LAPORAN TUGAS BESAR
PEMROGRAMAN MOBILE
APLIKASI RUNNING

OLEH :

IRA RIYANA SARI SIREGAR
1809075014

PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN

SAMARINDA
2021

DAFTAR ISI

halaman

DAFTAR GAMBAR

halaman

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dari masa ke masa mengalami perkembangan yang sangat signifikan. Seiring dengan kemajuan ilmu pengetahuan maka, akan memberikan dampak terhadap perkembangan teknologi. Perkembangan teknologi memberikan kemudahan kepada manusia dalam menjalankan aktivitas sehari – hari. Penggunaan teknologi yang banyak digunakan manusia adalah *smartphone*. Menurut kementrian komunikasi dan informatika menyatakan bahwa, jumlah pengguna *smartphone* di Indonesia mencapai 167 juta orang atau setara dengan 89% jumlah penduduk di Indonesia dan jumlah ini meningkat setelah pandemi Covid-19 pada akhir tahun 2020. Android menjadi versi yang paling banyak digunakan di Indonesia. Pada laporan perusahaan analitik *setCounter* mencatat Android 10 terinstall sebanyak 34,37% ponsel android di Indonesia. Berdasarkan dari perkembangan teknologi yang semakin cepat maka, penting pelajar atau mahasiswa dalam memahami pembelajaran mengenai pemrograman *mobile*. Pemrograman *mobile* adalah pembuatan aplikasi yang berjalan pada perangkat bergerak seperti *handphone* atau *tablet*. Dalam pembuatan aplikasi menggunakan android studio terdapat beberapa bahasa yang dapat digunakan antara lain java dan kotlin. Kotlin merupakan bahasa pemrograman *modern* yang bersifat *statically – typed* yang dapat dijalankan pada *platform Java Virtual Machine* (JVM). Bahasa pemrograman kotlin juga dapat di *compile* ke dalam bentuk *JavaScript.* Pada laporan tugas besar ini, penulis merancang aplikasi *running*. Pada masa pandemi Covid-19 pemerintah mengajak masyarakat untuk hidup lebih sehat dan rajin untuk berolahraga. Kegiatan olahraga jogging atau marathon merupakan salah satu olahraga yang digemari oleh banyak orang. Seiring meningkatnya kesadaran masyarakat akan pentingnya hidup sehat, olahraga lari menjadi gaya hidup sebagian besar masyarakat. Olahraga ini terkenal dengan olahraga yang paling mudah, sederhana dan ekonomis. Maka, tidak heran jika olahraga ini menjadi pilihan yang tepat. Oleh karena itu, dengan adanya aplikasi *running* yang di rancang dan dibuat oleh penulis diharapkan dapat membantu masyarakat dalam melaksanakan aktivitas berolahraga.

1.2 Rumusan Masalah

Adapun rumusan masalah yang dapat diperoleh berdasarkan latar belakang yang telah dipaparkan oleh penulis :

1. Bagaimana langkah – langkah dalam merancang sebuah aplikasi *running* ?
2. Bagaimana langkah – langkah dalam menghadapi *troubleshoot* pada penyelesaian aplikasi *running* ?

1.3 Tujuan Pembuatan Aplikasi Android

Adapun tujuan dalam pembuatan aplikasi *running* yang diharapkan oleh penulis :

1. Untuk mengetahui langkah – langkah dalam merancang sebuah aplikasi *running.*
2. Untuk mengetahui langkah – langkah dalam menghadapi *troubleshoot* pada penyelesaian aplikasi.

1.4 Manfaat

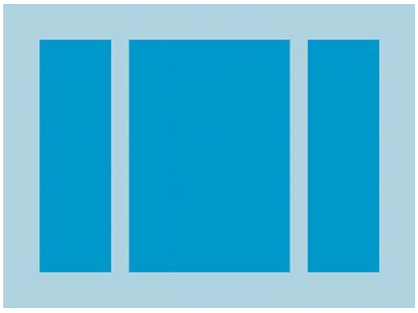
Adapun manfaat dari aplikasi Running yang diharapkan oleh penulis sebagai berikut :

1. Dapat mempermudah pengguna dalam melakukan loop training.
2. Dapat mempermudah pengguna dalam membuat list aktivitas yang akan dilakukan dengan durasi waktu yang telah ditentukan oleh pengguna.
3. Dapat mempermudah pengguna dalam memberikan notifikasi peringatan mengenai sisa waktu aktivitas yang sedang dilakukan.

BAB II
LANDASAN TEORI

2.1 Linear Layout

Linear layout adalah kelompok tampilan yang menyejajarkan semua turunan dalam satu arah, baik vertikal maupun horizontal. Semua turunan linear layout akan ditumpuk satu sama lain, sehingga daftar vertikal hanya akan memiliki satu turunan per baris, berapa pun lebarnya dan daftar horizontal hanya akan setinggi satu baris (tinggi turunan yang tertinggi, ditambah dengan pengisi). Linear layout mematuhi *margin* antara turunan dan gravitasi (sejajar kanan, tengah atau kiri) setiap turunan.



Gambar 1.1 Tampilan Linear Layout

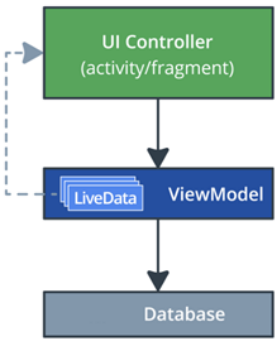
Linear layout juga mendukung penetapan bobot ke masing – masing turunan dengan atribut android:layout_weight. Atribut ini menetapkan nilai ke tampilan terkait seberapa banyak ruang yang akan ditempatinya pada layar. Nilai bobot yang lebih besar memungkinkannya diperluar untuk mengisi ruang yang tersisa di tampilan induk. Tampilan turunan bisa menetapkan nilai bobot, kemudian ruang yang tersisa dalam kelompok tampilan diberikan ke turunan dalam proporsi yang dideklarasikan untuk mereka. Bobot *default* adalah nol.

Untuk membuat tata letak *linear* tempat setiap turunan menggunakan jumlah ruang yang sama di layar, setel android:layout_height setiap tampilan dengan “0dp” (untuk tata letak vertikal atau horizontal). Kemudian setel android:layout_weight setiap tampilan ke “1”. Pada pembuatan tata letak *linear* tempat elemen turunan

menggunakan jumlah ruang yang berbeda di layar. Pertama jika terdapat tiga kolom teks dan dua diantaranya mendeklarasikan bobot 1, sementara yang lain tidak diberikan bobot, kolom teks ketiga yang tanpa bobot tidak akan berkembang. Sebagai gantinya, kolom teks ketiga ni hanya akan menempati area yang diperlukan oleh kontennya. Di sisi lain, dua kolom lainnya akan diperluas secara seimbang untuk mengisi ruang yang tersisa setelah ketiga kolom diukur. Kedua, apabila terdapat tiga kolom teks dan dua diantaranya mendeklarasikan bobot 1, sementara kolom ketika diberikan bobot 2 (sebagai ganti 0), maka kolom ketga akan dideklarasikan lebih penting daripada kedua kolom lainnya. Sehingga kolom tersebut akan mendapatkan separuh jumlah ruang yang tersisa, sedangkan dua yang pertama akan berbagi sisanya secara seimbang.

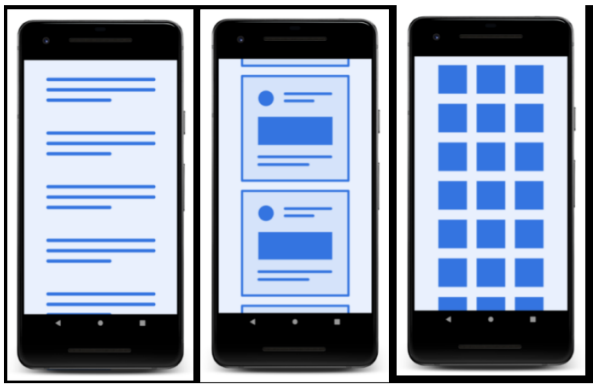
2.2 Recycler View

Recycler View adalah tampilan yang menggunakan arsitektur yang disederhanakan dengan UI *controller*, *ViewModel* dan *LiveData*.



Gambar 1.2 Tampilan arsitektur *recycler view*

Menampilkan list atau *grid* data adalah salah satu tugas UI paling umum di Android. Daftar bervariasi dari yang sederhana hingga yang sangat kompleks. Daftar tampilan teks mungkin menampilkan data sederhana contohnya adalah daftar belanja sedangkan, untuk daftar yang kompleks contohnya adalah daftar tujuan liburan yang beranotasi dapat menunjukkan kepada pengguna banyak detail di dalam *scrolling grid* dengan *header*. Untuk mendukung semua kasus android menyediakan *widget RecyclerView*.

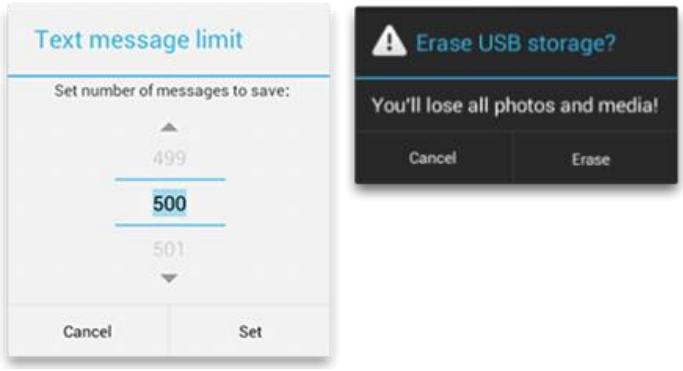


Gambar 1.3 Tampilan *recycler view*

Manfaat terbesar dari *recycler view* adalah sangat efisien untuk daftar besar secara *default*. *Recycler view* hanya berfungsi untuk memproses atau menggambar item yang saat ini terlihat di layar. Misalnya, jika list memiliki seribu elemen tetapi, hanya 10 elemen yang akan terlihat. Ketika pengguna melakukan *scroll*, *recycler view* akan mengetahui *item* baru apa yang seharusnya ada di layar dan tidak akan berfungsi untuk menampilkan *item* tersebut. Ketika suatu *item scroll* dari layar, tampilan *item* tersebut di daur ulang yang berarti *item* akan diisi dengan konten baru yang *scroll* ke layar. Perilaku pada *recycler view* akan menghemat banyak waktu pemrosesan dan membantu *scroll list* dengan lancar.

2.3 Dialog

Dialog adalah jendela kecil yang meminta pengguna untuk membuat keputusan atau memasukkan informasi tambahan. Dialog tidak memenuh layar dan biasanya digunakan untuk peristiwa modal yang mengharuskan pengguna untuk melakukan tindakan sebelum bisa melanjutkan.



Gambar 1.4 Tampilan dialog

Menggunakan dialog *fragment* untuk mengelola dialog akan memastikannya menanngani peristiwa siklus proses dengan benar seperti ketika, pengguna menekan tombol kembali atau memutar layar. *Class* dialog *fragment* juga memungkinkan pengguna untuk menggunakan kembali UI dialog sebagai komponen yang bisa di sematkan dalam UI yang lebih besar.

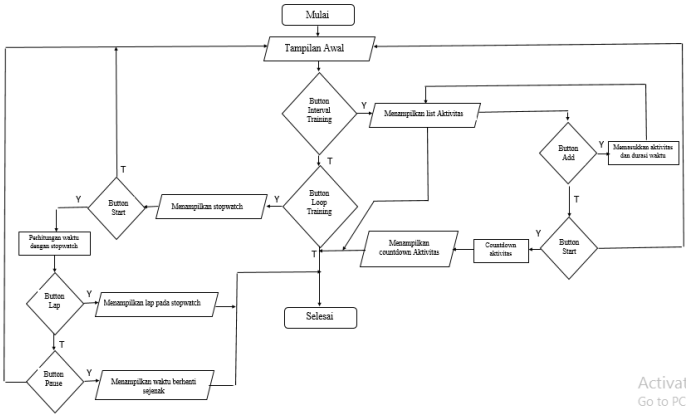
2.4 SQL Database

Database *SQLite* adalah bahasa yang digunakan untuk menjalankan perintah manipula atau mengakses data pada database. *SQLite* sebagai media penyimpanan utama untuk data aplikasi atau pengguna atau dapat menggunakan proses *caching* serta menyediakan data yang diambil dari *cloud*. Jika menggunakan database *SQLite* yang dinyatakan sebagai objek *SQLiteDatabase* adalah semua interaksi dengan database adalah melalui instance dari kelas *SQLiteOpenHelper* yang akan mengeksekusi permintaan dan pengelolaan database. Aplikasi hanya boleh berinteraksi dengan *SQLiteOpenHelpe*r. Singkatnya *SQLite database* memiliki metode untuk membuat, menghapus, menjalankan perintah SQL, dan melakukan tugas manajemen database umum lainnya. seperti perintah CRUD (*Create, Read, Update, Delete*) data.

BAB III
PERANCANGAN APLIKASI

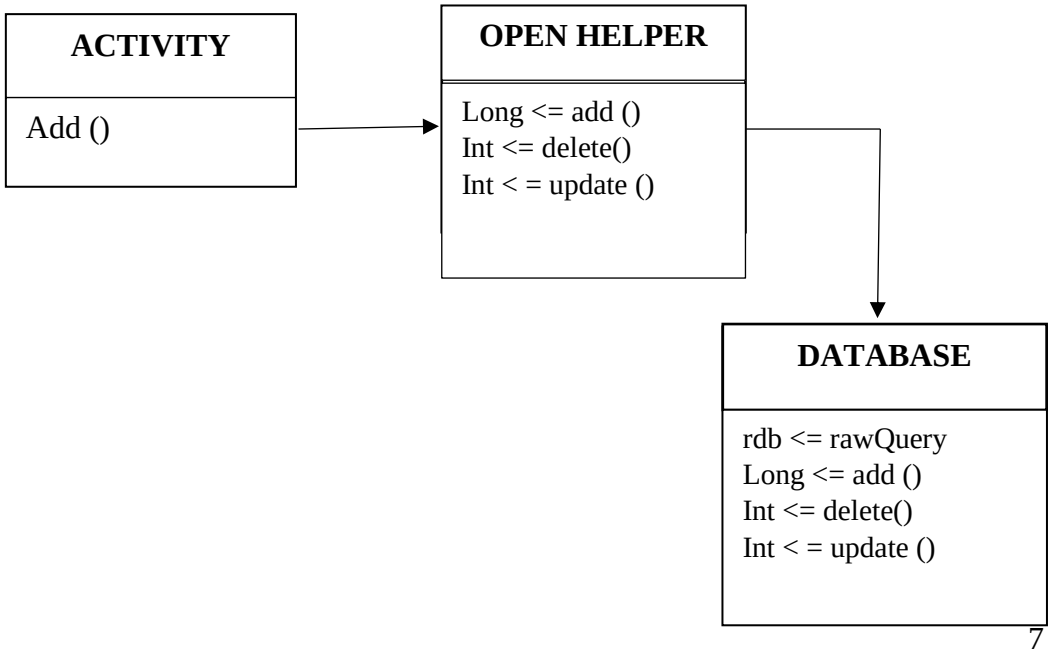
3.1 Perancangan Sistem

Pada perancangan sistem aplikasi menggunakan spesifikasi minimum *Software Development Kit* (SDK) yaitu android 4.4 Kitkat (API level 19). Pada pengambaran urutan proses intruksi pada program *running* dirancang *flowchat* aplikasi *running* sebagai berikut :

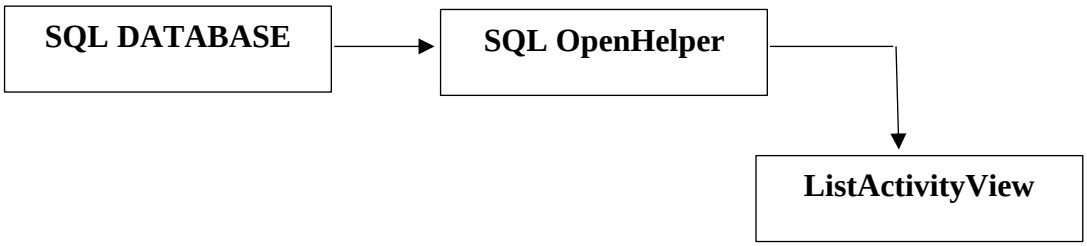


Gambar 1.5 *Flowchart* aplikasi

Diagram untuk memetakan struktur sistem aplikasi sebgai berikut :



Entity relatoinship diagram pada aplikasi sebagai berikut :



3.2 Perancangan UI / UX

Pada perancangan *user design interface* aplikasi *running* terdapat blok pembangun yang penting. Komponen adalah suatu titik masuk sistem atau pengguna ketika memasuki aplikasi. Komponen yang digunakan dalam perancangan aplikasi ini adalah Text View, Card View, Image View, Image Button dan Edit Text. Tata letak pada aplikasi *running* menggunakan *linear layout* dan *relative layout* untuk tampilannya. Style yang digunakan dalam perancangan aplikasi adalah Widget.MaterialComponents.TextInputLayout.OutlinedBox pada layout interval_training_edit_session_dialog.xml dan interval_training_save_dialog.xml. Pada main_menu.xml menggunakan style Theme.MaterialComponents.DayNight berarti bahwa tema yang digunakan pada aplikasi adalah mode dark. Pattern yang digunakan pada aplikasi *running* adalah *rectagle* dan *ring*. *Usability* pada aplikasi *running* adalah memberikan *effectiveness* kepada pengguna dalam melakukan aktivitas berolahraga, *utility* yaitu memberikan fungsi yang baik sehingga pengguna dapat melakukan yang dibutuhkan ketika menggunakan aplikasi dan *learnability* yaitu pengguna dapat mudah mempelajari aplikasi sebelum digunakan.

3.3 Coding

Berikut adalah coding untuk aplikasi *running:*

1. AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/andr
oid"
    package="com.nicknterm.runningapp">

    <application
        android:allowBackup="true"
```

```
            android:icon="@drawable/icon"
            android:label="@string/app_name"
            android:roundIcon="@drawable/icon"
            android:supportsRtl="true"
            android:theme="@style/Theme.RunningApp">
            <activity android:name=".Stopwatch"
                android:screenOrientation="portrait"
                android:theme="@style/MyAppTheme"/>
            <activity
                android:name=".MainMenu"
                android:screenOrientation="portrait"
                android:theme="@style/MyAppTheme">
                <intent-filter>
                    <action
android:name="android.intent.action.MAIN" />

                    <category
android:name="android.intent.category.LAUNCHER" />
                </intent-filter>
            </activity>
            <activity
                android:name=".FinishActivity"
                android:screenOrientation="portrait"
                android:theme="@style/MyAppTheme" />
            <activity

android:name=".IntervalTrainingExerciseActivity"
                android:screenOrientation="portrait"
                android:theme="@style/MyAppTheme" />
            <activity

android:name=".IntervalTrainingMainActivity"
                android:screenOrientation="portrait"
                android:theme="@style/MyAppTheme" />
        </application>

    </manifest>
```

2. DBHandler.kt

```
package com.nicknterm.runningapp

import android.annotation.SuppressLint
import android.database.Cursor
import android.database.SQLException
import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
```

```
class DBHandler(context: Context):
SQLiteOpenHelper(context,DATABASE_NAME,null,DATABASE_V
ERSION) {
    companion object{
        // Change DATABASE_VERSION every time you
change the structure of the database
        // For example if you add another column,
change the DATABASE_VERSION
        // Plus the onUpgrade function is going to get
called so change that so the user doesn't loses all
its data
        private const val DATABASE_VERSION = 1
        private const val DATABASE_NAME = "TrainTimer"
        private const val TABLE_NAME = "TrainTable"

        // Table columns
        private const val KEY_ID = "_id"
        private const val KEY_TRAINING_NAME =
"train_name"
        private const val KEY_ITEM_ID = "item_id"
        private const val KEY_DESCRIPTION =
"description"
        private const val KEY_TIMER = "times"

    }

    // If the Database is not made then it Creates the
main table
    override fun onCreate(db: SQLiteDatabase?) {
        val sql = ("CREATE TABLE $TABLE_NAME ($KEY_ID
INTEGER PRIMARY KEY, $KEY_TRAINING_NAME TEXT,
$KEY_ITEM_ID INTEGER, $KEY_DESCRIPTION INTEGER,
$KEY_TIMER TEXT)")
        db?.execSQL(sql)
    }

    // Is called only while the version has changed.
Change that
    // function accordingly so the user doesn't loses
its data
    override fun onUpgrade(db: SQLiteDatabase?,
oldVersion: Int, newVersion: Int) {
        db!!.execSQL("DROP TABLE IF EXISTS
$TABLE_NAME")
        onCreate(db)
    }

    // This function saves a train record into the
database
```

```
    // It returns a Long. I don't even know what this
is sooo
    fun saveIntervalTrainingItem(item:
IntervalTrainingItem, name: String):Long{
        val db = this.writableDatabase

        val contentValues = ContentValues()
        contentValues.put(KEY_ITEM_ID, item.getId())
        contentValues.put(KEY_TRAINING_NAME, name)
        contentValues.put(KEY_TIMER, item.getTime())
        contentValues.put(KEY_DESCRIPTION,
item.getDescription())

        val success = db.insert(TABLE_NAME, null,
contentValues)
        db.close()
        return success
    }

    // This function reads only the names of the saved
activities
    // It returns a ArrayList of Strings with the
names
    @SuppressLint("Recycle")
    fun
intervalTrainingSessionNames():ArrayList<String>{
        val nameList: ArrayList<String> =
ArrayList<String>()

        val selectQuery = "SELECT $KEY_TRAINING_NAME
FROM $TABLE_NAME"

        val db = this.readableDatabase
        val cursor: Cursor?

        try{
            cursor = db.rawQuery(selectQuery, null)
        } catch (e: SQLException){
            db.execSQL(selectQuery)
            return ArrayList()
        }

        var name: String

        if(cursor.moveToFirst()){
            do{
                name =
cursor.getString(cursor.getColumnIndex(KEY_TRAINING_NA
ME))
```

```
                nameList.add(name)
            }while(cursor.moveToNext())
        }
        return nameList
    }

    // Finally this function reads the Sessions by the
name of a certain activity
    // It returns a list of TrainItems in order to get
shown in the main RecycleView
    @SuppressLint("Recycle")
    fun intervalTrainingItemsInSession(trainName:
String):ArrayList<IntervalTrainingItem>{
        val intervalTrainingItemList:
ArrayList<IntervalTrainingItem> =
ArrayList<IntervalTrainingItem>()

        val selectQuery = "SELECT * FROM $TABLE_NAME
WHERE $KEY_TRAINING_NAME = '$trainName'"

        val db = this.readableDatabase
        val cursor: Cursor

        try{
            cursor = db.rawQuery(selectQuery, null)
        } catch (e: SQLException){
            db.execSQL(selectQuery)
            return ArrayList()
        }

        //var id:Int
        var itemId:Int
        var time:Int
        var description: String

        if(cursor.moveToFirst()){
            do{
                //id =
cursor.getInt(cursor.getColumnIndex(KEY_ID))
                itemId =
cursor.getInt(cursor.getColumnIndex(KEY_ITEM_ID))
                time =
cursor.getInt(cursor.getColumnIndex(KEY_TIMER))
                description =
cursor.getString(cursor.getColumnIndex(KEY_DESCRIPTION
))

                val finalItem =
```

```
IntervalTrainingItem(itemId,description,time)

intervalTrainingItemList.add(finalItem)
            }while(cursor.moveToNext())
        }
        return intervalTrainingItemList
    }
}
```

3. FinishActivity.kt

```
package com.nicknterm.runningapp

import android.media.MediaPlayer
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import
kotlinx.android.synthetic.main.activity_finish.*

class FinishActivity : AppCompatActivity() {
    private var player: MediaPlayer? = null
    override fun onCreate(savedInstanceState: Bundle?)
{
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_finish)

        // Just go to the first activity
        endExerciseButton.setOnClickListener{
            finish()
        }
    }
}
```

4. IntervalTrainingExerciseActivity.kt

```
package com.nicknterm.runningapp

import android.annotation.SuppressLint
import android.app.*
import android.content.Context
import android.content.Intent
import android.graphics.Color
import android.media.MediaPlayer
import android.os.Build
import android.os.Bundle
import android.os.CountDownTimer
import android.view.View
import androidx.appcompat.app.AppCompatActivity
```

```
import
kotlinx.android.synthetic.main.interval_training_exerc
ise_activity.*
import
kotlinx.android.synthetic.main.quit_training_dialog.*


class IntervalTrainingExerciseActivity :
AppCompatActivity() {
    private var player: MediaPlayer? = null
    private var timer: CountDownTimer? = null
    private var intervalTrainingList:
ArrayList<IntervalTrainingItem> = ArrayList()
    private var pauseSecond: Int = 0
    private var isPaused: Boolean = false
    private var position: Int = 0
    private var canPress: Boolean = false


    lateinit var notificationManager:
NotificationManager
    lateinit var notificationChannel:
NotificationChannel
    lateinit var builder: Notification.Builder
    private val channelId = "i.apps.notifications"
    private val description = "Test notification"

    @SuppressLint("UseCompatLoadingForDrawables",
"SetTextI18n")
    override fun onCreate(savedInstanceState: Bundle?)
{
        super.onCreate(savedInstanceState)

setContentView(R.layout.interval_training_exercise_act
ivity)
        intervalTrainingList =
intent.getParcelableArrayListExtra<IntervalTrainingIte
m>("TrainList") as ArrayList<IntervalTrainingItem>

        startTimer(position)

        notificationManager =
getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
        buttonsDisabled()

        player = MediaPlayer.create(this, R.raw.ring)

        LockButton.setOnLongClickListener {
```

```
            if(!canPress) {
                buttonsEnabled()
            }else{
                buttonsDisabled()
            }
            return@setOnLongClickListener true
        }

        // Skip Activity functionality
        // Sets timer for the next activity and resets
progress bar, buttons
        SkipButton.setOnClickListener{
            if(canPress) {
                if (position <
intervalTrainingList.size - 1) {
                    position++
                    startTimer(position)
                    TimerPausedProgressBar.visibility
= View.GONE
                    TimerProgressBar.visibility =
View.VISIBLE
                    TimerProgressBar.max =
TimerProgressBar.max
                    TimerProgressBar.progress =
TimerPausedProgressBar.progress
                    pauseSecond = 0
                    buttonsDisabled()
                    ResumeButton.visibility =
View.GONE
                    PauseButton.visibility =
View.VISIBLE
                    isPaused = false
                } else {
                    val intent =
Intent(this@IntervalTrainingExerciseActivity,
FinishActivity::class.java)
                    intent.putExtra("TrainList",
intervalTrainingList)
                    startActivity(intent)
                    finish()
                }
                buttonsDisabled()
            }
        }

        // Pauses timer and change the progress bar
color by showing other progress bar
        PauseButton.setOnClickListener {
            if(canPress) {
```

```
                if (timer != null) {
                    TimerPausedProgressBar.visibility
= View.VISIBLE
                    TimerProgressBar.visibility =
View.GONE
                    TimerPausedProgressBar.max =
TimerProgressBar.max
                    TimerPausedProgressBar.progress =
TimerProgressBar.progress
                    pauseSecond =
TimerProgressBar.progress
                    ResumeButton.visibility =
View.VISIBLE
                    PauseButton.visibility = View.GONE
                    isPaused = true
                }
            }
        }

        // Resumes timer and change the progress bar
color by showing first progress bar
        ResumeButton.setOnClickListener {
            if(canPress) {
                if (timer != null) {
                    TimerPausedProgressBar.visibility
= View.GONE
                    TimerProgressBar.visibility =
View.VISIBLE
                    TimerProgressBar.max =
TimerPausedProgressBar.max
                    TimerProgressBar.progress =
TimerPausedProgressBar.progress
                    pauseSecond =
TimerProgressBar.progress
                    isPaused = false
                    startTimer(position, pauseSecond)
                    ResumeButton.visibility =
View.GONE
                    PauseButton.visibility =
View.VISIBLE
                }
                buttonsDisabled()
            }
        }
    }

    // This function creates a notification with a
specific Title and Message
    private fun refreshNotifications(message: String,
```

```
Title:String) {
        val mNotificationManager =
getSystemService(NOTIFICATION_SERVICE) as
NotificationManager
        // Sets an ID for the notification, so it can
be updated
        val notifyID = 1
        if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.O) {
            notificationChannel =
NotificationChannel(channelId,
                description,
                NotificationManager.IMPORTANCE_HIGH)
            notificationChannel.enableLights(true)
            notificationChannel.lightColor =
Color.GREEN
            notificationChannel.enableVibration(false)

notificationManager.createNotificationChannel(notifica
tionChannel)

            builder = Notification.Builder(this,
channelId)
                .setSmallIcon(R.drawable.nav_image)
                .setContentTitle(Title)
                .setContentText(message)
                .setAutoCancel(true)
        } else {
            builder = Notification.Builder(this)
                .setSmallIcon(R.drawable.nav_image)
                .setContentTitle(Title)
                .setContentText(message)
                .setAutoCancel(true)
        }
        mNotificationManager.notify(
            notifyID,
            builder.build())
    }


    // Enables the buttons and Disable the lock button
    private fun buttonsEnabled(){

LockButton.setBackgroundResource(R.drawable.text_view_
button_disabled)

SkipButton.setBackgroundResource(R.drawable.text_view_
button_background_ripple)
```

```
PauseButton.setBackgroundResource(R.drawable.text_view
_button_background_ripple)

ResumeButton.setBackgroundResource(R.drawable.text_vie
w_button_background_ripple)
        canPress = true
    }

    // Disables the buttons and Enables the lock
button
    private fun buttonsDisabled(){

LockButton.setBackgroundResource(R.drawable.text_view_
button_background_ripple)

SkipButton.setBackgroundResource(R.drawable.text_view_
button_disabled)

PauseButton.setBackgroundResource(R.drawable.text_view
_button_disabled)

ResumeButton.setBackgroundResource(R.drawable.text_vie
w_button_disabled)
        canPress = false
    }

    // Make sure that when the activity ends the
timer, players are stopped
    override fun onDestroy() {
        player!!.stop()
        timer!!.cancel()
        super.onDestroy()
    }

    // Controls the BackPress
    override fun onBackPressed() {
        showQuitDialog()
    }

    // Shows the Quit Activity Dialog and controls the
ClickListeners of the buttons
    private fun showQuitDialog(){
        val quitDialog = Dialog(this)

quitDialog.setContentView(R.layout.quit_training_dialo
g)

        quitDialog.NoQuitButton.setOnClickListener{
            quitDialog.dismiss()
```

```
        }

        quitDialog.YesQuitButton.setOnClickListener {
            finish()
        }

        quitDialog.show()
    }

    // The main Timer structure. time is the whole
Exercise time
    // and progressPar parameter is for staring the
timer not always from the start
    // if progressPar is negative after the timer ends
it starts the next activity
    // plus it doesn't show the progress bar progress
    private fun startTimer(index: Int, progressPar:
Int = 0) {
        val time:Int
        if(progressPar>= 0) {
            time =
intervalTrainingList[index].getTime()
            DescriptionText.text =
intervalTrainingList[index].getDescription()
        }else{
            time = index
        }
        if (timer != null) {
            timer!!.cancel()
        }

        var progress = progressPar
        TimerProgressBar.max = time
        timer = object : CountDownTimer(((time-
progress) * 1000).toLong(), 1000) {
            @SuppressLint("SetTextI18n")
            override fun onTick(millisUntilFinished:
Long) {
                if(progressPar >= 0) {
                    if (!isPaused) {
                        progress++
                        TimerProgressBar.progress =
progress
                        if ((time - progress) % 60 <
10) {
                            TimerText.text = "${(time
- progress) / 60}:0${(time - progress) % 60}"

refreshNotifications("${(time - progress) /
```

```
60}:0${(time - progress) % 60}", "Activity Started")
                            } else {
                                TimerText.text = "${(time
- progress) / 60}:${(time - progress) % 60}"

refreshNotifications("${(time - progress) /
60}:${(time - progress) % 60}", "Activity Started")
                            }
                        } else {
                            cancel()
                        }
                    }
                }

                override fun onFinish() {
                    if (progressPar >= 0) {
                        if (position <
intervalTrainingList.size - 1) {
                            position++
                            player!!.start()
                            startTimer(position)
                        } else {
                            player =
MediaPlayer.create(this@IntervalTrainingExerciseActivi
ty, R.raw.final_sound)
                            player!!.start()
                            startTimer(2, -1)

refreshNotifications("Congratulations!!!", "Workout
Finished")
                        }
                    }else{
                        val intent =
Intent(this@IntervalTrainingExerciseActivity,
FinishActivity::class.java)
                        intent.putExtra("TrainList",
intervalTrainingList)
                        startActivity(intent)
                        finish()
                    }
                }
            }.start()
        }
}
```

5. IntervalTrainingItem.kt

```
package com.nicknterm.runningapp
```

```
import android.os.Parcel
import android.os.Parcelable


class IntervalTrainingItem(private var id: Int,
private var description: String?, private var time:
Int):
    Parcelable {
    constructor(parcel: Parcel) : this(
        parcel.readInt(),
        parcel.readString(),
        parcel.readInt()
    )


    // Set-Get methods
    fun getId(): Int{
        return id
    }

    fun getDescription(): String? {
        return description
    }

    fun getTime(): Int{
        return time
    }

    fun setTime(v: Int){
        time = v
    }

    fun setDescription(v: String){
        description = v
    }

    fun setId(v: Int){
        id = v
    }


    // Function for Parcelable
    override fun writeToParcel(parcel: Parcel, flags:
Int) {
        parcel.writeInt(id)
        parcel.writeString(description)
        parcel.writeInt(time)
    }
```

```
    override fun describeContents(): Int {
        return 0
    }

    companion object CREATOR :
Parcelable.Creator<IntervalTrainingItem> {
        override fun createFromParcel(parcel: Parcel):
IntervalTrainingItem {
            return IntervalTrainingItem(parcel)
        }

        override fun newArray(size: Int):
Array<IntervalTrainingItem?> {
            return arrayOfNulls(size)
        }
    }
}
```

6. IntervalTrainingLoadRecycleViewAdapter.kt

```
package com.nicknterm.runningapp

import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.LinearLayout
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import
kotlinx.android.synthetic.main.interval_training_load_
recycle_view_item.view.*

class IntervalTrainingLoadRecycleViewAdapter(private
val items: ArrayList<String>, private val context:
Context):

RecyclerView.Adapter<IntervalTrainingLoadRecycleViewAd
apter.ViewHolder>(){
    var selected:Int? = null

    // This is the ViewHolder of the RecycleView. This
holder just "holds"
    // the UI elements so we can later access them.
    // In this way you can refer a certain UI element
in a certain index position
    class ViewHolder(view:
View):RecyclerView.ViewHolder(view) {
        val nameText: TextView =
```

```
view.ItemActivityNameText
        val parentLL: LinearLayout = view.parentLayout
    }

    override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): ViewHolder {
        return
ViewHolder(LayoutInflater.from(context).inflate(R.layo
ut.interval_training_load_recycle_view_item,parent,fal
se))
    }

    // Its just the size of the items
    override fun getItemCount(): Int {
        return items.size
    }

    // This function is called every time something
change or you scroll to more items
    override fun onBindViewHolder(holder: ViewHolder,
position: Int) {
        val item= items[position]

        holder.nameText.text = item
        if(position != selected){

holder.parentLL.setBackgroundResource(R.color.bgSecond
ary)
        }
        holder.parentLL.setOnClickListener {
            selected = position

holder.parentLL.setBackgroundResource(R.drawable.recyc
le_view_item_selected)
            notifyDataSetChanged()
        }
    }

}
```

7. IntervalTrainingMainActivity.kt

```
package com.nicknterm.runningapp

import android.app.*
import android.content.Intent
import android.os.Bundle
import android.view.MenuItem
import android.view.View
```

```
import androidx.appcompat.app.ActionBarDrawerToggle
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.GravityCompat
import androidx.recyclerview.widget.ItemTouchHelper
import
androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import
com.google.android.material.navigation.NavigationView
import com.google.android.material.snackbar.Snackbar
import com.nicknterm.runningapp.R.id.*
import
kotlinx.android.synthetic.main.interval_training_main_
activity.*
import
kotlinx.android.synthetic.main.interval_training_add_s
ession_dialog.*
import
kotlinx.android.synthetic.main.interval_training_save_
dialog.*
import
kotlinx.android.synthetic.main.interval_training_load_
session_dialog.*


@Suppress("DEPRECATION")
class IntervalTrainingMainActivity :
AppCompatActivity(),NavigationView.OnNavigationItemSel
ectedListener {

    var itemList: ArrayList<IntervalTrainingItem> =
ArrayList<IntervalTrainingItem>()
    private var mCurrentId: Int = 0
    private var
intervalTrainingMainRecycleViewAdapter:
IntervalTrainingMainRecycleViewAdapter? = null
    private val dbHandler: DBHandler = DBHandler(this)
    override fun onCreate(savedInstanceState: Bundle?)
{
        super.onCreate(savedInstanceState)

setContentView(R.layout.interval_training_main_activit
y)
        setSupportActionBar(myToolBar) //set Toolbar

        val toggle = ActionBarDrawerToggle(Activity(),
            mainActivityLayout,
            myToolBar,
            R.string.nav_open,
```

```
            R.string.close_nav)
        mainActivityLayout.addDrawerListener(toggle)
        toggle.syncState() //add toggle button for the
Side Navigation

mainNavBar.setNavigationItemSelectedListener(this)
        rvItems.layoutManager =
LinearLayoutManager(this) //set RecycleView layout

        val itemTouchHelper =
ItemTouchHelper(itemTouchHelperCallback)

itemTouchHelper.attachToRecyclerView(rvItems)// use
ItemTouchHelper to th RecycleView

        // Just Show the Add Dialog
        CardViewAdd.setOnClickListener {
            showAddDialog()
        }

        // Just Show the Add Dialog
        addFloatButton.setOnClickListener{
            showAddDialog()
        }

        // Go to The ExerciseActivity and push
ItemList in the Activity
        StartButton.setOnClickListener{
            val intent = Intent(this,
IntervalTrainingExerciseActivity::class.java)
            intent.putExtra("TrainList", itemList)
            startActivity(intent)
        }
    }

    // This function shows the Add Button and hides
the Start Button and the main RecycleViewer
    fun showAddButtons(){
        rvItems.visibility = View.GONE
        CardViewAdd.visibility = View.VISIBLE
        StartButton.visibility = View.GONE
    }

    // This function hides the Add Button and shows
the Start Button and the main RecycleViewer
    private fun hideAddButtons(){
        rvItems.visibility = View.VISIBLE
        CardViewAdd.visibility = View.GONE
        StartButton.visibility = View.VISIBLE
```

```
    }

    // Basically is a OnClickListener of the Items
that get clicked in the Navigation Bar
    override fun onNavigationItemSelected(item:
MenuItem): Boolean {
        when(item.itemId){
            SaveButton -> {
                if (itemList.size > 0) {
                    showSaveDialog()
                } else {
                    Snackbar.make(SnackBarLayout,
                        "Cannot save and Empty
activity",
                        Snackbar.LENGTH_LONG)

.setTextColor(resources.getColor((R.color.textColor)))

.setBackgroundTint(resources.getColor(R.color.bgSecond
ary))
                        .show()
                }
            }
            LoadButton -> showSelectActivityDialog()
        }

mainActivityLayout.closeDrawer(GravityCompat.START)
        return true
    }

    // Shows the Save Dialog and controls the
ClickListeners of the Buttons
    private fun showSaveDialog() {
        val saveDialog = Dialog(this)

saveDialog.setContentView(R.layout.interval_training_s
ave_dialog)

saveDialog.DialogSaveButton.setOnClickListener{

if(saveDialog.NameInputSave.text.toString().isNotEmpty
()) {
                for (item in itemList){

dbHandler.saveIntervalTrainingItem(item,
saveDialog.NameInputSave.text.toString())
                    Snackbar.make(SnackBarLayout,
"Saved Successfully", Snackbar.LENGTH_LONG)
```

```
.setTextColor(resources.getColor((R.color.textColor)))

.setBackgroundTint(resources.getColor(R.color.bgSecond
ary))
                            .show()
                    }
                    saveDialog.dismiss()
                }else{
                    saveDialog.NameInputSave.error =
"Please Enter Name"
                }
            }

saveDialog.DialogCancelButton.setOnClickListener {
                saveDialog.dismiss()
            }
            saveDialog.show()
        }

        // Shows the Select Activity Dialog and controls
the ClickListeners of the Buttons
        private fun showSelectActivityDialog(){
            val selectDialog = Dialog(this)

selectDialog.setContentView(R.layout.interval_training
_load_session_dialog)
            selectDialog.SelectActivityRv.layoutManager =
LinearLayoutManager(this)
            var selectDialogAdapter:
IntervalTrainingLoadRecycleViewAdapter
            val nameList =
dbHandler.intervalTrainingSessionNames()
            if(nameList.isNotEmpty()) {
                val list = ArrayList<String>()
                for(item in nameList){
                   if(!list.contains(item)){
                       list.add(item)
                   }
                }
                selectDialogAdapter =
IntervalTrainingLoadRecycleViewAdapter(list, this)
                selectDialog.SelectActivityRv.adapter =
selectDialogAdapter
            }else{
                selectDialog.NoWorkoutText.visibility =
View.VISIBLE
                selectDialog.SelectActivityRv.visibility =
View.GONE
            }
```

```
selectDialog.DialogShowSavedSelectButton.setOnClickLis
tener {
            if(nameList.isNotEmpty()) {
                val list = ArrayList<String>()
                for(item in nameList){
                    if(!list.contains(item)){
                        list.add(item)
                    }
                }
                selectDialogAdapter =
selectDialog.SelectActivityRv.adapter as
IntervalTrainingLoadRecycleViewAdapter
                if(selectDialogAdapter.selected!=
null) {
                    itemList.clear()
                    itemList =
dbHandler.intervalTrainingItemsInSession(list[selectDi
alogAdapter.selected!!])

intervalTrainingMainRecycleViewAdapter =
IntervalTrainingMainRecycleViewAdapter(itemList, this)
                    rvItems.adapter =
intervalTrainingMainRecycleViewAdapter
                    hideAddButtons()

intervalTrainingMainRecycleViewAdapter!!.notifyDataSet
Changed()
                    selectDialog.dismiss()
                }
            }
        }

selectDialog.DialogShowSavedCancelButton.setOnClickLis
tener {
            selectDialog.dismiss()
        }
        selectDialog.show()
    }

    // Shows the Add Activity Dialog and controls the
ClickListeners of the Buttons
    private fun showAddDialog() {
        val addDialog = Dialog(this)

addDialog.setContentView(R.layout.interval_training_ad
d_session_dialog)

addDialog.cancel_button_add_dialog.setOnClickListener{
```

```
            addDialog.dismiss()
        }

addDialog.add_button_add_dialog.setOnClickListener{

if(addDialog.DescriptionTextInput.text.toString().isNo
tEmpty() &&
addDialog.TimeTextInput.text.toString().isNotEmpty())
{
                val newItem =
IntervalTrainingItem(mCurrentId,

addDialog.DescriptionTextInput.text.toString(),

addDialog.TimeTextInput.text.toString().toInt())
                mCurrentId++
                itemList.add(newItem)
                if
(intervalTrainingMainRecycleViewAdapter != null) {

intervalTrainingMainRecycleViewAdapter!!.notifyDataSet
Changed()
                } else {

intervalTrainingMainRecycleViewAdapter =
IntervalTrainingMainRecycleViewAdapter(itemList, this)
                    rvItems.adapter =
intervalTrainingMainRecycleViewAdapter
                }
                hideAddButtons()
                addDialog.dismiss()
            }else{

if(addDialog.DescriptionTextInput.text.toString().isEm
pty()){

addDialog.DescriptionTextInputLayout.error = "Please
Enter Description"
                }

if(addDialog.TimeTextInput.text.toString().isEmpty()){

addDialog.TimeTextInputLayout.error = "Please Enter
Time"
                }
            }
        }
        addDialog.show()
    }
```

```
    // This object controls the drag and drop, the
swipe functionality of the main RecycleView
    private val itemTouchHelperCallback = object:
ItemTouchHelper.Callback() {
        override fun getMovementFlags(
            recyclerView: RecyclerView,
            viewHolder: RecyclerView.ViewHolder,
        ): Int {
            return
makeMovementFlags(ItemTouchHelper.UP or
ItemTouchHelper.DOWN,
                ItemTouchHelper.RIGHT)
        }

        override fun onMove(
            recyclerView: RecyclerView,
            viewHolder: RecyclerView.ViewHolder,
            target: RecyclerView.ViewHolder,
        ): Boolean {

intervalTrainingMainRecycleViewAdapter!!.notifyItemMov
ed(viewHolder.adapterPosition, target.adapterPosition)
            itemList[viewHolder.adapterPosition] =
itemList[target.adapterPosition].also
{itemList[target.adapterPosition] =
itemList[viewHolder.adapterPosition]}
            return true
        }

        override fun isLongPressDragEnabled(): Boolean
{
            return true
        }

        override fun onSwiped(viewHolder:
RecyclerView.ViewHolder, direction: Int) {
            val removedItem =
itemList[viewHolder.adapterPosition]
            val position = viewHolder.adapterPosition
            itemList.removeAt(position)

intervalTrainingMainRecycleViewAdapter!!.notifyDataSet
Changed()
            Snackbar.make(SnackBarLayout, "Item
Deleted", Snackbar.LENGTH_LONG)
                .setAction("Undo") {
                    itemList.add(position,
removedItem)
```

```
intervalTrainingMainRecycleViewAdapter!!.notifyDataSet
Changed()
                    hideAddButtons()
                }

.setTextColor(resources.getColor(R.color.textColor))

.setBackgroundTint(resources.getColor(R.color.bgSecond
ary))

.setActionTextColor(resources.getColor(R.color.cyan))
                    .show()
                if(itemList.size == 0){
                    showAddButtons()
                }
            }
        }
}
```

8. IntervalTrainingMainRecycleViewAdapter.kt

```
package com.nicknterm.runningapp

import android.annotation.SuppressLint
import android.app.Dialog
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import
kotlinx.android.synthetic.main.interval_training_add_s
ession_dialog.*
import
kotlinx.android.synthetic.main.interval_training_edit_
session_dialog.*
import
kotlinx.android.synthetic.main.internal_training_main_
recycle_view_item.view.*

class IntervalTrainingMainRecycleViewAdapter(private
val items: ArrayList<IntervalTrainingItem>, private
val context:
Context):RecyclerView.Adapter<IntervalTrainingMainRecy
cleViewAdapter.ViewHolder>() {
    // This is the ViewHolder of the RecycleView. This
```

```
holder just "holds"
    // the UI elements so we can later access them.
    // In this way you can refer a certain UI element
in a certain index position
    class ViewHolder(view:
View):RecyclerView.ViewHolder(view) {
        val timeText: TextView = view.ItemTimeText
        val descriptionText: TextView =
view.ItemDescriptionText
        val deleteButton: ImageView =
view.DeleteButton
    }


    override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): ViewHolder {
        return
ViewHolder(LayoutInflater.from(context).inflate(R.layo
ut.internal_training_main_recycle_view_item, parent,
false))
    }

    // This function is called every time something
change or you scroll to more items
    @SuppressLint("SetTextI18n")
    override fun onBindViewHolder(holder: ViewHolder,
position: Int) {
        val item= items[position]
        holder.timeText.text = "${item.getTime()}
Seconds"
        holder.descriptionText.text =
item.getDescription()
        holder.deleteButton.setOnClickListener {
            showEditDialog(position)
        }
    }

    // Its just the size of the items
    override fun getItemCount(): Int {
        return items.size
    }

    // Shows the Edit Element Dialog and controls its
Buttons
    private fun showEditDialog(position: Int) {
        val editDialog = Dialog(context)

editDialog.setContentView(R.layout.interval_training_e
dit_session_dialog)
```

```
editDialog.NoEditDialogButton.setOnClickListener {
            editDialog.dismiss()
        }

editDialog.YesEditDialogButton.setOnClickListener {

if(editDialog.DescriptionEditTextInput.text.toString()
.isNotEmpty() &&
editDialog.TimeEditTextInput.text.toString().isNotEmpt
y()) {
                val activityIntervalTraining:
IntervalTrainingMainActivity = context as
IntervalTrainingMainActivity

activityIntervalTraining.itemList[position] =
IntervalTrainingItem(items[position].getId(),

editDialog.DescriptionEditTextInput.text.toString(),

editDialog.TimeEditTextInput.text.toString().toInt())
                notifyDataSetChanged()
                editDialog.dismiss()
            }else{

if(editDialog.DescriptionEditTextInput.text.toString()
.isEmpty()){

editDialog.DescriptionTextInputLayout.error = "Please
Enter Description"
                }

if(editDialog.TimeEditTextInput.text.toString().isEmpt
y()){

editDialog.TimeTextEditInputLayout.error = "Please
Enter Time"
                }
            }
        }
        editDialog.show()
    }
}
```

9. MainMenu.kt

```
package com.nicknterm.runningapp

import android.app.Dialog
```

```
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import
com.google.android.material.dialog.MaterialAlertDialog
Builder
import kotlinx.android.synthetic.main.main_menu.*
import
kotlinx.android.synthetic.main.quit_app_dialog.*

class MainMenu : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?)
{
        super.onCreate(savedInstanceState)
        setContentView(R.layout.main_menu)
        setSupportActionBar(MainMenuToolbar)
        StartIntervalTraining.setOnClickListener{
            startActivity(Intent(this,
IntervalTrainingMainActivity::class.java))
        }
        StartStopwatch.setOnClickListener {
            startActivity(Intent(this,
Stopwatch::class.java))
        }
    }

    // The function that controls the functionality of
BackPress
    override fun onBackPressed() {
        showQuitDialog()
    }

    // Shows the Quit From the App Dialog and controls
the ClickListeners of the Buttons
    private fun showQuitDialog() {
        /*MaterialAlertDialogBuilder(this)
            .setTitle("Quit?")
            .setMessage("Are you sure you want to
leave the app?")
            .setNegativeButton("leave") { dialog,
which ->
                dialog.dismiss()
            }
            .setPositiveButton("Stay") { dialog, which
->
                finish()
            }
            .show()
*/
```

```
        val quitDialog = Dialog(this)

quitDialog.setContentView(R.layout.quit_app_dialog)

quitDialog.YesQuitAppButton.setOnClickListener{
            finish()
        }
        quitDialog.NoQuitAppButton.setOnClickListener
{
            quitDialog.dismiss()
        }
        quitDialog.show()
    }
}
```

10. Stopwatch.kt

```
package com.nicknterm.runningapp

import android.annotation.SuppressLint
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.view.View
import
androidx.recyclerview.widget.LinearLayoutManager
import
kotlinx.android.synthetic.main.stopwatch_main_activit
y.*
import kotlin.collections.ArrayList

class Stopwatch : AppCompatActivity() {
    private var startTime: Long = 0
    private var pausedTime: Long = 0
    private var timeHandler: Handler? = null
    private var isPaused: Boolean = false
    private var timeToStart: Boolean = true
    private var lapTimes: ArrayList<StopwatchLapItem>
= ArrayList<StopwatchLapItem>()
    private var lapId:Int = -1
    private var bestLapId = 0
    private var worstLapId = 0
    private var lastLapTime:Long = 0

    private var
RvAdapter:StopwatchLapRecycleViewAdapter? = null

    override fun onCreate(savedInstanceState:
Bundle?) {
```

```
        super.onCreate(savedInstanceState)

setContentView(R.layout.stopwatch_main_activity)
        timeHandler = Handler()
        StopwatchRecycleView.layoutManager =
LinearLayoutManager(this)
        RvAdapter =
StopwatchLapRecycleViewAdapter(lapTimes, this)
        StopwatchRecycleView.adapter = RvAdapter

        StopwatchTimerText.setOnClickListener{
            if(PauseButton.text.toString() ==
"Start"){
                PauseButton.text = "Pause"
            }
            if(timeToStart) {
                startStopwatch(pausedTime)
            }else{
                pauseStopwatch()
            }
            timeToStart = !timeToStart
        }
        PauseButton.setOnClickListener {
            if(timeToStart) {
                PauseButton.text = "Pause"
                startStopwatch(pausedTime)
            }else{
                PauseButton.text = "Resume"
                pauseStopwatch()

            }
            timeToStart = !timeToStart
        }
        LapButton.setOnClickListener {
            createLap()
        }
    }

    private fun createLap() {
        if(!isPaused) {

StopwatchRecycleView.smoothScrollToPosition(0)
            lapId++
            if(lapTimes.size > 0) {
                var isBest: Boolean = false
                var isWorst: Boolean = false
                if (lapTimes[bestLapId].getLapTime()
> (System.currentTimeMillis() - startTime) -
lastLapTime) {
```

```
lapTimes[worstLapId].setIsTheWorst(true)

lapTimes[bestLapId].setIsTheBest(false)
                    bestLapId = lapId
                    isBest = true
                } else if
(lapTimes[worstLapId].getLapTime() <
(System.currentTimeMillis() - startTime) -
lastLapTime) {

lapTimes[bestLapId].setIsTheBest(true)

lapTimes[worstLapId].setIsTheWorst(false)
                    worstLapId = lapId
                    isWorst = true
                }

                lapTimes.add(StopwatchLapItem(lapId,
                    System.currentTimeMillis() -
startTime,
                    (System.currentTimeMillis() -
startTime) - lastLapTime,
                    isBest,
                    isWorst))
                lastLapTime =
System.currentTimeMillis() - startTime

                RvAdapter!!.notifyDataSetChanged()
            }else{
                lapTimes.add(StopwatchLapItem(lapId,
                    System.currentTimeMillis() -
startTime,
                    (System.currentTimeMillis() -
startTime) - lastLapTime,
                    false,
                    false))
                lastLapTime =
System.currentTimeMillis() - startTime
                lapTimes.reverse()
                RvAdapter!!.notifyDataSetChanged()
                lapTimes.reverse()
            }
        }
    }

    private fun startStopwatch(startFrom: Long = 0){
        StopWatchProgressBar.visibility =
View.VISIBLE
```

```
        StopWatchPausedProgressBar.visibility =
View.GONE
        LapButton.visibility = View.VISIBLE
        isPaused = false
        startTime = System.currentTimeMillis() -
startFrom
        val myRunnable: Runnable = object : Runnable
{
            @SuppressLint("SetTextI18n")
            override fun run() {
                if(!isPaused) {
                    var millis =
System.currentTimeMillis() - startTime
                    var seconds: Int = (millis /
1000).toInt()
                    val minutes = seconds / 60
                    seconds %= 60
                    millis %= 1000
                    millis /= 10
                    StopwatchTimerText.text =
                        String.format("%d:%02d.%02d",
minutes, seconds, millis)
                    timeHandler!!.postDelayed(this,
10)
                }else{

timeHandler!!.removeCallbacks(this)
                }
            }
        }
        timeHandler!!.post(myRunnable)
    }

    private fun pauseStopwatch(){
        LapButton.visibility = View.GONE
        StopWatchProgressBar.visibility = View.GONE
        StopWatchPausedProgressBar.visibility =
View.VISIBLE
        pausedTime = System.currentTimeMillis() -
startTime
        isPaused = true
    }
}
```

11. StopwatchLapItem.kt

```
package com.nicknterm.runningapp

class StopwatchLapItem(private val id: Int, private
```

```
val splitTime: Long, private val lapTime: Long,
private var isTheBest:Boolean,private var
isTheWorst:Boolean) {

    fun getId():Int{
        return id
    }

    fun getSplitTime():Long{
        return splitTime
    }

    fun getLapTime():Long{
        return lapTime
    }

    fun getIsTheBest():Boolean{
        return isTheBest
    }

    fun getIsTheWorst():Boolean{
        return isTheWorst
    }

    fun setIsTheBest(v: Boolean){
        isTheBest = v
    }

    fun setIsTheWorst(v: Boolean){
        isTheWorst = v
    }
}
```

12. StopwatchLapRecycleViewAdapter.kt

```
package com.nicknterm.runningapp

import android.annotation.SuppressLint
import android.content.Context
import android.graphics.Color
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.core.content.ContextCompat
import androidx.recyclerview.widget.RecyclerView
import
kotlinx.android.synthetic.main.stopwatch_lap_recycle_
view_item.view.*
```

```
class StopwatchLapRecycleViewAdapter(private val
items: ArrayList<StopwatchLapItem>, private val
context: Context):

RecyclerView.Adapter<StopwatchLapRecycleViewAdapter.V
iewHolder>() {
    // This is the ViewHolder of the RecycleView.
This holder just "holds"
    // the UI elements so we can later access them.
    // In this way you can refer a certain UI element
in a certain index position
    class ViewHolder(view: View):
RecyclerView.ViewHolder(view) {
        val lapId: TextView = view.IdOfLap
        val splitTime: TextView = view.SplitTime
        val lapTime: TextView = view.LapTime
    }


    override fun onCreateViewHolder(parent:
ViewGroup, viewType: Int): ViewHolder {
        return
ViewHolder(LayoutInflater.from(context).inflate(R.lay
out.stopwatch_lap_recycle_view_item, parent, false))
    }

    // This function is called every time something
change or you scroll to more items
    @SuppressLint("SetTextI18n", "ResourceAsColor")
    override fun onBindViewHolder(holder: ViewHolder,
position: Int) {
        val item= items[(items.size-1) - position]
        holder.lapId.text =
String.format("%02d",item.getId())
        holder.splitTime.text =
longToString(item.getSplitTime())
        holder.lapTime.text =
longToString(item.getLapTime())
        if (item.getIsTheBest()){

holder.lapId.setTextColor(ContextCompat.getColor(cont
ext, R.color.cyan))

holder.splitTime.setTextColor(ContextCompat.getColor(
context, R.color.cyan))

holder.lapTime.setTextColor(ContextCompat.getColor(co
ntext, R.color.cyan))
```

```
        }else if(item.getIsTheWorst()){

holder.lapId.setTextColor(ContextCompat.getColor(cont
ext, R.color.red))

holder.splitTime.setTextColor(ContextCompat.getColor(
context, R.color.red))

holder.lapTime.setTextColor(ContextCompat.getColor(co
ntext, R.color.red))
        }else{

holder.lapId.setTextColor(ContextCompat.getColor(cont
ext, R.color.textColor))

holder.splitTime.setTextColor(ContextCompat.getColor(
context, R.color.textColor))

holder.lapTime.setTextColor(ContextCompat.getColor(co
ntext, R.color.textColor))
        }
    }

    private fun longToString(l: Long): String{
        var millis = l
        var seconds: Int = (millis / 1000).toInt()
        val minutes = seconds / 60
        seconds %= 60
        millis %= 1000
        millis /= 10
        return String.format("%d:%02d.%02d", minutes,
seconds, millis)
    }

    // Its just the size of the items
    override fun getItemCount(): Int {
        return items.size
    }
}
```

13. Activity_finish.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
        android:background="@color/dark"
        android:orientation="vertical"
        tools:context=".FinishActivity">

        <TextView
            android:layout_marginTop="100dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Congratulations!!"
            android:textColor="@color/textColor"
            android:textSize="20sp"
            android:layout_gravity="center"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="You finish the workout"
            android:textColor="@color/textColor"
            android:textSize="20sp"
            android:layout_gravity="center"/>

        <TextView
            android:id="@+id/endExerciseButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="30dp"

    android:background="@drawable/text_view_button_backgr
    ound_ripple"
            android:paddingHorizontal="30dp"
            android:paddingVertical="10dp"
            android:text="FINISH"
            android:textColor="@color/textColor"
            android:textSize="@dimen/TextViewButtonText"
            android:textStyle="bold" />
    </LinearLayout>
```

14. Internal_training_main_recycle_view_item.xml

```
    <?xml version="1.0" encoding="utf-8"?>
    <androidx.cardview.widget.CardView

    xmlns:android="http://schemas.android.com/apk/res/and
    roid"
        android:layout_width="match_parent"

    app:cardCornerRadius="@dimen/RecycleViewItemCornerRad
```

```
ius"
    android:layout_marginHorizontal="10dp"
    android:layout_marginVertical="5dp"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    xmlns:app="http://schemas.android.com/apk/res-
auto">
    <LinearLayout
        android:id="@+id/llItem"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/bgSecondary"
        android:orientation="horizontal">
        <ImageView
            android:layout_width="30dp"
            android:layout_height="30dp"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:layout_gravity="center_vertical"
            android:src="@drawable/timer"/>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <TextView
                android:id="@+id/ItemTimeText"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingTop="10dp"
                android:textSize="20sp"
                android:textStyle="bold"
                android:textColor="@color/textColor"
                android:text="TEST"/>
            <TextView
                android:id="@+id/ItemDescriptionText"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingVertical="6dp"
                android:textSize="17sp"
                android:textStyle="bold"

android:textColor="@color/textColorSecondary"
                android:text="slow Run"/>

        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_gravity="center"
            android:gravity="right"
```

```
                android:layout_height="wrap_content">
                <ImageView
                    android:id="@+id/DeleteButton"
                    android:onClick="deleteItem"
                    android:layout_width="25dp"
                    android:layout_height="25dp"
                    android:layout_marginEnd="15dp"
                    android:gravity="right"
                    app:srcCompat="@drawable/edit_image"
/>
        </LinearLayout>
    </LinearLayout>


</androidx.cardview.widget.CardView>
```

15. Interval_add_session_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    android:orientation="vertical"
    android:background="@color/bgSecondary"
    android:layout_width="350dp"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/textColor"
        android:layout_marginTop="10dp"

android:layout_marginHorizontal="@dimen/MarginHorizon
talDialog"
        android:textSize="@dimen/DialogTitleText"
        android:text="Add Activity"/>

<com.google.android.material.textfield.TextInputLayou
t
        android:id="@+id/DescriptionTextInputLayout"
        android:paddingHorizontal="20dp"
        android:paddingTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Description"
        app:errorEnabled="true"
```

```
style="@style/Widget.MaterialComponents.TextInputLayo
ut.OutlinedBox">


<com.google.android.material.textfield.TextInputEditT
ext

            android:id="@+id/DescriptionTextInput"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            />

</com.google.android.material.textfield.TextInputLayo
ut>

<com.google.android.material.textfield.TextInputLayou
t
        android:id="@+id/TimeTextInputLayout"
        android:paddingHorizontal="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Time in seconds"

style="@style/Widget.MaterialComponents.TextInputLayo
ut.OutlinedBox">


<com.google.android.material.textfield.TextInputEditT
ext
            android:id="@+id/TimeTextInput"
            android:numeric="integer"
            android:inputType="number"
            android:maxLength="4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            />

</com.google.android.material.textfield.TextInputLayo
ut>
    <LinearLayout
        android:layout_marginVertical="20dp"
        android:layout_width="match_parent"
        android:orientation="horizontal"
        android:gravity="right"
        android:layout_height="wrap_content">
        <TextView

android:id="@+id/cancel_button_add_dialog"
            android:paddingRight="20dp"
```

```
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="@color/cyan"
                android:text="Cancel"

    android:textSize="@dimen/DialogOptionText"/>
            <TextView
                android:id="@+id/add_button_add_dialog"
                android:paddingRight="20dp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="@color/cyan"
                android:text="Add"

    android:textSize="@dimen/DialogOptionText"/>
        </LinearLayout>
    </LinearLayout>
```

16. Interval_add_edit_session_dialog.xml

```
    <?xml version="1.0" encoding="utf-8"?>
    <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/and
    roid"
        android:layout_width="@dimen/DialogWidth"
        android:layout_height="wrap_content"
        xmlns:app="http://schemas.android.com/apk/res-
    auto"
        android:background="@color/dark"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="@color/textColor"
            android:textSize="@dimen/DialogTitleText"

    android:layout_marginHorizontal="@dimen/MarginHorizon
    talDialog"

    android:layout_marginTop="@dimen/TitleTopMargin"
            android:text="Edit"/>


    <com.google.android.material.textfield.TextInputLayou
    t
            android:id="@+id/DescriptionEditTextLayout"
            android:paddingHorizontal="20dp"
            android:paddingTop="10dp"
            android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:hint="Description"
        app:errorEnabled="true"

style="@style/Widget.MaterialComponents.TextInputLayo
ut.OutlinedBox">

<com.google.android.material.textfield.TextInputEditT
ext

android:id="@+id/DescriptionEditTextInput"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            />

</com.google.android.material.textfield.TextInputLayo
ut>

<com.google.android.material.textfield.TextInputLayou
t
        android:id="@+id/TimeTextEditInputLayout"
        android:paddingHorizontal="20dp"
        android:paddingTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Time in second"
        app:errorEnabled="true"

style="@style/Widget.MaterialComponents.TextInputLayo
ut.OutlinedBox">

<com.google.android.material.textfield.TextInputEditT
ext
            android:id="@+id/TimeEditTextInput"
            android:numeric="integer"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            />

</com.google.android.material.textfield.TextInputLayo
ut>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="right"

android:layout_marginBottom="@dimen/OptionButtonsBott
omMargin"
```

```
            android:orientation="horizontal">
            <TextView
                android:id="@+id/NoEditDialogButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"

android:paddingVertical="@dimen/OptionPaddingVertical
"
                android:text="Cancel"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"

android:textSize="@dimen/DialogOptionText"
                android:textColor="@color/cyan"
                android:gravity="right"/>
            <TextView
                android:id="@+id/YesEditDialogButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"

android:paddingVertical="@dimen/OptionPaddingVertical
"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"
                android:text="Done"

android:textSize="@dimen/DialogOptionText"
                android:textColor="@color/cyan"
                android:gravity="right"/>
        </LinearLayout>
</LinearLayout>
```

17. Interval_taining_exercise_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:background="@color/dark"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent"

tools:context=".IntervalTrainingExerciseActivity">
    <TextView
        android:id="@+id/DescriptionText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Description"
        android:layout_marginTop="30dp"
        android:layout_gravity="center"
        android:textColor="@color/textColor"
        android:textSize="25sp"/>
    <FrameLayout
        android:layout_width="300dp"
        android:layout_margin="30dp"
        android:layout_gravity="center"
        android:layout_height="300dp">

        <ProgressBar
            android:id="@+id/TimerProgressBar"

style="@style/Widget.AppCompat.ProgressBar.Horizontal
"
            android:layout_width="300dp"
            android:layout_height="300dp"
            android:layout_gravity="center"
            android:indeterminate="false"

android:background="@drawable/time_progress_bar_backg
round"
            android:max="10"
            android:progress="6"

android:progressDrawable="@drawable/timer_progress_ba
r"
            android:rotation="-90"
            android:rotationY="180" />
        <ProgressBar
            android:visibility="gone"
            android:id="@+id/TimerPausedProgressBar"

style="@style/Widget.AppCompat.ProgressBar.Horizontal
"
            android:layout_width="300dp"
            android:layout_height="300dp"
            android:layout_gravity="center"
```

```
            android:indeterminate="false"

android:background="@drawable/time_progress_bar_backg
round"
            android:max="10"
            android:progress="4"

android:progressDrawable="@drawable/timer_progress_ba
r_paused"
            android:rotation="-90"
            android:rotationY="180" />
        <TextView
            android:id="@+id/TimerText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="0:96"
            android:textColor="@color/textColor"
            android:textSize="40sp" />
    </FrameLayout>


    <LinearLayout
        android:id="@+id/mainLL"
        android:layout_width="match_parent"
        android:layout_gravity="top"
        android:gravity="bottom|center"
        android:orientation="horizontal"
        android:layout_height="match_parent">
        <TextView
            android:clickable="false"
            android:id="@+id/SkipButton"
            android:layout_width="130dp"
            android:gravity="center"
            android:layout_height="wrap_content"
            android:layout_gravity="right|bottom"
            android:layout_marginBottom="10dp"

android:background="@drawable/text_view_button_disabl
ed"
            android:paddingHorizontal="20dp"
            android:paddingVertical="10dp"
            android:text="Skip"
            android:textColor="@color/textColor"

android:textSize="@dimen/TextViewButtonText" />

        <ImageButton
            android:id="@+id/LockButton"
```

```
                android:layout_width="60dp"
                android:layout_height="60dp"
                android:layout_margin="15dp"
                android:padding="10dp"
                android:scaleType="fitXY"

        android:background="@drawable/text_view_button_backgr
        ound_ripple"
                android:src="@drawable/lock"/>
            <TextView
                android:clickable="false"
                android:id="@+id/PauseButton"
                android:layout_width="130dp"
                android:gravity="center"
                android:layout_height="wrap_content"
                android:text="Pause"
                android:layout_gravity="right|bottom"
                android:layout_marginBottom="10dp"

        android:textSize="@dimen/TextViewButtonText"
                android:paddingVertical="10dp"
                android:paddingHorizontal="20dp"
                android:textColor="@color/textColor"

        android:background="@drawable/text_view_button_disabl
        ed"/>
            <TextView
                android:id="@+id/ResumeButton"
                android:visibility="gone"
                android:layout_width="130dp"
                android:gravity="center"
                android:layout_height="wrap_content"
                android:text="Resume"
                android:layout_gravity="right|bottom"
                android:layout_marginBottom="10dp"

        android:textSize="@dimen/TextViewButtonText"
                android:paddingVertical="10dp"
                android:paddingHorizontal="20dp"
                android:textColor="@color/textColor"

        android:background="@drawable/text_view_button_backgr
        ound_ripple"/>
        </LinearLayout>

    </LinearLayout>
```

18. Interval_training_load_recycle_view_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    android:orientation="vertical"

app:cardCornerRadius="@dimen/RecycleViewItemCornerRad
ius"
    android:layout_marginHorizontal="10dp"
    android:layout_marginVertical="5dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <LinearLayout
        android:id="@+id/parentLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/bgSecondary">
        <TextView
            android:id="@+id/ItemActivityNameText"
            android:layout_width="match_parent"
            android:text="Test"
            android:padding="10dp"
            android:gravity="center"
            android:textColor="@color/textColor"

android:textSize="@dimen/DialogNormalText"
            android:layout_height="match_parent"/>

    </LinearLayout>

</androidx.cardview.widget.CardView>
```

19. Interval_training_load_session_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    android:orientation="vertical"
    android:layout_width="@dimen/DialogWidth"
    android:background="@color/dark"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
android:layout_marginHorizontal="@dimen/MarginHorizon
talDialog"
        android:layout_marginBottom="10dp"
        android:textColor="@color/textColor"

android:layout_marginTop="@dimen/TitleTopMargin"
        android:textSize="@dimen/DialogTitleText"
        android:text="Select Activity"/>
    <androidx.recyclerview.widget.RecyclerView
        android:visibility="visible"
        android:id="@+id/SelectActivityRv"
        android:layout_width="match_parent"
        android:layout_height="180dp"/>
    <TextView
        android:id="@+id/NoWorkoutText"
        android:layout_width="wrap_content"
        android:layout_height="180dp"
        android:visibility="gone"
        android:gravity="center"
        android:textSize="@dimen/DialogNormalText"
        android:textColor="@color/textColor"
        android:text="You don't have save any
workout"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:gravity="right"
        android:layout_marginTop="10dp"

android:layout_marginBottom="@dimen/OptionButtonsBott
omMargin"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView

android:id="@+id/DialogShowSavedCancelButton"
            android:layout_width="wrap_content"

android:paddingVertical="@dimen/OptionPaddingVertical
"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"
            android:layout_height="wrap_content"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"

android:background="@drawable/dialog_text_view_button
```

```
_background_ripple"

android:textSize="@dimen/DialogOptionText"
            android:textColor="@color/cyan"
            android:text="Cancel"/>
        <TextView

android:id="@+id/DialogShowSavedSelectButton"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"

android:background="@drawable/dialog_text_view_button
_background_ripple"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:paddingVertical="@dimen/OptionPaddingVertical
"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"

android:textSize="@dimen/DialogOptionText"
            android:textColor="@color/cyan"
            android:text="Select"/>
    </LinearLayout>
</LinearLayout>
```

20. Interval_training_main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:background="@color/dark"
    android:id="@+id/mainActivityLayout"
    android:layout_height="match_parent"
    tools:context=".IntervalTrainingMainActivity">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    <LinearLayout
        android:id="@+id/linearLayout"
```

```
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <androidx.appcompat.widget.Toolbar
                android:id="@+id/myToolBar"

style="@style/Theme.MaterialComponents.DayNight"
                android:layout_width="match_parent"

android:layout_height="?android:attr/actionBarSize"/>

            <androidx.recyclerview.widget.RecyclerView
                android:id="@+id/rvItems"
                android:layout_width="match_parent"
                android:visibility="gone"
                android:layout_height="match_parent" />

            <androidx.cardview.widget.CardView
                android:id="@+id/CardViewAdd"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_marginTop="30dp"

android:backgroundTint="@color/bgSecondary"
                app:cardCornerRadius="10dp">

                <TextView
                    android:id="@+id/addInfoText"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_margin="15dp"
                    android:gravity="center"
                    android:text="Add New Activities"
                    android:textColor="@color/cyan"

android:textSize="@dimen/TextViewButtonText" />
            </androidx.cardview.widget.CardView>
        </LinearLayout>
<androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">


<com.google.android.material.floatingactionbutton.Flo
atingActionButton
            android:id="@+id/addFloatButton"
            android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
            android:layout_gravity="bottom|end"
            android:layout_margin="20dp"
            android:src="@drawable/add_icon"
            app:layout_anchorGravity="center"

        app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent" />

        <TextView
            android:id="@+id/StartButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="bottom|center"
            android:layout_margin="20dp"

        android:background="@drawable/text_view_button_backgr
        ound_ripple"
            android:paddingHorizontal="50dp"
            android:paddingVertical="10dp"
            android:text="START"
            android:textColor="@color/white"
            android:textSize="@dimen/TextViewButtonText"
            android:textStyle="bold"
            android:visibility="gone"

        app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
        />
        </androidx.constraintlayout.widget.ConstraintLayout>


        </RelativeLayout>

        <androidx.coordinatorlayout.widget.CoordinatorLayout
            android:layout_width="match_parent"
            android:id="@+id/SnackBarLayout"
            android:layout_marginBottom="80dp"
            android:layout_height="match_parent"/>

        <com.google.android.material.navigation.NavigationVie
        w
            android:id="@+id/mainNavBar"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="start"
            android:background="@color/dark"
            android:fitsSystemWindows="true"
```

```
app:headerLayout="@layout/interval_training_navigatio
n_bar_header"
        app:menu="@menu/menu_nav_bar"/>
</androidx.drawerlayout.widget.DrawerLayout>
```

21. Interval_training_navigation_bar_header.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    android:orientation="vertical"
android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:background="@color/bgSecondary"
        android:layout_gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="120dp"
        android:paddingVertical="20dp"
        android:src="@drawable/nav_image"/>

</LinearLayout>
```

22. Interval_training_save_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    android:orientation="vertical"
    android:layout_width="@dimen/DialogWidth"
    android:background="@color/dark"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="@dimen/DialogTitleText"
        android:text="Save Training"

android:layout_marginHorizontal="@dimen/MarginHorizon
talDialog"

android:layout_marginTop="@dimen/TitleTopMargin"
        android:textColor="@color/textColor"/>
```

```
<com.google.android.material.textfield.TextInputLayou
t
        android:paddingHorizontal="20dp"
        android:paddingTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        app:errorEnabled="true"

style="@style/Widget.MaterialComponents.TextInputLayo
ut.OutlinedBox">


<com.google.android.material.textfield.TextInputEditT
ext
            android:id="@+id/NameInputSave"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            />

</com.google.android.material.textfield.TextInputLayo
ut>
    <LinearLayout
        android:layout_width="match_parent"
        android:gravity="right"

android:layout_marginBottom="@dimen/OptionButtonsBott
omMargin"
        android:layout_height="wrap_content">
        <TextView
            android:id="@+id/DialogCancelButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:paddingVertical="@dimen/OptionPaddingVertical
"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"
            android:textColor="@color/cyan"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"

android:background="@drawable/dialog_text_view_button
_background_ripple"

android:textSize="@dimen/DialogOptionText"
            android:text="Cancel"/>
```

```
            <TextView
                android:id="@+id/DialogSaveButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="@color/cyan"

android:paddingVertical="@dimen/OptionPaddingVertical
"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"

android:background="@drawable/dialog_text_view_button
_background_ripple"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"

android:textSize="@dimen/DialogOptionText"
                android:text="Save"/>
        </LinearLayout>
</LinearLayout>
```

23. Main_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="@color/dark"
    android:layout_height="match_parent"
    tools:context=".MainMenu">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/MainMenuToolbar"

style="@style/Theme.MaterialComponents.DayNight"
            android:layout_width="match_parent"

android:layout_height="?android:attr/actionBarSize"/>
    </LinearLayout>
    <TextView
        android:id="@+id/StartIntervalTraining"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:background="@drawable/text_view_button_backgr
ound_ripple"
            android:paddingHorizontal="20dp"
            android:paddingVertical="12dp"
            android:text="Interval Training"
            android:textColor="@color/textColor"
            android:textSize="@dimen/TextViewButtonText"

app:layout_constraintBottom_toTopOf="@id/StartStopwat
ch"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/StartStopwatch"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:background="@drawable/text_view_button_backgr
ound_ripple"
            android:paddingHorizontal="20dp"
            android:paddingVertical="12dp"
            android:text="Loop Training"
            android:textColor="@color/textColor"
            android:textSize="@dimen/TextViewButtonText"

app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/StartInterva
lTraining" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

24. Quit_app_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    android:orientation="vertical"
android:layout_width="@dimen/DialogWidth"
    android:background="@color/dark"
    android:layout_height="wrap_content">
    <TextView
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="5dp"

android:layout_marginTop="@dimen/TitleTopMargin"

android:layout_marginStart="@dimen/MarginHorizontalDi
alog"
            android:textColor="@color/textColor"
            android:textSize="@dimen/DialogTitleText"
            android:text="@string/QuitTitle"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:layout_marginHorizontal="@dimen/MarginHorizon
talDialog"
            android:layout_marginBottom="10dp"
            android:textColor="@color/textColorGray"
            android:textSize="@dimen/DialogNormalText"
            android:text="Do you really want to quit the
app?"/>
        <LinearLayout
            android:layout_width="match_parent"
            android:orientation="horizontal"

android:layout_marginVertical="@dimen/OptionButtonsBo
ttomMargin"
            android:gravity="right"
            android:layout_height="wrap_content">
            <TextView
                android:id="@+id/NoQuitAppButton"
                android:layout_width="wrap_content"

android:paddingVertical="@dimen/OptionPaddingVertical
"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"
                android:layout_height="wrap_content"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"

android:background="@drawable/dialog_text_view_button
_background_ripple"
                android:textColor="@color/cyan"

android:textSize="@dimen/DialogOptionText"
```

```
            android:text="No"/>
        <TextView
            android:id="@+id/YesQuitAppButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:paddingVertical="@dimen/OptionPaddingVertical
"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"

android:background="@drawable/dialog_text_view_button
_background_ripple"
            android:textColor="@color/cyan"

android:textSize="@dimen/DialogOptionText"
            android:text="Yes"/>
    </LinearLayout>
</LinearLayout>
```

25. Quit_training_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    android:orientation="vertical"
android:layout_width="@dimen/DialogWidth"
    android:background="@color/dark"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

android:layout_marginTop="@dimen/TitleTopMargin"

android:layout_marginHorizontal="@dimen/MarginHorizon
talDialog"
        android:textColor="@color/textColor"
        android:textSize="@dimen/DialogTitleText"
        android:text="@string/QuitTitle"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
android:layout_marginHorizontal="@dimen/MarginHorizon
talDialog"
        android:textColor="@color/textColorGray"
        android:textSize="@dimen/DialogNormalText"
        android:text="Do you really want to quit the
training?"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:orientation="horizontal"

android:layout_marginVertical="@dimen/OptionButtonsBo
ttomMargin"
        android:gravity="right"
        android:layout_height="wrap_content">
        <TextView
            android:id="@+id/NoQuitButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"

android:background="@drawable/dialog_text_view_button
_background_ripple"
            android:textColor="@color/cyan"

android:paddingVertical="@dimen/OptionPaddingVertical
"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"

android:textSize="@dimen/DialogOptionText"
            android:text="No"/>
        <TextView
            android:id="@+id/YesQuitButton"

android:paddingVertical="@dimen/OptionPaddingVertical
"

android:paddingHorizontal="@dimen/OptionPaddingHorizo
ntal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

android:layout_marginEnd="@dimen/MarginBetweenOptions
"

android:background="@drawable/dialog_text_view_button
```

```
_background_ripple"
            android:textColor="@color/cyan"

android:textSize="@dimen/DialogOptionText"
            android:text="Yes"/>
    </LinearLayout>
</LinearLayout>
```

26. Stopwatch_lap_recycle_view_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/and
roid" android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="3dp"
    android:background="@color/dark">
    <TextView
        android:id="@+id/IdOfLap"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="01"
        android:layout_marginStart="20dp"
        android:textColor="@color/textColor"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:orientation="horizontal"
        android:gravity="center"
        android:layout_marginEnd="40dp"
        android:layout_height="wrap_content">
        <TextView
            android:id="@+id/LapTime"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="0:01:01"
            android:layout_marginEnd="70dp"
            android:textColor="@color/textColor"/>

        <TextView
            android:id="@+id/SplitTime"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="0:01:01"
            android:textColor="@color/textColor"/>
    </LinearLayout>

</LinearLayout>
```

27. Stopwatch_main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    xmlns:app="http://schemas.android.com/apk/res-
auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="@color/dark"
    android:orientation="vertical"
    android:layout_height="match_parent"
    tools:context=".Stopwatch">


    <FrameLayout
        android:id="@+id/StopwatchFrame"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_gravity="center"

app:layout_constraintBottom_toTopOf="@+id/StopwatchRe
cycleView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <ProgressBar
            android:id="@+id/StopWatchProgressBar"

style="@style/Widget.AppCompat.ProgressBar.Horizontal
"
            android:layout_width="300dp"
            android:layout_height="300dp"

android:background="@drawable/time_progress_bar_backg
round"
            android:max="12"
            android:progress="12"

android:progressDrawable="@drawable/timer_progress_ba
r"
            android:rotation="-90"
            android:rotationY="180" />

        <ProgressBar

android:id="@+id/StopWatchPausedProgressBar"
```

```
style="@style/Widget.AppCompat.ProgressBar.Horizontal
"
            android:layout_width="300dp"
            android:layout_height="300dp"

android:background="@drawable/time_progress_bar_backg
round"
            android:max="12"
            android:progress="12"

android:progressDrawable="@drawable/timer_progress_ba
r_paused"
            android:rotation="-90"
            android:rotationY="180"
            android:visibility="gone" />

        <TextView
            android:id="@+id/StopwatchTimerText"
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:layout_gravity="center"

android:background="@drawable/text_view_button_no_bac
kground_ripple"
            android:gravity="center"
            android:text="Start"
            android:textColor="@color/textColor"
            android:textSize="35sp" />
    </FrameLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"

app:layout_constraintBottom_toTopOf="@+id/StopwatchRe
cycleView"

app:layout_constraintTop_toBottomOf="@+id/StopwatchFr
ame">

        <TextView
            android:layout_marginEnd="55dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="@color/textColor"
            android:text="Lap Time" />
```

```
            <TextView
                android:textColor="@color/textColor"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Split Time" />
        </LinearLayout>
        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/StopwatchRecycleView"
            android:layout_width="match_parent"
            android:layout_height="300dp"
            android:layout_marginHorizontal="30dp"
            android:layout_marginVertical="20dp"

app:layout_constraintBottom_toTopOf="@+id/mainLL"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/StopwatchFra
me" />

        <LinearLayout
            android:id="@+id/mainLL"
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:layout_gravity="top"
            android:gravity="bottom|center"
            android:orientation="horizontal"

app:layout_constraintBottom_toBottomOf="parent">

            <TextView
                android:visibility="gone"
                android:id="@+id/LapButton"
                android:layout_width="130dp"
                android:layout_height="wrap_content"
                android:layout_gravity="right|bottom"
                android:layout_marginEnd="20dp"
                android:layout_marginBottom="10dp"

android:background="@drawable/text_view_button_backgr
ound_ripple"
                android:clickable="false"
                android:gravity="center"
                android:paddingHorizontal="20dp"
                android:paddingVertical="10dp"
                android:text="Lap"
                android:textColor="@color/textColor"

android:textSize="@dimen/TextViewButtonText" />
```

```
            <TextView
                android:id="@+id/PauseButton"
                android:layout_width="130dp"
                android:layout_height="wrap_content"
                android:layout_gravity="right|bottom"
                android:layout_marginBottom="10dp"

android:background="@drawable/text_view_button_backgr
ound_ripple"
                android:clickable="false"
                android:gravity="center"
                android:paddingHorizontal="20dp"
                android:paddingVertical="10dp"
                android:text="Start"
                android:textColor="@color/textColor"

android:textSize="@dimen/TextViewButtonText" />
        </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

28. Dialog_text_view_button_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/an
droid" android:shape="rectangle">
<solid android:color="@color/dark"/>
</shape>
```

29. Dialog_text_view_button_background_ripple.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ripple
xmlns:android="http://schemas.android.com/apk/res/an
droid" android:color="@color/textColor">
    <item
android:drawable="@drawable/dialog_text_view_button_
background"/>

</ripple>
```

30. Recycle_view_item_selected.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    android:shape="rectangle"
```

```
xmlns:android="http://schemas.android.com/apk/res/an
droid">
    <stroke android:width="2dp"
android:color="@color/cyan"/>
    <solid android:color="@color/bgSecondary"/>
    <corners
android:radius="@dimen/RecycleViewItemCornerRadius"/
>
</shape>
```

31. Splash_screen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list
    android:opacity="opaque"

xmlns:android="http://schemas.android.com/apk/res/an
droid">
<item android:drawable="@color/red"/>

</layer-list>
```

32. Text_view_button_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    android:shape="rectangle"

xmlns:android="http://schemas.android.com/apk/res/an
droid">
<solid android:color="@color/cyan"/>
    <corners android:radius="30dp"/>
</shape>
```

33. Text_view_button_background_ripple.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ripple
xmlns:android="http://schemas.android.com/apk/res/an
droid" android:color="@color/textColor">
<item
android:drawable="@drawable/text_view_button_backgro
und"/>
</ripple>
```

34. Text_view_button_disabled.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    android:shape="rectangle"

xmlns:android="http://schemas.android.com/apk/res/and
roid">
<solid android:color="@color/disabled"/>
    <corners android:radius="30dp"/>
</shape>
```

35. Text_view_button_no_background_ripple.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ripple
xmlns:android="http://schemas.android.com/apk/res/an
droid" android:color="@color/noBackgroundRipple">

</ripple>
```

36. Time_progress_bar_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/an
droid"
    android:shape="ring"
android:innerRadiusRatio="2.7"
android:thicknessRatio="50.0"
android:useLevel="false">
    <solid android:color="@color/bgSecondary"/>
</shape>
```

37. Timer_progress_bar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/an
droid"
    android:shape="ring"
    android:innerRadiusRatio="2.7"
    android:thicknessRatio="50.0"
    android:useLevel="true" >
        <solid android:color="@color/cyan"/>
</shape>
```

38. Timer_progress_bar_paused.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
```

```
xmlns:android="http://schemas.android.com/apk/res/an
droid"
    android:shape="ring"
android:innerRadiusRatio="2.7"
android:thicknessRatio="50.0"
android:useLevel="true">
        <solid android:color="@color/red"/>
</shape>
```

39. Color

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>

    <color name="noBackgroundRipple">#858585</color>
    <color name="textColor">#FFFFFF</color>
    <color name="textColorSecondary">#205B51</color>
    <color name="textColorGray">#B5B5B5</color>
    <color name="bgSecondary">#1B1E22</color>
    <color name="dark">#24282E</color>
    <color name="cyan">#18B192</color>
    <color name="red">#E8321E</color>
    <color name="disabled">#1B1E22</color>
</resources>
```

40. Build.grandle(module:RunningApp.app)

```
plugins {
    id 'com.android.application'
    id 'kotlin-android'
    id 'kotlin-android-extensions'
}

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    defaultConfig {
        applicationId "com.nicknterm.runningapp"
        minSdkVersion 22
        targetSdkVersion 30
        versionCode 1
```

```
        versionName "1.4"

        testInstrumentationRunner
"androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles
getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
}

dependencies {

    implementation "org.jetbrains.kotlin:kotlin-
stdlib:$kotlin_version"
    implementation 'androidx.core:core-ktx:1.3.2'
    implementation
'androidx.appcompat:appcompat:1.2.0'
    implementation
'com.google.android.material:material:1.3.0'
    implementation
'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'jp.wasabeef:recyclerview-
animators:3.0.0'
    implementation 'androidx.legacy:legacy-support-
v4:1.0.0'

    testImplementation 'junit:junit:4.+'
    androidTestImplementation
'androidx.test.ext:junit:1.1.2'
    androidTestImplementation
'androidx.test.espresso:espresso-core:3.3.0'
}
```

41. Menu_nav_bar.xml

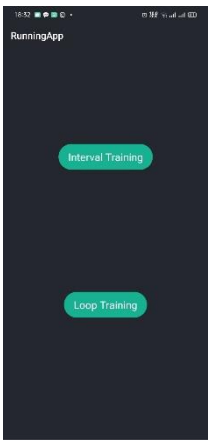
```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/and
roid">
<group>
    <item
        android:id="@+id/SaveButton"
        android:title="Save"/>
    <item
        android:id="@+id/LoadButton"
        android:title="Load"/>
</group>
</menu>
```

42. Strigs.xml

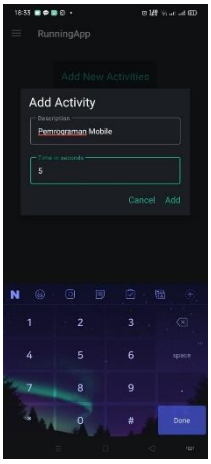
```
<resources>
     <string name="app_name">RunningApp</string>
     <string name="nav_open">Open Nav</string>
     <string name="close_nav">Close Nav</string>
     <string name="QuitTitle">Quit?</string>
     <string name="notificationButton">Add</string>

     <dimen name="DialogTitleText">26sp</dimen>
     <dimen name="DialogNormalText">18sp</dimen>
     <dimen name="DialogOptionText">18sp</dimen>
     <dimen name="TextViewButtonText">22sp</dimen>
     <dimen
 name="OptionButtonsBottomMargin">5dp</dimen>
     <dimen name="TitleTopMargin">10dp</dimen>
     <dimen name="DialogWidth">300dp</dimen>
     <dimen name="MarginBetweenOptions">10dp</dimen>
     <dimen
 name="OptionPaddingHorizontal">10dp</dimen>
     <dimen name="OptionPaddingVertical">5dp</dimen>
     <dimen
 name="MarginHorizontalDialog">17dp</dimen>
     <dimen
 name="RecycleViewItemCornerRadius">10dp</dimen>
 </resources>
```

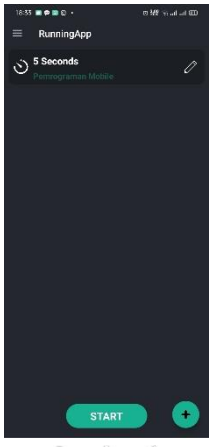
3.4 Testing



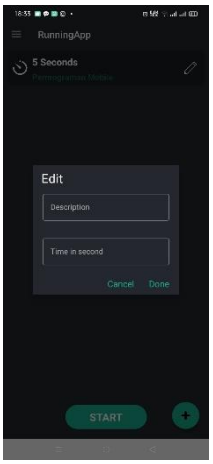
Gambar 1.6 Tampilan aplikasi setelah di run



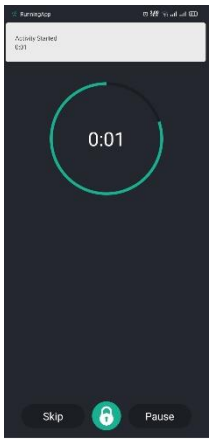
Gambar 1.7 Tampilan aplikasi apabila mengklik button *interval training*



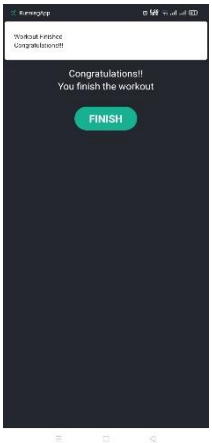
Gambar 1.8 Tampilan aplikasi apabila mengklik add



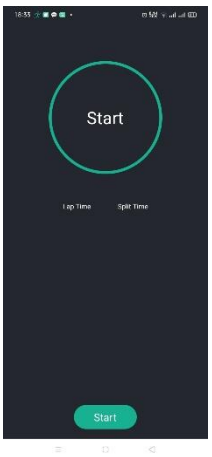
Gambar 1.9 Tampilan aplikasi apabila mengklik *edit*



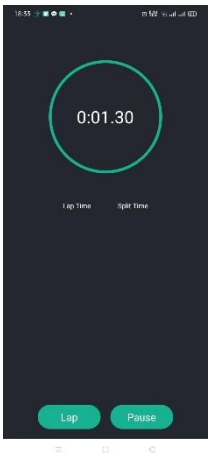
Gambar 1.10 Tampilan aplikasi apabila mengklik button *start*



Gambar 1.11 Tampilan aplikasi apabila waktu aktivitas berakhir



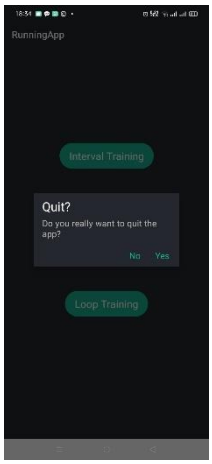
Gambar 1.12 Tampilan aplikasi apabila mengklik button *loop training*



Gambar 1.13 Tampilan aplikasi apabila mengklik button *start*

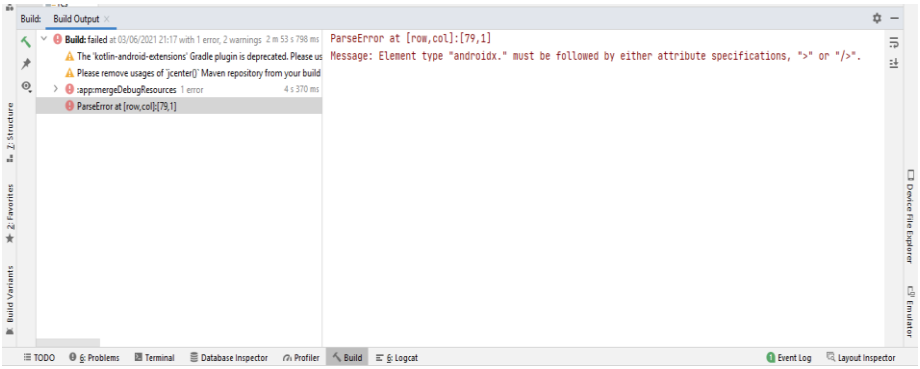
Gambar 1.14 Tampilan aplikasi apabila mengklik button lap

Gambar 1.15 Tampilan aplikasi apabila mengklik button *pause*



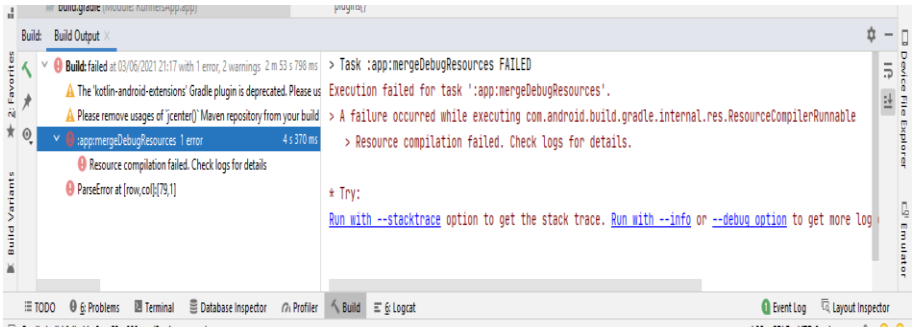
Gambar 1.16 Tampilan dialog aplikasi

3.5 Debugging



Gambar 1.17 Tampilan *parse error* pada android studio

Pada **Gambar 1.17** terjadi *troubleshot* dimana *programmer* tidak menambahkan penutup pada akhir coding yang dimasukkan. Penyelesaian yang dilakukan adalah dengan menambahkan penutup, yaitu “ > ” pada akhir coding.



Gambar 1.18 Tampilan *app:mergeDebugResouce* pada android studio

Pada **Gambar 1.18** terjadi *troubleshot* dimana *programmer* tidah dapat menghubungkan *resource layout* sehingga, penyelesaian yang dapat dilakukan adalah dengan menghapus .idea folder dan .gradle folder. Setelah itu menutup aplikasi android studio dan membuka aplikasi sebagai “*Open an existing android studio projec*t” dan setelah itu “*Refresh all Gradle projects*”.

BAB IV
PENUTUP

4.1 Kesimpulan

Pada saat akan melakukan perancangan aplikasi *programmer* harus memperhatikan perancangan sistem dan UI/UX agar aplikasi dapat berjalan dengan baik dan menarik pengguna dan *programmer* harus siap dalam menghadapi *troubleshooting* dan mencari jalan keluar dari permasalahan yang dihadapi.

4.2 Saran

Sebaiknya, untuk ke depannya dalam menjalankan praktikum *mobile programming* mahasiswa diberikan pendampingan dalam menjalankan praktikum. Setiap pertemuan, diberikan evaluasi mengenai praktikum yang telah dilakukan dan mahasiswa diberikan *deadline* dalam mengerjakan laporan modul praktikum setiap minggunya.

REFERENSI

https://www.youtube.com/watch?v=by3J1Gs60m0

https://www.youtube.com/playlist?list=PLQkwcJG4YTCQ6emtoqSZS2FVwZR9FT3BV

https://www.youtube.com/watch?v=wiW_LiAWwfY&t=43s

https://www.youtube.com/watch?v=vmIwntEn8WY

https://www.youtube.com/watch?v=T_wSEnqGPdo

https://github.com/vitaviva/CountdownTimer

https://github.com/PaulShan101/ListActivity

https://github.com/damu3024/ListActivity