



ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Fakulta riadenia a informatiky

Gym App

Semestrálna práca z VAMZ

Frederik Seko

Študijný odbor: Informatika a riadenie

Vedúci: doc. Ing. Patrik Hrkút, PhD.

Žilina 2024

Obsah

1	Popis a analýza riešeného problému.....	3
1.1	Špecifikácia zadania, definovanie problému	3
1.2	Podobné aplikácie	3
1.3	Rozdiely mojej aplikácie	4
2	Návrh riešenia problému	5
2.1	Návrh aplikácie	5
2.2	Krátka analýza.....	6
3	Popis implementácie	8
4	Zoznam použitých zdrojov	13
	Záver.....	14

1 POPIS A ANALÝZA RIEŠENÉHO PROBLÉMU

1.1 Špecifikácia zadania, definovanie problému

V dnešnej dobe sa ľudia často venujú cvičeniu z rôznych dôvodov, ako sú zlepšenie kondície, udržanie zdravej postavy alebo odreagovanie. Aplikácia, na ktorej som pracoval, je zameraná na výpomoc pri tréningu, špeciálne pre začiatočníkov v oblasti fitness. Cieľom tejto aplikácie je poskytnúť užívateľom stručného a efektívneho sprievodcu cvičením, ktorý ich naučí správne techniky a postupy pri cvičení na celé telo. Taktiež si ľudia môžu ukladať svoje progresy, a poznámky.

Aplikácia sa sústreďí na rozdelenie svalových skupín, tzv. split, ktoré umožnia užívateľom efektívne rozvíjať svalovú hmotu a celkovú kondíciu. Každý tréningový deň je zameraný na konkrétnu svalovú skupinu alebo ich kombináciu pre optimálne výsledky.

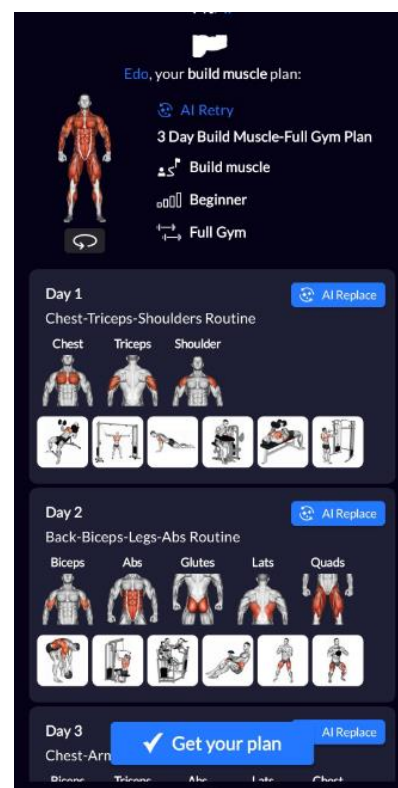
1.2 Podobné aplikácie

FitAI

Mobilná aplikácia, ktorá využíva umelú inteligencia na nahradenie fitness trénera. FitAI poskytuje odporúčania pre rôzne cviky a tréningové plány na základe analýzy údajov používateľa. Je dostupná vo forme plateného modelu, avšak umožňuje bezplatné vyskúšanie.

Funkcie a vlastnosti:

- Odporúčania cvičení založené na analýze umelej inteligencie.
- Zobrazenie videí s názorným sprievodom pre správne vykonávanie cvikov.
- Možnosť personalizovaných tréningových plánov na základe cieľov používateľa.

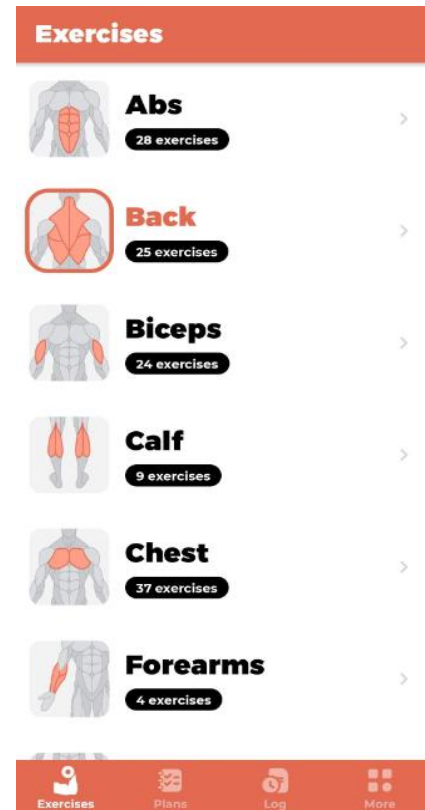


F&B

F&B je mobilná aplikácia zameraná na ukážku cvičení rôznych častí tela. Poskytuje detailné návody a vizuálnu podporu prostredníctvom fotografií, ktoré ukazujú správne techniky cvičení.

Funkcie a vlastnosti:

- Rozsiahla knižnica cvikov z rôznych oblastí fitness.
- Podrobné fotografie a popisy cvikov s správnou technikou.
- Zobrazenie svalových partií zapojených pri každom cviku.
- Možnosť vyberať cviky podľa cieľov alebo konkrétnych oblastí tela.



1.3 Rozdiely mojej aplikácie

Moja aplikácia je zameraná špeciálne na začiatočníkov vo fitness. Poskytuje detailné návody a vizuálnu podporu prostredníctvom fotografií a popisov, ktoré ukazujú správne techniky cvičení. Na rozdiel od FitAI moja aplikácia nevyužíva umelú inteligenciu, ale ponúka používateľom priamy prístup k tréningovým plánom a cvikom a môžu si vybrať cvik, ktorý im pasuje, pretože ku každej časti tela som sa snažil vybrať minimálne 2 cviky. Na rozdiel od F&B moja aplikácia umožňuje používateľom sledovať svoj pokrok tým, že si budú zapisovať vlastné poznámky a pridávať si ich, pričom sa im tieto poznámky ukladajú. Ďalšou výhodou tejto aplikácie je, že je zadarmo a používatelia majú plný prístup k celej aplikácii, čo pri spomínaných aplikáciách neplatí.

2 NÁVRH RIEŠENIA PROBLÉMU

2.1 Návrh aplikácie

Popis tried a ich vzťahov:

MainActivity: Hlavná aktivita aplikácie, ktorá inicializuje obsah a spravuje navigáciu medzi obrazovkami. Obsahuje metódy na vytvorenie notifikačného kanála a žiadosť o povolenie na notifikácie, taktiež sa tu nachádza WelcomeScreen.

NotifikacnyPomocnik: Pomocná trieda na správu notifikácií. Obsahuje metódy na vytvorenie notifikačného kanála a odosielanie notifikácií.

RegistraciaScreen: Composable funkcia, ktorá zobrazuje registračnú obrazovku. Obsahuje kroky registrácie, ako je zadávanie osobných údajov, stanovenie cieľov, výber kondičnej úrovne a úvodné testy.

PokrokScreen: Composable funkcia, ktorá zobrazuje obrazovku pokroku, kde používateľ môže pridávať nové tréningové záznamy a prezerať si predchádzajúce záznamy.

RegistraciaViewModel: ViewModel, ktorý spravuje stav registrácie a počiatočných testov. Obsahuje MutableStateFlow premenné na ukladanie osobných údajov a výsledkov testov.

PokrokViewModel: ViewModel, ktorý spravuje stav obrazovky pokroku. Obsahuje MutableStateFlow premenné na ukladanie aktuálnych vstupných údajov a zoznam tréningových záznamov.

TreningovyZaznam: Trieda reprezentujúca tréningový záznam, ktorá obsahuje dátum, hmotnosť, počet opakovaní a poznámky.

ExerciseScreen: Composable funkcia, ktorá zobrazuje zoznam cvičení pre rôzne časti tela (PushScreen, PullScreen, Legs_Abs_Screen).

2.2 Krátka analýza

UseCase diagram zobrazuje hlavné funkcionality aplikácie Gym App interakcie medzi používateľom a systémom. Diagram sa zameriava na tieto hlavné úlohy:

Registrácia používateľa: Používateľ sa zaregistruje v systéme, kde zadá svoje osobné údaje (Nastavenie osobných údajov).

Nastavenie cieľov: Používateľ si stanoví tréningové ciele, ktoré chce dosiahnuť.

Výber kondičnej úrovne: Používateľ vyberie svoju aktuálnu kondičnú úroveň.

Vykonávanie úvodných testov: Používateľ vykoná úvodné testy, ktoré pomôžu pri nastavení tréningového plánu.

Zobrazenie domovskej obrazovky: Po úspešnej registrácii alebo prihlásení sa používateľovi zobrazí domovská obrazovka aplikácie.

Zobrazenie cvičení: Používateľ si môže prezerať rôzne cvičenia dostupné v aplikácii.

Sledovanie pokroku: Používateľ sleduje svoj tréningový pokrok a pridáva nové tréningové záznamy.

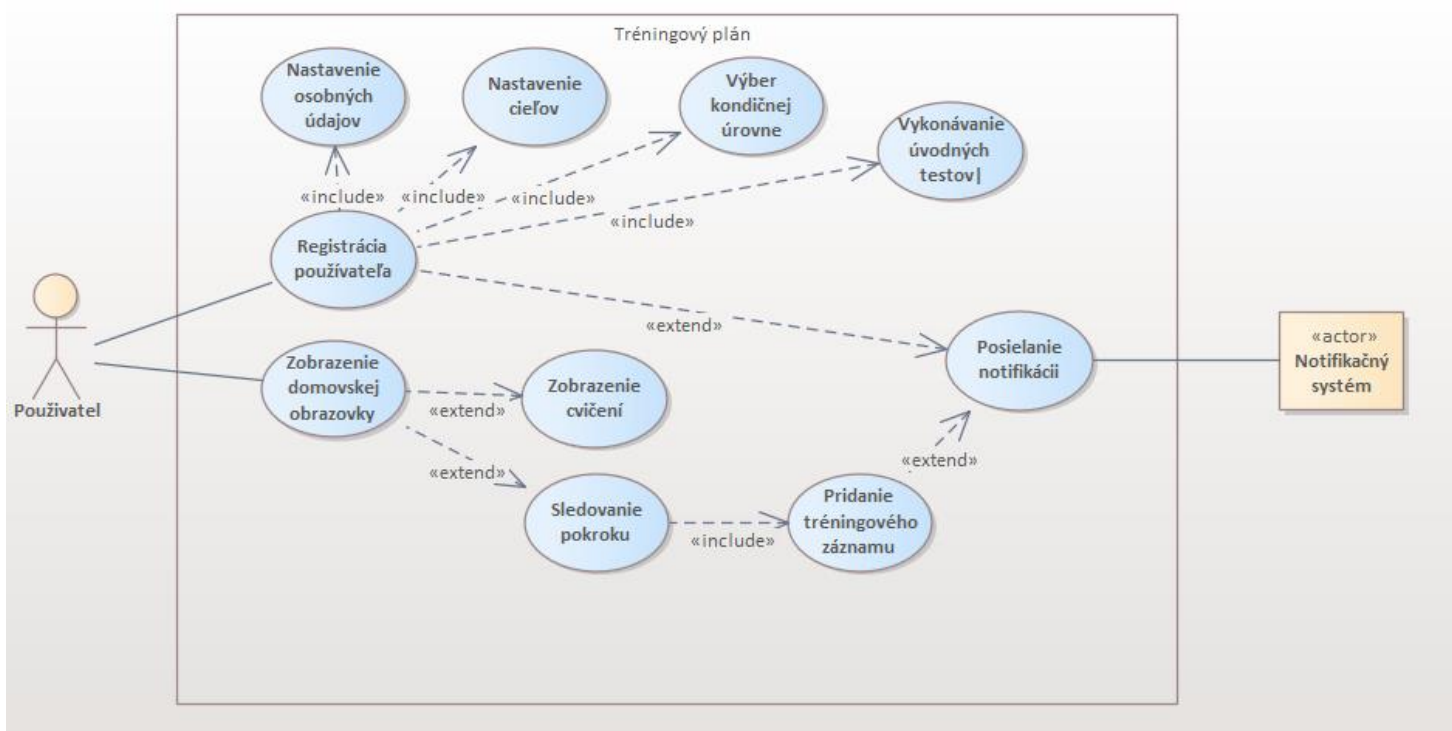
Pridanie tréningového záznamu: Používateľ môže pridávať nové tréningové záznamy.

Posielanie notifikácií: Notifikačný systém posielá používateľovi notifikácie, napríklad po úspešnej registrácii alebo pridaní tréningového záznamu.

Aktéri:

Používateľ: Hlavný aktér, ktorý interaguje so systémom.

Notifikačný systém: Systém, ktorý posielá notifikácie používateľovi.



3 POPIS IMPLEMENTÁCIE

Reakcia na otočenie displeja: Aplikácia je navrhnutá tak, aby správne reagovala na otočenie displeja z polohy na výšku na polohu na šírku a naopak.

Použitie **ViewModel** tried (RegistraciaViewModel a PokrokViewModel) zabezpečuje uchovávanie stavu aktivity pred otočením, čím sa zabezpečí, že všetky formulárové polia zostanú vyplnené aj po otočení displeja. Okrem toho na všetky screeny je použitý LazyColumn na posúvanie obrazovky.

```
/**
 * RegistraciaViewModel je zodpovedný za udržiavanie stavu registrácie a počiatočných testov.
 * A taktiež pri otočení obrazovky, aby vyplnené údaje zostali vyplnené
 */
new *
class RegistraciaViewModel : ViewModel() {
    private val _meno = MutableStateFlow( value: "" )
    new *
    val meno: StateFlow<String> get() = _meno

    private val _vyska = MutableStateFlow( value: "" )
    new *
    val vyska: StateFlow<String> get() = _vyska
}
```

Zdroje ako resources: Obrázky, farby a ďalšie multimédiá sú v projekte uvedené ako resources. Napríklad obrázky sú uložené v priečinku *res/drawable* a farby sú uložené v *colors.xml*

Obrazovky aplikácie: Aplikácia obsahuje viacero unikátnych obrazoviek, ktoré riešia rôzne funkcie aplikácie:

1. Registrácia používateľa:

RegistraciaScreen zahŕňa kroky registrácie: osobné údaje, cieľe, kondičná úroveň, úvodné testy. A taktiež obsahuje aj ViewModel

```
@Composable
fun RegistraciaScreen(navController: NavController, sharedPreferences: SharedPreferences, viewModel: RegistraciaViewModel) {
    var currentStep by rememberSaveable { mutableStateOf( value: 0 ) }

    val steps = listOf<@Composable () -> Unit>{
        { OsobneUdajeScreen(onNext = { currentStep++ }, sharedPreferences = sharedPreferences, viewModel = viewModel) },
        { FitnessScreen(onNext = { currentStep++ }) },
        { CieleScreen(onNext = { currentStep++ }) },
        {
            TestScreen(onNext = {
                sharedPreferences.edit().putBoolean("isFirstRun", false).apply()
                navController.navigate( route: "HomeScreen" )
                (navController.context as? MainActivity)?.ziadajPovolenieNaNotifikacie()
            }, sharedPreferences = sharedPreferences, viewModel = viewModel)
        }
    }

    Surface(
        color = colorResource(id = R.color.background_color),
        modifier = Modifier.fillMaxSize()
    )
}
```


2. Domovská obrazovka:

WelcomeScreen zobrazuje uvítací obrázok a tlačidlo na spustenie aplikácie, táto obrazovka sa spustí až po druhom otvorení aplikácie, keď sa užívateľ najskôr nastaví osobné údaje.

```
@Composable
fun WelcomeScreen(navController: NavController) {
    Surface(
        color = colorResource(id = R.color.background_color),
    ){
        LazyColumn(
            modifier = Modifier.fillMaxSize().padding(10.dp),
            verticalArrangement = Arrangement.Top,
            horizontalAlignment = Alignment.CenterHorizontally
        ) { this: LazyListScope
            item { this: LazyItemScope
                Image(
                    painter = painterResource(id = R.drawable.uvodna),
                    contentDescription = null,
                    modifier = Modifier
                        .fillMaxWidth()
                        .padding(10.dp)
                )
            }
        }
    }
}
```

3. Cvičenia:

Aplikácia je navrhnutá tak, aby poskytovala užívateľom možnosť vybrať si z rôznych cvičení a kategórií. Hlavná obrazovka **HomeScreen** umožňuje užívateľom vybrať si z nasledujúcich kategórií: **Pull**, **Push** a **Legs+Abs**. Po výbere kategórie sa užívatelia dostanú na príslušné obrazovky, ktoré sú rozdelené podľa konkrétnych častí tela. Tieto obrazovky sú nasledovné: PushScreen, PullScreen, Leg_Abs_Screen.

```
@Composable
fun HomeScreen(navController: NavHostController) {
    Surface(
        modifier = Modifier.fillMaxSize(),
        color = colorResource(id = R.color.background_color),
    ) {
        LazyColumn(
            modifier = Modifier.fillMaxSize().padding(10.dp),
            verticalArrangement = Arrangement.Top,
            horizontalAlignment = Alignment.CenterHorizontally
        ) { this: LazyListScope
            item { SplitButton(navController, route: "PushScreen", R.drawable.push, text: "Push") }
            item { SplitButton(navController, route: "PullScreen", R.drawable.pull, text: "Pull") }
            item { SplitButton(navController, route: "Legs_Abs_Screen", R.drawable.legs, text: "Legs + Abs") }
        }
    }
}
```

Každá z týchto obrazoviek obsahuje ďalšie podsekcie pre konkrétne svalové partie, ktoré sú prezentované pomocou karty (**Card**). Po kliknutí na lištu karty sa zobrazí podrobný popis jednotlivých cvikov.

Na odstránenie duplikácií som vytvoril **funkcie SplitButton** a **ExerciseScreen**, ktoré zjednodušujú a zlepšujú čitateľnosť kódu.

```
@Composable
fun SplitButton(navController: NavHostController, route: String, imageRes: Int, text: String) {
    Column(
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Image(imageRes, contentDescription = text)
        Text(text)
    }
}

@Composable
fun ExerciseScreen(navController: NavController, exercises: List<CastiTela>) {
    Surface(
        modifier = Modifier.fillMaxSize(),
        color = colorResource(id = R.color.background_color)
    ) {
        // ...
    }
}
```

4. Sledovanie pokroku:

PokrokScreen zobrazuje obrazovku na sledovanie pokroku a pridávanie nových tréningových záznamov. Záznam sa pridá, iba keď sú vyplnené všetky údaje. Záznam môžem aj vymazať pomocou *onLongClick*. Záznamy sa ukladajú a potom načítajú pri opätovnom spustení aplikácie pomocou *LaunchedEffect*.

Gson: Použitý na serializáciu a deserializáciu TrainingRecord objektov do/z SharedPreferences.

```
@Composable
fun PokrokScreen(navController: NavController, sharedPreferences: SharedPreferences, viewModel: PokrokViewModel) {
    val meno = sharedPreferences.getString(key = "meno", defValue = "Užívateľ") ?: "Užívateľ"
    val hmotnost by viewModel.hmotnost.collectAsState()
    val pocetOpakovani by viewModel.pocetOpakovani.collectAsState()
    val poznamka by viewModel.poznamka.collectAsState()
    val treningoveZaznamy by viewModel.treningoveZaznamy.collectAsState()
    var ukazVstupnyTest by remember { mutableStateOf(value = false) }
    var showValidationError by remember { mutableStateOf(value = false) }
    // ...
}
```

Využitie AndroidX komponentov:

ViewModel: Použitý pre správu stavu obrazoviek, napr.

RegistraciaViewModel, **PokrokViewModel**.

Navigation: Použitý na navigáciu medzi obrazovkami aplikácie.

```
setContent {
    Semestralka_VamZTheme {
        val navController = rememberNavController()
        val sharedPreferences = getSharedPreferences(name = "app_preferences", Context.MODE_PRIVATE)
        val isFirstRun = sharedPreferences.getBoolean(key = "isFirstRun", defValue = true)

        Surface(
            modifier = Modifier.fillMaxSize(),
            color = MaterialTheme.colorScheme.background
        ) {
            NavHost(navController, startDestination = if (isFirstRun) "RegistraciaScreen" else "WelcomeScreen") {
                composable(route = "WelcomeScreen") { WelcomeScreen(navController) }
                composable(route = "HomeScreen") { HomeScreen(navController) }
                composable(route = "PushScreen") { PushScreen(navController) }
                composable(route = "PullScreen") { PullScreen(navController) }
                composable(route = "Less Abs Screen") { LessAbsScreen(navController) }
            }
        }
    }
}
```

Notifikácie:

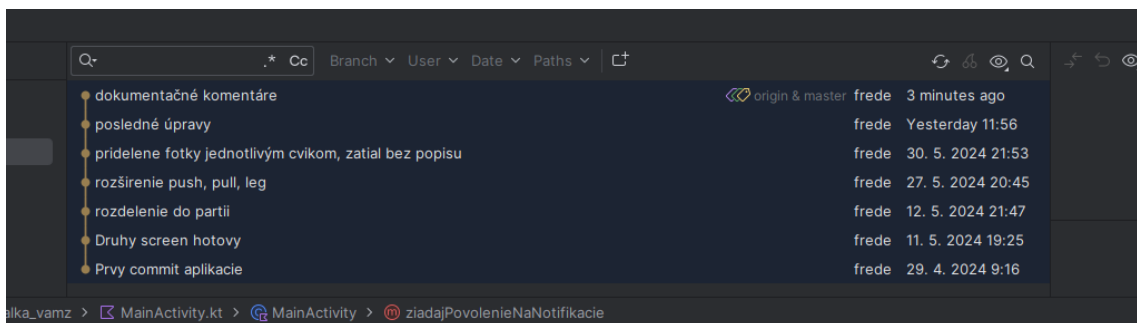
NotifikacnyPomocnik: Trieda na správu notifikácií, zahŕňa vytvorenie notifikačného kanála a odosielanie notifikácií.

```
fun vytvorNotifikacnyKanal(context: Context) {  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {  
        val name = "Example Channel"  
        val descriptionText = "This is an example channel"
```

```
fun posliNotifikaciju(context: Context, title: String, content: String) {  
    val builder = NotificationCompat.Builder(context, CHANNEL_ID)  
        .setSmallIcon(R.drawable.notifikacia) // Ikona notifikácie  
        .setContentTitle(title) // Názov notifikácie  
        .setContentText(content) // Obsah notifikácie  
  
    with(NotificationManagerCompat.from(context)) { this: NotificationManagerCompat  
        // Skontroluje povolenie na odosielanie notifikácií  
        if (ActivityCompat.checkSelfPermission(  
            context, Manifest.permission.POST_NOTIFICATIONS) == PackageManager.PERMISSION_GRANTED) {  
            notify(notificationId, builder.build())  
        }  
    }  
}
```

Práca s Git serverom

Aplikácia bola vyvíjaná s použitím Git systému, s pravidelnými commitmi dokumentujúcimi vývoj aplikácie.



Návrh aplikácie, architektúra

Aplikácia je **rozdelená** do **logických balíčkov**, napr. registracia, pokrok, push, pull, legs_abs, notifikacia, zaklad.

Oddelenie logiky a používateľského **rozhrania**: Použitie ViewModel pre správu stavu a logiku aplikácie, Composable funkcie pre používateľské rozhranie.

Dodržiavanie zásad kódovania

Identifikátory sú **pomenované** logicky a intuitívne.

Aplikácia obsahuje mechanizmy na **ošetrenie výnimiek**.

Kód je **optimalizovaný**, bez zbytočných duplícít.

Dizajn a UX aplikácie

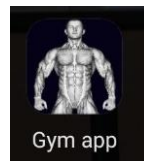
Aplikácia ponúka jedinečnú funkcionálnu **pre používateľov** vo **fitness** oblasti, s možnosťou sledovania pokroku a nastavovania tréningových cieľov.

Používateľské rozhranie je navrhnuté s dôrazom na **jednoduchosť** a **intuitívnosť**, s jasne označenými prvkami a vstupnými poliami.

Aplikácia poskytuje **jednoduché** a **prehľadné ovládanie**, prispôbené potrebám používateľov vo fitness oblasti.

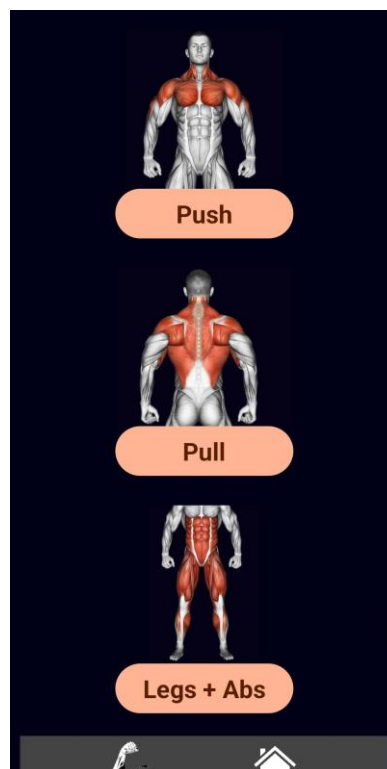
Aplikácia je prepracovaná, s **atraktívnym dizajnom** a zložitou, ktorá zodpovedá jej funkcionalite.

Zmenená **ikona aplikácie** a **názov**:

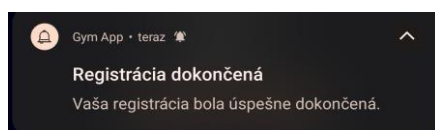


Vyplnenie osobných údajov

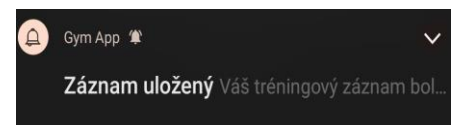
HomeScreen



Vyplnenie tréningového záznamu



notifikácie



4 ZOZNAM POUŽITÝCH ZDROJOV

1. Pomoc pri notifikácii: [Android - Maros Cavojsky \(mpage.sk\)](#)
2. Obrázky – z aplikácii google play a google image
3. [Serialize and Deserialize JSON using GSON Library in Android - GeeksforGeeks](#)

ZÁVER

Táto dokumentácia pokrýva všetky potrebné aspekty pre projekt Gym App, vrátane popisu problému, návrhu riešenia, popisu implementácie a zoznamu použitých zdrojov. Projekt bol vytvorený s dôrazom na prehľadnosť, efektivitu a používateľskú prístupnosť, čo by malo zaistiť jeho úspešné využitie začiatočníkmi vo fitness.

Túto semestrálnu prácu zameranú na fitko a cvičenie som si vybral aj z toho dôvodu, že som sám začal cvičiť a viem, aké je to vo fitku. Prostredníctvom tohto projektu som sa snažil spojiť teoretické vedomosti získané počas štúdia s praktickými skúsenosťami z fitka. Verím, že táto aplikácia bude užitočná pre ostatných, ktorí chcú začať s cvičením a hľadajú jednoduchý a efektívny nástroj na sledovanie svojho pokroku a plánovanie tréningov.

Vývoj tejto aplikácie mi poskytol hlbšie porozumenie procesov spojených s vývojom softvéru, najmä v oblasti mobilných aplikácií. Taktiež som mal príležitosť aplikovať moderné technológie a nástroje, ako sú Jetpack Compose a ViewModel, čo mi umožnilo vytvoriť efektívny a udržiavateľný kód.

Dúfam, že táto aplikácia bude prínosná pre všetkých, ktorí chcú zlepšiť svoju fyzickú kondíciu a dosiahnuť svoje fitness ciele. S pokračujúcim rozvojom a zlepšovaním tejto aplikácie som si istý, že sa stane neoceniteľným pomocníkom pre mnohých používateľov.

Do budúcnosti by sa mohla daná aplikácia rozšíriť. Medzi hlavné ciele patria:

- Integrácia so sociálnymi sieťami: Umožniť používateľom zdieľať svoje pokroky a tréningové plány na sociálnych sieťach.
- Pridanie pokročilých analytických nástrojov: Umožniť používateľom detailnejšie analyzovať ich pokroky a identifikovať oblasti na zlepšenie.
- Podpora pre nositeľné zariadenia: Umožniť synchronizáciu s nositeľnými zariadeniami, ako sú fitness náramky a smart hodinky, pre automatické sledovanie aktivity a zdravotných údajov.

- Komunitné funkcie: Vytvoriť priestor pre používateľov, kde môžu diskutovať o svojich pokrokoch, zdieľať tipy a motivovať sa navzájom.

Tieto vylepšenia by mali prispieť k ešte väčšej používateľskej spokojnosti a poskytnúť používateľom ešte viac nástrojov na dosahovanie ich fitness cieľov.