

titulní strana bakalářské práce

oficiální zadání bakalářské práce

ABSTRAKT

Bakalářská práce se zabývá vytvořením publikačního systému PubSys s využitím produktů a technologií firmy Microsoft.

Publikační systém PubSys je serverovská webová aplikace, pracující s daty v relačním databázovém serveru a poskytující je co by webové stránky. Systém se chová jako elektronické noviny nebo časopis, kde autoři publikují články a čtenáři o nich diskutují.

ABSTRACT

The main emphasis of this bachelor thesis is to create a publication system PubSys with technologies of Microsoft firm usage.

Publication system PubSys is a web sever based application, with data in relational database server working and providing them as a web pages. Behaviour of this system is like electronic news or journale, where articles makes public by authors and readers discuss about them.

Děkuji vedoucí mé bakalářské práce Ing. Zdence Prokopové, CSc. za odborné vedení, cenné rady a připomínky potřebné a přínosné pro vypracování mé bakalářské práce. Chtěl bych zde také vyjádřit poděkování své skvělé manželce, kterou jsem po čas programování tohoto systému a psaní této práce zanedbával. Dík patří také Bohu, neboť jsem téměř nevěřil, že najdu dostatek času, nápadů a trpělivosti především pro sepsání této práce.

OBSAH

ÚVOD.....	7
I. TEORETICKÁ ČÁST	9
1 WEBOVÉ TECHNOLOGIE.....	10
1.1 HISTORIE WEBOVÉ TVORBY	10
1.2 PROBLÉMY S PŮVODNÍ (PŘEDCHOZÍ) TECHNOLOGIÍ ASP	10
1.3 PŘEHLED ASP.NET	13
1.3.1 Základní informace	13
1.3.2 Typy webových programů	13
1.3.3 Novinky v ASP.NET	14
1.4 NĚCO MÁLO O .NET FRAMEWORK	15
II. PRAKTICKÁ ČÁST	17
2 STANOVENÍ CÍLE	18
2.1 POŽADAVKY	18
2.2 POUŽITÉ TECHNOLOGIE	19
3 ANALÝZA PROBLEMATIKY.....	20
3.1 DATABÁZE	20
3.2 UŽIVATELSKÉ ROZHRANÍ.....	21
3.3 APLIKAČNÍ OBJEKTY	22
4 NÁVRH PUBLIKAČNÍHO SYSTÉMU - Z POHLEDU SKLADOVÁNÍ DAT.....	23
4.1 VYTVOŘENÍ DATABÁZE	25
4.2 ULOŽENÉ PROCEDURY	26
4.3 INSTALAČNÍ SQL SCRIPT	26
4.4 PŘIPOJENÍ A KONFIGURACE DATABÁZE.....	27
5 NÁVRH PUBLIKAČNÍHO SYSTÉMU - Z POHLEDU GRAFICKÉHO DESIGNU.....	28
5.1 OBROVSKÝ POMOCNÍK	28
5.2 UŽIVATELSKÉ ROZHRANÍ.....	28
5.2.1 Titulní stránka: <i>default.aspx</i>	29
5.2.2 Strana pro článek: <i>co.aspx</i>	29
5.2.3 Diskusní strana: <i>komentare.aspx</i>	30
5.2.4 Strana pro přidání diskusního příspěvku: <i>komentar_novy.aspx</i>	30
5.2.5 Multifunkční chameleon <i>archiv.aspx</i>	31
5.2.6 Výpis článků autora: <i>autor.aspx</i>	31
5.2.7 Služba pro emailové zprávy: <i>zpravodaj.aspx</i>	32
5.2.8 Přihlašovací stránka: <i>login.aspx</i>	32

5.3	UŽIVATELSKÉ ROZHRAŇÍ - ČÁST PRO AUTORY	33
5.3.1	Vložení nového článku: <i>/redakce/default.aspx</i>	33
5.3.2	Sklad příloh: <i>/redakce/spravce_souboru.aspx</i>	34
5.3.3	Články k editaci: <i>/redakce/vyber_clanku_keditaci.aspx</i>	34
5.3.4	Stránka pro editaci článku: <i>/redakce/editace_clanku.aspx</i>	34
5.3.5	Administrace seznamu kategorií: <i>/redakce/kategorie.aspx</i>	35
5.3.6	Seznam emailů: <i>/redakce/zpravodaj.aspx</i>	35
5.3.7	Členové redakčního týmu: <i>/redakce/autori.aspx</i>	35
5.3.8	Informace o přístupech a odkazech: <i>/redakce/statistika_zajmu.aspx</i>	36
5.3.9	Pomůcka administrátora: <i>/redakce/komentar_smazat.aspx</i>	37
5.3.10	Informace o chybě: <i>chyba.aspx</i>	37
5.3.11	Ještě něco málo ke grafice	37
6	VYTVOŘENÍ FUNCČNÍ APLIKACE PUBLIKAČNÍHO SYSTÉMU	39
6.1	KONFIGURACE PUBSYS	39
6.1.1	Připojení k databázi	40
6.1.2	CACHEování menu	40
6.2	RSS KANÁLY	40
6.2.1	RSS kanál článků: <i>/rss/rss_clanky.aspx</i>	41
6.2.2	RSS kanál komentářů: <i>/rss/rss_komentaru.aspx</i>	41
6.3	CO KDE LEŽÍ	41
6.3.1	Složka <i>/aspnet_client/</i>	42
6.3.2	Složka <i>/bin/</i>	42
6.3.3	Složka <i>/DB PubSys/</i>	42
6.3.4	Složka <i>/grafika/</i>	42
6.3.5	Složka <i>/komponenty/</i>	42
6.3.6	Složka <i>/redakce/</i>	42
6.3.7	Složka <i>/rss/</i>	42
6.3.8	Složka <i>/sablony_xslt/</i>	43
6.3.9	Složka <i>/soubory/</i>	43
6.4	POUŽITÝ SOFTWARE	43
7	ROZŠÍŘENÍ SYSTÉMU O DALŠÍ DOPLŇUJÍCÍ FUNKCE	44
	SEZNAM POUŽITÉ LITERATURY	48
	SEZNAM POUŽITÝCH SYMBOLŮ, ZKRATEK A POJMŮ	49
	SEZNAM OBRÁZKŮ	51
	SEZNAM PŘÍLOH	52

ÚVOD

Význam publikování dat

Lidé měli od pradávna potřebu si předávat informace a poznatky. Nejprve to bylo ústní formou, poté vzniklo písmo, omnoho později přišel vynález tisku a v současnosti je jedním z největších prostředků k šíření informací osobní počítač, využívající počítačovou síť internet.

WWW

Jednou z nejznámějších a také nejpoužívanějších služeb internetu je WWW (dále jen web). Tato služba byla od samého začátku určena k publikování, poskytování a vyhledávání informací.

Web v průběhu let prošel bouřlivým vývojem. Dá se říct, že tento vývoj (resp. jeho technologií) se doposud nezastavil a v blízké budoucnosti bude určitě pokračovat.

Informace z webu získáváme hlavně přes program - www prohlížeč (dále jen prohlížeč) zobrazováním tzv. www stránek (dále jen stránek). Tyto stránky jsou buď fyzicky uloženy na centrálních počítačích - serverech nebo jsou na nich dynamicky generovány.

Vzhledem k tomu, že existuje mnoho platforem (MS Windows, Linux a pod.), vzájemně těžce spolupracujících (nekompatibilních), je právě web jedním z nejlepších prostředků pro poskytování / získávání informací, díky své nezávislosti na platformě.

Web funguje díky využití několika technologií a standardů. O jejich standardizaci se stará nezávislé konsorcium W3C. Bohužel ne všemi normami a doporučeními W3C se řídí výrobci prohlížečů a často je nedodržují ani autoři stránek.

Princip

Jednoduše řečeno pracuje web takto: uživatel zadá adresu stránek, prohlížeč se pokusí nalézt server, který požádá o tuto stránku. Jeli tato stránka uživateli přístupná pošle ji server prohlížeči, který ji zobrazí. Komunikace mezi serverem a uživatelským prohlížečem (dále jen klient) probíhá většinou pomocí protokolu HTTP. Samotná stránka

je textový soubor poskládaný z různých speciální značek (TAGů) a textu. Pro tyto značky existuje několik norem, nejznámější a nejpoužívanější je HTML a XML.

Už vidím, jak mě laskavý čtenář připomíná, že v prohlížeči vidí (slyší) nejen text, ale i obrázky a jiné multimedia. Multimedia a grafika jsou však většinou samostatné soubory, na které se stránka pouze odkazuje.

Servery

O skladování a poskytování informací (př. další činnosti) se starají speciální výkonné počítače - webové servery. Z pohledu serveru mohou být stránky statické nebo dynamické (není to totéž co rozumíme pod pojmem statické a dynamické stránky na klientu). Statická stránka na serveru je taková, kdy s ní server nic nedělá a jednoduše ji pošle na požadavek klienta. Dynamická stránka je buď celá nebo částečně generována serverem.

Z pohledu software je webový server program (nebo více programů) běžící na jednom nebo více počítačích (webové farmy). Webový server se může starat o více webových míst (web sites).

Na téměř každé platformě existuje minimálně jeden webový server. Např. u operačního systému Linux je to server Apache, u Windows zase IIS server.

Publikační systémy

Na internetu existuje mnoho webových sídel podávajících informace určitým specifickým způsobem, jednotným pro všechny své články. Tyto weby jsou zpravidla různé noviny (např. www.idnes.cz, www.ihned.cz, www.novinky.cz, www.sport.cz atd.), nebo magazíny vydávané z různou pravidelností zaměřené na určité oblasti, zpravidla odborné (např. www.interval.cz, www.grafika.cz atd.). Jednotný způsob katalogizace a publikování informací zajišťují tzv. publikační systémy (dále jen PS).

I. TEORETICKÁ ČÁST

1 WEBOVÉ TECHNOLOGIE

1.1 Historie webové tvorby

Od dob prvních webových serverů a prohlížečů stránek HTML neuběhlo mnoho času. Původní primitivní prohlížení statických dokumentů bylo obohaceno o různé technologie na straně prohlížečů ve formě Java skriptů, komponent Active X, až po různé server-side koncepce jako jsou server-side skripty (skriptlety), aktivní serverové stránky (ASP) a aplikace ISAPI. Velkých změn doznal i samotný jazyk HTML existující v mnoha různých verzích. Dostupný i v dynamické variantě – DHTML, dávající tvůrcům webových stránek možnosti přiblížit jejich vzhled běžným desktopovým aplikacím a samozřejmě XHTML – kombinující XML a HTML.

Vedle na první pohled viditelných změn, odrážejících se ve formě prezentace informací uživateli, doznaly webové technologie nemalých změn uvnitř. Prezentované informace na dynamicky generovaných stránkách vyžadují dobrou a rychlou technologii přístupu k datům, nejčastěji ve formě databází. Zcela samostatným, ale velmi důležitým aspektem webových technologií se stala bezpečnost. Všechny informace přenášené v síti Internet nejsou určeny každému oku. K zabezpečení muselo dojít na dvou úrovních. Transportní cesta, realizovaná bezpečnými protokoly, tvoří první z nich. Druhým stupněm bezpečnosti je ověření uživatele a individuální řízení přístupu k jednotlivým informacím.

Jak vývoj webových aplikací postupoval, začal se jejich formát intenzivně blížit k desktopovým aplikacím. Uživatelé již nepřistupují k webovým stránkám jako k elektronické podobě dokumentů, ale vedou s rozsáhlými aplikacemi „rozhovor“. Ovládají je, pohybují se z jedné části do druhé, zadávají a čtou informace. Po celou dobu předpokládají, že si „jejich web“ bude pamatovat vše, co v předchozích krocích provedli. Očekávají identické chování jako u běžných aplikací, lokálně nainstalovaných na počítači. Webové servery dnes podporují pro tento účel tzv. session management.

1.2 Problémy s původní (předchozí) technologií ASP

ASP.NET je zcela novou verzí pokračující v tradici technologie ASP stránek verze 3.0, implementované jako součást Internet Information Serveru (IIS). Nejde pouze o rozšířenou verzi, nesoucí několik velmi podstatných změn oproti předchůdkyni. ASP.NET bylo přebudováno zcela od základů a přináší v mnohých ohledech nový pohled

na tvorbu webových stránek a aplikací. Než se dostaneme k technickým detailům, představme si problémy, jimiž předchozí verze trpěly a postupně si popíšme jejich řešení. Jazyk HTML používaný k přenosu informací je velice jednoduchý a dobře čitelný. Každému z nás bude zajisté trvat pouze pár minut, než vytvoří jednoduchou stránku s nadpisem a prostým textem. Technologie ASP přinesla možnost generovat obsah stránek HTML na serveru dynamicky díky uložení různých skriptů a ASP tagů do kódu HTML. Typická ASP stránka se jen hemžila dvojicemi znaků `<% a %>`. Kód umístěný mezi těmito symboly nebyl zasílán klientovi, ale je zpracováván interpretem na straně serveru. Na následujícím řádku vidíme jednoduchý příklad, který nikoho nepřekvapí.

Aktuální čas je `<% = now %><BR>`

Stránky ASP díky prolínání HTML a ASP kódu trpěly dvěma neduhy. Za prvé vymizela jejich jednoduchost a orientovat se ve složitějším kódu bylo velice obtížné, i když mnohé speciální editory dovedly barevně odlišit ASP kód od zbytku obsahu stránky. Druhý, navazující problém se týkal obtížnosti tvorby těchto stránek.

Ty tam jsou doby, kdy bylo jedno, jak webové stránky vizuálně vypadají. Aby nepůsobily jako výtvor sedmiletého školáka, musí na nich pracovat grafický designér ovládající jazyk HTML. Pro tyto pracovníky je problematické vytvářet logiku stránky ve skriptovacích jazycích. Kolektiv musí být proto rozšířen o programátora. Všichni ale pracují s jedním souborem a jejich příspěvky se neustále prolínají. To přináší nemalé problémy. Výsledkem je něco, co Angličané výstižně nazývají „spaghetti code“.

Technologie ASP byly opřeny o skriptovací jazyky interpretované na straně serveru. Každý ví, že dosahovaná rychlost zpracování kódu nebyla nijak závratná (ze stejného důvodu se běžné aplikace překládají do nativního kódu mikroprocesoru a spouštějí tou nejvyšší možnou rychlostí). S tvorbou tehdejších ASP aplikací souvisí i další problém, kterým je odlišný programový model. V krátkosti a zjednodušeně lze říci, že zkušený programátor v jazyce Visual Basic nebo Visual C++ nemá mnoho znalostí, které by mohl využít při tvorbě ASP stránek. Způsob a architektura jejich tvorby je zcela jiná než ve formulářích VB nebo MFC knihovně ve VC++.

Jednou z velkých výhod serverových technologií je snížení nároků na klientské prohlížeče. Všechny specifické jazykové a technologické závislosti zůstávají na straně serveru a ten „chrlí“ na prohlížeč téměř nezávislý kód HTML. Potřebná univerzalita webového řešení je pouze částečná. Server ve skutečnosti odpovídá na http dotazy kódem

HTML, který mu navrhl autor a programátor stránky. Širé vody Internetu jsou plné různých verzí prohlížečů a je nešťastné nevyužívat jejich možností. Cíle webových stránek byly v posledních letech dokonce rozšířeny o zcela nová zařízení komunikující např. protokolem WAP (Wireless Application Protocol). I k nim by bylo dobré promlouvat z jednoho typu aplikace. Bohužel IIS a ASP nedovedly v dřívějších verzích automaticky rozpoznat „s kým mají tu čest“ a podle toho generovat odpovídající kód. Tato úloha plně zůstávala v rukou tvůrce webu trpícího pod tíhou univerzálního řešení.

Webové aplikace, jak jsme si je v předchozí kapitole nastínili, často začínají připomínat desktopová nebo klient/server řešení. Webové stránky se stávají uživatelským rozhraním, jednou z vrstev v rozsáhlé n-vrstevnaté architektuře celé aplikace. Na středních vrstvách se nejčastěji vyskytovaly COM komponenty, jejichž bolesti jsou známé. Všechny se dotýkají i webových aplikací. Problémy s ilustrací, DLL Hell apod. Jedním z typických, pro svět desktopových aplikací docela zanedbatelným problémem, je upgrade COM komponent. Před nahráním nové verze, by jsme museli běh COM serveru zastavit a s ním většinou i celou aplikaci. Jak by se na takový upgrade díval klient, očekávající funkcionalitu označovanou jako 7x24, si můžeme domyslet.

V neposlední řadě si uvedme problémy související s udržováním stavu relace (session) pro jednotlivé uživatele webové aplikace. ASP server nabízí session management pro připojení individuálních klientů k aplikaci. Často potřebujeme uchovat i stav na jednotlivých webových stránkách, zejména těch obsahujících HTML formuláře. Stránky samotné nepodporují ukládání stavu a jeho realizace je plně v rukou programátora v ASP. Vrátime-li se k řízení stavu spojení v IIS serveru, i ono samotné má jistý nedostatek – je závislé na konkrétním počítači. Již mnoho let jsou podporovány technologie nabízející rozkládání výkonu na farmu serverů. Již tehdy nebyl problém je vystavět na technologiích firmy Microsoft. Zádrhel u ASP vznikl ve chvíli, kdy jsme požadovali řízení stavu. Každý IIS server si udržoval stavy svých relací a nesdílel je s ostatními ve farmě. Tehdejší řešení byly vždy postavena na programátorském umu tvůrce aplikace.

V předchozích odstavcích bylo představeno několik bolestivých otázek, dotýkajících se tvůrců webových aplikací, kteří pracovali s ASP. Dále se bude práce věnovat pouze verzi ASP.NET a řešení uvedených neduhů.

1.3 Přehled ASP.NET

1.3.1 Základní informace

ASP.NET je úplně znovu vybudovanou technologií běžící nad .NET Framework a využívající veškerá vylepšení, která společná základna přináší. Postavíte-li své aplikace na ASP.NET, budete příjemně překvapeni, o jak kvalitní prostředí jde, jak se zvýší výkon aplikací a zmenší riziko pádu. Nezanedbatelným přínosem je jednoduchost, s jakou lze aplikace vytvářet. [1]

Dříve, než si přehledně shrneme základní charakteristiky ASP.NET je nutné zdůraznit, že i přes fakt komplexní přestavby, byla zachována plná zpětná kompatibilita, Zcela bezproblémově lze převést existující ASP stránky a aplikace do prostředí .NET a provozovat je nezměněnou funkcionalitou. Aby se odlišily „nové a staré“ ASP stránky, byla novým ASP.NET stránkám pozměněna přípona na aspx. Tyto soubory mohou nést čistý kód HTML, včetně dnes klasického ASP kódu nebo kód určený pro zpracování službami ASP.NET. Přejít od existujících řešení k novým stránkám může být zcela plynulý, postupný a bezbolestný.

ASP.NET je prostředí umožňující vytvářet webové aplikace ve všech jazycích kompatibilních s platformou .NET. Současné skriptovací jazyky jako jsou VBScript, JScript byly nahrazeny jazyky platformy .NET. Kód napsaný ve Visual Basicu (VBScript již neexistuje a byl plně nahrazen VB), C# nebo Jscript je běžnou součástí nových aplikací. A co více! Žádný z kódů již není jako v předchozích verzích interpretován, nýbrž vše je kompilováno do IL kódu a později do nativního Just-in-time kompilátoru. Výkonnost webových aplikací se tímto přístupem posouvá do jiných hodnot. Zpomalení způsobené prvotním procesem JIT kompilace je kompenzováno vyspělou konfigurovatelnou technologií web cache.

1.3.2 Typy webových programů

Vývojáři na platformě ASP.NET si mohou zvolit ze dvou typů programových modelů nových aplikací.

1. Webové formuláře – jde o zcela nový způsob tvorby formulářově založených webových stránek. Z pohledu programátora se jeví model jejich tvorby velice podobný desktopovým aplikacím v jazyce Visual Basic. Intuitivními vizuálními RAD nástroji lze

navrhnout vzhled formuláře na webové stránce a v odděleném kódu doplnit funkcionalitu formou zpracování předdefinovaných událostí. Programový model je natolik jednoduchý a přirozený, že jej rychle zvládnou i programátoři bez předchozích zkušeností s jazykem VB.

Pro webové formuláře byla navržena nová sada ovládacích prvků (controls), které jsou velice inteligentní a běží na straně serveru. Odtud plyne jejich pojmenování *ASP.NET SERVER-SIDE CONTROLS*.

2. Webové služby – jsou technologií, jak přistupovat k funkcionalitě webového serveru ze vzdálených aplikací a novým prostředkem pro šíření aplikací. Pomocí XML a dalších standardních technologií umožňují webové služby vzájemnou komunikaci aplikací s dalšími aplikacemi, či komponentami, a to zcela bez ohledu na to, kde jsou tyto komponenty nebo aplikace umístěny – zda na stejném počítači nebo na druhé straně zeměkoule.

1.3.3 Novinky v ASP.NET

Oba typy aplikací běží nad ASP.NET a potažmo i .NET Framework. Získáme tím přístup k různým vlastnostem a službám definovaným v CLR. Jako příklad poslouží již zmíněná Just-in-time kompilace nebo typová bezpečnost. Samotné prostředí ASP.NET přináší mnoho nových myšlenek. Jsou to zejména:

1. silná podpora vyrovnávací paměti (cache) s masivním API. *CACHING* je služba výrazně zvyšující výkonnost aplikací, které realizují časově náročné operace potřebné k získání výsledků. Není-li nutné při každém http dotazu generovat aspx stránky znovu a zatěžovat procesor, je nasnadě využít této služby. Caching je na platformě .NET velice dobře konfigurovatelný a tvůrce webu má široké možnosti, jak optimalizovat jeho funkcionalitu. ASP.NET poskytuje dva typy cache. První je *VÝSTUPNÍ CACHE (OUTPUT CACHE)*, sloužící k ukládání dynamických stránek a hodnot ovládacích prvků vygenerovaných na základě předchozího dotazu. Druhý typ cache realizuje tradiční *APLIKAČNÍ CACHE (OBJECT CACHE)*, do níž lze programově ukládat libovolné objekty a později je opakovaně použít bez nutnosti opětovného zakládání a inicializace.

2. podpora a řízení správy stavu aplikace a relace. Na rozdíl od ASP je možné stav společně sdílet na několika serverech tvořících farmu. Dokonce Existuje možnost kterýkoliv ze serverů restartovat a stavy zůstanou zachovány.

3. podpora řízení bezpečnosti webových aplikací. Bezpečnostní otázky na platformě ASP nebyly vyřešeny na uspokojivé úrovni. ASP.NET ve spolupráci s IIS a operačním systémem zpřístupňuje různé metody autentizace. Po zadání jména a hesla lze použít jednu z následujících metod:

- Windows nativní autentizace (basic, digest nebo integrovaná Windows autentizace NTLM nebo Kerberos).
- Microsoft passport autentizace
- Autentizace formulářem
- ASP.NET obsahuje také podporu pro vytváření vlastních autentizačních schémat.

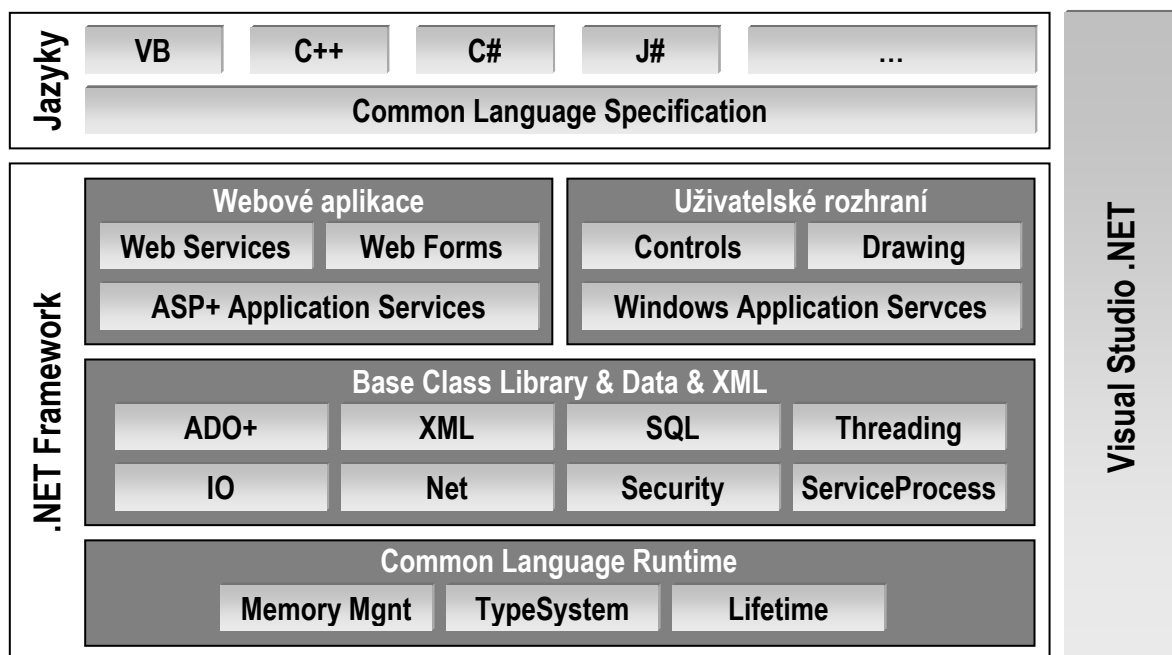
4. konfigurace ASP.NET aplikací formou souborů XML. Každá aplikace může mít své individuální konfigurační soubory, které jsou díky formátu XML nelimitovaně rozšiřované o vlastní konfigurační schémata.

5. nasazení upgrade webových aplikací jsou jednoduché jako příkaz `xcopy`. Runtime služby ASP.NET jsou schopny rozpoznat, že za běhu aplikace došlo k nahrazení souborů novějšími verzemi a zařídí automatický přechod na novou verzi. Běžící aplikace zpracuje aktuální dotazy a je ukončen její běh, včetně odstranění z paměti. Nové dotazy jsou již zpracovány novou verzí.

1.4 Něco málo o .NET Framework

Na tomto místě je nutné uvést pár odstavců o .NET Framework. Použít by se dalo vhodné přirovnání: „.NET Framework funguje doslovně jako ‚substrát‘, na kterém lze pěstovat SW. Stejně jako substrát pro květiny, i .NET Framework nabízí velké množství ‚živin‘, pro snadnou konstrukci a běh aplikací.“ [2]

Srdce platformy .NET Framework je nadstavba nad operačním systémem. Pokud bychom se podívali podrobně na jeho jednotlivé součásti, dostaneme následující strukturu.



Obr. 1. Struktura .NET Framework a navazující jazyky s vývojovým prostředím

Na úplně nejnižší úrovni se nachází Common Language Runtime (dále jen CLR) realizující základní infrastrukturu, nad kterou je celý Framework vybudován.

Nad CLR se nachází několik sad hierarchicky umístěných knihoven. Základem je knihovna nazvaná Base Class Library. Nad ní následuje podpora pro přístup k datům a práci s XML. Obě knihovny vytvářejí objektově orientovanou podporu pro efektivní a jednotné budování aplikací.

Poslední vrstvu tvoří dvě knihovny usnadňující programování webových aplikací a klasických aplikací s uživatelským rozhraním. Na tomto místě je nutné zdůraznit fakt, že jde o jazykově nezávislé knihovny, dostupné z libovolného vývojového nástroje. Poslední horizontální vrstvu tvoří nelimitovaná množina programovacích jazyků. Ve spodní části najdeme Common Language Specification, definující základní vlastnosti, jež jsou očekávány v každém programovacím jazyce na platformě .NET. Nad ní následují jednotlivé programovací jazyky. V současné době jsou firmou Microsoft podporovány např. Visual Basic, C++, C# a J#. Množina není uzavřená a libovolný výrobce ji může rozšířit o novou položku. Počet jazyků třetích stran, pokud je mi známo, dnes přesahuje hodnotu třicet a určitě není konečný (jsou mezi nimi např. PERL, Python, COBOL, aj.).

Ve vertikálním směru je celý .NET Framework podepřen vývojovým nástrojem Visual Studio .NET, velice úzce integrovaným s celou platformou. [2]

II. PRAKTICKÁ ČÁST

2 STANOVENÍ CÍLE

Cílem této práce je vytvoření publikačního systému PubSys ze zaměřením hlavně na jeho engine. Tedy na dynamické, funkční webové sídlo svázané s databází pro uložení dat a využívající dalších technologií.

2.1 Požadavky

- Design publikačního systému by měl být jen testovací, v reálném nasazení pak lehce změnitelný pomocí kaskádových stylů (dále jen CSS) a konfiguračních souborů.
- PS by měl evidovat příspěvky jednotlivých autentizovaných autorů, kde každý příspěvek sestává z nadpisu, abstraktu a vlastního textu, přiřazený pak může mít různé přílohy, obrázky a diskuzi čtenářů. Příspěvky by měli být zařazeny v kategoriích.
- Funkčnosti v PS by mělo být zobrazování aktuálních příspěvků (resp. jejich abstraktu) na hlavní stránce, dle pořadí a datumu publikování určeného redaktorem, který před publikováním provede jejich korekturu - kontrolu a schválí je pro uveřejnění.
- Příspěvky by také mělo být možno vypisovat podle datumu publikování, či kategorie. Editace příspěvků by měla být maximálně komfortní a bezproblémová pomocí WYSIWYG HTML editoru, s možností vkládání obrázků, jakož i ukládání roztodivných jiných souborů do vyhrazeného prostoru.
- Funkční emailový zpravodaj. Informace o nových článcích by se měly rozesílat na emaily zaregistrovaných čtenářů.
- K příspěvkům by čtenáři měli mít možnost psát komfortně komentáře. PS by měl zaznamenávat strukturu odpovědí do vláken (threadů) a umožňovat zasílání nových komentářů emailem.
- RSS kanál (feed) pro nejnovější příspěvky a také pro jednotlivé diskuse u článků. Vše dle standardu RSS 2.0. [3]
- Statistika přístupů.

2.2 Použité technologie

PS bude vytvořen na platformě MS Windows a z toho vyplývají i použité technologie dle následujícího výpisu.

- Databáze (dále jen DB) - databázovým strojem bude Microsoft SQL Server 2000, resp. Microsoft SQL Server 2000 Desktop Engine (MSDE). Licenční podmínky MSDE v minulosti Microsoft změnil, takže nyní může být používán a s aplikacemi distribuován bez omezení.
- Technologie XML bude použita pro ukládání hierarchických dat v DB. [4]
- Webovým serverem bude MS IIS (Microsoft Internet Information Services).
- Vyvíjen bude na OS Windows XP Professional, v němž je zabudován MS IIS verze 5.1. Ostré nasazení publikačního systému však předpokládá, kvůli výkonu, OS Windows Server 2003 nebo novější a web server MS IIS verze 6.0 nebo novější.
- O dynamičnost PS se bude starat současná webová technologie Microsoftu ASP.NET využívající Microsoft .NET Framework 1.1.

3 ANALÝZA PROBLEMATIKY

Nejdůležitějším krokem ve vývoji aplikace je analýza problematiky, z které pak vychází návrhy pro skladování dat, aplikační logiku a design aplikace. Bez dobré analýzy a návrhu se v průběhu dalšího vývoje mohou objevit velmi nepříjemné a těžko řešitelné problémy. Říká se, že každá strávená minuta nad analýzou a návrhem aplikace ušetří asi deset minut programování.

PS bude založen na třívrstevném aplikačním modelu. Uživatelské rozhraní, tedy to co vidí uživatel bude první vrstva, druhou vrstvou budou aplikační objekty zprostředkovávající komunikaci mezi uživatelským rozhraním a databází a konečně třetí vrstvou bude samotná databáze. O detailech každé vrstvy a nástinu požadavku na ni si povíme v následujících podkapitolách.

3.1 Databáze

První analýza bude analýza požadavku na data, která mají být na serveru ukládána. Již dopředu je jasné, že do této databáze bude potřeba zaznamenávat celkem pět druhů informací: autory a informace o nich; příspěvky - články; kategorie v nichž budou články zařazeny; seznam pro emailový zpravodaj a záznamy přístupů pro statistiku.

U autorů je potřeba uložit především jejich autentizační údaje, dále autorské jméno a také nutné kontaktní údaje.

Hlavní tabulkou databáze budou bezpochyby články. Zde by měly být uchovávány takové informace jako je titulěk abstrakt a tělo článku. Důležitý je také datum publikování, komentáře a datum posledního přidaného komentáře pro účely statistiky.

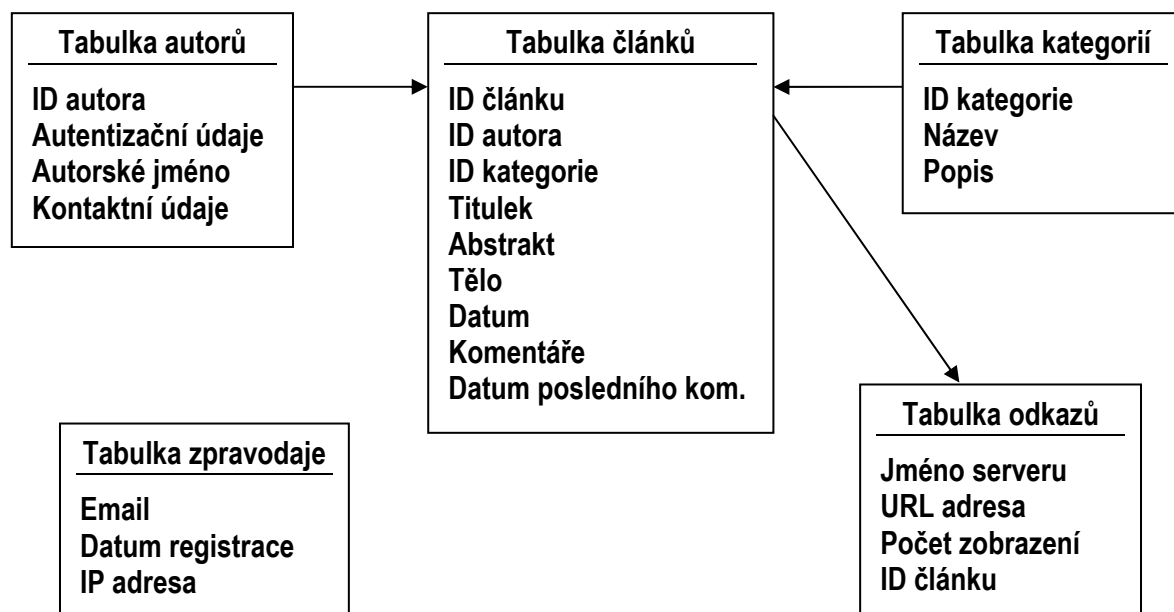
Příspěvky jednotlivých autorů je potřeba nějak členit. K tomuto poslouží seznam kategorií, kde budeme jednoduše evidovat pouze název a popis jednotlivých kategorií.

Seznam emailů pro rozesílání zpravodaje bude obsahovat samotnou emailovou adresu, datum registrace a IP adresu, ze které byla registrace provedena.

Posledními daty, které je potřeba zaznamenat jsou odkazy serverů a přístupy uživatelů k jednotlivým článkům. Ukládat tedy budeme jméno serveru, URL adresu, ze které bylo na článek odkázáno, počet zobrazení, které vedly z tohoto odkazu, a samozřejmě o který článek šlo.

Pro urychlení práce s databází se používají tzv. uložené procedury. Ale protože ještě nemáme zanalyzovány druhou a první vrstvu celé aplikace nemůžeme určit, které procedury by bylo vhodné vytvořit.

Výsledkem této analýzy je diagram zobrazený na následujícím obrázku. Tento diagram není závazný pro pozdější návrh, jeho smyslem je pouze poskytnout jednoduché schéma, které by v začátcích vývoje pomohlo.



Obr. 2. Analýza databáze – diagram ilustrující vztahy mezi daty

3.2 Uživatelské rozhraní

Požadavky na uživatelské rozhraní jsou především přehlednost, intuitivnost a rychlé načítání stránky do prohlížeče.

Samotné rozhraní by mělo jednoduše uživateli zobrazovat nově publikované články, kde u každého bude zobrazen název, abstrakt, kategorie, autor, datum publikování a počet příspěvku v diskuzi, s možností přejít přímo do diskuse.

Po kliknutí na název článku by se měl zobrazit tento článek celý včetně diskuse. Diskuse pod článkem bude zobrazovat pouze titulek autora a datum komentáře, s tím, že budou přístupné odkazy pro přesunutí přímo do diskuse, pro přidání komentáře a odkaz na RSS kanál diskuse.

V základním zobrazení by mělo být přístupné menu pro výběr kategorií a také funkční menu.

Po kliknutí na kategorii by se měly vylistovat do seznamu články publikované v této kategorii.

Kliknutím na autora článku se zobrazí seznam článků, které autor publikoval.

Další funkcí by mělo být přehledné vylistování článků dle zadaného časového období a kategorie.

Bude potřeba také stránka pro emailový zpravodaj a přihlašovací stránka pro autory.

Po přihlášení autora do systému by se mu mělo zpřístupnit redakční menu, díky kterému bude moci založit nový článek, vylistovat své články a editovat je. Také z tohoto menu by měl mít možnost šéfredaktor administrovat seznam kategorií, emailový seznam zpravodaje a seznam autorů. Přístupným pro všechny z tohoto redakčního menu by měla být statistika a správa souborů.

Díky zvolené technologii se využije uživatelských ovládacích prvků, které budou sloužit k zobrazení shodných částí uživatelského rozhraní. Typickým příkladem jsou jednotlivá menu nebo výpis komentářů - diskuze. K bližším informacím ohledně rozhraní se dostaneme až v kapitole návrhů grafického designu.

3.3 Aplikační objekty

Díky technologii ASP.NET má každá stránka vlastní třídu (je tedy objektem webového sídla) a v této třídě jsou definovány objekty nutné pro funkci stránky a uživatelských ovládacích prvků na ní použitých.

Je zde vyřešen např. přístup do databáze, dále zde najdeme validační logiku, autentizační, výpočtové a jiné funkce.

Stránky, resp. objekty stránek, však mohou využívat také objektů definovaných v samostatných třídách nebo objektů a uživatelských ovládacích prvků, které jsou součástí .NET Framework, příp. připojených objektů třetích stran.

4 NÁVRH PUBLIKAČNÍHO SYSTÉMU - Z POHLEDU SKLADOVÁNÍ DAT

Jak již bylo uvedeno, pro skladování a správu dat bude použit MS SQL Server díky němuž se oproti databázi v MS ACCESS podstatně zrychlí práce s daty. Existuje ještě tisíc dalších důvodů, jejichž výpis však není předmětem tohoto textu, přece jenom jde o čistokrevnou relační databázi.

Dle předchozí analýzy je již možné přistoupit k definování struktur jednotlivých tabulek. Tyto datové struktury jsou patrné z následujících obrázků:

Autori (PubSys)			
	Column Name	Condensed Type	Nullable
🔑	AID	int	NOT NULL
	Jmeno	varchar(20)	NOT NULL
	Prijmeni	varchar(30)	NOT NULL
	AutorskeJmeno	varchar(50)	NOT NULL
	Email	varchar(50)	NOT NULL
	ICQ	varchar(15)	NOT NULL
	Mobil	varchar(12)	NOT NULL
	Tel	varchar(12)	NOT NULL
	Fax	varchar(12)	NOT NULL
	Heslo	char(20)	NOT NULL
	PrihlasovaciJmeno	varchar(20)	NOT NULL

Obr. 3. Databázová tabulka Autori

Clanky (PubSys)			
	Column Name	Condensed Type	Nullable
🔑	CID	int	NOT NULL
	KID	int	NOT NULL
	AID	int	NOT NULL
	Datum	datetime	NOT NULL
	Titulek	varchar(100)	NOT NULL
	Abstrakt	varchar(500)	NOT NULL
	Telo	text	NOT NULL
	PocetKomentaru	int	NOT NULL
	Komentare	text	NOT NULL
	PosledniKomentar	datetime	NULL

Obr. 4. Databázová tabulka Clanky

Kategorie (PubSys)			
	Column Name	Condensed Type	Nullable
🔑	KID	int	NOT NULL
	Nazev	varchar(100)	NOT NULL
	Popis	varchar(500)	NOT NULL

Obr. 5. Databázová tabulka Kategorie

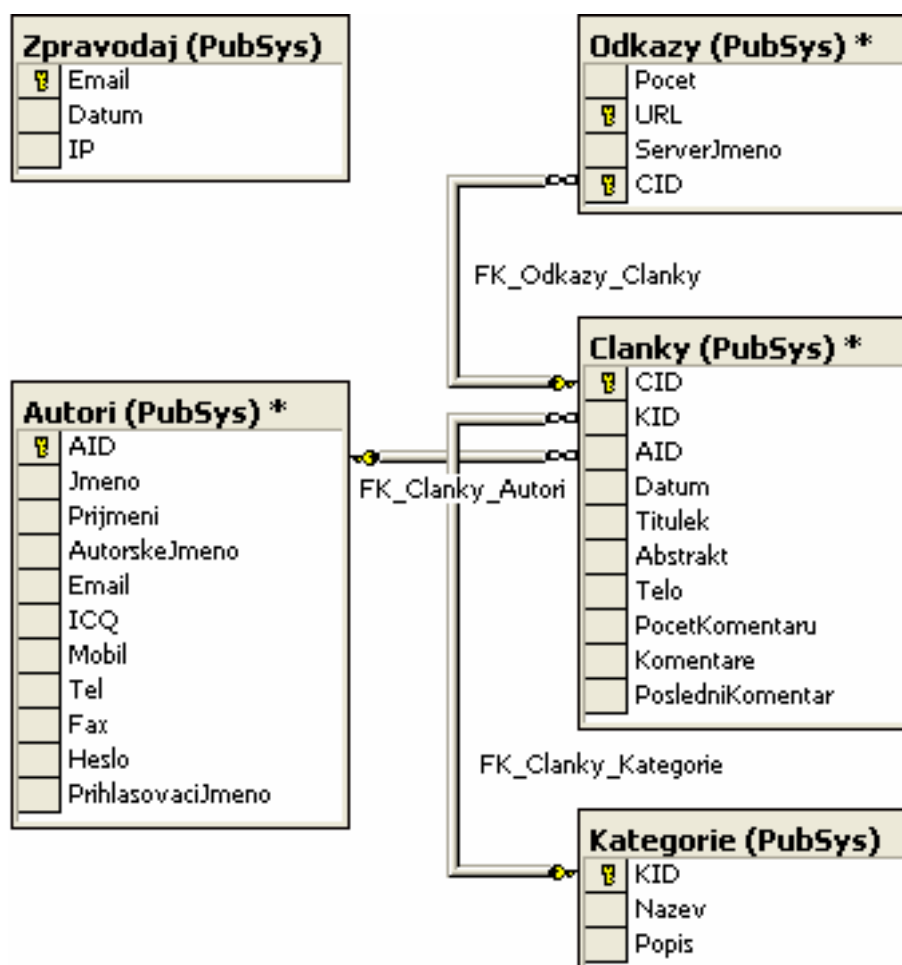
Zpravodaj (PubSys)			
	Column Name	Condensed Type	Nullable
🔑	Email	varchar(50)	NOT NULL
	Datum	datetime	NOT NULL
	IP	varchar(15)	NOT NULL

Obr. 6. Databázová tabulka Zpravodaj

Odkazy (PubSys)			
	Column Name	Condensed Type	Nullable
	Pocet	int	NOT NULL
	URL	varchar(200)	NOT NULL
	ServerJmeno	varchar(200)	NOT NULL
	CID	int	NOT NULL

Obr. 7. Databázová tabulka Odkazy

Dalším krokem po definici struktur je svázání dat pomocí relací. Vytvořené relace jdou vidět na následujícím obrázku.



Obr. 8. Relace databázových tabulek

Všimněte si, že tabulka zpravodaj není propojena s žádnou jinou tabulkou. Vychází to z logiky návrhu.

4.1 Vytvoření databáze

Existuje několik možností, jak tuto databázi vytvořit v SQL Serveru. Jednou z možností je vše dělat z administračního rozhraní SQL Server Enterprise Manager.

Toto rozhraní je velmi užitečné a získáme ho pouze při instalaci plnohodnotného SQL Serveru, ale v instalaci MSDE není obsažen.

V současnosti je ideální varianta pro vývojáře získat kompletní SQL Server pro účely vývoje zakoupením licence na SQL Server Developer Edition.

Přestože byl k dispozici tento software, rozhodnutí padlo pro rychlejší, přehlednější a znovu použitelnou variantu a tou je napsání SQL skriptu. [5] Tento skript je připojen k této práci jako příloha P I.

4.2 Uložené procedury

Uložené procedury jsou kusy kódu v SQL jazyku, které si zpracovává samotný SQL Server, který je na tuto činnost optimalizován. K těmto procedurám přistupujeme vložením vstupních údajů a jejich spuštěním. Podle kódu pak můžeme očekávat, nějakou činnost, která se provede v databázi, příp. tak získáme výstupní data.

Obrovskou výhodou uložených procedur je to, že můžeme uvnitř databáze jedním spuštěním procedury provést najednou více činností (např. načíst + upravit + přepočítat + uložit). Jednak se tyto činnosti díky optimalizaci vykonají v procedurách rychleji než kdyby jsme je řešili v aplikační logice, ale také by nás to stálo více připojení k databázi (konektů), které také požírají čas a hlavně výkon serveru.

V PS budou použity uložené procedury pro přidání a vymazání záznamu v tabulce zpravodaj protože je zde potřeba kontrolovat existenci záznamu. Další procedurou již trochu složitější je editace, případně přidání záznamu v tabulce odkazů; kde se již vyhodnocuje více údajů v závislosti na existenci dat v tabulce a vytváří se položka jméno serveru, který odkazoval na daný článek.

Tyto procedury byly sice laděny v SQL SE Manageru, ale jsou vloženy v "instalačním" SQL scriptu (viz příloha P I).

Do databáze je třeba přistupovat téměř s každým zobrazením stránky, a tak by zde mohlo být mnoho dalších procedur. Vzhledem k tomu, že většinou se jedná pouze o jednorázové vstupy nebo výstupy z tabulek a také pro jednodušší množnost editace, byly tyto SQL dotazy ponechány v kódu tříd objektů jednotlivých stránek. Převedení těchto dotazů do SQL procedur může být jedním z dalších úkolů vylepšení v následující verzi PubSys.

4.3 Instalační SQL script

Při spuštění tohoto scriptu proběhne nejprve vymazání existujících objektů v databázi, jako jsou tabulky, procedury a relační vazby.

Následně dojde k vytvoření tabulek poté k vytvoření indexů a relací.

Dalším bodem tohoto scriptu je vytvoření administrátorského účtu v tabulce autoři, díky kterému lze poprvé přistoupit do redakce PS, a vytvořit první kategorie a první autorské účty.

Posledním bodem scriptu je vytvoření uložených procedur.

4.4 Připojení a konfigurace databáze

Vzhledem k tomu, že bylo k dispozici vývojové prostředí MS Visual Studio.NET 2003 (dále jen VS) použil jsem toto prostředí nejen pro vývoj PubSys, ale také pro vytvoření a spuštění instalačního SQL scriptu.

Aby si VS rozumělo s databází je potřeba vytvořit databázové připojení, které se provádí jednak namapováním databáze PubSys v manažeru služby ODBC a přes toto namapování se poté VS připojí k databázi.

Byl zvolen tento postup: vytvoření nové databáze nazvané PubSys v SQL Serveru, dále v SQL Serveru vytvoření uživatele PubSys, který má k této databázi výhradní přístup (má přidělené právo db_owner). Po té bylo přes ODBC připojeno VS a spuštěn instalační script. Vytvořila se tím kompletní struktura potřebná pro PubSys.

5 NÁVRH PUBLIKAČNÍHO SYSTÉMU - Z POHLEDU GRAFICKÉHO DESIGNU

Ve vytýčení cíle je uvedeno, že vývoj PS by měl být zaměřen především na systém samotný a design PS by měl být jen testovací, v reálném nasazení pak lehce změnitelný pomocí CSS a konfiguračních souborů. Tento výrok však neznamena, že s návrhem uživatelského rozhraní nebude žádná práce, ba naopak.

5.1 Obrovský pomocník

Ten kdo nezná ASP.NET nepochopí proč je tato technologie tak převratná a oblíbená u programátorů. Práce s ASP.NET je spíše zábavou, protože programátor neztrácí čas s řešením běžných problémů mravenčí práce, jako u obdobných technologií, když chce např. zobrazit data z DB. Programový kód, který zobrazí databázovou tabulku může být vytvořen během minuty, a to si ještě můžeme vybrat z několika různých k tomu určených komponent - uživatelských ovládacích prvků. Na jednotlivých stránkách jsou proto využity již vytvořené komponenty (součást ASP.NET), které dokáží sofistikovaně na stránce zobrazit databázovou tabulku. Jsou to: *Repeater*, *DataGrid* a *DataList*.

Data z DB se do těchto prvků dostanou přes proměnnou *DataSource*, do které se pomocí funkce *DataBind* navážou data z DB, která je připojena přes některou z připojovacích funkcí např. *SqlDataReader*.

5.2 Uživatelské rozhraní

Dozrál čas k návrhu a vytvoření jednotlivých stránek dle předchozí analýzy. Ještě zmínka o tom, co bude všem stránkám společné a jak bude „layout“ navržen. Každá strana bude sestávat ze tří základních částí: hlavičky, menu a těla stránky. Aby byla alespoň trochu dodržena norma bude layout beztabulkový, tzn. že tyto tři hlavní bloky budou "DIVy" absolutně pozicované pomocí CSS. [7] Výhodou je i snadná realizace pomocí ASP.NET.

Všechny strany budou tedy obsahovat uživatelský ovládací prvek ***menu.ascx***, ve kterém je definována hlavička, boční menu a patička. V bočním menu tohoto prvku se využívá ještě jiný samostatný prvek ***list_kategorii.ascx***, ten velmi jednoduše pouze načte všechny kategorie s popisky z DB a zobrazí je, viz přílohy.

Jednou z funkcí menu, které si může všimnout pouze autor, je odkaz na přihlašovací stránku, který po úspěšné autentizaci zmizí a naopak se objeví v přibylém redakčním menu odkaz pro odhlášení.

Pozicování, barvy, rozměry, odsazení aj. vlastnosti prvků textů a téměř všeho na stránce bude v souboru kaskádových stylů *screen.css* a jeho obdobou pro případný tisk strany bude soubor *print.css*.

5.2.1 Titulní stránka: *default.aspx*

Titulní strana je ta, která se načte jako první po zadání adresy serveru do prohlížeče. Zpravidla má název *default* nebo *index*.

Na této straně jsou zobrazovány nejnovější články v systému PubSys. Pro jejich zobrazení, resp. jejich abstraktu a dalších údajů je použit prvek *DataList*. Ten u deseti nejnovějších článků zobrazí pro každý z nich jeho:

- nadpis (po prokliknutí nás přesune na zobrazení článku)
- abstrakt
- kategorie (po prokliknutí nás přesune do stránky zobrazující články v této kategorii)
- autora (po prokliknutí nás přesune do stránky zobrazující jeho články)
- datum publikování článku
- počet komentářů (které se zobrazí pouze jsou-li nějaké a pokud budou, tak po prokliknutí nás přesunou na stránku z komentáři k tomuto článku)

Navázání dat pro prvek *DataList* z databáze provedeme přes *SqlDataReader* a funkci *DataBind* v souboru třídy této stránky, při události *Page_Load*. Tento "datový scénář" je obdobný s různými úpravami i v ostatních stránkách.

5.2.2 Strana pro článek: *co.aspx*

Obsahem této stránky je výpis celého zvoleného článku, včetně doplňujících údajů (také proklikávacích), jako je kategorie, autor a datum - ten vypíše články ze stejného měsíce, viz příloha P XII.

Samozřejmostí je diskuse pod článkem, která je vytvořena pomocí uživatelského ovládacího prvku *list_komentaru.ascx*. Prvek zobrazuje komentáře s pomocí šablony *list_komentaru.xslt*. V tomto místě je odkaz pro přidání diskusního příspěvku, odkaz na čtení všech příspěvků a také na RSS kanál diskuse, viz příloha P IX.

Data se ve stránce zobrazí pomocí prvku *Repeater*, i přesto, že jde jen o jeden záznam, právě díky jednoduchému svázání s DB přes *SqlDataReader*.

Název souboru v prohlížeči však bude díky přepisu cesty jiný, o této věci však bude zmínka až u popisu konfigurace.

5.2.3 Diskusní strana: *komentare.aspx*

Diskuse pro daný článek bude zobrazovat hierarchicky jednotlivé příspěvky. Toho je dosaženo díky tomu, že datově je diskuse uložena jako XML a zobrazení je s pomocí šablony XSLT: *komentare.xslt* v už. ovl. prvku *xml*.

U každého komentáře se zobrazí jeho název, jméno autora komentáře (které je, je-li to povoleno, proklikávací a vyskočí nám tak okno mailového klienta), také datum vložení komentáře, webová stránka (pokud její adresu autor komentáře vložil) a odkaz na odpověď na tento komentář, viz příloha P X.

Druhá tvář tohoto souboru je pro autory a administrátora, v jejich zobrazení přibudou IP adresy jednotlivých autorů komentářů a také u každého příspěvku odkaz pro vymazání z diskuse. To se stane samozřejmě až po jejich přihlášení do systému. Na vzhledu pro autentizované osoby zpracovává šablona XSLT: *komentare_redakce.xslt*.

Stránce vévodí nadpis s titulkem článku, který je proklikávací na daný článek.

5.2.4 Strana pro přidání diskusního příspěvku: *komentar_novy.aspx*

Vložit příspěvek do diskuse touto stranou se Vám podaří pouze je-li zadáno ve formuláři vaše jméno, email a samotný text. Volitelnými parametry jsou: předmět komentáře, webová stránka autora příspěvku, možnost zveřejnění emailu ostatním diskutujícím a možnost nechat si zasílat na váš email zprávy o přidání odpovědi na váš komentář, podobně jako v příloze P XI.

Kontrola prvků formuláře je řešena pomocí některých validačních už. ovl. prvků z nabídky ASP.NET. Zde konkrétně jsou použity: *RequiredFieldValidator*

a *RegularExpressionValidator*. Zobrazování chybových zpráv provádí už. ovl. prvek *ValidationSummary*, který odchyty chyby jednotlivých validátorů a hromadně je zobrazí. V této stránce je umístěn nad formulářem.

Nahoře na stránce je nadpis s titulkem článku, který je proklikávací na daný článek.

Po odeslání komentáře dojde k validaci formuláře a odeslání emailů autorovi článku a také pokud se jedná o odpověď, autorovi předchozího komentáře, jestliže to povolil. Pak dojde k přesměrování na stránku komentářů daného článku.

5.2.5 Multifunkční chameleon *archiv.aspx*

Tento soubor má několik tváří, je totiž využíván k zobrazení seznamu článků dle kategorie (viz příloha P VI), seznamu článků v konkrétním měsíci (tento odkaz lze nalézt v hlavičce zobrazeného článku) a k funkci archivu. Této funkčnosti je dosaženo díky různým typům parametrů se kterými je tato stránka volána. Uživatel toto chování nepozná, protože název souboru v prohlížeči bude díky přepisu cesty jiný, o této věci však budu psát až u popisu konfigurace.

Seznam článků v této stránce je tvořen už. ovl. prvkem *DataSet*, který podle zadaných parametrů a toho co je potřeba zobrazit, naplníme daty událostí *Page_Load* v třídě objektu stránky pomocí *SqlDataReaderu*. Pro případ zobrazení stránky archivu se objeví nad seznamem článků panel s miniformulářem, v němž je seznam pro volbu měsíce a volbu kategorie.

Seznam je formátován a vypadá podobně jako ve stránce *default.aspx*, v těchto stránkách je však pro odlišení nadefinováno v souboru kaskádových stylů odlišné zobrazení pro každý druhý záznam a realizováno je jiným odstínem barvy. Seznam se tak stává přehlednější a lépe čitelný. Stránka *default.aspx* je sice podobná (vypisuje nejnovější články), mohla by tedy být realizována po úpravě také tímto souborem, ale předpokládá se, že titulní stránka webu bude obsahovat ještě mnoho jiných prvků, a tím by se zbytečně mnoho věcí zkomplikovalo.

5.2.6 Výpis článků autora: *autor.aspx*

Stránka je vnitřně téměř stejná jako předchozí *archiv.aspx*, pouze s odlišností v tom, že jako parametr se jí předává číslo autora a z databáze se vytahují články pouze konkrétního autora.

I tato stránka by mohla být jednou z tváří souboru *archiv.aspx*, ale protože se předpokládají v budoucnu další funkčnosti této stránky jako je např. zobrazení informací, příp. fotky autora či možnost mu poslat vzkaz (to z důvodu aby nebyly uveřejňovány emailové adresy autorů, server by ty zprávy pouze přeposílal), byl zvolen raději samostatný soubor.

5.2.7 Služba pro emailové zprávy: *zpravodaj.aspx*

Zpravodaj - stránka vypadá na první pohled (uživatele) velice jednoduše, je na ní totiž pouze políčko pro zadání emailu a tlačítko pro odeslání. V pozadí této stránky se však skrývá sofistikovaná kontrola a zařazení do seznamu emailů pro rozesílání zpravodaje.

Ve stránce jsou použity pro kontrolu vložených dat už. ovl. prvky: *RegularExpressionValidator* a *RequiredFieldValidator*. Je-li validace v pořádku odešle se na vloženou adresu email obsahující informační zprávu o požadavku na vložení do emailového zpravodaje a obsahující odkaz na tuto stránku s parametrem zakódovaným pomocí funkce *HashMD5*, která je pro tyto účely vytvořena v souboru *funkce.vb*, a která využívá funkcí zabudované kryptografie v ASP.NET. Je-li tato adresa žadatele, může kliknutím na odkaz v emailu aktivovat posílání zpravodaje na tuto adresu. Stane se tak pouze, je-li předaný bezpečnostní kód v pořádku, pak se tato adresa vloží do DB.

5.2.8 Přihlašovací stránka: *login.aspx*

I zde, za jednoduchým vzhledem můžeme najít zajímavé funkce. Funkcí stránky je přihlásit autory do redakčního, či administrativního módu. Děje se tak pomocí miniformuláře, kde autor zadá své uživatelské jméno a heslo, případně zaškrtně políčko pro zapamatování si těchto přihlašovacích údajů na jeho počítači a odešle, viz příloha P III.

Kontrola formuláře je opět řešena pomocí validačních komponent. Jsou-li zadány vhodné údaje, dojde k jejich ověření v DB a je-li vše v pořádku, je autor přihlášen do systému. Tato situace se projeví jednak tím, že se zobrazí zpráva o úspěšném přihlášení, ale také přibude redakční menu v levém sloupci layoutu.

Při přihlášení autora se uloží jeho ID do proměnných v *session*, přes kterou je připojen (pro pozdější identifikaci) a také se odstartuje autentizační timeout nakonfigurovaný v souboru *Web.config*. [6]

Tento soubor plní také funkci logout, tedy odhlášení autora, což je provázeno zprávou o úspěšném odhlášení.

5.3 Uživatelské rozhraní - část pro autory

Toto rozhraní je v podstatě stejné jako veřejné rozhraní pouze přibude v levé části redakční menu a díky němu několik užitečných funkcí - stránek pro autory a administrátora. Znovu připomínám, že tento redakční mód je přístupný pouze autentizovaným uživatelům, vztahuje se na všechny stránky z adresáře */redakce/*.

5.3.1 Vložení nového článku: */redakce/default.aspx*

První důležitou stránkou pro autora je tato stránka - Nový článek. První návštěva autora na této straně jistě velmi překvapí, protože kromě základních polí pro vložení titulku, abstraktu článku a rolovacího seznamu pro výběr kategorie (to vše je v horní části stránky), největší část plochy zabírá už. ovl. prvek vzhledem připomínající aplikaci MS Word XP, viz příloha P VII.

Editace těla nového článku probíhá ve zmíněné komponentě a autor má veškerý komfort, protože se může přepínat mezi WYSIWYG editorem a editorem HTML kódu. Je-li přepnut ve vizuálním módu má k dispozici většinu základních nástrojových lišt - panelů vyskytujících se v běžných textových editorech, inspirací byl především program MS Word XP. Tento prvek se jménem *FreeTextBox* je prací jiných autorů a bližší informace s downloadem aktuální verze (3.0.4) je možno ze stránek **www.freetextbox.com**. Použitá je verze 1.6, protože tato verze byla ještě k volnému použití (free) a také proto, že její chování je velice uspokojivé. Současná verze má sice přidáné některé další funkce, ale už je bohužel placená.

Nebude zde popis všech editačních možností této komponenty, předpokládejme, že ikonky v panelech jsou každému notoricky známé. Za zmínku stojí však jedna podstatná věc, ke které se vrátíme i později. Jedná se o úložiště souborů. Pokud totiž bude chtít autor přidat do své stránky obrázek nebo cokoli ke stáhnutí a bude-li to chtít uložit přímo v PS, má k dispozici adresář */soubory/*. *FreeTextBox* o existenci tohoto adresáře ví a zmáčkne-li autor ikonku pro vložení obrázku do stránky, nabídne mu prvek v novém okně obrázky z tohoto adresáře.

Podíváme-li se do vnitřností této stránky nalezneme několik nám již známých komponent jako jsou *TextBox*, *DropDownList* a validátory.

Zbývá již jen poznamenat, že nový článek se uloží do DB s číslem aktuálně přihlášeného autora a poté proběhne rozesílání zpravodaje, které poběží ve vlastním vlákně, tak aby toto odesílání (pokud těch adres bude hodně) nezatěžovalo server.

5.3.2 Sklad příloh: */redakce/spravce_souboru.aspx*

Tento jednoduchý souborový manažer se v podstatě stará pouze o upload souborů na server do PS, jejich uložení do adresáře */soubory/*, mazání těchto příloh a samozřejmě zobrazení obsahu tohoto adresáře, viz příloha P XIII.

V tabulce souboru (realizované prvkem *DataGrid*) se v řádcích k jednotlivým souborům zobrazí informace jako je název, velikost a datum poslední změny a také ikonka pro vymazání konkrétního souboru.

Po úspěšném uploadu na server se nám soubor objeví v tabulce souborů zařazen abecedně.

5.3.3 Články k editaci: */redakce/vyber_clanku_keditaci.aspx*

V tomto seznamu realizované prvkem *DataList* se vylistují články přihlášeného autora a autor má možnost tyto články editovat nebo mazat. Výpis vypadá obdobně jako např. výpis novinek pouze mu přibyli u titulku každého článku ikonky pro editaci a mazání.

V horní části strany je rozbalovací seznam realizovaný prvkem *DropDownList* pro výběr měsíce, ve které byl článek napsán, aby tak byl seznam přehlednější.

Zajímavostí této strany je to, že se zobrazí pouze články autentizovaného autora, ne jiných autorů. Výjimkou pro tuto funkci je pouze administrátor, kterému se zobrazí články všech autorů a má právo editace i mazání u všech článků. Další vlastností je také to, že již jednou vložený článek do DB nemůže odstranit ani autor pouze administrátor - šéfredaktor.

5.3.4 Stránka pro editaci článku: */redakce/editace_clanku.aspx*

Je volána ze seznamu článků k editaci s číslem konkrétního článku co by parametrem a vzhledem téměř stejná jako */redakce/default.aspx* (určená pro vytvoření nového

článku). Zde se však do formuláře načtou již existující údaje z DB a po nezbytné validaci, dojde ke změně těchto údajů v DB.

5.3.5 Administrace seznamu kategorií: */redakce/kategorie.aspx*

V této stránce (jejím formuláři) lze provést změnu názvu, či popisu jednotlivých kategorií a samozřejmě vytvoření nové. Stránka, v jejíž spodní části je seznam existujících kategorií, se sice zobrazí všem autorům, ale změnit, či smazat může kategorii pouze administrátor, a to pouze tehdy, není-li do ní přiřazen nějaký článek, viz příloha P VIII.

Autoři mohou pouze přidat novou, která se po úspěšné validaci prvků objeví ve stávajícím seznamu. Ten je tvořen prvkem *DataGrid* a v řádcích vypisuje ke každé kategorii její název, popis a nezbytné jsou ikonky pro editaci a mazání.

5.3.6 Seznam emailů: */redakce/zpravodaj.aspx*

Jedná se opravdu jen o seznam pro rozesílání zpravodaje, realizovaný prvkem *DataGrid*, zobrazující email, datum vložení do listu, IP adresu z které tak bylo učiněno a ikonku mazání. Ta je však potřebná jen administrátorovi, protože pouze on má právo odstranit daný email z tohoto seznamu. Je-li úspěšný, projeví se to ihned v seznamu. Emaily v tomto seznamu jsou proklikávací, tzn. otevřou okno aktuálního mailového programu uživatele pro novou zprávu, kde tato emailová adresa bude v políčku příjemce.

5.3.7 Členové redakčního týmu: */redakce/autori.aspx*

Ač na to možná vzhledem nevypadá, je tato stránka jednoznačně největší svým kódem, ať už *aspx* soubor, tak i soubor třídy této stránky.

Účel plní hned několikrát, slouží pro vytvoření nového účtu autora, editaci, příp. vymazání stávajícího účtu a zobrazení všech autorů i s jejich kontaktními údaji, kromě administrátorského účtu.

Stránka je rozdělena do tří částí, viz příloha P IV. Vezmeme-li to z vrchu, pak první je panel formuláře s kontaktními údaji, kde je možnost zadat jméno, příjmení, autorské jméno (to je to, co se bude zobrazovat veřejně), email a číslo: ICQ, mobilu, telefonu a faxu.

Další panel je autentizační, v něm lze vyplnit položky: přihlašovací jméno, heslo a ověření hesla. Pro případ editace stávajícího záznamu je zde zaškrťovací políčko, že není

třeba měnit autentizaci. To pro případ, že by chtěl např. admin (autoři si své údaje navzájem měnit nemohou - pouze své vlastní), změnit některému autorovi něco v kontaktech, aby mu nezměnil heslo. Heslo se totiž kvůli bezpečnosti nezobrazuje a tak by musel vložit kvůli validaci jiné a přepsal by tak to původní.

V třetí části je seznam autorů s jejich kontakty, tvořený prvkem *DataGrid* a kromě dat také zobrazující již známé ikonky pro editaci a vymazání autora. Emaily v tomto seznamu jsou proklikávací, tzn. otevřou okno aktuálního mailového programu uživatele pro novou zprávu, kde tato emailová adresa bude v políčku příjemce.

Logika je zde taková, že autor může pouze změnit své údaje, nebo svůj účet odstranit, ale jinak nikoho jiného. Administrátor pak může měnit údaje všem a také autory mazat. Mazání autorů v obou případech je podmíněno tím, že autor nesmí mít publikovaný jediný článek. Odstranit účet administrátora není možné a jeho editace není přístupná díky nepřítomnosti v tabulce autorů.

Po vytvoření, či změně účtu, je-li vše v pořádku, se toto projeví ihned v seznamu autorů. Validaci provádí mnoho validačních prvků typu: *RequiredFieldValidator*, *RegularExpressionValidator*, *CompareValidator* a centrálně vše vyhodnocuje *ValidationSummary* - ten případné nedostatky zobrazí nad panely formuláře.

5.3.8 Informace o přístupech a odkazech: [/redakce/statistika_zajmu.aspx](/redakce/statistika_zajmu.aspx)

Tato strana jak název napovídá je věnovaná statistikám a přístupná je všem autentizovaným osobám. Je rozdělena do tří částí, se seznamy tvořenými 4 prvky *DataGrid*. V horní části je miniformulář pro zadání max. stáří článků počítaných do statistiky ve dnech a letech. Toto se projeví pouze u prvních dvou částí (první dva seznamy).

První statistika je určena pro sledování komentářů u článků a vyhodnocuje nejkomentovanější články. Zobrazí pak jejich název (proklikávací na konkrétní článek), datum vytvoření článku, počet komentářů, datum posledního komentáře a rozdíl mezi těmito daty ve dnech.

Statistika 2 ukazuje nejodkazovanější články na serverech. V seznamu pak nalezneme opět proklikávací název článku s datumem publikování, počet zobrazení u daného odkazu pro daný server a název serveru po jehož prokliknutí se dostaneme, přímo na URL adresu, na které je odkaz na náš článek.

Ve třetí části této stránky je obdoba statistiky č. 2, s tím rozdílem, že jsou zde v prvním seznamu servery s největšími celkovými počty odkazů. Zobrazuje se celkový počet a jméno serveru, jehož prokliknutím můžeme sledovat podrobnou statistiku konkrétního serveru. Ta se zviditelní pod tímto seznamem jako čtvrtá tabulka s detaily ke konkrétním nejodkazovanějším článkům v duchu předchozích tabulek.

5.3.9 Pomůcka administrátora: */redakce/komentar_smazat.aspx*

Již bylo uvedeno v povídání o komentářích u článků, že autentizovaným osobám se zobrazí více informací a mimo jiné odkaz pro odstranění komentáře z diskuse. Ten má význam především proto, že někteří lidé jsou nevychovaní a tak je zde možnost jejich komentáře mazat. Stará se o to tato stránka, která po úspěšném odstranění komentáře přesměruje zpět do diskuse, jinak zobrazí chybovou zprávu.

5.3.10 Informace o chybě: *chyba.aspx*

Postavit se k chybě ve webové aplikaci lze několika způsoby: mít vypnuté chybové rozhraní (dělat mrtvého brouka), dát k dispozici kompletní informace o chybě (což není pro nikoho příjemné a je to také nebezpečné kvůli vyzrazení možná i důležité části kódu ve výpisu) a poslední možností je chybové rozhraní vytvořit, což byla jasná volba.

Tato stránka se objeví nastane-li jakákoliv chyba v PS, díky nastavení v souboru *Web.config*. O detailech tohoto nastavení se zde nemá význam rozepisovat.

Vyhodnocuje se pouze chyba 404 - neexistující dokument a chyba 500 - interní chyba serveru (projeví se zde i chyba DB, či SMTP serveru, dokonce i to, když bude nefunkční nějaký email a vrátí se SMTP serveru jako nedoručený). Ostatní chyby spadnou dle logiky serveru buď do chyby 500, nebo do blíže neurčené chyby. Tyto tři stavy jsou v souboru ošetřeny.

Ve všech případech dojde k poslání emailu na určenou adresu správce webu, která je uložena ve zmíněném konfiguračním souboru a obsahem je podrobný výpis informující o stavu aplikace.

5.3.11 Ještě něco málo ke grafice

O tom, že vizuálně lze téměř všechny prvky stránek přenastavit pomocí CSS již bylo psáno. Zmíněné soubory se nalézají v adresáři */grafika/*.

Zde také nalezneme veškeré obrázky použité v této aplikaci a podadresář */freetextbox/*, v němž jsou soubory pro prvek *FreeTextBox*.

6 VYTVOŘENÍ FUNČNÍ APLIKACE PUBLIKAČNÍHO SYSTÉMU

PS je vzhledově navržen a vytvořen, zbývá doplnit vnitřnosti a několik dalších důležitých souborů a nastavení. V technologii ASP.NET, je výkonný kód oddělen od designu stránky tak, že každá stránka má svou třídu, ve které je definováno co a jak má objekt stránky dělat. Nejvyužívanější metodou - událostí u všech tříd stránek je bezpochyby *Page_Load*. V ní se často iniciují prvky ve stránce, natahují data z DB, aj. činnosti potřebné před zobrazením stránky.

V této kapitole bude popsáno několik prozatím nezmíněných věcí. Bude to konfigurace webu, kanály RSS a umístění souborů.

6.1 Konfigurace PubSys

Každý web vytvořený v ASP.NET má několik možností konfigurace. Jako hlavní soubor pro nastavení webu slouží ***Web.config***. Tento soubor je typu XML a v jednotlivých klíčích lze nastavit různé parametry podle kterých se bude celý web chovat, např. nastavení chybových stránek, nastavení jazykové, autentizační, autorizační aj. Zde mimo jiné také nastavíme chování webu pro situaci, kdy se k němu připojí prohlížeče postavené na jádru GECKO, jako např. Mozilla. ASP.NET totiž tyto prohlížeče nezná a proto je vyhodnotí jako nízkourovňové prohlížeče. [6]

Dalším důležitým společným souborem je ***Global.asax.vb***, v kterém můžeme nadefinovat chování webu podle událostí, které na něm probíhají, jako např. pro zachování se při chybě (odešle email administrátorovi) při požadavku o nějakou stranu, startu a ukončení webové aplikace, či jednotlivé session. Vzhledem k tomu, že pro uživatele i pro katalogizační a vyhledávací servery je dobré, když číslo článku, nebo jiný parametr stránky je obsažen v názvu souboru stránky a ne za tímto názvem jako parametr (čtenář jistě zná ty dlouhé řetězce na některých serverech), v tomto souboru vytvoříme mapování - přepisování názvu a cesty souboru. Díky tomu můžeme například na článek č. 26 odkazovat odkazem */co/clanek-26.aspx*, obdobně i u kategorií, autorů a archivu.

Poslední soubor, neméně důležitý je ***AssemblyInfo.vb***. V něm můžeme nadefinovat některé důležité nastavení pro výslednou *Assembly*, která se po kompilaci webu vytvoří v adresáři */bin/* jako DLL soubor. Zde se nastavuje např. číslování verzí, název a popis produktu, výrobce, copyright, aj.

6.1.1 Připojení k databázi

Je potřeba, abychom se jednoduše, kdykoliv to bude potřeba, připojili k DB. Nejprve si v souboru **Web.config** vytvoříme připojovací řetězec, tzv. *ConnectionString*. To provedeme tak, že ho jednoduše přidáme do sekce */configuration/appSettings/* (kde budou i další naše nastavení). Díky .NET Frameworku je možné se kdykoliv připojovat těsně před použitím a odpojovat po použití, což není nejlepší metoda u jiných systémů, ale zde se o *connection pooling* (aby se spojení stále neotevíralo a nezavíralo) stará databázová vrstva .NET Frameworku.

Jelikož se kód pro otevření DB stále opakuje, je metoda *OpenDB* ve veřejné třídě **funkce.vb** a lze ji volat kdekoliv v kódu.

6.1.2 CACHEování menu

Část menu obsahuje seznam kategorií, ty jsou uloženy v DB. Pokud bychom nepoužili *CACHE*, znamenalo by to, že každý dotaz na stránku způsobí jeden dotaz do databáze, který bude vracet stále stejné výsledky. Seznam kategorií nebude měnit nějak extrémně často. A pokud už se změní, pak není nijak zásadní, aby se změna projevila okamžitě. Můžeme tedy s výhodou využít cacheování.

ASP.NET umožňuje uložit výsledek celé stránky nebo její části (už. ovl. prvku) do *CACHE*. Proto do prvku **list_kategorii.ascx** nastavíme direktivu *@OutputCache* na 600s, tj. 10 minut. Kategorie se natáhnou poprvé z DB a veškeré další dotazy pak půjdou během následujících deseti minut z *CACHE*.

6.2 RSS kanály

Jsou technologií poslední doby. Na různých webech je možnost zahlédnout často nenápadný odkaz RSS či XML, někdy RSS feed. Je možné RSS nazvat fenoménem.

O co jde? Jednoduše o to, že máte na PC nainstalovaný prohlížeč RSS (použit byl jednoduchý a zároveň plně dostačující, **FeedReader**, k němuž existuje i čeština, viz **www.feedreader.com**) a v něm zadány adresy RSS kanálů jednotlivých serverů, kam chodíte. Prohlížeč vypadá většinou jako emailový klient - zobrazuje u daných zdrojů nadpisy článků s dalšími údaji: datum, abstrakt, atd. Lze ho nastavit aby sledoval novinky na serverech v určité periodě, např. jednou za hodinu. Je to zkrátka takové informační centrum, či správce informací. Již nemusíte pravidelně obcházet své oblíbené zdroje,

informace o aktualizaci přijdou samy přes RSS. Tímto se tedy stává emailový zpravodaj nepotřebným pro ty, co používají RSS. Čtenář, kterého zaujme nadpis článku, přečte abstrakt, je-li k dispozici (bývá to pravidlem) a pokud ho to upoutá, klikne na odkaz článku a zobrazí si tak kompletní článek (to se již otevře celá stránka na zdrojovém serveru).

Soubory pro RSS jsou sice vnitřně typu XML, ale na webu se setkáme s různými příponami. PubSys generuje RSS co by aspx stránky - samozřejmě s XML uvnitř, dle poslední normy RSS 2.0. [3]

6.2.1 RSS kanál článků: */rss/rss_clanky.aspx*

Tento soubor nabídne 20 nejnovějších článků publikovaných aplikací PubSys. Objeví se samozřejmě všechny potřebné údaje. Stránka má opět 3 soubory.

6.2.2 RSS kanál komentářů: */rss/rss_komentaru.aspx*

Je-li diskuse u nějakého článku zajímavá, má čtenář možnost tuto diskusi sledovat právě pomocí tohoto RSS. Parametrem se předá číslo článku. Strana má zase 3 soubory a využívá šablonu *komentare_rss.xslt*.

6.3 Co kde leží

V kořenovém adresáři webu nalezneme:

- již popsané konfigurační soubory (Web.config, Global.asax, Global.asax.resx, Global.asax.vb, AssemblyInfo.vb),
- soubory všech volně dostupných stránek (každá má tři soubory s příponami **.aspx*, **.aspx.resx*, **.aspx.vb*, kde první je soubor stránky s HTML kódem, druhý je XML schéma a konečně třetí je již zmiňovaná třída stránky, tedy veškerý výkonný kód provádějící se na serveru),
- adresáře, které budou následně popsány,
- systémové soubory pro otevření projektu v MS Visual Studiu .NET 2003 (*PubSys.sln*, *PubSys.suo*, *PubSys.vbproj*, *PubSys.vbproj.webinfo*), podmínkou u nich však je (ostatně i pro správnou funkčnost PubSys), že kořenový adresář musí být zároveň hlavním kořenovým adresářem webu v případě WinXP,

- soubor *favicon.ico*, který ač by měl patřit do adresáře s grafikou musí být zde, protože to tak norma vyžaduje (příp. prohlížeče).

6.3.1 Složka */aspnet_client/*

Je systémová a je v ní adresář */system_web/* a */FreeTextBox/* s pomocnými soubory.

6.3.2 Složka */bin/*

Obsahuje soubor *freetextbox.dll* - prvek pro editaci článků, *PubSys.dll* a *PubSys.pdb* v nichž je zkompileována veškerá funkčnost webu PubSys.

6.3.3 Složka */DB PubSys/*

Patří DB projektu (je jedním z projektů zahrnutých do PubSys *solution*), kterým se dá vytvořit struktura DB pomocí scriptu uvnitř.

6.3.4 Složka */grafika/*

Uvnitř jsou veškeré soubory podstatné pro vzhled webu: CSS soubory, obrázky, aj.

6.3.5 Složka */komponenty/*

V této složce jsou umístěny už. ovl. prvky (každý má tři soubory s příponami **.ascx*, **.ascx.resx*, **.ascx.vb*, s podobným obsahem jako u stránek), *funkce.vb* - skrývá třídu funkce, kde je pár často používaných, společných metod (*SendMail*, *OpenDB*, či *HashMD5*), *zpravodaj.vb* - třída zpravodaj s jedinou metodou *Send* určená pro rozesílání emailového zpravodaje.

6.3.6 Složka */redakce/*

K jejímu obsahu vede přístup v prohlížeči jen přes úspěšnou autentizaci, obsahuje stránky potřebné pro autory a administrátora (opět má každá stránka 3 soubory).

6.3.7 Složka */rss/*

Jak název napovídá, zde můžeme hledat soubory pro RSS. Tyto soubory jsou záměrně ve své vlastní složce, protože jsou typu XML. Do ostatních stránek, kromě tohoto

adresáře se přidává definice dokumentu "<!DOCTYPE HTML", která by dle normy měla být na prvním řádku stránky. Přidávání probíhá programově díky ***Global.asax.vb***.

6.3.8 Složka */sablony_xslt/*

Tato složka je pouze pro přehlednost a jejím obsahem jsou pouze již zmíněné XSLT šablony: ***komentare.xslt***, ***komentare_redakce.xslt***, ***komentare_rss.xslt***, ***list_komentaru.xslt***.

6.3.9 Složka */soubory/*

Jen pro přesnost uvádím zde i tuto složku. Jak jsem již uvedl dříve, je používána systémem pro ukládání obrázků a ostatních příloh pro stažení.

6.4 Použitý software

Tato kapitola je důležitá pro představu, jaký software byl použit při vývoji tohoto systému.

Vše bylo vyvíjeno na OS Windows XP Professional SP2, jehož součástí po doinstalování je IIS Server a SMTP Server (standardně není v instalaci OS zahrnut).

Jako vývojový nástroj posloužilo Visual Studio .NET 2003. Zajímavou alternativou zdarma může být, pro čtenáře této práce, vývojové prostředí s WYSIWYG editorem The ASP.NET Web Matrix (adresa je <http://www.asp.net/webmatrix/>). Samozřejmě je mít nainstalovaný .NET Framework SDK, který je zdarma.

Databáze běžela na MS SQL Serveru 2000 Developer Edition, možné by však bylo použít MSDE, které je rovněž zdarma.

RSS bylo testováno v programu FeedReader, který lze stáhnout zdarma na adrese <http://www.feedreader.com>.

Velmi užitečným pak byl freeware: EasyIIS verze 1.7, díky kterému lze provozovat i na Windows XP více webových serverů. Je sice nutné přepínání, ale pro vývoj to stačí (nalézt ho lze na adrese: <http://www.codeproject.com/w2k/EasyIIS.asp>).

7 ROZŠÍŘENÍ SYSTÉMU O DALŠÍ DOPLŇUJÍCÍ FUNKCE

V této kapitole bude popsáno pár nápadů a cílů pro vylepšení, které bude v dalších verzích publikačního systému PubSys. Z různých důvodů nejsou obsaženy již v této verzi. Každý systém se vyvíjí a stejně tak i tento. Důležité je (po vzoru velkých výrobců SW), mít v šuplíku neustále nové nápady a cíle pro vylepšení. Takže zde jsou:

1. Zdokonalená autorizace - autoři by články předkládali ke korektuře a ke schválení šéfredaktorům. Teprve po schválení by se v předurčeném datumu publikovaly, od té chvíle by již autoři neměli právo editace těchto článků. V návaznosti by mohly být další funkce podporující odměňování autorů či vykupování článků.

2. Reklamní systém - samostatná soustava uživatelských ovládacích prvků umisťujících a střídajících reklamní bannery podle určených pravidel, počítajících jednotlivá zobrazení, atd.

3. Ankety, soutěže - modul sestavený z už. ovl. prvků pro ankety a soutěže, zpracovávajících jejich data, se statistickým vyhodnocením.

4. WAP - doplnit a doladit podporu drobných mobilních zařízení.

5. Vyhledávání - jedno základní přímo vložené pod menu každé stránky, druhé podrobné. V tomto rozšířeném vyhledávání zahrnuté funkce volby pozice (v nadpisu, popisu, v článku), volby kategorií (všechny nebo výběr některých), volbu autora (všech, nebo některých a volbu dle data (určitý den, měsíc, rok nebo rozsah datumů od - do).

6. SQL dotazy - optimalizovat (změřit) doby vykonávání, případně převést do uložených procedur v SQL serveru.

7. Podpora seriálů - číslování dílů, v každém dílu odkazy na ostatní díly, podpora seriálů v souhrnných stránkách.

8. Chytré výpisy souhrnných stran - uživatelsky nastavený počet článků pro souhrnné strany, listování v těchto stranách po nastaveném počtu.

9. Přepis cesty - dodělat tuto funkci i pro další stránky např. komentáře ke článku.

10. Informace o autorovi - každý autor si nastaví co vše chce o sobě zveřejnit. Tyto údaje by se pak vyobrazily při výpisu článků dle autora v horní části stránky. Bude potřeba přidat další položky jako fotku, web autora, aj.

11. Správce souborů - nyní je úložiště souborů společné pro všechny autory, což není právě ideální. Je potřeba zavést systém přejmenování souborů na fiktivní jména a v DB vést informace o těchto jménech a autorech, kterým patří a pro webové zobrazování provádět přepis jména. Toto bude i výhoda ochrany před „script injections“.

12. Vzhled na míru - uživatel si některé parametry či barvy PubSys nastaví a do budoucna se právě pro něho budou zobrazovat s jeho nastavením.

ZÁVĚR

Cílem této práce bylo vytvoření publikačního systému PubSys, se zaměřením především na jeho funkčnost a jednoduchost z pohledu uživatele. Od samého začátku byla předurčena k tomuto úkolu platforma firmy Microsoft s využitím jejich serverů a technologií.

Již na tomto místě se řada čtenářů této práce zastaví a rozdělí na dvě slupiny: příznivce a nepřátele gigantu – softwarové firmy Microsoft. Nehodlám zde vyvracet vaše názory a popíši ten svůj. Patřím, jak každému asi došlo, spíše mezi příznivce těchto technologií, i přesto, že jsem si vědom různých chyb této firmy v oblasti SW i licenční a cenové politiky s pohledu uživatele. To mě přivádí k myšlence, že tato firma je fenomenálně úspěšná především v obchodování, reklamě a prodeji svých výrobků, ale už ne tolik v jejich kvalitě. Shrnu to tak, že jako programátor s desetiletou praxí chápu, že napsat SW bez jediné chyby snad ani není možné. Další podstatnou věcí je množství resp. poměr uživatelů této platformy vůči konkurentům, to určuje i mé zaměření a to, kde naleznou nejvíce potencionálních zákazníků.

Jsem samostatný vývojář - podnikatel, proto vlastním licence na všechny využití technologie. To samozřejmě nebylo nejlevnější a tak hodlám PubSys, v dalších verzích, komerčně využít. Z toho plyne to, že nemohu udělit naší univerzitě, či její fakultě volnou licenci pro plné zveřejnění, tzv. "ihned". Pro výukové účely nabízím zveřejnění této práce po 2 letech. Díky posunu ve vývoji MS technologií (již teď vím o plánu ASP.NET verze 2.0), tento systém nebude v současných – budoucích technologiích aktuální. Přesto se domnívám že použitelný, protože zase tak rychle tyto technologie nestárnou – soudě podle toho, že ještě dnes se podporuje zastaralé, obyčejné ASP ve verzi 3.0.

Data v DB a na obrázcích jsou zkušební. Jedná-li se o články, jsou převzaty z různých informačních zdrojů, např. www.IDNES.cz, či www.ZIVE.cz, což je vždy uvedeno, včetně datumu publikace originálu i autora.

Praxe - reálné nasazení PubSys teprve ukáže kvalitu návrhu, při kterém jsem čerpal především ze svých zkušeností, ověřených technik a postupů, které původně vzešly z různých knih, internetových článků a diskusních fór. Ve většině případů již nedokážu s jistotou tvrdit co mám z kterého zdroje. Jen má knihovna již obsahuje tolik odborných knih, že vyjádřeno součtem pořizovacích cen přesahuje 30 tisíc Kč, což myslím, že i při dnešních vysokých cenách knih je úctyhodná suma.

Ač jsem přečetl mnoho knih aj. literatury, ještě jsem neměl tu čest veřejně publikovat nějaký dokument, proto se váženému čtenáři této práce omlouvám za veškerá rozčarování, které jsem mu možná nechtěně připravil. Při tvorbě jsem postupoval podle zadání bakalářské práce, předběžných požadavků na publikační systém a svého nejlepšího svědomí a vědomí. Systém byl vytvořen a otestován na počítačích a síti mé firmy. Domnívám se, že splnil veškeré požadavky a stanovené cíle.

SEZNAM POUŽITÉ LITERATURY

- [1] PAYNE, CH. Naučte se ASP.NET za 21 dní, Computer Press, Praha, 2002. ISBN 80-7226-605-5
- [2] Kačmář, D. Programujeme .NET aplikace ve Visual Studiu .NET, Computer Press, Praha, 2001. ISBN 80-7226-569-5
- [3] RSS at Harvard Law, RSS 2.0 Specification [online], Dostupný z WWW:
<<http://blogs.law.harvard.edu/tech/rss/>>
- [4] Laurent, S. Tvorba internetových aplikací v XML, Computer Press, Brno, 1999. ISBN 80-7226-170-3
- [5] Šimůnek, M. SQL Kompletní kapesní průvodce, GRADA Publishing, 1997, ISBN 80-7169-692-7
- [6] Avery, J. Microsoft ASP.NET Konfigurace a nastavení, Kapesní rádce programátora a administrátora, Computer Press, Brno, 2004, ISBN 80-251-0121-5
- [7] Staníček, P. CSS Kaskádové styly, Kompletní průvodce, Computer Press, Praha, 2003, ISBN 80-7226-872-4
- [8] Flanagan, D. JavaScript, Kompletní průvodce, Computer Press, 1998, ISBN 80-7226-093-6
- [9] Bureš, J. RSS 2.0 [online], XML, Dostupný z WWW:
<<http://interval.cz/clanek.asp?article=3567>>
- [10] Valášek, M. FreeTextBox: WYSIWYG editor pro ASP.NET [online], Tipy a triky, Dostupný z WWW: <<http://www.aspnet.cz/entry/article-20050421.aspx>>
- [11] McFarland, S. Caching with ASP.NET [online], Dostupný z WWW:
<<http://aspnet.4guysfromrolla.com/articles/022802-1.aspx>>

SEZNAM POUŽITÝCH SYMBOLŮ, ZKRATEK A POJMŮ

API	Application Programming Interface
ASP	Active Server Pages
ASP.NET	Název technologie firmy Microsoft, nová generace ASP
Autentizace	Rozpoznání – ověření identity (např. při přihlášení do systému)
Autorizace	Přidělení určitých práv pro různé úlohy
CLR	Common Language Runtime
COM	Component Object Model
CSS	Cascading Style Sheets
DB	Databáze
DHTML	Dynamic HyperText Markup Language (Dynamic HTML)
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
ICQ	Komunikační služba vytvořená firmou Mirabilis, název vytvořen dle vyslovení „I Seek You“
IIS	Internet Information Server <i>nebo také</i> Internet Information Services
IL	Intermediate Language
IP	Internet Protocol
JIT	Just In Time
Kerberos	Bezpečnostní systém, založený na kryptografii symetrických klíčů
MFC	Microsoft Foundation Classes
MSDE	Microsoft SQL Server 2000 Desktop Engine
NTLM	Windows NT® LAN Manager – autentizační metoda
ODBC	Open DataBase Connectivity
OS	Operační systém
PS	Publikační Systém

RAD	Rapid Application Development
RDF	Resource Description Framework
RSS	Rich Site Summary <i>nebo</i> RDF Site Summary <i>nebo</i> Real Simple Syndication <i>nebo také</i> Really Simple Syndication
SMTP	Simple Mail Transport Protocol
SQL	Structured Query Language
SW	Software
VB	Programovací jazyk Visual Basic
VC++	Programovací jazyk Visual C++
VS	Microsoft Visual Studio.NET 2003
WAP	Wireless Application Protocol
WWW	World Wide Web
WYSIWYG	What You See Is What You Get
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

Tento seznam byl doplněn s pomocí serveru **www.google.com** (kde se do vyhledávacího pole vložila direktiva "define:" a za ní vypsál pojem.

SEZNAM OBRÁZKŮ

Obr. 1. Struktura .NET Framework a navazující jazyky s vývojovým prostředím	16
Obr. 2. Analýza databáze – diagram ilustrující vztahy mezi daty	21
Obr. 3. Databázová tabulka Autori	23
Obr. 4. Databázová tabulka Clanky	23
Obr. 5. Databázová tabulka Kategorie	24
Obr. 6. Databázová tabulka Zpravodaj	24
Obr. 7. Databázová tabulka Odkazy	24
Obr. 8. Relace databázových tabulek	25

SEZNAM PŘÍLOH

- P I : Instalační SQL script
- P II : Výřez okna prohlížeče – prázdný systém po prvním spuštění
- P III : Výřez okna prohlížeče – přihlášení do redakce
- P IV : Výřez okna prohlížeče – stránka autorů
- P V : Snímek okna prohlížeče – nejnovější články
- P VI : Snímek okna prohlížeče – výpis jedné kategorie
- P VII : Snímek okna prohlížeče – nový článek
- P VIII : Snímek okna prohlížeče – kategorie
- P IX : Výřez okna prohlížeče – příklad diskuse – prvek
- P X : Výřez okna prohlížeče – příklad diskuse – výpis
- P XI : Výřez okna prohlížeče – nový komentář k článku
- P XII : Výřez okna prohlížeče – detail hlavičky článku
- P XIII : Výřez okna prohlížeče – správce souborů

Příloha P I : Instalační SQL SCRIPT

```
1  /*-----*/
2  /*--- Smazání existujících objektů ---*/
3  /*-----*/
4  IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[FK_Clanky_Kategorie]') and OBJECTPROPERTY(id,
    N'IsForeignKey') = 1) ALTER TABLE [PubSys].[Clanky] DROP CONSTRAINT
    FK_Clanky_Kategorie
5  IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[FK_Clanky_Autori]') and OBJECTPROPERTY(id,
    N'IsForeignKey') = 1) ALTER TABLE [PubSys].[Clanky] DROP CONSTRAINT
    FK_Clanky_Autori
6  IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[odkaz_pridej]') and OBJECTPROPERTY(id,
    N'IsPROCEDURE') = 1) DROP PROCEDURE [PubSys].[odkaz_pridej]
7  IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[zpravodaj_pridej]') and OBJECTPROPERTY(id,
    N'IsPROCEDURE') = 1) DROP PROCEDURE [PubSys].[zpravodaj_pridej]
8  IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[zpravodaj_odeber]') and OBJECTPROPERTY(id,
    N'IsPROCEDURE') = 1) DROP PROCEDURE [PubSys].[zpravodaj_odeber]
9  IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[Autori]') and OBJECTPROPERTY(id,
    N'IsUserTable') = 1) DROP TABLE [PubSys].[Autori]
10 IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[Clanky]') and OBJECTPROPERTY(id,
    N'IsUserTable') = 1) DROP TABLE [PubSys].[Clanky]
11 IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[Kategorie]') and OBJECTPROPERTY(id,
    N'IsUserTable') = 1) DROP TABLE [PubSys].[Kategorie]
12 IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[Zpravodaj]') and OBJECTPROPERTY(id,
    N'IsUserTable') = 1) DROP TABLE [PubSys].[Zpravodaj]
13 IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
    OBJECT_ID(N'[PubSys].[Odkazy]') and OBJECTPROPERTY(id,
    N'IsUserTable') = 1) DROP TABLE [PubSys].[Odkazy]
14 GO
15
16 /*-----*/
17 /*--- Vytvoření tabulek ---*/
18 /*-----*/
19 CREATE TABLE [PubSys].[Autori] (
20     [AID] [int] IDENTITY (1, 1) NOT NULL ,
21     [Jmeno] [varchar] (20) NOT NULL ,
22     [Prijmeni] [varchar] (30) NOT NULL ,
23     [AutorskeJmeno] [varchar] (50) NOT NULL ,
24     [Email] [varchar] (50) NOT NULL ,
25     [ICQ] [varchar] (15) NOT NULL ,
```

```

26  [Mobil] [varchar] (12) NOT NULL ,
27  [Tel] [varchar] (12) NOT NULL ,
28  [Fax] [varchar] (12) NOT NULL ,
29  [Heslo] [char] (20) NOT NULL ,
30  [PrihlasovaciJmeno] [varchar] (20) NOT NULL
31 ) ON [PRIMARY]
32 GO
33
34 CREATE TABLE [PubSys].[Clanky] (
35  [CID] [int] IDENTITY (1, 1) NOT NULL ,
36  [KID] [int] NOT NULL ,
37  [AID] [int] NOT NULL ,
38  [Datum] [datetime] NOT NULL ,
39  [Titulek] [varchar] (100) NOT NULL ,
40  [Abstrakt] [varchar] (500) NOT NULL ,
41  [Telo] [text] NOT NULL ,
42  [PocetKomentaru] [int] NOT NULL ,
43  [Komentare] [text] NOT NULL ,
44  [PosledniKomentar] [datetime] NULL
45 ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
46 GO
47
48 CREATE TABLE [PubSys].[Kategorie] (
49  [KID] [int] IDENTITY (1, 1) NOT NULL ,
50  [Nazev] [varchar] (100) NOT NULL ,
51  [Popis] [varchar] (500) NOT NULL
52 ) ON [PRIMARY]
53 GO
54
55 CREATE TABLE [PubSys].[Zpravodaj] (
56  [Email] [varchar] (50) NOT NULL ,
57  [Datum] [datetime] NOT NULL ,
58  [IP] [varchar] (15) NOT NULL
59 ) ON [PRIMARY]
60 GO
61
62 CREATE TABLE [PubSys].[Odkazy] (
63  [Pocet] [int] NOT NULL ,
64  [URL] [varchar] (200) NOT NULL ,
65  [ServerJmeno] [varchar] (200) NOT NULL ,
66  [CID] [int] NOT NULL

```

```

67 ) ON [PRIMARY]
68 GO
69
70 /*-----*/
71 /*--- Vytvoření indexů ---*/
72 /*-----*/
73 ALTER TABLE [PubSys].[Autori] WITH NOCHECK ADD CONSTRAINT
[PK_Autori] PRIMARY KEY CLUSTERED ([AID]) WITH FILLFACTOR = 90 ON
[PRIMARY]
74 ALTER TABLE [PubSys].[Clanky] WITH NOCHECK ADD CONSTRAINT
[PK_Clanky] PRIMARY KEY CLUSTERED ([CID]) WITH FILLFACTOR = 90 ON
[PRIMARY]
75 ALTER TABLE [PubSys].[Kategorie] WITH NOCHECK ADD CONSTRAINT
[PK_Kategorie] PRIMARY KEY CLUSTERED ([KID]) WITH FILLFACTOR = 90
ON [PRIMARY]
76 ALTER TABLE [PubSys].[Zpravodaj] WITH NOCHECK ADD CONSTRAINT
[PK_Zpravodaj] PRIMARY KEY CLUSTERED ([Email]) ON [PRIMARY]
77 ALTER TABLE [PubSys].[Clanky] ADD CONSTRAINT [FK_Clanky_Kategorie]
FOREIGN KEY ([KID]) REFERENCES [PubSys].[Kategorie] ([KID])
78 ALTER TABLE [PubSys].[Clanky] ADD CONSTRAINT [FK_Clanky_Autori]
FOREIGN KEY ([AID]) REFERENCES [PubSys].[Autori] ([AID])
79 GO
80
81 /*-----*/
82 /*--- Vytvoření administrativního účtu ---*/
83 /*-----*/
84 INSERT INTO [PubSys].[Autori] (Jmeno, Prijmeni, AutorskeJmeno,
Email, ICQ, Mobil, Tel, Fax, Heslo, PrihlasovaciJmeno) VALUES ('-',
'-', '-', '-', '-', '-', '-', '-', 'psheslo', 'psadmin')
85 GO
86
87 /*-----*/
88 /*--- Vytvoření procedur ---*/
89 /*-----*/
90 SET QUOTED_IDENTIFIER ON
91 SET ANSI_NULLS ON
92 GO
93
94 CREATE PROCEDURE odkaz_pridej
95 @URL varchar(200),
96 @CID int
97 AS
98 IF EXISTS (SELECT * FROM PubSys.Odkazy WHERE URL=@URL AND CID=@CID)
BEGIN
99 UPDATE PubSys.Odkazy SET Pocet=Pocet+1 WHERE URL=@URL AND
CID=@CID

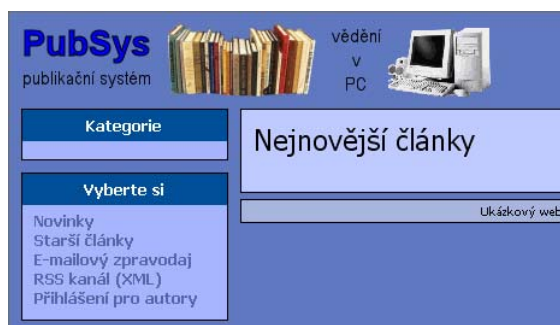
```

```

100 END ELSE BEGIN
101     DECLARE @ServerJmeno AS varchar(200)
102     IF CHARINDEX('/', @URL, 8) > 0 BEGIN
103         SET @ServerJmeno = SUBSTRING(@URL, 8, CHARINDEX('/',
@URL, 8) - 8)
104     END ELSE BEGIN
105         SET @ServerJmeno= SUBSTRING(@URL, 8, 999)
106     END
107     INSERT INTO PubSys.Odkazy (Pocet, URL, ServerJmeno, CID)
VALUES (1, @URL, @ServerJmeno, @CID)
108 END
109 GO
110 CREATE PROCEDURE zpravodaj_pridej
111     @Email varchar(50),
112     @IP    varchar(15)
113 AS
114     SET @Email = LOWER(@Email)
115     IF (SELECT COUNT(*) FROM PubSys.Zpravodaj WHERE Email=@Email) <> 0
BEGIN
116         SELECT 'CHYBA: Adresa se již nachází v seznamu zpravodaje'
117     END ELSE BEGIN
118         INSERT INTO PubSys.Zpravodaj (Email, Datum, IP) VALUES
(@Email, GETDATE(), @IP)
119         SELECT 'Adresa byla úspěšně přidána'
120     END
121 GO
122 CREATE PROCEDURE zpravodaj_odeber
123     @Email    varchar(50)
124 AS
125     SET @Email = LOWER(@Email)
126     IF (SELECT COUNT(*) FROM PubSys.Zpravodaj WHERE Email=@Email) = 0
BEGIN
127         SELECT 'CHYBA: Adresa se nenachází v seznamu zpravodaje'
128     END ELSE BEGIN
129         DELETE FROM PubSys.Zpravodaj WHERE Email=@Email
130         SELECT 'Adresa byla úspěšně odstraněna'
131     END
132 GO
133 SET QUOTED_IDENTIFIER OFF
134 SET ANSI_NULLS ON
135 GO

```

Příloha P II: Výřez okna prohlížeče – prázdný systém po prvním spuštění



Příloha P III : Výřez okna prohlížeče – přihlášení do redakce

The screenshot shows the PubSys web application interface with the login form for the editorial staff. At the top, there is a header with the PubSys logo, a bookshelf icon, and the text "vědění v PC" with a computer icon. Below the header, there is a sidebar on the left with a "Kategorie" section and a "Vyberte si" section. The "Kategorie" section contains a list of categories: "Cestování", "Ekonomika", "Počítače a technika", "Zdraví", and "Zprávy". The "Vyberte si" section contains a list of options: "Novinky", "Starší články", "E-mailový zpravodaj", "RSS (XML)", and "Přihlášení pro autory". The main content area on the right is titled "Přihlášení do redakce" and contains a form with the following fields: "Login:", "Heslo:", and a checkbox "pamatovat si mne na tomto počítači". There is a "Odeslat" button. A red box with the text "Tak právě zde ..." is overlaid on the top right of the form. At the bottom of the form, there is a footer with the text "Ukázkový web publikačního systému PubSys | © Marek Konvičný, [GEW](#), 2005".

Příloha P IV : Výřez okna prohlížeče – stránka autorů

The screenshot shows the PubSys web application interface with the author registration and list page. At the top, there is a header with the PubSys logo, a bookshelf icon, and the text "vědění v PC" with a computer icon. Below the header, there is a sidebar on the left with a "Kategorie" section and a "Vyberte si" section. The "Kategorie" section contains a list of categories: "Cestování", "Ekonomika", "Počítače a technika", "Zdraví", and "Zprávy". The "Vyberte si" section contains a list of options: "Novinky", "Starší články", "E-mailový zpravodaj", "RSS (XML)", and "Přihlášení pro autory". The main content area on the right is titled "Autoři" and contains a form for "Nový autor". The form has two sections: "Data a kontakty" and "Autentizace". The "Data a kontakty" section contains fields for "Jméno:", "Příjmení:", "Autorské jméno:", "Email:", "ICQ:", "Mobil:", "Tel:", and "Fax:". The "Autentizace" section contains fields for "Přihlašovací jméno:", "Heslo:", and "Ověření hesla:", and a "Vytvořit" button. Below the form, there is a section titled "Seznam autorů" which contains a table of authors.

	Jméno	Příjmení	Autorské jméno	Email	ICQ	Mobil	Tel	Fax
	Jan	Novák	JN	jan.novak@pubsys.cz	951736842	248631795	528693417	528693417
	Lucie	Bendů	LB	lucie.bendu@pubsys.cz	486245915	379128645		
	Axel	McConnor	AxMC	axel.mcconnor@pubsys.cz	654321851	813285215	182382453	215252135
	Karel	Petr	KP	karel.petr@pubsys.cz	813535426	154325455	794132457	281813554

Příloha P V : Snímek okna prohlížeče – nejnovější články

PubSys publikační systém vědění v PC

... zde mohla být Vaše reklama!

Nejnovější články

Sicílie: kde najdete tu neitalštější Itálii! [Cestování]
[IDNES.cz] Myslíte si, že v Corleone potkáte typického mafiána s placatou čepicí, rozšafným knírem a kulovnicí? Skoro jistě ne. To však neznamená, že zde mafiáni vymřeli. Být se však nemusíte. Sicílie je pro nás zatím málo známým, ale skvělým místem pro dovolenou!
[AxMC](#) (26.5.2005 18:41:03) [Počet komentářů: 7](#)

Na Mount Everest poprvé vylétl vrtulník [Zprávy]
[IDNES.cz] Případ nájemného vraha Jiřího Kajínka inspiruje filmaře. Na motivy jeho kauzy vznikne i celovečerní film. Režisérem a zároveň scénáristou filmu, který nese pracovní název Jiří Kajíněk – Nájemný vrah nebo český Monte Christo, je Jiří Svoboda. V kinech se objeví v roce 2007.
[LB](#) (25.5.2005 19:05:25)

Rodiče chtějí miliony, lékaři nepoznali otravu [Zprávy]
[IDNES.cz] Manželé Sklenářovi z Tachova žádají skoro 4 a půl milionu za smrt 15letého syna. Loni spadl ze stromu a poranil si stehno. Lékaři nepoznali otravu krve a ošetřili jen zhmožděninu. Když se školákův stav zhoršoval, rodiče s ním jeli do Plzně. Zemřel čtyři dny po úrazu.
[JN](#) (25.5.2005 18:42:48) [Počet komentářů: 2](#)

Sebevraha zachránil skok do plachty [Zprávy]
[IDNES.cz] Skokem z komínu v areálu Vítkovických železáren v Ostravě se rozhodl vyřešit své problémy 21letý muž. Vyšplhal na komín už kolem třinácté hodiny. Přijel ho přemluvit i policejní vyjednávač. Marně. Mladík nakonec skočil do záchranné plachty. UVNITŘ FOTOGRAFIE
[AxMC](#) (25.5.2005 18:39:07)

20 zemí spouští kampaň proti zombiím [Počítače a technika]
[ZIVE.cz] a tedy proti SPAMu
[KP](#) (25.5.2005 17:31:14)

Příloha P VI : Snímek okna prohlížeče – výpis jedné kategorie

PubSys publikační systém vědění v PC

Tak právě zde ...

Cestování

Sicílie: kde najdete tu neitalštější Itálii! [Cestování]
[IDNES.cz] Myslíte si, že v Corleone potkáte typického mafiána s placatou čepicí, rozšafným knírem a kulovnicí? Skoro jistě ne. To však neznamená, že zde mafiáni vymřeli. Být se však nemusíte. Sicílie je pro nás zatím málo známým, ale skvělým místem pro dovolenou!
[AxMC](#) (26.5.2005 18:41:03) [Počet komentářů: 7](#)

Napálí vás cestovka. Máte šanci se bránit? [Cestování]
[IDNES.cz] Mnozí z nás vyrazí brzy na vysněnou dovolenou. Někteří na vlastní pěst, jiní s cestovkou. Pokud pojedete s cestovní kanceláří, měli byste si dobře pročíst smlouvu a také vědět, co dělat v případě, když CK nesplní podmínky. I vám se totiž může stát, že místo v tříhvězdičkovém hotelu skončíte v ubytovně s plísní v koupelně.
[LB](#) (25.5.2005 17:20:29)

Pět tisíc let stará superstar [Cestování]
[IDNES] Archeologické muzeum - slova, která jsou pro mnohé synonymem nudy. Nedejte se mýlit. Z Jihočtyrského archeologického muzea v italském Bolzanu odejdete pobaveni a s novým přítelem. Budete znát jeho barvu vlasů i oči, výšku i velikost bot, co ho v životě potkalo, a dokonce i jak zemřel. Jmenuje se Ötzi.
[LB](#) (25.5.2005 14:07:25)

Léto na kole v Dolním Rakousku [Cestování]
[IDNES] Z mnoha míst jižní Moravy jste tam za několik desítek minut. Z Prahy asi za tři hodinky. Je to proto dobré místo na kratší výlet či pohodlnou letní dovolenou. Dolní Rakousy. Kraj romantických vinic, poklidného života a stovek kilometrů dnes tak populárních cyklostezek. Tipy na cyklostezky - v článku.
[KP](#) (25.5.2005 13:57:42)

Rumunsko II. - pohoří Bihor [Cestování]
Další díl seriálu o Rumunských horách - pohoří Bihor. Bihor je údajně nejkrásnější oblast Evropy. Nalezneme zde několik ledových jeskyní, fenomén ponorné řeky, které se z ničeho nic strátí např. uprostřed louky a o pár kilometrů dál zase vyvěrají.

Příloha P VII : Snímek okna prohlížeče – nový článek

The screenshot shows the 'PubSys' web application interface. The header includes the logo 'PubSys publikační systém', a bookshelf icon, and a computer icon with the text 'věděni v PC'. A red banner at the top right says '... zde mohla být Vaše reklama!'. The left sidebar contains three sections: 'Redakční menu' with links like 'Nový článek', 'Editovat článek', 'Správce souborů', 'Kategorie', 'Zpravodaj', 'Statistika zájmu', 'Autoři', and 'Odhlášení'; 'Kategorie' with 'Cestování', 'Ekonomika', 'Počítače a technika', 'Zdraví', and 'Zprávy'; and 'Vyberte si' with 'Novinky', 'Starší články', 'E-mailový zpravodaj', and 'RSS (XML)'. The main content area is titled 'Nový článek' and contains form fields for 'Titulek:', 'Abstrakt:', and 'Kategorie:' (set to 'Cestování'). Below these is a rich text editor with a toolbar including Paragraph, Font, Size, Color, Highlight, Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Link, Unlink, Image, Table, and other standard editing tools. The footer shows 'Hotovo' and 'Místní intranet'.

Příloha P VIII : Snímek okna prohlížeče – kategorie

The screenshot shows the 'PubSys' web application interface for the 'Kategorie' (Categories) page. The header and left sidebar are identical to the previous screenshot. The main content area is titled 'Kategorie' and contains a section for 'Nová kategorie' (New Category) with form fields for 'Název:' and 'Popis:', and a 'Vytvořit' (Create) button. Below this is a 'Seznam kategorií' (List of Categories) table. The footer includes the text 'Ukázkový web publikačního systému PubSys | © Marek Konvičný, GEW, 2005' and 'Místní intranet'.

Název	Popis
Cestování	Cestopisy, turistické informace o různých zemích, krajích a oblastech, recenze ubytovacích, stravovacích a rekreačních zařízení, nejen u nás, ale i v zahraničí. Atd.
Ekonomika	Vše kolem ekonomiky, účetnictví, bankovníctví, investic, apod.
Počítače a technika	Informace (nejen) ze světa IT.
Zdraví	Novinky, rady, postřehy a potřeby pro zdraví.
Zprávy	Co se aktuálně děje kolem nás.

Příloha P IX : Výřez okna prohlížeče – příklad diskuse – prvek

- [Krása](#) [Jana, 28.05.2005 12:22]
 - [RE: Krása](#) [Pepa, 29.05.2005 00:56]
- [sicílie](#) [barca, 28.05.2005 17:31]
 - [RE: sicílie](#) [jana, 28.05.2005 17:32]
 - [RE: sicílie - tak tady to máš](#) [Fanda, 29.05.2005 01:07]
 - [RE: sicílie](#) [Fanda, 29.05.2005 01:13]
- [pěkný článek](#) [Marek, 29.05.2005 01:24]

[Zobrazit komentáře](#) | [Přidat komentář](#) | [RSS komentářů](#)

Příloha P X : Výřez okna prohlížeče – příklad diskuse – výpis

Komentáře k článku [Sicílie: kde najdete tu neitalštější Itálii!](#)

Krása
[Jana](#), 28.05.2005 12:22 [Odpovědět](#)

Tam bych chtěla jet. Neznáte někdo která cestovka má Sicílii nejlevněji?

RE: Krása
[Pepa](#), 29.05.2005 00:56 [Odpovědět](#)

Bud' v klidu existují i krásnější místa. Co se týká levné cestovky, budeš se muset hold probrat jejich nabídkama.

sicílie
[barca](#), 28.05.2005 17:31 [Odpovědět](#)

Nevím, ale cedok to nebude

RE: sicílie
[jana](#), 28.05.2005 17:32 [Odpovědět](#)

Očekávala jsem lepší radu.

RE: sicílie - tak tady to máš
[Fanda](#), 29.05.2005 01:07 [Odpovědět](#)

Nejkrásnější vypadá CK Valaška, ale raději si to ověř...

Příloha P XI : Výřez okna prohlížeče – nový komentář k článku

Nový komentář k článku [Sicílie: kde najdete tu neitalštější Itálii!](#)

Žádné z polí formuláře nesmí obsahovat HTML formátování. Konce řádků budou převedeny na odstavce, internetové adresy a e-maily na odkazy.

Jméno:

E-mail:

☒ zveřejnit e-mail u komentáře

☒ posílat mailem odpovědi na můj komentář

Home page:

Předmět:

Text:

Odeslat

Příloha P XII : Výřez okna prohlížeče – detail hlavičky článku

The screenshot shows a web browser window with a light blue background. At the top, the article title "Sicílie: kde najdete tu neitalstějši Itálii!" is displayed in bold black text. To the right of the title, the text "Cestování - 26.5.2005 18:41:03" and "AxMC" are visible. Below the title, a smaller line of text reads "9.5.2005 18:32:04 - Na cestě po Itálii". A central image shows ancient ruins with a snow-capped mountain in the background. Two callout boxes are present: one on the left pointing to the title with the text "Titulek článku", and one on the right pointing to the top right area with the text "Proklikávací položky: kategorie, datum a autor".

Sicílie: kde najdete tu neitalstějši Itálii!

Cestování - 26.5.2005 18:41:03
AxMC

9.5.2005 18:32:04 - Na cestě po Itálii

Sicílie: kde najdete tu neitalstějši Itálii!

Titulek článku

Proklikávací položky: kategorie, datum a autor

Příloha P XIII : Výřez okna prohlížeče – správce souborů

The screenshot shows a file manager window titled "Správce souborů". At the top, there is a search bar and two buttons: "Procházet..." and "Upload". Below this is a table with three columns: "název", "velikost", and "poslední změna". Each row in the table starts with a red 'X' icon followed by a filename. The files listed are l1_12.jpg, l2_06.jpg, r2_05a.jpg, r2_15a.jpg, r3_18a.jpg, r3_23a.jpg, and r4_28a.jpg. The sizes and last modification times are listed in the subsequent columns.

Správce souborů

Procházet... Upload

název	velikost	poslední změna
✗ l1_12.jpg	45 662 B	29.05.2005 01:55:49
✗ l2_06.jpg	51 498 B	29.05.2005 01:56:02
✗ r2_05a.jpg	43 876 B	29.05.2005 01:56:16
✗ r2_15a.jpg	44 125 B	29.05.2005 01:55:54
✗ r3_18a.jpg	34 758 B	29.05.2005 01:56:34
✗ r3_23a.jpg	45 840 B	29.05.2005 01:56:41
✗ r4_28a.jpg	42 152 B	29.05.2005 01:56:45