

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

BAKALÁŘSKÁ PRÁCE



Webová aplikace pro správu elektronického obsahu

Lukáš ZELENKA

© 2016 ČZU v Praze

STRANA 1 ze zadání

STRANA 2 ze zadání

Prohlašuji tímto, že jsem zadanou bakalářskou práci vypracoval samostatně pod vedením Ing. Davida Buchtely, Ph.D. a uvedl v seznamu literatury veškerou použitou literaturu a další zdroje.

V dne

.....
(podpis autora)

Poděkování

Mé poděkování patří Ing. Davidovi Buchtelovi, Ph.D. za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnoval. Dále bych chtěl poděkovat panu Ing. Josefu Pavlíčkovi, Ph.D. za konzultaci.

Abstrakt

Tato práce je zaměřena na webové aplikace, které mají za úkol spravovat a prezentovat data. Zaměřuje se na problematiku menších webových prezentací, které jsou vhodné pro malé firmy prezentující se v síti internet. Práce se zaměřuje na jednu zvolenou společnost, ve které pomáhá vytvořit redakční systém na míru. Vytvořený redakční systém je psán v jazycích HTML, CSS, Javascript, PHP, SQL a je napojen na databázový systém MySQL. Vytvořené řešení na míru pro zvolenou organizaci poskytuje veškeré možnosti, které byly požadovány. Hlavním přínosem této práce je rozbor nejznámějších technologií pro tvorbu webových aplikací a poskytnutí přehledu o možnostech tvorby vlastního redakčního systému ve výše zmíněných jazycích.

Klíčová slova: webová aplikace, web, redakční systém, cms, Systém pro správu obsahu, php, mysql, html, css, oop

Abstract

This Bachelor's thesis is focused on website applications which are designed to manage and present data. It deals with the issues of smaller websites that are suitable for small businesses presenting on the Internet. The thesis focuses on one selected company in which it helps to create a customized content management system. The created CMS is written in HTML, CSS, Javascript, PHP, SQL, and is connected to the MySQL database system. The created customized solution for the selected company provides all the required functions. The main contribution of this thesis is to analyze the best-known technologies for creating web applications and to provide an overview about the possibilities of making one's own content management system in the above-mentioned languages.

Keywords: web application, web, content management system, cms, php, mysql, html, css, oop

Obsah

1	Úvod.....	11
2	Cíl práce a metodika	12
2.1	Cíl práce.....	12
2.2	Metodika.....	12
3	Přehled řešené problematiky.....	13
3.1	Vymezení pojmů	13
3.1.1	Internet	13
3.1.2	World Wide Web	13
3.1.3	Protokol http	14
3.1.4	Internetový prohlížeč	14
3.1.5	Webové stránky	14
3.1.6	Webové servery	16
3.1.7	Databázové systémy	16
3.2	Používané technologie pro tvorbu webové aplikace	17
3.2.1	Scriptovací, programovací a značkovací jazyky.....	18
3.2.2	Relační databáze	21
3.2.3	MVC: Model-View-Controller	23
3.2.4	Frameworky a knihovny	23
3.3	Redakční systémy	25
3.4	Metoda návrhu SW produktu	27
3.4.1	Vodopádový přístup.....	27
3.4.2	Spirálový přístup.....	27
4	Vlastní řešení	28
4.1	Analýza požadavků	29

4.2	Návrh webové aplikace	32
4.2.1	Použité technologie	32
4.2.2	Adresářová struktura	33
4.2.3	Databázový model	33
4.3	Implementace webové aplikace.....	34
4.3.1	Adresářová struktura	34
4.3.2	Databáze.....	36
4.3.3	Software třetích stran	43
4.3.4	Zásuvné moduly.....	44
4.4	Testování webové aplikace.....	47
4.5	Nasazení webové aplikace do provozu.....	48
5	Zhodnocení výsledku	49
	Závěr.....	50
	Seznam použitých zdrojů	51
	Seznam použitých zkratk	55
	Přílohy	56

Seznam Obrázků

Obrázek 1: Adresářová struktura aplikace	36
--	----

Seznam Tabulek

Tabulka 1: Ukázka užití CSS	19
Tabulka 2: Ukázka javascriptu	20
Tabulka 3: Databázová tabulka s uživateli	37
Tabulka 4: Databázová tabulka s oprávněním uživatelů	37
Tabulka 5: Databázová tabulka se zdroji / zásuvnými moduly	38
Tabulka 6: Databázová tabulka s druhy oprávnění	38
Tabulka 7: Databázová tabulka s CMS stránkami	39
Tabulka 8: Databázová tabulka s aktualitami	39
Tabulka 9: Databázová tabulka s kategoriemi	40
Tabulka 10: Databázová vazební tabulka mezi novinkami a kategoriemi	40
Tabulka 11: Databázová tabulka se záznamy o aktivitě	41
Tabulka 12: Databázová tabulka s informacemi o souborech	41
Tabulka 13: Databázová tabulka s informacemi o navigačních panelech	42
Tabulka 14: Databázová tabulka s položkami v navigačních panelech	42
Tabulka 15: Databázová tabulka s fotoalbami	43
Tabulka 16: Databázová tabulka s fotografiemi	43

1 Úvod

V dnešním moderním světě se lidé neobejdou bez elektronických dat. Tato data jsou uchovávána v různých zařízeních, ke kterým lze přistupovat různými metodami. Jedním ze způsobů, jak lze k datům přistupovat, je webová aplikace.

Webové aplikace jsou dnes velmi rozšířeným a využívaným nástrojem pro zprávu elektronického obsahu a jejich vývoj má smysl. Webové aplikace jsou uloženy na straně serveru, a tak není zapotřebí žádná speciální instalace na straně uživatele. Uživatel k aplikaci přistupuje pomocí webového prohlížeče, který je dnes součástí většiny operačních systémů.

Autor této práce se zabývá problematikou tvorby webové aplikace, technologiemi, které jsou vhodné pro její tvorbu, a způsoby, jakými lze k tvorbě přistupovat. Předmětem této práce jsou především malé redakční systémy, které mají za úkol menším organizacím a společnostem tvořit webové prezentace bez dalších znalostí programování či psaní webových dokumentů.

V práci si ukážeme jak svobodné produkty, tak i tvorbu vlastní aplikace, která bude disponovat podobnými funkcemi.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem této práce je seznámení s problematikou webových aplikací pro správu elektronického obsahu. Práce nabízí přehled o tom, které technologie jsou nejčastěji s touto problematikou spojeny. Hlavním cílem v této práci je dohledání organizace, která redakční systém poptává. Vybrané organizaci je vytvořeno funkční řešení, které splňuje jejich poptávané požadavky.

2.2 Metodika

Metodika práce spočívá ve studiu publikací zaměřených speciálně na webové inženýrství a webové aplikace. Prvním krokem je nastudování problematiky webových aplikací využívaných i možnosti vlastního řešení. Poté následuje analýza všech požadavků a dle vybraných kritérií rozhodování, zda využít hotové produkty s otevřeným zdrojovým kódem, nebo vytvořit řešení vlastní.

3 Přehled řešené problematiky

3.1 Vymezení pojmů

3.1.1 Internet

Internet je celosvětovou veřejnou počítačovou sítí. Síť je tvořena datovými spoji mezi servery (počítače nabízející služby) a jejími uživateli. V síti Internet se komunikuje za pomoci protokolu IP. Společným cílem všech uživatelů využívajících Internet je spolehlivá komunikace (výměna dat). Nejznámější službou posledních let poskytovanou v rámci Internetu je World Wide Web (WWW). [1]

3.1.2 World Wide Web

World Wide Web (WWW, web) je označení pro vzájemně propojené dokumenty za pomoci hypertextu v síti internet. Tyto dokumenty jsou interpretovány značkovacím jazykem a jsou ukládány na straně serveru. Ty jsou pak na základě http požadavku od klienta zaslány do webového prohlížeče k dalšímu zpracování. Webový prohlížeč obdrží ze strany serveru vyžádaný dokument ve formě zdrojového kódu, který dále zpracovává a zobrazuje. [1]

WWW je kombinací čtyř důležitých součástí:

- Hypertext – Jedná se o strukturovaný text, který je obohacen o odkazy. Tyto odkazy mohou být v rámci jednoho, ale i několika dokumentů.
- Identifikátor zdroje (resource identifier) – Unikátní identifikátor, který určuje přesné umístění serveru a dokumentu na něm uloženého. Označuje se jako URL, případně jako URI.
- Architektura Klient / Server – Klient se za pomoci URL dotazuje na dokumenty uložené na straně serveru. Tyto vyžádané dokumenty (za pomoci URL) server zašle zpět klientovi, který si je vyžádal.
- Značkovací jazyk – je jazyk, jehož zdrojový kód (text) obsahuje současně vlastní text i instrukce pro jeho zpracování. Příkladem je HTML, který určuje strukturu textu a s jeho pomocí se tvoří webové stránky. [2]

3.1.3 Protokol http

HTTP (Hypertext Transfer Protocol) je protokol využívaný v síti internet. Je přednostně určen k výměně hypertextových dokumentů psaných HTML jazykem. Tento protokol obvykle využívá TCP port 80 a je společně s protokoly pro elektronickou poštu tím nejvíce využívaným. HTTP se zasloužil o obrovský rozmach internetové sítě. [1]

HTTP protokol využívá některé další aplikace. Nejvíce známou je jednotný lokátor prostředků URL (Uniform Resource Locator), který má za úkol jednoznačně určit umístění nějakého zdroje v Internetu. [3]

Základní vlastnosti http protokolu:

- Není řešen transport dat mezi partnery
- Nemá na starost detekci přenosových chyb
- Je bezstavový (model požadavek – odpověď)
- Není uchována žádná informace mezi spojeními
- Požadavek i odpověď mají shodnou strukturu
- Délka zprávy ani řádku není omezena
- Není omezen pouze na znaky standardního ASCII kódu [2]

3.1.4 Internetový prohlížeč

Internetovým prohlížečem je software a ve většině případech je již součástí základním programového vybavení operačních systémů. Internetový prohlížeč je mnohdy nazýván webovým prohlížeče. Tento program slouží uživateli k procházení webových stránek, které jsou vyžádány za pomoci protokolu http. Cílem webového prohlížeče je zpracování přijatých ve formě zdrojového kódu a uživateli následně zobrazit a naformátovat obsah webové stránky přesně podle zdrojového kódu. [4]

3.1.5 Webové stránky

Webové stránky se dělí na statické WWW stránky a dynamické webové aplikace.

3.1.5.1 Statické WWW Stránky

Statická webová stránka je webová stránka, která je popsána za pomoci HTML jazyka. Takto napsána webová stránka nemá žádnou programovou logiku a její obsah je konkrétně

uložen na disku. Statické webové stránky nevyužívají žádného programovacího jazyka. Klientům je doručován obsah webové stránky tak, jak je uložen na disku webového serveru, případně jiném médiu. Z tohoto důvodu lze tvrdit, že obsah statické webové stránky je u všech klientů totožný.

Výhody:

- Není zapotřebí znát žádný programovací jazyk, tvůrce webových stránek si vystačí se HTML a kaskádovými styly (CSS).
- Velmi nízké nároky na internetové služby poskytující hostování webu. Není zapotřebí žádný interpret pro zpracování webové aplikace.

Nevýhody:

- Velmi špatná údržba většího množství webových statických stránek, kde je zapotřebí v případě úpravy navigačního panelu tuto změnu provést na všech HTML dokumentech webu.
- Velmi obtížné určit kontext jednotlivým uživatelům.

3.1.5.2 Dynamické webové aplikace

Dynamické webové aplikace nám umožňují nahradit některé z aplikací, které bychom normálně museli mít nainstalované v počítači. Prostřednictvím internetové sítě je pak možné tyto aplikace využívat bez nutnosti instalace dalšího softwarového vybavení počítače.

Výhody:

- Nezávislost na platformě
- Minimální nároky na hardware
- Aktuálnost aplikace
- Možnost online spolupráce

Nevýhody:

- Závislost na internetovém připojení
- Závislost na poskytovateli webové aplikace

Webové aplikace se v dnešní době stále rozšiřují. Příkladem webových aplikací mohou být WWW stránky facebook.com, google.com, seznam.cz, ebay.com či aukro.cz, ale například

i webové stránky hc-rakovník.cz. Webovou aplikací jsou také redakční systémy, které společnosti využívají jako nástroj pro sdílení obsahu na internetové síti. [5]

3.1.6 Webové servery

Webový server lze chápat jako aplikaci, která je spuštěna na pozadí počítače jako služba. Tato služba je zodpovědná za zpracování a vyřízení požadavků klienta, který vyžádá určitý soubor za pomoci URL adresy. Server tento http požadavek přijme a obstará jeho zpracování. Výsledkem celého procesu je vrácení HTML kódu klientovi, který požadavek zaslal.

3.1.6.1 Apache2

Apache2 je program (server) pro poskytnutí http služeb. Jedná se o program s otevřeným zdrojovým kódem a je dostupný pro platformy Linux, Microsoft Windows, Mac OS X, BSD, Solaris a další. Apache disponuje možností rozšíření za pomoci modulů. Cílem tohoto webového serveru je bezpečnost, rozšiřitelnost a efektivnost celého programu. Nejčastěji se využívá v kombinaci LAMP – Linux, Apache2, MySQL/MariaDB, PHP/Perl/Python. [6]

3.1.6.2 Nginx

Nginx je webovým serverem s otevřeným zdrojovým kódem zaměřujícím se na rychlost a malé nároky na operační paměť počítače, na kterém je spuštěn. Nejčastěji se využívá na operačním systému Linux, ale je k dispozici i pro Solaris, Mac OS X a Microsoft Windows. [7]

3.1.6.3 IIS

Internet Information Services je webovým serverem platformy Windows. Jde o velmi často využívaný softwarový server, který poskytuje kvalitní platformu pro spolehlivé hostování webů a služeb. Maximalizuje zabezpečení izolací jednotlivých aplikací. Nasazuje možnost spouštět webové aplikace psané v jazycích ASP, ASP.NET a PHP. IIS lze rozšiřovat za pomoci modulů. [8]

3.1.7 Databázové systémy

Databázové systémy jsou velmi často používaným nástrojem dodávaným se softwarem. Velmi často se databázové systémy pojí s názvem databáze (DB). Za databázi lze považovat spousty běžně využívaných případů. Příkladem může být telefonní seznam, adresář klientů

firmy, katalog zboží v konkrétním obchodě nebo seznam knih v knihovně. Tyto informace jsou evidovány v počítači a umožňují tak s informacemi pracovat rychleji a spolehlivěji. Tento produkt nahrazuje papírové kartotéky v administrativních pracovištích, se kterými jsme se dříve setkávali u lékařů, knihovnách či obchodních partnerů ve formě katalogu zboží. Každá jednotlivá karta v kartotéce obsahovala specifické údaje o sledovaném objektu, např. o konkrétním pacientovi či knize. V takovém případě bylo velmi časově náročné v této kartotéce vyhledávat jakákoliv data. Z tohoto důvodu nastupují elektronické systémy, které umožňují daleko efektivnější práci s evidovanými informacemi. [9]

Databázové systémy přichází v úvahu tehdy, když vzrůstá složitost nebo objem informací, které potřebuje uspořádat a spravovat a když ruční pořizování záznamů začíná být únavné a těžkopádné. [10] S databázovým systémem je pak komunikace poskytnuta za pomoci dotazovacího jazyka, který umožňuje pracovat s daty. Nejčastěji používaným dotazovacím jazykem je jazyk SQL. [11]

Structured Query Language je standardizovaným strukturovaným dotazovacím jazykem, využívaným k práci s daty v relační databázi. Tento jazyk je rozdělen do 4 základních skupin, které zahrnují manipulaci s daty, definici dat, řízení přístupových práv a řízení transakcí. Mezi manipulaci s daty patří klíčová slova SELECT pro výběr dat, INSERT pro vložení dat, UPDATE pro aktualizaci dat a DELETE pro mazání dat. Mezi definici dat lze zařadit příkazy CREATE pro vytvoření tabulek, ALTER pro editaci tabulek, DROP pro mazání tabulek. V řízení přístupových dat se využívá GRANT pro udělení práv, REVOKE pro odebrání práv. V řízení transakcí se pak využívá příkazů START TRANSACTION pro spuštění transakce, COMMIT pro potvrzení celé transakce a ROLLBACK pro odvolání prováděné transakce. Mimo výše zmíněné příkazy se v SQL můžeme setkat i se speciálními příkazy. [10]

3.2 Používané technologie pro tvorbu webové aplikace

Pro tvorbu webové aplikace jsou mimo základní tvorby webu - HTML, CSS a Javascript - využívány i programovací scriptovací jazyky spouštěné na straně serveru. Programovací jazyky na straně serveru jsou spouštěny při požadavku na webový server, který z URL adresy a jeho konfigurace rozpozná, který soubor má spustit. Programy jsou ve většině

případů u webových aplikací spouštěny za pomoci interpreta. Nejpoužívanějším jazykem pro webové aplikace je jazyk PHP. Dalším příkladem takového jazyka může být například Ruby, které se svojí nadstavbou (frameworkem) Ruby on Rails umožňuje tvořit webové aplikace. Tyto jazyky ovšem většinou využívají některý databázový systém jako bázi dat, do které si ukládá potřebná data a později je zase čerpá. Jako databázové systémy využíváme nejčastěji MySQL, Oracle, PostgreSQL a SQLite.

3.2.1 Scriptovací, programovací a značkovací jazyky

K tvorbě webových aplikací je zapotřebí více technologií, které je zapotřebí navzájem propojit. Hlavním stavebním kamenem webových stránek je značkovací jazyk. Nejčastěji používaným značkovacím jazykem pro tvorbu webových stránek je HTML. Dalším stavebním kamenem je poté jazyk CSS který popisuje, jak bude webová stránka vypadat. Při nutnosti interakce v prohlížeči využíváme skriptovacího jazyka Javascript. Pro tvorbu dynamických webových aplikací můžeme využívat různých programovacích jazyků. V současné době jsou nejrozšířenějšími pro tvorbu dynamické webové aplikace jazyky PHP či Ruby.

3.2.1.1 HTML

Hyper Text Markup Language je hypertextovým značkovacím jazykem, který byl navržen pro popis webových dokumentů – webových stránek. V tomto jazyce využíváme značek, nazýváme je také jako tagy, které popisují význam informací/textu, který nesou. Tyto značky se dělí na párové a nepárové. Párová značka nám vždy obaluje obsah dokumentu, ke kterému se váže. Příkladem párové značky může být kurzíva textu. Nepárová značka se vyskytuje samostatně a nevztahuje se k částem dokumentu, ale tvoří nový obsah. Příkladem nepárové značky může být obrázek nebo také zalomení textu na nový řádek. Každá HTML značka nám popisuje jiný typ obsahu v dokumentu. HTML značka je tvořena ze špičatých závorek a jejího názvu Otevírací značka pak vypadá následovně: <tag>. Závírací značka je obohacena o lomítko před názvem značky </tag>. Tyto značky mohou obsahovat i některé ze svých specifikovaných vlastností. Vlastnosti neboli atributy pak uvádíme mezi špičaté závorky otevíracího HTML značky. Pokud chceme uvést více atributů v rámci jedné značky, učiníme tak mezerou. Příkladem může být hypertextový odkaz Odkaz na example.com. [12]

3.2.1.2 XML

eXtensible Markup Language je velmi podobným jazykem k HTML. Tento jazyk je účelem určen pro popis dat na místo vzhledu. Jde o velmi rozšiřitelný jazyk, který umožňuje tvořit za pomoci vlastních značek a není tak omezen na předem značky předem definované. [13]

3.2.1.3 CSS

Kaskádové styly (Cascading Style Sheets) jsou jazykem, který má za úkol popisovat vzhled HTML, nebo XML dokumentu. Kaskádové styly popisují, jakým způsobem musí být elementy vykresleny na obrazovku, papír nebo kterékoliv jiné médium. [14] Soubor kaskádových stylů je koncipován následujícím způsobem: Jednotlivé elementy jsou vybrány takzvaným selektorem. Za selektorem je otevřen pomocí složených závorek blok všech vlastností, které se na vybrané elementy mají aplikovat. Jednotlivé vlastnosti od hodnot oddělujeme znakem dvojtečka a vlastnosti od vlastností oddělujeme za pomoci středníku. Po výčtu všech vlastností je blok uzavřen složenou závorkou. Obecný zápis může vypadat pak takto:

Obecný zápis	Reálný příklad
<pre>selector { vlastnost_1: hodnota_1; vlastnost_2: hodnota_2; }</pre>	<pre>body { background-color: black; color: white; }</pre>

Tabulka 1: Ukázka užití CSS

3.2.1.4 Javascript

Javascript je interpretovaný programovací scriptovací jazyk. Jeho interpretem je webový prohlížeč a je velmi často doplňujícím jazykem pro HTML a CSS. Javascript má za úkol dodat na webové stránky možnost interakce a je vykonáván na straně klienta. Javascript může být připojen jako externí soubor s koncovkou .js do HTML dokumentu za pomoci html značky <script> a atributem src="...", ve kterém je vložena URL adresa na externí script, nebo pouze značkou <script>, do které je pak vložen příslušný Javascript. [15]

Příklady:

Připojení externího programu v javascriptu main.js umístěném ve stejném adresáři, jako HTML soubor	<code><script src=“./main.js“></script></code>
Javascriptový program přímo v HTML dokumentu	<code><script> document.write(“Hello World!“); </script></code>

Tabulka 2: Ukázka javascriptu

3.2.1.5 AJAX

AJAX je označením pro asynchronní kombinaci javascriptu a XML. Jde o sadu, která je určena pro vývoj aktivních webových aplikací. Umožňuje měnit části webových stránek bez nutnosti jejich opakovaného načítání. K využití této sady je zapotřebí moderního internetového prohlížeče, který tuto technologii podporuje. [1]

3.2.1.6 PHP

Zkratka PHP je rekurzivním akronymem pro PHP: Hypertext Preprocessor. PHP je otevřený interpretovaný scriptovací jazyk sloužící hlavně pro tvorbu webových stránek. Tento scriptovací jazyk je hojně využíván v primitivních, ale i ve složitějších aplikacích. Jeho výhodou je vykonávání skriptu na straně serveru. Klient, který vzdáleně PHP script za pomoci http požadavku spustí, nevidí zdrojový kód programu, ale vidí pouze jeho výstupní data. Je důležité, aby byl na straně serveru nainstalovaný interpret tohoto jazyka. [16]

Skript v PHP je textový soubor s příponou, která mívá většinou tvar *.php. PHP Kód je uzavřen ve speciální značce `<?php a ?>`. Takto označený kód můžeme vkládat i mezi HTML kód tam, kde se nám zachce, a také kolikrát budeme chtít. Stačí dodržet pouze označení začátku a konce PHP kódu. [17]

Příklad:

```
<?php echo “Hello World!“ ?>
```

3.2.1.7 ASP.NET

ASP.NET je součástí .NET Frameworku a je určen pro tvorbu webových aplikací. Je nástupcem technologie ASP. Aplikace tvořené touto technologií jsou rychlejší díky

předkompilování. Předkompilace je prováděna do jednoho nebo více dynamických knihoven. [18]

3.2.1.8 Ruby

Ruby je programovacím jazykem, který se posouvá stále dopředu. Tento programovací jazyk se zaměřuje na jednoduchost a zároveň se snaží, aby bylo výhodné jej využívat. Jedná se o interpretovaný scriptovací programovací jazyk. Jazyk je plně objektově orientovaný, proto je vše v tomto jazyku objektem. Je důležité, aby byl na straně serveru nainstalovaný interpret tohoto jazyka. [19]

p “Hello World!“

3.2.2 Relační databáze

Relační databáze je zakládána na relačním datovém modelu (RDM). Relační datový model je na rozdíl od svých předchůdců založen na teoretickém základě. Relační databázový model je možné charakterizovat tímto způsobem:

- Hodnoty v tabulkách jsou atomické – skládání hodnot z dalších hodnot není přípustné.
- Hodnoty jsou skalární – Hodnoty musí být jednorozměrné.
- Hodnoty v tabulkách jsou evidovány jako součásti jednotlivých domén. Všechny prvky dané domény musejí být mezi sebou porovnatelné a musejí náležet jednomu datovému typu.
- Práce s tabulkami probíhá za pomoci výrokové logiky.
- Každá tabulka obsahuje jeden nebo více primárních klíčů ve formě domény (sloupce), které rozlišují jednotlivé řádky mezi sebou.
- Vazbu mezi jednotlivými tabulkami lze vytvořit za pomoci cizího klíče. Ten je uváděn v tabulce ve formě domény (sloupce).
- V tabulkách je umožněno vybrat podmnožinu ze všech evidovaných informací. Operace vybírající podmnožinu řádků se označuje jako selekce. Operace vybírající podmnožinu sloupců se nazývá projekce tabulky.
- Tabulky lze mezi sebou navzájem kombinovat jako běžné množiny. Kombinace můžeme provádět pomocí operací sjednocení, rozdíl, průnik a kartézský součin množin. Selekcce se v tomto případě nazývá spojení tabulek. [9]

Základní koncepcí v teorii relačních datovýchází je tedy relace, která je neformálním způsobem prezentována jako tabulka. [9]

3.2.2.1 *MySQL*

MySQL je svobodným systémem pro správu relační databáze. Jeho popularita je obrovská a stále silně narůstá. Dnes je MySQL významným produktem jak pro vytváření samostatných databázových aplikací, tak pro návrh a provozování databázově orientovaných webů. [10]

3.2.2.2 *Oracle*

ORACLE je prvním databázovým systémem navrženým pro podnikové výpočetní síť (grid computing) a je nejvíce flexibilním a nákladově efektivním způsobem, jak spravovat informace a aplikace. ORACLE využívá relačního datového modelu (RDM) a přístup k datům je zajištěn rozhraním SQL. [20]

3.2.2.3 *PostgreSQL*

PostgreSQL je velmi mocným objektově-relačním databázovým systémem. Jeho významným prvkem je vysoký stupeň spolehlivosti a integrity dat. Tento databázový systém je možné spustit ve většině hlavních operačních systémů, jako jsou Linux, BSD, Solaris a Windows. Tento databázový systém má plnou podporu cizích klíčů, spojování relačních tabulek, tvorbu pohledů, spouštěčů a spousty procedur. Je zahrnuta i většina datových typů, jako jsou Integer, Boolean, char, varchar, date, interval, timestamp. PostgreSQL má mimo jiné i podporu využívání úložiště pro binární soubory. Proto nabízí také možnost uložit obrázky, videa a zvukové stopy. [21]

3.2.2.4 *MariaDB*

MariaDB je jedním z nejpobulárnějších databázových systémů na světě. Je vytvořen původními vývojáři databázového systému MySQL, a to s otevřeným zdrojovým kódem. Jedná se o lepší náhradu za MySQL. Tento databázový systém je využíván kvůli jeho rychlosti, škálovatelnosti. MariaDB nabízí možnost rozšíření zásuvnými moduly a různými nástroji. MariaDB využívá relačního datového modelu (RDM), a proto je přístup k datům zajištěn rozhraním SQL. [22]

3.2.2.5 *SQLite*

SQLite je databázovým systémem, který je implementován jako knihovna do samotné aplikace. Proto není zapotřebí oddělený proces, který se stará zpracování SQL příkazů. Komunikace s databází probíhá za pomoci jazyka SQL. [23]

3.2.3 MVC: Model-View-Controller

MVC je softwarová architektura, která vznikla z potřeby oddělit u aplikací s grafickým rozhraním kód obsluhy (controller) od kódu aplikační logiky (model) a od kódu zobrazujícího data (view). Tím aplikaci zpřehledňuje, usnadňuje budoucí vývoj a umožňuje testování jednotlivých částí zvlášť. [24]

3.2.3.1 *Model*

Model je datovým a zejména funkčním základem v celé aplikaci a je v něm obsažena celá aplikační logika. Kterákoli akce vyvolaná uživatelem tak představuje akci zpracovanou modelem. Model disponuje vůči svému okolí pevně daným rozhraním. Změny či zjištění stavu pak probíhá voláním jeho vlastních funkcí. Model jako takový neví o existenci vrstvy View nebo Controller. [24]

3.2.3.2 *View*

View lze přeložit jako pohled a je vrstvou, která se stará o zobrazení výsledku zpracovávaného požadavku. Obvykle je v pohledu využíván šablonovací jazyk a ví, jak má zobrazit daný výsledek z modelové části aplikace. [24]

3.2.3.3 *Controller*

Controller je řadičem aplikace, který zpracovává požadavky zaslané uživatelem. Na základě těchto požadavků získá data z modelové vrstvy a požádá vrstvu View o vytvoření pohledu a jeho vykreslení. V Nette Frameworku je vrstva Controller nazývána Presenterem. [24]

3.2.4 **Frameworky a knihovny**

Framework je česky označován aplikačním rámcem. Jedná se o softwarovou strukturu, která slouží jako podpora programátorům při vývoji a organizaci dalších softwarových projektů.

Knihovna je sbírkou procedur a funkcí, které mohou být sdíleny mezi více programy. Obsah knihovny je poskytován pomocí aplikačního rozhraní.

3.2.4.1 *Nette Framework*

Nette Framework je stavěný tak, aby byl co nejpoužitelnější a nejvstřícnější. Jde o framework, s nímž je nejen snadné, ale i zábavné pracovat. Dává vám srozumitelnou a úspornou syntaxi, vychází vám vstříc při programování a debugování, nechává vás soustředit se na kreativní stránku vývoje a nepřidělavá vám vrásky. Eliminuje bezpečnostní rizika a podporuje využití Model-view-controller architektury. Můžete v něm tvořit e-shopy, wiki, blogy, CMS, co jen vymyslíte. [25]

3.2.4.2 *Dibi*

Dibi je chytrou knihovnou s aplikačním rozhraním pro práci s databázovými systémy psanou v jazyce PHP. Jedná se o abstraktní vrstvu pro různé databázové systémy, jako jsou MySQL, PostgreSQL, SQLite, MS SQL, Oracle, Access. Mimo jiné zahrnu také psanou v jazyce PHP. Tato vrstva umožňuje také práci s PDO a ODBC. [26] Tvůrcem této knihovny je David Grudl. Dibi layer má za svůj cíl maximálně ulehčit programátorům práci, eliminovat chyby a zachovávat maximální jednoduchost.

- zjednodušením psaní SQL příkazů
- snadnějším přístupem k metodám
- funkcemi pro rutinní úkony
- automatickou podporou konvencí (escapování)
- automatickým formátováním speciálních typů (například datum) [27]

3.2.4.3 *Zend Framework*

Zend Framework je objektově orientovaný a volně šiřitelný webový aplikační framework implementovaný v jazyce PHP. Tento aplikační rámec je licencován pod záštitou New BSD license. Zend Framework klade zřetel na jednoduchost tvorby webových aplikací. Využívá modulární architektury, kterou poskytuje vývojáři možnost využití jen potřebných komponent. Zend Framework zahrnuje komponenty pro tvorbu model-view-controller aplikací, autorizační a autentizační procesy. Mimo jiné také implementuje různé druhy

cache, filtrů, validatorů a jazykových komponent. Tento framework začal být vyvíjen v roce 2005, kdy mnoho mladých frameworků nabývalo své popularity. [28]

3.2.4.4 *Ruby on Rails*

Ruby On Rails, mnohdy zkracován jako Rails, je framework vytvořen v jazyce Ruby. Tento framework slouží k pohodlnému a rychlému vytváření moderních webových aplikací. Obsahuje spousty funkcí, které jsou k tvorbě zapotřebí jako například: abstraktní vrstvu pro práci s relačními databázemi, přehlednou implementaci model–view–controller architektury, zabudovanou podporu pro tvorbu automatizovaných testů, generátory kódu, jako je HTML, a spousty dalších. Rails byl publikován v roce 2004 dánským programátorem David Heinemeier Hanssonem a dnes je vylepšován tisíci přispěvateli. Rails používají i známé společnosti jako jsou: Apple, Github, Twitter, Electronic Arts, Oakley nebo The New York Times. [29]

3.2.4.5 *jQuery*

jQuery je chytrou javascriptovou knihovnou. Tato knihovna pracuje velice rychle a poskytuje snadné procházení HTML dokumentu a případnou manipulaci s HTML elementy. Knihovna zahrnuje spousty vlastností jako: odchyťávání událostí, animace, AJAX a spousty dalších. jQuery je možné rozšiřovat zásuvnými moduly. [30]

3.3 Redakční systémy

Úkolem Redakčního systému je poskytovat jak provozovateli, tak i uživateli (návštěvníkovi) příjemný a přehledný přístup k elektronickému obsahu – informacím - prostřednictvím internetové sítě. Provozovateli nabízí rychlou správu nového, ale i veškerého stávajícího obsahu. Uživatelům pak nabízí přehledné webové stránky s obsahem nastaveným provozovatelem a současně také nabízí funkce pro usnadnění. Nejčastější službou může být formátování obsahu webové stránky pro tisk.

Svobodné Redakční systémy

Redakční systémy jsou také vyvíjeny jako svobodně šířené aplikace, kde je celý zdrojový kód popsán dokumentací, a poskytuje tak každému programátorovi možnost zapojení se do jeho vývoje, upravovat jeho části a přizpůsobit celý software vlastním představám a tím celý software nastavit na vyřešení vlastní problematiky. Této skutečnosti lidé využívají a podílejí

se na vývoji svobodného software, čímž ho staví před komerční produkty, které tvoří jen pár jedinců a nikoliv široká veřejnost. [31]

Svobodný software se stává svobodným až uvolněním s licenční politikou, která vymezuje práva, jak může být s programem vynaloženo. Mezi nejčastěji využívané licence patří MIT, GPLv2, Apache, GPLv3 či BSD 3-clause. [32]

Nejznámějšími svobodně šiřitelnými redakčními systémy využívanými v osobní i komerční scéně jsou Wordpress, Joomla a Drupal. Všechny tyto 3 systémy jsou naprogramovány v jazyce PHP.

Komerční redakční systémy

Mimo svobodných redakčních systémů se na trhu můžeme setkat i s komerčními řešeními. Tyto redakční systémy jsou ve většině případů vlastními výtvary společností, které aplikaci na trhu nabízejí. Aplikace jsou pak nasazovány za daleko vyšší ceny, než svobodně šiřitelné. Důvodů, proč společnosti vyvíjejí vlastní redakční systémy je mnoho. Jedním z argumentů tvorby vlastního řešení je usnadnění práce do budoucna. Tvůrci vlastních redakčních systémů jsou schopni v reálné situaci reagovat daleko rychleji na požadavky klientů, jelikož své řešení znají dokonale. Nevýhodou tohoto řešení je délka vývoje a je finančně náročnější. Některé společnosti nabízejí na trhu své redakční systémy za měsíční paušální poplatek.

3.4 Metoda návrhu SW produktu

Tvorba kvalitního softwaru vyžaduje velké úsilí a plánování. Pro úspěšný vývoj jsou navrženy specifické přístupy metodik softwarového vývoje. Tyto metody umožňují usměrnění celého vývoje, a tím poskytnou i kvalitnější řešení.

3.4.1 Vodopádový přístup

Vodopádový přístup je sekvenčním vývojovým procesem, ve kterém je na vývoj nahlíženo jako na neustále se svažující tok postupně přes všechny fáze. Využití tohoto přístupu může vést k vyšší časové úspoře v pozdějších fázích jeho životního cyklu za předpokladu, že se věnuje dostatek času v počátečních etapách. Tento přístup je mnohdy kritizován a považován za nevhodný pro použití v praxi, jelikož mnoho detailů může vyjít najevo až při samotné implementaci. Fáze vodopádového přístupu:

1. Analýza požadavků
2. Návrh
3. Implementace
4. Integrace
5. Testování a ladění
6. Instalace
7. Údržba [33]

3.4.2 Spirálový přístup

Spirálový přístup reaguje na nedostatky, které vyplývají z vodopádového modelu. V tomto přístupu jsou zavedeny navíc 2 koncepty, které tyto chyby odstraňují – iterativní přístup a opakované analýze rizik. V tomto modelu se vychází ze situace, ve které je velmi obtížné specifikovat všechny potřebné funkce k dokončení. V úvodu se tedy určí pouze obecný rámec. Poté se provede implementace části vyvíjené aplikace, která se zkontroluje se zákazníkem. Podle konkrétní situace lze pak pokračovat opakováním těchto kroků (iterace). Jednotlivé fáze jsou:

1. Analýza
2. Vyhodnocení
3. Vývoj
4. Plánování [33]

4 Vlastní řešení

Pro moji bakalářskou práci jsem si vybral organizaci HC Rakovník, z.s. . Jde o zimní stadion ve městě Rakovník, který měl dlouhodobě problémy se správou svého webu. Starý redakční systém byl nevyhovující, jelikož neumožňoval to, co vedení vyžadovalo. Problematika byla řešena individuálně jednotlivými třídami na zimním stadionu, a tím byly všechny informace, které chtěli poskytnout veřejnosti, nepřehledně uspořádané. Pro vedení zimního stadionu je důležité, aby nad všemi informacemi měli kontrolu a mohli je v případě nutnosti změnit.

Celá tvorba projektu byla řízena metodikou vodopádového přístupu. Prvním krokem bylo seznámení s vlastní problematikou zvolené organizace. Seznámil jsem se se všemi požadavky, které jsme společně s organizací specifikovali. Tyto požadavky jsem postupně zanalyzoval a celkovou analýzu projednal se zimním stadionem. Analýzou jsme doladili detaily, které nebyly jisté při první schůzce.

V následující fázi jsem navrhl základní koncept celé aplikace. Do návrhu jsem promítl postupně všechny požadavky, které bylo zapotřebí uskutečnit. Návrh začal výběrem vhodné technologie pro vývoj aplikace. Dále jsem zvolil vhodné knihovny, které mi napomohly k realizaci celého projektu. V návrhu jsem zahrnul architekturu MVC, která mi umožnila oddělit grafické rozhraní od aplikační logiky.

Následujícím krokem byla implementace celé aplikace, která splňuje veškeré zjištěné a zanalyzované požadavky. V průběhu implementace jsem prováděl testování jednotlivých zásuvných modulů, které jsem tvořil, abych minimalizoval riziko výskytu chyb při plném testování.

Po dokončení implementace a mých interních testů jsem projekt spustil na testovací doméně, ke které dostal zimní stadion přístupové údaje. V tuto chvíli bylo testování aplikace plně v režii zimního stadionu. Jakmile dokončili testy, předali mi veškeré zjištěné informace a chyby tak, abych je byl schopný opravit.

Po opravě zjištěných závad testovacím procesem proběhla instalace webových stránek na hlavní doménu zimního stadionu (www.hc-rakovnik.cz). Instalace proběhla bez větších problémů. Problémem, se kterým nebylo možné nic udělat, byl A záznam, který zapříčinil

hodinovou desynchronizaci webových stránek. V průběhu instalace někteří návštěvníci mohli vidět starý web i přes fakt, že doména odkazovala již na nově připravenou aplikaci.

4.1 Analýza požadavků

- **WYSIWYG editor** – Pro tvorbu a editaci stávajícího elektronického obsahu na webových stránkách bude využíván editor, který umožňuje práci s daty i bez zkušenosti s HTML či CSS. Jako WYSIWYG je vyžadován CKEditor. CKEditor je požadován, jelikož byl využit i pro staré řešení webových stránek a správci webových stránek jsou na něj již navyklí.
- **Implementace vlastního řešení** - vyhnout se open source redakčním systémům
 - Důvodem pro tento požadavek jsou útoky robotů na redakční systémy s otevřeným zdrojovým kódem. U těchto redakčních systémů je tedy nezbytně nutné sledovat aktualizace. Ve vlastním řešení jsou útoky většinou velmi obecně zaměřené a nejsou tak cílené, jako u redakčních systémů s otevřeným zdrojovým kódem.
- **Zálohování obsahu CMS stránek a Novinek** – V aplikaci se data mění denně a může nastat situace, kdy je zapotřebí obnovit některé texty, případně data. Zálohy mohou probíhat i mimo aplikaci.
- **Administrační rozhraní** – Správa obsahu na webových stránkách by měla být oddělena od rozhraní pro návštěvníky webu.
- **Uživatelské rozhraní pro návštěvníky webových stránek** - Toto rozhraní bude vypadat jako běžné webové stránky. V uživatelském prostředí nebude možnost přihlášení ani žádné interakce mezi uživateli.
- **Modulární redakční systém** – Webová aplikace by měla být modulární a umožňovat pozdější rozšiřování formou zásuvných modulů – plug-ins. Každý zásuvný modul (plug-in) by měl umožňovat procházet, vytvářet, editovat či mazat vlastní data.

- **Základní zásuvné moduly redakčního systému**
 - **CMS stránky**
 - *Administrační část:* Jedná se o správu jednotlivých webových stránek v uživatelském prostředí.
 - *Uživatelská část:* zásuvný modul, který bude schopen zobrazit obsah zvolené stránky. Tyto stránky slouží jako prostředek pro předání informací. Příkladem může být stránka „o nás“.
 - **Záznamy o aktivitě**
 - *Administrační část:* Je třeba zaznamenávat veškeré úkony správců webových stránek. Jelikož se o obsah webu stará více lidí, je důležité monitorovat, který uživatel udělal jakou akci.
 - *Uživatelská část:* Napojením webu na službu Google Analytics bude správce webových stránek schopný zjistit spousty monitorovaných údajů, jako je návštěvnost stránek, průměrná doba návštěvy, chování uživatele na webu a další užitečné údaje.
 - **Navigační panely**
 - *Administrační část:* Obsah webových stránek se neustále mění a vyvíjí. Proto je vhodné, aby v případě potřeby změny struktury webových stránek bylo možné tuto strukturu vyjádřit v navigačním panelu.
 - *Uživatelská část:* Modul, který umožní uživatelům procházet webové stránky za pomoci navigace.

- **Aktuality** - Jsou zásadním modulem pro zvolenou organizaci. Potřebují informovat veřejnost o různých akcích, případně otevíracích dobách. Na webových stránkách se bude vyskytovat velké množství informací, jako např. výsledky zápasů, informace o veřejném bruslení, informace ke krasobruslení a další. Z těchto důvodů je vhodné, aby bylo možné přiřadit novinku do vhodné kategorie. Z těchto důvodů bude zapotřebí i zásuvný modul kategorií.
 - *Administrativní část:* Modul poskytuje možnost editovat kategorie, obsah a vše spojené s aktualitou.
 - *Uživatelská část:* Modul by měl umožňovat uživateli procházet všechny aktuality. Je vhodné zde umožnit filtrování novinek dle kategorií pro jednotlivé sekce webu.
- **Administrativní účty** – O webové stránky se bude starat více uživatelů a je tedy nezbytně nutné, aby na tvorbě obsahu mohlo spolupracovat i více osob.
- **Oprávnění administrativních účtů** – Všichni uživatelé, kterým je poskytnut přístup do administrativní části, nemusí mít hned plná oprávnění. Je nezbytně nutné mít možnost omezit práva administrátora.
- **Správa rozpisu ledové plochy** – Jedná se o zásuvný modul, který umožní spravovat rezervaci ledové plochy. Tento rozpis je tvořen v programu Microsoft Excel. Z tohoto důvodu je vhodné využít některou online aplikaci umožňující tyto dokumenty prohlížet. Vhodným řešením je webová aplikace Google documents, která umožňuje import z programu Microsoft Excel.
- **Nahrávání dokumentů a médií na web**
 - Administrativní část: pro webové stránky je nutností poskytovat různé dokumenty a média. Modul by měl umožnit nahrát potřebné soubory na webové úložiště a poskytnout permanentní odkaz k těmto datům.

- **Plakát přes úvodní stranu**
 - Administrační část: Umožnit správci webu nahrávat plakáty v papírové formě. Dále by měl poskytovat možnost vybrat plakát, který se bude na úvodní straně zobrazovat.
 - Uživatelská část: Modul je schovaný v úvodní straně a zobrazí uživateli preferovaný plakát. Uživatel by měl mít možnost plakát skrýt a nezobrazovat jej při načítání.
- **Fotogalerie a Fotogalerie** - Zvolená organizace má mnoho událostí, které jsou zdokumentovány právě fotografiemi. Z tohoto důvodu je tento modul příhodným pomocníkem k prezentování těchto fotografií. Fotografie by mělo být možné třídit do alb, na která je poté možné odkazovat.
- **Implementovat jednoduchý responzivní HTML design** – Zimní stadion v Rakovníku má připravený a již napsaný responzivní design, který by chtěl na webové stránky implementovat. Implementace připravené šablony nebude činit v připravované aplikaci problém, jelikož bude vzhled oddělen od veškeré aplikační logiky a se změnou designu počítá.

4.2 Návrh webové aplikace

Jako hlavní programovací jazyk jsem zvolil PHP. Z bezpečnostních důvodů jsem pro návrh aplikace využil také Nette Frameworku a databázového layeru dibi. Celý redakční systém jsem vytvořil na doporučené adresářové struktuře pro psaní webových aplikací uvedenou v dokumentaci Nette Frameworku.

4.2.1 Použité technologie

Pro tvorbu webové aplikace jsem využil značkovací jazyk **HTML**, který je výstupním jazykem webových stránek a internetový prohlížeč jej dále zpracovává. Dále tato aplikace využívá **kaskádových stylů**. Pro interakci ve webovém prohlížeči uživatele jsem využil **Javascript**. HTML je výstupem z **PHP** jazyka, který byl využit jako základní stavební kámen celé aplikace. Z bezpečnostních důvodů jsem využil **Nette Framework**, který umožňuje svojí sadou funkcí velmi kvalitní zabezpečení. Pro komunikaci s databází jsem zvolil namísto Nette Database knihovnu **Dibi**.

4.2.2 Adresářová struktura

Adresářová struktura je navržena tak, aby byla jednoznačně oddělena aplikační logika administračního rozhraní od uživatelského rozhraní. V jednotlivých rozhraních je třeba rozdělit části programu podle jednotlivých vrstev model-view-controller. Dále je zapotřebí vytvořit dočasný adresář temp pro dočasné soubory, adresář pro knihovny a framework. Dále založit adresář www, který bude volně dostupný v síti internet – je oddělen od celé aplikace, která není dostupnou ze sítě internet, ale zaváděna z veřejného adresáře. Adresářová struktura by měla vycházet z doporučení v dokumentaci Nette Frameworku.

4.2.3 Databázový model

Databázový model je navržen tak, aby zahrnul veškeré požadavky na redakční systém. Bude tvořen na databázovém systému MySQL a využíváno bude datové úložiště InnoDB, které je navrženo pro zpracovávání velkého množství transakcí. Navrhované schéma v příloze 1 popisuje databázové řešení z pohledu na datové typy a vystihuje tyto části:

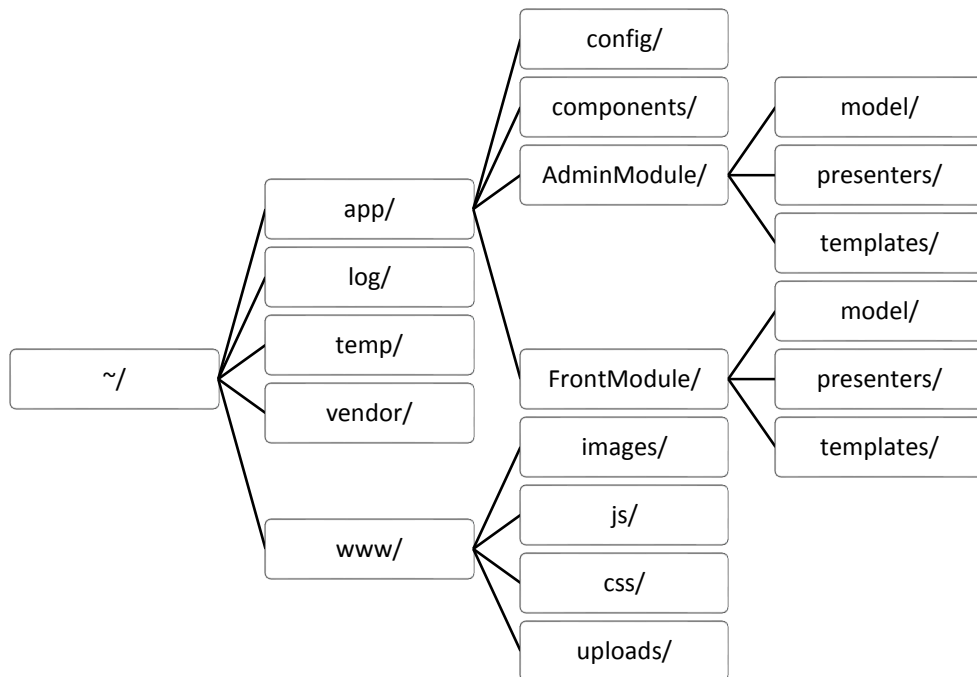
- Uživatelské účty
- Oprávnění uživatelů
- CMS Stránky
- Aktuality
- Kategorie
- Záznamy o aktivitě
- Informace o souborech
- Navigační panely
- Fotoalba a fotografie

4.3 Implementace webové aplikace

4.3.1 Adresářová struktura

- Adresář **www** je veřejným adresářem této aplikace. Obsahuje takzvaný root dokument - index.php, který má za úkol spustit celou aplikaci. Obsahuje také veškeré soubory, jako jsou obrázky, soubory kaskádových stylů, javascriptové knihovny, písma, nahrané soubory a jiná média. Pro tyto soubory jsou vytvořeny složky, které tyto data separují.
- Adresář **log** - Obsahuje veškeré logy, které jsou tvořeny v průběhu životního cyklu spuštěné aplikace. Do tohoto adresáře spadají i záznamy o chybách aplikace, které mohou při běhu nastat. Do tohoto adresáře jsou chyby ukládány pomocí knihovny Tracy. Tracy veškeré chyby ukládá ve výstupních souborech error.log, exception.log. Tracy do tohoto adresáře vytvoří vizualizaci chyby pomocí HTML výstupu. Do tohoto adresáře musí být nastavena práva pro zápis. Pokud nebudou nastavena, aplikace nebude schopna vytvářet logy.
- Adresář **temp** obsahuje veškeré dočasné soubory vytvářené webovou aplikací. V tomto adresáři se vyskytuje například cache. Zde jsou uchovávány například cache šablon, které jsou výstupem Latte z Nette Frameworku. Do tohoto adresáře musí být nastavena práva pro zápis. Pokud nebudou nastavena, aplikace nebude schopna využívat souborové cache a ztratí tím výkon.
- Adresář **vendor** obsahuje všechny závislosti potřebné k chodu aplikace. Může se jednat o knihovny třetích stran. V tomto adresáři nalezneme mimo jiné také Nette Framework a databázovou vrstvu dibi.

- Adresář **app** – obsahuje program celé aplikace a je dále větvený.
 - Adresář **config** – obsahuje konfigurační soubory aplikace. Zde jsou uloženy například přístupové údaje pro přístup do databázového systému a základní konfigurace aplikace v Nette.
 - Adresář **AdminModule** – Jedná se o modul webové stránky, ve kterém je oddělena administrační logika pro správu elektronického obsahu od aplikační logiky pro návštěvníky webových stránek. Tento adresář se dále větví dle softwarové architektury MVC:
 - Adresář **Model** – obsahuje Modelovou vrstvu a jsou v ní soubory s třídami modelů.
 - Adresář **Templates** – obsahuje pohledy. Reálně jsou zde uloženy jednotlivé pohledy v souborech Latte.
 - Adresář **Presenters** – Obsahuje třídy řadičů, které zpracovávají požadavky uživatele aplikace.
 - Adresář **FrontModule** – Jedná se o modul webové stránky, ve kterém je oddělena aplikační logika pro návštěvníky webové stránky od aplikační logiky pro správu obsahu webových stránek. Tento adresář se dále větví dle softwarové architektury MVC:
 - Adresář **Model** – obsahuje Modelovou vrstvu a jsou v ní soubory s třídami modelů.
 - Adresář **Templates** – obsahuje pohledy. Reálně jsou zde uloženy jednotlivé pohledy v souborech Latte.
 - Adresář **Presenters** – Obsahuje třídy řadičů, které zpracovávají požadavky uživatele aplikace.



Obrázek 1: Adresářová struktura aplikace

Takto navržená adresářová struktura umožňuje využití softwarové architektury Model – View – Controller.

4.3.2 Databáze

Databáze v implementačním kroku naroste, jelikož musí zahrnout vazby, které mezi jednotlivými částmi z návrhu vznikají.

4.3.2.1 Uživatelské účty

Uživatelské účty se skládají z jedné hlavní tabulky, která nese informace o uživateli, jako je jeho identifikátor (ID) jako primární klíč, kontaktní emailová adresa, uživatelské jméno, heslo, informace o posledním přihlášení a mimo jiné i informaci o tom, zda byl uživatel smazán či nikoliv.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
email	varchar(254)	emailová adresa
username	varchar(32)	uživatelské jméno
password	varchar(32)	heslo MD5
last_login	int(11)	čas posledního přihlášení
last_ip	varchar(45)	IPv6 support
login_count	int(11)	počet přihlášení
deleted	smallint(6) [0]	smazaný?

Tabulka 3: Databázová tabulka s uživateli

4.3.2.2 Oprávnění uživatelů

Mimo primární klíč obsahuje vazbu na uživatele za pomoci cizího klíče na zásuvný modul a také na vlajku s příslušným typem oprávnění. Implementace této části se dále větví do tabulek se zdroji a jednotlivými oprávněními – flagy. Tato tabulka má za úkol vázat oprávnění na zdroje a tuto kombinaci vázat na jednotlivé uživatele.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
users_id	int(11)	vazba na uživatele
resources_id	int(11)	vazba na zdroj
flags_id	int(11)	vazba na oprávnění
disallowed	smallint(1) [0]	restrikce specifického práva

Tabulka 4: Databázová tabulka s oprávněním uživatelů

4.3.2.2.1 Zdroje

Jedná se o tabulku, která určuje klíčové slovo zdroje včetně popisu. V této tabulce je primární klíč identifikátor zdroje, na který odkazují jednotlivé oprávnění uživatele. Zdroj nemá poněti o jednotlivých oprávněních, která jsou k dispozici.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
name	varchar(32)	klíč zdroje
description	varchar(256)	popis zdroje

Tabulka 5: Databázová tabulka se zdroji / zásuvnými moduly

4.3.2.2.2 Oprávnění

Oprávnění jsou specifikována za pomoci klíčových slov CREATE – vytváření, EDIT – editace, DELETE – mazání a ACCESS pro oprávnění přístupu ke zdroji. Existuje také zástupné oprávnění ALL, které má za následek udělení veškerého oprávnění k danému zdroji. Oprávnění nemá tušení o existenci o zdrojích, které jsou v redakčním systému obsaženy.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
name	varchar(32)	název vlajky / oprávnění
description	varchar(64)	Popis oprávnění

Tabulka 6: Databázová tabulka s druhy oprávnění

4.3.2.3 CMS Stránky

CMS Stránky obsahují primární klíč, podle kterého aplikace jednoznačně identifikuje cílenou stránku. Dále obsahuje informace jako nadpis, obsah celé stránky, klíčová slova, popis a vlastníka. Dále jsou zde uvedeny 2 domény, které umožňují zamezit smazání či editaci specifické stránky i přes oprávnění uživatelů.

Sloupec	Typ	Komentář
id	int(11) Auto Increment	
title	varchar(255)	nadpis CMS Stránky
content	longtext	Obsah CMS Stránky
keywords	text	Klíčová slova SEO
description	text	popis SEO
deleteable	tinyint(1) [1]	smazatelné?
editable	tinyint(1) [1]	editovatelné?
users_id	int(11)	odkaz na vlastníka

Tabulka 7: Databázová tabulka s CMS stránkami

4.3.2.4 Aktuality

Aktuality nesou primární klíč ID, který ji jednoznačně identifikuje. Dále obsahuje domény jako titulek aktuality, obsah aktuality, její viditelnost na úvodní straně a čas s datem, ke kterému se aktualita vztahuje.

Sloupec	Typ	Komentář
id	int(11) Auto Increment	
users_id	int(11)	odkaz na vlastníka
time	int(11)	čas, ke kterému se aktualita vztahuje
title	varchar(128)	nadpis aktuality
content	longtext	obsah celé aktuality
visible	tinyint(1) [1]	viditelnost na úvodní straně

Tabulka 8: Databázová tabulka s aktualitami

4.3.2.5 Kategorie

Kategorie nesou v databázi primární klíč ID za účelem identifikace. Dále nese název a informaci o tom, zda je kategorie smazatelná.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
name	varchar(64)	název kategorie
deleteable	tinyint(1) [1]	je smazatelná?

Tabulka 9: Databázová tabulka s kategoriemi

Mezi kategoriemi a aktualitami je vytvořena vazební tabulka, která říká, do kterých kategorií daná aktualita spadá.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
news_id	int(11)	vazba na novinku
categories_id	int(11)	vazba na kategorii

Tabulka 10: Databázová vazební tabulka mezi novinkami a kategoriemi

4.3.2.6 Záznamy o aktivitě

Záznamy o aktivitě nesou jako primární klíč identifikátor dané aktivity. Tento identifikátor je unikátní. Dále nese informace jako úroveň záznamu ve formě čísla, kde nízké číslo znamená nezávažnou změnu a s rostoucím číslem roste i závažnost záznamu. Dalšími sloupci této tabulky je vazba na uživatele, který záznam vyvolal, a detail pro určení specifického záznamu daného zdroje, ke kterému se vztahuje. Také nese zprávu záznamu této aktivity.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
level	smallint(6)	úroveň záznamu
time	int(11)	čas záznamu
users_id	int(11) <i>NULL</i>	odkaz na uživatele
resources_id	int(11)	odkaz na zdroj
detailid	int(11) <i>NULL</i>	detail vyjádřená číslem
message	text	zpráva samotného záznamu

Tabulka 11: Databázová tabulka se záznamy o aktivitě

4.3.2.7 Informace o souborech

V této tabulce bude obsažena kromě identifikátoru souboru také cesta k souboru, vlastník souboru a čas, kdy byl soubor nahrán.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
file_name	varchar(64)	název souboru / cesta
users_id	int(11)	Odkaz na vlastníka
time	int(11)	čas nahrání

Tabulka 12: Databázová tabulka s informacemi o souborech

4.3.2.8 Navigační panely

Navigační panely jsou děleny do 2 tabulek, kde jsou neneseny informace o jednotlivých navigačních panelech a jejich položkách. Každý navigační panel má mimo svého názvu i svůj identifikátor, na který odkazují jednotlivé položky. Navigačnímu panelu lze určit, jakým způsobem se má výsledný panel ve formě HTML vykreslovat. K tomu slouží domény, ve kterých je možné specifikovat, jak výsledný výstup panelu bude vypadat.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
name	varchar(128)	název navigace
editable	tinyint(1) [1]	je editovatelný?
root_ul_opening_tag	varchar(128)	otevírací tag celé navigace
ul_closing_tag	varchar(128)	zavírací tag celé navigace
child_ul_opening_tag	varchar(128)	otevírací tag podnavigace
li_opening_tag	varchar(128)	otevírací tag položky seznamu bez podnavigace
li_dropdown_opening_tag	varchar(128)	otevírací tag položky seznamu s podnavigací
li_closing_tag	varchar(128)	zavírací tag položky seznamu
a_opening_tag	varchar(128)	otevírací tag pro hypertextový odkaz bez podnavigace
a_dropdown_opening_tag	varchar(128)	otevírací tag pro hypertextový odkaz s podnavigací
a_closing_tag	varchar(128)	zavírací tag hypertextového odkazu

Tabulka 13: Databázová tabulka s informacemi o navigačních panelech

Položky v jednotlivých navigačních panelech jsou odděleny do zvláštní tabulky. V této tabulce je obsažena identifikace položky, odkaz na navigační panel, identifikátor svého předka (své předešlé položky) a pozici. Dále obsahuje domény pro hypertextový odkaz a popis položky.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
navigation_id	int(11)	odkaz na navigaci
parent_id	int(11)	odkaz na svého předka
pos_id	int(11)	pozice v navigaci
href	varchar(256)	hypertextový odkaz (URL)
content	varchar(256)	text hypertextového odkazu

Tabulka 14: Databázová tabulka s položkami v navigačních panelech

4.3.2.9 Fotoalba a fotografie

Fotografie jsou rozděleny do alb. I toto řeší datová struktura v databázi. Fotografie jsou tedy uloženy do 2 tabulek, kde první reprezentuje album a druhá uložené fotografie.

Alba nesou pouze vlastní identifikátor, název a svého majitele. Do alba se poté vážou jednotlivé fotografie.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
name	varchar(64)	Název alba
users_id	int(11)	Odkaz na vlastníka

Tabulka 15: Databázová tabulka s fotoalby

Fotografie pak nesou informace o jejich fyzickém uložení, o majiteli, odkaz na příslušné album, čas, kdy byla fotografie nahrána, a její meta data jako název a popis.

Sloupec	Typ	Komentář
id	int(11) <i>Auto Increment</i>	
albums_id	int(11)	Odkaz na fotoalbum
users_id	int(11)	Odkaz na vlastníka
file_name	varchar(64)	Název souboru / cesta
time	int(11)	čas nahrání
title	varchar(64) <i>NULL</i>	titulek fotografie
description	text <i>NULL</i>	popis fotografie

Tabulka 16: Databázová tabulka s fotografiemi

4.3.3 Software třetích stran

Z podrobné analýzy a následného návrhu vznikl seznam potřebných technologií k implementaci celého projektu. Mezi tyto technologie byl zahrnut i software třetích stran, který je využíván k tvorbě webových aplikací. Tento software musí být dodán společně s celou aplikací, a tak je třeba jej implementovat do celého řešení. Pro zvolený software je zaimplementován speciální adresář vendor. Na software třetích stran pak odkazuje zaváděcí soubor app/bootstrap.php, který je spuštěn root dokumentem index.php v adresáři WWW. Bootstrap má za úkol obstarat načtení všech knihoven a závislostí potřebných k běhu celé aplikace, jako je Nette Framework a databázový abstraktní layer dibi.

4.3.4 Zásuvné moduly

Zásuvné moduly jsou silnou zbraní této aplikace. Implementováním tohoto návrhu je možné systém v budoucnu bez problémů rozšiřovat.

Zásuvné moduly budou implementovány vždy jako Model, Presenter a Template do příslušného rozraní a dodržovat životní cyklus. Životní cyklus je sekvence následujících událostí:

- **Startup** – zde se provádí veškeré nastavení zásuvného modulu při jeho zapnutí. Modul je zapnut pouze tehdy, když je vyžadován
- **Action** – Vyvolání akce, která provede nějaký úkon. Jde o poslední část životního cyklu aplikace, ve které lze přeměřovat uživatele. Akce nemá za úkol cokoliv vykreslovat. Nevyžaduje vytvořený pohled v adresáři templates.
- **Handle** – slouží k zachycení a zpracování signálů, které jsou do aplikace z většiny případů zaslány AJAXem. AJAX má za úkol aplikaci zaslat požadavek na pozadí.
- **BeforeRender** – slouží k nastavení šablony webové aplikace. Tato událost bývá použita na místech, kde probíhá předání společných proměnných do více pohledů.
- **Render** – V této části probíhá plnění dat šablony. Po této události se na výstup aplikace vypíše výsledný zdrojový kód ve formě HTML, CSS a JS.
- **Shutdown** – Poslední část životního cyklu. [24]

V Redakčním systému jsou v základu implementovány následující moduly:

- **CMS stránky**
 - Administrační část: Modul bude poskytovat přístup k datům v databázi a umožní jejich snadnou správu – Přehled všech stránek, přidávání, editování či mazání. Veškeré úkony jsou vázány a ukládány na záznamy o aktivitě.
 - Uživatelská část: Modul umožní zobrazení obsahu stránky na základě požadavku. Požadavek musí obsahovat minimálně identifikátor požadované stránky

- **Záznamy o aktivitě**
 - Administrační část: Poskytuje pouze pohled do tabulky se záznamy.
 - Uživatelská část: V uživatelské části bude modul poskytovat adaptér k napojení na analýzu návštěvnosti. Tato část bude zajištěna pomocí externí aplikace třetí strany. Zásuvný modul bude obsahovat adaptér pro napojení na službu google analytics společnosti Google.
- **Navigační panely**
 - Administrační část: Administrační modul poskytne přehled všech panelů, které jsou v databázi uloženy, a umožní správu jednotlivých položek, které na panel odkazují. Mimo to poskytne také možnost za pomoci grafického rozhraní editovat strom navigace a tím určovat podpoložky.
 - Uživatelská část: Zásuvný modul by měl být schopný poskytnout vykreslení vyžádané navigace za pomoci identifikátoru a bude se držet předlohy jak tuto navigaci vykreslit. Tato předloha je obsahem databáze.
- **Aktuality**
 - Administrační část: Zásuvný modul aktualit bude poskytovat plnou správu dat. Mimo jiné bude umožňovat zařazení aktuality do 0 až nekonečně mnoho kategorií.
 - Uživatelská část: Modul uživateli poskytne pohled na všechny novinky, které mohou být na úvodní stránce stránkovány. Dále umožní tyto aktuality filtrovat dle zvolených kategorií.

- **Kategorie**
 - Administrační část: Modul poskytuje správu jednotlivých kategorií. Spravovat je možné pouze její název. Do kategorie se přiřazují články ve správě samotných aktualit.
 - Uživatelská část: Slouží pouze jako filtr v aktualitách a nejsou nijak samostatně zobrazovány v uživatelském rozhraní.
- **Administrační účty**
 - Administrační část: Obsahuje správu jednotlivých administračních účtů základními metodami jako přehled, přidávání, editace či mazání.
- **Oprávnění administračních účtů**
 - Administrační část: Modul umožní spravovat práva jednotlivých uživatelů pro různé zdroje – zásuvné moduly.
- **Nahrávání dokumentů a médií na web**
 - Administrační část: Modul poskytuje možnost nahrávat a spravovat soubory ve webovém úložišti. K těmto souborům poskytne i URL adresu, na kterou je v případě potřeby možné odkazovat. Tyto soubory budou ukládány do adresáře veřejného adresáře uploads.
- **Plakát přes úvodní stranu**
 - Administrační část: Základní modul nabídne nahrávání obrázků jako plakátů, u kterých je mimo cesty možnost zvolit, zda je aktivní, či nikoliv.
 - Uživatelská část: Na úvodní straně poskytne náhled plakátu, který je v administračním rozhraní určen zrovna jako aktivní. Modul by měl poskytnout uživateli možnost zobrazování vypnout a to do doby, než bude plakát změněn.

- **Fotogalerie a Fotogalerie -**
 - Administrační část: Modul je rozdělen na 2 části, kde v první je správa samotných alb v plném rozsahu. Tato správa zahrnuje možnost nahrát do fotogalerie nové fotky, ale také nahradit staré za nové. Jednotlivým fotografiím pak poskytuje rozhraní, ve kterém je možnost přidat metainformace jako popisek fotky či její název.
 - Uživatelská část: V uživatelské části bude poskytnut pohled, jehož úkolem je zobrazit všechna alba v systému vytvořená spolu s náhledem prvních 3 fotografií. Dále bude poskytnut pohled do jednotlivých fotoalb zahrnující náhledy fotografií a jejich popisky či názvy.

4.4 Testování webové aplikace

Testování celé aplikace bylo rozděleno na 2 vlákna. Na testování se podíleli lidé, které si HC Rakovník, z.s. sami zvolili. Software byl otestován pokládáním úkolů, které testeři museli plnit. Výsledek testování mi pak předali k doladění nedostatků, které byly objeveny.

Administrační rozhraní bylo testováno personálem, který byl pověřen správou webových stránek. Jednalo se o 5 správců, jejichž znalosti z oboru tvorby webových stránek měly různou úroveň. Díky různým úrovním znalosti problematiky se nám podařilo odhalit některé problémy, které by v budoucnu objevily. Největším odhaleným problémem byl fakt, že omezení práv bylo velmi obecné, a tak uživatel A s oprávněním mazat data mohl mazat data uživatele B a naopak. Tato problematika byla vyřešena omezením práva mazat - pouze vlastní příspěvky. Pro hlavního správce webových stránek je pak uděláno právo Super uživatele, který může dělat vše, co jednotlivé zásuvné moduly umožňují, bez omezení na vlastníka.

Uživatelské prostředí pro budoucí návštěvníky webu bylo testováno členy jednotlivých klubů na zimním stadionu. Zde byl objeven pouze problém s postranním panelem, který má za úkol zobrazovat sponzory zimního stadionu. Jeho načítání na mobilních zařízeních bylo shledáno jako zbytečné a navyšoval se tím objem dat potřebný pro načtení webu.

4.5 Nasazení webové aplikace do provozu

Po otestování celé aplikace následuje krok nasazení, v němž se celý redakční systém překlopí do produkční verze, která je označena v tomto momentě jako stabilní. Systém byl nainstalován na webové úložiště organizace, která systém poptávala. Instalace byla rozdělena do několika kroků. Prvním krokem byla příprava prostředí, ve kterém bude aplikace uložena. Následně proběhla instalace datové struktury a poté i dat do MySQL databáze za pomoci sekvence SQL příkazů. Dalším krokem byla konfigurace celé aplikace v adresáři config, kde jsou uvedeny přístupové údaje do databáze. Celá aplikace byla nakopírována do webového úložiště, a tím byla celá nainstalována a připravena na prvního návštěvníka.

5 Zhodnocení výsledku

Zimní stadion v Rakovníku webovou aplikaci využívá od října 2015. Využíváním administračního rozhraní se s ním všichni správci seznámili a spřátelili. Během používání se narazilo na některá vylepšení, která byla později dopsána. Výsledkem je kompletní webová aplikace, která umožňuje zimnímu stadionu poskytnout přesně to, co potřebuje. Během vývoje aplikace se dodržely veškeré požadavky na redakční systém, které byly na začátku celého projektu zadány. Nicméně estetický výstup dat, která jsou v systému zpracovávána, není ideální. Vzhledem k WYSIWYG editoru a funkcím, které organizace vyžadovala, je možné narušit webovou prezentaci po vizuální stránce. Tato skutečnost zapříčinila nadměrné využívání barevných kombinací na webové stránce a stránky jsou po vizuální stránce nekvalitně zpracovány.

Výsledek nového informačního systému pro návštěvníky webových stránek zimního stadionu se ukázal okamžitě a provozovatel webových stránek zaznamenal velký posun kupředu. Návštěvníci webových stránek se naučili na webových stránkách vyhledávat potřebné informace velice rychle. Nasazením produktu této bakalářské práce se na organizaci snížil počet telefonátů vyžadujících informace, které byly dostupné online.

Porovnáním dat z Google Analytics za období 1. listopadu 2015 až 31. ledna 2016 oproti datům z období 1. listopadu 2014 až 31. ledna 2015 jsem zjistil zvýšenou návštěvnost webu o 6.01%. Návštěvnost z hledání na vyhledávacích vzrostla o 3.56%. V neposlední řadě se tato změna projevila i na zvýšení návštěvnosti bez okamžitého opuštění a to o 12.08%. Celková návštěvnost zůstala zachována. Podrobný výstup z Google Analytics je poskytnut jako příloha 2 a příloha 3 této práce.

Provedl jsem také výzkum zaměřený na finanční úsporu. Zaslal jsem požadavky na redakční systém do profesionálních společností s požadavkem vytvoření cenové kalkulace takovéto aplikace. Ze získaných odpovědí jsem dospěl k cenovému rozmezí 45000 – 85000 Kč.

Závěr

Webová aplikace, jejíž vytvoření bylo cílem této práce, byla vytvořena a nasazena v reálném podniku. Byla při tom využita sada svobodného softwaru a kombinace zvolených technologií na základě podrobné analýzy HTML, CSS, Javascriptu, PHP, MySQL a knihoven Dibi (databázové vrstvy) a Nette Frameworku. Celá aplikace je psána za pomoci objektově orientovaného programování. Při psaní aplikace jsem se držel softwarové architektury MVC, která pomohla k řešení modularity celého redakčního systému. Napsaný software byl nasazen na webové adrese Zimního stadionu v Rakovníku – HC Rakovník z.s. a je využívána. Webové stránky, na které je aplikace nasazena, zaznamenaly úspěch.

Zimnímu stadionu aplikace ulehčila práci s daty na webových stránkách a zajistila sjednocení všech podstránek, které byly předtím rozesety na webových stránkách jednotlivých oddílů. Nabízelo se využít redakční systémy s otevřeným zdrojovým kódem, ale to si ve zvolené organizaci nepřáli a na vlastním řešení trvali. Vhodným otevřeným softwarem je wordpress.

Seznam použitých zdrojů

- [1] PROCHÁZKA, David. *PHP 6: začínáme programovat*. 1. vyd. Praha: Grada, 2012. Průvodce (Grada). ISBN 978-80-247-3899-4.
- [2] *Redakční systém realizovaný pomocí PHP a MySQL*. Jihočeská univerzita v Českých Budějovicích, 2007. Bakalářská práce. Jihočeská univerzita v Českých Budějovicích. Vedoucí práce Ing. Michal Šerý.
- [3] KUROSE, James F a Keith W ROSS. *Počítačové sítě*. 1. vyd. Brno: Computer Press, 2014. ISBN 978-80-251-3825-0.
- [4] Prohlížeče a internetové technologie. In: *Jak na Internet* [online]. © 2016 [cit. 2016-03-06]. Dostupné z: <http://www.jaknainternet.cz/page/1235/prohlizece-a-internetove-technologie/>
- [5] MIKULÁK, Martin. *Programujeme WWW stránky pro úplné začátečníky*. Vyd. 1. Brno: Computer Press, 2011. Pro úplné začátečníky. ISBN 978-80-251-3252-4.
- [6] AULDS, Charles. *Linux - administrace serveru Apache*. 1. vyd. Praha: Grada, 2003. Profesional. ISBN 80-247-0640-7.
- [7] NEDELCOU, Clément. *Nginx HTTP server adopt Nginx for your web applications to make the most of your infrastructure and serve pages faster than ever*. Birmingham, U.K: Packt Pub, 2010. ISBN 978-184-9510-875.
- [8] *Internet Information Services (IIS) 7.0 Administrator's Pocket Consultant*. Sebastopol: Microsoft Press, 2010. ISBN 978-073-5645-899.
- [9] VOSTROVSKÝ, Václav. *Vytváření databází v ORACLE*. Vyd. 1. Praha: Česká zemědělská univerzita v Praze, Provozně ekonomická fakulta, 2004. ISBN 80-213-1191-6.

- [10] DUBOIS, Paul. *MySQL profesionálně: kompletní průvodce použitím, programováním a správou MySQL*. Praha: Mobil Media, c2003. iDnes internet knihy. ISBN 80-865-9341-X.
- [11] POKORNÝ, Jaroslav. *Dotazovací jazyky*. 1. vyd. Praha: Karolinum, 2002. ISBN 978-80-246-0497-8.
- [12] Introduction to HTML. *W3Schools Online Web Tutorials* [online]. © 1999-2016 [cit. 2016-03-06]. Dostupné z: http://www.w3schools.com/html/html_intro.asp
- [13] HUDDLESTON, Rob. *XML: your visual blueprint for building expert Web sites with XML, CSS, XHTML, and XSLT*. Hoboken, N.J.: John Wiley [distributor], c2008. ISBN 978-047-1933-830.
- [14] CSS Tutorial. *W3Schools Online Web Tutorials* [online]. © 1999-2016 [cit. 2016-03-09]. Dostupné z: <http://www.w3schools.com/css/>
- [15] JavaScript Tutorial. *W3Schools Online Web Tutorials* [online]. © 1999-2016 [cit. 2016-03-06]. Dostupné z: <http://www.w3schools.com/js/>
- [16] *PHP: Hypertext Preprocessor* [online]. The PHP Group, © 2001-2016 [cit. 2016-03-06]. Dostupné z: <http://www.php.net/>
- [17] MACH, Jakub. *PHP pro úplné začátečníky*. 2. přeprac. a rozš. vyd. Brno: Computer Press, 2003. ISBN 80-722-6834-1.
- [18] SPAANJAARS, Imar. *Beginning asp.net 4.5.1: in c# and vb*. 1st edition. Indianapolis, IN: John Wiley and Sons, 2014. ISBN 11-188-4677-X.
- [19] *Ruby Programming Language* [online]. 2016 [cit. 2016-03-06]. Dostupné z: <https://www.ruby-lang.org/en/>

- [20] Introduction To the Oracle Database. *Oracle* [online]. Oracle, © 2016 [cit. 2016-03-09]. Dostupné z: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm
- [21] About. *PostgreSQL* [online]. The PostgreSQL Global Development Group, © 1996-2016 [cit. 2016-03-09]. Dostupné z: <http://www.postgresql.org/about/>
- [22] About MariaDB. *MariaDB.org* [online]. MariaDB Foundation, © 2016 [cit. 2016-03-09]. Dostupné z: <https://mariadb.org/about/>
- [23] About SQLite. *SQLite* [online]. 2016 [cit. 2016-03-09]. Dostupné z: <https://www.sqlite.org/about.html>
- [24] MVC Aplikace & Presentery. *Nette Framework* [online]. Nette Foundation, 2016 [cit. 2016-03-09]. Dostupné z: <https://doc.nette.org/cs/2.3/presenters>
- [25] GRUDL, David. Nette Framework: zvyšte svoji produktivitu. In: *Zdroják* [online]. 2009 [cit. 2016-03-09]. Dostupné z: <https://www.zdrojak.cz/clanky/nette-framework-zvyste-svoji-produktivitu/>
- [26] Dibi is Database Abstraction Library for PHP 5. *Dibi* [online]. Nette Foundation, 2016 [cit. 2016-03-09]. Dostupné z: <https://dibiphp.com/cs/>
- [27] Quick Start. *Dibi* [online]. Nette Foundation, 2016 [cit. 2016-03-09]. Dostupné z: <https://dibiphp.com/cs/quick-start>
- [28] Zend Framework. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2016-03-09]. Dostupné z: https://cs.wikipedia.org/wiki/Zend_Framework
- [29] *RubyOnRails.cz* [online]. 2016 [cit. 2016-03-09]. Dostupné z: <http://rubyonrails.cz/>

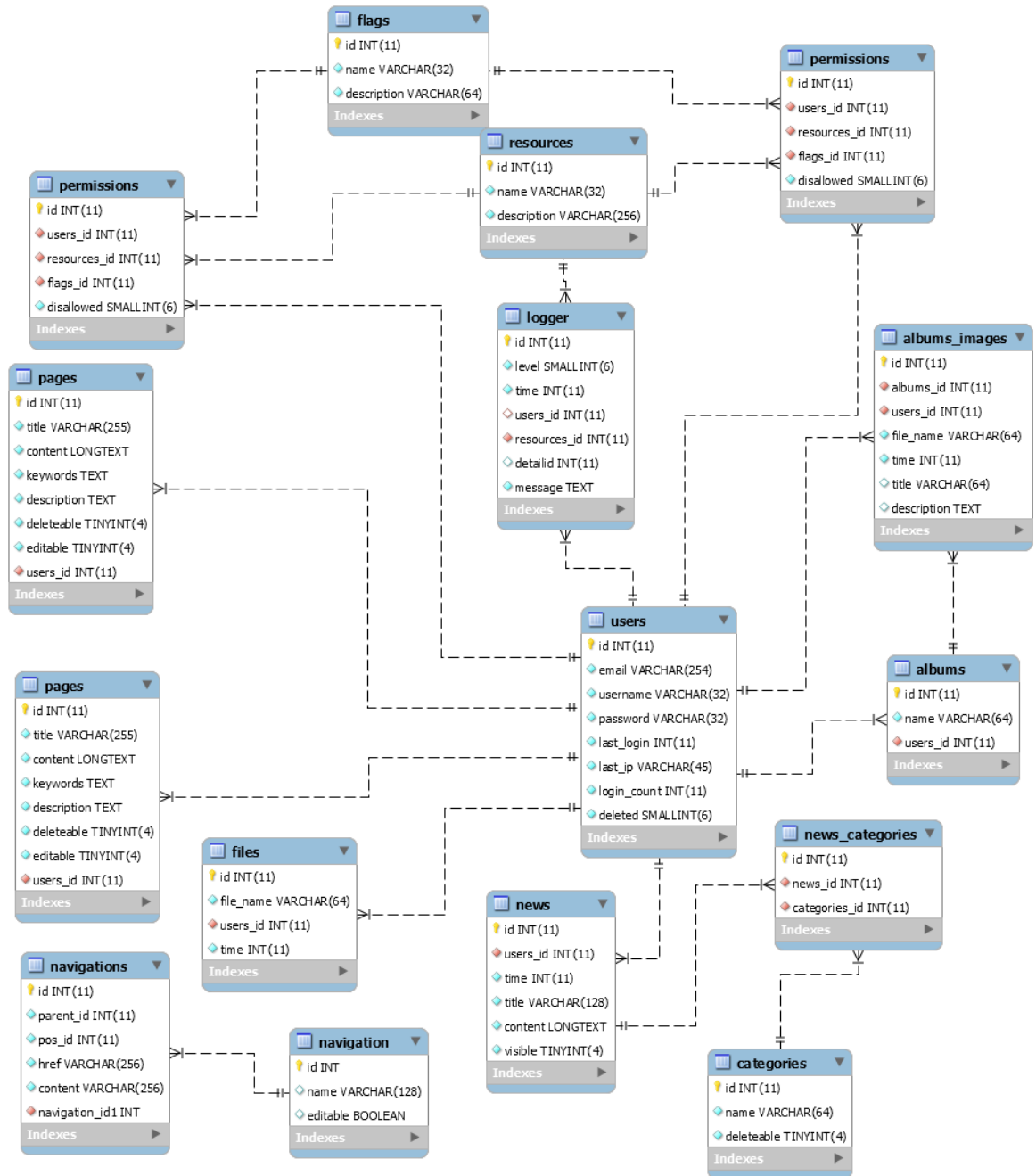
- [30] CHAFFER, Jonathan a Karl SWEDBERG. *Mistrovství v jQuery: [kompletní průvodce vývojáře]*. 1. vyd. Brno: Computer Press, 2013. Mistrovství. ISBN 978-80-251-4103-8.
- [31] *Wordpress portál: vše o redakčním systému zdarma* [online]. 2016 [cit. 2016-03-09]. Dostupné z: <http://www.cwordpress.cz/>
- [32] BALTER, Ben. Open source license usage on GitHub.com. In: *GitHub* [online]. GitHub, Inc, 2016 [cit. 2016-03-09]. Dostupné z: <https://github.com/blog/1964-open-source-license-usage-on-github-com>
- [33] PERGL, Robert. *Metody řízení softwarových projektů využívající moderní paradigmaty*. Praha, 2008. Disertační práce. Česká zemědělská univerzita v Praze. Vedoucí práce Prof. Ing. Ivan Vrana, DrSc.

Seznam použitých zkratek

Zkratka	Význam
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
JS	Javascript
PHP	PHP: Hypertext preprocesor
FW	Framework
WWW	World Wide Web
HTTP	Hypertext transfer protocol
SQL	Structured Query Language
OOP	Objektově orientované programování
DB	Databáze

Přílohy

Příloha 1: EER Diagram databázového modelu.

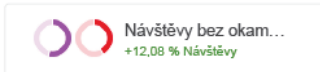
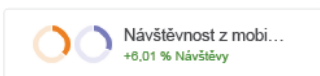
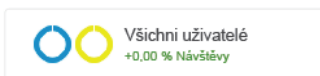


Příloha 2: Google Analytics – shrnující údaje

Přehled publika

1. 10. 2015 - 31. 1. 2016

Porovnat s hodnotou: 1. 10. 2014 - 31. 1. 2015



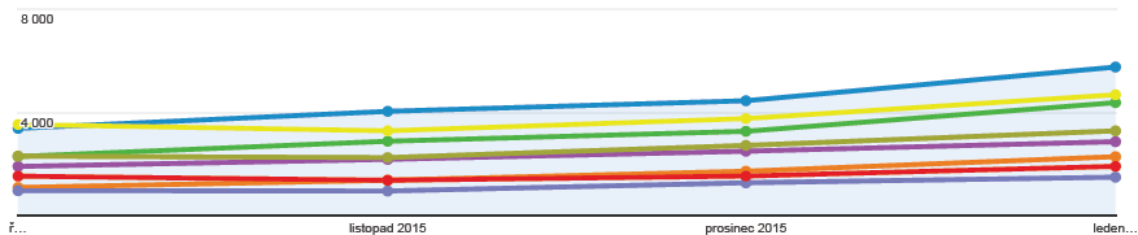
Přehled

1.10.2015 - 31.1.2016:

- Návštěvy (Všichni uživatelé)
- Návštěvy (Návštěvnost z mobilů a tabletů)
- Návštěvy (Návštěvnost z hledání)
- Návštěvy (Návštěvy bez okamžitého opuštění)

1.10.2014 - 31.1.2015:

- Návštěvy (Všichni uživatelé)
- Návštěvy (Návštěvnost z mobilů a tabletů)
- Návštěvy (Návštěvnost z hledání)
- Návštěvy (Návštěvy bez okamžitého opuštění)

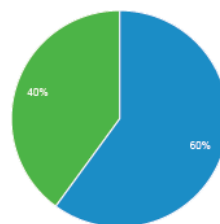
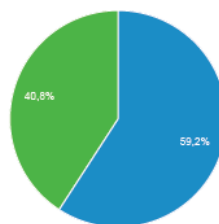


■ Returning Visitor ■ New Visitor

1.10.2015 - 31.1.2016

Všichni uživatelé

Návštěvnost z mobilů a tabletů



Príloha 3: Google Analytics – podrobné prehľady

