

Administrativní Informační systém pro časopisové nakladatelství

A Content Management System for a Journal Publisher

Milan Kleibel

Bakalářská práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Milan KLEIBEL**
Osobní číslo: **A09050**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Administrativní Informační systém pro časopisové nakladatelství**

Zásady pro vypracování:

1. Zpracujte požadavky a potřeby vydavatelství.
2. Navrhněte a popište použité technologie (jejich výhody a nevýhody).
3. Proveďte návrh aplikace na základě požadavků.
4. Vytvořte funkční prototyp aplikace podle návrhu.
5. Vyhodnoťte vytvořený prototyp a uveďte technický a ekonomický přínos.
6. Navrhněte další možný rozvoj systému.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **MACDONALD, Matthew a Mario SZPUSZTA. ASP.NET 3.5 a C-SHARP 2008: tvorba dynamických stránek profesionálně.** Vyd. 1. Brno: Zoner Press, 2008, 1584 s. ISBN 978-807-4130-083.
2. **JQuery. JQuery: kuchařka programátora [online].** Vyd. 1. Brno: Computer Press, 2010 [cit. 2012-01-17]. Dostupné z: <http://jquery.com/>
3. **MSDN Library: SQL Server Language Reference. MICROSOFT. MSDN Library [online].** 2008 [cit. 2012-01-17]. Dostupné z: <http://msdn.microsoft.com/cs-cz/library/default.aspx>
4. **C-SHARP Programming Guide. Microsoft Visual Studio [online].** 2008 [cit. 2012-01-17]. Dostupné z: <http://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>
5. **JavaScript. REFSNES, Hege. Learn JavaScript and Ajax with w3schools [online].** Hoboken, N.J.: Wiley, c2010 [cit. 2012-01-17]. Dostupné z: <http://www.w3schools.com/>
6. **CSS. REFSNES, Hege. W3schools [online].** Hoboken, N.J.: Wiley, c2010 [cit. 2012-01-17]. Dostupné z: www.w3schools.com
7. **Stack Overflow. Stackoverflow [online].** 2008 [cit. 2012-01-17]. Dostupné z: <http://stackoverflow.com/>

Vedoucí bakalářské práce:

Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

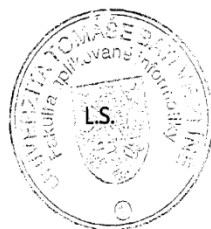
24. února 2012

Termín odevzdání bakalářské práce:

8. června 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Cílem bakalářské práce bylo navrhnout a vyvinout Administrativní informační systém pro časopisové nakladatelství. Výsledná aplikace byla napsána v jazyce C#. Klíčové pro systém je realizovat často používané operace (například vyhledávání, řazení, editace,...) v javascriptu, přesněji v jQuery. Tato aplikace umožňuje prohlížení, editaci a nastavení vazeb mezi jednotlivými časopisy, vydáními časopisů a rubrikami. Dále nastavení akcí, které symbolizují proces zhotovení vydání a tvorba časového harmonogramu. V Administraci není kladen důraz na obsah rubrik ani na aktuální schvalovací proces jednotlivých vydání. Při návrhu a vývoji Administrativního systému byl kladen důraz na rychlost aplikace a jednoduchost ovládání.

Tato aplikace byla vyvinuta ve vývojovém prostředí Microsoft Visual Studio 2010 pomocí technologie ASP.NET a jako databázová platforma byl použit Microsoft SQL Server 2008.

Klíčová slova: ASP.NET, C#, Administrace, Informační systém, Microsoft Visual Studio 2010, Microsoft SQL Server 2008, javascript, jQuery

ABSTRACT

The goal of the bachelor thesis was to design and developer Administrative informatic system for magazine publishers. The resulting application was written in C# language. For system it's important to develop the most common used operations (such as search, sort, editation,...) in javascript more precisely jQuery. This application allows to view, edit and settings of relations between single magazines, issues and rubrics. Settings of actions which represents process of creation of the issue and time schedule. The administration system does not contain rubric content or issue workflow process itself. Main goal for developing administration system is being put on speed of application and user friendly controls.

This application was developer by using Microsoft Visul Studio 2010 including ASP.NET technology with Microsoft SQL Server 2008 as diabase platform.

Keywords: ASP.NET, C#, Administration, Information system, Microsoft Visual Studio 2010, Microsoft SQL Server 2008, javascript, jQuery

Rád bych poděkoval Ing. Petru Šilhavému, vedoucímu mé bakalářské práce, za odborné Vedení, pomoc, postřehy a rady při zpracování bakalářské práce.

Také bych rád poděkoval své rodině za morální podporu při tvorbě této bakalářské práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	11
1 POŽADAVKY A POTŘEBY VYDAVATELSTVÍ	12
2 POUŽITÉ TECHNOLOGIE	14
2.1 .NET FRAMEWORK	14
2.1.1 Historie .NET Frameworku.....	14
2.1.1.1 .NET Framework 1.0	15
2.1.1.2 .NET Framework 1.1	15
2.1.1.3 .NET Framework 2.0	15
2.1.1.4 .NET Framework 3.0	16
2.1.1.5 .NET Framework 3.5	16
2.1.1.6 .NET Framework 4.0	17
2.1.2 Architektura .NET Frameworku	19
2.2 ASP .NET	21
2.2.1 Vznik a vývoj ASP.NET	22
2.2.1.1 ASP.NET 2.0	22
2.2.1.2 ASP.NET 3.5	23
2.2.1.3 ASP.NET 4.0	24
2.2.2 ASP.NET Podrobněji	25
2.3 SQL (STRUCTURE QUERY LANGUAGE).....	27
2.3.1 Co je a jak funguje SQL?	27
2.3.2 Proč Microsoft SQL Server 2008 Express?	29
2.4 PROGRAMOVACÍ JAZYK C#	29
2.4.1 Historický vývoj C#	30
2.4.2 Výhody a nevýhody	30
2.5 JAVASCRIPT A JQUERY	31
2.5.1 Proč javascript a jQuery	31
2.5.2 Výhody a nevýhody	32
II PRAKTICKÁ ČÁST	33
3 STRUKTURA DATABÁZOVÉHO SYSTÉMU	34
3.1 LOGOVÁNÍ ZMĚN.....	36
4 REALIZACE ADMINISTRATIVNÍHO SYSTÉMU	38
4.1 NÁVRH A ROZVRŽENÍ WEBOVÉ APLIKACE	38
4.2 IMPLEMENTACE ADMINISTRATIVNÍHO SYSTÉMU.....	39
4.2.1 Přihlašovací obrazovka	39
4.2.2 Editace Uživatelů	40
4.2.2.1 Nový uživatelský účet.....	40
4.2.2.2 Editace uživatelského účtu.....	41
4.2.2.3 Vytvoření, mazání a obnovení účtu	42
4.2.2.4 Vyhledávání mezi účty	43
4.2.3 Editace magazínů	43
4.2.3.1 Filtrační podmínky.....	45
4.2.3.2 Panel Magazíny.....	46
4.2.3.3 Panel Vydání.....	47

4.2.3.4	Panel Rubriky	49
4.2.3.5	Panel Akce	50
4.2.3.6	Panel Harmonogramy	51
4.2.3.7	Práva	52
4.3	TISK HARMONOGRAMŮ	53
4.4	SPRÁVA ČÍSELNÍKŮ	54
ZÁVĚR		55
ZÁVĚR V ANGLIČTINĚ.....		56
SEZNAM POUŽITÉ LITERATURY.....		57
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....		59
SEZNAM OBRÁZKŮ		61
SEZNAM PŘÍLOH.....		63

ÚVOD

V dnešní moderní a internetové době je stále těžší se prosadit na trhu tištěných časopisů, které stojí leckdy nemalé peníze. Pro mnoho uživatelů je dnes jednoduché, a prakticky zadarmo, si potřebné informace nalézt na internetu sami, aniž by za to platily. Jsou zde ale časopisy, jejichž obsah se nedá jednoduše na internetu nalézt nebo je dostupný, ale ve formě, které průměrný uživatel nerozumí (např.: příliš odborné, v cizím jazyce,...). Ale i tyto časopisy, aby byly i nadále úspěšné a prosperovali, musí snižovat své náklady na výrobu a vývoj časopisu a jeho vydání spolu s jeho zrychlením. Toto není možné bez použití příslušných aplikací nebo informačních systémů.

Časopisová nakladatelství využívají počítače a informační systémy pro návrh a výrobu časopisů a jejich vydání stále více a více. To klade velký důraz na kvalitu, jednoduchost a rychlost těchto systémů a aplikací a také na vývojáře, kteří je vytvářejí. Tyto systémy se i přes svoji komplexnost stávají v dnešní době zastaralými a je proto nutná inovace softwarového řešení. Původní (v mnoha ohledech Windows okenní aplikace) jsou běžně několik let staré a aktualizace by byla příliš náročná. Proto se jeví jako lepší a ekonomičtější řešení vytvořit nový systém, který by lépe vyhovoval novým a moderním požadavkům. Rozhodnutí vytvořit administrativní systém, přístupný přes webové rozhraní je vhodné pro možnost snadného přístupu k tomuto systému a jednodušší správě systému. Problémem zůstává, jak nahradit rychlé a interaktivní okenní aplikace. Odpovědí může být použití jQuery knihovny, která dokáže provádět operace velmi rychle a reaguje na podněty na straně klienta a se serverem komunikuje asynchronně pomocí vestavěných funkcí (zabudovaná podpora AJAXu).

Administrativní systém spravující vazby mezi jednotlivými časopisy, vydáními, rubrikami a jejich zařazení do pracovního postupu a tvorbě harmonogramu je tématem právě této bakalářské práce.

Pro tvorbu mé práce jsem použil jeden z moderních a oblíbených programovacích prostředků, platformu Microsoft .NET. Pro asynchronní práci s daty a rychlou prací v rámci webové stránky je použita, dnes již oblíbená a výše zmiňovaná, knihovna jQuery.

Tato bakalářská práce je věnována implementaci Administrativního informačního systému s využitím technologií od firmy Microsoft. Úvod práce se věnuje technologiím, které byly použity pro realizaci bakalářské práce a popisuje historii vzniku těchto technologií, obeznámení s jejich použitím a jejich výhody a nevýhody. V praktické části je

popsána realizace této práce od návrhu databázové struktury až po návrh a implementaci samotného administrativního systému.

Práce je řešena jako webová aplikace, primárně fungující v intranetové síti. Součástí administrace je správa uživatelů, tvorba časových harmonogramů a hlavní část – správa magazínů. Pro navigaci a specifikaci voleb, určující, s kterou položkou nebo záznamem chceme pracovat, slouží seznam časopisů a menu umístěné v levé části aplikace.

Výsledkem je administrativní systém, který rychle a přesně vykonává požadavky uživatele. K oblíbenosti v užívání by měla sloužit uživatelsky jednoduché ovládání a automatizovanou některých akcí nebo operací.

I. TEORETICKÁ ČÁST

1 POŽADAVKY A POTŘEBY VYDAVATELSTVÍ

Požadavky na administrativní systém byly inovace stávajícího informačního systému. Tato pomalá a již zastaralá aplikace bude nahrazena novější a výkonnější. Zejména kritická oblast spojená s dobou načítání jednotlivých stránek a časovou odezvou na zobrazení dat z databáze.

Tato aplikace by měla být realizována formou webové aplikace, pro snazší přístup zaměstnancům k této aplikaci i jejím datům. Důraz by tedy měl být kladen zejména na rychlost a jednoduchost uživatelského ovládání, jak je patrné z požadavků klienta. Tento systém bude obsahovat ovládání a nastavení vazeb v rámci časopisů, jejich vydání a rubrik. Současně je potřeba vytvořit seznam akcí, které jsou provedeny během času výroby a zhotovení každého jednotlivého vydání a možnost tvorby časového harmonogramu. Další podmínka na funkčnost aplikace je automatizovat procesy, které pracují se stejnými nebo podobnými daty, pro urychlení práce.

Automatické operace budou zahrnovat automatické generování hodnot formulářových polí, u kterých je hodnota jednoduše odvoditelná nebo dopočitatelná. Důležitá funkce tohoto administrativního systému je vytváření vazeb mezi vydáním a rubrikami, akcemi, daty v harmonogramu spojené s tímto vydáním. Při založení nového vydání do systému bude aplikace automaticky generovat hodnoty a vazby pro toto vydání. Data potřebná pro automatické vyplnění budou brána z předchozího vydání (pokud takové existuje), tj. nové vydání bude obsahovat stejné vazby jako vydání předchozí a data spojená s novým vydáním budou přiměřeně posunuta o daný časový úsek. Pokud se bude jednat o první vydání pro nový časopis, budou se automaticky generovat pouze informace odkazující se přímo k tomuto vydání. Vazby a požadované data budou zadána manuálně uživatelsky.

Jelikož se jedná o administrativní systém, bude součástí aplikace možnost založení uživatelů, kteří budou pověřeni prací nad tímto informačním systémem. Součástí uživatelské administrace bude dále možnost editace jednotlivých uživatelů, stejně jako možnost smazání a obnovení uživatele. Důležitou částí, která se zde bude zavádět a měnit je nastavení práv nad jednotlivými časopisy a nastavení oprávnění na záložky v hlavní menu nabídce. Požadavkem je také vytvoření administrátorského účtu, který bude mít

veškerá oprávnění a bude jako jediný mít přístup k uživatelskému menu a tedy povoleno pracovat s uživatelskými účty a přidělovat či odebírat příslušná oprávnění.

Pro zobrazení dat zahrnující časopisy, vydání, rubriky a akce budou použity 3 pohledy, které může uživatel použít.

1. **Stručný pohled** – uživatel uvidí pouze důležité informace
2. **Detailní pohled** – uživatel uvidí všechny informace dostupné pro daný záznam
3. **Historie změn** – možnost nahlédnout do historie veškerých změn prováděných nad danými záznamy v rámci daného časopisu, pokud nebude filtrovacími poli stanoveno jinak.

Dalším požadavkem byla implementace filtračních možností pro práci s konkrétními záznamy nebo specifickou skupinou záznamů na úrovni jednotlivých časopisů, vydání, rubrik, akcí nebo plánování časového harmonogramu.

2 POUŽITÉ TECHNOLOGIE

2.1 .NET Framework

.NET Framework představuje programový model využívaný programátory, kteří mohou psát svůj kód v podstatě jakémkoliv plně objektovém programovacím jazyce. Mezi tyto jazyky se řadí C#.NET, C++.NET, Visual Basic.NET, J#.NET.

Součástí tohoto frameworku je i vylepšený mechanismus pro správu paměti (Garbage Collector - GC), který monitoruje paměť a zdroje použité v aplikaci a podle potřeby je vrací zpět systému. Toto vylepšení pomáhá v programování v případech, kdy je kód špatně napsán a touto chybou by za normálních okolností došlo k vyčerpání systémových prostředků. Jednoduše řečeno, GC, nám sám uvolní paměťové prostředky nad již nepožívanými objekty, pokud to programátor „zapomene“ implementovat sám.

Další vylepšení souvisí se zpracováním chybových stavů, které se ve frameworku realizují pomocí výjimek a jejich následnému zpracování (chyba může být zpracována obecně nebo na konkrétní úrovni, například dělení nulou, indexování mimo povolené hranice pole,...). [1]

.NET Framework poskytuje tyto úkony:

- Slouží jako run-time prostředí, ve kterém běží aplikace (znamená to, že aplikace napsané v tomto prostředí nelze bez tohoto prostředí spustit.)
- Nabízí možnost pro tvorbu webových služeb a ASP.NET stránek
- Poskytuje skutečně velmi širokou řadu knihoven obsahující objektově orientované třídy umožňující snadněji řešit řadu úkolů (jako jsou práce s databází a datovými zdroji, bezpečnost, komunikace).

Aktuálně nejnovější a dostupná verze .NET Frameworku je verze 4.0. [1]

2.1.1 Historie .NET Frameworku

Microsoft začal s vývojem .NET Frameworku na konci 90. let minulého století pod označením NGWS. V listopadu 2000 byla zveřejněna první beta verze a počátkem roku 2002 byla na trh uvedena první stabilní verze .NET 1.0. Framework výhradně spoléhá na XML, jako na datový komunikační standard, ve kterém se data nejen ukládají, ale i slouží k předávání informací a vytváření datové struktury.[2]

2.1.1.1 .NET Framework 1.0

Byl zveřejněn na začátku roku 2002 a jeho kódové označení bylo **Rainier**. Obsahuje první verzi CLR 1.0 (Common Language Runtime) a první verze knihoven založených na třídách BCL (Based Class Library). Byl dostupný pro operační systémy (OS) Windows 98, Me, NT 4.0, 2000, XP. Ukončení běžné podpory tohoto Frameworku bylo v červenci 2007 a pro rozšířenou podporu v červenci 2009.[5]

2.1.1.2 .NET Framework 1.1

Byl zveřejněn roku 2003 a byl součástí Microsoft Visual Studio (VS) 2003 pod kódovým označením **Everett**. Je to první verze Frameworku, která byla součástí OS (konkrétně Windows Server 2003). CLR verze 1.1. Pro tento Framework byly vydány dva Service Packy (SP).

Změny oproti verzi 1.0 jsou:

- Zabudovaná podpora pro mobilní ASP.NET ovladače – původně jako doplněk
- Bezpečnostní změny – dovoluje vykonávat Windows Form sestavení z internetu v „polo-důvěryhodným“ způsobem a povoluje přístup ke kódu podle oprávnění (Code Access Security) v ASP.NET aplikacích.
- Zabudovaná podpora pro databáze Oracle a ODBC – dříve pouze jako doplněk
- .NET Compact Framework – verze Frameworku pro malá zařízení
- Podpora IPV6 (Internet Protokol verze 6)
- Mnoho API změn

Ukončení běžné podpory tohoto Frameworku bylo v říjnu 2008 a pro rozšířenou podporu bude v říjnu 2013.[5]

2.1.1.3 .NET Framework 2.0

Byl zveřejněn počátkem roku 2006 pod kódovým názvem **Whidbey** společně s VS 2005, Microsoft SQL serverem 2005 (kódový název **Yukon**) a BizTalkem 2006. Nová verze CLR 2.0 společně s novými třídami BCL (například generic, generic collections,...).

Změny oproti 1.1 jsou například:

- Plná 64-bitová podpora, jak pro x64, tak pro IA64 hardwarovou platformu.

- Generiky a zabudovaná jazyková podpora pro generiky přímo do .NET CLR
- Částečné (partial) třídy, Anonymní metody, Datové tabulky
- Nové datové ovladače s deklarativní vazbou dat

Tato verze byla ještě vylepšena v SP1 a SP2. [5]

2.1.1.4 .NET Framework 3.0

Označován jako **WinFX** byl zveřejněn v listopadu 2006. Obsahuje nové API, které jsou integrované ve Windows Vista a Windows Server 2008. Také dostupné pro XP SP2 a Windows Server 2003, ale není standardně zabudováno. Používá stále CLR verzi 2.0.

Obsahuje ale 4 hlavní komponenty:

- **WCF (Windows Communication Foundation – kódové označení Indigo)** – servisně orientovaná služba, která sjednocuje mnoho starších technologií (například MSMQ, WSE, ASP.NET WS, .NET Remoting,...)
- **WPF (Windows Presentation Foundation – kódový označení Avalon)** – nový subsystém uživatelského rozhraní a API založené na XML a vektorové grafice, používající 3D grafický hardware a Direct3D technologii.
- **WF (Windows Workflow Foundation)** – pomáhá vytvářet automatické úlohy a integruje transakce pomocí pracovních postupů
- **Windows CardSpace** (kódové označení **InfoCard**) – Softwarová komponenta, která dokáže bezpečně uložit uživatelskou digitální identitu a poskytuje mu jednotné rozhraní pro volbu identity na jednotlivé operace, např. přihlášení na web.

Pro tuto verzi byly vydány aktualizace ve formě SP1 a SP2. [5]

2.1.1.5 .NET Framework 3.5

Byl zveřejněn v listopadu 2007. Microsoft VS 2008 (kódové označení **Orcas**), poprvé v historii umožňuje podporu vývoje aplikací nad více .NET Frameworky (2.0, 3.0, 3.5). Zároveň byly uvedeny i Microsoft SQL Server 2008 (kódové označení **Katmai**) a Microsoft Windows Server 2008 (kódové označení **Longhorn**).

V produktu .NET Framework 3.5 jsou k dispozici celá řada nových funkcí z různých technologických oblastí, přidáné jako nová sestavení.[6] Mezi tyto funkce například patří :

- Nové vlastnosti jazyka pro C# 3.0 a VB.NET 9.0 kompilátor
- Přidána podpora pro strom výrazů a lambda metody
- Language Integrated Query (LINQ) spolu s množstvím poskytovatelů
 - LINQ to XML, LINQ to Objects, LINQ to SQL
- Podpora stránkování pro ADO.NET
- ADO.NET synchronizované API s lokálními a serverovými datovými sklady
- Asynchronní I/O API, ASP.NET AJAX, Podpora HTTP pipeline
- Vylepšení WCF, WPF, WF. WCF podpora pro práci s daty POX a JSON

V srpnu 2008 byla zveřejněna aktualizace SP1, která obsahovala zejména změny pro WPF. Byly zde přidány 2 komponenty datových služeb – ADO.NET Entity Framework a ADO.NET Datové služby (Data Services). Pro MVC přidány dvě sestavení Abstraction a Routing ze jmenného prostoru System.Web.[5]

2.1.1.6 .NET Framework 4.0

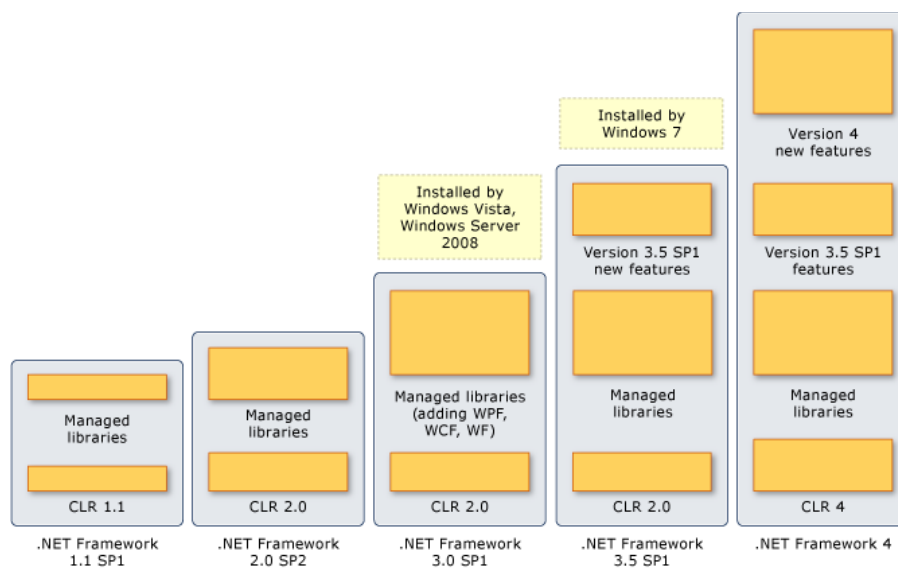
Byl zveřejněn v dubnu 2010 a funguje společně se staršími verzemi rozhraní Framework. Aplikace napsané ve starší verzi je možno spouštět ve výchozí verzi.

Tento Framework obsahuje následující vylepšení:

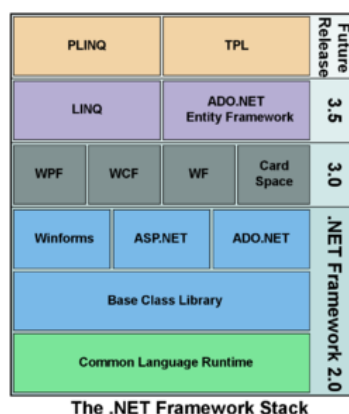
- Vylepšení modulu CLR verze 4.0 a knihovny BCL
 - Vyšší výkon a podpora vícejádrových procesorů, uvolnění paměti na pozadí
 - Nové číselné typy
 - Snazší ladění chyb, ladění ve smíšeném režimu pro 64-bitový kód
- Inovace jazyků VB a C#
- Lepší přístup k datům a modelování (Celá řada nových funkcí pro Entity Framework, Rozšíření funkcí pro datovou službu WCF)
- Vylepšené rozhraní pro ASP.NET
 - Lepší kontrola jazyka HTML, stylů CSS
 - Nová funkce pro dynamická data (filtr dotazů, šablony objektů, širší podpora Entity Framework 4.0)

- Podpora pro vylepšenou knihovnu AJAX pro webové formuláře
- Vylepšení technologie WPF a WCF, vylepšení služby WF
- Nové funkce paralelního programování – knihovna TPL (Task Parallel Library), paralelní dotaz LINQ (PLINQ), kontrola paralelní smyčky

Spojením s .NET Frameworkem 4.0, Microsoft nabízí sadu vylepšení (kódové označení **Dublin**) pro možnosti aplikačního serveru Windows Server 2008. Rozšíří tak IIS na ‚standardního hosta‘ pro aplikace, které využívají WCF nebo WF. [7]

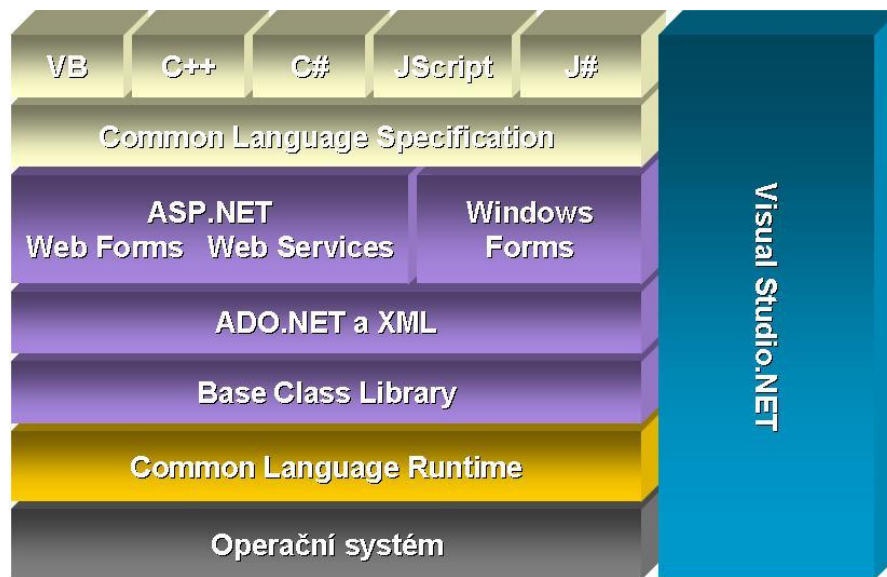


Obr. 1 - Souhrn historických verzí .NET Frameworku a které verze jsou již instalovány operačním systémem Windows [4]



Obr. 2 - .NET Framework zásobník

2.1.2 Architektura .NET Frameworku

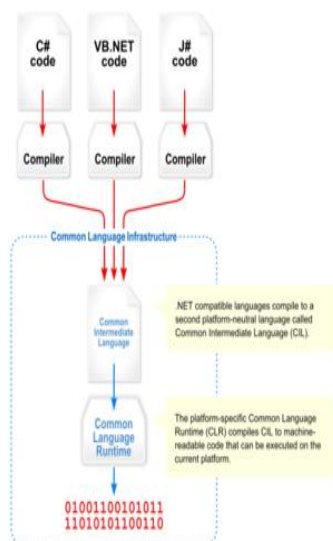


Obr. 3 - Architektura .NET Frameworku

Common Language Infrastructure (CLI)

Hlavní aspekty .NET Frameworku souvisí s Infrastrukturou společných jazyků (CLI). Smyslem CLI je poskytnout jazykově nezávislou platformu pro vývoj a spouštění aplikací. Zahrnuje funkce jako: Odchytávání výjimek, GC, zabezpečení, interoperabilitu (vzájemná spolupráce různých jazyků). O tento jazykově nezávislý kód se stará Společný běhový systém CLR. [5]

Společný běhový systém (CLR)

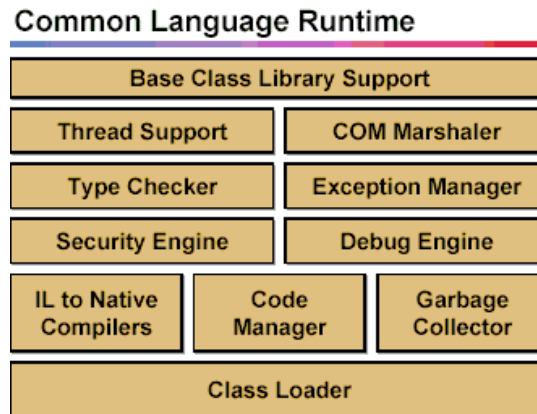


Obr. 4 - *Infrastruktura společných jazyků*

.NET Framework poskytuje run-time prostředí nazývané společný běhový systém (CLR), který spustí kód a poskytne služby ulehčující vývojový proces. CLR představuje virtuální stroj, na kterém pracují běhové funkce, této platformy a jsou zde dostupné všechny knihovny tříd systému. Kód, který zde běží, se nazývá Microsoft Intermediate Language (MSIL) a jedná se o mezikód mezi programovacím jazykem a strojovým kódem. Výhody tohoto mezikódu jsou řízený kód, správa paměti, jednotný typový systém. Mezi největší výhody CLR patří schopnost vývoje v různých

programovacích jazycích – tzn. možnost spolupráce mezi jednotlivými jazyky (tzv. cross-language interoperability).

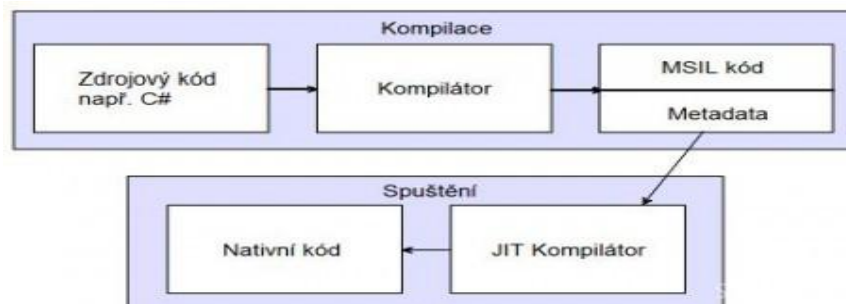
Na obrázku (Obr. 5), je vidět co všechno obsahuje CLR:



Obr. 5 - Souběžné běhové prostředí CLR

Microsoft Intermediate Language (MSIL)

Jedná se o jazyk podobný assembleru a byl vytvořen pro platformu .NET. Kód programovací jazyka se při kompilaci přeloží právě do jazyka MSIL a CLR nám zajistí, že pro spuštění aplikace se program v MSIL přeloží pomocí JIT (Just In Time) kompilátoru do strojového kódu.



Obr. 6 - Proces kompilace zdrojového do nativního kódu

Just In Time (JIT)

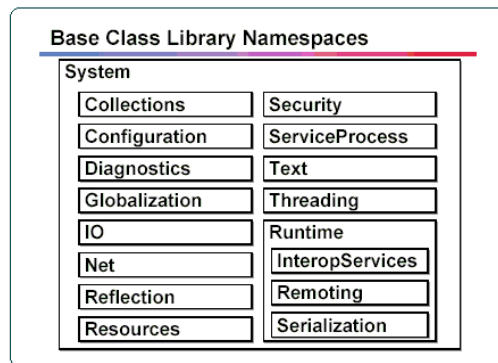
Dynamicky překládá MSIL kód do kódu nativního. Rozeznáváme dva typy kompilací:

Interpretovaná kompilace – překládá vysokoúrovňový jazyk do strojového kódu průběžně při každém spuštění

Statická JIT kompilace – tato kompilace se provede pouze jednou a to při prvním spuštění

Základní knihovna tříd (BCL)

Poskytuje základní stavební bloky pro jakýkoliv vývoj ASP.NET aplikací, Windows Forms aplikací nebo webových služeb. BCL vám v podstatě slouží jako hlavní bod pro interakci se běhovými třídami, zahrnující:



Obr. 7 - Třídy z BCL jmenného prostoru

Sestavení (Assembly)

Jedná se o logickou kolekci složenou z jednoho nebo i více *.dll*, *.exe* nebo *.module* souborů a zdrojů doplněných o *manifest*. Jsou uloženy v Portable Executable (PE) formátu, běžném pro právě *.dll*, *.exe* soubory na Windows platformě. Často jsou definována jako programová jednotka, která je určena k nasazení a opakovanému používání, bezpečnosti a řízení verzí. Dynamicky linkované knihovny jsou podobné těmto sestavením, ale nesplňují vysoké požadavky jako sestavení, které je splňují tím, že obsahují manifest.

Manifest je blok metadat, obsahující:

Identitu (jméno, verze a kultura), Seznam souborů + kryptografické zabezpečení, odkaz na další požité sestavení + jejich verze, exportované (veřejně viditelné) typy a zdroje, bezpečnostní požadavky.

2.2 ASP .NET

Každá webová aplikace skládá z webových stránek a je možné ji napsat v libovolném (pro takové účely vhodném) jazyce. Tyto stránky můžeme zpracovávat bohatě vybaveným HTML, používat JavaScript, vytvářet komponenty zapouzdřující programovací logiku nebo vyladřovat či přizpůsobovat své aplikace za pomoci různých konfiguračních voleb. Jednou takovou skriptovací technologií je ASP.NET, který, oproti klasickým ASP (Active Server Pages) nebo PHP (Personal Home Page), pracuje v pozadí odlišně v těchto věcech:

- Nabízí úplný, objektově orientovaný programovací model, který obsahuje architekturu řízenou událostmi, založenou na ovládacích prvcích. Tento model podporuje zapouzdřování kódu a jeho opětovné využívání
- Můžete v něm psát kód, který patří mezi podporované jazyky .NET (zejména C#, VB, J#, další jazyky mající kompilátory od jiných výrobců)
- Důležitost kladena na vysoký výkon. Stránky ASP.NET a její komponenty jsou kompilovány na žádost, což zamezuje interpretaci pokaždé, když se použijí. Pro ještě větší výkon ASP.NET obsahuje flexibilní ukládání dat do cache a vyladěný modul pro přístup k datům.
- Rozšířená správa stavu (State Management)
- Praktické vázání dat, Dynamická tvorba grafiky, Robustní model bezpečnosti

Důležité je upozornit, že stránky ASP.NET se neinterpretují, ale kompilují do tzv. assembly. [11]

2.2.1 Vznik a vývoj ASP.NET

Velmi rychle od vydání ASP.NET verze 1.0, firmou Microsoft, se tato technologie stala standardem pro vývoj webových aplikací a zdatným konkurentem všech ostatních platforem webového vývoje. Od vydání verze 1.0 však uplynulo mnoho času a Microsoft již postupně představil novější verze 1.1, 2.0, 3.5 a 4.0. Společně se zveřejněním první verze .NET Framework, byla uvedena i první verze ASP.NET 1.0, která se považovala za vylepšení a nástupce předcházejícího ASP. [11]

2.2.1.1 ASP.NET 2.0

Kvalita prvních verzí 1.0 a 1.1 je sama o sobě znát hned z pohledu na počet změn spojených s opravami, které se realizovali pro starší verze. Samotné vylepšení verze 2.0 se týká zejména:

- **Ovládací prvky** – je jich více a jsou lépe vybaveny. Zavedeno více jak 40 nových prvků. Mezi hlavní patří *TreeView* s možností sbalování a rozbalování a prvek *Menu* poháněný JavaScriptem.
- **Vzory stránek** – jsou to opětovně využitelné šablony pro stránky. Zajišťuje možnost stejného záhlaví, zápatí, navigačních prvků, ... na všech stránkách webové aplikace nebo webu.

- **Motivy** – pro webové ovládací prvky nabízí možnost definice standardizované sady charakteristik vzhledu. Ty je možné použít na celý web pro sjednocení vzhledu.
- **Bezpečnost a členství** – přidána spousta bezpečnostních funkcí. Jedná se například o automatickou podporu sledování přihlašovacích dokladů uživatele (credentials), autorizace založená na rolích nebo ovládací prvky určené pro správu bezpečnosti.
- **Ovládací prvky pro zdroje dat** – deklarativní určení modelu ovládacích prvků pro zdroje dat - jak bude stránka komunikovat s datovým zdrojem.
- **Webové části (Web parts)** – poskytují předem zabudovaný, kompletní portálový rámec s rozvržením, založeným na toku, konfigurovatelnými pohledy a dokonce s podporou přetahování myši.
- **Profily** – umožňuje ukládat specifické informace o daném uživateli do databáze, aniž bychom museli psát nějaký databázový kód. O vše ostatní se postará ASP.NET.

Mnohé z funkcionalit zavedených v ASP.NET 2.0 pracujících prostřednictvím abstrakce, se říká model poskytovatele. Elegance tohoto modelu spočívá v tom, že můžeme požívat jednoduché poskytovatele během budování kódu stránek. Při změně požadavků, si jednoduše vytvoříme vlastního poskytovatele, místo abychom měnili stránky. [11]

2.2.1.2 ASP.NET 3.5

Další změna pro ASP.NET přišla až s .NET Frameworkem 3.5 a přinesla s sebou několik nových vylepšení, mezi které bezpochyby patří LINQ a Ajax.

- **LINQ** – sada rozšíření pro jazyky C# a Visual Basic. Obsahuje asi 40 dotazovacích operátorů. Nejzákladnější odrůdou LINQ je LINQ to Objects. Umožňuje vykonat dotaz nad nějakou kolekcí objektů.

LINQ to DataSet poskytuje obdobné chování pro dotazy nad objekty typu sady dat (DataSet) jenž je v paměti. LINQ to XML pro práci s XML daty. A samozřejmě LINQ to SQL který v podstatě vytváří parametrizovaný dotaz SQL, přičemž se dotaz vykonává až tehdy, když se pokusíme přistoupit k jeho výsledkům.[11]

- **Ajax (Asynchronous javascript and XML)** – vznikl z důvodu potřeby získat data ze serveru, aniž bychom museli odesílat obsah celé stránky na server a opět přijmou celý obsah stránky a znovu jej vykreslit. Jelikož ASP.NET dělá svoji práci na serveru, tak při každé akci na webu potřebuje prohlížeč odeslat nějaká data na server, získat novou kopii stránky a aktualizovat zobrazení. Pomocí AJAXu

můžeme aktualizovat jen požadované ovládací prvky webu, tím že pošleme na server asynchronní požadavek, ten se na serveru zpracuje a odešlou se data zpět, místo aby se posílal obsah celé stránky. [11] Oblíbenosti používání tohoto programování je i to že objekt *XMLHttpRequest* (nezbytný pro podporu asynchronních požadavků) je přítomen ve všech, dnes nejpoužívanějších, prohlížečích:

- *Internet Explorer 5 a vyšší*
- *Opera 7.6 a vyšší*
- *Firefox od verze 1.0*
- *Safari od verze 1.2*
- *Chrome*

Z tohoto pohledu tak mezi důležité sestavení (assembly) ASP.NET 3.5 patří například:

- System.Core.dll (jádro funkcionality LINQ)
- System.Data.Linq.dll (Implementace LINQ to SQL)
- System.Data.DataSetExtensions.dll (Implementace LINQ to DataSet)
- System.XML.Linq.dll (Implementace LINQ to XML)
- System.Web.Extensions.dll (Implementace ASP.NET AJAX)

2.2.1.3 ASP.NET 4.0

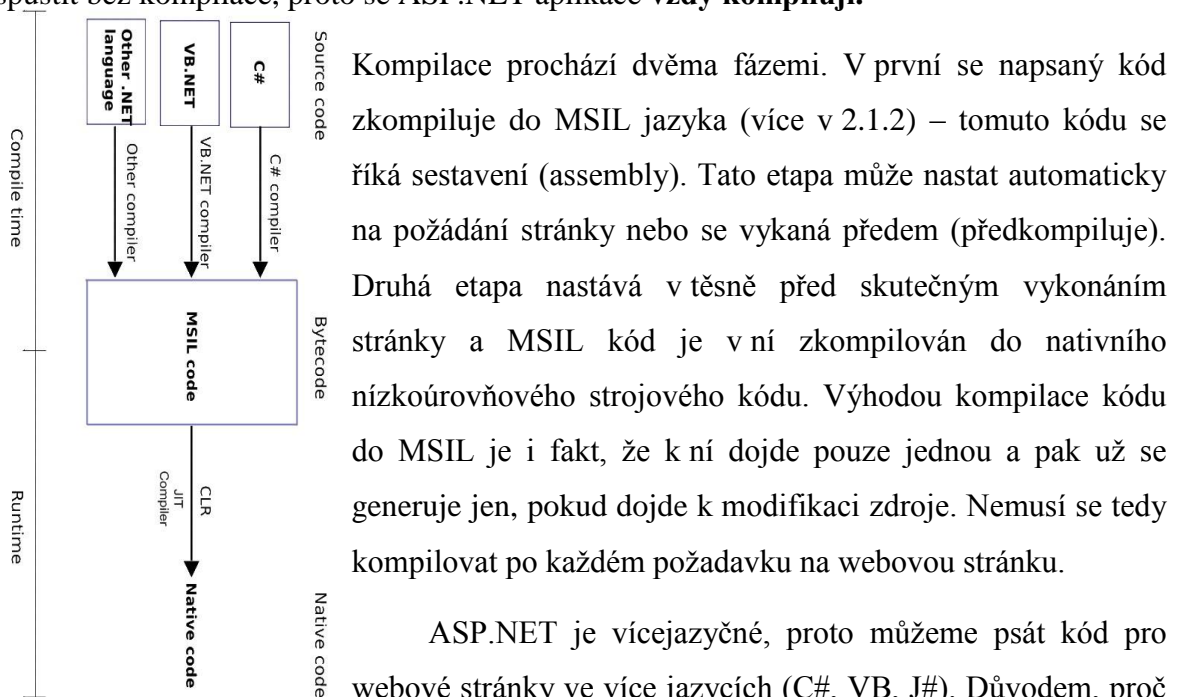
Zatím poslední aktualizace technologie ASP.NET (i když v době psaní této práce je již dostupná ASP.NET 4.5 RC). Tato verze dostala mnoha vylepšení v několika oblastech.[21]

- **Služby jádra** – sem patří například (přidání informace o .NET do *web.config* pro určení Frameworku se kterým se bude aplikace kompilovat, vylepšení výstupního úložiště – Output caching, auto start aplikace, komprese Session stavu, rozšířená validace Requestů, a další)
- **Webové formuláře** – obsahuje nastavení meta tagů pomocí Page.MetaKeywords a Page.MetaDescription vlastností, povolení ViewState pro jednotlivé ovládací prvky, podpora pro nové prohlížeče a zařízení (blacburry, chrome, iphone,...), směrování, nastavení uživatelských ID, rozšíření podpory pro práci s CSS styly a další
- **MVC**

- **Dynamické data** – povolení dynamických dat pro datově vázané prvky v existující webové aplikaci, deklarativní syntaxe prvků – `dynamicDataManager`, podpora dědění v datovém modelu, a další
- **Podpora grafů**
- **Visual Web Developer rozšíření** – zlepšená CSS kompatibilita, HTML a JavaScript přidáno do IntelliSense
- **Vylepšené Multi-Targeting** – jednodušší práce s projekty z dřívějších verzí .NET [21]

2.2.2 ASP.NET Podrobněji

První co je třeba si uvědomit je, že ASP.NET je integrovaný s .NET Frameworkem (více o .NET Framework v kapitole 2.1). Proto je možné používat stejné třídy v ASP.NET jako v jiném druhu aplikací .NET (samostatné aplikace Windows, služby Windows, příkazový řádek, atd.). Další fakt, který musíme mít na mysli, je že ASP.NET se neinterpretuje, ale kompiluje. To je jedna z výhod oproti pomalému výkonu ve skriptech ASP. Už i použití jazyků C# nebo Visual Basic napovídá, že tento kód nebude možné spustit bez kompilace, proto se ASP.NET aplikace **vždy kompilují**.



Obr. 8 - Kompilace
webové stránky
ASP.NET

Kompilace prochází dvěma fázemi. V první se napsaný kód zkompiluje do MSIL jazyka (více v 2.1.2) – tomuto kódu se říká sestavení (assembly). Tato etapa může nastat automaticky na požádání stránky nebo se vykoná předem (předkompiluje). Druhá etapa nastává v těsně před skutečným vykonáním stránky a MSIL kód je v ní zkompilován do nativního nízkoúrovňového strojového kódu. Výhodou kompilace kódu do MSIL je i fakt, že k ní dojde pouze jednou a pak už se generuje jen, pokud dojde k modifikaci zdroje. Nemusí se tedy kompilovat po každém požadavku na webovou stránku.

ASP.NET je vícejazyčné, proto můžeme psát kód pro webové stránky ve více jazycích (C#, VB, J#). Důvodem, proč je tohle možné je již několikrát zmiňovaný MSIL. [11]

Příklad kódu v C#:

```
using System;
namespace HelloWorld
{
    public class TestClass
    {
        private static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```

Příklad kódu ve Visual Basicu:

```
Imports System
Namespace HelloWorld
    Public Class TestClass
        Private Shared sub Main(args() as String)
            Console.WriteLine("Hello World")
        End Sub
    End Class
End Namespace
```

Příklad kódu ve MSIL:

```
.method private hidebysig static void Main(string[] args) cil
managed
{
    .entrypoint
    //Code size 13 (0xd)
    maxstack 8
    IL_0000: nop
    IL_0001: ldstr "Hello World"
    IL_0006: call void [mscorlib]System.Console::WriteLine(string)
    IL_000b: nop
    IL_000c: ret
} //end of method TestClass::Main
```

Pokud bychom chtěli kód MSIL recompileovat a zjistit původní kód, použijeme Disassembler MSIL (soubor `iladsm.exe`) [11]

ASP.NET samozřejmě běží uvnitř společného runtime jazyků CLR. To nám poskytuje hned několik výhod:

- Automatická správa paměti a svod odpadků (garbage collection)
- Typová bezpečnost – soběstačnost sestavení zkompilevaného kódu
- Rozšiřitelná metadata, Strukturované zpracování chyb, Multithreading

ASP.NET je objektově orientované. Můžeme tedy například vytvářet opětovně použitelné třídy, rozšiřovat existující třídy děděním, standardizovat kód rozhraní,... Krásným příkladem objektově orientovaného myšlení ASP.NET jsou serverové ovládací prvky, které jsou ideálním vzorem pro zapouzdření. Tyto ovládací prvky značíme v kódu prefixem `asp:` za kterým následuje název třídy. Ukázka textového pole:

```
<asp:TextBox ID="mojeID" Text="Ahoj" runat="server" />
```

Samozřejmě je možné změnit statické části HTML na plnohodnotný a funkční serverový ovládací prvek pouhým přidáním atributu `runat="server"` :

```
<input type="text" id="mojeID" runat="server" />
```

Dalším faktem je, že ASP.NET podporuje různá zařízení a různé prohlížeče. Značná různorodost webových prohlížečů, patří k jedné z největších výzev pro programátory pro jejich různou podporu standardu HTML. Výhoda v užívání bohatě vybavenou skupinu webových ovládacích prvků ASP.NET je realizace kódu HTML adaptivně, tzn. bere v úvahu schopnosti klienta. Samotná konfigurace a nasazení aplikace na ostrý server, je jednodušší právě díky jednotnému .NET Frameworku, které poskytuje stejné jádro tříd. V reálné praxi to znamená, že stačí pouze nakopírovat soubory do nějakého virtuálního adresáře na ostrém serveru. Konfigurace je usnadněna díky souboru `web.config`, který existuje právě pro tento účel a je umístěn ve stejném adresáři jako webové stránky. [11]

2.3 SQL (Structure Query Language)

I přesto, že SQL je ANSI (American National Standards Institute) standard, existuje mnoho verzí jazyka SQL. Avšak všechny podporují alespoň základní příkazy (`select`, `update`, `insert`, `delete`, `where`). Abychom mohli tento jazyk používat, musíme mít nějakou databázi, nad kterou budeme tyto dotazy pokládat (MS SQL Server, Oracle, Microsoft Access, MySQL,...). V případě této Bakalářské práce se jedná o **MS SQL Server 2008 R2 Express** edici. [13]

2.3.1 Co je a jak funguje SQL?

V červnu roku 1970 publikoval Dr. E. F. Codd článek „A Relational Model of Data for Large Shared Data Banks“ v časopisu Association of Computer Machinery (ACM). Jeho model je nyní akceptován, jako definitivní model pro relační databázový řídicí systém

(RDMBS). S použitím Coddova modelu, vyvinula firma IBM jazyk SEQUEL (Structured English Query Language), který později přešel na SQL. V roce 1979 firma Relational Software inc. (nyní Oracle) představil první komerčně dostupnou implementaci jazyka SQL. Dnes je jazyk SQL přijatý jako standard RDMBS.

Technicky řečeno je SQL datový podjazyk. Smyslem SQL je poskytnout rozhraní mezi relační databází (např. Oracle, MS SQL) a všemi SQL instrukcemi v databázi. Mezi vlastnosti jazyka SQL patří:

- Zpracovává sadu dat jako skupinu místo individuální jednotky
- Poskytuje automatickou navigaci k datům
- Používá výrazy, které jsou komplexní a samostatně mocné a proto samostatně.

SQL nám dovoluje pracovat s daty na logické úrovni. Nemusíme se zabývat implementací detailů, pokud ovšem nechceme manipulovat s daty. Například, pokud chceme získat skupinu řádků z tabulky, stačí definovat podmínku pro filtraci řádků. Všechny vyhovující řádky, budou vráceny v jednom kroku a mohou být dále použity jako výsledek pro uživatele, pro další SQL dotaz nebo jako data pro aplikaci. SQL poskytuje výrazy pro množství dotazů:

- Dotazování na data
- Vkládání, aktualizace nebo mazání řádků v tabulce
- Vytváření, nahrazování, změnu a zahození objektů
- Kontrolovaný přístup do databáze a k jejím objektům
- Garanci konzistence a integrity databáze

Všechny hlavní relační databáze řízených systémů podporují jazyk SQL, takže je možné využít znalosti SQL napříč těmito databázemi. Navíc všechny programy napsané v SQL jsou přenosné.[14]

Příklad SQL dotazu v MS SQL 2008:

```
SELECT jmeno, prijmeni, vek FROM Uzivatele WHERE vek >= 18  
ORDER BY vek DESC, prijmeni
```

Tento dotaz nám z tabulky Uzivatele vrátí záznamy všech uživatelů, kteří mají 18 nebo více let a tyto záznamy budou seřazeny podle věku (od nejstaršího) a podle příjmení (podle abecedy).

2.3.2 Proč Microsoft SQL Server 2008 Express?

Hlavní výhodou je bezplatná licence a poměrně slušná funkcionalita, která je dostatečná pro malé i středně velké firmy nebo projekty. Navíc Express edice je plně kompatibilní s ostatními (placenými) verzemi MS SQL Serveru. To znamená relativně snadný upgrade na placenou verzi (díky shodnému jádru). Snadno se instaluje a je podporován na většině operačních systémů MS Windows. Dále tato Express edice nabízí síťový přístup bez nutnosti souborového sdílení, snadné zálohování, omezení přístupovými právy, transakční zpracování, na WinXP/Vista/Win7 běží jako služba operačního systému, i když k počítači není žádný uživatel přihlášen a další. [15]

2.4 Programovací jazyk C#

C# byl navržen jako konkurent Javy. Je to první jazyk navržený přesně podle systému CLS (Common Language Specification – tzn. kód kompatibilní s CLR) a tím umožňuje asi nejlépe využít veškeré jeho možnosti. Nejedná se tedy o jazyk interpretovaný, jako třeba Java, ze které se C# hodně inspiroval, ale přímo o spustitelný kód (přesněji generovaný automaticky až těsně před spuštěním programu).[16]

C# je objektově orientovaný programovací jazyk, vyvinutý firmou Microsoft jako součást .NET platformy. Proto je C# zdrojový kód, stejně jako všech .NET jazyků, kompilován do MSIL kódu. C# je primárně odvozen z jazyků C++ a C společně s mixem některých vlastností Javy a VB.

Cíle jazyka C#:

- Jednoduchý, moderní, univerzální, objektově orientovaný jazyk
- Poskytnout silně typovou kontrolu, kontrolu hranic polí, detekci neinicializovaných proměnných a automatický GC.
- Přenositelnost kódu a znalostí programátorů (zvláště těch, se znalostmi C++ a C)
- Internacionalizace
- Vhodný pro psaní aplikací pro hostující i vestavěné systémy, od velmi velkých využívající sofistikované operační systémy a až po velmi malé mající dedikované funkce.

I přes požadavky na spotřebu paměti a výkon, nebylo záměrem dosáhnout přímo výkonu jaký má jazyk C nebo jazyky sestavení. [17]

2.4.1 Historický vývoj C#

Vlastní historie C# je neoddělitelně spjata s platformou Microsoft .NET. Vše začalo v roce 1993, kdy byly uvedeny modely COM (Component Object Model - softwarová architektura umožňující vystavět aplikace z binárních komponent) a DCOM (Distributed Component Object Model - protokol umožňující komunikovat softwarovým komponentám přes síť (včetně internetu pomocí HTTP protokolu)). Roku 1997 se tyto dva modely spojily do COM+, což bylo rozšíření původního COM rozhraní. To umožňovalo softwarovým komponentám, psaných v různých jazycích, spolu komunikovat. Tato myšlenka byla použita roku 1999, kdy se Microsoft zmiňuje o technologii .NET.[16]

C# verze 2.0 přišla s několika novinkami a vylepšeními mezi které patří: generiky, iterátory, anonymní metody, částečné typy (partial), statické třídy, typ NULL, omezený přístup k vlastnostem

C# verze 3.0 představila několik jazykových rozšíření. Mezi tyto novinky patří: Anonymní typy, Inicializace objektů, Implicitní typy lokálních proměnných, Implicitně typované pole, Rozšířené metody, Lambda výrazy, Stromy výrazů, Dotazovací výrazy. Verze 3.0 je kompatibilní s verzí 2.0. [17]

Hlavní změny v C# verzi 4.0 se týkaly dynamického programování a společnou evoluci s VB. Zachování hlavních znaků obou jazyků a přesto nové vlastnosti jsou uveřejněny ve stejnou dobu. Nové vlastnosti C# 4.0: Dynamické vázání, pojmenované a volitelné argumenty, COM specifikace interop vlastností, Variace (přiřazování hodnot mezi datovými typy a objekty a jejich kolekcemi jsou nyní bezpečné).

2.4.2 Výhody a nevýhody

Zde se hlavně zaměříme na porovnání jazyku C# s „konkurenční“ Javou:

- Oba jazyky mají víceméně stejnou výkonnost
- Pro C# je jednodušší napsat webovou aplikaci (ASP.NET), zvláště pokud se jedná o malou aplikaci, která musí být rychle napsaná.
- Otevřenost kódů – zde je na tom lépe JAVA, která má širší open-source komunitu
- Porovnání nástrojů – zde záleží, které nástroje potřebujete a preferujete. Protože jak C# (VS), tak JAVA (Eclipse, NetBeens, a mnoho dalších open-source) jsou velmi dobře vybaveny pro vývoj.
- Komunita a podpora – opět oba jazyky mají rozsáhlou komunitu a silnou podporu

- Hardware/Software kompatibilita – Hostování ASP.NET aplikací je finančně náročnější. Microsoft aplikace také lépe běží pod Microsoft prostředím, zatímco je JAVA v tomto aspektu více flexibilnější a má i více možností na hosting. [18]

2.5 JavaScript a jQuery

JavaScript je implementací ECMAScriptu, který je oficiální jazykový standard (ECMA-262). Byl vynalezen v roce 1996 Brendanem Eichem v Netscape (spolu s Navigátorem 2.0) a od té doby se objevuje ve všech prohlížečích. Roku 1997 přijato jako standard firmou ECMA (ECMA-262) a v roce 1998 schváleno organizací ISO jako standard (ISO/IEC 16262).

JavaScript byl navržen, aby přidal interaktivitu do HTML stránek. Jedná se o skriptovací jazyk, což je odlehčený programovací jazyk. Je převážně vestavěn přímo do HTML stránek. Kód JavaScriptu se interpretuje. Je bezplatný a volně dostupný. Veškeré akce se provádí na straně klienta (v prohlížeči), tzn.: kód běží na straně klienta.

S JavaScriptem můžeme vylepšit naše stránky. Díky němu dokážeme reagovat na události, které nastanou na webové stránce (například stisk tlačítka), můžeme manipulovat s elementy na stránce, validovat data, získat informace o klientově prohlížeči nebo vytvářet cookies.

Nadstavbu pro JavaScript tvoří knihovna jQuery. První verzi vytvořil a uvedl, roku 2006 v BarCampu v New Yorku, John Resig. Tato knihovna zjednodušuje a zefektivňuje kód JavaScriptu. Méně kódu, lepší funkcionalita. Velmi užitečnou se tato knihovna stává z důvodu stejného chování jQuery kódu, ve všech prohlížečích. Navíc v dnešní době existuje velké množství pluginů, které nám dovolují ještě více vylepšit chování našich stránek nebo webových aplikací.[20]

2.5.1 Proč JavaScript a jQuery

Protože se kód JavaScriptu a jQuery vykonává na straně klienta, nezatěžuje server s dotazy na zpracování požadavků nebo novou stránku. Jelikož je v knihovně jQuery implementovaná funkcionalita AJAXu, je práce s aktualizacemi dat nebo stránek snadná a rychlá – aktualizují se pouze prvky, které chceme. Široká podpora funkcí pro práci s elementy na stránce, dělá s jQuery velmi užitečnou knihovnu, která se těší stále větší a větší oblibě. S její pomocí je snadné přistupovat k různým elementům nebo ovládacím prvkům na stránce a manipulovat s nimi. Velmi jednoduché je navázání různých událostí

na jakýkoliv prvek na stránce a tím obohatit a vylepšit webovou aplikaci. Pro jQuery jsou již napsané a dostupné mnohé pluginy, které nám ušetří programovací čas a dodají potřebnou funkcionalitu, kterou momentálně potřebujeme (např. pagin pro validaci formulářových dat, plugin pro spojení s facebookem, twitterem nebo nápověda pro formulářová pole, a mnoho dalších). Tato knihovna je navíc stále ve vývoji a tak se můžeme těšit na další nové funkce a vylepšení, které nám usnadní práci alepší přehlednost našich kódů. Navíc s použitím jQuery můžeme zapomenout na nepříjemné problémy s různými prohlížeči a jejich různorodou interpretací JavaScriptových příkazů, protože jeho funkce fungují stejně ve všech podporovaných prohlížečích (s výjimkou známých problémů, které se řeší a opravují, aby v další verzi byly odstraněny).

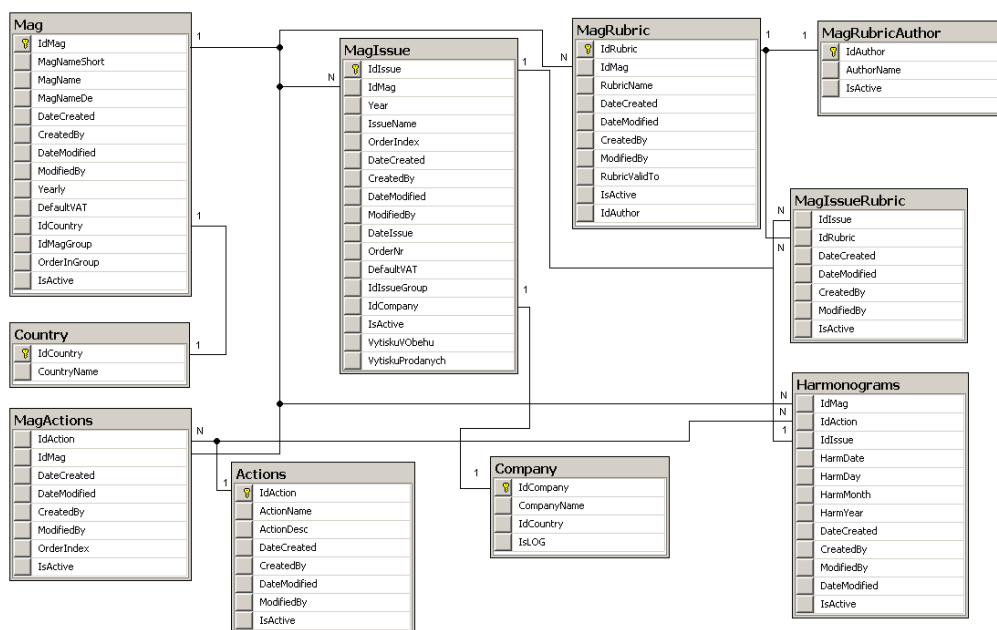
2.5.2 Výhody a nevýhody

Výhodou je, že nemusíme nic instalovat ani nic platit. Vše je zadarmo a, v případě JavaScriptu, již zabudované do všech hlavních prohlížečů. Navíc pracují na straně klienta a proto je reakce na uživatelské volby rychlá, plus snižují vytíženost serverů, nad kterými běží webové stránky nebo aplikace. Nevýhodou je, že v prohlížeči si může klient vypnout podporu JavaScriptu a v tomto případě nebude náš JavaScriptový nebo jQuery kód fungovat.

II. PRAKTICKÁ ČÁST

3 STRUKTURA DATABÁZOVÉHO SYSTÉMU

Samozřejmostí pro realizaci webové aplikace, která má reagovat na akce uživatele a zobrazovat požadovaná data, je přítomnost databázového serveru. V tomto případě tuto úlohu sehrává MS SQL Server 2008 Express. Abychom mohli potřebné údaje uchovávat v databázi, měli data k dispozici pro nahlédnutí nebo editaci a rychle k nim přistupovali, musíme si nejdříve navrhnout strukturu databázových tabulek a určit vazby mezi jednotlivými tabulkami. Základní strukturu pro časopisy a jejich další atributy popisuje Obr. 9.



Obr. 9 - Základní schéma databázové struktury Administrativního systému

Co by se ale stalo, kdyby došlo k nechtěné změně nebo mylnému smazání záznamu z nějaké tabulky nebo se prostě projít historií operací nad konkrétním záznamem? Pro tento případ jsou na důležité tabulky napojeny jejich rozšířené kopie, sloužící čistě pro logovací účely, jak je vidět na Obr. 10 a Obr. 11.

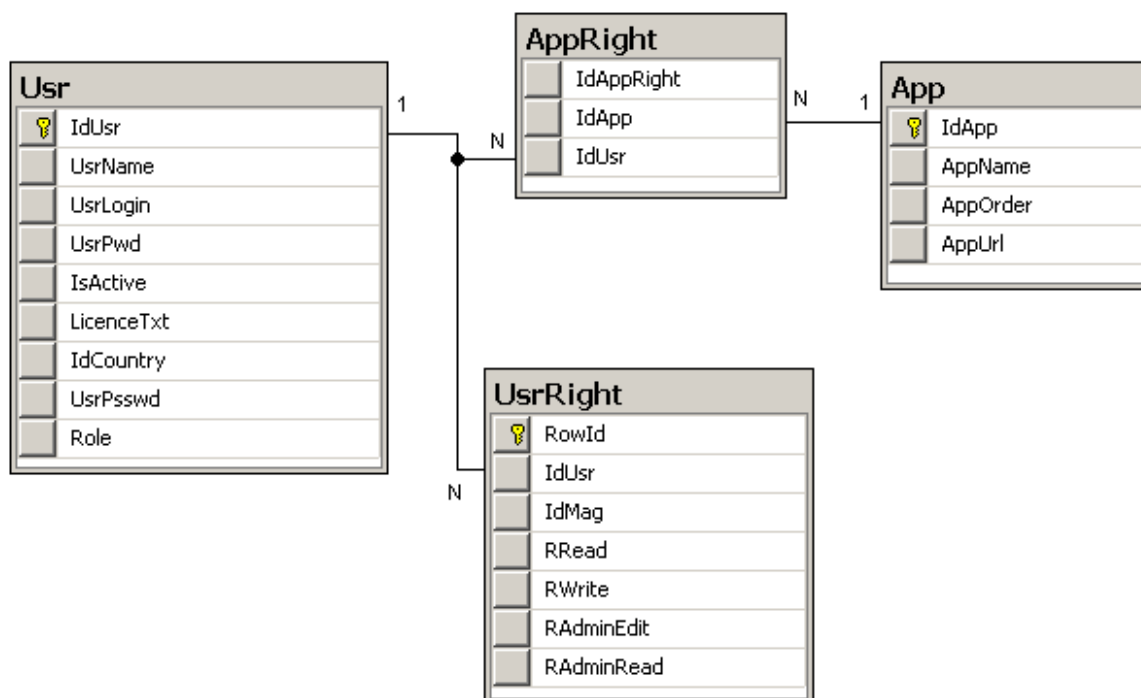
Mag	Mag_LOG	MagIssue	MagIssue_LOG	MagRubric	MagRubric_LOG
IdMag	RowId	IdIssue	RowId	IdRubric	RowId
MagNameShort	LOGFrom	IdMag	LOGFrom	IdMag	LOGFrom
MagName	LOGTo	Year	LOGTo	RubricName	LOGTo
MagNameDe	IdMag	IssueName	IdIssue	DateCreated	IdRubric
DateCreated	MagNameShort	OrderIndex	IdMag	DateModified	IdMag
CreatedBy	MagName	DateCreated	Year	CreatedBy	RubricName
DateModified	MagNameDe	CreatedBy	IssueName	ModifiedBy	DateCreated
ModifiedBy	DateCreated	DateModified	OrderIndex	RubricValidTo	DateModified
Yearly	CreatedBy	ModifiedBy	DateCreated	IsActive	CreatedBy
DefaultVAT	DateModified	DateIssue	CreatedBy	IdAuthor	ModifiedBy
IdCountry	ModifiedBy	OrderNr	DateModified		RubricValidTo
IdMagGroup	Yearly	DefaultVAT	ModifiedBy		IsActive
OrderInGroup	DefaultVAT	IdIssueGroup	DateIssue		DateDeleted
IsActive	IdCountry	IsActive	OrderNr		IdAuthor
	IdMagGroup	VytiskuVObehu	DefaultVAT		
	OrderInGroup	VytiskuProdanych	IdIssueGroup		
	IsActive		IdCompany		
	DateDeleted		IsActive		
			DateDeleted		
			VytiskuVObehu		
			VytiskuProdanych		

Obr. 10 - Tabulky Mag, MagIssue a MagRubric a jejich logovací tabulky

MagIssueRubric	MagIssueRubric_LOG	MagActions	MagActions_LOG
IdIssue	RowId	IdAction	RowId
IdRubric	LOGFrom	IdMag	LOGFrom
DateCreated	LOGTo	DateCreated	LOGTo
DateModified	IdIssue	DateModified	IdMag
CreatedBy	IdRubric	CreatedBy	IdAction
ModifiedBy	DateCreated	ModifiedBy	DateCreated
IsActive	DateModified	OrderIndex	DateModified
	CreatedBy	IsActive	CreatedBy
	ModifiedBy		ModifiedBy
	IsActive		OrderIndex
	DateDeleted		IsActive
			DateDeleted

Obr. 11 - Tabulky MagIssueRubric a MagAction a jejich logovací tabulky

Další součástí databázové struktury jsou tabulky určené pro uchování dat o uživateli, používající tuto aplikaci a tabulky obsahující informace o uživatelských právech k položkám menu, možnostem čtení nebo zápisu dat vztahujících se k danému časopisu. Práva a jejich nastavení má na starosti určený uživatel (nebo uživatelé), který má přístup k uživatelským účtům a může je tedy měnit. Jedná se tedy o administrátora nebo osobu s administrátorskými právy. Na Obr. 12 můžete vidět tabulky sloužící k uchování dat o uživateli a jejich právech pro administrativní systém.



Obr. 12 - Tabulka uživatelů a vazby na práva v systému

Jak je vidět z Tabulky UsrRight, tak uživatel může získat právo na čtení nebo zápis pro celý časopis a jeho součásti nevyjímaje. Stejně tak mu, může být zakázáno některý časopis vidět nebo jen editovat. Systém práv je chápán, že pokud uživatel má právo zápisu, je mu automaticky uděleno právo i na čtení (i když ho v nastavení nemá přiřazené). Stejně tak je uživatelům povoleno/zakázáno vstupovat (zobrazí se v horním menu) do hlavních menu položek. Související vazby, které na obrázcích (Obr. 9, Obr. 10, Obr. 11) nejsou nastíněny, jsou vazba sloupce IdMag z tabulky UsrRight na tabulku Mag (poměr 1:N), IdCountry v tabulce Usr na tabulku Country (poměr N:1) a samozřejmě mezi tabulkou Usr a ostatními tabulkami (v poměru 1:N).

3.1 Logování změn

Protože se jedná o důležitá data, je potřeba mít v každém okamžiku data nejen dostupná ale i dohledatelná. Uživatel nebo vedoucí pracovník si může přát nahlédnout do historie změn, které byly provedeny nad určitým časopisem nebo nějakou jeho částí. Nejjednodušší možností jak toho dosáhnout je logovat všechny potřebné změny a operace nad určenými tabulkami. Logování samotné je realizováno pomocí databázových triggerů. Jedná se o uživatelsky definovanou událost nad databázovou tabulkou. Pomocí triggeru

jsme schopni zachytit veškeré operace nad touto tabulkou jako přidávání (INSERT), aktualizaci (UPDATE) nebo mazání (DELETE) záznamů. A právě tyto tři události, jsou zachytávány v našem administrativním systému a hned po změně databázové tabulky (Mag, MagIssue, MagRubric, ...) je stejný záznam přidán do jejich logovací tabulky (MagLOG, MagIssueLOG, ...) doplněné o logovací záznamy:

- **RowID** – jednoznačný identifikátor záznamu v tabulce
- **LOGFrom** – označuje datum od kdy je daný záznam v tabulce platný (originál toho záznamu je ve stejné podobě v hlavní tabulce)
- **LOGTo** – označuje datum do kdy je daný záznam v tabulce platný (originál toho záznamu se do tohoto data nacházel ve stejné podobě v hlavní tabulce)

Na příkladu níže můžete vidět ukázkou, jak je vytvořen Trigger pro tabulku Mag (hlavní tabulku časopisů) :

```
CREATE TRIGGER [dbo].[TMAG]
  ON [dbo].[Mag]
  AFTER INSERT,DELETE,UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    ----- BEGIN -----
    if ((SELECT count(1) FROM INSERTED) > 0)
    -- INSERT OR UPDATE
    BEGIN
        UPDATE Mag_LOG set LOGTo=getdate() FROM INSERTED where
        INSERTED.IdMag=Mag_LOG.IdMag and LOGTo='2050/01/01'
        INSERT
        Mag_LOG(LOGFrom,LOGTo, IdMag, MagNameShort, MagName, MagNameDe, Date
        Created, CreatedBy, ModifiedBy, DateModified, Yearly, DefaultVAT, IdC
        ountry, IdMagGroup, OrderInGroup, IsActive, DateDeleted)
        SELECT
        getdate(), '2050/01/01', IdMag, MagNameShort, MagName, MagNameDe, Dat
        eCreated, CreatedBy, ModifiedBy, DateModified, Yearly, DefaultVAT, Id
        Country, IdMagGroup, OrderInGroup, IsActive, CASE IsActive WHEN 1
        THEN NULL ELSE GETDATE() END FROM INSERTED
    END ELSE BEGIN
    -- DELETE
        UPDATE Mag_LOG set DateDeleted=getdate() FROM DELETED
        where DELETED.IdMag=Mag_LOG.IdMag
        select 1
    END
    ----- END -----
```

4 REALIZACE ADMINISTRATIVNÍHO SYSTÉMU

Součástí bakalářské práce je i praktická realizace administrativního systému. Pro úspěšnou realizaci bylo potřeba nejprve navrhnout strukturu tohoto systému a definovat jeho chování. Pro tuto část bylo použito reálného zadání z praxe, které bylo patřičně upraveno pro potřeby bakalářské práce a zároveň se zamezilo k případné shodě nebo podobnosti s reálným informačním systémem, který je používán v praxi.

K realizaci bakalářské práce bylo použito vývojové prostředí Microsoft Visual Web Developer 2010 Express, technologie C#, ASP.NET společně se skriptovacím jazykem JavaScript a nadstavbovou knihovnou jQuery (aktuální verze 1.7.1). Jako datové úložiště je použito Microsoft SQL Server R2 2008 Express. Pro vzhled aplikace bylo použito kaskádových (CSS) stylů. Stránky jsou optimalizovány pro webové prohlížeče Internet Explorer (verze 8 a vyšší), Mozilla Firefox (verze 6.0 a vyšší), Opera (verze 11.0 a vyšší) a Google Chrome (verze 1.0 a vyšší).

4.1 Návrh a rozvržení webové aplikace

Aplikace bude situována do 4 položek menu, mezi které patří „*Editace Magazínů*“, „*Editace uživatelů*“, „*Tisk harmonogramů*“ a „*Správa číselníků*“. Po přihlášení bude uživatel přesměrován na úvodní obrazovku, kde podle přidělených oprávnění bude dispozici menu nabídka v horní části webového prohlížeče. Jednotlivé položky z menu nabídky poskytují:

- **Editace magazínů** – zde může uživatel vytvářet, měnit nebo mazat časopisy, jeho položky a vazby mezi nimi. Dále je zde nastavení akcí jednotlivých časopisů a správa datumů pro časový harmonogram.
- **Editace uživatelů** – umožňuje spravovat uživatelské účty a práva přidělená na časopisy a položky menu
- **Tisk Harmonogramů** – zde se provádí volba časopisů, jednotlivých akcí pro zvolený měsíc a rok. Na základě této volby se vygeneruje časový plán (harmonogram) pro jednotlivé akce daných časopisů na specifikovaný měsíc.
- **Správa číselníků** – slouží k správě a manipulaci s číselníky (tabulkové hodnoty obsahující zpravidla 2 nebo 3 sloupce). Do tohoto menu spadá správa dat pro tabulky *Country* (obsahující ID a název státu) a *Author* (obsahující ID, jméno a stát pro autora rubriky).

4.2 Implementace Administrativního systému

4.2.1 Přihlašovací obrazovka


Při přístupu k Administrativnímu systému je uživatel nasměrován na přihlašovací obrazovku (viz. Obr. 12), kde po zadání přihlašovacího jména a hesla bude ověřen a pokud údaje souhlasí, bude přesměrován na úvodní obrazovku (viz. Obr. 13). Pro potřeby ověření hesla je použito HASH funkce, která zakóduje heslo do jedinečné podoby a ověří s HASH hodnotou hesla uloženého v databázi pro daného uživatele. Kódování hesla se provádí na straně klienta a o tuto funkcionalitu se stará JavaScriptová knihovna *JShash.js* v adresáři **js** webové aplikace. Funkce z této knihovny, která vrací zakódované heslo je *calcMD5* a přijímá jeden parametr (textovou podobu hesla). Zde je ukázka, jak se volá vytvoření HASH hesla při události *onclick* na tlačítku „Přihlásit se“ :

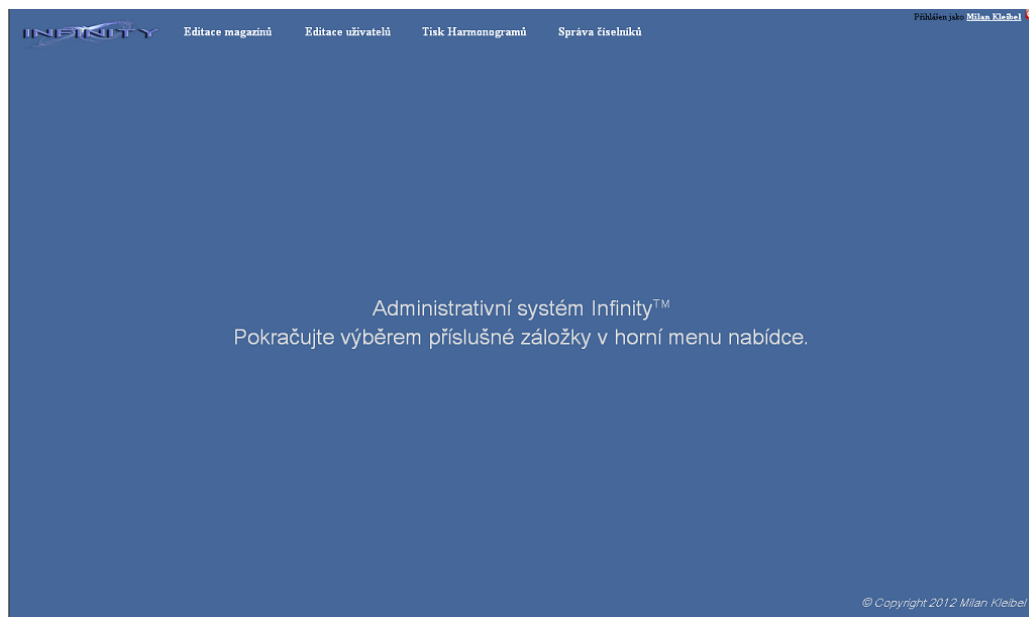
```
<script type="text/javascript" src="./js/JShash.js"></script>
<script type="text/javascript">
function g(ID) {
    return document.getElementById(ID)
}
function HASHfield() {
    g('TXB_Passwd').value = calcMD5(g('TXB_Passwd').value);
}
</script>
```

Ověření hesla je ošetřeno proti standardnímu útoku typu *SQLInjection*, aby nedošlo k neoprávněnému přihlášení do systému.



Obr. 13 - Přihlašovací okno

V pravém horním rohu aplikace se nachází informace o přihlášeném uživateli a ikona , pomocí které se uživatel může odhlásit ze systému. V levém horním rohu se nachází logo **Infinity**, které zároveň slouží jako odkaz na tuto úvodní stránku.

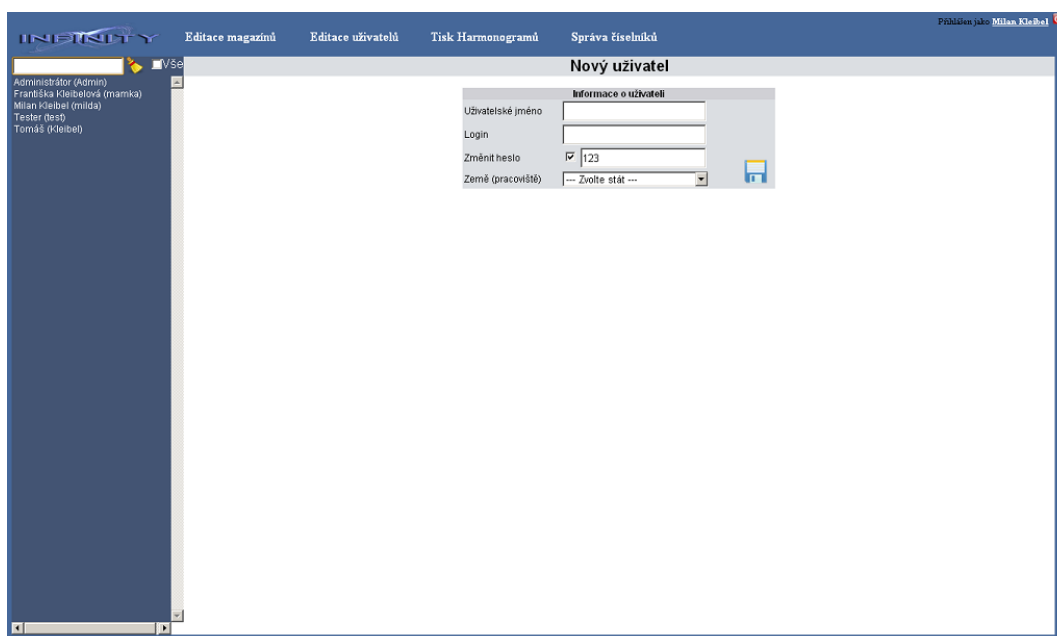


Obr. 14 - Úvodní obrazovka


4.2.2 Editace Uživatelů

Tato sekce je viditelná pouze pro administrátory systému. A jak už bylo zmíněno výše, zde se spravují uživatelské účty a nastavují příslušná oprávnění na časopisy a navigační menu položky.

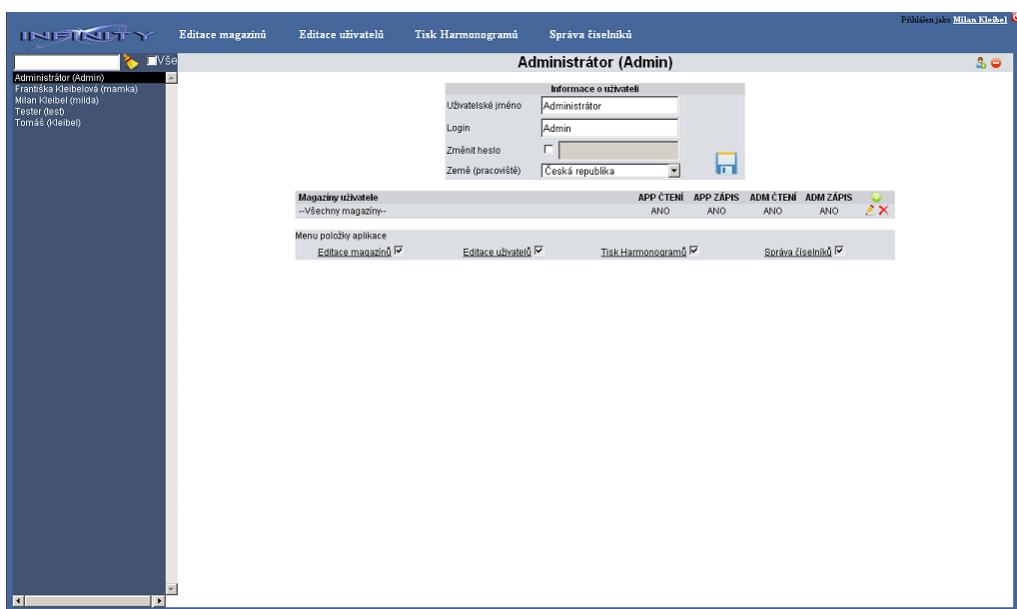
4.2.2.1 Nový uživatelský účet



Obr. 15 - Nový uživatelský účet




V této sekci se nachází administrace uživatelských účtů. Na obrázku (Obr. 15) vidíme obrazovku pro nového uživatele. Pro vytvoření stačí zvolit uživatelské jméno, login a zemi (pracoviště). Volitelná je volba hesla. Pokud heslo není zvoleno, vygeneruje se standardní heslo (generované podle pravidel správné tvorby hesla). Pomocí ikony  se zadané dají zkontrolovat a uložit do databáze (čímž dojde k vytvoření klienta). Pokud ale zvolený login již v databázi existuje, dojde k upozornění uživatele (administrátora) na tuto skutečnost a je vyzván ke změně loginu. Všechny tyto operace se provádí pomocí jQuery AJAX metody a děje se tak asynchronně.

4.2.2.2 Editace uživatelského účtu






Obr. 16 - Editace uživatelského účtu

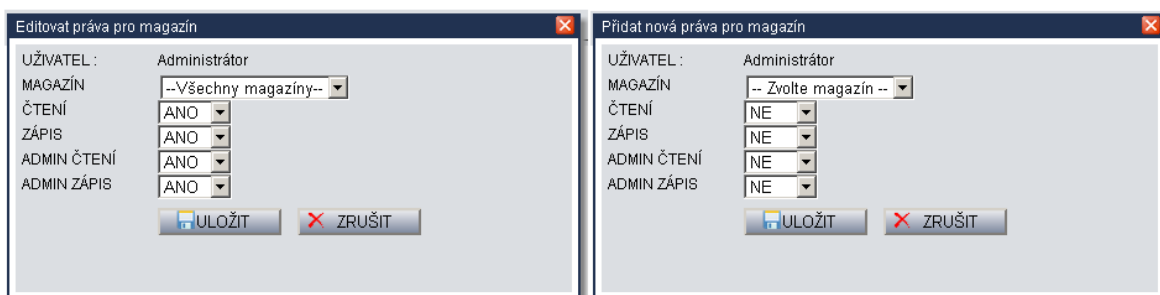
Po vytvoření uživatele se dostaneme na obrazovku editace uživatele (viz Obr. 15). Pod tabulkou uživatel se nachází tabulka magazínů a práv, které jsou k nim přiděleny (Obr. 16).

Magazíny uživatele	APP ČTENÍ	APP ZÁPIS	ADM ČTENÍ	ADM ZÁPIS	
--Všechny magazíny--	ANO	ANO	ANO	ANO	  

Obr. 17 - tabulka práv k daným magazínům

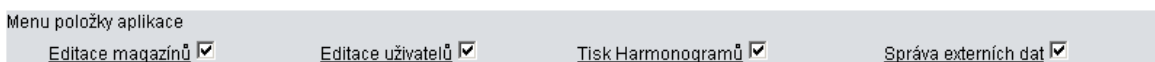
V této tabulce se nachází ikony pro přidání , editaci  a mazání  práv k magazínům. Kliknutím na ikony přidání (editace) se dostaneme do formuláře pro přidání (změnu) práv uživatele na magazín (Obr. 17). Po kliknutí na ikonu mazání se nás systém

dotáže, jestli si opravdu přejeme tento záznam smazat. Při potvrzení dojde k smazání záznamu z databáze.



Obr. 18 - Editace a přidání práv pro magazín




Pod tabulkou magazínů se nachází tabulka pro přiřazení práv na menu položky (Obr. 19). Práva se nastaví jednoduchým zaškrtnutím pole pro dané menu. Opět se změna ukládá asynchronně do databáze.

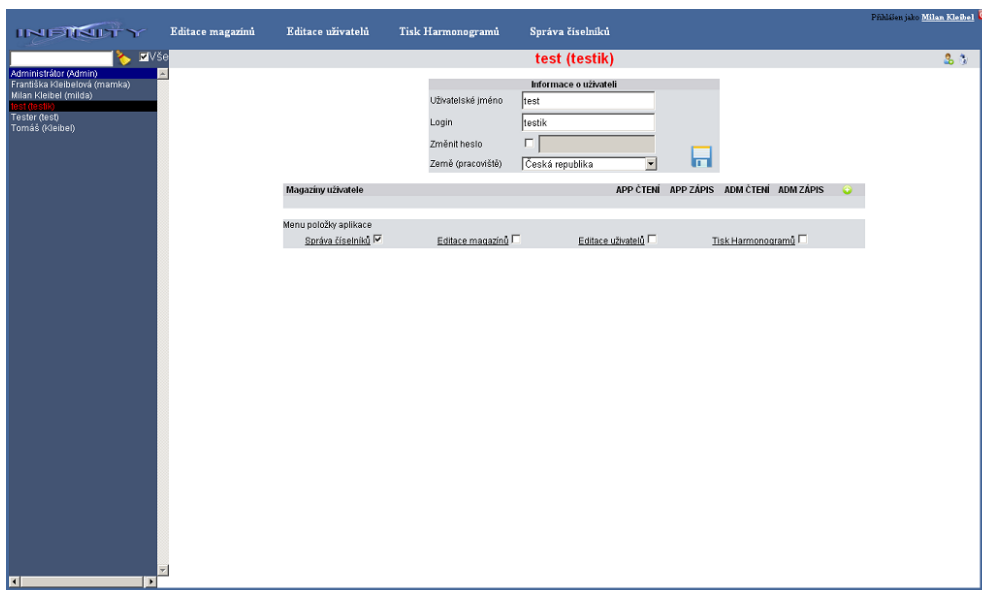


Obr. 19 - Editace a přidání práv na menu položky

4.2.2.3 Vytvoření, mazání a obnovení účtu


V aplikaci pro editaci uživatelů jsou dostupné ještě tři ikony (viz. Obr. 16, Obr. 20):

- **Vytvoření uživatelského účtu**  - zobrazí se obrazovka pro vytvoření uživatele (Obr. 15)
- **Smazání uživatelského účtu**  - dojde k dotazu zda skutečně chceme smazat tohoto uživatele. Při potvrzení, nedojde přímo ke smazání uživatele, pouze k jeho zneaktivnění. Zároveň dojde k přesměrování na obrazovku pro nového uživatele (Obr. 15)
- **Obnova uživatelského účtu**  - tato ikona je dostupná pouze pro zneaktivněné uživatele. Pro jejich zobrazení musí administrátor nejprve zobrazit i smazané účty (zaškrtnuté pole vše Vše) a zvolit daného uživatele. Po zvolení obnovy jsme opět vyzváni, zda si opravdu přejeme aktivovat uživatele a při potvrzení dojde k jeho opětovnému aktivování a přenesení na jeho profil.



Obr. 20 - Obnovení uživatelského účtu

4.2.2.4 Vyhledávání mezi účty

V levém menu, kde se nachází seznam uživatelů je ještě textové pole a ikona  (maže vyhledávací text) sloužící pro vyhledávání mezi uživateli (Obr. 21). Zaškrtnuté pole Vše umožňuje zpřístupnění zneaktivěných účtů. Vyhledávání se provádí pomocí JavaScriptu a je proto velmi rychlé.



Obr. 21 - Ukázka vyhledávání uživatelského účtu

4.2.3 Editace magazínů

Po přístupu do této části systému je uživateli zobrazeno prostředí pro manipulaci s magazíny a nastaveno standardní zobrazení na vydání a pro filtrovací pole použít stát odpovídající zemi, ve které uživatel pracuje (viz. Obr. 22). Zároveň žádný časopis není implicitně vybrán.

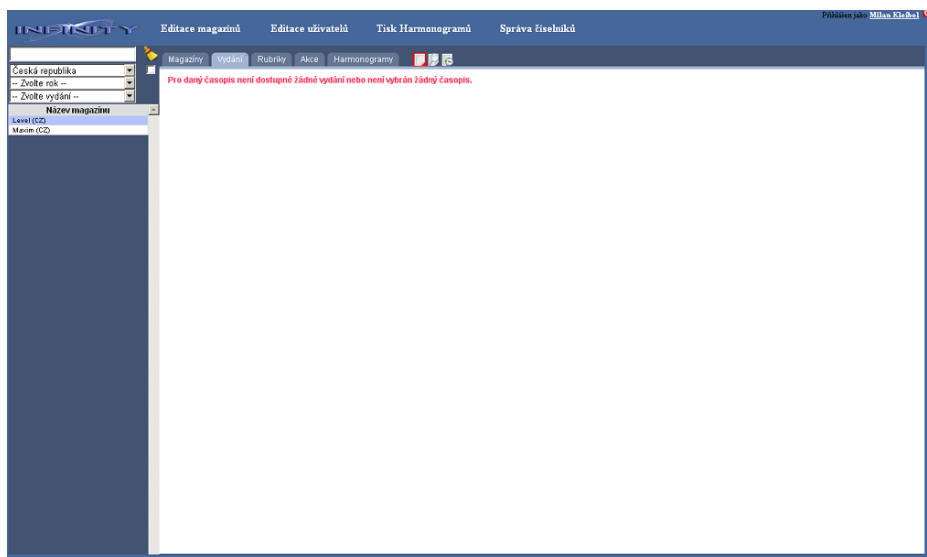
V této sekci je dostupných pět panelů (*Magazíny*, *Vydání*, *Rubriky*, *Akce*, *Harmonogramy*), ve kterých se realizuje hlavní část práce s magazíny. Popis a funkcionalitu v těchto panelech se budeme zabývat v dalších kapitolách. Nejprve objasnění a seznámení s ovládacími prvky, které nám usnadňují práci a můžeme je využít.

Za povšimnutí stojí tři ikony vedle panelové nabídky, z nichž jeden je červeně podbarven (označení jako aktivní). Jedná se o pohledy na data z databáze a v jaké formě budou prezentována uživateli. První (bráno zleva doprava) ikona, umožňuje nahlížet na data v stručném pohledu, druhá pak v detailním a třetí nám zobrazí historii změn, prováděných v daném panelu. Ukázka rozdílů v pohledech je ukázán na obrázcích Obr. 23, Obr. 24 a Obr. 25.

V levé části aplikace se opět nachází několik ovládacích prvků:

- seznam magazínů (obdobně jako v editaci uživatelů)
- vyhledávací textové pole (pro rychlé hledání mezi magazíny)
- ikona pro mazání textového pole
- zaškrtačací pole, pro umožnění nahlédnout na smazané (neaktivní) magazíny
- výběrové pole pro volbu státu – slouží i jako volba filtrace
- výběrové pole pro rok (vztahuje se k vybranému časopisu) – slouží jako filtrační podmínka pro práci se záznamy uvnitř panelu
- výběrové pole pro vydání (vztahuje se k vybranému časopisu) – slouží jako filtrační podmínka pro práci se záznamy uvnitř panelu

Vyhledávání je opět prováděno JavaScriptově a je proto velmi rychlé a interaktivní.



Obr. 22 – Úvodní zobrazení – editace magazínů

V stručném a detailním pohledu jsou uživateli dostupné funkce pro přidání a editaci záznamu. V některých panelech jsou dostupné i specifické funkce.

ID	Zkratka	Házev	Popis	Ročně	Daň[%]	Země	Group
4	CH	Chip	Časopis o PC technologiích	12	22.00	Slovenská republika	
1	LVL	Level	Počítačový magazín	12	21.00	Česká republika	IT
2	MXM	Maxim	Časopis pro pány	12	21.00	Česká republika	

Obr. 23 – Tabulka magazínů – stručný pohled

ID	Zkratka	Házev	Popis	Ročně	Daň[%]	Země	Group	Datum vytvoření	Vytvořil	Datum změny	Změnil
4	CH	Chip	Časopis o PC technologiích	12	22.00	Slovenská republika		27.05.2012 22:19:00	Milan Kleibel	27.05.2012 23:28:00	Milan Kleibel
1	LVL	Level	Počítačový magazín	12	21.00	Česká republika	IT	21.04.2012 00:00:00	Milan Kleibel	08.05.2012 00:34:00	Milan Kleibel
2	MXM	Maxim	Časopis pro pány	12	21.00	Česká republika		21.04.2012 00:00:00	Milan Kleibel	21.04.2012 00:00:00	Milan Kleibel

Obr. 24 – Tabulka magazínů – detailní pohled

Zobrazit záznamů nebo Zobrazit vše?

RowId	Platné od	Platné do	ID	Zkratka	Házev	Popis	Ročně	Daň[%]	Země	Group	Datum vytvoření	Vytvořil	Datum změny	Změnil
11	27.05.2012 23:28:00	01.01.2050 00:00:00	4	CH	Chip	Časopis o PC technologiích	12	22.00	Slovenská republika		27.05.2012 22:19:00	Milan Kleibel	27.05.2012 23:28:00	Milan Klei
10	27.05.2012 23:27:00	27.05.2012 23:28:00	4	CH	Chip	Časopis o PC technologiích	12	22.00	Slovenská republika		27.05.2012 22:19:00	Milan Kleibel	27.05.2012 23:27:00	Milan Klei
9	27.05.2012 23:27:00	27.05.2012 23:27:00	4	CH	Chip	Časopis o PC technologiích	12	22.00	Slovenská republika		27.05.2012 22:19:00	Milan Kleibel	27.05.2012 23:27:00	Milan Klei
8	27.05.2012 23:25:00	27.05.2012 23:27:00	4	CH	Chip	Časopis o PC technologiích	12	22.00	Slovenská republika		27.05.2012 22:19:00	Milan Kleibel	27.05.2012 23:25:00	Milan Klei
7	27.05.2012 23:25:00	27.05.2012 23:25:00	4	CH	Chip	Časopis o PC technologiích	12	22.00	Slovenská republika		27.05.2012 22:19:00	Milan Kleibel	27.05.2012 23:25:00	Milan Klei
6	27.05.2012 22:19:00	27.05.2012 23:25:00	4	CH	Chip	Časopis o PC technologiích	12	22.00	Slovenská republika		27.05.2012 22:19:00	Milan Kleibel	27.05.2012 22:19:00	Milan Klei
5	08.05.2012 00:34:00	01.01.2050 00:00:00	1	LVL	Level	Počítačový magazín	12	21.00	Česká republika	IT	21.04.2012 00:00:00	Milan Kleibel	08.05.2012 00:34:00	Milan Klei
4	21.04.2012 22:33:00	01.01.2050 00:00:00	2	MXM	Maxim	Časopis pro pány	12	21.00	Česká republika		21.04.2012 00:00:00	Milan Kleibel	21.04.2012 00:00:00	Milan Klei
3	21.04.2012 22:33:00	08.05.2012 00:34:00	1	LVL	Level	Počítačový magazín	12	21.00	Česká republika		21.04.2012 00:00:00	Milan Kleibel	21.04.2012 00:00:00	Milan Klei
2	21.04.2012 22:19:00	21.04.2012 22:33:00	2	MXM	Maxim	Časopis pro pány	12	21.00			21.04.2012 00:00:00	Milan Kleibel	21.04.2012 00:00:00	Milan Klei
1	21.04.2012 22:18:00	21.04.2012 22:33:00	1	LVL	Level	Počítačový magazín	12	21.00			21.04.2012 00:00:00	Milan Kleibel	21.04.2012 00:00:00	Milan Klei

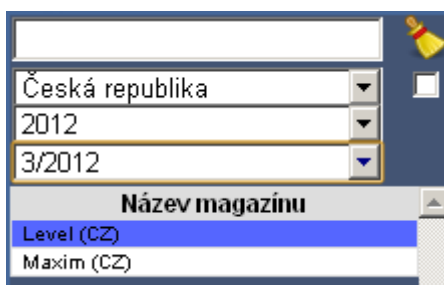
Obr. 25 – Tabulka magazínů – historie změn (přesahuje rozsah stránky)

4.2.3.1 Filtrační podmínky

Ve všech panelech je využíváno nastavení filtračních podmínek v levém menu (Obr. 26), které ovlivňují zobrazené záznamy pro uživatele. Filtr se skládá z následujících ovládacích prvků:

- **Textové pole** – podle zadaného textu se změní seznam zobrazených časopisů. Ovlivní tedy možnost volby magazínu (pokud magazín není zobrazen, není možné s ním pracovat). Navíc v panelu pro *magazíny*, pokud není magazín zvolen, dojde k vygenerování záznamů v panelu pro všechny časopisy.
- **Výběr státu** – rychlá volba, kde lze rychle a snadno zvolit skupinu časopisů, patřících do daného státu. Obsahuje i volbu *Všechny státy*.
- **Výběr roku** – tato volba je spojena s vybraným časopisem. Nabídne uživateli, seznam roků, v nichž se časopis vydával. Může být použito jako filtr pro zúžení zobrazených dat zejména v panelech *Vydání*, *Rubriky a Harmonogramy*.
- **Výběr vydání** - podobně jako Výběr roku, je závislé na výběru časopisu. Obsah tohoto prvku je ovlivněn zvoleným rokem. Výběr roku a vydání je naplňován automaticky v závislosti na výběru a děje se tak asynchronně pomocí jQuery.

- **Volba magazínu** – umožňuje označit i odznačit magazín. Volba magazínu je důležitou podmínkou pro všechny panely kromě panelu *Magazíny*.
- **Ikona štětce** – maže obsah textového vyhledávacího pole (po smazání se provede aktualizace seznamu)
- **Zaškrťovací pole** – umožňuje pracovat s neaktivními časopisy a vyhledávat mezi nimi.



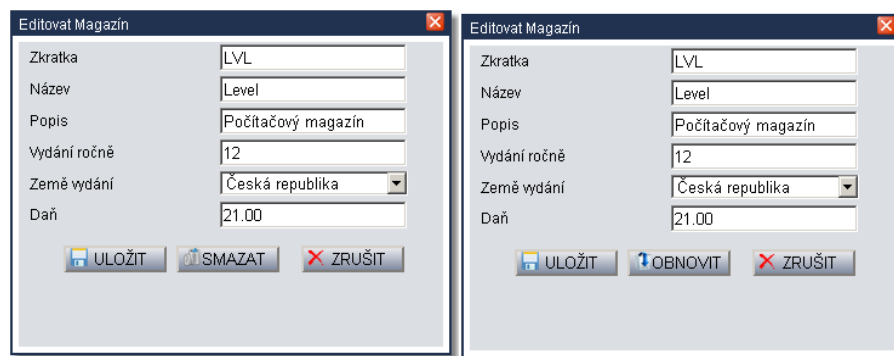
Obr. 26 – Filtrovací podmínky

4.2.3.2 Panel Magazíny

Na předchozích obrázcích (Obr. 23, Obr. 24 a Obr. 25) je vidět jaká data se v tomto panelu zobrazují a s jakými se dá manipulovat. Seznam zobrazených záznamů je ovlivněn nastavením v levém menu. V tomto panelu je možné pracovat s daty pomocí dvou ikon. Jedna pro přidání záznamu (na úrovni hlavičky tabulky) a druhá pro editaci záznamu (umístěná na začátku každého řádku). Po zvolení přidání záznamu se zobrazí formulářové okno pro zadání údajů k novému magazínu (Obr. 27).

Obr. 27 – Nový magazín

Po volbě editace záznamu, se zobrazí vyskakovací okno pro editaci zvoleného záznamu (Obr. 28).



Obr. 28 – Editace magazínu (mazání/obnova)

V těchto oknech je možné daný magazín smazat (zneaktivnit) nebo obnovit (zaktivnit). Aby nedocházelo k nechtěným mazáním nebo obnovám, je uživateli nabídnuta otázka, zda si opravdu přeje tuto akci vykonat. Při vytvoření nového magazínu se provede automaticky přiřazení *Akce - Vydání časopisu* k tomuto magazínu.

4.2.3.3 Panel Vydání

Opět nabízí tři pohledy na data v tomto panelu, obsahující informace dané pro vydání časopisu. Aby bylo možno zobrazit záznamy, je nutnou podmínkou, aby uživatel zvolil časopis z levého menu. Zde se nám může vyplatit použití filtrovacích polí v levém menu (pokud pro daný magazín bude existovat mnoho vydání) pro zúžení vyspaných záznamů, se kterými si přejeme pracovat.

Stejně jako v předchozím panelu se zde nachází ikona pro přidání záznamu vydání a k jeho editaci (Obr. 30). Při volbě nového vydání mohou nastat dva případy:

- **Přidávané vydání je první pro tento magazín** – v tomto případě dojde k předvyplnění pouze názvu a roku vydání (bráno z aktuálního roku a název zvolen jako číslo vydání/rok vydání – pro 1. vydání roku 2012 tedy 1/2012). S vytvořením tohoto vydání se *Datum vydání* uloží toto datum k danému vydání do tabulky Harmonogramů.
- **Již existuje předchozí vydání** – zde se automaticky zjistí informace z předchozího vydání a podle dostupných informací se předvyplní pole *Název*, *Datum vydání*, *Rok*, *Dodavatel*, *Počet výtisků v oběhu*, *Počet prodaných výtisků*. Název se odvodí z předchozího názvu (navýšení čísla vydání o jedna). Datum vydání se automaticky dopočítá a z tohoto data se použije informace o roku (K tomuto účelu se použije informace z tabulky magazínů - *Ročně*). Jelikož již existuje nějaké vydání a je nějakým způsobem nastavené, dá se předpokládat, že toto nové vydání bude mít

podobné (ne-li stejné) nastavení. Proto se zároveň s vytvoření nového vydání vygenerují stejné vazby (Rubriky a vydání), data v harmonogramu (posunutá o příslušný časový úsek – záznam *Ročně*) jako pro již existující záznam. Tato automatizovaná umožňuje uživateli pracovat rychle a bez zbytečného zdržování.

Obr. 29 – Přidání (první a každé další)

Obr. 30 – Editace vydání

Oproti panelu *Magazíny* je zde navíc tlačítko *Přidat více vydání* (Obr. 31). Podmínkou pro použití tohoto tlačítka je přítomnost alespoň jednoho vydání v časopise. Po zvolení, je uživatel dotázán na počet vydání, která se mají přidat. Automatizovanost je i v tomto případě stejná, jako pro přidání jednoho vydání. Provádí se pro každé vydání, které se takto generuje.

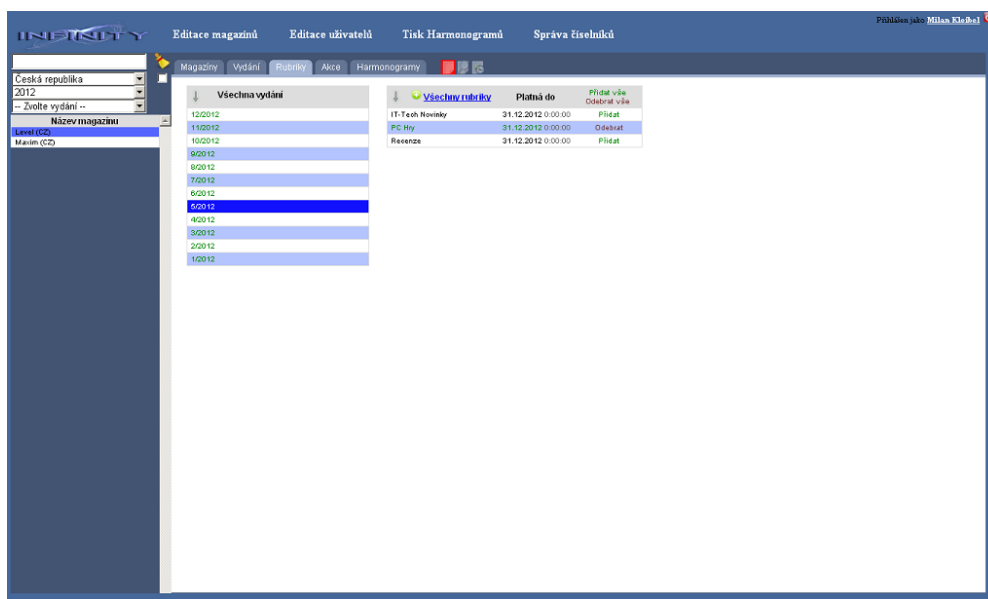
Obr. 31 – Přidání více vydání

4.2.3.4 Panel Rubriky

V této části aplikace se nastavují vazby mezi jednotlivými vydáními a rubrikami. Jednoznačně se zde určí, která vydání budou obsahovat jaké rubriky. Případně ve kterých vydáních se objeví daná rubrika. Vytvoření vazby se provádí dvěma způsoby:

1. **Zvolíme vydání** – po zvolení vydání se nám objeví rubriky a v pravé části této tabulky i text zda *přidat* nebo *odebrat* rubriku z vydání. Nebo je tu možnost *přidat vše* či *odebrat vše*. Tato akce se provádí asynchronně pomocí funkcí jQuery.
2. **Zvolíme rubriku** – obdobně jako v bodě 1. Zde naopak přiřazujeme zvolenou rubriku zobrazeným vydáním. Zde se mohou významně osvědčit filtrovací pole. Pokud budu chtít přidat jednu specifickou rubriku pro všechny vydání z roku 2012, tak si jednoduše nastavím rok 2012 ve filtru a pro danou rubriku zvolím *přidat vše*.

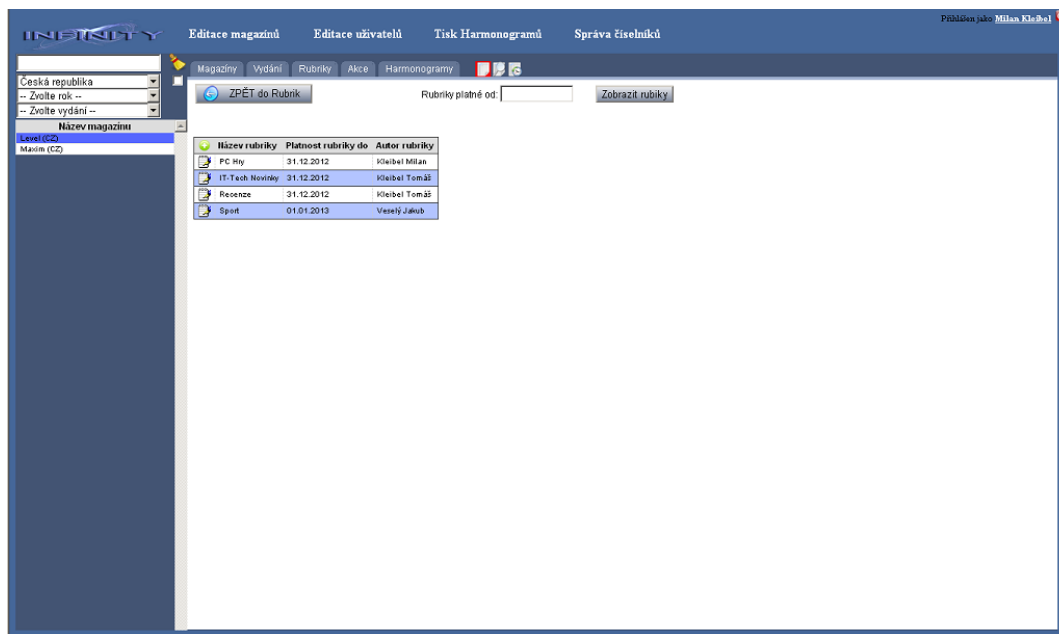
V těchto tabulkách jsou dostupné i dvě stejné ikony – šipka dolů. Slouží k řazení seznamu vydání/rubrik. Řazení je nastaveno standardně podle abecedy. Po kliknutí na danou ikonu se řazení změní podle přidanych/odebraných vydání/rubrik.



Obr. 32 – Rubriky (tvorba vazeb vydání-rubriky)

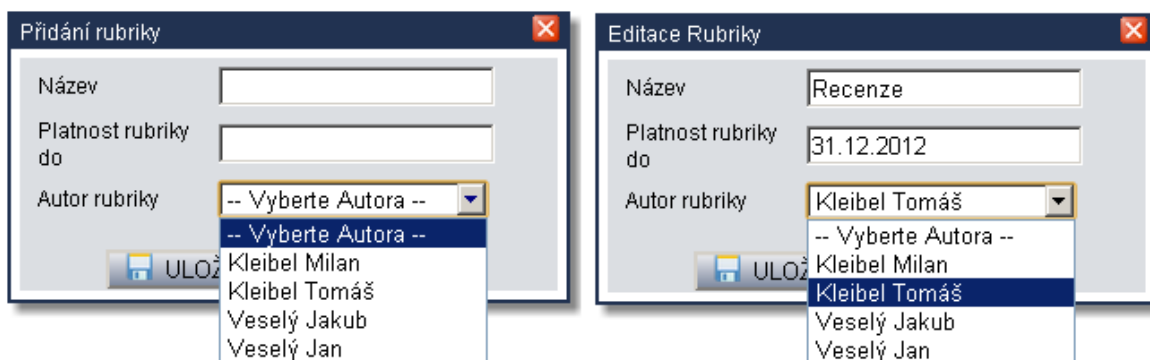
V tabulce rubrik je dostupný i hypertextový odkaz *Všechny rubriky* a ikona, které odkazují na správu rubrik (Obr. 33).

Na této stránce může uživatel vytvářet a editovat rubriky k daným magazínům. Slouží k tomu ikony pro editaci a přidání, které otevírají vyskakovací okna (Obr. 34).



Obr. 33 – Editace rubrik

Pro tuto část je zde dostupný i doplňující filtr na *Platnost rubriky*. Zobrazí jen rubriky, které jsou platné od zadaného data. K dispozici je i tlačítko pro přechod zpět na vazby mezi vydáními a rubrikami. Pro tuto tabulku rubrik jsou dostupné opět tři pohledy.

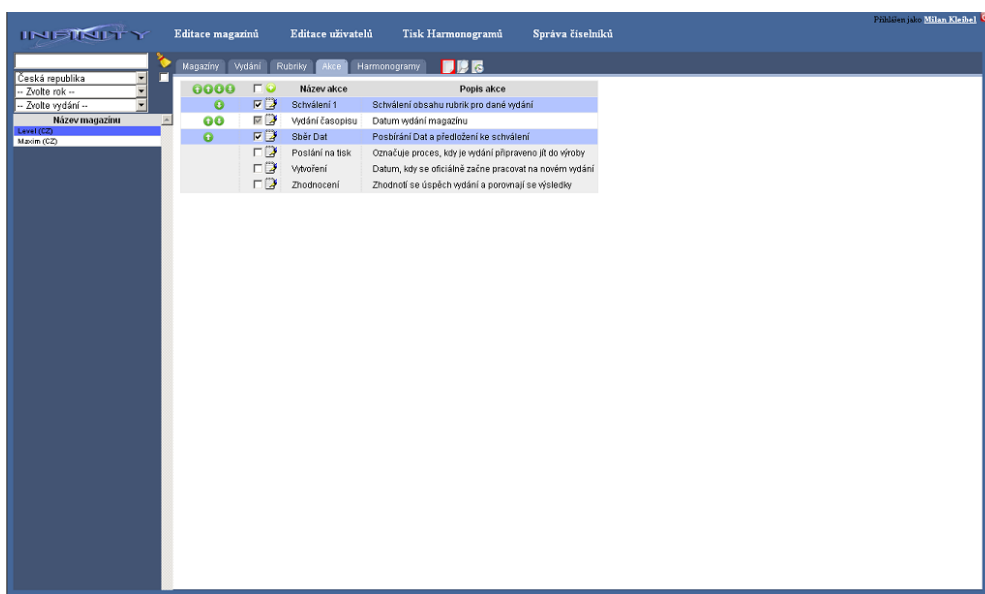


Obr. 34 – Přidání a editace rubriky

4.2.3.5 Panel Akce

V panelu *Akce* (Obr. 35) se nastavují akce, které je potřeba vykonat, před tím než časopis vyjde a i po tomto datu. Nastavení těchto akcí je důležité pro tvorbu harmonogramů. Stanoví se zde pořadí akcí, které konkrétně budou zahrnuty do pracovního postupu výroby vydání daného časopisu. K těmto akcím je později přiřazeno datum v panelu *Harmonogramy*.

Pro tabulku akcí je opět dostupná trojice pohledů, ikony pro přidání a editaci akce. Navíc je zde dostupná sada šipkových ikon umožňující měnit pořadí akcí v rámci zvoleného časopisu. Lze měnit pouze zvolenou akci (byla označena kliknutím) a akce musí být přiřazena k magazínu (zaškrťovací pole - zaškrtnuto). Přiřazení akce k magazínu se provádí pomocí jQuery. V tabulce je jedna akce *Vydání časopisu*, která nemůže být odebrána ze seznamu akcí. Je to proto, že se musí vyskytovat u každého časopisu a je automaticky vytvořena s novým časopisem.



Obr. 35 – Panel akcí

4.2.3.6 Panel Harmonogramy

Zde se provádí zadávání dat a je jednoznačně určen termín pro každou akci k danému vydání v rámci zvoleného časopisu. Datum se zadává přes kalendář, ale je podporováno i zadávání datumů manuálně. V případě, že je datum zadáno v nesprávném (nepodporovaném) formátu, dojde k upozornění (při pokusu o uložení). Změněná pole jsou pro lepší orientaci podbarvena červeně. Většina datumů je generována automaticky při vytvoření vydání, ale může dojít k situaci, kdy je potřeba daný termín upravit nebo některá data posunout (neočekávaná situace například). Tabulka podporuje navigaci mezi jednotlivými buňkami pomocí šipek. Data, která se zde zvolí, budou použita ke generování harmonogramu a v menu *Tisk harmonogramu*.

Magazín	1/2012	2/2012	3/2012	4/2012	5/2012	6/2012	7/2012	8/2012	9/2012	10/2012	11/2012	12/2012
Schválení 1	20.01.2012	20.02.2012	20.03.2012				20.07.2012	20.08.2012	20.09.2012	20.10.2012		20.12.2012
Vydání časopisu	26.01.2012	26.02.2012	26.03.2012	26.04.2012	26.05.2012	26.06.2012	26.07.2012	26.08.2012	26.09.2012	26.10.2012	26.11.2012	26.12.2012
Sběr Dat	10.01.2012	10.02.2012	10.03.2012				10.07.2012	10.08.2012				

Obr. 36 – Tabulka datumů pro přiřazené akce k magazínu

4.2.3.7 Práva

Pokud uživatel má přiděleny práva na daný magazín, uvidí a může měnit data přesně tak, jak je uvedeno na obrázcích v kapitole 4.2. Pokud ale má jenom právo ke čtení nebo žádné právo na magazín, je situace odlišná. Při právu čtení uživatel může prohlížet data, ale nemá možnost je nijak měnit. Pokud nemá dané právo nastaveno, daný časopis se mu vůbec nezobrazí.

Editovat Magazín

Zkratka:

Název:

Popis:

Vydání ročně:

Země vydání:

Daň:

Obr. 37 – Právo čtení magazínu

4.3 Tisk Harmonogramů

V této menu nabídce se realizuje finální část administrátorské práce a to konečné vytvoření harmonogramu na celý měsíc. Postup vygenerování harmonogramu je snadný:

1. Uživatel zvolí magazín(-y) – volba se provede kliknutím na jednotlivé časopisy případně zruší výběr (opětovné kliknutí na označený časopis). Po volbě časopisu se uživateli nabídne seznam akcí a doplní se seznam roků, ve kterých časopisy vychází.
2. Z nabídky akcí si vybere jen ty akce, které chce generovat – možnost volby více akcí přes klávesu SHIFT. Po zvolení alespoň jedné akce dojde k zpřístupnění tlačítka *VYTVOŘ HARMONOGRAM*.
3. Výběr roku – standardně nastaven aktuální rok (předpokládá se, tisk harmonogramu pro aktuální měsíc)
4. Výběr měsíce - standardně nastaven aktuální měsíc
5. Zmáčknout tlačítko pro vytvoření harmonogramu.

V dolní části tabulky se potom vygeneruje harmonogram, ve kterém jsou obsažena data pro jednotlivé akce. Formát, ve kterém jsou data prezentována je :

Zkratka časopisu + mezera + Název vydání + nový řádek + Název akce

A tato informace je vložena do příslušné buňky tabulky označující datum (nastaveno v tabulce harmonogramů).

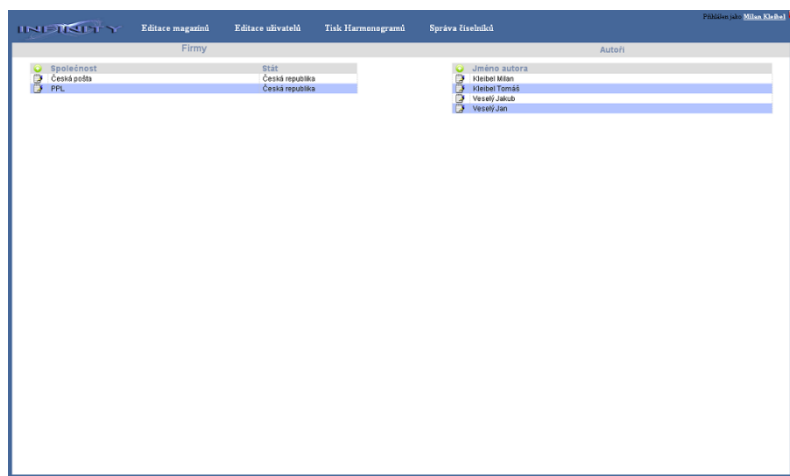
Pondělí	Úterý	Středa	Čtvrtek	Pátek	Sobota	Neděle
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25 CH 6/2012 Vydání časopisu	26 LVL 6/2012 Vydání časopisu	27	28	29	30	

Obr. 38 – Vygenerovaný harmonogram

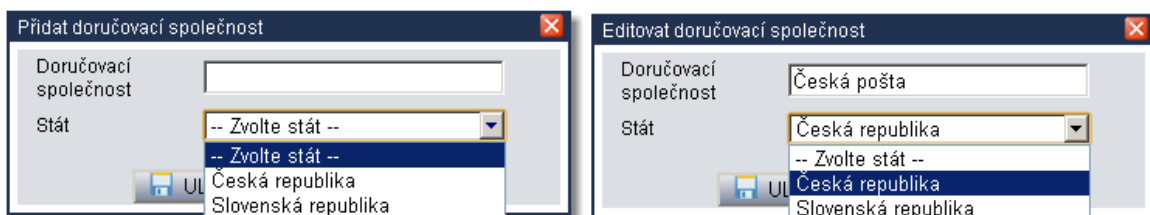
V levém menu je také přístupné vyhledávací pole, které usnadní hledání časopisu a jeho volby. V případě dlouhého seznamu časopisů, se toto vyhledávání stává užitečným pomocníkem pro výběr časopisů.

4.4 Správa číselníků

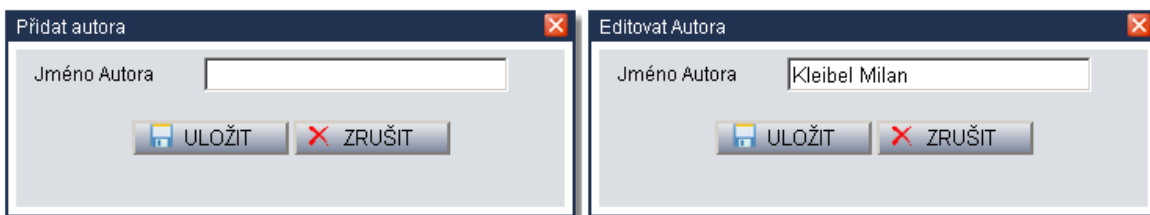
Pod touto menu nabídkou se skrývá správa dvou číselníků a to dodavatelských firem a autorů rubrik. Pro každou tabulku jsou dostupné ikony pro přidání a editaci záznamu, přes které uživatel pracuje se záznamy. Číselník firem je použit v panelu *Vydání* a určuje, každému vydání jeho dodavatele. Jména autorů jsou použita v tabulce rubrik, kdy ke každé rubrice přiřadíme právě jednoho jejího autora.



Obr. 39 – Číselníky Firem a Autorů



Obr. 40 – Přidání a editace doručovací firmy



Obr. 41 – Přidání a editace jména autora rubriky

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a realizovat administrativní systém pro časopisové nakladatelství. Realizace této aplikace byla provedena nad platformou Microsoft .NET 4.0. Hlavním úkolem, kladeným na systém, byla rychlost, jednoduchost a do jisté míry automatizovaná procesů. Aplikace se zabývá jak administrací uživatelů, tak nastavením a úpravami jednotlivých časopisů, spolu s jejich vydáními a rubrikami. Výstupem tohoto systému je hotová hierarchie v rámci magazínu a vygenerovaný časový harmonogram, podle kterého se řídí výroba vydání.

V teoretické části jsem se zaměřil na technologie použité pro praktickou realizaci zadání. Především jsem zde nastínil historické vývoje těchto technologií, jak fungují, případně i jejich výhody a nevýhody. Nejvíce jsem se v teoretické části zaměřil na platformu .NET, jelikož tvoří základ mé práce.

V praktické části jsem nejprve popsal databázovou strukturu pro potřeby aplikace a ukázal propojení jednotlivých tabulek. V druhé části jsem představil hotovou aplikaci administrativního systému. Nejtěžší pro mne bylo převedení některých funkcionalit z programovacího jazyku C# na JavaScriptový a jQuery kód, aby bylo dosaženo plynulejšího a rychlejšího chodu aplikace. Dalším problémem bylo navržení a implementování automatických akcí, zejména spojené s vytvářením nových vydání. Pro mne jako programátora (nikoliv designéra) bylo i náročné se vypořádat s grafickým designem aplikace, což se mi podle mého úsudku povedlo splnit v přijatelné úrovni.

I přesto, že aplikace splňuje všechny dané cíle bakalářské práce, je zde několik možností jak tuto aplikaci vylepšit. Například bych navrhl rozšíření v podobě modulů, které by pracovali s hierarchickou strukturou časopisů v podobě grafického designera, propracovaného schvalovacího procesu spojeného s výrobou vydání nebo nějakými grafy popisujícími důležité informace (například prodejnost časopisů).

Závěrem bych podotkl, že mi tato práce pomohla lépe pochopit vývoj interaktivních informačních systémů, ačkoliv nebyly použity, ve všech směrech, nejnovější dostupné technologie, protože ne vždy moderní znamená nejlepší.

ZÁVĚR V ANGLIČTINĚ

Main goal of this thesis was to design and develop a content management system for a journal publisher. I used Microsoft .NET 4.0 platform for implementation. Swiftiness, simplicity and process automation were the main requirements for the application. System covers user administration and settings or updates of magazines and their issues and rubrics. As a result the system provides complete structure for magazines and time schedule to control development process for each issue.

In theoretical part of the thesis I focused on technology I was using for realization of the task. I described historical development of these technologies, how they work and their advantages and disadvantages. I focused mainly on Microsoft .NET platform, as it is the base technology for my thesis.

In practical part I described database structure for application purposes and shown relationship between these tables. Second part is describing Administration system itself and how to work with it. I faced several challenges when I was developing the application. One of the problems was that I moved some functionality from C# language to JavaScript and jQuery, because I needed to achieve to faster and more fluent execution of my web application. Another challenge was the implementation of automatic actions such as creating new issue. Because I'm a programmer (and not a web designer) it was a bit hard for me to create appropriate graphic design, but in my judgment I achieved solid design level throughout the whole application.

There are several possibilities how to continue with improvement of the system, even though the application meets all the objectives required in the thesis. For example I would like to extend the application through new modules, such as graphical designer for work with magazine structure. Another idea is to allow complete management of workflow including approving specific actions and materials in issue production. Also there could be some graphs to show important statistics (e.g. marketability of journals).

At the end I would like to say that this thesis helped me to better understand the development process of interactive information systems. I however did not use the newest technology for all parts of the application, because modern doesn't always mean the best.

SEZNAM POUŽITÉ LITERATURY

- [1] Co je .NET Framework. *Co je .NET framework | Svět hostingu - vše o hostingu a doménách* [online]. 2009 [cit. 2012-05-28]. Dostupné z: <http://svet-hostingu.cz/2009/10/12/co-je-net-framework/>
- [2] Historie .NET. *Stackoverflow* [online]. 2009 [cit. 2012-05-29]. Dostupné z: <http://stackoverflow.com/questions/1083368/whats-the-early-history-of-the-net-framework>
- [3] Verze .NET. *Aspnet.cz* [online]. 2007 [cit. 2012-05-29]. Dostupné z: <http://www.aspnet.cz/Articles/161-jeste-jednou-a-dukladneji-k-verzim-microsoft-net-frameworku.aspx>
- [4] .NET Framework. *.NET Framework Versions and Dependencies* [online]. 2010 [cit. 2012-05-30]. Dostupné z: <http://msdn.microsoft.com/en-us/library/bb822049.aspx>
- [5] Framework 4.0. *Microsoft .NET Framework Wiki* [online]. 2009 [cit. 2012-05-30]. Dostupné z: <http://microsoft-net-framework4.software.informer.com/wiki/>
- [6] Microsoft .NET Framework 3.5. *Microsoft .NET Framework 3.5* [online]. 2007 [cit. 2012-05-30]. Dostupné z: <http://www.microsoft.com/downloads/cs-cz/details.aspx?FamilyID=333325fd-ae52-4e35-b531-508d977d32a6>
- [7] Microsoft .NET Framework 4. *Microsoft .NET Framework 4* [online]. 2011 [cit. 2012-05-30]. Dostupné z: <http://www.microsoft.com/downloads/cs-cz/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992>
- [8] Architektura .NET Framework rozebrána dopodrobna. *SWMAG.cz* [online]. 2010 [cit. 2012-05-30]. Dostupné z: <http://www.swmag.cz/670/architektura-net-framework-rozebrana-do-podrobna/>
- [2] .NET Framework. *Academic Tutorials* [online]. 2010 [cit. 2012-05-30]. Dostupné z: <http://www.academictutorials.com/microsoft.net/dotnet-framework.asp>
- [3] Programovací jazyk C#. *VŠB | Katedra informatiky FEI VŠB-TUO* [online]. 2007 [cit. 2012-05-31]. Dostupné z: <http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text/>
- [4] MACDONALD, Matthew a Mario SZPUSZTA. *ASP.NET 3.5 a C# 2008: tvorba dynamických stránek profesionálně*. Vyd. 1. Brno: Zoner Press, 2008, 1584 s. ISBN 978-807-4130-083.

- [5] ASP.NET 4 and Visual Studio 2010 Web Development Overview. *ASP.NET* [online]. 2012 [cit. 2012-05-31]. Dostupné z: <http://www.asp.net/whitepapers/aspnet4>
- [6] Introduction to SQL. REFSNES, Hege. *W3schools* [online]. Hoboken, N.J.: Wiley, c2010 [cit. 2012-05-31]. Dostupné z: http://www.w3schools.com/sql/sql_intro.asp
- [7] ORACLE. *Oracle documentation* [online]. 2010 [cit. 2012-05-31]. Dostupné z: <http://docs.oracle.com/>
- [8] MS SQL Server. *Promotic* [online]. 2011 [cit. 2012-06-01]. Dostupné z: <http://www.promotic.eu/cz/pmdoc/Subsystems/Db/MsSQL/Install.htm>
- [9] Historie programovacího jazyka C#. *Historie a vývoj jazyka C (od C až po C#)* [online]. 2003 [cit. 2012-06-01]. Dostupné z: <http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xcernoh1.htm>
- [10] C# Overview. *Visual C# Certification, Development, Studio, Training, Tutorials* [online]. 2010 [cit. 2012-06-01]. Dostupné z: http://en.csharp-online.net/CSharp_Overview
- [11] Stack Overflow. *Stackoverflow* [online]. 2008 [cit. 2012-06-01]. Dostupné z: <http://stackoverflow.com/>
- [12] MSDN Library: SQL Server Language Reference. MICROSOFT. *MSDN Library* [online]. 2008 [cit. 2012-01-17]. Dostupné z: <http://msdn.microsoft.com/cs-cz/library/default.aspx>
- [13] JQuery. *JQuery - write less, do more* [online]. 2005 [cit. 2012-06-01]. Dostupné z: <http://jquery.com/>
- [21] ASP.NET 4. *What's New in ASP.NET 4 and Visual Web Developer* [online]. 2011 [cit. 2012-06-03]. Dostupné z: <http://msdn.microsoft.com/en-us/library/s57a598e.aspx>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

.NET	Microsoft .NET Framework
ADO	ActiveX Data Objects
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
BCL	Base Class Library
CLI	Common Language Infrastructure
CLR	Common Language Runtime
CLS	Common Language Specification
COM	Component Object Model
CSS	Cascading Style Sheets
DCOM	Distributed Component Object Model
GC	Garbage Collector
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IIS	Internet Information Services
ISO	International Organization for Standardization
JIT	Just In Time
JSON	JavaScript Object Notation
LINQ	Language-Integrated Query
MS	Microsoft
MSIL	Microsoft Intermediate Language
MVC	Model-View-Controller
NGWS	Next Generation Windows Services
PE	Portable Executable

PHP	Personal Home Page
POX	Plain Old XML
RDBMS	Relational Database Management System
SEQUEL	Structured English Query Language
SP	Service Pack
SQL	Structured Query Language
TPL	Task Parallel Library
VB	Visual Basic
VS	Visual Studio
WCF	Windows Communication Foundation
WF	Windows Workflow Foundation
WPF	Windows Presentation Foundation
XML	Extensible Markup Language

SEZNAM OBRÁZKŮ

Obr. 1 - Souhrn historických verzí .NET Frameworku a které verze jsou již instalovány operačním systémem Windows [4].....	18
Obr. 2 - NET Framework zásobník	18
Obr. 3 - Architektura .NET Frameworku	19
Obr. 4 - <i>Infrastruktura společných jazyků</i>	19
Obr. 5 - Souběžné běhové prostředí CLR.....	20
Obr. 6 - Proces kompilace zdrojového do nativního kódu	20
Obr. 7 - Třídy z BCL jmenného prostoru	21
Obr. 8 - Kompilace webové stránky ASP.NET	25
Obr. 9 - Základní schéma databázové struktury Administrativního systému.....	34
Obr. 10 - Tabulky Mag, MagIssue a MagRubric a jejich logovací tabulky	35
Obr. 11 - Tabulky MagIssueRubric a MagAction a jejich logovací tabulky.....	35
Obr. 12 - Tabulka uživatelů a vazby na práva v systému	36
Obr. 13 - Přihlašovací okno	39
Obr. 14 - Úvodní obrazovka	40
Obr. 15 - Nový uživatelský účet	40
Obr. 16 - Editace uživatelského účtu	41
Obr. 17 - tabulka práv k daným magazínům	41
Obr. 18 - Editace a přidání práv pro magazín.....	42
Obr. 19 - Editace a přidání práv na menu položky	42
Obr. 20 - Obnovení uživatelského účtu	43
Obr. 21 - Ukázka vyhledávání uživatelského účtu	43
Obr. 22 - Úvodní zobrazení – editace magazínů.....	44
Obr. 23 – Tabulka magazínů – stručný pohled.....	45
Obr. 24 – Tabulka magazínů – detailní pohled.....	45
Obr. 25 – Tabulka magazínů – historie změn (přesahuje rozsah stránky).....	45
Obr. 26 – Filtrovací podmínky	46
Obr. 27 – Nový magazín.....	46
Obr. 28 – Editace magazínu (mazání/obnova).....	47
Obr. 29 – Přidání (první a každé další).....	48
Obr. 30 – Editace vydání	48
Obr. 31 – Přidání více vydání	48

Obr. 32 – Rubriky (tvorba vazeb vydání-rubriky).....	49
Obr. 33 – Editace rubrik	50
Obr. 34 – Přidání a editace rubriky.....	50
Obr. 35 – Panel akcí.....	51
Obr. 36 – Tabulka datumů pro přiřazené akce k magazínu	52
Obr. 37 – Právo čtení magazínu.....	52
Obr. 38 – Vygenerovaný harmonogram	53
Obr. 39 – Číselníky Firem a Autorů	54
Obr. 40 – Přidání a editace doručovací firmy	54
Obr. 41 – Přidání a editace jména autora rubriky	54

SEZNAM PŘÍLOH

P I CD-ROM se zdrojovými kódy, obrázky, databází a dokumentací v elektronické podobě