

DELTA – Střední škola informatiky a ekonomie, s.r.o.  
Ke Kamenci 151, Pardubice

# ZappaTrack

Kováč, Šimon

4.A

Programování

2025/2026

Prohlašuji, že jsem maturitní projekt vypracoval(a) samostatně, výhradně s použitím uvedené literatury.

V Pardubicích 31. 3. 2026

## **Zadání maturitního projektu z informatických předmětů**

Jméno a příjmení: *Šimon Kováč*

Pro školní rok: *2025/2026*

Třída: *4. A*

Obor: *Informační technologie 18-20-M/01*

Téma práce: ZapaTrack – webová aplikace pro sledování a sdílení zapalovačů mezi uživateli

Vedoucí práce: doc. Mgr. Josef Horálek, Ph.D.

### **Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:**

Cílem projektu je vytvořit android aplikaci, která umožní registrovaným uživatelům sledovat historii, vlastníky a umístění konkrétních zapalovačů, včetně sdílení fotografií a komentářů. Projekt simuluje malou sociální síť s profilovými stránkami uživatelů i zapalovačů, možností jejich „předávání“ mezi lidmi, evidencí změn vlastnictví a tvorbou statistik. Zároveň zábavnou formou simuluje principy jednoduché komunitní aplikace.

Funkce systému:

- Registrace a přihlášení uživatelů (autentizace)
- Profil uživatele: aktuální zapalovače, historie zapalovačů, galerie, komentáře
- Profil zapalovače: jméno, fotografie, aktuální a bývalí vlastníci, historie předávání, komentáře
- Možnost přidat nový zapalovač (včetně fotky a názvu)
- Předání zapalovače jinému uživateli (nutná autorizace – např. pouze aktuální vlastník může předat)
- Komentářový systém (na profilu zapalovače i uživatele)
- Galerie obrázků zapalovače
- Statistiky a žebříčky:
  - Nejvíce předaných zapalovačů
  - Uživatelé s nejvíce zapalovači
- Zapalovače s nejvíce fotkami apod.
- Ověřování přes e-mail, reset hesla, notifikace, API

Technické požadavky:

- Backend: PHP / Node.js / Python (např. Flask/Django) / jiné
- Databáze: MySQL / SQLite / PostgreSQL
- Frontend: Java (může využívat framework např. Bootstrap, Vue, React)

### **Stručný časový harmonogram (s daty a konkretizovanými úkoly):**

- Září – Plánování a analýza:
  - upřesnit a finálně schválit zadání
  - definovat hlavní funkce aplikace (MVP – minimální funkční verze),

- navrhnout základní datový model (tabulky: uživatel, zapalovač, historie, komentář, fotka),
- vybrat technologii (např. Django + SQLite, nebo PHP + MySQL),
- vytvořit wireframy (náčrtky obrazovek: profil, přihlášení...)

***Výstup: Zadání, návrh funkcí, ER diagram/tabulka, první skici UI***

- Říjen – Základ aplikace a databáze:
  - nastavit vývojové prostředí (repozitář na GitHubu)
  - implementovat registraci a přihlášení uživatele
  - vytvořit databázový model a napojit backend
  - založit základní administraci a první šablony (např. výpis zapalovačů)

***Výstup: Fungující přihlášení, registrace, databázová struktura***

- Listopad – Práce s obsahem a profily:
  - profil uživatele: seznam aktuálních a minulých zapalovačů
  - vytvoření a úprava zapalovače (název, foto)
  - historie předání zapalovače – jen vlastník může převést
  - detailní stránka zapalovače

***Výstup: CRUD operace pro zapalovače, přehled uživatelů, předávání***

- Prosinec – Galerie, komentáře, UI vylepšení:
  - přidávání více fotek k zapalovači
  - komentáře pod zapalovačem i pod profilem uživatele
  - úprava vzhledu (např. Bootstrap, dark/light tema)
  - testování uživatelských akcí

***Výstup: Přívětivé rozhraní, funkční galerie a komentáře***

- Leden – Statistiky a žebříčky:
  - ladění, vyčištění a optimalizace kódu

***Výstup: Statistické přehledy, připravený backend na finální fázi***

- Únor – Testování a dokumentace:
  - testování všech funkcí (včetně edge case – např. neoprávněné převzetí)
  - napsání dokumentace projektu (popis systému, databáze, funkcí)
  - připravení výstupů do praktické části (snímky obrazovek, výpisy kódu)

***Výstup: Plně funkční aplikace, první verze kompletní dokumentace***

- Březen – Finální příprava:
  - odevzdatelná tištěná i elektronická dokumentace
  - příprava na obhajobu – např. PowerPoint s ukázkou aplikace
  - zálohování projektu a finální kontrola

***Výstup: Kompletně odevzdaný projekt + připravenost na obhajobu***

Děkuji doc. Mgr. Josefu Horálkovi, Ph.D. za odborné vedení, pravidelné konzultace a zpětnou vazbu při zpracování maturitního projektu. Dále děkuji ing. Martinu Zmítkoví Ph.D. za výuku základů vývoje Android aplikací v předmětu Cvičení z programování.

Česky:

Android, Java, Supabase, PostgreSQL, REST API, autentizace, mobilní aplikace, sociální síť

ZappaTrack je android aplikace sloužící k jednoduchému sledování a sdílení sbírky zapalovačů mezi přáteli. Hlavními funkcemi jsou ukládání zapalovačů s fotkami a popisem na svůj profil, přidávání komentářů na své i cizí zapalovače a předávání zapalovačů mezi uživateli.

English:

Android, Java, Supabase, PostgreSQL, REST API, authentication, mobile application, social site

ZappaTrack is an Android app designed to make it easy to track and share your lighter collection with friends. Its main features include saving lighters with photos and descriptions to your profile, adding comments to your own and others' lighters, and trading lighters with other users.

## Contents

1. Úvod .....	9
1.1 Název aplikace.....	9
1.2 Účel aplikace .....	9
1.3 Cílová skupina uživatelů.....	9
2. Analýza a návrh .....	9
2.1 Funkční požadavky.....	9
2.2 Diagram případů užití (Use Case Diagram) .....	9
2.3 ER diagram databáze .....	11
2.4 Použité technologie .....	12
3. Databázový model .....	13
3.1 Popis tabulek .....	13
3.2 Vztahy mezi tabulkami.....	16
3.3 RLS policie.....	18
4. Architektura aplikace.....	20
5. Popis obrazovek .....	21
5.1 Přihlašovací obrazovka (LoginActivity).....	21
5.2 Domovská obrazovka (HomeActivity).....	22
5.3 Profil uživatele (ProfileActivity) .....	23
5.4 Detail zapalovače (LighterDetailActivity) .....	24
5.5 Přidání zapalovače (AddLighterActivity) .....	25
5.6 Vyhledávání (SearchActivity).....	26
5.7 Žebříčky (LeaderboardsActivity).....	27
5.8 Přátelé a oblíbené (FavoritesActivity) .....	28
6. Implementace klíčových funkcí.....	29
6.1 Autentizace uživatelů .....	29
6.2 Správa zapalovačů .....	29
6.3 Systém oblíbených.....	30
6.4 Přenos vlastnictví.....	30
6.5 Komentáře .....	30

7. Uživatelská příručka.....	31
7.1 Instalace aplikace.....	31
7.2 Registrace a první přihlášení.....	32
7.3 Přidání zapalovače do sbírky.....	34
7.4 Správa profilu .....	35
7.5 Vyhledávání a prohlížení.....	37
7.6 Práce s oblíbenými .....	38
7.7 Přenos zapalovače jinému uživateli.....	40
7.8 Žebříčky .....	42
8. Závěr .....	43
8.1 Shrnutí projektu .....	43
8.2 Možná budoucí rozšíření.....	44
8.3 Získané zkušenosti .....	44
9. Přílohy .....	45
9.1 Snímky obrazovek .....	46
9.2 Zdrojový kód (vybrané části) .....	55
9.3 Seznam použitých knihoven .....	56
9.4 Odkazy a zdroje.....	57

## **1. Úvod**

### **1.1 Název aplikace**

ZappaTrack

### **1.2 Účel aplikace**

ZappaTrack je android aplikace sloužící k jednoduchému sledování a sdílení sbírky zapalovačů mezi přáteli. Hlavními funkcemi jsou ukládání zapalovačů s fotkami a popisem na svůj profil, přidávání komentářů na své i cizí zapalovače a předávání zapalovačů mezi uživateli.

### **1.3 Cílová skupina uživatelů**

Skupiny přátel se zálibou ve sbírání zapalovačů. Teoreticky by se aplikace dala využít na sledování sbírek i jiných předmětů, ale v UI aplikace jsou předměty referovány jako zapalovače.

## **2. Analýza a návrh**

### **2.1 Funkční požadavky**

Registrace a přihlášení uživatelů

Správa sbírky zapalovačů

Vyhledávání uživatelů a zapalovačů

Systém oblíbených (přátelé a zapalovače)

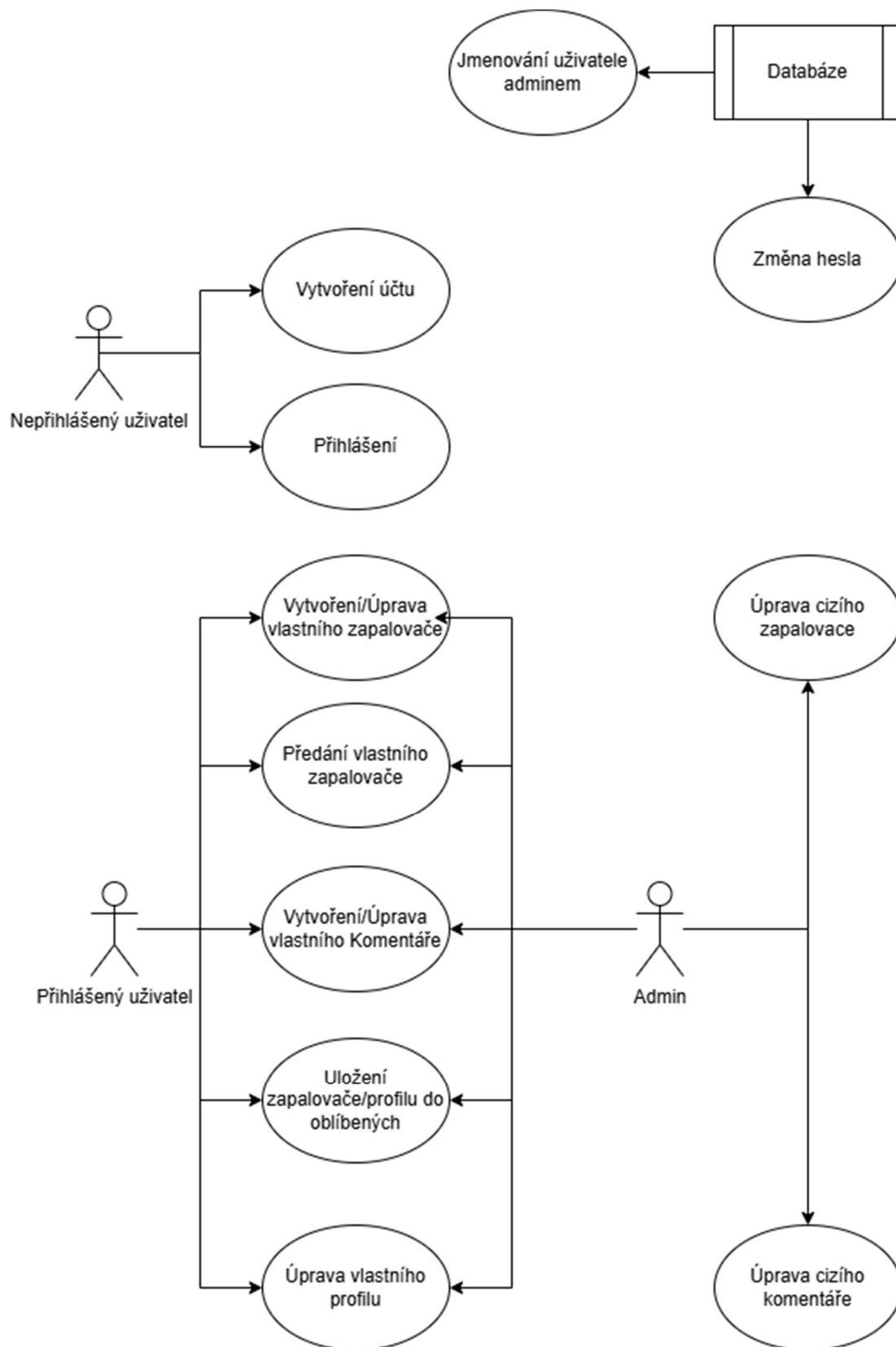
Přenos vlastnictví zapalovačů

Komentáře

Žebříčky

### **2.2 Diagram případů užití (Use Case Diagram)**

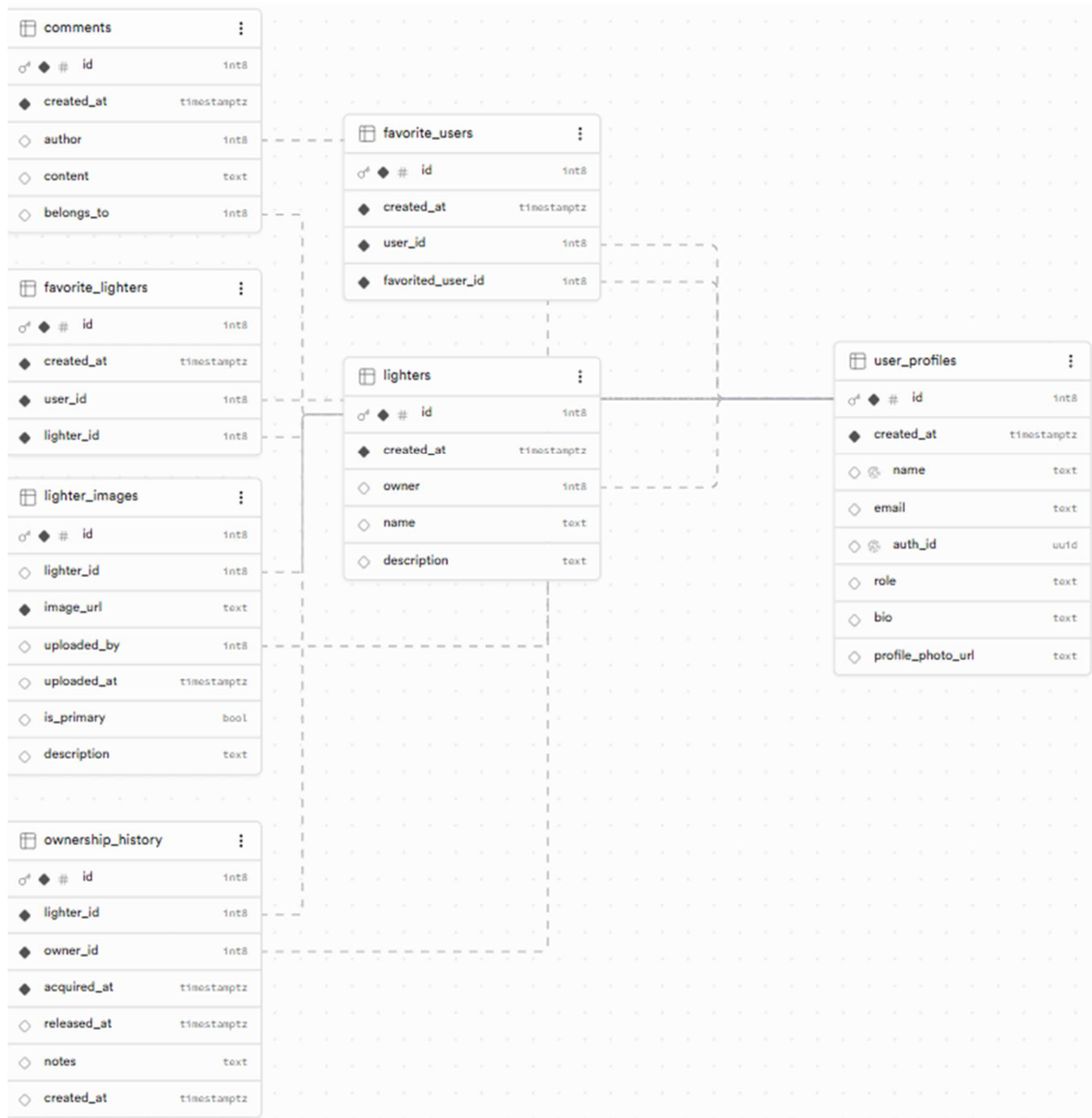
Vycházejte z Obr. 1. Nepřihlášený uživatel nemá žádné pravomoci. Po přihlášení si uživatel může upravit profil, sledovat jiné uživatele a manipulovat se zapalovači. Uživatel může kromě přidání do oblíbených a psaní komentáře upravovat pouze své vlastní zapalovače. Admin, který se aktuálně dá udělit pouze prostřednictvím rozhraní Supabase, má navíc práva upravovat a mazat komentáře jiných uživatelů. Přes Supabase je navíc řešena i změna hesla, jelikož by jinak bylo nutné vytvořit a hostovat webovou stránku.



Obrázek 1 Use Case Diagram

## 2.3 ER diagram databáze

Tabulky a závislosti jsou dále rozepsány v sekci Databázový model



Obrázek 2 Diagram databáze

## 2.4 Použité technologie

Android Studio - Prostředí pro vývoj Android aplikací jehož součástí je i emulátor android zařízení. Jednoduché pro užití, zdarma. [3]

Java - Objektivě orientovaný programovací jazyk vhodný pro vývoj mobilních aplikací. Jednoduchý na naučení se spoustou veřejně dostupných zdrojů.[1]

Supabase - Zdarma dostupný service umožňující hosting Postgres databáze, ukládání souborů a Auth service. Využil jsem i funkce storage bucketů pro ukládání fotografií a RLS manager pro přidělení potřebných oprávnění. Supabase jsem vybral z toho důvodu, že s ním mám předešlé zkušenosti. [2]

Supabase Auth - Autentifikační a Autorizační služba poskytována Supabase umožňující přihlašování uživatelů pomocí emailu a jiných platform. V ZappaTracku je využito přihlašování přes email. [2]

### 3. Databázový model

#### 3.1 Popis tabulek

##### user\_profiles

Ukládá informace o uživatelských profilech. Hodnota *auth\_id* odkazuje na tabulku autentizačního systému Supabase. Uživatelské profily nelze ukládat přímo tam, jelikož Supabase nedovoluje Auth tabulky upravovat. Profilová fotka je uložena do storage bucketu profile-photos a její URL adresa uložena v hodnotě *profile\_photo\_url*.

Sloupec	Typ	Popis
id	bigint	Automaticky generovaný primární klíč
created_at	timestampz	Datum a čas vytvoření profilu
name	text (UNIQUE)	Unikátní uživatelské jméno
email	text	E-mailová adresa uživatele
auth_id	uuid (UNIQUE)	ID z autentizačního systému Supabase
role	text	Role uživatele: 'user' nebo 'admin'
bio	text	Krátký popis uživatele (max. 200 znaků)
profile_photo_url	text	URL adresa profilové fotografie

Tabulka 1 - Tabulka user\_profiles

##### lighters

Ukládá informace o zapalovačích.

Sloupec	Typ	Popis
id	bigint	Automaticky generovaný primární klíč
created_at	timestampz	Datum přidání zapalovače
owner	bigint (FK)	Odkaz na aktuálního vlastníka (user_profiles.id)
name	text	Název zapalovače
description	text	Popis zapalovače

Tabulka 2 - Tabulka lighters

## lighter\_images

Ukládá fotografie zapalovačů. Umožňuje ukládat více fotografií pro jeden zapalovač. Fotografie jsou uloženy v storage bucketu [13] lighter-images a jejich URL adresa uložena v hodnotě *image\_url*.

Sloupec	Typ	Popis
id	bigint	Automaticky generovaný primární klíč
lighter_id	bigint (FK)	Odkaz na zapalovač (lighters.id)
image_url	text	URL adresa fotografie
uploaded_by	bigint (FK)	Kdo fotografii nahrál (user_profiles.id)
uploaded_at	timestampz	Datum a čas nahrání
is_primary	boolean	Zda je fotografie hlavní (výchozí: false)
description	text	Volitelný popis fotografie

Tabulka 3 - Tabulka lighter\_images

## comments

Ukládá komentáře.

Sloupec	Typ	Popis
id	bigint	Automaticky generovaný primární klíč
created_at	timestampz	Datum a čas přidání komentáře
author	bigint (FK)	Autor komentáře (user_profiles.id)
content	text	Text komentáře
belongs_to	bigint (FK)	Zapalovač, ke kterému komentář patří (lighters.id)

Tabulka 4 - Tabulka comments

## favorite\_users

Ukládá oblíbené profily jednotlivých uživatelů.

Sloupec	Typ	Popis
id	bigint	Automaticky generovaný primární klíč
created_at	timestampz	Datum přidání
user_id	bigint (FK)	Uživatel, který sleduje (user_profiles.id)
favorited_user_id	bigint (FK)	Sledovaný uživatel (user_profiles.id)

Tabulka 5 - Tabulka favorite\_users

## favorite\_lighters

Ukládá oblíbené zapalovače jednotlivých uživatelů.

Sloupec	Typ	Popis
id	bigint	Automaticky generovaný primární klíč
created_at	timestamptz	Datum přidání
user_id	bigint (FK)	Uživatel, který přidal oblíbený (user_profiles.id)
lighter_id	bigint (FK)	Oblíbený zapalovač (lighters.id)

Tabulka 6 - Tabulka favorite\_lighters

## ownership\_history

Ukládá historii vlastnictví každého zapalovače. Při přenosu se zaznamená datum předání novému vlastníkovi (*released\_at*) a vytvoří se nový záznam pro nového vlastníka. Poznámky u přenosu nejsou aktuálně implementovány, jednalo se především od debugovací funkci.

Sloupec	Typ	Popis
id	bigint	Automaticky generovaný primární klíč
lighter_id	bigint (FK)	Zapalovač (lighters.id)
owner_id	bigint (FK)	Vlastník v daném období (user_profiles.id)
acquired_at	timestamptz	Kdy vlastník zapalovač získal
released_at	timestamptz	Kdy zapalovač předal dál (NULL = aktuální vlastník)
notes	text	Volitelná poznámka k přenosu
created_at	timestamptz	Datum vytvoření záznamu

Tabulka 7 - Tabulka ownership\_history

### 3.2 Vztahy mezi tabulkami

Tabulky v databázi se navzájem propojují pomocí cizích klíčů. Cizí klíč je sloupec v jedné tabulce, který odkazuje na záznam v jiné tabulce. Například tabulka *lighters* má sloupec *owner*, který obsahuje ID uživatele z tabulky *user\_profiles*. Díky tomu aplikace ví, kdo daný zapalovač vlastní.

Většina vztahů je typu 1:N, což znamená že jeden záznam může mít mnoho navázaných záznamů v druhé tabulce. Například jeden uživatel může mít mnoho zapalovačů, ale každý zapalovač má právě jednoho vlastníka. Vztahy M:N jsou složitější. Například jeden uživatel může sledovat mnoho jiných uživatelů a zároveň jeden uživatel může být sledován mnoha uživateli. Takový vztah nelze uložit přímo, proto existuje vazební tabulka *favorite\_users*, která tento vztah zprostředkovává.

Zdrojová tabulka	Cílová tabulka	Typ vztahu	Popis
user_profiles	lighters	1 : N	Jeden uživatel může vlastnit mnoho zapalovačů
lighters	lighter_images	1 : N	Jeden zapalovač může mít mnoho fotografií
lighters	comments	1 : N	Jeden zapalovač může mít mnoho komentářů
user_profiles	comments	1 : N	Jeden uživatel může napsat mnoho komentářů
user_profiles	user_profiles	M : N	Uživatelé se mohou navzájem sledovat (přes favorite_users)
user_profiles	lighters	M : N	Uživatel může mít mnoho oblíbených zapalovačů (přes favorite_lighters)
lighters	ownership_history	1 : N	Jeden zapalovač má záznamy o všech svých vlastnících
user_profiles	ownership_history	1 : N	Uživatel může mít mnoho záznamů o přenosu vlastnictví.
user_profiles	lighter_images	1:N	Jeden uživatel může uploadnout mnoho fotografií.

Tabulka 8 - Vztahy mezi tabulkami



### 3.3 RLS policies

RLS policies slouží k zamezení přístupu do částí tabulky neoprávněným uživatelům. Např. uživatel smí upravovat jen své vlastní zapalovače. Když je RLS na tabulce zaplé, není možné do tabulky zasahovat bez oprávnění. [9]

RLS policies se vytváří v rozhraní Supabase a vypadají takto:

```
alter policy "Users can delete own comments, admins can delete any"
on "public"."comments"
to authenticated
using (
owns_or_admin(author)
);
```

Tato policy zaručuje, že komentáře může mazat pouze autor komentáře.

Využívá funkci owns\_or\_admin která vypadá takto:

```
BEGIN
RETURN EXISTS (
SELECT 1
FROM public.user_profiles
WHERE auth_id = auth.uid()
AND (
id = resource_owner_id -- User owns the resource
OR role = 'admin' -- User is admin
)
);
END;
```

Tato funkce vezme profil z tabulky public.user\_profiles, ve kterém se shoduje auth\_id které je do funkce posláno jako argument, v tomto využití auth.id právě přihlášeného uživatele. Pokud je profil vlastníkem komentáře nebo adminem, povolí akci.

Tímto způsobem jsou ošetřeny všechny tabulky.

Zde jsou další příklady těchto policí, jejich účel je vysvětlen jejich názvem:

```
alter policy "Anyone can view comments"
on "public"."comments"
to authenticated
using (
true
);
```

```
alter policy "Users can create own profile"
on "public"."user_profiles"
to public
with check (
((auth.uid())::text = (auth_id)::text)
);
```

```
alter policy "Users can delete own favorites"
on "public"."favorite_lighters"
to authenticated
using (
(auth.uid() = ( SELECT user_profiles.auth_id
FROM user_profiles
WHERE (user_profiles.id = favorite_lighters.user_id)))
);
```

#### 4. Architektura aplikace

Aplikace má celkem 10 obrazovek, z toho 8 hlavních. Další 2 se využívají k zobrazení všech fotek zapalovače a detailu fotografie zapalovače.

Jednotlivé předměty z databáze (zapalovače, profily, komentáře...) jsou v aplikaci reprezentovány classami. K tomuto je využívána knihovna Gson, která umožňuje převod Json souborů na Java classy a naopak.

Dále jsou využívány adaptéry, které jsou v podobě class. Adaptéry slouží k zobrazování seznamů předmětů pomocí RecyclerView [6]. RecyclerView slouží k efektivnímu vytváření scrollovatelných seznamů, kde se nevyužívané řádky vymažou a reálně načtené jsou pouze ty potřebné. Adaptéry jsou pro tuto funkci RecyclerView nezbytné, neboť kontrolují správné fungování listů a načítání dat. Dále ovládá zobrazování správných tlačítek. Např. u komentáře se tlačítko "delete" zobrazí pouze pokud je komentář váš. Reálné mazání komentářů je však ochráněno RLS na databázi, takže tato funkce je čistě kosmetická.

Autentizace uživatele a interakce s Supabase Auth databází je ovládána classou SupabaseAuthManager. Ta je zodpovědná za správnou funkci přihlašování. Classa je napsána jako singleton [10], aby se předešlo problémům s auth tokenem a dalšími informacemi.

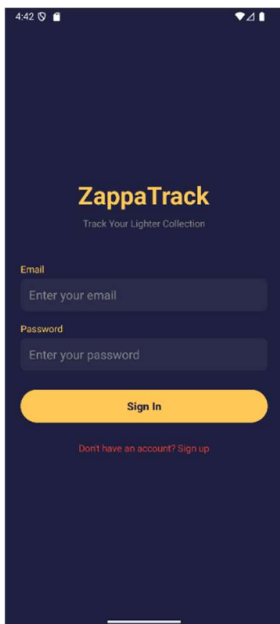
Aktuální session je uložena v classe AuthSession, která obsahuje informace o právě přihlášeném uživateli.

Interakce s databází mimo autentizaci je ovládána classou SupabaseClient. Ta se stará o veškerou komunikaci s databází a obsahuje všechny metody k tomu určené.

Autorizace je řešena prostřednictvím RLS policí na straně databáze.

## 5. Popis obrazovek

### 5.1 Přihlašovací obrazovka (LoginActivity)



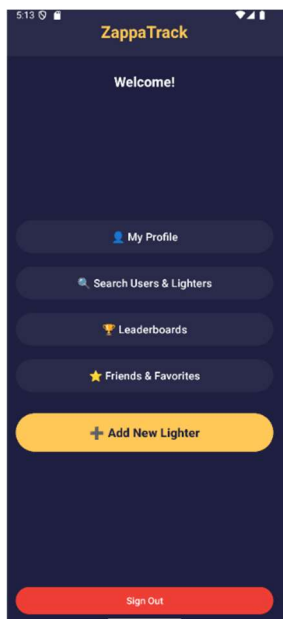
Obrázek 3 - Přihlašovací obrazovka

Tato obrazovka slouží jako vstupní bod pro nepřihlášené uživatele. Obsahuje přepínač mezi režimem přihlášení a registrace.

Prvky obrazovky:

- Pole pro zadání e-mailu
- Pole pro zadání hesla
- Pole pro zadání uživatelského jména (zobrazí se pouze při registraci)
- Tlačítko Přihlásit / Registrovat
- Odkaz pro přepnutí mezi režimy

## 5.2 Domovská obrazovka (HomeActivity)

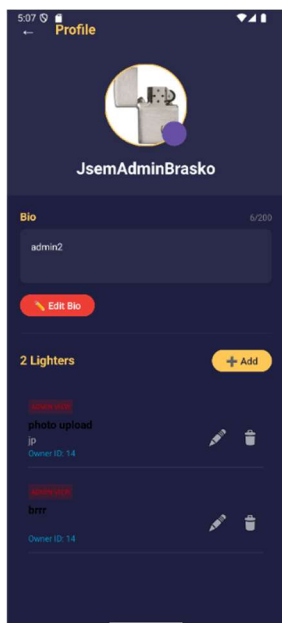


Obrázek 4 - Hlavní obrazovka

Jednoduchá navigační obrazovka s tlačítky vedoucími do hlavních sekcí aplikace:

- Přidat zapalovač
- Můj profil
- Vyhledávání
- Oblíbení
- Žebříčky
- Odhlásit se

### 5.3 Profil uživatele (ProfileActivity)



Obrázek 5 - Obrazovka profilu

Obrazovka zobrazuje profil uživatele, vlastní nebo cizí. Obsah se liší podle toho, zda si uživatel prohlíží svůj vlastní profil.

Zobrazené informace:

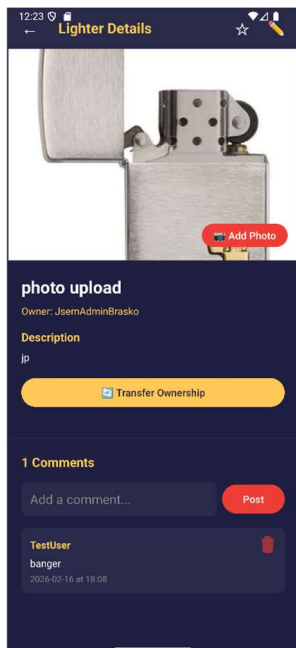
- Profilová fotografie
- Uživatelské jméno
- Bio (popis, max. 200 znaků)
- Seznam zapalovačů uživatele

Dostupné akce (pouze pro vlastní profil):

- Změna profilové fotografie
- Úprava bio
- Přidání nového zapalovače

Na cizím profilu je navíc tlačítko pro přidání/odebrání z oblíbených.

## 5.4 Detail zapalovače (LighterDetailActivity)



Obrázek 6 - Obrazovka detailu zapalovače

Detailní pohled na konkrétní zapalovač. Obsahuje:

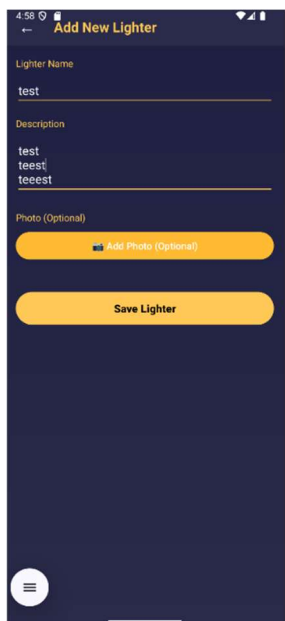
- Název a popis zapalovače
- Jméno aktuálního vlastníka
- Hlavní fotografii a galerii dalších fotografií
- Sekci komentářů
- Tlačítko pro přidání do oblíbených

Akce dostupné vlastníkovi (nebo adminovi):

- Úprava názvu a popisu
- Přidání fotografie
- Přenos vlastnictví na jiného uživatele
- Smazání zapalovače

Klepnutím na fotografii je dále možné zobrazit všechny fotografie tohoto zapalovače.

## 5.5 Přidání zapalovače (AddLighterActivity)



Obrázek 7 - Obrazovka přidání zapalovače

Formulář pro vytvoření nového záznamu zapalovače. Obsahuje:

- Pole pro název zapalovače
- Pole pro popis
- Možnost vybrat a nahrát fotografii z galerie zařízení
- Tlačítko pro uložení

Po uložení je zapalovač přiřazen aktuálně přihlášenému uživateli jako vlastníkovvi.

## 5.6 Vyhledávání (SearchActivity)



Obrázek 8 - Obrazovka vyhledávání

Obrazovka umožňuje vyhledávání ve dvou kategoriích současně:

- Zapalovače - vyhledávání podle názvu
- Uživatelé - vyhledávání podle uživatelského jména

Vyhledávaný text zároveň hledá i v biu uživatelů. Vyhledávání se spouští automaticky při psaní s 500ms zpožděním aby se zabránilo přílišnému zasílání požadavků při rychlém psaní. Výsledky jsou zobrazeny v oddělených sekcích.

K vyhledávání se využívá Wildcard search. Zadaný výraz se vyhledá v tabulkách *lighters* a *user-profiles* a každá shoda se odešle zpět. Pro velikost projektu je toto dostačující, ale neumožňuje to širší funkce jako relevance výsledků nebo hledání s překlepem.

Vyhledávání není case-sensitive. [8]

## 5.7 Žebříčky (LeaderboardsActivity)

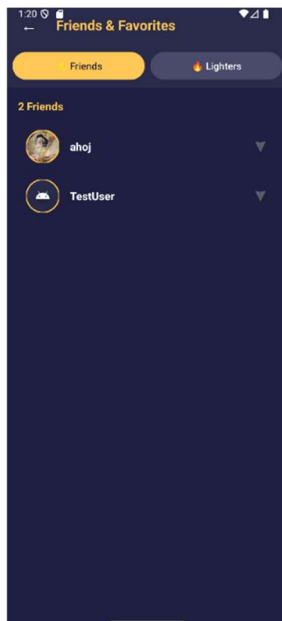


Obrázek 9 - Obrazovka žebříčeků

Obrazovka zobrazuje dva žebříčky:

- Nejpřeváděnější zapalovače - zapalovače, které změnily vlastníka nejvícekrát
- Top sběratelé - uživatelé s největším počtem zapalovačů ve sbírce

## 5.8 Přátelé a oblíbené (FavoritesActivity)



Obrázek 10 - Obrazovka oblíbených

Obrazovka se dvěma záložkami:

- Přátelé - seznam uživatelů, které si přihlášený uživatel přidal mezi oblíbené
- Oblíbené zapalovače - seznam zapalovačů označených jako oblíbené

Z každé položky lze přejít na příslušný profil uživatele nebo detail zapalovače.

## 6. Implementace klíčových funkcí

### 6.1 Autentizace uživatelů

Aplikace využívá službu Supabase Auth, která zajišťuje autentizaci uživatelů. Tato funkce je v aplikaci implementována pomocí classy SupabaseAuthManager, která je navržena jako singleton, takže existuje pouze jedna její instance sdílená v rámci celé aplikace. To je provedeno pro uložení dat mezi obrazovkami.

Při registraci uživatel zadá e-mail, heslo a jméno profilu. Tyto údaje jsou odeslány na server, kde dojde k vytvoření nového uživatelského účtu. Přihlášení probíhá zadáním e-mailu a hesla.

Po úspěšném ověření:

1. aplikace uloží informace o přihlášení (session)
2. načte zbylé údaje o uživateli z databáze (např. jméno nebo role)
3. a umožní uživateli přístup k hlavní části aplikace.

Informace o přihlášení jsou uchovávány po celou dobu běhu aplikace v classě SupabaseAuthManager, takže není nutné opakované přihlašování. V průběhu komunikace je využíván JWT token k autentizaci dotazu. [\[11\]](#)

Při odhlášení dojde ke zrušení aktuální session a uživatel je přesměrován zpět na přihlašovací obrazovku.

### 6.2 Správa zapalovačů

#### Vytvoření

Uživatel vyplní název, popis a volitelně přidá fotografii. Zapalovač se uloží do tabulky lighters s odkazem na aktuálního vlastníka. Pokud byla vybrána fotografie, nahraje se do Supabase Storage a její URL se uloží do tabulky lighter\_images.

#### Úprava

Název a popis zapalovače může měnit pouze jeho aktuální vlastník nebo administrátor.

#### Mazání

Před smazáním zapalovače se nejprve odstraní všechny jeho fotografie, komentáře a záznamy v oblíbených. Až pak se smaže samotný zapalovač.

#### Nahrávání fotografií

Fotografie se před odesláním převede na Base64 [\[12\]](#) řetězec a nahraje do storage bucketu lighter-images. Výsledná URL adresa se uloží do tabulky lighter\_images. Stejný postup platí pro profilové fotografie, s rozdílem, že se uloží do bucketu profile\_photos a do tabulky user-profiles. Pro zobrazování fotografií v aplikaci se používá knihovna Picasso.

## 6.3 Systém oblíbených

### Přátelé

Klepnutím na tlačítko na profilu uživatele se vytvoří záznam v tabulce favorite\_users. Opětvným klepnutím se záznam smaže. Tlačítko vždy zobrazuje aktuální stav.

### Oblíbené zapalovače

Funguje stejně, jen přes tabulku favorite\_lighters. Oblíbené lze spravovat z detailu zapalovače nebo ze sekce Oblíbení.

## 6.4 Přenos vlastnictví

Přenos probíhá ve třech krocích:

- Aplikace vyhledá nového vlastníka podle uživatelského jména
- V tabulce lighters se aktualizuje sloupec owner na ID nového vlastníka
- V tabulce ownership\_history se uzavře záznam předchozího vlastníka (vyplní se released\_at) a vytvoří se nový záznam pro nového vlastníka

## 6.5 Komentáře

Komentáře se načítají společně se jménem autora pomocí dotazu na Supabase REST API. Nový komentář se odešle POST požadavkem a seznam komentářů se ihned aktualizuje. Administrátor může jakýkoliv komentář smazat.

## **7. Uživatelská příručka**

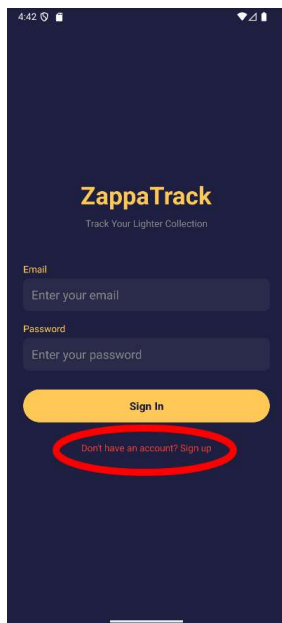
### **7.1 Instalace aplikace**

Aplikace je distribuována jako APK soubor. Postup instalace:

1. V nastavení Android zařízení povolte instalaci aplikací z neznámých zdrojů
2. Stáhněte soubor ZappaTrack.apk do zařízení.
3. Otevřete soubor APK a potvrďte instalaci.
4. Po instalaci spusťte aplikaci ZappaTrack.

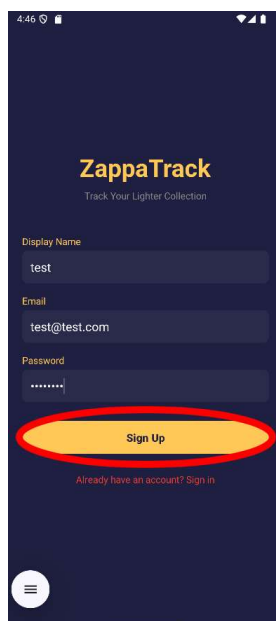
## 7.2 Registrace a první přihlášení

1. Na přihlašovací obrazovce klepněte na odkaz "Don't have an account? Sign up".



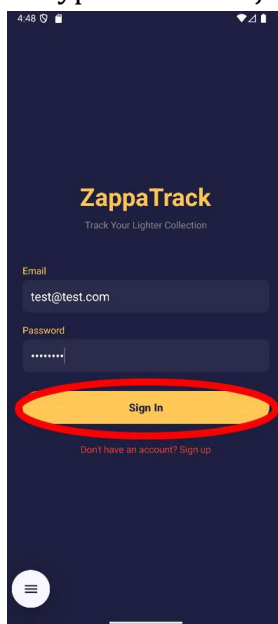
Obrázek 11 - Klepneme na "Don't have an account? Sign up"

2. Vyplňte uživatelské jméno, e-mailovou adresu a heslo.
3. Klepněte na tlačítko "Sign up".



Obrázek 12 - Klepneme na "Sign up"

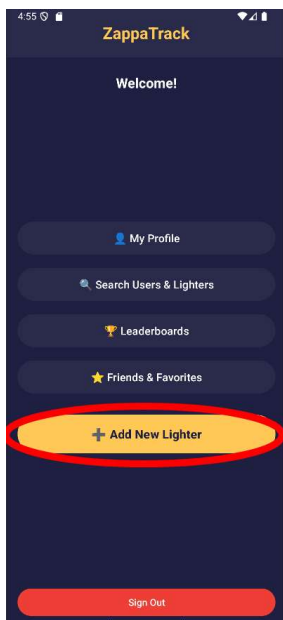
4. Vyplňte své údaje a přihlašte se klepnutím na tlačítko “Sign in“



Obrázek 13 - Klepněte na tlačítko "Sign in"

## 7.3 Přidání zapalovače do sbírky

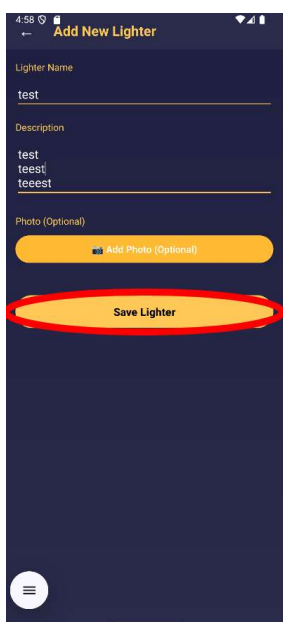
1. Na domovské obrazovce klepněte na "Add New Lighter".



Obrázek 14 - Klepneme na "Add New Lighter"

2. Zadejte název a popis zapalovače. Pokud chcete přidat fotografii, klepněte na "Add photo (optional)"

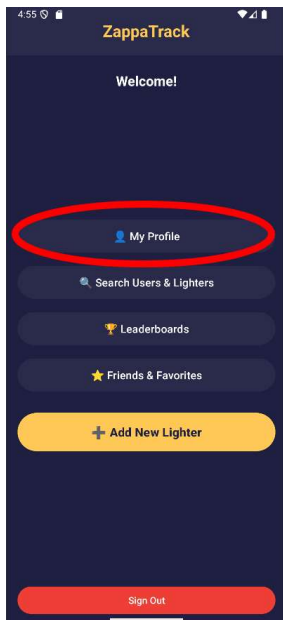
3. Klepněte na "Save lighter". Zapalovač se zobrazí ve vaší sbírce.



Obrázek 15 - Klepneme na "Save Lighter"

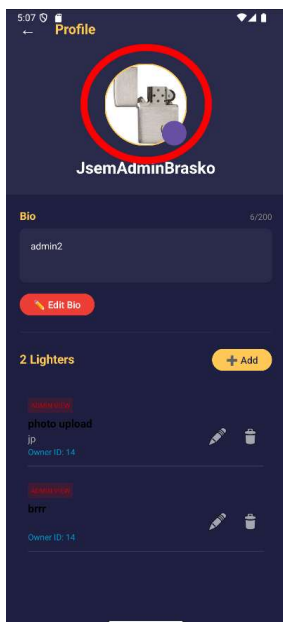
## 7.4 Správa profilu

Na domovské obrazovce klepněte na "My profile". Zde můžete:



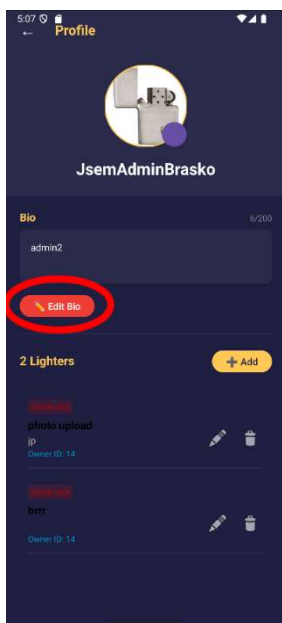
Obrázek 16 - Klepneme na "My profile"

- Klepnutím na profilovou fotografii můžete nahrát nový profilový snímek



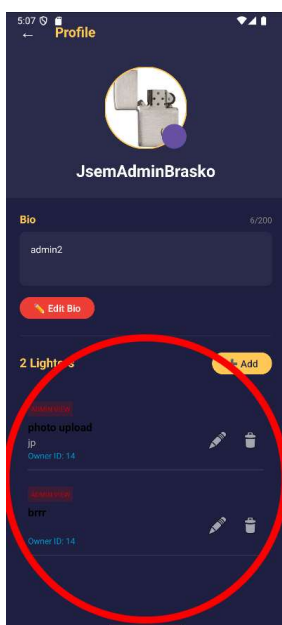
Obrázek 17 - Klepnutím na profilovou fotografii ji změňíme

- Klepnutím na "Edit Bio" přidat nebo změnit krátký popis profilu (max. 200 znaků)



Obrázek 18 - Klepneme na "Edit Bio"

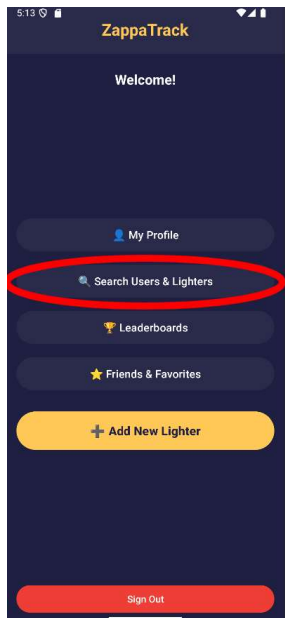
- Prohlížet všechny zapalovače ve vaší sbírce



Obrázek 19 - Zapalovače ve sbírce

## 7.5 Vyhledávání a prohlížení

1. Na domovské obrazovce klepněte na "Search users and lighters".



Obrázek 20 - Klepneme na "Search Users & Lighters"

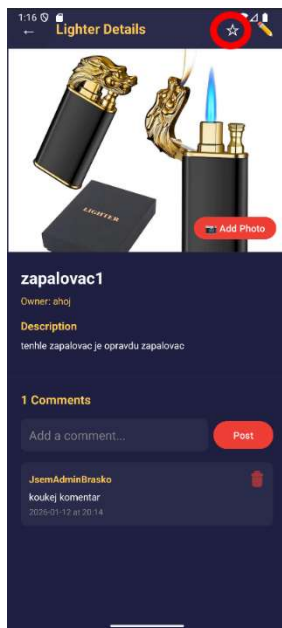
2. Do vyhledávacího pole zadejte hledaný výraz. Výsledky se zobrazí automaticky.



Obrázek 21 - Zadejte hledaný výraz

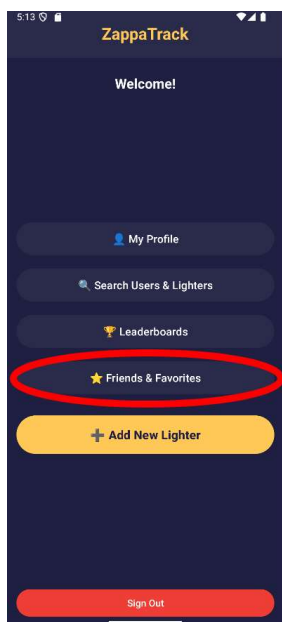
## 7.6 Práce s oblíbenými

Přítele přidáte tak, že navštívíte jeho profil a klepnete na tlačítko hvězdy v pravém horním rohu. Stejným způsobem lze přítele odebrat. Zapalovač přidáte mezi oblíbené stejným způsobem.



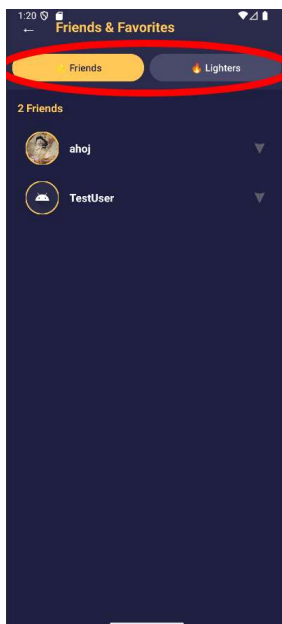
Obrázek 22 - Přidáme do oblíbených hvězdičkou

Všechny oblíbené jsou dostupné v sekci "Friends & Favorites" z domovské obrazovky.



Obrázek 23 - Klepneme an "Friends & Favorites"

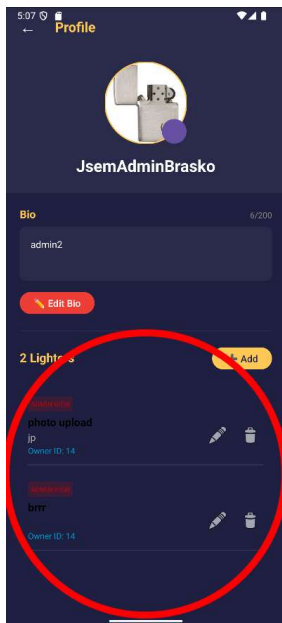
Mezi oblíbenými profily a zapalovači přepínáme tlačítka na horní části obrazovky.



Obrázek 24 - Přepínání mezi profily a zapalovači

## 7.7 Přenos zapalovače jinému uživateli

1. Otevřete detail zapalovače, který chcete převést.



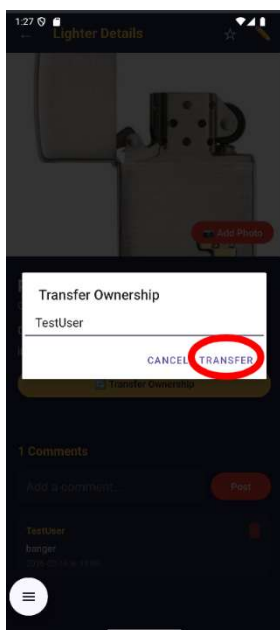
Obrázek 25 - Vybereme zapalovač

2. Klepněte na tlačítko "Transfer ownership".



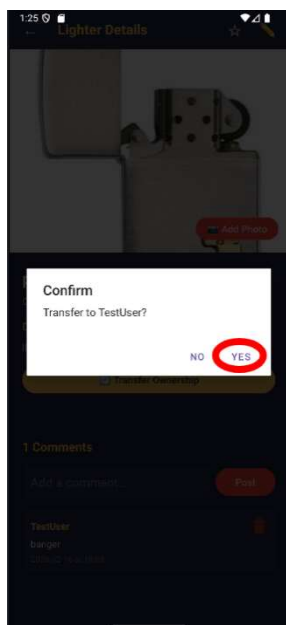
Obrázek 26 - Klepneme na "Transfer Ownership"

3. Zadejte uživatelské jméno nového vlastníka a klepněte na "Transfer". Jméno musí přesně odpovídat jménu cílového profilu.



Obrázek 27 - Zadejte jméno a klepněte na "Transfer"

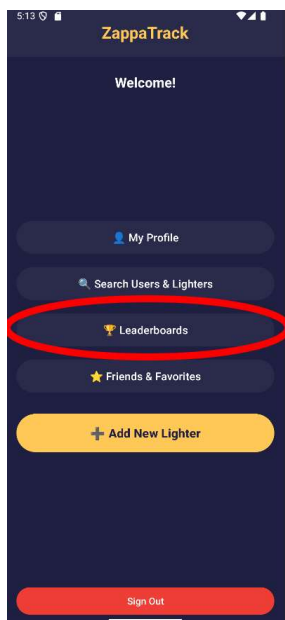
4. Potvrďte přenos. Zapalovač zmizí z vaší sbírky a přejde do sbírky nového vlastníka.



Obrázek 28 - Klepněte na "Yes"

## 7.8 Žebříčky

V sekci Leaderboards můžete sledovat uživatele s nejvíce zapalovačema a nejvíce předávané zapalovače.



Obrázek 29 - Klepneme na "Leaderboards"



Obrázek 30 - Můžeme vidět žebříčky uživatelů a zapalovačů

## **8. Závěr**

### **8.1 Shrnutí projektu**

I přes téměř nulové předešlé zkušenosti s vývojem Android aplikací byly do projektu implementovány veškeré potřebné funkce. V průběhu práce se postupně měnila struktura databáze podle aktuálních potřeb a výsledná struktura je efektivnější než původní návrh. Díky předešlým zkušenostem s platformou Supabase bylo nastavování backendu intuitivní.

Největší výzvou byla platforma Android, kde bylo nutné se naučit základní principy vývoje Android aplikací, například registraci obrazovek v souboru AndroidManifest.xml. Organizace souborů v projektu byla provedena do standardní struktury balíčků rozdělených podle funkce (screens, adapters, databaseitems).

Projekt byl zároveň zpracován jako finální projekt do předmětu Cvičení z programování, kde byly osvojeny základy fungování Android aplikací. Vedoucím práce byl doc. Mgr. Josef Horálek, Ph.D., který poskytoval pravidelné konzultace, zpětnou vazbu a pomohl s časovým rozložením práce.

## 8.2 Možná budoucí rozšíření

- Webové rozhraní s responzivním designem pro iPhone uživatele.
- Historie majitelů zapalovače. Databáze již tyto data skladuje, ale nejsou zobrazovány v aplikaci. Jsou tam pouze prostřednictvím žebříčku nejvíce předávaných zapalovačů, kde je zobrazen jen počet předání.
- Širší admin pravomoci. Aktuálně jsou pravomoci admina velmi omezené, umožňující jen upravovat a mazat zapalovače a komentáře. Užitečné rozšíření by byla možnost zapalovače předávat, blokovat uživatele a upravovat fotografie a bio uživatelů. Funkce jmenování admina přímo v aplikaci je již z části napsána, ale není implementována z nedostatku času.
- UI rework. Aktuální design je velice jednoduchý a místy obsahuje nedokonalosti jako tlačítka bez textury (např. tlačítko u úpravy fotografie). Kompletní předělání by nebylo nutné, ale vítané.
- Popisky u error zpráv. Některá chybová oznámení (např. pokus o vytvoření uživatele jehož jméno je již zabrané) nejsou doplněna o dostatečně informující chybový text, místo toho zobrazí chybovou zprávu vrácenou databází.
- Základní sound-design

## 8.3 Získané zkušenosti

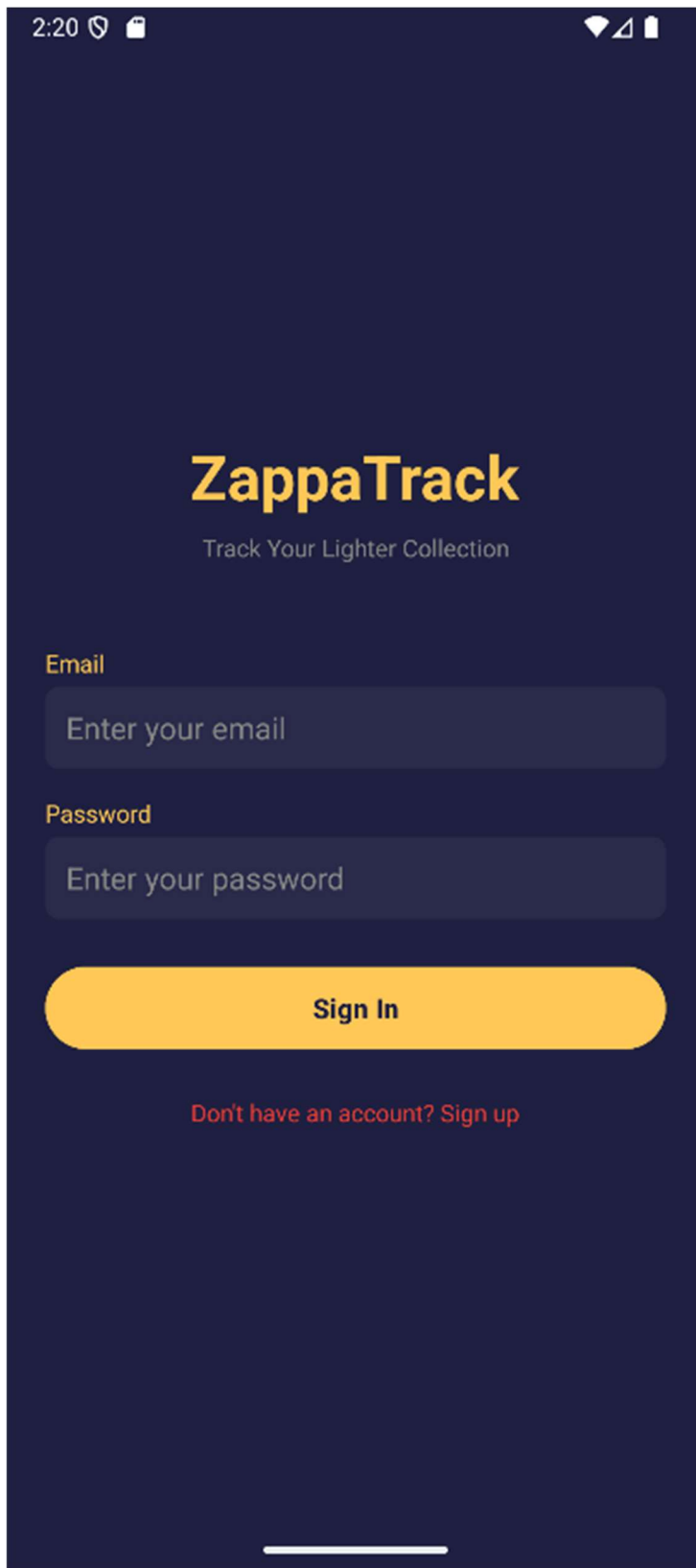
Naučil jsem se pracovat s vývojovým prostředím Android Studio a rozšířil své znalosti Supabase. Dále jsem se naučil využívat storage buckety na ukládání souborů a RLS policies pro bezpečnost databáze.

## 9. Přílohy

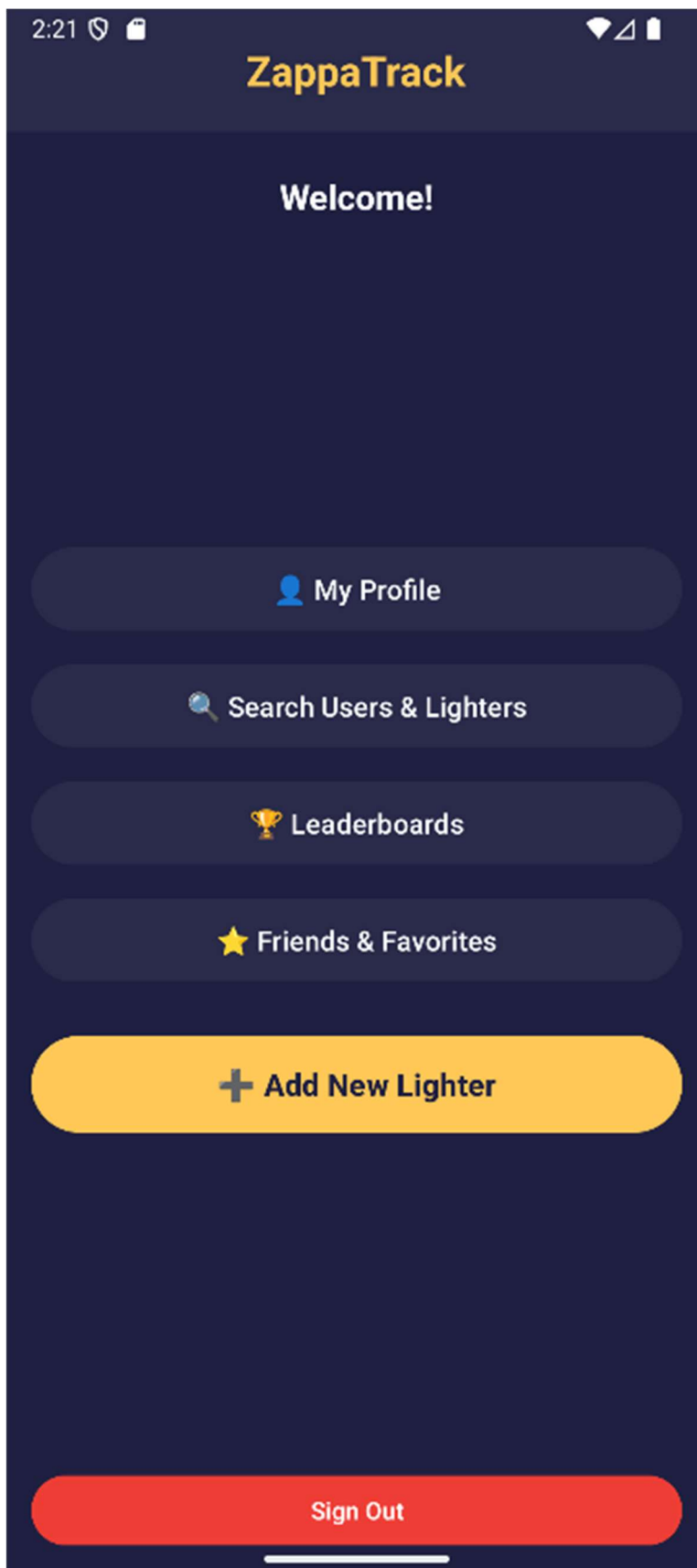
Odkaz na github repo: <https://github.com/simon-kovac/ZappaTrack>

Odkaz na Google Drive s .apk souborem a dokumentací.

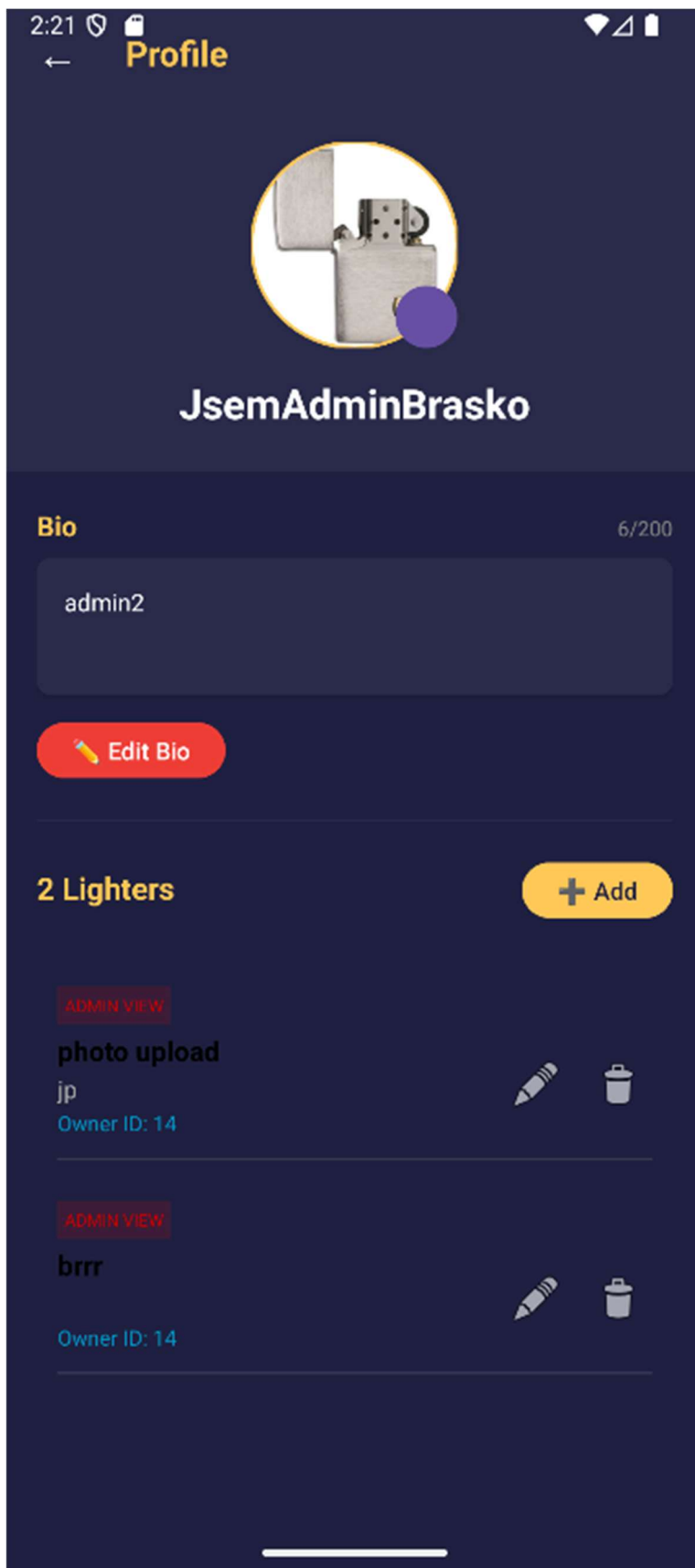
## 9.1 Snímky obrazovek



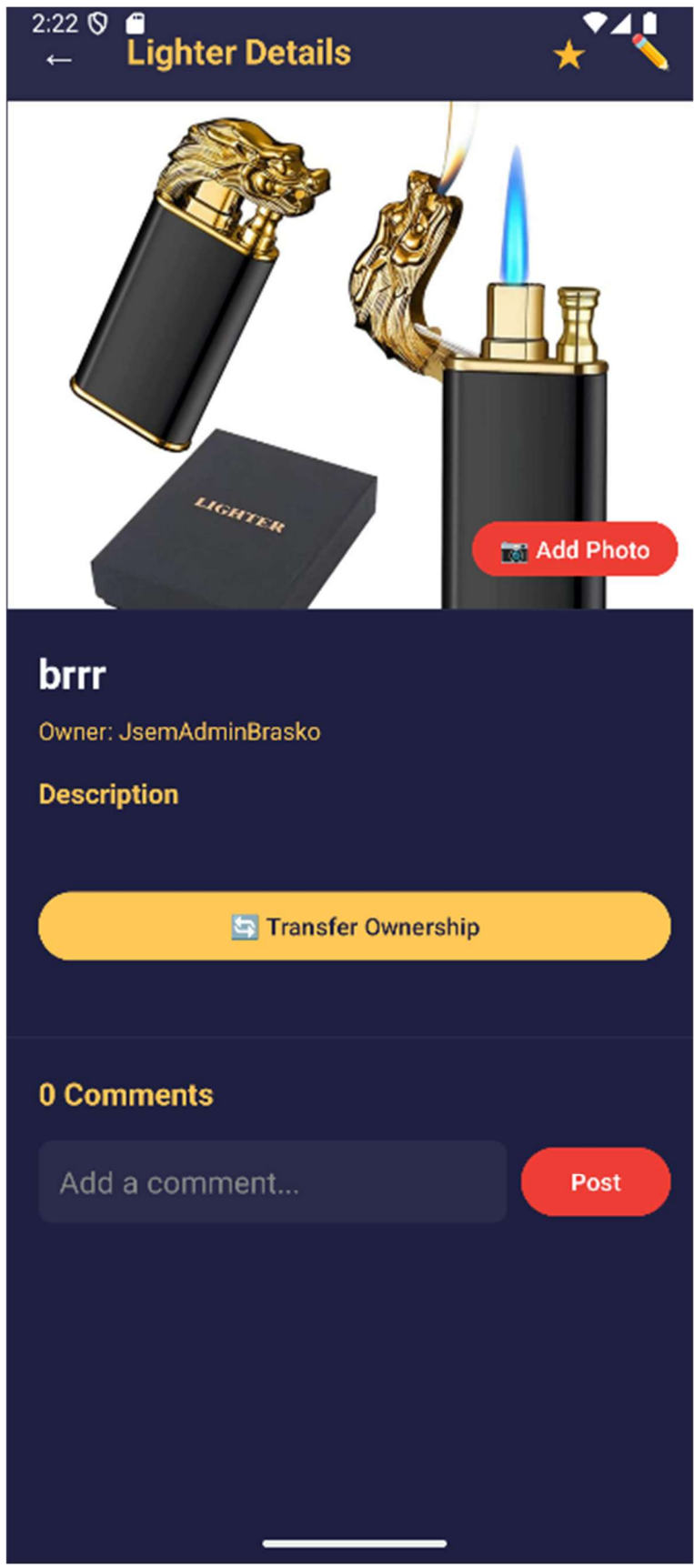
Obrázek 31 Přihlášení



Obrázek 32 Hlavní obrazovka



Obrázek 33 Detail profilu



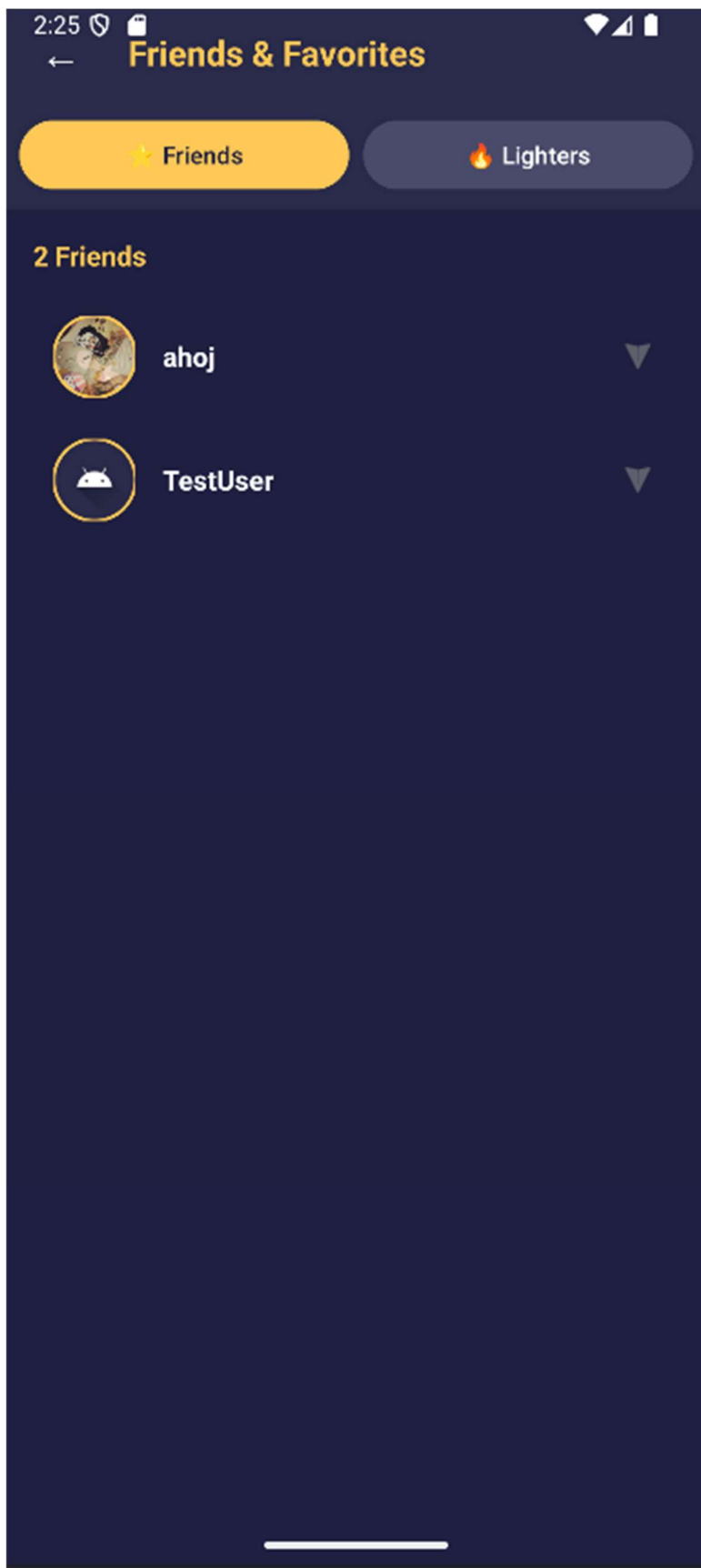
Obrázek 34 Detail zapalovače



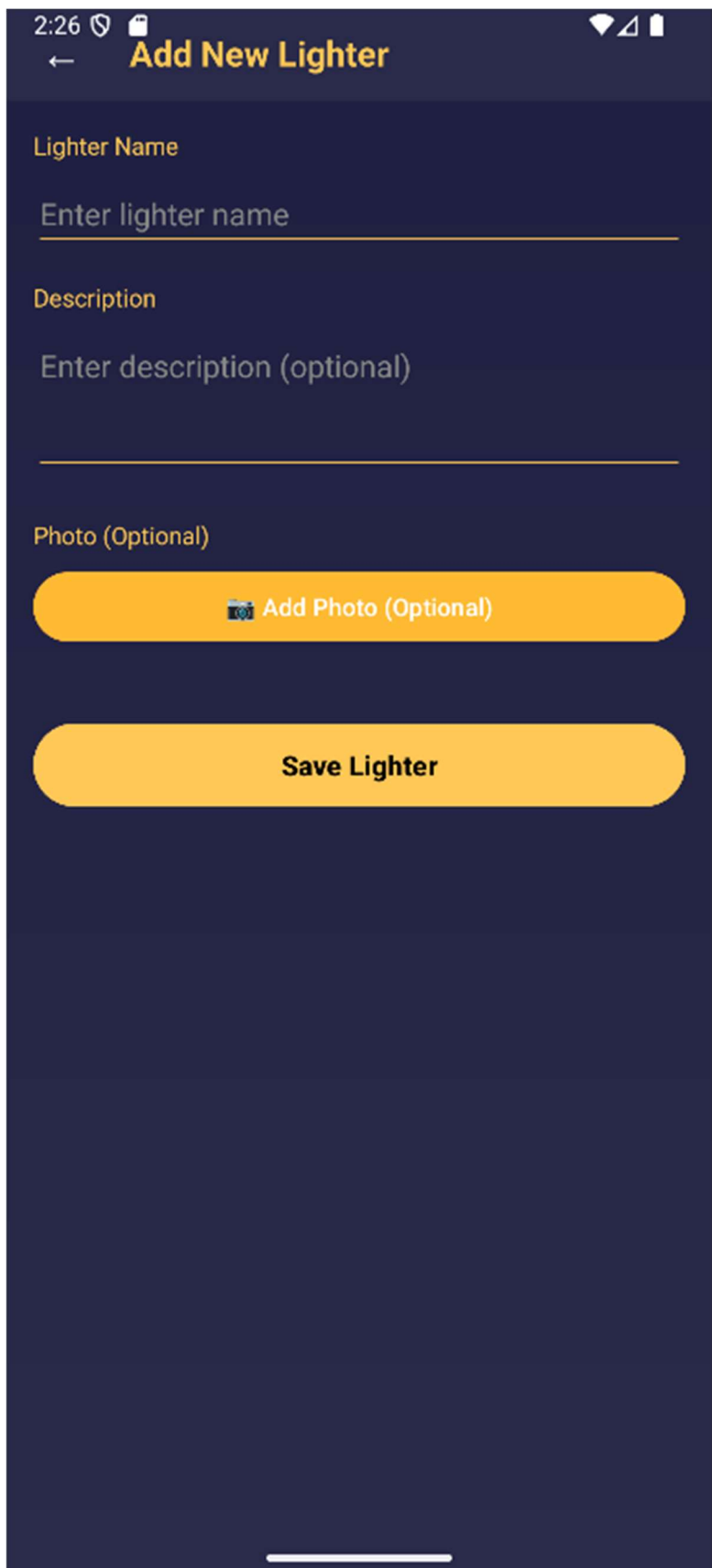
Obrázek 35 Vyhledávání



Obrázek 36 Žebříčky



Obrázek 37 Oblíbené



Obrázek 38 Přidání zapalovače



Obrázek 39 - Galerie zapalovače

## 9.2 Zdrojový kód (vybrané části)

```
boolean canModify = false;
if (currentSession != null) {
    canModify = (lighter.getOwner() == currentSession.getProfileId()) || isAdmin;
}

holder.editButton.setVisibility(canModify ? View.VISIBLE : View.GONE);
holder.deleteButton.setVisibility(canModify ? View.VISIBLE : View.GONE);
```

Toto je část kódu z třídy `LighterAdapter`. Tato specifická část se stará o správné zobrazení tlačítka `editButton`. To zajišťuje zobrazování tlačítka pouze pokud uživatel může provést operaci. Samotná úprava je ošetřena na databázi, takže tato funkce je pouze kosmetická.

```
if (responseCode == 200) {
    AuthResponse authResponse = gson.fromJson(response.toString(), AuthResponse.class);
    currentSession = new AuthSession(
        authResponse.user.id,
        authResponse.user.email,
        authResponse.access_token,
        authResponse.refresh_token );
}
```

Toto je část kódu z přihlašovací metody třídy `SupabaseAuthManager`. Tato specifická část se nachází po odeslání přihlašovacích údajů. Pokud databáze vrátí kód 200, uloží odpověď od databáze s informacemi o přihlášeném uživateli do třídy `AuthSession`.

```
BEGIN

    INSERT INTO public.ownership_history (lighter_id, owner_id, notes)
    VALUES (NEW.id, NEW.owner, 'Created lighter');

    RETURN NEW;
END;
```

Toto je funkce na straně databáze. Je automaticky spouštěna triggerem při vytvoření nového zapalovače. Stará se o zapsání prvního majitele do historie zapalovače s poznámkou "Created lighter".

### 9.3 Seznam použitých knihoven

**Gson 2.10.1** implementation("com.google.code.gson:gson:2.10.1") [4]

Převod mezi JSON odpověďmi ze Supabase a Java objekty jako Lighter nebo UserProfile.

**RecyclerView 1.3.2** implementation("androidx.recyclerview:recyclerview:1.3.2") [6]

Zobrazování seznamů – zapalovače, komentáře, výsledky vyhledávání. Efektivně recykluje řádky a šetří paměť.

**Picasso 2.8** implementation("com.squareup.picasso:picasso:2.8") [5]

Načítání a zobrazování fotografií zapalovačů a profilových fotek ze Supabase Storage.

AndroidX Preference 1.2.1 implementation("androidx.preference:preference:1.2.1") [7]

Ukládání přihlašovacích tokenů mezi spuštěními aplikace, aby se uživatel nemusel přihlašovat při každém otevření.

## 9.4 Odkazy a zdroje

1. *What is Java?* Online. IBM. 2026. Dostupné z: <https://www.ibm.com/think/topics/java>. [cit. 2026-03-01].
2. *Docs*. Online. Supabase. 2026. Dostupné z: <https://supabase.com/docs>. [cit. 2026-03-01].
3. *Android Studio Download*. 2026. Online. Developers Android. Dostupné z: <https://developer.android.com/studio>. [cit. 2026-03-29].
4. *Gson*. Online. Google Github. 2026. Dostupné z: <https://google.github.io/gson/>. [cit. 2026-03-30].
5. *Picasso*. Online. Square Open Source. 2026. Dostupné z: <https://square.github.io/picasso/>. [cit. 2026-03-30].
6. *RecyclerView*. Online. Developers Android. 2026. Dostupné z: <https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView>. [cit. 2026-03-30].
7. *AndroidX Preference*. Online. Developers Android. 2026. Dostupné z: <https://developer.android.com/reference/androidx/preference/package-summary>. [cit. 2026-03-30].
8. *Wildcard Pattern Matching*. Online. GeeksForGeeks. 2025. Dostupné z: <https://www.geeksforgeeks.org/dsa/wildcard-pattern-matching/>. [cit. 2026-03-30].
9. *Row Level Security*. Online. Supabase. 2026. Dostupné z: <https://supabase.com/docs/guides/database/postgres/row-level-security>. [cit. 2026-03-30].
10. *Singleton Design Pattern in Java*. Online. GeeksForGeeks. 2024. Dostupné z: <https://www.geeksforgeeks.org/system-design/singleton-design-pattern-in-java/>. [cit. 2026-03-30].
11. *JSON Web Token (JWT)*. Online. Supabase. 2026. Dostupné z: <https://supabase.com/docs/guides/auth/jwts>. [cit. 2026-03-30].
12. *What is a Base64 Image and How to Encode It?* Online. B64 Encode. 2026. Dostupné z: <https://b64encode.com/blog/what-is-a-base64-image-and-how-to-encode-it/>. [cit. 2026-03-30].

13. *Storage*. Online. Supabase. 2026. Dostupné

z: <https://supabase.com/docs/guides/storage>. [cit. 2026-03-30].

GitHub repositář: <https://github.com/simon-kovac/ZappaTrack>