

# WWW stránky hudebního interpreta

Web pages for band or vocalist

Petr Bernatík

---

Bakalářská práce  
2008

 Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav aplikované informatiky  
akademický rok: 2007/2008

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr BERNATÍK**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Webové stránky hudebního interpreta**

Zásady pro vypracování:

1. Cílem je zpracovat modelové stránky hudebního interpreta.
2. Analyzovat a navrhnout vhodné technické řešení.
3. Možný obsah prezentace: informace o interpretovi, diskografii, diskusní fórum, on-line obchod, fotogalerii.
4. Realizace prototypu prezentace.
5. Hodnocení existujících prezentací.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **Creating web sites the missing manual /**. – 1st ed.. – Beijing ; Sebastopol, Calif. : Pogue Press/OReilly, 2006. – ix, 548 s. : il. ; 24 cm.
2. **333 tipů a triků pro maximální využití Internetu /**. – Vyd. 1.. – Brno : Computer Press, 2007. – 215 s. : il. ; 23 cm.
3. **Programming ASP.NET AJAX**. – 1st ed.. – Beijing ; Cambridge : OReilly, c2007. – xviii, 454 s. : il.
4. **ASP.NET 2.0 cookbook**. – 2nd ed.. – Sebastopol, Calif. : OReilly, c2006. – 1014 s. : il.
5. **ASP.NET 2.0 a C-sharp : tvorba dynamických stránek profesionálně /**. – Vyd. 1.. – Brno : Zoner Press, 2006. – 1376.

Vedoucí bakalářské práce:

**Ing. Petr Šilhavý**

Ústav aplikované informatiky


Datum zadání bakalářské práce:

**20. února 2008**

Termín odevzdání bakalářské práce:

**5. května 2008**

Ve Zlíně dne 20. února 2008



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Bakalářská práce je modelem webové aplikace určeným k reprezentaci hudebního interpreta nebo hudební skupiny a nabízí většinu moderních prvků pro interaktivitu s uživatelem. Práce není vytvořena na základě přímých požadavků zadavatele, ale přesto je vypracována pro reálného hudebního interpreta – hudební skupinu, která přispěla svými návrhy pro zpracování a použitou funkcionalitu. Stránky jsou plně editovatelné bez znalostí základů HTML a neklade žádné technické znalosti na správce webu. Obsahem je grafické rozhraní, databáze, přihlášení uživatele a podpora uživatelských rolí. Výsledkem je přátelská webová aplikace, nabízející pokročilou komunikaci mezi uživatelem a webovým správcem.

Klíčová slova: ASP. NET, SQL Server, hudební interpret, www stránky, databáze, informace

## **ABSTRACT**

The bachelor document is model of web application dedicated to music band or vocalist. It should be a representative and fully modern builded web site. There is not a company submission but site have been builded for real band that have been supporting me with their own suggestions about appearance and behaviour. There is no previous experience with HTML needed for administrator. The main content is graphical interface, database and role logging. The result is a user friendly application that supports advanced communication between user and web site administrator.

Keywords: ASP.NET, SQL Server, musician, web site, database, informations

Poděkování:

Děkuji vedoucímu práce panu Ing. Petr Šilhavému za odborné vedení a podnětné rady při realizaci mé práce.

**Motto:**

*Překážky v nás vyburcují vlohy, které by v nás za příznivých okolností zůstaly dřímat.*

Q. Horatius Flaccus

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně

.....  
Podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 NÁVRH ŘEŠENÍ</b> .....	<b>11</b>
1.1 ANALÝZA ODVĚTVÍ TRHU A POŽADAVKŮ ZADAVATELE .....	11
1.2 NÁVRH ŘEŠENÍ UŽIVATELSKÉHO PROSTŘEDÍ .....	11
1.2.1 Grafické zpracování a design webu .....	11
1.2.2 Registrace uživatele a editace obsahu webu .....	13
1.2.3 Editace textového obsahu webu .....	14
1.2.4 Vkládání souborů .....	14
1.3 POUŽITÉ PROGRAMOVÉ VYBAVENÍ .....	15
1.3.1 Kaskádové styly (CSS) .....	15
1.3.2 Jazyk ASP.NET.....	16
1.3.3 XML .....	17
1.3.4 AJAX.....	18
1.3.5 SQL Server Express .....	18
1.3.6 C# .....	18
1.3.7 .NET Framework.....	19
<b>II PRAKTICKÁ ČÁST</b> .....	<b>20</b>
<b>2 VOLBA PÍSEM A ZNAKOVÉ SADY</b> .....	<b>21</b>
2.1 POUŽITÍ FONTŮ .....	21
2.2 VOLBA ZNAKOVÉ SADY .....	23
<b>3 VZOROVÁ STRÁNKA A MOTIVY</b> .....	<b>24</b>
3.1 MASTER PAGE .....	24
3.1.1 Záhloví a hlavička Master page .....	25
3.1.2 Tělo Master Page.....	25
3.2 VOLBA MOTIVŮ .....	26
3.2.1 Třída BasePage.....	27
3.2.2 Řídící prvek ThemeSelector.....	27
3.3 TVORBA MENU.....	28
3.4 TVORBA PRVNÍ STRÁNKY DEFAULT.ASPX .....	28
<b>4 DATABÁZE</b> .....	<b>30</b>
4.1 VYTVOŘENÍ DATABÁZE .....	30
4.2 STRUKTURA TABULEK .....	30
4.2.1 Tabulky a jejich položky .....	31
4.3 DATABÁZOVÉ DIAGRAMY.....	33
4.3.1 Diagram sekce Články .....	33
4.3.2 Diagram sekce Forum .....	34
4.3.3 Diagram sekce Ankety .....	34
4.3.4 Diagram sekce Obchod .....	35

4.4	ULOŽENÉ PROCEDURY .....	35
<b>5</b>	<b>LOGICKÉ ROZDĚLENÍ A ARCHITEKTURA .....</b>	<b>37</b>
5.1	UŽIVATELSKÉ PROFILY .....	38
5.1.1	Vestavěné funkce ASP.NET 2.0 Membership .....	39
5.1.2	Stránka EditProfile.aspx .....	39
5.1.3	Stránka Register.aspx .....	39
5.1.4	Řídící prvek UserProfile.ascx .....	40
5.2	SEKCE ČLÁNKY .....	40
5.2.1	Uživatelského rozhraní .....	40
5.2.2	Vrstva přístupu k datům .....	41
5.2.3	Vrstva aplikační logiky .....	41
5.2.4	Uložené procedury .....	42
5.3	SEKCE ANKETY .....	42
5.3.1	Uživatelské rozhraní .....	42
5.3.2	Vrstva přístupu k datům .....	43
5.3.3	Vrstva aplikační logiky .....	43
5.3.4	Uložené procedury .....	43
5.4	E-MAILOVÝ ZPRAVODAJ .....	43
5.4.1	Uživatelské rozhraní .....	44
5.4.2	Vrstva přístupu k datům .....	44
5.4.3	Vrstva aplikační logiky .....	44
5.4.4	Uložené procedury .....	45
5.5	DISKUSNÍ FÓRUM .....	45
5.5.1	Uživatelské rozhraní .....	46
5.5.2	Vrstva přístupu k datům .....	47
5.5.3	Vrstva aplikační logiky .....	47
5.5.4	Uložené procedury .....	47
5.6	INTERNETOVÝ OBCHOD .....	47
5.6.1	Uživatelské rozhraní .....	48
5.6.2	Vrstva přístupu k datům .....	49
5.6.3	Vrstva aplikační logiky .....	49
5.6.4	Uložené procedury .....	50
<b>6</b>	<b>SPRÁVA PORTÁLU .....</b>	<b>51</b>
6.1	SPRÁVA UŽIVATELŮ .....	51
6.2	SPRÁVA ČLÁNKŮ .....	52
6.3	SPRÁVA DISKUSNÍCH FÓR .....	53
6.4	SPRÁVA ANKET .....	53
6.5	SPRÁVA NOVINEK .....	53
6.6	SPRÁVA OBCHODU .....	54
6.7	SCHVALOVÁNÍ PŘÍSPĚVKŮ .....	54
	<b>ZÁVĚR .....</b>	<b>55</b>
	<b>ZÁVĚR V ANGLIČTINĚ .....</b>	<b>56</b>

<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>57</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>58</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>59</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>60</b>



## ÚVOD

S rozvojem internetu v posledních několika desetiletích nastala potřeba kvalitně prezentovat téměř jakýkoliv obor, firmu či společnost pomocí webových stránek, které budou přehledně graficky zpracovány a poskytnou dostatečnou možnost interaktivity s uživatelem.

Rozvoj programovacích jazyků ke tvorbě webových stránek začal pomocí určených jednoduchých technologií, jako bylo HTML. Tyto jazyky jsou velmi přehledné a zvládnutelné i člověkem, který nestuduje programování a pokročilou práci s počítačem. S rozvojem ovšem přicházejí stále větší požadavky a webové stránky se pomalu stávají spíše aplikacemi, které se možnostmi i strukturou podobají stále více desktopovým aplikacím. Vznikají tak jazyky jako PHP a JSP, které nabízejí programátorovi větší množství nástrojů ke tvorbě inteligentní webové aplikace. Mezi další dekádu vývoje se považuje vznik ASP (active server pages) a ASP.NET. Tento programovací jazyk vyvinula společnost Microsoft, která se stará o jeho vývoj a propagaci. Technologie ASP byla spíše předchůdcem ASP.NET, která poskytovala sadu nástrojů pro vkládání dynamického obsahu do webových stránek, teprve s ASP.NET přichází rozvinutá platforma pro vývoj plnohodnotných a velmi rychlých aplikací. Mezi hlavní přednosti patří integrace jazyka C# a užití běhového prostředí .Net Framework. Ať už programátor ovládá Javu, C++, Visual Basic či jiný jazyk setká se v ASP.NET s podporou, která mu nabízí velkou kreativitu. V současné době je dostupná technologie ASP.NET již ve verzi 3.5, jež od původní verze nese řadu vylepšení a novinek.

Trendem současnosti není být správcem webového portálu a zároveň zručným programátorem, ale umožnit člověku, jež se o webovou aplikaci stará aby ji mohl spravovat bez sebemenších technických znalostí. Veškerá práce týkající se naprogramování a nasazení hotového webu na portál je na programátorovi. O údržbu a správu webu už se stará pouze redaktor nebo webový administrátor s příslušnými právy.

## I. TEORETICKÁ ČÁST

# 1 NÁVRH ŘEŠENÍ

## 1.1 Analýza odvětví trhu a požadavků zadavatele

Prezentace hudebního interpreta případně kapely na internetu se odlišuje od jiných zejména specifickým grafickým návrhem a použitím prvků webu jako je registrace uživatele, možnost hodnocení článků a recenzí, fórem pro registrované členy, možností automatického odběru informačních emailů a čtečkou RSS pro nejnovější zprávy. Je nutno také respektovat specifické požadavky zadavatele, který není technicky zdatný, a přesto požaduje, jako správce stránek prostředí ve kterém bude mít možnost plné administrace bez nutnosti stát se sebemenším programátorem a žádných technických dovedností.

Menší analýzou webu zahraničních webových serverů známých hudebních interpretů podobného žánru jsem vysledoval určité podobné prvky, zejména grafická stránka je vždy velmi přehledná a je uzpůsobená pro uživatele, kteří vědí, co hledají a neradi ztrácí čas průzkumem celého webu. Rovněž se často objevuje možnost nákupu nějakých maličkostí, lístků a suvenýrů takže v mém návrhu implementuji internetový obchod. Možnosti anket, komentářů ke článkům a recenzím přistoupilo rovněž většina webů takže je hodlám také implementovat.

Hlavním požadavkem zadavatele je nasazení webu na některý z českých hostingových serverů podporující technologii ASP.NET 2.0, který nabídne zpočátku hosting s databází zdarma a v budoucnu bude možno přejít na některou z placených hostingových služeb nabízející doménu druhého řádu - domena.cz.

## 1.2 Návrh řešení uživatelského prostředí

### 1.2.1 Grafické zpracování a design webu

V současné době se mezi vývojáři webových aplikací hojně využívají služby profesionálních návrhářů grafického designu a prvků. Může jít o přímou zakázku zadavatele o specifické rozložení a styl webu nebo je zde možnost výběru ze stovek předem vytvořených grafických šablon. Taková šablona vlastně vznikne celkovým

grafickým návrhem a jeho rozřezáním na části jako jsou hlavička, patička, menu, sloupce atd. Součástí takového celistvého návrhu je i soubor s příponou .css, který je již naprogramován a stará se o rozmístění všech prvků na stránce. I já sám jsem několik grafický šablon prošel, konkrétně se jedná o návrhy, jež jsou k dispozici na [www.templatemonster.com](http://www.templatemonster.com). Nakonec jsem zvolil alternativu vlastního grafického návrhu stránek s charakteristickým rozmístěním prvků podle šablony a tak jsem dostal ucelenou představu o tom, jak bude naše webová aplikace vypadat.

Rozhodující kritérium pro náš web je použití relativního centrálního pozicování webové stránky s pevnou a neměnnou šířkou 780 pixelů. Do této oblasti se nám bez problémů vejdu tři obsahové sloupce ohraničené hlavičkou nahoře a patičkou dole.

Levý obsahový sloupec o pevné šířce 201 pixelů bude sloužit zejména pro zobrazení anket, výsledků anket a bannerů odkazující na přátelené weby. Žádné další využití levého obsahového sloupce není požadováno.

Pravý obsahový sloupec o šířce 190 pixelů nabídne informativní pohled na nákupní košík a nabídne možnost zadat svůj email a nechat si zasílat informativní novinky do své emailové schránky.

Centrální sloupec disponuje největší šířkou 375 pixelů, pojme tak většinu informací a slouží pro zobrazení hlavních částí obsahu webu. Od ostatních dvou sloupců se liší svým mírným odsazením od hlavičky a to z toho důvodu použití dalšího velmi efektivního prvku a to je mapa webu. Tato „mapa“ nám neustále zobrazuje celou cestu, ve které části webové aplikace se zrovna nacházíme, přičemž základní sekcí je Úvod. Pokud si například budeme prohlížet recenze v sekci Články, zobrazí se nám naše cesta jako Úvod >> Články >> Prohlížení. Tento prvek je zvolen pro snadnou orientaci a výrazně zvyšuje efektivitu práce při prohlížení našich stránek.

Všechny tři výše popsané sloupce slouží k zobrazení specifických obsahů a jsou nahoře ohraničeny hlavičkou, která je jakýmsi grafickým reprezentantem webu a kromě loga názvu interpreta obsahuje ve své pravé části loginbox což je vstupní okénko pro přihlášení registrovaného uživatele. Těsně pod loginboxem se nachází roletové menu, které nabídne ze dvou možných variant motivů, které kompletně barevně promění celý web. Tato možnost je velmi vhodná například pro uživatele s horším zrakem, kterým může jiný motiv vyhovovat lépe.

Jako na většině přehledných webových aplikací najdeme pod hlavičkou menu, které disponuje rozdílnou grafickou úpravou pozadí. Zde máme na výběr ze všech částí webu a to: Úvod, Články, Obchod, Fórum, O Kapele, Kontakt a Spravovat, které je přístupné a viditelné pouze přihlášeným uživatelům a správcům webu.

Celý další obsah našich sloupců rozhoduje o výšce stránky, která se dynamicky umí přizpůsobit a je na spodní části ohraničena patičkou s grafickým pozadím. Patička nám opět nabízí stejné menu jako v horní části stránky ale navíc zobrazuje autorské práva a případně kontakt na webmastera, které najdeme ve většině profesionálních webových aplikací.

### **1.2.2 Registrace uživatele a editace obsahu webu**

Pro snadnou správu webu bylo využito propracovaného systému ASP.NET 2.0 pro registraci členů a správu rolí. Celkem je vytvořeno 6 různých rolí, z nichž každá má jinou úroveň možnosti zásahu do obsahu webu. Neomezené pravomoce jsou uděleny uživatelům, jejichž role se nazývá Administrators, dále mají možnost spravovat web Redaktoři a Správci. Mezi omezené role patří Diskutující, Obchodníci a Přispěvatelé. Pro přidělení nějaké role se musíte nejdříve nalogovat do systému pod svým jménem a heslem.

Vyšší pravomoc a postavení Správci uděluje kompetenci schvalovat příspěvky od Přispěvatele nebo jej případně zrušit za porušení pravidel našeho webu. Příspěvky do naší webové aplikace mohou mít několik podob. Mezi editovatelné sekce patří i sekce Články, do které je Správce schopen přidávat recenze, ke kterým mohou i ostatní uživatelé připojit svůj názor formou komentáře pod článkem.

Další volně editovatelnou sekcí je Fórum kde se zobrazí příspěvek ve formě vlákna na které je možno opovédět a navázat další diskuzi. Opět je zde možnost zásahu ze strany Správce a to formou editace obsahu odpovědi, případně i úplného odstranění reakce některého z diskutujících. Existuje i možnost vlákno uzamknout a tím zablokovat možnosti další diskuze či vytvořit fórum moderované. Moderované fórum má tu vlastnost, že všechny příspěvky před zveřejněním musí být schváleny Správcem v opačném případě nebudou zveřejněny.

Mezi správu obsahu patří samozřejmě i sekce Obchod, ve které je Správcem umožněno smazání, úprava nebo přidání nového zboží včetně popisu, dostupnosti i

fotografií. Ostatní uživatelé mohou na určitý typ zboží reagovat a přispívat svým hodnocením.

Poslední možností příspěvku a tím vlastní editace webové aplikace jsou ankety. Ankety se nacházejí v levém obsahovém sloupci a příchozímu uživateli je umožněno jednou hlasovat a vyjádřit tak svůj názor na zmíněné téma. Mezi hlavní motivaci uživatele ke svému hlasu patří okamžité zobrazení statistiky všech hlasování včetně grafické reprezentace. Samozřejmě jsou ankety plně editovatelné Správci. Lze ji smazat, změnit či přidat novou.

### 1.2.3 Editace textového obsahu webu

K práci na textovém obsahu webu byl zvolen a integrován uživatelsky přívětivý WYSIWYG editor. Což je vlastně i podle obsahu slov z názvu What You See Is What You Get nástroj, který nám umožní upravovat text jako v pokročilém textovém editoru. Výstupem z něj je náš text obalený nutným HTML kódem, který je pro webové prohlížeče nezbytný ke správnému zobrazení. K dispozici je několik variant těchto editorů, přičemž jsem zvolil nekomerční „open-source“ FCK editor, jež je volně ke stažení. Jak jsem již zmínil výše tento nástroj, je plně integrován do prostředí webové aplikace, ve které nám umožňuje opravdu velmi pohodlnou editaci textů v sekcích, jako jsou Články, Fórum a Správa.

### 1.2.4 Vkládání souborů

Jelikož nejen z textového obsahu je webová prezentace „živa“, je nutno zprostředkovat Správci vkládání „upload“ souborů na server a to co nejpřívětivějším způsobem. Jak jsem se již zmínil, správa webu se musí obejít bez technických znalostí Správce a tak je nutno vybudovat nějaké jednoduché rozhraní. Toto rozhraní je použito na stránce Správa, Produkty, přidat produkt. Jednoduchým způsobem zde zvolím tlačítko „Procházet“ a soubor, který chci umístit na server označím a potvrdím. V textovém rámečku se nám zobrazí zvolená cesta k souboru. Volbou tlačítka nahrát jednoduše pošleme soubor na server. Po skončení transferu se nám zobrazí pod textovým rámečkem potvrzující hlášení o tom, jestli proces proběhl úspěšně a ve kterém adresáři se nyní náš soubor nachází.

### 1.3 Použité programové vybavení

Jako nástroj k vývoji bude použit nástroj společnosti Microsoft, Visual Studio Express 2008 a MS SQL Server 2005 pro správu databáze. Vzhledem k nasazení hotové aplikace na hostingový server je nutno nejdříve zjistit jaké funkce máme k dispozici a které nástroje si mohou dovolit použít. Náš web co se týče velikosti a vytížení můžeme počítat mezi malé až střední webové aplikace. Celkový objem dat se dá počítat na desítky MB včetně použité databáze. K nasazení našeho webu plně vyhoví hostingový server [www.aspone.cz](http://www.aspone.cz), na kterém bude založen bezplatný účet.

Některé klíčové funkce našeho účtu:

- Diskový prostor: 40 MB
- Doména třetího řádu: [domena.aspone.cz](http://domena.aspone.cz)
- Podpora MS SQL 2008 databází
- Velikost databáze: 30MB
- Podporované technologie: ASP.NET 3.5
- ASP.NET AJAX
- Správa přes FTP
- Cena: zdarma

#### 1.3.1 Kaskádové styly (CSS)

Pro jednotný a dokonalý vzhled našich webových stránek je nutno zvolit nějaký jednotný vizuální styl. Jednotné formátování webových stránek nám velmi usnadňují kaskádové styly, jež jsou rozšířeny mezi všemi platformami a jsou podporovány všemi moderními i staršími prohlížeči. Cíle toho jazyka je specifikace způsobu vykreslení HTML- značek pomocí nejrůznějších stylistických prvků, jako je velikost fontu, barva, zarovnání atd. Aplikovat styly lze dvěma způsoby a to buďto přímo ve stránce anebo

nepřímo a uložit je zvlášť. Výhoda uložení souboru se styly nepřímo a ne v samotné stránce je jasná, budeme mít jen jeden soubor přehledně upravovatelný a udržovatelný a na jednotlivé části tohoto souboru se budeme odkazovat pomocí ID nebo jména přímo na stránce. Specifický soubor má příponu .CCS a svou strukturou připomíná funkce či třídy v jazyce C#.

Práce s kaskádovými styly rovněž výrazně zvyšuje výkon celé webové aplikace, jelikož prohlížeč si soubor se styly stáhne jen jednou a uloží jej do vyrovnávací paměti a rychlost načítání je výrazně rychlejší a efektivnější.

### 1.3.2 Jazyk ASP.NET

Jedná se o produkt firmy Microsoft, do kterého jsou dávány nemalé naděje budoucnosti vývoje webových aplikací. Modelu ASP.NET 2.0 předcházely méně dokonalé avšak o nic méně revoluční verze ASP.NET a ASP.NET 1.0. Mezi základní vlastnosti ASP.NET 1.0 patří:

- Jedná se úplný, objektově orientovaný programovací model s architekturou řízenou událostmi, založenou na ovládacích prvcích, což podporuje zapouzdření kódu a jeho opětovné využívání.
- Dává možnost psát kód ve kterémkoliv z podporovaných jazyků .NET: Visual Basic, C#, J# a mnoho dalších.
- Slouží jako platforma pro budování webových služeb, což jsou opětovně využitelné jednotky kódu, které mohou volat jiné aplikace přes platformy a hranice dané počítači.
- Stránky a komponenty se kompilují na požádání, a tudíž se neinterpretují pokaždé, když se použijí. Obsahuje vyladěný model pro přístup k datům a flexibilní ukládání dat do cache, aby bylo možné ještě více zvyšovat výkon.

Jazyk samotný ve své nové verzi ASP.NET 2.0 byl rozšířen o několik hodnotných vlastností jako jsou:



- Vzory stránek, které umožňují implementovat jednotný vzhled na více stránkách. Definice šablony master pages dále využijeme při navázání na ostatní stránky velmi jednoduše a efektivním způsobem.
- Ovládací prvky pro zdroje dat jako jsou GridView, DetailsView, MultiView, Wizard, Menu, SiteMap umí velmi efektivně zobrazovat a pracovat s daty z databáze.
- Personalizace je také nový prvek, jež usnadňuje uživatelům uzpůsobovat vzhled a chování stránek podle jejich potřeb a chutí.
- Bezpečnost a členství.
- Administrace pomocí uživatelsky přívětivého prostředí ASP.NET Configuration.
- Webové části umožňují uživatelům přesouvat jednotlivé části webu a měnit tak web prakticky podle chuti uživatele.

### 1.3.3 XML

Jedná se stejně jako v jazyce HTML o text psaný mezi značky oddělené úhlovými závorkami. Editace zdrojového kódu je možná i v nejobyčejnějším textovém editoru. Rozdílem mezi jazykem HTML je, že XML nemá stanovenou sadu pevných značek a umožňuje tak vytvářet naše vlastní značky, říkáme mu metajazyk. XML definuje některé pravidla syntaxe, které je nutno dodržovat, ale nabízí velkou volnost oproti HTML. Mezi výhody tohoto jazyka jednoznačně patří jednoduchost a lehká čitelnost. Výhody:

- Široké použití zejména na firemní půdě, kde je často XML první volbou pro sdílení dat.
- Rozšiřitelnost tohoto jazyka je opravdu neomezená z čehož plyne použitelnost na jakýkoliv druh dat při zachování nízkých provozních nákladů.
- Nástroje a příbuzné standarty.

Rámec .NET framework umožňuje manipulaci s daty XML pomocí jmenného prostoru System.XML a dalších.

S jazykem XML se při psaní naší webové aplikace setkám poměrně často například v konfiguračním souboru `web.config`.

#### 1.3.4 AJAX

Asynchronous JavaScript and XML je poměrně nová technologie, jejíž první publikace od Jasse Jamese Garretta v roce 2005 nesla název “Nový přístup k webovým aplikacím”. Jde o technologii vývoje webových stránek měnících svůj obsah bez nutnosti znovunačítání. Nutností je použití moderních prohlížečů což se nám odmění uživatelsky příjemnějším prostředím, které známe z desktopových aplikací. Mezi nevýhody patří změna zaběhlé koncepce `www` stránek a jejich navigací pomocí tlačítek zpět a dopředu. Jedná se o aplikace se složitou vnitřní strukturou. Další nevýhodou je nutnost rychlého připojení k internetu bez něhož může být odeslaný požadavek pomalejší odezvou, což může být pro netrpělivé uživatele zavádějící.

#### 1.3.5 SQL Server Express

Jedná se o rozsáhlý databázový nástroj jež je výborně integrován do prostředí MS Visual Studio. Nabízí možnost správy a tvorby relačních databází a úkony nad nimi. Mezi základní rysy patří adresáře *Tables*, ve kterém nalezneme veškeré vytvořené tabulky. *Database Diagrams* slouží k uchování relací mezi tabulkami. *Stored procedudes* se chová o uložené procedury pro práci s databází. Ve verzi Express je jako nástroj pro studenty a nadšence poskytován zdarma.

#### 1.3.6 C#

Jedná se objektově orientovaný programovací jazyk vyvinutý společností Microsoft se snahou o zjednodušení programování bez sebemenšího omezení možností. Společnost Microsoft cílí vývoj k vyšší interakci internetu s operačního systémem. Nejde však jen o jazyk samotný ale o celou koncepci rámce zvaného .NET Framework, což je vlastně běhovým prostředím jazyka C#. Mezi některé novinky v jazyce C# 2005 patří parciální třídy, jež umožňují dělení třídy jazyka do několika souborů. Generičnost umožňuje vytvářet třídy podporující silné přetypování a pokročilou práci s jinými třídami. Anonymní

metody a iterátory patří mezi další exotičtější novinky jazyka, které jsou zaměřeny spíše pro specialisty.

### 1.3.7 .NET Framework

Jedná se o běhové aplikační prostředí od společnosti Microsoft, které bys se dalo přirovnat k běhovému prostředí Java. Slouží hlavně k usnadnění práce programátorovi. Mezi základní rysy patří:

- Správa paměti, vytváření a rušení objektů
- Spouštění a zastavování vláken kódu
- Bezpečnost kódu a kontrola oprávnění k prováděným operacím
- Natahování potřebných knihoven a komponent do paměti apod.

CRL (Common language Runtime) je neviditelný prvek jež se stará o důležité operace. Nad tímto prvkem se nachází knihovna BCL obsahující nejčastější pomocné funkce pro práci se soubory, síťovou komunikaci, diagnostiku atd. Pro práci s daty s možností jejich XML prezentace se stará další knihovna ADO.NET. Další knihovnou je knihovna starající se o uživatelské rozhraní – Windows forms určená pro desktopové aplikace a ASP.NET pro webové aplikace. Důležitým rysem .NET Framework je jazyková nezávislost a podpora pro jazyky jako C#, C++, Java, Perl, Python a další.

## **II. PRAKTICKÁ ČÁST**

## 2 VOLBA PÍSEM A ZNAKOVÉ SADY

V dnešním světě není nutno brát ohled jen na vcelku zdravé a ničím neomezené spoluobčany, ale také na lidi mezi námi, kteří jsou nějakým způsobem postiženi a omezeni. Tito lidé mají stejné právo k přístupu k informacím a zábavě jako ostatní. Proto se v současné době přistupuje a větší měrou se dbá na přizpůsobení obsahu i těmto lidem. Co se týče webových aplikací, můžeme podstatnou měrou zjednodušit hlavně čitelnost textů a zřetelnost grafické stránky webu. Proto jsem se rozhodl do našich webových stránek implementovat možnost volby ze dvou motivů, z nichž jedem má atraktivnější grafickou úpravu a je tak použit jako výchozí a druhý motiv nabízí světlejší pozadí větší písma a tím snadněji čitelný obsah.

Ke splnění výše předepsaného požadavku použijeme dva rozdílně adresáře, v každém z nich budou jiné soubory s obrázky, kaskádovými styly (CSS) a skiny (.skin). K výběru mezi jednotlivými motivy nám nakonec poslouží řídicí prvek themeselector a roletové menu jež bude k dispozici pod přihlašovacím rámečkem uživatele. Po volbě jednotného vzhledu se nám již bude kompletně celý web zobrazovat námi zvoleným stylem.

### 2.1 Použití fontů

Volbou písma a fontu má smysl se zabývat vzhledem k tomu, že každá aplikace z například bankovního nebo zábavního sektoru žádá po textovém obsahu jiný typ reprezentace. Pro zrakově postižené spoluobčany je hodně důležitá změna velikosti fontu. Definice našeho písma najdeme již v souborech s kaskádovými styly. Jedná se o písma Verdana, Arial, Serif. S texty dále pracujeme a definujeme jejich velikosti a barvy např. pro nadpis jsou jiné proporce než pro menu. Vše je definováno dále v souboru .css. Mnohačetnými zásahy do tohoto souboru nakonec docílíme našeho požadovaného vzhledu fontů, barev a uspořádání. Návrhem fontů se zabývá typografie a jedná se o velmi obsáhlý obor. Nabízím základní přehled používaných písem vztahujícím s k webu, každé z těchto písem je samozřejmě možno použít jako tučné, s kurzívou nebo podtržené. Není ovšem vždy pravidlem, že naše písmo bude zobrazeno stejně, jako jsme měli v úmyslu, jelikož záleží na tom, jaký prohlížeč používáme a jaké znakové sady máme nainstalovány v našem systému. Generické – všeobecně podporované písma v HTML a CSS:

- Serif

- Sans-serif
- Cursive
- Fantasy
- Monospace

Další fonty, jež jsou podporovány většinou prohlížečů (IE, Mozilla, Firefox, NetCaptor) :

- Times
- Roman
- Garamond
- Palatino
- Antiqua
- Minion
- Helvetica
- Swiss
- Impact
- Script
- Decorative
- Blackletter
- Fraktur
- Comic Sans
- Modern
- Couries
- Calibri
- Frosty

## 2.2 Volba znakové sady

Další důležitou volbou pro náš web je volba znakové sady. Jedná se vlastně o kódování sekvencí znaků z dané množiny, jednoduchým příkladem může být znaková sada ASCII. Principem kódování je vlastně přiřazení jednotlivých kódů znakům (character code) za pomoci určité znakové sady (character set), která se stará o to kterému znaku přísluší který kód. Ve světě se objevuje velké množství znakových sad, a proto vznikla organizace IANA, která se stará o seznamy doporučených znakových sad pro webové aplikace a internet.

Pokud bychom naši webovou aplikaci chtěli prezentovat v celosvětovém měřítku což není podmínkou bylo by důležité brát hodně ohledu nad správnou volbou znakové sady. Mezi základní standardní znakové sady patří ISO-8859-2, která se dá považovat za tím za nepodporovanější, ale jen z hlediska toho že zde byla nejdříve a tudíž má podporu i starších prohlížečů. Další sadou je Windows-1250 jež vznikla ve společnosti Microsoft kvůli podpoře východoevropských jazyků v MS Windows 3.0. Název této znakové sady je ovšem trochu zavádějící jelikož v současné době funguje multiplatformě. Pro nasazení naší aplikace s podporou více jazyků by byla nejvhodnější sada UTF-8, která je kompatibilní s ASCII a disponuje nejmenší kódovou jednotkou délky 8 bitů. Pro náš web jsem s ohledem na požadavky použil sadu Windows-1252, která nečiní problémy většině prohlížečů.

Naši volbu znakové sady umístíme v našem projektu do hlavičky vzorové stránky (master page). Syntaxe je následující: `<meta http-equiv="content-Type; charset=Windows-1252"`. Pro náš účel je tento typ znakové sady nejvhodnější.

### 3 VZOROVÁ STRÁNKA A MOTIVY

Charakteristický prvek ASP.NET 2.0, který hodlám využít je definice jednotného vzhledu pomocí jedné vzorové stránky na kterou se budou všechny ostatní obyčejné stránky odkazovat a zdědí tak společný vzhled. Jak jsem se zmínil již výše je třeba začlenit do naší webové podporu i pro zrakově slabší spoluobčany a implementovat volbu ze dvou zcela jiných grafických podob naší prezentace. Písma jsem již probral výše. Nyní je nutné zaměřit se na uspořádání jednotlivých bloků, pozadí a grafiky.

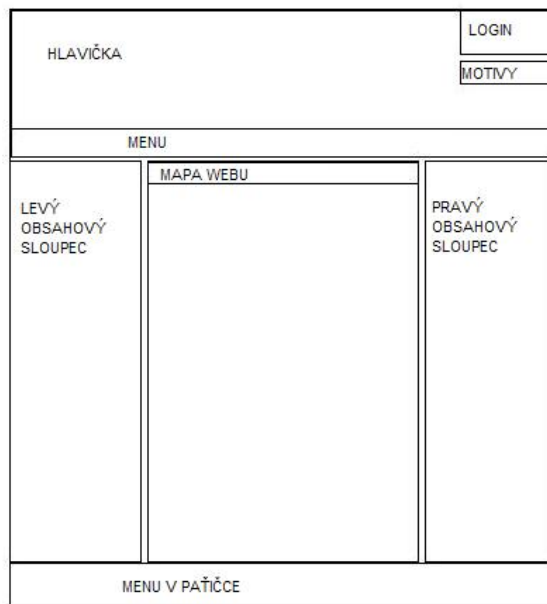
O základní uspořádání prvků na stránce se stará prvek ASP.NET 2.0 tzv. Master page – vzorová stránka všem dalším obyčejným stránkám. Jedná se o jednu z unikátních vlastností ASP.NET 2.0. Technologie ASP.NET 2.0 zaujímá vzhledem k fungování naší webové aplikace prvotní místo. Běží na webovém serveru a s výhodou využívá funkční prvky poskytované rámcem .NET Framework. Nicméně ASP.NET neběží na počítači uživatele, ale dynamicky generuje prvky, které používá prohlížeč pro vykreslování webových stránek. Patří mezi ně HTML-značky, obrázky a definice kaskádových stylů (CSS), které definují barvy, velikost a polohy nejrůznějších objektů na HTML-stránce. ASP.NET dále generuje procedurální kód jazyka JavaScript, který posílá prohlížeči pro kontrolu uživatelem zadaných údajů také pro ustavení způsobu interakce s webovým serverem.

#### 3.1 Master Page

První věcí, kterou většinou začíná programování a návrh projektu obecně je definice vzorové stránky. V našem projektu se jedná o stránku o Template.master se souborem kódu v pozadí Template.master.cs, který obsahuje kód v jazyce C#. Tyto dvě stránky jsou společně logicky svázané. Definice souboru Template.master sestává z definice hlavičky, těla a značkovacího kódu obvyklého z kódu HTML. Rozdílem mezi webovým formulářem což je vlastně základní stránka jež má příponu .aspx se souborem kódu v pozadí .cs je, že master page začíná direktivou Master. Direktiva je deklarována na začátku stránky kódem `<%@ Master Language="C#" ...atd %>`. Vedle direktivy jsou použity ještě další parametry ze kterých jsem ponechal v ukázce pouze "Language", který určuje v jakém jazyce píšeme kód v souboru v pozadí - .cs.



Dalším podstatným rozdílem ve vzorové stránce je to, že používá ovládací prvek ContentPlaceholder jež je v obyčejných stránkách zakázán. Je to velmi užitečný prvek, který definuje tu část stránky do které potom může stránka s obsahem ukládat obsah.



Obr. 1. Návrh rozložení jednotlivých prvků na stránce

### 3.1.1 Záhloví a hlavička Master page

Jak jsem se již zmínil podoba vzorové stránky template.master je podobná kódu HTML. Nyní se podívám podrobněji na strukturu obsahu záhlaví a hlavičky. Jako první zde obvykle najdeme direktivu Master popsanou výše, definice jazyku kterým programujeme soubor v pozadí. Pro náš projekt jsem zvolil jazyk C#, ale samozřejmě je i možnost programování v jazyce Visual Basic. Dalším důležitým atributem je atribut CodeFile jež odkazuje na soubor s doprovodným kódem. Mezi další atributy patří Inherits a AutoEventWireup. Dalším obvyklým prvkem známým i z HTML je definice hlavičky v párovém tagu <head> v němž deklarujeme id hlavičky. Vnořený může být i titulek <Title> jež je zapsán mezi párovým tagem.

### 3.1.2 Tělo Master Page

V těle jež je jasně deklarováno párovým tagem <body> nadefinujeme nejdříve id formuláře nepárovým tagem <form>. Dále již pokračuji v kompletním návrhu prvků stránky. Zde využiji takzvaných kontejnerů, jež mají formát párového tagu <div id="...">. Pomocí id obsaženého v kontejneru se odkazují na jednotlivé části souboru s kaskádovými

styly. Jak je uvedeno v návrhu výše na webových stránkách chci využívat a zobrazit hlavičku (konkrétně definovanou jako *Header*), přihlašovací rámeček (*loginbox*), výběr motivů (*themeselector*), menu s odkazy na jednotlivé části webu (*menu*), levý obsahový sloupec (*leftcol*), pravý obsahový sloupec (*rightcol*), centrální obsahový sloupec (*centercolcontent*), mapu kde se webové aplikaci nacházím (*breadcrumb*), patičku (*footer*), menu v patičce (*footermenu*), text v patičce (*footertext*). Všechny tyto prvky se nacházejí v souboru kaskádových stylů, kde je určena pomocí různých parametrů jejich rozmístění.

Abych dosáhl požadovaného uspořádání vzorové stránky a tím vlastně všech stránek musím, ještě navíc definovat další prvky jež budou všechny součásti vypsané výše držet pohromadě a nedovolí jim “uskakovat”. Hlavní prvek bude mít id *content*, tehle se postará o to aby byla výsledná webová aplikace vždy vycentrována doprostřed prohlížeče a měla pevnou šířku. Posledním novým prvkem je *centercol* jež bude pevně držet centrální sloupec pohromadě s prvkem mapy webu (*breadcrumb*).

Nyní mám všechny prvky definované a naprogramované v souboru s kaskádovými styly takže stačí už je jen vhodně uspořádat do těla *template.master*. Nejdříve pochopitelně ohraničím prostor stránky a její centrální pozici prvkem *centercol*, a pokračuji dále přičemž do každého prostoru, který chci definovat vnořím všechny potřebné prvky.

Mezi nejdůležitější části do kterých budou vkládány obsahy všech dalších obsahových stránek musím umístit ovládací prvek *ContentPlaceHolder*. Celkem jsem použil a pro můj účel upotřebím celkem tři ovládací prvky typu *ContentPlaceHolder* a to v levém, centrálním a pravém sloupci.

Tělo stránky *master page* končí standartním uzavřením tagů *form*, *body* a *html*.

## 3.2 Volba motivů

Nyní se již pustím do realizace dvou graficky rozdílných motivů. Celá tato operace souvisí s vytvořením hlavního adresáře (*App\_Themes*) v němž vytvořím dva adresáře s názvem *svetly motiv* a *TemplateMonster*. Oba adresáře obsahují složku s vlastními obrázky a grafikou (*images*), vlastní soubor s kaskádovými styly *Default.css* a souborem *Control.skin*. Soubor s příponou *.skin* je charakteristickou vlastností v ASP.NET a jde vlastně o jeden soubor, ve kterém můžeme definovat vzhled ovládacích prvků, jako jsou

GridView, Calendar, DataGrid a mnoha dalších. Krom těchto prvků bude ještě použit ovládací prvek *themeselector*, který nám poslouží k dynamické změně vzhledu za běhu webové aplikace a nová třída definující společné vlastnosti mé aplikace.

### 3.2.1 Třída BasePage

Pro společné chování stránek, ale musíme udělat ještě více, jelikož společný vzhled a motivy stránek fungují většinou spolehlivě, já ale potřebuji také sdílet společné chování kódu stránek jež běží v pozadí a který poběží v určité fázi životního cyklu. Jedná se například o záznam přístupů na všechny stránky, statistiky atd. Je několik způsobů jak toho dosáhnout. Já jsem zvolil vytvoření třídy *BasePage* (*BasePage.cs*) v adresáři *App\_Code*. Tato třída se mě postará o to aby všechny doprovodné třídy dědily místo ze standartní třídy *System.Web.UI.Page* přímo z mojí třídy *BasePage*. V rámci této třídy máme možnost reagovat na jakoukoliv událost jen tím, že přetížíme příslušnou metodu ve tvaru *OnJmeno\_požadované\_události*. Prakticky potom všechny soubory v pozadí s doprovodnými kódy stránek odvodím od mé třídy *BasePage*.

### 3.2.2 Řídící prvek ThemeSelector

Tento ovládací prvek se stará o zobrazení a přepínání motivů za běhu webové aplikace. Řídící prvky mají obecně příponu *.ascx* a soubor kódu v pozadí s příponou *ascx.cs*. Nejprve je nutno si vytvořit adresář, který nazvu *Controls*, a do nějž budu později vkládat všechny moje nové řídicí prvky. Uvnitř souboru *.ascx* nalezneme pouze definici roletového menu což je řídicí prvek typu *DropDownList*. Další zajímavou funkcí, jež využiji je funkce *AutoPostBack*, kterou nastavím na hodnotu *true*. Tato se nám postará o automatické zaslání celé stránky na server po volbě motivu uživatelem. V souboru s doprovodným kódem naprogramuji zbytek aplikační logiky, aby došlo k naplnění roletového menu.



Obr. 2. Návrh přepínače

*motivů*

### 3.3 Tvorba Menu

Posledním důležitým prvkem vzorové stránky je vytvoření menu v hlavičce a patičce stránky. Menu obvykle slouží k nejpodstatnějším prvkům webových stránek a mívá různé podoby. Někdy se lze setkat s vertikálně či pozicovaným menu anebo různými grafickými „výstřelky“, pomocí kterých se pohybujete po celé webové aplikaci. Rozhodl jsem se pro jednu z nejoblíbenějších variant a tou je horizontálně centrované menu v hlavičce a patičce naší stránky.

Ke splnění toho v ASP.NET 2.0 poměrně jednoduchého úkolu využiji řídicí prvek typu SiteMapPath. Soubor s příponou .sitemap je nutno přidat do kořenového adresáře našeho projektu. Pojmenoval jsem jej *web.sitemap* a do jeho výchozí struktury, jež se nám vygeneruje už po otevření tohoto souboru jsem doplnil sekce, jež budou odkazy na naše jednotlivé stránky webu. Tyto titulky jsem nazval: *Úvod, Obchod, Košík, Fórum, O kapele, Kontakt, Spravovat*. Ke každému titulkovi pomocí atributu url přidám později odkazy na existující stránky. Nutno poznamenat, že tento soubor nemá soubor s kódem v pozadí.

### 3.4 Tvorba první stránky Default.aspx

K tomu abych otestoval, jak se mi podařilo implementovat návrh, všechny vlastnosti a charakter mojí webové aplikace je nutno vytvořit obyčejný webový formulář s příponou .aspx a souborem kódu v pozadí s příponou .aspx.cs. Při tvorbě této stránky ve Visual Studiu je nutno po zadání volby jazyku C# nebo VisualBasic v němž bude programován kód v pozadí dále zvolit název a potvrdit dialog spojující tuto stránku se vzorovou stránkou Template.master.

Struktura souboru je poté velmi jednoduchá. Direktiva Page na začátku stránky vypovídá o typu souboru, dále najdeme jako obvykle odkaz na soubor kódu v pozadí (*default.aspx.cs*), pomocí atributu *CodeFile*. Důležitým atributem je *MasterPageFile*, který odkazuje na naši vzorovou stránku. Do souboru je nyní možno zasahovat jen v rámci řídicích prvků *Content* což jsou vlastně odkazy na prvky typu *ContentPlaceHolder*, které

jsme dříve definovali v naší vzorové stránce. Z toho logicky plyne, že stránka již neobsahuje žádné značky typu `<body>` atd. K souboru v pozadí bych dodal snad je to, že dědí z naší bazové třídy *BasePage*.

## 4 DATABÁZE

Volba databáze, obecně datového úložiště záleží na mnoha faktorech. V případě, že by náš web byl statický, tedy bez častých změn, pak se nabízí XML-soubory nebo formát databáze Access, což by nebylo vhodné řešení jelikož se jedná o desktopovou databázi nepodporující přístup více uživatelů. Dále se nabízejí databáze jako jsou SQL Server, Oracle, DB2, MySQL nebo Progress. Ve své podstatě jsou použitelné všechny a záleží jen na uživateli či firmě, kterou z těchto moderních databází preferuje z jakéhokoliv důvodu.

Vzhledem k tomu, že můj projekt je vyvíjen na platformě .NET a s použitím MS Visual Studia, bude nejjednodušší volba šáhnout po databázi SQL Server společnosti Microsoft, která je výborně integrována do tohoto prostředí a nabízí vysoký výkon. Konkrétně se bude jednat o Microsoft SQL Server 2005 Express Edition, který je jako produkt zdarma a je určen studentům a začínajícím vývojářům. Tato databáze nabízí všechny dostupné funkce a její omezení spočívá jen v podpoře pouze jednoho procesoru, 1GB paměti RAM a databázi o maximální velikosti 4GB. Rovněž nejsou podporovány některé pokročilé prvky, které ovšem neupotřebím jako např. dělení databáze, zrcadlení databáze, notifikační služby a full-textové vyhledávání. Mezi výhody této databáze patří například způsob nasazení, jež umožňuje pohé překopírování souboru s příponou .MDF do specifického adresáře *App\_Data* a následné propojení specifikací cesty k souboru pomocí atributu *AttachDBFilename* v řetězci popisujícím připojení k databázi.

### 4.1 Vytvoření databáze

Po přidání databáze (konkrétně *ASPNETDB.MDF*) do našeho projektu využiji Database Explorer, což je funkce integrovaná přímo ve Visual Studiu. Po rozbalení naší databáze uvidíme několik podadresářů se specifickými jmény: *Database Diagrams*, *Tables*, *Views*, *Stored Procedures*, *Functions*, *Synonyms*, *Types*, *Assemblies*. Během vývoje projektu použiji adresář *Tables*, kde definuji své tabulky včetně jejich částečného naplnění. Dále použiji *Database Diagrams*, kde definuju svázání jednotlivých tabulek a dále *Stored Procedures*, což jsou uložené procedury pracující s databází.

### 4.2 Struktura tabulek

Vzhledem k velikosti a obsáhlosti webu musím vytvořit 14 tabulek, které se budou týkat nějakým způsobem každé sekce webové aplikace. Použiji přitom datové typy

určitých položek. K uložení většího obsahu informací, konkrétně více než 8kB, poskytuje SQL Server 2005 tři nové datové typy a to jsou: VARCHAR(MAX), NVARCHAR(MAX) a VARBINARY(MAX).

Nyní popíšu jednotlivé tabulky, vynechám použité datové typy jednotlivých položek, které budou graficky znázorněny později v databázových diagramech včetně propojení tabulek do vzájemných relací.

#### 4.2.1 Tabulky a jejich položky

Tabulka *tbh\_Categories* uchovává údaje o kategoriích článků. Najdeme zde položky *CategoryID* jež je jedinečné ID kategorie, *AddedDate* Datum a čas vytvoření kategorie, *AddedBy* jméno uživatele, který kategorii vytvořil, *Title* název kategorie, *Importance* důležitost kategorie, použito k uspořádání, *Description* popis kategorie, *ImageUrl* url-adresa obrázku reprezentujícího kategorii.

Tabulka *tbh\_Articles* pojímá obsah všech článků a veškeré informace vrámci kategorií. *ArticleID* jde o jedinečné id článku, *AddedDate* datum a čas předán článku, *AddedBy* jméno uživatele jež článek vytvořil, *CategoryID* id kategorie do níž článek patří, *Title* název článku, *Abstrakt* krátké shrnutí článku, *Body* obsah článku, *Country* země týkající se článku, *State* kraj jehož se článek týká, *City* město týkající se článku, *ReleaseDate* čas a datum zveřejnění článku, *ExpireDate* datum a čas stažení článku, *Approved* stav schválení článků, *Listed* viditelnost článku v seznamu na stránce s články, *CommentsEnabled* povolení komentářů od uživatelů, *OnlyForMembers* pouze pro registrované a autentizované uživatele, *ViewCount* počet shlédnutí článku, *Votes* počet hlasů jež článek obdržel, *TotalRating* celkové hodnocení článku.

Tabulka *tbh\_Comments* je úložištěm pro zpětnou vazbu jako například komentáře, otázky a odpovědi pro zveřejněné články. Jednotlivé položky a jejich použití: *CommentID* jedinečné id komentáře, *AddedDate* datum a čas vložení komentáře, *AddedBy* jméno autora komentáře, *AddedByEmail* emailová adresa uživatele, *AddedByIP* IP adresa uživatele, *ArticleID* článek jež se komentář týká, *Body* vlastní text komentáře.

Tabulka *tbh\_Pools* nám slouží uchování informací o anketách. Význam jednotlivých položek je zcela zřejmý a podobný jako u tabulek pracujících s články, proto

se zmíním jen o nejzajímavějších položkách. Jednotlivé položky: *PoolID*, *AddedByDate*, *AddedBy*, *QuestionText*, *IsCurrent*, *IsArchived*, *ArchivedDate*.

Tabulka *tbh\_PollOptions* je druhou a zároveň poslední, která slouží k práci s anketama a obsahuje všechny konfigurovatelné volby pro každou anketu. Obsahuje položky: *OptionID*, *AddedDate*, *PoolID*, *OptionText* a *Votes* jež obsahuje počet hlasů, jež tato volba od uživatele obdržela.

Tabulka *tbh\_Newsletters* se stará o sekci našeho webu, jež má na starosti rozesílání novinkových a informační emailů dle přání uživatele. Obsahuje položky: *NewsletterID*, *AddedDate*, *AddedBy*, *Subject*, *PlainTextBody*, *HtmlBody*.

Tabulka *tbh\_Forums* je jednou z tabulek starajících se o příspěvky v sekci fórum. Její jednotlivé položky jsou: *ForumID*, *AddedDate*, *AddedBy*, *Title*, *Moderated* se stará o to jestli mají být jednotlivé příspěvky před jejich zveřejněním schváleny, *Importance*, *Description*, *ImageUrl*.

Tabulka *tbh\_Posts* je druhou tabulkou starající se o fóra. Všechny její položky: *PostID*, *AddedDate*, *AddedBy*, *AddedByIP*, *ForumID*, *ParentPostID*, *Title*, *Body*, *Approved*, *Closed*, *ViewCount*, *ReplyCount*, *LastPostBy*, *LastPostDate*.

Tabulka *tbh\_Departments* a ostatní další tabulky jež budou uvedeny níže se starají o poslední sekci mé webové aplikace a tou je malý internetový obchod. Položky: *DepartmentID*, *AddedDate*, *AddedBy*, *Title*, *Importance*, *Description*, *ImageUrl*.

Tabulka *tbh\_Products* starající se o zboží a produkty jež chceme prodávat. Její položky: *ProductID*, *AddedDate*, *AddedBy*, *DepartmentID*, *Title*, *Description*, *SKU* číslo modelu produktu, *UnitPrice*, *DiscountPercentage*, *UnitsInStock*, *SmallImageUrl*, *FullImageUrl*, *Votes*, *TotalRating*.

Tabulka *tbh\_Orders* se stará o objednávky v našem obchodě. Položky: *OrderID*, *AddedDate*, *AddedBy*, *StatusID*, *ShippingMethod* udává způsob dopravy, *SubTotal*, *Shipping*, *ShippingFirstName*, *ShippingLastName*, *ShippingStreet*, *ShippingPostalCode*, *ShippingCity*, *ShippingState*, *ShippingCountry*, *CustomerEmail*, *CustomerPhone*, *CustomerFax*, *ShippedDate*, *TransactionID*, *TrackingID*.

Tabulka *tbh\_OrderItems* jež obsahuje informace o objednaném zboží obsahuje položky: *OrderItemID*, *AddedDate*, *OrderID*, *ProductID*, *Title*, *SKU*, *UnitPrice*, *Quantity*.



Tabulka `tbh_OrderStatuses` uchovává informace o stavu objednávek. Položky: `OrderStatusID`, `AddedDate`, `AddedBy`, `Title`.

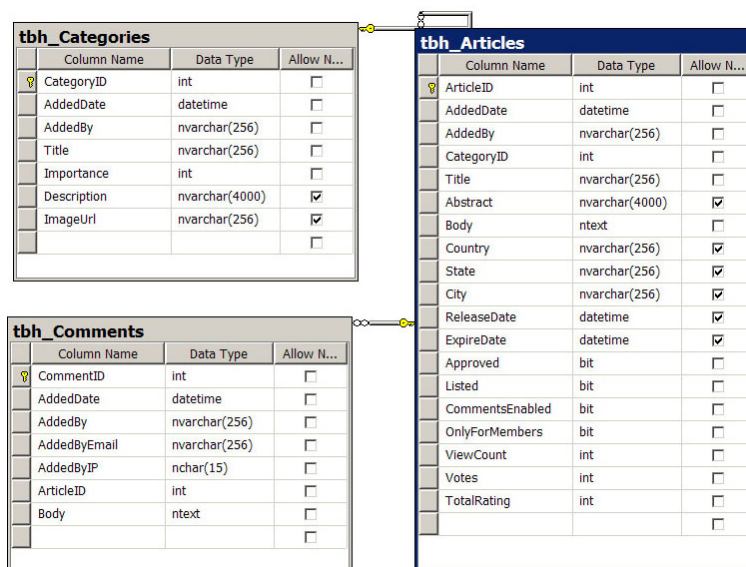
Tabulka `tbh_ShippingMethods` uchovávající jen pár položek slouží ke shromažďování celkové ceny a přepravní metody. Položky: `ShippingMethodID`, `AddedDate`, `AddedBy`, `Title`, `Price`.

### 4.3 Databázové diagramy

Abych uvedl jednotlivé tabulky do provozu musím mezi nimi určit tzv. relační vztahy. Těmito relacemi dostanou tabulky určité závislosti mezi sebou a jsou schopny pracovat efektivně. K vytvoření závislosti musím u každé tabulky určit primární klíč tzv. *Primary Key* a další klíče u jednotlivých položek, které budu chtít logicky provázat. Tato technika je obecně používána v relačních databázích.

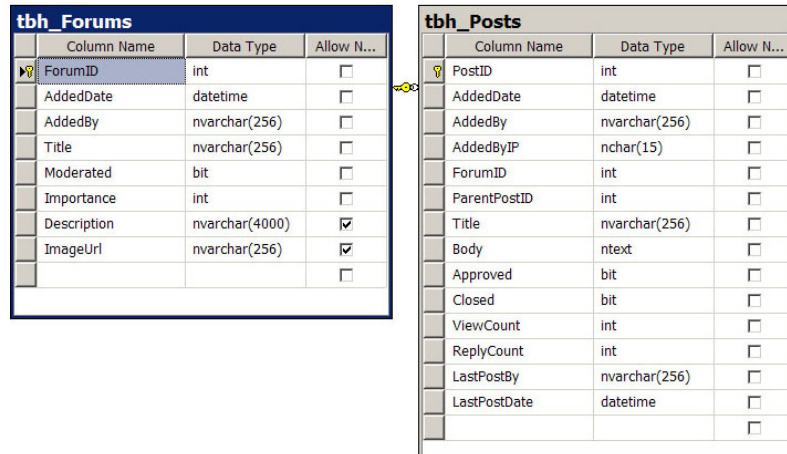
Pro můj projekt potřebuju vytvořit čtyři databázové diagramy, každý týkající se určité sekce naší webové aplikace. Tvorba pomocí nástrojů v databázovém exploreru je velmi jednoduchá a intuitivní, ve složce *Database Diagrams* zvolím vytvořit nový diagram a myší vložím tabulky jež chci propojit, následné propojení lze realizovat opět jednoduchými tahy myši. Následující grafické zobrazení schémat obsahuje jednotlivé položky včetně jejich datových typů.

#### 4.3.1 Diagram sekce Články



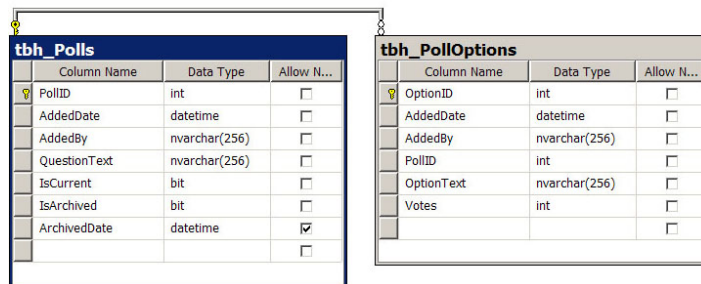
Obr. 3. Schéma relačního spojení tabulek článků

### 4.3.2 Diagram sekce Forum



Obr. 4. Schéma relačního spojení tabulek fóra

### 4.3.3 Diagram sekce Ankety



Obr. 5. Schéma relačního spojení tabulek anket

#### 4.3.4 Diagram sekce Obchod

Table Name	Column Name	Data Type	Allow Nulls
tbh_Departments	DepartmentID	int	<input type="checkbox"/>
	AddedDate	datetime	<input type="checkbox"/>
	AddedBy	nvarchar(256)	<input type="checkbox"/>
	Title	nvarchar(256)	<input type="checkbox"/>
	Importance	int	<input type="checkbox"/>
	ImageURL	nvarchar(4000)	<input type="checkbox"/>
tbh_Products	ProductID	int	<input type="checkbox"/>
	AddedDate	datetime	<input type="checkbox"/>
	AddedBy	nvarchar(256)	<input type="checkbox"/>
	DepartmentID	int	<input type="checkbox"/>
	Title	nvarchar(256)	<input type="checkbox"/>
	Description	ntext	<input type="checkbox"/>
	SKU	nvarchar(50)	<input type="checkbox"/>
	UnitPrice	money	<input type="checkbox"/>
	DiscountPercentage	int	<input type="checkbox"/>
	UnitsInStock	int	<input type="checkbox"/>
	SmellImageURL	nvarchar(256)	<input type="checkbox"/>
tbh_Orders	OrderID	int	<input type="checkbox"/>
	AddedDate	datetime	<input type="checkbox"/>
	AddedBy	nvarchar(256)	<input type="checkbox"/>
	StatusID	int	<input type="checkbox"/>
	ShippingMethod	nvarchar(256)	<input type="checkbox"/>
	SubTotal	money	<input type="checkbox"/>
	Shipping	money	<input type="checkbox"/>
	ShippingFirstName	nvarchar(256)	<input type="checkbox"/>
	ShippingLastName	nvarchar(256)	<input type="checkbox"/>
	ShippingStreet	nvarchar(256)	<input type="checkbox"/>
	ShippingPostalCode	nvarchar(50)	<input type="checkbox"/>
	ShippingCity	nvarchar(256)	<input type="checkbox"/>
	ShippingState	nvarchar(256)	<input type="checkbox"/>
	ShippingCountry	nvarchar(256)	<input type="checkbox"/>
	CustomerEmail	nvarchar(256)	<input type="checkbox"/>
tbh_OrderItems	OrderItemID	int	<input type="checkbox"/>
	AddedDate	datetime	<input type="checkbox"/>
	AddedBy	nvarchar(256)	<input type="checkbox"/>
	Quantity	int	<input type="checkbox"/>
tbh_OrderStatuses	OrderStatusID	int	<input type="checkbox"/>
	AddedDate	datetime	<input type="checkbox"/>
	AddedBy	nvarchar(256)	<input type="checkbox"/>
	Title	nvarchar(256)	<input type="checkbox"/>
	Price	money	<input type="checkbox"/>

Obr. 6. Schéma relačního spojení tabulek obchodu

#### 4.4 Uložené procedury

Pro přístup do databáze se používají dva různé návrhové modely. První spočívá v uložení všech používaných SQL-příkazů do uložených procedur, přičemž se objekt *SqlCommand* použije pouze pro jejich spouštění dle jmen. Princip druhého modelu tkví v uložení používaných SQL-příkazů ve formě textu, který se poté standartně spouští pomocí objektů typu *SqlCommand*. V mém projektu budu používat první způsob a s databází budu pracovat pomocí uložených procedur. Tyto umožňují jemnější řízení bezpečnostního přístupu k datům, poněvadž můžu uživatelům webového SQL-účtů přiřadit oprávnění provádět pouze jisté procedury, aniž by měli plnou kontrolu nad databázovými tabulkami. Další výhodou tohoto způsobu je fakt, že obvykle neobsahují jen jeden příkaz, ale hned celou dávku příkazů. Výhoda z toho plynoucí je menší síťový provoz a volání procedury pouze jejím krátkým jménem.

Uložené procedury poskytují možnost pro další vrstvení kódu. Při jejich používání se kód datové vrstvy sotva dotýká databáze a pouze vytváří příkaz odkazující na uloženou proceduru, nastaví jeho parametry a vše odešle k provedení. Datová vrstva tak neobsahuje

vůbec žádný SQL kód, což je často žádoucí, neboť nám to umožňuje nasazení přeložené datové vrstvy, přičemž později můžeme něco měnit v SQL kódu uložené procedury, aniž by to měla jakýkoliv vliv na datovou vrstvu, kterou tak již není potřeba znovu překládat a distribuovat.

V mém projektu použiji celkem 98 uložených procedur, které obsahují jednoduché SQL dotazy nad tabulkama v databázi. Jednotlivé procedury a jejich názvy vypíši v kapitole logické rozdělení a architektura.

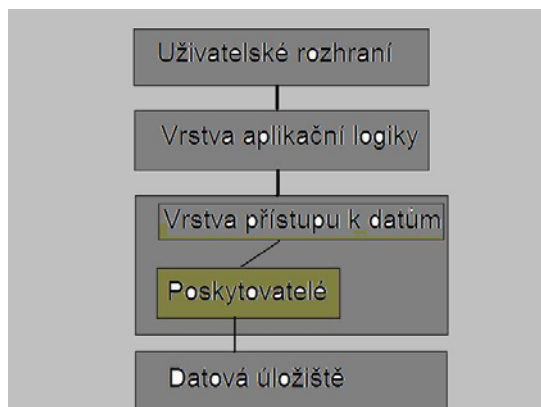
## 5 LOGICKÉ ROZDĚLENÍ A ARCHITEKTURA

Mezi důležité kroky v pokračování a zdárného splnění projektu patří návrh hlavních stránek, které budou reprezentovat každou sekci webu jejich provázání. Každá z nich samozřejmě bude obsahovat odkazy na další stránky, jež se budou týkat jen jedné sekce. Důležité je rozhodnutí i o jejich umístění, v kořenovém adresáři se budou nacházet veřejně přístupné stránky a v adresáři *Admin* už půjde o stránky, jež budou přístupné pouze správci webu. Veškerá konfigurační nastavení jsou deklarována v souboru *web.config*, což je konfigurační soubor mého projektu.

O celkové logické provázání všech stránek, jejich správnou funkci a přístup do vlastní databáze se musí postarat soubory tříd, které budu umísťovat do adresáře *App\_Code*. V tomto adresáři budou zároveň dva podadresáře, jeden DAL (data access layer), který se postará o přístup k datům, druhý BLL (business logic layer) bude obsahovat soubory pro práci s aplikační logikou. Takovému návrhu se často říká víceúrovňový nebo vícevrstvý návrh. V poslední době se tento návrh v oblasti vývoje software stal velmi známý. Stručně řečeno rozděluje funkční prvky, komponenty a kód projektu do samostatných úrovní. Na obecné úrovni existují čtyři:

- **Datové úložiště:** jedná se o relační databázi, XML, textový nebo jiný soubor.
- **Vrstva přístupu k datům (DAL)** : manipulace a získávání dat z datového úložiště.
- **Vrstva aplikační logiky (BLL)** : kód, jež přebírá data získané datovou úrovní a nabízí je klientům v abstraktní a intuitivní formě.
- **Prezentační vrstva (UI)** : kód, který definuje, co uživatel uvidí na obrazovce, včetně formátu dat atd., v mém případě se jedná o návrh pro webový prohlížeč.

Můj webový projekt si nyní rozdělím na části, jež postupně navrhnu a naprogramuji pomocí dostupných prvků a již navržené databáze.



Obr. 7. Schéma vrstvého návrhového vzoru

## 5.1 Uživatelské profily

V mojí webové aplikaci chci umožnit uživatelům registraci a členství. Nejvyšší pravomoce budou přiděleny správci, který bude spravovat celý obsah www stránek, včetně možnosti správy uživatelů, možnosti mazání nebo schvalování příspěvků nebo článků. Poté co se uživatel zaregistruje, bude mít jisté výhody oproti neregistrovaným uživatelům. Například pokud je registrovaný uživatel z okresu Zlín budou se mu zobrazovat novinky pouze z jeho okolí. Můj systém pro evidenci členů by měl mít následující vlastnosti:

- Uživatelé budou mít možnost samostatné tvorby účtů pouhým vyplněním registračního formuláře na webové stránce.
- Dále mají možnost měnit své přihlašovací údaje či jejich opětovné získání v případě, že je zapoměli.
- Správce má plnou kontrolu nad uživatelem a je schopen povolit či zamítnout přístup do specifických částí nebo na jednotlivé stránky. Je schopen dočasně nebo trvale vyřadit účet uživatele.
- Správce musí mít přehled o souhrnných a statistických údajích, jako je celkový počet registrovaných uživatelů nebo kolik z nich je v konkrétním čase přihlášených.
- Systém profilů bude uchovávat informace o uživatelích v datovém úložišti.

K řešení tohoto problému použiju pokročilé funkce ASP.NET 2.0 jako jsou bezpečnost a profily uživatelů.

### 5.1.1 Vestavěné funkce ASP.NET 2.0 Membership

K realizaci tohoto úkolu jsou předem připraveny prvky zjednodušující správu uživatelů. Ústřední třídou bezpečnostního rámce prostředí ASP.NET 2.0 jsou třídy *System.Web.Security.Membership* a *System.Web.Security.MembershipUser*, jež vystavují mnoho statických metod pro vytvoření, smazání, aktualizaci a získání registrovaných uživatelů. Máme také předem připravené databázové tabulky a jejich relační spojení v databázovém diagramu. Tento model nám umožňuje už přidání prvků, jako jsou *Loginbox*, *ChangePassword* a *PasswordRecovery*. Jako již názvy těchto prvků napovídají, jedná se o dialogové okna sloužící k přihlášení uživatele, změně hesla a zaslání zapomenutého hesla.

Autentizační a autorizační systém by nebyl kompletní bez podpory rolí, pomocí nichž lze totiž seskupovat uživatele dohromady za účelem přidělení sady oprávnění nebo autorizací. Verze ASP.NET 2.0 má vestavěnou podporu rolí, kterou s ohledem na výkon a flexibilitu realizuje bezchybně. Správa rolí je ve výchozím nastavení vypnutá kvůli vyššímu výkonu pro weby, které nepotřebují role. Podpora rolí totiž vyžaduje provádění databázových dotazů a z toho plynoucí síťový provoz mezi databázovým a webovým serverem. Zapneme ji pomocí elementů `<rolemanager>` v souboru *web.config*. Verze ASP.NET 2.0 nabízí vestavěný mechanismus pro řízení uživatelských profilů, a to jednoduchým, přesto však kompletním a flexibilním způsobem díky němuž ušetříme hodiny práce. O vše se stará modul *Profile*, který nastavíme pomocí souboru *web.config*. K ještě většímu zjednodušení je nám poskytnuto rozhraní ASP.NET Configuration.

### 5.1.2 Stránka EditProfile.aspx

Tato stránka nabízí uživatelské rozhraní pro uživatele, jež chtějí změnit nebo doplnit svůj profil o některé informace. Zaujímá obsah prostředního sloupce a nabízí dialogy pro textový vstup včetně podmínek nutnosti vyplnění důležitých informací prostřednictvím indikace hvězdičkou jako je zvykem u mnoha webů.

### 5.1.3 Stránka Register.aspx

Tato stránka slouží pro registraci nového uživatele. Zabírá obsah prostředního sloupce a nabízí dialogy *uživatelské jméno*, *heslo*, *potvrzení hesla*, *email*, *bezpečnostní*

otázka, odpověď'. Ještě je nutno naprogramovat stránky: *AccessDenied.aspx* a *PasswordRecovery.aspx* sloužící k odepření přístupu a znovuzaslání zapomenutého hesla.

#### 5.1.4 Řídící prvek *UserProfile.ascx*

Uživatelské rozhraní a logika nutná k zobrazení a aktualizaci profilu uživatele se nachází v tomto uživatelském řídicím prvku umístěném v adresáři *Controls* kde se nacházejí všechny řídicí prvky webové aplikace.

## 5.2 Sekce články

Tato sekce je jedním z nejdůležitějších informačních zdrojů naší webové aplikace. Do této sekce bude přispívat správce, který má pravomoc k přidávání, úpravě a zveřejňování článků a dále schvalování reakcí, jež vkládají uživatelé pod článek. Článek lze do databáze kdykoliv přidat s možností zadat datum pro odložené zveřejnění. Dále musí mít osoba odesílající článek možnost stanovit datum vypršení platnosti, po kterém dojde k automatickému stažení článku. Pokud nespecifikují žádné z těchto dat, bude článek zveřejněn okamžitě na dobu neurčitou. Umožníme-li zároveň ostatním osobám jako jsou uživatelé webu tzv. *příspěvatelé* zasílat jejich vlastní zprávy a články budou tyto čekat na schválení správcem a následné zveřejnění. Správce má také ještě možnost určit, jestli daný článek bude viditelný pro všechny nebo jen pro registrované uživatele.

Články můžeme řadit do kategorií, jež jsou odděleny obrázkem a nadpisem. Bude také existovat možnost upřednostňovat jednotlivé články pro uživatele z podobné lokace například města, kraje. Uživatelé mohou zanechávat komentáře, které budou zobrazeny pod článkem, včetně jejich hodnocení. Nový obsah musí být dostupný jako RSS-zdroj, ke kterému se mohou uživatelé přihlásit a číst prostřednictvím své oblíbené RSS čtečky. Tento způsob je oblíbený uživateli, jež mají rádi aktuality ze svých oblíbených webů, ale nemají čas na weby chodit, přičemž RSS čtečka umožňuje stáhnout nový obsah z jejich oblíbených www stránek.

### 5.2.1 Uživatelského rozhraní

K vybudování uživatelského prostředí bude potřeba několika stránek, jež budou mít přístup k již navržené databázi. Jako první stránka zobrazující jednotlivé kategorie bude



stránka *ShowCategories.aspx*. Seznam kategorií implementuji jako řídicí prvek typu *DataList* s jedním sloupcem (atribut *repeatColumns* nastavený na hodnotu 1).

Další stránka se nazývá *BrowseArticles.aspx* a jedná se o zobrazení zveřejněných článků v určité kategorii, přičemž je nutno vybrat si článek, který se nám teprve celý zobrazí.

Pro zobrazení celého článku, který je uložen v databázi slouží stránka *ShowArticle.aspx*. Tento se nám celý zobrazí včetně možnosti hodnocení a příspěvku od uživatele.

K získání obsahu pomocí RSS čtečky potřebuji vybudovat uživatelský řídicí prvek *RssReader.ascx*. V ascx-souboru definuji seznam typu *DataList*, jenž zobrazuje název a popis svázaných RSS položek, přičemž název je zároveň odkazem na plné znění příslušného článku. Dále zde mám hlavičku, kterou nastavím na název kanálu, geografický odkaz ukazující na zdrojový RSS souboru a odkaz úplně dole, jenž ukazuje na stránku s dalšími články.

Schopnost nahrávat soubory na server pomocí jednoduchého uživatelského rozhraní nám umožňuje prvek *FileUploader.ascx*. Jakmile se soubor uloží, zobrazí řídicí prvek URL adresu, kterou může redaktor snadno zkopírovat do pole *ImageUrl* nějaké vlastnosti nebo se na soubor s obrázkem odkázat ve WYSIWYG-editoru článku.

### 5.2.2 Vrstva přístupu k datům

Jelikož jsem se rozhodl implementovat vrstevný návrh přístupu k datům musím vybudovat vrstvu přístupu k datům, která se nachází ve speciálním adresáři *App\_Code/DAL*. Zde vybuduji několik tříd, jež budou obsahovat kód v jazyce C# a budou mít příponu *.cs*.

Jednotlivé třídy: *ArticleDetails.cs*, *ArticlesProvider.cs*, *SqlArticlesProvider.cs*, *SiteProvider.cs*. Tyto třídy se starají o přístup k databázi pomocí vzoru *Provider*.

### 5.2.3 Vrstva aplikační logiky

Další třídy starající se o aplikační logiku se nacházejí v adresáři *App\_Code/BLL*. Aplikační třídy používají pro přístup k datům třídy datové vrstvy a ponejvíce se používají k uplatňování validačních pravidel, ke kontrole omezení a k poskytování objektově

orientované reprezentace dat a metod s nimi pracujících. Na tuto vrstvu se lze dívat jako na jisté mapování, díky němuž se kódu uživatelského rozhraní jeví relační databáze jako objekty. Relační databáze nejsou objektivě orientované a aplikační vrstva mi tak nabízí daleko užitečnější reprezentaci dat.

Jednotlivé třídy: *BaseArticle.cs*, *Article.cs*, *Category.cs*, *Comment.cs*, *CommentComparer.cs*.

#### 5.2.4 Uložené procedury

Pro práci s databází mám vytvořeny následující procedury: *tbh\_Articles\_InsertArticle*, *tbh\_Articles\_UpdateArticle*, *tbh\_Articles\_ApproveArticle*, *tbh\_Articles\_InsertVote*, *tbh\_Articles\_IncrementViewCount*, *tbh\_Articles\_DeleteArticle*, *tbh\_Articles\_GetArticleByID*, *tbh\_Articles\_GetArticles*, *tbh\_Articles\_GetArticleCount*, *tbh\_Articles\_GetPublishedArticlesByCategory*, *tbh\_Articles\_GetPublishedArticleCountByCategory*. Tyto procedury obsahují příkazy v SQL jazyce.

### 5.3 Sekce ankety

Kvůli vyšší interakci s uživatelem jsem se rozhodl použít anket, které mi umožní získat názory na jednotlivé položené otázky a témata. Motivací k hlasování pro uživatele je zobrazení výsledku ankety, přičemž každý uživatel má možnost hlasovat jen jednou. Pro zobrazení anket je určen levý obsahový sloupec, jenž umožňuje zobrazit více než jednu anketu. Správce webu má možnost zadávat jakékoliv zadání ankety včetně možných odpovědí a možnosti ankety aktualizovat a archivovat. Opakovanému hlasování jednoho uživatele je zamezeno pomocí IP klíče, který sleduje uživatelskou adresu IP a znemožňuje mu tak hlasovat dvakrát po sobě na stejnou anketu.

#### 5.3.1 Uživatelské rozhraní

Jednotlivé ankety jsou zobrazeny v levém obsahovém sloupci, který nabízí odkaz na stránku archivovaných anket, kde jsou zobrazeny výsledky všech uběhlých anket. Tato

stránka je *ArchivedPolls.aspx* a její obsah je programován do prostředního obsahového sloupce.

### 5.3.2 Vrstva přístupu k datům

V této vrstvě jsem navrhl tři třídy pro přístup do databáze. Jsou to třídy: *PollsProvider.cs*, *PollDetails.cs* a *PollOptionDetails.cs*.

### 5.3.3 Vrstva aplikační logiky

Jednotlivé třídy této vrstvy: *BasePoll.cs*, *Option.cs*, *Poll.cs*. Musím podotknout, že tyto třídy jsou uloženy v adresáři *App\_Code/BLL/Polls*.

### 5.3.4 Uložené procedury

Stránky pro zprávu anket budou využívat uložené procedury: *tbh\_Polls\_InsertPoll*, *tbh\_Polls\_UpdatePoll*, *tbh\_Polls\_DeletePoll*, *tbh\_Polls\_ArchivePoll*, *tbh\_Polls\_GetPolls*, *tbh\_Polls\_GetPollByID*, *tbh\_Polls\_GetCurrentPollID*, *tbh\_Polls\_InsertOption*, *tbh\_Polls\_UpdateOption*, *tbh\_Polls\_DeleteOption*, *tbh\_Polls\_GetOptions*, *tbh\_Polls\_GetOptionByID*, *tbh\_Polls\_InsertVote*. Většina z těchto uložených procedur je jednoduchá a nepotřebuje vysvětlení.

## 5.4 E-mailový zpravodaj

Pro ještě větší interakci s uživatelem jsem se rozhodl do mojí webové aplikace zahrnout oblíbený prvek, který usnadní přístup k informacím pro uživatele, jež chtějí přijímat informační emaily s novinkami. Je nutné web spravovat pravidelně a nabízet tak nové články uživatelům, abychom je motivovali k častějšímu návratu na naše stránky. Náš emailový zpravodaj ovšem nemá za úkol plnit schránky uživatelů nevyžádanou poštou, ale poskytnout jim tuto možnost pouze v případě jejich zájmu.

Pro práci s elektronickou poštou máme v ASP.NET 2.0 možnost využít řadu vestavěných funkcí jež nám usnadní práci. Jmenný prostor *System.Net.Mail* definovaný v prvku sestavení *System.dll* obsahuje všechny třídy pro posílání emailů. Ústředními

třídami jsou třídy *MailMessage*, jež představuje e-mailovou zprávu, a *SmtClient*, která nabízí metody, s jejichž pomocí lze získat objekty typu *MailMessage* přes spojení do nastaveného SMTP-serveru (SMTP znamená *Simple Mail Transfer Protocol*, tedy jednoduchý protokol pro přenos pošty, což je nízkoúrovňový protokol používaný serverem Microsoft Exchange a ostatními servery).

Třída *MailMessage* plně popisuje e-mailovou zprávu, tedy včetně jejího předmětu, těla, adresy To, CC (adresát kopie) a BCC (adresát skryté kopie) a jakékoliv přílohy.

#### 5.4.1 Uživatelské rozhraní

Prezentační vrstvu tvoří soubory s uživatelským rozhraním: *ShowNewsletter.aspx* jež zobrazí plné znění těla zprávy, jejíž ID se nachází v řetězci *querystring*, a to jak ve formě holého textu, tak i v HTML-formátu. Další stránkou je *ArchivedNewsletters.aspx*, jež použitím řídicího prvku typu *GridView* vypisuje archivované e-mailové zpravodaje.

Musím vyvinout taky uživatelsky řídicí prvek *NewsletterBox.ascx* v adresáři *Controls*. Tento se nejdříve podívá, je-li aktuální uživatel přihlášen. Pokud není, pak předpokládá, že není zaregistrován, a že tedy není ani odběratelem e-mailových zpravodajů, takže zobrazí zprávu vybízející jej k přihlášení mezi odběratele tím, že do textového pole zadá svou e-mailovou adresu. Jakmile uživatel klepne na tlačítko Odeslat, bude přesměrován na stránku *Register.aspx*.

#### 5.4.2 Vrstva přístupu k datům

I v tomto modulu je datová vrstva založena na návrhovém vzoru Provider, přičemž jako příklad obsahuje zprostředkovatele pro server SQL Server, který v podstatě jen zabaluje uložené procedury. Soubory této vrstvy: *NewsletterProvider.cs*, *NewsletterDetails.cs*.

#### 5.4.3 Vrstva aplikační logiky

Do adresáře *App\_Code/BLL/Newsletters* umístím třídy starající se o vrstvu aplikační logiky. Jsou to třídy: *Enums.cs* a *Newsletter.cs*.

#### 5.4.4 Uložené procedury

Jelikož v databázovém návrhu použijeme pouze jednu tabulku, uložené procedury jsou velmi jednoduché. Obsahují základní CRUD-operace (*Create*-vytvoř, *Retrieve*-získej, *Update*-aktualizuj a *Delete*-smaž). Přehled uložených procedur: *tbh\_Newsletters\_InsertNewsletter*, *tbh\_Newsletters\_UpdateNewsletter*, *tbh\_Newsletters\_DeleteNewsletter*, *tbh\_Newsletters\_GetNewsletters*, *tbh\_Newsletters\_GetNewslettersByID*.

### 5.5 Diskusní fórum

Pro komunikaci mezi uživateli implementují oblíbený prvek mnoha webů a to je diskusní fórum. Lidé si rádi vyměňují své názory a sdělují novinky. Nyní popíši, co by mělo moje diskusní fórum obsahovat a podporovat.

- Podpora pro více kategorií neboli podfór. Podfóra mají i jméno, popis a volitelně také obrázek.
- Možnost moderovat fóra, která je ovšem náročnější na správu ze strany správce. Moderované fórum je plně kontrolováno a příspěvky do něj musí být schváleny správcem.
- Zasílání příspěvků bude povoleno pouze registrovaným členům, kdežto prohlížení je povoleno každému.

- Možnost editovat své zprávy pomocí omezené, bezpečné sady HTML-značek. K tomu nám poslouží řídicí prvek typu FCKeditor s redukovanou nástrojovou lištou.
- Po vytvoření nového vlákna musí být uživatel schopen okamžitě vláknou uzavřít, aby již ostatní nemohli odpovídat. Jsou-li odpovědi povoleny mohou je později vypnout pouze správci, redaktoři a moderátoři.
- Uživatelé mohou kdykoliv upravovat své vlastní příspěvky.
- Uživatelé mohou mít ke svému účtu připojený obrázek, který se zobrazí u každého příspěvku, který odešlou.
- Příspěvky zasílané uživateli se sčítají a tento počet se zobrazuje společně s obrázkem uživatele u každé jeho zprávy.
- Sledování nových příspěvků je podporováno pomocí zdrojů v jazyku RSS 2.0
- Správci, redaktoři a moderátoři mohou upravit a smazat jakýkoliv příspěvek. K tomu mohou celé vlákno přesunout do jiného diskusního fóra.

### 5.5.1 Uživatelské rozhraní

K plné implementaci našeho modulu je nutno vytvořit stránky uživatelského rozhraní a uživatelské řídicí prvky. Jednotlivé soubory včetně popisu: *ShowForums.aspx* zobrazí seznam všech podfór, včetně názvu, popisu a obrázku. Navíc je implementován odkaz na RSS zdroj. *BrowseThreads.aspx* se stará o prohlížení vláken diskusního fóra po jednotlivých stránkách. *ShowThread.aspx* zobrazí všechny příspěvky daného vlákna v datové mřížce se stránkováním. *AddEditPost.aspx* podle parametrů na příkazovém řádku nabízí vytvoření nového vlákna, odeslání nové odpovědi nebo úpravu existující zprávy. *GetThreadsRss.aspx* vrací zdroj RSS s obsahem diskusního fóra. V adresáři *Controls*, kde standardně ukládám uživatelské řídicí prvky se nachází soubor *UserProfile.ascx*, který byl již vyvinut v části pro evidenci členů a profilů.

### 5.5.2 Vrstva přístupu k datům

Pro práci s daty jsem navrhl následující třídy datové vrstvy které jsou opět umístěny v adresáři *App\_Code/DAL*. Jsou to třídy: *Forumsprovider.cs*, *ForumDetails.cs*, *PostDetails.cs*, *SqlForumProvider.cs*, *DataAccess.cs*.

### 5.5.3 Vrstva aplikační logiky

Do adresáře *App\_Code/BLL/Forums* jsou vloženy následující třídy: *BaseForum.cs*, *Forum.cs*, *Post.cs*.

### 5.5.4 Uložené procedury

Nad mými dvěma databázovými tabulkami operují následující procedury, které slouží například pro ošetření schvalování příspěvků, přesouvání vláken či zvyšování počtu zhlédnutí. Jednotlivé procedury: *tbh\_Forums\_ApprovePost*, *tbh\_Forums\_CloseThread*, *tbh\_Forums\_DeleteForum*, *tbh\_Forums\_DeletePost*, *tbh\_Forums\_GetForumByID*, *tbh\_Forums\_GetForums*, *tbh\_Forums\_GetPostBody*, *tbh\_Forums\_GetPostID*, *tbh\_Forums\_GetPostCountByThread*, *tbh\_Forums\_GetThreadByID*, *tbh\_Forums\_GetThreadCount*, *tbh\_Forums\_GetThreadCountByForum*, *tbh\_Forums\_GetUnapprovedPosts*, *tbh\_Forums\_IncrementViewCount*, *tbh\_Forums\_InsertForum*, *tbh\_Forums\_InsertPost*, *tbh\_Forums\_MoveThread*, *tbh\_Forums\_UpdateForum*, *tbh\_Forums\_UpdatePost*

## 5.6 Internetový obchod

Abych umožnil správci webové aplikace nejen prezentovat svůj projekt, ale také mu poskytl jistou možnost výtěžku, implementuji internetový obchod. Bude se jednat pouze o menší variantu známých velkých on-line obchodů, která nabídne pouze omezené množství upomínkových a jiných předmětů. Následuje přehled funkcí, jež je zahrnut do mého modulu.

- Podpora více oddělení pro rozřídění produktů, snadné vyhledání produktu.
- Produkty je potřeba co nejlépe vizuálně znázornit pomocí fotografií.
- Podpora procentních slev, jež může nastavit obchodník.
- Zdroj RSS pro RSS čtečky a sledování novinek.

- Jednoduchý přehled toho jestli je daný produkt skladem.
- Snadné přidávání a odebrání produktů obchodníkem.
- Podpora stabilního nákupního košíku, jež zobrazí vložené zboží.
- Podpora různého druhu zasílání a plateb zboží.

V tomto modulu využiji nové řídicí prvky rámce ASP.NET 2.0, mezi něž patří *Wizard*, *MultiView*, *GridView* a *DetailsView*, z nichž poslední dva jsem již použil v předcházejících modulech. Rovněž zahrnu podporu internetového placení přes server PayPal, jež se ve světě usadil, jako nejoblíbenější server zprostředkující platby přes internet.

### 5.6.1 Uživatelské rozhraní

Následuje výpis stránek z mého návrhu: *ShowDepartments.aspx*, která pro koncového uživatele zobrazuje seznam obchodních oddělení s obrázkem, popisem a odkazem na seznam produktů u každého z nich.

*BrowseProducts.aspx*, jež vykresluje seznam produktů s podporou stránkování pro stanovené nebo pro všechna oddělení.

*ShowProduct.aspx* zobrazuje všechny údaje o konkrétním produktu, umožňuje zákazníkovi, aby jej ohodnotil a přidal do svého nákupního košíku pro pozdější posouzení či nákup.

*ShoppingCart.aspx* zobrazuje aktuální obsah nákupního košíku zákazníka, kterému dovoluje změnit počet kterékoliv položky, odebrat položku, zvolit přepravní metodu a přepočítat součet, náklady na dopravu a celkovou částku. Tato stránka také nabízí tří-krokového průvodce pro dokončení objednávky. Prvním krokem je výše popsany vlastní nákupní košík, ve druhém zákazník poskytne doručovací adresu (její výchozí hodnoty získáme s adresy uživatele uložené v jeho profilu, je-li k dispozici) a v posledním kroku může uživatel zkontrolovat veškeré údaje objednávky, tj. seznam objednávaných položek, mezisoučet, přepravní metodu a její cenu, celkovou částku a adresu, na kterou bude zboží expedováno. Po potvrzení posledního kroku se objednávka uloží do databáze a zákazník je odeslán na web systému PayPal, kde je vybídnut k jejímu zaplacení.



*OrderHistory.aspx* tato stránka umožňuje zákazníkům prohlížet jejich předchozí objednávky, kontrolovat jejich stav nebo kdy, pokud vůbec, byli expedováni. U objednávek, které byly v průběhu placení stornovány, se zobrazuje odkaz na stránku systému PayPal pro dokončení platby.

*GetProductsRss.aspx* generuje RSS zdroj pro katalog internetového obchodu a vrací několik produktů (jejich počet určuje konfigurační nastavení *RssItems*) seřazených podle parametrů v řetězci *querystring*. Může například vrátit deset nejnovějších výrobků, deset nejlevnějších výrobků nebo deset nejvíce zlevněných výrobků.

V návrhu jsou zahrnuty tyto uživatelské řídicí prvky: *ShoppingCartBox.ascx*, který staticky zobrazuje aktuální obsah nákupního košíku zákazníka se jmény a množstvím jednotlivých produktů. *AvailabilityDisplay.ascx*, zobrazuje dostupnost produktů ikonkami různých barev.

Pro systém PayPal jsem vyhradil nový adresář */PayPal*, kde umístím stránky: *OrderCompleted.aspx*. Jedná se o stránku, na kterou systém přesměruje zákazníka poté, co zaplatil objednávku. Stránka uživateli nabízí zpětnou vazbu, že objednávka byla uložena a že ji může zaplatit později. *Notify.aspx* kde služba Instant Payment Notification posílá výsledek transakce. Stránka ověří pravost oznámení a v případě kladné odpovědi označí objednávku jako ověřenou. *OrderCancelled.aspx* na tuto systém PayPal přesměruje zákazníka poté, co stornoval objednávku.

### 5.6.2 Vrstva přístupu k datům

Do adresáře *App\_Code/DAL* jsem přidal následující třídy: *DepartmentDetails.cs*, *ShippingMethodDetails.cs*, *ProductDetails.cs*, *OrderDetails.cs*, *OrderItemDetails.cs*, *OrderStatusDetails.cs*, *DataAccess.cs*, *StoreProvider.cs*, *SqlStoreProvider.cs*.

### 5.6.3 Vrstva aplikační logiky

V adresáři *App\_Code/BLL/Store* sem navrhl následující třídy: *BaseStore.cs*, *Department.cs*, *Product.cs*, *Enums.cs*, *Order.cs*, *OrderItem.cs*, *OrderStatus.cs*, *ShippingMethod.cs*, *ShoppingCart.cs*

#### 5.6.4 Uložené procedury

Nad databází opět vybuduji pestrou paletu uložených procedur:  
*tbh\_Store\_DecrementUnitsInStock*, *tbh\_Store\_DeleteDepartment*, *tbh\_Store\_DeleteOrder*,  
*tbh\_Store\_DeleteOrderStatus*, *tbh\_Store\_DeleteProduct*,  
*tbh\_Store\_DeleteShippingMethod*, *tbh\_Store\_GetDepartmentByID*,  
*tbh\_Store\_GetDepartments*, *tbh\_Store\_GetOrderByID*, *tbh\_Store\_GetOrderItems*,  
*tbh\_Store\_GetOrdersByCustomer*, *tbh\_Store\_GetOrdersByStatus*,  
*tbh\_Store\_GetOrderStatusByID*, *tbh\_Store\_GetOrderStatuses*,  
*tbh\_Store\_GetProductByID*, *tbh\_Store\_GetProductCount*,  
*tbh\_Store\_GetProductCountByDepartment*, *tbh\_Store\_GetProductDescription*,  
*tbh\_Store\_GetShippingMethodByID*, *tbh\_Store\_GetShippingMethods*,  
*tbh\_Store\_InsertDepartment*, *tbh\_Store\_InsertOrder*, *tbh\_Store\_InsertOrderItem*,  
*tbh\_Store\_InsertOrderStatus*, *tbh\_Store\_InsertProduct*, *tbh\_Store\_InsertShippingMethod*,  
*tbh\_Store\_InsertVote*, *tbh\_Store\_UpdateDepartment*, *tbh\_Store\_UpdateOrder*,  
*tbh\_Store\_UpdateOrderStatus*, *tbh\_Store\_UpdateProduct*,  
*tbh\_Store\_UpdateShippingMethod*.

## 6 SPRÁVA PORTÁLU

Pro správce mé webové aplikace je velmi důležité přívětivé prostředí pro údržbu celého portálu. K tomuto účelu jsem vyvinul celý nový odkaz v menu a to odkaz *SPRAVOVAT*. Tento odkaz je ovšem viditelný pouze uživateli, který má příslušnou roli a k tomu náleží pravomoc jeho správy webové aplikace.

Plnou kontrolu má nad portálem pouze role *Správce*. Další možnost úpravy mají uživatelé, jimiž je přidělena role *Redaktoři*, ti mohou spravovat vše kromě modulu obchod. *Příspěvatelé* mají nejomezenější přístup pro správu a to jen zasílání nových článků. Uživatelé přihlášení pod rolí *Obchodníci* mohou spravovat pouze modul internetového obchodu.

Pro oddělení přístupu obyčejných uživatelů a uživatelů, jež mají možnost spravovat webový portál je vytvořen adresář */Admin*, kde jsou umístěny stránky uživatelského rozhraní. Díky nastavení souboru *web.config* mají přístup do adresáře */Admin* pouze pověření uživatelé.

### 6.1 Správa uživatelů

Pro výpis uživatelů podle abecedního pořadí s možností vyhledávat podle počátečních písmen či pomocí textového políčka hledej, slouží stránka *ManageUsers.aspx*. Uvnitř stránky je použit ovládací prvek *GridView*, který umožňuje výpis podrobností o uživateli včetně podpory stránkování. Jako správce mám možnost smazat či upravit jednotlivé profily uživatelů.

Úvod Spravovat Uživatelé Správa uživatele

**Obecné údaje uživatele**

Uživatelské jméno: PetrPe  
E-mail: b@b.b  
Registrován: 7. dubna 2008 14:50  
Poslední přihlášení: 13. května 2008 10:46  
Poslední aktivita: 13. května 2008 10:47

Nyní přítomen:   
Povolen:   
Uzamčen:

**Uživatelské role**

Administrators  Obchodníci  Příspěvatelé  Redaktoři  Správci  
 Diskutující

Aktualizuj  
Vytvořit novou roli:  Vytvořit

**Profil uživatele**

**Předvolby**

Novinky: Nezasílat  
Jazyk: Čeština

**Údaje týkající se diskusního fóra**

URL-adresa Avatara:   
Podpis:

Obr. 8. Část formuláře údajů o uživateli

*EditUsers.aspx* je stránka pro kompletní úpravu uživatelského profilu. Mám zde výpis obecných údajů uživatele, uživatelských rolí, profilu uživatele, údajů týkajících se diskusního fóra, osobních údajů, adresy a dalších kontaktních údajů. Pomocí této stránky máme absolutní kontrolu nad registrovanými uživateli.

## 6.2 Správa článků

K jednoduché editaci modulu články slouží stránka *ManageArticles.aspx*. Zde se nám nabízejí odkazy *spravovat kategorie*, *spravovat komentáře* a *přidat nový článek*. Ve spodní části je náhled na jednotlivé články s možností mazání a úpravy článku.

*ManageCategories.aspx* je stránka k úpravě kategorií jednotlivých článků. Mám zde možnost kompletně změnit název, obrázek, popis reprezentující kategorii. Rovněž je možnost zobrazení zařazených článků, nebo úplného smazání kategorie.

*ManageComments.aspx* nabízí správu komentářů k jednotlivým článkům a jejich schvalování.

*AddEditArticle.aspx* je stránka pro jednoduché vytvoření článku za pomoci WYSIWYG editoru. K článku je možno nastavit datum expirace, datum vydání článku, viditelnost článku pouze registrovaným uživatelům, možnosti komentářů a dalších volitelných nastavení.

### 6.3 Správa diskusních fór

Rovněž tento modul je nutno nějakým způsobem editovat. K tomu nám slouží stránka *ManageForums.aspx*, kde máme výpis kategorií fór pomocí ovládacího prvku *GridView*. Tyto kategorie můžeme upravovat, mazat a určovat jestli bude fórum moderované, což znamená potřebu schvalování jednotlivých příspěvků fóra správcem. K tomu se nám nabízí odkaz na stránku *ManageUnapprovedPosts.aspx* kde máme možnost zaslané příspěvky schválit či smazat.

### 6.4 Správa anket

Pro modul anket je vytvořená stránka *ManagePolls.aspx*, který nabízí výpis jednotlivých anket dle ID s možností úpravy, mazání a archivování anket. K určení, která anketa se nám bude právě zobrazovat slouží zaškrtačací políčko s nadpisem *aktuální*.

*ArchivedPolls.aspx* je stránka pro zobrazení již proběhlých archivovaných anket a jejich výsledků.

### 6.5 Správa novinek

Jedná se o velmi jednoduchý modul obsahující stránku *SendNewsletter.aspx* jež obsahuje textová políčka, kde vepíšeme text do políček *předmět*, *tělo s holým textem*, text pomocí WYSIWYG editoru. Kliknutím na tlačítko *odeslat* budeme přesměrováni na stránku *SendingNewsletter.aspx*. Na této stránce se nachází procentní ukazatel stavu odesílání našich e-mailů, který je realizován pomocí Java skriptu. Rozeslané e-maily obdrží pouze uživatelé, jež požádali o tuto službu.

## 6.6 Správa obchodu

Do tohoto rozhraní patří správa katalogů zboží a správa objednávek. Odkazem *správa katalogu zboží* se dostaneme na stránku *ManageProducts.aspx*. Zde máme výpis a náhled na produkty s možností filtru oddělení a rovněž se zde nachází odkazy: *Spravovat oddělení*, *Spravovat způsoby přepravy*, *Spravovat stav objednávky* a *Přidat nový produkt*.

*ManageDepartments.aspx* je stránka jež se stará o kompletní nastavení oddělení obchodu, jako je editace, mazání atd. Navíc také nabízí dialog pro stažení souboru na server. Tuto funkci jsem již popsal výše.

*ManageShippingMethods.aspx*, spravuje možnosti, jakým způsobem bude zasláno zboží směrem k příjemci.

*ManageOrderStatuses.aspx*, se stará o stav objednávky. Opět implementují vlastnosti editace a mazání.

*AddEditProduct.aspx* je stránka velmi podobná stránce pro přidání nového článku a obsahuje stejné podobné uživatelské rozhraní, včetně možnosti stáhnutí souboru na server.

*ManageOrders.aspx*, umožňuje vyhledat jednotlivé objednávky podle ID, nebo jména zákazníka a spravovat jejich stav.

## 6.7 Schvalování příspěvků

Tato sekce je obecně sekci, se kterou bude správce schvalovat reakce uživatelů na obsah webu. Mělo by jít čistě jen o korekci či mazání neslušných reakcí, jež nerespektují pravidla webového portálu, stránka starající se o tuto správu se nazývá *ManageUnapprovedPosts.aspx*.

## ZÁVĚR

Cílem vytvoření webové aplikace pro hudebního interpreta bylo co nejlépe jej reprezentovat v celosvětové internetové síti a to hlavně na území České republiky. Důležité kritérium bylo ovšem také vytvořit si jistý kruh lidí „fanoušku“, jež se budou rádi na náš web vracet. Proto jsem jim nabídnul velkou možnost a svobodu vlastního vyjádření se k obsahu a interakce správce s uživatelem.

Nejdříve jsem navrhl přibližnou grafickou podobu celého webu, jež se dále díky kaskádovým stylům a vzorové stránce stala vlastní pro všechny stránky portálu. Dále jsem již jen využil všech dostupných programových prostředků ASP.NET 2.0, abych splnil funkčnost podle návrhu zadavatele (hudební skupiny). Rovněž jsem musel použít dostupnou databázi, vzhledem k integraci do mého vývojového prostředí MS Visual Studio 2008 Express jsem použil SQL Server 2005 společnosti Microsoft.

Úkolem bylo vytvoření co nejsnadnějšího rozhraní pro správce webu. Celá správa včetně nahrávání souboru na server nebo plné editace textu pomocí WYSIWYG editoru je implementována přímo ve webové aplikaci a to v odkazu „SPRAVOVAT“ v menu. Tento odkaz je však viditelný pouze registrovanému správci a dalším registrovaným uživatelům, jež mají od správce přiděleny jednotlivé role.

Samotnou webovou aplikaci je nutno nasadit na nějaký dostupný hostingový server, jež nabídne slušné a dostatečné podmínky hostingu. K mému účelu jsem zaregistroval doménu třetího řádu jako <http://dickinson.aspone.cz> společnosti Aspone. Webové stránky bylo také nutné otestovat na více prohlížečích. Testoval jsem na IE 6.0, NetCaptor 7.5.4, Opera 9.26. Ve všech těchto prohlížečích se nevyskytly chyby v zobrazení a funkčnosti webu.

Osobně doufám, že má práce bude dobře reprezentovat cílenou osobu a nabídne nadstandardní interakci s uživatelem. Toho všeho je dosaženo za pomoci přátelského prostředí, ve kterém je radost spravovat a udržovat portál v chodu.

## ZÁVĚR V ANGLIČTINĚ

The main purpose of creating this web application was great representation of musical band or vocalist on internet mainly in Czech Republic. Rich interaction between user and administrator is also very important in this case because this web application should help to bunch up users and „fans“. Make all this people happy to visit our web site regularly means to give them rich possibility of expression themselves.

At first I tried to manage a final appearance of my application. This was very important to me. All the features like CSS and Master Page helped me to realize my intention. After that I used all accessible features of ASP.NET 2.0 to make my web site fully dynamic as my submitter wished. My application is using database and because of great integrity of MS Visual Studio 2008 Express and SQL Server 2005 I choosed SQL Server 2005 by Microsoft.

Make a user friendly interface for administrator was also very important. I decided to integrate WYSIWYG editor and really easy way of uploading files to server straight from web site. Then a link „SPRAVOVAT“ is accessible straight from main menu but only for registered users with important rights. To realize this feature I used ASP.NET 2.0 manage roles configuration setup.

Of course I needed to place my web site to any hosting server that is acceptable. I decided to make account and register my web site as <http://dickinson.aspone.cz>. Before that I tested my application in different browsers as IE 6.0, NetCaptor 7.5.4, Opera 9.26. There wasn't any problems with that.

Personally I hope that my web site will make great representation and will offer good interaction within user and administrator. The user friendly interface should make pleasure with browsing and editing portal.



**SEZNAM POUŽITÉ LITERATURY**

- [1] Creating web sites the missing manual /. -1<sup>st</sup> ed.. Beijing ; Sebastopol, Calif. : Pogue Press/Oreilly, 2006. – ix, 586 s. :il. ;24 cm.
- [2] 333 tipů a triků pro maximální využití Internetu /. – Vyd. 1.- Brno : Computer Press, 2007. – 215 s. : il. ; 23 cm.
- [3] Programming ASP.NET AJAX. -1st ed.. – Beijing ; Cambridge : Oreilly, c2007. – xviii, 454 s. : il.
- [4] ASP.NET 2.0 cookbook. – 2nd ed.. – Sebastopol, Calif. :Oreilly, c2006. – 1014 s. :il.
- [5] ASP.NET 2.0 a C-Sharp : tvorba dynamických stránek profesionálně /. – Vyd. 1.. – Brno Zoner Press, 2006. 1376.
- [6] C-Sharp : začínáme programovat ; Addison-Wesley ; Frank Eller, Grada Publishing, 2002. – 240 s. :il.
- [7] Webové programování v ASP.NET 2.0 ; - Bellinaso; Wrox – Vyd. 2.0.- Brno : Computer Press, 2007, 648 s.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

ASCII	American Standart Code for Information Interchange
ASP	Active Server Pages
BCL	Base class library
CLR	Common Language Runtime
CSS	Cascading Style Sheets
FTP	File Transfer Protocol
HTML	HyperText Markup Language.
JSP	Java Server Pages
SQL	Structured Query Language
URL	Uniform Resource Locator
UTF-8	UCS Transformation Format
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
W3C	World Wide Web Consortium.
WYSIWIG	What You See Is What You Get
WWW	World Wide Web

**SEZNAM OBRÁZKŮ**

Obr. 1 – Návrh rozložení jednotlivých prvků na stránce.....	25
Obr. 2 – Návrh přepínače motivů .....	28
Obr. 3 – Schéma relačního spojení tabulek článků.....	34
Obr. 4 – Schéma relačního spojení tabulek fóra.....	34
Obr. 5 – Schéma relačního spojení tabulek anket.....	35
Obr. 6 – Schéma relačního spojení tabulek obchodu .....	35
Obr. 7 – Schéma vrstvého návrhového vzoru.....	38
Obr. 8 – Část formuláře údajů o uživateli.....	52

## SEZNAM PŘÍLOH

- P I    Návod ke správě portálu
- P II    CD se zdrojovými kódy, obrázky, dokumentace v elektronické podobě

# PŘÍLOHA P I: SPRÁVA PORTÁLU

## 1. Přihlášení

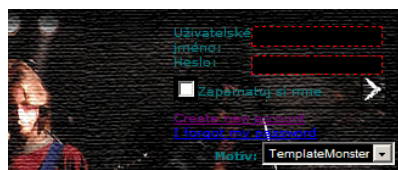
Pro snadnou správu určitých částí webu jsou učeny role. Pro administrátora je určena role s absolutní možností editace všech „oddělení“. Budu se zabývat tedy jen tímto typem správce.

Dokud je správce nepřihlášen není pro něj viditelný odkaz *SPRAVOVAT* v menu. Proto provedeme přihlášení pomocí rámečku v horním pravém rohu.



*Zobrazení menu před přihlášením.*

Do přihlašovacího rámečku přidáme plané jméno a heslo uživatele, které je správci přiděleno při přebírání projektu.



*Přihlašovací rámeček*

Následně se nám zobrazí navíc odkaz v pravém rohu menu.



*Zobrazení menu pro přihlášení.*

## 2. Správa uživatelů

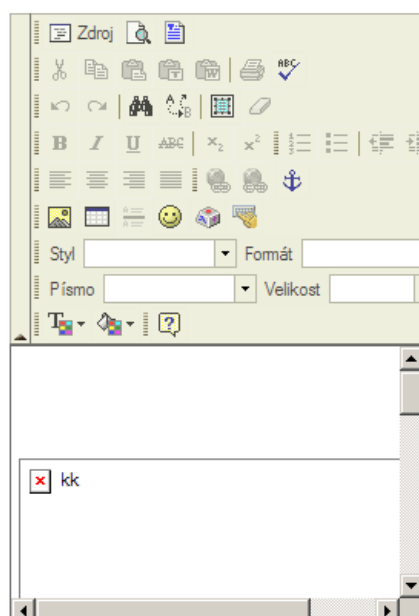
V této sekci správce zjistí počet uživatelů, jež jsou zaregistrováni, včetně jejich jmen možností vymazat, upravit jejich profil. Také se zde dozvíme kolik uživatelů je právě přihlášených.

Uživatelské jméno	E-mail	Vytvořen	Poslední aktivita	Povoleno
Admin	admin@contoso.com	23.01.2006 22:27	09.05.2008 12:52	<input checked="" type="checkbox"/>
Anne	anne@wrox.com	09.11.2005 11:10	27.12.2005 9:00	<input checked="" type="checkbox"/>
Homer	homer@wrox.com	23.01.2006 22:30	21.04.2008 13:12	<input checked="" type="checkbox"/>

*Tabulka zobrazující informace o zaregistrovaných uživateli*

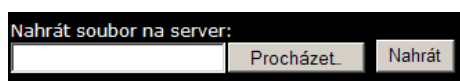
### 3. Správa článků

Po výběru kategorie a názvu článku máme možnost celkové editace a smazání článku. Pro pohodlnou editaci textu slouží integrovaný WYSIWYG FCKeditor. Pomocí něj jednoduše a přirozeně píšeme texty s možností formátování či přidávání obrázků.



*Integrovaný WYSIWYG editor textu*

Často je také nutno uložit nějaký nový obrázek či fotografii na server aby se mohla stát součástí článku. K tomuto účelu najdeme ve spodní části stránky textové okénko se dvěma tlačítky, pomocí nichž intuitivně vyhledáme a nahrajeme soubor na server.



*Textové okénko pro vložení souboru na server.*

O tom jestli soubor úspěšně nebo neúspěšně dorazil na server obdržíme hlášení pod dialogovým okénkem. Zároveň se nám zde zobrazí cesta, kde se nyní soubor v adresářové struktuře nachází.



*Textové okénko se zobrazením cesty*

#### 4. Správa diskusních fór

Mezi další sekce správy patří diskusní fóra, kde editujeme názvy témat k diskuzi a další možnosti. Pokud zatrhneme možnost, že fórum má být moderované budeme muset všechny příspěvky od uživatelů schválit a až poté budou zveřejněny.



*Dialog pro správu diskusního fóra*

#### 5. Správa anket

Co se týče anket je na nás volba jak se budou nazývat, jaké budou možnosti odpovědi a či bude anketa aktuální. Samozřejmostí je archivace výsledků, jež budou čtenářům zveřejněny.

ID	Anketa	Hlasů	Aktuální
29	Jak se Vám líbil náš koncert v Brně?	4	<input checked="" type="checkbox"/>

Anketa			
ID	29		
Přidáno	8.4.2008 14:31:37		
Přidal(a)	PetrPe		
Hlasů	4		
Otázka	Jak se Vám líbil náš koncert v Brně?		
Aktuální	<input checked="" type="checkbox"/>		

Update Cancel

Odpověď	Hlasů	%
nic moc	2	50,0
ušel	1	25,0
byl dobrý	0	0,0
byl výborný	1	25,0
perfektní	0	0,0

Podrobnosti odpovědí

Odpověď

Insert Cancel

*Formulář možných editací ankety*

## 6. Správa e-mailového zpravodaje

Práce s rozesíláním emailů je zcela intuitivní a není potřeba jí podrobně popisovat. Snad jen dodám, že námi vytvořená zpráva obsahuje předmět, tělo s holým textem a je rozeslána všem uživatelům jež si ve svém profilu zvolili tuto možnost příjmu novinek na svou e-mailovou schránku.

## 7. Správa obchodu

Jedná se o rozvinutější modul, který je ovšem velmi jednoduché ovládat. Mezi hlavní sekce patří správa katalogu zboží a správa objednávek. Katalog se zbožím je možné plně editovat a to včetně zobrazení ceny, dostupnosti zboží a malé i velké fotografie. Objednávky lze vybírat podle mnoha kritérií.



*Zobrazení jedné položky zboží*

## 8. Správa neschválených příspěvků

Pokud uživatelé přispějí do sekce, kde je potřeba nejprve zkontrolovat obsah příspěvku je nutné jej nejdříve schválit a až poté jej zveřejnit. Tato operace je opět velmi intuitivní a pro správce bude přirozená.



