

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky

Podniková platforma pro vyhledávání komplexních geolokačních údajů  
Ondřej Chrbolka

Bakalářská práce  
2020

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Ondřej Chrbolka**  
Osobní číslo: **I17246**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Téma práce: **Podniková platforma pro vyhledávání komplexních geolokačních údajů**  
Zadávací katedra: **Katedra informačních technologií**

## Zásady pro vypracování

Cílem bakalářské práce je vytvořit aplikační rozhraní (API) umožňující zpracování komplexních geolokačních informací zahrnujících název státu, ulice, případně číslo popisné pro vyhledávaný bod. Zdrojem dat pro API budou mapové podklady OpenStreetMap. V teoretické části budou uvedeny různé možnosti geolokace se zaměřením na GPS případně Galielo. Podrobněji budou popsány: mapové podklady OpenStreetMap včetně licencování, ukládání a orientace v datových souborech, dále způsoby importu datových podkladů do relační databáze a možnosti optimalizace práce s již hledanými záznamy.

V praktické části bude naprogramováno a otestováno RESTful API v jazyce Java, s cílem implementace do podnikového informačního ekosystému. API umožní detailní monitorování firemních automobilů vybavených geolokačním systémem. API bude poskytovat generování náhledu mapy a reverzní geokódování s využitím datových souborů OpenStreetMap. Dále bude navržena optimalizace pro provádění dotazů na opakovaně dotazované části mapy.

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

Seznam doporučené literatury:

ARSANJANI, Jamal Jokar. OpenStreetMap in GIScience : Experiences, Research, and Applications. Cham, Switzerland: Springer International Publishing, 2015. ISBN 3319142798.  
LAURET, Arnaud. The Design of Web APIs. New York, United States: Manning Publications, 2019. ISBN 1617295108.  
GERARDUS BLOKDYK, Gerardus. Open api a complete guide – 2019 edition. Emereo. ISBN 9780655812654.  
OpenStreetMap, „About Open Street Map,“ OpenstreetMap, 2020. [Online]. Available: <https://www.openstreetmap.org/about>.

Vedoucí bakalářské práce: **Ing. Soňa Neradová, Ph.D.**  
Katedra informačních technologií

Datum zadání bakalářské práce: **15. listopadu 2019**  
Termín odevzdání bakalářské práce: **7. května 2020**



---

**Ing. Zdeněk Němec, Ph.D.**  
děkan

---

**Ing. Lukáš Čegan, Ph.D.**  
vedoucí katedry

V Pardubicích dne 17. prosince 2019

Prohlašuji:

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

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

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 20. 8. 2020

Ondřej Chrbolka

## **PODĚKOVÁNÍ**

Mé poděkování patří Ing. Soně Neradové, Ph.D., za poskytnuté konzultace a pomoc při řešení obtížných částí bakalářské práce. Poděkovat bych chtěl také rozsáhlé komunitě OpenStreetMap, zejména pak uživatelům vystupujícím pod přezdívkami „alan\_gr“ a „PHerison“, kteří mi poskytli cenné informace a rady.

## **ANOTACE**

Bakalářská práce se zaměřuje na vytvoření podnikové platformy pro vyhledávání komplexních geolokačních údajů při zadání jednoduchých geolokačních primitiv. V práci jsou popsány různé způsoby geolokace, orientace a vyhledávání v mapových podkladech OpenStreetMap®. Uvedena je tvorba RESTful aplikačního rozhraní (API) pro testování funkčnosti s využitím frameworku Spring a programovacího jazyka Java.

## **KLÍČOVÁ SLOVA**

Geolokace, mapy, OpenStreetMap, PostgreSQL, REST, API, Spring, Java

## **TITLE**

Company platform for geolocation data search

## **ANNOTATION**

The bachelor thesis focuses on establishing a company platform for complex geolocation data identification after entering simple geolocation primitives. The thesis describes various ways of geolocation, orientation and locating in base maps of Open Street Map®. It provides an establishment of RESTful of Application Programming Interface (API) for functions testing with the use of the Spring framework and the Java programming language.

## **KEYWORDS**

Geolocation, maps, OpenStreetMap, PostgreSQL, REST, Spring, Java.

# OBSAH

<b>Seznam obrázků</b> .....	<b>9</b>
<b>Seznam tabulek</b> .....	<b>10</b>
<b>Seznam ukázek zdrojového kódu</b> .....	<b>11</b>
<b>Seznam zkratk</b> .....	<b>12</b>
<b>Úvod</b> .....	<b>13</b>
<b>1 Postup plnění práce</b> .....	<b>14</b>
<b>2 Teoretická část</b> .....	<b>15</b>
2.1 Geolokace .....	15
2.2 Druhy geolokací .....	15
2.3 Využití geolokace .....	17
2.4 Komerčně dostupné mapové nástroje .....	17
2.4.1 Google Maps API .....	17
2.4.2 MapBox .....	18
2.5 OpenStreetMap .....	19
2.5.1 Historie .....	20
2.5.2 Licencování .....	20
2.5.3 Struktura .....	20
2.5.4 Druhy cest .....	21
2.5.5 Definice domů .....	22
2.5.6 Definice ulic .....	23
2.5.7 Chybavost mapových podkladů .....	27
2.5.8 Problematika adresace v OpenStreet mapách .....	27
2.5.9 Uložení mapových podkladů .....	28
2.5.10 OSM XML .....	29
2.5.11 PBF formát .....	29
<b>3 Tvorba geolokační komponenty</b> .....	<b>30</b>
3.1 Použité nástroje a programovací jazyky .....	30
3.1.1 IntelliJ IDEA .....	30
3.1.2 Osmosis .....	31
3.2 Proces reverzní geolokace .....	32
3.2.1 Určení ulice .....	32
3.2.2 Určení čísla domu .....	34
3.2.3 Určení města .....	34
3.2.4 Určení státu .....	37
3.3 Vykreslování obrázkových map .....	38
3.4 Optimalizace vyhledávání v mapových podkladech .....	39
3.4.1 Cashování často dotazovaných částí mapy .....	39
3.4.2 Serializace již načtených mapových podkladů .....	40
3.4.3 Načtení již stažených mapových podkladů .....	40
<b>4 Tvorba aplikačního rozhraní (API)</b> .....	<b>41</b>
4.1 Spring .....	41

4.1.1	Spring Boot .....	41
4.1.2	REST rozhraní .....	42
4.1.3	Mapování požadavků .....	42
4.2	Zprovoznění primárního mapového serveru .....	43
4.3	Import mapových podkladů do databáze .....	44
4.3.1	Závislosti .....	44
4.3.2	Nastavení serveru .....	46
4.3.3	Návrh možné optimalizace .....	47
<b>Závěr .....</b>		<b>48</b>
<b>Použitá literatura .....</b>		<b>49</b>
<b>Přílohy .....</b>		<b>52</b>



## SEZNAM OBRÁZKŮ

Obrázek 1: Havlíčkovo náměstí vyobrazené v OpenStreetMap .....	24
Obrázek 2: Vyznačení prvního uzlu na mapovém podkladu .....	25
Obrázek 3: Vyznačení koncového uzlu a cestu, která jej spojuje.....	25
Obrázek 4: Nová cesta společně se svým koncovým bodem .....	26
Obrázek 5: Ukázka chybně označené ulice v OpenStreetMap, navigaci Waze a Apple mapách .....	27
Obrázek 7: Vývojové prostředí IntelliJ s postranními panely se strukturou projektu a databázovými připojeními .....	31
Obrázek 9: Vyobrazení přibližné uživatelské polohy a cest reprezentující ulice .....	33
Obrázek 10: Mapová oblast Jihlavy.....	35
Obrázek 11: Mapová oblast obce Kratonohy .....	35
Obrázek 12: Mapová oblast Ondříčkovy ulice v Praze .....	36
Obrázek 13: Mapová oblast Prahy .....	37
Obrázek 14: Webová stránka <a href="https://www.naturalearthdata.com/">https://www.naturalearthdata.com/</a> .....	38
Obrázek 15: Dotazované umístění na mapové dlaždici .....	39

## SEZNAM TABULEK

Tabulka 1: Ceník Google Maps API.....	18
Tabulka 2: Ceník Mapbox API pro mobilní zařízení.....	18
Tabulka 3: Ceník Mapbox API pro webové rozhraní.....	19
Tabulka 4: Klasifikace cest v České republice.....	22
Tabulka 5: Doporučený zápis adresy pro Českou republiku.....	28
Tabulka 6: Obecné atributy elementu v OSM souboru.....	29
Tabulka 7: Nároky na výkon serveru pro mapový podklad celé planety.....	43
Tabulka 8: Nároky na výkon serveru pro mapový podklad Velké Británie.....	44
Tabulka 9 Ukázka výstupu dotazu na databázi s importovanými mapovými podklady.....	46

## SEZNAM UKÁZEK ZDROJOVÉHO KÓDU

Ukázka zdrojového kódu 1 Označení druhu cesty.....	21
Ukázka zdrojového kódu 2: Uzel popisující budovu v mapových podkladech.....	23
Ukázka zdrojového kódu 3: Cesta na Havlíčkovo náměstí definována v mapovém podkladu	24
Ukázka zdrojového kódu 4: Reference na uzly obsažené v cestě.....	24
Ukázka zdrojového kódu 5: Zeměpisná šířka a délka v definici prvního uzlu.....	25
Ukázka zdrojového kódu 6: Zeměpisná šířka a délka v definici druhého uzlu.....	25
Ukázka zdrojového kódu 7: Ukázka druhé cesty, která je ve vztahu s předchozí.....	26
Ukázka zdrojového kódu 8: Koncový bod druhé cesty.....	26
Ukázka zdrojového kódu 9: Adresace domu v obci.....	35
Ukázka zdrojového kódu 10: Označení hlavního města, nevhodné ke zpracování.....	36
Ukázka zdrojového kódu 11: Spouštěcí třída Springové aplikace.....	41
Ukázka zdrojového kódu 12: Interface zajišťující CRUD operace pro entitu mapového podkladu.....	42
Ukázka zdrojového kódu 13: Ukázka směrování metodou GET s vyžadovanými parametry a kódem, který je po přijetí požadavku proveden.....	43

## **SEZNAM ZKRATEK**

ARM	Advanced RISC Machine
CRUD	Create, Retrieve, Update, Delete
CSS	Cascading Style Sheets
GNSS	Global Navigation Satellite System
GNU	GNU's Not Unix
GPS	Global Positioning System
IP	Internet protocol
JDK	Java Development Kit
JVM	Java Virtual Machine
OSM	OpenStreetMap
PBF	Protocolbuffer Binary Format
PDF	Portable Document Format
SQL	Structured Query Language
URL	Uniform Resource Locator

## ÚVOD

Bakalářská práce je zaměřena na tvorbu geolokačního rozhraní (API) pro společnost Zauzoo s. r. o. Toto rozhraní poskytuje vyhledávání pokročilých informací o umístění na základě geolokačních primitiv, jako jsou zeměpisná šířka a délka. Využití nalezne hlavně v dopravní infrastruktuře, kde bude využito pro mapování firemních vozidel.

V současné době poskytuje na trhu tyto služby několik společností (např.: Google Inc.). Jejich sazba se ale pohybuje v průměru 100 Kč za 1 000 požadavků. Při monitorování v reálném čase může počet požadavků mnohonásobně převýšit akceptovatelné finanční náklady.

Cílem bakalářské práce byla optimalizace nákladů tvorbou vlastní knihovny s využitím již existujících nástrojů a mapových podkladů OpenStreetMap®. Ty jsou volně dostupné pod licencí Open Database Licence (ODbL) a jejich obsah je pravidelně aktualizován více než šesti milióny uživateli.

OpenStreetMap® ukládají mapová data celé planety nebo jednotlivých regionů v souborech XML (\*.osm). Pro komunikaci s již existujícími firemními subsystémy je vytvořeno API rozhraní, jež vrací odpovědi na požadavky ve formátu JSON. Backend aplikace je řešena v jazyce Java s využitím frameworku Spring.

# 1 POSTUP PLNĚNÍ PRÁCE

Základním cílem bakalářské práce byl návrh a následná implementace aplikačního rozhraní. Jeho funkcí je zpracování požadavků od klientů na reverzní geolokaci, vykreslení mapových podkladů a monitorování firemních automobilů. Reverzní geolokaci se rozumí proces, při kterém je na základě zasláné zeměpisné šířky a délky vyhodnocena přibližná adresa obsahující alespoň ulici, číslo popisné, město a stát.

Pro minimalizaci nákladů byly použity již existující mapové podklady OpenStreetMap. Detailně byla popsána orientace v souborech reprezentujících mapové podklady a získání informací potřebných pro realizaci rozhraní. Následně také byly popsány technologie využité při implementaci backendové části.

Dále byla navržena optimalizace vyhledávání v mapových podkladech, aby se zajistila plynulost zpracování dotazů a minimalizovalo se zbytečné opakované zpracování informací. Některé z těchto optimalizací byly i přímo implementovány ve výsledném projektu.

Výstupem praktické části je serverová aplikace založená na Spring frameworku, která je schopná provádět CRUD operace nad firemními vozidly, zpracování OpenStreetMap souborů, reverzní geolokaci a vykreslování mapových podkladů.

## 2 TEORETICKÁ ČÁST

Teoretická část je rozdělena do více úseků. První obsahuje stručný popis a druhy geolokace včetně jejího využití v moderní době. V druhém jsou cenově srovnána již existující aplikační rozhraní pro danou problematiku. Následně je detailně popsán OpenStreetMap včetně historie, licencování a orientace v mapových podkladech.

### 2.1 Geolokace

Proces, při kterém je hledáno a určeno umístění počítače, chytrého příslušenství nebo jiného zařízení podporujícího lokalizační služby. Nejčastěji se využívá jako součást aplikací v chytrých zařízeních. Ty mají v dnešní době implementovaný nějaký druh lokalizačního zařízení, nejčastěji GPS. [1]

Umístění zařízení je vyjadřováno v **zeměpisné šířce** a **zeměpisné délce**. Jedná se o desetinné číslo, které může nabývat hodnot ve vymezeném intervalu. Zeměpisná šířka vyjadřuje umístění od západního do východního bodu s maximální hodnotou  $+180^\circ$  a minimální hodnotou  $-180^\circ$ . Zeměpisná délka je naopak vyjádření umístění od severního do jižního bodu s maximální hodnotou  $90^\circ$  a minimální hodnotou  $-90^\circ$ . Při určování přibližné adresy je využívána technika **reverzního geokódování**, kdy je na základě vložených údajů určena přibližná adresa zařízení.

### 2.2 Druhy geolokací

#### a) Wifi geolokace

Proces určení lokace zařízení je možné urychlit s využitím wifi geolokace. V současné době je možné v každé oblasti najít alespoň jeden přístupový bod wifi. Společnosti, jako jsou Apple, Microsoft či Mozilla poskytují geolokační služby, které jsou na základě viditelných wifi sítí schopné určit uživateli polohu. Při požadavku jsou odeslány údaje jako SSID, BSSID a MAC adresa, ty jsou následně vyhledávány ve wifi geolokační mapě. Odchylka při wifi geolokaci se pohybuje okolo 35 metrů [2].

Komerční společnosti jako například Google získávaly jednotlivé Wifi přístupové body pomocí firemních vozidel při vizualizaci ulic po celém světě pro službu Google Street View. Další společnosti jednoduše využily samotné uživatele svých zařízení, které při využití geolokačních služeb odeslali GPS pozici (pokud byla dostupná) společně s viditelnými sítěmi [2].

Při wifi geolokaci jsou tedy sbírána data nejen komerčních wifi sítí, ale také domácností. Může tedy vzniknout otázka, zda tento proces není v rozporu se směrnicí o ochraně osobních údajů. Získávané informace ale nejsou dostatečně podrobné pro určení konkrétní osoby a vytěžení jejích citlivých údajů [2].

## b) GPS

Nejčastější způsob určení lokace, vlastněný a provozovaný Spojenými státy americkými. Navigace je složena z 31 satelitů, které vysílají rádiové signály na oběžné dráze země. Společně s ruským systémem GLONNAS tvoří mezinárodní satelitní navigaci dostupnou pro širokou veřejnost, známou jako GNSS [24]. Evropskou alternativou je pak systém Galileo, který je financován z prostředků Evropské unie. Ten má zajistit nezávislost a dostupnost polohových služeb od výše zmíněných systémů.

Spolehlivost polohových údajů je přibližně 7,8 metrů na 95 % zemského povrchu v jakémkoliv čase. Zjištění polohy probíhá na principu posílání signálů z každého satelitu do přijímače chytrého zařízení. Ta je následně vypočítána pomocí rozdílu času, kdy byl signál vyslán a kdy byl přijat. Udávaná spolehlivost časových údajů, které jsou vyslány v signálu, je extrémně vysoká díky atomovým hodinám v každém satelitu [24].

Přijímač zařízení navíc musí počítat se zpožděním, která mohou vzniknout průchodem signálu ionosférou a troposférou. S informacemi o umístění a dosahu tří satelitů může přijímač vypočítat vlastní trojrozměrnou polohu. Atomické hodiny synchronizované s GPS jsou nutné při použití pouze tří satelitů. Pokud ale využijeme údaje z dodatečného čtvrtého satelitu, tato nutnost opadá. Přijímač tedy může využít čtyři satelity k výpočtu své zeměpisné šířky, délky, výšky a času [24].

## c) Další způsoby geolokace

Mezi další způsoby geolokace patří na základě IP adresy zařízení, kdy je v databázi zmapovaných IP adres vyhledáváno přibližné umístění. Tento způsob využívají hlavně bankovní instituce k monitorování aktivit svých uživatelů. Jeho nevýhodou je nestálost polohových údajů při použití offline databáze. K monitorování lze použít i metadata souborů, jako jsou například fotky.



## 2.3 Využití geolokace

### a) Chytrá karanténa

V době pandemie koronaviru společnost Seznam představila novou funkcionalitu pro svoji aplikaci Mapy.cz. Po aktualizování na nejnovější verzi byla uživateli nabídnuta možnost zaznamenávání historie polohy. Po udělení souhlasu mapová aplikace zaznamenávala veškerou polohu společně s časovými údaji a dobou pobytu. Pokud byla uživateli pozitivně testována nákaza koronavirem, mohl přes aplikaci poskytnout svá data hygienické stanici a zpětně zmapovat další potenciálně nakažené spoluobčany [22].

### b) Zábavní průmysl

Elektronické mapové podklady s využitím geolokace našly uplatnění také v zábavním průmyslu. V červenci 2016 vydala společnost Niantic ve spolupráci s Nintendem hru Pokémon Go, která fungovala na principu rozšířené reality. Hráči pomocí mobilních zařízení s operačními systémy iOS nebo Android lovili virtuální postavičky umístěné na mapách reálného světa. Mezi další hry vydané touto společností a založené na podobném principu patřily například Ingress Prime nebo Harry Potter: Wizard Unite [23].

## 2.4 Komerčně dostupné mapové nástroje

K realizaci podnikového systému by se mohla použít i komerčně dostupná aplikační rozhraní. Při využití takového API je potřeba zvážit možnosti, které nabízí, zda je možné jej volně editovat, přizpůsobovat potřebám vývojářů samozřejmě tak, aby odezva dotazovaných informací byla co nejrychlejší a nejpřesnější. Všichni poskytovatelé si nechávají platit za určitý počet požadavků na jejich servery. Někteří nabízejí i individuální ceník, pokud se počet požadavků nevejde do jejich standardního plánu.

### 2.4.1 Google Maps API

Asi nejnámější a nejpoužívanější služba od společnosti Google, jejíž mapové podklady byly poprvé spuštěny v roce 2005. O rok později představila možnost vizuálního zobrazení Google Street View. Následně přibyly další funkcionality, jako je například hodnocení restaurací. V roce 2007 byly použity jako mapové podklady pro první chytrý telefon iPhone od společnosti Apple. Spolupráci narušilo až představení vlastního řešení od společnosti Apple v roce 2012 [3].

Google na svých webových stránkách věnovaných mapovým podkladům uvádí, že v současné době pokrývají 99 % světové plochy, která byla sestavena 1 miliardou aktivních uživatelů z více než 200 států a teritorií. Mapové podklady byly použity i při tvorbě již zmíněné hry Pokemon Go a tato možnost je propagována dalším herním vývojářům [4].

Tabulka 1: Ceník Google Maps API [28]

Platforma	Cena za 1 000 požadavků	Druh služby
Mobilní zařízení	Zdarma	Dynamická
Webové služby (Javascript API)	7.0 USD	Dynamická
Mobilní zařízení	Zdarma	Statická
Webové služby (Javascript API)	2.0 USD	Statická

#### 2.4.2 MapBox

Platforma založená na otevřeném kódu pro poskytování geolokačních dat pro mobilní a webové aplikace. Aktuálně se chlubí základnou o kapacitě šest set miliónů uživatelů měsíčně. Poprvé byla spuštěna v únoru roku 2011 a postupně uvolňovala řešení pro operační systémy iOS, Android. Následně společnost vydala i aplikace pro úpravu mapových podkladů MapBox Studio. Řešení pro herní vývojáře jako plugin do Unity frameworku. V srpnu roku 2017 oznámilo podporu pro procesory ARMKit [5].

Tabulka 2: Ceník Mapbox API pro mobilní zařízení [29]

Počet požadavků	Cena za 1 000 požadavků
0–25 000	Zdarma
25 001–125 000	4.00 USD
125 001–250 000	3.20 USD
250 001–1 250 000	2.40 USD
1 250 000 +	Individuální plán

Tabulka 3: Ceník Mapbox API pro webové rozhraní [29]

Počet požadavků	Cena za 1 000 požadavků
0–50 000	Zdarma
50 001–100 000	5.00 USD
100 001–200 000	4.00 USD
200 001–1 000 000	3.00 USD
1 000 000 +	Individuální plán

## 2.5 OpenStreetMap

Cílem projektu OpenStreetMap je vytvořit bezplatnou a upravitelnou mapu celého světa pod otevřenou licenci. Geografická data jsou v mnoha zemích zpoplatněna nebo poskytována uživatelům pod komerčně chráněnými autorskými právy. Tvorba mapových podkladů některých zemí je dokonce hrazena z veřejných daní a následně uživatelům znovu zpoplatněna pro využití. Například v USA jsou surová data mapových podkladů (například TIGER) veřejně přístupná, hotové mapové podklady jsou ale chráněny autorskými zákony [6].

K odhalení neoprávněně využitých mapových pokladů jsou využívány chytáky (tzv. Easter Egg) v podobě falešných údajů v mapových podkladech. Může se jednat například o chybně označené ulice, neexistující podniky nebo vzdělávací instituce. Z hlediska vymahatelnosti porušení zákona se pak stávají tyto chytáky důkazem. Aktualizace těchto mapových podkladů je také problematická, data jsou navíc formátována tak, že je nelze vlastním úsilím upravit pro své potřeby. Využívání fotokopie surové mapy může být také nezákonné [6].

Všechna mapová data byla tvořena dobrovolníky z oblastí po celém světě. Může tedy vzniknout otázka, že komunitní projekty mohou obsahovat ať už záměrné nebo nechtěné chyby. Přece jenom každý přispěvatel může přidat chybné informace. Zde ale funguje komunitní kontrola. Pokud jeden uživatel vloží špatnou informaci, mohou ho ostatní poctiví uživatelé opravit a tím ji zneplatnit [6].

Tento princip využívá například navigace Waze, v níž mohou řidiči hlásit na úseku mimořádné události (například policejní měření rychlosti). Pokud ale mimořádná událost zmizí (hlídka opustí danou lokalitu), stává se tato informace nerelevantní. Další uživatelé, kteří tímto úsekem projedou, mohou v notificačním listě hlasováním určit, zda je tato informace stále pravdivá. Pokud se více řidičů nezávisle na sobě shodne na nepravdivosti, je tato informace odstraněna a už není neposkytována.

### 2.5.1 Historie

Počátky OpenStreetMap se mapují, až do roku 2004, kdy byl tento projekt založen programátorem Stevem Coastem. Za cíl měl pouze mapování ulic měst ve Velké Británii. Mapové podklady v té době byly tvořeny pouze z vládních projektů, které nebyly efektivní a často zpoplatněné. V roce 2006 se skupina kolem Steva Coasta rozrostla a vznikl projekt „OpenStreetMap Foundation“, který měl za cíl poskytnout svobodné mapové poklady každému uživateli [7].

Začaly vznikat první nástroje pro úpravu mapových dat. Mezi doposud nejznámější patří nástroj JOSM, který byl napsán v jazyce Java a je nadále udržován v aktuální verzi. V roce 2008 projekt vyvinul nástroje pro exportování mapových podkladů a jejich následný import do GPS zařízení, která do té doby používala zastaralé mapy. V současné době se počet uživatelů OpenStreetMap odhaduje na více než šest miliónů [7].

### 2.5.2 Licencování

Data obsažená v OpenStreet mapách jsou dostupná pod svobodnou licencí za podmínek ODbL. Uživatel může kopírovat, distribuovat a libovolně upravovat při uvedení zdroje a přispěvatelů. Kartografická díla obsažená v mapových podkladech (dlaždicích) jsou k dispozici pod licencí Creative Commons [8].

Autorství je vyžadováno ve formátu „© Přispěvatelé OpenStreetMap“ s uvedením, že data jsou dostupná za podmínek CC-BY-SA, přičemž je možné tuto podmínku splnit pouze odkazem na oficiální stránky vývojářů do sekce copyright. V případě vykreslování mapy by mělo být uvedeno v pravé dolní části obrázku pod licencí Open Database License. Přispěvatelům je zakázáno zahrnovat do kartografických dat obsah autorsky chráněných dokumentů bez výslovného svolení držitelů autorských práv [8].

### 2.5.3 Struktura

Základní komponentou OpenStreetMap® jsou tzv. „elementy“ které rozdělujeme podle jejich významu na tři druhy:

- Uzel (Node).
- Trasa (Way).
- Vztah (Relation).

**Uzel** specifikuje bod na planetě Zemi popsany koordináty, nejčastěji zeměpisnou šířkou a délkou. Může se jednat o jakýkoliv neměnný objekt umístěný na mapě, např. světelná signalizace nebo vysílací věž [9].

**Cesta** je seřazená množina dvou a více uzlů, které tvoří křivku. V OpenStreetMap je maximální počet uzlů cesty omezen na 2 000. Poté už nelze element reprezentovat jako cestu a je třeba využít jinou datovou strukturu pro reprezentaci [9]. Cesty mají následující dělení:

- **Uzavřené cesty (Closed way)** – jejich počáteční bod je zároveň i koncovým.
- **Otevřené cesty (Open way)** – jejich počáteční a koncový bod jsou rozdílné.
- **Oblasti (Area)** – uzavřená cesta reprezentuje výplň.

Typickým příkladem cest jsou silnice, cesty, řeky nebo zalesněné oblasti [9]. Cesty se dále dělí na jednosměrné a vícesměrné. Tato informace bývá obsažena v tagu s klíčem **oneway**. Hodnota pak určuje vícecestnou nebo jednocestnou ulici:

Ukázka zdrojového kódu 1: Označení druhu cesty

```
<tag k="oneway" v="yes"/>
```

- **Yes** definuje jednocestnou ulici (alternativně lze použít i true nebo 1).
- **No** definuje vícecestnou ulici (alternativně lze použít i false nebo 0).
- **Reverse** definuje jednocestnou ulici, ale v opačném směru, než byla zaznamenána (častěji se používá hodnota -1).
- **Alternating** – hodnota informuje o častém a pravidelném střídání směru cesty.
- **Reversible** – hodnota informuje o střídání stavů této cesty, které ale nemají pravidelnost.

**Vztah** udává nějaký druh vazby mezi jednotlivými elementy. Může vyjadřovat například omezení na silničních komunikacích, kde dvě trasy mohou být ve vztahu nepojízdnosti [9].

## 2.5.4 Druhy cest

Každá cesta se může lišit v závislosti na jejím účelu, přístupech pro veřejnost nebo stavu. Je důležité vzít v potaz, že kategorizaci vozovek upravuje v každé zemi příslušný zákon a informace obsažené v OpenStreetMap datech a že jsou vždy v souladu s daným zákonem. Při tvorbě aplikací je nutné vzít

v potaz, že označení jedné země nemusí být nutně použitelné v ostatních státech. V České republice kategorizaci vozovek upravuje zákon o pozemních komunikacích <sup>1</sup>[10].

Tabulka 4: Klasifikace cest v České republice [35]

Hodnota klíče	Druh cesty	Znění zákona
motorway	Dálnice	Dálnice je pozemní komunikace určená pro rychlou dálkovou a mezistátní dopravu silničními motorovými vozidly, která je budována bez úrovnových křížení, s oddělenými místy napojení pro vjezd a výjezd a která má směrově oddělené jízdní pásy.
primary	Silnice I. třídy	Silnice I. třídy, která je určena zejména pro dálkovou a mezistátní dopravu.
	Silnice II. třídy	Silnice II. třídy, která je určena pro dopravu mezi okresy,
	Silnice III. třídy	Silnice III. třídy, která je určena k vzájemnému spojení obcí nebo jejich napojení na ostatní pozemní komunikace.
living_street	Obytná zóna	Obytná zóna je zastavěná oblast, jejíž začátek je označen dopravní značkou „Obytná zóna“ a konec je označen dopravní značkou „Konec obytné zóny“.
pedestrian	Pěší zóna	Pěší zóna je oblast, jejíž začátek je označen dopravní značkou „Pěší zóna“ a konec je označen dopravní značkou „Konec pěší zóny“.

### 2.5.5 Definice domů

Číselné označení domů je v OpenStreet mapách uvedeno v tagu s klíčem „addr:housenumber“. Pro většinu států by tato definice stačila. Pro Českou republiku je situace komplikovanější, protože popisný systém domů může obsahovat více označení. A ta mohou být ještě kombinována s písmeny abecedy.

<sup>1</sup> Komplettní popis cest (i pro Českou republiku) je možné nalézt na oficiálních stránkách OpenStreetMap [35]

- a) **Číslo popisné** se využívalo ke značení domů již za vlády Marie Terezie. Tehdy ještě nemělo trvalý charakter a jednalo se o označení v konkrétním soupise, které provádělo vojsko při průchodu místem. V dnešní době mají čísla trvalou platnost, avšak žádné logické uspořádání. Při stavbě nového domu se jednoduše přiřadí další volné číslo v pořadí daného území, což způsobuje při orientaci značné problémy. V mapových podkladech se číslo popisné značí tagem „`addr:conscriptioinnumber`“ [11].
- b) **Čísla orientační** jsou nepovinnou informací, která mají značit pořadí budov v rámci ulice nebo od středu náměstí. Problém může nastat, pokud se postaví nový dům mezi již očíslovanými budovami. Pak je číslo orientační doplněno o písmeno české abecedy neobsahující diakritiku. I když mají mapové podklady pro orientační číslo vymezený tag „`addr:streetnumber`“, můžeme se setkat s oficiálním značením ve tvaru číslo popisné, lomítko, číslo orientační vsazené do tagu „`addr:housenumber`“. Tento způsob umožňuje zajistit kompatibilitu se službami třetích stran, které nemusí být s problematikou značení v České republice dostatečně obeznámeny [11].
- c) **Čísla evidenční** jsou obdobou čísel popisných a značí pouze dočasné budovy nebo stavby rekreačního charakteru. Označení v mapových podkladech je realizováno tagem „`addr:provisionalnumber`“ [11]

Ukázka zdrojového kódu 2: Uzel popisující budovu v mapových podkladech

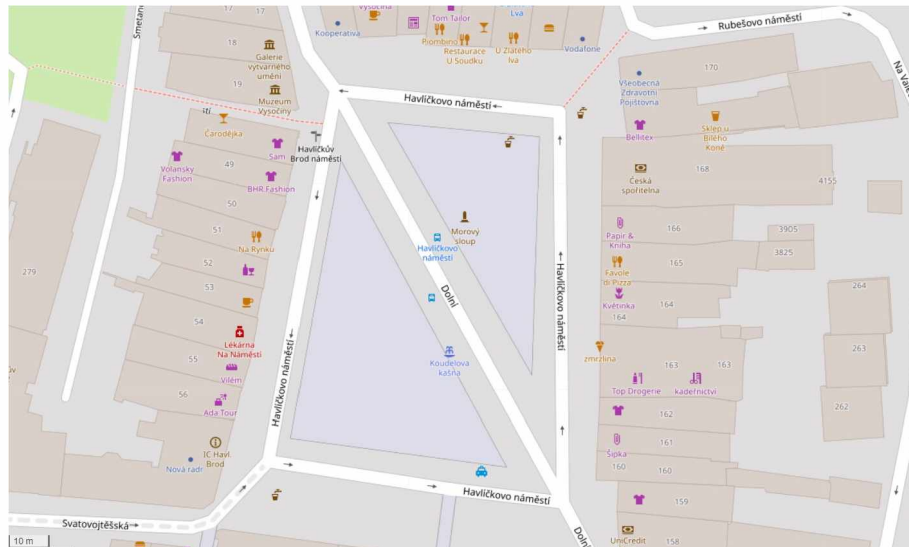
```
<node id="2299477691" user="JandaM" uid="2169558"
lat="49.6023015" lon="15.5780928">
  <tag k="addr:city" v="Havlíčkův Brod"/>
  <tag k="addr:conscriptioinnumber" v="1108"/>
  <tag k="addr:country" v="CZ"/>
  <tag k="addr:housenumber" v="1108"/>
  <tag k="addr:postcode" v="58001"/>
  <tag k="addr:street" v="Jihlavská"/>
</node>
```

## 2.5.6 Definice ulic

Ulice jsou v OpenStreet mapách definovány jako cesty s více body. Přičemž jeden z bodů se může opakovat v definici dalších cest v OSM souboru. Takto je možné do mapových podkladů zahrnout i složité úseky. V následujícím příkladu je ukázáno, jak je definována část ulice. Jako demonstrační příklad posloužilo Havlíčkově náměstí v Havlíčkově Brodě.

Pro následující část mapy byl vyexportován OSM soubor:

Ukázky souborů jsou zkráceny o některé atributy a tagy, které ztrácejí význam pro demonstrační ukázkou. Údaje o autorovi dané informace byly pro zachování licence ponechány.



Obrázek 1: Havlíčkovo náměstí vyobrazené v OpenStreetMap [32]

V souboru je definována následující cesta:

Ukázka zdrojového kódu 3: Cesta na Havlíčkovo náměstí definovaná v mapovém podkladu

```
<way id="381435419" user="Milancer"
uid="1189910">
  <nd ref="343571099"/>
  <nd ref="30103261"/>
  <tag k="hgv" v="destination"/>
  <tag k="highway" v="residential"/>
  <tag k="name" v="Havlíčkovo náměstí"/>
  <tag k="oneway" v="yes"/>
```

Z vnořného tagu můžeme zjistit, že se jedná o jednosměrnou ulici. První dva tagy s označením **nd** jsou zároveň koncový a počáteční bod. Way tagy obsahují v rámci úspory místa pouze referenci na id uzlu, nikoliv souřadnice bodu.

Ukázka zdrojového kódu 4: Reference na uzly obsažené v cestě.

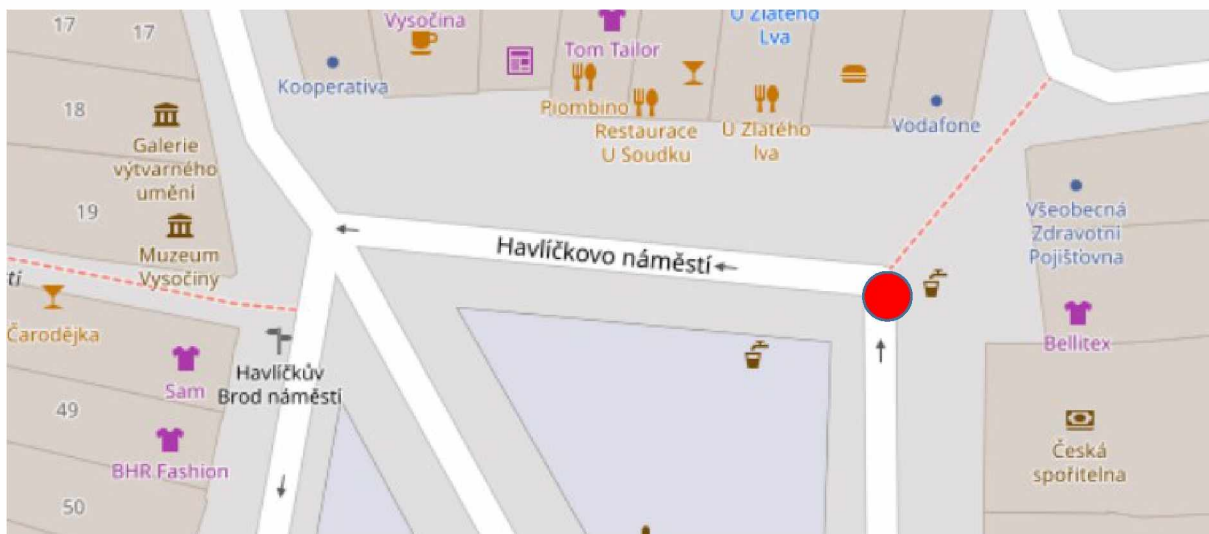
```
<nd ref="343571099"/>
<nd ref="30103261"/>
```

První z uzlů má souřadnice 49.6073050 a 49.6073050. V mapě je vyznačen červeným bodem.



Ukázka zdrojového kódu 5: Zeměpisná šířka a délka v definici prvního uzlu.

```
<node id="343571099" user="hanoj" uid="4788" lat="49.6073050"  
lon="15.5798013"/>
```

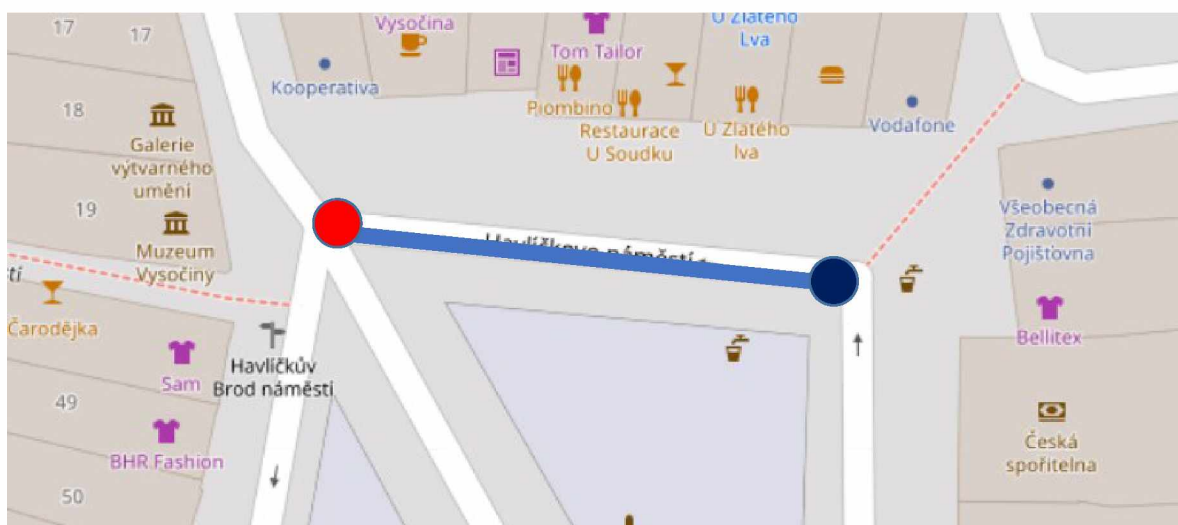


Obrázek 2: Vyznačení prvního uzlu na mapovém podkladu [32]

Druhý z bodů má souřadnice 49.6073519 a 15.5790553. V mapě je vyznačen červeným bodem. Počáteční původně zmiňovaný bod má nyní modrou barvu. Cesta, kterou body symbolizují, je vyznačena světle modrou barvou.

Ukázka zdrojového kódu 6: Zeměpisná šířka a délka v definici druhého uzlu.

```
<node id="30103261" user="hanoj" uid="4788" lat="49.6073519"  
lon="15.5790553"/>
```



Obrázek 3: Vyznačení koncového uzlu a cesty, která jej spojuje [32]

Při dalším zkoumání souboru zjistíme, že druhý z uvedených bodů je definován v další cestě jako počáteční bod:

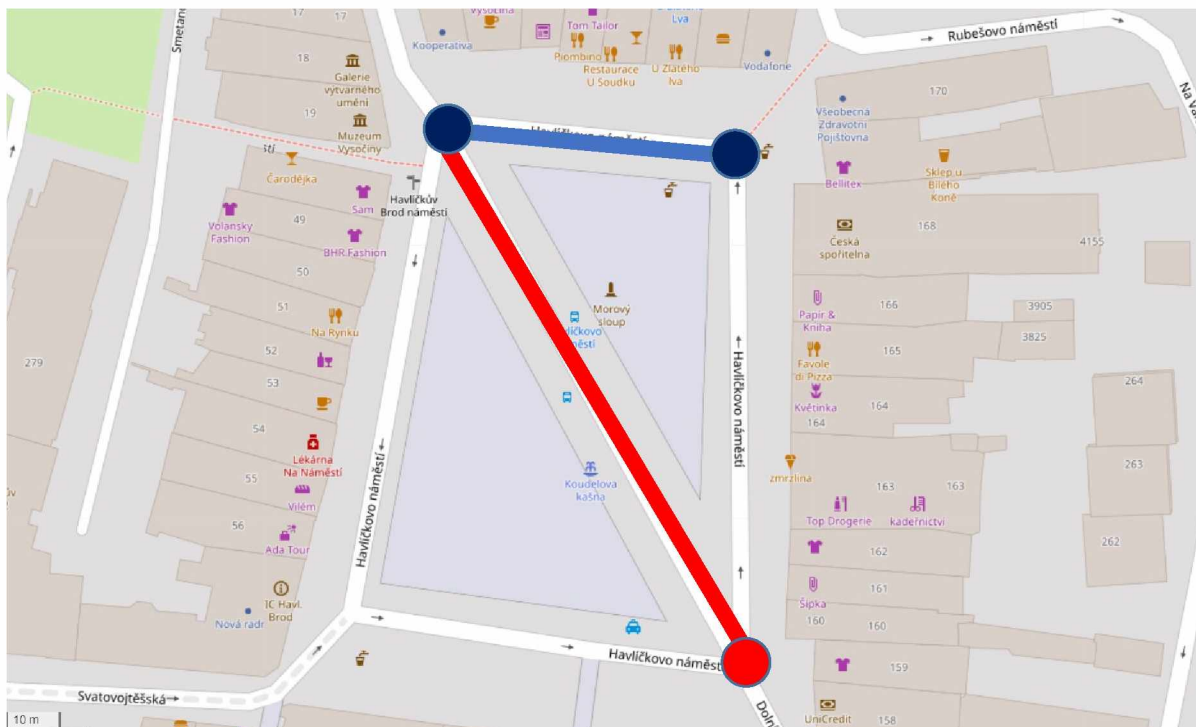
Ukázka zdrojového kódu 7: Ukázka druhé cesty, která je ve vztahu s předchozí.

```
<way id="229138258" user="Milancer" uid="1189910">  
  <nd ref="30103260"/>  
  <nd ref="30103261"/>  
  <tag k="hgv" v="destination"/>  
  <tag k="highway" v="residential"/>  
  <tag k="name" v="Dolní"/>  
</way>
```

Koncový bod se souřadnicemi 49.6064629 a 15.5798127 společně s nově objevenou cestou je na mapě vyznačen červenou barvou.

Ukázka zdrojového kódu 8: Koncový bod druhé cesty.

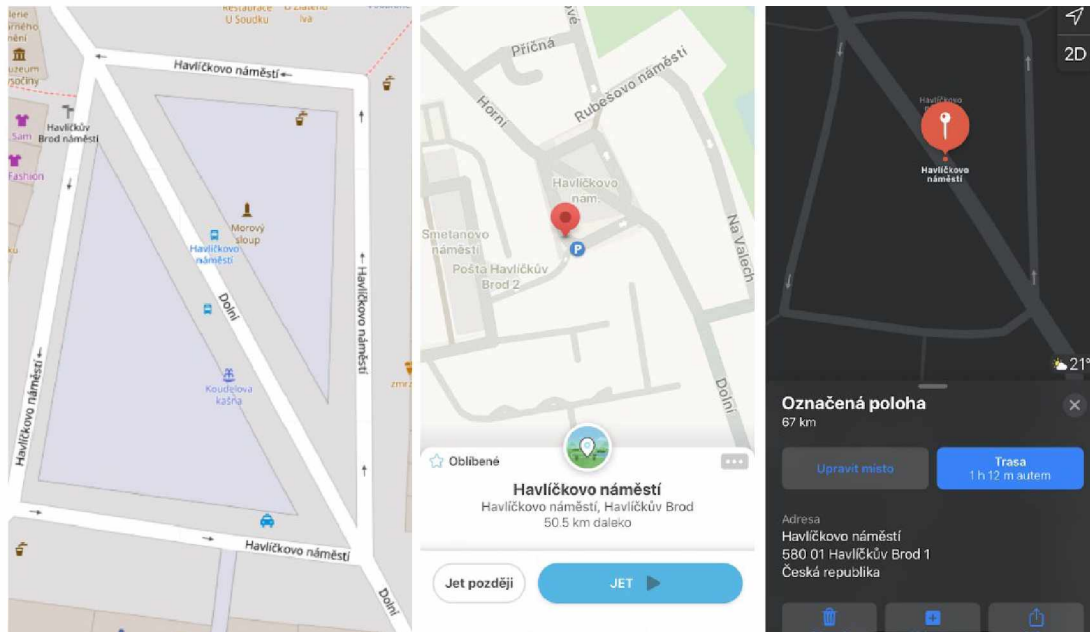
```
<node id="30103260" user="vrabcak" uid="8007" lat="49.6064629"  
lon="15.5798127"/>
```



Obrázek 4: Nová cesta společně se svým koncovým bodem [32]

### 2.5.7 Chybovost mapových podkladů

Při zkoumání ukázek XML souborů je možné si všimnout, že nově objevená cesta má jiný název ulice (Dolní). Jedná se o chybu, která byla autorem odhalena náhodou při tvorbě demonstrační aplikace. Pokud využijeme informace od jiných mapových dodavatelů, zjistíme, že název ulice je stále Havlíčkovo náměstí:



Obrázek 5: Ukázka chybně označené ulice v OpenStreetMap, navigaci Waze a Apple mapách

Předchozí příklad ukázal, že se nelze na mapové podklady OpenStreetMap vždy spolehnout. Přece jenom je jejich obsah tvořen komunitou dobrovolníků. Na popisu výše uvedené trasy se podíleli tři různí uživatelé (za předpokladu, že pod více přezdívkami nevystupuje jeden uživatel). Aktuálnost informací tedy závisí čistě na dobré vůli komunity.

### 2.5.8 Problematika adresace v OpenStreet mapách

Vzhledem k tomu, že se jedná o komunitní projekt, nemají OpenStreet mapy žádná pevná pravidla, jak zapisovat adresy do mapových podkladů. Situaci navíc komplikují zákony a normy, které se mohou lišit v každém státu. Obecně by se tyto informace měly zapisovat do tagů s klíčovým prefixem „addr“ [12].

Všechny informace by pak měly být uvedeny u každého uzlu nebo cesty samostatně. Nicméně existují i alternativy v podobě vztahů, kdy je ulice vyjádřena jedním tagem a pomocí vztahových operací jsou k ní pak přidruženy cesty a uzly. Byť by se jednalo o logičtější a datově méně náročný způsob, je využíván jen zlomkem procent uživatelů. Na oficiálních stránkách se dodnes vedou diskuze, zdali tento způsob označit jako jediný správný [12].

Následující tabulka ukazuje adresaci v České republice:

Tabulka 5: Doporučený zápis adresy pro Českou republiku [34]

Klíč	Popis	Očekávaná hodnota	Příklad
addr:housenumber	Číslo domovní	Kombinace čísel, písmen a speciálních znaků	2238/1a
addr:street	Ulice	Název ulice	Chotěbořská
addr:place	Vesnice nebo místní část obce	Část obce nebo městské čtvrti	Vysočany
addr:city	Značení města	Název města	Brno
addr:postcode	Poštovní směrovací číslo	Poštovní směrovací číslo ve tvaru pětimístného čísla	580001
addr:country	Název státu	Název státu	Česká republika

Popis ale nelze v plném rozsahu aplikovat na cesty. Pokud exportujeme mapové podklady **Havlíčkov** **Brodu**, zjistíme, že místní cesty jsou osazeny tagem „addr:street“ mnohem hustěji než v případě **Hradce Králové**. Podle komunity OpenStreetMap, která odpověděla na moji otázku ohledně dané problematiky, slouží atribut „addr:street“ spíše k označení budov nebo objektů, které nerepresentují ulici jako takovou, ale mají k ní pouze vztah. Mnohem univerzálnější způsob je zapsat název ulice do tagu s klíčem „name“ a osadit danou cestu dalším dodatečným tagem s klíčem „highway“, který popisuje druh cesty.

### 2.5.9 Uložení mapových podkladů

Existuje více způsobů, jakým jsou uchovány mapové podklady OpenStreetMap:

- **PBF formát** ukládá informace do komprimovaného binárního souboru.
- **OSM** ukládá data do XML souboru. K ukládání dat je možné použít i JSON formát.
- **o5m** poskytuje vysokorychlostní zpracování dat za využití PBD komprese. Mezi hlavní nevýhody patří žádná nebo jen částečná podpora některých nástrojů.
- **Level0L** reprezentuje data ve více čitelné podobě pro běžného uživatele [13].

### 2.5.10 OSM XML

Mapová data jsou strukturována ve formátovaných XML souborech označovaných příponou „osm“. Soubory obsahují informace o elementech ve formě tagů ve vztahu 1:N, kdy jeden element může mít více přidružených tagů. Jako dodatečné informace může každý tag obsahovat atributy [14]:

Tabulka 6: Obecné atributy elementu v OSM souboru [14]

Název	Datový typ	Popis
id	Celé číslo	Identifikátor elementu.
user	Řetězec	Autor poslední modifikace objektu.
uid	Celé číslo	Identifikátor autora poslední modifikace objektu.
timestamp	W3C časový formát	Datum a čas poslední modifikace.
visible	Boolean	Viditelnost objektu.

### 2.5.11 PBF formát

Je alternativou k OSM XML souboru a byl vyvíjen za účelem podpory efektivnosti a flexibility stále rostoucího množství datových údajů. Data celé planety uložená do PBF souboru zaberou přibližně o 30 % méně místa než při použití standardního formátu OSM komprimovaného bzip2 algoritmem. Soubor je rozdělen do bloků a skupin, které mohou být dekodovány nezávisle na sobě. V reprezentaci dat využívá entity, které jsou uloženy v podobě řetězce. Každá entita obsahuje 64bitový primární klíč, který ji reprezentuje napříč datovými podklady [15].

## 3 TVORBA GEOLOKAČNÍ KOMPONENTY

Geolokační komponenta je nejdůležitější část aplikačního rozhraní. Jedná se o část projektu, se kterou následně komunikuje aplikační rozhraní serveru při obdržení požadavku na reverzní geokódování nebo mapové renderování. Stahování a následné zpracování mapových podkladů zajišťuje třída **OsmMapComponent**, optimalizovanou verzi s cashováním mapových podkladů pak její potomek, třída **SectorMapComponent**. Metody pro reverzní geokódování, matematické výpočty a vyhledávání informací v OpenStreetMap entitách obsahuje třída **AddressTriangulation**.

### 3.1 Použité nástroje a programovací jazyky

Mapová komponenta je naprogramována v jazyce Java od společnosti Oracle. Důvod výběru programovacího jazyka spočíval hlavně v možnosti použití objektově orientovaného přístupu, rozsáhlého množství knihoven třetích stran pro práci s OpenStreetMap a podpora frameworku Spring. Celý projekt je vyvíjen ve verzi JDK 14.

Historie jazyka sahá až do roku 1991, kdy jej vyvíjela firma Sun Microsystems primárně pro vestavěné systémy na principech C a C++. Původní název jazyka byl Oak, kvůli shodě s již existujícím jazykem tohoto názvu byl přejmenován na Java neboli slangově kafe. Jazyk prochází celkem pěti fázemi překladu: editováním, překladem neboli kompilací, zavedením, ověřováním a prováděním. Výsledný kód je přeložen do Java byte kódu, který následně spouští JVM [21].

#### 3.1.1 IntelliJ IDEA

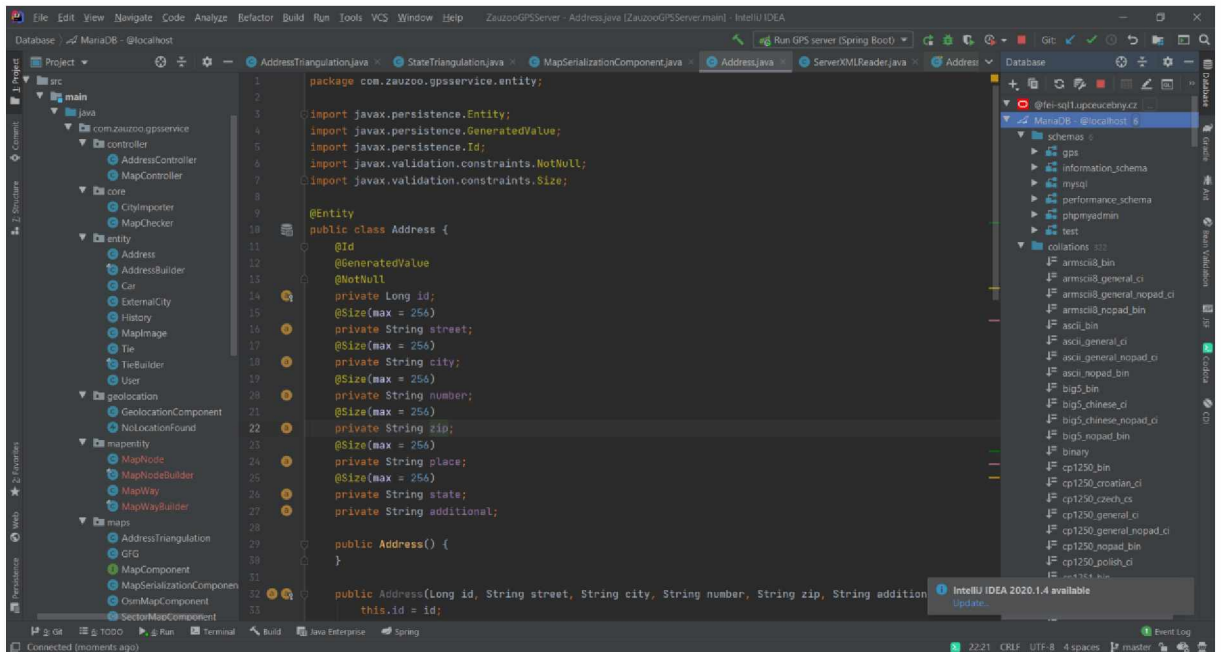
Vývojové prostředí vyvinuté společností JetBrains s.r.o. (dříve IntelliJ Software s. r. o.) sídlící v Praze Na Hřebenech II. První verze nástroje vyšla již v lednu roku 2001 a oproti tehdejším konkurenčním nástrojům nabídla jako první funkce pro formátování kódu v jazyce Java. Od té doby vycházejí pravidelné verze ve čtvrtletních kvartálech. Od verze 9.0 je rozdělena do dvou edicí [16].

**Community Edition (komunitní verze)** má otevřený zdrojový kód pod licencí Apache 2.0. Obsahuje všechny základní funkcionality a je dostupná pro každého uživatele k použití zdarma [17].

**Ultimate Edition (ultimátní verze)** je komerční nástroj, který obsahuje všechny funkcionality komunitní verze plus další dodatečné. Některé z nich jsou zde vyjmenovány [17]:

- Podpora ostatních programovacích jazyků a nástrojů, které jsou jinak dostupné v dalších produktech této firmy.
- Rozsáhlejší podpora frameworků třetích stran, mezi něž patří například Spring, Play a ReactJs.
- Rozsáhlejší podpora pro nasazování serverových aplikací.

Roční cena pro jednoho uživatele se pohybuje okolo 13 000 Kč. Pro vzdělávací účely je dostupný zdarma všem studentům, kteří jakkoliv prokáží platnost svého studia [17].



Obrázek 6: Vývojové prostředí IntelliJ s postranními panely se strukturou projektu a databázovými připojeními

### 3.1.2 Osmosis

Osmosis je konzolová aplikace napsaná v jazyce Java dostupná pro operační systémy Microsoft Windows, OS X a linuxové distribuce. Mezi hlavní funkcionality patří čtení a zápis do OpenStreetMap souborů, import a vyhledávání dat v databázi. Veškeré moduly, které aplikace používá, jsou dostupné v oficiálním Maven repozitáři a zdrojové kódy je možné získat na oficiálním GitHub repozitáři [18].

23. října 2018 vývojáři oznámili, že po 11 letech ukončují aktivní tvorbu nástroje. Tehdy vydaná verze s označením 0.47 byla tedy jejich posledním oficiálním počinem. Další funkcionality a aktualizace pro podporu nových mapových podkladů bude pravděpodobně tvořit komunita. Osmosis je v serverové aplikaci využíván při zpracování osm a pbd souborů s daty mapových podkladů [18].

Před spuštěním procesu čtení se čítači nastavuje takzvaný „Sink“. Tento termín by se volně dal přeložit jako „dřez“. Jedná se o interface, jehož metody může vývojář modifikovat a vložit kód, který se má aktivovat při událostech čtení. Hlavní kód je implementován do metody process, která v parametru zpřístupní abstraktní třídu „EntityContainer“. Jedná se o kontejner, který přenáší OpenStreetMap entit.

## **3.2 Proces reverzní geolokace**

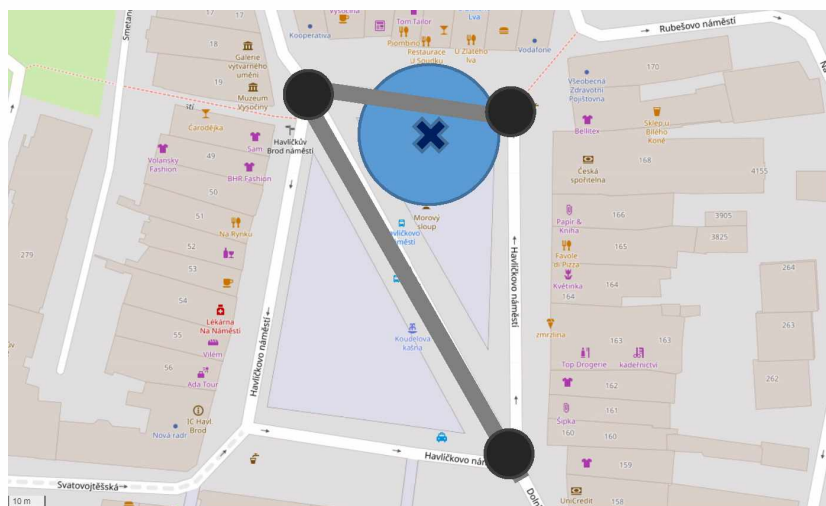
Proces započne zpracováním přichozího požadavku na server. Ten vyžaduje v parametru dodání zeměpisné šířky a délky. Tyto informace jsou předány do geolokační komponenty, kde se určí severní, jižní, západní a východní bod mapové oblasti, do níž hledaný bod spadá. Pokud nebyla mapová oblast doposud zpracována, je zavoláno API OpenStreetMap serveru uvedenému v konfiguračním souboru, který zahájí stahování souboru na daných souřadnicích.

Následně jsou ze souboru za pomoci nástroje Osmosin získány potřebné informace k adresaci daného bodu, tedy pouze uzly a cesty obsahující tagy mající nějaký vztah k adresaci. Poté jsou tyto informace zaneseny do hashovací mapy a předány k dalšímu zpracování.

### **3.2.1 Určení ulice**

Jako první se pokusíme určit ulici objektu. Stačí k tomu využít jednoduchou středoškolskou matematiku. Každá cesta je vlastně přímka určená koncovým a počátečním bodem. Bod je vyjádřen zeměpisnou šířkou, která představuje souřadnici x a zeměpisnou délkou představující souřadnici y. Umístění zařízení je také souřadnicový bod, který je doplněn o poloměr tolerance chybovosti geolokace.





Obrázek 7: Vyobrazení přibližné uživatelské polohy a cest reprezentující ulice [32]

V demonstračním příkladu jsou vyznačeny světle šedou barvou dvě cesty, které byly získány z mapového souboru. Tmavší barvou jsou zvýrazněny jejich koncové, počáteční a spojové body. Modrý křížek označuje uživatelskou polohu a modrý kruh toleranci chybovosti. Pokud z demonstračního příkladu odebereme mapové pozadí, získáme vlastně geometrické objekty.

Stačí pouze určit, jestli se cesta dá popsat jako přímka protínající kruh. K tomu poslouží jednoduchý vzorec, který určí vzdálenost bodu od přímky:<sup>2</sup>

$$d = \frac{|a * x + b * y + c|}{\sqrt{a^2 + b^2}}$$

Nejdříve musíme určit rovnici přímky k získání konstant  $a$ ,  $b$  a  $c$ . K tomu určíme vektory  $V_1$  a  $V_2$  obou souřadnic. Předpokládejme, že  $x_1$  a  $y_1$  jsou počáteční souřadnice přímky a  $x_2$  a  $y_2$  jsou koncové souřadnice přímky.

$$V_1 = [x_2 - x_1]$$

$$V_2 = [y_2 - y_1]$$

Následně se určí normálový vektor s označením  $n$ . Jehož hodnoty jsou vektory  $V_2$  a  $V_1$ . U jedné ze souřadnic otočíme znaménko. Pokud je hodnota  $V_2$  kladná, zvolíme raději druhou hodnotu  $V_1$ , neboť první z hodnot normálového vektoru musí být kladná.

$$\vec{n} = (\pm V_1; \pm V_2)$$

Hodnoty normálového vektoru představují konstanty  $a$ ,  $b$  v rovnici přímky. K získání poslední hodnoty  $c$  musíme do rovnice přímky dosadit nově získané konstanty, zvolit libovolný bod na přímce

<sup>2</sup> Určení obecné rovnice přímky: [30]

(včetně koncového a počátečního) a dosadit do přímky. Následně stačí matematickými úpravami vyjádřit konstantu  $c$ .

$$c = -(\vec{n}_1 * x + \vec{n}_2 * y)$$

V případě strojového zpracování odpadá nutnost volit bod na základě náročnosti výpočtu. Lze tedy klidně zvolit počáteční, nebo koncový bod.

### 3.2.2 Určení čísla domu

Domy jsou určeny samostatným bodem. Takže by k výpočtu stačila pouze vektorová vzdálenost. Tento proces není tak jednoduchý, pokud se jedná o geografická data, kde by se mělo vzít v potaz i zakřivení zemského povrchu. K výpočtu vzdálenosti dvou geografických bodů se používá takzvaná Haversinova formule [31]. Ta je vyjádřena jako:

$$d = 2r \sin 2 \left( \sqrt{\sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

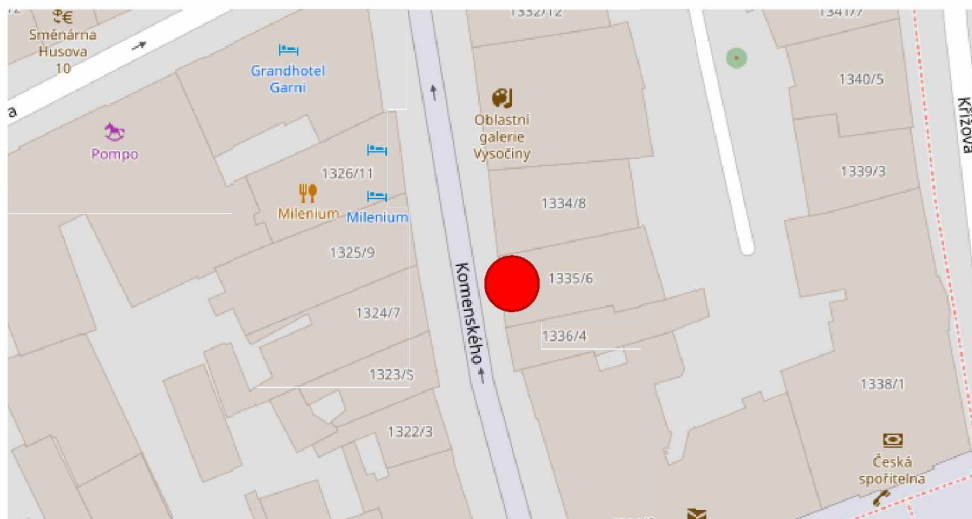
Kde  $r$  je poloměr země,  $\phi_2, \phi_1$  jsou zeměpisné délky a  $\lambda_2, \lambda_1$  zeměpisné šířky obou bodů. I při použití této formule se však nedočkáme přesných výsledků. Při výpočtu se předpokládá, že země je dokonalá koule, kdežto ve skutečnosti se jedná o zaoblený sféroid.

Může tedy vzniknout otázka, proč není zakřivení bráno v potaz i při určování ulice. Tyto objekty jsou vždy velmi blízko od sebe, protože známe uživatelskou přibližnou polohu, zakřivení je tedy zanedbatelné a ve výsledku by stačila pouhá vektorová vzdálenost. Pro budoucí účely aplikace je však v projektu využit výpočet Haversinovy formule.

### 3.2.3 Určení města

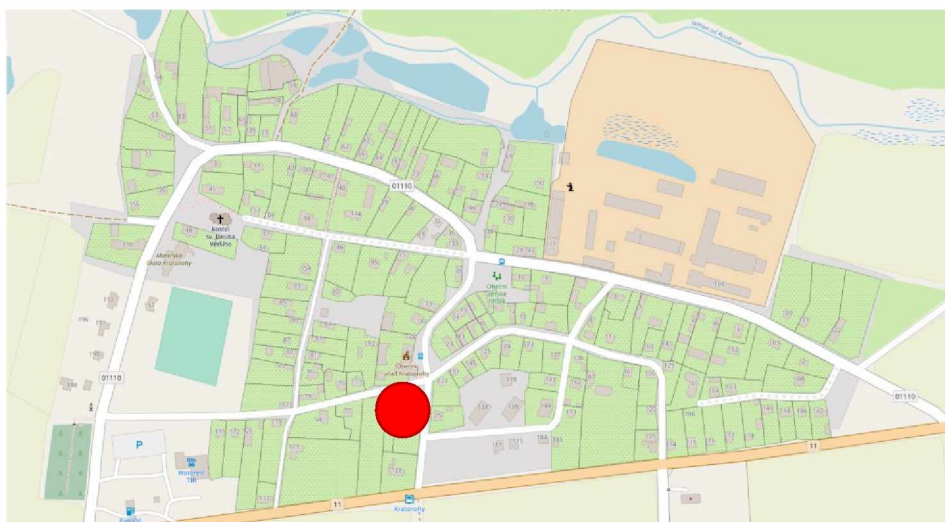
Nečekaný problém může nastat v případě určení města zkoumaného bodu. Tato informace může být obsažena kdekoli v staženém souboru. Záleží zde opět na dobré vůli dobrovolníků, zdali uvedou vždy co nejpodrobnější popis u každé informace. Je tedy dobré vhodně zvolit měřítko stažené oblasti!

V následujícím příkladu je ukázána mapová oblast nejmenšího krajského města Jihlavy. Červeným bodem je označena zkoumaná adresa. Pokud stáhneme mapový soubor dané oblasti, zjistíme, že informace o městu je uvedena hned šestkrát v okolních bodech. Určení tedy nedělá problém.



Obrázek 8: Mapová oblast Jihlavy [32]

Platí tedy, že čím menší je zalidněnost dané oblasti, tím větší je pravděpodobnost, že se nám podaří danou informaci nalézt. V dalším příkladu je ukázána mapová oblast obce Kratonohy ležící v okrese Hradec Králové. Rozloha obce činí 11,34 km<sup>2</sup> a má okolo 600 obyvatel. Pokud stáhneme mapovou oblast celé obce, zjistíme, že informace o ní je uvedena téměř u každého domu.

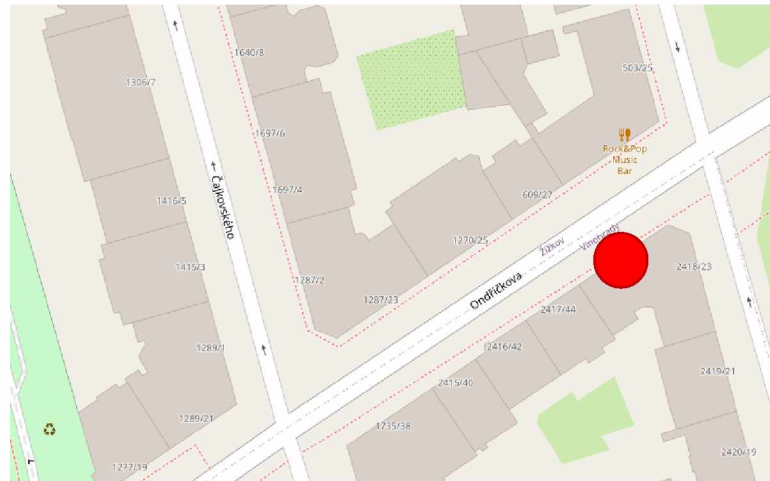


Obrázek 9: Mapová oblast obce Kratonohy [32]

Ukázka zdrojového kódu 9: Adresace domu v obci.

```
<node id="658497395" user="CzechAddress" uid="1942179"
  <tag k="addr:country" v="CZ"/>
  <tag k="addr:place" v="Kratonohy"/>
  <tag k="addr:postcode" v="50324"/>
</node>
```

Pokud ale přejdeme do nějaké více zalidněné oblasti, kde uvádění podrobnějších informací způsobí datovou zahlcenost, můžeme tuto informaci hledat marně a pravděpodobně nebude v souboru uvedena vůbec. V následujícím příkladu je ulice Ondříčkova v Praze. Pokud stáhneme mapovou oblast, zjistíme, že informace o hlavním městě České republiky je uvedena pouze v tagách s informačním charakterem. Nelze se tedy na ně se stoprocentní jistotou spolehnout.

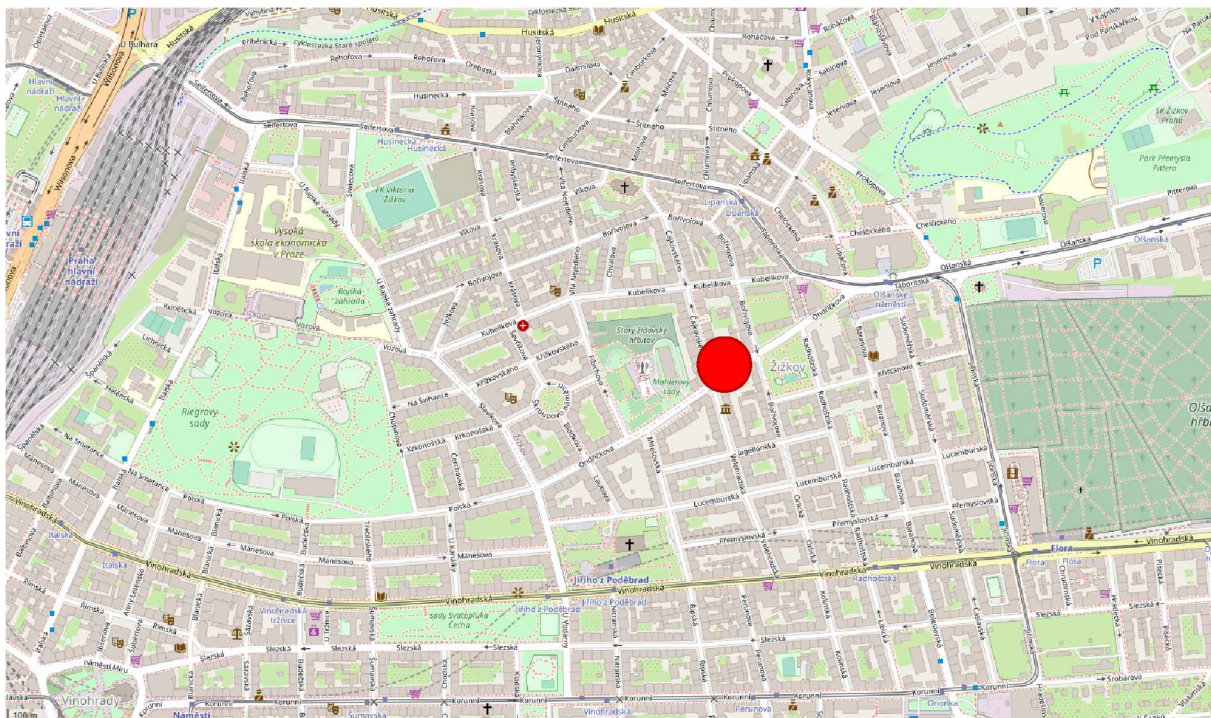


Obrázek 10: Mapová oblast Ondříčkovy ulice v Praze [32]

Ukázka zdrojového kódu 10: Označení hlavního města, nevhodné ke zpracování

```
<node id="3429944734" user="Kudej" uid="1558400">  
  <tag k="highway" v="crossing"/>  
  <tag k="source" v="CZ:IPRPraha:ortofoto"/>  
</node>
```

Musíme tedy rozšířit měřítko stažené oblasti. V posledním příkladu je měřítko hlavního města dostatečně velké, abychom informaci získali. V souboru je poté obsažena na 277 místech (k 26. 07. 2020). Nicméně výsledný soubor měl velikost přes 22 MB, což způsobilo nárůst výpočetní náročnosti. Je tedy potřeba vždy měřítko zvolit co nejvhodněji.



Obrázek 11: Mapová oblast Prahy [32]

### 3.2.4 Určení státu

OpenStreet mapy popisují státy pomocí polygonů reprezentujících hranice daného státu nebo tagem „`addr:country`“. Tyto informace ale při stahování malých oblastí nemusí být vždy dostupné. Pokaždé zpracovávat soubor jenom kvůli zjištění státu se jeví také jako neefektivní cesta.

Na webové stránce „Natural Earth“ je možné stáhnout tzv. „`shape`“ soubor obsahující polygony všech států na planetě. Tento soubor je možné zpracovat při startu serveru a pak už se jen dotazovat na určení státu. Nástroj `geopackage` obsahuje potřebné funkcionality k vyfiltrování státu podle dotazované polohy.

The screenshot shows the 'Downloads' section of the Natural Earth website. It is organized into three columns, each representing a different scale of data:

- Large scale data, 1:10m:** Offers Cultural, Physical, and Raster themes. It is described as the most detailed, suitable for zoomed-in maps of countries and regions. Scale: 1:10,000,000 (1" = 158 miles, 1 cm = 100 km).
- Medium scale data, 1:50m:** Offers Cultural, Physical, and Raster themes. Suitable for zoomed-out maps of countries and regions. Scale: 1:50,000,000 (1" = 790 miles, 1 cm = 500 km).
- Small scale data, 1:110m:** Offers Cultural and Physical themes. Suitable for schematic maps of the world. Scale: 1:110,000,000 (1" = 1,736 miles, 1 cm = 1,100 km).

Obrázek 12: Webová stránka <https://www.naturalearthdata.com/>

### 3.3 Vykreslování obrázkových map

Základní jednotkou obrázkové mapy je tzv. „Map tile“, jejíž název lze přeložit jako mapová dlaždice. Jedná se o čtvercový bitmapový obrázek. Rozlišení mapových dlaždic se může lišit podle zařízení, na kterém je mapa vykreslována [25]:

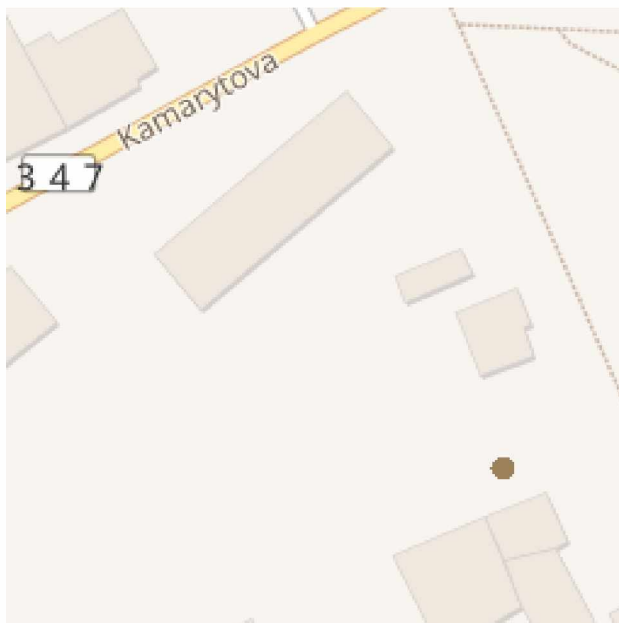
- **256 × 256** pixelů je nepsaným standardem.
- **512 × 512** pixelů je obvyklé pro vykreslování s vysokým rozlišením.
- **64 × 64** pixelů je především určeno pro mobilní zařízení.

Soubor jedné a více mapových dlaždic se nazývá „tileset“. Při vykreslování mapových podkladů není rozumné vykreslovat celý obrázek, ale dotazovat se mapových serverů vždy na určitou část. Tomuto postupu se říká „Slippy map“, kdy výsledná mapa je zobrazena z několika po sobě dotazovaných dlaždic [25].

Každá dlaždice je určena souřadnicí x a y. Ta se liší podle toho, jaká úroveň přiblížení (zoom) je požadována. Jelikož vykreslování mapových podkladů je již řešeno nástrojem Mapnik, bylo v rámci podnikové platformy využito volání již existujícího API serveru pro mapové podklady. Tento nástroj je volně dostupný a je možné ho používat společně s importovanými mapovými podklady do relační PostgreSQL databáze [25].

Server nicméně dokáže vykreslit polohu na dotazovanou mapovou dlaždici. Následující ukázka vykreslí polohu při zaslání adresy:

<http://localhost:8080/map/tile?latitude=49.5432776&longitude=15.3558498&zoom=16>



Obrázek 13: Dotazované umístění na mapové dlaždici [32]

### 3.4 Optimalizace vyhledávání v mapových podkladech

Všechny použité algoritmy v praktické části obsahují jenom minimální optimalizaci. Při nasazení do produkčního prostředí je ale na efektivitu a rychlost dotazu brán zřetel. Následující podkapitoly se budou věnovat způsobům, jak dané algoritmy optimalizovat, aby co možná nejvíce urychlily proces zpracování požadavku od uživatele.

#### 3.4.1 Cashování často dotazovaných částí mapy

Pokud budeme uvažovat, že je systém primárně tvořen pro monitorování firemních vozidel. Dá se předpokládat, že budou často měnit svou polohu. Někteří pracovníci (například poštovní doručovatelé), ale svá vozidla zastavují nebo přesouvají jen po malých částech mapy. Pokud polohovací zařízení vozu neobsahuje optimalizování pro dotazování jen při změně polohy, bude se ptát na stejnou nebo velmi blízkou polohu opakovaně. Stejnou úsporu získáme, i pokud se více vozidel pohybuje po stejné oblasti.

K tomu by stačilo nějakým způsobem mapové podklady zavést do cashovací struktury. V případě jazyka Java připadá v úvahu hashovací mapa. Problém nastává při zvolení vhodného klíče, aby indexace byla okamžitá a správná. Každý mapový podklad je určen severním, jižním, východním a západním bodem, které tvoří oblast stažených dat na mapě.

Tyto oblasti na mapě světa pak představují mřížku. Na základě dotazované zeměpisné šířky a délky lze vždy indexovat danou část pomocí souřadnic x pro sloupce a y pro řádky. Tyto souřadnice pak tvoří klíč mapy, na kterou se lze dotázat, zdali daná oblast již není zpracována, a hned ji vrátit k hledání polohy.

### **3.4.2 Serializace již načtených mapových podkladů**

Zpracování mapových podkladů je po procesu stahování časově nejdělnější činností. Pokud nastane výpadek systému, je výše zmíněná optimalizace neúčinná, protože všechny objekty jsou uloženy ve volatilní operační paměti. Jejich serializací do lokálního média by při znovu nastartování systému mohlo okamžitě objekty zprovoznit bez nutnosti jejich zpracování.

Tato operace je však nerealizovatelná kvůli použití nástroje Osmosin, který u svých tříd neposkytuje možnost serializace. Při pokusu o zápis objektu do binární formy končí operace výjimkou. Řešením by bylo vytvořit si vlastní objekty s touto funkcionalitou a provést převod. Ten by však zabral další výpočetní čas při provedení prvního požadavku.

### **3.4.3 Načtení již stažených mapových podkladů**

Pokud nemůžeme využít serializaci, můžeme alespoň zpracovat již dříve získané mapové podklady. Všechny nové podklady server ukládá do lokálního úložiště, které je zapsáno v nastavení. Informace o zpracovaném mapovém podkladu je také zapsána do databáze. Pokud nastane výpadek nebo plánovaný restart serveru, můžeme z databáze získat umístění těchto souborů a znovu je zpracovat a zaneš do cashovací mapy.



## 4 TVORBA APLIKAČNÍHO ROZHŘANÍ (API)

Aplikační rozhraní serveru přijímá požadavky od klientů na reverzní geokódování, mapové vykreslování a REST operace pro firemní automobily. Hlavní komponentou při tvorbě rozhraní byl framework Spring. Firemní ekosystém, do něhož bude API zasazeno, je již postaven na této technologii.

### 4.1 Spring

Spring je označení pro ekosystém modulů, které umožňují tvořit backendové prostředí naprogramované v jazyce Java. Umožňují programátorovi tvořit REST aplikace, řídit zabezpečení, MVC architekturu a využívat další funkcionality. Jádrem je tzv. Spring Framework, který je aktuálně v páté verzi vývoje. Poskytuje vlastnosti, jako jsou řešení závislostí, webové aplikace, testování (nejčastěji JUnit), vzdálený přístup. Pro fungování je vyžadováno alespoň JDK 8 [19].

Otázky bezpečnosti řeší Spring security. Ten poskytuje již funkční řešení autentizace i autorizace, ochranu proti již známým pravděpodobným útokům a poskytuje podporu známým protokolům, jako jsou OpenID, LDAP, HTTPS [19].

#### 4.1.1 Spring Boot

Nejpoužívanější modul, který je nejrychlejší k vytvoření springové aplikace. Mezi největší výhody patří snadná konfigurace. Spring Boot si řeší většinu závislostí zcela sám a s minimem požadavků nastavení. Většinu konfigurací je nutné bez použití Spring Bootu definovat předem ve složitých souborech XML. Nastavení různých databází je také automaticky řešeno a je jedno, jestli uživatel využívá komerční řešení například od společnosti Oracle nebo jednu z volně dostupných databází. O vše se postará jednoduchá anotace u hlavní třídy [19]:

Ukázka zdrojového kódu 11: Spouštěcí třída Springové aplikace.

```
@SpringBootApplication
@EnableScheduling
public class Main extends SpringBootServletInitializer {
    public static void main(String[] args) {
        SpringApplication.run(Main.class, args);
    }

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder
builder) {
        return builder.sources(Main.class);
    }
}
```

### 4.1.2 REST rozhraní

Klient může využít pro komunikaci se serverovou částí REST rozhraní, které svým návrhem využívá většina moderních aplikací. Rozhraní bylo poprvé představeno v disertační práci doktorem Royem Fieldem. REST jako takové implementuje čtyři základní metody, které jsou známé pod označením CRUD [20].

1. CREATE zastupovaný metodou POST vytváří nový záznam na straně serveru. Ve chvíli jejího volání ještě není znám konečný záznam, protože doposud neexistoval [20].
2. RETRIEVE neboli GET je způsob získání konkrétního zdroje nebo nějaké množiny [20].
3. UPDATE slouží pro aktualizaci nebo změnu konkrétního zdroje a je zastupován metodou PUT [20].
4. DELETE slouží pro vymazání konkrétního zdroje [20].

Rest společně s JSON formátem prakticky definuje základ komunikace klient–server. Moderní frontend frameworky již obsahují alespoň částečnou podporu této architektury [20]. Všechny dostupné entity v serverové části podporují CRUD operace. Následující volání například vypíše všechny zpracované mapové podklady `http://localhost:8080/api/v1/ties_`

Ukázka zdrojového kódu 12: Interface zajišťující CRUD operace pro entitu mapového podkladu.

```
public interface TieRepository extends CrudRepository<Tie, Long> {  
}
```

### 4.1.3 Mapování požadavků

Mapování funkcionalit, jako je reverzní geokódování nebo vykreslování mapových podkladů, jsou řešena v samostatných směrovačích neboli tzv. „Controllerech“ umístěných v balíčku „com.zauzoo.gpsservice.controller“.

Ukázka zdrojového kódu 13: Ukázka směrování metodou GET s vyžadovanými parametry a kódem, který je po přijetí požadavku proveden.

```

@RequestMapping(value = "geolocation", method = RequestMethod.GET)
@ResponseBody
public Address getAddress(@RequestParam("latitude") Double latitude,
    @RequestParam("longitude") Double longitude, @RequestParam("api") String api){
    Address address = geolocationComponent.getAddress(latitude, longitude);
    User user = userRepository.findByApiKey(api);
    addressRepository.save(address);
    historyRepository.save(new History(address, user, latitude, longitude));

    return address;
}

```

## 4.2 Zprovoznění primárního mapového serveru

Největším problémem při získávání dat mapových podkladů je mapový server. Aplikace jako primárního a jediného dodavatele využívá oficiální API od vývojářů OpenStreetMap. Ty jsou, ale dostupné pro každého uživatele a při větším vytížení mapových požadavků se může odezva radikálně zvýšit a ovlivnit tak efektivitu.

Řešením by tedy mohlo být zprovoznit vlastní primární mapový server a oficiální ponechat jako sekundární v případě výpadku nebo většího vytížení. Nástroje a knihovny pro zprovoznění OpenStreetMap serveru jsou volně dostupné k využití. Největší problém je výpočetní náročnost pro mapový server. Mapové podklady přece jenom s každým uživatelským příspěvkem narůstají a tím pádem roste i náročnost výkonu.

Tabulka 7: Nároky na výkon serveru pro mapový podklad celé planety [33]

Komponenta		Doporučeno
Procesor		10jádrový
Operační paměť		32 GB
Úložiště		1 TB

Požadavky se vymykají výkonu běžných serverů. V případě úložiště je navíc počítáno, že server bude využívat dražší SSD disky místo klasických plotnových. Měsíční cena při pronajímání VSD serveru splňujícího výše zmíněné požadavky se pohybuje okolo 27 € za měsíc (při nacenění byl využit ceník poskytovatele služeb Contabo).

Tabulka 8: Nároky na výkon serveru pro mapový podklad Velké Británie [33]

Komponenta	Doporučeno
Procesor	neuvedeno
Operační paměť	12 GB
Úložiště	60 GB

PBD soubor mapových podkladů České republiky zabírá přibližně 718 MB. Uváděná rozmezí velikosti městských částí se pohybuje od 1–50 MB.

### 4.3 Import mapových podkladů do databáze

V následujícím příkladu je zprovozněn OpenStreetMap server s využitím databáze PostgreSQL. Exportovaná mapová oblast byla omezena pouze na město Pardubice, aby se zamezilo vysokým úložným požadavkům. Jako serverové prostředí byl použit linuxový subsystém Ubuntu 18.04 LTS pro Windows 10.

#### 4.3.1 Závislosti

Ke zprovoznění je potřeba nainstalovat následující služby a doplňky:

##### a) PostgreSQL

Objektově relační databázový systém vyvíjený firmou PostgreSQL Global Development Group v jazyce C. První verze byla uvedena 8. července 1996 pod open-source licencí PostgreSQL pro unixové systémy a pod licencí MIT pro systémy Windows. Databáze je často zmiňována právě s OpenStreet mapovými podklady kvůli doplňku PostGIS. Mezi hlavní velké aplikace využívající PostgreSQL patří například Yahoo!, MySpace a Skype [27].

Mezi vlastnosti patří:

- Referenční a datová integrita;
- Podpora funkcí a triggerů;
- Objekty a dědičnost.

Maximální limit velikosti jedné tabulky je 32 TiB.

##### b) PostGIS

Je geografický informační systém pro PostgreSQL databázi, které přidává podporu pro geometrické objekty a umožňuje tím databázi zpracovávat požadavky pro mapové podklady. První verze spatřila světlo světa v roce 2001. Vyvíjen je pod licencí GNU původně společností

Refractions Research. Nové funkcionality jsou přidávány skupinou přispěvatelů, kteří jsou však nadále kontrolováni vedením společnosti. Mezi alternativy toho doplňku patří například Oracle's Spatial extension [26].

Mezi vlastnosti patří:

- Podpora geometrických objektů, jako jsou lomené čáry a polygony [26].
- Prostorové funkce pro výpočet vzdálenosti dvou bodů, výměry a obsahu plochy. Instalace potřebných balíčků [26].

### 4.3.2 Nastavení serveru

Nejdříve musíme aktivovat výše zmíněné doplňky přímo v konzoli PostgreSQL databáze. Předpokládá se, že uživatel má všechna potřebná oprávnění.

Přihlášením do administrátorské konzole PostgreSQL aktivujeme doplňky příkazy:

```
psql -c "CREATE EXTENSION postgis;" -d osm
```

```
psql -c "CREATE EXTENSION hstore;" -d osm
```

Kde „osm“ v části příkazu reprezentuje název databáze, do níž bude proveden import. Následně pak už jen stačí nainstalovat doplněk „osm2pgsql“, který je dostupný v oficiálních repozitářích, a provést import jakéhokoliv osm souboru:

```
osm2pgsql -U admin -W -d osm -H aws-us-west-1-portal.3.dblayer.com -P 17124 --  
hstore --hstore-add-index map.osm
```

Poté se v databázi vytvoří následující schéma tabulek:

- planet\_osm\_line
- planet\_osm\_polygon
- planet\_osm\_roads
- spatial\_ref\_sys
- planet\_osm\_point

Poté lze nad databází provádět dotazy na získání jednotlivých entit, například následující dotaz by se dal využít k získání potenciálních ulic v dané oblasti:

```
SELECT * FROM planet_osm_roads where highway is not null and name is not null;
```

Tabulka 9: Ukázka výstupu dotazu na databázi s importovanými mapovými podklady

Jana Palacha	secondary	NULL
Hlaváčova	primary	NULL
17. listopadu	secondary	NULL
17. listopadu	secondary	asphalt

Hlaváčova	primary	asphalt
-----------	---------	---------

### 4.3.3 Návrh možné optimalizace

Pokud bychom uvážili, že současné API si zpracovává mapové podklady samo, mohl by import do databáze urychlit zpracování dotazů. Přece jenom se jedná o nástroje, které jsou vyvíjeny již delší dobu a můžou obsahovat již nějaké vlastní druhy optimalizací. Nároky na provozování serveru, který by dokázal průběžně zpracovávat data celé planety jsou, ale příliš vysoké a musela by se vzít v potaz nutnost pravidelné aktualizace mapových podkladů.

## ZÁVĚR

Implementace podnikové platformy pro zpracování geolokačních údajů za využití OpenStreetMap se ukázala jako vhodná náhrada za komerční řešení od ostatních společností. Rozsáhlá komunita a snadný přístup k informacím z nich dělá vhodného kandidáta pro využití nejenom v podnikovém systému, ale ve všech druzích aplikací, které potřebují nějakým způsobem zpracovávat uživatelskou polohu. Nicméně i přes fakt, že jsou mapové podklady dostupné zdarma, jejich efektivitu využití v podnikové platformě ukáží až reálné výsledky při nasazení do produkčního prostředí.

Během tvorby aplikačního rozhraní bylo nalezeno několik obtíží. I když podniková platforma zpracovává dotazy na reverzní geokódování zcela sama, musí nějakým způsobem získat mapová data. Při vykreslování mapových podkladů je sama o sobě odkázána na API třetích stran. Při reverzním geokódování musí na některém ze serverů stáhnout požadované soubory s mapovými informacemi a ty zpracovat. Mělo by tedy smysl ještě vedle založit vlastní mapové servery, se kterými by služba komunikovala a vyhnula by se tak bodu selhání v případě nedostupnosti těchto služeb. Je zde ale otázka, zdali se tento způsob cenově vyplatí.

Mezi další rozšíření, která by bakalářská práce mohla nabídnout, je i frontedová část aplikace, do které by klient mohl vstupovat a zadávat přes ni dotazy na API rozhraní serveru. Užitečné by bylo trasování cesty firemních automobilů podle náročnosti terénu, aby byla co nejrychlejší nebo co nejúspornější, a následná vizualizace trasy. Prostor by také dostaly další možnosti optimalizace, mezi které by patřilo předzpracování mapových podkladů na základě uživatelsky předem definované trasy. Zavedení aktualizace mapových podkladů po vypršení jejich minimální platnosti a spolehlivosti.



## POUŽITÁ LITERATURA

- [1] TECHOPEDIA. Geolocation. *Techopedia.com* [online]. Edmonton (Canada): Janalta Interactive, 2012 [cit. 2020-08-02]. Dostupné z: <https://www.techopedia.com/definition/1935/geolocation>
- [2] WATTS, Mark, James BRUNGER a Kate SHIRES. Do European data protection laws apply to the collection of WiFi network data for use in geolocation look-up services? *International Data Privacy Law* [online]. 2011, 1(3), 149–160 [cit. 2020-08-02]. DOI: 10.1093/idpl/ipr013. ISSN 2044-3994.
- [3] AIMEE, Laurence. The evolution of Google Maps. In: *Geospatialworld.net* [online]. Uttar Pradesh (India): Geospatial Media & Communications, 23. 1. 2020 [cit. 2020-08-02]. Dostupné z: <https://www.geospatialworld.net/blogs/the-evolution-of-google-maps/>
- [4] Google. Google Maps Platform. *Cloud.google.com* [online]. Silicon Valley (San Francisco): Google, 2020 [cit. 2020-08-02]. Dostupné z: <https://cloud.google.com/maps-platform/>
- [5] *Mapbox* [online]. San Francisco: Mapbox, 2020 [cit. 2020-08-02]. Dostupné z: <https://www.mapbox.com/>
- [6] OpenStreetMap. Why OpenStreetMap? In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://wiki.openstreetmap.org/wiki/Why\\_OpenStreetMap%3F](https://wiki.openstreetmap.org/wiki/Why_OpenStreetMap%3F)
- [7] OpenStreetMap. History of OpenStreetMap. In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://wiki.openstreetmap.org/wiki/History\\_of\\_OpenStreetMap](https://wiki.openstreetmap.org/wiki/History_of_OpenStreetMap)
- [8] OpenStreetMap. Copyright and License. *Openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: <https://www.openstreetmap.org/copyright>
- [9] OpenStreetMap. Elements. In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: <https://wiki.openstreetmap.org/wiki/Elements>
- [10] OpenStreetMap. Cs: Česko/Značení silnic. In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://wiki.openstreetmap.org/wiki/Cs:%C4%8Cesko/Zna%C4%8Den%C3%AD\\_silnic](https://wiki.openstreetmap.org/wiki/Cs:%C4%8Cesko/Zna%C4%8Den%C3%AD_silnic)
- [11] OpenStreetMap. Cs: Česko/Systém adres. In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://wiki.openstreetmap.org/wiki/Cs:%C4%8Cesko/Syst%C3%A9m\\_adres](https://wiki.openstreetmap.org/wiki/Cs:%C4%8Cesko/Syst%C3%A9m_adres)
- [12] OpenStreetMap. Addresses. In: *Wiki.openstreetmap.org* [online]. OpenStreetMap Wiki: OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: <https://wiki.openstreetmap.org/wiki/Addresses>

- [13] OpenStreetMap. OSM file formats. In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://wiki.openstreetmap.org/wiki/OSM\\_file\\_formats](https://wiki.openstreetmap.org/wiki/OSM_file_formats)
- [14] OpenStreetMap. OSM XML.. In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://wiki.openstreetmap.org/wiki/OSM\\_XML](https://wiki.openstreetmap.org/wiki/OSM_XML)
- [15] OpenStreetMap. PBF Format. Wiki. In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://wiki.openstreetmap.org/wiki/PBF\\_Format](https://wiki.openstreetmap.org/wiki/PBF_Format)
- [16] IntelliJ IDEA. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://en.wikipedia.org/wiki/IntelliJ\\_IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA)
- [17] JetBrains. Choose Your Edition. *Jetbrains.com* [online]. Prague: JetBrains, 2020 [cit. 2020-08-02]. Dostupné z: [https://www.jetbrains.com/idea/features/editions\\_comparison\\_matrix.html](https://www.jetbrains.com/idea/features/editions_comparison_matrix.html)
- [18] OpenStreetMap. Osmosis. In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: <https://wiki.openstreetmap.org/wiki/Osmosis>
- [19] CARYN, Matthias. Spring IO 2017: The Spring Ecosystem. In: *Ordina.jworks.github.io* [online]. Mechelen (Belgium): Ordina JWorks, 7. 6. 2017 [cit. 2020-08-02]. Dostupné z: <https://ordina-jworks.github.io/spring/2017/06/07/Spring-IO-2017-The-Spring-ecosystem.html>
- [20] MALÝ, Martin. REST: architektura pro webové API. In: *Zdrojak.cz* [online]. Praha (Česká republika): Devel.cz Lab, 3. 8. 2009 [cit. 2020-08-05]. Dostupné z: <https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>
- [21] HEROUT, Pavel. *Java: rozšířené vydání zahrnuje změny od Java 5*. 5. rozš. vyd. České Budějovice: Kopp, 2011. ISBN 978-80-7232-398-2.
- [22] Seznam.cz. Mapy.cz a COVID-19 / Sdílím polohu. *Napoveda.seznam.cz* [online]. Praha: Seznam.cz, 2020 [cit. 2020-08-06]. Dostupné z: <https://napoveda.seznam.cz/cz/mapy/covid-19/>
- [23] Pokemon Go. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2020 [cit. 2020-08-06]. Dostupné z: [https://cs.wikipedia.org/wiki/Pok%C3%A9mon\\_Go](https://cs.wikipedia.org/wiki/Pok%C3%A9mon_Go)
- [24] Federal Aviation Administration. Satellite Navigation – GPS – How It Works. *Faa.gov* [online]. Washington, DC: U. S. Department of Transportation, 2020 [cit. 2020-08-06]. Dostupné z: [https://www.faa.gov/about/office\\_org/headquarters\\_offices/ato/service\\_units/techops/navservices/gnss/gps/howitworks/](https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/howitworks/)
- [25] OpenStreetMap. Slippy map tilenames. In: *Wiki.openstreetmap.org* [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://wiki.openstreetmap.org/wiki/Slippy\\_map\\_tilenames](https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames)

- [26] PostGIS. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2020 [cit. 2020-08-07]. Dostupné z: <https://cs.wikipedia.org/wiki/PostGIS>
- [27] PostgreSQL: The world's most advanced open source database. About. *Postgresql.org* [online]. Dresden (Germany): The PostgreSQL Global Development Group, 2020 [cit. 2020-08-07]. Dostupné z: <https://www.postgresql.org/about/>
- [28] Pricing that scales to fit your needs. Google Cloud [online]. San Francisco (CA): Google, 2020 [cit. 2020-08-13]. Dostupné z: <https://cloud.google.com/maps-platform/pricing>
- [29] Mapbox pricing. Mapbox [online]. San Francisco (CA): Mapbox, 2020 [cit. 2020-08-13]. Dostupné z: <https://www.mapbox.com/pricing/>
- [30] Jak určit obecnou rovnici přímky určené dvěma body. E-Matematika.cz [online]. Praha: Jazyková škola Březinka, 2020 [cit. 2020-08-14]. Dostupné z: <https://www.e-matematika.cz/stredni-skoly/jak-urcit-obecnou-rovnici-primky-urcene-dvema-body.php>
- [31] Haversine formula to find distance between two points on a sphere. GeeksForGeeks [online]. Noida (India): GeeksForGeeks, 2020 [cit. 2020-08-14]. Dostupné z: <https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/>
- [32] Autorská práva a licence. In: OpenStreetMap [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: <https://www.openstreetmap.org/copyright>
- [33] GUOAN, Xiao. How to Set Up OpenStreetMap Tile Server on Ubuntu 18.04. LinuxBabe [online]. Chicago: Xiao Guoan, 2019 [cit. 2020-08-14]. Dostupné z: <https://www.linuxbabe.com/ubuntu/openstreetmap-tile-server-ubuntu-18-04-osm>
- [34] Key:addr. OpenStreetMap Wiki [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: <https://wiki.openstreetmap.org/wiki/Key:addr>
- [35] Cs:Česko/Značení silnic. OpenStreetMap Wiki [online]. Cambridge (United Kingdom): OpenStreetMap Foundation, 2020 [cit. 2020-08-02]. Dostupné z: [https://wiki.openstreetmap.org/wiki/Cs:%C4%8Cesko/Zna%C4%8Den%C3%AD\\_silnic](https://wiki.openstreetmap.org/wiki/Cs:%C4%8Cesko/Zna%C4%8Den%C3%AD_silnic)

## **PŘÍLOHY**