

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## KONTROLNÍ APLIKACE PRO TRASY KČT V OSM

BAKALÁŘSKÁ PRÁCE

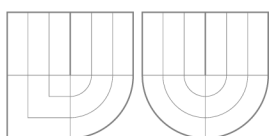
BACHELOR'S THESIS

AUTOR PRÁCE

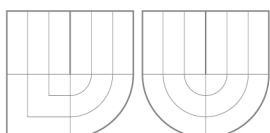
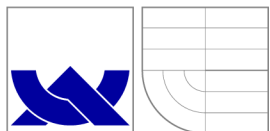
AUTHOR

PETR ŠVAŇA

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ



FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# KONTROLNÍ APLIKACE PRO TRASY KČT V OSM

MANAGEMENT APPLICATION FOR TOURISTIC ROUTES IN OSM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR ŠVAŇA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ KAŠPÁREK

BRNO 2015

## Abstrakt

Tato práce se zabývá kontrolou turistických tras Klubu českých turistů v datech internetové mapy OpenStreetMap. Ve výsledné aplikaci má uživatel k dispozici vizualizovaná data z OpenStreetMap a implementované mechanismy pro usnadnění kontroly těchto dat. V tomto textu je popsán úvod do problematiky a informace o datovém modelu OpenStreetMap spolu informacemi o mapování turistických tras. Dále následuje analýza současných možností kontrol a návrh takové aplikace, která dokáže tyto kontroly zavést do praxe. V dalších dvou kapitolách lze nalézt popis vlastní implementace této aplikace a podrobnější manuál.

## Abstract

Bachelor thesis deals with analysis of Czech Tourist Club hiking routes and checking their data validity in OpenStreetMap internet map source. In the resulting application has user at his disposal visualized OpenStreetMap data and implemented gadgets to simplify the control of these data. First, introduction to this topic is discussed including information about both data model, that OpenStreetMaps uses, and mapping of the tourist trails. Second, analysis of current ways for map data checking along with development of the application, that can bring these controls into practice are concerned. In following chapters, created implementation of the application with detailed manual can be found.

## Klíčová slova

OpenStreetMap, KČT, kontrola turistických tras, Leaflet, PostgreSQL databáze, aktualizace OSM databáze, vizualizace turistických tras

## Keywords

OpenStreetMap, KČT, tourist routes control, Leaflet, PostgreSQL database, update of OSM database, tourist routes visualization

## Citace

Petr Švaňa: Kontrolní aplikace pro trasy KČT v OSM, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Kontrolní aplikace pro trasy KČT v OSM

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Tomáše Kašpárka.

.....  
Petr Švaňa  
20. května 2015

## Poděkování

Zde bych rád poděkoval svému vedoucímu, panu Ing. Tomáši Kašpárkovi, za cenné rady a odbornou pomoc při řešení této bakalářské práce.

© Petr Švaňa, 2015.

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

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Projekt OpenStreetMap</b>	<b>4</b>
2.1 Úvod do OpenStreetMap	4
2.2 Elementární datová primitiva v OSM	4
2.3 OSM data	5
2.4 Mapování turistických tras v OpenStreetMap	7
2.5 Mapování cyklistických tras v OpenStreetMap	10
<b>3 Současné možnosti kontroly turistických tras v datech OSM</b>	<b>11</b>
3.1 Ruční možnosti kontroly	11
3.2 Automatické možnosti kontroly	12
3.3 Poloautomatické možnosti kontroly	13
<b>4 Návrh aplikace pro kontrolu turistických tras</b>	<b>14</b>
4.1 Struktura aplikace	14
4.2 Nástroje pro práci s daty	14
4.3 Databáze	16
4.4 Zobrazení dat	16
4.5 Mapový podklad	17
4.6 Vytváření překryvných vrstev	17
4.7 Návrh uživatelského rozhraní	17
<b>5 Implementace aplikace</b>	<b>19</b>
5.1 Adresářová struktura	19
5.2 Databázová část	19
5.3 Mapová část	22
5.4 Uživatelské rozhraní	24
5.5 Tabulková část	26
<b>6 Manuál k výsledné aplikaci</b>	<b>27</b>
6.1 Obecný popis mapové části	27
6.2 Volba zobrazení vrstev a podkladu	27
6.3 Interakce se zobrazenými prvky	28
6.4 Vkládání uživatelských vstupů	28
6.5 Obecný popis tabulkové části	30
<b>7 Závěr</b>	<b>31</b>

<b>A Ukázky práce s aplikací</b>	<b>34</b>
<b>B Obsah CD</b>	<b>37</b>

# Kapitola 1

## Úvod

Mezi klasické zástupce celosvětových internetových map, kde jsou zavedení zástupci z řad Googlu nebo Microsoftu, se již několik let řadí také OpenStreetMap. Tento projekt má trochu odlišný princip než jeho konkurenti a o mapování se zde starají především sami uživatelé. Výhody a nevýhody tohoto přístupu lze pozorovat například na populární internetové encyklopedii Wikipedia: obsah a rozsah encyklopedie je opravdu velký, ale aby tento princip mohl fungovat, je třeba výsledky uživatelské práce kontrolovat. U digitálních map tomu není jinak. Ty získávají v běžném životě stále větší význam a dostávají se tak na místa, která byla doposud doménou klasických papírových map. S rozvojem mobilních dat a jiných mobilních technologií je možné tyto mapy aktualizovat doslova za pochodu.

Kontrola turistických tras v OpenStreetMap datech je ústředním motivem této bakalářské práce. Pro mapování turistických tras v České republice existují pravidla, jimiž by se měli uživatelé řídit. S rostoucím počtem uživatelů se však logicky stává, že roste i celkový počet chyb, jenž při mapování vznikne. Takové chyby mohou být různého druhu, od tras, které vedou ve skutečnosti jinak, až po chybějící hodnotu některého tagu. Chyby mohou také vznikat i při samotném značení, případně se může něco v terénu změnit. Tato bakalářská práce má za úkol přinést mechanismy pro kontrolu těchto nesrovnalostí, a zároveň také nabízet dostatečné nástroje pro ruční kontrolu turistických tras a automaticky upozornit na chyby, které lze zjistit již přímo z dat.

Výsledný produkt popisují v následujících kapitolách. Ve druhé kapitole je tématem OpenStreetMap projekt, a to především jeho datová část, popis souborových typů a elementárních datových primitiv. Je zde také popsáno, jak probíhá mapování turistické trasy z pohledu dat, které se spolu s ní ukládají, a která jsou jádrem automatických kontrol. V kapitole č. 3 jsou rozebrány současné možnosti kontrol turistických tras. Z těchto možností se vychází při návrhu výsledné aplikace, který je popsán na stránkách čtvrté kapitoly. Je zde popsán jak samotný návrh, tak také nástroje, s nimiž pro dosažení cíle. Jeho implementace je podrobně popsána v kapitole č. 5, která je rozdělena postupně na implementaci databázové, mapové a tabulkové části. Poslední kapitolou před závěrem je manuál pro výslednou aplikaci, který popisuje implementované prvky z pohledu uživatele.

## Kapitola 2

# Projekt OpenStreetMap

Tato kapitola obsahuje obecný popis projektu OpenStreetMap, zaměřím se hlavně na práci s daty, o která se bude tato aplikace především opírat. Část kapitoly je poté věnována krátkému popisu mapování, zejména turistických tras.

### 2.1 Úvod do OpenStreetMap

Projekt OpenStreetMap (OSM)[1] si klade za cíl vytvořit editovatelnou mapu celého světa. Jedná se o svobodná data, která jsou poskytována za podmínek Open Data Commons Open Database License<sup>1</sup>. Většina dat je dodávána samotnými uživateli, ať už z GPS přijímačů nebo jiných zdrojů. Díky široké členské základně<sup>2</sup> a volnému šíření dat se tento projekt rozšířil také na jiné platformy, mimo počítače také na různá mobilní zařízení. Protože OpenStreetMap používá tento Wikipedii podobný koncept, kdy o tom, co v mapě je, rozhodují ve velké míře dobrovolníci, je zde dosaženo nerovnoměrného rozvržení informací na mapě podle toho, jak je daná oblast obydlená (navštěvovaná) uživateli OSM.

### 2.2 Elementární datová primitiva v OSM

OpenStreetMap využívá pro mapování speciální datový model[2], který slouží k popisu reálného světa. Jeho základní prvky jsou:

- Uzel (**node**): značí specifické místo na zemském povrchu. Minimální údaje o uzlu sestávají z jeho id čísla, zeměpisné šířky a zeměpisné délky. Lze jej použít pro popis polohy bodového prvku (např. turistický rozcestník) nebo jako součást cest.
- Cesta (**way**): popisuje objekty, které lze vyjádřit linkou (silnice, řeky). Také ji lze využít pro popis plochy (budova, parkoviště), kdy cesta začíná i končí ve stejném uzlu. Cesta jako datový prvek se skládá z uzlů.
- Relace (**relation**): jedná se o seskupení uzlů a cest do jednoho logického celku.

Mimo tyto základní prvky ještě existuje další prvek, který se jmenuje značka. Značka (**tag**) popisuje nějakou vlastnost výše zmíněných elementárních datových prvků. Značka se skládá z klíče (**key**) a hodnoty (**value**). Teoreticky může mít klíč i hodnota libovolný obsah, většinou však již existují pro daný účel dohodnuté konvence určené uživateli.

<sup>1</sup><http://opendatacommons.org/licenses/odbl/>

<sup>2</sup>přes 1,8 miliónů v říjnu 2014



### 2.2.1 Relace

Relace[3] je jeden ze základních datových prvků OSM. Používají se pro modelování logických anebo geografických vztahů mezi objekty. Jsou určeny pro seskupování objektů hlavně na místní úrovni. Skládá se z jedné či více značek a z uspořádaného seznamu, obsahujícího minimálně jeden (většinou více) uzel anebo cestu. Uzly a cesty relace se nazývají členy. Členy relace mohou mít svou roli určenou volitelným textovým polem. Role určuje, jaký má daný člen relace význam. Pro stabilní relaci je doporučeno, aby měla maximálně 300 prvků.

Zde je výčet a popis některých nejpoužívanějších relací<sup>3</sup>.

- Cesta (**route**): popisuje všechny možné druhy cest, jako jsou např. silnice, železnice, tramvajová linka, turistická trasa, cyklostezka.
- Multipolygon: používá se ke kreslení složitějších ploch, např. mapování lesa bez jeho travnatých ploch.
- Směrové tabule (**destination signs**): poskytuje informace o směrových tabulích na křižovatkách.
- Vodní cesta (**waterway**): slouží k mapování řek, cílem je mít pro každou řeku relaci tohoto typu.
- Ulice (**street**): spojuje dohromady vše, co patří k jedné ulici (cesty, domy/adresy,...).

## 2.3 OSM data

Důsledkem početné komunity přispěvatelů OSM, kdy mapu edituje najednou několik tisícovek uživatelů, je velká datová náročnost. Vzhledem k tomu, že je OSM otevřený projekt, poskytuje svá data k volnému využití komukoliv. To vede k potřebě ukládat data tak, aby šla relativně snadno přečíst. Nejčastěji se můžeme setkat s formátem OSM XML (2.3.1), popřípadě s vysoce komprimovaným binárním formátem PBF. Dále ještě existují speciální formáty pro soubory obsahující změny, z nich si blíže přiblížíme formát OsmChange (2.3.2).

### 2.3.1 OSM XML

Pro základní OSM data jsou používány `.osm` soubory. Soubory mají strukturu ve formátu XML[4]. Mezi výhody takového ukládání patří již zmíněná dobrá čitelnost souboru, nezávislost na platformě, možnost použití existujících XML parserů a dobrý kompresní poměr. Nevýhodou je, že se většinou pracuje s velkými soubory. To s sebou může nést další problémy a časovou náročnost jak při kompresi a dekompresi, tak i při samotném zpracování `.osm` souborů.

OSM soubor má danou strukturu následovně:

- Na začátku souboru je XML deklarace obsahující informaci o verzi XML a použitém kódování.
- Následuje OSM element, který v sobě nese informace o verzi OSM API a o použitém generátoru, který tento soubor vytvořil.

---

<sup>3</sup>více typů lze nalézt zde: [http://wiki.openstreetmap.org/wiki/Types\\_of\\_relation](http://wiki.openstreetmap.org/wiki/Types_of_relation)

- Blok uzlů. Jeden uzel je reprezentován elementem **node**. Uvnitř elementu jsou obsaženy značky (**tag**) patřící ke konkrétnímu uzlu.
- Blok cest. Cesta je reprezentovaná elementem **way**. Element obsahuje odkazy na své uzly a značky vztahující se ke konkrétní cestě.
- Blok relací. Relace je reprezentovaná elementem **relation** a obsahuje reference na své členy a své značky.

Použití této struktury zaručuje to, že bloky datových primitiv budou ve specifikovaném pořadí. Nezaručuje už však, že budou obsaženy všechny bloky, nebo že budou prvky bloků podle něčeho seřazeny.

Listing 2.1: Ukázka OSM XML souboru

```
<?xml version='1.0' encoding='UTF-8'?>
<osm version="0.6" generator="Osmosis 0.39">
  <node id="172516" version="4" lat="50.0811737" lon="14.4045229">
    <tag k="highway" v="traffic_signals"/>
  </node>
  <way id="36931649" version="3">
    <nd ref="74048532"/>
    ...
    <tag k="source" v="cuzk:km;uhul:ortofoto"/>
  </way>
  <relation id="1604518">
    <member type="way" ref="115385636" role=""/>
    <member type="node" ref="1304234173" role="guidepost"/>
    ...
    <tag k="route" v="foot"/>
    <tag k="type" v="route"/>
  </relation>
</osm>
```

Soubor `planet.osm`, který pokrývá celou Zemi, má v současné době přes 500GB. Soubory s touto strukturou se proto obvykle distribuují v komprimovaném formátu, a to pomocí `bzip2` nebo `gzip`. Výhodou formátu `bzip2` je větší efektivita a tedy menší výsledné soubory, formát `gzip` potom zaručuje oproti formátu `bzip2` rychlejší zpracování. Pro porovnání, komprimovaný `planet.osm` pomocí `bzip2` má velikost kolem 38 GB. Ekvivalentní soubor PBF má potom velikost okolo 26 GB.

### 2.3.2 OsmChange formát

OsmChange[5] je speciální formát souboru, který používá mj. program Osmosis (4.2.3). Jeho hlavním účelem je to, že obsahuje popis rozdílů mezi dvěma instancemi OSM dat, typicky se jedná o hlavní server a pravidelně aktualizovaný soukromý server. Hlavní OpenStreetMap server právě skrze tento formát poskytuje tzv. `changesety`. OsmChange soubor tedy obsahuje data se změnami, které se v mapových datech OSM uskutečnily za specifikovanou dobu (např. den, týden). Používá se především pro synchronizaci a analýzu těchto změn.

Formát OsmChange má podobnou strukturu jako jiný OSM soubor (2.3.1). Na nejvyšší úrovni je `osmChange element`, který obsahuje informace o verzi `osmChange` a o použitém generátoru, který tento soubor vytvořil. Dále se soubor typicky dělí na tyto sekce:

- create - zde se nachází nově vytvořené prvky
- modify - zde se nachází prvky, jenž byly změněny oproti původnímu stavu
- delete - prvky, které v nové verzi nejsou obsaženy

Každá sekce obsahuje dále popis prvků strukturovaný stejně jako v jiném OSM souboru, takže obsahuje blok uzlů, blok cest a blok relací.

## 2.4 Mapování turistických tras v OpenStreetMap

V této práci není příliš podstatný samotný obecný proces mapování<sup>4</sup>. Důležité však je, jakým způsobem jsou uložena data o turistických trasách, a které informace lze z těchto dat získat.

Obecně je turistická trasa sama o sobě relace, která má za členy (member) jednotlivé úseky trasy ve formě cest (way) a rozcestníky ve formě uzlů (node). Je tedy potřeba využít všechna tři základní datová primitiva OSM. Dále mohou být u každé této části uvedeny doplňující značky (tag), které jsou nositeli bližších informací relevantních k dané části (rozcestí, úsek, celá trasa). Tento model je jednotný pro turistické trasy ve světě v OSM. Dále se jednotlivé země a oblasti od sebe více či méně odlišují.

Pro Českou republiku existuje snaha sjednotit některé aspekty mapování v OSM[6]. Tato snaha se týká také turistických tras. Tyto jsou ještě dále rozvinuty ve značkovém klíči[7] projektu OpenTrackMap<sup>5</sup> (t.č. neaktivní), který využívá OpenStreetMap pro turistiku v České Republice. Tento značkový klíč je také využíván projektem MTBmap<sup>6</sup>. Dále popisují informace, které lze z tohoto značkového klíče získat, ve formě popisu konkrétních značek.

### 2.4.1 značka: type

Tato značka určuje typ relace. Hodnoty těchto klíčů odpovídají typům relací (2.2.1), typickými hodnotami jsou např. `multipolygon`, `route` nebo `waterway`. Aby mohla být relace vůbec prohlášena za turistickou, nutnou podmínkou je použití hodnoty `route`, jenž se obecně používá pro jakékoliv trasy.

### 2.4.2 značka: route

Značka `route` se u relací používá pro typ trasy dle druhu doporučeného transportu. Z hodnoty této značky by mělo být patrné, pro koho je trasa určena. Pro turistické relace se užívají především tyto hodnoty:

- `foot` - souhrnné označení pro trasy určené pro pěší chůzi. Dle českého OpenTrackMap značkového klíče by se měla pro pěší trasy používat tato značka.
- `hiking` - oficiální označení pro trasy, které mají turistický charakter (dlouhá, na venkově). Z praktických důvodů se nabízí řešení toto nerozlišovat a uvažovat o hodnotách `foot` a `hiking` jako o ekvivalentních.

<sup>4</sup>jeden z mnoha návodů pro mapování: <http://www.gisportal.cz/2013/06/uvod-do-mapovani-s-openstreetmap/>

<sup>5</sup><http://opentrackmap.cz/>

<sup>6</sup><http://mtbmap.cz/>

- **ski** - označení lyžařských tras
- **bicycle** - označení tras pro cyklisty
- **wheelchair** - označení speciálních tras pro vozíčkáře
- **horse** - označení stezek pro jezdce na koni

### 2.4.3 značka: network

Tag **network** má obecně určovat, do které sítě se má daná relace zařadit[8]. Používá se např. pro sítě silnic, či pro linky veřejné hromadné dopravy. U turistických tras určuje, zda se jedná o trasu dálkovou (national), regionální nebo lokální. Pro pěší trasy se používá hodnot **nwn** (national walking trail network), **rwn** (regional) a **lwn** (local). Výjimkou jsou mezinárodní trasy (např. E3), pro které je určena síť **iwn** (international). Pro cykloturistické trasy existují obdobné hodnoty **ncn**, **rcn** a **lcn**. Dle zvyklostí KČT se turistickým trasám přiděluje barva podle významnosti trasy[9], kde červenou trasu nosí dálkové a páteřní trasy a např. žlutou barvu má zkratka či spojovací trasa. Podle tohoto modelu by se dal automaticky zjistit charakter trasy a podle toho kontrolovat hodnotu tohoto tagu, nicméně na tuto kontrolu se nelze příliš spoléhat.

### 2.4.4 značka: operator

Tento tag se používá pro uvedení jména sdružení, osoby nebo jiné entity, která je za trasu zodpovědná. O české turistické trasy se stará Klub Českých Turistů (KČT), hodnota v Česku používaného tagu je potom **cz:KČT**.

### 2.4.5 značka: kct\_barva

Zde je klíč částečně proměnný, místo přípony **barva** se na jejím místě uvádí barva, kterou je trasa značena. Tento tag používá zavedené hodnoty tagu **route**, jenž se liší jen u pěších tras, které se zde dále dělí následovně:

- **major** - klasické pásové značení
- **local** - místní značení
- **learning** - naučná stezka
- **peak** - odbočka k vrcholu nebo vyhlídce
- **ruin** - odbočka ke zřícenině
- **spring** - odbočka ke studánce nebo k pramenu
- **interesting\_object** - odbočka k jinému zajímavému objektu

### 2.4.6 značka: ref

Tag **ref** má smysl jako položka pro umístění evidenčního čísla relace, pokud tato nějaké má (např. číslo silnice). Dle značkového klíče OTM by se měl používat pro cykloturistické trasy. Pro ostatní typy tras je tato značka volitelná, pravděpodobně z toho důvodu, že KČT v poslední době upouští od číslování tras[6].



Obrázek 2.1: Ukázky současných KČT značek pro pěší turistiku (ve stejném pořadí jako výčet výše, [10]).

#### 2.4.7 značka: `osmc:symbol`

Tato značka byla vynalezena pro usnadnění vykreslování symbolů a tras pro turistické trasy. Výhoda této značky je, že sama o sobě poskytuje kontrolu ostatním zadaným tagům, což je vlastnost, která je v této práci velmi přínosná (viz 3.2).

#### 2.4.8 značka: `name`

Tento klíč zpravidla obsahuje hodnotu, která je oficiálním názvem daného prvku. Stejně tak pro České turistické relace by se měl využívat pouze pro oficiální název (např. *cesta T. G. Masaryka*) a ne pro obecné názvy (např. *žlutá kolem potoka*).

#### 2.4.9 značka: `destinations`

V OTM značkovém klíči má tento určovat destinace dané značky, tzn. významná místa po cestě. Hodnotou je seznam významných cílů oddělených středníkem. Zde se značka ukládá přímo do příslušné relace. Nezaměňovat s tagem `destination`, který se využívá při mapování silnic.

#### 2.4.10 značka: `complete`

Zda je relace kompletně zmapovaná nebo nikoliv, to je uloženo v tagu `complete` jako jedna z hodnot `yes/no`.

#### 2.4.11 značka: `abandoned`

Podobně funguje ukazatel toho, zda je relace již zrušená (opuštěná). Velmi zřídka používaný tag u turistických relací, uvádím jej, protože je také součástí OTM značkového klíče.

#### 2.4.12 značka: `source`

Značka `source` slouží pro označení zdroje použitého při mapování. Častou hodnotou je `survey`, tedy průzkum přímo na místě.

#### 2.4.13 značka: `wikipedia`

Tag slouží pro uložení odkazu na wikipedii, který se přímo vztahuje k mapovanému prvku.

## 2.5 Mapování cyklistických tras v OpenStreetMap

Mapování cyklotras v ČR se řídí v základu obecnými mezinárodními pravidly[11] doplněnými o tag `operator`[12], který by měl mít tak jako (cyklo)turistické trasy hodnotu `cz:KČT`, případně jiného lokálního garanta cykloznačení.

## Kapitola 3

# Současné možnosti kontroly turistických tras v datech OSM

Kontrolovat turistické trasy v OSM datech lze vícero způsoby. Tyto způsoby můžeme rozdělit na ruční a automatické podle toho, které informace ke kontrole používáme a také to, kolik relací najednou takto zkontrolujeme. V této kapitole si popíšeme ty nejvíce relevantní způsoby.

### 3.1 Ruční možnosti kontroly

Zde patří takové metody, které nelze vždy vyhodnotit strojově, nebo je to příliš nákladné na zdroje. Typickým znakem ručních kontrol je tedy menší rychlost zpracování. Problémem může být možná menší míra konzistence, kdy se data nekontrolují rovnoměrně v čase a prostoru. Výhodou je pak vysoká úroveň detailu takové kontroly.

#### 3.1.1 Porovnání s jinou mapou

Tato kontrola může využívat jiné mapové zdroje, ať už klasické papírové mapy nebo jiné internetové zdroje (např. turistická mapa na serveru Mapy.cz). Zde se nabízí pro porovnání viditelné aspekty turistické relace, tzn. délka, vedení trasy, rozmístění rozcestníků, barva atp. Vzhledem k tomu, že se většinou jedná o projekty s komerční licencí, není tento postup doporučen pro doslovný přepis či překreslení, ale spíše pro určitou orientaci v kontrolované lokaci.

#### 3.1.2 Porovnání přímo z terénu

Porovnání na základě zkušenosti s příslušným místem, kde si mapující buď přímo pamatuje konkrétní aspekty místa, nebo použije pro porovnání zobrazené trasy vykreslení stejné trasy např. z GPS trackeru. Tento způsob je výhodnější než výše uvedený v tom, že mapující uživatel se na místě skutečně pohyboval. Ovšem záleží také na přesnosti GPS přístroje nebo na času uplynulém od takového průzkumu.

#### 3.1.3 Kontrola s využitím jiných projektů nebo externích zdrojů

Pro kontrolu lze využít také jiná data, např. data z jiných projektů podobného, ale i odlišného zaměření. Kupříkladu import některých dat z projektu, jež se specializuje na shro-

mažďování fotek rozcestníků, může pomáhat s orientací v místech, kde pouze OSM data nestačí.

## 3.2 Automatické možnosti kontroly

Pod automatické možnosti kontroly turistických tras lze zařadit ty metody, kterými lze hodnotit více tras najednou. Lze tímto např. jednoduše odlišit relace turistické od relací, které pravděpodobně turistické nejsou. Dalším příkladem využití je kontrola chybně vyplněných značek u relací, jež už za turistické považujeme. Jádrem automatického způsobu jsou kontroly různých tagů, které by měly být součástí relací turistických tras, případně součástí uzlu, jež je členem turistické relace.

### 3.2.1 Nevyplněné tagy

Kontrola založená na tom, zda jsou vyplněny všechny tagy, které mají být vyplněny podle OTM značkového klíče. Pravděpodobně se jedná o nejjednodušší strojovou kontrolu, kde se pouze zkontroluje přítomnost daného prvku v databázi.

Při této kontrole může vznikat problém u některých cyklistických tras. Z hlediska mapování můžeme rozlišovat dva druhy:

- **cykloturistické trasy** - mapují se pomocí OTM klíče (2.4), mají svou turistickou značku a barvu
- **obyčejné cyklistické trasy** - mapují se podle jiných pravidel (2.5), jsou charakteristické žlutými cedulemi u silnic, specifickou barvu nemají

Problém zde může nastat proto, že klíč pro obyčejné cyklistické trasy obsahuje stejné hodnoty, jako OTM klíč pro cykloturistické trasy, ten však obsahuje hodnot více. Není tedy vždy zřejmé, jestli se jedná o obyčejnou cyklistickou trasu, nebo o cykloturistickou trasu s chybějícími tagy. Zde se nabízí možnost kontrolovat tyto nejasné relace jen na explicitní povolení kontrol tras s parametrem `route=bicycle` a chybějícím tagem `kct_barva`.

### 3.2.2 Hodnoty tagů ve špatném formátu

U některých značek je požadována také určitá forma hodnoty, která by měla splňovat určitá pravidla. Příkladem může být např. hodnota tagu `destinations`, kde by mezi jednotlivými cíli měl sloužit jako oddělovač znak středníku, nikoliv např. pomlčka. Dále např. u značky `osmc:symbol` může nastat obdobný problém.

### 3.2.3 Rozpory ve vyplněných údajích

Tato možnost kontroluje hodnoty tagů, které jsou navzájem v rozporu. Zde je přehled těch nejčastějších rozporů vyskytujících se v turistických relacích:

- **barva: `kct_barva` vs. `osmc:symbol`** - Srovnání značky, která ve svém názvu obsahuje základní barvu relace a tagu `osmc:symbol`, který mj. také obsahuje barvu. Zde se barvy mohou navzájem lišit a vzniká rozpor.
- **typ trasy: `kct_barva` vs. `route` vs. `osmc:symbol`** - Jak je patrné z kapitoly, kde popisují důležité tagy pro mapování turistických tras (2.4), tagy `route` a `kct_barva`



mají ohledně své hodnoty víceméně stejný účel. Oba rozlišují několik typů dané relace, kde hodnoty tagu `route` jsou obecnějšího charakteru. V některých situacích by měly být hodnoty těchto tagů identické. Pokud nejsou, případně jsou v jiném rozporu (např. `kct.barva=peak` a `route=wheelchair`), lze takovou relaci označit za kandidáta na opravu. Podobně z tagu `osmc:symbol` lze nepřímo vyčíst typ trasy, a tak i tento se může zapojit do této kontroly.

- **nepovolené hodnoty** - Tato relativně obecná kontrola může vyčlenit ty relace, jenž obsahují hodnoty mimo množinu povolených v daném případě. Toto se týká těch značek, které neposkytují místo pro invenci a mají tuto množinu pevnou a pro tento účel neměnnou (jako např. `complete=yes/no`).
- **a další...** - Kontroly lze relativně jednoduše vymýšlet dle potřeby konkrétní lokace, uvést zde dopodrobna všechny není nutné.

### 3.3 Poloautomatické možnosti kontroly

Poloautomatické možnosti jsou možnosti, které kombinují jak ruční tak automatické metody. Kontrolu tedy provádí uživatel, avšak strojová kontrola může uživatele navést na relaci, kde se nachází nesrovnalosti, případně na oblast, kde se vyskytuje více chyb pohromadě. V ruční kontrole a následné opravě by měly uživateli být nápomocny také informace ze zdrojů mimo OpenStreetMap data. Na podobném principu by měla stát také aplikace v rámci této bakalářské práce.

## Kapitola 4

# Návrh aplikace pro kontrolu turistických tras

Na základě předchozích dvou kapitol jsem navrhnul webovou aplikaci pro účel poloautomatické kontroly turistických tras KČT.

### 4.1 Struktura aplikace

Aplikace má podobu klasické internetové mapy s přidanou funkcionalitou ve formě nástroje pro samotnou kontrolu. Do mapy jsou přidány překryvné vrstvy, které zobrazují turistické trasy, rozcestníky, a některé kontroly. Důležitou funkcionalitou je především umožnění uživatelům vkládání vlastních vstupů ve formě komentářů nebo obrázků. Dalším důležitým aspektem je umožnění využití více zdrojů dat pro kontrolu. Data z OpenStreetMap se zpracovávají a ukládají do databázového subsystému. Zde se také bude ukládat uživatelem zadaná data, která se vztahují k určitému bodu nebo k určité části trasy. Jelikož se jedná o mapovou aplikaci, která je v principu náročná na stahování a vykreslování dlaždic, je potřeba zvolit asynchronní způsob komunikace mezi zobrazením dat a databází. Pro použití jazyků se tedy nabízí značkový jazyk HTML spolu se skriptovacím jazykem JavaScript s využitím knihovny jQuery a technologie AJAX. Pro získávání dat z databáze a některé kontroly jazyk PHP. Aplikace by měla také zobrazovat výstup ve formě přehledových statistik a tabulek pro snadnější hromadné zpracování. Tuto část lze implementovat jako internetovou stránku pomocí technologií HTML a PHP s přístupem do databáze pro získání dat.

### 4.2 Nástroje pro práci s daty

Existuje vícero různých nástrojů, se kterými se lze dopracovat k požadovanému stavu databáze. Jelikož neexistuje jeden nástroj, který by v sobě zahrnoval robustní databázový systém s geometrickými datovými typy a s funkcí pro zpracování OpenStreetMap dat, je potřeba použít kombinaci několika nástrojů. Zde jsem se snažil blíže popsat ty, které jsem vyhodnotil jako nejvhodnější pro výslednou aplikaci.

### 4.2.1 PostgreSQL

PostgreSQL<sup>1</sup> je pokročilý a robustní objektově-relační databázový systém. Jedná se o open-source multiplatformní systém, je dostupný na všech dnes hlavních operačních systémech včetně Linuxu, FreeBSD, Solaris, MacOS a Windows. Je vyvíjen komunitou vývojarů po celém světě. PostgreSQL je v současné době využíván jak projektem OpenStreetMap pro svou centrální databázi, tak i většinou ostatních projektů využívajících OSM data. Především díky tomuto faktu se jedná o nejlepšího kandidáta pro mou bakalářskou práci. PostgreSQL implementuje velkou část SQL:2011 standardu<sup>2</sup> a nabízí mj. trigger, cizí klíče, komplexní dotazy a transakční integritu.

### 4.2.2 PostGIS

PostGIS<sup>3</sup> je rozšíření pro PostgreSQL databázi pro prostorová data. Stejně jako PostgreSQL se jedná o open-source software. PostgreSQL již sice obsahuje některé geometrické datové typy, ovšem používají se právě ty, které přináší PostGIS. Ve spolupráci s nástrojem Mapnik se PostgreSQL a PostGIS v projektu OpenStreetMap využívají pro vytváření čtyř hlavních vrstev mapy na hlavní stránce OSM. Nové datové typy zahrnují geometrické typy pro body, lomené čáry, polygony, multibody, multipolygony apod. Dále zahrnují prostorové funkce průniku geoprvků, pro určení vzdálenosti, výměry, obvodu ploch a další.

### 4.2.3 Osmosis

Pokud pro svou databázi vybereme schéma `pgsnapshot`, bude potřeba použití programu Osmosis<sup>4</sup>. Jedná se o speciální aplikaci pro zpracování OSM dat a jejich nahrání do databáze. Mezi nejdůležitější schopnosti tohoto nástroje patří to, že dokáže vytvořit tzv. `change set` a ten aplikovat na databázi. Funguje i opačně, tzn. dokáže vygenerovat soubor ve formátu používaném OSM (`planet dump`, 2.3.1), nebo dokáže porovnat dva různé OSM soubory a vytvořit `change set`. Ceněnou vlastností je, že dokáže redukovat velký OSM soubor (např. svět, Evropa) na menší (ČR). Podobných nástrojů existuje více, jedním z rozšířenějších je např. `osm2pgsql`<sup>5</sup>.

### 4.2.4 Osmconvert

Osmconvert<sup>6</sup> je utilita, která má za úkol zpracování OSM souborů a konverzi mezi OSM soubory. Tento program toho umí méně než univerzální Osmosis, je však rychlejší, a tak se nabízí možnost využít jej pro některé specifické úkony. Mezi typické úkony programu patří: vytvoření OSMChange souboru (tzv. `diffu`) ze dvou datových souborů, ořezání souboru podle zeměpisné šířky/délky a podle polygonu, získání statistických dat o souboru atp.

### 4.2.5 Osmfilter

Program Osmfilter<sup>7</sup> se používá pro filtrování OSM souborů podle vybraných tagů. Aplikace umožňuje aplikovat různé typy filtrů (např. na uzly, relace...) včetně jejich závislostí, např.

---

<sup>1</sup>PostgreSQL: <http://www.postgresql.org/>

<sup>2</sup>[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=53681](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53681)

<sup>3</sup>PostGIS: <http://postgis.net/>

<sup>4</sup>Osmosis: <http://wiki.openstreetmap.org/wiki/Osmosis>

<sup>5</sup>osm2pgsql: <http://wiki.openstreetmap.org/wiki/Osm2pgsql>

<sup>6</sup>Osmconvert: <http://wiki.openstreetmap.org/wiki/Osmconvert>

<sup>7</sup>Osmfilter: <http://wiki.openstreetmap.org/wiki/Osmfilter>

cesty a uzly k relaci. Je doporučeno pro rychlejší zpracování používat soubory ve formátu o5m, s jejich získkem pomůže již zmíněný Osmconvert(4.2.4). Tento program lze také využít pro generování podrobných statistických dat ohledně použitých tagů.

### 4.2.6 Osmupdate

Osmupdate<sup>8</sup> je malý a jednoduchý program, jehož jedinou úlohou je nashromáždění a stáhnutí změn různých kategorií (tzn. denní, hodinové, minutové). Tato taktika umožňuje mít aktuální databázi kdykoliv, ať už se aktualizuje pravidelně či nepravidelně. Nejčastějším užitím je pravděpodobně vytvoření nového aktualizovaného souboru na základě starého. Aplikace pro svou činnost využívá utilitu Osmconvert(4.2.4).

## 4.3 Databáze

Funkční databáze je jedním z pilířů této práce. Důležitý je přístup k databázi. Při mnou zvoleném databázovém systému PostgreSQL (4.2.1) a jeho doplňku PostGIS (4.2.2) lze využívat klasický přístup do tabulek jako u jiných systémů. Je potřeba znát síťové jméno serveru, port serveru<sup>9</sup>, jméno příslušné databáze, uživatelské jméno a heslo.

### 4.3.1 Výběr schématu databáze

Při práci s daty z databáze v projektu OSM lze využít různá databázová schémata pro různé aplikace. Vybírat lze podle více kritérií. Jedním z těch nejdůležitějších z hlediska této práce je podpora schématu pro udržování databáze aktuální pomocí tzv. OsmChange diffů (2.3.2). Díky této vlastnosti stačí vložit objemný soubor s kompletními daty do databáze jen jednou. Mezi další důležité vlastnosti patří to, zda schéma uchovává všechna OSM data, nebo jestli má schéma vestavěnou podporu geometrie. Důležitým parametrem je také to, zda schéma využívá jako databázi PostgreSQL. Databázových schémat pro OSM existuje celá řada<sup>10</sup>, pro náš účel podle zadaných kritérií nejlépe vyznívá databázové schéma `pgsnapshot`.

## 4.4 Zobrazení dat

Aplikace z principu vyžaduje zobrazení dat jak v podobě tabulek a statistik, tak především ve formě mapy. Zde je výhodné využít některou z dostupných knihoven, které se k tomuto účelu hodí. K podobným účelům se dnes využívají<sup>[13]</sup> nejčastěji tyto:

- **Google maps** - Historicky nejjednodušší způsob, jak vložit na svou webovou prezentaci mapu. Mezi výhody Google patří nepochybně rychlost, spolu s velkým počtem různých funkcí. Nevýhodou je malá volnost při přizpůsobování aplikace na míru v porovnání s open-source knihovnami.
- **OpenLayers** - Dnes pravděpodobně nejpoužívanější open-source knihovna pro tvorbu webových map. Výhodou je rozsáhlá paleta funkcí, z čehož pramení relativně velká velikost kódu (téměř 1MB).

<sup>8</sup>Osmupdate: <http://wiki.openstreetmap.org/wiki/Osmupdate>

<sup>9</sup>implicitně 5432

<sup>10</sup>výčet schémat lze nalézt zde: [http://wiki.openstreetmap.org/wiki/Databases\\_and\\_data\\_access\\_APIs](http://wiki.openstreetmap.org/wiki/Databases_and_data_access_APIs)

- **Leaflet** - Novější open-source knihovna, jenž se vyznačuje malou velikostí a zaměřením na výkon.

#### 4.4.1 Leaflet

Pro svou aplikaci jsem vyhodnotil jako nejlepší volbu JavaScriptovou knihovnu Leaflet[14] pro její jednoduchost, a z toho pramenící flexibilitu při programování. Jedná se o open-source knihovnu. K Leafletu patří také relativně velké množství dostupných zásuvných modulů, které dokáží některé operace výrazně zjednodušit nebo zrychlit. V její prospěch hraje také přehledná dokumentace. Nevýhodou oproti dalším kandidátům je menší rozsah funkcí, což však vynahrazují právě dostupné plug-iny a volnost při programování.

### 4.5 Mapový podklad

Výsledná aplikace bude potřebovat pro správnou funkčnost mapový podklad, na který budou vykreslovány vrstvy (např. turistické trasy). Mapový podklad je typicky reprezentován bitmapovými dlaždicemi[15] (tiles) o rozměrech 256x256 pixelů, které na sebe navazují a jsou uspořádány do mřížky. Takto poskládaný vedle sebe tvoří na pohled souvislou mapu. Pro každé přiblížení mapy se používá jiná sada dlaždic, s úrovní detailů odpovídající aktuálnímu přiblížení. Nejvíce obrázků je tedy vyžadováno pro nejdetailejší zobrazení mapy.

Existují nástroje pro renderování map<sup>11</sup>, je to však proces náročný na zdroje a především na diskový prostor. Výhodnějším z hlediska této práce je využít již existující podklad, který je k dispozici na určitých serverech<sup>12</sup>. Serverů, jenž poskytují mapové podklady, je více. Odlišují se od sebe grafickou reprezentací dlaždic, jejich určením, nebo například podmínkami užívání.

### 4.6 Vytváření překryvných vrstev

Pro vytvoření překryvných vrstev existují dva odlišné způsoby. Prvním z nich je vygenerování částečně průhledných mapových dlaždic na serveru a poté je zobrazovat jako další vrstvu. Tento postup má výhodu především v rychlosti na straně klienta. Já jsem pro tuto aplikaci zvolil druhý způsob, kde se data vykreslují přímo do mapy jako překryvná vrstva knihovny Leaflet. Tento způsob může více zatížit klientskou stranu aplikace, což vynahrazuje teoreticky menší zátěž a menší obsazená disková kapacita na straně serveru. Také je tento způsob vhodnější pro častější změny zobrazených dat a programování uživatelského rozhraní je intuitivnější a přímočařejší.

### 4.7 Návrh uživatelského rozhraní

Pokud uživatel pracuje s mapou, je potřeba více dbát na rozvržení prostoru pro zobrazení dat a manipulaci s daty. Ovládací prvky jsou rozprostřeny po stranách okna prohlížeče. Nejvýraznějším prvkem je postranní panel na pravé straně obrazovky, který zajišťuje většinu ovládání a funkčnosti, včetně nastavení zobrazení vrstev. Ostatní prvky (vyhledávání, posuvný zoom, měřítko atp.) jsou rozmístěny podle vzoru jiných mapových serverů pro rychlejší a intuitivnější ovládání mapy.

<sup>11</sup>např. Mapnik - <http://wiki.openstreetmap.org/wiki/Mapnik>

<sup>12</sup>přehled dostupných serverů: [http://wiki.openstreetmap.org/wiki/Tile\\_servers](http://wiki.openstreetmap.org/wiki/Tile_servers)



Obrázek 4.1: Ukázka dlaždice vytvořené na serveru tile.openstreetmap.org

#### 4.7.1 Zadávání dat uživatelem

Uživatel by měl být schopen zadávat data do aplikace, ať už pro svou potřebu, nebo pro ostatní uživatele. Způsobů zadávání by mělo být vícero:

- Vytvoření bodu na mapě, který bude obsahovat nějakou informaci.
- Vyznačení části cesty, opět s informací. Ideálně by mělo jít označit i cesty, které nejsou součástí tras KČT.
- Vložení informace přímo k již existujícímu bodu - rozcestníku.

Vloženou informaci může být komentář, případně rovnou fotografie (např. rozcestníku). Uživatel by měl pro přehled uvést uživatelské jméno (které se mu poté bude předvyplňovat) a o jaký typ informace se jedná (zhodnocení stavu, pouze komentář atp.). Také by měl zadat datum, ke kterému je informace platná. Uživatel by měl mít také možnost tato data importovat hromadně.

# Kapitola 5

## Implementace aplikace

Aplikace byla implementována jako internetová stránka, která je rozdělena na mapovou a tabulkovou část. Obě části komunikují s databázovým systémem. Mapová část byla na implementaci náročnější, proto se jí budu v této kapitole věnovat více.

### 5.1 Adresářová struktura

Také adresářová struktura je rozdělena především na složky `map` a `tables`, které obsahují svou část aplikace. Každá část má dále podobnou strukturu. Soubory jsou rozříděny do složek podle svého typu (kaskádové styly, JavaScript, PHP atp.). Jádrem této aplikace je soubor `OsmHicheck.js`, umístěný v mapové části ve složce `js`. V mapové části se ve složkách `js` a `css` nachází složka `leaflet`, kde se nachází soubory stejnojmenné knihovny a použitých zásuvných modulů.

### 5.2 Databázová část

Z dostupných databázových systémů byl vybrán PostgreSQL (4.2.1) verze 9.3.6 s doplňkem PostGIS (4.2.2) verze 2.1.5, protože obsahuje podporu pro geografická data a funkce pro práci s nimi. Byly zvoleny v tu dobu nejnovější stabilní verze, držet se některých starších nemá v tomto případě význam.

#### 5.2.1 Konfigurace databáze

V databázovém systému byla vytvořena databáze, která obsahuje jak OSM data, tak data zadaná uživateli. Po vytvoření databáze bylo potřeba povolit rozšíření `postgis`, které obsahuje podporu geometrických typů, a také rozšíření `hstore` pro podporu efektivního uložení tagů. Tyto doplňky jsou vyžadovány mnou zvoleným schématem `pgsnapshot` (4.3.1).

Po této přípravě je možné nahrát do databáze schéma `pgsnapshot` prostřednictvím importu sql souboru, který je distribuován spolu s nástrojem `osmosis` (4.2.3). Vedle základního sql skriptu pro nahrání schématu jsem použil ještě další rozšíření ve formě skriptu, a to uložení uzlů cesty ve formě tzv. `linestringu`<sup>1</sup> přímo v tabulce `ways`, což výrazně ušetří přístupy do databáze. Další dostupné rozšíření mi umožňuje tamtéž uložit krajní body celé cesty ve formě tzv. `bounding boxu`, což je další geometrický typ rozšíření `postgis`. Za

---

<sup>1</sup>geometrický typ rozšíření `postgis`, umožňující uložení více bodů do jedné položky

zmínku stojí, že nástroj `osmosis` pozná, že jsou tato sql rozšíření použita a sám automaticky tyto hodnoty do tabulky uloží při importu.

## 5.2.2 Prvotní import

Nyní jsem už mohl provést samotné naplnění databáze prvními daty. Protože opětovné nahrávání v určitém intervalu je časově náročná operace, obecně tento import proběhne jednou na začátku, poté se data aktualizují jiným způsobem (5.2.3). Samotný proces prvotního importu lze rozčlenit do pěti následujících kroků:

- **A) Stáhnutí dat z OpenStreetMap.** Jelikož pro účely této aplikace je zbytečné stahovat data z celé planety, postačil mi k tomuto účelu datový sklad serveru geofabrik.de [16], kde jsou uložena denně aktualizovaná data nejen z České republiky ve formátech `osm.bz2`, `osm.gz` a `pbf` (2.3.1). Protože jsem soubor v dalším kroku konvertoval na jiný formát, vybral jsem nejmenší `pbf`.
- **B) Konverze formátu.** Pomocí nástroje `osmconvert` (4.2.4) byl stažený soubor pře-konvertován z binárního formátu `pbf` na formát `o5m`, ve kterém je následující operace efektivnější.

```
osmconvert downloaded_dump.osm.pbf converted_dump.o5m
```

- **C) Odfiltrování nepotřebných dat.** Protože aplikace potřebuje jen data ohledně turistických tras, v tomto kroku jsem je z předchozího souboru exportoval pomocí utility `osmfilter` (4.2.5). Pro turistické trasy tento nástroj exportoval všechny relace s tagem `operator` a jeho hodnotou `cz:KČT`, která se používá jako označení turistických relací v ČR. Kromě těchto relací se při tomto kroku exportují také s nimi svázané cesty a uzly. Aby uživatel mohl vyznačit část cesty, je potřeba v databázi také nechat všechny pro tento účel použitelné cesty.

```
osmfilter converted_dump.o5m --keep-ways=highway=  
--keep=operator=cz:KČT --out-o5m > filtered_dump.o5m
```

- **D) Konverze formátu zpět.** Pro import do databáze je výhodnější použít soubor ve formátu `pbf`. Pro tuto operaci jsem opět využil nástroj `osmconvert`.

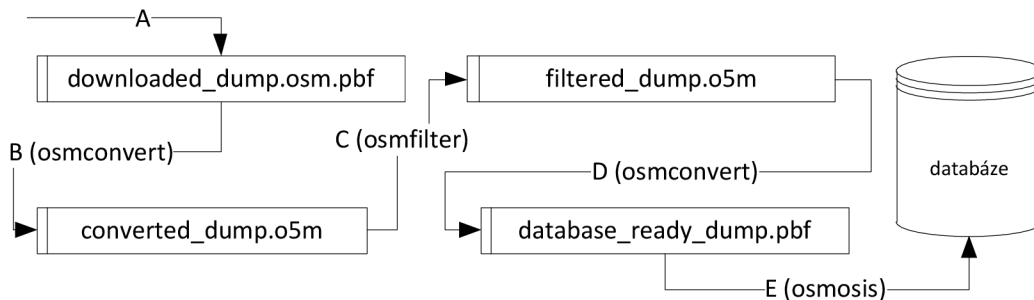
```
osmconvert filtered_dump.o5m first_import.pbf
```

- **E) Import do databáze.** Pro import jsem použil nástroj `osmosis`.

```
osmosis --rb first_import.pbf --wp database=... user=... password=...
```

Nástroj `osmosis` dokáže tyto úkoly provést i sám a ve stejné kvalitě, avšak z hlediska rychlosti je efektivnější použít na jednotlivé úkoly specializovanější nástroje. Pro porovnání, zatímco samotný program `osmosis` zvládne tento import (body B-E) za přibližně 12 minut, sekvence všech těchto příkazů to stihne průměrně za 8 minut. Za zmínku také stojí, že z hlediska přípravy na aktualizace databáze je nutné ponechat soubory z bodu B) a C).





Obrázek 5.1: Schéma samostatného databázového importu. Velká písmena značí operace dle výčtu výše. V závorce je uveden nástroj, pomocí kterého je krok proveden

### 5.2.3 Pravidelná aktualizace databáze

Proces aktualizace databáze se v mnohém podobá prvotnímu importu. Hlavní rozdíl je v tom, že není potřeba stahovat specifický soubor a tak bylo výhodné posloupnost příkazů automatizovat do podoby skriptu v jazyce `bash`. Tento skript byl napsán při snaze zohlednit možnou znovupoužitelnost v jiných projektech. Samotný akt aktualizace lze opět rozdělit do několika na sebe navazujících částí.

- **A) Stažení aktualizace.** V tomto kroku jsem použil nástroj `osmupdate` (4.2.6), který stáhnul relevantní aktualizaci a aplikoval ji na původní soubor ( B) v 5.2.2). V tomto bodu existují dva `o5m` soubory, původní a aktualizovaný. Původní po tomto kroku není potřeba a může být nahrazen souborem aktualizovaným, který se v další iteraci stává opět souborem původním.

```
osmupdate converted_dump.o5m updated_converted_dump.o5m
```

- **B) Odfiltrování nepotřebných dat.** Protože program `osmupdate` stahuje všechny aktualizace bez ohledu na to, jaká oblast se v databázi nachází, je potřeba tento aktualizovaný soubor odfiltrovat stejným způsobem jako v kroku C) v podkapitole Prvotní import (5.2.2).

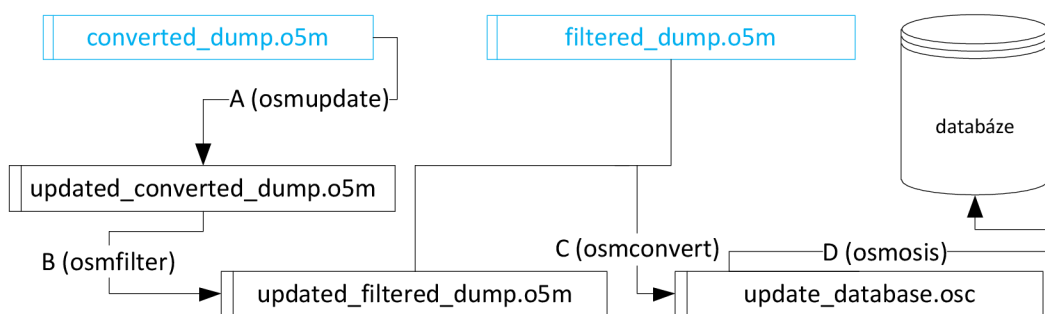
```
osmfilter updated_converted_dump.o5m --keep-ways=highway=
--keep=operator=cz:KČT --out-o5m > updated_filtered_dump.o5m
```

- **C) Vytvoření OSM change souboru.** Zde se v každé iteraci porovnává obsah původního vyfiltrovaného souboru a souboru vzniknutého v původním kroku. Výsledek porovnávání se zapisuje do souboru `osc` (2.3.2).

```
osmconvert filtered_dump.o5m updated_filtered_dump.o5m --diff
-o=update_database.osc
```

- **D) Aktualizace databáze.** Soubor `osc` lze již programem `osmosis` bez problémů importovat do databáze a tím ji aktualizovat.

```
osmosis --rx file=update_database.osc --wpc database=database=...
        user=... password=...
```



Obrázek 5.2: Schéma jednoho cyklu aktualizace databáze. Velká písmena značí operace dle výčtu výše. V závorce je uveden nástroj, pomocí kterého je krok proveden, modře jsou uvedeny soubory z předchozí iterace.

Jeden cyklus aktualizace databáze má dobu trvání určenou především délkou času mezi cykly. Update OSM dat, od jejichž poslední aktualizace uplynuly 2 hodiny, trvá přibližně 7 minut.

### 5.2.4 Tabulky pro uživatelská data

Kromě tabulek, které jsou součástí schématu `pgsnapshot`, potřebuje mnou implementovaná aplikace také tabulky, které uchovají uživatelské vstupy aplikace. Tyto 3 tabulky svou strukturou odpovídají požadavkům, které jsou blíže popsány v návrhu aplikace (viz 4.7.1). Tabulky, které obsahují jako součást uživatelského vstupu také geometrické údaje (bod, řetězec linií), potřebují také specifický druh geografického sloupce. K tomu posloužila funkce doplnku PostGIS (viz 4.2.2) `AddGeometryColumn`<sup>2</sup>. Tabulky pro uživatelská data jsou od tabulek schématu `pgsnapshot` odlišena vlastním schématem `hichack`. Pro účely uchovávání statistik o kontrolách byla vytvořena tabulka `stats`, kam se při každé aktualizaci OSM dat uloží aktuální procentuální hodnota počtu relací s chybějícími tagy a počtu relací s chybami.

## 5.3 Mapová část

Mapa je implementována jako internetová stránka, do které je pomocí knihovny Leaflet vykreslen objekt s mapou a jejími ovládacími prvky. Veškerá funkcionalita je kvůli svému charakteru implementována na straně uživatele ve formě JavaScriptových skriptů. Pokud je potřeba stáhnout z databáze data, je zavolána příslušná funkce, ve které se vytvoří

<sup>2</sup>více o této funkci zde: <http://postgis.net/docs/manual-2.1/AddGeometryColumn.html>

pomocí technologie AJAX požadavek, které se zašle příslušnému PHP skriptu. Ten vybere požadovaná data z databáze a serializovaná je vrátí klientovi. V mapové části jsou data přímo zobrazována na základě své geografické polohy, což usnadňuje orientaci. Základem mapy je knihovna Leaflet, která z velké části řídí vlastní zobrazení dat. Jako podkladová vrstva jsou zvoleny základní dlaždice OpenStreetMap (4.5). Na tomto základu se poté zobrazují překryvné vrstvy, které si vybere uživatel.

### 5.3.1 Komunikace s databází

Mapa musí pro zobrazení překryvných vrstev komunikovat s databází. Komunikace probíhá asynchronně pomocí technologie AJAX, prohlížeč tedy nemusí mapu stále načítat. O samotný přístup do databáze se stará série PHP skriptů.

### 5.3.2 Vykreslení překryvné vrstvy

Postup vykreslení překryvné vrstvy (v tomto případě konkrétně turistických relací) lze popsat následujícími kroky:

- **A) Povolení vrstvy uživatelem.** Poté, co je vrstva povolena, přidá se do objektu mapy. Potom převezme řízení funkce pro získání dat z databáze.
- **B) Získání seznamu relací.** Získají se geografické hranice okna prohlížeče a na jejich základě se vygeneruje seznam relací, jejichž cesty zasahují do daného výřezu. Seznam relací se generuje pomocí PHP skriptu, data se mezi klientskou stanicí a serverem přenášejí pomocí HTTP požadavku.
- **C) Vykreslení relací.** Na základě získaného listu se z databáze získají bližší informace o relacích a jejich cestách. Tato informace se přenesou klientovi. Relace se nyní postupně vykreslují díky získaným geografickým údajům.
- **D) Zobrazení mapy s vrstvou.** Pokud uživatel změní zoom nebo pohne mapou, pokračuje se od kroku B. Změna zoomu má za následek nové načtení vrstvy. Při pohybu mapy se již znova nepřistupuje do databáze k relacím, jenž jsou již vykreslené. Pokud se vykreslená relace dostane při pohybu mapy mimo viditelný výřez prohlížeče, je tato z vrstvy odstraněna.
- **E) Vypnutí vrstvy.** Když uživatel vrstvu deaktivuje, tato se odebere z objektu mapy a vymaže svá dočasná data, tím se uvede do původního stavu.

U jiných vrstev může být popsán způsob rozšíření o některé další kroky. Záleží také na tom, které další vrstvy jsou aktivní. Každá vrstva je reprezentována svým objektem, který obsahuje referenci na samotnou vrstvu, referenci na svou značku a seznam aktuálně vykreslených prvků.

### 5.3.3 Optimalizace vykreslení

S vyšším oddálením mapy rostou při zapnutých překryvných vrstvách nároky na vykreslování, a to v některých případech velmi výrazně. Problém se týká především těch vrstev, které jsou reprezentovány liniovými řetězci. Zde jsem problém alespoň částečně vyřešil implementací filtru uzlů. Filtr pracuje na straně serveru při stahování údajů o cestě. Vlastní filtrování probíhá tím způsobem, že se porovnají směry dvou po sobě jdoucích vektorů,

kde se jeden vektor skládá ze dvou sousedících bodů. Pokud je rozdíl těchto směrnic menší než stanovená mez, bod se vyčlení a nepošle se klientovi. Mez je stanovená aktuálním přiblížením, tzn. mez je větší při větším oddálení. Hodnoty přiblížení jsou voleny jako kompromis mezi plynulostí manipulace s mapou a viditelností filtrace na mapě. Při ještě větším oddálení se potom už relace nezobrazují jako linie, ale jsou reprezentovány značkou.

### 5.3.4 Kontrola chybné relace

Kontrola chyb v relacích, pokud je povolena, probíhá na straně serveru při získávání informací o konkrétní relaci. Kontrola chyb probíhá v rámci vlastní překryvné vrstvy, proto pokud se zde tato relace neoznačí jako chybná, k uživateli se neposílá. Pokud se vyhodnotí opačně, spolu s relačními daty se pošlou také příslušné chybové kódy, které jsou zpracovány na straně klienta.

### 5.3.5 Kontrola chybějících prvků relace

Pokud je tato vrstva povolena, postup probíhá podobně jako kontrola chybné relace (viz 5.3.4). Při detekci nevyplněného povinného tagu se opět pošlou data klientovi, avšak samotné označení chybějících prvků proběhne na straně klienta.

## 5.4 Uživatelské rozhraní

Kromě standardních uživatelských prvků, které lze na mapě najít běžně (vyhledávání, zoom), má mít aplikace implementovány nástroje pro kontrolu a uživatelské vstupy. Pro ovládání těchto je využit dostupný zásuvný modul Sidebar<sup>3</sup>, který jsem nakonfiguroval tak, aby zobrazoval místo pro ovládací prvky na pravé straně okna prohlížeče. Tento panel je rozdělen na jednoduchou přepínací nabídku ve vrchní části a na obsah ve zbytku panelu. Přepínání okna je realizováno JavaScriptem, pro zobrazení formulářů je pomocí knihovny jQuery načten příslušný html soubor.

### 5.4.1 Zobrazení informací

Obecně je v této aplikaci vyřešeno zobrazení informací tak, že se zobrazí po kliknutí na příslušný prvek zobrazený v mapě (relace, bod, část cesty). Pokud uživatel klikne na prvek zobrazené turistické relace, do proměnné se uloží ID této relace a navíc ID cesty, na kterou uživatel kliknul. Poté se provede asynchronní požadavek, serverový PHP skript dodá klientské části všechny informace k těmto prvkům, které se zobrazí v postranním panelu. Také se vygeneruje odkaz pro přímou editaci těchto prvků v editoru JOSM<sup>4</sup>. Pokud zobrazovaná relace při svém stažení získá informace o svých chybách, ty se na klientské straně zpracují a v tabulce se zobrazí zvýrazněně.

### 5.4.2 Vkládání informací

Uživatel může vkládat do mapy (a potažmo do databáze) informace třemi různými způsoby. Každý způsob sám o sobě je součástí své vlastní vrstvy, kterou je možné ovládat.

<sup>3</sup>seznam zásuvných modulů Leafletu: <http://leafletjs.com/plugins.html>

<sup>4</sup>více o tomto editoru zde: <https://josm.openstreetmap.de/>

- **Vložení bodu.** Pokud uživatel klikne na tento odkaz, aplikace čeká na jeho vstup v podobě kliknutí do mapy na místo, kde chce vložit bod s poznámkou. Když se tak stane, vytvoří se na tomto místě dočasná značka, která se liší barvou od jiných. V postraním panelu se mezitím načte formulář z předpřipraveného `html` souboru. Po stisknutí potvrzovacího tlačítka se provede validace uvedených hodnot a informace se asynchronně posílají na server, kde se pomocí PHP skriptu uloží do databáze. Po úspěšném uložení se daná vrstva načte znovu a dočasná značka se smaže, aby se okamžitě reflektovaly provedené změny.
- **Vyznačení části trasy.** Zde se opět čeká na uživatelský vstup, jen kromě bodu uživatel vybírá z již zobrazených cest. Pro tento účel je zde implementována zvláštní vrstva, která při tomto vstupu a určitém přiblížení zobrazí kromě turistických tras i všechny ostatní trasy v mapě. Jindy se vrstva nezapíná. Prozatím je výběr části limitován délkou cesty tak jak je uložena v databázi. Implementace této části byla ulehčena použitím zásuvného modulu `LineStringSelect`. Po výběru obou konců cesty je postup obdobný jako při vkládání bodu.
- **Vložení informace k existujícímu rozcestníku.** Tato uživatelská interakce předpokládá zapnutou vrstvu s rozcestníky. Informace se vkládá podobným způsobem jako u předchozích prvků. Zobrazuje se při zobrazení dat o rozcestníku. Zde nemají informace vlastní vrstvu (nesou samy o sobě informaci o své poloze), jejich stahování započne při kliknutí na značku konkrétního rozcestníku.
- **Hromadný import dat.** Zde se data přečtou z příloženého souboru a uloží do databáze. Struktura souboru závisí od typu importovaných dat, jenž je jeden ze tří (viz předchozí položky seznamu). Obrázek v tomto případě musí být uložený ve formě dostupného odkazu ke stažení.

Listing 5.1: Ukázka JSON souboru pro hromadný import informací k rozcestníkům

```
[ { "user": "Petr Švaňa", "type": "PROBLEM",
  "note": "Na tomto miste chybi rozcestnik",
  "image": "http://example.cz/image.png", "date": "11/5/2015",
  "node_id": "1445764671" },
  { "user": "Petr", "type": "OK",
  "note": "Zkontrolovano, v poradku",
  "image": "http://example.cz/image2.png", "date": "11/5/2015",
  "node_id": "1445764671" },
  { "user": "Petr", "type": "COMMENT", "note": "komentar",
  "image": "", "date": "11/5/2015", "node_id": "1445764674" } ]
```

### 5.4.3 Uložení stavu mapy

Při použití zásuvného modulu `Leaflet.Hash` se do adresního řádku ukládají informace o zeměpisné poloze a aktuálním přiblížení okna. K těmto informacím jsem úpravou tohoto modulu přidal také vlastnost přechovávání informace o vyplněném uživatelském jménu, které se uchovává po prvním zadání tohoto jména do některého z formulářů, a při dalším použití se již vyplňovat znovu nemusí.

#### 5.4.4 Načítání vlastních souborů z GPS

Pro podporu ruční kontroly, konkrétně porovnání podle vlastních GPS dat, byl do mapy vložen zásuvný modul FileLayer, jenž dokáže přečíst soubory ve formátu GeoJSON, GPX a KML.

### 5.5 Tabulková část

Tabulková část je implementovaná jako jednoduchá webová stránka s přepínacím menu, které přepíná jednotlivé výstupy. Data jsou zobrazeny ve formě tabulek. Při zobrazení chyb se chyby detekují při generování stránky. K vykreslení grafu byla použita PHP knihovna `phplot`<sup>5</sup>.

---

<sup>5</sup>více o této knihovně zde: <http://www.phplot.com/>

## Kapitola 6

# Manuál k výsledné aplikaci

Zatímco v předchozí kapitole jsem popisoval implementaci, tato kapitola popisuje výslednou aplikaci z pohledu uživatele a jeho možností práce s aplikací. Jak již bylo zmíněno na předchozích stránkách, aplikace je rozdělena na dvě na sobě nezávislé části. Volbu mezi částmi lze učinit na úvodní stránce, poté lze mezi nimi přepínat podle potřeby. Tento manuál se zaměřuje především na popis mapové části, ovládání tabulkové části je shrnuto do jedné sekce na konci této kapitoly.

### 6.1 Obecný popis mapové části

Tato část aplikace je podobná jiným internetovým mapám. Ovládací prvky mapy jsou rozmístěny kolem okrajů mapy. Nalevo nalezneme vyhledávání, posuvný zoom a tlačítko pro nahrání vlastních GPS souborů. Napravo je umístěn hlavní ovládací prvek mapy, kterým je postranní panel. Zde je několik odkazů, které zajišťují přepínání obsahu okna.

### 6.2 Volba zobrazení vrstev a podkladu

Volby mapového podkladu a překryvných vrstev se nacházejí na pravém panelu. Toto nastavení je výchozím zobrazením postranního panelu. K nastavení se lze kdykoliv dostat kliknutím na odkaz **Nastavení zobrazení**. Zobrazení mapového podkladu je v současné době omezeno na dvě varianty: klasický mapový podklad známý se serveru OpenStreetMap.org (viz 4.5) a nezobrazení žádného podkladu.

Pro uživatele je k dispozici několik vrstev:

- **A) Turistické relace.** Vrstva s daty původem z OpenStreetMap. Zobrazení závisí na aktuálním zoomu aplikace. Jestliže je jeho hodnota nižší než 12, je jedna relace reprezentována zelenou značkou se symbolem šipky. Jinak je zobrazena klasicky jako linie v mapě s barevnou reprezentací podle uvedeného tagu `kct_barva`. Pokud je zobrazení zoomem přepnuto do klasického zobrazení, je třeba brát v potaz, že od úrovně zoomu 16 a méně jsou trasy vykresleny zjednodušeně pro zrychlení vykreslování (detailněji viz 5.3.3). Toto platí pro všechny zde uvedené liniové vrstvy.
- **B) Rozcestníky.** Tato vrstva zobrazuje rozcestníky turistických relací. Rozcestník je reprezentován modrou značkou se symbolem bodu.

- **C) Zvýraznění relací s chybějícími tagy.** Při klasickém zobrazení je postižená relace vykreslena jako oranžová linie s větší šířkou a průhledností než turistická relace, což tuto trasu zvýrazní při zapnutí obou vrstev. Při větším oddálení se linie transformuje na oranžovou značku také s symbolem šipky.
- **D) Kontrolní vrstva.** Při zapnutí této vrstvy se aktivují kontroly chybných relací. Takto označená relace je na mapě reprezentována podobně jako vrstva C, avšak specifickou barvou je v tomto případě červená. Symbolem na značce je vykřičník.
- **E) Uživatelské poznámky.** Zde se po aktivaci zobrazí body přidané uživateli této aplikace. Bod se zobrazí jako fialová značka se symbolem poznámky.
- **F) Uživateli zvýrazněné úseky cest.** Vrstva obsahuje uživateli zvýrazněné úseky cest světle modrou barvou. Tato cesta se zvyšujícím přiblížením nijak nemění svůj tvar.

Pod výberem zobrazení je zde navíc přítomná volba **Zahrnout cyklotrasy bez barvy**. Takto jsou zahrnuty i ty relace, u kterých není jasné, zda se jedná o trasy cykloturistické nebo cyklistické (více o problému zde: [3.2.1](#)).

### 6.3 Interakce se zobrazenými prvky

Každý prvek, který je vykreslen v rámci některé z překryvných vrstev je interaktivní a provede po kliknutí nějakou akci, typicky se jedná o zobrazení dat. U vrstev, jejichž obsah tvoří sami uživatelé této aplikace (vrstvy E a F v [4.5](#)) se data zobrazují ve formě bubliny nad dotyčným prvkem. U ostatních vrstev se zobrazí data a tagy vztahující se k danému prvku zobrazí v postranním panelu. Je zde také odkaz, jenž umožní přímou editaci dotyčných prvků v programu JOSM. Navíc pokud byl dotyčným prvkem rozcestník, zpřístupní se odkaz pro přidání informace k rozcestníku (viz [6.4.3](#)). Odkaz **Data & Tagy** zajišťuje přístup k těmto datům i z jiného kontextu panelu. Tato diverzita mezi zobrazením uživatelských a OSM dat je zde implementována proto, že např. uživatelská poznámka se může vztahovat k některé relaci a je dle mého pohodlnější mít možnost zobrazit oba druhy informací najednou.

### 6.4 Vkládání uživatelských vstupů

Aplikace umožňuje vkládat vlastní příspěvky, jejichž typy jsou uvedeny níže. Jejich zadání se liší, všechny mají jednu společnou část - formulář pro zadání upřesňujících dat. Po uživateli je požadováno zadání uživatelského jména, kde lze zaškrtnout volbu spojení uživatele s jeho jménem na openstreetmap.org. Při vybrání této volby se poté při zobrazení vstupu u uživatelského jména zobrazí odkaz na uživatelův profil na openstreetmap.org. Uživatelské jméno se v aplikaci ukládá a při dalším použití formuláře bude předvyplněno. Bude také vedle aktuálních souřadnic mapy a přiblížení součástí adresy v adresním řádku prohlížeče, kterou je možné takto uložit a použít pro pohodlnější vyplňování formulářů v budoucnu.



Dalším povinným údajem je výběr typu příspěvku. Zde uživatel vybírá mezi následujícími možnostmi:

- Vše v pořádku
- Je zde problém
- Chci jen vložit komentář/fotografii

Ke každé variantě se hodí uvést doplňující komentář. Posledním společným prvkem je datum, ke kterému je vkládaná informace platná. Standardně je předvyplněno aktuální datum. Datum musí být ve formátu den/měsíc/rok.

Po kliknutí na tlačítko **Vložit** se zadané hodnoty zkontrolují a vloží se do databáze, případně se při chybě vypíše chybové hlášení na začátku formuláře. Tlačítko **Reset** potom celý nedokončený proces vložení uvede do původního stavu.

#### 6.4.1 Vložení uživatelské poznámky - bod

Proces vložení poznámky na specifický bod se aktivuje kliknutím na příslušný odkaz v nabídce. Následujícím krokem je kliknutí do mapy na místo, kde chceme bod vložit. V tomto kroku lze kliknout na případnou zobrazenou trasu bez toho, aby se v postranním panelu zobrazily informace. Po vybrání se na místě kliknutí vytvoří stejná značka jako u již vytvořeného uživatelského bodu, jen v bílé barvě. V postranním panelu se zobrazí formulář, kde lze kromě údajů specifikovaných výše doladit také zeměpisnou šířku a délku. Zde se jako výchozí hodnoty nastaví zeměpisná poloha bílé dočasné značky. Souřadnice lze zobrazit a upravovat buď v klasickém formátu DMS<sup>1</sup>, nebo ve formátu DecDeg<sup>2</sup>. Mezi těmito formáty lze přepínat tlačítkem pod nastavením těchto hodnot. K uživatelskému bodu je také možno nahrát relevantní fotografii (o tomto více zde: 6.4.6). Po úspěšném nahrání se dočasná značka smaže a na místě zadaných souřadnic se objeví už standardní uživatelský bod.

#### 6.4.2 Vložení uživatelské poznámky - zvýraznění části trasy

Pro povolení zvýraznění části trasy je potřeba opět zvolit příslušný odkaz v menu. Nyní je potřeba zvolit část cesty, kterou chceme vyznačit. K tomu účelu lze využít linií, které patří již vykresleným vrstvám. Pokud je hodnota zoomu 15 a více, zobrazí se speciální vrstva, která obsahuje veškeré ostatní použitelné cesty (tzn. s tagem **highway**). Kliknutím na některou cestu se vytvoří na místě bílý kruh, dalším kliknutím na konec úseku se objeví i druhý a cesta mezi nimi bude zvýrazněna sítlem modře. S oběma kruhy lze pohybovat a měnit tak kurčený úse podle potřeby. Délka úseku je dána hranicemi vykreslené cesty. Po vyznačení druhého kruhu se opět objeví v postranním panelu formulář pro vyplnění informací a dále je postup stejný.

#### 6.4.3 Vložení dodatečných informací k rozcestníku

Zde je postup lehce odlišný, neboť se na mapě nevytváří žádný nový objekt. Informace se předávají přímo k rozcestníku, kde se poté zobrazí pod tabulkou s tagy. Pro vložení je potřeba kliknout na vykreslený rozcestník a poté v horní části postranního panelu kliknout

<sup>1</sup>DMS = Degrees Minutes Seconds = Stupně, minuty, sekundy, př: 40° 26' 46"

<sup>2</sup>DecDeg = Decimal degrees = Desetinné stupně, př: 40.446°

na příslušný odkaz pro přidání informace. Zobrazí se již známý formulář, kde lze nahrát také relevantní fotky rozcestníku.

#### **6.4.4 Hromadný import dat**

Nástroj pro hromadný import dat je dostupný po kliknutí na stejnojmennou volbu v nabídce. Uživatel si zde vybere typ odesílaných dat a ze svého počítače nahraje soubor ve formátu `json`. Soubor musí mít odpovídající strukturu pro daný typ dat. Pro každý typ je přiložen ukázkový soubor, ze kterého lze strukturu vyčíst.

#### **6.4.5 Mazání uživatelských vstupů**

Každý uživatelský příspěvek lze z mapy odstranit, pokud z nějakého důvodu není potřeba. Příspěvek však nelze odstranit z databáze. Ačkoliv aplikace nemá implementován systém přihlašování uživatelů, pro zamezení vandalismu je tato operace chráněna zvláštním heslem.

#### **6.4.6 Nahrávání fotografií**

U některých vstupů je možnost nahrávat také obrázky. Lze nahrát jeden obrázek ve formátu `png` nebo `jpeg` a celková velikost nesmí překročit 5 MB. Pokud fotografie přesahuje jedním svým rozměrem 1200 px, bude při ukládání ořezána tak, aby se zachoval poměr stran.

### **6.5 Obecný popis tabulkové části**

Tabulková část je opticky rozdělena na menu v horní části stránky a na tabulku. Základním zobrazením jsou statistická data aplikace, kde může uživatel nalézt procentuální zastoupení chyb a chybějících tagů v relacích. Lze zde také po kliknutí na konkrétní počet chyb vypsát tyto relace v tabulkové formě. Pod statistikami se nachází graf, který značí procentuální chybovost dat v čase. Dalším zobrazením je tabulka, která je dostupná po kliknutí na druhý odkaz v menu. Zde jsou zobrazeny všechny relace po aplikování kontrol, a chybějící nebo chybné hodnoty jsou zde zvýrazněny.

# Kapitola 7

## Závěr

Úkol této práce byl zadáním specifikován následovně: nastudovat systém OpenStreetMap a mapování turistických tras, analyzovat možnosti jejich kontrol a poté navrhnout a implementovat aplikaci pro poloautomatickou kontrolu turistických relací.

Úvod práce pro mě znamenal seznámení se s OpenStreetMap a systémem jeho fungování. Důležitá byla důkladná analýza datového modelu OpenStreetMap, jakožto i seznámení se s reprezentací a distribucí dat. Tyto znalosti bylo potřeba zkombinovat s informacemi o způsobu mapování turistických relací a na jejich základě vymyslet varianty různých kontrol. Kontroly mohou být ruční, kde chybu v relaci detekuje uživatel, a automatické, kde chybu zjistí přímo výsledný program. Aby program splňoval požadavky na poloautomatickou kontrolu, muselo se při návrhu myslet na kombinaci obou přístupů ke kontrolám. V návrhu byl proveden výběr nástrojů pro implementaci a analýza uživatelských potřeb.

Tyto se poté povedlo implementovat a vznikla webová aplikace, která kombinuje jak mapovou část s uživatelskou interakcí, tak i tabulkový výstup zjištěných dat. Obě části mají přístup k databázi, která obsahuje jak OSM data, tak i data od uživatelů aplikace. Tabulky s OSM daty se pravidelně aktualizují. V mapové části je zajištěno vykreslování turistických tras, které bylo optimalizováno pro větší rychlost aplikace. Uživatel má k dispozici celkem 6 vrstev, z toho 4 pracují s OSM daty (turistické relace, 2 kontrolní vrstvy a rozcestníky) a 2 s uživatelskými daty (uživatelské body a úseky). Krom toho může uživatel vkládat informace k rozcestníkům. Při vkládání uživatelského bodu nebo informace k rozcestníku lze také vložit popisnou fotografii.

Rezervy v aplikaci je možné najít při rychlosti reakce aplikace, zde je jistě prostor pro zlepšení. Stejně tak je možné do budoucna přidat další funkcionalitu, a to jak do mapové, tak i do tabulkové části. Aplikace by měla být relativně rozšiřitelná i pro jiné země a tím pádem jiné mapovací pravidla, není to však úplně triviální záležitost a rozhodně by stálo za to aplikaci vylepšit i v tomto směru. Jistě by také šlo vymyslet další sofistikovanější metody kontrol turistických relací.

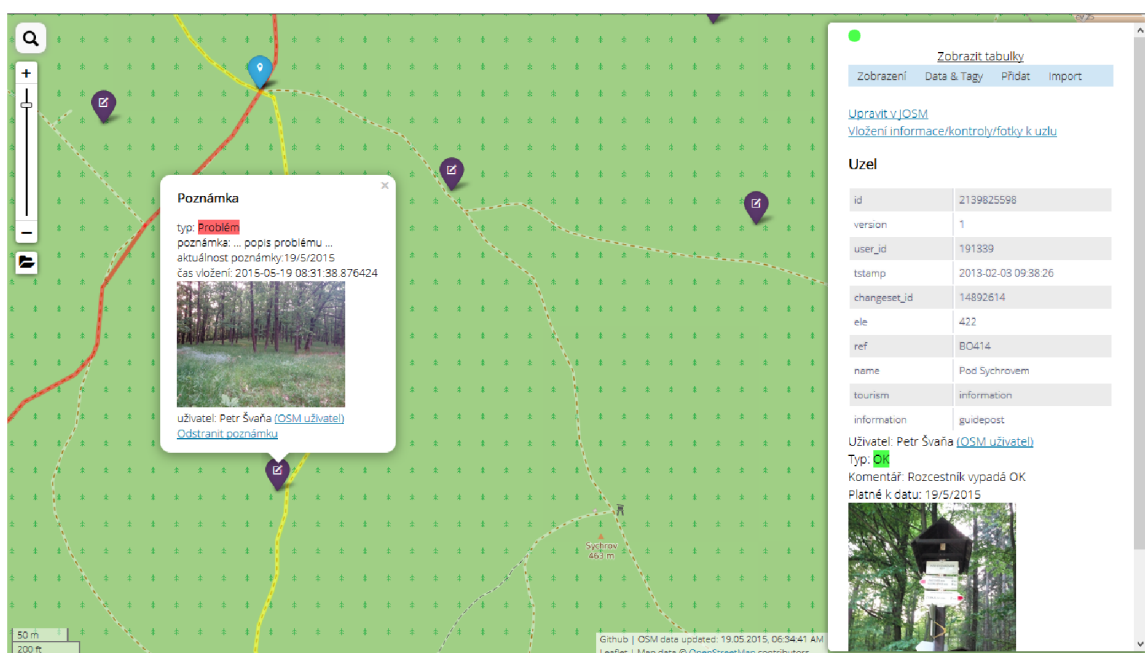
# Literatura

- [1] Open Street Map homepage. <http://www.openstreetmap.org>. Navštíveno: 2014-10-04.
- [2] Elements - openstreetmap wiki. <http://wiki.openstreetmap.org/wiki/Cs:Elements>. Navštíveno: 2014-10-14.
- [3] Relations - openstreetmap wiki. <http://wiki.openstreetmap.org/wiki/Relation>. Navštíveno: 2014-10-14.
- [4] Osm xml - openstreetmap wiki. [https://wiki.openstreetmap.org/wiki/OSM\\_XML](https://wiki.openstreetmap.org/wiki/OSM_XML). Navštíveno: 2014-11-05.
- [5] Osmchange - openstreetmap wiki. <https://wiki.openstreetmap.org/wiki/OsmChange>. Navštíveno: 2015-01-13.
- [6] Wikiproject czech republic/editing standards and conventions - openstreetmap wiki. [http://wiki.openstreetmap.org/wiki/WikiProject\\_Czech\\_Republic/Editing\\_Standards\\_and\\_Conventions](http://wiki.openstreetmap.org/wiki/WikiProject_Czech_Republic/Editing_Standards_and_Conventions). Navštíveno: 2015-03-07.
- [7] Wikiproject czech republic/otm\_značkový\_klíč - openstreetmap wiki. [http://wiki.openstreetmap.org/wiki/WikiProject\\_Czech\\_Republic/OTM\\_\\_značkový\\_klíč](http://wiki.openstreetmap.org/wiki/WikiProject_Czech_Republic/OTM__značkový_klíč). Navštíveno: 2015-03-07.
- [8] Key:network - openstreetmap wiki. <http://wiki.openstreetmap.org/wiki/Key:network>. Navštíveno: 2015-02-19.
- [9] Turistické cesty a značení - zálesáctví.cz. <http://www.zalesactvi.cz/2014/08/turisticke-cesty-a-znaceni/>. Navštíveno: 2015-02-19.
- [10] KČt - turistické značení kČt. <http://www.kct.cz/cms/turisticke-znaceni-kct>. Navštíveno: 2015-02-19.
- [11] Cycle routes - openstreetmap wiki. [http://wiki.openstreetmap.org/wiki/Cycle\\_routes](http://wiki.openstreetmap.org/wiki/Cycle_routes). Navštíveno: 2015-05-14.
- [12] Wikiproject czech republic/cycle ways - openstreetmap wiki. [http://wiki.openstreetmap.org/wiki/WikiProject\\_Czech\\_Republic/cycle\\_ways](http://wiki.openstreetmap.org/wiki/WikiProject_Czech_Republic/cycle_ways). Navštíveno: 2015-05-14.
- [13] Testing web map apis - google vs openlayers vs leaflet. <http://robinlovelace.net/software/2014/03/05/webmap-test.html>. Navštíveno: 2015-05-07.

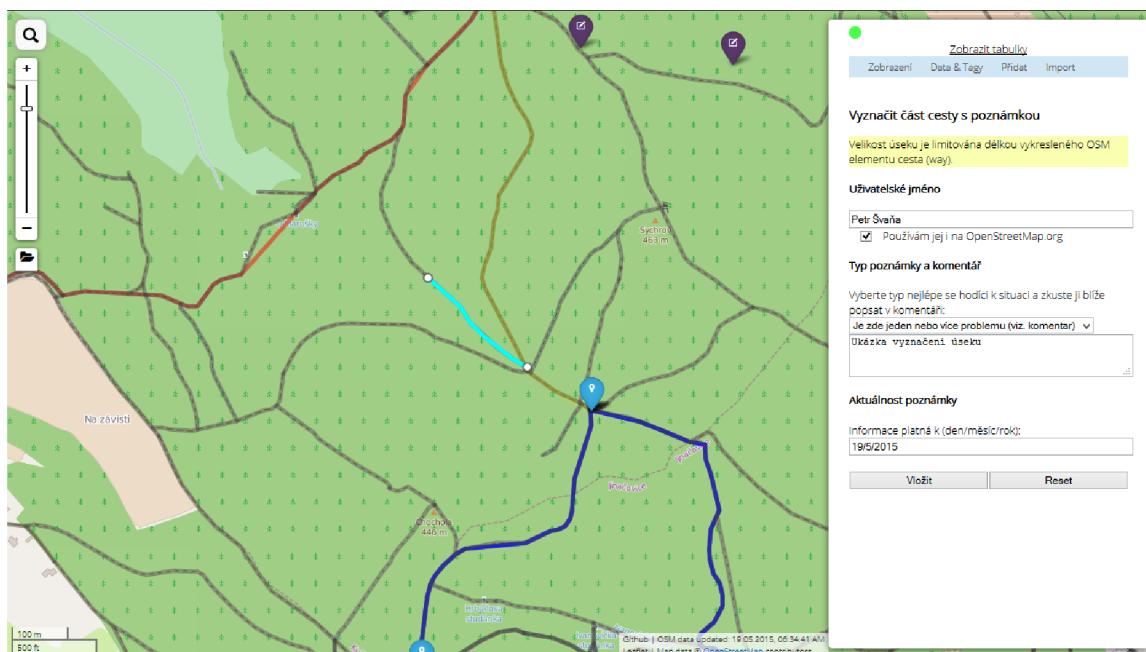
- [14] Leaflet - a javascript library for mobile-friendly maps. <http://leafletjs.com/>.  
Navštíveno: 2015-05-07.
- [15] Tiles - openstreetmap wiki. <http://wiki.openstreetmap.org/wiki/Tiles>.  
Navštíveno: 2015-01-19.
- [16] [OpenStreetMapDataExtracts](http://download.geofabrik.de). <http://download.geofabrik.de>. Navštíveno:  
2015-03-11.

## Dodatek A

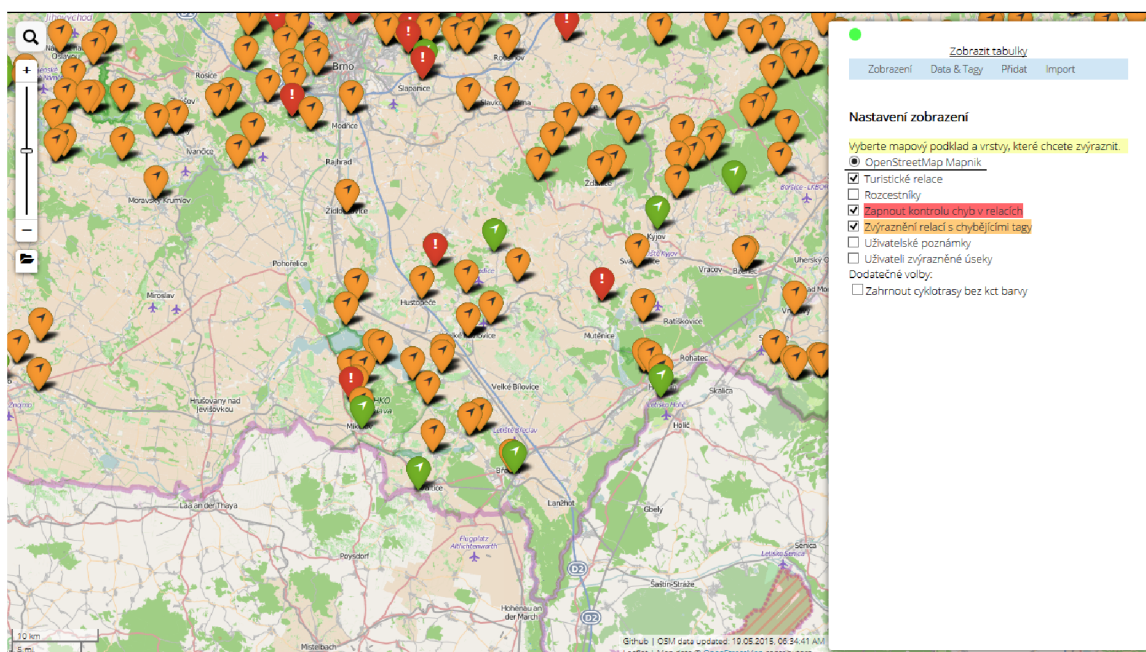
# Ukázky práce s aplikací



Obrázek A.1: Ukázka zobrazení dat z rozcestníku s přidanou informací a zároveň rozkliknuté uživatelské poznámky.

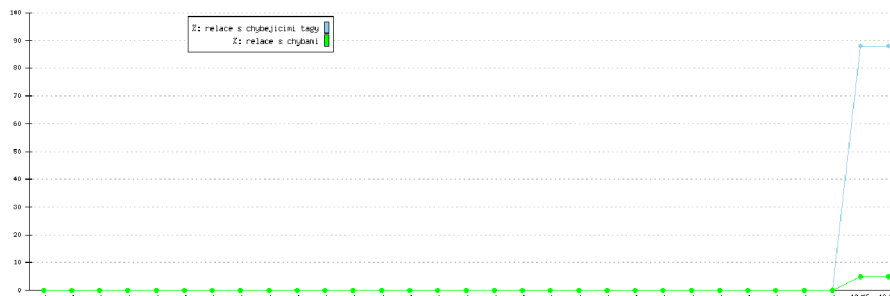


Obrázek A.2: Ukázka vyznačení úseku cesty s poznámkou.



Obrázek A.3: Zobrazení turistických relací s kontrolami při větším oddálení.

Základní statistické údaje		
Pocet KCT turistických relací v databázi:	3684	100%
Pocet těchto relací, kde chybí některý z tagů:	3234	88,20%
Pocet těchto relací, kde se vyskytuje chybný tag nebo rozpor:	211	5,76%
<b>Chybějící hodnoty tagů:</b>		
network:	2491	67,99%
complete:	448	12,23%
osmc:symbol:	26	0,71%
destination:	2168	59,12%
<b>Chybné hodnoty tagů:</b>		
network:	70	1,91%
complete:	7	0,19%
osmc:symbol:	34	0,93%
route:	7	0,19%
kct_*:	2	0,05%
<b>Nalezené rozpor:</b>		
hodnota network nekoresponduje s hodnotou kct_*:	27	0,74%
typ cesty v rozporu s ostatními u tagu osmc:symbol, kct_* nebo route:	56	1,53%
barva cesty v rozporu u osmc:symbol a kct_*:	17	0,46%



Obrázek A.4: Tabulková část se statistikami a průběžným grafem oprav.



## Dodatek B

# Obsah CD

Obsah přiloženého CD má následující adresářovou strukturu:

- **src/** - zdrojové soubory potřebné pro běh aplikace
- **tex-src/** - zdrojové soubory textové části bakalářské práce
- **thesis/** - tato bakalářská práce ve formátu PDF
- **README.txt** - doplňující informace k dostupnosti a spuštění aplikace