

# GYMNASIUM JANA KEPLERA

Parléřova 2/118, 169 00 Praha 6



## Random Voyager

Maturitní práce

Autor: Václav Tichý

Třída: 4.B

Školní rok: 2023/2024

Předmět: Informatika

Vedoucí práce: Ing. Šimon Schierreich

Praha, 2024





**GYMNASIUM JANA KEPLERA**  
*Kabinet informatiky*

## **ZADÁNÍ MATURITNÍ PRÁCE**

*Student:* Václav Tichý  
*Třída:* 4.B  
*Školní rok:* 2023/2024  
*Platnost zadání:* 30. 9. 2024  
*Vedoucí práce:* Šimon Schierreich  
  
*Název práce:* Random Voyager

*Pokyny pro vypracování:*

Student v rámci maturitní aplikace vypracuje mobilní aplikaci pro systémy Android a iOS. Tato aplikace bude uživateli zobrazovat obrázek nějakého místa v jeho blízkosti a úkolem uživatele bude toto místo najít. Po nalezení místa je uživateli připsán nález do jeho statistiky. Aplikace bude podporovat několik režimů, například zobrazení náhodného místa, týdenní úkoly, nebo významná turistická místa.

*Doporučená literatura:*

- [1] EVANS, Eric. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston: Addison Wesley Longman Publishing, 2003. ISBN 978-0-321-12521-7.
- [2] MARTIN, Robert C. *Design Principles and Design Patterns*. [www.objectmentor.com](http://www.objectmentor.com), 2000. Dostupné z: [https://fi.ort.edu.uy/innovaportal/file/2032/1/design\\_principles.pdf](https://fi.ort.edu.uy/innovaportal/file/2032/1/design_principles.pdf).
- [3] FOWLER, Martin. *Patterns of enterprise application architecture*. Boston: Addison-Wesley Professional, 2003. ISBN 978-0-321-12742-6.

*URL repozitáře:*

[https://github.com/ticvac/random\\_voyager](https://github.com/ticvac/random_voyager)

---

*student*

---

*vedoucí práce*

*V Praze dne 20. 10. 2023*



## **Prohlášení**

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Nemám žádné námitky proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne 3. března 2024

Václav Tichý



## **Poděkování**

Svému kamarádovi Honzovi za pomoc při vymýšlení tématu projektu a všem kteří mi pomáhali při testování. A sestře za výbornou blondýnu.





## **Abstrakt**

Mobilní aplikace Random Voyager slouží k prozkoumávání měst bez mapy. Cílem projektu bylo vytvořit aplikaci pro platformy Android a iOS, která uživatelům zobrazí místo z jejich okolí pomocí Google Street View. Když uživatelé na místo dojdou, mohou si ho označit jako dosažené. Dále uživatelé mohou publikovat zajímavá místa, která objeví, a sdílet je s ostatními uživateli. Pro implementaci aplikace byl využit framework Flutter, který umožňuje kompilaci aplikace pro obě platformy z jednoho zdrojového kódu.

## **Klíčová slova**

mobilní aplikace, Google Street View, procházky

## **Abstract**

Mobile app Random Voyager is for exploring cities without a map. The goal of the project was to create an application for the Android and iOS platforms that will show users a place from their surroundings using Google Street View. When users reach a location, they can mark it as found. Furthermore, users can publish interesting places they discover and share them with other users. The Flutter framework was used to implement the application, which supports the compilation for both platforms from one source code.

## **Keywords**

mobile application, Google Street View, short walks



# Obsah

<b>1</b>	<b>Teoretická část</b>	<b>3</b>
1.1	Cíl projektu . . . . .	3
1.2	Jak aplikace funguje . . . . .	3
1.3	Využití aplikace . . . . .	3
<b>2</b>	<b>Implementace</b>	<b>5</b>
2.1	Flutter . . . . .	5
2.2	FlutterGoogleStreetView . . . . .	5
2.3	Geolocator . . . . .	5
2.4	Databáze . . . . .	6
2.4.1	SQLite . . . . .	6
2.4.2	SharedPreferences . . . . .	6
2.4.3	Firebase . . . . .	6
2.5	Pomocné knihovny . . . . .	6
2.5.1	CupertinoIcons . . . . .	6
2.5.2	FlutterNativeSplash . . . . .	7
2.5.3	FlutterLauncherIcons . . . . .	7
2.6	Vzdálenost bodů na kouli . . . . .	7
2.7	Admin mode . . . . .	7
2.8	arch -x86_64 pod install . . . . .	7
2.9	FlutterGoogleStreetView - bug . . . . .	8
2.10	Pozor na vesnice . . . . .	8
<b>3</b>	<b>Technická dokumentace</b>	<b>9</b>
3.1	Git repositář . . . . .	9
3.2	Programátorská dokumentace . . . . .	9
3.2.1	Jak nainstaloval potřebné nástroje . . . . .	9
3.2.2	Údržba aplikace . . . . .	10
3.2.3	Struktura aplikace . . . . .	10
3.3	Aplikace volně ke stažení . . . . .	11
3.4	Uživatelská dokumentace . . . . .	11
	<b>Závěr</b>	<b>15</b>



# 1. Teoretická část

## 1.1 Cíl projektu

Cílem mého maturitního projektu z informatiky bylo vytvořit mobilní aplikace pro android a iOS, která uživatelům umožní zobrazit si místo v jejich okolí jako Google Street View a následně zaznamenat, když na dané místo došli. Aplikace tak uživatelům umožní prozkoumávat město bez mapy.

## 1.2 Jak aplikace funguje

Před několika lety vyšla hra GeoGuessr<sup>1</sup>, ve které hráč dostane obraz z Google Street View a musí uhodnout kde na světě se místo nachází. Já jsem se touto hrou inspiroval a vytvořil podobnou aplikaci s tím rozdílem, že místo, které se vám zobrazí, je ve vašem okolí, a vy na něj musíte dojít ve skutečném světě.

Moje aplikace vykazuje jistou podobnost s geocachingem<sup>2</sup>, ale má tu výhodu, že se dá hrát v městech po celém světě a nevyžaduje údržbu míst (v tomto případě kešek).

Podrobněji se fungováním aplikace zabývá uživatelská dokumentace.

## 1.3 Využití aplikace

Díky tomu, že vidíme naši vzdálenost od hledaného místa, můžeme aplikaci Random Voyager používat i v městech, ve kterých jsme nikdy nebyli. Aplikaci lze ale využít i v místech, která dobře známe, a to na procvičení znalosti našeho okolí nebo pokud se chceme vydat jinými, než často vychozenými cestami. Můžeme se také podívat na místa, která ostatní uživatelé označili jako zajímavá, a tím obejít časté výhledy a oblíbené památky.

---

<sup>1</sup><https://www.geoguessr.com>

<sup>2</sup><https://www.geocaching.com/play>



## 2. Implementace

Kořen repositáře maturitaVT\_repo je dále zmiňován pouze jako /.

### 2.1 Flutter

Pokud chceme vytvořit aplikaci pro mobilní zařízení, musíme podporovat iOS i android. V takovou chvíli máme před sebou dvě možnosti. Udržovat dvě oddělené code base pro každý operační systém, nebo zvolit řešení, které je kompilovatelné pro obě zařízení.

I když první varianta poskytuje větší kontrolu nad konkrétními prvky systému, zvolil jsem druhou variantu, jelikož je praktičtější. Flutter<sup>1</sup> je framework vytvořený společností Google. Jeho největší výhodou je, že je kompilovatelný pro iOS, android a díky své javascriptové knihovně i pro web.

Největší konkurencí pro Flutter je v současné době React Native<sup>2</sup>, který je také kompilovatelný pro oba operační systémy a též nabízí hot reload (možnost kompilace za běhu aplikace při testování).

Jelikož mají také oba frameworky velkou aktivní komunitu pracující na knihovnách třetích stran, je volba mezi nimi prakticky jenom otázkou toho, v čem se člověku chce více programovat. Já jsem si vybral Flutter, jelikož nemám moc rád javascriptu podobné prvky, která React Native obsahuje.

### 2.2 FlutterGoogleStreetView

Google maps poskytují svůj API pro stahování street views. Pro Flutter není příliš vhodná, ale naštěstí existuje knihovna FlutterGoogleStreetView<sup>3</sup>, se kterou se street views stahují do aplikace docela jednoduše.

Pro běžného uživatele jsou Google Street Views zadarmo, ale Google nějak vydělávat musí, proto je API zpoplatněna. Naštěstí však nabízejí tříměsíční zkušební dobu zadarmo. Aby aplikace fungovala, je potřeba jednou za čtvrtletí vytvořit nový API klíč. Služba není nijak drahá, ale když aplikace není určena ke generování zisku, nemohu si dovolit za její údržbu platit.

### 2.3 Geolocator

Geolocator<sup>4</sup> nám umožňuje rychle získat aktuální pozici uživatele a vrátí nám ji ve formátu lat lon. Implementováno v /lib/support/geolocator.dart

---

<sup>1</sup><https://flutter.dev>

<sup>2</sup><https://reactnative.dev>

<sup>3</sup>[https://pub.dev/packages/flutter\\_google\\_street\\_view](https://pub.dev/packages/flutter_google_street_view)

<sup>4</sup><https://pub.dev/packages/geolocator>

## 2.4 Databáze

V aplikaci Random Voyager používám tři databáze. Každá se hodí na něco jiného a taky jsem si chtěl vyzkoušet implementaci jednotlivých knihoven. Zpětně bych řekl, že funkce databáze SharedPreferences by se daly implementovat pomocí SQLite. Není k tomu ale z hlediska výkonu a UX důvod.

### 2.4.1 SQLite

Jednoduchá knihovna pro vytvoření SQLite<sup>5</sup> databáze na mobilním zařízení. V aplikaci je používána na uložení jednotlivých míst, na která se uživatel rozhodne jít.

Implementováno v `/lib/support/database.dart`

### 2.4.2 SharedPreferences

SharedPreferences<sup>6</sup> poskytují jednoduchou databázi na key-value pairs struktuře. Využívám ji na uložení informace, jestli se veřejné místo uživateli líbilo, nebo ne. Součet hlasů je uložen na cloudu, ale každý uživatel má jen jeden hlas, takže je potřeba si ho pamatovat pro případ změny.

Implementováno v `/lib/support/shares_prefs_database.dart`

### 2.4.3 Firebase

Firebase<sup>7</sup> je jistě nejsložitější použitá databáze, stále se ale jedná o jedno z nejpříjemnějších cloudových řešení. Firebase nám umožňuje ukládat veřejná místa, aby si je pak ostatní uživatelé mohli zobrazit.

Implementováno v `/lib/support/firebase_database.dart`

## 2.5 Pomocné knihovny

Následující knihovny příliš neovlivňují UX, ani nepřidávají složité funkce. Jsou praktickým řešením problémů pro oba operační systémy.

### 2.5.1 CupertinoIcons

CupertinoIcons<sup>8</sup> je knihovna ikon, která poskytuje jednoduchý přístup ke grafikám a je zdarma i pro komerční účely.

---

<sup>5</sup><https://pub.dev/packages/sqlite>

<sup>6</sup>[https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences)

<sup>7</sup><https://firebase.google.com>

<sup>8</sup>[https://pub.dev/packages/cupertino\\_icons](https://pub.dev/packages/cupertino_icons)



### 2.5.2 FlutterNativeSplash

Když se aplikace poprvé načítá, potřebujeme zobrazit nějakou grafiku, aby uživatelé nekoukali pouze na prázdnou obrazovku. Flutter native splash<sup>9</sup> poskytuje řešení funkční pro iOS a android.

### 2.5.3 FlutterLauncherIcons

FlutterLauncherIcons<sup>10</sup> poskytuje bezbolestnou cestu k nastavení ikony aplikace, která se zobrazí uživateli na ploše.

## 2.6 Vzdálenost bodů na kouli

Při ověřování, zda je uživatel do 50m od hledaného místa, nebo při zobrazování místa z uživatelova okolí, potřebujeme spočítat vzdálenost dvou bodů na kouli (země je koule a to nemůžeme zanedbat, jelikož používání čistě karteziánské plochy by vytvořilo nepřesné výsledky, buď na rovníku, nebo naopak na pólech). Jelikož nám geolocator vrací souřadnice ve formátu lat lon, vypočítání vzdálenosti není triviální.

V aplikaci jsem použil vzorec popsany zde<sup>11</sup>.

Implementováno v `/lib/support/geolocator.dart`, funkce `calculateDistance(args)`

## 2.7 Admin mode

V `/lib/constants.dart` se nachází konstanta `adminState`, která nám umožňuje z jedné aplikace mimo jiné schvalovat místa, která uživatelé označili jako zajímavá, ale ještě nejsou veřejně k vidění. Pozor, pokud spouštíte pro produkci, je potřeba tuto konstantu nastavit na `false`.

Myslím si že v praxi by bylo bezpečnější mít samostatnou aplikaci nebo nějaké jiné rozhraní na schvalování míst. V mém projektu to ale problém není.

## 2.8 arch -x86\_64 pod install

Cocoa pods<sup>12</sup> je dependency manager, který používá verze pro iOS. Když ale Apple před 4 lety přešel na nové Apple silicon čipy s architekturou ARM, vznikla spousta problémů. Jedním z nich je skutečnost, že některé knihovny (u mě konkrétně Firebase) nebudou na macOS zkompileovatelné a příkaz `pod install` vyhodí chybu bez jakékoliv užitečné nápovědy co se děje. Zabralo mi to pár dní, ale nakonec jsem narazil na příkaz `arch -x86_64 pod install`, který zmíněnou chybu obejde tím, že `pod install` spustí simulovaný na staré architektuře.

<sup>9</sup>[https://pub.dev/packages/flutter\\_native\\_splash](https://pub.dev/packages/flutter_native_splash)

<sup>10</sup>[https://pub.dev/packages/flutter\\_launcher\\_icons](https://pub.dev/packages/flutter_launcher_icons)

<sup>11</sup>[https://en.wikipedia.org/wiki/Great-circle\\_distance](https://en.wikipedia.org/wiki/Great-circle_distance)

<sup>12</sup><https://cocoapods.org>

Je to jedna z mnoha věcí, které vývoj technologií přináší, naštěstí je zde snaha pro zpětnou kompatibilitu. Ale tento problém postihuje všechny Apple počítače z řady procesorů MX.

### 2.9 FlutterGoogleStreetView - bug

FlutterGoogleStreetView občas spadne :/

Zjistil jsem, že jde o interní problém knihovny, a ani přes velkou snahu se mi ho nepodařilo opravit. Našel jsem ale spoustu lidí se stejným problémem. Chyba nastává při zavírání Street View, kdy knihovna špatně přeruší spojení s google maps API a aplikace crushne.

Chyba se mi v projektu projevovala jen v simulátoru, ale při testování jsem zjistil, že postihuje i méně výkonná zařízení android. Na iOS se projevuje jen zřídka. Bug se mi nepodařilo opravit, ale alespoň jsem aplikaci upravil, aby nikdy nespadla s neuloženými daty.

### 2.10 Pozor na vesnice

Bohužel (nebo možná naštěstí) neexistuje Google Street View každého místa. Google Street View nám poskytne nejblíže obraz, který je pro požadované lat lon k dispozici. Pokud v blízkosti žádné není, volání vrátí černou obrazovku.

Ve volné přírodě a občas na vesnicích tedy není možné Random Voyager používat, ve městech však funguje bez problémů.

## 3. Technická dokumentace

### 3.1 Git repositář

Zdrojový kód je k dispozici ve veřejném git repositáři. Následujícím programem si ho můžete naklonovat. Kořen repositáře maturitaVT\_repo je dále zmiňován pouze jako /.

```
$ git clone https://github.com/ticvac/maturitaVT_repo.git
```

### 3.2 Programátorská dokumentace

Následující sekce se věnuje tomu, jak aplikaci spustit po stažení git repositáře. Pokud si chcete aplikaci pouze vyzkoušet, doporučuji si ji zadarmo stáhnout z veřejných obchodů - sekce Aplikace volně ke stažení.

#### 3.2.1 Jak nainstaloval potřebné nástroje

##### Simulátor iphonu a instalce na macOS

Pokud používáte jiný operační systém než macOS, simulátor iphonu nemůžete používat. Na macOS je potřeba si nainstalovat Xcode<sup>1</sup>. Není to ale nutné, simulátor androidu funguje i na MacOS. Po instalaci Xcode otevřte soubor maturita\_tichy\_repo/ios/Runner.xcworkspace a v xcode klikněte v horní liště vedle Runner na Get iOS XX.XX.XX. Random Voyager pak můžete spustit na simulátoru který se otevře.

##### Textový editor, IDE

I když není pokročilé IDE nutnou podmínkou pro spuštění kódu, usnadní vám spoustu práce a ulehčí orientaci. Doporučil bych AndroidStudio<sup>2</sup>, které jsem při vývoji používal i já. Velkou výhodou oproti ostatním IDE zde představuje možnost jednoduše přidávat a spravovat emulátory android zařízení, které potřebujeme pro testování aplikace.

(Nebo můžete využívat simulátory iOS z Xcode, to ale lze pouze na macOS).

##### Flutter

Nejdůležitější částí je samotný framework Flutter<sup>3</sup>. Jeho instalace není úplně triviální, ale google udržuje velmi dobrý návod pro všechny 3 hlavní operační systémy.

Všechny postupy instalací zde nemá smysl přesně popisovat (především proto, že se často mění).

---

<sup>1</sup><https://apps.apple.com/cz/app/xcode/id497799835?l=cs&mt=12>

<sup>2</sup><https://developer.android.com/studio>

<sup>3</sup><https://docs.flutter.dev/get-started/install>

## Emulátor android

V android studio klikněte na Tools/DeviceManager a poté Create device. Osobně používám Nexus 6, ale je jedno které zařízení si vyberete. Nainstalujte podle pokynu nejnovější verzi android a spusťte emulátor.

### 3.2.2 Údržba aplikace

V programování a především v oboru vývoje aplikací pro mobily se běžně stává, že co běželo včera dneska vyhazuje error. Flutter má naštěstí dobrou historii co se týče udržování předchozích verzí a následujícím příkazem můžeme jednoduše upgradovat veškeré knihovny a připravit je pro importování. Druhý příkaz pak program spustí, nebo je možné využít tlačítka run vpravo na horní liště. (Nezapomeňte vybrat na jakém zařízení chcete aplikaci spouštět).

```
$ flutter pub get
```

```
$ flutter run
```

### 3.2.3 Struktura aplikace

Při spuštění aplikace se nejprve spustí kód /lib/main.dart, takže je podle mě vhodné začít od něho. Inicializují se zde spojení s firebase a nastaví se cesty pro jednotlivé stránky. Cesta je na začátku nastavena na home\_page.

#### /lib/application

Zde se nachází všechny stránky aplikace, každá v samostatném souboru. Struktura jednotlivé stránky je vždy stejná. Metoda build vrací Frontend aplikace a před ní jsou definované metody na obsluhu, většinou pouze odkazují na funkce kategorie support.

#### /lib/support

Zde se nachází naše tři databáze a kód pro geolocator. Také je zde soubor alert\_dialogues.dart ve kterém jsou pop\_up okna aplikace, která nevyžadují samostatnou stránku. Také obsahuje konstanty a pro nás velmi důležitou konstantu adminState.

#### /lib/tests

Obsahuje jenom jednu stránku admin.dart. Pokud je v /lib/support/constants.dart nastavena konstanta adminState na true, na hlavní stránce se objeví tlačítko, které nás zavede právě na tuto stránku. Na ní je možné schvalovat, případně zamítnout místa, která uživatelé označili jako zajímavá a ještě nejsou k dispozici veřejnosti.

### 3.3 Aplikace volně ke stažení

Pro ios na AppStore<sup>4</sup>.

Pro android na GooglePlay<sup>5</sup>.

### 3.4 Uživatelská dokumentace

Když jsem došel do fáze testování, potřeboval jsem uživatelům poskytnout jednoduchý tutoriál k používání aplikace. Je k dispozici zde<sup>6</sup>. Následuje kopie webové stránky.

#### **Jak aplikace funguje**

V mobilu si můžete zobrazit náhodná místa ve vaší blízkosti a pokud místo znáte nebo vám přijde zajímavé, klikněte na "go find" a lokace se vám uloží v mobilu do databáze.

K databázi máte přístup v sekci "My voyages" a vidíte v ní všechny vaše lokace rozdělené na ty, které jste nenašly a na ty, které ano.

Po kliknutí na konkrétní lokaci můžete kliknout na "I am here" a pokud jste ve vzdálenosti do 50m lokace, uloží se vám jako naleznutá.

#### **Nice places**

Když už jste nějaké místo našli a zdá se vám pěkné, můžete ho zveřejnit ostatním uživatelům. Ti, kteří ho také najdou, k němu mohou přidat jestli se jim líbí nebo ne.

Z bezpečnostních důvodů musejí být v současné době nově zveřejněná místa schválena adminem.

#### **Na co si dát pozor?**

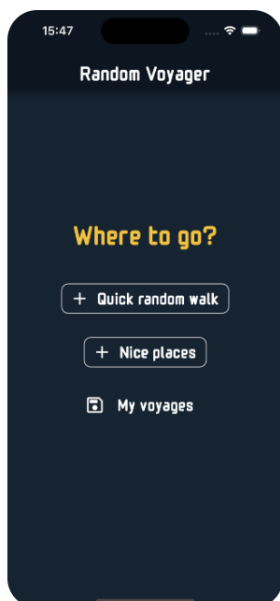
- 1) - Pokud aplikaci zakážete přístup k polohovým údajům, nebude se chovat jak má.
- 2) - Pokud aplikace nereaguje nebo padá, pravděpodobně jste narazili na bug. Jestli máte náladu se o něj podělit, můžete mi napsat na - xticva01@gjk.cz ;)

---

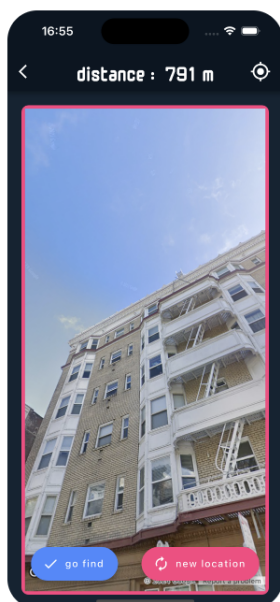
<sup>4</sup><https://apps.apple.com/cz/app/random-voyager/id6475198274?l=cs>

<sup>5</sup>[https://play.google.com/store/apps/details?id=com.vaclav.tichy180.maturita\\_tichy\\_repo&pli=1](https://play.google.com/store/apps/details?id=com.vaclav.tichy180.maturita_tichy_repo&pli=1)

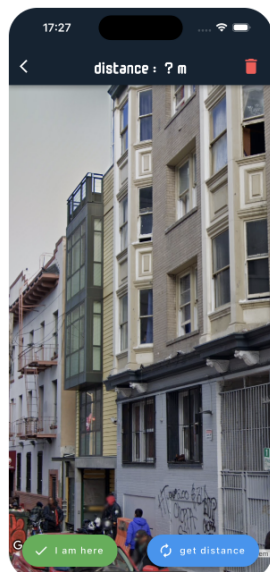
<sup>6</sup>[http://vaclavsilent.cz/random\\_voyager\\_tutorial/](http://vaclavsilent.cz/random_voyager_tutorial/)



- 1) Vydat se na náhodnou procházku,
- 2) Zobrazit veřejně zalogovaná místa,
- 3) Zobrazit vaši databázi,



- 1) Vydat se na místo a uložit ho do databáze.
- 2) Zobrazit nové náhodné místo.



- 1) Jakmile jste 50m od místa můžete si ho zalogovat. Pokud jste dál, nic se nestane.
- 2) Zjistěte vaši vzálenost od místa.



- 1) Místa nadepsané pouze časem jsou vaše soukromá.
- 2) Místa s názvem jsou veřejně dostupná.
- 3) Vaše nalezená místa můžete zveřejnit ostatním uživatelům.



- 1) Klikněte na publish.



1) Veřejná místa se zobrazí v "Nice places".



1) Pokud jste veřejné místo našli, znáčkňte zda se vám líbilo nebo ne.

2) Hodnocení místa je možné vidět všemi i při náhledu.

### Prohlášení a citlivých údajích

Lokace uživatelů nejsou ukládány žádnou třetí stranou.



# Závěr

Práci se mi dařilo plnit podle časového plánu, i když často po větších částech naráz, jak může být viděno v mém git repozitáři. Ze zadání jsem nakonec neimplementoval týdenní úkoly, jelikož realisticky nevím, jestli budu v budoucnosti aplikaci udržovat. Kromě zadání jsem ale přidal pro uživatele možnost hodnocení hledaného místa, která je podle mě mnohem užitečnější.

Během práce jsem narazil na spoustu neočekávaných problémů, od knihoven třetích stran až po konkrétní implementační buggy na mém procesoru. Jelikož musím používat cloudový backend, snažil jsem se minimalizovat šanci najít slabiny programu, které by se daly využít pro ukradení dat z cloudu, případně jeho zahlcení.

Poloha uživatelů je velmi citlivá informace a tak jsem se ji z bezpečnostních důvodů rozhodl nikde neukládat.

S výsledkem aplikace jsem spokojený, mimo rámec mé maturitní práce jsem ji dal veřejně na dva největší obchody aplikací (což samo o sobě nebylo vůbec snadné). Co se týče programování, věnoval jsem se především praktické stránce vývoje aplikace. Pokud bych v budoucnosti chtěl udělat něco jinak, uvažoval bych o spolupráci s frontend designérem.



# Bibliografie

- [1] <https://docs.flutter.dev/>
- [2] <https://firebase.google.com>
- [3] [https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences)
- [4] <https://pub.dev/packages/sqlite>
- [5] <https://pub.dev/packages/geolocator>
- [6] [https://pub.dev/packages/flutter\\_google\\_street\\_view](https://pub.dev/packages/flutter_google_street_view)
- [7] <https://pub.dev/packages/geolocator>
- [8] [https://en.wikipedia.org/wiki/Great-circle\\_distance](https://en.wikipedia.org/wiki/Great-circle_distance)