

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Mobilní aplikace průvodce po městě Pardubice

Michal Jelínek

Bakalářská práce

2025

Rozsah pracovní zprávy: **min. 30 stran**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

BORY, Pavel. C# bez předchozích znalostí. 2. vydání. V Brně: Computer Press, 2022. ISBN 978-80-251-5061-0.  
GOLDMAN, Matt. .NET MAUI in Action. Manning, 2023. ISBN 1633439402.

Vedoucí bakalářské práce: **Ing. Jan Merta, Ph.D.**  
Katedra softwarových technologií

Datum zadání bakalářské práce: **15. prosince 2023**  
Termín odevzdání bakalářské práce: **10. května 2024**

**Ing. Zdeněk Němec, Ph.D.** v.r.  
děkan

L.S.

**Ing. Jan Panuš, Ph.D.** v.r.  
vedoucí katedry

V Pardubicích dne 28. února 2024

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2023/2024

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Michal Jelínek**  
Osobní číslo: **I21358**  
Studijní program: **B0688A140009 Informační technologie**  
Téma práce: **Mobilní aplikace průvodce po městě Pardubice**  
Zadávající katedra: **Katedra informačních technologií**

## Zásady pro vypracování

Cílem práce je vytvoření mobilní aplikace, která běžným uživatelům zobrazí nebo doporučí památky, restaurace, kavárny, hotely, penziony a další zajímavá místa, které mohou navštívit v Pardubicích. Bude umožňovat navigovat uživatele na tato místa a ohodnocovat je. Bude umět zobrazit informace o daných místech a památkách a bude umět pracovat s polohou uživatele. V teoretické části aplikace student rozebere existující alternativy, pardubické informační portály a weby a obecně další průvodcovské aplikace. V praktické části student navrhne a implementuje samotnou mobilní aplikaci. Mobilní aplikace by měla být vytvořena pomocí jazyku C# (nebo Java) a studentem zvoleného frameworku.

## **Prohlášení autora**

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 8.8.2025

Michal Jelínek

## **Poděkování**

Tímto bych chtěl poděkovat Ing. Janu Mertovi, Ph.D. za jeho odborné vedení a cenné rady. Dále také děkuji své rodině, která mě podporovala během mého studia.

## **Anotace**

Cílem práce je vytvoření mobilní aplikace, která běžným uživatelům zobrazí nebo doporučí památky, restaurace, kavárny, hotely, penziony a další zajímavá místa, které mohou navštívit v Pardubicích. Bude umožňovat navigovat uživatele na tato místa a ohodnocovat je. Bude umět zobrazit informace o daných místech a památkách a bude umět pracovat s polohou uživatele. V teoretické části aplikace student rozebere existující alternativy, pardubické informační portály a weby a obecně další průvodcovské aplikace. V praktické části student navrhne a implementuje samotnou mobilní aplikaci. Mobilní aplikace by měla být vytvořena pomocí jazyku C# (nebo Java) a studentem zvoleného frameworku.

## **Klíčová slova**

Mobilní aplikace, Pardubice, Turistický průvodce, doporučování míst, polohové služby

## **Title**

Mobile app guide to the city of Pardubice

## **Annotation**

The purpose of this work is to create a mobile application that allows ordinary users to recommend sights, restaurants, cafés, hotels, guesthouses, and other interesting places that ordinary users can visit in Pardubice. It will enable users to navigate to these places and rate them. It will be able to display information about specific places and sights and will be able to work with the user's location. In the theoretical part of the application, the student will analyze existing alternatives, Pardubice information portals and websites, and other guide applications in general. In the practical part, the student will design and implement the mobile application itself. The mobile application should be created using the C# (or Java) language and a framework chosen by the student.

## **Keywords**

Mobile application, Pardubice, Tourist guide, place recommendation, location services

# Obsah

<b>Seznam zkratk</b> .....	<b>9</b>
<b>Seznam obrázků</b> .....	<b>10</b>
<b>Úvod</b> .....	<b>11</b>
<b>1 Mobilní aplikace pro turisty</b> .....	<b>12</b>
1.1 Význam uživatelských recenzí .....	12
1.2 Přehled existujících řešení a jejich limity.....	13
1.3 Existující aplikace pro turisty a jejich srovnání .....	13
1.3.1 Google Maps .....	13
1.3.2 TripAdvisor .....	14
1.3.3 Mapy.cz .....	14
1.3.4 Další aplikace .....	14
<b>2 Technologie použité ve vývoji</b> .....	<b>15</b>
2.1 Jazyk C# .....	15
2.1.1 Architektura .NET .....	15
2.1.2 Kompilace a spuštění.....	15
2.2 .NET MAUI.....	16
2.2.1 Platformová podpora a API .....	16
2.2.2 Klíčové funkce a výhody.....	17
2.3 Interní relační databáze (SQLite) .....	17
2.4 bcrypt.....	17
2.5 Firebase.....	17
2.5.1 Firebase Cloud Storage.....	18
2.5.2 Cloud Firestore .....	18
2.6 XAML .....	18
2.7 MVVM model .....	19
2.8 Visual studio .....	21
2.8.1 Funkce a možnosti sady Visual Studio.....	21
2.8.2 Instalace a konfigurace .....	21
<b>3 Datový model</b> .....	<b>22</b>
3.1 Architektonický návrh .....	22
3.2 Interní databáze.....	22

3.3	Cloudová databáze Firebase Firestore.....	23
3.4	Propojení datového modelu s aplikací.....	23
<b>4</b>	<b>Návrh aplikace.....</b>	<b>24</b>
4.1	Funkční požadavky.....	24
4.2	Nefunkční požadavky.....	25
<b>5</b>	<b>Vývoj mobilní aplikace.....</b>	<b>26</b>
5.1	Správa dat a session management.....	26
5.1.1	Datový model.....	26
5.1.2	Třída Db.....	26
5.2	Vzhled aplikace.....	27
5.3	Registrace a přihlášení.....	28
5.3.1	Registrace.....	28
5.3.2	Přihlášení.....	29
5.4	Hlavní obrazovka.....	30
5.5	Dlaždice pro přesunutí na místa v Pardubicích.....	30
5.5.1	Společný XAML layout.....	31
5.5.2	Filtrování podle štítků (tagů).....	32
5.5.3	Práce s oblíbenými.....	32
5.5.4	Navigace do map.....	32
5.5.5	Recenze.....	33
5.6	Profil uživatele.....	34
5.6.1	Oblíbené.....	35
5.6.2	Recenze.....	36
5.6.3	Změna hesla.....	37
	<b>Závěr.....</b>	<b>38</b>
	<b>Literatura.....</b>	<b>39</b>

## Seznam zkratk

FF	Firestore
GPS	Global Positioning System
IL	Intermediate Language
IoT	Internet of Things
MAUI	Multi-platform App UI
MVVM	Model-View-ViewModel
NoSQL	non-SQL
OS	Operační systém
XAML	Extensible Application Markup Language

## Seznam obrázků

Obrázek 1 – Vývoj aplikací pro iOS, Android, macOS a Windows z jednoho sdíleného kódu (zdroj: [10]) .....	16
Obrázek 2 – Model MVVM (zdroj [15]).....	19
Obrázek 3 – Diagram případu užití pro mobilní aplikaci (zdroj: vlastní) .....	25
Obrázek 4 – Nastavení pro usera (zdroj: vlastní) .....	26
Obrázek 5 – Obrazovka pro registraci (zdroj: vlastní) .....	28
Obrázek 6 – Obrazovka pro přihlášení (zdroj: vlastní) .....	29
Obrázek 7 – Úvodní obrazovka přihlášeného uživatele (zdroj: vlastní) .....	30
Obrázek 8 – Seznam restaurací (zdroj: vlastní).....	31
Obrázek 9 – Zadní recenze (zdroj: vlastní) .....	33
Obrázek 10 – Profil uživatele (zdroj: vlastní) .....	34
Obrázek 11 – Oblíbené položky (zdroj: vlastní) .....	35
Obrázek 12 – Recenze uživatele (zdroj: vlastní).....	36
Obrázek 13 – Změna hesla (zdroj: vlastní).....	37

## Úvod

V době, kdy chytré telefony a mobilní aplikace hrají čím dál významnější roli v každodenním životě, roste potřeba nástrojů, které lidem usnadní orientaci ve městě, objevování nových míst a sdílení osobních zkušeností. Mobilní zařízení dnes poskytují nepřetržitý přístup k informacím a zároveň vytvářejí prostor pro komunitní sdílení. Proto jsem se rozhodl zaměřit svou bakalářskou práci na vývoj mobilní aplikace, která propojuje užitečné informace o konkrétních místech s možností jejich hodnocení a recenzování.

Cílem aplikace je nabídnout uživatelům jednoduchý a přehledný způsob, jak vyhledávat památky, restaurace, bary nebo obchody ve městě Pardubice, zobrazovat jejich detailní informace a současně se podělit o vlastní zkušenosti formou recenzí a hvězdičkového hodnocení. Díky těmto funkcím může každý uživatel nejen rychle najít místo podle svých preferencí, ale také přispět ke komunitnímu doporučení a pomoci ostatním uživatelům s výběrem. Recenze navíc slouží i jako cenný zdroj zpětné vazby pro provozovatele služeb, kteří mohou reagovat na názory návštěvníků a zlepšovat kvalitu svých podniků.

Při návrhu a vývoji bylo rozhodnuto využít framework .NET MAUI, který umožňuje tvorbu multiplatformních mobilních aplikací. Tento přístup umožnil psát aplikaci jednou a cílit zároveň na více OS. I když MAUI podporuje jak Android, tak i iOS, primárně bylo zaměřeno na platformu Android, a to kvůli mým zkušenostem a skutečnosti, že sám jsem uživatel OS Android

Jako databázové řešení byla zvolena interní databáze s Blowfish Crypt a Firebase Firestore, cloudovou platformu od Googlu, která umožňuje rychlé ukládání, čtení a správu recenzí v reálném čase. Díky tomu si mohou uživatelé snadno prohlížet recenze ostatních, přidávat své vlastní, filtrovat místa podle klíčových slov (například „pizza“, „kavárna“, „tradiční“), a spravovat své příspěvky ve speciální sekci.

Při navrhování funkcionality aplikace se nejvíce zaměřilo na jednoduchost ovládání, rychlou odezvu a celkovou uživatelskou přívětivost. Snahou bylo vytvořit řešení, které bude intuitivní i pro méně technicky zdatné uživatele a zároveň dostatečně robustní pro budoucí rozšíření funkcí.

K výběru tohoto tématu byla moje osobní zkušenost s nedostatkem lokalizovaných aplikací, které by efektivně kombinovaly funkci turistického průvodce s uživatelským hodnocením. Vytvořená aplikace má potenciál usnadnit turistům i místním obyvatelům orientaci v městě, zvýšit kvalitu služeb prostřednictvím recenzí a zároveň podpořit komunitní sdílení informací. Práce mi umožnila propojit praktické programování s reálným dopadem na uživatelskou zkušenost, což považuji za značnou přidanou hodnotu této bakalářské práce.

# 1 Mobilní aplikace pro turisty

V posledních letech se mobilní aplikace staly nepostradatelným nástrojem pro turisty při plánování cest i při samotném pohybu v destinaci. Aplikace jako Google Maps, Mapy.cz nebo TripAdvisor nabízejí rozsáhlé možnosti, jak si vyhledat zajímavá místa, zjistit jejich otevírací dobu, hodnocení od ostatních uživatelů nebo si naplánovat trasu.

Tyto aplikace sice poskytují obrovské množství informací, často ale nejsou dostatečně přizpůsobené potřebám konkrétního města nebo regionu. Proto vznikla myšlenka vytvořit vlastní mobilní aplikaci zaměřenou výhradně na město Pardubice, která má za cíl nabídnout uživatelům přehledné a kvalitní informace o místních památkách, restauracích, barech a obchodech.

Na rozdíl od globálních platforem je má aplikace zaměřená lokálně, což umožňuje pracovat s ověřenými daty, která jsou relevantní právě pro Pardubice. Díky tomu může uživatel rychle získat přehled o zajímavých místech v okolí, snadno je filtrovat podle typu nebo tagů (štítků) a vidět průměrné hodnocení ostatních návštěvníků

Navíc je kladen důraz na jednoduchost použití, přehlednost uživatelského rozhraní a možnost aktivního zapojení uživatelů prostřednictvím recenzí. Cílem je vytvořit komunitní prostředí, kde si lidé mohou navzájem doporučovat místa, která stojí za to navštívit [1]

## 1.1 Význam uživatelských recenzí

V době digitálních technologií hrají uživatelské recenze klíčovou roli při rozhodování o návštěvě určitého místa. Když si samy vybíráme restauraci, bar nebo zajímavost ve městě, často se spoléháme právě na zkušenosti ostatních. Pozitivní recenze nás mohou přesvědčit k návštěvě, zatímco negativní zkušenosti nás naopak varují.

Recenze navíc poskytují cennou zpětnou vazbu nejen dalším uživatelům, ale i samotným provozovatelům podniků. Díky hodnocení a komentářům mohou lépe pochopit, co zákazníci oceňují a kde je naopak prostor ke zlepšení. Tento obousměrný přenos informací je důležitý pro budování kvalitních služeb i důvěryhodnosti.

V rámci aplikace bylo proto rozhodnuto integrovat možnost hodnocení a komentování jednotlivých míst. Každý uživatel má možnost přidat recenzi, ohodnotit dané místo pomocí hvězdiček a sdílet svou zkušenost. Data jsou ukládána do cloudové databáze FF, což zajišťuje jejich okamžitou dostupnost a možnost dalšího zpracování.

Možnost sdílení názoru v aplikaci zvyšuje uživatelské zapojení. Místní obyvatelé i turisté tak přispívají ke komunitnímu charakteru aplikace, který bylo potřeba od začátku podpořit. [2]

## 1.2 Přehled existujících řešení a jejich limity

Před samotným vývojem aplikace byla provedena rešerše existujících mobilních řešení, která nabízejí podobné funkce. Bylo potřeba se zaměřit na známé platformy, jako jsou Google Maps, TripAdvisor, Seznam Mapy nebo Yelp, které umožňují uživatelům vyhledávat zajímavá místa a přidávat k nim recenze.

Při testování těchto aplikací bylo nalezeno několik omezení. Především jde o jejich globální zaměření, kvůli kterému často chybí podrobnější informace o menších městech, jako jsou právě Pardubice. Mnoho míst v těchto aplikacích buď vůbec není zaneseno, nebo jsou údaje o nich zastaralé, případně chybí recenze od českých uživatelů, což snižuje jejich přínos v lokálním kontextu.

Dalším limitem, se kterým se lze setkat, je nedostatečná možnost filtrování. Uživatelé si sice mohou zobrazit kategorie jako „restaurace“ nebo „památky“, ale často není možné zobrazit například jen bary nebo obchody, případně filtrovat podle specifických štítků jako „pizza“, „kavárna“ nebo „regionální kuchyně“.

Některé z těchto aplikací navíc působí složitě a jsou přehlceny funkcemi, které běžný uživatel ani nevyužije. To vedlo k rozhodnutí vytvořit jednodušší, přehlednější a lokalizovanou aplikaci, která by se zaměřila výhradně na město Pardubice a nabídla intuitivní rozhraní pro vyhledávání, hodnocení a recenzování míst.

Cílem aplikace je tedy nabídnout alternativu, která lépe reflektuje potřeby místních obyvatel i turistů. Aplikace, která nebude zahlcena přebytečnými funkcemi, ale naopak nabídne jednoduchý přístup k relevantním informacím a zároveň prostor pro komunitní sdílení zkušeností.

## 1.3 Existující aplikace pro turisty a jejich srovnání

Na trhu dnes existuje celá řada mobilních aplikací, které se zaměřují na podporu turistiky a objevování nových míst. Mezi nejznámější patří Google Maps, TripAdvisor, Mapy.cz, Yelp nebo moderní průvodci typu SmartGuide či Sygic Travel. Tyto aplikace nabízejí širokou paletu funkcí – od navigace, vyhledávání památek, restaurací a hotelů až po možnost přidávat vlastní recenze či nakupovat vstupenky.

Z pohledu uživatele mají tyto aplikace řadu výhod, mezi které patří zejména globální dostupnost, velká uživatelská základna a propojení s dalšími službami. Na druhou stranu se ale vyznačují také nevýhodami – často jim chybí dostatečné pokrytí menších měst, jako jsou Pardubice, a jejich uživatelská rozhraní bývají složitější.

### 1.3.1 Google Maps

Google Maps jsou v současnosti nejrozšířenější aplikací pro navigaci a vyhledávání míst. Nabízejí interaktivní mapy, plánování tras, informace o dopravě, možnost ukládání oblíbených míst a také recenze a hodnocení uživatelů. Silnou stránkou Google Maps je propojení s dalšími službami Google, což umožňuje jednotné přihlášení a personalizaci.

Nevýhodou je, že detailní informace o menších městech, jako jsou Pardubice, mohou být omezené. [3]

### **1.3.2 TripAdvisor**

TripAdvisor je platforma silně orientovaná na cestování, ubytování a gastronomii. Uživatelé zde mohou hodnotit hotely, restaurace i turistické atrakce. Výhodou TripAdvisoru je rozsáhlá databáze recenzí, která pomáhá cestovatelům lépe se rozhodnout. Nevýhodou je naopak nedostatečné množství recenzí z menších měst, kde často chybí lokální zpětná vazba, což snižuje jeho přínos v kontextu Pardubic. [4]

### **1.3.3 Mapy.cz**

Mapy.cz jsou českou alternativou, která je mezi uživateli velmi populární díky důrazu na turistické a cyklistické trasy. Nabízejí offline mapy, plánování výletů a v posledních letech také možnost přidávat recenze a hodnocení. Hlavní výhodou Mapy.cz je detailní znalost českého prostředí a funkce přizpůsobené domácím uživatelům. Slabinou je ale menší propojení s mezinárodními databázemi a menší množství uživatelských recenzí oproti globálním platformám. [5]

### **1.3.4 Další aplikace**

Yelp [6] je populární především v USA a slouží k hodnocení podniků a služeb, ale v České republice není tolik rozšířený. SmartGuide [7] nabízí interaktivní průvodce a offline režim, avšak jeho databáze míst je v menších městech omezená. Sygic Travel [8] poskytuje cestovní itineráře a plánování dovolených, ale je více zaměřen na mezinárodní destinace než na lokální prostředí.

Celkově lze říct, že všechny zmíněné aplikace mají své přednosti, ale zároveň jim chybí plnohodnotná podpora menších měst, jako jsou Pardubice. Z toho důvodu vznikla myšlenka vytvořit aplikaci, která bude kombinovat výhody existujících řešení, avšak s důrazem na lokální specializaci, jednoduchost a komunitní charakter.

## 2 Technologie použité ve vývoji

Cílem bylo vytvořit mobilní aplikaci, která bude uživatelsky přívětivá, technicky spolehlivá a snadno rozšiřitelná o nové funkce. Od začátku jsem kladl důraz na multiplatformní vývoj, možnost snadné správy dat a rychlou odezvu aplikace při práci s databází.

Pro vývoj byl zvolen framework .NET MAUI, který umožňuje psát jednu kódovou základnu pro více platform, přičemž primárním cílem je podpora systému Android. Data aplikace jsou ukládána a spravována v cloudové databázi FF, což zajišťuje snadnou synchronizaci a přístup k datům z různých zařízení.

### 2.1 Jazyk C#

C# je nejrozšířenější jazyk pro platformu .NET které je bezplatné, multiplatformní a open-source vývojové prostředí. Umožňuje vytvářet aplikace pro širokou škálu zařízení, od IoT přes mobilní telefony a počítače až po cloudové servery. [9]

Jedná se o univerzální, multiplatformní jazyk, který kombinuje objektově orientované principy s prvky funkcionálního programování. Díky tomu nabízí vysoký výkon, bezpečnost a produktivitu. Většina knihoven a částí .NET runtime je napsána právě v C#, což z něj dělá klíčový jazyk ekosystému. [9]

Syntaxe C# vychází z rodiny jazyků C, takže je blízká uživatelům C, C++, Javy či JavaScriptu. Používá středníky k ukončení příkazů, složené závorky pro bloky kódu a běžné řídicí struktury jako `if`, `else`, `switch`, `for`, `while` nebo `foreach` pro práci s kolekcemi. [9]

Rozhodnutí pro použití jazyka C# pro tento projekt bylo motivováno zkušenostmi s tímto jazykem, což umožnilo efektivní využití jeho pokročilých funkcí pro vývoj této aplikace.

#### 2.1.1 Architektura .NET

Programy v jazyce C# běží na platformě .NET, která je založena na virtuálním běhovém prostředí CLR a obsahuje rozsáhlou sadu knihoven. CLR představuje implementaci standardu CLI a umožňuje spolupráci mezi více programovacími jazyky. [9]

Významnou předností .NET je schopnost propojit kód napsaný v různých jazycích, například F#, Visual Basic nebo C++. Tyto jazyky jsou překládány do mezikódu IL, jenž odpovídá specifikaci CTS. Díky tomu je možné kombinovat různé moduly, jako by byly vytvořeny v jednom jazyce. Knihovny dostupné v rámci .NET pokrývají široké spektrum potřeb – od práce se soubory a zpracování XML až po vývoj webových aplikací či aplikací s uživatelským rozhraním Windows Forms. [9]

#### 2.1.2 Kompilace a spuštění

Zdrojový kód napsaný v jazyce C# je nejprve přeložen do mezikódu IL. Tento IL je uložen v sestavení s příponou `.dll`, spolu s doprovodnými prostředky, například rastrovými obrázky či textovými řetězci. Součástí sestavení je také manifest, který uchovává metadata o typech,

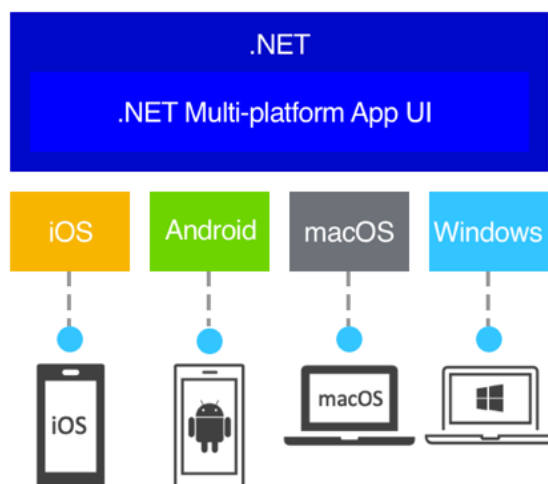
verzích a jazykové mutaci sestavení. Při spuštění aplikace načte CLR toto sestavení a pomocí JIT kompilátoru převádí IL kód na strojové instrukce konkrétního počítače. [9]

## 2.2 .NET MAUI

.NET Multi-platform App UI (.NET MAUI) je open-source framework pro tvorbu nativních mobilních a desktopových aplikací v C# a XAML. Umožňuje vyvíjet aplikace pro Android, iOS, macOS i Windows z jednoho sdíleného kódu, přičemž podle potřeby lze doplnit i části specifické pro konkrétní platformu. Viz. Obrázek 1 – Vývoj aplikací pro iOS, Android, macOS a Windows z jednoho sdíleného kódu (zdroj: [10])

### Pro koho je .NET MAUI vhodný:

- Pro vývojáře, kteří chtějí psát multiplatformní aplikace v C# a XAML ve Visual Studiu.
- Pro ty, kdo chtějí sdílet vzhled a logiku aplikace napříč platformami.
- Pro projekty, kde je cílem opakovaně využít kód, testy i obchodní logiku na více platformách. [10]



Obrázek 1 – Vývoj aplikací pro iOS, Android, macOS a Windows z jednoho sdíleného kódu (zdroj: [10])

### 2.2.1 Platformová podpora a API

.NET MAUI sjednocuje rozhraní API pro Android, iOS, macOS a Windows do jednoho společného celku. Díky tomu je možné psát jednotný kód, který lze následně kompilovat do

nativních balíčků pro jednotlivé platformy. To zahrnuje i specifické případy, jako je Mac Catalyst pro macOS nebo WinUI 3 pro Windows. [10]

Framework současně poskytuje bohaté rozhraní API pro využití nativních funkcí zařízení. V aplikaci tak mohou snadno přistupovat k senzorům, jako je akcelerometr, kompas nebo gyroskop, sledovat změny síťového připojení, získávat podrobnosti o zařízení či pracovat s textem a soubory. Dále je k dispozici podpora pro bezpečné ukládání dat, převod textu na řeč a také pro různé autentizační mechanismy. [10]

### **2.2.2 Klíčové funkce a výhody**

.NET MAUI nabízí sadu ovládacích prvků, podporu datových vazeb i API pro přístup k nativním funkcím zařízení. Součástí jsou také grafické možnosti a projektový systém, který umožňuje cílit jednu aplikaci na více platform. Velmi praktickou funkcí je hot reload, díky němuž lze provádět změny v kódu nebo XAML přímo za běhu aplikace a okamžitě vidět výsledky bez nutnosti celé rekompile. To výrazně urychlilo proces vývoje a testování aplikace. [10]

## **2.3 Interní relační databáze (SQLite)**

Pro ukládání uživatelských účtů, přihlašovacích údajů a dalších lokálních dat bylo použito SQLite lehkou, zabudovanou relační databází, která je součástí mnoha mobilních platform. SQLite je ideální pro ukládání dat, která musí být dostupná offline a rychle načtená bez nutnosti připojení k internetu. [11]

## **2.4 bcrypt**

Opravená, vylepšená a s jmenným prostorem kompatibilní verze BCrypt.Net portu jBCrypt implementovaná v C#. Používá variantu klíčovacího plánu šifrovacího algoritmu Blowfish a zavádí pracovní faktor, který umožňuje určit, jak nákladná bude hash funkce, což umožňuje algoritmu být „odolný do budoucna“. [12]

## **2.5 Firebase**

Google Firebase je sada cloudových vývojových nástrojů, která umožňuje vývojářům mobilních a webových aplikací rychle vytvářet, nasazovat a škálovat jejich projekty. Mezi klíčové funkce Firebase patří autentizace uživatelů, cloudové úložiště, hosting a NoSQL databáze Firestore. [13]

V rámci této aplikace je využívána z platformy Firebase pouze služba Firestore, a to výhradně pro ukládání a načítání uživatelských recenzí a hodnocení míst (památky, restaurace, bary, obchody). Ostatní části aplikace, jako je správa uživatelských účtů a autentizace, jsou řešeny interně pomocí SQLite databáze a zabezpečeny algoritmem bcrypt. [13]

Služba Firebase Firestore je dostupná v rámci bezplatného plánu, který nevyžaduje zadání platebních údajů. Po vyčerpání bezplatných kvót se služba automaticky zastaví, což eliminuje riziko neočekávaných poplatků.

### **2.5.1 Firebase Cloud Storage**

Firebase Cloud Storage nabízí výkonné, bezpečné a škálovatelné řešení pro ukládání souborů a dat v cloudu, které je navrženo speciálně pro potřeby mobilních a webových aplikací. Tato služba umožňuje vývojářům snadno ukládat obrázky, videa a další typy obsahu bez nutnosti spravovat infrastrukturu pro ukládání dat. Integrovaná s robustními bezpečnostními opatřeními, včetně šifrování dat a podrobných pravidel přístupu, zajišťuje, že data jsou chráněna a dostupná pouze pro oprávněné uživatele. Navíc díky automatické škálovatelnosti služba roste s potřebami aplikace, což vývojářům umožňuje bez obav přizpůsobovat kapacitu úložiště aktuálním požadavkům. Cloud Storage je také úzce integrován s ostatními službami Firebase, což umožňuje snadnou synchronizaci s autentizačními a databázovými službami a poskytuje vývojářům komplexní řešení pro správu aplikací. [13]

### **2.5.2 Cloud Firestore**

Cloud Firestore je flexibilní a škálovatelná cloudově-hostovaná NoSQL databáze, navržená pro synchronizaci dat v reálném čase napříč klientskými aplikacemi. Umožňuje přístup prostřednictvím nativních SDK pro platformy Apple, Android a web, stejně jako prostřednictvím Node.js, Java, Python, Unity, C++ a Go SDK.

Firestore je organizována do kolekcí a dokumentů. Data jsou ukládána do dokumentů, které obsahují pole mapující na hodnoty, a tyto dokumenty jsou seskupeny do kolekcí. Firestore podporuje různé typy dat, od jednoduchých řetězců a čísel až po složité, vnořené objekty. Umožňuje vytvářet vnořené kolekce uvnitř dokumentů a stavět hierarchické datové struktury, které se mohou rozšiřovat podle potřeb databáze.

Zabezpečení dat v Cloud Firestore je primárně zajištěno pomocí Firebase Authentication. ale dá se také zabezpečit pomocí různých bezpečnostních pravidel. [13]

## **2.6 XAML**

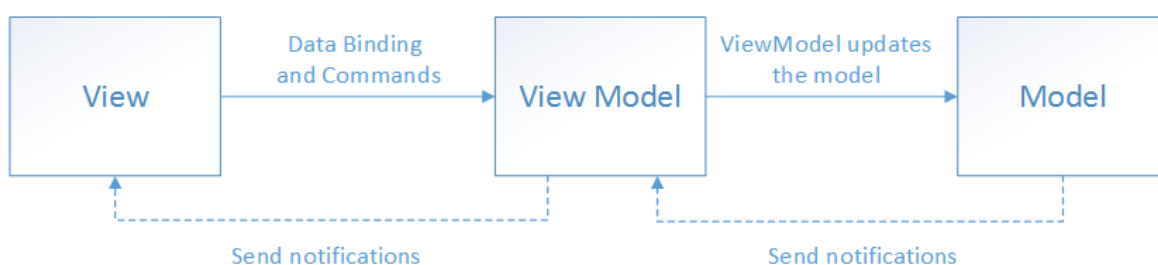
XAML je deklarativní jazyk vycházející z XML, který slouží k popisu uživatelského rozhraní ve frameworku .NET MAUI. Umožňuje navrhovat vzhled aplikace pomocí značek místo psaní procedurálního kódu, což zjednodušuje práci a zvyšuje přehlednost. Přestože použití XAMLu v aplikacích .NET MAUI není povinné, v praxi se často doporučuje, protože struktura rozhraní je díky němu jasně oddělena od aplikační logiky a kód je stručnější.

K výhodám XAMLu patří například možnost využívat speciální typy vlastností, připojené vlastnosti nebo rozšíření značek. Díky tomu je XAML dobře uzpůsobený pro návrhový vzor MVVM, kdy je vzhled (View) oddělen od aplikační logiky (ViewModel) a data se mezi nimi přenášejí prostřednictvím datových vazeb.

Nevýhodou tohoto přístupu je, že XAML neumožňuje přímé vložení programového kódu. Veškerá logika obsluhující události musí být implementována v odpovídajících souborech code-behind v jazyce C#. [14]

## 2.7 MVVM model

Model MVVM (Model–View–ViewModel) slouží k jasnému oddělení uživatelského rozhraní od aplikační a obchodní logiky. Díky tomuto rozdělení se aplikace lépe udržuje, snadněji testuje a je přehlednější pro další rozvoj. Tento přístup také zvyšuje možnost znovupoužití kódu a umožňuje vývojářům a designérům pracovat na svých částech aplikace nezávisle.



Obrázek 2 – Model MVVM (zdroj [15])

Architektura MVVM se skládá ze tří hlavních částí:

- Model – reprezentuje samotná data a obchodní logiku aplikace.
- View (zobrazení) – definuje vzhled aplikace a způsob, jakým jsou uživateli prezentována data.
- ViewModel (model zobrazení) – propojuje View s Modelem, zajišťuje zpracování uživatelských akcí a předávání dat.

Hlavní myšlenkou MVVM je, že View zná ViewModel, ViewModel zná Model, ale Model a View spolu přímo nekomunikují. Tím se docílí toho, že změny v jedné vrstvě neovlivní ostatní, pokud jsou dodrženy principy architektury. [15]

View určuje strukturu a vzhled uživatelského rozhraní, které je prezentováno uživateli. Obvykle je definováno v jazyce XAML a mělo by být odděleno od ostatních vrstev. ViewModel představuje prostředníka mezi View a Modelem. Poskytuje data a příkazy, na které se View může navázat, a reaguje na uživatelské akce. Zároveň převádí data z Modelu do podoby vhodné pro zobrazení. [15]

Model obsahuje samotná data aplikace a obchodní logiku. Zajišťuje datové operace, pravidla a funkcionalitu nezávislou na uživatelském rozhraní.

Hlavní výhodou MVVM je jasné oddělení vrstev, což usnadňuje údržbu i testování. ViewModel je snadno testovatelný, protože není závislý na uživatelském rozhraní. Tento přístup také umožňuje paralelní práci vývojářů a designérů. [15]

## 2.8 Visual studio

Visual Studio je komplexní integrované vývojové prostředí (IDE), které poskytuje vývojářům nástroje a funkce pro kompletní vývojový cyklus aplikací, od psaní kódu přes jeho ladění až po nasazení. Visual Studio podporuje množství programovacích jazyků a platforem, umožňuje efektivní spolupráci na projektech a integraci správy verzí. Díky svým rozsáhlým funkcím je ideální pro tvorbu, testování a úpravu aplikací pro různé platformy, včetně mobilních a desktopových aplikací v .NET, ASP.NET a C++.

### 2.8.1 Funkce a možnosti sady Visual Studio

Sada Visual Studio je vybavena nástroji, které zjednodušují vývojářské procesy. Editor kódu a jeho přizpůsobení, uspořádání kódu a návrhové nástroje usnadňují psaní čistého a efektivního kódu. Dále IDE nabízí integrované ladění, které pomáhá diagnostikovat a řešit problémy v kódu, podporuje testování jednotek pro zajištění kvality kódu, a umožňuje snadné nasazení aplikací na různé platformy. Podpora vývoje s asistencí umělé inteligence, jako jsou GitHub Copilot a IntelliCode, zvyšuje efektivitu psaní kódu tím, že předpovídá potřeby vývojářů a automaticky doplňuje kód. [16]

### 2.8.2 Instalace a konfigurace

Pro prvotní začátek využívání Visual Studia musí vývojáři nejprve vybrat a stáhnout požadovanou edici IDE, která nejlépe odpovídá jejich potřebám. Existuje několik edicí od bezplatné komunitní verze pro studenty a individuální vývojáře, přes profesionální pro malé týmy, až po enterprise verzi pro velké organizace. [16]

### 3 Datový model

Pro návrh datového modelu aplikace bylo potřeba zohlednit dva hlavní zdroje dat – interní relační databázi pro správu uživatelských účtů a cloudovou NoSQL databázi FF pro ukládání uživatelských recenzí. Tato kombinace umožňuje efektivně oddělit uživatelskou autentizaci od dynamického obsahu, což přispívá k vyšší bezpečnosti a lepší škálovatelnosti.

#### 3.1 Architektonický návrh

Architektura aplikace je rozdělena do několika vrstev a modulů, které zajišťují přehlednost, modularitu a snadnou rozšiřitelnost:

- Aplikace je navržena jako modulární, s jasným oddělením jednotlivých částí.
- Prezentační vrstva – tvořená XAML soubory a jejich code-behind částmi v C#, zajišťuje zobrazování dat a interakci s uživatelem.
- Logická vrstva – obsahuje třídy a metody pro zpracování dat, práci s recenzemi, filtrování a vyhledávání.
- Datová vrstva – kombinuje interní databázi pro správu uživatelů a jejich autentizaci se zabezpečením hesel pomocí algoritmu bcrypt a cloudovou databázi FF, která slouží výhradně pro ukládání a načítání uživatelských recenzí.
- Autentizační modul – zajišťuje přihlašování a registraci uživatelů proti interní databázi, přičemž hesla jsou ukládána bezpečně pomocí bcrypt hashování.

#### 3.2 Interní databáze

Interní databáze je relačního typu a slouží k uchovávání informací o registrovaných uživateli. Obsahuje například údaje jako:

- ID uživatele (primární klíč),
- jméno a příjmení,
- e-mailová adresa (jedinečný identifikátor),
- hash hesla (uložený pomocí bcrypt),
- role uživatele (např. běžný uživatel, administrátor).

Pro autentizaci se používá bcrypt hashování, které zajišťuje vysokou úroveň zabezpečení hesel. Díky tomu není možné hesla jednoduše zpětně získat ani při kompromitaci databáze.

### 3.3 Cloudová databáze Firebase Firestore

Firebase Firestore je dokumentově orientovaná NoSQL databáze, která slouží pro ukládání uživatelských recenzí a hodnocení jednotlivých míst. Každá recenze je uložena jako samostatný dokument v kolekci reviews a obsahuje například:

- ID recenze,
- e-mail uživatele, který recenzi přidal,
- text recenze,
- hodnocení (číselná hodnota 1–5),
- zeměpisné souřadnice místa (lat, lon),
- název a typ místa (např. restaurace, památka, bar, obchod),
- čas přidání recenze (timestamp).

### 3.4 Propojení datového modelu s aplikací

Aplikace využívá architekturu MVVM, což znamená, že datový model je oddělen od prezentační logiky. Data z interní databáze i Firestore jsou načítána přes servisní třídy, které zajišťují validaci a případné transformace dat před jejich předáním do uživatelského rozhraní.

Takové řešení zajišťuje:

- vyšší přehlednost kódu,
- snadnou údržbu a rozšiřitelnost,
- možnost snadné výměny databázového řešení v budoucnosti.

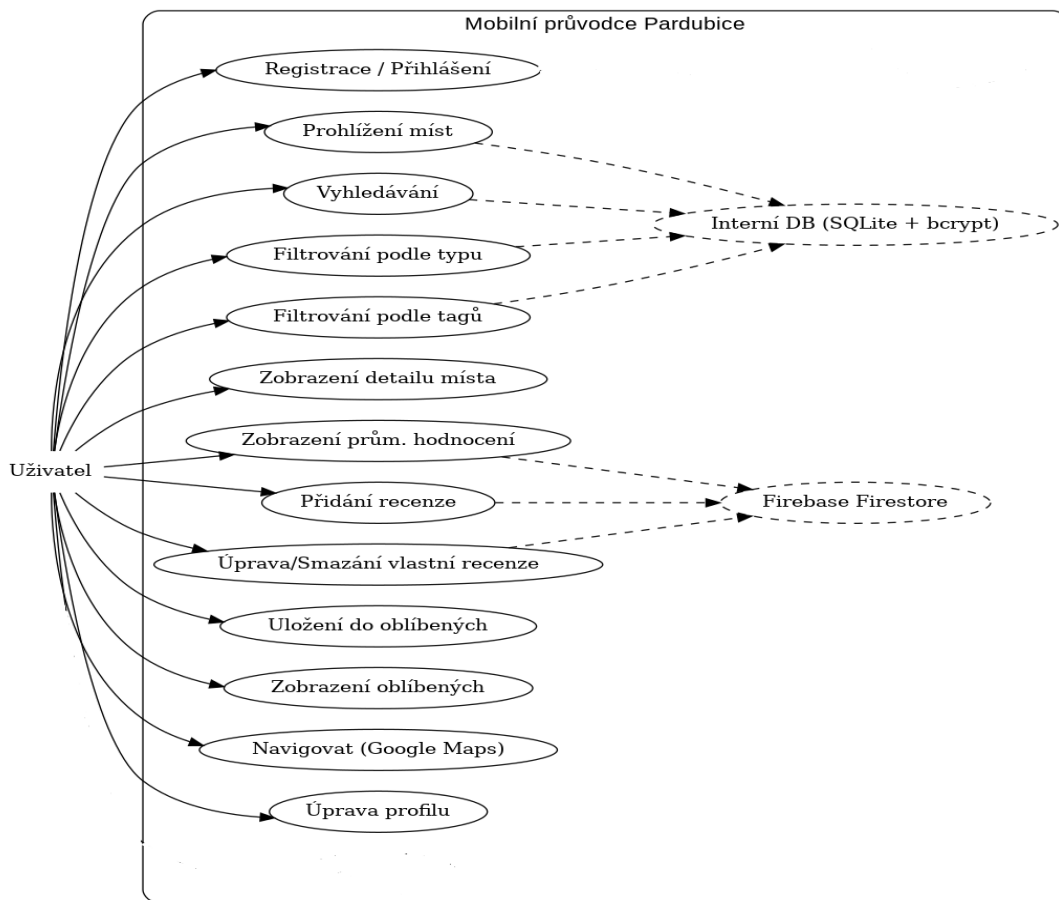
## 4 Návrh aplikace

Mobilní aplikace průvodce po městě Pardubice je určena pro širokou veřejnost, jak místním obyvatelům, tak návštěvníkům města. Slouží k objevování a hodnocení zajímavých míst (památky, restaurace, bary, obchody). Cílem aplikace je poskytnout uživatelům jednoduchý a přehledný nástroj, který jim umožní nejen získat základní informace o jednotlivých místech, ale také sdílet vlastní zkušenosti prostřednictvím recenzí a hodnocení

### 4.1 Funkční požadavky

- FP 1: Autentizace uživatelů
  - FP 1.1: Aplikace bude umožňovat registraci a přihlášení uživatelů pomocí e-mailu a hesla.
  - FP 1.2: Uživatelé se budou moci odhlásit a podívat se na svůj profil na hlavní stránce.
- FP 2: Správa uživatelského profilu
  - FP 2.1: Uživatelé budou moci na svém profilu upravovat své údaje:
    - FP 2.1.1: Uživatelé budou moci měnit své heslo.
    - FP 2.1.2: Uživatelé budou moci měnit svůj avatar.
- FP 3: Správa oblíbených a recenzovaných míst pro přihlášené uživatele:
  - FP 3.1: Přihlášení uživatelé mohou zobrazit své oblíbená místa.
    - FP 3.1.1: Uživatelé mohou své položky v oblíbených filtrovat pomocí dodatečný typu (památky, bary, hospody, obchody.
    - FP 3.1.2: Uživatelé taky mohou odebrat místo z oblíbených míst.
    - FP 3.1.3 Uživatelé mohou se přes tlačítko navigovat nechat navést přímo k místu, kam chtějí zajít.
  - FP 3.2: Přihlášení uživatele si mohou zobrazit taky své zadané recenze
    - FP 3.2.1: Uživatelé budou moci vidět, jaké dali místu hodnocení jak hvězdičkové, tak i textové, dále uvidím i čas a datum, kdy byla recenze zadaná.
    - FP 3.2.2: Uživatelé budou moci i danou recenzi odstranit.
- Prohlížení a filtrování podniků a další funkcionality
  - FP 3.1: Aplikace bude zobrazovat seznam všech podniků podle výběru z hlavní stránky.
  - FP 3.2: Uživatelé budou moci filtrovat podniky podle určitých tagů.
  - FP 3.3: Uživatel bude moci si určitý podnik nebo místo přidat mezi svá oblíbená místa.
  - FP 3.4: Uživatel bude moci k určitému místu či podniku napsat recenzi textem, fotkou a zároveň i ohodnotí hvězdami od 1 do 5.
  - FP 3.3: Uživatel bude moc si přes tlačítko navigovat nechat přenes na Google mapy které pomocí zeměpisné polohy ukážou cestu k danému místu či podniku

Pro lepší zobrazení funkcionalit aplikace byl vytvořen diagram případu užití, který tyto funkcionality specifikuje.



Obrázek 3 – Diagram případu užití pro mobilní aplikaci (zdroj: vlastní)

## 4.2 Nefunkční požadavky

- NP 1 – Kompatibilita
  - NP 1.1 – Aplikace bude kompatibilní s nejnovějšími verzemi operačního systému Android.
- NP 2 – Responzivita
  - NP 1.2 – Aplikace bude responzivní a optimalizovaná pro různé velikosti mobilních obrazovek.
- NP 3 – Rozšiřitelnost
  - Aplikace bude navrhována tak, aby umožnila rozšiřování a přidávání nových funkcí v budoucnu.
- NP 4 – Výkon
  - Aplikace bude navrhována tak, aby umožnila minimální spotřebu systémových zdrojů.

## 5 Vývoj mobilní aplikace

Aplikace je vytvořena s využitím .NET MAUI a pomocí implementace architektury MVVM jsou třídy rozděleny podle funkcionality, což zajišťuje jasné oddělení prezentační vrstvy od aplikační logiky. Tento přístup umožňuje snadnou údržbu, testování a budoucí rozšiřování aplikace.

### 5.1 Správa dat a session management

#### 5.1.1 Datový model

V aplikaci jsou uživatelská data uložena v SQLite tabulce User. Třída User slouží zároveň jako datový model pro ORM mapování a je označena atributy PrimaryKey a AutoIncrement pro identifikátor. Takových datových modulů se tu nachází více, a to pro ukládání oblíbených míst ve třídě „favorites“, která přebírá potřebné atributy pro uložení do databáze oblíbených.

```
namespace Bakalarka_Jelinek
{
    [Table("users")]
    Počet odkazů: 51
    public class User
    {
        [PrimaryKey, AutoIncrement]
        Počet odkazů: 0
        public int Id { get; set; }

        [NotNull]
        Počet odkazů: 3
        public string Name { get; set; }

        [NotNull, Unique]
        Počet odkazů: 27
        public string Email { get; set; }

        [NotNull]
        Počet odkazů: 4
        public string Password { get; set; }

        [NotNull]
        Počet odkazů: 1
        public string Role { get; set; }

        Počet odkazů: 4
        public string ProfilePhotoPath { get; set; }

        // Statické vlastnosti pro udržení stavu přihlášení
        [Ignore]
        Počet odkazů: 40
        public static User? CurrentUser { get; set; }

        [Ignore]
        Počet odkazů: 1
        public static bool IsLoggedIn { get; internal set; }
    }
}
```

Obrázek 4 – Nastavení pro usera (zdroj: vlastní)

#### 5.1.2 Třída Db

Třída Db zajišťuje kompletní komunikaci aplikace s interní databází SQLite a implementuje metody pro práci s uživateli, jako jejich registrace a pak možné přihlášení do aplikace, zpracovává i uživatelské změny hesla. Dále pracuje i s oblíbenými položkami, kde ukládá k uživateli jeho oblíbená místa. Při prvním spuštění aplikace inicializuje potřebné tabulky a provádí kontrolu jejich struktury. Díky použití SQLiteAsyncConnection jsou všechny databázové operace asynchronní, aby nedocházelo k blokování hlavního vlákna uživatelského rozhraní.

Funkce a odpovědnosti:

Inicializace databáze – vytvoření tabulek User a Favorite při prvním spuštění nebo po změně schématu. Tato funkce zajišťuje, aby byla databáze vždy připravena k použití.

Správa uživatelů – registrace, ověřování přihlášení (bcrypt hash), kontrola existence uživatele, změna hesla, aktualizace profilové fotografie.

Správa oblíbených položek – přidání, odebrání, kontrola, získání seznamu oblíbených míst podle typu (památky, restaurace, obchod, bar).

Napojení na třídu Mapa – při načítání oblíbených položek podle typu si aplikace stahuje kompletní data z externích zdrojů (API) a filtruje je podle uložených souřadnic. Napojuje se na mapy od OpenStreetMap.

## 5.2 Vzhled aplikace

Uživatelské rozhraní aplikace bylo navrženo s důrazem na jednoduchost, přehlednost a snadnou orientaci uživatele. Vzhled je konzistentní napříč celou aplikací a využívá barevnou kombinaci červené, modré a bílé, která odpovídá městským barvám Pardubic.

Úvodní obrazovka (Home) nabízí rychlý přístup ke čtyřem hlavním kategoriím Památky, Restaurace, Obchody a Bary. Tyto kategorie jsou zobrazeny ve formě velkých barevných tlačítek s ikonami a popisem, což uživateli umožňuje intuitivní navigaci. V horní části je také umístěna možnost přihlášení a registrace.

Seznamové stránky (např. seznam památek) jsou navrženy tak, aby poskytovaly jasné a stručné informace o jednotlivých místech. Každá položka obsahuje název, kategorii, adresu (pokud je dostupná) a GPS souřadnice. Součástí každého záznamu jsou i dvě výrazná tlačítka *Navigovat* (otevřít trasu v Google Maps) a *Recenze* (umožňuje uživateli přidat hodnocení a komentář platí jen pro přihlášené uživatele. Ti mají u místa ještě hvězdu pro přidání do oblíbených).

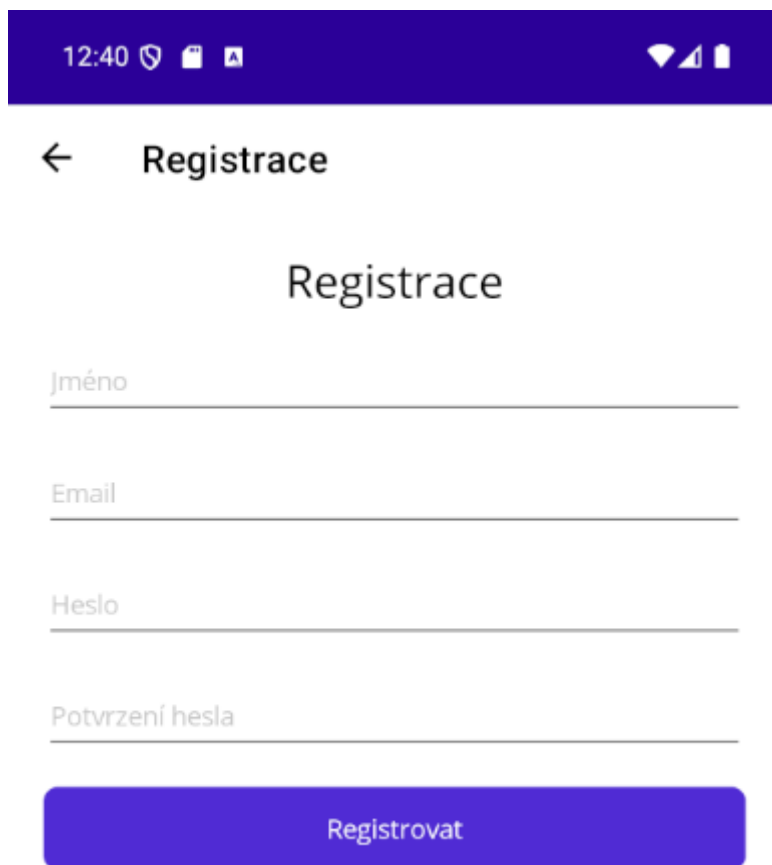
Rozhraní je optimalizováno pro mobilní zařízení s různými velikostmi obrazovky. Při větším množství obsahu se využívá vertikální posuvník, který zajišťuje dostupnost všech prvků.

Ikony, barvy a velikosti prvků byly voleny tak, aby byly snadno čitelné a přístupné i pro uživatele s horším zrakem.

## 5.3 Registrace a přihlášení

### 5.3.1 Registrace

Registrace je řešena lokálně přes interní relační databázi SQLite. Vytváří se tabulka user a při ukládání hesla se používá knihovna BCrypt.net (automaticky generuje unikátní salt a ukládá se pouze hash).



The image shows a mobile application interface for registration. At the top, there is a status bar with the time 12:40 and various icons. Below it is a navigation bar with a back arrow and the title 'Registrace'. The main content area is titled 'Registrace' and contains a form with four input fields: 'Jméno', 'Email', 'Heslo', and 'Potvrzení hesla'. At the bottom of the form is a large blue button labeled 'Registrovat'.

Obrázek 5 – Obrazovka pro registraci (zdroj: vlastní)

Uživatel musí vyplnit formulář o základních údajích, poté jsou provedeny základní validace, jestli je vše správně vyplněno (neprázdná pole, formát e-mailu, shoda hesel, minimální síla hesla). Ověření, že e-mail v DB neexistuje (`UserExistsByEmailAsync`). Heslo je zahashované přes `BCrypt.HashPassword` a nový uživatel se vloží do SQLite (`AddUserAsync`). Poté se zobrazí potvrzení a je nabídnuto uživateli se přihlásit.

Datová vrstva je zajištěna třídou `Db`, která udržuje `SQLiteAsyncConnection` do souboru v `FileSystem.AppDataDirectory`. V konstruktoru je volaná inicializace schématu a vytváří tabulky `User`, `Favorite`:

- `AddUserAsync(User user)` – vloží uživatele (uložen je pouze bcrypt hash).
- `UserExistsByEmailAsync(email)` – kontrola duplicity e-mailu.

- `AuthenticateUserAsync(email, password)` – načte uživatele a ověří heslo přes `BCrypt.Verify(...)`.

### 5.3.2 Přihlášení

Přihlášení využívá metodu `AuthenticateUserAsync(email, password)`. Nejprve se načtou data uživatele podle e-mailu; pokud existuje, ověří se heslo pomocí `BCrypt.Verify(plain, hash)`. Při úspěchu se uloží přihlášený do `User.CurrentUser` a poté se provede navigaci zpět do aplikace.

---

← Přihlášení

Přihlášení

Email

---

Heslo

---

Přihlásit se

Obrázek 6 – Obrazovka pro přihlášení (zdroj: vlastní)

## 5.4 Hlavní obrazovka

Hlavní obrazovka je vybavena čtyřmi tlačítky dlaždicového stylu, ve kterých si uživatel může načíst vybraná místa v Pardubicích, Tlačítka jsou stylována do čtvercových dlaždic, s piktoqramy, které naznačují, o jaký typ místa jde. Tento vzhled byl inspirován dopravním podnikem města Pardubic, kde využívají podobný vizuální styl. Tlačítka jsou i různě zbarvená. Do palety byly zvoleny barvy červená a modrá, jelikož se tyto barvy objevují ve znaku Pardubického kraje.

Dále se hlavní obrazovka v pravém horním rohu obrazovky liší podle toho, zda je uživatel přihlášen, či nikoliv. Když není přihlášen, může vidět tlačítka „Přihlásit“ a „Registrace“. Pokud už je ale přihlášen, tak uvidí tlačítka „Profil“ a „Odhlásit“.

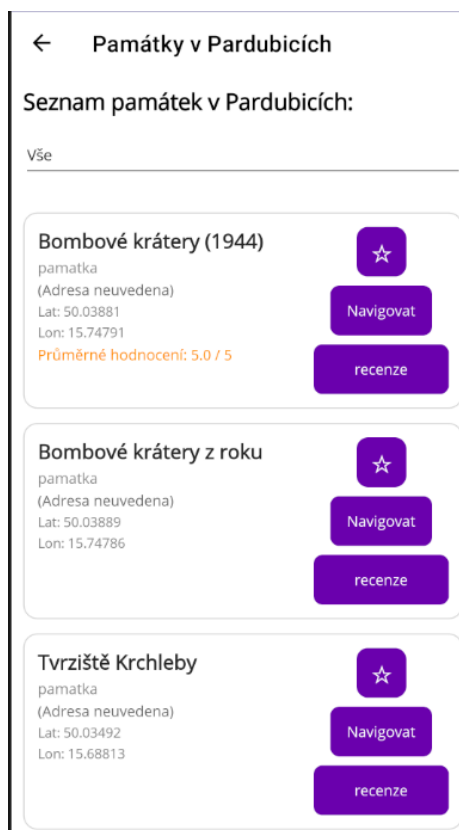


Obrázek 7 – Úvodní obrazovka přihlášeného uživatele (zdroj: vlastní)

## 5.5 Dlaždice pro přesunutí na místa v Pardubicích

Cílem všech těchto stránek je zobrazit seznam pardubických míst daného typu a umožnit jejich filtrování podle štítků (tagů). Tagy rozlišující místa podle toho, jaká je kuchyně, nebo podle toho co daný obchod prodává. Přidávání do oblíbených, spuštění navigace, otevření obrazovky s recenzemi a následně i možnost napsat svojí vlastní recenzi. Každé toto místo je ještě mezi sebou rozděleno pomocí rámečků s kulatými rohy, aby mezi sebou nesplývaly a vše bylo uspořádané. Všechny stránky sdílí stejný UX vzor, XAML layout i programovou logiku. Liší se jen zdrojovou metodou pro načtení dat ze serveru OpenStreetMap a drobnosti tagování, které mají každý jiné, protože u každého míst.

Při vytvoření stránky se volá asynchronní metoda `Nacist*()` (`NacistPamatky`, `NacistRestaurace`, `NacistObchod`, `NacistBar`). Podle toho, jaké místo vyhledáváme dle typu.



Obrázek 8 – Seznam restaurací (zdroj: vlastní)

Data se získávají ze třídy `Mapa` (např. `Mapa.ZiskejPamatkyVPardubicichAsync()`).

Po načtení se doplní stav „oblíbených“ z lokální SQLite DB přes třídu `Db` (`GetFavoritesByTypeAsync(...)`) a u každé položky se nastaví: `IsUserLoggedIn` (podle aktuální session), `IsFavorite` a `FavoriteIcon` („★/☆“). U barů/restaurací/památek/obchodů se rovnou načítá průměrné hodnocení z Firebase Firestore (agregace uživatelských recenzí). Hodnota se drží v poli `PrumerneHodnoceni` a v UI se zobrazuje jen pokud je větší než 0.

### 5.5.1 Společný XAML layout

Všechny seznamy používají stejnou šablonu karty (`Border + Grid`). V levé části je název místa, jeho typ, popřípadě adresa, pokud byla na `openstreetmap` uvedena. Pod adresou je ještě uvedena zeměpisná šířka a výška. V pravé části jsou tři tlačítka

1. První tlačítko hvězda, která slouží k přidávání do oblíbených. Mohou ji vidět pouze přihlášení uživatelé.
2. Druhé tlačítko slouží k navigaci uživatele na dané místo.
3. Třetí tlačítko `recenze`, které uživatele přesune na stránku s recenzemi.

Nad tímhle vším je položka filtru, kde je možnost filtrovat podle různých tagů.

### 5.5.2 Filtrování podle štítků (tagů)

Všechny tagy se berou z načteného seznamu (`SelectMany(p => p.Tagy)`), očistí (např. od „yes“), a seřadí. Do Pickeru se vloží položka „Vše“ + získané tagy. Změna výběru spouští filtr (in-memory) a rebinding `CollectionView`. Počet tagů a jejich možnosti se liší od typu míst, které uživatel rozklikl. U restauraci jsou tagy podle typu kuchyně např. pizza, sushi... U obchodů zase podle nabídky od potravin, kosmetiku, kola, oblečení až třeba po pracovní nástroje u obchodů je zrovna nejvíce tagů.

### 5.5.3 Práce s oblíbenými

Tlačítko s hvězdou přepíná stav oblíbené položky. Nevybarvená hvězda znamená, že uživatel nemá dané místo přidáno ve svých oblíbených položkách. Naopak, když je hvězda vybarvená, znamená to, že dané místo má uživatel přidané ve svém seznamu oblíbených míst. Pokud není uživatel přihlášen, nemůže si nic přidávat do oblíbených míst.

Přidání/odebrání volá metody:

- `Db.AddFavoriteAsync(...)`,
- `Db.RemoveFavoriteAsync(...)`,
- UI se ihned obnoví (rebind).

### 5.5.4 Navigace do map

Přes tlačítko „Navigovat“ aplikace přesune uživatele do Google map, kde se start navigování vyplní podle aktuálních GPS souřadnic uživatele a cíl navigování se načte ze zeměpisné šířky a výšky (lat, lon) daného místa a najde cestu pro uživatele. Pak už si způsob dopravy může uživatel vybrat podle sebe, jestli pěšky, autobusem nebo třeba autem.

### 5.5.5 Recenze

Tlačítko „recenze“ otevře novou stránku s recenzemi, kde lze vidět, jaké recenze byly k danému místu uděleny a počet hvězd. Dále je vidět textové ohodnocení (pokud bylo uděleno) a to samé platí pro fotografii, kterou je možné k recenzi také přidat. V horní části je i vidět datum a čas, kdy byla recenze udělena a kdo recenzi pro dané místo udělil

V dolní části stránky je tlačítko, pomocí kterého má uživatel možnost sám přidat svoji recenzi k danému místu. Na stránce uživatel uvidí název místa, které hodnotí, pod názvem je textové pole pro text hodnocení. Pod tímto polem je hvězdičkové hodnocení, kde může udělit danému místu počet hvězd. Je zde i možnost „přidat fotku“, která mu otevře galerii a umožní přidat fotografii daného místa z uživatelské galerie. Pak už jen stačí odeslat recenzi pomocí tlačítka, recenze se uloží k danému místu do online databáze a je viditelná ostatním uživatelům aplikace.

← Přidat recenzi

Přidat recenzi pro: Bombové krátery  
(1944)

Zde napište svou recenzi...

Hodnocení:

☆☆☆☆☆

Hodnocení: 0 / 5

Přidat fotku

Odeslat recenzi

Obrázek 9 – Zadní recenze (zdroj: vlastní)

## 5.6 Profil uživatele

Uživatelský profil je unikátní pro každého uživatele, nicméně jeho rozvržení je stejné pro všechny. Profil obsahuje profilový obrázek, který si uživatel může vložit/změnit sám z galerie svého telefonu, kam se dostane pomocí tlačítka „změnit fotku“. Pod tímto tlačítkem se dále nachází pole další tlačítek jako „oblíbené“ a „recenze“. Pod těmito tlačítky je viděn formulář s informacemi o profilu (jméno, e-mail), a možnost změnit heslo.

← Profil

Změnit fotku

Nastavení Oblíbené Recenze

Jméno:  
pepa

Email:  
pepik@seznam.cz

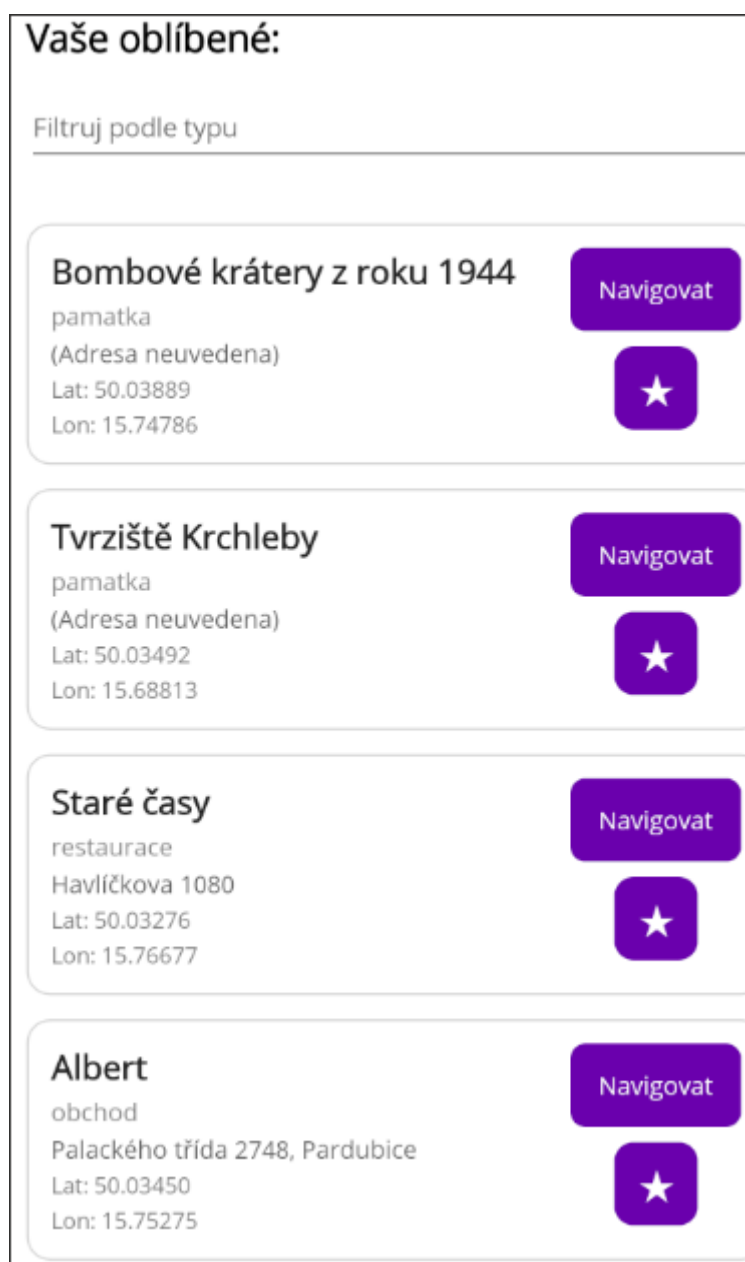
Heslo:  
.....

Změna hesla

Obrázek 10 – Profil uživatele (zdroj: vlastní)

### 5.6.1 Oblíbené

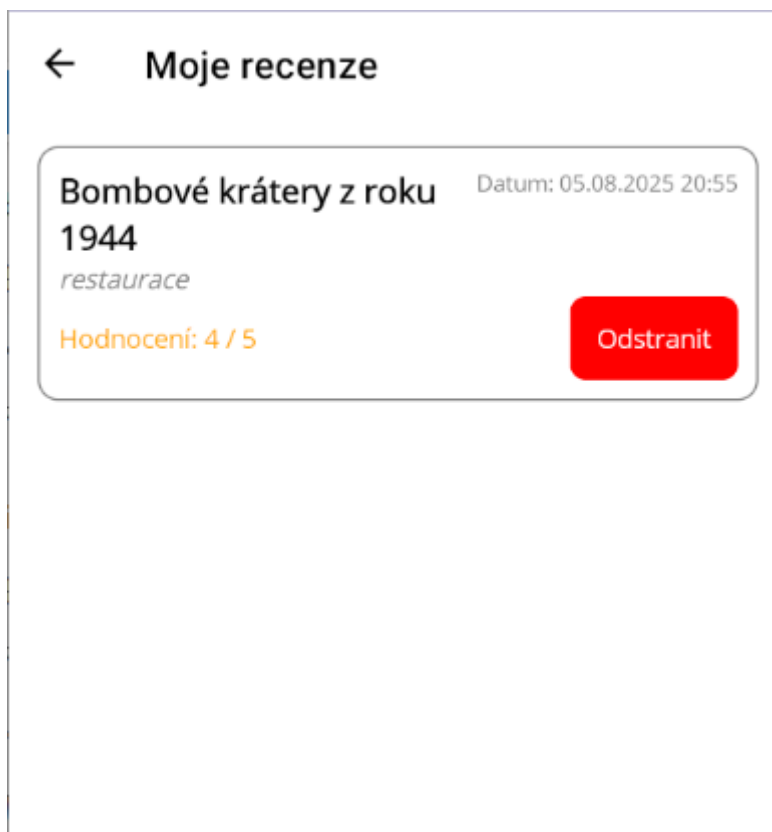
První tlačítko pro zobrazení oblíbených míst uživateli otevře kartu s jeho oblíbenými místy a podniky. Limit na oblíbená místa není nijak stanoven. Každé místo má svůj vymezený box se zaoblenými rohy, kde je uvedeno název, typ, adresa a zeměpisná šířka a výška. Na pravé straně jsou tlačítka na odebrání z oblíbených (symbol vyplněné hvězdy) a navigovat, kde navigovat přesune uživatele na Google mapy a nasměruje k danému místu. Pro odebrání ze seznamu je potřeba kliknutí na hvězdičku, poté se zobrazí vyskakovací okno, jestli chce dané místo uživatel odebrat z oblíbených. Dále je nad všemi oblíbenými místy možnost filtrovat podle typu (památky, bar, restaurace, obchod).



Obrázek 11 – Oblíbené položky (zdroj: vlastní)

### 5.6.2 Recenze

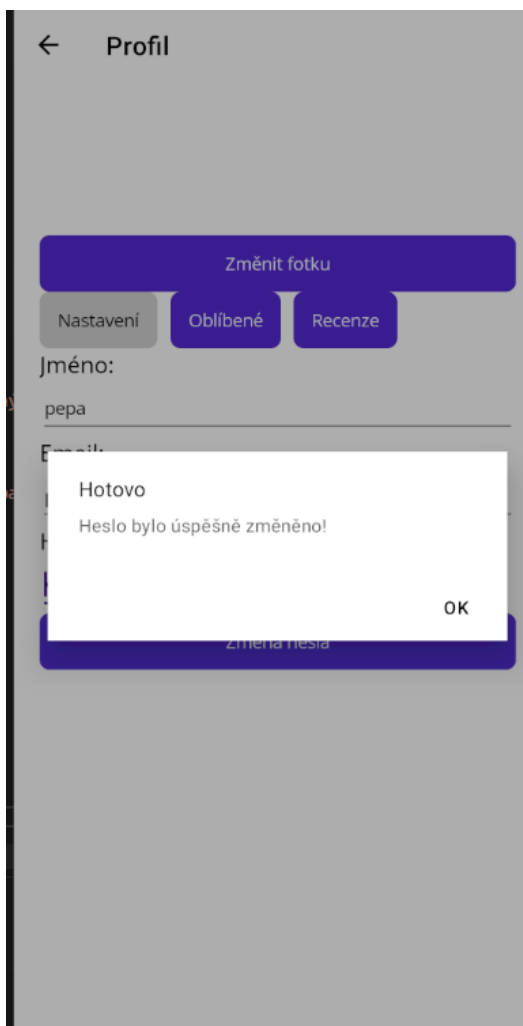
Zobrazí stránku, kde jsou vidět místa, která uživatel ohodnotil. Zobrazí se místo, jeho název, typ, a to jaké hvězdičkové hodnocení a text, který uživatel napsal k danému místu. V pravém horním rohu je zobrazen datum a čas, kdy byla recenze vytvořena, a pod ním tlačítko k odstranění recenze. Po kliknutí na něj se zobrazí vyskakovací okno s dotazem, jestli chce uživatel danou recenzi opravdu odstranit. Uživatel musí svůj záměr potvrdit, aby recenze byla odstraněna. Po odsouhlasení se uživateli odstraní recenze ze seznamu a je i následně odstraněna z firebase databáze.



Obrázek 12 – Recenze uživatele (zdroj: vlastní)

### 5.6.3 Změna hesla

Uživatel má možnost i změnit si heslo. Kliknutím na tlačítko „změna hesla“ je uživatel nejdříve vyzván pro napsání nového hesla, které musí splnit určité podmínky a poté ho musí podruhé potvrdit, aby se heslo opravdu změnilo. Tato dvě hesla se musí shodovat. Poté vyskočí informativní okno, že bylo heslo změněno. Poté stačí odkliknout potvrzení OK a uživatel může pokračovat v prohlížení aplikace už s novým heslem



Obrázek 13 – Změna hesla (zdroj: vlastní)

## Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat mobilní aplikaci, která uživatelům poskytne přehled o památkách, restauracích, barech a obchodech v Pardubicích, a zároveň jim umožní tato místa hodnotit, psát k nim recenze a ukládat si je mezi oblíbené. Během vývoje jsem se zaměřil na propojení více technologií od zpracování mapových dat z OpenStreetMap, přes práci s databází SQLite pro lokální úložiště oblíbených položek, až po využití služby Firebase Firestore pro ukládání a načítání uživatelských recenzí a hodnocení.

Aplikace byla vyvinuta v prostředí .NET MAUI s architekturou MVVM, což umožnilo jasně oddělit logiku aplikace od uživatelského rozhraní. Tento přístup výrazně zlepšil čitelnost kódu, jeho udržovatelnost a možnost dalšího rozšiřování. Rozhraní aplikace byla navržena s důrazem na jednoduchost a konzistenci, aby byla orientace pro uživatele co nejintuitivnější.

Velký důraz byl kladen na funkce, které zvyšují uživatelský komfort – možnost filtrování podle štítků, rychlý přístup k navigaci přes Google Maps, přehledné zobrazení průměrného hodnocení a intuitivní správa oblíbených položek. Zároveň byl kladen důraz na to, aby byla aplikace dostatečně flexibilní pro budoucí rozšíření o nové typy míst či funkce (např. jednotné zpracování dat).

Během realizace bylo ověřeno, že kombinace cloudových a lokálních úložišť přináší řadu výhod. Lokální databáze zajišťuje rychlou odezvu a offline dostupnost oblíbených míst, zatímco cloudové řešení umožňuje sdílení a synchronizaci uživatelských dat napříč zařízeními.

Z pohledu dalšího rozvoje je potenciál v integraci pokročilých vyhledávacích a doporučovacích algoritmů, implementaci personalizovaných návrhů na základě uživatelského chování nebo rozšíření aplikace o vícejazyčnou podporu pro turisty. Dalším možným směrem je propojení s dalšími datovými zdroji, které by mohly přinést aktuální informace například o otevírací době, akcích či sezónních nabídkách podniků.

Práce na této aplikaci přinesla cenné zkušenosti nejen s vývojem v .NET MAUI, a návrhem architektury, integrací různých technologií a optimalizací uživatelského zážitku. Výsledkem je funkční a připravená aplikace, která může sloužit jako praktický průvodce po městě Pardubice a kterou lze v budoucnu dále rozvíjet.

## Literatura

- [1] ColorWhistle. *Travel Mobile App Features*. [online]. (bez data) [cit. 2025-08-16]. Dostupné z: <https://colorwhistle.com/travel-mobile-app-features/>
- [2] Trustpilot. *Trustpilot*. [online]. (bez data) [cit. 2025-08-16]. Dostupné z: <https://www.trustpilot.com/>
- [3] Google. *Google Maps*. [online]. (bez data) [cit. 2025-08-16]. Dostupné z: <https://maps.google.com/>
- [4] Tripadvisor. *Tripadvisor*. [online]. (bez data) [cit. 2025-08-16]. Dostupné z: <https://www.tripadvisor.com/>
- [5] Seznam. *Mapy.cz*. [online]. (bez data) [cit. 2025-08-16]. Dostupné z: <https://mapy.cz/>
- [6] Yelp. *Yelp*. [online]. (bez data) [cit. 2025-08-16]. Dostupné z: <https://www.yelp.com/>
- [7] SmartGuide. *SmartGuide*. [online]. (bez data) [cit. 2025-08-16]. Dostupné z: <https://smart-guide.org/>
- [8] Sygic. *Sygic Travel*. [online]. (bez data) [cit. 2025-08-16]. Dostupné z: <https://travel.sygic.com/>
- [9] Microsoft. *Prohlídka jazyka C#*. [online] 2024 [cit. 2024-04-17]. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/>
- [10] Microsoft. *Co je .NET MAUI?*. [online] 2024 [cit. 2024-04-17]. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/maui/what-is-maui?view=net-maui-8.0>
- [11] SQLite. *SQLite*. [online]. (bez data [cit. 2025-08-16]. Dostupné z: <https://www.sqlite.org/>
- [12] BCrypt.Net. *BCrypt.Net*. [online]. (bez data) [cit. 2025-08-16]. Dostupné z: <https://github.com/BcryptNet/bcrypt.net>
- [13] Google for Developers. *Firebase pricing plans*. [online] 2024 [cit. 2024-05-01]. Dostupné z: <https://firebase.google.com/docs/projects/billing/firebase-pricing-plans>

- [14] Microsoft. *XAML*. [online] 2024 [cit. 2024-04-28]. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/maui/xaml/?view=net-maui-8.0>
- [15] Microsoft. Model-View-ViewModel (MVVM). [online] 2023 [cit. 2024-04-28]. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/architecture/maui/mvvm>
- [16] Microsoft. Co je Visual Studio?. [online] 2023 [cit. 2024-04-17]. Dostupné z: <https://learn.microsoft.com/cs-cz/visualstudio/get-started/visual-studio-ide?view=vs2022>