ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Semestrálna práca z predmetu

vývoj aplikácií pre mobilné zariadenia

FitnessFlow

Vypracoval: Marián Bobček

Študijná skupina: 5ZIF11

Akademický rok: 2024/2025

V Žiline dňa 7.4.2025

Obsah

Úvod

Nápad na aplikáciu Fitness Flow vznikol z osobnej potreby vytvoriť si nástroj, ktorý by mi pomohol systematicky pristupovať k vlastnému tréningovému režimu a celkovo organizácii voľného času, v rámci nastavovania dátumu/času tréningov. V minulosti som si častokrát zapisoval odcvičené tréningy, ale chýbala mi jednotná platforma, ktorá by tieto zápisky prepojila s plánovaním, pripomienkami, časovým manažmentom a zároveň umožnila jednoduchú spätnú analýzu môjho pokroku.

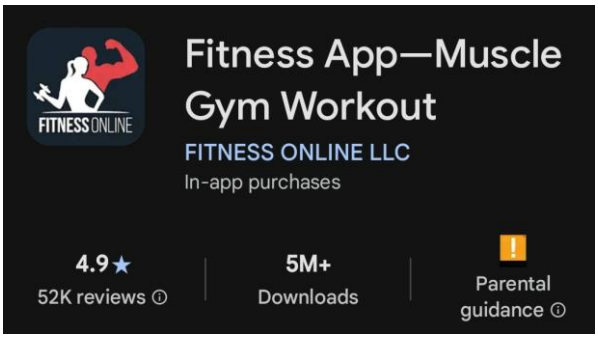
Hlavnou myšlienkou aplikácie je teda pomôcť jednotlivcom, zatiaľ hlavne mne a mojím kamarátom, efektívne sledovať, plánovať a vyhodnocovať svoj tréningový proces. Okrem evidencie samotných tréningov (počet sérií, opakovaní, váhy, druh cviku, partia tela, typ, poznámka) chcem, aby aplikácia pomáhala aj s doplnkovými aktivitami, ako je strečing, aktívna regenerácia alebo plávanie a iné. Všetko bude záležať od zamerania jednotlivca a od toho čo všetko by som tam stihol pridať. Zároveň má slúžiť ako pripomienkový systém – pomôže mi zaradiť tréning do bežného dňa tak, aby bol vyvážený s ostatnými aktivitami.

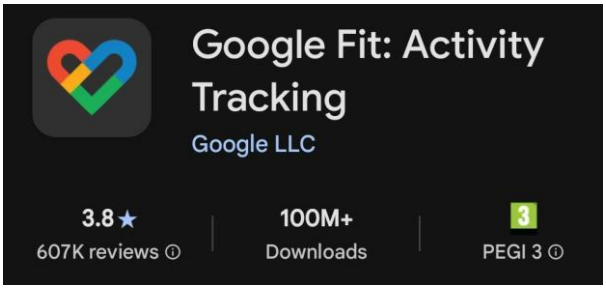
Zámerom mojej práce je teda vytvoriť mobilnú aplikáciu, ktorá bude intuitívna a zároveň poskytne praktické funkcie pre každodenný tréning – od základného nastavenia cieľov, cez sledovanie progresu, až po plánovanie tréningov v kalendári. Ak to bude reálne, doplnil by som aj možnosť stanovenia cieľa v rámci hmotnosti jedinca(chudnutie, pribratie), zápis cieľu zdvihnutia určitej hmotnosti(maximálna hmotnosť drepu).

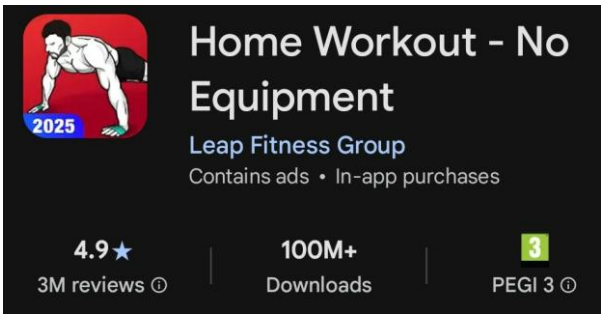
Postupne si chcem túto aplikáciu vyvíjať pre seba ako určitý spis o všetkom čo sa týka môjho pôsobenia. Napríklad to čo som zatiaľ spravil urobiť len ako podmnožinu v rámci ďalších smerov. Tým myslím napríklad výdavky, príjmy, celkovo financie, kalendár pre školu,aktivity,stretnutia, narodeniny, pripomienky, zápisník nápadov, vylepšení. V budúcnosti by som chcel aby sa jednalo o veľmi komplexnú a robustnú aplikáciu, ktorá bude vyhovovať mojím predstavám a dufám, že mi pomôže v mojom plánovaní a celkovo zlepší môj výkon. To čo som zatiaľ ponúkol(figma,diagramy) nie je finálna verzia ale to čo mi zatiaľ napadlo a stihol som to dať dokopy. Graficky návrh som urobil asi dosť náročný a priblížiť sa mu bude namáhavé ale budem sa snažiť aspoň niektoré obrazovky dotiahnuť k tomu čo som navrhol. Bol by som vďačný za spätnú odozvu, či to bude reálne poprípade čo vyhodiť alebo pozmeniť.

Prehľad podobných aplikácií

Pre porovnanie som si vybral tieto dostupné aplikácie na google play







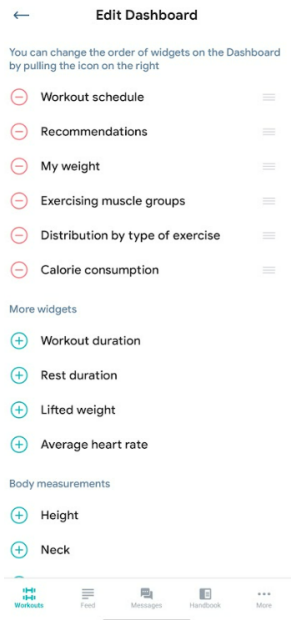
Aplikácia Fitness App – Muscle Gym Workout

Funkcie:

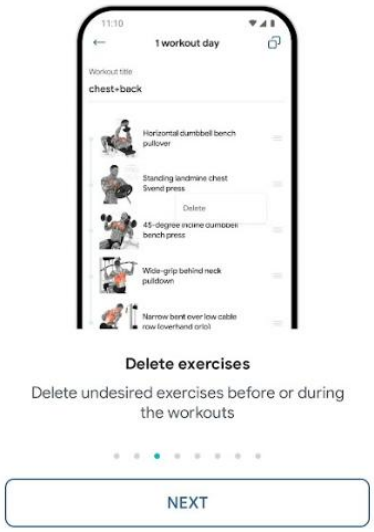
- Tvorba vlastných cvičebných plánov
- Pridávanie cvikov do týchto plánov
- Kalendár kde mi doplní čo a kedy mám cvičiť
- Spustenie daného plánu kde ráta strávený čas, pauzu, počet sérií,komentár
- Nastavenie plánu jedenia teda zadanie času kedy mám aké jedlo
- Pridanie doplnkov stravy
- Kanál kde môžem pridať post, ktorý môže niekto lajknúť,komentovať
- Komunikačný kanál s ostatnými ľuďmi napríklad trénermi

Výhody:

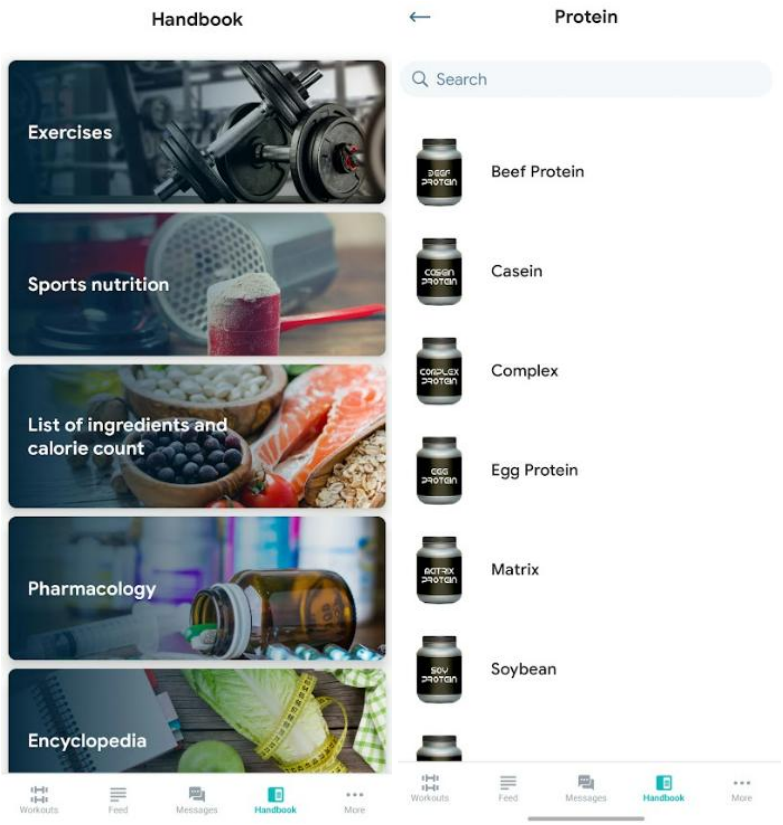
- Animácie a popis jednotlivých cvikov ako aj zobrazenie záťaže jednotlivej partie tela
- Upravenie čo chcem zobraziť na obrazovke



- Vysvetlivky ako čo funguje pri pridávaní cvikov a celkovo k obsluhe aplikácie



- Príručka s rôznymi informáciami

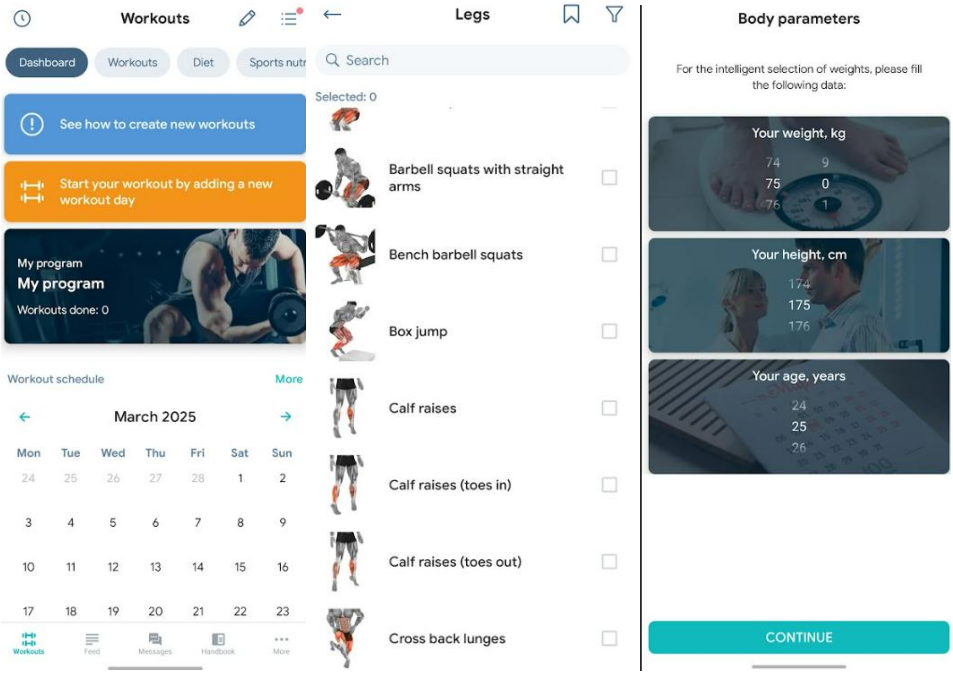


Nevýhody:

- Predplatné bez ktorého sú niektoré funkcie veľmi obmedzené
- Obmedzené ovládanie kalendáru

Porovnanie oproti ostatným

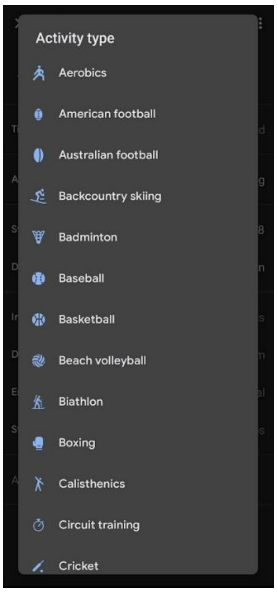
- Páči sa mi komunikačný kanál a príručka a vcelku intuitívne ovládanie aplikácie



Aplikácia Google Fit: Activity Tracking

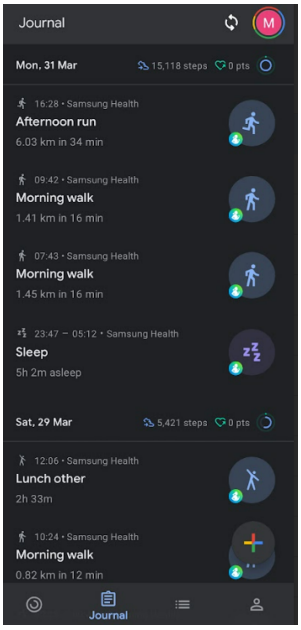
Funkcie:

- Rátanie krokov, kalórií, doby pohybu
- Plnenie denných cieľov so zobrazením v rámci mesiaca, týždňa
- Pridanie vykonaných aktivít



Výhody:

- Veľmi sa mi páči prepojenie s SamsungHealth kde mi zobrazí jednotlivé aktivity spolu s tepom, tempom napríklad pri behu alebo pri cvičení



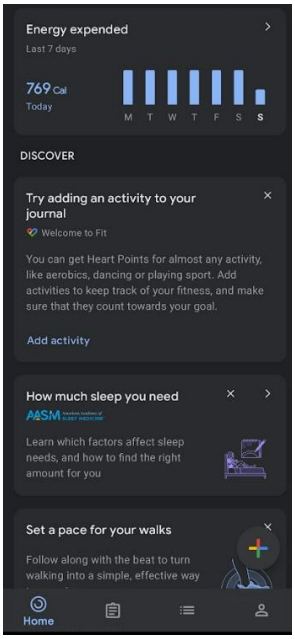
- Jednoduché ovládanie
- Odkazy napríklad na to koľko spánku potrebujem a čo ovplyvňuje spánok

Nevýhody:

- Asi žiadne okrem pridania budúcej aktivity

Porovnanie oproti ostatným:

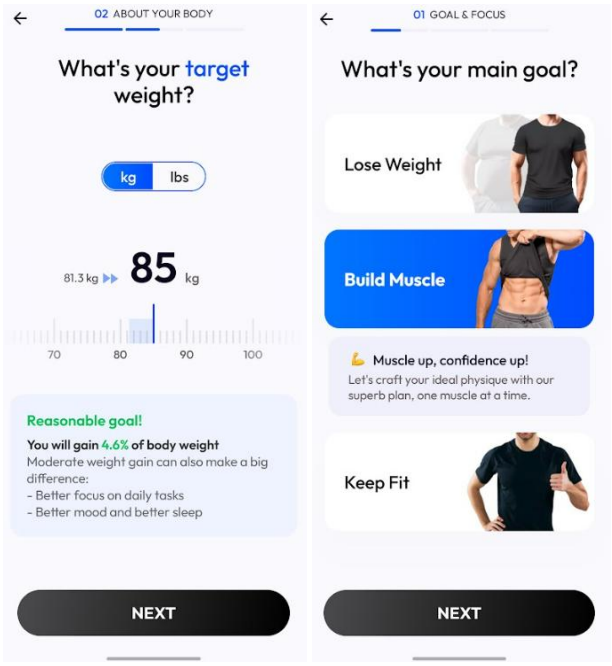
- Páči sa mi prepojenie s Samsung Health
- Oproti ostatným nie je táto aplikácia taká robustná
- Je rýchla



Aplikácia Home Workout – No Equipment

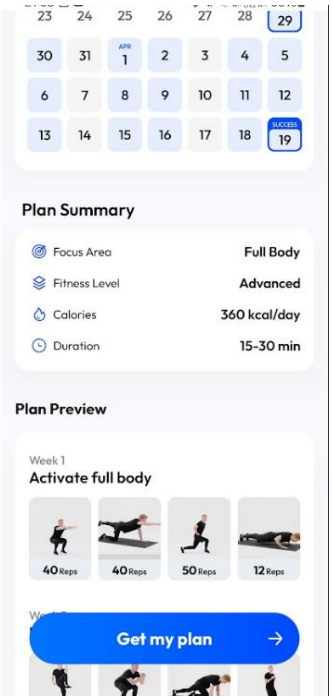
Funkcie:

- Oproti prvej aplikácií Fitness App – Muscle Gym Workout, ponúka cviky len na doma
- Taktiež je možný výber z rôznych svalových partií a úrovní podľa ktorých potom nastaví intenzitu a cviky



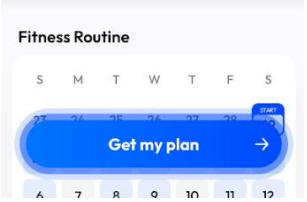
Výhody :

- Aj bez predplatného sú funkcie kvalitné
- 7 dní skúšky bez platenia
- Prednastavenie dĺžky jednotlivých cvikov ich poradia alebo zmeny za iný podobný cvik



Nevýhody:

- Reklamy
- Chcem si len niečo pozrieť a už ma to znásilňuje aby som šiel cvičiť
- Trošku prestrelené ciele, aby som za cca 22-23 dní pribral 3,7kg je veľmi náročné. Záleží či to je teda svalová hmota alebo mix vody tukov a svalu, vtedy je to možné. Teoreticky viem pribrať 1kg za minútu ak vypijem liter tekutiny.



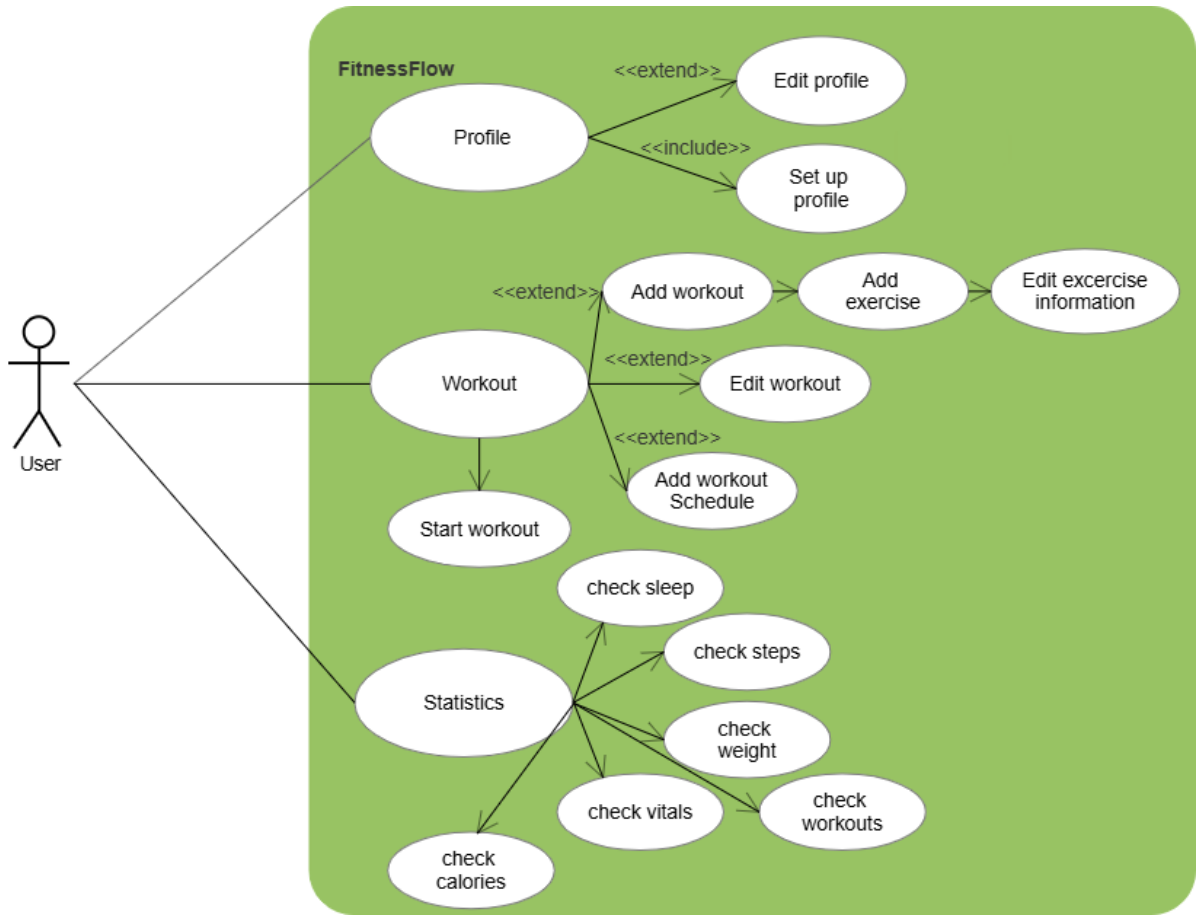
Porovnanie oproti ostatným:

- Ponúka množstvo cvičebných plánov pre rôzne úrovne
- Veľmi kvalitné spracovanie popisov jednotlivých cvikov a rôznych tipov

Zhodnotenie a porovnanie s mojou pripravovanou aplikáciou

- Každá z aplikácií ponúka rôzne funkcie ktoré by som chcel v budúcnosti implementovať
  - Spojenie s Samsung Health na zber údajov z hodiniek
  - Príručky ohľadom cvičenia a spánku
  - Zoznam doplnkov stravy
- Moja aplikácia sa bude podobať na aplikácie Aplikácia Home Workout – No Equipment a Fitness App – Muscle Gym Workout dáky taký mix medzi nimi
- Bude mať uvítacie obrazovky, ktoré majú aj dve vyššie spomenuté aplikácie
- Pridanie vlastných tréningov, cvikov
- Ak to bude možné pridanie štatistík spánku, tepu a iné

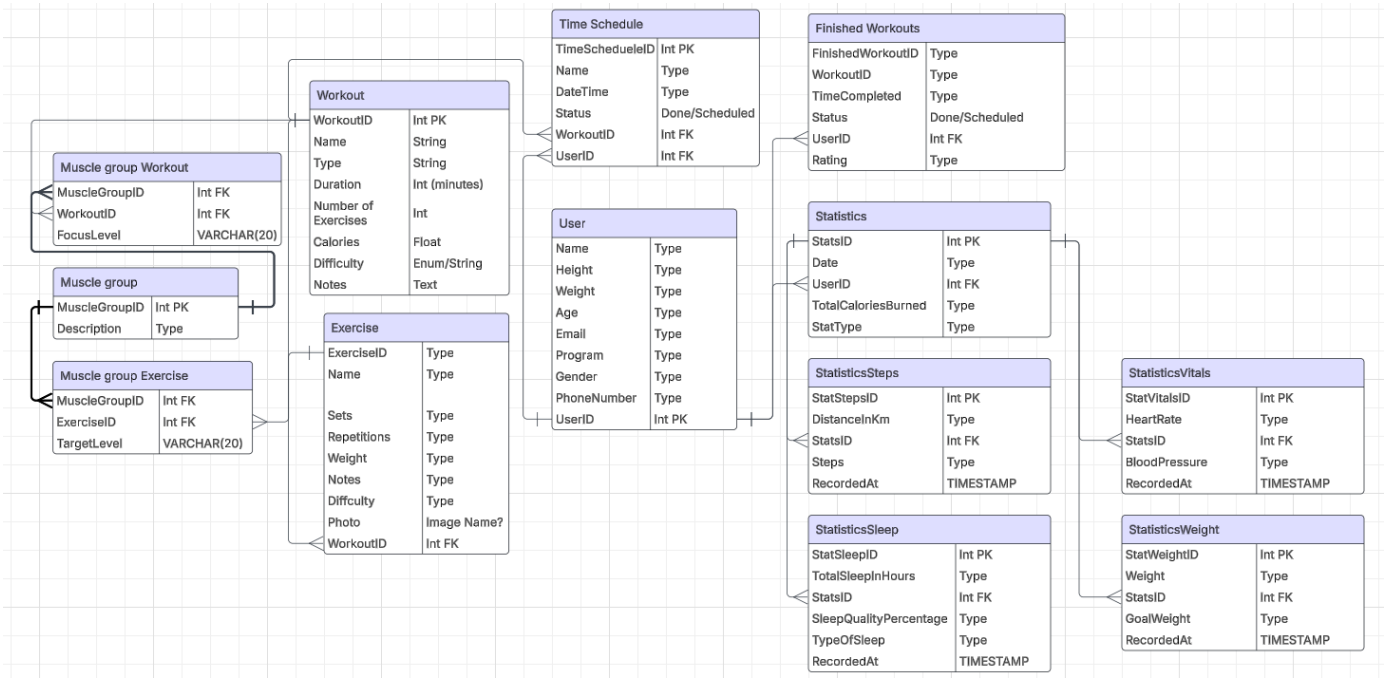
Analýza navrhovanej aplikácie

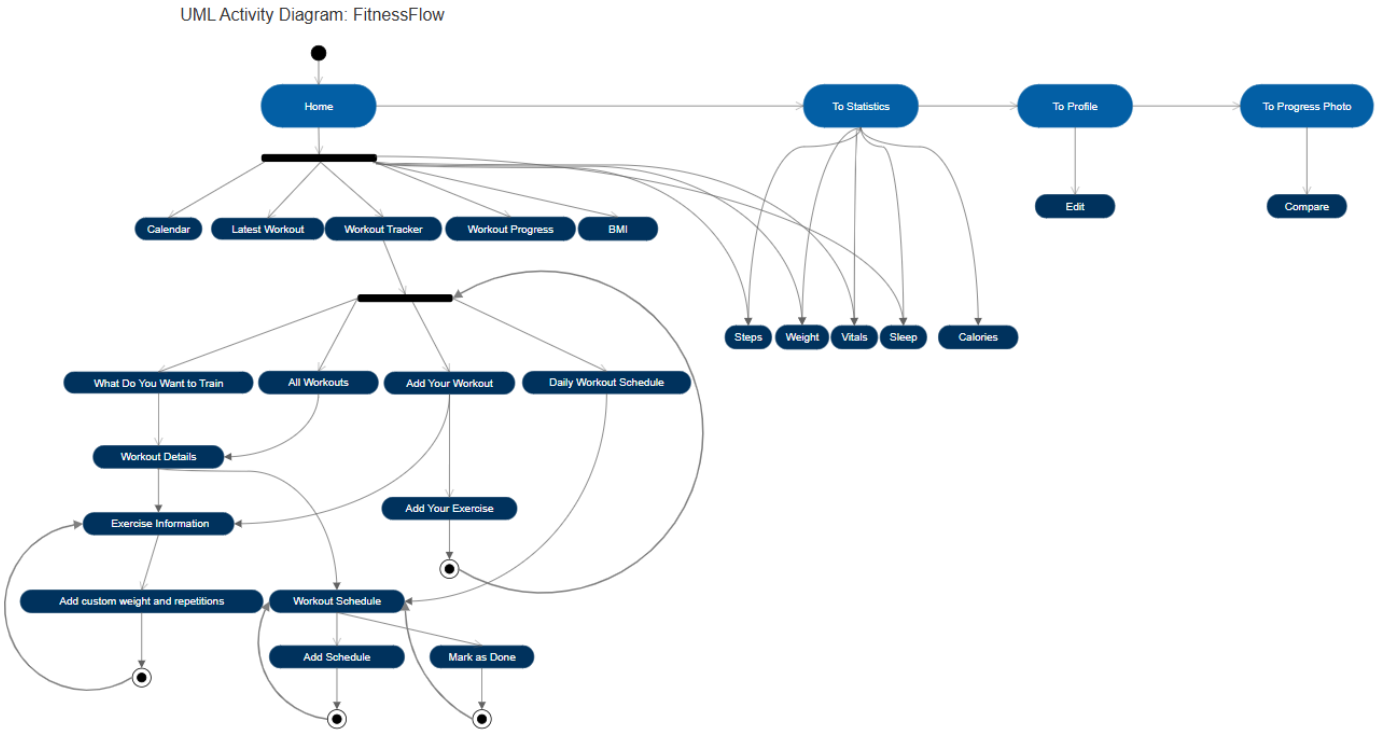


Mimofunčkné požiadavky:

- Bezpečnosť – ukladať dáta ako vek, váha, fotky len lokálne a možno šifrovane? (nie som si istý)

Návrh architektúry aplikácie





1. View (UI vrstva)

- zobrazuje obsah obrazoviek ako Home, Workout Tracker, Profile, Statistics a ďalšie
- stav z ViewModelu sa premieta do UI prvkov (napr. zobrazenie BMI, grafu progresu alebo rozvrhu cvičení)

2. ViewModel (Stavová logika)

- uchováva aktuálny UI stav a spracováva používateľské interakcie
- zabezpečuje komunikáciu medzi View a dátovou vrstvou (napr. správa tréningov, plánovanie rozvrhu, načítanie štatistík)

3. Model (Dátová vrstva)

- zahŕňa databázové entity, ako je znázornené v UML dátovom diagrame

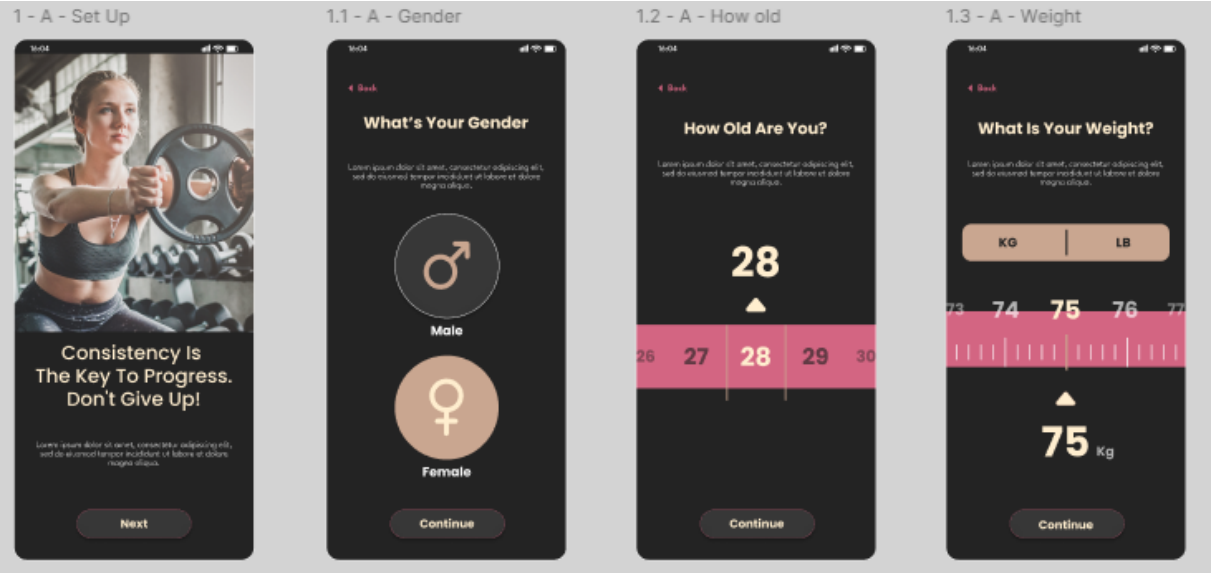
Návrh vzhľadu obrazoviek

V tejto časti opíšem pár vybraných obrazoviek. Celý doterajší progres môjho návrhu bude tu čo je github(mám to ako sukromné, asi to nepôjde) alebo priamo link na figmu (ak nepôjde dajte mi vedieť).

Úvodné obrazovky

Jedná sa o uvítacie obrazovky pri prvom spustení aplikácie. Horná lišta sa nebude zobrazovať. Použité komponenty sú navigačné tlačidlá teda šípky a textové bloky.

Zber vstupných údajov



Tak isto pri prvom spustení sa vykoná zadávanie základných informácií/údajov, zatiaľ navrhnutý tmavý režim ale postupom času by som chcel implementovať aj svetlý. Použité komponenty sú navigačné tlačidlá Continue.

Domovská obrazovka

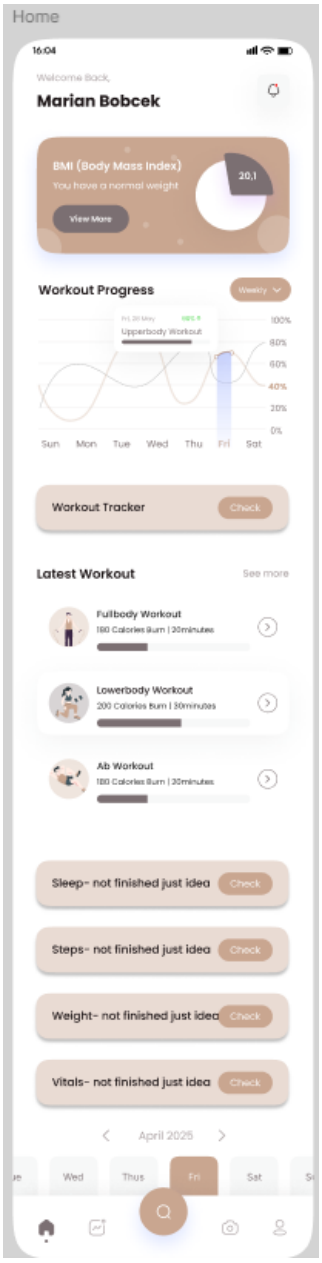
Hlavná obrazovka(Home) teda domovská obrazovka bude slúžiť ako hlavný prehľad.

Použité komponenty sú:

- Karty s cvičením
- Tlačidlá Check alebo teda celá karta kde sa tieto tlačidlá nachádzajú

Zobrazuje:

- BMI a jeho hodnotenie
- Týždenný progres v cvičení vo forme grafu
- Posledné cvičenia
- Ostatné sekcie ako spánok, kroky, hmotnosť (zatiaľ ako návrh)



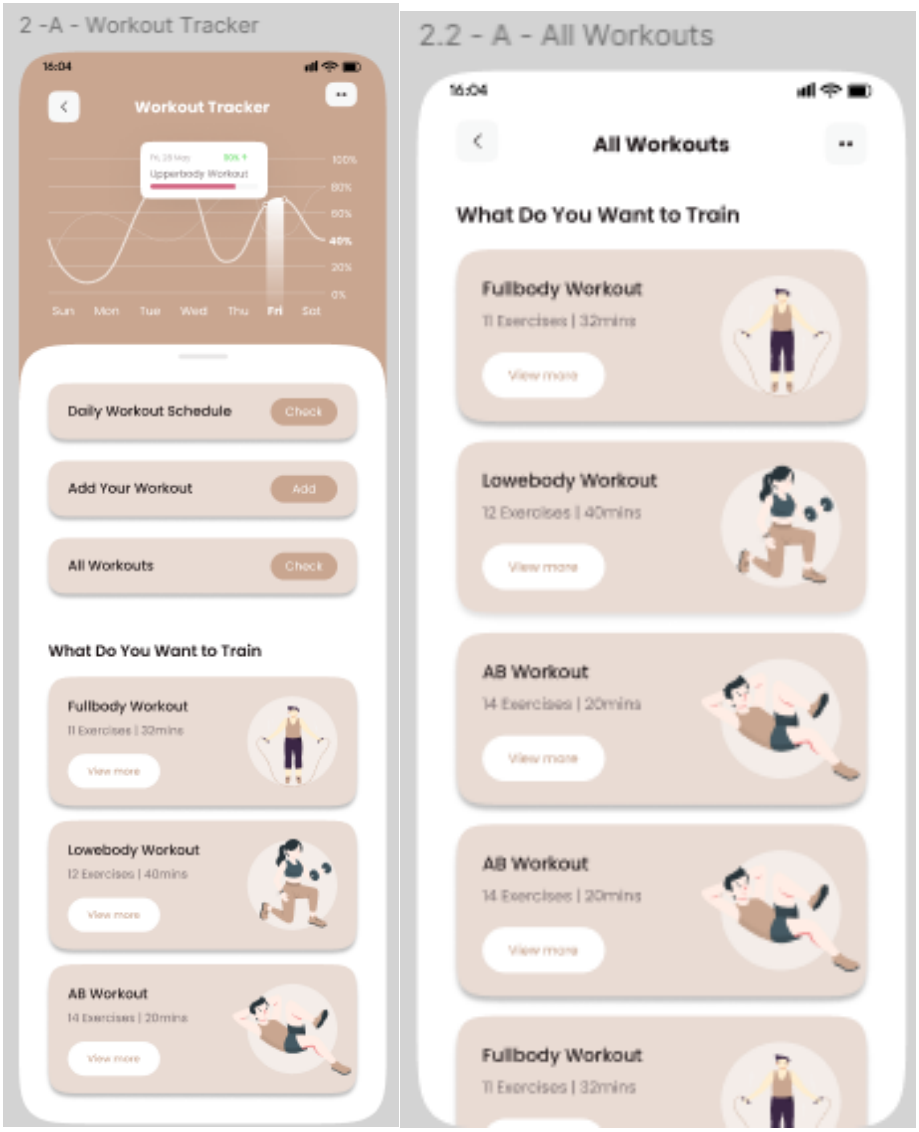
Obrazovka Workout Tracker a All Workouts

Zobrazuje možnosti ako

- Denný tréningový plán
- Pridanie vlastného cvičenia
- Všetky dostupné cvičenia

Použité komponenty sú:

- Karty s cvičením
- Tlačidlá Check,Add alebo teda celá karta kde sa tieto tlačidlá nachádzajú

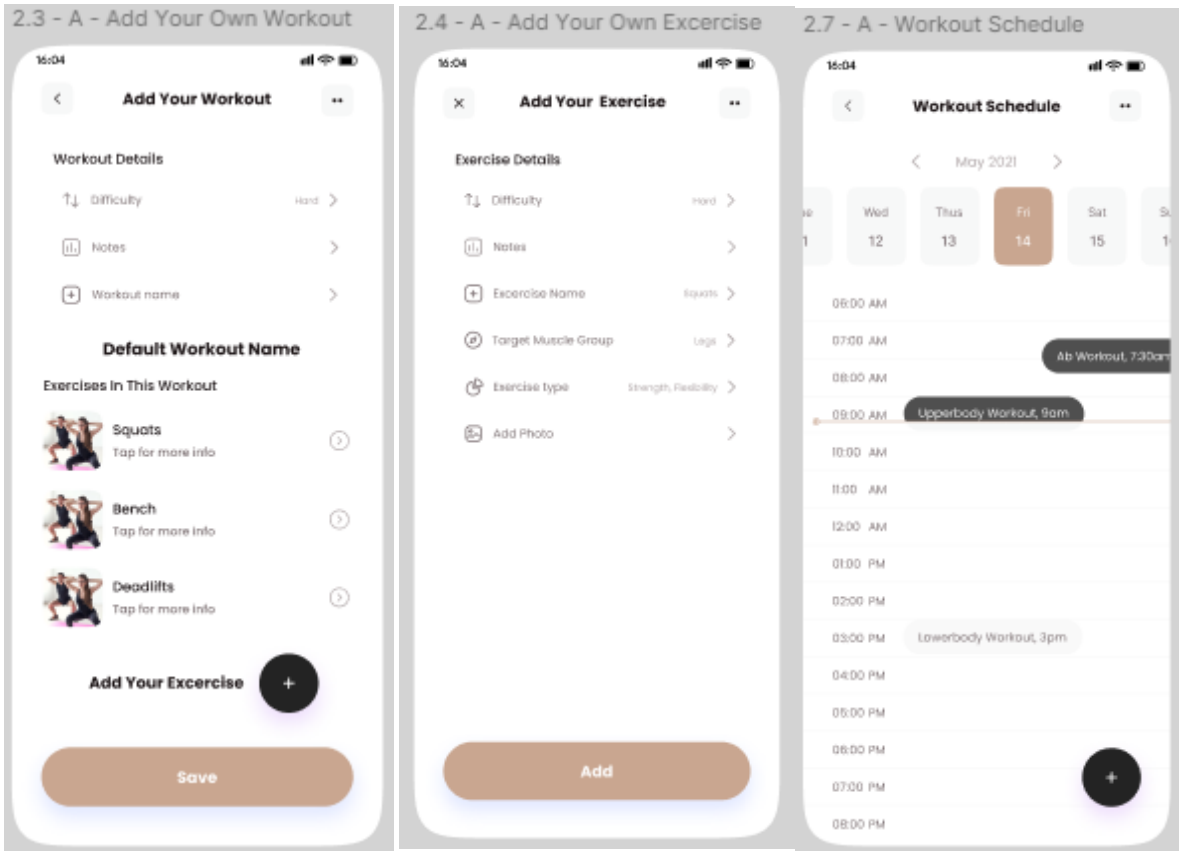


Obrazovky Add Your Own Workout, Add Your Own Excercise a Workout Schedule

Používateľ si môže vytvoriť vlastný workout, zadať jeho názov, úroveň obtiažnosti, a pridať jednotlivé cviky (napr. Squats, Bench, Deadlifts). Pridanie cviku s jeho typom a zameraním ako aj výberom obrázka. Kalendár kde si užívateľ môže prehliadať naplánované alebo hotové cvičenia.

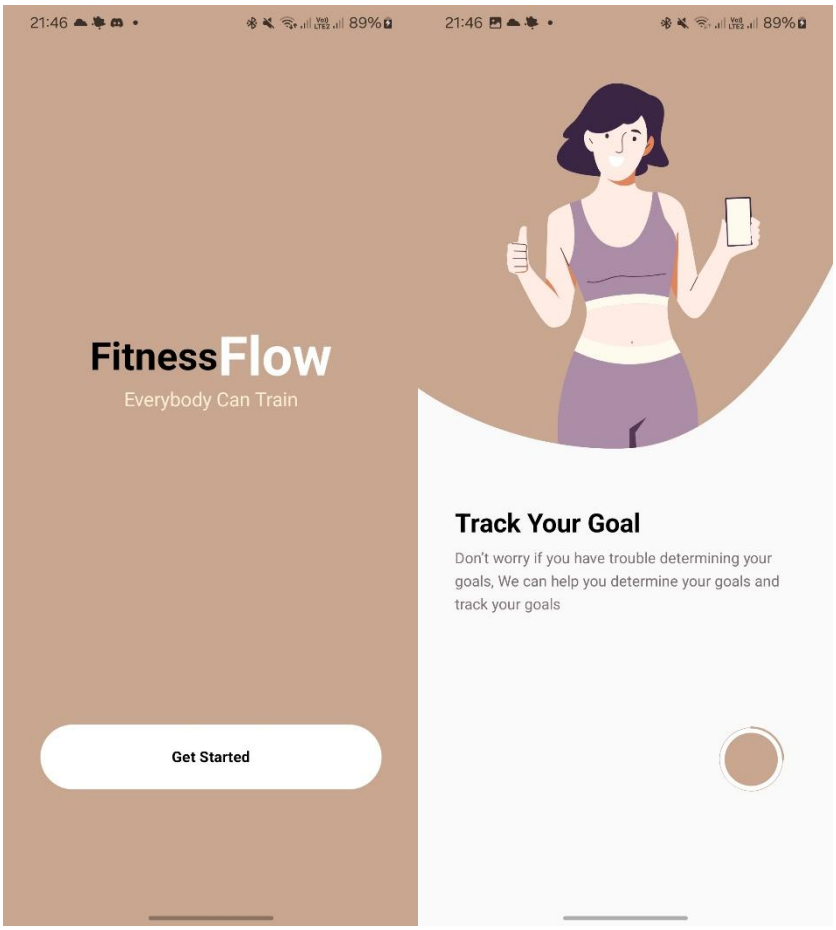
Použité komponenty sú:

- Karty s cvikmi
- Detaily cvičení a cviku
- Tlačidlá Save,Add a „+“

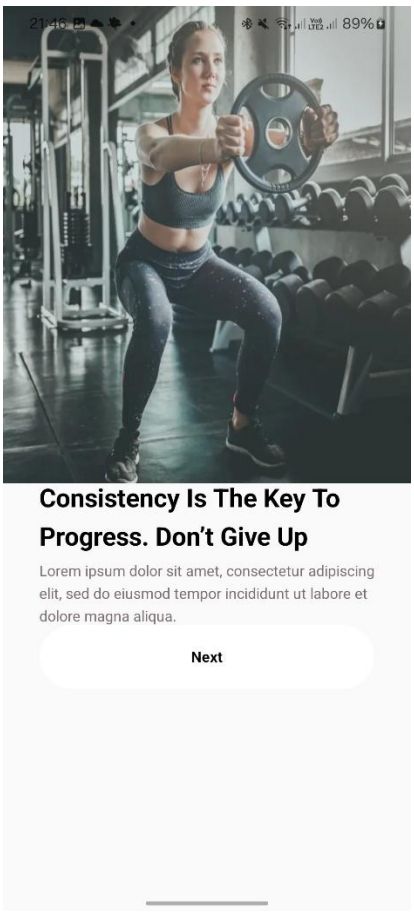


Skutočný návrh riešenia problému

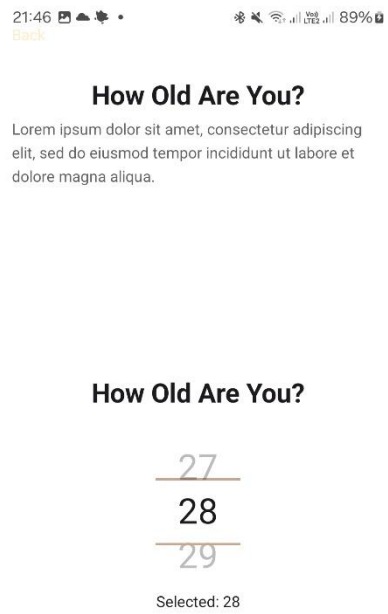
Ospravedlňujem sa podcenil som to a nestihol som tuto sekciu pridal som aspoň screenshoty obrazoviek


21:46 89%
FitnessFlow
Everybody Can Train
Get Started

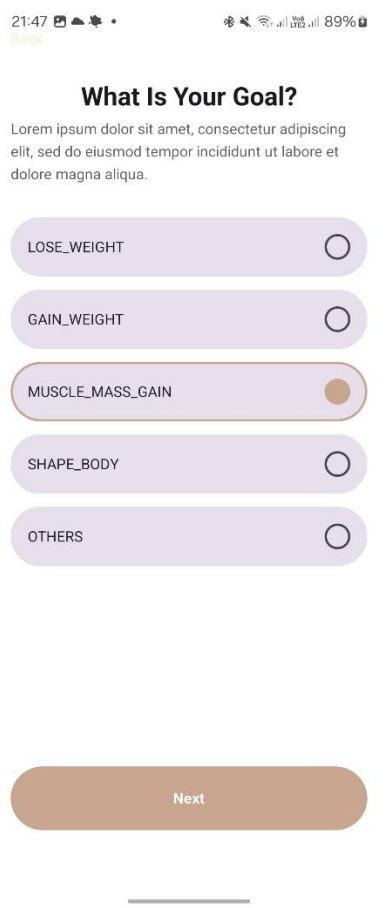
21:46 89%
Track Your Goal
Don't worry if you have trouble determining your goals, We can help you determine your goals and track your goals

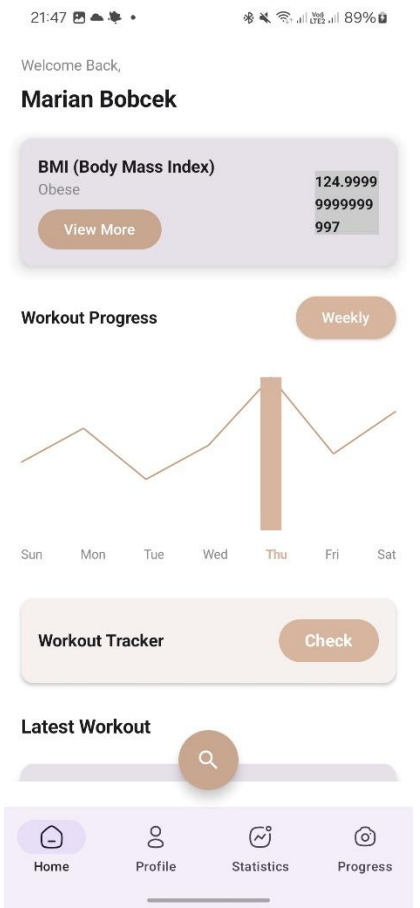

89%
Consistency Is The Key To Progress. Don't Give Up
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Next

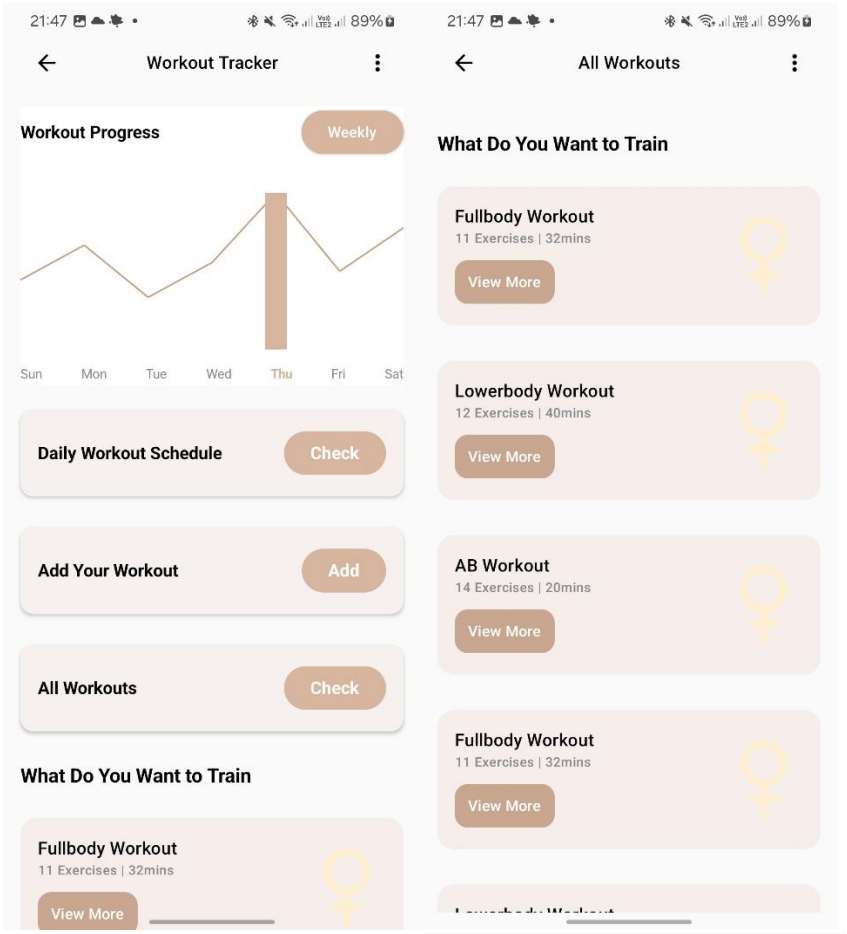

21:46 89%
Back
What's Your Gender
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Next


21:46
89%
Back
How Old Are You?
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
How Old Are You?
27
28
29
Selected: 28
Next

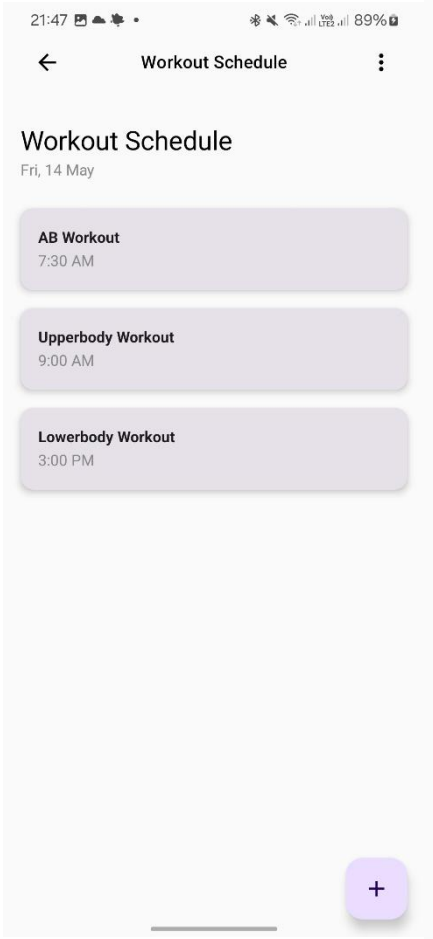

21:46
89%
Back
What Is Your Weight?
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
What Is Your Weight?
79
80
81
Selected: 80 kg
Next

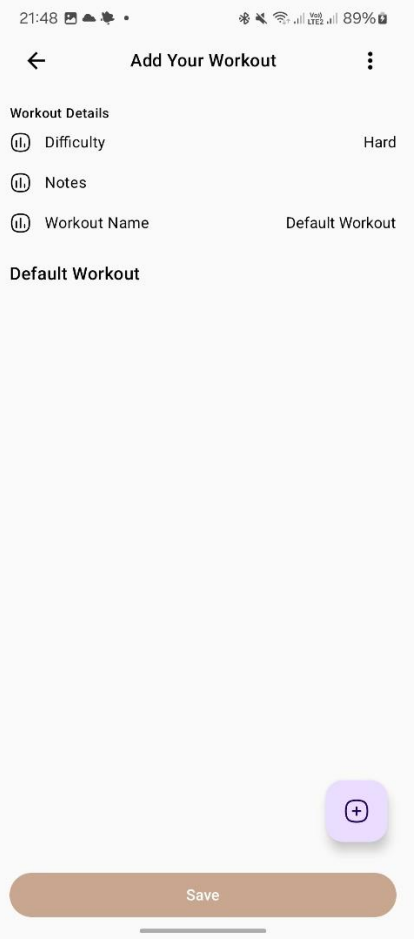

21:47
89%
Back
What Is Your Goal?
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
LOSE_WEIGHT
GAIN_WEIGHT
MUSCLE_MASS_GAIN
SHAPE_BODY
OTHERS
Next

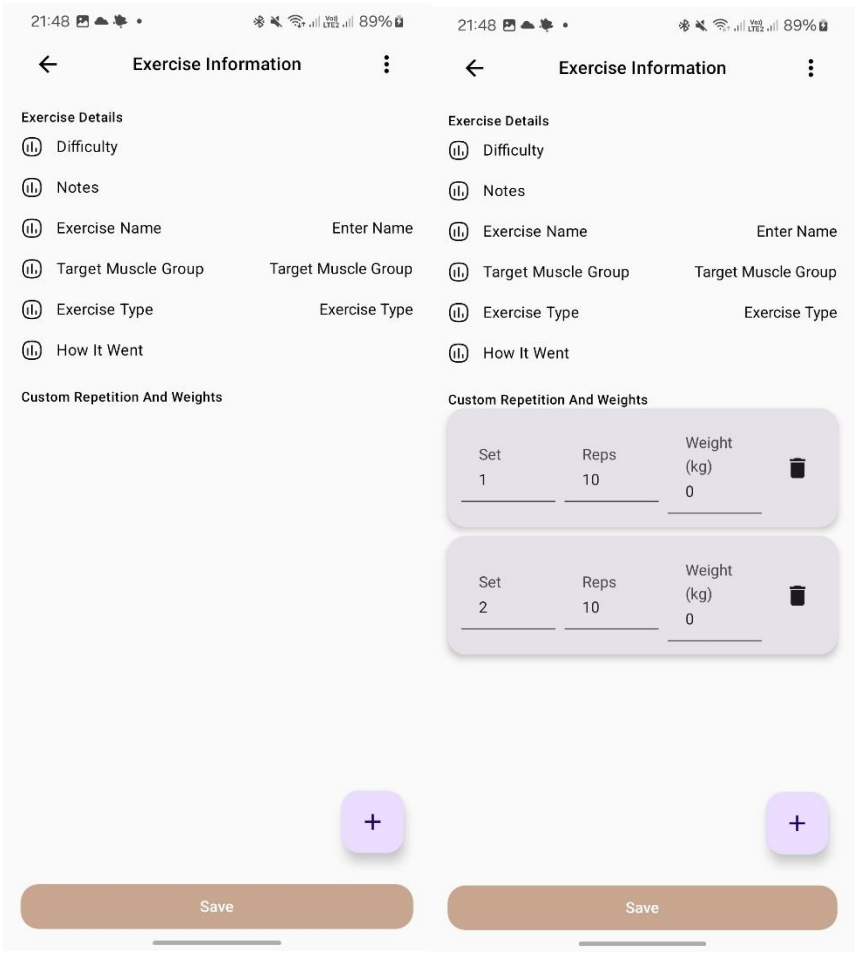

21:47
89%
Welcome Back,
Marian Bobcek
BMI (Body Mass Index)
Obese
View More
124.9999999999997
Workout Progress
Weekly
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Workout Tracker
Check
Latest Workout
Home
Profile
Statistics
Progress

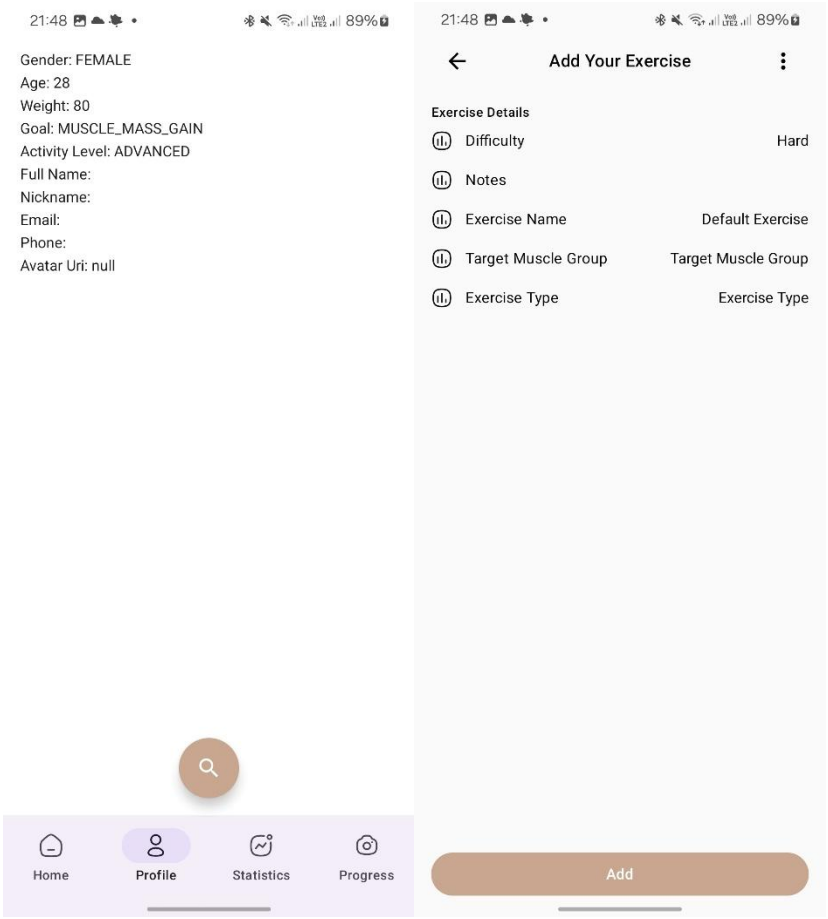

21:47 89%
Workout Tracker
Workout Progress
Weekly
Sun Mon Tue Wed Thu Fri Sat
Daily Workout Schedule
Check
Add Your Workout
Add
All Workouts
Check
What Do You Want to Train
Fullbody Workout
11 Exercises | 32mins
View More

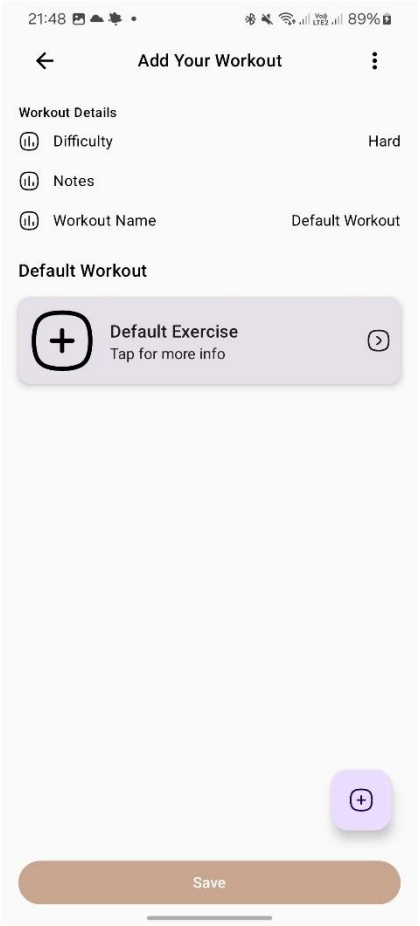
21:47 89%
All Workouts
What Do You Want to Train
Fullbody Workout
11 Exercises | 32mins
View More
Lowerbody Workout
12 Exercises | 40mins
View More
AB Workout
14 Exercises | 20mins
View More
Fullbody Workout
11 Exercises | 32mins
View More


21:47 89%
Workout Schedule
Workout Schedule
Fri, 14 May
AB Workout
7:30 AM
Upperbody Workout
9:00 AM
Lowerbody Workout
3:00 PM


21:48 89%
Add Your Workout
Workout Details
Difficulty
Hard
Notes
Workout Name
Default Workout
Default Workout
Save


21:48 89%
Exercise Information
Exercise Details
Difficulty
Notes
Exercise Name
Enter Name
Target Muscle Group
Target Muscle Group
Exercise Type
Exercise Type
How It Went
Custom Repetition And Weights
Save
21:48 89%
Exercise Information
Exercise Details
Difficulty
Notes
Exercise Name
Enter Name
Target Muscle Group
Target Muscle Group
Exercise Type
Exercise Type
How It Went
Custom Repetition And Weights
Set
1
Reps
10
Weight
(kg)
0
Set
2
Reps
10
Weight
(kg)
0
Save


21:48 89%
Gender: FEMALE
Age: 28
Weight: 80
Goal: MUSCLE_MASS_GAIN
Activity Level: ADVANCED
Full Name:
Nickname:
Email:
Phone:
Avatar Uri: null
Home
Profile
Statistics
Progress
21:48 89%
Add Your Exercise
Exercise Details
Difficulty
Hard
Notes
Exercise Name
Default Exercise
Target Muscle Group
Target Muscle Group
Exercise Type
Exercise Type
Add


21:48
89%
Add Your Workout
Workout Details
Difficulty
Hard
Notes
Workout Name
Default Workout
Default Workout
Default Exercise
Tap for more info
Save

Obrazovky

Popis implementácie

```
dependencies {
    implementation("androidx.navigation:navigation-compose:2.9.0")

    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.lifecycle.runtime.ktx)
    implementation(libs.androidx.activity.compose)
    implementation(platform(libs.androidx.compose.bom))
    implementation(libs.androidx.ui)
    implementation(libs.androidx.ui.graphics)
    implementation(libs.androidx.ui.tooling.preview)
    implementation(libs.androidx.material3)
    implementation(libs.androidx.navigation.runtime.android)
    implementation(libs.androidx.navigation.compose)
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
    androidTestImplementation(platform(libs.androidx.compose.bom))
    androidTestImplementation(libs.androidx.ui.test.junit4)
    debugImplementation(libs.androidx.ui.tooling)
    debugImplementation(libs.androidx.ui.test.manifest)
    //view model
    implementation("androidx.lifecycle:lifecycle-viewmodel-compose:2.9.0")
    //room
    implementation("androidx.room:room-runtime:${rootProject.extra["room_version"]}"
    ksp("androidx.room:room-compiler:${rootProject.extra["room_version"]}")
    implementation("androidx.room:room-ktx:${rootProject.extra["room_version"]}")
    //number picker
    implementation("com.chargemap.compose:numberpicker:1.0.3")
}
```

Použitie externého frameworku / knižnice

Názov knižnice:

com.chargemap.compose:numberpicker:1.0.3

Na obrazovkách HeightScreen, WeightScreen a AgeScreen som využil externý framework NumberPicker z knižnice com.chargemap.compose:numberpicker

```
NumberPicker(
    value = height,
    range = 50 ≤ .. ≤ 220,
    onValueChange = onHeightChanged,
    dividersColor = MaterialTheme.colorScheme.primary,
    textStyle = MaterialTheme.typography.headlineLarge.copy(
        color = MaterialTheme.colorScheme.onBackground
    )
)
```

Využitie *AndroidX* komponentov

Navigation

Navigácia v aplikácii FitnessFlow je riešená pomocou komponentu androidx.navigation:navigation-compose. Každá obrazovka je registrovaná cez composable funkciu a prechody medzi nimi sú riadené pomocou NavHostController.

```
@Composable
fun AppNavGraph(
    navController: NavHostController,
    startDestination: String
) {
    val context = LocalContext.current
    val setupVm: SetupViewModel = viewModel()
    val workoutVm : WorkoutCreationViewModel = viewModel()
    val uiSetupState by setupVm.uiState.collectAsState()

    NavHost(
        navController = navController,
        startDestination = startDestination
    ) {
        composable(Screen.Welcome.route) {
            WelcomeScreen(
                onGetStartedClick = {
                    //navController.navigate(Screen.Setup.route) //na
                    navController.navigate(Screen.Onboarding.route)
                }
            )
        }
```

ViewModel

WorkoutCreationViewModel, ProfileViewModel, SetupViewModel

Trieda SetupViewModel využíva komponent AndroidViewModel na správu dát súvisiacich s nastavením používateľského profilu. Pomocou StateFlow uchováva a aktualizuje aktuálny stav používateľa počas úvodného onboarding procesu.

```
class SetupViewModel(
    app: Application
) : AndroidViewModel(app) {
    private val repository = UserProfileRepository.getInstance(app)

    private val _uiState = MutableStateFlow(SetupUiState())
    val uiState: StateFlow<SetupUiState> = _uiState.asStateFlow()

    val formState: StateFlow<ProfileFormState> = uiState
        .map { s ->
            ProfileFormState(
                avatarUri = s.avatarUri?.let(Uri::parse),
                fullName = s.fullName,
                nickname = s.nickname,
                email = s.email,
                mobileNumber = s.mobileNumber
            )
        }
        .stateIn(viewModelScope, SharingStarted.Eagerly, ProfileFormState(null, "", "", "",

    fun updateProfileField(field: ProfileField, value: String) {
        _uiState.value = when (field) {
            ProfileField.FullName -> _uiState.value.copy(fullName = value)
            ProfileField.Nickname -> _uiState.value.copy(nickname = value)
            ProfileField.Email -> _uiState.value.copy(email = value)
            ProfileField.Phone -> _uiState.value.copy(mobileNumber = value)
        }
    }
```

```
fun ProfileSetupRoute(
    navController: NavHostController,
    vm : SetupViewModel
) {
    val form by vm.formState.collectAsState()
    val context = LocalContext.current
    val page = SetupPages.getPages(context)[SetupStep.Profile.pageIndex]

    //vyber obrazka z galerie
    val pickImage = rememberLauncherForActivityResult(
        contract = ActivityResultContracts.GetContent()
    ) { uri: Uri? ->
        uri?.let { vm.updateAvatarUri(it) }
    }

    FillYourProfileScreen(
        title = page.title,
        description = page.description,
        form = form,
        onEditPictureClick = { pickImage.launch("image/*") },
        onFieldChanged = { field, value ->
            vm.updateProfileField(field, value)
        },
        onBack = { navController.popBackStack() },
        onNext = {
            vm.saveAllAndFinish()
            navController.navigate(Screen.Home.route) {
                popUpTo(SetupStep.Setup.route) { inclusive = true }
            }
        }
    )
}
```

```
class WorkoutCreationViewModel(
    private val app: Application
) : AndroidViewModel(app) {

    private val repository = WorkoutRepository.getInstance(app)

    val allWorkouts: StateFlow<List<Workout>> = repository.getAllWorkouts()
        .stateIn(viewModelScope, SharingStarted.Lazily, emptyList())


    var workoutName by mutableStateOf(app.getString(R.string.default_workout_name))
        private set

    var workoutDifficulty by mutableStateOf(app.getString(R.string.difficulty_hard))
        private set

    var workoutNotes by mutableStateOf("")
        private set

    var exerciseName by mutableStateOf(app.getString(R.string.default_exercise_name))
        private set

    var exerciseDifficulty by mutableStateOf(app.getString(R.string.difficulty_hard))
        private set

    var muscleGroup by mutableStateOf("")
        private set

    var exerciseType by mutableStateOf("")
```

Room

Aplikácia využíva Room databázu ako lokálne úložisko na správu profilu, tréningov a cvikov. Všetky CRUD operácie (create, read, update, delete) sú implementované cez DAO rozhrania. Room databáza je navrhnutá ako singleton a využíva TypeConverters pre správu enum hodnôt v entitách.

```
fun saveAllAndFinish() = viewModelScope.launch {
    val profileValues = _uiState.value
    val profile = UserProfile(
        id = 0,
        gender = profileValues.gender,
        age = profileValues.age,
        weight = profileValues.weight,
        height = profileValues.height,
        goal = profileValues.goal,
        activityLevel = profileValues.activityLevel,
        avatarUri = profileValues.avatarUri,
        fullName = profileValues.fullName,
        nickname = profileValues.nickname,
        email = profileValues.email,
        phone = profileValues.mobileNumber
    )
    Log.d("DEBUG_PROFILE", "gender=${profileValues.gender}, goal=${profileValues.goal},
    repository.saveProfile(profile)
```

```
class UserProfileRepository private constructor(
    private val dao: UserProfileDao
) {
    suspend fun saveProfile(profile: UserProfile) = dao.insertProfile(profile)
    suspend fun getProfile(): UserProfile?    = dao.getProfile()

    companion object {
        @Volatile private var INSTANCE: UserProfileRepository? = null
        /**
         * singleton implementacia repozitara - vrati existujucu alebo vytvori novu instanciu
         *
         * @param context  ziskanie pristupu k databaze
         * @return singleton instancia
         */
        fun getInstance(context: Context): UserProfileRepository =
            INSTANCE ?: synchronized(this) {
                INSTANCE ?: UserProfileRepository(
                    AppDatabase.getInstance(context).userProfileDao()
                ).also { INSTANCE = it }
            }
    }
}
```

```
interface UserProfileDao {
    @Query("SELECT * FROM user_profile WHERE id = 1")
    suspend fun getProfile(): UserProfile?

    @Query("DELETE FROM user_profile WHERE id = 1")
    suspend fun deleteProfile(): Int

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insertProfile(profile: UserProfile)

    //val updatedProfile = currentProfile.copy(age = 25, weight = 80)
    //userProfileDao.updateProfile(updatedProfile)
    //pozor na to aby ostali ostatne hodnoty ulozene nech si neprepisem celu databazu na null
    @Update
    suspend fun updateProfile(profile: UserProfile)
```

LifeCycle

V aplikácii je Lifecycle automaticky využitý cez ViewModel a Compose (collectAsState),

```
private val _userProfile = mutableStateOf<UserProfile?>(null)
val userProfile: State<UserProfile?> = _userProfile

init {
    viewModelScope.launch {
        _userProfile.value = userProfileDao.getProfile()
    }
}
```

```
    val context = LocalContext.current
    val setupVm: SetupViewModel = viewModel()
    val workoutVm : WorkoutCreationViewModel = viewModel()
    val uiSetupState by setupVm.uiState.collectAsState()

    NavHost(
        navController = navController,
        startDestination = startDestination
    ) {
        composable(Screen.Welcome.route) {
```

Zoznam zdrojov

https://www.figma.com/community/website-templates?resource_type=mixed&editor_type=figma&price=all&sort_by=all_time&creators=all

https://www.figma.com/templates/

https://elements.envato.com/graphic-templates/compatible-with-figma

https://piqodesign.gumroad.com/l/iconly

https://coolors.co/

https://www.youtube.com/watch?v=xsg9BDiwiJE

https://www.youtube.com/watch?v=hktyW5Lp0Vo

https://www.youtube.com/watch?v=OOpCN9le6ok

https://app.smartdraw.com/

https://www.lucidchart.com/pages/er-diagrams

https://developer.android.com/codelabs/jetpack-compose-navigation8

https://developer.android.com/develop/ui/compose/navigation

https://developer.android.com/codelabs/basic-android-kotlin-compose-persisting-data-room#0

https://developer.android.com/codelabs/basic-android-kotlin-compose-update-data-room#0

https://developer.android.com/codelabs/android-room-with-a-view-kotlin#7

https://developer.android.com/training/data-storage/room/relationships

https://developer.android.com/reference/androidx/room/Entity

https://developer.android.com/codelabs/basic-android-kotlin-compose-coroutines-android-studio#5

https://developer.android.com/topic/libraries/architecture/viewmodel

https://developer.android.com/codelabs/basic-android-kotlin-compose-navigation?hl=en#0

https://github.com/google-developer-training/basic-android-kotlin-compose-training-lunch-tray/tree/main

https://developer.android.com/codelabs/basic-android-kotlin-compose-viewmodel-and-state#10

https://developer.android.com/codelabs/basic-android-kotlin-compose-using-state#10