VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

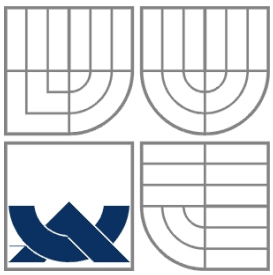
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

REDAKČNÍ SYSTÉM V .NET

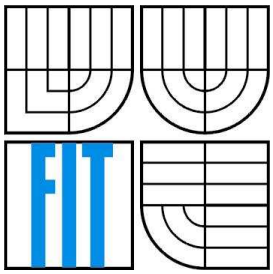
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE VLASTIMIL MAREŠ
AUTHOR

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

REDAKČNÍ SYSTÉM V .NET

EDITORIAL SYSTEM IN .NET

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

VLASTIMIL MAREŠ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. ZBYNĚK KŘIVKA

BRNO 2007

Zadání diplomové práce/4791/2006/xmares05

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2006/2007

Zadání diplomové práce

Řešitel: **Mareš Vlastimil**
Obor: Výpočetní technika a informatika
Téma: **Redakční systém v .NET**
Kategorie: Web

Pokyny:

1. Seznamte se s prostředím .NET, ASP.NET a s open-source aplikací DotNetNuke (DNN).
2. Na základě prostudování portálového systému DNN navrhněte jeho rozšíření, aby byl efektivně použitelný pro vytváření redakčních systémů pro webové prezentace malých a středních firem.
3. Implementujte několik (nejméně 4) podpůrných modulů do DNN (např. modul Rubrika, Článek, Aktualita, Podpora vícejazyčnosti obsahu). S využitím DNN a nových modulů realizujte prototyp redakčního systému.
4. Na jednoduché firemní prezentaci postavené na vaší internetové aplikaci demonstrujte vlastnosti vašeho systému.
5. Zhodnoťte další možnosti vašeho systému a nastiňte možnosti budoucího rozvoje.

Literatura:

- Archer, T.: Myslíme v jazyce C#, Grada, Praha, 2002.
- Kačmář, D.: Programujeme .NET aplikace ve Visual Studiu .NET, Computer Press, Praha, 2001.
- Prosise, J.: Programování v Microsoft .NET: webové aplikace v .NET Framework, C# a ASP.NET, Computer Press, Brno, 2003.
- Perpetual Motion Interactive Systems Inc.: DotNetNuke - Web Application Framework. Dokument dostupný na URL http://www.dotnetnuke.com (říjen 2005) (anglicky)

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Křivka Zbyněk, Ing.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 22. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

Abstrakt

Tato diplomová práce se zabývá použitím systému pro správu obsahu DotNetNuke jako redakčního systému. Jsou zde popsány všechny vlastnosti a možnosti systému DotNetNuke. Na základě prostudování těchto možností jsou navržena rozšíření, pomocí kterých půjde DotNetNuke používat jako plnohodnotný redakční systém.

Výsledkem této diplomové práce jsou čtyři rozšíření systému DotNetNuke, vytvořená podle předchozích návrhů.

Klíčová slova

ASP.NET, C#, Redakční systém, Systém pro správu obsahu, Informační systém, DotNetNuke, Open source, Modul

Abstract

This master’s thesis with the usage of the system for the content managing - the DotNetNuke system as an editorial system. It describes all options and features of the DotNetNuke system. On its basis of the system's features there have been made proposals for extensions which, when implemented, will let the DotNetNuke system to be used as full-value editorial system.

Keywords

ASP.NET, C#, Editorial system, Content Management System, Information system, DotNetNuke, Open source , Module

Citace

Vlastimil Mareš: Redakční systém v .NET, diplomová práce, Brno, FIT VUT v Brně, 2007

Redakční systém v .NET

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením ing. Zbyňka Křivky
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jméno Příjmení
Datum

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu projektu Ing. Zbyňku Křivkovi za připomínky a rady, které mi pomohly projekt vypracovat.

Obsah

1 Úvod

Tento diplomový projekt je zaměřen na tvorbu redakčního systému v .NET. Respektive na upravení systému pro správu obsahu DotNetNuke (dále jako DNN) na redakční systém. Tuto úpravu jsem provedl navrhnutím a naprogramováním čtyř rozšíření (dále jako moduly). Tyto moduly zajišťují, že redakční systém bude použitelný pro prezentace malých a středních firem. Tento systém jsem nemusel vytvářet sám, ale pouze šlo o vylepšení již existujícího. To znamená, že jsem si vyzkoušel práci, při které jsem musel pracovat s již existujícím zdrojovým kódem vytvořeným jinou osobou než mnou a s kterou se bude setkávat v praxi častěji než s tvorbou kompletně celé aplikace.

Projekt je vytvořen v prostředí ASP.NET jazykem C#. Při tvorbě jsem využíval vývojové prostředí Microsoft Visual Studio 2005. To byly další důvody, proč jsem si tento projekt zvolil. Chtěl jsem si vyzkoušet po práci s PHP (vytvořil jsem v něm svůj ročníkový projekt) i práci s ASP.NET.

V projektu jsem také využil svých znalostí z práce na ročníkovém projektu. Jde hlavně o znalost HTML a SQL. Dále jsem také využil znalosti s tvorbou informačního systému (v ročníkovém projektu šlo o informační systém fotbalového klubu) a prezentací cizího subjektu.

1.1 Popis jednotlivých kapitol

Základní pojmy – popis programovacích jazyků, technologií a dalších důležitých pojmů použitých v tomto diplomovém projektu.

Popis systému DNN – vysvětlení co je to DNN, jak funguje, můj názor na tento systém a zhodnocení výhod nevýhod DNN.

Návrh a popis modulů – popis jednotlivých modulů, jejich vlastností a funkcí a také zdůvodnění, proč jsem navrhla naprogramoval právě tyto moduly.

Popis implementace – popis jak jsem tvořil jednotlivé moduly, problémy které jsem během implementace musel řešit a zdůvodnění řešení, které jsem při tvorbě modulů použil.

ER diagramy – názorné ukázání propojení mnou vytvořených modulů a systému DNN.

Závěr – zhodnocení výsledků, kterých jsem dosáhl při tvorbě tohoto diplomového projektu a nastínění možností jak by se dalo v budoucnosti v tomto projektu pokračovat.

2 Základní pojmy

2.1 Systém pro správu obsahu

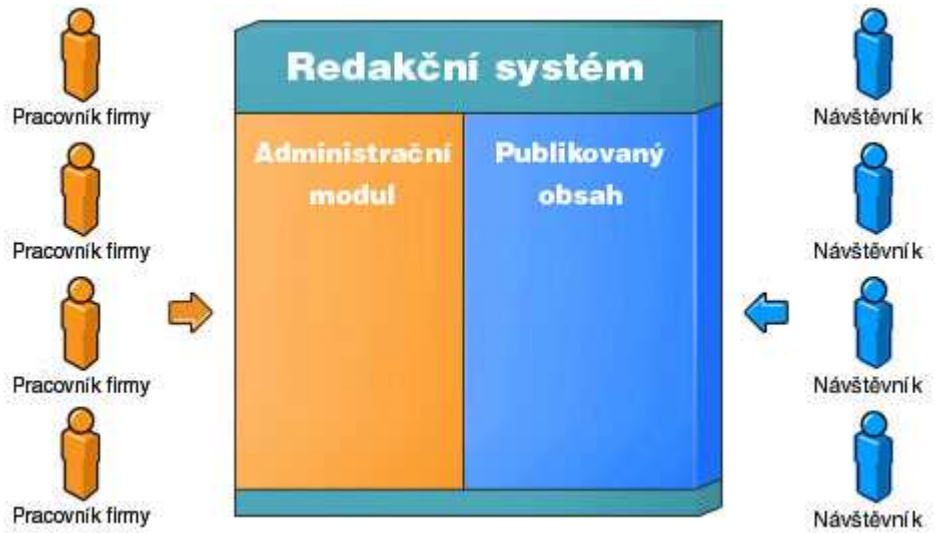
Systém pro správu obsahu je takový systém, který dovolí spravovat libovolné dokumenty v systému i lidem, kteří neumí programovat ani navrhovat systémy. U systému pro správu obsahu lze vidět velkou podobnost s redakčními systémy. Ty jsou ale založeny hlavně na publikování obsahu, takže se hodí skvěle například pro internetové prezentace apod. Systém pro správu obsahu je ale potřeba chápat poněkud obecněji. Zde tedy nejde jen o co nejsrozumitelnější zobrazování dokumentů pro návštěvníky webových stránek, ale celkově o správu dokumentů.

2.2 Redakční sytém

Redakční systém nebo též publikační systém umožňuje měnit a následně publikovat informace na www stránkách (ale třeba i v interní aplikaci na intranetu) pracovníkům firmy, kteří nemají znalosti web designérů nebo web programátorů.

Editace obsahu v redakčním systému se svým komfortem blíží editaci dokumentů pomocí klasických editorů například MS Word. Umožňuje volbu stylu a barvy písma, vkládání obrázků, tabulek, souborů ke stažení atd.

Redakční systém se skládá z administračního modulu a z modulu určeného pro konečné zobrazení samotných www stránek. Administrační obsahuje komfortní uživatelské rozhraní umožňující uživatelům (pověřeným pracovníkům firmy) provádět změny obsahu.



Obrázek 2-1: struktura redakčního systému *zdroj [6]*

Za základní vlastnosti publikačního systému lze považovat: editace obsahu, publikace obsahu, přístupová práva k položkám systému, jazykové verze, statistiky přístupů, možnost SEO optimalizace.

Za vlastnosti publikačního systémů lze považovat:

- Editace a vystavení obsahu dle přístupových práv Pracovníci společnosti se v dané firmě na různých pozicích. Manažér, asistent manažéra, pracovník marketingového oddělení. Každá s těchto osob může mít v systému různá práva. Jde především o právo na změnu určitého textu a právo na jeho publikaci. Před publikací je nová verze měněného textu přístupná pouze v administračním modulu zatímco na webu je stále poslední publikovaná verze. Publikací se změněný text přenese na www stránky. Redakční systému musí umožňovat přidělit práva na editaci a publikaci různým osobám.
- Editace různých jazykových verzí Tato funkce je dnes neodmyslitelnou součástí v dnešní době má většina firem vícejazyčné stránky. Systém umožňuje editaci více jazykových verzí v rámci jednoho administračního rozhraní. Přístupná však mohou být na různých doménách
- SEO optimalizace Redakční systém musí být naprogramován tak aby umožňoval SEO optimalizaci v opačném případě není jeho obsah správně pochopen vyhledávači a www stránky pak nejsou vyhledatelné pomocí standardních vyhledávačů www.seznam.cz, www.google.com atd.
- Sledování přístupů na www stránky Systém musí umožňovat sledovat četnost přístupů na stránky a zjišťovat další potřebné informace. Především jakým způsobem návštěvníci dané stránky nalezli a o jaké informace měli zájem.
- Existence dalších podpůrných modulů Redakčního systému obvykle obsahují další moduly jako jsou galerie obrázků, ankety a další.
- Více viz. [Reda2006]

2.3 SEO

SEO je anglická zkratka pro Search Engine Optimization, což česky znamená optimalizace pro vyhledávače. Jde o metodologii vytváření a upravování internetových stránek takovým způsobem, aby byly po zadání dotazu v internetovém vyhledávači zobrazeny mezi prvními výsledku tohoto dotazu. Cílem je, aby se na stránky podívalo co nejvíce návštěvníků.

Pro běžné dotazy se v databázi internetového vyhledávače najdou tisíce stránek, které odpovídají zadanému dotazu. Vyhledávač tyto stránky poté seřadí do výsledků podle toho, jak je která stránka kvalitní a relevantní k dotazu (seřazeny budou od nejkvalitnějších a nejdůležitějších).

Samotné algoritmy internetových vyhledávačů, které hodnotí stránky, nejsou známy. Ale je zřejmé, že nejlépe budou hodnoceny ty stránky, které budou podle algoritmu internetového vyhledávače nejlépe splňovat očekávání od vyhledávajícího uživatele. Na základě obecně známých informací o těchto algoritmech lze odvodit, že pozici stránky ve výsledcích vyhledávání může napomoci, pokud na stránku odkazuje hodně dalších stránek, pokud se vyhledávaný výraz na stránce vyskytuje vícekrát, na důležitých místech (např. v titulku) apod.

Cílem SEO je navrhnout nebo upravit internetovou stránku tak, aby byla pro relevantní dotazy hodnocena co nejkvalitněji a aby byla vždy zobrazována na nejlepších místech ve výsledcích relevantního dotazu. Je to z toho důvodu, že většina uživatelů při hledání věnuje pozornost pouze několika prvním odkazům. K tomu abychom vytvořili takovou stránku existuje velmi mnoho technik. Některé z těchto technik jsou považovány za správné a etické a doporučují se používat, jiné z těchto technik za nesprávné a neetické. Techniky by se daly rozdělit na dvě hlavní části podle toho jaký faktor hodnocení stránek využívají. Jsou to Off-page faktory a On-page faktory.

Off-page faktory nemůžeme přímo ovlivnit strukturou a vlastnostmi stránky. Jde hlavně o zpětné odkazy (stránky, které odkazují na námi vytvořenou stránku). Takže jde o to, aby na naši stránku odkazovalo co nejvíce jiných stránek. U nekomerčních projektů je to snazší. Když má stránka kvalitní obsah, tak na ni bude odkazovat hodně dalších nekomerčních stránek. U komerčních projektů je důležitá důkladná registrace do většiny katalogů (nelépe ruční). Dále je u nich dobré, aby se na stránce nacházel také nekomerční obsah, na který budou další stránky odkazovat. Zde uvedu několik způsobů jak ovlivňovat Off-page faktory:

Link trading je placená služba. Některé firmy nabízí, že za poplatek budou odkazovat na naši stránku. Firmy provozující internetové vyhledávače nemají tento přístup rády a stránky, které se snaží zviditelnit tímto přístupem, penalizují. Takže výsledný efekt může být v konečném důsledku přesně opačný než byl původní záměr. Jde o jednu z neetických metod. Další oblíbenou neetickou metodou je spam. Odkazy na stránku se pak umísťují do různých diskusí, návštěvních knih apod.

Text v odkazu je další důležitý faktor, protože text mezi <a> a <\> většinou udává co se na dané stránce bude vyskytovat. Proto je důležité před registrací stránky do katalogů vymyslet takový titulek stránky, který bude obsahovat klíčová slova, pro než chcete stránku optimalizovat, protože právě titulek se většinou používá jako odkaz na stránku. Je také důležité používat v odkazech smysluplný text. To znamená vyvarovat se odkazů typu <a>více zde<\a> a podobně. Vždy je lepší rozepsat, co se pod daným odkazem skrývá.

Zjišťování zpětných odkazů je technika, kterou podporuje většina internetových vyhledávačů. Po vyhledání adresy stránky ve vyhledávači a kliknutí na možnost vypsat zpětné odkazy, se vypíší všechny stránky, které na danou stránku ukazují. Jedná se o velmi důležitou techniku.

On-page faktory jsou všechny ty, které můžeme kódem a obsahem stránky ovlivnit. Nejdůležitější je psát dobře přístupný kód. Tím myslím kód, který zajistí, že po vypnutí grafiky, obrázků a CSS souborů, na stránce nebude kromě již zmíněné grafiky nic chybět. Je to důležité z toho

důvodu, že stejně stránku vidí i vyhledávací roboti internetových vyhledávačů, takže pokud na stránce bude něco chybět, tak to neuvidí ani vyhledávací robot.

Struktura webu je nejdůležitější částí prezentace. Problém může nastat už u adresy úvodní stránky. Vyhledávače vidí *www.nazev_stranky.cz, http://www.nazev_stranky.cz, www.nazev_stranky.cz/index..htm, http://www.nazev_stranky.cz* jsou pro internetové vyhledávače čtyři různé stránky. Proto je důležité používat vždy důsledně jen jednu verzi adresy a to jak ve všech různých katalozích, tak při odkazování na úvodní stránku z ostatních stránek webu. S tím souvisí i pravidlo, že by se nikdy neměl na více doménách provozovat stejný obsah. Dále je důležité provozovat pouze jednu doménu se stejným obsahem. Pokud máme více domén se stejným obsahem, tak si internetový vyhledávač vybere jen jednu podle něho nejdůležitější doménu a ostatní ignoruje. Další věcí, která hlavně u rozsáhlých webů může pomoci je mapa webu. U statických webů, kde každá stránka je vlastní HTML dokument, obvykle nenastávají výrazné problémy. Ty mohou nastat pokud máme dynamický web. Pokud se stránky od sebe liší pouze koncovkou (například .aspx), tak k žádným problém docházet nebude. Ale pokud se stránky generují podle obsahu databáze a liší se i parametry za otazníkem, tak nastává problém.

Ideálním nástrojem, který takové problémy řeší, je **mod_rewrite**. Pak bude URL vypadat takto: *http://www.nazev_stranky.cz/parametr1/parametr2/stranka.htm,* ale díky mod_rewrite bude tato stránka ukazovat například na:
http://www.nazev_stranky.cz/index.aspx?dat=parametr1&kat=parametr2&page=stranka
V současnosti je toto považováno za jedno z nejlepších řešení.

Robots.txt je textový soubor, ve kterém jsou uloženy pokyny pro vyhledávacího robota, jak procházet web.

Také je vhodné, mít správně titulky stránek (aby podle titulku bylo poznat, co se na stránce vyskytuje). Také u nadpisů používat tagy pro nadpisy (h1, h2….), pokud pouze ručně zvětšíme font, tak vyhledávač nepozná, že jde o nadpis. Dále je výhodné mít správně vyplněna meta elementy (Language, Descriptions, Robots, Key words).

Mezi neetické metody například patří používání skrytého textu, který běžný návštěvník stránek nevidí, ale vyhledávací robot ano nebo zobrazování jiného obsahu pro vyhledávacího robota a jiného pro návštěvníka stránek. Více viz [1] a [8].

2.4 ASP.NET

Před příchodem ASP.NET 1.0 spoléhaly starší technologie, určené pro webové aplikace založené na serveru, na skriptovací jazyky nebo na proprietární konvence značkování. Ty neposkytovaly moderní a integrovaný pracovní rámec (framework) pro webové programování.

Obecně se dá říct, že většina pracovních rámců pro webový vývoj, které byly vytvořeny před ASP.NET, se dají rozdělit do těchto dvou kategorií:

- Skripty, které interpretuje nějaký zdroj na serveru.
- Oddělené, malé aplikace, které se vykonávají voláním na straně serveru

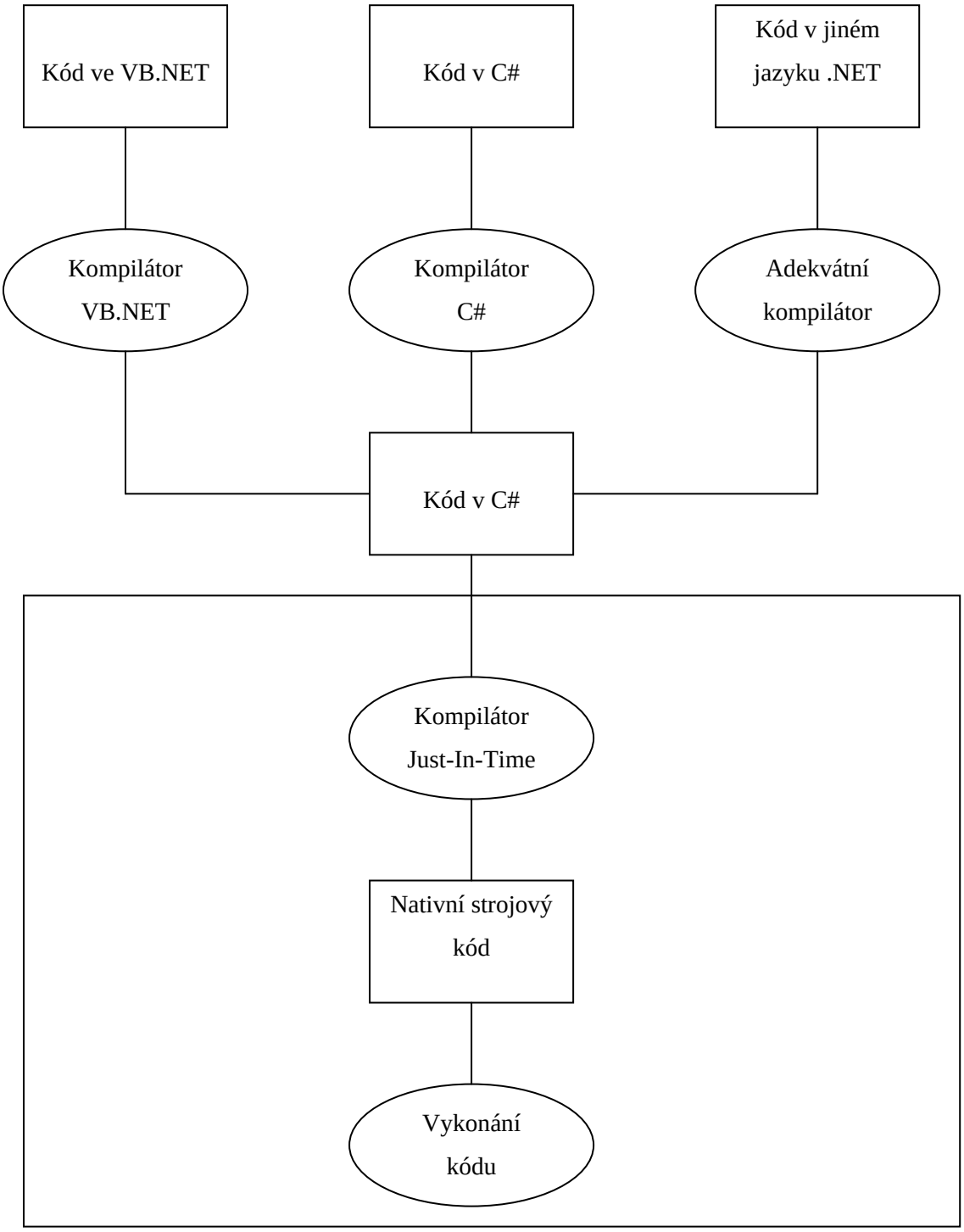
Klasické ASP (Active Server Pages, předchůdce ASP.NET) nebo například PHP spadají do první kategorie. V těchto jazycích je cílem vytvořit soubor se skriptem, který obsahuje vložený kód. Soubor skriptu prozkoumá jiná komponenta, která střídá zpracování běžného HTML kódu a vykonávání vloženého kódu. Asi největší nevýhodou tohoto přístupu je, že se kód aplikace významně prodlužuje. To je jeden z důvodů, že kód je směs skriptu realizovaného na straně serveru a HTML. Webové stránky napsané tímto způsobem mohou velmi narůstat do nezvladatelných délek. Navíc má tento přístup také negativní vliv na výkon aplikace.

Druhý přístup, který ve značné míře využívá Perl přes CGI, přináší zase jiné problémy. V těchto pracovních rámcích spouští webový server oddělenou aplikaci, která má za úkol zpracovat požadavek klienta. Aplikace vykoná svůj kód a dynamicky vytváří HTML, který je poté odeslán zpět klientovi. Tyto aplikace se sice vykonávají rychleji než jejich skriptované protějšky, ale zase spotřebovávají mnohem více paměti. Dále se tento druh aplikací těžko programuje i těžko ladí.

V ASP.NET takové problémy neexistují. Webové stránky se píší s ohledem na tradiční objektově orientované pojmy, a obsahují takové ovládací prvky, že se s nimi pracuje velmi podobným způsobem jako v případě desktopových aplikacích. To znamená, že se nemusí dělat směs z HTML značek a přímého kódu.

Sedm důležitých faktů o ASP.NET

- **ASP.NET je integrováno s .NET Frameworkem** (popis .NET Frameworku níže)
- **ASP.NET se neinterpretuje, ale kompiluje** – u skriptovacích jazyků, když se aplikace vykonává, musí skriptovací hostitel na serveru interpretovat kód a přeložit jej do strojového kódu řádek po řádku, což je značně pomalé. Aplikace se kompilují vždy a je nemožné spouštět kód C# nebo jiného .NET jazyka, aniž by se předtím nejprve zkompiloval. Aplikace ASP.NET procházejí dvěma kompilační etapami. V první etapě se kód C# zkompiluje do přechodného jazyka MSIL (Microsoft Intermediate Language). Druhá úroveň kompilace nastává těsně předtím, než se stránka skutečně vykoná. Kód MSIL se zkompiluje do nativního nízkoúrovňového kódu (operace se nazývá kompilace Just-In-Time). Kompilace je rozdělena na dva kroky, kvůli co nejjednodušší přenositelnosti. Kompilace Just-In-Time se neprovádí pokaždé, když nějaký uživatel požádá o zobrazení stránky. Kód MSIL se vytvoří pouze jednou, a pak se už generuje jen tehdy, pokud dojde k modifikaci zdroje.



Obrázek 2-2: Dva kroky kompilace ASP.NET *zdroj [6]*

- **ASP.NET je vícejazyčné –** Je to proto, že pokud použijete jakýkoliv jazyk z rodiny .NET, tak se kód vždy zkompiluje do MSIL. Tento kód je identický ať se jedná o jakýkoliv jazyk

- **ASP.NET běží uvnitř společného runtime jazyků** – plyne z toho hned několik výhod
  - Automatická správa paměti a svoz odpadků (garbage collection) - .NET se stará o paměť, programátor ji nemusí uvolňovat ručně.
  - Typová bezpečnost – když se kompiluje nějaké aplikace, .NET přidá do assembly informaci, specifikující některé podrobnosti o ní. Výsledkem je, že assembly zkompilovaného kódu je plně soběstačné.
  - Rozšiřitelná metadata – metadata popisují kód a umožňují poskytnout dodatečně informace pro runtime nebo pro jiné služby.
  - Strukturované zpracování chyb – pomocí strukturovaného zpracování výjimek se dá reagovat na chyby, vytvářet oddělené bloky v nichž se budou zpracovávat různé druhy chyb.
  - Multithreading – je poskytován fond vláken, který můžou využívat různé třídy. Můžou se volat metody, číst soubory nebo komunikovat s webovými službami asynchronně, aniž by se museli explicitně vytvářet nová vlákna.
- **ASP.NET je objektově orientované –** ASP.NET je plně objektově orientované prostředí. Kód má přístup ke všem objektům, programátor může vytvářet opětovně využitelné třídy, standardizovat kód s rozhraním nebo začlenit nějakou užitečnou funkcionalitu do zkompilované komponenty určené k šíření.
- **ASP.NET podporuje různá zařízení a různě prohlížeče –** ASP.NET obsahuje bohatě vybavenou skupinu webových ovládacích prvků. Ty realizují svůj kód HTML přizpůsobivě, protože berou v úvahu schopnosti klienta.
- **ASP.NET se snadno rozšiřuje a konfiguruje –** Každá instance .NET Frameworku poskytuje stejné jádro tříd, takže se aplikace ASP.NET rozmisťují relativně snadno. Obvykle stačí pouze zkopírovat všechny soubory do nějakého virtuálního adresáře na ostrém serveru. Pokud má hostitelský stroj .NET Framework, nejsou potřeba žádné registrační kroky, které by mohly být náročné. Většina nastavení ASP.NET se ukládá do soubory web.config, který existuje právě pro tento účel. Obsahuje hierarchická seskupení nastavení aplikace zapsaná ve formátu XML. Takže se dá snadno editovat s pomocích pouhého testového editoru. Soubor web.config není nikdy uzamčen, takže se dá aktualizovat kdykoliv. Více viz [6].

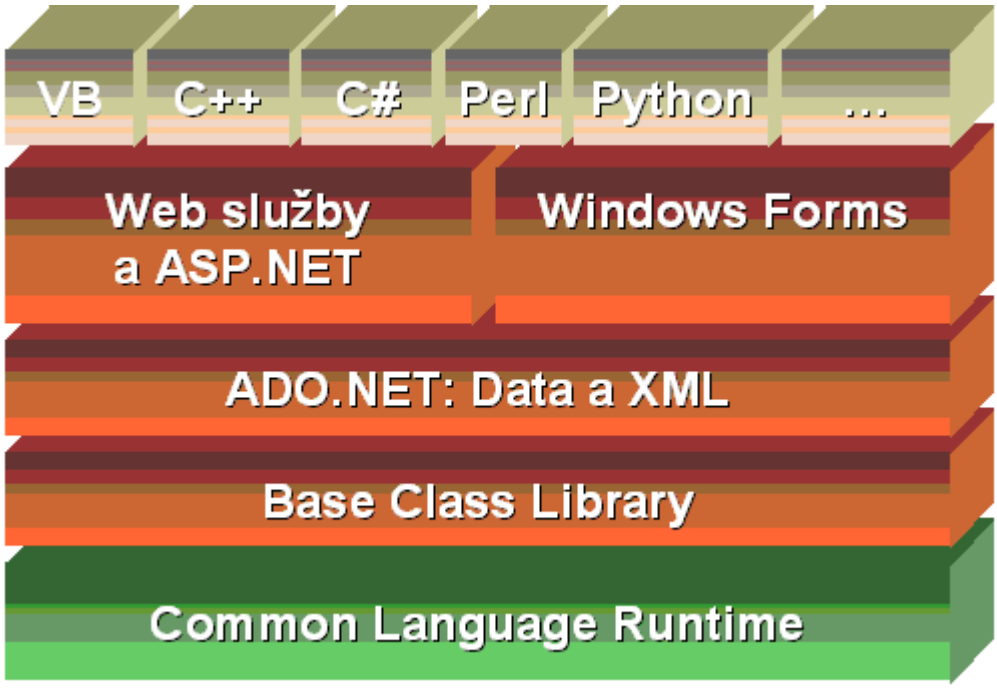
2.5 .NET framework

Pro činnost webových stránek v ASP.NET je třeba komponenta zvaná *Microsoft .NET Framework*. .NET Framework se stará se o věci, které dříve museli vývojáři často řešit a dnes je považují za tak samozřejmé, že si je ani neuvědomují. Například jako teplo v bytě, u kterého také nikdo nepřemýšlí

jak funguje. .NET Framework se podobně „otočením kohoutku“ postará o řadu nízkoúrovňových a nezáživných povinností jakými jsou:

- 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.

O tyto téměř neviditelné, ale velmi důležité operace se stará část zvaná *Common Language Runtime* (CLR) – viz obrázek 2-3 celého .NET frameworku:



Obrázek 2-3: Architektura .NET *zdroj [4]*

Base Class Library (BCL) je knihovna obsahující nejčastější pomocné funkce – práci se soubory, třídění, diagnostiku, síťovou komunikaci apod. ADO.NET je knihovna pro práci s daty s možností jejich XML reprezentace. Dále jsou na obrázku dvě knihovny pro vývoj uživatelského rozhraní – Windows Forms pro desktopové aplikace a ASP.NET pro webové uživatelské rozhraní.

Velmi důležitou částí .NET frameworku jsou podporované jazyky. .NET framework je jazykově nezávislý, pro libovolnou úlohu lze v principu použít jakýkoliv z podporovaných jazyků. Ne všechny jazyky jsou ale vyvíjeny Microsoftem, řada jich je vyvíjena partnerskými firmami. Z jazyků vyvíjených Microsoftem nemají všechny stejnou vizuální podporu pro vývoj toho kterého typu aplikace. Po těchto vylučovacích kritériích nám zůstanou dva použitelné jazyky – C# a Visual Basic.NET. Více viz [4].

2.6 Jazyk C#

C# je vyvinutý jazyk pro Microsoft .NET. Je navržen pro maximální využití této platformy. Jedná se o silně objektově orientovaný jazyk vycházející z programovacích jazyků Java a C++. Stejně jako tyto jazyky je i C# case-sensitive. V tomto jazyce je realizováno 80% základních knihoven .NET frameworku. I přesto, že je koncipován hlavně pro psaní řízeného kódu, na jehož užití je platforma .NET postavena, lze jej v případě potřeby využít i pro tvorbu kódu neřízeného(bloky unsafe). Použití neřízeného kódu znamená, že běhové prostředí CLR neověřuje zda-li je napsaný kód bezpečný (například se neověřuje jinak vyžadovaná typová bezpečnost).

Výčet několika vlastností jazyku C#:

- **Třídy** – základní stavební prvek při tvorbě objektově orientovaných aplikací obsahující akce (metody) a atributy.
- **Struktury** – lze je chápat jako zjednodušené třídy, jejich užitím jsou nejčastěji popisovány vlastní datové struktury.
- **Výčtové typy**
- **Vlastnosti** – někdy označované jako chytré proměnné. Pole a jejich „chytrá“ verze nazývaná indexery.
- **Zástupci** – typově bezpečné ukazatele na funkce.
- **Události** – druh zástupců sloužící ke zpracování asynchronních operací.

Pro realizaci diplomového projektu jsem se rozhodoval mezi programovacími jazyky C# a Visual Basic.NET. Síla obou programovacích jazyků i jejich možnosti jsou stejné (jako všechny jazyky .Net). V minulosti jsem již programoval v jazyce C a proto jsem se rozhodl v projektu použít jazyk C#, který je syntakticky jazyku C velmi podobný. Navíc syntaxe jazyku Visual Basic.NET je pro mě méně přehledná.

2.7 ADO.NET

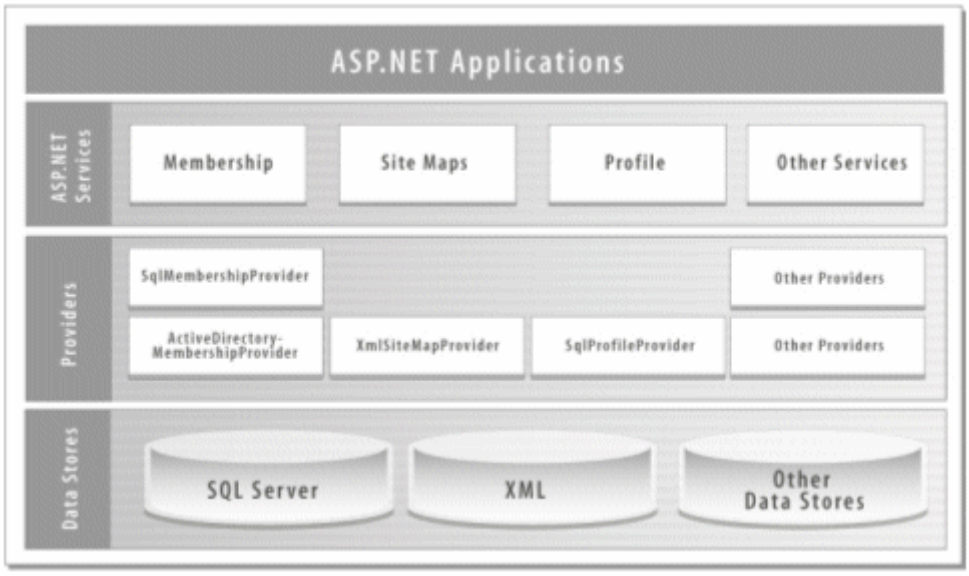
ADO.NET představuje soubor typů (tříd), které jsou vytvořeny pro přístup k datům v různých datových zdrojů v technologii .NET. Typy spojené s komponentou ADO .NET se nacházejí ve jmenných prostorech rozšiřující jmenný prostor **System.Data** (*System.Data.Common, System.Data.SqlClient, System.Data.Odbc* atd.). Přístup k datům v různých rozhraních, slouží k tomu určená rozhraní (*IDbConnection, IDbCommand, IDataReader, IDataAdapter* atd.) tato rozhraní předepisují funkčnost, kterou musí splňovat jednotliví zprostředkovatelé přístupu ke konkrétnímu datovému zdroji, pro který je zprostředkovatel implementován. Tito zprostředkovatelé jsou nazýváni

ADO.NET data providers. Při použití přístupu pomocí těchto rozhraní je dosaženo toho, že kód takto napsaný bude fungovat pro jakýkoliv datový zdroj.

Podle způsobu jakým aplikace komunikuje s datovým zdrojem se dají tyto způsoby rozdělit na komunikaci on-line a komunikaci off-line. Při on-line komunikaci je aplikace trvale připojena k datovému zdroji. To se používá pokud je potřeba přistupovat k datům, která jsou často měněna a je nutné mít stále co nejaktuálnější data. Nevýhoda tohoto řešení je zvýšená komunikace mezi aplikační a datovou vrstvou. V ADO.NET se na tento způsob používá kombinace jednotlivých implementací rozhraní *IDbConnection*, *IDbCommand* a *IDataReader* pro konkrétní datový zdroj. Novějším způsobem je off-line komunikace. V tomto případě jsou data získána z datového zdroje do aplikace a po té je spojení s datovým zdrojem ukončeno. Od chvíle odpojení od datového zdroje aplikace manipuluje s obrazem získaných dat, který je uložen v paměti. Po dokončení potřebných modifikací dat na úrovní aplikační logiky je opět vytvořeno aktivní spojení k datovému zdroji, zjištěny rozdíly mezi daty z aplikace a mezi daty v datovém zdroji a po té jsou tyto rozdíly promítnuty do příslušného datového zdroje. Toto řešení je vhodné pro aplikace, u kterých chceme redukovat komunikaci mezi jednotlivými vrstvami aplikace. To se hodí v případech, kdy jsou data a aplikace na různých počítačích a je potřeba snížit provoz na síti. Toto řešení se nehodí, pokud by k datům přistupovalo hodně uživatelů a mohly by nastat konflikty při modifikaci dat. Více viz [4].

2.8 Provider model

Microsoft ASP.NET 2.0 obsahuje množství služeb (providerů), které jsou uložené v databázích a na ostatních zálohovacích médiích. Pomocí těchto služeb si může programátor ušetřit spoustu práce a spoustu napsaného kódu, protože spoustu věcí nebude muset programovat, ale místo svého kódu využije právě providery. Například Membership Provider, který poskytuje programátorům kompletní servis pro přihlašování uživatelů do systému, aplikace. Nevýhodou těchto od Microsoft připravených SQL providerů je, že vyžadují velice specifickou databázovou strukturu, ale zase jsou vytvořeny velice univerzálně a počítají s velkým množstvím možností, které by při tvorbě aplikací mohl programátor potřebovat (proto také ta specifická databázová struktura, která všechny tyto možnosti zahrnuje). Tím se stávají pro většinu běžných aplikací nepoužitelných. Provider model, ale umožňuje tvorbu vlastních providerů, takže si je může programátor naprogramovat přesně podle potřeb své aplikace a své databáze, což je asi největší výhodou tohoto modelu. Více viz [11]

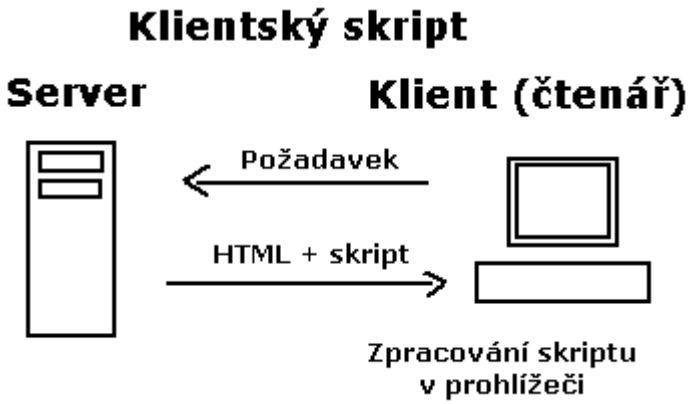


Obrázek 2-4: Provider Model v ASP.NET *zdroj [11]*

2.9 JavaScript

JavaScript je programovací jazyk, který se používá v internetových stránkách. Zapisuje se přímo do HTML kódu, což je velká výhoda, protože je to jednoduché.

JavaScript je klientský skript. To znamená, že se program odesílá se stránkou na klienta (do prohlížeče) a teprve tam je vykonáván. (Protikladem klientských skriptů jsou skripty serverové, které jsou vykonávány na serveru a na klienta jdou už jen výsledky.)



Obrázek 2-5: jak pracuje JavaScript *zdroj [9]*

JavaScript je často zaměňován s Javou. Java je samostatný programovací jazyk. Má s JavaScriptem pouze podobnou syntaxi.

Charakteristiky jazyka

JavaScript je jazyk

- **Interpretovaný** – nemusí se kompilovat
- **objektový** – využívá objektů prohlížeče a zabudovaných objektů
- **závislý na prohlížeči** – funguje ale ve většině prohlížečů
- **case sensitivní** – záleží na velikosti písem v zápisu
- syntaxí podobný jazykům C, Java a podobným

Omezení jazyka

- JavaScript funguje pouze v prohlížeči.
- Uživatel může JavaScript zakázat
- Existují různé odlišné verze jazyka i prohlížečů, což vede k častým chybám.
- Neumí přistupovat k souborům (kromě cookies) ani k žádným systémovým objektům.
- Neumí žádná data uložit (kromě cookies).

To vše z něj dělá pouze jazyk druhořadý, účelově použitelný pouze v HTML stránkách. Více viz [9].

2.10 SQL

SQL je anglickou zkratkou Structured Query Language (strukturovaný dotazovací jazyk) a jedná se o neprocedurální jazyk. (Pomocí procedurálního jazyka říkáme, *jak chceme konkrétní věc provést*, pomocí neprocedurálního jazyka naopak říkáme, *co chceme provést*.) První prototypová implementace tohoto jazyka vznikla ve firmě IBM v roce 1974 a až do konce sedmdesátých let byla pod názvem Sequel a byla použita ve firemním systému s názvem R. Cílem bylo (a stále je) poskytnout vývojářům standardní metodu přístupu k datům uloženým v databázovém systému, která by byla nezávislá na dalších použitých vývojových nástrojích. Jedná se o velmi mocný jazyk, který je navíc velice jednoduchý. Stačí si přeložit do angličtiny, co vlastně chceme udělat a samotný SQL příkaz bude vypadat velice podobně. Ale nelze jej považovat za samostatně dostačující k tvorbě kvalitní aplikace, protože v něm nejsou implementovány například příkazy pro uživatelský vstup a výstup. Jazyk *SQL* také není možné chápat striktně jako dotazovací - samotný *SQL* se skládá z několika částí. Některé části jsou určeny pro administrátory a návrháře databázových systémů, jiné pak pro koncové uživatele a programátory. První částí jazyka SQL je jazyk *DDL* - Data Definition Language. Jedná se o jazyk pro vytváření databázových schémat a katalogů. Způsob ukládání tabulek definuje jazyk *SDL* - Storage Definition Language. Třetí částí pro návrháře a správce je jazyk *VDL* - View Definition Language, určující vytváření pohledů (pohled si lze představit jako virtuální tabulku složenou z různých jiných tabulek). Poslední částí, kterou se budu převážně zabývat, je jazyk *DML* -

Data Manipulation Language, který obsahuje základní příkazy `INSERT`, `UPDATE`, `DELETE` a nejpoužívanější příkaz `SELECT`. S jazykem DML pracují nejvíce koncoví uživatelé a programátoři databázových aplikací.

Přestože je SQL spojováno hlavně s některým SŘBD (Systém Řízení Báze Dat), není nikde stanoveno, že by nebylo použitelné v systémech založených na jiném modelu nebo že všechny relační databáze musí *SQL* podporovat. Nikde také není řečeno, že se musí jednat o jazyk interpretační či vyžadující kompilaci. Možné jsou všechny tři varianty - čistá interpretace, překlad do pre-kódu i překlad do samostatně spustitelného kódu. Každá varianta má své výhody a nevýhody. Například, pokud se jedná o program zkompilovaný do samostatně spustitelného kódu, je nutno zajistit jeho rekompilaci, pokud dojde k nějaké změně v použitých databázových strukturách.

Na trhu se v první polovině 80. let objevily asi čtyři desítky komerčních *SQL* systémů. Situace začala být poměrně vážná, a tak se přikročilo k prvnímu normalizačnímu kroku. Organizace *ANSI* přijala roku 1986 jako standard variantu společnosti *IBM* a ve svém důsledku se jednalo o jakýsi průnik existujícími implementacemi. Tento standard bývá označován jako *ANSI SQL* či jako *SQL86*. O rok později byl přijat také v *ISO*. Roku 1989 pak následoval další standard, který umožnil definovat integritní omezení. Vývoj šel nezadržitelně kupředu a roku 1992 dochází k přijetí standardu *SQL92*, o kterém se již v době vzniku vědělo, že se nejedná o konečnou podobu standardu *SQL*. Proto nabízí dvě úrovně pro překlenutí určitého období. Jedná se o tzv. vstupní úroveň (*Entry SQL*) a tzv. prostřední úroveň (*Intermediate SQL*). Zatímco v případě prvně jmenované se jedná jen o drobné kosmetické úpravy oproti předchozím standardům, *Intermediate SQL* již obsahuje celou řadu nových prvků. Ty se objevují již několik let v mnoha *SŘBD*, bohužel však často s mírně odlišnou syntaxí (a vzhledem k někdy nejasné implementaci i sémantikou). Další nevýhodou (z tehdejšího pohledu) byla nepřítomnost triggerů, přestože se v mnoha systémech objevují od roku 1989. Proto byl přijat standard označovaný jako *SQL3*, který nabízí mimo jiné například možnost psát uložené procedury. Proces standardizace je sice prospěšný, nese však s sebou i jednu hlavní nevýhodu - zpětnou kompatibilitu (obdobně jako v případě mikroprocesorů či operačních systémů pro osobní počítače). Více viz [4].

2.11 MSSQL

Microsoft SQL server je relační databázový systém. Je primárně založen na dotazovacím jazyku Transact-SQL, což je implementace ANSI/ISO standardu SQL. MSSQL Server byl původně využíván firmami pro malé a středně velké databáze, ale v současnosti se už vyvinul na produkt, který je schopný vytvářet ty největší databáze. MSSQL není původně produktem Microsoftu, ale společnosti Sysbase a později upravován společností Oracle. Microsoft jej pouze začal komerčně využívat a na jeho vývoji se začal podílet až s jeho úspěchem.

MSSQL Serve Express Edition je server, který běží na stejném jádru jako klasický MSSQL Server, ale je ochuzen o některé pokročilé funkce. Více viz [1].

Tuto verzi jsem ve svém projektu použil. Rozhodl jsem se tak z několika důvodů. Tím prvním důvodem bylo, že tato verze se nainstalovala s Visual Studiem a nemusel jsem tedy instalovat žádný další systém. Navíc DNN tuto verzi databáze sám standardně podporuje a nebylo třeba žádných úprav v souboru web.config. DNN samozřejmě podporuje i jiný databázové systémy jako jsou například Oracle nebo MySQL. MySQL jsem používal ve svém ročníkovém projektu a v diplomové práci jsem si chtěl vyzkoušet a naučit se něco nového, tak jsem se rozhodoval mezi Oracle a MSSQL. Pro MSSQL jsem se rozhodl hlavně kvůli dostupnosti MSSQL Express Edition.

2.12 Informační systém

Systém je účelově definovaný soubor komponent, mezi kterými existují určité vztahy a které jsou definovány k určitému objektu.

Objektem může být například uživatelská aplikace (knihovna, úřad apod.). Uživatelské aplikaci se proto často říká výsek reálného světa nebo svět objektů. Hranice systému vymezuje samotný systém nebo odděluje více systémů. Logická hranice je pomyslnou hranicí a vymezuje podsystémy v rámci systému, ovšem okolí systému je již „viditelnou“ hranicí. Prvky vně hranice pak ovlivňují chování systému.

V systémech může nastat zpětná vazba, kdy výstupní veličina opětovně ovlivňuje vstupní veličinu, a tedy i samotný systém. Každý systém má tedy tendence být nestabilní. Podle tvaru zpětné vazby můžeme systémy dělit na systémy s otevřeným cyklem a na systémy s uzavřeným cyklem. Systémy s otevřeným cyklem jsou takové systémy, které neobsahují zpětnou vazbu a které neobsahují prvky ukazující na cíl (například ponorný elektrický ohřívač, který topí, když je zapnut v elektrice, ale nelze jej nijak regulovat a tím řídit výstup). Systémy s uzavřeným cyklem jsou takové systémy, které obsahují zpětnou vazbu i prvky ukazující na cíl.

Systémy se podle interakce s okolím dále dělí na otevřené systémy a na uzavřené systémy. Otevřený systém je připojen k okolí tokem zdrojů a uzavřený systém s okolím nijak nekomunikuje a není na něm nijak závislý. Podle typů zdrojů se systémy dále dělí na systémy fyzické a na systémy konceptuální. Fyzický systém pracuje nad fyzickými zdroji (firma, podnik) a konceptuální systém obvykle reprezentuje fyzický systém a užívá konceptuální zdroje jako jsou informace a data.

Systémy můžeme dělit i na tvrdé a měkké. Tvrdé systémy jsou spojovány s jedním specifickým (strukturovaným) problémem (většinou technické systémy), naopak v měkkých systémech vystupuje celá řada faktorů, jsou obecnější (člověk je aktivním prvkem systému=individualita - problém protože každý má jiné znalosti a jinak uvažuje).

Informační systémy obecně chápeme jako systémy pro zpracování dat a jejich definice zní: Jde o konceptuální, otevřený systém, který pokud je ve spojení s objektem, tak jde o systém s uzavřeným

cyklem. Informační systémy se dělí na systémy pro zpracování dat a procesů (databáze) a na komunikační systémy (připojení k síti. Informační systém lze realizovat i bez výpočetní techniky, ale v současné době se tak již nedělá. Jejich hlavním účelem je řešení informačních procesů.

Informační systémy se dají členit podle převládající funkce na systémy dokumentaristické, faktografické a na měřící a regulační systémy. Členění informačních systémů podle režimu činností obsahuje tyto režimy: individuální zpracování požadavků, dávkové zpracování dat, zpracování dat v reálném čase v síti, zpracování dat v centralizovaných databázích a na zpracování dat v distribuované bázi dat na síti.

Informační systémy se dají klasifikovat podle informačního prostředí, organizační úrovně řízení, převládající funkce a režimu činnosti. Členění informačních systémů podle informačního prostředí je členění podle typu objektů. To vede ke vzniku typových projektů (například knihovny, účetnictví atd.). U členění informačních systémů podle organizační úrovně se uplatňuje hierarchie řízení respektive vertikální členění organizací.

Důležitým prvkem dnešních informačních systémů jsou systémy řízení báze dat (SŘBD). U SŘBD nemá uživatel přímý přístup k datům (tak to bylo často hlavně u dřívějších systémů), ale přístup k nim se odehrává v SŘBD. Samotná data jsou uložena v databázi. Data jsou nezávislá na uživatelských programech a naopak (což je hlavní přínos databázové technologie). Programy jsou osvobozeny od popisu dat a tím jsou zcela nezávislé na fyzickém uložení dat. V programech se vyskytují pouze odkazy na tyto popisy. Data jsou udržovány jednotně a centrálně, čímž se snižuje jejich redundance a zvyšuje se konzistence dat. Díky tomu je snadné navrhovat informační systémy, které vyžadují více zdrojů a přístup k těmto zdrojům je jednotný. Více viz [7]

2.13 WYSIWYG

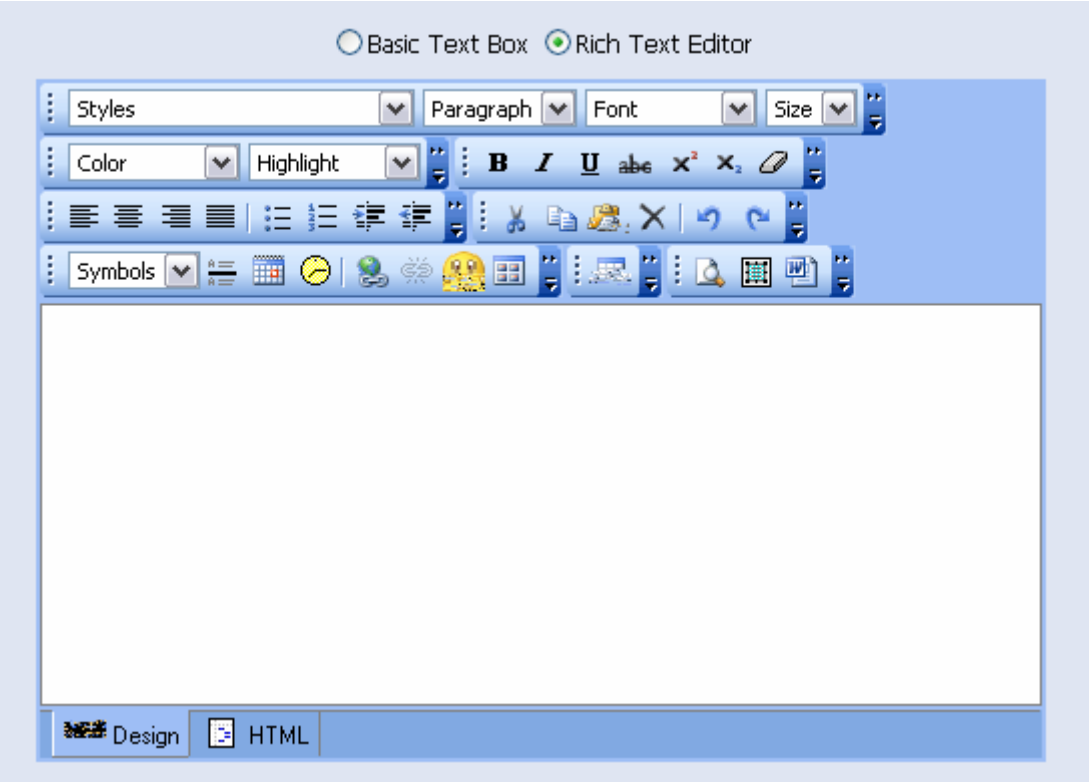
WYSIWYG je zkratkou anglických slov What you see is what you get, což přeloženo do češtiny znamená: Co vidíš, to dostaneš. Tyto slova označují takový způsob editace textu v počítači, při kterém vidíme je na obrazovce zobrazeno přesně totéž, jako ve výsledném dokumentu. Mezi neocenitelné výhody WYSIWYG editoru patří především komfort s jakým umožňuje uživatelům pracovat s texty. S komfortem samozřejmě úzce souvisí skutečnost, že prostředí editoru není něčím neznámým, s čím bychom se museli nově seznamovat, ale je obvyklým prostředím se kterým se v jistých obměnách setkáváme u celé řady softwarových produktů Nejčastěji se jako WYSIWYG editory označují textové procesory. Jako je například Microsoft Word.

V současné době jsou době jsou hodně populární tzv. online WYSIWYG editory. Jde o editory textu, které jsou dostupné z internetové stránky. Mohou se používat pro formátování textu v dokumentech (souborech, stejně jako třeba ve Wordu) nebo na úpravu textů na samotné stránce. Druhá možnost je v současnosti velmi populární u systémů pro správu obsahu či redakčních systémů,

které by bez něj byly téměř nepoužitelné, protože tím by ztratily možnost vytvářet stránky bez znalosti HTML a kaskádových stylů.

Současné online WYSIWYG editory jsou vytvářeny v JavaScriptu. Má to své výhody i nevýhody. Jako hlavní výhoda, je že WYSIWYG editor neběží na serveru, ale na počítači uživatele a odlehčí se tak serveru. Navíc je to univerzální jazyk, který běží ve všech dnešních běžně používaných internetových prohlížečích. Nevýhoda tohoto řešení je, že JavaScript má poměrně omezené možnosti a nikdy nepůjde pomocí něho naprogramovat stejně dobrý WYSIWYG editor, jako pomocí některého z klasických programovacích jazyků (C++,C# apod.). Další problém může nastat pokud má uživatel v nastavení internetového prohlížeče JavaScripty zakázány. Pak se WYSIWYG editor vůbec nezobrazí.

Další nevýhodou online WYSIWYG editorů je, že uživatel nedostane co vidí, tedy že návrh a výsledek se mohou lišit. U klasických WYSIWYG editorů je prioritou vizuálním uspořádání (ze kterého si lidský mozek odvodí strukturu, tj. co je nadpis, co je text a co je popiska obrázku). Ale web má úplně jiná specifika a hlavní je pro něj právě struktura. Takže WYSIWYG editor se v první řadě snaží o správný zápis tagů a generování kódu. Z toho poté vyplývá, že ne vždy pak výsledek odpovídá našemu očekávání. Více viz [1].



Obrázek 2-6: Ukázka WYSIWYG editoru v systému DNN

Na obrázku 2-6 je uveden příklad, jak takový online WYSIWYG editor může vypadat. V tomto případě jde o WYSIWYG editor ze systému DNN. Je z něj vidět, že možnosti online WYSIWYG editorů nabízí podobné úpravy textu jako například MS Word.

3 DotNetNuke

DNN je systém pro správu obsahu vytvořený v prostředí ASP.NET 2.0 a v jazyce Visual Basic.NET. Tento systém byl původně vyvíjen společností Microsoft, ale v současnosti se jedná o open source projekt, který se neustále vyvíjí.

3.1 Historie DNN

Když Microsoft vydával na trh .NET Framework 1.0 Beta, tak spolu sním vydal spoustu projektů a zdrojových kódů vytvořených právě v této platformě. Všechny tyto projekty vydávány pod licencí EULA (End User Licence Agrament). Mezi těmito projekty byl IBuySpy Portal, který Microsoft vyvíjel společně s firmou Vertigo software a ten měl sloužit jako nejlepší praktická ukázka, jak mohou vypadat aplikace vytvořené v novém prostředí ASP.NET. Také byla tato aplikace vyvinuta jako odpověď na aplikace tvořené v Linuxu/Apachi/MySQL/PHP. IBuySpy Portal umožňoval vytvořit kompletní dynamický web, který mohl obsahovat neomezené množství stránek. Každá stránka obsahovala standardní hlavičku a tři panely – levý panel, pravý panel a střední panel (to byly standardní součásti u většiny podobných aplikací). IBuySpy Portal byl votvořen ve dvou programovacích jazycích, Visual Basic.NET a C#.

Microsoft ve stejné době spustil diskusní fórum (www.asp.net). Na něm se vytvořila silná komunita uživatelů. Několik diskusí se věnovalo právě IBuySpy Portalu. Uživatelé na těchto diskuzích vytvořili několik úprav. I když se Microsoft snažil velmi agresivně upřednostňovat svůj nový jazyk C#, tak většina těchto úprav byla prováděna v jazyce Visual Basic.NET. Spojením IBuySpy Portalu a jeho úprav, vznikl IBuySpy workshop, který byl vytvořen právě v jazyce Visual Basic.NET a který se dá považovat za přímého předchůdce DNN.

Tato aplikace se stala rychle velice populární a proto se na diskuzích o IBuySpy workshopu se vytvořil tým, který začal z této aplikace vyvíjet DNN. Název DotNetNuke byl zvolen jako odpověď na aplikaci PHP-nuke. Hlavním důvodem vytvoření tohoto týmu byla snaha, aby šel vývoj DNN jedním směrem a aby byl i nadále konkurencí pro ostatní aplikace.

V současnosti (9.5.07) je DNN již ve verzi 4.5.1. Tato verze je již vytvořena pomocí ASP.NET 2.0. S IBuySpy Portalem již z hlediska zdrojového kódu nemá téměř nic společného a spojuje je už jen stejný koncept. Kolem DNN se za tu dobu vytvořila silná komunita uživatelů, kteří se podílí na

jeho dalším vývoji. Svědčí o tom například diskusní fórum na domovské stránce DNN, které obsahuje více jak 36 000 témat. Více viz [10]

3.2 Jak DNN definují tvůrci

DNN je open source verze systému pro správu obsahu (Content Management System) vhodný pro vytváření a nasazování webových projektů jako jsou například komerční webové stránky, korporátní intranety a extranety a online publikační systémy a portály. Je to také projekt s jasnou vizí: vyvíjet světově známý software na základě zapojení komunity a vzájemného sdílení znalostí.

DNN je poskytován jako open-source software sířený pod licenčním ujednáním typu BSD. Obecně tato licence dává právo získat software bez jakýchkoliv poplatků. Kromě toho dává tato licence uživatelům právo nakládat s tímto software libovolně a to jak v případě komerčního tak nekomerčního využití, a to s jedinou podmínkou, aby uživatelé napomáhali rozvoji komunity kolem projektu DNN.

DNN je postaven na platformě Microsoft ASP.NET (VB.NET) a lze jej velmi snadno instalovat a spravovat. Díký rostoucí komunitě (dnes již přes 160.000 registrovaných uživatelů) a dedikované skupině vývojářů a profesionálů, kteří jsou základní vývojářskou skupinou DNN, je podpora pro DNN vždy po ruce a prakticky okamžitě dostupná. Více viz [5].

3.3 Vlastnosti DNN

Na jedné instalaci DNN lze provozovat několik samostatných a nezávislých portálů (webů). Jejich počet není nijak omezen. Tyto portály se dají rozdělit na dvě skupin. Rodičovský portál a následnický portál (child portal). Rozdíl mezi těmito portály je ve způsobu jaký se k nim dá dostat. Na rodičovský portál se návštěvník dostane po zadání URL tohoto typu: *www.RodičovskýPortal.com*. K následnickému portálu se musí přistupovat přes jeho rodičovský portál, takže URL pro následnický portál bude vypadat například takto: *www.RodičovskýPortál.com/NáslednickýPortál*. Při vytváření nového portálu v DNN si uživatel může zvolit kromě jiných vlastností a nastavení také jakého typu tento portál bude.

Stránky (tabs)

Na každém portálu lze vytvořit libovolné množství stránek (v DNN pojmenovaných jako záložky). Ty slouží pro zobrazování obsahu portálu respektive k zobrazování modulů dostupných pro danou stránku. Stránky se v jednotlivých portálech od sebe dělí podle levelu zanoření. Pokud je level zanoření stránky jedna, tak je stránka zobrazena v hlavním menu portálu. Pokud má stránka mít level zanoření vyšší, tak je potřeba jí nastavit rodičovskou stránku. Přes ni potom půjde ke stránce přistupovat (stránky jsou zobrazovány v roletových menu pod svou rodičovskou stránkou). Stránka

může být vytvořena nová nebo může být zkopírována nebo mohou být moduly na stránku nakopírovány z jiné stránky.

Moduly

Na každou stránku lze vložit libovolné množství modulů. Tyto moduly se dají považovat za základní stavební kameny celého DNN. V DNN slouží moduly pro zobrazování samotného obsahu stránek. Ten v DNN jinak než přes moduly zobrazovat nejde. DNN nepodporuje jeden univerzální typ modulů, ale množství specializovaných typů modulů. Takže existují moduly, pro psaní a zobrazování textu, moduly pro vytváření galerií obrázků apod. Pomocí modulů se dají rozšiřovat možnosti DNN a přidávat mu jimi funkce, které po instalaci nemá.

Skiny

DNN umožňuje změnu vzhledu celého portálu. Tato změna se provádí pomocí skinů. Skin je vlastně šablona, ve které jsou uloženy informace o vzhledu, obrázky, rozvržení HTML atd. Skinů může být nahráno v DNN libovolné množství. Z nich si administrátor vybírá, který použije.

Kontejnery

Kromě skinů existuje ještě jedna možnost jak měnit vzhled DNN. Vzhled se dá ještě měnit pomocí kontejnerů. Ty se už aplikují pouze na moduly. Takže pomocí kontejnerů se dá měnit vzhled modulů. Pomocí kontejnerů se tak dá pozměnit vzhled modulů takovým způsobem, že každý modul bude mít jiný vzhled. Kontejner lze také použít na všechny moduly na stránce nebo na všechny moduly na portálu.

Uživatelé v DNN

Pro zajištění větší bezpečnosti je v systému DNN vytvořeno několik typů (rolí) uživatelských účtů. Jsou to účty host, administrátor, přispěvovatel, registrovaný uživatel a anonymní uživatel.

Anonymní uživatel je nejpoužívanějším typem uživatele v DNN. Zastupuje v něm běžného návštěvníka stránek, který se nepřihlásí do systému. Tento uživatel má práva prohlížet stránky a moduly, které nevyžadují přihlášení nebo speciální oprávnění.

Dalším uživatelem v DNN je registrovaný uživatel. Má stejná práva jako anonymní uživatel a navíc může tento uživatel prohlížet stránky a moduly, které vyžadují přihlášení do systému. Může také vytvářet a upravovat obsah u vybraných modulů.

Dalším uživatelem DNN je Subscriber (přispěvovatel). Tento uživatel je v systému vytvořen pro vytváření a spravování obsahu vybraných modulů. Moduly, které bude spravovat mu přiřazuje uživatel Administrátor nebo Host. Ostatní práva jsou stejná jako u registrovaného uživatele.

Důležitým uživatelem v DNN je Administrátor. Tento uživatel má práva pro spravování jednoho portálu, který mu přiřadí uživatel Host. Na tomto portálu může spravovat všechny stránky. V systému má vytvořeno své vlastní administrátorské menu.

Nejdůležitějším uživatelem v DNN je uživatel Host. Tento uživatel má práva na všechny portály vytvořené v DNN. Jako jediný uživatel může vytvářet nové portály a odstraňovat existující portály. Také může spravovat obsah a nastavení jednotlivých portálů. V systému má vytvořeno své vlastní Host menu. Více viz [10]

3.3.1 Shrnutí

DNN je zpočátku po instalaci použitelný pouze pro jednoduché prezentace či blogy. Obsahuje v sobě sice možnost vytvořit si vlastní fórum nebo blog, ale jinak je to systém poměrně chudý a nabízí jen málo možností, kterými disponují komerční systémy. Pokud v něm chce tvořit své soukromé stránky běžný uživatel, tak bude DNN dostačující, ale pro komerční využití se příliš nehodí, protože nároky firem jsou vyšší a DNN je po základní instalaci nemusí splňovat. Může, ale být snadno rozšiřován o další moduly. Díky nim již pak může konkurovat i komerčním aplikacím. Některé moduly jsou na internetu volně k dispozici, ale většinu z nich je potřeba zakoupit (čímž se zčásti stává také komerční aplikací a ztrácí tak zčásti výhodu v tom, že je poskytován zdarma). Dá se to, ale vynahradit tím, že moduly do DNN si každý může naprogramovat sám.

3.3.2 DNN pro programátora

Pro programátory bych DNN charakterizoval jako aplikační framework pro rychlý vývoj internetových aplikací. Je to open source projekt a je snadno rozšiřitelný. Prakticky každý, tak může programovat moduly (rozšíření). Velmi tomu napomáhá dobrá dokumentace a také velká komunita, která se již stačila vytvořit. Programátor, tak nemusí složitě pátrat, co a jak funguje a kde co sežene. Literatura o DNN už tak snadno dostupná není. Je pouze pár publikací a pouze v angličtině.

3.3.3 DNN pro uživatele

DNN lze chápat jako stavebnici s různými součástkami. Tvůrce stránek si vybere jaké součástky chce použít a z nich stránky vytvoří. Nemusí k tomu umět programovat, ale stačí mu základní znalosti o PC. Pokud umí obsluhovat MS Office nebo jiný podobný program nebude mít s tvorbou problémy. I tak je nezbytnost přečíst si manuál, protože nabídky a ovládání jsou občas nezvyklé.

3.3.4 Největší výhody a nevýhody DNN

Výhody

- **DNN je open source projekt** – Plyne z toho hned několik výhod. První výhodou je, že DNN je poskytován zdarma. Další výhoda, která z toho plyne, že zdrojový kód DNN je volně k dispozici a každý se může podílet na jeho vylepšování.

- **Je snadno rozšiřitelný pomocí modulů –** Na internetu jsou v současnosti ke stažení stovky modulů, kterými se dají rozšířit možnosti a funkčnost DNN. Velké množství těchto modulů je k dispozici zdarma. Náročnější moduly jsou prodávány komerčně.
- **Vhodný i pro uživatele, kteří neumí programovat –** V DNN si je schopen udělat kvalitní stránky každý, kdo slušně zvládá ovládání PC a ovládání programů typu MS Word. To platí v případě, že DNN je už nainstalován a připraven k použití. Instalace totiž není zdaleka triviální a vyžaduje znalosti ASP.NET (nastavování souboru web.config) a znalosti databází (tvorba nové databáze pro DNN)
- **DNN má kolem sebe už velkou komunitu nadšenců a uživatelů a snadno se dají získat potřebné informace o DNN –** Jelikož se DNN vyvíjí již dlouhou dobu, tak se kolem něj stačila vytvořit silná komunita uživatelů. Toto tvrzení dokazuje například diskusní fórum na domovské stránce DNN (*www.dotnetnuke.com*), které obsahuje přes 36 000 témat, týkajících se DNN. Dále existuje množství menších fór a stránek, které se DNN nebo moduly pro DNN zabývají.
- **DNN dává uživateli vybrat, co na svých stránkách bude mít a co ne –** Celá stránka v DNN tvoří pomocí modulů. Žádný z modulů nemusí být na stránce povinně, takže má administrátor webu možnost si vybrat pouze ty moduly, které na své stránce potřebuje a nejsou mu nuceny takové, které nikdy nevyužije.

Nevýhody

- **Často velmi podobné weby vytvořené v DNN –** Weby vytvořené pomocí DNN jsou si často velmi podobné. Je to dáno tím, že jsou založeny na modulech. Takže stránka vždy obsahuje několik viditelně oddělených modulů. To tyto weby asi nejvíce spojuje. Dále také to, že k dispozici není tolik skinů jako například modulů, takže není zajištěn tak velký výběr.
- **Nekompatibilita různých verzí –** DNN prochází stálým vývojem a proto bylo vydáno již několik verzí DNN. V současné době je nejaktuálnější verze 4.5.1. Ta je ovšem zpětně kompatibilní pouze s moduly vyrobené pro verze 4.x.x. Moduly vyrobené pro starší verze nelze ve verzích 4.x.x použít.
- **HTML kód** – HTML kód, který generuje DNN je velice nepřehledný a nečitelný. Dal by se přirovnat například ke kódům, které generuje MS Front Page.

3.4 Architektura DNN

DNN je vytvořen pomocí těchto technologí:

- ASP.NET 2.0
- Visual Basic.NET
- Webové formuláře

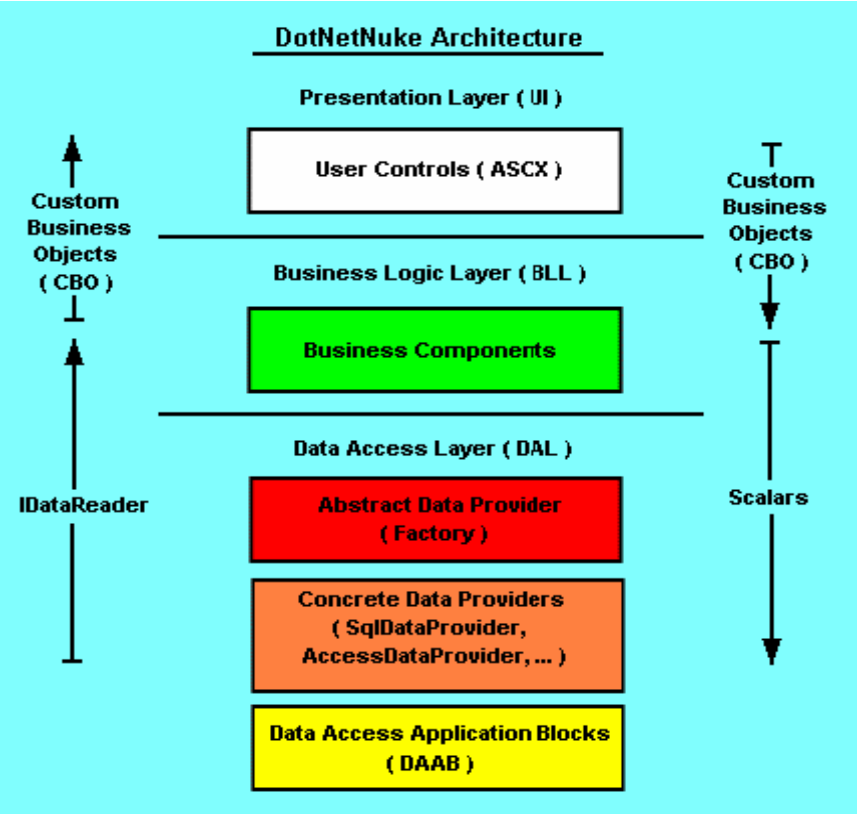
- Microsoft Internet Information Services
- ADO.NET
- Microsoft SQL Sever 2005

Pro přehlednost a zachování funkcionality při provádění zněm, využívá DNN Provider model. Proto nemusí být při každé změně přepisován zdrojový kód jádra DNN. DNN využívá jak providery implementované v ASP.NET 2.0, tak providery vyvinuté speciálně pro DNN. Nejdůležitějším providerem v DNN je Data provider. Ten slouží pro získávání dat z databáze. Původně DNN podporoval pouze MS SQL Server. Uživatelům DNN podpora pouze tohoto řešení nestačila, a proto byl vytvořeno Data Provider API, které podporuje všechny dnes běžné databáze. Standardně je nastavena podpora pro MS SQL Serve Expres Edition, ale úpravou souboru web.config se dá nastavit podpora i pro ostatní databáze. Více viz [10].

Další providery použité v DNN jsou například:

- Scheduling Provider
- Logging Provider
- HTML Editor Provider
- Search Provider
- Friendly URL Provider atd.

DNN využívá vícevrstvou architekturu. To znamená, že vzhled systému, funkčnost systému, přístup k datům systému a samotná data jsou vytvořeny jako samostatné celky, které mezi sebou spolupracují. Vrstvy v DNN jsou umístěny na dvou různých místech. Prezentační vrstva, aplikační vrstva a vrstva pro přístup k datům jsou umístěny na webovém serveru (viz obrázek 3-1) a datová vrstva je umístěna na databázovém serveru. Více viz [10]



Obrázek 3-1: Architektura DNN *zdroj [12]*

3.4.1 Popis jednotlivých vrstev systému DNN

Prezentační vrstva (Presentation Layer)

Prezentační vrstva poskytuje uživatelské rozhraní. Tedy takové rozhraní, které zobrazuje návštěvníků stránek jejich obsah. Tato vrstva se skládá z následujících elementů:

- **Webové formuláře –** Nejdůležitější webový formulář je Default.aspx. Tento formulář je vstupní bod do portálu. Je nepostradatelný pro nahrání ostatních elementů prezentační vrstvy.
- **Skiny –** Pomocí skinů se dá měnit vzhled celé stránky (portálu). Webový formulář Default.aspx nahrává skin pro stránku podle nastavení, které je pro každou stránku portálu unikátní.
- **Kontejnery –** Kontejnery slouží jako obal pro moduly. Pomocí nich se dá měnit vzhled modulů nezávisle na zvoleném skinu a bez nutnosti provádět úpravu skinu. Webový formulář Default.aspx nahrává kontejnery pro moduly podle nastavení, které je pro každý modul unikátní.
- **Uživatelské ovládací prvky pro moduly –** Každý modul musí mít alespoň jeden uživatelský ovládací prvek. Tyto ovládací prvky nahrává webový formulář Default.aspx a jsou přitom začleněny do skinů a kontejnerů.

- **JavaScripty –** Několik JavaScriptů je použito přímo uvnitř DNN (například SolPartMenu). Další JavaScripty mohou být uvnitř modulů.

Při navštívení portálu DNN se spustí webový formulář Default.aspx. Kód stránky (code behind) nahraje vybraný skin. Poté jsou do paměti nahrány moduly dostupné pro aktuální stránku. Každý modul má přiřazený kontejner, do kterých je poté modul vložen. Kontejner může být přiřazený pouze k jednomu modulu nebo ke všem modulům na jedné stránce nebo ke všem modulům na jednom portálu. Po vložení modulů do kontejnerů jsou kontejnery přidány do tzv. tabule (panel) skinů. Poté jsou ještě aplikovány CSS soubory pro jednotlivé moduly.

Aplikační vrstva (Business Logic Layer)

V této vrstvě je uložen kód, který určuje, co a jak bude DNN dělat. Aplikační vrstva poskytuje služby jak pro jádro DNN, tak pro vestavěné moduly i pro moduly třetích stran. Tyto služby obsahují například:

- Lokalizaci
- Caching
- Personalizaci
- Vyhledávání
- Bezpečnost a jiné

V této vrstvě jsou uloženy objekty (na obrázku 3-1 CBO), které pracují s daty získanými v databázi a s daty, které budou do databáze uloženy. Dále jsou v ní uloženy funkce, které slouží k naplňování objektů daty podle požadavků programátora.

Vrstva pro přístup k datům (Data Access Layer)

Vrstva pro přístup k datům poskytuje služby pro aplikační vrstvu. Tato vrstva zajišťuje tok dat z a do databáze a zpřístupňuje je aplikační vrstvě.

Tato vrstva používá pro přístup k datům Provider Model. Proto je v ní množství různých data providerů. Jsou v ní uloženy jak provideři pro jádro DNN, tak provideři pro jednotlivé moduly. Obvykle pro každý modul speciální data provider. Jak je z obrázku 3-1 vidět, v této vrstvě se pomocí rozhraní IDataReader naplňují daty CBO objekty, se kterými pracuje aplikační vrstva. Také se zde berou data z CBO objektů pro uložení do databáze..

Datová vrstva (Data Layer)

Datová vrstva poskytuje data pro vrstvu pro přístup k datům. Data uložená v datové vrstvě musí být uložena ve formátu, který potřebují Data Provideři ve vrstvě pro přístup k datům. Pokud by v tomto formátu nebyla, tak by s nimi nebylo možné pracovat. V datové vrstvě jsou také uloženy instalační

skripty. Tyto skripty slouží pro vytvoření databázových tabulek, uložených procedur a vložení dat do databázových tabulek. Skripty jsou spouštěny při instalaci a jsou spouštěny pouze jednou. Více viz [10].

4 Specifikace požadavků na redakční systém

Redakční systém by měl uživateli umožnit, co nejlepším způsobem vytvářet, měnit a publikovat co nejlépe informace na internetu. Systém pro správu obsahu DNN není příliš vhodné bez použití modulů třetích stran, používat jako redakční systém. Obsahuje pouze moduly, které se hodí pouze pro prezentace, které nejsou příliš rozsáhlé a které není potřeba často upravovat. Pro každý článek by musel mít svůj vlastní modul. Také by bylo složité dodržovat stále stejný formát, pro všechny články, protože neexistuje žádná šablona, podle které by mohly být vytvářeny. Ani pro návštěvníky stránek není toto řešení ideální. Všechny články by byly zobrazeny pod sebou na jedné stránce. Nebylo by možné použít stránkování, ale pouze vytvořit pro další články novou stránku. Také by si návštěvník stránek nemohl filtrovat články podle kategorií. Opět by se musela pro každou kategorii vytvořit zvlášť stránka a na tu by se ukládaly všechny články, které do této kategorie spadají.

Podobné potíže by nastaly i pro práci s aktualitami. Nešlo by nastavit kolik aktualit se má na stránce zobrazit a dále by chyběla šablona, podle které by se aktuality vytvářely.

Aby šel DNN použít jako plnohodnotný redakční systém, je potřeba jej rozšířit alespoň o tři moduly. Byly by to modul *Články*, který by sloužil pro zobrazování a správu článků, modul *Kategorie*, který by sloužil pro zobrazování a správu kategorií a pomocí kterého by se daly filtrovat články a modul *Aktuality*, který by sloužil pro zobrazování a správu aktualit.

DNN také neumožňuje měnit obsah stránek podle zvoleného jazyka. Mění se pouze menu stránek. Proto je potřeba přidat ještě modul *pro vícejazyčnou podporu*, který bude tuto možnost realizovat.

4.1 Modul články

Prvním modulem v mé práci je modul *Články*. Tento modul je z pohledu prezentace nejdůležitějším modulem z celé diplomové práce. Slouží pro prezentaci a správu článků a využití najde hlavně při prezentaci malé firmy nebo pro vedení jednoduchého internetového časopisu. Pro větší časopisy a firmy by nemusel stačit, protože ty vyžadují systémy vyrobené na míru. Modul *Články* obsahuje články stejné struktury jako například článek v internetovém (nejenom) časopise či novinách. Může obsahovat jak obrázky tak text.

Proč modul *Články*

DNN žádný podobný modul v základní instalaci neobsahuje. Kdyby chtěl někdo v DNN psát články, musel by pro každý článek vložit modul *(Text/HTML)* a v něm psát článek. Pro návštěvníky budou stránky přehlednější a administrátorům či správcům stránek může modul *Články* velmi usnadnit práci. Vše bude na jednom místě a navíc nabídne šablonu pro tvorbu článků. Tím se zabrání problémům s jednotným formátem článků, který by jinak mohl nastat. Při použití tohoto modulu jsou všechny články zobrazeny pouze v jednom modulu (pokud není těchto modulů na stránce více). Správa i vkládání nových článků je mnohem jednodušší, protože se na každý článek nemusí vytvářet nový modul. U tohoto řešení lze také použít stránkování a zjednodušit tím stavbu webu.

4.2 Modul Kategorie

Mít všechny články pohromadě je velmi nepřehledné a to jak pro návštěvníka stránek tak pro její správce. Výhodnější je členit články do různých kategorií podle jejich zaměření. Kategorie nebudou pevně dané, ale bude si je vytvářet správce, protože každý bude od systému očekávat něco jiného a pevně stanovené kategorie se nikdy netrefí do potřeb všech uživatelů. Jediné pevně dané kategorie jsou kategorie článků pro registrované uživatele a zobrazení všech článků. Každá kategorie může mít libovolný počet podkategorií.

Proč modul *kategorie*

Bez tohoto modulu by větší prezentace firmy na internetu mohla být nepřehledná a odradit případné zákazníky. Pokud by tento modul nebyl k dispozici, tak by se zákazník musel probírat velkým množstvím článků různých typů a zaměření. Když tento modul implementován bude, může si už vybírat pouze z článků typu, který ho zajímá. Navíc to ušetří práci i správci webu (například při hledání určitého článku).

4.3 Modul Aktuality

Ve svém projektu chápu aktualitu jako krátkou důležitou zprávu, kterou je potřeba na webu zobrazit. Modul *aktuality* bude sloužit k zobrazování a správě těchto aktualit.

Proč Modul *aktuality*

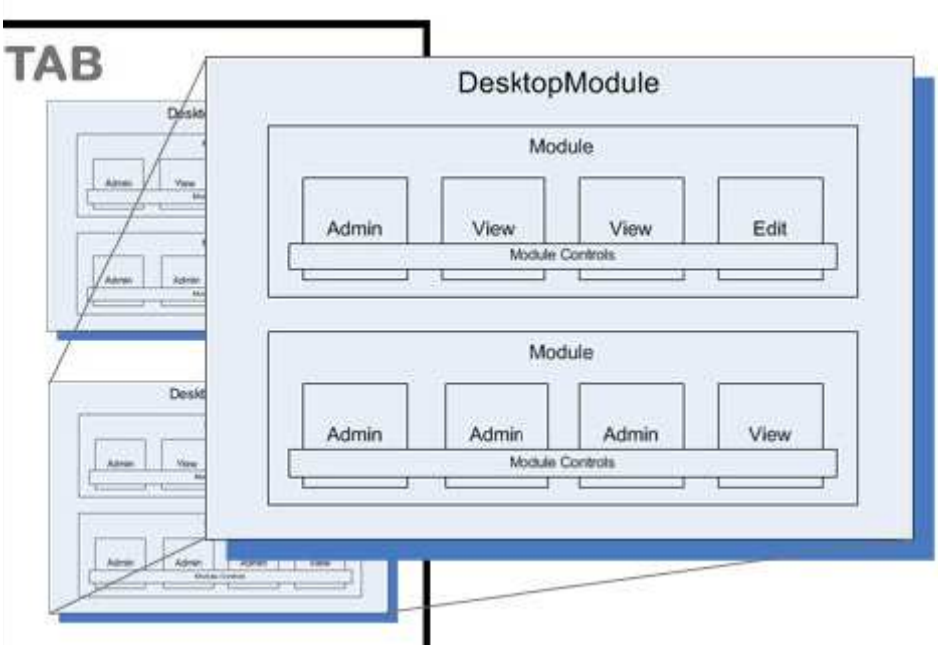
Když je potřeba zobrazit informaci, která není rozsahově tak obsáhlá, aby se vyplatilo pro ni vytvářet nový článek nebo pouze upozornit na zajímavý článek, který není na úvodní stránce, a který by proto mohl snadno zapadnout. Administrátor nebo editor webu může o zajímavé informaci napsat aktualitu a tím na tuto informaci upoutat pozornost návštěvníků a poté teprve napsat obsáhlý článek, který se bude informaci věnovat podrobněji. Přesně tuto problematiku bude řešit modul *aktuality*.

5 Návrhy modulů do DNN

Moduly slouží v DNN jako nástroje pro správu a zobrazování samotného obsahu stránek. Po instalaci je DNN chudým a jednoduchým systémem pro správu obsahu a obsahuje pouze několik typů modulů. Dá se ale pomocí modulů třetích stran téměř libovolně rozšiřovat. Dostupných modulů je dostatek a dají se snadno instalovat.

5.1 Struktura modulu v DNN

Nejvyšší úrovní pro každý modul je stránka (na obrázku 5-1 je stránka pojmenovaná jako TAB) a pouze na stránky mohou být moduly vkládány.



Obrázek 5-1: Struktura modulu *zdroj [12]*

Každý modul se skládá ze tří částí. Je to část administrátorská, prezentační a editační. Administrátorská část slouží k nastavení vlastností modulu. Nastavuje se v ní, kterým typům uživatelů bude modul zobrazován a které typy uživatelů bude vytvářet a spravovat obsah modulu. Dále se v této části dá měnit vzhled modulu pomocí přiřazení kontejneru nebo barvy pozadí. Může se v ní nastavit také od jakého datumu se má modul na stránce zobrazovat a od jakého datumu bude ze stránek stažen. Tato část obsahuje také speciální nastavení určené pouze pro daný typ modulu. Prezentační část modulu slouží k zobrazování obsahu modulu a editační část slouží k vytváření a upravování obsahu modulu.

Pro správnou funkci modulů je potřeba navrhnout alespoň prezentační a editační část modulu. U administrátorské části modulu, se navrhuje pouze nastavení, které je specifické pro daný modul. A není nutné tyto nastavení pro správnou funkci modulu implementovat a i bez nich bude modul bez potíží fungovat. Pouze se sníží možnosti úprav jeho vlastností.

5.2 Jednotlivé návrhy modulů

Výsledkem této diplomové práce mají být rozšíření, které umožní používat DNN jako redakční systém pro prezentaci malých a středních firem. Proto jsem navrhl a naprogramoval tyto moduly: *Články*, *Kategorie*, *Aktuality* a modul *Podpory vícejazyčnosti*.

5.2.1 Modul Články

Modul *články* slouží pro správu a prezentaci článků.

Struktura článků v modulu *Články*

Článek má několik součástí (jelikož je editační část modulu vytvořena v anglickém jazyce, tak i součásti článku uvádím v anglickém jazyce):

- **Nadpis** (title) – Nadpis článku.
- **Kategotie** (category) – Výběr rubriky, do které bude článek patřit.
- **Anotace** (Anotation) – Jde o krátký text, který má přiblížit obsah článku. Může obsahovat i obrázky.
- **Obsah** (content) – Zde se jedná o samotný obsah článku. Může obsahovat i obrázky.
- **Datum vytvoření** (date create) – Datum vytvoření článku. Je vytvořeno automaticky.
- **Datum vydání** (date release) – Datum kdy, bude článek zobrazen v modulu. Před tímto datem bude článek pro uživatele bez oprávnění neviditelný. Pokud datum uživatel nevyplní, bude doplněn automaticky o aktuální datum.
- **Čas vydání** (time release) – Čas vydání článku. V tento čas se článek zobrazí v modulu (pokud již bylo dosaženo datumu vydání). Před tímto časem bude článek pro uživatele bez oprávnění neviditelný. Pokud uživatel čas nevyplní, bude automaticky doplněn o aktuální čas.
- **Datum vypršení platnosti** (date expire) – Po dosažení tohoto datumu přestane být článek zobrazován v modulu. Pokud datum uživatel nezadá, bude datum automaticky doplněn o maximální hodnotu, kterou lze uložit do databáze.
- **Čas vypršení platnosti** (time expire) – Po dosažení tohoto času přestane být článek zobrazován (pokud již bylo dosaženo datumu vypršení). Pokud čas uživatel nezadá, bude čas automaticky doplněn na čas maximální čas (23:59:59).
- **Auhor** (author) – Jméno autora. Bude vyplněno automaticky z údajů z uživatelského účtu.

- **Vydat** (publish) – Je to vlastnost článku, která určuje, jestli byl článek vydán. Nevydané články vidí jen přihlášení uživatelé s příslušným oprávněním. Je to důležitá vlastnost, když potřebuje uživatel uložit ještě nedokončený článek nebo když je potřeba článek rychle stáhnout například z důvodu chyby v článku. Tímto lze snadno přestat zobrazovat článek a opravit chybu až později a není nutné článek mazat.
- **Viditelný pro registrované uživatele** (visible for registered users) – Článek s touto vlastností vidí pouze přihlášení uživatelé. Výhodné když mají být některé články viditelné například jen pro zákazníky firmy.
- **Na první stránce** (on first page) – Článek s touto vlastností bude vždy zobrazen na první stránce modulu ať je nastavené jakýkoliv druh řazení článku. Výhodné pokud je potřeba, aby některý článek s důležitou informací byl stále na první stránce i po vydání dalších článků
- **Jazyk** (language) – Jazyk, ve kterém je článek napsán (tato možnost je u článku jen pokud je nainstalován i modul ***vícejazyčné podpory***)
- **Počet zobrazení** (nuber of views) – Počet zobrazení článku.

Do článků jdou vkládat i obrázky v libovolném množství, ale vkládají se přímo do textu anotace nebo obsahu.

5.2.1.1 Struktura modulu *Články*

Modul *články* se dělí na tři části. Prezentační a administrátorskou a editační. Pro administrátorskou a editační je nutné je být přihlášen do systému a mít potřebné oprávnění pro tento modul. Prezentační část je přístupná i nepřihlášeným uživatelům.

Prezentační část

Prezentační část se dělí na dvě části. Tyto části se od sebe liší způsobem jakým zobrazují články. Tyto části by se spíš daly nazvat jako pohledy na články.

Pohled první

Zobrazuje náhledy článků (zobrazí se jen nadpis a anotace) v modulu. Náhledy jsou vypisovány pod sebe a počet těchto náhledů nastavuje administrátor. Pokud jejich počet není určen, tak se zobrazí pět článků (přednastavená hodnota) Modul obsahuje stránkování, takže náhledy článků, které se nevešly na první stránku modulu, budou zobrazeny na zbylých stránkách modulu. Zobrazeny budou jen ty články, které již byly vydány a kterým ještě nevypršela platnost. Nejdříve jsou vypsány ty články u nichž administrátor určil, že budou zobrazovány na první stránce. Teprve poté budou vypsány i ostatní články. Články nejsou vypisovány náhodně, ale jsou řazeny buď podle datumu vytvoření, datumu vydání nebo podle počtu zobrazení. Podle čeho budou zobrazeny si určí administrátor webu. Články si může návštěvník stránek procházet všechny nebo si nechat zobrazit články jen jedné

kategorie (podmínkou pro správné fungování modulu *články*, je mít nainstalován i modul *kategorie* a v něm mít vytvořenou alespoň jednu kategorii, kterou je možno přiřadit článku). Modul podporuje i tvorbu článků, které se zobrazí jen registrovaným uživatelům (pro jejich zobrazení je nutné být v systému přihlášen). Tyto články jsou standardně dostupné mezi ostatními články, ale lze si vypsat pouze tyto články kliknutím na kategorii tomu určenou. Uživatelé s příslušnými právy si mohou zobrazit také nepublikované články (při psaní článku se nezaškrtla možnost publikovat článek) nebo články, které ještě nevyšly či již prošla doba jejich platnosti. Tyto články pak mohou dále editovat či je rovnou smazat. Kliknutím na nadpis článku se zobrazí druhý pohled. Pokud má uživatel dostatečná oprávnění, tak se může z prvního pohledu přejít do administrátorské části. V horní části modulu je vypsána aktuální kategorie, ve které se právě návštěvník stránek nachází, i se všemi rodičovskými kategoriemi. Všechny tyto kategorie jsou vypsány jako odkazy, takže je možné se pomocí nich přesouvat do daných kategorií.

Pohled druhý

Zobrazuje jeden článek s těmito atributy: nadpis, anotace, obsah, autor a počet zobrazení článku. Jde vlastně o zobrazení celého článku se všemi atributy, které mají být návštěvníkovi stránek zobrazeny. Z tohoto pohledu se po přečtení článku uživatel vrátí zpět do prvního pohledu. U tohoto pohledu na článek jsem narazil na jisté omezení systému DNN. Při zobrazení článku se přes celou stránku zobrazí pouze tento modul. Ostatní moduly budou tím pádem nedostupné. To je samozřejmě omezující a není to příliš uživatelsky příjemné. Pokud chce uživatel změnit kategorii nebo provést jinou akci, musí se vrátit zpět do prvního pohledu, kde již budou vypsány všechny dostupné moduly.

Editační část

Do této části se může uživatel s dostatečným oprávněním dostat dvěma způsoby. Buď kliknutím na možnost tvorby nového článku nebo kliknutím na ikonku tužky vpravo od nadpisu článku. V této části pak může uživatel provádět tyto akce:

Vytvoření nového článku – po zvolení této možnosti může uživatel vytvořit nový článek. Povinně musí uživatel vyplnit nadpis. Dále pak určí viditelnost článku. Ostatní údaje a součásti již nejsou tak důležité a vyplněny být nemusí a záleží pouze na uživateli, zda se rozhodne je využít. Článek rozhodně bude vypadat lépe a profesionálněji při vyplnění všech údajů a součástí.

Smazání článku – uživatel má možnost smazat jakýkoliv vybraný článek. Před smazáním každého článku bude uživatel ještě dotázán, zda chce článek opravdu smazat

Úprava existujícího článku – každý článek může být po vytvoření upravován a uživatel může měnit všechny jeho vlastnosti a součásti včetně jeho viditelnosti.

Po zvolení možností vytvoření nového článku nebo úpravy existujícího se uživateli otevře příslušný formulář, který bude následně vyplňovat nebo upravovat jeho obsah. Pokud zůstane jedna či

více povinných položek nevyplněna nebo bude vyplněna špatně, bude uživatel vyzván k opravě. Údaje, které již byly vyplněny, zůstanou vyplněny a nebude potřeba je zadávat opakovaně. Jméno autora bude doplněno automaticky, podle uživatelského účtu uživatele, který článek vytvořil. Dále zde bude zobrazeno datum poslední úpravy a jméno uživatele, který tuto úpravu provedl. Při tvorbě anotace a obsahu článku, může uživatel využít DNN WYSIWYG editor a upravit pomocí něj text. Pomocí něj může uživatel do článku také vkládat obrázky. V tomto WYSIWYG editoru se uživatel může podívat, jaký HTML kód bude pro text anotace nebo obsahu WYSIWYG editorem generován. Pokud jej nebude chtít použít, může uživatel přepnout WYSIWYG editor na klasický textbox.

Administrátorská část

V této části může uživatel, který má dostatečná oprávnění, provádět tyto akce:

Nastavit počet článků, které se budou zobrazovat na první stránce modulu. Modul nedovolí, aby měli atribut na první stránce více článků než kolik jich je na první straně.

Nastavit, podle jakého atributu budou články řazeny při vypisování. Články mohou být řazeny podle datumu vytvoření, datumu vydání nebo počtu zobrazení.

Pokud bude na stránce více modulů *kategorie*, tak si uživatel může vybrat, z kterého modulu si bude kategorie vybírat.

Dále zde může nastavit různé atributy pro tento modul, které jsou stejné pro všechny moduly (například barvu pozadí nebo zarovnání modulu atd.)

5.2.2 Modul kategorie

Modul *kategorie* slouží pro správu a zobrazení kategorií článků. Pomocí kategorií jdou články filtrovat.

Struktura kategorie v modulu *kategorie*

Kategorie má několik součástí (jelikož je editační část modulu vytvořena v anglickém jazyce, tak i součásti článku uvádím v anglickém jazyce):

- **Jméno** (name) **–** Jméno, pod kterým bude kategorie zobrazena.
- **Pořadí** (rank) – určuje na jakém místě bude tato rubrika zobrazena. Při vložení nové kategorie se automaticky přidělí číslo odpovídající poslednímu místu v seznamu. Pokud ho bude chtít uživatel změnit může tak udělat ještě před uložením kategorie.
- **Jazyk** (language) – Jazyk, ve kterém je kategorie vytvořena (pouze pokud bude nainstalován i modul *vícejazyčné podpory*)
- **Level** – Určuje, jak hluboko je tato kategorie zanořenořena. Nastaví si automaticky systém podle zvolené rodičovské kategorie. Pokud se level kategorie rovná jedné, určuje to, že kategorie není vůbec zanořena.

- **Rodič** (parent) – Určuje, která kategorie je pro tuto kategorii rodičovská (tedy o jeden level nad touto kategorií)

5.2.2.1 Struktura modulu *kategorie*

Modul *kategorie* se dělí na tři části. Prezentační a administrátorskou a editační, pro administrátorskou a editační část je nutné je být přihlášen do systému a mít potřebné oprávnění pro tento modul. Prezentační část je přístupná i nepřihlášeným uživatelům.

Prezentační část

Všem uživatelům bude zobrazena lišta podle volby administrátora na pravém nebo levém okraji stránky (může být umístěna kdekoliv, ale předpokládám toto umístění), ze které si vybere návštěvník stránek požadovanou kategorii. Po jejím zvolení mu budou zobrazeny jen články, které do této kategorie patří. Nad seznamem těchto kategorií bude odkaz na zobrazení článků všech kategorií. Vybraná kategorie bude červeně vyznačena. Pokud obsahuje kategorie nějaké podkategorie, tak ty budou vypsány pod ní, ale budou pro přehlednost více odsazeny od kraje modulu. Podkategorie se vypisují maximálně do třetího levelu zanoření. Počet levelů zanoření není omezen a může být libovolně velký, ale v modulu se zobrazí vždy maximálně tři levely zanoření. Vybrané kategorie se zobrazují také v modulu *články*. Registrovaní a přihlášení uživatelé, vidí také kategorii, která slouží pro zobrazení článků určených pro registrované uživatele. Uživatelům s dostatečnými právy pro tento modul se zobrazí také kategorie pro nepublikované články, pro články, které ještě nebyly vydány (datum jejich vydání ještě nenastalo) a články jejichž platnost již vypršela.

Editační část

Do této části se může uživatel s dostatečným oprávněním dostat dvěma způsoby. Buď kliknutím na možnost tvorby nové kategorie nebo kliknutím na ikonku tužky vpravo od názvu kategorie. V této části pak může uživatel provádět tyto akce:

Vytvoření nové kategorie – po zvolení této možnosti může uživatel vytvořit novou kategorii. Je nutné vyplnit pouze jméno kategorie a vybrat rodičovskou kategorii. Pokud uživatel chce, aby kategorie byla v prvním levelu, tak jako rodičovskou kategorie musí vybrat root. Poté se kategorie přidá do seznamu kategorií. V tomto seznamu jsou kategorie podle vybrané rodičovské kategorie. Poté může uživatel určit pořadí všech kategorií.

Úprava existujícího kategorie – každá kategorie může být upravována. Můžeme u ní měnit název, pořadí při zobrazení i rodičovskou kategorii.

Smazání existující kategorie – uživatel má možnost smazat jakoukoliv kategorii. Mazání kategorií je složitější než mazání článků. Smazáním článku se pro ostatní články nic nemění. Ale problém by nastal, pokud by se měla smazat kategorie, která obsahuje nějaké podkategorie, nebo pokud by kategorie obsahovala nějaké články. Když nastane tato situace, je uživateli nahlášena. On

poté dostane na výběr, co se po smazání kategorie stane s články a s podkategoriemi. Uživatel si vybere zda chce podkategorie smazat nebo zda je chce přiřadit jiné kategorii. Stejný výběr má i pro články (pro články může být volba jiná než pro podkategorie). Před smazáním kategorie je ještě vyzván k potvrzení své volby.

Administrační část

V této části může uživatel nastavovat pouze obecná nastavení stejná jako u všech modulů. Žádná speciální nastavení pro tento modul nejsou.

5.2.3 Modul aktuality

Modul *aktuality* slouží pro správu a prezentaci aktualit.

Struktura aktuality v modulu *aktuality*

Aktualita má několik součástí (jelikož je editační část modulu vytvořena v anglickém jazyce, tak i součásti článku uvádím v anglickém jazyce):

- **Nadpis** (title) **–** Nadpis aktuality
- **Obsah** (content) **–** Jde o samotný obsah aktuality. Může obsahovat i obrázky, ale ty by měly být vkládány v rozumné míře a velikosti, aby příliš nenarušily vzhled modulu.
- **Vydat** (publish) **–** Je to vlastnost aktuality, která určuje, jestli byla aktualita vydána. Funguje stejně, jako vlastnost Publish u článků.
- **Datum vydání** (date release) – Datum kdy, bude článek zobrazen v modulu. Před tímto datem bude článek pro uživatele bez oprávnění neviditelný. Funguje stejně, jako tato vlastnost u článků.
- **Čas vydání** (time release) – Čas vydání článku. V tento čas se článek zobrazí v modulu (pokud již bylo dosaženo datumu vydání). Funguje stejně, jako tato vlastnost u článků**.**
- **Datum vypršení platnosti** (date expire) – Po dosažení tohoto datumu přestane být článek zobrazován v modulu. Funguje stejně, jako tato vlastnost u článků.
- **Čas vypršení platnosti** (time expire) – Po dosažení tohoto času přestane být článek zobrazován (pokud již bylo dosaženo datumu vypršení). Funguje stejně, jako tato vlastnost u článků.
- **Jazyk** (language) – Jazyk, ve kterém je aktualita vytvořena (pouze pokud bude nainstalován i modul *vícejazyčné podpory*)

5.2.3.1 Struktura modulu *aktuality*

Modul *aktuality* se dělí na tři části. Prezentační a administrátorskou a editační, pro administrátorskou a editační část je nutné je být přihlášen do systému a mít potřebné oprávnění pro tento modul. Prezentační část je přístupná i nepřihlášeným uživatelům.

Prezentační část

Prezentační část modulu *aktuality* by se dala podle způsobu zobrazení aktualit rozdělit na dvě části respektive na dva způsoby zobrazení.

Při první zobrazení se aktuality zobrazují v boční liště napravo nebo nalevo podle (pozici si může určit administrátor webu. Modul může být samozřejmě umístěn kdekoliv na stránce, ale jiné umístění nepředpokládám) přímo na stránce, kde je umístěn modul. Administrátor také určuje, kolik aktualit se v této liště bude zobrazovat. Pokud jejich počet neurčí, budou se zobrazovat dvě aktuality, což je přednastavená hodnota. Aktuality jsou při zobrazování řazeny vždy podle datumu vydání. Nejsou zde další možnosti jako u modulu *články*, protože jde o to zobrazovat na prvních místech vždy nejnovější aktuality. V dolní části lišty je odkaz (pokud má uživatel dostatečná oprávnění pro tento modul, tak více odkazů), kterým se uživatel dostane do druhého způsobu zobrazení.

Při druhém zobrazení se už vypisují všechny dostupné aktuality (tedy ty aktuality, které jsou již vydané a kterým nevypršela doba platnosti). Toto zobrazení již není na stránce s ostatními moduly, ale je již na samostatné stránce. Z toho vyplývají nevýhody popsané již výše u modulu *články*. V tomto zobrazení je opět použité stránkování, takže administrátor si může zvolit, kolik aktualit bude na jedné stránce vypsáno. Aktuality jsou zde opět řazeny podle datumu vydání. Pokud má uživatel dostatečná práva pro tento modul, může si v tomto zobrazení nechat vypsat všechny nepublikované aktuality nebo aktuality, které ještě nebyly vydány nebo aktuality, kterým již vypršela doba platnosti.

Editační část

Do editační části se uživatel s dostatečným oprávněním dostane stejně jako u ostatních modulů a může provádět tyto akce:

Vytvoření nové aktuality – po zvolení této možnosti může uživatel vytvořit novou kategorii. Je nutné zadat pouze nadpis. Ostatní položky jsou nepovinné.

Smazání existující aktuality – uživatel má možnost smazat jakoukoliv aktualitu. Před smazáním musí potvrdit, zda chce aktualitu opravdu smazat.

Úprava existující aktuality – každá aktualita může být upravena. Upravovat lze všechny atributy.

Po zvolení možností vytvoření nové aktuality nebo úpravy existující se uživateli otevře příslušný formulář, který bude následně vyplňovat nebo upravovat jeho obsah. Pokud zůstane jedna čí více povinných položek nevyplněna nebo bude vyplněna špatně, bude uživatel vyzván k opravě. Údaje, které již byly vyplněny, zůstanou vyplněny a nebude potřeba je zadávat opakovaně. Uživatel zde může opět využít pro úpravy textu obsahu aktuality WYSIWYG editor.

Administrační část

V této části může uživatel s dostatečným oprávněním provádět tyto akce:

Nastavit kolik aktualit se má zobrazit v liště na stránce s ostatními moduly. Přednastaveno je, aby se zobrazovaly dvě aktuality.

Nastavit kolik aktualit se má zobrazit na stránce se všemi aktualitami. Pokud bude aktualit více, budou stránkovány.

Dále už jdou nastavovat pouze obecné vlastnosti, které jsou stejné pro všechny moduly.

5.2.4 Modul vícejazyčné podpory

Modul *vícejazyčné podpory* slouží pro správu jazyků a umožňuje zobrazovat články, kategorie a aktuality ve více než jednom jazyce.

Struktura jazyka v modulu *vícejazyčné podpory*

Jazyk má několik součástí (jelikož je editační část modulu vytvořena v anglickém jazyce, tak i součásti článku uvádím v anglickém jazyce):

- **Jméno** (name) **–** Jde o název jazyka.
- **Pořadí** (rank) – Určuje v jakém pořadí bude jazyk vypsán
- **Viditelnost** (visibility) – Určuje, zda bude jazyk zobrazován. U jazyka, který tuto vlastnost nebude mít zaškrtnutou nebudou zobrazovány ani články, kategorie a aktuality napsané v tomto jazyce.
- **Přednastavený jazyk** (default language) – v tomto jazyce se vždy zobrazí stránka při prvním spuštění.
- **Porovnávání s ostatními jazyky** (compare with others languages) – po zaškrtnutí této možnosti se zobrazí seznam dostupných jazyků. Z nich může uživatel vybrat, se kterými jazyky bude vybraný jazyk porovnáván.

5.2.4.1 Struktura modulu *vícejazyčné podpory*

Modul *podpory více jazyčnosti* se dělí na tři části. Prezentační a administrátorskou a editační, pro administrátorskou a editační část je nutné být přihlášen do systému a mít potřebné oprávnění pro tento modul. Prezentační část je přístupná i nepřihlášeným uživatelům.

Prezentační část

Prezentační část modulu *vícejazyčné podpory* je poměrně jednoduchá. V boční liště (přesné umístění určí administrátor) jsou pod sebou vypsány všechny vytvořené jazyky. Boční lištu jsem vybral, protože mi z možností, které DNN nabízí přišla nejvhodnější. Původně jsem počítal, že jazyky budou v horní liště, ale tím se vytvářely pouze problémy. Hlavní z nich by byl, že tato lišta leží přímo pod hlavním menu, a proto by mohlo často docházet k ukliknutí, kdy místo kliknutí do menu by uživatel změnil jazyk. Další důvod byl desigenový. Jazyky vypsané vedle sebe na mě nepůsobily moc

esteticky. Navíc by mohl nastat problém, pokud by jich bylo víc, než by se na lištu vešlo. Pak by se na ní vytvořil další řádek s jazyky, což by ještě zvýšilo nepřehlednost tohoto řešení.

Změna jazyka se provede jednoduše pouze kliknutím na název jazyka, do kterého chceme zobrazení webu přepnout.

Editační část

Editační část modulu *vícejazyčné podpory* obsahuje ještě čtyři další části, pomocí kterých se vytváří a upravují jednotlivé jazyky a pomocí kterých se překládají jednotlivé jazyky.

První část *tvorba a úprava jazyků*

První část se zabývá tvorbou a úpravou jednotlivých jazyků. Uživatel s dostatečnými oprávněními v ní může provádět tyto akce:

Vytvoření nového jazyka – po zvolení této možnosti, může uživatel vytvořit nový jazyk. Nutné je vyplnit pouze název jazyka.

Úprava existujícího jazyka – každý jazyk může být upravován. Upravovat lze všechny jeho atributy.

Smazání existujícího jazyka – každý jazyk může být smazán. Pokud jsou v jazyce, který má být smazán vytvořeny nějaké články, rubriky nebo kategorie. Je to uživateli oznámeno a může si zvolit co se s nimi stane po smazání jazyka, ve kterém byly vytvořeny. Volba je pro články, kategorie i aktuality společná. Může si vybrat, zda chce všechno přiřadit k novému jazyku a určit, který jazyk to má být nebo zda má být vše smazáno společně s jazykem. Aby si uživatel omylem všechno nesmazal, tak je přednastaveno, aby bylo vše přiřazeno jinému jazyku. Před samotným smazáním jazyka musí ještě uživatel potvrdit, že chce jazyk opravdu smazat.

Druhá část *překládání konstantního textu v jednotlivých modulech*

Do této části se uživatel dostane kliknutím na příslušný odkaz v první části. Může v ní překládat texty, které jsou v pro každý modul pevně stanoveny. Jedná se u modulu *Kategorie* o text odkazu pro zobrazení všech článků bez rozdílů kategorií a o text odkazu pro zobrazení článků určených pouze pro registrované uživatele. U modulu *Aktuality* se jedná o text odkazu pro zobrazení všech aktuality.

Nejdříve si uživatel vybere, pro který modul chce překlad provádět. Podle jeho výběru jsou mu vypsány odpovídající položky a zobrazen původní text. Pokud již překlad pro vybraný modul prováděl, tak je vypsán i překlad, který může upravit.

Třetí část *výběr článků, aktualit či kategorií k překladu*

Do této části se uživatel dostane kliknutím na příslušný odkaz v první části. Už tam si musí vybrat zda bude chtít překládat články, kategorie nebo aktuality. Podle jeho volby se uživateli zobrazí příslušný formulář. Pokud je na stránce více modulů stejného typu (například několik modulů *Články*), musí si uživatel zvolit z kterého modulu chce překlad provádět. Na formulář se poté

uživateli vypíší příslušné články, aktuality nebo kategorie. Články a kategorie jsou vypsány pouze ty, které byly vydány a kterým ještě nevypršela doba platnosti. U kategorií jsou vypsány jejich jména a u článků a aktualit jsou vypsány jejich nadpisy. U článků si uživatel může nechat vypsat pouze články patřící do jedné z kategorií. Ze seznamu si vybere, co bude chtít přeložit a kliknutím na příslušný odkaz se dostane do čtvrté části. Pokud ze seznamu nic nevybral, tak se po kliknutí na odkaz nestane nic. Pokud je už všechno ve vybraném modulu přeloženo, zobrazí se uživateli oznámení, že již není co překládat.

Čtvrtá část *překlad obsahu modulů Články, Kategorie a Aktuality*

Podle volby, kterou uživatel provedl v třetí části, je mu otevřen příslušný formulář. Na něm jsou zobrazeny jednotlivé atributy článku, aktuality nebo kategorie. Každý z těchto atributů je nejdříve zobrazen v původním jazyce a pod ním je vyhrazeno místo, do kterého uživatel provede překlad. U kategorie si pak i u překladu může určit rodičovskou kategorii a pořadí kategorie. U článků a aktualit si uživatel může nastavit datum a čas vydání, datum a čas vypršení platnosti a zda má být článek nebo aktualita vydána. Dále lze u nich nastavit, zda je článek již kompletně přeložen nebo zda se ještě bude v překladu pokračovat. Pokud ještě článek nebo aktualita přeložena není, tak se k nim může uživatel kdykoliv vrátit a překlad dokončit. Když se k nim chce vrátit postupuje stejně jako když chce překládat od začátku. Nedokončené překlady budou dále zobrazovány v seznamech ve třetí části. Až dokončení překladu v ní přestanou být zobrazovány. U překladu kategorie není možnost uložit nedokončený překlad, protože tam se překládá pouze název, takže by to bylo zbytečné.

6 Implementace modulů

Implementace všech modulů je provedena v prostředí ASP.NET a programovacím jazyku C#. Jako databáze je použit MS SQL Server Express Edition.

Modul dodržuje stejnou architekturu jako samotný DNN. Proto je nutné naprogramovat všechny tyto vrstvy.

Prezentační vrstva

Pro tuto vrstvu se vytváří ascx soubory. V nich je uloženo, jak bude modul vypadat, kde budou rozmístěny ovládací prvky a kde se bude zobrazovat obsah modulu.

Aplikační vrstva

Pro tuto vrstvu se vytváří dva typy souborů. Prvním typem souborů jsou soubory Info, ve kterých se definují CBO objekty. Druhým typem souborů jsou soubory Controller, které slouží pro naplnění CBO objektů daty z databáze.

Vrstva pro přístup k datům

Pro tuto vrstvu se vytváří soubory, které obsahují Data Provideri, které odpovídají databázové struktuře vytvářeného modulu. Pomocí nich se přistupuje k datům uloženým v databázi.

Datová vrstva

Pro datovou vrstvu se vytváří instalační skripty, pomocí kterých se při instalaci modulu vytvoří odpovídající tabulky a pomocí kterých jsou vytvořeny uložené procedury, které budou pracovat s daty uloženými v těchto tabulkách.

V popisu implementace jednotlivých modulů nebudou psány moduly ve stejném pořadí jako v předchozích kapitolách diplomové práce, ale budou seřazeny podle toho v jakém pořadí jsem je vytvářel. U všech modulů bude popsána implementace aplikační a datové vrstvy. Aplikační vrstvy a vrstvy pro přístup k datům jsou implementovány velmi podobně pro všechny moduly, a proto jsou popsány pouze obecně. Popsány jsou nejzajímavější části implementace nebo ty části implementace, které mi činily největší potíže.

Ve všech modulech využívám komponentu DNN LabelControl. Oproti klasickému ASP.NET Labelu obsahuje kromě textu ještě ikonku s otazníkem, pod kterým se skrývá nápověda.

6.1 Implementace aplikační vrstvy a vrstvy pro přístup k datům (pro všechny moduly)

Naplňování CBO objektů daty se v projektu naplňují dvěma způsoby. První způsob se používá, pokud je potřeba naplnit pouze jeden objekt (z databáze se načítá například pouze jeden článek). V tomto případě se použije funkce `FillObject,` která tento objekt naplní. Využívá k tomu Data Provider uložený ve vrstvě pro přístup k datům a třídu IDataReader. Druhý způsob se používá, pokud je potřeba naplnit daty více objektů zároveň. V tomto případě je použita třída IDataReader, do které se pomocí Data Provideru načtou data z databáze. Pomocí cyklu, díky kterému se bude přistupovat k jednotlivím instancím třídy IDataReader, se poté naplní seznam CBO objektů.

K datům přistupuje vrstva pro přístup k datům pomocí třídy SqlHelper a funkcí určených k čtení a vkládání dat do databáze (`ExecuteNonQuery, ExecuteReader`). Důležitým parametrem těchto funkcí je název uložené procedury, která se má provést. Tyto vrstvy jsou uloženy v souborech DataProvider.cs, SQLDataProvider.cs, JménoModuluInfo.cs a JménoModuluController.cs.

6.2 Implementace modulu Kategorie

Modul *Kategorie* byl první modul, který jsem v diplomové práci naprogramoval. Bylo to také moje první větší setkání s ASP.NET.

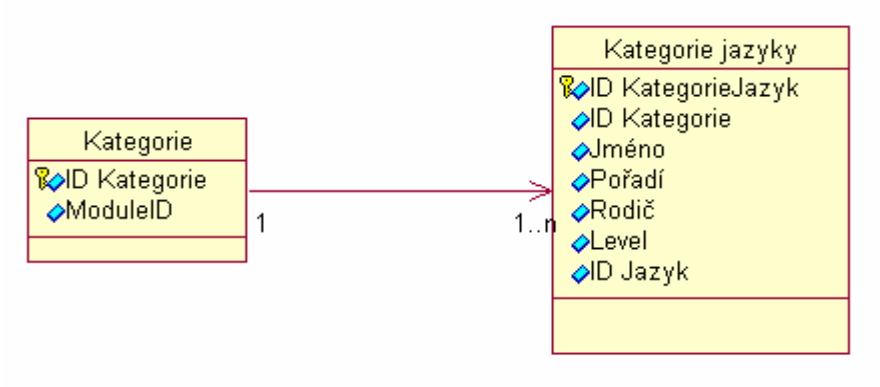
Prezentační vrstva

Kategorie jsou v prezentační části modulu vypisovány pomocí DataListu. Načítání dat do DataListu může probíhat až ve třech krocích v závislosti na tom, kolik levelů zanoření má být zobrazeno. Kategorie jsou načítány nejdříve do proměnné typu List. Jsou do ní načítány podle levelu zanoření. Nejdříve jsou tedy načteny kategorie s nejnižším levelem zanoření, po tom jsou na správné místo vloženy kategorie s vyšším levelem atd. Při druhém a třetím kroku jsou před názvy kategorií vloženy znaky mezer. Tím se zajistí odsazení kategorií s vyšším levelem zanoření. Každé kategorii je přiřazena URL adresa, do ve které je zapsáno ModuleID a ID kategorie. Podle ModuleID rozpozná modul *Články*, zda se jedná o kategorie z modulu, který mu přiřadil administrátor. ID kategorie určuje, které články budou vypsány (v souborech Viewpages.ascx a Viewpages.ascx.cs).

V editační část vychází z ListBoxu s kategoriemi. V ListBoxu jsou zobrazeny vždy jen kategorie se stejnou rodičovskou kategorií a stejným jazykem (pokud je nainstalován modul *vícejazyčné podpory*). DropDownListy s jazyky i kategoriemi obsahují atribut AutoPostBack a při změně rodičovské kategorie nebo jazyka jsou kategorie v ListBoxu změněny. Nová kategorie se vkládá do tohoto ListBoxu. Při ukládání nové kategorie nebo upravování existující se pracuje vždy se všemi kategoriemi z ListBoxu. Prochází se cyklem a pokud se jedná o novou kategorii, tak je zavolána funkce na uložení nové kategorie a pokud se jedná o existující kategorie a její pořadí v ListBoxu bylo změněno, je zavolána funkce pro upravení pořadí kategorie (v souborech Editpages.ascx a Editpages.ascx.cs).

Datová vrstva

Tabulky pro tento modul jsou vytvořeny tak, aby byla možnost přidání podpory více jazyků. Proto jsou pro tento modul vytvořeny dvě tabulky. V první tabulce je uloženo pouze ID kategorie a ModuleID. ModuleID je v tabulce uloženo, aby se v modulu zobrazovaly vždy jen záznamy vytvořené v tomto modulu. V druhé tabulce jsou uloženy všechny potřebné údaje kategorií. Tabulky jsou mezi sebou spojeny přes ID kategorie. Díky tomu jdou realizovat překlady kategorií, pokud je nainstalován modul *Vícejazyčné podpory*. Podle ID kategorie půjde poznat, v kterých jazycích je kategorie vytvořena.



Obrázek 6-1: Tabulky modulu *Kategorie*.

6.3 Implementace modulu Články

Aby tento modul správně fungoval, je potřeba mít nainstalován modul *Kategorie* a mít vytvořenou alespoň jednu kategorii. Pokud tato situace nastane, je uživateli napsána chybová hláška o nepřítomnosti modulu *Kategorie*. Řešeno je to pomocí volání a zachytávání výjimek.

Prezentační vrstva

Pro zobrazování náhledů článků je opět použit DataList. Do DataListu jsou načítány články buď všechny nebo články pouze jedné kategorie. Pro zjištění zda nemají být články filtrovány podle kategorií se kontroluje URL adresa, jestli neobsahuje parametr idcat, ve kterém je uloženo ID kategorie. Pokud existuje, kontroluje se obsah parametru IDModul. V tomto parametru je uloženo ID modulu, ve kterém byla kategorie vybrána. Hodnota parametru IDModul se musí shodovat s hodnotou uloženou v nastavení modulu. Pokud se neshodují, tak je parametr idcat ignorován, protože patří jinému modulu *Články*.

V tomto modulu je použito stránkování. Na stránkování je použita komponenta systému DNN. Je nutné nastavit pouze počet záznamů na stránku a celkový počet záznamů. O všechno ostatní se postará tato komponenta (v souborech ViewArticle.ascx a ViewArticle.ascx.cs).

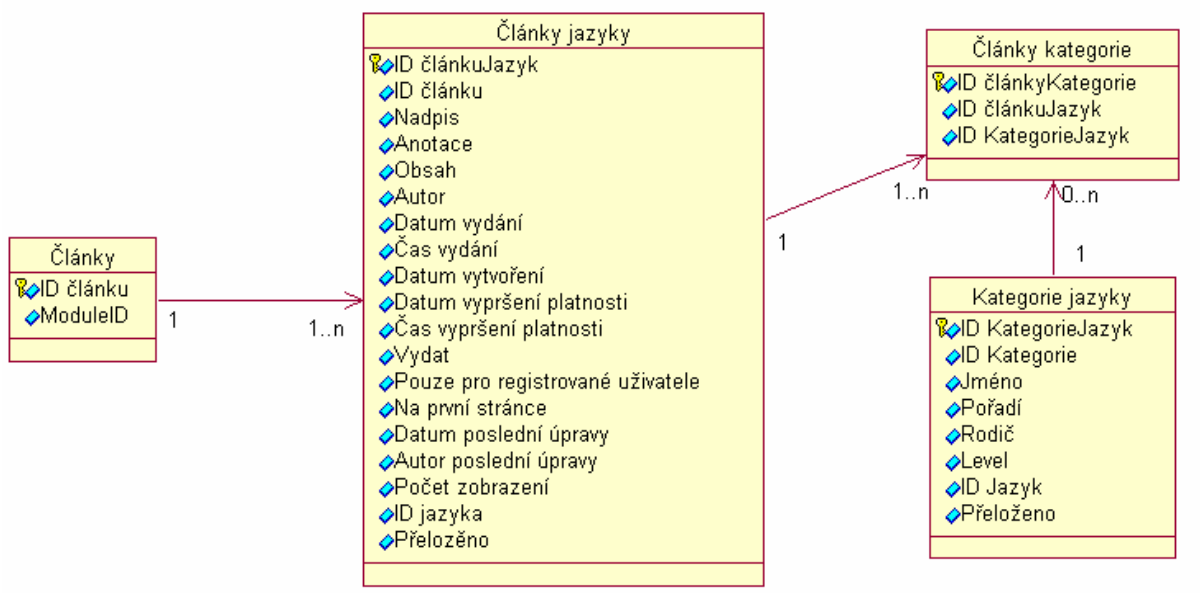
V editační části modulu se musí ke každému článku vybrat kategorie, do které bude patřit. Pokud je nainstalován modul *Vícejazyčné podpory*, tak jsou v DropDownListu vypsány jen kategorie z daného jazyka. Vybírat lze pouze z jazyků, ve kterých je vytvořena alespoň jedna kategorie. Opět je to řešeno pomocí atributů AutoPostBack a při změně jazyka se změní i kategorie (v souborech EditArticle.ascx a EditArticle.ascx.cs).

Při tvorbě článku lze zaškrtnout atribut *Zobrazit na úvodní stránce*. Pokud se už počet článků s tímto atributem rovná počtu článků možných zobrazit na stránku, zobrazí se místo CheckBoxu pro tento atribut oznámení, že již nelze přidávat články na první stránku. Je to hlídáno zvlášť pro jednotlivé jazyky a moduly (v souborech EditArticle.ascx a EditArticle.ascx.cs).

Nadpis musí být vždy u článku vyplněn. Requiredfieldvalidator hlídá, aby nedošlo k ukládání článku s nevyplněným nadpisem a vypíše uživateli chybovou hlášku. Datum vydání a datum vypršení platnosti vyžadují zápis ve správném formátu. Comparevalidator hlídá, aby nešel zadat špatný formát datumu nebo aby nezůstalo datum nevyplněné. Také čas vydání a čas vypršení platnosti vyžadují zápis ve správném formátu. Ten je hlídán pomocí regularexpressionvalidator. Hodnota regulárního výrazu je ^((([0-1][0-9]:)|(2[0-3]:)|([0-9]:))([0-5][0-9]))(:[0-5][0-9])? (v souborech EditArticle.ascx a EditArticle.ascx.cs).

Datová vrstva

I pro tento modul jsou vytvořeny dvě tabulky. Důvody jsou stejné jako u modulu *Kategorie*. Také způsob propojení je shodný jako u modulu *Kategorie*. Do tabulky nejsou ukládány obrázky samostatně, ale jakou součást textu anotace nebo obsahu. WYSIWYG editor v DNN umožňuje vkládání obrázků přímo do textu. Proto by bylo zbytečné mít je uložené v databázi zvlášť. Navíc, má při tomto řešení uživatel svobodu při výběru, kam obrázek vložit. Tabulka *články kategorie* slouží, aby byl článek vypsán i po kliknutí na rodičovskou kategorii kategorie, do které článek spadá.



Obrázek 6-2: Tabulky modulu *Články*

6.4 Implementace modulu Aktuality

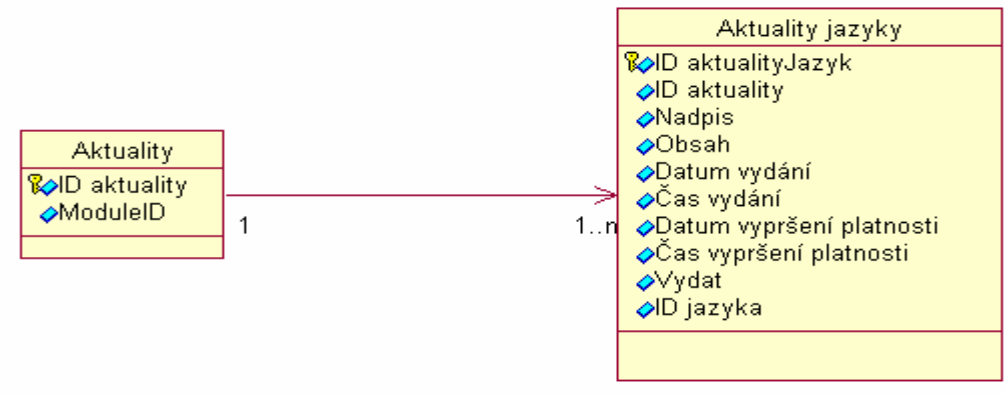
Implementace modulu *Aktuality* má hodně společných znaků s implementací modulu *Články*. Aktualita je vlastně jednoduchý článek.

Prezentační vrstva

Implementace této vrstvy je shodná jako u modulu *Články* pouze ochuzená o několik vlastností.

Datová vrstva

I tento modul je připraven pro podporu více jazyků a používá dvě tabulky.



Obrázek 6-3: Tabulky v modulu *Aktuality*

6.5 Implementace modulu vícejazyčné podpory

Z pohledu implementace jde o nejnáročnější modul, protože je bylo potřeba propojit se všemi ostatními moduly diplomové práce. DNN obsahuje podporu vícejazyčnosti, která má ale omezené možnosti. Tímto způsobem lze překládat obsahy modulů, ale už při tomto způsobu nelze zjistit, co přeloženo je a co ještě přeloženo není. Navíc nejde zobrazit na jedné stránce původní text a formulář pro překlad. Proto je překlad textu řešen jenom v modulu *Vícejazyčné* podpory, který tyto nevýhody nemá. Nevýhodou tohoto řešení je, že v modulu *Vícejazyčné podpory* se překládá obsah modulů a pomocí podpory vícejazyčnosti DNN se překládá hlavní menu a názvy modulů.

Prezentační vrstva

Pro výběr jazyků, které mají být kontrolovány je použit DualList. Jde o komponentu DNN. Obsahuje dva seznamy. V prvním jsou zobrazovány dostupné jazyk a v druhém zvolené jazyky (v souborech EditMultiLanguage.ascx a EditMultiLanguage.ascx.cs).

Pomocí SQL dotazů a volání výjimek je zjištěno, které typy modulů jsou v systému nainstalovány. Data, která obsahují, se mohou překládat. Pokud modul žádná data neobsahuje, tak není uživateli k překladu nabídnut. K překladu jsou nabízeny jen ty články, aktuality a kategorie, které ještě nebyly přeloženy (určeno podle atributu přeloženo).

Pro překlady článků, aktualit a kategorií jsou použity podobné formuláře jako pro jejich tvorbu. Možnosti při překladu jsou stejné jako při vytváření nového obsahu, protože jsou využívány třídy a funkce vytvořené pro předchozí moduly (soubory WhatTranslate.ascx a WhatTranslate.ascx.cs).

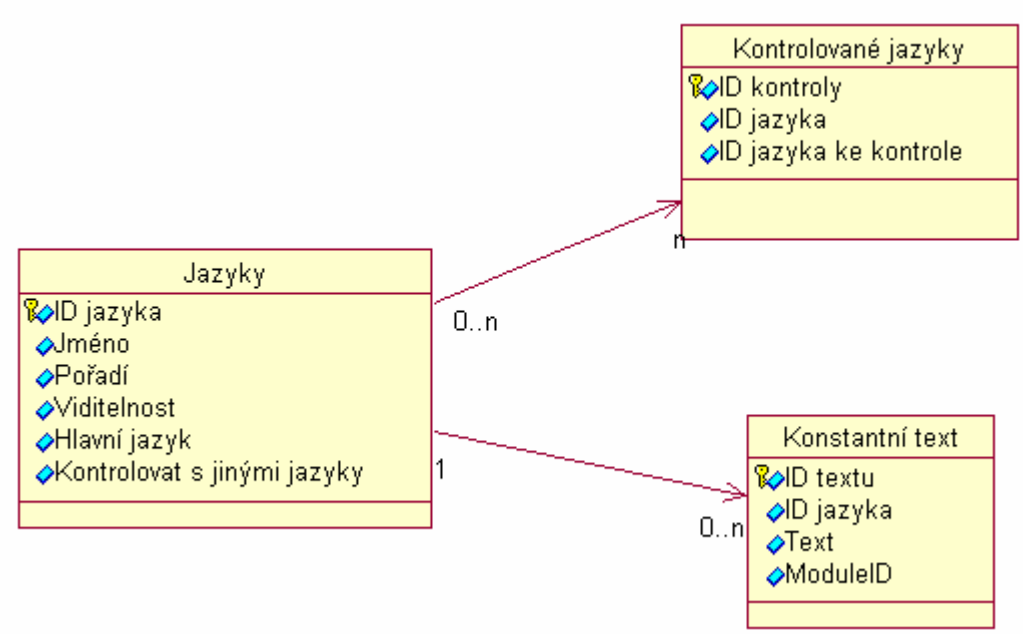
Pro překlad konstantního textu v modulech je využíváno tabulky v databázi a ne resx souborů, protože se tak nemusí pro každý modul vytvářet několik resx souborů a ušetří se tak prostor na

webovém serveru. V tabulce se také mohou případné informace lépe hledat než ve velkém množství souborů.

Při překladu je potřeba zjistit, pro které moduly umístěné na stránce se má překlad provádět. To se dá určit pomocí FriendlyName modulu. FriendlyName modulu, ale může administrátor DNN změnit a poté by modul přestal fungovat. Řešení použité v modulu je složitější a časově náročnější. Nejdříve se zjistí, které moduly se nachází na stránce. Poté se zjišťuje, zda modul obsahuje články, aktuality nebo kategorie. Pokud ano, tak jsou vloženy do seznamu modulů určených k překladu (soubory WhatTranslate.ascx a WhatTranslate.ascx.cs).

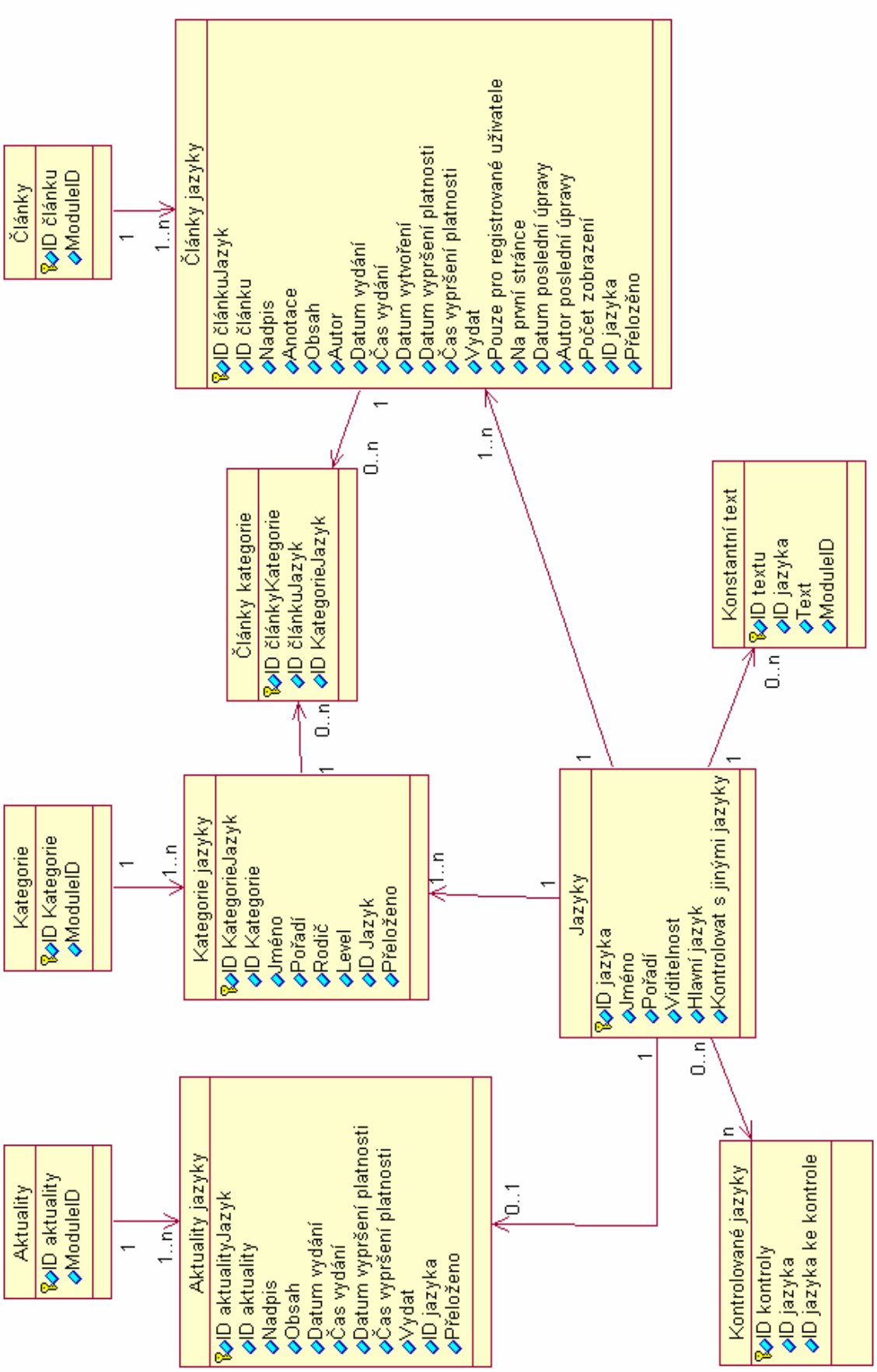
Datová vrstva

V tomto modulu jsou používány tři tabulky. V tabulce *Jazyky* jsou uloženy informace o jednotlivých jazycích. V tabulce *Kontrolované jazyky* je uloženo, se kterými jazyky má být jazyk srovnáván. V poslední tabulce jsou uloženy překlady konstantních textů v jednotlivých modulech.



Obrázek 6-4: Tabulky modulu *Vícejazyčné podpory*

Pro správnou funkci modulu *Vícejazyčné podpory* je tabulka *Jazyky* napojena na tabulky ostatních modulů pomocí ID jazyka. Do tabulek s články, aktualitami a kategoriemi je vložen nový sloupec *Přeloženo*. Ten obsahuje informaci, zda byl překlad dokončen. Pokud informaci neobsahuje, je překlad článku nebo aktuality považován za dokončený.



Obrázek 6-5: ER diagram všech modulů

7 Zhodnocení výsledků práce

Výsledkem této diplomové práce jsou čtyři moduly, které umožňují využívat portálový systém DNN jako plnohodnotný redakční systém. Dále budou popsány výsledky a možná rozšíření pro každý modul zvlášť.

Modul Články

V návrhu a implementaci tohoto modulu nejsou závažnější nedostatky. Modul umožňuje tvorbu i správu článků. Umožňuje u článků nastavit datum a čas vydání i datum a čas konce platnosti.

Možným rozšířením pro tento modul je přidání podpory komentářů k článku. Je sice možnost, dát si do článku odkaz na diskusi v modulu fórum, ale přímá podpora komentářů je pro návštěvníky stránek pohodlnější.

Modul Kategorie

Návrh tohoto modulu šel udělat více obecněji. Podle návrhu modul kategorie umožňuje zobrazovat tři levely zanoření podkategorií.

Možným rozšířením by tedy měla být možnost dát uživateli na výběr, kolik levelů zanoření při zobrazování podkategorií chce použít.

Modul Aktuality

Tento modul byl také bezproblémový jak na návrh, tak i na tvorbu. Je vytvořen dostatečně obecně a měl by vyhovovat požadavkům většiny uživatelů.

Jako možné rozšíření by bylo zavedení posílání aktualit na email. Návštěvník stránek by vyplnil žádost o aktuality (aby bylo předejito problémům se spamem) a měl by možnost si posílání aktualit na email kdykoliv odhlásit.

Modul Vícejazyčné podpory

Jak z pohledu návrhu, tak z pohledu implementace byl tento modul nejproblematičtější. I když problémy v implementaci byly způsobeny nedokonalým návrhem. Při návrhu jsem počítal, že celá podpora vícejazyčnosti bude řešena pouze přes modul *Vícejazyčné podpory*. Až později jsem zjistil, že tímto způsobem nepůjdou přeložit názvy modulů a hlavní menu. K tomuto je potřeba využít vestavěnou podporu vícejazyčnosti sytému DNN. Mé řešení je sice funkční, ale není příliš uživatelsky přívětivé, protože jazyk stránek se musí nastavovat na dvou místech. V modulu *Vícejazyčné podpory* a přímo v DNN. Druhá možnost je v hlavním menu používat názvy, které jsou stejné snad ve všech jazycích (například Home atd.) nebo ho nepoužívat vůbec.

Jako možné rozšíření by tedy bylo lepší propojení mezi modulem a vestavěnou podporou vícejazyčnosti. V diplomové práci to implementováno není, protože by to vyžadovalo velký zásah do zdrojových kódů a časově to již nebylo možné stihnout. Dalším možným rozšířením by bylo zobrazování vlaječek států k jednotlivým jazykům. Tyto obrázky by byly na webovém serveru uloženy a uživatel by si jeden z nich vybral. Pokud by mu nabídka dostupných vlaječek nevyhovovala, měl by možnost nahrát si obrázek svůj.

Závěr

Tato práce se zabývá změnou systému pro správu obsahu DNN na redakční systém. V současné době je na trhu několik kvalitních redakčních systémů a DNN se jim možnostmi nemůže rovnat. Cílem této práce bylo navrhnout a implementovat rozšíření, která umožní využívat DNN jako redakční systém.

V diplomové práci byl podrobně popsán portálový systém DNN a jeho možnosti. Důraz byl kladen na architekturu jádra DNN a architekturu modulů a popsání jednotlivých vrstev architektury. Na základě nastudování portálového sytému DNN a redakčních systémů obecně, byly navrženy čtyři rozšíření, které umožňují používat DNN jako redakční systém.

Výsledkem praktické části jsou čtyři funkční moduly, které byly vytvořeny podle dříve představených návrhů. Moduly jsou udělány dostatečně obecně, aby měly co nejuniverzálnější použití. Využití mohou tyto moduly najít například při prezentaci malé firmy na internetu nebo pro provozování jednoduchého elektronického magazínu.

Do budoucna, by mohly být pro lepší fungování systému přidány nové moduly a funkce. Například modul s podrobným vyhledáváním článků a aktualit na místo využívání interního vyhledávání DNN. Dále také lepší propojení modulu *Vícejazyčné podpory* s podporou vícejazyčnosti implementované v DNN. Ale i v současná podoba sytému je dobře použitelná.

Literatura

[1] **Wikipedia** [online]. 2007 , 1. 1. 2007 [cit. 2007-01-03]. Dostupný z WWW: http://cs.wikipedia.org/wiki/ASP.NET

[2] KAČMÁŘ, Dalibor. **Programujeme .NET aplikace ve Visual Stuidiu .NET**. Praha : Computer Press, 2001. 335 s. ISBN 80-7226-569-5.

[3] **Redakční systém DWE: o redakčních systémech (obecně)** [online]. 2006 [cit. 2007-01-03]. Dostupný z WWW: http://www.edituj.cz/o-redakcnich-systemech-obecne-9.html

[4] Zive.cz, **Začínáme s ASP.NET 2.0** [online]. 2005 [cit. 2007-01-03]. Dostupný z WWW: http://zive.cz/h/Programovani/AR.asp?ARI=125787&CAI=2037

[5] **Co je to DotNetNuke?** [online]. 2005 [cit. 2007-01-03]. Dostupný z WWW: <http://dotnetnuke.cz/DotNetNuke/CojeDotNetNuke/tabid/63/Default.aspx>.

[6] MACDONALD, Matthew, SZPUSZTA, Mario. **ASP.NET 2.0 a C# : tvorba dynamických stránek PROFESIONÁLNĚ**. Brno, Česká republika : ZONER software, s.r.o., 2006. 1376 s. ISBN 80-86815-38-2

[7] HRUŠKA, Tomáš. **Úvod do informačních systémů**. [s.l.] : [s.n.], 2002. 111 s.

[8] WEIDA, Petr. **SEO - Search Engine Optimization**. *Interval.cz* [online]. 2003 [cit. 2007-05-05]. Dostupný z WWW: <http://interval.cz/clanky/seo-search-engine-optimization/>.

[9] **Javascript - úvod**. *Jakpsatweb.cz* [online]. 2002 [cit. 2007-05-05]. Dostupný z WWW: <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>.

[10] WALKER, Shaun, et al. **Professional DotNetNuke ASP.NET Portals**. Indianapolis, Indiana, USA : Wiley Publishing, Inc., 2005. 421 s. ISBN 0-7645-9563-6.

[11] Microsoft Corporation. MSDN [online]. c2007 [cit. 2007-05-05]. Dostupný z WWW: <http://msdn2.microsoft.com/en-us/default.aspx>.

[12] WASHINGTON, Michael. DotNetNuke Documentation. [s.l.] : [s.n.], 2007. 142.

Seznam příloh

Příloha 1. Uživatelská příručka pro práci s moduly vytvořenými v této diplomové práci

Příloha 2. CD se zdrojovými texty

Příloha 1.

Uživatelská příručka

Tato uživatelská příručka obsahuje návod na instalaci a používání modulů *Články, Aktuality, Kategorie a Vícejazyčné podpory*.

Instalace

Před instalací modulu je nutné se nejprve přihlásit do systému jako uživatel Host. Možnost instalace nového modulu se nachází pod nabídkou Module Definition v Host. V této nabídce se nachází možnost Install New Module. Po kliknutí na tuto možnost je uživatel vyzván k vložení souboru s modulem. Po provedení této akce, je modul vložen do systému a připraven k použití.

Vkládání modulu na stránku se provádí pomocí kontrolního menu v horní části stránky. Pro vkládání modulů je potřeba být přihlášen jako uživatel Host nebo jako uživatel Administrátor. V kontrolním menu musí být zvolena možnost Add New Module. Nyní lze z nabídky vybrat určený modul a místo, kde na stránce se bude nacházet. Po kliknutí na možnost Add nebo ikonku plus, je modul vložen na stránku a připraven k použití.

Ovládání modulu Kategorie

Modul by měl být vložen do levého nebo pravého panelu stránky, protože pro tyto panely je uzpůsoben jeho vzhled. Jde vložit i do ostatních panelů na stránce, ale může poté narušit design stránky.

Pro otevření formuláře pro vytvoření nové kategorie je potřeba kliknout na tlačítko Add Category nebo vybráním položky Add Category z rozbalovacího menu modulu. Pro úpravu již existující kategorie je potřeba kliknout na ikonku tužky, která se nachází vedle názvu každé kategorie. Po stisknutí tlačítka se otevře formulář pro tvorbu nové kategorie.

Při tvorbě nové kategorie je potřeba nejdříve zadat jméno kategorie, zvolit jazyk (pokud je nainstalován modul *Vícejazyčné podpory*) a vybrat rodičovskou kategorii. Pokud má kategorie být mezi hlavními, zvolí se možnost root. Po stisku tlačítka Add Category je kategorie vložena do seznamu kategorií se stejnou rodičovskou kategorií. V tomto seznamu se dá nastavit pořadí zobrazovaní jednotlivých kategorií. Po stisknutí tlačítka update je nová kategorie uložena.

Při úpravě kategorie se otevře stejný formulář jako při tvorbě nové kategorie, ale bude už obsahovat polžky vybrané kategorie. U kategorie jde měnit jméno, rodičovská kategorie i pořadí zobrazování. Změna je provedena po stiknutí tlačítka update.

Pro smazání kategorie je potřeba opět kliknout na ikonku tužky vedle názvu kategorie. Smazání se provede stisknutím tlačítka delete. Pokud byly v této kategorii vytvořeny nějaké články nebo kategorie obsahuje nějaké podkategorie, je potřeba zvolit, co se s nimi má stát po smazání

kategorie. Na výběr jsou dvě možnosti. První možnost je vložit články a podkategorie do jiné kategorie. Nová kategorie se vybírá z nabídky dostupných kategorií. Druhou možností je smazání těchto článků a kategorií. Tuto možnost je potřeba potvrdit.

Ovládání modulu články

Pro správné fungování tohoto modulu je potřeba mít na stránce vložen modul *Kategorie* a v něm mít vytvořenu alespoň jednu kategorii. Modul by měl být vložen do středního panelu, protože na tento panel je uzpůsoben jeho vzhled. Jde vložit i do ostatních panelů na stránce, ale může poté narušit design stránky.

Pro otevření formuláře pro tvorbu nového článku je potřeba kliknout na tlačítko Add New Article nebo vybráním položky Add New Article z rozbalovacího menu modulu. Pro úpravu již existujícího článku je potřeba kliknout na ikonku tužky, která se nachází vedle nadpisu článku.

Při tvorbě nového článku je nutné zadat nadpis článku, vybrat kategorii pro článek a jazyk, ve kterém bude článek napsán (pokud je nainstalován modul *Vícejazyčné podpory*). Ostatní možnosti jsou nepovinné a mohou zůstat nevyplněné. Při psaní textu anotace a obsahu článku je možné využít WYSIWYG editor a přizpůsobit tak text, co nejvíce svým představám. Obrázky se do článku vkládají také pomocí WYSIWYG editoru. Je možné nastavit datum a čas, kdy má článek vyjít a datum a čas, kdy bude ze stránek stažen. Pokud tyto údaje zůstanou nevyplněny, budou doplněny automaticky. Datum a čas vydání článku se nastaví na aktuální datum a datumu a času konce platnosti článku se nastaví nejpozdější možné datum (12/31/9999 23:59:59). Pokud má být článek zobrazován jen registrovaným uživatelům, je potřeba zaškrtnou možnost Visible only for registered users. Pokud se jedná o důležitý článek, je možnost nechat ho zobrazovat vždy na první stránce. Slouží k tomu možnost On first page. Pokud se již na první stránku více článků nevejde, není tato možnost zobrazována. Jestliže ještě článek není hotov, ale je potřeba ho uložit, tak možnost publish musí zůstat nezaškrtnuta. Článek bude uložen do databáze, ale nebude se návštěvníků stránek zobrazovat a může být kdykoliv dokončen. Uložení článku se provede stisknutím talčítka update.

Při úpravě článku se zobrazí stejný formulář jako při tvorbě nového článku, ale bude již obsahovat atributy upravovaného článku. Upravovat lze všechny možnosti. V dolní části formuláře bude zobrazeno jméno uživatel, který provedl poslední změnu v článku a datum této změny. Smazání článku se provede stisknutím tlačítka delete.

Pokud je na stránce více modulů *Kategorie*, je možné v nastaení modulu vybrat, ze kterého modulu se budou kategorie čerpat. Dále lze v nastavení modulu vybrat, kolik článků na jednu stránku má být zobrazeno a podle čeho budou články seřazeny.

Ovládání modulu aktuality

Tento modul by měl být vložen v pravém nebo levém panelu, protože pro tyto panely je přizpůsoben jeho vzhled. Jde vložit i do ostatních panelů na stránce, ale může poté narušit design stránky.

Pro otevření formuláře pro tvorbu nové aktuality je potřeba kliknout na tlačítko Add Currnet News nebo vybráním položky Add Current News z rozbalovacího menu modulu. Pro úpravu již existující aktuality je potřeba kliknout na ikonku tužky, která se nachází vedle nadpisu článku.

Při tvorbě nové aktuality je nutné zadat nadpis aktuality a jazyk, ve kterém bude aktualita napsána (pokud je nainstalován modul *Vícejazyčné podpory*). Ostatní možnosti jsou nepovinné a mohou zůstat nevyplněné. Při psaní textu aktuality je možné využít WYSIWYG editor a přizpůsobit tak text, co nejvíce svým představám. Obrázky se do aktuality vkládají také pomocí WYSIWYG editoru. Je možné nastavit datum a čas, kdy má aktualita vyjít a datum a čas, kdy bude ze stránek stažena. Pokud tyto údaje zůstanou nevyplněny, budou doplněny automaticky. Datum a čas vydání článku se nastaví na aktuální datum a datumu a času konce platnosti článku se nastaví nejpozdější možné datum (12/31/9999 23:59:59).

Při úpravě aktuality se zobrazí stejný formulář jako při tvorbě nové aktuality, ale bude již obsahovat atributy upravované aktuality. Upravovat lze všechny možnosti. V dolní části formuláře bude zobrazeno jméno uživatel, který provedl poslední změnu v aktualitě a datum této změny. Smazání článku se provede stisknutím tlačítka delete.

V nastavení modulu lze zvolit, kolik aktualit se bude zobrazovat v panelu na stránce. Dále lze nastavit kolik aktualit se bude zobrazovat na stránce při zobrazení všech aktualit.

Ovládání modulu Vícejazyčné podpory

Tento modul by měl být vložen v pravém nebo levém panelu, protože pro tyto panely je přizpůsoben jeho vzhled. Jde vložit i do ostatních panelů na stránce, ale může poté narušit design stránky. Tento modul poskytuje jazyky pro celý portál, proto je vhodné nechat si ho zobrazit na všech stránkách.

Pro otevření formuláře pro tvorbu nového jazyka je potřeba kliknout na tlačítko Add New Language nebo vybráním položky Add New Language z rozbalovacího menu modulu. Pro úpravu již existujícího jazyka je potřeba kliknout na ikonku tužky, která se nachází vedle nadpisu článku.

Při tvorbě jazyka je nutné zadat pouze jeho název. Ostatní položky nejsou povinné. Je možné nechat si kontrolovat, zda je obsah modulů *Články*, *Kategorie* a *Aktuality* shodný jako v jiném jazyce. Na to slouží možnost Control with other languages. Po zatrhnutí této možnosti se zobrazí seznam dostupných jazyků a lze z něj vybrat ty, se kterými bude jazyk kontrolován.

Při úpravě jazyka je možné měnit všechny jeho atributy. Smazání jazyka se provádí stiknutím tlačítka delete. Pokud jazyk obsahuje nějaké články, kategorie či aktuality, je potřeba zvolit, co se s nimi má po smazání jazyka stát. Na výběr jsou dvě možnosti. První možnost umožňuje vložit články, aktuality nebo kategorie do jiného jazyka. Druhá možnost je nechat všechno smazat. Tuto možnost je potřeba potvrdit.

Pro překládání obsahu modulů je potřeba otevřít formulář pro úpravu jazyka, do kterého se bude překládat. V tomto formuláři se zobrazí čtyři možnosti pro překlad: překlad článků, překlad kategorií, překlad aktualit a překlad konstatního textu v modulech. Při zvolení jedné z prvních tří

možností se otevře formulář, ve kterém budou vypsány články, kategorie nebo aktuality určené k překladu. U kategorií je v seznamu napsán název, u článků a aktualit je vypsán jejich nadpis. Je možné si vybrat z jakého jazyka a moduly mají být vybrány. Po zvolení článku, kategorie nebo aktuality se otevře formulář určený pro samotný překlad. Ten bude vypadat podobně jako formuláře popsané v dřívějších kapitolách. I možnosti budou stejné jako u těchto formulářů. Pro příklad u překladu článků bude na formláři nejdříve originální nadpis a pod ním místo pro nový text. Stejné pro všechny ostatní atributy článku. Při překladu lze zvolit zda byl již překlad dokončen nebo ne. Slouží pro to možnost Complete. Pokud zůstane nezaškrtnutá, článek bude nadále považován za nepřeložený a je možné se k němu kdykoliv vrátit a překladu pokračovat.