



ŽILINSKÁ UNIVERZITA V ŽILINE

Fakulta riadenia  
a informatiky

Semestrálna práca z predmetu VAMZ

## **WORKOUT TRACKER**

**vypracoval:** Richard Laššo

**študijná skupina:** 5ZYI23

**cvičiaci:** doc. Ing. Patrik Hrkút, PhD.

**termín cvičenia:** pondelok bloky 2-3



## Obsah

1.	ÚVOD.....	3
2.	OBDOBNÉ APLIKÁCIE .....	3
1.	Gym Workout Tracker a Workout Planner .....	3
2.	Strava .....	4
3.	ANALÝZA NAVRHOVANEJ APLIKÁCIE .....	5
4.	POPIS VYTVORENEJ APLIKÁCIE.....	6
5.	Popis implementácie.....	7
6.	Záver.....	14



## 1. ÚVOD

Mojím cieľom je zhotoviť aplikáciu, ktorá bude pomocníkom cvičencovi pri samotnom cvičení ale aj pri evidencii samotných odtrénovaných tréningoch alebo pri plánovaní budúcich tréningov. Analýzu navrhovanej aplikácie rozoberieme podrobnejšie ďalej v dokumentácii. Dôvodom zhotovenia nasledovnej aplikácie je nedostačujúce funkcie už obdobných aplikácií ale aj uzamknuté množstvo funkcií za „paywallom“. V budúcnosti možnosť rozšíriť funkcionality o komunikáciu medzi rôznymi užívateľmi.

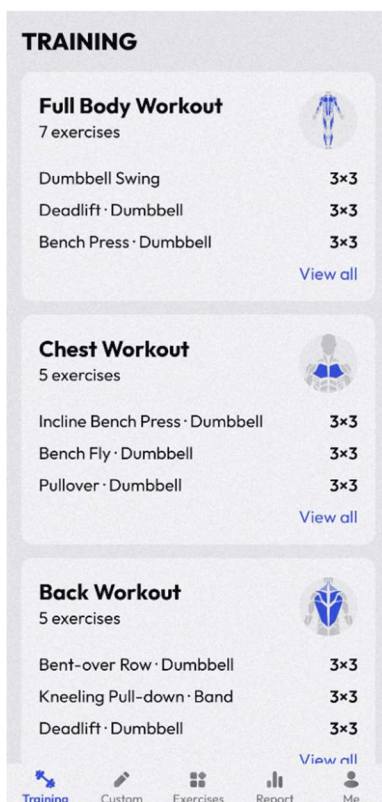
### Využitie aplikácie

- Sledovanie pokroku v tréningu
- Motivácia k dosiahnutiu fitness cieľov
- Zdieľanie/inšpirácia tréningov s priateľmi a trénermi
- Spustenie a sledovanie tréningov v reálnom čase
- Zobrazenie štatistík za určité obdobie

## 2. OBDOBNÉ APLIKÁCIE

### 1. Gym Workout Tracker a Workout Planner

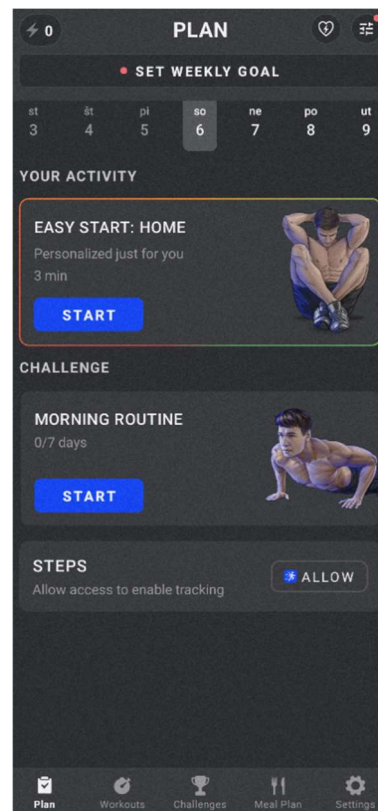
Aplikácie Gym Workout Tracker a Workout Planner slúžia na vytvorenie tréningu a nasledovné sledovanie posunu. Po inštalácii aplikácie pri prvom spustení zisťujú miery užívateľa a na základe toho vytvárajú tréningový plán. Zo začiatku sa rozhrania zdajú byť komplikované a niektoré funkcie sú uzatvorené za „paywallom“.



Zdroje:

[Gym Workout Tracker: Gym Log](#)

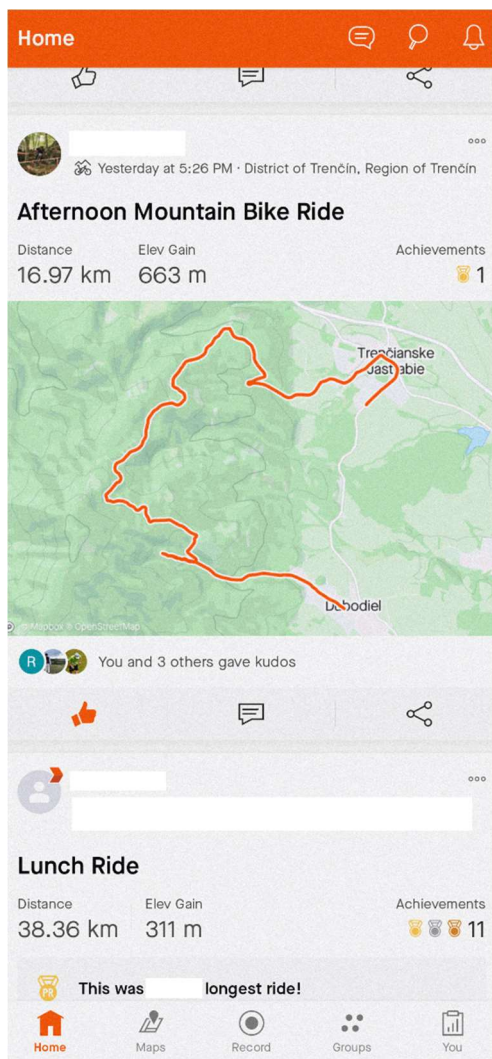
[Workout Planner Muscle Booster](#)





## 2. Strava

Aplikácia Strava obsahuje prehľadné zobrazenie dát a štatistík s možnosťou porovnania s predchádzajúcimi aktivitami (Beh, Bicykel,...). Taktiež sa tieto aktivity dajú jednoducho zdieľať s priateľmi, sledovateľmi a na sociálnych sieťach, s podporou lajkovania a komentovania pre posilnenie komunity a motivácie. Rôzne miesta obsahujú takzvané segmenty, kde používatelia môžu navzájom súťažiť o najrýchlejšie časy - leaderboardy.



Results		
Best Efforts	Segments	Achievements
6	7	6
Longest Ride 16.7 km - 3:48:16		
Biggest Climb 62 m		
Elevation Gain 256 m - 16.7 km		
<a href="#">View All Results</a>		

Zdroj:

[Strava](#)

### 3. ANALÝZA NAVRHOVANEJ APLIKÁCIE

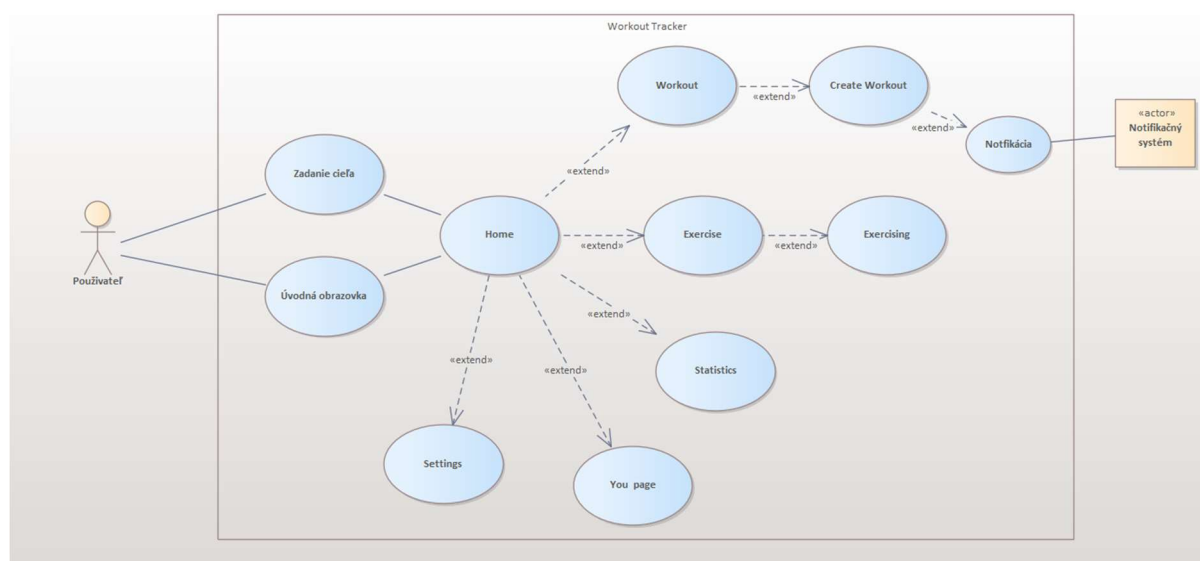
#### Funkcie

- **Sledovanie tréningov** – zaznamenávanie cvikov, sérii, opakovaní, váhy, času a rekordy „PR“
- **Štatistiky** – vytváranie grafov o určitých veciach, napr. posun v danom cviku za určité obdobie
- **Zdieľanie tréningov** – zdieľanie tréningov medzi priateľmi
- **Časovač a stopky** – meranie času oddychu medzi sériami a aj čas vykonávania jednotlivých cvičení
- **Ocenenia** – odomykanie ocenení za dosiahnutie cieľov – forma motivácie
- **Pridávanie poznámok** – možnosť spoznámkovať tréning

**Cieľové publikum** – aplikácia je vhodná pre pokročilejších atlétov, ktorí potrebujú evidovať, sledovať a plánovať svoje tréningy. Prípadne pre cvičencov, ktorí majú záujem jednoducho zdieľať a porovnávať svoje výkony s priateľmi. Aplikácia neslúži na určenie ideálneho tréningového plánu ale na evidenciu, prípadne zjednodušiť komunikáciu medzi trénerom a cvičencom.

**Dizajn** – Dizajn aplikácie by mal byť intuitívny, jednoduchý, prehľadný a príjemný na pozeranie aby sme zabezpečili jednoduchú navigáciu v aplikácii a tým aj neodrádzali potenciálnych užívateľov.

#### Use case diagram :





## 4. POPIS VYTVORENEJ APLIKÁCIE

Popis jednotlivých tried

- **Trieda Main** je hlavnou aktivitou aplikácie Workout Tracker. Obsahuje metódu `onCreate`, ktorá inicializuje obsah a spravuje navigáciu medzi rôznymi obrazovkami.
- **Trieda MainViewModel** je zodpovedná za riadenie logiky hlavnej aktivity
- **Trieda HomeScreen** definuje obrazovku domovskej stránky aplikácie Workout Tracker. Obsahuje zoznam cvikov s ich popismi a obrázkami. Slúži na motiváciu používateľa.
- **Trieda ExerciseScreen** definuje obrazovku cvičenia v aplikácii Workout Tracker. Táto obrazovka umožňuje používateľovi vybrať si cvičenie a začať cvičiť.
- **Trieda ExercisingScreen** predstavuje obrazovku cvičenia v aplikácii Workout Tracker. Táto obrazovka poskytuje užívateľovi kontrolu nad tréningom a zobrazuje mu aktuálne informácie o vykonávaní cvičení.
- **Trieda ExerciseViewModel** obsahuje logiku pre obrazovku cvičenia v aplikácii Workout Tracker. Táto trieda zabezpečuje načítanie dát z databázy a riadenie stavov pre zobrazenie cvičení a spustenie tréningu. Využívajú ju obe triedy **ExerciseScreen** a **ExercisingScreen**.
- **Trieda Notifications** zodpovedá za zobrazovanie oznámení v aplikácii Workout Tracker.
- **Trieda SettingsScreen** je zodpovedná za zobrazenie obrazovky nastavení v aplikácii Workout Tracker. Poskytuje možnosť nastavenia upozornení.
- **Trieda StatisticsScreen** predstavuje obrazovku štatistík v aplikácii Workout Tracker. Zobrazuje vizuálne štatistiky alebo grafy na základe dát získaných z používateľských cvičení. **Nie je doimplementovaná.**
- **Trieda WelcomeScreen** zodpovedá za zobrazenie úvodnej obrazovky aplikácie Workout Tracker. Používatelia tu môžu zadávať svoj cieľ a na základe toho dostanú motivačnú správu.
- **Trieda WelcomeViewModel** je zodpovedná za správu dát a biznis logiku pre úvodnú obrazovku aplikácie.
- **Trieda CreateWorkoutScreen** zodpovedá za zobrazenie obrazovky pre vytvorenie nového tréningu v aplikácii Workout Tracker. Umožňuje používateľovi pridávať cvičenia, počty opakovaní, série a váhy.
- **Trieda CreateWorkoutViewModel** slúži na manipuláciu s dátami cvičení v aplikácii. Obsahuje metódy na pridávanie, mazanie a aktualizovanie cvičení v databáze, ako aj na získavanie zoznamu cvičení pre aktuálny tréning.





- **Trieda WorkoutsScreen** je zodpovedná za zobrazenie zoznamu tréningov a cvičení v aplikácii. Obsahuje funkciu na vytvorenie nového tréningu a zobrazenie existujúcich tréningov s príslušnými cvičeniami.
- **Trieda WorkoutsViewModel** slúži na manipuláciu s dátami tréningov a cvičení v aplikácii pomocou Android architektúry životného cyklu a databázových modelov.
- **Trieda YouScreen** slúži na zobrazenie obrazovky s informáciami o používateľovi v aplikácii. **Nie je doimplementovaná.**
- **Trieda DatabaseViewModel** zabezpečuje prístup k dátam v databáze aplikácie.
- **Trieda Exercise** predstavuje entitu v databáze, reprezentujúcu jedno cvičenie.
- **Trieda Workout** predstavuje entitu v databáze, reprezentujúcu jeden tréning.
- **Rozhranie WorkoutDao** definuje prístupové metódy pre manipuláciu s dátami tréningov a cvičení v databáze.
- **Trieda WorkoutDatabase** predstavuje hlavnú databázu aplikácie, ktorá obsahuje tabuľky pre tréningy a cvičenia.
- **Trieda WorkoutWithExercises** je pomocná trieda, ktorá slúži na zoskupenie tréningu a príslušných cvičení do jednej jednotky dát.

## 5. Popis implementácie

### I. Otáčanie displeja

Aplikácia je navrhnutá tak, aby správne reagovala na otáčanie displeja. Využitie ViewModel tried zabezpečuje, že dáta vyplnené vo formulári zostanú zachované aj po zmene orientácie obrazovky. Rovnaký efekt dosiahneme aj pomocou rememberSaveable premenných, ktoré uchovávajú vyplnené polia aj po otáčaní obrazovky. Využitie LazyColumn zabezpečuje to aby mohol používateľ pri otočení displeja „scrollovať“ obrazovku.

```
var selectedType by rememberSaveable { mutableStateOf( value: "" ) }
var selectedExercise by rememberSaveable { mutableStateOf( value: "" ) }
var sets by rememberSaveable { mutableIntStateOf( value: 0 ) }
var reps by rememberSaveable { mutableIntStateOf( value: 0 ) }
var weight by rememberSaveable { mutableDoubleStateOf( value: 0.0 ) }
var weightText by rememberSaveable { mutableStateOf( value: "" ) }
var nameOfWorkout by rememberSaveable { mutableStateOf( value: "" ) }
var exercises by rememberSaveable { mutableStateOf<List<Exercise>>(emptyList()) }
var selectedExerciseIndex by rememberSaveable { mutableStateOf<Int?>( value: null ) }
var selectedExerciseCard by rememberSaveable { mutableStateOf<Exercise?>( value: null ) }
val showDialog = rememberSaveable { mutableStateOf( value: false ) }
```



## II. Resources

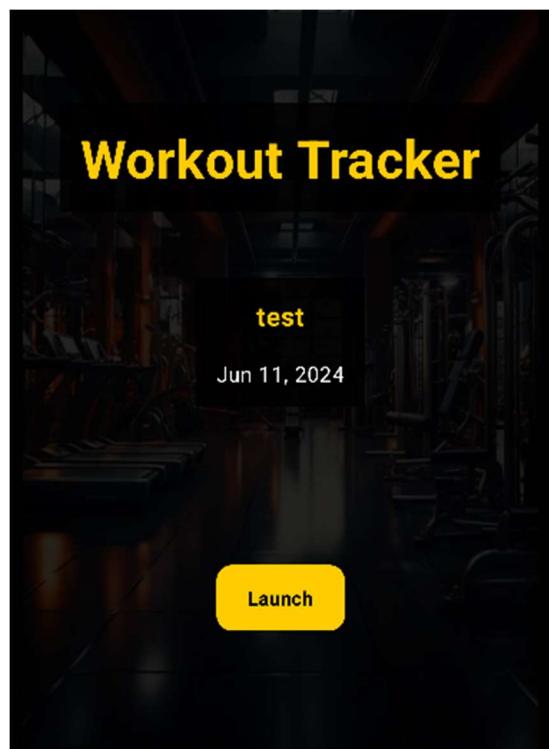
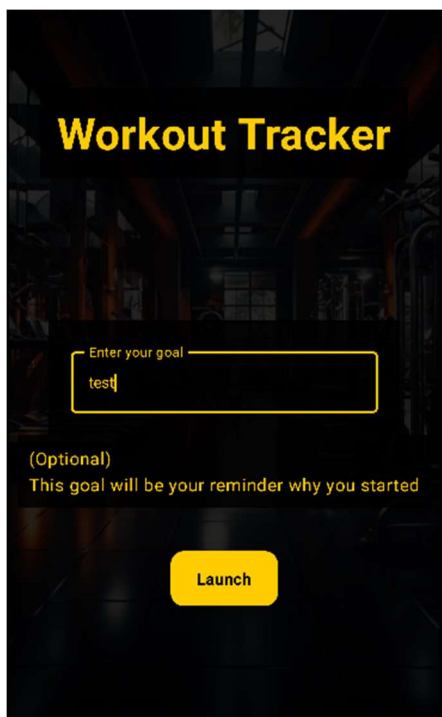
Všetky obrázky, ikony a názvy sú uložené v res. Konkrétne obrázky majú cestu /res/drawable a stringy /res/values/strings.xml

```
<resources>
  <string name="app_name">Workout Tracker</string>
  <string name="Welcome"/>
  <string name="Settings">Settings</string>
  <string name="Home">Home</string>
  <string name="Workouts">Workouts</string>
  <string name="You">You</string>
  <string name="Exercise">Exercise</string>
  <string name="Statistics">Statistics</string>
  <string name="Create">Create Workout</string>
  <string name="Exercising">Exercising</string>
</resources>
```

```
▼ drawable
  back_black.png
  benchpress.png
  deadlift.png
  </> dumbbell_solid.xml
  </> gear_solid.xml
  gym_background.jpg
  </> home.xml
  </> ic_launcher_background.xml
  </> ic_launcher_foreground.xml
  </> marker_solid.xml
  </> minus_solid.xml
  </> plus_solid.xml
  squat.png
  </> start_solid.xml
  </> statistics_solid.xml
  </> user_solid.xml
```

## III. Obrazovky aplikácie

- a. **Welcome screen** – táto obrazovka sa zobrazí ako prvá. V prípade prvého spustenia sa zobrazí input field do ktorého môže používateľ zadať svoj cieľ. Po opätovnom spustení sa tento cieľ zobrazuje aj s dátumom vloženia.



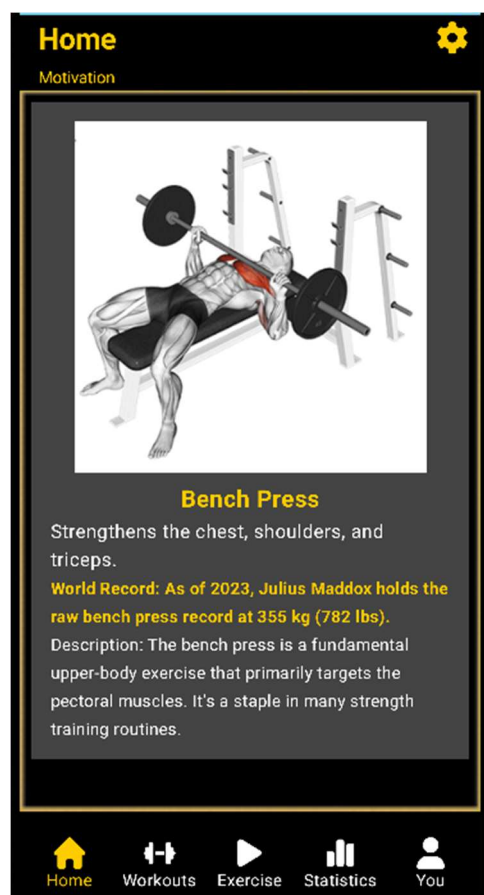


```
private var sharedPreferences = pSharedPreferences
private val userInput = mutableStateOf( value: sharedPreferences.getString( key: "user_goal", defValue: "" ) ?: "" )
private val inputDate = mutableLongStateOf( sharedPreferences.getLong( key: "input_date", defValue: 0L ) )
private val isFirstRun = mutableStateOf( sharedPreferences.getBoolean( key: "first_run", defValue: true ) )
```

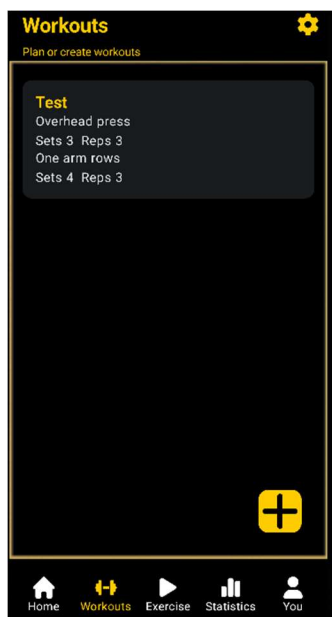
Využitie sharedPreferences na zapamätanie zadaných údajov pri opätovnom spustení.

- b. **Home screen** – obrazovka na ktorej sú vyobrazené cviky veľkej trojky s údajmi o svetových rekordoch, môže obsahovať aj iné motivačné informácie.

```
LazyColumn(
    modifier = Modifier
        .fillMaxSize()
        .padding(10.dp),
    horizontalAlignment = Alignment.CenterHorizontally,
    verticalArrangement = Arrangement.spacedBy(40.dp)
) { this: LazyListScope
    item { this: LazyListItemScope
```



LazyColumn využitý pre scrollovanie Home screenu.



- c. **Workouts screen** – obrazovka, na ktorej sú vyobrazené všetky aktuálne vytvorené tréningy. Tréningy sú uložené v databáze. Vpravo dole sa nachádza + tlačidlo, ktoré otvorí nový screen – Create workout.

```

@reaz
@SuppressLint("CoroutineCreationDuringComposition")
@Composable
fun WorkoutsScreen(navigateToCreateWorkoutScreen: () -> Unit,
    modifier: Modifier, painter: Painter,
    workoutsViewModel: WorkoutsViewModel)
{
    val workoutsViewScope = rememberCoroutineScope()

    val workouts by workoutsViewModel.workouts.observeAsState(emptyList())
    val exercises by workoutsViewModel.exercises.observeAsState(emptyList())

    val workoutsWithExercises = workouts.map { workout ->
        val exercisesForWorkout = exercises.filter { it.workoutId == workout.id }
        WorkoutWithExercises(workout, exercisesForWorkout) }map
    }
}

```

```

val workouts: LiveData<List<Workout>> = databaseViewModel.workouts

val exercises: LiveData<List<Exercise>> = databaseViewModel.exercises

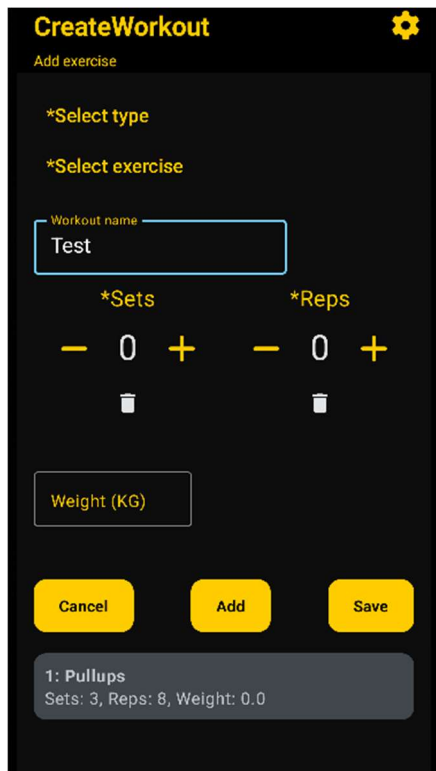
@reaz
suspend fun createNewWorkout()
{
    val newWorkout = Workout(
        id = getNextWorkoutId()
    )
    database.workoutDao().insertWorkout(newWorkout)
}

```

**LiveData** - zabezpečenie sledovania zmeny údajov v rámci ViewModelu a správneho aktualizovania v užívateľskom rozhraní



- d. **CreateWorkout screen** – obrazovka pomocou ktorej je možné vytvárať tréning a následne ho uložiť do databázy. V dolnej časti sú vyobrazené všetky cviky, ktoré vytváraný tréning obsahuje. Cvik je možné zmazať klikom na daný cvik, kde sa zobrazí možnosť na odstránenie.

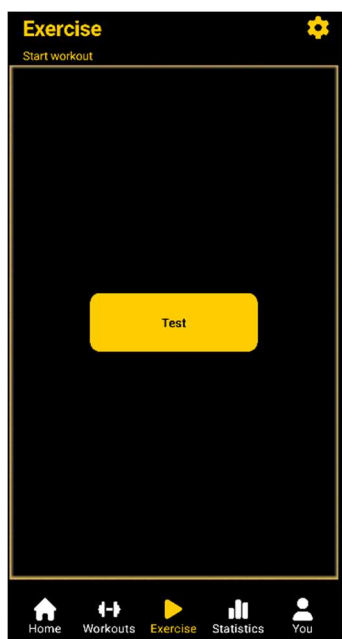


**rememberSaveable** – zachovanie stavu komponentov pri zmene konfigurácie (otáčanie obrazovky)

**kotlinx.coroutines a Dispatchers.IO** – vykonávanie asynchrónnych operácií ako je napr pridávanie cvikov alebo aktualizácia mena tréningu

```
reez
suspend fun updateWorkoutName(newName: String) {
    var name = newName
    if (newName == "") name = "Workout"
    withContext(Dispatchers.IO) { this: CoroutineScope
        database.workoutDao().updateWorkoutName(getActualWorkoutId(), name)
    }
}
```

- e. **Exercise screen** – obrazovka, ktorá ponúkne na výber používateľovi všetky tréningy, ktoré boli vytvorené. Používateľ má možnosť si vybrať jeden z tréningov. Po výbere je presunutý na Exercising screen.

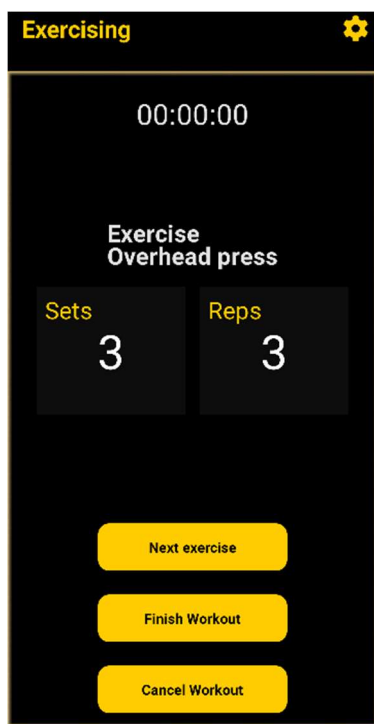




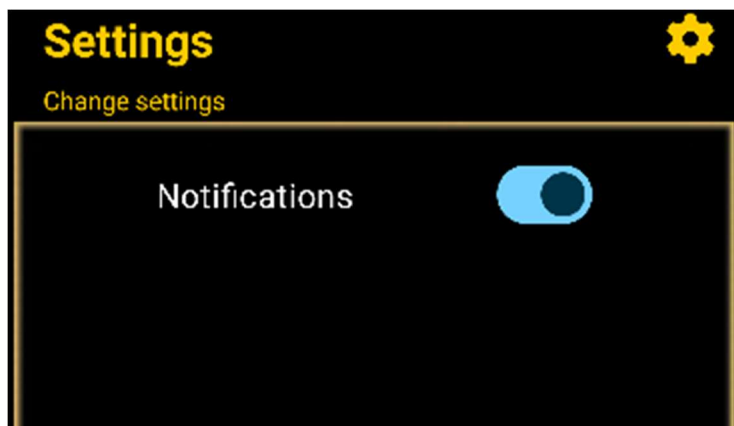
Keď sa zmenia dáta v databáze, LiveData automaticky upozorní čo umožňuje aktualizáciu zobrazených údajov v reálnom čase.

```
val workouts: LiveData<List<Workout>> = databaseViewModel.workouts  
  
val exercises: LiveData<List<Exercise>> = databaseViewModel.exercises
```

- f. **Exercising screen** – obrazovka, ktorá zobrazuje aktuálny tréning. Zobrazuje počet sérii a opakovaní pre daný cvik a má možnosti ako ukončiť tréning a uložiť, zrušiť a posunúť sa na ďalší cvik.



- g. **Settings screen** – obrazovka, ktorá zobrazí nastavenia, konkrétne zapnutie a vypnutie notifikácií.





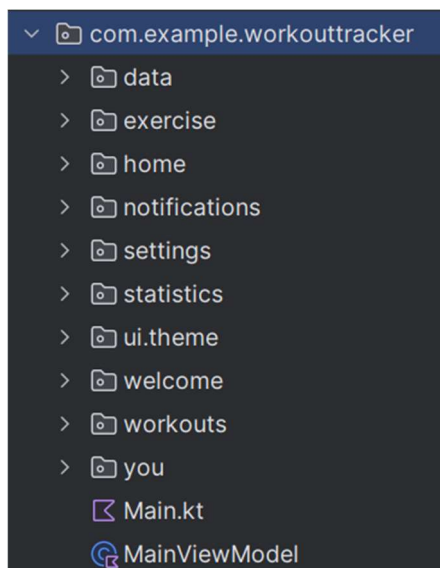
#### IV. Práca s Gitom

Je dôležité pravidelne uploadovať prácu na Git, aby sa zachoval aktuálny stav kódu, preto bola aplikácia vyvíjaná s použitím Git. Zmeny boli pravidelne uploadované do tohto systému.

Final push	reez	Today 21:37
Final push	reez	Today 21:31
Pridanie cvicenie screenu	reez	Today 16:31
Maly fix	reez	Today 11:33
Pridanie 2 notifikacii	reez	Today 11:28
Spravenie cvicenie screenu	reez	Today 1:53
Zobrazenie treningov z databazy	reez	Yesterday 23:52
Pridany textfield na zmenu mena treningu	reez	Yesterday 20:54
Pridanie scrollu na create workout screen	reez	Yesterday 20:04
Dokoncenie create workout screeny	reez	Yesterday 18:38
Pridavanie do databazy cez CreateWorkout a zobrazenie cvikov	reez	Yesterday 15:52
Pridanie databazy	reez	Yesterday 12:35
Otacanie screenu input fix	reez	Yesterday 11:07
Pridanie screenu na vytvorenie treningu	reez	9. 6. 2024 23:09
Graficke upravu	reez	9. 6. 2024 18:23
Mensie upravu na uvitacej obrazovke	reez	9. 6. 2024 15:53
Fix navigacne bary	reez	9. 6. 2024 15:34
Spravena uvodna obrazovka	reez	9. 6. 2024 15:05
Pridanie dalsich okien	reez	9. 6. 2024 11:12
Pridanie xml ikon, prerobenie logiky navigacneho baru	reez	9. 6. 2024 10:53
Prerobenie navigacneho baru, pridanie top baru a pridanie screenov	reez	9. 6. 2024 0:37
Pridany navigancy bar s tlačidlami	reez	7. 6. 2024 12:23
Pridanie projektu	reez	7. 6. 2024 11:24

#### V. Architektúra aplikácie

Aplikácia je organizovaná do viacerých balíčkov, ktoré pomáhajú pri riadení kódu, udržiavaní prehľadnosti a zlepšujú škálovateľnosť projektu. Každý balíček obsahuje súvisiace triedy, komponenty alebo funkcionality, čo umožňuje lepšiu organizáciu a modularitu aplikácie.



#### VI. Dizajn aplikácie

Cieľom bolo vytvoriť aplikáciu na sledovanie tréningov, ktorá poskytuje užívateľom jednoduchý a efektívny spôsob správy ich cvičebných plánov. Hlavnými bodmi boli jednoduché použitie, prehľadnosť a organizácia a estetický dizajn



## 6. Záver

Pri vývoji tejto aplikácie som získal hlbšie porozumenie procesov spojených s tvorbou softvéru, najmä v oblasti mobilných aplikácií. Taktiež som mal príležitosť aplikovať moderné technológie, ako sú Jetpack Compose a ViewModel. Taktiež som si okúsil vývoj softvéru s databázou a prácu s databázou v mobilnej aplikácii.

Vývoj tejto semestrálnej práce bol pre mňa zložitý. Stalo sa mi, že mi vývojové prostredie niekoľkokrát reštartovalo notebook. Ďalej na vypracovanie som nemal ani moc času z dôvodu uprednostnenia ostatných povinných predmetov ale aj kvôli osobnému životu. Napriek tomu si myslím, že som vytvoril niečo zaujímavé. Danú aplikáciu sa určite pokúsim rozšíriť ale nevyužijem na to Kotlin, pretože z tejto skúsenosti som usúdil, že tento jazyk sa mi moc nepáči.