VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

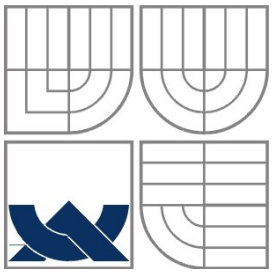
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

WEBOVÉ ROZHRANÍ PRO MAPOVÝ SERVER ZALOŽENÝ NA INFRASTRUKTUŘE OPENSTREETMAP

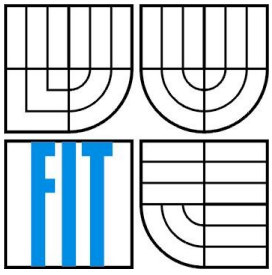
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE MICHAL BAČA
AUTHOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

WEBOVÉ ROZHRANÍ PRO MAPOVÝ SERVER ZALOŽENÝ NA INFRASTRUKTUŘE OPENSTREETMAP

OPENSTREETMAP INFRASTRUCTURE BASED MAP SERVER WEB INTERFACE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL BAČA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. LUKÁŠ POLOK

BRNO 2013

SEM VLOŽIT
ORIGINÁL ZADANÍ

Abstrakt

Tato práce se zabývá návrhem a implementací webové mapové aplikace se zaměřením na turisty. Hlavním smyslem aplikace je posunout používaní mapových podkladů od pouhého zobrazování statické informace do stavu, kdy mapa slouží jako základová vrstva, k níž uživatel vztahuje pro něj relevantní informace se záměrem plánování pěších výletů. Součástí práce je vytvoření atraktivního uživatelsky přívětivého interaktivního rozhraní. Aplikace je implementována pomocí HTML, JavaScriptu, CSS a PHP s MysSQL na straně serveru. Klientská aplikace je vytvořena pomocí knihoven OpenLayers, která se stará o práci s mapou, a knihovnami jQUery a jQuery UI pro logiku a rozhraní aplikace.

Abstract

This thesis deals with designing and implementing web map application targeted at tourist. The main focus of this application is to shift usage of map from just displaying static data to point, where the map itself serves as a canvas which user can use to store pieces of information relevant to planned trip. Part of the thesis is to design attractive easy to use user interface. The application was implemented with HTML, JavaScript, CSS and PHP with MySQL on server. The client uses OpenLayers library for displaying map and jQery and jQuery UI libraries to implement application logic and user interface.

Klíčová slova

OpenStreetMap, OpenLayers, mapová aplikace, JavaScript, jQuery, turistika, webová aplikace, plánování výletů

Keywords

OpenStreetMap, OpenLayers, map application, JavaScript, jQuery, tourism, rich internet application, trip planning

Citace

Michal Bača: Webové rozhraní pro mapový server založený na infrastruktuře OpenStreetMap, bakalářská práce, Brno, FIT VUT v Brně, 2013

Webové rozhraní pro mapový server založený na infrastruktuře OpenStreetMap

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Lukáše Poloka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Michal Bača
6.5.2013

Poděkování

Děkuji panu Ing. Lukáši Polokovi za odborné vedení práce.

Obsah

1 Úvod

Cílem této práce je navrhnout a vytvořit uživatelské rozhraní pro mapovou aplikaci cílenou na pěší turistiku. Mapové podklady jsou přebírány z projektu OpenStreetMap. Jedná se o projekt, jehož cílem je vytvořit svobodné mapy světa. Mapy jsou tvořeny dobrovolníky, kteří pomocí systému GPS sledují svoji polohu a na základě těchto dat pak vytváří mapu. Do projektu také přispívají různé společnosti a státní instituce.

Dnešní mapové aplikace pouze napodobují klasické papírové mapy. Slouží jen pro prohlížení a případně pro vyhledávání míst a tras, prohlížení statických dat. Pro využití v turistice, například jako plánovač, neexisují uživatelsky přívětivé služby. Z tohoto důvodu vznikla tato aplikace, měla by poskytnou uživatelům nástroje pro naplánování výletu a zasazení pro ně důležitých informací do kontextu mapy. Takto naplánovaný výlet společně se všemi informacemi, které uživatel do mapy vložil, je pak možné sdílet s ostatními.

Práce je strukturována následujícím způsobem. První část je věnována v obecné rovině současným mapovým aplikacím, její kapitoly popisují projekt OpenStreetMap, způsob, jakým vznikají mapy, jak se uchovávají a také informace o hlavním portálu projektu. Poslední kapitola této části je věnovaná knihovně OpenLayers, která je srdcem vytvářené aplikace. Pomocí této knihovny probíhá vykreslování mapy ve webovém prohlížeči a také zajišťuje interakci s mapou.

V další části je popsán smysl aplikace a v navazujících kapitolách je pak podrobněji popsána klíčová funkčnost tvořící jádro aplikace. Následuje návrh uživatelského rozhraní, v podkapitolách jsou pak popisy jednotlivých elementů navrhovaného rozhraní.

Třetí část se pak podrobně zabývá implementací aplikace. Je rozdělena na dvě kapitoly, jedna je věnována implementaci serverové části, druhá pak části klientské. Popis klientské části je dále rozdělen do podkapitol zachycujících jednotlivé funkční bloky aplikace. Kapitola o serverové části se pak zabývá schématem databáze, ve které se ukládají uživatelova data a PHP skripty, které zprostředkovávají komunikaci mezi klientem a databází.

2 Webové mapy, OpenStreetMap, OpenLayers

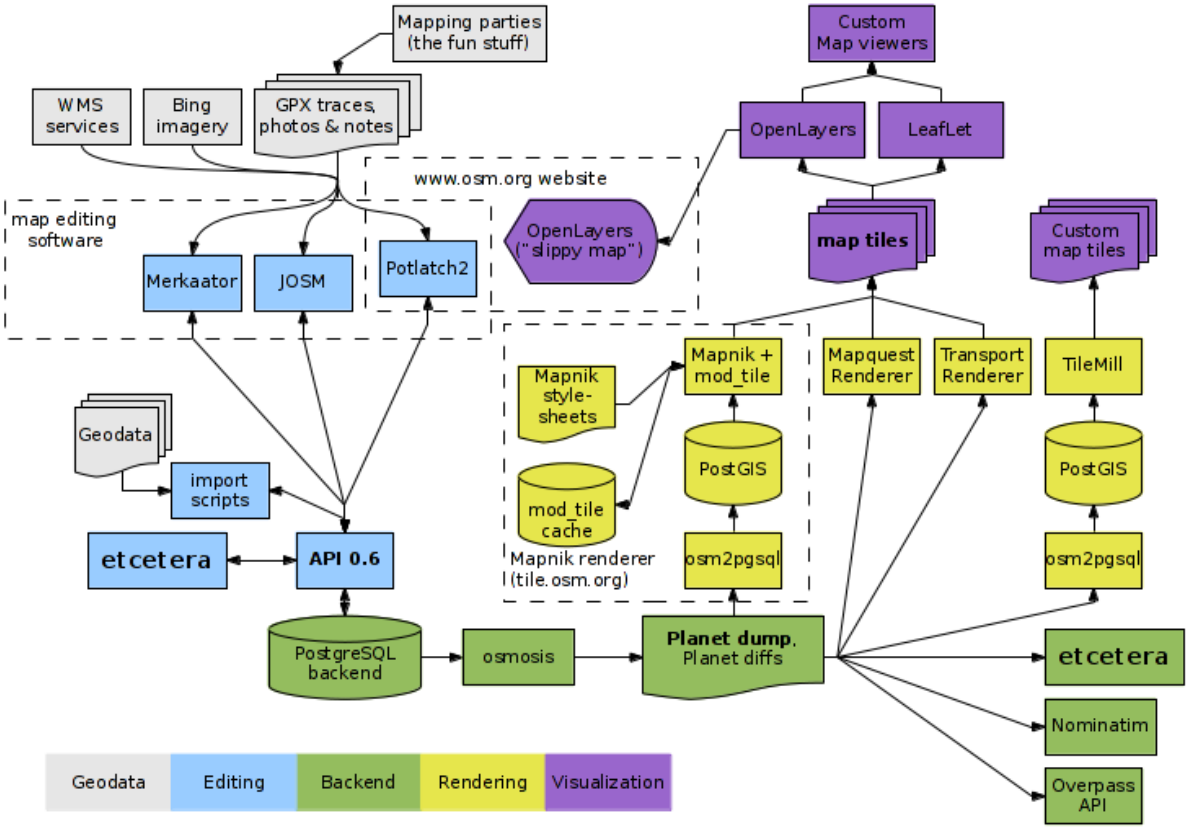
Existuje celá řada webových map, ale ve většině případů pouze přejímají mapové podklady a funkčnost od Google Maps, Here Maps (bývalé Nokia Maps) a z českých zástupců od Mapy.cz. Nenabízí tedy nic, co by výše zmíněné portály neměly. Pak máme OpenStreetMap a od ní odvozené projekty. Jedním z takto odvezených projektů je například OpenTrackMap.

Většina map nabízí především motoristické mapy a letecké nebo satelitní snímky. Podporují vyhledávání adres a firem a také vyhledávání tras. Mapy.cz pak ještě nabízejí zobrazení turistických stezek, podobně jako OpenTrackMap. Všechny výše zmíněné služby pracují s digitální mapou jako s její papírovou předchůdkyní. Mapa slouží k zobrazení statický dat, bez nějaké hlubší interakce s uživateli. Pouze usnadňují vyhledání míst a nalezení (silniční) trasy. Hlubší spolupráce s mapami se začíná prosazovat zatím na mobilních zařízeních. Webových služeb, které mapu požívají jako podklad pro uživatelova data, moc není.

2.1 OpenStreetMap

OpenStreetMap je projekt, jehož cílem je vytvořit svobodné mapy světa, které by byly dostupné komukoliv a kdokoliv by se mohl účastnit jejich tvorby. Mapová data jsou vytvářena uživateli, podobně jako například Wikipedia, kteří zaznamenávají pomocí navigačního systému GPS svoji polohu a trasu. Na základě těchto dat jsou pak vytvořeny mapy. Na základě zaznamenaných tras jsou vytvořeny silnice a stezky, jejich průsečíky (křižovatky) a jsou jim přiřazeny atributy popisující například typ silnice, její značení a podobně. Stejným způsobem se vkládají budovy, železniční trasy, památky a podobně. Dalším způsobem vytváření map je překreslování leteckých snímků, které k tomuto účelu poskytla například společnost Microsoft. Mezi přispěvatele mapových dat kromě členů komunity patří i firmy a různé státní organizace po celém světě.

Vlastní projekt se skládá z databáze, obsahující jednotlivé elementy a jejich vazby, API pro přístup k datům, a webového portálu. Schéma systému je uvedeno na obrázku 1.



Obrázek 1: Schéma projektu OpenStreetMap

zdroj: http://wiki.openstreetmap.org/wiki/File:OSM_Components.png [cit. 2013-05-06]

Z databáze (nebo ze souboru Planet.osm, více dále) se vytvářejí vektorové nebo rastrové obrázky, mapové dlaždice, pomocí vykreslovacích systémů. Pro web projektu se používá systém Mapnik, dalším populárním systémem je Osmarender. Vykreslené mapové podklady jsou uloženy na kešovacích serverech, ze kterých jsou stahovány a poté zobrazeny klientskou aplikací, vytvořenou například pomocí knihovny OpenLayers. Celá mapa je rozdělena na čtvercové oblasti, dlaždice, které se podle potřeby stahují a poskládáním vedle sebe vytváří ucelenou mapu. Webový portál kromě stránek určených pro vývojáře a přispěvatele obsahuje modul pro zobrazování a editaci map přímo z prohlížeče, vyhledávání a diáře [4]. Některé části systému budou popsány dále.

2.1.1 Databáze

Tato databáze udržuje všechna mapová data. Data jsou tvořena ze 4 základních prvků: uzel (Node), cesta (Way) a vazba (Relation). Výše zmíněné prvky pak mohou mít jeden nebo více přidělených značek (Tag).

Uzel zastupuje jeden prostorový bod zadaný souřadnicemi a volitelnou nadmořskou výškou. Slouží k vytvoření bodových objektů (např. telefonní budka) nebo se uzly používají k sestavení cesty. Cesty značí silnice, železnice a podobně. Plochy lze vytvořit pomocí cesty, kde první a poslední bod je stejný. Vazba je seřazený seznam uzlů, cest a někdy i jiných vazeb. Používá se k seskupení objektů. Například pro vytvoření polygonů s otvorem – seskupí se dva polygony, jeden jako vlastní polygon, druhý jako díra. Značky pak slouží k popisu výše uvedených elementů. Značka je dvojice klíče a hodnoty nesoucí informaci, například o typu cesty (silnice, dálnice, stezka...) a další pomocné informace jako číslo, název a mnoho dalších [5].

Pro přístup k databázi se v současnosti používá API v0.6. Dalším způsob, jak získat data, je použít soubor Planet.osm. Jedná se o každý týden aktualizovaný soubor obsahující kompletní databázi (tedy všechny uzly, cesty, vazby a jejich značky) pro celý svět. Tento soubor je šířen jako XML soubor nebo v binární podobě [6].

Editace map vlastně upravuje záznamy databáze. Upravovat lze přímo data na serverech, ke kterým se přistupuje pomocí API prostřednictvím nástrojů jako JOSM, ArcGIS Editor for OSM, Potlach a mnoho dalších. API je založena na principech architektury REST pro distribuovaná prostředí. Data jsou přenášena ve formátu XML [7].

2.1.2 Vykreslování dlaždic

Z databáze je potřeba vytvořit člověkem čitelnou mapu. K tomuto účelu slouží vykreslovací systémy, které z databázových tabulek nebo ze souboru Planet.osm vytvoří (typicky) rastrový obraz obsahující silnice, dálnice, železnice, lesy, budovy a jiné prvky nacházející se v mapách. Vykreslování probíhá nezávisle na databázi OpenStreetMap, na serverech speciálně určených pro tento úkol. Vytvořené dlaždice pak klientské aplikace zobrazují jako mapu.

Projekt OpenStreetMap používá vykreslovací systém Mapnik, dalším je například systém Osmarender, který generuje vektorové soubory.

Sytém Mapnik je otevřený multiplatformní nástroj pro vykreslování mapových dat, je napsán v C++, má rozhraní pro Python. Nabízí vykreslování s antialiasingem se subpixelovou přesností, podporuje velké množství vstupních a výstupních formátů. Vykreslovat umí jak do rastrových formátů, tak i do SVG. Obvyklý způsob vykreslování dat OpenStreetMap je naplněním PostgreeSQL databáze ze souboru Planet.osm a následné vykreslování z této databáze. Vzhled map lze měnit pomocí stylů uložených v XML souborech [8].

2.1.3 Portál OpenStreetMap.org

Portál je napsán s použitím frameworku Ruby on Rails. V dřívějších verzích se pro zobrazování mapy používala knihovna OpenLayers, v současnosti se používá knihovna Leaflet. Kromě zobrazování mapy portál umožňuje přímou editaci dat prostřednictvím nástroje Potlach2. Další funkce portálu souvisí s komunitním charakterem projektu – diáře, které mají pomoci uživateli udržet přehled o změnách, které provedli v mapě. Dále pak komunikaci mezi uživateli a wiki stránky pro přispěvatele a vývojáře [9].

Potlach2 je editor pro úpravu map, který je součástí portálu. Jedná se přepis původního nástroje Je napsán v jazyce ActionScript3 pomocí technologií Flex Podporuje import tras zaznamenaných pomocí GPS a snadnou editaci mapových dat. Lze ho vložit i do jiných stránek [10].

2.2 OpenLayers

OpenLayers je svobodná JavaScriptová knihovna pro zobrazování mapových dat ve webovém prohlížeči bez závislosti na serveru, vyvíjena je komunitou jako svobodný projekt pod záštitou konsorcia OSGeo. Je uvolněna pod BSD licencí se dvěma klauzulemi.

OpenLayers implementuje průmyslové standardy pro přístup k geografickým datům, protokoly Web Mapping Service (WMS) a Web Feature Service (WFS). Podporuje však mnohem více zdrojů jako GeoJSON, GeoRSS, KML, dokáže zobrazit mapové podklady od firem Google, Yahoo či Microsoft.

Objektově orientované API je rozděleno na několik základních tříd, které budou dále stručně popsány. Kompletní popis všech tříd je dostupný v dokumentaci projektu [11].

`OpenLayers.Map` – je třída reprezentující interaktivní mapu vloženou do webové stránky. Pro rozšíření funkčnosti se do mapy vkládají ovládací prvky (objekty `OpenLayers.Control`)

a vrstvy (`OpenLayers.Layer`). Vrstvy jsou v navrhované aplikaci hojně využity pro logické členění aplikace.

`OpenLayers.Layer` – jedna z důležitých tříd. Všechny viditelné prvky v mapě včetně mapových podkladů jsou umístěny ve vrstvách. Tedy v instancích této třídy. Jedná se obecnou třídu, od níž se dědičností odvozují další podtřídy. Podporovány jsou například následující typy vrstev: `Markers` – vrstva obsahující markery, což jsou ikony zobrazené v mapě, `Vector` – vrstva pro zobrazení vektorových dat z různých zdrojů. V této aplikaci se používají výhradně tyto vrstvy pro svou všestrannost. Ve výchozím nastavení se vykreslují geometrické objekty do vektorového formátu SVG. Dále vrstvy `WMS` pro zobrazování dat získaných protokolem WMS, `OSM` pro zobrazování mapových podkladů OpenStreetMap, a vrstvy pro komerční služby jako například Google Maps, Bing, Yahoo Maps.

`OpenLayers.Control` – je základní třída pro prvky, které ovlivní zobrazování nebo chování mapy. Jedná se například o ovládací prvky pro pohyb po mapě, přiblížení a oddálení, měření, výběr prvků a mnoho jiných. Všechny ovládací prvky se přidají do mapového objektu, ať už přímo či nepřímo jako součást panelu. `PanZoom` – je panel pro pohyb po mapě, obsahuje 4 šipky pro pohyb na sever, jih, východ, západ a tlačítko pro přiblížení a oddálení mapy. `PanZoomBar` obsahuje kromě výše zmíněných elementů také posuvník pro přiblížení nebo oddálení mapy. `OverviewMap` je malá náhledová interaktivní mapa pro snazší orientaci. Pomocí této mapky lze posouvat i mapu zobrazenou v hlavním okně. `KeyboardDefaults` přidá mapě ovládání pomocí klávesnice. Pomocí šipek se mapa posunuje, klávesou plus se přiblíží, klávesou mínus se oddálí, klávesy Page Down, Page Up, Home a End posunou obrazovku o tři čtvrtiny.

`Events` – funkce pro obsluhu událostí – stisky kláves, tlačítek, detekce vícenásobných stisků, různých události navázaných na objekty v mapě (přidání, odebraní, a mnoho dalších), posuny a změny mapy.

`LonLat` – uchovává geografické souřadnice jako dvojici zeměpisné délky a šířky. Souřadnice lze transformovat mezi různými projekcemi.

Tato knihovna se používá pro zobrazování map na webu OpenStreetMap.org, a mnoha komerčních aplikacích. V současnosti se chystá 3 verze této knihovny, která by měla přinést například podporu WebGL, HTML5 a CSS3.

3 Návrh aplikace

3.1 Cíl aplikace

V současnosti veškeré mapové aplikace pouze slouží ke konzumaci obsahu. Převádějí pouze mapové podklady do digitální podoby. Princip použití zůstává stejný jako u papírových map – prohlédnout si mapy a najít stezku nebo nějaké místo, kde je mapa brána jako statická, neměnná množina informací. Digitální mapy ale mají větší potenciál, mohou uživatelům dovolit zasahovat do mapy bez jejího poškození a ztráty hodnoty. Doplnit do ní pro ně relevantní informace, ať už textové poznámky, nebo vlastní trasy, body zájmu, zajímavá místa atd. A pak je i sdílet. Z mapy lze vytvořit „poznámkový blok“, plánovač výletů či jakoukoliv jinou aplikaci, která dokáže smysluplně využít mapová data. Ale takových aplikací zabývající se turistikou mnoho není. OpenStreetMapy obsahují velké množství historických a přírodních památek, vyhlídek, stezek či jiných turisticky zajímavých míst. Tato aplikace by toho chtěla využít. Principem je vzít již existující data, dát je k dispozici uživatelům v atraktivní formě a dát jim nástroje, aby mapu (a její data) využili, tak, jak jim to vyhovuje. Nejen prohlížení dat, ale i „editaci“. Tato aplikace by měla uživatelům (turistům) dát možnost si v jedné mapě, v jedné aplikaci, najít zajímavá místa, naplánovat si výlet a sdílet plán s přáteli. K tomu jim nabídne řadu nástrojů. Některé běžné z ostatních webových map, jiné ne.

3.1.1 Itinerář

Jednou z důležitých součástí je itinerář. Itinerář lze použít jako číslovaný seznam zajímavých míst, kam by se chtěl turista podívat. Lze ho využít jako seznam bodů na trase výletu. Itinerář je kompletně tvořen uživatelem, který do mapy přidává body (místa) a k nim komentář, popisek. Tyto body se nacházejí ve vlastní vrstvě a lze je pro lepší přehlednost dle potřeby skrýt.

3.1.2 Zobrazení fotek

Jednou z podpůrných funkcí je zobrazovaní fotek ze služby Panoramio.com. Pro lepší představu o místech, které uživatel na mapě nalezl, může využít integraci fotek, bez opouštění této aplikace. Prohlížením mapy se zapnutím zobrazování fotek lze také narazit na nová místa.

3.1.3 Kreslení

Do kteréhokoliv místa lze do mapy zakreslit body, polygony a čáry. Tyto objekty se nacházejí ve vlastní vrstvě a lze je kdykoliv zobrazit nebo skrýt. Nakreslené objekty je možno dále upravovat – lze je přesouvat, mazat a, v případě polygonů, také editovat.

Nakreslené objekty by měly především sloužit pro označení míst a areálu, u kterých nevyžadujeme další vysvětlující text. Dále jako upozornění na nějakou známou skutečnost, data v mapě či jako záchytný bod na trase nebo pro prezentaci. Čáry mohou sloužit také k výše zmíněným účelům, nebo také pro zaznačení cesty, případně pro tvorbu šipek.

Kreslit lze ve čtyřech barvách, každý objekt může být jinou barvou. Po zakreslení do mapy barva měnit nelze.

3.1.4 Měření

Pro lepší představu o vzdálenostech v reálném světě aplikace podporuje měření vzdáleností mezi dvěma nebo více body. Měření probíhá zakreslováním čar do mapy a aplikace zobrazuje celkovou vzdálenost a vzdálenost od počátku k poslednímu přidanému segmentu čáry.

3.1.5 Značky

Značky slouží pro zaznačení míst, u kterých na rozdíl od kreslení požadujeme vysvětlující text. Aplikace rozlišuje dva typy značek, a to textovou značku a odkaz. Textová značka slouží jako komentář, nebo textová poznámka. Skládá se z titulku o maximální délce 50 znaků a z neomezeně[1] dlouhého textu.

Odkaz slouží pro vložení hypertextového odkazu. Nějaké místo je možné označit odkazem na web pro více informací, např. Stránky památky s informacemi o vstupném. Kliknutím na odkaz je pak uživatel přesměrován na tuto adresu.

3.1.6 Persistence dat a sdílení

Veškerá uživatelská data, to znamená itinerář, zakreslené body, čáry, polygony, a značky se ukládají v databázi na serveru. Identifikační číslo uživatele se ukládá do souboru cookies v prohlížeči. Do jisté míry je to omezující (jeden uživatel používající více počítačů nebo prohlížečů), ale nepotřebuje řešit

1 Prakticky je délka omezena datovým typem v databázi na 65,535 znaků.

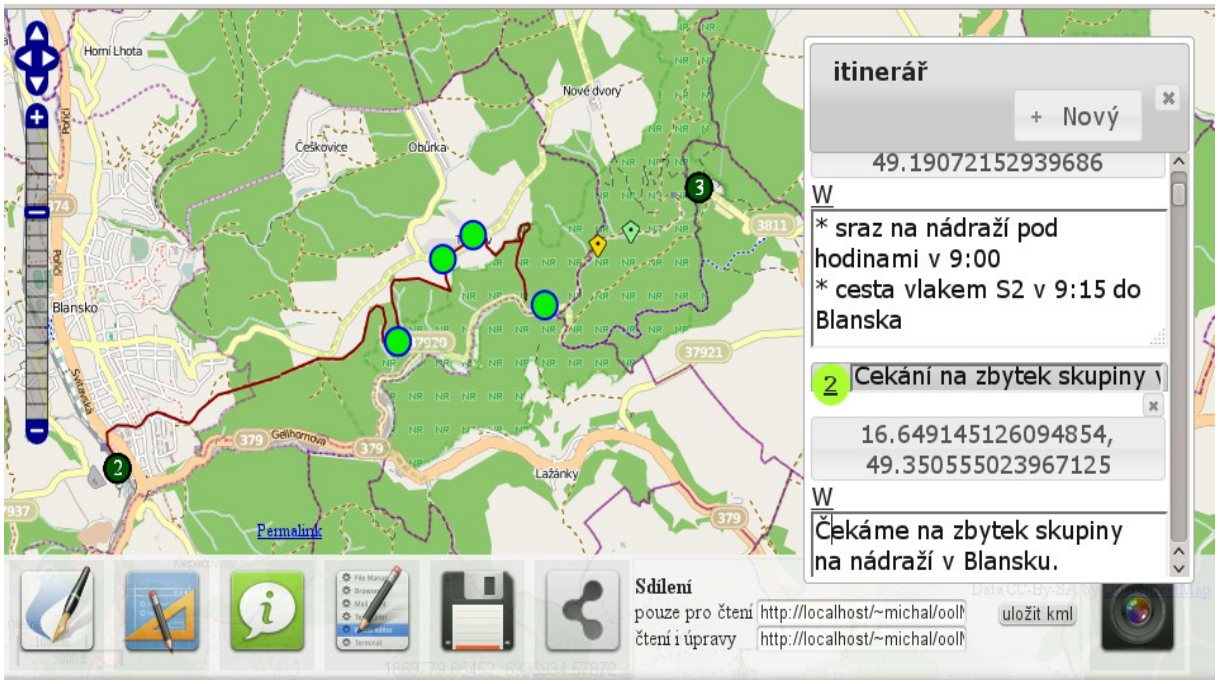
autentizaci, která by mohla mnoho uživatelů odradit. Tento problém lze do jisté míry obejít pomocí odkazu pro sdílení.

Uložená data by byla k ničemu, pokud by nešla sdílet. Proto je možné vygenerovat odkaz pro sdílení. A to buď pouze ke čtení, nebo i pro úpravy. Tento odkaz pak lze sdílet běžnými komunikačními kanály. V případě odkazu pro úpravy se upravuje pouze lokální kopie dat, nikoliv data původního vlastníka. Upravená data si pak může uživatel uložit. Itinerář lze také stáhnou jako KML (formát pro přenos a publikaci geografických dat) soubor a dále s ním pracovat v aplikacích typu Google Earth nebo Marble.

3.1.7 Návrh uživatelského rozhraní

Cílem bylo navrhnout a posléze implementovat atraktivní, interaktivní, uživatelsky přívětivé rozhraní pomocí webových technologií, tedy HTML, CSS a JavaScript. Pro jednoduchost a intuitivnost by každá akce měla mít svoji vizuální reprezentaci – např. změna kurzoru myši, grafiku kolem kurzoru, zvýraznění aktivní položky v panelu a podobně tak, aby v každém okamžiku byl uživatel schopen rozeznat, s jakým nástrojem pracuje. Také každý prvek nesoucí další informace, které nejsou přímo viditelní jeho grafickou podobou, by měl nenásilným způsobem uživateli dát najevo, že tuto informaci nese. Toho bylo dosaženo vytvořením plovoucí nabídky a zvýrazněním prvku na mapě po najetí myši. To dá uživateli signál, že s tímto prvkem může nějak dál manipulovat. Plovoucí nabídka je vysvětlena v dalších kapitolách. Pro snazší navigaci byla vytvořena plochá struktura dělící veškerou funkcionalitu do několika kategorií.

Hlavní dvě součásti rozhraní jsou ovládací panel na spodní hraně obrazovky a itinerář. Tyto součásti jsou k vidění na na obrázku 2. Ovládací panel je rozdělen do kategorií a obsahuje tlačítka pro jednotlivé funkce. Itinerář obsahuje seznam míst, kde každé místo má editovatelné položky, kam si lze zaznačit další informace. Mapu lze ovládat jak myší, tak i klávesnicí. Přibližovat/oddalovat se lze pomocí kolečka, pomocí posuvníku na mapě nebo klávesami plus a mínus. Pohybovat se pak lze šipkami klávesnice, myší nebo pomocí čtyř směrových šipek umístěných nad posuvníkem pro přiblížení/oddálení.



Obrázek 2: Prototyp uživatelského rozhraní

3.1.8 Uvítací obrazovka

Po načtení aplikace se zobrazí uvítací obrazovka, na které jsou uživateli nabídnuty předchozí uložené trasy. Každá trasa je identifikovaná názvem, který zadá uživatel při vytvoření, a datem vytvoření. Součástí je tlačítko a pole pro text, do kterého se zadá název trasy, a stiskem tlačítka se přejde do aplikace.

Zde také dochází k upozornění uživatele, zda je jeho prohlížeč podporován a zda je povoleno ukládání souborů cookies. Soubory cookies jsou nutné pro uložení tras, respektive pro uložení uživatelova identifikačního čísla, podle kterého se trasy dohledají v databázi.

Uvítací obrazovka je vlastně modální dialogové okno obsahující informační a varovné zprávy a tabulku s uloženými trasami. Na pozadí mezitím dochází k načítaní aplikace. Po zvolení uložené trasy nebo vytvoření nové, je okno zavřeno a je možná pracovat s aplikací. Z této obrazovky je také možné smazat předchozí uložené trasy.

3.1.9 Hlavní ovládací panel

Celá aplikace se ovládá z panelu na spodní hraně obrazovky, který zpřístupňuje veškerou funkčnost uživateli. Panel je rozdělen do kategorií a každá kategorie je reprezentována ikonou. Posunem myši nad ikonu se zobrazí popisek, kliknutím se rozbalí daná kategorie panelu a odhalí možné akce. Stiskem tlačítka je aktivována akce spojená s tímto tlačítkem. V případě některých akcí je po dokončení (např. vložení bodu itineráře) automaticky ukončena, u jiných akcí je nutno explicitně ukončit práci stiskem stejného nebo libovolného tlačítka. Vždy aktivní může být jen jedna kategorie, předchozí aktivní kategorie a ovládací prvky jsou automaticky deaktivovány. Pro co možná největší komfort navigace v panelu je možné využít kombinaci myši a klávesnice (klávesy Tab a Enter). Právě aktivní funkce, respektive její tlačítko je zvýrazněno.

3.1.10 Plovoucí nabídka

Dalším prvkem rozhraní, který zvyšuje interaktivitu aplikace a uživateli zobrazuje popisek objektu a s ním spojené akce, je plovoucí nabídka. Smyslem toho prvku je dát vizuálně uživateli najevo, že prvky nacházející se pod kurzorem nesou další informaci a také je zobrazí bez nutnosti cílené snahy o zjištění těchto dat. Nabídka se objeví u objektu, nad kterým se nachází myš a který nese uživatelem definovaná data. Po ztrátě zaměření (posunu myši mimo objekt) je zavřena. Nabídka je vždy tvořena popiskem a tlačítkem akcí. Akce se pak liší dle typu objektu, pro který byla nabídka zobrazena.

Mezi podporované objekty patří položky itineráře, značky, fotky a měření vzdálenosti. V případě itineráře se zobrazí titulek položky a tlačítko pro otevření itineráře. Stiskem tlačítka se zobrazí okno itineráře a aktivují se všechny funkce, jako by došlo k otevření itineráře z panelu.

U značek se po stisku tlačítka zobrazí dialog pro uložení odkazu na web, respektive bublina pro vložení textového popisku. Veškeré popisky a odkazy lze pak kdykoliv změnit pomocí této nabídky nebo vybráním příslušné akce z panelu a vybráním objektu.

Pro fotky tato nabídka zobrazí název fotografie a akci pro zobrazení většího náhledu fotografie. Kliknutím na fotku se otevře vyskakovací okno vedoucí na stránku fotky na webu Panoramio.com.

3.1.11 Itinerář

Itinerář je tvořen nemodálním dialogovým oknem, které lze libovolně přemisťovat v okně prohlížeče. Obsahuje číslovaný seznam míst, které by rád uživatel navštívil. Každá položka seznamu má upravitelný titulek a text, tlačítko s polohou bodu v mapě, číslo položky a tlačítko pro vymazání. Nové místo lze přidat stisknutím tlačítka „Nový“ v hlavičce dialogu. Stiskem tohoto tlačítka se vytvoří nové místo v seznamu a uživatel je změnou kurzoru myši vyzván k vložení bodu. Kliknutím do mapy je vytvořen bod s pořadovým číslem a je vložena jeho souřadnice k položce v seznamu. Po najetí myši na bod na mapě se zobrazí, pomocí výše zmiňované plovoucí nabídky, titulek vložený v okně itineráře. Stisknutím souřadnic položky itineráře se mapa vystředí na polohu místa.

Se zobrazením itineráře jsou současně aktivovány ovládací prvky umožnující posun bodů. Pro posun bodu stačí přejít s kurzorem myši nad tento bod, stisknout pravé tlačítko a táhnout. Stiskem tlačítka myši dojde taktéž ke změně kurzoru, aby byl uživatel informován o možné akci.

3.1.12 Měření vzdálenosti

Měření probíhá kreslením cesty (křivky) do mapy. Aktuální vzdálenost se zobrazuje v plovoucí nabídce, která je popsána výše. Pro jednoduchost a intuitivnost celý proces měření dává uživateli okamžitou zpětnou vazbu – při pohybu myši je okamžitě přepočítána vzdálenost. Stiskem pravého tlačítka myši dojde ke vložení bodu do mapy s popiskem vzdálenosti k počátku. Měření je ukončeno dvojklikem. Při zahájení nového měření nebo aktivací jiné funkce dojde ke smazaní předchozího měření.

3.1.13 Kreslící funkce

Pro zakreslení ploch, čar či bodů do mapy je sada kreslících funkcí. Otevřením kategorie „Kreslení“ na ovládacím panelu aplikace se zobrazí nástroje na kreslení a také barevná paleta. Všechny ovládací prvky se chovají stejně jako ve zbytku aplikace – jen s tím rozdílem, že před zahájením nebo i během kreslení uživatel volí barvu. Změna barvy se ihned projeví i na kurzoru myši, takže lze v kterémkoliv okamžiku jednoznačně určit aktuální barvu bez nutnosti nahlédnutí do panelu.

3.1.14 Zobrazení fotek, přepínaní mapových podkladů

Zobrazování fotek se povolí stisknutím příslušného tlačítka v panelu v kategorii „Fotky“. Fotky se pak načítají po skončení posunu nebo přiblížení, respektive oddálení mapy. V mapě se zobrazují pouze miniatury fotek, po najetí náhled a název v plovoucí nabídce. Větší náhled se zobrazí rozbalením nabídky a na stránku fotky (s plnou velikostí) lze přejít kliknutím na náhled v plovoucí nabídce.

Přepínaní mapových podkladů a zobrazení/schování dalších vrstev lze provést z nabídky u levého horního rohu obrazovky. V rozbalené nabídce se nachází všechny vrstvy a pomocí zaškrtávacího tlačítka lze zapnou, respektive vypnout viditelnost vrstev.

4 Implementace

Aplikace používá architektury klient-server. Klient je webová aplikace, zobrazuje mapový podklad, zpracovává události. Komunikuje se serverem pomocí HTTP protokolu především asynchroním přenosem (AJAX). Výměna dat probíhá ve formátu JSON, respektive JSONP v případě načítání fotek. Převážná část logiky je oddělena od uživatelského rozhraní. Pouze některé vizuální části, jako plovoucí nabídka a částečně itinerář, nemají přísnou separaci logiky a rozhraní.

Server zajišťuje persistenci dat. Je implementován v jazyce PHP. Na serveru běží MySQL (MariaDB) databáze uchovávající data o uživatelích a jejich trasy.

Vývoj probíhal ve dvou fázích, v první byl vytvořen prototyp rozhraní se základní funkčností. Na základě tohoto prototypu byla pak vytvořena aplikace nová, u které probíhal následný inkrementální vývoj. Nejdříve byla vytvořena klientská část. V momentě, kdy se dostala do fáze, v níž byly implementovány prvky, které se mají uložit, byl vytvořen server. Jako první byla vytvořena a otestována databáze. Poté byl vytvořen skript, který byl postupně vyvíjen společně s klientským modulem pro ukládání. Na základě toho skriptu byly vytvořeny skripty ostatní.

4.1 Použité knihovny

Pro implementaci výše popsaného rozhraní byla použita knihovna jQuery UI. Tato knihovna bylo vybrána společně s jQuery a OpenLayers pro svou jednoduchost, kvalitní sadu elementu uživatelského rozhraní, dobrou dokumentaci a také kvůli návaznosti na knihovnu jQuery, která je použita v logice aplikace. Použity byly zejména dialogová okna (v modální i nemodální variaci), tlačítka a animace.

Knihovna jQuery byla zvoleno pro snazší práci s asynchronními přenosy, selektory a bohatou dokumentaci. Pro práci se soubory cookies se používá doplněk jquery-cookies. Tyto knihovny jsou uvolněny pod svobodnou licencí MIT. Doplněk jquery-cookie je pak duálně licencován pod svobodnými licencemi MIT/GPL.

Knihovna OpenLayers, která zajišťuje práci s mapou, je licencována pod BSD licenci. V projektu jsou také použity ikony Kfaenza[2]. Ty jsou uvolněny pod licencí GPL.

Všechny knihovny jsou umístěny v adresáři `oolMap/lib/libs/` a ikony a styly knihoven v adresářích `oolMap/theme/default/` a `oolMap/theme/toolbar/`.

2 http://kde-look.org/content/show.php?content=143890 [citováno dne 2013-05-06]

4.2 Klientská aplikace

Klient je tvořen jako interaktivní webová aplikace využívající technologie AJAX pro komunikaci se serverem. Veškeré datové přenosy jsou realizovány pomocí funkcí jQuery `ajax`, `post a get`. Uživatelské rozhraní je implementováno pomocí HTML, JavaScriptu a CSS za pomoci knihoven jQuery a jQUery UI. Logika aplikace kombinací OpenLayers a jQuery.

Jádro aplikace – mapa – je vykreslena pomocí knihovny OpenLayers. Aplikace reaguje na DOM události (onhover, atd) a události mapy definované v OpenLayers (například událost mapmove) vyvolané činností uživatele. Jedná se tedy o událostmi řízený systém.

V aplikace se využívá objektová orientace JavaScriptu. Některé části však používají procedurální přístup. Mezi nejdůležitější objekty patří objekt `oolUser`, který slouží k uchovávání informací o uživateli a k ukládání a načítání dat. Procedurálně implementovaná je například plovoucí nabídka.

Pro lepší logické členění uživatelských dat je vytvořeno několik vrstev. Kreslení, itinerář, značky, měření a fotky mají vlastní vrstvu. Jednotlivé vrstvy lze kdykoliv skrýt nebo schovat, logicky rozdělují aplikaci na menší součásti a zjednodušují proces ukládání. Zjednodušují také implementaci rozdílného chování pro různé objekty. Ve všech případech se jedná o vrstvy třídy `OpenLayers.Layer.Vector`. Všechny objekty, přes různý vzhled a chování, jsou tedy stejného typu. Rozdílného chování je dosaženo různými styly těchto vrstev. Každý ovládací prvek pracující nad více vrstvami totiž převezme styl z vrstvy vybraného objektu a obslužné funkce implementují různou funkčnost pro každou vrstvu.

4.2.1 Objekt oolUser

Pro správu uchování informací o uživateli slouží objekt `oolUser`. Zapouzdřuje identifikátor uživatele a metody pro čtení/získání tohoto identifikátoru, metody pro získání uložených dat a metody pro ukládání.

Uživatelský identifikátor je číslo jednoznačně identifikující každého uživatele. Na straně klienta se uchovává v souborech cookies a za běhu aplikace i v tomto objektu. Identifikátor je nutný pro načtení uložených tras. Z tohoto důvodu je vyžadována podpora cookies ze strany prohlížeče. Pokud podporovány nejsou, nebo je jejich ukládání zakázáno, je na to uživatel upozorněn na uvítací obrazovce. Při načítaní aplikace se metodou `getUid` objektu `user` získá identifikátor. Pokud

existuje, je přečten ze souboru cookies. Pokud ne, tak se získá pomocí metody GET protokolu HTTP voláním PHP skriptu `newUID.php`. Skript vytvoří v databázi nového uživatele a vygeneruje heslo. Tyto hodnoty vrátí volajícímu objektu. Heslo se používá pro autentizaci uživatele pro další požadavky. Tento a další skripty jsou podrobně popsány v kapitole 4.3.2.

Metoda `getSessions` získá všechny uložené trasy pro uživatele. Každá trasa má svůj jedinečný identifikátor (sid). Trasy jsou pak zobrazeny na uvítací obrazovce, odkud lze trasy i mazat. Mazání probíhá voláním metody `deleteSession`, která pošle asynchronní požadavek metodou POST na skript `deleteSession.php`, který smaže z databáze danou trasu.

4.2.2 Persistence dat

Persistenci dat ze strany klienta je zajišťovaná pomocí metod objektu `user` a jsou to `saveFeatures`, `saveFeaturesFromLayer`, `loadFeatures`, `loadFeaturesToLayer` a `deleteFeature`. Metody `saveFeatures` a `loadFeatures` přijímají jako parametr pole vrstev, ze kterých se mají uložit uživatelem vytvořené objekty (kresby, položky itineráře a značky) a následně volají pro každou vrstvu z pole metodu `saveFeaturesFromLayer`, respektive `loadFeaturesToLayer`. V první zmíněné metodě dojde k serializaci dat. Geometrie tělesa se uloží do formátu WKT (viz kapitola 4.3.2) a extrahují se atributy. Z geometrie a atributů se vytvoří objekt a uloží do pole. Tato data jsou převedena do formátu JSON a poslána na server metodou POST. Na serveru jsou data ve skriptu `saveFeatures.php` uložena do databáze. V případě, že se znovu ukládá již existují trasa, jsou data jen aktualizována. Pro všechny typy objektů se data posílají stejným způsobem, pro rozeznání vrstvy (a tím pádem typu dat) se přidává k požadavku parametr `layer`. Jedná se o celočíselnou `hodnotu` identifikující zdrojovou vrstvu. V asociativním poli se ukládá jméno vrstvy a jeho přidružené číslo. Při ukládání dat z vrstvy se pak název vrstvy použije jako klíč do pole. Získaná hodnota je číslo vrstvy. Na serveru je objekt napodobující chování výčtového typu, který mapuje číslo vrstvy na tabulku v databázi.

Načítání dat probíhá obdobně. Metodou GET se získají data trasy opět ve formátu JSON, pro každou vrstvu zvlášť. Deserializují se, z dat se vytvoří nové objekty, nastaví se jim atributy a vloží se do vrstvy. Pro všechny vrstvy se vytvoří nový objekt s geometrií načtenou z WKT, pro kresby se nastaví styl, pro itinerář pořadové číslo, popisek a text a pro značky typ (text/odkaz), titulek a text, respektive odkaz. V případě itineráře se ještě vytvoří položky v okně itineráře pomocí jeho metod. Více podrobností o ukládání dat z pohledu databáze a ukládaných hodnot je v kapitole 4.3.1.

4.2.3 Inicializace aplikace

Inicializace aplikace je svázána s událostí DOM onload objektu `document`. Poté, co je celá webová stránka načtena, je zavolána funkce `initMap`. V této funkci dochází k inicializaci mapy a všech ostatních součástí aplikace – ovládacího panelu a jeho modulů, vytvoření ovládacích prvků mapy, vytvoření mapy, vytvoření všech vrstev, získání uživatelova identifikátoru, inicializace sdílení, navázání akcí na tlačítka atp.

Po zavolání této funkce jsou vytvořeny globální objekty `user` a `share`. Objekt user získá uživatelův identifikátor, objekt `share` zpracuje parametry adresy.

Poté je vytvořena mapa a všechny vrstvy, je naplněn objekt `uniControls` a jsou zavolány funkce pro inicializaci ovládacího panelu. `UniControls` je globální objekt s referencemi na všechny vrstvy, na styly vrstev a na ovládací prvek mapy reagující na událost hover (`uniControls.Hover`). Používá se při implementaci plovoucího menu a jako výchozí ovládací prvek mapy.

Následuje inicializace panelů měření, kreslení, značek, okna itineráře a zobrazování fotek. V případě prvních 3 panelů se do hlavního ovládacího panelu vytvoří panely jako instance třídy `OpenLayers.Control.Panel`. Do panelu se pak přidají jednotlivé funkce a vytvoří se pro ně tlačítko. Inicializací itineráře se vytvoří HTML kód reprezentující itinerář a k ovládacím tlačítkům se zaregistrují příslušné ovládací funkce. U zobrazování fotek dojde k navázání události onclick tlačítka s funkcí, která spustí nebo vypne načítaní fotek.

Po načtení všech panelů se ověřuje, zda se uživatel snaží načíst sdílenou trasu. Pokud tomu tak je, ověří se oprávnění, případně se nezobrazí ovládací panel, a dojde načtení dat trasy. Nejedná-li se o sdílenou trasu, je uživateli zobrazena uvítací obrazovka. O to se stará objekt `share`.

Uvítací obrazovka je modální okno, prvek ze sady komponent knihovny jQuery UI, zobrazuje se voláním funkce `oolWelcomeScreen`, ve které se zkontrolují požadavky na prohlížeč (podpora cookies). Pokud prohlížeč nevyhovuje požadavkům, do uvítacího okna se připojí varovná hláška. Uživatel je upozorněn na nesplněný požadavek, ale je mu umožněno pokračovat. Dále se načtou uložené trasy, pokud existují. Funkcí `addRow` se přidají do tabulky řádky s trasami. Pomocí selektoru z knihovny jQuery se vytvoří pro všechny HTML elementy s třídou oolGoToSession tlačítka pro přechod na uloženou trasu. Obdobně se vytvoří tlačítka pro smazání trasy. Na konec je přidáno

tlačítko pro vytvoření nové trasy a okno je zobrazeno. Mezitím, než si uživatel rozmyslí akci, se načítají mapové dlaždice.

4.2.4 Ovládací panel

Hlavní ovládací panel je tvořen pouze HTML elementy , obrázky a vygenerovanými tlačítky. Každá položka v panelu má svoji ikonu a rozbalovací část, ve které se zobrazují ovládací tlačítka. Po načtení stránky se pomocí funkce `animateToolBar5` panelu přidá funkčnost. Ikony mají třídu oolToolBarItemImg, pro kterou je pomocí selektoru jQuery vytvořena obsluhující funkce reagující na klik myši. Po stisknutí tlačítka dojde k vyvolání obslužné funkce, která manipulací s CSS vlastnostmi schová poslední rozbalenou nabídku a obdobně zobrazí nabídku stisknuté ikony. Pro uchování reference na poslední použitou nabídku se používá globální proměnná, do které se po stisku tlačítka vloží aktuálně rozbalená nabídka.

Při stisku ovládacího prvku také dochází k volaní funkce `activeControlsRun`, která deaktivuje všechny ostatní prvky, aby nedocházelo ke kolizím (a případně kombinacím – mohlo by například docházet ke kreslení bodu zároveň s čárou).

Každá vrstva má svoji sadu ovládacích prvků. Existuje jeden globální ovládací prvek, který je aktivní po inicializaci aplikace, dokud nepřevezme řízení prvek z některé vrstvy. Tento prvek umožňuje funkci plovoucí nabídky. Ostatní ovládací prvky jsou přiřazeny vždy k jedné vrstvě, respektive k ovládacímu panelu vrstvy. Některé, jako například kreslení nebo vkládání poznámek, pracují pouze pro tu vrstvu, ke které jsou přiřazeny, jiné fungují nad více vrstvami. Vzhledem k omezení OpenLayers, kdy v případě přidání jedné instance ovládacího prvku do více panelů funguje pouze naposled přidaný, je pro zajištění konzistence chování a k omezení duplikace kódu vytváření ovládacích prvků přesunuto do funkcí `createFakeUniDelete`, `createFakeUniHover`. Zavoláním funkce se vytvoří nový ovládací prvek, všechny mají stejné chování bez duplikace kódu, ale jedná se o jiné objekty. Případem takového prvku může být „mazání“ nebo „pohyb“, který zároveň zobrazuje plovoucí nabídku, v případě, kdy není aktivní globální ovládací prvek.

4.2.5 Plovoucí nabídka

Plovoucí menu je hlavním způsobem, kterým se zobrazují data objektů v mapě. Při získání zaměření objektu, tedy při najetí myši nad objekt, se zavolá funkce `uniHoveOnSelect`, která zobrazí nabídku. Při ztrátě zaměření se nabídka schová voláním funkce `uniHoverOnUnselect`, která

vyprázdní obsah HTML elementů reprezentujících nabídku, a manipulací CSS vlastností ji schová. Zobrazení nabídky je spojeno s ovládacími prvky z ovládacího panelu a jedním globálním prvkem z objektu `uniControls`. Jedná se o objekty třídy `OpenLayers.Control.SelectFeature` vytvořené funkcí `createFakeUniHover`.

Plovoucí nabídka je vlastně skupinou HTML elementů , do kterých se dle kontextu generuje obsah. Všechny objekty mapy mají titulek a akci. Dle typu objektu se akce a rozbalovací část nabídky liší. Pro práci s nabídkou existuje několik funkcí, např. `setToolTipPosition` a `setToolTipText`.

Po zavolání funkce `uniHoveOnSelect` se z objektu, pro který byla volána, zjistí jeho pozice a vrstva, na které je. Podle vrstvy, tedy typu, se generuje další HTML kód. Pro značky mohou nastat dva případy – textová značka nebo odkaz. Pro typ odkaz se do titulku nabídky vytvoří odkaz, element <a>, pomocí metody jQuery `append` se k nabídce připojí formulář pro vložení adresy odkazu. V případě textu se vytvoří vyskakovací okno. Pro okno se vygeneruje HTML kód s oblastí pro text a připojí se k oknu.

Pokud byl vybrán objekt položky itineráře, zobrazí se jeho titulek, akce otevře itinerář a aktivuje jeho ovládací prvky. Pro fotografii se zobrazí popisek a přidružená akce vygeneruje okno s větším náhledem fotky, odkazem na web Panoramio.com a jménem autora.

Po vytvoření kódu se nabídka zobrazí na pozici objektu voláním funkce `setToolTipPosition`.

4.2.6 Itinerář

Itinerář je nemodální dialogové okno, kterému lze měnit pozici a velikost. Je implementován jako objekt a inicializační funkce, která vytvoří ovládací prvky (`OpenLayers.Control`) specifické pro itinerář, vytvoří nový objekt itineráře a provede jeho inicializaci.

Nové položky se přidávají stiskem tlačítka „Nový“, které volá metodu objektu itineráře `newItem`. Aby nedocházelo k mnohonásobnému stisknutí tlačítka před vložením bodu, obsahuje jednoduchý zámek. Při spuštění metody `newItem` dojde k „zamčení itineráře“, funkce okamžitě skončí, když je itinerář zamčen. Pokud není, do dialogového okna se vygeneruje HTML kód pro novou položku, vygeneruje se vstupní pole, pořadové číslo, textové pole a tlačítko pro vymazání. K tlačítku se připojí funkce, která tento generovaný kód zase odstraní a odstraní bod z mapy. Součástí položky je i tlačítko se souřadnicemi bodu, po jehož zmáčknutí se mapa vystředí na tento bod. Po

vygenerování HTML se aktivuje ovládací prvek třídy `OpenLayers.Control.DrawFeature`, pomocí kterého uživatel vloží do mapy bod. Položce se nastaví pozice a itinerář se uvolní.

Naplnění itineráře uloženými daty probíhá obdobně. Pomocí metody `newItem` se vytvoří nová položka. Pořadové číslo položky, titulek a text se předají jako parametry této metody. Do mapy se vloží bod, který je na serveru uložen ve formátu WKT, funkcí `setItemPos` se nastaví souřadnice bodu položce v itineráři.

Itinerář má přidružené ovládací prvky, které se automaticky aktivují při zobrazení okna. Jedná se o ovládání pro změnu pozice a zobrazování plovoucí nabídky.

4.2.7 Značky, Kreslení

V obou případech se z technického hlediska jedná o vektorové objekty uložené ve vektorových vrstvách `OpenLayers.Layer.Vector`. Pro logické členění aplikace se však značky i kreslené objekty mají vlastní vrstvu. Usnadňuje to implementaci ukládání, nastavení stylů i rozpoznávání objektů v plovoucí nabídce.

Značky, na rozdíl od kreslených objektů, mají ikonu a nesou uživatelem definovaná data, která jsou uložena v objektu atributů. Mezi typem značky (textem nebo odkazem) se rozlišuje atributem `tType`, který může nabývat hodnot `tText` nebo `tUrl`. Atribut `tIcon` je odvozen od typu značky a obsahuje adresu ikony, která se má používá v mapě pro vizuální rozlišení značky. Dalšími atributy jsou `tTitle`, nesoucí nadpis, respektive název odkazu a `tValue` nesoucí textový popisek, nebo adresu odkazu. Hodnoty atributů lze upravit pomocí plovoucího ovládání nebo z hlavního ovládacího panelu. Panel obsahuje ovládací prvky pro vytvoření nové značky, pro přesun již existující značky, mazání a pro úpravu textu značky. Poslední tři ovládáni jsou implementována jako objekty třídy `OpenLayers.Control.SelectFeature`. Liší se chováním a obslužnými funkcemi. Třída `SelectFeature` se stará pouze o vybrání prvku, ať už kliknutím nebo najetím kurzorem myši, vlastní funkčnost je implementovaná v obslužných funkcích volaných pro každou událost. Mazání funguje jako univerzální, maže objekty ze všech vrstev. Knihovna OpenLayers však neumožňuje mít jednu instanci ovládacího prvku ve více panelech, a proto se pomocí funkce `createFakeUniDelete` vytvoří nová instance pro každou vrstvu. Tímto se vytvoří konzistentní chování napříč všemi vrstvami, bez nutnosti duplikace kódu. Stejným způsobem je řešeno ovládání plovoucí nabídky.

Text značky se upravuje pomocí vyskakovacího okna, které lze vyvolat buď z plovoucí nabídky nebo aktivací ovládacího prvku „upravit text“ v panelu. V obou případech se volá funkce `popupCreate`, která vytvoří nové vyskakovací okno. Okno je objektem třídy `OpenLayers.Popup.FramedCloud`. Funkcí `popupHtml` se vygeneruje HTML kód obsahu okna, vytvoří se ukládací funkce, která po ztrátě zaměření textového pole uloží text do atributů.

Pro značku typu odkaz se generuje kód do plovoucí nabídky. Do titulku se vloží odkaz (element <a>). Vytvoří se formulář obsahující pole pro text odkazu, adresu a tlačítko pro uložení. Po uložení se adresa vloží do atributů vybrané značky.

Pro správu barev (stylů) pro kreslená geometrická tělesa byl vytvořen objekt `oolColorPicker`. Po vytvoření do panelu přidá tlačítka pro změnu stylu kreslení. Vektorové objekty přejímají styl od vrstvy, do které byly vloženy. Z toho důvodu je jim nutno po přidání vložit nový objekt stylu. `oolColorPicker` obsahuje čtyři objekty stylů (pro čtyři volitelné barvy z panelu). Výběrem barvy se do atributu `currentStyle` vloží objekt zvolené barvy a po přidání tělesa do vrstvy se mu nastaví tento styl. Stejným atributem se nastavuje styl grafických prvků u kurzoru myši během kreslení.

Vlastní kreslení je realizováno pomocí ovládacího objektu třídy `OpenLayers.Control.DrawFeature`, kde je jako parametr předána třída požadovaného tělesa, parametry této třídy (např. styl, který se má požít pro vykreslování, který je však po vložení do vrstvy přepsán na výchozí styl vrstvy) a funkce, volaná po vložení nakresleného tělesa do vrstvy. Po přidání tělesa se mu nastaví správný styl, atributy a je nutné překreslit vrstvu.

Ovládání pro mazání, plovoucí nabídku a posun tělesa je implementován obdobně jako u značek.

4.2.8 Načítaní fotografií

Fotografie se načítají ze služby Panoramio.com pomocí asynchronních přenosů ve formátu JSONP, protože API Panoramia se nachází v jiné doméně, nelze použít jen JSON a pro přenos je potřeba nastavit správné atributy (crossDomain: true) a očekávaný formát dat.

Zapíná se stiskem zaškrtávacího tlačítka v příslušné kategorii v ovládacím panelu. Zaškrtnutím tlačítka dojde k registraci funkce `getPhotosPanoramio` k události mapy moveend. Jedná se událost z knihovny OpenLayers, je vyvolána po skončení posunu nebo přiblížení, respektive oddálení mapy. Opětovné stisknutí tlačítka tuto událost odregistruje a dojde k odstranění náhledů z mapy.

Vlastní načítaní fotek probíhá ve funkci `getPhotosPanoramio`. Po zavolání této funkce se zjistí zobrazená oblast mapy, obdélníková oblast ohraničující mapu, sestaví se dotaz podle API Panoramio.com a získají data.

Přijatá data jsou ve formátu JSONP, jedná se o pole fotografií. Cyklem se projde celé pole. Pro každý objekt v poli se vytvoří vektorový objekt třídy `OpenLayers.Feature.Vector`, přidají se mu atributy jako adresa fotky, jméno autora a jiné. Fotografie se umisťují do vlastní vrstvy jako náhledy a jsou opatřeny rámečkem. Větší náhled fotky je možno zobrazit pomocí plovoucího menu.

4.2.9 Sdílení

Sdílení je realizováno v objektu `share`. Metodou `getParam` se zpracují parametry adresy, která byla zadána do prohlížeče (DOM objekt `window.location.search`). Kompletně celé zpracování leží na straně klienta, podle zjištěných dat se různě konfiguruje aplikace bez zásahu serveru. Zpracování je realizováno pomocí regulárních výrazů. Parametry Adresy jsou vždy ve tvaru `share=xxx&yyyyyy`, kde `xxx` je číslo trasy, kterou chceme sdílet, a `yyyyy` je nepovinný parametr, heslo vygenerované serverem při vytváření trasy opravňující k (lokálním) úpravám a uložení trasy. Celá adresa pak vypadá následovně: http://osmmap.8u.cz/oolMapV2/?share=11&1366742432.

4.3 Server

Serverová část aplikace zajišťuje jednoduchou autentizaci a správu uživatelů a především persistenci dat. Pro správnou činnost serveru je potřeba webový server s podporou PHP5, např Apache2, a databáze MySQL/MariaDB. MariaDB je zpětně kompatibilní větev databáze MySQL vyvíjená původními autory MySQL a komunitou po zakoupení původní databáze společností Oracle.

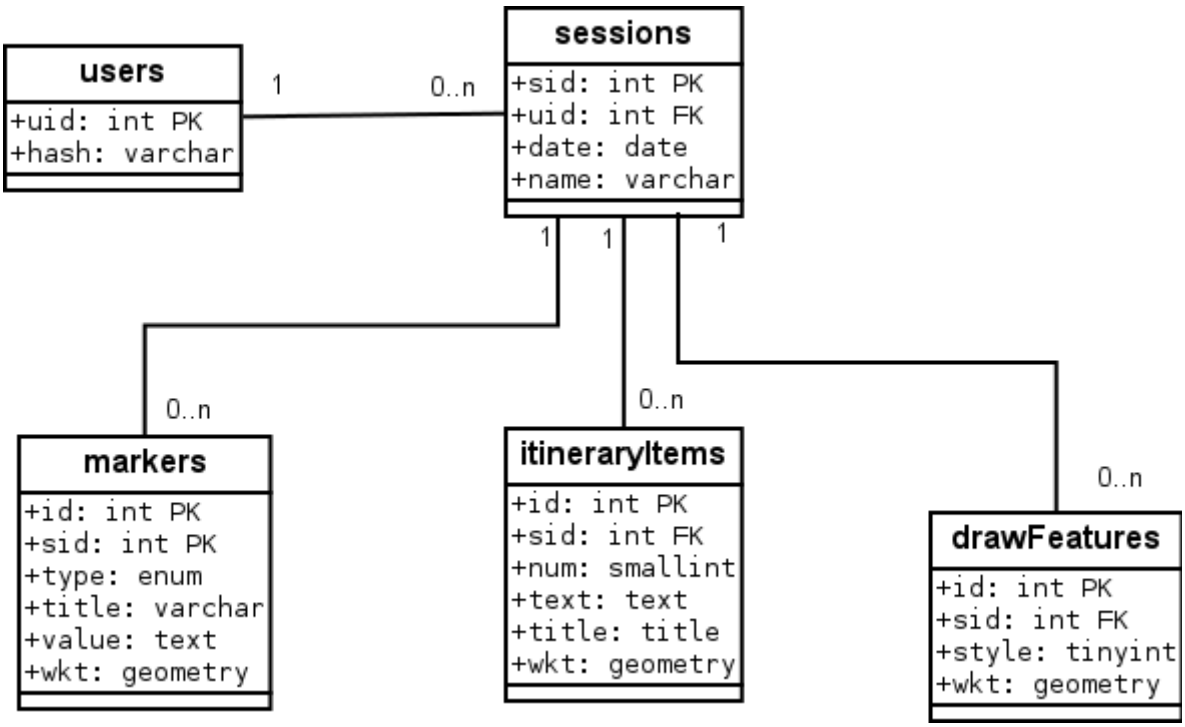
Pro připojení k databázi se používá MySQLi, jedná se o rozšíření pro práci s databází MySQL 4.1.3 a vyšší. Je součástí PHP od verze PHP5. Podporuje nové vlastnosti databáze a zároveň objektový i procedurální styl programování a mimo jiné i předem připravené SQL dotazy. Takto vytvořené dotazy jsou výkonnější, odolné vůči SQL injektáži a oddělují logiku a data v dotazu. Dotaz je připraven předem, při vykonání se pouze doplní parametry, nelze tedy ovlivnit logiku dotazu.

Komunikace mezi klientem a serverem probíhá pomocí formátu JSON určeného pro přenos dat. Jedná se o textový, člověkem čitelný formát nezávislý na platformě a použitém jazyku. Je odvozen od způsobu zápisu objektů a polí v JavaScriptu. V PHP skriptech se vstupní data převedou pomocí funkce

`json_decode` na pole objektů, které se dál zpracovává. Výstupní data jsou zpět předána výpisem kódovaných dat na standardní vstup. Kódovaní je provedeno funkcí `json_encode` a výpis pomocí konstruktu `echo`. Pro správné rozpoznání dat na straně klienta je potřeba pro odeslaná data připojit hlavičku s typem dat, funkcí `header('Content-type: application/json')`.

4.3.1 Schéma databáze

V databázi je potřeba ukládat data o uživatelích, jejich trasách, a všechny kresby, poznámky a položky itineráře. Protože požadavky na uložená dat jsou u těchto tří kategorií dost rozdílné, byla vytvořena jedna tabulka pro každou kategorii. Schéma databáze je na obrázku 3.



Obrázek 3: Schéma databáze

O uživatelích ukládáme pouze uživatelský identifikátor (uid) a heslo. Identifikátor je číslo jednoznačně identifikující každého uživatele. Každému uživateli je také vygenerováno a uloženo heslo (hash). Tyto informace se uchovávají v tabulce user, kde uživatelský identifikátor slouží jako primární klíč.

Dále ukládáme trasy. Jelikož může mít každý uživatel nula až N tras, vazba mezi uživateli a trasami je 0 ku N, pro trasy je vytvořena nová tabulka sessions. Jako primární klíč slouží pole

session id (sid) jednoznačně identifikující trasu. Dále je potřeba uchovávat název a datum vytvoření. Pro vytvoření vazby mezi uživatelem a trasou obsahuje tabulka sessions identifikátor uživatele (uid) jako cizí klíč.

Kresby, poznámky a itinerář mají společné pouze 3 položky – identifikátor, geometrii (geometrický obrazec – čára, bod, polygon) a id trasy (sid). Kromě těchto položek mají několik různých požadavků. Proto má každý typ svoji tabulku. Tyto tabulku jsou spojeny vazbou 0 ku N k tabulce tras. Id trasy tedy slouží jako cizí klíč. Tabulka ukládající kresby drawFeatures kromě vlastního identifikátoru, id trasy a geometrie, uložené jako datový typ `Geometry`, nese pouze číslo stylu, které je při načítání klientskou aplikací převedeno na objekt stylu OpenLayers. U položek itineráře se uchovává titulek a text jako typ `Text` v tabulce itineraryItems. Značky potřebují uchovat informaci o typu (text/odkaz), titulku a textu (v případě typu odkaz pak adresa odkazu). Typ značky je uložen jako datový typ `enum`, titulek a text jako Text a jsou uloženy v tabulce markers.

Geometrie se ukládá v interním formátu MySQL `Geomertry`, přenos však probíhá v textovém formátu WKT – Well Known Text. Jedná se o textový značkovací jazyk pro popis geometrických obrazců.

4.3.2 PHP skripty

`Resources.php` – obsahuje pomocné třídy pro práci s databází a třídu napodobující výčtový typ, známý z jiných jazyků - třídu `EnumLayers`.

Protože databáze obsahuje 3 různé tabulky pro ukládání uživatelem vytvořeních dat (itinerář, kresby a značky), a metoda klienta pro ukládání nerozlišuje zdrojovou vrstvu, je potřeba určit, kam se mají data uložit. K tomu slouží celočíselný parametr požadavku `layer`. Třída `EnumLayers` obsahuje konstanty, pomocí kterých se číselné hodnoty mapují na typ dat.

Skript také obsahuje třídu `Feature` reprezentující ukládaný objekt. Jedná se obecný objekt, který je pomocí dědičnosti přizpůsoben pro položku itineráře, kresbu nebo značku. Tento skript se pak připojuje k ostatním skriptům.

Nové identifikátor uživatele se získává pomocí skriptu `newUID.php`. Skript se pak připojí k databázi, vytvoří heslo MD5 součtem konkatenace aktuálního času a náhodného čísla. a vytvoří nového uživatele vložením řádku do tabulky users. Primární klíč, který je současně identifikátorem uživatele, je automaticky vytvořen databází a je uložen jako atribut `insert_id` v objektu připojení k databázi. Z identifikátoru a hesla se vytvoří asociativní pole a zakóduje do formátu JSON funkcí

`json_encode`. Takto zakódované pole je pak na straně klienta přijato jako JavaScriptový objekt s atributy `uid` a `hash`.

Pro uložení nové trasy je potřeba vytvořit nový řádek v tabulce sessions s identifikátorem, vlastníkem, názvem a datem ve skriptu `newSession.php`. Klientská aplikace pomocí metody GET protokolu HTTP zavolá tento skript. Jako parametry předá identifikátor uživatele a jména trasy. Skript se připojí k databázi, vloží řádek do tabulky a vrátí identifikátor trasy, který se získá také z objektu připojení k databázi a zakóduje.

Všechny uložené trasy se získají skriptem `getSessions.php`. Z databáze se vyberou všechny řádky z tabulky sessions, pro které se rovná pole uid (identifikátor uživatele) s identifikátorem přijatým v požadavku. Z vybraných řádků se vytvoří pole objektů `session`. Objekt `session` má jako atributy identifikátor trasy, jméno a datum. Pole se zakóduje a předá.

Jednotlivé trasy lze smazat pomocí skriptu `deleteSession.php`. Pro smazání trasy (a všech k ní přiřazených dat) je potřeba předat identifikátor uživatele a trasy a heslo. Návratová hodnota je 0 v případě úspěšného provedení, pokud dojde k chybě, je vrácena 1. Mazaní probíhá následujícím SQL dotazem: `DELETE FROM `sessions` WHERE EXISTS ( SELECT users.uid FROM users WHERE users.hash = ? AND users.uid = ? AND sessions.uid = users.uid AND sessions.sid = ?)`. Smaže se ten řádek z tabulky sessions, pro který platí, že uživatel se zadaným id a heslem existuje a zároveň je zadaná trasa jeho. Otazníky jsou doplněny příslušnými hodnotami z požadavku GET.

Pro ukládání kreseb, itineráře a značek slouží skript `saveFeatures.php`. Klientská aplikace ukládá data po jednotlivých vrstvách. Pro každou vrstvu je poslán jeden dotaz typu POST s parametry – identifikátor uživatele a trasy, vrstva, vlastní data a parametr `share`. Parametr `share` udává, zda se jedná o uložení nesdílené trasy. Pro vlastní ukládání jsou vytvořeny funkce `saveDrawFeatures`, `saveItineraryFeatures` a `saveMarkers`. Přepínačem se podle vrstvy zavolá příslušná funkce. Všechny tři funkce jsou podobné, liší se ukládáním dat. Na vstupu mají tyto funkce pole dat. Cyklem se projdou všechny prvky pole a pro každý se provede vložení nebo úprava záznamu v tabulce. Každá kresba, položka itineráře nebo značka má identifikátor. Pokud tato položka nebyla ještě uložena, její identifikátor je záporný, pokud uložena byla, je kladný. O tom, zda se má upravit nebo vytvořit záznam nový, se rozhoduje na základě kladnosti identifikátoru. Pokud je kladný, již v databázi existuje, tudíž dojde k aktualizace záznamu, Pokud je záporný, dojde ke vložení záznamu nového. V případě, že se jedná o sdílení trasy, objekt má identifikátor, tudíž by došlo k aktualizování

dat, přestože je tento uživatel nemá. Proto v případě sdílení (parametr `share` má hodnotu pravda) dojde taktéž vložení nového záznamu.

Data uložená pro jednu vrstvu se získají skriptem `getFeatures.php`, který má vstupní parametry – identifikátor uživatele a trasy a číslo vrstvy. Podle čísla vrstvy se volá funkce, která načte data. Jsou to funkce `getDrawFeature`, `getItineraryFeature` a `getMarkers`. Princip těchto funkcí je podobný, liší se SQL dotazem a použitými objekty. Používají třídy definované v souboru `resources.php`. Načítají se pouze data jedné vrstvy (tabulky). Z databáze se vyberou všechny kresby, položky itineráře, respektive značky patřící k trase se zadaným identifikátorem. Cyklem se projdou všechny vybrané řádky, z jejich dat jsou vytvořeny objekty a ty jsou vloženy do pole a funkce se ukončí. Výsledné pole je překódováno do formátu JSON a předáno volající funkci klientské aplikace.

O mazání dat se stará skript `deleteFeature.php`. Tento skript smaže pouze jeden objekt zadaný jeho identifikátorem a vrstvou. Parametry jsou identifikátor uživatele a trasy, vrstvou a identifikátorem kresby, značky respektive položky itineráře. Podle zadané vrstvy se sestaví SQL dotaz vložením jména tabulky do předpřiraveného SQL kódu. Pomocí metody `bind_param` se vloží do dotazu hodnoty parametrů a provede se vymazání řádku. Skript vrátí 0 v případě úspěšného mazání, 1 v případě chyby.

Poslední skript `downloadKML.php` umožňuje uživateli stáhnou si itinerář právě prohlížené trasy k sobě do počítače ve formátu KML. KML (Keyhole Markup Language) je na XML založený jazyk pro přenos a publikaci geografických dat. Jedná se on standard konsorcia OGC. Na straně klienta dojde k serializaci vrstvy itineráře do formátu KML pomocí objektu třídy `OpenLayers.Format.KML`. Výsledný řetězec je poslán na sever metodou POST, který jej obratem vrátí jako textový soubor uživateli, který si jej může uložit. Přítomnost serveru je v tomto procesu nutná, JavaScript nedisponuje nástroji, jak uživateli dovolit stáhnout soubor.

5 Závěr

Cílem této bakalářské práce bylo vytvoření interaktivní mapové aplikace pro podporu pěší turistiky, přesněji řečeno, pro plánování výletů. V současné době není mnoho nástrojů, které umožňují uživatelsky přívětivým a přehledným způsobem naplánovat a poté sdílet logistiku výletu. To zahrnuje vše, od mapových podkladů po vlastní trasu a náplň výletu, až po doplňující informace a informace pro další účastníky výletu. Mapa je využita více jako prostředník komunikace, než jako pouhá statická informace.

V úvodu textu jsou uvedeny požadavky, které by tato aplikace měla splňovat. Veškerá funkčnost popsaná v těchto kapitolách byla implementována. Výsledná aplikace naplnila stanovené cíle, podařilo se vytvořit uživatelsky atraktivní interaktivní rozhraní, které intuitivně zpřístupňuje všechny funkce aplikace. Po funkční stránce se podařilo implementovat stanovené cíle s oddělením logiky aplikace od uživatelského rozhraní.

Jako největší překážkou v plné interaktivitě aplikace se ukázaly ovládací prvky mapy, které by měly pracovat nad více vrstvami současně a přitom být dostupné z každé kategorie ovládacího panelu, aby se uživatel nemusel neustále přepínat mezi různými kategoriemi. Toto v současné verzi knihovny OpenLayers není možné. Jako nejefektivnější řešení se ukázalo prosté nahrazení univerzálních prvků za prvky jednoúčelové, které univerzálnost imitují. Pro zachování konzistence v chování a pro zabránění duplikace kódu se tyto ovládací prvky tvoří funkcí – pro každou vrstvu je zachována stejná funkčnost bez duplikace kódu, pouze se vytváří více instancí téže třídy pro každou vrstvu.

Největším přínosem vytvořené aplikace je dostupnost nového nástroje, který usnadní plánování především pěších nebo cyklistických výletů, kdy je potřeba seznámit skupinu lidí s plánem. Mimo to se ukázalo, že lze aplikaci využít mimo turistiku, kdykoliv je potřeba zasadit a vizualizovat nějaké informace do kontextu mapy. Dalším přínosem by mohlo být rozšíření řad přispěvatelů do projektu OpenStreetMap – a tedy i zlepšení svobodných mapových dat.

Co se týče dalšího vývoje projektu, bylo by vhodné vyřešit problémy související s reálným nasazením do provozu. V první řadě se jedná o vykreslování a ukládání vlastních dlaždic, aby nedocházelo k zatěžovaní infrastruktury OpenStreetMap.org. Dále pak zabezpečit financování provozu serveru, například zobrazováním komerčních sdělení a poskytováním plných služeb pouze

přihlášeným uživatelům, a s tím spojenou důmyslnější autentizaci uživatelů a pravděpodobně také nastavení limitů pro ukládání dat.

Po vyřešení těchto záležitostí by mohlo dojít k doplnění funkčnosti – např export tras do jiných formátu jako například gpx pro použití v přenosných navigačních zařízeních. Nebo například rozšířená podpora pro sdílení (vkládání interaktivní mapy do jiných stránek), podpora animací pro prezentační účely nebo zobrazování komerčních sdělení do mapy jako interaktivní vrstvu – vhodné třeba pro akce, kdy by do mapy mohly být zaznačena místa konání a další související informace. Další možným rozšířením by mohlo být zobrazení turistických stezek do interaktivní vrstvy. K tomuto účelu by však bylo potřeba získat tato data.

Literatura

[1] PEREZ, Antonio Santiago. *OpenLayers Cookbook*. Birmingham: Packt Publishing, 2012. ISBN 978-1-84951-784-3.

[2] COOPER, Alan, Robert REIMANN, Dave CRONIN. *About face 3: the essentials of interaction design*. [3rd ed.], Completely rev. Indianapolis, IN: Wiley Pub., c2007, xxxv, 610 p. ISBN 04-700-8411-1.

[3] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: průvodce webového programátora*. Vyd. 1. Brno: Computer Press, 2007, 607 s. ISBN 978-80-251-1813-9.

[4] Component overview. *OpenStreetMap Wiki* [online]. [cit. 2013-05-06]. Dostupné z: http://wiki.openstreetmap.org/wiki/Component_Overview

[5] Elements. *OpenStreetMap Wiki* [online]. [cit. 2013-05-06]. Dostupné z: http://wiki.openstreetmap.org/wiki/Data_Primitives

[6] Planet.osm. In: *OpenStreetMap Wiki* [online]. [cit. 2013-05-06]. Dostupné z: http://wiki.openstreetmap.org/wiki/Planet.osm

[7] Databases and data access APIs. *OpenStreetMap Wiki* [online]. [cit. 2013-05-06]. Dostupné z: http://wiki.openstreetmap.org/wiki/Databases_and_data_access_APIs

[8] Rendering. *OpenStreetMap Wiki* [online]. [cit. 2013-05-06]. Dostupné z: http://wiki.openstreetmap.org/wiki/Rendering

[9] Web front end. *OpenStreetMap Wiki* [online]. [cit. 2013-05-06]. Dostupné z: http://wiki.openstreetmap.org/wiki/Web_front_end

[10] Potlatch 2. *OpenStreetMap Wiki* [online]. [cit. 2013-05-06]. Dostupné z: http://wiki.openstreetmap.org/wiki/Potlatch

[11] *OpenLayers* [online]. [2013] [cit. 2013-05-06]. Dostupné z: http://dev.openlayers.org/releases/OpenLayers-2.12/doc/apidocs/files/OpenLayers-js.html

[12] *JQuery API Documentation* [online]. © 2013 [cit. 2013-05-06]. Dostupné z: http://api.jquery.com/

[13] *JQuery UI 1.9 API Documentation* [online]. © 2013 [cit. 2013-05-06]. Dostupné z: http://api.jqueryui.com/1.9/

Seznam příloh

Příloha 1 Manuál

Pro instalaci aplikace je zapotřebí mít nainstalován a nakonfigurován webový server (např. Apache2) s podporou php5 a MySQL/MariaDB databázi v aktuální verzi.

Instalace pak probíhá v následujících krocích:

1. vytvoření databáze se jménem „oolUser“ (nebo lze změnit název databáze ve PHP skriptech)
2. spuštění skriptu oolUser.sql, který vytvoří uživatele, tabulky a naplní je vzorovými daty
3. nakopírovat obsah archivu do složky webového severu (dle konfigurace např `/home/<username>/public_html` nebo `/srv/www/htdocs/` a podobně)
4. změnit adresu v souboru oolMap.js na řádku 8 (proměnná `shareUrl`) na adresu serveru, na který byla aplikace nainstalovaná

Případně lze vyzkoušet aplikaci umístěnou na adrese http://osmmap.8u.cz/oolMapV2/

Seznam obrázků