

Přehrávání a záznam on-line médií

Online media playing and recording

Jaromír Uhřík



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jaromír UHŘÍK**
Osobní číslo: **A08890**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Přehrávání a záznam on-line médií**

Zásady pro vypracování:

1. Provedte analýzu možných přístupů k přehrávání a záznamu on-line médií.
2. Zpracujte přehled požadavků typových uživatelů.
3. Navrhněte vhodné uživatelské rozhraní.
4. Zpracujte funkční prototyp aplikace.
5. Prototyp aplikace vyhodnoťte.
6. Uvedte další možný rozvoj aplikace.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. JEFFREY, Richter. .NET Framework : programování aplikací. [s.l.] : GRADA Publishing, a.s., 2003. 552 s. ISBN 978-80-247-0450-1.
2. AUSTERBERRY, David. The Technology of Video and Audio Streaming : US. 2nd Edition. [s.l.] : Focal Press Title, 2005. 360 s. ISBN 978-0-240-80580-1.
3. FOLLANSBEE, Joe. Hands-On Guide to Streaming Media : an Introduction to Delivering On-Demand Media. 2nd Edition. US : Focal Press Title, 2006. 336 s. ISBN 978-0-240-80863-5.
4. LENZ, Gunther; MOELLER, Thomas. .NET – A Complete Development Cycle. [s.l.] : Addison-Wesley Professional, 2004. 592 s. ISBN 0-321-16882-8.
5. un4seen developments. BASS audio library [online]. 2010 [cit. 2011-02-03]. Audio library. Dostupné z WWW: [http://www.un4seen.com/].

Vedoucí bakalářské práce:

Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

25. února 2011

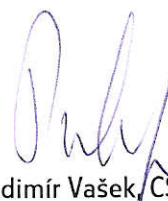
Termín odevzdání bakalářské práce:

7. června 2011

Ve Zlíně dne 25. února 2011



prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Bakalářská práce se zabývá přístupem k poslechu a zaznamenávání online médií. Teoretická část uvádí obecné principy fungování těchto procesů a charakterizuje technologie, které jsou dále použity při implementaci reálné ukázkové aplikace. Praktická část popisuje vnitřní návrh aplikace a možnosti, které jsou výsledkem její implementace.

Klíčová slova:

Přehrávač, Streaming, Internetová rádia, Nahrávání, Cloud-recorder, .NET, Mp3, Ogg, Aac, Wma

ABSTRACT

This thesis deals with access to online listening and recording media. The theoretical part presents the general principles of these processes and describes technologies that are also used in the implementation of a real sample application. The practical part describes application's internal design and possibilities that result from its implementation.

Keywords:

Player, Streaming, Internet radio, Recording, Cloud-recorder, .NET, Mp3, Ogg, Aac, Wma

PODĚKOVÁNÍ

Tímto bych rád poděkoval svému vedoucímu panu Ing. Petru Šilhavému, Ph.D. za vedení bakalářské práce a čas strávený při konzultacích.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PŘEHRÁVÁNÍ ONLINE MÉDIÍ.....	11
1.1 STREAMING.....	11
1.1.1 Technická realizace - serverové technologie	11
1.1.2 Služby využívající streaming.....	13
1.2 PROGRESSIVE DOWNLOAD	14
1.3 POROVNÁNÍ STREAMING VS PROGRESSIVE DOWNLOAD.....	14
1.4 ZPŮSOBY PŘEHRÁVÁNÍ.....	15
1.4.1 Desktopové přehrávače.....	15
1.4.2 Přehrávání na webu	15
2 NAHRÁVÁNÍ ONLINE MÉDIÍ.....	17
2.1 PŘÍSTUPY K NAHRÁVÁNÍ.....	17
2.2 DIGITAL RIGHTS MANAGEMENT.....	18
2.3 AUTORSKÝ ZÁKON	19
3 SOUVISEJÍCÍ TECHNOLOGIE	20
3.1 AJAX	20
3.2 APACHE	20
3.3 CSS	21
3.4 HTML.....	21
3.5 JAVASCRIPT.....	22
3.6 JQUERY	23
3.7 JSON.....	23
3.8 .NET	24
3.9 MYSQL.....	26
3.10 PHP.....	26
3.11 SQL.....	27
3.12 XML	27
4 AUDIO FORMÁTY	28
4.1 MP3.....	28
4.2 AAC.....	28
4.3 OGG VORBIS	28
4.4 WMA	28
II PRAKTICKÁ ČÁST.....	29

5	DATABÁZE RÁDIÍ	30
5.1	NÁVRH DATABÁZE	30
5.2	ROZHRANÍ API.....	30
5.2.1	Inicializační rozhraní	30
5.2.2	Editace položky	35
6	UKÁZKOVÁ APLIKACE.....	37
6.1	ZÁKLADNÍ NÁVRH APLIKACE	37
6.2	IMPLEMENTACE PŘEHRÁVACÍHO ENGINE.....	38
6.2.1	Použitá knihovna.....	39
6.2.2	Metody objektu.....	41
6.2.3	Vlastnosti objektu	43
6.2.4	Grafické znázornění veřejných metod a vlastností.....	45
6.3	DATOVÝ OBJEKT	45
6.4	UŽIVATELSKÉ ROZHRANÍ.....	47
6.4.1	Přehrávání.....	47
6.4.2	Nahrávání	47
6.4.3	Oblíbené	48
6.4.4	Nahrávky	48
6.4.5	TimeMachine	48
6.4.6	Nastavení.....	49
6.4.7	Ostatní funkce.....	49
6.5	NÁHLED UKÁZKOVÉ APLIKACE	49
7	DEMONSTRACE ONLINE ŘEŠENÍ PŘEHRÁVAČE A FUNKCE TIMEMACHINE	54
7.1	ČASOVÁ OSA.....	54
7.2	PŘEHRÁVÁNÍ.....	54
7.3	ORIENTACE V HISTORII VYSÍLÁNÍ	54
7.4	POUŽITÉ TECHNOLOGIE.....	55
	ZÁVĚR	56
	ZÁVĚR V ANGLIČTINĚ	57
	SEZNAM POUŽITÉ LITERATURY.....	58
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	60
	SEZNAM OBRÁZKŮ	61
	SEZNAM TABULEK	62
	SEZNAM PŘÍLOH	63

ÚVOD

Přehrávání a záznam online médií je oblast, která se stává být s rozvojem internetu stále více a více populární. Technologická realizace umožňovala distribuci audio-video médií prostřednictvím internetu již od svého ranného počátku. Při prvotním rozmachu internetu v České republice však bylo masové využívání těchto technologií nemyslitelné, a to zejména z důvodu tehdejší minimální propustnosti linek. Nemalou měrou k útlumu používání těchto způsobů distribuce tehdy přispěla také metodika účtování poplatků za internetové připojení.

V době vytáčeného připojení, nebo pozdějších ADSL linek s poplatky za přenesené množství dat, bylo nepředstavitelné, aby někdo sledoval online televize nebo poslouchal internetová rádia.

Když se v roce 2003 objevila vůbec první česká aplikace zaměřená na poslech internetových rádií - NetRadio, tak jí tehdy Jiří Laudát, redaktor Hospodářských novin, nedával mnoho šancí na přežití.

Situace ohledně způsobu poskytování internetové konektivity se naštěstí razantně změnila a nabrala v následujících letech zcela opačný směr. Dnes má přístup k internetu již většina populace a jeho rychlostí a množstvím přenesených dat se až na výjimky nemusíme zabývat.

Trendem dnešní doby je naopak prosazování technologií provozovaných vzdáleně, kdy uživatelé k aplikacím přistupují online a k jejich používání stačí minimální výpočetní výkon počítače, který je tak pouze v roli klienta.

Internet se stal pro mnohé nedílnou součástí života a jakousi samozřejmostí, bez které by si svou práci nebo relaxaci nedokázali představit. Online média nás doprovázejí prakticky na každé webové stránce. Web již dávno není jen o textových informacích doplněných maximálně několika ilustračními obrázky. Internet nám přinesl rozsáhlé fotogalerie, video servery, hudební portály a tisíce internetových rádií. Všechna tato média nám zpřjemňují procházení webu a tím ho dotvářejí do zajímavé interaktivní podoby. Neopomenutelnou událostí posledních let se stalo rozšíření tzv. sociálních sítí, které významně napomáhají šíření, chcete-li sdílení, těchto zdrojů a vytvářejí jejich adekvátní provázanost.

Dále se budeme zabývat zejména audio obsahem v prostředí internetu.

I. TEORETICKÁ ČÁST

1 PŘEHRÁVÁNÍ ONLINE MÉDIÍ

Přehrávání online médií lze realizovat několika způsoby, z nichž nejznámější jsou streaming a progressive download.

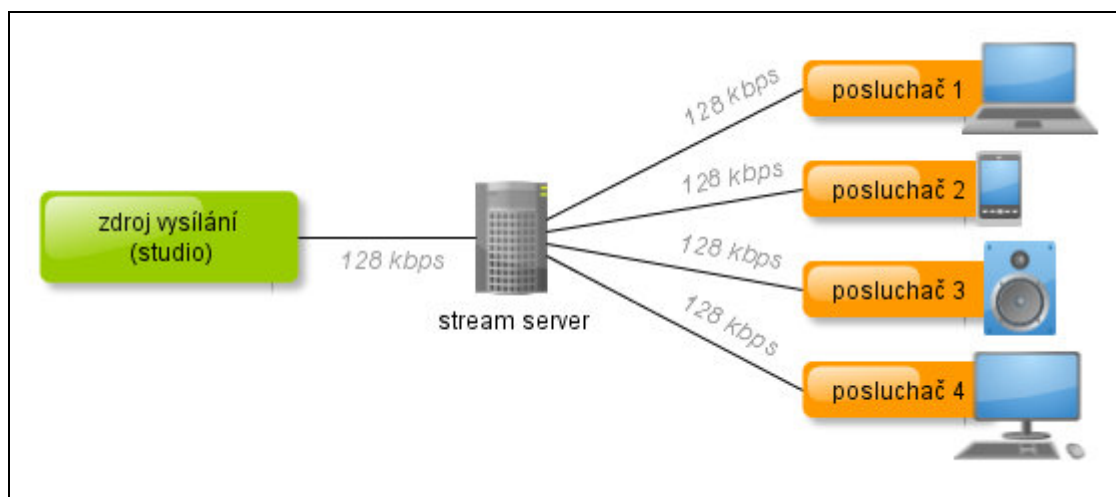
1.1 Streaming

Streaming je technologie kontinuálního přenosu multimediálních dat mezi zdrojem (stream server) a koncovým uživatelem (přehrávač). V současné době se streamingu využívá především pro přenášení audiovizuálního materiálu po internetu – často nazýváno také jako tzv. webcasting. Pro streamování obsahu více uživatelům zároveň musí mít provozovatel k dispozici kromě obsahu také streamovací server, který zajišťuje komunikaci s cílovými počítači a plynulé vysílání dat. [1], [2]

1.1.1 Technická realizace - serverové technologie

Základem distribuce streamovaného obsahu je návrh typu zdroj-server-klient. Toto rozložení umožňuje oddělit místo vzniku zdroje vysílání (nejčastěji studio) od distribučního bodu, ke kterému se připojují jednotliví posluchači. Distribučním bodem je stream server, který je nejčastěji situován v housing centru a je připojen k pátevní síti s dostatečnou kapacitou. Díky tomuto oddělení nejsou kladeny vysoké nároky na konektivitu přímo ve studiu rádia.

Na obrázku 1 můžeme vidět základní schéma realizace vysílání. Rádio vysílá svůj obsah v kvalitě vysílání (bitrate) 128 kbps - datový tok směrem od zdroje vysílání ke stream serveru je pouze 128 kbps (upload). Server tento stream dále poskytuje koncovým posluchačům a datové nároky na kapacitu linky jsou přímo úměrné počtu posluchačů. Například pro obsloužení 1000 současně připojených posluchačů je nutné, aby byla šířka pásma pro upload $1000 \times 128 \text{ kbps}$, což je 125 Mbps.



Obrázek 1 - Zjednodušené schéma vysílání internetového rádia [vlastní zpracování]

Nejrozšířenější stream servery

- SHOUTcast
 - SHOUTcast je multiplatformní software sloužící pro streamování médií.
 - Je celosvětově rozšířenou technologií pro distribuci rádiového vysílání prostřednictvím sítě internet, v současné době jej využívá více jak 48 000 rádiových stanic po celém světě, stoupající oblíbenost tohoto software je vidět v tabulce 1.
 - Podporuje dva základní audio formáty Mp3 a AAC.
 - Od roku 1999 vyvíjen společností Nullsoft, autorem oblíbeného multimediálního přehrávače Winamp.
- Icecast
 - Stream server původně umožňoval distribuovat vysílání pouze ve formátu OGG, aktuální verze podporuje audio formáty OGG, AAC a Mp3.
 - Kompatibilita s protokolem používaným u SHOUTcast serveru.

Tabulka 1 - Oblíbenost stream serveru SHOUTcast [20]

rok	počet rádiových stanic
1999	900
2000	1 217
2001	2 400
2002	3 000
2003	4 000
2004	5 000
2005	10 000
2006	15 000
2007	18 000
2008	24 000
2009	32 353
2010	43 814
2011	48 816

1.1.2 Služby využívající streaming

Shoutcast.com

Server sloužící jako souhrnný seznam rádiových stanic, které pro vysílání využívají stejnojmenný stream server. Nabízí desetitisíce rádiových stanic z celého světa. Nabídku lze procházet podle hudebních stylů a výsledky je možné dále řadit podle kvality vysílání, oblíbenosti stanice nebo formátu, ve kterém je vysílána.

Rádiové stanice jsou k dispozici ve formátech Mp3 nebo AAC v datovém toku 8 – 320 kbps.

Play.cz

Portál momentálně nabízí poslech necelé stovky českých FM rádií, v nabídce najdeme ale i ryze internetové stanice.

Nabídka dostupných formátů se u jednotlivých stanic liší, k dispozici je WMA, Mp3, OGG a AAC.

ABradio.cz

Portál stejného zaměření jako Play.cz. Poskytované formáty vysílání: WMA nebo Mp3.

1.2 Progressive download

Progressive Download je termín používaný k popisu přenosu digitálních multimediálních souborů ze serveru ke klientovi. Realizace probíhá typicky pomocí HTTP protokolu. Přehrávač může začít přehrávání médií před dokončení stahování, není tak nutné čekat na dokončení stažení celého souboru. Klíčový rozdíl mezi streamováním médií a metodou progressive download je v tom, jak jsou multimediální data přijímána a ukládána klientskou aplikací, která k obsahu přistupuje.

Multimediální přehrávač schopný přehrávání metodou progressive download spoléhá na meta data umístěná v hlavičce souboru, který postupně načítá do lokální vyrovnávací paměti ze vzdáleného serveru. Jakmile lokální vyrovnávací paměť dosáhne určitého množství dat, které je dostačující k zahájení přehrávání, provede se spuštění přehrávání obsahu.

Informace o tomto minimálním množství dat ve vyrovnávací paměti je vloženo do souboru poskytovatelem obsahu pomocí nastavení encoderu, informace je dále poskytnuta přehrávači obsahu, který na ni reaguje korektním nastavením podmínek přehrávání. [3]

1.3 Porovnání Streaming vs Progressive download

Základní rozdíly mezi zmíněnými metodami přehrávání [4]:

- Při streamingu se pomocí vhodného protokolu předávají multimediální data, včetně potřebných technických informací a řídicích příkazů, v reálném čase rychlostí odpovídající přehrávání.
- Progressive download naopak využívá maximální dostupné rychlosti linky.
- Řídicí příkazy jakou jsou například pozastavení, opětovné spuštění, přechod na požadovaný čas, apod. při streamingu interpretuje server, který na ně reaguje a na je jejich základě mění obsah vysílaného streamu.
- Progressive download vše interpretuje na straně klienta a distribuční serverem je běžný HTTP server, pro který je požadavek klienta pouze běžným požadavkem na download souboru.

- Při streamingu klient vůbec neukládá obdržená data, nanejvýš ukládá pouze jejich část do vyrovnávací paměti z důvodu plynulejšího přehrávání při nespolehlivém připojení, není to však přímo podmínkou.
- Progressive download naopak z principu nutně vždy ukládá celý přijatý soubor na disk a dynamicky přitom přehrává jeho obsah.

1.4 Způsoby přehrávání

Přehrávání online médií, konkrétně rádiové vysílání, je možné realizovat mnoha způsoby.

V případě přehrávání na běžném počítači můžeme metody rozdělit na přehrávání v rámci desktopových přehrávačů a v rámci „webu“. Samostatnou kapitolou by mohla být mobilní zařízení, kde se možnosti liší dle používaného operačního systému.

1.4.1 Desktopové přehrávače

Desktopové přehrávače jsou mezi uživateli velmi oblíbené pro svůj zažitý způsob používání. Obvykle vyžadují instalaci na cílový počítač a dokáží pracovat s lokálním uložištěm (pevný disk). Veškerá nastavení jsou ukládána lokálně. Nabízejí širší možnosti integrace do systému, ať už je to možnost minimalizace do tray nebo nastavení aplikace jako výchozí pro otevírání určitého typu souborů. Tento typ přehrávačů je ideální pro uživatele, kteří pracují stabilně s jedním konkrétním počítačem a nevyužívají současně více zařízení, mezi kterými by chtěli data a svá nastavení sdílet.

1.4.2 Přehrávání na webu

Přehrávání médií online začíná být stále populárnější, a to nejen v podobě jednorázového spuštění internetového rádia na webu rádiové stanice, ale spíše v oblasti, která je trendem dneška – „cloud-computingu“.

Online přehrávače médií nenabízejí jen ryze internetová média, ale dokáží přehrávat už i lokální hudbu, kterou uživatel musí pochopitelně nejprve importovat do pomyslného „mraku“. Některým to možná připadá nesmyslné a přinejmenším zbytečné, ale mít svou hudbu online sebou nese mnoho výhod. Největším přínosem tohoto konceptu je možnost

přistoupit ke svému hudebnímu archívu odkudkoliv, třeba jen prostřednictvím webového prohlížeče.

Tento přístup je tak ideální pro ty, kteří využívají více pracovních stanic a mají rádi vše na jednom místě. U všech online přehrávačů samozřejmě odpadá také nutnost cokoli instalovat a složitě nastavovat.

Ať už se jedná jen o přehrávač konkrétní rádiové stanice, nebo komplexní aplikaci pro přehrávání hudby, vždy je cíleno na přívětivé uživatelské prostředí a jednoduchou ovladatelnost.

2 NAHRÁVÁNÍ ONLINE MÉDIÍ

Pořizování nahrávek nám přináší schopnost opakovaného poslechu nebo shlédnutí vybrané části média, ta může být uchována pro pozdější použití.

2.1 Přístupy k nahrávání

Nahrávání online médií, zvuku nebo obrazu, se dá realizovat dvěma způsoby. Jedna z možností je zaznamenávat výstup reprodukcího nebo zobrazovacího zařízení, který získáme při přehrávání zdroje. Tohoto způsobu využívají mnohé aplikace. Jde v praxi o případ, kdy si třeba naladíme internetové rádio. Přehrávač provádí přehrávání a výstup je slyšet ze zvukové karty. V tomto okamžiku nám nebrání nic tomu, abychom tento výstup zachytávali a ukládali na disk počítače, buď za pomoci speciální, k tomu určené, aplikace nebo přímo v přehrávači internetových rádií, pokud možnost nabízí.

Nevýhodou tohoto postupu je nutnost správně nakonfigurovat zvukovou kartu tak, aby se skutečně zaznamenával výstup z přehrávače, a ne třeba jiný zdroj, kupříkladu mikrofon. Rozhodující je určitě i správné nastavení úrovně hlasitosti tak, aby nebyla nahrávka příliš potichu nebo naopak přebuzená. Z uvedeného vyplývá, že výsledný soubor může mít různou kvalitu a jeho podoba je nejednoznačná a závislá na nastavených podmínkách. Nahrávání výstupu zvukové karty se při záznamu encoduje na zvolený formát a datový tok. Pokročilejší aplikace dokáží podmínky nastavení dodržet pro každé zahájené nahrávání tím, že si hodnoty pamatují a před spuštěním zaznamenávání provedou konfiguraci zvukové karty. V některých případech dokonce může dojít k přimíchání jiných zvuků, které se objeví na výstupu zvukové karty. Nahrávka v takovém případě může obsahovat i nežádoucí systémové zvuky nebo audio výstupy jiných aplikací.

Alternativou, jejíž implementace je také dále demonstrována v ukázkové aplikaci, je nahrávání přímo ze zdroje streamu. Jedná se o postup, vyžadující implementaci přímo na úrovni přehrávacího engine. Metodě, která se stará o zpracovávání přijatých dat ze stream serveru, je předán ukazatel na samostatnou metodu, která dokáže data odchytávat ještě před jejich zasláním na zvukové zařízení. Jak je z principu tohoto postupu vidět, tak odpadá jakékoliv konfigurování zvukové karty. Pořízené nahrávky v rámci jednoho streamu navíc mají shodnou úroveň kvality a hlasitosti.

2.2 Digital rights management

Některé zdroje internetových médií jsou opatřeny ochranou proti kopírování a pořizování záznamu, jedná se o technologii nazvanou DRM (Digital Rights Management). Nasazování těchto metod je mezi uživateli velmi nepopulární, také proto se od tohoto modelu postupně upouští. Nicméně určitě je třeba se alespoň okrajově s touto problematikou seznámit. Princip spočívá v serverové a klientské podpoře konkrétní technologie, která si klade za cíl zpravidla kontrolovat a omezovat používání digitálního obsahu, hlavně obsahu s požadavkem na dodržování autorského práva. Obecně se pracuje s restrikcemi typu počet přehrání média, možnost vytvářet rozmnoženiny, omezení počtu kopií, vymezení užití díla na předepsanou dobu, apod. Vždy se uplatnění tohoto nastavení váže na konkrétní pracovní stanici nebo případně uživatele.

Nutno však podotknout, že sebelepší technologie ochrany digitálního obsahu ztrácí význam v okamžiku, kdy dojde k platnému přehrání tohoto obsahu na klientském zařízení, kde uživatel využije metodu záznamu popsanou výše (například záznam zvuku ze zvukové karty). Na takto pořízenou nahrávku již omezení definovaná DRM neplatí. Takový postup rozmnožování díla je ale vždy v rozporu s podmínkami použití.

2.3 Autorský zákon

Dle autorského zákona o provozování díla ze záznamu a přenosu provozování díla by uživatel pořizováním záznamu zasahoval do autorského zákona:

§18 (1) „*Sdělováním díla veřejnosti se rozumí zpřístupňování díla v nehmotné podobě, živě nebo ze záznamu, po drátě nebo bezdrátově.*“

§18 (2) „*Sdělováním díla veřejnosti podle odstavce 1 je také zpřístupňování díla způsobem, že kdokoli může mít k němu přístup na místě a v čase podle své vlastní volby zejména počítačovou nebo obdobnou sítí.*“

§18 (3) „*Sdělováním díla veřejnosti není pouhé provozování zařízení umožňujícího nebo zajišťujícího takové sdělování.*“ [21]

Nicméně to jsou pouze některé ze způsobů užití díla sdělováním veřejnosti. Dalšími způsoby, na něž se již legislativní omezení volného užití nevztahuje, je například:

Vysílání rozhlasem nebo televizí (§21 autorského zákona)

Přenos rozhlasového nebo televizního vysílání (§22 autorského zákona)

Provozování rozhlasového či televizního vysílání (§23 autorského zákona)

Pojmy provozování ze záznamu a přenos provozování ze záznamu jsou autorským zákonem vymezeny paragrafem §20 - Provozování ze záznamu a jeho přenos:

§20 (1) „*Provozováním díla ze záznamu se rozumí zpřístupňování díla ze zvukového nebo zvukově obrazového záznamu pomocí přístroje, s výjimkou užití díla způsoby podle §21 až 23.*“

§20 (2) „*Přenosem provozování díla ze záznamu se rozumí současné zpřístupňování díla ze záznamu pomocí reproduktoru, obrazovky nebo podobného přístroje umístěného mimo prostor provozování ze záznamu.*“ [21]

Z uvedených paragrafů autorského zákona vyplývá, že pokud dochází k vysílání audiovizuálního díla rozhlasem nebo televizí, tak lze zhotovit jeho rozmnoženinu určenou k osobnímu užití.

3 SOUVISEJÍCÍ TECHNOLOGIE

3.1 Ajax

AJAX (Asynchronous JavaScript and XML) je obecný termín pro technologie používané při vývoji interaktivních webových aplikací, které dynamicky na pozadí komunikují se vzdáleným serverem a svůj obsah dokáží měnit bez nutnosti znovunačítání celých stránek. Takové aplikace jsou uživatelsky přívětivé, vyžadují ale použití moderních webových prohlížečů.

Využívá se následujících technologií:

- HTML (nebo XHTML) a CSS pro přívětivou prezentaci dat
- DOM a JavaScript pro zobrazování a dynamické změny prezentovaných informací
- XMLHttpRequest pro asynchronní výměnu dat se serverem
- Jako formát vyměňovaných dat se využívá XML, JSON, HTML a jiné

Jak je vidět, tak Ajax není konkrétní jednotlivá technologie, ale termín označující využití několika moderních technologií pro dosažení konkrétní funkcionality.

Hlavní výhodou je nenutnost načítání celých stránek při jakékoliv změně obsahu nebo interakci uživatele, což vede ke snížení zatížení serveru a větší dynamičnosti aplikace. [5]

3.2 Apache

Apache je softwarový webový server, který je dostupný pro většinu používaných platform operačních systémů. V současné době se jedná o nejpoužívanější webový server na světě. Aktuálně servery Apache obsluhují více jak 59,13% všech stránek na internetu. [19]

Lze jej charakterizovat jako open source software, jeho udržování a vývoj je zastřešován otevřenou komunitou Apache Software Foundation.

Apache podporuje celou řadu funkcí, mnoho jich je implementováno formou modulů, které rozšiřují základní funkce jádra serveru. Moduly mohou rozšiřovat například podporu server-side programovacích jazyků nebo metody ověřování, zabezpečení, komprimace apod. Oblíbenými jazyky jsou například Perl, Python, Tcl nebo PHP.

3.3 CSS

Kaskádové styly (Cascading Style Sheets) je jazyk používaný k popisu zobrazení vzhledu a formátování dokumentů napsaných v značkovacích jazycích. Nejčastěji je používán k nastýlování webových stránek napsaných v HTML a XHTML, může být ale aplikován na jakýkoli druh XML dokumentu, včetně prostého XML.

CSS slouží k oddělení obsahové stránky dokumentu od grafické reprezentace prvků, jako je rozložení, barva a písmo. Napomáhá zpřehlednění dokumentu, jeho jednoduché přenositelnosti a snadné změně vzhledu.

Jelikož je oddělena obsahová část od grafické, je možné dokument logicky strukturalizovat a umožnit tak větší flexibilitu v poskytování a reprezentaci obsahu. Dokument může být na jedné straně jednoduše rozpoznán zobrazovacím zařízením pro nevidomé, a na druhé straně jej lze zobrazit v mnoha rozmanitých grafických podobách, a to vše bez zbytečných duplicít kódu.

Definice kaskádových stylů je tvořena několika pravidly. Každé pravidlo obsahuje selektor a blok deklarací. Každý blok deklarací dále obsahuje seznam deklarací oddělených středníky „;“ a každá deklarace sestává z identifikátoru vlastnosti, následuje dvojtečka „:“ a hodnota vlastnosti. Nepovinné je označení „!important“, které zvyšuje sílu deklarace. Způsob použití je znázorněn na obrázku číslo 2. [6]

```
selektor [ , selektor2 , ... ] [ :pseudo-class ] {  
    identifikátor : hodnota ;  
    [ identifikátor2 : hodnota2 ;  
    ... ]  
}  
/* komentář */
```

Obrázek 2 - Definice kaskádových stylů [vlastní zpracování]

3.4 HTML

HyperText Markup Language, označovaný zkratkou HTML, je převládajícím značkovacím jazykem pro vytváření webových stránek. HTML je základním stavebním kamenem webu, umožňuje jednoduchou prezentaci obsahu na internetu.

Cílem webových prohlížečů je číst HTML dokumenty a skládat je do vizuální podoby. Prohlížeče nezobrazují samotné HTML tagy, ale používají je k interpretaci obsahu stránky.

HTML umožňuje vkládání obrázků a jiných objektů, které přináší široké možnosti interakce přímo na webu. Poskytuje prostředky pro vytváření strukturovaných dokumentů, kde je významnost textu rozlišována jako nadpisy, odstavce, seznamy, odkazy, citace a další položky. Podporuje využívání skriptů, například jazyka JavaScript, kterým je docíleno možnosti ovlivňovat chování webových stránek.

Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které měly zpětně vliv na definici jazyka, a to dle všeobecných požadavků na funkčnost a postupný vývoj, kterým tato oblast prošla.

3.5 JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk. Autorem je Brendan Eich, který v minulosti působil ve společnosti Netscape.

V současnosti se obvykle využívá jako interpretovaný programovací jazyk pro webové stránky. Často je vkládáný přímo do HTML kódu stránky a je tak prováděn na straně klienta, přesněji řečeno ve webovém prohlížeči. Jsou jím ovládány nejrozličnější interaktivní prvky grafického rozhraní. Od tlačítek, validace hodnot textových polí až po animace a grafické efekty.

Syntaxí připomíná jazyky C/C++/Java. Slovo Java v názvu má pouze marketingový důvod a s programovacím jazykem Java má společnou pouze podobnou syntaxi. V roce 1997 byl standardizován asociací ECMA (European Computer Manufacturers Association) a v roce 1998 ISO (International Organization for Standardization). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a byly z ní odvozeny i další implementace, jako je například ActionScript.

Díky tomu, že je JavaScriptu obvykle prováděn až po načtení stránky, na straně klienta, plynou z toho určitá bezpečnostní omezení. JavaScript například nemůže pracovat s lokálními soubory, aby tím nebylo ohroženo soukromí uživatele a bezpečnost systému.

Zajímavostí je možnost provádět JavaScript i na straně serveru, jako je tomu u ostatních interpretovaných jazyků (například PHP nebo ASP). Existuje několik projektů, které se

touto možností zabývají (například Rhinola nebo LiveWire). Tento typ použití však není příliš rozšířený. [7]

3.6 jQuery

jQuery je JavaScriptová knihovna navržena pro snadné skriptování HTML na straně klienta. Výhodou je multi-prohlížečová podpora a její obecnost.

Byla zpřístupněna v lednu 2006 na BarCamp v New Yorku Johnem Resignem.

Knihovna je používána na více jak 43% z 10 000 nejnavštěvovanějších webů a je nejpoužívanější JavaScriptovou knihovnou současnosti.

Syntaxe jQuery je navržena tak, aby co nejvíce usnadnila orientaci v dokumentu, vybírání DOM elementů, tvorbu animací, zpracování událostí a zjednodušila vývoj Ajax aplikací.

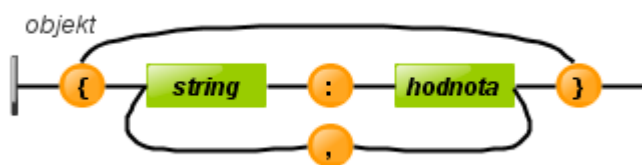
Podobně jako CSS odděluje „zobrazovací“ charakteristiky od struktury HTML, jQuery odděluje „chování“ od struktury HTML. Kupříkladu místo specifikace onClick události přímo v HTML kódu ovládacího prvku by stránka řízená jQuery nejprve našla vhodný element prvku a potom změnila jeho manipulátor události. Takovéto oddělení chování od struktury se také často nazývá jako princip nevtíravého JavaScriptu. [8]

3.7 JSON

JSON (JavaScript Object Notation, JavaScriptový objektový zápis) je odlehčený formát pro výměnu dat. Je jednoduše čitelný i zapisovatelný člověkem a snadno analyzovatelný i generovatelný strojem. Je založen na podmnožině programovacího jazyka JavaScript. JSON je textový, na jazyce zcela nezávislý formát, využívající však konvence dobře známé programátorům jazyků rodiny C (C, C++, C#, Java, JavaScript, Perl, Python a dalších). Díky tomu je JSON pro výměnu dat opravdu ideálním jazykem.

JSON je založen na dvou strukturách:

- Kolekce párů název-hodnota. Ta bývá v rozličných jazycích realizována jako objekt, záznam (record), struktura (struct), slovník (dictionary), hash tabulka, klíčový seznam (keyed list) nebo asociativní pole.



Obrázek 3 - JSON - kolekce páru název-hodnota [vlastní zpracování]

- Tříděný seznam hodnot. Ten je ve většině jazyků realizován jako pole, vektor, seznam (list) nebo posloupnost (sequence).



Obrázek 4 - JSON - tříděný seznam [vlastní zpracování]

Jedná se o univerzální datové struktury a v podstatě všechny moderní programovací jazyky je v nějaké formě podporují. Je tedy logické, aby na nich byl založen i na jazyce nezávislý výměnný formát, jakým JSON je. [9]

3.8 .NET

Základní komponentou .NET je Microsoft .NET Framework, jedná se o prostředí potřebné pro běh aplikací a nabízející jak spouštěcí rozhraní, tak potřebné knihovny.

Platforma .NET Framework se skládá ze dvou základních částí:

- CLR - společné běhové jazykové prostředí pro běh aplikací
- FCL - knihovna Framework Class Library

Platforma .NET Framework je součástí snahy maximálně usnadnit vývoj služeb i aplikací, s cílem především na platformu operačního systému Windows. [10]

Dostupné platformy

- **Microsoft .NET Framework** - určeno pro osobní počítače s operačním systémem Windows (Windows 98 a vyšší), nejrozšířenější platforma
- **Microsoft .NET Compact Framework** - určeno pro kapesní počítače a mobilní telefony s operačním systémem Windows Mobile
- **Microsoft .NET Micro Framework** - platforma pro embedded zařízení s ještě menším výpočetním výkonem než mobilní zařízení
- **Mono** - produkt nezávislé open source iniciativy, jedná se o implementaci .NET runtime pro operační systémy Unixového typu (Linux, MacOS X)

Pro vývoj .NET aplikací je využíváno vývojové prostředí Microsoft Visual Studio .NET.

Platforma .NET není vyhrazená co se týče používání programovacího jazyka. Zdrojový kód aplikace, ať už je napsán v jakémkoliv jazyce, je vždy přeložen do tzv. mezijazyka Common Intermediate Language. [11]

Nejpoužívanější programovací jazyky pro vývoj .NET aplikací jsou:

- C#
- Visual Basic .NET

Další používané jazyky:

- Delphi
- Managed C++
- F# - funkcionální programovací jazyk
- J# - jazyk podobný Javě
- IronPython
- Boo

[12], [13]

3.9 MySQL

Jedná se o databázový systém s multiplatformní podporou. Komunikace a práce s daty probíhá pomocí jazyka SQL. Dalo by se říct, že jde o dialekt tohoto jazyka, kdy MySQL přináší nová rozšíření. Díky jednoduché použitelnosti, volné licenci a svému výkonu je v současnosti velmi rozšířeným databázovým systémem používaným především v prostředí internetových aplikací. Oblíbené je zejména použití v následující kombinaci: Linux, Apache, MySQL a PHP, často uváděné jako technologie LAMP.

MySQL si kladlo za cíl optimalizaci hlavně v oblasti rychlosti, na úkor toho byly některé funkce a možnosti v prvotních verzích opomenuty, jejich podpora byla slabá nebo vůbec žádná (pohledy, trigger, apod.). Aktuální verze je již však zahrnují.

3.10 PHP

PHP je hypertextový preprocesor (Hypertext Preprocessor, původně Personal Home Page), který na straně serveru interpretuje stránky HTML s vlastními příkazy před jejich odesláním ke klientovi (webový prohlížeč). Jedná se o interpretovaný jazyk a je primárně určen pro programování dynamických webových stránek. To, že je jazyk PHP interpretovaný znamená, že je až do chvíle svého použití (spuštění) na serveru uchován v podobě zdrojového kódu. Překlad do strojového kódu probíhá až interpretem tohoto jazyka, což je pochopitelně o něco pomalejší, než v případě kompilovaného kódu, který je k dispozici ihned ve spustitelné podobě. [14]

Syntaxe jazyka je inspirována hned několika různými programovacími jazyky - Perl, C, Pascal a Java. PHP není závislý na platformě, odlišnosti v různých operačních systémech jsou minimální a omezují se pouze na několik funkcí přímo závislých na daném operačním systému. Skripty lze tak většinou bez jakýchkoliv úprav používat na různých platformách.

Velkou výhodou je podpora mnoha knihoven určených pro nejrůznější účely (zpracování textu, grafické knihovny, práce se soubory, práce s databázovými systémy – MySQL, MSSQL, SQLite3, MongoDB, Oracle, PostgreSQL). Nechybí ani široká podpora internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, apod.).

Díky tomu, že je PHP velmi jednoduchý, nabízí široké množství funkcí a kombinuje v sobě několik oblíbených programovacích jazyků, tak je v současnosti nejrozšířenějším

skriptovacím jazykem na webu. Jeho podíl v této oblasti je 76,2 %. Jako v pořadí další skriptovací jazyky jsou ASP.NET s 22,7 % a také Java s 3,8 %. [15]

3.11 SQL

SQL (Structured Query Language) je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích.

Jazyk SQL je podporován v podstatě každou relační databází. Obvykle však ale nejsou implementovány vždy všechny požadavky jeho normy. Naopak každý databázový systém přidává vlastní možnosti a konstrukce, které běžně nejsou ve standardech uvedeny. Z toho důvodu není kompatibilita dotazů napříč různými databázemi plně zaručena.

Příkazy lze podle svého účelu rozdělit do následujících hlavních skupin:

- Manipulace s daty (SELECT, INSERT, UPDATE, DELETE, ...)
- Definice dat (CREATE, ALTER, DROP, ...)
- Řízení přístupových práv (GRANT, REVOKE)
- Řízení transakcí (START TRANSACTION, COMMIT, ROLLBACK)
- Ostatní příkazy - dle databázového systému [16]

3.12 XML

XML (Extensible Markup Language, rozšiřitelný značkovací jazyk) je soubor pravidel pro kódování dokumentů v podobě čitelné jak pro člověka, tak také pro počítače.

Cílem návrhu XML je jednoduchost, obecnost, a použitelnost v prostředí internetu.

Jedná se o textový formát dat se silnou podporou kódování pomocí znakové sady Unicode (lze využít i libovolné jiné kódování, které je však nutné v dokumentu definovat). Tím je zajištěna jeho univerzálnost pro všechny světové jazyky.

Přestože se XML zaměřuje na dokumenty, je často používán pro reprezentaci libovolných datových struktur, nejčastěji u internetových služeb. Z XML vychází velké množství nejrozličnějších jazyků a je často používán jako komunikační jazyk pro nejrozličnější programová rozhraní aplikací (API). [17]

4 AUDIO FORMÁTY

4.1 Mp3

Mp3 patří mezi nejznámější a nejrozšířenější formáty využívající ztrátovou kompresi. Byl navržen tak, aby snižoval datový tok a přitom zněl pro většinu posluchačů co možná nejvěrněji vzhledem k originální bezztrátové nahrávce.

Komprese funguje na principu odebírání informací, které člověk běžně neslyší nebo si je neuvědomuje. Tento způsob je nazýván jako „psychoakustický“ model.

Platí, že s rostoucím datovým tokem se zvyšuje kvalita. V oblasti streamingu internetových rádií je tento formát využíván zejména na stream serverech SHOUTcast a IceCast. Vysílání většinou probíhá v datových tocích 32, 48, 64, 96, 128, 192, 256 až 320 kbps.

4.2 AAC

Advanced Audio Coding je standard pro ztrátovou kompresi zvuku. Byl navržen tak, aby se stal nástupcem formátu Mp3. AAC obvykle dosahuje lepší kvality zvuku než Mp3 při podobných přenosových rychlostech. Nejedná se o jednotný formát, existuje jeho několik modifikací. Patří mezi nejpokročilejší kompresní formáty. Pro vysílání internetových rádií se používá konkrétně varianta AAC+.

4.3 Ogg Vorbis

Ogg Vorbis je ztrátovým audio formátem kladoucím si za cíl nahradit zastaralý formát Mp3. Kvalitou se blíží například formátu WMA. Při shodném datovém toku zní výrazně lépe než Mp3.

4.4 WMA

Windows Media Audio je komprimovaný zvukový formát vyvinutý jako součást Windows Media. Od verze 9 je jeho kvalita srovnatelná kupříkladu s Ogg Vorbis. Na operačním systému Windows je přímo integrovaný do WMP, na ostatních platformách je nutné doinstalovat potřebné pluginy.

II. PRAKTICKÁ ČÁST

5 DATABÁZE RÁDIÍ

Databáze rádií je serverová část zajišťující centrální správu seznamu rádiových stanic. Jedná se o seznam rádií šířených prostřednictvím sítě internet, k nimž jsou přiřazeny streamy reprezentující distribuční kanály v konkrétních audio formátech a datových tocích, uváděných jako bitrate.

5.1 Návrh databáze

Jako databázový systém byla zvolena relační databáze MySQL, a to z důvodu nabízených možností a její optimální výkonnosti pro předpokládaný malý rozsah projektu v počtu záznamů v databázi. Rozhodujícím faktorem byla také její dostupnost a oblíbenost.

Dále bude databáze internetových rádií popisována jako služba, jejíž metody jsou dostupné prostřednictvím definovaného API rozhraní. Toto rozhraní je prakticky využito v ukázkové aplikaci.

5.2 Rozhraní API

Rozhraní API slouží k poskytování služeb umožňujících využívání centrální databáze v prostředí klientské aplikace. Je poskytováno prostřednictvím HTTP protokolu, konkrétně na portu 80, což je výhodné zejména v případě používání aplikace v síti s omezeným přístupem k internetu.

Jako datový formát pro komunikaci je použito XML se znakovou sadou UTF-8 (Unicode/Universal Character Set).

5.2.1 Inicializační rozhraní

Inicializační rozhraní slouží jako hlavní služba API, která poskytuje veškerá data potřebná k inicializaci klientské aplikace. Autorizace k tomuto rozhraní je realizována prostřednictvím unikátního identifikátoru (klíče), který je jedinečný pro každou instalaci aplikace (v URL adrese je uváděn jako parametr *uid*).

Identifikátor je vygenerován na straně klienta při prvním spuštění aplikace na uživatelské pracovní stanici. Dojde-li k přístupu nově instalované aplikace k inicializačnímu rozhraní

pomocí nového identifikátoru, nejprve na straně serveru dojde k jeho validaci a následnému založení do databáze.

Přístupovým bodem k tomu rozhraní je webová adresa ve tvaru:

http://server/login/?uid=unikatni_identifikator

Odpověď v případě nekorektního požadavku

Pokusí-li se k API rozhraní přistoupit kdokoliv bez platného unikátního identifikátoru, nejsou mu služby dostupné. Odpověď v takovém případě obsahuje informaci o neúspěšném přihlášení, numerické označení chyby a také textový popis problému.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <netradio>
  <login status="0" />
- <error>
  <code>1</code>
  <message>Neplatné přihlášení</message>
</error>
</netradio>
```

Obrázek 5 – API rozhraní - chybné přihlášení [vlastní zpracování]

Význam jednotlivých elementů

- *login* – Atribut *status* tohoto elementu informuje o korektní nebo nekorektní autorizaci, konkrétně 0 vypovídá o neschopnosti spolupracovat s API rozhraním pomocí zasláního autorizačního identifikátoru.
- *error* – Přítomnost tohoto elementu značí chybu, její bližší specifikace je uvedena v následujících elementech:
 - *code* – Numerické určení problému, nebo také kód chyby, číslo 1 informuje o neplatném přihlášení z důvodu nekorektního identifikátoru.
 - *message* – Jedná se o textový popis problému, slouží jako text určený k zobrazení v klientské aplikaci jako odůvodnění vzniklého problému.

Korektní odpověď

Dojde-li ke korektnímu přístupu k inicializačnímu rozhraní, tak je odpověď serveru v následujícím tvaru:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <netradio>
  - <login status="1">
    <key>edb2ae71fc23f546a8ef71fb237688df</key>
  </login>
  - <formats>
    <format id="1" ext="mp3">Mp3</format>
    <format id="2" ext="ogg">Ogg</format>
    <format id="3" ext="aac">Aac+</format>
    <format id="4" ext="wma">Wma</format>
  </formats>
  - <stations>
    - <station id="1">
      <name>Název rádia 1</name>
      <logo>http://www.adresa-serveru.cz/images/logo.png</logo>
      <description>Krátký popis rádiové stanice</description>
      - <streams>
        <stream id="1" format="1" bitrate="128" rec="1">http://.../p3</stream>
        <stream id="3" format="3" bitrate="64" rec="1">http://.../3</stream>
        <stream id="5" format="3" bitrate="32" rec="1">http://.../5.m3u</stream>
        <stream id="7" format="4" bitrate="128" rec="0">http://.../7.asx</stream>
        <stream id="8" format="4" bitrate="32" rec="0">http://.../8.asx</stream>
        <stream id="6" format="4" bitrate="16" rec="0">http://.../6.asx</stream>
      </streams>
    </station>
    - <station id="2" user="1">
      <name>Název rádia 2</name>
      <logo>http://www.adresa-jineho-serveru.cz/images/logo.png</logo>
      <description>Krátký popis rádiové stanice číslo 2</description>
      - <streams>
        <stream id="21" format="1" bitrate="128" rec="1">http://.../21.p3</stream>
        <stream id="32" format="3" bitrate="64" rec="1">http://.../32</stream>
        <stream id="25" format="3" bitrate="32" rec="1">http://.../25.m3u</stream>
      </streams>
    </station>
  </stations>
+ <timemachines>
</netradio>
```

Obrázek 6 - API rozhraní - korektní odpověď [vlastní zpracování]

Význam jednotlivých elementů

- *login* – Atribut *status* tohoto elementu, obsahující hodnotu 1, nás informuje o korektním přihlášení k API rozhraní, dále je k dispozici element *key* s informací o klíči, pomocí kterého jsme se korektně přihlásili.
- *formats* – Elementy *format* obsažené v této části představují výčet všech dostupných audio formátů, přičemž význam uvedených atributů je následující:
 - *id* – Číselný identifikátor formátu.
 - *ext* – V případě nahrávání streamu, který je vysílán v tomto formátu, předepisuje koncovku výsledného fyzického souboru (např. *nahravka.mp3*).
- *stations* – Poskytuje výpis všech dostupných rádiových stanic pro daný unikátní klíč, kterým se připojení autorizovalo. Jsou to jednak veřejné rádiové stanice a také rádia, která si uživatel přidal pod tímto klíčem.

Struktura definice rádiové stanice

Každé rádio je reprezentováno jedním elementem *station*, patřícím pod element *stations*. Element *station* obsahuje atribut *id* s unikátním číselným označením stanice v rámci celé databáze.

Jedná-li se o stanici, která byla přidána pod stejným klíčem, pod jakým je přistupováno k tomuto autorizačnímu rozhraní, tak je přítomný také atribut *user* s hodnotou 1 vypovídající o oprávnění editovat tento záznam v rámci aktuálně používaného klíče (z pohledu ukázkové aplikace si tuto stanici přidal sám uživatel).

Následují tyto elementy:

- *name* – Pojmenování rádiové stanice, dále se zobrazuje přímo v ukázkové aplikaci v nabídce předvolených stanic.
- *logo* – Volitelný element s URL adresou směřující na obrázek loga rádia v rámci sítě internet. Obrázky mohou být ve formátu JPG, PNG nebo GIF a jsou použity dále v ukázkové aplikaci pro zpříjemnění procházení nabídek stanic.

- *description* – Krátký textový popis nebo slogan rádia, může nést informaci o stylu nebo zaměření rádia. Délka textu je maximálně 255 znaků.
- *streams* – Každé rádiové stanici je přiřazen seznam dostupných variant vysílání, každá varianta má vlastní element *stream* s těmito parametry:
 - *id* – Označení streamu číselným identifikátorem, unikátní v rámci celé databáze, nikoliv pouze v rámci nadřazené stanice, což zjednodušuje přístup k této položce napříč celou databází.
 - *format* – Číselné označení formátu vysílání tohoto streamu, odkazuje na některou konkrétní variantu formátu ze seznamu formats.
 - *bitrate* – Datový tok vysílání uváděný v kbps (kilobit za sekundu). Společně s určením formátu vypovídá o subjektivní kvalitě přehrávání. Neznáme-li formáty streamů, nemůžeme pouze podle hodnoty datového toku rozhodnout, zda je stream s vyšší hodnotou kvalitnější než jiný. Bitrate nám pouze říká, jaký bude mít vysílání nárok na zatížení linky internetového připojení.
 - *rec* – Povoluje nebo zakazuje provádět nahrávání tohoto zdroje vysílání. Hodnota 1 nahrávání povoluje, 0 naopak zakazuje. Tento příznak se v ukázkové aplikaci zohledňuje a řídí vykreslování některých ovládacích prvků.

Hodnota každého elementu *stream* obsahuje informaci o URL adrese přípojného bodu vysílání, zpravidla se jedná o přímý odkaz na stream server nebo případně na seznam více takových odkazů (tzv. playlist).

V prvním uvedeném případě má URL adresa následující tvar:

http://stream_server:port/pripojny_bod

Část *pripojny_bod* je u některých stream serverů vynechána, konkrétně jde o starší verze Shoutcast serverů, kde byla nutná pro každý samostatný stream server zvláštní instance aplikace s unikátním portem. Poslední dobou je ale běžnější využívání tzv. přípojných bodů, které rozlišují více streamů v rámci jedné instance stream serveru na jediném přístupovém

portu, obvykle 80. Tím lze vynechat část port z uvedené ukázkové URL adresy, jde totiž o výchozí port http komunikace.

5.2.2 Editace položky

Přidávání nových rádiových stanice nebo úpravu těch dříve přidaných je možno provádět prostřednictvím služby editace položek. Práci s touto službou demonstruje následující ukázka:

```
http://server/edit/?uid=unikatni_identifikator&id=11&name=Nazev_radia&description=popis&img=adresa_loga&url[0]=stream_url1&format[0]=5&bitrate[0]=192&url[1]=stream_url2&format[1]=5&bitrate[1]=128
```

Popis možných parametrů (koresponduje s popisem uvedeným v inicializačním rozhraní):

- *uid* – Unikátní identifikátor.
- *id* – Identifikátor rádiové stanice, která se bude upravovat. Pro přidání nového rádia je třeba nastavit hodnotu 0.
- *name* – Název rádiové stanice.
- *description* – Krátký popis.
- *img* – URL adresa umístění loga rádia.
- *url[0]* – URL adresa prvního streamu, *url[1]* - URL adresa druhého streamu, ...
- *format[0]* – Identifikátor formátu prvního streamu, *format[1]* – Identifikátor formátu druhého streamu, ...
- *bitrate[0]* – Datový tok prvního streamu, *bitrate[1]* – Datový tok druhého streamu, ...

Odpověď serveru po korektním provedení přidání nové stanice nebo úpravě existující, je velmi jednoduchá:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <netradio>
  - <login status="1">
    <key>edb2ae71fc23f546a8ef71fb237688df</key>
  </login>
  <station>11</station>
</netradio>
```

Obrázek 7 - API rozhraní – editace [vlastní zpracování]

Důležitý je pouze element *station* nesoucí informaci o ID rádiové stanice, která byla korektně upravena, nebo případně ID nově přidaného rádia. Takto přidaná stanice je samozřejmě ihned dostupná prostřednictvím inicializačního rozhraní po autorizaci stejným klíčem, pod kterým byla editační služba zavolána.

Stejně jako inicializační rozhraní, tak i editační může vracet chybu v případě neplatného přihlášení (Obrázek 5). Navíc ale může dojít k situaci, kdy se pokoušíme ovlivňovat rádiovou stanici, u níž nemáme práva k její editaci (např. předdefinované rádiové stanice nebo rádia jiných uživatelů). V těchto případech vrací rozhraní následující chybu:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <netradio>
  - <login status="1">
    <key>edb2ae71fc23f546a8ef71fb237688df</key>
  </login>
  - <error>
    <code>2</code>
    <message>Nemáte oprávnění editovat tuto rádiovou stanici</message>
  </error>
</netradio>
```

Obrázek 8 - API rozhraní - chyba oprávnění [vlastní zpracování]

Struktura chybového hlášení je standardní: element *error* s položkami *code* a *message*. Tento druh chyby má konkrétně identifikátor 2. Při běžném používání ukázkové aplikace by k vyvolání tohoto stavu nemělo dojít, dodržování oprávnění u dílčích položek je hlídáno.

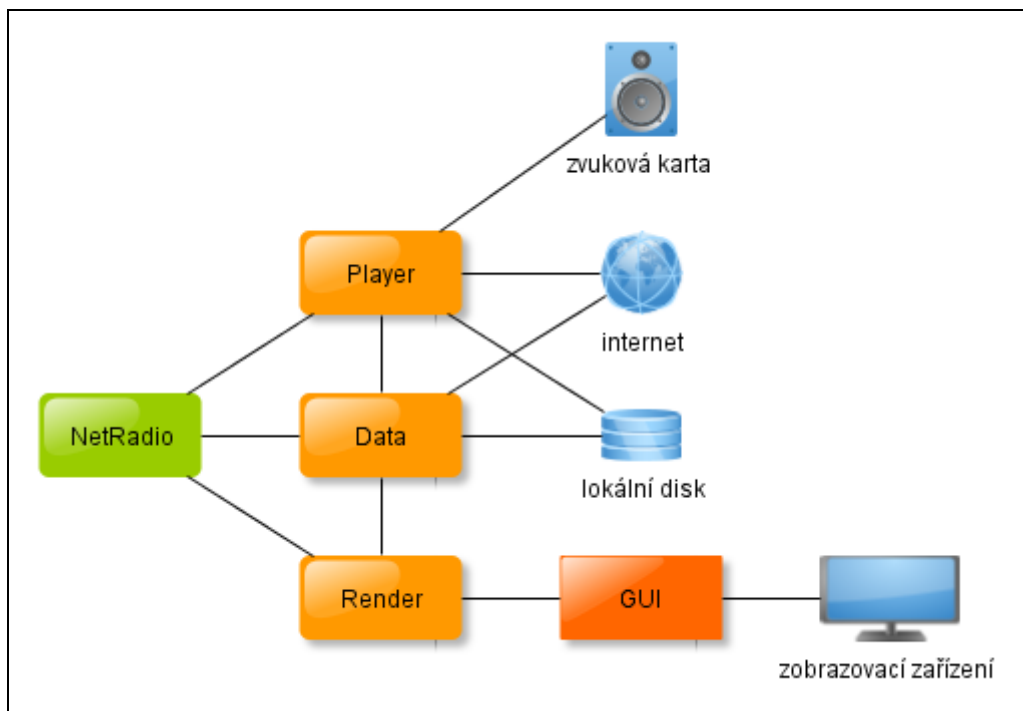
6 UKÁZKOVÁ APLIKACE

Hlavním cílem bylo navrhnout a následně implementovat jednoúčelovou aplikaci zaměřenou na přehrávání internetových rádií s možností jejich nahrávání na pevný disk počítače. Aby výsledek nebyl pouze demonstrací popisovaných technologií, ale praktickým produktem schopným reálného fungování a provozu v konkurenčním prostředí, bylo nutné vytyčit několik základních a cílených vlastností.

- Velmi jednoduché a intuitivní ovládání
- Primitivní nebo žádná instalace
- Bez nutnosti cokoliv nastavovat, ať už v samotné aplikaci nebo dokonce snad v systému
- Funkce nahrávání jedním tlačítkem bez zásahu do nastavení zvukové karty
- Předdefinovaný základní seznam stanic obsahující minimálně česká a slovenská rádia
- Podpora co nejvíce hudebních formátů

6.1 Základní návrh aplikace

Celou aplikaci lze rozdělit do tří základních objektů, které společně zajišťují veškeré funkce programu. Konkrétně se jedná o objekty *player*, *render* a *data*. Rozvržení a úlohy jednotlivých částí jsou znázorněny na obrázku 9.



Obrázek 9 - Základní návrh aplikace [vlastní zpracování]

Objekt *player* je tvořen podle třídy *cPlayer* a zajišťuje veškeré funkce týkající se přehrávání a komunikace se zvukovou kartou. Nechybí také podpora pro nahrávání. Důležitou vlastností tohoto objektu je možnost vyvolávat události, které jsou iniciovány různými procesy v průběhu činnosti přehrávače. Jejich zpracování v rámci dalších částí aplikace tak ulehčuje a zpřehledňuje implementaci dílčích funkcí. Bližší popis této části je uveden v kapitole 6.2. *Render* se stará o grafický výstup a obecně o uživatelské prostředí a jeho správné vykreslování. Objekt *data* plní úlohu správce veškerých datových zdrojů. Zahrnuje podporu komunikace se vzdálenou databází rádiových stanic a jejím API rozhraním (viz kapitola 5.2). Provádí také ukládání a obnovu uživatelského nastavení v rámci aplikace.

6.2 Implementace přehrávacího engine

Hlavní přehrávací část byla navržena jako třída, pomocí které lze vytvořit objekt nabízející základní funkce přehrávání a komunikace se zvukovým zařízením (zvuková karta). Rozhraní bylo navrženo tak, aby umožnilo snadné a efektivní používání objektu s důrazem na sdružení skupiny elementárních vnitřních příkazů použité knihovny do logicky nazvaných metod, které obsluhují dané funkcionality.

6.2.1 Použitá knihovna

Díličí funkcionality jsou implementovány za pomoci knihovny Bass audio library. Jedná se o multiplatformní knihovnu nabízející velké množství individualizace. Na cílové platformě tohoto projektu, operačním systému Windows, je pro běh vyžadováno aplikační rozhraní DirectX 3 nebo vyšší, které je běžně používáno k ovládání většiny moderního hardware, zde konkrétně zvukových zařízení. Využívá se také DirectSound a DirectSound3D hardwarové akcelerace, je-li samozřejmě k dispozici. [18]

Jelikož je knihovna Bass napsána v C/C++, tedy unmanaged kódu, a ukázková aplikace bude implementována v prostředí .NET, typický zástupce managed kódu, tak je potřeba věnovat této problematice větší pozornost.

Garbage Collector

V případě managed kódu je zde Garbage Collector, který je součástí běhového prostředí. Spravuje všechny objekty a má za úkol automaticky řídit, které části paměti programu se již nepoužívají a případně je uvolňovat k dalšímu použití.

Garbage Collector v .NET Frameworku spravuje přidělování a uvolňování paměti v rámci aplikace. Pokaždé, když je použit nový operátor pro vytvoření objektu, tak běhové prostředí alokuje paměť pro objekt ze spravované haldy.

Do té doby, dokud je na haldě volná paměť k dispozici, běhové prostředí přiděluje prostor z této nepoužité oblasti. Nicméně, dostupná paměť není nekonečná a nakonec Garbage Collector musí provést zákrok s cílem uvolnit část paměti. Snaží se vybrat vždy nejvhodnější okamžik, při kterém zkontroluje dlouho nepoužívané objekty v haldě a provede potřebné operace k uvolnění těchto zdrojů. Časté je také přeskupování objektů na jiné místo v haldě za účelem zabránit segmentaci.

A zde se dostáváme k samotnému problému, kdy v případě unmanaged kódu knihovna Bass nemá o možných výše popsanych změnách v paměti žádné informace.

Typickým příkladem je předání (sdílení) textového řetězce z prostředí .NET knihovně Bass. Aby bylo možné tento proces realizovat, je nutné nejprve alokovat prostor v haldě.

1. Deklarujeme proměnnou, která poslouží jako handler spravující předávanou informaci. Konkrétně jde o datový typ *Structure* - *Runtime.InteropServices.GCHandle*.
2. Textovou informaci, která je uchována v proměnné datového typu *String* převedeme na pole bytů pomocí metody *System.Text.Encoding.ASCII.GetBytes*.
3. Vytvoříme handler na manager objekt pomocí *Runtime.InteropServices.GCHandle.Alloc*.
4. Nyní máme k dispozici bezpečný ukazatel na *String*, který je dostupný jako funkce našeho handleru, nazvaná *AddrOfPinnedObject*, a můžeme jej předat do unmanaged kódu.

Důležité je ale zachování handleru pro případ, že se na straně unmanaged kódu nevytváří kopie přijatých dat, ale čte se opakovaně z paměti na předané adrese.

Vytvoříme-li handler pouze jako lokální proměnnou v rámci metody, dojde po jejím skončení k uvolnění handleru a Garbage Collector již neví, že by měl toto paměťové místo chránit před přepsání a další manipulací. Nepředává-li se pouze jednorázová informace nebo není-li na straně unmanaged kódu vytvořena kopie dat pro vlastní použití, je třeba zaručit existenci handleru i po skončení metody. Ideální je globální deklarace nebo deklarace na úrovni nadřazeného objektu, který zaručí platnost na potřebnou dobu.

V praxi by se neošetření této situace projevovalo jako občasný nepředvídaný pád aplikace z důvodu manipulace s neplatnou částí paměti. Jak vyplývá z výše popsaného, tak k pádu by docházelo v okamžiku, kdy by Garbage Collector uvolnil paměť, která byla dříve předána unmanaged kódu a ten ji stále používal. Takové chyby se velmi obtížně hledají, nepřispívá tomu ani fakt, že k jejich projevení dochází pouze občasně a zcela náhodně právě po činnosti Garbage Collectoru.

6.2.2 Metody objektu

Konstruktor

Při vytvoření objektu dojde k načtení základní knihovny Bass, a také knihoven zajišťujících podporu dalších nestandardních hudebních formátů.

Provede se inicializace zvukového zařízení, kdy se jako výstupní zařízení nastaví výchozí zvuková karta podle systémové konfigurace. Jako výstupní vzorkovací frekvence je nastaveno 44100 Hz.

playURL

Metoda slouží pro spuštění přehrávání konkrétně zadané URL adresy streamu. V rámci těla metody proběhne nejprve uvolnění předchozího streamu, probíhalo-li doposud jiné přehrávání. Následně se vytvoří nový stream ze zadané URL adresy s požadavkem na zpracování od nultého bytu. Předá se také ukazatel na metodu obsluhující nahrávání, pro případ jeho spuštění v budoucnu. Ukazatel na stream je uložen do privátní proměnné objektu. Dojde k načtení metadat a vytvoří se trigger hlídající konec streamu. Proběhne spuštění připraveného streamu a je generována událost objektu, které signalizuje změnu streamu v rámci přehrávacího engine.

playStation

Podobně jako metoda *playURL* slouží k zahájení přehrávání, tak v případě *playStation* jde o přehrávání konkrétní rádiové stanice a jejího vybraného streamu (hudebního formátu a kvality - bitrate). Specifikace požadované stanice je dána parametrem *station*, který akceptuje ukazatel na objekt vytvořený podle třídy *cStation*. Konkrétní stream rádia je určen parametrem *streamID* nesoucím informaci o identifikátoru dostupného streamu v rámci předaného objektu rádia v parametru *station*.

Třída *cStation* byla navržena tak, aby jednoduše a v logické podobě nabídla všechny potřebné informace o rádiové stanici, včetně všech variant streamů.

Díky zvolenému návrhu lze tak v rámci metody *playStation* jednoduše získat samotnou URL adresu vybraného streamu následujícím způsobem: *station.streams(streamID).url*.

K nabídce streamů přistoupíme pomocí vlastnosti *streams*, která poskytuje kolekci všech variant pro naladění daného rádia. Pomocí unikátního identifikátoru streamu načteme konkrétní položku a vybereme údaj url. Dále se postupuje podobně, jakoby se přehrávala zadaná URL adresa, pouze s tím rozdílem, že je do proměnných v rámci objektu uložen ukazatel na objekt aktuální přehrávané stanice a také identifikátor streamu.

playRecord

Díky této metodě lze provést spuštění vybrané nahrávky přímo v rámci aplikace. Pro použití je nezbytné předání proměnné typu *sRecord* (struktura dostupná v třídě *cData*). Oproti ostatním dvěma metodám zaměřeným na přehrávání se tato liší tím, že neprovádí přehrávání vzdáleného streamu, ale souboru uloženého na lokálním disku počítače. Demonstruje tím tak další odlišné možnosti implementace funkce přehrávání pomocí hostitelské knihovny.

recStart

Provede aktivaci nahrávání, která spočívá v sestavení vhodného názvu fyzického souboru, povolení nahrávání a vyvolání události *recordingStart*, která o tomto procesu informuje okolí objektu. Poznačí se také čas spuštění procesu sloužící k výpočtu délky trvání nahrávky. Samotné nahrávání probíhá za pomoci privátní metody, jež dokáže přijímaná data ze stream serveru zpracovávat a v patřičném formátu ukládat do definovaného souboru. Ukazatel na tuto metodu byl předán funkci provádějící přehrávání při inicializaci streamu metodou *playURL*.

recStop

Probíhá-li nahrávání, provede jeho ukončení a zařazení souboru do nabídky nahrávek. Ukončení nahrávání vyvolá událost *recordingStop* nesoucí informaci o místě a názvu vytvořeného souboru. Tato událost je následně odchycena a GUI na ni reaguje přepnutím zobrazení do výpisu nahrávek, kde je tato nově vzniklá nahrávka vyznačena.

setProxy

Metoda sloužící k nastavení proxy připojení, je-li to nutné nebo žádoucí. Celkem umožňuje konfigurovat následující případy:

- Zakázat používání proxy připojení

- Použít výchozí systémové nastavení
- Použije výchozí proxy server, avšak individuální autorizaci (uživatel, heslo)
- Nastaví konkrétní proxy server, port a také autorizaci (uživatel, heslo)
- Nastaví proxy server s připojením bez autorizace

Metoda provede detekci některého z uvedených případů na základě zadaných vstupních parametrů. Dalším krokem je převedení hodnot v patřičném uspořádání na pole bytů, které je předáno obslužné knihovně. Je třeba zajistit uchování ukazatele na alokovaný datový prostor, proto je ukazatel deklarován jako *Static*.

stopStream

Zastaví probíhající přehrávání streamu. Probíhá-li nahrávání, je rovněž zastaveno. Při ukončení přehrávání je v rámci objektu vyvolána událost *streamEnd* s příznakem informujícím o tom, že se nejednalo o samovolný konec přehrávání (ukončení bylo iniciováno vnějším zásahem).

6.2.3 Vlastnosti objektu

isPlaying

Vlastnost datového typu *Boolean* sloužící pouze pro čtení. Poskytuje aktuální informaci o stavu přehrávání. *True* - pokud probíhá přehrávání, *False* – nepřehrává-li se.

isRecording

Podobně jako u *isPlaying* se jedná o vlastnost datového typu *Boolean* sloužící pouze pro čtení. Vypovídá o aktuálním stavu nahrávání. Probíhá-li nahrávání, vrací *True*, v opačném případě *False*.

position

Informuje o pozici v přehrávání, době od spuštění přehrávání. V rámci metody implementující tuto vlastnost je nejprve zjištěno počet doposud zpracovaných bytů. Tato hodnota je uložena do proměnné datového typu *Long* a dále na základě informací

o přehrávaném streamu je převedena na sekundy. Vlastnost *position* je určena pouze pro čtení a vrací desetinné číslo. Není-li možné pozici zjistit, vrací hodnotu 0.

positionRec

Poskytuje informaci o délce nahrávání, době od startu nahrávání. Nenahrává-li se, je návratová hodnota rovna nule. Probíhá-li nahrávání, je doba reprezentována jako desetinné číslo v sekundách.

url

Poskytuje informaci o URL adrese aktuálně přehrávaného streamu. URL obsahuje doménové jméno stream serveru (případně zápis pomocí IP adresy), jeho port (je-li stream distribuován na portu 80, tak je tato hodnota vynechána) a případně také tzv. přípojný bod (běžné zejména u Icecast serverů).

Hodnota této vlastnosti nemusí nutně obsahovat pouze přímou adresu streamu, ale může směřovat na playlist, a až ten obsahuje informace o konkrétních URL adresách reálných streamů.

volume

Vlastnost umožňující jednak zjištění dříve nastavené hlasitosti, tak také její přímé nastavení. Hodnota je celočíselné číslo v rozmezí 0 až 100, kdy 0 reprezentuje nejnižší možnou úroveň – ticho.

station

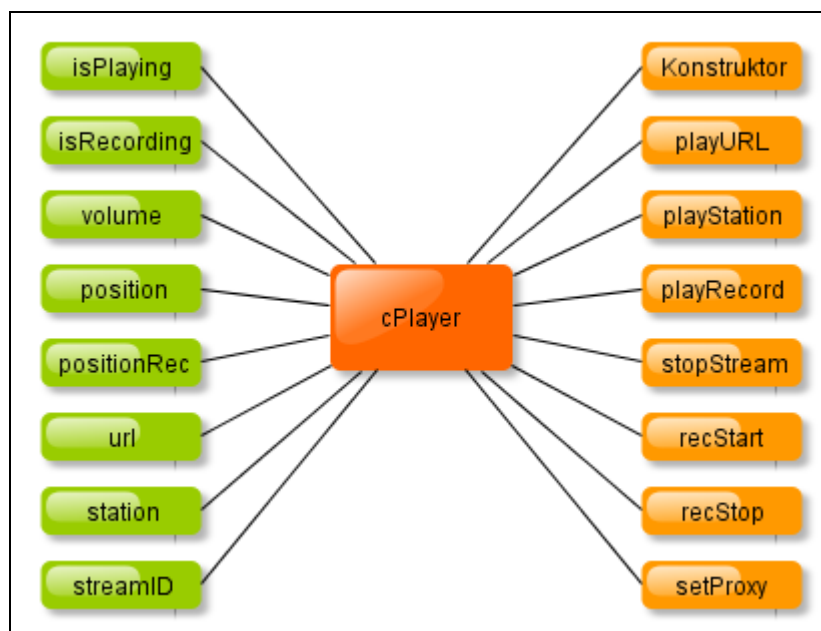
Ukazuje na objekt aktuálně přehrávané rádiové stanice, pokud naladění proběhlo prostřednictvím metody *playStation*, v opačném případě je návratová hodnota *Nothing* (například pokud bylo přehrávání iniciováno metodou *playURL*). Vlastnost tak poskytuje jednoduchý přístup ke všem dostupným informacím o rádiové stanici. Jednoduše je tak možné programově nechat zobrazit stanici přímo v uživatelském prostředí.

streamID

Návratová hodnota konkretizuje specifický stream v rámci přehrávané rádiové stanice, jedná se o unikátní identifikátor datového typu *String*. Unikátnost je zaručena v rámci celé databáze. Neprobíhá-li přehrávání streamu se známým identifikátorem, je návratová hodnota rovna prázdnému řetězci.

6.2.4 Grafické znázornění veřejných metod a vlastností

Souhrnné znázornění dostupných vlastností a metod třídy *cPlayer* graficky ukazuje základní možnosti a funkcionality, které lze v rámci objektu využít a nimiž je možné přehrávání ovlivňovat. Zelená barva odlišuje vlastnosti, oranžová pak metody.



Obrázek 10 - Grafické znázornění veřejných metod a vlastností [vlastní zpracování]

6.3 Datový objekt

Datový objekt byl navržen a implementován za účelem správy veškerých datových zdrojů v rámci aplikace. Poskytuje jak metody pro kompletní komunikaci se vzdáleným API rozhraním nabízejícím centrální databázi rádií, tak také obsluhu metod pro načtení nebo naopak obnovení uživatelského nastavení, které je uchováváno lokálně.

Za účelem načtení informací o nabídce rádiových stanic je dostupná metoda *loadServerData*. Dokáže data načíst ze serverové databáze přes API rozhraní nebo, pro

případ nedostupnosti tohoto zdroje, využijte lokálně uložený soubor obsahující demonstrační data. To se hodí v situaci, kdy není dostupné internetové připojení, ale chceme demonstrovat možnosti aplikace. Metoda *loadServerData* provede načtení potřebných hodnot do kolekce typu klíč-hodnota. Konkrétně u seznamu rádiových stanic dojde k sestavení objektů podle třídy *cStation* a uložení ukazatelů na tyto objekty do slovníkové kolekce pod globálním identifikátor stanice. Pro zjednodušení přístupu k objektům na základě identifikátoru streamu rádia dojde také k sestavení indexové tabulky, kde je vždy klíčem unikátní identifikátor streamu a hodnota prvku ukazuje na objekt rádiové stanice. Obě uvedené definice pak umožnily vzniknutí patřičných vlastností v rámci datového objektu. Každá z vlastností poskytuje na výstupu to samé, liší se však v přístupu ke vstupním parametrům, na základě kterých má dojít k poskytnutí žádaných dat. Vlastnost *getStation* poskytuje objekt žádané rádiové stanice na základě zadaného identifikátoru *stationID*, přičemž *getStationByStream* pracuje s indexovou tabulkou a vybírá stanici podle specifikovaného identifikátoru jejího streamu.

Datový objekt poskytuje také možnosti dynamické definice struktury základního menu. Nabídku, nacházející se v uživatelském prostředí v levé části, je tak možné libovolně předdefinovat jediným XML souborem udávajícím pořadí a skladbu položek v menu. Struktura XML byla navržena tak, aby se menu dalo opravdu libovolně rozšiřovat. GUI tak může při vykreslování nabídek reagovat na kliknutí definovaných položek, nebo mohou být nové položky získány v inicializačním rozhraní a přidány přímo do stromové struktury menu.

Také textové popisky nejsou přiřazeny staticky, ale prostřednictvím XML struktury, kde název elementu vystupuje jako identifikátor popisku, hodnota elementu nese textovou informaci.

Jak dynamická definice struktury menu, tak oddělení jazykových specifikací, ukazují možnosti připravenosti aplikace na vícejazyčnou podporu a provádění změn bez zásahu do zdrojového kódu. Obě struktury mohou být v budoucnu například definovány přímo v inicializačním rozhraní, čímž se docílí jednoduché správy těchto položek z centrálního místa. Nehledě na to, že různé jazykové mutace by mohly být editovány online přímo komunitou uživatelů.

Mezi další důležité možnosti datového objektu patří například schopnost přidávat stanice, respektive konkrétní streamy, k oblíbeným položkám, parsování názvů nahrávek, uchovávání nastavení proxy serveru, ukládání a obnova nastavení aplikace, zjednodušený přístup k URL adresám streamů, přidávání a editování nových rádiových stanic včetně komunikace s editačním API rozhraním, a jiné.

6.4 Uživatelské rozhraní

Grafické znázornění uživatelského rozhraní zajišťuje objekt *render*, který je tvořen podle třídy *cRender*. Ihned po vytvoření objektu se připraví grafické okno, k jehož aktivaci dojde po zavolání metody *start* objektu *render*.

Samotný objekt plní úlohu koordinátora mezi jednotlivými objekty, kde zajišťuje korektní reakce na dílčí události a podmínky, a tím je docíleno správného zobrazení ovládacích prvků pro specifické stavy aplikace.

Důležitá je zejména spolupráce s objektem *data*, který nabízí potřebné podklady pro vykreslení uživatelsky dostupných nabídek v grafickém okně.

6.4.1 Přehrávání

Přehrávání je základní funkcí celé aplikace. Libovolnou rádiovou stanicí je možno naladit několika způsoby. Jedním z nich je například vyhledání stanice, zvolení preferované varianty streamu a kliknutí na *Naladit*. Po navázání spojení se vzdáleným stream serverem se v levé části, pod hlavní nabídkou, zobrazí informační obsah s názvem aktuálně přehrávané stanice, čas přehrávání a několik tlačítek. Nechybí tlačítko pro zastavení přehrávání, zobrazení profilu rádia v rámci nabídky stanic nebo funkce nahrávání.

6.4.2 Nahrávání

Nahrávání přináší možnost, pomocí které lze pohodlně nahrávat rádiové vysílání bez zdoluhavého a komplikovaného nastavování čehokoliv. Jednoduše vyhledáte rádiovou stanicí, zvolíte stream a kliknete na *Naladit a nahrávat*. Alternativní možností je zahájení nahrávání již přehrávané stanice. Stačí kliknout na tlačítko s červeným symbolem v informační oblasti přehrávání. Po ukončení nahrávání je nahrávka zařazena do seznamu

nahrávek a lze ji kdykoliv přehrát přímo v aplikaci. V aktuální verzi aplikace je dostupné nahrávání formátů Mp3 a AAC+.

6.4.3 Oblíbené

Možnost přidání konkrétního streamu mezi oblíbené položky výrazně zjednodušuje používání aplikace a urychluje tak spuštění často používaných rádiových stanic, respektive jejich preferovaných formátů a přenosových kvalit.

V okně aplikace je přidání streamu mezi oblíbené velmi jednoduché. Stačí najít oblíbenou stanici, vybrat požadovaný formát a kvalitu a na položku kliknout pravým tlačítkem. Oblíbené streamy jsou při výpisu zvýrazněny tučně. Jednoduchý přístup ke všem takto označeným položkám je k dispozici po kliknutí na *Oblíbené* v levém menu. Kliknutím pravého tlačítkem v tomto výpisu se zobrazí profil příslušné rádiové stanice.

6.4.4 Nahrávky

Seznam všech pořízených nahrávek je k dispozici po kliknutí na záložku *Nahrávky* v levém menu. Po rozkliknutí je možné výsledky filtrovat podle data pořízení, a to ve třech kategoriích: *Dnes*, *Aktuální měsíc*, *Aktuální rok*. V samotném výpisu je u každé nahrávky vidět datum pořízení, čas, zdrojové rádio, hudební formát, kvalita a také velikost fyzického souboru na disku.

6.4.5 TimeMachine

Funkce *TimeMachine* přináší zcela novou možnost v poslechu vybraných internetových rádií. Díky vlastní serverové implementaci tzv. cloud-recorderu nabízí unikátní možnost poslechu rádia z minulosti vysílání. Jednoduše je třeba zvolit stanici, datum, čas a kliknout na naladit. Okamžitě dojde ke spuštění vysílání právě od zvoleného data a času, a to s přesností na sekundy. Tato funkce je v této práci popsána pouze z pohledu klientské aplikace, nikoli z hlediska serverové implementace.

Odezva služby na požadavek spuštění zvoleného rádia od libovolného data a času je v řádu milisekund. Okamžitě je tak možné naladit vysílání, které proběhlo třeba i před několika měsíci. Tato funkce tak přináší jedinečnou možnost poslechnout si vysílání ze záznamu i bez

toho, aniž bychom měli v požadovaný čas spuštěné nahrávání nebo dokonce zapnutý počítač.

6.4.6 Nastavení

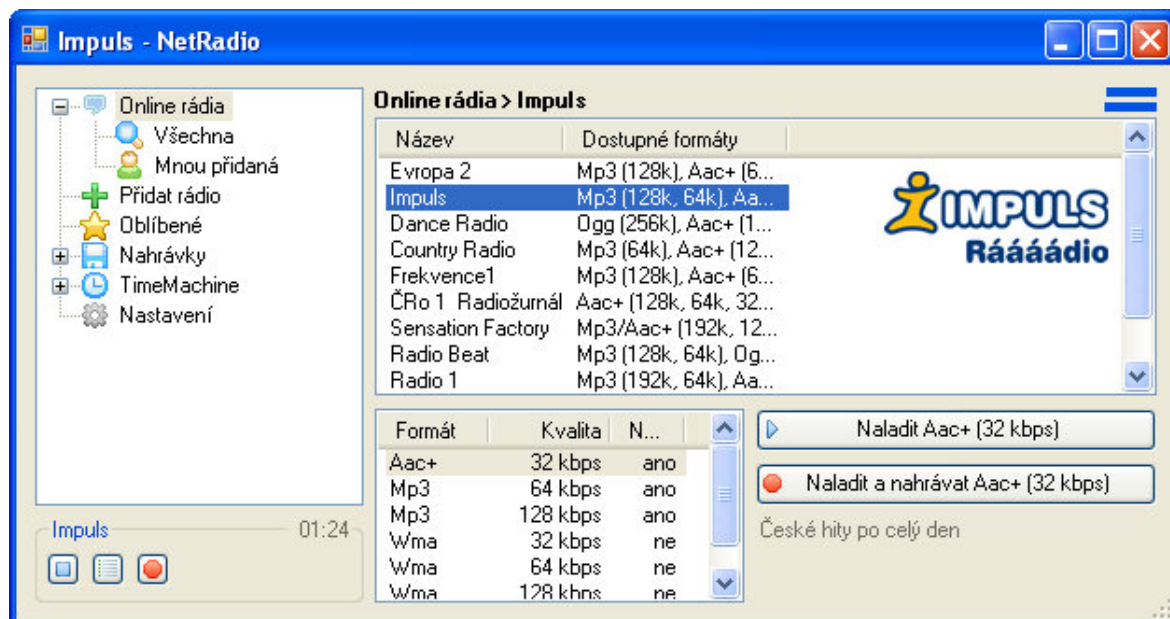
Sekce nastavení zpřístupňuje jednoduchou možnost konfigurace způsobu navázání spojení se stream serverem. Lze provést nastavení proxy serveru, a to v několika variantách, od využití výchozího systémového nastavení až po definici autorizačních údajů jména a hesla. Tato možnost je nezbytná například v sítích, kde není k dispozici neomezený přístup k internetu a spojení je realizováno přes centrální proxy server.

6.4.7 Ostatní funkce

Mezi další zajímavé funkce v rámci okna ukázkové aplikace patří například drobečková navigace, zobrazování loga rádia nebo krátký a výstižný popis, případně slogan rádia. Za povšimnutí stojí také reálné znázorňování úrovně signálu levého a pravého reproduktoru v průběhu přehrávání.

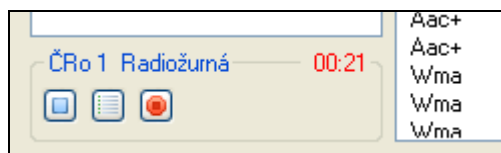
6.5 Náhled ukázkové aplikace

Na obrázku 11 je znázorněno procházení seznamu dostupných rádiových stanic. Vidět je také seznam variant streamů, řazených podle formátu a datového toku. Nechybí nabídka pro naladění nebo spuštění nahrávání. V levém spodním rohu je vidět informační oblast přehrávání s ovládacími tlačítky a době poslechu.



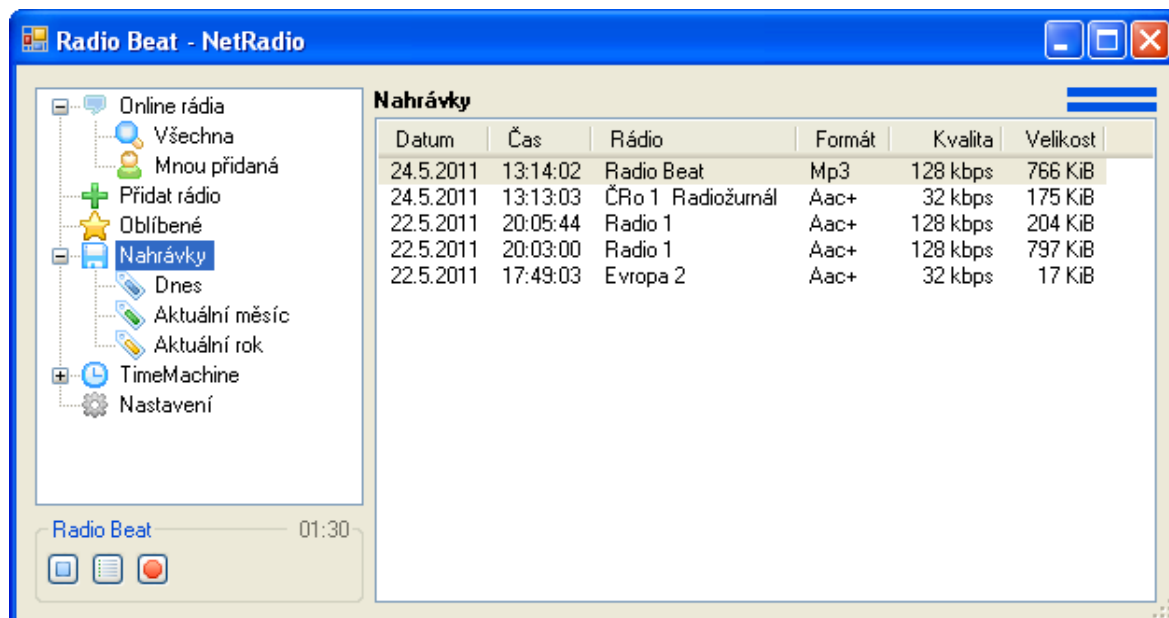
Obrázek 11 - Nabídka rádiových stanic, přehrávání [vlastní zpracování]

Spustíme-li nahrávání, jak je vidět na obrázku 12, údaj o době poslechu se přepne na informaci o délce trvání nahrávání. Graficky je hodnota zvýrazněna červenou barvou.



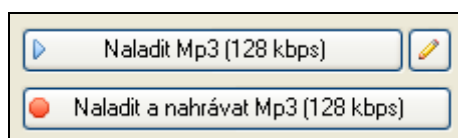
Obrázek 12 – Nahrávání [vlastní zpracování]

Všechny pořízené nahrávky jsou dostupné v záložce *Nahrávky*, obrázek 13. Fyzicky jsou soubory uloženy v adresáři *records* v místě, kde je aplikace uložena. Dvojklikem na vybranou nahrávku se spustí její přehrávání.



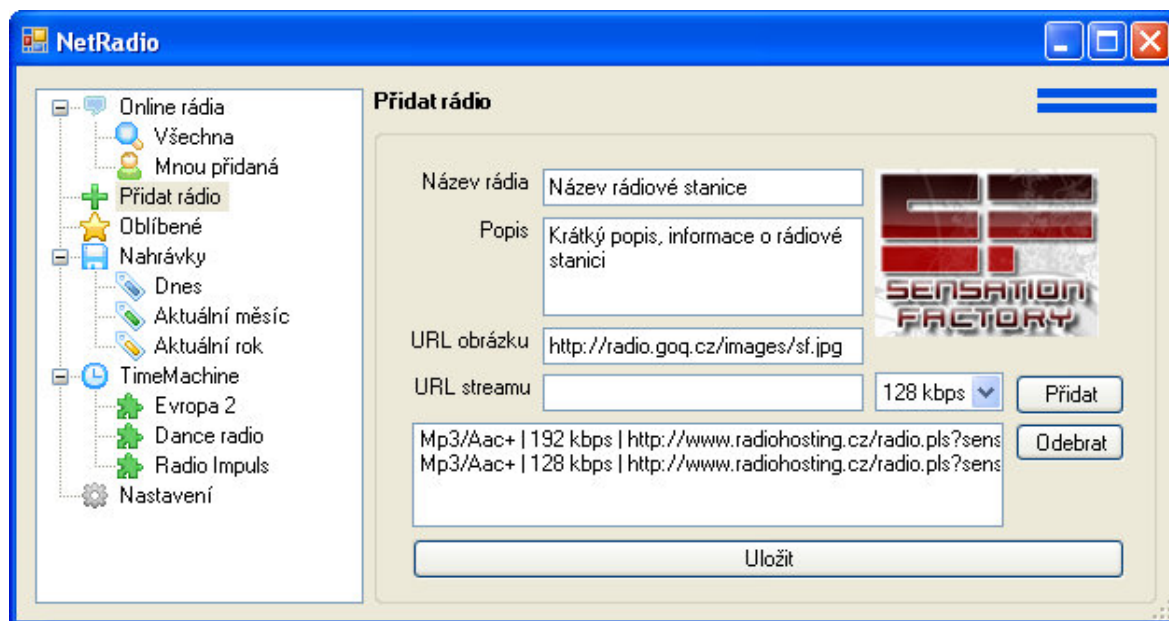
Obrázek 13 - Procházení nahrávek [vlastní zpracování]

Vybereme-li ze seznamu rádiovou stanici, u níž máme oprávnění měnit její údaje, objeví se v nabídce editační tlačítko, obrázek 14. Možnost je dostupná například u všech rádiových stanic, které si do aplikace sami přidáme.



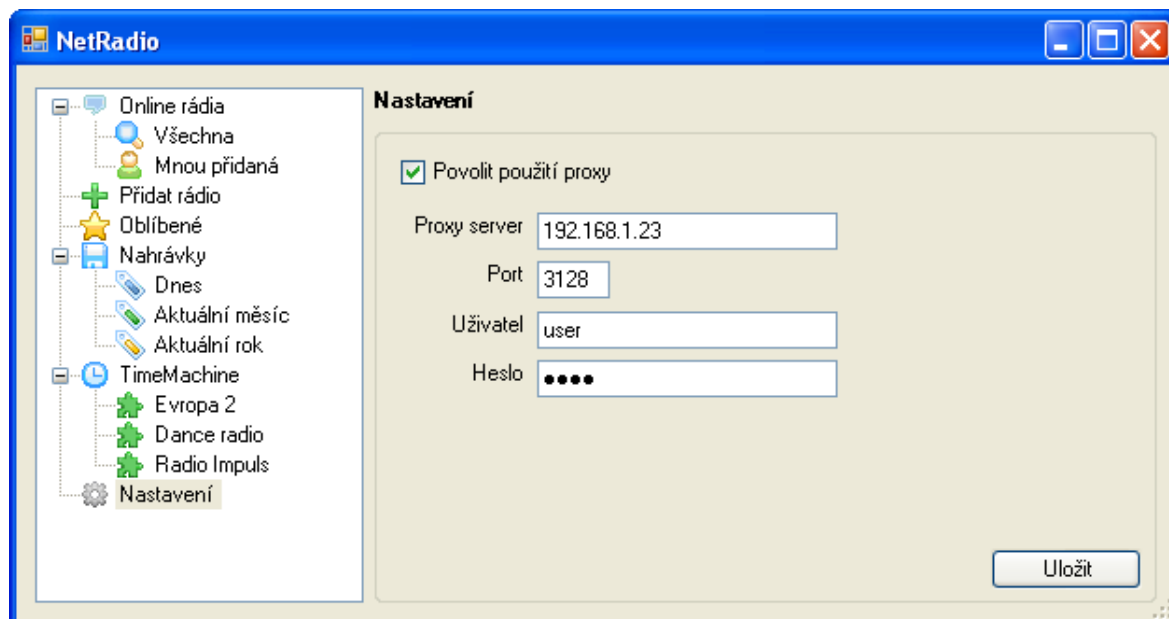
Obrázek 14 - Editací tlačítko [vlastní zpracování]

Sekce pro editaci a přidávání nových rádií je společná, obrázek 15. Mimo běžné vložení informací typu název rádia a popis, nabízí také políčko URL obrázek, do kterého lze zadat adresu směřující na logo rádia v rámci internetu. Ihned po vložení se v pravé části zobrazí náhled přidávaného loga. Důležitou vlastností je přidávání URL adres streamů. Po zadání adresy a kliknutí na tlačítko *Přidat* dojde ke zkušebnímu naladění streamu a ověření jeho funkčnosti. Provádí se také detekce formátu podle přijatých metadat ze stream serveru.



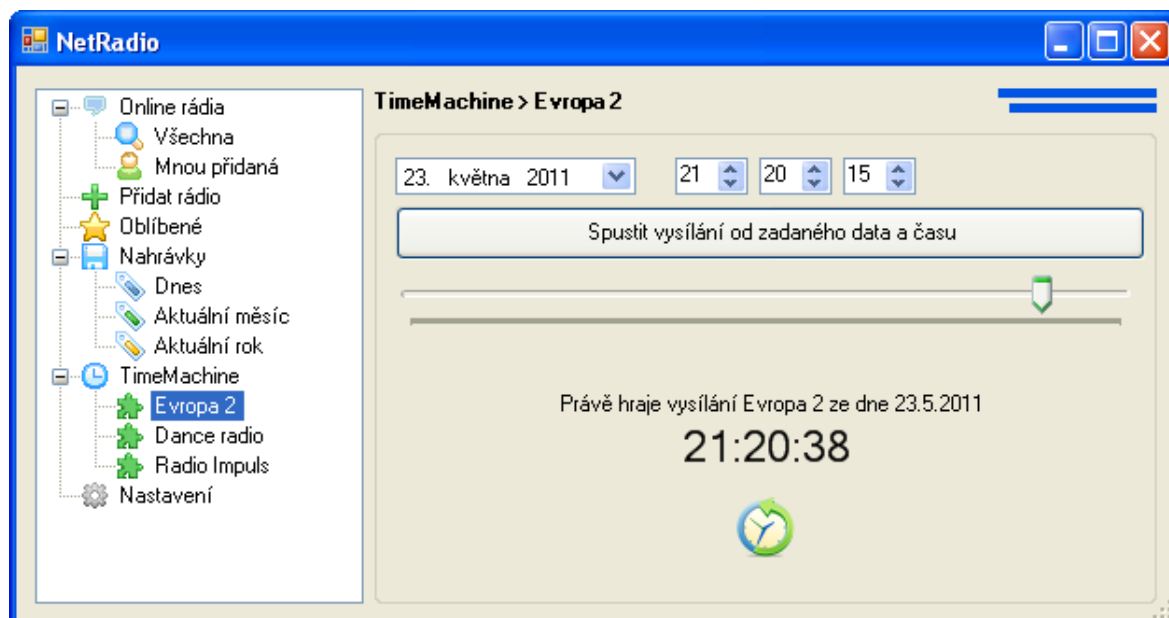
Obrázek 15 - Přidání nebo editace záznamu [vlastní zpracování]

Konfigurace proxy připojení je znázorněna na obrázku 16.



Obrázek 16 – Nastavení [vlastní zpracování]

Funkce *TimeMachine*, která je v rámci ukázkové aplikace nejunikátnější, je vidět na obrázku 17. Ačkoliv jde o zajímavou funkcionalitu, tak její používání je velmi intuitivní a jednoduché. Stačí vybrat některou z dostupných rádiových stanic, vybrat datum, přesný čas spuštění rádia a kliknout na *Spustit vysílání od zadaného data a času*. Okamžitě proběhne naladění vysílání z minulosti.



Obrázek 17 - Funkce TimeMachine [vlastní zpracování]

7 DEMONSTRACE ONLINE ŘEŠENÍ PŘEHRÁVAČE A FUNKCE TIMEMACHINE

Za účelem demonstrace přehrávání internetových rádií přímo v online prostředí internetu byl v rámci této práce vytvořen speciální ukázkový přehrávač. Pro zajištění všech jeho funkcí bylo použito množství technologií, které v důsledku nabízejí přehrávač, jež není nutné instalovat, ale běží v okně internetového prohlížeče.

Nenabízí pouze běžné přehrávání streamu rádiové stanice, ale podporuje také již zmíněnou funkci *TimeMachine*, tzn. přehrávání z minulosti vysílání.

7.1 Časová osa

Časová osa znázorňuje časový úsek jednoho dne. Ovládací prvek je rozložen přes celou šířku přehrávače. V případě potřeby je možné rozsah zmenšit. Docílí se tím větší přesnosti a časová osa tak zobrazuje pouze specifikovanou část dne, kdy její přesný rozsah je vždy uveden na levém a pravém konci číselnou hodnotou. Toto omezení lze provést ovládacím prvkem se dvěma posuvníky vymezujícími žádaný úsek dne.

7.2 Přehrávání

Je-li zvýrazněná ikona „mikrofonu“ v informační oblasti, probíhá přehrávání živého vysílání rádiové stanice. Informační oblast se plynule pohybuje nad osou v závislosti na aktuálním čase. Časový údaj je uveden také přímo v informační oblasti. Přetažením posuvníku na libovolnou část osy se přehrávač automaticky přepne do módu přehrávání z historie. Dojde k naladění vysílání od žádaného času a ikona v informační oblasti se změní na „hodiny“. Průběžně zobrazovaný čas a pozice nad osou se samozřejmě vztahují k přehrávání v minulosti, takže vidíme reálný čas pro probíhající vysílání.

7.3 Orientace v historii vysílání

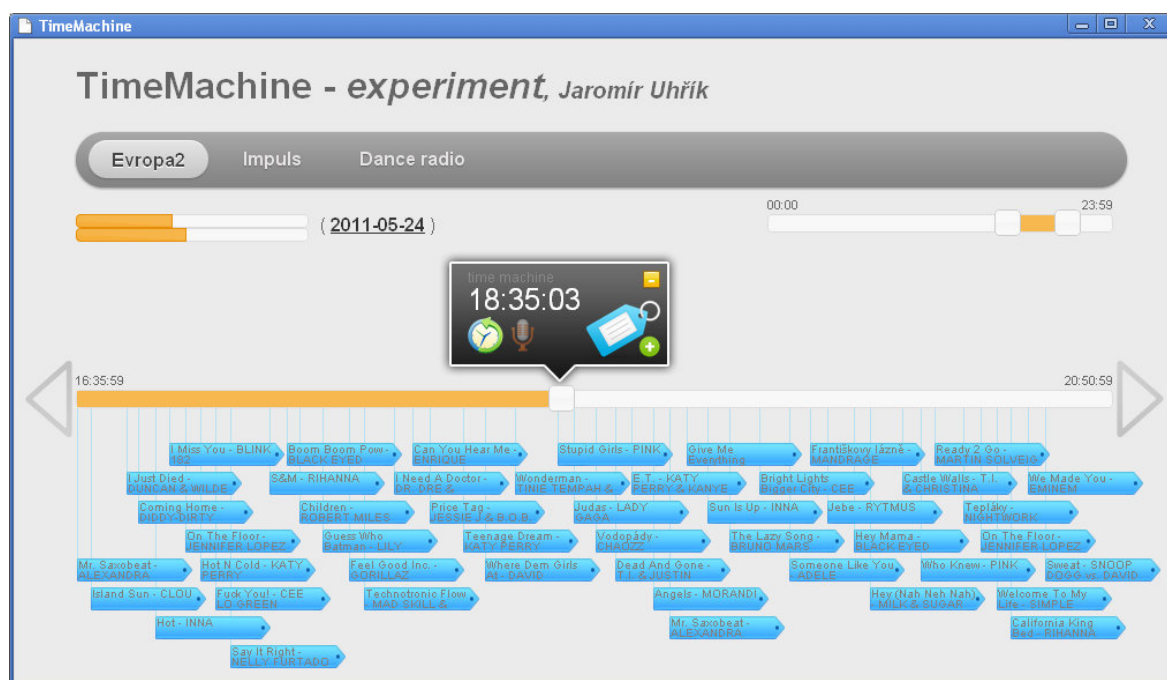
Přehrávač je rozšířen o podporu štítků. Ty jsou rozloženy po časové ose a korespondují s programem vysílání v historii. Štítky nesou časové razítko a textový popis. Nabízejí zajímavou možnost orientace v zobrazené části časové osy. Jejich umístění a popis vždy odpovídají názvu skladby, která v daný čas hrála. Kliknutím na vybraný štítek se spustí vysílání z historie a z přehrávače zní vybraná skladba. Aby nedocházelo k překrývání štítků,

provádí se jejich rozřazování do pomyslných vrstev. Štítky se načítají ze serveru a zobrazují se průběžně i po spuštění přehrávače.

7.4 Použité technologie

K vykreslování grafického prostředí online přehrávače je použito jazyka HTML a kaskádových stylů (CSS). Dynamické prvky jsou překreslovány za pomoci JavaScriptu a knihovny jQuery.

Průběžné načítání štítků je realizováno pomocí technologie Ajax, která data čte ze vzdáleného serveru. Nosným formátem pro předávaná data je JSON. Na serveru běží operační systém Debian s nainstalovaným web serverem Apache a podporou skriptovacího jazyka PHP. Data jsou skladována v databázovém systému MySQL a k jejich výběru je využíváno dotazovacího jazyka SQL.



Obrázek 18 - Online přehrávač, funkce TimeMachine [vlastní zpracování]

ZÁVĚR

Před realizací této bakalářské práce jsem si vytyčil několik cílů, které jsem se snažil realizovat. V teoretické části se zabývám možnostmi přístupu k přehrávání a záznamu online médií. Uvedeny jsou obecné principy fungování těchto procesů, a také možné přístupy samotných uživatelů vzhledem k jejich preferencím. Důležitou částí je charakteristika technologií, které jsou dále použity při realizaci reálně fungující ukázkové aplikace.

Její vnitřní návrh, implementace a nabízené možnosti popisuje praktická část. Tato část je doplněna také několika grafickými náhledy uživatelského rozhraní vytvořené desktopové aplikace. Zajímavou funkcionalitou je možnost nazvaná TimeMachine, která v bakalářské práci vystupuje jako vzdálená služba. Ukázkový přehrávač je v roli klienta, jež ji využívá. Tato funkce demonstruje další možný vývoj aplikace v souladu s aktuálními trendy v oblasti IT. Přináší zcela nový přístup k nahrávání internetových rádií v rámci tzv. cloud-recorderu. V závěru je uvedena demonstrace online řešení přehrávače živého vysílání internetových rádií. Dominantní je zejména návrh uživatelského prostředí v podobě časové osy s rozmístěnými šítky pro intuitivní ovládání funkce TimeMachine.

Výstupem této bakalářské práce je dokument seznamující čtenáře s danou problematikou a technologiemi, jejichž praktickou ukázkou využití je reálně fungující aplikace, kterou je možné používat v praxi. S ohledem na některé nové funkce má zajímavou konkurenční výhodu, která může být díky svému potenciálu dalším vývojem upevněna.

ZÁVĚR V ANGLIČTINĚ

Before implementing this thesis I have set itself some goals that I have tried to implement. The theoretical part deals with the possibilities of access to play and record online media. Included are general principles of these processes, and possible approaches to the users themselves, given their preferences. The important part is the characterization of technologies that are further used to actually implement a working sample application.

Its internal design, implementation and offered possibilities are described in practical part. This section is supplemented by several previews of graphical user interfaces of desktop application. Interesting functionality is option called TimeMachine that appears as remote service in this thesis. Sample player is a client that uses it. This feature demonstrates another potential application development in accordance with current trends in IT. It brings a whole new approach to record Internet radio stations in the cloud-recorder. In the end there is a demonstration of online solution of live streaming internet radio player. The dominant is a design of the user interface in the form of a timeline laid out with tags for intuitive control TimeMachine.

The outcome of this thesis document is to familiarize readers with the issues and technologies, the practical example of application is realistically functioning application that can be used in practice. With regard to some interesting new features application has competitive advantage that can be due to its potential strengthened by further development.

SEZNAM POUŽITÉ LITERATURY

- [1] Streaming. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 17. 8. 2007, last modified on 15. 2. 2011 [cit. 2011-05-23]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Streaming>>.
- [2] AUSTERBERRY, David. The Technology of Video and Audio Streaming : US. 2nd Edition. [s.l.] : Focal Press Title, 2005. 360 s. ISBN 978-0-240-80580-1.
- [3] Progressive download. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 10 February 2009, last modified on 5 March 2011 [cit. 2011-05-23]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Progressive_download>.
- [4] ČADA, Ondřej. MujMac [online]. 2010 [cit. 2011-03-23]. Streaming videa - 2. Protokoly RTSP a RTP. Dostupné z WWW: <<http://www.mujmac.cz/art/multimedia/MEZIPROUDY2.html>>.
- [5] Ajax (programming). In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 15 March 2005, last modified on 15 April 2011 [cit. 2011-05-23]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))>.
- [6] CYROŇ, Miroslav. CSS - kaskádové styly. Praha : Grada Publishing a.s., 2006. 340 s. ISBN 8024714205.
- [7] JavaScript. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 12. 8. 2004, last modified on 29. 3. 2011 [cit. 2011-05-23]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/JavaScript>>.
- [8] JQuery. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 10. 7. 2009, last modified on 22. 3. 2011 [cit. 2011-05-23]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/JQuery>>.
- [9] Introducing JSON [online]. 2002 [cit. 2011-04-03]. Dostupné z WWW: <<http://www.json.org>>.
- [10] JEFFREY, Richter. .NET Framework : programování aplikací. [s.l.] : GRADA Publishing, a.s., 2003. 552 s. ISBN 978-80-247-0450-1.

- [11] LENZ, Gunther; MOELLER, Thomas. .NET - A Complete Development Cycle. [s.l.] : Addison-Wesley Professional, 2004. 592 s. ISBN 0-321-16882-8.
- [12] Microsoft .NET Framework [online]. 2011 [cit. 2011-04-03]. Dostupné z WWW: <<http://www.microsoft.com/net/>>.
- [13] .NET. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2. 8. 2005, last modified on 30. 12. 2010 [cit. 2011-05-23]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/.NET>>.
- [14] BRÁZA, Jiří. PHP 5. Praha : Grada Publishing a.s., 2005. 244 s. ISBN 9788024711461.
- [15] W3Techs - World Wide Web Technology Surveys [online]. 2011 [cit. 2011-04-09]. Usage Statistics and Market Share of Server-side Programming Languages for Websites. Dostupné z WWW: <http://w3techs.com/technologies/overview/programming_language/all>.
- [16] SQL. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 28 June 2001, last modified on 21 April 2011 [cit. 2011-05-24]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/SQL>>.
- [17] XML. In Wikipedia : the free encyclopedia [online]. St. Petersburg (Florida) : Wikipedia Foundation, 27 September 2001, last modified on 19 April 2011 [cit. 2011-05-24]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/XML>>.
- [18] un4seen developments. BASS audio library [online]. 2010 [cit. 2011-02-03]. Audio library. Dostupné z WWW: <<http://www.un4seen.com/>>.
- [19] Netcraft - Internet Research [online]. 2011 [cit. 2011-04-03]. January 2011 Web Server Survey. Dostupné z WWW: <<http://news.netcraft.com/archives/2011/01/12/january-2011-web-server-survey-4.html>>.
- [20] SHOUTcast Radio [online]. 2011 [cit. 2011-04-03]. Dostupné z WWW: <<http://www.shoutcast.com/>>.
- [21] Zákony online [online]. 2011 [cit. 2011-04-29]. Autorský zákon. Dostupné z WWW: <<http://zakony-online.cz/?s4&q4=all>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AAC	Advanced Audio Coding
ADSL	Asymmetric Digital Subscriber Line
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASP	Active Server Pages
CLR	Common Language Runtime
CSS	Cascading Style Sheets
DOM	Document Object Model
DRM	Digital Rights Management
FCL	Framework Class Library
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
LAMP	Linux Apache MySQL PHP/Perl/Python
Mp3	Motion Picture experts group - layer 3 (MPeg layer 3)
PHP	Hypertext Preprocessor, původně Personal Home Page
PNG	Portable Network Graphics
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
UCS	Universal Character Set
URL	Uniform Resource Locator
UTF	UCS Transformation Format
WMA	Windows Media Audio
WMP	Windows Media Player
XML	Extensible Markup Language

SEZNAM OBRÁZKŮ

Obrázek 1 - Zjednodušené schéma vysílání internetového rádia [vlastní zpracování]	12
Obrázek 2 - Definice kaskádových stylů [vlastní zpracování]	21
Obrázek 3 - JSON - kolekce páru název-hodnota [vlastní zpracování]	24
Obrázek 4 - JSON - tříděný seznam [vlastní zpracování]	24
Obrázek 5 – API rozhraní - chybné přihlášení [vlastní zpracování]	31
Obrázek 6 - API rozhraní - korektní odpověď [vlastní zpracování]	32
Obrázek 7 - API rozhraní – editace [vlastní zpracování]	36
Obrázek 8 - API rozhraní - chyba oprávnění [vlastní zpracování]	36
Obrázek 9 - Základní návrh aplikace [vlastní zpracování]	38
Obrázek 10 - Grafické znázornění veřejných metod a vlastností [vlastní zpracování]	45
Obrázek 11 - Nabídka rádiových stanic, přehrávání [vlastní zpracování]	50
Obrázek 12 – Nahrávání [vlastní zpracování]	50
Obrázek 13 - Procházení nahrávek [vlastní zpracování]	51
Obrázek 14 - Editační tlačítko [vlastní zpracování]	51
Obrázek 15 - Přidání nebo editace záznamu [vlastní zpracování]	52
Obrázek 16 – Nastavení [vlastní zpracování]	53
Obrázek 17 - Funkce TimeMachine [vlastní zpracování]	53
Obrázek 18 - Online přehrávač, funkce TimeMachine [vlastní zpracování]	55

SEZNAM TABULEK

<i>Tabulka 1 - Oblíbenost stream serveru SHOUTcast [20]</i>	13
---	----

SEZNAM PŘÍLOH

P I Zdrojový kód ukázkové aplikace na CD

P II Zdrojový kód demonstrace online řešení přehrávače a funkce TimeMachine na CD

