

# **Webová prezentace města, obce – redakční systém**

**Bakalářská práce**

**Vedoucí práce:**

**Ing. Jiří Třináctý**

**Jan Vodák**

**Brno 2015**



## **Poděkování**

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Jiřímu Třináctému, za poskytnutí odborných a velmi cenných rad, věcné připomínky, ochotu a vstřícný přístup během zpracování této práce.

Velké poděkování náleží celé mé rodině za podporu a poskytnutou možnost studia na vysoké škole.

Na závěr patří poděkování všem, kteří mě podporovali a pomáhali mi během studia a psaní této práce včetně mé přítelkyně, která mi byla neustále oporou a motivací.



### **Čestné prohlášení**

Prohlašuji, že jsem tuto práci: **Webová prezentace města, obce - redakční systém** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 19. prosince 2014

---



## **Abstract**

Vodák, J. Web presentation of the city – content management system. Bachelor thesis. Brno: Mendel University, 2015

The bachelor thesis deals with analysis of web presentation and content management system of Vnorovy village, followed by designing and implementing some new improvements. If web site doesn't have any content management system, it must be designed and implemented new by using scripting languages and databases.

## **Keywords**

Content management system, analysis, application, administration, framework, database system, data, and information

## **Abstrakt**

Vodák, J. Webová prezentace města, obec – redakční systém. Bakalářská práce. Brno: Mendlova univerzita v Brně, 2015

Práce se zabývá analýzou webové prezentace a redakčního systému obce Vnorovy a následným návrhnutím a implementováním zlepšení. Pokud obec redakční systém nemá, navrhne se a poté implementuje nový za pomoci skriptovacích jazyků a databáze.

## **Klíčová slova**

Redakční systém, analýza, aplikace, administrativa, framework, databázový systém, data, informace





# Obsah

<b>1</b>	<b>Úvod</b>	<b>13</b>
<b>2</b>	<b>Cíle a metodika práce</b>	<b>15</b>
<b>3</b>	<b>Literární rešerše</b>	<b>16</b>
3.1	Seznámení s redakčním systémem a jeho historií .....	16
3.2	Dělení a funkce redakčního systému .....	16
3.3	Framework .....	17
3.4	Bezpečnost.....	17
<b>4</b>	<b>Principy zveřejňování informací</b>	<b>18</b>
4.1	Zákon č. 106/1999 sb. ....	18
4.1.1	Žadatel .....	18
4.1.2	Informace .....	18
4.1.3	Zveřejňování informací.....	18
4.2	Co to pro obecní web znamená.....	19
<b>5</b>	<b>Analýza webové stránky obce</b>	<b>20</b>
5.1	Analýza .....	20
5.2	Vyhláška č 64/2008 Sb. o přístupnosti.....	22
<b>6</b>	<b>Návrh webové aplikace</b>	<b>23</b>
6.1	Uživatelé systému .....	23
6.2	Vlastní MVC redakční systém v PHP .....	24
6.2.1	Model .....	25
6.2.2	Pohled .....	26
6.2.3	Kontroler .....	26
6.2.4	Životní cyklus stránky .....	26
6.3	Princip přesměrování v praxi.....	27
6.4	Zabezpečení.....	28
6.4.1	XSS neboli tzv. cross-site scripting.....	28
6.4.2	SQL injection .....	29

---

6.4.3	Mass assignment.....	30
6.4.4	Výjimka uživatele .....	30
6.5	Databáze.....	30
6.5.1	Práce s relační databází jako s objektem .....	30
6.5.2	Návrh databáze .....	31
<b>7</b>	<b>Správa souborů</b>	<b>34</b>
7.1	Životní cyklus informace .....	34
7.2	Výsledný redakční systém.....	35
7.2.1	Správa souborů úřední desky.....	35
7.2.2	Mechanismus zpráv.....	37
<b>8</b>	<b>Diskuze a závěr</b>	<b>38</b>
8.1	Diskuze .....	38
8.2	Závěr.....	38
<b>9</b>	<b>Přehled informačních zdrojů</b>	<b>40</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>43</b>
<b>B</b>	<b>CD příloha</b>	<b>44</b>

## Seznam obrázků

Obr. 1	Obrázek 1 - Životní cyklus stránky webu Zdroj: ITnetwork, 2015.....	27
Obr. 2	Obrázek 2 - Příklad SQL injection Zdroj: 9lessons.info .....	29
Obr. 3	Obrázek 3 - ER diagram databáze .....	32
Obr. 4	Obrázek 4 - Stavový diagram zveřejněné informace.....	34
Obr. 5	Obrázek 5 - Editor pro vkládání a upravování informací na úřední desku.....	35
Obr. 6	Obrázek 6 - Sekce kompletní správy příspěvků úřední desky.....	36
Obr. 7	Obrázek 7 - Mechanismus zpráv .....	37

## Seznam tabulek

Tab. 1	Tabulka 1 - Příklady rolí a jejich oprávnění.....	24
--------	---	----

# 1 Úvod

V dnešní době, tedy v době počítačů, internetu a nejrůznějších mobilních vymožeností, se každá organizace, sdružení, skupina či celek všech velikostí snaží dostat do podvědomí veřejnosti a upoutat její pozornost, například z důvodu propagace, či přilákání potenciálních zákazníků nebo návštěvníků. Výjimkou nejsou ani města a obce. K tomuto budou využívat internetové stránky a nejnovější vymoženosti jako například aplikace do mobilních telefonů, prostřednictvím kterých umožní lehčí přístup k informacím.

Avšak pokus o upoutání pozornosti není jediná starost, o kterou by se města a obce měly zajímat při poskytování informací. Zajisté po svém městě budou pátrat i obyvatelé, žijící v dané lokalitě. Budou tak činit, aby našli důležité informace, vyhlášky, novinky, různá oznámení nebo v neposlední řadě záznamy, jak vlastně lidé, které si zvolili do vedení, pracují a zda plní své povinnosti správně. To vše by jim měly umožnit dokumenty a informace umístěné na webových stránkách města.

Nyní se dostávám k dané problematice, které se budu v téhle práci věnovat. Bystrého člověka hned na začátku určitě napadne, že přece musí existovat někdo, kdo se o stránky stará a poskytuje informace, které lidé budou na stránkách vyhledávat. A když už se najde pověřená osoba, která informace poskytuje, jak se vlastně informace dostávají na stránky? Odpověď na otázku zní skrze redakční systém. Co to vlastně je redakční systém a podrobné informace o něm popíši v dalších částech práce.

Jak se zdá, v dnešní době se stává naprostým standardem poskytnutí možnosti zviditelnění se všem organizacím ve městě, nehledě na jejich velikost. Co je vlastně město o velikosti třiceti tisíc obyvatel? Na mapě jen špendlíková hlavička. Pokud bude každý ve městě, jak se říká, dělat vše na vlastní pěst, je šance, že si o něm najde informace i někdo jiný, než jen místní obyvatelé o poznání menší. A vlastně ani místní obyvatel nemusí znát všechny možnosti svého bydliště. Navíc všechny informace nemusí být umístěny jen na stránkách města. Je velmi běžné, že na stránkách města najdete jen menší vizitku, která odkazuje na konkrétní stránky dané organizace, což pořád splňuje možnost seskupení pod jednu stránku.

Tento seznam je jen základním výčtem, co by město mělo na internetu poskytovat svým občanům. Dále by mělo být bráno v potaz, že ne všichni lidé budou zdatní počítačová odborníci a ne všichni najdou vše na první pokus. Většina návštěvníků je časově vytížená, proto vyhledávání probíhá v časovém presu a informace budou chtít najít na co nejmenší počet kliknutí, která jim zaberou co nejméně času. Tím jsem zabrousil na základní požadavek, aby stránky byly přehledné a aby nebylo těžké se v nich okamžitě zorientovat, jelikož hlavním účelem je spokojenost návštěvníka. Nikdo se nechce zdlouhavě takzvaně proklikávat a ve výsledku si uvědomit, že někde špatně odbočil a bude se muset znovu vracet. Dělení sekcí a jejich přehlednost tedy hraje klíčovou roli. Stránky by ve finále neměly být jen kusy kódu, ale měly by se zde navíc vyskytovat i dokumenty, které si uživatel může prohlížet či stáhnout.

Webové stránky města, nebo obce, nejsou tedy jen souhrn informací, po kterých návštěvníci pátrají, ale je to komplexní stránka, rozdělena do více částí. Je tedy dobré si na začátku ujasnit, jaké skupiny lidí budou mít ke stránce přístup. Velmi jednoduše se už nyní dá vydedukovat, že existuje běžný uživatel pátrající po informacích a někdo, kdo mu pravidelně informace poskytuje. Samozřejmě, že existuje více členění, na které se později zaměřím.

Tím se však rýsuje další podstatný problém. Musí se nějak oddělit běžná stránka, kde se vyskytují výčty informací od prostředí, ve kterém se pohybuje člověk poskytující informace a kde probíhá celková správa stránky. A musí se objasnit, kolik lidí bude mít přístup do správy stránky. Jaké budou jejich práva při práci s obsahem? Budou je mít všichni stejná? Když se najdou odpovědi na tyto základní otázky, hned se objevují další. Jako například, jak se vlastně dosáhne tohoto rozdělení?

Tyto otázky si pokládají všichni lidé, pohybující se kolem redakčních systémů, a čím více se zamýšlejí, tím více otázek se objevuje. V dnešní době se vyskytuje mnoho možností, jak vše řešit. Existuje celá řada takzvaných open source aplikací, které mohou práci značně usnadnit. Nebo je zde možnost si vytvořit svůj vlastní redakční systém, ve kterém se tvůrce zaměřuje na zdokonalení nedostatků ostatních open source systémů a jako bonus jej poté může nabídnout ostatním. I na nejmenších detailech by mělo záležet a každému by se měla věnovat pozornost, protože i ten nejmenší z nich může rozhodovat, jestli řešení bude úspěšné. Důležité je také to, co se poskytne navíc, přidaná hodnota. Jestli se autor dokáže odpoutat od zaběhlých standardů, nebo zvolí riskantnější cestu a přijde se zcela novým řešením, které může způsobit průlom. Avšak inovovat by se mělo jen v tom případě, že řešení je schváleno všemi dotázanými.

V mé práci se pokusím vytvořit vlastní redakční systém a přizpůsobit ho na potřeby obce.

## 2 Cíle a metodika práce

Cílem práce je navrhnutí zlepšení, nebo vytvoření nového redakčního systému pro město, nebo obci. Tomu přechází analýza zvoleného segmentu IS, jeho návrh a implementace (s použitím skriptovacích jazyků a databáze). Na závěr následuje zhodnocení, zda by aplikace mohla sloužit v praktických podmínkách zvolené obce. Práce se postupně rozčlení do částí, které na sebe postupně navazují.

Prvním krokem je prostudování literatury a nález konkrétního města, nebo obce, která by měla zájem spolupracovat. V tomto případě se jedná o obci Vnorovy na Jižní Moravě. Následuje konzultace se zastupiteli obce a konzultace s pověřenou osobou z praxe, která má na starost již fungující redakční systém a webovou prezentaci. Díky tomu se získá povědomí, jaké musí mít webová prezentace a redakční systém funkce, co od toho lidé, pracující s aplikací, očekávají a jaké jsou požadavky pro práci. V potaz by se mělo brát také to, zda existují omezení určené vyhláškami pro město, nebo obec a jaký to bude mít vliv na naši práci.

V další fázi následuje seznámení se s již existujícím redakčním systémem a webovou prezentací a prostudování vnitřní funkcionality.

Po seznámení se s používaným systémem a webem, přichází na řadu návrh na zlepšení webové aplikace na základě zjištění, vycházejících z provedené analýzy s použitím standartních programovacích nástrojů pro tvorbu informačních a redakčních systémů.

Konečně nastává čas navrhované zlepšení redakčního systému obce implementovat, což by mělo poskytnout zlepšení stavující situace. Vše za pomoci programovacích jazyků (PHP, JavaScript s možností využití knihovny jQuery pokud bude potřeba) včetně HTML, CSS a databázového jazyka SQL.

Na závěr se provede zhodnocení naimplementované aplikace a provedení kompletní dokumentace. Prozkoumá se, jestli by aplikace mohla být použita v praxi, zda poskytuje, nebo by byla schopna poskytnout zlepšení. Bude se prověřovat, zda by práce ekonomicky ovlivnila činnost existujícího systému.

## 3 Literární rešerše

### 3.1 Seznámení s redakčním systémem a jeho historií

V první řadě se vyskytuje otázka, co to vlastně je redakční systém? Redakční systém, neboli content management system (ve zkratce CMS), je software pro správu dokumentů. V dnešní době se nejčastěji využívá pro správu webového obsahu.

Nejdříve by bylo vhodné se zmínit o historii uchovávání dat, neboť základem redakčního systému je databáze, kde jsou data, které se zpracovávají, uchovávána. Zpočátku lidé uchovávali data pomocí papírové kartotéky, které řadily podle různých kritérií. Dalším krokem, nahrazujícím papírové kartotéky, bylo zpracování dat na stroji, které se poprvé objevilo v roce 1890 ve Spojených státech Amerických. Médiem byl děrný štítek. Největší průlom však znamenal rozvoj počítačů v padesátých letech 20. století. S nimi se začínají rozvíjet databáze, které jsou klíčové pro redakční systémy. V šedesátých letech přichází první universální, databázový jazyk COBOL, který je využit pro vznik prvních SŘBD (systém řízení báze dat). V sedmdesátých letech se začínají objevovat hierarchické a relační databáze. Ty zpracovávají data jako tabulky a k nim se vyvíjí jazyk SQL. V devadesátých letech se ve snaze vytlačit relační databáze vyvíjí objektově orientované databáze, založené na objektově orientovaných jazycích. Vše nevychází, jak se očekává, a proto dochází ke kompromisu, vzniká objektově-relační technologie (Závodný, 2010).

Historie redakčních systémů jako takových sahá do začátku osmdesátých let dvacátého století, kdy řada prodejců začala vyvíjet software pro správu dokumentů v tištěné podobě, které dokázaly spravovat nejen zmíněné výtisky, ale také fotografie a grafy. Později se započal vývoj dalšího systému, tentokrát pro správu dokumentů v elektronické podobě, které byly vytvořeny a uloženy na lokálním disku uživatele. Tyto systémy (EDM – electronic document management system) se později rozvinuly do podoby, kdy dokázaly pracovat s jakýmkoliv formátem, který byl uložen v počítačové síti. Obsahovaly nástroje pro spolupráci, bezpečnost a tak dále a začaly umožňovat firmám uchovávat faxy a formuláře, ukládat kopie dokumentů (například ve formátu obrázku) do uložiště pro rychlou, snadnou a bezpečnou možnost vyhledávání. Zatímco mnoho systému ukládá v té době dokumenty v jejich nativním formátu (Excel, Word, PDF), začínají se objevovat internetové redakční systémy, které svůj obsah ukládají ve formě HTML, což umožňuje rychlejší vyhledávání (Závodný, 2010).

### 3.2 Dělení a funkce redakčního systému

Základem pro správu webového obsahu je rozdělit prostředí na administrátorskou a uživatelskou sekci. Administrátorská sekce je taková část webové stránky, která slouží pro správu obsahu. Je skryta všem návštěvníkům a je přístupná jen vybraným jedincům, kteří k ní přistupují. Je vyžadováno prokázat svou totožnost identifikačními údaji. Toto členění od sebe odděluje část, kde administrátor vkládá na-



příklad článku, které jsou následně v uživatelské sekci veřejně dostupné komukoli. S tímto rozdělením souvisí i další vlastnost, a to rozdělení uživatelů dle pravomocí. Je nežádoucí, aby každý uživatel dle libosti mohl upravovat strukturu webu a proto jsou uživatelům rozdány pravomoce dle toho, jakou funkci na webu zastávají. Obvykle existuje administrátor, který může tato oprávnění přidělovat, pak řadový uživatelé, kteří přes systém například spravují svou mailovou schránku, a v neposlední řadě redaktor, který má oprávnění přidávat texty na web a upravovat je. Známým zástupcem redakčního systému je open source systém DRUPAL, který obsluhuje například stránky prezidenta Baraka Obamy (Macháň, 2003).

Mezi nejzákladnější funkce webového redakčního systému patří tvorba a publikace dokumentů v online podobě. To je ale poměrně málo, neboť člověk využívající tento systém potřebuje mít schopnost co nejvíce spravovat svůj systém. Je tedy velmi důležité přizpůsobit systém přímo potřebám uživatele a zařadit do jeho repertoáru funkce, které zahrnou veškerou úpravu, jako například úpravu obsahu již publikovaného dokumentu, mazání obsahu a tak dále, aniž by musel za dodatečnou úpravu vynaložit další prostředky původnímu tvůrci (Závodný, 2010).

### 3.3 Framework

Vyskytuje se zde možnost využít již existujících, velmi rozšířených frameworků, protože tato práce se zabývá redakčním systémem ve webovém prostředí. Avšak aby mohl být framework vůbec posouzen, je potřeba stanovit, co se od něj očekává a co je od něj vyžadováno. Framework, který je běžně používán v praxi, musí být navržen a zkonstruován tak, aby splňoval standard, který je v dnešní době na webu vyžadován (Lošťák, 2010).

Základní vlastností frameworku je usnadnit práci programátorům. Konkrétně to dělá tak, že obsahuje předem předpřipravené sady kódů, určené přímo pro vykonávání základní funkcionality aplikace či některé její části. Aplikace může využívat buďto celý framework a nebo konkrétní knihovnu pro práci jen s určitou skupinou funkcí. Toto komplexní, nebo částečné užití frameworku je i jejich základním členěním (McArthur, 2008). Mezi nejznámější frameworky patří Nette, Zend, Symfony.

### 3.4 Bezpečnost

Podle Nette (Nette Foundation, 2014) by měl být kladen důraz nejen na bezpečnost dat, ale i na zabezpečení celého systému. Zanedbání bezpečnosti může vést k umístění škodlivého kódu v systému, odcizení dat, nebo až k úplnému smazání webu. Příklady typu narušení:

- Cross-site-scripting (XSS)
- SQL injection
- Mass assignment

## 4 Principy zveřejňování informací

### 4.1 Zákon č. 106/1999 sb.

Hned na začátku by bylo ještě vhodné se seznámit se zákonem a základními pojmy v něm, který se týká uveřejňování informací nacházejících se na webu jakékoliv obce.

Jedná se o zákon č. 106/1999 sb. ze dne 11. května 1999, o svobodném přístupu k informacím. Podle ministerstva kultury (Sbírka zákonů České republiky, 1999) tento zákon zapracovává příslušný předpis Evropských společenství a upravuje pravidla pro poskytování informací a dále upravuje podmínky práva svobodného přístupu k těmto informacím. Pojednává o povinnosti poskytovat informace a to státním orgánům, územním samosprávním celkům a jejich orgánům a veřejným institucím.

#### 4.1.1 Žadatel

Žadatelem pro účel tohoto zákona je každá fyzická i právnická osoba, která žádá o informaci. Možností dálkového přístupu pro účel tohoto zákona je přístup k informaci neomezeného okruhu žadatelů pomocí sítě nebo služby elektronických komunikací (Sbírka zákonů České republiky, 1999).

#### 4.1.2 Informace

Informací se pro účely tohoto zákona rozumí jakýkoliv obsah nebo jeho část v jakékoliv podobě, zaznamenaný na jakémkoliv nosiči, zejména obsah písemného záznamu na listině, záznamu uloženého v elektronické podobě nebo záznamu zvukového, obrazového nebo audiovizuálního.

Zveřejněnou informací pro účel tohoto zákona je taková informace, která může být vždy znovu vyhledána a získána, zejména vydaná tiskem nebo na jiném nosiči dat umožňujícím zápis a uchování informace, vystavená na úřední desce, s možností dálkového přístupu nebo umístěná v knihovně poskytující veřejné knihovnické a informační služby podle knihovního zákona (Sbírka zákonů České republiky, 1999).

#### 4.1.3 Zveřejňování informací

Ohledně zveřejňování informací platí, že do 15 dnů od poskytnutí informací na žádost povinný subjekt tyto informace zveřejní způsobem umožňujícím dálkový přístup. O informacích, poskytnutých v jiné než elektronické podobě, nebo mimořádně rozsáhlých elektronicky poskytnutých informací postačí zveřejnit doprovodnou informaci vyjadřující jejich obsah.

Povinné subjekty jsou povinny zveřejňovat informace uvedené v odstavci 1 a 2 též způsobem umožňujícím dálkový přístup. V případě informací uvedených v odstavci 2 písm. a) postačuje ke splnění této povinnosti uvedení odkazu na místo,

kde jsou tyto informace již zveřejněny způsobem umožňujícím dálkový přístup. Strukturu zveřejňovaných informací stanoví prováděcí právní předpis.

Povinné subjekty, které vedou a spravují registry, evidence, seznamy nebo rejstříky obsahující informace, které jsou na základě zvláštního zákona každému přístupné, jsou tyto informace povinny zveřejňovat v přehledné formě způsobem umožňujícím i dálkový přístup. Na tyto subjekty se pro tento účel nevztahuje povinnost zamezit sdružování informací podle zvláštního právního předpisu.

Povinnost zveřejnit informace podle odstavců 4 a 5 splní povinný subjekt tím, že je způsobem umožňujícím dálkový přístup bez zbytečného odkladu zpřístupní správci portálu veřejné správy nebo mu je předá. Formu a datový formát zpřístupňovaných a předávaných informací stanoví prováděcí právní předpis (Sbírka zákonů České republiky, 1999).

## 4.2 Co to pro obecní web znamená

K zákonu č. 106/1999 se vztahuje vyhláška 442/2006 o povinném zveřejňování informací, která stanoví strukturu informací zveřejňovaných o povinném subjektu umožňující dálkový přístup dle § 5 odst. 1 a 2 zákona č. 106/1999 Sb. Povinná struktura zveřejňovaných informací podle vyhlášky 442/2006:

1. Název
2. Důvod a způsob založení
3. Organizační struktura - Seznam organizací
4. Kontaktní spojení (kontaktní poštovní adresa, adresa úřadovny pro osobní návštěvu, úřední hodiny, telefonní čísla, čísla faxu, adresa internetové stránky, adresa e-podatelný, další elektronické adresy)
5. Případné platby lze poukázat
6. IČ
7. DIČ
8. Dokumenty (Seznam hlavních dokumentů, rozpočet)
9. Žádosti o informace
10. Příjem žádostí a další podání
11. Opravné prostředky
12. Formuláře
13. Popisy postupů - návody pro řešení životních situací
14. Předpisy (nejdůležitější používané předpisy, vydané právní předpisy)
15. Úhrady za poskytování informací (sazebník úhrad za poskytování informací, usnesení nadřízeného orgánu o výši úhrad za poskytnutí informací)
16. Licenční smlouvy (Vzory licenčních smluv, Výhradní licence)
17. Výroční zpráva podle zákona č. 106/1999 Sb.

## 5 Analýza webové stránky obce

### 5.1 Analýza

Nyní se konečně dostávám k práci jako takové. Po seznámení se s redakčním systémem a se zákonem, který se vztahuje ke zveřejňování informací pro obecní úřad na svém webu, přišel čas na seznámení se s již zaběhlou webovou prezentací města Vnorovy, který se nachází na adrese <http://vnorovy.cz/>.

Pojem webová prezentace byl napsán záměrně, jelikož na schůzce s místostarostou obce Vnorovy bylo zjištěno, že obec vlastně k internetové stránce nemá žádný redakční systém, kterým by se starali o obsah své stránky. Původní verze webové prezentace jako takové byla vytvořena asi před 15 lety. Poté prošel web úpravou vnitřní struktury v roce 2006. Úpravu provedl člověk, jehož spolupráce neskončila, naopak o web se stará ještě dnes a dá se říct, že působí v roli externího administrátora. Externího z toho důvodu, protože tuto roli vykonává jako vedlejší činnost ve volném čase, nikoli jako hlavní náplň práce.

Jedná se o poměrně zastaralou statickou webovou stránku, na které se během let rozšiřoval spíše jen její obsah, než její funkcionality. Pojem statická webová stránka v tomto slova smyslu znamená, že zde není žádný jednoduchý způsob, kterým by lidé, kteří se okolo webu pohybují, mohli editovat, nebo vkládat nový obsah, aniž by nemuseli při těchto činnostech zasahovat do kódu stránky. Před lety, kdy web prošel základní úpravou, bylo oznámeno zastupitelstvu obce, že stránka by potřebovala značnější rekonstrukci, než jen změnu struktury obsahu, avšak taková změna by vyžadovala i značnější náklady, které si obec zřejmě nemohla dovolit, a tak web zůstal statický. Vzhled ani detailní popis původního HTML statického kódu zde nebude rozebírán, protože pro práci je důležitý obsah webu a s ním spojená analýza redakčního systému. Proto se zde jen stručně zmíním, jak vlastně momentální web funguje, protože práce se má zabývat redakčním portálem.

Na správě obsahu stránek se podílí kromě již zmíněného externího poradce ještě jeden správce, avšak jeho možnosti jsou dosti omezené kvůli chybějícímu systému. Pokud je potřeba nahrát soubor na internet, tak se vše provádí přes program, kterým se přes FTP spojení lze připojit k uložišti dat, kde je web uložen. Jakmile se soubor nahraje do příslušného adresáře s daty, tak se ručně dopíše odkaz na příslušnou stránku, kam soubor patří a přes kterou si jej můžou návštěvníci stáhnout. Je to nejhorší možný způsob, ale jak bylo prohlášeno, „funguje to“. Pokud je potřeba nahrát cokoli jiného než je běžný text nebo soubor, jako například fotoalbum do fotogalerie, nebo je potřeba vykonat jiný nezvyklý požadavek, musí se kontaktovat externí poradce, který požadavek vykoná, jelikož se na obecním úřadě nenachází nikdo, kdo by se ve stránkách orientoval a měl k nim přístup.

Stránka jednoduchým dojmem nejen působí, ale opravdu jednoduchá je. To ovšem nemusí být na škodu, protože lidé, kteří ji navštíví, zajisté ocení rychlý přístup k hledaným informacím. Je velmi lehké se na stránce zorientovat.

Jak již bylo zmíněno, stránka je zcela statická jen s minimem PHP kódu, což znamená, že kdokoli může vidět zdrojový kód stránky, žádná funkcionality a

vnitřní struktura není skryta. Soubory tvořící „kostru“ a obsah internetové stránky jsou jednoduše rozčleněny do pár podadresářů. Hlavní adresář obsahuje složky colorbox, foto, IMG a soubory. Složka colorbox slouží pro zobrazování obrázků na webu, jinak je zcela nepodstatná. Předpokládám, že se jedná o nějakou open source knihovnu. Další adresář je foto, kam se umísťují fotky z akcí obce. V složce IMG jsou uloženy pomocné obrázky a ikonky pro vzhled stránky (logo a obrázky pozadí). Adresář soubory obsahuje všechny HTML soubory tvořící web jako takový. Tyto soubory budou později jen volány na základě toho, kde se návštěvník webu nachází, ale to se popíše níže v textu.

Spouštěčem stránky je index, který obsahuje kromě HTML tagů také zbytečně velké množství jednoduchého PHP kódu, který se stará o automaticky měnící se titulek na jednotlivých stránkách. Titulek se zadává do parametrů URL, odkud je následně předán a vypsán pomocí PHP přepínače.

Pohyb po webu má na starost hlavní menu tlačítek, které obsahuje 9 položek jako informace o obci, obecní úřad a úřední deska. Každá tato část má v sobě další podčásti. Pro tento pohyb na stránce je zde k dispozici levá část stránky, kde jsou další tlačítka – sekundární navigace. Stránka je tedy rozdělena na levou část a pravou část. Levá část obsahuje zmíněné vedlejší menu, pravá část slouží pro výpis obsahu na základě klasických parametrů získaných z URL stránky pomocí metody GET. Vše závisí na tomto obsahu, protože podle něj se zavolá soubor, který se přidá do indexu pomocí metody include. Obsahem souborů jsou většinou tabulky, odstavce textu a obrázky, tedy jen způsob výpisu informací. Vše je ošetřeno pár dalšími podmínkami, které kontrolují, jestli byl předán nějaký parametr, jestli soubor podle parametru existuje, nebo jestli nebyl místo parametru použit vyhledávač na stránce.

Vyhledávač neprochází databázové tabulky, kde jsou uloženy data, jak je u největších webů běžné, ale hledaný výraz hledá jako klíčové slova ve vyhledávači Google, a výsledné odkazy které byly nalezeny, zobrazí.

Kromě povinného obsahu, který musí být obsažen na základě zákona, je zde celá řada dokumentů, ať už to jsou dokumenty farního časopisu, ale také důležité vyhlášky, záznamy ze zasedání rady, nebo veřejný rozpočet. Všechny dokumenty jsou zde bez jakékoliv možnosti editace. A to by se mělo změnit, protože dnešní redakční systémy se řídí heslem „Edituj vše, co vidíš“. Jako plus webové stránky se jeví snad asi jen to, že neobsahuje strohé obecné informace o obci, ale snaží se sjednotit všechny organizace, i ty malé, jako knihovnu, farnost, příspěvkové organizace a různé spolky pod jednu stránku, takže je vše na jednom místě.

Za zmínku ještě stojí poměrně krkolomné řešení jedině, automaticky měnící se části webu – Aktuální informace, což slouží jako hlavní strana. Zde je zobrazena informace o počasí, využívající portál In-pocasi a jejich služeb prostřednictvím JavaScript. Vedle předpovědi počasí je zde informace o momentální otevírací době obecního úřadu a knihovny.

Ovšem způsob provedení není ideální. Nejdříve se testuje, jestli v daný den není svátek, což by bylo v pořádku. Ale problém je v tom, že svátky jsou ručně vypsané v externím textovém souboru formou měsíc-den a tyto hodnoty se musí při-

dat pomocí metody include. Pokud je svátek, nastaví se hodnota den na 0, což značí neděli a vypíše se zavřeno, jinak se vypíše otevírací doba příslušného dne jak pro obecní úřad, tak pro knihovnu. Hodnoty jsou ručně vypsány do souboru a nelze s nimi nijak jednoduše manipulovat. Lidé pracující s webem snad ani netuší, že by mohli v případě potřeby jednoduše upravit otevírací dobu, nebo vlivem neočekávané události ohlásit prostřednictvím webové stránky zavření příslušného institutu.

## 5.2 Vyhláška č 64/2008 Sb. o přístupnosti

Původní podoba stránky, ještě před úpravami provedenými v roce 2006, už předvedena nebyla a jednou z částí úpravy, která se na webu provedla, se týkala podoby a způsobu uveřejňování informací. Konalo se tak na základě další vyhlášky ze dne 7. února 2008 o formě uveřejňování informací souvisejících s výkonem veřejné správy prostřednictvím webových stránek pro osoby se zdravotním postižením (vyhláška o přístupnosti). I na tuhle vyhlášku se musí dát pozor.

Vyhláška stanovuje formu uveřejňování informací, které souvisí s výkonem veřejné správy a jsou uveřejňovány způsobem umožňující dálkový přístup. Tato forma uveřejňování zajistí seznámení s příslušnými informacemi v nezbytném rozsahu i pro osoby se zdravotním postižením (vyhláška č. 64/2008 Sb.).

Obsahem vyhlášky je 32 pravidel, které stanovují podle závaznosti doplňkové informace k obsahu webové stránky. Všechny pravidla zde nebudou zbytečně vypsána, ale jen pro příklad je zde jasně definováno, že obsah webových stránek musí být dostupný a čitelný (každý netextový prvek musí mít textovou alternativu, barvy pozadí a popředí musí být kontrastní atd.), práci s webovou stránkou řídí uživatel, informace musí být srozumitelné a přehledné, zdrojový kód musí být technicky způsobilý a strukturovaný.

Na závěr se zde nachází prohlášení o přístupnosti webových stránek, které říká: „Každá webová stránka musí vždy obsahovat prohlášení o tom, že forma uveřejnění informací je v souladu s touto vyhláškou (prohlášení o přístupnosti) nebo odkaz na toto prohlášení.“, což stránka Vnorov bez problému splňuje.

## 6 Návrh webové aplikace

Při návrhu zlepšení se musí postupovat úplně od začátku, jelikož web žádný redakční systém neobsahuje. Takže, kde začít? Nejvhodnější je určit si, co vlastně bude potřeba zlepšit. Na stránce se nachází velké množství souborů, takže by bylo vhodné navrhnout nějakou správu pro práci se soubory. Již byla zmíněna úvodní stránka, obsahující kromě souborů s novinkami i otevírací dobu obecního úřadu a knihovny, která se dá měnit jedině přepisem textového souboru. Začal bych tedy úvodní stranou a poté bych navrhnul základ pro správu souborů na úřední desce. K tomu je potřeba jednoduchý, objektový MVC redakční systém.

Než se začne něco upravovat, tak by bylo vhodné si stanovit pár základních informací, například kdo k redakčnímu systému má mít přístup, jak jej může využívat a poté si ujasnit, kde uchovávat data. Budu vycházet z analýzy, kde jsem pár uživatelů už jmenoval a kteří k webu mají přístup. Dále by měl být kladen důraz především na zabezpečení dat, rychlost redakčního systému a hlavně příjemné uživatelské prostředí, protože ne všichni uživatelé budou tak zdatní, jak ten kdo jej navrhnul, z čehož vyplývá, že velké riziko pro systém představují kromě lidí z venku sami uživatelé systému. Navíc, běžní uživatelé nebudou vědět, jak funguje vnitřní struktura systému. Z toho vyplývá, že by se při jeho návrhu mělo počítat se všemi možnostmi, které mohou nastat při manipulaci s ním. Běžný uživatel neví, na co si má dát pozor a ani se tím nebude chtít zatěžovat.

Na závěr by nemělo být zapomínáno, že se systémy neustále vyvíjí, technický pokrok jde kupředu. Jak bude plynout čas, vyskytnou se nové technické inovace, které mohou vyžadovat zásahy do systému, nebo i jeho samotní uživatele mohou vyžadovat další zlepšení. Při návrhu a implementaci by se tedy mělo počítat i s nepřetržitým vývojem a zásahy do struktury a tyto zásahy by neměly mít vliv na chod a vnitřní funkcionalitu stránky. Proto se musí obsah, nebo struktura webu rozčlenit do mezi sebou nezávislých bloků.

### 6.1 Uživatelé systému

Jak již bylo zmíněno v úvodu mé práce, je potřeba rozdělit pravomoc pro správu stránky, neboť přístup do redakčního systému by měli mít jen ti jedinci, kteří se o web budou starat. Tím oddělím veřejnou a soukromou sekci stránky.

Abstrahuji se již zavedeného režimu práce se stránkou a navrhnou pár základních, abstraktních rolí, které se potom dají aplikovat přímo na konkrétní osoby v praxi. Pro někoho, kdo se stará o obsah webu, zde vytvořím roli s názvem správce obsahu. Tato role má přístup do téměř všech sekcí a může editovat veškerý obsah. Měla by zde být i role, která může využít veškerou funkcionalitu systému, někdo, kdo má všechna práva a bude moct provádět dozor nad ostatními, případně měnit role. Nazvu ji administrátor.

Z výčtu rolí, mají do chráněné sekce přístup, je tedy patrné, že ne všichni uživatelé mohou vykonávat všechny funkce. Část obsahu by měla být skryta i těm, co mají přístup do systému. Pokud by zastupitelé nebyli spokojeni s prací, kterou od-

vádí správce a pověřili by správou systému někoho jiného, mohl by se cítit dotčen a chtít se mstít třeba tím, že bude úmyslně tvořit chaos v systému. Proto by mu jeho nadřízený měl jeho práva omezit, případně zrušit. Tím vzniká hierarchie uživatelů. Systém by tedy měl poskytovat aspoň základní správu těchto uživatelů, kterou dočím díky RBAC, neboli role based access control. Jedná se o seznam jednotlivých uživatelů a k nim přiřazených rolí. Správa by měla obsahovat možnosti měnit a nastavovat role jednotlivým uživatelům. Toto privilegium by měl mít tedy jen jeden člověk, aby se této pravomoci nezneužívalo.

Tab. 1 Tabulka 1 - Příklady rolí a jejich oprávnění

Název role	Oprávnění
Administrátor	Přístup do všech sekcí Editace všech sekcí Editace uživatelů a jejich rolí
Správce obsahu	Přístup do skoro všech sekcí Editace skoro všech sekcí Nemá přístup do editace RBAC
Editor	Přístup jen do zvolené sekce Editace obsahu v příslušné sekci
Bez oprávnění	Pouze čtení obsah webu Možnost editace jen svého profilu, pokud nějaký má

V tabulce je základní návrh rolí a jejich práv. Role editor slouží pro uživatele, kteří budou mít přístup jen do vybrané sekce, kterou budou mít na starost, jako například hlášení rozhlasu, knihovna, nebo příspěvková organizace.

## 6.2 Vlastní MVC redakční systém v PHP

Byla zmíněna možnost využít hotový framework, který by se postaral o minimalizaci, přehlednost a hlavně bezpečnost kódu aplikace. Získal bych kvalitní funkcionality bez práce. Avšak využití cizího frameworku sebou přináší řadu rizik. Uvedu zde nevýhody využití vlastního a cizího frameworku.

Příklady nevýhod již hotového frameworku:

- Negarantovaný vývoj: každý framework není standartní součástí PHP, ale jedná se o kód třetí strany, což už samo o sobě představuje nebezpečí. Nikdo nezaručí, že se autor nerozhodne přestat s vývojem, jako se to stalo s českým frameworkem PhpRS, nebo se přestane vyvíjet pro nové verze PHP.
- Změny rozhraní: aby se mohl framework používat, musí se dobře naučit jeho rozhraní, které je pro každý jiné
- Naivní pocit bezpečí: S nesprávně použitým frameworkem se může naprogramovat stejně zranitelná aplikace jako bez něj, je to stále jen PHP



- Všem přístupný kód: Už jen to, že se jedná o všem přístupný kód, může znamenat riziko. Jakákoliv chyba je všem známá a útok na takové systémy už potom není žádný problém. Přes opensource aplikace vede přibližně 80% útoků.

Nevýhody vlastního frameworku:

- Špatný návrh: je velmi pravděpodobné, že pokud se s návrhem frameworku někdo setká poprvé, vyskytnou se nějaké chyby, které mohou být fatální pro chod aplikace
- Bezpečnost: musíte se sám postarat o bezpečnost a nastudovat si nové možnosti bezpečnosti

Kdy tedy použít vlastní framework, nebo se přiklonit již k hotovému? Pro ty, kteří programují jeden web za druhým, je ideálním řešením již hotový framework, Avšak pro lidi, kteří delší dobu spravují jeden projekt, může být výhodnější použít vlastní framework z toho důvodu, že nikdo nebude měnit rozhraní a nebude ohrožovat budoucnost vývoje. Z tohoto důvodu jsem se tedy přiklonil k variantě naprogramovat vlastní šablonu.

Navrhnou plně objektový redakční systém, při němž využiji objektově orientované PHP a budu dodržovat MVC architekturu. Zaměřím se na přívětivé, snadno zapamatovatelné URL (žádné dlouhé kusy textu s parametry za otazníky) a pro editaci textů využiji WYSIWYG editor.

MVC architektura byla původně navržena pro desktopové aplikace, avšak nejvíce se uchytila na webu. Na této architektuře je postaven Zend nebo Nette. Základní myšlenkou je vyřešit nevýhody takzvaného špagetového kódu, se kterým se setkal snad každý, kdo se zabývá problematikou webových aplikací. Jedná se kód, ve kterém se společně (v jednu souboru) objevují logické operace (PHP), výpis výstupu (HTML prvky) a databázové dotazy. Kód se špatně udržuje, natož rozšiřuje, velmi těžko se orientujeme v HTML, protože bývá špatně naformátováno. MVC se snaží toto vyřešit tím, že oddělí logiku od výstupu.

Aplikace je rozdělena do souborů tří typů – modely (models), pohledy (views), kontrolery (controllers), zkratka těchto typů tvoří název. Striktní definice pro MVC však neexistuje, proto se každý může setkat s více výklady, avšak důležitý je princip – dělení na tři typy souborů (Čápka, 2014).

### 6.2.1 Model

Třídy typu model obsahují veškerou logiku a vše, co se logiky týče. Mohou to být výpočty, validace, databázové dotazy a tak podobně. Model se však vůbec nezajímá o problematiku výstupu. Jeho funkce spočívá v přijetí parametrů zvenku a vydání dat ven. Pojmem parametry nejsou myšleny URL adresy ani žádné jiné parametry od uživatele. Model neví, odkud požadavky přišly a ani jak budou výstupní data zformátována a vypsána.

Použiji principu manažerů, logika aplikace se tedy rozdělí např. mezi manažery „ManazerUzivatelu“ a tak podobně. Ty obsahují metody k výběru potřebných údajů z databáze nebo k jejich modifikaci, např. metodu k výběru informací o uži-

vateli při vypsání jeho profilu. Jinou metodou bych vybíral otisk hesla z databáze, abych ověřil přihlášení (Čápka, 2014).

### 6.2.2 Pohled

Pohled (View) se stará o zobrazení výstupu uživateli. Nejčastěji se jedná o PHTML šablonu, obsahující HTML stránku a tagy nějakého značkovacího jazyka, který umožňuje do šablony vkládat proměnné, případně provádět iterace (cykly) a podmínky. Pro šablony se často používají speciální značkovací jazyky (např. Smarty).

Pro PHP nemá ale příliš velký smysl používat značkovací jazyk, jelikož je samo značkovacím jazykem a umožňuje takový styl zápisu kódu, aby struktura HTML stránky zůstala zachována. Pohledy lze samozřejmě vkládat do sebe, abych se neopakoval (šablona s rozložením stránky, šablona s menu atd....).

View není jen šablona, ale hlavně zobrazovač výstupu. Obsahuje tedy minimální množství logiky, která je pro výpis nutná (cykly pro výpis). View podobně jako model vůbec neví, odkud mu data přišla, stará se jen o jejich zobrazení uživateli (Čápka, 2014).

### 6.2.3 Kontroler

Kontroler je nyní onen chybějící prvek, který osvětlí funkčnost celého vzoru. Jedná se o jakéhosi prostředníka, se kterým komunikuje uživatel, model i pohled. Drží tedy celý systém pohromadě a propojuje jednotlivé komponenty. Jeho funkci lépe vysvětlí obrázek ukázky životního cyklu stránky. Opět existuje mnoho různých přístupů, nejčastěji má však každá entita jeden kontroler, máme tedy „UzivatelKontroler“, „ClanekKontroler“ a tak podobně.

MVC architektura usnadňuje i myšlení při vývoji projektu. Když se píše logika, patří do modelu, formátování a stylování výstupu se řeší v šabloně, to co uživatel vyžaduje z parametrů, zjišťuje kontroler. Tři různé problémy na třech různých místech, oddělené tak, aby do sebe nezasahovaly a nedělaly vývoj složitější (Čápka, 2014).

### 6.2.4 Životní cyklus stránky

V rychlosti popíši, jak framework pracuje a jaký je životní cyklus zobrazované stránky.

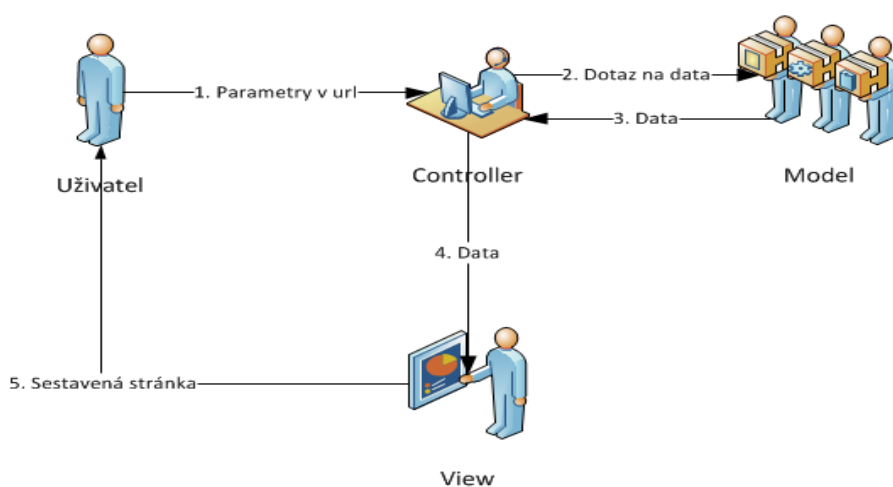
Životní cyklus zahajuje uživatel, který zadá do prohlížeče adresu webu a parametry, kterými sdělí, kterou podstránku si přeje zobrazit. Bude chtít například zobrazit detail uživatele s identifikačním číslem 15:

`http://www.domena.cz/uzivatel/detail/15`

Požadavek jako první zachytí tzv. směrovač. Ten podle parametrů pozná, který kontroler je volán. Jistě existuje mnoho způsobů, jak jméno kontroleru rozpoznat. Pro udržování maximální jednoduchosti se vynechá tzv. manuální směrování a název kontroleru se jednoduše uvede jako první parametr v URL. Zde je tedy za-

volán kontroler „UzivatelKontroler“, jemuž jsou předány 2 parametry "detail" a hodnota „15“.

Daný kontroler podle parametrů pozná, co se po něm chce, tedy že má zobrazit detail uživatele. Zavolá model, který uživatele vyhledá v databázi a vrátí jeho údaje. Dále zavolá další metodu modelu, která např. vypočítá věk uživatele. Tyto údaje si kontroler ukládá do proměnných. Nakonec vypíše pohled. Název pohledu se pozná podle akce, jaká je prováděna. Pohledu jsou předány proměnné s příslušnými daty. Kontroler tedy poslechl uživatele, obstaral podle parametrů dotazu data od modelu a předal je pohledu, který přijme data od kontroleru a vloží je do připravené šablony. Hotová stránka je zobrazena uživateli, který často o tomto procesu ani netuší (Čápka, 2014).



Obr. 1 Obrázek 1 - Životní cyklus stránky webu  
Zdroj: ITnetwork, 2015.

### 6.3 Princip přesměrování v praxi

Princip „hezkých“ URL je docílen pomocí nastavení HTTP serveru Apache prostřednictvím souboru htaccess. Apache má na starost, co se stane, když uživatel zadá příslušnou URL adresu. Vše se děje ještě předtím, než se spustí vůbec nějaké PHP.

Zpracování URL adres nechám plně na PHP, v Apache tuto možnost proto vypnu a všechny dotazy přesměruji na soubor „index.php“, kde si URL sám zpracuji. Apache by totiž jinak bral parametr za lomítkem jako složku a tam hledal podsložku s názvem odpovídajícím dalšímu parametru, pokud by byl zadán a v ní soubor „index“. Zapnu možnost umožňující přesměrování a přidám pravidlo, které až na určité typy souborů v URL přesměruje vždy na spouštěcí soubor „index.php“.

Možnost stahovat soubory bez tohoto přesměrování obstarám pomocí „RewriteCond“. To určí, že se nemá přesměrovat v případě, že soubor nebo složka existuje. Pokud se tedy zadá za lomítkem název souboru s příslušnou koncovkou, bude tento soubor stažen a pokud neexistuje, bude přesměrováno na index. Kdybych

přesměroval úplně vše (bez těch několika podmínek), nemohl by uživatel stahovat žádné soubory a vždy by se mu zobrazil index. Přesměrování je navíc ještě pojištěno výčtem nejdůležitějších přípon, které se nebudou přesměrovávat. Všechny ostatní URL adresy se směřují.

Přesměrování zabezpečí v PHP naprogramovaný směrovač, který je potomek všeobecného kontroleru. Směrovač dostane URL adresu, kterou si zpracuje a na jejím základě zavolá příslušný kontroler. Adresu se zpracuje pomocí metody „parse\_url“ a pomocí dalších pomocných tříd.

V indexu se vytvoří instanci směrovače, zavolá se základní metoda „zpracuj“, která je definována jako abstraktní v obecném kontroleru a každý zděděný kontroler si ji proto implementuje sám. V tomto případě se metoda postará o předané parametry, což je URL adresa. Tu získá pomocí super globálního pole „\$\_SERVER“ pod klíčem „REQUEST\_URI“. V hlavní metodě „zpracuj“ je tedy získán z parametru název kontroleru, který obsahuje název pohledu a ten se uživateli následně zobrazí.

Zobrazení je realizováno v indexu, kde se zavolá další zděděná metoda od předka „vypisPohled“. Tato metoda předá příslušnému pohledu data získané od modelů a další parametry, které se vypíší. Data jsou jedním ze tří atributů, které jsou zděděny kromě ostatních tříd potomci obecného kontroleru. Druhý atribut je název pohledu a poslední je pole tří atributů hlavičky HTML. Kromě principu URL jsem tedy popsal i základná princip propojení (Čápka, 2014).

## 6.4 Zabezpečení

Výhodou, ale zároveň i nevýhodou vytváření vlastního frameworku je, že vše si musím navrhnout sám. Jak již bylo mnohokrát zmiňováno, základem dobrého frameworku je bezpečnost. Pokud by byl využit již existující, nemusel bych se o bezpečnost tolik zajímat a využil bych připravené metody, které by vše zabezpečili za mě. V rychlosti shrnu, jak je redakční systém zabezpečen proti základním bezpečnostním ohrožením.

### 6.4.1 XSS neboli tzv. cross-site scripting

XSS je metoda narušení webových stránek, zneužívající neošetřených výstupů. Útočník pak dokáže do stránky podstrčit svůj vlastní kód a tím může stránku pozměnit nebo dokonce získat citlivé údaje o návštěvnících.

Ochrana proti XSS je docílena tím, že hodnoty jsou těsně před výpisem na stránku předány funkci „htmlspecialchars“, která z HTML udělá obyčejné textové řetězce, které se prohlížečem nespustí, ale jen se vypíší jako běžný text. Takto budou ošetřovány všechny proměnné, než se vypíší.

Avšak vyskytnou se případy, kdy je potřeba vložit neošetřenou proměnnou. Těch je však velmi málo, proto se proces ošetřování provádí automaticky a nutnost vložení neošetřené proměnné se považuje za výjimku.

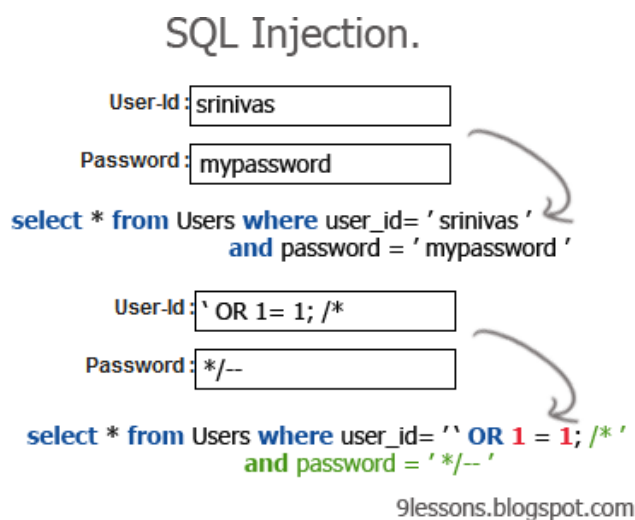
Jelikož se v daném MVC frameworku pracuje téměř vždy s asociativními poli, je vytvořena funkce, která toto pole rekurzivně převede na entity. Jinými slovy za-

volá funkci „htmlspecialchars“ na všechny položky v poli s daty a pokud je v poli vloženo nějaké další pole, udělá to samé s ním a tak dále.

Proměnné předhodím ošetřující funkci těsně před výpisem z databáze, nikoli před jejich uložením. Číním tak proto, že databáze by měla obsahovat co nejvíce surová data, tedy přesně to, co uživatel zadal. Formátování provádí až aplikace na výstupu, avšak nemělo by se zapomínat i na nějaká potřebná ošetření před uložením dat (Čápka, 2014).

### 6.4.2 SQL injection

Myšlenka spočívá v tom, že se do databáze pokusí vložit cizí data prostřednictvím například URL, nebo pomocí formuláře, které pak databáze vykoná.



Obr. 2 Obrázek 2 - Příklad SQL injection  
Zdroj: 9lessons.info

U starých ovladačů se proměnné vkládali do textového řetězce s dotazem, například:

```
mysql_query('SELECT * FROM `uzivatele` WHERE `jmeno` = " . $jmeno . "');
```

Tento dotaz je však potenciálně nebezpečný, protože jsou použity hodnoty v proměnné přímo od uživatele a ten poté může propašovat do databáze škodlivý kód. Dříve se to řešilo pomocí funkce „mysql\_real\_escape\_string“, která se postará o nebezpečné znaky, avšak pro čísla toto řešení nebylo dostatečné.

Proto je využito takzvaných „prepared statements“. Jedná se o metodu vkládání proměnných do dotazů, která řeší problém jinak. Proměnou do dotazu není vložena přímo, ale je oddělena a místo ní je vložen zástupný znak. Takový dotaz se nejprve připraví a poté je spuštěn s polem parametrů, které se dosazují místo zástupných znaků.

Oddělení probíhá pomocí metod „prepare“ a „execute“. Do metody „prepare“ je vložen text pro SQL dotaz se zástupnými znaky. Dotaz se připraví, poté se zavolá metoda „execute“, která připravený dotaz provede s polem parametrů (Čápka, 2014).

### 6.4.3 Mass assignment

Útok spočívá v tom, že pokud se vkládají do databáze hodnoty z formuláře předané pomocí metody „\$\_POST“, vystavujeme se riziku, že by si uživatel mohl navíc do formuláře přidat nějaké pole a ta by se vložila do databázového dotazu. Proto jsou uloženy z „\$\_POST“ do pomocného pole hodnoty klíčů, které odpovídají požadovaným názvům formulářových polí a pomocí PHP funkcí „array\_intersect\_key“ a „array\_flip“ je pole oříznuto jen na tyto klíče. Toto pole bude předáváno metodám pro zápis do databáze.

### 6.4.4 Výjimka uživatele

Nejedná se o klasický útok na webové stránky, ale i klasické výjimka v PHP může být nebezpečná. Nebezpečí spočívá v tom, že jejím obsahem může být heslo k databázi, nebo jiné citlivé údaje. Proto jsem si dovolit se o tomhle nebezpečí zmínit v této kapitole. Z toho vyplývá, že uživateli by měly být zobrazeny jen ty výjimky, které jsou pro něj určené. Bylo by proto vhodné vytvořit si vlastní výjimku, která dědí od klasické výjimky a v kódu je odchyťovat pomocí „try“ a „catch“.

## 6.5 Databáze

Bylo objasněno, z čeho se vytvořený framework skládá a jak funguje, proto se s ním může začít pracovat. Uvedl jsem, že se nejprve zaměřím na správu souborů a úpravu základní stránky webu obce. K tomu bude nezbytné předělat strukturu původní stránky ze statické do vytvořeného frameworku a navíc je potřeba navrhnout si databázi pro ukládání dat. K tomu využiji asi nejrozšířenější databázový systém MySQL s výchozím formátem uložiště InnoDB.

Mezi přednosti systému patří výkon, možnost instalace jak na Linuxu a MS Windows ale i na jiných operačních systémech a v poslední řadě také to, že se jedná o volně šiřitelný software.

Z toho vyplývá, že se nevydám směrem objektových databází, ale směrem relačních databází, protože objektové databáze nenaplnili svůj potenciál. Avšak tím se vyskytuje problém, protože relační databáze neumí ukládat objekty vznikající objektovým programováním. Existuje několik možných řešení. Například vyhnout se objektovému programování, což je v dnešní době zcela nemyslitelné, protože kód by se stal nekvalitním. Je proto udělán kompromis a využit CRUD Wrapper.

### 6.5.1 Práce s relační databází jako s objektem

Přístup přes tzv. wrapper umožňuje s databází pracovat jako s objektem, nicméně komunikace probíhá stále v jejím jazyce SQL. Je tedy míchán objektový a

relační kód. Přístup je jakýmsi kompromisem a vyžaduje filozofii OOP trochu „ohnout“. Výhodou je zachování výkonu a schopností databáze za cenu mírné degradace myšlenek OOP.

Databázové wrappery jsou velmi často vytvořeny jako CRUD, které vykonávají čtyři základní operace nad daty – CREATE, READ, UPDATE, DELETE.

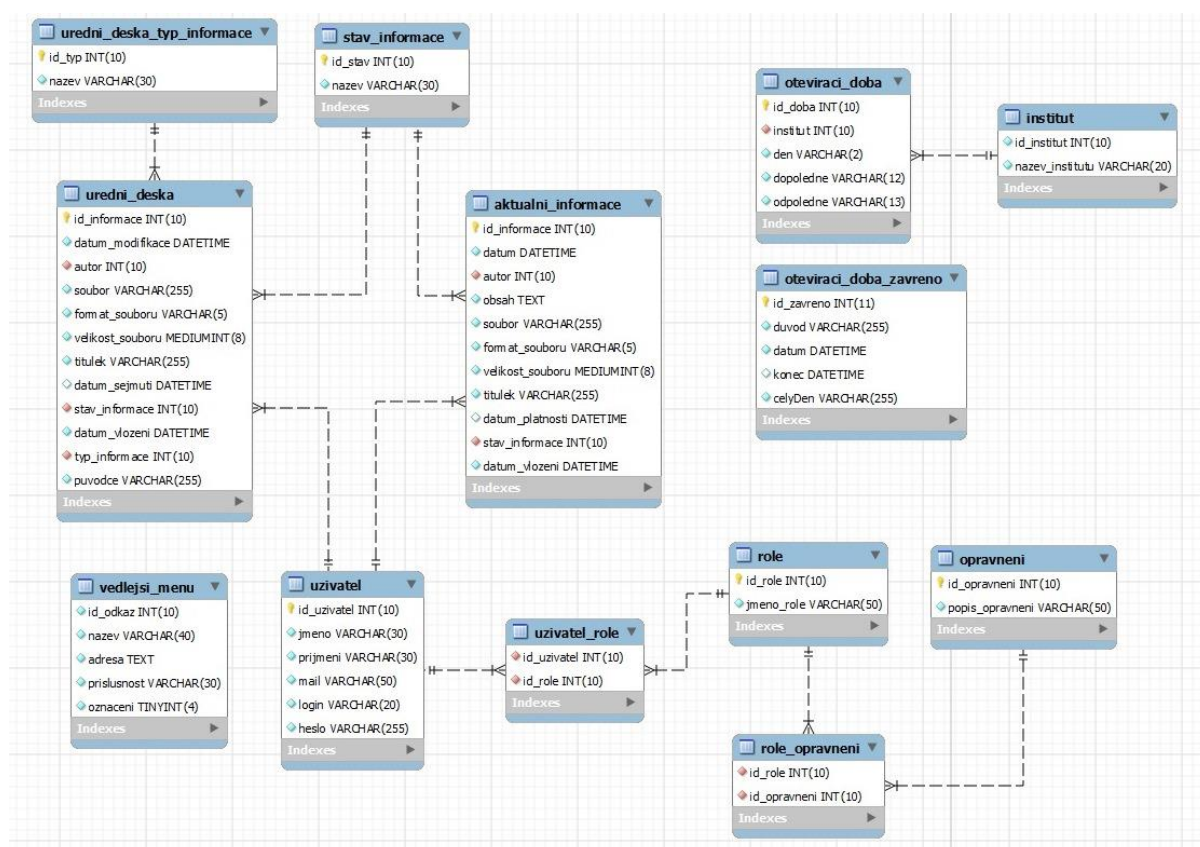
Vytvořím tedy třídu patřící do modelů, které slouží jako zmíněný wrapper a přes kterou probíhá komunikace s databází a pomáhá mi začlenit data do objektového kódu. Jedná se o „obal“ nad něčím složitějším, kde si definuji své, často jednodušší rozhraní. Vše postavím okolo ovladače PDO, který poskytuje přímo PHP a který je jeden z nejlepších. Umožňuje i přepnout na jinou databázi, mapovat entity jako objekty a podobně. Zmíním fakt, že samotné PDO je vlastně také wrapper, čili balím už existující wrapper, čímž si zjednoduším další kód aplikace.

Jak budou dotazy prováděny, bylo popsáno v podkapitole „Zabezpečení“, kde jsem se zabýval SQL Injection. Připravím si zde statické funkce pro vrácení jednoho řádku tabulky, všech řádků tabulky, dotaz na jeden sloupec, dotaz na vrácení počtu ovlivněných řadů a dále dotazy pro vložení nové hodnoty, pro změnu (Update) a nakonec pro vrácení indexu posledního vloženého záznamu. V jiných modelech budou tyto funkce volány a v parametrech se jim předá například SQL dotaz a pole s hodnotami (Čápka, 2014).

### 6.5.2 Návrh databáze

Bylo popsáno, jak se s databází pracuje, ale ještě žádná konkrétní databáze nebyla vytvořena. V textu byl popsán příklad uživatelů systému a zmínka o jejich přístupových právech. Dále že prvotní úpravy by měly směřovat ke správě velkého množství dokumentů a měl by být položen základ pro editaci statické otevírací doby. K tomu všemu bude potřeba tabulky pro uchování potřebných dat. Zde popíši základní kostru tabulek pro tyto úpravy, které uchovávají určité informace. Množství tabulek není konečné a s rostoucí funkcionalitou poroste. Při práci s databází ctím konvenci pojmenování tabulek a parametrů s podtržítka místo mezer.

Jak je patrné z obrázku na další straně, údaje o uživateli budou uchovávány v tabulce „uživatel“. Je potřeba uchovávat základní údaje jako jméno, příjmení, heslo, e-mail a tak podobně. Pro kontrolu, kam má mít uživatel přístup a co v sekci může vykonávat, jsou vytvořeny ještě další tabulky, které to pomůžou zrealizovat. Je použit jednoduchý RBAC princip, který přiřadí uživatelům role, a těm se potom nastaví práva, které budou moci vykonávat. V posledním stádiu se jen zkontroluje, jestli před vykonáním nějaké činnosti má uživatel tyto práva ve své roli. Tabulka role obsahuje kromě id role její název. Tabulka oprávnění obsahuje jen výčet popisů oprávnění. Pro propojení rolí a oprávnění bude potřeba další tabulka („role\_opravneni“), která propojí role s oprávněním pomocí jejich id. Každá role má specifický výčet oprávnění. Role s nejvyšším oprávněním (administrátor) bude mít všechny oprávnění. Jakou roli uživatel v systému zastává, zjistíme z další propojovací tabulky „uživatel\_role“. Ta obsahuje id uživatele a k němu přiřazenou roli pomocí id role.



Obr. 3 Obrázek 3 - ER diagram databáze

Je potřeba uchovávat informace o novinkách, které budou zveřejňovány jako aktuální informace. Kromě titulku, o jakou novinku se jedná a o obsahu sdělení, bude potřeba vědět, kdo informaci vložil a modifikoval, kdy to bylo a do kdy by měla být informace zveřejněna. O životním cyklu zveřejněné informace později. Navíc může být k textu přiložen i soubor. I ten by měl být propojen s informací a měl by být o něm v databázi záznam.

Pro práci s otevíracími hodinami jsou zde další dvě tabulky. Jedna z nich obsahuje otevírací dobu pro obecní úřad a knihovnu, což jsou hlavní dva instituty města v systému. Můžeme uchovávat i jiné informace než provozní dobu institutů, například adresy a proto je zde vytvořena samostatná tabulka institut. Otevírací hodiny kromě hodin obsahují tedy cizí klíč, ke kterému institut se vztahuje. Druhá tabulka pro práci s otevírací dobou je „oteviraci\_doba\_zavreno“, která obsahuje dny, kdy bude zavřeno, ať už z důvodu státního svátku, nebo mimořádné události.

Alfa a omega pro redakční systém města je práce s úřední deskou, která obsahuje všechny důležité zveřejněné dokumenty od formulářů přes výpisy z jednání zastupitelstva a rady obce, až po zveřejněné rozpočty a veřejné zakázky. Z tohoto důvodu se zde vyskytuje samostatná tabulka s titulkem, který říká, o jakou informaci se jedná, datem modifikace a vložení informace. Musí se vědět, kdo informaci vložil a do kdy bude informace zveřejněna jako aktuální. Informace se také musí rozřadit podle typu, jako například zmíněné formuláře, výpisy jednání a rozpočty.



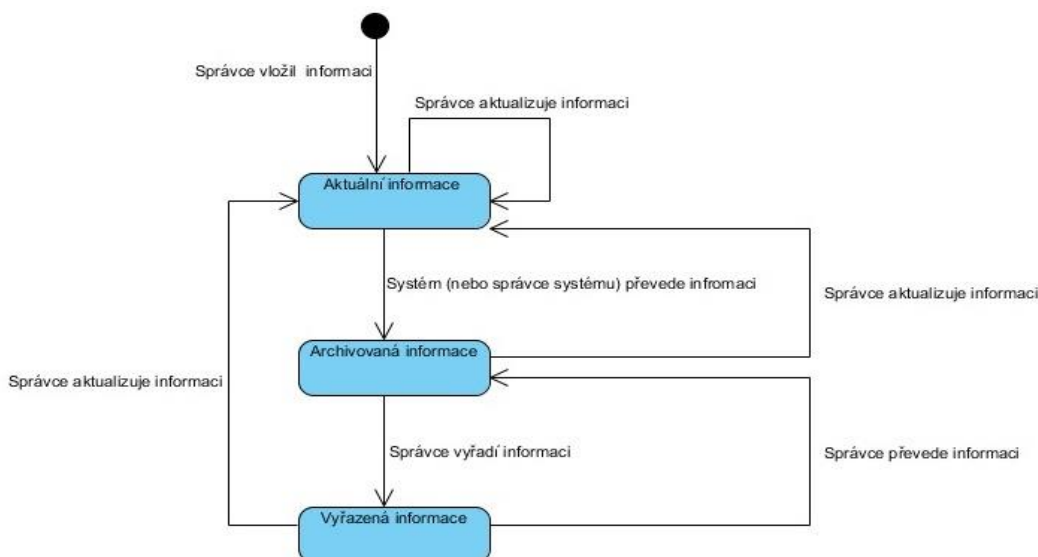
Položky typ a stav se uchovávají v dalších pomocných tabulkách. Dále bude potřeba k informacím úřední desky připojit i informace o souboru, kde bude vše pečlivě popsáno.

Pro lehčí práci s mapou webu a pro případnou editaci menu je vytvořena tabulka „vedlejsi\_menu“. V ní budou uchovávány všechny informace o tlačítkách vedlejšího menu, jako například text tlačítka, odkaz na stránku, příslušnost kam položka patří (aktuální informace, úřední deska, obecní úřad atd.) a taky číslo v seznamu při výpisu (slouží pro práci se styly a změnou stylu, pokud se v sekci nachází – zviditelnění).

## 7 Správa souborů

### 7.1 Životní cyklus informace

Hlavním prvkem systému je zveřejněná informace, buď jako aktualita, nebo na úřední desce. Bude mít tři stavy – Aktuální, Archiv, Vyřazena. Informace by se neměly z databáze mazat, jelikož musí být vždy dohledatelná, ale pokud nastane opravdu výjimečný případ a nebude se už hodit ani do archívu, může být vyřazena, ale ne smazána. To platí i o uživatelích. Může nastat situace, kdy někdo, kdo měl přístup k systému a vkládal do něj nějaké informace, může být propuštěn, přeložen na jinou pozici nebo úplně nahrazen. Jeho účet by se neměl hned úplně vymazat, ale měla by se mu změnit jen jeho role a oprávnění. Jeho index může být stále uložen jako cizí klíč v jiných tabulkách, jakožto autor informace a kdyby autor neexistoval, mohla by nastat chyba.



Obr. 4 Obrázek 4 - Stavový diagram zveřejněné informace

Informace má tedy tři stavy. Jako aktuální je vedena do té doby, než vyprší datum platnosti (nebo sejmoutí, záleží na pojmenování). Pokud je aktuální, je zveřejněna hned na vrcholu pořadí. V tomto stavu může být přeřazena do archívu. Pokud se ale přeřazuje z archívu do stavu aktuální, musí se kontrolovat datum platnosti. Pokud je datum platnosti starší než momentální datum, změna se neprovede.

Do stavu vyřazeno se smí přeřadit jen archivované informace, aktuální se rovnou vyřadit nemůžou. Naopak to však jde. Ze stavu vyřazeno lze převést informaci do stavu aktuální, ale opět se musí hlídat datum.

U úřední desky lze měnit i typy informace. Jedná se o rozdělení informací do sekcí, pod kterými je uživatelé potom můžou vyhledávat.

## 7.2 Výsledný redakční systém

### 7.2.1 Správa souborů úřední desky

Při analýze jsem si všiml, že na webu jsou některé informace a soubory chybně zařazeny. Například v sekci aktuální informace se objevují místo obyčejných aktualit i věci z úřední desky, která naopak není tolik vytěžována, jak by měla být a proto jsem se zaměřil na rozdělení těchto informací, protože by správné rozdělení mělo být základ, ať nedochází k omylům při vyhledávání. Na obrázku je vidět editor pro vkládání a upravování informací na úřední desce. Přístup k němu mají jen ti uživatelé, kteří dostanou oprávnění.



Obr. 5 Obrázek 5 - Editor pro vkládání a upravování informací na úřední desku

Navíc je zde vidět, jak vlastně web vypadá a jak je rozčleněn. Konkrétní editor je řešen pomocí formuláře a nejedná se o klasický formulář, ale o „multipart/form-data“. Je to z toho důvodu, protože je potřeba přenášet text i datové soubory.

Kromě klasických textových polí jsou zde navíc pole pro výběr z možností, které mají na starost stav a typ informace, které zařadí informaci do příslušné sekce úřední desky. Jedná se o povinné údaje, proto jsou textové pole ošetřeny atributem `required`. Datum sejmutí hraje podstatnou roli, protože podle něj se příspěvky automaticky, nebo ručně přesunují ze stavu aktuálního do stavu archiv. Automa-

ticky to má na starost funkce v kontroleru úřední desky, která se volá vždy před výpisem. Proto i zde je atribut require.

Pro konkrétní výběr data sejmutí je zde pomocník v podobě datepickeru z nadstavbové JavaScript knihovny jQuery, který se objeví po kliknutí do příslušného textového pole. Výběr souboru z příslušného adresáře má na starost pole typu soubor.

Správce příspěvků na úřední desce bude mít přístup do sekce, kde může vidět i podrobné informace o příspěvcích na úřední dece.

Úvodní strana > Dnes je: pondělí 4. května 2015 12:36

## Editace všech příspěvků úřední desky

[Vložit nový příspěvek](#)

ID	Název	Stav	Typ	Vloženo	Datum sejmutí	Poslední úprava		
13	Záměr prodeje, pronájmu, směny	Archiv	Záměr prodeje, pronájmu, směny	29. dubna 17:09	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
12	Záměr prodeje, pronájmu	Archiv	Záměr prodeje, pronájmu	29. dubna 17:08	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
11	Veřejná vyhláška	Archiv	Veřejná vyhláška	29. dubna 17:08	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
10	Záměr prodeje	Archiv	Záměr prodeje	29. dubna 17:07	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
9	Pozvánka	Archiv	Pozvánka	29. dubna 17:06	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
8	Nařízení obce	Archiv	Nařízení obce	29. dubna 17:06	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
7	Oznámení obce	Archiv	Oznámení obce	29. dubna 17:05	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
6	Dražební vyhláška	Archiv	Dražební vyhláška	29. dubna 17:04	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
5	Výběrové řízení	Archiv	Výběrové řízení	29. dubna 17:03	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
4	rozpočet	Archiv	Rozpočet obce	29. dubna 16:59	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
3	různé - ostatní	Archiv	Různé - ostatní	29. dubna 16:51	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
2	Usnesení	Archiv	Usnesení	29. dubna 16:47	30. dubna 00:00	30. dubna 20:59 Jan Vodák		
1	Záměr pronajmout pozemky	Aktuální	Záměr pronájmu	29. dubna 10:54	6. května 00:00	29. dubna 17:22 Jan Vodák		

Obr. 6 Obrázek 6 - Sekce kompletní správy příspěvků úřední desky

Na obrázku jsou vidět, o jaké podrobné informace se jedná, například kromě jinak viditelných informací v editoru i datum vložení a kdy a kdo provedl poslední úpravu příspěvku. To se může hodit, pokud se bude jednat o nějakou chybnou úpravu, protože se hned může zjistit, kdo je za chybu zodpovědný.

Na obrázku je vidět, že jeden z příspěvků je aktuální a proto nemůže být vyřazen. Ostatní příspěvků jsou ve stavu archivované informace, a proto mohou být vyřazeny.

Editace provozní doby se nachází v samostatné sekci, kde jsou vypsány dvě tabulky. V jedné jsou vypsány všechny hodnoty otevírací doby pro obecní úřad a knihovnu a k nim je samozřejmě vypsána i ikonka pro přístup do editace. V druhé

tabulce jsou vypsaný dny, kdy dochází k omezení otevírací doby, je tedy zavřeno. Hodnoty lze editovat zase pomocí datepickeru a pokud se chce vložit nové datum omezení, vybírá se z kalendáře, který je dodán opět přes jQuery knihovnu.

### 7.2.2 Mechanismus zpráv

Uživatel by měl dostávat informace o tom, zda se jeho dotaz nebo výsledek činnosti provedl správně, nebo nastala nějaká chyba, tak jak je tomu zvyklý ze všech desktopových aplikací. Proto je potřeba zavést podobný mechanismus i do naimplementovaného redakčního systému.

Úvodní strana > Dnes je: čtvrtek 7. května 2015 19:17

- ✓ Příklad potvrzující zprávy.
- i Příklad informační zprávy.
- ! Příklad chybné zprávy.

[Vložit nový příspěvek](#) [Editace všech příspěvků](#)

**Aktální příspěvky na úřední desce obce Vnorovy kde dni 7.05.2015**

Vyvěšení	Sejmutí	Název dokumentu	Původce	Typ	Editovat
----------	---------	-----------------	---------	-----	----------

[Archiv úřední desky](#)

Obr. 7 Obrázek 7 - Mechanismus zpráv

Mechanismus se zavádí do kontroleru, protože dochází ke komunikaci přímo s uživatelem a konkrétní znění zprávy se vypisuje do šablony. Nemělo by se využívat vyskakovacích oken, jelikož k tomu uživatelé chovají jistou averzi a pokud by zpráv vzniklo během životního cyklu více, objevilo by se zbytečně mnoho oken.

Nesmí být zapomenuto, že velmi často dochází k přesměrovávání, při kterém se smaže obsah všech proměnných. Jednalo by se např. o situaci, kdy se vypisuje zpráva, že k dané funkci musí být uživatel přihlášený a přesměrujeme ho na přihlašovací formulář. V tu chvíli dojde k přechodu na novou stránku (ještě než se něco stihne vypsat) a ta už nebude zprávu obsahovat. Místo obyčejných proměnných budou zprávy uloženy do pole „\$\_SESSION“. Toto pole se jeví jako ideální řešení, protože je unikátní pro každého uživatele stránky a data v něm přežívají mezi požadavky 24 minut nebo do zavření prohlížeče (Čápka, 2014).

## 8 Diskuze a závěr

### 8.1 Diskuze

Jelikož je redakční systém města velmi komplikovaná a rozsáhlá věc, některé části budou potřebovat ještě trošku pozornosti. Protože původní webová stránka neobsahovala žádný redakční systém, zaměřil jsem se na návrh základní funkcionality jako je databáze a práci s ní, vyřešil jsem základní zabezpečení, přihlašování, správu a nahrávání souborů včetně správy uživatelů. Byl vytvořen dobře fungující, moderní základ s drobnými, ale podstatnými detaily jako například zaslání e-mailu s novým heslem, pokud si o něj uživatel zažádá.

Dalo by se navrhnout zlepšení třeba pro sekci knihovna, protože v původní stránce zaujímal nezanedbatelnou roli. Mohl by se zde vytvořit účet s rolí Editor, který by měl přístup do sekce knihovny a staral by se o ni. To stejné platí i pro část Kultura, sport a spolky, kde by se dalo vymyslet i další zlepšení. Obec by jistě ocenila možnost přímé komunikace s návštěvníky stránky například pomocí diskuzního fóra.

### 8.2 Závěr

Cílem práce bylo navrhnout zlepšení, nebo vytvoření nového redakčního systému pro město nebo obec. Po provedení průzkumu jsem zjistil, že webová prezentace obce Vnorovy neobsahuje žádný redakční systém a proto bude zapotřebí vytvořit nový. V průběhu studie jsem zvažoval možnosti využití jiných existujících šablon, ale z důvodu ohrožení budoucnosti vývoje a změn v rozhraní, jsem navrhnul plně objektový redakční systém, při němž jsem využil objektově orientované PHP a dodržel architekturu MVC. Jednotlivé prvky systému jsou na sobě zcela nezávislé a díky tomu lze různě měnit nebo inovovat funkcionality systému a neohrozit tím jiné prvky. Zaměřil jsem se na URL (žádné parametry za otazníky) a pro editaci textů jsem využil WYSIWYG editor. Setkal jsem se i s jinými redakčními systémy a udělal si představu o tom, co by komplexní systém měl obsahovat.

V první řadě se jednalo o bezpečnost systému při práci s ním, ale i o bezpečnost dat. Během práce jsem navrhnul základní mechanismus, kterým se snažím předejít základním bezpečnostním rizikům jako například cross site scripting, SQL injection, mass assignment, nebo mimo jiné i chybám při vyvolávání výjimek.

Musel jsem si dát pozor na omezení dané na základě příslušných zákonů a vyhlášek. Konkrétně to byl zákon č. 106/1999 a k němu se vztahující vyhláška 442/2006 o povinném zveřejňování informací, která stanoví strukturu informací zveřejňovaných o povinném subjektu umožňující dálkový přístup. Další omezení představuje č. 64/2008 o formě uveřejňování informací souvisejících s výkonem veřejné správy prostřednictvím webových stránek pro osoby se zdravotním postižením (vyhláška o přístupnosti)

Další důležitou částí je přístup uživatelů do systému. Zjistil jsem, že všichni uživatelé systému nebudou mít stejná přístupová práva a že ne všichni budou moct

vykonávat všechny činnosti. Proto jsem navrhnul jednoduchý RBAC mechanismus na správu přístupu. Vytvořil jsem pár rolí uživatelů a k nim jsem přiřadil oprávnění, které budou moct vykonávat.

Alfa a omega systému je vkládání souborů a zveřejňování informací na úřední desku, nebo oznamování novinek a aktuálních informací o dění v obci. V původní stránce se vyskytovalo pár chyb v určení, o jakou informaci se jedná, kam ji zařadit a v jaké části bude zveřejněna. Proto jsem se zaměřil na typy informací, a snažil jsem se informace nějak rozčlenit. Pro správu informací jsem navrhnul jednoduché, uživatelsky přívětivé editory, které umožní nahrávání souborů a informací do příslušných sekcí.

Z pohledu ekonomického zhodnocení je velmi obtížně stanovit přesně vložené prostředky na zhotovení systému, jelikož jsem stále student s nedostatečným množstvím zkušeností z praxe. Avšak pokud bych svou práci ohodnotil na 5 až 10 tisíc korun, tak si myslím, že by to bylo asi přeměřené v porovnání s časem, kterým jsem tomu obětoval, ale v rámci studia jsem vše činil zcela zdarma. Průměrná firma si za vývoj redakčního systému účtuje odhadem sumu v řádech desítek tisíc korun. Z pohledu obce by se jednalo velmi výhodnou spoluprací a určitě by na tom vydělala.

Hodnocení z jiného pohledu. Obec se rozhodla ponechat si svou stávající stránku bez redakčního systému a obsah chce spravovat tak jako do teď. Zaměstnanec, který to má na starost, bude muset obětovat velké množství svého pracovního času, aby vykonal potřebné úkony, na místo toho, aby využil systém, který by mu vše usnadnil a on svůj čas mohl lépe využít. Redakční systém tedy usnadní práci, ale i náklady společně s prací. Proto i z tohoto pohledu je mnou navržený redakční systém ekonomicky efektivní.

## 9 Přehled informačních zdrojů

MACHÁŇ, V. Strukturovaný redakční systém. Brno, 2003. 88 s. Diplomová práce MZLU v Brně.

LOŠŤÁK, M. Webové frameworky a tvorba redakčního systému. Brno, 2010. 48 s. Bakalářská práce. MZLU v Brně.

ZÁVODNÝ, V. Programové vybavení 2. Břeclav, 2010. 69 s. Výuková skripta SOŠ průmyslové Edvarda Beneše Břeclav

Nette Foundation, Nette [online]. 2014 [cit. 2014-12-01]. Nette\Forms. Dostupné z WWW: <<http://doc.nette.org/cs/2.2/vulnerability-protection>>.

MCARTHUR, Kevin. Pro PHP: Patterns, Frameworks, Testing & More: Patterns, Frameworks, Testing and More. [s.l.] : Apress, 2008. 349 s. Dostupné z WWW: <<http://books.google.cz/books?id=5IRcTr13SBcC&pg=PR2&dq=1-59059-819-9&cd=1#v=onepage&q=1-59059-819-9&f=false>>. ISBN 1-59059-819-9.

ČESKO. Zákon č. 106 ze dne 11. května 1999 o svobodném přístupu k informacím. In: *Sbírka zákonů České republiky*. 1999, částka 39, s. 2578-2582. Dostupný také z: [http://www.vlada.cz/assets/urad-vlady/poskytovani-informaci/nejdulezitejsi-predpisy/zakon\\_106\\_1999.pdf](http://www.vlada.cz/assets/urad-vlady/poskytovani-informaci/nejdulezitejsi-predpisy/zakon_106_1999.pdf)

Vyhláška ministerstva informatiky České republiky č. 442/2006 Sb. ze dne 31. srpna 2006, kterou se provádějí některá ustanovení zákona ČNR č. 106/1999 Sb. o svobodném přístupu k informacím

Vyhláška ministerstva vnitra České republiky č. 64/2008 Sb. ze dne 28. února 2008, kterou se provádějí některá ustanovení zákonů ČNR č. 81/2006 Sb., č. 110/2007 Sb. a č. 269/2007 Sb.

ČÁPKA, David. ITnetwork [online]. 2013 [cit. 2015-04-03]. Objektový MVC redakční systém v php - popis architektury. Dostupné z WWW: <<http://www.itnetwork.cz/objektovy-mvc-redakcni-system-v-php-popis-architektury/>>.

ČÁPKA, David. ITnetwork [online]. 2013 [cit. 2015-04-03]. Objektový MVC redakční systém v php – htaccess, autoloader, obecný kontroler. Dostupné z WWW: <<http://www.itnetwork.cz/objektovy-mvc-redakcni-system-v-php-htaccess-autoloader-kontroler/>>.

ČÁPKA, David. ITnetwork [online]. 2013 [cit. 2015-04-03]. Úvod do knihoven a frameworků v PHP. Dostupné z WWW: <<http://www.itnetwork.cz/php-tutorial-uvod-do-knihoven-a-frameworku/>>.



ČÁPKA, David. ITnetwork [online]. 2013 [cit. 2015-04-03]. Databázový wrapper. Dostupné z WWW: < <http://www.itnetwork.cz/objektovy-mvc-redakcni-system-v-php-pdo-crud-wrapped/>>.

ČÁPKA, David. ITnetwork [online]. 2013 [cit. 2015-04-03]. Mechanismus zpráv. Dostupné z WWW: < <http://www.itnetwork.cz/objektovy-mvc-redakcni-system-php-mechanismus-zprav/>>.

9lessons.info [online]. 2008 [cit. 2015-05-05]. Hack your Own web project? SQL incjtcion. Obrázek ve formátu PNG. Dostupné z WWW: < <http://www.9lessons.info/2008/12/sql-injection.html/>>.

# **Přílohy**

## A Seznam použitých zkratk

CSS – Content Management Systém - kaskádové styly  
CMS – Cascading Style Sheets - redakční systém  
COBOL - COmmon Business Oriented Language – programovací jazyk  
DB – Database - databáze  
EDM - electronic document management system – system pro správu dokumentů  
ERD – Entity relationship diagram – entitně relační diagram  
FTP – File Tranfer Protocol – protokol pro přenos souborů  
HTML – Hyper Text Markup Language - hypertextový značkovací jazyk  
HTTP – Hyper Transfer Protocol – internetový protokol  
Id – Index – hodnota v databázi, slouží pro práci s konkrétní položkou  
MVC – Model View Controler – softwarová architektura  
OOP –Object-oriented programming - objektově orientované programování  
PDF - Package Definition File – typ souboru  
PDO – PHP Data Objects - PHP rozhraní pro práci s SQL databází  
PHP – Hypertext Preprocessor – skriptovací programovací jazyk  
PHTML – HTML soubor obsahující části PHP  
RBAC – Role-based access control – metoda pro přístup k informacím  
SQL – Structured Query Language - strukturovaný dotazovací jazyk  
SŘBD – Systém řízení báze dat – systém pro práci s databází  
URL – Uniform Resource Locator - přesná specifikaci umístění zdrojů informací  
WWW – Word Wide Web - prohlížení, ukládání a odkazování dokumentů  
WYSIWYG – What you se eis what you get – editor pro práci s textem na stránce  
XSS – Cross - site scripting – druh útoku na data

## B CD příloha

Na přiloženém CD se nachází kromě elektronické verze tohoto dokumentu také zdrojové kódy redakčního systému, vyexportovaná základní kostra databáze včetně vzorových dat, všechny použité utility, obrázky a ikony.

### Obsah přiloženého CD

- **BP\_vodak.pdf** – text této práce ve formátu PDF
- **database.sql** – vyexportovaná databáze včetně vzorových dat
- **img** – adresář s obrázky a ikonami
- **utility** – adresář s utilitami
- **kontrolery** - adresář se zdrojovými kódy typu kontroler
- **modely** - adresář se zdrojovými kódy typu model
- **pohledy** – adresář se zdrojovými kódy typu view