

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

INFORMAČNÍ SYSTÉM MALÉ OBCE POMOCÍ ASP.NET

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

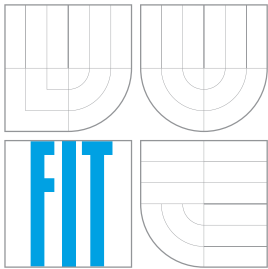
AUTOR PRÁCE
AUTHOR

VÁCLAV PACHTA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

INFORMAČNÍ SYSTÉM MALÉ OBCE POMOCÍ ASP.NET

VILLAGE WEB INFORMATION SYSTEM BASED ON ASP.NET

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VÁCLAV PACHTA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVEL JURKA

BRNO 2007

Abstrakt

Obsahem této práce je návrh a realizace informačního systému malé obce. Cílem je navrhnout systém, který zrychlí, zjednoduší a zefektivní informování občanů ze strany obecního úřadu. Tento systém umožní oprávněným uživatelům spravovat obsah stránek, obsah úřední desky, fotogalerii. To vše s využitím sítě internet.

Klíčová slova

Obec, úřední deska, elektronická podatelna, informační systém, databáze, aplikace, ASP.NET, Microsoft SQL, XHTML, CSS.

Abstract

This work deals with a concept and realization of a village web information system. The object is to create a system that improves the information distribution effectiveness, speed and availability from the side of local municipal office. This system allows to the competent user to manage content of web-pages, municipal bulletin board and photogallery.¹ All of this services is accessible through the internet.

Keywords

Village, bulletin board, electronic registry, information system, database, application, ASP.NET, Microsoft SQL, XHTML, CSS

Citace

Václav Pachta: Informační systém malé obce pomocí ASP.NET, bakalářská práce, Brno, FIT VUT v Brně, 2007

Informační systém malé obce pomocí ASP.NET

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Pavla Jurky.

.....

Václav Pachta
11. května 2007

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Pavlu Jurkovi, za odborné vedení, ochotu při řešení problému, cenné rady a připomínky při zpracování a řešení bakalářské práce.

© Václav Pachta, 2007.

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

Obsah

1	Úvod	3
1.1	Průvodce prací	3
2	Cíl práce	4
2.1	Povinnosti obce	4
2.2	Požadavky na funkčnost systému	5
2.3	Technické požadavky	5
2.4	Uživatelské požadavky	5
3	Návrh řešení	6
3.1	Dekompozice IS	6
3.2	Prostředky pro návrh	6
3.2.1	UML	6
3.2.2	Toad Data Modeler	7
3.3	Režimy a uživatelé aplikace	7
3.3.1	Anonymní uživatel	7
3.3.2	Správce obsahu stránek	7
3.3.3	Správce článků	7
3.3.4	Správce úřední desky	7
3.3.5	Správce odkazů	8
3.3.6	Správce fotogalerie	8
3.3.7	Správce kalendáře akcí	8
3.3.8	Moderátor knihy návštěv	8
3.3.9	Administrátor	9
3.4	Moduly aplikace	9
3.4.1	Základní obsah stránek	10
3.4.2	Články aktualit v obci	10
3.4.3	Úřední deska	10
3.4.4	Odkazy	10
3.4.5	Fotogalerie	10
3.4.6	Kalendář akcí	10
3.4.7	Knihy návštěv	10
3.4.8	Elektronická podatelna	10
3.4.9	Správa uživatelů a rolí	10
3.5	Zabezpečení	11
3.6	Návrh databáze	11

4 Implementace	13
4.1 Výběr a popis technologických prostředků	13
4.1.1 PHP	13
4.1.2 ASP.NET	14
4.1.3 XHTML	17
4.1.4 CSS	17
4.1.5 Microsoft SQL server 2005	18
4.1.6 MySQL	19
4.1.7 WYSIWYG editor	19
4.1.8 Závěr	19
4.2 Implementace databáze	20
4.2.1 Tabulka Články	20
4.2.2 Tabulka Typ článku	20
4.2.3 Tabulka Odkazy	20
4.2.4 Tabulky Album a Fotky	20
4.2.5 Tabulka Akce	20
4.2.6 Tabulka Typ akce	20
4.2.7 Tabulka vzkazy	21
4.2.8 Tabulka Users	21
4.2.9 Tabulka Roles	21
4.2.10 Tabulka Users in Roles	21
4.3 Implementace jednotlivých modulů a stránek	21
4.3.1 Master page	21
4.3.2 Obsah stránek	21
4.3.3 Úřední deska	22
4.3.4 Články	23
4.3.5 Elektronická podatelna	23
4.3.6 Odkazy	24
4.3.7 Fotogalerie	24
4.3.8 Kalendář akcí	24
4.3.9 Kniha návštěv	24
4.3.10 Správa uživatelů a rolí	25
4.4 Nastavení přístupových práv ke stránkám	25
5 Závěr	28

Kapitola 1

Úvod

Informační systémy obcí v České republice doznaly za posledních několik let řady změn. Bylo to způsobeno zejména vydáním několika zákonů a vyhlášek, které upřesňují pravidla pro tyto informační systémy.

Obsahem této práce je analýza a návrh informačního systému s důrazem na systémy pro malé obce. Domnívám se totiž, že mnou navržený systém bude pro města a velké obce nedostatečný. Jak z hlediska množství a kvality funkcí, tak z hlediska všech současných a neustále se vyvíjejících možností.

Důvodem, proč jsem se rozhodl pro téma internetové aplikace informačního systému malé obce, je můj zájem o danou problematiku a přesvědčení, že mnou navržený systém má reálnou šanci na uplatnění v praxi především menších obcí v České republice.

1.1 Průvodce prací

Při psaní kapitoly 2 „Cíl práce” jsem vyšel z [7], [2] a z rozhovoru se starostou obce Batelov panem Bc. Petrem Janouškem.

Kapitola 3 „Návrh řešení” je můj vlastní tvůrčí návrh. Tvořený za pomoci [8] a programu Toad Data Modeler [3].

V kapitole 4 „Implementace” u podkapitoly 4.1 „Výběr a popis technologických prostředků” jsem čerpal z [6] [9], [4] a z [1]. Zbylé tři podkapitoly 4.2 „Implementace databáze”, 4.3 „Implementace jednotlivých modulů a stránek” a 4.4 „Nastavení přístupových práv ke stránkám” jsou mou vlastní tvorbou.

Kapitola 2

Cíl práce

Cílem této práce je vytvoření internetové aplikace informačního systému malé obce, ve které se přehledně vyzná každý uživatel a která pokryje všechny základní požadavky dle daných zákonů týkajících se obce. Jak ze strany návštěvníků webu, tak z pohledu jejích správců.

2.1 Povinnosti obce

Obec je podle zákona o svobodném přístupu k informacím povinna zveřejnit následující informace:

1. Oficiální název
2. Důvod a způsob založení obce
3. Organizační strukturu obce
4. Kontaktní spojení
5. Bankovní spojení
6. IČO a DIČ
7. Rozpočet obce
8. Místo a způsob, jak získat příslušné informace
9. Čas a způsob příjmu žádostí a dalších podání
10. Opravné prostředky
11. Formuláře
12. Nejdůležitější používané předpisy
13. Sazebník úhrad za poskytování informací
14. Výroční zpráva podle zákona č.106/1999 Sb.
15. Seznam organizací

Dále je obec povinna podle § 26 zákona č. 500/2004 Sb. zřizovat úřední desku (viz [2]), která musí být nepřetržitě veřejně přístupná. Její obsah se musí zveřejňovat též způsobem umožňujícím dálkový přístup.

Nařízení 495/2004 a vyhlášky 496/2004 nařizuje i malým obcím zřídit a používat elektronickou podatelnu (viz [7]) a zaručený elektronický podpis.

2.2 Požadavky na funkčnost systému

Z předchozí podkapitoly 2.1 vyplývají základní požadavky na funkčnost. Jsou to tedy:

1. Správa povinných údajů a obsahu stránek
2. Úřední deska a její správa (přidávání, editace a mazání)
3. Elektronická podatelna (formulář pro odesílání e-mailů na adresu E-podatelný)

Kromě toho by zde samozřejmě měli být nějaké informace o samotné obci (obsah stránek).

Další funkčnost webové aplikace záleží na dané obci. Pro tuto práci implementují ještě následující funkce:

1. Správa článků a obsahu stránek pomocí WYSIWYG editoru
2. Správa odkazů
3. Moderovaná kniha návštěv
4. Kalendář akcí a jeho správa
5. Fotogalerii a její správu
6. Správa uživatelů a jejich rolí

2.3 Technické požadavky

- Spolehlivost a stabilita - kvalitní a spolehlivé technologie
- Bezpečnost - autorizovaný a omezený přístup, ochrana osobních údajů
- Přístupnost - minimalizace nároků na webovou aplikaci

2.4 Uživatelské požadavky

- Uživatelské rozhraní - jednoduché a intuitivní ovládání
- Rychlost - vysoký výkon, okamžitá odezva

Kapitola 3

Návrh řešení

3.1 Dekompozice IS

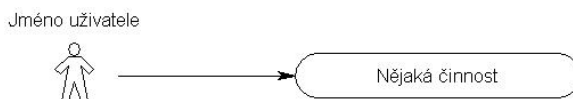
Informační systém obce se člení na dvě hlavní části. Část pro anonymní návštěvníky stránek a část pro správu systému. Úkolem anonymní části je zobrazovat obsah stránek, případně napsat příspěvek do knihy návštěv, nebo odeslat požadavek do elektronické podatelny. Druhá část umožňuje spravovat obsah stránek, úřední desku atd. K této části mají přístup pouze oprávnění uživatelé.

3.2 Prostředky pro návrh

3.2.1 UML

Unified Modeling Language (viz [8]) je v softwarovém inženýrství grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů. UML nabízí standardní způsob zápisu jak návrhů systému včetně, tak konkrétních prvků jako jsou příkazy programovacího jazyka, databázová schémata a znovupoužitelné programové komponenty. Jeho prvotní autoři jsou Grady Booch, Jim Rumbaugh a Ivar Jacobson. Standard UML definuje standardizační skupina Object Management Group (OMG). Poslední standardizovaná verze je 2.1.1. UML se nejčastěji používá pro tvorbu diagramů např. pro diagramy tříd, objektů, případů použití (use case diagram) atd.

V této práci používám z UML pouze diagramy případů použití. Případ užití je popis chování systému z pohledu uživatele. Pro vývojáře systému je to osvědčený způsob shromáždění údajů o požadavcích systému z hlediska uživatele. Příklad use case diagramu je na obrázku 3.1.



Obrázek 3.1: Příklad diagramu případu použití

Panáček představující nějakého uživatele se nazývá participant (účastník). Elipsa znázorňuje případ užití. Přičemž účastníkem může být jak osoba, tak i jiný systém.

3.2.2 Toad Data Modeler

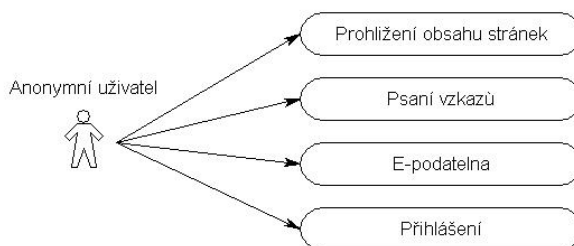
Toad Data Modeler je program sloužící k návrhu databází (viz [3]). Umožňuje navrhovat ER diagramy databází. Pro navrhnutý diagram umí vygenerovat skript pro vytvoření příslušné struktury databáze a to pro víc než 20 různých databázových systémů. Kromě toho umí i tzv. „Reverse Engineering“ (načtení struktury již existující databáze).

3.3 Režimy a uživatelé aplikace

Navrhovaná aplikace bude pracovat v několika různých režimech. Každý režim bude učen pro jinou uživatelskou skupinu. Každá skupina uživatelů bude mít různá práva a různé možnosti přístupu k jednotlivým modulům aplikace. Přičemž jeden uživatel může být členem více uživatelských skupin. Pro každou uživatelskou skupinu zde uvedu její popis a její diagram případů užití.

3.3.1 Anonymní uživatel

Anonymní uživatel je výchozím typem uživatele aplikace. Může pouze procházet obsah stránek, psát příspěvky v knize návštěv, posílat požadavky do elektronické podatelny nebo se přihlásit do systému.



Obrázek 3.2: anonymní uživatel

3.3.2 Správce obsahu stránek

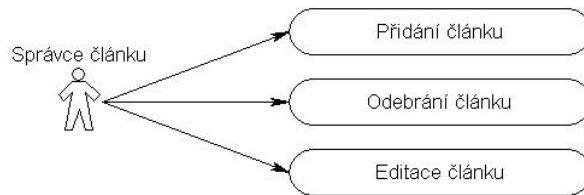
Tento uživatel může pomocí WYSIWYG editoru měnit základní obsah stránek. Jako jsou například informace o obci, povinné údaje atd.

3.3.3 Správce článků

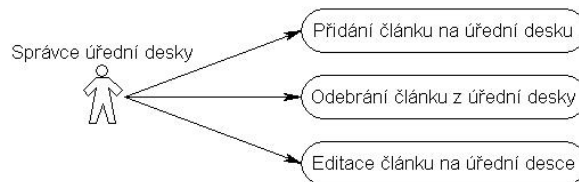
Pomocí WYSIWYG editoru vkládá a upravuje články z dění v obci a okolí. kromě úpravy a vkládání článků má možnost články také mazat.

3.3.4 Správce úřední desky

Stará se o zveřejňování informací na úřední desce. Může přidávat, upravovat a odebírat články z úřední desky.



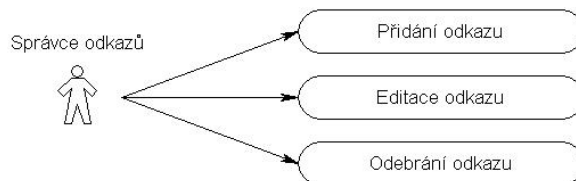
Obrázek 3.3: Správce článků



Obrázek 3.4: Správce úřední desky

3.3.5 Správce odkazů

Uživatelé z této skupiny se starají o odkazy na ostatní webové stránky. Jsou schopní přidávat, odebírat a měnit odkazy (viz obrázek 3.5).



Obrázek 3.5: Správce odkazů

3.3.6 Správce fotogalerie

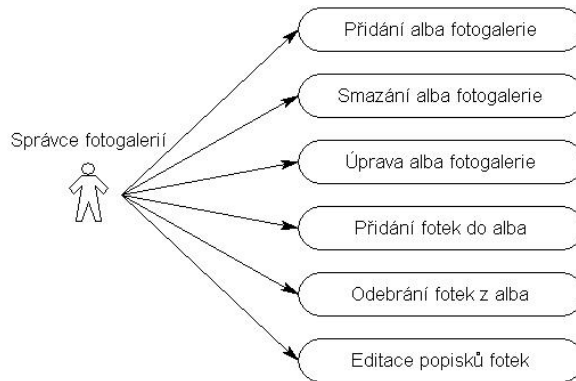
Stará se o fotogalerii. Je schopen přidávat, mazat a editovat alba s fotogaleriemi. Přidávat fotky do jednotlivých alb, mazat je a nastavovat jim popisky.

3.3.7 Správce kalendáře akcí

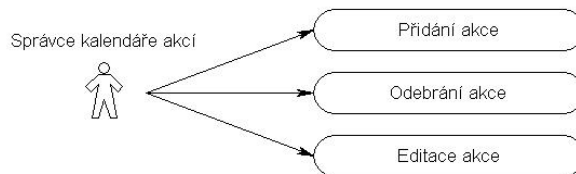
Má za úkol starat se o kalendář akcí v obci. Zveřejňuje informace o různých akcích, ať už kulturních nebo sportovních, které se budou dít v obci.

3.3.8 Moderátor knihy návštěv

Jediným úkolem moderátora knihy návštěv je mazání nevhodných a pobuřujících vzkazů, které jsou na webových stránkách obce nežádoucí.



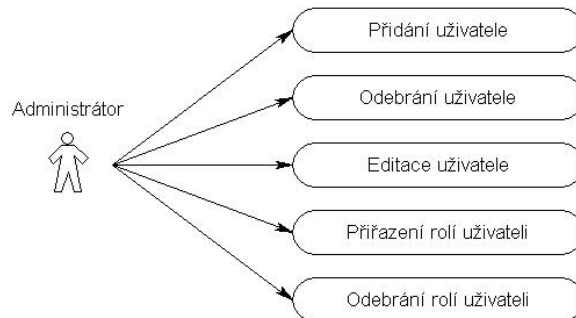
Obrázek 3.6: Správce fotogalerie



Obrázek 3.7: Správce kalendáře akcí

3.3.9 Administrátor

Administrátor má veškerá oprávnění, které jsem uvedl u předchozích uživatelských skupin. Navíc má na za úkol správu uživatelů a jejich rolí.



Obrázek 3.8: Administrátor

3.4 Moduly aplikace

Aplikace se skládá z několika modulů. Každý modul se skládá ze dvou částí. Část pro zobrazení modulu a část pro správu modulu (kromě elektronické podatelny). Ke správě daného modulu budou mít přístup pouze privilegovaní uživatelé.

3.4.1 Základní obsah stránek

Tento modul slouží pro zobrazení běžných informací pro všechny uživatele. kromě povinných informací se budou pomoci tohoto modulu. Týká se např. informace o historii obce, demografie atd. Uživatel patřící do skupiny „Správce obsahu stránek” (viz 3.3.2) bude mít možnost měnit texty v těchto modulech.

3.4.2 Články aktualit v obci

Pomocí tohoto modulu se budou zobrazovat články z dění v obci a jejím okolí např. reportáže ze sportovních nebo kulturních událostí. V části pro správu tohoto modulu budou mít uživatelé ze skupiny „Správce článků” (viz 3.3.3 po přihlášení možnost vkládat, editovat a samozřejmě i mazat články.

3.4.3 Úřední deska

Modul pro úřední desku je používán k zobrazením jejích příspěvků. Např. obecně závazné vyhlášky a nařízení obce, uzavřené veřejnoprávní smlouvy atd. Funkčnost tohoto modulu je velice podobná modulu pro články. Správu tohoto modulu zajišťují uživatelé ze skupiny „Správce úřední desky” (viz 3.3.4)

3.4.4 Odkazy

Modul slouží k zobrazení zajímavých odkazů pro obec. Např. odkaz na stránky se zprávami, na webové stránky spřátelených obcí atd. Správu tohoto modulu zajišťuje uživatelská skupina „Správce odkazů” (viz 3.3.5)

3.4.5 Fotogalerie

Modul sloužící pro zobrazení a správu galerie fotek z obce a okolí.

3.4.6 Kalendář akcí

Modul sloužící pro zadávání vyhledávání a uchovávání údajů o kulturních či sportovních akcích, které se budou v obci dít. O vkládání, editaci a mazání akcí se starají uživatelé ze skupiny „Správce kalendáře akcí” (viz 3.3.7)

3.4.7 Kniha návštěv

Slouží pro návštěvníky webové aplikace ke vkládání vzkazů a připomínek. Kniha návštěv je moderována uživateli „Moderátor knihy návštěv” (viz 3.3.8), kteří odstraňují nežádoucí vzkazy.

3.4.8 Elektronická podatelna

Modul sloužící k odesílání požadavků na e-mailovou adresu elektronické podatelny.

3.4.9 Správa uživatelů a rolí

Modul určený pouze pro skupinu „Administrátor” (viz 3.3.9). Umožňuje administrátorovi přidávat uživatele, přiřazovat jim skupiny atd.

3.5 Zabezpečení

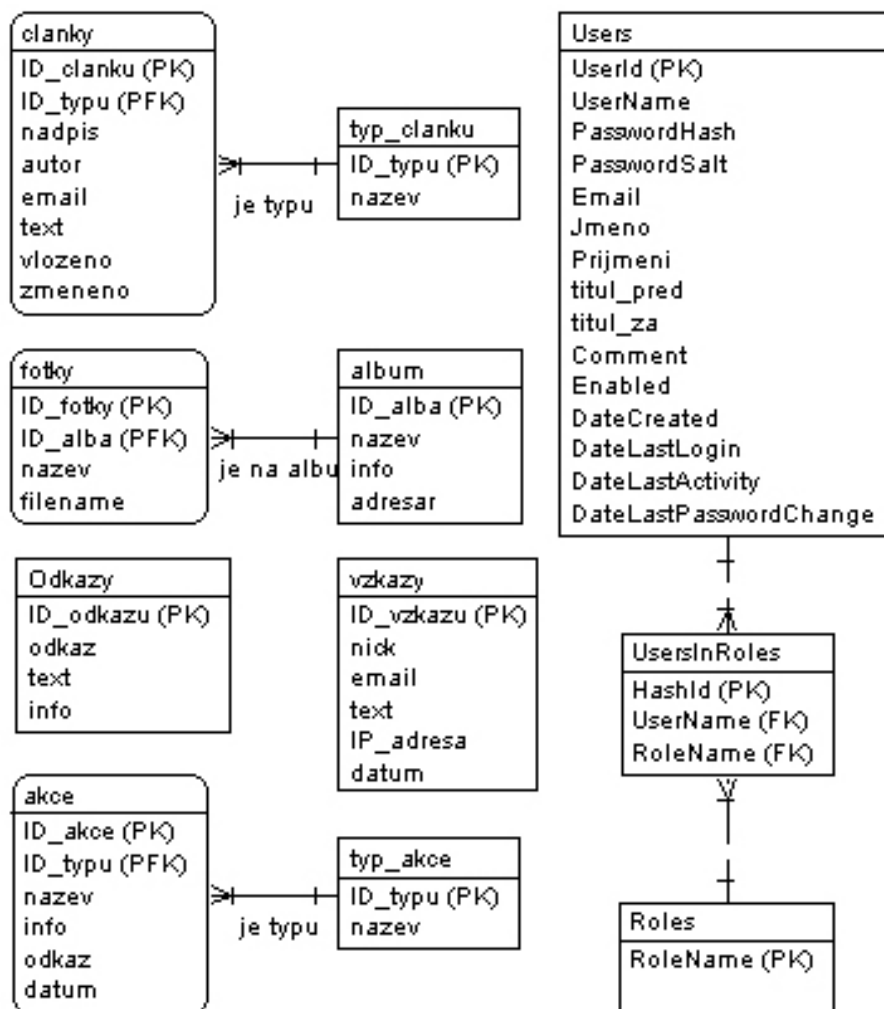
Jedním z podmínek bezproblémového chodu systému je zajištění přístupu uživatel a specifikace jejich práv.

Každý registrovaný uživatel se do systému bude přihlašovat za pomoci jedinečného uživatelského jména a hesla. Toto jméno a heslo zná pouze daný uživatel. Každý uživatel má jasně daný přístup k jednotlivým modulům aplikace.

Uživatele může registrovat pouze administrátor. Přičemž zadává pouze uživatelské jméno, jméno, příjmení, titul a e-mailovou adresu uživatele. Heslo je vygenerováno náhodně a je odesláno uživateli na jeho e-mailovou adresu.

3.6 Návrh databáze

Po analýze funkčních požadavků kladených na aplikaci byla navržena databáze s následujícím schématem viz obrázek 3.9.



Obrázek 3.9: ER diagram

Tabulka “Clanky” slouží nejen pro uložení článků a příspěvků na úřední desce, ale i pro uložení vlastního obsahu stránek. Pro rozpoznání ke které kategorii článek patří slouží vazební tabulka “typ_clanku”. Tabulky “akce” a “typ_akce” byla navržena pro uložení dat, které jsou použity u kalendáře akcí. Tabulka “vzkazy” slouží k zachování vzkazů pro knihu návštěv. Tabulka „odkazy” slouží k uložení informací o odkazech. Tabulky „album” a „fotky” slouží pro fotogalerii. Zbylé tři tabulky (“Users”, “UsersInRoles” a “Roles”) jsou určeny pro správu uživatelů a jejich rolí.

Kapitola 4

Implementace

Po etapě návrhu následuje fáze implementace.

4.1 Výběr a popis technologických prostředků

Dnešním trendem jsou dynamické webové stránky tj. při požadavku klienta server vykoná programový kód a klientovi posílá pouze data nejčastěji ve formátu (X)HTML nebo XML. Klient tedy nekomunikuje přímo s programem webové aplikace, ale posílá pouze žádost o daný dokument, přičemž server vykonává vlastní program a klientovi pošle výsledný dokument. Nejčastěji používané programovací jazyky pro vývoj dynamických webových aplikací jsou ASP, ASP.NET, Perl a PHP. Data se ukládají za pomoci databázových systémů, nejčastěji MySQL, MS SQL a Oracle.

4.1.1 PHP

PHP (Hypertext Preprocessor, nebo původně Personal Home Page) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Většinou se začleňuje přímo do struktury jazyka HTML, XHTML či WML. Syntakticky je PHP nejvíce podobné jazyku C. PHP se stalo velmi oblíbeným především díky své jednoduchosti.

Výhody PHP:

- Je jednoduché na naučení.
- Je multiplatformní (funguje jak na systémech Windows, tak i na unixových systémech).
- Jednoduchá administrace pomocí PHP.INI
- Má malé nároky na hardware
- Veliká knihovna funkcí, knihoven a aplikací zdarma.
- PHP je open source.

Nevýhody:

- Nedostatečný objektový model. Knihovna základních funkcí PHP je procedurální, téměř nevyužívá objekty. Základní funkce nevyhazují výjimky, což v podstatě znemožňuje opravdové objektové programování.

- Nesystematičnost v přidávání a pojmenovávání nových funkcí. PHP obsahuje funkce skoro na všechno, na co si vzpomenete, ale tyto funkce jsou přidávány bez ladu a skladu a pojmenovávány jednou s podtržítkem, podruhé bez, jednou celým jménem, jednou zkratkou...
- PHP je interpretovaný jazyk.
- Není třeba deklarovat a inicializovat proměnné.

4.1.2 ASP.NET

Objektově orientovaná technologie pro tvorbu webových aplikací od společnosti Microsoft založená na .NET frameworku (viz [6]). Název je odvozen od staršího ASP, ale nemá s ním principiálně takřka nic společného. Kompiluje se stejně jako každá aplikace v .NET frameworku do Microsoft Intermediate Language. Programovací přístup je zde podobný jako při programování aplikací pro windows (událostní programování). Při tvorbě webových stránek je tedy možno používat prvky jako je třeba nápis (label), tlačítko (button) a další. Na těchto prvcích se dají zachytávat události a přiřazovat jim vlastnosti. Webový protokol HTTP je bezstavový. K zachování stavu se v ASP.NET používá kombinace (X)HTML a javascriptu. Stav je zachován pomocí dvou technik nazvaných View State a Session State. View State uchovává data mezi postbacky (odeslání formuláře zpět na server) v zakódovaném tvaru ve skrytých formulářových polích. Jeho nevýhodou je, že se zvýší přenos dat. Proto se View State vypíná všude, kde ho není zapotřebí. Session State ukládá veškeré informace na straně serveru a předává (nejčastěji pomocí cookie) pouze jednoznačný identifikátor. To sice zmenšuje objem přenášených dat, ale klade vyšší nároky na výkon serveru.

Výhody ASP.NET:

- Striktně objektový přístup.
- Oddělení kódu od designu.
- Zdrojový kód je možno ve více programovacích jazycích
- Vynikající vývojový nástroj VisualStudio, které je ve verzi Express zadarmo.
- Řešení řady problémů je přímo součástí ASP.NET (autentizace, cachování stránek, validátory...)
- Možnost vytváření vlastních kontrolů, nebo zdědění hotových a následné upravení.
- Generuje validní XHTML od verze 2.0.
- Je to kompilovaný jazyk.

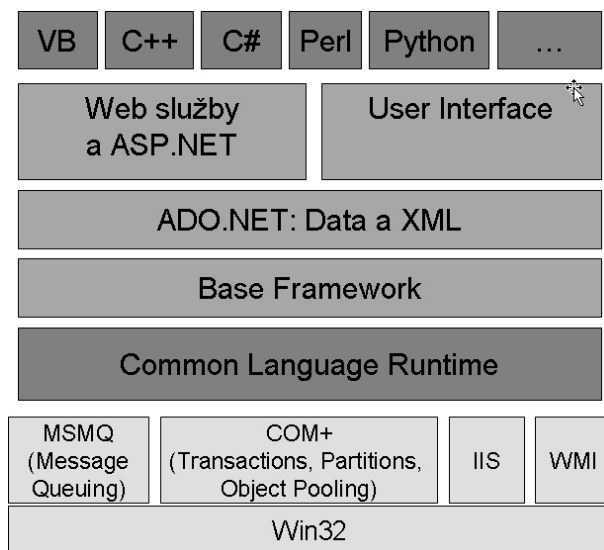
Nevýhody ASP.NET:

- Vyžaduje povědomost o objektově orientovaném programování a základech používané technologie obecně.
- Do verze 2.0 nebyly zdarma k dispozici vhodné vývojové nástroje.
- Vyšší režie při prvním spuštění. Stránka se musí překompilovat při každé změně.
- Vyšší nároky na hardware serveru.

V několika dalších podkapitolách uvedu popis .NET frameworku a několika částí ASP.NET.

.NET framework

Platforma Microsoft .Net je technologie od společnosti Microsoft pro vývoj aplikací, určených pro windows, ale i pro web. Je založena na takzvaném řízeném kódu (MSIL, Microsoft Intermediate Language), což není ani strojový jazyk ani zdrojový kód. Je to tzv. mezijazyk, který se vytvoří při kompilaci a při spuštění je zpracováván prostředím CLR (Common Language Runtime). Kód pro .NET je tedy možno psát v jakémkoli jazyce, pro který existuje překladač do tohoto mezikódu. Nejčastěji jsou to jazyky C#, C++ a Visual Basic, ale existují i další překladače například pro J#, PHP (vyvíjen studenty z Karlovy univerzity v Praze), Perl atd. Microsoft .NET framework byl původně pouze pro systémy Windows. Nyní existuje i multiplatformní verze frameworku nazývaná Mono.



Obrázek 4.1: .NET framework

Životní cyklus požadavku na stránku ASPX

Základní verze životního cyklu má 7 fází:

1. Start - inicializace vlastností Request, Response, nastavení Themes a Master pages
2. Page initialization - vytváření serverových ovládacích prvků, aplikace témat
3. Load - načtení vlastností y ViewState/ControlState
4. Validation - validace stránky
5. Postback event handler - vykonání událostí
6. Rendering - vygenerování výsledného (X)HTML kódu
7. Unload

Těchto sedm fází zaštiťuje přibližně 30 různých metod a událostí, které jsou během cyklu volány a které je možno změnit. Tím pádem můžeme upravit chování ASP.NET aplikace tak, aby se chovala přesně tak jak potřebujeme.

Serverové ovládací prvky

Jednou z hlavních částí ASP.NET jsou serverové ovládací prvky (server controls). Jsou to třídy .NET frameworku reprezentující vizuální prvky na webovém formuláři. Některé jsou přímočaré a renderují se přímo na konkrétní značku (X)HTML, jiné jsou mnohem komplexnější a realizují skupinu (X)HTML prvků.

Pokud chceme některý ovládací prvek renderovat jinak než se renderuje standardně, stačí napsat třídu zděděnou od třídy „System.Web.UI.WebControls.Adapters.WebControlAdapter” a přepsat její metodu „Render” a pomocí tak zvaného „browser souboru” (soubory určující třídy pro renderování jednotlivých ovládacích prvků pod jednotlivými prohlížeči) zapíšeme, aby se daný ovládací prvek renderoval pomocí námi napsané třídy.

Je také možné si napsat vlastní serverový ovládací prvek. Jsou dvě možnosti jak toto učinit a to buď zdědit si některý hotový ovládací prvek a upravit jeho činnost, nebo zděděním od System.Web.UI.UserControl vytvořit vlastní. Při kompletní tvorbě nového server controlu můžeme používat již hotové controly a sloužit tedy nový prvek z více již hotových.

Vzory stránek a motivy

Vzory stránek (master pages) umožňují vytvářet opětovně využitelné šablony stránky. Pomocí master pages můžeme definovat rozložení stránek. Každá master page obsahuje minimálně jeden tzv. ContentPlaceHolder, do kterého je pak vkládán obsah stránek zděděných od dané master page. Je možné i zanořování více vzorových stránek do sebe. Po připravení vzoru stránek ho můžeme použít všude na webu, čímž zajistíme, že všechny stránky budou mít stejný design.

Motivy (themes) mají podobnou funkci jako CSS (Cascading Style Sheets). Umožňují definovat pro různé serverové ovládací prvky různé formátovací detaily a tyto formátovací detaily opětovně využívat na více stránkách. Motivy tedy slouží k formátování serverových ovládacích prvků ASP.NET a pro formátování (X)HTML značek se používá CSS přiřazené k danému motivu. Pro jednu aplikaci můžeme vytvořit více motivů a ty se poté dají přepínat za běhu aplikace. Toto se dá využít například pro vypnutí grafiky.

Navigace

ASP.NET 2.0 obsahuje několik komponent pro navigaci na webu. Jsou to hlavně Menu, Treeview a SiteMapPath, které slouží pro zobrazení menu webové aplikace. Váží se na xml soubor, který se nazývá plán webu (sitemap). Plán webu umožňuje definovat navigační strukturu webu.

Správa uživatelů a rolí

V ASP.NET je integrován systém pro správu uživatelů, rolí a profilů. Tento systém je generický tzn. je použitelný takřka pro každou aplikaci. Z těchto důvodů je příliš složitý a robustní. Byl navržen tak, aby bylo možno provozovat současně více aplikací na jedné databázi. Do databáze by jsme tedy museli přidat tabulky, pohledy a uložené (stored) procedury, které tento systém vyžaduje (asi 11 tabulek, 50 stored procedur a několik pohledů). Na vytvoření této databázové struktury slouží nástroj aspnet_regsql.exe, který je umístěn v adresáři:

```
WINDOWS\Microsoft.NET\Framework\v2.0.(verze frameworku)\
```

Po případě jsou k dispozici databázové skripty umístěné v témže adresáři.

Tento způsob implementace je pro mojí aplikaci příliš složitý. ASP.NET umožňuje díky svému objektově orientovanému modelu předělat téměř jakoukoli část svých integrovaných prostředků. Stačí tedy napsat vlastní, nebo použít jiné třídy pro poskytování členství a rolí. Pro mojí aplikaci tedy použiji Altairis Simple ASP.NET SQL Providers (viz [?]AltairProviders)), které jsou podstatně jednodušší a praktičtější pro informační systém obce. Navíc můžeme zasahovat do databáze a přidávat vlastní sloupce k tabulkám.

Ukládání do cache

Další integrovanou vlastností ASP.NET je možnost zapnout cachování. Cachování znamená ukládání do paměti již vygenerovaných (X)HTML výstupů stránky, nebo jejich části, nebo libovolné jiné objekty. Můžeme si také přizpůsobit expirační politiky, aby se položky z cache automaticky odstraňovali. Např. na základě modifikací zdrojů (databáze), nebo po určitém čase. Při správném užití této technologie můžeme výrazně snížit zátěž nejen webového, ale také databázového serveru.

4.1.3 XHTML

XHTML (extensible hypertext markup language) je značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW. Je to formát založený na SGML (Standard Generalized Markup Language), využívající tagy ke strukturování textu do nadpisů, odstavců, tabulek atd. Je následníkem jazyka HTML, jehož vývoj byl ukončen, a na rozdíl od svého předchůdce se jedná o aplikaci XML.

Kód psaný v XHTML je oproti předcházejícímu HTML přehlednější, sémantičtější a lépe se v něm orientuje. Mezi základní pravidla XHTML patří [5]:

- Všechny tagy a atributy se musí psát malými písmeny
- Všechny tagy musí být párové, každý tag musí být uzavřen (
)
- Hodnoty atributů musí být v uvozovkách
- Nepoužívá se minimalizace atributů
- Tagy se nesmí nikdy křížit
- Nepoužívají se žádné formátovací tagy (align, font, b, i), používají se CSS
- Znak & musí být převeden na html entitu.

4.1.4 CSS

Kaskádové styly CSS (Cascading Style Sheets) vznikly jako souhrn metod pro úpravu vzhledu stránek. CSS se využívá k formátování obsahu HTML, XHTML a XML dokumentů. Ve srovnání s formátováním pomocí atributů v HTML se formátovací schopnosti rozšiřují. Styly umožňují přesně určit, jak bude který element vypadat. Je možno definovat jak jednotný vzhled elementů pro celý dokument tak i vzhled jediného výskytu elementu.

Výhody CSS:

- Dobře strukturované dokumenty, oddělení obsahu od formy

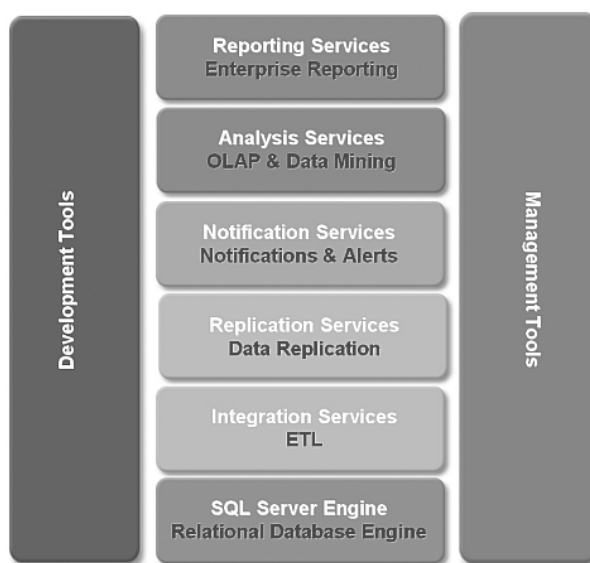
- Přehledný a jednoduchý kód
- Snadnou kompletní změnu vzhledu stránek pouze pomocí CSS
- Menší datovou velikost stránky
- Rychlejší načítání stránek.

Nevýhodou CSS je špatná podpora v některých prohlížečích. Různé prohlížeče interpretují stejný CSS kód jinak a je někdy velmi obtížné jej napsat tak, aby se na všech (resp. na několika vybraných) prohlížečích výsledek zobrazil stejně. Nejčastějším zdrojem těchto problémů je Internet Explorer, který striktně nedodržuje specifikaci CSS.

4.1.5 Microsoft SQL server 2005

Databáze je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu. V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento systém se v české odborné literatuře nazývá systém řízení báze dat (SŘBD). Běžně se označením databáze - v závislosti na kontextu - myslí jak uložená data, tak i software (SŘBD).

Komplexnost SQL Serveru 2005 nejlépe vyjadřuje blokové schéma jeho architektury (viz obrázek 4.2). Základním stavebním prvkem architektury je relační databázový stroj. Má na



Obrázek 4.2: MS SQL server 2005

starosti správu údajů, jejich vkládání, vymazávání a vyhledávání ...

Dále pak:

- reportní služby (generování reportů a sestav na základě údajů z databáze)
- analytické služby (OLAP analýzy, datamining)
- notifikační služby

- replikační služby
- služby pro transformaci údajů

K těmto šesti základním blokům můžeme přiřadit bloky pro administraci a vývoj aplikací. Databázový server s naznačenou infrastrukturou najde uplatnění nejen pro budování rozsáhlých relačních databází, ale také analytických databází a datových skladů (datawarehouse).

Podporuje vyváření transakcí, uložených (stored) procedur, funkcí a triggerů v procedurální nadstavbě jazyka Transact-SQL (T-SQL tj. obdoba Oraclovského PL/SQL).

Ve verzi Express je k dispozici zdarma. Tato verze má pouze několik omezení:

- Používá maximálně jeden procesor (lze nainstalovat i na server s více procesory, ale používat bude pouze jeden).
- Používá maximálně 1GB operační paměti.
- Velikost databáze je maximálně 4GB.
- 64 bitové procesory podporuje pouze v tzv. WoW režimu.

4.1.6 MySQL

MySQL je multiplatformní databázový systém vytvořený švédskou firmou MySQL AB. Pro svou snadnou implementovatelnost, výkon a také díky tomu, že je to volně šiřitelný software se dnes hojně používá (hlavně pro webové aplikace). MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení. Až donedávna nepodporovalo pohledy, triggerů, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech.

4.1.7 WYSIWYG editor

WYSIWYG (What you see is what you get) označuje způsob editace dokumentů v počítači, při kterém je výsledný dokument stejný jako verze kterou vidíme na obrazovce. Na internetu je mnoho různých editorů tohoto typu viz [4].

4.1.8 Závěr

Pro začátky s technologií ASP.NET je zapotřebí podstatně větších znalostí než u PHP, ale po získání těchto znalostí je podle mého názoru vývojář schopen vyvíjet aplikace s vyšší efektivitou než s PHP. Navíc pak není problémem přejít od programování webových aplikací k programování aplikací určených pro desktop. Nejen proto použijí pro tuto práci technologii ASP.NET verze 2.0.

Při práci s ASP.NET se lépe pracuje s Microsoft SQL, proto ho použijí raději než MySQL.

Jako WYSIWYG editor jsem si vybral FCKeditor (viz [1]), který se mi zdál být nejzajímavější. Je napsán v JavaScriptu, podporuje různé prohlížeče, má spoustu užitečných funkcí a generuje validní XHTML kód. Jako jednu z jeho hlavních předností беру možnost vkládání textu z programu Word což si myslím, že bude velice užitečná funkce pro informační systém obce.

4.2 Implementace databáze

Navrhovaná databáze celkem obsahuje 8 tabulek a 12 stored procedur. Stored procedury zde slouží pro složitější dotazy a pro vkládání, editaci a mazání dat. U každé tabulky uvedu její krátký popis a procedury příslušící k dané tabulce.

4.2.1 Tabulka Články

Data v této tabulce jsou použity k uchování základního obsahu stránek, článků a příspěvků na úřední desce. Jednotlivé atributy jsou tedy „nadpis“, „autor“, „email“ (email autora), „text“ pro uchování vlastního X(HTML) obsahu článku, „vlozeno“ a „zmeneno“ pro uložení datum a času vložení a poslední změny článku. Dále je zde primární klíč složený z „ID_clanku“ a cizího klíče „ID_typu“, sloužící jako reference do tabulky typ článku.

K této tabulce náleží stored procedury „novyClanek“, „delClanek“, „upravClanek“ a „strClanky“. Uvedu popis pouze poslední z nich, která slouží pro účely stránkování článků. Podle parametrů (typ článku, strana a počet po kolika řádkách se má stránkovat) vrátí pouze potřebnou stránku. Tato procedura používá funkci ROW_NUMBER(), která očísluje řádky v tabulce a následně vybere pouze potřebné řádky.

4.2.2 Tabulka Typ článku

Určuje typ daného článku. Obsahuje pouze dva atributy. Jsou to primární klíč „ID_typu“ a „nazev“.

4.2.3 Tabulka Odkazy

V této tabulce se ukládají odkazy a jejich popisky. Obsahuje atributy pro informace o odkazu, odkaz a text pro odkaz. Pro přidávání, změnu a mazání dat z tabulky jsou v databázi k dispozici uložené procedury „novyOdkaz“, „upravOdkaz“ a „delOdkaz“.

4.2.4 Tabulky Album a Fotky

Slouží pro zpracování fotogalerie. Tabulka album obsahuje cestu k adresáři, kde je uložena daná galerie a informace o této galerii. Druhá tabulka slouží k uložení názvů souborů s fotkami a jejich popisků. Pro tyto tabulky existují uložené procedury pouze pro vkládání nových prvků („noveAlbum“ a „novaFotka“).

4.2.5 Tabulka Akce

Tabulka je použita pro zachování informací o akcích, které budou probíhat v obci. Její atributy jsou „nazev“ (název akce), „info“ (upřesňující informace o akci), „datum“ konání akce a složený primární klíč z „ID_akce“ a cizího klíče „ID_typu“. Pro tuto tabulku jsou k dispozici uložené procedury „novaAkce“, „upravAkce“, „delAkce“ a „getAkce“. První tři jsou klasickou reprezentací příkazů INSERT, UPDATE a DELETE. Procedura „getAkce“ má 3 parametry „od“, „do“ určující rozmezí datumů pro vyhledávání a „ID_typu“, které určuje typ akce a které může nabývat speciální hodnoty pro zobrazení všech typů.

4.2.6 Tabulka Typ akce

Tabulka určující typ akce. Její struktura je obdobná jako u tabulky „Typ článku“.

4.2.7 Tabulka vzkazy

Používá se pro uložení příspěvků v knize návštěv. Obsahuje atributy „nick” (přezdívka uživatele), „email” (e-mailová adresa uživatele), „text” (text příspěvku), „IP_adresa” pro uložení IP adresy a případné zjištění uživatele, „datum” (datum a čas přidání příspěvku) a primární klíč „ID_vzkazu”. Pro tabulku jsou použity stored procedury „novyVzkaz”, „delVzkaz” a „strVzkazy”. Poslední z uvedených slouží pro stránkování. Struktura procedury je podobná jako u „strClanky” (viz 4.2.1).

4.2.8 Tabulka Users

Slouží pro uložení informací o uživateli. Struktura tabulky je definována v dokumentaci k Altairis Simple ASP.NET SQL Providers [9]. K této struktuře navíc přidám čtyři vlastní sloupce „jmeno”, „prijmeni” (pro uložení jména a příjmení uživatele), „titul_pred” a „titul_za”. Do posledních dvou sloupců se ukládají eventuelní tituly uživatele.

4.2.9 Tabulka Roles

Tabulka obsahuje pouze jeden sloupec a tím je „RoleName” (viz [9], ve kterém jsou uloženy názvy jednotlivých rolí (uživatelských skupin).

4.2.10 Tabulka Users in Roles

Vazební tabulka mezi dvěma předchozími. Slouží k přiřazení rolí k uživatelům. Více informací najdeme opět v dokumentaci k Altairis Simple ASP.NET SQL Providers [9].

4.3 Implementace jednotlivých modulů a stránek

Tato část bakalářské práce popisuje implementaci jednotlivých modulů.

4.3.1 Master page

Všechny stránky používají jednotnou master page. Tím docílíme standardizovaného vzhledu. Master page pro tento informační systém má název „obec.master”. Obsahuje kromě hlavičky, zápatí a menu i ContentPlaceholder, kam se po použití master page vkládá obsah všech dalších stránek. Rozmístění je provedeno pomocí tabulky.

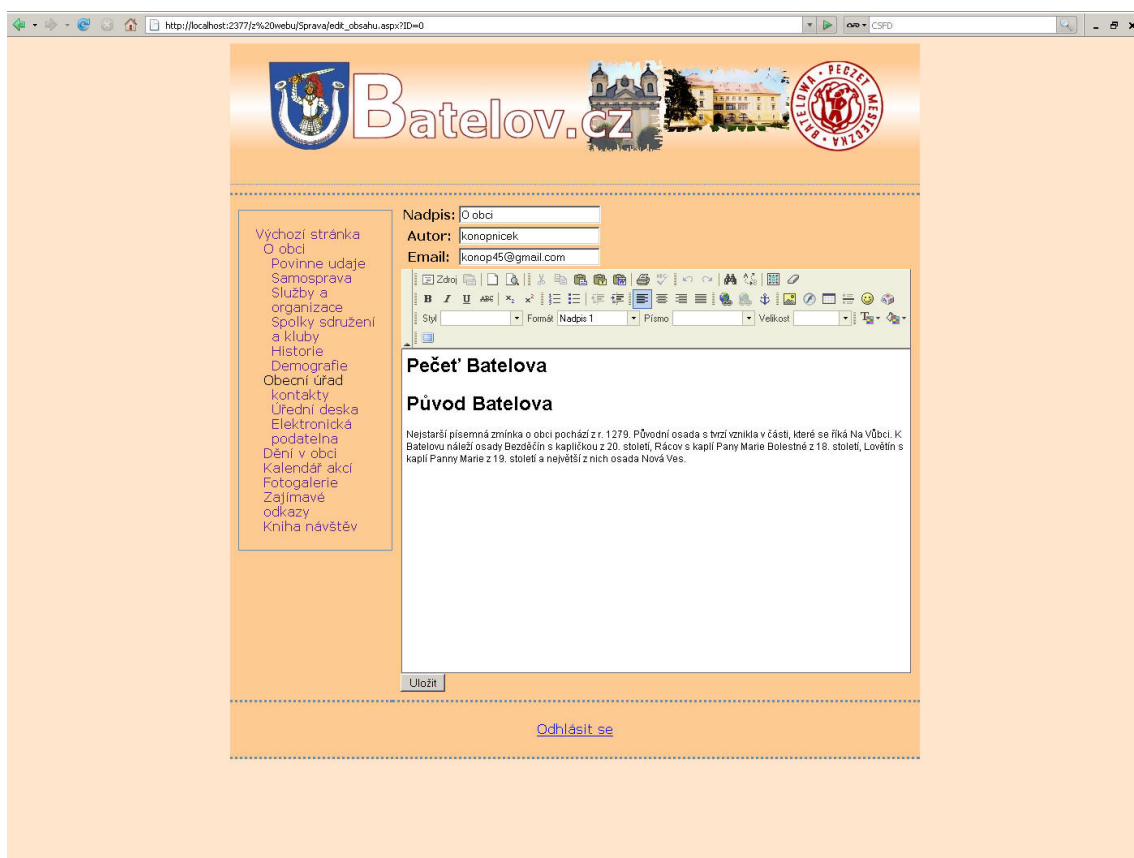
4.3.2 Obsah stránek

Obsah stránek je uložen v databázové tabulce „clanky” podle atributu „ID_typu” poznáme, do které kategorie v menu článek spadá. Modul je tvořen stránkou „clanek.aspx”. Pro zobrazení jednoho údaje z tabulky jsem si vytvořil vlastní serverový ovládací prvek nazvaný „clanek.ascx”. Je to třída, která je zděděna od „System.Web.UI.UserControl”. Má několik vlastností, které slouží pro určení databáze, tabulky, názvu a způsobu získání parametru, podle kterého se získají data z databáze. Obsahuje pět nápisů (labelů), pro zobrazení nadpisu, textu, autora, e-mailu autora a datumu vložení (změny) článku. Tyto nápisy jsou v metodě PageLoad ovládacího prvku vyplněny hodnotami z databáze. Podle vlastnosti „zobrazitVse” se zobrazí buď všechny labely nebo jen label s textem (slouží pro zobrazení obsahu stránek).

ASP.NET standardně renderuje text prvku label do tagu ``, který nemůže obsahovat blokové tagy, což je pro zobrazení obsahu stránek nedostačující. Proto jsem napsal třídu, která renderuje text z labelu přímo tak jak je zapsán do výsledného kódu stránky. V případě kdy má label zadánu vlastnost `cssClass` se renderuje klasicky jako `span`.

Uživatelé ze skupiny „Správce obsahu stránek“ mají navíc pod obsahem textu zobrazeno tlačítko, které je odkáže na stránku „editace_clanku.aspx“ (viz obrázek 4.3). Na této stránce je použit mnou vytvořený ovládací prvek pro editaci článků. Obsahuje tři textboxy pro zadání nadpisu, autora a emailu a již dříve zmiňovaný WYSIWYG editor FCKeditor (viz 4.1.7). Načtení probíhá stejně jako u prvku pro zobrazení článků v události `Page_Load`. Pro uložení změn je zde tlačítko „Uložit“.

Pro zobrazovací část tohoto modulu můžeme zapnout cachování stránky. Čas cachování nastavíme na 30 sekund, aby jsme nemuseli dlouho čekat při editaci obsahu. Tímto zajistíme snížení zátěže serveru.



Obrázek 4.3: Editace obsahu

4.3.3 Úřední deska

Úřední deska („uredni_deska.aspx“) je zobrazena pomocí ovládacího prvku `Repeater`, který slouží k vypisování hodnot z databáze pomocí šablon (template). `Repeateru` je předáván `SqlDataSource` což je prvek, který umožňuje spojení s databází a volání dotazu, nebo uložené procedury. `SqlDataSource` je použit k volání stored procedury „strClanky“ s potřebnými

parametry. Pro stránkování jsem opět vytvořil vlastní serverový ovládací prvek, který do URL adresy doplňuje parametr s číslem stránky. Ke změně tohoto parametru používá hyperlinků.

Následuje krátká ukázka kódu šablony prvku Repeater:

```
<ItemTemplate>
  <asp:Label ID='lbl_autor' cssClass='uredniDeska_autor'
    Text='<\\%# DataBinder.Eval(Container.DataItem, 'autor')+ ': ' \\%>'
    runat='server'>
  </asp:Label>
  <asp:Label ID='lbl_nadpis' cssClass='uredniDeska_nadpis'
    Text='<\\%# DataBinder.Eval(Container.DataItem, 'nadpis') \\%>'
    runat='server'>
  </asp:Label>
  <br />
  <asp:Label ID='lbl_vlozeno' cssClass='uredniDeska_vlozeno'
    Text='<\\%# 'Vloženo: ' +
    DataBinder.Eval(Container.DataItem, 'vlozeno') \\%>'
    runat='server'>
  </asp:Label>
  <br />
  <asp:Button ID='btn_detail' cssClass='uredniDeska_Detail'
    Text='Zobrazit Detail' runat='server'
    CommandArgument='<\\%# DataBinder.Eval(Container.DataItem,
    'ID_clanku') \\%>' CommandName='detail'
  />
</ItemTemplate>
```

Tato šablona se aplikuje pro každý řádek z tabulky vrácený SqlDataSourceem. Pomocí `<%#DataBinder.Eval(Container.DataItem, "název sloupce")%>` se přistupuje k jednotlivým prvkům z databáze. Tomuto přístupu se u ASP.NET říká DataBinding. Pro zobrazení detailu je k dispozici tlačítko, které pomocí cross-side postbacku (odeslání postback požadavku na jinou stránku) zobrazí stránku „detail.aspx”, která používá stejně jako modul „Obsah stránek” komponentu „clanek.ascx” sloužící pro zobrazení článku z databáze. Cross-side postback je zde použit proto, aby se uživatelé nedostali pomocí „detail.aspx” k jiným článkům (např. obsahu stránek).

V případě přihlášení uživatele ze skupiny „Správce úřední desky” je uživatel přesměrován na stránku pro správu úřední desky, na které jsou oproti normální úřední desce kromě tlačítka pro zobrazení detailu navíc tlačítka sloužící k přidání, editaci a smazání příspěvku.

4.3.4 Články

Modul pro zobrazení článků z dění v obci je funkčně stejný jako úřední deska.

4.3.5 Elektronická podatelna

Elektronická podatelna je pouze formulář sloužící k odeslání e-mailu na adresu elektronické podatelny dané obce. Obsahuje tři textboxy pro e-mail autora, předmět a text e-mailu. Dále obsahuje tlačítko sloužící k odeslání.

4.3.6 Odkazy

Tato součást webové aplikace zobrazuje za pomoci již popisovaného prvku Repeater odkazy. Administrační část tohoto modulu je tvořena formulářem pro nový odkaz a prvkem GridView, který umožňuje vypsání hodnot z databáze do tabulky. Kromě toho má spoustu dalších užitečných funkcí např. umožňuje editaci a mazání jednotlivých řádků, seřazení tabulky atd. GridView se stejně jako Repeater váže na SqlDataSource. Pokud chceme používat možnost mazání a editace řádků pomocí GridView musíme pro ně vytvořit odpovídající SQL dotazy v SqlDataSource.

4.3.7 Fotogalerie

Modul se skládá ze čtyřech stránek. Dvě pro zobrazení a dvě pro administraci. Stránky pro zobrazení jsou opět tvořeny za pomoci již dobře známého prvku Repeater. První stránka („alba.aspx“) zobrazuje za pomoci šablony Repeateru informace o fotogaleriích a odkazy na stránku („fotogalerie.aspx“) sloužící pro zobrazení fotek z galerie. Administrační část pro alba je tvořena tabulkovým výpisem prvku GridView s možností editace a mazání. Správa fotek v daném albu je vytvořena opět prvkem GridView. Pro zlepšení funkčnosti je zde tlačítko pro načtení fotek z daného adresáře příslušícího albu. Při kliknutí na toto tlačítko je naplněna databáze jmény souborů, ale také proběhne skript, který případně zmenší velké obrázky na přijatelnou velikost. Nejjednodušší způsob jak tedy přidat galerii je:

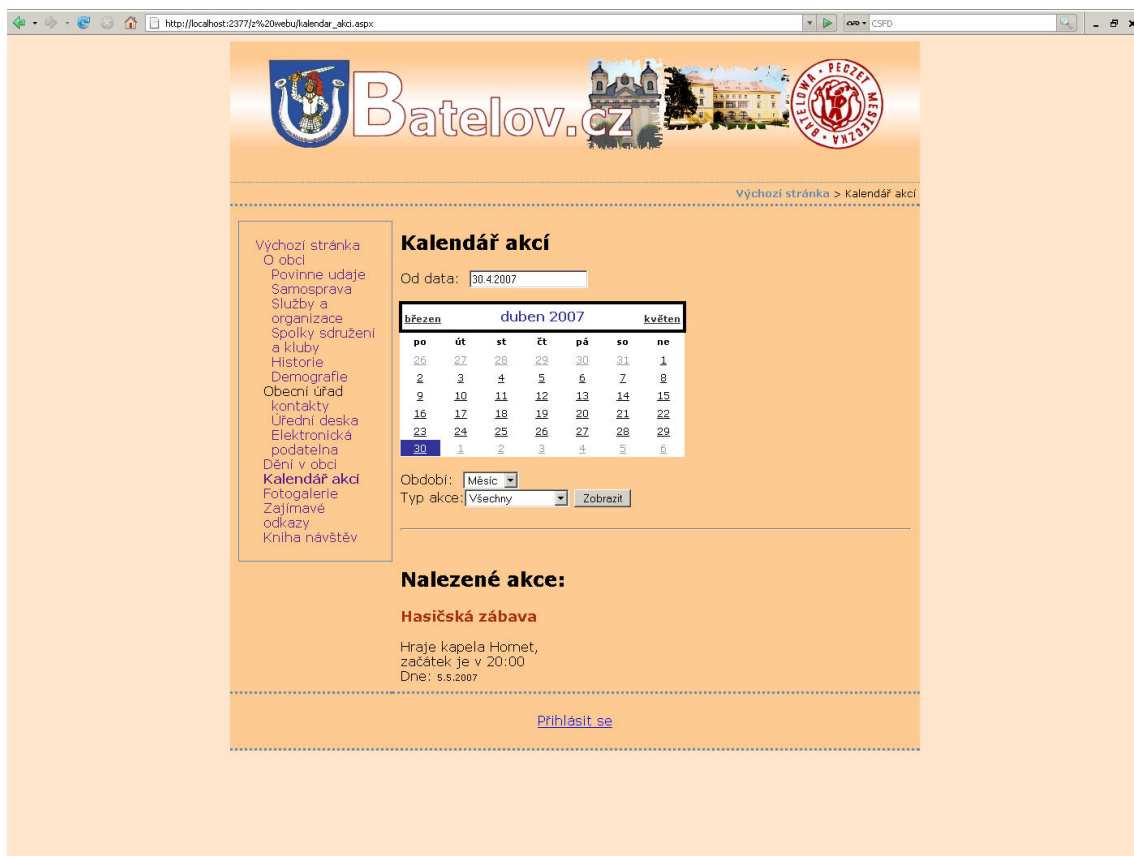
- Nahrát fotky na webový server např. za pomoci FTP připojení
- Přidat album
- Ve správě fotogalerie kliknout na tlačítko pro vyplnění z adresářové struktury.
- Případně ještě vyplníme popisky jednotlivých fotek.

4.3.8 Kalendář akcí

Tento modul používá k vyhledání akcí textbox a komponentu kalendář pro zadání datumu, rozbalovací nabídku (dropDownList) pro výběr období, na které chceme akce vyhledat (týden, měsíc nebo rok), rozbalovací nabídku sloužící k výběru typu akce (kulturní, sportovní, kino atd.) a tlačítko pro vyhledání akcí (obrázek 4.5). Vyhledané akce se vypisují opět pomocí prvku Repeater za přispění SqlDataSource a jeho parametrů získaných z předchozích ovládacích prvků. SqlDataSource volá stored proceduru s názvem „getAkce“, která má tři parametry „od“, „do“ typu DateTime sloužící pro rozmezí vyhledávaných akcí a parametr „typ“ určující typ akce. Poslední z uvedených může kromě standardních čísel pro typy obsahovat hodnotu -1. Při této hodnotě procedura vrátí akce všech typů. Stránka pro administrační rozhraní obsahuje navíc v šabloně Repeateru dvě tlačítka pro editaci a smazání akce. Dalším prvkem na administrační stránce je tlačítko pro přidání nové akce, které provede přesměrování na stránku „nova_akce.aspx“. Ta obsahuje formulář pro vyplnění informací o akci a tlačítko pro uložení.

4.3.9 Kniha návštěv

Je tvořena stránkou „Kniha_navstev.aspx“, která se skládá z komponenty pro stránkování, Repeateru pro vypsání příspěvků a formuláře pro vyplnění přezdívky, e-mailu, textu příspěvku a tlačítka pro odeslání. Pro získání dat z databáze je volána uložená procedura „strVzkazy“.



Obrázek 4.4: Editace obsahu

4.3.10 Správa uživatelů a rolí

Správa je tvořena několika stránkami. Hlavní z nich je „uzivatele.aspx“ slouží k přiřazování rolí uživatelům, přidávání nových a mazání starých uživatelů. Je tvořena pomocí rozbalovací nabídky s uživatelskými jmény a checkListBoxu (skupina zaškrtačkových políček s popisky) se seznamem rolí a pěti tlačítky. Tři tlačítka slouží pro správu uživatelů (přidání, smazání a zobrazení informací o uživateli). Zbylé dvě jsou pro vyznačení rolí uživatele zvoleného v rozbalovací nabídce a pro přiřazení vybraných rolí k danému uživateli. Přidání uživatele probíhá pomocí stránky „novy_uzivatel.aspx“, kde se nachází formulář pro údaje o uživateli a tlačítko pro uložení.

4.4 Nastavení přístupových práv ke stránkám

Přístupová práva se v APS.NET nastavují v souboru „web.config“, který může být umístěn v každém adresáři daného webu, přičemž se aplikuje pro tento adresáři pro jeho podadresáře. Pokud se některý uživatel snaží přistupovat ke stránce kam nemá přístup je automaticky přesměrován na stránku sloužící k přihlášení.

Všechny stránky, které implementují správu modulů, umístíme do adresáře „/Sprava/“. Následně nastavíme práva pro tento adresář, kde zakážeme přístup všem anonymním uživatelům přidáním konfiguračního souboru do tohoto adresáře. Obsah konfiguračního souboru vypadá



Obrázek 4.5: Správa uživatelů

nějak takto:

```
<configuration>
  <system.web>
    <authorization>
      <deny users='?' />
    </authorization>
  </system.web>
</configuration>
```

Nyní musíme nastavit práva jednotlivých uživatelských skupin pro dané stránky s moduly. Toto nastavujeme přidáním textu do sekce „<configuration >“. Tento text vypadá například pro stránku „novy_clanek.aspx“ takto:

```
<location path='novy_clanek.aspx' >
  <system.web>
    <authorization>
      <deny users='?' />
      <allow roles='administrator' />
      <allow roles='spravce_clanku' />
      <allow roles='spravce_uredni_desky' />
      <deny users='*' />
    </authorization>
  </system.web>
</location>
```

```
</authorization>  
</system.web>  
</location>
```

Tímto způsobem nastavíme přístupová práva pro všechny stránky administrace.

Kapitola 5

Závěr

Cílem této práce bylo vytvoření takového informačního systému, který bude sloužit pro malé obce. Jedním z hlavních úkolů bylo vytvoření jednoduché správy tohoto informačního systému tak, aby mohli jednotliví uživatelé používat systém bez větších znalostí daných technologií.

K získání zpětné vazby jsem po dokončení práce nechal vytvořenou aplikaci odzkoušet několika nezávislými laickými uživateli. Odezva byla vesměs kladná. Přípomínky a návrhy jsem vyhodnotil a případně zpracoval.

Ačkoliv je na současném trhu velká nabídka produktů podobného zaměření, myslím si, že můj systém by mohl být atraktivní pro malé obce, zejména pro jeho snadné ovládání.

Předkládaný systém umožní uživatelům snadno přistupovat k informacím o dané obci. Registrovaný uživatelům umožní správu některých částí aplikace, podle rolí ke kterým daný uživatel patří. Správce systému má přístup k administraci celého systému, může také vkládat nové uživatele a přiřazovat jim role.

V případě zájmu obce, pro kterou by se systém používal, je možno naimplementovat další funkce jako např. zobrazení anket, fulltextové vyhledávání, RSS kanál atd.

Díky této bakalářské práci jsem si prohloubil znalosti z oblasti informačních systémů obcí, znalosti týkající se ASP.NET, MS SQL a samozřejmě také znalosti z oblasti návrhu informačních systémů.

Literatura

- [1] FCKeditor, the text editor for Internet. [online], 2003-2007 [cit. 2007-05-11].
URL <http://www.fckeditor.net/>
- [2] Nová pravidla pro úřední desky. [online], 2006 [cit. 2007-05-11].
URL <http://www.denik.obce.cz/go/clanek.asp?id=6207802>
- [3] Toad Data Modeler. [online], 2000-2007 [cit. 2007-05-11].
URL <http://www.casestudio.com/csy/default.aspx>
- [4] TTW (“Through the Web”) WYSIWYG Web Editors - The List. [online], 1996-2007 [cit. 2007-05-11].
URL <http://www.geniisoft.com/showcase.nsf/WebEditors>
- [5] XHTML eXtensible HyperText Markup Language. [online], 2004 [cit. 2007-05-11].
URL <http://www.webtvorba.cz/xhtml/>
- [6] MacDonald, M.; Szpuszta, M.: *ASP.NET 2.0 a C# tvorba dynamických webových stránek profesionálně*. Zoner press, 2006, ISBN 80-86815-38-2.
- [7] Nápravník, J.: Elektronická podatelna: Prostor pro kšeft? [online], 1999-2007 [cit. 2007-05-11].
URL <http://www.mesec.cz/clanky/elektronicka-podatelna-prostor-pro-kseft/>
- [8] Schmuller, J.: *Myslíme v jazyce UML*. Grada, 2001, ISBN 80-86815-38-2.
- [9] Valášek, M.: Altairis Simple ASP.NET SQL Providers. [online], 2000-2007 [cit. 2007-05-11].
URL <http://www.aspnet.cz/Articles/115-simple-providers.aspx>

Příloha A

Popis instalace informačního systému na server

1. Nakopírujeme soubory se zdrojovými kódy informačního systému na server.
2. Vytvoříme databázovou strukturu pomocí skriptu genDB.sql.
3. V souboru web.config v rootu webu nastavíme connectionString k příslušné databázi.