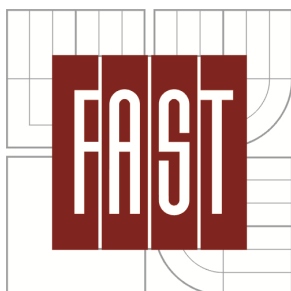


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STAVEBNÍ

ÚSTAV GEODÉZIE

FACULTY OF CIVIL ENGINEERING

INSTITUTE OF GEODESY

## TVORBA ON-LINE GEO-WEB APLIKACÍ - LOKALITA RUDOLECKO

CREATION OF ON-LINE GEO-WEB APPLICATION - RUDOLEC LOCALITY

BAKALÁŘSKÁ PRÁCE

BACHELOR THESIS

AUTOR PRÁCE

AUTHOR

JIŘÍ KNĚŽÍČEK

VEDOUCÍ PRÁCE

SUPERVISOR

DOC. ING. DALIBOR BARTONĚK, CSC.

BRNO 2012

## **ZADÁNÍ VŠKP**

**Prohlášení:**

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Použité zdroje, prameny a literaturu, které jsem při zpracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

V Brně dne .....

.....

podpis autora

**Poděkování:**

Děkuji svému vedoucímu bakalářské práce doc. Ing. D. Bartoňkovi, CSc. za cenné rady a konstruktivní připomínky při tvorbě této práce. Dále bych rád poděkoval paní Ing. arch. Marii Ryškové za konzultace týkající se grafiky webu a cenné připomínky k funkcím webu a vzhledu jeho jednotlivých prvků. V neposlední řadě patří poděkování panu Ing. Tomáši Procházkovi za návrh tématu práce s možností užitečného využití do budoucna.

V Brně dne:.....

.....

## **Bibliografická citace VŠKP**

KNĚŽÍČEK, Jiří. *Tvorba on-line geo-web aplikací - lokalita Rudolecko*. Brno, 2012. 43 s., 43 s. příl. Bakalářská práce. Vysoké učení technické v Brně, Fakulta stavební, Ústav geodézie. Vedoucí práce doc. Ing. Dalibor Bartoněk, CSc.

## **Abstrakt v českém jazyce**

Cílem této bakalářské práce bylo vytvoření geo-informační webové aplikace oblasti Rudolecka, s důrazem na jednoduché a přehledné ovládání a možnost správy systému více uživateli a to prostřednictvím webových formulářů. Aplikace byla vytvořena pomocí nástrojů PHP, MySQL a JavaScript, s uživatelským rozhraním tvořeným formuláři v HTML. Webová aplikace je tvořena částí statickou, která shrnuje historické a geografické skutečnosti oblasti Rudoleckého panství a částí dynamickou s obsahem uloženým v SQL databázi. Takto vytvořená aplikace bude nadále provozována na webu Rudolecko.cz a bude sloužit jako geo-informační web pro návštěvníky této turistické lokality.

## **Abstract in English**

The aim of this work was the creation of geo-informative web-application of Rudolec locality, with emphasis on simple and easy way to control and manage multi-user system through web forms. The application was created using the tools of PHP, MySQL and JavaScript, the user interface consisted of an HTML form. The web application is composed of static part, which summarizes the historical and geographical facts of Rudolec estate and the dynamic content stored in SQL database. This application will continue to run applications on the Web Rudolecko.cz and will be used as a geo-informative website for visitors of this tourist locality.

## **Klíčová slova v českém jazyce:**

Rudolecko, Rudolecké panství, Český Rudolec, API Mapy.cz, turistický informační web

## **Key words in English:**

Rudolec locality, Rudolec estate, Český Rudolec, API Mapy.cz, web for tourists

## Obsah

1. ÚVOD .....	7
1.1 Motivace.....	7
2 ZÁJMOVÁ OBLAST .....	7
2.1 Česká Kanada.....	8
2.2 Rudolecké panství .....	9
2.3 Zámek Český Rudolec .....	9
2.3.1 Historie.....	9
2.3.2 Současnost.....	10
3 TECHNOLOGIE .....	12
4 APLIKACE RUDOLECKO .....	13
4.1 Grafická podoba.....	13
4.2 Redakční systém .....	15
4.2.1 Intranet .....	15
4.2.2 Články .....	19
4.2.3 Galerie .....	21
4.2.4 Počítadlo.....	24
4.2.5 Func.....	25
4.2.6 Points.....	26
4.2.7 Trasy.....	32
4.2.8 JavaScript.....	34
4.2.9 Status.....	34
5 API .....	34
5.1 Mapy.cz API .....	35
5.2 Využití funkce API Mapy.cz.....	36
6 ZÁVĚR .....	39
ZDROJE INFORMACÍ .....	41
SEZNAM POUŽITÝCH ZKRATEK.....	41
SEZNAM PŘÍLOH.....	43

# 1 ÚVOD

František Šokoň, významný rodák oblasti Rudolecka, kdysi řekl: „*Touha po vědění je to, co nás žene kupředu, ale teprve poznání je tím, co nás skutečně naplňuje!*“ [1]

Chceme-li tedy něco vědět, musíme si patřičné informace najít, případně najít někoho, kdo nám informace předá. V dřívějších dobách musel člověk navštěvovat knihovny, univerzity a celé hodiny hledat konkrétní informace v knihách. Avšak dnešní doba, zaměřená především na výkon a efektivitu, v tomto případě tedy zjistit co nejvíce informací za co nejkratší čas, otevírá prostor rychlejším a moderním technologiím s internetem a internetovými aplikacemi v popředí. Za vše hovoří rychlý rozvoj internetových aplikací a četnost, s jakou pro hledané informace chodíme právě tam.

Jednoduchost získávání informací a jejich množství však s sebou nesou nutnost s nimi pracovat a brát v úvahu jejich relevantnost a důvěryhodnost. Důvěryhodnost informacím dodává právě ono poznání a prožití této informace na vlastní kůži. Přečteme-li si například na internetu o nově nalezeném živočišném druhu, musíme brát informaci s rezervou a prověřit si její pravdivost z více zdrojů. Avšak teprve to, že nový živočišný druh uvidíme na vlastní oči, nám dodá ono poznání a potvrzení pravdivosti nabytých vědomostí. A stejně je to se vším, co v životě poznáváme.

## 1.1 Motivace

Cílem této bakalářské práce je vytvořit webovou geo-aplikaci, která bude nabízet aktuální a komplexní informace o oblasti Rudolecka lidem, kteří by ji rádi nejenom teoreticky poznali, ale i prožili na vlastní kůži.

Motivací k tvorbě takovéto aplikace je jednak neexistence webu, který by nabízel ucelené a aktuální informace o dané oblasti, a také události, které se v posledních letech dějí v obci Český Rudolec, především v souvislosti s místním zámekem, který v roce 2009 zakoupila brněnská firma PENA Plus a.s. a svojí aktivitou zahájila cestu k opravě této ohrožené památky, přilehlého okolí a pozvednutí kulturního a společenského dění v této malé obci.

Ve spolupráci s panem Ing. Tomášem Procházkou z občanského sdružení Paměť Rudolecka, které úzce spolupracuje s majitelem zámku, pak vzniknul nápad na vytvoření takového webu, který by poskytoval potenciálním návštěvníkům Českého Rudolce a okolí kvalitní a aktuální informace. Z podstaty spolupráce a návaznosti na zámek Český Rudolec pak vzešlo vymezení oblasti, jakožto historické Rudolecké panství, které za doby svého největšího rozkvětu vlastnil major ženijního vojska Michael Angelo Ritter von Picchioni.

## 2 ZÁJMOVÁ OBLAST

Středem zájmové oblasti je obec Český Rudolec. Český Rudolec, spadající v současnosti do jižních Čech, se nachází v okrese Jindřichův Hradec, přírodním celku Česká Kanada, který je nejjižnějším koutem Českomoravské vrchoviny. Obec však

historicky spadá pod Moravu, jejíž hranice probíhá ve vzdálenosti cca 10 km od Českého Rudolce.

## 2.1 Česká Kanada

Oblast České Kanady by se dala vymezit obcemi Slavonice, Kunžak a Nová Bystřice. V pomyslném srdci tohoto trojúhelníku se pak nachází hrad Landštejn, jakožto neodmyslitelná dominanta České Kanady. Svůj název si oblast vysloužila díky rozsáhlým smrkovým lesům, zvlněnému terénu s četnými potůčky a specifickým mikroklimatem, který v zimním období skutečně připomíná kanadské zimy. Název Česká Kanada měl poprvé použít spisovatel a novinář, jindřichohradecký rodák, Jaroslav Arnošt Trpák v roce 1925. Krajina mu připomínala Kanadu, kterou předtím patrně navštívil. Popsal ji v roce 1928 takto: *"přehlížíte nejromantičtější koutek české země, vlní se terén do kopců a tmaví se lesními porosty, díváte se na nesčetné hladiny nejen velkých rybníků, ale hotových jezer orámovaných hustými lesy. A kamkoliv pohlédnete, všude se lesknou rybníky, široké potoky, točí se řeky, lesnaté kopce řadí se za sebou v nekonečných řadách".*[2]

Historicky se Česká Kanada nachází na střetu tří hranic: Čech, Moravy a Rakouska. Hranice mezi Českou republikou a Rakouskem ohraničuje tuto oblast od jihu, kde na rakouském území navazuje přírodní celek Waldviertel. Historická hranice Čechy-Morava pak prochází napříč celou oblastí.

Specifické mikroklima a průběh historické hranice Čechy-Morava dělalo z této oblasti místo nesnadné pro život. Kolonizace tohoto kraje nebyla jednoduchým a jednorázovým aktem. Započala ve 12. století a skončila rozdělením území na dva celky, z nichž jeden spadal pod Landštejn a druhý pod Novou Bystřici. Už od 12. století zde započalo budování sítě vesnic, cest, hospodářských stavení a sakrálních staveb. V patnáctém století zde nastal rozvoj rybníkářství a s tím započalo i přetváření krajiny člověkem do dnešní podoby. To souviselo s obživou zdejšího obyvatelstva - především se jednalo o zemědělce. Průmyslová revoluce se tohoto kraje téměř nedotkla, čímž se zde zachovala klasická barokní krajina, která nebyla, až na výjimky, narušena.[3]

Novodobá historie tohoto kraje pak započala koncem 2. světové války. Původní, především německé obyvatelstvo, bylo násilně vysídleno. Následovalo zbudování železné opony, která tomuto kraji zasadila velmi citelnou ránu po stránce kulturní i společenské. Vlivem těchto událostí došlo k likvidaci několika vesnic a k nastěhování nového, především českého obyvatelstva. Doba totality pak byla ve znamení železné opony, hlídek pohraniční stráže a kontroly jakéhokoli podezřelého pohybu v oblasti. Zároveň tak docházelo k chátrání mnoha kulturních památek, z nichž nejedna zcela zanikla.

Dalším zlomem byl rok 1989, kdy došlo k uvolnění poměrů a k likvidaci železné opony. Otevření hranic, volný trh a pocit nově nabyté svobody vedly k četným změnám v životě místního obyvatelstva. Zánik mnoha JZD připravil mnoho lidí o práci, nicméně zvýšený zájem o krásnou a zachovalou přírodu České Kanady nabídl nové možnosti podnikání v oblasti turistického ruchu. Následné opravování zchátralých památek, budování cyklostezek, lyžařských a turistických tras, z velké části podpořených dotacemi z Evropské unie, pak byly tím, co oblasti přidalo na její atraktivitě.



V současné době tak Česká Kanada nabízí širokou škálu sportovního, kulturního a duchovního vyžití. Pro milovníky sportu je zde k dispozici rozsáhlá síť turistických stezek, cyklistických tras, v zimním období pak několik desítek kilometrů udržovaných běžeckých tras. Zájemci o architekturu pak jistě ocení historické památky ve Slavonicích a Nové Bystřici, hrad Landštejn, klášter v Klášteře u Nové Bystřice a mnoho dalších.

## 2.2 Rudolecké panství

Jednou z mnoha chátrajících památek byl i zámek v obci Český Rudolec, kterému bývá přezdíváno Malá Hluboká. Většina plánů na jeho opravu zkrachovala, především pro nedostatek finančních prostředků. Po roce 1989 pak snahy o opravu pokračovaly až do roku 2009, kdy objekt zakoupila společnost PENA Plus a.s. V prvních letech vlastnictví nového majitele došlo k základním záchranným pracím a zastřešení obvodového zdiva zámku, aby se tak zabránilo jeho dalšímu zbytečnému chátrání. Poloha zámku, tedy prakticky uprostřed obce, byla jedním z mnoha faktorů, které zapříčinily, že nebyl rozebrán na stavební materiál a nezaniknul úplně. Takový osud potkal například zámek v nedaleké Maříži, kterému bylo přezdíváno svého času Malá Versailles.

Aktivita nového majitele vede k postupnému opravování zámku, přilehlých budov, opravě zámeckého parku a snaze o pozvednutí kulturního dění v obci pořádáním každoročních zámeckých slavností. Budoucnost zámku závisí na finančních prostředcích, které bude nový majitel schopen na opravu památky sehnat, a především na zájmu veřejnosti.

## 2.3 Zámek Český Rudolec

### 2.3.1 Historie

První písemná zmínka o zámku, resp. vodní tvrzi, pochází z roku 1351. Z této doby pochází zápis uvádějící, že Pešík koupil od svého bratra část tvrze v Rudolci. V roce 1466 zakoupil zámek Václav z Mařejže a rod Vladyků, ze kterého pocházel, držel zámek až do roku 1630. V roce 1630 zámek zakoupil císařský hejtman Ferrante Coquito.

Následovalo střídání majitelů bez širšího popisu zámku a jeho inventáře až do roku 1810, kdy zakoupil zámek Řehoř Hrabě Radzumovský, šlechtic ruského původu. Za dob jeho působení a působení jeho syna Lva Radzumovského došlo ke zbudování železáren v nedalekém Dolním Bolíkově a k četným stavebním úpravám okolí.

Zlomem byl rok 1856, kdy zámek zakoupil Michael Angelo Ritter von Picchioni, za jehož panování doznalo Rudolecké panství největšího rozkvětu. Po bezmála 4 letech však zámek kompletně vyhořel a Anggelo Picchioni se rozhodnul pro přestavbu po vzoru zámku Mira-Mare ve stylu tzv. tudorovské novogotiky, čímž zámek získal svou dnešní podobu. K přestavbě došlo údajně dle vlastního projektu majitele zámku.

Po smrti Anggela Picchioniho se stává majitelem jeho syn Ludvík Picchioni. Ten však umírá bezdětný v roce 1908. Majetek tak převzala jeho sestra Jana, která se provdala za německého obchodníka Kurta Reinhartha. Posledním majitelem zámku byl pak jejich syn Arnošt, který však kvůli své nacistické minulosti musel po válce zámek v rychlosti opustit a utíká do Argentiny. [4]



*Obr. 2.1 Podoba zámku z 18. století a z doby kolem roku 1900*

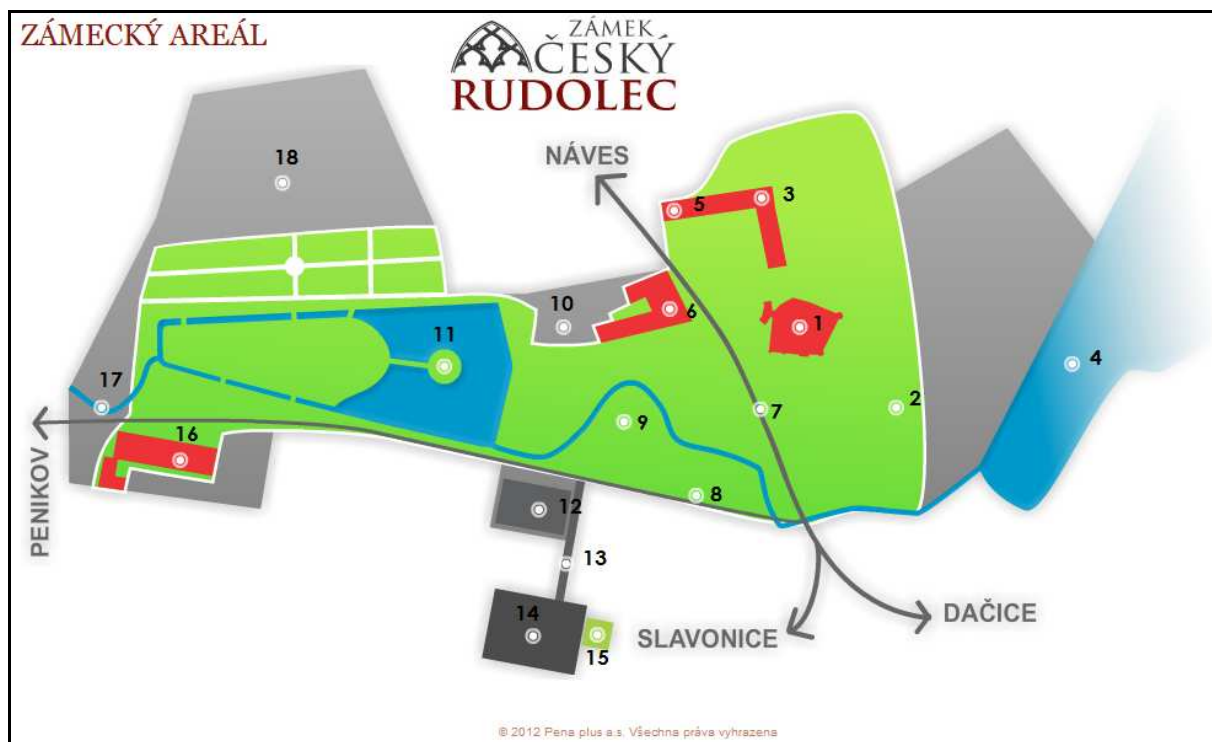
Po válce stihlo zámek několik pohrom. První ránou bylo osvobození Rudolce Rudou armádou, která se svým řáděním podepsala především na interiérech zámku a jeho devastaci. Zbudování infekčního pavilonu k zajateckému táboru, který se nacházel v lese za Pilečským rybníkem, bylo další ranou pro osud zámku.

V roce 1945 došlo na základě Benešových dekretů ke konfiskaci majetku rodu Reinharthů. Plány na využití vydrancovaného zámku braly povětšinou rychle za své, a to z důvodu nedostatku financí. Zlomem byl až rok 1988, kdy byl zámek převeden do majetku JZD Chýně, pod jehož péčí měl být zámek přestavěn na rekreační a vzdělávací středisko. Bohužel celý plán opět zkrachoval na nedostatku finančních prostředků. Následovalo opět střídání majitelů a to až do roku 2009.

### 2.3.2 Současnost

V roce 2009 zámek zakoupila společnost PENA Plus a.s., nynější majitel zámku, a pod správou této společnosti dochází postupně k opravě jednotlivých budov náležících k zámku. Zároveň dochází ke spolupráci s obcí a místními organizacemi, především občanským sdružením Paměť Rudolecka, a společně jsou pořádány pravidelné Rudolecké zámecké slavnosti. Snaha na kulturní a turistické pozvednutí celé obce tak nabrala jasný směr a v současné době dostává i reálné obrysy v podobě opravy budov náležících k zámku.

Aktuálně dochází k přestavbě a rekonstrukci zájezdního hostince na restauraci s penzionem a minipivovarem. Souběžně s tím se rekonstruuje bývalá budova správce zámku, ve které bude zřízeno turistické informační centrum s příslušenstvím pro turisty.



Obr. 2.2 Zámecký areál a přilehlé okolí

- 1) Zámek Český Rudolec
- 2) Zámecký park
- 3) Zámecké konírnny
- 4) Pilský rybník
- 5) Budova správce zámku
- 6) Zájezdni hostinec
- 7) Silnice před zámekm – historická trasa Praha-Vídeň
- 8) Lipová alej
- 9) Anglický park
- 10) Oranžerie
- 11) Rybník Vrzalák s umělým ostrůvkem
- 12) Barokní pětipatrová sýpka
- 13) Bývalá křížová cesta
- 14) Hřbitov s kaplí sv. Kříže
- 15) Hrobka rodu Picchioni
- 16) Bývalý rudolecký pivovar
- 17) Rudolecký potok
- 18) Horní francouzský park a zahrada

### 3 TECHNOLOGIE

Původním záměrem světové sítě internet bylo propojení vědeckých pracovišť a usnadnění výměny vzájemných poznatků. Poprvé byla služba World-Wide Web spuštěna na půdě výzkumného centra CERN v roce 1990. V té době si vystačila s pouhými třemi technologiemi. Jednou byl jazyk HTML, který slouží k zápisu webových stránek a je dodnes stěžejní technologií při tvorbě webu. Druhou technologií byl protokol HTTP, který sloužil a dodnes slouží k přenosu HTML stránek od serveru do prohlížeče uživatele. Třetí technologií této služby je URL, což je jedinečná adresa v rámci webu a identifikuje tak jednotlivé objekty.

Podíváme-li se na tyto technologie z pohledu dnešního uživatele, nenabízely příliš mnoho možností. V podstatě se jednalo o několik dokumentů, které byly vzájemně provázány odkazy. Časem však přišla potřeba interakce s koncovým uživatelem a možnost generovat výsledný dokument na základě proměnlivosti času. Nabízela se tak možnost odkazovat statickým souborem na spustitelný soubor nebo program, jehož výsledkem bude HTML stránka s aktuálními informacemi. Proto byla potřeba vytvořit rozhraní, které by zprostředkovalo výměnu dat a informací mezi uživatelem a serverem. Vzniklo tak rozhraní zvané CGI, podle něhož jsou dodnes serverové programy generující HTML stránky nazývány CGI skripty.

Nastartovaný vývoj směřoval k tomu, aby samotný uživatel mohl ovlivňovat chod CGI skriptu. Vzniklo tak HTML 2.0, které umožňovalo včlenit do stránky formulář, jenž následně vyplněné údaje předal CGI skriptu ke zpracování. Zpočátku se pro psaní CGI skriptů používaly některé interpretované jazyky, avšak nebyl problém použít libovolný programovací jazyk. Bylo proto nutné znát nějaký programovací jazyk a zároveň ovládat rozhraní CGI skriptů.

Problémem CGI skriptů však byla jejich pomalá odezva a velká zátěž pro server, na kterém skript běžel. To nastartovalo snahu o přesun programů na stranu uživatele. V této době byly představeny dvě nové technologie, které toto řešily. Jednou z nich byla Java od firmy Sun Microsystems, která umožňovala psaní java-apletů, jakožto krátkých programů včleněných do samotného HTML dokumentu. Dodnes je Java velice často používanou technologií, která s postupným navyšováním výkonu osobních počítačů našla velmi široké uplatnění na poli webových technologií.

Další takovou technologií byl JavaScript, coby jednoduchý jazyk vycházející z programovacího jazyku Java. S JavaScriptem přišla v roce 1996 firma Netscape ve svém prohlížeči Netscape Navigator. JavaScript se zapisoval přímo do HTML kódu a měl široké využití, nejčastěji však byl používán ve spojení s formuláři.

Velkým krokem kupředu byla technologie firmy MicroSoft nazvaná ASP, která pracovala na straně serveru a nabízela širokou škálu využití pro programování CGI skriptů. Nevýhodou této technologie však bylo, že se jednalo o komerční a nikterak levný produkt, omezený navíc funkcí pouze na platformě Windows.

Nedostatky těchto technologií vyřešil systém PHP. Ten byl od začátku šířen a tvořen jako freeware a jeho velkou výhodou byla nezávislost na platformě. Dodnes je PHP k dispozici jak pro Unix, jakožto nejčastěji používaný serverový operační systém, tak pro Windows. PHP bývá nejčastěji využíváno v kombinaci se serverem Apache, který je taktéž šířen jako freeware. U zrodu PHP stál Rasmus Lerdorf, z jehož původního osobního systému, který se zalíbil mnoha uživatelům, vznikl nástroj PHP/FI 2.0. Odtud už byl pouze

krůček k PHP 3.0, který byl oproti původní verzi značně rozšířen a doplněn o mnoho užitečných funkcí. Na vývoji tohoto nástroje pak nadále pracuje Rasmus Lendorf společně s několika dalšími vývojáři a dnes již máme k dispozici PHP 5.0. Díky freeware licenci a jednoduchosti implementace PHP interpretu do struktur serverových systémů, především již zmiňovaného Apache, se stal tento nástroj nejpoužívanějším a nejrozšířenějším nástrojem pro tvorbu dynamických WWW stránek. [5]

Pro tvorbu geo-web aplikace, která je výstupem této práce, byla zvolena nejčastěji používaná kombinace nástrojů. Aplikace tak byla vytvořena za pomoci programovacího jazyka PHP, databázového nástroje MySQL, výstup je pak tvořen HTML dokumentem s definováním vlastností dokumentů pomocí CSS a s dílčím zpracováním získávaných dat pomocí skriptovacího jazyka JavaScript.

## 4 APLIKACE RUDOLECKO

S ohledem na požadavky o budoucím využití celé aplikace jakožto veřejného webu lokality Rudolecka byla zvolena komplexní forma redakčního systému, v rámci kterého je zakomponována i geo-web aplikace pro práci s body a se zobrazením těchto bodů na mapě. Pro provoz takového systému je tedy potřeba serverový program s interpretem skriptovacího jazyk, ve které je tato aplikace vytvořena a databázovým programem pro uchovávání dat. Celá aplikace, včetně redakčního systému, je původním dílem autora této práce a nejedná se o žádný z všeobecně rozšířených redakčních systémů. Převzaty jsou pouze funkce poskytované prostřednictvím *API* rozhraní webem Mapy.cz.

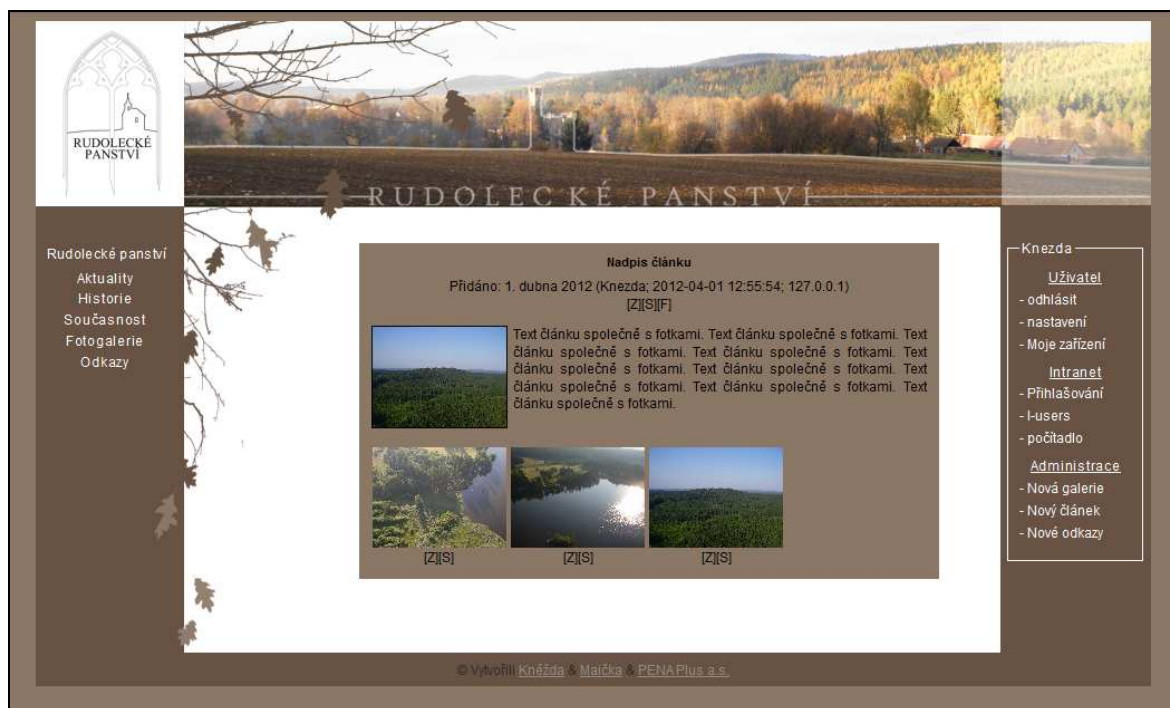
### 4.1 Grafická podoba

Grafická podoba webu byla tvořena tak, aby svým vzhledem upoutala běžného návštěvníka a zároveň nabízela jednoduše a přehledně uspořádané informace. Návrh byl proto konzultován a vytvořen ve spolupráci s paní Ing. arch. Marií Ryškovou.

Grafický návrh byl vytvořen za pomoci programů Adobe Photoshop a IrfanView. Samotná grafika byla poté rozdělena na několik dílčích částí, z nichž byl následně v HTML nakódován základní vzhled jedné stránky webu. Struktura takové stránky je vytvořena v HTML jako tabulka s několika dalšími vnořenými tabulkami, jejichž vlastnosti jsou definovány v samostatném souboru CSS. Jednotlivé tabulky jako takové pak obsahují různé navigační a grafické prvky webu. Samotná grafická část je vystředěna a po obvodu podbarvena světle hnědou barvou.

Grafická část je tvořena horním titulním pruhem, který vlevo obsahuje nové navržené logo Rudolecké panství, odkazující na domovskou stránku webu, ve zbytku pak obsahuje titulní foto, zobrazující pohled na obec Český Rudolec s dominantou zámecké věže.

Ústřední část webu je rozdělena na tři svislé části. Levý sloupec slouží jako hlavní menu, s hnědým pozadím a bílým textem, který, po najetí myši na odkaz, mění barvu a zobrazuje se jako podtržený text.



Obr. 4.1 Grafická podoba webu

Prostřední sloupec slouží jako hlavní část webu, ve které se zobrazují informace jednotlivých sekcí. Sekce jsou pak samostatně řešeny tak, aby se jejich obsah zobrazoval v omezeném prostoru středního pruhu.

Pravý sloupec obsahuje prvky sloužící ke změně obsahu webu. V první řadě je jím formulář pro přihlášení s odkazem na možnost registrace uživatele. Po přihlášení uživatele se na místě přihlašovacího formuláře zobrazí uživatelské menu v závislosti na typu přihlášeného uživatele.

Web je pak zdola zakončen hnědým pruhem ve stejné barvě, jako oba postranní sloupce a opticky tak uzavírá zdola celý obsah webu, čímž vytváří dojem uceleného bloku. Spodní informační pás obsahuje informace o autorství a vlastnictví autorských práv s odkazy na weby majitelů práv.

Grafické části, z nichž je web tvořen, jsou uloženy ve složce *root/images*, jednotlivé části jako je horní titulní pruh, středový pruh a spodní pruh jsou pak načítány ze samostatných souborů. Jednotlivé části jsou uloženy ve složce *root/requires* a umožňují tak jednoduše upravit změnu vzhledu všech stránek webu a to prostou změnou jednoho souboru.

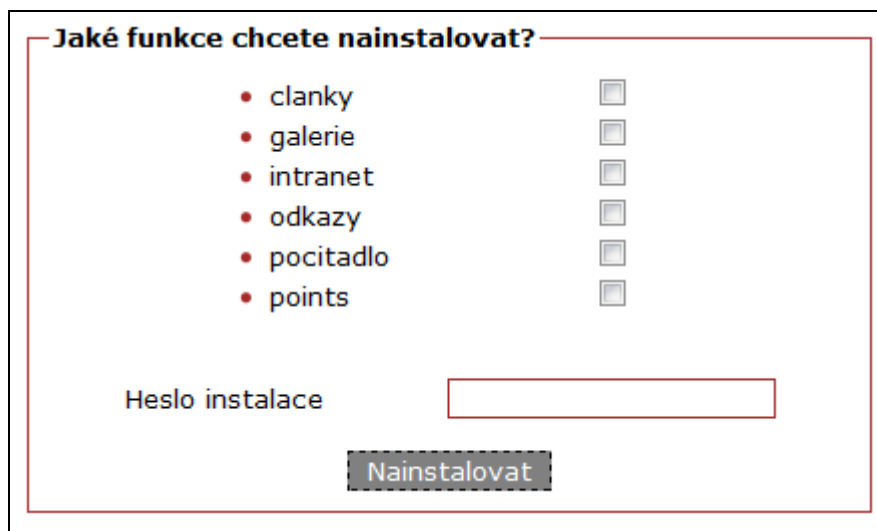
Vlastnosti jednotlivých prvků webu jako jsou texty, tabulky, formulářová políčka a grafické prvky, jsou nadefinovány prostřednictvím CSS v samostatném souboru. Soubor *css.css* je uložen ve složce *root/css* a prostřednictvím změny jednotlivých stylů je tak opět možné měnit vlastnosti prvků příslušného stylu na všech stránkách webu.

## 4.2 Redakční systém

S ohledem na požadavky na využití celé aplikace do budoucna bylo potřeba vytvořit takový systém, který by umožňoval přidávat, editovat a mazat uchovávaná data skupině několika uživatelů, a to na základě jejich důležitosti a funkce v rámci této skupiny a bez jakékoli znalosti HTML nebo CSS. Forma celého webu tak byla zvolena jako systém několika statických stránek se stálými a neměnnými informacemi o Rudolecku a přilehlém okolí a dynamická část editovatelná a uchovávaná jako data v databázi MySQL.

Pro dynamickou část webu byla zvolena forma *CMS* s několika základními moduly. Veškeré moduly jsou naprogramovány v programovacím jazyku PHP, data jsou ukládána do databáze MySQL.

Moduly jsou uloženy ve složce *root/plugins* a jsou vždy načítány do souboru *index.php* nebo souboru *work.php*, které jsou uloženy ve složce *root*. Každý modul obsahuje složku *create* se startovacím souborem *index.php* této funkce. Samostatné spuštění startovacích souborů funkce není možné, pro instalaci jednotlivých modulů slouží složka *root/start*. Instalační soubor obsažený v této složce vypíše seznam všech modulů, které jsou k dispozici ve složce *root/plugins* a je potřeba je před vlastním spuštěním nainstalovat. Instalační soubor tak nabízí možnost nainstalovat buď všechny dostupné moduly, případně jen zvolit, které moduly je potřeba nainstalovat. Pro instalaci je potřeba zadat heslo nastavené autorem systému a slouží i pro potřeby přeinstalování některé funkce v budoucnu.



Jaké funkce chcete nainstalovat?

- clanky
- galerie
- intranet
- odkazy
- pocitadlo
- points

Heslo instalace

Nainstalovat

Obr. 4.2 Ukázka instalačního souboru

### 4.2.1 Intranet

Hlavním modulem je funkce intranet, která slouží jako rozhraní pro přihlašování a přístup ke správě obsahu webu. Základem této funkce jsou tabulky *Rud\_Users* a *Rud\_RegUsers*.

Tabulka *Rud\_Users* slouží k uchovávání informací o uživatelích a jejich nastavení. Uchovává data jako je uživatelské jméno, přístupová práva, heslo (v zašifrované podobě),

jméno, příjmení, email, web, datum posledního přihlášení, proměnnou log k identifikaci aktuálního přihlášení, datum, čas a IP adresu posledního přihlášení a další systémové informace. Tabulka *Rud\_RegUsers* slouží jako úložiště informací o uživateli, kteří žádají o registraci, a uchovává tak prakticky stejná data, jako tabulka *Rud\_Users*.

Samotní uživatelé jsou členěni do 3 základních skupin – *admin*, *redaktor*, *user*. Uživatelé pak, podle nastavených přístupových práv, mají možnost pracovat s obsahem webu a měnit jej.

Uživatel s právy *Admin* má právo měnit a mazat jakékoli informace na webu, jakožto administrátor má zároveň přístup k účtům ostatních uživatelů, jejichž nastavení a hesla tak může měnit. Zároveň schvaluje žádosti o registraci nových uživatelů a určuje jejich přístupová práva.

Uživatel s právy *Redaktor* má možnost přidávat nový obsah do jednotlivých sekcí a měnit a mazat příspěvky, které sám přidal. Zároveň má možnost měnit svoje nastavení a údaje o svém účtu v části *Nastavení*.

Uživatel s právy *User* má možnost měnit své nastavení, vystupovat pod svým uživatelským jménem v rámci webu a má rozšířený přístup k některým informacím na webu.

#### 4.2.1.1 Přihlašování

Přihlašování do systému je založeno na proměnných *sessions*. Po zadání uživatelského jména a hesla a odeslání formuláře nalezne skript uživatele a údaje o něm v tabulce *Users*. Zadané heslo *zaheshuje* pomocí hashe MD5 a srovná heslo s heslem v databázi. Pokud jsou hesla shodná, je vygenerován 24místný kód. Ten je uložen jako identifikátor uživatele do databáze společně s datem a časem poslední aktivity a je zapsán jako *session* s označením *log*. Při jakémkoli dalším přístupu k webu tak dojde ke kontrole, zda je zaregistrovaná *log*, pokud ano, je na základě její hodnoty nalezen uživatel, kterému patří, a upraven čas poslední aktivity této *log*. Pokud neaktivita při přihlášení překročí 45 minut, dojde k automatickému odhlášení a odstranění aktuální *log* a je potřeba provést nové přihlášení. Funkci pro přihlašování zajišťuje soubor *login.php* a soubor s přihlašovacím formulářem *login\_form.php*. Kontrolu přihlášení, která probíhá při načtení jakéhokoli souboru webu, zajišťuje soubor *sessions.php*. Všechny tyto soubory jsou součástí modulu *intranet* a jsou uloženy ve složce *root/plugins/intranet*.

#### 4.2.2.2 Zabezpečení

Z bezpečnostních důvodů byl zaveden nový prvek, kterým je autorizace zařízení. Tuto funkci si může každý uživatel aktivovat v nastavení svého účtu a především u skupiny uživatelů administrátorů je vhodné mít toto zabezpečení aktivní. Při jakémkoli přihlášení do systému je vždy ověřena proměnná, která je zapsaná jako *cookies* v počítači uživatele. Tato proměnná identifikuje zařízení, ze kterého se uživatel přihlašuje. Pokud je nalezena, zkontroluje se její platnost v databázové tabulce *Rud\_Zari*, která slouží k registraci používaných zařízení. Pokud je zařízení nalezeno, zapíše se současně se *session* s označením *log* ještě druhá proměnná, označená jako *aza*. Proměnná *aza* obsahuje řetězec identifikující aktuální zařízení.



Obr. 4.3 Ukázka formuláře pro přihlášení a formuláře pro autorizaci zařízení

Pokud se uživatel přihlašuje z nějakého zařízení poprvé, dojde při přihlášení k vygenerování šestimístného kódu, který je odeslán na email uživatele. Zároveň je nové zařízení přidáno do databázové tabulky *Rud\_Zari*, ovšem s atributem neautorizovaného zařízení. Uživatel už je sice přihlášen pod svým jménem, ale z neautorizovaného zařízení. Proto při načtení jakékoli stránky je vyzván k autorizaci zařízení, tedy zadání autorizačního kódu a názvu zařízení. Bez autorizace zařízení nemá uživatel přístup k administraci webu. Jakmile uživatel zařízení autorizuje, může se z něj v budoucnu přihlašovat bez nutnosti jakékoli další autorizace. Každý uživatel může mít neomezený počet autorizovaných zařízení, u každého zařízení je omezena pouze doba autorizace, a to na jeden rok. Funkci pro autorizaci zařízení zajišťuje soubor *azar.php* v modulu *intranet*.

Jelikož je toto zabezpečení zařízení přes proměnné *cookies*, je potřeba autorizovat zařízení pro každý prohlížeč, ze kterého je k redakčnímu systému přistupováno, a mít ve svém prohlížeči povoleny *cookies*. Autorizace je vyžadována pouze u uživatelů, kteří si toto zabezpečení aktivovali ve svém osobním nastavení.

#### 4.2.1.3 Uživatelské menu

Po přihlášení se uživateli zobrazí v pravém sloupci menu, jehož obsah závisí na právech uživatele.

Všichni uživatelé mají v menu nabídku Uživatel s možnostmi:

- Odhlásit, která slouží k odhlášení uživatele od intranetu. Pro odhlášení slouží soubor *logout.php* v modulu *intranet*, v němž dojde k odstranění proměnných *log* a *aza* a tím i k odhlášení uživatele. V případě, že uživatel přestane být aktivní, dojde k jeho automatickému odhlášení po 45 minutách.

- Nastavení, které slouží ke změně informací o uživateli, změně hesla, a nabízí možnost aktivování funkce k autorizaci nového zařízení, ze kterého je k účtu přistupováno. Tato funkce je uložena v souboru *user\_form.php* modulu *intranet*.

- Moje zařízení slouží ke správě autorizovaných zařízení. V tabulce se zde vypisují všechna aktuální autorizovaná zařízení s informacemi o názvu zařízení, datu a času a IP adrese jeho autorizace a možnost smazání tohoto zařízení. Pokud se například uživatel přihlásí jednorázově z nějakého zařízení, ze kterého se běžně nepřihlašuje, má možnost jej z autorizovaných zařízení odstranit. V případě smazání aktuálního zařízení však dojde k automatickému odhlášení uživatele, na což je však uživatel upozorněn po kliknutí na odkaz Smazat. Tuto funkci zajišťuje soubor *spza.php* v modulu *intranet*.

Administrátorům pak přibývá část menu Intranet s možnostmi:

- Přihlašování slouží jako zabezpečení proti vstupu cizích uživatelů do redakčního systému, především k procházení historie přihlášení všech uživatelů, včetně neúspěšných přihlášení a pokusů o přihlášení. Funkce nabízí možnost nadefinovat kritéria, podle kterých se budou data vypisovat a třídit. U každého přihlášení se tak ukládají: zadané uživatelské jméno, datum a čas, IP adresa, informace o úspěšnosti nebo neúspěšnosti přihlášení a název autorizovaného zařízení, ze kterého k přihlášení došlo. Tuto funkci zajišťuje soubor *logs.php* v modulu *intranet*.

- I-users slouží k vypisování informací o uživateli. Výpis se zobrazí v tabulce pod jmény uživatelů a po kliknutí na příslušného uživatele se zobrazí podrobné informace o uživateli, jako jsou jméno, příjmení, email, web, informace, uživatelská práva a tlačítka nabízející možnost změnit nastavení uživatele, smazat uživatele a možnost procházet historii přihlašování tohoto uživatele. Tuto funkci zajišťuje soubor *user\_form.php* v modulu *intranet*.

- K-users slouží ke kontrole žádostí o registraci. Žádosti se vypisují v tabulce stejným způsobem jako u funkce I-users, kde se podrobné informace vypíší při kliknutí na konkrétní žádost. Administrátor tak vidí informace, jako jsou jméno, příjmení, email, web, informace, datum, čas a IP adresu, odkud bylo o registraci žádáno. Administrátor má pak k dispozici tlačítka pro potvrzení a zamítnutí registrace a formulář pro vložení vzkazu, který se poté odešle na email žadatele o registraci. Zároveň je zde možnost nastavit přístupová práva nového uživatele. Tato funkce vypisuje data uložená v tabulce *Rud\_RegUsers*. Po schválení registrace jsou data smazána a přesunuta do tabulky *Rud\_Users*. Uživatel tak získá možnost se přihlašovat ke svému účtu. Výpis žádostí zajišťuje soubor *user\_info.php* a schvalování registrací soubor *user\_form.php*. Oba soubory jsou součástí modulu *intranet*.

- Nový uživatel slouží k vytvoření nového uživatelského účtu bez další potřeby schvalování uživatele. Po vyplnění patřičného formuláře a jeho uložení, dojde k automatickému vytvoření záznamu v tabulce *Rud\_Users*. Nově vložený uživatel má pak možnost se normálně přihlásit ke svému účtu. Tuto funkci zajišťuje soubor *user\_form.php* modulu *intranet*.

- počítadlo slouží k zobrazení počítadla návštěvnosti, tato funkce je blíže popsána v části 4.2.4 Počítadlo.

Pro uživatele s právy *Admin* a *Redaktor* je k dispozici část Administrace s volbami na vytvoření nového obsahu webu.

- Nový článek slouží k vytvoření nového článku do sekce *Aktuality*. Podrobněji je tato funkce popsána v části 4.2.2 Články.

- Nová galerie slouží k vytvoření nové složky v sekci *Galerie*. Funkce je opět podrobněji popsána v části 4.2.3 Fotogalerie.

- Nový odkaz slouží k přidání nového odkazu do sekce *Odkazy*. Podrobnější popis funkce je popsán v části 4.2.3 Odkazy.

- Nový odkaz slouží k přidání nového odkazu do sekce *Odkazy*. Podrobnější popis funkce je popsán v části 4.2.4 Odkazy.

- Nová trasa slouží k vytvoření nové trasy do sekce Tipy na výlet. Tato funkce je blíže popsána v části 4.2.7 Trasy

- Nový bod zájmu slouží k přidání nových položek do databázové tabulky *Rud\_Points*, které jsou potom vypisovány v různých sekcích. Podrobný popis funkce je popsán v části 4.2.6 Points.



Obr. 4.4 Ukázka administrátorského menu uživatele s právy Admin

Administrační funkce pro přidávání, editaci a mazání obsahu webu zajišťují vždy soubory *admin.php* v jednotlivých modulech. Ty jsou podrobněji popsány v příslušných částech této práce.

#### 4.2.2 Články

Modul články uložený ve složce *root/clanky* slouží jako obslužný modul pro sekci Aktuality. Tento modul je zároveň navržen tak, aby umožňoval případnou obsluhu dalších sekcí, které by měly fungovat na stejném principu jako sekce aktuality. Modul umožňuje přidávat textové články a ke každému článku nahrát několik fotografií nebo obrázků. Funkce zároveň umožňuje zvolit jeden obrázek, který slouží jako titulní obrázek ke článku. Zvolený obrázek se pak zobrazuje vlevo nahoře vedle nadpisu.

Články jsou pak vypisovány ve středovém pásu jako samostatné bloky. V levé horní části obsahuje článek titulní obrázek, pokud je nějaký přiřazen, vedle pak nadpis a pod nadpisem samotný text článku. Pod textem se poté zobrazují jednotlivé obrázky, které jsou ke článku přidány. Po kliknutí na obrázek se tento otevře ve větším rozměru v novém okně v prohlížeči obrázků galerie článku.



Obr. 4.5 Ukázka výpisu článku s titulní fotografií a fotogalerií

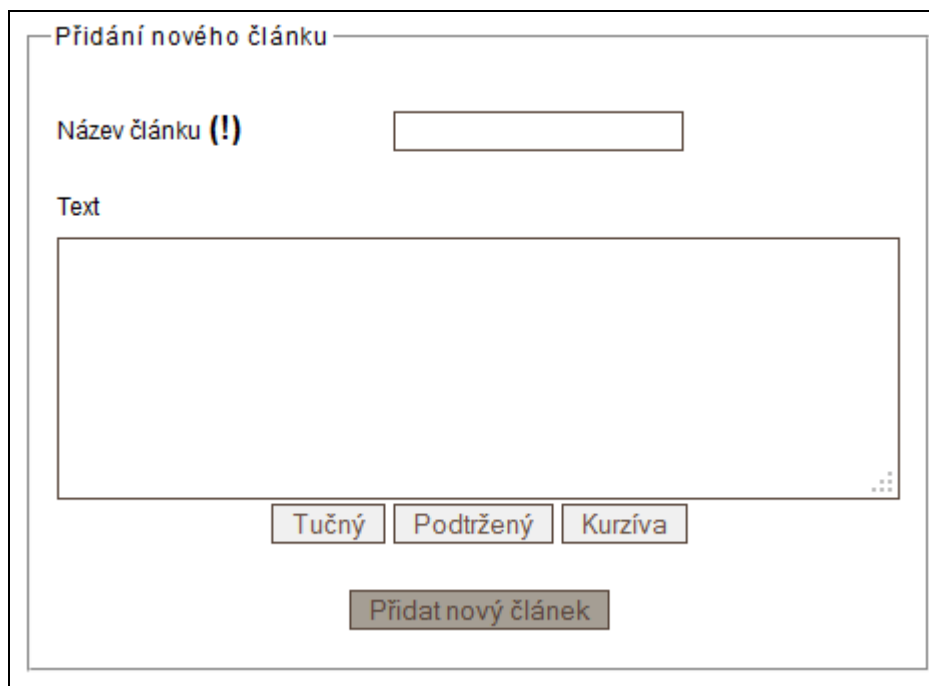
Prohlížeč obrázků galerie slouží k prohlížení jednotlivých obrázků v galerii článku, zobrazuje zvětšeninu obrázku na středu stránky a v případě, že galerie obsahuje více obrázků, vypíše vlevo a vpravo od obrázku šipky, které umožňují přechod k předchozímu nebo následujícímu obrázku, pokud je nějaký k dispozici.

V případě, že je přihlášen uživatel s právy editace článku, vypíše se mu pod nadpisem informace o tom, kdo, kdy a z jaké IP adresy článek přidal. Zároveň se vypisují i možnosti administrace článku v podobě odkazů s písmeny *Z* (změnit), které slouží k editaci článku, *S* (smazat), které slouží ke smazání článku, a v případě jehož svolení je uživatel kontrolně dotázán, zda chce opravdu článek smazat, a písmenem *F* (foto), které odkazuje na formulář pro nahrávání fotografií. Přihlášený a oprávněný uživatel má zároveň možnosti *Z* pro editaci a *S* pro smazání u jednotlivých fotografií v galerii článku.

Přidávání článků probíhá prostřednictvím souboru *admin.php*, formulář pro přidání článku obsahuje jedno políčko pro nadpis článku a velké textové pole pro text článku. Pod textovým polem jsou k dispozici tři tlačítka pro jednoduché formátování textu. Uživateli nabízí základní formátování pro tučné písmo, podtržený text a kurzívu. Funguje tak, že po kliknutí na příslušné tlačítko je na konec textu v poli vložen nový text, který je umístěn mezi HTML tagy příslušné vlastnosti. Celý formulář je pak ukončen tlačítkem *Přidat nový článek*, jímž se formulář odešle a dojde k uložení nového článku. O uložení je uživatel informován tabulkou *StatusTable*, která zároveň vypíše odkazy s možnostmi *Nahrát fotky* a *Přidat další článek*.

Přidávání obrázků do galerie článku probíhá přes samostatný formulář. Na ten je možné se dostat prostřednictvím odkazu ve *StatusTable* po přidání článku, případně z článku samotného prostřednictvím výše popsaného odkazu v podobě písmene *F*. Formulář obsahuje jedno políčko pro výběr nahrávaného souboru, jedno textové políčko

pro popis obrázku, zaškrťovací políčko pro nastavení obrázku jako titulního obrázku příslušného článku a tlačítko *Uložit fotku*, které slouží pro odeslání formuláře a následné uložení obrázku do galerie. Obrázek je po odeslání zpracován tak, že originál je uložen ve složce *root/upload/clanky/originál*. Ten je poté načten prostřednictvím funkce *resampleImg*, která je blíže popsána v části 4.2.5 Func, a zmenšen na limitní rozměry 800x800px a 120x90px, přičemž dojde ke změně velikosti tak, aby nebyly překročeny limitní rozměry, ale zároveň nedošlo ke změně poměru výšky a šířky obrázku. Takto zmenšené obrázky jsou pak uloženy do složek *w800* a *w120* na adrese *root/upload/clanky*.



The image shows a web form titled "Přidání nového článku". It features a text input field for "Název článku (!)", a large text area for "Text", and three buttons: "Tučný", "Podtržený", and "Kurzíva". At the bottom is a "Přidat nový článek" button.

Obr. 4.6 Ukázka formuláře pro přidání nového článku

Data pro chod této funkce uchovává tabulka *Rud\_Clanky\_sekce*, která obsahuje informace o jednotlivých sekcích článků a umožňuje tak vytvoření více sekcí. Tabulka *Rud\_Clanky\_clanek* obsahuje údaje o samotném článku, ukládají se zde informace jako je nadpis, text, sekce, titulní obrázek, autor článku, datum přidání a IP adresa. Poslední tabulkou je tabulka *Rud\_Clanky\_foto*, která obsahuje informace o jednotlivých obrázcích a uchovávají se zde informace o názvu souboru, příslušnosti ke článku, popis obrázku, informace o tom, kdy a kdo obrázek přidal a z jaké IP adresy.

#### 4.2.3 Galerie

Modul galerie slouží jako obslužný modul pro sekci fotogalerie. Nabízí možnost vytvářet fotogalerie a přidávat do nich následně fotografie.

Výpis obsahu fotogalerie je vykonáván prostřednictvím souboru *index.php*, který vypisuje jednotlivé fotogalerie jako samostatné obdélníky, vždy 4 fotogalerie na jeden řádek. Ve výpisu fotogalerií je vždy vypsán název fotogalerie, pod ním počet fotografií a titulní fotografie. V případě, že není titulní fotografie zvolena, je na jejím místě uveřejněn šedý obrázek s otázníkem *tf\_fotog.jpg*, který je uložený ve složce *root/images*. Titulní

obrázek slouží zároveň jako odkaz pro vstup do příslušné fotogalerie. V případě, že je přihlášený oprávněný uživatel, vypisují se pod titulním obrázkem odkazy *Z* pro změnu údajů o fotogalerii, *S* pro smazání fotogalerie, po jehož zvolení je uživatel kontrolně dotázán, zde chce fotogalerii skutečně smazat, a *F* pro nahrávání nových fotografií do fotogalerie.



Obr. 4.7 Ukázka výpisu fotogalerií s administrátorskými tlačítky pro úpravu

Výpis obsahu konkrétní fotogalerie probíhá na samostatné stránce ve formě obdélníku s náhledem fotografií, vždy po 4 fotografiích na jeden řádek. Nad výpisem fotografií je uveden název fotogalerie a její popis, pod výpisem pak odkaz *ZPĚT*, který slouží pro návrat na předchozí stránku. Každý náhled fotografie slouží zároveň jako odkaz na její zvětšeninu. Zvětšené fotografie se otevírají v novém okně, což je provedeno opět prostřednictvím souboru *index.php*. Na stránce je pak uprostřed zobrazena zvětšená verze fotografie, která je po stranách opatřena šípkami odkazujícími na předchozí a následující fotografii ve fotogalerii, pokud tedy existují. Nad samotnou fotografií je uveden její popis a v závorkách potom pořadové číslo fotografie v rámci fotogalerie a za lomítkem údaj o celkovém počtu fotografií. V případě přihlášení oprávněného uživatele se pod jednotlivými fotografiemi v jejich výpisu opět zobrazují možnosti *Z* pro editaci fotografie a údajů o ní a *S* pro smazání fotografie, které je nutné opětovně potvrdit.

Administrace fotogalerie probíhá prostřednictvím souboru *admin.php*, který také umožňuje vytvořit novou fotogalerii. K vytvoření fotogalerie slouží formulář obsahující políčka pro název fotogalerie, datum pořízení fotografií, jméno autora a popis fotogalerie. Celý formulář je ukončen tlačítkem *Vytvořit novou fotogalerii* pro odeslání formuláře a vytvoření nové fotogalerie. O uložení fotogalerie je uživatel informován prostřednictvím tabulky *StatusTable* s odkazem *Nahrát fotky*, směřujícím na formulář pro nahrávání fotografií, a odkazem *Vytvořit další fotogalerii*, který odkazuje na formulář pro vytvoření nové fotogalerie.

Vytvoření nové fotogalerie

Název (!)

Datum (rrrr-mm-dd)

Autor

Popis galerie

Obr. 4.8 Ukázka formuláře pro přidání nové fotogalerie

Stránka pro nahrávání fotografií obsahuje nadpis s názvem fotogalerie, do které je přidávána aktuální fotografie. Samotný formulář pak obsahuje políčka pro výběr nahrávaného souboru, popis fotografie, jméno autora, datum pořízení fotografie a zaškrťovací políčko pro nastavení titulní fotografie. Políčka autor a datum jsou implicitně vyplněna textem, který je pro dané hodnoty uložen v informacích o fotogalerii. Celý formulář je zakončen tlačítkem *Uložit fotku*, které slouží k odeslání formuláře a následnému uložení fotografie. Po odeslání formuláře je soubor uložen do složky *root/upload/galerie/original*. Originální fotografie je poté načtena funkcí *resampleImg*, která nastaví fotografii limitní rozměry 800x800px a 120x90px a následně zmenšené fotografie uloží do složek *w800* a *w120* na adrese *root/upload/galerie*. O uložení fotografie je uživatel informován prostřednictvím tabulky *StatusTable*, která obsahuje dále odkaz *nahrát další fotku*, odkazující na formulář pro přidání fotografie do stejné fotogalerie, a odkaz *Prohlížet fotogalerii*, která odkazuje přímo do fotogalerie, do které byla vložena aktuální fotografie.

Modul galerie je postaven na databázových tabulkách *RudSFoto* a *RudFoto*. Tabulka *RudSFoto* slouží k ukládání informací o jednotlivých fotogaleriích a uchovává informace o fotogalerii, jako jsou identifikační kód fotogalerie, název, popis, autor, titulní fotografie, datum, jméno uživatele, který fotografii přidal, datum a čas, kdy byla fotogalerie vytvořena a IP adresu, ze které vytvoření proběhlo. Tabulka *RudFoto* uchovává informace o jednotlivých fotografiích, jako jsou popis, jméno autora, název souboru, datum pořízení, jméno uživatele, který fotografii přidal, datum a čas kdy byla fotografie vložena, IP adresa, ze které k vložení došlo, a identifikační kód fotogalerie, do které fotografie patří.

Přidání fotky do fotogalerie Zámek v opravě

Vyberte fotku (!)

Popis fotky

Autor

Datum (rrrr-mm-dd)

Titulní fotka fotogalerie?

Obr. 4.9 Ukázka formuláře pro nahrání nové fotografie do konkrétní fotogalerie

#### 4.2.4 Počítadlo

Modul počítadlo slouží k evidenci počtu návštěv webu. Tato funkce je založena na tabulkách *Rud\_Pocitadlo* a *Rud\_Visitip*.

Tabulka *Rud\_Pocitadlo* slouží k uchovávání počtu návštěvníků pro aktuální den, aktuální měsíc a celkový počet návštěv. Funkce započítává návštěvu z každé IP adresy pouze jednou za den. Tabulka *Rud\_Visitip* slouží k uchovávání IP adresy, která web navštíví. U každé IP adresy se uloží datum poslední návštěvy, celkový počet návštěv a název počítadla, ke kterému záznam patří. Pokud je již IP adresa v databázi, přepíše se pouze datum a čas poslední návštěvy a hodnota celkového počtu návštěv se zvýší o jedna.

<b>Počítadlo</b>	
Celkem:	14
Tento měsíc:	7
Dnes:	1

Obr. 4.10 Ukázka výpisu počtu návštěv funkcí počítadlo

Grafický výstup o počtu návštěv je ve formě tabulky, která vypisuje postupně na třech řádcích hodnoty celkového počtu návštěv, počtu za aktuální měsíc a počtu za aktuální den.



#### 4.2.5 Func

Složka *root/plugins* obsahuje zároveň složku *func*, která není samostatným modulem, ale obsahuje několik souborů s přeprogramovanými univerzálními funkcemi pro možnost externího načtení a použití v rámci jakéhokoli dalšího skriptu. K dispozici jsou zde soubory:

- *delfile.php* – obsahující funkci *DelFile(\$dsoubor, \$slozka)*. Tato funkce odstraní soubor *\$dsoubor* složce *\$slozka*. Funkce poté vrací hodnoty TRUE v případě úspěšného smazání souboru, případně hodnotu FALSE v případě neúspěšného smazání. Tato funkce je využívána ve všech modulech, kde dochází k mazání souborů.

- *formatdate.php* – obsahuje funkci *FormatDate(\$datum, \$dtype)*. Tato funkce zformátuje proměnnou *\$datum* ve výchozím formátu RRRR-MM-DD HH:MM:SS dle kritéria *\$dtype* do předdefinovaného formátu, a to:

1- dd.mm.rrrr hh:mm:ss; 2- dd.mesic.rrrr hh:mm:ss; 3- dd.mm.rrrr hh:mm

4- dd.mesic.rrrr hh:mm; 5- dd.mm.rrrr; 6- dd.mesic.rrrr

- *genstring.php* – obsahuje funkci *GenString(\$long, \$strType)*, která slouží ke generování náhodných řetězců o délce *\$long* (počet znaků), vytvořeného z:

- čísel pro *\$strType* hodnoty 1

- písmen pro *\$strType* hodnoty 2

- kombinace číslic a písmen pro *\$strType* hodnoty 3

Tato funkce je využívána ve všech modulech, kde je uživateli poskytována informace o čase vložení (např. u výpisu v sekci Aktuality).

- *nadpistable.php* – obsahuje funkce *NadpisTable(\$text)*, která vypíše *\$text* v předdefinovaném formátu tabulky pro nadpis. Dále obsahuje funkci *NadpisTable2(\$text, \$table\_class)*, která vypíše *\$text* ve formátu *\$table\_class* tabulky pro nadpis. Funkce je používána ve všech modulech, kde je potřeba jednotný styl výpisu nadpisu.

- *readadr.php* – obsahuje funkci *ReadAdr(\$adresar, \$styp)*, která vypíše obsah složky *\$adresar* na základě kritéria *\$styp*. Proměnná *\$styp* může být prázdná, přičemž funkce vypíše všechny soubory ve složce *\$adresar*, případně může být ve formátu \*.koncovka, pak funkce vypíše pouze soubory s danou koncovkou. Jednotlivé vypsané položky slouží zároveň jako odkaz na samotné soubory.

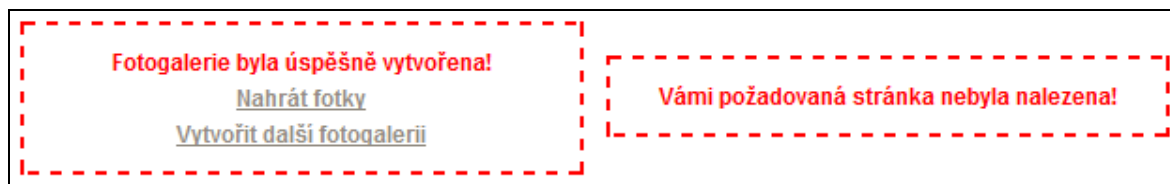
- *rezise.php* – obsahuje funkci *resampleImg(\$img, \$imgPath, \$newPath, \$newWidth, \$newHeight, \$quality)*, která upraví rozměry obrázku ve formátu \*.jpg nebo \*.gif. Funkce načte obrázek *\$img* uložený na adrese *\$imgPath*, ten upraví na rozměry *\$newWidth* nebo *\$newHeight*, které určují maximální rozměry zmenšeného obrázku, ale zároveň zachová poměr výšky a šířky, aby nedocházelo k deformaci obrázku. Zmenšený obrázek je poté uložen v kvalitě *\$quality* do složky *\$newPath* pod stejným názvem jako původní obrázek. Funkce pracuje pouze s obrázky ve formátu \*.jpg nebo \*.gif. Tato funkce je používána ve všech modulech, kde se pracuje s obrázky, které jsou následně zmenšovány na miniaturu.

- *selmaxid.php* – obsahuje funkce *SelMid(\$stable)*, která nalezne v tabulce *\$stable* nejvyšší hodnotu položky Id a její hodnotu zvýší o 1. Funkce poté vrací novou proměnnou *\$maxid*, jakožto nové nejvyšší Id. Dále obsahuje funkci *SelMidS(\$stable, \$selidname, \$sel\_podm)*, která nalezne nejvyšší hodnotu položky *\$selidname* v tabulce *\$stable* na

základě nadefinovaných podmínek  $\$sel\_podm$ . Tato funkce navýší hodnotu o 1 a vrací proměnnou  $\$maxid$ , jakožto novou nejvyšší hodnotu pro stanovenou položku a podmínky. Tato funkce je použita všude tam, kde slouží jako identifikátor položky v databázi číselná hodnota.

- *statustable.php* – obsahuje funkci `StatusTable($text)`, která vypíše  $\$text$  v předdefinovaném formátu pro tabulku statusu. Tato funkce je používána tam, kde je potřeba vypsát jednotně formátovaný text, především u chybových hlášek.

- *statustableadmin.php* – obsahuje funkci `StatusTableAdmin($text, $odkaz, $text_odkazu)`, která vypíše  $\$text$  společně s odkazem na adresu  $\$odkaz$  s textem odkazu  $\$text\_odkazu$  v předdefinovaném formátu pro tabulku formátu *status* s odkazem. Dále obsahuje funkci `StatusTableAdmin2($text, $odkaz 1, $text odkazu 1, $odkaz 2, $text odkazu 2)`, která funguje stejně jako funkce první, vypisuje však dva různé odkazy. Tyto funkce jsou používány tam, kde je potřeba vypsát jednotně formátovaný text a zároveň uživateli nabídnout možnost pokračovat na nějakou další související stránku (např. u uložení informací do databáze).



Obr. 4.11 Ukázka tabulky *StatusTable* a *StatusTableAdmin*

- *upload.php* – obsahuje funkci `copyFile($file, $fileName, $path, $namePart)`, která zkopíruje soubor  $\$file$  o názvu  $\$fileName$ , vytvoří nový název souboru podle klíče  $\$newPartName\_řetězec$ , kde *řetězec* je pětitimístný náhodný řetězec, a soubor uloží do složky  $\$path$ . Funkce vrací nový název souboru  $\$newFileName$ . Funkce je využívána tam, kde je potřeba zkopírovat uživatelem odeslaný soubor do nějaké složky na serveru.

#### 4.2.6 Points

Samotná geo-web aplikace je tvořena modulem Points, který je založen na databázové tabulce *Rud\_Points*. Tato tabulka slouží k ukládání bodů zájmu, přičemž umožňuje jejich členění do různých sekcí podle typu cíle, o který se jedná. Podle těchto kritérií je potom možné dané body vypisovat a členit do jednotlivých sekcí webu. U každého bodu je evidován jeho název, zeměpisné souřadnice místa ve tvaru  $SS^{\circ}MM.MMM'$ , textové informace o místě, informace o přiloženém obrázku, odkaz na související webovou stránku, pokud taková existuje, informace o tom, který uživatel, kdy a z jaké IP adresy položku vložil. V případě, že je bod součástí nějaké výletní trasy, jsou u něj navíc evidovány informace o příslušnosti k trase, číslo zastavení na této trase a vzdálenost k dalšímu bodu trasy.

Soubory pro obsluhu této databázové tabulky jsou uloženy ve složce *root/plugins/points*. Pro editaci a přidávání obsahu slouží soubor *admin.php*, který je přístupný pouze uživatelům s právy *Admin*. Soubor obsahuje formulář pro přidávání nového obsahu s formulářovými políčky pro název místa a sekce, do které bude nový bod zařazen. V případě zvolení sekce *Výletní trasy* se pod polem pro výběr sekce zobrazí

doplňkový formulář pro výběr trasy a čísla zastavení na této trase a dále políčko pro vzdálenost k dalšímu bodu trasy včetně jednotek této vzdálenosti. Hlavní formulář obsahuje dále políčka pro zeměpisné souřadnice bodu, přičemž u každé souřadnice se zadává samostatně hodnota pro stupně a pro minuty. Minuty je možné zadávat na tři desetinná místa, k oddělení je potřeba použít desetinnou tečku z důvodu budoucího přepočítávání souřadnic pro další použití. Dalšími políčky formuláře jsou políčka pro výběr souboru, kterým může být obrázek nebo fotografie ve formátu \*.jpg nebo \*.gif, políčko pro adresu webové stránky, která s místem souvisí, a textové pole pro vložení informací o bodu. Celý formulář je ukončen tlačítkem pro jeho odeslání, přičemž kliknutím na něj je uživatel kontrolně dotázán, zda chce skutečně uložit zadané informace, aby nedocházelo k náhodnému odesílání neúplně vyplněných formulářů. Povinnými údaji, bez kterých nedojde k uložení zadaných dat, jsou název, sekce a zeměpisné souřadnice.

Souřadnice bodu lze získat buď přímým měřením prostřednictvím turistického GPS přijímače, nebo odměřením z mapy prostřednictvím dynamického nástroje pro zjišťování souřadnic nějakého mapového serveru. Původní záměr využití API Mapy.cz pro možnost zjištění souřadnic přímo ve formuláři pro přidání bodu byl nakonec zavrhnut z důvodu stanoveného podmínkami využití. Smluvní podmínky využití služby API Mapy.cz stanovují v bodu 3.1. *Uživatel může použít Službu pouze za těchto podmínek tuto podmínku: „- uživatelé internetu nesmí být nijak omezeni v přístupu na stránky, kde je Služba použita - povoleno je pouze přihlašování uživatelů, avšak - nikoliv například zpoplatněný přístup uživatelů, nebo provozování Služby v rámci intranetu či za firewallem“.*[6] Proto je pro odměření souřadnic z mapy potřeba využít přímo webovou stránku Mapy.cz, která umožňuje výstup souřadnic v různých formátech, mimo jiné i v požadovaném formátu  $SS^{\circ}MM.MMM'$ . Zvolený formát souřadnic byl vybrán především proto, že v tomto formátu jsou poskytovány souřadnice pro hru *Geocaching*, která se v posledních letech stala v České republice velmi rozšířenou a díky které tento formát souřadnic v podstatě zlidověl.

Vytvoření nového bodu zájmu

Název bodu (!)

Sekce (!)

Jedná-li se o bod na trase

Trasa-[zastavení]

Vzdálenost k dalšímu bodu(Včetně jednotek!)

Souřadnice ve tvaru SS° MM.MMM' (desetinná tečka!) (!)

N	<input type="text"/>	°	<input type="text"/>	'
E	<input type="text"/>	°	<input type="text"/>	'

Foto

Odkaz

Informace o bodu

Obr. 4.12 Ukázka formuláře pro přidání nového bodu zájmu pro bod na výletní trase

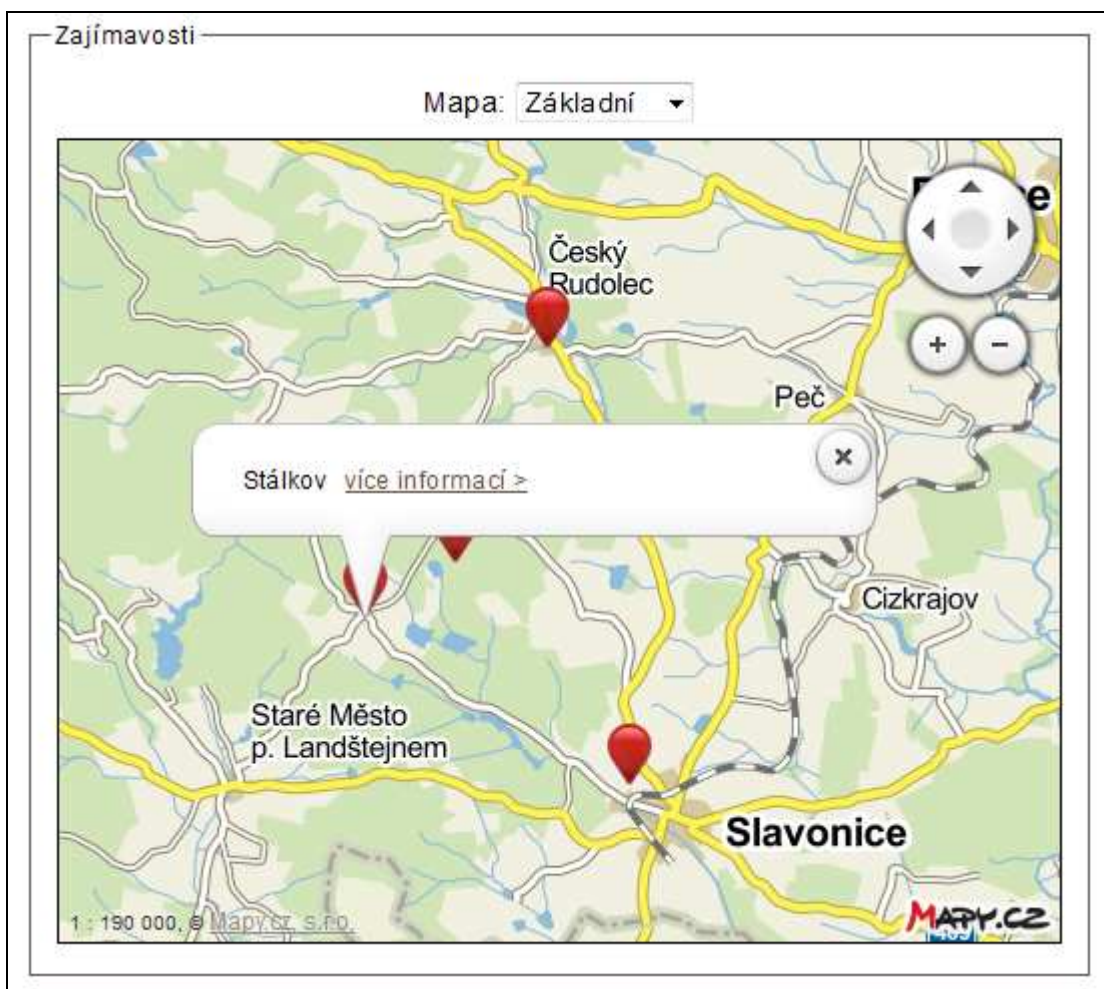
Po odeslání formuláře dojde ke kontrole povinných položek, u názvu a sekce se kontroluje, zda se nejedná o prázdné proměnné, a u zeměpisných souřadnic se ověří jejich číselná hodnota, která musí být v rozmezí 48.9°- 49.2° pro severní šířku a 14.7°-15.6° pro východní délku, přičemž je takto vymezené území stanoveno jako širší okolí obce Český Rudolec. Pokud některé z proměnných nevyhovují kontrole, je uživatel vrácen na formulář pro přidání, který obsahuje všechny zadané hodnoty a na chybné hodnoty je upozorněn chybovou hláškou ve formě tabulky *StatusTable*.

Stejný formulář, jaký byl popsán výše pro vložení nového bodu, se zobrazuje i v případě, kdy dochází ke změně informací u již existujícího bodu. U změnového formuláře však dochází k načtení aktuálních informací o bodu z databáze a tyto jsou následně vloženy do textových polí.

Pokud jsou vloženy hodnoty v pořádku, dojde k vygenerování šestimístné proměnné *Id*, která bude jednoznačně identifikovat novou položku v databázi. Pokud je společně s formulářem odeslán i soubor, dojde k vygenerování šestimístného kódu pro identifikaci souboru a k složení nového názvu souboru z identifikátoru sekce, do které položka spadá, z *Id* položky, ke které obrázek patří, a z identifikátoru obrázku. Obrázek je následně uložen do složky *root/upload/points/original*. Originální obrázek je poté načten a prostřednictvím funkce *resampleImg* zmenšen na limitní rozměry 800x800px a 200px150px a zmenšené soubory uloženy do složek *w800* a *w120* ve složce *root/upload/points*. Následuje vykonání SQL příkazu a uložení nové položky do databáze. O uložení nového bodu zájmu je uživatel informován prostřednictvím tabulky *StatusTable* s odkazem *Přidat další bod*, směřujícím na formulář pro přidávání nových bodů a odkazem *Prohlížet body zájmu*, který odkazuje na výpis všech bodů zájmu v databázi.

Výpis obsahu tabulky *Rud\_Points* je prováděn prostřednictvím souboru *index.php* ze složky *root/plugins/points*. Tento soubor nabízí hned několik forem výpisu obsahu. Jednou z forem je výpis všech bodů zájmu v databázi. Tato forma je určena pro administrátory webu a vypisuje body tematicky pod sebou podle příslušnosti k sekci. Zároveň je možné vypisovat body pouze z určité sekce. Výpis jednotlivých sekcí je prováděn v prvku *fieldset* s nadpisem vypisované sekce. V tomto prvku jsou vypisovány jednotlivé body zájmu opět jako prvek *fieldset*, který je pomocí funkce v jazyku *JavaScript* omezen pouze na obvod prvku a jeho nadpis. Prvek se rozbalí teprve po kliknutí na něj a zobrazí se tak informace o bodu zájmu. Tyto informace je opět možné omezit pouze na obvod prvku *fieldset* dvojklikem na něj.

Další možností je výpis ve formě mapy. Tato forma je určena pro běžného návštěvníka webu. K tomuto účelu je využíván web *Mapy.cz*, a to prostřednictvím jejich rozhraní API verze 4.7, která je blíže popsána v části 5.1 *Mapy.cz* API. Výpis ve formě mapy je k dispozici v rozměru 500px šířky a 400px výšky. Na stránku je nejdříve načten externí skript *http://api4.Mapy.cz/loader.js* v jazyku *JavaScript*, který zajišťuje načítání samotného mapového podkladu. Nad rámem mapy je k dispozici přepínací tlačítko prvku *select*, které umožňuje přepínání mezi vrstvami mapy. K dispozici jsou zde vrstvy základní, letecká a turistická. V takto načteném okně se poté zobrazují jednotlivé body zájmu načtené z tabulky *Rud\_Points*. Program v souboru *index.php* načte z databáze údaje o jednotlivých bodech a doplní potřebnými informacemi API skript, který poté zajistí jejich výpis a zobrazení v mapě. Okno s mapou umožňuje ovládání prostřednictvím myši a zároveň pomocí tlačítka pro posun a změnu velikosti mapy. Výpis jednotlivých bodů následně probíhá formou ukazatele na mapě, přičemž po kliknutí na něj se uživateli zobrazí tabulka s názvem bodu, která uživateli umožňuje přejít prostřednictvím odkazu na podrobné informace o bodu, případně jej zavřít křížkem v pravém horním rohu a pokračovat dále v prohlížení mapy.



Obr. 4.13 Ukázka mapy pro zobrazení všech bodů se zobrazenou vizitkou jednoho z bodů

Při prohlížení jednotlivých bodů a podrobných informací o nich jsou z databáze načteny veškeré informace o tomto bodu, které se následně zobrazují v HTML prvku *fieldset*. Tento prvek je opatřen popisem s názvem zobrazovaného bodu zájmu, uvnitř prvku je vlevo nahoře zobrazena miniatura obrázku načtená ze složky *root/upload/points/w120*, která slouží zároveň jako odkaz na nové okno, ve kterém se zobrazuje stejný obrázek, avšak jako větší verze ze složky *root/upload/points/w800*. Pod miniaturou obrázku se načítá malá mapa o rozměrech 200px X 200px, která zároveň slouží jako odkaz na nové okno s velkou mapou o rozměrech 800px X 600px, se zobrazenou polohou bodu. V pravé horní části, vedle miniatury obrázku, se zobrazují textové informace o bodu, v pravé spodní části, vedle náhledu mapy, se pak zobrazují zeměpisné souřadnice bodu ve formátu  $SS^{\circ} MMM.MM^{\circ}$  a odkaz na případný web o příslušném bodu, který se po kliknutí na odkaz otevírá v novém okně.

Zobrazují-li se informace o bodu uživateli s právy *Admin*, zobrazí se nad miniaturou obrázku odkazy pro práci s bodem. Tlačítko *Z* odkazuje na formulář pro změnu informací o bodu, tlačítko *S* pak odkazuje na skript pro smazání bodu a v případě kliknutí na něj je uživatel kontrolně dotázán, zda chce skutečně bod smazat, aby byl nucen tuto volbu opětovně potvrdit a nedocházelo tak k náhodnému smazání bodu.

STÁLKOV

[Z][S]



Stáلكov (německy Stallek) byl dříve samostatnou obcí 6 km severozápadně od Slavonic v nadmořské výšce 637 m. Malá obec, ve které žilo německé obyvatelstvo, po roce 1918 s českou menšinou, po r. 1945 české obyvatelstvo. V současné době je obec místní částí Slavonic a je v ní 34 obytných domů, z nichž necelá polovina je obydlena trvale.



Souřadnice místa [ MAPY.CZ ]  
 N 49 ° 1.682 '  
 E 15 ° 17.065 '  
<http://www.stalkov.cz>

© Mapy.cz s.r.o. MAPY.CZ

Obr. 4.14 Ukázka výpisu podrobných informací o bodu včetně administrátorských tlačítek

Zobrazení zvětšené verze obrázku u jednotlivých bodů probíhá v samostatném okně zcela mimo grafiku webu. V horní části webu se zobrazuje název bodu, ke kterému je obrázek přiřazen, vedle názvu se pak zobrazuje odkaz sloužící pro zavření okna. Pod těmito textovými informacemi se zobrazuje samotný zvětšený obrázek o limitních rozměrech 800px X 800px a po kliknutí na něj dojde k uzavření samostatného okna.

Zobrazení zvětšené mapy o rozměrech 800px X 600px je prováděno v samostatném okně, taktéž mimo grafiku webu. V horní části je uveden název bodu, který se zobrazuje na mapě, vedle názvu jsou zobrazeny dva odkazy, z nichž jeden slouží pro zavření okna a druhý odkazuje na web Mapy.cz, kde se zobrazí poloha bodu na mapě a je možné dále využívat všechny nástroje, které web Mapy.cz poskytuje. Na dalším řádku je k dispozici prvek *select*, který slouží k přepínání mezi vrstvami mapy. Mapa nabízí vrstvy základní, letecká a turistická. Ve spodní části se pak zobrazuje vlastní mapa se zobrazenou polohou bodu, kterou je možné ovládat jak prostřednictvím myši, tak pomocí tlačítek pro posun mapy a změnu velikosti mapy. Tlačítka pro ovládání mapy jsou zobrazena v jejím pravém horním rohu.

Možnost využití více vrstev mapy byla zvolena cíleně s ohledem na přehlednost a čitelnost jednotlivých vrstev mapy. *Obecná mapa* má uživateli poskytnout širší pohled na polohu bodu, jeho dopravní dostupnost a jeho vztah vůči dalším bodům v rámci širšího

území. *Turistická mapa* má uživateli poskytnout přehled o poloze bodu ve vztahu k blízkým objektům, především dalším turistickým cílům a nastínit topografické skutečnosti lokality bodu. *Letecká mapa* je pak určena pro získání konkrétní představy o bezprostředním okolí bodu a vztahu bodu k sousedním objektům.

#### 4.2.7 Trasy

Modul trasy je uložen ve složce *root/plugins/trasy* a slouží jako obslužný modul pro část webu *Tipy na výlet*. Modul je založen na databázové tabulce *Rud\_Trasy*, kde se evidují u každé vložené trasy údaje jako Id trasy, počet zastavení na trase, celková délka trasy, název, pojmenování příloženého souboru, čas a datum vložení, uživatele, který trasu přidal a IP adresu, ze které k přidání došlo.

Pro práci s daty je zde k dispozici soubor *admin.php*, který slouží pro přidávání a mazání tras. Soubor obsahuje formulář pro vytvoření nové trasy. Tento formulář obsahuje políčka pro vložení názvu trasy, počtu zastavení na trase, včetně počátečního a koncového bodu, délky trasy, včetně jednotek a možnost přiložit soubor, například s mapou trasy. Položky název, délka a počet zastavení trasy jsou povinné, což je kontrolováno po odeslání formuláře. Pokud jsou tyto položky vyplněny, dojde k uložení nové trasy do databáze, v opačném případě je uživatel vrácen na stránku s formulářem, kde jsou vyplněna políčka tak, jako před odesláním, přičemž nad formulářem se zobrazí chybová hláška ve formě *StatusTable* o důvodech neuložení trasy. Pokud je trasa bez problémů uložena, je o tom uživatel informován tabulkou formátu *StatusTableAdmin* s možností pokračovat dále na přidání bodu trasy nebo na stránku s výpisem tras.

Vytvoření nové trasy

Název trasy: (!)

Počet zastávek (minimálně 3) (!) :

Délka trasy (zadejte i jednotky) (!) :

Mapka k této trase (není povinná):

Procházet...

Uložit

Obr. 4.15 Ukázka formuláře pro přidání nové trasy

Pro přidání nového bodu trasy slouží modul *Points*, jak již bylo popsáno výše a přidávání probíhá prostřednictvím stejného formuláře. Data o jednotlivých zastaveních trasy jsou tak uchovávána v tabulce *Rud\_Points*. Pro přidávání nových bodů trasy je formulář ošetřen tak, že je možné přidat pouze doposud nepřidaná zastavení na trase. Již



přidaná zastavení se nezobrazují v hodnotách tlačítka *select* pro volbu trasy a čísla zastavení.

K prohlížení výletních tras slouží soubor *index.php*, který vypisuje pod sebou názvy jednotlivých tras s uvedením délky trasy, počtu zastavení a odkazem na příložený soubor. Po kliknutí na vlastní název mapy se uživateli zobrazí výpis informací o prvním zastavení trasy, který je prováděn modulem *Points*. Zobrazují se zde tedy stejné informace, jako u výpisu podrobných informací o bodech zájmu. Oproti klasickému bodu zájmu je zde navíc zobrazeno schéma trasy v podobě hnědých koleček. Tato kolečka mají hnědé vyplnění a zobrazují jednotlivá zastavení trasy, přičemž slouží zároveň jako odkazy na stránku s podrobnými informacemi o jednotlivých zastaveních. Aktuální zastavení je zobrazeno vždy zobrazeno s bílou výplní, aby měl uživatel vždy přehled, kde se na trase nachází. Pro plynulý posun po trase, tedy buď vpřed, případně vzad, je na začátku a na konci schématu obrázek se šipkou, který odkazuje na předchozí nebo následující zastavení trasy. Pod tímto schématem je uveden název trasy a v závorce pak celková délka trasy.

GRASELOVA SLUJ



Putování s Grázlem Českou Kanadou (25 Km)

[\[Z\]](#) [\[S\]](#)



Graselova sluj je malá jeskyně, kde se údajně v roce 1814 skrýval obávaný lapka Jan Jiří Grázl.



Souřadnice místa [ MAPY.CZ ]  
N 49 ° 2.108 '  
E 15 ° 18.314 '

Obr. 4.16 Ukázka výpisu podrobných informací u bodu trasy

#### 4.2.8 JavaScript

Pro dílčí zpracování dat a usnadnění práce se systémem bylo využito několik jednoduchých funkcí naprogramovaných ve skriptovacím jazyku *JavaScript*. Tyto funkce jsou uloženy v samostatném souboru *js.js*, který je uložený ve složce *root/js*. Tento soubor obsahuje funkce:

- *noveOkno(adresa,width,height)*, která slouží k otevření nového okna událostí *Javascriptu*, okno je šířky *width*, výšky *height* a je v něm načten soubor uložený na adrese *adresa*. Tato funkce je používána pro otevírání nových oken se samostatnými fotografiemi.

- *noveOknoFoto(adresa,width,height)* je totožná s předchozí funkcí, je pouze určená pro využití v jiných částech webu, kde lze předpokládat do budoucna změnu.

- *schovPrvek(element, viditelnost)* nastaví viditelnost prvku u objektu *element* na hodnotu *viditelnost*. Je však potřeba znát, zda je prvek viditelný nebo skrytý. Tato funkce slouží pro schovávání některých prvků, které je potřeba zobrazit při určité události.

- *ukazPrvekAK(element)* plní podobnou funkci jako předchozí funkce, avšak s tím rozdílem, že prvek buď skrývá, nebo zobrazuje a jeho aktuální vlastnost si zjistí sama.

- *Insert(S)* slouží pro vložení textového řetězce *S* do textového pole formuláře. Je využíváno například u vkládání HTML tagů do formulářových polí.

- *InsertC(S)* plní stejnou funkci jako předchozí funkce, jen je využívána v jiných částech webu, kde lze do budoucna předpokládat změnu.

- *UkazVolby()* slouží v modulu *Points* pro zobrazení formulářových políček pro volbu trasy a vzdálenost k dalšímu bodu, pokud je sekce bodu zvolena jako *Výletní trasy*.

#### 4.2.9 Status

Tento modul slouží pro zobrazování hlášek událostí webu, například při odhlášení nebo přihlášení uživatele, při neplatném pokusu o přihlášení, při volání neexistující stránky, při přidání nebo odebrání zařízení. Hlášky jsou zpravidla vypisovány se formátu tabulky *StatusTable* nebo *StatusTableAdmin*, případně jsou doplněny jednoduchým formulářem, který s událostí souvisí (např. formulář pro přihlášení apod.).

## 5 API

*API*, neboli *Application Programming Interface*, je pojem označující rozhraní pro programování aplikací, volně přeloženo se jedná o *programovací rozhraní aplikace*. *API* nabízí soubor různých procesů, funkcí a menších programů, které může programátor využívat. *API* určuje to, jak jsou funkce takové knihovny volány a stanovuje podmínky pro práci s nimi. *API* pak poskytuje programátorovi možnost využívat jeho jednotlivé funkce místo toho, aby je musel programovat. Obecně lze *API* rozdělit podle toho, k jakým účelům je používán:

- *Obecné API* poskytuje uživateli kompletní soubor rozhraní *API* v podobě ucelené knihovny programovacího jazyka.

- *Konkrétní API* poskytuje funkce pro řešení konkrétních událostí.

- *Jazykově závislé API* poskytuje pouze možnost volání v konkrétním programovacím jazyku, a to pomocí konkrétních prvků tohoto jazyka

- Jazykově *nezávislé API* poskytuje možnost volání z různých programovacích jazyků a není tak svázána pouze jedním systémem. Takto tvořené *API* je pak možné využít například jako webovou službu typu *Mapy.cz API*. [7]

## 5.1 Mapy.cz API

Pro využití v geo-web aplikaci Rudolecko byl zvolen mapový portál Mapy.cz a služba *API* poskytovaná tímto serverem. Hlavním hodnotícím kritériem pro využití této služby bylo přehledné a snadné využívání jejich *API*, snadné a intuitivní ovládání okna s mapou a především kvalita nabízených mapových podkladů se zohledněním účelnosti jejich jednotlivých vrstev, která byla popsána v závěrečném odstavci části 4.2.6 Points.

V současné době je k dispozici *API* označené jako verze 4.7, které nabízí možnost práce s třemi mapovými podklady. Toto rozhraní je přístupné na adrese <http://api4.mapy.cz/>. společně s dokumentací k jejímu využití a ukázkami využití. Toto rozhraní umožňuje umístit na web klasickou mapu, stejnou jako nabízí na svých stránkách web Mapy.cz. Pro využití *API Mapy.cz* jsou stanoveny smluvní podmínky, které nabízejí možnost využít tuto službu i pro komerční účely.

```
<!doctype html>
<html>
<head>
  <script src="http://api4.mapy.cz/loader.js"></script>
  <script>Loader.load()</script>
</head>

<body>
  <div id="mapa" style="width:600px; height:400px;"></div>
  <script type="text/javascript">
    var stred = SMap.Coords.fromWGS84(14.41, 50.08);
    var mapa = new SMap(JAK.gel("mapa"), stred, 10);
    mapa.addDefaultLayer(SMap.DEF_BASE).enable();
    mapa.addDefaultControls();
  </script>
</body>
</html>
```

Obr. 5.1 Ukázka zdrojového kódu pro jednoduché vložení mapy do webové stránky

Samotná služba *API Mapy.cz* je pro uživatele velmi přehledně popsána na výše uvedené webové adrese společně s ukázkami použití. V části *Ukázky* uživatel nalezne několik konkrétních ukázek využití této služby, které jsou rozděleny podle formy a typu využití. Rozdělení využití je zde na části:

- *Základy*, které nabízí vložení jednoduché mapy, využití výchozích ovládacích prvků mapy, přidávání samostatného ovládacího prvku, dynamické měnění velikosti mapy, odečítání souřadnic z mapy, signály – interní události, které slouží pro práci s událostmi mapy na základě uživatelem odeslaných požadavků pro práci s mapou.

- *Značky* nabízejí možnost vytvoření vrstvy se značkami, práci s konkrétní značkou s možností přidání titulku a vizitky ke konkrétní značce, vložení posouvateľné značky, možnost pracovat s obsahem vizitky, možnost práce s více značkami v mapě a vložení vlastní značky. Značky jsou v mapě definovány souřadnicemi WGS84 ve stupních.

- *Geometrie*, která nabízí možnost práce s vektorovými prvky v mapě, jež jsou zobrazovány v samostatné vrstvě a nabízejí možnost nadefinování vizuálních vlastností prvku a jednoznačného Id pro práci s ním. Jednoduché prvky jsou nadefinovány pomocí souřadnic lomových bodů ve WGS84. Složitější prvky jsou definovány pomocí základních bodů se souřadnicemi WGS84 a řetězcem popisujícím geometricky vlastnosti samotného prvku. Poslední možností je práce s geometrií prostřednictvím nástroje *AJAX*, který načítá v reálném čase informace o poloze bodů ze samostatného souboru *XML*.

- *Vrstvy*, které dávají uživateli možnost manuálně přepínat mezi vrstvami mapy. Vrstvy je potřeba nejprve vytvořit a načíst do nové vrstvy mapy. K přepínání mezi vrstvami pak slouží HTML prvek *select* a na základě změny jeho hodnoty dochází i k zobrazení zvolené mapy. Práci s vrstvami umožňuje také vestavěný nástroj pro přepínání vrstev mapy, který se zobrazuje v levém horním rohu mapy. Další možností využití vrstev je vrstva *ptačí pohled*, která používá pro interpretaci speciální asynchronní projekci, která je pro každé místo na zemském povrchu jiná a využití této funkce je náročnější, jelikož je *API* nuceno pro zobrazení mapy pokaždé dotazovat server na parametry zobrazení. Poslední možností pro práci s vrstvami je využití vrstvy typu *SMART*, která mění zobrazení mapového podkladu na základě přiblížení mapy.

- *GPX* nabízí možnost práce se standardizovaným formátem souborů pro GPS s koncovkou \*.gpx. Obsah takového souboru se vloží do připraveného textového pole, po odeslání dojde k vytvoření nového *XML* souboru a ten je následně načten a zobrazen v mapě. Další možností této funkce je přímo volání souboru z webové adresy pomocí nástroje *AJAX*.

- *Geokódování/Plánování* nabízí tři možnosti různého využití. Samotné geokódování nabízí buď vyhledání zeměpisné polohy objektu na základě hledaného řetězce (např. název ulice), nebo vyhledání zeměpisného objektu na základě zeměpisných souřadnic. Druhou možností je tzv. *reversní kódování*, které nabízí možnost zjištění zeměpisných objektů přímo z mapy, prostřednictvím zadání souřadnic kliknutím do mapy. Třetí možnost nabízí vyhledávání a plánování trasy, která vzniká na základě vyhledání objektů pomocí geokódování a následné zobrazení v mapě pomocí *geometrie* popsané výše.

- *Ostatní* nabízí možnost vložení *Mini-API*, neboli malé mapky s polohou bodu, která sice není interaktivní, ale o to rychleji se načítá. Další možností je vložení kontextové nabídky, která se zobrazí po kliknutí pravým tlačítkem myši v mapě a v současné době nabízí pouze zobrazení souřadnic bodu a volby pro přiblížení a oddálení mapy. Třetí možností je vložení náhledové mapky do plochy celé mapy, která zobrazuje viditelnou část mapy v jiném přiblížení. Čtvrtou možností je Mercatorova projekce. Tato funkce nabízí projekci z referenční plochy do roviny u jiných mapových podkladů prostřednictvím jiné projekce. Web *Mapy.cz* používá standardně mapové podklady v projekci UTM33, tato funkce pak nabízí možnost využití Mercatorovy projekce nebo projekce pro šikmé snímky.

## 5.2 Využité funkce API Mapy.cz

Z výše popsaných funkcí byly vybrány takové, které vyhovovaly použití v rámci celé aplikace Rudolecko a tyto pak byly zakomponovány do příslušných sekcí webu.

Pro zobrazení všech bodů zájmu z databáze byla nejprve vytvořena mapa, a to stejným způsobem, jak je vidět na obrázku 5.1. Na obrázku 5.2 je poté vidět konkrétní skript, jehož jednotlivé části a řádky jsou popsány v následujícím odstavci.

```

1 <div id="mapa" style="width:500px; height:400px; border: 1px solid;"></div><script type="text/javascript">
2 var obrazek = "http://api4.mapy.cz/img/api/marker/drop-red.png";
3
4 var m = new SMap(JAK.gel("mapa"));
5 m.addControl(new SMap.Control.Sync()); /* Povolení ovalidání mapy */
6 m.addDefaultLayer(SMap.DEF_BASE).enable(); /* základní mapa */
7 var mouse = new SMap.Control.Mouse(SMap.MOUSE_PAN | SMap.MOUSE_WHEEL | SMap.MOUSE_ZOOM); /* Ovládání myši */
8 m.addControl(mouse);
9 m.addDefaultControls();
10
11 var data = {
12 "Bukový vrch/hi64ap": "49.04425\N,15.27605\E",
13 "Rybník Vrzalák/hi2hui": "49.06645\N,15.324383333333\E",
14 "Schillerovy kameny/po3prz": "49.03575\N,15.303883333333\E",
15 };
16 var znacky = [];
17 var souradnice = [];
18

```

Obr. 5.2 Skript pro vytvoření mapy a dat o bodech zájmu

Na obrázku 5.2 se nyní podíváme na funkci celého skriptu pro zobrazení mapy a bodů zájmu v mapě. Na řádce 1 je prostřednictvím HTML prvku *div* vytvořen objekt mapy s identifikátorem *mapa*, který následně slouží pro práci s tímto objektem prostřednictvím funkcí vytvořených v programovacím jazyku *JavaScript*. Na řádce 2 je definován výchozí ukazatel pro polohu zobrazovaných bodů. Prvek mapy je skriptem načten jako proměnná *m*, jak je patrné na řádce 4, s jejímiž vlastnostmi se následně pracuje prostřednictvím dalších funkcí *API*. Na řádcích 5-7 je povoleno ovládání mapy pomocí myši. Na řádcích 11-15 jsou načteny jednotlivé body zájmu s vlastnostmi *Název bodu/id* v *databázi* a *souřadnice bodu* ve *WGS84*. Tyto řádky jsou dynamicky generovány skriptem *PHP*, který získává údaje o bodech z interní databáze a předává je dále funkci v jazyku *JavaScript*. Aby bylo možné odkazovat z mapy přímo na stránku s detailními informacemi o konkrétním bodu zájmu, je jeho databázový identifikátor vložen za název bodu, od něhož je oddělen lomítkem. Toto opatření bylo nutné kvůli nemožnosti přidat další vlastnosti do již předdefinované funkce pro práci s bodem. Na řádcích 16 a 17 dochází k vytvoření prázdných proměnných *nazev* a *souradnice*, které jsou v další fázi skriptu naplněny daty v proměnné *data*, jak je popsáno v dalším odstavci.

Obrázek 5.3 nám nyní přiblíží, jak provedeme načtení dat o bodech a z nich vytvoříme značky, které následně načteme do samostatné vrstvy mapy. Na řádce 22 dojde k provedení cyklu *for*, který probíhá, dokud nejsou načteny všechny údaje z proměnné *data*. Tento cyklus postupně projde všemi položkami proměnné *data*, na řádce 23 získá souřadnice bodu a ty uloží do proměnné *c*. Na řádce 25 skript načte název bodu do dvourozměrného pole tak, že obsah rozdělí pomocí lomítka a tyto dvě části umístí do pole. Z tohoto pole pak nadefinuje název bodu na řádce 26 a ID bodu v databázi na řádce 27. Tímto krokem je vyřešen problém s předáváním databázového identifikátoru, který byl popsán výše. Na řádcích 29-34 dojde k načtení vlastností ukazatele polohy bodu, na řádce 30 přiřazení adres obrázku ukazatele, řádek 31 přiřadí ukazateli titulek s názvem bodu, řádek 32 poté vytvoří definiční bod ukazatele. Aby se po kliknutí na ukazatel zobrazila vizitka s názvem bodu a odkazem na detailní informace o bodu, dochází na řádce 36 k vytvoření takovéto vizitky, jejíž obsah je následně přiřazen na řádce 38 a 39 (ve skutečnosti je toto provedeno na jednom řádku, pro potřeby této práce a jejího grafického doplnění byl tento řádek uměle rozdělen na dva řádky z důvodu lepší přehlednosti kódu). Na řádce 42 vytvoří skript značku pro bod a přiřadí jí výše získané hodnoty souřadnic a vzhledu značky. Na řádce 43 je pak ke značce přiřazena vizitka, řádek 44 přidá souřadnice do proměnné *souradnice* a značku bodu do proměnné *znacky*. Na řádce 47 dojde

k vytvoření vrstvy pro značky, která je na řádce 48 přidána jako nová vrstva mapy a na řádce 49 je tato vrstva aktivována. Cyklus *for* na řádcích 50-52 postupně přidává nové značky do vytvořené vrstvy *znacky*. Aby se zobrazily na jednom náhledu mapy všechny značky, které byly přidány, spočítá skript na řádce 54 střední pozici ze získaných souřadnic a nastaví jí na řádce 55 jako střed mapy.

```

22 for (var name in data) { /* Vyrobit značky */
23     var c = SMap.Coords.fromWGS84(data[name]); /* Souřadnice značky, z textového formátu souřadnic */
24
25     pole = name.split("/")
26     var nazev = pole[0]
27     var pid = pole[1]
28
29     var options = {
30         url:obrazek,
31         title:nazev,
32         anchor: {left:10, bottom: 1} /* Ukotvení značky za bod uprostřed dole */
33     }
34
35
36     var card = new SMap.Card();
37
38     card.getHeader().innerHTML = "<div class='sekce_ltext'>"+nazev+"
39     <a href='point-info-"+pid+"' title='Podrobné informace' class='points'>více informací </a></div>";
40
41
42     var znacka = new SMap.Marker(c, null, options);
43     znacka.decorate(SMap.Marker.Feature.Card, card);
44     souradnice.push(c);
45     znacky.push(znacka);
46 }
47 var vrstva = new SMap.Layer.Marker(); /* Vrstva se značkami */
48 m.addLayer(vrstva); /* Přidat ji do mapy */
49 vrstva.enable(); /* A povolit */
50 for (var i=0;i<znacky.length;i++) {
51     vrstva.addMarker(znacky[i]);
52 }
53
54 var cz = m.computeCenterZoom(souradnice); /* Spočítat pozici mapy tak, aby značky byly vidět */
55 m.setCenterZoom(cz[0], cz[1]);

```

Obr. 5.3 Skript pro vytvoření bodů zájmu a jejich načtení do samostatné vrstvy mapy

Obrázek 5.3 ukazuje, jak skript načte další vrstvy mapy a definuje funkci pro jejich přepínání. Na řádce 57 je vytvořena proměnná *layers*, do které jsou načteny postupně nové vrstvy pro základní mapu, ortofotomapu, hybridní mapu a turistickou mapu. Toto je provedeno postupně na řádcích 59-62. Na řádce 63 je zapnuta vrstva se základní mapou a na řádce 64 povoleno její ovládání. Na řádcích 66-80 je definována funkce *switchLayer*, která nejprve na řádce 67 vypne pomocí cyklu *for* všechny vrstvy a pomocí cyklu *switch* aktivuje, na základě vložených argumentů, jednu z vrstev mapy, přičemž jednotlivé vrstvy jsou tvořeny příslušnou mapou, pouze vrstva *letecká* je tvořena ortofotomapou a hybridní mapou. Na řádcích 82-90 je nadefinován HTML prvek *select* pro přepínání jednotlivých vrstev, přičemž na řádce 82 je získávána jeho aktuální hodnota, která je následně použita pro přepínání mapy. Samotné přepínání mapy provádí řádek 91, a to s pomocí funkce *switchLayer*.

```

57 var layers = {};
58
59 layers[SMap.DEF_BASE] = m.addDefaultLayer(SMap.DEF_BASE);
60 layers[SMap.DEF_OPHOTO] = m.addDefaultLayer(SMap.DEF_OPHOTO);
61 layers[SMap.DEF_HYBRID] = m.addDefaultLayer(SMap.DEF_HYBRID);
62 layers[SMap.DEF_TURIST] = m.addDefaultLayer(SMap.DEF_TURIST);
63 layers[SMap.DEF_BASE].enable(); /* pro začátek zapnout základní podklad */
64 m.addDefaultControls();
65
66 var switchLayer = function(e, elm) {
67     for (var p in layers) { layers[p].disable(); }
68     switch (elm.selectedIndex) {
69         case 0:
70             layers[SMap.DEF_BASE].enable();
71             break;
72         case 1:
73             layers[SMap.DEF_OPHOTO].enable();
74             layers[SMap.DEF_HYBRID].enable();
75             break;
76         case 2:
77             layers[SMap.DEF_TURIST].enable();
78             break;
79     }
80 }
81
82 var s = JAK.mel('select');
83 JAK.gel('controls').innerHTML = 'Mapa: ';
84 JAK.gel('controls').appendChild(s);
85 var names = ['Základní', 'Letecká', 'Turistická']
86 for (var i=0; i<names.length; i++) {
87     var o = JAK.mel('option');
88     o.innerHTML = names[i];
89     s.appendChild(o);
90 }
91 var e = JAK.Events.addListener(s, 'change', window, switchLayer);

```

Obr. 5.4 Vytvoření vrstev pro další mapové podklady a funkce pro jejich přepínání

Fungování aplikace tak, jak bylo popsáno výše, je použito obdobným způsobem jak u výpisu miniatury u podrobných informací o konkrétním bodě, tak u zobrazení samostatné mapy pro konkrétní bod zájmu. U těchto dvou forem zobrazení mapy je pak zdrojový kód zjednodušen o nepoužité prvky a funkce, jelikož zde dochází pouze k zobrazení polohy jediného bodu, případně není povolena možnost přepínat vrstvy mapy.

## 6 ZÁVĚR

Výsledkem této práce je komplexní systém, který bude prozatím k dispozici na webu <http://rudolecko.knezicek.cz> a bude sloužit především k testování celé aplikace, seznámení s aplikací pro uživatele, kteří budou v budoucnosti tvořit administrační tým webu a samozřejmě k představení této aplikace jako závěrečné práce mého bakalářského studia. Zároveň se předpokládá odladění jejích drobných chyb a bezpečnostních děr, které se u aplikace tohoto charakteru dají očekávat, stejně jako u každé nové webové aplikace.

Aplikace byla po celou dobu jejího vytváření testována v prohlížeči Mozilla Firefox, ve kterém je zaručeno jeho bezchybné fungování. Využití v dalších prohlížečích je pochopitelně možné, ale lze u nich předpokládat chyby ve fungování některých funkcí a rozdíly ve vizuálních prvcích webu.

Samotná aktuální forma aplikace je pouze základem celého webu Rudolecko.cz a v budoucnu bude rozšířena o další moduly a funkce, které budou vytvořeny na základě požadavků uživatelů, administračního týmu a s ohledem na to, do jaké míry bude aplikace využívána veřejností. Do budoucna je předpokládána spolupráce s nově budovaným informačním turistickým centrem v Českém Rudolci, jehož zaměstnanec by měl být součástí administračního týmu. Aplikace je tedy prvním krokem k nastartování propagace celé oblasti Rudolecka, o jehož potenciálu bylo dosti řečeno v úvodu této práce.

Zprovoznění aplikace a její oficiální uvedení do provozu na doméně Rudolecko.cz je předpokládáno během června až července 2012.



## ZDROJE INFORMACÍ

- [1] František Šokoň, info web Česká Kanada. [online]. [2004-2012] [cit. 2012-05-13].  
dostupný z WWW: < <http://sokon.cekanka.com/dilo.php> >
- [2] Wikipedie, otevřená encyklopedie. [online]. [2001] [cit. 2012-05-12].  
dostupný z WWW: <[http://cs.wikipedia.org/wiki/%C4%8Cesk%C3%A1\\_Kanada](http://cs.wikipedia.org/wiki/%C4%8Cesk%C3%A1_Kanada)>
- [3] TIRAY, Jan. Vlastivěda moravská, Slavonický okres. Musejní spolek v Brně. Brno 1926
- [4] Zámek Český Rudolec. [online]. [2012] [cit. 2012-05-12].  
dostupný z WWW: <<http://www.zamek-ceskyrudolec.cz>>
- [5] KOSEK, Jiří. PHP – Tvorba interaktivních webových aplikací. Grada Publishing spol. s.r.o. Praha 1998, ISBN 80-7169-373-1
- [6] API Mapy.cz – Návod k použití API. [online]. [2012] [cit. 2012-05-12].  
dostupný z WWW: <[http:// http://api4.Mapy.cz](http://http://api4.Mapy.cz)>
- [7] Wikipedie, otevřená encyklopedie. [online]. [2001] [cit. 2012-05-12].  
dostupný z WWW: < <http://cs.wikipedia.org/wiki/API>>

## SEZNAM POUŽITÝCH ZKRATEK

WWW – World wide web, označení pro aplikace protokolu HTTP

CERN - Evropská organizace pro jaderný výzkum

HTTP - Hypertext Transfer Protocol, protokol sloužící pro výměnu hypertextových dokumentů

URL - Uniform Resource Locator, řetězec sloužící ke zpecifikaci zdroje informací

HTML - HyperText Markup Language, značkovací jazyk hypertextových souborů

CGI - Common Gateway Interface, protokol pro propojení externích aplikací s webovým serverem

Java – objektivě orientovaný programovací jazyk

ASP – skriptovací jazyk od společnosti MicroSoft

MySQL – databázový systém multiplatformní databáze

CSS - Cascading Style Sheets, kaskádové styly sloužící pro popis zobrazení HTML stránek

PHP - Hypertext Preprocessor, programovací jazyk pro programování webových aplikací

CMS - anglického content management system, česky systém pro správu obsahu

UTM – Univerzální transverzální Mercatorův systém souřadnic  
WGS84 – World Geodetic System 1984, světový souřadnicový systém  
GPS – *Global Positioning System*, veřejně dostupný navigační systém  
HASH - je matematická funkce pro převod vstupních dat do malého čísla

## **SEZNAM OBRÁZKŮ**

- 2.1: Podoba zámku z 18. století a z doby kolem roku 1900
- 2.2: Zámecký areál a přilehlé okolí
- 4.1: Grafická podoba webu
- 4.2: Ukázka instalačního souboru
- 4.3: Ukázka formuláře pro přihlášení a formuláře pro autorizaci zařízení
- 4.4: Ukázka administrátorského menu uživatele s právy Admin
- 4.5: Ukázka výpisu článku s titulní fotografií a fotogalerií
- 4.6: Ukázka formuláře pro přidání nového článku
- 4.7: Ukázka výpisu fotogalerií s administrátorskými tlačítky pro úpravu
- 4.8: Ukázka formuláře pro přidání nové fotogalerie
- 4.9: Ukázka formuláře pro nahrání nové fotografie do konkrétní fotogalerie
- 4.10: Ukázka výpisu počtu návštěv funkcí počítadlo
- 4.12: Ukázka formuláře pro přidání nového bodu zájmu pro bod na výletní trase
- 4.13: Ukázka mapy pro zobrazení všech bodů se zobrazenou vizitkou jednoho z bodů
- 4.14: Ukázka výpisu podrobných informací o bodu včetně administrátorských tlačítek
- 4.15: Ukázka formuláře pro přidání nové trasy
- 4.16: Ukázka výpisu podrobných informací u bodu trasy
- 5.1: Ukázka zdrojového kódu pro jednoduché vložení mapy do webové stránky
- 5.2: Skript pro vytvoření mapy a dat o bodech zájmu
- 5.3: Skript pro vytvoření bodů zájmu a jejich načtení do samostatné vrstvy mapy
- 5.4: Vytvoření vrstev pro další mapové podklady a funkce pro jejich přepínání

## SEZNAM PŘÍLOH

Obsah přiloženého CD:

**\rudolecko** Obsahuje kompletní zdrojové kódy webové aplikace Rudolecko.

**\css** – Obsahuje soubor s definovanými kaskádovými styly.

**\images** – Obsahuje obrázky použité na webu.

**\js** – Obsahuje soubor s předdefinovanými funkcemi JavaScriptu.

**\plugins** – Obsahuje složky s jednotlivými moduly:

**\clanky**

**\func**

**\galerie**

**\historie**

**\intranet**

**\odkazy**

**\pocitadlo**

**\points**

**\soucasnost**

**\status**

**\trasy**

**\requires** – Obsahuje soubory, které jsou načítány při použití většiny souborů.

**\start** – Obsahuje soubory pro instalaci redakčního system.

**\upload** – Obsahuje složky, do nichž se ukládají uživatelem nahrávané soubory.

**\clanky**

**\galerie**

**\points**

**\trasy**