

## ZÁVĚREČNÁ STUDIJNÍ PRÁCE

### dokumentace

## Mobilní aplikace Rep Track ve Flutteru



**Autor:** Filip Říčný  
**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování  
**Třída:** IT4  
**Školní rok:** 2024-25

## **Poděkování**

Chtěl bych vyjádřit své upřímné poděkování panu učiteli Mgr. Marku Lučnému za jeho odborné vedení, vstřícnost a cenné rady, které mi poskytoval během práce na projektu. Dále bych rád poděkoval Adamovi Stuchlíkovi za jeho ochotu konzultovat jednotlivé problematiky. Velké díky patří také Matěji Říčnému za jeho psychickou podporu.

## **Prohlášení**

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2024

.....  
Podpis autora

## **Abstrakt**

Cílem projektu bylo vytvořit plně funkční mobilní aplikaci určenou k zaznamenávání cvičení. Aplikace je navržena s důrazem na uživatelskou přívětivost a umožňuje registraci, která poskytuje možnost přizpůsobit si prostředí podle individuálních potřeb. Uživatelé si mohou vytvářet a ukládat vlastní tréninkové rutiny nebo využít již připravené rutiny pro inspiraci. Aplikace rovněž umožňuje sledování pokroku prostřednictvím detailní historie tréninků.

Pro vývoj aplikace byl použit framework Flutter, který využívá programovací jazyk Dart. Pro správu dat a autentizaci uživatelů byla implementována cloudrová služba Firebase. Projekt klade důraz na jednoduchost ovládání, okamžité uložení změn dat a moderní design, aby splňoval potřeby uživatelů, kteří hledají efektivní způsob, jak si vést přehled o svém cvičení.

## **Klíčová slova**

aplikace, Flutter, Firebase, trénink, mobilní aplikace, autentizace ...

# Obsah

<b>Úvod</b>	<b>5</b>
<b>1 Struktura aplikace a práce s daty</b>	<b>6</b>
1.1 Flutter a jazyk Dart . . . . .	6
1.2 Vývojové prostředí a vytvoření projektu . . . . .	6
1.3 Adresářová struktura . . . . .	7
1.4 Získávání dat . . . . .	8
1.5 Ukládání dat . . . . .	8
1.6 Databázový model aplikace . . . . .	10
1.7 Autentizace a správa uživatelských účtů . . . . .	11
<b>2 Mobilní aplikace</b>	<b>15</b>
2.1 Úvod . . . . .	15
2.2 Struktura a funkcionalita aplikace . . . . .	15
2.3 Úvodní stránka . . . . .	16
2.4 Stránka s profilem . . . . .	17
2.5 Tréninková stránka . . . . .	20
2.6 Rutinní stránka . . . . .	21
2.7 Stránka se cviky . . . . .	22
<b>3 Závěr</b>	<b>23</b>

## ÚVOD

Rozhodl jsem se vytvořit mobilní aplikaci zaměřenou na správu tréninků, protože mě samotného cvičení baví a rád zkouším nové aplikace. Chtěl jsem vytvořit nástroj, který by kombinoval jednoduchost a praktičnost s funkcemi, které usnadní sledování výkonu. Vývoj aplikace ve frameworku Flutter mi přinesl cenné zkušenosti v oblasti multiplatformního vývoje a práce s databázemi. Inspirací mi byly podobné aplikace zaměřené na záznam tréninkových rutin a fitness plány, jako Hevy a Liftoff.

Hlavním cílem projektu bylo vytvořit plně funkční mobilní aplikaci, která ukládá data do cloudu v reálném čase a umožňuje uživatelům snadný přístup k jejich statistikám. Tento projekt mi také pomohl prohloubit znalosti v práci s moderními technologiemi, jako jsou Firebase Cloud Firestore, Firebase Authentication a Firebase Storage.

Další části dokumentu obsahují podrobné informace o struktuře aplikace a způsobu práce s daty, včetně ukládání a autentizace uživatelů. Následuje popis jednotlivých stránek aplikace, jejich funkcí a vzhledu. V závěru dokumentu je uvedeno hodnocení projektu a plánovaná vylepšení.

# 1 STRUKTURA APLIKACE A PRÁCE S DATY

## 1.1 FLUTTER A JAZYK DART

Při vývoji jsem se rozhodl použít framework Flutter, který v kombinaci s programovacím jazykem Dart poskytuje ideální prostředí pro tvorbu moderního a rychlého uživatelského rozhraní. Velkou výhodou Flutteru je jeho jednoduchá syntaxe a přehledná struktura, která mi umožnila efektivně organizovat kód a rychle se zorientovat i při složitějších úpravách. Díky integrovaným nástrojům jsem měl okamžitou zpětnou vazbu při vývoji, což značně urychlilo ladění a celkovou práci na aplikaci.

## 1.2 VÝVOJOVÉ PROSTŘEDÍ A VYTVOŘENÍ PROJEKTU

Pro vývoj aplikace jsem nainstaloval Android Studio pro simulaci mobilního zařízení a vytvořil virtuální zařízení se systémem Android pro testování aplikace. Poté jsem stáhl Flutter SDK a pomocí rozšíření ve Visual Studio Code zajistil podporu pro spouštění a ladění projektů. Bylo nutné nastavit cestu k SDK v proměnných prostředí, aby příkazy jako `flutter run` byly funkční přímo v terminálu.

Nový projekt jsem vytvořil zkratkou `CTRL + SHIFT + P`, zadal jeho název a zvolil cílovou složku. Výsledkem byla základní struktura aplikace připravená pro další vývoj uživatelského rozhraní a funkcionalit.

## 1.3 ADRESÁŘOVÁ STRUKTURA

```

1 |--- android // složka pro spustitelnost aplikace na androidu
2 |--- app // konfigurace a soubory aplikace pro Android
3   |--- build.gradle // vybudování spuštění
4   |--- src // zdrojové soubory pro Android
5   |--- build.gradle // hlavní nastavení buildu
6   |--- gradle.properties // položky pro spuštění
7   |--- settings.gradle // nastavení pro spuštění
8 |--- lib // hlavní složka pro aplikaci
9   |--- auth // autentizace a přihlášení/registrace
10  |   |--- auth.dart // hlavní logika autentizace
11  |   |--- login_or_register // obrazovky přihlášení/registrace
12  |--- components // vlastní utility a komponenty
13  |--- buttons // tlačítka v aplikaci
14  |   |--- login_buttons.dart // tlačítka na prihlásení
15  |   |--- exercise_buttons.dart // tlačítka úvodní stránky
16  |--- my_bottom_bar.dart // spodní navigační panel
17  |--- my_bold_text.dart // komponenta pro tučný text
18  |--- my_textfield.dart // vlastní textové pole
19  |--- .....
20  |--- images // obrázky v aplikaci
21  |--- pages // hlavní stránky aplikace
22    |--- exercise_details // podstránky detailu cvičen
23      |--- exercise_detail_about_page.dart // informace o cvičení
24      |--- exercise_detail_best_page.dart // nejlepší výkony
25      |--- exercise_detail_history_page.dart // historie cvičení
26      |--- exercise_detail_page.dart // detailní stránka cvičení
27    |--- routines // stránky rutin jsou podobné nebudu vypisovat celé
28      |--- .....
29    |--- home_page.dart // přepínání mezi friends, add_workout a profile
30    |--- profile_page.dart // stránka profilu
31    |--- add_workout_page.dart // stránka přidání tréninku
32    |--- workout_details_page.dart // detail tréninku
33    |--- edit_profile_page.dart // úprava profilu
34    |--- .....
35    |--- services // služby aplikace
36      |--- exercise_list.dart // kontrola přihlášení
37      |--- firestore.dart // správa databáze
38      |--- storage_service.dart // správa úložiště
39    |--- theme // nastavení vzhledu
40      |--- dark_mode.dart // tmavý režim
41      |--- light_mode.dart // světlý režim
42      |--- theme_provider.dart // výběr režimu
43    |--- firebase_options.dart // nastavení firebase
44    |--- main.dart // hlavní stránka spuštění
45 |--- README.md // soubor s informacemi o projektu
46 |--- analysis_options.yaml // konfigurace pro analýzu kódu
47 |--- devtools_options.yaml // konfigurace DevTools
48 |--- firebase.json // nastavení Firebase databáze
49 |--- pubspec.lock // specifikace verzí balíčků
50 |--- pubspec.yaml // seznam používaných balíčků

```

Kód 1.1: Adresářová struktura aplikace

## 1.4 ZÍSKÁVÁNÍ DAT

Veškerá data v aplikaci pocházejí od uživatelů nebo byla předem připravena administrátorem. Aplikace umožňuje vytvářet cviky, tréninkové rutiny a spravovat profil. Uživatelé mohou přidávat vlastní záznamy, zatímco administrátor má právo vytvářet globální rutiny viditelné všem uživatelům.

```
1 //READ
2 Stream<QuerySnapshot> getStream(String collectionName){
3     final stream = FirebaseFirestore.instance.collection(collectionName).orderBy('name').snapshots();
4     return stream;
5 }
6 Future<DocumentSnapshot> getDocumentById(String collection, String id) {
7     return FirebaseFirestore.instance.collection(collection).doc(id).get();
8 }
```

Kód 1.2: Čtení dat kolekcí

## 1.5 UKLÁDÁNÍ DAT

Pro ukládání dat využívám clouдовou databázi Firebase Cloud Firestore, do které jsem ukládal:

- Uživatelské profily (jméno, e-mail, URL profilového obrázku ...).
- Záznamy tréninků (počet opakování, váhy, délka tréninku, cviky ...).
- Jednotlivé cviky (jméno, typ, svaly používané, potřebné náčiní ...).
- Rutiny (počet opakování, váhy, odpočinkový čas, jednotlivé cviky ...).

Historie tréninků se zobrazuje na stránce profilu pomocí kalendáře. Všechny tyto záznamy jsou aktualizovány asynchronně, což znamená, že změny se okamžitě zobrazí bez nutnosti manuálního obnovení.

```

1
2 void submit() async{
3     /* Získání unikátního ID cviku */
4     final exerciseId = FirebaseFirestore.instance.collection('exercises').doc().id;
5     AppLogger.logInfo("Attempting to save a exercise...");
6     /* Zobrazení načítacího kolečka */
7     showDialog(context: context, builder: (context)=> const Center(
8         child: CircularProgressIndicator(),));
9     /* Kontrola, zda jsou všechna pole vyplněná */
10    if(typeController.text.isEmpty || muscleGroupController.text.isEmpty || muscleController.text.isEmpty
11        ||nameController.text.isEmpty || equipmentController.text.isEmpty ){
12        if(mounted){
13            /* Skrytí načítacího kolečka */
14            Navigator.pop(context);
15            /* Zobrazení upozornění pro uživatele */
16            displayMessageToUser("All fields must be filled ", context);}
17        return;}
18    /* Získání reference na úložiště obrázku */
19    final imageRef = storageRef.ref().child('exercises').child("${currentUser?.email}-${nameController.text
20 }");
21    /* Kontrola, zda je přiložen obrázek */
22    if(_imageFile!=null){
23        final imageBytes= await _imageFile!.readAsBytes();
24        /* Uložení obrázku do cloudového úložiště */
25        await imageRef.putData(imageBytes);
26        /* Získání URL obrázku pro uložení v databázi */
27        imageUrl = await imageRef.getDownloadURL();}
28        if(mounted)Navigator.pop(context);
29        try{
30            /* Uložení dat cviku do databáze */
31            await FirebaseFirestore.instance.collection("Exercises").doc(exerciseId).set({
32                'exerciseId': exerciseId,
33                'name': nameController.text,
34                'trackingType': typeController.text,
35                'muscleGroup': muscleGroupController.text,
36                'muscles' : muscleController.text,
37                /* Určení, zda je cvik veřejný pro všechny, nebo pouze pro daného uživatele */
38                'type': currentUser?.email == "admin@admin.cz" ? "predefined": "custom",
39                'createdBy': currentUser?.uid,
40                'imageUrl': imageUrl,
41                'equipment': equipmentController.text,
42                'createdAt': dateFormat.format(DateTime.now()),
43                "updatedAt": dateFormat.format(DateTime.now()), });
44        if(mounted){
45            Navigator.pop(context);
46            /* Zobrazení zprávy uživateli o úspěšném uložení */
47            displayMessageToUser(currentUser?.email == "admin@admin.cz" ? "Exercise created for everyone": "Exercise created", context);
48            AppLogger.logInfo("Exercise saved successfully.");}}
49        on FirebaseAuthException catch (e, stackTrace) {
50            /* Výpis chyby do konzole logger pro vývojáře */
51            AppLogger.LogError("Failed to save exercise.", e, stackTrace);}}
```

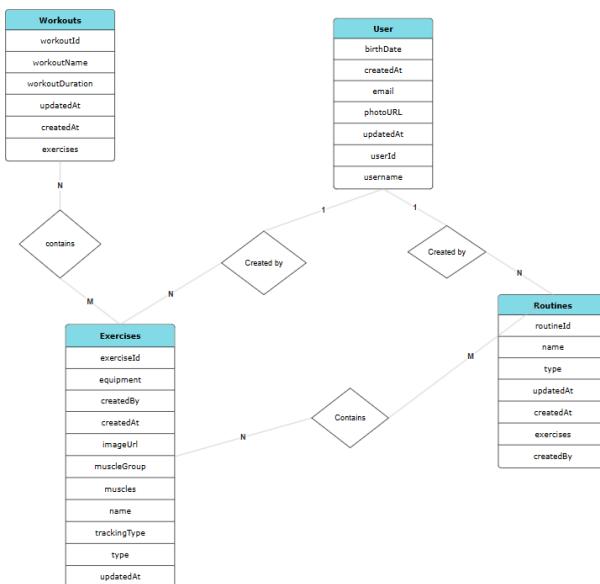
Kód 1.3: Ukládání dat cviků

### 1.5.1 Databáze v aplikaci

V rámci své mobilní aplikace jsem využil dvě cloudové služby od Firebase pro správu dat a souborů: Firebase Cloud Firestore a Firebase Storage. Obě tyto technologie mi umožnily efektivní práci s daty uživatelů a správu souvisejících mediálních souborů, jako jsou obrázky profilů a cviků.

- **Firebase Cloud Firestore** je cloudová NoSQL databáze, která umožňuje ukládat strukturovaná data ve formě kolekcí a dokumentů. Je známá svou schopností pracovat v reálném čase, což zajišťuje, že změny provedené v databázi se okamžitě projeví ve všech připojených zařízeních bez nutnosti manuálního obnovení. V mé aplikaci jsem Cloud Firestore využil k ukládání informací o uživatelích, záznamech tréninků, jednotlivých cvicích a tréninkových rutinách.
- **Firebase Storage** je cloudové úložiště sloužící k ukládání mediálních souborů, jako jsou obrázky a videa. Umožňuje snadnou správu přístupových práv a práci s velkými soubory. V mé aplikaci jsem jej využil k ukládání profilových obrázků uživatelů, aby si mohli přizpůsobit svůj účet, a také k ukládání obrázků cviků, což poskytlo vizuální podporu pro lepší identifikaci cviků v tréninkových rutinách.

## 1.6 DATABÁZOVÝ MODEL APLIKACE



Obrázek 1.1: Databazový model aplikace

## 1.7 AUTENTIZACE A SPRÁVA UŽIVATELSKÝCH ÚČTŮ

Pro autentizaci jsem použil Firebase Authentication. Každý uživatel má při registraci přiřazené jedinečné ID, e-mail, uživatelské jméno, datum přihlášení a datum vytvoření účtu. Tyto informace jsou uloženy v databázi spolu s URL profilového obrázku.

Uživatelé mají přístup pouze k těm cvičením a rutinám, které si sami vytvořili. Může se jednat o různé cviky, které si uživatel může přizpůsobit a vytvořit své vlastní tréninkové plány. V systému existují dva typy uživatelských profilů – administrátor a běžný uživatel:

- **Administrátor** má rozšířená práva. Může upravovat cvičení a tréninkové plány ostatních uživatelů a také vytvářet předdefinované cvičení a rutiny, které budou k dispozici pro všechny uživatele.
- **Běžný uživatel** má možnost měnit své osobní údaje, profilový obrázek a nastavení. Kromě toho může požádat o obnovení zapomenutého hesla nebo si účet úplně smazat.

Tento systém zajišťuje flexibilitu a personalizaci pro každého uživatele, zatímco administrátorovi umožňuje správu a údržbu obsahu pro všechny uživatele.

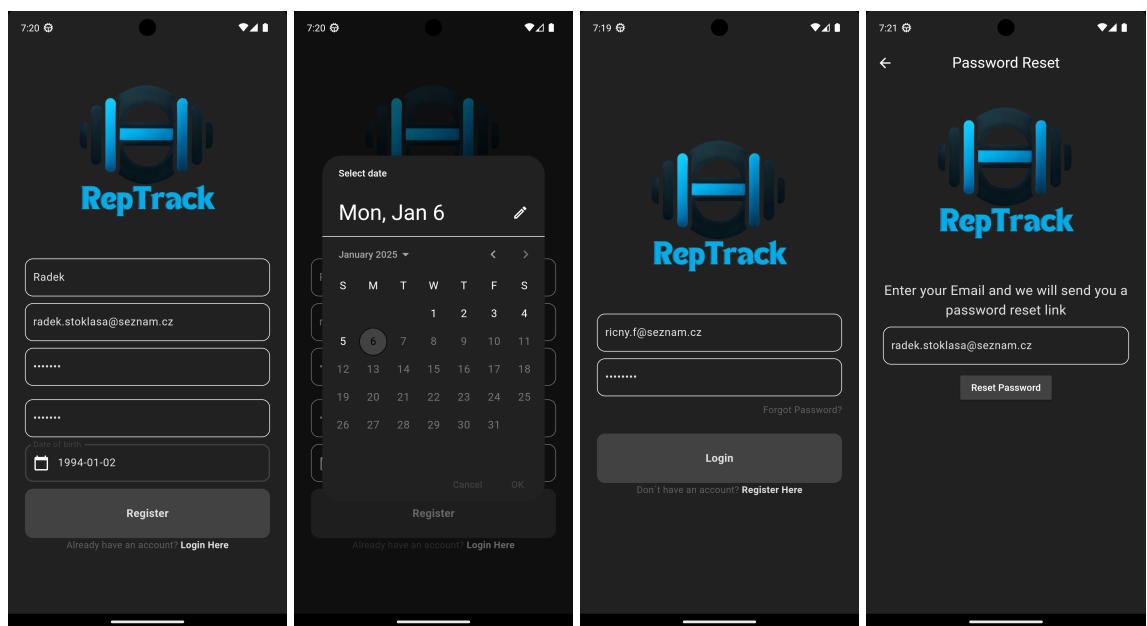
Identifier	Providers	Created ↓	Signed In	User UID
ricny.f@seznam.cz	✉️	Jan 2, 2025	Jan 2, 2025	dM4qO94DGuThe7CuYiNjjAal...
admin@admin.cz	✉️	Jan 2, 2025	Jan 4, 2025	azm3H0px2oa3PQ5F1G7iEgZ...
jonasbosshd@seznam.cz	✉️	Jan 2, 2025	Jan 3, 2025	Aeiw5xH4MEa9q8enT3zfLVH...
pokus@gmail.com	✉️	Dec 17, 2024	Dec 17, 2024	rwLC0AoCpIP8RBFuxv99uJiH...

Obrázek 1.2: Firebase console

### 1.7.1 Registrace a přihlášení uživatele

Na registrační stránce zadává uživatel své uživatelské jméno, e-mail, heslo a jeho potvrzení. Dále musí vyplnit svůj věk, který musí být minimálně 13 let. Po splnění všech podmínek se provede registrace, při které se uživatelská data uloží do databáze. Uložené údaje obsahují unikátní ID uživatele, přihlašovací údaje a další informace.

Po registraci se uživatel může přepnout na přihlašovací stránku, kde zadává svůj e-mail a heslo pro přihlášení. V případě zapomenutého hesla může využít funkci pro jeho obnovení. Po zadání e-mailové adresy mu přijde na e-mail odkaz pro resetování hesla, který jej provede procesem vytvoření nového hesla. Celý proces registrace i přihlášení je zabezpečený pomocí služby Firebase Authentication, která zajišťuje bezpečné ukládání a správu uživatelských přístupů.



Obrázek 1.3: Nastavení aplikace

## 1.7.2 Kód autentizace

```

1 /* Dotaz na získání informací o cviku */
2 final List<DocumentSnapshot> exercisesList = snapshot.data!.docs.where((doc) {
3     final data = doc.data() as Map<String, dynamic>;
4     final createdBy = data['createdBy'];
5     final type = data['type'];
6
7     // Administrátor vidí všechny cviky
8     if (currentUser?.email == "admin@admin.cz") {
9         return true;
10    }
11
12    // Běžní uživatelé vidí pouze cviky, které sami vytvořili, nebo předdefinované cviky
13    return createdBy == currentUser?.uid || type == "predefined";
14 }).toList();

```

Kód 1.4: Čtení dat cviků

```

1 Future passwordReset()async{
2     // Záznam pokusu o reset hesla do loggeru
3     AppLogger.logInfo("Attempting to reset a password...");
4     // Kontrola, zda je textové pole pro e-mail prázdné
5     if(emailController.text.isEmpty){
6         //napíše uživateli chybu a do loggeru
7         displayMessageToUser("Email cant be empty!", context);
8         AppLogger.LogError("Failed to reset password. Email is empty.");
9         return;
10    }
11    // Pokus o odeslání e-mailu s odkazem na reset hesla
12    try{
13        await FirebaseAuth.instance.sendPasswordResetEmail(email: emailController.text.trim());
14        // Zobrazení zprávy uživateli o úspěšném odeslání e-mailu
15        if(mounted)displayMessageToUser('If the email is registered, a reset link has been sent.', context);
16        //kontrola chyb
17        on FirebaseAuthException catch (e, stackTrace){
18            // Zobrazení chybové zprávy uživateli a záznam chyby do loggeru
19            if(mounted)displayMessageToUser(e.message.toString(), context);
20            AppLogger.LogError("Failed to reset password.", e, stackTrace);}}}

```

Kód 1.5: Restartování hesla

```
1 void login() async{
2     /* Záznam pokusu o přihlášení do loggeru */
3     AppLogger.logInfo("Attempting to login...");
4     /* Zobrazení načítacího kolečka */
5     showDialog(context: context, builder: (context)=> const Center(
6         child: CircularProgressIndicator(.));
7     // Kontrola, zda jsou obě textová pole vyplněná
8     if(emailController.text.isEmpty || passwordController.text.isEmpty){
9         if(mounted){
10            // Skrytí načítacího kolečka
11            Navigator.pop(context);
12            // Zobrazení chybové zprávy uživateli
13            displayMessageToUser("All fields must be filled", context); }
14        else{
15            // Pokus o přihlášení
16            try {
17                // pokus o prihlaseni s emailem a heslem
18                await FirebaseAuth.instance.signInWithEmailAndPassword(email: emailController.text,
19                password: passwordController.text);
20                // Skrytí načítacího kolečka po úspěšném přihlášení
21                if (mounted) Navigator.pop(context);
22                // Záznam úspěšného přihlášení do loggeru
23                AppLogger.logInfo("Logged in successfully.");
24            // ešení chyb při přihlášení
25            on FirebaseAuthException catch (e, stackTrace){
26                if (mounted) {
27                    // Zobrazení chybové zprávy uživateli a zápis chyby do loggeru
28                    displayMessageToUser(e.message.toString(), context);
29                    AppLogger.LogError("Failed to login.", e, stackTrace);
30                }}}}
```

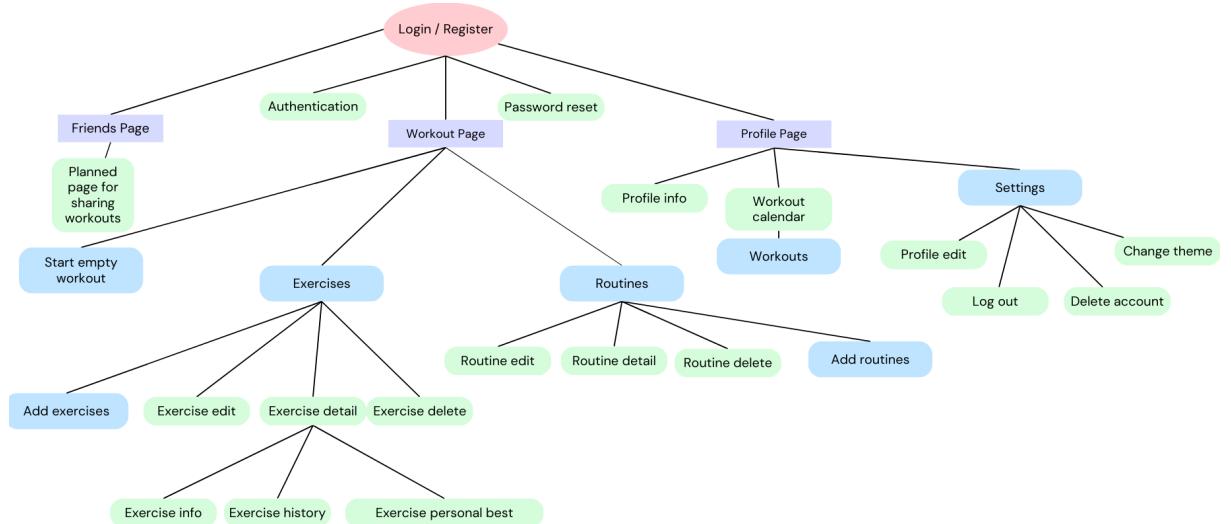
Kód 1.6: Login funkce

## 2 MOBILNÍ APLIKACE

### 2.1 ÚVOD

V následující části dokumentu se zaměřím na popis struktury a funkcionality mobilní aplikace. Bude představen způsob organizace jednotlivých komponent aplikace a přehled hlavních stránek uživatelského rozhraní.

### 2.2 STRUKTURA A FUNKCIONALITA APLIKACE

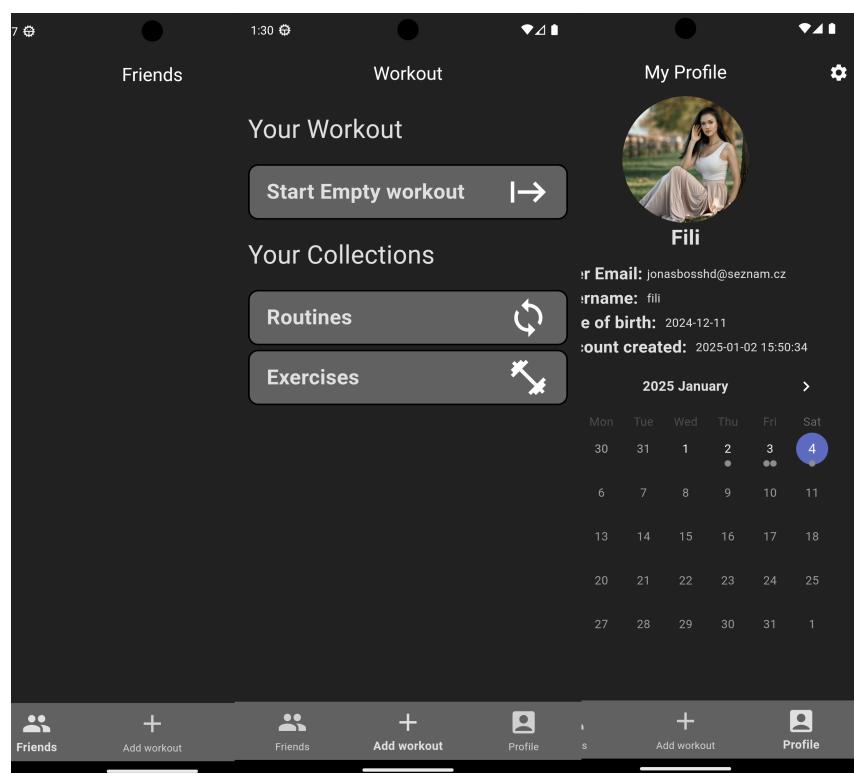


Obrázek 2.1: Struktura aplikace

## 2.3 ÚVODNÍ STRÁNKA

Úvodní stránka aplikace obsahuje dolní navigační lištu (bottom bar), která umožňuje přepínání mezi hlavními sekczemi:

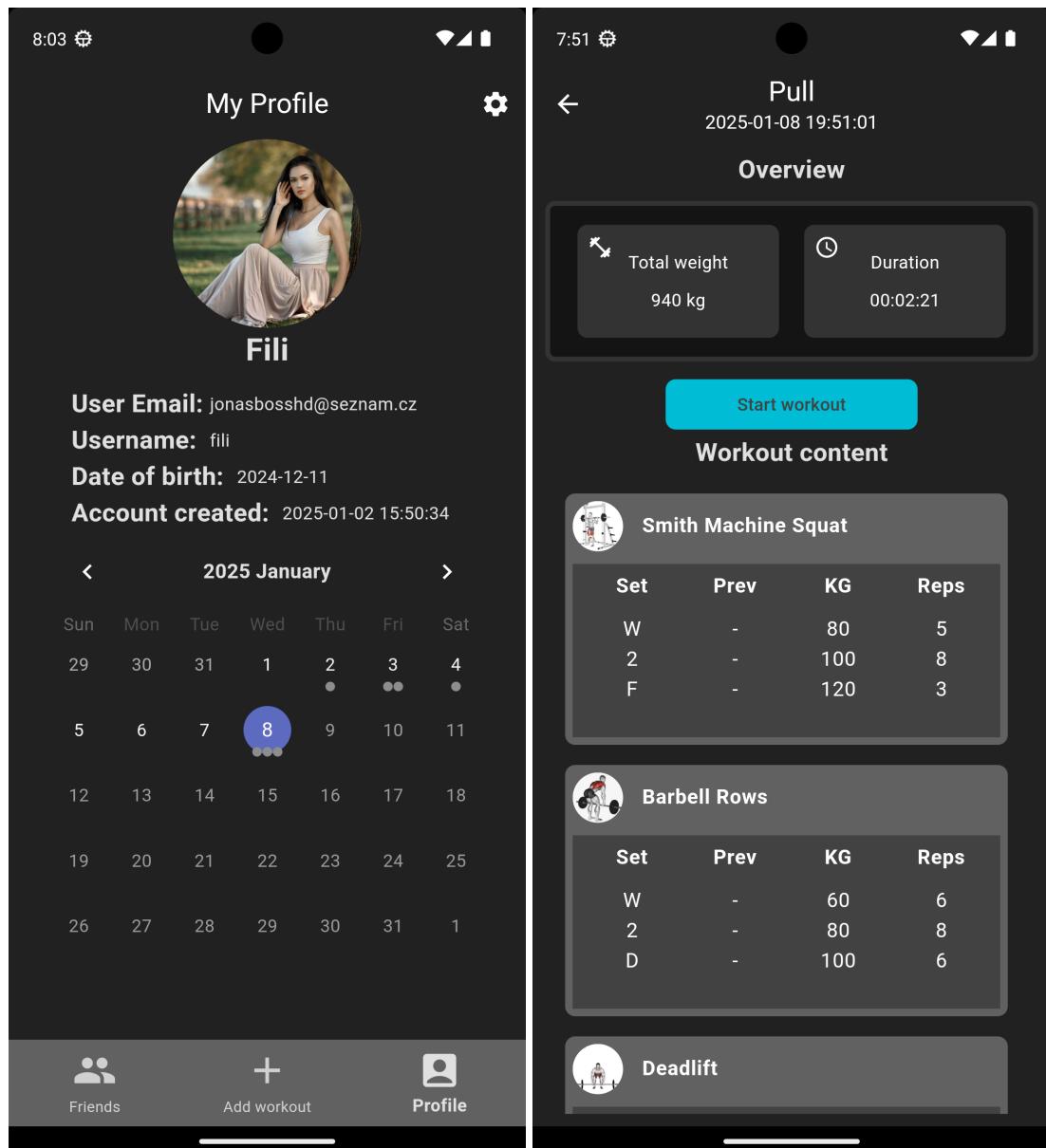
- **Friends** je sekce pro správu přátel – momentálně není dokončená.
- **Add Workout** slouží k zahájení cvičení, výběru rutin nebo cviků a jejich následné úpravě.
- **Profile** obsahuje uživatelský profil s možností úpravy informací, součástí je kalendář, ve kterém lze zobrazit historii tréninku a také nastavení aplikace.



Obrázek 2.2: Úvodní stránka

## 2.4 STRÁNKA S PROFILEM

Na stránce profilu se nacházejí údaje o uživateli spolu s profilovým obrázkem. Pod nimi je umístěn kalendář, ve kterém jsou zvýrazněny dny, kdy byl zaznamenán trénink. Po kliknutí na konkrétní den si může uživatel zobrazit podrobnosti a vybrat konkrétní cvik pro zobrazení. Tato funkce umožňuje snadnou orientaci v historii tréninků a rychlý přístup k jednotlivým záznamům.

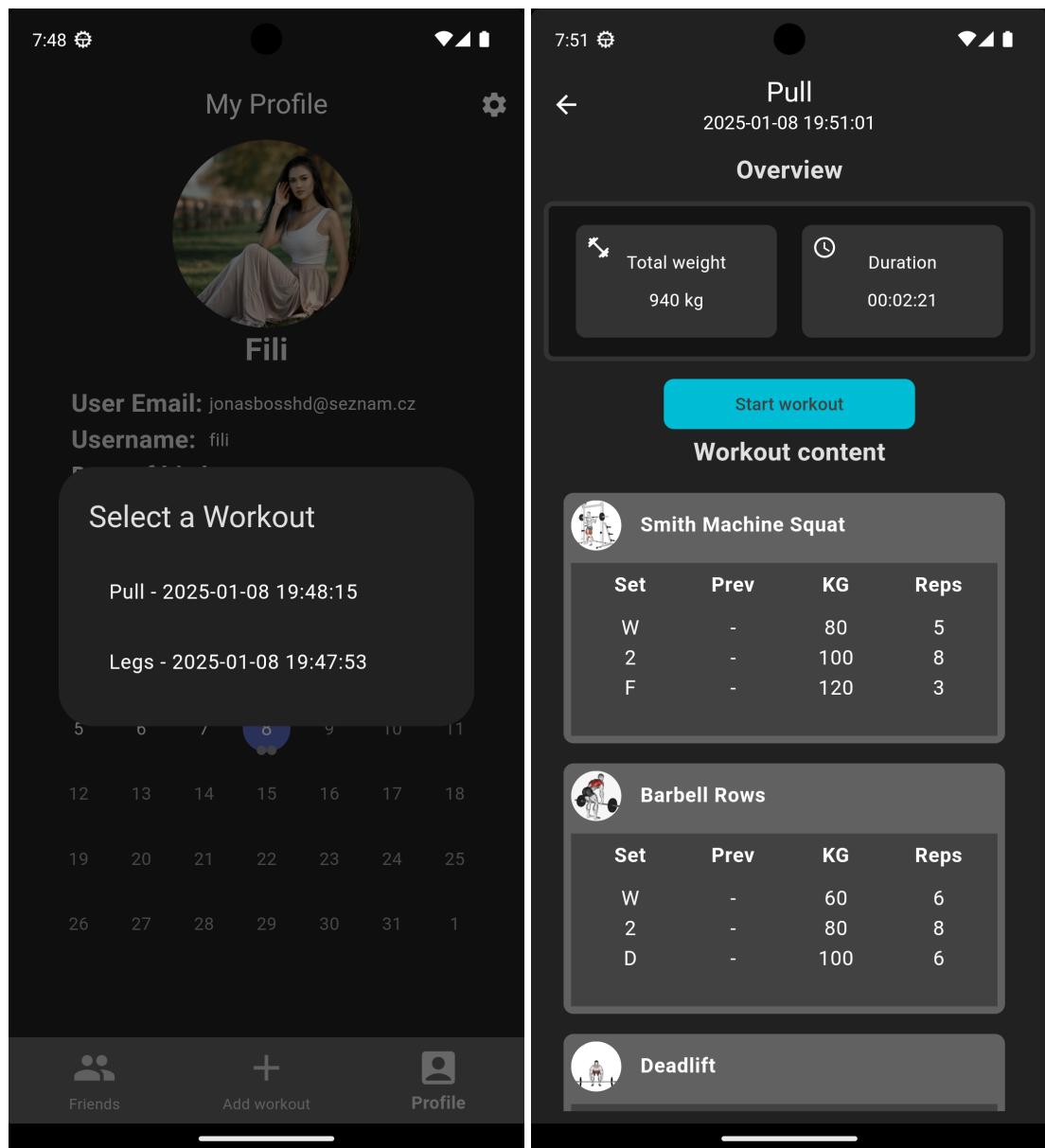


Obrázek 2.3: Výběr tréninků

Obrázek 2.4: Informace o tréninku

### 2.4.1 Kalendář cviků

Pro kalendář jsem použil balíček `table_calendar`, který umožňuje přehledné zobrazení zařazených tréninků. V kalendáři jsou zvýrazněny dny, kdy byl trénink proveden. Po kliknutí na konkrétní den se zobrazí podrobné informace o tréninku, včetně cviků a časů. Uživatel má také možnost spustit trénink znovu přímo z této sekce, což zajišťuje snadné opakování oblíbených rutin bez zdlouhavého nastavování.

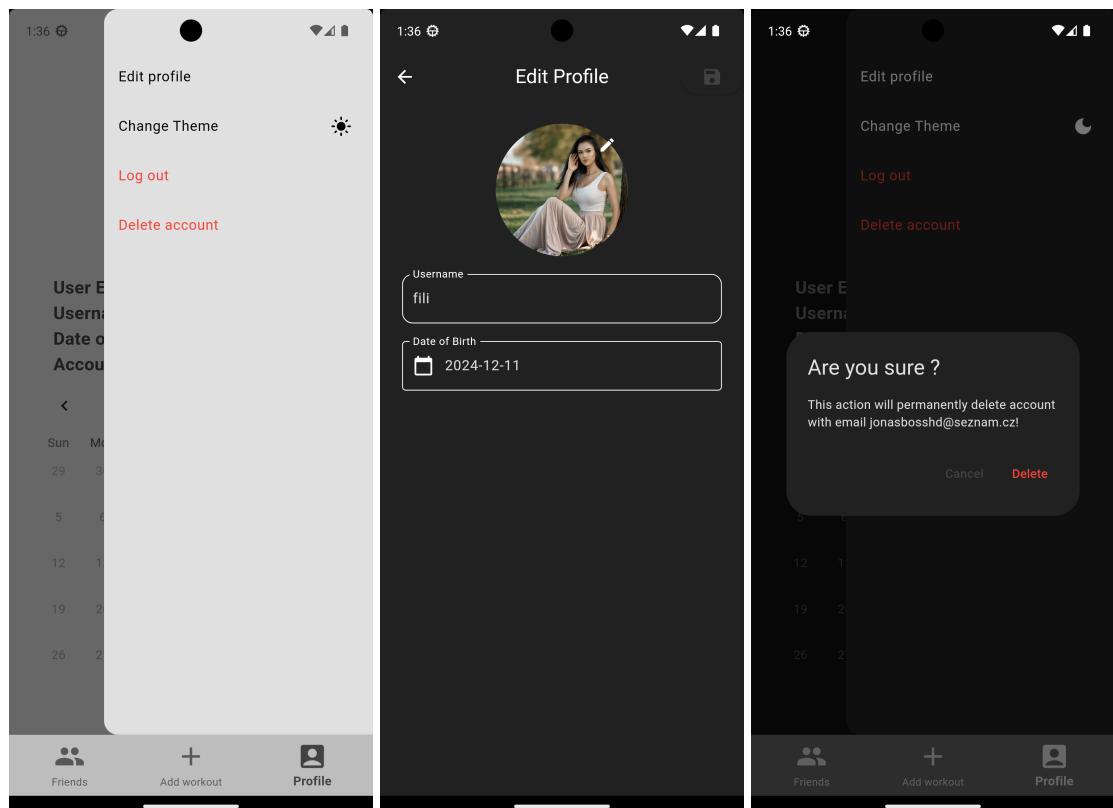


Obrázek 2.5: Výběr tréninků

Obrázek 2.6: Informace o tréninku

## 2.4.2 Nastavení aplikace

Na stránce profilu se v postranním panelu nachází nastavení aplikace, kde si uživatel může přizpůsobit svůj profil. Může změnit své jméno, datum narození a profilový obrázek. K dispozici je také možnost přepnutí vzhledu aplikace mezi světlým a tmavým režimem. Uživatel se může rovněž odhlásit ze svého účtu nebo svůj účet trvale smazat včetně všech dat.



Obrázek 2.7: Nastavení aplikace

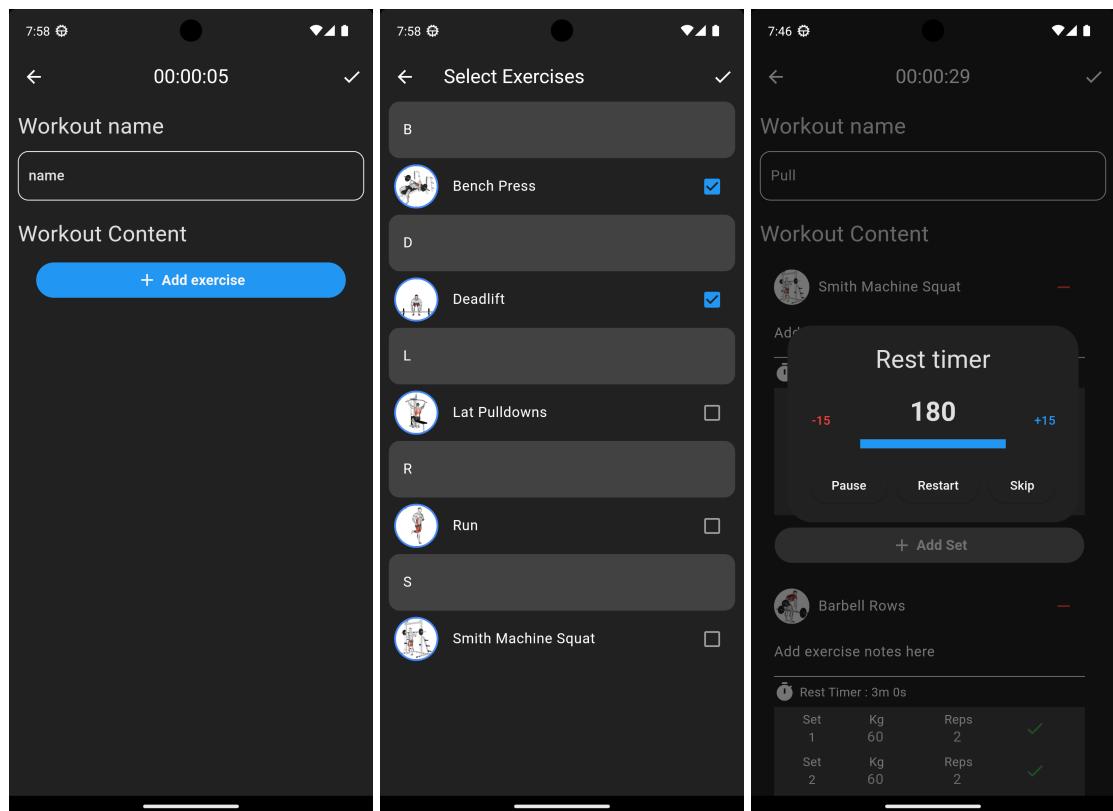
## 2.5 TRÉNINKOVÁ STRÁNKA

Na tréninkové stránce se ihned po spuštění automaticky zapne časovač. Uživatel musí nejprve zadat název tréninku a přidat alespoň jeden cvik – bez přidání cviku ho aplikace nepustí dál. Po přidání cviků se tyto položky zobrazí v obsahu stránky a uživatel může postupně přidávat jednotlivé série s váhami a počtem opakování. Jakmile uživatel označí sérii za splněnou, spustí se odpočítávání odpočinkového času, který je ve výchozím nastavení nastaven na 3 minuty, ale lze ho změnit podle potřeby.

Odpocinkový čas lze během odpočinku:

- **Přeskočit**
- **Restartovat**
- **Přidat nebo odebrat 15 sekund**

Uživatel může přidat libovolný počet cviků, ale všechny cviky musí být označeny jako dokončené, než lze celý trénink uložit do databáze Users/Workouts.

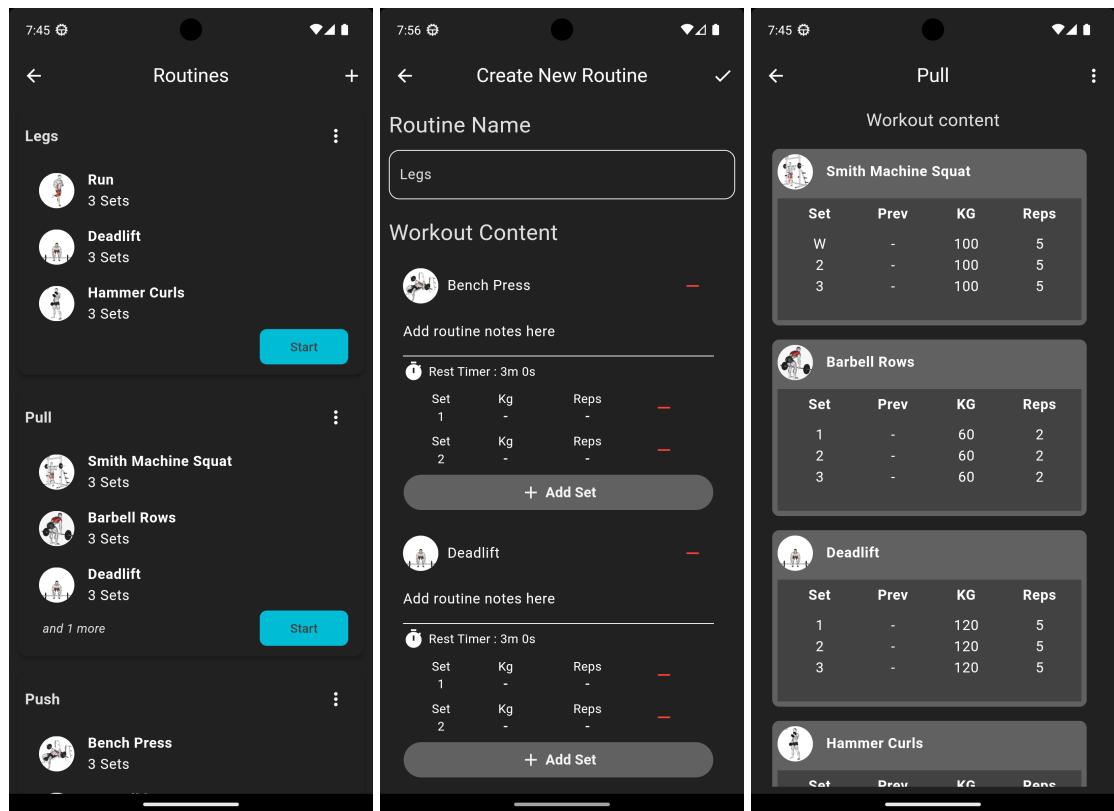


Obrázek 2.8: Tréninková stránka

## 2.6 RUTINNÍ STRÁNKA

V rutinách vidí běžný uživatel pouze své vlastní vytvořené rutiny a předdefinované rutiny dostupné všem. Administrátor má přístup ke všem rutinám a může je upravovat a mazat. Běžný uživatel může rutiny upravovat pouze tehdy, pokud si je sám vytvořil, ostatní rutiny si může pouze prohlížet.

Každou rutinu lze spustit, čímž se otevře tréninková stránka (workout page) s předem definovanými detailemi podle zvolené rutiny.

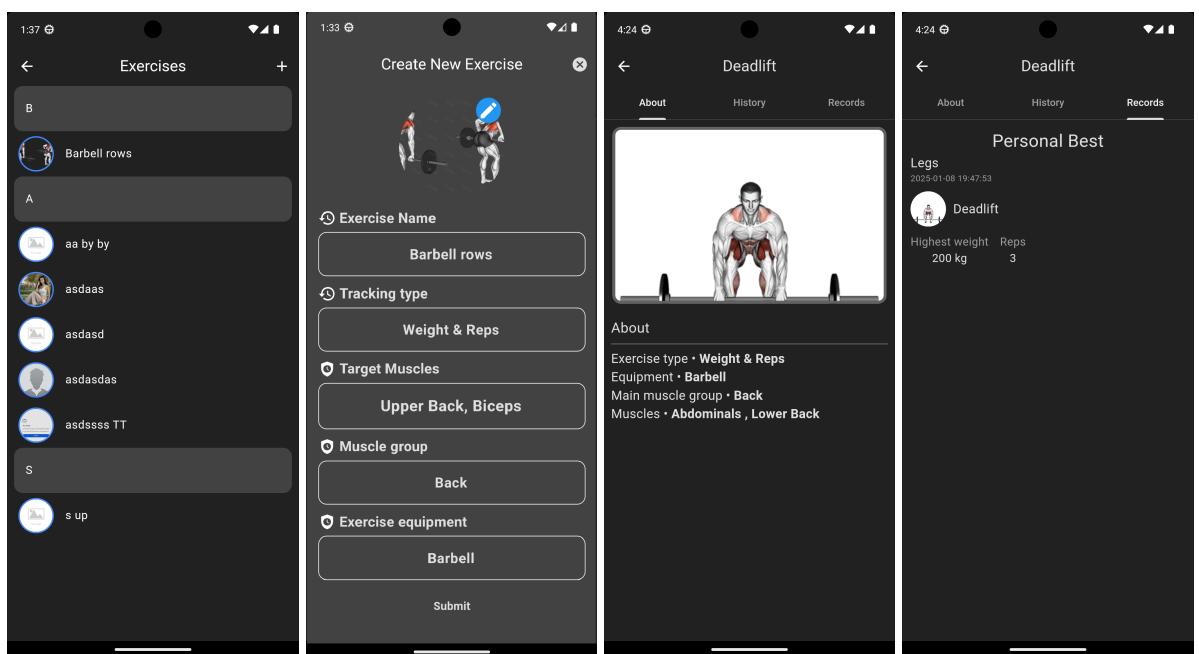


Obrázek 2.9: Stránky s rutinami

## 2.7 STRÁNKA SE CVIKY

Na stránce se cviky se zobrazuje seznam dostupných cviků, který zahrnuje předdefinované cviky pro všechny uživatele a vlastní cviky vytvořené běžnými uživateli. Administrátor má přístup ke všem cvikům a může je upravovat nebo mazat. Běžný uživatel vidí pouze své vlastní cviky a předdefinované cviky, které však nemůže upravovat.

Po rozkliknutí konkrétního cviku se zobrazí stránka s detailními informacemi o cviku, včetně popisu a zaměření na svalové skupiny. Dále je zde sekce historie cviků s přehledem předchozích provedení a nejlepšího zaznamenaného výkonu. Stránka také obsahuje možnost editace, která je dostupná pouze administrátorovi nebo autorovi cviku.



Obrázek 2.10: Stránky se cvičením

### 3 ZÁVĚR

Cílem této aplikace bylo vytvořit moderní a přehledný nástroj, který uživatelům umožní efektivně sledovat jejich tréninky, zaznamenávat pokrok a mít k dispozici personalizované tréninkové rutiny. Hlavním zaměřením vývoje byla především intuitivní uživatelská zkušenost, která zahrnuje správu cviků, rutin a přizpůsobitelné možnosti, jako je výběr světlého či tmavého režimu aplikace.

Aplikace v aktuální podobě splnila stanovené cíle, ale stále je zde prostor pro další rozvoj. Plánuji implementovat **seznam přátele**, který uživatelům umožní sdílet své výsledky s přáteli a vzájemně se motivovat. Dalším plánem je vylepšení **grafického zobrazení výsledků tréninků** – detailnější grafy pro jednotlivé cviky přinesou uživatelům lepší přehled o jejich vývoji.

Tato aplikace mi poskytla cenné zkušenosti s frameworkm Flutter, databází Firebase a tvorbou multiplatformního řešení. Především jsem si prohloubil znalosti v oblasti návrhu uživatelského rozhraní a práce s cloudovými službami. Věřím, že budoucí aktualizace přispějí k ještě lepší funkcionalitě a atraktivitě aplikace a že bude uživatelům sloužit jako spolehlivý nástroj pro zlepšení jejich tréninkové rutiny.

Kód je zcela dostupný zde: <https://github.com/RicnyF/Maturitni-projekt-RepTrack>

## SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] Firebase x Flutter Masterclass: Auth / Firestore / Storage / Hosting [Online] 14. 4. 2024 [cit. 2024-10-1]. Dostupné z: <https://www.youtube.com/watch?v=0RwLaJxW70c&t=1364s>
- [2] Email Sign Up • Flutter Auth Tutorial [Online] 22. 12. 2022 [cit. 2024-10-20]. Dostupné z: <https://www.youtube.com/watch?v=q1Vj-0vpaW0>
- [3] Flutter documentation [Online] [cit. 2024-09-26]. Dostupné z: <https://docs.flutter.dev/>
- [4] Reset Password • Firebase x Flutter Tutorial [Online] 2. 5. 2022 [cit. 2024-09-23]. Dostupné z: [https://www.youtube.com/watch?v=Sp4\\_2zi0kZg](https://www.youtube.com/watch?v=Sp4_2zi0kZg)
- [5] Flutter Crash Course - Net Ninja [Online] 13. 3. 2024 [cit. 2024-09-22]. Dostupné z: [https://www.youtube.com/watch?v=j\\_rCDc\\_X-k8&list=PL4cUxeGkcC9giLV](https://www.youtube.com/watch?v=j_rCDc_X-k8&list=PL4cUxeGkcC9giLV)
- [6] Dart Crash Course - Net Ninja [Online] 8. 1. 2024 [cit. 2024-09-21]. Dostupné z: <https://www.youtube.com/watch?v=QGqMJzywas>
- [7] Getting Started with Firebase 9 [Online] 8. 11. 2021 [cit. 2024-10-3]. Dostupné z: <https://www.youtube.com/watch?v=9zdvmgGsww>
- [8] Your First Flutter App [cit. 2024-09-24]. Dostupné z: <https://codelabs.developers.google.com/codelabs/flutter-codelab-first#3>