

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Prohlížeč map v HTML 5

2014

Jan Alexandr Janíček

Anotace

Výsledkem práce je aplikace, která umožňuje zobrazení veřejně dostupných mapových dat (OpenStreetMaps) s využitím technologií HTML5 a podporuje standardní funkce mapových prohlížečů jako přibližování a oddalování a posun po mapě. Navíc je možné s objekty interaktivně pracovat, např. zvýraznit je nebo zobrazit související informace. Předkládaná práce je rozdělena do 3 hlavních částí. V první je čtenář stručně seznámen s jazykem HTML, novinkami, které přináší, a současným stavem problematiky prohlížečů map. Druhá část je věnována analýze a specifikaci požadavků na aplikaci. Kapitola č. 3 představuje implementaci mapového prohlížeče.

Děkuji vedoucímu práce, Mgr. Petru Krajčovi, Ph.D., za příkladné vedení a cenné rady a děkuji své rodině a přátelům za podporu.

Obsah

1. Úvod	8
1.1. Jazyk HTML5	8
1.1.1. Specifikace jazyka HTML5	8
1.1.2. Nová vize	8
1.1.3. Přehled nových vlastností jazyka	9
1.1.4. Element canvas	10
1.1.5. Podpora ve webových prohlížečích	12
1.2. Prohlížeče map	12
1.2.1. Přehled současných řešení	12
1.2.2. Google Maps	13
1.2.3. GIS Cloud	14
1.2.4. Cartagen	15
1.3. OpenStreetMaps	16
1.3.1. Prohlížeč map OpenStreetMap.org	16
1.3.2. Formát dat	17
1.3.3. OSM Extended API	18
2. Analýza problému a návrh řešení	20
2.1. Cíle projektu	20
2.2. Specifikace požadavků	20
2.2.1. Funkční požadavky	21
2.2.2. Mimofunkční požadavky	22
2.3. Koncepce programu	22
3. Implementace mapového prohlížeče	24
3.1. Proces vývoje	24
3.2. Klientské skripty	25
3.3. HTML a CSS dokumenty	26
3.4. Konfigurační soubor config.php	27
3.5. Serverové skripty	28
3.5.1. selectFromDB.php	28
3.5.2. selectInfo.php	30
3.5.3. updateMap.php	30
3.5.4. indexes.php	31
3.5.5. cronScript.php	31
3.6. Zabezpečení skriptů souborem .htaccess	31
3.7. Databáze	32
3.7.1. Schéma	33
3.8. Uživatelské rozhraní	34
3.9. Zprovoznění aplikace	36
3.10. Testování	37

3.10.1. Test na základní použitelnost	37
3.10.2. Výsledek testu na základní použitelnost	39
3.10.3. Test na selhání	41
3.10.4. Výsledek testu na selhání	42
4. Závěr	43
Reference	45
A. Zadání projektu	49
B. Testovací scénáře	50
C. Obsah přiloženého CD/DVD	54

Seznam obrázků

1.	Cesta na plátně tvořená 3 souřadnicemi.	11
2.	Kontextové menu v prostředí Google Maps	14
3.	Prostředí GIS Cloud	15
4.	Prostředí prohlížeče Cartagen.	16
5.	Prohlížeč OSM	17
6.	Use Case Diagram (Diagram případů užití).	21
7.	Diagram znázorňující architekturu aplikace	23
8.	Uživatelské rozhraní skriptu cronScript.php.	32
9.	Diagram schéma databáze.	33
10.	Uživatelské rozhraní.	34
11.	Ukazatel načtení mapy.	35
12.	Kontextový panel s ikonou umístění nad označeným objektem. . .	35
13.	Panel pro nastavení zobrazení mapy.	36

Seznam tabulek

1.	Seznam testovacích zařízení.	38
2.	Seznam testovacích webových prohlížečů.	38
3.	Výsledek testovacího scénáře č. 1 pro úroveň přiblížení 6.	39
4.	Výsledek testovacího scénáře č. 1 pro úroveň přiblížení 12.	40
5.	Doby načítání kontextového menu na testovacích zařízení.	41
6.	Výsledek aktualizace mapy za hodinu provozu skriptu cron-Script.php.	41

1. Úvod

Při hledání tématu této závěrečné práce mě zaujala možnost zpracovat prohlížeč map, který v běžném životě využívám téměř na denní bázi, a prozkoumat nejmodernější technologie HTML5 [1] a CSS3 [2] pro využití v této problematice. Cílem projektu bylo vyzkoušet dané technologie. Vzniklá aplikace, která ukazuje možnosti těchto technologií, může sloužit jako prototyp pro komplexnější projekt.

Úvodní část práce popisuje technologie použité při implementaci aplikace a shrnuje aktuální stav problematiky mapových serverů.

1.1. Jazyk HTML5

Následující kapitola předpokládá u čtenáře základní znalost syntaxe a sémantiky jazyka HTML a jeho historie.

1.1.1. Specifikace jazyka HTML5

Pro verzi 5 jazyka HTML (dále jen HTML5) není v době psaní této práce dosud hotová specifikace [1]. Jak uvádí publikace HTML5 Programujeme moderní webové aplikace [3][s. 21] po verzi HTML 4.01 [4], která byla uvedena v roce 1999, byl jazyk HTML považován spíše za slepou uličku, což vedlo k přesunutí pozornosti směrem k jazykům XML a XHTML. O vznik první verze specifikace HTML5 se zasadila v roce 2004 malá skupina WHATWG (Web Hypertext Application Working Group).

„Zahájili práce na nových funkcích specificky cílených na webové aplikace – na oblast, která podle nich strádala nejvíce. Zhruba v té době se objevil termín Web 2.0. A skutečně to bylo jako nový web, když statické weby uvolnily místo těm více dynamickým a sociálním webům.“ [3][s. 21]

O dva roky později se do vývoje opět zapojilo konsorcium W3C a v roce 2008 zveřejnilo první pracovní návrh HTML5. O rok později se objevil i první návrh XHTML 2 [9]. Podle plánu zveřejněného konsorciem W3C [5] by měla být schválena specifikace HTML5 v posledním čtvrtletí roku 2014. V roce 2016 bychom se pak mohli dočkat verze HTML 5.1.

1.1.2. Nová vize

Jazyk HTML5 je spíše evolucí než revolucí [3]. Internet je plný obsahu různé datace, a tedy není možné přestat podporovat stávající standardy. Pokud tedy nejsou z nějakého důvodu funkce HTML5 dostupné, musí dojít k degradaci chování.

Specifikace HTML5 jednoznačně upřednostňuje uživatele internetu před autory či programátory, tvůrci specifikace a čistotou kódu. Uvažme následující kódy: `class="header"`, `class=header`, `CLASS="header"`. Všechny zápisy jsou v HTML5 platné a uživatelům se zobrazují stejným způsobem. Zda je tato cesta správná by se dalo polemizovat, ale většina uživatelů internetu na implementaci webové aplikace hledí jako na černou skříňku. Součástí vize je také zjednodušení syntaxe, kterou si představíme v další kapitole.

Přestože HTML5 již od počátku klade důraz na bezpečnost, o čemž svědčí i přítomnost částí specifikace věnované úvahám o zabezpečení, odborníci však poukazují na jistá bezpečnostní rizika spojená s novými technologiemi nebo například podporou dříve zakázané cross-domain komunikace [7].

Novou vizí jazyka je také podpora standardu WAI (Web Accessibility Initiative), který zvyšuje přístupnost webu uživatelům s postižením, a nezávislost jazyka na médiu. HTML5 by měl fungovat napříč všemi zařízeními a platformami.

1.1.3. Přehled nových vlastností jazyka

Jak se lze dočíst na webových stránkách konsorcia W3C [6], HTML5 přináší syntaktické změny oproti HTML 4.1. Dochází ke zjednodušení zápisu tagů v hlavičce stránky. HTML syntaxe vyžaduje, aby byl specifikován element doctype k zajištění zobrazení dokumentu v prohlížeči v standardizovaném módu. Element doctype lze zapsat namísto dlouhého zápisu používaného v předchozích verzích: `<!doctype html public "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd" >`, pouze zkráceně `<!doctype html>`. Odkaz na DTD (Document Type Definition) již není vyžadován. Pro definici kódování lze využít zkrácený zápis `<meta charset="UTF-8" >` ekvivalentní k původnímu a stále platnému zápisu `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >`.

Množina elementů jazyka je rozšířena o nové prvky. Pro lepší sémantiku dokumentu jsou zavedeny elementy `section`, `article`, `aside`, `header`, `time` aj. Pro zobrazení multimediálního obsahu existují elementy `video` a `audio`.

Nový element `canvas` je využíván pro vykreslování grafiky.

Formulářový element `input` byl rozšířen o nové typy. Prohlížeče mohou na jejich základě přizpůsobit uživatelské rozhraní. Jako příklad uveďme typ `time`, kde může uživatelům prohlížeč nabídnout po kliknutí do pole kalendář. Dochází také k odstranění méně využívaných elementů a atributů.

Zároveň s novými značkami přichází další technologie (geolokace, Web Storage, Web Sockets, Web Workers), které stojí někde na rozhraní mezi prohlížečem a JavaScriptem. Geolokační rozhraní [8] představuje jednoduchou službu pro různé aplikace, pracující s aktuální polohou svých uživatelů. O způsobech lokalizace uživatelů se lze dočíst například na webu Programujte.com [10]. Technologie Web Storage [11] umožňuje ukládat data na straně klienta v uložištích, které

lze připodobnit k *asociativním polím*. Uložiště na straně klienta umožňuje provozování offline aplikací¹. Pomocí Web Sockets můžeme napsat webovou aplikaci, která může v roli klienta navázat obousměrné spojení se serverem a vyměňovat si informace v reálném čase.

Při provádění skriptů na HTML stránce dochází k „zamrznutí“, dokud není skript dokončen. Jazyk HTML5 přináší řešení v podobě objektů Web Workers [12], které si lze představit jako JavaScriptový kód běžící na pozadí bez vlivu na výkon webové stránky.

V jazyce lze při vývoji webových aplikací využít i nová API, a to s novými i stávajícími elementy. Například výše zmíněné elementy `video` a `audio` mají API pro ovládání playbacku, synchronizaci s jinými elementy a podporu titulků. Konsorcium W3C provedlo změny i v existujících API [6][oddíl 5.2]. Drobné změny zaznamenáváme v Document Object Modelu. Pro kompletní seznam změn, doporučuji navštívit web konsorcia W3C [6].

1.1.4. Element canvas

V předchozí kapitole byl zmíněn element `canvas` (v překladu plátno) bez důkladného popisu. V této části práce nastíním možnosti jeho využití pro zobrazení mapových dat.

Popis struktury elementu lze nalézt například na serveru Wikipedia [23]:

Prvek `canvas` sestává z regionu, na který lze kreslit, definovaném v HTML kódu šířkou a výškou. Po zadefinování k němu lze přistupovat pomocí JavaScriptu za použití mnoha kreslících funkcí podobných jiným běžným 2D API. To umožňuje dynamicky generovat grafiku. Předpokládané využití (opakují, že celé HTML5 stále není schváleno jako standard) je v oblasti vykreslování grafů, animací, her (i 3D) a úpravy obrázků.

Pro demonstraci si uveďme jednoduchý příklad, kdy na plátně nakreslíme cestu tvořenou 3 souřadnicemi.

```
...
<canvas id="myCanvas" width="300" height="300"></canvas>

<script>

//ziskani elementu canvas a jeho 2d kontextu
var canvas = document.getElementById('myCanvas');
var ctx = canvas.getContext('2d');
```

¹Offline aplikace - je označení pro aplikace, které se nainstalují do PC a je možné je využívat, i když není aktivní připojení k internetu.

```

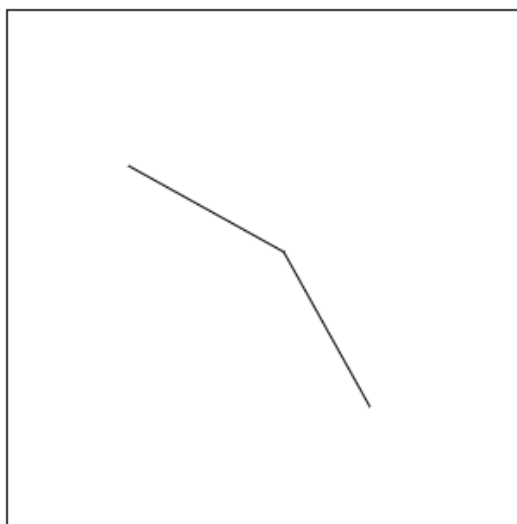
//vytvoreni cesty v absolutnich souradnicich
ctx.beginPath();
ctx.moveTo(70,90);
ctx.lineTo(160,140);
ctx.lineTo(210,230);

//vytazeni cesty
ctx.stroke();

</script>
...

```

Výsledkem kódu je následující cesta zakreslená na plátno.



Obrázek 1. Cesta na plátně tvořená 3 souřadnicemi.

Výše uvedený kód je součástí HTML dokumentu. V JavaScriptovovém kódu nejprve nalezneme objekt `canvas` přes atribut `ID` a uložíme jej do proměnné `canvas`. Metoda `getContext` s argumentem `'2d'` vrací 2D kontext daného plátna. Pro každý `canvas` můžeme inicializovat pouze jeden kontext. Na tomto místě můžeme zmínit, že v případě použití volání metody s argumentem `'webgl'` je návratovou hodnotou 3D kontext, který lze pak prakticky využít pro zobrazení trojrozměrných objektů.

JavaScriptové API nabízí několik metod pro vykreslování cest, geometrických tvarů, textu a vložení obrázků. V našem příkladu nejprve využijeme metodu `beginPath()` pro signalizaci začátku cesty. Pomyslné pero nejprve přesuneme na požadovanou počáteční souřadnici zavoláním metody `moveTo()`. Metoda `lineTo()` přidává nový bod a vede do něj přímkou z předcházejícího bodu na plátně. Tato metoda však nekreslí viditelnou linku. O tuto funkci se stará

poslední použitá metoda `stroke()`. Před použitím metody `stroke()` lze například také nastavit styl vykreslení pomocí atributů objektu `context`. Například příkaz `context.lineWidth = 1`; nastaví šířku linky na 1 px. Kompletní seznam dostupných atributů je dostupný na stránkách konsorcia W3C [18].

Pro zobrazení mapy je funkcionalita JavaScriptového API dostačující. Další výhodou plátna je jeho průhlednost. Elementy `canvas` lze na sebe vrstvit a použít například jednu množinu pláten pro cesty a jinou například pro zobrazení budov.

1.1.5. Podpora ve webových prohlížečích

Přestože není specifikace jazyka HTML5 hotová, některé její části jsou již poměrně stabilní a podporuje je většina moderních webových prohlížečů. Podpora méně stabilních funkcí jazyka se v různých webových prohlížečích liší. Z nejpoužívanějších renderovacích jader prohlížečů [13] (Trident, Gecko, WebKit a Presto) má jednoznačně nejhorší podporu engine Trident, který využívá rodina prohlížečů Internet Explorer od společnosti Microsoft. Detailní přehled podpory jednotlivých funkcí HTML5 lze nalézt například na serveru fmbip.com [14]. Pro účely této práce nás bude zajímat jen podpora elementu `canvas`, který byl vybrán pro zobrazení dat, a to jak na desktopových, tak mobilních webových prohlížečích. Na operačních systémech OS X a Windows je element podporován ve všech nejnovějších verzích prohlížečů Chrome, Firefox, Opera a Safari. Internet Explorer podporuje tag od verze 9 a výše [14]. Mobilní webový prohlížeč Opera Mini 5.0 - 7.0 podporuje zobrazení obsahu elementu `canvas`, ale nepodporuje přehrávání animací nebo běh složitějších aplikací využívajících plátno. Nejnovější verze prohlížečů iOS Safari (7.1), Android (4.4), Blackberry (10.0) a IE Mobile (10.0) plně podporují základní funkce tohoto elementu na mobilních zařízeních [17].

1.2. Prohlížeče map

1.2.1. Přehled současných řešení

Současné prohlížeče map využívají několik přístupů pro zobrazení geografických dat. První skupina využívá tzv. *dlaždice*, což je vícevrstvá digitální rastrová mapa čtvercového rozměru vygenerovaná z databáze vektorových dat. Každé měřítko má jinou sadu *dlaždic* s jinou úrovní detailů. Rozdělení na malé *dlaždice* výrazným způsobem urychluje zobrazování dat a přechod mezi přednastavenými měřítky. Stejný princip používají např. Google Maps², Bing Maps³, nebo ESRI ArcGIS online⁴. Pro rastrovou grafiku se na Internetu využívají formáty GIF, PNG a JPEG [15].

²Aplikace Google Maps dostupná z: <https://www.google.com/maps/preview>.

³Aplikace Bing Maps dostupná z: <http://www.bing.com/maps/>.

⁴Aplikace ESRI ArcGIS online dostupná z: <http://www.esri.com/software/arcgis/take-a-look-at-web-maps>.

Druhou skupinu tvoří prohlížeče zobrazující jednotlivé objekty ve vektorové grafice. Server na základě požadavků klienta připraví množinu objektů v podobě vektorových dat, provede případný ořez a zašle klientovi. Tento způsob otevírá možnosti interakce s mapou, kdy je možné jednotlivé objekty po najetí myši vysvětlit nebo zobrazit dodatečná data po kliknutí. Pro zobrazení se nejčastěji využívá element `canvas` nebo formát SVG [16]. HTML5 umožňuje vložit kód SVG obrázku přímo do HTML dokumentu prostřednictvím tagu `svg`.

V poslední době se začínají rozmáhat 3D mapy. Nejznámějším virtuálním glóblem je aplikace Google Earth⁵ dostupná v desktopové i webové verzi. Za zmínku stojí, že nejznámější český mapový server Mapy.cz⁶ společnosti Seznam nedávno oznámil vývoj trojrozměrných map České Republiky [19]. Pro zobrazení 3D grafiky ve webovém prohlížeči je možné využít technologii WebGL v kombinaci s elementem `canvas` (prohlížeče 3D map jako Google Maps, Here⁷ či JavaScriptová knihovna Cesium⁸ tuto technologii již prakticky používají).

V následujících odstavcích si důkladněji popíšeme některé prohlížeče map využívající HTML5, které mě inspirovaly při vývoji vlastního mapového prohlížeče.

1.2.2. Google Maps

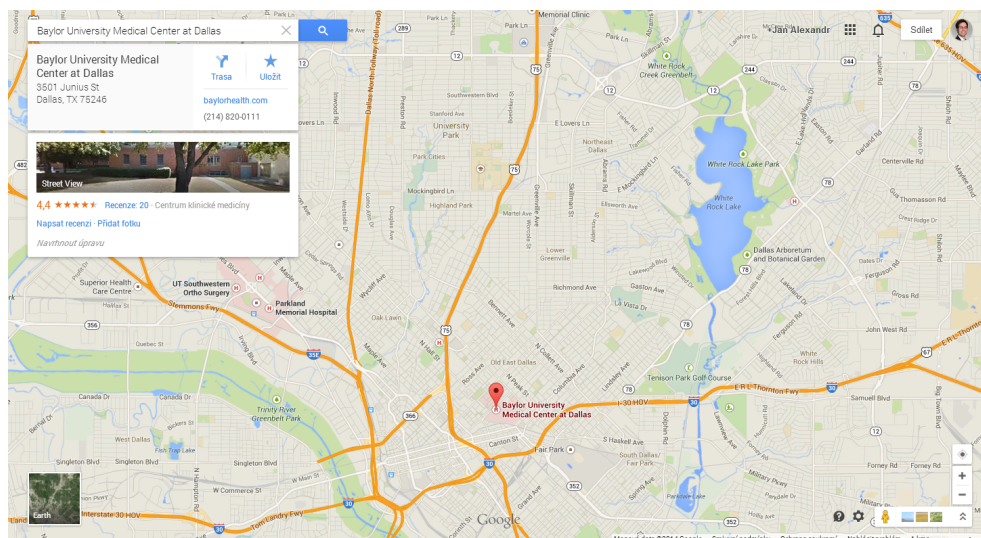
Nejznámější prohlížeč map od společnosti Google po poslední výrazné aktualizaci v roce 2013 [24] využívá jednoho elementu `canvas` pro zobrazení mapových dat. Z dostupných informací pro vývojáře na adrese `developers.google.com` spíše vyplývá, že mapa je stále tvořena mozaikou rastrových *dlaždic*. Nad ní je ještě několik vrstev [25], do kterých lze vykreslovat doplňující data jako textové popisky míst, symboly či vektorové objekty. Google Maps využívají nově také geolokační API HTML5 pro určení aktuální polohy uživatele.

⁵Aplikace Google Earth dostupná z: <http://www.google.cz/intl/cs/earth/>.

⁶Aplikace Mapy.cz dostupná z: <http://www.mapy.cz/>.

⁷Aplikace Here dostupná z: <http://www.here.com/>.

⁸Knihovna Cesium dostupná z: <http://cesiumjs.org/>.



Obrázek 2. Kontextové menu v prostředí Google Maps

Mapová data pocházejí od společnosti Tele Atlas, která má se společností Google uzavřeno partnerství. Aplikace Google Maps spolupracuje například s aplikací Google Earth. Obě služby využívají navzájem svá data pro koordinaci satelitních snímků a aktualizaci nově zmapovaných oblastí.

V aplikaci Google Maps nelze provádět interakce se všemi objekty na mapě. Po kliknutí například na ikony restaurací, parků, letišť, nemocnic apod. je zobrazeno kontextové menu (vyobrazeno na obrázku 2.) pod vyhledávacím polem a označený objekt je zvýrazněn ikonou umístění. Po kliknutí na libovolné místo je v kontextovém menu uveden alespoň dostupný panoramatický snímek aplikace Street View⁹.

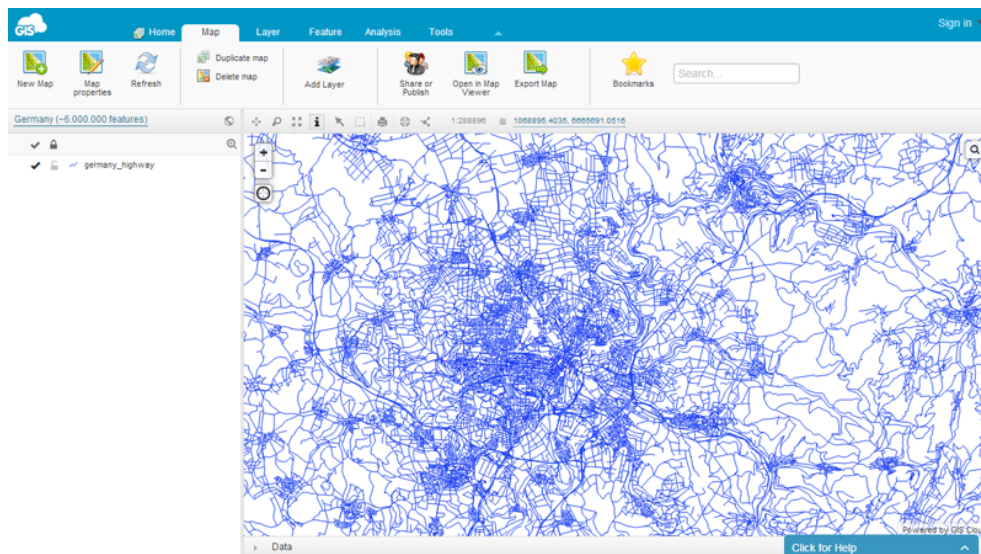
1.2.3. GIS Cloud

GIS Cloud¹⁰ je světově první plnohodnotný webový geografický informační systém (GIS) využívající technologii Cloud Computingu¹¹ s pokročilými schopnostmi vytváření, editace, nahrávání, sdílení, publikování a zpracování a analýzou geoprostorových a atributových dat. Jedná se také o první HTML5 vektorový mapový engine využívající *vektorové dlaždice* (zde dlaždicemi myslíme kolekci vektorových dat v dané lokalitě určené hranicemi dlaždice), které byly původně generovány ve Flash formátu a s nástupem HTML5 adaptovány na využití elementu *canvas*.

⁹Aplikace Street View dostupná z: <https://www.google.com/maps/views/streetview?gl=czhl=cs>

¹⁰Aplikace GIS Cloud dostupná z: <http://www.giscloud.com/>

¹¹Cloud Computing - je na Internetu založený model vývoje poskytující služby či programy uložené na serverech, ke kterým lze přistupovat pomocí webového prohlížeče nebo klienta dané aplikace.



Obrázek 3. Prostředí GIS Cloud

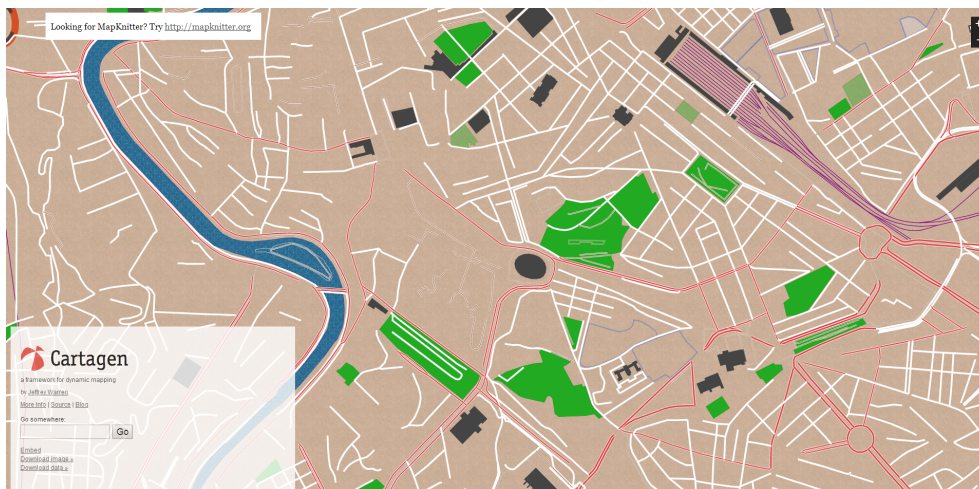
Ústředním prvkem systému je serverová komponenta, jež rychle a efektivně čte geometrii a atributy dat. Následně je zasílá na stranu klienta v podobě optimalizovaných vektorových *dlaždic*. Pro vizualizaci dat je využita knihovna Leaflet¹², která dodává plynulé animace pro přibližování a oddalování mapy, optimalizuje zobrazení pro mobilní zařízení s vysokým rozlišením, CSS3 vyskakovací okna a dialogy apod. Interakce lze provádět s každým vykresleným objektem.

1.2.4. Cartagen

Cartagen¹³ je JavaScriptový framework na straně klienta pro renderování vektorových map v HTML5. Mapy jsou stylovány pomocí Geographic Style Sheets (GSS), kde jsou jednotlivé deklaráce stylů zapsány jako hodnoty atributu JavaScriptového objektu. Jména atributů vychází z názvosloví OpenStreetMap, jelikož framework využívá mapových dat z tohoto zdroje. Data jsou stejně jako u předchozích prohlížečů zobrazena do elementu `canvas`.

¹²Knihovna Leaflet dostupná z: <http://leafletjs.com/>.

¹³Aplikace Cartagen dostupná z: <http://cartagen.org/>



Obrázek 4. Prostředí prohlížeče Cartagen.

1.3. OpenStreetMaps

OpenStreetMaps¹⁴ (dále jen OSM) je projekt, který vytváří a distribuuje volně dostupná geografická data světa na koncepci Open source¹⁵ a v praktické části projektu byl využit jako zdroj mapových dat. OSM vytváří a sbírá polohopisná data pro celý svět. Pro zmapování se využívá dat, které uživatelé sesbírali v terénu pomocí ručních GPS přijímačů, satelitních snímků a informací z veřejných či komerčních zdrojů mapových dat jako TIGER, SRTM nebo GTOPO. V rámci mapování České republiky v OSM velké množství dat poskytl přímo Český úřad zeměměřický a katastrální (ČÚZK). Přístup k prostorovým datům pořizovaným a aktualizovaným v rezortu tohoto úřadu je možné na Geoportálu ČÚZK¹⁶.

K ukládání dat do databáze slouží editory. Jelikož OSM je otevřená platforma s editačním API, existuje spousta editorů, které jsou vyvíjeny nezávisle. Z důvodů, že editace map nespadá do rozsahu této práce, nebudeme se možností editace a tvorby map dále zabývat. Seznam dostupných editorů je k dispozici na OpenStreetMap Wiki [20].

1.3.1. Prohlížeč map OpenStreetMap.org

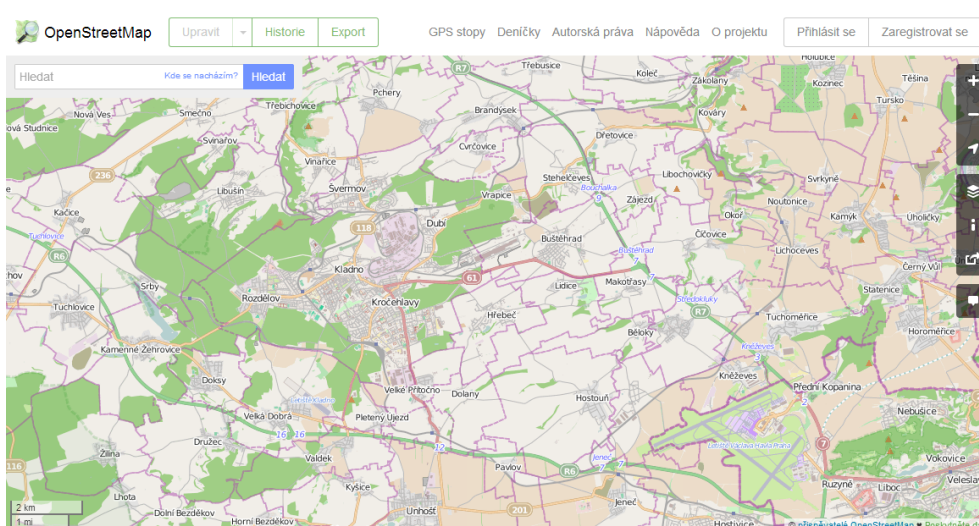
Mimo editorů a API nabízí OSM i možnost zobrazit si mapová data přes [webový prohlížeč](#) či vkládat mapy do webových stránek. Používá JavaScripto-

¹⁴Projekt OpenStreetMaps dostupný z: <http://www.openstreetmap.org/>.

¹⁵Jak uvádí server Wikipedia.org, Open source (v překladu otevřený software) je počítačový software s otevřeným zdrojovým kódem. Otevřenost zde znamená jak technickou dostupnost kódu, tak legální dostupnost - licenci software, která umožňuje, při dodržení jistých podmínek, uživatelům zdrojový kód využívat, například prohlížet a upravovat.

¹⁶Geoportál ČÚZK dostupný z: <http://geoportál.cuzk.cz/>.

vou knihovnu OpenLayers¹⁷ pro zobrazení mapových dat ve většině moderních webových prohlížečů. Obsahuje dva různé pohledy na mapu, renderované pomocí systémů Mapnik¹⁸ a Osmarender¹⁹. Ty periodicky nebo na požádání překreslují mozaiku *dlaždic* z aktuálních dat. Prohlížeč map OSM byl v praktické části využit především při kontrole správnosti zobrazení map a k identifikaci základních funkcí prohlížeče při specifikaci požadavků.



Obrázek 5. Prohlížeč OSM

1.3.2. Formát dat

Projekt OSM vyvíjí vlastní souborový formát (s příponou .osm) pro vektorová geodata postavený na XML. Databáze obsahuje elementy následujících typů:

1. Uzly - zeměpisné souřadnice (body) nebo kolekce bodů (vícenásobné body)
2. Cesty - sekvence spojených zeměpisných souřadnic (čáry) nebo kolekce čar (vícenásobné čáry) a uzavřené sekvence zeměpisných souřadnic (mnohoúhelníky) nebo kolekce mnohoúhelníků (vícenásobné mnohoúhelníky).
3. Relace - skupina uzlů, cest a dalších relací.
4. Atributy - nejedná se o samostatný element, ale o vlastnosti uzlů, cest a relací ve formě klíč = hodnota. Atributy rozlišují například zda daná cesta je řeka či dálnice.

¹⁷Knihovna OpenLayers dostupná z: <http://openlayers.org/>.

¹⁸Systém Mapnik dostupný z: <http://mapnik.org/>.

¹⁹Nástroj Osmarender popsán na: <http://wiki.openstreetmap.org/wiki/Osmarender>.

1.3.3. OSM Extended API

Součástí projektu OSM je také OSM Extended API [21] (dále jen XAPI). XAPI je read-only API protokol založený na modifikované verzi hlavního OSM API [22]. Poskytuje rozšířené vyhledávací a dotazovací funkce. Existuje několik nezávislých implementací XAPI (jXAPI, Overpass API a Xappy.js).

Dotaz na mapový server má tvar:

```
GET /api/0.6/map?bbox=<left>,<bottom>,<right>,<top>
```

a vrací všechny uzly, cesty, relace a atributy uvnitř oblasti ohraničené hodnotami `left`, `bottom`, `right`, `top`, kde:

- `left` reprezentuje zeměpisnou délku západní hranice oblasti (někdy udávané jako `minlon`).
- `bottom` reprezentuje zeměpisnou šířku jižní hranice oblasti (někdy udávané jako `minlat`).
- `right` reprezentuje zeměpisnou délku východní hranice oblasti (někdy udávané jako `maxlon`).
- `top` reprezentuje zeměpisnou šířku severní hranice oblasti (někdy udávané jako `maxlat`).

Ukázkovým dotazem by mohl být: `http://xapi.openstreetmap.org/api/0.6/map?bbox=11.54,48.14,11.543,48.145`. Pokud bychom chtěli ze serveru získat pouze uzly, mohli bychom zavolat adresu: `http://xapi.openstreetmap.org/api/0.6/node[bbox=11.54,48.14,11.543,48.145]`. Analogicky bychom mohli vytvořit dotazy pro cesty a relace. Při implementaci vlastního prohlížeče map využijeme téměř všechna data, proto si vystačíme s tvarem prvního ukázkového dotazu.

Výsledkem dotazu je XML dokument v následujícím tvaru, který obsahuje elementy splňující vyhledávací podmínku.

```
<?xml version='1.0' standalone='no'?>
<osm version='0.6' generator='xapi: OSM Extended API'
  xmlns:xapi='http://www.informationfreeway.org/xapi/0.6'
  xapi:uri='/api/0.6/*[amenity=hotel]'
  xapi:planetDate='200803150826'
  xapi:copyright='2008 OpenStreetMap contributors'
  xapi:instance='zappy2'>
  <node id='218963' lat='52.5611324692581' lon
    ='-1.79024812573334' timestamp='2006-03-22T16:47:48+00:00'
    version='1' changeset='2211'>
  </node>
```

```

<node id='331193' lat='53.7091237972264' lon
    ='-1.50282510180841' timestamp='2007-03-31T00:09:22+01:00'
    version='1' changeset='2211'>
  <tag k='amenity' v='hotel' />
</node>
...
<way id='4958218' timestamp='2007-07-25T01:55:35+01:00' version
    ='1' changeset='2211'>
  <nd ref='218963' />
  <nd ref='331193' />
  ...
  <tag k='amenity' v='hotel' />
  <tag k='building' v='hotel' />
</way>
<relation id='123456' timestamp='2007-10-25T03:05:34Z' version
    ='32' changeset='2211'>
  <member type='node' ref='331193' role='' />
  <member type='node' ref='331194' role='' />
  ...
  <tag k='amenity' v='hotel' />
  <tag k='operator' v='Premier Inns' />
  <tag k='type' v='operators' />
</relation>
</osm>

```

Pro zobrazení jsou v případě uzlů (element `node`) důležité následující atributy: `id` pro identifikaci souřadnice, `lat` a `lon` pro umístění uzlu do mapy. Každý uzel může mít žádný, jeden nebo více potomků typu `tag`, které budou sloužit především pro určení stylu zobrazení na mapě. Každý tag je určen dvojicí atributů `k` jako klíč (key) a `v` jako hodnota (value).

Struktura uvozená elementem `way` popisuje cestu. Mimo atributů sloužících především k identifikaci, může obsahovat 2 a více potomků typu `nd` s jediným atributem obsahující hodnotu ID uzlu, jenž je součástí cesty. Analogicky k uzlu může cesta obsahovat potomky typu `tag`.

Relaci popisuje element `relation`. Nachází se v něm několik členů relace (elementů `member`), kteří jsou definovány typem objektu, identifikátorem ID a svou rolí. Pro související informace relace jsou k dispozici opět elementy typu `tag`.

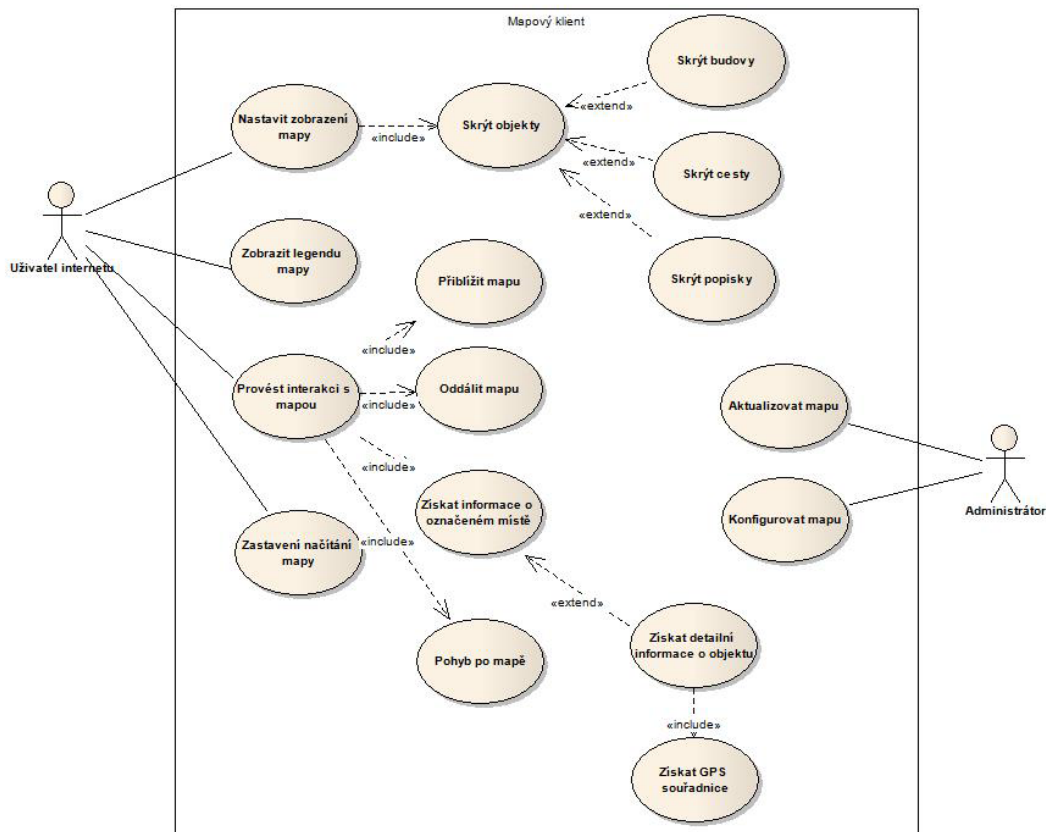
2. Analýza problému a návrh řešení

2.1. Cíle projektu

- Primárním cílem práce je prozkoumat možnosti technologie HTML5, která by v této oblasti mohla nalézt uplatnění.
- Vytvořit aplikaci, která bude zobrazovat mapová data OpenStreetMap v rozsahu České republiky a související informace v prohlížeči s využitím možností HTML5.
- Poskytnout běžnou funkcionalitu pro prohlížeče map.
- Vytvořit aplikaci, která funguje ve většině současných prohlížečů (podporujících HTML5), včetně prohlížečů na mobilních zařízeních.

2.2. Specifikace požadavků

Ze zadání projektu (uvedeném v příloze A) a jeho cílů vyplynulo několik funkčních a mimofunkčních požadavků na systém. Uživatelské požadavky na program zobrazuje diagram případů užití na obrázku 6.



Obrázek 6. Use Case Diagram (Diagram případů užití).

2.2.1. Funkční požadavky

- Uživatel by měl mít možnost přibližovat, oddalovat mapu a pohybovat s mapou.
- Uživatel by měl mít možnost nastavit zobrazení mapy jako je skrytí a zobrazení jednotlivých vrstev.
- Uživatel by měl mít možnost zastavit načítání mapy, pokud trvá příliš dlouho.
- Uživatel by měl mít možnost zobrazit související data o jednotlivých mapových objektech po kliknutí na daný objekt. Souvisejícími daty rozumíme GPS souřadnice a atributy objektů v databázi OSM.
- Administrátor by měl mít možnost aktualizovat mapová data z databáze OSM.
- Administrátor by měl mít možnost konfigurovat prohlížeč (například nastavit, které objekty jsou viditelné při daném přiblížení).

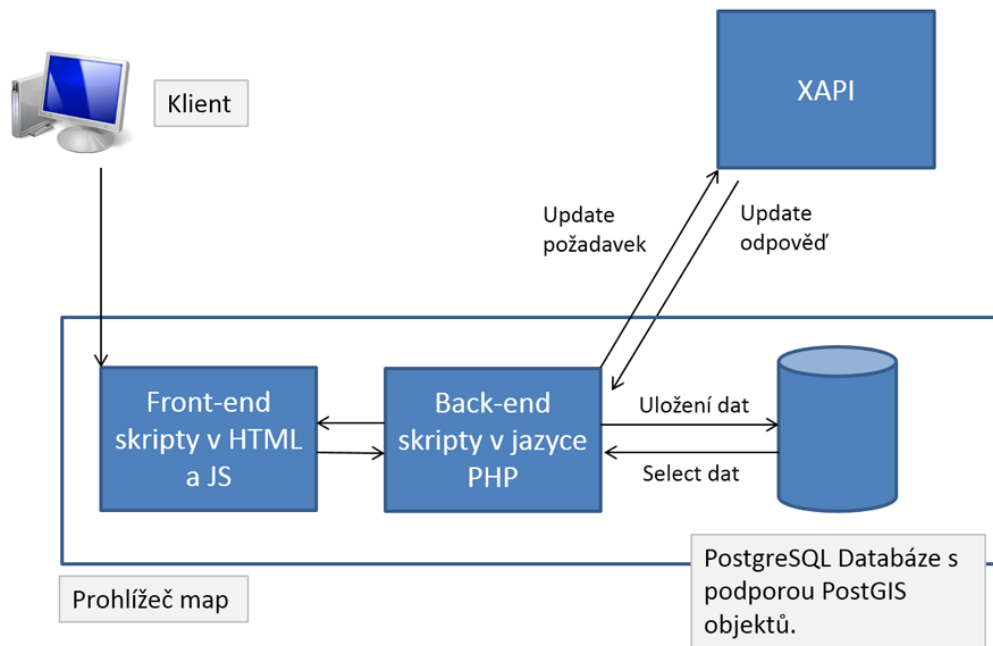
2.2.2. Mimofunkční požadavky

- Zobrazení mapy by nemělo trvat déle než 10 sekund.
- Program by měl fungovat ve většině soudobých prohlížečů podporujících HTML5 a na většině zařízení včetně přenosných zařízení (verze webových prohlížečů uvedené v kapitole 1.1.5.).

2.3. Koncepce programu

Pro implementaci serverové části programu byl vybrán jazyk PHP pro svou specializaci na webové stránky a nativní podpory mnoha databázových systémů. Klientská část je napsaná v jazycích HTML5, CSS3 a JavaScript. Koncepce programu se spolu s implementací jednotlivých částí vyvíjela a byla přizpůsobována novým požadavkům.

Zdrojem zobrazovaných geografických dat je projekt OpenStreetMaps, který jsem představil v kapitole 1.3. Data jsou ukládány do databáze PostgreSQL [33] s podporou rozšíření PostGIS [34], umožňující pracovat s geografickými objekty. Databáze plní úlohu cache paměti. Data jsou získávána přes XAPI ve formátu XML a proudově zpracována přes Simple API for XML (SAX) [31]. Důvodem tohoto způsobu zpracování je především velikost XML souborů (pro představu mapová data světa obsahují téměř 2,5 mld. bodů, 250 mil. cest a 2 mil. relací [26] a celková velikost je 400GB nekomprimovaných XML dat a 29GB komprimovaných dat [27]). Volání aktualizace mapových dat probíhá buď spuštěním skriptu manuálně nebo automaticky v intervalech přes CRON skript. Data jsou vykreslována do elementů `canvas`, které jsou na sebe skládány ve vrstvách. Množina vykreslovaných objektů do jednotlivých vrstev lze nastavit v konfiguračním souboru. Počet vrstev je pevně dán. Koncepce je znázorněna na obrázku č. 7.



Obrázek 7. Diagram znázorňující architekturu aplikace

Konkrétní způsob implementace tohoto konceptu je představen v následující kapitole.

3. Implementace mapového prohlížeče

Cílem této kapitoly je detailně popsat postupy a způsob realizace mapového prohlížeče. Implementace vychází z kapitoly 2.3. popisující koncepci programu.

3.1. Proces vývoje

Úvodní fázi projektu vystihuje pojem *investigace*, tedy soustavné zkoumání, vyhledávání a sbírání informací [28]. Zdrojem pro pochopení struktury a významu mapových dat byla dokumentace OpenStreetMaps Wiki²⁰. K pochopení technologie HTML5 jsem využíval knihu HTML5 Programujeme moderní webové aplikace [3] a dostupné online zdroje uvedené na závěr tohoto textu. Některé klíčové poznatky byly shrnuty v úvodní kapitole. Vývoj probíhal iterativně. Základní funkčnost se postupně rozšiřovala a zdokonalovala.

Na začátku se XML data parsovala přímo z odpovědi XAPI a zobrazovala na straně klienta do jednoho elementu `canvas`. Mezikrokem bylo načítání dat z uložených XML dokumentů exportovaných z OSM. Přítomnost databáze se s rostoucím množstvím zobrazených dat ukázala jako nutnost, jelikož čekání na odpověď XAPI či parsování celých XML souborů trvalo déle než bylo přijatelné. Následovalo zprovoznění objektově-relačního databázového systému PostgreSQL, instalace modulu PostGIS a vytvoření vhodného databázového schématu. Byly naprogramovány první serverové skripty zajišťující vkládání a selektování dat. Databáze zaujala pozici cache paměti mezi klientskými skripty a XAPI. Více o způsobech získávání a zobrazování dat se lze dočíst v kapitole 3.5.

Paralelně s vývojem serverových skriptů byla vytvářena i klientská část. Její úlohou je plynulé zobrazování dat a provádění základních operací s mapou jako je posun, přibližování, interakce a zobrazování souvisejících dat. Při implementaci posunu po mapě bylo předpokládáno, že načítání mapy po menších částech bude rychlejší a nebude nutné mapu neustále překreslovat, jelikož zobrazené elementy `canvas` bude stačit vykreslit pouze jednou. Z toho důvodu došlo k rozdělení jednoho plátna na více elementů představujících jednotlivé *dlaždice*. Elementy `canvas` byly umístěny přes sebe do několika vrstev, aby bylo možné skrývání a odkrývání jednotlivých kategorií mapových objektů. Následoval vývoj dalších funkcí mapy jako oddalování a přibližování. Byl vytvořen konfigurační soubor pro nastavení detailů jednotlivých vrstev a rychlý přístup k základnímu nastavení mapového prohlížeče. Stěžejní funkcí klientské části je možnost získat po kliknutí detailnější informace o viditelných objektech na mapě. Základní parametry jednotlivých zobrazených objektů, s nimiž lze provádět interakce, jsou uložena v poli na straně klienta. Dodatečné informace jsou po kliknutí získány asynchronním dotazem do databáze. S rostoucím počtem zobrazených objektů na mapě roste i doba pro identifikaci objektu a načtení dat, proto je vhodné povolit

²⁰Stránka OpenStreetMaps Wiki dostupná z: http://wiki.openstreetmap.org/wiki/Main_Page.

interakcí jen ve 3 úrovních s nejvyšším přiblížením (10, 11, 12).

Závěrečná fáze projektu zahrnovala psaní dokumentace v podobě tohoto textu, testování na vybraných zařízeních a webových prohlížečích a nahrávání většího množství mapových dat do databáze. V průběhu vývoje bylo zjištěno několik omezení, které byly průběžně zaznamenány a jsou shrnuty v kapitole 4. věnované závěru práce.

3.2. Klientské skripty

Za klientské skripty jsou považovány všechny JavaScriptové soubory, které se vykonávají na straně klienta. Soubor *common-functions.js* obsahuje všechny funkce a procedury související s inicializací mapy, načítáním, stylováním a vykreslováním dat do elementů `canvas`, pohybu, přibližování a oddalování mapy a detekce mobilních zařízení. Většina kódu využívá JavaScriptovou knihovnu jQuery verze 1.10.2 (soubor *jquery-1.10.2.min.js*). Po načtení HTML dokumentu jsou nejprve deklarovány globální proměnné sloužící pro přístup k zeměpisným souřadnicím získaných z URL adresy a z nich vypočítaných souvisejících hodnot. Po inicializaci proměnných jsou volány 3 inicializační procedury, které se starají o nastavení úvodních pozic elementů `canvas`, zobrazení mapy dle nastavených parametrů a nastavení obsluhy jednotlivých akcí prvků uživatelského rozhraní:

```
initializeMapProperties();  
initializeCanvases();  
setInteractions();
```

Pro posun po mapě je využit volně dostupný skript pod MIT licenci²¹ *Draggabilly* [29] (skript *draggabilly.pkgd.min.js*), jenž funguje na všech webových prohlížečích, které podporují také element `canvas`. Obsluhu událostí posunu představují procedury `onDragMoveOn`, `onDragMove` a `onDragMoveOff`. Procedury zajišťující obsluhu událostí posunutí a kliknutí myši a dotyku na dotykových zařízeních jsou `onMouseMove()`, `onMouseClick()` a `onTouchAction()`. Oddalování a přibližování mapy se děje bez nutnosti znovunačtení stránky.

Změna polohy je dynamicky zapsána do URL adresy a je přidán nový záznam do historie prohlížeče pomocí příkazu: `history.pushState('', 'Map Browser', newHref)`. Prohlížeč umožňuje celkem 12 úrovní přiblížení od 1 do 12 a parametry jednotlivých úrovní lze nastavit v konfiguračním souboru. Procedury zajišťující přiblížení a oddálení se jmenují `zoomIn()` a `zoomOut()`. Mapová data jsou z databáze získávána asynchronně v proceduře *loadObjects()* voláním skriptu *selectFromDB.php* s parametry udávajícími souřadnice mapové oblasti, úrovní

²¹MIT licence - software uvolněný pod touto licencí je možné použít jak v proprietárním software (s podmínkou, že text licence MIT musí být dodáván spolu s daným software), tak i s GPL licencovaným software (díky tomu, že GPL explicitně povoluje kombinaci s licencí MIT). Zdroj: Wikipedia.org

přiblížení a typem zobrazovaných objektů. Vykreslování cest a bodů do elementů `canvas`, umístování značek a popisků probíhá v procedurách `drawNode()` a `drawWay()`, kde jsou využity procedury JavaScriptového API popsané v kapitole 1.1.4. Mapa obsahuje také měřítko, které využívá data z konfiguračního souboru a procedura `setScale()` je formuje do grafické podoby. Popisky jsou na mapu umístovány jako HTML elementy a jejich pozice a rotace je nastavena pomocí CSS, díky čemuž lze v mapě hledat s pomocí základních vyhledávacích nástrojů webových prohlížečů.

Aby bylo možné nabídnout jiný kód na mobilních zařízeních, obsahuje skript *common-functions.js* jednoduchou detekci mobilních zařízení. Pokud informace o uživatelském agentovi obsahují specifický podřetězec, je v proměnné `isMobile` uložena hodnota `TRUE`.

Skript *jquery.ajaxq-0.0.1.js* je volně distribuován s licencí MIT a obsahuje jednoduchou proceduru rozšiřující prototyp objektu JQuery sloužící k vytvoření fronty ajaxových volání. Je využíván skriptem *cronScript.php*.

3.3. HTML a CSS dokumenty

Klientské skripty doplňují soubor *index.php* napsaný v jazyce HTML a stylpis *styles.css*. Zmiňme několik zajímavých aspektů těchto dokumentů.

Soubor *index.php* využívá zjednodušený zápis HTML5. Obsahuje základní HTML strukturu prohlížeče map. Aby bylo možné stránku zobrazit na mobilních zařízeních a použít techniky responzivního webdesignu, je v hlavičce stránky následující meta tag, který upravuje měřítko stránky.

```
<meta name=viewport content="width=device-width, initial-scale=1.0, minimum-scale=0.5 maximum-scale=1.0">
```

Stylpis *styles.css* používá CSS3 jednotku *rem* [30] pro velikost fontů relativní vůči kořenovému HTML elementu. Výhody jako spolehlivost, konzistence a snadné zvětšování či zmenšování fontů našly uplatnění ve funkci pro zvětšení fontů aplikace. Bylo předpokládáno, že velikost textů vhodná pro osobní počítače, nemusí být vhodná pro mobilní zařízení s vysokým rozlišením (především Retina display²²). Rem jednotky umožňují definovat jednu základní velikost fontu pro celou webovou stránku a všechny ostatní *rem* jednotky budou odvozeny přímo z tohoto rozměru. V předkládané aplikaci je základní rozměr fontů nastaven na hodnotu 62.5%, díky čemuž se hodnoty snadno převádějí z pixelů do *rem* jednotek. Ukázkou jsou následující CSS vlastnosti:

```
font-size: 12px;  
font-size: 1.2rem;
```

²²Více informací o Retina Display lze nalézt na http://en.wikipedia.org/wiki/Retina_Display.

První nastavení velikosti fontu v pixelech slouží jako fallback pro webové prohlížeče nepodporující jednotky *rem*. Více o jednotkách *rem*, jejich požití a podpoře se lze dočíst například na stránce [igloonet blog\[32\]](#).

3.4. Konfigurační soubor `config.php`

Soubor obsahuje jedinou proměnnou `$config`, v níž je definováno asociativní pole. Nejdůležitější klíče a hodnoty jsou popsány v následujícím seznamu. Konfigurace vychází částečně ze strukturu XML dokumentů získávaných z XAPI, které jsem popsal v kapitole 1.3.3..

- *host* - Hodnotou klíče je absolutní adresa adresáře se skripty aplikace.
- *interactionAvailable* - Obsahuje pole s čísly úrovní, na kterých jsou dostupné uživatelské interakce s objekty mapy.
- *defaults* - Obsahuje pole s hodnotami parametrů (*lon*, *lat* a *zoom*) v případě, že nejsou v URL adrese definovány. Tato konfigurace je využívána při spuštění webové aplikace pouze se znalostí adresy či domény. Uživatel nemusí znát strukturu URL a explicitně uvést parametry.
- *edges* - Hodnotou spojenou s tímto klíčem je pole obsahující mezní zeměpisné délky a šířky pro aktualizaci mapy.
- *dblogin* - Základní nastavení přístupu do databáze (obsahuje hodnoty *host*, databázové jméno, uživatelské jméno a heslo).
- *layers* - Obsahuje strukturu popisující viditelnost jednotlivých elementů na mapě. Jednotlivé vrstvy mapy jsou v konfiguračním souboru pojmenovány stejně jako hodnota atributu `objects` u elementu `<div class='layer'>` obsahující jednotlivé elementy `canvas`. V základní konfiguraci jsou vrstvy pojmenovány *background*, *ways*, *buildings* a *symbols*. Každá konfigurace vrstvy obsahuje vždy klíče *visible* a *objects*, které odkazují na pole. Pole pod klíčem *visible* obsahuje čísla úrovní přiblížení, ve kterých jsou objekty v poli *objects* zobrazené na mapě. Obvykle toto pole obsahuje čísla všech vrstev a zobrazení jednotlivých objektů se řeší na úrovni konfigurace objektů. Jelikož se objekty v databázi OSM popisují pomocí tagů s klíčem a hodnotou, konfigurace objektů je rozdělena na 2 úrovně. Ve vyšší úrovni je nastaveno zobrazení klíče a v další úrovni je nastaveno zobrazení hodnot. Konfigurace je tak flexibilní a lze jednoduše skrývat či zobrazovat objekty po skupinách či vrstvách. Uvedme si následující příklad:

```
'waterway' => array(  
    'visible' => array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,  
        12),
```

```

    'values' => array(
        'riverbank' => array(
            'visible' => array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
                10, 11, 12)
        ),
        'river' => array(
            'visible' => array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
                10, 11, 12)
        ),
        'stream' => array(
            'visible' => array(10, 11, 12)
        )
    )
),

```

Kód popisuje viditelnost objektů s tagem, jehož klíč je *waterway*. Pokud má tag hodnotu *riverbank* či *river*, je viditelný na všech úrovních přiblížení. Výčet vrstev, v nichž je objekt zobrazen, je opět uveden v poli s klíčem *visible*. Objekty s hodnotou tagu *stream* jsou do mapy vykreslené pouze v úrovních 10, 11 a 12. Pokud tato úroveň konfigurace obsahuje ještě další pole s klíčem *values*, lze v něm konkretizovat, zda má být objekt na mapě zobrazen, pokud má například tag s klíči *name* a zároveň tag s klíčem *place* a hodnotou *city*. Tento způsob je využit například při nastavení zobrazení popisků na mapě.

- *zoom* - Poslední klíč popisuje jednotlivé úrovně přiblížení. Lze specifikovat jak velké je přiblížení (klíče *lon100* a *lat100* obsahují zeměpisnou délku a zeměpisnou šířku popsanou na ploše 100 × 100 pixelů), základní šířku linky (klíč *line*) a tolerance zjednodušení geometrie (klíč *simplify*).

3.5. Serverové skripty

Většina serverových skriptů je volána asynchronně, klient tak nemusí čekat na jejich odpověď.

3.5.1. selectFromDB.php

Skript je volán, pokud je nutné vybrat objekty z databáze na základě parametrů v adrese, kterými jsou *minlat*, *minlon*, *maxlat*, *maxlon*, *zoom* a *layer*. Uveďme si příklad. Následující SQL dotaz byl vygenerován při volání tohoto skriptu s parametry `minlon=16.79296&minlat=49.03219&maxlat=49.15313&maxlon=16.97842&layer=background&zoom=6`.

```

SELECT ways.id, ways.visible, ways.nodes, ST_AsText( (ST_Dump(
  ST_Simplify(geom,0.0005)).geom ) AS geometry, tags_ways.key,
tags_ways.value
FROM (ways INNER JOIN (SELECT * FROM tags_ways WHERE key = '
  landuse' AND value = 'residential' OR key = 'landuse' AND
  value = 'forest' OR key = 'landuse' AND value = 'reservoir'
  OR key = 'leisure' AND value = 'park' OR key = 'place' AND
  value = 'city' OR key = 'waterway' AND value = 'riverbank'
  OR key = 'waterway' AND value = 'river' OR key = 'natural'
  AND value = 'water' OR key = 'area')AS tags_ways ON id =
tags_ways.way_id) WHERE ST_Intersects(ST_GeomFromText('
POLYGON((49.15313 16.79296,49.03219 16.79296,49.03219
16.97842,49.15313 16.97842,49.15313 16.79296))', 0 ), geom)

```

Vyhledáváme data z relační tabulky *ways* a řádků z relační tabulky *tags_ways*, které splňují podmínku `key = 'landuse' AND value = 'residential' OR key = 'landuse' AND value = 'forest' OR key = 'landuse' AND value = 'reservoir' OR key = 'leisure' AND value = 'park' OR key = 'place' AND value = 'city' OR key = 'waterway' AND value = 'riverbank' OR key = 'waterway' AND value = 'river' OR key = 'natural' AND value = 'water' OR key = 'area'`. Tyto 2 relace jsou spojeny vnitřním spojením. Aby bylo možné získat jen objekty ležící v určité oblasti na mapě, využívám funkci `ST_Intersects()` z rozšíření PostGIS, která přijímá jako argumenty 2 geometrie a vrací hodnotu `TRUE`, pokud se geometrie prostorově protínají (sdílejí libovolnou část prostoru), a `FALSE`, pokud tomu tak není. Použité funkce `ST_GeomFromText()` a `ST_AsText()` provádějí převod geometrie na text a textu na geometrie. Funkce `ST_Dump()` vypisuje jednotlivé geometrie složených geometrií a funkce `ST_Simplify()` vrací zjednodušenou verzi geometrie s použitím Douglas–Peuckerova algoritmu [35]. Tolerance použitá při zjednodušení je získána z konfiguračního souboru.

Výsledkem skriptu jsou pole nalezených objektů (cest a uzlů) ve formátu JSON s atributy, geometriemi a se všemi tagy. Uveďme si příklad cesty tvořené 11 uzly ve formátu JSON. Jak si lze povšimnout, geometrie byla zjednodušena funkcí `ST_Simplify()` na cestu tvořenou pouze 4 body.

```

{id": "90429957",
"visible": null,
"nodes": [
  "1049045693",
  "1049044990",
  "1049045841",
  "1049045320",
  "1049044728",
  "1049045627",

```

```

    "1049045523",
    "1049044538",
    "1049045400",
    "1049045138",
    "1049044574",
    "1049045484",
    "1049045693"
  ],
  "geometry": "POLYGON((49.1497577 16.8112791,49.1512832
    16.8114447,49.1524705 16.8132074,49.1526863
    16.8176941,49.1497577 16.8112791))",
  "tags": [
    {
      "key": "landuse",
      "value": "forest"
    }
  ]
]

```

3.5.2. selectInfo.php

Skript získává podobně jako skript selectFromDB.php informace z databáze, tentokrát však pouze o objektech udaných jejich identifikátory, které jsou skriptu předány v parametru *id*. Výsledkem je obdobný JSON kód uvedený v předchozím příkladu. Na straně klienta jsou tyto informace zobrazeny po označení objektu v kontextovém menu.

3.5.3. updateMap.php

Skript přijímá parametry *minlat*, *minlon*, *maxlat*, *maxlon* a s nimi volá XAPI. Získaný XML soubor zpracovává proudově využitím SAX a ukládá či aktualizuje data v databázi. Databáze PostgreSQL nepodporuje příkazy MERGE či UPSERT [40], které umožňují v jediném příkazu buď aktualizovat záznam, pokud v databázi existuje, nebo vložit nový záznam, pokud tomu tak není. Od verze PostgreSQL 9.1 je však možné toto chování simulovat za pomoci příkazu WITH [39]. Toto řešení bylo převzato z webu The Art OF Web [38] a funguje dle očekávání.

```

BEGIN;
LOCK TABLE nodes IN SHARE ROW EXCLUSIVE MODE;
WITH upsert AS (UPDATE nodes SET geom = ST_GeomFromText('POINT
  (49.7977173 15.8182971)', 26910) WHERE id = 284720912 RETURNING
  *) INSERT INTO nodes (id, geom) SELECT 284720912,
  ST_GeomFromText('POINT(49.7977173 15.8182971)', 26910) WHERE
  NOT EXISTS (SELECT * FROM upsert);
COMMIT;

```

Podobný formát dotazu je použit i pro aktualizaci cest a tagů.

3.5.4. indexes.php

Jediným parametrem skriptu je parametr *type*. V něm lze specifikovat, zda mají být databázové indexy vytvořeny (hodnota parametru `create`) nebo odstraněny (hodnota parametru je libovolná). Skript je volán ze souboru `cronScript.php` na začátku a na konci aktualizacího procesu.

3.5.5. cronScript.php

Obsahuje kód aplikačního procesu, který se skládá z několika fází:

1. Odstranění databázových indexů voláním skriptu `indexes.php?type=drop`.
2. Asynchronní volání skriptu `updateMap.php` s vypočítanými parametry.
3. Vytvoření databázových indexů voláním skriptu `indexes.php?type=create`.

Krok 2 se opakuje tak dlouho, dokud není aktualizovaná celá oblast mapy. Jak velkou oblast aktualizujeme lze uvést v konfiguračním souboru v poli s klíčem `edges`. Aby aktualizace pobíhala rychleji a bylo možné sledovat její postup, je nutné nastavit v proměnných `$count_x` a `$count_y` rozdělení mapy do částí v horizontálním a vertikálním směru. Výsledný počet částí mapy je získán vynásobením obou proměnných a využit při výpočtu parametrů v kroku 2.

Skript lze spustit jak přímo z webového prohlížeče administrátorem, tak softwarovým démonem CRON [36]. Zobrazení ve webovém prohlížeči nabízí uživatelské rozhraní vyobrazené na obrázku 8. Chyby při aktualizaci jsou zobrazeny s červeným pozadím a jejich celkový počet je uveden v hlavičce stránky.

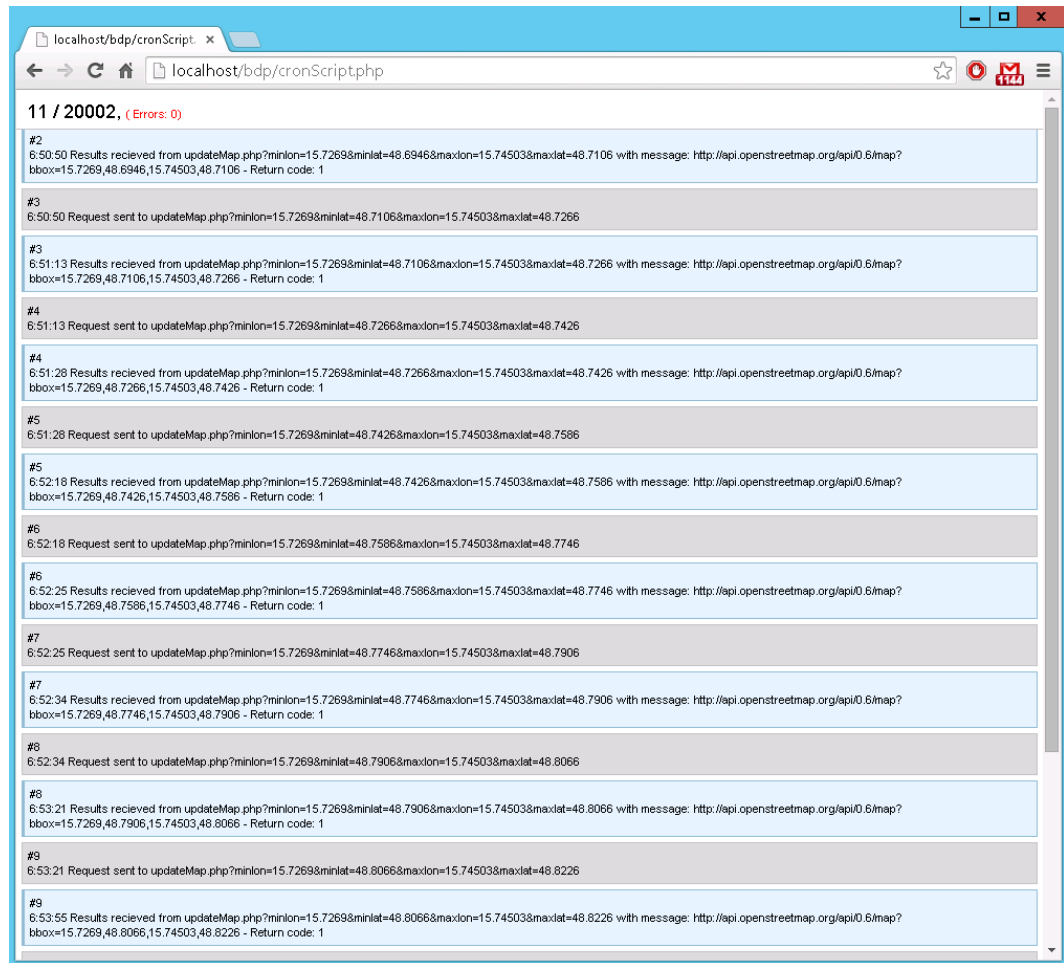
3.6. Zabezpečení skriptů souborem `.htaccess`

Jelikož je aktualizací proces výpočetně náročný, doporučuji tento skript zabezpečit pomocí serveru Apache či konfiguračního souboru `.htaccess` [37]. Opětovným voláním skriptu by totiž mohlo dojít k přetížení serveru. Aplikace popsaná v tomto textu má v souboru `.htaccess` nastaven přístup ke skriptům `indexes.php` a `cronScript.php` pouze z IP adresy 127.0.0.1. Administrátor je schopen po přihlášení spustit dané skripty pouze na serveru. Toho bylo docíleno následující konfigurací:

```
<Files indexes.php>
  order allow,deny
  allow from 127.0.0.1
```

```
</Files>
```

```
<Files cronScript.php>  
  order allow,deny  
  allow from 127.0.0.1  
</Files>
```



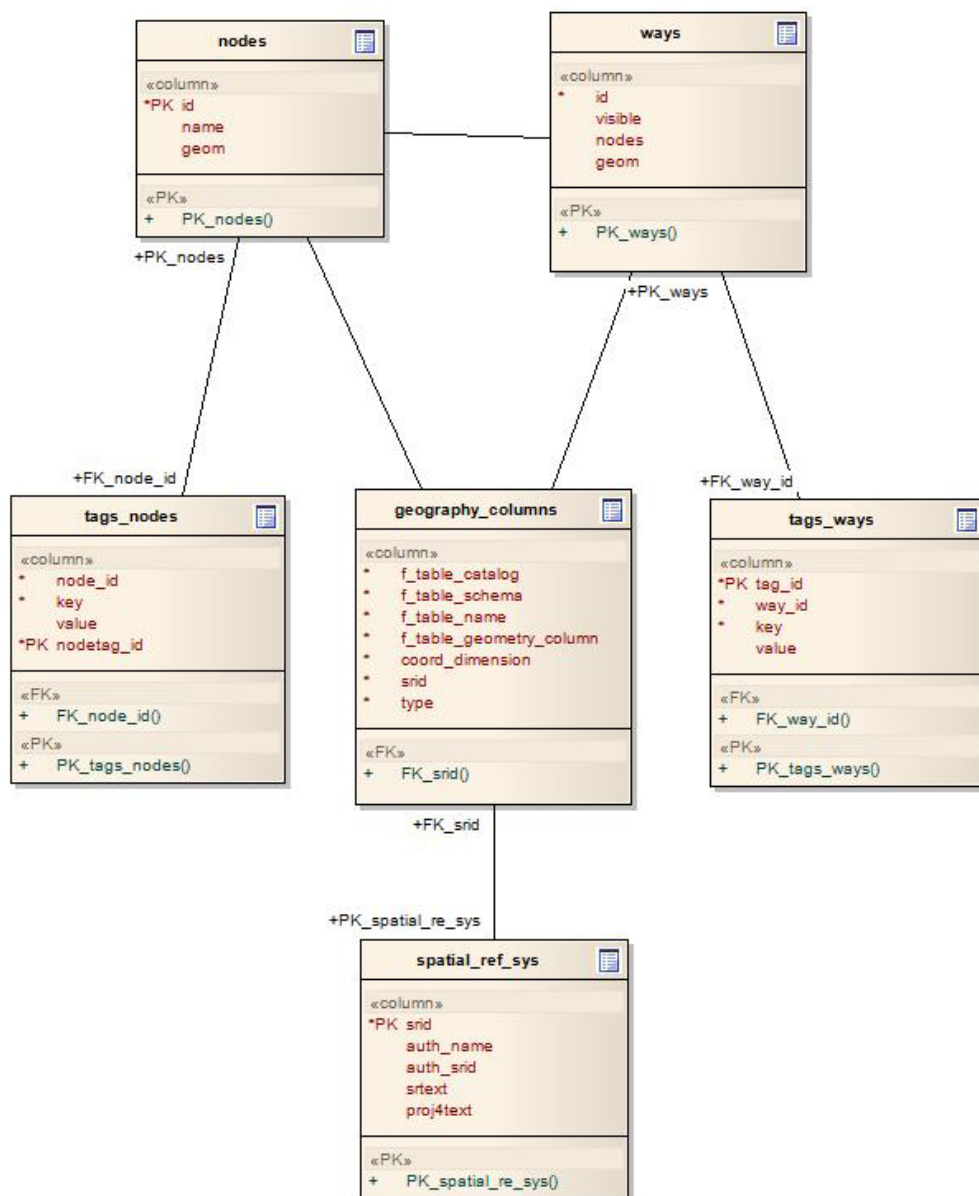
Obrázek 8. Uživatelské rozhraní skriptu cronScript.php.

3.7. Databáze

Jak bylo zmíněno v kapitole 2.3., jako úložiště dat byla vybrána databáze PostgreSQL. Aby bylo možné s mapovými objekty pracovat efektivně, je instalováno rozšířením PostGIS. Detailnější popis použitých verzí při vývoji aplikace je k dispozici v oddíle 3.9..

3.7.1. Schéma

Databázové schéma znázorňuje níže uvedený diagram č. 9.



Obrázek 9. Diagram schéma databáze.

Mimo tabulky, které byly vytvořeny pro uchování mapových dat ve vyvíjené aplikaci (*nodes*, *tag_nodes*, *ways*, *tag_ways*), byla při vytváření databáze s rozšířením PostGIS automaticky nainstalována tabulka *spatial_ref_sys* [41]. Ta

standardně slouží v databázích k definici, které SRID (Spatial Reference System Identifier)[42] jsou povoleny v geometriích, a poskytuje informace o příslušném systému souřadnic pro každé SRID. Tato tabulka je definována ve specifikaci OGC (Open Geospatial Consortium) [43]. V tabulce *geometry_columns* jsou zaregistrovány všechny sloupce určené datovým typem *geometry*, tedy sloupce *geom* z tabulky *nodes* a *geom* z tabulky *ways*. Pro rychlejší vyhledávání v datech bylo zavedeno celkem 10 indexů na attributech, které jsou součástí podmínek dotazů.

3.8. Uživatelské rozhraní

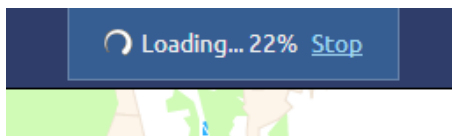
Návrh uživatelského rozhraní je velmi důležitou částí při vývoji softwaru. Klád jsem důraz na jednoduchost a splnění požadavků uvedených v kapitole 2.2. Zároveň jsem uživatelské rozhraní podrobil heuristické analýze [44] kontrolující 10 Nielsenových heuristik [45].

Mapový prohlížeč je tvořen hlavičkou a oblastí s mapou. Hlavními ovládacími prvky jsou tlačítka pro přiblížení a oddálení mapy umístěná na pravé straně. Pro pohyb po mapě lze použít myš či dotyk na dotykových obrazovkách. Mapa obsahuje měřítko umístěné v levém dolním rohu.



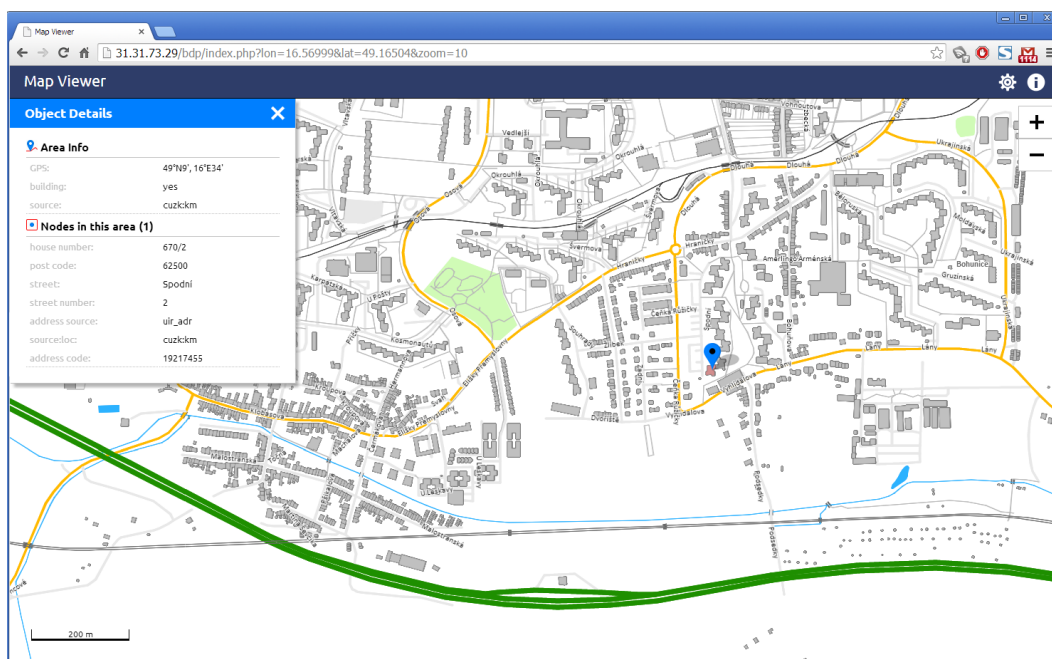
Obrázek 10. Uživatelské rozhraní.

Aby měl uživatel přehled o stavu načtení mapy, je v průběhu načítání dat zobrazen indikátor ukazující stav načtení v procentech s možností zastavit načítání kliknutím na odkaz *Stop*. Následně je možné mapu znovu načíst kliknutím na odkaz *Restart*.



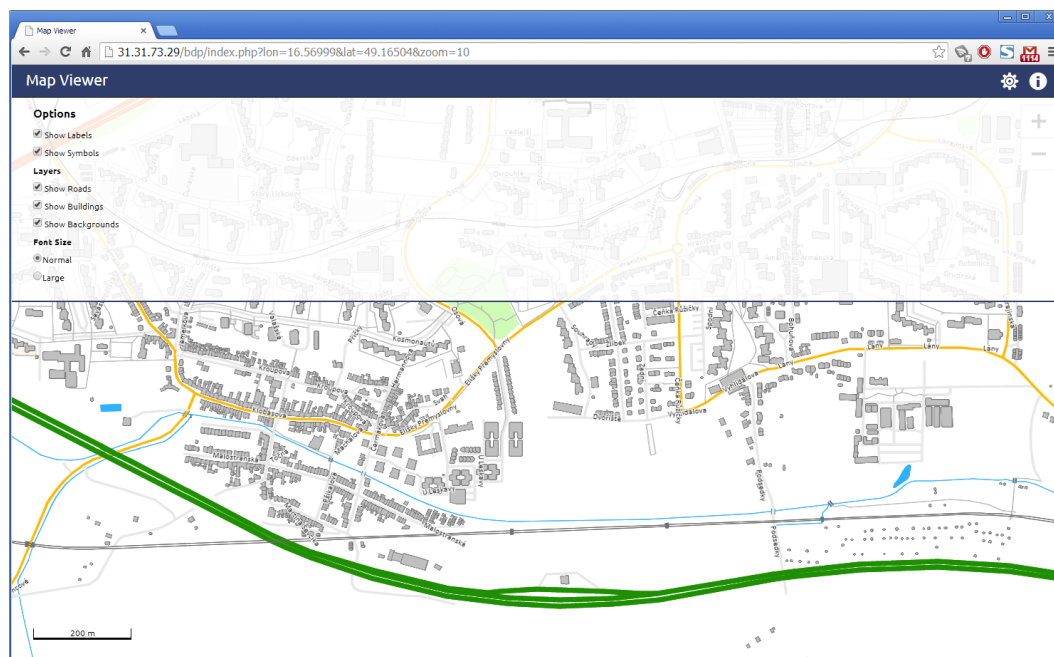
Obrázek 11. Ukazatel načtení mapy.

Po kliknutí na objekt na úrovni, kde jsou povoleny interakce s objekty, je zobrazen levý kontextový panel obsahující informace o označeném objektu (cestě) a souvisejících uzlech. Označený objekt na mapě je zvýrazněn červenou barvou a modrou ikonou umístění. Kliknutím v kontextovém panelu je objekt odznačen a panel skryt.



Obrázek 12. Kontextový panel s ikonou umístění nad označeným objektem.

Hlavička obsahuje 2 aktivní prvky. Kliknout lze na obě ikony v pravé části. Levá z nich otevře panel s možností nastavit zobrazení mapy. Lze skrýt či zobrazit jednotlivé vrstvy mapy (cesty, pozadí, popisky, symboly apod.) a lze přepnout mezi větší či menší velikostí textů. Ikona napravo otevře panel zobrazující informace o projektu a autorských právech. Při velkém přiblížení (vrstvy 11 a 12) jsou pro ukázkou umístěny v mapě i symboly představující křižovatky, restaurace, kavárny a bary. Z časových důvodů nebyly implementovány žádné interakce se symboly. Skrytí symbolů uživatel dosáhne odznačením zaškrťovacího pole s popiskem *Show Symbols*.



Obrázek 13. Panel pro nastavení zobrazení mapy.

3.9. Zprovoznění aplikace

Pro běh aplikace je nutný databázový server PostgreSQL (verze 9.1 a vyšší) s rozšířením PostGIS a HTTP server, například Apache nebo IIS.

Pro účely vývoje a testování aplikace byl zřízen virtuální server u společnosti WEDOS Internet, a.s. s následujícími parametry:

- IP adresa: 31.31.73.29
- Operační systém: Windows Server 2012 R2 Datacenter 64bitový
- Procesor: QEMU Virtual CPU Version, 1.8GHz (2 CPU)
- Paměť: 4096 MB RAM
- Kapacita disku: 60 GB
- Grafická karta: RDPUDD Chained DD

Na virtuální server jsem nainstaloval balíček PostgreSQL verze 9.3.4 Win x86-34 nabízený společností [EnterpriseDB](#), který obsahuje databázový server, Apache server verze 2.4.7 s podporou PHP verze 5.4.23 a nástroje pro správu pgAdmin a nástroj pro instalaci doplňků StackBuilder, který využijeme k instalaci PostGIS 2.1.3. Jednoduchý návod instalace databáze PostgreSQL a PostGISu naleznete například na stránkách [Portál FreeGIS](#).

Po instalaci potřebného softwaru je nutné překopírovat všechny soubory aplikace (adresář /src v příloženém CD - příloha C.) do adresáře www v hlavním adresáři Apache serveru, který je na platformě Windows obvykle na absolutní cestě C:\ProgramFiles(x86)\PostgreSQL\EnterpriseDB-ApachePHP\apache.

Pro definici databázového schématu lze využít SQL skript dostupný na příloženém CD. Přístup aplikace do databáze lze nastavit v konfiguračním souboru. Zároveň doporučuji v tomto souboru před spuštěním aplikace zkontrolovat a případně přenastavit všechny konfigurace. Databáze je ve výchozím nastavení bez mapových dat. Pro jejich načtení z XAPI stačí na serveru spustit skript cron-Script.php. Upozorňuji, že pokud je tento aktualizací skript v provozu, může dojít ke zpomalení načítání map.

3.10. Testování

V závěru vývojového procesu probíhalo statické a dynamické testování programu a následné ladění chyb. U statického testování jsem se zaměřoval na chyby referencí na data, chyby v deklaracích, výpočtu a srovnání. Vzhledem k časovým možnostem jsem však kladl větší důraz na dynamické testování spuštěného programu, a to testům na základní použitelnost a na selhání.

3.10.1. Test na základní použitelnost

Test obsahoval 3 scénáře, umístěné v příloze B., které zkoumaly rychlost načítání mapy, základní funkčnost aplikace a rychlost aktualizace zvolené mapové oblasti.

Tabulka č. 1. zobrazuje seznam zařízení, na kterých byla aplikace testována.

Název	Parametry zařízení	Testované scénáře
VPS	Operační systém: Windows Server 2012 R2 Datacenter 64bitový, Procesor: QEMU Virtual CPU Version, 1.8GHz (1 CPU), Paměť: 2048 MB RAM, Grafická karta: RDPUDD Chained DD.	1, 2, 3
PC	Operační systém: Windows 7 Professional 64bitový, Procesor: Intel Core i3 CPU, M 350, 2.24GHz (4 CPUs), Paměť: 4096 MB RAM, Zobrazení: Intel HD Graphics.	1, 2, 3

Smartphone	Název: LG Optimus L5 IIm (E460) Uhlopříčka: 4"m Operační systém: Android 4.1 (Jelly Bean)m Procesor: 1 000 MHz (1 GHz)m Paměť: 512 MB RAM.	1, 2
Tablet	Název: Prestigio MultiPad PMP5770D, Uhlopříčka: 7", Operační systém: Android 4.0, Procesor: 1,5 GHz (2 CPU), Paměť: 1024 MB RAM, GPU: Mali-400.	1, 2
iPad	Název: Apple iPad mini MD529, Uhlopříčka: 7,6", Operační systém: iOS 6, Procesor: Apple A5X (2 CPU), Paměť: 0,5 GB.	1, 2

Tabulka 1.: Seznam testovacích zařízení.

Na každém zařízení byl vybrán jeden nebo více webových prohlížečů, na kterých probíhaly nezávislé testy. Seznam prohlížečů použitých při testování naleznete v tabulce č. 2..

Název zařízení	Testovací webové prohlížeče
VPS	Google Chrome verze 36.0.1985.125 m
PC	Mozilla Firefox 30.0, Opera 22.0.1471.70
Smartphone	Google Chrome for Android
Tablet	Android Browser 4.0.4
iPad	Safari 6

Tabulka 2.: Seznam testovacích webových prohlížečů.

Testované webové prohlížeče byly zvoleny s ohledem na podporu elementu `canvas` (oddíl 1.1.5.) a statistiky dostupné na adrese StatCounter.com. Pomineme-li různé verze prohlížečů, statistika uvádí, že prohlížeče Google Chrome, Firefox a Opera používalo v červnu 2014 v součtu 69,78 % [46] uživatelů internetu z osobních počítačů. V případě prohlížečů na mobilních zařízeních a tabletech používalo v červnu 2014 prohlížeče Android, Google Chrome a Safari celkem 53,62 % [47] uživatelů internetu.

První testovací scénář zkoumal rychlost načítání mapy na adrese <http://31.31.73.29/bdp/index.php?lon=16.59880&lat=49.16863&zoom=6>, která obsahovala v době testování přibližně 140 tisíc cest a 960 tisíc uzlů uložených v databázi, přičemž na straně klienta bylo zobrazeno 4 tisíce cest tvořených 74 ti-

síci uzly²³. Jelikož aplikace používá výpočetně náročnější techniku vykreslování mapy než je běžné, chceme si testem ověřit, zda je tento přístup použitelný v praxi.

Druhý testovací scénář prověřoval základní funkce mapového prohlížeče na všech zvolených testovacích zařízeních. Z důvodů, že dotyková zařízení (Smartphone, Tablet a iPad) nepodporují *mouse hover efekty* [48], byly u nich vynechány kroky s čísly 3, 8 a 16. Objekty, na které se při testování v krocích 4, 9 a 17 kliklo, byly pro všechna zařízení stejné. Měření doby načítání kontextového menu začalo v okamžiku, kdy bylo kliknuto na mapu a bylo zastaveno po jeho zobrazení.

Poslední scénář testoval průběh aktualizace mapy na testovacím serveru (VPS) a na zvoleném osobním počítači (PC). Aktualizovaná oblast představující velkou část Moravy a Slezska (souřadnice 'minlon' => 15.7269, 'minlat' => 48.6946, 'maxlon' => 19.3524, 'maxlat' => 50.2946) byla rozdělena na 20 tisíc částí. Jelikož je databáze OSM obsáhlá a v průběhu aktualizace může dojít z důvodů absence indexů ke zpomalení zobrazení mapy všem uživatelům, bylo motivací testu zjistit, jak dlouho by případná aktualizace větší mapové oblasti trvala na vybraných zařízeních.

Každý scénář byl pro každé zařízení 3 krát zopakován. Výsledné číselné údaje byly zprůměrovány aritmetickým průměrem. Časové údaje znázorňují čas načítání a čekání z pohledu uživatele. Je tedy nutné brát v úvahu rychlost jednotlivých testovacích zařízení. K měření časových údajů jsem využil stopky na svém mobilním zařízení. Čas byl měřen od zaslání požadavku uživatelem po obdržení požadovaného výsledku ve webovém prohlížeči.

3.10.2. Výsledek testu na základní použitelnost

Tabulka č. 3. obsahuje přehled doby načítání mapy pro jednotlivá testovací zařízení.

Zařízení (prohlížeč)	Doba načítání
VPS	14.1 s.
PC (Firefox)	10.2 s.
PC (Opera)	10.8 s.
Smartphone	52.5 s.
Tablet	43.1 s.
iPad	29.5 s.

Tabulka 3.: Výsledek testovacího scénáře č. 1 pro úroveň přiblížení 6.

Nejhůře dopadl test načítání mapy na zařízení Smartphone, kde jednotlivé testy trvaly 52.6 sekundy, 48.5 sekundy a 56.4 sekundy. Nejlepší výsledky byly

²³Kompletní testovací databáze obsahovala v době testování přibližně 3 miliony uzlů, 210 tisíc cest a 2 miliony tagů.

zaznamenány u zařízení PC v prohlížeči Mozilla Firefox, kde průměrná doba načtení mapy byla 10.2 sekundy. U mobilních zařízení (Smartphone, Tablet, iPad) načítání mapy nebylo plynulé. Obraz po zobrazení hlavičky s indikátorem načítání a měřítko „zamrzl“ a výsledek se zobrazil náhle po kompletním načtení. Při vyšším přiblížení bylo načítání mapy plynulejší a ukazatel načtení mapy zobrazoval stav průběžného načítání. U zařízení iPad se občas objevil výpadek načítání některých vrstev, což bylo dle informací v logu zapříčiněno nemožností navázat spojení s databází. Po znovu načtení stránky již mapa byla zobrazena korektně. U zařízení se systémem Android (Smartphone a Tablet) se pod spodní vrstvou mapy zobrazovalo místy černé pozadí, které při pokusu o interakci s mapou či posunu dočasně zmizelo.

Pro srovnání byl proveden stejný test také pro úroveň přiblížení 12 (konkrétně pro adresu <http://31.31.73.29/bdp/index.php?lon=16.5988&lat=49.16863&zoom=12>). Na mapě je v tomto případě zobrazeno 205 cest a 1370 uzlů na straně klienta. Výsledky jsou obsaženy v tabulce 4.

Zařízení (prohlížeč)	Doba načítání
VPS	2.9 s.
PC (Firefox)	1.6 s.
PC (Opera)	1.7 s.
Smartphone	4.0 s.
Tablet	2.8 s.
iPad	3.1 s.

Tabulka 4.: Výsledek testovacího scénáře č. 1 pro úroveň přiblížení 12.

Výsledky prvního testovacího scénáře ukázaly, že nebyl splněn požadavek na rychlost načítání mapy do 10 sekund při větším oddálení a současných parametrech serveru uvedených v kapitole 3.9.. Zprovozněním aplikace na výkonnějším serveru by bylo dosaženo lepších výsledků.

Druhý testovací scénář proběhl na zařízeních VPS a PC dle očekávání bez zjištěných chyb. Horších výsledků bylo dosaženo na dotykových zařízeních, kdy byl při úrovni oddálení 6 výběr neuzavřených objektů obtížný kvůli menšímu rozlišení obrazovky a nepřesnosti lidského prstu. U zařízení Smartphone se výběr objektů na této úrovni ukázal jako nepoužitelný. Aby bylo možné zobrazit kontextové menu i pro neuzavřené menší cesty, bylo nutné přejít na vyšší úroveň přiblížení. Časy načítání kontextového menu jsou uvedeny v tabulce č. 5.

Přibližování a oddalování mapy fungovalo u všech zařízení dle očekávání a nebylo možné mapu přiblížit přes maximální definovanou úroveň a oddálit přes minimální definovanou úroveň. Pohyb po mapě byl na zařízeních VPS a PC plynulý. Na zařízeních Smartphone, Tablet a iPad byl pohyb po mapě občas trhaný, realizován se zpožděním a někdy byl místo pohybu mapy označen objekt.

Zařízení (prohlížeč)	Doba načítání
VPS	2.4 s.
PC (Firefox)	2.5 s.
PC (Opera)	2.1 s.
Smartphone	8.9 s.
Tablet	15.2 s.
iPad	4.5 s.

Tabulka 5.: Doby načítání kontextového menu na testovacích zařízeních.

Při aktualizaci mapy z OSM se objevily chyby při zpracování dat. Podstatu těchto chyb se nepodařilo identifikovat. Evidentně souvisí s konfigurací daného serveru.

Zaznamenané chyby byly dvojího typu. Nejčastěji (4×) se vyskytovala na zařízení VPS chyba s kódem 73 a popiskem *> required*. Byla to také nejčastější chyba na PC. Vyskytla se v průběhu testování celkem 2 krát. Druhou chybou, která se u obou zařízení vyskytla jednou, je chyba s kódem 5 a popiskem *Invalid document end*. Kontrola XML dokumentů z OSM, které se zpracovaly s chybou, s využitím validátoru ²⁴ neprokázala špatný formát, na něž obě chyby odkazují.

Pokud se vyskytly výše zmíněné chyby, data byla v dané části mapy aktualizována či vložena pouze částečně a na mapě se objevila bílá místa.

Z naměřených hodnot za hodinu provozu skriptu `cronScript.php`, zobrazených v tabulce č. 6., vyplývá, že by celou mapovou oblast aktualizovalo zařízení VPS přibližně 19 dní a 9 hodin a zařízení PC 8 dní a 20 hodin.

Zařízení	Počet načtených částí mapy	Počet chyb (%)
VPS	43	5 (11,6 %)
PC	94	3 (3,2 %)

Tabulka 6.: Výsledek aktualizace mapy za hodinu provozu skriptu `cronScript.php`.

3.10.3. Test na selhání

Účelem tohoto testu bylo zjistit, zda aplikace dokáže reagovat na nestandardní data, která může ovlivnit uživatel či administrátor. Uživatelé mohou ovlivnit parametry v URL adrese a administrátor může upravovat hodnoty v konfiguračním souboru.

Parametry získané z URL adresy jsou ve skriptu `common-functions.js` ve funkci `updateMapDataFromUrl()` testovány na datový typ. Například pokud není hodnota parametru *lon* číslo a není v intervalu od -180 do +180 (zeměpisná

²⁴Validátor dostupný z: <http://validator.w3.org/>.

délka je úhel v rozsahu od 0 do 180 pro východní polokouli a od 0 do -180 pro západní polokouli), jsou použity základní souřadnice nastavené v konfiguračním souboru. Popsanou kontrolu představuje následující JavaScriptový kód ze skriptu `common-functions.js` (řádek 281):

```
if(isNaN(lon) || lon < -180 || lon > 180){  
    lon = defaults[1];  
}
```

Analogicky k tomuto příkladu jsou kontrolovány hodnoty ostatních parametrů.

U konfiguračního souboru byla testována reakce aplikace především na změnu datových typů uvnitř jednotlivých konfigurací, či nahrazení hodnoty určitého klíče speciální hodnotou *null*. Například klíč *leisure* byl nahrazen číslem *25*, libovolný prvek pole s klíčem *visible* byl nahrazen řetězcem nebo bylo celé toto pole nahrazeno hodnotou *null*.

3.10.4. Výsledek testu na selhání

Aplikace uspěla ve všech provedených testech na selhání. Mapový prohlížeč je možné spustit i s nesmyslnými hodnotami parametrů v URL adrese např. `http://31.31.73.29/bdp/index.php?lon=-255&lat=aaa&zoom=-50`. Test konfiguračního souboru prokázal, že změna má vliv pouze na zobrazení mapových objektů konfigurovaných změněnými strukturami.

4. Závěr

Cíle projektu uvedené v kapitole 2.1. byly naplněny. Podařilo se prozkoumat technologie HTML5 a CSS3 a ze získaných vědomostí vytvořit aplikaci, která představuje nový přístup k práci s mapami. Výhodou oproti současným řešením shrnutým v kapitole 1.2.1. je možnost interaktivní práce s mapovými objekty, snadná změna stylů a konfigurace mapy.

Aplikaci je možné do budoucna rozšířit o kompletní data České republiky, lepší zobrazení silniční sítě, nastavení stylů v konfiguračním souboru, interakce s ikonami umístěnými na mapě a zakřivení popisků podle cest. Také by bylo možné využít relace (zmíněné v kapitole 1.3.2.) z databáze OSM, které by mohly nalézt uplatnění například při vyhledávání tras či reprezentaci několika souvisejících cest jako jeden objekt na mapě.

Aplikace bez problémů funguje ve většině webových prohlížečů, avšak v případě mobilních zařízení se objevují nedostatky převážně způsobené omezeným hardwarem a softwarem těchto zařízení. Rychlost načítání mapy na všech testovacích zařízeních bylo možné zvýšit zprovozněním aplikace na výkonnějším serveru. V budoucnu by mohlo být užitečné prozkoumat techniky responzivního web designu (RWD) [49] a Progressive Enhancement [50], které by mohly dopomoci k optimálnímu zobrazení mapy na mobilních zařízeních.

Conclusion

Project objectives set out in the chapter 2.1. were met. It has been managed exploration of HTML5 and CSS3 technologies and the knowledge gained has been used to create an application that presents a new approach to mapping. The advantage of the method presented in this document against the current solution summarized in the chapter 1.2.1. are a possibility to work interactively with map objects and easy definition of styles and map configuration.

The application can be extended in the future of the entire data of the Czech Republic, a better view of the road network, style settings in the configuration file, interactions on icons placed on the map and labels placed on the paths. It would also be possible to use the relations (mentioned in chapter 1.3.2.) of the OSM, which could find application, for example, when searching for routes or represent several related pathways as a single object on the map.

The application works in most web browsers, but in the case of mobile devices, there are errors mainly due to the limited hardware and software on these devices. Speed of loading maps on all test equipment could be increased, if the application will be running on server with better performance. It would also be useful to explore techniques of responsive web design (RWD) [49] and Progressive Enhancement [50] in the future that could help to optimize the map view on mobile devices.

Reference

- [1] HTML5. *w3.org* [online]. [vid. 2014-07-19]. Dostupné z: <http://www.w3.org/TR/html5/>
- [2] Cascading Style Sheets (CSS) Snapshot 2010. *w3.org* [online]. [vid. 2014-07-19]. Dostupné z: <http://www.w3.org/TR/CSS/css3>
- [3] LUBBERS, Peter, ALBERS, Brian, SALIM, Frank. HTML5 Programujeme moderní webové aplikace. 300 s., Computer Press, Brno, 2011.
- [4] HTML 4.01 Specification. *w3.org* [online]. [vid. 2014-07-30]. Dostupné z: <http://www.w3.org/TR/html401/>
- [5] Plan 2014. *w3.org* [online]. [vid. 2014-07-01]. Dostupné z: <http://dev.w3.org/html5/decision-policy/html5-2014-plan.html>
- [6] Differences from HTML4 - W3C Working Draft 28 May 2013. *w3.org* [online]. [vid. 2014-07-01]. Dostupné z: <http://www.w3.org/TR/html5-diff/>
- [7] NOSKA, Martin. HTML5 přináší nové výzvy pro bezpečnost. *Computerworld.cz* [online]. 1. prosinec 2010. [vid. 2014-07-01]. Dostupné z: <http://computerworld.cz/securityworld/html5-prinasi-nove-vyzvy-pro-bezpecnost-8114>
- [8] Geolocation API Specification. *w3.org* [online]. [vid. 2014-07-19]. Dostupné z: <http://dev.w3.org/geo/api/spec-source.html>
- [9] XHTMLTM2.0. *w3.org* [online]. [vid. 2014-07-01]. Dostupné z: <http://www.w3.org/TR/xhtml2/>
- [10] ŠŤASTNÝ, Jan. HTML5 – geolokační rozhraní. *Programujte.com* [online]. 27. srpen 2012. [vid. 2014-07-01]. Dostupné z: <http://programujte.com/clanek/2011052400-html5-geolokacni-rozhrani/>
- [11] Web Storage *w3.org* [online] [vid. 2014-07-19]. Dostupné z: <http://www.w3.org/TR/webstorage/>
- [12] Web Workers *w3.org* [online] [vid. 2014-07-19]. Dostupné z: <http://www.w3.org/TR/workers/>
- [13] Comparison of web browser engines *Wikipedia.org* [online] [vid. 2014-07-19]. Dostupné z: http://en.wikipedia.org/wiki/Comparison_of_web_browser_engines
- [14] HTML5 & CSS3 Support. *fmbip.com* [online]. [vid. 2014-07-03]. Dostupné z: <http://fmbip.com/litmus/>

- [15] Graphics on the Web. *w3.org* [online]. [vid. 2014-07-03]. Dostupné z: <http://www.w3.org/Graphics/>
- [16] Scalable Vector Graphics (SVG). *w3.org* [online]. [vid. 2014-07-03]. Dostupné z: <http://www.w3.org/Graphics/SVG/>
- [17] Canvas (basic support). *Can I use...* [online]. [vid. 2014-07-03]. Dostupné z: <http://caniuse.com/feat=canvas>
- [18] HTML Canvas 2D Context. *w3.org* [online]. [vid. 2014-08-03]. Dostupné z: <http://www.w3.org/TR/2dcontext/the-canvas-state>
- [19] 3D mapy na Mapy.cz s českou technologií Melown Maps. *CAD.cz* [online]. 17. duben 2014. [vid. 2014-07-03]. Dostupné z: <http://www.cad.cz/aktuality/77-aktuality/5105-3d-mapy-na-mapycz-s-ceskou-technologie-melown-maps.html>
- [20] Editing. *OpenStreetMaps Wiki* [online]. [vid. 2014-07-04]. Dostupné z: <http://wiki.openstreetmap.org/wiki/Editing>
- [21] Xapi. *OpenStreetMaps Wiki* [online]. [vid. 2014-07-04]. Dostupné z: <http://wiki.openstreetmap.org/wiki/Xapi>
- [22] API v0.6. *OpenStreetMaps Wiki* [online]. [vid. 2014-07-04]. Dostupné z: http://wiki.openstreetmap.org/wiki/API_v0.6
- [23] HTML5 canvas. *Wikipedia.org* [online]. [vid. 2014-07-04]. Dostupné z: http://cs.wikipedia.org/wiki/HTML5_canvas
- [24] KŮŽEL, Filip. Nové Google Mapy zvedají laťku pro konkurenci. *Navigovat.cz* [online]. 16. květen 2013. [vid. 2014-07-04]. Dostupné z: <http://navigovat.mobilmania.cz/clanky/nove-google-mapy-zvedaji-latku-pro-konkurenci/sc-265-a-1323716>
- [25] Layer - Google Maps JavaScript API v3. *Google Developers* [online]. [vid. 2014-07-04]. Dostupné z: <https://developers.google.com/maps/documentation/javascript/layers>
- [26] Stats. *OpenStreetMaps Wiki* [online]. [vid. 2014-07-04]. Dostupné z: <http://wiki.openstreetmap.org/wiki/Stats>
- [27] Planet.osm. *OpenStreetMaps Wiki* [online]. [vid. 2014-07-04]. Dostupné z: <http://wiki.openstreetmap.org/wiki/Planet.osm>
- [28] Investigace. *Wikipedia.org* [online]. [vid. 2014-07-10]. Dostupné z: <http://cs.wikipedia.org/wiki/Investigace>

- [29] *Draggabilly* [online]. [vid. 2014-07-10]. Dostupné z: <http://draggabilly.desandro.com/>
- [30] CSS Values and Units Module Level 3. Font-relative lengths: the ‘em’, ‘ex’, ‘ch’, ‘rem’ units *w3.org* [online]. [vid. 2014-07-28]. Dostupné z: <http://www.w3.org/TR/css3-values/font-relative-lengths>
- [31] *SAX* [online]. [vid. 2014-07-10]. Dostupné z: <http://www.saxproject.org/>
- [32] PEŠOUT, Martin. Určujte relativní velikost fontů pomocí CSS3. *igloonet blog* [online] 9. únor 2012. [vid. 2014-07-16]. Dostupné z: <http://blog.igloonet.cz/urcujte-relativni-velikost-fontu-pomoci-css3/>
- [33] *PostgreSQL: The world’s most advanced open source database* [online]. [vid. 2014-07-10]. Dostupné z: <http://www.postgresql.org/>
- [34] *PostGIS 2.0.2 Manual* [online]. [vid. 2014-07-10]. Dostupné z: <http://postgis.org/docs/index.html>
- [35] SAALFELD, Alan. Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm. *Cartography and Geographic Information Science, Vol. 26, No.1* [online]. Str. 7-18. 1999. [vid. 2014-07-22]. Dostupné z: <http://web.cs.sunyit.edu/~poissad/projects/Curve/pdf/1999.pdf>
- [36] Cron. *Wikipedia.org* [online]. [vid. 2014-07-22]. Dostupné z: <http://cs.wikipedia.org/wiki/Cron>
- [37] Apache HTTP Server Tutorial: .htaccess files. *Apache HTTP Server Version 2.4* [online]. [vid. 2014-07-22]. Dostupné z: <http://httpd.apache.org/docs/current/howto/htaccess.html>
- [38] SQL: A basic UPSERT in PostgreSQL. *The Art of Web* [online]. [vid. 2014-07-22]. Dostupné z: <http://www.the-art-of-web.com/sql/upsert/>
- [39] PostgreSQL: Documentation: 9.1: WITH Queries (Common Table Expressions) *Postgresql.org* [online]. [vid. 2014-07-22]. Dostupné z: <http://www.postgresql.org/docs/9.1/static/queries-with.html>
- [40] SQL MERGE *PostgreSQL wiki* [online]. [vid. 2014-07-22]. Dostupné z: https://wiki.postgresql.org/wiki/SQL_MERGE
- [41] Chapter 4. Using PostGIS: Data Management and Queries *Postgis.net* [online]. [vid. 2014-07-22]. Dostupné z: http://postgis.net/docs/using_postgis_dbmanagement.htmlspatial_ref_sys
- [42] SRID *Wikipedia.org* [online]. [vid. 2014-07-22]. Dostupné z: <http://en.wikipedia.org/wiki/SRID>

- [43] OGC® Standards and Supporting Documents *OGC®* [online]. [vid. 2014-07-22]. Dostupné z: <http://www.opengeospatial.org/standards>
- [44] NIELSEN, Jakob. How to Conduct a Heuristic Evaluation *Nielsen Norman Group* [online]. 1. leden 1995. [vid. 2014-07-22]. Dostupné z: <http://www.nngroup.com/articles/ten-usability-heuristics/>
- [45] NIELSEN, Jakob. 10 Usability Heuristics for User Interface Design *Nielsen Norman Group* [online]. 1. leden 1995. [vid. 2014-07-22]. Dostupné z: <http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>
- [46] Top 5 Desktop Browsers from June 2013 to June 2014. *StatCounter Global Stats* [online]. [vid. 2014-07-25]. Dostupné z: <http://gs.statcounter.com/desktop-browser-ww-monthly-201306-201406>
- [47] Top 9 Mobile, Tablet Browsers from June 2013 to June 2014. *StatCounter Global Stats* [online]. [vid. 2014-07-25]. Dostupné z: <http://gs.statcounter.com/mobile+tablet-browser-ww-monthly-201306-201406>
- [48] PETCULESCU, Irina. How to deal with :hover on touch screen devices. *Prowebdesign.ro* [online]. 20. leden 2013. [vid. 2014-07-31]. Dostupné z: <http://www.prowebdesign.ro/how-to-deal-with-hover-on-touch-screen-devices/>
- [49] Responzivní web design. *Wikipedia.org* [online]. [vid. 2014-08-03]. Dostupné z: http://cs.wikipedia.org/wiki/Responzivn%C3%AD_web_design
- [50] SLÁDEK, Jan. Graceful degradation vs. progressive enhancement. *Zdrojak.cz* [online]. 4. březen 2009. [vid. 2014-08-03]. Dostupné z: <http://www.zdrojak.cz/clanky/graceful-degradation-vs-progressive-enhancement/>

A. Zadání projektu

Cílem práce je vytvořit aplikaci, která bude zobrazovat mapová data (OpenStreetMap) a související informace v prohlížeči s využitím možností HTML5. Aplikace by měla poskytovat funkcionalitu běžnou pro prohlížení map, zejména plynulý pohyb v mapě a plynulé přibližování nebo oddalování mapy. Vedle toho by aplikace měla být použitelná ve většině soudobých prohlížečů (podporujících HTML5) a na většině zařízení včetně přenosných zařízení. Aplikace by měla být schopna vhodným způsobem redukovat množství přenášených dat na klienta.

B. Testovací scénáře

ID:	1
Název:	Zobrazení mapy
Účel:	Ověření, zda proběhne zobrazení mapy
Typ testu:	Verifikační
Čas:	1 min.
Kroky:	<ol style="list-style-type: none">1. Zadáání adresy <code>http://31.31.73.29/bdp/</code> do webového prohlížeče testovaného zařízení.2. Zapnutí časomíry.3. Čekání na načtení mapy.4. Vypnutí časomíry.
Očekávaný výsledek:	<ol style="list-style-type: none">1. Mapa se načte i bez zadaných parametrů v URL.2. Doba načítání mapy není delší než 10 sekund.

ID:	2
Název:	Test základních interakcí s mapou
Účel:	Ověření, zda jsou dostupné uživatelské interakce dle specifikace
Typ testu:	Verifikační
Čas:	5 min.

Kroky:

1. Zadání adresy `http://31.31.73.29/bdp/` do webového prohlížeče testovaného zařízení.
2. Čekání na načtení mapy.
3. Pohyb myši nad mapou, sledování interakcí.
4. Kliknutí na libovolnou neuzavřenou cestu.
5. Zapnutí časomíry.
6. Čekání na načtení doplňkových informací o objektu.
7. Vypnutí časomíry.
8. Pohyb myši nad mapou, sledování interakcí.
9. Kliknutí na libovolnou uzavřenou cestu.
10. Kontrola zobrazených doplňkových dat a pozice ikony umístění nad označeným objektem.
11. Zavření okna doplňkových dat kliknutím na křížek v pravém horním rohu dialogu.
12. Kliknutí na prázdné místo na mapě.
13. Posunutí mapy v libovolném směru.
14. Čekání na načtení chybějících částí mapy.
15. Klikání na tlačítko přiblížení až do úrovně 11.
16. Pohyb myši nad mapou, sledování interakcí.
17. Kliknutí na libovolnou uzavřenou cestu.
18. Posunutí mapy v libovolném směru.
19. Přiblížení na maximální úroveň (12).
20. Kliknutí na tlačítko přiblížení.
21. Oddálení na minimální úroveň (1).
22. Kliknutí na tlačítko oddálení.

Očekávaný výsledek:	<ol style="list-style-type: none"> 1. Po najetí myši nad mapový objekt je objekt zvýrazněn. 2. Po kliknutí je objekt zvýrazněn a na pozici kliknutí je zobrazena ikona umístění. 3. Pohyb po mapě je plynulý. 4. Při přibližování či oddálení nedochází ke znovunačtení stránky. 5. Nelze přibližovat či oddalovat mapu mimo mezní úroveň (1 a 12). 6. Zvýraznění objektů trvá od 1 do 3 sekund po kompletním načtení mapy a umístěním kurzoru. 7. Doba načítání souvisejících dat objektu je menší než 10 sekund od kliknutí na objekt.
---------------------	---

ID:	3
Název:	Aktualizace mapy
Účel:	Ověření, zda proběhne aktualizace mapových dat
Typ testu:	Verifikační
Čas:	1 hodina.
Kroky:	<ol style="list-style-type: none"> 1. Zadání adresy <code>http://localhost/bdp/cronScript.php</code> do webového prohlížeče testovaného serveru. 2. Sledování průběžných výsledků. 3. Opakování chybného výsledku.

Očekávaný výsledek:	<ol style="list-style-type: none"><li data-bbox="675 389 1402 461">1. Výsledky budou průběžně aktualizovány na webové stránce.<li data-bbox="675 495 1402 602">2. Doba čekání na jeden výsledek nepřesáhne maximální dobu nastavenou ve skriptu update-Map.php.<li data-bbox="675 636 1402 707">3. Počet chybných výsledků bude menší než 10 % z celkového počtu zpracovaných výsledků.
---------------------	---

C. Obsah příloženého CD/DVD

V samotném závěru práce je uveden stručný popis obsahu příloženého CD/DVD, tj. závazné adresářové struktury, důležitých souborů apod.

`doc/`

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PřF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

`src/`

Kompletní zdrojové texty webové aplikace PROHLÍŽEČ MAP v HTML5 se všemi potřebnými (převzatými) zdrojovými texty, knihovnamy a dalšími soubory pro zkopírování na webový server.

`createDBScheme.sql`

Soubor slouží k vytvoření databázového schématu.

`readme.txt`

Instrukce pro nasazení webové aplikace PROHLÍŽEČ MAP v HTML5 na webový server, včetně požadavků pro její provoz, a webová adresa, na které je aplikace nasazena pro testovací účely a pro účel obhajoby práce.

U veškerých odjinud převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro materiály, u kterých toto není splněno, je uveden jejich zdroj (webová adresa) v textu dokumentace práce nebo v souboru `readme.txt`.