

# **Webový portál pro handicapované**

Bc. Rostislav Dočkal

---

Diplomová práce  
2016



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2015/2016

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Rostislav Dočkal**  
Osobní číslo: **A13477**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **kombinovaná**

Téma práce: **Webový portál pro handicapované**  
Téma anglicky: **A Web Portal for Disabled People**

Zásady pro vypracování:

1. Seznamte se a prostudujte webové projekty zabývající se pomocí handicapovaným.
2. Navrhňte vhodné řešení webového portálu pro elektronické zmapování veřejných míst z hlediska jejich přístupnosti pro handicapované.
3. Implementujte daný systém s ohledem na požadavky responzivního designu, tj. možnost přístupu z mobilních zařízení (mobil, tablet apod.).
4. Vytvořte administrátorskou část systému, která bude dovolovat snadnou správu míst a uživatelských účtů.
5. Věnujte pozornost zabezpečení celého systému.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. 1. vyd. Brno: Computer Press, 2012, 439 s. ISBN 978-80-251-3733-8.**
2. **KADLEC, Tim. Responzivní design profesionálně. Vyd. 1. Brno: Zoner Press, 2014, 246 s. Encyklopedie Zoner Press. ISBN 978-80-7413-280-3.**
3. **PEHLIVANIAN, Ara a Don NGUYEN. JavaScript okamžitě. 1. vyd. Brno: Computer Press, 2014, 160 s. ISBN 978-80-251-4163-2.**
4. **FREEMAN, Adam. Pro ASP.NET MVC 5. Fifth edition. New York, New York: Apress, 2013, xxvii, 799 pages. Expert's voice in ASP.NET. ISBN 9781430265290.**
5. **DRISCOLL, Brian. Entity Framework 6 recipes. Second edition. Berkeley: Apress, 2013, xxxiv, 511 pages. ISBN 1430257881.**

Vedoucí diplomové práce:

**doc. Ing. Jiří Vojtěšek, Ph.D.**

Ústav řízení procesů

Datum zadání diplomové práce:

**5. února 2016**

Termín odevzdání diplomové práce:

**20. května 2016**

Ve Zlíně dne 5. února 2016



doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



doc. Mgr. Roman Jašek, Ph.D.  
*ředitel ústavu*

### Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomové práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky. Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 20. 5. 2016

.....  
podpis autora

## **ABSTRAKT**

Diplomová práce se zabývá návrhem a vývojem webové aplikace, která má za úkol poskytnout informace o přístupnosti důležitých míst a společenských institucí ve Zlíně a jeho okolí. Portál obsahuje rovněž administrátorskou část, ve které je možné všechny tyto klíčové informace spravovat. Implementace systému proběhla v jazyce C# s využitím třívrstvé architektury ASP.NET MVC 5. Při vývoji využity technologie jako Microsoft SQL Server, Entity Framework, Bootstrap, Team Foundation Server nebo mapové podklady Google Maps.

Klíčová slova: World Wide Web, bezbariérovost, ASP.NET MVC 5, Microsoft SQL Server, Entity Framework, Google Maps, Bootstrap

## **ABSTRACT**

This Thesis deals with the design and development of a web application that aims to provide information on the accessibility of important places and social institutions in Zlin and its surrounding. The portal also contains the administrator section, where it is possible to manage all important information. System was created in C# language using a three-layer architecture ASP.NET MVC 5. During application development technologies such as Microsoft SQL Server, Entity Framework, Bootstrap, Team Foundation Server and map data from Google Maps were also used.

Keywords: World Wide Web, disabled access, ASP.NET MVC 5, Microsoft SQL Server, Entity Framework, Google Maps, Bootstrap

Tímto bych rád poděkoval vedoucímu mé diplomové práce panu doc. Ing. Jiřímu Vojtěškovi, Ph.D. za důležité rady a doporučení, konzultace s pracovníky Magistrátu města Zlína, vstřícnost a ochotu při konzultacích, ale především za čas, který mi věnoval. Dále bych chtěl poděkovat své rodině a snoubence, kteří po celou dobu studia a psaní mé diplomové práce stáli za mnou a podporovali mě.

## OBSAH

ÚVOD .....	9
<b>I</b> <b>TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1</b> <b>INTERNET A WORLD WIDE WEB (WWW)</b> .....	<b>12</b>
1.1    HISTORIE INTERNETU .....	12
1.2    HYPERTEXT A SLUŽBA WORLD WIDE WEB (WWW) .....	13
1.3    SOUČASNOST SLUŽBY WWW .....	13
<b>2</b> <b>TECHNOLOGIE TVORBY WEBU</b> .....	<b>15</b>
2.1    HTML + CSS + JAVASCRIPT .....	15
2.1.1    HTML .....	15
2.1.2    CSS .....	16
2.1.3    JavaScript .....	17
2.1.4    Responzivní design .....	17
2.2    PHP .....	19
2.3    ASP.NET .....	19
2.4    ASP.NET MVC .....	20
2.5    DATABÁZE .....	21
2.5.1    Microsoft SQL Server .....	22
2.5.2    MySQL .....	22
2.6    OBJEKTOVĚ RELAČNÍ MAPOVÁNÍ (ORM) .....	23
2.6.1    Entity Framework .....	23
2.6.2    NHibernate .....	23
2.7    VERZOVACÍ NÁSTROJE .....	24
2.7.1    Subverison (SVN) .....	24
2.7.2    Team Foundation Server (TFS) .....	24
2.8    MAPOVÉ PODKLADY .....	25
2.8.1    Google Maps .....	25
2.8.2    Mapy.cz .....	25
<b>3</b> <b>EXISTUJÍCÍ WEBY PRO POMOC HANDICAPOVANÝM</b> .....	<b>26</b>
3.1    WEBY S MAPOVÁNÍM PŘÍSTUPNOSTI MÍST .....	27
3.1.1    Vozejkmap.cz .....	27
3.1.2    Mapybezbarier.cz .....	28
3.1.3    Wheelmap.org .....	28
3.2    SHRNUÍ EXISTUJÍCÍCH ŘEŠENÍ .....	30

<b>II</b>	<b>PRAKTICKÁ ČÁST</b> .....	<b>30</b>
<b>4</b>	<b>SPECIFIKACE ZADÁNÍ</b> .....	<b>32</b>
<b>5</b>	<b>NÁVRH APLIKACE</b> .....	<b>34</b>
5.1	PUBLIKACE O BEZBARIÉROVOSTI .....	34
5.2	INFORMACE ZÍSKANÉ Z EXISTUJÍCÍCH ŘEŠENÍ .....	36
<b>6</b>	<b>IMPLEMENTACE, POUŽITÉ TECHNOLOGIE</b> .....	<b>37</b>
6.1	VÝVOJ WEBOVÉ APLIKACE .....	37
6.1.1	TFS (Team Foundation Server) .....	37
6.1.2	Responzivní design webu .....	38
6.1.3	Google Maps – mapové podklady .....	41
6.2	DATABÁZE A ORM .....	44
6.2.1	Entity Framework .....	44
6.2.2	Entity Framework – Migrations (Migrace) .....	46
6.3	ADMINISTRÁTORSKÁ ČÁST SYSTÉMU .....	47
6.3.1	Uživatelé a uživatelské role .....	47
6.3.2	Přihlašování, autorizace a autentifikace .....	48
6.3.3	Správa objektů .....	49
6.3.4	Správa štítků .....	52
6.3.5	Správa přístupností .....	52
6.3.6	Správa typů kontaktů .....	52
6.3.7	Správa stránky O projektu .....	53
6.3.8	Správa uživatelů .....	53
6.4	FILTROVÁNÍ PODLE ŠTÍTKŮ .....	53
6.5	ZABEZPEČENÍ .....	54
6.6	NASAZENÍ DO OSTRÉHO PROSTŘEDÍ A BUDOUCNOST PROJEKTU .....	56
	<b>ZÁVĚR</b> .....	<b>57</b>
	<b>SEZNAM POUŽITÉ LITERATURY</b> .....	<b>59</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK</b> .....	<b>62</b>
	<b>SEZNAM OBRÁZKŮ</b> .....	<b>64</b>
	<b>SEZNAM TABULEK</b> .....	<b>65</b>
	<b>SEZNAM PŘÍLOH</b> .....	<b>66</b>

## ÚVOD

V minulosti bylo možné předávat důležité informace buďto slovem, nebo písemně. S nástupem knihtisku se možnosti razantně změnila a lidé se k informacím dostávali stále častěji. Avšak skutečnou revolucí a nejdůležitější změnou bezprostředně se týkající potřeby předávání informací se v poslední době stalo využití internetu. Konkrétně se jedná o službu World Wide Web (WWW), jenž nám požadované informace poskytuje a zpřístupňuje. Díky ní jsme v dnešní době schopni na počítači s připojením na internet během okamžiku zjistit odpověď na velkou většinu našich otázek. Ať už se týkají našich zájmů, toho, co potřebujeme zjistit kvůli studiu či práci, nebo aktuálních problémů lidstva, o kterých se chceme dovědět víc. Můžeme se jednoduše dostat ke zprávám a informacím z celého světa.

S příchodem chytrých mobilních telefonů se situace ještě více změnila. Dnes již nepotřebujeme být připojeni doma u počítače, abychom si mohli prohlížet internetové stránky. Můžeme se na ně jednoduše připojit pomocí notebooku, mobilního telefonu či tabletu odkudkoli.

Samozřejmě každý z nás si pod pojmem důležitá informace představí něco trochu jiného. Pro některé to může být současné dění ve světě, jiné zajímají informace týkající se oblasti jejich koníčků či zájmů. Pro jiného zase může být důležité zjistit, na kterém úřadu si může vyřídit například cestovní pas kvůli své plánované dovolené. A například pro tělesně postižené lidi v našem okolí je velmi důležitou informací odpověď na otázku, zda se na takový úřad vůbec mohou dostat i na svém invalidním vozíku, jestli je u budovy této instituce možnost parkování apod. Právě na tuto skupinu tělesně handicapovaných spoluobčanů mysleli i pracovníci Magistrátu města Zlín, kteří pro ně připravili publikaci shrnující přehled přístupností důležitých míst ve Zlíně a jeho blízkém okolí, a od kterých také vzešlo jádro zadání mé práce – vytvoření webového portálu, na němž budou lidem k dispozici informace, které pro ně tyto pracovníci získali.

Má diplomová práce se tedy zabývá vytvořením webové aplikace, která bude návštěvníkům poskytovat cenné informace, které se týkají přístupnosti důležitých společenských institucí, jako jsou úřady, zdravotnická zařízení, informační a poradenská centra, kulturní a sportovní zařízení aj. Textová část mé práce je rozdělena do dvou hlavních celků.

V teoretické části jsem se stručně věnoval historii internetu a služby WWW, dále jsem se zaměřil na přehled technologií spojených s tvorbou webových stránek. Před samotným vývojem mé webové aplikace jsem provedl průzkum již existujících řešení s tematikou pomoci handicapovaným lidem. Uvedl jsem zde tedy i informace a postřehy, které jsem touto rešerší zjistil, a které mi byly nápomocny i při vývoji mé aplikace.

Praktická část práce poté začíná specifikací zadání, tedy přesným vymezením požadavků, které má systém splňovat. Dále pokračuje vlastním návrhem, při jehož vytváření jsem zohlednil uvedené požadavky na systém, informace získané v teoretické části práce a rovněž také údaje obsažené ve výše zmíněné publikaci nazvané Vybrané instituce ve Zlíně, Fryštáku, Lípě, Lukově, Sazovicích a Želechovicích nad Dřevnicí z pohledu (bez)bariérovosti. Největší část jsem poté věnoval kapitole týkající se samotné tvorby webového portálu, a to jak z pohledu použitých technologií a postupů, tak i z hlediska vytvořené funkcionality. V závěru jsem se rovněž věnoval zabezpečení systému, a to především proti nejčastějším útokům, které bývají na webové aplikace vedeny.

# I. TEORETICKÁ ČÁST

## 1 INTERNET A WORLD WIDE WEB (WWW)

Jelikož je mým úkolem vytvořit webovou aplikaci, považuji za nutnost uvést na začátek alespoň základní informace o dvou důležitých termínech - internetu a službě World Wide Web, zkráceně WWW.

Internet byl ještě v nedávné minulosti spíše výsadou akademických a vědeckých pracovníků. Přibližně před dvaceti lety se k němu běžný člověk dostal opravdu jen velmi výjimečně. V současnosti je ale situace úplně jiná. S internetem se dnes setkáváme téměř na každém rohu. Využívají ho děti, studenti, dospělí i důchodci a řada z nás si bez něj svůj život nedokáže vůbec představit, protože nám poskytuje celou řadu služeb.[2]

Jako příklad nejpoužívanějších mohu uvést následující služby:

- Webové stránky – WWW (World Wide Web)
- Elektronická pošta - E-mail
- Online komunikace - Instant messaging neboli chat
- Přenos a sdílení souborů – FTP
- Telefonování přes internet – VoIP a mnoho dalších

### 1.1 Historie internetu

Internet coby celosvětová počítačová síť, jenž spojuje jednotlivé menší sítě v jeden celek, začala vznikat v 60. letech minulého století. Hlavní motivací bylo vytvoření počítačové sítě, která by spolehlivě propojila větší množství počítačů, a bylo možné mezi nimi navzájem komunikovat. Rovněž byl kladen důraz i na to, aby byla co nejméně zranitelná - aby i při poškození části její infrastruktury tato síť přežila. Právě z tohoto důvodu byla zvolena tzv. decentralizovaná síť. Tedy taková, u které neexistuje žádný centrální řídicí člen, ale všechny uzly jsou navzájem rovnocenné.[3]

Zde je souhrn nejdůležitějších milníků při vývoji celosvětové počítačové sítě známé jako internet:

- 1969 – Vytvoření experimentální sítě ARPANET, která obsahovala 4 uzly.
- 1972 – Vytvoření první e-mailové aplikace. Autorem je Ray Tomlinson.
- 1980 – Experimentální provoz komunikačního protokolu TCP/IP.
- 1984 – Vyvinut systém doménových jmen (DNS).
- 1987 – Vzniká označení sítě internet.
- 1990 – Ukončení provozu ARPANETu.
- 1991 – Nasazení služby World Wide Web (WWW).

1994 – Internet přechází do komerčního užití.

2000 – Internet používá 250 milionů uživatelů.

2005 – Internet používá více než 1 miliarda uživatelů.[5]

2015 – Internet používá více než 3,2 miliardy uživatelů.[6]

Dnes tedy můžeme s jistotou hovořit o tom, že slovo internet slyšel již téměř každý. Většina z nás ale tento název nepoužívá pro pojmenování celosvětové počítačové sítě. Mnohem častěji máme na mysli obsah a informace, jenž jsou nám díky této síti dostupné. Za tímto ale ve skutečnosti stojí jedna ze služeb, která nám je díky síti internet poskytována. Jedná se o nejrozšířenější z nich - World Wide Web.

## 1.2 Hypertext a služba World Wide Web (WWW)

Hypertext je označení pro možnost propojit a uspořádat velké množství informací jako například texty, obrázky atd. Jeho koncept vznikl již ve 40. letech 19. století. Konkrétně ve chvíli, kdy Vannevar Bush napsal článek, ve kterém se zmiňuje o systému pro „prohlížení a pořizování poznámek z rozsáhlých textů a grafiky“. V následujících několika dekádách se tento koncept dále rozvíjel, avšak jeho největší boom přišel ruku v ruce s počítači.[7]

Hlavní zlom přišel na konci 80. let ve Švýcarsku, konkrétně v CERNu – Evropském výzkumném centru částicové fyziky. Zde se zrodila myšlenka mít možnost po síti získat snadný přístup k dokumentům týkajících se výzkumu odkudkoli bez potřeby vědět, na kterém počítači v síti jsou tyto dokumenty uloženy. Toto dalo vzniknout hypertextové službě World Wide Web, za jejímž vytvořením stál od počátku Tim Berners-Lee. Ten je díky tomuto považován za otce webu, jak se této službě také zkráceně říká.[7] [8]

## 1.3 Současnost služby WWW

V současnosti je pro nás právě World Wide Web, zkráceně WWW nebo jen Web, velmi cenným zdrojem informací, které jsme díky němu schopni získat během okamžiku. Obsahuje totiž nespočet webových stránek s nejrůznějším obsahem a jejich počet stále roste. K prohlížení a zobrazení jejich obsahu nám slouží tzv. webové prohlížeče. Jedním z prvních, který se poměrně rychle rozšířil do celého světa, byl prohlížeč Mosaic, jenž byl světu představen v roce 1993. Za jeho vývojem stál Marc Anderssen se svými kolegy z NCSA (National Center for Supercomputing Applications). Právě z něj se později vyvinuly dva ve své době velmi známé prohlížeče – Internet Explorer a Netscape Navigator. Dnes však na trhu s převahou dominuje webový prohlížeč Google Chrome, který používá téměř 70% uživatelů. Další v pořadí je prohlížeč Firefox, jenž využívá 17,8% uživatelů. Jako třetí nejrozšířenější je aktuálně Internet Explorer, následovaný

prohlížeči Safari a Opera.[10]

Za několik posledních let můžeme pozorovat razantní zlepšení právě zmíněných webových prohlížečů, které jsou rychlejší a nabízejí stále nové a nové možnosti. Ruku v ruce s tím se také průběžně zkvalitňují způsoby psaní kódů a stylů webových stránek. Zároveň již také nejsme tak jako v minulosti vázáni prohlížet si tyto stránky pouze z desktopových počítačů a notebooků, ale můžeme si je doslova „vzít s sebou“ na cesty – na chytrých telefonech, tabletech a mnoha dalších zařízeních. Díky tomu se můžeme k informacím uvedených na webových stránkách dostat téměř kdekoli a z jakéhokoli zařízení. Stačí, aby obsahovalo webový prohlížeč a bylo jej možné připojit k internetu.[9]

V následující kapitole se zaměřím na některé konkrétní webové stránky, které si kladou za cíl pomoci handicapovaným lidem.

## 2 TECHNOLOGIE TVORBY WEBU

Při vytváření webových stránek mohou vývojáři při jejich činnosti v současnosti využít nespočet technologií, programovacích jazyků a nástrojů. Tato kapitola obsahuje přehled některých z nich se zaměřením na důležité oblasti, jež byly využity při tvorbě webové aplikace, jež vznikla v rámci této diplomové práce.

### 2.1 HTML + CSS + JavaScript

Spojení těchto tří technologií najdeme v současnosti na drtivé většině webových aplikací. Každá z nich má na stránkách na starost určitý úkol. HTML se stará především o správu obsahu stránek, CSS má na starost grafickou úpravu tohoto obsahu a JavaScript slouží k vytváření dynamického chování jednotlivých prvků na klientské části webu, tedy u uživatele.[2]

#### 2.1.1 HTML

HTML (Hypertext Markup Language) je jednoduchý značkovací jazyk, jehož základy se lze poměrně snadno naučit a používá se jako základ pro definování významu obsahu webových stránek. Za jeho vytvořením stál na počátku 90. let minulého století Tim Berners-Lee.

Jazyk HTML sloužil a dodnes slouží pro psaní hypertextových dokumentů, což jsou textové dokumenty, které obsahují tzv. hypertextové odkazy. Pomocí nich je možné přenést se na jinou část tohoto dokumentu, případně na úplně nový dokument. Kromě prostého textu mohou dokumenty obsahovat také multimediální obsah jako například obrázky, zvuky či videa.

První definici jazyka HTML vytvořil v roce 1991 již zmíněný Tim Berners-Lee v rámci projektu WWW. Tato verze označována jako HTML 0.9 obsahovala pouze základní možnost členění textu do několika logických úrovní, bylo možné použít několik druhů zvýraznění textu a přidat do textu hypertextové odkazy a obrázky. Samozřejmě nároky uživatelů se časem zvyšovaly, což dalo vzniknout následující verzi.

Verze HTML 2.0 byla vydána v roce 1995 a doplněna do ní byla grafika a práce s formuláři.

Další verze nesla označení HTML 3.2 a obsahovala novinky v oblasti tabulek a toku textu okolo obrázků. Byla to rovněž první verze standardu, který vznikl již pod konsortiem W3C<sup>1)</sup>.

Pokračováním byla v roce 1997 verze HTML 4.0, jež přinášela celou řadu nových

---

<sup>1)</sup>World Wide Web Consortium - <https://www.w3.org/>

možností především v oblasti používání skriptů, vkládání objektů, vložených dokumentů (obrázky, videa, zvuky, matematické výrazy apod.).

Poslední verzí aktuálně používanou je HTML 5. Na jejím vývoji spolupracovala organizace W3C společně s WHATWG<sup>2)</sup>. Tak jako předešlé verze je zachována kompatibilita s předcházejícími verzemi standardu HTML. Byly doplněny elementy pro popis obsahu (např. `article`, `section`), dále byla zavedena podpora přehrávání zvukových souborů a videosouborů přímo v prohlížečích, bez nutnosti instalace jakéhokoli doplňku a mnoho dalších. Většina z nich je v současnosti podporována běžně používanými webovými prohlížeči, což značně ulehčuje práci vývojářů.[4] [9]

Momentálně jsou již vyvíjeny verze 5.1 a 5.2, které by podle plánu měly spatřit světlo světa na konci roku 2016, respektive v roce 2018<sup>3)</sup>.

Základním stavebním kamenem jazyka HTML jsou tzv. tagy, jinak řečeno značky. Ty jsou doplněny atributy, jenž vždy určují vlastnosti daného tagu.

### 2.1.2 CSS

Cascading Style Sheets (CSS) neboli kaskádové styly dokumentů vznikly kvůli potřebě rozšíření možností jazyka HTML. V začátcích uživatelům stačilo zobrazení potřebného obsahu na webových stránkách, kde nebyl kladen důraz na přesnou definici výsledného vzhledu. Netrvalo však dlouho, web se rozšířil i do komerční sféry a díky tomu se začaly množit požadavky na možnost lepší kontroly výsledného vzhledu dokumentů. V první fázi byly do tagů přidány nové atributy jako například `font` pro určení vzhledu písma, nebo `align` pro určení zarovnání. To se však ukázalo jako poměrně nevhodné řešení, především kvůli dvěma hlavním důvodům – namísto struktury dokumentu se do jejich kódu začaly zapisovat spíše definice grafického vzhledu a navíc bylo potřeba tyto atributy nadefinovat na každém tagu zvlášť. Docházelo tak k velké duplicitě kódu, jenž v případě potřeby upravit některý atribut napříč celým dokumentem. To vyústilo v nutnost tuto úpravu provést na velkém množství tagů.

Za tímto účelem vznikly kaskádové styly (CSS), které dovolují oddělit definici vzhledu od definice obsahu stránek. Navíc je díky nim možné určit například druh písma, způsob zarovnání, barvu a další vlastnosti vhodné pro určitý tag. Tento styl se poté použije v rámci celého dokumentu, případně v rámci celého webu kdekoliv, kde se tento tag vyskytne.

---

<sup>2)</sup>Web Hypertext Application Technology Working Group - <https://whatwg.org/>

<sup>3)</sup><https://dev.w3.org/html5/decision-policy/html5-2014-plan.html>

### 2.1.3 JavaScript

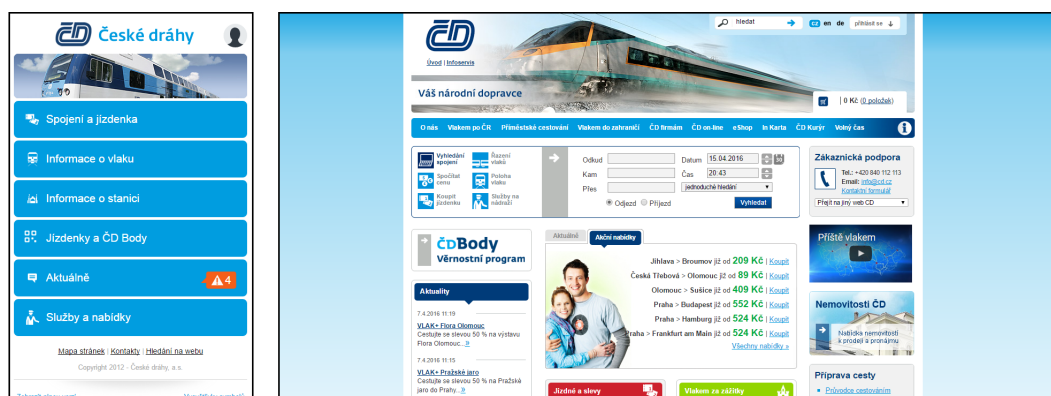
JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, který se v současnosti při tvorbě webových portálů těší stále větší oblibě. Vytvořil jej v roce 1995 Brendan Eich, tehdy pod křídly společnosti Netscape. Javascript určitým způsobem doplňuje zmíněný HTML pro zápis obsahu stránek a CCS pro definici designu, vzhledu těchto stránek. Primárně byl určen pro spuštění na straně prohlížeče. Odezva na akce uživatele je tak téměř okamžitá, na rozdíl od podstatně pomalejší komunikace s webovým serverem.[4]

JavaScript do statického HTML přidává prvky interaktivity – validace správnosti vyplnění formulářů, oživení webu v podobě různých měnících se obrázků, pohybujících se elementů stránky, náhledů obrázků, videí apod.

Nevýhodou Javascriptu je fakt, že je závislý na prohlížeči – v různých prohlížečích se skript může chovat odlišně. Rovněž si jeho použití může uživatel v prohlížeči vypnout a tak veškeré skripty na stránce odříznout.

### 2.1.4 Responzivní design

Nejedná se sice o konkrétní technologii jako v případě předešlých tří, ale považuji za důležité zde tento termín zmínit a osvětlit jej. V minulosti, kdy si webové stránky zobrazovali uživatelé pouze na počítači či notebooku, plně postačovalo navrhnout web tak, aby byl dobře čitelný na těchto zařízeních. S příchodem internetu na mobilní zařízení se však objevil nový problém, nebo chcete-li výzva – zobrazení stránek na displejích, které se svou velikostí ani zdaleka neblíží velikosti displeje běžně používaných notebooků či velikosti monitorů desktopových počítačů. V počátcích se objevila tendence ponechat stávající weby tak, jak jsou a vytvořit novou verzi speciálně pro mobilní zařízení. Jako příklad mohu uvést stránky Českých drah. Klasické stránky jsou dostupné z URL <http://www.cd.cz> a mobilní verzi nalezneme na <http://m.cd.cz>.



Obr. 2.1 Mobilní (vlevo) a klasická verze webových stránek Českých drah (vpravo)

Nevýhodou tohoto řešení je potřeba udržovat v podstatě dva oddělené weby, výhodou naopak rychlost načítání mobilní verze. A to především tehdy, pokud vezmeme v potaz možnou nižší rychlost mobilního připojení k Internetu. S tím je spojena i menší datová náročnost, která potěší uživatele, jejichž mobilní připojení je omezeno pomocí datových limitů.

Další možností je vytvoření mobilní aplikace, kterou si uživatel stáhne do svého telefonu či tabletu a pomocí které se může dostat k informacím, jež mu chcete dát k dispozici. Nevýhodou jsou náklady spojené s vývojem takovéto aplikace, rozmanitost operačních systémů a hardwarových konfigurací mobilních zařízení, kterým je třeba danou aplikaci přizpůsobovat. V neposlední řadě také potřeba si aplikaci nainstalovat a případně i průběžně aktualizovat.

Třetí možností, která je aktuálně stále více populární, je tzv. responzivní webdesign. Jedná se v podstatě o jeden web, který se však vykresluje na různých zařízeních odlišným způsobem. Přesněji řečeno záleží na velikosti zobrazované plochy daného zařízení. Podle ní se poté zobrazuje obsah stránek. V praxi volí vývojáři několik úrovní responzivnosti. Například desktop s velkým displejem, notebook, tablet a mobil. Tyto úrovně si může vývojář určit téměř libovolně.

Výhodou responzivního designu je potřeba implementace pouze jednoho webu, kterému je ale potřeba věnovat velkou pozornost především s ohledem na design jednotlivých úrovní responzivnosti. Obvykle se využívá tzv. Mobile first přístup, který spočívá v návrhu vzhledu stránek od nejmenšího rozlišení a pokračuje k rozlišením větším. Nevýhodou může být pro některé uživatele skutečnost, že pokud zmenším okno webového prohlížeče, změní se mi i rozložení stránky, které je závislé na šířce zobrazovací plochy.[11] [12]



Obr. 2.2 Příklad responzivního webu pondios.cz

Na obrázku vidíme příklad responzivního webu, který je dostupný z jedné URL adresy, avšak uživatelům je prezentován jinak na mobilním telefonu, tabletu i na notebooku.

## 2.2 PHP

PHP (Hypertext Preprocessor) je multiplatformní skriptovací jazyk, který je používán především při tvorbě dynamických webových stránek. První jeho verze spatřila světlo světa v červnu roku 1995 a od té doby bylo vydáno několik jeho verzí, přičemž poslední z nich a zároveň nejaktuálnější nese označení 7.1.

Princip jazyka PHP spočívá v tom, že zapsaný kód je vykonán ještě na straně serveru. Po jeho provedení je na klientskou část odeslána výsledná stránka již v čistém HTML. Klient prohlížející si webové stránky se tak ke zdrojovým kódům v PHP nedostane a vidí pouze výsledek jejich zpracování. Toto je hlavní rozdíl oproti jazyku JavaScript, jenž se provádí až na straně klienta a uživatel k němu má přístup.[20]

Díky velmi rozsáhlé vývojářské komunitě jazyka, a také vzhledem k době, po kterou je jazyk vyvíjen a stále hojně používán, vznikla celá řada frameworků. Jako příklad je možné uvést Zend Framework<sup>4)</sup>, CodeIgniter<sup>5)</sup> nebo český Nette Framework<sup>6)</sup>.

## 2.3 ASP.NET

Technologie ASP.NET je součástí .NET Frameworku od společnosti Microsoft a je nástupcem jazyka ASP. Oproti němu je však ASP.NET o poznání rychlejší, a to především díky vytvoření a použití kompilovaného kódu. Zároveň je díky tomu řada chyb odhalena již při vývoji v době kompilace a ne až při běhu webové aplikace.

První verze ASP.NET 1.0 byla vydána v lednu 2002. V současnosti je nejnovější verze označena jako 4.6.1, jenž oficiálně spatřila světlo světa v listopadu 2015. Následovnicem měla být verze 5.0, avšak společnost Microsoft se rozhodla celou tuto technologii převést pod licenci open-source. Vzhledem k tomuto kroku a také velmi zásadním úpravám bylo rozhodnuto o změně názvu na ASP.NET Core 1.0.[13]

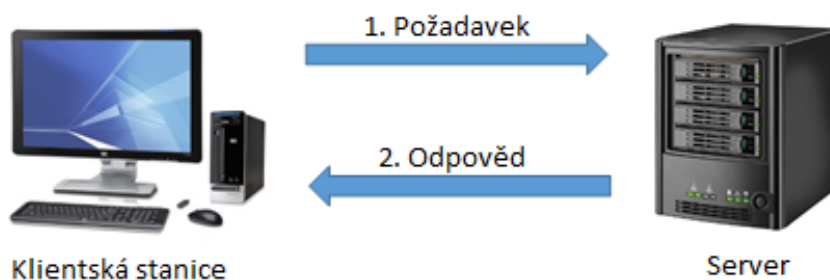
Mezi výhody použití technologie ASP.NET můžeme zařadit schopnost simulace stavového chování. Klasické použití bezstavového webového HTTP protokolu toto totiž neumožňuje. Typickým příkladem použití ASP.NET je komunikace klient-server. Na klientské stanici je vytvořen požadavek, který je odeslán na server. Většinou se jedná o reakci na nějakou uživatelskou akci – stisk tlačítka, odeslání formuláře, požadavek na zobrazení stránky apod. Server tento požadavek následně přijme. Pokud se jedná

<sup>4)</sup><http://framework.zend.com>

<sup>5)</sup><http://www.codeigniter.com>

<sup>6)</sup><https://nette.org>

o korektní požadavek a server je schopný jej zpracovat, provede to a následně odešle zpět na klienta odpověď. Ve finální části dojde na klientské stanici ke zpracování odpovědi ze serveru. Typicky je výsledek zobrazen uživateli ve formě potvrzení korektního provedení požadované akce, zobrazení požadované stránky aj.



Obr. 2.3 Schéma komunikace klient-server

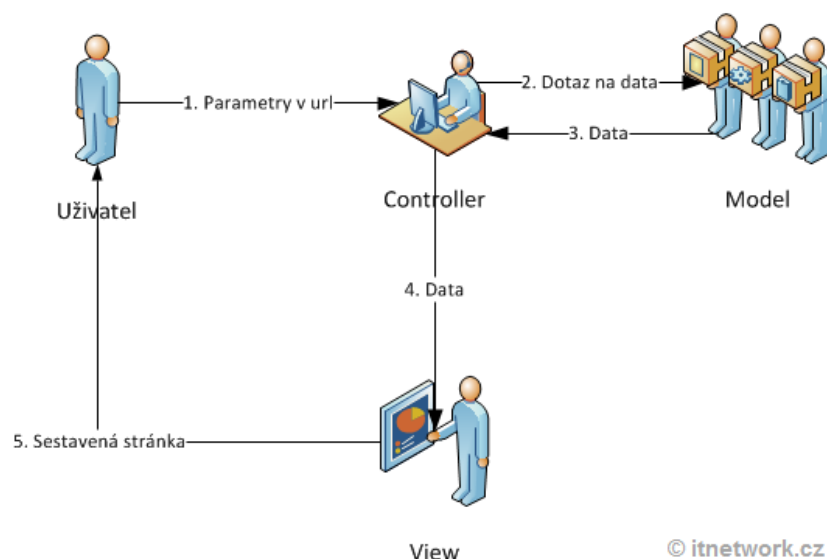
## 2.4 ASP.NET MVC

Framework ASP.NET MVC je jak již název napovídá spojením technologie ASP.NET a architektury MVC, což umožňuje vyvíjet webové aplikace s podporou Frameworku ASP.NET za využití návrhového vzoru MVC (Model View Controller). Fakticky se jedná o rozdělení aplikace do tří částí, které spolu navzájem spolupracují a každá z nich má svůj specifický úkol. [14]

- **Model** (datová vrstva) obsahuje business logiku a reprezentaci dat. Prakticky se stará o přijetí parametrů zvenku a vydání požadovaných dat. Velmi často bývá doplněn tzv. ORM (Objektově-Relační Mapování), která zpřístupňuje data z datového úložiště do formy objektů, se kterými poté aplikace pracuje.
- **View** (prezentační vrstva) se stará o zobrazení výstupu uživateli. Zajišťuje grafickou část aplikace a určuje, jakým způsobem bude výsledná stránka s informacemi vypadat.
- **Controller** (aplikační vrstva) je jakýmsi prostředníkem, který zajišťuje komunikaci mezi Uživatelem, Modelem a View. Zajišťuje požadovanou funkcionalitu aplikace, přebírá požadavky od uživatele, stará se o práci s daty atd.

Typický příklad fungování MVC architektury je následující:

1. Uživatel zadává URL adresu webu, respektive konkrétní požadované stránky.
2. Controller, který má tuto stránku „na starosti“, si požadavek převezme a požádá Model o potřebná data k této stránce.
3. Po dodání dat Modelem vyrenderuje Controller patřičné View a předá mu obdržená data.
4. View přijme data a vloží je do připravené šablony stránky.
5. Uživateli je zobrazena kompletní požadovaná webová stránka.



Obr. 2.4 Schéma funkce architektury MVC[14]

## 2.5 Databáze

Pod pojmem databáze je možné si představit datové úložiště, kam jsou ukládána strukturované a organizované údaje, tedy jakousi kolekci informací, se kterými je možné pracovat. V širším slova smyslu se jedná o celý databázový systém, který se skládá z datového úložiště a tzv. SŘBD (Systém řízení báze dat), jenž poskytuje právě prostředky pro práci s uloženými daty – jejich vkládání, odstraňování, upravování, vyhledávání, ale také například různé statistiky nad uloženými daty apod. Databáze jsou ve velké míře využívány při tvorbě webových stránek pro uchovávání potřebných informací.

### 2.5.1 Microsoft SQL Server

Microsoft SQL Server, zkráceně MS SQL, je relační databázový systém od společnosti Microsoft. První verze s označením Microsoft SQL Server 1.0 se objevila na trhu v roce 1989 a postupně vznikaly další a další verze. Zásadní zlom přišel v roce 2000, kdy byla vydána verze 8.0 a všechny následující verze jsou s ní zpětně kompatibilní. Jinak řečeno databáze pro verzi 8.0 poběží korektně i na verzi 9.0 a dalších. Zatím poslední verze 12.0 byla vydána v roce 2014 a nese označení Microsoft SQL Server 2014.[15] [16]

Jakožto relační databázový systém může MS SQL obsahovat relační databáze, což jsou zjednodušeně řečeno určitá úložiště pro data. Zde jsou některé důvody, kvůli kterým se vlastně databáze používají.

- Databáze poskytuje rychlejší přístup k datům nežli soubory.
- Databáze dovoluje a kontroluje svými mechanismy paralelní přístup k datům.
- Databáze obsahuje systém uživatelských práv.
- Databáze umožňuje pomocí tzv. dotazů poměrně jednoduše získat určitou množinu dat, která vyhovuje požadovaným kritériím.

Takováto databáze může obsahovat libovolné množství tabulek, které slouží k ukládání dat, přičemž každá tabulka obsahuje libovolné množství záznamů (řádků). Každý z nich musí mít svůj jedinečný identifikátor, tzv. primární klíč. Tímto termínem je označován jeden sloupec či kombinace více sloupců, kterým/i je jednoznačně určen vždy jeden konkrétní řádek v dané tabulce.

Dalším specifikem pro relační databáze je možnost navzájem propojit jednotlivé tabulky. K tomuto propojení slouží tzv. cizí klíče. Jedná se opět o jeden či více sloupců, jenž umožňují identifikovat, které řádky z různých tabulek spolu navzájem souvisí.[17]

### 2.5.2 MySQL

MySQL je databázový systém, jenž byl vyvinut společností MySQL AB. Tu založili ve Švédsku v roce 1995 pánové David Axmark, Michael Widenius a Allan Larsson. Později v roce 2008 od ní odkoupila práva na MySQL společnost Sun Microsystems, kterou o rok později koupila společnost Oracle Corporation. Ta se dnes také stará o její vývoj.

Jelikož se jedná o multiplatformní databázový systém, který je šířen pod bezplatnou licencí GPL<sup>7)</sup>, je mezi vývojáři poměrně oblíben. A to především ve spojení s programovacím jazykem PHP.

<sup>7)</sup><http://www.gnu.org/licenses/gpl-3.0.htm>

## 2.6 Objektově relační mapování (ORM)

V jakékoli webové aplikaci, která potřebuje ukládat nějaká svá data do databáze, souborového systému, či jiného datového úložiště, je třeba vyřešit vrstvu, která pro ni bude poskytovat tato uložená data. Někteří vývojáři si tuto mezivrstvu řeší sami, většina se ale přiklání k použití tzv. ORM systémů. Ty se starají o převod dat mezi jejich úložištěm (relační databází) a objektově orientovaným programovacím jazykem. Jinak řečeno dokáže určitým způsobem provázat objekty, jenž jsou používány v programu, s daty, které jsou uloženy v neobjektově relační databázi.

### 2.6.1 Entity Framework

Společnost Microsoft pro tuto činnost nabízí ORM nazvaný Entity Framework, jenž je založen na ADO.NET <sup>8)</sup>. Tento Entity Framework, zkráceně EF, ulehčuje vývojářům práci s databází. Ti díky němu nemusí pracovat přímo s databází, ale vytvářejí dotazy na třídy modelu v aplikaci a EF za ně řeší jednoduché i složité dotazy do databáze pro čtení a zápis požadovaných dat. Postupem času vznikly tři možnosti jak za použití EF vytvořit modely objektů.

- **Database first** je přístup, kdy máme existující databázi, na základě které nám EF vytvoří do modelu odpovídající třídy a kontext.
- **Model first** je označením pro postup, kdy se pomocí modelování v EF vytvoří předloha pro databázi, kterou následně EF vytvoří.
- **Code first** je nejmladší a aktuálně nejpoužívanější přístup, kde je základem vytvoření C# tříd, podle kterých EF vygeneruje databázové tabulky a potřebný kontext. V rámci tohoto přístupu nabízí EF tzv. Migrations (migrace), kde je schopen reagovat na změny modelu a podle toho v průběhu vývoje i automaticky upravovat databázové schéma.

### 2.6.2 NHibernate

Rovněž se jedná o vyspělý ORM od společnosti Microsoft, avšak zásadním rozdílem je, že defaultně nepodporuje přístupy Database first, Model first ani Code first. Ty je možné použít pouze s využitím dalších komerčních nástrojů. Obecně se dá říct, že NHibernate je vhodné použít u rozsáhlých projektů, kde jsou kladeny vysoké požadavky na škálovatelnost a výkon ORM, kde lehce převyšuje Entity Framework. V současnosti se však snižuje počet vývojářů, pracujících na této technologii a zároveň tak postupně zpomaluje tempo implementace funkcí z nových verzí .NET Frameworku. Jeho budoucnost je tedy nejistá.[21]

<sup>8)</sup><https://msdn.microsoft.com/en-us/library/h43ks021.aspx>

## 2.7 Verzovací nástroje

V minulosti se ani mezi profesionálními vývojáři žádné verzovací nástroje nepoužívaly, dnes je však situace jiná a s verzovacími nástroji se můžeme setkat u drtivé většiny projektů. A nemusí se jednat pouze o tvorbu webových aplikací, ale obecně jakýchkoli aplikací bez rozdílu zaměření či použitého programovacího jazyka.

Jednou z možností je průběžně si ukládat vždy celý obsah adresáře se zdrojovými kódy bokem a v případě potřeby se k němu vrátit. Toto řešení je ale náročné jak na správu, která je v režii vývojáře, tak i na velikost paměti, jež tyto průběžné zálohy zabírají. Čas, který zabere nalezení správného záložního adresáře, je také určitě nezanedbatelná záležitost.

Druhou určitě vhodnější možností je využití verzovacího nástroje, jehož hlavní úloha je ve správě zdrojových kódů. Konkrétně v zaznamenávání provedených úprav tak, abychom se k nim byli schopni zpětně dostat a přesně zjistit jaké změny byly kdy v minulosti provedeny. Pokud na projektu spolupracuje více vývojářů, měl by poskytovat informaci i o tom, kdo z nich danou úpravu provedl.[22]

### 2.7.1 Subversion (SVN)

SVN neboli Subversion je nástupcem nástroje CVS (Concurrent Versions System)<sup>9)</sup>. Jedná se o open-source systém založený na centrálním repozitáři, do kterého jsou změny zdrojových kódů zaznamenávány, a ze kterého se pořizují kopie. Pokud k tomu máte práva, můžete po stažení této kopie provádět úpravu kódů a tu poté odeslat do centrálního repozitáře. V něm je tedy vždy uložena poslední, nejaktuálnější verze zdrojových kódů. Samozřejmě je ale možné se dostat i ke starším takto uloženým verzím kódů v repozitáři. Umožněn je ale také v případě potřeby výběr některé ze starších verzí a jej následné stažení. Není nutné stahovat vždy jen nejnovější verzi kódů.[22]

### 2.7.2 Team Foundation Server (TFS)

Team Foundation Server není jen klasickým verzovacím nástrojem pro správu zdrojových kódů, ale ukrývá v sobě i mnoho možností pro správu dalších částí životního cyklu aplikací. Typickým příkladem je řešení požadavků, úkolů, chyb a realizovaných změn. Tedy dovoluje uchovávat informaci o tom, za jakým účelem byla daná změna v kódu provedena. Případně je možné ještě doplnění informaci o provedeném testování dané změny. Další záležitostí využívanou především u větších projektů, je možnost spojení s buildovacími systémy. Nespornou výhodou je také možnost využití TFS nejenom přes klasického .NET klienta pro Visual Studio, ale i pomocí webového klienta

---

<sup>9)</sup><http://www.nongnu.org/cvs>

v libovolném prohlížeči.[23]

## 2.8 Mapové podklady

Poskytovatelů kvalitních mapových podkladů, jenž je možné použít na webových stránkách, existuje celá řada. Níže jsou uvedeny dva pro tento projekt nejvhodnější adepti – Google Maps a Mapy.cz. Mimo tyto mohu zmínit například ještě projekt OpenStreetMap<sup>10)</sup>, který je tvořen otevřenými daty a může do něj přispívat prakticky kdokoli. Dále jsou ve světě rozšířeny i mapové podklady od společnosti Yahoo, Bing či HERE.

### 2.8.1 Google Maps

Google Maps jsou celosvětově nejznámější a nejpoužívanější mapové podklady využívané webovými vývojáři [19]. Google Maps nabízejí několik verzí API (Application Programming Interface), v závislosti na oblasti použití. Jako příklad mohu uvést API pro aplikace běžící na zařízeních s operačním systémem Android<sup>11)</sup> nebo pro zařízení s operačním systémem iOS<sup>12)</sup>. Pro webové aplikace je potom z poskytovaných API nevhodnějším kandidátem Google Maps JavaScript API<sup>13)</sup>. To je jak již název napovídá založeno na programovacím jazyce JavaScript (viz kapitola 2.1.3). Vzhledem k obrovskému počtu vývojářů, kteří toto API využívají a velmi kvalitní a obsáhlé dokumentaci na stránkách Google Developers, je možné dohledat nespočet návodů a řešených příkladů včetně částí zdrojových kódů.

### 2.8.2 Mapy.cz

Jedná se mapové podklady, které má pod svými křídly společnost Seznam.cz. Dříve byly určeny pouze pro komerční využití, ale aktuálně je jejich použití zdarma i pro komerční účely. Navíc je zpřístupněno Mapy API, jež poskytuje celou řadu funkcionalit (např. vyhledávání, plánování, měření, apod.), která je průběžně doplňována o nové. Rovněž mapové podklady jsou stále zkvalitňovány. Určitou nevýhodou je zaměření víceméně pouze na Českou republiku a okolí, což značně omezuje cílovou skupinu vývojářů. Nemožnost najít informace potřebné k vývoji určité funkcionality tak díky tomu může být určitým problémem. Avšak věřím, že do budoucna bude Mapy API vývojáři stále více používáno a tím dojde k rozšíření množiny dostupných informací ohledně nejrůznějších otázek spojených s používáním mapových podkladů a jejich řešení.

---

<sup>10)</sup><https://www.openstreetmap.org>

<sup>11)</sup><https://developers.google.com/maps/documentation/android-api/>

<sup>12)</sup><https://developers.google.com/maps/documentation/ios-sdk/>

<sup>13)</sup><https://developers.google.com/maps/documentation/javascript/>

### 3 EXISTUJÍCÍ WEBY PRO POMOC HANDICAPOVANÝM

Jelikož se jádro mé diplomové práce zabývá pomocí handicapovaným, v této části mé práce zmapuji již existující projekty s podobným zaměřením. Našel jsem jich opravdu celou řadu. Od jednoduchých webů se základními důležitými údaji, přes stránky různých spolků a sdružení, až po rozsáhlé webové portály obsahující nespočetné množství informací a odkazů. Pro příklad mohu uvést následující.

Komunitní weby:

- Vozejkov (<http://www.vozejkov.cz>)
- Vozka (<http://vozka.org>)
- Česká asociace paraplegiků (<http://www.czepa.cz>)
- Centrum Paraple (<http://www.paraple.cz>)
- Vozičkář (<http://vozickar.com>)
- Žiju s handicapem (<http://www.zijushandicapem.cz>)

Weby s důležitými informacemi pro handicapované:

- Ministerstvo práce a sociálních věcí – Zdravotní postižení (<http://www.mpsv.cz/cs/8>)
- Národní rada osob se zdravotním postižením ČR (<http://www.nrzp.cz>)
- Katalog sociálních služeb ve Zlínském kraji (<http://socialnisluzbyzk.cz>)

Spolky a sdružení:

- Svaz tělesně postižených Zlín (<http://www.stp-zlin.cz>)
- Alfa Handicap Přerov (<http://www.alfaprerov.cz>)
- Vozičkáři Ostrava (<http://www.vozickari-ostrava.cz>)

Tento uvedený seznam obsahuje pouze několik příkladů aktuálně existujících stránek v ČR. Webů s tematikou pomoci handicapovaným lidem však existuje u nás i v zahraničí nespočet.

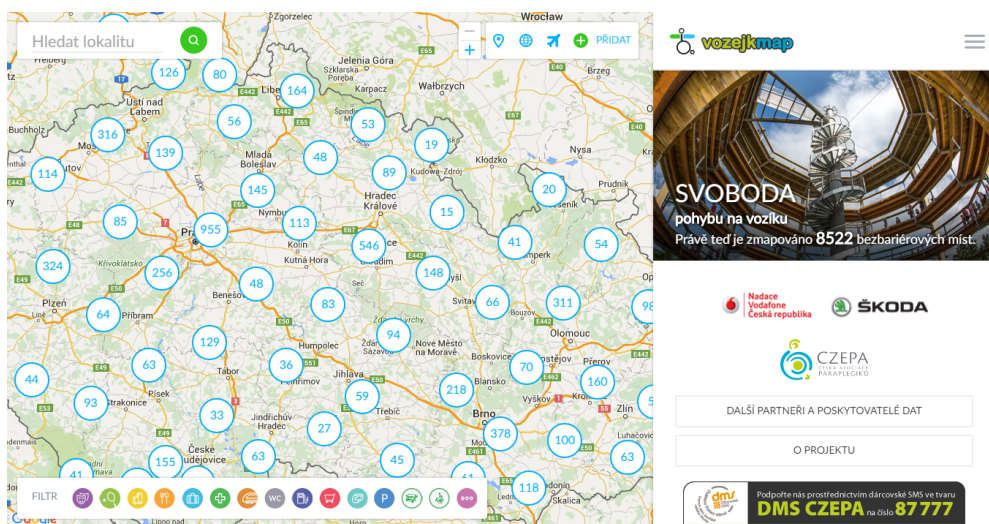
### 3.1 Weby s mapováním přístupnosti míst

Těžiště mého projektu však leží především v potřebě předat informaci o zmapovaných důležitých veřejných místech z hlediska jejich přístupnosti pro handicapované. Tedy především pro osoby se sníženou pohyblivostí a pro osoby upoutané na invalidní vozík. Jako základní stavební kámen tudíž bude sloužit mapa, na které budou získané informace prezentovány. Tato myšlenka mě dovedla až k následujícím, do jisté míry možná konkurenčním řešením.

#### 3.1.1 Vozejkmap.cz

Jedná se o projekt, který se skládá z webové aplikace a aplikace pro mobilní zařízení s operačními systémy Android a iOS, přičemž data jsou mezi těmito aplikacemi sdílena. Podle informací uvedených v části stránek „O projektu“ data do systému zadávají a ověřují samotní vozíčkáři. Zároveň je databáze těchto bezbariérových míst po dodržení licence Creative Commons CC BY-NC-SA 4.0<sup>1)</sup> volně přístupná k nekomerčnímu použití. Jako mapové podklady jsou použity známé Google Maps.

Po zobrazení stránky zabírá hlavní část prostoru mapa, na níž jsou umístěny značky jednotlivých míst. Lze mezi nimi vyhledávat, filtrovat a zobrazovat si detaily k požadovaným místům. Jak jsem již uvedl v předchozím odstavci, je zde k dispozici i možnost přidání nového místa, které lze po vyplnění všech povinných polí odeslat ke schválení. Ověření obvykle probíhá do 24 hodin a poté je toto místo zobrazeno na mapě pro všechny návštěvníky stránek.



Obr. 3.1 Webová stránka Vozejkmap.cz

Celkově působí řešení velmi profesionálním dojmem, což je s velkou pravděpodobností způsobeno tím, že se na něm podílejí CZEPA (Česká asociace paraplegiků), spo-

<sup>1)</sup><http://creativecommons.org/licenses/by-nc-sa/4.0>

lečně se Škoda Auto a.s., Nadací Vodafone ČR a s mnoha dalšími partnery, jenž jsou uvedeny na stránkách.

### 3.1.2 Mapybezbarier.cz

Stránka Mapybezbarier.cz vzniká pod záštitou Nadace Charty 77 – Konta Bariéry (<http://www.kontobariery.cz>), s podporou Ministerstva pro místní rozvoj a je financována z prostředků Evropské unie, Evropského fondu pro regionální rozvoj. Z parterů projektu bych zmínil například Seznam.cz, Forpsi.cz, Asociaci krajů České republiky a Svaz měst a obcí České republiky. Webové stránky jsou založeny na mapových podkladech Google Maps.

Co se týká samotného obsahu, opět se jedná o podobný koncept jako v případě Vozejkmapp.cz, kde hlavní část prostoru zabírá mapa s jednotlivými místy. Je zde možnost filtrovat podle přístupnosti a typu objektu.

Na stránkách nalezne také možnost registrace, po níž může uživatel vytvořit tzv. „Nové mapování“ – zadat nové místo včetně nastavení přístupnosti, typu objektu, adresy apod.



Obr. 3.2 Webová stránka Mapybezbarier.cz

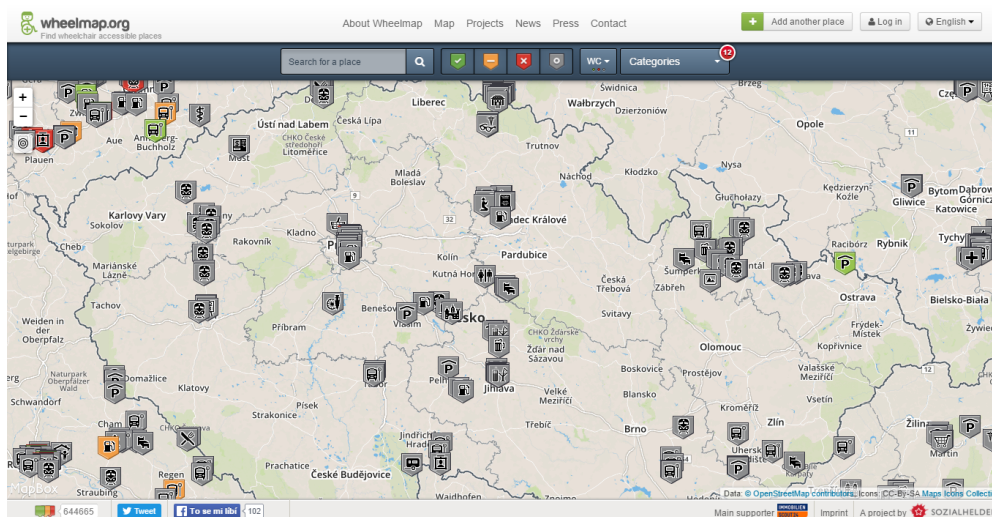
### 3.1.3 Wheelmap.org

Projekt Wheelmap.org je spojením webové a mobilní aplikace, podobně jako tuzemský Vozejkmapp.cz. Na rozdíl od něj se však nevztahuje výhradně na Českou republiku, nýbrž je zaměřen celosvětově. Jedním z hlavních tvůrců je Raul Krauthausen z Německa, který je sám upoután na invalidní vozík<sup>2)</sup>. Opět jako v obou předchozích projektech mohou přístupnost jednotlivých míst definovat a přidávat uživatelé, kteří se na stránkách

<sup>2)</sup><http://www.gisportal1.cz/2011/08/state-of-the-map-2011-shrnuti-a-plany-do-budoucnosti-den-3/>

zaregistrují. Díky tomu aktuálně můžeme v mapě najít téměř 650 tis. míst po celém světě, přičemž velká většina z nich je vozíčkářům přístupná<sup>3)</sup>.

Coby zdroj mapových podkladů jsou použity data z OpenStreetMap<sup>4)</sup>, což je, jak již název napovídá, otevřený komunitní mapový systém, který je možné šířit pod licenci ODbL<sup>5)</sup> respektive pod licencí CC BY-SA2.0<sup>6)</sup>, jenž se vztahuje ke kartografickému obsahu map, pokud je využíván.



Obr. 3.3 Webová stránka Wheelmap.org

Poměrně výraznou nevýhodou pro použití v České republice je fakt, že prozatím chybí překlad stránek do češtiny. I když valná většina uživatelů by určitě sáhla po slovenské verzi stránek – ta již aktuálně podporovaná je. Jelikož však mobilní aplikace pro Android již v české lokalizaci funguje, předpokládám, že nebude trvat dlouho a i tyto webové stránky se dočkají svého překladu.

<sup>3)</sup><https://community.wheelmap.org/en/about/press/>

<sup>4)</sup><http://www.openstreetmap.org/about>

<sup>5)</sup><http://opendatacommons.org/licenses/odbl/>

<sup>6)</sup><http://creativecommons.org/licenses/by-sa/2.0/deed.cs>

### 3.2 Shrnutí existujících řešení

Ve srovnání se zaměřím pouze na poslední tři zmíněné příklady existujících webových stránek zabývajících se potřebou zobrazit místa z hlediska jejich přístupnosti tělesně handicapovaným lidem na mapě. Tímto směrem se chci ubírat i v případě mé diplomové práce.

Přestože jsou stránky do jisté míry odlišné, můžeme si u nich všimnout následujících společných rysů:

- Kvalitní mapové podklady
- Otevřená databáze míst, do které může přispívat i široká veřejnost
- Možnost filtrování mezi místy
- Možnost zobrazení detailu k požadovanému místu
- Za projektem a vývojem stojí rozsáhlý tým lidí

Pokud bych měl určit, která z variant se mi zdá jako nejpovedenější, nejpoužitelnější, a kterou bych v případě potřeby využíval, jedná se o první zmíněnou aplikaci [Vozejkmap.cz](http://Vozejkmap.cz). Její výhody oproti ostatním řešením vidím v množství míst, která jsou v ní zachycena. Alespoň tedy na území České republiky. V celosvětovém měřítku má samozřejmě aplikace [Wheelmap.org](http://Wheelmap.org) navrch, jelikož obsahuje těchto míst statisíce a další neustále přibývají. Rovněž možnost vyhledávání a filtrování v aplikaci [Vozejkmap.cz](http://Vozejkmap.cz) mi přišla jako dobře použitelná a uživatelsky přívětivá.

Informace, které jsem získal díky rešerši existujících webových aplikací, jsem dále využil při návrhu a realizaci mé práce.

## II. PRAKTICKÁ ČÁST

## 4 SPECIFIKACE ZADÁNÍ

Zadání vzniklo z námětu Magistrátu města Zlína a jeho základním požadavkem bylo vytvoření interaktivního webu, ve kterém by byla uvedena důležitá místa ze Zlína a okolí. U každého z nich by poté měly být k dispozici informace o přístupnosti těchto míst pro vozíčkáře a tělesně handicapované spoluobčany, přičemž tato místa včetně jejich přístupností již má město zmapované. Zároveň ale tento v současnosti aktuální seznam míst má být možné rozšířit o nová místa, případně upravovat ta stávající. Tuto správu by měl zvládat běžný uživatel, konkrétně některý z pracovníků magistrátu, čemuž musí být administrátorská část systému uzpůsobena.

Mezi základní požadavky, jenž byly pro systém vymezeny, a kterými jsem se při vývoji řídil, patří několik níže uvedených.

### 1. Místa (objekty)

Nejdůležitějším stavebním prvkem aplikace má být místo neboli objekt. Ke každému z nich musí aplikace dovolovat uložení následující množiny potřebných informací:

- Název
- Poznámka
- Adresa
- Typy přístupnosti
- Kontakty
- Typy štítků

Aplikace rovněž musí dovolovat tyto informace v případě potřeby pro jednotlivá místa upravovat. Avšak kompletní správa míst musí být zcela v rukou magistrátu. Tedy jednotlivá místa nesmí zadávat anonymní uživatelé, ale pouze kompetentní pracovníci magistrátu, kteří budou mít do webové aplikace přístup.

### 2. Uživatelské role

Systém musí umožňovat rozlišení několika uživatelských rolí, podle kterých se řídí přístupová práva k jednotlivým částem aplikace. Zde se uživatelé rozdělí do dvou hlavních částí:

- **Nepřihlášený uživatel** - klasický návštěvník webových stránek
- **Přihlášený uživatel** - musí mít vytvořen uživatelský účet, kterému je nastavena určitá uživatelská role. Po přihlášení pod určitou roli má možnost

provádět administraci těch částí systému, ke kterým má jeho role povolený přístup. Specifika jednotlivých rolí jsou popsány dále v kapitole 6.3.1.

3. **Možnost rozšiřitelnosti míst** Seznam míst je sice v úvodní fázi projektu již definován v rámci zmapovaných míst v publikaci o bezbariérovosti - viz kapitola 5.1, avšak tento seznam není zcela jistě finální a v budoucnu bude docházet k jeho úpravám. I tuto skutečnost musí aplikace reflektovat a být na ni připravena. Musí být tedy dostupná možnost místa přidávat, editovat a mazat.
4. **Možnost správy typů přístupností** Při vytváření zadání projektu existovalo defaultních šest druhů přístupností jednotlivých míst:
  - Přístupný
  - Částečně přístupný
  - Nepřístupný
  - Přístupná toaleta - WC I
  - Přístupná toaleta - WC II
  - Vyhrazené parkovací místo

V budoucnu je ale možné, že přibudou nové kategorie. Případně bude potřeba změnit informace k některé z již existujících. Každé místo uvedené v aplikaci může mít přiřazeno libovolný počet z těchto kategorií. Všechny tyto požadavky musí být aplikace schopna pokrýt.

5. **Využití štítků** Jednotlivá místa je možné zařazovat do určitých kategorií podle jejich významů a zařazení, přičemž tyto kategorie jsou označovány jako štítky. V aplikaci má být podle nich možné místa filtrovat a zobrazit si tak jen určitou skupinu, či vybranou množinu skupin. Tyto štítky opět musí mít možnost administrátorské správy.

Pro splnění všech výše uvedených specifických bodů zadání nebylo možné použít žádný z již existujících webových projektů. Jasně definované uživatelské role, či uzavřenost celého systému vedly jasně k potřebě vytvořit novou webovou aplikaci na míru.

## 5 NÁVRH APLIKACE

Po úvodním nastudování již vytvořených řešení následoval návrh předpokladů a rysů, které by měla má práce obsahovat a splňovat. Základním stavebním kamenem byla specifikace požadavků na aplikaci a také následující publikace.

### 5.1 Publikace o bezbariérovosti

Jádrem mé práce je PDF dokument s názvem Vybrané instituce ve Zlíně, Fryštáku, Lípě, Lukově, Sazovicích a Želechovicích nad Dřevnicí z pohledu (bez)bariérovosti. Tento přehled spadá do projektu s názvem Komunitní plánování sociálních služeb ve Zlíně. Zpracoval jej Odbor sociálních věcí Magistrátu města Zlína. Za vznikem stála iniciativa pomoci tělesně handicapovaným, poskytnout jim přehled přístupností důležitých míst ve Zlíně a okolí. Konkrétně je dokument zaměřen na vzdělávací, informační, správní, zdravotní a poradenské instituce, dále kulturní a sportovní zařízení, restaurace a dopravu.[1]

Dokument obsahuje legendu typů přístupností, jejichž kategorie včetně ikon jsou uvedeny v tabulce níže.








Tab. 5.1 Legenda kategorií přístupností

Kategorie	Ikona kategorie
Objekt přístupný	
Objekt částečně přístupný	
Objekt nepřístupný	
Přístupná toaleta – WC I	
Přístupná toaleta – WC II	
Vyhrazené parkovací místo	

Struktura publikace je přehledná, jednotlivé místa jsou rozděleny podle polohy a následně podle typu zařazení, např. Veřejná správa, Kultura, Školství, Zdravotnická zařízení, Veřejná parkoviště apod. U každého místa jsou definovány tyto čtyři základní typy informací:

- Název
- Kontakty
  - Adresa místa
  - Odkaz na web
  - E-mail
  - Telefon
- Přístupnost objektu
- Poznámka

Příklad několika míst z uvedené publikace můžeme vidět na obrázku 5.1. Je z něj patrné, že každé místo může mít více kategorií přístupnosti.

Název	Kontakty	Přístupnost objektu	Poznámka
Základní škola Komenského II, Zlín, p. o.	Havlíčkovo nábřeží 2567 761 63 Zlín		
	www.zskom2.cz alex@zskomdva.zlinnedu.cz		
	+420 577 210 076		
Základní škola Zlín, Kvítková, p. o.	Kvítková 4338, 760 01 Zlín	 	Je možné přizpůsobit výuku pro dítě na vozičku v přízemí školy, možno parkovat v ulici.
	www.zskvitkova.cz zskvitkova@zskvitkova.cz		
	+420 577 213 405		
Základní škola praktická a Základní škola speciální Zlín	Středová 4694, 760 06 Zlín	  	2× parkovací místo u budovy.
	skola-spc.cz skola@skola-spc.cz		
	+420 577 142 747		
Církevní základní škola a mateřská škola ve Zlíně	Česká 4787, 760 05 Zlín		K dispozici rampa, díky které lze najet bočním vchodem, možno parkovat v ulici.
	www.czsztlin.cz czs@czsztlin.cz		
	+420 577 011 686		

Obr. 5.1 Publikace Vybrané instituce ve Zlíně, Fryštáku, Lípě, Lukově, Sazovicích a Želechovicích nad Dřevnicí z pohledu (bez)bariérovosti

## 5.2 Informace získané z existujících řešení

V úvodní části mé práce jsem prováděl rešerši existujících řešení webů, které mají za úkol pomáhat handicapovaným lidem. V kapitole 1.2 jsem uvedl několik bodů, jenž mají společně všechny tři zmiňované weby. Nyní z nich musím vybrat ty, které jsou vhodné i pro mé řešení.

### 1. Kvalitní mapové podklady

S tímto naprosto souhlasím, kvalitní mapa jako základ aplikace bude opravdu důležitým bodem mé práce.

### 2. Otevřená databáze míst, do které může přispívat i široká veřejnost

Tento bod do své práce zahrnout nemohu, jelikož mají být místa zcela ve správě Magistrátu města Zlín. Počáteční data budu čerpat z publikace o bezbariérovosti, viz kapitola 5.1. Další úpravy – přidání, odebrání či editaci budou mít již na starost zaměstnanci magistrátu.

### 3. Možnost filtrování mezi místy

Opět se jedná o část aplikace, která je podle mého velmi důležitá a kterou chci zahrnout i do svého řešení.

### 4. Možnost zobrazení detailu k požadovanému místu

Tento bod beru jako samozřejmost, jenž se musí objevit i v mé práci. Zobrazení konkrétních informací pro zvolené místo je nutností.

### 5. Za projektem a vývojem stojí rozsáhlý tým lidí

Tento bod již z podstaty diplomové práce nemůžu splnit, avšak během vývoje své práce ji budu konzultovat jak se svým vedoucím diplomové práce panem doc. Ing. Jířím Vojtěškem, Ph.D., tak s kolegy, známými a kamarády, kteří se tvorbou webů zabývají, abych dosáhl co možná nejlepšího konečného výsledku. Samozřejmostí je potom přizpůsobení webu požadavkům a přáním Magistrátu města Zlína.

## 6 IMPLEMENTACE, POUŽITÉ TECHNOLOGIE

Praktická část mé práce spočívala ve vytvoření webové aplikace, která má za cíl usnadnit život především tělesně handicapovaným lidem ze Zlína a okolí. Konkrétně by se díky ní měli dovědět informace o dostupnosti důležitých míst v okolí jejich bydliště z hlediska jejich přístupnosti. Jinak jednoduše řečeno mým úkolem bylo vytvořit webovou prezentaci na základě Publikace o bezbariérovosti, kterou jsem popsal v kapitole 5.1, jež by vhodným způsobem prezentovala informace v ní obsažené.

### 6.1 Vývoj webové aplikace

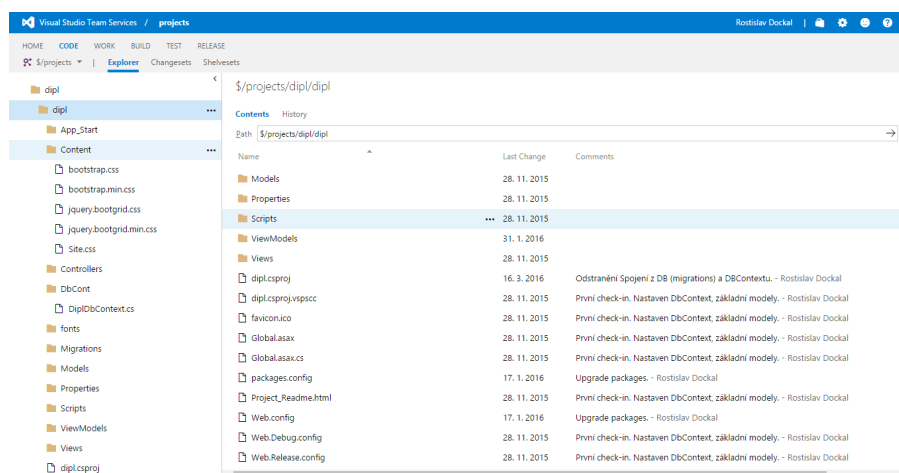
Webovou aplikaci jsem tvořil ve frameworku ASP.NET MVC 5 společnosti Microsoft s použitím JavaScriptové knihovny jQuery<sup>1)</sup> a s využitím dalších technologií, které jsou popsány v teoretické části mé práce, v kapitole 2. V této části textu je uvedeno jejich praktické použití při realizaci webové aplikace s využitím několika dalších technologií a vývojových nástrojů, které dopomohly zlepšit návrh a zkvalitnit tvorbu samotného projektu.

#### 6.1.1 TFS (Team Foundation Server)

V dnešní době jsou při vývoji aplikací využívány verzovací nástroje, které usnadňují vývojářům implementaci a kontrolu nad zdrojovými kódy a jejich úpravami. Při vývoji mé aplikace jsem zvolil verzovací nástroj Team Foundation Server, respektive jeho online verzi Visual Studio Online, jež Microsoft nabízí. Do pěti lidí je jeho použití bezplatné, za každého dalšího vývojáře je nutné zaplatit určitý měsíční poplatek. Tuto situaci jsem však nemusel řešit, jelikož jsem na projektu pracoval sám. Další nespornou výhodou je záloha takto verzovaných zdrojových kódů a možnost k nim přistupovat téměř odkudkoliv, stačí webový prohlížeč a připojení na internet. Příklad z webového rozhraní TFS je vidět na obrázcích níže.

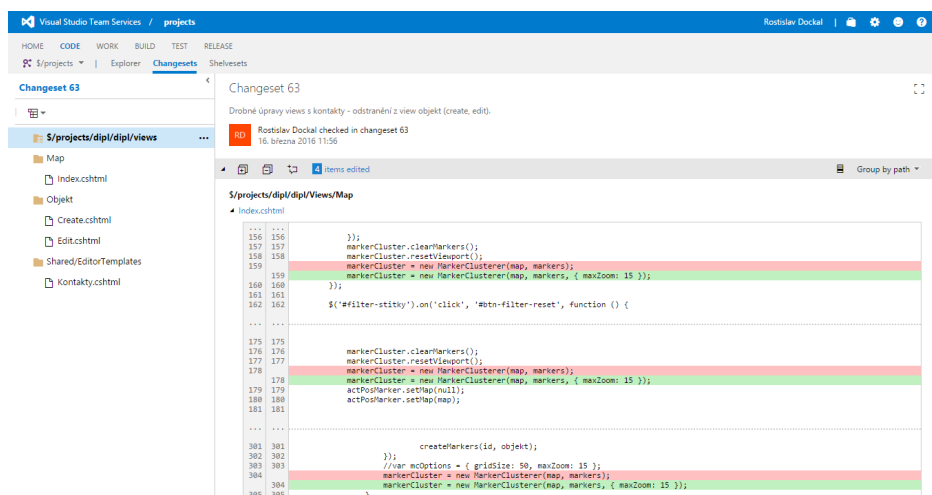
---

<sup>1)</sup><https://jquery.com>



Obr. 6.1 Struktura projektu ve webovém rozhraní TFS

Díky TFS je možné vytvářet tzv. changesety, což jsou určité sady změn ve zdrojových kódech aplikace. Díky tomuto verzování zdrojových kódů je možné jednoduše dohledat, jaké změny byly ve kterém changesetu provedeny a najít tak případně příčinu problému, který dříve v aplikaci nebyl, ale objevil se tam až společně s některou z pozdějších úprav. Této možnosti jsem v projektu sám několikrát využil. Navíc jsem díky tomu měl projekt zálohovaný i v případě poruchy mého počítače, na kterém jsem práci tvořil.



Obr. 6.2 Detail changesetu ve webovém rozhraní TFS

### 6.1.2 Responzivní design webu

Jedním z úkolů zadání mé diplomové práce bylo zajistit responzivnost vytvořené webové aplikace, tedy aby se do určité míry přizpůsobovala zařízením, respektive jejich displejům, na kterých je zobrazována. Za tímto účelem jsem zvolil použití frameworku Bootstrap, který jsem doplnil o vlastní úpravy vzhledu aplikace s využitím kaskádových stylů (CSS) a JavaScriptu.

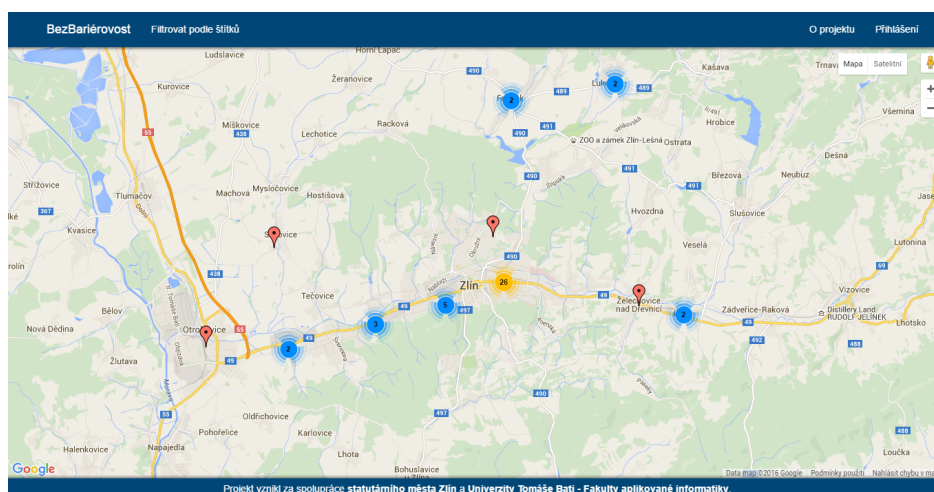
## Bootstrap

Bootstrap je frontedový Framework používaný při vytváření webových aplikací. Někdy je také označován jako „responzivní framework“, i když ve skutečnosti neřeší pouze responzivnost webu, nýbrž celou řadu další důležitých oblastí. Jedná se například o práci s typografií, formátováním tabulek, tvorby layoutu stránky a především vytváření komponent uživatelských rozhraní. Bootstrap využívá HTML, CSS a JavaScript a byl vyvinut společností Twitter, Inc.[18]

V diplomové práci jsem využil Bootstrap především při řešení základního layoutu stránky, včetně „mobilního menu“. Rovněž v administrátorské části aplikace byly využity možnosti tohoto frameworku. Konkrétně některé jeho formulářové komponenty – tlačítka, vstupní formulářová pole - tzv. inputy, varovná oznámení. Za pomoci bootstrapu bylo také dosaženo responzivnosti webových stránek, tedy jejich zobrazení na různých velikostech a rozlišeních displejů, od telefonů, přes tablety až po stolní počítače. Jednoduché otestování přechodů je možné zmenšením či zvětšením okna webového prohlížeče.

## Grafické rozvržení hlavní stránky (Layout)

Základní struktura hlavní stránky klade důraz na zobrazení mapy s jednotlivými místy. Mapa zobrazující mapové podklady od společnosti Google, Inc. je doplněna o ikony jednotlivých míst. Tato část zabírá většinu prostoru stránky jak je vidět na obrázku níže.

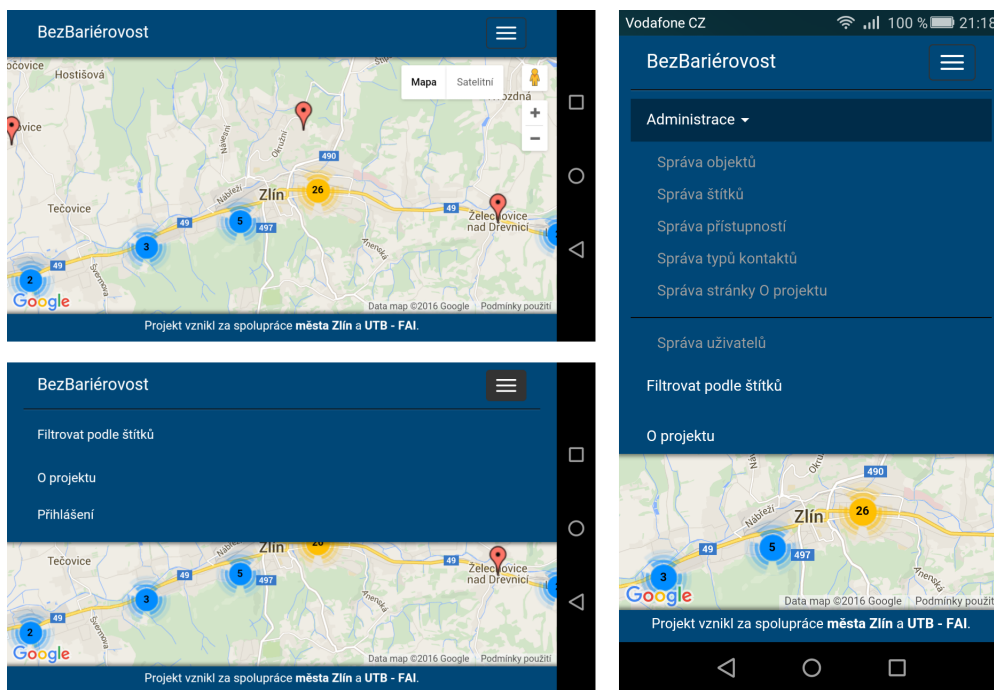


Obr. 6.3 Grafické rozvržení hlavní stránky pro PC

V horní části se nachází menu, které se v případě zobrazení na zařízeních s malým rozlišením displeje, přeskupí do tzv. sandwich menu<sup>2)</sup>, což je responzivní menu, kde jsou jednotlivé jeho položky seřazeny nikoli horizontálně do šířky jako v klasickém zobrazení,

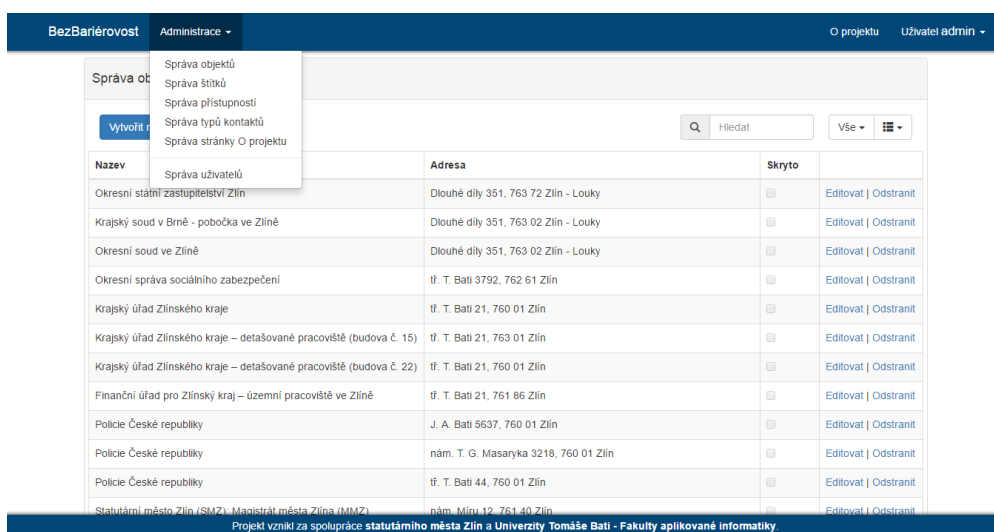
<sup>2)</sup><https://www.newfangled.com/user-experience-mobile-menus-and-nav-burgers>

ale vertikálně pod sebou. Ve spodní části je vytvořena patička stránky, která je pevně přichycena ke spodní části stránky. Při změnách výšky okna prohlížeče se tedy mění pouze velikost segmentu s mapou a menu i patička zůstávají umístěny stejně. Příklady zmíněného menu jsou vidět na následujícím obrázku.

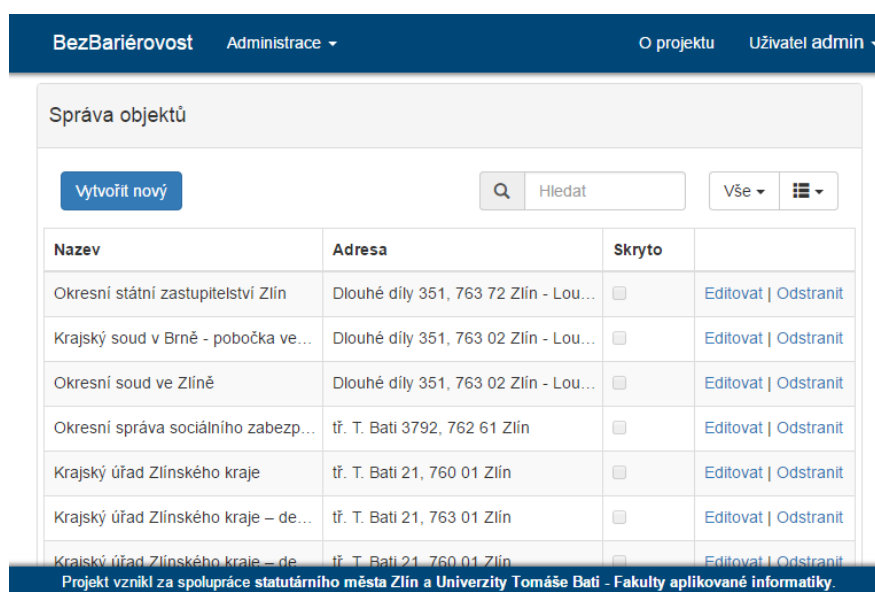


Obr. 6.4 Použití sandwich menu u mobilního zobrazení stránek

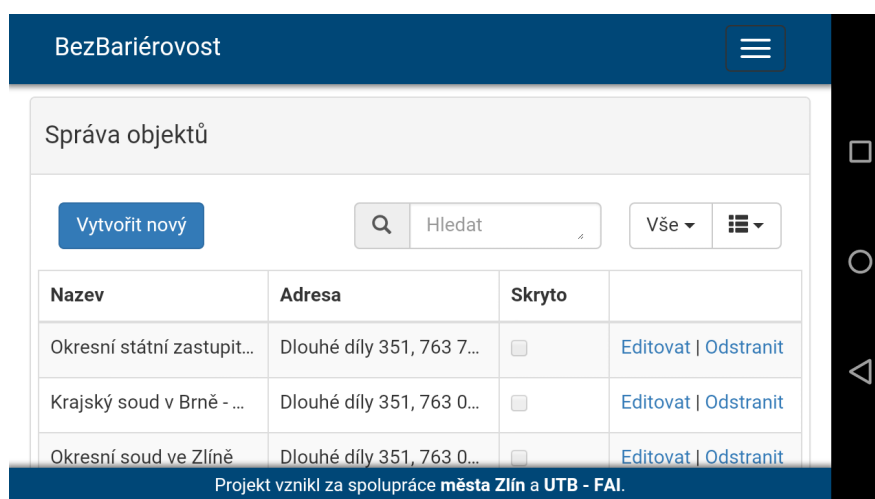
Administrátorská sekce byla také vytvořena s přihlédnutím k možnosti správy důležitých dat systémů v mobilních zařízeních - telefonech a tabletech. Na uvedených obrázcích je vidět náhled aplikace na klasickém počítači, dále zobrazení na tabletu a také na mobilním telefonu.



Obr. 6.5 Administrátorská sekce zobrazena v PC



Obr. 6.6 Administrátorská sekce zobrazena v tabletu



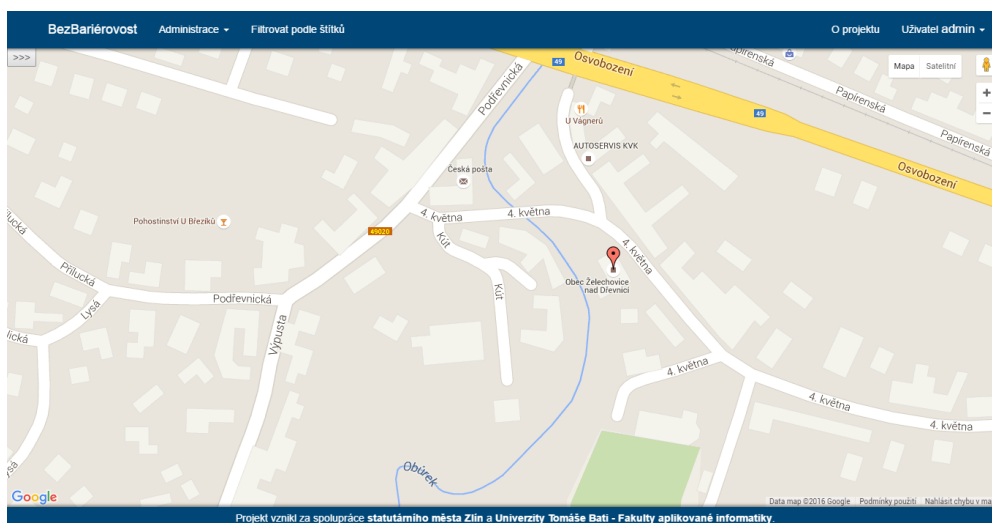
Obr. 6.7 Administrátorská sekce zobrazena v mobilním telefonu

### 6.1.3 Google Maps – mapové podklady

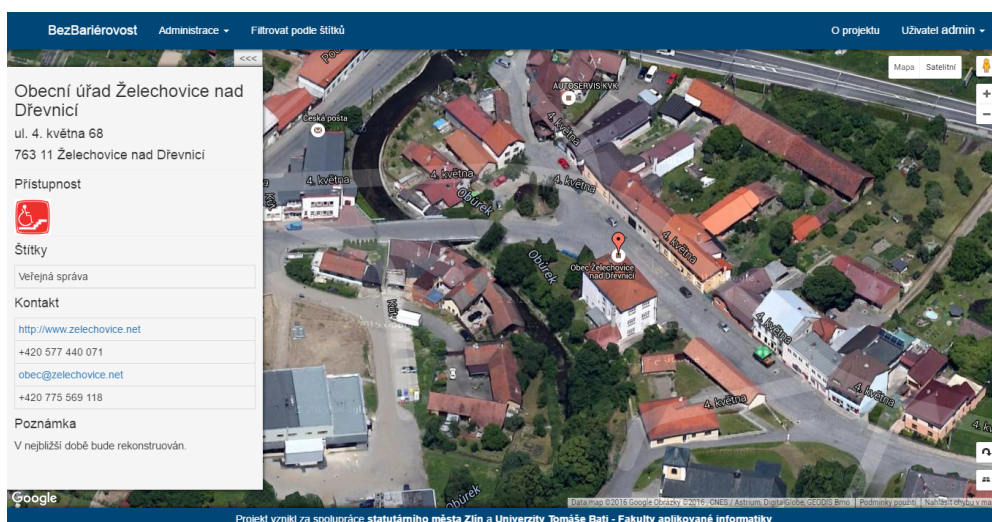
Jak bylo zmíněno v předešlém odstavci, většinu hlavní stránky má zabírat mapa. Bylo tedy potřeba sáhnout ke kvalitním mapovým podkladům. Zároveň by zvolené mapy měly také nabízet možnost posunu na mapě, přibližování a oddalování apod. Rozhodnutí padlo na použití mapových podkladů společnosti Google, Inc., která poskytuje mimo základní zobrazení mapy i pokročilé možnosti pro práci s mapou pomocí Google Maps JavaScript API (viz kapitola 2.8.1). To podporuje možnost vkládání vlastních značek přímo do mapy, čehož jsem využil pro zobrazení jednotlivých míst na mapě. S použitím mapových podkladů je svázána nutnost vygenerování a použití tzv. API-KEY, což je jedinečný identifikátor, který umožňuje použití tohoto API a váže se

na vývojářský účet a projekt, na kterém mají být mapy zobrazovány. Jistým omezením těchto mapových podkladů je možnost bezplatného použití do 25 tisíc zobrazení za den. Nad tuto hodnotu je již využívání API zpoplatněno. S takovou návštěvností se ale u projektu vzhledem k jeho zaměření nedá počítat.

Níže na obrázcích je vidět možnost zobrazení různých typů mapových podkladů, přičemž se jedná o totéž místo. Na druhém obrázku je navíc vidět detail s bližšími informacemi o vybraném místě.



Obr. 6.8 Defaultní styl zobrazení mapy



Obr. 6.9 Satelitní zobrazení mapy s detailem objektu

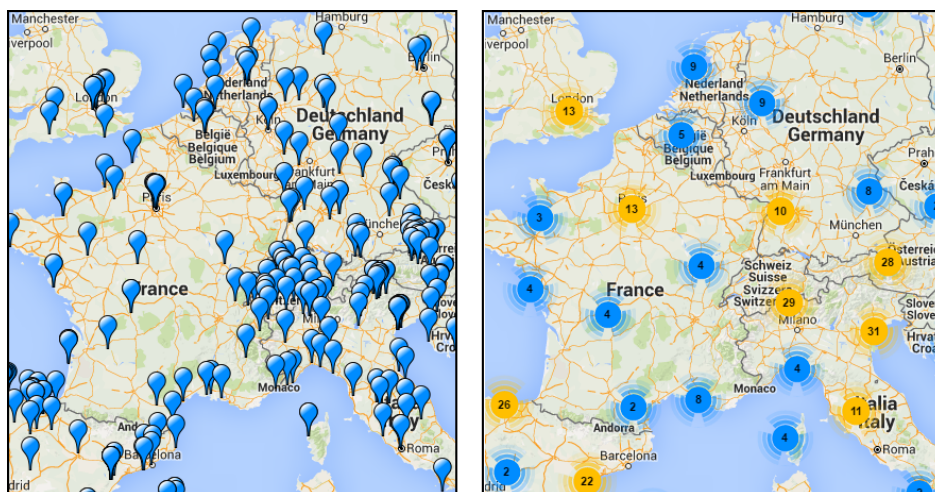
Samotné vložení základní mapy na web je poměrně jednoduché. Příklad části zdrojového kódu je vidět na níže uvedeném obrázku. Jedná se o nastavení div elementu stránky s atributem `id = map`, do něhož je umístěna mapa. Jsou nastaveny základní parametry jako přiblížení, vycentrování mapy, povoleny ovládací prvky a povoleno zob-

razení StreetView<sup>3)</sup>). Vložení mapy do stránky bylo použito, kromě již uvedeného zobrazení na hlavní stránce, také v administrátorské části při zadávání případně editaci míst.

```
<div id="map"></div>
<script src="https://maps.googleapis.com/maps/api/js?v=3.exp&key=API-KEY"></script>
<script type="text/javascript">
  function initialize() {
    map = new google.maps.Map(document.getElementById('map'), {
      zoom: 12,
      center: { lat: 49.226835, lng: 17.666720 },
      mapTypeControl: true,
      streetViewControl: true,
    });
  }
  google.maps.event.addDomListener(window, 'load', initialize);
</script>
```

Obr. 6.10 Zdrojový kód pro vložení mapy do stránky

Jelikož bude ve Zlíně a okolí na mapě zobrazeno velké množství míst, respektive značek těchto míst, což by vedlo k obrovské nepřehlednosti, musel jsem tuto otázku zobrazování velkého počtu značek vyřešit. Ve spojení s Google Maps Javascript API jsem použil JavaScriptovou knihovnu MarkerClusterer for Google Maps v3<sup>4)</sup>. Jejím úkolem je především shlukování značek tak, aby jich nebylo v jednom okamžiku na mapě neúměrně velké množství. Tento problém je řešen vytvořením zástupné značky, která pod sebou ukrývá větší množství původních značek, přičemž je v ní uveden počet těchto zastoupených značek. Příklad rozdílu zobrazení mapy se značkami bez a s řešením pomocí shlukování značek je uveden na obrázku níže.



Obr. 6.11 Mapa bez shlukování značek (vlevo) a mapa s využitím shlukování (vpravo)

<sup>3)</sup><https://www.google.com/maps/streetview>

<sup>4)</sup><https://github.com/googlemaps/js-marker-clusterer>

## 6.2 Databáze a ORM

Pro ukládání potřebných dat v aplikaci jsem zvolil použití relační databáze, což je v dnešní době standardem na poli webových aplikací. Konkrétně jsem využil databázový systém Microsoft SQL Server od společnosti Microsoft (viz kapitola 2.5.1). Jedná se o prověřenou, odladěnou a v praxi velmi často používanou technologii, která se navíc velmi dobře hodí pro použití ve spojení s objektově relačním mapování pomocí Entity Frameworku (EF). Ten se rovněž v současnosti těší velké oblibě vývojářů. Jeho použití je bezpochyby vhodné pro menší a střední projekty, což ten můj systém splňuje. Pro rozsáhlé projekty a pro projekty pracující s velkým množstvím dat je pravděpodobně vhodnější použití technologie NHibernate. Pro výběr EF ale hovoří i fakt, že vývoj v něm je poměrně rychlý, má strmou učitelskou křivku a hodí se v případě, kdy máme ve správě i fyzický model DB, což se v mém případě stalo.

### 6.2.1 Entity Framework

Jak jsem zmínil, Entity Framework je dobře použitelný ve spojení s Microsoft SQL databází. V aplikaci jsem využil tzv. Model first přístup. Díky tomu dokáže EF podle třídy v aplikaci vytvořit odpovídající tabulku v databázi. Pokud v aplikaci vytvořím objekt této třídy, je možné jej za pomoci EF uložit do této připravené databázové tabulky. Příkladem využití v mém projektu může být následující situace. V aplikaci jsem potřeboval pracovat s třídou Objekt, jenž zastupuje jednotlivé objekty (místa). Vytvořil jsem tedy v jazyce C# odpovídající třídu.

```
public class Objekt
{
    public int ID { get; set; }
    public string Nazev { get; set; }

    [DataType(DataType.MultilineText)]
    [AllowHtml]
    public string Poznamka { get; set; }

    public bool Skryto { get; set; }

    public int? AdresaID { get; set; }
    public virtual Adresa Adresa { get; set; }

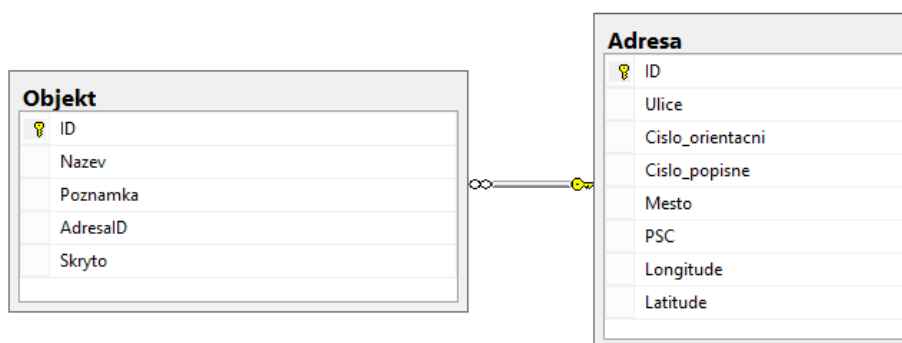
    public virtual ICollection<Stitek> Stitky { get; set; }
    public virtual ICollection<Pristup> Pristupnost { get; set; }
    public virtual ICollection<Kontakt> Kontakty { get; set; }
}
```

Obr. 6.12 Definice třídy Objekt

Za pomoci EF jsem v databázi vytvořil odpovídající tabulku nazvanou Objekt. Na výše uvedeném příkladu je navíc vidět i princip, jakým se v Entity Frameworku

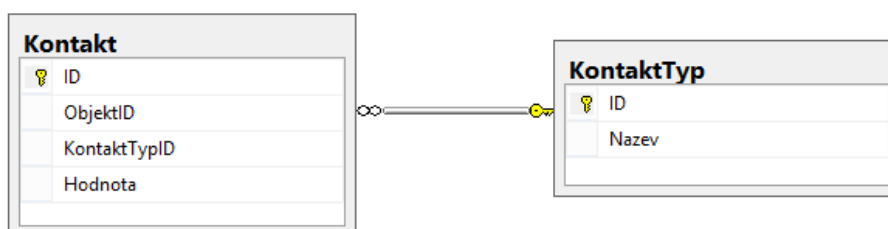
mohou definovat vazby 1:1 a M:N mezi třídami v aplikaci, respektive mezi tabulkami v databázi. Samozřejmě, že je možné použít i vazbu 1:N, ale ta na tomto příkladu nefiguruje.

Pro provázání 1:1 slouží dvojice atributů `public int? AdresaID` a `public virtual Adresa Adresa` určuje, že v tabulce `Objekt` bude figurovat cizí klíč z tabulky `Adresa`. Tedy že pro jeden `Objekt` bude možné definovat vždy jednu `Adresu`. Schéma tohoto vztahu v databázi je vidět na obrázku níže.



Obr. 6.13 Příklad vazby 1:1

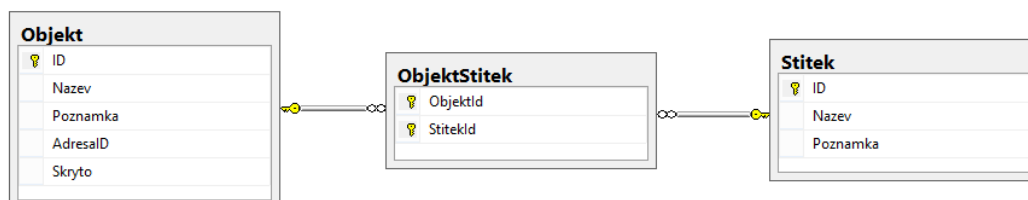
Vazbu 1:N jsem použil v projektu například u typů kontaktů. Každý objekt může mít definovanou celou řadu kontaktů. Každý tento kontakt však musí mít svůj typ – je potřeba mít informaci o tom, zda se jedná o telefonní spojení, webovou stránku, e-mailovou adresu aj. Zde vznikl prostor pro vazbu mezi samotnými kontakty a typem kontaktu.



Obr. 6.14 Příklad vazby 1:N

Příkladem vazby M:N je poté atribut `public virtual ICollection<Stitek> Stitky`. Na třídě `Objekt` je ale pouze polovina potřebné informace, protože i na třídě `Stitek` musí být uvedena zpětná vazba na `Objekt` jako `public virtual ICollection<Objekt> Objekty`. Pokud provedeme oboustrannou definici vazby, EF je díky tomu schopno interpretovat požadovanou vazbu M:N i do databáze. Ve výsledku je v databázi vytvořena tzv. vazební tabulka, která v sobě obsahuje alespoň dva sloupce, ve kterých jsou ukládány identifikátory dvou propojovaných entit. Jako příklad této M:N vazby mohu z projektu zmínit propojení tříd `Objekt` a `Stitek`, kdy jeden objekt může mít

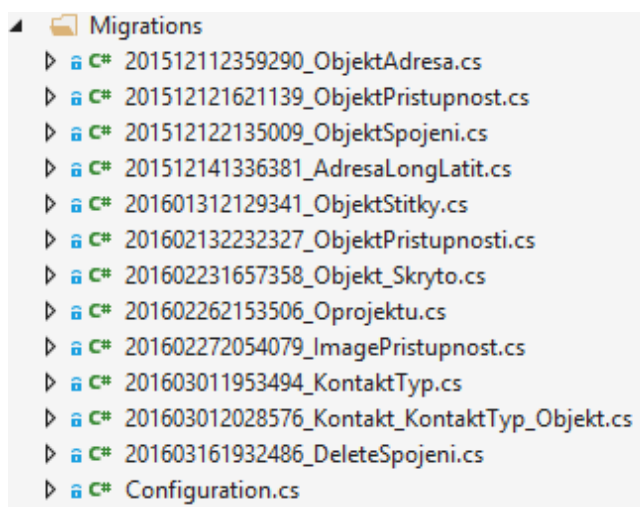
přiřazeno více štítků a naopak jeden štítek může mít přiřazeno více objektů. Muselo tak dojít k vytvoření vazební tabulky, která v každém záznamu obsahuje identifikátor objektu a štítku. Díky uvedení výše zmíněné dvojice atributů EF pozná, že chceme třídu **Objekt** a **Stítek** propojit pomocí vazby M:N a díky tomu se postará o vytvoření potřebné vazební tabulky.



Obr. 6.15 Příklad vazby M:N

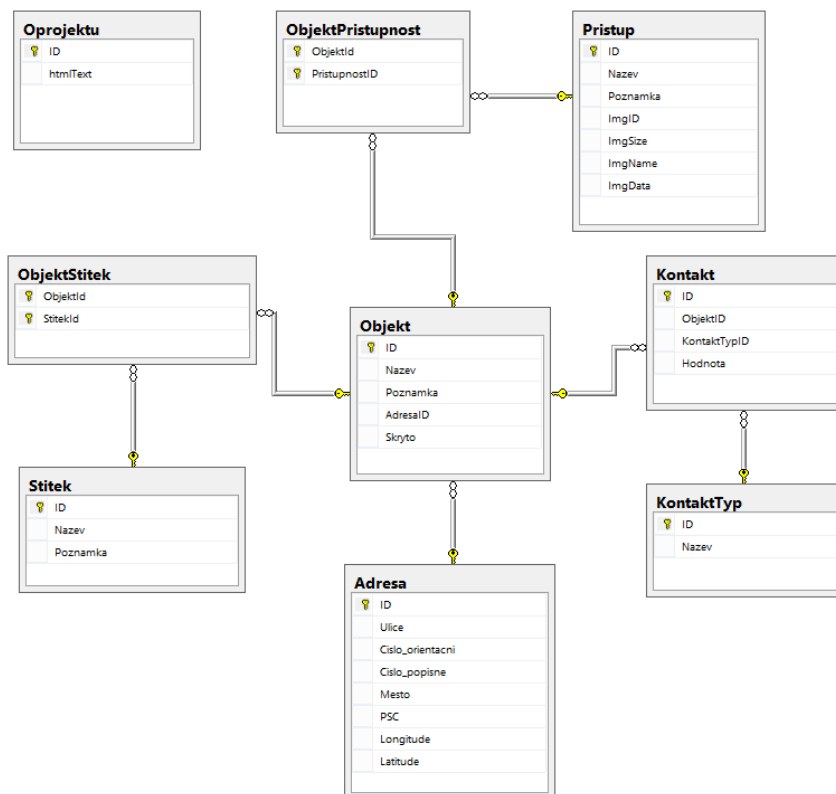
### 6.2.2 Entity Framework – Migrations (Migrace)

Entity Framework mi pomohl na základě definic jednotlivých tříd připravit potřebné databázové tabulky. Při vyvíjení aplikace jsem se ale dostal do situace, kdy jsem potřeboval provést změny na některých dříve definovaných třídách, například přidat nebo odebrat nějaký atribut. Za tímto účelem jsem použil tzv. migrace (migrations). Jedná se o jednu z funkcionalit EF, která dovoluje určitým způsobem verzování databáze – přechod mezi jednotlivými verzemi tam i zpět. Na začátku jsem si vytvořil první inicializační migraci, která měla za úkol základní nastavení databáze, včetně vytvoření prvotní verze tabulek. V mém případě nesla název `201512112359290_ObjektAdresa`, jenž se skládal z časové značky jejího vytvoření, ke které jsem přidal označení, čeho se tato migrace týká. Výpis posloupnosti jednotlivých migrací je vidět na následujícím obrázku.



Obr. 6.16 Seznam migrací

Přehled důležitých databázových tabulek, jejich atributů, datových typů a povinností je uveden v příloze P2. Níže na obrázku jsou uvedeny tyto důležité databázové tabulky a jejich vzájemné propojení.



Obr. 6.17 Diagram důležitých databázových tabulek systému

### 6.3 Administrátorská část systému

Jedním z hlavních požadavků na systém, který figuruje i mezi body oficiálního zadání mé diplomové práce, je vytvoření administrátorské části systému, jenž má dovolovat snadnou správu míst a uživatelských účtů. Vzhledem k tomuto faktu musí aplikace dovolovat vytváření uživatelů (uživatelských účtů). Každý z nich poté bude mít určenu tzv. systémovou roli, která bude přesně vymezovat, jaké sekce z administrátorského rozhraní systému bude mít daný uživatel dostupné. Každá ze šesti vytvořených sekcí je popsána níže v této kapitole.

#### 6.3.1 Uživatelé a uživatelské role

V systému jsou vedle klasického návštěvníka, u kterého není vyžadována žádná registrace ani přihlášení, vytvořeny tři uživatelské role, jenž vyžadují přihlášení k předem definovanému uživatelskému účtu. Původně byly vytvořeny pouze dvě – Admin a Pracovník. Později během vývoje přibyla ještě jedna role Zadavatel. Na následující tabulce

je uveden přehled rolí a jejich možnosti přístupu do jednotlivých sekcí administrátorské části.

Tab. 6.1 Přehled přístupu k administrátorským sekcím podle uživatelských rolí

	Admin	Pracovník	Zadavatel
Správa objektů	✓	✓	✓*
Správa štítků	✓	✓	–
Správa přístupností	✓	✓	–
Správa typů kontaktů	✓	✓	–
Správa stránky O projektu	✓	✓	–
Správa uživatelů	✓	–	–
* Uživatel s rolí Zadavatel nemá ve Správě objektů oprávnění smazat existující objekt.			

### 6.3.2 Přihlašování, autorizace a autentifikace

Jelikož je potřeba mít v aplikaci rozděleny uživatele podle výše jejich práv, bylo třeba vyřešit dvě základní věci – jak vytvořit uživatelský účet a jak se k existujícímu účtu přihlásit. Vytvoření uživatelského účtu je popsáno níže v kapitole 6.3.8. V této části se zaměřím na přihlašovací proces.

Pro upřesnění je potřeba definovat dva pojmy, které s přihlášením a následnou prací v aplikaci pod určitým uživatelem, by měly být zřejmé a objevují se téměř ve všech publikacích ohledně zabezpečení webových aplikací.[24]

#### 1. Autentizace (Authentication)

Pod tímto termínem se v podstatě skrývá proces uživatelského přihlášení. Autentizace označuje ověření uživatele, že je skutečně tím, za koho se vydává. V drtivé většině webových aplikací je autentizace tvořena kontrolou zadaného uživatelského jména a odpovídajícího hesla. Existuje ale i také tzv. dvoufaktorová autentizace, která spočívá v posloupnosti dvou nezávislých ověření. Jako příklad mohu uvést přihlašování do internetového bankovníctví, kde je třeba zadat přihlašovací jméno a správné heslo. Pokud splníme tuto první část, systém odešle SMS na telefonní číslo přiřazené k tomuto uživatelskému účtu. Pro úspěšné přihlášení je poté třeba ještě zadat i kód, který je doručen pomocí zmíněné SMS.

#### 2. Autorizace (Authorization)

Autorizací se rozumí proces, kdy je autentizovaný uživatel ověřován, zda má přístup k požadovanému obsahu webových stránek, na který se snaží přistoupit.

Samozřejmě zde není řešeno pouze, zda má či nemá k požadovanému přístup, ale i jaký typ přístup mu má být povolen – zda může obsah pouze číst, nebo jej může editovat, případně má možnost vytvářet nové záznamy apod.

## ASP.NET Identity

Pro práci s rolemi a uživatelskými účty jsem ve své práci použil technologii ASP.NET Identity<sup>5)</sup>. Podporuje jak možnost autentizace uživatele, tak i následnou autorizaci uživatele pro jednotlivé části aplikace. Navíc defaultní nastavení databázových tabulek, které slouží k ukládání potřebných informací o uživateli a jejich rolích, je možné podle potřeb modifikovat. Tuto možnost jsem prozatím v aplikaci nevyužil a plně mi pro mé potřeby dostačovala předdefinovaná struktura. Pro přihlašování slouží klasický jednofaktorová autorizace pomocí uživatelského jména a hesla. V budoucnu by ale v případě potřeby bylo možno do systému doplnit například ověřování ještě i pomocí SMS kódu.

### 6.3.3 Správa objektů

Stěžejním bodem projektu jsou objekty, nebo jinak řečeno místa, okolo kterých je celá aplikace postavena. V administrátorské části má tedy určitě své místo správa těchto míst. Na obrázku níže je vidět úvodní přehled objektů, ve kterém je možné fultextově vyhledávat či si tabulku seřadit podle některého z uvedených sloupců. Uživatel zde má přístup k možnostem vytvoření nového objektu, případně editaci nebo odstranění již existujících.

The screenshot displays the 'Správa objektů' (Object Management) interface. At the top, there is a navigation bar with 'BezBariérovost', 'Administrace', 'O projektu', and 'Uživatel admin'. Below this, there is a 'Vytvořit nový' (Create new) button and a search bar labeled 'Hledat'. The main content is a table with the following columns: 'Název' (Name), 'Adresa' (Address), 'Skryto' (Hidden), and a column for actions ('Editovat | Odstranit'). The table lists various locations and institutions, such as 'Okresní státní zastupitelství Zlín', 'Krajský soud v Brně - pobočka ve Zlíně', and 'Policie České republiky'. The footer of the page reads: 'Projekt vznikl za spolupráce statutárního města Zlína a Univerzity Tomáše Bati - Fakulty aplikované informatiky'.

Název	Adresa	Skryto	
Okresní státní zastupitelství Zlín	Dlouhé díly 351, 763 72 Zlín - Louky	<input type="checkbox"/>	Editovat   Odstranit
Krajský soud v Brně - pobočka ve Zlíně	Dlouhé díly 351, 763 02 Zlín - Louky	<input type="checkbox"/>	Editovat   Odstranit
Okresní soud ve Zlíně	Dlouhé díly 351, 763 02 Zlín - Louky	<input type="checkbox"/>	Editovat   Odstranit
Okresní správa sociálního zabezpečení	tř. T. Bati 3792, 762 61 Zlín	<input type="checkbox"/>	Editovat   Odstranit
Krajský úřad Zlínského kraje	tř. T. Bati 21, 760 01 Zlín	<input type="checkbox"/>	Editovat   Odstranit
Krajský úřad Zlínského kraje – detašované pracoviště (budova č. 15)	tř. T. Bati 21, 763 01 Zlín	<input type="checkbox"/>	Editovat   Odstranit
Krajský úřad Zlínského kraje – detašované pracoviště (budova č. 22)	tř. T. Bati 21, 760 01 Zlín	<input type="checkbox"/>	Editovat   Odstranit
Finanční úřad pro Zlínský kraj – územní pracoviště ve Zlíně	tř. T. Bati 21, 761 86 Zlín	<input type="checkbox"/>	Editovat   Odstranit
Policie České republiky	J. A. Bati 5637, 760 01 Zlín	<input type="checkbox"/>	Editovat   Odstranit
Policie České republiky	nám. T. G. Masaryka 3218, 760 01 Zlín	<input type="checkbox"/>	Editovat   Odstranit
Policie České republiky	tř. T. Bati 44, 760 01 Zlín	<input type="checkbox"/>	Editovat   Odstranit
Statutární město Zlín (SMZ) - Magistrát města Zlína (MMZ)	nám. Míru 12, 761 40 Zlín	<input type="checkbox"/>	Editovat   Odstranit

Obr. 6.18 Správa objektů – úvodní stránka

<sup>5)</sup><http://www.asp.net/identity>

Zadání nového objektu (místa) probíhá na formuláři, který je uveden na obrázku 6.19. Vytvoření nového objektu. První položku, která se nabízí k vyplnění, je název objektu. Poté je možné vyplnit poznámku. Toto pole obsahuje WYSIWYG editor, který dovoluje mimo klasickou poznámku ve formě jednoduchého textového řetězce i možnost základního formátování textu. Volby jako tučné či podtržené písmo, kurzíva, výběr velikosti a barvy písma, jeho zarovnání, číslování nebo vložení odkazu byly přidány na požadavek pracovníků magistrátu města Zlín. Všechny tyto možnosti jsou ihned po jejich aplikaci zobrazeny a uživatel tak vidí přímo výsledek, v němž bude poznámka uložena a později i zobrazována návštěvníkům. V aplikaci jsem zvolil použití WYSIWYG editoru NicEdit<sup>6)</sup>, který je přehledný a nabízí možnosti editace, jenž odpovídají požadavkům.

Další částí formuláře je určena pro zadání adresy včetně označení místa na mapě tam, kde bude zobrazen návštěvníkům stránek. Po zadání ulice, čísla popisného a orientačního, města a PSČ je možné tlačítkem „Najít místo na mapě podle adresy“ vyhledat pozici objektu na mapě. Pro tuto činnost je využito možností Google API pro vyhledávání. Pokud toto určení polohy není dostatečně přesné, je možné v mapě ukazatel (ikonu) objektu posunout podle potřeby, přičemž posun se provádí kliknutím a tažením ikony objektu na požadované místo. Automaticky jsou potom doplněny pole pro zeměpisnou šířku a délku.

Po vyplnění umístění objektu je třeba určit, které typy přístupností objekt splňuje. K tomuto slouží checkboxy (zaškrtačací políčka) s názvy jednotlivých přístupností. Jejich seznam je sestavován dynamicky podle typů přístupností definovaných v jedné z dalších částí administrátorské části systému, viz 6.3.5. Následující krok spočívá ve vyplnění kontaktů, jenž jsou se zadávaným objektem spojeny. Jedná se například o adresu webových stránek, telefonní čísla nebo emailové adresy. Správa jednotlivých možností je opět dostupná v administrátorské sekci, viz 6.3.6.

Posledním krokem je určení štítků, pod které daný objekt patří. Přiřazen může být k libovolnému množství z nich. Jejich definování je opět na uživateli starajících se o správu aplikace. Štítky mohou přidávat, upravovat či odstraňovat viz 6.3.4.

Celý proces vytvoření objektu je dokončen tlačítkem „Vytvořit nový objekt“.

---

<sup>6)</sup><http://nicedit.com>

BezBariérovost Administrace - O projektu Uživatel admin -

### Vytvoření nového objektu

Název

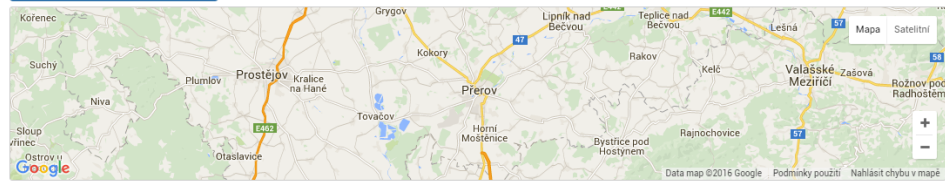
Poznámka  
B I U X<sub>2</sub> Velikost písma...

Adresa

Ulice  Čís. popisné  Čís. orientační

Město  PSČ

[Najít místo na mapě podle adresy](#)



Zeměpisná šířka (Latitude)  Zeměpisná délka (Longitude)

Přístupnosti

Přístupný  Částečně přístupný  Nepřístupný

Přístupná toaleta - WC I  Přístupná toaleta - WC II  Vyhrazené parkovací místo

Kontakty

Upozornění: Odkazy na webové stránky musí začínat `http://` nebo `https://`.

Web  Telefon

Telefon 2  E-mail

E-mail 2

Štítky

škola  úřad  restaurace

[Zpět](#) [Vytvořit nový objekt](#)

Obr. 6.19 Formulář pro vytvoření nového objektu

V rámci objektů má uživatel ještě jednu možnost a tou je skrytí objektu pro návštěvníky. Ve správě jej lze libovolně upravovat, ale návštěvníkům není takovýto objekt zobrazen. Tohoto mechanismu je možné využít například v případě, že se určitý úřad stěhuje. Bylo by zbytečné o celý objekt přijít jeho odstraněním a po přestěhování jej znovu zakládat. Lepší volbou je jeho dočasné skrytí. V této době je u dotyčného objektu změněna adresa na nové působiště a po dokončení stěhování, jakmile začne úřad opět fungovat, pouze nastavíme objekt opět jako viditelný.

Podobně jako zadání nového objektu probíhá i editaci dříve vytvořených, přičemž formulář vypadá na první pohled totožně. Jsou v něm však předvyplněny údaje z editovaného objektu a ty je následně možné upravovat.

### 6.3.4 Správa štítků

Jedním z požadavků bylo vytvoření štítků, tedy jakýchsi kategorií, do kterých mohou být objekty zařazovány. Návštěvníci mají poté možnost podle těchto štítků objekty filtrovat a nechat si zobrazit jen určitou část z nich. Jelikož množina štítků není pevně vymezena a systém má dovolovat jejich přidávání, editaci a mazání, přidal jsem do administrátorské části všechny tyto možnosti.

### 6.3.5 Správa přístupností

Podobná situace jako v případě štítků byla i u typů přístupností. V publikaci o bezbarierovosti (viz kapitola 5.1) byla sice definována šestice typů přístupností objektů, ale do budoucna se nedá předpokládat, že tento seznam bude finální. V aplikaci je tedy možné ve správě přístupností přidávat, editovat či odstraňovat tyto přístupnosti. Ke každé z nich je možné přidat i obrázek (ikonu), která bude zobrazena v detailu objektu, jenž má tuto přístupnost přiřazenu. Náhled formuláře pro vytvoření nového typu přístupnosti je vidět na obrázku níže.

Obr. 6.20 Formulář pro vytvoření nové přístupnosti

### 6.3.6 Správa typů kontaktů

Původně jsem v aplikaci typy kontaktů definoval napevno. Rozšiřujícím požadavkem však bylo, aby bylo možné tyto typy kontaktů mít také pod správou. Tedy mít možnost v případě potřeby přidat nějaký nový typ kontaktu, případně některý existující upravit či úplně odstranit. Vytvořil jsem proto sekci v administrátorské části aplikace, kde je možné všechny potřebné úkony provést. Jakmile je přidán nový typ kontaktu, po následujícím otevření formuláře pro vytvoření či editaci objektu je ihned možno vyplnit tento nový typ kontaktu.

### 6.3.7 Správa stránky O projektu

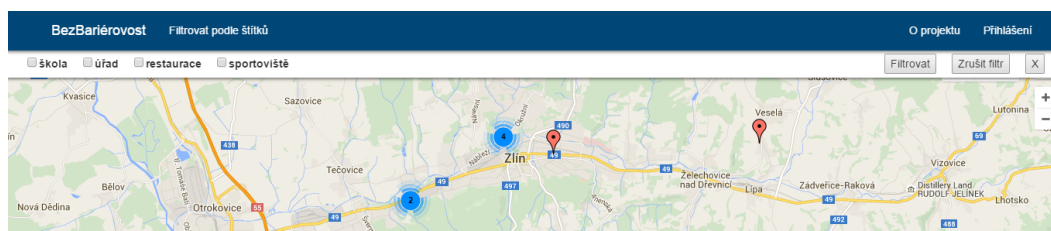
Správa stránky O projektu obsahuje pouze jednu možnost a tou je editace textu, který se zobrazí v aplikaci při otevření stránky O projektu. Opět jako v případě pole poznámky u objektu je zde k dispozici WYSIWYG editace tohoto textu. Na samotné stránce O projektu se potom zobrazuje tento zadaný text včetně jeho formátování. Pod ním je potom vložen seznam definovaných typů přístupností. Ten zde slouží jako určitá legenda k uvedeným přístupnostem.

### 6.3.8 Správa uživatelů

Správu uživatelů má dostupnou pouze jediná z uživatelských rolí, a to konkrétně role „administrátor“. Pouze uživatel, který má přiřazenu tuto roli, může přidávat nové uživatelské účty, přičemž každému musí určit roli, pod kterou bude daný uživatel v aplikaci vystupovat, a podle které bude mít přiřazeny práva. Dále má možnost editovat již existující záznamy, a rovněž může odstranit uživatelský účet v případě, že daný uživatel již do systému nemá mít přístup.

## 6.4 Filtrování podle štítků

V administrátorské části aplikace existuje možnost správy štítků, které jsou poté přiřazovány jednotlivým objektům. Běžný návštěvník stránek pak má možnost si z horního menu otevřít seznam těchto štítků. Zaškrtnutím si z nich vybrat ty, které ho zajímají, a poté si nechat kliknutím na tlačítko „Filtrovat“ zobrazit na mapě pouze ty objekty, jenž mají některý z vybraných štítků přiřazen. Tento seznam štítků je možné tlačítkem „X“ skrýt. Stisknutím tlačítka „Zrušit filtr“ je potom aplikovaný filtr odstraněn a na mapě jsou zobrazeny všechny dostupné objekty. Dříve označené štítky v seznamu dále zůstávají vybrané pro možnost opětovného vyfiltrování objektů podle nich.



Obr. 6.21 Panel pro filtrování podle štítků

## 6.5 Zabezpečení

Zabezpečení aplikace je v dnešní době v rámci tvorby webových aplikací velmi důležitým a často omílaným tématem. I z tohoto důvodu bylo zabezpečení systému uvedeno jako jeden z bodů oficiálního zadání mé práce. Bezesporu do této kapitoly spadá přihlašování uživatelů. Není totiž přípustné, aby se k chráněným informacím dostal jakýkoli uživatel, který stránky navštíví. K tomuto sloučí konkrétně procesy autentizace a autorizace uživatele, kterým jsem se věnoval v jedné z předchozích částí práce, viz kapitola 6.3.2. Pokud se uživatel pokusí přistoupit do části aplikace, ke které nemá jeho role povolen přístup, zobrazí se mu následující informace a přístup k požadovanému obsahu je mu tak odepřen.[25]



Obr. 6.22 Stránka zobrazující informaci, že uživatel nemá k požadovanému obsahu povolen přístup.

Kromě správného fungování jednotlivých rolí a uživatelských účtů v aplikaci je ale nutno řešit i potenciálně možné bezpečnostní útoky a předcházet jim. Níže je výčet útoků a u každého z nich je uvedeno, jakým způsobem je proti němu aplikace chráněna.

### 1. Injection a SQL Injection

Injection útokem se rozumí snaha útočníka pomocí využití špatného ošetření uživatelského vstupu podvrhnout systémů jinou zprávu, než tento systém očekává. Pokud takovouto zprávu útočník dokáže sestavit ve vhodném tvaru, a pokud není aplikace na tuto možnost nijak připravena, můžou být díky tomu získána například požadovaná data z databáze, ke kterým by útočník neměl mít správně přístup apod. Nejčastějším případem tohoto útoku je tzv. SQL Injection, který má právě za úkol získat na první pohled nedostupná data.

V mém řešení jsem možnosti tohoto útoku předešel tím, že nikde přímo v kódu neskládám SQL dotazy do databáze přímo z textových řetězců zadaných uživateli, ale pro získání potřebných dat využívám možnosti poskytované ORM technologií Entity Framework, jenž data z databáze pro aplikaci zprostředkovává.

## 2. Cross-Site Scripting

Tento útok využívá situace, kdy jsou na stránkách vstupy od uživatele zobrazovány přímo ve své surové podobě, tedy přesně tak jak je uživatel zadal. Pokud například uživatel do textového pole pro adresu zadal na místo správné informace do tohoto pole škodlivý skript, tak ve chvíli, kdy bychom se mu snažili tento obsah pole v nezměněné podobě zobrazit, provedl by se tento skript, který zadal. Je tedy potřeba nahradit klíčové znaky jazyka HTML za jejich entitní ekvivalenty, např. > za `&gt;`; [25]

Jelikož ASP.NET MVC Framework automaticky testuje veškeré uživatelské vstupy, neměl by v aplikaci tento útok nikdy nastat. Respektive pokud by k útoku došlo, tak aplikace takovýto vstupní řetězec nepřijme a vrátí výjimku

`System.Web.HttpRequestValidationException`. Jediným místem, kde jsem musel tuto kontrolu potlačit pomocí direktivy `[AllowHtml]` je použití WYSIWYG editoru u poznámek ve správě objektů, přístupností a také na editaci stránky O projektu. Pro přístup k těmto funkcionalitám však musí být uživatel autentizován a autorizován, jednalo by se tedy o jednoho z interních uživatelů aplikace.

## 3. Cross-Site Request Forgery

Cross-Site Request Forgery znamená v překladu „podvržení požadavku mezi různými stránkami“. Jedná se o útok na službu či webovou aplikaci, při němž se snaží uživatel přimět k návštěvě určité stránky aplikace. Ta provádí na pozadí nějakou akci, o které uživatel vůbec netuší, že je prováděna. Typickým příkladem může být využití HTML tagu `img`, do jehož parametru `src` vložíme adresu stránky, kterou útočník chce, aby přihlášený uživatel navštívil. Může se jednat například o zneužití URL pro odstranění určitého článku. Tuto činnost má přihlášený uživatel v případě potřeby k dispozici, avšak pokud navštíví stránku, kde bude vloženo `<img src='http://www.napadenaaplikace.cz/admin/clanek/smazat/35'>`, tak bude smazán článek s ID=35, avšak přihlášený uživatel o tomto vůbec nebude vědět a tento proces proběhne na pozadí. [26]

V mé aplikaci tomuto problému předcházím využitím tzv. Anti Forgery Tokenu. Jedná se o mechanismus, který slouží jako ochrana proti neoprávněné práci s formulářem. Konkrétně znemožňuje odeslání formuláře, pokud se o to pokouší nekompetentní uživatel. Ošetření jsem provedl tak, že jsem ke každé metodě na `controllers`, která v aplikaci pracuje s daty, přidal direktivu `[ValidateAntiForgeryTokenToken]`. Zároveň jsem do formuláře na view vložil `@Html.AntiForgeryToken()`. V praxi potom celý proces funguje tak, že do zobrazeného formuláře je vložen skrytý vygenerovaný jedinečný řetězec. Při odeslání formuláře je spuštěna příslušná metoda, která díky direktivě `[ValidateAntiForgeryTokenToken]` ověří, zda se

shoduje požadovaný řetězec Anti Forgery Tokenu s tím, který reálně přišel z formuláře. Pokud jsou shodné, tak byla metoda z formuláře volána korektně. V opačném případě systém vrátí výjimku `System.Web.Mvc.AntiForgeryException` a požadovaná akce není provedena.

## 6.6 Nasazení do ostrého prostředí a budoucnost projektu

V době bezprostředně před odevzdání této diplomové práce byla dořešena otázka registrace domény a webhostingu pro spuštění ostré verze systému. Ta bude v budoucnu dostupná z adresy <http://bezbarierzlin.cz/>. Aktuálně běží aplikace pouze v testovacím prostředí na <http://rdsoft.aspone.cz>. Po vyřešení administrativních záležitostí bude následovat domluva s pracovníky magistrátu, zda budou chtít nasadit do ostrého provozu aktuální verzi aplikace, případně jestli si zvolí možnost počkat na zpracování jejich nově vzniklých požadavků na systém. Ať už se rozhodnou jakkoli, aplikaci mám v plánu po nasazení do ostrého provozu průběžně udržovat a v případě potřeby i nadále rozšiřovat o nově požadovanou funkcionalitu.

## ZÁVĚR

Cílem mé diplomové práce bylo vytvoření webového portálu, který bude k dispozici především návštěvníkům s tělesným handicapem. Těm má pomoci získat cenné informace ohledně přístupnosti důležitých míst ve Zlíně a okolí. Po úvodním zmapování požadavků na systém jsem pokračoval rešerší existujících řešení s podobným zaměřením - jak pomoci tělesně postiženým lidem. Za tímto účelem byla vytvořena i publikace o bezbariérovosti, kterou vydal Magistrát města Zlín, a ze které jsem čerpal potřebná data pro svou aplikaci. Následoval výběr vhodných technologií, se zaměřením na možné budoucí zabezpečení systému. Všechny tyto uvedené činnosti jsou zpracovány v teoretické části mé práce.

V praktické části jsem konkrétně specifikoval požadavky, které jsou na systém kladeny. Na základě těchto požadavků a dříve zjištěných skutečností jsem vytvořil návrh aplikace. Poslední a časově nejnáročnější částí byla samotná implementace webové aplikace, která proběhla v programovacím jazyce C# s využitím .NET Frameworku a technologií jako ASP.NET MVC 5, Microsoft SQL Server, Entity Framework, Bootstrap, Team Foundation Server nebo Gogole Maps. Klíčovou součástí aplikace je administrátorské rozhraní, které poskytuje možnost správy důležitých informací. Jsou to údaje o objektech a definovaných štítcích, dále o typech přístupností a kontaktů, jenž jsou v systému evidovány. V neposlední řadě je zde k dispozici správa uživatelských účtů, včetně rolí, které určují práva jednotlivých uživatelů v aplikaci.

Jelikož jsem před započatím této práce neměl s použitím řady uvedených technologií téměř žádné zkušenosti, získal jsem při vytváření této webové aplikace velké množství praktických znalostí. Zjistil jsem řadu důležitých informací ohledně fungování a implementace návrhového vzoru Model-View-Controller v prostředí webu s využitím technologie ASP.NET, objektově relačního mapování dat pomocí Entity Frameworku, verzování zdrojových kódů v Team Foundation Serveru či o fungování a možnostech tvorby responzivního designu webových aplikací. Rovněž jsem se při vývoji setkal s řadou problémů, které jsem byl nucen vyřešit. Jednalo se například o rozdílnost chování JavaScriptového kódu pro načítání dat o objektech, které jsem musel upravit pro prohlížeč Internet Explorer, ve kterém nefungoval správně. Dále jsem se z počátku potýkal s problémy při snaze korektně plnit a následně ukládat vazby mezi jednotlivými entitami systému. Správně fungující uložení, editaci a mazání propojených entit v administrátorské části aplikace jsem do finálního stavu ladil a upravoval nesčetněkrát. Dalším úskalím bylo vytvoření vhodného responzivního designu aplikace tak, aby funkcionální její verze pro mobilní zařízení s malými displeji nebyla nijak ořezána a poskytovala všechny možnosti jako verze určena pro velké monitory klasických osobních počítačů.

V současnosti je systém nasazen pouze na testovacím prostředí, které je dostupné z <http://rdsoft.aspone.cz>, a obsahuje veškerou požadovanou funkcionalitu. V době bezprostředně před odevzdáním této práce byla zaregistrována doména a zajištěn web-hosting pro ostrou verzi aplikace, která tak bude v blízké budoucnosti dostupná z adresy <http://bezbarierzlin.cz/>. S vytvořenou webovou aplikací byli při jejím předvedení pracovníci magistrátu spokojeni. Zároveň z jejich strany při testování a zkoušení vznikají nové a nové podněty ke zlepšení a rozšíření možností systému, které postupně zapracovávám. Aktuálně mám v plánu přidání nové verze webu, která bude šetřit návštěvníkům data tím, že nebude automaticky zobrazovat mapové podklady, ale pouze textové informace o objektech. Odtud potom bude umožněn přechod na zobrazení mapy. Celkový vývoj systému bude probíhat nadále i po odevzdání této diplomové práce.

Přidanou hodnotou projektu je zajisté fakt, že díky variabilitě a možnostem nastavení v administrátorské části, je možné aplikaci využít nejenom pro zaměření na bezbariérovost. Další možností je vytvoření nové instance aplikace a nastavení systému například pro přehled restaurací a kaváren, či dětských hřišť ve Zlíně a okolí.

## SEZNAM POUŽITÉ LITERATURY

- [1] ODBOR SOCIÁLNÍCH VĚCÍ MAGISTRÁTU MĚSTA ZLÍNA. *Vybrané instituce ve Zlíně, Fryštáku, Lípě, Lukově, Sazovicích a Želechovicích nad Dřevnicí z pohledu (bez)bariérovosti*. In: Komunitní plánování sociálních služeb ve Zlíně [online]. Zlín: Centrum služeb postiženým Zlín, o. p. s., 2015 [cit. 2016-03-23]. Dostupné z: <http://www.zlin.eu/komunitni-planovani-socialnich-sluzeb-ve-zline-cl-885.html?sekce=clanky/dokumenty&slozka=2855&soubor=publikace-vybrane-instituce-ve-zline-frystaku-lipe-lukove-sazovicich-a-zelechovicich-nad-drevnici-z-pohledu-bez-barierovosti.pdf>
- [2] HAUSER, Marianne. *HTML a CSS: velká kniha řešení*. Vyd. 1. Brno: Computer Press, 2006. ISBN 80-251-1117-2.
- [3] PROCHÁZKA, David. *První kroky s internetem*. 3., aktualiz. vyd. Praha: Grada, 2010. Snadno a rychle (Grada). ISBN 978-80-247-3255-8.
- [4] OSEK, Jiří. *HTML: tvorba dokonalých www stránek: podrobný průvodce*. Vyd. 1. Praha: Grada, 1998. Průvodce (Grada). ISBN 80-7169-608-0.
- [5] *One Billion Internet Users*. In: Internet World Stats News [online]. 2006 [cit. 2016-04-05]. Dostupné z: <http://www.internetworldstats.com/pr/edi014.htm>
- [6] *Internet used by 3.2 billion people in 2015*. In: BBC News [online]. 2015 [cit. 2016-04-05]. Dostupné z: <http://www.bbc.com/news/technology-32884867>
- [7] PÍSEK, Slavoj. *HTML: začínáme programovat*. 3., aktualiz. vyd. 1. vyd. Praha: Grada, 2010. Průvodce (Grada). ISBN 978-80-247-3117-9.
- [8] ŠŤASTNÁ, Barbora. *CERN - místo, kde se narodil Web* [online]. 2003 [cit. 2016-04-06]. Dostupné z: <http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xstastna.htm>
- [9] CASTRO, Elizabeth a Bruce HYSLOP. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. 1. vyd. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.
- [10] *Browser Statistics*. W3schools.com [online]. [cit. 2016-04-07]. Dostupné z: [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp)
- [11] DRŽKA, Martin a Eva PROVAZNÍKOVÁ. *10 příkázání špičkového responzivního webdesignu*. In: EBrána[online]. 2015 [cit. 2016-04-15]. Dostupné z: <http://ebrana.cz/magazin/10-prikazani-spickoveho-responzivniho-webdesignu>

- [12] FORGÁČ, Ján. *Mobilní nebo responzivní web?* In: ArtWeby [online]. 2013 [cit. 2016-04-15]. Dostupné z: <http://www.artweby.cz/blog/mobilni-stranky-nebo-responzivni-web>
- [13] HANSELMAN, SCOTT. *ASP.NET 5 is dead - Introducing ASP.NET Core 1.0 and .NET Core 1.0*[online]. 2016 [cit. 2016-04-17]. Dostupné z: <http://www.hanselman.com/blog/ASPNET5IsDeadIntroducingASPNETCore10AndNETCore10.aspx>
- [14] ČÁPKA, David. *MVC architektura*. In: ITnetwork.cz [online]. 2013 [cit. 2016-04-17]. Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor/>
- [15] *Microsoft SQL Server Version List* [online]. [cit. 2016-04-17]. Dostupné z: <http://sqlserverbuilds.blogspot.cz/>
- [16] JECHA, Tomáš. *Seznámení a instalace Microsoft SQL Serveru*. In: DotNETportal.cz [online]. 2009 [cit. 2016-04-17]. Dostupné z: <http://www.dotnetportal.cz/clanek/140/Seznameni-a-instalace-Microsoft-SQL-Serveru>
- [17] HORVÁTH, Tomáš. *Teoretický úvod do relačních databází*. In: Programujte.com [online]. 2007 [cit. 2016-04-17]. Dostupné z: <http://programujte.com/clanek/2007110801-teoreticky-uvod-do-relacnich-databazi/>
- [18] MICHÁLEK, Martin. *K čemu je dobrý Bootstrap a frontend frameworky?* In: Zdroják.cz [online]. 2013 [cit. 2016-04-23]. Dostupné z: <https://www.zdrojak.cz/clanky/k-cemu-je-dobry-bootstrap-frontend-frameworky/>
- [19] WAGNER, Janet. *Top 10 Mapping APIs: Google Maps, Microsoft Bing Maps and MapQuest*. In: ProgrammableWeb [online]. 2015 [cit. 2016-05-06]. Dostupné z: <http://www.programmableweb.com/news/top-10-mapping-apis-google-maps-microsoft-bing-maps-and-mapquest/analysis/2015/02/23>
- [20] LAHVIČKA, Jiří. *PHP - základní informace*. In: Interval.cz [online]. [cit. 2016-05-07]. Dostupné z: <https://www.interval.cz/clanky/php-zakladni-informace/>
- [21] *NHibernate nebo Entity Framework?* In: Profinit.eu [online]. 2014 [cit. 2016-05-08]. Dostupné z: <https://www.profinit.eu/blog/nhibernate-nebo-entity-framework-dil-c-2/>

- 
- [22] BÍLEK, Petr. *Verzování*. In: Sallyx.org [online]. 2014 [cit. 2016-05-08]. Dostupné z: <http://www.sallyx.org/sally/c/linux/cvs>
- [23] JUŘEK, Michael. *Integrované systémy pro správu kódu*. In: Zdroják.cz [online]. 2014 [cit. 2016-05-08]. Dostupné z: <https://www.zdrojak.cz/clanky/integrované-systémy-pro-spravu-kódu/>
- [24] KOPP, Ondřej. *Jak zabezpečit aplikaci v ASP.NET*. In: Interval.cz [online]. [cit. 2016-05-08]. Dostupné z: <https://www.interval.cz/clanky/jak-zabezpečit-aplikaci-v-aspnet/>
- [25] FERSCHMANN, Petr. *Bezpečnost na webu – přehled útoků na webové aplikace*. In: Zdroják.cz [online]. [cit. 2016-05-08]. Dostupné z: <https://www.zdrojak.cz/clanky/prehled-utoku-na-webove-aplikace/>
- [26] VRÁNA, Jakub. *Cross-Site Request Forgery*. In: PHP triky [online]. [cit. 2016-05-10]. Dostupné z: <http://php.vrana.cz/cross-site-request-forgery.php>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

.NET	Framework společnosti Microsoft určený pro tvorbu webových aplikací a služeb.
ADO.NET	ActiveX Data Objects for .NET - Technologie, která se používá v prostředí .NET pro čtení a modifikaci dat uložených v relačních databázích, ale i nerelačních datových zdrojích.
ASP	Active Server Pages - Skriptovací platforma společnosti Microsoft, která slouží k dynamickému zpracování webových stránek na straně serveru.
ASP.NET	Nástupce technologie ASP, která je součástí .NET Frameworku.
ASP.NET MVC	Framework založený na spojení technologii ASP.NET a třívrstvé architektury MVC.
CSS	Cascading Style Sheets - Jazyk pro propis stylů zobrazení HTML elementů na webových stránkách.
EF	Entity Framework - Objektově relační mapovací systém (ORM) společnosti Microsoft, který je založen na technologii ADO.NET.
HTML	HyperText Markup Language - Značkovací jazyk určený pro vytváření webových stránek s využitím hypertextových odkazů.
HTTP	Hypertext Transfer Protocol - Internetový protokol, který slouží k výměně hypertextových dokumentů ve formátu HTML.
JS	JavaScript - Multiplatformní, objektově orientovaný skriptovací jazyk, který se používá ve spojení s jazyky HTML a CSS při tvorbě webových stránek.
MVC	Model-View-Controller - Označení třívrstvé architektury, případně zkratka návrhové vzoru.
ORM	Object-relational mapping - V češtině překládáno jako Objektově relační mapování, je programovací technika, jenž zajišťuje automatickou konverzi dat mezi objektově orientovaným programovacím jazykem a relační databází.
SQL	Structured Query Language - Standardizovaný dotazovací jazyk, který se používá především v relačních databázích pro práci s daty.
SVN	Subversion - Systém pro správu a verzování zdrojových kódů, který umožňuje bezplatné komerční použití.
TFS	Team Foundation Server - Systém pro verzování zdrojových kódů a správu celého životního cyklu aplikací od společnosti Microsoft.
URL	Uniform Resource Locator - Jedná se o řetězec znaků, který má definovanou strukturu a je určen k určení umístění zdrojů na internetu.

---

W3C	World Wide Web Consortium - Mezinárodní konzorcium, které vyvíjí a vydává webové standardy pro internetovou službu WWW.
WWW	World Wide Web - Internetová služba, která slouží k ukládání, prohlížení a odkazování hypertextových dokumentů.
WYSIWYG	What you see is what you get - V češtině volně přeloženo jako "co vidíš, to dostaneš". Jedná se o označení způsobu editace dokumentů, kdy je uživateli zobrazena verze vzhledově totožná s výslednou verzí dokumentu.

## SEZNAM OBRÁZKŮ

Obr. 2.1	Mobilní (vlevo) a klasická verze webových stránek Českých drah (vpravo)	17
Obr. 2.2	Příklad responzivního webu pondios.cz . . . . .	18
Obr. 2.3	Schéma komunikace klient-server . . . . .	20
Obr. 2.4	Schéma funkce architektury MVC[14] . . . . .	21
Obr. 3.1	Webová stránka Vozejkmap.cz . . . . .	27
Obr. 3.2	Webová stránka Mapybezbarier.cz . . . . .	28
Obr. 3.3	Webová stránka Wheelmap.org . . . . .	29
Obr. 5.1	Publikace Vybrané instituce ve Zlíně, Fryštáku, Lípě, Lukově, Sazovicích a Želechovicích nad Dřevnicí z pohledu (bez)bariérovosti . . .	35
Obr. 6.1	Struktura projektu ve webovém rozhraní TFS . . . . .	38
Obr. 6.2	Detail changesetu ve webovém rozhraní TFS . . . . .	38
Obr. 6.3	Grafické rozvržení hlavní stránky pro PC . . . . .	39
Obr. 6.4	Použití sandwich menu u mobilního zobrazení stránek . . . . .	40
Obr. 6.5	Administrátorská sekce zobrazena v PC . . . . .	40
Obr. 6.6	Administrátorská sekce zobrazena v tabletu . . . . .	41
Obr. 6.7	Administrátorská sekce zobrazena v mobilním telefonu . . . . .	41
Obr. 6.8	Defaultní styl zobrazení mapy . . . . .	42
Obr. 6.9	Satelitní zobrazení mapy s detailem objektu . . . . .	42
Obr. 6.10	Zdrojový kód pro vložení mapy do stránky . . . . .	43
Obr. 6.11	Mapa bez shlukování značek (vlevo) a mapa s využitím shlukování (vpravo) . . . . .	43
Obr. 6.12	Definice třídy Objekt . . . . .	44
Obr. 6.13	Příklad vazby 1:1 . . . . .	45
Obr. 6.14	Příklad vazby 1:N . . . . .	45
Obr. 6.15	Příklad vazby M:N . . . . .	46
Obr. 6.16	Seznam migrací . . . . .	46
Obr. 6.17	Diagram důležitých databázových tabulek systému . . . . .	47
Obr. 6.18	Správa objektů – úvodní stránka . . . . .	49
Obr. 6.19	Formulář pro vytvoření nového objektu . . . . .	51
Obr. 6.20	Formulář pro vytvoření nové přístupnosti . . . . .	52
Obr. 6.21	Panel pro filtrování podle štítků . . . . .	53
Obr. 6.22	Stránka zobrazující informaci, že uživatel nemá k požadovanému obsahu povolen přístup. . . . .	54

**SEZNAM TABULEK**

Tab. 5.1	Legenda kategorií přístupností . . . . .	34
Tab. 6.1	Přehled přístupu k administrátorským sekcím podle uživatelských rolí	48

## SEZNAM PŘÍLOH

- P I. Přehled databázových tabulek
- P II. Příložené CD

## PŘÍLOHA P I. PŘEHLED DATABÁZOVÝCH TABULEK

<b>Adresa</b>			
<i>Název</i>	<i>Datový typ</i>	<i>Povinnost</i>	<i>Poznámka</i>
ID	int	not null	Identifikátor, primární klíč
Ulice	nvarchar	null	
Cislo_orientacni	nvarchar	null	
Cislo_popisne	nvarchar	null	
Mesto	nvarchar	null	
PSC	nvarchar	null	
Longitude	decimal(12,6)	null	Zeměpisná délka
Latitude	decimal(12,6)	null	Zeměpisná šířka

<b>Kontakt</b>			
<i>Název</i>	<i>Datový typ</i>	<i>Povinnost</i>	<i>Poznámka</i>
ID	int	not null	Identifikátor, primární klíč
ObjektID	int	not null	Cizí klíč - tabulka Objekt
KontaktTypID	int	not null	Cizí klíč - tabulka KontaktTyp
Hodnota	nvarchar	null	

<b>KontaktTyp</b>			
<i>Název</i>	<i>Datový typ</i>	<i>Povinnost</i>	<i>Poznámka</i>
ID	int	not null	Identifikátor, primární klíč
Nazev	nvarchar	null	

<b>Objekt</b>			
<i>Název</i>	<i>Datový typ</i>	<i>Povinnost</i>	<i>Poznámka</i>
ID	int	not null	Identifikátor, primární klíč
Nazev	nvarchar	null	
Poznamka	nvarchar	null	
AdresaID	int	not null	Cizí klíč - tabulka Adresa
Skryto	bit	not null	

<b>ObjektPristupnost</b>			
<i>Název</i>	<i>Datový typ</i>	<i>Povinnost</i>	<i>Poznámka</i>
ObjektID	int	not null	Cizí klíč - tabulka Objekt
PristupnostID	int	not null	Cizí klíč - tabulka Pristup

<b>ObjektStitek</b>			
<i>Název</i>	<i>Datový typ</i>	<i>Povinnost</i>	<i>Poznámka</i>
ObjektID	int	not null	Cizí klíč - tabulka Objekt
StitekID	int	not null	Cizí klíč - tabulka Stitek

<b>Oprojektu</b>			
<i>Název</i>	<i>Datový typ</i>	<i>Povinnost</i>	<i>Poznámka</i>
ID	int	not null	Identifikátor, primární klíč
htmlText	nvarchar	null	

<b>Pristup</b>			
<i>Název</i>	<i>Datový typ</i>	<i>Povinnost</i>	<i>Poznámka</i>
ID	int	not null	Identifikátor, primární klíč
Nazev	nvarchar	null	
Poznamka	nvarchar	null	
ImgID	int	not null	Atributy pro uložení informací k ikonám přístupností
ImgSize	int	not null	
ImgName	nvarchar	null	
ImgData	varbinary	null	

<b>Stitek</b>			
<i>Název</i>	<i>Datový typ</i>	<i>Povinnost</i>	<i>Poznámka</i>
ID	int	not null	Identifikátor, primární klíč
Nazev	nvarchar	null	
Poznamka	nvarchar	null	

## PŘÍLOHA P II. PŘILOŽENÉ CD

Obsah příloženého CD:

- fulltext.pdf - technická zpráva ve formátu PDF
- latex - adresář zdrojových souborů technické zprávy ve formátu jazyka  $\text{\LaTeX}$
- dipl - adresář zdrojových souborů implementované aplikace