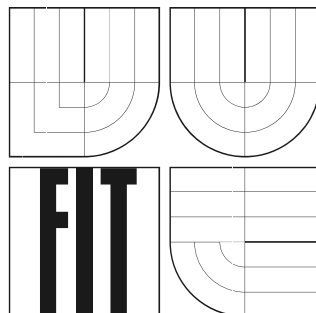


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dynamicky rekonfigurovatelný web

Semestrální projekt

Dynamicky rekonfigurovatelný web

© Radek Preuss, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Prohlášení

Prohlašuji tímto, že jsem tento semestrální projekt vypracoval samostatně pod vedením Ing. Radka Burgeta, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal. A to včetně internetových portálů zabývajících se danou problematikou.

.....
Radek Preuss
3. ledna 2007

Abstrakt

Obsahem práce je návrh webové aplikace, umožňující vkládat a editovat dokumenty. Systém je specifický tím, že umožňuje provádět správu některých činností díky grafickému rozhraní. Dále také tím, že umožňuje šablonování dokumentů. Zpráva pojednává o důvodech a metodách dynamického generování obsahu webu. Dále zpráva pojednává také o grafických uživatelských rozhraních, o důvodech jejich používání, také o metodách jaké se v dnešní době mohou použít. Následující část práce je věnována šablonování a metodám použitelných pro šablonování. Hlavní částí zprávy je návrh vlastní webové aplikace. Je zde zmíněna kompletní struktura, každá z částí je detailně rozepsána a jsou uvedeny problémy, které je nutné při implementaci zohlednit. Také je zde popis grafického rozhraní včetně funkcí, které umožňuje. Zpráva taktéž obsahuje databázový návrh.

Klíčová slova

Dynamicky Generované stránky, grafická rozhraní, šablonování, webová aplikace, PHP, MySQL, Flash.

Poděkování

Rád bych na tomto místě poděkoval, svému vedoucímu Ing. Radku Burgetovi, Ph.D., za odborný dohled nad semestrálním projektem. Také bych rád poděkoval panu Jiřímu Koskovi a všem lidem, kteří se sdružují okolo PHP a MySQL, za informace zveřejněné na jejich internetových stránkách uvedených v seznamu literatury jako [1],[2],[3].

Abstract

The content of this document is design of web application, which allows inserting and editing documents. Specificity of this system is graphical user interface to make some operations easily. The system also supports using of the templates. This document discusses reasons and methods of dynamic generated web content. Next part of document discusses graphical user interfaces, reasons of their using and methods using at present day. Another part is about methods of working with templates. The design of web application is a main part of this document. There is complete structure and detailed description of several sections. Also there are problems which must be handled at implementation. This document also describes graphical user interface with their functions and database scheme.

Keywords

Dynamic generated web content, graphical user interface, templates, web application, PHP, MySQL, Flash.

Obsah

Obsah	5
1 Úvod.....	7
2 Dynamicky generované stránky.....	8
2.1 Proč generovat dynamicky	8
2.2 Metody používané při generování stránek	8
2.2.1 CGI skripty	8
2.2.2 Active Server Pages (ASP).....	9
2.2.3 Hypertextový preprocesor PHP	9
2.2.4 Technologie s použitím Javy.....	9
2.3 Technologie použita v projektu.....	10
3 Grafická uživatelská rozhraní	11
3.1 Důvody grafického rozhraní.....	11
3.2 Základní principy	11
3.2.1 DHTML a JavaScript.....	11
3.2.2 Statické obrázky.....	11
3.2.3 Java Aplety	12
3.2.4 ActiveX.....	12
3.2.5 Flash.....	12
3.3 Zvolená technologie	13
4 Šablonování.....	14
4.1 Důvody k šablonování.....	14
4.2 Metody šablon	14
4.2.1 Statické šablony	14
4.2.2 HTML šablony s využitím CSS.....	14
4.2.3 Šablony s pseudo-značkami	15
4.3 Zvolená varianta.....	15
4.4 Šablony a jejich součásti	15

5	Návrh webové aplikace.....	17
5.1	Struktura webové aplikace	17
5.1.1	Administrátorská část.....	17
5.1.2	Uživatelská část	18
5.2	Databázový návrh.....	19
5.2.1	Co a proč ukládat do databáze	19
5.2.2	Formát ukládaných dat.....	20
5.3	Grafické uživatelské rozhraní.....	21
5.3.1	Administrace článků a vazeb mezi nimi	21
5.3.2	Administrace kategorií.....	22
5.4	Další uživatelská rozhraní	23
5.4.1	Administrace šablon.....	23
5.4.2	Uživatelská část systému	24
6	Závěr	26
	Literatura	27
	Příloha A : Diagram případů použití	28
	Příloha B : Databázový model.....	29

1 Úvod

Jelikož se blíží konec mého studia na fakultě, byl jsem postaven před problém vybrat si téma Diplomové práce. Jelikož se zajímám o programování webových stránek a aplikací ohlížel jsem se po podobném tématu. Zaujalo mě téma „Dynamicky rekonfigurovatelný web“. Po konzultaci s vedoucím tohoto projektu jsem si jej vybral, hlavně protože se nejedná o klasický redakční systém, ale o systém spoléhající se převážně na grafické rozhraní, které má správci webu umožnit jednoduše a srozumitelně měnit kdykoliv provázání jednotlivých článků a tudíž i strukturu celého webu.

Ovšem toto všechno samozřejmě je v pozadí obsluhováno nějakou technologií pro práci s dynamickými internetovými stránkami. A všechny informace, které se mění jsou uloženy v databázi, odkud se dají snadno zjistit a také modifikovat.

Tento semestrální projekt pojednává o návrhu specifického informačního systému, určeného primárně pro zákazníky, kteří mají potřebu jednou za čas změnit celou strukturu webu a vyžadují proto jednoduché a intuitivní grafické rozhraní.

Tato zpráva pojednává nejprve obecně o technologiích pro dynamickou tvorbu stránek a grafických uživatelských rozhraní na webu. Popisuje základní metody používané více, či méně v praxi. Hodnotí je z několika hledisek a nakonec zdůvodňuje vybranou metodu.

Následně ve zprávě popisuji vhodný systém šablon dokumentů. Jelikož systém umožňuje pro snadnou změnu zobrazovaného stylu používání šablon k určení grafického rozhraní webu. Jsou zde zmíněny návrhy, které mě ohledně šablonování napadly i s jejich vlastnostmi. Taktéž je zde rozebrána varianta pro kterou jsem se rozhodl i s jejími praktickými podrobnostmi, nutnými pro implementaci.

V další části zprávy je návrh webové aplikace dle specifických požadavků, včetně principu ukládání do databáze. Je zde zmíněno rozdělení celé aplikace a jací uživatelé mají k systému přístup. Také je zde uvedeno jaké činnosti může kdo v systému vykonávat, a jaké problémy mohou při práci nastat. A to včetně základních podmínek, které je třeba dodržet, aby nenastaly problémy s aplikací.

V poslední části zprávy hodnotím výsledek projektu v porovnání se zadáním. Zmiňuji také jaké zkušenosti mi práce přinesla. A co jsem díky ní zjistil.

2 Dynamicky generované stránky

Tato kapitola obsahuje základní popis technologií umožňujících dynamické generování obsahu webových stránek, či aplikací. A zmíním v ní v praxi nejpoužívanější metody.

2.1 Proč generovat dynamicky

V počátcích World-Wide Webu (dále jen WWW) se nepočítalo s příliš častou změnou dat prezentovaných pomocí HTML na internetových stránkách. Z tohoto důvodu se veškerá data ukládala přímo do souborů, které v případě potřeby mohl autor změnit, čímž změnil obsah i pro uživatele. Tato jednoduchá metoda, nicméně časem přestávala stačit.

Díky rozšíření internetu se začalo uvažovat o dalších převážně interaktivních možnostech využití. Jelikož je koncept WWW velice obecný, není problémem, aby výsledné stránky, vytvářel nějaký jednoduchý program. Tento program by poté odesílal stránky k zobrazení v prohlížeči uživatelů. Také by tento program dokázal s uživateli komunikovat, například získávat a zpracovávat data odesílaná na server pomocí formulářů apod. Takovéto stránky tudíž označujeme za dynamicky generované stránky.

Tyto programy umožnili provázání internetových stránek do různých webových aplikací, ale také spojení různých informačních a vnitřních firemních systémů s internetovými prezentacemi. Taktéž bylo možné provázání webových aplikací s databázovými servery, a zobrazování dat nejen ze souborů na webu.

2.2 Metody používané při generování stránek

Postupem doby se objevilo několik technologií umožňujících v různé míře generovat webové stránky. Některé po čase vymizely, či si našly pouze specifické využití. V následujících bodech bych zmínil ty nejčastěji používané, či dle mého osobního názoru stojící za povšimnutí.

2.2.1 CGI skripty

CGI neboli „Common Gateway Interface“ je rozhraní sloužící k definici způsobu komunikace klienta se serverem. Pokud dodržíme konvence psaní CGI skriptů, můžeme tyto skripty psát v libovolném jazyce (C/C++, shell, Python, Pascal). Mezi hlavní nevýhody patří, že pro obsluhu každého požadavku je třeba spustit nový proces, což příliš zatěžuje server. Tato vlastnost vyřazuje z použití na takových serverech, které by nápor požadavků nezvládly.

2.2.2 Active Server Pages (ASP)

ASP funguje jako rozšíření jazyka HTML o další příkazy, které uvádíme přímo do HTML kódu. ASP je jakýsi framework umožňující použít libovolný jazyk podporující Active Scripting. Jako dva základní se používají JScript a VBScript, což jsou s ohledem na funkce poměrně chudé jazyky.

Od počátku je technologie ASP vyvíjena firmou Microsoft. Od začátku proto naráželo na veliký problém, jedná se totiž o komerční produkt a při cenové politice firmy Microsoft, je pro spousty uživatelů nemožné jej legálně používat. Tudíž se tolik nerozšířilo mezi studenty a drobnými tvůrci, nicméně pro velké firmy může být zastřešení technologie firmou Microsoft důležité. Další a zřejmě největší problém je, že ASP běží jen na serverech pod operačním systémem MS Windows.

2.2.3 Hypertextový preprocesor PHP

O jazyce PHP bych se krátce zmínil od jeho počátku. V roce 1994 Rasmus Lerdorf vytvořil pro svoji potřebu v Perlu (později v jazyce C) jednoduchý systém pro evidování přístupů na svých stránkách. Systém se stal oblíbený a používalo ho stále více uživatelů, kteří autorovi přinášeli nápady na vylepšení. Autor tento systém během let rozšířil a zdokonalil, později jej doplnil o podporu databází.

V té době se poprvé začíná tento jazyk označovat jako PHP. Jedná se o skriptovací jazyk, tedy části kódu, které se vkládají přímo do HTML kódu. Tento jazyk je šířen jako freeware (tzn. bezplatně) a díky tomu je také hojně využíván. Další jeho výhodou je, že se na jeho vývoji podílí mnoho různých organizací po celém světě a tudíž je stále zdokonalován .

Jazyk PHP je nejčastěji nasazován na serveru Apache, což ovšem není podmínkou. Dokáže tak stejně fungovat i na jiných WWW serverech např. od firmy Microsoft. Ovšem vzhledem k tomu, že Apache je celosvětově nejrozšířenější, bývá nejčastěji používán právě on. Server Apache je možné provozovat jak na strojích se systémem na bázi Unix, tak i na systémech MS Windows.

2.2.4 Technologie s použitím Javy

Taktéž jazyk Java se pokoušela proniknout do generování obsahu webových prezentací, za povšimnutí dle mého názoru stojí dvě metody :

2.2.4.1 Java Servlety

Jedná se o speciální třídy napsané přímo v jazyce Java, běžící na straně serveru. Server spustí Java Virtual Machine (JVM) a v ní poté běží servlet. Od prvního načtení zůstává servlet v paměti a následně obsluhuje všechny požadavky.

2.2.4.2 Java Server Pages (JSP)

Podobně jako předchozí metody ASP a PHP se používá příkazů, které se zapisují přímo do HTML kódu. K dispozici jsou speciální objekty umožňující práci s formuláři obdobně jako tomu je u technologie ASP.

Spuštěním JSP na straně serveru se zabývá servlet, který dané JSP automaticky převede do Javy a následně tento kód zkompiluje do byte-code a spustí.

2.3 Technologie použité v projektu

Po zvážení základních vlastností jazyků, především rozšíření a možnost implementace v daném jazyce na různých nejen školních serverech, jsem se rozhodl pro freeware jazyk PHP. Tato volba nemusí být tou nejlepší, což se může ukázat v průběhu návrhu a tvorby. Já jsem si tento jazyk taktéž vybral z důvodu, že jsem v něm tvořil svoji bakalářskou práci. A jelikož je třeba ukládat také data do databáze, zvolil jsem k PHP nejčastěji využívanou databázi MySQL, hlavně kvůli přenositelnosti.

3 Grafická uživatelská rozhraní

Jelikož v zadání projektu je tvorba grafického uživatelského prostředí, rozeberu nyní proč a jak grafická data uživateli zobrazovat. Dále uvedu jakou metodu a z jakých důvodů jsem si pro návrh a tvorbu projektu vybral.

3.1 Důvody grafického rozhraní

Základem WWW byly textová data, popřípadě tabulky. Tyto data se dala bez větších problémů zobrazit v jakémkoliv třeba i textovém prohlížeči. Nicméně během doby bylo třeba uživateli zobrazit i jiná data. Mezi tyto data patřily například grafy, fotografie, či jiné grafické elementy. Proto již v HTML jsou obsaženy možnosti jak zobrazit grafická data. S postupným vývojem výpočetní techniky se ovšem zdokonalovaly i možnosti zobrazování dat. Už nestačil jen textový formát, ale začínal se ztrácet i formát prosté grafiky. Začaly se proto používat i animované obrázky, které grafiku oživily. Ovšem v době, kdy operační systémy umožňují jednoduchá grafická interaktivní rozhraní, objevuje se potřeba nejen grafické prvky zobrazit, ale umožnit uživateli i práci s nimi. Proto se objevují nové metody, které umožňují uživatelům v různém rozsahu s grafickými prvky pracovat.

3.2 Základní principy

Grafická uživatelská rozhraní mohou mít spousty podob a řešení. V následujících bodech zmíním dle mého názoru ty zajímavé, což znamená, že výčet metod není kompletní.

3.2.1 DHTML a JavaScript

Rozšířením klasického HTML o kaskádové styly CSS umožnilo měnit grafické vlastnosti většiny prvků na stránce, čímž jsme získali DHTML. Spojením DHTML a JavaScriptu umožnilo, uživateli měnit na své straně různé vlastnosti prvků, čímž dosahujeme jednoduchých grafických rozhraní. Tyto rozhraní nám na příklad umožní měnit velikosti a pozice prvků použitím myši, či měnit vlastnosti elementů v závislosti na pohybu myši. Největším problémem je, že operace a vlastnosti prvků si může každý prohlížeč interpretovat dle svého, tudíž co funguje v jednom nemusí fungovat v jiném.

3.2.2 Statické obrázky

Nejednodušším grafickým uživatelským rozhraním je zobrazení obrázků. Obrázky mají své opodstatnění, pokud grafické prvky nemají uživatelé možnost měnit. Využití například pro design webové prezentace. Pokud dokážeme generovat dynamické stránky, můžeme taktéž generovat jako výstup skriptu, či programu obrázků. Toto se hodí například při zobrazování grafů a podobných

informací založených na proměnlivých datech (například v databázi). Ovšem generování obrázku zatěžuje server, proto je nutné zvážit, které obrázky nám stojí za to generovat a které už můžeme mít předpřipravené na serveru.

3.2.3 Java Aplety

Java Aplety jsou zkompileované Java programy, které je možné přidat do HTML stránky. O spouštění a práci s Java Apletem se stará interpret Java Virtual Machine (JVM), který může být přímo součástí prohlížeče, či jeho doplňkem. Ovšem také Java Aplety prohlížeč nemusí podporovat, či je může přímo zakázat. V Java Apletu pracujeme s grafikou stejně jako v samotné Javě, tudíž pokud umíme pracovat s Javou není velký problém tvořit Java Aplety.

3.2.4 ActiveX

Technologie ActiveX vychází z technologie OLE 2.0, vytvořené firmou Microsoft za účelem jednoduchého používání multimediálních dat v prohlížeči. ActiveX technologie funguje převážně na operačních systémech Windows, a prohlížeči Internet Explorer. I když se jedná o nejrozšířenější kombinaci na počítačích ve světě, není tato technologie většinou dále přenositelná.

Jednou zajímavou vlastností je, že technologie umožňuje spolupráci prohlížeče s jinými součástmi systému. Tato vlastnost umožňuje kupříkladu pracovat s různými typy dat přímo v prohlížeči, nicméně v případě bezpečnostních nedostatků, také vhodný skript umožní provádět útokům z internetu i činnosti, které bychom mu zřejmě nedovolili.

3.2.5 Flash

Flash je grafický vektorový program, který vyvinula společnost Macromedia, nyní je ve vlastnictví firmy Adobe. Flash umožňuje tvorbu vektorových obrázků a animací. Díky implementovanému vlastnímu programovacímu jazyku ActionScript, je možné v něm tvořit interaktivní aplikace na vyšší úrovni. ActionScript je neustále upravován a stává se z něj postupně rozsáhlý objektově orientovaný programovací jazyk, nicméně využitelný jen v programu Flash.

Díky malé velikosti výsledních souborů se technologie uchytila i na webu, a vytlačila animované GIF soubory, do té doby používanou možnost animovaných obrázků (například pro reklamní bannery).

Flash animace, exportované standartně do souboru s příponou .swf potřebují ke svému zobrazení speciální přehrávač označovaný jako Flash Player. Tento přehrávač je součástí, či doplňkem většiny internetových prohlížečů. Pro práci s animacemi na disku, umožňuje Flash exportovat také animace včetně přehrávače do souboru s příponou .exe, čím ovšem vzroste velikost cílového souboru přibližně o 500kB, což je v některých případech nezanedbatelná velikost.

3.3 Zvolená technologie

Po zvážení několika hledisek, mezi které patřily hlavně přenositelnost stránek, jednoduchost tvorby a navázání na jádro systému fungující pomocí dynamicky generovaných souborů PHP, jsem se rozhodl pro technologii Flash.

Flash je zobrazitelný ve většině prohlížečů, a je také často na ostatních internetových stránkách či aplikacích využíván. Takže jej případní uživatelé, jejichž prohlížeč ho implicitně nepodporoval, mají nainstalován formou doplňku.

Flash je nezávislý na platformě a tudíž poběží i v prohlížečích na většině operačních systémů, a nikoliv jako některé technologie jen na systému Windows s prohlížečem Internet Explorer.

Díky vestavěnému jazyku ActionScript můžeme grafické rozhraní přímo navázat na skripty systému a můžeme provádět různé změny v systému provedené v tomto grafickém prostředí.

4 Šablonování

Následující kapitola pojednává o principu šablonování, návrhu a problémech vhodného návrhu šablon pro dokumenty, ukládaných v systému.

4.1 Důvody k šablonování

Jelikož se v našem systému budou nacházet různé typy článků, je nutné, aby administrátor systému měl možnost si jednoduchým způsobem navrhnout, či změnit jejich vzhled. K tomuto účelu systém používá pro zobrazování článků šablony. Šablona je soubor pravidel umožňující specifikovat, jak která část dokumentu bude ve výsledku uživateli zobrazena.

4.2 Metody šablon

V následujících kapitolách shrnu základní metody, které by se daly použít při tvorbě a používání šablon.

4.2.1 Statické šablony

Tento princip spočívá v tom, že tvůrce systému předpřipraví administrátorům šablony již definitivně hotové. Šablony jsou detailně připraveny na libovolný obsah ve člancích, nicméně jejich detaily se administrátorovi nezobrazují. Proto může k jejich popisu být využito libovolných metod a nemusí být v žádném případě snadno pochopitelné, jelikož s jejich zdrojovými kódy pracuje pouze tvůrce systému a administrátoři si jen vybírají z nabízených možností.

4.2.2 HTML šablony s využitím CSS

Výsledné internetové stránky se v prohlížeči zobrazují pomocí HTML a CSS a veškerý obsah včetně článků, se do této formy převádí. Tudíž není nic jednoduššího, než nechat administrátorovi možnost napsat si přímo HTML a CSS kód, jak bude jeho rozvržení článků na internetu vypadat. Tímto odpadá nutnost tvůrce aplikace, aby se tvořil s hotovými šablonami a tuto činnost si zařizuje administrátor sám. U této varianty nicméně je nutné, aby administrátor znal HTML a CSS programování a tvořil korektní kód, protože výsledek ovlivňuje také to jak bude fungovat celý systém.

4.2.3 Šablony s pseudo-značkami

Zjednodušením práce pro tvůrce a správce šablon je použití tzv. pseudo-značek. Jedná se o značky vycházející z principu HTML značek, nicméně jsou to značky jejichž význam v HTML není definován. Tyto pseudo-značky může administrátor použít v šabloně i když nezná přímo HTML kód. Jejich vlastnosti jsou v šabloně přednastavené, nicméně administrátor má možnost si je upravit dle svého vkusu. Tyto pseudo-značky se při zobrazování převedou na klasické HTML značky. Tato varianta je obtížná tím, že tvůrce systému musí pro tyto pseudo-značky určit jejich převod na HTML značky. Čímž dochází pouze k opsání významu HTML značek do šablon.

4.3 Zvolená varianta

Po zvážení metod ohledně používání šablon, jsem se rozhodl použít možnost editovat šablony administrátorem systému. Z diskutovaných variant jsem se rozhodl použít metodu přímí editace HTML a CSS kódu. Jelikož pokud chce administrátor editovat šablonu k obrazu svému, bude stejně základní pravidla (jako např. pozicování prvků) muset ovládat. Přesný princip tvorby a editace šablon uvedu později v sekci návrhu webové aplikace, až zmíním všechny potřebné součásti.

4.4 Šablony a jejich součásti

Jelikož se jedná o šablony, které jsou univerzální pro několik článků, které v době tvorby a nastavení šablony ani nemusí existovat, nemůže šablona pracovat s konkrétním obsahem článku. Nicméně musíme umožnit autorovi šablony nastavit vlastnosti pro článek. Z tohoto důvodu se domnívám, bude nejjednodušší použít proměnné, které budou před zobrazení nahrazeny skutečným obsahem daného článku. Vlastní proměnné vychází převážně ze struktury ukládání informací o článku. Měly by obsahovat hlavně :

- **Titulek článku** – tato položka bude obsahovat nadpis článku, zobrazována bývá většinou v záhlaví článku větším písmem.
- **Obsah článku** – kompletní text článku, včetně všech jeho součástí jako jsou tabulky, obrázky, grafy, či seznamy. Tato proměnná obsahuje nejdelší, ale také hlavní informace.
- **Kategorii článku** – jelikož máme možnost členit články do kategorií, můžeme umožnit zobrazení kategorie do které daný článek patří.
- **Provázané články** – tato položka obsahuje seznam provázaných článků k článku zobrazenému, sloužící k jednodušší navigaci mezi články. Tato proměnná by měla obsahovat pouze jednu z položek článků, aby bylo možno rozhodnout, zda chceme položky zobrazit jako seznam s odrážkami, či jen jako text oddělený oddělovačem.

Jako další, ne již tak důležité položky, bychom měli mít proměnné pro informace, které administrátor nemusí chtít zobrazit, nicméně může tyto informace požadovat. Patří mezi ně :

- **Datum napsání** – Datum, kdy autor článek do systému vložil, případně kdy článek napsal, pakliže se liší od data vložení do systému.
- **Autora článku** – jedná se o autor článku, což ocení kupříkladu systémy, kde přispívá několik různých autorů. Obsahuje jméno a příjmení, případně přezdívku.
- **Zkrácený text** – v některých případech může administrátor mít zájem o zobrazení krátkého text článku, kupříkladu před článkem, či jako jeho shrnutí za ním. Tato proměnná takovýto zkrácený text bude obsahovat.

Další proměnné by mohli vzniknout na přání administrátora až v průběhu práce se systémem. Nicméně bude se jednat o informace, které jsou již v systému uloženy, aby nebylo nutno přepracovávat návrhy a strukturu databáze. Taktéž by bylo možné rozšířit systém o položky, které nesouvisí přímo s obsahem článku, ale spíše s rozložením celého systému. Jedná se o položky jako zaslat článek e-mailem, či tisk článku. Tyto proměnné jsem ve svém návrhu neuvedl, zobrazení a práci s těmito položkami jsem nechal globální pro všechny články společně, což je součástí aplikace a nikoliv každého jednotlivého článku.

5 Návrh webové aplikace

V zadání je uvedeno, že mám navrhnout webovou aplikaci umožňující vkládat a editovat dokumenty, taktéž ovšem graficky definovat navigační strukturu webu a vše uložit do databáze. Dovolil bych si tento problém rozdělit do následujících bodů a u nich uvést návrh a případné problémy.

5.1 Struktura webové aplikace

Jelikož aplikace umožňuje editaci dokumentů, bude nejjednodušší rozdělit ji na dvě části z nichž jedna bude administrátorská, a druhá uživatelská.

5.1.1 Administrátorská část

Vstup do této sekce bude umožněn pouze administrátorovi systému a to po ověření přístupovým heslem. Jelikož se jedná o systém, který bude obsluhovat pouze jeden administrátor nemusíme tudíž v systému obsluhovat správu uživatelů ani provedené změny. Co vše může administrátor dělat je uvedeno v případě použití (viz příloha A). Základní popis těchto činností s případnými problémy následuje dále.

5.1.1.1 Správa článků

Jak je z diagramu použití patrné, může administrátor pracovat s články v systému obsaženými. Může vytvářet nové, editovat a mazat existující. Administrátor pro každý článek eviduje Titulek, Text článku, Datum vytvoření, Autora a Zkrácený popis. Články jsou taktéž provázány s kategoriemi, tím způsobem, že každý článek náleží jedné do kategorie, kterou při tvorbě nového, či editaci stávajícího článku zvolíme z nabídky kategorií v systému obsažených. Do jedné kategorie ovšem může patřit více článků. Taktéž jsou články provázány s šablonami, s tím že jednu šablonu může používat více článků.

Dále také při přidání a editaci může administrátor určit provázání s jinými články. Výsledkem této části je vazba článků mezi sebou.

Při mazání musí systém upozornit, pokud existuje s aktuálně mazaným článkem nějaká vazba. Systém v tomto případě nabídne administrátorovi, zda chce vazbu smazat, či článek zachovat.

Jelikož všechny operace s články budou prováděny v grafickém rozhraní, bude jeho návrh a specifičtější popis zobrazení a rozčlenění uveden až v samostatné kapitole.

5.1.1.2 Správa kategorií

Jelikož systém obsahuje kategorie, může také administrátor spravovat kategorie. Kategorie jako jeden ze svých atributů obsahují rodiče. V systému je tato vlastnost proto, aby bylo v případě zájmu možné

kategorie zanořovat do sebe navzájem. Každá kategorie bude mít svůj identifikátor, který bude v případě potomka uveden v atributu rodič.

System bude již od počátku obsahovat tzv. kořenovou položku, tj. položku, jejímiž potomky budou kategorie nejvyššího řádu. Při přidávání nové kategorie, či při přidávání existující, systém nabídne uživateli všechny existující kategorie jako rodiče, a to včetně tzv. kořenové položky.

Při mazání je nutné pohlídat, zda je daná kategorie rodičem jiné (případně jiných) kategorií a tudíž nepovolit smazání dané kategorie. Taktéž je nutné ošetřit mazání kategorií ke kterým jsou provázány nějaké články. V tomto případě jsou na výběr dvě varianty, a to nepovolit smazání případně nabídnout náhradní kategorii pro všechny články, či každý změnu provést pro článek samostatně. Pro náš případ se mi jeví jako nejlepší možnost zakázat smazání, jelikož systém umožňuje editaci v jiné své části. Jelikož systém umožňuje definovat přednastavenou šablonu pro kategorii, je taktéž v této části umožněno pro každou kategorii definovat tuto šablonu.

Detailnější popis rozhraní pro správu kategorií nechám do kapitoly věnované grafickému uživatelskému rozhraní.

5.1.1.3 Správa šablon

Administrátor může taktéž pracovat se šablonami. System pracuje s dvěma typy šablon. První jsou šablony systémové, které jsou již v systému od počátku vytvořeny a administrátor je může použít, ale nemůže je editovat ani mazat. Dále jsou v systému šablony uživatelské, které si vytváří administrátor sám, a to buďto z existující šablony její modifikací a následným uložením pod novým názvem, či vytvořením z prázdného modelu. Šablony se váží k jednotlivým článkům, tudíž musíme při mazání ošetřit zda nemažeme šablonu, dle které se některý ze článků vykresluje. Editace šablon neprobíhá přes téměř jako jediná přes grafické rozhraní. Jedná se o to, že šablony se zapisují textově a jejich výsledek je náhled stránky zobrazitelný v prohlížeči, tudíž by grafické rozhraní, pouze provádělo stejnou práci jakou zvládne textové rozhraní. Popis rozvržení bude následovat v další části.

5.1.2 Uživatelská část

Tato část bude veřejně přístupná všem uživatelům, bez potřeby se jakkoliv přihlašovat, či registrovat. Ovšem také uživatel může pouze zobrazit kategorie, procházet články a ty si případně tisknout, či zaslat e-mailem. Tato je ovšem hlavní a nejčastěji zobrazovanou částí systému, proto by měla mít příjemný vzhled, také by měla být jednoduše ovladatelná.

Uživateli se zobrazí menu s kategoriemi, do kterých články jsou rozděleny, a krátký popis článku v aktuální kategorii. Při zobrazení určitého článku se místo krátkého popisu více článků, zobrazí kompletní článek včetně nadpisu a ostatních uložených informací. U každého článku budou uvedeny také články, které jsou s ním ve vazbě. Vše bude rozmístěno v závislosti na administrátorem zvolené šabloně pro daný článek.

Taktéž ke každému článku budou zobrazeny položky, které jsem vynechal v šablonách, jedná se o možnost tisku daného článku, taktéž možnost nechat si článek zaslat e-mailem.

5.2 Databázový návrh

Tato kapitola pojednává o návrhu relační databáze, jejíž výsledný model je jsem uvedlo u této zprávy jako přílohu B. Jak jsme se již dříve ve zprávě zmínil, použiji databázový server MySQL, díky snadné provázanosti s PHP a také díky tomu, že je pro nekomerční účely zdarma.

5.2.1 Co a proč ukládat do databáze

Na začátku práce jsem si musel rozvrhnout, jaká data budeme jakým způsobem v systému ukládat. Provázání s databází nám umožňuje většinu dat odstínit od zdrojových kódů aplikace, nicméně není dobře do databáze cpát i informace, které bychom mohli mít uloženy v souborech. V následujících bodech uvedu základní členění ukládaných dat.

5.2.1.1 Články

Jelikož se jedná o systém obsahující články budou hlavní částí ukládanou v tabulce články. Zde jsem se rozhodl uložit tři základní informace a to ID (jednoznačný identifikátor článku), Titulek článku, Obsah článku a jeho Kategorii (jedná se o identifikátor z tabulky kategorií). Kvůli šablonování jednotlivých článků také pro každý článek evidujeme identifikátor do tabulky šablon.

Během tvorby návrhu jsem nicméně dospěl k dalším věcem, které je minimálně vhodné u každého článku uvést. Jedná se o informace kdy a kým byl článek vytvořen. Pro jednodušší orientaci mezi články jsem také uvedl položku zkrácený text článku, který se bude zobrazovat jen při výpisu seznamu článků a který nebude nutné zadat.

5.2.1.2 Kategorie

Jelikož systém umožňuje rozčlenění článků do kategorií, musíme ukládat i tyto kategorie. Z důvodu, že administrátor má možnost si kategorie modifikovat a vzájemně zanořovat je nejjednodušší uložit i kategorie to vlastní tabulky v databázi. U kategorií evidujeme ID (jednoznačný identifikátor), Název kategorie, Popis kategorie a jejího rodiče. Při vytvoření systému je třeba definovat hlavní (kořenovou) kategorii, do které budou patřit všechny ostatní, protože atribut rodič je identifikátorem nadřazené kategorie a tudíž bychom nebyli schopni zanořování kategorií zobrazit. Taktéž evidujeme pro každou kategorii přiřazenou šablonu díky identifikátoru z tabulky šablon.

5.2.1.3 Šablony

Systém umožňuje ukládání, editaci a přiřazování šablon k článkům a kategoriím. Tudíž jsem stál před rozhodnutím, zda šablony ukládat jako soubory (obdobně jako se ukládají kaskádové styly), či je jako

data uložit do tabulky databáze. Rozhodl jsem se pro uložení do databáze a to z důvodu, že administrátor může šablony modifikovat, a taktéž vytvářet nové, což by mohlo v některých případech kolidovat s nastavením práv na serveru. Taktéž jsem si vybral uložení do databáze z důvodu, že při přiřazování k článkům a kategoriím, jednoduše použiji identifikátory použité jako primární klíč.

U šablon eviduji v systému její ID (jednoznačný identifikátor), Název šablony, Obsah šablony a Typ šablony. Obsahem šablony rozumíme přesné znění pravidel pro zobrazení, obsahující proměnné a ostatní informace sloužící k uplatnění šablony. V atributu typ ukládáme zda je šablona uživatelská, či systémová, tudíž zda ji uživatel může či nemůže změnit.

5.2.1.4 Vazby mezi články

Jelikož máme možnost články mezi sebou provazovat, musíme taktéž uložit tyto vazby. Rozložením relace M:N jsem si zavedl další tabulku, do které ukládám tyto vazby. Tato tabulka obsahuje pouze identifikátory obou článku, mezi kterými je vytvořena vazba s tím, že který je uveden jako první není podstatné. Při práci s těmito daty vyhledávám v obou sloupcích.

5.2.2 Formát ukládaných dat

V příloze B je uveden návrh databáze ale bez typů jednotlivých sloupců. V této části zprávy bych zmínil jaké datové typy a proč jsem v návrhu použil.

5.2.2.1 Identifikátory (článků, kategorií, šablon)

Pro identifikátory ve všech tabulkách v databázi používám stejný datový typ Integer, který má rozsah od -2147483648 do 2147483647, což se mi zdá dostatečné i pro případ častých změn a archivaci starších článků.

5.2.2.2 Názvy Kategorií a šablon a autor článků

Pro tyto atributy jsem použil typ Varchar s délkou 30 znaků, což je dle mých odhadů postačující k uložení požadovaných hodnot a zároveň není velikost příliš velká z důvodu komfortního zobrazování na webu.

5.2.2.3 Titulek článků a popis kategorie

Jelikož titulek může obsahovat informace důležité o článku, nastavil jsem vyšší rozsah proměnné Varchar a to na 50 znaků. Tutéž velikost jsem použil i pro popis kategorie, jehož primárním účelem je rozšířit informace o kategorii.

5.2.2.4 Zkrácený text u článků

Jelikož se jedná o krátký popis článku, či jen začátek jeho obsahu, nastavil jsem tuto proměnnou na typ Varchar o délce 250 znaků, což by mělo na popis stačit. Zároveň jsem nepoužil větší délku a to z důvodu, aby popis u kratších článků nepřekonal délkou obsah.

5.2.2.5 Text článků a obsah šablon

Jelikož oba dva atributy budou uchovávat velké objemy dat, nastavil jsem pro tento sloupec formát Text, který umožní uložit přibližně 64 kB informací.

5.2.2.6 Typ šablony

V této fázi máme v systému pouze dva druhy šablon a to předpřipravené a uživatelské. Tudiž by nám stačila informace na jednom bitu, je ovšem možné, že v dalším rozšíření aplikace bude třeba uložit více typů, proto jsem použil datový typ TinyInt se svým rozsahem od – 127 do 128.

5.2.2.7 Datum vložení článku

Pro tento atribut jsem rozmýšlel, zda ukládat také čas vložení, jelikož nevíme, zda uživatel bude vkládat několikrát denně různé články. Raději jsem sloupec nechal ve včetně času a to v datovém typu DateTime. Tento datový typ ukládá ve formátu YYYY-MM-DD HH:MM:SS.

5.3 Grafické uživatelské rozhraní

Jak jsem se již zmínil, z hlediska používání je grafické uživatelské prostředí tou nejzajímavější částí. Nicméně z pohledu návrhu jsou ostatní části stejně důležité ne-li důležitější. Také k němu má přístup pouze administrátor aplikace, nikoliv běžný uživatel. Zmínil bych zde popis a funkce jednotlivých částí. Jejichž rozdělení je již dle zobrazení v systému.

5.3.1 Administrace článků a vazeb mezi nimi

Hlavní částí je možnost spravovat články a vazby mezi nimi. Jedná se o grafické prostředí, ve kterém administrátor pracuje jako v jednoduchém grafickém editoru. Má na výběr z několika činností, které uvedu dále v pořadí důležitosti nikoliv v pořadí v jakém budou zobrazeny v systému.

Hlavní okno grafického rozhraní obsahuje zobrazeny všechny články a to formou obdélníkových boxů, které obsahují titulek článku. Abychom věděli do jaké kategorie článek patří, jsou také články odlišeny barevným rámečkem lišícím se dle kategorií. Mezi články zobrazenými na ploše jsou linkami zobrazeny vazby.

Jednotlivé změny se neukládají přímo, uložení se provede až po stisknutí tlačítka uložit. Tlačítko uložit je zobrazeno ve spodní části grafického rozhraní, se sousedícím tlačítkem reset, které slouží k navrácení do původních na serveru uložených hodnot.

5.3.1.1 Úprava informací o článku

Administrátorovi po vybrání této akce je nabídnuto vybrání článku, který bude modifikovat. Jednoduchým kliknutím si administrátor vybere článek a bude zobrazeno okno s vlastnostmi, které daný článek má, včetně jeho textu. Vlastnosti jako Kategorie a Šablona mu budou zobrazeny formou

již připravených a z tabulek naplněných roletkových menu, ve kterých uživatel vybere tu správnou. Uživatel potvrdí změnu dat, nicméně ta se provede na serveru, až při stisknutí tlačítka uložit.

5.3.1.2 Přidání článku

Přidání článku bude probíhat obdobně jako úprava, jen s tím, že uživateli bude zobrazeno okno s prázdnými položkami. A to včetně roletek, ve kterých budou načteny údaje z dalších tabulek. Administrátor potvrdí vložení článku a může s ním případně provádět úpravy, ovšem uložení na server se také provede až po stisku tlačítka uložit.

5.3.1.3 Smazání článku a vazby

Administrátor má také právo mazat články, či vazby mezi nimi. Po zvolení akce mazání, uživatel může vybrat článek, či vazbu a bude dotázán, zda chce danou položku skutečně odstranit. Odstranění se provede pouze za předpokladu, že nedochází ke kolizím, zmiňovaných v dřívějších bodech zprávy. Pokud by došlo k nějakému problému (mazání článku, který je s jiným ve vazbě) bude administrátor upozorněn výrazným oknem s popisem problému a bude na něm, zda a jak se rozhodne pokračovat.

5.3.1.4 Vytváření vazeb mezi články

Pokud se administrátor rozhodne tvořit vazby, učiní tak spojením dvou různých článků, pomocí tažení myši. Pokud taková vazba doposud neexistuje a jsou vybrány dva různé články, je vazba vykreslena a evidována, pokud dojde k chybě je o ní administrátor informován.

5.3.1.5 Posuny článků

Jelikož pozice článků na ploše určuje z počtu prvků automaticky systém, je občas nutné, aby si administrátor měl možnost prvky přesunout. Například, aby viděl jasněji vazby vytvořené mezi články. Také tuto možnost má administrátor, přesun článků funguje metodou drag&drop s tím, že nesmí uživatel umístit článek mimo meze plochy, či na jiný existující článek. Při přesunu se automaticky překreslují i všechny vazby s daným článkem.

5.3.2 Administrace kategorií

Jelikož se kategorie mohou nekonečně zanořovat, je pro jejich zobrazení vhodné použít nějaké formy stromu. Vrcholem všech kategorií je kořenová položka, která se v systému nachází automaticky a již nelze ani změnit ani odstranit. Ostatní kategorie může administrátor spravovat. A to pomocí následujících funkcí, které má zobrazeny v menu. Obdobně jako při editaci článků, i zde se ukládání změn provádí, až po stisku tlačítka uložit, taktéž se zde nachází tlačítko reset, pro uvedení do původního stavu.

5.3.2.1 Úprava kategorií

Při zvolení editace, administrátor pomocí myši vybere kategorii, kterou chce upravit. Po té co vybere kategorii ze stromu s kategoriemi, tak se mu zobrazí okno s vlastnostmi kategorie s tím, že šablonu a rodiče si vybere z roletkových menu, naplněných z databáze. Při výběru rodiče je brán zřetel, aby si nevybral kategorii, která je jeho potomkem, jelikož by došlo k zacyklení.

5.3.2.2 Přidání kategorie

Přidání probíhá obdobně jako úprava s tou změnou, že pole v okně, které se zobrazí administrátorovi bude mít nevyplněné kolonky, a v roletkách s rodičem a šablonou budou pouze naplněny informace z jiných tabulek v databázi.

5.3.2.3 Mazání kategorie

Pokud uživatel bude chtít smazat kategorii, vybere ji kliknutím myši ve stromě. Před smazáním se provedou nezbytné kontroly a pokud dojde k chybě, bude administrátor informován zobrazením chybové hlášky s možností provedení opravy.

5.4 Další uživatelská rozhraní

Jelikož systém se nebude skládat pouze z administračního grafického rozhraní, musí obsahovat i další části. V následujících bodech zmíním ty základní.

5.4.1 Administrace šablon

Jak jsem se již zmínil, tato sekce se z hlavní části skládá z práce HTML a CSS, tudíž je nadbytečné její nastavení provádět v grafickém rozhraní. Postačí jednoduché textové rozhraní, které bude mít dvě části, jednu ve které bude administrátor mít možnost editovat pravidla a formát šablony, a druhou ve které se mu na ukázkovém příkladu budou změny zobrazovat. Jak jsem se již zmínil šablona bude kombinací HTML, CSS a proměnných symbolizujících určité části článků. Z důvodu většího komfortu, by někde v okolí editační části měla být zobrazena nápověda obsahující všechny proměnné, které systém zná a dokáže je nahradit částmi článku. Se šablonami může administrátor provádět různé operace uvedené dále. Při výběru práce se šablonami jsou tyto zobrazeny v seznamu na jedné obrazovce.

5.4.1.1 Úprava šablony

Administrátor může změnit šablonu editací pravidel v dané části okna při zobrazení pravidel a náhledu. Provedené změny jsou mu poté zobrazeny v okně s ukázkou na vzorovém článku.

Až si bude administrátor jist, že změny které provedl jsou vyhovující, provede uložení. Při ukládání se hlídá zda se nepokouší uložit změny do šablony, která je systémová a tudíž se nedá

změnit. V takovém případě by byl administrátor upozorněn a byla mu nabídnuta možnost uložení do nové nesystémové šablony.

5.4.1.2 Přidání šablony

Administrátor může vytvořit i novou šablonu z prázdného návrhu. Zobrazí se taktéž editační okno, jako při úpravě existující šablony, nicméně prázdné. Je zde očekáváno, že administrátor již zná systém šablon a ovládá HTML a CSS. V jiném případě je jednodušší novou šablonu vytvořit úpravou existující a jejím uložení pod novou položkou. Při přidání šablony se kontroluje, zda již šablona se stejným názvem v systému neexistuje, aby nedocházelo ke špatným zadáním při návaznosti na články a kategorie.

5.4.1.3 Smazání šablony

Administrátor může šablonu taktéž smazat, je mu to umožněno v seznamu šablon a před smazáním je nejprve vyzván, zda si opravdu přeje danou šablonu odstranit a následně je provedena kontrola, zda není daná šablona použita u kategorií, či článků. Před provedením smazání je provedena kontrola, zda se nejedná o šablonu systémovou, pokud ano, je administrátorovi zakázáno ji smazat a je o tom informován.

5.4.2 Uživatelská část systému

Jelikož k systému kromě administrátora přistupují i uživatelé, je třeba vytvořit část systému i pro ně. Jedná se o část systémů, ve které lze pouze zobrazovat články a kategorie. Stránka bude rozdělena na přibližně následující části. Přičemž na úvodu bude zobrazena informační stránka o portálu, odkazující na některé ze článků, při výběru kategorií bude zobrazen seznam článků pro danou kategorii.

5.4.2.1 Záhloví

Tato část se bude nacházet v horní části každé stránky pro uživatele a bude neměnná. Bude v ní název webové aplikace a její logo. Případně zde může být ještě nějaký podtitul, či motto blíže popisující aplikaci.

5.4.2.2 Menu s kategoriemi

Tato sekce bude v levém sloupci a bude zobrazovat seznam kategorií a to včetně jejich zanoření. Pro její přehlednost by měla umožnit postupné rozkrývání potomků každé kategorie. S tím, že na počátku se zobrazí pouze kořenové kategorie (ty již mají rodičem kořenovou kategorii) a v případě, že se nacházíme v nějaké z kategorií, poté by se rozložila větev vedoucí k dané kategorii.

5.4.2.3 Tělo článku

Tato část bude zabírat největší prostor stránky a je to hlavní, co chceme uživateli zobrazit. Bude sousedit s menu s kategoriemi. A jeho šířka bude větší. Vlastní obsah a forma zobrazení je věcí šablony, která je pro každý článek definována.

Nad článkem bude pouze zobrazena postupná cesta kategoriemi, kterou se dostaneme ke konkrétnímu článku. Taktéž zápatí článku bude rozšířeno o položky poslat e-mailem a tisknout. Jedná se o položky stejné pro všechny články, tudíž nejsou zahrnuty v šablonách, ale až zde.

5.4.2.4 Zápatí stránky

Tato část bude pouze opticky ukončovat stránku a bude obsahovat nutné položky jako autora projektu s kontaktem něj a také bude obsahovat odkaz na mapu celého webu.

6 Závěr

První částí semestrálního projektu bylo seznámit se technologiemi pro dynamickou tvorbu webu a grafických uživatelských prostředí na webu. Při psaní této části jsem prostudoval několik různých publikací a velký počet internetových stránek a diskuzí. V některých jsem se dozvěděl zajímavé nápady, v jiných naopak další podněty ke studiu.

Celkově jsem během této části narazil na několik zajímavých internetových stránek, převážně nejvíce informací jsem čerpal z osobních stránek pana Jiřího Koska [1], který se již delší dobu věnuje tématu programování webových aplikací, převážně technologií PHP s podporou databází MySQL. Také ovšem vyučuje, vyučoval, či se jen účastnil spousty zajímavých kurzů a přednášek. Ze kterých na svých stránkách zveřejnil prezentace, či jen podklady v elektronické podobě.

Při seznamování se s možnostmi grafických uživatelských rozhraní na webu jsem prostudoval velké množství stránek, zkoumající jak na podobných projektech pracovali ostatní. Nakonec jsem se rozhodl po zvážení všech možností se rozhodnout sám.

Taktéž v další části, kterou byl návrh šablon pro webovou aplikaci, jsem přemýšlel o možných řešeních, z nichž jsem použil to z mého hlediska nejefektivnější i když mírně náročnější. Při tvorbě této části jsem se potýkal s problémy, co vše bude šablona obsahovat a co nikoliv. Zvolil jsem proto podle mě nejdůležitější informace a ostatní jsou součástí návrhu grafického rozvržení aplikace.

V poslední a zřejmě hlavní části jsem se snažil navrhnout aplikaci přesně dle požadavků, nicméně v průběhu tvorby jsem byl nucen některé požadavky, či svoje nápady změnit a upřesnit do realizovatelných mezí. Při návrhu systému a databáze jsem se snažil dát pozor na jednoduchost, ale funkčnost. Návrh grafického rozhraní jsem pojal textovou formou, jelikož specifický vzhled by byl pro implementaci příliš svazující a nemusel by být plně realizovatelný.

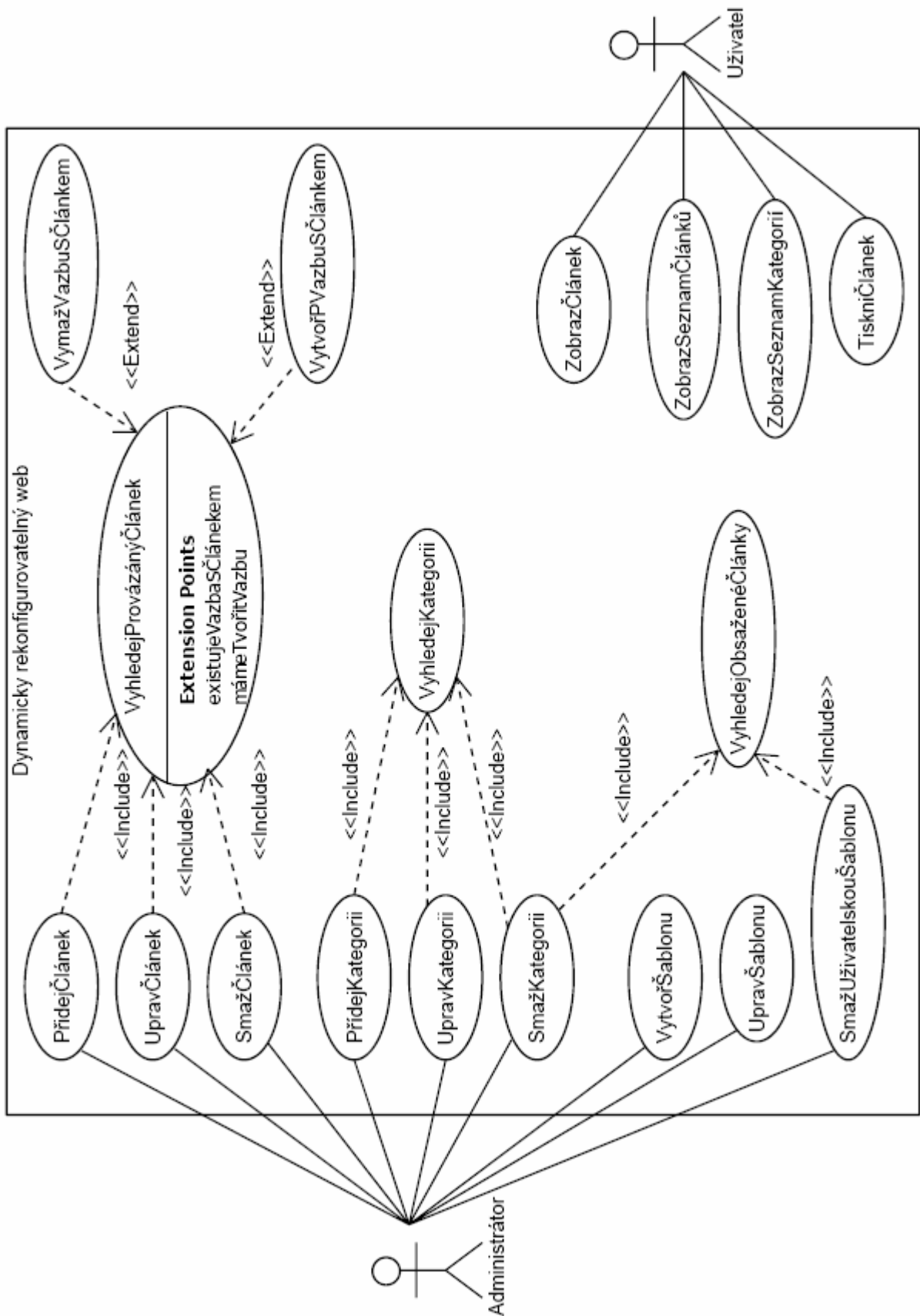
Během tvorby jsem se seznámil s metodami na tvorbu dynamických webových stránek a s metodami zobrazování grafických prostředí na internet. Zároveň jsem si vyzkoušel kompletní návrh aplikace. Což je do budoucna pokud mě programování webových aplikací zajímá, docela přínosné.

Tuto práci by bylo možné ještě rozšířit o další vlastnosti systému, které již nejsou podstatné, ale při tvorbě by mohli být specifikovány. Kupříkladu o evidování uživatelů a zasílání novinek.

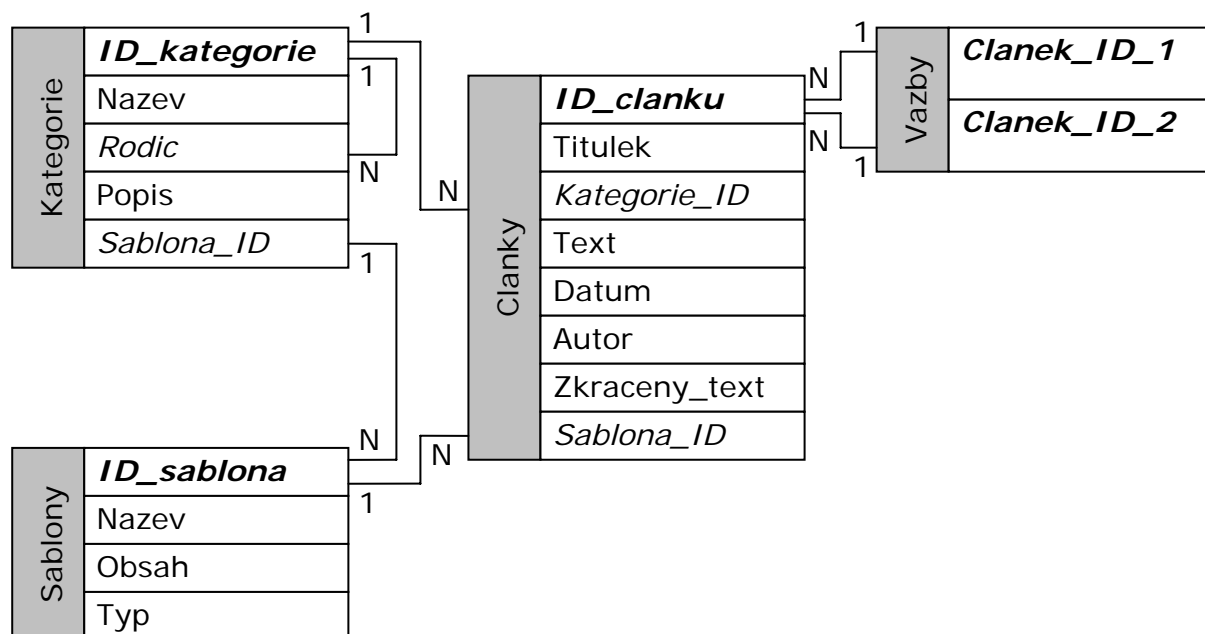
Literatura

- [1] *Domovská stránka J. Koska*, <http://www.kosek.cz>.
- [2] *PHP - Hypertext Preprocessor*, <http://www.php.net>.
- [3] *MySQL - The World's Most Popular Open Source Database*, <http://www.mysql.org>.
- [4] Kosek, J. *PHP - Tvorba interaktivních internetových aplikací*, Grada, Praha, 1998

Příloha A : Diagram případů použití



Příloha B : Databázový model



Legenda k modelu :

- Primární klíč je zobrazen *tučnou kurzívou*
- Cizí klíče jsou zobrazeny *kurzívou*
- Názvy tabulek jsou uvedeny vlevo ve svislém směru