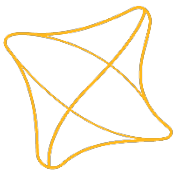


ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Vývoj aplikácií pre mobilné zariadenia (VAMZ)

Tréningová aplikácia

Marek Klocok.
5ZYI24

2023/2024

1. Špecifikácia zadania

Aplikácia slúži na zaznamenávanie tréningov. Používateľ je schopný si vytvoriť svoj vlastný tréning. Do tohto tréningu je možné vkladať rôzne cviky, ich počet sérií, opakovaní, váhy, s ktorými bol cvik vykonaný. Dokáže ich taktiež neskôr upravovať a mazať. Tieto tréningy je taktiež možné odcvičiť, kedy sa dajú upravovať počas tréningu. Pri cvičení daného tréningu je taktiež časovač, ktorý meria dĺžku tréningu a je zaznamenaný dátum tréningu. Každý úspešne docvičený tréning je zaznamenaný v histórií aj s popisom cvikov, trvania tréningu a jeho deň uskutočnenia. Používateľ má aj svoj vlastný profil, kde si vie zadať súkromné informácie.

2. Podobné aplikácie

Hevy - Gym Log Workout Tracker

Hevy Gym Workout Tracker
Nákupy v aplikácii

4,9★ 67,4 tis. recenzií | 1 mil.+ Stiahnuté | Rodičovský dohľad

StrongLifts Weight Lifting Log

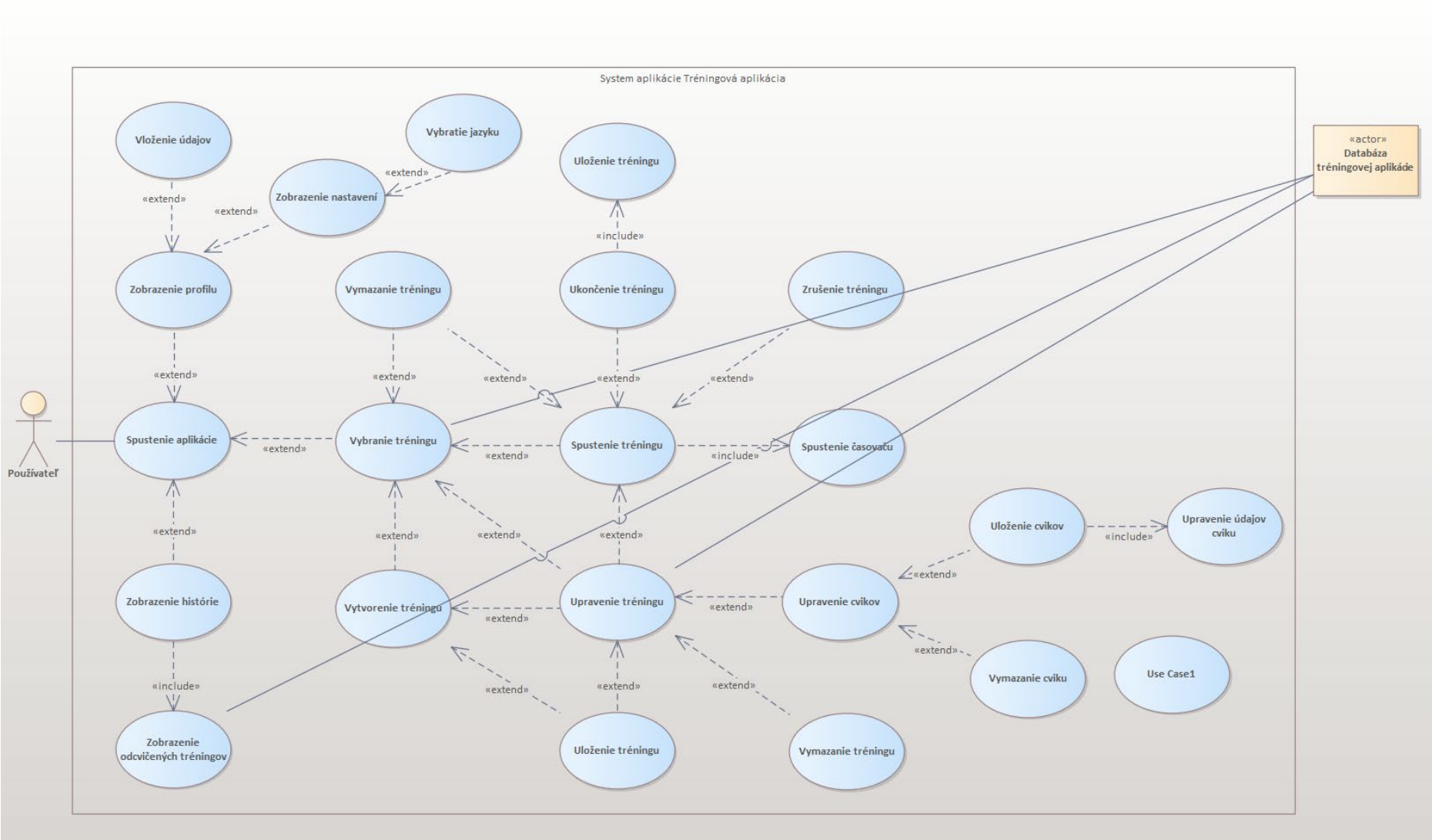
StrongLifts
Nákupy v aplikácii

4,8★ 98,8 tis. recenzií | 1 mil.+ Stiahnuté | Pre všetkých

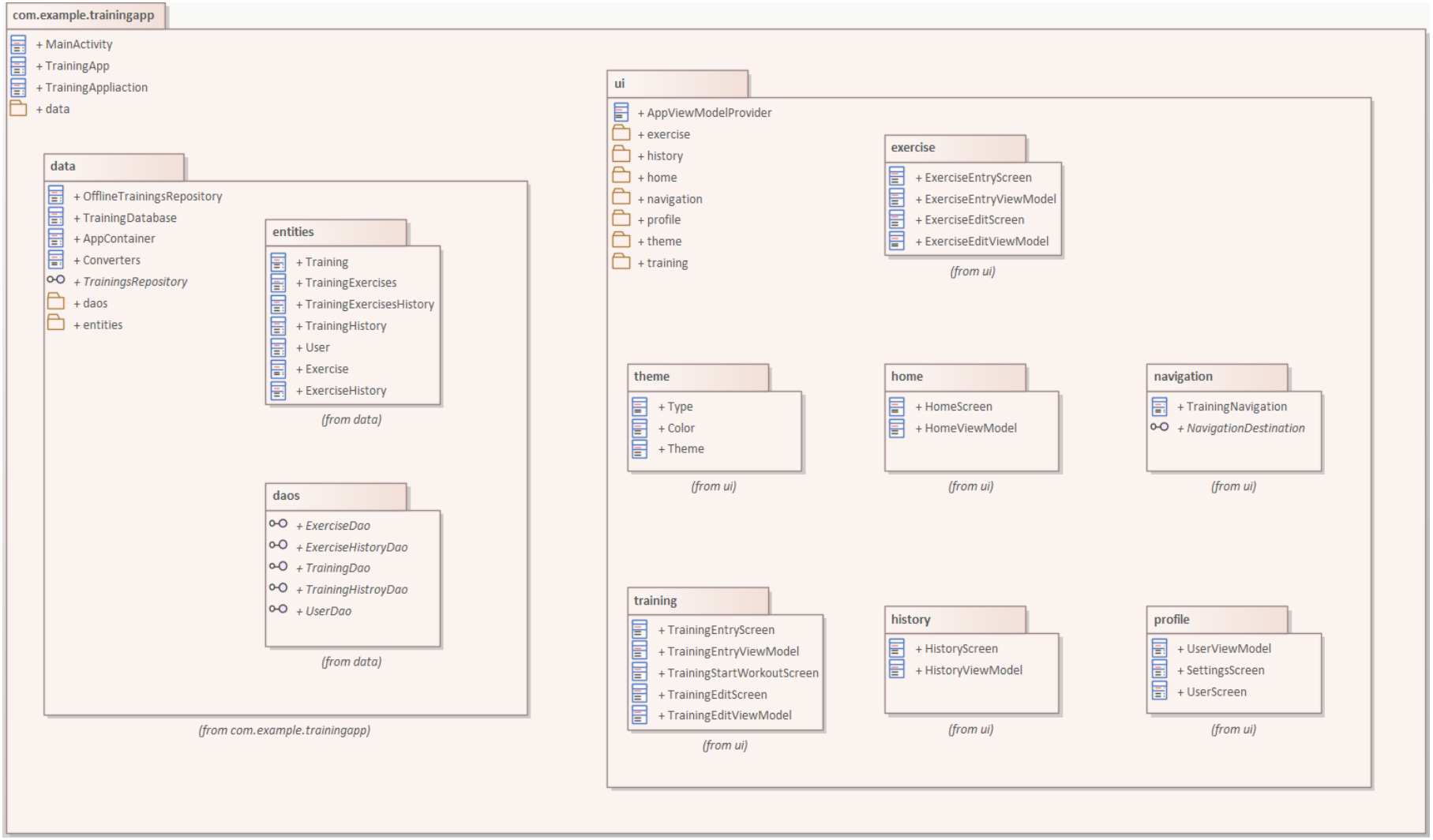
3. Odlišnosť aplikácie

V mojej aplikácií je ihneď možné začať cvičiť svoj tréning a nie je potrebné sa zložito preklikávať medzi obrazovkami. Moje rozloženie tréningov je prehľadnejšie, keďže mám v jednom riadku 2 tréningy, ktoré obsahujú informáciu o názve tréningu a jeho posledný dátum odcvičenia, čo je jednoduchšie pre užívateľa. V histórií tréningov sa taktiež nachádza informácia o dĺžke trvania tréningu, čo tieto aplikácie nemajú.

4.Návrh riešenia problému



5.Diagram balíčkov



6. Popis implementácie

Aplikácia sa spúšťa cez MainActivity, ktorá využíva LifeCycle metódu „onCreate“, kde na začiatku nastaví tému aplikácie na základe zariadenia používateľa. Následne spustí tréningovú aplikáciu a jej Navigation, ktorý umožňuje sa pohybovať v celej aplikácií. Aplikácia využíva lokálnu databázu Room, ktorá umožňuje uchovávať všetky tréningy, cviky a informácie o užívateľovi. Skoro každá obrazovká má ViewModel na uchovávanie stavov pri otočení obrazovky a aktualizovanie stavov v aplikácií.

Užívateľ po spustení aplikácie je schopný vytvoriť, spustiť, upraviť a vymazať tréning kedy ho aplikácia pomocou Dialog okna vyzve či naozaj chce vykonať túto akciu. Cviky v tréningoch je taktiež možné vytvárať, meniť a mazať, kedy taktiež sa zobrazí Dialog okno. Po odcvičení tréningu je každý jeden tréning zapísaný do databázy a v histórií je možné ich prezerať. V profile si môže užívateľ nastaviť základné informácie.

Navigation

```
package com.example.trainingapp.ui.navigation

import ...

MrKerinoo
@Composable
fun TrainingNavigation(
    navController : NavHostController,
    modifier: Modifier = Modifier,
) {
    NavHost(
        navController = navController,
        startDestination = HomeDestination.route,
        modifier = modifier
    ) { this: NavGraphBuilder
        composable(route = HomeDestination.route) { this: AnimatedContentScope  it: NavBackStackEntry
            HomeScreen(
                navigateToHistory = { navController.navigate(HistoryDestination.route) },
                navigateToProfile = { navController.navigate(UserDestination.route) },
                navigateToTrainingEntry = { navController.navigate(TrainingEntryScreenDestination.route
                navigateToTrainingEdit = { it: Int
                    navController.navigate( route: "${TrainingEditDestination.route}/$it")},
                navigateToTrainingStartWorkout = { it: Int
                    navController.navigate( route: "${TrainingStartWorkoutDestination.route}/$it")}
            )
        }
```

Room

```
MrKerinoo
@Database(entities = [Training::class, Exercise::class,
    TrainingHistory::class, ExerciseHistory::class, User::class],
    version = 28, exportSchema = false)
@TypeConverters(Converters::class)
abstract class TrainingDatabase : RoomDatabase() {
    MrKerinoo
    abstract fun trainingDao(): TrainingDao
    MrKerinoo
    abstract fun exerciseDao(): ExerciseDao
    MrKerinoo
    abstract fun historyDao(): TrainingHistoryDao
    MrKerinoo
    abstract fun exerciseHistoryDao(): ExerciseHistoryDao
    MrKerinoo
    abstract fun profileDao(): UserDao
```

ViewModel

```
MrKerinoo
class TrainingEditViewModel(
    savedStateHandle: SavedStateHandle,
    private val trainingsRepository: TrainingsRepository
) : ViewModel()
{
    private val trainingId: Int = checkNotNull(savedStateHandle[TrainingEditDestination.trainingIdArg])
    val timer = MutableStateFlow( value: 0)

    MrKerinoo
    init {
        viewModelScope.launch { this: CoroutineScope
            while (true) {
                delay( timeMillis: 1000L)
                timer.value++
            }
        }
    }

    val exercisesUiState: StateFlow<ExercisesUiState> =
        trainingsRepository.getExercisesForTrainingStream(trainingId).map {ExercisesUiState(it)}
            .stateIn(
                scope = viewModelScope,
                started = SharingStarted.WhileSubscribed( stopTimeoutMillis: 5000),
                initialValue = ExercisesUiState()
            )


    MrKerinoo
    var trainigUiState by mutableStateOf(TrainingUiState())
        private set
```