

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A
INFORMATIKY

DIPLOMOVÁ PRÁCE

2010

Bc. Petr Bělohávek

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Realizácia webového informačného systém pre správu a optimalizáciu
finančných portfólií pomocou zvoleného štandardného riešenia
(frameworku)

Bc. Petr Bělohlávek

Diplomová práce

2010

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Petr BĚLOHLÁVEK**
Osobní číslo: **I08344**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Realizácia webového informačného systém pre správu a optimalizáciu finančných portfólií pomocou zvoleného štandardného riešenia (frameworku)**
Zadávací katedra: **Katedra softwarových technológií**

Z á s a d y p r o v y p r a c o v á n í :

V teoretickej časti bude uvedený prehľad existujúcich štandardných riešení pre tvorbu webových informačných systémov. Jedno z týchto riešení bude zvolené pre realizáciu daného informačného systému. Jedným z vyhodnocovaných štandardných riešení bude technológia Microsoft Office SharePoint Server 2007. Použitelnosť vybraného riešenia bude overená a vyhodnotená v praxi. Pomocou tohoto riešenia bude realizovaných niekoľko prípadov užitia systému pre správu a optimalizáciu finančných portfólií. Pri realizácii systému budú dodržiavané zásady vývoja informačných systémov podľa metodiky RUP.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

1. Dokumentácia Microsoft Office SharePoint, Zend, prípadne ďalších frameworkov
2. Arlow Jim, Neustadt IIa UML 2 a unifikovaný proces vývoje aplikaci

Vedoucí diplomové práce:

Ing. Jaroslav Lach

Katedra informatiky v dopravě

Datum zadání diplomové práce:

30. října 2009

Termín odevzdání diplomové práce:

21. května 2010



prof. Ing. Simeon Karamazov, Dr.

děkan

L.S.



doc. Ing. Antonín Kavička, Ph.D.

vedoucí katedry

V Pardubicích dne 10. listopadu 2009

Prohlášení autora

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Třebíči dne 18. 8. 2010

Bc. Petr Bělohávek

Touto cestou děkuji panu Ing. Jaroslavu Lachovi za vedení mé práce formou pravidelných konzultací, za náměty a cenné připomínky.

Anotace

Práce se zabývá hledáním z dostupných technologií právě té, která umožní realizaci webového informačního systému pro správu a optimalizaci finančních portfolií. V teoretické části jsem se zabýval popisem vybrané technologie Microsoft Office SharePoint Server a všemi dalšími použitými technologiemi v moji práci. Praktická část popisuje přípravné práce a práci s vybranou technologií a jejím rozšířením.

Klíčová slova

Microsoft Office Sharepoint Server; informační systém; optimalizace finančních programů

Title

Implementation of Web Information System for Management and Optimization of Financial Portfolios Using Selected Standard solution (Framework)

Annotation

This work is about searching for the right and available technology which allows the implementation of a web information system for managing and optimizing financial portfolios. In the theoretical part I dealt with a description of the technology of Microsoft Office SharePoint Server and all other technologies applied in my work. The practical part describes the preparatory work and working with selected technology and its enlargement.

Keywords

Microsoft Office Sharepoint Server; information system; optimalization of financial programs

Obsah

Úvod	10
1 Specifikace projektu	11
1.1 <i>Optimalizace finančních programů</i>	11
1.2 <i>Požadavky na aplikaci</i>	11
2 Úvod do technologie Microsoft SharePoint	12
2.1 <i>Microsoft Office SharePoint Server</i>	13
2.1.1 Single sign-on	13
2.1.2 In-Browser Rendering	14
2.1.3 Business Data Catalog	14
2.1.4 Indexované vyhledávání	15
2.2 <i>Windows SharePoint Services</i>	15
3 Přehled použitých technologií	17
3.1 <i>Virtualizace</i>	17
3.1.1 Virtualizační systém VMware Workstation	19
3.2 <i>Operační systém Windows Server 2003</i>	20
3.3 <i>Internet Information Services</i>	21
3.3.1 Funkce IIS	21
3.4 <i>Microsoft .NET Framework</i>	22
3.4.1 Architektura .NET Framework	23
3.5 <i>ASP.NET</i>	24
3.5.1 ASP.NET stránky	24
3.5.2 Rozšíření ASP.NET	25
3.6 <i>Microsoft SQL Server</i>	25
3.6.1 Verze Microsoft SQL Server	25
3.6.2 Microsoft SQL Server 2005	26
3.6.3 SQL Server Management Studio Express	26
4 Přípravné práce	28
4.1 <i>Případy užití</i>	28
4.2 <i>Způsob implementace</i>	28
4.3 <i>Vytvoření iteračního plánu</i>	29

4.4	<i>Instalace VMware Workstation</i>	30
4.4.1	Instalace pomocí grafického průvodce	30
4.4.2	Instalace pomocí příkazového řádku	31
4.5	<i>Vytvoření virtuálního stroje ve VMware Workstation</i>	33
4.6	<i>Instalace Microsoft Windows Server 2003 R2 na virtuální stroj</i>	34
4.7	<i>Aktualizace Microsoft Windows Server 2003 R2</i>	35
4.8	<i>Instalace dalších potřebných softwarových balíčků</i>	36
4.9	<i>Instalace Microsoft Office SharePoint Server 2007</i>	38
4.10	<i>Kontrola po nainstalování MOSS</i>	41
5	Řešení	44
5.1	<i>Vytvoření stránky</i>	44
5.2	<i>Vytvoření listů</i>	45
5.3	<i>Vytvoření sloupců</i>	47
5.3.1	Práce s jednotlivými datovými typy	48
5.4	<i>Přehled vytvářených listů a jejich sloupců</i>	54
5.5	<i>Pojistné matematické vzorce</i>	56
5.6	<i>Upravení vzorce Hodnota konec</i>	57
5.7	<i>Vložení nové položky</i>	59
5.8	<i>Ověření správnosti vzorce</i>	60
5.9	<i>Vytvoření a integrace Web Parts</i>	61
5.10	<i>Ladění Web Parts</i>	64
5.11	<i>Vlastní Web Part</i>	65
	Závěr	68
	Použitá literatura	70
	Přílohy	72
	<i>Příloha č. 1 – Obsah přiloženého CD</i>	72

Úvod

Před dvěma lety jsem zakončil bakalářské studium prací, na kterou nyní v této diplomové práci navazuji. V bakalářské práci jsem zhotovil desktopovou aplikaci, která splnila svůj účel, a to vytvoření postupu a následnou realizaci optimalizace finančních programů. Vzhledem k tomu, že se trend uživatelů v oblasti komunikace a podnikání přesunul směrem do prostředí Internetu, rozšiřování původního projektu by bylo velmi časově náročné a neefektivní.

Cílem této práce je analýza současného stavu již dostupných hotových řešení (frameworků, informačních systémů) a vyhledání nejefektivnějšího z nich, který je schopen implementovat veškerou funkčnost původního systému navrženého v mé bakalářské práci *Návrh a tvorba aplikace pro evidenci a optimalizaci clientských smluv* s dalšími, v současnosti důležitými parametry. Mezi ně patří technologie založená na webovém rozhraní s centrálním úložištěm dat, efektivní týmovou spoluprací finančních poradců a jejich manažerů, strukturovaně řazenými informacemi, přehledností a rychlým vyhledáváním potřebných údajů.

Díky rozsáhlosti projektu jsme se s vedoucím diplomové práce Ing. Jaroslavem Lachem a spolužákem Bc. Petrem Plavcem dohodli, že projekt budeme řešit společně. Petr Plavec se rozhodl pro analýzu stávajícího systému a nově zanalyzoval náš diplomový úkol. Já jsem si vzal na starosti jeho realizaci.

První kapitola obsahuje specifikaci projektu, požadavky na aplikaci a vysvětlení pojmu optimalizace finančních programů. V druhé kapitole je popsána námi vybraná technologie pro realizaci systému. Třetí kapitola se věnuje přehledu všech použitých technologií v této práci. Čtvrtá se zabývá přípravnými pracemi, mezi něž patří obecný popis systému, způsob implementace, vytvoření iteračního plánu a instalace virtuálního prostředí, operačního systému a použité technologie. Poslední, pátá kapitola, je ryze praktická, v níž pracuji s vybranou technologií. V ní realizuji požadavky stanovené ve specifikaci projektu. Jednotlivé kapitoly jsou podrobněji popsány ve vlastní práci.

1 Specifikace projektu

V mé bakalářské práci jsem se zabýval *optimalizací stávajících finančních programů*, které jsou i v této práci stěžejním bodem (více v kapitole 1.1). Základním požadavkem nově vytvářeného systému je prostředí obsahující klientskou část a část finančních programů (životních pojištění), rozdělenou na stávající a optimalizovanou. Vzhledem k tomu, že se většina podnikatelských aktivit obecně přesunula do prostředí Internetu, tato práce by měla řešit rozvoj původního desktopového systému na systém webového rozhraní.

1.1 Optimalizace finančních programů

„Optimalizací je označován proces, při kterém se analyzují finanční možnosti klienta a porovnávají se s navrženými programy tak, aby nabídka pro klienta byla co nejvýhodnější. Po této analýze dochází k vytvoření návrhu změn dosavadních finančních programů tak, aby klientovy potřeby a cíle byly naplněny.“ [1]

1.2 Požadavky na aplikaci

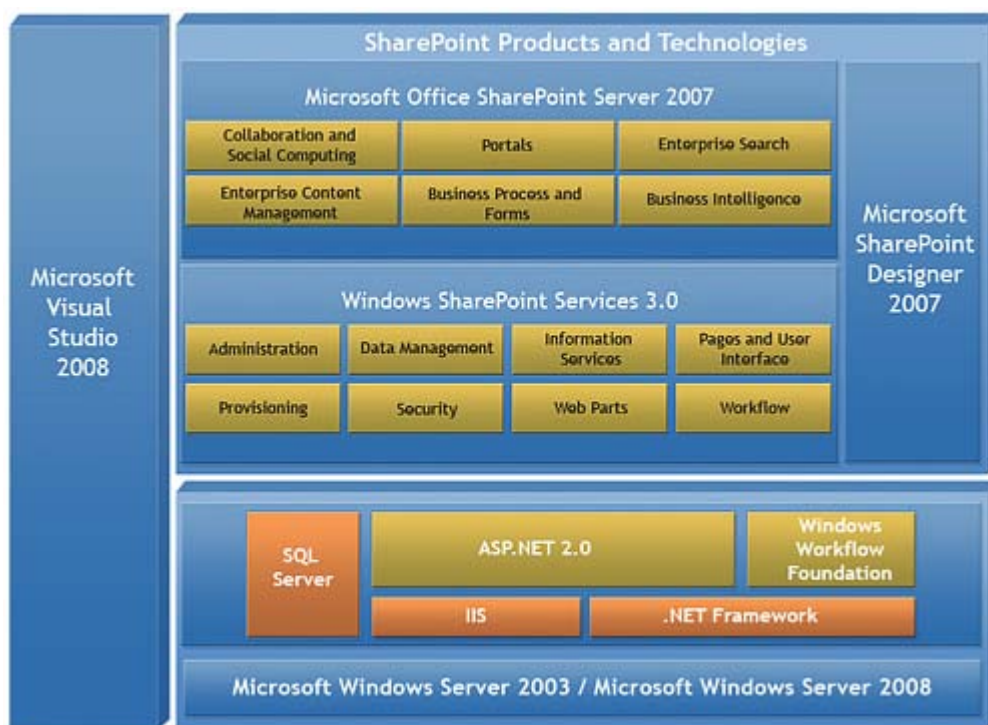
Jsou rozděleny do 3 kategorií:

- Funkční požadavky
 - optimalizace stávajících finančních programů (vytvoření finančního portfólia klienta)
 - vytvoření tiskového výstupu finančního portfólia s využitím statistických analýz a grafů
 - zprostředkování finančních portfólií v rámci týmu
- Systémové požadavky
 - webová aplikace přístupná z veřejného internetu
 - diferencovaný přístup finančních poradců
 - centrální uložení dat
- Obecné požadavky
 - zamezení opakovaného pořizování stejných dat

2 Úvod do technologie Microsoft SharePoint

V dnešním podnikatelském prostředí, kdy jsou distribuované pracovní síly, musí být členové týmu v užším kontaktu než kdy předtím. Efektivní spolupráce je stále důležitější, ale často je těžké ji dosáhnout. *Microsoft Windows SharePoint Services* řeší tento problém tím, že integroval základní prostředky týmových kolaborací a komunikačních technologií do jediného prostředí založeném na webovém rozhraní, které se snadno integruje s desktop aplikacemi, jako je například sada nástrojů Microsoft Office.

SharePoint (sdílený bod) je technologií, která se snaží centralizovat veškeré aktivity firmy do jednoho místa. To je jakýmsi průnikem informací všech firemních zdrojů. Pro snadnou orientaci a přehlednost jsou tyto informace strukturovaně zařazeny do jednotlivých kategorií. Díky tomu je zaručena velká přehlednost a rychlé vyhledávání potřebných údajů.



Obrázek 1: SharePoint produkty a technologie [2]

2.1 Microsoft Office SharePoint Server

Microsoft Office SharePoint Server (MOSS) je placeným produktem z rodiny produktů Microsoft SharePoint, který běží nad *Windows SharePoint Services* (WSS). MOSS je portálovým řešením pracujícím na webovém rozhraní. Hlavní předností je organizace a agregace informací v jednom centrálním on-line webovém prostoru, poskytujícím taxonomii firemních dat. MOSS úzce integruje aplikace v sadě Microsoft Office a přidává různé funkce, jako je např. hierarchická organizace obsahu oblastí, zvýšená navigace, zabezpečený přístup Single Sign-On, personalizace funkcí, indexované vyhledávání, vytváření vícejazyčných stránek, Business Data Catalog a in-browser rendering.

Obsahuje také *Microsoft Office SharePoint Server 2007 Excel Services*, což je technologie pro prohlížení, výpočet a extrahování hodnot ze sešitu aplikace Microsoft Excel prostřednictvím webového prohlížeče nebo webové služby. Další technologií je *Microsoft Office Forms Server 2007*, který umožňuje pracovníkům použití webového prohlížeče k interakci s navrženými šablonami formulářů Microsoft Office InfoPath 2007.

Klíčové funkce MOSS jsou uvedeny v následujících podkapitolách 2.1.1, 2.1.2, 2.1.3 a 2.1.4.

2.1.1 Single sign-on

Single sign-on systémy (SSO) se zaměřují na zvýšení zabezpečení přístupů do IT systémů. Běžně si musí uživatel pamatovat mnoho přístupových uživatelských jmen a k nim příslušných hesel. V závislosti na frekvenci používání dochází často k jejich zapomenutí a s tím jsou spojené požadavky na jejich reset, což je časově náročné pro helpdesk nebo administrátory systémů. Zároveň jsou často uživateli používána jednoduchá a snadno zapamatovatelná hesla, která mohou být snadno zneužita ke kompromitaci systému. Nejhorší možný případ je, když jsou hesla ukládána na jiná snadno zneužitelná místa než v hlavě uživatele.

SSO systémy řeší oba uvedené neduhy spojené s používáním hesel. Využívají silná generovaná hesla, která automaticky doplňují za uživatele. Ten získává komfort rychlého a bezpečného přihlášení bez nutnosti si pamatovat více hesel. Databáze

hesel je pak efektivně chráněna prostředky systému a je do ní umožněn bezpečný přístup pouze uživatele. Aby byl účinek ochrany maximální, je implementace SSO běžně spojována s implementací vícefaktorové autentizace (pro přihlášení je nutné mít přihlašovací předmět a k němu ještě znát heslo nebo PIN, podobně jako pro platební kartu). [3]

2.1.2 In-Browser Rendering

In-Browse Rendering přenechává veškerou činnost zobrazení stránky na prohlížeči koncového uživatele. Zde budou probíhat všechny s tím související operace a činnosti, generování grafických dat, která budou následně uživateli zobrazována.

Pokud se právě naznačený přístup důsledně dodrží a aplikuje na všechny součásti WWW stránek, má to jeden velmi příznivý efekt na fungování celé služby WWW – veškeré aktivity, spojené s vytvořením výsledné grafické podoby, budou svěřeny koncovému prohlížeči. [4]

2.1.3 Business Data Catalog

Business Data Catalog (BDC) umožňuje napojení SharePointu na externí data, která umí získat přímo z SQL databáze (nejenom MS SQL) přes webovou službu, ODBC¹ nebo Microsoft Access. Získaná data je možné následně použít:

- v seznamech,
- v rámci metadat v dokumentech,
- přímo zobrazit v rámci předpřipravených Web Part,
- vyhledávat v nich pomocí integrovaného vyhledávání; lze i zadávat nové údaje či měnit existující.

Připojení zdrojů spočívá ve vybrání tabulek, které se budou používat, spojení jejich primárních klíčů a nastavení filtrů, podle kterých bude uživatel moci vyhledávat. [5]

¹ Open Database Connectivity (ODBC) je standardizované softwarové API pro přístup k datábázovým systémům.

2.1.4 Indexované vyhledávání

Každý uživatel může sdílet informace určené i ostatním spolupracovníkům a ukládat je do konkrétní hierarchie, která je předem vytvořena. Čím více informací je ale v takovém systému uloženo, tím více vyvstává potřeba jejich prohledávání. Je potřeba zadat hledaný výraz a získat v co nejkratší době ten nejrelevantnější výsledek. Proto i MOSS má implementovaný silný a příjemný nástroj pro vyhledávání v informacích.

MOSS 2007 je možné použít pro podnikové vyhledávání, pro vyhledávání v rámci knihoven dokumentů a skupin uživatelů. Plně indexuje všechny dokumenty uložené v knihovně, kromě toho také indexy dat uložených v externích databázích, které jsou vystaveny prostřednictvím ADO.NET nebo webových služeb s dobře definovaným schématem WSDL².

SharePoint servery, webové stránky, sdílené soubory, veřejné složky a databáze mohou být nastaveny jako zdroje dat, které se následně indexují. Stejný indexovací systém se používá v systému Windows Desktop Search. Indexovací stroj používá specifická pravidla pro rozhodování, která určují, co má být indexováno. MOSS 2007 také obsahuje návrhová řešení v případě typografických chyb. [6]

2.2 Windows SharePoint Services

Windows SharePoint Services (WSS) je služba, která nabízí výkonnou sadu nástrojů pro uspořádání informací, řízení dokumentů, což zvyšuje efektivitu obchodních procesů a poskytuje prostředí pro týmovou spolupráci.

WSS pomáhá týmům zůstat ve spojení a produktivně poskytuje infrastrukturu umožňující snadný přístup k lidem, dokumentům a informacím. S WSS mohou týmy vytvářet webové stránky pro sdílení informací a rozvíjet spolupráci s ostatními uživateli. Umožňuje přístup k obsahu uloženého v rámci SharePoint webu pomocí webového prohlížeče a prostřednictvím desktopových aplikací.

² Web Services Description Language (WSDL) webová služba ve formě XML dokumentu, která popisuje rozhraní pro komunikaci s ostatními systémy.

WSS 3.0 ke svému fungování vyžaduje operační systém Windows Server 2003 a vyšší, ASP.NET 2.0, Windows Workflow Foundation a databázový server SQL Server 2000 a vyšší. [7]

3 Přehled použitých technologií

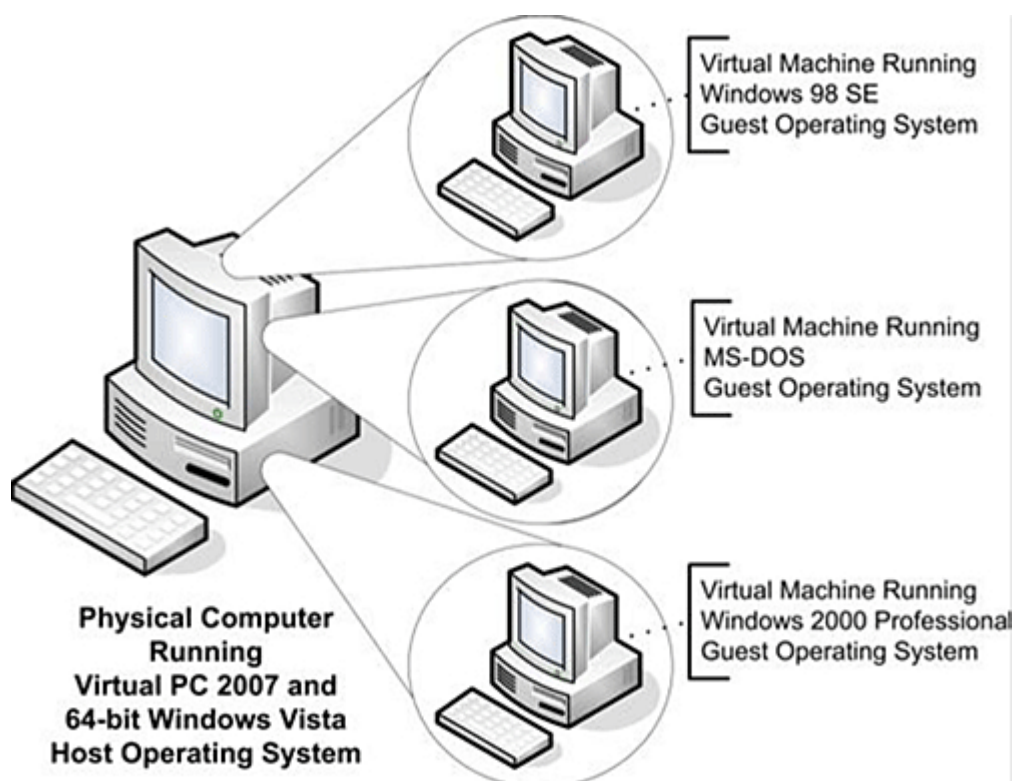
První technologií, kterou jsem použil, byl virtualizační software VMware Workstation. Díky němu jsem vytvořil virtuální stroj, který z mého počítače udělal hostitelský systém. Do tohoto stroje jsem následně nainstaloval operační systém Microsoft Windows Server 2003 R2, který se tak stal hostovaným systémem. Pro práci s MOSS jsem využíval další technologie (Internet Information Services, .NET Framework, ASP.NET, Microsoft SQL Server), které jsou níže popsány.

3.1 Virtualizace

Virtualizace umožňuje vytvoření virtuálního, zdánlivého počítače uvnitř skutečného počítače pomocí vhodného programového vybavení. Instalací virtualizačního softwaru se současný operační systém stane *hostitelem*.

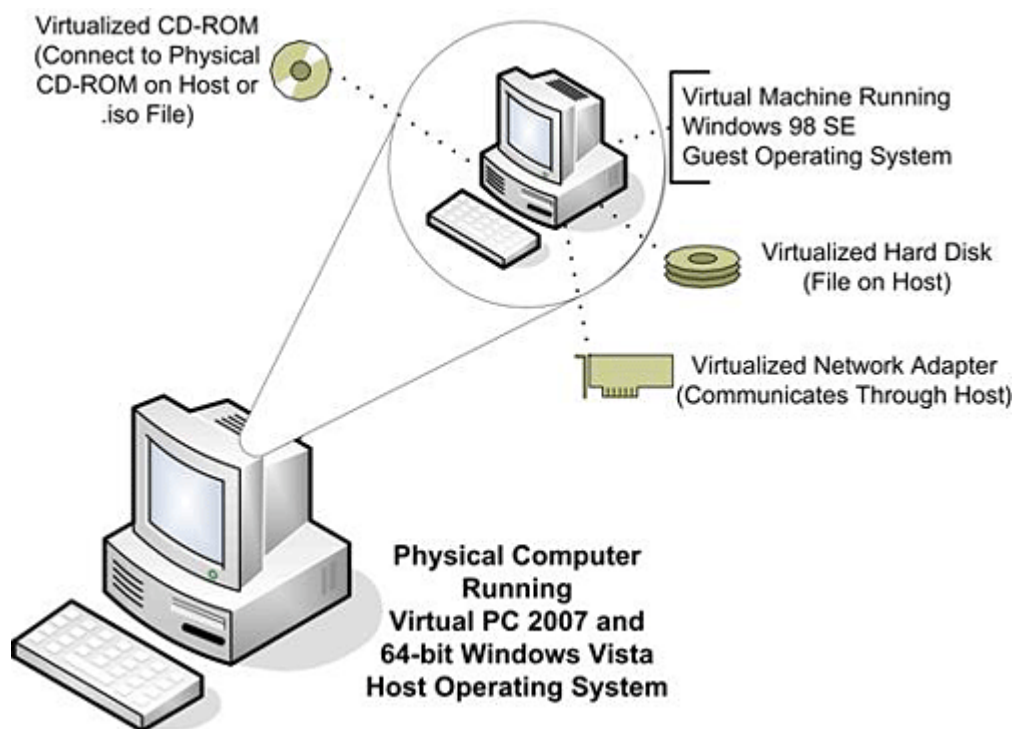
Hostitelský počítač dokáže simulovat dostatečné množství hardwaru tak, aby umožnil oddělený běh dalších operačních systémů. V rámci jednoho počítače je tedy možné vytvořit více virtuálních strojů, na kterých je možné spouštět zcela jiné operační systémy, přičemž vše běží v reálném čase a dle potřeby i souběžně.

Na Obrázku 2 v levé části je fyzický počítač (physical computer), na kterém běží 64bitový operační systém *Windows Vista*. Na tomto počítači je nainstalovaný virtualizační software *Virtual PC 2007*, díky kterému se stroj stává *hostitelem*. Ten může vytvářet libovolný počet virtuálních strojů; na obrázku v pravé části jsou vytvořeny 3 virtuální stroje s *hostovanými operačními systémy* Windows 98 SE, MS-DOS, Windows 2000 Professional.



Obrázek 2: Hostitelský počítač s 3 virtuálními stroji [8]

Hostované operační systémy jsou naprosto oddělené od *hostitele* (Obrázek 3). Každému virtuálnímu stroji je přidělena operační paměť, diskový prostor a další zařízení. Veškeré zpřístupnění jednotlivých zařízení lze běžně nastavit ve virtualizačních aplikacích. Hostované operační systémy žádným způsobem nemůžou ohrozit hostitelský systém nebo ostatní hostované systémy.



Obrázek 3: Přidělení zařízení hostovanému operačnímu systému [8]

Obrovskou výhodou virtualizace je možnost vytvoření *snapshotu* – uložení aktuálního stavu systému. Tato možnost se využívá v situacích, kdy je v případě problémů potřeba systém vrátit do předchozího stavu. Funkce je velmi podobná *Obnovení systému ve Windows*. [8]

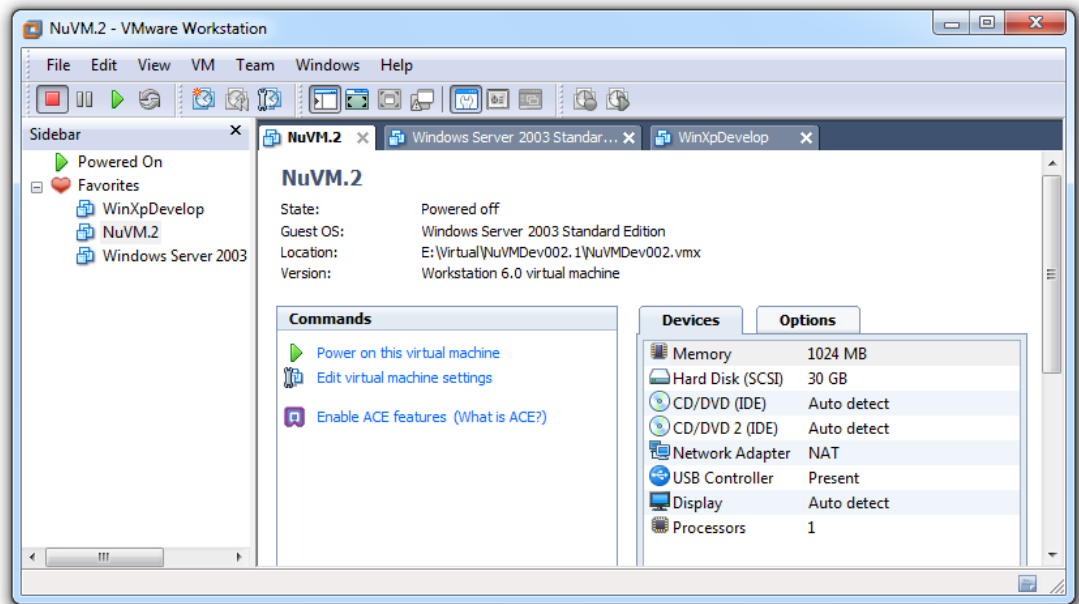
3.1.1 Virtualizační systém VMware Workstation

VMware Workstation (Obrázek 4) je virtualizační program společnosti *EMC Corporation*. Software umožňuje uživatelům na jednom počítači nastavit několik x86 a x86-64 virtuálních strojů a použít jeden nebo více z těchto virtuálních strojů současně s hostujícím operačním systémem. Na každém virtuálním stroji může být spuštěna instance hostovaného operačního systému.

Aktuální verze VMware Workstation 7.1 podporuje:

- 8 virtuálních procesorů (nebo 8 virtuálních jader) a 2 TB virtuální disky,
- podpora OpenGL 2.1 pro Windows Vista a Windows 7,
- výrazné zlepšení DirectX 9.0 grafického výkonu pro Windows Vista a Windows 7 (až 2x rychlejší než Workstation 7),

- optimalizovaný výkon rodiny procesorů Intel Core i3, i5, i7 pro rychlejší virtuální strojové šifrování a dešifrování,
- podpora více hostitelských a hostovaných operačních systémů Windows 2008 R2, Ubuntu 10.04, RHEL 5.4, a dalších: Fedora 12, Ubuntu 10.04, RHEL 5.4, SEL 11 SP1 a další. [9]



Obrázek 4: Virtualizační systém VMware Workstation

3.2 Operační systém Windows Server 2003

Windows Server 2003 je serverový operační systém od společnosti Microsoft³, který byl vydán v lednu roku 2003. Aktualizovaná verze *Windows Server 2003 R2* byla uvolněna v listopadu roku 2005. Tu jsem použil ve virtuálním stroji jako hostovaný operační systém.

Systém *Windows Server 2003 R2* rozšiřuje systém *Windows Server 2003* poskytováním nejefektivnějšího způsobu správy a řízení přístupu k místním a vzdáleným prostředkům a současně umožňuje snadnou integraci s existujícím prostředím systému *Windows Server 2003*. Systém *Windows Server 2003 R2* představuje zabezpečenou webovou platformu a podporuje nové možnosti využití, jako jsou zjednodušená správa pobočkových serverů, vylepšená správa identit a přístupů a efektivnější správa úložišť. [11]

³ <http://www.microsoft.com>

3.3 Internet Information Services

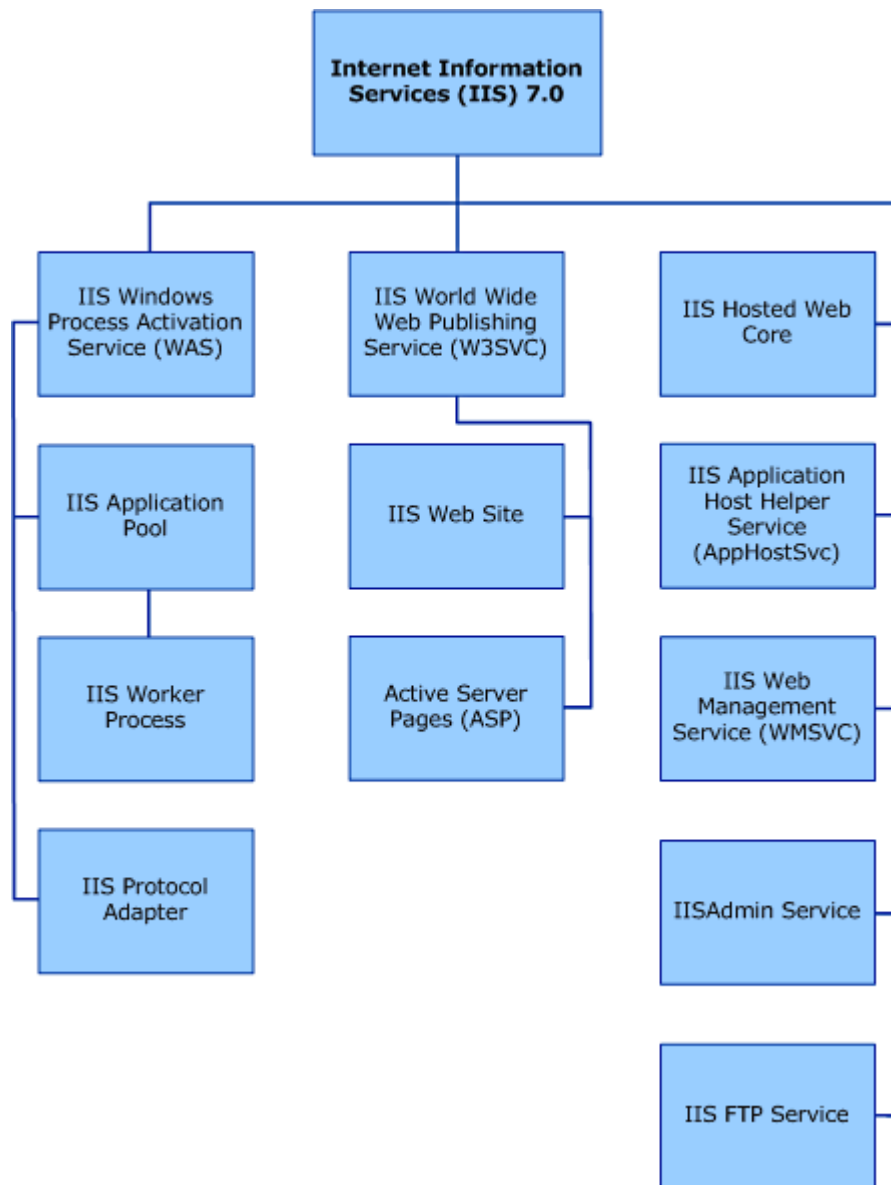
Internet Information Services (IIS) je webový server vytvořený společností Microsoft, který pracuje v operačních systémech Microsoft Windows. Je druhým nejvyužívanějším webovým serverem, hned za webovým serverem Apache HTTP Server. Mezi podporované protokoly v nejaktuálnějším IIS 7 patří: FTP, FTPS, SMTP, NNTP, HTTP a HTTPS (šifrovaný HTTP protokol).

3.3.1 Funkce IIS

Aktuální verze IIS 7 je postavena na modulární architektuře (Obrázek 5). Moduly mohou být přidány nebo odstraněny individuálně. IIS 7 obsahuje nativní moduly jako součást úplné instalace. Tyto moduly jsou jednotlivými funkcemi, které server používá pro zpracování žádosti a musí obsahovat následující:

- **HTTP moduly** – slouží k provádění specifických HTTP úkolů (žádost a zpracování), reakce na informace a dotazy zaslané v hlavičce klienta, vrací HTTP chyby a provádí přesměrování požadavků.
- **Bezpečnostní moduly** – používají se k provádění úkolů spojených s bezpečností při žádosti a zpracování, jako je například stanovení autentizačních modelů, provádění URL autorizace a filtrování požadavků.
- **Moduly obsahu** – slouží k provádění úkolů souvisejících s obsahem při žádosti a zpracování, jako je zpracování žádostí statických souborů, vrací výchozí stránku, pokud klient neuvádí zdroj v žádosti, a provádí výpis obsahu adresáře.
- **Kompresní moduly** – slouží k provádění úkolů souvisejících s kompresí při žádosti a zpracování, jako například komprese odpovědi, použití Gzip komprese u přenášených kódovaných odpovědí a provádění předkomprese statického obsahu.
- **Moduly s ukládáním do paměti** – slouží k provádění úkonů souvisejících s cache při žádosti a zpracování, jako je ukládání zpracovávaných informací v paměti na serveru a použití vyrovnávací paměti obsahu v následných žádostech u stejných zdrojů.

- **Moduly s logováním a diagnostikou** – slouží k provádění úkonů souvisejících s logováním a diagnostikou při žádosti a zpracování, jako je předávání informací a zpracování statusu HTTP.sys při logování, hlášení událostí a sledování požadavků v aktuální době vykonávaných procesů. [12]



Obrázek 5: Hierarchie IIS 7.0 [12]

3.4 Microsoft .NET Framework

Microsoft .NET Framework je softwarový framework, který může být instalován na počítačích s operačním systémem Microsoft Windows. Obsahuje velkou knihovnu naprogramovaných řešení a virtuální stroj, který řídí výkon programů, vytvořených speciálně pro tento framework. Zajišťuje vysoce produktivní prostředí, ve

kterém se může společně vyskytovat více programovacích jazyků pro snadnější vývoj různorodých počítačových aplikací.

Vývojová platforma *.NET Framework* je postavena na dvou základních pilířích:

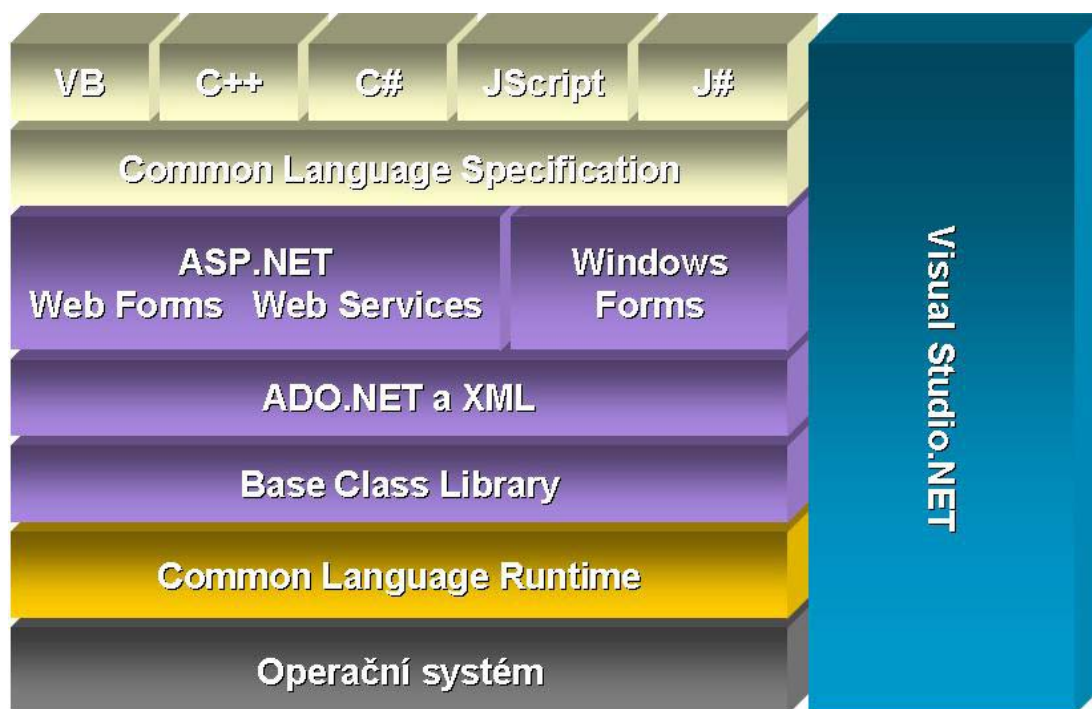
- společné běhové prostředí *Common Language Runtime (CLR)*,
- jednotná a hierarchicky uspořádaná knihovna tříd *Base Class Library (BCL)*.

CLR je exekučním prostředím pro běh *.NET Framework* aplikací. Poskytuje mnoho služeb, včetně zavedení a spuštění kódu, izolaci aplikační paměti, správu paměti, zpracování výjimek, přístup k metadatům (rozšířené typové informace) a konverzi mezikódu (*Microsoft Intermediate Language* neboli MSIL) na nativní kód platformy.

BCL zapouzdřuje třídy pro vývoj webových aplikací pomocí ASP.NET, inteligentních klientských aplikací pro Windows (pomocí Windows Forms) a databázových aplikací prostřednictvím subsystému ADO.NET. [13]

3.4.1 Architektura .NET Framework

Na nejnižší úrovni (Obrázek 6) se nachází *CLR*, realizující základní infrastrukturu, nad kterou je framework vybudován. Nad *CLR* se nachází několik hierarchicky umístěných knihoven. Ty jsou rozděleny do jmenných prostorů. Základem je knihovna nazvaná *BCL*. Nad ní je knihovna pro přístup k datům a práci s XML soubory. Poslední vrstvou je sada knihoven, usnadňující práci s uživatelským rozhraním. Ta je rozdělena do dvou skupin: pro usnadnění vytváření webových aplikací a pro vytváření klasických aplikací. Poslední vrstvu tvoří nelimitovaná množina programovacích jazyků. Jejich základní vlastnosti definuje *Common Language Specification (CLS)*. [14]



Obrázek 6: Architektura .NET Framework [14]

3.5 ASP.NET

ASP.NET je webový aplikační framework, uvedený na trh společností Microsoft, který umožňuje programátorům vytvářet dynamické webové stránky, webové aplikace a webové služby. První verze .NET Framework 1.0 byla vydána v lednu 2002 jako nástupce technologie Active Server Pages (ASP). *ASP.NET* je založen na Common Language Runtime (CLR), který umožňuje programátorům psát ASP.NET kód pomocí libovolného podporovaného .NET jazyka. Rozšířený framework *ASP.NET SOAP* umožňuje ASP.NET komponentám zpracování SOAP⁴ zpráv.

3.5.1 ASP.NET stránky

ASP.NET stránky jsou hlavním stavebním kamenem pro vývoj aplikací. Stránky jsou obsaženy v souborech s .aspx rozšířením; tyto soubory obsahují obvykle statické značení (X)HTML, stejně jako značení definované stranou serveru u webových a uživatelských ovládacích prvků. Zde vývojáři umísťují všechny požadované statické a dynamické obsahy webové stránky. Kromě toho mohou umístit do

⁴ Simple Object Access Protocol (SOAP) je protokolem pro výměnu zpráv založených na jazyku XML.

stránky v rámci bloku dynamický kód `<% -- dynamic code -- %>`, který běží na straně serveru.

3.5.2 Rozšíření ASP.NET

Microsoft vydal několik rozšířených frameworků, které se zapojí do ASP.NET, tím se rozšíří jeho funkčnost. V dnešní době jsou velmi uplatňované frameworky:

- ASP.NET AJAX – rozšíření jak na straně klienta, tak i na straně serveru pro psaní ASP.NET stránek, které obsahují AJAX funkčnost.
- ASP.NET MVC framework – rozšíření pomocí architektury MVC (model – view – controler).

3.6 Microsoft SQL Server

Microsoft SQL Server je relační databázový server vyvinutý společností Microsoft. Jeho primárními dotazovacími jazyky jsou T-SQL a ANSI SQL. Již léta patří k velmi populárním a výkonným databázovým serverům.

3.6.1 Verze Microsoft SQL Server

Jednotlivé verze jsou chronologicky seřazeny v následující tabulce:

Tabulka 1: Verze Microsoft SQL Serveru [15]

Verze	Rok	Název vydání	Název kódu
1.0	1989	SQL Server 1.0	
1.1	1991	SQL Server 1.1	
4.21	1993	SQL Server 4.21	SQLNT
6.0	1995	SQL Server 6.0	SQL95
6.5	1996	SQL Server 6.5	Hydra
7.0	1998	SQL Server 7.0	Sphinx
	1999	SQL Server 7.0 OLAP Tools	Plato
8.0	2000	SQL Server 2000	Shiloh
8.0	2003	SQL Server 2000 64 bit version	Liberty
9.0	2005	SQL Server 2005	Yuko
10.0	2008	SQL Server 2008	Katmai
10.5	2010	SQL Server R2	Kilimanjaro

3.6.2 Microsoft SQL Server 2005

Microsoft SQL Server 2005 je komplexní databázový systém s integrovanou správou dat. Umožňuje organizacím spolehlivě řídit informace a získat větší pohled na obchodní data a dosáhnout rychlejší a efektivní výsledky.

SQL Server 2005 vyšel v říjnu 2005 pod kódovým označením a je nástupcem SQL Server 2000. Kromě relačních dat je jeho součástí nativní podpora pro správu XML dat.

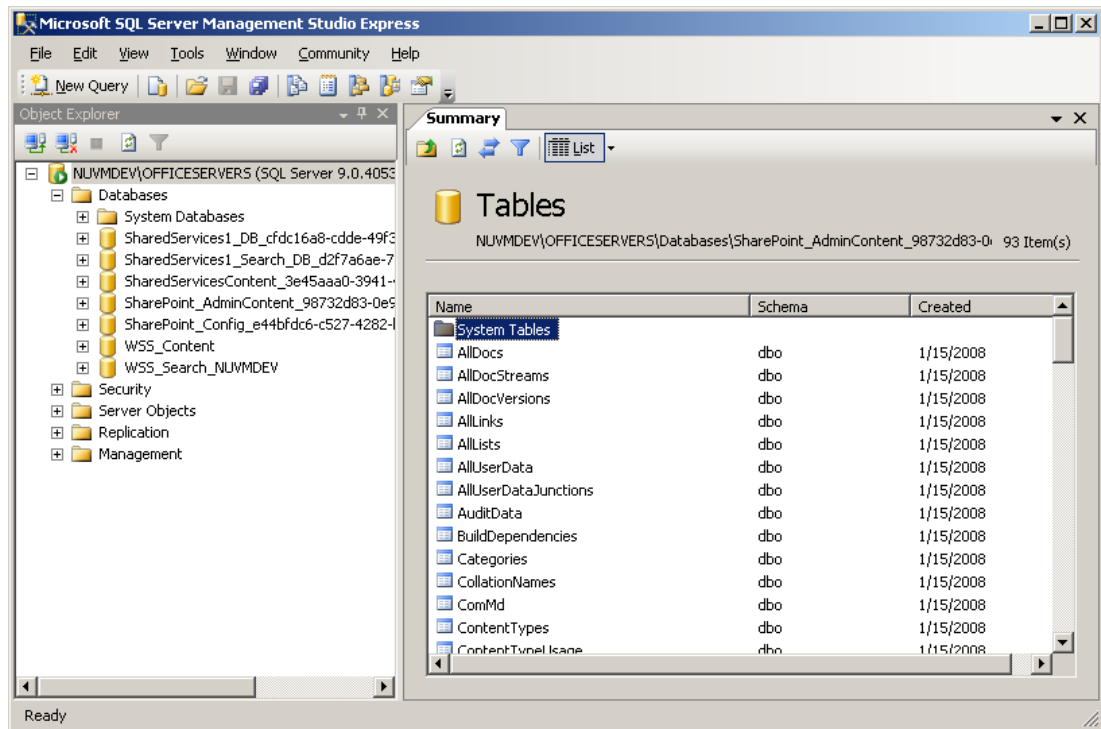
Mezi komponenty Microsoft SQL Serveru 2005 patří:

- databázový stroj,
- reportování,
- služby pro analýzu a integraci dat,
- replikace,
- hlášení,
- fulltextové vyhledávání,
- service Broker. [16]

3.6.3 SQL Server Management Studio Express

SQL Server Management Studio (Obrázek 7) je GUI nástroj pro konfiguraci, správu a administraci všech komponent v Microsoft SQL Server. Nástroj obsahuje editor skriptů a grafické nástroje, které pracují s objekty a funkcemi serveru. Verze SQL Server Management Studio je také k dispozici i pro SQL Server Express Edition, které je známé jako *SQL Server Management Studio Express* (SSMSE).

Hlavní funkcí SQL Server Management Studio je objekt Object Explorer, který umožňuje uživateli procházet, vybírat a spravovat některý z objektů v rámci databázového serveru. Lze analyzovat dotazy, optimalizovat výkon databáze, vytvářet nové databáze, měnit stávající databázová schémata, přidávat nebo upravovat tabulky a indexy nebo analyzovat výkon.



Obrázek 7: Microsoft SQL Server Management Studio Express

4 Přípravné práce

V této kapitole popisují všechny přípravné práce před samotnou realizací projektu. Ty jsme začali s Petrem Plavcem vytvořením modelu případů užití a jejich specifikacemi. Dále jsme vybrali framework pro implementaci našeho systému a vytvořili iterační plán pro dodržování stanovených cílů v daných termínech.

Praktickou část jsem zahájil instalací virtualizačního prostředí a vytvořením virtuálního stroje. Do něho jsem nainstaloval operační systém, který jsem aktualizoval. Poté jsem doinstaloval další potřebné softwarové balíčky. Posledním krokem přípravných prací byla instalace zvoleného produktu MOSS a následná kontrola jednotlivých nainstalovaných částí.

4.1 Případy užití

Pro přehlednost realizovaného systému jsme vytvořili model případů užití a jejich specifikaci. Ty vycházejí ze specifikace projektu (kapitola 1), původní funkčnosti aplikace v bakalářské práci, dodatečných požadavků a připomínek. Model a všechny vytvořené specifikace případů užití jsou uloženy v jednotlivých dokumentech na přiloženém CD (v příloze je popsáno umístění těchto dokumentů).

4.2 Způsob implementace

Důležitým bodem naší práce bylo nalezení frameworku nebo předpřipraveného řešení, pomocí kterého by bylo vhodné realizovat informační systém. U něj bylo nutné, aby splňoval veškeré předpoklady zadané ve specifikaci projektu (kapitola 1.2). Museli jsme tedy prozkoumat veškeré technologie, které jsou v současné době dostupné. Po jejich prozkoumání jsme vybírali z následujících frameworků a předpřipravených řešení:

- Kohana,
- ZEND,
- DotNetNuke,
- ASP.NET MVC,
- Microsoft Office SharePoint Server (MOSS).

Z těchto možností jsme vybrali poslední možnost, tedy MOSS, jako nejvhodnější variantu pro implementaci našeho systému. Podrobný popis jednotlivých technologií je přiložen na CD.

4.3 Vytvoření iteračního plánu

Následně jsme vytvořili iterační plán, který nám sloužil po celou dobu diplomové práce jako hlavní podklad pro splnění cílů jednotlivých iterací podle časového rozvržení.

Ten se skládá z iteračního plánu (Tabulka 2), který popisuje začátky a konce jednotlivých iterací. V druhé tabulce (Tabulka 3) jsou obsaženy iterace s jejich určitými cíli.

Tabulka 2: Iterační plán [17]

Iterace		Začátek	Konec
Zahájení	I1	28. 10. 2009	4. 1. 2010
Rozpracování	E1	5. 1. 2010	26. 2. 2010
	E2	1. 3. 2010	30. 4. 2010
	E3	3. 5. 2010	13. 5. 2010
	E4	13. 5. 2010	20. 5. 2010
Konstrukce	C1	21. 5. 2010	30. 6. 2010
	C2	1. 7. 2010	19. 8. 2010
Předání	T1	20. 8. 2010	

Tabulka 3: Cíle iterací [17]

Iterace	Cíl
I1	Vize, iterační plán, seznam rizik, vyhodnocení
E1	Nainstalování a nastavení serveru, vyzkoušení použitelnosti navrženého frameworku
E2	Případy užití, analytický model s vlastními stereotypy
E3	Vytvoření modelu architektury MOSS
E4	Výpis listů jen pomocí MOSS
C1	Uživatelské účty, skupiny, finanční programy
C2	Vytvoření vlastní Web Part pro výpis finančních portfolií
T1	Sepsání

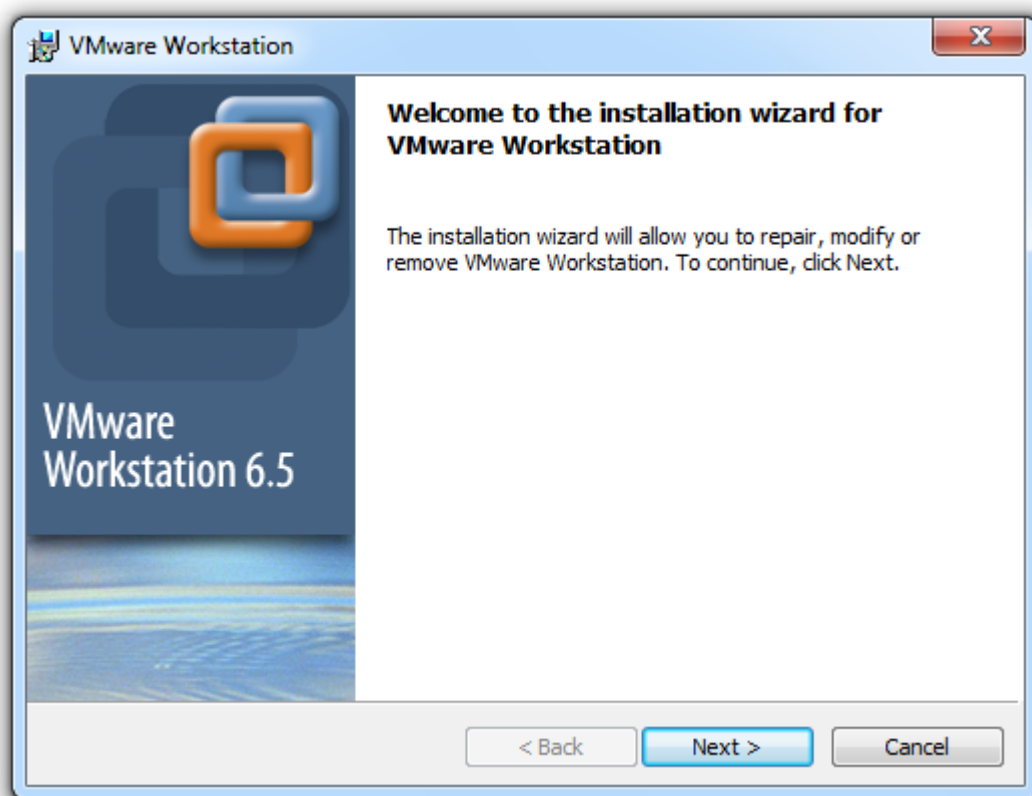
4.4 Instalace VMware Workstation

Pro práci s operačním systémem a zvolenou technologií jsem vytvořil virtuální stroj pomocí virtualizačního programu *VMware Workstation* ve verzi 6.5.3 (Build 185404).

Instalace VMware Workstation může být provedena jak pomocí grafického průvodce, tak i pomocí příkazového řádku.

4.4.1 Instalace pomocí grafického průvodce

Instalace s grafickým průvodcem (Obrázek 8) je velmi snadná. Uživatel si zvolí umístění, kam bude VMware Workstation nainstalován. Dále si vybere, jaké přídatné moduly budou k vizualizačnímu nástroji nainstalovány.



Obrázek 8: Instalace VMware Workstation pomocí grafického průvodce

4.4.2 Instalace pomocí příkazového řádku

Tato možnost je vhodná například ve velkém podniku – při instalaci několika hostitelských počítačů se systémem Windows s použitím funkce *Microsoft Windows Installer* (MSI), bez reagování na dotazy průvodce.

Před instalací je potřeba ověřit, zda je nainstalován MSI ve verzi 2.0 nebo vyšší. Tato verze instalačního systému je k dispozici od verze Windows XP.

Průběh instalace je v následujících krocích:

1. Otevření příkazového řádku a vložení příkazu pro administrativní instalaci obrazu VMware Workstation:

```
setup.exe /s /e <install_temp_path>
```

setup.exe je název instalátoru na instalačním médiu. Stažený instalátor má název podobný tvaru `VMwareWorkstation-<xxxx>.exe`, kde `<xxxx>` je série čísel, která představují verzi a build.

<install_temp_path> je úplná cesta ke složce, kde se nachází instalační obraz.

2. Spuštění bezobslužné instalace pomocí příkazu `msiexec`, který extrahuje administrativní instalační obraz zadaný v předchozím kroku:

```
msiexec -i "<install_temp_path>\VMware Workstation.msi"  
[INSTALLDIR="<path_to_program_directory>"] ADDLOCAL=ALL  
[REMOVE=<feature_name,feature_name>] /qn
```

Při instalaci Workstation na jiné místo než je výchozí, je třeba určit cestu, která je uvedena v klauzuli `INSTALLDIR`.

Pomocí volitelného `REMOVE = <property>` lze přeskočit instalaci některých funkcí, které jsou uvedeny v tabulce.

Tabulka 4: Hodnoty pro odebrání vlastností VMware Workstation

Hodnota	Popis
Authd	VMware autorizační služba, která se používá pro plnění úkolů, když Workstation neběží pod administrátorským účtem.

Network	Síťové prvky včetně virtuálního mostu a hostitelských adaptérů (pouze pro hostitelské a NAT sítě). Pokud chcete používat NAT a DHCP, neodstraňujte tuto složku.
DHCP	Virtuální DHCP server.
NAT	Virtuální NAT zařízení.

Při zadání více jak jedné hodnoty, je potřeba jejich oddělení, jako například REMOVE = Authd, NAT.

Při zadání REMOVE = Network instalace přeskočí některé síťové komponenty včetně NAT a DHCP. Není tedy třeba specifikovat samostatně DHCP nebo NAT.

Dále lze upravovat instalaci ve formátu <property>=<value> přidáním některé z instalačních vlastností uvedených v tabulce. Při použití vlastnosti sériového čísla je třeba zadávat toto sériové číslo s pomlčkami (xxxxx-xxxxx-xxxxx-xxxxx). [18]

Tabulka 5: Vlastnosti hodnot

Vlastnost	Výsledek vlastnosti	Default
DESKTOP_SHORTCUT	Instalace zástupce na ploše.	1
DISABLE_AUTORUN	Zakázání Autorun CD na hostovaném počítači.	1
REMOVE_LICENSE	Odstranění všech uložených licencí (pouze při odinstalování).	0
SERIALNUMBER	Vložení sériového čísla.	

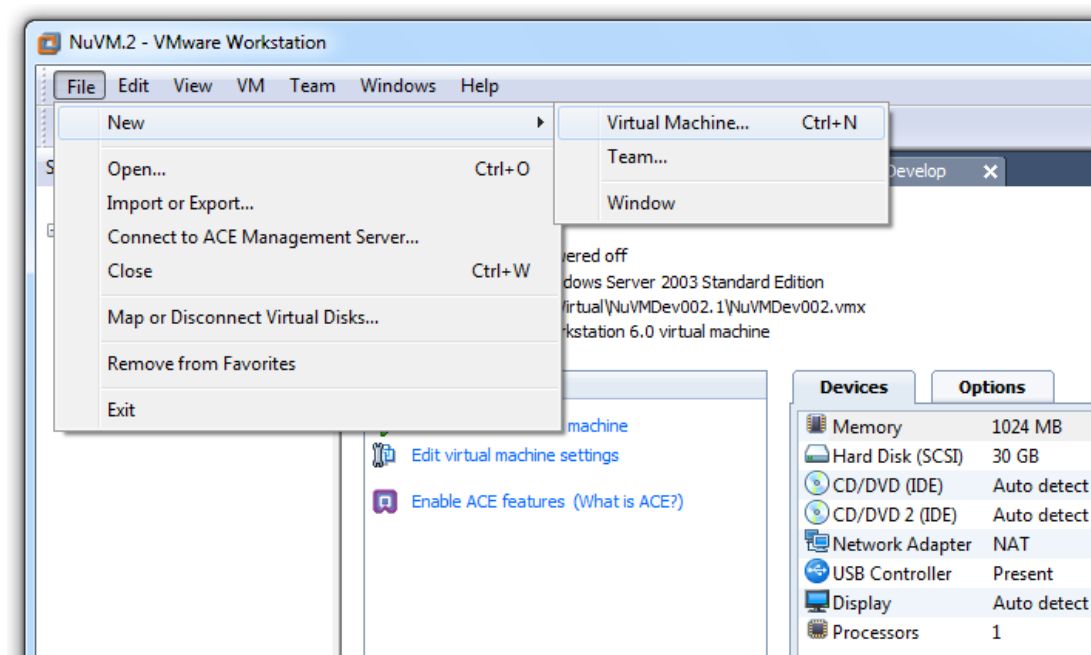
4.5 Vytvoření virtuálního stroje ve VMware Workstation

Po nainstalování virtualizačního programu se počítač stal hostitelským systémem. Pro práci s hostovaným operačním systémem jsem vytvořil nový virtuální stroj.

Rychlé vytvoření virtuálního stroje a spuštění instalace hostovaného operačního systému je realizováno pomocí následujících kroků:

1. Pokud nebude operační systém instalován z ISO obrazu, je potřeba vložit do mechaniky instalační CD nebo DVD s patřičným operačním systémem.
2. Spuštění VMware Workstation.
3. Vybrat **File** → **New** → **Virtual Machine** (Obrázek 9).
4. Na úvodní stránce zvolit **Typical** (standardní vytvoření virtuálního stroje) a zvolit tlačítko **Next**.
5. Na stránce „Guest Operating System Installation“ vybrat instalační disk nebo instalační disk image souborů (iso), zvolit tlačítko **Další**.
6. Na stránce „Easy Install Information“ vyplnit potřebné informace.
Zadání hesla je pro Windows volitelné a pro Linux potřebné. Na všech hostovaných operačních systémech Windows (kromě Windows 2000) je zadané heslo použito jako účet s administrátorským oprávněním. Na Windows 2000 je heslo použito jako administrátorský účet.
7. Zbytek stránek průvodce lze ponechat tak, jak je defaultně nastaveno.
Virtuální počítač je vytvořen, jeho jméno je přidáno do seznamu oblíbených položek.

Následně se objeví pohled konzole vytvářeného virtuálního stroje. Detekuje se zařízení (mechanika nebo ISO obraz) a po ní začíná instalace operačního systému. Tato instalace je popsána v následujícím odstavci.



Obrázek 9: Vytvoření nového virtuálního stroje

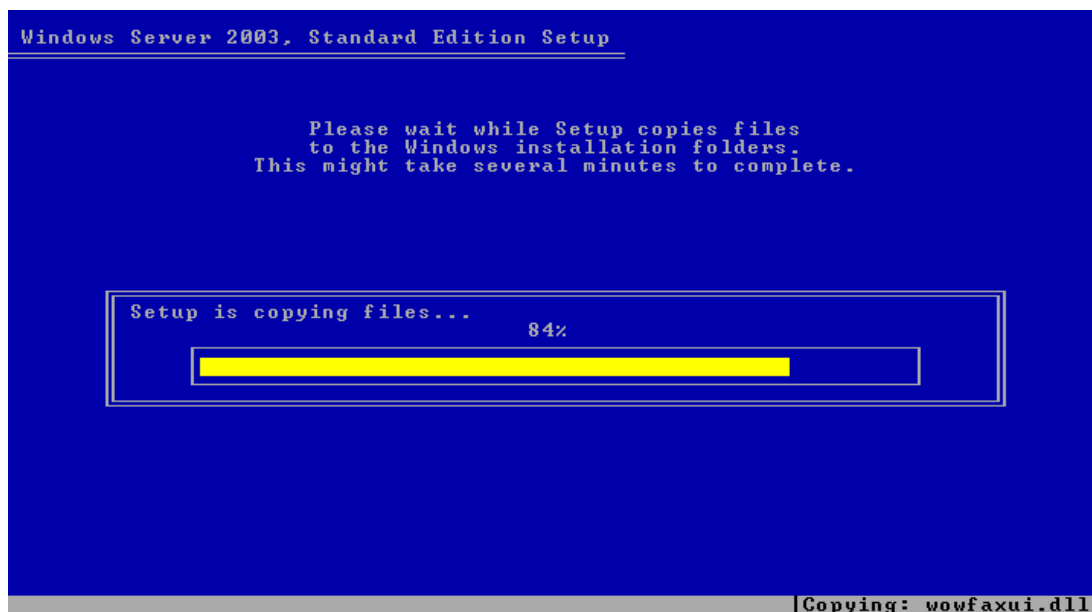
4.6 Instalace Microsoft Windows Server 2003 R2 na virtuální stroj

Na hostitelském počítači jsem pomocí nainstalovaného virtualizačního nástroje *VMware Workstation 6.5* začal instalovat na vytvořeném virtuálním stroji hostovaný operační systém *Microsoft Windows Server 2003 R2 Standard Edition*.

Systém *Windows Server 2003 R2* obsahuje dva disky produktu. Disk 1 obsahuje nejnovější verzi systému *Windows Server 2003*. Disk 2 obsahuje instalační soubory volitelných součástí systému *Windows Server 2003 R2*.

Při instalaci Disku 2 nebudou automaticky nainstalovány volitelné součásti systému *Windows Server 2003 R2*. Po dokončení instalace lze volitelné součásti nainstalovat pomocí funkce *Přidat nebo odebrat součásti systému* v *Ovládacích panelech*. Lze také přidat nebo upgradovat role serveru pomocí nástroje *Správa serveru*.

Tato instalace má stejný průběh jako instalace jakéhokoliv operačního systému na fyzický počítač. Průběh instalace jsem zaznamenal pomocí zachycení snímků obrazovky (Obrázek 10). Po dokončení instalace je automaticky nainstalován *VMware Tools*.

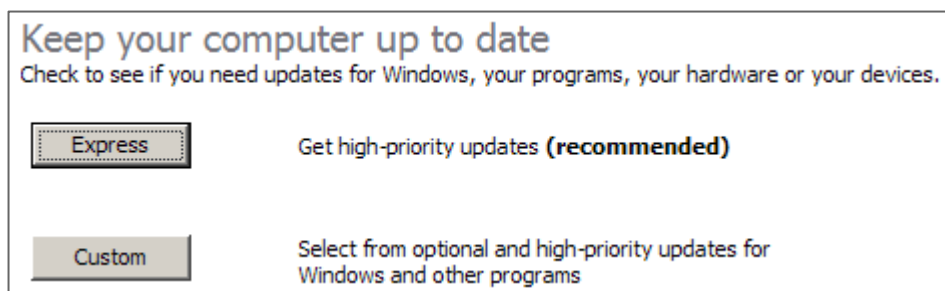


Obrázek 10: Instalace Microsoft Windows Server 2003 R2 na vytvořeném virtuálním stroji

4.7 Aktualizace Microsoft Windows Server 2003 R2

Po nainstalování operačního systému *Microsoft Windows Server 2003 R2* jsem aktualizoval software pomocí online služby Windows Update:

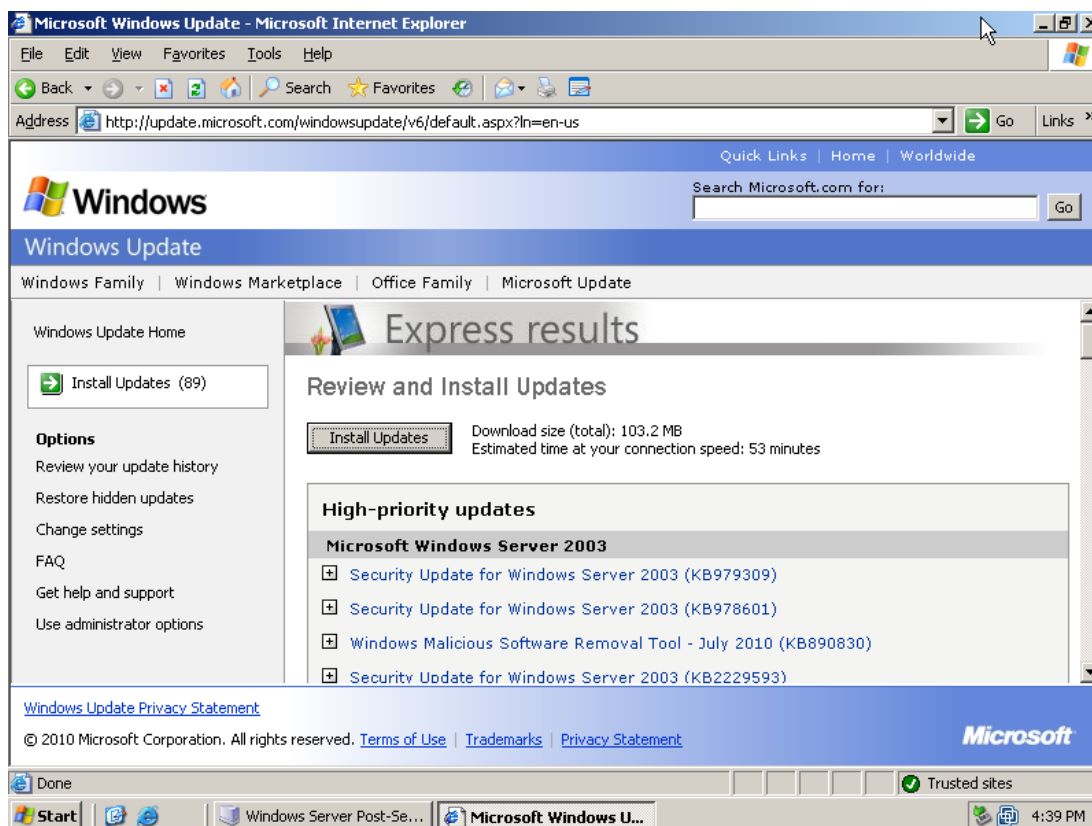
1. Položka **Start** → **Windows Update** (počítač musí být připojen k síti Internet).
2. Na výběr jsou 2 možnosti aktualizace: **Express** (expresní) a **Custom** (volitelná). Pro vyhledání aktualizací jsem využil doporučenou expresní instalaci (Obrázek 11).



Obrázek 11: Možnosti aktualizace systému Windows

3. Služba Windows Update prohledá počítač a zobrazí seznam doporučených aktualizací (jedná se pouze o některé produkty společnosti Microsoft).

4. Po nalezení všech doporučených aktualizací jsem zvolil odkaz **Install Updates** (Obrázek 12).



Obrázek 12: Instalace doporučených aktualizací online služby Windows Update

Aktualizovaný počítač je nutné udržovat i po stránce bezpečnostní. Aktualizacemi se ošetřují důležité bezpečnostní chyby operačního systému, nové a stále hrozby potenciálních útoků a celkový bezproblémový chod stanice.

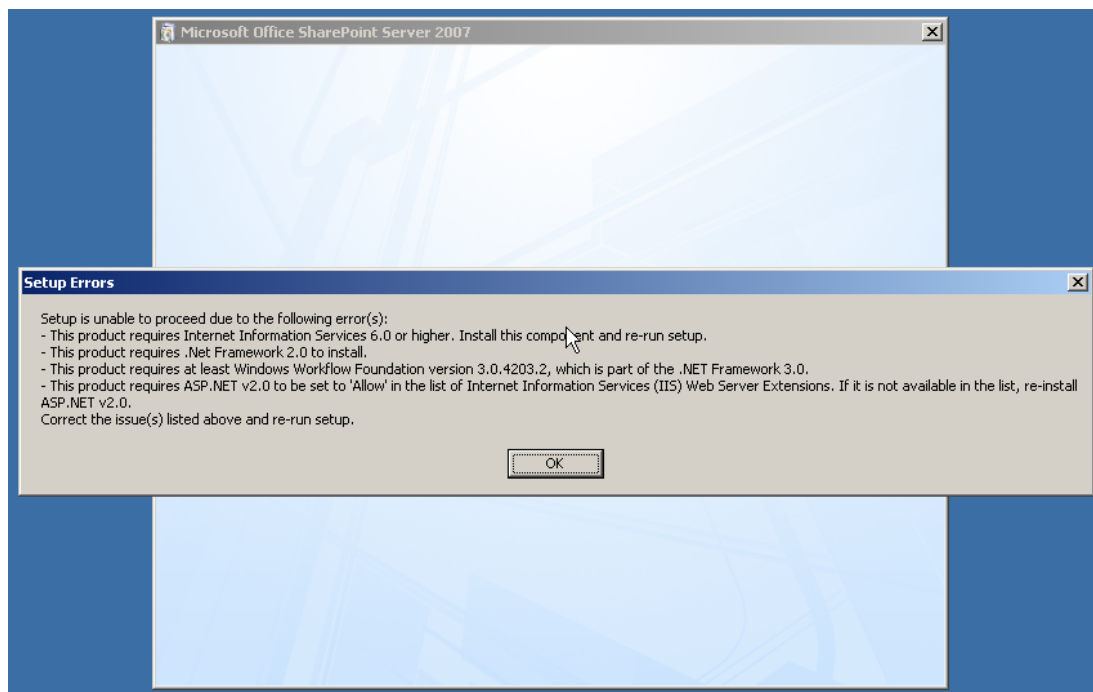
4.8 Instalace dalších potřebných softwarových balíčků

Po nainstalování operačního systému a jeho zaktualizování jsem přistoupil ke stažení a následné instalaci *Microsoft Office SharePoint Server 2007* (MOSS 2007). Ve VMware Workstation jsem si připojil stažený instalační ISO obraz v následujících krocích:

1. Vybrat VM → *Removable Devices* → zvolit aktivní mechaniku; v mém případě *CD/DVD (IDE) (Auto Detect)* → *Settings...*
2. Zde vybrat instalační ISO obraz (*Use ISO image file*), uložený na disku hostitelské stanice.

3. ISO obraz je připojen, nyní lze spustit instalaci MOSS 2007.

Při spuštění instalace MOSS 2007 se objevilo dialogové okno s chybami (Obrázek 13), které hlásily doinstalování dalších potřebných softwarových balíčků.



Obrázek 13: Dialogové okno s oznámením nutnosti doinstalování dalších potřebných softwarových balíčků

V závislosti na roli konkrétního serveru je tedy předpokladem instalace MOSS 2007 přítomnost dalších softwarových balíčků. Zatímco všechny vyžadují několik základních nástrojů (jako je např. .NET Framework), může být přítomnost softwarových balíčků určována i typem operačního systému a jeho možnostmi povolení.

Při vývoji informačního systému je potřeba mít k dispozici prostředí umožňující testování Master Pages, Page Layouts a interakci s SharePoint objektovým modelem, ladění vytvořených skriptů. To vše se neobejde bez nainstalování následujících softwarových balíčků:

- Microsoft Windows Server 2003 s alespoň Service Packem 1 (vyjma Web Edition),
- Windows SharePoint Services V3 nebo Microsoft Office SharePoint Server 2007,
- požadovaný souborový systém NTFS,

- .NET 2.0 Framework a Software Development Kit (SDK), .NET 3.0 Framework a volitelně SDK,
- Internet Information Services 6.0 nebo vyšší,
- Visual Studio 2005 nebo vyšší,
- SharePoint Designer (doporučeno),
- Microsoft Office (alespoň Word, Excel a Outlook).

Dále jsem v mém případě doinstaloval softwarové balíčky požadované operačním systémem:

- Windows Imaging Component,
- Windows Work Flow Foundation 3.0.4203.2.

4.9 Instalace Microsoft Office SharePoint Server 2007

Po doinstalování veškerých potřebných softwarových balíčků jsem mohl pokračovat s instalací MOSS 2007. Musel jsem opět ověřit, zda je instalační ISO obraz připojen v hostovaném operačním systému.

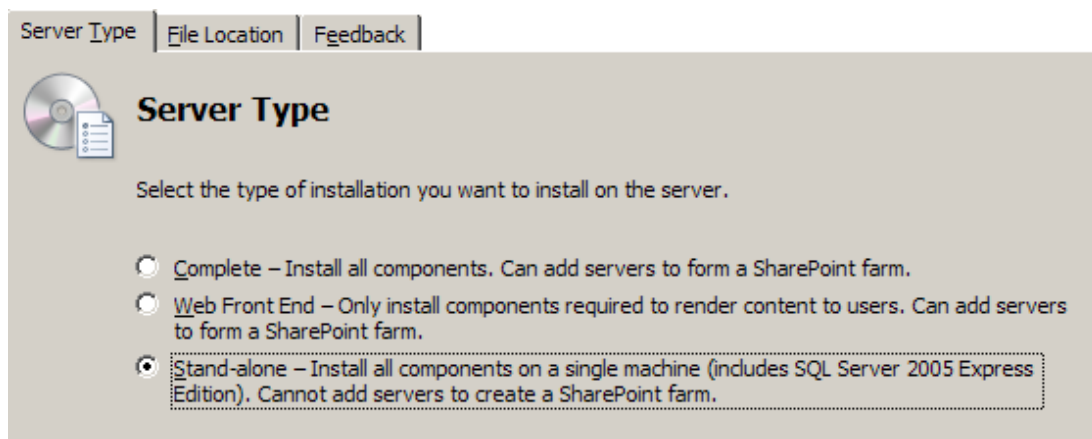
V této fázi probíhá instalace v následujících krocích:

1. Otevření virtuální mechaniky, na které je připojen ISO obraz, kde se následně spustí instalace MOSS 2007, nebo druhým způsobem je spuštění souboru *setup.exe* ze složky x64 (pro 64bitové procesory) nebo x86 (pro 32bitové procesory). Po spuštění instalačního programu bude opět provedena kontrola, zda jsou nainstalovány veškeré potřebné komponenty. Pokud se objeví nějaké chyby, zobrazí se upozornění s jejich výpisem.
2. Zadání produktového licenčního čísla. Při úspěšném zadání se zobrazí platnost ve formě „zeleného zaškrtnutí“ a povolení tlačítka *Continue* (Pokračovat).
3. Potvrzení licenční smlouvy a následné potvrzení tlačítkem *Continue*.
4. V okně „Choose the installation you want“ jsou 2 možnosti – *Basic* (Základní) nebo *Advanced* (Upřesnit). Každá možnost má specifickou instalaci, proto je třeba vědět, co která znamená. V mém případě jsem zvolil variantu *Advanced*.

Varianta *Basic* – MOSS bude instalována na jediném serveru a později nebude možno přidávat další servery nebo vytvářet farmy.

Varianta *Advanced* – produkt bude instalován buď na jediném serveru, nebo v serverové farmě.

5. Při výběru možnosti *Advanced* se zobrazí následující obrazovka, ve které jsou tři záložky (Obrázek 14): *Server Type*, *File Location*, *Feedback*.



Obrázek 14: Obrazovka po výběru možnosti *Advanced*

V záložce *File Location* lze zvolit místo, kam bude nainstalován MOSS 2007.

V záložce *Feedback* se lze stát součástí Microsoft Feedback Program.

Podstatná je záložka *Server Type*, ve které jsou k dispozici tři různé možnosti instalace:

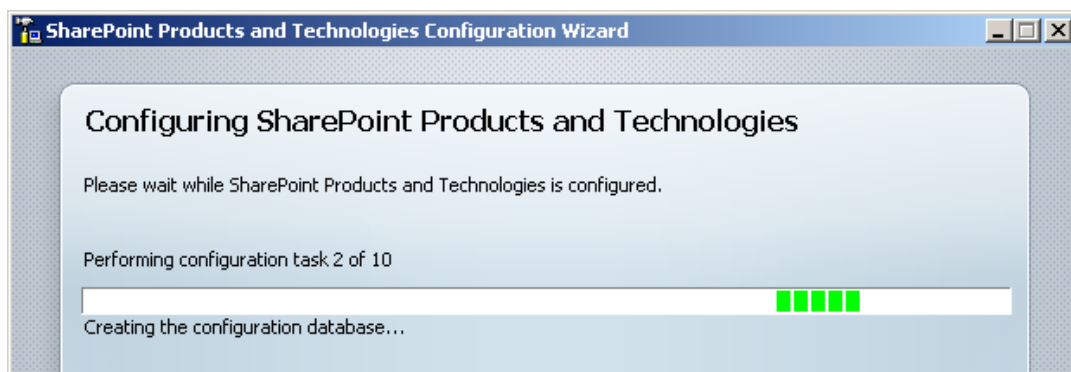
- *Complete* – instalace všech SharePoint binárních souborů na server.
- *Web Front End* – instalace pouze služeb nezbytných pro práci s webovými službami a funkcemi.
- *Stand-alone* – volba je stejná jako u varianty *Basic*, která byla v předchozí obrazovce.

U možnosti *Web Front End* a *Complete* je vyžadován výběr SQL Serveru, který bude jako back-end datovým prvkem SharePoint Serveru.

V mé práci jsem nepotřeboval přidávat další servery nebo vytvářet farmy, proto jsem zvolil možnost instalace typu *Stand-alone*.

6. Zobrazí se okno, kde je nutno potvrdit spuštění průvodce (Run the SharePoint Products and Technologies Configuration now) a následně pokračovat tlačítkem *Close*.

7. Pokračování potvrdit tlačítkem *Next*. Pokud jsou zastavené služby během konfigurace, zobrazí se výstražná zpráva.
8. V poslední fázi pokračovat tlačítkem *Yes*, kterým konfigurační aplikace spustí instalační proces (Obrázek 15), v němž jsou zobrazovány jednotlivé vykonávané kroky.

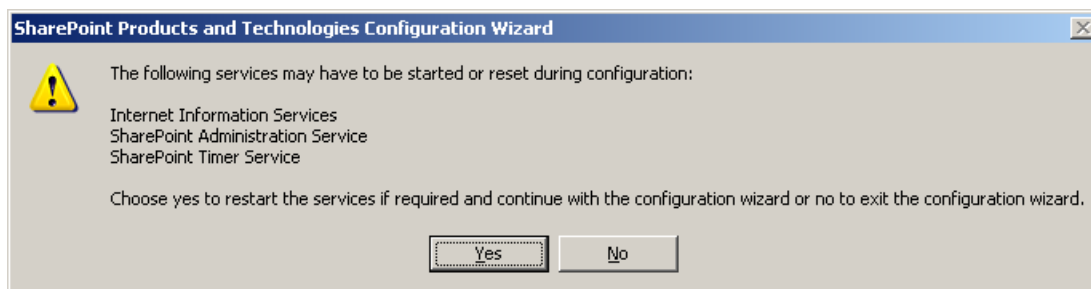


Obrázek 15: Instalační proces se zobrazením jednotlivých vykonávaných kroků

Během instalace se provedou následující kroky:

- a) **instalace SQL a inicializace nastavení** – instalace SQL Express a konfiguračních souborů
- b) **vytvoření konfigurační databáze** – vytváří SharePoint konfigurační databázi v SQL Expressu a instaluje schéma
- c) **instalace nápovědy** – instaluje zkompilevané soubory nápovědy
- d) **instalace zdrojů** – instalace assemblies, šablon a obrázků
- e) **registrace SharePoint služeb** – instalace WSS webových služeb
- f) **registrace SharePoint funkcí** – registrování assemblies pro funkce
- g) **zajištění a vytvoření centrální administrační stránky (Central Administration Site)** – vytváří nový virtuální adresář v IIS, vytváří application pool a na tomto místě instaluje SharePoint *Central Administration Site*
- h) **vytvoření vzorových dat**
- i) **instaluje Application Content Files** – ukládá šablony na server, vytváří nový virtuální adresář, novou standardní webovou stránku a nejvyšší úroveň stránky
- j) **dokončení** – dokončuje kompletní instalaci a restartování služeb

Po dokončení se zobrazí okno s úspěšně vykonanou instalací a výběrem restartování služeb (Obrázek 16).



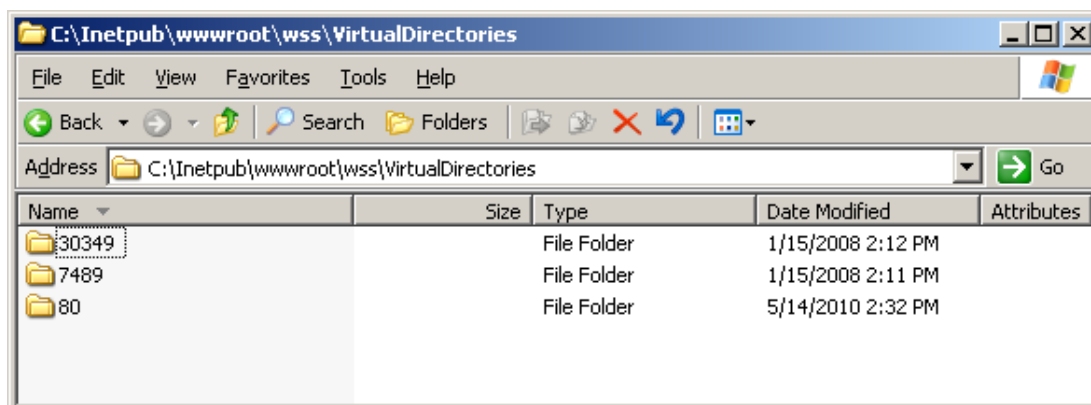
Obrázek 16: Restartování služeb po úspěšně vykonané instalaci

Nyní je vytvořena databáze, nainstalována centrální administrace (Central Administration) a vytvořena výchozí stránka na portu 80. Po klepnutí na tlačítko *Finish* (Dokončit) bude v top-úrovni stránky otevřen v prohlížeči nainstalovaný Microsoft Office SharePoint Server.

Při prvním načítání stránek je běžné vidět chybovou stránku. Stačí kliknout na odkaz *Zpět* a opakovat načtení stránek. Pokud ani po několika pokusech nedojde k úspěšnému spuštění MOSS, musí se restartovat stanice.

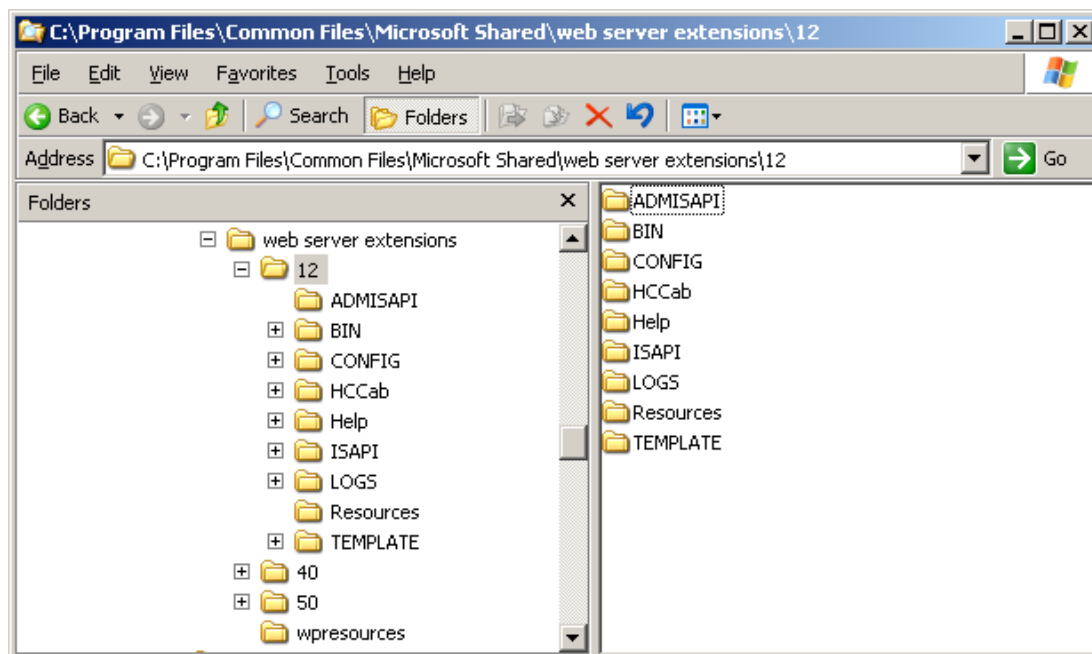
4.10 Kontrola po nainstalování MOSS

Pro správu Central Administration a výchozí top-úrovně stránky jsou vytvořeny adresáře ve složce *C:\Inetpub\wwwroot*, kde má adresář novou cestu *WSS\VirtualDirectories* (Obrázek 17).



Obrázek 17: Virtuální adresář WSS

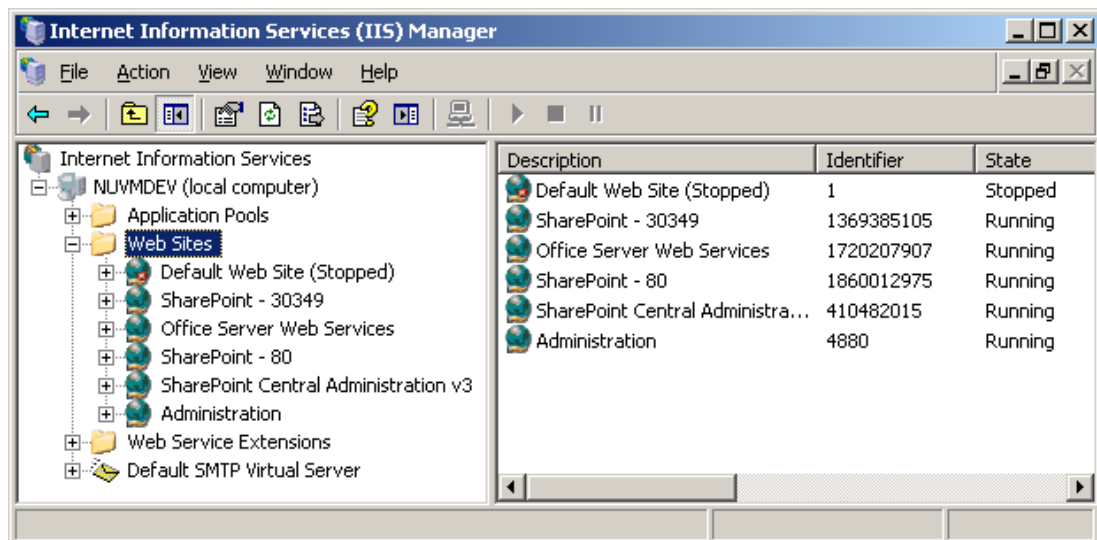
SharePoint soubory, DLL knihovny, šablony, XML soubory a další, jsou instalovány v rámci SharePoint složky, která je umístěna (ve výchozím nastavení) *C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions* (Obrázek 18).



Obrázek 18: Hlavní SharePoint složka

Pod složkou *C:\Program Files\Common Files\Microsoft Shared\web server extensions\12* jsou veškeré definice, obrázky, vlastnosti složky, SharePoint administrativní nástroje a další.

Při otevření IIS manažera přes *Start → Programs → Administrative Tools → Internet Information Services (IIS) Manager* (Obrázek 19) je výchozí webový server (Default Web Site) zastaven a nově nainstalované webové stránky jsou spuštěny (SharePoint – 30349, Office Server Web Services, SharePoint – 80, SharePoint Central Administration, Administration).



Obrázek 19: IIS a jeho spuštěné webové stránky

5 Řešení

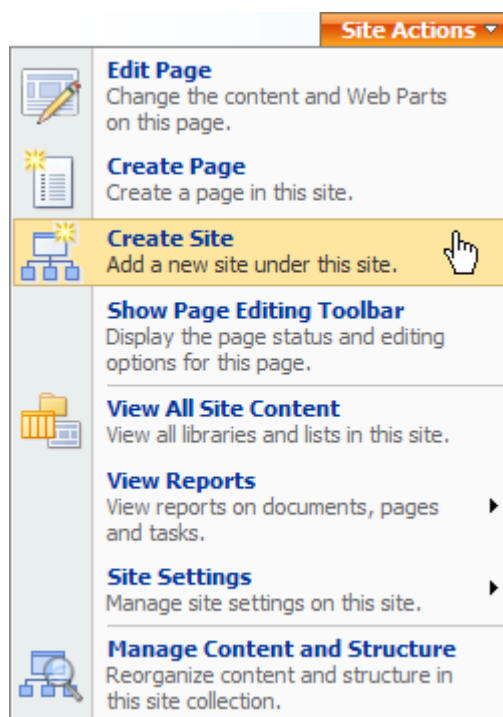
S vytvořeným hostovaným prostředím ve virtuálním stroji, nainstalovanými technologiemi a vybraným řešením MOSS jsem začal se zkoumáním a následnou implementací systému. Ta probíhala v portálu MOSS 2007.

Nejprve jsem si vytvořil vlastní stránku, na které jsem realizoval jednotlivé požadavky. Na ní jsem vytvořil datový model pomocí listů a jejich nadefinovaných sloupců. Na tyto sloupce jsem aplikoval pojistné matematické vzorce, které řeší optimalizaci finančních programů. Pro ověření správnosti optimalizace jsem vytvořil skript. Výsledek skriptu jsem porovnal s výsledkem v MOSS. Kvůli přehlednosti výpisu finančních portfolií jsem vytvořil vlastní Web Part.

5.1 Vytvoření stránky

Pro práci s klienty a finančními programy jsem si v MOSS vytvořil vlastní stránku (Site) *Finanční portfolia* pomocí následujících kroků:

1. V MOSS je třeba být na domovské stránce (v nejvyšší úrovni).
2. V pravém horním rohu zvolit rozevírací menu *Site Action*.
3. Vybrat možnost *Create Site* (Obrázek 20).



Obrázek 20: Nabídka Site Actions s výběrem Create Site

4. Otevře se stránka *New SharePoint Site*, na níž jsem zvolil a vyplnil následující údaje:

- **Title** – Finanční portfólia
- **Description** – Práce s klienty a finančními programy
- **URL name** – <http://nuvmdev/financniportfolia>
- **Select a template** – záložka *Collaboration*, položka *Blank Site*
- **User Permissions** – možnost *Use same permissions as parent site*
- **Use the top link bar from the parent site?** – možnost *Yes*
- Ostatní položky jsem ponechal v původním stavu.

Nyní je na domovské stránce SharePointu vytvořena nová stránka *Finanční portfólia*, která je zobrazena v horním panelu odkazů rodičovské stránky a v boční levé navigaci (Obrázek 21).



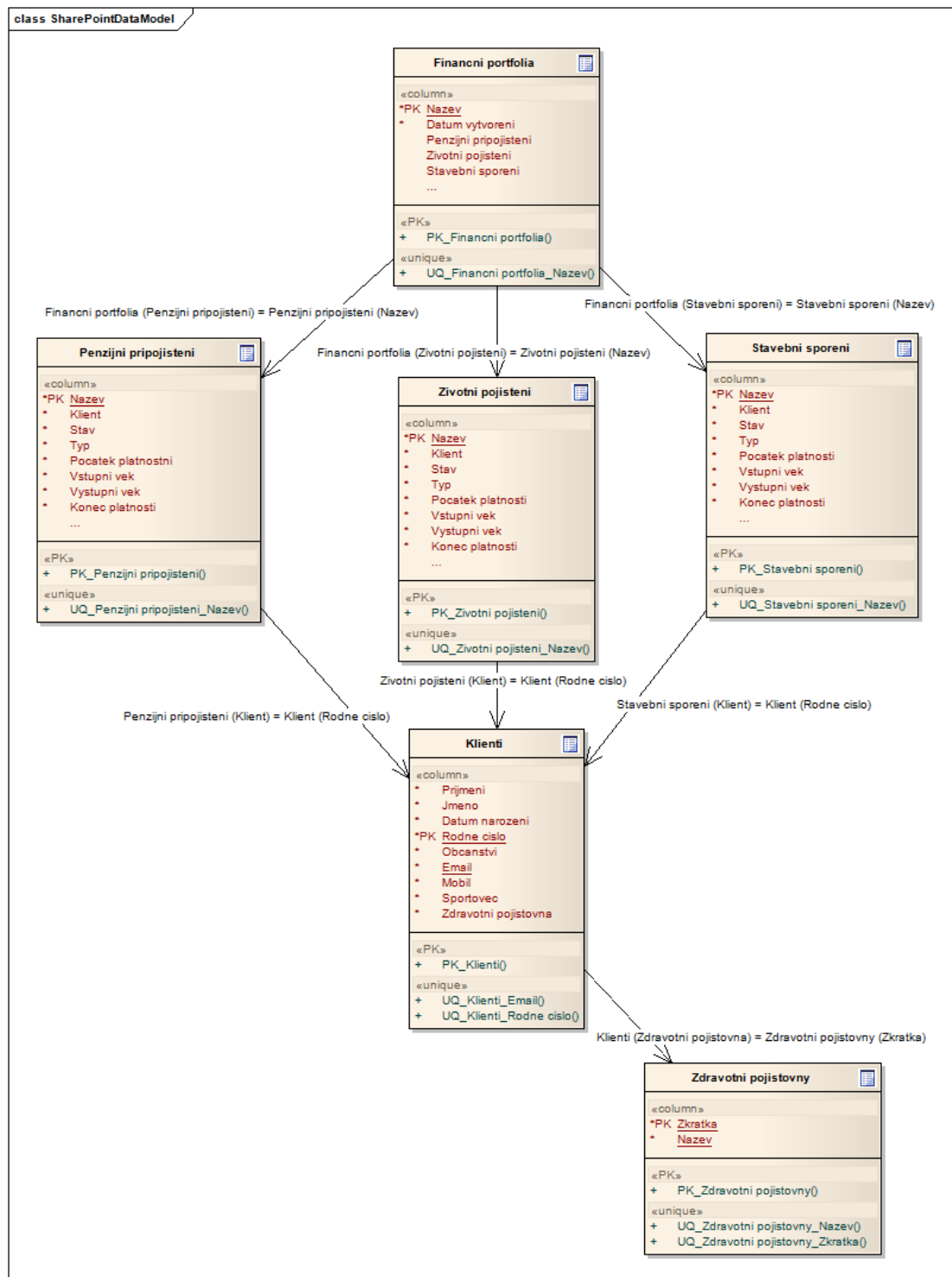
Obrázek 21: Domovská stránka s vytvořenou stránkou *Finanční portfólia*

5.2 Vytvoření listů

Na stránce *Finanční portfólia* jsem vytvořil datovou část pomocí listů (Lists). Při jejich zhotovování jsem vycházel z navrženého hierarchického datového modelu (Obrázek 22). Ten se skládá ze tří hlavních datových částí:


- Klienti
- Finanční programy – Penzijní přípojištění, Životní pojištění, Stavební spoření
- Finanční portfólia

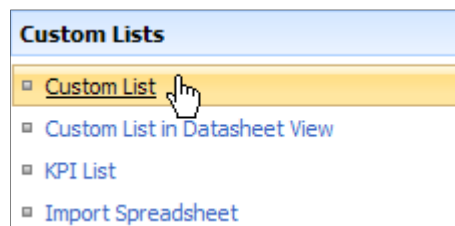
Finanční portfolia na sebe vážou *Finanční programy* (vazba 1:N). Každý z těchto *Finančních programů* na sebe váže *Klienta* (vazba 1:1); tím je určována uzavřená smlouva vůči klientovi. Každý z těchto *Klientů* na sebe váže *Zdravotní pojišťovnu* (vazba 1:1).



Obrázek 22: Hierarchický datový model finančních portfolií

Realizaci listů jsem řešil pomocí následujících kroků:

1. V MOSS je třeba být na stránce *Finanční portfólia*.
2. V levém bočním menu zvolit odkaz *View All Site Content* (Zobrazit veškerý obsah webu).
3. V hlavičce textové části zvolit odkaz *Create*  *Create*.
4. Otevře se stránka *Create* (Home → Finanční portfólia → Create), kde pro vytvořenou stránku *Finanční portfólia* je třeba zvolit ze sekce *Custom lists* odkaz *Custom list* (Obrázek 24).



Obrázek 23: Vytváření datové části pomocí Custom List

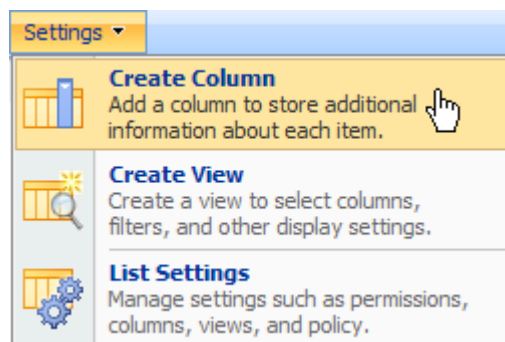
5. Otevře se stránka *New* (Home → Finanční portfólia → Create → New), kde jsem vložil požadované údaje:
 - **Name** – Zdravotní pojišťovny
 - **Description** – List zdravotních pojišťoven
 - **Display this list on the Quick Launch?** – možnost *Yes*

Stejným postupem jsem vytvořil listy: *Klienti*, *Životní pojištění*, *Penzijní připojištění* a *Finanční portfólia*.

5.3 Vytvoření sloupců

Ke každému listu jsem vytvořil i patřičné sloupce (atributy). V následujících krocích popisují vytvoření nového sloupce:

1. V MOSS je třeba být na stránce *Finanční portfólia*.
2. Z bočního levého menu vybrat takový odkaz listu z kategorie *Lists*, ve kterém mají být vytvářeny sloupce (např. list *Zdravotní pojišťovny*).
3. V rozevíracím menu *Settings* je třeba zvolit možnost *Create Column* (Obrázek 25).



Obrázek 24: Vytváření sloupců

- Otevře se stránka *Create Column* (Home → Finanční portfólia → Zdravotní pojišťovny → Settings → Create Column), kde je třeba pro vytvoření sloupce zadat *Column name* (název sloupce) a zvolit *The type of information in this column* (datový typ sloupce). Vše zobrazeno na Obrázku 25. Práci s jednotlivými datovými typy jsem popsal v následující podkapitole 5.3.1.

Obrázek 25: Vytváření nového sloupce – název a datový typ

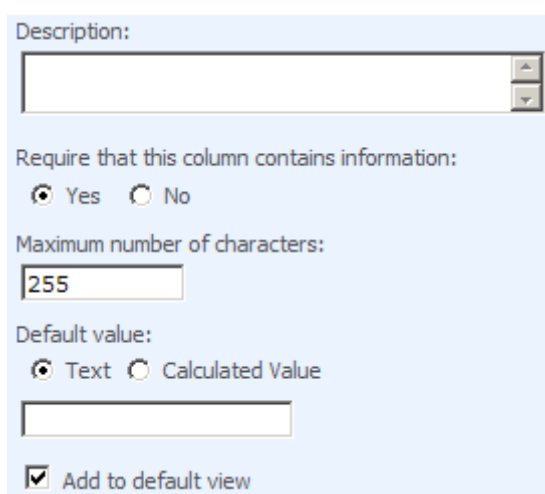
5.3.1 Práce s jednotlivými datovými typy

Při vytváření sloupce lze definovat datový typ (*The type information in this column is*). V projektu jsem pracoval s následujícími datovými typy: *Single line of text*, *Choice*, *Number*, *Date and Time*, *Lookup*, *Yes/No* a *Calculated*. Práce s jednotlivými z nich je popsána v následujících odstavcích této kapitoly.

Single line of text

Single line of text pracuje s řetězcem znaků zadané délky a obsahuje parametry (Obrázek 26):

- **Description** – popis sloupce
- **Require that this column contains information** – požadovaný záznam
- **Maximum number of characters** – maximální počet vkládaných znaků
- **Default value** – počáteční hodnota při nezadaném řetězci, může být zadán řetězec nebo vypočítaná hodnota z jiných sloupců daného listu
- **Add to default view** – přidání do základního pohledu (záznam bude vypsán)



The image shows a configuration window for a 'Single line of text' data type. It contains the following elements:

- Description:** A text input field with up and down arrow buttons on the right.
- Require that this column contains information:** Two radio buttons, with 'Yes' selected and 'No' unselected.
- Maximum number of characters:** A text input field containing the number '255'.
- Default value:** Two radio buttons, with 'Text' selected and 'Calculated Value' unselected, followed by a text input field.
- Add to default view:** A checked checkbox.

Obrázek 26: Parametry datového typu Single line of text

Choice

Choice pracuje se zadaným výběrem a obsahuje parametry (Obrázek 27):

- **Description** – popis sloupce
- **Require that this column contains information** – požadovaný záznam
- **Type each choice on a separate line** – zadání jednotlivých hodnot oddělených novým řádkem
- **Display choices using** – volba zobrazení (Drop-Down Menu, Radio Buttons, Checkboxes)
- **Allow 'Fill-in' choices** – možnost vložení vlastní hodnoty
- **Default value** – počáteční hodnota při prázdném výběru, může být zadán řetězec nebo vypočítaná hodnota z jiných sloupců daného listu
- **Add to default view** – přidání do základního pohledu (záznam bude vypsán)

Description:

Require that this column contains information:
 Yes No

Type each choice on a separate line:

Enter Choice #1
Enter Choice #2
Enter Choice #3

Display choices using:
 Drop-Down Menu
 Radio Buttons
 Checkboxes (allow multiple selections)

Allow 'Fill-in' choices:
 Yes No

Default value:
 Choice Calculated Value

Enter Choice #1

Add to default view

Obrázek 27: Parametry datového typu Choice

Number

Number pracuje s číselnými hodnotami a obsahuje parametry (Obrázek 28):

- **Description** – popis sloupce
- **Require that this column contains information** – požadovaný záznam
- **Specify min a max value** – minimální a maximální hodnota
- **Number of decimal place** – nastavení desetinné čárky (hodnota Automatic nebo 0–5)
- **Default value** – počáteční hodnota při nezvolené hodnotě, může být zadán řetězec nebo vypočítaná hodnota z jiných sloupců daného listu
- **Show as percentage** – nastavení hodnoty jako procentuální
- **Add to default view** – přidání do základního pohledu (záznam bude vypsán)

Description:

Require that this column contains information:
 Yes No

You can specify a minimum and maximum allowed value:
 Min: Max:

Number of decimal places:

Default value:
 Number Calculated Value

Show as percentage (for example, 50%)

Add to default view

Obrázek 28: Parametry datového typu Number

Date and Time

Date and Time pracuje s datem a časem a obsahuje parametry (Obrázek 29):

- **Description** – popis sloupce
- **Require that this column contains information** – požadovaný záznam
- **Date and Time Format** – volba zobrazení buď data, nebo data a času
- **Default value** – počáteční hodnota při nezvoleném datu nebo času, může být zadán konkrétní den, konkrétní čas nebo vypočítaná hodnota pracující s datem nebo časem
- **Add to default view** – přidání do základního pohledu (záznam bude vypsán)

Description:

Require that this column contains information:

Yes No

Date and Time Format:

Date Only Date & Time

Default value:

(None)

Today's Date

12 AM 00

Enter date in M/D/YYYY format.

Calculated Value:

Add to default view

Obrázek 29: Parametry datového typu Date and Time

Lookup

Lookup spojuje vytvářený sloupec s již vytvořeným sloupcem vybraného listu a obsahuje parametry (Obrázek 30):

- **Description** – popis sloupce
- **Require that this column contains information** – požadovaný záznam
- **Get information from** – výběr listu
- **In this column** – výběr sloupce z vybraného listu, lze vybírat i více hodnot zvolením možnosti *Allow multiple values* (povolit vícenásobné hodnoty)
- **Add to default view** – přidání do základního pohledu (záznam bude vypsán)

Description:

Require that this column contains information:

Yes No

Get information from:

Zdravotní pojišťovny

In this column:

Zkratka

Allow multiple values

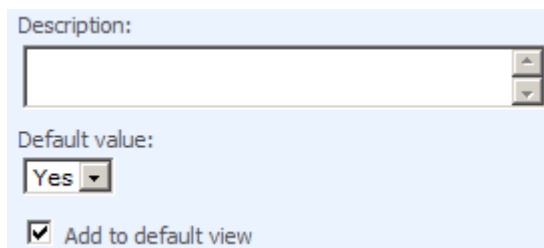
Add to default view

Obrázek 30: Parametry datového typu Lookup

Yes/No

Yes/No pracuje s logickou hodnotou ano/ne (Obrázek 31).

- **Description** – popis sloupce
- **Default value** – nastavení počáteční hodnoty; možnost *Yes* nebo *No*
- **Add to default view** – přidání do základního pohledu (záznam bude vypsán)



The image shows a configuration window for a 'Yes/No' data type. It has a light blue background. At the top, there is a label 'Description:' followed by a text input field. Below that is a label 'Default value:' followed by a dropdown menu currently showing 'Yes'. At the bottom, there is a checkbox labeled 'Add to default view' which is checked.

Obrázek 31: Parametry datového typu Yes/No

Calculated

Calculated umožňuje vytvořit vypočítaný sloupec (sum, dif) založený na hodnotách jiných sloupců aktuálního listu (Obrázek 32):

- **Description** – popis sloupce
- **Formula** – vkládaný matematický vzorec
- **The data type returned from this formula** – volba datového typu vráceného výsledku ze zadaného matematického vzorce, možnosti datových typů: Single line of text, Number, Currency, Date and Time, Yes/No
- **Add to default view** – přidání do základního pohledu (záznam bude vypsán)

Description:

Formula:

Insert Column:

- Content Type
- Created
- Modified
- Název
- Zkratka

Add to formula

The data type returned from this formula is:

- Single line of text
- Number (1, 1.0, 100)
- Currency (\$, ¥, €)
- Date and Time
- Yes/No
- Add to default view

Obrázek 32: Parametry datového typu Calculated

5.4 Přehled vytvářených listů a jejich sloupců

Vytváření sloupců u jednotlivých listů je podle stejného postupu, jako je tomu v kapitole 5.3. Celkový přehled vytvořených listů a sloupců této práce je uveden v následujících tabulkách (Tabulka 6, Tabulka 7, Tabulka 8, Tabulka 9).

Tabulka 6: List Zdravotní pojišťovny

Zdravotní pojišťovny		
Název sloupce	Datový typ	Požadované údaje
Zkratka	Single line of text	Ano
Název	Choice	Ano

Tabulka 7: List Klienti

Klienti		
Název sloupce	Datový typ	Požadované údaje
Příjmení	Single line of text	ano
Jméno	Single line of text	ano
Datum narození	Date and Time	ano

Rodné číslo	Single line of text	ano
Občanství	Choice	
Email	Single line of text	
Mobil	Single line of text	
Telefon	Single line of text	
Sportovec	Yes/No	
Zdravotní pojišťovna	Lookup	ano

Tabulka 8: List Životní pojištění

Životní pojištění		
Název sloupce	Datový typ	Požadované údaje
Název	Single line of text	ano
Klient	Lookup	
Stav	Choice	ano
Typ	Single line of text	
Rok narození	Number	ano
Počátek platnosti	Number	ano
Konec platnosti	Number	ano
Vstupní věk	Calculated (calculation based on other columns)	
Výstupní věk	Calculated (calculation based on other columns)	
Zhodnocení	Number	
Poplatek	Number	ano
Doba platnosti	Calculated (calculation based on other columns)	
Frekvence plateb	Choice	
Běžné pojistné	Number	ano
Zaplaceno celkem	Calculated (calculation based on other columns)	
Hodnota konec	Calculated (calculation based on other columns)	
Bilance	Calculated (calculation based on other columns)	

Tabulka 9: List Finanční portfólia

Finanční portfólia		
Název sloupce	Datový typ	Požadované údaje
Název	Single line of text	ano
Datum vytvoření	Date and Time	
Penzijní připojištění	Lookup	
Životní pojištění	Lookup	

5.5 Pojistné matematické vzorce

Pro výpočet hodnot stávajících a optimalizovaných finančních programů životního pojištění jsem použil pojistné matematické vzorce. Ty jsem aplikoval do jednotlivých sloupců datového typu Calculated listu *Životní pojištění* v poli *Formula*.

Vstupní věk

Vstupní věk pojištěného je roven rozdílu *počátku platnosti* a *roku narození klienta* (Vzorec 1).

$$\text{Vstupni vek} = \text{Pocatek platnosti} - \text{Rok narozeni klienta}$$

Vzorec 1

Výstupní věk

Výstupní věk pojištěného je roven rozdílu *konce platnosti* a *roku narození klienta* (Vzorec 2).

$$\text{Vystupni vek} = \text{Konec platnosti} - \text{Rok narozeni klienta}$$

Vzorec 2

Doba platnosti

Doba platnosti je doba, na kterou bylo pojištění sjednáno a je rovna rozdílu *výstupního věku* a *vstupního věku* (Vzorec 3).

$$\text{Doba platnosti} = \text{Vystupni vek} - \text{Vstupni vek}$$

Vzorec 3

Zaplaceno celkem

Zaplaceno celkem je součin hodnot *doba platnosti*, *frekvence plateb* a *běžného pojistného* (Vzorec 4).

$$\text{Zaplaceno celkem} = \text{Doba platnosti} \times \text{Frekvence plateb} \times \text{Bezne pojistne}$$

Vzorec 4

Hodnota konec

Hodnota konec je vypočítaná pomocí následujících matematických vzorců (Vzorec 5, Vzorec 6, Vzorec 7, Vzorec 8). Pro její použití v MOSS jsem musel provést úpravu na jiný matematický výraz; vše jsem popsal v kapitole 5.6.

$$PPER = Doba\ platnosti \times Frekvence\ plateb$$

Vzorec 5

$$Splatka = Bezne\ pojistne - Poplatek$$

Vzorec 6

$$Sazba = \frac{Zhodnoceni \times 0,01}{Frekvence\ plateb}$$

Vzorec 7

$$Hodnota\ konec = \sum_{i=1}^{PPER} (Hodnota\ konec + Splatka) \times (1 + Sazba)$$

Vzorec 8

Bilance

Bilance udává návratnost vkladu do pojistné smlouvy, a to je rozdíl *hodnoty konec* a *běžného pojistného* za celou dobu trvání pojistné smlouvy (Vzorec 9).

$$Bilance = Hodnota\ konec - Zaplaceno\ celkem$$

Vzorec 9

5.6 Upravení vzorce Hodnota konec

Datový typ Calculated neumožňuje pracovat s programováním cyklů. Vzorec 8 jsem tedy musel upravit na takový matematický výraz, který nevyužívá sumu. Pro tuto úpravu jsem tedy použil geometrickou posloupnost, která vychází ze základního matematického vzorce (Vzorec 10):

$$s_n = a_1 \times \frac{1 - q^n}{1 - q}$$

Vzorec 10

Proměnná a_1 je *prvním členem* geometrické posloupnosti, tj. $a_1 = \text{Splatka} \times (1 + \text{Sazba})$. Proměnná q je poměr dvou po sobě jdoucích členů nazývaný *kvocient*, tj. $q = 1 + \text{Sazba}$. Pro výpočet *kvocientu* jsem vycházel z následujících kroků:

$$\begin{aligned} a_0 &= \text{Splatka} \\ a_1 &= \text{Splatka} \times (1 + \text{Sazba}) \\ a_2 &= \text{Splatka} \times (1 + \text{Sazba})^2 \\ q &= \frac{a_2}{a_1} = \frac{\text{Splatka} \times (1 + \text{Sazba})^2}{\text{Splatka} \times (1 + \text{Sazba})} = 1 + \text{Sazba} \end{aligned}$$

Po zjištění proměnných a_1 a q získávám vzorec (Vzorec 11) použitelný pro datový typ Calculated:

$$\begin{aligned} \text{Hodnota konec} &= \left((\text{Bezne pojistne} - \text{Poplatek}) \times \left(1 + \frac{\text{Zhodnoceni} \times 0,01}{\text{Frekvence plateb}} \right) \right) \\ &\times \left(\frac{1 - \left(1 + \frac{\text{Zhodnoceni} \times 0,01}{\text{Frekvence plateb}} \right)^{\text{Doba platnosti} \times \text{Frekvence plateb}}}{1 - \left(1 + \frac{\text{Zhodnoceni} \times 0,01}{\text{Frekvence plateb}} \right)} \right) \end{aligned}$$

Vzorec 11

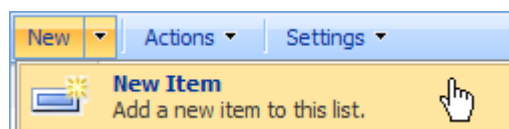
U listu *Životní pojištění* a jeho sloupce *Hodnota konec* jsem následně do pole *Formula* aplikoval Vzorec 11 ve tvaru:

```
=ROUND(( [Běžné pojistné] - [Poplatek] ) *
( 1 + (( [Zhodnocení] * 0.01) / [Frekvence plateb] ) ) *
( ( 1 - ( 1 + (( [Zhodnocení] * 0.01) / [Frekvence plateb] ) ) ^
( [Doba platnosti] * [Frekvence plateb] ) ) ) /
( 1 - ( 1 + (( [Zhodnocení] * 0.01) / [Frekvence plateb] ) ) ) ) ) , 0)
```

5.7 Vložení nové položky

Po vytvoření listů, jejich sloupců s definováním jednotlivých parametrů a formulí jsem mohl přistoupit k vložení nové položky (vkládání položky do mnou vytvořeného listu *Životní pojištění* na stránce *Finanční portfólia*):

1. V MOSS je třeba být na stránce *Finanční portfólia*.
2. Z bočního levého menu vybrat takový odkaz listu z kategorie *Lists*, kam má být vložen nový záznam; v mém případě list *Zdravotní pojišťovny*.
3. Zvolit rozevírací odkaz *New* → *New Item* (Obrázek 33).



Obrázek 33: Nová položka

4. Otevře se stránka *Životní pojištění: New Item* (Home → *Finanční portfólia* → *Životní pojištění* → *New Item*), která obsahuje mnou vytvořené oblasti pro vkládání dat. Do nich jsem vložil data, která jsou na Obrázku 34.

Název *	SPZ001
Klient	800101/1111
Stav *	stávající
Typ	B150
Rok narození *	1985
Počátek platnosti *	2010
Konec platnosti *	2030
Zhodnocení	0.06
Poplatek *	0
Frekvence plateb	2
Běžné pojistné *	500

OK Cancel

Obrázek 34: Data nové položky

5. Po správném vložení se nová položka uloží do listu a ten se zobrazí s veškerými již vloženými položkami (v mém případě list *Životní pojištění*; Obrázek 35 a Obrázek 36).

Název	Klient	Stav	Typ	Počátek platnosti	Konec platnosti	Rok narození	Vstupní věk	Výstupní věk
SZP001	401009/1234	stávající	B150	2,010	2,030	1,985	25	45

Obrázek 35: Vložená položka v listu Životní pojištění (část 1)

Doba platnosti	Frekvence plateb	Běžné pojistné	Zaplaceno celkem	Zhodnocení	Poplatek	Hodnota konec	Bilance
20	2	500	20,000	0.06	0	20,123	123

Obrázek 36: Vložená položka v listu Životní pojištění (část 2)

5.8 Ověření správnosti vzorce

Pro ověření správnosti Vzorce 11 jsem naprogramoval kód, který vychází z matematických vzorců popsanych v kapitole 5.5. Počáteční parametry (v kódu konstanty `const`) jsem zvolil stejné jako při vkládání dat u nové položky v kapitole 5.7 na Obrázku 34:

- Frekvence plateb = 2
- Rok narození = 1985
- Počátek platnosti = 2010
- Konec platnosti = 2030
- Běžné pojistné = 500
- Poplatek = 0
- Zhodnocení = 0,06

Naprogramovaný kód obsahuje konstanty, se kterými pracuji při matematických výpočtech. Důležitou proměnnou je hodnota `konec`, která po provedení programu vypíše číselnou hodnotu.

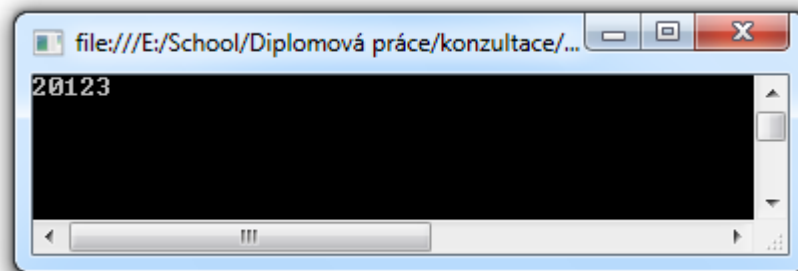
```
class Program
{
    const int frekvencePlateb = 2;
    const int rokNarozeni = 1985;
    const int pocatekPlatnosti = 2010;
    const int konecPlatnosti = 2030;
    const int beznePojistne = 500;
    const int poplatek = 0;
    const float zhodnoceni = 0.06f;
    static void Main(string[] args)
    {
        int vstupniVek = pocatekPlatnosti - rokNarozeni;
        int vystupniVek = konecPlatnosti - rokNarozeni;
    }
}
```

```

int dobaPlatnosti = vystupniVek - vstupniVek;
int zaplacenoCelkem = dobaPlatnosti * frekvencePlateb *
int pper = dobaPlatnosti * frekvencePlateb;
int splatka = beznePojistne - poplatek;
float sazba = (zhodnoceni / 100) / frekvencePlateb;
float hodnotaKonec = 0;
for (int i = 0; i < pper; i++)
    hodnotaKonec = (hodnotaKonec + splatka) * (1 + sazba);
Console.WriteLine(Math.Round(hodnotaKonec));
Console.ReadKey();
}
}

```

Konzolí vypsaná hodnota `hodnotaKonec` vypočítaná pomocí cyklu (Obrázek 37) se shoduje s vypočítanou hodnotou upraveného matematického výrazu řešeného v MOSS (Obrázek 36, sloupec *Hodnota konec*).



Obrázek 37: Výsledek naprogramovaného kódu

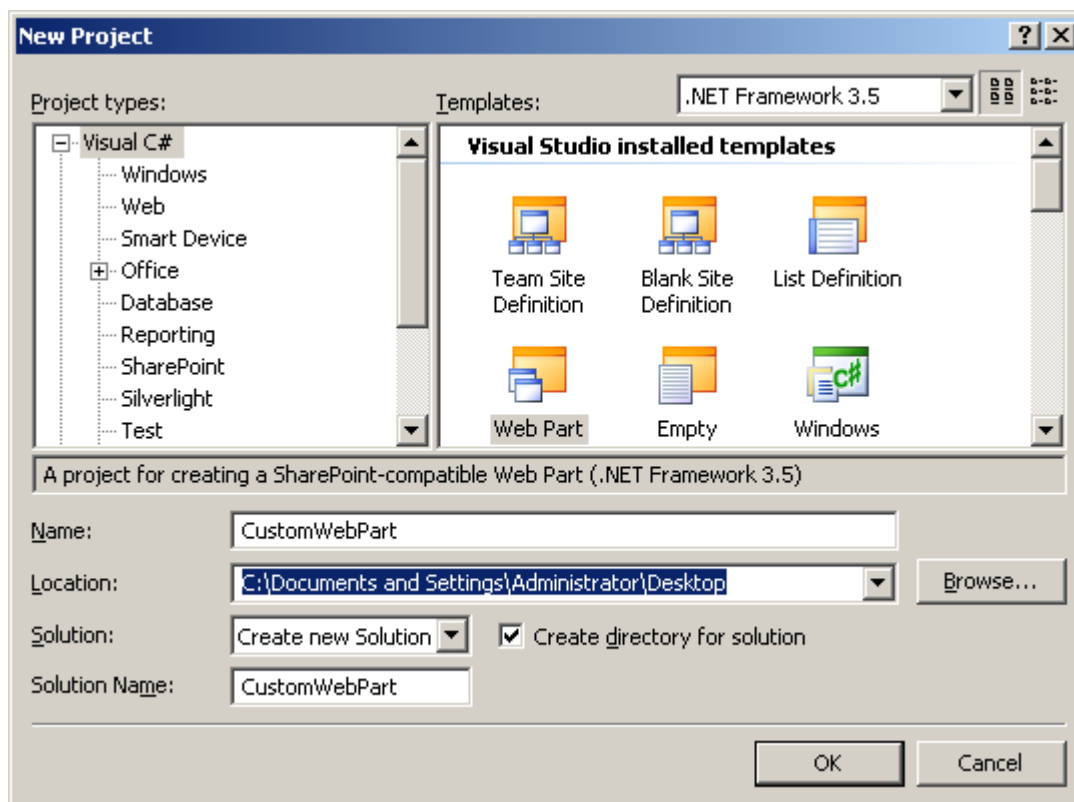
5.9 Vytvoření a integrace Web Parts

Po vytvoření datového modelu, který byl naplněn daty a ověřena jeho funkčnost, bylo potřeba tyto údaje vypsat ve speciálním tvaru. Protože MOSS neumožňoval takový výpis, který by byl vhodný pro uživatele, musel jsem ho naprogramovat pomocí webových částí neboli *Web Parts*.

Pro realizaci své vlastní *Web Part* jsem použil vývojové prostředí Visual Studio 2008, ve kterém je po nainstalování WSS předpřipravená šablona (template) pro práci s těmito webovými částmi. Tu jsem následně připojil k portálu MOSS.

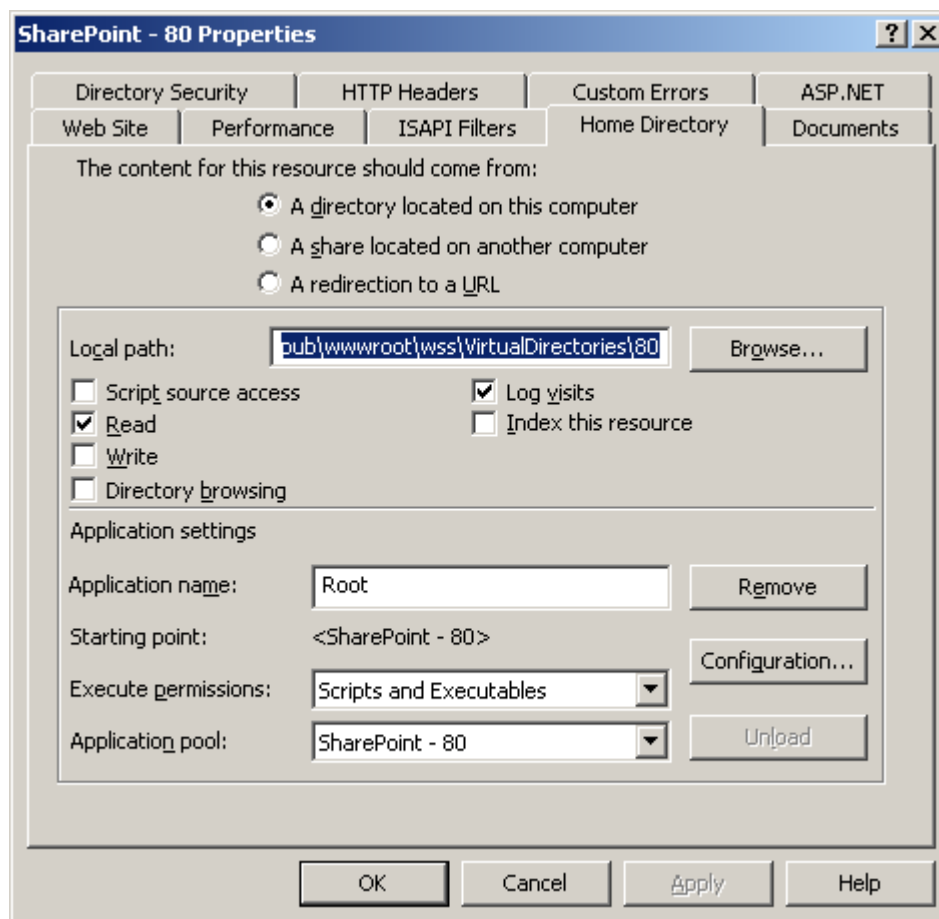
Postup vytvoření a integrace *Web Part* je v následujících krocích:

1. Ve Visual Studiu 2008 vybrat nabídku *File* → *New* → *Project*.
2. V nabízených šablonách (templates) vybrat položku *Web Part* (Obrázek 38).



Obrázek 38: Vytvoření nové Web Part

3. Napsat vlastní kód, který bude určovat funkčnost *Web Part*; mnou vytvořená Web Part je popsána v kapitole 5.11.
4. Nalézt cestu k portálu pomocí IIS (Start → Programs → Administrative Tools → Internet Information Services). V něm zvolit *Web Sites* a vybrat pravým tlačítkem *SharePoint – 80* s možností *Properties*, dále v něm vybrat záložku *Home Directory*. Zde potom najdeme umístění uloženého MOSS v části *Local path* (Obrázek 39).




Obrázek 39: Zjištění umístění MOSS pomocí IIS

5. Vytvořenou a zkompilovanou *Web Part* s příponou *dll* (soubor umístěn ve vytvářeném projektu, složka *bin* → *debug*) nahrát do složky *bin*, která je umístěna na zjištěné cestě ze 4. kroku (Local Path).
6. Zkompilovanou knihovnu vytvářené *Web Part* dále zkopírovat do složky *C:\WINDOWS\Assembly*.
7. Ve složce *C:\WINDOWS\Assembly* nalézt zkopírovanou *Web Part*. Na ni pomocí pravého tlačítka vybrat možnost *Properties* a zde zkopírovat *Public Key Token*, který lze použít v následujícím kroku.
8. Pro využití a zároveň bezpečné řízení *Web Part* je třeba upravit soubor *web.config*, který se nachází opět na zjištěné cestě ze 4. kroku (Local Path). Ten otevřít a do části *<SafeControls>* vložit následující kód, kde *PublicKey-Token* bude obsahovat uložený klíč z předešlého kroku:

```
<SafeControls>
  <SafeControl Assembly="CustomWebPart, Version=1.0.0.0,
```

```
Culture=neutral, PublicKeyToken=9f4da00116c38ec5"
Namespace="CustomWebPart" TypeName="*" Safe="True"
AllowRemoteDesigner="True" />
</SafeControls>
```

9. V prohlížeči otevřít portál MOSS v nejvyšší úrovni, vybrat v pravé horní části možnost *Site Actions* → *Site Settings* → *Modify All Site Settings*. Na stránce *Site Settings* v části *Galleries* zvolit odkaz *Web Parts*.
10. Zvolit odkaz New .
11. Na stránce *Web Part Gallery: New Web Parts* (Home → Web Part Gallery → New Web Parts) vyhledat v seznamu vytvořenou Web Part (CustomWebPart). Zaškrtnout její checkbox a klepnout na tlačítko *Populate Gallery* v horní části stránky. Tím se vytvoří položka *Web Part* v seznamu galerie webových částí a zároveň je připravena k přidání do stránky.
12. V posledním kroku je třeba vytvořit stránku s vlastní *Web Part*. Vzhled stránky se modifikuje kliknutím na tlačítko *Edit Page*. V této situaci lze přidat webovou část do zóny pomocí tlačítka *Add a Web Part*. Dále je třeba vybrat vlastní Web Part v sekci *Miscellaneous* a kliknout na tlačítko přidat. Při volbě *Exit Edit Mode* se stránka vrátí do režimu zobrazení s již přidanou Web Part.

5.10 Ladění Web Parts

Při vytváření vlastní Web Part jsem potřeboval i její ladění (debugging). Abych to však mohl uskutečnit, musel jsem připojit spuštěný proces k vývojovému prostředí:

1. Spustit příkazovou řádku a nastavit cestu na *C:\WINDOWS\system32*
2. Zadat příkaz *iisapp.vbs*
3. Zjistit číslo procesu (PID; Process ID) portálu SharePoint; tedy řádek, kde je *SharePoint* – 80. Na Obrázku 40 je identifikační číslo portálu 2752.


```
cmd C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>cd C:\WINDOWS\system32

C:\WINDOWS\system32>iisapp.vbs
W3WP.exe PID: 2752 AppPoolId: SharePoint - 80
W3WP.exe PID: 2252 AppPoolId: SharePoint Central Administration v3

C:\WINDOWS\system32>
```

Obrázek 40: Zjištění spuštěného procesu

4. Ve Visual Studiu 2008 zvolit *Debug* → *Attach to Process*.
5. Připojit proces s PID, který byl zjištěn pomocí příkazové řádky (tedy PID 2752).
6. Nyní je nastaven proces k ladění. Ten bude spuštěn ve Visual Studiu při jeho vyvolání z MOSS. Pro ladění procesu je třeba nastavit breakpoint u spouštěné Web Part.

5.11 Vlastní Web Part

Pro výpis finančních portfolií jsem naprogramoval ve Visual Studiu 2008 vlastní *Web Part*. Při realizaci jsem pracoval s klíčovými třídami SharePointu (Microsoft.SharePoint): *SPSite*, *SPWeb* a *SPList*.

Hlavní částí *Web Part* je metoda *CreateChildControls*, ve které jsou vytvářeny všechny ovládací prvky.

```
protected override void CreateChildControls()  
{  
    base.CreateChildControls();  
}
```

Pro reprezentaci tabulky dat v paměti jsem využil objekt *sourceDataTable* typu *DataTable* s nadefinovanými sloupci *Název*, *Smlouvy* a *Klient*.

```
sourceDataTable = new DataTable();  
sourceDataTable.Columns.Add("Název");
```

```
sourceDataTable.Columns.Add("Smlouvy");  
sourceDataTable.Columns.Add("Klient");
```

Vytvořená instance *dataRow* třídy *DataRow* představuje jednotlivé záznamy v objektu *sourceDataTable*. Při každém přidání záznamu do tabulky se vždy vytvoří instance třídy *DataRow*, která se potom přidá do kolekce řádků instance *sourceDataTable*.

```
DataRow dataRow = new DataRow();
```

Použil jsem instanci *mySite* třídy *SPSite* pro získání kolekce webů ze zadané adresy, na které jsou jednotlivé listy.

```
using (SPSite mySite = new  
SPSite("http://nuvmdev/financniportfolia/Default.aspx"))
```

Instance *myWeb* třídy *SPWeb* představuje webovou stránku WSS. Pro její otevření jsem použil metodu *OpenWeb* třídy *SPSite*, která navrácí konkrétní webovou stránku objektu *mySite*.

```
using (SPWeb myWeb = mySite.OpenWeb())
```

Instance *spListFP* třídy *SPList* představuje konkrétní seznam na SharePoint webové stránce. Vlastnost *Lists[]* vrací kolekci seznamů objektu *myWeb*; tedy kolekci finančních portfolií.

```
SPList spListFP = myWeb.Lists["Finanční portfolia"];
```

Instance *zaznamFP* třídy *SPListItem* reprezentuje jeden řádek seznamu. Prostřednictvím vlastnosti *Items* objektu *spListFP* získám kolekci prvků (finančních portfolií) typu *SPListItem*, reprezentujících všechny řádky seznamu, které příkazem *foreach* procházím.

Pro uložení finančních programů daného finančního portfolia vytvářím instanci *seznamFinProgramu*. Pomocí vytvořeného rozhraní *SpravujFinancniProgramy* a jeho realizací *DejFinancniProgramy* s parametrem *finPortfolio* typu *SPListItem* jsou navraceny do instance *seznamFinProgramu* všechny finanční programy, které spadají do daného finančního portfolia.

Instanci *seznamFinProgramu* procházím příkazem *foreach* a v něm pomocí metody *VytvorDataRow* vkládám záznamy do instance *dataRow*, které jsou následně předány instanci *sourceDataTable*.

```
foreach (SPListItem zaznamFP in spListFP.Items)
```

```

{
    SPListItem finPortfolio = zaznamFP;
    spravujFinancniProgramy = new SpravujFinancniProgramy();
    List<FinancniProgram> seznamFinProgramu = new
        List<FinancniProgram>();
    seznamFinProgramu =
        spravujFinancniProgramy.DejFinancniProgramy(finPortfolio);
    foreach (FinancniProgram finProgram in seznamFinProgramu)
    {
        DataRow = VytvorDataRow(
            finProgram.FinPortfolio["Název"].ToString(),
            finProgram.FinProgram["Název"].ToString(),
            finProgram.Klient["Rodné číslo"].ToString());
    }
}

```

Vypsání dat je řešeno pomocí komponenty *SPGridView*. Po dokončení jsem *Web Part* integroval do MOSS (integrace popsána v kapitole 5.9). Její výpis je řešen pro finanční portfolia, ve kterých jsou uzavřené finanční programy penzijního připojištění (Obrázek 41).

Název	Smlouvy	Klient
FP001	PP001	401009/1234
	PP002	451145/9876
FP002	PP001	401009/1234

Obrázek 41: Výpis finančních portfolií realizované vlastní Web Part

Závěr

V této diplomové práci jsme zaměřili pozornost s Bc. Petrem Plavcem na současné dostupné technologie, pomocí nichž bychom realizovali můj původní záměr z bakalářské práce a využili tak i její stávající funkčnost. Provedli jsme tedy teoretickou analýzu současných dostupných technologií. Z nich jsme si pak vybrali tu nejefektivnější, která řeší stěžejní části původního systému a zároveň je silným nástrojem pro realizaci dalších případů užití. Dále jsme hledali možnost umístění vybrané technologie na nějaký z dostupných serverů.

Tato fáze práce byla pro nás náročná časově. Rozhodovali jsme se, kde námi vybranou technologii můžeme provozovat. Původně se nabízely dvě možnosti hostování naší technologie na serveru. První cestou byl placený hosting na komerčním serveru, který jsme ale zamítli pro finanční náročnost. Druhou variantou bylo použití školního serveru. Jeho nevýhodou byl přístup pouze z akademické pudy, což nám nevyhovovalo, protože jsme většinou pracovali doma. Během našeho rozhodování se pro nás ovšem naskytla třetí varianta, a to hostování technologie ve virtualizovaném prostředí na našich osobních počítačích. Tuto cestu jsme si zvolili.

V další etapě jsme už každý pracovali samostatně na svém úkolu. V přípravné části jsem nejdříve hledal vhodné virtualizační prostředí. Rozhodl jsem se pro VMware Workstation, který jsem si nainstaloval do svého počítače a ten se tak stal hostitelským systémem. Do něho jsem následně nainstaloval operační systém Microsoft Windows Server 2003 R2, který se tak stal hostovaným systémem. V něm jsem provedl všechny dostupné systémové aktualizace a doinstaloval veškeré softwarové balíčky. Těch bylo poměrně dost a musel jsem pracně vyhledávat právě ty, které byly potřebné pro nainstalování technologie. Tu jsem následně nainstaloval. Ovšem i zde jsem strávil dost času s jejím nastavením pro potřebné bezproblémové fungování.

Potom jsem zkoumal tuto dost časově náročnou technologii, zda je pro náš účel dostačující. Po jejím otestování jsme se ji rozhodli použít pro naši práci. Pokračoval jsem tedy realizací stěžejních případů užití, na nichž jsme se dohodli s vedoucím práce a spolužákem. Při práci s touto technologií jsem řešil úkoly spojené s vytvářením struktury stránek systému. Věnoval jsem se vytvoření datové části pomocí listů a jejich vzájemnému propojení, které pracují s údaji klientů a finančních programů. Následně jsem aplikoval do finančního programu životního pojištění po-

jistné matematické vzorce, které vypočítávají jednotlivé hodnoty stávajícího a optimalizovaného finančního programu. Zde ale nastal problém, protože použitý systém neumožňoval ve svém nabízeném prostředí vypočítat hodnotu založenou na ostatních sloupcích pomocí programování cyklů. Celou záležitost jsem tedy řešil nahrazením za matematický výraz řešený pomocí geometrické posloupnosti. Tento výpočet jsem ověřil prostřednictvím naprogramovaného modulu.

Systém neumožňoval takový výpis z uložených dat, který potřebuje finanční poradce pro přehledné zobrazení finančního portfolia. Proto jsem nastudoval další možnost rozšíření systému a našel jsem řešení díky Web Part. Tento výpis jsem realizoval pomocí vývojového prostředí Visual Studio 2008, při němž jsem využíval knihovny Microsoft SharePoint.

Přínosem této práce pro mě je získání přehledu a dalších informací o současných dostupných technologiích. Seznámil jsem se s novou, v dnešní době, převážně v zahraničí uznávanou technologií Microsoft Office SharePoint Server. Tu jsem prostudoval do té míry, kterou jsem potřeboval pro realizaci nejen v této práci, ale i pro využití v mých dalších projektech. Dále byla pro mě užitečná týmová spolupráce s Bc. Petrem Plavcem za soustavného vedení a konzultací Ing. Jaroslava Lacha.

Tato diplomová práce je pouze částí velmi rozsáhlého projektu. Po jeho případném dokončení by přinesl užitek finančním poradcům v jejich práci.

Použitá literatura

- [1] BĚLOHLÁVEK, Petr. *Návrh a tvorba aplikace pro evidenci a optimalizaci klientských smluv*. Pardubice, 2008. 56 s. Bakalářská práce. Univerzita Pardubice, FEI, Katedra IT. Dostupné z WWW: <<http://hdl.handle.net/10195/30237>>.
- [2] *Sharepointlabs.com* [online]. 2010 [cit. 2010-08-17]. Microsoft SharePoint Portal Development. Dostupné z WWW: <<http://www.sharepointlabs.com/microsoft-sharepoint-portal-development.htm>>.
- [3] *Trask solution* [online]. 2010 [cit. 2010-08-17]. Single Sign-On. Dostupné z WWW: <<http://www.trask.cz/single-sign-on>>.
- [4] PETERKA, Jiří. *eArchiv.cz* [online]. 2010 [cit. 2010-08-17]. Rendering. Dostupné z WWW: <<http://www.earchiv.cz/a96/a644k130.php3>>.
- [5] HUMPOLEC, Martin. *Blog Martina Humpolce* [online]. 25.11.2008 [cit. 2010-08-17]. Business Data Catalog v SharePointu. Dostupné z WWW: <<http://martinhumpolec.cz/A55BE2/Blog.nsf/dx/business-data-catalog-v-sharepointu.htm>>.
- [6] STERLING, David Matthew. *Microsoft Office SharePoint Server 2007 : The Complete Reference*. 2008. United States of America : The McGraw-Hill Companies, 2008. 788 s.
- [7] NEUWIRTH, Michal. *Vyvojar.cz* [online]. 20.8.2007 [cit. 2010-08-17]. Úvod do WSS 3.0. Dostupné z WWW: <<http://www.vyvojar.cz/Articles/524-uvod-do-wss-3-0.aspx>>.
- [8] VAŠEK, Jiří. *PCTuning.cz* [online]. 24.9.2007 [cit. 2010-08-17]. Virtualizace aneb Jak vytvořit systém v systému. Dostupné z WWW: <<http://pctuning.tyden.cz/software/ladeni-windows/9429-virtualizace-aneb-jak-vytvorit-system-v-systemu>>.
- [9] *VMware Workstation* [online]. 2010 [cit. 2010-08-17]. VMware. Dostupné z WWW: <<http://www.vmware.com/products/workstation/>>.
- [10] STANEK, William R. *Microsoft Windows Server 2003 : Kapesní rádce administrátora*. Brno : Computer Press, 2003. 535 s.

- [11] *Microsoft Corporation* [online]. 2010 [cit. 2010-08-17]. Novinky v systému Windows Server 2003 R2. Dostupné z WWW:
<<http://www.microsoft.com/cze/windowsserver2003/r2/whatsnewinr2.msp>>
- [12] *Windows Server TechCenter* [online]. 2010 [cit. 2010-08-17]. Internet Information Services (IIS) 7.5. Dostupné z WWW:
<<http://technet.microsoft.com/en-us/library/dd364124%28WS.10%29.aspx>>.
- [13] *Průručka vývojáře pro přechod na platformu Microsoft .NET*. [s.l.] : Microsoft Corporation, 2003. 199 s.
- [14] BĚHÁLEK, Marek. *VŠB TU Ostrava* [online]. 2007 [cit. 2010-08-17]. Architektura .NET Framework. Dostupné z WWW:
<<http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text/ch01s01.html>>.
- [15] ASANKA, Dinesh. *Windows Live* [online]. 2010 [cit. 2010-08-17]. History of SQL server. Dostupné z WWW:
<<http://dineshasanka.spaces.live.com/blog/cns!22A79FCE82651673!1347.entry>>.
- [16] *Microsoft : Overview* [online]. 2009 [cit. 2010-08-17]. Microsoft SQL Server 2005. Dostupné z WWW:
<<http://www.microsoft.com/sqlserver/2005/en/us/overview.aspx>>.
- [17] PLAVEC, Petr. *Projekt návrhu a tvorby webového informačního systému pro správu a optimalizaci finančních portfolií*. Pardubice, 2010. 56 s. Diplomová práce. Univerzita Pardubice, FEI, Katedra softwarových technologií.
- [18] *VMware* [online]. 2009 [cit. 2010-08-17]. Workstation and ACE Online Library. Dostupné z WWW:
<http://pubs.vmware.com/ws7_ace26/wwhelp/wwhimpl/js/html/wwhelp.htm?context=ws_user&file=intro_sysreqs_ws.html>.

Přílohy

Příloha č. 1 – Obsah příloženého CD

Na příloženém CD jsou vytvořené složky: *Datová část*, *Diplomová práce*, *Frameworky*, *Obrázky*, *Pojistný matematický vzorec*, *Případy užití* a *Web Part*.

Složka Datová část

Obsahuje hierarchický datový model realizovaný v MOSS pomocí listů a jejich sloupců, který je uložen v souboru *Hierarchický datový model SharePoint.png*.

Složka Diplomová práce

Obsahuje dokument diplomové práce uložený v souboru *Diplomová práce – Bělohlávek Petr.pdf*.

Složka Frameworky

Obsahuje dokument zkoumaných frameworků pro implementaci informačního systému. Dokument je uložen v souboru *Frameworky.pdf*.

Složka Obrázky

Obsahuje všechny obrázky, které byly použity v této diplomové práci.

Složka Pojistný matematický vzorec

Obsahuje dokument, ve kterém je upravován matematický vzorec pro nasazení do MOSS. Dokument je uložen v souboru *Vzorec Hodnota konec.pdf*.

Složka Případy užití

Obsahuje dokumenty ve formátu *PDF*, které popisují chování systému pomocí vytvořených specifikací případů užití.

Složka Web Part

Obsahuje realizovanou Web Part vytvořenou ve vývojovém prostředí Visual Studio 2008.