

**제1회 경남소프트웨어경진대회**

**[ WOR - 운동일지 ]개발 완료 보고서**

**소속 : [ A-TEAM ]**

**버전번호 : [Version 1.0]**

제출일자 : [2020-09-23]

**목 차**

목차

[1. 개발 개요 3](#_Toc85202621)

[1.1 개발 명 3](#_Toc85202622)

[1.2 개발 기간 3](#_Toc85202623)

[1.3 개발 목적 3](#_Toc85202624)

[1.4 개발 순서 3](#_Toc85202625)

[1.5 개발 기대효과 4](#_Toc85202626)

[2. 개발환경 4](#_Toc85202627)

[2.1 개발환경 4](#_Toc85202628)

[2.2 운영환경 4](#_Toc85202629)

[3. 개발 어플리케이션 설명 5](#_Toc85202630)

[3.1 어플리케이션의 기능 및 구성 소개 5](#_Toc85202631)

[3.2 사용한 데이터베이스 5](#_Toc85202632)

[3.3 액티비티 및 프레그먼트 구성 6](#_Toc85202633)

[3.4 사용자 입력 처리 6](#_Toc85202634)

[3.5 디자인 및 테마 7](#_Toc85202635)

[4. 어플리케이션 클래스 설명 7](#_Toc85202636)

[4.1 Model 및 데이터베이스 관련 클래스 7](#_Toc85202637)

[4.2 View, Control 관련 클래스 8](#_Toc85202638)

[5. 참고 자료 9](#_Toc85202639)

# 1. 개발 개요

### 1.1 개발 명

제1회 경남소프트웨어경진대회 [ WOR - 운동일지 ] (이하 “프로그램 개발”

이라 칭한다.)

### 1.2 개발 기간

2021년 09월 16일 ~ 2021년 10월 15일

### 1.3 개발 목적

1. 안드로이드 스튜디오 사용법 숙달 및 데이터베이스를 활용한 애플리케이션 개발.
2. 근육은 점진적 과부화를 통하여 성장함. 따라서 체계적으로 이전에 어떤 운동을 어떤 강도로 진행했는지 숙지하고 있어야 다음 단계로의 성장이 가능. 이 점에 착안하여 근육 성장을 돕기 위해 운동 일지를 기록하는 어플리케이션을 개발하고자 함.
3. 운동 후에 일지를 작성하여 체계적으로 운동을 할 수 있는 애플리케이션 개발. 본인의 운동 루틴을 저장하고, 불러올 수 있도록 하고, 캘린더를 활용하여 언제 운동을 했는지 알 수 있게 하는 기능을 포함.

### 1.4 개발 순서

1. 개발 명:[ WOR - 운동일지 ]  
   - 운동 일지를 작성할 수 있는 앱 구현
2. 애플리케이션의 레이아웃 구성 설정 및 데이터베이스 구현  
   - 액티비티와 프레그먼트의 전반적 레이아웃 요소 구성

- 데이터베이스 구현

1. 사용자의 입력 이벤트 처리   
   - 프레그먼트 간 화면전환 이벤트 처리  
   - 데이터베이스의 데이터 호출 등 이벤트 처리  
   - 일지 작성 및 삭제 시에 데이터베이스에 저장 및 삭제 구현
2. 디자인

- 전체적인 테마 색상 및 글자 색 정하기

- 앱 아이콘 디자인  
- 다크 모드 색상 구현

### 1.5 개발 기대효과

1. 본인의 운동을 체계적으로 기록하여, 시민들의 건강 증진에 도움이 될 수 있도록 함.
2. 캘린더와 연계되는 메모 기능으로, 운동만 아니라 학생들의 당일 공부 내용 등을 기록하는 앱으로 사용할 가능성도 있음.

# 2. 개발환경

### 2.1 개발환경

|  |  |  |
| --- | --- | --- |
| **구분** | | **Spec.** |
| 시스템 환경  (자체 개발 서버) | OS | Window10 64bit |
| CPU | Intel(R) Core(TM) i5-11400F |
| Memory | 32GB |
| Disk | 500GB |
| 개발방법 | Android Studio |
| Data Base | SQLite |
| 디자인 환경 |  | Android Studio |

### 2.2 운영환경

|  |  |  |
| --- | --- | --- |
| **구분** | | **Spec.** |
| 시스템 환경 | OS | API 22 : Android 5.1(Lollipop) 이상 |
| CPU | 1.5GHz 싱글코어 |
| Memory | 1GB |
| HDD | 8GB |
| 개발방법 | Android Studio, kotlin |
| Data Base | SQLite |

# 3. 개발 어플리케이션 설명

### 3.1 어플리케이션의 기능 및 구성 소개

* 제작한 어플리케이션은 크게 운동 일지를 기록, 수정하는 기능, 삭제하는 기능, 작성한 루틴을 저장, 불러오는 기능, 작성한 일지를 검색하는 기능, 캘린더에서 일지를 확인하는 기능이 있음. 기능을 사용하는데 대한 설명은 별도 첨부한 소프트웨어 사용자 매뉴얼[[1]](#endnote-1)에 작성되어있음
* 크게 보면 운동 일지를 기록하는 기능에 관한 뷰, 운동 일지 목록을 볼 수 있는 뷰, 캘린더에서 확인하는 뷰 이렇게 3가지로 이루어져 있고, 뷰는 프레그먼트를 통하여 구성하였음.
* 이외의 일지 제거, 일지 추가, 루틴 저장 및 불러오기 기능은 app bar의 메뉴 기능, alert dialog 기능을 통하여 구현하였음.
* 일지 목록 화면은 recycler view를 사용하여 일지를 추가하여 목록의 수가 많아졌을 때 스크롤을 이용하여 많은 목록을 확인할 수 있도록 구현하였음
* 완성된 어플리케이션에 대한 시운전 가능한 결과물은 apk 파일[[2]](#endnote-2)로 별도 첨부하였음

### 3.2 사용한 데이터베이스

* SQLite 사용 (안드로이드의 Room database 사용)

@Database(entities = [Record::class], version = 1)

abstract class RecordDatabase : RoomDatabase() {

}

* 위 코드 예시처럼 데이터베이스 선언, 데이터베이스의 테이블은 Record 클래스를 참조하고, 해당 데이터베이스의 테이블은 첨부파일의 데이터베이스 테이블 명세서[[3]](#endnote-3)에 기재되어 있음.
* 해당 데이터베이스로의 접근은 Database Access Object(이하 dao)를 통해 이루어지며, 이 dao에는 데이터를 추가, 수정, 삭제 그리고 검색할 수 있는 쿼리문이 메소드로 구현되어 있음.
* SQLite와 kotlin 클래스에서의 데이터 형식이 다르게 사용되는 데이터(ex UUID, Date)를 처리하기 위해 @TypeConverter을 사용했음.

### 3.3 액티비티 및 프레그먼트 구성

* 메인 액티비티는 activity\_workout\_record이고, 사용한 레이아웃은 FrameLayout, 이곳을 fragment의 컨테이너로 사용했음
* 사용한 프레그먼트는 fragment\_workout\_list, fragment\_workout\_detail, fragment\_workout\_calendar 세 가지의 fragment를 사용하였고, 각각 일지의 목록을 보여주는 화면, 일지를 작성하고 세부 내용을 볼 수 있는 화면, 달력에서 일지를 볼 수 있는 화면을 구성하고 있음.
* 각 프레그먼트는 WokroutRecordActivity.class에서 supportFragmentManager의 replace 메소드를 통해 전환하고 있음. list 프레그먼트에서 detail 프레그먼트로 전환할 때 데이터의 UUID를 가지고 화면을 전환하며, 이를 통해 데이터에 접근하여 데이터를 읽어오고 수정, 추가, 저장 등을 진행함.

### 3.4 사용자 입력 처리

* 사용자 입력을 처리하기위해 메뉴를 활용했음. fragment\_delete\_list, fragment\_workout\_detail, fragment\_workout\_list 총 세가지의 메뉴 xml 파일을 만들었고, 각각 일지 목록 화면에서 삭제를 진행할 때, 일지를 작성하는 화면일 때, 일지 목록 화면일 때 표기되는 메뉴임. 대부분 OnClickListner의 OnClick 메소드를 활용함.
* list 프레그먼트에서 detail프레그먼트로 이동하는 부분은 recyclerView의 adapter에 onClickListener을 추가하여 프레그먼트간의 이동을 구현하였고, adapter에 추가로 onLongClickListener을 구성하여 꾹 눌러서 삭제 기능을 구현.
* detail 프레그먼트에서 작성한 메모를 저장하는 방법으로는 해당 프레그먼트의 onStop 생명주기에 record를 기록하는 메소드를 사용하여 작성 화면에서 뒤로가기 버튼을 누르거나 홈버튼을 누르는 등의 활동을 통해 메모 내용을 저장할 수 있게 구현.
* 캘린더 뷰에서는 onDateChangeListener를 구성하여 유저가 날짜를 선택했을 때의 이벤트를 처리하였음.

### 3.5 디자인 및 테마

* 전체적인 테마색은 연두색.
* 안드로이드 스마트폰에서 기본 제공하는 기능인 다크모드에서도 앱이 정상적으로 작동하기 위하여 텍스트 색상은 회색으로 구성함.
* alertDialog 위젯의 모서리의 각을 둥글게 만들기 위해 radius=”30dp” 속성을 가진 alert\_dialog\_backgroud.xml 파일을 사용하였음.
* recyclerView의 어댑터 배경을 위하여 color, radius 속성을 가진 edittext\_background.xml 파일을 사용하였음.

# 4. 어플리케이션 클래스 설명

### 4.1 Model 및 데이터베이스 관련 클래스

* Record 클래스는 데이터 클래스로, 하나의 자료형처럼 사용하는 클래스, 여기는 운동 부위, 운동 루틴, 운동 반복횟수, 시간, UUID의 자료형을 포함하고 있음.
* RecordDatabase 클래스는 abstract 클래스이며, RecordDAO 인터페이스를 포함해야함. Record 클래스를 데이터베이스의 테이블로 사용함.
* RecordDAO 인터페이스는 데이터베이스와 통신하는 쿼리문들을 포함함. 데이터베이스와의 통신을 해당 클래스를 통해서 하게 됨.
* RecordTypeConverters 클래스는 SQLite와 kotlin 간의 자료형 변환을 담당하는 클래스로, RecordDatabase에서 선언되어 사용함.
* RecordRepository 클래스는 RecordDatabase 클래스를 constructor를 통해 데이터베이스로 선언하고, recordDAO를 선언, 이 둘을 사용하여 데이터베이스를 만들고 데이터베이스에 접근하는 역할을 모두 해당 클래스를 통하여 진행하게 됨. 따라서 companion object(동반 객체)를 사용하여 어플리케이션 사용 중에는 단 하나의 RecordRepository 인스턴스가 생성되어 사용되도록 하였음.
* WorkoutRecordApplication 클래스는 개발한 어플리케이션의 서브클래스로, 여기서 RecordRepository 인스턴스를 생성하고, 후술할 ViewModel들 또한 여기서 인스턴스를 생성함.
* RecordsViewModel 클래스는 일지 목록 화면에서 주로 사용하는 ViewModel로, 매번 데이터를 가져올 때 Database와 직접 연결해서 통신하지 않고 ViewModel을 통해 데이터를 임시로 저장하여 사용하기 위한 클래스, 주로 List<Record>를 다루는 뷰모델
* RecordAndRoutineViewModel 클래스는 일지 추가 화면에서 주로 사용하는 ViewModel로, Record 객체를 하나씩 다룰 때 사용하는 ViewModel이고 여기에는 공유 프리퍼런스를 사용해 루틴을 저장하고 불러오는 메소드도 포함되어 있음.
* QueryPreferences object는 공유 프리퍼런스 사용을 위한 메소드들로 구성되어 있음. 키와 데이터를 쌍으로 저장하고 불러오는 메소드를 포함함.

### 4.2 View, Control 관련 클래스

* 4.2에서 후술할 액티비티, 프레그먼트 클래스들에 사용자와의 상호작용 처리 및 화면전환 등의 내용들이 작성되어있음. View에 해당하는 xml파일에 대한 내용은 서술하지 않을 예정. 첨부한 소스코드[[4]](#endnote-4)에서 확인할 수 있음.
* WorkoutRecordActivity 클래스는 프레그먼트들의 컨테이너 역할을 하는 액티비티 클래스, 여기서 프레그먼트들 간의 전환을 담당하고 있음.
* WorkoutListFragment 클래스는 운동 일지 목록을 보여주는 프레그먼트.
* WorkoutListDeleteFragment 클래스는 운동 일지 목록 화면에서 리사이클러뷰의 어댑터를 LongClick 하거나 상단의 삭제 버튼을 눌렀을 때 진입하는 화면으로, WorkoutListFragment와 같은 xml 파일로 디자인되어 있지만, 메뉴가 달라지고 리사이클러뷰의 어뎁터가 달라짐. 삭제 기능을 담당하는 프레그먼트.
* WorkoutDetailFragment는 운동일지를 작성, 수정하거나 자세히 보는 화면을 제공해주는 프래그먼트.
* WorkoutCalendarFragment는 캘린더뷰를 사용하여 달력에서 일지 작성 현황을 볼 수 있게 해 주는 프레그먼트.

# 5. 참고 자료

* Do it! 코틀린 프로그래밍 (황영덕 지음, 이지스퍼블리싱)
* 실무에 바로 적용하는 안드로이드 프로그래밍 4판 (심재철 옮김, 제이펍)
* <https://stackoverflow.com/>

별첨 목록

1. 소프트웨어 사용자 메뉴얼 [↑](#endnote-ref-1)
2. 실행가능한 형태의 결과물(apk 파일) [↑](#endnote-ref-2)
3. 데이터베이스 테이블 명세서 [↑](#endnote-ref-3)
4. 소스파일 [↑](#endnote-ref-4)