



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## **NAPORCHÁZKU.CZ - WEBOVÁ SLUŽBA PRO POZNÁVÁNÍ OKOLÍ**

NAPROCHÁZKU.CZ - WEB SERVICE FOR NEIGHBOUR WALKS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MARTIN KLUSKA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. IGOR SZÖKE, Ph.D.**

BRNO 2016

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

**Zadání bakalářské práce**

Řešitel: **Kluska Martin**

Obor: Informační technologie

Téma: **Naprocházkuzk.cz - webová služba pro poznávání okolí**  
**Naprocházkuzk.cz - Web Service for Neighbour Walks**

Kategorie: Web

**Pokyny:**

1. Zvolte si a nastudujte framework pro vývoj webových aplikací.
2. Navrhněte webovou aplikaci, která bude sloužit pro vyhledávání, sdílení a ukládání krátkých procházek po blízkém okolí. Cílem je nabídnout uživateli tipy na krátké výlety (do několika hodin) na základě různých kritérií (průjezdnost s kočárkem, dětské hřiště po cestě, snadné parkování, ...). Součástí webové aplikace by mělo být i API, které bude sloužit pro komunikaci s mobilní aplikací.
3. Implementujte a otestujte webovou aplikaci.
4. Vytvořenou aplikaci zveřejněte. Sledujte chování uživatelů. Reflektujte změny v aplikaci podle chování uživatelů.
5. Zhodnoťte dosažené výsledky a navrhněte směry dalšího vývoje.
6. Vytvořte A2 plakátek a 20-30 vteřinové video o vašem projektu.

**Literatura:**

- Podle pokynů školitele

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a část bodu 3 a 4 ze zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

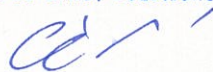
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szöke Igor, Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2



---

doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Tato bakalářská práce se zabývá vytvářením nové verze webového portálu pro sdílení tipů na krátké výlety v okolí. Webový portál je napsán v jazyce PHP za použití frameworku Nette. Cílem webové aplikace je vytvořit přehledné rozhraní pro zadávání trasy a doplnit jej o automatické získávání informací o okolí trasy z veřejně dostupných databází.

## Abstract

In this bachelor thesis we are going to describe the process of creation of a new web portal for sharing of tips for walks. The website is written in PHP using Nette framework. The goal of this application is to create user friendly interface for adding new walks and upgrade it with automatic retrieval of information about the walk surroundings from publicly accessible databases.

## Klíčová slova

naprochazku.cz, Procházka, PHP, Nette framework, API, Google, Mapy, Doctrine 2, Geolokace, Chování uživatelů

## Keywords

naprochazku.cz, Walk, PHP, Nette framework, API, Google, Maps, Doctrine 2, Geolocation, User behavior

## Citace

KLUSKA, Martin. *NaPorcházku.cz - Webová služba pro poznávání okolí*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Szöke Igor.

# NaPorcházku.cz - Webová služba pro poznávání okolí

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szökeho PhD. Všechnu literaturu a zdroje, z kterých jsem čerpal, cituji v práci s uvedením úplného odkazu na příslušný zdroj.

.....

Martin Kluska  
18. května 2016

## Poděkování

Chtěl bych poděkovat svému vedoucímu diplomové práce, Ing. Igoru Szökemu, Ph.D, za odbornou pomoc, cenné rady a profesionální přístup při tvorbě práce.

© Martin Kluska, 2016.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Aktuální stav</b>	<b>4</b>
2.1 Úvodní strana . . . . .	4
2.2 Detail procházky . . . . .	6
2.3 Zadávání procházky . . . . .	6
2.4 Mobilní aplikace . . . . .	7
2.5 Návštěvnost . . . . .	7
2.6 Zhodnocení původní verze . . . . .	8
<b>3 Návrh aplikace</b>	<b>10</b>
3.1 Požadavky . . . . .	10
3.2 Vzhled stránky . . . . .	11
3.2.1 Hlavní menu . . . . .	11
3.2.2 Vyhledávání . . . . .	12
3.2.3 Mapa . . . . .	13
3.2.4 Seznam procházek . . . . .	13
3.3 Uživatelé . . . . .	13
3.3.1 Zjišťování a ukládání polohy uživatele . . . . .	14
3.4 Procházka . . . . .	14
3.4.1 Body zájmu . . . . .	14
3.4.2 Výškový profil trasy . . . . .	15
3.5 Mapové podklady . . . . .	15
3.5.1 Požadavky . . . . .	15
3.5.2 Mapy.cz . . . . .	15
3.5.3 Google Maps . . . . .	16
3.5.4 Závěr . . . . .	16
3.6 Výpočet vzdálenosti uživatele od procházky . . . . .	16
3.7 API . . . . .	17
3.8 Stažení existujících procházek . . . . .	17
<b>4 Implementace</b>	<b>19</b>
4.1 Použité technologie . . . . .	19
4.1.1 Nette framework . . . . .	19
4.1.2 Bootstrap . . . . .	19
4.1.3 SASS . . . . .	19
4.1.4 Gulp . . . . .	20
4.2 Google maps API . . . . .	20

4.2.1	API klíč	20
4.2.2	Zobrazení procházky	20
4.2.3	Places service	22
4.2.4	Elevation service	23
4.3	Geocoding API	24
4.3.1	Geokódování	24
4.3.2	Reverzní geokódování	24
4.4	Databázová vrstva	25
4.4.1	Skládání dotazů	26
4.4.2	Query object	26
4.5	Databáze	26
4.6	Struktura aplikace	28
4.7	Zjišťování aktuální polohy uživatele	29
4.7.1	HTML5 geolocation API	29
4.7.2	Geolokace podle IP adresy	29
4.8	Zabezpečení proti robotům	30
4.8.1	Ověřování účtu pomocí emailu	30
4.8.2	CAPTCHA	30
4.8.3	Skryté pole	30
4.8.4	Odesílání časového razítka	31
4.8.5	Závěr	31
<b>5</b>	<b>Zveřejnění aplikace a sledování uživatelů</b>	<b>32</b>
5.1	Zveřejnění aplikace	32
5.2	Přímé sledování	32
5.3	Nepřímé sledování	33
5.3.1	Google Analytics	33
5.3.2	Heatmap	34
5.3.3	Návštěvnost	34
<b>6</b>	<b>Závěr</b>	<b>36</b>
6.1	Navrhovaná rozšíření	36
	<b>Literatura</b>	<b>37</b>
	<b>A Obsah CD</b>	<b>38</b>

# Kapitola 1

## Úvod

Cílem této bakalářské práce je provést novou implementaci stávající webové služby na prochazku.cz. Jedná se o webovou aplikaci poskytující služby pro sdílení tipů na krátké procházky z okolí. Trasa procházky je zaznamenána ve formě bodů na mapě. Kromě samotné trasy se u procházky zaznamenávají její vlastnosti, body zájmu z jejího okolí jako jsou přílehlá parkoviště, dostupnost autobusové dopravy nebo třeba i přílehlé lavičky, na kterých si můžeme odpočinout. Hlavním důvodem k vytvoření nové implementace je vylepšení uživatelské přívětivosti zejména při zadávání nových procházek. Kromě vylepšení samotného zadávání tras do mapy bude aplikace napojena na API, které umožňuje získávat dodatečné informace o jejím okolí.

V kapitole 2 se zaměříme na původní verzi webové stránky. Rozebereme její současnou podobu a ukážeme si, které věci jsou zde vyřešeny dobře a které by naopak měly být upraveny. Zároveň navrhneme, jak dané problémy co nejlépe vyřešit. Návrh nové verze aplikace je podrobně rozebrán v kapitole 3. Nejdříve zopakujeme požadavky, které musí výsledná aplikace splňovat. Poté se pustíme do návrhu vzhledu nové verze a nakonec vybereme poskytovatele mapových podkladů. V kapitole 4 se zaměříme na implementaci. Budou zde prakticky ukázána práce s API službami, které budou v rámci aplikace využívány. Dále bude popsáno schéma databáze a spolu s tím bude představena práce s databází, která je řešena pomocí knihovny Doctrine 2<sup>1</sup>. V kapitole 5 otestujeme aplikaci na reálných uživateli. V závěru zhodnotíme dosažené výsledky a naznačíme směr možného budoucího vývoje.

---

<sup>1</sup><http://www.doctrine-project.org/>

## Kapitola 2

# Aktuální stav

Pro účel sdílení tras procházek byla v rámci bakalářské práce Jiřího E. Špačka [5] vytvořena aplikace `naprochazku.cz`. Systém byl implementován jako webová aplikace napsaná v jazyce PHP za použití frameworku Nette<sup>1</sup>.

Aplikace slouží především ke sdílení tipů na krátké výlety. Umožňuje tedy zaznamenávat trasy, přidávat k nim popisky a body zájmu v okolí trasy. Dále nabízí vyhledávání již zadaných tras a jejich sdílení pomocí sociálních sítí.

K webové aplikaci byla v rámci bakalářské práce Romana Dočkala [3] vytvořena mobilní aplikace, která využívá několika metod API rozhraní. Nová implementace bude poskytovat stejné API, aby byla zajištěna funkčnost mobilní aplikace.

### 2.1 Úvodní strana

Na obrázku 2.1 je vyfocena aktuální verze úvodní strany. Její obsah můžeme rozdělit do několika komponent:

- **Hlavní menu**

Poskytuje základní navigaci po celém webu. Nepřihlášeným uživatelům se zde zobrazují odkazy na seznam všech procházek a seznam procházek dle regionu. Pro přihlášené uživatele jsou tyto odkazy doplněny o odkaz na seznam mých procházek a odkaz na vytvoření nové procházky.

- **Mapa**

Prvek zabírající největší plochu stránky obsahuje ikony znázorňující všechny dostupné procházky a jednu ikonu znázorňující aktuální polohu uživatele. Po kliknutí na ikonu procházky se zobrazí dialog s krátkým popisem a odkazem na její detail.

- **Tipy na procházky**

Tento prvek nabízí uživateli procházky, které by se měly nacházet v jeho okolí. V případě, že uživatel nemá uloženou svoji polohu, je tento blok prázdný.

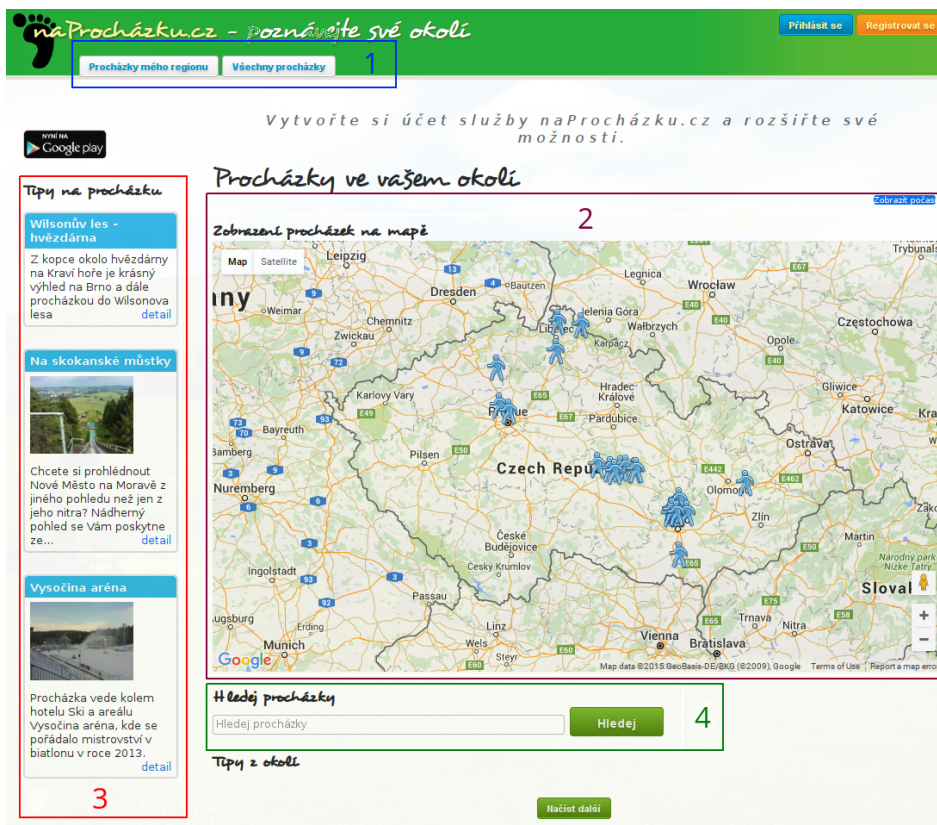
- **Vyhledávání**

Umožňuje vyhledávání v existujících procházkách. Je řešeno jako jedno vstupní pole, které vyhledává jak v názvu a popisu procházky tak i v jejich bodech zájmu a vlastnostech.

---

<sup>1</sup><https://nette.org/>

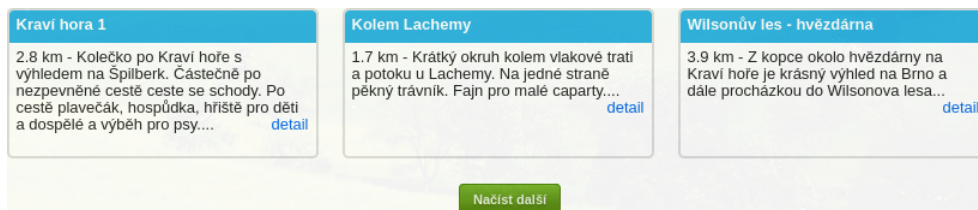




Obrázek 2.1: Úvodní strana  
 1 - Hlavní menu, 2 - Mapa, 3 - Tipy na procházku, 4 - Vyhledávání

Pokud uživatel povolí sdílení své aktuální polohy, tak se na úvodní stránce ještě zobrazí blok se seznamem procházek viz obrázek 2.2. Každá procházka je v seznamu zobrazena ve vlastním bloku. Blok kromě názvu procházky obsahuje ještě část textového popisu procházky.

Seznam ve výchozím stavu zobrazuje šest tipů na procházky. Pomocí tlačítka „Načíst další“ je možné načíst tipy na další procházky, které jsou přidány na konec seznamu. Při použití metody, která pouze přidává nové bloky s procházkami do stránky může uživatel ztratit přehled, kde na stránce se nachází.



Obrázek 2.2: Seznam procházek

## 2.2 Detail procházky

Na obrázku 2.3 je zobrazen textový popis detailu procházky. Je zde zobrazen kraj, do kterého procházka náleží, její oblíbenost, délka trasy a přibližná doba potřebná k projití trasy a autor procházky. V tabulce s detaily můžeme vidět seznam vlastností trasy, dostupných bodů zájmu a zadaný textový popis.



Obrázek 2.3: Detail procházky

Pod textovým popisem procházky je mapa, na které je znázorněna její trasa, která je doplněna o vložené body zájmu.



Obrázek 2.4: Detail trasy

## 2.3 Zadávání procházky

Každý z registrovaných uživatelů má možnost přidat novou procházku do systému. Přidávání je rozděleno na dvě fáze. V první fázi uživatel vyplní základní informace jako jsou název, popis, kraj a vlastnosti. Ve druhé fázi zadává trasu a přidává body zájmu.

Rozdělení na více fází není příliš uživatelsky přívětivé, protože nutíme uživatele při zadávání procházky postupovat v pevně daném pořadí. Vhodnější bude, když zadávání celé procházky bude v jednom kroku. Uživatel si tak může sám rozhodnout, které hodnoty chce vyplnit dříve a které později.

Druhá fáze zadávání procházky je dále rozdělena do tří módů - zadávání trasy, přidávání bodů zájmu a úprava zadaných údajů. V módu zadávání trasy klikáním na mapu vytváříme jednotlivé body, které jsou propojeny čarou. Zadávání trasy ukončíme tím, že spojíme její

poslední bod s jejím prvním bodem. To tedy znamená, že systém umožňuje zadávat pouze kruhové trasy. Další nevýhodou rozdělení na několik módů je, že nemůžeme při zadávání trasy procházky průběžně přidávat nové body zájmu, ale musíme je přidat až po dokončení zadávání. V průběhu fáze zadávání trasy nemůžeme upravovat pozici již zadaných bodů, takže pokud nějaký bod vložíme na špatné místo, můžeme jej přesunout až po dokončení zadávání v módu úpravy. Systém zadávání tras dále neumožňuje odstraňovat jednotlivé body, ale pouze celou trasu.

V módu přidávání bodů zájmu přidáváme kliknutím na mapu nový bod zájmu. Ihned po přidání se zobrazí modální dialog, ve kterém můžeme zvolit typ bodu zájmu a přidat textový popis. Obdobně jako v módu zadávání trasy zde také nemůžeme upravovat již vložené hodnoty.

## 2.4 Mobilní aplikace

K webové stránce byla vytvořena také mobilní aplikace pro operační systém Android. Ta umožňuje uživatelům zobrazovat procházky, které byly zadané do systému. Podstatnou funkcí je zobrazení detailu procházky, která obsahuje mapový podklad se znázorněnou trasou a vyznačenými body zájmu v jejím okolí.

Jednou z nejdůležitějších součástí aplikace je možnost vytváření nové procházky. Využívá k tomu GPS modulu, který dnes obsahuje většina mobilních telefonů. Po stisknutí tlačítka „Začít ukládat trasu“ začne GPS modul snímat aktuální polohu, která je dále ukládána. Kdykoliv v průběhu zaznamenávání trasy může uživatel přidávat body zájmu, ale bod zájmu lze přiřadit pouze k místu, na kterém se uživatel aktuálně nachází. Až uživatel dojde na konec trasy procházky, nabídne mu aplikace tři možnosti, které může s nahranou procházkou udělat:

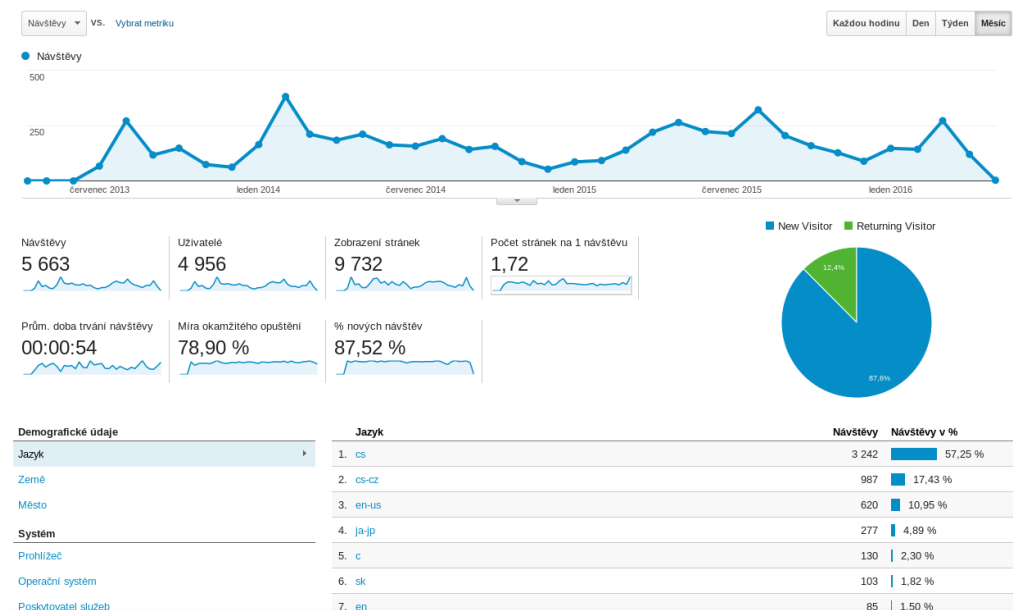
- **Odeslat ihned**  
Procházka bude nahrána na webový portál.
- **Uložit, odeslat později**  
Uloží aktuálně zaznamenanou trasu do lokální databáze a umožní uživateli nahrát procházku na webový portál později.
- **Neukládat**  
Odstraní aktuálně zaznamenanou trasu z paměti.

## 2.5 Návštěvnost

Na původní verzi webové stránky byla nasazena služba Analytics<sup>2</sup>, takže máme přehled o návštěvách uživatelů. Webová stránka neměla žádnou formu reklamní kampaně, proto se nedostala do podvědomí většího množství uživatelů. Zajímavým místem pro propagaci této stránky by mohla být turistická fóra.

---

<sup>2</sup><https://analytics.google.com>



Obrázek 2.5: Statistiky návštěv původní verze webové aplikace.

## 2.6 Zhodnocení původní verze

Na původní verzi aplikace se mi velice líbí mapa na úvodní straně, která zobrazuje ikony procházek a aktuální polohu uživatel. Umožňuje tak získat rychlý přehled o možných procházkách v jeho okolí.

Pod mapou se nachází vyhledávání. Většina uživatelů si však jako první všimne velké mapy a vyhledávání přehlédne. Vyhledávání je důležitou součástí webu, proto by mělo být umístěno nad mapou, kde by bylo více viditelné. Některé uživatele nemusí vyhledávání zajímat, proto by jim mělo být umožněno celý filtr skrýt. Vyhledávání je řešeno jedním vstupním polem. Text zadaný do pole je vyhledáván ve všech hodnotách, které jsou přiřazeny k procházce. To přináší určité problémy, kdy například vyhledávání nedokáže rozlišit mezi slovy park a parkoviště. V nové verzi bude proto filtr rozšířen na vyhledávání podle jednotlivých vlastností.

Uživatelům může činit problém zadávání nových procházek, které je rozdělené do několika fází. Uživatel tak musí při zadávání nové procházky postupovat v pevně daném pořadí a nemůže si zvolit, jaké hodnoty bude vyplňovat dříve. Pro větší uživatelskou přívětivost budou tyto fáze sjednoceny do jedné.

Ve fázi zadávání trasy chybí možnost odstranění jednotlivých bodů, systém umožňuje pouze odstranění celé trasy najednou. Dalším nedostatkem je, že nemůžeme kombinovat zadávání trasy a bodů zájmu.

V tabulce 2.6 jsou popsány některé vlastnosti systému, které budou v rámci nové implementace upraveny.

<b>Funkce</b>	<b>Původní verze</b>	<b>Nová verze</b>
Typ trasy procházky	– Kruhová	– Kruhová – z bodu A do bodu B
Zadávaní bodů zájmu	– ručně	– ručně – získávání bodů zájmu z dostupných API
Zadávaní hodnot do mapy	tři módy – zadávání trasy – zadávání bodů zájmu – úprava již zadaných hodnot	– módy budou sjednoceny
Vyhledávání	– jedno textové pole	– sada několika filtrů pro každou vlastnost zvlášť
Vkládání nové procházky	– rozděleno do dvou fází 1. zadávání textových informací 2. zadávání trasy a bodů zájmu	– vše se zadává v jedné fázi
Zjišťování polohy uživatele	– HTML 5 geolocation API	– HTML 5 geolocation API – IP adresa – manuální zadání

Tabulka 2.1: Shrnutí vlastností aplikace.

## Kapitola 3

# Návrh aplikace

V této kapitole se seznámíme s návrhem nové aplikace, ve které se pokusíme odstranit většinu nedostatků původní verze. Návrh aplikace vychází z již existující implementace. V první řadě tedy musí uživatelé nabídnout stejné funkce, jaké poskytovala původní verze. Hlavním cílem nové implementace je co nejvíce usnadnit proces zadávání nové procházky a vylepšit vyhledávání již existujících procházek.

### 3.1 Požadavky

V rámci analýzy aplikace vzniklo několik požadavků, které má služba splňovat. Nová verze by měla nabídnout stejné funkce, které nabízí její původní implementace. Musí tedy uživatelé umožnit:

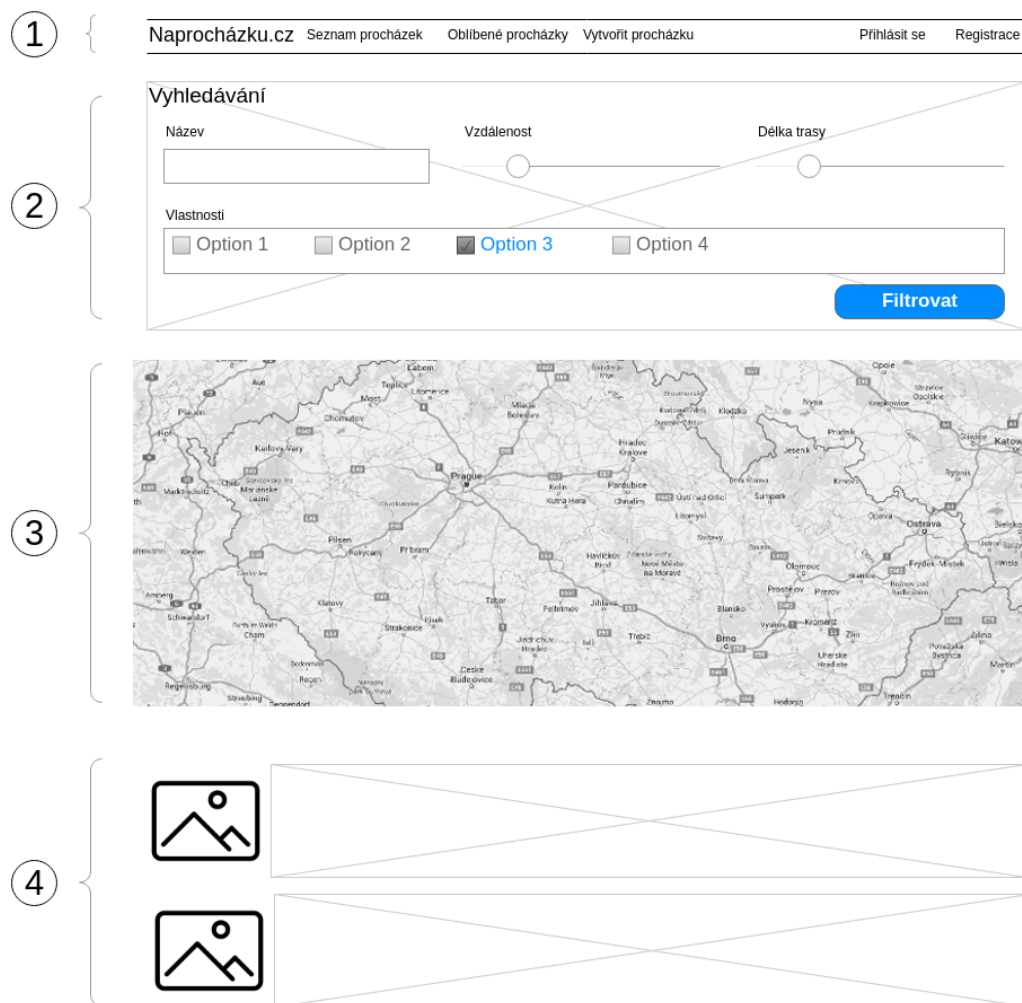
- Zobrazit seznam procházek
- Zobrazit detail procházky
- Vyhledávat procházky
- Vytvořit procházku
- Upravit procházku
  - Upravit trasu procházky
  - Upravit body zájmu procházky
  - Upravit popis procházky
  - Upravit vlastnosti procházky
- Zjišťování aktuální polohy uživatele
- Nahrát fotografii k procházce
- Odstranit procházku

Hlavním cílem nové implementace je poskytnout lepší rozhraní z hlediska uživatelské přívětivosti. Nejvíce se zaměříme na úpravu přidávání trasy, protože chceme, aby uživatelé v první fázi existence webové stránky přidali co nejvíce nových procházek. Dále bude

kompletně přepracován systém vyhledávání procházek. Cílem je vytvořit filtr, který umožňuje vyhledávat dle jednotlivých atributů. Díky tomu si uživatel bude moci snadněji najít procházku, která odpovídá jeho představám. Důležitou novou funkcí je propojení aplikace se službami třetích stran, které poskytují informace o bodech zájmu v okolí procházky. Tyto služby budeme využívat, abychom získali co nejvíce informací k procházce a nenutili uživatele zadávat vše ručně.

## 3.2 Vzhled stránky

Ve fázi návrhu stránky vznikl wireframe úvodní stránky 3.1. Webová stránka se skládá z několika komponent, které budou dále popsány.



Obrázek 3.1: Wireframe úvodní stránky. 1 - Hlavní menu, 2 - Vyhledávání, 3 - Mapa, 4 - Seznam procházek

### 3.2.1 Hlavní menu

Základní navigaci na stránce poskytuje hlavní menu, které dále můžeme rozdělit na dva samostatné celky. Levá část obsahuje odkazy na další stránky webu. Nepřihlášení uživatelé

zde budou mít pouze odkaz na seznam procházek. Pokud je uživatel přihlášen, uvidí zde navíc seznam oblíbených procházek a odkaz na vytvoření nové procházky.

Obsah pravé části je rovněž závislý na tom, zda je uživatel přihlášen. Nepřihlášení uživatelé zde mají odkazy na přihlášení a na vytvoření nového účtu. Přihlášeným uživatelům se zde vypíše jejich jméno, odkazy na nastavení účtu a odhlášení. Pokud má uživatel administrátorská oprávnění, bude zde mít ještě odkaz do administrátorské sekce.

### 3.2.2 Vyhledávání

Vyhledávání procházek je důležitou součástí aplikace. Musí poskytovat jednoduchou možnost k filtrování dle jednotlivých atributů procházky. Filtr bude nabízet vyhledávání podle textu, který se vyskytuje v názvu nebo popisu procházky. Dále bude nabízet vyhledávání podle vzdálenosti procházky od aktuální polohy uživatele, její celkové délce a podle vlastností procházky a jejich bodů zájmu.

Pro každou z filtrovacích podmínek bude vhodné použít jinou vizuální komponentu. Tyto komponenty jsou přesněji popsány níže.

- **Název nebo popis procházky**

Pro vyhledání v názvu nebo popisu procházky bude sloužit obyčejné textové pole.

- **Vzdálenost procházky od aktuální polohy uživatele**

Komponenta pro vyhledávání dle vzdálenosti je řešena jako posuvník, kterým nastavíme maximální hodnotu vzdálenosti.



Obrázek 3.2: Komponenta pro vyhledávání dle vzdálenosti

- **Délka procházky**

Pro vyhledávání podle délky procházky je opět použit posuvník. V tomto případě ale umožňuje nastavovat minimální i maximální hodnotu délky procházky.



Obrázek 3.3: Komponenta pro vyhledávání dle délky trasy

- **Vlastnosti**

Seznam zaškrtnutých políček se třemi stavy (nezáleží, obsahuje, neobsahuje)

#### Vlastnosti



Obrázek 3.4: Komponenta pro vyhledávání dle vlastností

Celý filtr je ve výchozím stavu zobrazený, ale je ho možné pomocí tlačítka skrýt. Webová aplikace si pamatuje, zda byl při poslední uživatelské návštěvě skrýty nebo zobrazený.



### 3.2.3 Mapa

Hlavním prvkem stránky je mapa, na které jsou zobrazeny všechny procházky odpovídající zvoleným filtrovacím podmínkám. Mapa ve výchozím stavu zobrazuje celou Českou republiku, ale pokud má uživatel zadanou svoji aktuální polohu, mapa na ni bude vycentrována a přiblížena.

Každá procházka je na mapě zobrazena formou ikonky. Po kliknutí na ikonku se zobrazí dialog s textovým popisem procházky a odkazem na její detail. Pokud jsou k procházce nahrány nějaké fotografie, budou rovněž zobrazeny v dialogu vedle textového popisu.

### 3.2.4 Seznam procházek

Pod mapou se dále nachází seznam procházek, který slouží k zobrazení dalších detailů k procházce, které není možné zobrazit v dialogu v mapě. Ke každé procházce je zde uveden její název, autor a textový popis. Pokud byla k procházce nahrána fotografie, bude zde také zobrazena.

Protože seznam procházek může být dlouhý, je zde doplněno stránkování. Na jednu stránku se zobrazuje až 5 procházek. Přejít na další stranu je řešen pomocí AJAXu.

## 3.3 Uživatelé

V rámci naší aplikace se můžeme setkat s několika skupinami uživatelů. Každá z těchto skupin uživatelů má svoji vlastní roli, které jsou přiřazeny určitá oprávnění. Tato oprávnění jsou popsány v následujícím seznamu.

- **Návštěvník**

Jedná se o běžného nepřihlášeného uživatele, který nemá vytvořený účet. Ze všech skupin uživatelů má nejmenší možná oprávnění. Mezi jeho práva patří:

- Zobrazit seznam procházek
- Vyhledávat procházky pomocí filtru
- Aktualizovat svoji aktuální polohu

- **Registrovaný uživatel**

Uživatel, který má již vytvořený svůj vlastní účet, automaticky získává veškerá oprávnění neregistrovaného návštěvníka, ale jeho práva jsou navíc doplněna o:

- Přidání nové procházky
- Vložení procházky do seznamu oblíbených procházek
- Upravovat své přidané procházky

- **Administrátor**

Je nejvyšší stupeň oprávnění. Kromě práv registrovaného uživatele má navíc možnost spravovat všechny procházky, spravovat uživatele.

### 3.3.1 Zjišťování a ukládání polohy uživatele

Protože aplikace pracuje s polohou uživatele, musíme ji nejdříve nějakým způsobem zjistit. K tomu můžeme využít geolokačního rozhraní prohlížeče, nicméně musíme také počítat s možnostmi, že uživatel používá prohlížeč, který toto rozhraní nepodporuje, případně ho z nějakého důvodu nepovolí. Pro tyto případy můžeme využít zjišťování polohy podle IP adresy.

V aplikaci chceme uživateli nabídnout možnost vyhledávání i v okolí jiného místa než na kterém se právě nachází. V tomto případě nelze použít žádnou automatickou detekci polohy. Musíme proto umožnit zadávat polohu i nějakým jiným způsobem. Do aplikace proto bude přidán prvek, který umožňuje převést textový popis na polohu. K převodu jsou využity API služby, které poskytuje např. Google.

Jelikož chceme, aby měli přístup k vyhledávání podle polohy jak přihlášení, tak nepřihlášení uživatelé, musíme vyřešit, kam ukládat informaci o jejich poloze. U přihlášených uživatelů se nabízí možnost ukládání této informace jako jeden z atributů v databázi. Ukládání této informace do databáze i pro nepřihlášené uživatele není vhodné, protože jako jedinou informaci o identitě uživatele je identifikace jeho sezení, která se ale mění při ukončení prohlížeče. Z toho důvodu by se nám databáze mohla zaplnit již nepoužívanými údaji. Pro nepřihlášené uživatele tedy bude lepší využít ukládání aktuální polohy do session.

Aby byla práce se dvěma odlišnými úložišti v dalších částech aplikace příjemnější, vznikne nová třída zajišťující detekci nejvhodnějšího úložiště a práci s ním.

## 3.4 Procházka

Procházka je v našem systému složena z jejího textového popisu, množiny bodů, které definují její trasu, množiny vlastností a množiny bodů zájmu, které se vyskytují v okolí trasy. Jednotlivé body trasy jsou uloženy ve formátu GPS souřadnic a je podle nich vykreslena cesta na mapovém podkladu. Body zájmů v okolí jsou uloženy podobně jako body trasy, tedy ve formě GPS souřadnic, které jsou v tomto případě doplněny o typ bodu zájmu a o jeho textový popis.

V původní implementaci je jako jeden z atributů pro procházku kraj, do kterého procházka náleží. Tento údaj slouží především pro vyhledávání, nicméně toto řešení s sebou přináší také nedostatky. Pokud chce uživatel vyhledat procházky ve svém okolí a nachází na hranici několika krajů, musí vyhledávat pro každý kraj samostatně. Tento problém lze vyřešit tím, že se u každé trasy bude navíc ukládat její střed ve formě GPS souřadnic. Díky tomu můžeme vyhledávat procházky v nějaké vzdálenosti od určité pozice (typicky od aktuální polohy uživatele).

Každá procházka může obsahovat několik bodů zájmu ve svém okolí. Jedná se například o přilehlé parkoviště, autobusovou zastávku, park a mnoho dalších. Tyto atributy se jednak zobrazují na mapě a zároveň je díky nim možno vyhledávat. Dalšími atributy trasy jsou hodnoty, které se nezobrazují na mapě, ale pouze nějak popisují trasu samotnou. Mezi tyto hodnoty patří například celková délka trasy nebo výškový profil trasy.

### 3.4.1 Body zájmu

Jednou z důležitých hodnot procházky jsou body zájmu v okolí její trasy. Body zájmu slouží k detailnějšímu popisu okolí, umožňují tak uživateli získat větší přehled. Mohou obsahovat informace o odpočinkových zónách v blízkém okolí (např. lavička) nebo slouží

k popisu okolní krajiny (např. výhled z kopce). Dalším důležitým atributem jsou informace o přilehlých parkovištích nebo autobusových zastávkách. Tyto atributy slouží jako informace o dosažitelnosti trasy.

### 3.4.2 Výškový profil trasy

U trasy budeme rovněž zobrazovat nadmořskou výšku podél trasy. Výškový profil bude zobrazen ve formě grafu. Uživatel díky tomu uvidí, zda je trasa kopcovitá nebo spíše rovinná. Může si proto vybrat procházky, které více odpovídají jeho fyzickým možnostem.

## 3.5 Mapové podklady

Trasa procházky bude zobrazena na mapě, musíme proto zvolit vhodného poskytovatele mapových podkladů. Výběr poskytovatelů jsem zúžil na dvě u nás nejznámější služby - Mapy.cz a Google Maps.

### 3.5.1 Požadavky

Než začneme porovnávat jednotlivé poskytovatele, musíme si stanovit požadavky, podle kterých se rozhodneme pro jednu ze služeb.

Při výběru jsem se zaměřil na několik parametrů, které jsem porovnával:

- Kvalita mapových podkladů
- Oficiální dokumentace k API
- Neoficiální dokumentace (různé návody, články, ukázky) k API
- Další služby poskytované API
- Rychlost načítání

### 3.5.2 Mapy.cz

Jedná se o nejznámějšího poskytovatele mapových služeb v České republice. Pro Českou republiku a Slovenskou republiku nabízí několik druhů map - základní, turistickou, zeměpisnou, leteckou a další. Pro ostatní státy nabízí pouze základní mapu, u které přebírá mapové podklady ze služby OpenStreetMap<sup>1</sup>.

Pro práci s API existuje oficiální javascriptová knihovna<sup>2</sup>, která umožňuje základní operace s mapou. Obsahuje funkce na kreslení čar, přidávání bodů zájmu a zobrazování modálních dialogů.

Velkou výhodou tohoto poskytovatele je, že nemá nastaveny žádné limity pro používání. Naproti tomu je však nevýhoda slabší oficiální dokumentace.

---

<sup>1</sup><http://www.cnews.cz/seznam-ma-podrobnejsi-mapu-evropy-vyuzil-podklady-z-openstreetmap>

<sup>2</sup><https://api.mapy.cz/>

### 3.5.3 Google Maps

Jeden z největších poskytovatelů map z celého světa. Nabízí dva druhy map - základní a leteckou. Kromě základních map poskytuje v rámci služby StreetView<sup>3</sup> nafocené 3D snímky většiny cest v České republice i v jiných státech.

Pro práci s API existuje oficiální javascriptová knihovna<sup>4</sup>, která kromě základních operací s mapou umožňuje i využívání dalších API od společnosti Google. Umožňuje tak jednoduché propojení se službami, které budeme v rámci aplikace využívat.

V rámci oficiální dokumentace je poskytován detailní systém ukázek jednotlivých komponent, který usnadní jejich použití. V rámci dokumentace je k dispozici i referenční příručka k jednotlivým třídám z oficiální knihovny.

Na využívání map je nastaven limit na 25000 zobrazení za den. Tento limit je však pro naši aplikaci dostatečně vysoký.

### 3.5.4 Závěr

Z hlediska kvality mapových podkladů pro Českou republiku a Slovenskou republiku je na tom lépe služba Mapy.cz, která nabízí i kvalitní turistické mapy, na kterých jsou mimo jiné zaznamenány polní cesty a další zajímavé cíle, ale v celosvětovém měřítku poskytují větší detaily mapy od společnosti Google. V současné době jsou Mapy.cz postupně doplňovány o 3D panoramatické snímky, ale zatím jsou dostupné pouze v několika větších městech<sup>5</sup>. Z tohoto hlediska je na tom mnohem lépe služba Maps Google, která ve *Street view* nabízí 3D pohled na většinu míst v České republice i v okolních státech. Díky využití služby Street view si může uživatel „projít“ část trasy přímo ve svém prohlížeči, což mu umožní se lépe rozhodnout, zda tuto procházku absolvuje i ve skutečnosti.

Dalším důvodem pro zvolení Maps Google je velice pokročilá úroveň oficiální knihovny. Kromě samotného zobrazení mapy umožňuje i využití dalších poskytovaných služeb, které budeme k zobrazování procházky využívat. Pomocí této knihovny můžeme pracovat s Places service<sup>6</sup>, což je služba, která umožňuje získávání informací o bodech zájmu.

V rámci mobilní aplikace jsou již využívány mapové podklady od firmy Google, je proto vhodné použít stejného poskytovatele i ve webové aplikaci. Pokud by byl použit jiný poskytovatel, mohly by obě aplikace působit nejednotným dojmem. Mapy.cz sice poskytují detailnější mapy v rámci České republiky, ale pro ostatní státy poskytuje více detailů Maps Google, takže jejich použití usnadní rozšíření do dalších zemí.

## 3.6 Výpočet vzdálenosti uživatele od procházky

Jelikož chceme uživateli nabídnout informaci o tom, jak jsou od něj jednotlivé procházky vzdálené, musíme tuto vzdálenost nejdříve vypočítat.

Pro výpočet vzdálenosti dvou GPS bodů můžeme použít Haversinův vzorec<sup>7</sup>, který slouží k výpočtu vzdálenosti 2 bodů na kulové ploše. Tato metoda nemusí vracet přesné výsledky, protože počítá s tím, že je Země dokonalá koule, ale pro naše účely je přesnost dostatečná.

<sup>3</sup><https://www.google.com/maps/streetview/>

<sup>4</sup><https://api.mapy.cz/>

<sup>5</sup><http://seznam.seznamblog.cz/post/142625807976/mapycz-spustily-3d-pohled-na-dal%C5%A1%C3%ADch-22-m%C4%9Bst-a>

<sup>6</sup><https://developers.google.com/places/>

<sup>7</sup><http://www.movable-type.co.uk/scripts/latlong.html>

$$\begin{aligned}
a &= \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2\left(\frac{\Delta lon}{2}\right) \\
c &= 2 \cdot \arctg\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \\
d &= R \cdot c
\end{aligned}
\tag{3.1}$$

### 3.7 API

Původní implementace poskytuje API pro přístup k procházkám z mobilní aplikace. Nová verze tedy musí implementovat stejné rozhraní, aby s novým webem mohla spolupracovat stávající mobilní aplikace. Z dokumentu k původní bakalářské práci [5] jsem zjistil, že je třeba implementovat následující funkce:

- Seznam všech procházek
- Hledání v procházkách
- Načtení detailu procházky
- Vytvoření nové procházky
- Přihlášení uživatele
- Seznam všech uživatelských jmen
- Vytvoření nového uživatele
- Seznam všech dostupných tagů
- Seznam tagů pro zadanou procházku
- Uložení poslední známé pozice uživatele

Každý z těchto požadavků má pevně danou svoji základní URL adresu a množinu parametrů funkce. Parametry jednotlivých funkcí jsou kódovány do query stringu požadavku. Naše aplikace tedy tento požadavek zpracuje a odpoví daty ve formátu JSON s kódováním UTF-8.

### 3.8 Stažení existujících procházek

Na původní verzi aplikace je nyní zadáno okolo 50 procházek. Byla by škoda o tato data přijít, proto je potřeba nějakým způsobem data o procházkách získat. Bohužel nemáme přímý přístup k databázi, takže musíme použít nějaký jiný způsob k jejich získání.

Na získání dat nám tedy zbývají dvě možnosti.

- Parsování dat ze struktury webu
- Použití API

Jelikož API poskytuje funkce pro získání veškerých procházek a zobrazení detailu procházky, můžeme využít tyto metody. Nejdříve tedy pomocí funkce pro výpis všech procházek získáme základní informace o každé jednotlivé procházce. Mezi nimi je i identifikátor procházky, který použijeme pro získání detailních informací o procházce, včetně její trasy a bodů zájmů. Tato data vhodným způsobem transformujeme pro naši databázi a uložíme je do ní.

# Kapitola 4

## Implementace

V této kapitole je popsán přístup k vývoji aplikace. Jsou zde popsány technologie, které byly při tvorbě použity. Dále je zde popsána práce s jednotlivými API službami, popsána databáze a celková struktura aplikace.

### 4.1 Použité technologie

V rámci zadání již bylo rozhodnuto, že se bude jednat o webovou aplikaci. To poměrně jasně určuje, jaké jazyky budou pro jeho tvorbu použity. Pro frontend je použita kombinace jazyků HTML, CSS a JavaScript. Pro tvorbu backendu máme větší množství možností, nicméně zdaleka nejpoužívanějším jazykem pro tvorbu webových stránek je PHP, který byl nakonec použit.

#### 4.1.1 Nette framework

Pro zjednodušení implementace aplikace byl použit framework Nette. Jedná se o nejrozšířenější framework v České republice. Je poskytován pod licencí New BSD<sup>1</sup>, lze tedy zdarma použít i v komerčních aplikacích.

#### 4.1.2 Bootstrap

Framework Bootstrap poskytuje základní CSS styly pro webové aplikace. Je poskytován pod licencí MIT<sup>2</sup>, lze tedy zdarma použít i v komerčních aplikacích. Pro použití v aplikaci si můžeme stáhnout buď plnou verzi knihovny a nebo jednotlivé komponenty ve formě SASS<sup>3</sup>. Pokud si stáhneme verzi ve formátu SASS, můžeme si sami zvolit, které komponenty na svém webu použijeme. Díky tomu nemusíme k uživateli přenašet kompletní sadu stylů, ale pouze ty, které skutečně využijeme.

#### 4.1.3 SASS

SASS je tzv. preprocesor pro CSS styly. Do klasických CSS stylů doplňuje některé prvky ze skriptovacích jazyků jako jsou například proměnné, podmínky a cykly. Překladač umožňuje

---

<sup>1</sup><https://nette.org/cs/license>

<sup>2</sup><https://github.com/twbs/bootstrap/blob/master/LICENSE>

<sup>3</sup><https://github.com/twbs/bootstrap-sass>

kromě samotného překladu i další možnosti. Jednou z nejzajímavějších je možnost odstranění nadbytečných bílých znaků ve výsledném CSS souboru. Tím docílíme menší velikosti výsledného souboru.

#### 4.1.4 Gulp

SASS s sebou kromě jiného přináší i nutnost překladu. Po každé změně ve stylech tedy musíme tento překlad provést, aby se projevil na webových stránkách. Abychom nemuseli pokaždé ručně spouštět překladač, můžeme použít Gulp s rozšířením `watch`<sup>4</sup>. Ten sleduje změny v souborech a při detekci nějaké změny provede určenou akci - v našem případě tedy překlad SASS souborů.

## 4.2 Google maps API

V předchozí kapitole jsme vybrali Google maps API jako poskytovatele mapových podkladů. Integrace do webových stránek je snadná, protože Google poskytuje ke svým službám javascriptovou knihovnu<sup>5</sup>, která lze přímo využít na stránkách.

V této kapitole budou dále popsány základní části knihovny, které jsou využívány v rámci naší aplikace.

### 4.2.1 API klíč

Protože na využívání API existují jisté limity, musí se aplikace jednoznačným způsobem identifikovat. V rámci Google API k tomu slouží tzv. API klíč<sup>6</sup>. Je to řetězec dlouhý 39 znaků, který obsahuje malá písmena, velká písmena a číslice.

Pro získání klíče zaregistrujeme naši aplikaci u společnosti Google. V konzoli pro vývojáře<sup>7</sup> povolíme využívání Maps API a tím získáme klíč, který budeme dále využívat u všech požadavků.

Díky použití API klíče je naše aplikace jednoznačně identifikována, proto nám může Google poskytovat statistiky využívání API. V rámci administrace účtu se můžeme podívat například na počet zobrazení mapy na našem webu.

### 4.2.2 Zobrazení procházky

Pro zobrazení procházky na mapě budeme využívat oficiální knihovnu k API Google Maps<sup>8</sup>. V kódu 4.1 je ukázáno zobrazení trasy procházky a bodů zájmu na mapě.

Nejdříve vytvoříme nový objekt třídy `google.maps.Map`, který vytvoří mapu a vloží ji do HTML struktury webové stránky. Poté získáme jednotlivé body trasy a uložíme je do pole `coords`. Využitím tohoto pole vypočítáme souřadnice mezních bodů trasy, které máme uložené v proměnné `bounds`. Díky nim můžeme nastavit střed mapy na střed trasy a nastavit požadované přiblížení. Abychom zobrazili trasu procházky v mapě, vytvoříme novou instanci třídy `google.maps.Polyline`, které předáme souřadnice všech bodů. Nakonec získáme všechny body zájmu, které jsou k procházce přiřazeny a na mapě je zobrazíme jako `google.maps.Marker`. Na obrázku 4.1 vidíme výsledné zobrazení trasy.

---

<sup>4</sup><https://www.npmjs.com/package/gulp-watch>

<sup>5</sup><https://developers.google.com/maps/documentation/javascript/3.exp/reference>

<sup>6</sup><https://developers.google.com/maps/documentation/javascript/get-api-key>

<sup>7</sup><https://console.developers.google.com>

<sup>8</sup><https://developers.google.com/maps/documentation/javascript/reference>



```

// vytvoreni mapy
var map = new google.maps.Map(document.getElementById('map'));

// pridani bodu trasy do pole
var coords = [];
foreach $walk->getCoordinates() as $coordinate
    coords.push({
        lat: {$coordinate->getLatitude()},
        lng: {$coordinate->getLongitude()}
    });
{/foreach}

// vypocitani hranic trasy
var bounds = new google.maps.LatLngBounds();
coords.forEach(function (e) {
    bounds.extend(e);
});

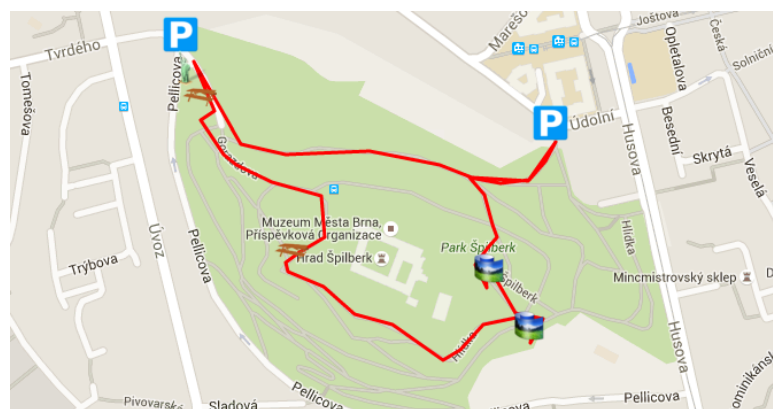
map.setCenter(bounds.getCenter()); // nastavit stred mapy na stred trasy
map.fitBounds(bounds); // nastavit uroven priblizeni

// pridani trasy do mapy
var line = new google.maps.Polyline({
    path: coords,
    strokeColor: '#ff0000',
});
line.setMap(map);

// pridani bodu zajmu
foreach $walk->getPointsOfInterest() as $poi
    var marker = new google.maps.Marker({
        position: {
            lat: {$poi->getLatitude()},
            lng: {$poi->getLongitude()}
        },
        map: map,
        title: {$poi->getDescription()},
        icon: MapMarkerTypes.typeToIcon({$poi->getType()})
    });
{/foreach}

```

Kód 4.1: Ukázka použití třídy Map



Obrázek 4.1: Zobrazení trasy procházky s body zájmu

### 4.2.3 Places service

Pro získání více informací o trase budeme využívat službu Google Places, která umožňuje získávat informace o lokacích, jako jsou body zájmu nebo významné geografické lokace.

Služba poskytuje dva základní režimy vyhledávání. Prvním způsobem je vyhledávání podle textového vstupu a druhým způsobem je vyhledávání podle zadané GPS souřadnice, maximální vzdálenosti od tohoto bodu a seznamem povolených typů. Pro získávání informací v okolí zadané trasy budeme využívat druhého způsobu, tedy vyhledávání bodů zájmu v okolí bodu.

Protože tato služba poskytuje i informace o místech, která nemají význam uvádět u procházek, musíme nejdříve zvolit povolené typy míst. Seznam všech povolených míst lze nalézt v oficiální dokumentaci k Google Places API<sup>9</sup>.

#### Použití

Pro používání této služby je k dispozici třída `google.maps.places.PlacesService`. Tato třída obsahuje několik metod, pomocí nichž můžeme vyhledávat.

- **radarSearch**

Tato metoda umožňuje vyhledávat informace o místech v okolí zadaného bodu. Vrací seznam informací o nalezených bodech, který ovšem obsahuje pouze id místa a jeho souřadnice.

- **textSearch**

Metoda slouží k vyhledávání podle zadaného textového řetězce.

- **nearbySearch**

Metoda je velmi podobná metodě `radarSearch`, ale ve vráceném výsledku obsahuje navíc další informace o nalezených místech.

Z těchto funkcí je pro naše účely je nejvhodnější funkce `nearbySearch`, která vyhledává body zájmu v okolí určité lokace.

```
var service = new google.maps.places.PlacesService(map);

service.nearbySearch({
  location: coord,
  radius: 500,
  types: getAlowedTypes()
}, function(results) {
  for (var i = 0; i < results.length; i++) {
    var marker = new google.maps.Marker({
      position: results[i].geometry.location,
      map: map,
      title: results[i].name,
      icon: MapMarkerTypes.typeToIcon(place.types)
    });
  }
});
```

Kód 4.2: Načítání bodů zájmu pomocí služby Places

---

<sup>9</sup>[https://developers.google.com/places/supported\\_types/](https://developers.google.com/places/supported_types/)

V kódu 4.2 je ukázka načítání bodů zájmu v okolí bodu. Nejdříve je vytvořena nová instance třídy *google.maps.places.PlacesService*, nad kterou je zavolána metoda *nearbySearch*. Ta převezme bod, v jehož okolí se má vyhledávat, maximální vzdálenost od tohoto bodu a seznam povolených typů. Po vrácení metody je zavolána funkce, které je v parametru předáno pole s jednotlivými body zájmu.

## Ukládání bodů zájmu

V rámci zadávání jedné trasy vyhledáváme body zájmu pro každý bod zvlášť. Jelikož se jednotlivé body trasy často nachází blízko u sebe, dochází k tomu, že získáváme jeden bod zájmu vícekrát. Z toho důvodu musíme ukládat seznam již použitých bodů zájmu z Places API. Poté při načtení odpovědi zkontrolujeme, zda jsme každý jednotlivý bod zájmu nevložili do mapy při jednom z předchozích požadavků.

### 4.2.4 Elevation service

Stoupání u trasy pomůže uživatelům získat přehled o náročnosti trasy. Mohou si tak lépe vybrat procházku, která odpovídá jejich fyzickým zdatnostem. Pro získání nadmořské výšky podél trasy procházky využijeme Elevation service. Pro práci s touto službou je připravena třída *google.maps.ElevationService*. Tato třída poskytuje dvě základní metody *getElevationAlongPath* a *getElevationForLocations*.

- **getElevationForLocations**  
Zjistí nadmořskou výšku jednotlivých bodů, které jsou zadány ve formě GPS souřadnic.
- **getElevationAlongPath**  
Zjistí nadmořskou výšku podle zadané cesty, která je zadána pomocí GPS souřadnic jednotlivých bodů. Tato funkce nezjišťuje nadmořskou výšku přímo v jednotlivých bodech, ale celou cestu si rozdělí na stejně dlouhé úseky a v rámci každého úseku zjistí nadmořskou výšku.

Protože dopředu nevíme, jaká bude vzdálenost mezi jednotlivými body, bude lepší použít funkci *getElevationAlongPath*. Při použití funkce *getElevationForLocations* existuje riziko, že budou dva sousední body procházky od sebe příliš vzdálené. Nejistili bychom tedy tvar terénu mezi nimi.

V kódu 4.3 je ukázka základní práce s tímto API. Nejdříve vytvoříme novou instanci třídy *google.maps.ElevationService*. Nad ní zavoláme metodu *getElevationAlongPath*, které předáme body trasy a počet vzorků, pro které chceme zjistit nadmořskou výšku. Metoda po zpracování požadavku a získání odpovědi zavolá funkci, které v parametru předáme informace o nadmořské výšce podél trasy. Funkce dále vytvoří graf, do kterého znázorní průběh stoupání podél trasy.

```
var elevation = new google.maps.ElevationService;

elevation.getElevationAlongPath({
  'path': coords,
  'samples': 256
}, function (elevations) {
  // Create a new chart in the elevation_chart DIV.
  var chart = new google.visualization.LineChart(document.
    getElementById('elevation_chart'));
```

```

var data = new google.visualization.DataTable();
data.addColumn('string', 'Vzdalenost');
data.addColumn('number', 'Nadmorska_vyska');
for (var i = 0; i < elevations.length; i++) {
    data.addRow(['', elevations[i].elevation]);
}

// Draw the chart using the data within its DIV.
chart.draw(data, {
    height: 250,
    legend: 'none',
    pointSize: 1,
    titleY: 'Nadmorska_vyska_(m)',
});

```

Kód 4.3: Získání nadmořské výšky podél trasy

## 4.3 Geocoding API

V rámci aplikace pracujeme jak s GPS souřadnicemi, tak s textovou adresou uživateli aktuální polohy. Pro převod mezi těmito tvary využíváme Geocoding API, které poskytuje společnost Google<sup>10</sup>.



Obrázek 4.2: Adresa uživatele

### 4.3.1 Geokódování

Geokódování je proces, který převádí textovou adresu na GPS souřadnice. Používáme ho v případě, že chce uživatel nastavit ručně svoji polohu. Pro získání souřadnic odešleme požadavek na adresu <https://maps.googleapis.com/maps/api/geocode/json>, kterou doplníme o parametr `address`, ve kterém bude požadovaná adresa. V odpovědi získáme souřadnice adresy ve formátu JSON.

```

public function geocode($address)
{
    $query = 'https://maps.googleapis.com/maps/api/geocode/json?address='
        . rawurlencode($address);
    $request = $this->client->request('GET', $query)->getBody();

    return json_decode($request);
}

```

Kód 4.4: Geokódování

### 4.3.2 Reverzní geokódování

Reverzní geokódování je proces opačný geokódování. Převádí tedy GPS souřadnice na textovou adresu. Pro reverzní geokódování můžeme využít existující metodu `geocode` z ukázky

<sup>10</sup><https://developers.google.com/maps/documentation/geocoding/intro>

4.4, které předáme jako adresu GPS souřadnice.

```
public function reverse($lat, $lng)
{
    return $this->geocode($lat . ', ' . $lng);
}
```

Kód 4.5: Reverzní geokódování

## 4.4 Databázová vrstva

Podobně jako většina ostatních webových stránek potřebuje i naše aplikace ukládat velké množství dat do trvalého úložiště. Kromě samotného ukládání je vhodné, aby úložiště umělo jednoduchým způsobem data načítat, filtrovat a případně i odstraňovat z paměti.

Pro práci s databází byla použita knihovna Doctrine 2. Jedná se o PHP knihovnu poskytující objektově relační mapování (ORM). Poskytuje jistou míru abstrakce nad databázovou vrstvou. V aplikaci tedy s jednotlivými záznamy pracujeme jako s PHP objekty. To nám umožňuje implementovat určitou část aplikační logiky přímo do databázových objektů (tzv. Entit).

Jednotlivé záznamy z relační databáze se v terminologii Doctrine 2 nazývají *entity*. Entita obsahuje instanční proměnné, které reprezentují jednotlivé atributy z tabulky. Atributy mohou být skalárního typu (integer, string, double) nebo mohou představovat určitý vztah mezi více entitami (1:N, N:1 nebo M:N). Pomocí atributu představující vztah můžeme získat z databáze i tyto potřebné záznamy. Význam atributů můžeme definovat několika způsoby. Nejčastěji používaným způsobem je zápis pomocí anotačních komentářů.

V následujícím kódu je ukázka jednoduché entity uživatele obsahující dva atributy. Atribut login je typu string o maximální délce 25 znaků. Atribut walks představuje vztah mezi uživatelem a procházkami. V tomto případě obsahuje kolekci všech procházek, jejichž autorem je daný uživatel.

```
/** @ORM\Entity */
class User
{
    /** @ORM\Column(type="string", length=25) */
    protected $login;

    /** @ORM\ManyToOne(targetEntity="Walk", mappedBy="author") */
    protected $walks;
}
```

Kód 4.6: Ukázka entity

Entity z databáze získáváme pomocí Repository tříd. Jsou to třídy, které využívají připojení k databázi a pomocí SQL dotazů získávají z databáze data. Kromě samotného získávání dat umožňují aktualizaci případně mazání existujících entit.

Jelikož Doctrine podporuje hned několik databázových vrstev, jejichž SQL se často liší, zavádí svůj vlastní dotazovací jazyk Doctrine query language - DQL. DQL je jazyk velmi podobný SQL, ale na rozdíl od většiny SQL nepracuje s názvy tabulek a sloupců, ale s názvy PHP tříd a jejich členskými proměnnými. Při vyhodnocování DQL výrazu se dotaz nejprve zpracuje a převede do SQL jazyka databázové vrstvy, která je použita. Po vyhodnocení je SQL dotaz předán databázové vrstvě, která dotaz vyhodnotí. Výsledek je předán zpět do

Doctrine, která tuto množinu převezme a vytvoří odpovídající entity, které jsou naplněny odpovídajícími daty. Tento proces je také nazýván hydratace entit<sup>11</sup>.

#### 4.4.1 Skládání dotazů

Doctrine obsahuje třídu pro dynamické skládání databázových dotazů - Query Builder.

```
$qb = $this->em->createQueryBuilder()  
->select('walk')  
->from(Entity\Walk::class, 'walk');
```

Kód 4.7: Query Builder

Tato třída sice poskytuje dobré API pro sestavování dotazů, nicméně na všech místech musíme znát přesnou strukturu databáze. V případě jakékoliv změny struktury databáze bychom museli upravovat dotazy na spoustě míst. Tento problém řeší speciální třídy pro sestavování SQL dotazu - tzv. query object.

#### 4.4.2 Query object

Query object je speciální třída, která reprezentuje SQL dotaz. Jeho hlavním cílem je umožnit další úroveň abstrakce. Programátor využívající query object nemusí nutně znát jakým přesným způsobem se SQL dotaz sestavuje, protože mu třída obvykle nabízí bohaté API, kterým může výsledný dotaz do databáze upravovat [4].

Query object nabízí rozhraní, které zpravidla reprezentuje jednotlivé vlastnosti z hlediska aplikační logiky.

Ukázka metod, které Query Object může nabízet

```
class WalkQuery extends QueryObject  
{  
    public function byDistance(Coordinate $coordinate, $distance);  
    public function byProperties($properties)  
    public function byName($name)  
    public function byUserFavorite($user)  
    public function withFavorites()  
    public function withAttributes()  
    public function sortByDistance(Coordinate $coordinate)  
}
```

Kód 4.8: Query Object

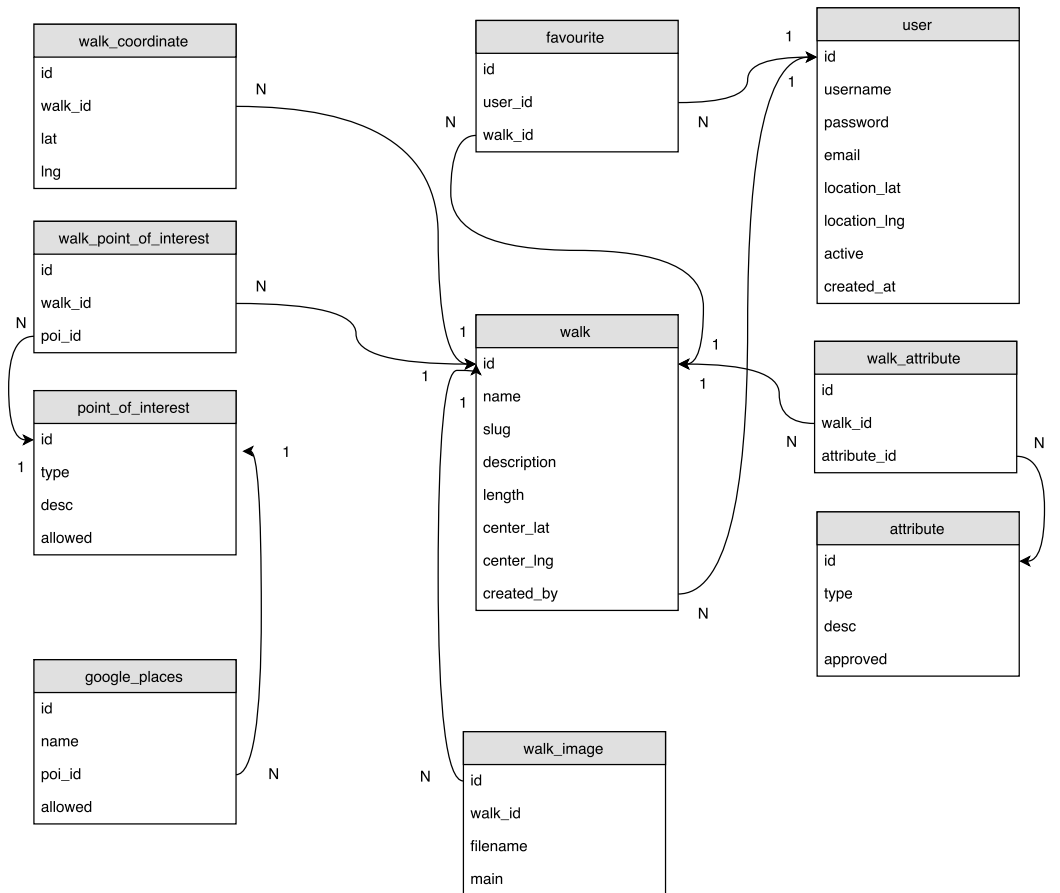
V předchozím příkladu je ukázka několika skutečných metod, které v rámci aplikace používám. Metody začínající slovem *by* značí filtrovací podmínky. Do výsledného SQL dotazu tedy doplňují podmínky do WHERE. Dále se zde vyskytuje několik metod začínajících klíčovým slovem *with*. Tyto metody slouží k připojení dalších informací k hlavnímu záznamu. Do výsledného SQL dotazu tedy doplňují JOIN. Poslední kategorie metod, které můj Query Object poskytuje, jsou řadící dotazy, které do výsledného SQL dotazu doplňují klauzuli ORDER BY.

## 4.5 Databáze

Pro uložení dat do databáze byl vytvořen relační model, který je znázorněn na ER schématu na obrázku 4.3. Základní tabulkou je walk, která obsahuje základní informace o procházce jako její název, popis a autora. Jednotlivé body procházky jsou uloženy v tabulce

<sup>11</sup><http://doctrine-orm.readthedocs.io/projects/doctrine-orm/en/latest/reference/unitofwork.html>

walk\_coordinates. Tabulka obsahuje GPS souřadnice bodu procházky a id procházky, ke které se vztahují.



Obrázek 4.3: Schéma databáze

- **User**

Uchovává všechny důležité informace o uživateli. Mezi nimi jsou i přihlašovací údaje, kterými je dvojice uživatelského jména a otiskem hesla. Není zde tedy uloženo heslo v čisté textové podobě, ale otisk provedený algoritmem blowfish <sup>12</sup>. Dalšími důležitým údajem, který se zde vyskytuje, je uživatelova poloha, kterou využíváme především k vyhledávání.

- **Walk**

Entita obsahující základní informace o procházce. Obsahuje její popis, název, délku a souřadnice středu procházky, které jsou vypočítány na základě zadaných trasy. Bod středu procházky se sice dá dopočítat pomocí entity WalkCoordinate, nicméně z hlediska výkonu při vyhledávání se tato hodnota ukládá i přímo do hlavní entity.

- **Attribute**

Obsahuje informace o možných vlastnostech procházky. Kromě samotného názvu vlastnosti obsahuje příznaky, které určují, zda je vlastnost povolena pro výpis a zda se má vlastnost vypisovat na předních místech ve filtru.

<sup>12</sup><http://php.net/manual/en/function.crypt.php>

- **PointOfInterest**  
Definuje seznam možných typů bodů zájmu. Obsahuje pouze název typu bodu zájmu.
- **WalkCoordinate**  
Je tabulka obsahující body, které tvoří trasu procházky. Entita obsahuje referenci na procházku a GPS souřadnice bodu.
- **WalkAttribute**  
Tabulka jejímž úkolem je přiřadit vlastnost k procházce. Obsahuje referenci na procházku a referenci na vlastnost.
- **WalkPointOfInterest**  
Entita přiřazuje bod zájmu k procházce. Kromě reference na bod zájmu a procházku obsahuje ještě GPS souřadnice, na kterých se bod zájmu nachází. K bodu zájmu je zde ještě možné přiřadit textový popis.
- **GooglePlaces**  
Obsahuje seznam všech dostupných typů bodů zájmu, které můžeme vyhledávat pomocí služby Google Places. Entita obsahuje identifikátor druhu bodu zájmu, který je použit při odesílání požadavku na API. Dále obsahuje příznak, který určuje, zda je typ povolen pro odeslání a jako poslední atribut je reference na bod zájmu, na který je mapován.

## 4.6 Struktura aplikace

Aplikace je rozdělena do několika základních modulů. Každý z těchto modulů má pevně danou strukturu. Obsahuje minimálně adresáře

- **DI**  
Zde se modul registruje do systémového kontejneru, přidává do něj své služby a případně modifikuje již existující služby.
- **presenters**  
Obsahuje části aplikace, které jsou zodpovědné za převod HTTP požadavku na požadovanou činnost aplikace.
- **components**  
Obsahuje znovupoužitelné části modulu jako jsou například formuláře.

Výsledná aplikace je složena ze čtyř modulů. Každý z těchto modulů má svůj úkol, který je popsán v následujícím přehledu.

- **WalkModule**  
Modul slouží k práci s jednou konkrétní procházkou. Jeho základní funkcí je zobrazení detailu procházky. Dále obsahuje formuláře pro přidání a úpravu procházky včetně jejich zpracování. Dalším úkolem tohoto modulu je zobrazit seznam procházek. Kromě samotného zobrazování obsahuje rovněž komponentu pro vyhledávání v těchto procházkách.



- **AdminModule**  
Poskytuje rozhraní pro administrátora. Umožňuje základní správu uživatelů, atributů procházek. Dále umožňuje nastavení služby Google Places, u kterého lze povolit, které typy lokací se budou načítat.
- **UserModule**  
Zajišťuje přihlašování a registraci uživatelů.
- **ApiModule**  
Modul implementuje veškeré metody dostupné pro API k mobilní aplikaci.

## 4.7 Zjišťování aktuální polohy uživatele

Abychom mohli vyhledávat procházky v okolí uživatele, musíme nejdříve znát jeho přesnou polohu. Existuje několik způsobů, pomocí kterých můžeme lokalizovat uživatele. Jednotlivé způsoby zjišťování polohy můžeme rozdělit na 2 základní druhy, podle toho, zda vyžadujeme interakci uživatele.

### 4.7.1 HTML5 geolocation API

Jedná se o rozhraní, které umožňuje aplikacím získat aktuální polohu uživatele. Způsob, kterým tato metoda zjišťuje polohu, je závislý na dostupném hardwarovém vybavení počítače<sup>13</sup>. Od dostupné metody se odvíjí i přesnost lokalizace. Lokalizace může být provedena pomocí GPS modulu, který nabízí největší přesnost. Pokud není GPS modul dostupný, provede se geolokace z dalších zdrojů, kterými mohou být

- IP adresa
- RFID
- WiFi síť,
- GSM/CDMA ID

Prohlížeče nesdílejí polohu automaticky, ale uživatel musí sdílení explicitně povolit. Je tedy možné, že uživatel sdílení své polohy nepovolí.

Protože se jedná o citlivé osobní údaje, některé prohlížeče (například Chrome od verze 50<sup>14</sup>) vyžadují, aby webová stránka zjišťující aktuální polohu uživatele využívala zabezpečený komunikační kanál, tedy HTTPS.

### 4.7.2 Geolokace podle IP adresy

Pokud uživatel nepovolí geolokaci ve svém prohlížeči nebo jeho prohlížeč tuto funkci nepodporuje, pokusíme se alespoň zjistit přibližnou polohu podle jeho IP adresy. Abychom mohli zjistit polohu uživatele podle IP adresy, musíme mít nejdříve přístup k databázi obsahující tyto informace. Jednou z nich je IPInfoDB<sup>15</sup>, která je v rámci aplikace využívána.

<sup>13</sup><https://www.w3.org/TR/geolocation-API/>

<sup>14</sup><https://codereview.chromium.org/1530403002/>

<sup>15</sup><http://www.ipinfodb.com/>

## 4.8 Zabezpečení proti robotům

Krátce po zveřejnění aplikace na internet docházelo k velkému množství nově registrovaných účtů pomocí robotů. Naším cílem bylo zamezit nebo aspoň z velké části omezit robotům zakládat nové účty.

V této kapitole jsou popsány jednotlivé metody, které řeší tento problém.

### 4.8.1 Ověřování účtu pomocí emailu

U této metody dochází k odeslání emailu s ověřovacím kódem na emailovou adresu uvedenou při registraci. Kód je obvykle odeslán ve formě odkazu, na který stačí uživateli jen přejít a tím ověří svoji identitu. Tato metoda avšak neřeší to, že budeme mít v databázi uloženou spoustu neaktivních účtů.

### 4.8.2 CAPTCHA

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) je metoda, jejímž cílem je rozlišit počítač od člověka. Základem je úloha, která je pro běžného člověka snadno řešitelná, ale pro počítače by měla být neřešitelná<sup>16</sup>.

V dnešní době se nejčastěji můžeme setkat s obrázkovou verzí tohoto testu, kdy má uživatel za úkol přepsat text z obrázku do textového pole. Do obrázku je vykreslen text, který je určitým způsobem zdeformován, aby se nedaly znaky dekódovat pomocí OCR<sup>17</sup>. V některých verzích obrázkových CAPTCHA kódů bývá ale deformace tak veliká, že mají lidé problémy s přečtením znaků [2].

Tato metoda je sice vysoce účinná, ale nutíme běžné návštěvníky webu dělat jeden krok navíc, což může vést i k odrazení od využívání našich služeb.



Obrázek 4.4: Ukázka CAPTCHA kódu.

### 4.8.3 Skryté pole

Abychom měli nějakou míru zabezpečení formulářů, ale nenutili jsme uživatele dělat další krok navíc, můžeme do formuláře přidat skrytý prvek, ve kterém budeme po odeslání očekávat určitý obsah. Tato metoda nedokáže vyřešit všechny roboty, ale dokáže eliminovat jejich většinu.

Využívá dvou základních vlastností, které má většina robotů společné:

- nepodporují JavaScript
- snaží se vyplnit všechna dostupná pole

Do formuláře přidáme nový prvek, který bude před uživatelem vizuálně skryt. V tomto prvku budeme při odeslání vyžadovat předem definovaný text. Jelikož většina robotů neumožňuje spouštět javascriptový kód, můžeme tento prvek vyplnit automaticky. Musíme

<sup>16</sup><http://wiki.knihovna.cz/index.php/CAPTCHA>

<sup>17</sup>Optical character recognition

ovšem počítat i s uživateli, kteří ve svém prohlížeči nemají povolený JavaScript a nabídnout jim možnost vyplnění prvku formuláře, proto bude tento prvek skrytý pouze pokud má klient povolený JavaScript.

Tato ochrana je účinná pouze pro obecné roboty. Pokud někdo zná náš systém, nebude mu dělat žádný problém robota upravit tak, aby tuto ochranu obešel. V současné době ale nepředpokládám, že někdo bude vytvářet robota specifického pro naše stránky, proto bude tato ochrana dostatečná.

#### 4.8.4 Odesílání časového razítka

Další účinnou metodou pro zamezení vytváření nových účtů roboty je odesílání časového razítka v jednom ze skrytých prvků formuláře. Roboti zpravidla vyplní formulář daleko rychleji než běžný uživatel. Do skrytého formulářového prvku tedy uložíme aktuální serverový čas, který po odeslání zkontrolujeme. Jestliže se čas z formulářového prvku liší od aktuálního serverového času naprosto minimálně (v řádu jednotek sekund), můžeme předpokládat, že se jedná o vyplnění formuláře robotem [1]. Přesný časový limit, který označíme jako hraniční pro detekci robota, je nutno volit s ohledem na složitost odesílaného formuláře. V našem případě byl tento čas stanoven na 3 sekundy.

#### 4.8.5 Závěr

Před nasazením jakékoliv ochrany docházelo k vytvoření přibližně 10 nevyžádaných účtů denně. Proto byla nasazena ochrana metodou skrytého pole a metodou odesílání časového razítka. Po nasazení těchto ochran nebyl roboty vytvořen žádný další účet.

## Kapitola 5

# Zveřejnění aplikace a sledování uživatelů

V této kapitole bude popsán proces zveřejnění aplikace na internet. Následně bude vyhodnoceno chování uživatelů na webové stránce.

### 5.1 Zveřejnění aplikace

Výsledná aplikace byla zveřejněna na internetové stránce <http://www.naprochazku.cz>. Jako hosting byl vybrán Wedos s tarifem NoLimit<sup>1</sup>, který nabízí dostatečný výkon pro naši aplikaci.

### 5.2 Přímé sledování

Pro přímé sledování byla zvolena skupina 10 lidí, kterým byl zadán úkol, který měli splnit. Při plnění úkol bylo zaznamenáváno, co uživatelům dělá největší problémy.

V tabulce 5.1 je uveden seznam uživatelů, kteří testovali tento systém.

<b>Věk</b>	21	25	49	47	19	29	16	41	36	51
<b>Pohlaví</b>	M	Ž	Ž	M	M	M	Ž	Ž	M	Ž
<b>Vzdělání</b>	S	V	S	V	Z	S	Z	S	V	S

Tabulka 5.1: Testovací skupina uživatelů. Pohlaví: M - muž, Ž - žena. Vzdelání: Z - základní, S - středoškolské, V - vysokoškolské

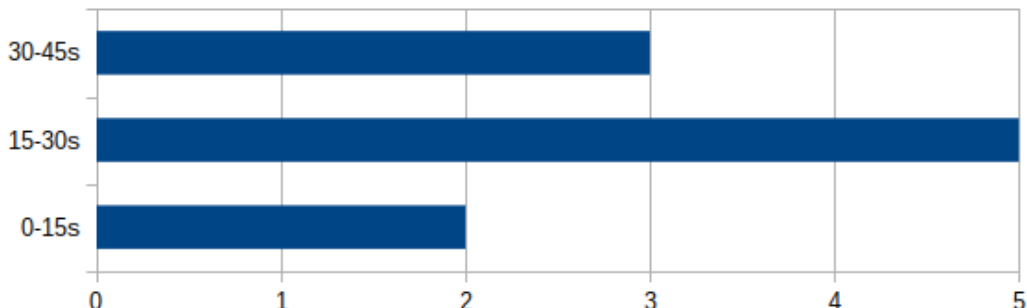
V rámci prvního úkolu měli uživatelé vyhledat všechny procházky, které splňují zadaná kritéria, kterými byly:

- Vzdálenost do 25km
- Obsahuje hřiště

Pro každého z testovacích uživatelů se jednalo o první seznámení se systémem. Před samotným vypracováním úkolu jim byl poskytnut krátký čas na seznámení se systémem.

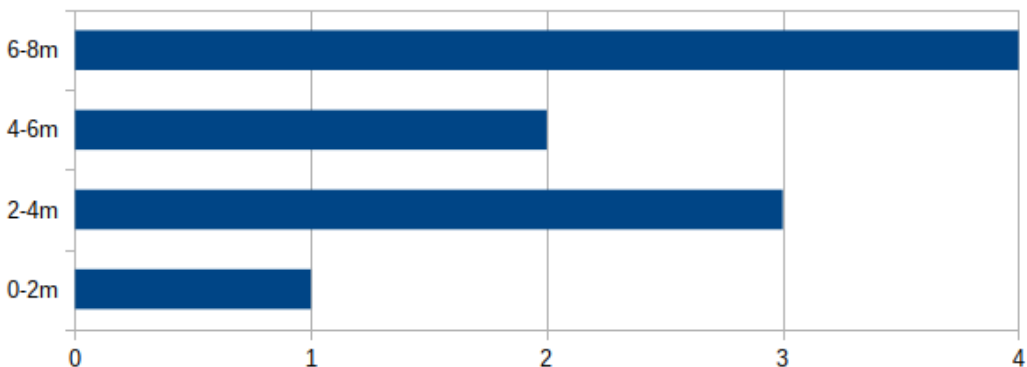
<sup>1</sup><https://hosting.wedos.com/cs/webhosting.html>

V grafu 5.1 je znázorněna doba, která jednotlivým uživatelům zabrala k získání výsledku. S vyhledáváním podle vzdálenosti neměl žádný z uživatelů problémy. Některým uživatelům dělal problém vyhledat požadovanou vlastnost v celé nabídce vlastností. Nicméně celkový čas strávený pro vyhledávání procházek nebyl u nikoho výrazně dlouhý.



Obrázek 5.1: Doba potřebná k vyhledání procházky.

Druhým úkolem bylo zadání procházky. Uživatelům jsme nechali čas na rozmyšlení, kterou procházku budou zadávat a čas jsme začali měřit až při začátku zadávání. V grafu 5.2 je znázorněna doba, kterou uživatelé zadávali procházku. Tato úloha nemá takový vypočítací charakter, jako předchozí úkol, protože se velmi lišila délka zadávaných tras, kde obecně platí, že delší trasu uživatelé zadávají delší čas.



Obrázek 5.2: Doba potřebná k vytvoření nové procházky.

## 5.3 Nepřímé sledování

V rámci nepřímého sledování chování uživatelů využíváme technik, které zaznamenávají pohyb uživatele na stránce a později nám umožňují tyto statistiky zobrazovat a vyhodnocovat.

### 5.3.1 Google Analytics

Základní službou, která byla využita pro sledování chování uživatelů je Google Analytics. Ihned po zveřejnění aplikace na internet byla nasazena služba Google Analytics, která má za úkol sledovat uživatele, kteří navštíví naši stránku. Kromě samotné návštěvy dokáže

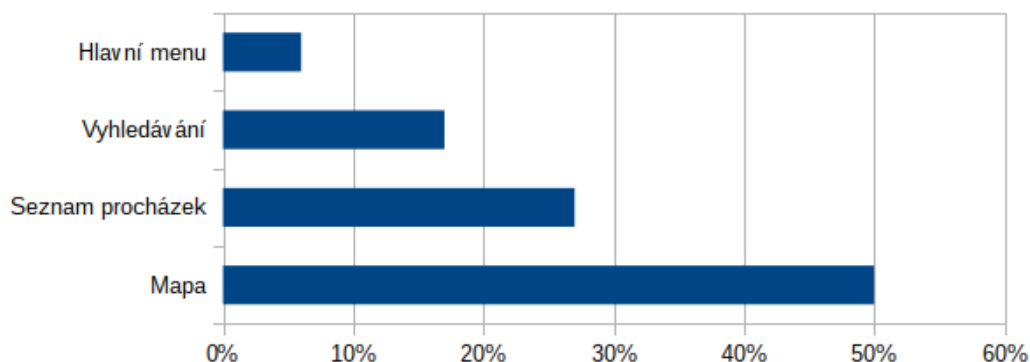
sestavovat i diagramy stránek, které jednotlivý uživatel navštívil. Z nich poté můžeme zjistit, jak se na stránce jednotliví uživatelé pohybují.

Bohužel po nasazení tohoto sledovacího nástroje neodpovídal počet návštěv skutečnému počtu. Jak se později ukázalo, tak mnoho z mých testovacích subjektů používalo rozšíření pro prohlížeče, které blokují odesílání informací na Google Analytics. Jedním z těchto rozšíření je například Ghostery<sup>2</sup>.

### 5.3.2 Heatmap

Abychom získali větší přehled o pohybu uživatelů na stránkách, byla na něj nasazena služba Hotjar<sup>3</sup>, která umožňuje zaznamenávat veškeré akce, které uživatel na naší stránce provedl.

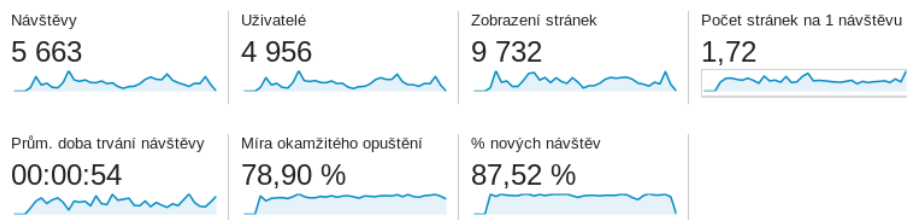
Při sledování mapy pomocí heatmap byla stránka rozdělena do několika sekcí a pro každou byl zaznamenáván počet kliknutí. U seznamu procházek byla jednoznačně nejpoužívanějším prvkem mapa, která má přesně 50 % ze všech kliknutí. Druhým nejpoužívanějším prvkem stránky je seznam procházek se stránkováním, na které bylo kliknuto ve 27 %. Dalším prvkem je vyhledávání, na které bylo kliknuto v 17 % případů a posledním prvkem je hlavní menu, na které bylo kliknuto ve zbývajících 6 % případů.



Obrázek 5.3: Statistika cíle kliknutí.

### 5.3.3 Návštěvnost

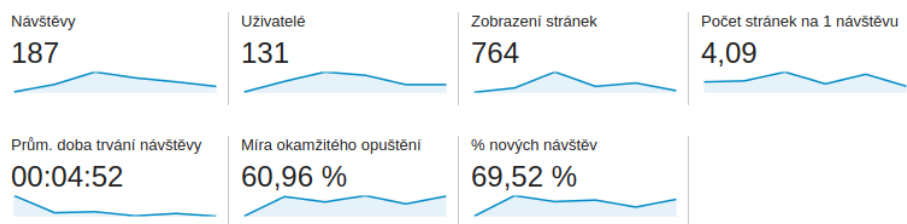
Jelikož byla webová aplikace pouze měsíc v provozu, nemusí být data o návštěvnosti směrodatná, ale mohou naznačit změny v chování uživatelů proti původní verzi. Největší změna nastala v průměrném počtu zobrazení stránek na jednu návštěvu, která se zvedla z 1,72 na 4,09 zobrazených stránek na návštěvu. Zároveň s tím se zvedla průměrná doba trvání návštěvy z 54 sekund na 4 minuty a 52 sekund.



Obrázek 5.4: Statistiky z původní verze

<sup>2</sup>[https://apps.ghostery.com/en-GB/apps/google\\_analytics](https://apps.ghostery.com/en-GB/apps/google_analytics)

<sup>3</sup><https://insights.hotjar.com>



Obrázek 5.5: Statistiky ze současné verze

# Kapitola 6

## Závěr

Cílem této práce bylo provést novou implementaci webového portálu pro sdílení tipů na krátké výlety v okolí. Pro realizaci byl zvolen PHP framework Nette. Aplikace byla napojena na API společnosti Google, které umožňují získávat dodatečné informace k procházkám. Díky tomu jsme dosáhli, že většina procházek obsahuje mnohem více bodů zájmu.

Důležitou součástí aplikace je získávání aktuální polohy uživatele, které je implementováno pomocí geolokačního rozhraní, které poskytuje většina moderních prohlížečů. Pokud uživatel používá prohlížeč, který nepodporuje geolokační rozhraní nebo nepovolil sdílení své polohy, je využita geolokace pomocí IP adresy, která sice nedosahuje takové přesnosti, jako geolokační rozhraní prohlížeče, ale pro použití v naší aplikaci je stále dostatečná.

### 6.1 Navrhovaná rozšíření

V rámci testování aplikace na reálných uživateliích vzešlo několik návrhů na možná další rozšíření. Některá z nich byla ještě v závěru práce implementována.

Častým požadavkem byla možnost deaktivace nebo aspoň skrytí automaticky doplňovaných bodů zájmu ve fázi zadávání trasy. Některé uživatele to rozptylovalo a hůře se jim poté soustředilo na zadávání trasy.

Ze sledování uživatelů vyplynulo, že je málo využívaná možnost registrace na naše stránky. Registraci by bylo vhodné uživatelům ještě více usnadnit třeba možností přihlásit se pomocí již existujícího účtu na serveru Google nebo Facebook.

Pro další vývoj je třeba dostat aplikaci do povědomí co nejvíce lidí. Důležité je získat lepší umístění ve vyhledávačích, k čemu by mohlo dopomoci vytvoření několika zpětných odkazů. Dalším způsobem jak nalákat lidi k využívání naší služby by mohlo být vytvoření stránek na sociálních sítích, kam by se mohly přidávat tipy na zajímavé procházky z našeho webu.

V průběhu vývoje webové aplikace došlo ke stažení mobilní aplikace z obchodu Google Play. V dalším vývoji je možné se zaměřit na vytvoření nové mobilní aplikace. Další možností je vytvoření jazykových mutací, které by mohly zajistit přísun nových uživatelů. Webová aplikace by poté mohla být využívána i lidmi, kteří navštíví cizí zemi a rádi by se podívali na nějaká turisticky atraktivní místa.



# Literatura

- [1] Securing Forms Without Captcha.  
<https://jumping-duck.com/tutorial/securing-forms-without-captcha/>,  
2013-05-29 [cit. 2016-05-04].
- [2] Bursztein, E.; Bethard, S.; Fabry, C.; aj.: How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation. 2010.
- [3] Dočkal, R.: *Naprocházku.cz - mobilní služba pro poznávání okolí*. Bakalářská práce, Vysoké učení technické, 2013.
- [4] Fowler, M.; Rice, D.; Foemmel, M.; aj.: *Patterns of Enterprise Application Architecture*. Addison-Wesley UK, 2002, iSBN 0-321-12742-0.
- [5] Špaček, J. E.: *NaProcházku.cz - Webová služba pro poznávání okolí*. Bakalářská práce, Vysoké učení technické, 2013.

# Příloha A

## Obsah CD

**thesis/naprochazku.pdf** technická zpráva ve formátu pdf

**thesis/naprochazku.avi** prezentační video ve formátu avi

**thesis/latex** adresář zdrojových souborů technické práce ve formátu jazyka Latex

**src/readme.txt** návod k instalaci

**src/doc** složka s programovou dokumentací

**src/app.zip** archiv zdrojových souborů implementované aplikace

**src/database.sql** soubor ve formátu SQL pro instalaci schématu databáze