

## ZÁVĚREČNÁ STUDIJNÍ PRÁCE

### dokumentace

**Mobilní geolokační aplikace pro geocaching  
s podporou offline režimu a herními prvky**



**Autor:** Antonín Bujnovský

**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování

**Třída:** IT4

**Školní rok:** 2025/2026

## **Poděkování**

Rád bych touto cestou vyjádřil upřímné poděkování pánům učitelům Ing. Petru Grussmannovi a Mgr. Marku Lučnému za jejich ochotu, odborné připomínky a čas, který mi věnovali při konzultacích během vývoje aplikace GeoHunt.

## **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 6. ledna 2026

.....  
Podpis autora

## **Abstrakt**

Tato maturitní práce se zabývá návrhem a realizací mobilní geolokační aplikace GeoHunt, která je určena pro příznivce geocachingu. Cílem projektu bylo vytvořit stabilní a výkonnou platformu v prostředí Flutter s využitím cloudového řešení Supabase. Práce detailně popisuje technické výzvy spojené s optimalizací zobrazení velkého množství bodů zájmu na mapě, implementaci herních prvků pro odemykání obsahu a zajištění funkčnosti v offline režimu pomocí lokální SQLite databáze. Výsledkem je plně funkční aplikace, která kombinuje moderní technologie s prvky gamifikace pro zlepšení uživatelského zážitku v terénu.

## **Klíčová slova**

Flutter, Geocaching, Supabase, Dart, Geolokace, Mobilní aplikace, PostgreSQL, Gamifikace, Offline synchronizace.

## **Abstract**

This graduation thesis focuses on the design and implementation of the GeoHunt mobile geolocation application, intended for geocaching enthusiasts. The goal of the project was to develop a stable and high-performance platform using the Flutter framework and Supabase cloud services. The thesis describes in detail the technical challenges associated with optimizing the display of a large number of points of interest on a map, the implementation of gaming elements for unlocking content, and ensuring offline functionality using a local SQLite database. The result is a fully functional application that combines modern technologies with gamification elements to enhance the user experience in the field.

## **Keywords**

Flutter, Geocaching, Supabase, Dart, Geolocation, Mobile application, PostgreSQL, Gamification, Offline synchronization.

# Obsah

<b>Úvod</b>	3
<b>1 Teoretická a metodická východiska</b>	4
1.1 O projektu . . . . .	4
1.2 Framework Flutter a programovací jazyk Dart . . . . .	4
1.3 Architektonický vzor Clean Architecture . . . . .	5
1.4 Clouдовá platforma Supabase a databáze PostgreSQL . . . . .	5
<b>2 Analýza a návrh řešení</b>	6
2.1 Analýza stávajících řešení a konkurence . . . . .	6
2.2 Specifikace funkčních a nefunkčních požadavků . . . . .	6
2.2.1 Funkční požadavky . . . . .	6
2.2.2 Nefunkční požadavky . . . . .	7
2.3 Návrh uživatelského rozhraní (UI) a zkušenosti (UX) . . . . .	7
2.4 Návrh databázového schématu a bezpečnostních pravidel . . . . .	7
2.4.1 Row Level Security (RLS) . . . . .	8
<b>3 Realizace mobilní aplikace</b>	9
3.1 Inicializace projektu a konfigurace prostředí . . . . .	9
3.2 Implementace autentizačního systému . . . . .	9
3.2.1 Logika AuthGuard a správa relací . . . . .	9
3.3 Integrace mapového engine a práce s geografickými daty . . . . .	10
3.4 Vývoj herních mechanik a dovednostní mini-hry . . . . .	10
3.4.1 Implementace mini-hry Tap-Game . . . . .	10
<b>4 Specifické technické výzvy a jejich řešení</b>	11
4.1 Optimalizace vykreslování velkého množství bodů (Clustering) . . . . .	11
4.2 Implementace offline režimu a lokálního úložiště SQLite . . . . .	11
4.3 Logika synchronizace dat mezi klientem a serverem . . . . .	12
4.4 Generování žebříčků pomocí SQL Views . . . . .	12
<b>5 Zhodnocení, testování a nasazení</b>	13
5.1 Průběh testování v reálném terénu . . . . .	13
5.2 Proces sestavení aplikace (Debug vs. Release build) . . . . .	13
5.3 Řešení problémů s podepisovacími klíči a Google API . . . . .	14
5.4 Zhodnocení výsledků a možnosti budoucího rozvoje . . . . .	14
<b>Závěr</b>	16
<b>Seznam použité literatury</b>	17



# ÚVOD

V dnešním digitálním světě se hranice mezi volnočasovými aktivitami a moderními technologiemi neustále stírají. Geocaching, jakožto celosvětově rozšířený fenomén propojující turistiku s vyhledáváním skrytých schránek, představuje ideální příklad této symbiózy. Ačkoliv na trhu existuje řada nástrojů pro tuto aktivitu, mnoho z nich naráží na technická omezení v podobě nízkého výkonu při zobrazení velkého množství dat, nespolehlivosti v oblastech bez stabilního internetového připojení nebo závislosti na fyzických schránkách.

Hlavní motivací pro výběr tohoto tématu byla snaha vytvořit komplexní mobilní řešení, které tyto bariéry odstraňuje a zároveň obohacuje tradiční geocaching o moderní herní prvky. Zvolená kombinace technologií Flutter a Supabase nabízí progresivní přístup k vývoji, který umožňuje velkou míru interaktivnosti, plynulost uživatelského rozhraní a vysokou stabilitu systému. Klíčovou technickou výzvou projektu bylo především efektivní vykreslování desítek tisíc bodů zájmu na mapových podkladech a zajištění plné funkčnosti aplikace i v náročných podmínkách terénu bez přístupu k síti.

Cílem této práce je dokumentace kompletního cyklu vývoje softwarového produktu Geo-Hunt – od teoretického vymezení použitých technologií přes analýzu uživatelských potřeb až po praktickou implementaci klíčových funkcí, jako je geolokace, offline synchronizace a gamifikace. Práce si klade za cíl demonstrovat, že pomocí moderních frameworků lze vytvořit robustní a uživatelsky přívětivou platformu, která motivuje studenty i širší veřejnost k aktivnímu pohybu a objevování okolního světa.

# 1 TEORETICKÁ A METODICKÁ VÝCHODISKA

## 1.1 O PROJEKTU

Výsledkem tohoto maturitního projektu je komplexní mobilní aplikace GeoHunt, která modernizuje tradiční koncept geocachingu. Cílem práce bylo navrhnout a realizovat systém, který propojuje fyzickou aktivitu v přírodě s digitálními herními prvky. GeoHunt umožnuje uživatelům vyhledávat virtuální objekty (kešky) na základě GPS souřadnic v reálném čase.

Klíčovým prvkem, kterým se projekt odlišuje od stávajících řešení, je zavedení mini-her. Po dosažení fyzické lokace kešky se uživateli neotevře obsah automaticky, ale musí nejprve úspěšně absolvovat jednu náhodně vybranou mini-hru. Tento přístup zvyšuje interaktivitu aplikace a motivuje hráče k lepším výkonům. Součástí projektu je také systém žebříčků a úplná podpora offline režimu, která je nezbytná pro použití v místech s omezeným mobilním signálem.

## 1.2 FRAMEWORK FLUTTER A PROGRAMOVACÍ JAZYK DART

Pro vývoj klientské části aplikace jsem zvolil framework Flutter od společnosti Google. Jedná se o moderní UI toolkit, který umožňuje vytvářet nativně kompilované aplikace pro mobilní zařízení, web i desktop z jediné kódové báze. Hlavním argumentem pro volbu tohoto nástroje byla jeho schopnost vykreslovat složitá rozhraní (včetně mapových podkladů s tisíci body) rychlostí až 120 snímků za sekundu díky grafickému engine Impeller a taky moje zkušenost s tímto frameworkm.

Programovací jazyk Dart, na kterém je Flutter postaven, nabízí silnou typovou kontrolu a vynikající podporu pro asynchronní operace, což je pro geolokační aplikaci pracující s externími API kritické. Funkce *Hot Reload* navíc umožnila plynulý vývojový cyklus s okamžitou vizuální odezvou na změny v kódu.

## 1.3 ARCHITEKTONICKÝ VZOR CLEAN ARCHITECTURE

---

Aby byl projekt GeoHunt udržitelný a snadno testovatelný, tak jsem implementoval architekturu *Clean Architecture*. Tato metodika striktně odděluje obchodní logiku od vnějších vlivů, jako jsou databáze nebo uživatelské rozhraní. Projekt je rozdělen do tří nezávislých vrstev:

- **Domain layer:** Jádro aplikace obsahující entity a rozhraní repozitářů. Je nezávislá na Flutetu i backendu.
- **Data layer:** Implementace repozitářů, která zajišťuje komunikaci se Supabase a lokální cache v SQLite.
- **Presentation layer:** Obsahuje uživatelské rozhraní (Widgety) a logiku pro správu stavu (*State Management*).

## 1.4 CLOUDOVÁ PLATFORMA SUPABASE A DATABÁZE POSTGRESQL

Jako hlavní úložiště dat a poskytovatel backendových služeb jsem vybral platformu Supabase. Jedná se o systém typu *Backend-as-a-Service* (BaaS) postavený na relační databázi PostgreSQL. Tento výběr byl strategický z hlediska podpory geografických dat a možnosti využít technologii PostGIS pro vzdálenostní dotazy. Důležitým faktorem pro volbu této platformy byla rovněž má předchozí praktická zkušenosť s tímto prostředím, což umožnilo efektivnější návrh databázové struktury a výrazně urychlilo proces následné implementace backendových služeb.

Bezpečnost je řešena přímo na úrovni databáze pomocí *Row Level Security* (RLS). Každý požadavek je ověřován na straně serveru, což zajišťuje, že uživatel může přistupovat pouze ke svým vlastním logům nebo veřejným keškám. Pro výpočet globálních statistik a žebříčků v reálném čase jsou využívány SQL Views, které agregují data bez nutnosti přetěžování klientské aplikace.

## 2 ANALÝZA A NÁVRH ŘEŠENÍ

V této kapitole je popsán proces analýzy trhu, stanovení konkrétních požadavků na funkcionality aplikace a návrh datové struktury. Tato fáze byla nezbytná pro pochopení cílové skupiny uživatelů a eliminaci potenciálních technických rizik.

### 2.1 ANALÝZA STÁVAJÍCÍCH ŘEŠENÍ A KONKURENCE

Při návrhu aplikace GeoHunt bylo nutné prozkoumat existující geolokační platformy. Trhu dominuje oficiální aplikace *Geocaching®*, která však pro neplatící uživatele omezuje řadu funkcí. Dalšími konkurenty jsou české projekty *GEOFUN* nebo *Adventure Lab®*.

Hlavním zjištěním analýzy bylo, že většina aplikací se soustředí buď na fyzické schránky (tradiční geocaching), nebo na statické textové úkoly. GeoHunt tuto mezeru vyplňuje zavedením virtuálních kešek na reálných souřadnicích, jejichž odemčení je podmíněno úspěšným splněním dovednostní mini-hry. Tímto způsobem se aplikace stává dynamičtější a atraktivnější pro mladší generaci hráčů.

### 2.2 SPECIFIKACE FUNKČNÍCH A NEFUNKČNÍCH POŽADAVKŮ

Na základě analýzy byly stanoveny priority vývoje rozdělené do dvou kategorií.

#### 2.2.1 Funkční požadavky

- **Autentizace:** Registrace a přihlášení pomocí emailu nebo Google účtu.
- **Mapové rozhraní:** Zobrazení aktuální polohy uživatele a kešek v blízkém okolí.
- **Herní mechanika:** Detekce blízkosti kešky (do 20 metrů) a spuštění mini-hry.
- **Statistiky:** Real-time žebříček nejlepších hráčů na základě počtu nálezů.
- **Offline mód:** Možnost stažení dat pro využití v lesním terénu bez signálu.

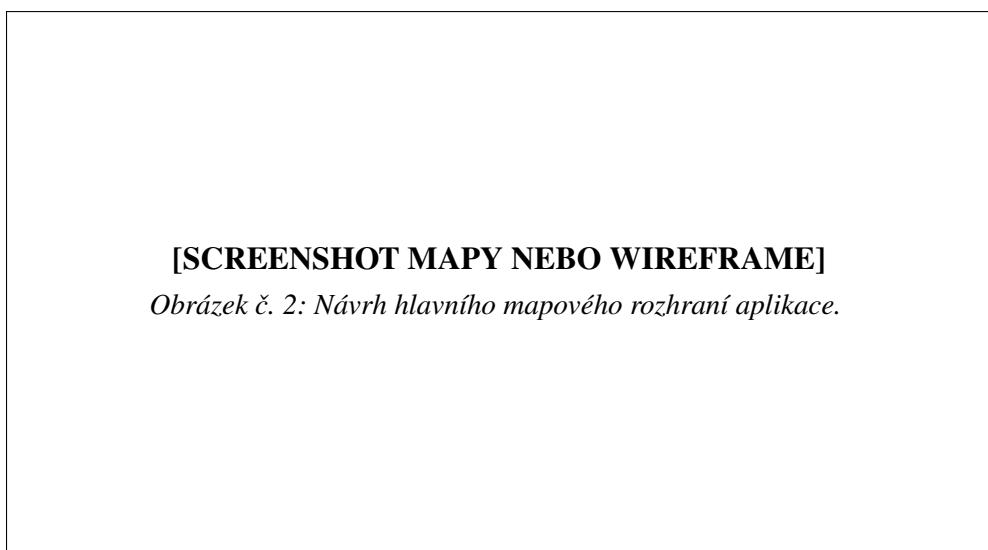
---

## 2.2.2 Nefunkční požadavky

- **Vysoký výkon:** Plynulý rendering mapy i při velkém počtu bodů (clustering).
- **Bezpečnost:** Ochrana databáze proti neoprávněným zápisům (RLS).
- **Multiplatformita:** Funkčnost na systémech Android i iOS.

## 2.3 NÁVRH UŽIVATELSKÉHO ROZHRANÍ (UI) A ZKUŠENOSTI (UX)

Cílem návrhu UI bylo vytvořit čisté a intuitivní prostředí, které nebude uživatele v terénu rozptylovat. Dominantním prvkem je interaktivní mapa využívající podklady OpenStreetMap. Pro interakci s keškami byla navržena vizuální signalizace – kešky mění barvu na základě stavu (nalezeno/nenalezeno).



[SCREENSHOT MAPY NEBO WIREFRAME]

Obrázek č. 2: Návrh hlavního mapového rozhraní aplikace.

Obrázek 2.1: Vizuální návrh mapové části.

## 2.4 NÁVRH DATABÁZOVÉHO SCHÉMATU A BEZPEČNOSTNÍCH PRAVIDEL

Jádrem systému je relační model v PostgreSQL. Schéma se skládá ze tří hlavních entit: *profiles* (data o uživatelích), *geocaches* (souřadnice a popisy bodů) a *logs* (záznamy o nálezech). Vztahy mezi tabulkami jsou řešeny pomocí cizích klíčů, což zajišťuje referenční integritu.

---

### **2.4.1 Row Level Security (RLS)**

Bezpečnostní politika byla nastavena tak, aby zápis do tabulky *logs* mohl provést pouze autentizovaný uživatel pro své vlastní ID. Veřejné kešky jsou přístupné pro čtení všem, ale jejich editace je omezena pouze na uživatele s rolí *admin*. Tato vrstva ochrany přímo v databázi nahrazuje složitou logiku na straně serveru.

## 3 REALIZACE MOBILNÍ APLIKACE

V této kapitole je popsán technický postup vývoje aplikace GeoHunt. Pozornost je věnována konfiguraci vývojového prostředí, implementaci klíčových modulů a propojení klientské části s cloudovým backendem.

### 3.1 INICIALIZACE PROJEKTU A KONFIGURACE PROSTŘEDÍ

Vývoj byl zahájen vytvořením nového projektu ve frameworku Flutter. Pro zajištění bezpečnosti citlivých údajů, jako jsou API klíče k Supabase nebo Google Maps (které nakonec nebyly použity), byl do projektu integrován balíček *flutter\_dotenv*. Tyto údaje jsou uloženy v souboru *.env*, který není součástí veřejného repozitáře, čímž je zamezeno jejich zneužití.

Struktura projektu byla od počátku organizována podle pravidel *Clean Architecture*. Složka *lib/* byla rozdělena na podadresáře *core* (sdílené prvky) a *features*, kde každá funkce aplikace (např. mapa, autentizace, žebříčky) má svou vlastní logiku oddělenou od ostatních.

### 3.2 IMPLEMENTACE AUTENTIZAČNÍHO SYSTÉMU

Pro přihlašování uživatelů byla využita služba *Supabase Auth*. Systém podporuje dvě metody: klasickou registraci pomocí emailu a hesla a nativní *Google OAuth2* přihlášení.

#### 3.2.1 Logika AuthGuard a správa relací

Klíčovou součástí realizace bylo vytvoření widgetu *AuthGuard*. Ten využívá *StreamBuilder* k poslechu stavu autentizace v reálném čase. Pokud uživatel není přihlášen, aplikace automaticky zobrazí přihlašovací obrazovku. Po úspěšném ověření identity je uživatel přesměrován na hlavní mapové rozhraní. Tato logika zajišťuje, že nepovolané osoby nemají přístup k hernímu obsahu.

### 3.3 INTEGRACE MAPOVÉHO ENGINE A PRÁCE S GEOGRA-

---

## FICKÝMI DATY

Vzhledem k požadavku na využití otevřených standardů byl zvolen balíček *flutter\_map* ve spojení s OpenStreetMap. Na rozdíl od Google Maps SDK toto řešení umožňuje plnou kontrolu nad vizuálním stylem mapy bez nutnosti platit licenční poplatky při vysokém počtu zobrazení.

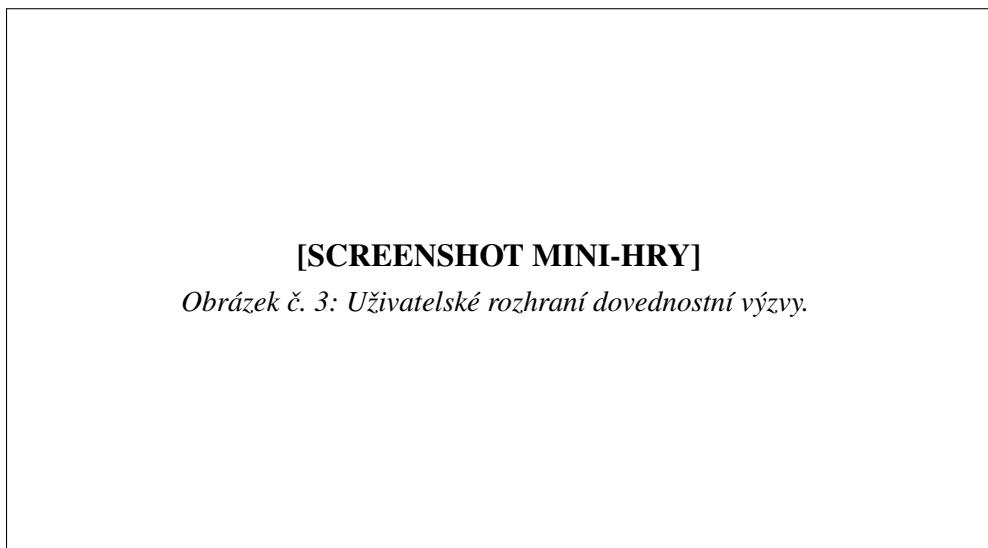
Pro určování polohy uživatele v reálném čase je využívána knihovna *geolocator*, která přistupuje k GPS modulu zařízení s vysokou přesností. Souřadnice jsou v aplikaci reprezentovány objekty typu *LatLang*, se kterými pracují všechny výpočetní moduly.

## 3.4 VÝVOJ HERNÍCH MECHANIK A DOVEDNOSTNÍ MINI-HRY

Herní smyčka začíná v modulu pro detekci vzdálenosti. Aplikace periodicky vypočítává vzdálenost mezi aktuální polohou hráče a nejbližší keškou pomocí algoritmu *Haversine*. Pokud vzdálenost klesne pod hranici 20 metrů, aktivuje se tlačítko pro odemčení.

### 3.4.1 Implementace mini-hry Tap-Game

Dovednostní výzva byla navržena jako interaktivní prvek vyžadující postřeh. Hráč musí v časovém limitu naplnit ukazatel postupu rychlým klepáním na displej, přičemž obtížnost (rychlosť ubývání postupu) je definována atributem v databázi. Po úspěšném dokončení hry je odeslán požadavek na Supabase pro zápis nálezu do tabulky *logs*.



[SCREENSHOT MINI-HRY]

Obrázek č. 3: Uživatelské rozhraní dovednostní výzvy.

Obrázek 3.1: Ukázka herní mechaniky pro odemykání kešky.

## **4 SPECIFICKÉ TECHNICKÉ VÝZVY A JEJICH ŘEŠENÍ**

Tato kapitola se zaměřuje na nejsložitější technické problémy, které bylo nutné během vývoje vyřešit pro zajištění produkční kvality aplikace GeoHunt.

### **4.1 OPTIMALIZACE VYKRESLOVÁNÍ VELKÉHO MNOŽSTVÍ BODŮ (CLUSTERING)**

Při testování aplikace s reálnými daty, která obsahuje desítky tisíc záznamů o keškách, došlo k výraznému poklesu snímkové frekvence (FPS). Standardní vykreslování každého bodu jako samostatného markeru "zabilo" UI vlákno aplikace.

Řešením bylo nasazení balíčku *flutter\_map\_marker\_cluster*. Tato technologie shlukuje body do skupin (clusterů) na základě aktuální úrovně přiblížení. K rozbalení clusteru na jednotlivé body dochází až při dostatečném zoomu. Tím se počet aktivních prvků na displeji snížil z tisíců na desítky, což zajistilo plynulost mapy i na starších zařízeních.

### **4.2 IMPLEMENTACE OFFLINE REŽIMU A LOKÁLNÍHO ÚLOŽIŠTĚ SQLITE**

Geocaching často probíhá v místech se špatným signálem. Pro zajištění funkčnosti byla implementována vrstva *OfflineCacheService* využívající databázi *sqlite*. Uživatel má možnost stáhnout data o keškách do lokální paměti zařízení. Aplikace v reálném čase detekuje stav připojení a v případě výpadku signálu automaticky přepne zdroje dat (*Repository*) ze Supabase na lokální SQLite.

### **4.3 LOGIKA SYNCHRONIZACE DAT MEZI KLIENTEM A SER-**

---

## VEREM

Zápis nálezů v offline režimu představoval výzvu pro integritu dat. Byl vytvořen systém dočasných logů. Pokud hráč odemkne kešku offline, záznam se uloží do lokální tabulky *offline\_logs*. Aplikace na pozadí sleduje obnovení konektivity (*ConnectivityResult.wifi* nebo *mobile*) a následně spustí proces synchronizace, který odešle všechny nashromážděné záznamy do cloudu Supabase.

## 4.4 GENEROVÁNÍ ŽEBŘÍČKŮ POMOCÍ SQL VIEWS

Pro výpočet herních statistik (denní, měsíční a celkové pořadí) by bylo neefektivní stahovat všechny logy a třídit je v mobilním telefonu. Místo toho byla logika přenesena na stranu databáze PostgreSQL pomocí *SQL Views*. Tyto pohledy automaticky agregují data z tabulek *profiles* a *logs*. Flutter aplikace k tému datům přistupuje jako k virtuálním tabulkám, což minimalizuje množství přenášených dat a výrazně urychluje zobrazení žebříčku.



[SCREENSHOT ŽEBŘÍČKU]

Obrázek č. 4: Zobrazení herního pořadí čerpajícího z *SQL Views*.

Obrázek 4.1: Uživatelské rozhraní žebříčku nejlepších hráčů.

## 5 ZHODNOCENÍ, TESTOVÁNÍ A NASAZENÍ

Závěrečná fáze projektu se soustředila na ověření funkčnosti aplikace v reálných podmírkách a přípravu pro veřejné publikování. Tato kapitola popisuje proces testování geolokačních funkcí a technické překážky spojené s autentizací v produkčním prostředí.

### 5.1 PRŮBĚH TESTOVÁNÍ V REÁLNÉM TERÉNU

Vzhledem k povaze aplikace GeoHunt nebylo možné provádět testy pouze v emulátoru. Probehlo několik testovacích cyklů v městské i lesní zástavbě s cílem ověřit přesnost GPS modulu a stabilitu herní smyčky.

Klíčovým bodem byla validace algoritmu pro detekci blízkosti. Testy potvrdily, že hranice 20 metrů je ideálním kompromisem mezi herní náročností a přesností civilního GPS signálu, který může v husté zástavbě vykazovat odchylky. Rovněž byla ověřena funkčnost offline režimu, kdy aplikace po ztrátě mobilních dat plynule přešla na lokální databázi SQLite, umožnila odemčení kešky a zapsala nález do synchronizační fronty.

### 5.2 PROCES SESTAVENÍ APLIKACE (DEBUG VS. RELEASE BUILD)

Během vývoje byly využívány dvě odlišné verze aplikace. Verze *Debug* sloužila k rychlému ladění chyb pomocí nástroje *Hot Reload*. Pro finální testování výkonu byl však nutný *Release build*.

Tento proces zahrnuje komplikaci do nativního kódu pomocí příkazu *flutter build apk*. V této fázi byl aplikován nástroj *R8*, který provádí optimalizaci kódu (*shrinking*) a odstraňuje nepoužité části knihoven. Tím došlo k výraznému snížení velikosti výsledného instalačního souboru a zvýšení plynulosti animací v mini-hře.

### 5.3 ŘEŠENÍ PROBLÉMŮ S PODEPISOVACÍMI KLÍČI A GOO-

---

## GLE API

Největší technickou výzvou při nasazení byla konfigurace služby *Google Sign-In*. Zatímco ve vývojovém prostředí autentizace fungovala bezchybně, v produkční verzi (po podepsání aplikace) docházelo k chybě *ApiException: 10*.

Příčinou byl nesoulad mezi otisky certifikátů SHA-1. Pro vyřešení problému bylo nutné:

1. Pomocí nástroje *keytool* extrahovat SHA-1 otisk z produkčního souboru *keystore*.
2. Tento otisk zaregistrovat v konzoli Google Cloud i v nastavení projektu Supabase.
3. Sjednotit *Package Name* (cz.geohunt.app) napříč všemi platformami.

Toto řešení zajistilo plnou funkčnost OAuth2 autentizace i v distribuované verzi aplikace.

## 5.4 ZHODNOCENÍ VÝSLEDKŮ A MOŽNOSTI BUDOUCÍHO ROZVOJE

Projekt GeoHunt úspěšně splnil všechny stanovené cíle. Byla vytvořena stabilní geolokační platforma s přijemným uživatelským rozhraním a inovativní herní mechanikou.

V rámci budoucího rozvoje se nabízí několik směrů:

- **Sociální prvky:** Implementace systému přáatel a sdílení nálezů na sociální síti.
- **Rozšířená realita (AR):** Nahrazení 2D mini-hry zobrazením virtuální kešky přímo v prostředí skrze kameru telefonu.

---

**[FOTKA Z TESTOVÁNÍ V TERÉNU]**

*Obrázek č. 5: Aplikace GeoHunt spuštěná na reálném zařízení v terénu.*

Obrázek 5.1: Testování geolokačních funkcí v praxi.

# ZÁVĚR

Cílem tohoto maturitního projektu bylo navrhnout a realizovat moderní geolokační aplikaci GeoHunt, která by propojila tradiční geocaching s dynamickými herními prvky. Po dokončení vývojového cyklu lze konstatovat, že všechny stanovené cíle byly úspěšně splněny.

V průběhu realizace byla vytvořena robustní mobilní aplikace ve frameworku Flutter, která efektivně využívá cloudové služby Supabase pro správu uživatelů a dat. Za největší technický přínos práce považuji vyřešení optimalizace zobrazení velkého množství geografických bodů pomocí technologie clusteringu a implementaci plně funkčního offline režimu. Tyto aspekty zajišťují vysokou uživatelskou hodnotu aplikace i v náročných podmírkách terénu bez přístupu k síti.

Během vývoje jsem se musel vypořádat s řadou netriviálních problémů, zejména v oblasti bezpečnosti a produkčního nasazení. Řešení nesouladu certifikátů SHA-1 pro autentizaci Google a správné nastavení Row Level Security v databázi PostgreSQL mi přineslo cenné zkušenosti s architekturou moderních informačních systémů.

Projekt GeoHunt je v tuto chvíli ve stavu plně funkčního MVP (Minimum Viable Product). V budoucnu by bylo možné aplikaci dále rozšiřovat o prvky rozšířené reality nebo sociální funkce pro interakci mezi hráči. Práce na tomto projektu mi umožnila prohloubit znalosti v oblasti mobilního vývoje, práce s prostorovými daty a správy cloudové infrastruktury.

## **SEZNAM POUŽITÉ LITERATURY**

- [1] GOOGLE. *Flutter documentation* [online]. 2025 [cit. 2026-01-05]. Dostupné z: <https://docs.flutter.dev/>
- [2] SUPABASE. *Supabase Docs: The Open Source Firebase Alternative* [online]. 2025 [cit. 2026-01-05]. Dostupné z: <https://supabase.com/docs>
- [3] POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL 16 Documentation* [online]. 2025 [cit. 2026-01-05]. Dostupné z: <https://www.postgresql.org/docs/>
- [4] DART. *Dart language guide* [online]. 2025 [cit. 2026-01-05]. Dostupné z: <https://dart.dev/guides>
- [5] OPENSTREETMAP FOUNDATION. *OpenStreetMap Wiki* [online]. 2025 [cit. 2026-01-05]. Dostupné z: <https://wiki.openstreetmap.org/>
- [6] MARTIN, Robert C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. First edition. Boston: Prentice Hall, 2017. ISBN 978-0134494166.
- [7] PUB.DEV. *Flutter Map Marker Cluster package* [online]. 2025 [cit. 2026-01-05]. Dostupné z: [https://pub.dev/packages/flutter\\_map\\_marker\\_cluster](https://pub.dev/packages/flutter_map_marker_cluster)

## **SEZNAM PŘÍLOH**

- **Příloha č. 1:** Zdrojový kód aplikace (přiložen na digitálním nosiči / odkaz na GitHub).
- **Příloha č. 2:** Obrázková dokumentace uživatelského rozhraní (screenshoty z aplikace).
- **Příloha č. 3:** SQL skript pro inicializaci databázového schématu v Supabase.
- **Příloha č. 4:** Uživatelský manuál pro instalaci a první spuštění aplikace.