VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMU

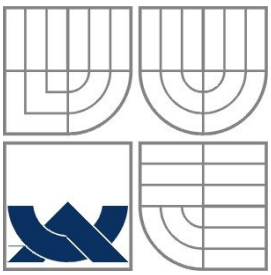
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

OSOBNÍ MULTIMEDIÁLNÍ KNIHOVNA

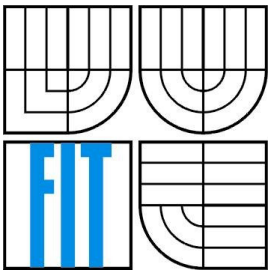
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JAN KOUKAL
AUTHOR

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

OSOBNÍ MULTIMEDIÁLNÍ KNIHOVNA

PERSONAL MULTIMEDIA LIBRARY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN KOUKAL

VEDOUCÍ PRÁCE
SUPERVISOR

ING. DAVID KUBÁT

BRNO 2011

Abstrakt

Předmětem této bakalářské práce bylo seznámit se se specifikacemi multimediálních souborů a jejich atributů a také prostudovat možnosti, kterých lze využít pro jejich správu. Dále bylo nutné zaměřit se na již hotová řešení podobného typu, z nichž bylo možné se inspirovat a čerpat zkušenosti.
Cílem práce bylo z nabytých zkušeností a poznatků navrhnout funkční multiplatformní aplikaci, která bude umožňovat efektivní správu a prohlížení těchto souborů a poskytovat tak uživateli dostatek relevantních informací. V další části pak byla aplikace implementována a testována a představuje plně funkční výsledek celé práce.

Abstract

The objective of this bachelor's thesis was to get acquainted with specifications of multimedia files and their attributes and study options that can be used for their management too. Further was necessary to focus on already done solutions of similar type from which it was able to inspire and draw on experience.
The goal of this work was to design a functional multiplaform application on past experience that will allow effecitve management and browsing of these files and provide user enough of relevant informations. In the next step was application implemented and tested so it represents a fully functional result of the thesis.

Klíčová slova

Multimédia, knihovna, soubory, grafické uživatelské rozhraní, lokální databáze, objektově orientovaný model, audio, video, obrázky, SQLite, MediaInfo, Java, XML.

Keywords

Multimedia, library, files, graphic user interface, local database, object-oriented model, audio, video, images, SQLite, MediaInfo, Java, XML.

Citace

Jan Koukal: Osobní multimediální knihovna, bakalářská práce, Brno, FIT VUT v Brně, 2011

OSOBNÍ MULTIMEDIÁLNÍ KNIHOVNA

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Davida Kubáta. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jan Koukal
18.5.2011

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu Ing. Davidu Kubátovi za ochotu, rady, pomoc při řešení práce a vstřícné jednání. Pod jeho vedením se podařilo vytvořit funkční aplikaci, která splňuje všechny požadavky, za což jsem mu vděčný.

© Jan Koukal, 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 Cíl práce a její metody....4
2.1 Cíl práce....4
2.2 Metody práce....4
3 Multimédia....6
3.1 Rozdělení multimediálních dat....6
3.2 Audio....6
3.3 Video....8
3.4 Obrázky....9
4 Návrh aplikace....10
4.1 Databázový model....10
4.2 Struktura XML dokumentů....11
4.2.1 Ukládání nastavení....11
4.2.2 Export/Import multimédií....11
4.3 Grafické uživatelské rozhraní....12
4.3.1 Hlavní nabídka....13
4.3.2 Tlačítka multimédií....13
4.3.3 Adresářový strom....14
4.3.4 Tabulka multimédií....14
4.3.5 Tlačítka pro práci s multimédii....15
4.3.6 Stavový řádek....15
4.3.7 Okno nastavení....15
4.3.8 Okno vyhledávání....16
4.3.9 Okno pro úpravy....16
4.3.10 Nápověda....16
4.3.11 Tray ikona....17
4.4 Objektový model....17
4.4.1 Třídy diagramu....17
5 Použité nástroje....18
5.1 Programovací jazyk – Java....18
5.1.1 Vývojové prostředí....18
5.2 Databáze....19
5.2.1 Výběr databáze....19

5.2.2 SQLite....19
5.3 MediaInfo....20
6 Implementace....21
6.1 Struktura aplikace....21
6.1.1 Systém nastavení....22
6.1.2 Práce s databází....22
6.1.3 Multimediální soubory....23
6.1.4 Okna aplikace....23
6.2 Propojení s dalším softwarem....24
6.2.1 Načítání a ukládání XML souborů....24
6.2.2 Použití programu MediaInfo....24
6.2.3 Zobrazení nápovědy....24
7 Ovládání a funkce aplikace....25
7.1 Ovládání....25
7.1.1 Výběr z hlavní nabídky....26
7.1.2 Přepínání typu multimédií a vyhledávání....26
7.1.3 Manipulace s adresářovým stromem....26
7.1.4 Zobrazování a manipulace s multimediálními daty....27
7.1.5 Nastavení aplikace....27
7.2 Funkce....27
7.2.1 Přidávání a odebírání multimédií....27
7.2.2 Export a import dat....28
7.2.3 Oblíbené položky....28
7.2.4 Nastavení podporovaných formátů a aplikací....29
8 Testování aplikace a její výsledky....30
8.1 Testování....30
8.2 Výsledky práce....30
9 Možnosti rozšíření....32
9.1 Multimediální přehrávač....32
9.2 Editor obrázků....32
9.3 Inteligentní rozhraní....33
9.4 Vyhledání informací....33
10 Závěr....34
Literatura....35
Seznam příloh....36

1 Úvod

Téma bakalářské práce „Osobní multimediální knihovna“ si autor vybral především z důvodu zájmu o uživatelská rozhraní, ukládání a správu velkého množství dat a v neposlední řadě i o samotná multimédia.

Uživatelská rozhraní lze v dnešní době nalézt u většiny počítačových aplikací, usnadňují a zprostředkovávají pohodlnou manipulaci jak přímo s aplikací, tak s jejím obsahem. Jsou tedy nedílnou součástí každodenní práce s osobním počítačem a mnoho aplikací by se bez nich neobešlo. Tato nesporná výhoda umožňuje běžným uživatelům počítače pracovat s daty, aniž by museli znát detaily, příkazy či jiné odbornější postupy, které jsou vlastní počítačovým odborníkům. Je tedy možné považovat uživatelská rozhraní za prvek, který výrazně přispěl k dnešní masové rozšířenosti osobních počítačů.

Multimédia jsou v současnosti jedním z nejrozšířenějších a nejpoužívanějších typů dat, které lze na osobních počítačích najít. Ve světě, kde je připojení k internetu téměř standardem v každé domácnosti a kde již jiný než digitální fotoaparát prakticky neexistuje, je jednoduchá a rychlá správa multimediální dat téměř nutností. Velké množství těchto dat spolu s jejich různými formáty a způsoby kódování potom po řešení, jakou multimediální knihovna bezesporu je, přímo vybízejí.

Aplikace, která kombinuje jak přívětivé a snadné uživatelské rozhraní a umožňuje správu velkého množství multimediálního obsahu, může koncovému uživateli výrazně usnadnit práci, vyhledávání, ale také čas.

2 Cíl práce a její metody

2.1 Cíl práce

Cílem bakalářské práce je vytvořit aplikaci, která bude shromažďovat informace o různých multimediálních datech, tyto informace bude v přehledné formě zobrazovat a umožňovat jejich snadnou manipulaci a uspořádání a také efektivní vyhledávání.

Manipulací je myšlena především možnost data ukládat do větších celků či skupin, snadno zobrazovat, přesouvat je mezi jednotlivými skupinami, pohodlným způsobem je označovat. Dále aplikace musí poskytovat jednoduchý export a import dat, umožňující migraci multimediálních dat mezi jednotlivými instancemi této aplikace. Aplikace by také měla poskytovat otevření multimediálních souborů v externím programu, který si uživatel může nadefinovat.

Uspořádání dat by mělo být založeno hlavně na snadném řazení dat podle různých klíčů, což zpřehlední a zjednoduší obsluhu aplikace a orientaci v datech. Jednotlivé klíče by měly odpovídat atributům multimediálních souborů, čímž bude zajištěno řazení dle jakéhokoliv atributu souboru.

V případě, kdy uživatel vloží do multimediální knihovny nějaká data, tedy multimediální soubory, je již na ní, aby tato data udržovala aktuální. Musí tedy poskytovat mechanismy, které toto umožňují. Může jít jak o zjištění, zda jsou soubory dostupné na zvoleném úložišti, tak možnost uložená data obnovit – tedy načíst jejich aktuální informace a atributy.

Nedílnou součástí takovéto aplikace je také vyhledávání, které je při větším množství dat, v tomto případě multimediálního obsahu, nenahraditelným pomocníkem pro uživatele a tím přispívá k celkově pohodlnému používání aplikace. Vyhledávání by mělo pro uživatele snadným a intuitivním způsobem zobrazit relevantní výsledky tak, aby je nemusel dlouze ručně dohledávat. Vhodné je pak použít tzv. fulltextové vyhledávání, což je vyhledání libovolného slova či řetězce. Tento princip lze potom použít na všechny atributy jednotlivých souborů a nabídnout uživateli všechny takové, v nichž byla nalezena shoda minimálně v jednom z atributů.

Výsledný program by měl demonstrovat použití všech výše zmíněných technik a vlastností, efektivním způsobem je využít a především je integrovat tak, aby byly funkční. V neposlední řadě by pak měl poskytnout potenciálnímu uživateli přehledný a snadno použitelný nástroj pro každodenní potřebu.

2.2 Metody práce

Jelikož cílem bakalářské práce je vytvoření aplikace, bude hlavní metodou práce především její tvorba. Tuto činnost lze rozdělit na několik hlavních bodů, které po sobě bezprostředně následují. Jednotlivé body jsou potom detailně rozebrány v dalších kapitolách.

První, avšak ne nedůležitý bod, je určení, k čemu aplikace má sloužit, co se od ní má očekávat a jaké budou výsledky práce. Je tedy nutné si stanovit cíle a ty dodržet.

V okamžiku, kdy víme, co budeme vytvářet, je vhodné se porozhlédnout po již hotových řešeních, zjistit, co je v tomto oboru standardem, inspirovat se a pokusit se některé postupy vylepšit či se

pokusit navrhnout jiné, vlastní a inovativní řešení. Toto nám pomůže udělat si obrázek o vlastních představách a možnostech.

Důležité je také zjistit podrobnosti o multimédiích (viz kapitola 3), vybrat vhodné formáty a jejich atributy. Každý formát může obsahovat jiné atributy a údaje. Proto je třeba tyto sjednotit a zobecnit tak, aby byly použitelné pro všechny soubory stejně. Pro lepší orientaci je však vhodnější rozdělit multimédia na více skupin a každé skupině potom definovat dané atributy.

V další fázi musíme definovat samotný návrh aplikace (viz kapitola 4), především její model a to v různých ohledech. Tato část je jednou z nejnáročnějších, jelikož správně vytvořený návrh výrazně usnadní pozdější práci a umožní nám efektivně integrovat všechny funkce, které jsme si dali za cíl. Návrh je také dobrým pomocníkem při pozdějších úpravách a umožňuje dobrou představu o struktuře aplikace.

Jakmile máme hotový a odladěný návrh, je nutné vybrat správné nástroje pro tvorbu samotné aplikace (viz kapitola 5). Jde jak o výběr programovacího jazyka a některých knihoven, tak o další externí a podobné aplikace, které nám umožní efektivně vytvořit aplikaci. Po pečlivém zvážení již můžeme začít se samotným programováním (viz kapitola 6), což je současně i časově nejnáročnější etapa práce.

Hotovou aplikaci je následně nutné odladit a provést co nejdetailnější testy (viz kapitola 8), aby byly co nejvíce eliminovány chyby. Bez tohoto kroku není možné považovat práci za hotovou, jelikož se mohou vyskytnout takové chyby, které by prakticky znemožnily používání aplikace samotné.

Vzhledem k tomu, že očekáváme, že aplikaci bude používat velké množství uživatelů, měli bychom detailně popsat její chování, funkce a ovládání (viz kapitola 7). Bez tohoto by bylo velmi obtížné nasadit knihovnu do běžného provozu.

Nakonec je vhodné se zamyslet, jakým dalším směrem by se výsledná práce mohla ubírat, jaké má ještě nedostatky a co by bylo možné do budoucna přidat či vylepšit (viz kapitola 9).

3 Multimédia

Za multimédia jsou v této práci považovány veškerá data či soubory s audiovizuálním obsahem. Tyto jsou v dnešní době jedním ze stěžejních využití osobních počítačů. Lidé na nich sledují filmy, televizi, poslouchají hudbu, prohlížejí si obrázky a fotky. Všechna tato multimédia mají v sobě uloženy jisté atributy, které mohou být významným zdrojem informací pro uživatele a je tedy vhodné je využít.

3.1 Rozdělení multimediálních dat

Z důvodu rozdílných atributů u každého typu souborů je lepší rozdělit je do různých skupin. Bylo by sice možné všechna multimédia zobrazit dohromady, ale pro uživatele by to mělo nepříjemné důsledky. Různé multimediální soubory mají různé atributy a nastal by problém s korektním zobrazením všech atributů jednotlivých souborů najednou. Mnohem přehlednějším řešením je rozdělení na kategorie či typy, které mají společné atributy. Budeme vycházet z toho, jaká multimediální data soubor obsahuje.

Formát dat lze určit jak podle typu stopy, kterou obsahuje, tak podle kódování, ukládání dat a dalších vlastností. Jelikož chceme, aby byla naše výsledná aplikace byla multiplatformní a uživatelsky přívětivá, rozdělíme formáty snadno podle přípony. Můžeme tedy říci, že co přípona, to jiný formát. Ve skutečnosti to sice tak úplně neplatí, protože například přípony JPG a JPEG jsou tentýž formát, ale z výše uvedených důvodů si můžeme dovolit toto opomenout

Při výběru, jaké formáty souborů bychom chtěli podporovat, musíme vzít v potaz také to, jaký nástroj použijeme k extrakci dat ze souborů. Měli bychom si tedy promyslet naše možnosti a podle toho navrhnout atributy, které bychom chtěli ukládat.

Jelikož budeme využívat programu MediaInfo [1] (viz kapitola 5.3), budeme vycházet ze specifikací, které tento nástroj poskytuje. Jedná se jak o podporu různých formátů, tak jejich stop a atributů. Atributy každého souboru jsou rozděleny na stopy, konkrétně se jedná o general (obecná), audio a video. Lze najít i některé další stopy, ale ty pro nás nejsou příliš zajímavé. Podle toho, jaké stopy multimediální soubor obsahuje, potom rozdělíme multimediální obsah na tři typy – audio, video a obrázky.

3.2 Audio

Multimediální soubory typu audio obsahují obecnou (general) a zvukovou (audio) stopu. V obecné stopě najdeme především název, délku, autora a další metadata, které můžeme znát například jako ID3 tagy. Audio stopa potom obsahuje informace jako kodeky, počet kanálů apod. Obsah jednotlivých stop je shrnut v tabulce 3.1, kde je zároveň i uvedeno, které formáty jsou podporovány a které atributy obsahují. Ta byla vytvořena na základě specifikací programu MediaInfo, obsahuje však jen atributy důležité pro aplikaci. Některé z nich byly záměrně vynechány.

	mp3	**wav**	**wma**	**ogg**	**ac3**	**aac**	**flac**
Obecná stopa	ano	část	část	ano	ano	ano	ano
Délka	ano	ano	ano	ano	ne	ne	ano
Autor	ano	ne	ano	ano	ne	ne	ano
Název	ano	ne	ano	ano	ne	ne	ano
Album	ano	ne	ano	ano	ne	ne	ano
Číslo stopy	ano	ne	ne	ano	ne	ne	ano
Počet stop	ano	ne	ano	ne	ne	ne	ne
Datum	ano	ne	ne	ano	ne	ne	ne
Datum nahrání	ano	ne	ano	ne	ne	ne	ne
Komentář	ano	ne	ano	ano	ne	ne	ano
Copyright	ano	ne	ano	ano	ne	ne	ne
URL	ano	ne	ne	ano	ne	ne	ne
Audio	ano	ano	ano	ano	ano	ano	ano
Kodek	ano	ano	ano	ano	ano	ano	ano
Přen. rychlost	ano	ano	ano	ano	ano	ano	ano
Počet kanálů	ano	ano	ano	ano	ano	ano	ano

Tabulka 3.1 – audio atributy

Veškeré formáty, které jsou uvedeny v tabulce 3.1 by měla aplikace podporovat. Jde o nejrozšířenější audio formáty a je tedy vhodné je všechny zahrnout.

K vybraným formátům je nutné zvolit také jisté atributy, které budeme extrahovat z multimediálních souborů. Jelikož ne všechny formáty podporují stejné atributy, je třeba vybrat nějaký společný jmenovatel. Budeme však vycházet především z formátu mp3, který je dnes nejrozšířenějším a nejvíce používaným. Ten obsahuje navíc také tzv. ID3 tagy [2], tedy jistá metadata informující o interpretovi, albu apod. Existuje více verzí ID3 tagů, my se však můžeme spokojit s verzí 1, která je jednodušší. Tato verze poskytuje tato metadata: umělec (artist), album, název (title), žánr (genre), hodnocení (rating), obal alba (album art). Tyto atributy potom můžeme zahrnout do konečné množiny. Mezi dalšími formáty, které podporují tyto tagy, je například Ogg Vorbis.

Aplikace tedy bude podporovat následující atributy pro audio soubory:

- délka
- interpret
- název
- žánr
- album
- komentář
- přenosová rychlost
- počet kanálů
- formát

3.3 Video

Multimediální video soubory se vyznačují hlavně tím, že jsou audiovizuální, tedy obsahují kromě stopy general (obecná) také audio a video stopu. To přináší především velké množství atributů, které lze z těchto souborů extrahovat. Kromě již v kapitole 3.2 popsaných stop general a audio, je nutné se seznámit i s video stopou. V té můžeme nalézt například video kodek, poměr stran videa, přenosovou rychlost a další. Všechny atributy jsou potom shrnuty v tabulce 3.2.

	avi	**mpeg**	**mp4/m4a**	**wmv/asf**	**mkv**	**mov**
Obecná stopa	část	ano	ano	část	část	část
Délka	ano	ano	ano	ano	ne	ano
Autor	nene	ne	ano	ano	ne	ano
Název	ne	ne	ano	ano	ne	ano
Komentář	ne	ne	ano	ano	ne	ne
Copyright	ne	ne	ne	ano	ne	ne
URL	ne	ne	ne	ne	ne	ne
Audio stopa	ano	ano	ano	ano	ano	ano
Kodek	ano	ano	ano	ano	ano	ano
Přen. rychlost	ano	ano	ano	ano	ano	ano
Počet kanálů	ano	ano	ne	ano	ano	ano
Video stopa	ano	ano	ano	ano	ano	ano
Kodek	ano	ano	ano	ano	ano	ano
Přen. rychlost	ano	ano	ano	ano	ano	ano
Poměr stran	ano	ano	ano	ano	ano	ano

Tabulka 3.2 – video atributy

Aplikace by měla stejně jako u audio souborů podporovat veškeré formáty uvedené v tabulce 3.2. Opět budeme vycházet z běžně používaných formátů.

Atributy budeme volit především s ohledem na video stopu, zde tedy nechceme zahrnout například ID3 tagy, které se u tohoto typu multimédií příliš nevyužívají. Částečně tedy vypustíme informace z obecné stopy a zaměříme se především na audiovizuální atributy. Z těch potom vybereme tyto:

- délka
- komentář
- formát
- audio – kodek
- audio – přenosová rychlost
- audio – počet kanálů
- video – kodek
- video – přenosová rychlost
- video – poměr stran

3.4 Obrázky

Obrázky můžeme také považovat za multimediální data, žádnou zvláštní stopu však neobsahují. Naštěstí ovšem můžeme dostatek informací nalézt přímo v obecné stopě a není proto nutné hledat další možnosti. Vzhledem k tomu, že MediaInfo neposkytuje specifikace pro obrázky, je nutné hledat jejich atributy přímo ve výstupu tohoto programu. Ty potom musíme otestovat na zvolených formátech a zjistit tak, které může aplikace podporovat. Nejdůležitější je potom podpora především formátu jpeg, který je nejrozšířenější a slouží především pro ukládání fotografií, což je jedna z hlavních funkcí multimediální knihovny. Po ověření tedy můžeme podporovat tyto formáty:

- jpeg/jpg
- png
- bmp
- gif

K těmto formátům potom zvolíme množinu atributů:

- formát
- výška
- šířka
- barevná hloubka

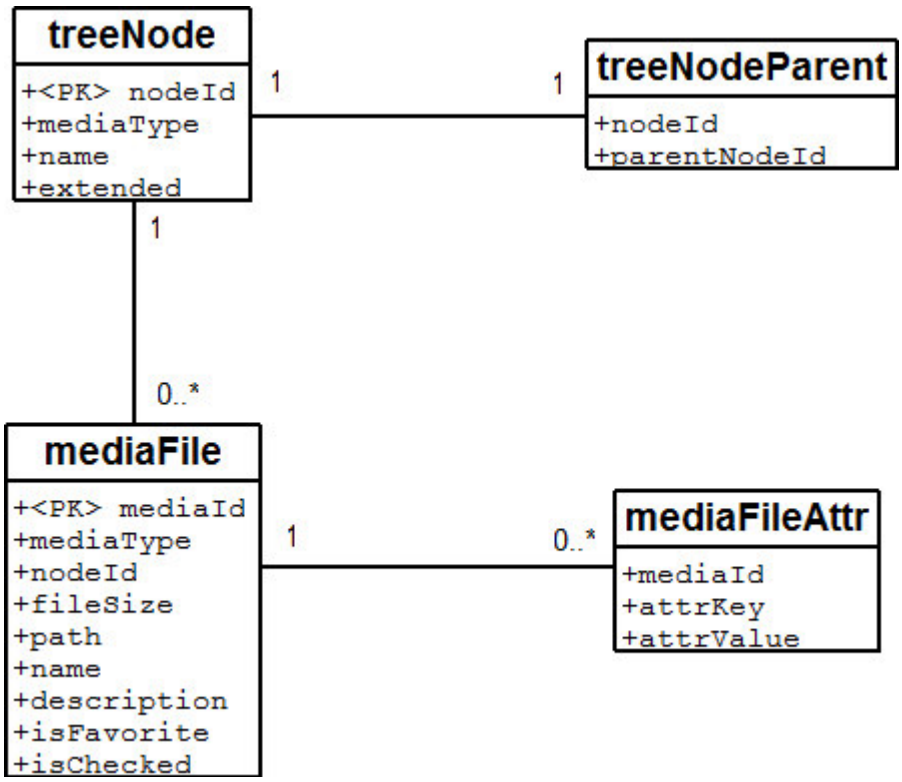
Mohli bychom sice nalézt mnohem širší spektrum obrázků a multimédií tohoto typu, avšak velmi často se jedná o velice specializované formáty např. pro účely tvorby grafiky a vektorových dat, které ale jsou mimo rámec multimediální knihovny a nemusíme se s nimi proto zabývat.

4 Návrh aplikace

4.1 Databázový model

Databázový model je jedním ze stěžejních modelů celé aplikace. V době běhu aplikace se přistupuje k databázi téměř stále, a proto je kvalitní a dobře navržený model velmi důležitý.

Vzhledem k tomu, že veškerá nastavení jsou ukládána zvlášť (viz kapitola 4.2), je nutné do databáze ukládat pouze informace o multimédiích a jejich zařazení do adresářového stromu. Toto je pak realizováno 4 tabulkami (viz obrázek 4.1). První tabulkou je `treeNode`, která ukládá data o jednotlivých uzlech stromu, tedy o adresářích. Jednotlivé propojení mezi uzly je pak uloženo v tabulce `treeNodeParent`, kde má každý uzel definován svůj rodičovský uzel. Ke každému uzlu navíc náleží libovolný počet souborů z tabulky `mediaFile`. Reprezentuje to tedy složku a její obsah. Každý multimediální soubor může navíc mít definován libovolný počet atributů, které jsou uloženy v tabulce `mediaFileAttr`. Výhodou je, že není nutné mít pro každý typ multimediálního souboru zvláštní tabulku, jelikož všechny mají společné některé vlastnosti. Ty atributy, které se liší, jsou potom po jednom uloženy právě v tabulce `mediaFileAttr`. Toto zajišťuje dobrou flexibilitu aplikace a návrh je tak použitelný pro jakékoliv atributy multimediálních souborů.



Obrázek 4.1 – databázový model

4.2 Struktura XML dokumentů

Pro ukládání některých dat aplikace do souborů bude výhodné použít dokumentů XML verze 1.0 [3], které poskytují mnoho možností, jak data efektivně zapsat a opětovně je číst.

4.2.1 Ukládání nastavení

Dokumenty formátu XML budou využívány pro uložení veškerého nastavení programu. K dispozici sice bude i databáze, avšak je vhodné tyto typy dat oddělit. Důvodů je pro to více – databáze nemusí být vždy dostupná, samotné nastavení programu by tedy bylo závislé na ní a aplikace se tím pádem nedala ani spustit. To je samozřejmě nepřípustné. Dalším problémem je potom fakt, že samotná data nastavení potřebujeme pro připojení databáze. Je tedy nutné je uložit mimo ni. Posledním závažným faktem pak může být, že připojení k databázi a načtení potřebných dat může trvat delší čas a nastavení programu jsou potřebná především při spuštění aplikace.

Struktura XML dokumentu pro uložení nastavení je prostá, vždy je třeba ukládat pouze klíč, pomocí kterého k danému nastavení přistupujeme a jeho hodnotu. Tato struktura je znázorněna na obrázku 4.2.

```
<settings>
    <option>
        <key></key>
        <value></value>
    </option>
</settings>
```

Obrázek 4.2 – XML schéma pro nastavení

Každá položka nastavení pak bude tvořit element `option`, který bude dále obsahovat klíč a hodnotu tohoto nastavení.

4.2.2 Export/Import multimédií

Pro exportování či importování multimédií je XML vhodným prostředkem. Bylo by sice možné použít například SQL dotazy a uložit je do textového souboru, avšak to s sebou nese několik problémů. Pravděpodobně nejzávažnějším je nutnost při importování multimédií namapovat identifikátory na nové hodnoty, takže by bylo nutné procházet celé dotazy. K tomuto je XML vhodnější, jelikož je velmi snadné ho procházet.

Při exportu je nutné uložit jak multimédia samotná, tak složky, do kterých jsou zařazena. Výsledný XML dokument tedy obsahuje nejprve samotné složky včetně identifikátorů a následně multimédia, opět s identifikátory. Právě tyto jsou pak při importu mapovány na nové. U multimédií je důležité, že jsou zde uloženy jednak obecné informace, které jsou pro každý typ multimédia shodné a dále atributy ve tvaru klíč a hodnota. To umožní uložit do XML dokumentu jakékoliv atributy a jejich hodnoty a toto řešení je tedy univerzální. Při změně názvu či počtu atributů nebude problém je uložit do dokumentu.

Schéma dokumentu je znázorněno na obrázku 4.3, v tomto případě pro jeden uzel stromu a jeden multimediální soubor.

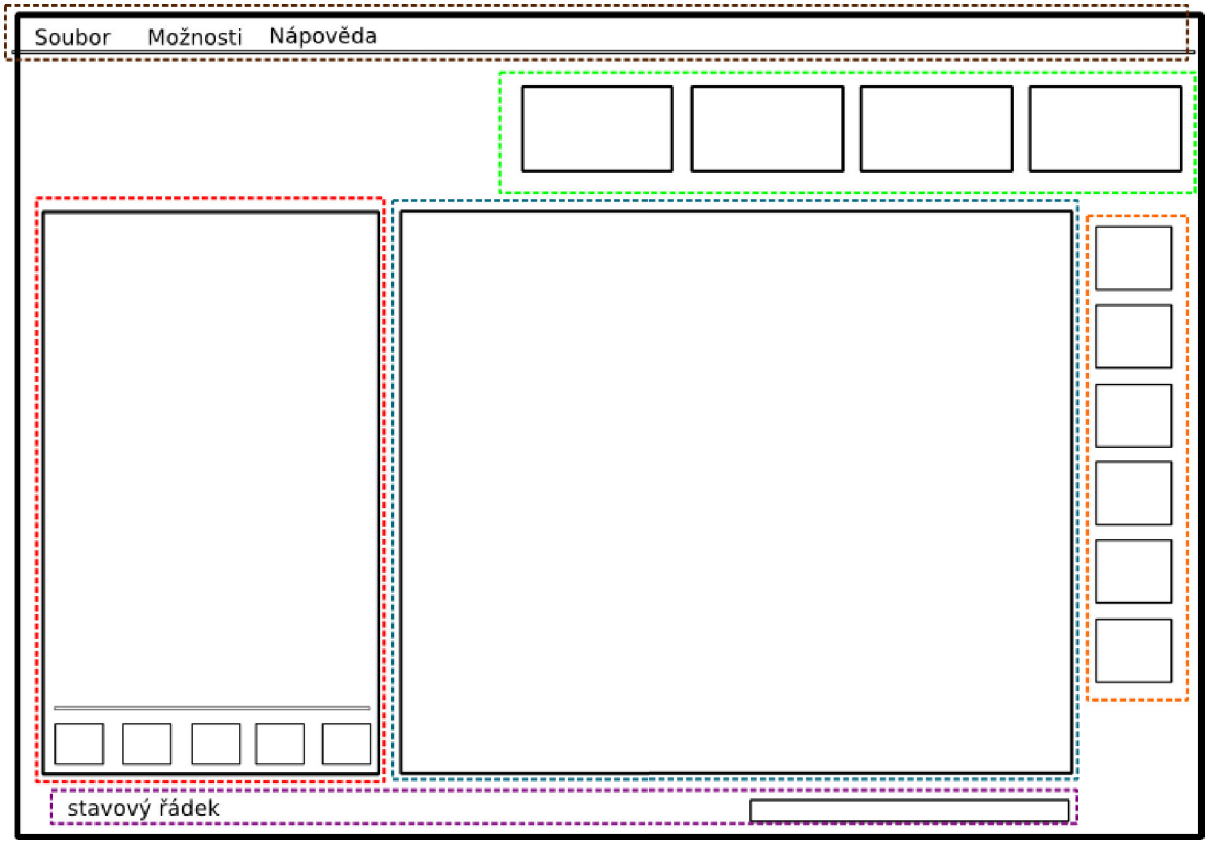
```
<export>
      <treeNodes>
            <node>
                  <nodeId></nodeId>
                  <nodeName></nodeName>
                  <nodeMediaType></nodeMediaType>
                  <parentNodeId></parentNodeId>
            </node>
      </treeNodes>
      <mediaFiles>
            <mediaFile>
                  <mediaId></mediaId>
                  <name></name>
                  <description></description>
                  <fileSize></fileSize>
                  <nodeId></nodeId>
                  <mediaType></mediaType>
                  <path></path>
                  <isFavorite></isFavorite>
                  <attributes>
                        <attribute>
                              <key></key>
                              <value></value>
                        </attribute>
                  </attributes>
            </mediaFile>
      </mediaFiles>
</export>
```

Obrázek 4.3 – XML schéma exportu

4.3 Grafické uživatelské rozhraní

Uživatelské rozhraní je prostředek pro ovládání aplikace. Toto může být jak textové, tedy pouze pomocí příkazů v příkazové řádce či může mít grafickou podobu. [4] Jelikož má multimediální knihovna za úkol zobrazovat velké množství dat v co možná nejpřehlednější formě, je nejvýhodnější volbou právě grafické uživatelské rozhraní. Výhodou je bezesporu kromě přehlednosti i snadná manipulace s daty a především také jednoduché a intuitivní ovládání. Uživatel není nucen se učit spoustu příkazů pro ovládání aplikace a pro naprostou většinu úkonů mu postačí pouze myš.

Celé rozhraní lze rozdělit na několik menších logických celků, kde má každý na starost jistou funkčnost aplikace (viz obrázek 4.4). Je vhodné, aby každý celek působil celistvě a měl své pevné místo v rámci celého grafického vzhledu. Tím je zaručeno, že uživatel ovládání bez problémů pochopí, nebude mít potíže s ovládáním programu a vše bude pohodlné.



Obrázek 4.4 – rozdělení grafického rozhraní na logické celky

4.3.1 Hlavní nabídka

Hlavní nabídka (na obrázku 4.4 v hnědém rámečku) se typicky používá u grafických rozhraní a je vhodná pro zobrazení běžné nabídky operací jako např. uložení, nastavení apod. Bylo by možné do ní umístit veškeré akce, které lze v aplikaci provádět, avšak nebylo by to příliš použitelné. Obecně lze říci, že by se zde neměly nacházet ty, které uživatel bude používat často a byl by proto nucen neustále je vyhledávat v této nabídce. Jistým urychlením je sice použití klávesových zkratek, které lze těmto operacím přiřadit, ne každý uživatel ale může být na toto chování zvyklý, a proto to nelze považovat za obecné řešení.

4.3.2 Tlačítka multimédií

Důležitým a ve velké míře používaným prvkem by měly být tlačítka pro přepínání mezi jednotlivými multimédii (typy multimédií). Tento blok je na obrázku 4.4 znázorněn zeleným rámečkem. Jednoduchým použitím daného tlačítka se aplikace přepne na zobrazení vybraného typu multimediálních souborů, jde tedy o jakýsi výběr kategorie. Dále sem lze zahrnout i vyhledávací tlačítko, které je možné použít pro vyhledávání v rámci zvolené kategorie. To může místo přepnutí kategorie například zobrazit nové okno s kritérii vyhledávání.

4.3.3 Adresářový strom

Adresářový strom (na obrázku 4.4 vyznačen červeně) je nutný především pro přehledné zobrazení jednotlivých multimediálních souborů. Díky němu je možné tyto soubory snadno třídit do různých skupin, které představují jednotlivé uzly stromu – adresáře (viz obrázek 4.5). Strom bude podporovat jak vytvoření vlastních adresářů nezávislých na adresářové struktuře pevného disku či jiného datového úložiště, tak kopírovat právě jeho strukturu. Také by měl umožňovat jednoduché přesuny multimédií mezi adresáři pouhým přetažením myší. Takto to funguje běžně v operačních systémech a pro uživatele je to velmi pohodlné a intuitivní. Celý strom pak lze spravovat pomocí tlačítek umístěných v jeho spodní části jako například přidání adresáře do stromu, jeho odebrání, vložení souborů do adresáře apod.

Obr. 4.5 – adresářový strom

4.3.4 Tabulka multimédií

Hlavní částí celého grafického rozhraní je tabulka multimédií (na obrázku 4.4 modře orámována), která zobrazuje uložené soubory ve zvoleném adresáři stromu. Forma tabulky je výhodná hned z několika důvodů – je přehledná, lze v ní zobrazit velké množství dat a především je snadno řadit podle zvoleného klíče. Řazení probíhá jednoduchým kliknutím na záhlaví zvoleného sloupce. V tabulce pak lze zobrazit jak data a jejich atributy, tak další informace jako obrázky, zaškrtávací políčka a jiné. Ty je potom možné použít například pro zobrazení oblíbených položek apod. (viz obrázek 4.6).

		jméno	popis	velikost [KB]
★	☐	Track 1	První track	3446.66
☆	☑	Track 2	Druhý track	5316.04

Obrázek 4.6 – tabulka multimédií

4.3.5 Tlačítka pro práci s multimédii

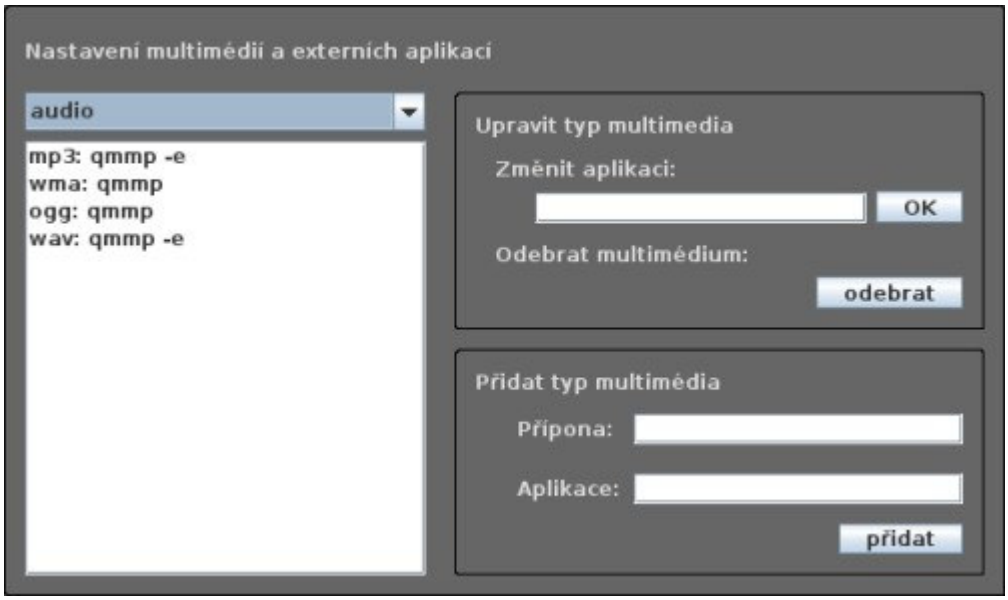
Blok tlačítek pro různé operace nad multimediálním soubory (na obrázku 4.4 orámovaný oranžově) je úzce spjatý s daty v tabulce multimédií. Tato tlačítka poskytují různé funkce pro označené položky tabulky, např. úpravu položek, jejich vymazání apod. Jde o ty, které by měly být z pohledu uživatele nejčastěji používané. Jelikož prostor pro tlačítka je omezený, je nutné dobře zvolit ty, jenž opravdu nejvíce usnadňují práci a jejich umístění má opodstatnění. Ostatní funkce je možné zařadit do hlavní nabídky.

4.3.6 Stavový řádek

Stavový řádek (na obrázku 4.4 fialově označený) má pouze informativní charakter, avšak uživateli poskytuje velmi hodnotné informace o stavu aplikace. V jeho textové části je zobrazena informace o načtených a případně označených položkách (např. jejich počet). V případě, že aplikace provádí nějakou akci, lze potom informovat o tom, co se právě provádí. S tím úzce souvisí i umístění stavového pruhu, který zobrazuje, kolik procent položek již bylo zpracováno. To je velmi důležité, jelikož některé operace mohou trvat dlouhou dobu (i v řádu minut) a uživatel by mohl nabýt dojmu, že aplikace není aktivní.

4.3.7 Okno nastavení

Pro nastavení programu bude vytvořeno zvláštní okno. Toto je pak důležité především pro asociaci jednotlivých multimediálních souborů s externími aplikacemi. Okno by mělo respektovat rozdělení multimédií a umožňovat každému formátu přiřadit aplikaci a také komplexně jednotlivé formáty spravovat. To představuje především přidávání či odstraňování těchto formátů. Návrh tohoto okna je zobrazen na obrázku 4.7.



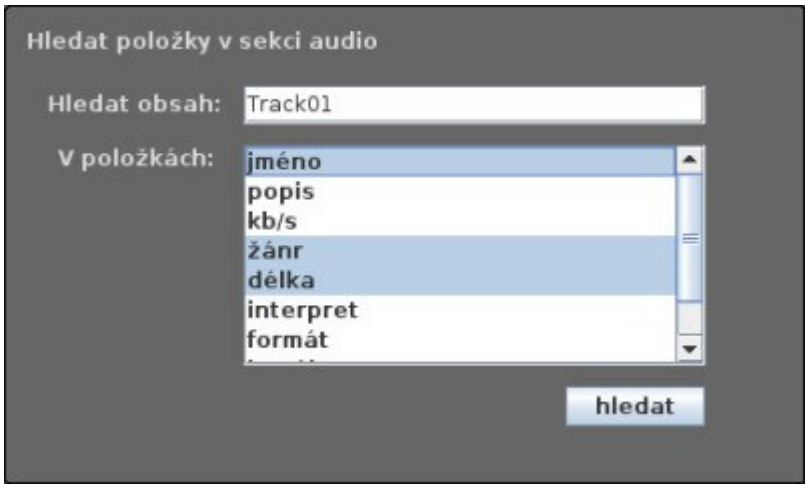
Obrázek 4.7 – okno nastavení

4.3.8 Okno vyhledávání

Vyhledávání ve velkém množství dat je jednou z důležitých vlastností multimediální knihovny a je velmi pravděpodobné, že bude často používáno. Je proto nutné ho účelně integrovat do grafického rozhraní. V nastíněném konceptu se nabízí několik variant – nové okno může obsahovat jak grafické prvky pro vyhledávání, tedy textové pole, výběr vyhledávacích klíčů apod., avšak také kompletní tabulku s výsledky vyhledávání. Další možností je tabulku z vyhledávacího okna odstranit a použít tu, která je již integrována v grafickém rozhraní. Poslední variantou je potom použít jednoduchý filtr, který v aktuálně zobrazené tabulce pouze filtruje data a ty potom zobrazuje.

Nejvhodnějším z těchto řešení bude zvolit vyhledávací okno bez další tabulky, jelikož by byla nová tabulka zbytečnou komplikací jak z hlediska návrhu, tak z uživatelského pohodlí. Filtr by naopak byl příliš jednoduchý a bylo by možné filtrovat data pouze v aktuálně zobrazených.

Jednoduché vyhledávací okno s kritérii vyhledávání je potom zobrazeno na obrázku 4.8.



Obrázek 4.8 – vyhledávací okno

4.3.9 Okno pro úpravy

Je nutné, aby měl uživatel možnost do jisté míry upravovat samotná data. Samozřejmostí jsou jistá omezení, jelikož úpravy některých dat nejsou žádoucí. Toto by mělo zajistit jednoduché okno, které se zobrazí např. po kliknutí na určité tlačítko. V něm se potom zobrazí položky, které lze upravovat a tlačítko pro potvrzení. Jako minimum toho, co by zde mělo být možné upravovat, lze považovat jméno souboru a jeho popis, případně zobrazit absolutní cestu k souboru pro lepší orientaci.

4.3.10 Nápověda

Alespoň základní nápověda je pro aplikaci velmi důležitá. Napomáhá uživateli se základními úkony a především při seznámení s aplikací může být užitečná. Měla by se zobrazit v novém okně, a to v běžném stylu, tedy vlevo kapitoly nápovědy či rejstřík a vpravo samotný text. Takto je to dnes zavedeno ve velkém množství aplikací a není tedy důvod toto schéma nijak měnit.

4.3.11 Tray ikona

Tray ikona je ikona v systémovém panelu, kde se zobrazují některé právě běžící procesy. Její použití může usnadnit uživateli přístup k aplikaci v operačním systému, jelikož je ikona stále vidět. To mohou ocenit zejména uživatelé, kteří používají ve svém operačním sytému více pracovních ploch a v pruhu úloh tak nemusí mít aplikaci stále zobrazenou. Jednoduchým kliknutím na ikonu se pak aplikace může okamžitě zobrazit. Navíc jí lze přiřadit některé operace přes kontextové menu.

4.4 Objektový model

Jelikož budeme aplikaci vytvářet v objektově orientovaném jazyce (viz kapitola 5.1), je nutné si předem připravit objektový model. Podle tohoto modelu budeme následně postupovat při implementaci celé aplikace. Vhodným prostředkem pro tento model je E-R diagram, který popisuje vztahy mezi jednotlivými entitami, v tomto případě objekty. [5] Diagram je vložen jako příloha 1.

Vzhledem k tomu, že aplikace bude sestávat z většího množství tříd, kde ne všechny jsou významné, omezíme se kvůli přehlednosti při návrhu pouze na základní třídy, které dobře reprezentují strukturu objektového modelu. Z modelu tak vynecháme pomocné třídy, které poskytují pouze vedlejší operace a nejsou nezbytné pro pochopení struktury aplikace. Jedná se především o třídy reprezentující některé grafické prvky jako okna či vykreslování komponent a dále také nástroje pro práci se soubory, jejich filtrování či manipulaci.

Další zjednodušením potom je vynechání některých atributů, které zastupují grafické prvky a jsou generovány vývojovým prostředím při tvorbě grafického rozhraní. To stejné platí i pro některé metody jako inicializace okna apod.

Jelikož se jedná pouze o návrh, je velmi pravděpodobné, že bude ještě v průběhu implementace změněn především ve smyslu přidání některých atributů či metod. Není tedy kladen důraz na stoprocentní správnost vytvořeného modelu, ale spíše na uvědomění si určitých principů a postupů při samotném návrhu. To potom usnadní samotnou implementaci.

4.4.1 Třídy diagramu

V diagramu (příloha 1) jsou obsaženy nejdůležitější a základní třídy aplikace. Jádro tvoří třída `Core`, která propojuje a integruje funkčnost ostatních tříd. Třída `Database` bude umožňovat připojení se k databázi a manipulaci s ní, `Settings` potom obstará načítání a ukládání nastavení aplikace pomocí XML souborů. Pomocí třídy `Window` bude ovládáno a vykresleno grafické rozhraní včetně manipulace s adresářovým stromem a tabulkou multimédií. Důležité je také propojení s třídou `Media`, která bude obsahovat právě načtená multimédia. Mezi výše uvedenými třídami platí vztah asociace, zatímco mezi třídou Media a abstraktní `MediaFile` jde o vztah kompozice, kde `MediaFile` je částí třídy `Media`. Tento vztah se také nazývá celek/část. Jednotlivé třídy `MediaAudio`, `MediaVideo` a `MediaImage` potom z třídy `MediaFile` dědí a implementují její abstraktní metody.

Popis jednotlivých tříd je pouze velmi stručný, avšak hlavní vypovídající hodnotu má především E-R diagram. Detailněji je pak propojení tříd popsáno v kapitole 6, kde je rozebrána implementace.

5 Použité nástroje

5.1 Programovací jazyk – Java

Autor po pečlivém zvážení zvolil programovací jazyk Java (verze Java SE 6) [6], který je v dnešní době velmi rozšířen. Obsahuje spoustu vhodných nástrojů, návrhové vzory, integrovaná vývojová prostředí, která významně usnadňují práci a také velké množství volně dostupných knihoven. Dalším významným faktorem je řada knih a příruček [7, 8, 9] popisujících tvorbu programů v tomto jazyce a také mnoho referenčních případů, které lze dohledat na internetu. Jazyk Java navíc poskytuje výraznou výhodu v multiplatformnosti, tedy možnosti spustit aplikaci na různých operačních systémech. Jelikož toto byl od začátku jeden ze zásadních pilířů celé aplikace, lze to považovat za nezanedbatelnou vlastnost.

Jazyk Java je objektově orientovaný, kompilovaný programovací jazyk. Přináší to však nejen výhody, ale i jistá úskalí. Jelikož je kladen důraz na multiplatformnost, je nutné mít pro běh aplikace nainstalován její virtuální stroj (Java Virtual Machine – JVM). Tento stroj potom kompiluje zdrojový kód do tzv. „bytekódu“, který následně provádí. V praxi je tato činnost výrazně složitější, například nejsou kompilovány všechny části zdrojového kódu apod., ale to pro nastínění situace není důležité. Tento mechanismus však může částečně zpomalovat start či běh aplikace, což není vždy žádoucí. Může to však vynahradit velkou robustností, bezpečností, značnou jednoduchostí či například propracovaným systémem výjimek a jejich odchycením. Tento systém může do značné míry usnadnit eliminaci nežádoucích stavů aplikace a tím výrazně zvýšit bezpečnost.

Při vývoji aplikací v tomto programovacím jazyce je nutné dopředu myslet na řadu dalších aspektů. Vzhledem k tomu, že je striktně objektově orientovaný a nelze v něm tedy programovat jiným způsobem, musí tomu odpovídat také postup při řešení a návrhu aplikace. Důležitý je návrh kostry aplikace, nejlépe formou diagramu tříd. Toto nám jednak pomůže při pozdějším programování, kdy budeme přesně znát vztahy mezi jednotlivými třídami a jednak také při pozdějším náhledu do zdrojového kódu, kdy se lépe zorientujeme v jeho souvislostech.

5.1.1 Vývojové prostředí

Jako vhodnou platformu pro tvorbu aplikace si autor vybral integrované vývojové prostředí (IDE) NetBeans, které je volně dostupné pod licencí GNU GPL 2. S tím již měl jisté zkušenosti a volba proto nebyla náročná. Toto prostředí nabízí řadu výhod jako neustálou kontrolu syntaxe programu, částečnou sémantickou kontrolu - například správně zapsané identifikátory, třídy, typovou kontrolu a mnoho dalších. Dále usnadňuje spuštění aplikace a odhalování chyb, jelikož lze program při běhu snadno odkrokovat a ladit. Pro tvorbu uživatelského rozhraní je velmi výhodná možnost grafického návrhu jednotlivých oken aplikace a jejich obsahu. Celkově lze říci, že vývojové prostředí zásadním způsobem usnadňuje práci a vývoj aplikace.

5.2 Databáze

Databáze je důležitým prvkem pro ukládání většího množství dat. Jelikož aplikace jako multimediální knihovna potřebuje ukládat veškerá data o svém obsahu, tedy o multimediálních souborech, je databáze ideálním řešením. Je to určitá spojitá množina informací či dat, v širším smyslu obsahuje i prostředky pro manipulaci s těmito daty a pro přístup k nim. Toto se nazývá systém řízení báze dat. V dnešní době jsou běžné relační databáze, které data ukládají ve formě tabulek.

5.2.1 Výběr databáze

Existují různé typy databází, lze je rozdělit jak dle jazyka, kterým se s databází komunikuje, tak podle jejich architektury. Nejrozšířenější databáze dnes používají jazyk SQL a jeho různé modifikace. Architektura potom bývá nejčastěji typu klient/server, kdy se klient připojuje na databázový server, který zpracuje jeho požadavek a pošle klientovi požadovaná data. Tyto servery jsou pak většinou složitějšími programy s různými architekturami a systémy ukládání dat. Další možností jsou různé jednoduché databáze, které mohou být uloženy pouze v jednom souboru, tedy jak systém řízení báze dat, tak vlastní data. Ty jsou potom většinou zjednodušené oproti velkým serverům, poskytují však výhody v podobě malé velikosti, snadné přenositelnosti a použitelnosti u menších aplikací.

5.2.2 SQLite

Autor si pro svoji aplikaci vybral odlehčenou databázi SQLite verze 3.7.6 [10]. Oproti velkým serverům má nespornou výhodu v malé velikosti a naprosté přenositelnosti. Je uložena pouze v jednom souboru společně s transakčním systémem řízení báze dat a nevyžaduje žádné další knihovny pro svou vlastní činnost. Z toho tedy vyplývá, že je databáze nezávislá a lze ji jednoduše připojit k aplikaci. Další výhodou je také její uložení na disku, které nezávisí na platformě. Bez problémů ji lze kopírovat mezi různými počítačovými architekturami, aniž by došlo k poškození či ztrátě dat. Tyto výhody ji přímo předurčují k použití pro multiplatformní aplikaci. Jako nevýhoda může být považováno, že databáze nemá svůj vlastní proces, a tudíž by mohla s vyššími nároky ztrácet výkon, avšak požadavky menší aplikace bez problémů splní.

Databáze SQLite implementuje značnou část z jazyka SQL92, i když jí některé součásti chybí. Pro tuto aplikaci jsou však běžné databázové operace naprosto dostačující a její omezení nehrají žádnou roli. Její kód je licencován jako Public domain, v překladu tedy Volné dílo, takže ji lze bezproblémově využít pro jakékoliv aplikace.

Tato databáze je také vysoce odolná proti poškození, umožňuje zamykání při provádění operací a tudíž znemožňuje přístup více aplikacím naráz. To je výhodné především pokud je spuštěno více instancí stejné aplikace a mohlo by dojít k porušení referenční integrity. Dalším důležitým faktorem z hlediska bezpečnosti je například systém ukládání dat na pevný disk. Data jsou zapisována okamžitě po uložení do databáze a je zajištěna atomicita operace, nemůže tak dojít k nechtěným poškozením datové báze.

5.3 MediaInfo

Vzhledem k tomu, že multimediální knihovna shromažďuje informace o multimediálních souborech, je nutné, aby byla schopna tyto informace ze souborů efektivně extrahovat. Toto lze udělat více způsoby – lze si přímo naprogramovat nástroj, který načte každý soubor a požadované informace si z něj extrahuje. Bohužel však multimédia nejsou jednotná a formátů je celá řada, takže by tento nástroj pravděpodobně výrazně přesáhl rozsah práce. Existují však i volně dostupné nástroje, které umožňují efektivně extrahovat požadované informace a vypsat je. Zároveň je zde zaručená vysoká kompatibilita s běžnými multimediálními formáty. Jedním z nich je i pro knihovnu použitý program MediaInfo verze 0.7.44 [1].

Tento program je volně dostupný pod GNU GPL licencí, je k dispozici pro celou řadu platforem od Windows, přes různé distribuce Linuxu až po Mac OSX a je pravidelně aktualizován. Je schopen extrahovat informace z široké řady multimediálních formátů (kontejnerů), prakticky mezi nimi najdeme všechny běžné formáty. Z multimédií navíc umí extrahovat i jednotlivé stopy a jejich atributy, takže lze snadno analyzovat soubor ve všech ohledech.

MediaInfo lze používat ve dvou verzích, a to buď grafické nebo jako konzolová aplikace. Pro potřeby multimediální knihovny je konzolová verze ideální. Lze ji snadno spustit jako externí aplikaci a předat jí požadované soubory jako argumenty. Výstup je potom možné přepnout dle potřeb. Program podporuje jak nativní výstup ve formátu klíč : hodnota, který je sice použitelný, ale jako lepší varianta se jeví XML. Toto lze pohodlně načíst a následně pomocí některé knihovny jazyka Java snadno najít a uložit atributy, které hledáme.

6 Implementace

Implementace je stěžejní částí celé práce. Máme sice již hotový návrh, ale nyní přichází na řadu převést ho do reality. Není to vždy snadné a často se může stát, že některé naše myšlenky z předchozích kapitol nemusí být tak snadno proveditelné, případně zjistíme, že jdou věci udělat lépe či snadnější cestou. Přesto je vhodné se návrhu co nejvíce držet, abychom si nezanesli do programu nějaké neočekávané chyby a problémy. Především databázový model, struktura XML dokumentů a grafické rozhraní by měly být implementovány v pokud možno nezměněné podobě. Co se týče objektového modelu, zde je situace již složitější a dá se předpokládat, že k nějakým změnám dojde. Může jít například o přidání dalších metod, se kterými se původně nepočítalo, nebo přidání nějakých pomocných atributů.

6.1 Struktura aplikace

Aplikace je rozdělena do několika balíčků, kde každý sdružuje třídy zajišťující jistou sadu operací. V každém balíčku jsou třídy, které spolu nějak souvisejí, mají jisté společné znaky. Seznam jednotlivých balíčku je uveden v tabulce 6.1.

balíček	**funkce**
pml.core	jádro aplikace
pml.db	rozhraní pro manipulaci s databází
pml.file	třídy pro filtrování souborů
pml.gui	grafické rozhraní aplikace, jednotlivá okna
pml.media	třídy pro práci s multimediálními soubory
pml.settings	rozhraní pro načítání a ukládání nastavení aplikace
pml.table	třídy pro vykreslování multimediální tabulky
pml.tree	třídy pro vykreslování adresářového stromu
pml.utility	pomocné třídy a nástroje

Tabulka 6.1 – struktura balíčků

Jádro celé aplikace tvoří třída `Core`. V té jsou integrovány další třídy, které se specializují na jednotlivé operace. Zároveň reprezentuje i hlavní okno celé aplikace. Tento přístup byl zvolen na základě autorových předchozích zkušeností s programovacím jazykem Java. Jeho výhodou je především možnost přistupovat ke všem dalším operacím celé aplikace z jednoho místa a integrovat tak veškerou funkčnost. Její atributy jsou potom další specializované třídy pro přístup k databázi, nastavení aplikace, manipulaci s multimediálními soubory, vykreslování obsahu do okna aplikace či další okna aplikace jako okno vyhledávání nebo nastavení. Dále má třída svoje metody, které především slučují do vyšších celků operace nad jednotlivými atributy nebo poskytují propojení mezi více třídami. Také bychom zde mohli nalézt jednoduché metody pro zobrazení dialogových oken.

Spuštění aplikace obstarává třída `Main`, která však pouze vytvoří novou instanci třídy `Core` a spustí ji. V té jsou pak již provedeny všechny inicializace aplikace. Postupně je tedy provedeno načtení nastavení, které je nutné k vlastnímu běhu. Dále může být připojena databáze. Není-li nastavení či databáze dostupná, je vytvořen nový soubor s výchozím obsahem. V případě databáze se jedná pouze o zkopírování výchozího databázového souboru, jelikož k vytvoření nového databázového souboru jsou nutné speciální nástroje. Tímto postupem lze tento požadavek obejít. Následně se inicializují samotné součásti hlavní okna, jako hlavní nabídka, obsah okna, systémová ikona apod. Po úspěšném dokončení se zobrazí hlavní okno aplikace. V případě velkého objemu dat načítaného z databáze může dojít ke zpomalení spuštění, je to však dáno především rychlostí vykreslování součástí okna.

Veškerá chybová hlášení jsou opět řízena třídou `Core`. Jedná se jak o typické uživatelské chyby jako špatně zadané vstupy, dotazy potvrzující operaci, tak o systém výjimek. Většina výjimek programu je posílána této třídě, aby mohl být uživatel upozorněn, že došlo k chybě aplikace. Některé méně podstatné jsou však ošetřeny přímo v jednotlivých třídách.

V dalších podkapitolách jsou dále rozebrány některé důležité prvky aplikace a jejich provedení.

6.1.1 Systém nastavení

Nastavení má na starost třída `Settings`. Při spuštění aplikace jsou načtena nastavení z XML souboru, který má pevně dané umístění v rámci adresářové struktury aplikace. Není-li soubor dostupný, automaticky se vytvoří nový. V tom případě jsou načtena výchozí nastavení, která jsou předem dána. Podle nich také lze zkontrolovat, zda jsou všechna nastavení přítomna. Pokud ne, automaticky jsou doplněna. Tímto postupem je zaručeno, že nedojde k situaci, kdy není nějaké nezbytné nastavení definováno.

Jednotlivá nastavení jsou uložena jako mapa ve tvaru klíč : hodnota. Takto se k nim také přistupuje. Má to tu výhodu, že lze uložit jakákoliv nastavení a není nutné předem definovat jejich počet či název.

Do nastavení jsou ukládány všechny proměnné aplikace, jelikož je nechceme kombinovat s daty v databázi. Jedná se především o umístění a velikost jednotlivých oken aplikace, výchozí cesty pro databázi, dialogová okna, velikosti sloupců tabulky nebo poslední zobrazená data. Právě zobrazená data jsou však složitější na uložení a musíme tedy ukládat jak typ multimediálních dat, tak načtený adresář a navíc označená data v multimediální tabulce. Toto však výrazně zvyšuje uživatelský komfort.

Nastavení je ukládáno do souboru automaticky, a to jak při ukončení aplikace, tak automaticky každou minutu. Kdyby se soubor přepisoval při každé změně stavu aplikace, způsobovalo by to zbytečné zatížení aplikace. Plně dostačující je to po minutách, navíc je to řešeno ve vlastním vlákně a běh aplikace to tak téměř neovlivňuje.

6.1.2 Práce s databází

Připojení a manipulaci s databází řídí třída `Database`. Avšak hned první důležitá operace, a sice připojení, vyžaduje použít knihovnu, která to umožňuje. Jazyk Java sám o sobě totiž nepodporuje připojení ke zvolenému typu databáze. Naštěstí existuje knihovna SqliteJDBC, která toto umožňuje. Konkrétní použitá verze je 056. Knihovna je dostupná pod licencí BSD, která povoluje její volné šíření, užívání či úpravy. [16]

Třída samotná obsahuje metody pro práci s databází SQLite. Respektuje databázový model navržený v kapitole 4.1 a beze zbytku jej také implementuje. Jedná se také o jednu z nejdůležitějších a nejsložitějších tříd celé aplikace. Je zde třeba nejen správně načíst požadovaná data z databáze, ale také je efektivně promítnout do aplikace. K tomu je třída bezprostředně propojena s třídou `Core`, aby se veškerá data mohly co nejefektivněji přenést do grafického rozhraní. Vzhledem k tomu, že načítání či ukládání do databáze je činnost, která může při velkém množství dat trvat delší dobu, třída podporuje také zobrazování stavu činnosti v grafické podobě, tedy ve stavové liště.

Mezi nejdůležitější obsažené metody lze jmenovat načtení či uložení adresářů včetně jejich uložení či smazání a prakticky totožné operace pro multimediální soubory, kde je navíc přidáno i vyhledávání, které dle zadaných kritérií vytváří složitý SQL dotaz na databázi. Také je implementován import dat, který se liší od běžného přidání dat. Je především nutné namapovat původní identifikátory importovaných dat na nové tak, aby nedošlo ke kolizím či porušení referenční integrity.

6.1.3 Multimediální soubory

Multimediální soubory jsou udržovány v třídě Media. Ta je vždy znovu načtena při změně obsahu multimediální tabulky. Můžeme tedy říci, že tato třída reprezentuje data zobrazená právě v této tabulce. Je zde však nejen seznam zobrazených multimédií, ale také údaje o neuložených souborech, nastavení sloupců tabulky dle daného typu multimédií apod. Třída obsahuje jak metody pro manipulaci se soubory, tak jejich další přidávání, otevírání v externích aplikacích, listování adresáři apod. Podobnou třídou je pak `MediaDirImport`, která má však mírně odlišné chování a bylo nutné ji proto dodatečně přidat. Její jedinou funkcí je importovat do aplikace adresářovou strukturu datového úložiště, dále se s ní již nepracuje, proto obsahuje výrazně méně metod.

Každý soubor je reprezentován třídou `MediaFile`. Od té ovšem dědí třídy `MediaAudio`, `MediaVideo` a `MediaImage`. Takto jsou jednotlivé soubory rozděleny dle třídy, kde každá obsahuje jinou sadu speciálních atributů, nastavení apod. Jsou zde definovány atributy, které mají být extrahovány z multimediálních souborů, jejich názvy, které potom určují názvy sloupců a další.

6.1.4 Okna aplikace

Hlavní okno aplikace je sice tvořeno třídou `Core`, ale většinu obsahu zobrazuje třída `Window`. Ta obsahuje kromě hlavní nabídky veškerý obsah hlavního okna (viz kapitola 4.3). Můžeme ji tedy považovat za hlavní část grafického rozhraní. Kromě toho, že definuje rozložení grafických prvků aplikace, tak především obstarává manipulaci s obsahem. Jsou zde definovány metody pro práci se stromem [12], jeho naplnění a úpravy a taktéž pro tabulku multimédií [11]. Tyto operace jsou prováděny přímo touto třídou, avšak je propojena i s třídou `Core`. Veškeré zprávy o stisku tlačítek

jsou posílány právě této třídě a je tak zajištěno propojení i s ostatními. V okamžiku stisknutí tlačítka je tedy poslána zpráva pro `Core`, kde je vyhodnocena a jsou provedeny metody všech tříd, kterých se operace týká. Mimo hlavní okno má aplikace ještě několik vedlejších. Těmi jsou například okno vyhledávání, nastavení, úprav či nápovědy. Každé z oken je reprezentováno vlastní třídou a je otevíráno opět z třídy `Core`. V případě změny velikosti či přesunu je to zaznamenáno a uloženo.

6.2 Propojení s dalším softwarem

Vzhledem k tomu, že nemáme prostředky pro provedení všech operací, které by naše aplikace měla podporovat, je nutné použít software třetích stran. Jedná se buď o knihovny nebo o celé další aplikace. Některé z nich byly již v práci zmíněny, jiné si popíšeme právě v této kapitole. Všechny jsou však dostupné pod vhodnou licencí a je tedy možné je volně používat.

6.2.1 Načítání a ukládání XML souborů

Soubory XML jsou v aplikaci používány jak pro ukládání nastavení, tak pro export a import dat. To již bylo zmíněno v návrhu (kapitola 4.2). Také je však nutné ho analyzovat při extrakci informací, které jsou výstupem programu MediaInfo (kapitola 5.3). Tyto XML vstupy jsou analyzovány zvlášť v každé příslušné třídě, kde se s nimi pracuje. Důležité však je, jakým způsobem je toto prováděno. Za nejlepší řešení autor považoval vybrat knihovnu DOM4J verze 1.6.1 [13], která je dostupná pod BSD licencí a můžeme ji tedy použít. Tato knihovna umožňuje procházet XML dokumenty různými metodami, například v aplikaci použité nalezení elementu dle jeho názvu. Také podporuje vytváření a ukládání XML dokumentů.

6.2.2 Použití programu MediaInfo

Nástroj MediaInfo byl již zmíněn v kapitole 5.3. Je dostupný pro všechny platformy, avšak s jistými rozdíly. Pro operační systém Windows existuje již zkompilovaná verze, kterou lze snadno přímo spustit a používat. Oproti tomu pro platformu Linux je nutné stáhnout si instalační balíček a nástroj zvlášť nainstalovat. To by však znamenalo různé chování na různých platformách, což může být nežádoucí. Naštěstí existují i zdrojové kódy, které si můžeme sami zkompilovat. Toto je pak potřeba řádně otestovat na platformách, které bychom chtěli prohlásit za podporované. Je to však zjednodušení především při spouštění aplikace, kdy uživatel není obtěžován dodatečnými instalacemi. Kromě toho si pak uživatel může sám dodat vyšší verzi, která je volně dostupná ke stažení. Nástroj je totiž neustále vyvíjen a uživatel by tak měl mít možnost používat nejvyšší dostupnou verzi.

6.2.3 Zobrazení nápovědy

Pro nápovědu a její zobrazení byla zvolena standardní Java knihovna JHALL, která je dostupná ze systému JavaHelp [14]. Jedná se o nástroj, pomocí kterého lze zobrazit nápovědu dle návrhu (kapitola 4.3.10). K tomuto stačí pouze vytvořit si vlastní jar soubor, do kterého jsou vloženy dle specifikace konfigurační XML soubory a dále HTML soubory s vlastním textem nápovědy.

7 Ovládání a funkce aplikace

7.1 Ovládání

Ovládání aplikace bylo navrženo a vytvořeno s ohledem na jednoduchost, intuitivnost a uživatelské pohodlí. Celou aplikaci až na nezbytné vstupy z klávesnice lze ovládat pouze myší, což je u grafického uživatelského rozhraní žádoucí. Jelikož toto rozhraní obsahuje velké množství ovládacích prvků, především tlačítek, velmi často postačí pouze jedno kliknutí k provedení požadované operace. Veškerá tlačítka jsou jak graficky znázorněna příslušnou ikonou, tak obsahují popis, který se zobrazí po najetí myší. To umožňuje dobrou orientaci a uživatel okamžitě ví, jaká operace bude provedena.

Mimo běžné ovládání pomocí myši poskytuje aplikace také některé klávesové zkratky, které mohou pokročilejšímu uživateli usnadnit a urychlit práci s knihovnou. Tyto zkratky vychází ze zavedeného systému klávesových zkratek, které bývají běžně používány v aplikacích a dále přidává několik dalších. Využity jsou také další klávesy, které jsou uživatelsky zažité. Veškeré použité klávesové zkratky jsou potom shrnuty v tabulce 7.1.

Klávesa či zkratka	**Význam**
CTRL + F	zobrazí vyhledávací okno
CTRL + S	uloží všechna data a nastavení
CTRL + A	vybere všechny položky v tabulce multimédií
CTRL + E	provede export zatržených dat
CTRL + I	provede import dat ze zvoleného souboru
CTRL + Q	ukončí celou aplikaci
CTRL + J	přepne typ multimédií na audio
CTRL + K	přepne typ multimédií na video
CTRL + L	přepne typ multimédií na obrázky
CTRL + N	zobrazí okno s nastavením aplikace
CTRL + Z	zruší zaškrtnutí všech položek
F1	zobrazí okno nápovědy
DELETE	odstraní položku z tabulky multimédií či z adresářového stromu podle toho, který je zrovna zaměřen (focus)
ESC	uzavře aktuální vedlejší okno aplikace

Tabulka 7.1 – klávesové zkratky a jiné funkční klávesy

7.1.1 Výběr z hlavní nabídky

Hlavní nabídka je naprosto běžným typem menu, kdy se v horní části grafického rozhraní nachází lišta (viz kapitola 4.3.1), která se po najetí myši rozbalí. Kliknutím na požadovanou položku se potom provede daná operace. Tato nabídka má však kromě jednoduchosti ještě další výhodu, a sice možnost ovládat ji také klávesnicí. Každá položka má jednak definovanou klávesovou zkratku, která je přímo u ní zobrazena, a jednak také podtržené písmeno. Je-li stisknuta tato klávesa v kombinaci s klávesou alt, je vyvolána právě tato nabídka. Uživatelům, kteří jsou zvyklí používat více klávesnici může toto ušetřit čas.

7.1.2 Přepínání typu multimédií a vyhledávání

Mezi typy multimédií, tedy audio, video a obrázky, lze přepínat jednoduchým kliknutím na velká tlačítka znázorňující jednotlivé typy. Stejným způsobem lze zobrazit i vyhledávání, které je s těmito tlačítky úzce spojeno. Každé z tlačítek má zároveň definovánu klávesovou zkratku.

Vyhledávání na rozdíl od ostatních tlačítek nepřepne typ multimédií, ale zobrazí nové vyhledávací okno. Vyhledávání se provádí v označeném adresáři a jeho podadresářích. Není-li žádný vybraný, automaticky se vyhledává ve všech. Samotná operace spočívá pouze v zadání hledaného výrazu a označení atributů, ve kterých má být vyhledávání provedeno. Po potvrzení se požadovaná data načtou přímo do tabulky a uživatel je již může běžně prohlížet.

7.1.3 Manipulace s adresářovým stromem

Adresářový strom je jednou z hlavních částí grafického rozhraní knihovny. Jeho ovládání lze rozdělit na dvě části – pomocí myši či pomocí tlačítek umístěných pod ním.

Pomocí myši se provádí velmi jednoduchá operace označení adresáře ve stromu. Tím dojde k otevření tohoto adresáře a načtení jeho obsahu do tabulky multimédií. Také můžeme opětovným kliknutím na již označenou položku vyvolat změnu jejího jména. Po přepsání této hodnoty ji uložíme stiskem klávesy Enter.

Tlačítka umístěná pod stromem pak fungují jak k manipulaci s jeho položkami, tak k přidávání či odebírání. Vždy se manipuluje s označenou položkou (adresářem). Pomocí tlačítek tedy lze do adresáře vložit soubory přes dialogové okno, vložit obsah celých složek, kdy vybereme libovolný počet složek, jejichž obsah se opět vloží do adresáře. Dále potom lze importovat celou adresářovou strukturu pevného disku. To nám do stromu načte všechny podsložky zvolené složky i s jejich obsahem. Toto je zvláště vhodné, pokud chceme sledovat změny obsahu těchto složek, jelikož je aplikace bude průběžně kontrolovat. Dále lze vytvořit složku novou či zvolenou složku odebrat i s jejími podsložkami. Při odebírání však je třeba dát pozor, které podsložky budou spolu se zvolenou odebírány, mohlo by totiž dojít k nechtěné ztrátě dat. Aby však nedošlo k nějakým problémům či chybám v chování aplikace, nelze takto odebrat kořenovou složku stromu. Tím je zajištěna větší bezpečnost dat ve stromech.

7.1.4 Zobrazování a manipulace s multimediálními daty

Veškerá multimediální data jsou zobrazena ve formě tabulky. Chceme-li je zobrazit, stačí označit kliknutím příslušnou položku v adresářovém stromě a data se automaticky načtou. Nyní můžeme s těmito daty libovolně pracovat. Podobně jako ve stromě jsou změny a operace prováděny vždy s označenými položkami – v tomto případě řádky tabulky. Vpravo od tabulky se v grafickém rozhraní nachází skupina tlačítek, které umožňují provést operace jako odstranění, změnu položky, jejich zatržení či nastavení oblíbenosti. Nacházejí se zde však také tlačítka, která mají odlišnou funkci od těch ostatních. Tyto provádějí operaci nad všemi položkami a nikoliv jen nad těmi označenými. Jedná se o hromadné spuštění všech zatržených (nikoli označených) položek, tlačítko indikující, zda jsou zobrazeny všechny položky či jen oblíbené a také tlačítko provádějící obnovení celé složky včetně informací o multimediálních datech.

Některé informace lze měnit přímo kliknutím na příslušnou buňku tabulky, konkrétně se jedná o označení oblíbené položky (hvězdička) a zatržení. Zatržení položky lze využít pro různé operace – můžeme například otevřít všechny zatržené položky v externí aplikaci. Tak lze docílit například spuštění většího množství souborů naráz v aplikaci se seznamem souborů a vytvořit si tak vlastní playlist. Další možností je zatržené položky exportovat.

Pokud chceme otevřít v externí aplikaci pouze jeden konkrétní soubor, není nutné jej zatrhnout a následně otevřít tlačítkem, stačí pouze dvakrát kliknout na položku (dvojklik). Aplikace se okamžitě spustí.

Samozřejmostí je také možnost přesunovat jednotlivé soubory do jiného adresáře. To lze naprosto jednoduchým přesunem tak, jak to funguje běžně v operačních systémech. Označené položky stačí uchopit a přetáhnout je do požadované složky.

7.1.5 Nastavení aplikace

Hlavním účelem nastavení programu je možnost uživatelsky definovat podporované formáty a externí aplikace, tedy ty, ve kterých se mají jednotlivé multimediální soubory otevírat. Nastavení je, stejně jako celá aplikace, rozděleno na audio, video a obrázky. Pro každý typ multimédia lze definovat formáty, které budou aplikací podporovány a potom tedy i načítány. Předem jsou však již nějaké pevně dány, a proto nelze zvolit jakékoliv. Ke každému formátu potom po označení lze nastavit externí aplikaci.

7.2 Funkce

Některé z funkcí aplikace jsou již popsány v kapitole 7.1, kde je rozebráno její ovládání. Jelikož jsou zde však popsány pouze velmi zběžně, podíváme se na některé detailněji. Obzvláště pak na ty, které jsou složitější a vyžadují správné pochopení.

7.2.1 Přidávání a odebírání multimédií

Aplikace poskytuje několik možností, jak do ní přidat multimediální soubory. Každá je potom v grafickém rozhraní zastoupena zvláštním tlačítkem, aby si mohl uživatel snadno vybrat. Odebírání souborů lze potom provést také různými způsoby.

Jednou z možností, jak do aplikace přidat soubory je jejich jednoduché vložení. Po kliknutí na příslušné tlačítko je vyvoláno dialogové okno pro výběr souborů. Můžeme zvolit libovolný počet z podporovaných souborů a ty jsou po potvrzení okamžitě vloženy.

Pro usnadnění vložení většího počtu souborů má uživatel možnost přidat obsah celé složky. To je provedeno tak, že složky zvolené pomocí dialogového okna jsou prohledány včetně jejich podsložek a veškeré multimediální soubory, které odpovídají podporovaným formátům, jsou potom přidány do knihovny. Avšak všechny jsou nahrány pouze do jednoho zvoleného adresáře z adresářového stromu.

Vzhledem k tomu, že aplikace má zajišťovat také kontrolu souborů na datovém úložišti, je další možností, jak přidat soubory, import adresářové struktury ze zvoleného úložiště. Zvolená složka je potom načtena do aplikace včetně podsložek a pro každou je vytvořena zvláštní položka v adresářovém stromu. Do té jsou umístěny odpovídající multimediální soubory. Tímto dojde ke svázání složky z datového úložiště s adresářem ve stromu a lze sledovat její obsah. Pokud dojde ke změně jejího obsahu, je tato situace reflektována při manuálním obnovení informací například přidáním nových souborů. Naopak chybí-li soubor (není fyzicky dostupný), odpovídající položka v multimediální tabulce se stane neaktivní. To však platí pro veškeré soubory, jelikož lze vždy ověřit, zda je soubor dostupný – není tedy vždy nutné svázat adresář aplikace s fyzickou složkou.

Odebrat soubory z knihovny lze přímo z multimediální tabulky, tedy například po jednom. Jinou možností však je odstranit celý adresář ze stromu. To však vyžaduje opatrnost, jelikož jsou tímto krokem odebrány všechny soubory zde umístěné. Krom toho je toto provedeno i pro veškeré podsložky.

7.2.2 Export a import dat

Export či import dat je prováděn pomocí XML souborů (viz kapitola 4.2). Po zaškrtnutí položek, které chceme exportovat je vyvoláno dialogové okno, kde zvolíme XML soubor pro export. V případě, že soubor neexistuje, je automaticky vytvořen. Následně jsou do něj uložena všechna data, avšak nejen informace o samotných multimediálních souborech, ale také adresáře stromu, ve kterých jsou uloženy. Tento soubor pak lze jednoduše importovat s tím, že importované adresáře jsou vloženy do právě zvoleného adresáře ve stromu. Jelikož nelze zajistit po importu stejnou adresářovou strukturu na fyzickém úložišti, nelze je navzájem propojit. Při obnovení dat tedy nedojde k promítnutí případných změn ve složce na datovém úložišti. Je zajištěna pouze dostupnost souborů na úložišti a jejich případné označení jako nedostupné.

7.2.3 Oblíbené položky

Tato funkce byla navržena především kvůli usnadnění správy velkého množství multimediálních souborů. Uživatel může některé z nich využívat výrazně častěji něž jiné a je proto pohodlné mít možnost ty ostatní nějak odfiltrovat. Postup, jak nastavit či zrušit oblíbené položky byl již popsán v kapitole 7.1.4. Aplikaci lze potom jedním tlačítkem přepnout na zobrazování pouze oblíbených položek a usnadnit tak jejich snadné vyhledání.

7.2.4 Nastavení podporovaných formátů a aplikací

V kapitole 7.1.5 bylo již zmíněno nastavení aplikace, především tedy jejích formátů a externích aplikací, ve kterých mají být soubory otvírány. Toto je však velmi důležité popsat detailněji, jelikož se jedná o poměrně složitý systém.

Nastavit externí aplikaci pro jednotlivé formáty můžeme několika způsoby – buď lze zadat přímo název aplikace, přesněji tedy ve tvaru příkazu, kterým se aplikace spouští (ve Windows například wmplayer pro Windows media player), nebo absolutní cestu k této aplikaci. U samostatného příkazu může někdy dojít k problému, kdy není operačním systémem rozpoznán a je tedy třeba zadat absolutní cestu k aplikaci. V obou případech však můžeme k tomuto zápisu přidat i argumenty, se kterými bude aplikace spuštěna. Není tedy problém spouštět aplikace, které to umožňují (například WinAmp), s argumentem, který například přidá soubor do playlistu, místo aby byla spuštěna nová instance programu. Takto lze snadno generovat playlisty či si jinak uzpůsobit aplikace ke svému vlastnímu užitku.

8 Testování aplikace a její výsledky

8.1 Testování

Aplikace byla již od začátku navrhována jako multiplatformní. Z tohoto důvodu byl zvolen i programovací jazyk (viz kapitola 5.1). Všechny součásti aplikace a další externí nástroje a knihovny byly rovněž zvoleny tak, aby tuto podmínku splňovaly.

Aplikace samotná byla vyvíjena na operačním sytému Linux Ubuntu 10.10. V tomto systému byla stoprocentně funkční a nedošlo k žádným nepředvídaným chybám či pádům aplikace či některé z jejích součástí. Protože však OS Linux existuje ve velkém množství distribucí, bylo nutné otestovat alespoň některé z těch nejpoužívanějších. Proto byla aplikace opakovaně spouštěna a testována také na distribucích Debian 6 a openSUSE 11.4. Ani zde nedošlo k žádným problémů a lze tedy prohlásit, že pro operační systém Linux poskytuje aplikace dostačující podporu. Na ostatních distribucích by nemělo docházet k větším problémům, jelikož často vycházejí právě z těchto distribucí.

Jelikož nejpoužívanějším operačním systémem je Windows, je třeba, aby právě zde byl běh aplikace bezproblémový. To bylo vyzkoušeno na jeho nejnovější verzi, tedy Windows 7. Během práce na tomto OS opět nedošlo k žádným pádům a veškeré součásti programu pracovaly tak, jak mají.

Na všech systémech byl běh aplikace rychlý a práce s ní pohodlná. Nebyly zjištěny žádné nedostatky či obtíže, které by její používání mohly znemožnit.

Mimo testu operačních systému bylo také nutné vyzkoušet veškeré funkce aplikace, jejich chování za daných situací a především toto provést v co největším počtu. Již při samotném vývoji byly jednotlivé části testovány v různých situacích a problémy a chyby byly vždy odstraněny. Po dokončení bylo testování některých kritických situací provedeno znovu a k chybám již nedošlo. Přesto nikdy nelze říci, že je výsledek stoprocentně bezchybný. Autor se snažil veškeré chyby eliminovat na minimum, ale některé situace při testování nemusely nastat. Výjimečně tak může dojít k chybě například důsledkem odlišné konfigurace počítače či některých programů. Nelze to však považovat za neúspěch aplikace, jelikož každý produkt má vždy nějaké chyby, které je nutné časem odladit.

8.2 Výsledky práce

Porovnáme-li zadání práce a její cíle (viz kapitola 2.1) s návrhem (kapitola 4) a následně s výslednou aplikací, která je popsána především v kapitolách 6 a 7, můžeme konstatovat, že z podstatné části se povedlo vytvořit to, co jsme si na začátku předsevzali. Byla vytvořena aplikace, která byla řádně popsána právě návrhem a následně podle něj implementována. Testy (kapitola 8.1) prokázali, že ji lze považovat za funkční při dodržení multiplatformnosti a dalších požadavků.

Z hlediska uživatelského pohodlí poskytuje program vhodné grafické rozhraní, které usnadňuje jeho používání a práci s ním. K ovladatelnosti přispívají např. klávesové zkratky (viz kapitola 7.1) a přehledné zobrazení ve formě tabulky a adresářového stromu. Dostatečné množství tlačítek potom urychluje některé činnosti.

Hlavním faktorem úspěchu aplikace však jsou její možnosti. Záleží už ale především na uživatelích, jak toto zhodnotí. Bylo uděláno maximum pro to, aby program poskytoval co nejvíce funkcí a především zobrazoval informace, které jsou hodnotné a užitečné. To bylo také hlavním úkolem. Jejich zobrazení je přehledné a uživateli umožňuje snadnou manipulaci. V kombinaci s možností uložit veškerá multimediální data na jednom místě, by mohl být program vhodným pomocníkem každodenních činností.

9 Možnosti rozšíření

Nyní, když máme již dokončenou a otestovanou aplikaci, máme dostatek poznatků o jejich možnostech a funkcích, můžeme se pokusit navrhnout, jaká rozšíření a vylepšení by bylo možné do budoucna udělat. Tento krok je logickým vyústěním dlouhodobé práce a vzhledem k tomu, že očekáváme její nasazení do provozu a běžné činnosti, dá se považovat téměř i za nutnost. Veškeré počítačové vybavení se neustále vyvíjí a pokud chceme držet krok s dobou a jejími možnostmi, je třeba si stanovit jisté cíle, ke kterým bychom se chtěli dále přibližovat.

V této kapitole se pokusíme shrnout pouze několik důležitých bodů, kterých by bylo možné dosáhnout. Samozřejmě nikdy nelze obsáhnout veškeré možnosti, zaměříme se proto na ty nejdůležitější a nejpravděpodobnější.

9.1 Multimediální přehrávač

Multimediální knihovna sama o sobě již shromažďuje velké množství souborů a informací o nich. Je tedy nasnadě rozšířit ji i o další velmi užitečnou funkci, a sice možnost přehrávání či zobrazování medií. Přehrávání je myšleno pro audio a video soubory, zobrazování pak pro obrázky, fotky a jiné. S tím úzce souvisí tvorba playlistů, tedy seznamů pro přehrávání, dále fotoalb, možnost jejich ukládání apod.

Tento krok má však jistá úskalí, nad kterými je třeba se pozastavit. Pravděpodobně největším problémem zde bude samotné přehrávání, jelikož je nutné veškerá multimediální data správně dekódovat, tedy použít kodeky [15]. Bylo by možné si tyto nástroje buď vytvořit, což je však velmi pracné, nebo použít již stávající nástroje. S tím ovšem přichází druhý problém, což je požadavek multiplatformnosti u současné aplikace. Každá platforma však používá jiný systém kodeků a bylo by proto nutné vytvořit pro každou platformu jinou verzi. To by však snížilo samotnou přenositelnost aplikace.

Toto rozšíření by ve výsledku bylo velice zajímavé a posunulo by aplikaci o další třídu výše, ale za poměrně vysokou cenu. Je proto na zvážení, kterou cestou by bylo nejlepší se vydat.

9.2 Editor obrázků

Jedním z multimédií, které aplikace ukládá, jsou právě obrázky. A s těmi lze poměrně snadno a efektně pracovat. Právě tímto směrem by bylo možné rozšířit multimediální knihovnu. Obrázky by bylo možné nejen otevřít přímo v aplikaci bez nutnosti používat nějakou externí, ale také přidat sadu operací, jenž by umožnily uživateli je upravovat a ukládat.

Operací, které mohou obrázky upravovat, je velké množství. Mezi ty základní, které by bylo možné aplikovat, lze jmenovat zmenšení či zvětšení, oříznutí nebo otočení. Dále bychom mohli přidat další složitější funkce, jako různé úpravy barevnosti, světlosti a další filtry, typicky černobílý, sépiový či inverzi barev.

Toto rozšíření by bylo zajímavé především z hlediska fotoalba, kdy potřebujeme jednoduchou úpravu fotografií, často například jen zmenšení, a nechceme zbytečně využívat jiných nástrojů. Vhodným doplněním by pak byla možnost aplikovat změny na celou složku či vybrané soubory.

9.3 Inteligentní rozhraní

Aplikace je vytvořena s tím, že bude používána velmi často, můžeme říci prakticky každodenně. Očekává se tedy od ní jisté uživatelské pohodlí. K tomuto pohodlí by mohla velkou měrou přispět jistá forma zabudované inteligence. Pod tímto si můžeme představit takové chování aplikace, které bude uživateli nabízet ta data, která nejčastěji využívá či prohlíží.

Ve skutečnosti by mohlo jít například o takové funkce, jako výchozí nabídka multimédií. Ta by se zobrazila například po spuštění programu a uživateli by nabídla ta data, která nejčastěji zobrazuje či spouští. K těmto datům by mohly být rovnou i přiřazeny nejčastěji prováděné operace. Uživatel by si pak mohl sám definovat chování aplikace a přizpůsobit si ji tak co nejvíce pro svoji vlastní potřebu.

9.4 Vyhledání informací

Multimediální knihovna shromažďuje pouze informace, které extrahuje přímo ze souborů. Avšak v době, kdy je již internet rozšířen tak, že k němu má přístup téměř každý, se nabízí možnost získávat další informace právě odsud. Bylo by zbytečné připravovat uživatele o tak velké množství často hodnotných informací. Aplikaci by tak šlo rozšířit o možnost vyhledávat je na různých specializovaných server, které poskytují informace například o filmech či seriálech. Můžeme zmínit např. servery csfd.cz či imdb.com.

Uživatel by měl možnost nastavit si, které z aplikací podporovaných serverů chce využívat a knihovna by se potom vždy připojila k serveru a extrahovala data buď přes speciální rozhraní nebo třeba přímo analyzováním HTML kódu stránky. V aplikaci by se tak mohlo zobrazit například hodnocení jednotlivých filmů z těchto serverů či oficiální text filmu. Také by mohla uživateli pouze nabídnout odkaz, který by otevřel webovou stránku s příslušným obsahem.

Další možností by mohlo být vyhledání titulků k vybranému filmu dle jeho atributů, tedy například podle jména, velikosti souboru či délky stopy. Titulky by potom aplikace automaticky uložila a při přehrávání je i přímo připojila.

Možností, jak propojit aplikaci s různými internetovými službami, je opravdu velké množství, proto je zde zmíněno jen několik málo situací. Z uživatelského hlediska se však jedná o velmi zajímavou funkci, která by určitě našla své uplatnění.

10 Závěr

Výsledkem našeho snažení v průběhu celé práce je nová a funkční aplikace, která poskytuje funkce, které jsou v souladu s jejím zadáním a splňuje vlastnosti, které bychom od multimediální knihovny očekávali. Během práce jsme se také seznámili s řadou nových informací a postupů, které lze dále uplatnit.

Na počátku se nám podařilo získat důležité informace a specifikace multimédií, bez kterých bychom se jen těžko obešli a byly klíčové pro další vývoj aplikace. Během návrhu programu bylo dodržováno tradičních postupů a jeho výsledkem pak byly různé modely, ať již objektový či databázový, dále také struktury pro ukládání různých dat a v neposlední řadě i propracované grafické rozhraní. Na toto bylo navázáno především výběrem správných prostředků, které byly použity při vývoji. Všechny tyto prostředky splnily naše požadavky a kritéria a posunuli naše možnosti o kus dále. Nejpracnější a nejvíce problémová implementace pak především prověřila programátorské schopnosti a využití různých technik objektově orientovaného programování. V této části práce se také vyskytlo nejvíce nečekaných problémů, od zásahů do objektového modelu, kdy bylo třeba přehodnotit některé přístupy a metody, přes komplikace s grafickým rozhraní a nutnost naučit se nové možnosti programovacího jazyka, až po ladění některých součástí na různých operačních systémech. Přes toto všechno se veškeré problémy podařilo vyřešit a aplikace tak mohla být úspěšně dokončena a následně podrobena testům funkčnosti.

Po dokončení prací na aplikaci samotné přišla otázka – co bude dál? Samozřejmě by byla škoda se spokojit s momentálním stavem, a proto bylo nutné se zamyslet, jak aplikaci posunout ještě dále, vylepšit ji, dodat jí nové funkce. Jelikož jsme však měli možnost sledovat vývoj práce od počátku až po dokončení, nápadů během této doby bylo více než dost. Některé z nich byly nastíněny, některé přijdou ještě časem.

Nyní již je tedy především na uživatelích samotných, jak se k využití aplikace a celé práci postaví. Autor pro to v této fázi udělal maximum, což mu také přineslo velké množství zkušeností, které určitě ještě dále zužitkuje. Možná se mu s touto prací dále podaří nějaké úspěchy a neskončí tak pouze u této fáze vývoje. Vhodnou příležitostí pro pokračování na tomto projektu by pak jistě mohla být například magisterská práce, kde by určitě našlo uplatnění některé z navrhovaných rozšíření.

Literatura

[1] MediaInfo [online]. 2011 [cit. 10.5.2011]. Dostupný z WWW: <http://mediainfo.sourceforge.net>

[2] Wikipedia, ID3 tag [online]. 2011 [cit. 10.5.2011]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/ID3_tag>

[3] Extensible Markup Language (XML) 1.0 (Fifth Edition) [online]. 2008 [cit. 10.5.2011]. Dostupný z WWW: <http://www.w3.org/TR/2008/REC-xml-20081126>

[4] ZEMČÍK, Petr. *Tvorba uživatelských rozhraní*. Studijní opora, Vysoké učení technické v Brně, 2006

[5] KOČÍ, Radek; KŘENA, Bohuslav. *Úvod do softwarového inženýrství*, s. 33 – 40. Studijní opora, Vysoké učení technické v Brně, 2007.

[6] Java SE [online]. 2010 [cit. 17.4.2011]. Dostupný z WWW: <http://download.oracle.com/javase>

[7] HEROUT, Pavel. *Učebnice jazyka Java.*. 4. vydání. České Budějovice : Kopp, 2008. 381 s. ISBN 978-80-7232-355-5.

[8] KISZKA, Bogdan. *1001 tipů a triků pro jazyk Java*. Brno : Computer Press, 2009. 544 s. ISBN 978-80-251-2467-3.

[9] HEROUT, Pavel. *Grafické uživatelské prostředí a čeština*. České Budějovice : Kopp, 2007. 320 s. ISBN 80-7232-328-9.

[10] SQLite [online]. 2010 [cit. 28.4.2011]. Dostupný z WWW: <http://www.sqlite.org>

[11] The Java tutorials, How to use tables [online]. 2011 [cit. 20.4.2011]. Dostupný z WWW: <http://download.oracle.com/javase/tutorial/uiswing/components/table.html>

[12] The Java tutorials, How to use trees [online]. 2011 [cit. 20.4.2011]. Dostupný z WWW: <http://download.oracle.com/javase/tutorial/uiswing/components/tree.html>

[13] Java knihovna DOM4J [online]. 2011 [cit. 10.5.2011]. Dostupný z WWW: <http://www.dom4j.org/dom4j-1.6.1>

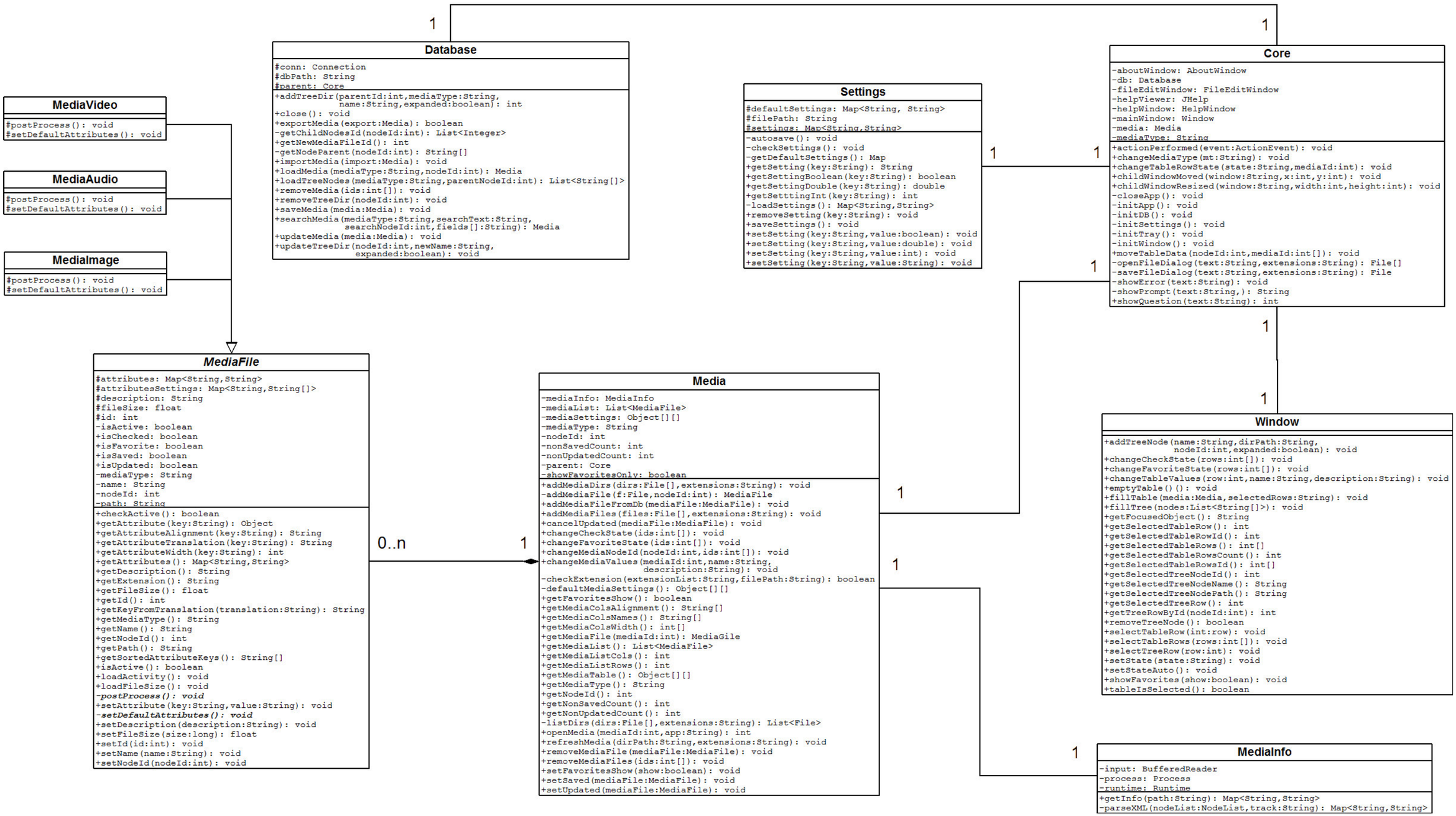
[14] JavaHelp system [online]. 2011 [cit. 10.5.2011]. Dostupný z WWW: <http://javahelp.java.net>

[15] Wikipedia, Kodek [online]. 2011 [cit. 10.5.2011]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Kodek>

[16] SQLiteJDBC [online]. 2010 [cit. 28.4.2011]. Dostupný z WWW: <http://www.zentus.com/sqlitejdbc/>

Seznam příloh

Příloha 1. E-R diagram objektového modelu

Příloha 2. CD obsahující zdrojové kódy a spustitelnou verzi aplikace

Příloha 1. E-R diagram objektového modelu