

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

NAPROCHÁZKU.CZ - WEBOVÁ SLUŽBA PRO
POZNÁVÁNÍ OKOLÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ E. ŠPAČEK

BRNO 2013



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

NAPROCHÁZKU.CZ - WEBOVÁ SLUŽBA PRO POZNÁVÁNÍ OKOLÍ

NAPROCHÁZKU.CZ - WEB SERVICE FOR NEIGHBOUR WALKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JIŘÍ E. ŠPAČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÓKE, Ph.D.

BRNO 2013

Abstrakt

Podělení se o své zážitky s ostatními na internetu je v dnešní době velmi oblíbené. A proto vznikla služba, která má za cíl usnadnit návštěvníkům sdílení svých oblíbených procházek. Tato práce se zabývá návrhem a implementací aplikace, která bude schopna toto nabídnout. Jako hlavní prvek zobrazování jsou použity Google Maps. Součástí implementace je i vlastní otevřené aplikační rozhraní pro komunikaci s mobilní aplikací. K zobrazování tipů na procházky je využito sledování aktuální zeměpisné polohy návštěvníka. Práce se zabývá i sledováním chování návštěvníků pracujících s touto aplikací a je dosaženo zajímavých výsledků. Je naplánováno další rozšiřování aplikace, které by mělo vést k udržení stávajících uživatelů a získávání stále nových návštěvníků.

Abstract

Nowadays the sharing of the experiences by the internet is very popular and that was the reason for establishment of service that aims to facilitate visitors to share their favorite walks. This work describes the design and implementation of application that will be able to offer this. As a major element of the display is used the Google Maps. The implementation is an own open application interface to communicate with mobile applications. To view tips for the walks is used an actual geographic location of the visitor. The work deals with the monitoring behavior of visitors working with this application as well. It is achieved interesting results. In the future there is a plan to expansion the application to keep the existing users and gaining new.

Klíčová slova

Procházka, poznávání okolí, web, navigace, geolokace, Google Maps, PHP, Nette Framework, chování uživatelů.

Keywords

Walk, cognition neighborhood, web, navigation, geolocation, Google Maps, PHP, Nette Framework, user behavior.

Citace

Jiří E. Špaček: Naprocházku.cz - webová služba pro poznávání okolí, bakalářská práce, Brno, FIT VUT v Brně, 2013

Naprocházku.cz - webová služba pro poznávání okolí

Prohlášení

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

.....

Jiří E. Špaček
13. května 2013

Poděkování

Chtěl bych poděkovat svému vedoucímu diplomové práce, Ing. Igoru Szökemu, Ph.D, za cenné rady, za poskytnutí zajímavého tématu a za jeho vlastní iniciativu, dále všem přátelům a známým, kteří se podíleli na testování aplikace a také paní učitelce za korekci textu.

© Jiří E. Špaček, 2013.

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

1 Úvod	3
2 Průzkum trhu	4
2.1 Kudy z nudy.cz	4
2.2 Turistika.cz	4
2.3 Tripomatic.com	5
2.4 Závěr	5
3 Návrh služby	6
3.1 Procházka	6
3.1.1 Atributy trasy	6
3.1.2 Doplnkové informace	6
3.1.3 Trasa na mapě	7
3.1.4 Zájmové body	7
3.1.5 Sdílení	7
3.2 Přehled procházek	7
3.2.1 Přehled v tabulce	8
3.2.2 Mapa	8
3.2.3 Tipy na procházky	8
3.3 Uživatel	8
3.3.1 Údaje o uživateli	8
3.3.2 Role	8
3.4 API	9
3.4.1 Komunikační protokol	9
3.5 Uživatelské rozhraní	9
3.5.1 Steve Krug	9
3.5.2 Jakob Nielsen	10
3.5.3 Vlastní shrnutí	10
3.5.4 Návrh	10
3.6 Podporované webové prohlížeče	11
3.7 Určení polohy	11
3.7.1 Ochrana soukromí	12
3.8 Diagram užití	12
3.9 Databáze	13

4	Použité technologie	14
4.1	Nette Framework	14
4.1.1	Vlastnosti	14
4.1.2	MVC	14
4.1.3	Podpůrné integrované nástroje	15
4.2	Google Maps API	15
4.2.1	API Mapy.cz v4	16
4.2.2	Google Maps API v3	16
4.2.3	API Access key	16
4.3	Databáze InnoDB	16
5	Implementace	17
5.1	Použitý software a vývojové prostředí	17
5.1.1	NetBeans	17
5.1.2	WampServer	17
5.1.3	SVN	17
5.2	Zpracování akcí a zobrazení v HTML	17
5.3	Google Maps	18
5.3.1	Zobrazení detailu procházky na mapě	19
5.3.2	Zaznamenání trasy na mapu	19
5.4	Atributy	21
5.4.1	Správa atributů	22
5.5	Zájmové body	22
5.6	Určení polohy	22
5.6.1	Nabídka tipů dle polohy	23
5.7	Vyhledávání	23
5.8	Zaznamenávání událostí	24
5.9	Zveřejnění	25
6	Sledování chování uživatelů	26
6.1	Přímé sledování	26
6.2	Nepřímé sledování	28
7	Navrhovaná rozšíření	31
8	Závěr	32
A	Obsah CD	34
B	Definice metod aplikačního rozhraní	35

Kapitola 1

Úvod

V dnešní době, kdy se mnoho informací hledá na internetu a ne v papírové podobě, jak tomu bylo dříve, je přínosné publikovat své poznatky a zážitky právě na internetu a sdílet je s ostatními.

Právě sdílení našich poznatků je cílem této práce jakožto webové služby. Ta má za cíl umožnit sdílení procházek z našeho okolí. Z okolí, které známe, kde máme své oblíbené trasy, procházky a rádi bychom se o ně s ostatními podělili. Cesta celé procházky je zaznamenána na mapě, a to včetně zajímavých zastávek, odpočívadel, přístřešků, stánku s občerstvením a především informacemi o procházce samotné (jako je dosah parkoviště, možnost absolvovat ji na kolečkových bruslích, dostupnost s kočárkem či invalidním vozíkem nebo stav cesty po deštivém dni a mnoho dalších). Samozřejmostí je propojení s mobilní aplikací, jež nám usnadní zaznamenání trasy dle GPS souřadnic, a to pomocí implementovaného otevřeného aplikačního rozhraní.

Po úvodní kapitole přiblížím webové služby, které nabízejí podobné možnosti. Třetí kapitola se věnuje návrhu této služby se všemi možnostmi, které nabízí. Ve čtvrté se budu zabírat technologiemi, které jsou v aplikaci použity spolu s jejich popisem. V páté kapitole je popsána implementace samotné webové služby naProcházku.cz s ukázkami kódů funkcí a jejich popisem. Kapitola šestá analyzuje získaná data od prvních uživatelů služby a hodnotí chování návštěvníků. A v sedmé navrhnu další možnosti vylepšení a rozšíření této webové služby.

Kapitola 2

Průzkum trhu

Oblast působení webové služby zaměříme především na českého uživatele. Trh není malý, působí zde hlavně dva české portály s podobnými službami. Avšak žádný zatím neposkytuje propojení webu s mobilní aplikací v obou směrech. To znamená zobrazení procházky v mobilu s možností zobrazovat aktuální polohu na trase a nebo využít funkci zaznamenání trasy. Třetí služba, také českého původu, která ale působí celosvětově, mě velmi zaujala, i když je určena především k plánování výletů na více jak jeden den.

2.1 Kudy z nudy.cz

Kudyznudy.cz je projekt státní agentury CzechTourism podporovaný dotacemi Evropské unie. Jeho oficiálně udávaná měsíční návštěvnost¹ se pohybuje kolem 450 tisíc unikátních návštěvníků a 10 tisíc registrovaných uživatelů. Avšak nezávislé měření hovoří o návštěvnosti poloviční.

Příspěvky vkládají jak správci, tak registrovaní uživatelé služby. To jsou především provozovatelé, turistická informační centra nebo organizace cestovního ruchu. Podmínkou zveřejnění je umístění reklamy Kudyznudy.cz na webových stránkách provozovatele akce či aktivity. Každý příspěvek je správcem autorizován. Je na výběr mezi akcí, aktivitou a výletem. Výlet je nejbližší naší procházce. Lze shlédnout již naplánovaný výlet včetně zastávek a zajímavých míst. Můžeme si vytvořit i vlastní výlet a naplánovat trasu, ale spojit lze pouze body, které jsou na mapě již vyznačeny jako aktivita. Takovýto vytvořený výlet lze uložit a následně pak doporučit redakci. Je na její zvážení, zda výlet umožní sdílet s ostatními či nikoli.

2.2 Turistika.cz

Další rozšířenou službou je Turistika.cz. Partnerem této služby je mimo jiné Klub českých turistů a sběratelů turistických známek. Oficiálně udávaná návštěvnost² nezávislým měřením společností NetMonitor je 400 tisíc unikátních návštěvníků za měsíc a počet registrovaných uživatelů je 40 tisíc.

Obsah portálu budují sami uživatelé. Na výběr je opět mezi několika typy, a to výletem, turistickým cílem a trasou. Trasa je nám nejbližší. Trasu na mapě můžeme vytvořit buď načtením souboru GPX, nebo ručním zadáním na mapě. V obou případech vytvoříme na

¹<http://www.kudyznudy.cz/O-kudy-z-nudy.aspx>

²<http://www.turistika.cz/texty/reklama>

mapě několik bodů, ke kterým můžeme následně přidat popisek. K celé trase lze přidat fotogalerii. Trasu můžeme označit jako absolvovanou, či si ji přidat do seznamu míst, které chci v budoucnu navštívit.

2.3 Tripomatic.com

Na tuto velice zajímavou službu jsem narazil až v průběhu implementace mé služby. Zaujala mě velice příjemným a intuitivním uživatelským rozhraním. Jak už jsem zmínil, je především určena pro plánování a poté sdílení vícedenních výletů po celém světě. Nabízí propojení s mobilní aplikací. V případě, že uživatel vytváří svůj první plán trasy, aplikace ho naviguje, zobrazuje mu tipy a učí ho správnému postupu.

Má zaveden takový přístup, že plán cesty si může návštěvník vytvořit, aniž by byl do služby zaregistrován. Poté až v případě, že chce plán uložit, je vyzván k přihlášení nebo možnosti vytvoření nového uživatelského účtu. Tento způsob je podle mě přínosný, protože neodradí návštěvníka v tom, aby si službu vyzkoušel bez nutnosti se registrovat.

2.4 Závěr

Z těchto třech služeb mě nejvíce zaujal Tripomatic, a to interaktivní tvorbou plánu trasy. Při tvorbě trasy uživateli nabízí mnoho tipů, které ho vedou ke zdárnému dokončení jeho plánu. Prvně jmenovaná služba Kudyznudy zase hezkým grafickým návrhem, který zapůsobí na nejednoho návštěvníka. Naopak Turistika.cz je příkladem, kde obsah tvoří sami návštěvníci, a je tak zcela v rukou všech lidí se stejnými zálibami. Ve službě naProcházku.cz bych chtěl proto nabídnout od každého trochu - interaktivní uživatelské prostředí, příjemný design, nebránit uživatelům v kreativité při tvorbě procházek. A v neposlední řadě bych chtěl zaujmout kvalitním propojením s mobilní aplikací, aby jak web, tak aplikace na mobil byly samy zcela životaschopnými službami, ale spolu tvořily neocenitelného společníka při plánování a sdílení procházek. V těchto aspektech by byla tato služba naprosto vyjimečná oproti třem výše jmenovaným.

Kapitola 3

Návrh služby

Webová služba naProcházku.cz je zejména koncipována jako služba pro sdílení procházek. Má za cíl návštěvníkovi a uživateli poskytnout nástroje na zaznamenání jeho oblíbené procházkové trasy. K ní na mapě vyznačit a připojit zajímavá místa, přidat popisek či doporučení, přiložit fotografie z procházky a sdílet s ostatními návštěvníky. Návrh tohoto bude v nynější kapitole popsán.

3.1 Procházka

Procházka je entita, kterou lze definovat určitými vlastnostmi. Mezi ně patří skupina atributů (tagy), doplňkové informace a trasa zobrazená na mapě spolu se zájmovými body. Právo zobrazit procházku má kterýkoliv návštěvník, právo editovat má její autor a také administrátor (popis uživatelských rolí je níže).

3.1.1 Atributy trasy

Atribut trasy neboli tag, je vlastnost procházky, kterou lze definovat jedním slovem. Těchto atributů může mít každá procházka teoreticky neomezené množství. Slouží především pro vyhledávání. Registrovaný uživatel má na výběr z již dostupných atributů, nebo může vytvořit nový. Ten však musí být ve schvalovacím procesu schválen administrátorem služby. Do té doby už ale může být využíván i jinými uživateli. Pokud bude nový atribut shledán jako nevhodný, může být ze všech aktuálně přidělených procházek odstraněn. Po zavedení služby mohou být nastaveny jiné parametry schvalovacího procesu atributů, a to např. takové, že neschválený atribut nebude při vyhledávání brán v potaz a bude k dispozici jen samotnému autorovi.

3.1.2 Doplňkové informace

Procházka navíc obsahuje informace jako je:

- popis,
- trasa samotná vyznačená na mapě,
- fotogalerie.

3.1.3 Trasa na mapě

Trasa samotné procházky je vyznačena na mapě formou bodů, které jsou spojeny dohromady barevnou čarou a tvoří tak cestu. Každý bod představuje jednu GPS souřadnici. Navíc jsou zde vyznačeny i zájmové body, a to formou ikon. Na mapě se zobrazují i doporučovaná místa komerčních subjektů z nejbližšího okolí procházky.

3.1.4 Zájmové body

Zájmový bod reprezentuje jedna ikona dle typu bodu a titulek. Po rozkliknutí bodu se zobrazí detailnější popis s komentářem nebo odkazem na webovou stránku daného bodu. Pokud přidáme do mapy některý ze zájmových bodů, přidá se i do přehledu vlastností procházky spolu s krátkým popisem (titulkem bodu). Výchozí zájmové body jsou typu:

- obchod,
- odpočívadlo,
- hotel,
- parkoviště,
- restaurace,
- socha,
- výhled do krajiny,
- jiné zajímavé místo.

Jejich počet ale není pevný ani konečný. Postupem času může k tomu oprávněný uživatel vytvořit nový zájmový bod a přiřadit mu ikonu, vždy s ohledem na četnost používání stejnojmenného atributu (tagu).

3.1.5 Sdílení

Každý návštěvník služby, ať registrovaný nebo ne, může procházku doporučit svému známému prostřednictvím sociálních sítí Facebooka a Google Plus.

3.2 Přehled procházek

Uživateli budou primárně nabízeny procházky z jeho nejbližšího okolí. Dále může tyto procházky filtrovat podle regionu, ve kterém se nachází, zobrazovat pouze procházky určitého autora nebo procházky vlastní. Uživatel může procházky vyhledávat pomocí klíčových slov, jména a shody v popisu procházky. Každý přehled procházek je realizován pomocí dvou technik, a to tabulky a mapy. Tipy na procházku jsou realizovány pomocí samostatných elementů.

3.2.1 Přehled v tabulce

Je realizován pomocí tabulky o několika sloupcích, kde každý řádek představuje jednu procházku. Spolu s malým náhledem fotografie (pokud je k dispozici) je zobrazován název procházky spolu s krátkým popisem, délkou trasy, názvem regionu a v případě přihlášeného uživatele i autor procházky. Všechny výsledky jsou stránkované, s možností řazení dle jednotlivých sloupců a zobrazení libovolné stránky.

3.2.2 Mapa

Všechny procházky zobrazené v tabulce dle zadaného filtru jsou taktéž zobrazeny na mapě formou bodů. Každý bod představuje jednu procházku a po kliknutí na tento bod je zobrazen malý náhled. Z něj je možné přistoupit na detail procházky.

3.2.3 Tipy na procházky

Třetím typem přehledu je zobrazení tipů. Je realizován pomocí samostatných elementů, kdy jeden element vždy představuje jednu procházku. Obsahuje jméno procházky, její krátký popis, případně fotografii, délku trasy a odkaz na detail procházky. Tyto elementy jsou využity jak v levém postranním panelu, tak na úvodní stránce pro zobrazení tipů procházek dle aktuální polohy.

3.3 Uživatel

Každý návštěvník služby se zároveň stane uživatelem, kterému jsou přidělena základní uživatelská práva. Z pohledu zobrazení procházek nemá neregistrovaný návštěvník žádná omezení. Ta nastávají až v případě vytváření procházek a jejich organizování, které je přístupné pouze pro registrované uživatele. Tím se návštěvník stane po vyplnění registračního formuláře a jeho odeslání. Registrace na webové službě je dobrovolná a bez finančních poplatků.

3.3.1 Údaje o uživateli

Při registraci je nutné zadat unikátní uživatelské jméno, které slouží pro jeho vnitřní identifikaci, jméno uživatele, které se zobrazí u všech příspěvků, email a region, ve kterém uživatel bydlí. Dále popis uživatele, který je jeho charakteristikou a slouží jako dodatečné informace v případě návštěvy profilu uživatele.

3.3.2 Role

Každému návštěvníkovi/uživateli je přidělena jeho role, a to již při prvním návštěvě služby, což je role běžného návštěvníka. Role registrovaného návštěvníka se dále dělí.

- Běžný návštěvník - má právo procházky prohlížet a dále je sdílet.
- Registrovaný návštěvník - uživatel
 - Běžný registrovaný uživatel - má právo procházky vytvářet, editovat jím vytvořené a přikládat k nim fotografie. Procházky ostatních autorů může přidávat do svého seznamu oblíbených.

- Správce regionu - má navíc právo editovat a odstraňovat duplicitní procházky vytvořené v regionu, jehož je správcem.
- Administrátor - má oproti správci právo spravovat registrace uživatelů, měnit jejich role, aktivovat a deaktivovat uživatelské účty. Dále má právo schvalovat nové atributy procházek a odstraňovat je, organizovat atributy do skupin a měnit jejich váhu pro vyhledávání. Má právo vytvářet nové typy zájmových bodů. Má právo zobrazovat statistiky vyhledávání, přihlašování a přehledy zaznamenávaných akcí.

3.4 API

Webová aplikace implementuje aplikační rozhraní pro spolupráci s mobilní aplikací. Je typu klient - server, kdy webová služba naProcházku.cz je server a aplikace na mobilních zařízeních jsou klienty. Požadavky jsou zasílány formou HTTP Request a odpovědi HTTP Response. Využívá se zapouzdření dat do formátu JSON a kódování znaků UTF-8.

3.4.1 Komunikační protokol

Rozhraní aplikačního protokolu je definováno předem. Obsahuje metody pro přihlašování uživatelů, vytváření nových uživatelských účtů a jejich pomocné funkce, metody pro načítání seznamů procházek s jejich základními údaji a metodu pro načtení celého detailu procházky, metody pro vytváření nových procházek včetně všech atributů a souřadnic trasy. Poskytuje metody pro práci se souřadnicemi, s atributy a se zájmovými body procházek, metody pro hledání v procházkách na základě vstupních dat. Poskytuje metody pro zaznamenání informací o polohách uživatelů webové i mobilní aplikace. Je implementováno na způsob HTTP požadavků typu POST a odpovědi jsou zasílány spolu s definicí odpovědi v JSON formátu. Všechny požadavky zachovávají stejný formát vstupních dat a očekávají příslušnou odpověď. Všechny definice metod a jejich odpovědi jsou popsány v příloze B.

3.5 Uživatelské rozhraní

Uživatelské rozhraní je neméně důležitou součástí webové aplikace jako funkce služby samotné. Jakýsi standart uživatelského rozhraní položili dvě významné osoby moderního návrhu už. prostředí, každý s odlišným přístupem. První z nich je Steve Krug a druhý Jakob Nielsen.

Po počátečním návrhu aplikace budeme věnovat převážnou část práce právě návrhu uživatelského rozhraní.

3.5.1 Steve Krug

Jak definovat použitelnost webu dobře popsal Steve Krug ve své knize *Webdesign: Nenutíte uživatele přemýšlet!*, která je o vztahu člověka k počítačům a webové stránce. Dobrý web by neměl dát šanci člověku dlouze přemýšlet o akci, kterou chce udělat. Měla by jít provést jednoduše a přímo. Pro něj použitelnost znamená, "že něco dobře funguje a že osoba s průměrnými (ba dokonce podprůměrnými) schopnostmi a zkušenostmi může používat určitou věc (...) k účelu, ke kterému je určena, aniž musela být beznadějně mučena." [3]. Při minimálním vynaložení úsilí by měl uživatel plně dosáhnout svého cíle. Jeho pravidla znějí:

- Uživatel obsah webové stránky nečte, ale prohlíží.
- Lze snadno odvést jeho pozornost výraznými nepatřičnými prvky na stránce.
- Pokud má na výběr mezi dvěma způsoby, obvykle volí ten první.
- Pokud najde způsob, který funguje, jiný už nehledá.
- Zvýraznit objekty, se kterými lze pracovat, jako je navigace a odkazy.
- Stránka by měla být hierarchicky rozdělena a toto využívat na všech částech webu stejně.

3.5.2 Jakob Nielsen

Naopak Nielsenovo tvrzení "použitelnost popisuje, jak rychle se člověk dokáže naučit určitou věc používat, jak efektivně ji využívá, jak je pro něj tento proces učení zapamatovatelný" [4]. Uživatelé jsou zvyklí používat web již zažitým způsobem z jiných webů a podobnou strukturu očekávají v naší aplikaci. Upřednostňuje konvenci před inovacemi. Pokud se odchýlíme, naši návštěvníci se na webu nevyznají a opustí ho. Jeho pravidla:

- Na první stránce představit účel i za pomoci krátkého sloganu.
- Jasně definovat odkazy a odhalit obsah stránky.
- Navést uživatele k cíli, za kterým web navštívil.
- Design stránky je pouze k obohacení, nikoli definování struktury stránky.
- Přehledná navigace s možností návratu.

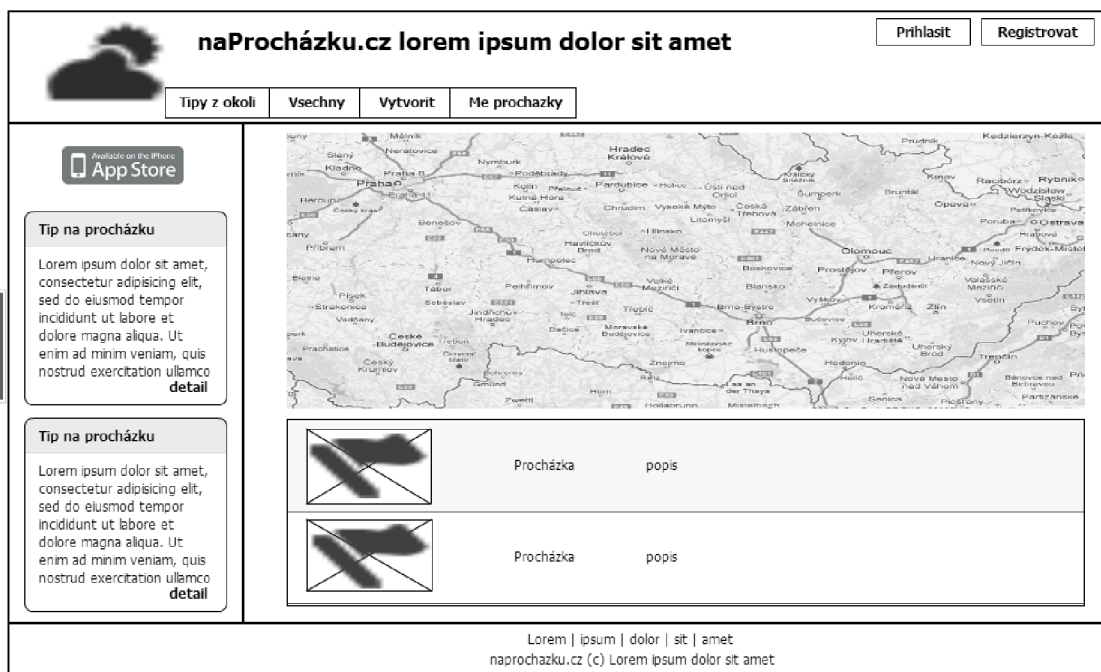
3.5.3 Vlastní shrnutí

Webové stránky služby zachovávají již zažité postupy, neučí uživatele novým složitým věcem. Jsou přehledné a prováděné akce jsou předem čitelné. O všech procesech, které uživatel sám inicioval, je srozumitelně a vždy informován. Náповěda je provedena formou titulkování tlačítek a odkazů. Veškeré chybové zprávy jsou ošetřeny a uživateli zobrazeny v patřičném znění. Šířka webových stránek aplikace je navržena pro monitor s vyšším rozlišením, protože jak vyplývá ze statistik měření¹ rozlišení přistupujících na webové stránky celého světa, většina uživatelů má dnes rozlišení monitoru větší jak 1024x768 pixelů.

3.5.4 Návrh

Úvodní stránka je rozdělena do 5 základních částí (obr. 6.1), kde největší zaujímá hlavní obsah. V horní části se nachází logo spolu se sloganem, které slouží zároveň jako odkaz na tuto úvodní stránku. V pravém rohu je blok, který nabízí přihlášení či registraci uživatele, a v případě již přihlášeného uživatele jsou zde tlačítka pro správu účtu.

¹http://www.w3schools.com/browsers/browsers_resolution_higher.asp



Obrázek 3.1: Návrh úvodní obrazovky

3.6 Podporované webové prohlížeče

Existuje obrovské množství internetových prohlížečů různých verzí. Novější vydání jednotlivých prohlížečů obsahují opravy chyb předchozích vydání, ale také nové funkce. Funkční rozdíly mezi jednotlivými prohlížeči mohou způsobit rozdílné chování a zobrazení některých stránek. Pro zajištění správného zobrazení a funkčnosti naProcházku.cz, je aplikace testována na čtyřech nejběžnějších webových prohlížečích. Mezi ně patří Google Chrome (aktuální verze 26), Microsoft Internet Explorer (verze 10, 9, 8), Mozilla Firefox (aktuální verze 20) a Opera (aktuální verze 12.4). Pro správné chování aplikace je nutné mít povolený JavaScript, což většina uživatelů má. Dále je nutné mít ze standartu HTML5 podporu geolokačního rozhraní, což splňují všechny prohlížeče, mimo IE8 se zastoupením 6.5%², ve většině dnes používaných prohlížečů.

3.7 Určení polohy

Aby návštěvníkovi mohly být zobrazeny tipy na procházku z jeho okolí, je nutné nejdříve zjistit jeho polohu. K tomu je využito geolokačního rozhraní webového prohlížeče. Jeho API je specifikováno v dokumentu organizace W3C *Geolocation API Specification*[1]. Po zjištění polohy jsou návštěvníkovi tipy na procházky nabídnuty. V případě, že zjištění polohy nepovolí, je na tuto skutečnost upozorněn a z důvodů neznámé polohy nemůže dojít k načtení žádných tipů.

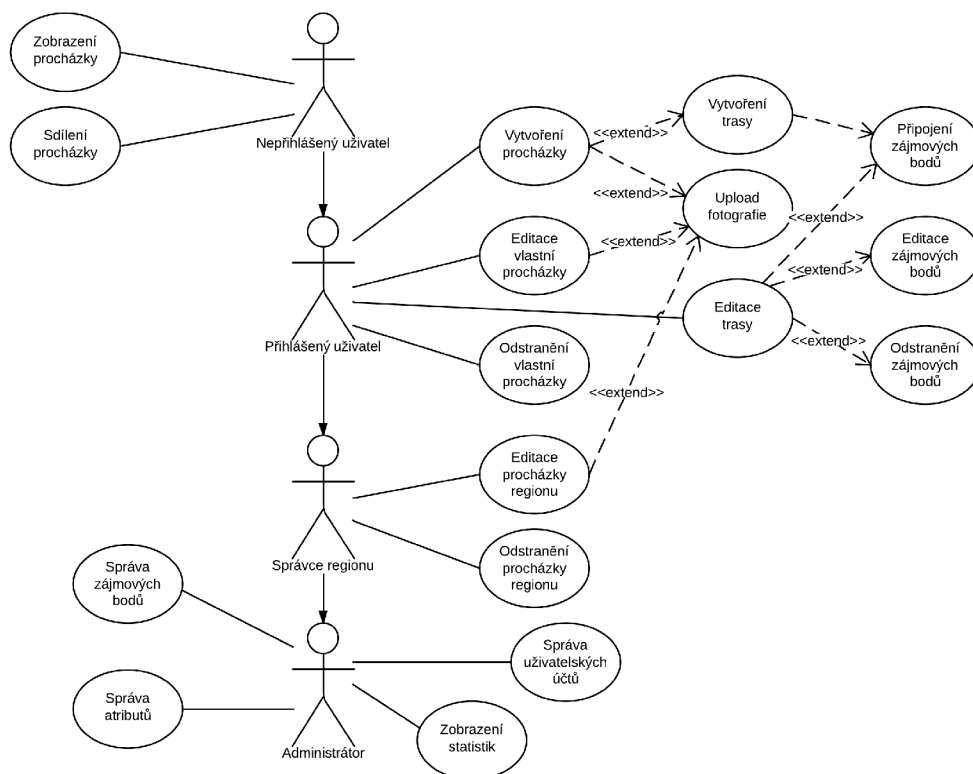
²http://www.w3schools.com/browsers/browsers_explorer.asp

3.7.1 Ochrana soukromí

Otázkou sledování polohy z hlediska ochrany soukromí se zabývá práce *Geolokace z hlediska ochrany soukromí* Hany Sloupenské[5]. Vyplývá z ní, že ohrožení soukromí v důsledku znalosti polohy je nebezpečné, ale pouze v kombinaci s jinými službami, metodami a zařízeními. Geolokace by byla pouze dílkem v této skládačce. To je i můj názor z hlediska bezpečnosti záznamu polohy, kdy se i tento záznam uchovává pro statistiky a další optimalizace aplikace. Při využití jsou data vždy anonymní, jsou zachovávány všechny aspekty ochrany soukromí, aby nedocházelo k jejich porušování.

3.8 Diagram užití

Use-case diagram (obr. 3.2), neboli diagram užití, je určen k definici chování systému, aniž by odhalil jeho vnitřní strukturu. Zobrazuje dynamické (funkční) struktury systému z pohledu uživatele. Zobrazuje, jaké akce může uživatel vykonat, za jakých podmínek a jaké akce na ně mohou navazovat. Akce vychází z uživatelských práv rolí, které jsou definovány hierarchickou dědičností. Proto nejvyšší role má stejná práva jako role, od které dědí, a je navíc rozšířena o nová práva.

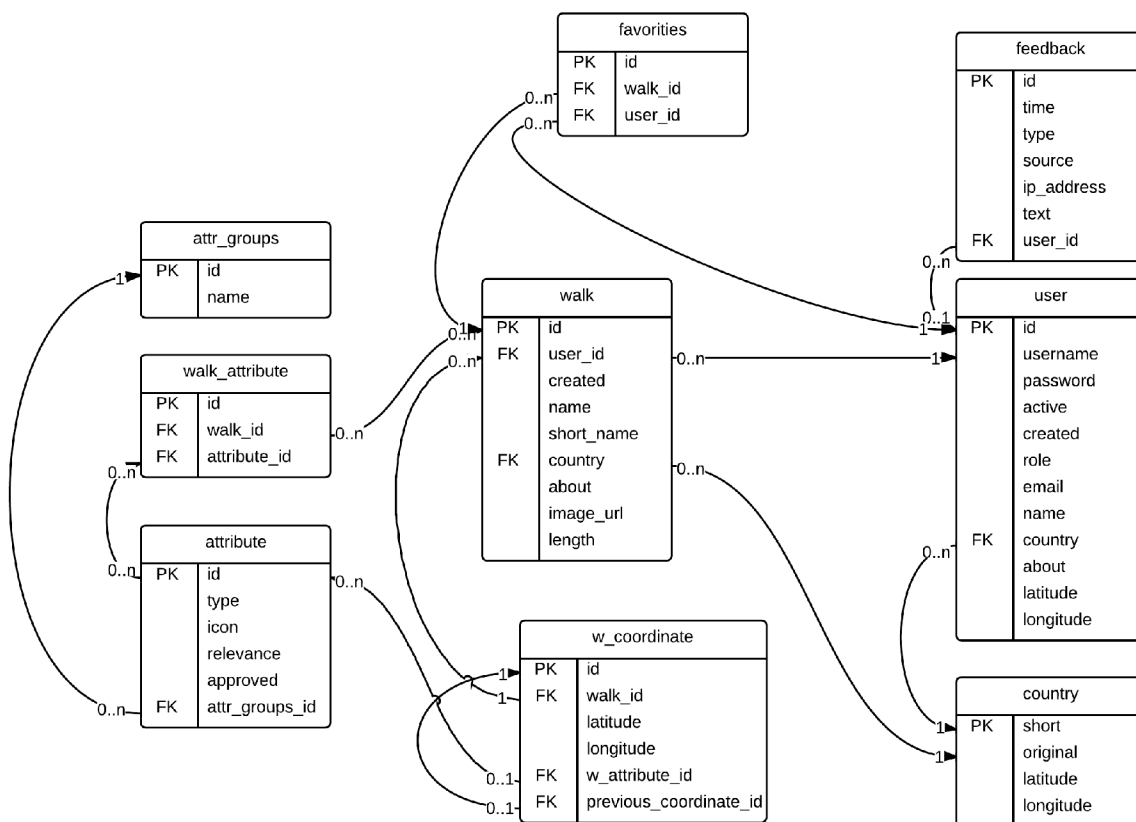


Obrázek 3.2: Navrhovaný diagram užití aplikace

3.9 Databáze

K popisu systému při analýze požadavků je použit ER diagram (obr. 3.3), na kterém je zobrazen celkový model databáze včetně vzájemných vazeb. Jde o fyzický návrh a je proveden stylem shora dolů.

Z hlediska náročnosti návrhu je důležité přihlížet k tomu, že se pracuje s velkým množstvím souřadnicových dat. Je zvolen návrh, kde každý bod, souřadnice, má v tabulce vyhrazen jeden řádek. Pro uložení zeměpisné délky a šířky je zvolen datový typ double. Pro sestavení trasy je také nutné zachovávat pořadí bodů, proto je ke každé souřadnici přiřazen i identifikátor (cizí klíč) předchozího bodu. Je využito cizího klíče. Je také ukládána informace o tom, zda se jedná o bod trasy, či zájmový bod. Ve zbylém návrhu nenastala žádná neobvyklá situace.



Obrázek 3.3: Návrh databáze pomocí ER diagramu

Kapitola 4

Použité technologie

V této kapitole jsou popsány všechny hlavní technologie, které jsou ve službě naProcházku.cz použity.

4.1 Nette Framework

Zprvu jsem chtěl webovou službu implementovat pomocí vlastního frameworku vytvořeného pouze pro tuto aplikaci. Po první implementaci jsem však od vlastního frameworku opustil a vsadil jsem na už ověřený Nette Framework, abych se mohl více věnovat implementaci služby než frameworku samotnému a aplikace se stala pro mne lépe proveditelnou.

Nette (použitá verze 2.0.8 stabil) je PHP framework českého původu napsaný komunitou kolem Davida Grudla určený pro vývoj webových aplikací. S licencí BSD¹ se řadí mezi open-source i v komerčních projektech. Jednou z hlavních výhod a důvodem, proč jsem si tento framework vybral, je dobrá podpora pomocí oficiálního fóra, kde je již mnoho problému spojených s vývojem aplikace vyřešeno a popsáno. Dokumentace není úplná, ale v základní verzi dostačující.

4.1.1 Vlastnosti

Nette Framework jsou PHP knihovny s otevřeným kódem, které ulehčují programování aplikace. Je využito objektového programování. V Nette Frameworku je kladen důraz na zabezpečení a ochranu aplikace před útoky, moderní technologie, výkon frameworku a na ladící nástroje použitelné během vývoje aplikace. Podporuje technologie jako je HTML5, AJAX a pozitivně přispívá k SEO. Je vhodný pro aplikace typu CMS (content management system), což naše aplikace je.

4.1.2 MVC

Architektura MVC (Model-View-Controller) nám zajišťuje správný vývoj, přehledný a udržitelný kód. Tento vzor poprvé popsal již v roce 1979 Trygve Reenskaug. Rozděluje nám aplikaci do tří samostatných částí - datový model, uživatelské rozhraní a řídicí logiku.

¹<http://nette.org/cs/license>

Model Reprezentuje informace a data a měl by být co nejméně závislý na samotné databázi. Přechod na jiný typ databáze by tak neměl být problém. Poskytuje pevně dané rozhraní. Obsahuje veškerou aplikační logiku. Jakákoliv akce uživatele je i akcí modelu. Model také zajišťuje validaci dat před jejich zápisem do databáze.

View Prezentační vrstva zajišťuje vykreslení výsledků modelu. Nette Framework používá šablonovací systém, kde jej představují šablony typu *latte* nebo *phtml*.

Controller Neboli řadič přijímá požadavky uživatele a volá příslušnou akci v modelu. Ve výsledku předá data šabloně k vykreslení. V Nette Framework controllery představují jednotlivé presentery, ke kterým přísluší šablony.

4.1.3 Podpůrné integrované nástroje

Nette už v instalaci obsahuje několik podpůrných nástrojů. Mezi ně patří Laděnka, DebugBar a externí aplikace Adminer².

Laděnka Pokud v aplikaci během vývoje nastane neošetřená výjimka, chyba, je na tuto skutečnost programátor upozorněn právě tímto ladicím nástrojem, a to je Laděnka. Zobrazí, co za chybu se stalo, v jakém místě kódu a na jakém řádku, zobrazí aktuální prostředí webového prohlížeče. Toto nám dovoluje odladit všechny chyby před vydáním produkční verze. V té se už Laděnka nezobrazuje a všechny chyby jsou řádně logovány a slouží pro pozdější analýzu chyb.

DebugBar Je dalším užitečným pomocníkem při vývoji aplikace. Je to plovoucí panel přímo v aplikaci a zobrazuje nám informace o času načtení webové stránky, o počtu přenesených dat a v případě použití routovacích pravidel i to, jaké pravidlo bylo pro upravení url adresy použito. Poskytne nám informace o všech provedených databázových dotazech včetně času jejich zpracování. Tento panel jde rozšířit o přídatné moduly, jako v mém případě o mnou upravený TimerPanel, který počítá již strávenou dobu práce na projektu.

Adminer Je podobný nástroj jako zažitý standart pro správu MySQL databází phpMyAdmin³. Avšak má oproti němu několik nesporných výhod. První je snadná použitelnost, a to i proto, že se jedná pouze o jeden soubor, který se nahraje spolu s aplikací na server (hosting). Druhá výhoda je dobrá podpora cizích klíčů, které se vytváří a spravují velmi snadno. Další výhodou je přehlednost celé aplikace a, v případě použití omezené verze Adminer Editor, je práce s ní vhodná i pro méně databázově znalé uživatele.

4.2 Google Maps API

Pro zobrazení tras na mapě, její vytvoření a editace bylo potřeba zvolit vhodnou knihovnu pro vykreslování dat s příslušnými mapovými podklady. Výběr jsem zúžil na dvě u nás nejznámější, a to Google Maps API a API Mapy.cz.

²<http://www.adminer.org/>

³<http://www.phpmyadmin.net/>

4.2.1 API Mapy.cz v4

Mají dobře zpracované a často aktualizované mapové podklady pro Českou republiku, zejména turistické části mapy. Neposkytují však vyspělé API pro práci s těmito mapami a i podpora, jak z hlediska dokumentace, tak diskuzních fór programátorů, je u druhé zmiňované lepší. Naproti tomu však nejsou nastaveny žádné limity při používání bezplatné verze.

4.2.2 Google Maps API v3

Poskytuje mapové podklady celého světa. Aplikační rozhraní javascriptové verze je na vyspělé úrovni a poskytuje programátorovi mnoho možností, jak jej využít. Při detailnějším přiblížení mapového podkladu se místo 2D mapy zobrazí Street View⁴, které je pro službu naProcházku.cz výhodné i z hlediska lepší navigace v neznámém prostředí z pohodlí webové stránky. Omezení pro neplacenou verzi je 25 tisíc přístupů za den přes toto rozhraní. To však z pohledu bakalářské práce nepovažuji za omezení, a proto jsem se rozhodl pro tuto variantu. Je využito Google Maps JavaScript API v3⁵. Poskytuje dále metody pro zaznamenávání trasy na mapě, měření její délky, přidávání zájmových bodů a zobrazování polohy uživatele.

4.2.3 API Access key

Pro využití aplikačního rozhraní je nutné se nejdříve do této služby zaregistrovat a získat tak přístupový klíč, který se uvádí při inicializaci knihoven mapy. Opravňuje nás k využití všech zaregistrovaných služeb. V našem případě tedy Google Maps API v3 pro práci s mapou, Analytics API pro sledování chování uživatelů a Google+ API pro propojení služby naProcházku.cz s její stránkou na Google+.

4.3 Databáze InnoDB

Všechna data, se kterými aplikace a dále mobilní aplikace pomocí API pracuje, jsou uložena na serveru v databázovém systému MySQL formátu InnoDB. Engine InnoDB jsem zvolil proto, že podporuje cizí klíče a transakce. Těchto vlastností využívá webová služba ve velké míře, transakce při práci se souřadnicemi a cizí klíče pro provázanost záznamů v tabulkách.

⁴<http://www.google.cz/streetview>

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

Kapitola 5

Implementace

V této kapitole je popsána implementace programu se všemi problémy, které se při vývoji vyskytly. Je vysvětlena celá problematika vykreslování tras a zájmových bodů, jejich zaznamenávání na mapu a následná editace. Je přiblížena i technika určování polohy pomocí rozhraní webového prohlížeče, dále práce s atributy procházek a vyhledávání s jejich pomocí.

5.1 Použitý software a vývojové prostředí

5.1.1 NetBeans

Pro vývoj aplikace v PHP frameworku Nette jsem použil prostředí NetBeans IDE s rozšířením pro PHP. Je vhodné také použít doplněk pro usnadnění práce s Nette, a to PHP Nette Framework plugin¹. Dále je v tomto prostředí možno editovat soubory kaskádových stylů a html kód.

5.1.2 WampServer

Jde o vývojové prostředí pro Windows, ve kterém je možno vytvářet webové aplikace. WampServer² obsahuje vlastní Apache server, na kterém je nainstalováno PHP verze 5.4, MySQL verze 5.5. Pro snazší správu databází je doplněn o PHPMyAdmin a SQLiteManager a poskytuje intuitivní rozhraní pro snadnou konfiguraci lokálního serveru.

5.1.3 SVN

K udržení přehlednosti vývoje jsem byl nucen udržovat historii jednotlivých verzí aplikace. Využil jsem proto systém pro verzování zdrojových kódů SVN s repozitářem na školním serveru. To mi umožnilo, v případě řešení problému, který vedl do slepé uličky, vrátit se k funkční verzi aplikace a rozhodnout se pro jiné řešení.

5.2 Zpracování akcí a zobrazení v HTML

Zpracování všech akcí je rozděleno do několika tříd - presenterů. Každý se stará o určitý typ.

¹<http://plugins.netbeans.org/plugin/32720/php-nette-framework>

²<http://www.wampserver.com/en/>

Pro zobrazení úvodní stránky je HTTP požadavek přeložen Routerem a následně je volán `HomepagePresenter` s příslušnou akcí. V tomto případě `actionDefault` bez přidavných parametrů. Ta vykoná požadovaná volání metod, která jsou v těle metody akce. Poté dojde k předání dat šabloně. Jméno šablony se shoduje (pokud nedojde k přesměrování na jinou šablonu pomocí metody `setView()`) s jménem akce. Je vykonána metoda `renderDefault` a v ní jsou předána data do proměnných šablony `default`. Pokud šablona obsahuje nějakou komponentu či formulář, musí být v presenteru příslušná metoda pro vytvoření této komponenty. Příklad názvu metody je `createComponentWalkTips` a tato požadovanou vytvořenou komponentu `WalkTips` vrátí.

Aplikace je rozdělena do presenterů:

- `BasePresenter` - je abstraktní společný předek všech presenterů v aplikaci. Provádí se v ní inicializace společné komponenty a načítání dat, která jsou společná pro všechny presentery.
- `HomepagePresenter` - stará se o úvodní stránku a příslušné stránky *O projektu*, *Soukromí*, *Links* a *Kontakt*.
- `SignInPresenter` - je použit při přihlašování uživatelů a jejich registraci.
- `UserPresenter` - slouží pro zobrazení detailu uživatele, jeho editaci a změnu hesla.
- `WalksPresenter` - je určen pro zobrazení seznamu procházek, jejich filtrování a vyhledávání.
- `WalkDetailPresenter` - má na starost zobrazení detailu procházky, vytvoření nové a její editaci.
- `AdminPresenter` - je určen pro správu uživatelů, atributů, zobrazování přehledů a statistik.
- `APIPresenter` - jsou v něm implementovány metody pro komunikaci pomocí aplikačního rozhraní. Přijímá POST HTTP požadavky a odpovídá na ně. Metody jsou blíže popsány v příloze **B**.
- `ErrorPresenter` - ošetřuje zobrazení chybových stavů, zobrazuje chybové stránky.

5.3 Google Maps

Nejdříve je nutné načíst knihovny Google Maps, a to vložením skriptu do hlavičky stránky (ukázka 5.1). Je nutné připojit parametr pro požadovanou verzi a klíč pro přístup k api. Načítají se knihovny `drawing`, `geometry` a `weather`.

```
<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?
  v=3.exp
  &key=AIzaSyC_U1ignrSiNTSKYysPMBOLsTW8-xxTxTk
  &sensor=true
  &libraries=weather,drawing,geometry">
</script>
```

Ukázka kódu 5.1: JavaScriptový kód v hlavičce dokumentu.

Poté už stačí v těle dokumentu vytvořit element s id `map_canvas` a reagovat na event `load` celé stránky. Jakmile se načte, je volána metoda `initialize`, kde dojde k vytvoření mapy, nastavení jejího středu a přiblížení. Jelikož metoda pro setování mapy nahradí veškerý obsah v tomto použitém elementu, je zde vložen obsah indikující načítání mapy. To je zejména vhodné pro uživatele s pomalejším internetovým připojením, kde se jím zobrazí informace o probíhající akci, a to inicializaci mapy.

5.3.1 Zobrazení detailu procházky na mapě

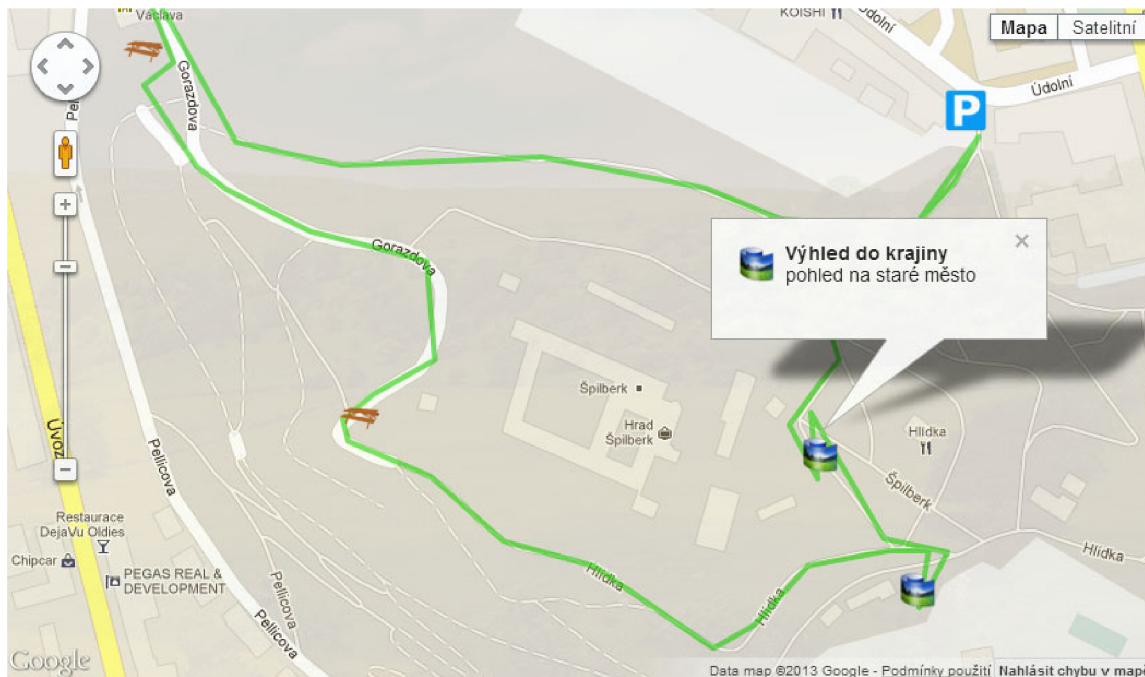
Z presenteru jsou dostupná data o bodech, která jsou uložena v poli. Procházení pole je usnadněno makrem `foreach` šablonovacího systému Latte. Javascriptový kód je vygenerován pomocí tohoto cyklu (ukázka 5.2). Např. pro zájmový bod detailu procházky jsou důležité informace o typu bodu (z něho je odvozena zastupující ikona), souřadnice na mapě a popisek. Je provedeno vytvoření informačního okna s detailní informací, které je zobrazeno při označení tohoto bodu. Podobné je to s vykreslením trasy procházky. Pomocí makra `foreach` je vytvořeno javascriptové pole se souřadnicemi. To je pak použito při inicializaci objektu `google.maps.Polyline`. Výsledkem je vykreslení tohoto poligonu na mapě.

```
var mapCenter = new google.maps.LatLng(  
    {$coordinates->latitude},  
    {$coordinates->longitude}  
    );  
var map = new google.maps.Map(document.getElementById('map_canvas'), {  
    center: mapCenter,        // řsted mapy  
    zoom: 15,                // řížípiblen  
    mapTypeId: google.maps.MapTypeId.ROADMAP // typ mapy  
});  
//souradnice mapy - trasa  
var mapCoordinates = [];  
{foreach $walkCoordinates as $coordinate}  
    mapCoordinates.push(  
        new google.maps.LatLng(  
            {$coordinate->latitude},  
            {$coordinate->longitude}  
        )  
    );  
}{/foreach}  
// vykresleni polygonu  
var heatmap = new google.maps.Polyline({  
    path: mapCoordinates,    // řsouadnice spojnic  
    strokeColor: "#1dd300", // barva spojnice  
    strokeOpacity: 0.6,     // viditelnost  
    strokeWeight: 4         // šířka  
});  
heatmap.setMap(map);
```

Ukázka kódu 5.2: Kód pro zobrazení mapy a vykreslení trasy na ní.

5.3.2 Zaznamenání trasy na mapu

Jiná situace je v případě zaznamenávání této trasy. Uživateli je umožněno tahem a klikáním myši tuto trasu na mapě nakreslit. K tomu je právě využita knihovna `drawing`. Na mapě je



Obrázek 5.1: Vykreslená trasa na mapě spolu s detailem zájmového bodu

zobrazen příslušný panel s volbou nástrojů. Mezi ně patří *ručička* pro snadnější posun na mapě, dále nástroj pro vkládání zájmových bodů a nástroj pro kreslení trasy.

Kreslení trasy Ve funkci pro inicializaci mapy je přidán listener na event `polylinecomplete`. Ten nám zajistí, že při každém dokončení kreslení trasy je zavolána jeho anonymní funkce, která má jako parametr právě vytvořený polygon s trasou. Z tohoto polygonu je možné opět získat pole se souřadnicemi bodů. Statická metoda `computeLength()` dokáže spočítat ze zadaných souřadnic délku trasy v metrech. V aplikaci je však použit formát v kilometrech s jedním desetinným číslem. Je proto vytvořena javascriptová metoda (ukázka 5.3), která provede správné zaokrouhlení a vrátí délku trasy v požadovaném formátu. Jelikož ale také zároveň existuje možnost úpravy právě vytvořené trasy procházky,

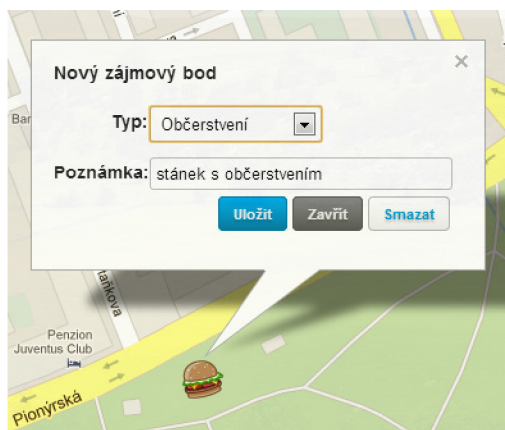
```
function getLength(mapCoordinates) {
    return Math.round(
        google.maps.geometry.spherical.computeLength(mapCoordinates)/100
    )/10;
}
```

Ukázka kódu 5.3: Metoda pro vypočítání délky trasy ze zadaných souřadnic.

jsou nastaveny listenery na editaci polygonu, případně přidání nového bodu do trasy či jeho odstranění. Po každé této akci dochází k přepočítání souřadnic trasy a její délky. Délka je vždy aktuálně zobrazena v poli formuláře nad mapou. V případě, že by uživatel chtěl zadat druhou trasu na mapu, je na tuto skutečnost upozorněn a vyzván, aby již vloženou trasu editoval nebo ji odstranil. K odstranění je implementovaná javascriptová metoda `deleteTrack()`, která aktuální trasu odstraní a umožní nakreslit novou. Získané souřadnice z mapy jsou převedeny do JSON formátu, a to pomocí vestavěné ja-

vascriptové funkce `JSON.stringify()`. Takto získaná hodnota je přiřazena do skrytého pole ve formuláři pro editaci trasy. Jakmile dojde k potvrzení změn provedených na mapě, je tento formulář odeslán do `WalkDetailPresenter`, kde je hodnota z tohoto skrytého pole zpátky převedena na odpovídající datovou strukturu.

Zájmový bod Podobná situace je ve vytvoření, editaci a odstranění zájmového bodu. Je nastaven listener na událost `markercomplete`. Anonymní funkci je předán vytvořený bod, který poskytuje informace o souřadnicích. Dále jsou tomuto vytvořenému bodu nastaveny listenry pro události `click` (pro zobrazení informačního okna), `dragend` (v případě přetažení bodu do jiného umístění), vytvořeno informační okno s detailem bodu a toto okno je následně zobrazeno (obr. 5.2). Jsou zobrazena tři tlačítka pro možnost uložení změny, ignorování změny a odstranění zájmového bodu z mapy. Informace o tomto bodu a ostatních



Obrázek 5.2: Okno pro vytvoření nového zájmového bodu na mapě.

vytvořených bodech, jsou uloženy v javascriptovém poli. Při ukončení práce se zájmovými body je toto pole opět převedeno do JSON formátu a uloženo ve skrytém poli formuláře. Po ukončení editace trasy a zájmových bodů jsou data ze skrytého pole dekodována a uložena v databázi.

Editace V případě editace trasy a zájmových bodů je situace podobná. Pouze jsou získaná data z databáze zpracována pomocí maker šablonovacího systému, jsou vložena do mapy a listery na události editace a změny jsou připojeny následně.

5.4 Atributy

Atribut, neboli tag, je určitá vlastnost procházek. Pokud chce uživatel přiřadit nějaký tag k procházce, je mu nabídnuto z již existujících tagů. Pokud však zadá neexistující tag, je vytvořen nový. V případě editace vlastností procházky dochází k předání vložených tagů. Je proveden výběr tagů dané procházky z databáze. Pokud již nějaký přiřazený tag není obsažen v předaném poli tagů, je z databáze vymazán. V opačném případě, kdy v předaném poli je nějaký tag, který není k procházce v databázi přiřazen, je provedeno vyhledávání z dostupných, již existujících tagů. Pokud nebyl nalezen, je vytvořen a k procházce přiřazen. Takto vytvořený tag má v databázi příznak neschváleného tagu.

5.4.1 Správa atributů

Každý atribut má určenou svoji váhu, ke které se přihlíží při vyhledávání. Jsou organizovány do skupin. Každý tag může být nejvýše v jedné skupině. Pomocí správce atributů je možné je takto organizovat. Přiřazovat jim příslušnou váhu, přidávat do skupin či je z nich odebírat, schvalovat je a odstraňovat. U každého atributu je také počet použití, který je ale dostupný pouze v tomto správci. Presenter `AdminPresenter` předá šabloně pole atributů a pole skupin atributů. V šabloně je pomocí makra provedeno vypsání atributů do tabulky. Roztřídění atributů do skupin a přiřazení akcí tlačítkám provádí JavaScript. Všechny akce mimo vytváření nového atributu a nové skupiny jsou prováděny pomocí Ajaxu a metod v `APIPresenter`. Jejich správné dokončení je kontrolováno pomocí návratového kódu a zprávy o úspěšnosti provedené změny. Poté dojde ke změně elementu, který je zobrazen v šabloně. Pokud dojde k odstranění skupiny atributů, atributy v ní obsažené jsou z této skupiny vyjmuty, ale atributy jako samotné zůstávají.

5.5 Zájmové body

V případě vytváření procházkové trasy je situace jednoduchá, všechny zájmové body jsou přiřazeny k dané procházce. Při editaci je však nutné, aby celá transakce byla atomická a aby došlo ke konzistenci dat. Je nejprve nutné, aby došlo k získání přiřazených zájmových bodů k dané procházce v databázi. Následně jsou porovnávány s předaným polem zájmových bodů z editačního formuláře. Ty zájmové body, které již v databázi jsou, porovnáme, zda nebyly pozměněny. Pozměněné body jsou aktualizovány, nové jsou přidány a ty, které v předaném poli nejsou, z databáze odstraníme. Pokud se však při některé akci nepovede operaci provést, je proveden `rollback` celé transakce. V takovém případě je uživatel na tuto skutečnost upozorněn a je požádán o opětovné provedení požadovaných změn zájmových bodů.

5.6 Určení polohy

Každý webový prohlížeč s podporou HTML5 a geolokačního rozhraní umožňuje získat informace o poloze návštěvníka webu. Tuto možnost sledování však musí návštěvník po vyzvání povolit.

Webový prohlížeč vždy vybere ten způsob určení polohy, který umožňuje nejlepší přesnost zaměření. Poloha se určuje těmito způsoby:

- Pomocí GPS (Global Positioning System),
- IP adresy s porovnáním údajů v databázi,
- WiFi a Bluetooth MAC adresy s porovnáním údajů v databázi,
- GSM / CDMA Cell ID, tj. pomocí BTS bodů, ke kterému je mobilní zařízení připojeno.

Implementace je provedena postupem[2], ve které se zjistí podpora tohoto rozhraní v prohlížeči, a návštěvník je vyzván k povolení zjištění polohy. Pokud už návštěvník v minulosti tento web navštívil a zjišťování povolil je tato volba zapamatována a již není k opakovanému potvrzení vyzýván. Metoda `getCurrentPosition()` vrátí přes callback funkci aktuální zeměpisnou šířku, délku a přesnost zaměření. Volitelnými informacemi, které nám

předaný objekt poskytuje, je nadmořská výška a přesnost jejího zaměření, rychlost pohybu zařízení v metrech za sekundu a směr pohybu zařízení od severu ve stupních (ve směru hodinových ručiček). V šabloně, kde je potřeba zjistit polohu návštěvníka, potom stačí pouze zavolat metodu `getPosition()` 5.4. Pokud je zjištěna poloha s vyhovující přesností,

```
function getPosition() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(function(position) {
      var latitude = position.coords.latitude;
      var longitude = position.coords.longitude;
      var accuracy = position.coords.accuracy;
      if(accuracy <= 125000) { // presnost vetsi jak 125km
        var center = new google.maps.LatLng(latitude, longitude, true);
        map.setCenter(center);
        map.setZoom(11);
      }
    }, error);
  }
}
```

Ukázka kódu 5.4: Zjednodušená metoda zjištění aktuální zeměpisné polohy návštěvníka.

Je mapě nastaven střed na tuto polohu, mapa přiblížena, na ní je vložena značka s polohou a přesnost zaměření je vyznačena lehce zbarveným kruhem.

5.6.1 Nabídka tipů dle polohy

Za předpokladu úspěšného získání zeměpisné polohy návštěvníka je přistoupeno k načtení tipů na procházky právě z jeho okolí. Provádí se za pomoci Ajaxu a příslušných metod v `APIPresenter`. Je vytvořen požadavek na načtení určitého počtu tipů s určitým posunem (limit a offset SQL dotazu) pro zadanou zeměpisnou polohu a vzdálenost od této polohy. V případě, že odpověď neobsahuje požadovaný počet tipů, je zvětšena vzdálenost od polohy návštěvníka a požadavek je poslán znovu. Nyní ovšem je posun zvětšen o již získaný počet tipů a požadovaný počet tipů je naopak snížen. Pokud už nejsou dostupné žádné procházky, nebo je překročena hranice poloměru vzdálenosti 200km, je načítání zastaveno. Vzdálenost 200km je nastavena z důvodu omezení načítání takto vzdálených procházek.

5.7 Vyhledávání

Aplikace umožňuje vyhledávat procházky jak pomocí jejich názvů, tak určité shody v popisu a především pomocí atributů. Jak už bylo zmíněno výše, některé atributy jsou organizovány ve skupinách. To nám dodá tu zajímavou vlastnost, že v případě vyhledávání procházky s jedním určitým atributem, která je ve skupině i s jinými, nám poskytne procházky i s atributy, které jsou v této skupině zařazeny (obr.5.3).

Samotný způsob vyhledávání v databázi je realizován pomocí dočasných tabulek (v MySQL příkaz `CREATE TEMPORARY TABLE`), které se vytvoří pouze na krátkou dobu připojení k databázi. Každý řádek v této tabulce je jedno klíčové slovo, na která se vyhledávací dotaz rozloží. Je proveden dotaz na získání všech ID procházek s relevancí výsledku, kde proběhne hledání na shodu klíčového slova a atributu, nebo shodu klíčového slova v názvu či popisu. V případě hledání shody klíčového slova a atributu je prohledávána vazební tabulka mezi atributem a skupinou atributů. Pokud je hledaný atribut obsažen v nějaké skupině, jsou

Hledaný výraz: **lavička**

Obsah databáze:

Procházky:

Procházka1: lavička, občerstvení
Procházka2: odpočívadlo, hřiště
Procházka3: restaurace, koupaliště
Procházka4: lavičky

Atributy:

Skupina1:

lavička
lavičky
odpočívadlo
odpočívadla

Skupina2:

občerstvení
restaurace
hospoda

Výsledek: Procházka1, Procházka2, Procházka4

Obrázek 5.3: Názorné vyhledávání procházek pomocí atributů.

vybrány všechny ID atributů v ní obsažené. Jestliže v žádné skupině není, je vybráno pouze samotné ID shodného atributu. Dále už probíhá pouze hledání mezi vazební tabulkou procházek a atributů.

Získaná ID procházek jsou poté použita v konstrukci dotazu `SELECT * FROM walk WHERE walk.id IN(...)`. Takto jsou získány všechny potřebné informace jako např. v úvodním přehledu procházek. Nedostatkem tohoto způsobu vyhledávání je fakt, že dočasná tabulka může být v dotazu použita pouze jednou. A protože je potřeba tabulku využít jak v hledání pomocí atributů ve skupině, tak v hledání samotných atributů a hledání v názvu a popisu, je nutné vytvořit tyto tabulky tři. I za použití optimalizací při vytváření těchto dočasných tabulek tento úkon využije značnou část času při provádění vyhledávacího dotazu. Je to způsobeno tím, že engine InnoDB není přímo optimalizován pro vkládání záznamů, ale pro čtení. Avšak po zvážení všech jiných variant je tento způsob vyhledávání nejrychlejší a nejvíce odpovídá požadavkům.

5.8 Zaznamenávání událostí

Veškeré akce, které se v aplikaci provádí, jsou zaznamenávány. Je využito logovacího souboru. Pokud uživatel například zobrazí detail procházky, je možné zjistit, jakým způsobem a odkud se na tento detail dostal. Zda použil mapu nebo přehled procházek pod ní, zda klikl na název procházky nebo využil tlačítka *detail* sloužící pro přesměrování. Je možné zjistit, jak dlouhý čas stráví uživatel vytvořením procházky či její trasy, nebo kolikrát uživatel procházku edituje, než je se svým výsledkem spokojen. Do logovacího souboru se dále zapisují hledané výrazy, které návštěvníci zadali, i počtem výsledků provedeného hledání. Těchto údajů je využito při optimalizaci uživatelského rozhraní a zlepšení použitelnosti aplikace. V případě povolení zjištění polohy návštěvníka, jsou zaznamenávány i souřadnice těchto návštěvníků. Toto sledování má dva cíle, a to statistiku návštěvníků, kteří povolí své sledování, a také statistiku přesnosti zaměření, která se dá využít při optimalizaci nabízení tipů na procházky.

5.9 Zveřejnění

Výsledná aplikace byla po testování zveřejněna. Jako hosting služby byl vybrán server Endora³, který poskytuje bezplatné úložiště pro různé projekty. Základní kapacita úložiště je 2GB a traffic 5GB za měsíc. Velikost databáze je omezena na 30MB. Jelikož ale v základní Free verzi není podporován databázový engine InnoDB, byl jsem nucen připlatit za program Plus. Ten obsahuje již zmíněnou podporu InnoDB, kapacitu databáze 60MB, velikost úložiště 5GB a traffic 30GB za měsíc. S tím došlo i k navýšení `memory_limit` v PHP na 128MB, to je aktuálně ve verzi 5.4 a MySQL ve verzi 5.1. Hosting garantuje dostupnost služeb na 98%, což není úplně ideální, ale pro bakalářskou práci dostačující. Internetová doména naProchazku.cz, která již byla v předstihu registrována vedoucím mé práce u společnosti Web4u⁴, byla přeměrována na tento webhosting. Pro podporu služby a získání fanoušků jsou vytvořeny stránky na službách Google+⁵ a Facebook⁶. Bylo provedeno několik kampaní, které měly za účel zvýšit návštěvnost služby naProchazku.cz a získat tak reprezentativní vzorek návštěvníků, z kterého částečně vychází následující sledování chování uživatelů. Webové stránky jsou spravovány pomocí *Nástrojů pro webmastery* a *Analytics* od společnosti Google.

³<http://endora.cz>

⁴<http://www.web4u.cz/>

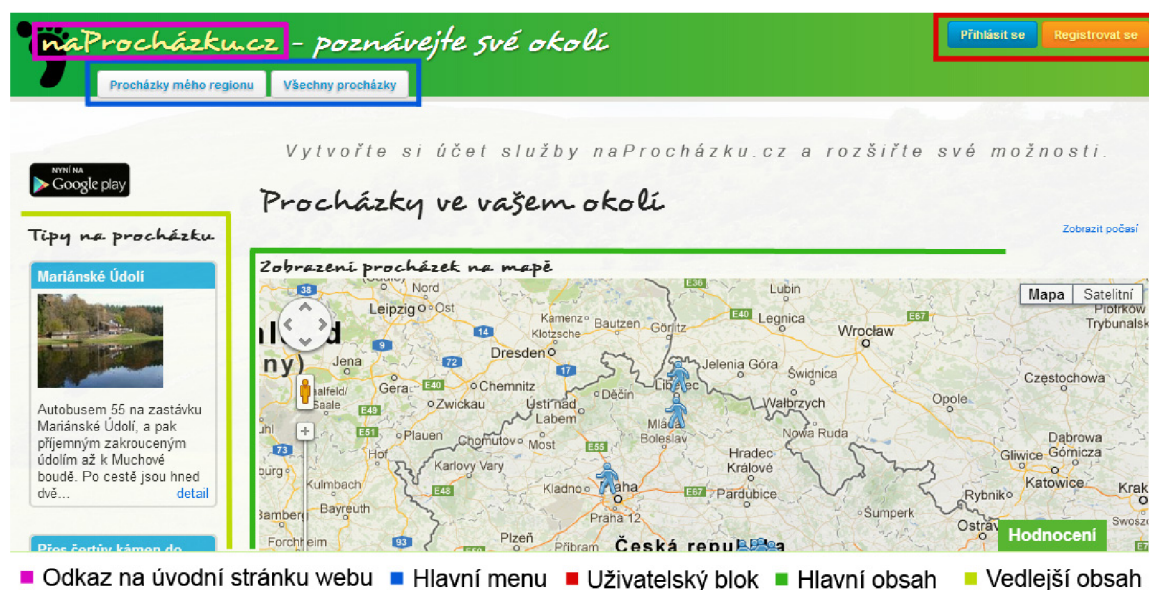
⁵<https://plus.google.com/112168309627199598387>

⁶<https://www.facebook.com/naProchazku.cz>

Kapitola 6

Sledování chování uživatelů

Sledovat chování uživatelů je přínosné pro další vývoj aplikace. Je možné získat informace o tom, kde na uživatele čekají největší úskalí a co uživatele nejvíce zaujalo. V dalším vývojovém cyklu aplikace je nejlepší těmito úskalím předejít a vylepšit aplikaci tak, aby uživatelé přinesla ještě lepší komfort používání. Při sledování chování byly použity dvě metody. Tou první je přímé sledování, kdy jsem měl příležitost pozorovat uživatele přímo při práci. Druhá metoda je anonymní sledování, kdy se vychází pouze ze statistik.



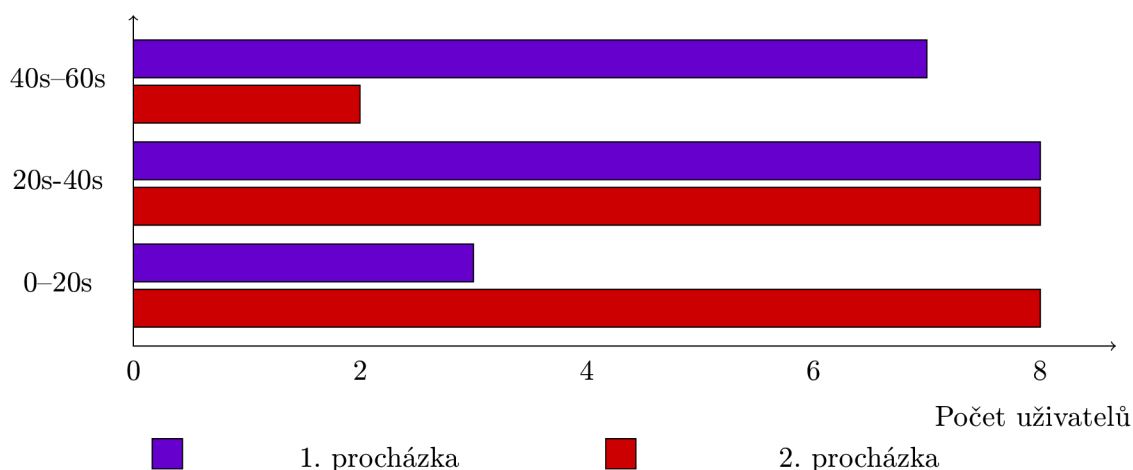
Obrázek 6.1: Testované uživatelské rozhraní spolu s popisem jednotlivých částí.

6.1 Přímé sledování

V metodě přímého sledování chování uživatelů byl uživatelé předložen konkrétní problém a sledoval jsem reakce a postup při řešení tohoto problému. Prvním úkolem bylo najít 2 konkrétní procházky, aniž by uživatel kdykoliv předtím aplikaci viděl. Výsledek je zobrazen na grafu 6.2. Testování jsem prováděl na vzorku 18 lidí. Je možné si povšimnout toho, že návštěvník se po prvním úspěšném pokusu už naučí postup a druhé hledání je jednodušší a časově méně náročné.

Osoba č.	1	2	3	4	5	6	7	8	9
Věk	19	24	35	53	26	23	19	23	15
Pohlaví	muž	žena	žena	muž	muž	muž	žena	žena	žena
Vzdělání	Z	VO	S	V	VŠ	V	V	S	Z
Zkušenosti PC	4	3	4	2	4	2	3	4	3
Osoba č.	10	11	12	13	14	15	16	17	18
Věk	13	48	43	22	58	17	19	27	24
Pohlaví	muž	muž	žena	muž	muž	žena	žena	muž	muž
Vzdělání	Z	VŠ	VŠ	S	V	Z	V	VŠ	S
Zkušenosti PC	3	3	3	5	2	4	3	5	5

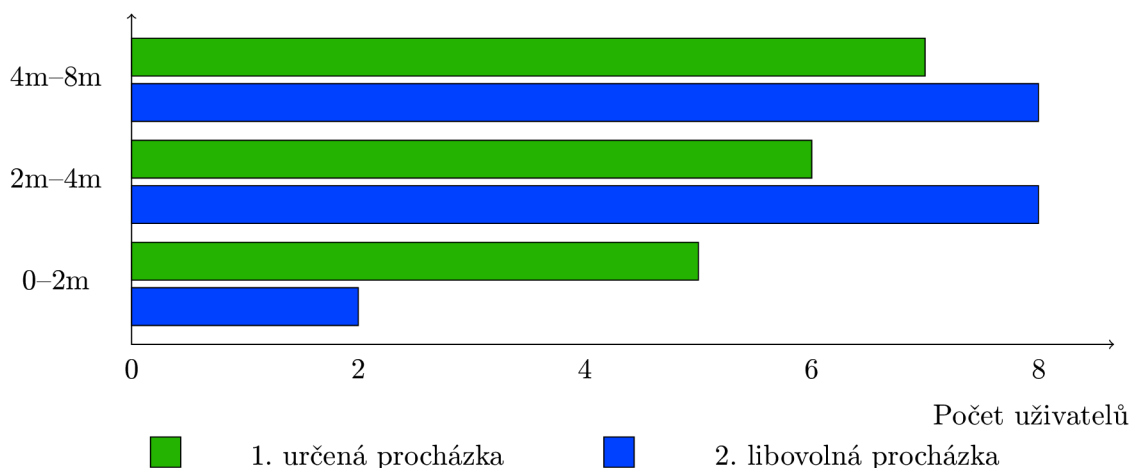
Tabulka 6.1: Složení testované skupiny. Vzdelání: Z-základní, V-vyučen, S-středoškolské, VO-vyšší odborné, VŠ-vysokoškolské. Zkušenosti PC: od 1(nezkušený) do 5(zkušený).



Obrázek 6.2: Čas, který potřeboval uživatel pro vyhledání určité procházky.

Druhým úkolem bylo vytvořit jednu konkrétně určenou procházku a k ní připojit trasu a vytvořit druhou procházku dle fantazie uživatele. Výsledek je opět zobrazen v grafu 6.3. Z tohoto pokusu jsem zjistil, že pokud má uživatel přesně zadáno, co má dělat, a má názornou ukázkou, jak má vypadat trasa procházky, splní úkol rychleji, než při druhém pokusu. Tam už uživatel zkouší, co všechno může aplikace nabídnout, a nesoustředí se tak přímo na vytváření trasy procházky. Zjistil jsem také, že pokud uživatel vytváří procházku dle libosti, několikrát ji edituje, než je s konečným výsledkem spokojen. Neméně významným poznatkem je také fakt, že některý uživatel není schopen pochopit, že v rámci jedné procházky nemůže vytvořit více tras. Nevím, zda je to dáno špatným návrhem aplikace, kdy jsem měl tuto možnost zahrnout do funkcí služby, nebo je to dáno kladením velkých nároků uživatele na službu. Každopádně je to důvod se nad tímto poznatkem zamyslet a zvážit rozšíření i v tomto směru.

Třetím testem bylo, aby uživatel, po dokončení dvou předchozích testů, popsal, k čemu služba podle něj slouží a pro jakou cílovou skupinu lidí je určena. Většina dotazovaných osob na první otázku odpovídala ve třech různých variantách, které jsem shrnul.



Obrázek 6.3: Čas, který potřeboval uživatel pro vytvoření dvou kompletních procházek.

- Slouží pro prezentaci oblíbených procházek z okolí svého domova.
- Pro nabídku aktivit v okolí procházek, nebo vytvoření procházky za účelem navštívení této aktivity.
- Je užitečným nástrojem pro sdílení a doporučování výletů a turistických tras z míst, které uživatel navštívil.

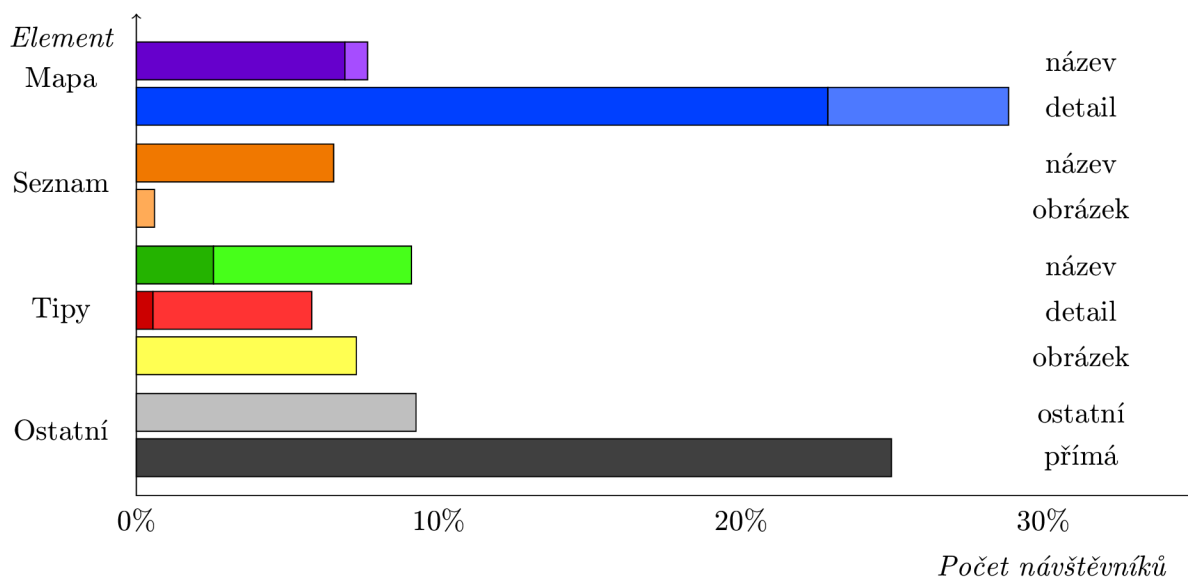
První dvě odpovědi považuji za úspěšné, třetí jen z části. Druhá otázka „Pro jakou cílovou skupinu lidí je služba určena?“ reflektovala odpovědi první otázky. Opět jsem to shrnul do dvou odpovědí.

- Je určena pro rodiny, páry i jednotlivce, kteří mají rádi přírodu.
- Je určena pro turisty a cestovatele, kteří rádi poznávají zapomenuté kouty naší vlasti.

První odpověď mě uspokojila pouze částečně, druhá mě zklamala, protože testované osoby nerozpoznaly, že procházky lze také vykonat v městském prostředí a objevovat tak zajímavá místa i v centrech měst. Samozřejmě potvrzují, že procházka v přírodě je mnohem lákavější. Proto je na zvážení, jak návštěvníkovi přesněji sdělit, k čemu je služba určena.

6.2 Nepřímé sledování

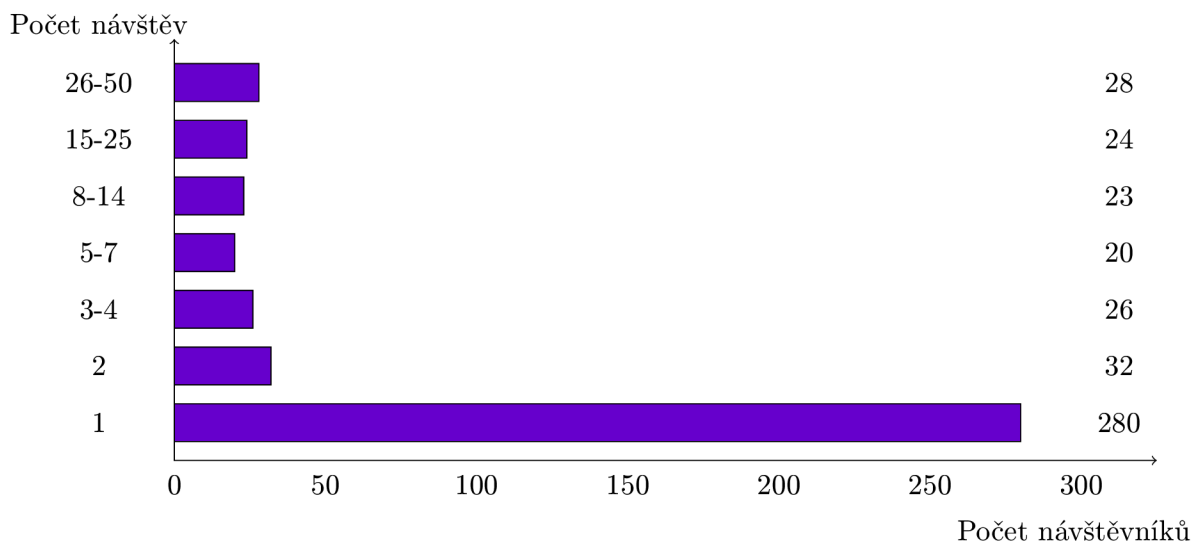
Toto sledování bylo vytvořeno pomocí statistik z Google Analytics a logovacího souboru akcí za období 3 týdnů. Nejdříve je zde statistika, z jakých stránek a jakým způsobem se návštěvníci dostali na detail procházky. Statistika na obrázku 6.4 vychází z celkového počtu 550 zobrazení detailu procházky. A vyplývá z ní, že návštěvníci nejčastěji vybírají procházku z mapy a poté kliknutím na popis *detail*, po kterém dojde k načtení nové stránky s detailem procházky. Pokud je však k dispozici aktivní obrázek, jako je to v případě tipů na procházky, pak už rozhodnutí uživatelů není tak jednoznačné a zhruba ve stejném počtu klikají na název procházky (který je podbarven), poté hned na aktivní obrázek a následně až na popis *detail*. Naopak, pokud jsou procházky seřazeny v tabulce, naprostá většina zvolí cestu k detailu procházky pomocí jejího aktivního jména a obrázek nepoužijí. Došel jsem tedy k názoru, že uživatelé nejlépe vyhovuje výběr procházky z mapy a upoutají ho



Obrázek 6.4: Statistika zdrojů návštěvnosti detailu procházky. Je procentuálně vyjádřeno jakým způsobem se uživatel dostal (jaký element vybral) na detail procházky.

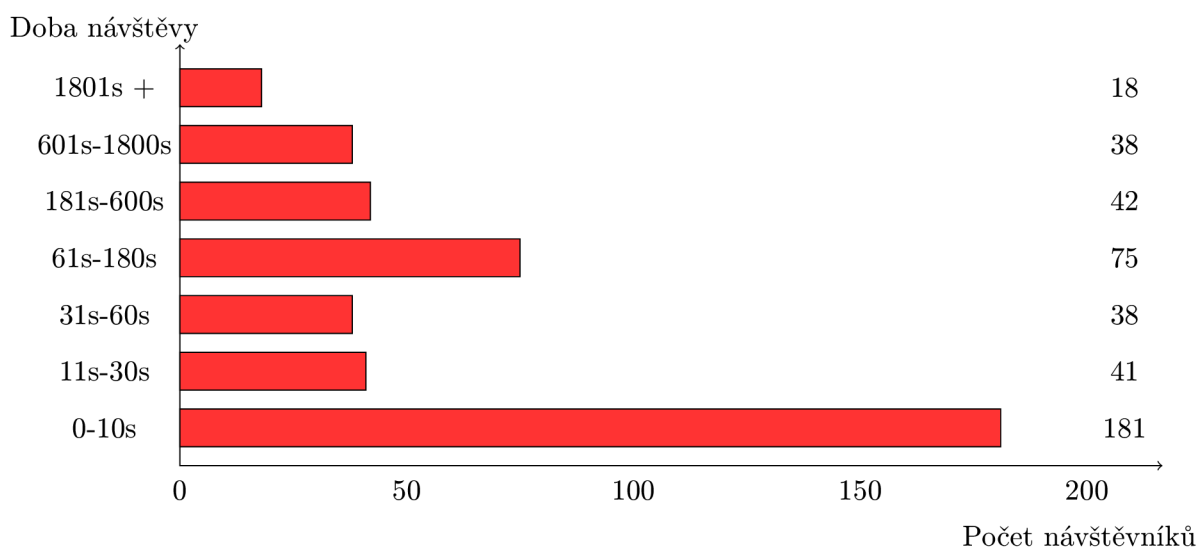
také tipy na procházku z postranního panelu.

Ze statistiky, která udává kolik návštěvníků povolí zjištění své zeměpisné polohy, je zcela zřejmé, že více jak 65% uživatelů tuto volbu nepovolí. Z toho vyplývá, že tito uživatelé jsou ochuzeni o možnost nabídky tipů procházek z okolí. Z celkových 280 unikátních návštěvníků se registrovalo 8 přes webovou službu a 10 přes mobilní aplikaci. To znamená, že každý 15. návštěvník je ochoten se do nové služby registrovat. Těchto 18 uživatelů vytvořilo během 21 dnů 20 nových procházek. Pokud sledujeme návštěvníky a počet jejich návštěv, zjistíme, že velké procento lidí navštívilo web pouze jednou. Statistika je zobrazena v grafu 6.5. Nejspíše nenašli to, co hledali. Je tu ale také nezanedbatelná část návštěvníků, která se vrátila několikrát. To je pro mě důležitý údaj, který udává, kolik návštěvníků tato služba zaujala a vrací se pro nové informace. Pokud nebereme v potaz uživatele kteří se na web už nevrátili, průměrná doba mezi dvěma návštěvami je 2.8 dne. Míra okamžitého opuštění je okolo 40% a průměrná doba jedné návštěvy webu je 4 minuty 30 sekund.



Obrázek 6.5: Statistika počtu návštěv na uživatele.

Další statistika na obrázku 6.6 vyjadřuje, jak dlouhou dobu uživatel na celém webu strávil. Opět se zde projevuje velká část návštěvníků, kteří nenašli to co hledali. Z těchto na-



Obrázek 6.6: Statistika počtu doby strávené jednoho uživatele.

měřených dat jasně vyplývá, že služba si získala své uživatele, avšak neochota sdílet své procházky s ostatními degraduje účel, za kterým služba vznikla. Je tedy nutné určitým způsobem návštěvníka motivovat, aby tak učinil a nebál se podělit o své oblíbené trasy s ostatními.

Kapitola 7

Navrhovaná rozšíření

Z komentářů a reakcí uživatelů vyplývá, že hlavním nedostatkem je vzhled stránek. Proto je nutné tuto změnu vykonat co nejdříve, aby vzhled stránek odpovídal moderní době, ve které vznikl, a stále respektoval trendy v této oblasti designu.

Jeden z nápadů, který mě velmi oslovil, byl takový, že v případě vyhledávání procházky na mapě bude přímo v ní zobrazena i trasa procházky. A to stylem, že pokud kliknu na zástupný symbol, který procházku reprezentuje, vykreslí se mi do aktuálně zobrazované mapy trasa procházky. Po zavření malého dialogu, který se také zobrazí, by vykreslení trasy zmizelo.

Některé akce by šlo implementovat pomocí Ajaxu. Zabránilo by tak tak např. opětovnému a především zdlouhavému načítání mapy a snížil by se také počet výpočetně náročných dotazů do databáze.

Další z poznatků během sledování chování byl, že návštěvníci se neradi registrují. Bylo mi doporučeno, že pokud by služba umožňovala přihlášení např. přes rozhraní Google, Facebook nebo MojeID, počet registrovaných (přihlášených) uživatelů by stoupl. Zároveň by služba z pohledu návštěvníka vypadala více věrohodně.

Je nutné vytvořit alternativní způsob zobrazování tipů na procházky z okolí návštěvníka v případě nepovolení sledování zeměpisné polohy.

Nejeden uživatel si posteskl nad tím, že při vytváření trasy procházky nelze práci přerušit, vložit na mapu několik zájmových bodů a dále pokračovat v kreslení trasy.

Určitě přínosným rozřířením by bylo možnost procházky komentovat a tyto komentáře sdílet, či zobrazovat počet shlédnutí dané procházky.

Co vyplynulo z testování je fakt, že připojit více tras k jedné procházce, nebo umožnit propojení dvou procházek, je funkce, kterou uživatelé od služby očekávají. Je tedy vhodné tuto možnost zvážit a zamyslet se nad elegantní implementací a především návrhem uživatelského rozhraní, aby uživatel poznal, jaké jsou možnosti aplikace, a mohl bez dlouhého přemýšlení tuto funkci využít.

Kapitola 8

Závěr

V závěru mé práce bych chtěl zhodnotit mnou vytvořenou webovou službu naProcházku.cz. Toto vybrané téma mi bylo blízké, protože mám rád procházky a rád poznávám krásu přírody nejen v České republice. Věřím tomu, že tento web bude více a více navštěvován a že budou uživatelé sdílet své zážitky z procházek.

Služba nabízí jedinečné sdílení procházek pomocí mapy jenž je hlavním vizualizačním nástrojem celé aplikace. Uživatelům umožňuje vytváření vlastních procházek s možností následné editace, ostatní procházky je možné přidávat do seznamu oblíbených nebo je sdílet na sociálních sítích. Dle aktuální polohy zobrazuje tipy na procházky. Aplikace implementuje otevřené aplikační rozhraní, které využívá spolupracující mobilní aplikace. Je řešena v rámci souběžné bakalářské práce, jejímž autorem je Rostislav Dočkal, pod názvem *Naprocházku.cz - mobilní služba pro poznávání okolí*.

Otestoval jsem 18 lidí, kteří tuto službu viděli poprvé. Zjišťoval jsem, jak rychle se budou na webu naProcházku.cz orientovat, jaké budou jejich reakce na nepředvídatelné události. Byl jsem spokojen s výsledky prvních měření. Uživatelé se začali velmi rychle orientovat a osvojovat si postupy. Po provedení prvních změn a novém testování však byli uživatelé zaskočení. Jejich prověřené postupy nešli provést, a tak museli přijít na nové. Ovšem naučili se je rychle. Bylo pro mne ale překvapením, že uživatelé více zdatní v ovládnání PC si se službou rozuměli méně, než uživatelé méně zdatní. Podle mého zjištění to bylo zapříčeno také tím, že více zdatní uživatelé očekávali od služby více funkcí, někdy až nemožných. I to mi ale dodalo spoustu nových nápadů, které chci v aplikaci dále realizovat. Co se týče anonymních návštěvníků, zdá se, že některým se služba opravdu zalíbila a několikrát se vrátili. Míra okamžitého opuštění je okolo 40%, to znamená v kolika procentech návštěvník opustil web již na první stránce. Průměrná doba jedné návštěvy webu je 4 minuty 30 sekund. Výsledky měření dokazují, že návštěvníci pracují především s mapou a upřednostňují klikání na jasně znějící popisky, než na obrázky a titulky.

V budoucnu bych chtěl především zlepšit grafickou stránku webu, zkvalitnit uživatelské prostředí při práci s mapou, navrhnout důmyslnější nabídku tipů na procházky z okolí a zvýšit návštěvnost pomocí lepší SEO.

Tento projekt považuji za úspěšný a očekávám, že se postupem času stane úspěšným i u návštěvníků.

Literatura

- [1] Consortium, W. W. W.: Geolocation API Specification: W3C Proposed Recommendation 10 May 2012. [online], 2012.
URL <http://www.w3.org/TR/geolocation-API/>
- [2] HOGAN, B. P.: *HTML5 a CSS3*, kapitola Využití dalších API: Jak se najít: geolokace. Computer Press, vyd. 1. vydání, 2011, ISBN 978-80-251-3576-1, s. 226–229.
- [3] KRUG, S.: *Web design: nenutte uživatele přemýšlet!* Brno: Computer Press, druhé vydání, 2006, ISBN 80-251-1291-8.
- [4] NIELSEN, J.: *Web design*. Praha: SoftPress, vyd. 1. vydání, 2002, ISBN 80-864-9727-5.
- [5] Sloupenská, H.: *Geolokace z hlediska ochrany soukromí*. 2011.

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 L^AT_EX
- src/readme.txt - návod k instalaci
- src/doc - složka s programovou dokumentací
- src/app.zip - archiv zdrojových souborů implementované aplikace
- src/db_install_schema.sql - soubor ve formátu sql pro instalaci schématu databáze
- src/db_install_data.sql - soubor ve formátu sql obsahující vzorová data databáze

Příloha B

Definice metod aplikačního rozhraní

Všechny metody začínají `http://example.tld/api?api=naprochazku` a je možné využít jak POST tak GET.

Některé metody už prošly změnou, co se týče obsahu odpovědi, proto je u nich nutné uvádět další parametr `v=2`.

Všechny odpovědi a některá složitá vstupní data jsou ve formátu JSON.

Odpověď je vždy ve tvaru

```
{
status: „success“, // může nabývat hodnot success nebo při chybě fault
result: „data“ // zde je výsledek, který je dále pro každou metodu rozepsán. Při chybě je
zde chybová hláška
}
```

Souřadnice v odpovědi jsou vždy ve tvaru

formát: `coordinate { walk_id, id, latitude, longitude, previous_coordinate_id, attribute_id }`

Zájmový bod v odpovědi je vždy ve tvaru

formát: `pointOfInterest { walk_id, latitude, longitude, type, title, note, w_attribute_id }`

Jednotlivé metody a jejich odpovědi:

Seznam všech procházek

vstup: `&v=2&method=allWalks`

výstup: `{ [user_name, country { short, original } , walk_id, short_name, name, about, image_url], ... }`

Hledání v procházkách

vstup: `&method=findWalks&value=tags_or_key_words`

výstup: stejný jako v případě seznamu procházek: Každá procházka má navíc položku `rank` s relevancí shody.

Načtení detailu procházky

vstup: `&v=2&method=walkDetail&short_name=libovolna-prochazka`

výstup: `user_name, country { short, original } , walk_id, short_name, name, about,`

image_url, *tags*[{ *name*, *relevance*, *group_id*, *group_name*}, ...], *coordinates* { seznam souřadnic, viz. více *coordinate*},
pointsOfInterest { seznam zájmových bodů, viz. více *POI*}

Vytvoření nové procházky

vstup: `&method=createWalk&input={ walk_name, user_id, country, walk_about, coords[{latitude, longitude},...], tags[{name},...]}`
výstup: { *walk_id*, *short_name*}

Přihlášení uživatele

vstup: `&method=login&login=...&password=...`
výstup: { *identity*{*id*, *username*, *email*, *country*{*short*, *original*, *latitude*, *longitude*}, *about*, *latitude*, *longitude*, *roles*[...]} }

Následující *api_key* je unikátní klíč vždy pro danou spolupracující aplikaci.

Seznam všech uživatelských jmen

vstup: `&method=getAllUsername&k=api_key`
výstup: { [*username*, ...] }

Vytvoření nového uživatele

vstup: `&method=createUser&k=api_key&input={username, name, password, email, country}`
výstup: { *user_id* }

Seznam všech dostupných tagů

vstup: `&method=getAllTags`
výstup: { [{ *id*, *name*, *relevance*, *uses*, *group_id*, *group_name* }, ...] }

Seznam tagů pro zadanou procházku

vstup: `&method=getTagsForWalk&walkId=walk_id`
výstup: stejný formát jako předchozí

Uložení poslední známé pozice uživatele

vstup: `&method=saveLastPositionOfUser&latitude=...&longitude=...&user_id=...` //
parametr *user_id* není povinný
výstup: success message