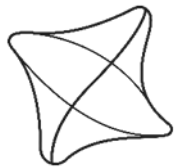


ŽILINSKÁ UNIVERZITA V ŽILINE  
FAKULTA RIADENIA A INFORMATIKY



ŽILINSKÁ UNIVERZITA V ŽILINE  
Fakulta riadenia  
a informatiky

# FITNESS TRACKER

Študijný program: Informatika a Riadenie

Autor: Samuel Billý

Školský rok: 2022/2023

Skupina: 5ZYR22

Navštevované cvičenia: PIA Blok: 9,10.

## Obsah

# 1 Obsah

2	Prehľad dostupných aplikácií podobného zamerania .....	4
2.1	My Workout Plan .....	4
2.2	Muscle Booster Workout Planner.....	4
2.3	Fitbod.....	4
3	Analýza navrhovanej aplikácie .....	4
3.1	Tvorba prípadov použitia.....	4
4	Návrh architektúry aplikácia .....	5
4.1	Popis aplikácie .....	5
4.2	Návrh procesov .....	5
4.3	Cieľová skupina .....	5
5	Návrh riešenia problému .....	6
5.1	Krátka analýza .....	6
	.....	7
	.....	7
5.2	Návrh aplikácie .....	8
	.....	8
5.3	Implementácia .....	9
5.4	Aktivity a Fragменты .....	9
5.4.1	Main Activity .....	9
5.4.2	Home .....	9
5.4.3	StartPlans.....	9
5.4.4	PlanAdapter .....	9
5.4.5	ExerciseDetailsFragment .....	9
5.4.6	AfterWorkout .....	10
5.4.7	PlansShow .....	10
5.4.8	ExerciseListFragment.....	10
5.4.9	ExerciseAdapter .....	10
5.4.10	Profile .....	10
5.4.11	StatsProfile .....	10
5.4.12	StatsExerciseAdapter .....	10
5.4.13	DetailsExercise .....	11
5.4.14	ProfileSettings .....	11

5.4.15	NotificationUtil .....	11
5.5	Navigation .....	11
5.6	LifeCycles.....	12
5.7	LiveData .....	12
5.8	Room .....	12
5.9	ViewModel .....	13
5.10	WorkManager.....	13
5.11	ViewBinding.....	14
5.12	Broadcast receiver .....	14
5.13	RecyclerView .....	14
5.14	Použitie externého frameworku / knižnice .....	14
5.14.1	Glide .....	14
<b>5.14.2</b>	<b>com.github.mikephil.charting.charts.LineChart .....</b>	<b>15</b>
5.15	Notifikácie .....	15

## **2 Prehľad dostupných aplikácií podobného zamerania**

### **2.1 My Workout Plan**

Aplikácia My Workout plan je aplikácia veľmi podobná našej. V bezplatnej verzii je v nej možné zostaviť si plán a tiež aj zapisovať svoje údaje o tele ale aj o cvičení. V platenej verzii v cene 3,19E za mesiac je možné vytvoriť si viacero svojich plánov na cvičenie alebo pridať si cvik behom cvičenia. V Platenej verzii je tiež možné si sledovať svoje štatistiky o cvičení či už trvanie cvičenia alebo ako sme zapájali svaly.

### **2.2 Muscle Booster Workout Planner**

V tejto aplikácii sa po stiahnutí a prihlásení vyskočí dialógové okno, kde zadáme informácie o nás a tiež plán nášho cvičenia nám automaticky vygeneruje náš plán. Aplikácia nám približne spočíta spálené kalórie behom cvičenia alebo tiež pri každom cviku zobrazí ako ho správne vykonať. Aplikácia samotná neslúži primárne na zaznamenanie tréningu ale na pomoc pri cvičení hlavne pre začiatočníkov. Cena tejto aplikácie je 9,99 E mesačne pričom pri ročnom predplatnom je cena za mesiac 3,25 eur.

### **2.3 Fitbod**

Fitbod je aplikácia, ktorá nám po zadaní našich údajov, našich cieľov a cvičebných prostriedkov, ktoré máme k dispozícii zostaví plán pre rôzne svalové partie alebo pre svalové partie ktoré dlho necvičili. Výhodou tejto aplikácie je že dokáže zobrazovať využitie svalov pri daných cvikoch či takzvanú 'Heat map' čiže zobrazenie svalov na tele a ich predpokladané poškodenie po posledných tréningoch. Nevýhodou tejto aplikácie je že nedokáže prehľadne zobraziť pokrok v cvičení a tiež že v trial verzii je možné odcvičiť len 2 tréningy pričom aplikácia si mesačné účtuje 7,08 eur. Nevýhodou tiež je že výber tréningového plánu je neprehľadný a zaznamenať odcvičené váhy a počty je opakovaní je možné až po dokončení celého tréningu, čo pri dlhších tréningov je veľmi zložité zapamätať si.

## **3 Analýza navrhovanej aplikácie**

### **3.1 Tvorba prípadov použitia**

Osobný tréningový plán: Aplikácia umožňuje používateľom vytvoriť si vlastný tréningový plán, prispôsobený ich cieľom a kondičnému stavu. Používatelia si môžu zvoliť cviky, ktoré chcú absolvovať, a nastaviť si počet opakovaní, súprav a váhu. Po každom tréningu môžu používatelia označiť, ktoré cviky absolvovali a zaznamenajú si, akú váhu použili.

Denný záznam tréningu: Aplikácia umožňuje používateľom zaznamenávať svoj denný tréning. Používatelia si môžu zvoliť cviky, ktoré absolvovali, a zaznamenať váhu a počet opakovaní. Tieto údaje sa potom ukladajú do denníka tréningu.

Sledovanie pokroku: Aplikácia umožňuje používateľom sledovať ich pokrok. Používatelia môžu vidieť, koľko cvičení a sérií absolvovali, akú váhu použili a koľko opakovaní zvládli. Tieto údaje sa zobrazujú v grafoch.

Záznamy o tele: Aplikácia umožňuje zaznamenať údaje o hmotnosti a tiež možnosť zaznamenať veľkosť určitých svalov .

## **4 Návrh architektúry aplikácia**

### **4.1 Popis aplikácie**

Mobilná aplikácia umožňuje užívateľovi zostaviť si tréningový plán z cvikov a zaznamenávať váhy odcvičených. Užívateľ má možnosť vybrať si z databázy cvikov, ktoré aplikácia poskytuje. Pre každý cvik je možné definovať počet opakovaní, váhu, dátum a čas tréningu. Aplikácia tiež umožňuje užívateľovi sledovať pokrok a zhodnocovať výsledky.

### **4.2 Návrh procesov**

Správa tréningových plánov: Tento proces zahŕňa vytvorenie a úpravu tréningových plánov. Užívateľ môže vytvoriť nový tréningový plán, upravovať existujúci plán alebo mazať plán, ktorý už nepoužíva. Pre každý plán si užívateľ môže vybrať cviky, ktoré chce zahrnúť, definovať počet opakovaní a váhu.

Zaznamenávanie tréningov: Tento proces zahŕňa zaznamenávanie tréningov a ukladanie dát do databázy. Užívateľ si môže zvoliť konkrétny tréningový plán a cviky, ktoré chce v danom tréningu absolvovať. Počas tréningu môže zaznamenávať počet opakovaní a váhu pre každý cvik. Aplikácia potom uloží tieto údaje do databázy pre ďalšiu analýzu a zhodnotenie pokroku.

Sledovanie pokroku: Aplikácia umožňuje používateľom sledovať ich pokrok. Používatelia môžu vidieť, koľko cvičení a sérií absolvovali, akú váhu použili a koľko opakovaní zvládli. Tieto údaje sa zobrazujú v grafoch a tabuľkách.

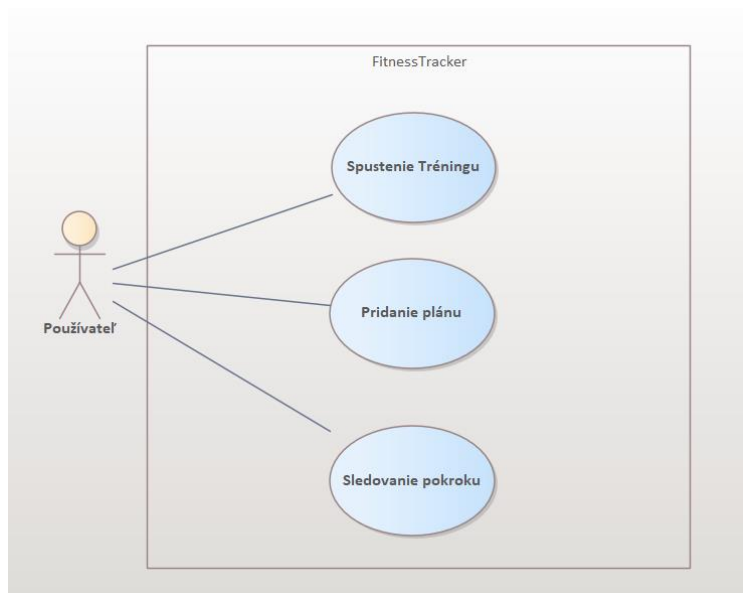
### **4.3 Cieľová skupina**

Cieľová skupina našej aplikácie sú používatelia, ktorí vedľa ako a čo cvičiť a potrebujú si len zapisovať údaje aby mohli sledovať ich pokrok.

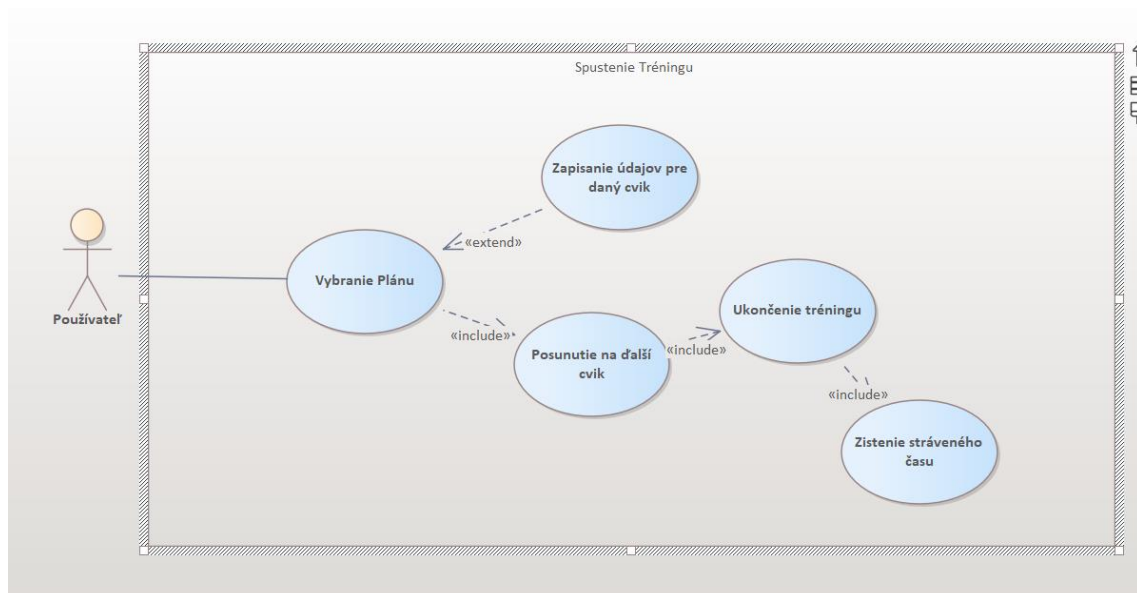
## 5 Návrh riešenia problému

### 5.1 Krátka analýza

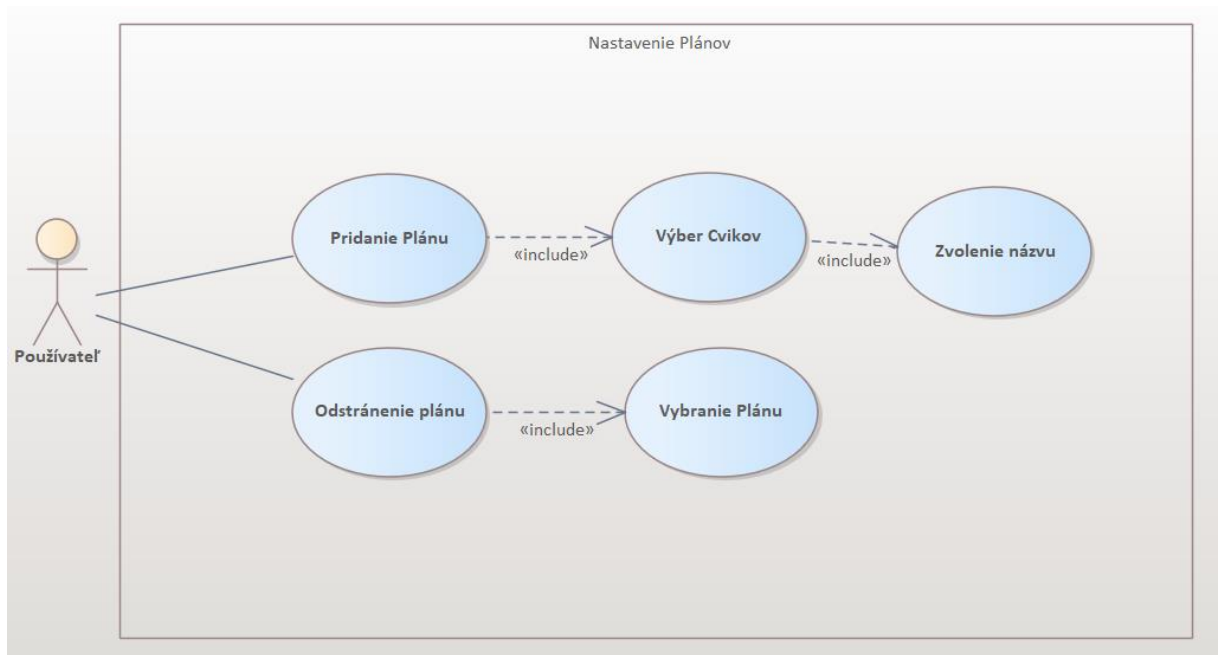
Používateľ si môže zvoliť medzi spustením tréningu, úpravou plánov alebo kontrolou svojích údajov.



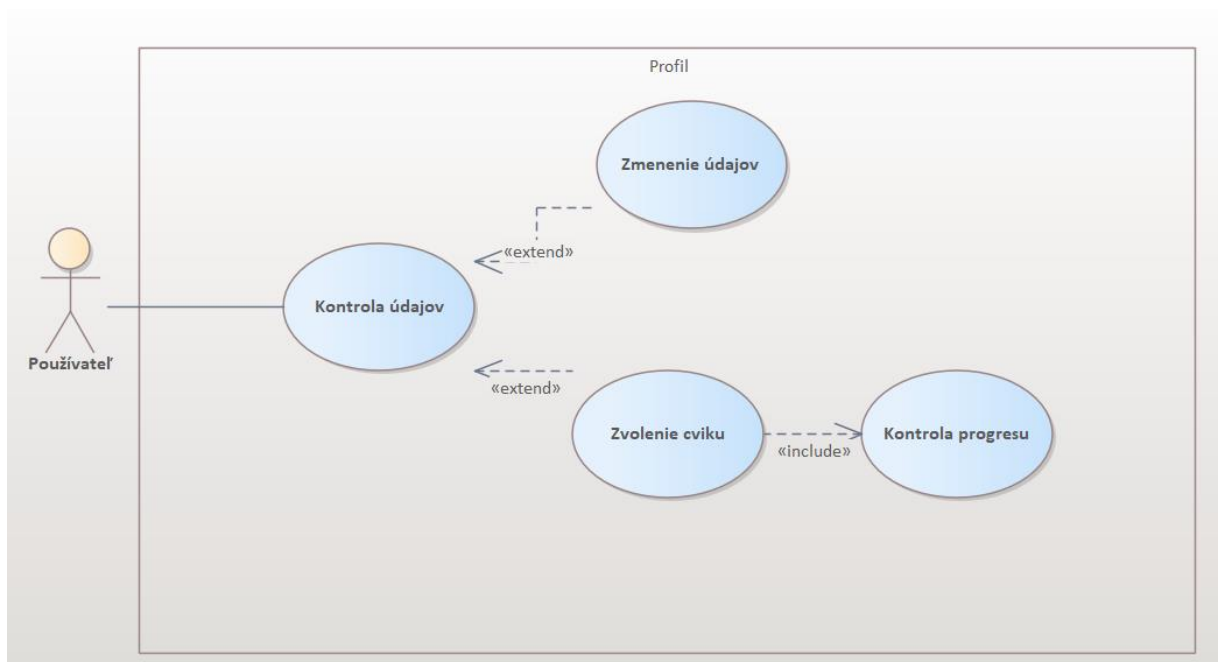
Používateľ si môže ale nemusí zaznamenať údaje pre daný cvik predtým ako prejde k ďalšiemu cviku.



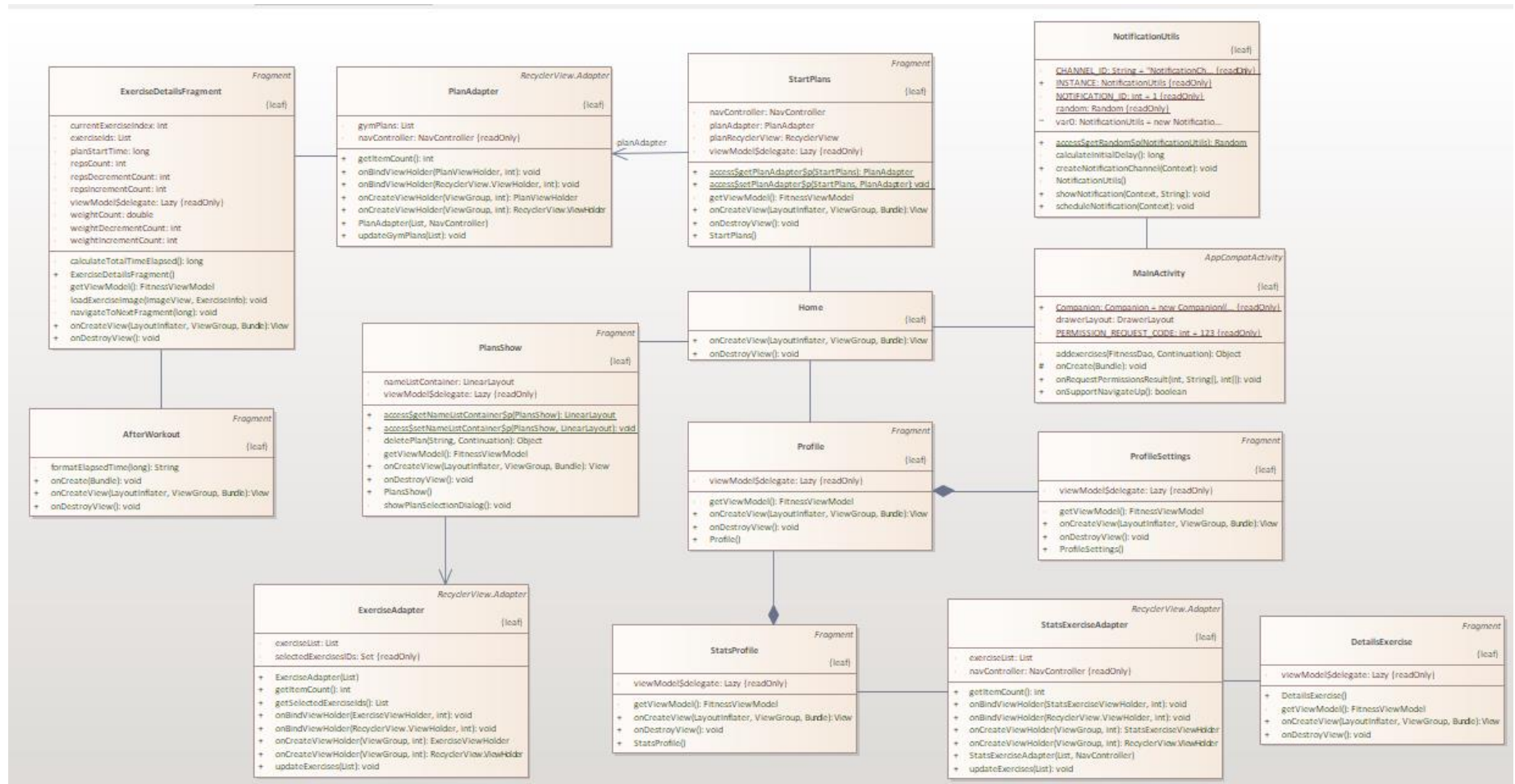
Používateľ môže pridať plán alebo odstrániť plán z existujúcich plánov. Pri pridávaní plánu môže zvoliť ľubovoľný počet cvikov. A následne musí zadať názov plánu.



Používateľ v sekcii profil si môže zmeniť svoju váhu alebo výšku. A tiež si môže zobrazit' graf pre cvik, ktorý si zvolí.



## 5.2 Návrh aplikácie





## 5.3 Implementácia

V tejto časti si popíšeme komponenty využité v semestrálnej práci

### 5.4 Aktivity a Fragmenty

V našej aplikácii využívame 13 fragmentov a 1 aktivitu.

#### 5.4.1 Main Activity

V aktivite si skontrolujeme, či databáza v našej aplikácii obsahuje cviky, ak nie tak si ich pridáme do databázy. Následne zistíme či má naša aplikácia povolenie na zobrazovanie notifikácií a ak áno tak pomocou triedy NotificationsUtil vytvoríme notifikáciu a naplánujeme ju.

Tiež si vytvoríme action bar, ktorý nám pomôže sa navigovať v aplikácii a tiež NavController, pomocou ktorého sa budeme v aplikácii navigovať.

#### 5.4.2 Home

V tomto fragmente máme 3 tlačidlá, ktoré majú tieto funkčnosti. Tlačidlo Start slúži na spúšťanie tréningu, Tlačidlo Plan slúži na vytváranie nových plánov a na vymazanie existujúcich. A tlačidlo Profile slúži na zobrazovanie informácií o používateľovi.

#### 5.4.3 StartPlans

Fragment obsahuje zoznam všetkých vytvorených plánov, ktoré sú načítané z databázy a vložené do PlanAdaptéra.

#### 5.4.4 PlanAdapter

Adaptér vypíše všetky plány ,pričom pomocou OnClickListenera zaznamenáva kliknutie na názov. Ak používateľ klikne na nejaký plán presunie ho to do fragmentu ExerciseDetailsFragment pričom pošle ako paramater názov plánu.

#### 5.4.5 ExerciseDetailsFragment

Tento fragment na základe názvu plánu prejde všetky cviky, ktoré sa v danom pláne nachádzajú. Každý cvik je zobrazený na obrázku. K dispozícii pre používateľa su 2 textové polia na zadávanie informácií. Tiež sa tam nachádzajú 2 tlačidlá pre nastavenie počtu opakovaní a 2 tlačidlá pre nastavenie váhy. Tlačidlá slúžia na jednoduchšie zmeny v hodnotách v prípade ďalších sérií. Pričom počet opakovaní sa mení o 1 a váha sa mení o 2,5 kg. V tomto fragmente sú 2 ďalšie tlačidlá. Pomocou tlačidla Confirm potvrdíme naše zadané informácie a pomocou

tlačidla Next sa presunieme na ďalší cvik. V prípade že už sa ďalší cvik v danom pláne nenachádza presunieme sa do fragmentu AfterWorkout.

### **5.4.6 AfterWorkout**

Fragment prijíma ako parameter čas tréningu vo formáte long a následne ho zmení do formátu HH:MM:SS a zobrazí na obrázovku. Obsahuje tlačidlo Done, pomocou ktorého sa dostaneme na domovskú obrázovku aplikácie, čiže do fragmentu Home.

### **5.4.7 PlansShow**

V tomto fragmente sa zobrazia všetky vytvorené plány a tiež 2 tlačidlá. Tlačidlo DELETE, slúži na odstránenie existujúceho plánu. Tlačidlo ADD nás presunie do fragmentu ExerciseListFragment.

### **5.4.8 ExerciseListFragment**

Fragment pomocou triedy ExerciseAdapter zobrazí všetky cviky z našej databázy. Vo fragmente sa nachádza tlačidlo Add, ktoré všetky označené cviky vloží do nového plánu, ktorého názov zadame.

### **5.4.9 ExerciseAdapter**

Tento adapter zobrazí všetky cviky, pričom každý jeho prvok obsahuje obrázok, názov a checkbox . Obsahuje v sebe metódu getSelectedExerciseIds, ktorá vracia list s číslami cvikov.

### **5.4.10 Profile**

Fragment zobrazuje informácie o používateľovi ako sú výška, váha a jeho BMI index. Vo fragmente sa tiež nachádza tlačidlo ,ktoré nás presunie do StatsProfile fragmentu, kde sú zobrazené všetky cviky. Hodnoty výška a váha slúžia tiež ako tlačidlá. Pokiaľ ich užívateľ stlačí presunie ho to do fragmentu ProfileSettings, kde si môže nastaviť svoje údaje.

### **5.4.11 StatsProfile**

V tomto fragmente používame StatsExerciseAdapter, ktorý nám zobrazí všetky cviky.

### **5.4.12 StatsExerciseAdapter**

Adapter, ktorý zobrazuje všetky cviky databázy a pri stlačení nejakého cviku, pošle ako parameter číslo cviku do fragmentu DetailsExercise.

### 5.4.13 Details Exercise

Tento fragment zobrazí graf s hodnotami odcvičených váh pre cvik, ktorého ID prijme ako parameter. Graf zobrazujeme z knižnice `mikephill.charting.charts`, ktorá je dostupná na githube. V našom prípade používa `LineChart` graf. Hodnoty si beriem z databázy.

### 5.4.14 Profile Settings

V tomto fragmente si užívateľ môže zadať svoju výšku a váhu, pričom tieto údaje si zapíšeme do databázy.

### 5.4.15 Notification Util

Tento fragment obsahuje tieto metódy. Metóda `createNotificationChannel(context: Context)` vytvára kanál oznámení pre zobrazenie oznámení v Android aplikácii. Vytvára sa iba v prípade, že zariadenie používa Android verzie 8.0 (API úroveň 26). Metóda `scheduleNotification(context: Context)`: Táto metóda je zodpovedná za naplánovanie oznámenia na zobrazenie v neskoršom čase. Využíva `WorkManager` na naplánovanie jednorazovej práce (`OneTimeWorkRequest`), ktorá je vykonaná v určenom čase. Metóda `calculateInitialDelay(): Long`: Táto metóda je zodpovedná za výpočet časového oneskorenia (v milisekundách) pred zobrazením oznámenia. Určuje sa pomocou `Calendar` a nastavuje sa požadovaný čas zobrazenia oznámenia.

Trieda `NotificationReceiver`: Táto trieda je `BroadcastReceiver`, ktorý môže byť použitý na spracovanie udalostí oznámení spúšťaných systémom alebo inými komponentami. V metóde `onReceive()` sa získa náhodný index a vyberie sa náhodný text oznámenia. Následne sa zavolá metóda `showNotification()` na zobrazenie oznámenia.

## 5.5 Navigation

Navigation slúži na navigovanie medzi fragmentami pričom funguje na základe `OnClickListenerov` používateľa

```
homeBinding.profilebutton.setOnClickListener { it: View!
    findNavController().navigate(R.id.action_home2_to_profile)
}
homeBinding.planButton.setOnClickListener { it: View!
    findNavController().navigate(R.id.action_home2_to_plansShow)
}
homeBinding.playButton.setOnClickListener { it: View!
    findNavController().navigate(R.id.action_home2_to_startPlans)
}
```

## 5.6 LifeCycles

Lifecycle v Androidu je súbor metód a callbackov, ktoré poskytujú informácie o stavoch a udalostiach, ktoré sa týkajú životného cyklu komponentov aplikácie, ako sú aktivity a fragmenty.

V našej aplikácii využívame LifecycleScope, ktorá je súčasťou knižnice Android Jetpack. LifecycleScope.launch, je asynchrónna operácia vykonáva sa v IO vlákne (Dispatchers.IO). Keď sa komponent dostane do stavu, v ktorom už nie je potrebné spúšťať túto operáciu, napríklad ak sa aktivita zruší alebo fragment zničí, coroutine bude automaticky zrušená a vykonávanie sa zastaví. To pomáha predchádzať únikom pamäte a nežiaducemu vykonávaniu operácií v momente, keď už nie sú potrebné. Používame ho pri práci s databázou, pri vkladaní nových prvkov do databázy.

```
LifecycleScope.launch { this: CoroutineScope
    withContext(Dispatchers.IO){ this: CoroutineScope
        if (dao.getExerciseCount() == 0){
            addexercises(dao) }
        }
    }
```

## 5.7 LiveData

LiveData je súčasťou Android Architecture Components a slúži na reprezentáciu a sledovanie hodnôt, ktoré môžu byť asynchrónne alebo meniť svoj stav v priebehu času. LiveData poskytuje jednoduché rozhranie na zachytávanie týchto zmien a aktualizáciu používateľského rozhrania (UI) na základe nových hodnôt. V našej aplikácii slúži na získavanie informácií z databázy.

```
@Query("SELECT name FROM GymPlan ORDER BY name ASC")
fun getPlanNames():LiveData<List<String>>
```

## 5.8 Room

Room je súčasťou Android Architecture Components a je knižnicou, ktorá umožňuje jednoduchú implementáciu persistencie údajov v aplikácii. Room poskytuje vrstvu abstrakcie nad SQLite databázou a zjednodušuje prácu s databázovými operáciami, ako je vytváranie, čítanie, aktualizácia a mazanie (CRUD).

Naša databáza s názvom fitness\_db obsahuje 4 tabuľky. User na zapisávanie údajov váha a výška o používateľovy pričom jej primárny kľúč je dátum zapísania údajov. GymPlan, ktorého primárny kľúč je jeho názov a jeho ďalší atribút je list Integerov, ktoré sú primárne kľúče

cvikov. ExerciseInfo , kde primárna kľúč je jeho číslo, názov cviku a názov jeho obrázka sú jeho ďalšie atribúty.ExerciseStat má kompozitný primárny kľúč, je to číslo cviku a dátum zadania , tiež ma 2 atribúty reps, ktorý zobrazuje počet opakovaní a tiež weight, pričom to je váha s ktorou bolo cvičené a je vo formáte double.

Databáza ma triggre, ktoré sú v triede FitnessDao.

```
abstract class FitnessDatabase: RoomDatabase() {

    abstract val fitnessDao: FitnessDao

    companion object{
        private var INSTANCE: FitnessDatabase? = null

        fun getInstance(context: Context): FitnessDatabase{
            synchronized( lock: this){
                return INSTANCE?: Room.databaseBuilder(
                    context.applicationContext,
                    FitnessDatabase::class.java,
                    name: "fitness_db"
                )
                    .build().also { it: FitnessDatabase
                        INSTANCE = it
                    }
            }
        }
    }
}
```

## 5.9 ViewModel

ViewModel je súčasťou architektúry Android Jetpack a slúži na správu a uchovávanie dát spojených s UI a stavom aplikácie. Jeho hlavným účelom je oddeliť obchodnú logiku a dáta od vrstvy používateľského rozhrania (UI) a zabezpečiť, že dáta sú dostupné aj pri zmene konfigurácie (napríklad otáčanie obrazovky).

```
class FitnessViewModel(private val dao: FitnessDao): ViewModel() {
    fun addUserInfo(user: User) {
        viewModelScope.launch(Dispatchers.IO) { this: CoroutineScope
            dao.addUserInfo(user)
        }
    }
}
```

## 5.10 WorkManager

WorkManager je súčasťou Android Architecture Components a je knižnicou, ktorá poskytuje plánovanie a správu asynchrónnych úloh v Android aplikáciách. Hlavným cieľom

WorkManageru je zabezpečiť správne vykonávanie úloh bez ohľadu na stav zariadenia alebo životného cyklu aplikácie.

## 5.11 ViewBinding

ViewBinding je technika vývoja v Androidu, ktorá umožňuje väzbu (binding) komponentov užívateľského rozhrania (UI) s príslušnými zdrojmi v rámci XML súborov. Hlavným cieľom ViewBinding je poskytnúť typovú bezpečnosť a jednoduchšiu prácu s komponentmi UI v porovnaní s tradičným prístupom, ktorý využíva volanie metód findViewById().

Používame ho v každom fragmente, ktorý má svoj xml súbor.

```
val homeBinding = FragmentHomeBinding.inflate(layoutInflater)
homeBinding.profilebutton.setOnClickListener { it: View!
    findNavController().navigate(R.id.action_home2_to_profile)
}
homeBinding.planButton.setOnClickListener { it: View!
    findNavController().navigate(R.id.action_home2_to_plansShow)
}
```

## 5.12 Broadcast receiver

## 5.13 RecyclerView

RecyclerView slúži na dynamicke zobrazenie údajov. Funguje spoločne s adapterom. V našej aplikácii ho používame v 3 fragmentoch. V fragmente StartPlans s použitím adaptéra PlanAdapter. V fragmente StatsProfile s použitím adaptéra StatExerciseAdapter. A v fragmente ExerciseList s použitím exerciseAdapter.

## 5.14 Použitie externého frameworku / knižnice

### 5.14.1 Glide

Glide je knižnica pre správu a zobrazenie obrázkov v aplikáciách Android. Glide je špeciálne navrhnutý pre zobrazovanie obrázkov v RecyclerView alebo ListView. Umožňuje automatickú recykláciu a manažment pamäte pre optimalizované zobrazenie obrázkov v zoznamoch s veľkým množstvom prvkov. V našej aplikácii ho používame na zobrazenie obrázkov v Adaptéroch.

```

Glide.with(holder.itemView.context) RequestManager
    .load(
        holder.itemView.context.resources.getIdentifier(
            currentExercise.img,
            defType: "drawable",
            holder.itemView.context.packageName
        )
    ) RequestBuilder<Drawable!>
    .into(holder.exerciseImageView)

```

### 5.14.2 com.github.mikephil.charting.charts.LineChart

Komponent `<com.github.mikephil.charting.charts.LineChart>` je súčasťou knižnice MPAndroidChart, ktorá poskytuje grafické zobrazenie dát pomocou rôznych typov grafov. Konkrétne, LineChart je graf, ktorý zobrazuje dáta pomocou čiar, ktoré spájajú jednotlivé body.

```

<com.github.mikephil.charting.charts.LineChart
    android:id="@+id/chart"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

```

### 5.15 Notifikácie

V aplikácii využívame notifikácie pre tzv. Motivačné účely. Každé ráno o 8:00 príde používateľovy notifikácia s motivačným citátom. Zoznam citátov sa nachádza v súbore NotifikationText.kt.

```

class NotificationWorker(context: Context, params: WorkerParameters) : Worker(context, params) {
    override fun doWork(): Result {
        val applicationContext = applicationContext
        val randomIndex = Random().nextInt(notificationTexts.size)
        val notificationText = notificationTexts[randomIndex]
        showNotification(applicationContext, notificationText)
        return Result.success()
    }
}

```

