameter를 인자로 받아 그에 따른 Objects들을 반환하는 클래스이다.

예를 들어<ServerURL>?limit=3&offset=2일 때, object(2)부터 3개를 반환한다.

PlayerMotionList

영상 데이터 분석 & IMU 데이터 분석을 통해 사용자의 스윙 실력을 점수화하고 피드백을 제공해주는 클래스이다. 영상 데이터 분석은 MoveNet 클래스의 progress_video()를, IMU 데이터 분석은 SwingAnalysis 클래스의 analysis() 메서드를 사용한다.

MatchRecordList

HTTP 요청에 따라 경기 데이터를 저장하거나 조회하는 클래스이다. 이 때, 경기데이터에IMU 데이터가 포함되었다면 SwingClassification 클래스의 classification()와 makeAnalysisData() 메서드를 통해 데이터를 분석하여 사용자에게 제공한다.

SwingClassification

IMU데이터의 Peak를 찾아 특정 스윙 구간을 추출하고 이를 KNN 알고리즘을 통해 분류한 후, analysis() 메서드를 통해 스윙에 대한 점수를 매긴다. SwingAnalysis가 선언되어야 올바르게 기능할 수 있다.

SwingAnalysis

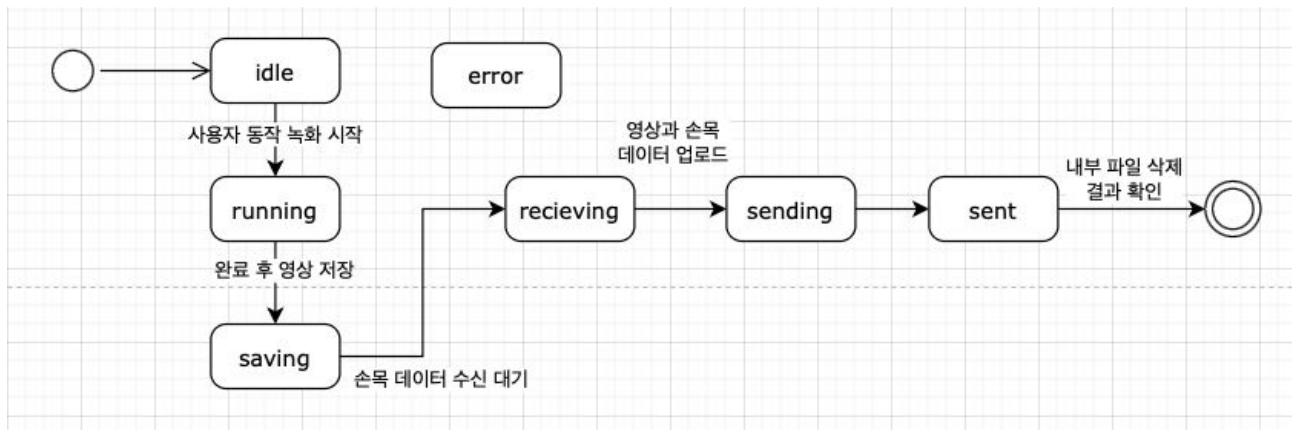
마찬가지로 IMU데이터의 Peak를 기준으로 스윙 구간을 추출한 후, 모범 데이터와의 편차를 통해 스윙에 대한 점수를 매긴다.

MoveNet

영상 데이터를 프레임 별 배열로전처리한 후, 이를 movenet을 통해 분석하여 17개의 관절 좌표를 추출한다. 관절 좌표들을 활용하여 스윙에 중요한 요소들을 계산하고 이를 제공한다.

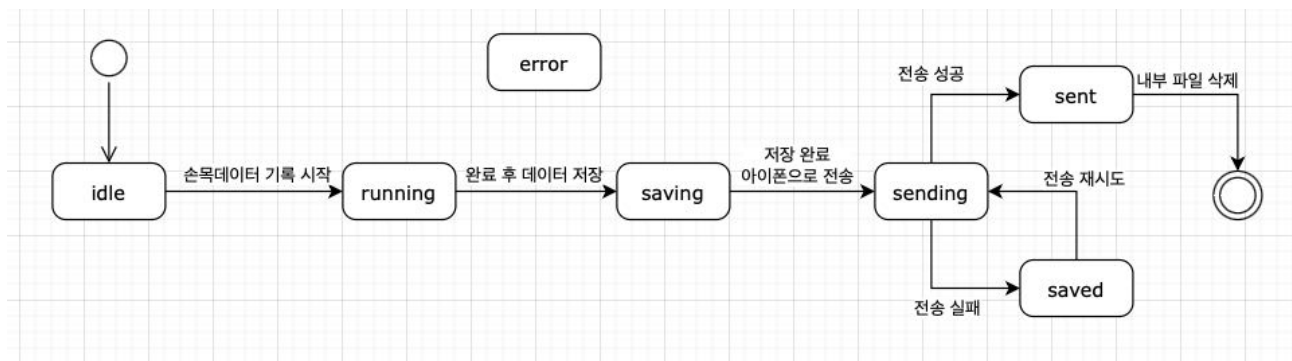
◇ 상태 다이어그램

1. SwingRecordManagerPhone



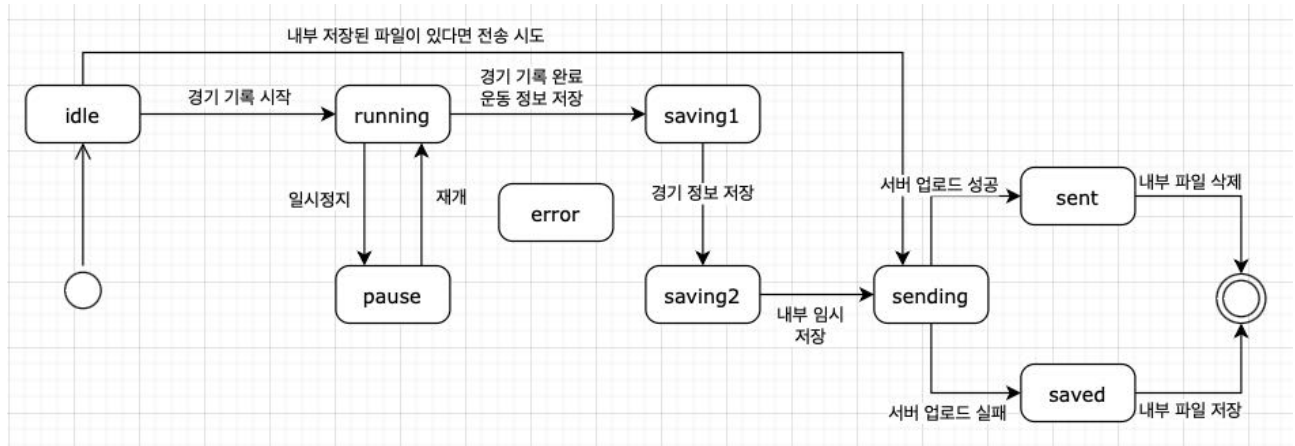
녹화 시작 신호를 받거나 직접 녹화를 시작하면 watchOS로 녹화 시작 신호를 보내며 녹화를 시작한다. 영상을 저장한 이후, recieving 상태가 되면 watchOS로부터 IMU데이터를 수신받고, 영상과 함께 서버에 업로드한다.

2. SwingRecordManagerWatch



녹화 시작 신호를 받거나 직접 녹화를 시작하면 iOS로 녹화 시작 신호를 보내며 녹화를 시작한다. 녹화가 종료되면 iOS로 IMU데이터를 전송한다.

3. WorkoutManager



경기 기록 중 기록을 일시정지할 수 있다. 운동 정보와 경기 정보를 저장하면 IMU 데이터를 메타데이터와 함께 서버에 '지연 업로드' 한다.

지연 업로드



IMU 데이터와 메타데이터 파일을 우선 내부 스토리지에 저장하고, 서버에 업로드를 시도한다. 서버에 업로드하였다면 이후 남은 파일을 삭제하고, 업로드하지 못했다면 내부에 저장하고 상태를 종료한다. 이후 자동으로 서버와의 연결을 체크하여 업로드 가능하다면 위 작업을 다시 시도한다.