

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

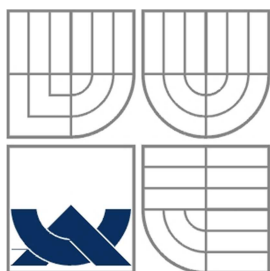
PORTÁL PRO PRODEJ ZVUKOVÝCH NAHRÁVEK
A VIDEÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

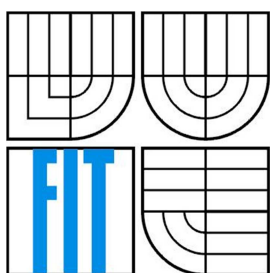
AUTOR PRÁCE
AUTHOR

JAKUB RANDA

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PORTÁL PRO PRODEJ ZVUKOVÝCH NAHRÁVEK A VIDEÍ

ONLINE MEDIA SHOPPING SYSTÉM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB RANDA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LADISLAV RUTTKAY

BRNO 2011

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací portálu, který umožňuje nákup či výpůjčku zvukových a video nahrávek a jejich distribuci ke koncovému uživateli formou streamingu. Popisuje použité návrhové technologie jazyka UML, použité vývojové technologie a technologie použité při realizaci streamingu. Obsahuje popis vlastní implementace aplikace a jejího nasazení do provozu.

Abstract

This bachelor's thesis covers designing and developing of the web application, which allows to buy or rent the multimedia records and then watch them using the streaming form of distribution. It describes designing, developing and streaming technologies, which were used during the process of realization. It contains the description of implementing and publishing of the application.

Klíčová slova

Informační portál, webová aplikace, multimédia, objektově orientované programování, třívrstvá architektura, ASP.NET, MS SQL, MS Silverlight, IIS Smooth Streaming, UML, XHTML, CSS

Keywords

information portal, web application, multimedia, object-oriented programming, three-layer architecture, ASP.NET, MS SQL, MS Silverlight, IIS Smooth Streaming, UML, XHTML, CSS

Citace

Randa Jakub: Portál pro prodej zvukových nahrávek a videí, bakalářská práce, Brno, FIT VUT v Brně, 2011

Portál pro prodej zvukových nahrávek a videí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Ladislava Ruttkaye.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jakub Randa
17.05.2011

Poděkování

Děkuji panu Ing. Ruttkayi za odborné vedení, projevenou ochotu a trpělivost a poskytnutí cenných rad a připomínek v průběhu tvorby této práce.

© Jakub Randa, 2011

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

Obsah

Obsah	1
1. Úvod	3
2. Analýza řešení.....	4
2.1 Dostupná řešení.....	4
2.1.1 Youtube.com.....	4
2.1.2 Hulu.com.....	4
2.1.3 Apple iTunes.....	4
2.1.4 Amazon.com.....	5
2.1.5 Last.fm.....	5
2.1.6 Situace na českém trhu	5
2.1.7 Topfun.cz.....	6
2.2 Specifikace cílů práce	6
2.2.2 Technické a právní možnosti prodeje	7
2.2.3 Platba pomocí platební karty	7
3. Použité technologie.....	9
3.1 XHTML, CSS	9
3.2 UML	9
3.3 Microsoft .NET framework	10
3.4 ASP.NET	11
3.5 Microsoft Silverlight.....	11
3.6 Použité nástroje.....	12
3.6.1 Microsoft Visual Studio 2010.....	12
3.6.2 Visual Paradigm for UML 8.2	12
4. Návrh portálu	13
4.1 Model případů užití.....	13
4.1.1 Diagram případů užití	13

4.1.2 Detail případu užití	14
4.2 Databázový model	15
4.2.1 E-R diagram	16
4.3 Diagramy tříd	17
5. Implementace portálu	18
5.1 Objektově orientovaný přístup	18
5.2 Třívrstvá architektura	18
5.2.1 Databáze	20
5.2.2 Datová vrstva	22
5.2.3 Aplikační vrstva	24
5.2.4 Prezentační vrstva	25
5.3 Implementace rolí	28
5.3.1 Uživatelská část	28
5.3.2 Správcovská část	29
5.4 Technické řešení streamingu	30
5.4.1 IIS Smooth Streaming	31
5.4.2 Příprava streamu a jeho distribuce	32
5.4.3 Silverlight klient	34
5.4.4 DRM ochrana	35
5.5 Nasazení do provozu a testování	36
6. Závěr	37
6.1 Přínos práce	37
6.2 Možná rozšíření práce	38
Použitá literatura	39
Obsah příloženého DVD	41

1. Úvod

Hudební a kinematografický průmysl se rozvíjí již po řadu desetiletí. Autoři svá díla zaznamenávají na různé typy nosičů, z jejichž prodeje poté mají určitý zisk. Od svého vzniku se tato odvětví neustále vyvíjejí a spolu s nimi také technické možnosti reprezentace těchto děl.

Masové rozšíření digitálního záznamu informací odstartovalo novou éru i v oblasti záznamu hudby a videa. Digitální podoby těchto děl existují již značné množství let a v současnosti tvoří prakticky jedinou možnost distribuce, když digitální nosiče (CD, DVD, internetové formy přenosu) téměř vytlačily klasické analogové nosiče (magnetická páska, gramofonová deska), které se využívají spíše pro speciální účely. Spolu se změnou nosičů jde ruku v ruce i forma jejich distribuce.

Tato bakalářská práce se zabývá jednou z nejnovějších forem distribuce digitálního audio-vizuálního materiálu – streamingu. Nasazení této technologie začalo být možné spolu s rozšířením vysokorychlostních připojení k internetu, tzv. broadband připojení. Oproti původním běžně rozšířeným druhům připojení k internetu, která byla značně pomalá a pro větší objemy dat nevyhovující, již tato připojení poskytují dostatečnou rychlost a kvalitu k přenosu audio a video datových toků v reálném čase. Tato metoda přináší koncovému uživateli značně pohodlný způsob prezentace těchto dat, kdy například nepotřebuje pro přehrání nějakého filmu žádné fyzické médium, které zabírá místo při skladování (ať už fyzický prostor u optických nosičů CD či DVD, tak bytový prostor v případě uložení těchto dat na pevný disk počítače). Zároveň je tento způsob významně méně logisticky náročný, protože odpadá nutnost přepravy fyzického média k uživateli.

Tento projekt využívá výhod internetového prostředí v oblasti prodeje a půjčování hudebních a kinematografických děl. Proces vypůjčení díla je realizován prostřednictvím zpřístupnění streamového toku koncovému uživateli po dohodnutou dobu. Platba za zprostředkování díla je realizována pomocí platby kreditní kartou, případně bankovním převodem na účet majitele systému.

Práci jsem rozdělil do šesti kapitol. Po tomto úvodu následuje kapitola Analýza řešení, kde se zabývám aktuální situací na trhu a specifikací cílů mé práce. V následující kapitole popisují technologie, které jsem využil pro tvorbu výsledné aplikace, a nástroje, ve kterých byly tyto technologie realizovány. Čtvrtá kapitola obsahuje postup návrhu jednotlivých součástí aplikace. V páté kapitole pak popisují samotnou implementaci portálu a streamingu. Závěrečná kapitola shrnuje dosažené výsledky a diskutuje možné budoucí rozšíření práce.

2. Analýza řešení

2.1 Dostupná řešení

Ve světě existuje již obrovské množství funkčních portálů nabízejících prodej a vypůjčování hudebních a video materiálů prostřednictvím streamingu. Ve své práci se pokusím uvést nejúspěšnější řešení z jednotlivých kategorií a shrnout jejich vlastnosti. Uvedu také některá řešení, které nenabízejí přímo vypůjčování streamu za poplatek, ale poskytují je zdarma. Některá z těchto řešení totiž mají zásadní význam pro prosazení virtuálních půjčoven a prodejen. Na závěr zhodnotím situaci na českém trhu, která je oproti zahraničním trhům v mnoha ohledech značně specifická.

2.1.1 Youtube.com

Server youtube.com společnosti YouTube, LLC je v současné době nejnavštěvovanějším serverem poskytujícím streamovaný obsah. Tato služba nespadá pod diskutovaný problém, nabízí převážně možnost nahrání a streamingu uživatelských videí bez poplatků. Uvádím ji zde proto, že server youtube.com měl v posledních letech zásadní podíl na masovém rozšíření streamovaného obsahu přes internet. V prosinci roku 2008 navíc začal tento server nabízet streamování některých videí v HD kvalitě a spolu s tímto krokem začal nabízet i streamování filmů v této kvalitě. V současnosti se spekuluje o možnosti, že i tento server nabídne možnost půjčování audio-vizuálního obsahu za poplatek, což by vzhledem k jeho obrovské popularitě mohlo značným způsobem ovlivnit situaci na trhu. [1]

2.1.2 Hulu.com

Druhým a zároveň posledním zástupcem služeb, které nenabízejí vypůjčování streamovaného obsahu za poplatek a kterého zde uvedu, je server hulu.com společnosti Hulu LLC. Tato služba pracuje na odlišném způsobu od služby youtube.com. Neumožňuje nahrávání uživatelských videí, oproti tomu nabízí streamování některých pořadů amerických televizních společností, zejména pak seriálů. K dispozici je vždy s určitým zpožděním několik posledních dílů těchto pořadů v závislosti na smlouvě s poskytovatelem, většinou se jedná maximálně o 10 dílů. Služba je dostupná zadarmo, od června 2010 však nabízí možnost předplaceného prémiového účtu s bonusovým obsahem. [2]

2.1.3 Apple iTunes

Portál iTunes společnosti Apple Inc. patří k největším portálům, které umožňují jak prodej, tak vypůjčování hudebních materiálů, filmů i televizních pořadů. Využívá především obrovské popularity přístrojů společnosti Apple Inc. v některých státech, pro které je tato služba nativní. Služba však není omezená pouze na MAC platformu, je k dispozici i pro platformy jiné, včetně

PC a operačního systému Windows. Kromě audio-vizuálních materiálů nabízí velké množství dalších digitálních souborů, zejména pak aplikací. Některé materiály jsou k dispozici bez poplatku, ostatní je možno si zakoupit a tam, kde to je možné, je nabízeno i jejich vypůjčování.

2.1.4 Amazon.com

Server amazon.com společnosti Amazon.com, Inc. patří k největším internetovým obchodům na světě. Není specializovaný na nákup či vypůjčování materiálů formou streamu či downloadu, ale na nákup zboží ve fyzické podobě a jeho dopravě k zákazníkovi pomocí přepravní společnosti. Od roku 2006 však tento portál nabízí také službu „Amazon Video On Demand“, která umožňuje vypůjčování některých prodávaných filmů. Platí se za každý titul zvlášť, každý titul má také vlastní výpůjční dobu, většinou 24 či 48 hodin. Zaplacený materiál je pak možné spustit jako stream nebo si ho na omezenou dobu stáhnout do počítače ve formě video souboru.

2.1.5 Last.fm

Server last.fm společnosti Last.fm Ltd. se specializuje na hudební streamy. Funguje na zcela jiném principu než předchozí zmiňované služby. Služba last.fm neumožňuje prodej nahrávek a nabízí zajímavou koncepci jejich vypůjčování. Skládá se z databáze jednotlivých hudebních nahrávek, která je opatřena množstvím tagů ke každé skladbě. Uživatel si může při zvolení konkrétní skladby poslechnout pouze 30 sekundovou ukázkou. Server však umožňuje vysílání hudebních streamů podle uživatelových požadavků. Tyto požadavky služba může zjistit podle databáze skladeb, které si již uživatel přehrál, nebo podle jména interpreta, který uživatel zadá. Služba pak pomocí tagů u jednotlivých skladeb vybírá podobné skladby od podobných interpretů a vysílá je pomocí streamu k uživateli. Uživatel však nemůže ovlivnit, které konkrétní skladby mu služba vybere. Z tohoto důvodu je tato služba klasifikována spíše jako internetové rádio, díky čemuž je možné ji provozovat pouze za mírný poplatek. Obdobně jako server hulu.com i server last.fm dříve nabízel tuto službu bez poplatku, zpoplatnil ji až dne 22. dubna 2009 pro uživatele mimo USA, Velkou Británii a Německo. [3]

2.1.6 Situace na českém trhu

Na českém trhu se tato forma služeb zatím příliš neprosazuje. Svůj podíl na tom má bezesporu vysoká míra počítačového pirátství v České republice, ale také fakt, že společnostem, které se zde pokoušely o provozování těchto portalů, se většinou nedařilo uzavírat smlouvy s velkými a důležitými společnostmi, které by poskytovaly obsah. Jediným portálem, kterému se tato podmínka daří splňovat, je server topfun.cz, který rozeberu v další podkapitole. Zajímavým projektem byla Videopůjčovna České televize, která nabízela k vypůjčení streamy některých svých pořadů, které nemohly být nabízeny k vysílání bez poplatků, zejména pak tedy filmů a seriálů. Některé své pořady pak nabízela za poplatek v předpremiéře oproti jejich vysílání v televizní síti. Tato služby začala fungovat 17. prosince 2007, pro svoji prodělečnost však byla

ukončena o 2 roky později, 17. prosince 2009. [4] Alespoň částečné popularity se v České republice těší služby televizních poskytovatelů, které nabízejí možnosti streamování určitých materiálů za měsíční poplatek, obdobně jako služba netflix.com. Množství nabízených materiálů je však většinou malé a nepříliš atraktivní, navíc se nejedná o druh prodeje, který je diskutován v této práci.

2.1.7 Topfun.cz

Portál topfun.cz společnosti Divida, s.r.o. je v současné době největším poskytovatelem těchto služeb v České republice. Tento portál má uzavřeny smlouvy s některými významnějšími poskytovateli, což zaručuje relativně atraktivní obsah. Portál umožňuje vypůjčování a nákup filmových materiálů, nepodporuje hudební soubory. Obdobně jako u serveru amazon.com se pak za každý vypůjčený titul platí zvlášť a je možné si ho prohlédnout buďto pomocí streamu, nebo prostřednictvím stáhnutí souboru na počítač. Vypůjční doba je zde 24 hodin a kromě platby kartou je možné platit také prostřednictvím zaslání SMS a následného obdržení kódu. Nelze však platit prostřednictvím dobíjených kreditů (např. formou bankovního převodu), portál také neposkytuje nákup ani vypůjčení audio materiálů.

2.2 Specifikace cílů práce

Informační portál bude zprostředkovávat koupi či pronájem zvukových a obrazových děl pomocí streamingu. Uživatel bude moci zaplatit objednávku platební kartou, nebo si bankovním převodem zakoupit kredity vázané k jeho uživatelskému účtu, které poté bude moci použít pro platbu objednávek. Po zaplacení mu budou vygenerovány streamy, jež bude moci spouštět ze svého uživatelského účtu. V případě výpůjček bude tento stream zpřístupněn po dobu 48 hodin, poté se pro uživatele zneaktivní a již nepůjde přehrát.

Celý systém se bude skládat ze serverové aplikace a klientské aplikace. Na straně serveru bude kromě zajištění komunikace s klientem uložena databáze uživatelů a audio-vizuálních děl, server také bude zařizovat vysílání streamů. Klientská část pak bude sloužit pro zadávání požadavků uživatele a pro zobrazování jednotlivých streamů vysílaných serverem.

Aplikace bude dále poskytovat administrativní prostředí pro správce a provozovatele portálu. K dispozici budou informace o prodeji, evidovaných zákaznících, vygenerovaných streamech a příchozích požadavcích na dobítí kreditu uživatelských účtů. Zejména na poslední jmenovanou vlastnost pak bude kladeny speciální požadavky, neboť kredity nakoupené za převod peněz pomocí bankovních účtů nemohou být připisovány automaticky. U těchto transakcí musí být nejdříve externě zkontrolovány příchozí platby na bankovním účtu a teprve poté může administrátor tyto kredity připsat na uživatelský účet.

2.2.2 Technické a právní možnosti prodeje

V současné době se prosazují především dva způsoby elektronického prodeje nahrávek. Prvním z nich je umožnění stažení zakoupeného souboru na uživatelský počítač. Většinou se jedná o nějaký druh komprimovaného souboru, u hudebních souborů v drtivé většině formátu MPEG-2 Audio Layer III (.mp3), u souborů s videem se pak jedná o formát Audio Video Interleave (.avi) zkomprimovaným patřičným kodekem, případně pak o formát Windows Media Video (.wmv). Z pohledu této práce zajímavějším způsobem je však distribuce pomocí tzv. streamu, tedy datového toku těchto souborů ze strany serveru na klientskou část, která tento tok zpracuje a výsledek zpřístupní koncovému uživateli. Při použití tohoto způsobu se soubor neukládá na uživatelův počítač, nýbrž je přehráván přímo ze serveru a na uživatelské straně je ukládána vždy jen jeho část do vyrovnávací paměti. Tento způsob umožňuje lepší kontrolu nad akcemi, které budou se souborem prováděny, protože při uložení souboru na uživatelský počítač poskytovatel prakticky ztrácí možnost jakékoli kontroly akcí, které s tímto souborem uživatel provede. Toto je zásadní fakt pro poskytovatele obsahu, kteří získávají větší garance ochrany poskytovaných nahrávek.

2.2.3 Platba pomocí platební karty

Důležitou součástí portálů, které nabízejí okamžitý prodej svých produktů, je implementace platby pomocí platební karty. Pro zpřístupnění těchto plateb existují na českém trhu 2 hlavní možnosti:

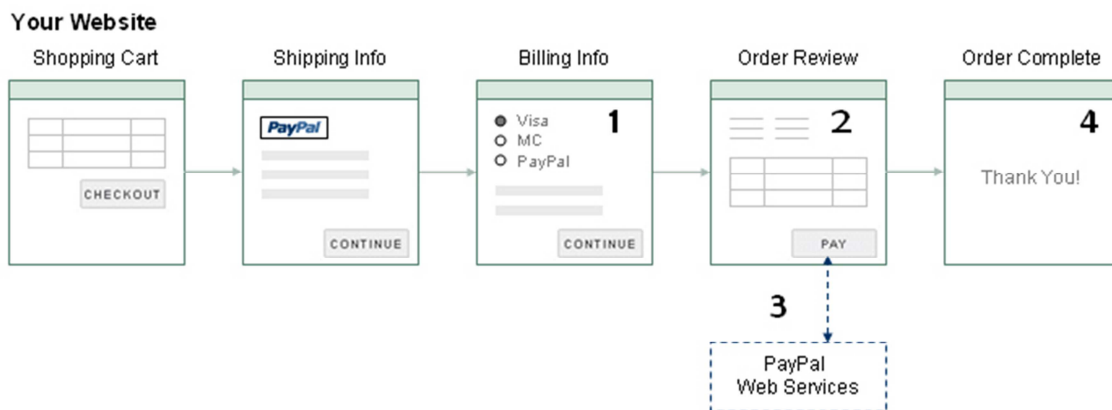
- Uzavření smlouvy s některou z českých bank, které tyto služby poskytují (např. Komerční banka, Česká spořitelna)
- Uzavření smlouvy s některou finanční firmou, které tyto služby poskytují (např. PayPal, Pay MUZO)

Každá z těchto institucí po uzavření smlouvy poskytne vývojáři své vlastní rozhraní pro programování aplikací (API – Application Programming Interface), skrze které je pak možné provést platbu. Finanční podmínky těchto služeb si opět určuje samotný poskytovatel. Jako příklad může posloužit služba PLATBA 24 České spořitelny [5], u které si poskytovatel účtuje poplatek za zpřístupnění služby, dále měsíční poplatek za provozování služby a také poplatek z každé provedené transakce.

Volba konkrétního poskytovatele je pak dána hlavně rozhodnutím provozovatele portálu. Protože jsou však poplatky za zpřístupnění těchto služeb včetně jejich vývojového rozhraní pro účely této práce příliš nákladné, je potřeba aplikaci pro nějaký tento systém pouze připravit a případnou implementaci konkrétního řešení provést až po nalezení provozovatele portálu.

Pro účely demonstrace platby kartou v této práci jsem se rozhodl použít principů systému Website Payments Pro od společnosti PayPal. Tento systém využívá pro samotnou platbu webové služby na serveru poskytovatele, která se zavolá spolu s parametry získaných na straně

vlastního informačního portálu. Služba poté provede samotnou platbu a vrátí její výsledek, který se opět na straně vlastního portálu zpracuje. Celý postup znázorňuje obrázek 2.1. Výhodou této služby je, že zákazník vůbec neopustí prostředí aplikace, ve které provádí nákup, celá platba proběhne pouze prostřednictvím kódu na pozadí stránky. [6]



Obr 2.1: Použití služby *Website Payments Pro* od společnosti *PayPal* (dostupný na ¹)

Simulaci platby provedu vytvořením formulářů pro získání potřebných informací o platební kartě. Jakmile uživatel vydá požadavek na zaplacení, pošlu tyto informace vlastní zástupné metodě (v reálné implementaci bych místo toho volal metodu poskytnutého API třetí strany), která bude simulovat úspěšnou platbu, a tu poté v aplikaci zpracuji.

¹ (2.5.2011) https://cms.paypal.com/cms_content/US/en_US/images/developer/ECFlow_DCC_new.jpg

3. Použité technologie

3.1 XHTML, CSS

Pro definici vzhledu a rozmístění ovládacích prvků webové stránky aplikace na straně klienta používám značkovací jazyk XHTML (eXtensible HyperText Markup Language) a kaskádové styly CSS (Cascading Style Sheets).

Jazyk XHTML je novější derivací jazyka HTML (HyperText Markup Language). Tyto jazyky jsou nejrozšířenějším prostředkem používaným pro popis hypertextových dokumentů. Zatímco jazyk HTML je aplikací jazyka SGML (Standard Generalized Markup Language), mnou používaný XHTML je aplikací novějšího XML (eXtensible Markup Language). Při vytváření jazyka XHTML bylo myšleno na co největší syntaktickou podobnost s jeho předchůdcem, hlavní změnou jsou pak tedy obecné vlastnosti jazyků, na které jsou tyto popisovací jazyky aplikovány. Mezi nejzásadnější změny tedy patří: [7]

- XHTML je striktní, jeden konkrétní úkon je možné popsat pouze jedním možným syntaktickým způsobem – tato vlastnost značně usnadňuje validaci dokumentu
- XHTML nepodporuje nepárové značky, některé nepárové značky z původního HTML je však možné zapsat zkráceným způsobem (např. `
`)
- Všechny značky musí být napsány malými písmeny
- Je zakázáno křížení značek
- Atributy jednotlivých značek musí být uvedeny v uvozovkách
- Byly odstraněny značky jazyka HTML určené pro definici vzhledu, jejich funkcionalitu je možné suplovat pomocí CSS stylů

Tabulky kaskádových stylů CSS poté slouží pro popis vzhledu jednotlivých elementů značkovacích jazyků. Nejčastější způsob jejich implementace spočívá v uložení do externího souboru, který se pak do jednotlivých XHTML dokumentů příkazem importuje. Výhoda tohoto způsobu spočívá ve sdílení těchto tabulek mezi více dokumenty, a tedy dosažení shodného vzhledu jednotlivých prvků napříč celým webovým systémem. [7]

3.2 UML

UML (Unified Modeling Language) je prostředek sloužící převážně k návrhu a modelování objektově orientovaných softwarových systémů. Jako průmyslový standard existuje od roku 1997, jeho kořeny však sahají hlouběji do minulosti. Jeho vznik byl podpořen snahou sjednotit existující postupy v oblasti modelovacích technik a softwarového inženýrství. [8]

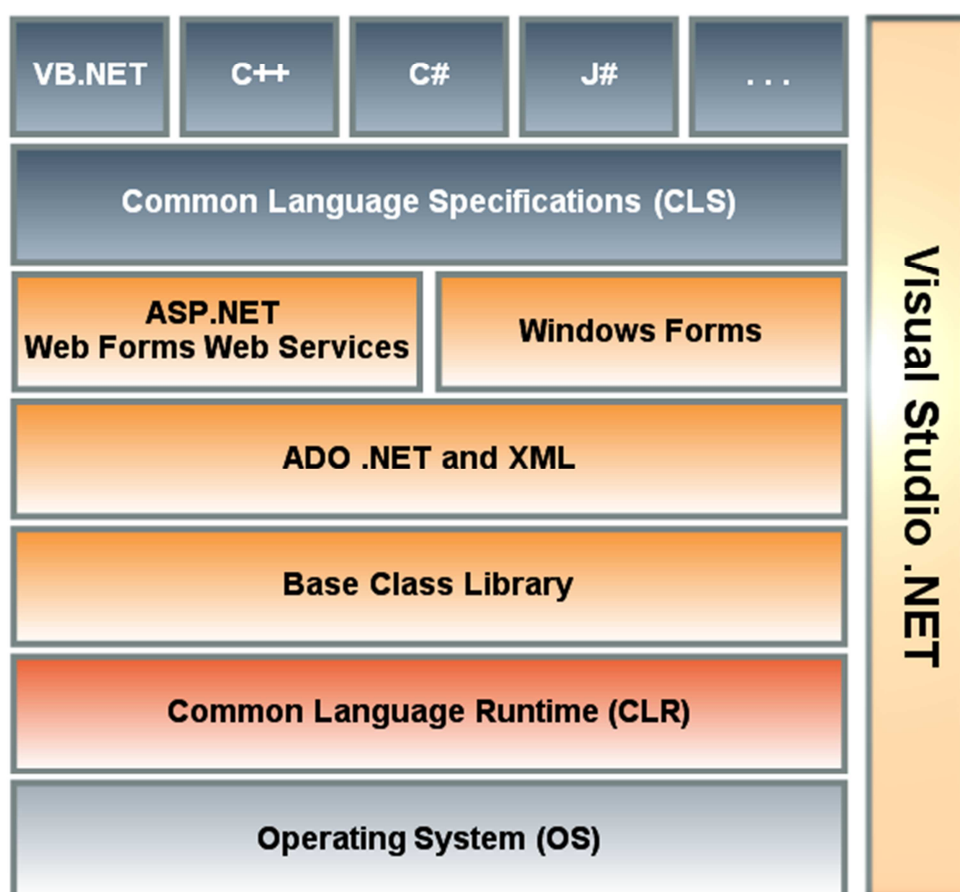
Jazyk UML se, na rozdíl od ostatních vývojových technologií zmiňovaných v této kapitole, vyznačuje grafickou syntaxí. Jeho reprezentací jsou tedy různé diagramy a grafické

značky. Jazyk je navržen tak, aby byl srozumitelný pro lidi, ale také snadno zpracovatelný tzv. CASE (computer-aided software engineering) programy, které podporují modelování a návrh aplikací.

Z možností jazyka UML využívám ve své práci především Model případů užití (Use-Case) a diagram tříd. Dále jsem také vytvořil databázový model realizovaný Entity-Relationship diagramem.

Jednotlivé body z tohoto seznamu jsou popsány v patřičných kapitolách této práce spolu s konkrétní ukázkou jejich realizace.

3.3 Microsoft .NET framework



Obr 3.1: *Architektura Microsoft .NET Framework (dostupný online na²)*

² (2.5.2011) <http://www.nevron.com/gallery/FullGalleries/diagram/miscellaneous/images/dotNetFramework.png>

Na obrázku 3.1 je znázorněna struktura Microsoft .NET Frameworku, na kterém je tato aplikace postavena. Základní myšlenkou architektury je poskytnutí běhového prostředí a sady knihoven, které toto prostředí využívá.

Šedá vrstva na obrázku znázorňuje nejvyšší úroveň architektury, a to jazyky, ve kterých se vyvíjí samotná aplikace. Podporované jazyky musí splňovat specifikace obsažené v CLS vrstvě tak, aby pomocí nich bylo možno přistupovat ke knihovnám, které jsou na obrázku vyznačeny oranžově. Z těchto knihoven stojí za zmínku zejména knihovny pro podporu webových aplikací (ASP.NET) či grafických uživatelských aplikací (např. WinForms), dále knihovny pro práci s daty (ADO.NET) a XML soubory a v neposlední řadě tzv. Base Class Library, obsahující prostředky umožňující základní funkcionalitu, například vstupně-výstupní operace. Výsledný program se poté předkompiluje do tzv. Intermediate Language, což je platformově nezávislý jazyk podobný assembleru.

Samotné vykonání předkompilovaného kódu má poté na starosti běhové prostředí CLR (na obrázku vyznačeno červeně), které tento kód zkompiluje do nativního strojového kódu procesoru, na kterém běží. Tento způsob dvojí kompilace umožňuje snadnou platformovou přenositelnost i vývojářskou nezávislost, kdy se shodný přechodný kód v Intermediate Language vygeneruje bez ohledu na použitý programátorský jazyk a platformu, na které aplikace poběží. [9]

3.4 ASP.NET

Z předchozí podkapitoly plyne, že ASP.NET je součástí Microsoft .NET Frameworku. Je to nástupce technologie ASP (Active Server Pages), která byla však od základu předělána tak, aby mohla využívat možnosti .NET Frameworku, které nyní nabízí spolu s původními prostředky pro tvorbu dynamických webových stránek.

ASP.NET obsahuje značné množství ovládacích prvků, které je možné umístit na vytvářenou stránku. Tyto ovládací prvky jsou pak vytvářeny na straně serveru a spouštějí vlastní události, které jsou taktéž na serverové straně obsluhovány. Jako všechny aplikace napsané v .NET Frameworku, i kód těchto aplikací je nejdříve předkompilován do Intermediate Language a poté zpracován pomocí CLR, který také běží na serverové straně. [10]

3.5 Microsoft Silverlight

Silverlight je nová technologie společnosti Microsoft, která si klade za cíl rozšíření možností, které poskytuje kód na straně klienta. Na klientském zařízení se instaluje jako zásuvný modul (plug-in) do internetového prohlížeče, kde poté poskytuje část tříd Microsoft .NET Frameworku. Na straně klienta je poté možné například vykreslit 2D grafiku či animaci, a co je podstatné z hlediska této práce, Silverlight nabízí i možnosti pro přehrávání multimediálního obsahu. [10]

3.6 Použité nástroje

3.6.1 Microsoft Visual Studio 2010

Integrované vývojové prostředí Visual Studio 2010 je primárně určené k tvorbě aplikací založených na .NET Frameworku, ať už se jedná o webové aplikace využívající ASP.NET, nebo například grafické uživatelské aplikace či konzolové aplikace. Jeho součástí je například systém IntelliSense, který automaticky v průběhu psaní kódu nabízí vhodné varianty pro doplnění právě psaného výrazu, dále integrovaný debugger, který umožňuje ladění aplikace, a nebo prostředí pro připojení se k databázovému serveru, které umožňuje provádění základních operací s tímto připojením.

3.6.2 Visual Paradigm for UML 8.2

Tento software od firmy Visual Paradigm International poskytuje řadu nástrojů pro návrh a analýzu projektů a databází. Využívá prostředky jazyka UML ve verzi 2.1 a přidává k nim několik vlastních nástrojů, zejména pak pro realizaci týmové spolupráce. Z jeho možností v této práci využívám zejména modelování diagramu případů použití a modelování databáze pomocí E-R diagramu. Program je pro nekomerční použití k dispozici zadarmo (dostupné na ³), pro využití v komerčních aplikacích je pak možné zakoupit jednu z jeho čtyř různých verzí.

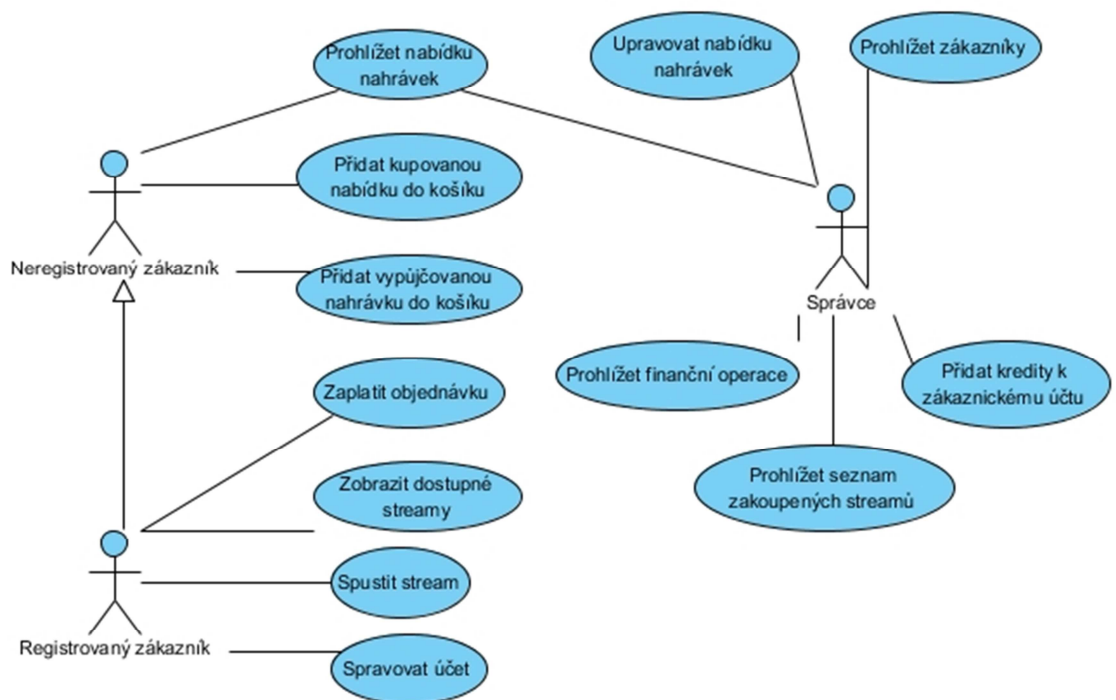
³ <http://www.visual-paradigm.com/product/vpuml/>

4. Návrh portálu

4.1 Model případů užití

Model případů užití, neboli Use-case model, graficky zobrazuje systém z pohledu uživatelů a akcí, které budou moci provádět. Představuje jeden z nejzásadnějších prostředků návrhu aplikace pomocí jazyka UML. Základní komponenty používané při modelování jsou „účastníci“ (actors), jež představují role jednotlivých uživatelů používajících modelovaný systém, dále „případy užití“ (use cases), což jsou činnosti, které mohou účastníci v systému vykonávat, a „relace“ (relationships), které znázorňují vztahy mezi účastníky a případy užití. Výstupem tohoto modelu je diagram případů užití. [8]

4.1.1 Diagram případů užití



Obr 4.1 : Use-case diagram

Diagram modeluje 3 účastnické role:

- *Neregistrovaný zákazník* – standardní uživatel bez vytvořeného uživatelského účtu, kterému bude umožněna základní práce s nabídkou nahrávek

- **Registrovaný zákazník** – uživatel s vytvořeným uživatelským účtem, oproti neregistrovanému zákazníkovi může navíc provádět operace vázané na jeho účet a zakupovat další nahrávky
- **Správce** – administrátor portálu, který má přístup k funkcím potřebným pro správu portálu a pro zobrazení statistik vzniklých z prodeje nahrávek

Dále jsou modelovány jednotlivé případy užití, ke kterým mají jednotliví účastníci přístup podle relací znázorněných na use-case diagram (viz obrázek 4.1):

- **Prohlížet nabídku nahrávek** – umožňuje prohlížení dostupných audio a video nahrávek, jejich filtraci podle kategorií a vyhledávání podle jména v těchto seznamech
- **Přidat kupovanou nabídku do košíku** – provede přidání do seznamu nakupovaných nahrávek v košíku, přes který poté bude možno objednávku vyřídit
- **Přidat vypůjčovanou nahrávku do košíku** – provede přidání do seznamu vypůjčovaných nahrávek v košíku, přes který poté bude možno objednávku vyřídit
- **Zaplatit objednávku** – dokončí vyřízení objednávky, zkontroluje proběhlou platbu a jako produkt objednávky vygeneruje patřičné streamy
- **Zobrazit dostupné streamy** – zobrazí seznam platných zakoupených či vypůjčených streamů uživatele
- **Spustit stream** – přehraje vybraný stream
- **Spravovat účet** – umožní uživateli spravovat informace o svém účtu a dobýt kredity přes bankovní převod
- **Upravovat nabídku nahrávek** – umožní zásah do databáze nabízených nahrávek, zejména pak nabídne prostředky pro vložení nových nahrávek.
- **Prohlížet zákazníky** – zobrazí seznam registrovaných uživatelů
- **Prohlížet finanční operace** – zobrazí proběhlé platby zákazníků
- **Prohlížet seznam zakoupených streamů** – zobrazí seznam vygenerovaných streamů nakoupených zákazníky
- **Přidat kredity k zákaznickému účtu** – umožní připsat zakoupené kredity na základě bankovních převodů

4.1.2 Detail případu užití

Každý z případů užití je rozebrán ve svém detailu. Pro detail případů užití neexistuje UML standard, většinou se tedy zapisuje ve formě co nejpřehlednější tabulky. Každý detail případu užití má svoje identifikační číslo, název a poté informace, které k němu náleží, například vstupní podmínky nebo tok událostí. V tabulce Tab 4.2 je zobrazen příklad takového detailu, konkrétně pro případ užití Zobrazit dostupné streamy.

Případ užití: Zobrazit dostupné streamy
ID: UC5
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel je přihlášen přes svůj uživatelský účet
Tok událostí: <ol style="list-style-type: none"> 1. Uživatel zadá příkaz k zobrazení dostupných streamů 2. Systém nalezne všechny zakoupené streamy tohoto uživatele a zobrazí je 3. Systém nalezne všechny vypůjčené streamy tohoto uživatele, zkontroluje čas výpůjčky a pokud je výpůjčka stále platná, zobrazí tyto streamy 4. KDYŽ je seznam streamů neprázdný <ol style="list-style-type: none"> 4.1 Systém nabídne u zobrazených streamů možnost vybrání jednoho z nich pro spuštění přehrávání 4.2 Po vybrání streamu systém pokračuje detailem popsaným v UC6 (Spustit stream)
Následné podmínky: Systém zobrazil dostupné streamy pro konkrétního uživatele

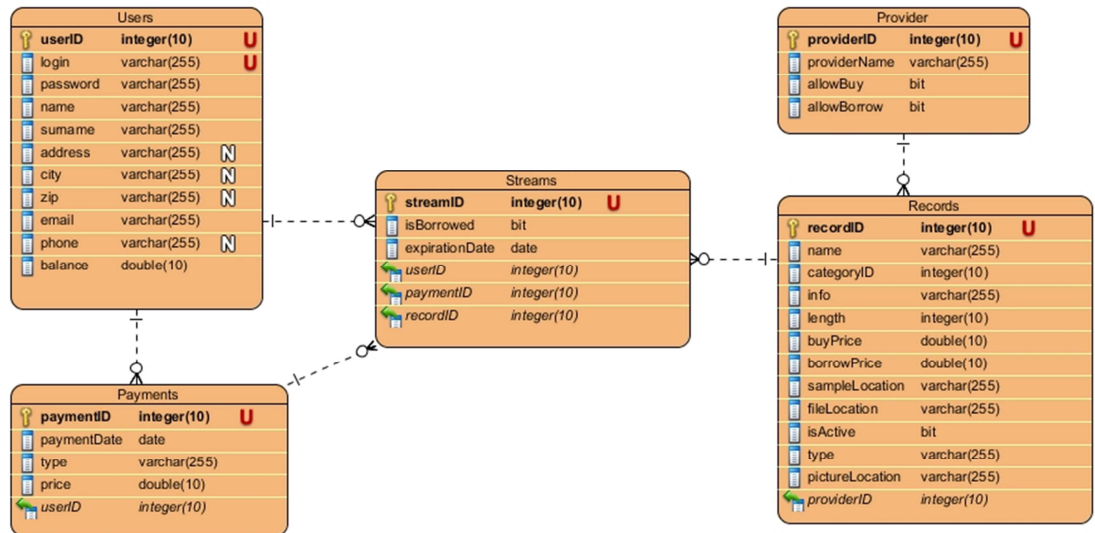
Tab 4.2: Detail případu užití

4.2 Databázový model

Databázový model zachycuje koncept dat, které bude portál potřebovat pro svoji práci. Tyto data budou uloženy v databázi, která se v tomto modelu navrhne.

Pro její vymodelování použijí tzv. Entity-Relationship diagram. Jeho základ tvoří jednotlivé entity, které symbolizují tabulky uložených dat a vztahy mezi těmito entitami. Každý prvek v entitě je poté identifikován svým unikátním primárním klíčem a vztahy mezi entitami jsou pak technicky řešeny přidáním cizího klíče do modelované entity. V následující kapitole je na obrázku 4.3 vidět E-R diagram mé aplikace a popis jednotlivých modelovaných entit.

4.2.1 E-R diagram



Obr 4.3: E-R diagram

E-R diagram modeluje 5 základních entit, se kterými bude databáze pracovat.

- **Users** – tabulka uživatelů, eviduje informace o zaregistrovaných uživateli.
- **Payments** – entita reprezentující provedené platby, jejich typ a hodnotu. Je vázána na uživatele, který platbu provedl.
- **Provider** – poskytovatel nahrávek, se kterým má společnost provozující portál uzavřenou smlouvu. Tato entita mimo jiné určuje, jestli poskytovatel bude umožňovat své nahrávky prodávat či půjčovat, případně bude nabízet oba způsoby zároveň.
- **Records** – entita jednotlivých nahrávek, obsahuje zobrazované informace o nahrávce, její cenu pro vypůjčení i zakoupení, lokace, na kterých je na serveru uložena samotná nahrávka, krátká ukázka nahrávky a obrázek, který nahrávku reprezentuje a její typ. Je vázána na entitu Provider.
- **Streams** – reprezentuje nakoupený stream a zaznamenává informace o vypůjční době u vypůjčených nahrávek. Cizími klíči je vázán na uživatele, jenž si tento stream zakoupil, dále na platbu, v rámci které byl zakoupen a na nahrávku, kterou tento stream reprezentuje.

4.3 Diagramy tříd

Diagram tříd představuje grafické znázornění objektů použitých v systému a vztahů mezi nimi. U jednotlivých objektů vyjadřuje jejich vlastnosti a metody a zobrazuje omezení, které vyplývají z jejich zapouzdření. [11]

Při návrhu aplikace jsem vytvořil dva diagramy tříd. První z nich je diagram tříd aplikační vrstvy, kde jednotlivé objekty reprezentují logiku samotné aplikace, druhým je pak diagram tříd datové vrstvy, kde jednotlivé objekty odpovídají logice databáze. Diagramy jsou k dispozici na přiloženém DVD.

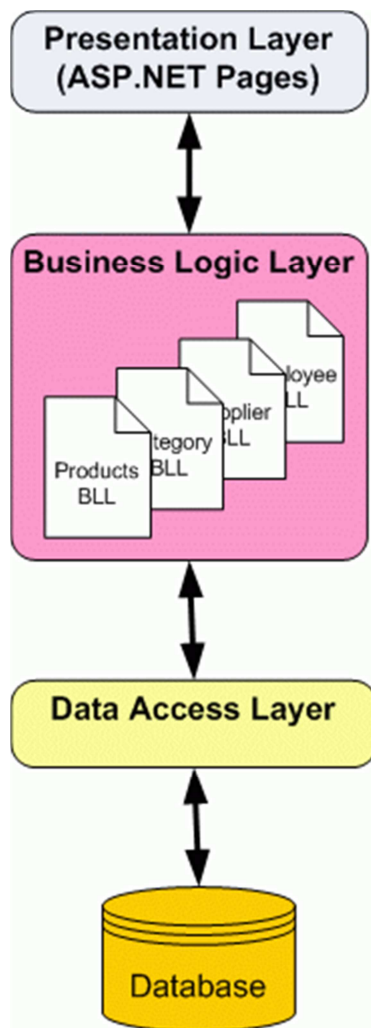
5. Implementace portálu

5.1 Objektově orientovaný přístup

Pro implementaci tohoto portálu byl zvolen objektově orientovaný přístup. Na rozdíl od staršího strukturového přístupu, který se na problém díval z globálního pohledu a aplikoval funkce, jenž tento problém řešily, objektový přístup dělí problém na samostatné logické celky, tzv. objekty, které spojí tyto funkce s daty nad kterými jsou vykonávány. Tyto objekty pak spolu komunikují pomocí metod. Díky principu zapouzdření, kdy se určí, které součásti objektu budou moci spravovat pouze instance tohoto objektu, a ke kterým budou mít přístup i ostatní objekty podle zadaných podmínek, se zaručuje větší přehlednost a bezpečnost návrhu. Tento způsob navíc lépe odpovídá lidskému pohledu na problém, což velmi zásadně usnadňuje návrh programu, zejména pak v týmovém prostředí. [12]

5.2 Třívrstvá architektura

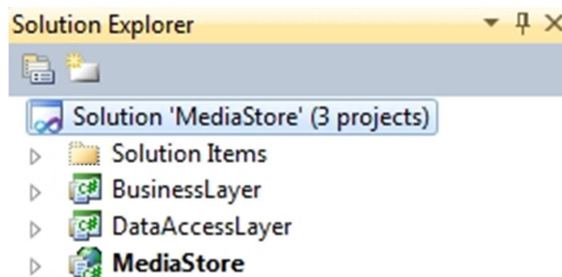
Třívrstvá architektura rozděluje aplikaci na tři na sobě nezávislé celky – prezentační vrstvu (Presentation Layer), aplikační vrstvu (Business Logic Layer) a datovou vrstvu (Data Access Layer), které budou podrobněji rozebrány v následujících podkapitolách. Při návrhu aplikace se domluví rozhraní, přes které budou mezi sebou jednotlivé vrstvy komunikovat. Poté již nic nebrání vývoji jednotlivých vrstev odděleně, čehož se dá s výhodou využít zejména u rozsáhlejších aplikací, které jsou vyvíjeny v týmech. Jednotlivé týmy mohou totiž pracovat na různých vrstvách aplikace nezávisle na sobě. Další nezanedbatelnou výhodou je větší znovupoužitelnost takto vyvíjené aplikace. Jelikož jsou jednotlivé vrstvy členěny podle technického účelu, je nutné v případě potřeby změny konfigurace přepsat pouze odpovídající vrstvu. Pokud tedy chci změnit databázový systém z Microsoft SQL Server na, řekněme, Oracle Database, upravím pouze patřičné části kódu v datové vrstvě, pokud budu potřebovat vizuální výstup programu namísto formou webových formulářů v konzoli, upravím vrstvu prezentační. Lehčí nevýhodou je pak mírný nárůst obsáhlosti kódu, kdy je potřeba implementovat i dohodnuté rozhraní mezi vrstvami.



Obr 5.1: *Třívrstvá architektura (dostupný online na⁴)*

Na obrázku 5.1 je znázorněno propojení mezi jednotlivými vrstvami a databází systému. Šipky zde znázorňují komunikaci mezi vrstvami. Jak je tedy z obrázku patrné, přímo mezi sebou mohou komunikovat pouze sousední vrstvy.

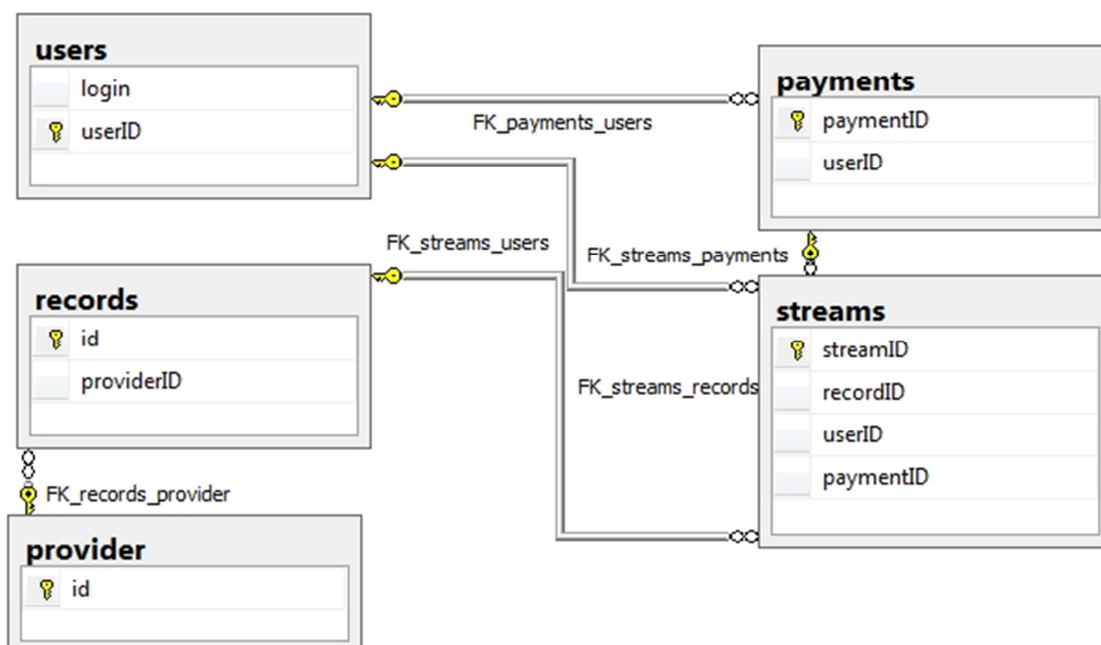
⁴ (2.5.2011) <http://i.msdn.microsoft.com/dynimg/IC132931.gif>



Obr 5.2: Implementace vrstev

5.2.1 Databáze

Pro uchování persistence aplikace je použito ukládání dat do databáze, v tomto případě konkrétně do SQL databáze Microsoft SQL Server 2008. Realizace databáze vychází z návrhového E-R diagramu, diskutovaného v kapitole 4.2.1. Jednotlivé entity jsou reprezentovány tabulkami, každá z nich pak kromě informací specifických pro danou entitu obsahuje primární klíč (ve většině případů identifikační číslo daného objektu), který je pro daný prvek jedinečný. Jedinečnost zaručuje vlastnost Microsoft SQL Serveru „IDENTITY“, kterou jsem nastavil u primárních klíčů a která při vložení nového prvku do tabulky zaručí vygenerování unikátního identifikátoru. Na obrázku 5.3 je k vidění diagram této databáze, kdy jsou pro přehlednost zobrazeny pouze primární a cizí klíče.



Obr 5.3: Schéma databáze s primárními a cizími klíči

Pro práci s daty se využívají dotazy nad databází, v tomto případě se pak konkrétně jedná o dotazy v jazyce T-SQL, se kterým Microsoft SQL Server pracuje. Jednotlivé dotazy jsou pak uloženy na serveru jako tzv. Stored Procedures (uložené procedury). Tento způsob uložení procedur urychluje jejich vyvolání, protože server si může podle potřeby tyto dotazy předpřipravit a poté je již jenom provede na základě předaných parametrů.

Při návrhu procedur byl kladen důraz na ochranu proti SQL injektáži (SQL Injection). SQL injektáž je druh internetového útoku, při kterém útočník využívá špatného přístupu k formování SQL dotazů, kdy k této formulaci dotazu programátor přistupuje pomocí spojování částí textových řetězců. V takovém případě může útočník vepsat do ovládacího prvku na napadené stránce část vlastního SQL dotazu, který může být potenciálně nebezpečný a způsobit chování databáze, které nebylo programátorem zamýšleno. Uložené procedury Microsoft SQL Serveru však nepřijímají své vstupy jako textové řetězce, ale jako parametry, které jsou odděleny od uloženého dotazu. V případě zadání škodlivého kódu pak zůstane vlastní struktura dotazu nezměněna. Na obrázku 5.4 je k nahlédnutí jedna z použitých uložených procedur.

```
ALTER PROCEDURE dbo.streams_insert
(
    @recordID      int,
    @userID        int,
    @paymentID     int,
    @expirationDate datetime,
    @isBorrowed   bit,
    @streamID      int OUTPUT
)
AS
INSERT INTO streams (recordID, userID, paymentID, expirationDate, isBorrowed)
VALUES (@recordID, @userID, @paymentID, @expirationDate, @isBorrowed);

SET @streamID = @@IDENTITY
```

Obr 5.4: Uložená procedura

Pro připojení databáze k projektu se využívá připojovacího řetězce (Connection string). Tento řetězec je uložen v souboru *web.config*, ve kterém se uchovává nastavení aplikace. Po spuštění aplikace si server převezme informace právě z tohoto souboru a zařídí připojení k databázi. Výhoda tohoto způsobu spočívá v jednoduchém přepsání připojovacího řetězce v případě instalace na jiný server, kdy se změní pouze toto nastavení v konfiguračním souboru bez nutnosti zásahu do jiných částí aplikace.

```

<connectionStrings>
  <add name="store"
    connectionString="Data Source=PC_JAKUB;Initial Catalog=store;Integrated Security=True"
    providerName="System.Data.SqlClient" />
</connectionStrings>

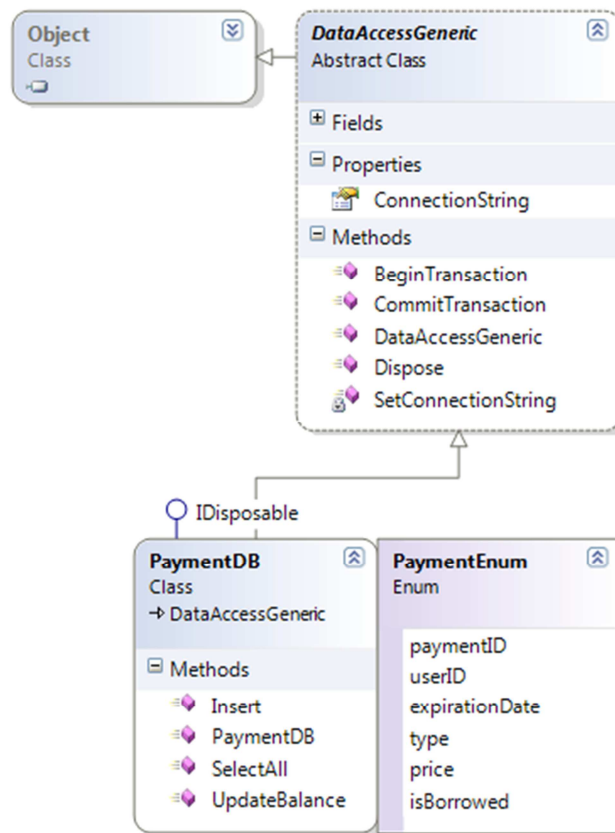
```

Obr 5.5: Připojovací řetězec

5.2.2 Datová vrstva

Datová vrstva (Data Access Layer) zajišťuje komunikaci aplikace s databází, jako jediná ze všech vrstev má právě ona přístup k tomuto zdroji dat. Poskytuje metody pro ustanovení spojení s databází, řízení této komunikace a její bezpečné ukončení. Výsledky dotazů do databáze poté posílá do vrstvy aplikační k dalšímu zpracování. Je uzavřena ve jmenném prostoru *Mediastore.DataAccessLayer*.

Pro účely této práce je datová vrstva napsána pro práci s Microsoft SQL Serverem a používá třídy z jmenného prostoru *System.Data.SqlClient*. Je však koncipována tak, aby v případě změny SQL poskytovatele byla možnost tuto vrstvu přepsat bez nutnosti zásahů do ostatních vrstev, a to pouze za podmínky dodržení rozhraní komunikace mezi vrstvami.



Obr 5.6: Ukázka tříd pro obsluhu tabulek Payments v databázi

Na obrázku 5.6 je znázorněna část diagramu tříd datové vrstvy (kompletní diagram viz Příloha č.1), na které založím následující příklad. Základem této vrstvy je abstraktní třída **DataAccessGeneric**, která poskytuje prostředky pro načtení a nastavení připojovacího řetězce z konfigurace aplikace, konfiguraci samotného připojení k databázi a prostředky pro obsluhu SQL transakcí. Všechny ostatní třídy z datové vrstvy dědí vlastnosti této abstraktní třídy. Tyto třídy pak implementují volání uložených procedur na SQL databázovém serveru a jejich zpracování do podoby, která bude předána vrstvě aplikační. Ke každé takové třídě je pak ještě dopsána enumerace, která obsahuje jednotlivé položky v SQL tabulce v pořadí, v jakém jsou tyto položky uloženy. Tato enumerace usnadňuje úpravu kódu v případě potřeby změny struktury tabulky v databázi.

```
namespace DataAccessLayer
{
    /// <summary>
    /// abstraktní třída zapouzdřující globální operace s databází MSSQL
    /// </summary>
    public abstract class DataAccessGeneric
    {
        Fields

        Properties

        Constructors

        Destructors

        Methods
    }
}
```

Obr 5.7: *Struktura abstraktní třídy DataAccessGeneric*

```
namespace DataAccessLayer
{
    /// <summary>
    /// třída zajišťující přístup do tabulek Payments v databázi
    /// </summary>
    public class PaymentDB : DataAccessGeneric, IDisposable
    {
        Constructors

        Insert Methods

        Update Methods

        Delete Methods

        Select Methods
    }
}
```

Obr 5.8: *Struktura třídy PaymentDB*

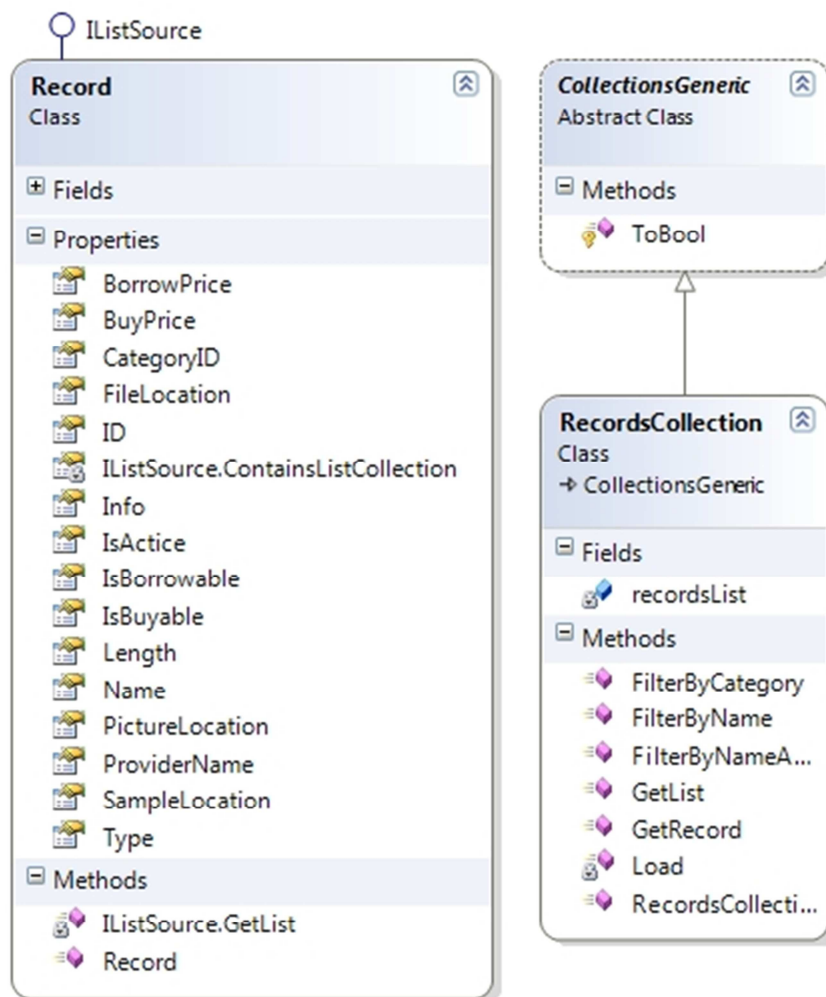
Systém nejdříve otevře konekci k databázi, která je reprezentována proměnnou typu *SqlConnection* specifickou pro Microsoft SQL server. V rámci tohoto připojení je vytvořena proměnná typu *SqlCommand*, na kterou se namapuje uložená procedura, kterou má tento příkaz zavolat, a dále parametry, které jsou potřeba uložené proceduře předat. Po odpovědi SQL databáze je do aplikační vrstvy vrácena patřičná informace – v případě dotazu typu „SELECT“ se jedná o proměnnou typu *SqlDataReader*, která obsahuje kolekci se všemi řádky, které byly vyžádány z databáze. Protože je nutné procházet tuto kolekci pomocí čísel sloupců odpovědi, je na místě použít naimplementovanou enumeraci, která tato čísla symbolizuje. Tento postup pak značně vylepšuje čitelnost kódu.

5.2.3 Aplikační vrstva

Aplikační vrstva (Business Layer) je prostředníkem mezi vrstvou datovou a prezentační. Její hlavní podstatou je tedy získání dat pomocí datové vrstvy, jejich zpracování a připravení do formy, kterou poté může prezentační vrstva zobrazit uživateli. Tato vrstva aplikuje jednotlivé objekty aplikace tak, jak byly připraveny v době návrhu aplikace, a implementují jejich vnitřní logiku. Dalo by se říci, že se jedná o samotné jádro aplikace, změnami provedenými v této vrstvě dochází ke změně funkce celé aplikace. V mé práci je tato vrstva uzavřena ve jmenném prostoru *Mediastore.BusinessLayer*.

Na obrázku 5.9 jsou zobrazeny třídy pro práci s nahrávkami. Kompletní diagram tříd aplikační vrstvy je vložen jako Příloha č. 2. Třída *Record* zde reprezentuje jednu konkrétní nahrávku. Obsahuje informace o této nahrávce a dále vlastnosti a metody, které se zavázala dodržet prostřednictvím rozhraní *IlistSource*. Díky dodržení těchto podmínek je možno instanci této třídy použít jako datový zdroj pro prvek *DetailsView*, který je v prezentační vrstvě využíván pro zobrazení detailů konkrétní nahrávky.

Protože aplikace často pracuje s více instancemi těchto základních tříd, je potřeba tyto instance sdružovat. Pro tyto účely používám kolekce definovaných objektů. Každá z tříd, u nichž potřebuji pracovat s větším množstvím jejich instancí, má naimplementovanou vlastní specifickou kolekční třídu, která vždy dědí z univerzální abstraktní třídy *CollectionsGeneric*. Tato abstraktní třída pak obsahuje prostředky, které se mohou hodit všem kolekcím a je možné do ní kdykoliv doplnit další vlastnosti, které všechny ostatní kolekce zdědí. Tyto kolekce po načtení a zpracování dat slouží jako datové zdroje pro prvky umístěné v prezentační vrstvě, zejména pak prvek *GridView*.



Obr 5.9: Ukázka tříd pro práci s nahrávkami

5.2.4 Prezentací vrstva

Prezentací vrstva (Presentation Layer) představuje grafický výstup, přes který je komunikováno se samotným uživatelem. Je vázána na aplikační vrstvu, od které přebírá připravená data k zobrazení, případně jí naopak předává uživatelské vstupy ke zpracování. U větších projektů bývá častým jevem implementace více prezentací vrstev (v případě webových aplikací například s rozhraním pro stolní počítače a s rozhraním pro mobilní zařízení), což je jedna z hlavních výhod třívrstvé architektury.

V této práci je prezentací vrstva provedena jako ASP.NET webové stránky. Skládá se ze sady formulářů, ve kterých jsou použity ovládací prvky této technologie. Zároveň využívá řady výhod jež jsou v těchto prvcích nabízeny, například kontrola vstupních informací přímo na straně klienta.

```

<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs" Inherits="MediaStore.SiteMaster" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
|<head>...</head>
|<body>
|   <form runat="server">
|     <div class="page">
|       <div class="header">...</div>
|       <div class="main">
|         <asp:ContentPlaceHolder ID="MainContent" runat="server">
|           </asp:ContentPlaceHolder>
|       </div>
|       <div class="clear">
|         </div>
|     </div>
|     <div class="footer">
|
|   </div>
| </form>
|</body>
|</html>

```

Obr 5.10: *Struktura Master Page a umístění komponenty ContentPlaceHolder*

Všechny zobrazované formuláře jsou součástí tzv. Master Page. Tato Master Page je stránka, která obsahuje prvky společné pro všechny zobrazované formuláře (např. hlavička, menu) a jejíž součástí je komponenta **ContentPlaceHolder**, ve které je zobrazován obsah samotných formulářů. Tento způsob prezentace umožňuje jednoduchou unifikaci základního vzhledu stránek. V jednotlivých formulářích se pak pouze odkáže na tuto Master Page a naimplementuje se daný **ContentPlaceHolder** (viz obrázek 5.11).

```

<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeBehind="Default.aspx.cs" Inherits="MediaStore._Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
|<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
|   <p>...</p>
|</asp:Content>

```

Obr 5.11: *Implementace formuláře s použitím Master Page*

Pro zobrazování získaných dat používám datové prvky **GridView** pro seznamy prvků a **DetailsView** pro zobrazení detailu konkrétního prvku. Oba tyto prvky fungují na principu vázání dat, kdy se jim jako zdroj dat předá prvek připravený v aplikační vrstvě. Datové komponenty si pak zdroj dat zpracují a zobrazí podle definovaných vlastností. Na obrázcích 5.12 a 5.13 je demonstrováno svázání zdroje dat s požadovaným objektem a výsledek zobrazení tohoto prvku.

```

// získání ID nahrávky z URL
id = Page.Request.QueryString["id"];

// získání konkrétní nahrávky z kolekce
Record record = videoCollection.GetRecord(Int32.Parse(id));

// svázání nahrávky se zobrazovacím prvkem
dwDetail.DataSource = record;
dwDetail.DataBind();

```

Obr 5.12: Svázání objektu s prvkem *DetailsView*

The screenshot shows a web application interface for 'MEDIASTORE'. At the top, there is a navigation menu with links for 'Video', 'Audio', 'Streamy', 'Nastavení účtu', and 'Košík'. In the top right corner, there are links for '[Vytvořit účet]' and '[Přihlásit se]'. The main content area displays details for a music album. On the left, there is a vertical list of labels: 'Obrázek', 'Jméno', 'Info', 'Distributor', 'Délka', 'Cena za nákup', 'Cena za vypůjčení', and 'Vypůjčit'. The corresponding values are: 'Arcade Fire - The Suburbs', 'I feel like I've been living in the city with no children in it', 'Warner', '45 minut', '200 Kč', and '40 Kč'. There are two buttons: 'Zakoupit' (Purchase) and 'Vypůjčit' (Rent).

Obr 5.13: Zobrazení datového zdroje prvkem *DetailsView*

Poslední důležitou součástí prezentační vrstvy, kterou v této podkapitole uvedu, jsou tzv. validátory. Tyto validátory kontrolují uživatelský vstup podle zadaných kritérií, a pokud tento vstup kritériím neodpovídá, vypíší chybové hlášení, aniž by stránku poslaly na zpracování na server. Takovéto zpracování na straně klienta tím pádem ušetří zbytečné odeslání požadavku a odpovědi a stránku odešle až v momentě, kdy jsou vstupní údaje korektní. Na obrázku 5.14 lze vidět funkčnost těchto validátorů v případě špatně vyplněného formuláře.

- Nebyl zadán login
- Chybné potvrzení hesla

Obr 5.14: Validátory vstupních údajů

System dále využívá vestavěných funkcí pro enkódování vstupních informací, kdy si hlídá informace převzaté z uživatelských vstupních prvků a enkóduje je tak, aby nemohly být zpracovány jako XHTML kód. Tohoto nechtěného zpracování by jinak bylo možné využít při tzv. Cross-Site Scriptingu (XSS), což je druh útoku, kdy je uživatel schopen vložit do aplikace své vlastní potenciálně škodlivé skripty, většinou formou odkazu. [13]

5.3 Implementace rolí

Při pohledu na diagram případů užití (na obrázku 4.1) je patrné, že možnosti neregistrovaného zákazníka jsou podmnožinou případů užití registrovaného zákazníka, a dále také že správce portálu má své případy užití diametrálně odlišné od uživatelských rolí. Těchto faktů jsem tedy využil při implementaci vymodelovaných rolí.





5.3.1 Uživatelská část

Uživatelská část aplikace implementuje na jednotlivých formulářích všechny modelované případy užití. Jakýkoliv návštěvník webu, který se nepřihlásí do systému, je považován za neregistrovaného zákazníka a může provádět patřičné akce, které mu podle diagramu na obrázku 4.1 přísluší. Pokud se pokusí vykonat akci, kterou může vykonat pouze registrovaný zákazník, je tento návštěvník přesměrován na formulář pro přihlášení se do systému. Tato část aplikace je cílená pro široké spektrum zákazníků, proto je zde kladen důraz na uživatelskou přehlednost.

MEDIASTORE [[Vytvořit účet](#)] [[Přihlásit se](#)]

Video Audio Streamy Nastavení účtu Košík

Vyhledat:

Obrázek	Jméno	Informace	Délka
	Arcade Fire - The Suburbs	I feel like I've been living in the city with no children in it	45 minut
	Radiohead - King of Limbs	Slowly we unfold, like lotus flower	32 minut
	PJ Harvey - Let England Shake	How is our glorious country plowed? Not by iron ploughs!	38 minut
	The Joy Formidable - The Big Roar	whirling and stuff	38 minut

Obr 5.15: Ukázka uživatelské části aplikace

5.3.2 Správcovská část

Na správcovskou (administrátorskou) část aplikace jsou kladeny značně odlišné nároky než na část uživatelskou. Proto jsem se rozhodl tyto součásti plně oddělit a co se týká grafické prezentace, ke každé z nich přistoupit jiným způsobem.

Na rozdíl od části uživatelské je u administrátorské části potřebné přihlášení se do systému již při vstupu na první formulář, veškeré správcovské akce pak probíhají pouze pod přihlášeným stavem. Pro administrátora je především důležité mít přehled o situacích probíhajících v systému a uložených datech, není zde proto kladen důraz na uživatelskou přístupnost nebo na používání HTPC systému, ale na přehledné zobrazení dat a jejich snadnou editaci.

MEDIASSTORE - ADMINISTRACE											
Přidat kredity		Přidat nahrávku		Přidat poskytovatele nahrávek							
Zákazníci	Streamy	Poskytovatelé	Nahrávky	Platby	Odhlásit se						
Active	ID	Name	Type	CategoryID	Info	ProviderName	Length	BuyPrice	BorrowPrice	FileLocation	PictureLocation
<input checked="" type="checkbox"/>	1	Family Guy	video	2	lucky there's a family guy	Warner	22	100	10	Videos/Bunny/BigBuckBunny.ism	~/Pictures/w2.jpg
<input checked="" type="checkbox"/>	2	How I Met Your Mother	video	2	legendary	Fox	23	110	11	Videos/Bunny/BigBuckBunny.ism	~/Pictures/social_security.jpg
<input checked="" type="checkbox"/>	3	BBT	video	1	this is my spot!	Fox	22	150	15	Videos/Bunny/BigBuckBunny.ism	~/Pictures/w2.jpg
<input checked="" type="checkbox"/>	4	Arcade Fire - The Suburbs	audio	33	I feel like I've been living in the city with no children in it	Warner	45	200	40	Videos/Elephant/ElephantsDream.ism	Pictures/Arcade Fire - The Suburbs/front.jpg
<input checked="" type="checkbox"/>	5	Radiohead - King of Limbs	audio	31	Slowly we unfold, like lotus flower	Fox	32	180	35	Videos/Elephant/ElephantsDream.ism	Pictures/Radiohead - The King Of Limbs/thekingoflimbs.jpg
<input checked="" type="checkbox"/>	6	PJ Harvey - Let England Shake	audio	33	How is our glorious country plowed? Not by iron ploughs!	Warner	38	210	40	Videos/Elephant/ElephantsDream.ism	Pictures/PJ Harvey - Let England Shake/front.jpg
<input type="checkbox"/>	8	Lord of the rings	video	4	o prstenu	Fox	190	200	50	Videos/Bunny/BigBuckBunny.ism	Pictures/Arcade Fire - The Suburbs/front.jpg
<input checked="" type="checkbox"/>	9	Inception	video	9	hard to explain	Warner	150	250	60	Videos/Bunny/BigBuckBunny.ism	The Joy Formidable - The Big Roar
<input checked="" type="checkbox"/>	10	The Joy Formidable - The Big Roar	audio	38	whirling and stuff	Warner	38	160	25	Videos/Elephant/ElephantsDream.ism	Pictures/The Joy Formidable - The Big Roar/cover.jpg

Obr 5.16: Ukázka správcovské části aplikace

5.4 Technické řešení streamingu

Jednou z největších výzev, které jsem v rámci této práce řešil, je samotná forma distribuce nahrávek formou jejich streamování. Protože se jedná o datově náročnou formu přenos přes rozhraní, které se u koncových uživatelů může zásadně lišit rychlostí a kvalitou přenosu, je potřeba implementovat prostředky, které tuto rozdílnost budou řešit. Zároveň je při této činnosti potřeba co nejmenší spoluúčast uživatele, která dané problematice nemusí dostatečně rozumět.

Ve své práci jsem se rozhodl naimplementovat technologii IIS Smooth Streaming od společnosti Microsoft. Jednou z jeho hlavních výhod v kontextu mé práce je snadná integrace s ostatními technologiemi Microsoft, na kterých je moje aplikace založena (.NET Framework, ASP.NET, IIS, SQL Server...). Obdobně jako u demonstrace platby pomocí platební karty, i u implementace streamingu jsem musel řešit některá omezení spojená se zpoplatněním některých součástí streamovacího systému. V tomto případě se však jedná hlavně o nástroje pro tvorbu samotných nahrávek a společnost Microsoft na svých stránkách poskytuje ukázkové nahrávky, které lze použít pro vývoj a testování vlastního řešení. Proto jsem mohl naimplementovat kompletní streamovací systém bez výrazných omezení. Jedinou výjimkou je systém ochrany digitálních práv (DRM – Digital Rights Management), který je v technologii IIS

Smooth Streaming realizován pomocí platformy Microsoft PlayReady. Pro používání této platformy je potřeba zakoupení licence, z tohoto důvodu jsem ji nemohl do své aplikace naimplementovat. Jednotlivé součásti systému tuto technologii nicméně podporují a jsou pro její nasazení připraveny, v případě komerčního nasazení aplikace a zakoupení této licence tedy nic nebrání v jejím používání. Pro účely demonstrace funkčnosti aplikaci navíc není tato součást zapotřebí.

5.4.1 IIS Smooth Streaming

Smooth Streaming je technologie společnosti Microsoft pokrývající komplexní problém streamování multimediálního obsahu přes internet. Poskytuje prostředky pro přípravu streamu, serverové komponenty pro jeho běh, zabezpečení DRM a technické řešení klientské části. Jejím jádrem je serverová část, která je distribuována jako rozšíření webového serveru IIS. Protože server IIS již zajišťuje samotné hostování webové aplikace a její překlad, není pro technické zabezpečení streamování zapotřebí žádného dalšího serveru, což zásadním způsobem šetří zdroje na serverovém počítači. Zpracování odesílaného multimediálního streamu na klientské straně je poté řešeno pomocí technologie Microsoft Silverlight, popisované v kapitole 3.5.

Obrovská síla této technologie spočívá ve způsobu distribuce streamu ke koncovému uživateli v závislosti na konkrétní infrastruktuře, kterou tento stream prostupuje. Díky možnostem aplikací postavených na platformě Silverlight je možné na klientské straně detekovat jak propustnost sítě, tak aktuální vyžití procesoru, který tento stream zpracovává. V závislosti na těchto zjištěných informacích poté posílá na server požadavek na kvalitu zasílaného streamu, přičemž celá tato operace probíhá dynamicky v průběhu přehrávání a bez zásahu uživatele. Hlavní výhody tohoto přístupu jsou:

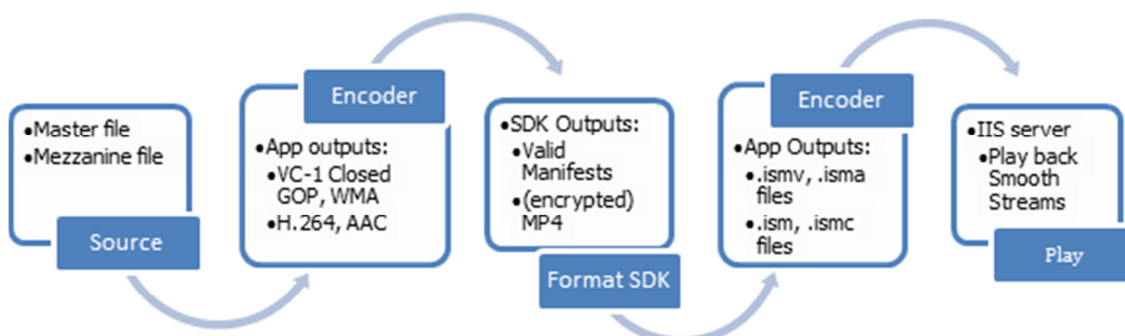
- 1) Nezávislost na prostředcích koncového uživatele – kvalita streamu se automaticky přizpůsobí koncovému zařízení, zejména pak jeho výpočetnímu výkonu a kvalitě jeho připojení k internetu. Je tedy schopna obsloužit širokou škálu zařízení, od smartphonů s mobilním přístupem k internetu (podmínkou je podpora technologie Silverlight) až po vysoce výkonné počítače s broadbandovým internetovým připojením a nabídnout těmto zařízením adekvátně kvalitní stream.
- 2) Dynamická reakce na změnu podmínek na klientské straně – kontrola dostupných výpočetních prostředků probíhá po celou dobu přehrávání streamu. Protože dostupný výpočetní výkon počítače a aktuální kvalita jeho internetové konektce jsou proměnlivé veličiny, velmi často dochází k jejich drastickým změnám v průběhu času. Technologie Smooth Streaming na tyto změny automaticky reaguje změnou datového průtoku streamu.
- 3) Uživatelská nenáročnost a odstranění zbytečného čekání na načtení streamu – protože nastavení kvality přehrávaného streamu probíhá automaticky na pozadí, není od uživatele požadována žádná akce týkající se nastavení parametrů přehrávání. Zároveň tento způsob odstraňuje nutnost čekání na nabufferování streamu, která

u fixně zvolené kvality může s ohledem na proměnlivé prostředí přijít kdykoliv v průběhu streamování a projeví se pozastavením přehrávání do doby, než bude mít přehrávač k dispozici další data. U technologie Smooth Streaming namísto toho dojde pouze ke snížení kvality přehrávaného obsahu, k pozastavení přehrávání se přehrávač uchýlí pouze ve chvíli, kdy nemá dostatek dat pro zpracování ani streamu nejnižší dostupné kvality.

Na adrese <http://www.iis.net/media/experiencesmoothstreaming> je k dispozici multimediální přehrávač demonstrující vlastnosti této technologie v praxi.

5.4.2 Příprava streamu a jeho distribuce

Multimediální soubor, který bude k dispozici ke streamování, je zapotřebí nejdříve převést do potřebného formátu a rozdělit do jednotlivých kvalit a datových toků přenášejících dat. U technologie Smooth Streaming k tomuto účelu slouží program Expression Encoder Pro. K aktuálnímu datu (12.5.2011) je cena licence stanovena na \$199,95 (viz ⁵) Na obrázku 5.17 je zobrazen princip činnosti tohoto programu.



Obr. 5.17: Příprava streamovaných souborů

Program zpracuje vstupní soubor a vytvoří následující zdroje streamu:

- Server manifest soubory (*ism, *ismc) – jedná se o XML soubory obsahující informace o samotném streamu, jeho rozdělení na jednotlivé segmenty a informace o kvalitách jednotlivých vygenerovaných souborů s multimediálním obsahem
- Multimediální soubory (*isma, *ismv) – jedná se o soubory se samotným multimediálním obsahem, ať již čistě zvukovým (*.isma), nebo kombinovaným audio-vizuálním (*.ismv), a to ve formátu MP4. Pro každý datový tok je vygenerován vlastní soubor. Systém umožňuje i tvorbu video souborů s více audio zdroji (typicky

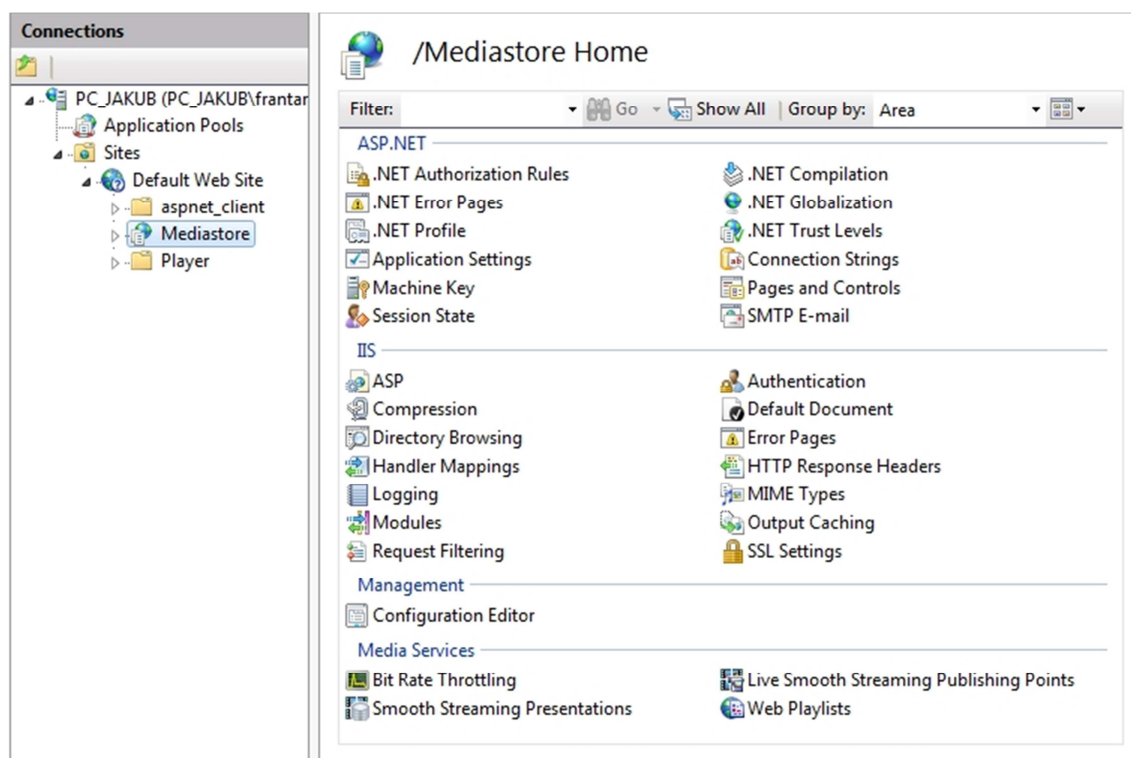
⁵

http://www.microsoftstore.com/store/msstore/en_US/buy/pageType.product/externalRefID.0BE96B98?WT.mc_id=expressionsite_EncoderPro4Buy

dabing u filmů či seriálů) – v tomto případě jsou vygenerovány jednotlivé *.ismv soubory s původní audio stopou pro každý datový tok a další audiostopy jsou generovány do samostatných souborů obsahující pouze toto audio. Expression Encoder Pro je dále schopen vytvořit šifrované soubory s DRM ochranou a zaregistrovat je u služby Microsoft PlayReady, která se poté stará o distribuci licencí potřebných pro rozkódování těchto souborů jednotlivým klientům.

- Ostatní soubory (např *.ismt) – soubory poskytující podporu pro doplňkové služby technologie Smooth Streaming. Jako příklad uvedu soubory *.ismt, což jsou XML soubory obsahující informace o titulcích pro danou nahrávku

Takto zpracované soubory se poté nahrají na IIS server. Na tomto serveru je dále potřebné mít nainstalovaný rozšiřující balíček IIS Media Services ⁶, který poskytne samotné prostředky pro streaming pomocí IIS serveru a dále také prostředky pro klientskou část služby.



Obr 5.18: IIS server s nainstalovaným balíčkem IIS Media Services

⁶ Viz <http://www.iis.net/media>

5.4.3 Silverlight klient



Obr 5.19: *Smooth Streaming Client 1.5*

Přehrávání na straně klienta obstarává Microsoft Smooth Streaming Client, aktuálně ve verzi 1.5. Tento přehrávač je do stránky vložen jako objekt, který si klientský internetový prohlížeč natáhne ze serveru. Z hlediska požadavků tedy stačí, aby měl uživatel nainstalované rozhraní Microsoft Silverlight.

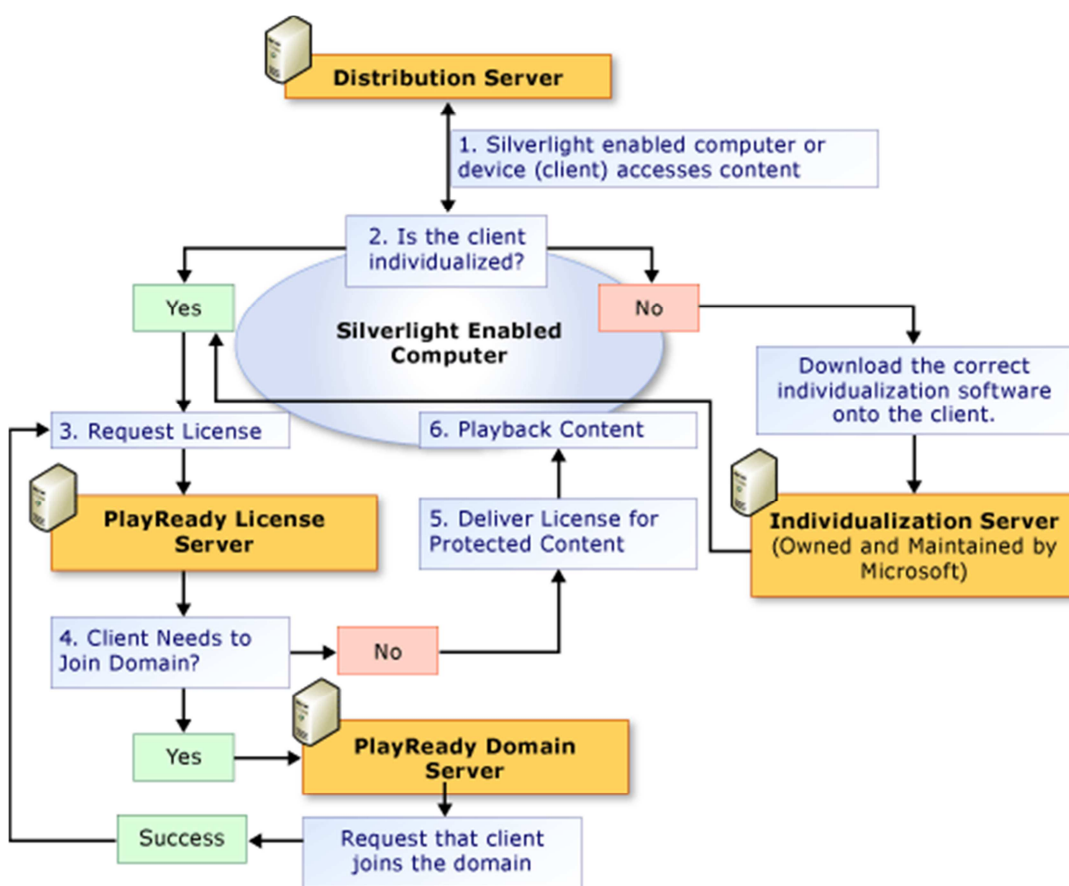
Přehrávač implementuje všechny podstatné vlastnosti technologie Smooth Streaming, jež byly vyjmenovány v předchozích podkapitolách. Zvládá tedy dynamickou kontrolu dostupných výpočetních zdrojů a kvality internetové konekce, podle které volí kvalitu streamu (ta je graficky indikována v pravém dolním rohu), dále podporuje volbu zvukové stopy, zobrazování titulků či přeskokování mezi předem definovanými částmi nahrávky (čehož se dá využít při prezentaci hudebních alb jako celku, kde jsou tyto body nadefinovány jako začátky jednotlivých skladeb).

Další užitečnou vlastností technologie Silverlight je tzv. „Cross-domain policy“. Jedná se o soubor pravidel (v podobě XML souboru), který definuje, jaké domény a porty budou mít přístup k Silverlight obsahu generované na tomto serveru. V základním nastavení je tento přístup udělen pouze objektům, které jsou umístěny na stránkách generovaných na tomto

serveru. To je v případě tohoto typu aplikace žádané chování, které zabraňuje neautorizovanému přístupu k uloženým streamům ze strany cizích subjektů.

5.4.4 DRM ochrana

Jak jsem se již na začátku této kapitoly zmiňoval, vlastní DRM ochrana v mé práci naimplementována není z důvodu finanční nákladnosti. Aplikace přesto ošetřuje požadavek na spuštění streamů od neautorizovaných zdrojů pomocí „Cross-domain policy“ pravidel. Samotný stream je však přenášen v nešifrované podobě, a tudíž náchylný na případný cílený útok. Těmto útokům zabrání až zašifrování přenášeného streamu, které u technologie Smooth Streaming řeší zakoupení PlayReady DRM serveru. Pro nasazení do komerčního provozu je tato ochrana nezbytná, protože poskytovatelům obsahu dodává potřebnou záruku ochrany jejich produktů. Aplikace je pro použití PlayReady DRM připravena, po případném zakoupení této technologie tak tedy stačí pouze nakonfigurovat PlayReady server a zašifrovat poskytované streamy program Expression Encoder Pro.



Obr 5.20: Diagram funkčnosti PlayReady serveru ⁽⁷⁾

⁷ (12.5.2011) <http://i.msdn.microsoft.com/dynimg/IC400721.png>

Na obrázku 5.20 je znázorněna funkčnost takového řešení. Silverlight aplikace nejdříve zjistí, zda-li se na počítači nachází software pro zpracování DRM ochrany. Pokud je výsledek tohoto zjištění negativní, tak si tento software obstará a nainstaluje (tomuto procesu se říká individualizace). Jakmile je již software nainstalován, silverlight aplikace ho bude používat bez nutnosti opětovného přístupu na individualizační server, s výjimkou případů zamítnutí tohoto software ze strany požadavků distribučního serveru (typicky při požadavku na novější verzi DRM software). Poté se již Silverlight aplikace připojí na PlayReady licenční server, který podle implementované logiky rozhodne o udělení licence. V případě kladného vyřízení požadavků je na klientský počítač zaslána licence, která slouží k přehrávání multimediálního obsahu streamu, který byl zašifrován v době jeho vytvoření. [14]

5.5 Nasazení do provozu a testování

S ohledem na licenční omezení diskutovaná v předchozích kapitolách jsem provedl nasazení a testování výsledné aplikace v odpovídajícím stavu. Jako webový server jsem použil Microsoft Internet Information Services ve verzi 7.5 s rozšířením IIS Media Services 4.0 pro podporu multimediálního streamování a databázový server Microsoft SQL Server 2008.

Testování pak probíhalo ve dvou fázích. První fáze spočívala v průběžném testování jednotlivých komponent v průběhu jejich vývoje a po jejich dokončení. Kromě základní funkčnosti komponent byla vždy zaměřena na konkrétní vlastnosti dané součásti aplikace a zkoušku špatných či krajních hodnot, které byly pro danou komponentu specifické. Kromě opravy chyb bylo v této fázi upřesňováno i chybové hlášení při zadání chybných hodnot tak, aby uživatel měl co nejlepší přehled o chybách způsobených jeho vstupy, ale zároveň tak, aby mu nebyla vypsána nějaká konkrétnější chyba obsahující interní programové informace. V případě databázové chyby tak například uživatel získá informaci o tom, že proběhla chyba při práci s databází, nezjistí ale, že v tabulce „Users“ nebyl nalezen sloupec s názvem „Country“.

Druhá fáze testování poté spočívala v ověření činnosti aplikace jako celku. Testování bylo zaměřeno na vzájemnou spolupráci jednotlivých komponent, jejichž samostatná činnost již byla ověřena v první fázi testování. Zároveň jsem ověřoval splnění vstupních předpokladů aplikace a zkoumal možnosti jejího vylepšení, zejména pak z hlediska uživatelské přívětivosti.

Speciální pozornost jsem pak věnoval ověření činnosti streamu a jeho zabezpečení prostřednictvím pravidel Cross-domain policy, dále pak zabezpečení databáze proti útoku SQL Injection a zabezpečení stránek vůči Cross-Site skriptování.

6. Závěr

6.1 Přínos práce

Cílem práce bylo vytvoření webové aplikace, která umožňuje prodej a vypůjčování multimediálních nahrávek a realizuje jejich distribuci k uživateli formou jejich streamingu.

Při vývoji aplikace jsem využil značného množství technologií, které jsem před započatím práce znal pouze okrajově, případně vůbec. Jedná se především o vývojové prostředí Visual Studio, implementační technologie .NET Framework a ASP.NET a databázový server Microsoft SQL Server. Mé znalosti a zkušenosti s těmito technologiemi tedy značně vzrostly a pevně doufám, že je v budoucnu budu moci řádně použít. Největší výzvou a zároveň z mého subjektivního pohledu nejzajímavější částí této práce pak byla implementace technologií streamingu multimediálního obsahu. Zde jsem zvolil moderní technologii Microsoft IIS Smooth Streaming a klientského zpracování pomocí Microsoft Silverlight. Výsledkem je funkční spojení informačního systému evidující dostupné nahrávky a systému pro jejich prezentaci koncovému uživateli. Dále jsem získal znalosti v oblasti právního aspektu prodeje digitálních nahrávek a technických možností implementace platby platební kartou v internetovém prostředí.

V průběhu návrhu a implementace jsem se držel principů softwarového inženýrství, popsaného možnostmi jazyka UML, a třívrstvé implementační architektury. Díky těmto postupům je výsledná aplikace snadno udržitelná a rozšiřitelná. Třívrstvá architektura ve výsledku umožňuje značnou škálovatelnost aplikace, například změnou databázového poskytovatele, při které by bylo potřeba udělat úpravy pouze v datové vrstvě, nebo přidáním jiného ovládacího rozhraní, které by bylo provedeno změnami v prezentační vrstvě. Takovým rozhraním může být například výstup na mobilní telefony, nebo pluginy do media center, které využívají tzv. HTPC systémy.

Přestože na trhu již existuje značné množství podobných systémů, tato aplikace má určitá specifika, která se od ostatních řešení v určitých ohledech liší. Vzhledem k jejímu cílení na český trh je důležité především porovnání s videopůjčovnou Topfun.cz. Oproti ní nabízí možnost širšího záběru nabízených nahrávek, které jsou obohaceny o hudební tituly. Dále se jedná o zcela odlišnou formu implementace streamingu, kde moje aplikace využívá automatické kontroly kvality přenášeného streamu technologie IIS Smooth Streaming, což snižuje nároky na technické znalosti uživatele v oblasti přehrávání multimediálních streamů. Oproti Topfun.cz je dále naimplementována, kromě platby kartou, také možnost platby pomocí kreditů, které si uživatel předem nabije prostřednictvím bankovního převodu. Kromě bodů popsaných v zadání práce navíc rozšiřuje funkcionalitu portálu o možnost vypůjčování nahrávek, čímž značně mění možnost potenciálního komerčního nasazení. Dále také implementuje alternativní způsob platby za objednávky využívá moderního způsobu distribuce streamovaného obsahu.

6.2 Možná rozšíření práce

V případě komerčního nasazení aplikace je zapotřebí doimplementovat součásti, které vyžadují ke své činnosti zakoupení licencí. Jedná se o systém pro platbu platební kartou, který je dodáván vybranou finanční organizací a na který se váže podpis smlouvy, ve kterém jsou popsány finanční podmínky za prováděné transakce. Aplikace je nyní připravena pro použití se systémem PayPal Website Payments, v případě zakoupení jiného řešení platby je však možné lehce aplikaci přizpůsobit. Druhou potřebnou implementací je zakomponování Microsoft PlayReady DRM Licencing serveru do systému poskytování streamingu. Tento systém umožní šifrování přenášených multimediálních streamů a zařídí udělování licencí pro jejich dešifrování jednotlivým koncovým stanicím. Klientská část mé aplikace je v tomto případě opět připravena šifrované streamy vysílat i přehrávat, v případě zakoupení PlayReady DRM technologie je tedy pouze potřeba nastavit logiku udělování licencí.

Budoucí rozšíření aplikace by se mohlo týkat především v dalších možnostech klientského výstupu aplikace. Protože je využito třívrstvé architektury, lze tyto výstupy realizovat pomocí přepsání prezentační vrstvy bez nutnosti upravovat ostatní dvě vrstvy, tedy bez nutnosti měnit logiku aplikace a její práci s daty. Může se jednat například o implementaci grafického rozhraní do různých druhů mediálních center, které využívají systémy pro domácí zábavu HTPC, či do uživatelského rozhraní některé z herních konzolí. Další možností v tomto ohledu je také implementace do prostředí mobilních telefonů, kdy je možno především v zahraničí sledovat nárůst zájmu o tyto multimediální služby v mobilních zařízeních, spojené poklesem cen za datové připojení.

Použitá literatura

1. **Raby, Mark.** *TG Daily* [online]. 2011 [cit. 2011-05-02]. Youtube video-on-demand service finally moves forward. Dostupné z WWW: <<http://www.tgdaily.com/software-brief/55568-youtube-video-on-demand-service-finally-moves-forward>>.
2. **Hulu** [online]. 2011 [cit. 2011-05-02]. Hulu Plus. Dostupné z WWW: <<http://www.hulu.com/plus>>.
3. **Jones, Richard.** *Last.fm - the Blog* [online]. 2009 [cit. 2011-05-02]. Radio Subscriptions. Dostupné z WWW: <<http://blog.last.fm/2009/04/22/radio-subscriptions>>.
4. **Sedlák, Jan.** *Živě.cz* [online]. 2009 [cit. 2011-05-02]. Internetová Videopůjčovna ČT po dvou letech končí, nevyplatila se. Dostupné z WWW: <<http://www.zive.cz/bleskovky/internetova-videopujcovna-ct-po-dvou-letech-konci-nevyplatila-se/sc-4-a-150084/default.aspx>>.
5. *Česká spořitelna* [online]. 2011 [cit. 2011-02-24]. PLATBA 24. Dostupné z WWW: <<http://www.csas.cz/banka/nav/podnikatele-firmy-a-institute/financni-institute/platba-24/o-produktu-d00010484>>.
6. *PayPal* [online]. 2010 [cit. 2011-02-24]. Introducing Direct Payment. Dostupné z WWW: <https://cms.paypal.com/us/cgi-bin/?cmd=_render-content&content_ID=developer/e_howto_api_WPDirectPayment>.
7. **Burget, Radek; Zeman, David.** *Tvorba webových stránek, studijní opora* [online]. 2006 [cit. 2011-05-02]. Dostupné z WWW: <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/ITW-IT/texts/opora_itw_061020.pdf>.
8. **Arlow, Jim a Neustadt, Ila.** *UML a unifikovaný proces vývoje aplikací*. Brno : Computer Press, 2003. ISBN 80-7226-947-X.
9. **Kačmář, Dalibor.** *Programujeme .NET aplikace ve Visual Studiu .NET*. Brno : Computer Press, 2001. ISBN 80-7226-569-5.
10. **MacDonald, Matthew, Freeman, Adam a Szpuszta, Mario.** *ASP.NET 4 a C# 2010 - tvorba dynamických stránek profesionálně, kniha 1*. Brno : Zoner Press, 2011. ISBN 978-80-7413-131-8.
11. **Fowler, Martin.** *Destilované UML*. Praha : Grada Publishing, 2009. ISBN 978-80-247-2062-3.
12. **Page-Jones, Meilir.** *Základy objektově orientovaného návrhu v UML*. Praha : Grada Publishing, 2001. ISBN 80-247-0210-X.

13. **Meier, J.D., et al.** *MSDN* [online]. 2005 [cit. 2011-05-08]. How To: Prevent Cross-Site Scripting in ASP.NET. Dostupné z WWW: <<http://msdn.microsoft.com/en-us/library/ff649310.aspx>>.

14. *MSDN* [online]. 2011 [cit. 2011-05-12]. Digital Rights Management (DRM). Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/cc838192\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/cc838192(v=vs.95).aspx)>.

Obsah přiloženého DVD

- Přeložená webová aplikace
- Zdrojové kódy aplikace
- Dokumentace
- Návod na instalaci
- Ukázkové nahrávky
- Elektronická verze této technické zprávy
- Diagramy tříd