

# 융합 소프트웨어 프로젝트 4조 결과 보고서

2019250012	김주원
2019210117	김보건
2019250016	박건희
2019250035	이동현

## 1. 프로젝트 목표

covid-19 확산세의 감소에 따라 다중 이용시설 규제 완화 및 건강 관리에 대한 관심이 높아졌으며 헬스 트레이닝에 대한 관심 또한 증가하였다.

이에 헬스를 처음 접하는 사람의 경우 개인 블로그, 유튜브 등에 산재해 있는 정보들을 취합, 관리함에 있어 어려움을 느끼고 있다는 점에서 착안하여 헬스 트레이닝을 위한 루틴의 제공과 운동시간 관리를 보조하는 어플리케이션의 제작을 목표로 하였다.

## 2. 기능 리스트

### 목표 기능 리스트

1. 사용자 정보 입력, 관리
2. 프리셋 운동 루틴 생성, 관리
3. 캘린더에 운동 정보(사진, 메모) 저장
4. 프로필 설정
5. 업적 달성
6. 운동 루틴 타이머
7. 운동 영상 링크
8. 운동 중량 계산
9. 운동 루틴 추천
10. 운동 결과 확인

### 달성 기능 리스트

1. 사용자 정보 입력, 관리
2. 프리셋 운동 루틴 생성, 관리
3. 일일 루틴 생성, 사용 기능
4. 캘린더에 운동 정보(메모) 저장
5. 프로필 설정
6. 업적 달성
7. 운동 루틴 타이머
8. 운동 영상 링크
9. 운동 루틴 추천
10. 운동 결과 확인
11. 위젯 기능

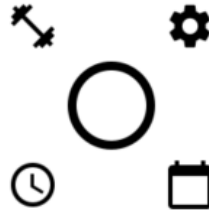
### 3. 페이지/ 기능별 설명

#### 메인 화면

2023/11/30  
07:28



2023/11/30  
07:29

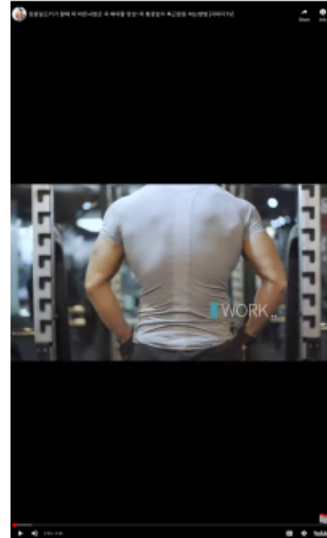


▲ 그림 1. 앱의 메인 화면      |      ▲ 그림 2. 메인 버튼을 눌렀을 때 나오는 화면

**개요(요약) :** 앱을 제일 먼저 실행하면 보이는 화면. 사용자의 **편의성**을 높이고 **핵심적인 네비게이션 역할**을 수행한다.

- 앱을 실행하면 년/월/일과 현재 시간을 표시해주는 **디지털 시계**와 **메인 버튼**이 존재.
- **메인 버튼**(그림 1의 빨간 네모 안 동그라미)을 누르면 그림 2와 같이 아이콘이 나타남.
- 아이콘을 누르면 각각 할당된 화면(설정, 루틴, 운동, 캘린더)으로 이동함.

## 유튜브 버튼



▲ 그림 3. 타이머 페이지의 유튜브 버튼 | ▲ 그림 4. 유튜브 영상

개요(요약) : 타이머 화면에 유튜브 버튼을 누르면 각 운동에 맞는 운동 영상을 시청 가능하다.

- 빨간 네모 표시가 되어 있는 버튼은 운동 영상을 시청할 수 있는 버튼.
- 유튜브 버튼을 누르면 현재 화면에 표시된 운동의 이름을 가져와 이름에 맞는 영상을 재생한다.

## 결과 페이지



▲ 그림 5. 결과 창



목표를 달성하셨습니다! 더 높은 강도로 루틴을 수정해 보시겠어요?

아니오 예

▲ 그림 6. 팝업 창



▲ 그림 7. 날짜 창

**개요(요약) :** 운동을 완료하면 사용자의 행동에 따라 **운동 강도를 올릴 수 있고**, 특정한 날짜에 어떤 운동을 했는지 **직관적으로 확인** 할 수 있다.

- 타이머에서 정지 버튼을 누르면 그림 5와 같이 결과 창이 나타난다.
- 결과 창에는 오늘 진행한 운동 이름들이 **체크박스**와 함께 나타난다.
- 루틴에 기재한 대로 운동을 완료했으면 체크박스에 체크하고, 운동을 완료하지 못했다면 체크박스에 체크를 하지 않고 확인 버튼을 누른다.
- 확인 버튼을 누르면 체크박스의 유/무에 따라 운동 완료 또는 미완료를 저장한 후 캘린더에 운동 완료한 운동 종목의 이름을 기록한다.
- 체크박스가 하나라도 체크되어 있다면, 그림 6과 같은 팝업 창을 띄워 사용자에게 더 높은 강도의 운동을 시도할 것인지 묻는다.
- 예 버튼을 누르면 완료한 운동에 따라 무게/운동 시간/횟수 중 하나를 늘려 Routine 테이블에 저장되어 있던 기존 루틴을 수정해 저장한다.

- 또한 체크된 운동 이름에 따라 계산을 수행, AchieveInfo 테이블에 저장해 매 운동마다 업적을 갱신 가능하도록 했다.

# 일일 운동 루틴 설정 화면

개요 요약 : 사용자가 그 날에 할 운동을 설정하는 페이지, 각 부위와 종목에는 텍스트와 더불어 해당하는 사진과 함께 표기

운동 부위 선택

가슴

등

하체

어깨

이두


삼두


복근


선택한 운동


운동 시작


▲ 그림 8. 운동 부위 선택 페이지


푸쉬업


벤치프레스


풀업


데드리프트

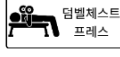
스쿼트맨몸


스쿼트바벨


사이드레터럴레이즈


밀리터리프레스


체스트프레스


덤벨체스트프레스


랫풀다운

바벨로우

레그프레스

덤벨숄더프레스

머신숄더

로우머신

선택한 운동

선택한 운동

선택한 운동

선택한 운동

확인

취소

확인


취소


확인


취소

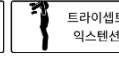
확인


취소


바벨컬


덤벨컬


푸쉬다운

트라이셉트익스텐션

윗몸일으키기

플랭크

머신컬

크런치

선택한 운동

선택한 운동

선택한 운동

확인

취소

확인

취소

확인

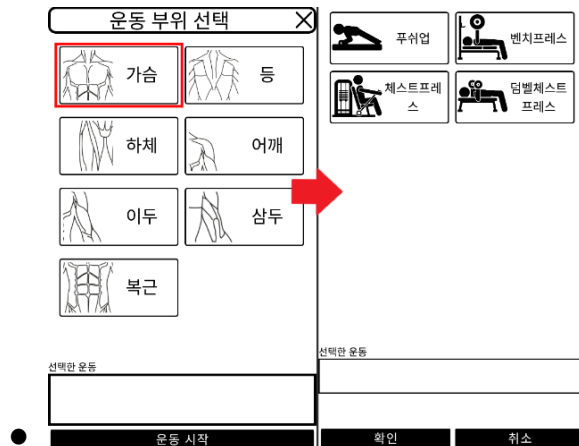
취소

▲ 그림 9. 운동 종목 선택 페이지 (가슴, 등, 하체, 이두, 삼두, 복근)

- RecyclerView와 각 운동 부위표시를 text와 image를 사용해 운동 부위 배치 및 drawable의 xml을 사용하여 각 운동 부위마다 테두리 생성

## 일일 운동 루틴 설정

개요 요약 : 사용자가 그 날에 할 운동을 설정하는 페이지, 각 부위와 종목에는 텍스트와 더불어 해당하는 사진과 함께 표기

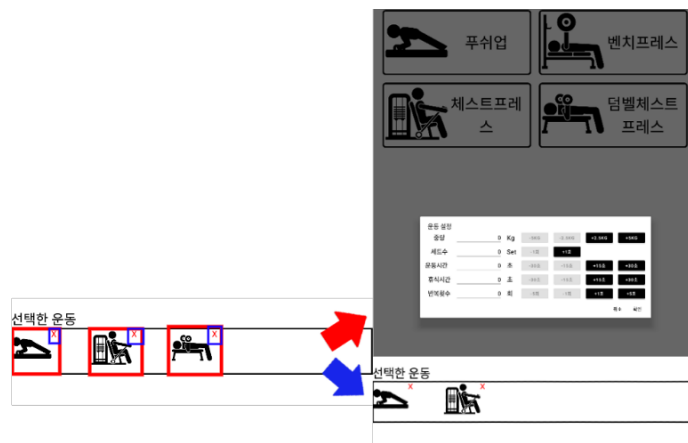


▲ 그림 10. 운동 부위, 종목 선택 페이지

- 운동 부위 화면에서 해당 부위를 클릭 시, intent를 사용해 각 부위에 맞는 종목 페이지로 이동



▲ 그림 11 운동 설정



▲ 그림 12 운동 삭제 및 세부설정

- 사용자는 해당 화면에서 원하는 운동 종목을 선택할 수 있음
- 각 종목을 선택 할 때마다 동적으로 ImageView(운동그림)와 TextView(x)을 하단 Layout에 생성 후 각 image마다 number부여
- 하단 레이아웃 image(빨간색) : 중량, 세트 수 등 운동에 대한 세부설정 dialog 생성
- X(파란색) : 해당 운동 종목 Layout에서 삭제와 동시에 어레이에 저장된 정보도 삭제

운동 설정					
중량	0	Kg	-5KG	-2.5KG	+2.5KG +5KG
세트수	0	Set	-1회	+1회	
운동시간	0	초	-30초	-15초	+15초 +30초
휴식시간	0	초	-30초	-15초	+15초 +30초
반복횟수	0	회	-5회	-1회	+1회 +5회
취소 확인					

▲ 그림 13 처음상태

운동 설정					
중량	2.5	Kg	-5KG	-2.5KG	+2.5KG +5KG
세트수	1	Set	-1회	+1회	
운동시간	15	초	-30초	-15초	+15초 +30초
휴식시간	30	초	-30초	-15초	+15초 +30초
반복횟수	0	회	-5회	-1회	+1회 +5회
취소 확인					

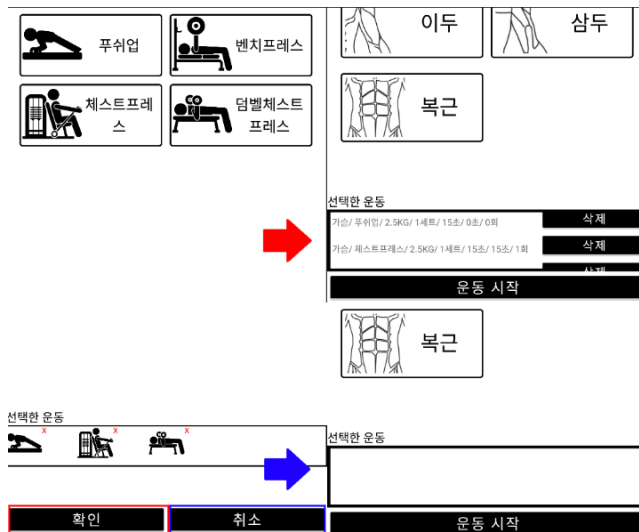
▲ 그림 14 운동 세부 설정 (증가)

운동 설정					
중량	2.5	Kg	-5KG	-2.5KG	+2.5KG +5KG
세트수	1	Set	-1회	+1회	
운동시간	15	초	-30초	-15초	+15초 +30초
휴식시간	0	초	-30초	-15초	+15초 +30초
반복횟수	0	회	-5회	-1회	+1회 +5회
취소 확인					

▲ 그림 15 운동 세부 설정 (감소)

- 사용자는 각 목록에 대한 EditText에 원하는 수치를 직접 입력과 더불어 버튼을 통해 증감 가능
- 각 설정 목록이 0일 경우 -(감소) 버튼 비활성화(회색) -> 그림 13
- +(증가) 버튼 : 각 버튼에 적혀 있는 수치만큼 증가 -> 그림 14
- -(감소) 버튼 : 각 버튼에 적혀 있는 수치만큼 감소, 이 때 EditText의 수치가 0이면 다시 -(감소) 버튼 비활성화 -> 그림 15
- “확인” 버튼 : chestModel 및 앞서 부여한 image number로 해당 이미지의 대한 어레이를 찾음
- 이때, 기존 이미지의 어레이가 존재 ( 기존에 설정했던 운동이 존재 ) 하면 다이얼로그에서 입력된 값을 얻어와서 어레이를 수정
- 존재하지 않으면 새로운 어레이(newSetChestList) 추가한 뒤 입력된 값을 저장 후에 ChestDetailList에 저장
- “취소” 버튼 : 입력된 수치 전부 초기화 후 dialog창 종료





▲ 그림 16 운동 설정 확인 및 취소

- 확인(빨간색) - 설정한 운동 종목들을 저장한 ChestDetailList를 Intent로 운동부위 화면으로 정보 전달 후 운동 종목 설정화면 finish()로 종료
- 이때 Flag설정(FLAG\_ACTIVITY\_REORDER\_TO\_FRONT)을 통해 새로운 운동부위 화면을 불러오는 것이 아닌 기존의 운동부위 화면을 불러옴
- 운동 부위 화면에서는 받아온 ChestDetailList과 더불어 각 부위에 해당하는 \_DetailList들을 finalExerciseList에 2차원 어레이로 저장
- 운동 시작 - intent로 finalExerciseList를 타이머 화면으로 넘겨줌과 동시에 타이머 화면으로 넘어감
- 취소(파란색) - 저장된 ChestDetailList 저장하지않고 운동 종목 화면을 finish()로 종료

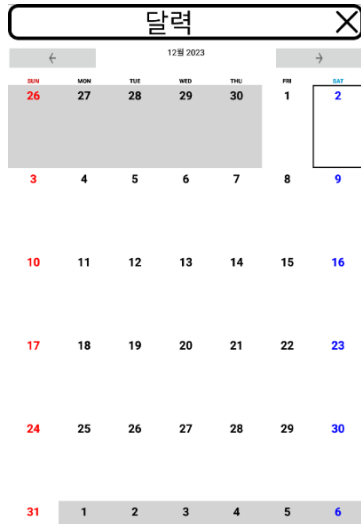
선택한 운동	선택한 운동
가슴/ 푸쉬업/ 2.5KG/ 1세트/ 15초/ 0초/ 0회	가슴/ 체스트프레스/ 2.5KG/ 1세트/ 15초/ 15초/ 1회
가슴/ 체스트프레스/ 2.5KG/ 1세트/ 15초/ 15초/ 1회	가슴/ 덤벨체스트프레스/ 2.5KG/ 1세트/ 15초/ 15초/ 1회

▲ 그림 17 최종 운동 표시 화면

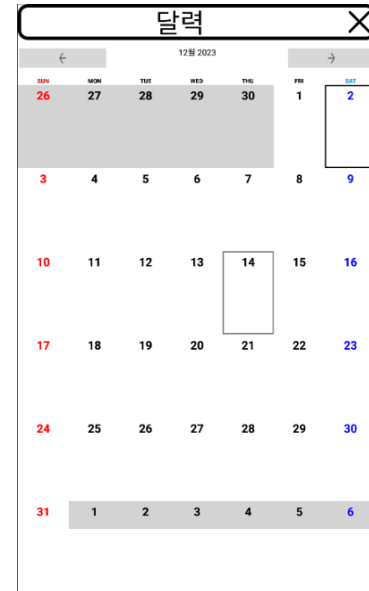
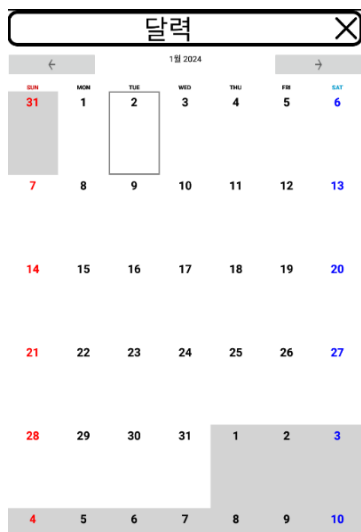
- 선택한 운동종목이 너무 많아 화면을 벗어난 경우 위로 슬라이드를 통해 전체 종목 확인 가능

## 캘린더

개요 (요약) : 사용자가 완료한 운동을 표시해주는 캘린더, 운동한 날짜에 맞춰 운동 정보 저장



▲ 그림 18 캘린더 (오늘 날짜)



▲ 그림 19 캘린더 (날짜 선택)

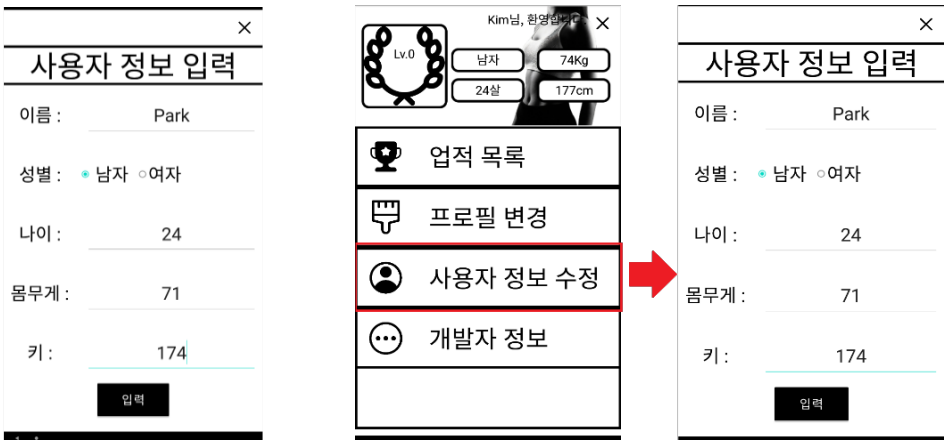
▲ 그림 20 캘린더 (월 이동)

- recycler view를 통해 캘린더를 제작
- drawable폴더에 xml을 생성해 이전달과 다음달로 넘어가게하는 버튼ui 생성
- 구현하고자 하는 MainActivity.xml에 요일표시와, 해당 년,월을 표시하는 textview와 recyclerview를 넣고 recyclerview에 보여질 itemView( item\_cell.xml)을 생성 후 해당파일 에 캘린더날짜가 표시될 textView 삽입
- recyclerview 어댑터 클래스를 생성하고 (CalenderAdapter.java) 현재 년,월이 같으면 하얀

색, 다르면 연한회색, 오늘 날짜는 검은색 사각 테두리로 표시. 텍스트 색은 토요일은 파랑색, 일요일은 빨간색, 나머지(평일)은 검은색으로 표현 -> 그림 18

- 사용자가 임의의 날짜 클릭 시 회색테두리로 표시 -> 그림 19
- MainActivity에서 CalenderAdapter와 recylerview를 연결. 양쪽 버튼(<- , ->)을 누르면 다음월 또는 이전월로 변경 -> 그림 20

# 사용자 정보 입력



▲ 그림 21. 사용자 정보 입력 페이지 | ▲ 그림 22. 사용자 정보 수정 페이지 경로  
개요(요약): 사용자 정보를 입력받는 작성 페이지. 작성된 내용은 내부 DB에 저장된다.

- 사용자의 '이름', '성별', '나이', '몸무게', '키'에 대한 정보를 입력받는다.
- 입력된 정보는 내부 DB의 'UserInfo' 테이블에 저장된다.
- 어플을 최초 실행 하는 경우에 자동으로 사용자 정보 입력 페이지로 넘어가고, 정보 수정을 원하는 경우에는 설정 페이지의 '사용자 정보 수정' 버튼을 클릭하면 그림 22 페이지로 이동하여 새롭게 사용자 정보를 입력할 수 있다.
- 사용자 정보를 수정하지 않고 나가고 싶다면 우상단의 X버튼을 통해 닫을 수 있다.

## 사용자 정보 입력 페이지\_초보자 루틴

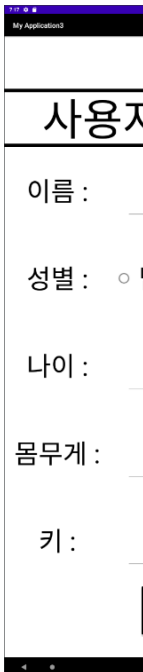
다.

개요: 근력 운동에 익숙하지 않은 초보자 전용 루틴을 자동 생성한다.

- 사용자 정보 입력 페이지의 '?' 버튼을 누르면 'RoutineReq' 테이블에 저장된 초보자용 루틴을 'Routine' 테이블에 복사한다. 이미 초보자 루틴이 복사가 되어있을 경우 토스트 메시지를 통해 표시한

그림 38 초보자 루틴 복사

## 타이머 페이지





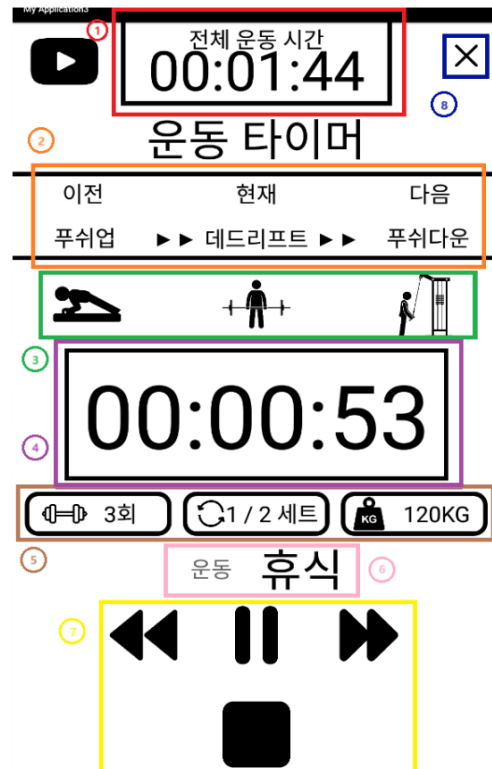
▲ 그림 233. 타이머 이동 경로



| 그림 24. 타이머 페이지 ▲

**개요(요약):** 운동 루틴 정보를 받아와 시간을 측정해주는 타이머 페이지. 타이머에는 각 운동별 시간 이외에도 전체 시간과 운동명, 심볼, 횟수, 세트 수, 중량이 함께 표기된다.

- ① 번(빨간색)은 한 운동 루틴의 전체 시간을 보여준다. 각 운동 시간과는 별개로 일시정지나 휴식 시간에도 타이머가 계속 작동하여 시간이 흐른다.
- ② 번(주황색)은 이전, 현재, 다음 운동명을 출력한다.
- ③ 번(초록색)은 운동명에 따라 심볼 이미지를 표시한다. 22종의 운동별 심볼이 있다.
- ④ 번(보라색)은 각 운동(휴식)별 시간을 출력한다.
- ⑤ 번(갈색)은 각 운동별로 시행해야 할 운동횟수, 세트 수, 중량을 출력한다.
- ⑥ 번(분홍색)은 현재 상태가 운동 진행 상태인지, 한 세트를 마치고 휴식 상태인지를 알려주는 상태 표시등 역할을 한다.
- ⑦ 번(노란색)은 사용자가 타이머를 직접 조작할 수 있는 버튼이다. 실행과 일시 정지, 넘어가기, 돌아가기가 존재하고, 운동을 마치면 종료 버튼을 통해 결과 페이지로 넘어간다.
- ⑧ 번(파란색)은 타이머 동작을 원치 않을 경우 닫을 수 있는 버튼이다.



▲ 그림 25. 타이머

## 설정 페이지

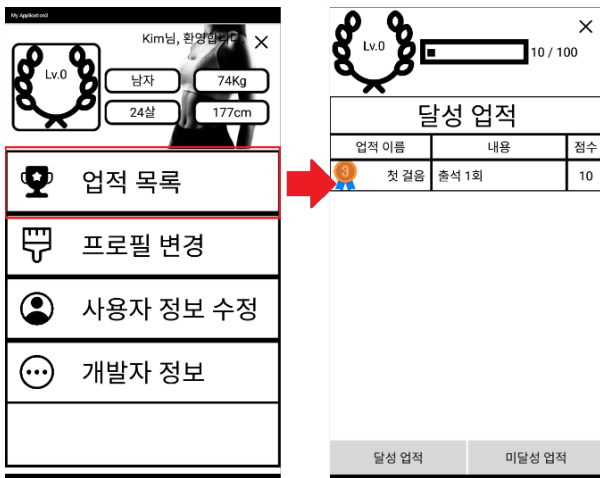


▲ 그림 26. 설정 이동 경로 & 초기 설정 페이지 | 그림 27. 84레벨 사용자의 설정 ▲

**개요(요약):** 어플의 상세한 부분들을 조정할 수 있는 설정 페이지. 설정 페이지에서는 사용자의 정보를 프로필 카드로 확인할 수 있고, 설정 페이지에서 업적 목록, 프로필 설정, 사용자 정보 수정, 개발자 정보 페이지로 이동할 수 있다.

- 상단에는 사용자의 프로필이 출력된다. 사용자의 업적 레벨과 성별, 나이, 키, 몸무게가 함께 출력된다.
- 프로필 카드의 이미지는 성별에 따라 다른 이미지가 출력된다. 출력되는 이미지의 종류는 10종이 존재하고, 프로필 설정 페이지에서 확인 가능하다.
- 설정 페이지에는 업적 목록, 프로필 설정, 사용자 정보 수정 페이지로 이동할 수 있는 버튼이 존재하며, 각 페이지를 상징하는 아이콘은 버튼 좌측에 배치시켜 두었다.
- 이전의 페이지들과 마찬가지로 X버튼을 통해 페이지를 닫을 수 있다.

## 업적 페이지



▲ 그림 28. 업적 페이지 이동 경로

달성 업적			미달성 업적		
업적 이름	내용	점수	업적 이름	내용	점수
첫 걸음	출석 1회	10	언더아머 허용	데드리프트 1RM 200kg 달성	700
			3대 300	데드리프트 1RM 240kg 달성	1000
			3대 400	벤치프레스 1RM 40kg 달성	100
			3대 500	벤치프레스 1RM 70kg 달성	250
			3대 1000	벤치프레스 1RM 100kg 달성	400
			3대 1500	벤치프레스 1RM 120kg 달성	700
			3대 2000	벤치프레스 1RM 150kg 달성	1000
			3대 2500	풀업 연속 5회 달성	50
			3대 3000	풀업 연속 10회 달성	100

▲ 그림 29. 업적 달성과 미달성 페이지 | 그림 30. 84레벨 달성 페이지 예시 ▲



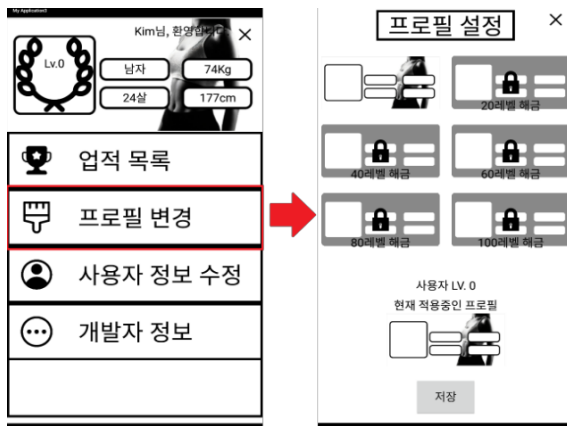
▲ 업적 메달 이미지 6종

▲ 업적 레벨 이미지 15종 (1종 누락됨)

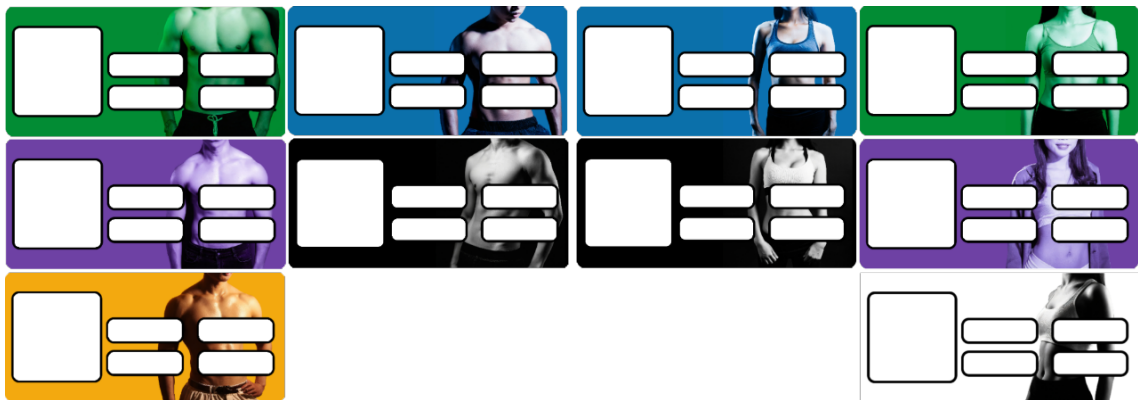
개요(요약): 운동 목표점을 설정해주고, 업적마다 메달 이미지를 출력한다. 업적 달성으로 쌓인 성취도는 사용자의 레벨이 올라가도록 설계되어 높은 레벨에서는 더 퀄리티 높은 레벨 이미지가 출력되도록 만들었다.

- 사용자의 업적(성취) 레벨을 텍스트와 레벨 이미지로 출력한다.
- 사용자의 업적 레벨 달성에 쓰이고 남은 성취도는 바 형태로 출력된다.
- 사용자의 업적 레벨 달성 이미지는 총 16종으로, 10레벨 단위로 150레벨까지 다양한 레벨 달성 이미지가 존재한다. 레벨 이미지는 그림 '업적 레벨 이미지 15종'에 나와있다.
- 업적은 총 71종이 존재하고, 윗몸일으키기, 푸쉬업, 플랭크, 데드리프트, 스쿼트, 벤치프레스 등 17가지의 항목으로 분류되어있다.
- 업적은 난이도에 따라 얻는 점수(성취도)와 메달 이미지가 다르다. 업적에 사용되는 메달 이미지는 6종이 있으며, 그림 '업적 메달 이미지 6종'에 나와있다.
- 업적의 달성 유무에 따라 달성 업적과 미달성 업적으로 분류되어 탭에 따라 출력된다.
- 이전 페이지들과 마찬가지로 우상단의 X버튼을 통해 페이지를 닫을 수 있다.

## 프로필 설정 페이지

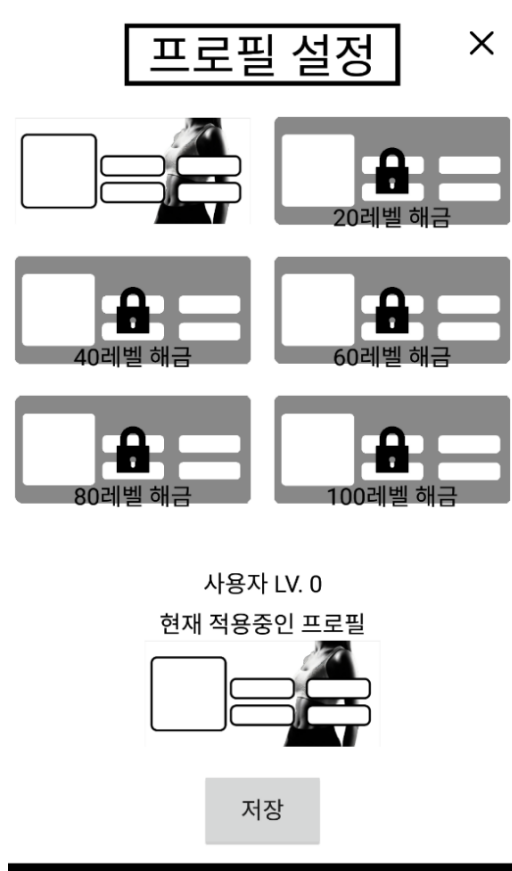


▲ 그림 31. 프로필 설정 경로



▲ 그림 32. 프로필 사진 10종 (남자 5종, 여자 5종)





▲ 그림 33. 0레벨 사용자 프로필 설정

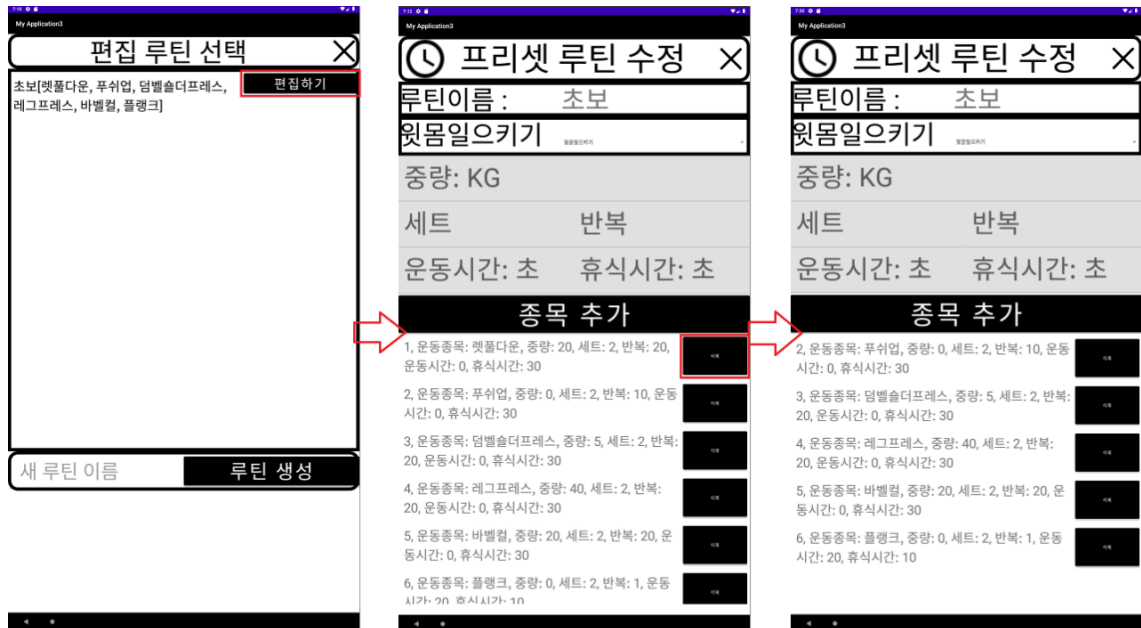


▲ 그림 34. 84레벨 사용자 프로필 설정

**개요(요약):** 설정 페이지에서 띄울 **프로필**을 설정하는 페이지. 총 **6종의 프로필**이 존재하며, 업적 레벨에 따라 사용할 수 있는 프로필이 늘어난다. 사용자 정보에서 입력한 **성별**에 따라 다른 프로필이 적용된다.

- 설정 페이지에서 띄울 **프로필 카드**를 선택할 수 있다.
- 사용자가 달성한 업적 레벨에 따라 새로운 프로필 카드가 해금되며, 해금 조건 달성 시 잠금 아이콘과 텍스트가 없어지고 프로필 카드 원본 이미지가 출력된다. 해금된 프로필 카드는 전부 색상과 이미지가 다르다.
- 기본적으로 흰색이 해금되고, 20레벨에 초록색, 40레벨에 보라색, 60레벨에 노란색, 80레벨에 파란색, 100레벨에 검은색 프로필 카드가 해금된다.
- 선택한 색깔에 따라 성별 상관없이 출력되는 공용 프로필카드 2종(흰색, 노란색)이 존재하고, 성별에 따라 서로 다른 이미지가 출력되는 전용 프로필카드가 성별 별로 각 4종(초록색, 보라색, 파란색, 검정색)으로 총 8종이 존재한다. 예시는 그림 34에 나와있다.
- 사용자의 업적 레벨과 선택중인 프로필 카드는 하단에서 확인가능하고, 저장 버튼을 누르면 설정창의 프로필이 선택한 프로필 카드로 바뀐다.

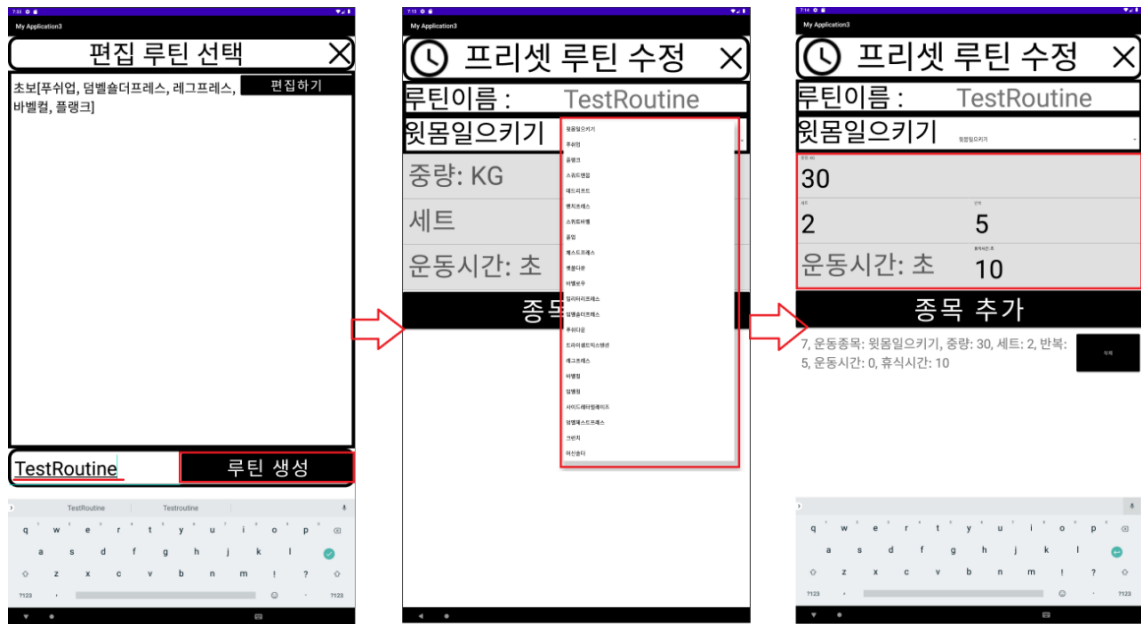
## 편집 루틴 선택 페이지



▲그림 35. 루틴 편집 경로

개요: DB에 저장 되어있는 루틴을 선택하거나 새로운 루틴을 생성하여 운동종목을 추가하고 삭제하며 변경사항을 DB에 저장한다.

- 1. 리사이클러 뷰를 사용한 프리셋 루틴 표현, DB의 'Routine' 테이블에 저장된 루틴의 이름과 루틴에 포함된 운동 이름을 표시한다.
- 2. '편집하기' 버튼을 누르면 해당 루틴의 상세정보가 표시되는 레이아웃으로 넘어간다. 'Routine' 테이블에 저장된 루틴의 각 운동 별 저장번호, 이름, 중량, 세트, 반복, 운동시간, 휴식시간을 가져와 순서대로 리사이클러 뷰에 적재한다.
- 3. '삭제' 버튼을 누르면 해당하는 리사이클러 뷰 포지션의 저장번호를 참조하여 'Routine' 테이블에 DELETE Query문을 실행한다. 이후 리사이클러 뷰에 다시 루틴을 적재, 수정 사항은 반영하여 표시한다.
- 4. 'X' 버튼을 통해 페이지를 닫고 이전 페이지로 돌아갈 수 있다.



▲그림 36. 새 루틴 생성 경로

개요: 새로운 루틴을 생성하고 운동종목을 추가하여 사용자만의 루틴을 관리한다.

- 1. 새롭게 생성할 루틴의 이름을 입력한다. ‘루틴 생성’버튼을 누르면 ‘프리셋 루틴 수정’ 페이지로 넘어가며 생성한 루틴의 이름을 표시한다. 이미 저장되어 있는 루틴과 이름이 중복 되거나 이름을 입력하지 않고 버튼을 누를 경우 “이미 사용중이거나 빈 이름입니다.”라는 토스트 메시지를 출력한다.



▲그림 37. 루틴 이름 중복/공백 메시지

- 2. 스피너를 이용해 루틴에 추가할 운동종목을 선택한다.
- 3. 추가할 운동종목의 세부사항을 입력 받는다. ‘종목 추가’ 버튼을 누르면 입력된 정보를 ‘Routine’ 테이블에 저장한다.
- 4. ‘X’ 버튼을 통해 페이지를 닫고 이전 페이지로 돌아갈 수 있다.

## 운동 루틴 선택 페이지

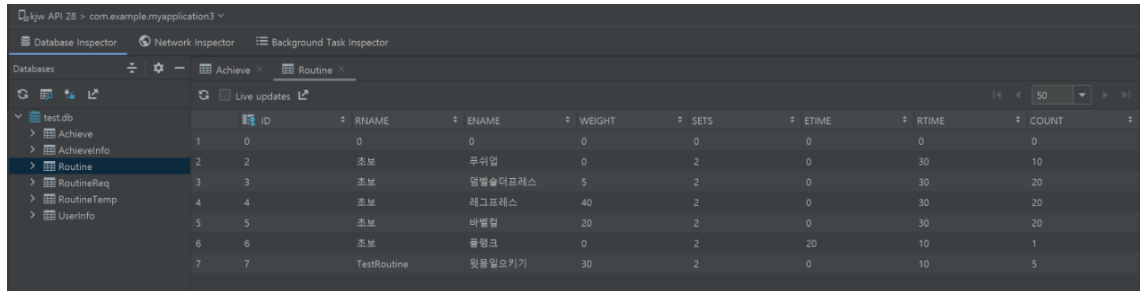


▲그림 39 운동 루틴 선택

개요: 운동 할 루틴을 선택하여 타이머로 연결한다.

- 'Routine' 테이블에 저장된 루틴이 루틴이름 + [운동종목]의 형태로 리사이클러 뷰에 적재되어 표시된다.
- 불러오기 버튼을 누르면 해당하는 포지션의 루틴 이름을 기준으로 'Routine' 테이블에 저장된 루틴 데이터를 복사하여 초기화 된 'RoutineTemp' 테이블에 저장한다.
- 타이머 페이지는 이전 페이지에서 Intent로 넘어온 루틴이름을 기준으로 'RoutineTemp' 테이블의 각 컬럼 값을 리스트로 가져와 사용한다.
- 일일 루틴 생성 버튼을 누르면 그날의 상황에 따라 만들고 싶은 루틴을 만들어 운동 할 수 있다.

## 데이터 베이스 구조



The screenshot shows a database inspector interface with the 'Routine' table selected. The table has columns: ID, RNAME, ENAME, WEIGHT, SETS, ETIME, RTIME, and COUNT. The data is as follows:

ID	RNAME	ENAME	WEIGHT	SETS	ETIME	RTIME	COUNT
1	초보	무위업	0	0	0	0	0
2	초보	달달솔더프레스	5	2	0	30	10
3	초보	레그프레스	40	2	0	30	20
4	초보	바벨컬	20	2	0	30	20
5	초보	플랭크	0	2	20	10	1
6	초보	윗몸일으키기	30	2	0	10	5

### 1. Routine: 사용자가 생성한 프리셋 루틴이 저장되는 테이블

컬럼 : [ID, RNAME(루틴이름), ENAME(운동이름), WEIGHT, SETS, ETIME(운동시간), RTIME(휴식시간), COUNT]

설명: 사용자가 생성한 루틴이 저장된다. 하나의 행에는 루틴에 속한 하나의 운동종목이 들어간다. 같은 루틴에 속하는 운동종목은 모두 같은 'RNAME' 속성 값을 가지고 있다. 동일한 루틴에 속한 다른 운동을 구분하는 방법은 'RNAME'과 'ENAME'속성의 조합을 기준으로 한다.

ID	RNAME	ENAME	WEIGHT	SETS	ETIME	RTIME	COUNT
1	초보	무위업	0	2	0	30	10
2	초보	달달솔더프레스	5	2	0	30	20
3	초보	레그프레스	40	2	0	30	20
4	초보	바벨컬	20	2	0	30	20

### 2. RoutineTemp: 프리셋 루틴이나 임시 루틴의 데이터를 타이머에 넘기기 위해 거쳐가는 테이블

컬럼 : [ID, RNAME(루틴이름), ENAME(운동이름), WEIGHT, SETS, ETIME(운동시간), RTIME(휴식시간), COUNT]

설명: 오늘의 루틴 선택 페이지와 일일 루틴 생성 페이지에서 타이머로 넘어갈 때 운동을 하기로 선택한 루틴을 해당 테이블에 저장한다. 이때 테이블을 초기화 하고 저장한다. 타이머는 어느 페이지에서 넘어오더라도 'RoutineTemp' 테이블에 저장된 각 컬럼 값의 리스트를 받아 동작한다.

ID	RNAME	ENAME	WEIGHT	SETS	ETIME	RTIME	COUNT
1	초보	윗몸일으키기	20	2	0	30	20
2	초보	무위업	0	2	0	30	10
3	초보	달달솔더프레스	5	2	0	30	20
4	초보	레그프레스	40	2	0	30	20
5	초보	바벨컬	20	2	0	30	20
6	초보	플랭크	0	2	20	10	1

### 3. RoutineReq: 초보자용 루틴이 저장되어 있는 테이블

컬럼 : [ID, RNAME(루틴이름), ENAME(운동이름), WEIGHT, SETS, ETIME(운동시간), RTIME(휴식시간), COUNT]

설명: 루틴을 생성하기 어려운 사용자를 위한 초보자용 루틴이 저장된 테이블로 사용자 정보 수정 페이지의 '?' 버튼을 누르면 해당 테이블의 데이터가 복사되어 'Routine' 테이블에 저장된다.

별도의 테이블 수정은 하지 않는다.

### 4. Achieve: 업적의 정보와 달성조건이 저장된 테이블

컬럼: [ACHNAME, ACHINFO, SCORE(획득점수), ACHCAT(업적 종류), ACHCON(달성 기준수치)]

	ID	ACHNAME	ACHINFO	SCORE	ACHCAT	ACHCON
1	1	첫 걸음	출석 1회	10	출석	1
2	2	시작이 반	출석 7회	50	출석	7
3	3	겉이지 않는	출석 15회	100	출석	15
4	4	스스로 잘하는	출석 30회	200	출석	30
5	5	노력의 천재	출석 90회	300	출석	90
6	6	전문 헬스인	출석 180회	500	출석	180
7	7	함께한지 1년	출석 365회	800	출석	365
8	8	시작은 가볍게	윗몸일으키기 연속 30회 달성	40	윗몸일으키기	30
9	9	자신감이 쌓이는	윗몸일으키기 연속 50회 달성	80	윗몸일으키기	50
10	10	우매함의 봉우리	윗몸일으키기 연속 80회 달성	150	윗몸일으키기	80
11	11	원만한 수련법 - 윗몸	윗몸일으키기 연속 100회 달성	300	윗몸일으키기	100
12	12	진지한 자세 - 윗몸	윗몸일으키기 연속 150회 달성	400	윗몸일으키기	150

설명: 업적에 대한 정보가 저장된 테이블, 'AchieveInfo' 테이블과 'ACHCON'값을 업적 종류 별로 비교하여 달성 여부를 판단, 점수를 획득한다.

	ID	출석	윗몸일으키기	무위업	플랭크	스쿼트맨	데드리프트	벤치프레	스쿼트바	물업	체스트프	엳들다운	바벨로우
1	1	1	0	20	0	0	0	0	0	0	0	400	0

5. AchieveInfo: 사용자가 달성한 업적종류별 수치가 저장된 테이블

컬럼: [출석, 윗몸일으키기,.....(업적 종류)]

설명: 사용자가 달성한 업적 종류별 수치로 결과창에서 완료한 운동종목을 선택하면 해당하는 업적종류의 수치가 상승한다. 업적달성의 기준이 된다.

	ID	NAME	AGE	HEIGHT	WEIGHT	ATTENDENCE	LEVEL	SEX	NEWSTART	PROFILECOLOR
1	1	0	0	0	0	0	0	0	1	흰색

6. UserInfo: 상용자의 정보가 저장되는 테이블

컬럼: [ID, NAME, AGE, HEIGHT, WEIGHT, ATTENDENCE(출석), LEVEL, SEX, NEWSTART, PROFILECOLOR]

설명: 앱을 사용하는 사용자의 정보가 저장되는 테이블, 신체정보는 사용자 정보 입력 페이지를 통해 입력 받는다. 'NEWSTART' 컬럼은 사용자가 해당 앱을 최초로 사용하는 지 검사하기 위한 값으로 최초 실행 시 0이 저장되어 있으며 바로 사용자 정보 입력 페이지로 넘어간다. 이후 1을 저장한다.

## 위젯 설정



그림 40. 위젯

개요: 기기의 바탕화면에 위젯을 생성하여 바로 운동 선택 페이지로 이동할 수 있다.

## 빌드 공유

개요: 단순 변경사항은 카카오톡을 통해 공유, 최종 빌드는 깃허브에 저장

깃허브 주소:

[https://github.com/chungsim/team4\\_SC/blob/main/UI%EC%88%98%EC%A0%95%EB%B3%B8\\_11\\_30.zip](https://github.com/chungsim/team4_SC/blob/main/UI%EC%88%98%EC%A0%95%EB%B3%B8_11_30.zip)

## 느낀 점 및 감상

### 1. 김주원

이번 팀별 프로젝트를 진행하기에 앞서 걱정이 제일 먼저 들었다. 자신의 실력이 그리 높지 않다는 생각과 다른 사람들과 어느 정도의 차이가 있을지 모르는 불안감이 있었고 그 무엇보다 여러 사람과 함께하는 활동이 오랜만이었던 것이 큰 부분을 차지했던 것 같다.

프로젝트가 시작되고 우리 조가 헬스 어플리케이션을 제작하기로 결정하며 직함으로나마 팀장을 맡게 되었다. 어플리케이션에 대한 아이디어를 정리하고 가닥을 잡아가는 과정은 순탄했다고 생각한다. 최초 회의에서는 개발 툴을 beeware와 안드로이드 스튜디오 중 어느 것으로 할 지 고민 하였지만 이전 학기에서 배운 안드로이드 스튜디오를 사용하기로 하였다. 이는 실로 잘한 결정이라고 생각한다. 팀원 모두가 강의를 통해 어느 정도 프로그램에 익숙해져있는 상태라 프로젝트 초기에 많은 시간을 절약할 수 있었다고 생각한다.

어플에 대한 아이디어 회의를 할 때 개인적으로 가장 중요하게 생각한 것은 통일이었다. 4명이 리스트에 작성한 기능을 분배하여 제작하기로 하였을 때 각 기능 사이의 데이터 교환 방식이 통일되지 않는다면 개발이 오래 걸릴 것이라고 생각했다. 그래서 우리 조는 모든 기능에 필요한 데이터를 나열하고 이를 효과적으로 전달 할 수 있는 데이터 형식부터 토의했다. 토의 결과 어플에서 가장 빈번하게 일어나는 데이터 교환은 루틴 데이터로 보고 그 구조를 2중 리스트로 정하였다. DB에는 테이블의 행마다 하나의 루틴이 들어가도록 설계하였다.

이후 개발에 들어가면서 이 구조를 완전히 바꾸게 되었다. DB 테이블의 한 행에 루틴 전체를 넣으면 쿼리문을 통해 가져온 루틴의 데이터를 모두 가져오는 것은 문제가 없었으나 타이머나 캘린더, 결과창과 같이 루틴에 속한 운동들의 운동시간과 같은 하나의 속성만 가져오는 연산은 쿼리문 이외에도 추가 연산이 필요하게 되었다. 이에 루틴 테이블의 한 행이 각 루틴에 속한 운동종목 하나의 데이터를 담도록 변경하고 이로 인한 유일성 문제는 루틴 이름과 운동이름 2가지 조건을 통해 쿼리문을 사용하는 방식으로 해결했다. 비교적 간단히 해결이 가능했지만 DB는 모든 어플리케이션의 기능에 밀접한 연관이 있으므로 개발 일정의 지연이 불가피하다고 생각한다. 실제로 변경을 위해 1호의 토의와 구조 공유, 개발이 필요하였다. 이를 통해 데이터베이스 구조 설계의 중요성을 실감하게 되었다. 앞으로 적절한 데이터베이스 구조를 찾는 것에는 시간을 아끼지 말아야겠다는 교훈 하나만으로도 이번 프로젝트는 나에게 값진 프로젝트라고 생각한다.

또 하나 깨달은 점은 일정관리의 중요성이다. 최초에 우리 조는 기능을 크게 DB/루틴, 타이머/업적, 유튜브 연동/UI, 캘린더/결과창 이렇게 4개로 분류하여 인원별로 하나씩 맡기로 하였다. 각각의 파트가 어느 정도의 시간이 걸릴지 예측하기 어려워 기능별로 2주씩, 총 4주의 시간을 할당하였다. 개발이 진행되어 가면서 루틴 페이지가 프리셋 기능과 일일 루틴 2가지로 나누어 지고 업적 및 프로필, 개인정보 설정이 추가됨에 따라 개인별로 진행도의 차이가 생기게 되었고 이미 할당된 파트를 마친 인원이 있는 반면 아직 만들어야 될 기능이 2~3개 남아있는 인원도 생기게 되었다. 빠르게 파트를 마친 인원에게 남은 기능을 할당하였지만 그 시기가 늦었다고 생각한다. 1주에 한번 토의를 진행하면서 진행도를 공유하였지만 가용 인원이 적은 이러한 프로젝트에서는 상황에 대한 대체가 늦어질 수밖에 없었다고 생각하여 앞으로 비슷한 인원이 프로젝트를 진행하게 된다면 더욱 짧은 주기로 회의를 할 것이다.



그 외에도 팀원들이 개발한 각 기능을 1차로 결합하여 어플의 초본을 만들 때에는 미리 데이터를 통일하기 잘했다 생각 하면서도 각각 다른 빌드를 합치는 과정에서 최신 빌드를 공유하고 이를 바탕으로 한 개발이 필요하다고 느꼈다 개인적으로 1주일에 한번보다는 더 짧은 주기로 회의를 진행하고 이때까지 개발한 파트를 합쳐서 나온 빌드로 다음 회의까지 개발을 진행하는 것이 이번 프로젝트의 올바른 흐름이었다고 생각한다. 여러 요인으로 개발일정이 미루어지며 빌드가 잘 관리가 되지 못한점이 아쉬웠다고 생각한다.

이번 프로젝트는 나에게 많은 생각거리를 안겨주었다. 개발 툴을 선정하면서 나 스스로 얼마나 많은 언어와 프로그램에 경험이 있는지 돌아보게 되었고 데이터베이스 구조설계와 작업의 효율적 분배, 각 작업의 예상 소요기간 산정의 중요성과 적절한 회의 빈도의 필요성 또한 느끼게 되었다. 이 모든 내용이 소프트웨어 융합 전공과목에서 수강한 강의에 포함된 내용들이다. 이번 경험을 통해 단순히 지식으로만 아는 것이 아닌 필요성을 느끼게 되어 뜻 깊은 시간이었다고 생각한다.

## 2. 김보건

### 느낀점

기존에 했던 프로젝트는 거진 1~2개월 정도의 미니게임 제작 정도가 전부 였는데, 반학기 정도를 가지고 프로젝트를 진행하다보니 욕심이 앞서 여러 기능을 이것저것 추가하기도 했다. 이후 작업에 들어갈 때, 스케일이 생각보다 커져서 처음에는 만들 수 있나 걱정이 앞섰다.

해당 프로젝트를 진행하면서 가장 어려웠던 점은 기능구현보단 그 기능의 필요성에 대한 생각과 ux/ui구현이 가장 힘들었다. 매 주 조별분들과 회의하면서도 기능적인 부분이 아닌 어떻게 각 기능들을 배치하고 연결할지가 가장 우선이고 그에 대응하는 기능을 추가 또는 삭제와 같은 수정을 회의했던 것으로 기억한다. 각 기능을 구현하기 전에 앞서 여러 헬스어플을 사용했을 때, 이 기능은 불편했고 저 기능은 편했고 매 번 이생각이 뒤따랐다. 초기 계획서와 지금 결과 보고서를 비교해보니 매주 피드백을 하면서 조금씩 기능과 ui를 수정되어 별로 바뀐 것이 없을 줄 알았는데, 상당히 많은 것들이 바뀐 것을 보고 놀랐다 그래서 지금 보면 초창기에 생각했던 이미지와 꽤 다른 모습이 되었다.

### 알게된 내용

이번 프로젝트를 하면서 RecyclerView와 각 Layout안의 배치에 대해 엄청 많은 공부를 했다. 프로젝트를 시작하기전에는 어플들을 사용하면서 각종 ux/ui를 만드는 것이 꽤나 쉬울 줄 알았으나, 하나하나 직접 세부적으로 설정하면서, 특히 개발자의 의도대로 화면을 제작하기 위해 RecyclerView의 활용이 생각보다 크다는 것을 알게 되었다. Layout에 대해서도 각 Layout간의 xml파일 코드내의 배치에 따라서도 화면에 나타나는 것이 달라 코드 순서에 대해 꽤나 신경을 기울였다. LinearLayout, RelativeLaout등 여러 Layout이 있지만 가장 많이 사용했던 Layout은 요소들을 수평or수직으로 정렬하는데 사용하는 LinearLayout과 화면을 벗어난 요소들을 보기위해 스크롤 가능한 영역을 만드는 ScrollView이다.

또한 intent를 통해 stack의 중요성을 알게 되었다. 특히 여러 화면을 넘나들면서 정보전달하는 것에 대해 생각보다 stack에 대한 개념이 꽤 중요했다. 기존 intent로만 화면을 전환하면 정보를 전환할 때는 전환되는 화면이 새로운 화면이어서 뒤로가기를 누르면 바로 원했던 화면이 아닌 기존에 실행시켰던 화면들이 연달아 나온다거나, 정보전달을 할때도 한 화면으로 모든 정보를 전달할 때, 그 화면에 정보가 누적으로 저장되는 것이 아니어서 꽤나 애먹었다. 후에 Flag 설정을 통해 새 화면으로 전환하지 않고 기존 화면을 불러온다던가, 기존에 쌓여있던 stack의 배치를 바꾼다거나 하는 설정으로 정보의 누적 저장과 화면전환이 더욱 자연스러워지게 되었다.

결과적으로 프로젝트를 마무리하면서 피드백의 중요성에 대해 알게 되었다. 앞서 말한바와 같이 회의를 할 때마다, 어레이의 구조통일, 기능의 연결등 과 같은 여러 요소들을 가지고 회의를 했었다. 이러한 피드백이 자주 없었다면 지금의 결과가 기간에 맞춰 나올지도 의문

이며, 아예 의도와는 다른 이상한 결과로 갈 수 있었을 것 같다.

### 3. 박건희

#### 느낀점 & 배운점

팀 프로젝트로 진행한다는 소식을 들었을 땐 막연히 어려울 것 같다는 느낌을 받았다. 이 전에도 팀 프로젝트를 진행해보긴 했지만, 당시에는 주제가 정해져있어서 어떤 방식으로 만들지 정하는 것에 오래걸리지 않았고, 팀 인원도 제한이 없어서 이미 알고있던 사람들끼리 팀을 이뤄 제작하였기에 어색할 일도 없었다. 하지만 이번에는 처음으로 손발을 맞춰보는 팀원분들이 존재했고, 주제도 완전히 자율적이었기에 ‘다른 팀원들보다 개발 실력이 떨어져 진도를 따라가지 못하면 어쩌지?’라는 걱정을 많이 했다. 그러나 걱정과 동시에 과연 팀을 이뤄 프로젝트를 진행하면 어떤 것까지 만들어 낼 수 있을지와 이 프로젝트를 진행하면서 어떤 점들을 배울 수 있을지에 대한 기대감도 함께 부풀어 올랐던 것 같다.

처음 팀 미팅을 진행하였을 때는 어색했다. 하지만 팀 프로젝트를 진행하면서 본인이 할 수 있는 일에 최선을 다하자라는 생각으로 시간을 보내다보니 점차 어색함이 누그러진 것 같다. 팀 프로젝트를 진행하며 도움이 필요한 부분에 대해서는 적극적으로 도움을 요청하여 많은 의견을 나눴고, 수정이 필요한 부분은 서로 피드백을 나누며 수정해 나갔다. 여러 번의 팀 미팅을 거치면서 본인이 맡은 부분에 대해서 욕심이 생겼고, 부족한 부분을 더 높은 퀄리티로 채워나가보자라는 생각으로 어떻게 만들어야 더 편리하게 사용할 수 있을지, 어떤 식으로 만들어야 더 직관적으로 보일지에 대해 고민하며 많은 시간을 보냈던 것 같다.

처음에는 어떤 것을 만들지 고민하였다. 고민 끝에 나온 기획은 실시간 번역기와 일정 플래너 어플이었다. 결국 일정 플래너와 보건님이 기획한 운동 어플을 결합한 운동 플래너 어플을 만들기로 정하였고, 주제가 정해지자마자 어플 제작에 필요한 기능이 무엇일지 고민을 하였다. 동시에 나는 사람들에게 설문지를 진행하였다. 처음엔 설문지를 해봤자 도움이 될지에 대한 의구심을 가지고 있었다. 예상과는 다르게 결과적으로 굉장히 유의미하였고, 생각보다 많은 인원이 참가해주어 너무 감사한 마음이 들었다. 결국 설문으로 인해 현재의 어플에 들어간 업적 기능을 구현할 생각을 할 수 있었다. 이를 계기로 확실히 여러 사람들의 이야기를 들어보는 것이 많은 도움이 된다는 것을 배웠고, 덕분에 다른 운동 어플들과는 확실하게 차별화될 수 있는 업적 기능을 구현하는 것에 성공했다. 그리고 프로젝트를 진행하며 어떤 기능을 어떤 식으로 배치하고, 표현할 지에 대해 고민을 하는 시간이 많았다. 확실히 처음 구현했던 페이지와 비교하였을 때 시각적으로나 기능적으로 훨씬 개선된 점을 볼 수 있다. 이 밖에 팀과 협업하며 어떤 방식으로 진행하는 것이 효율적인 개발이 될지, 어떤 점이 부족한지 등을 경험할 수 있었던 시간이었다고 생각한다.

팀 프로젝트를 진행함에 있어 어려운 점 없이 순조롭게만 진행된 것은 아니었다. 여러 번 미팅을 진행하여 많은 의견을 나누다보니 방대한 내용으로 인해 피드백한 기능이 제대로 구현되지 않고 누락되는 경우도 있었고, 개인마다 일정 차이로 인해 프로젝트가 딜레이되는

경우도 종종 발생했다. 그리고 의견을 나눈 내용과 다르게 페이지가 구현되어 결합하는 데에 고생하는 경우도 있었다. 무엇보다 가장 아쉬웠던 점은 시간이 부족했다는 점이다. 시간이 흐를수록 더 좋은 어플을 만들고자하는 욕심이 들었지만 생각보다 주어진 시간이 많이 없었기에 급하게 기능만 구현해둔 채 마무리되어 아쉽다는 생각이 들었다. 하지만 짧은 시간임에도 큰 트러블 없이 함께 프로젝트를 잘 진행해준 팀원들에게 고마움을 느낀다.

개인적으로 매우 재미있게 진행되었고 만족스러웠다. 평소 학기 중에는 어플 제작에만 온전히 몰두할 수 있는 여유가 없었는데, 해당 과목을 수강하면서 어색한 사람과 손발을 맞추고, 함께 제작하면서 느끼고 배울 수 있는 점이 많았고, 제작에 몰두하면서 재미를 느낄 수 있었다. 마지막으로 다른 사람들이 말기 어려워하는 DB를 전달해서 말아준 주원이나 캘린더와 일일루틴 페이지를 담당하고 필요한 기능에 대해서 센스있게 잘 챙겨 구현해주셨던 보건님, 그리고 일정이 좀 딜레이되긴 했지만 그래도 마지막에 피드백한 내용을 잘 따라와 메인 화면과 결과 페이지, 유튜브 페이지를 잘 마무리해주신 동현님까지 다들 너무 고생하셨다는 말을 전하고 싶다.

## 4. 이동현

### 이번 미니 프로젝트를 진행하면서 느낀점

이번 미니 프로젝트를 진행하며 제 자신의 한계를 실감했습니다. 동료들에 비해 부여받은 업무량이 적었음에도 불구하고, 기능 구현에 많은 시간을 소요하였고, 때론 프로젝트 일정을 지키지 못해 팀원들에게 민폐를 끼치기도 하였습니다. 이러한 과정에서 저의 부족함을 느끼는 한편, 같이 고생하며 저를 지지해준 팀원들에게 깊은 감사의 마음을 가지게 되었습니다.

작년에 게임 프로젝트를 진행하며, 그리고 이번 안드로이드 앱 개발에서도 느꼈지만, 큰 규모의 프로젝트를 수 년 동안 진행하는 개발자들의 노력과 고생이 얼마나 클지를 상상해봅니다. 이를 통해 저 자신이 얼마나 더 성장하고 노력해야 업계에서 뒤처지지 않을지를 깊이 생각하게 되었습니다.

또한 이번 프로젝트에서 AI를 활용하였는데, 앞으로 AI의 활용이 기술적으로 중요한 요소가 될 것이라는 생각을 하게 되었습니다.

### 이번 미니 프로젝트를 진행하면서 알게된 내용

이번 프로젝트를 통해 알게 된 기술적 측면 중 하나는 데이터베이스 활용입니다.

단순히 데이터를 입력하고 사용한다는 개념에서 벗어나, SQL을 활용하여 데이터베이스를 효율적으로 관리하고, 필요한 데이터를 쉽게 조회하고, 갱신하거나 삭제하는 등 다양한 기능을 구현할 수 있음을 알게 되었습니다.

그리고 개념적 측면에서는 '유지보수성'에 대해 더 깊게 이해하게 되었습니다.

소프트웨어 비용의 대부분이 유지보수에 소요된다는 사실을 알게 되었고, 이를 통해 유지보수의 중요성을 체감하였습니다. 특히, 이번 프로젝트에서는 팀원들과 협업하며 코드에 주석을 달아 작업 흐름을 명확히 하고, UI/UX 보수와 기능 추가 등을 통해 효과적인 유지보수를 경험하였습니다. 이런 경험은 앞으로의 프로젝트에서도 큰 도움이 될 것입니다.