

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Informační systém**  
**Audiovizuálního centra VŠB-TUO**

**2005**

**David Pilarš**



# Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.

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

V Ostravě dne 2. května 2005

.....

David Pilarš



# Poděkování

V první řadě chci na tomto místě poděkovat Doc. RNDr. Janě Šarmanové, CSc. za odbornou pomoc, věcné připomínky a vedení při tvorbě této diplomové práce.

A za neustávající morální, psychickou a fyzickou podporu a trpělivost děkuji své rodině a milující přítelkyni.



# Abstrakt

Tato diplomová práce se zabývá vytvořením on-line informačního systému pro Audiovizuální centrum VŠB-TUO. Cílem práce bylo navrhnout novou podobu informačního systému, která by umožňovala jeho pohodlné využívání uživateli, poskytovala informace o videokazetách, pořadech, objednávkách a výpůjčkách. Celý projekt je napsán za použití programovacích a skriptovacích jazyků PHP a JavaScript a značkovacího jazyka XHTML s použitím šablon. Informační systém poběží na libovolném web serveru, který podporuje PHP, a nad databází MySQL.

# Abstract

This dissertation's concern is the development of the on-line information system for the Audiovisual centre of VŠB-TUO. The goal of the dissertation was to design a new form of information system that would enable comfortable use by users; it would also provide information about videocassettes, video programs, and orders and borrows. The whole project is implemented with programming and scripting languages such as PHP and JavaScript and marking language XHTML using templates. Information system can run on any web server supporting PHP and MySQL database.





# Klíčová slova

Informační systém, internetová aplikace, intranetová aplikace, databáze, web server, SQL, MySQL, PHP, JavaScript, HTML, XHTML, Audiovizuální centrum, videokazety, videopořady

# Keywords

Information system, internet application, intranet application, database, web server, SQL, MySQL, PHP, JavaScript, HTML, XHTML, Audiovisual centre, videocassettes, video programs



# Obsah

<b>Obsah</b>	<b>13</b>
<b>1. Úvod</b>	<b>15</b>
<b>2. Specifikace zadání</b>	<b>17</b>
2.1 Neformální popis .....	17
2.2 Funkční požadavky .....	17
2.2.1 Důvod vytvoření systému.....	17
2.2.2 Kdo bude se systémem pracovat.....	18
2.2.3 Požadované funkce systému.....	18
2.2.4 Vstupy a výstupy systému .....	19
2.3 Nefunkční požadavky .....	19
2.4 Databázová terminologie .....	19
<b>3. Analýza projektu</b>	<b>21</b>
3.1 Datová analýza .....	21
3.1.1 Relační schéma databáze.....	21
3.2 Funkční analýza.....	31
3.2.1 Kontextový diagram.....	31
3.2.2 Data-flow diagram .....	32
3.2.3 Minispecifikace funkcí.....	34
<b>4. Návrh implementace</b>	<b>49</b>
4.1 Výběr technologií pro implementaci .....	49
4.2 Autentizace uživatele.....	50
4.3 Rozdělení systému na moduly.....	50
4.3.1 Struktura adresářů systému .....	50
4.3.2 Předávání a ověřování proměnných.....	51
4.3.3 Práce s databází .....	51
<b>5. Implementace</b>	<b>53</b>
5.1 Popis implementace.....	53
5.2 Popis typických formulářů .....	54
5.3 Instalace .....	58
5.3.1 Instalace aplikace.....	58
5.3.2 Konverze dat a naplnění databáze.....	59
5.4 Testování a provoz IS .....	59
<b>6. Použité internetové technologie</b>	<b>61</b>
6.1 World Wide Web.....	61
6.2 Web server .....	62

6.3 Databázové systémy .....	62
6.3.1 MySQL Server.....	63
6.3.2 Jazyk SQL .....	64
6.4 Technologie pro vývoj internetových aplikací.....	65
6.4.1 PHP .....	65
6.4.2 JavaScript.....	65
6.4.3 HTML, XHTML, XML .....	66
6.4.4 CSS .....	66
6.4.5 Šablony .....	67
<b>7. Závěr</b>	<b>69</b>
<b>Literatura</b>	<b>71</b>
<b>Příloha A – DFD diagramy</b>	<b>73</b>
<b>Příloha B – obecný popis souborů</b>	<b>101</b>
<b>Přiložené CD</b>	<b>105</b>
Text diplomové práce v MS Word	
Uživatelské příručka v MS Word	
Programátorská dokumentace v MS Word	
Zdrojové texty aplikace	
Programové prostředky	

# 1. Úvod

Informační systém Audiovizuálního centra VŠB-TUO (dále jen IS) vznikl na podnět tohoto centra. Důvodem vzniku byla potřeba převést současnou podobu IS ze „zastaralého“ a nevyhovujícího „DOSovského“ prostředí do nového, tj. prostředí intranetu či internetu, a vytvoření tak jednoduchého přístupu k informacím o nabízených videokazetách a videopořadech a možnosti jejich objednávání.

Výsledný návrh a implementace informačního systému je určen pro potřeby Audiovizuálního centra VŠB-TUO a je naprogramován k potřebám tohoto centra.

Výsledná aplikace bude přístupná formou webových stránek z prostředí intranetu či internetu.



## **2. Specifikace zadání**

### **2.1 Neformální popis**

Současná podoba informačního systému je implementována v dnes již nevyhovujícím prostředí DOS. Nabízí pouze jedinouživatelský přístup k evidovaným datům a data jsou přístupná pouze z počítače, na kterém je IS nainstalován. Tato nemožnost přístupu k informacím o nabízených datech odkudkoliv je hlavním motorem k vytvoření nové podoby informačního systému a umožnění tak všem jeho uživatelům získávat informace o pořadech.

Informační systém samozřejmě obsahuje funkce, které nesmí být v nové podobě opomenuty. Těmito funkcemi je myšleno vedení evidence existujících videokazet, pořadů a objednávek promítání pořadů v místnosti k tomu určené.

Nová podoba informačního systému bude nabízet víceuživatelský přístup k datům. Jednotliví uživatelé budou mít různá práva, z toho také vyplývá přístup k různým funkcím systému. V IS tedy přibude možnost objednání si videokazety jednotlivými uživateli, možnost spravovat informace o své osobě a možnost listování v datech, které IS obsahuje. Na straně správce systému zůstane možnost vést evidenci videokazet, pořadů a přibude nutnost vést evidenci uživatelů a jejich výpůjček. Obě uživatelské strany IS budou mít také možnost listovat v historii vypůjčování videokazet nebo jejich projekcí v promítací místnosti. IS bude automatizován na straně serveru tak, že se bude starat o upozorňování uživatele na nutnost vrácení vypůjčených kazet, vyzvednutí objednávky apod. Upozorňování bude probíhat elektronickou poštou pomocí e-mailových zpráv.

### **2.2 Funkční požadavky**

#### **2.2.1 Důvod vytvoření systému**

Hlavním důvodem vytvoření systému, jak již bylo zmíněno, je nemožnost přístupu k datům odkudkoliv a požadavek víceuživatelského přístupu.

Přechodem do prostředí intranetu (internetu) se zjednoduší přístup zaměstnanců (studentů) VŠB-TUO k nabízeným videokazetám. Zjednoduší se také možnost jejich pohodlného objednání k zapůjčení nebo k projekci v promítací místnosti. Toto pohodlné objednávání je jednou z hlavních výhod přechodu k novému systému. Dalšími výhodami je možnost sledovat

objednané, vypůjčené a volné videokazety, objednávky, výpůjčky a upomínky a v neposlední řadě i zobrazení historie výpůjček a promítání.

### 2.2.2 Kdo bude se systémem pracovat

Informační systém bude navržen jako víceuživatelský systém, se kterým budou pracovat uživatelé s různými právy. Budou tak mít přístupné i různé funkce.

Se systémem bude pracovat:

- **zaměstnanec**, který má možnost si objednávat videokazety k zapůjčení nebo k promítání, sledovat své aktuální vypůjčené videokazety, sledovat historii výpůjček a promítání a sledovat upomínky,
- **správce**, který spravuje veškerou evidenci, zaznamenává výpůjčky a vrácení videokazet do systému.

Poznámka: Na straně serveru se bude jednou denně spouštět (nejlépe v době, kdy se se systémem nepracuje) úloha, která bude vykonávat automatické akce související s upozorňováním uživatelů na vyzvednutí objednávky, upozornění na vrácení výpůjčky a zaslání upomínky.

### 2.2.3 Požadované funkce systému

Informační systém bude pro svou komunikaci s uživatelem obsahovat následující funkce:

- 1) vedení evidence uživatelů, tzn. jejich vkládání, upravování a mazání,
- 2) vedení evidence videokazet, tzn. jejich vkládání, upravování a mazání,
- 3) vedení evidence pořadů, tzn. jejich vkládání, upravování a mazání,
- 4) vedení evidence číselníků (obor, země původu, jazyková verze, katedra), tzn. jejich vkládání, editace a mazání,
- 5) vytvoření, upravení a vymazání objednávky,
- 6) vytvoření a ukončení výpůjčky existující objednávky,
- 7) přehledy a zobrazení uživatelů, videokazet, pořadů, objednávek a výpůjček,
- 8) možnost listování v historii výpůjček a objednávek,
- 9) upozorňování na vyzvednutí objednávky, vrácení výpůjčky, zaslání upomínky.

Tyto funkce budou přístupné pouze uživateli přihlášenému do systému. Každý uživatel má k dispozici jen ty funkce, které odpovídají jeho uživatelským právům v informačním systému. Uživatel *správce* bude mít přístupné funkce číslo 1, 2, 3, 4, 5, 6, 7 a 8. Uživatel *zaměstnanec*



bude mít naopak přístupné funkce číslo 5, 7 a 8. Poslední funkce číslo 9 bude spouštěna automaticky na straně serveru v době, kdy se s aplikací nebude pracovat.

## 2.2.4 Vstupy a výstupy systému

- **vstupy:** informace o uživateli systému, videokazetách, pořadech a jejich umístění na videokazetách,
- **výstupy:** údaje o videokazetách, pořadech, objednávkách, výpůjčkách, historii výpůjček.

## 2.3 Nefunkční požadavky

Informační systém bude naprogramován v jazyce PHP (PHP4) s databází MySQL (MySQL4). Veškerá komunikace mezi uživatelem informačního systému a IS bude probíhat pomocí značkovacího jazyka XHTML.

## 2.4 Databázová terminologie

**atribut** – charakteristika, vlastnost entity (např. *osob* entity *Uživatel*); každý atribut je zadán svým názvem (identifikátorem) a datovým typem

**databáze** – místo na disku, které slouží pro ukládání dat

**databázový systém** – implementuje relační model, zajišťuje správu dat uložených v databázi

**datový tok** – vyjadřuje přesun dat nebo informací z jedné části systému do jiné, z okolí systému do systému nebo ze systému do okolí

**DFD** – data-flow diagram (diagram datových toků) je grafický prostředek pro návrh a zobrazení funkčního modelu systému

**entita** – libovolný objekt reálného světa; musí být jednoznačně odlišitelná od ostatních entit a existovat nezávisle na nich

**entitní typ** – množina objektů (entit) stejného typu (např. *Uživatel*)

**integritní omezení** – omezující podmínky kladené na atributy nebo tabulky

**model** – abstraktní popis reality

**paměť** – místo pro dočasné uchování dat pro jejich pozdější využití

**primární klíč** – jeden nebo množina atributů, které jednoznačně určují entitu v množině entit

**proces** – provádí transformaci vstupních dat na data výstupní, tzn. realizuje nějakou funkci nad daty

**relační model** – popisuje vztahy (relace) mezi tabulkami, do kterých se zapisují objekty vstupních vztahů

- řádek** – odpovídá jedné entitě a obsahuje všechny atributy entity; nazývá se taky záznam
- SQL** – dotazovací jazyk pro komunikaci s databázovým systémem; jazyk SQL představuje množinu příkazů pro správu databází, tabulek a samotných dat
- SŘBD** – systém řízení báze dat je programový systém, který umožňuje definování datových struktur a datových souborů, řeší fyzické uložení dat, umožňuje manipulaci s daty a formátování vstupních a výstupních informací
- tabulka** – entitní typ neboli množina entit, kde se zapisují data; obsahuje sloupce neboli atributy a řádky, do kterých se ukládají data
- vazba** – vztah neboli relace mezi dvěma nebo více entitními typy

## 3. Analýza projektu

Analýzou projektu je myšlena specifikace problému a jeho studium. Skládá se ze dvou základních částí, které popisují různé úrovně projektu. První částí je datová část a jí odpovídající datová analýza, druhou částí je část funkční, které odpovídá funkční analýza.

Datová analýza se zabývá datovou strukturou projektu a obsahuje následující prvky:

- přehled a význam entitních typů (tabulek);
- popis vazeb mezi jednotlivými entitami;
- popis typů atributů jednotlivých entit, význam a integritní omezení těchto atributů (integritním omezením je myšleno: datový typ atributu, rozsah jeho hodnoty, zda je klíčem, zda musí být vyplněn, apod.).

Funkční analýza se zabývá vztahy systému se svým okolím a procesy, které v systému probíhají. Detailně rozebírá funkce, které se v těchto procesech vyskytují. Funkční analýza obsahuje následující prvky:

- kontextový diagram, který obsahuje celý systém jako jednu funkci, definuje hranice systému a zobrazuje interakci systému se svým okolím;
- data-flow diagram, který popisuje dynamiku systému, modeluje funkce systému a graficky znázorňuje transformaci dat z jedné formy do druhé;
- minispecifikaci funkcí, která popisuje procesy na nejnižší úrovni hierarchického rozkladu modelu systému a upřesňuje logiku systému.

### 3.1 Datová analýza

Datová analýza modeluje pouze základní datové struktury systému a nezabývá se jeho funkční stránkou. Při vývoji celého systému bylo použito databáze relační, proto se pro popis této struktury používá *relační schéma*.

#### 3.1.1 Relační schéma databáze

Relační schéma databáze se skládá z následujících základních prvků:

- lineární zápis;
- E-R (Entity-Relationship) diagram;
- úplné tabulky atributů a jejich integritních omezení;
- popis jednotlivých entit a jejich význam.

### Lineární zápis seznamu typů entit a jejich atributů

Popisuje tabulky a jejich atributy:

**Uživatel** (osob, heslo, jméno, příjmení, telefon, titulpřed, titulza, *id\_katedry*, email, stav, práva, datum, čas)

**Kazeta** (id\_kaz, c\_kaz, typ\_kaz, dat\_zal, dat\_kon, délka)

**Program** (id\_prog, dat\_zal, dat\_kon, *id\_oboru*, mdt, název, režisér, výrob, *id\_země*, rok, zisk, *id\_ver*, délka, popis)

**Katedra** (id\_katedry, c\_katedry, název)

**Obor** (id\_oboru, název)

**Země** (id\_země, jméno)

**Verze** (id\_ver, název)

**Umístění** (id\_um, *id\_kaz*, *id\_prog*, pořadí, začátek, dat\_zal)

**Výpůjčka** (id\_vyp, *osob*, dat\_vyp, čas\_vyp, dní, upomínka, dat\_od, dat\_do)

**Výpůjčka\_položka** (*id\_vyp*, *id\_kaz*)

**Objednávka** (id\_obj, *osob*, dat\_obj, čas\_obj, typ, dní, dat\_od, dat\_do, čas\_od, čas\_do, stav)

**Objednávka\_položka** (*id\_obj*, *id\_kaz*)

**H\_umístění** (id\_um, *id\_kaz*, *id\_prog*, pořadí, začátek, dat\_zal, dat\_kon)

**H\_výpůjčka** (id\_vyp, *osob*, dat\_vyp, čas\_vyp, dní, upomínka, dat\_od, dat\_do, dat\_vrac, čas\_vrac)

**H\_výpůjčka\_položka** (*id\_vyp*, *id\_kaz*)

**H\_mítnost** (id\_obj, *osob*, datum, čas\_od, čas\_do)

**H\_mítnost\_položka** (*id\_obj*, *id\_kaz*)

**Nastavení** (id, dní\_zap, dní\_obj, dní\_upom, dní\_upozor, email, daemon, daemončas)

**Novinky** (id, datum, čas, text)

### Lineární zápis seznamu vztahových entit

Popisuje vztahy mezi tabulkami:

**NÁLEŽÍ\_DO** (Program, Obor)

**JE\_V** (Program, Verze)

**JE\_VYROBEN** (Program, Země)

**JE\_UMÍSTĚN** (Program, Umístění)

**JE\_UMÍSTĚN\_H** (Program, H\_umístění)

**OBSAHUJE\_1** (Kazeta, Umístění)

**OBSAHUJE\_H1** (Kazeta, H\_umístění)

**NACHÁZÍ\_SE\_1** (Kazeta, Výpůjčka\_položka)

**NACHÁZÍ\_SE\_2** (Kazeta, Objednávka\_položka)

**NACHÁZÍ\_SE\_H1** (Kazeta, H\_výpůjčka\_položka)

**NACHÁZÍ\_SE\_H2** (Kazeta, H\_místnost\_položka)

**OBSAHUJE\_2** (Výpůjčka, Výpůjčka\_položka)

**OBSAHUJE\_3** (Objednávka, Objednávka\_položka)

**OBSAHUJE\_H2** (H\_výpůjčka, H\_výpůjčka\_položka)

**OBSAHUJE\_H3** (H\_místnost, H\_místnost\_položka)

**MÁ\_VÝPŮJČENO** (Uživatel, Výpůjčka)

**MĚL\_VÝPŮJČENO** (Uživatel, H\_výpůjčka)

**PATRÍ\_DO** (Uživatel, Katedra)

**MÁ\_OBJEDNÁNO** (Uživatel, Objednávka)

**MĚL\_OBJEDNÁNO** (Uživatel, H\_místnost)



## Popis tabulek

- Uživatel** – tabulka popisuje uživatele IS Audiovizuálního centra. Odstranění uživatelé ze systému nejsou zcela vymazáni, pouze hodnota atributu *stav* se nastaví na hodnotu „smazán“.
- Kazeta** – tabulka obsahuje informace o videokazetách, které jsou nabízeny systémem. Odstraněné videokazety nejsou zcela vymazány ze systému, pouze hodnota atributu *dat\_kon* se nastaví na hodnotu „odstraněno“.
- Program** – tabulka obsahuje informace o videopořadech, které jsou k dispozici v systému. Smazané pořady jsou označeny pouze hodnotou „odstraněno“ v atributu *dat\_kon*.
- Katedra** – tabulka obsahuje informace o katedrách. Slouží jako číselník kateder IS.
- Obor** – tabulka obsahuje informace o oborech a slouží tak jako číselník oborů příslušných videopořadů.
- Země** – tabulka slouží jako číselník zemí původů příslušných videopořadů.
- Verze** – tabulka obsahuje informace o jazykových verzích a slouží tak jako číselník.
- Umístění** – tabulka obsahuje informace o aktuálním umístění videopořadů na videokazetách. Slouží tak jako vazební tabulka mezi tabulkami **Program** a **Kazeta**.
- Výpůjčka** – tabulka slouží k uchovávání aktuálních výpůjček uživatelů systému.
- Výpůjčka\_položka** – tabulka obsahuje informace o videokazetách, které jsou položkami jednotlivých výpůjček.
- Objednávka** – tabulka slouží ke dvojímu účelu. Prvním účelem je uchovávání informací o objednaných videokazetách. Druhým účelem jsou objednávky videokazet k promítání, tzn. tímto si uživatel objedná místnost sloužící k promítání.
- Objednávka\_položka** – tabulka obsahuje informace o videokazetách, které jsou položkami jednotlivých objednávek.
- H\_umístění** – tabulka obsahuje informace o historii umístění videopořadů na videokazetách. Informace se zde přesunou z tabulky **Umístění**, je-li vymazán nějaký pořad z videokazety (což je méně častý jev), nebo je-li odstraněna videokazeta nebo videopořad.
- H\_výpůjčka** – tabulka obsahuje informace o historii výpůjček jednotlivých uživatelů. Údaje se zde přesouvají z tabulky **Výpůjčka**.
- H\_výpůjčka\_položka** – tabulka obsahuje informace o videokazetách, které byly položkami jednotlivých výpůjček. Údaje jsou zde přesunuty z tabulky **Výpůjčka\_položka**.
- H\_místnost** – tabulka obsahuje informace o historii promítání videokazet v místnosti určené k projekci těchto videokazet. Údaje jsou do této tabulky přesunuty z tabulky **Objednávka**.
- H\_místnost\_položka** – tabulka obsahuje informace o videokazetách, které byly položkami jednotlivých promítání v promítací místnosti. Údaje se zde přesouvají z tabulky **Objednávka\_položka**.
- Nastavení** – tabulka obsahuje informace, které jsou důležité pro běh celé aplikace IS.

**Novinky** – tabulka obsahuje informace o novinkách týkající se IS Audiovizuálního centra, které vkládá správce systému.

#### Atributy tabulek a jejich integritní omezení

##### **Uživatel**

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Uživatelské jméno	osob	varchar	6	Ano	Ne	Osobní číslo
Heslo	heslo	varchar	40	Ne	Ne	MD5 hash hesla
Jméno	jmeno	varchar	40	Ne	Ne	
Příjmení	prijmeni	varchar	40	Ne	Ne	
Telefonní číslo	telefon	integer	9	Ne	Ano	
Tituly před jménem	titulpred	varchar	20	Ne	Ano	
Tituly za jménem	titulza	varchar	20	Ne	Ano	
Identifikátor katedry	id_katedry	integer	4	Ne	Ano	Ukazuje do tabulky Katedra
E-mailová adresa	email	varchar	100	Ne	Ne	
Stav uživatele	stav	varchar	1	Ne	Ne	A – aktivní, D – smazaný
Práva uživatele	prava	integer	1	Ne	Ne	0 – zaměstnanec, 1 – správce
Datum posledního přihlášení uživatele	datum	date		Ne	Ano	Ve formátu RRRR-MM-DD
Čas posledního přihlášení uživatele	cas	time		Ne	Ano	Ve formátu HH:MM:SS

##### **Kazeta**

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor kazety	id_kaz	integer	6	Ano	Ne	Autoincrement
Číslo kazety	c_kaz	integer	6	Ne	Ne	Číslo kazety nemusí být shodné s identifikátorem
Typ kazety	typ_kaz	varchar	4	Ne	Ano	
Datum vložení kazety do systému	dat_zal	date		Ne	Ne	Ve formátu RRRR-MM-DD
Datum odstranění kazety ze systému	dat_kon	date		Ne	Ano	Ve formátu RRRR-MM-DD
Délka kazety v minutách	delka	integer	3	Ne	Ne	



## Program

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor pořadu	id_prog	integer	8	Ano	Ne	Autoincrement
Datum vložení pořadu do systému	dat_zal	date		Ne	Ne	Ve formátu RRRR-MM-DD
Datum odstranění pořadu ze systému	dat_kon	date		Ne	Ano	Ve formátu RRRR-MM-DD
Identifikátor oboru	id_oboru	integer	4	Ne	Ne	Ukazuje do tabulky Obor
MDT pořadu	mdt	varchar	7	Ne	Ne	
Název pořadu	nazev	varchar	100	Ne	Ne	
Režisér pořadu	reziser	varchar	100	Ne	Ano	
Výrobce pořadu	vyrob	integer	100	Ne	Ano	
Identifikátor země původu	id_zeme	integer	6	Ne	Ne	Ukazuje do tabulky Země
Rok výroby pořadu	rok	integer	4	Ne	Ano	Ve formátu RRRR
Způsob získání pořadu	zisk	varchar	50	Ne	Ano	
Identifikátor jazykové verze	id_ver	integer	6	Ne	Ne	Ukazuje do tabulky Verze
Délka pořadu v minutách	delka	integer	3	Ne	Ne	
Popis pořadu	popis	text		Ne	Ano	

## Katedra

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor katedry	id_katedry	integer	4	Ano	Ne	Autoincrement
Číslo katedry	c_katedry	integer	4	Ne	Ne	
Název katedry	nazev	varchar	50	Ne	Ne	

## Obor

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor oboru	id_oboru	integer	4	Ano	Ne	Autoincrement
Název oboru	nazev	varchar	100	Ne	Ne	

## Země

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor země původu	id_zeme	integer	6	Ano	Ne	Autoincrement
Jméno země původu	jmeno	varchar	100	Ne	Ne	

### Verze

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor jazykové verze	id_ver	integer	6	Ano	Ne	Autoincrement
Název jazykové verze	nazev	varchar	100	Ne	Ne	

### Umístění

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor umístění	id_um	integer	8	Ano	Ne	Autoincrement
Identifikátor kazety	id_kaz	integer	6	Ne	Ne	Ukazuje do tabulky Kazeta
Identifikátor pořadu	id_prog	integer	8	Ne	Ne	Ukazuje do tabulky Program
Pořadí pořadu na kazetě	poradi	integer	2	Ne	Ne	
Minuta začátku pořadu na kazetě	zacatek	integer	3	Ne	Ne	
Datum přidání pořadu na kazetu	dat_zal	date		Ne	Ne	Ve formátu RRRR-MM-DD

### Výpůjčka

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor výpůjčky	id_vyp	integer	8	Ano	Ne	Autoincrement
Uživatelské jméno	osob	varchar	6	Ne	Ne	Ukazuje do tabulky Uživatel
Datum uskutečnění výpůjčky	dat_vyp	date		Ne	Ne	Ve formátu RRRR-MM-DD
Čas uskutečnění výpůjčky	cas_vyp	time		Ne	Ne	Ve formátu HH:MM:SS
Počet upomínek	upominka	integer	3	Ne	Ne	
Datum počátku výpůjčky	dat_od	date		Ne	Ne	Ve formátu RRRR-MM-DD
Datum konce výpůjčky	dat_do	date		Ne	Ne	Ve formátu RRRR-MM-DD

### Výpůjčka\_položka

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor výpůjčky	id_vyp	integer	8	Ano	Ne	Ukazuje do tabulky Výpůjčka
Identifikátor kazety	id_kaz	integer	6	Ne	Ne	Ukazuje do tabulky Kazeta

### Objednávka

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor objednávky	id_obj	integer	8	Ano	Ne	Autoincrement
Uživatelské jméno	osob	varchar	6	Ne	Ne	Ukazuje do tabulky Uživatel
Datum objednávky	dat_obj	date		Ne	Ne	Ve formátu RRRR-MM-DD
Čas objednávky	cas_obj	time		Ne	Ne	Ve formátu HH:MM:SS
Typ objednávky	typ	integer	2	Ne	Ne	1 – objednávka kazet 2 – objednávka místnosti
Počet dní na vyzvednutí objednávky	dni	integer	3	Ne	Ano	Pro typ objednávky 1
Datum počátku výpůjčky nebo datum promítání	dat_od	date		Ne	Ano	Pro typ objednávky 1
Datum konce výpůjčky	dat_do	date		Ne	Ano	Pro typ objednávky 1
Čas počátku promítání	cas_od	time		Ne	Ano	Pro typ objednávky 2
Čas konce promítání	cas_do	time		Ne	Ano	Pro typ objednávky 2
Stav objednávky	stav	integer	1	Ne	Ano	0 – objednáva se 1 – objednáno

### Objednávka\_položka

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor objednávky	id_obj	integer	8	Ano	Ne	Ukazuje do tabulky Objednávka
Identifikátor kazety	id_kaz	integer	6	Ne	Ne	Ukazuje do tabulky Kazeta

### H\_umístění

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor umístění	id_um	integer	8	Ano	Ne	Autoincrement
Identifikátor kazety	id_kaz	integer	6	Ne	Ne	Ukazuje do tabulky Kazeta
Identifikátor pořadu	id_prog	integer	8	Ne	Ne	Ukazuje do tabulky Program
Pořadí pořadu na kazetě	poradi	integer	2	Ne	Ne	
Minuta začátku pořadu na kazetě	zacatek	integer	3	Ne	Ne	
Datum přidání pořadu na kazetu	dat_zal	date		Ne	Ne	Ve formátu RRRR-MM-DD
Datum odstranění pořadu z kazety	dat_kon	date		Ne	Ne	Ve formátu RRRR-MM-DD

### H\_výpůjčka

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor výpůjčky	id_vyp	integer	8	Ano	Ne	Autoincrement
Uživatelské jméno	osob	varchar	6	Ne	Ne	Ukazuje do tabulky Uživatel
Datum uskutečnění výpůjčky	dat_vyp	date		Ne	Ne	Ve formátu RRRR-MM-DD
Čas uskutečnění výpůjčky	cas_vyp	time		Ne	Ne	Ve formátu HH:MM:SS
Počet upomínek	upominka	integer	3	Ne	Ne	
Datum počátku výpůjčky	dat_od	date		Ne	Ne	Ve formátu RRRR-MM-DD
Datum konce výpůjčky	dat_do	date		Ne	Ne	Ve formátu RRRR-MM-DD
Datum vrácení výpůjčky	dat_vrac	date		Ne	Ne	Ve formátu RRRR-MM-DD
Čas vrácení výpůjčky	cas_vrac	time		Ne	Ne	Ve formátu HH:MM:SS

### H\_výpůjčka\_položka

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor výpůjčky	id_vyp	integer	8	Ano	Ne	Ukazuje do tabulky Výpůjčka
Identifikátor kazety	id_kaz	integer	6	Ne	Ne	Ukazuje do tabulky Kazeta

### H\_místnost

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor objednávky	id_obj	integer	8	Ano	Ne	Autoincrement
Uživatelské jméno	osob	varchar	6	Ne	Ne	Ukazuje do tabulky Uživatel
Datum promítání	datum	date		Ne	Ne	Ve formátu RRRR-MM-DD
Čas počátku promítání	cas_od	time		Ne	Ano	Ve formátu HH:MM:SS
Čas konce promítání	cas_do	time		Ne	Ano	Ve formátu HH:MM:SS

### H\_místnost\_položka

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor objednávky	id_obj	integer	8	Ano	Ne	Ukazuje do tabulky Objednávka
Identifikátor kazety	id_kaz	integer	6	Ne	Ne	Ukazuje do tabulky Kazeta

### Nastavení

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor nastavení	id	integer	1	Ano	Ne	1 – defaultní nastavení 2 – nastavení správcem
Počet dní na zapůjčení kazety	dni_zap	integer	3	Ne	Ne	
Počet dní na vyzvednutí objednávky	dat_obj	integer	3	Ne	Ne	
Počet dní, za které se pošle další upomínka	dni_upom	integer	3	Ne	Ne	
Počet dní před koncem výpůjčky, upozorní se na její vrácení	dni_upozor	integer	3	Ne	Ne	
Systémový e-mail	email	varchar	100	Ne	Ne	
Datum posledního spuštění daemona	daemon	Date		Ne	Ano	Ve formátu RRRR-MM-DD
Čas posledního spuštění daemona	daemon_cas	time		Ne	Ano	Ve formátu HH:MM:SS

### Novinky

Atribut	Identifikátor	Dat.typ	Délka	Klíč	Null	Poznámka
Identifikátor novinky	id	integer	6	Ano	Ne	Autoincrement
Datum vložení novinky	datum	date		Ne	Ne	Ve formátu RRRR-MM-DD
Čas vložení novinky	cas	time		Ne	Ne	Ve formátu HH:MM:SS
Text novinky	text	text		Ne	Ne	

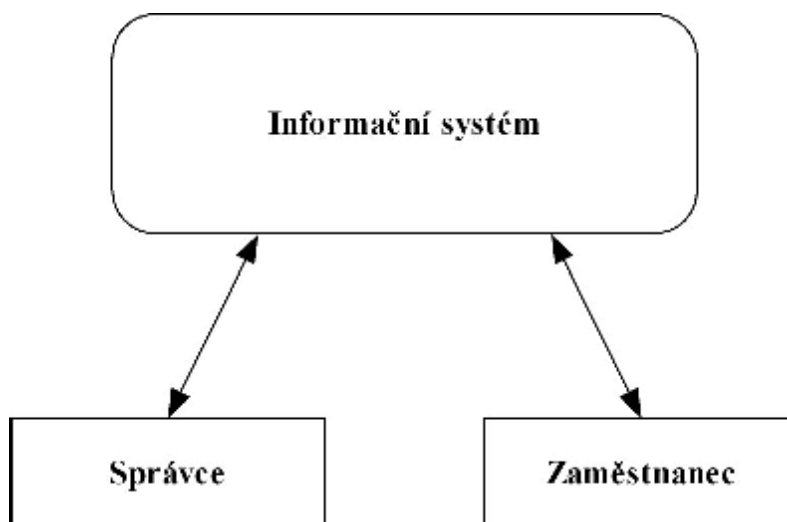
## 3.2 Funkční analýza

Funkční analýza modeluje systém jako množinu navzájem spolupracujících akcí. Systém je modelován pomocí procesů, které jsou propojeny datovými toky s datovými paměťmi a terminátory. Proces provádí transformaci vstupních dat na data výstupní a realizuje nějakou funkci.

### 3.2.1 Kontextový diagram

Kontextový diagram obsahuje celý systém jako jednu funkci, definuje hranice systému a všechny terminátory. Terminátory jsou zde myšleny externí zdroje a cíle dat, které jsou vně systému.

Kontextový diagram Informačního systému Audiovizuálního centra VŠB-TUO je zobrazen na obr. 3.2.



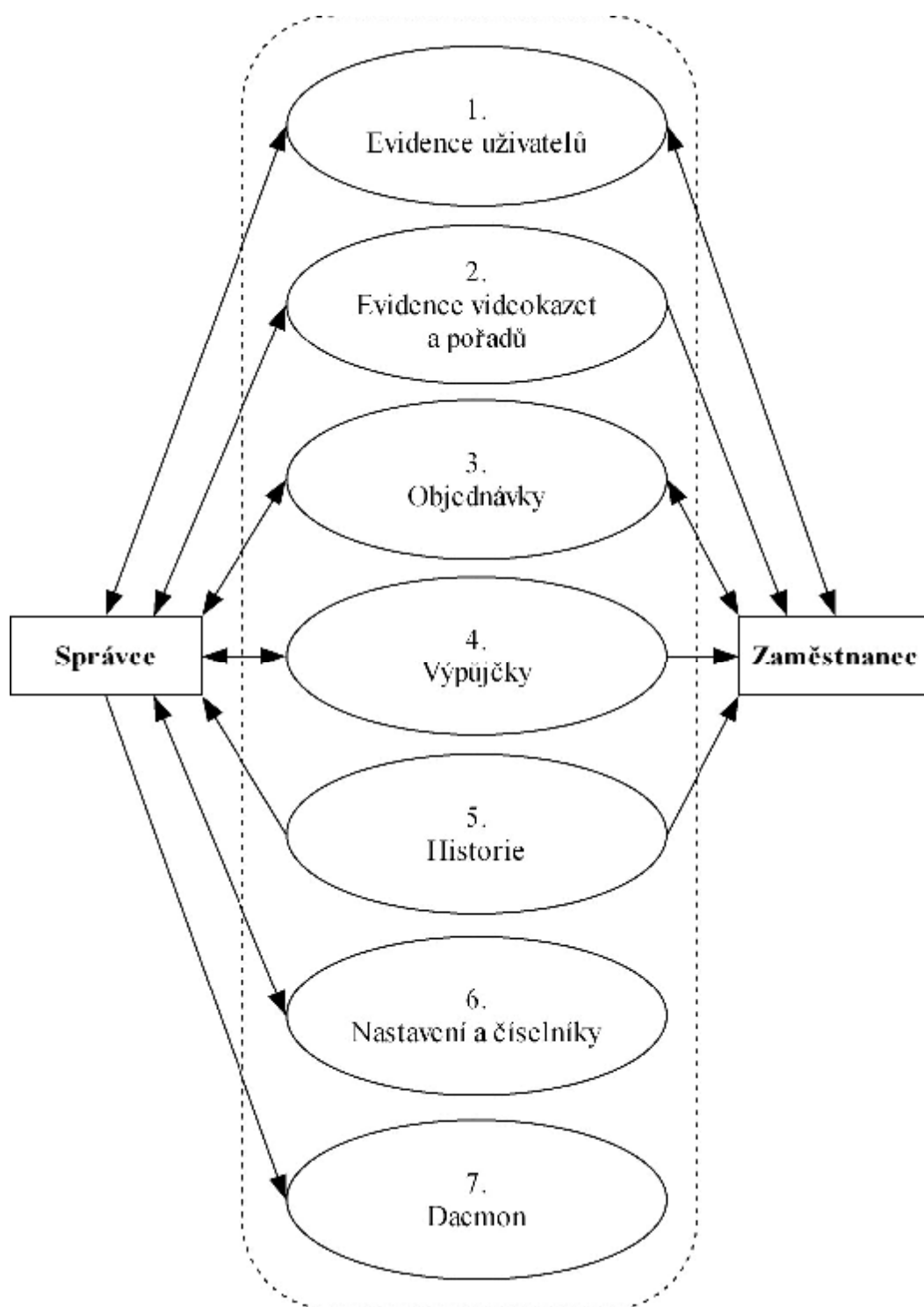
**Obr. 3.2** Kontextový diagram funkční analýzy IS Audiovizuálního centra VŠB-TUO

### 3.2.2 Data-flow diagram

Data-flow diagram (DFD, diagram datových toků) je grafický prostředek pro návrh a zobrazení funkčního modelu systému. DFD popisuje dynamiku systému, vyjadřuje transformace dat z jedné formy do druhé, modeluje funkce systému pomocí grafu a přitom používá následujících grafických prvků:

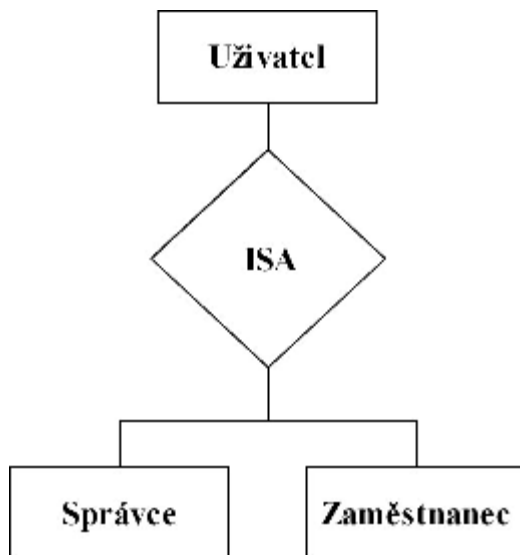
- proces – jak již bylo zmíněno na počátku kapitoly 3.2, provádí transformaci vstupních dat na data výstupní a realizuje nějakou funkci;
- paměť – je místo pro dočasné uchování dat pro jejich pozdější použití;
- terminátor – znázorňuje externí zdroj nebo cíl dat, je to objekt vně systému, který se systémem komunikuje;
- datový tok – vyjadřuje přesun dat z jedné části systému do druhé, z okolí systému do systému nebo ze systému do jeho okolí.

Bezprostředním rozkladem kontextového diagramu je DFD úrovně 0, který zobrazuje rozklad systému na subsystémy. Pro IS Audiovizuálního centra VŠB-TUO je toto schéma zobrazeno na obr. 3.3. Schéma je pro větší přehlednost znázorněno jako Use Case diagram.



**Obr. 3.3** DFD úrovně 0 (Use Case) IS Audiovizuálního centra VŠB-TUO

Za DFD úrovně 0 následují DFD dalšího rozkladu systému až na elementární úroveň rozkladu. Tyto DFD jsou zobrazeny na obr. A-1 až A-28 v příloze A.



**Obr. 3.4** ISA hierarchie sloučení terminátorů Správce a Zaměstnanec

V některých data-flow diagramech došlo ke sloučení terminátorů (uživatelů) Správce a Zaměstnanec v terminátor Uživatel. Toto sloučení bylo provedeno v případě, že oba dva z terminátorů měli společné funkce a datové toky. Tím bylo dosaženo větší přehlednosti DFD. Sloučení obou terminátorů je vyobrazeno ISA hierarchií na obr. 3.4.

### 3.2.3 Minispecifikace funkcí

#### *1.: Evidence uživatelů*

##### *1.1.: Evidence uživatelů*

##### *1.1.1.: Nový záznam uživatele*

1. ZOBRAZ formulář pro vložení nového uživatele
2. Správce zadá údaje o novém uživateli (povinné: osob, heslo, jméno, příjmení, email, práva; nepovinné: titulpred, titulza, telefon, id\_katedry)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů, existence osob v tabulce Uživatel)
4. ZAPIŠ nového uživatele (do Uživatel)
5. VYPIŠ nového uživatele



### *1.1.2.: Upravení uživatele*

1. ZOBRAZ formulář s přehledem uživatelů
2. Správce zadá osob uživatele, u kterého se budou měnit údaje
3. NAJDI uživatele podle osob (v Uživatel)
4. ZOBRAZ formulář pro změnu údajů o nalezeném uživateli
5. Správce zadá změněné údaje o uživateli (povinné: osob, heslo, jméno, příjmení, email, práva; nepovinné: titulpred, titulza, telefon, id\_katedry)
6. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů, existence osob v tabulce Uživatel)
7. ZAPIŠ změněné údaje o uživateli (do Uživatel)
8. VYPIŠ změněného uživatele

### *1.1.3.: Smazání uživatele*

1. ZOBRAZ formulář s přehledem uživatelů
2. Správce zadá osob uživatele pro smazání
3. NAJDI A SMAŽ všechny výpůjčky uživatele (v Výpůjčka, Výpůjčka\_položka)
4. NAJDI A SMAŽ všechny objednávky uživatele (v Objednávka, Objednávka\_položka)
5. NASTAV stav na hodnotu „D“
6. ZAPIŠ změněný údaj (do Uživatel)
7. VYPIŠ (Uživatel byl úspěšně smazán)

### *1.1.4.: Obnovení uživatele*

1. ZOBRAZ formulář s přehledem smazaných uživatelů
2. Správce zadá osob uživatele pro obnovení
3. NAJDI uživatele podle osob (v Uživatel)
4. NASTAV stav na hodnotu „A“
5. ZAPIŠ změněný údaj (do Uživatel)
6. VYPIŠ obnoveného uživatele

## **1.2.: Přehled a zobrazení uživatele**

### *1.2.1.: Přehled uživatelů*

1. ZOBRAZ formulář pro zadání kritérií k vyhledání uživatelů
2. Uživatel zadá kritéria pro vyhledání uživatele
3. NAJDI všechny uživatele dle zadaných kritérií (v Uživatel)
4. VYPIŠ přehled s nalezenými uživateli

### *1.2.2.: Zobrazení uživatele*

1. ZOBRAZ formulář s přehledem uživatelů
2. Uživatel zadá osob uživatele pro zobrazení
3. NAJDI uživatele podle osob (v Uživatel)

4. VYPIŠ údaje o uživateli

### ***1.3.: Osobní údaje uživatele***

#### ***1.3.1.: Zobrazení osobních údajů***

1. NAJDI uživatele podle osob přihlášeného uživatele (v Uživateli)
2. VYPIŠ údaje o uživateli

#### ***1.3.2.: Upravení osobních údajů***

1. NAJDI uživatele podle osob přihlášeného uživatele (v Uživateli)
2. ZOBRAZ formulář pro změnu osobních údajů
3. Uživatel zadá změněné osobní údaje (povinné: osob, heslo, jméno, příjmení, email, práva; nepovinné: titulpred, titulza, telefon, id\_katedry)
4. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů, existence osob v tabulce Uživateli)
5. ZAPIŠ změněné osobní údaje uživatele (do Uživatele)
6. VYPIŠ změněné osobní údaje uživatele

#### ***1.3.3.: Změna hesla***

1. ZOBRAZ formulář pro změnu hesla přihlášeného uživatele
2. Uživatel zadá staré heslo, nové heslo a potvrzení nového hesla
3. PROVEĎ kontrolu starého hesla a shody nového hesla s potvrzením
4. NASTAV heslo na nové heslo
5. ZAPIŠ změněný údaj (do Uživatele)
6. VYPIŠ (Změna hesla proběhla úspěšně)

## ***2.: Evidence videokazet a pořadů***

### ***2.1.: Evidence videokazet***

#### ***2.1.1.: Editace videokazet***

##### ***2.1.1.1.: Nový záznam videokazety***

1. ZOBRAZ formulář pro vložení nové videokazety
2. Správce zadá údaje o nové videokazetě (povinné: c\_kaz, délka; nepovinné: typ\_kaz)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů, existence c\_kaz v tabulce Kazeta)
4. ZAPIŠ novou videokazetu (do Kazeta)
5. VYPIŠ novou videokazetu

#### *2.1.1.2.: Upravení videokazety*

1. ZOBRAZ formulář s přehledem videokazet
2. Správce zadá id\_kaz videokazety, u které se budou měnit údaje
3. NAJDI videokazetu podle id\_kaz (v Kazeta)
4. ZOBRAZ formulář pro změnu údajů o nalezené videokazetě
5. Správce zadá změněné údaje o videokazetě (povinné: c\_kaz, délka; nepovinné: typ\_kaz)
6. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů, existence c\_kaz v tabulce Kazeta)
7. ZAPIŠ změněné údaje o videokazetě (do Kazeta)
8. VYPIŠ změněnou videokazetu

#### *2.1.1.3.: Smazání videokazety*

1. ZOBRAZ formulář s přehledem videokazet
2. Správce zadá id\_kaz videokazety pro smazání
3. PROVEĎ kontrolu existence výpůjčky videokazety (je-li nalezena, videokazetu nelze smazat)
4. NAJDI A SMAŽ všechny položky objednávek s videokazetou id\_kaz (v Objednávka\_položka)
5. NAJDI A SMAŽ všechna umístění na této videokazetě s id\_kaz (v Umístění)
6. UKONČI umísťování pořadů na tuto videokazetu (je-li umísťování započato)
7. NASTAV dat\_kon na aktuální datum
8. ZAPIŠ změněný údaj (v Kazeta)
9. VYPIŠ (Videokazeta byla úspěšně smazána)

### **2.1.2.: Přehled a zobrazení videokazet**

#### *2.1.2.1.: Přehled videokazet*

1. ZOBRAZ formulář pro zadání kritérií k vyhledání videokazet
2. Uživatel zadá kritéria pro vyhledání videokazet
3. NAJDI všechny videokazety dle zadaných kritérií (v Kazeta)
4. VYPIŠ přehled s nalezenými videokazetami

#### *2.1.2.2.: Zobrazení videokazety*

1. ZOBRAZ formulář s přehledem videokazet
2. Uživatel zadá id\_kaz videokazety pro zobrazení
3. NAJDI videokazetu podle id\_kaz (v Kazeta)
4. VYPIŠ údaje o videokazetě

## **2.2.: Evidence pořadů**

### **2.2.1.: Editace pořadů**

#### *2.2.1.1.: Nový záznam pořadu*

1. ZOBRAZ formulář pro vložení nového pořadu
2. Správce zadá údaje o novém pořadu videokazetě (povinné: mdt, název, id\_oboru, id\_země, id\_ver, délka, dat\_zal; nepovinné: režisér, výrob, rok, zisk, popis)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
4. ZAPIŠ nový pořad (do Program)
5. VYPIŠ nový pořad

#### *2.2.1.2.: Upravení pořadu*

1. ZOBRAZ formulář s přehledem pořadů
2. Správce zadá id\_prog pořadu, u kterého se budou měnit údaje
3. NAJDI pořad podle id\_prog (v Program)
4. ZOBRAZ formulář pro změnu údajů o nalezeném pořadu
5. Správce zadá změněné údaje o pořadu (povinné: mdt, název, id\_oboru, id\_země, id\_ver, délka, dat\_zal; nepovinné: režisér, výrob, rok, zisk, popis)
6. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
7. ZAPIŠ změněné údaje o pořadu (do Program)
8. VYPIŠ změněný pořad

#### *2.2.1.3.: Smazání pořadu*

1. ZOBRAZ formulář s přehledem pořadů
2. Správce zadá id\_prog pořadu pro smazání
3. PROVEĎ kontrolu existence výpůjčky videokazety s pořadem (je-li nalezena, pořad nelze smazat)
4. NAJDI A SMAŽ všechna umístění pořadů na kazetách s id\_prog (v Umístění)
5. NASTAV dat\_kon na aktuální datum
6. ZAPIŠ změněný údaj (v Program)
7. VYPIŠ (Pořad byl úspěšně smazán)

### **2.2.2.: Přehled a zobrazení pořadů**

#### *2.2.2.1.: Přehled pořadů*

1. ZOBRAZ formulář pro zadání kritérií k vyhledání pořadů
2. Uživatel zadá kritéria pro vyhledání pořadů
3. NAJDI všechny pořady dle zadaných kritérií (v Program)
4. VYPIŠ přehled s nalezenými pořady

#### *2.2.2.2.: Zobrazení pořadu*

1. ZOBRAZ formulář s přehledem pořadů
2. Uživatel zadá id\_prog pořadu pro zobrazení
3. NAJDI pořad podle id\_prog (v Program)
4. VYPIŠ údaje o pořadu

### **2.3.: Evidence umístění**

#### *2.3.1.: Smazání umístění*

1. ZOBRAZ formulář s umístováním pořadu na videokazetu
2. Správce zadá id\_kaz videokazety pro smazání umístění všech pořadů z videokazety nebo id\_prog pro smazání umístění jednoho pořadu ze všech videokazet nebo id\_kaz a id\_prog pro smazání umístění jednoho pořadu z jedné videokazety
3. NAJDI všechna umístění s id\_kaz nebo s id\_prog nebo s id\_kaz a id\_prog (v Umístění)
4. ZAPIŠ (id\_um, id\_kaz, id\_prog, pořadí, začátek, dat\_zal, dat\_kon (= aktuální datum)) pro všechny nalezené údaje (do H\_umístění)
5. SMAŽ údaje s id\_kaz nebo s id\_prog nebo s id\_kaz a id\_prog (z Umístění)
6. VYPIŠ (Smazání umístění proběhlo úspěšně)

#### *2.3.2.: Vytvoření umístění*

1. ZOBRAZ formulář pro vytvoření nového umístění
2. Správce zadá údaje o novém umístění (id\_kaz, id\_prog, pořadí, začátek)
3. ZAPIŠ zadané údaje (do Umístění)
4. VYPIŠ (Umístění proběhlo úspěšně)

### **3.: Objednávky**

#### **3.1.: Editace objednávky**

##### *3.1.1.: Vytvoření objednávky*

1. ZOBRAZ formulář pro vytvoření nové objednávky
2. Uživatel zadá údaje o objednavce (povinné: osob, dat\_obj, čas\_obj, typ; povinné pro typ „kazety“: dní, dat\_od, dat\_do; povinné pro typ „promítání“: dat\_od, čas\_od, čas\_do)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
4. ZAPIŠ novou objednávku (do Objednávka)
5. VYPIŠ právě probíhající objednávku
6. VYPIŠ (Objednávka byla úspěšně započata, nyní lze vkládat kazety)
7. ZOBRAZ formulář pro přidání nebo vyjmutí videokazety z objednávky (viz 3.1.3., 3.1.4.)

##### *3.1.2.: Smazání objednávky*

1. ZOBRAZ formulář s přehledem objednávek

2. Správce zadá id\_obj objednávky pro smazání
3. NAJDI A SMAŽ všechny položky objednávky s id\_obj (v Objednávka\_položka)
4. NAJDI A SMAŽ objednávku s id\_obj (v Objednávka)
5. VYPIŠ (Objednávka byla úspěšně smazána)

#### *3.1.3.: Přidání kazety do objednávky*

1. ZOBRAZ formulář s probíhající objednávkou
2. Uživatel zadá id\_kaz kazety přidávané do objednávky
3. PROVEĎ kontrolu, zda přidávaná kazeta již není obsažena v objednávce
4. ULOŽ položku objednávky (do Objednávka\_položka)
5. VYPIŠ (Kazeta byla úspěšně přidána do objednávky)

#### *3.1.4.: Vyjmutí kazet z objednávky*

1. ZOBRAZ formulář s probíhající objednávkou
2. Uživatel zadá id\_kaz kazety vyjímané z objednávky
3. SMAŽ položku objednávky (v Objednávka\_položka)
4. VYPIŠ (Kazety byla úspěšně vyjmuta z objednávky)

### **3.2.: Přehled a zobrazení objednávek**

#### *3.2.1.: Přehled objednávek*

1. ZOBRAZ formulář pro zadání kritérií k vyhledání objednávek
2. Uživatel zadá kritéria pro vyhledání objednávek
3. NAJDI všechny objednávky dle zadaných kritérií (v Objednávka)
4. VYPIŠ přehled s nalezenými objednávkami

#### *3.2.2.: Zobrazení objednávky*

1. ZOBRAZ formulář s přehledem objednávek
2. Uživatel zadá id\_obj objednávky pro zobrazení
3. NAJDI objednávku podle id\_obj (v Objednávka, Objednávka\_položka)
4. VYPIŠ údaje o objednávce a její položky

#### *3.2.3.: Ukončení promítání*

1. ZOBRAZ formulář s přehledem objednávek
2. Správce zadá id\_obj objednávky pro ukončení promítání
3. PROVEĎ kontrolu, zda promítání mohlo proběhnout a lze ukončit
4. NASTAV A ZAPIŠ údaje pro vložení do historie (id\_obj, osob, datum, čas\_od, čas\_do) (do H\_mítnost)
5. NAJDI položky objednávky s id\_obj (z Objednávka\_položka)
6. NASTAV A ZAPIŠ údaje o položce objednávky pro vložení do historie (id\_obj, id\_kaz) (do H\_mítnost\_položka) pro všechny nalezené záznamy

7. SMAŽ položky objednávky s id\_obj (z Objednávka\_položka)
8. SMAŽ objednávku s id\_obj (z Objednávka)
9. VYPIŠ (Promítání videokazet bylo úspěšně ukončeno)

#### **4.: Vypůjčky**

##### **4.1.: Vypůjčení a vrácení**

###### **4.1.1.: Vypůjčení**

1. ZOBRAZ formulář s přehledem objednávek
2. Správce zadá id\_obj objednávky pro vypůjčení
3. NAJDI objednávku s id\_obj (v Objednávka)
4. PROVEĎ kontrolu, zda již nastal počátek výpůjčky a zda byly všechny kazety v objednávce vráceny
5. NASTAV A ZAPIŠ údaje pro vytvoření výpůjčky (osob, dat\_vyp, čas\_vyp, dat\_od, dat\_do) (do Výpůjčka)
6. ZÍSKEJ vytvořené id\_vyp
7. NAJDI položky objednávky podle id\_obj (z Objednávka\_položka)
8. NASTAV A ZAPIŠ údaje o položce výpůjčky (id\_vyp, id\_kaz) (do Výpůjčka\_položka) pro všechny nalezené údaje
9. SMAŽ položky objednávky s id\_obj (v Objednávka\_položka)
10. SMAŽ objednávku s id\_obj (z Objednávka)
11. VYPIŠ (Objednávka byla úspěšně vypůjčena)

###### **4.1.2.: Vrácení**

1. ZOBRAZ formulář s přehledem výpůjček
2. Správce zadá id\_vyp výpůjčky
3. NAJDI výpůjčku podle id\_vyp (z Výpůjčka)
4. NASTAV A ZAPIŠ údaje pro vložení do historie (id\_vyp, osob, dat\_vyp, čas\_vyp, upomínka, dat\_od, dat\_do, dat\_vrac, čas\_vrac) (do H\_výpůjčka)
5. NAJDI položky výpůjčky s id\_vyp (z Výpůjčka\_položka)
6. NASTAV A ZAPIŠ údaje o položce výpůjčky pro vložení do historie (id\_vyp, id\_kaz) (do H\_výpůjčka\_položka) pro všechny nalezené údaje
7. SMAŽ položky výpůjčky s id\_vyp (z Výpůjčka\_položka)
8. SMAŽ výpůjčku s id\_vyp (z Výpůjčka)
9. VYPIŠ (Výpůjčka byla úspěšně vrácena)

##### **4.2.: Přehled a zobrazení výpůjček**

###### **4.2.1.: Přehled výpůjček**

1. ZOBRAZ formulář pro zadání kritérií k vyhledání výpůjček
2. Uživatel zadá kritéria pro vyhledání výpůjček
3. NAJDI všechny výpůjčky dle zadaných kritérií (v Výpůjčka)

4. VYPIŠ přehled s nalezenými výpůjčkami

#### *4.2.2.: Zobrazení výpůjčky*

1. ZOBRAZ formulář s přehledem výpůjček
2. Uživatel zadá id\_vyp výpůjčky pro zobrazení
3. NAJDI výpůjčku podle id\_vyp (ve Výpůjčka, Výpůjčka\_položka)
4. VYPIŠ údaje o výpůjčce a její položky

### **5.: Historie**

#### *5.1.: Přehled historie*

1. ZOBRAZ formulář pro zadání kritérií k vyhledání výpůjček a promítání v historii
2. Uživatel zadá kritéria pro vyhledání v historii
3. NAJDI všechny údaje v historii dle zadaných kritérií (v H\_umístění, H\_výpůjčka)
4. VYPIŠ přehled s nalezenými pořady

#### *5.2.: Zobrazení historie*

1. ZOBRAZ formulář s přehledem historie
2. Uživatel zadá id\_vyp výpůjčky nebo id\_obj promítání historie pro zobrazení
3. NAJDI údaj historie podle id\_vyp nebo id\_obj (v H\_výpůjčka, H\_výpůjčka\_položka nebo v H\_místnost, H\_místnost\_položka)
4. VYPIŠ údaje o historii a její položky

### **6.: Nastavení a číselníky**

#### **6.1.: Nastavení programu**

##### *6.1.1.: Zobrazení nastavení*

1. NAJDI nastavení (v Nastavení)
2. ZOBRAZ formulář s údaji o nastavení aplikace

##### *6.1.2.: Upravení nastavení*

1. ZOBRAZ formulář pro změnu údajů s nastavením
2. Správce zadá změněné údaje o nastavení (povinné: dní\_zap, dní\_obj, dní\_upom, dní\_upozor, email)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
4. ZAPIŠ změněné údaje (do Nastavení)
5. VYPIŠ změněné údaje



## **6.2.: Číselníky**

### **6.2.1.: Evidence kateder**

#### **6.2.1.1.: Nový záznam katedry**

1. ZOBRAZ formulář pro vložení nové katedry
2. Správce zadá údaje o nové katedře (povinné: c\_katedry, název)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
4. ZAPIŠ novou katedru (do Katedra)
5. VYPIŠ (Katedra byla úspěšně vložena)

#### **6.2.1.2.: Upravení katedry**

1. ZOBRAZ formulář s přehledem kateder
2. Správce zadá id\_katedry katedry, u které se budou měnit údaje
3. NAJDI katedru podle id\_katedry (v Katedra)
4. ZOBRAZ formulář pro změnu údajů o nalezené katedře
5. Správce zadá změněné údaje o katedře (povinné: c\_katedry, název)
6. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
7. ZAPIŠ změněné údaje o katedře (do Katedra)
8. VYPIŠ (Katedra byla úspěšně změněna)

#### **6.2.1.3.: Smazání katedry**

1. ZOBRAZ formulář s přehledem kateder
2. Správce zadá id\_katedry katedry pro smazání
3. PROVEĎ kontrolu existence uživatele, kterému je přiřazena katedra (je-li nalezen, katedra nelze smazat)
4. SMAŽ katedru s id\_katedry (z Katedra)
5. VYPIŠ (Katedra byla úspěšně smazána)

#### **6.2.1.4.: Přehled kateder**

1. NAJDI všechny katedry (v Katedra)
2. ZOBRAZ formulář s přehledem kateder

### **6.2.2.: Evidence oborů**

#### **6.2.2.1.: Nový záznam oboru**

1. ZOBRAZ formulář pro vložení nového oboru
2. Správce zadá údaje o novém oboru (povinné: název)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
4. ZAPIŠ nový obor (do Obor)
5. VYPIŠ (Obor byl úspěšně vložena)

#### 6.2.2.2.: *Upravení oboru*

1. ZOBRAZ formulář s přehledem oborů
2. Správce zadá id\_oboru oboru, u kterého se budou měnit údaje
3. NAJDI obor podle id\_oboru (v Obor)
4. ZOBRAZ formulář pro změnu údajů o nalezeném oboru
5. Správce zadá změněné údaje o oboru (povinné: název)
6. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
7. ZAPIŠ změněné údaje o oboru (do Obor)
8. VYPIŠ (Obor byl úspěšně změněn)

#### 6.2.2.3.: *Smazání oboru*

1. ZOBRAZ formulář s přehledem oborů
2. Správce zadá id\_oboru oboru pro smazání
3. PROVEĎ kontrolu existence pořadu, kterému je přiřazen obor (je-li nalezen, obor nelze smazat)
4. SMAŽ obor s id\_oboru (z Obor)
5. VYPIŠ (Obor byl úspěšně smazán)

#### 6.2.2.4.: *Přehled oborů*

1. NAJDI všechny obory (v Obor)
2. ZOBRAZ formulář s přehledem oborů

### 6.2.3.: *Evidence zemí původu*

#### 6.2.3.1.: *Nový záznam země původu*

1. ZOBRAZ formulář pro vložení nové země původu
2. Správce zadá údaje o nové zemi původu (povinné: jméno)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
4. ZAPIŠ novou zemi (do Země)
5. VYPIŠ (Země původu byla úspěšně vložena)

#### 6.2.3.2.: *Upravení země původu*

1. ZOBRAZ formulář s přehledem zemí původů
2. Správce zadá id\_země země původu, u které se budou měnit údaje
3. NAJDI zemi podle id\_země (v Země)
4. ZOBRAZ formulář pro změnu údajů o nalezené zemi původu
5. Správce zadá změněné údaje o zemi původu (povinné: jméno)
6. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
7. ZAPIŠ změněné údaje o zemi (do Země)
8. VYPIŠ (Země původu byla úspěšně změněna)

#### *6.2.3.3.: Smazání země původu*

1. ZOBRAZ formulář s přehledem zemí původů
2. Správce zadá id\_země země původu pro smazání
3. PROVEĎ kontrolu existence pořadu, kterému je přiřazena země původu (je-li nalezen, země původu nelze smazat)
4. SMAŽ zemi s id\_země (z Země)
5. VYPIŠ (Země původu byla úspěšně smazána)

#### *6.2.3.4.: Přehled zemí původů*

1. NAJDI všechny země původů (v Země)
2. ZOBRAZ formulář s přehledem zemí původů

### **6.2.4.: Evidence jazykových verzí**

#### *6.2.4.1.: Nový záznam jazykové verze*

1. ZOBRAZ formulář pro vložení nové jazykové verze
2. Správce zadá údaje o nové jazykové verzi (povinné: název)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
4. ZAPIŠ novou verzi (do Verze)
5. VYPIŠ (Jazyková verze byla úspěšně vložena)

#### *6.2.4.2.: Upravení jazykové verze*

1. ZOBRAZ formulář s přehledem jazykových verzí
2. Správce zadá id\_ver jazykové verze, u které se budou měnit údaje
3. NAJDI jazykovou verzi podle id\_ver (v Verze)
4. ZOBRAZ formulář pro změnu údajů o nalezené jazykové verzi
5. Správce zadá změněné údaje o jazykové verzi (povinné: název)
6. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
7. ZAPIŠ změněné údaje o verzi (do Verze)
8. VYPIŠ (Jazyková verze byla úspěšně změněna)

#### *6.2.4.3.: Smazání jazykové verze*

1. ZOBRAZ formulář s přehledem jazykových verzí
2. Správce zadá id\_ver jazykové verze pro smazání
3. PROVEĎ kontrolu existence pořadu, kterému je přiřazena jazyková verze (je-li nalezen, jazykové verze nelze smazat)
4. SMAŽ verzi s id\_ver (z Verze)
5. VYPIŠ (Jazyková verze byla úspěšně smazána)

#### *6.2.4.4.: Přehled jazykových verzí*

1. NAJDI všechny jazykové verze (ve Verze)
2. ZOBRAZ formulář s přehledem jazykových verzí

### **6.3.: Novinky**

#### *6.3.1.: Vložení novinky*

1. ZOBRAZ formulář pro vložení novinky
2. Správce zadá údaje o nové novince (povinné: text)
3. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
4. ZAPIŠ novou novinku (do Novinky)
5. VYPIŠ (Novinka byla úspěšně vložena)

#### *6.3.2.: Upravení novinky*

1. ZOBRAZ formulář s přehledem novinek
2. Správce zadá id novinky, u které se budou měnit údaje
3. NAJDI novinku podle id (v Novinky)
4. ZOBRAZ formulář pro změnu údajů o nalezené novince
5. Správce zadá změněné údaje o novince (povinné: text)
6. PROVEĎ kontrolu zadaných údajů (vyplnění povinných údajů)
7. ZAPIŠ změněné údaje o novince (do Novinky)
8. VYPIŠ (Novinka byla úspěšně změněna)

#### *6.3.3.: Smazání novinky*

1. ZOBRAZ formulář s přehledem novinek
2. Správce zadá id novinky pro smazání
3. SMAŽ novinku s id (z Novinky)
4. VYPIŠ (Novinka byla úspěšně smazána)

### **7.: Daemon**

#### *7.1.: Automatický daemon (automaticky spuštěná služba)*

1. NAJDI poslední spuštění daemona (v Nastavení)
2. JESTLIŽE je čas na spuštění kompletního daemona, PROVEĎ body 3, 4, 5, 6, 7, 8 a 9
3. NAJDI a SMAŽ nepotvrzené objednávky a její položky (stav = 0) (z Objednávka, Objednávka\_položka)
4. NAJDI objednávky, u kterých je nutnost upozornit na vyzvednutí, a POŠLI e-mail uživateli
5. NAJDI objednávky videokazet k zapůjčení, které nebyly vyzvednuty ve stanovený termín, a SMAŽ je včetně jejich položek (z Objednávka, Objednávka\_položka)
6. NAJDI objednávky k promítání, které již proběhly (z Objednávka, Objednávka\_položka) a PŘESUŇ je do historie (do H\_místnost, H\_místnost\_položka)

7. NAJDI výpůjčky, u kterých je nutno upozornit na vrácení, a POŠLI e-mail uživateli
8. NAJDI výpůjčky, u kterých je nutno zaslat upomínku, a POŠLI e-mail s upomínkou uživateli
9. AKTUALIZUJ poslední spuštění daemona

## 7.2.: Uživatelský daemon (služba spuštěna správcem)

1. NAJDI poslední spuštění daemona (v Nastavení)
2. JESTLIŽE je čas na spuštění kompletního daemona, PROVEĎ body 3, 4, 5, 6, 7, 8 a 9, JINAK PROVEĎ body 10, 11, 12, 13 a 14
3. NAJDI a SMAŽ nepotvrzené objednávky a její položky (stav = 0) (z Objednávka, Objednávka\_položka)
4. NAJDI objednávky, u kterých je nutnost upozornit na vyzvednutí, a POŠLI e-mail uživateli
5. NAJDI objednávky videokazet k zapůjčení, které nebyly vyzvednuty ve stanovený termín, a SMAŽ je včetně jejich položek (z Objednávka, Objednávka\_položka)
6. NAJDI objednávky k promítání, které již proběhly (z Objednávka, Objednávka\_položka) a PŘESUŇ je do historie (do H\_místnost, H\_místnost\_položka)
7. NAJDI výpůjčky, u kterých je nutno upozornit na vrácení, a POŠLI e-mail uživateli
8. NAJDI výpůjčky, u kterých je nutno zaslat upomínku, a POŠLI e-mail s upomínkou uživateli
9. AKTUALIZUJ poslední spuštění daemona
10. NAJDI a SMAŽ nepotvrzené objednávky a její položky (stav = 0) (z Objednávka, Objednávka\_položka)
11. NAJDI objednávky k promítání, které již proběhly (z Objednávka, Objednávka\_položka) a PŘESUŇ je do historie (do H\_místnost, H\_místnost\_položka)
12. AKTUALIZUJ poslední spuštění daemona
13. VYPIŠ chyby, pokud nějaké nastaly
14. VYPIŠ (Daemon byl úspěšně vykonán)



## 4. Návrh implementace

Tato kapitola popisuje důvody volby jednotlivých technologií vybraných pro implementaci Informačního systému Audiovizuálního centra VŠB-TUO. Popisuje, jak se bude program chovat věcně. Podkapitola 4.1 se zabývá výběrem technologií pro implementaci IS, podkapitola 4.2 se zaměřuje na popis způsobu autentizace uživatele do systému a podkapitola 4.3 popisuje rozdělení systému na moduly, práci s proměnnými a práci s databází.

Základní funkce použité pro implementaci informačního systému vychází z minispifikací funkcí popsanych v kapitole 3.2.3.

### 4.1 Výběr technologií pro implementaci

Implementovaný program by měl využívat jeden ze standardních formátů pro uložení dat, který je u velkých a rozšířených systémů řízení báze dat (dále jen SŘBD) používán, aby byl snadněji přenositelný. Také je kladen důraz na rychlost programovacího prostředku, který bude nad daty uloženými v databázi provádět dotazovací příkazy. Tento programovací prostředek by měl mít možnost využití jednoho ze základních „meta-jazyků“ pro manipulaci s daty (dále jen JMD), kterými jsou SQL (Structured Query Language – dále jen SQL) nebo QBE (Query By Example – dále jen QBE).

Pro implementaci Informačního systému Audiovizuálního centra VŠB-TUO se jeví z těchto JMD vhodnějším kandidátem jazyk SQL. Podporuje jej také databázový server MySQL, jenž byl zvolen jako prostředek pro ukládání dat. Je také vhodný po stránce zabezpečení databáze, i když neumožňuje šifrovat její obsah. Pro účely IS není šifrování obsahu databáze nutné.

Jelikož IS bude implementován ve formě webových stránek, které budou přístupné z intranetu (internetu), byl programovacím jazykem zvolen skriptovací jazyk PHP. Ten přímo podporuje řadu databází, včetně databáze MySQL, a proto se jeví jako vhodný kandidát pro implementaci Informačního systému. Podpůrným prostředkem při implementaci IS se stal jazyk JavaScript. Ten je interpretován na straně klienta. K zobrazení informací ve webovém prohlížeči slouží značkovací jazyk XHTML a pro formátování je použito kaskádových stylů CSS.

Pro zálohování a obnovení dat se bude taktéž využívat databáze MySQL. Technika obnovení dat vychází ze zápisu transakčního protokolu, tzn. že veškeré změny, jež se udály při manipulaci s databází, se zapisují do zvláštního souboru. Ten se využívá k vrácení transakcí nebo při obnovení databáze po havárii. Společně s instalací MySQL a PHP je vhodné instalovat i produkt phpMyAdmin, který umožňuje pohodlnou manipulaci s MySQL databází, umožňuje

vrácení a ukládání transakcí do souborů pro zálohování, umožňuje obnovu databáze po havárii a umožňuje provádět i zálohu celé databáze nebo jejích částí.

Veškeré technologie, které byly zvolené pro implementaci Informačního systému Audiovizuálního centra VŠB-TUO, jsou popsány v samostatné kapitole 6.

## 4.2 Autentizace uživatele

Autentizace neboli ověření identity uživatele při vstupu do informačního systému bude prováděna uživatelským jménem (loginem) a heslem.

Uživatelské jméno je složeno až ze šesti znaků (kombinace písmen a číslic) a odpovídá osobnímu číslu, které mají přiděleno všichni zaměstnanci a studenti VŠB-TUO.

Heslo nebude v databázi kvůli vyšší bezpečnosti uloženo. V rámci jazyka PHP bude vypočítán tzv. hash pomocí hashovací funkce MD5. Dostaneme z něj tak 16-ti bytový řetězec v hexadecimální formě. Tato transformace je pouze jednocestná. Znamená to, že z vypočteného MD5 hash hesla nemůžeme nikdy dostat zpět původní heslo. Tento přístup uložení hesla je na jednu stranu velice žádoucí, neboť při výpisu hesel z databáze nezískáme jejich původní hodnotu. Na druhou stranu však při opomenutí hesla nemůžeme jeho hodnotu poslat uživateli, ale musíme mu přidělit zcela nové heslo.

## 4.3 Rozdělení systému na moduly

Před vlastní implementací celého systému bylo vhodné položit si otázku, jaká bude adresářová struktura systému, jak se budou předávat data mezi jednotlivými skripty, jak bude oddělená aplikační logika od zobrazovacích dat apod. Veškeré tyto otázky budou zodpovězeny v této podkapitole.

### 4.3.1 Struktura adresářů systému

Adresářová struktura informačního systému je jednoduchá. V kořenovém adresáři se nacházejí skripty, jejichž vykonání lze spustit jejich zavoláním. Tyto skripty mají příponu *php* a nedělají nic jiného, než že inicializují aplikaci a volají třídu na její spuštění.

V adresáři *include* se nacházejí třídy, které vždy implementují akce nad jednou tabulkou v databázi. Každá z těchto tříd obsahuje metody od načtení vstupu od uživatele, přes vyhledání,



vložení nebo aktualizaci dat v databázi, až po vytvoření konkrétních dialogů, které je možno následně poslat uživateli k zobrazení.

Adresář *template* obsahuje v první řadě šablony (označeny příponou *tpl*), ze kterých se vytváří webová stránka k zobrazení dat. Obsahuje i veškeré obrázky vkládané do stránek. V neposlední řadě také obsahuje dva jazykové balíčky (XML soubory, anglický a český) s veškerými texty aplikace.

Obecný popis souborů, a co implementují, je uveden v příloze B. Detailní popis souborů je pak uveden v programátorské příručce na přiloženém CD.

### 4.3.2 Předávání a ověřování proměnných

Data se mezi skripty posílají a předávají pomocí metod GET a POST. Základním rozdílem mezi oběma metodami je způsob předávání dat. Pomocí metody GET se předávají v adrese požadavku na dokument nacházející se na serveru. Na rozdíl od metody POST, kde se předávají jako součást hlavičky HTTP. Data jsou pak uložena v patřičných shromaždištích, kde si je můžeme v PHP vyzvednout pomocí tzv. superglobálních proměnných `$_GET` a `$_POST`.

Pro monitorování chování a přiřazování vlastností uživateli přihlášenému do systému je využit objekt *Session* (tzv. sezení), který se vytváří na straně serveru. Je vždy společný pro jednu instanci klientského webového prohlížeče. Typickým příkladem využití tohoto objektu je zaznamenání informací o tom, zda uživatel již prošel ověřením a má přístup k privátní části aplikace. Tento typický příklad je využit i při implementaci Informačního systému Audiovizuálního centra VŠB-TUO. V PHP je k tomuto objektu přístup přes superglobální proměnnou `$_SESSION`.

Při ověřování dat, která jsou přijímána od uživatele, se řídím zásadou: „Nikdy nevěř tomu, co uživatel posílá.“ A proto jsou data vždy ověřována až na straně serveru. Sice nám jazyk JavaScript nabízí širokou škálu a možnost ověřování dat ještě před jejich odesláním, ale i tato skutečnost nemusí být stoprocentní a lze ji jednoduchým přepsáním stránky obejít.

Základním nosným kamenem celé aplikace jsou dvě proměnné předávané v adrese požadavku. Jedná se o proměnné *action* a *subaction*. *Action* určuje základní akci systému, která se bude vykonávat, a *subaction* upřesňuje, co se bude v aplikaci právě vykonávat.

### 4.3.3 Práce s databází

Databáze Informačního systému Audiovizuálního centra je navržena tak, aby nedocházelo k velkým redundancím vkládaných dat. Důležitou vlastností IS (ve většině případů) při odstraňování záznamů je tzv. logické mazání. Tzn. záznam se neodstraňuje fyzicky z databáze,

ale je mu přidělen příznak určující jeho neplatnost pomocí SQL příkazu UPDATE. Logické mazání souvisí hlavně s bezpečností, kdy mohou být důležitá data omylem smazána. Ta však lze v tomto případě zpětně obnovit. Využití vlastnosti logického mazání pro IS nastává v případě, že chceme vypsát historii výpůjček a promítání a tyto výpůjčky uskutečnil již smazaný uživatel.

Tabulky, ve kterých se očekává časté vkládání, upravování a mazání dat, obsahují pouze aktuální data, se kterými se pracuje. Mazané záznamy nejsou tedy v tabulkách odstraňovány logicky, ale jsou přesouvány do jiných tabulek. Ty slouží jako shromaždiště dat a nejsou již k dalšímu běhu aplikace důležité. Využívají se jen k zobrazení historie běhu aplikace.

## 5. Implementace

Tato kapitola se nezabývá úplným detailním popisem implementovaného Informačního systému Audiovizuálního centra, jejím obsahem je popis logiky implementace systému, popis několika typických formulářů a popis instalace, testování a provozu IS.

### 5.1 Popis implementace

Tato podkapitola popisuje logiku sestavení stránky, která se zobrazí uživateli, logiku vkládání dat od uživatele, logiku práce se šablonami a logiku vkládání textů v jednotlivých jazycích. Neslouží k detailnímu popisu implementace Informačního systému a popisu tříd a jejich metod. K detailnímu popisu je určena programátorská dokumentace, která je uložena na přiloženém CD.

Sestavení celé stránky počíná požadavkem na php skript s určitými parametry předávanými již zmiňovanou metodou POST nebo GET. Nejprve je vytvořena instance třídy *Gui*, která obstarává chod celé aplikace. Na ni je zavolána metoda *execute*, jež nejprve uloží hodnoty proměnných *action* a *subaction*. Podle hodnoty proměnné *action* jsou vyvolávány následující akce: *login* – pro přihlášení do IS, *logout* – pro odhlášení z IS, *news* – pro práci s novinkami, *user* – pro práci u uživatelskými údaji, *person* – pro práci s osobními údaji, *tables* – pro práci s číselníky, *option* – pro práci s nastavením aplikace, *program* – pro práci s pořady, *cassette* – pro práci s videokazetami, *place* – pro práci s umístěním pořadů na videokazetě, *order* – pro práci s objednávkami, *lend* – pro práci s výpůjčkami, *daemon* – pro práci s automatickou správou aplikace, *language* – pro změnu jazyka aplikace, *history* – pro práci s historií aplikace. Tyto akce vyvolají vždy instanci příslušné třídy a metodu *execute* této třídy, která vždy provede elementární funkci. Elementární funkce jsou popsány v kapitole 3.2.3 (minispecifikace funkcí). Tyto funkce vykonají ověření vstupních dat, nalezení, uložení, modifikaci či smazání dat z databáze; nakonec výsledek zasadí do své šablony, která je popsána souborem s příponou *tpl* v adresáři *template*, a vrátí jej ke zpracování instanci třídy *Gui*. Dále následují akce, které do výsledné stránky vloží komentáře chyb a zpráv, vloží příslušné texty z jazykového *xml* souboru a smažou komentáře, které se užívají k označení bloku v šabloně. Výsledná stránka se pošle uživateli do klientského webového prohlížeče.

Jak již bylo zmíněno, při vkládání dat do Informačního systému Audiovizuálního centra se celá aplikace řídí pravidlem: „Nikdy nevěř datům, která jsou posílána uživatelem.“ A proto jsou veškerá data testována a validována. Ze vstupních dat jsou odebrány netisknutelné znaky, odstraněny HTML tagy, které by mohly obsahovat škodlivý kód, a jsou doplněna zpětná


lomítka. Kontrolou také samozřejmě prochází veškerá data, které jsou předávána společně s požadavkem na PHP skript.

Při vytváření stránky je nedílnou součástí i podíl šablon. Každý soubor se šablonou je tematicky rozdělen podobně jako implementované třídy, tzn. že každá šablona se váže k jedné třídě. Šablona je rozdělena na bloky uzavřené mezi počáteční a koncovou značku a ty jsou identifikovány svým jménem. Při načtení celé šablony se nejprve odstraní bloky, které nejsou potřebné. Proměnné v bloku (označené takto: {proměnná\_šablony}) se nahradí svými skutečnými hodnotami získanými z databáze. Poslední a důležitou částí před výsledným zobrazením stránky je nutno v šabloně vyhledat nepřirazené proměnné a nahradit je skutečnými hodnotami z jazykového balíčku.

## 5.2 Popis typických formulářů

Tato podkapitola se zaměřuje na popis několika typických formulářů aplikace z hlediska uživatelského rozhraní. Nebude zde popisován zdrojový kód aplikace. K tomu je připravena programátorská příručka uložená na příloženém CD.

Informační systém má pouze privátní část. Jeho informace jsou před neregistrovanými uživateli skryty. Každý uživatel se musí nejprve do informačního systému přihlásit prokázáním správných údajů (uživatelského jména, tzv. login a hesla). Dialog pro přihlášení je zobrazen na obr. 5.1.



**Obr. 5.1** Dialog pro přihlášení uživatele

Po správném přihlášení uživatele slouží k navigaci celého programu tzv. menu IS. To má různé položky podle uživatelských práv. Menu aplikace je zobrazeno na obr. 5.2.

Menu	Menu
▶ Novinky	▶ Novinky
▼ Osobní údaje	▼ Osobní údaje
▶ Zobrazit	▶ Zobrazit
▶ Změnit heslo	▶ Změnit heslo
▼ Uživatelé	▼ Uživatelé
▶ Přehled	▶ Přehled
▶ Nový záznam	
▶ Obnovení smazaných	
▼ Videokazety	▼ Videokazety
▶ Přehled	▶ Přehled
▶ Nový záznam	
▼ Pořady	▼ Pořady
▶ Přehled	▶ Přehled
▶ Nový záznam	
▼ Objednávky	▼ Objednávky
▶ Přehled	▶ Přehled
▶ Vlastní	▶ Vlastní
▶ Objednat	▶ Objednat
▼ Výpůjčky	▼ Výpůjčky
▶ Přehled	▶ Přehled
▶ Vlastní	▶ Vlastní
▼ Historie	▼ Historie
▶ Přehled	▶ Přehled
▶ Vlastní	▶ Vlastní
▼ Systémové nastavení	
▶ Přehled	
▶ Daemon	
▶ Vložit novinku	
▶ Číselníky (tabulky)	
▶ Odhlásit [pil032]	▶ Odhlásit [pil033]

**Obr. 5.2** Menu aplikace: vlevo – menu správce, vpravo – menu zaměstnance

Pro vkládání nových údajů do IS je použit typický formulář, kde jsou povinné údaje označeny symbolem (\*). Veškerá vkládaná data jsou kontrolována na správnost. Jsou-li některé položky nesprávně vyplněny, je zobrazeno chybové hlášení a jejich vyplnění je vyžadováno znovu. Na obr. 5.3 je zobrazen formulář pro vložení nového pořadu do systému.

Přehledy dat jsou dalším typickým formulářem IS. Pro vytvoření tabulky s přehledem jednotlivých informací je potřeba nejprve vyplnit kritéria, podle kterých se vyhledají záznamy v databázi. Na obr. 5.4 je zobrazena tabulka s přehledem již nalezených záznamů. V pravé části této tabulky jsou vždy zobrazena grafická tlačítka, které slouží k vykonání akce s příslušným záznamem. Tlačítka jsou různá podle práv aktuálně přihlášeného uživatele.

K upravování záznamů slouží další typický formulář, který je v základu stejný s formulářem pro vkládání nových dat a pravidla pro jeho vyplnění jsou tedy stejná.

Vložení pořadu	
MDT:	<input type="text"/> (*)
Název pořadu:	<input type="text"/> (*)
Režisér:	<input type="text"/>
Výrobce:	<input type="text"/>
Obor:	--- žádný --- (*)
Země původu:	--- žádná --- (*)
Rok výroby:	<input type="text"/>
Způsob získání:	<input type="text"/>
Jazyková verze:	--- žádná --- (*)
Délka v minutách:	<input type="text"/> (*)
Popis:	<div style="border: 1px solid black; height: 100px;"></div>
Datum vložení:	14.04.2005 (*)
(*) povinné údaje	
<input type="button" value="Vložit"/> <input type="button" value="Smazat"/>	

Obr. 5.3 Formulář pro vložení nového pořadu

▼ **Vyhledávací kritéria**

- ▶ ID pořadu: 12
- ▶ Název pořadu:
- ▶ MDT:
- ▶ Obor:
- ▶ Jazyková verze:

#	ID	Název pořadu	MDT	Obor	Jazyk	Délka	
1.	12	A WEEK BY THE SEA	80.11	X	A	32	<input type="button" value="🔍"/> <input type="button" value="📄"/> <input type="button" value="🗑️"/> <input type="button" value="✕"/>

Třídit podle:

Obr. 5.4 Formulář s přehledem vyhledaných pořadů

Základním kamenem celé aplikace je objednávání videokazet a jejich půjčování. Formulář pro vytvoření objednávky, tzn. zadání základních údajů na kdy se kazeta objednává či kdy se má uskutečnit promítání, je typickým formulářem pro vložení nového záznamu. Takto vytvořená objednávka ještě není zcela hotová. Do objednávky se musí přidat videokazety, které se vkládají z přehledu videokazet kliknutím na příslušnou ikonu. Vyobrazení právě probíhající objednávky je vidět na obr. 5.5.

Právě si objednáváte	
<b>Typ objednávky:</b>	Kazety
<b>Datum počátku výpůjčky:</b>	12.12.2005
<b>Datum konce výpůjčky:</b>	13.12.2005
<b>Objednané kazety:</b>	1 
Nezapomeňte tuto objednávku potvrdit. Nepotvrzená objednávka bude smazána!	
<input type="button" value="Potvrdit objednávku"/> <input type="button" value="Zrušit objednávku"/>	

**Obr. 5.5** Dialog s informacemi právě probíhající objednávky

Vypůjčení objednaných videokazet proběhne kliknutím na příslušnou ikonu z přehledu objednávek. Stejným způsobem je realizováno vrácení vypůjčených videokazet, tzn. kliknutím na příslušnou ikonu z přehledu výpůjček.

Nedílnou součástí IS je automatická služba nazvaná *daemon*, která slouží k udržování databáze IS a upozorňuje uživatele formou elektronické pošty. Tento *daemon* je spuštěn automaticky na serveru vždy v nastavený čas (toto nastavení musí provést správce serveru) a vykoná následující akce:

- 1) Nalezení a smazání nepotvrzených objednávek.
- 2) Nalezení objednávek, u kterých je nutnost upozornit na vyzvednutí, a upozornění uživatele.
- 3) Nalezení objednávek, které nebyly vyzvednuty ve stanovený termín, a jejich smazání.
- 4) Nalezení objednávek promítání, které již proběhly, a jejich přesunutí do historie.
- 5) Nalezení výpůjček, u kterých je nutnost upozornit na vrácení, a upozornění uživatele e-mailem.
- 6) Nalezení výpůjček, u kterých je nutnost zaslat upomínku, a posílání upomínky uživateli e-mailem.

Daemon může být také spouštěn uživatelem, který má k tomu příslušná práva. Ovšem kompletní vykonání akcí je možno provést jednou denně. Není-li tato podmínka splněna, *daemon* se spustí jen částečně a vykoná akce číslo 1 a 4.

Další a podrobnější popis formulářů a použití aplikace je uveden v uživatelské příručce, která se nachází na přiloženém CD.

## 5.3 Instalace

Instalace Informačního systému je důležitou součástí inicializace celé aplikace. K tomu je také připraven PHP skript, který tuto velmi důležitou část usnadní.

### 5.3.1 Instalace aplikace

Po nahrání všech souborů na server, kde bude aplikace běžet, se při každém přístupu kontroluje, zda v adresáři *include* existuje soubor *dbinclude.inc*, který obsahuje údaje o přístupu do databáze. Pokud tento soubor neexistuje, znamená to jediné. Aplikace ještě nebyla nainstalována. Je tedy spuštěn PHP skript *install.php*. Tento skript vykoná následující akce:

- 1) Zobrazí stránku pro zadání údajů nutných pro provoz aplikace, pro přístup do databáze a pro vytvoření jednoho uživatele (správce).
- 2) Po zadání potřebných údajů a jejich potvrzení zkontroluje instalační skript jejich správnost a ověří správnost přístupu do databáze.
- 3) Jsou-li všechny náležitosti splněny, vytvoří se soubor *dbinclude.inc* s informacemi pro přístup do databáze, vytvoří se tabulky v databázi a je vložen první uživatel.
- 4) Po nainstalování je zobrazena stránka pro přechod do samotné aplikace a je možno využívat již nainstalovanou a plně funkční aplikaci bez naplněné databáze.
- 5) Pro kompletní instalaci aplikace je nutno ještě zprovoznit automatickou funkci aplikace nazvanou „daemon“. Na serveru je tedy nutno spustit službu, která bude běžet nepřetržitě. Jednou denně (nejlépe v brzkých ranních hodinách, např. 2:00) za nejslabšího provozu aplikace se spustí PHP skript *daemon.php* (nachází se v kořenovém adresáři aplikace). Tento skript si již obstará veškeré potřebné prostředky. Takovou službu na linuxové platformě zajišťuje např. daemon „Cron“, na platformě Windows zase „Plánovač úloh“. Pro nastavení automatického spuštění skriptu *daemon.php* v určitou hodinu kontaktujte správce svého webového serveru.



### 5.3.2 Konverze dat a naplnění databáze

Jelikož se přechází ze staršího systému na nový, je třeba přenést i data, která byla za dobu používání původní aplikace vytvořena. Z tohoto důvodu byla vytvořena aplikace v PHP (resp. třída), která tato data zkonvertuje. Data jsou konvertována z DBF souborů pomocí tohoto programu a jsou vytvářena nová data v SQL příkazech. Ty jsou použitelné nad jakoukoliv databází.

Bylo tedy nutno konvertovat data do všech číselníků, data videokazet, pořadů a jejich umístění na kazetách. Konverzní program samozřejmě kontroluje duplicitu výskytu nějakého záznamu a tyto záznamy pak vynechává.

Již zkonvertovaná data je pak možno naplnit do databáze. Nová data jsou ve formátu SQL příkazů, proto se stačí přepnout do uživatelského prostředí, které běží nad databázovým serverem, a vykonat výsledné SQL příkazy.

## 5.4 Testování a provoz IS

Informační systém byl v počátku testován na serveru <http://www.webzdarma.cz>. Většina jeho služeb je poskytována zdarma. Zde IS běžel nad databází MySQL verze 4.0.24. Prostředí phpMyAdmin je ve verzi 2.6.0-pl3. Testovací verze Informačního systému Audiovizuálního centra je přístupná na adrese <http://pil032.wz.cz>.

Ostrá verze Informačního systému běží na školním serveru Barborka. Je přístupná na adrese **!!!VYPUŠTĚNO!!!**. Zde IS běží nad databází MySQL 4 a PHP 4.



## 6. Použité internetové technologie

Tato kapitola popisuje podrobněji internetové technologie, které byly využity ke tvorbě Informačního systému Audiovizuálního centra VŠB-TUO. V současné době je mnoho dostupných vývojových technologií, pomocí nichž se vytvářejí on-line informační systémy, redakční systémy, internetové obchody a diskusní fóra.

Cílem této kapitoly je poskytnout přehlednější orientaci v prostředcích využívaných k tvorbě internetových a on-line aplikací, zejména pak při tvorbě Informačního systému Audiovizuálního centra VŠB-TUO.

### 6.1 World Wide Web

World Wide Web (zkráceně WWW nebo web), ve volném překladu „celosvětová pavučina“, je označením pro aplikace využívající ke komunikaci internetového protokolu HTTP. Je tím myšlena soustava propojených hypertextových dokumentů, jenž jsou uloženy na internetových serverech. Ty jsou propojeny v celosvětovou počítačovou síť.

Protokol, který se využívá ke komunikaci a transakcím na webu, se jmenuje Hypertext Transfer Protokol (zkráceně HTTP). Tento internetový protokol je určený pro výměnu hypertextových dokumentů ve formátu HTML. Protokol funguje způsobem dotaz-odpověď. Uživatel (pomocí programu, obvykle internetového prohlížeče, např. Mozilla Firefox, MS Internet Explorer atd.) pošle serveru dotaz ve formě čistého textu obsahujícím označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server poté odpoví pomocí několika řádků textu obsahujících výsledek dotazu (zda se dokument podařilo najít, jakého je typu atd.), za kterými následují data samotného požadovaného dokumentu.

Jedním ze základních stavebních kamenů pro vytváření hypertextových dokumentů je značkovací jazyk HTML (Hypertext Markup Language). Dokumenty musí být v rámci celosvětové počítačové sítě jednoznačně identifikovatelné. Takovou možnost nabízí technologie URL (Uniform Resource Locator), která v zásadě definuje webovou adresu dokumentu.

Hypertextové dokumenty umístěné na webových serverech nabízí pouze statický obsah webových stránek. Tento přístup již není v dnešní době zcela vyhovující a vzniká tak potřeba generovat obsah hypertextových dokumentů „za běhu“, tzn. dynamicky vytvářet jejich obsah s využitím databází. Tuto možnost nám nabízí řada technologií, jako např. CGI, JavaScript, PHP, JSP nebo ASP.

## 6.2 Web server

Webový server lze popsat jako démona (službu) HTTP, který přijímá klientské požadavky na zdroje. Provádí základní rozbor požadavku pro zjištění, zda zdroj existuje, a následně jej odevzdává aplikační logice ke zpracování. Jakmile aplikační logika vrátí odpověď, vrací ji démon HTTP zpět klientovi.

Nejrozšířenějšími webovými servery jsou ISS firmy Microsoft a Apache HTTP Server od Apache Software Foundation. Hlavním důvodem rozšířenosti a oblíbenosti Apache HTTP Serveru je jeho kvalitní zpracování, rychlost při zpracovávání požadavků a použití zdarma se všemi dostupnými funkcemi. Tento server je přístupný jak pro platformu Windows, tak i pro platformu Linux.

## 6.3 Databázové systémy

Databázový systém je softwarový balík (množina komponent), jehož úlohou je ukládání, manipulace, vyhledávání a správa dat v databázi.

Databáze je množina datových souborů, uchovávající data o nějakém uceleném úseku reality.

Databázovým systémem obvykle myslíme programový systém (prázdný, bez datových a aplikačních souborů), jenž umožňuje definovat datové struktury a datové soubory, řeší fyzické uložení dat na magnetickém médiu počítače (obvykle na pevném disku), umožňuje manipulaci s daty a formátování vstupních i výstupních informací. Nazýváme jej systém řízení báze dat (SŘBD). Většina dnes používaných SŘBD implementuje relační datový model.

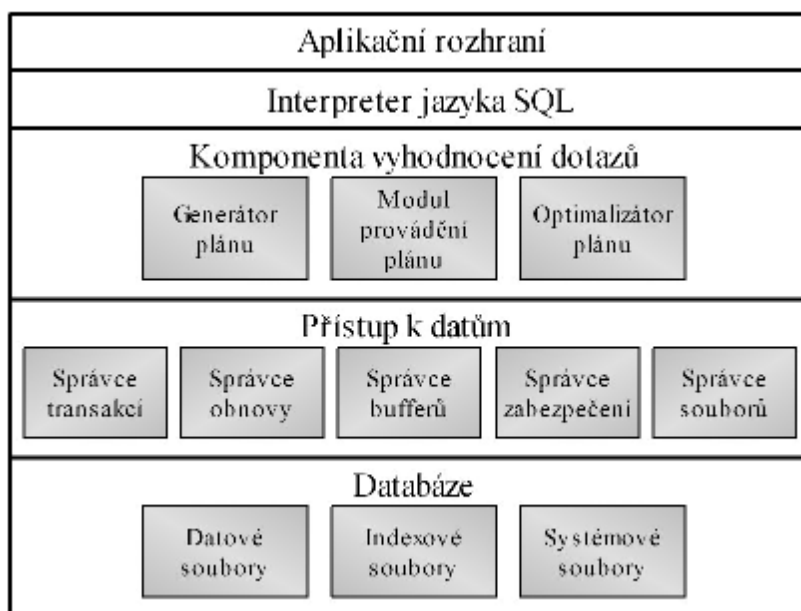
Zjednodušený pohled na architekturu typického databázového systému vidíme na obr. 6.1.

Typický databázový systém se tedy skládá z několika komponent:

- **Aplikační rozhraní**, které obsahuje knihovny pro komunikaci s databázovým systémem. Většina databázových systémů má jednoduchý interpret z příkazového řádku, který často předává požadavky zapsané z klávesnice do databázového systému a zobrazuje odpovědi. Ve webové databázové aplikaci je tento interpret příkazového řádku nahrazen knihovnou funkcí, která je součástí skriptovacího jazyka.
- **Interpret jazyka SQL** je lexikálním analyzátozem, který kontroluje syntaxi všech příchozích příkazů s dotazy a převádí je do interní reprezentace.
- **Komponenta vyhodnocení dotazů** generuje různé plány provádění (vyhodnocení) dotazů na základě databázových statistik a jejich vlastností.

- **Přístup k datům** je komponentou, která obsahuje moduly pro zajištění přístupu k datům uloženým na disku.

**Databáze** je již komponenta, která obsahuje samotná fyzická data uložená v datových souborech. Data také obsahují indexové soubory, které se využívají pro rychlejší přístup k datům, a souhrnné databázové a systémové statistiky, využívané zejména pro generování plánů dotazu a jejich optimalizaci.



**Obr. 6.1** Architektura typického databázového systému

Mezi SŘBD (databázový systém), který je velmi populární a využívaný, patří MySQL Server. Svou popularitu si získal hlavně díky své dostupnosti a možnosti využití zdarma jako Open Source/Free Software šířený pod licencí GNU/GPL (<http://www.gnu.org/licenses/>). MySQL Server je podrobněji popsán v následující kapitole 6.3.1.

Nicméně existují ještě komerční SŘBD, které jsou pro využití informačním systémem Audiovizuálního centra VŠB-TUO cenově nedostupné (pro využití dostačuje již zmíněný MySQL Server), jako například MS SQL, Oracle SQL Server, 602 SQL Server apod.

### 6.3.1 MySQL Server

MySQL Server je jedním z programových SŘBD střední velikosti; má většinu funkcí velkého databázového systému a dokáže spravovat i velké objemy dat. Je to velmi rychlý

databázový systém založený na jazyku SQL. Součástí distribuce MySQL Serveru je také jednoduchý interpret příkazů, který se využívá ke komunikaci s databázovým systémem pomocí příkazů jazyka SQL. Příkazy se mohou interpretovat z příkazové řádky nebo lze také využít pohodlnějšího a příjemnějšího grafického prostředí např. programu MySQL Control Center nebo phpMyAdmin. V těchto programech uživatel nemusí znát syntaxi jazyka SQL a přesto může pohodlně spravovat databázi.

MySQL Server má také svá omezení, která je nutno vzít na vědomí při vytváření databázových aplikací na něm založených. Mezi tato omezení patří podpora vnořených dotazů. Ta by již měla být vyřešena ve verzi 4.1. Dalším omezením je podpora transakčního zpracování. MySQL Server využívá k ukládání tabulky dvojího typu. Prvním a zároveň implicitním typem jsou tabulky MyISAM (beztransakční bezpečný typ), které podporují pouze uzamykání celých tabulek. Dalším typem tabulek, které podporují uzamykání různých částí, jsou tabulky typu BDB (Berkeley Database), InnoDB nebo Gemini. Chceme-li plnohodnotně využívat transakce a uzamykání různých částí tabulek, je nutno využít tento typ tabulek. Jeho popisu se nebudeme nadále věnovat, protože při implementaci informačního systému Audiovizuálního centra VŠB-TUO bylo použito tabulek typu MyISAM.

V současnosti je MySQL velmi rozšířený a dostupný jak pro platformu Windows, tak i pro platformu Linux.

### 6.3.2 Jazyk SQL

Jazyk SQL (Structured Query Language, strukturovaný dotazovací jazyk) je standardním jazykem pro komunikaci s relačními databázemi. Byl původně navržen firmou IBM jako dotazovací jazyk, postupem doby se z něj vyvinul plnohodnotný jazyk, jenž umožňuje správu dat v databázi. Jazyk SQL tedy mimo dotazovacích příkazů obsahuje i příkazy potřebné pro definici, modifikaci a rušení databází, tabulek, indexů a pohledů, jež jsou součástí jazyka pro definici dat (dále jen JDD), a příkazy pro ukládání, modifikaci a rušení dat z databáze, které jsou součástí jazyka pro manipulaci s daty (dále jen JMD).

JDD je reprezentován základními příkazy CREATE DATABASE (pro vytvoření nové databáze), DROP DATABASE (pro odstranění existující databáze), CREATE TABLE (pro vytvoření tabulky v existující databázi), ALTER TABLE (pro modifikaci struktury tabulky), RENAME TABLE (pro přejmenování tabulky), DROP TABLE (pro zrušení tabulky včetně její definice), CREATE INDEX (pro vytvoření indexu), DROP INDEX (pro zrušení indexu).

JMD obsahuje základní příkazy INSERT (pro vkládání údajů do tabulky v databázi), UPDATE (pro modifikaci údajů tabulky v databázi) a DELETE (pro odstranění údajů tabulky v databázi).

Dotazovací jazyk reprezentuje mohutný příkaz SELECT.

## 6.4 Technologie pro vývoj internetových aplikací

### 6.4.1 PHP

PHP (PHP: Hypertext Preprocessor, PHP: hypertextový preprocesor) je mezipatformním (nezávislým na platformě) skriptovacím jazykem zpracovávaným na straně serveru. Znamená to, že pracuje uvnitř dokumentu HTML a propůjčuje mu tak možnost dynamického generování obsahu. Kód jazyka PHP je odlišen v HTML dokumentu vkládáním mezi značky `<?php ?>` (případně `<? ?>`).

Syntaxe jazyka PHP obsahuje konstrukce, které jsou důvěrně známé programátorům objektově orientovaných nebo procedurálních jazyků, avšak oproti těmto jazykům je beztypovým a interpretovaným. Ovládá-li programátor některý z jazyků, jako jsou C, C++, Java, Pearl, Pascal nebo Visual Basic, bude pro něj základní konstrukce PHP hračkou. PHP navíc obsahuje řadu užitečných funkcí, které usnadňují práci, a nabízí i základní podporu objektově orientovaného programování (dále jen OOP).

Jednou z příjemných vlastností jazyka PHP je jeho databázová konektivita. Podporuje tak řadu databází pomocí nativních ovladačů (např. MySQL, Oracle, Sybase, mSQL atd.). Také umožňuje přístup k databázím pomocí rozhraní ODBC.

### 6.4.2 JavaScript

JavaScript je interpretovaný, multiplatformní programovací jazyk se základními objektově orientovanými schopnostmi. Univerzální jádro jazyka bylo vloženo do webových prohlížečů a rozšířeno přidáním objektů reprezentující okno prohlížeče a jeho obsah. Tato *klientská* (v dnešní době více oblíbená) verze JavaScriptu umožňuje vložit do webových stránek proveditelný obsah. Stránky se tak stávají dynamickými – mohou obsahovat nejrůznější programy, které komunikují s uživatelem, řídí prohlížeč, či vytváří obsah HTML. Při práci skriptu není potřeba kontaktovat server, veškerou práci skriptu totiž zajišťuje sám prohlížeč.

Jádro jazyka Javascript syntakticky připomíná C++ a Javu. Avšak syntaxí podobnost končí. JavaScript je jazyk beztypové kontroly, což znamená, že proměnná nemusí mít specifikovaný typ, a je čistě interpretovaný.

### 6.4.3 HTML, XHTML, XML

HTML (HyperText Markup Language), značkovací jazyk pro hypertext, je základním jazykem pro vytváření stránek v systému World Wide Web. Je odvozen ze staršího univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language), který byl určován vývojem webových prohlížečů, jež zpětně ovlivňovaly definici jazyka. Vývoj jazyka HTML byl ukončen verzí 4.01. Dalším pokračováním vývoje technologie pro weby se stal jazyk XHTML.

XHTML (eXtensible HyperText Markup Language) je v podstatě pokračovatelem jazyka HTML. Je to varianta HTML, která plně vyhovuje novějším požadavkům, jako je kompatibilita s XML.

XML (eXtensible Markup Language), rozšiřitelný značkovací jazyk, který byl vyvinut konsorciem W3C. Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí. Prezentace dokumentu (vzhled) se potom definuje přiloženým stylem.

Syntaxe všech tří jazyků je obdobná. Obsahují buď neprázdné elementy určené párovými tagy (např.: <body>, </body>) nebo prázdné elementy určené tagem „prázdný element“ (např.: u HTML <br>, u XHTML <br />). Hlavním rozdílem mezi syntaxí jazyka HTML a XHTML (XML) je ten, že vnořené tagy se u XHTML nesmí překrývat a všechny párové tagy musí mít jak počátek, tak i konec.

Vytvořený HTML, XHTML nebo XML dokument můžeme ověřit, zda odpovídá standardům jazyka, ve kterém je napsaný, na stránkách konsorcia W3C.

### 6.4.4 CSS

CSS (Cascading Style Sheets, kaskádové styly) je kolekce metod pro grafickou úpravu webových stránek. Představuje způsob pro zajištění jednotného vzhledu webových stránek. Užitím kaskádových stylů můžeme jednoduše, rychle a efektivně měnit vzhled stránek. CSS v dnešní době rozlišuje několik verzí: CSS1, CSS2 a CSS3. Prohlížeči nejpodporovanější verzí CSS je verze CSS2, proto je taky nejuzívanější.

V dnešní době je CSS doporučován jako standard pro vývoj a stylování HTML dokumentů.



### 6.4.5 Šablony

Technika šablon se využívá hlavně k oddělení logiky aplikace od její reprezentace, což je vždy hlavním důvodem jejího použití.

Základní myšlenkou užití šablon je jediná stránka, která se skládá z více logických částí. Tyto části jsou pojmenované a nazývají se šablony. Každá šablona může obsahovat více proměnných, které jsou nahrazovány buď čistým textem (tj. HTML), nebo výstupem jiných šablon. Proměnné v šablonách jsou označeny ve tvaru {NÁZEV} pro zřetelné odlišení od HTML kódu. V šablonách se také vyskytují tzv. bloky, které strukturují šablonu. Bloky se uzavírají mezi dvojicí značek (využívající komentář HTML kódu), kde počáteční značkou je `<!-- BEGIN: název_bloku -->` a koncovou značkou je `<!-- END: název_bloku -->`.

K vytvoření stránky pomocí šablony vedou následující čtyři kroky: načtení šablony ze souboru, přiřazení hodnot proměnných šablon, analýza šablon, výstup výsledku.

Šablony jsou tedy základním stavebním kamenem k dosažení lepší přehlednosti reprezentace aplikace. Navíc nám tato technika umožňuje pohodlné přepsání vzhledu aplikace bez nutnosti zasahovat do programového kódu.



## 7. Závěr

Cílem této diplomové práce bylo vytvořit novou verzi Informačního systému Audiovizuálního centra VŠB-TUO, která bude přístupna na školním intranetu případně i na internetu.

Nová verze Informačního systému byla navržena plně k potřebám Audiovizuálního centra s možností vytváření objednávek videokazet a promítání přímo uživateli bez nutnosti někam volat či chodit.

Se systémem lze pracovat všude tam, kde je k dispozici běžný webový prohlížeč, jakými jsou Mozilla, Mozilla FireFox, MS Internet Explorer, Opera, Netscape Navigátor, a kde je k dispozici připojení ke školnímu intranetu. Se systémem může pracovat každý uživatel, ať zaměstnanec školy nebo její student, který má právo k přístupu do privátní sekce aplikace. Bez splnění této podmínky nelze s Informačním systémem pracovat.

Vytvořený informační systém splňuje všechny požadavky Audiovizuálního centra, které byly zadány a požadovány na nové verzi IS. Hlavním požadavkem byla možnost víceuživatelského přístupu s různými právy uživatelů. Dodržuje také některé moderní metody při vytváření webových aplikací, jako je využití šablon, dodržení XHTML a CSS standardů apod. Díky tomu se naskýtá možnost jednoduchého přepsání vzhledu aplikace bez nutnosti zasahovat do programového kódu.

Další osud a vývoj aplikace vidím v možnosti dopsání funkcí Informačního systému, které vzniknou jako požadavek při jejím používání a umožní ještě více usnadnit práci uživatelům tohoto IS.

Tvorba tohoto informačního systému byla pro mne velikou zkušeností, při níž jsem si ověřil znalosti programování a vytváření webové aplikace a získal jsem další znalosti hlavně při tvorbě šablon, XHTML, CSS a vytváření PHP skriptů spolupracujících s databází MySQL. V neposlední řadě byla pro mne obzvlášť přínosná zkušenost při komunikaci se zadavatelem. Cílem bylo tedy vytvořit novou verzi Informačního systému Audiovizuálního centra VŠB-TUO. Věřím, že byl tento cíl dosažen ke spokojenosti zadavatele.

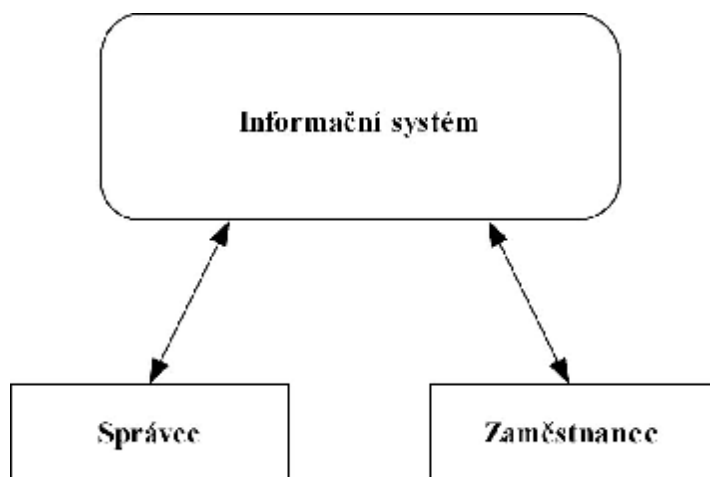


# Literatura

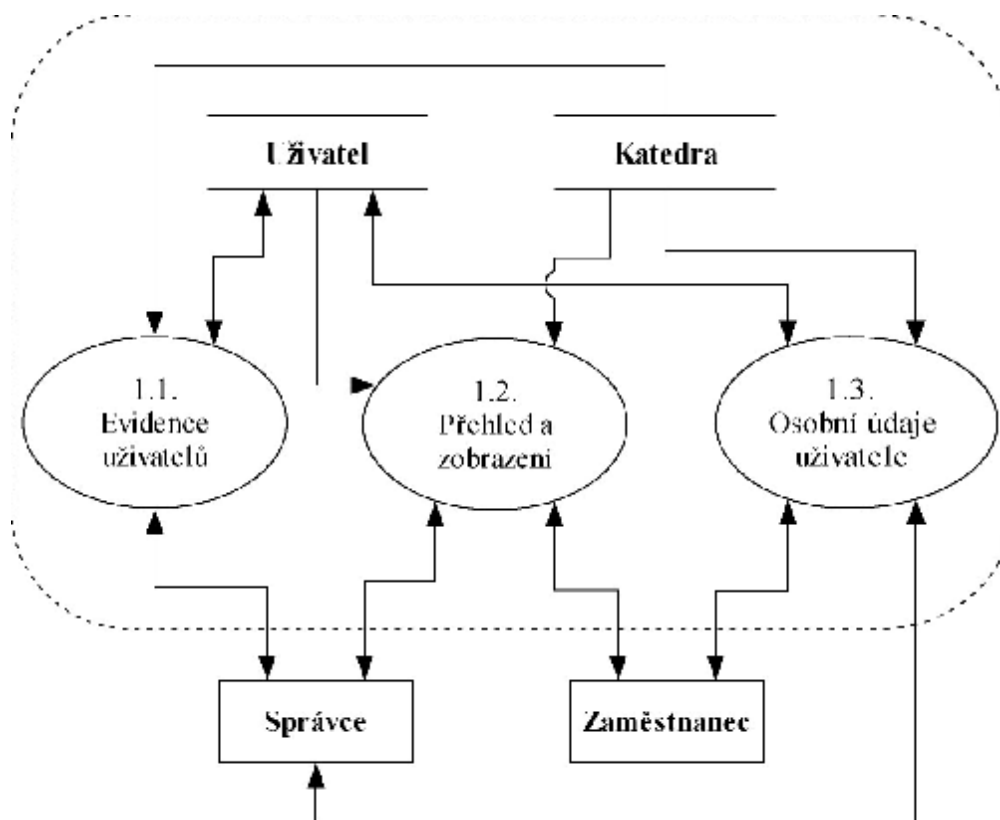
- [1] Castagnetto Jesus, Rawat Harish, Schumann Sascha, Scyllo Chris, Veliath Deepak, *Programujeme PHP profesionálně*, Druhé vydání, Computer Press, Praha, 2002, ISBN 80-7226-310-2
- [2] Scambray Joel, Shema Mike, *Hacking bez tajemství: Webové aplikace*, První vydání, Computer Press, Brno, 2003, ISBN 80-7226-769-8
- [3] Šarmanová Jana, *Teorie zpracování dat*, VŠB-TU, Ostrava, 1997
- [4] Williams Hugh E. a Lane David, *PHP a MySQL: Vytváříme webové databázové aplikace*, První vydání, Computer Press, Praha, 2002, ISBN 80-7226-760-4
- [5] Bakken Stig Sæther, Aulbach Alexander, Schmid Egon, Winstead Jim, Wilson Lars Torben, Lerdorf Rasmus, Zmievski Andrei, Ahto Jouni, *Manuál PHP*, vydání 22-04-2004, <http://www.php.net/download-docs.php>, 2004
- [6] Janovský Dušan, *CSS: Kaskádové styly*, <http://www.jakpsatweb.cz/css>, 2005
- [7] Janovský Dušan, *JavaScript: Návod na použití jazyka*, <http://www.jakpsatweb.cz/javascript>, 2005
- [8] Kosek Jiří, *Dynamické HTML*, <http://www.kosek.cz/clanky/dhtml/index.html>, 1998
- [9] Wikipedie: Otevřená encyklopedie, *HTML*, <http://cs.wikipedia.org/wiki/HTML>, 2004
- [10] Wikipedie: Otevřená encyklopedie, *HTTP*, <http://cs.wikipedia.org/wiki/HTTP>, 2004
- [11] Wikipedie: Otevřená encyklopedie, *XHTML*, <http://cs.wikipedia.org/wiki/XHTML>, 2004
- [12] Wikipedie: Otevřená encyklopedie, *XML*, <http://cs.wikipedia.org/wiki/XML>, 2005
- [13] Wikipedie: Otevřená encyklopedie, *World Wide Web*, [http://cs.wikipedia.org/wiki/World\\_Wide\\_Web](http://cs.wikipedia.org/wiki/World_Wide_Web), 2004



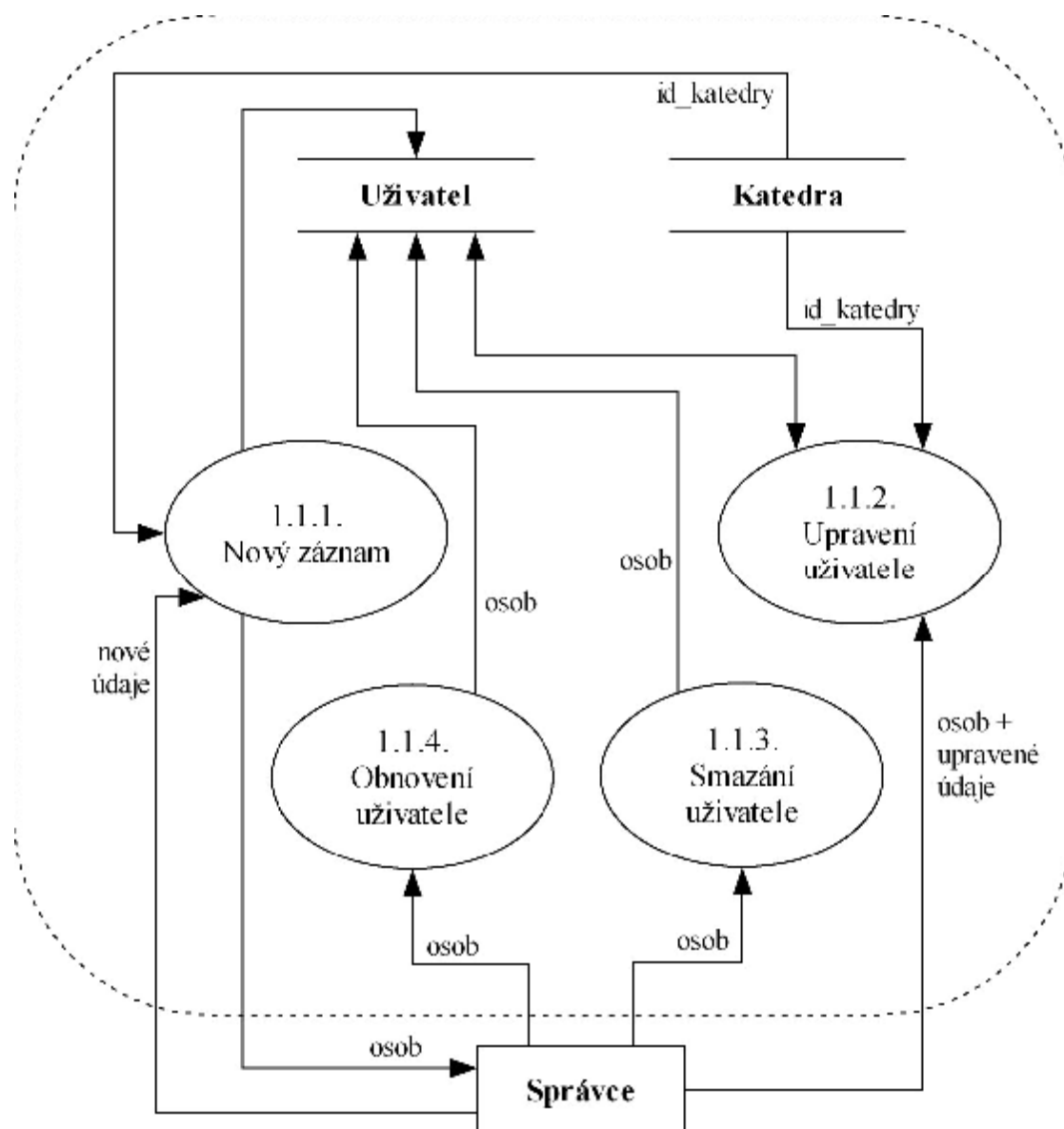
## Příloha A – DFD diagramy



Obr. A-0 Kontextový diagram

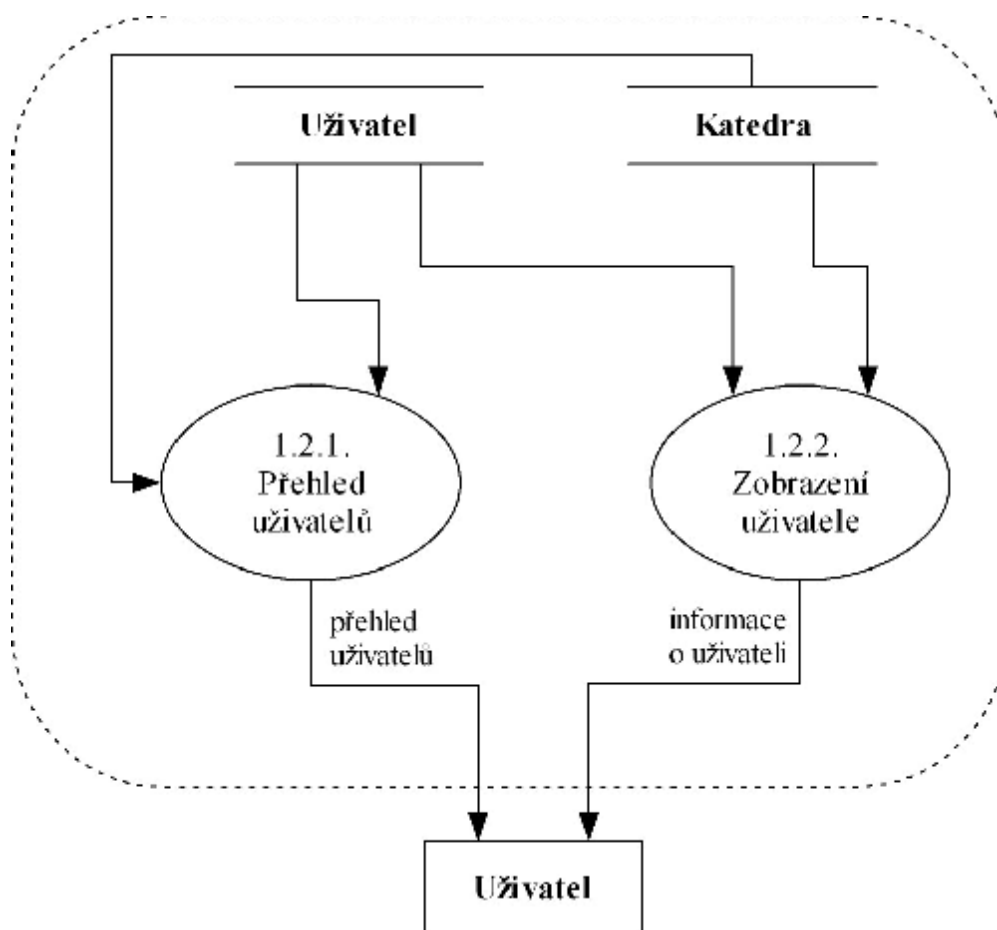


Obr. A-1 DFD 1. Evidence uživatelů

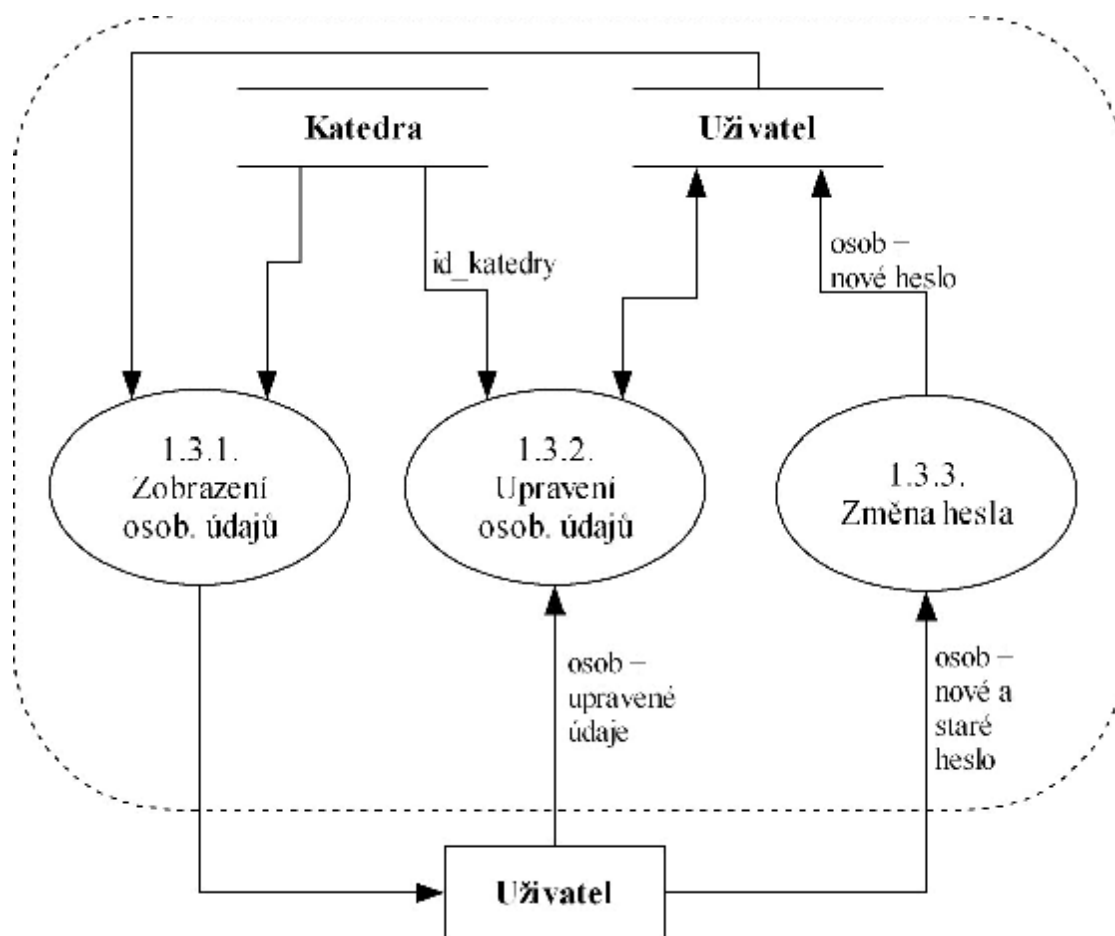


**Obr. A-2** DFD 1.1. Evidence uživatelů

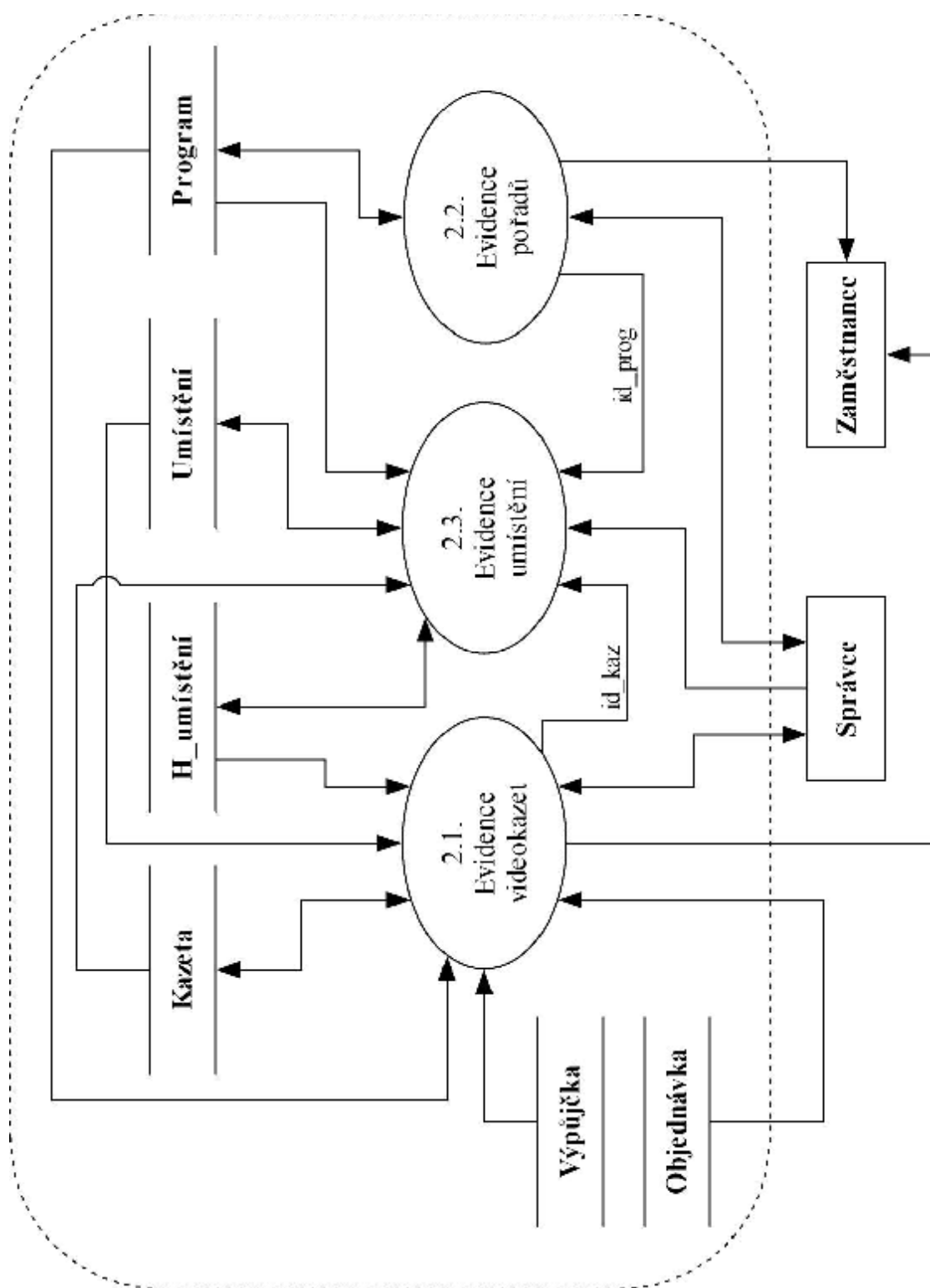




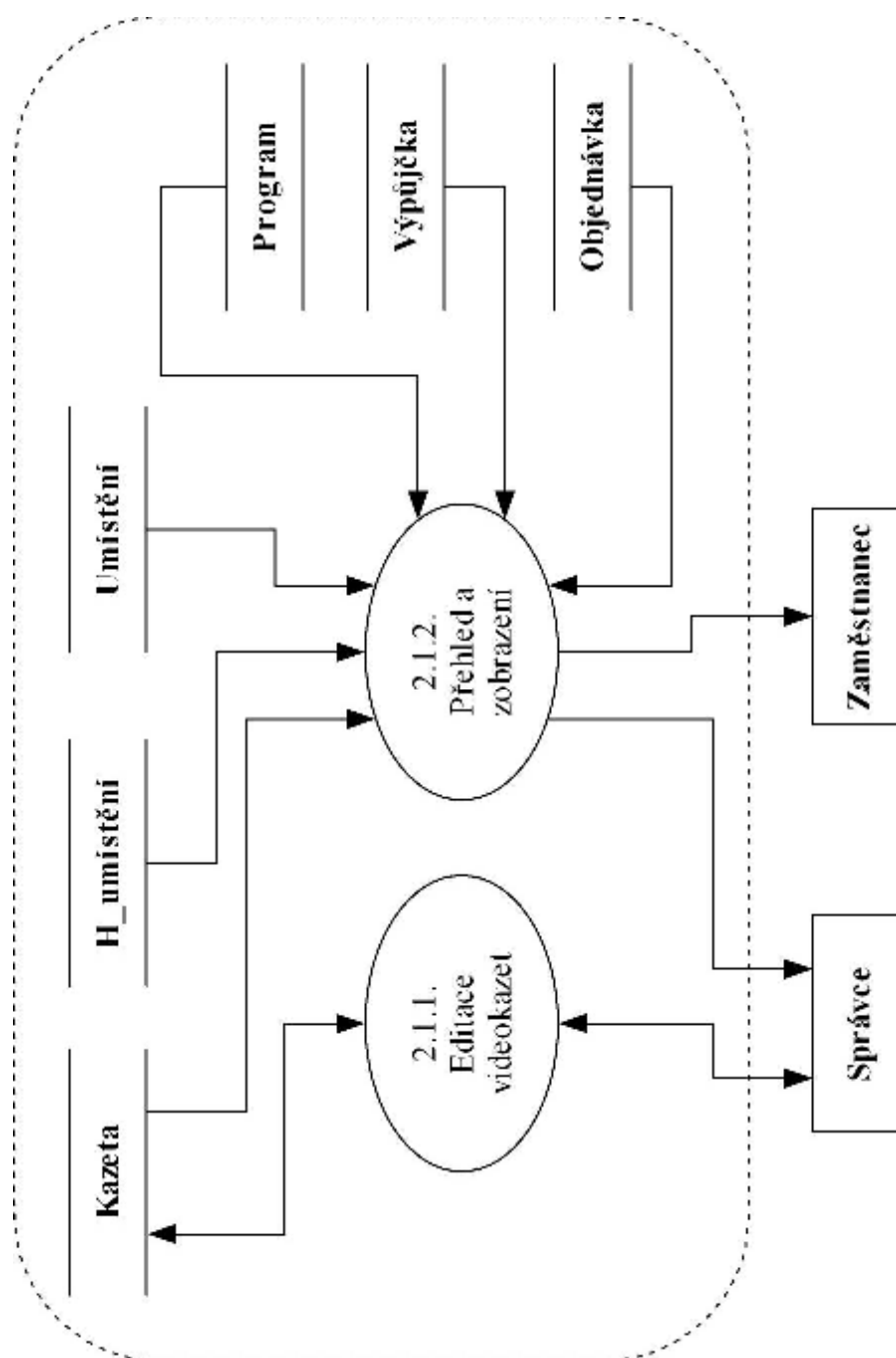
Obr. A-3 DFD 1.2. Přehled a zobrazení uživatelů



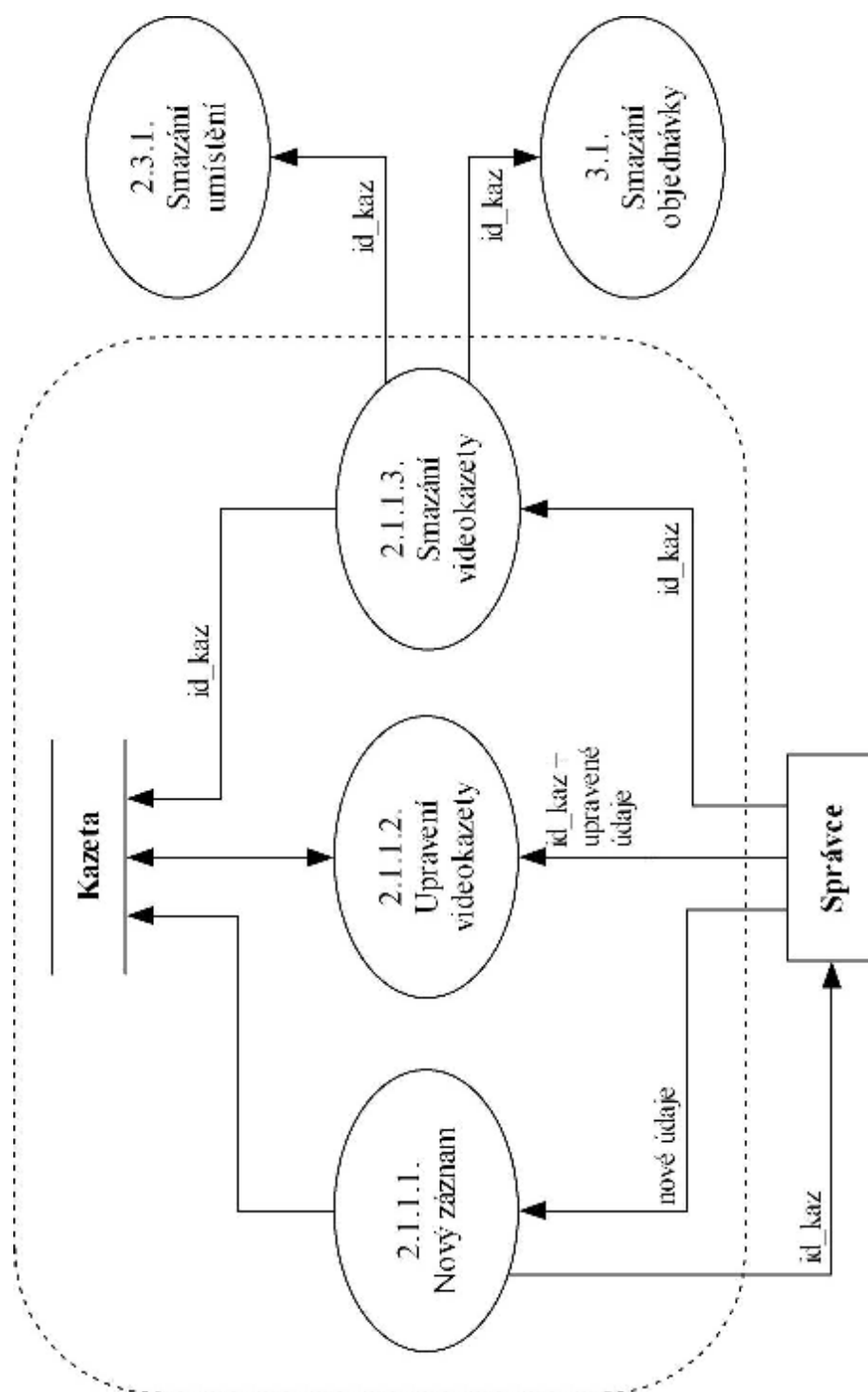
Obr. A-4 DFD 1.3. Osobní údaje uživatele



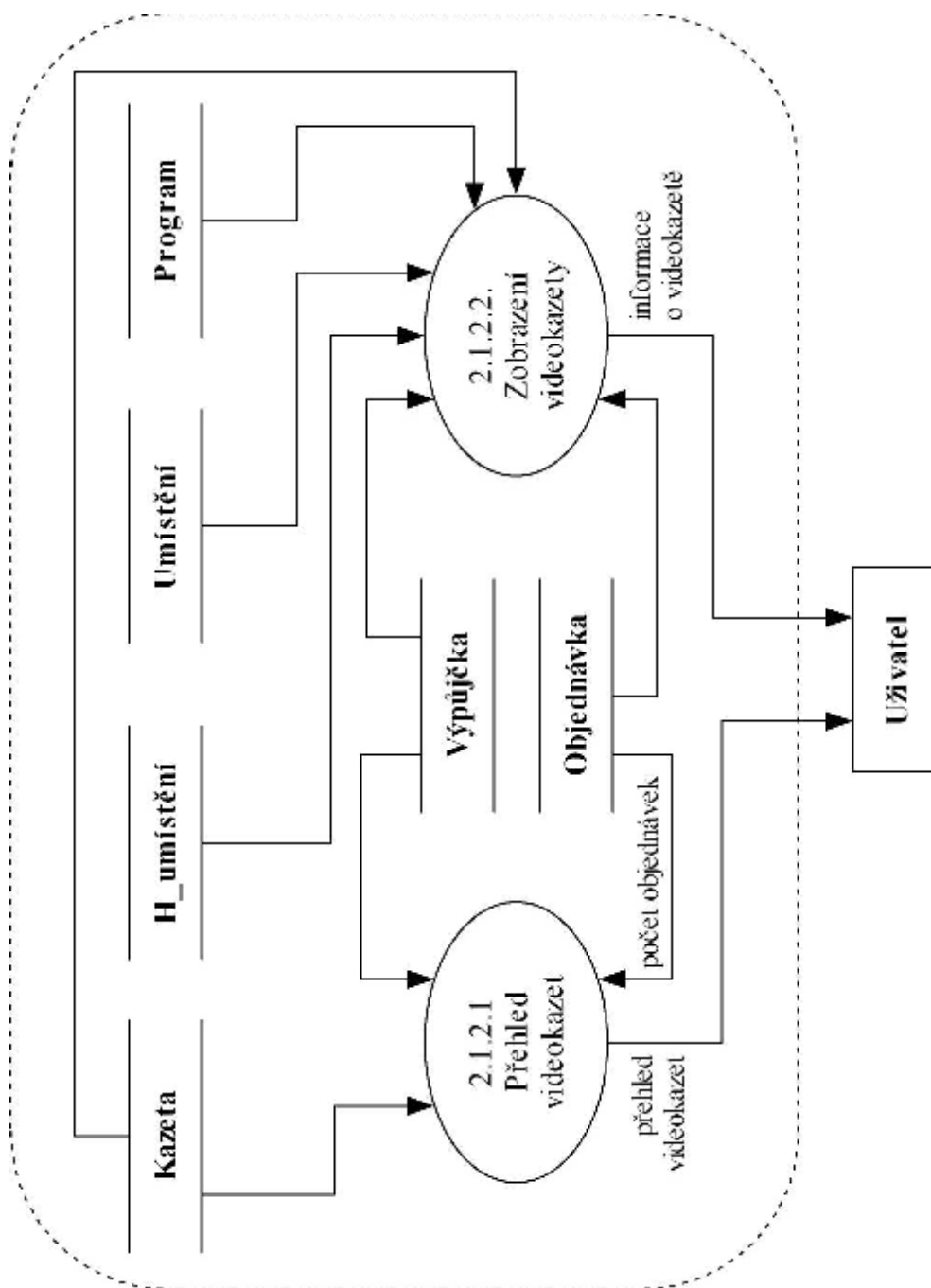
Obr. A-5 DFD 2. Evidence videokazet a pořadů



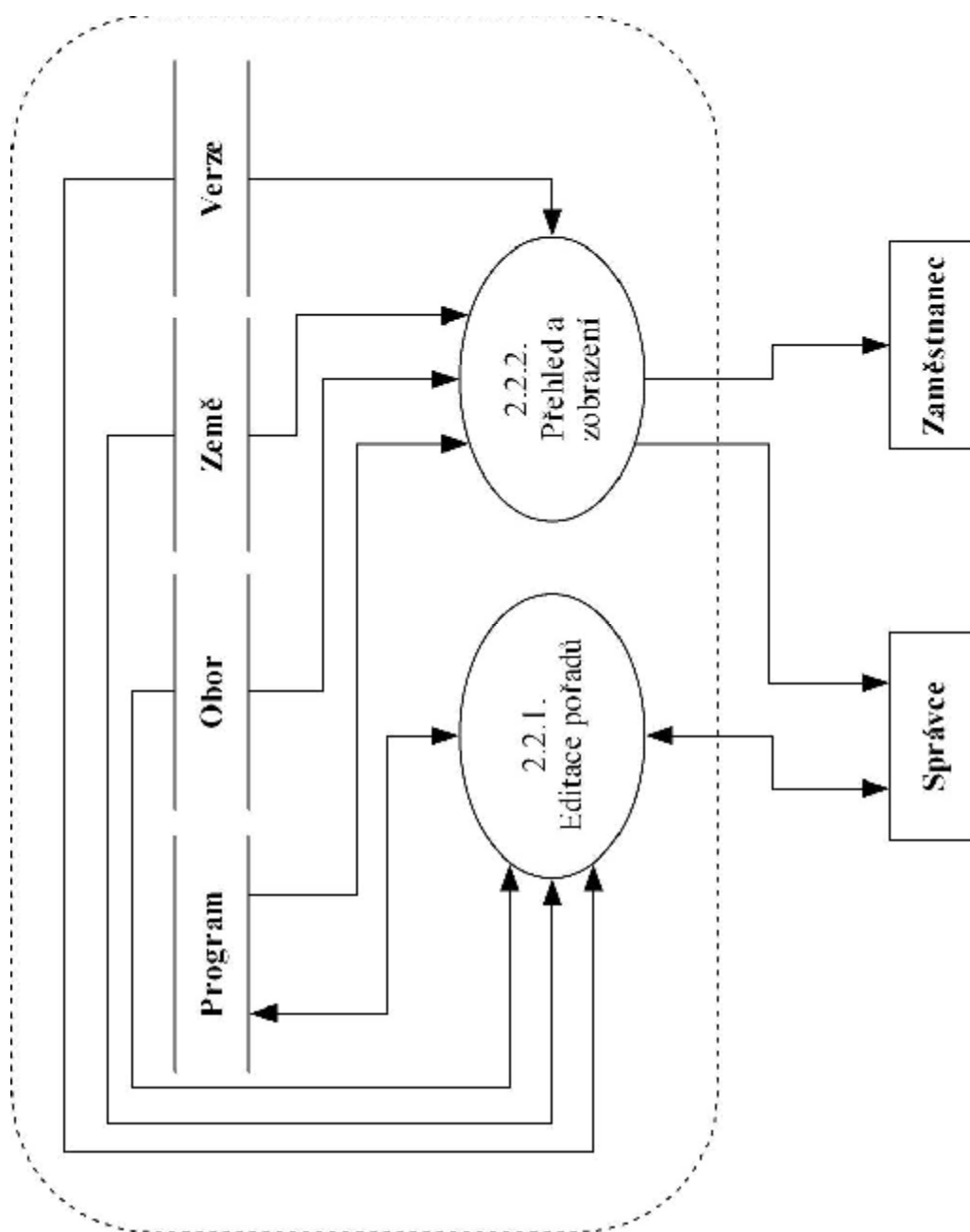
**Obr. A-6** DFD 2.1. Evidence videokazet



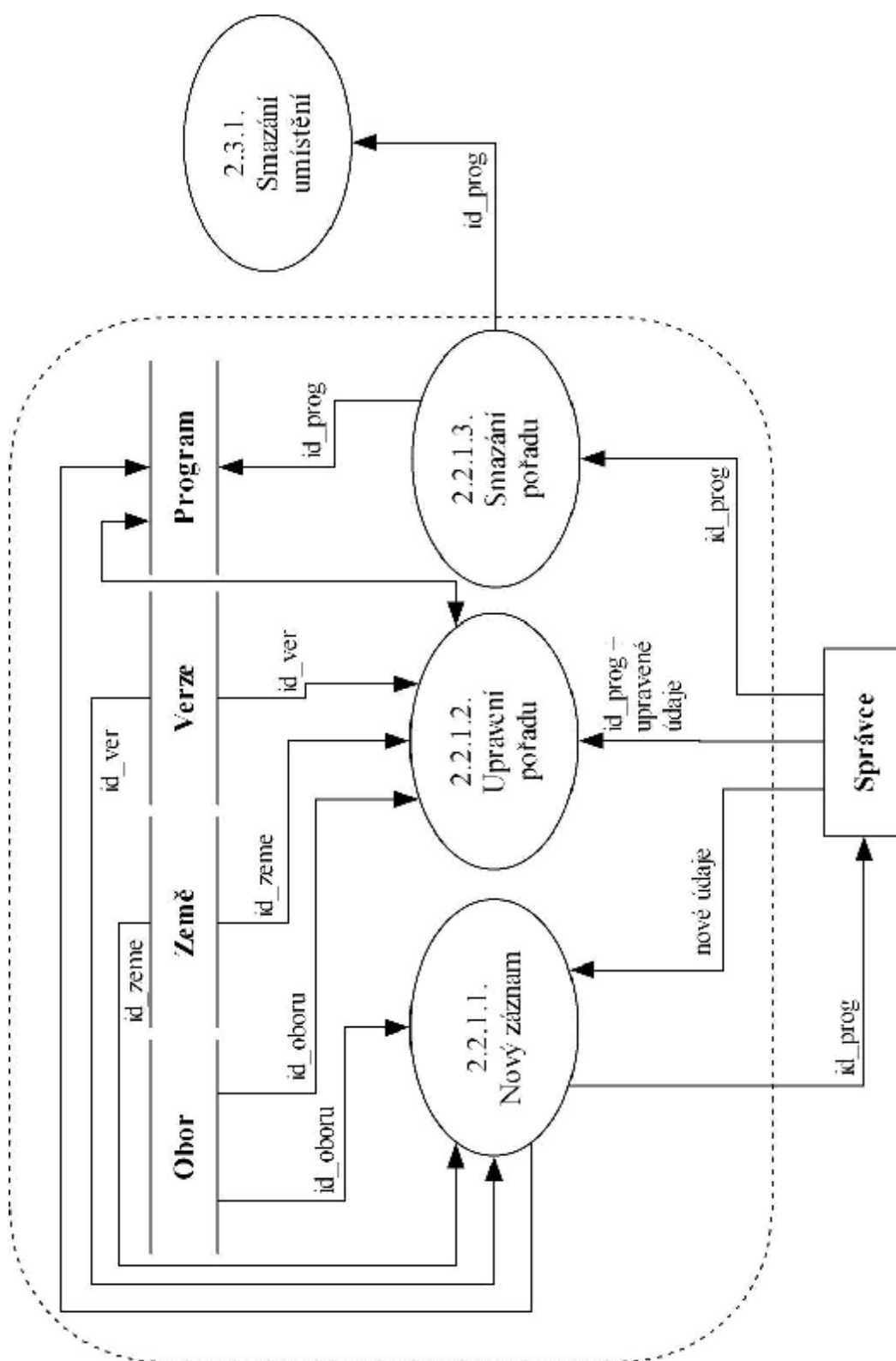
**Obr. A-7** DFD 2.1.1. Editace videokazet



Obr. A-8 DFD 2.1.2. Přehled a zobrazení videokazet

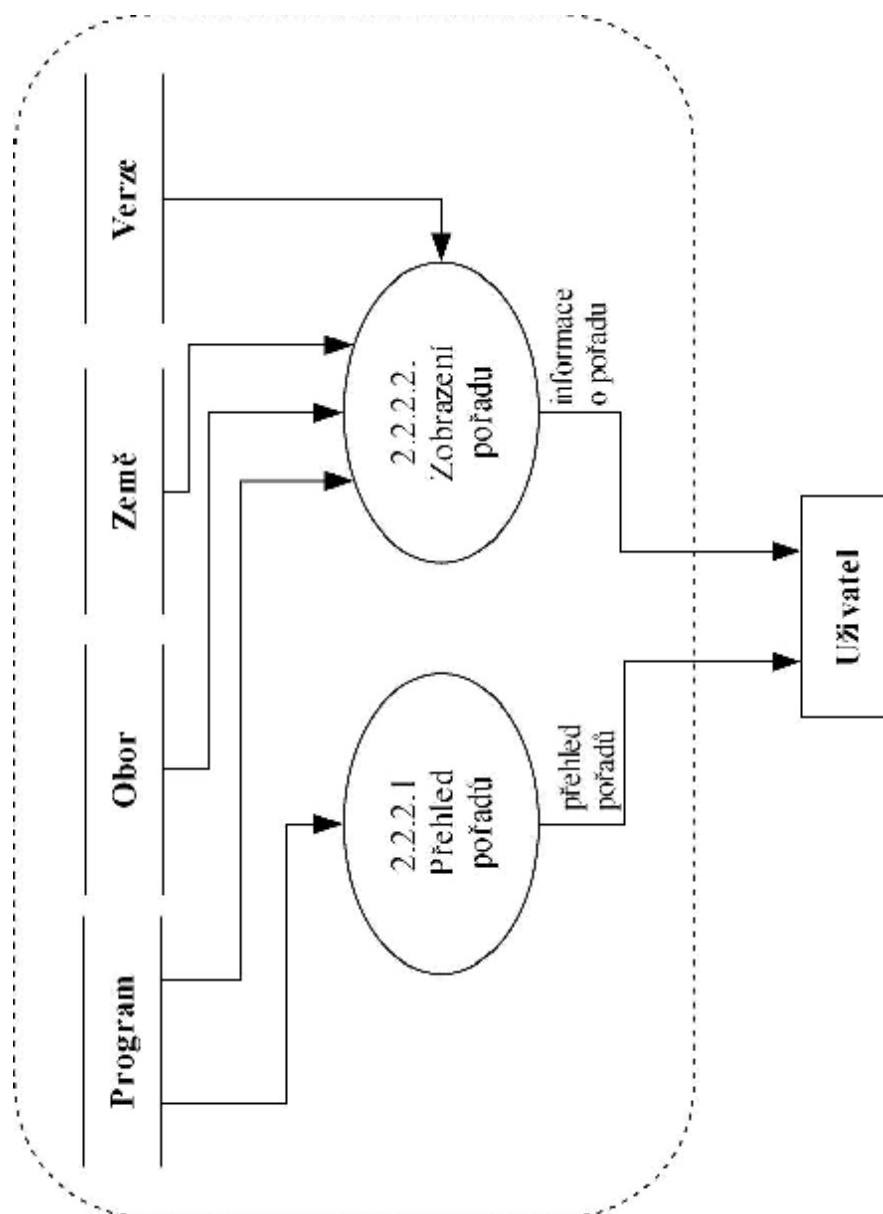


**Obr. A-9** DFD 2.2. Evidence pořadů

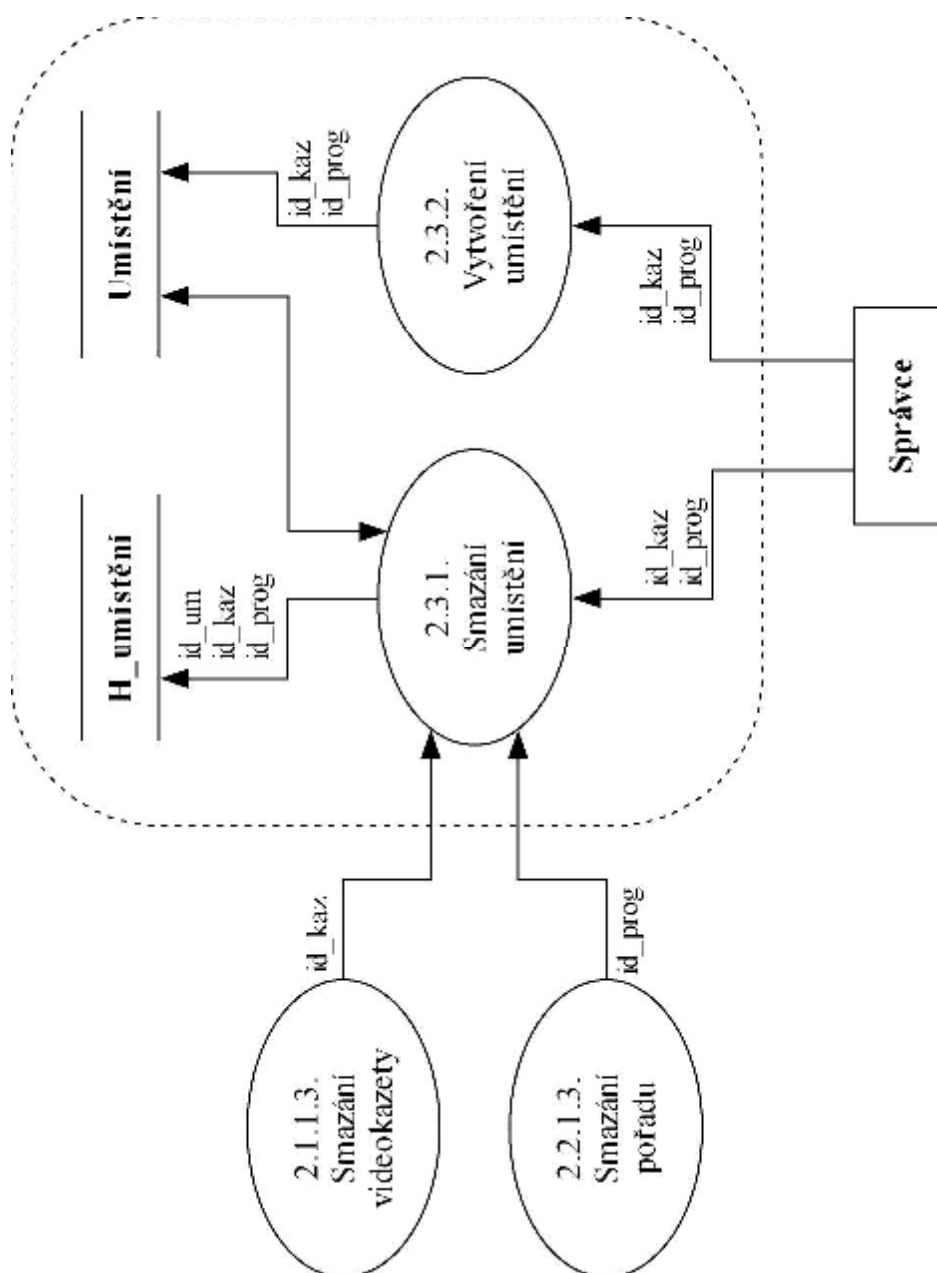


Obr. A-10 DFD 2.2.1. Editace pořadů

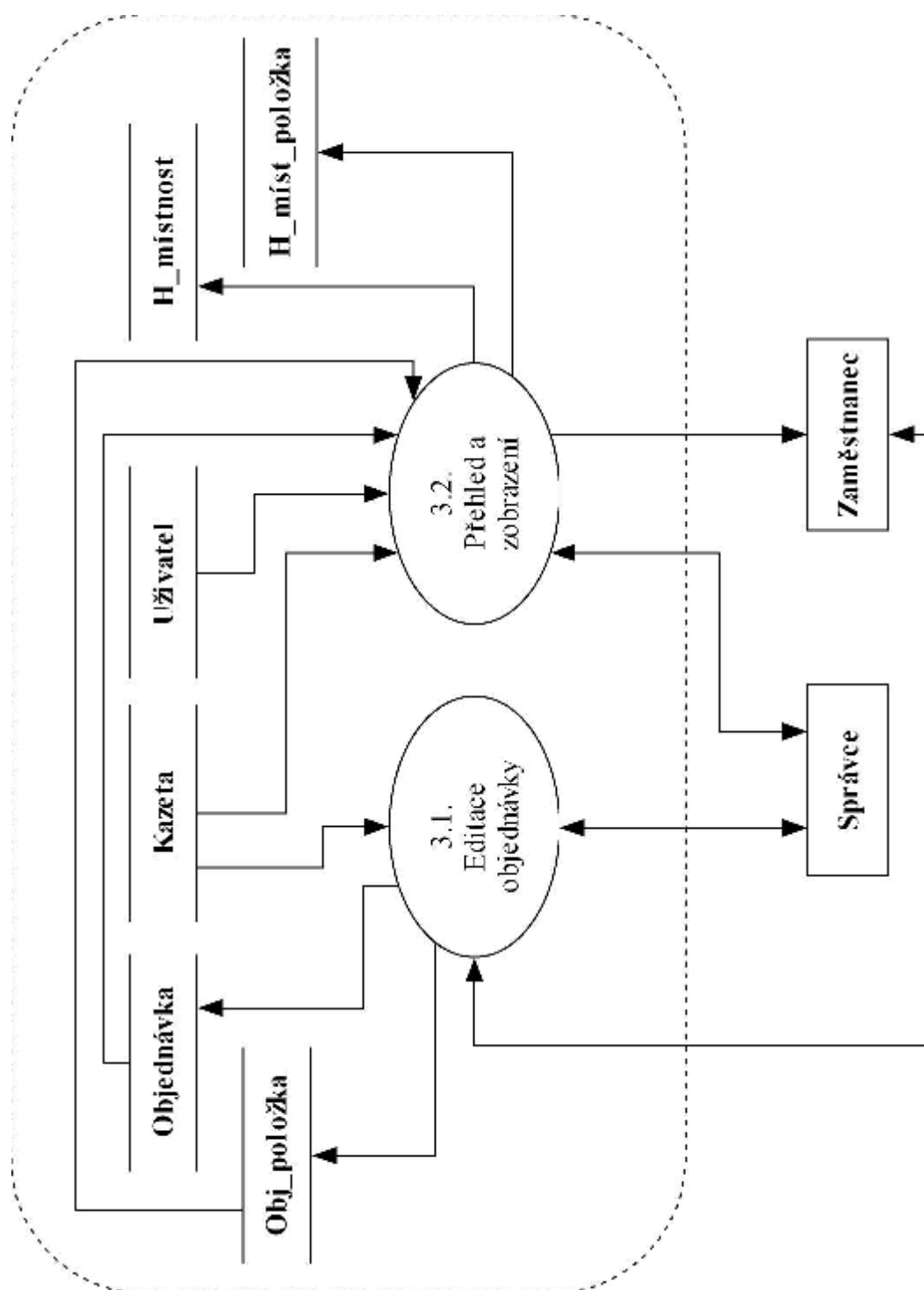




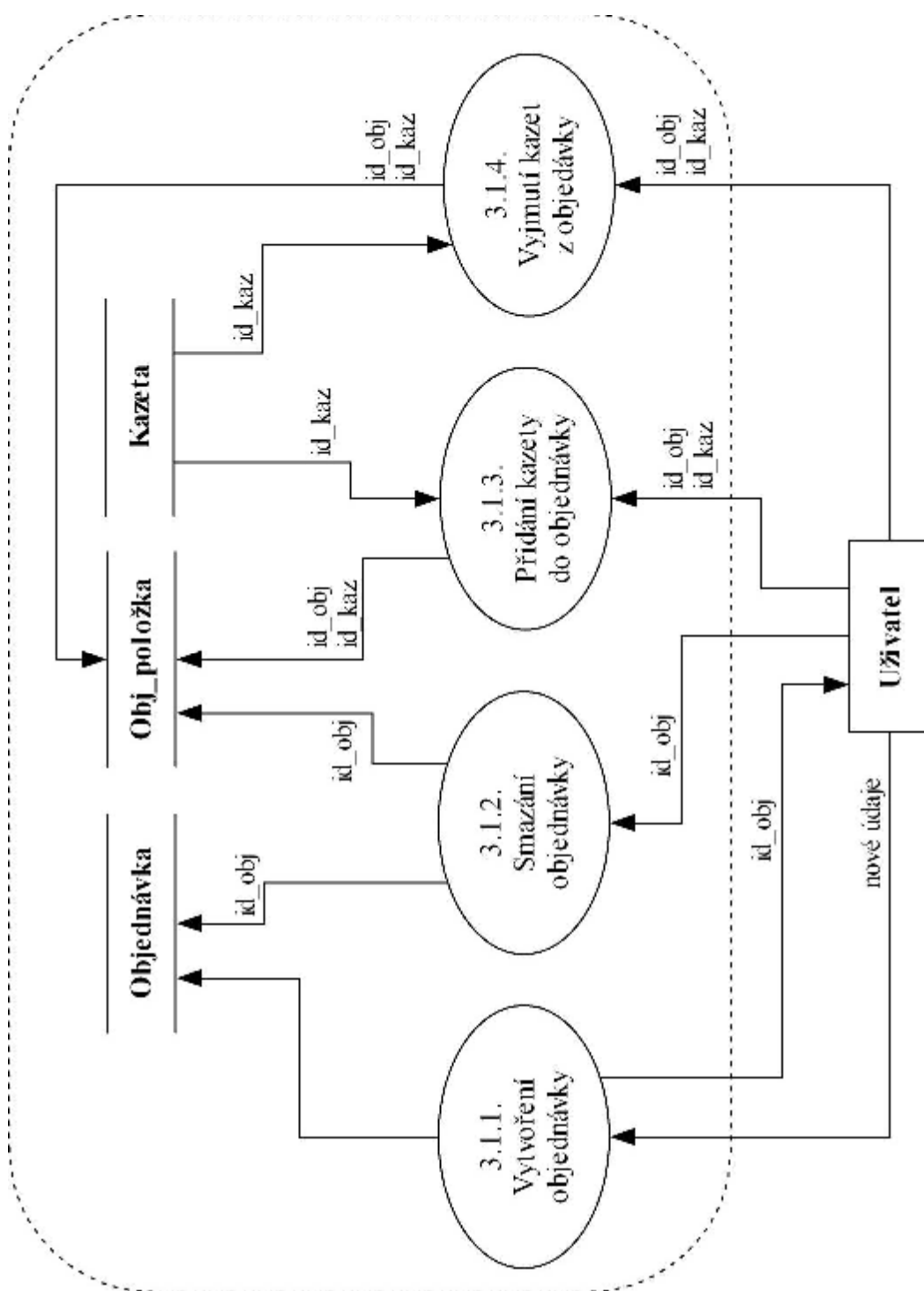
**Obr. A-11** DFD 2.2.2. Přehled a zobrazení pořadů



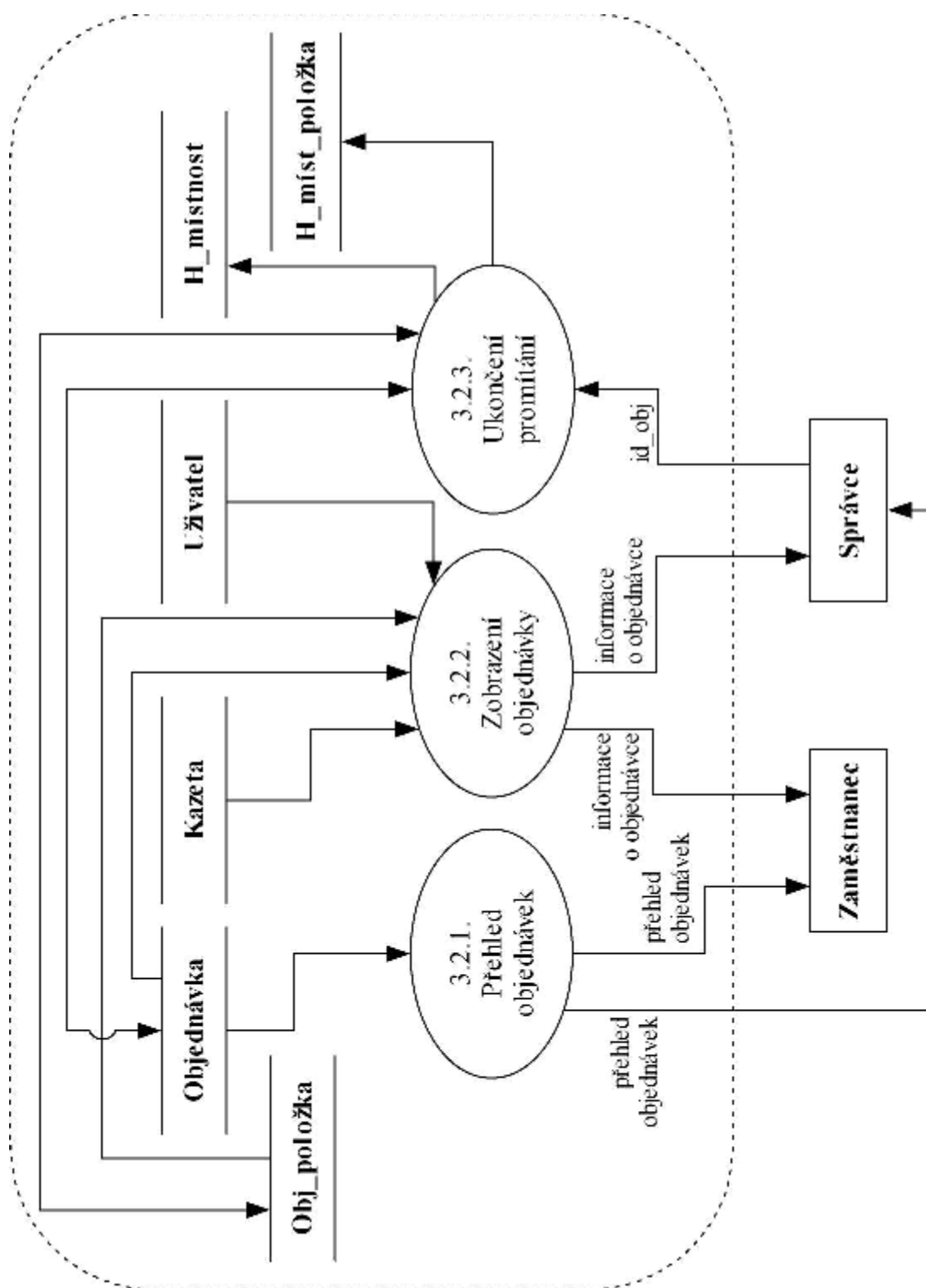
**Obr. A-12** DFD 2.3. Evidence umístění



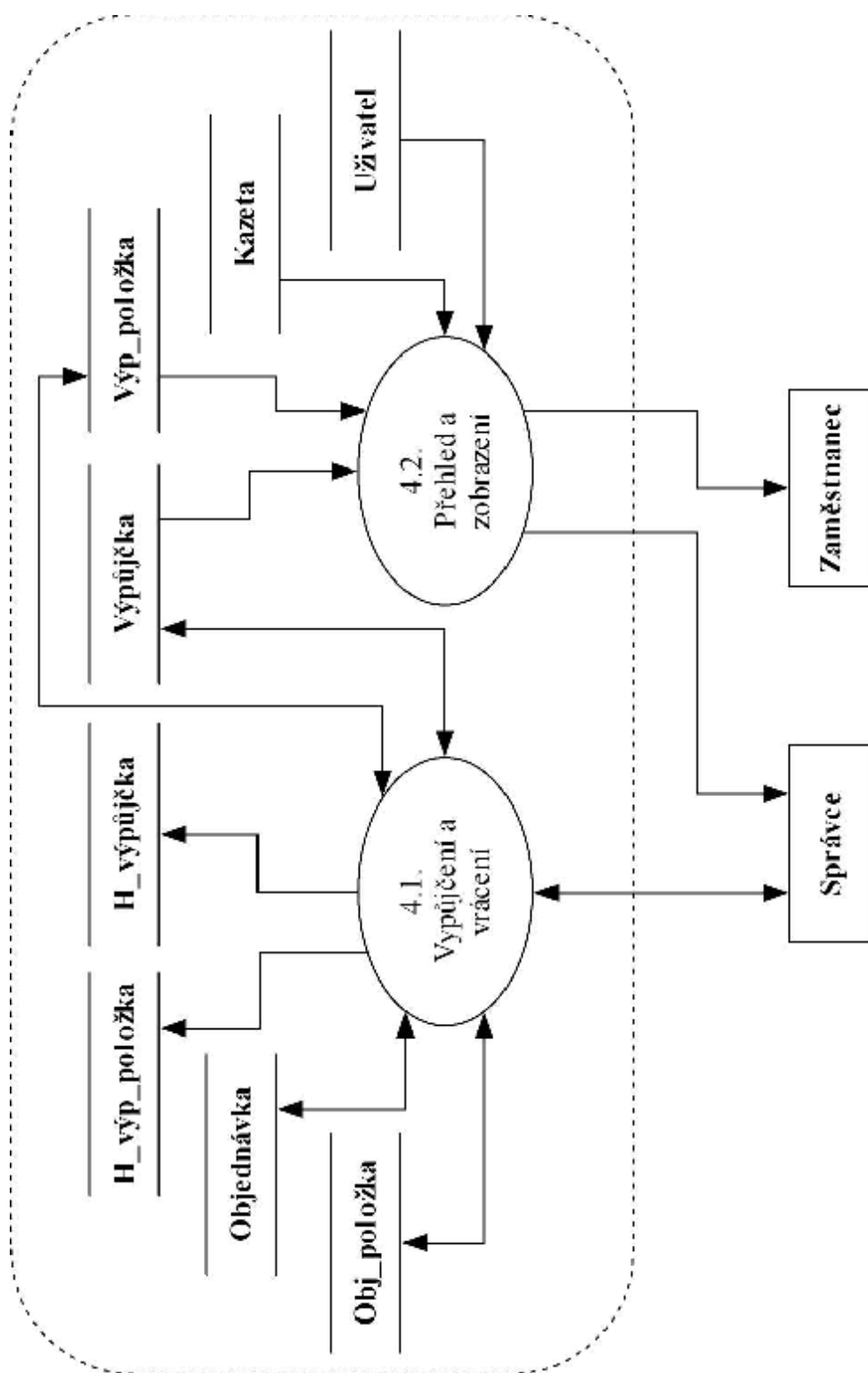
Obr. A-13 DFD 3. Objednávky



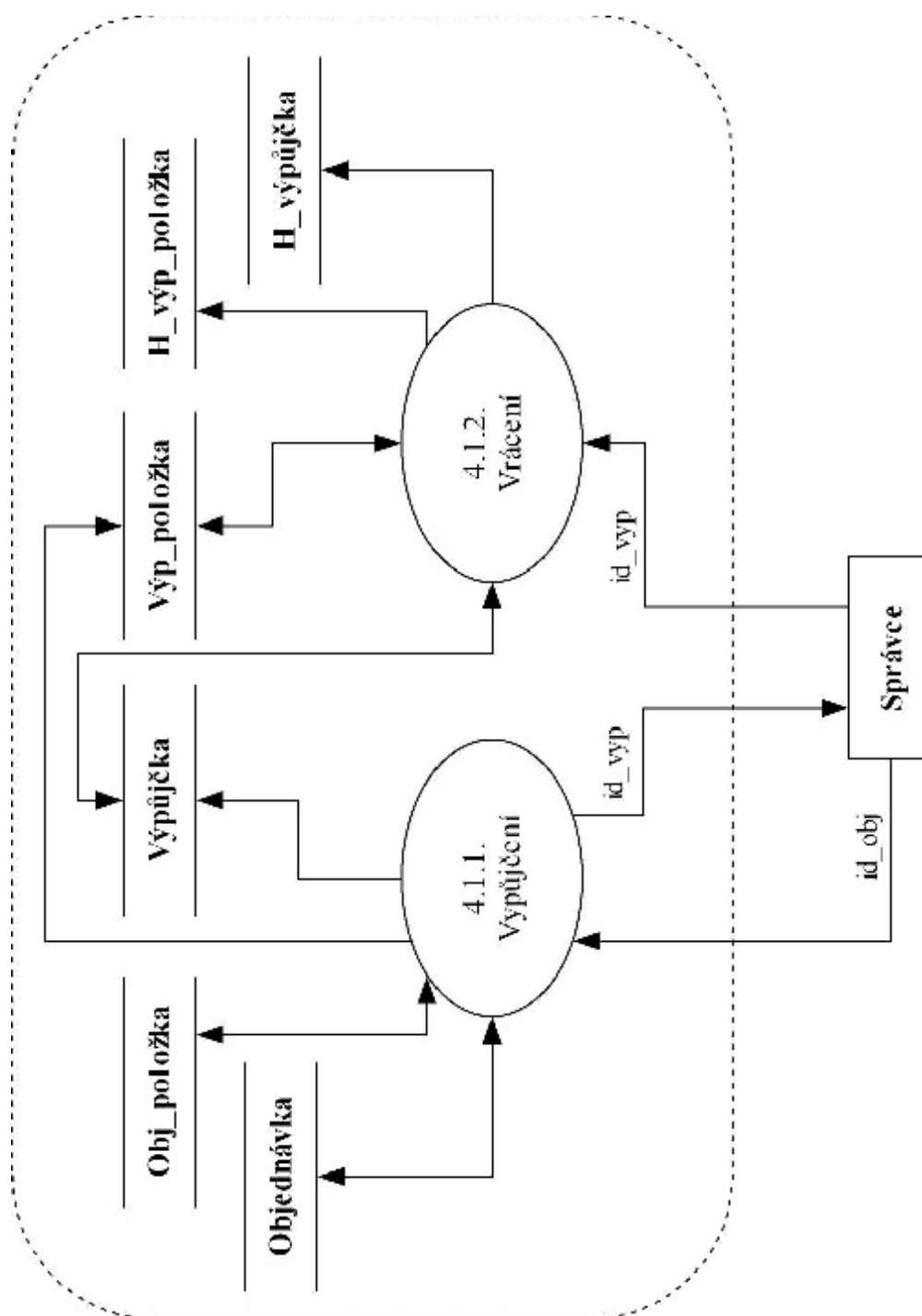
Obr. A-14 DFD 3.1. Editace objednávky



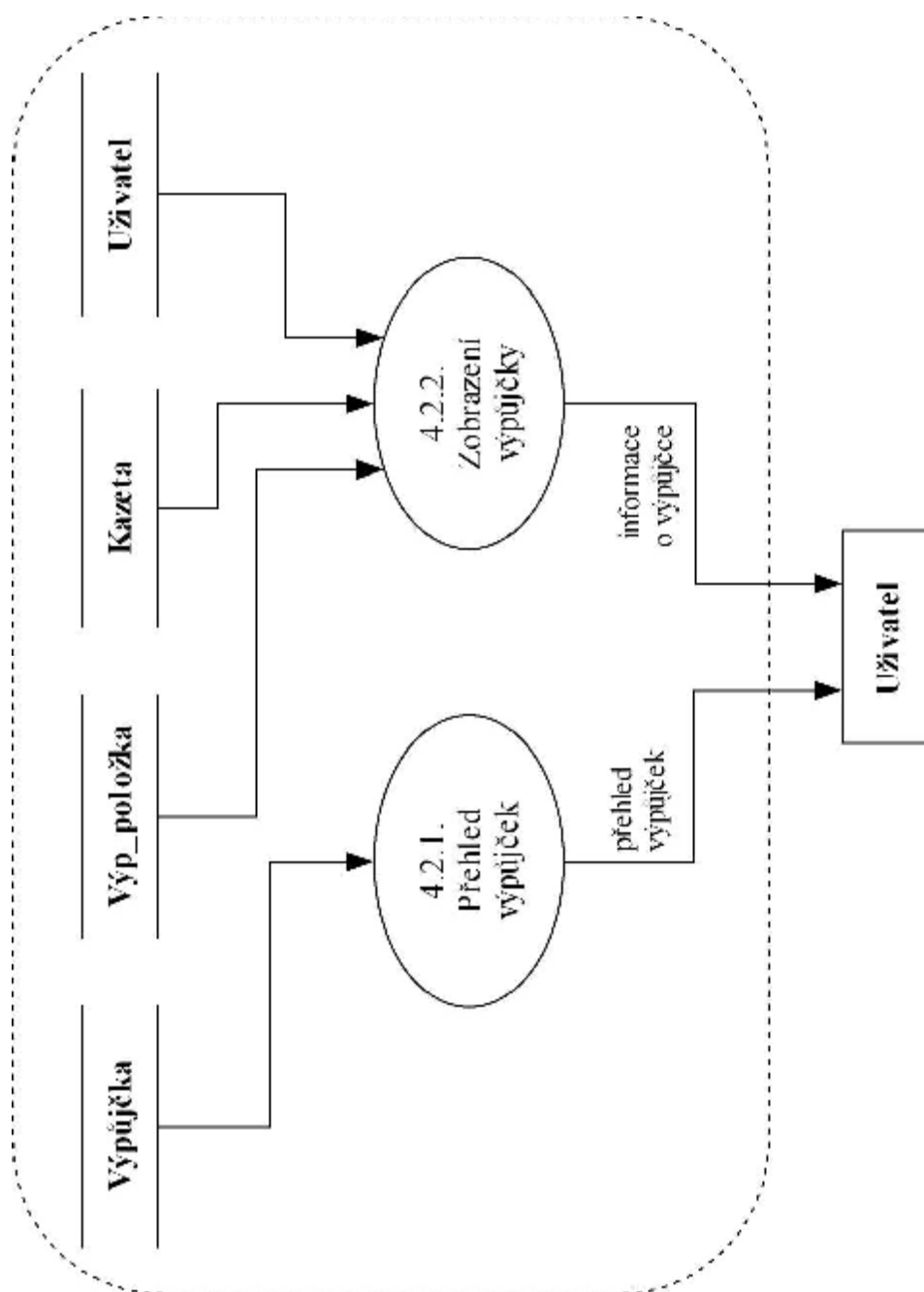
Obr. A-15 DFD 3.2. Přehled a zobrazení objednávek



Obr. A-16 DFD 4. Výpůjčky

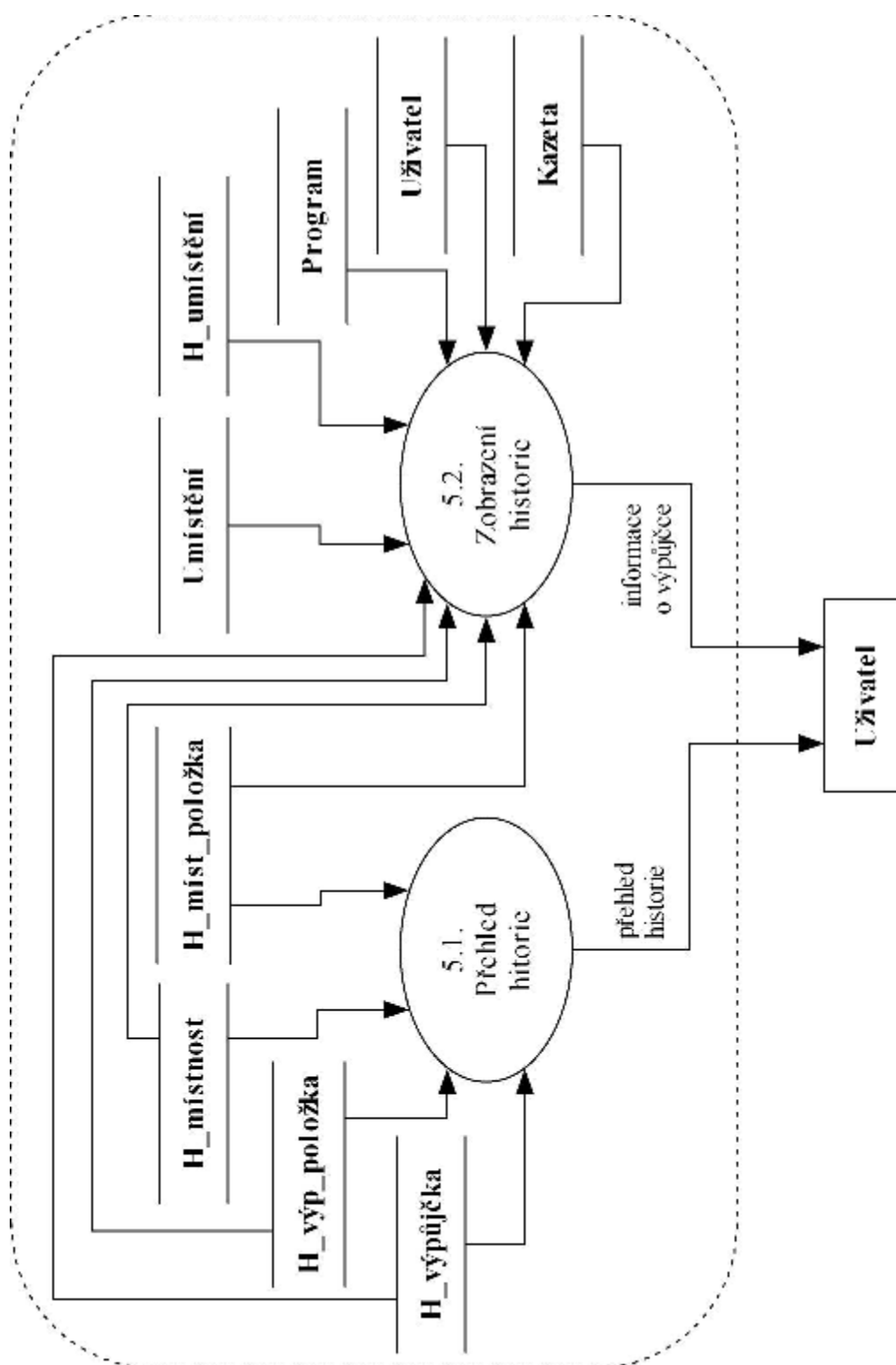


Obr. A-17 DFD 4.1. Vypůjčení a vrácení

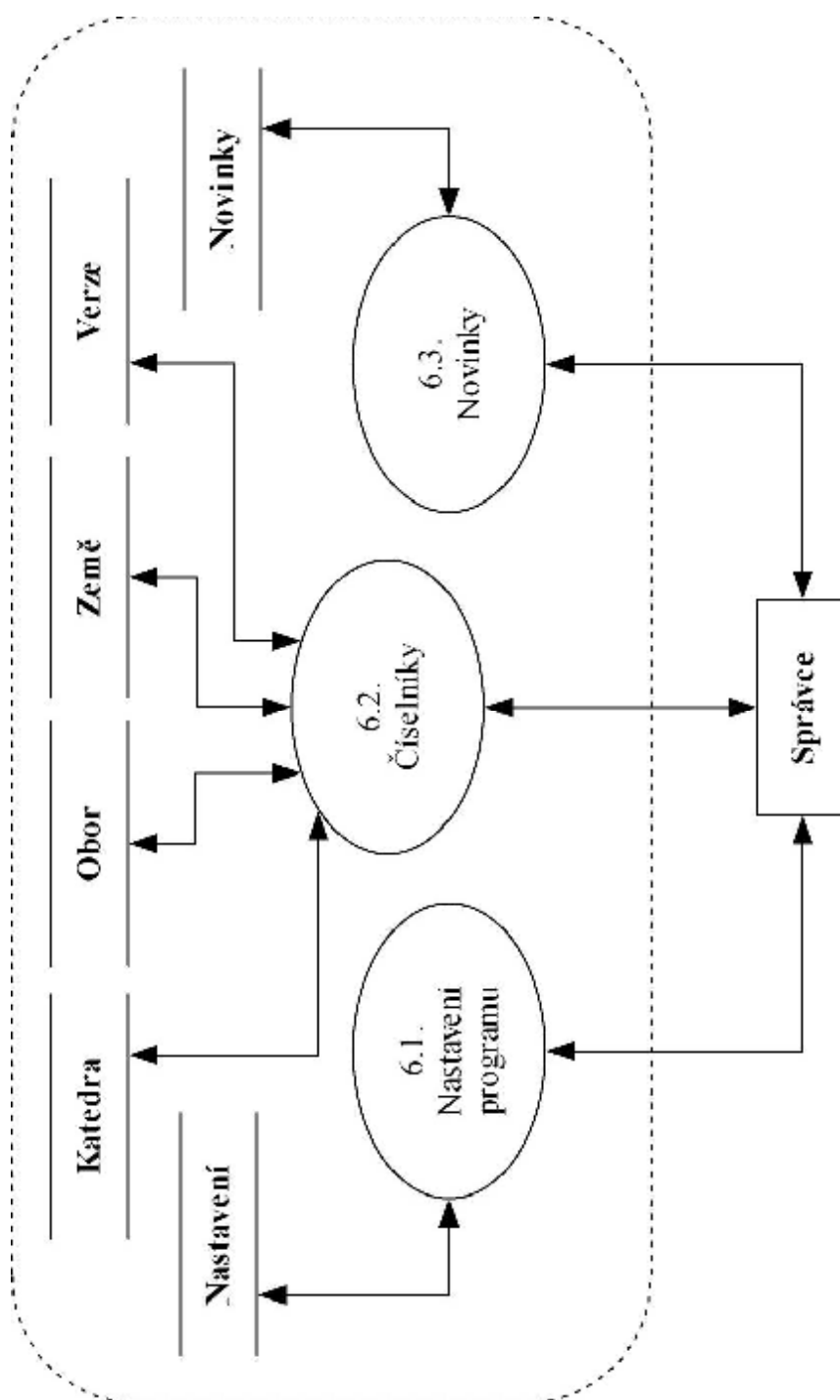


**Obr. A-18** DFD 4.2. Přehled a zobrazení výpůjček

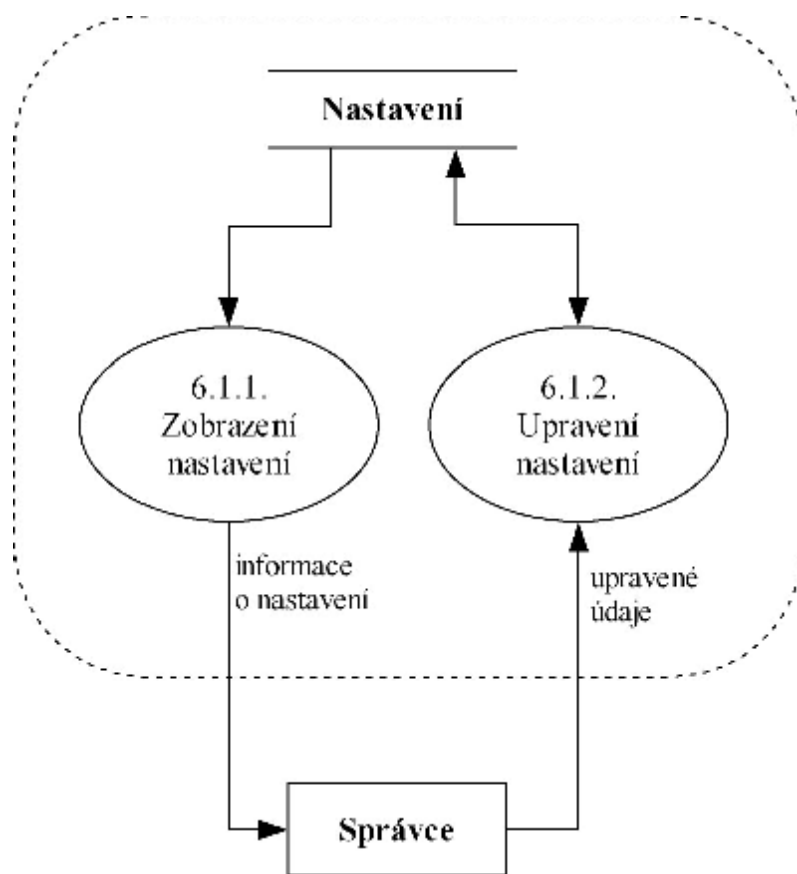




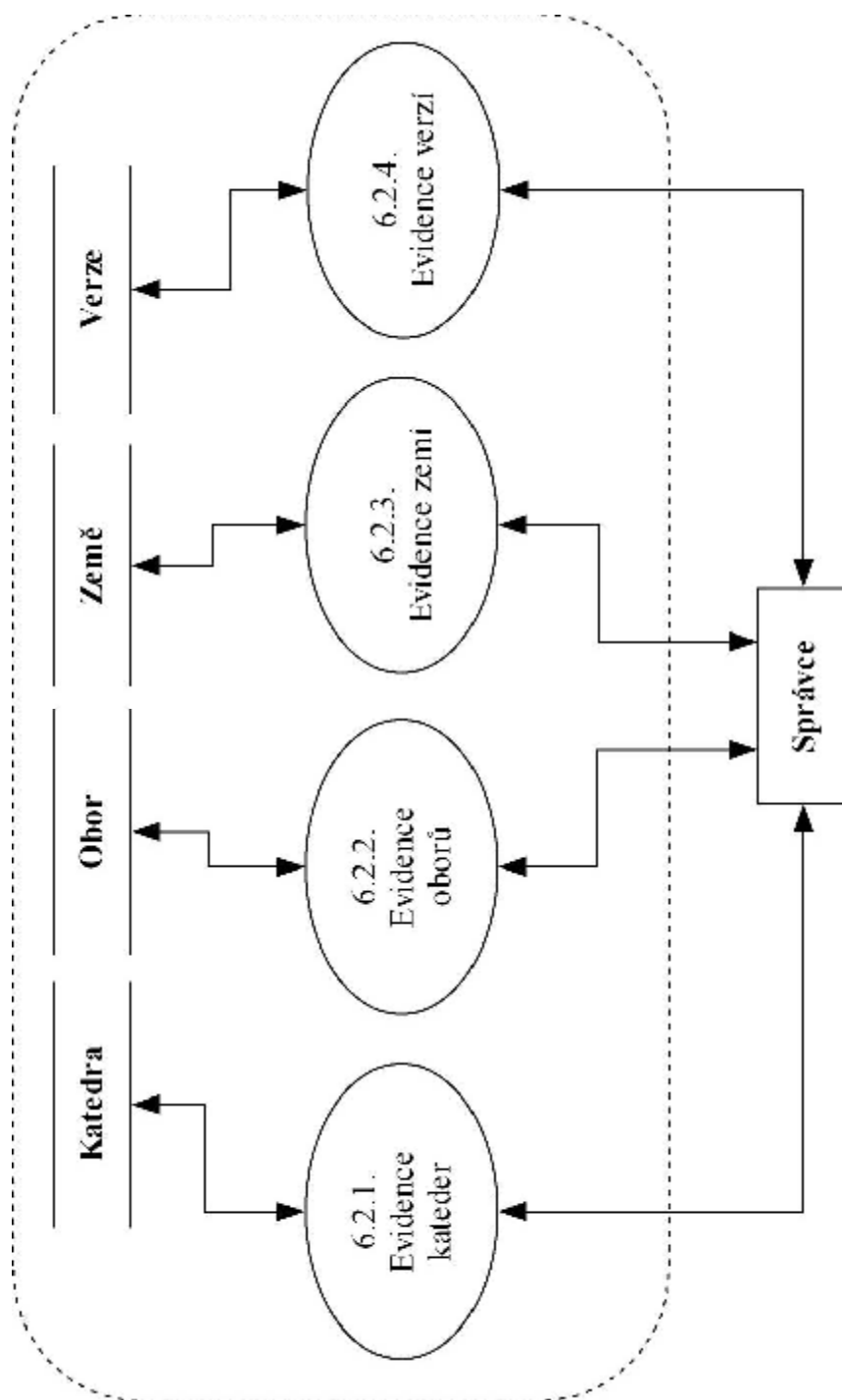
Obr. A-19 DFD 5. Historie



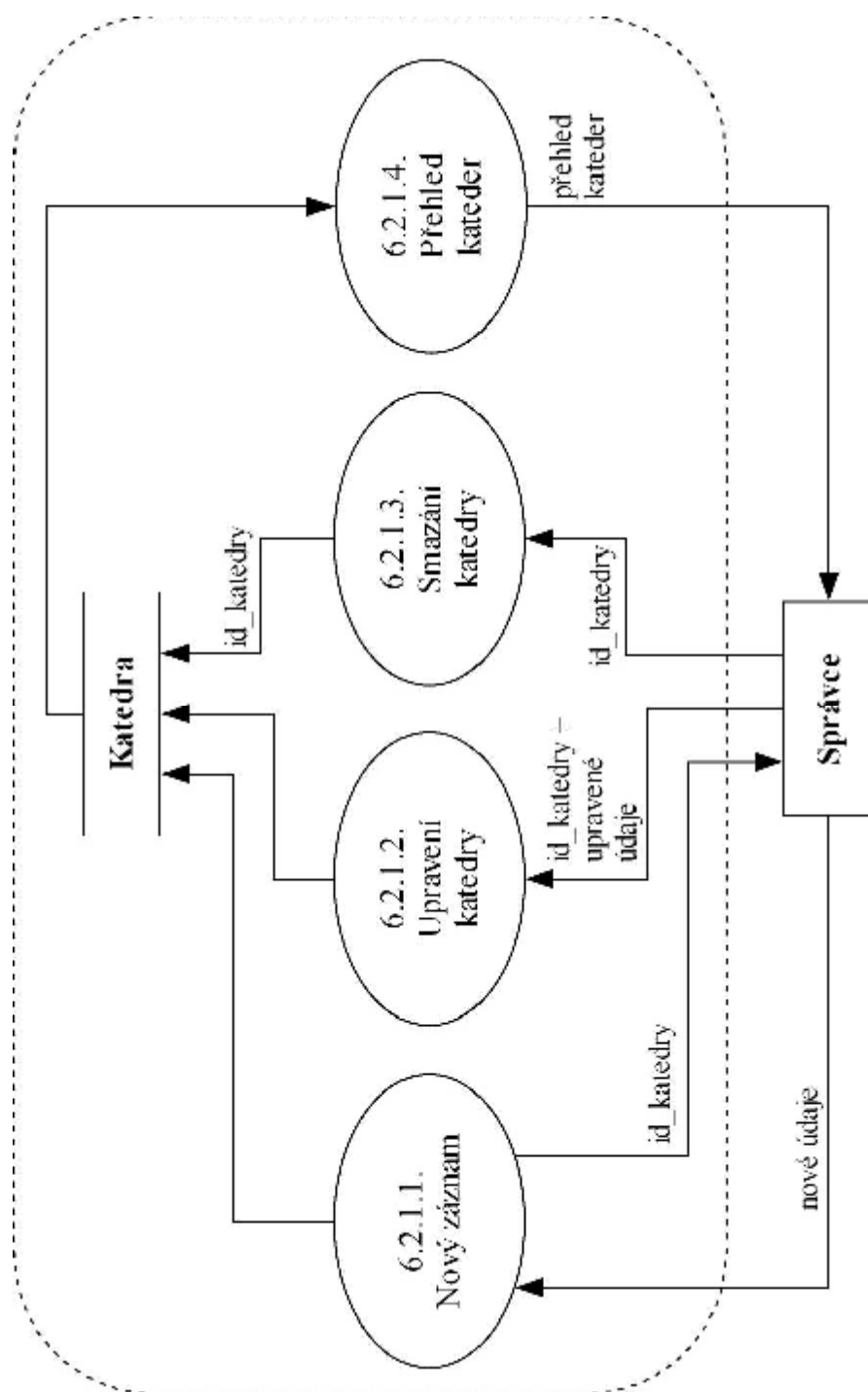
Obr. A-20 DFD 6. Nastavení a číselníky



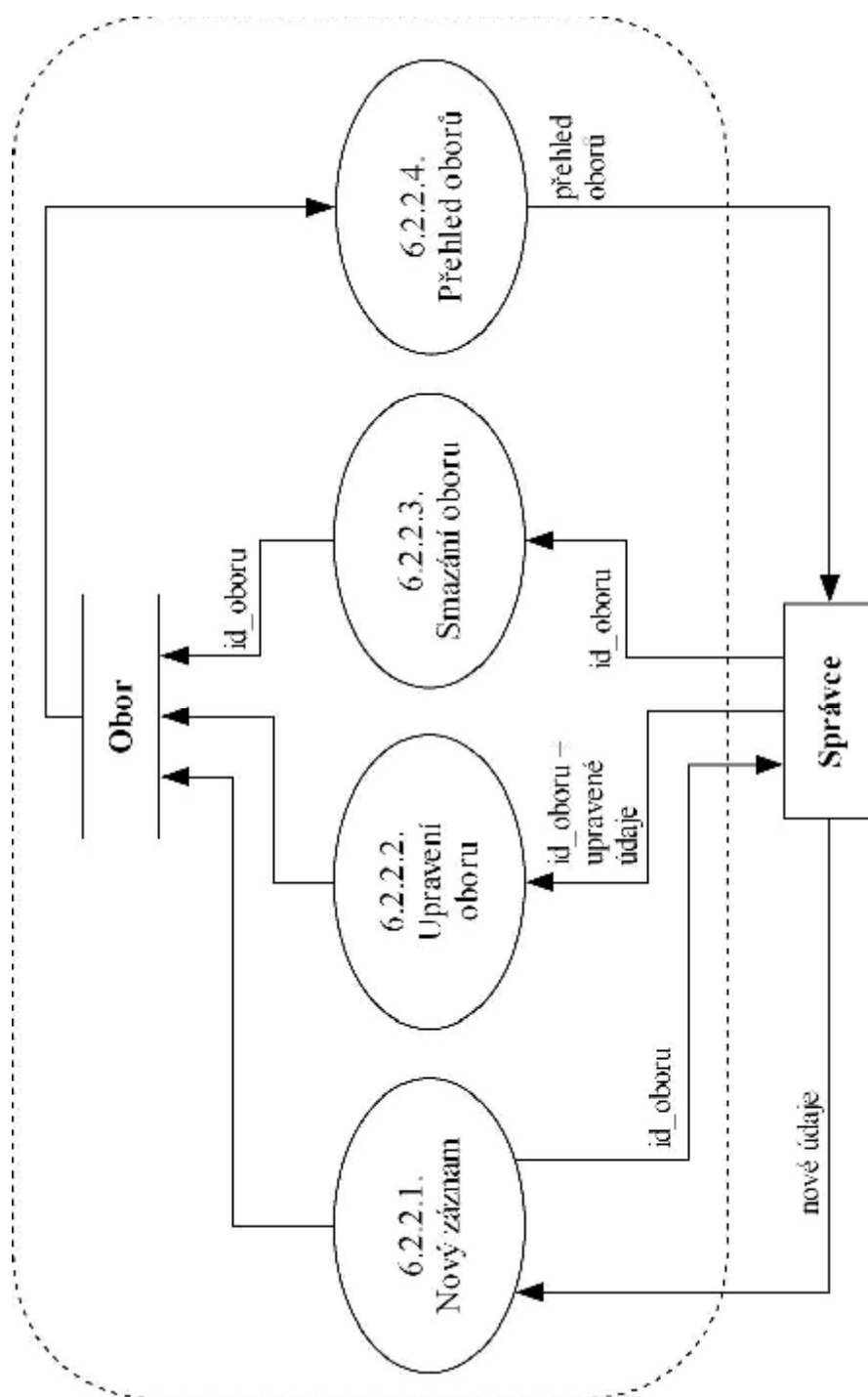
**Obr. A-21** DFD 6.1. Nastavení programu



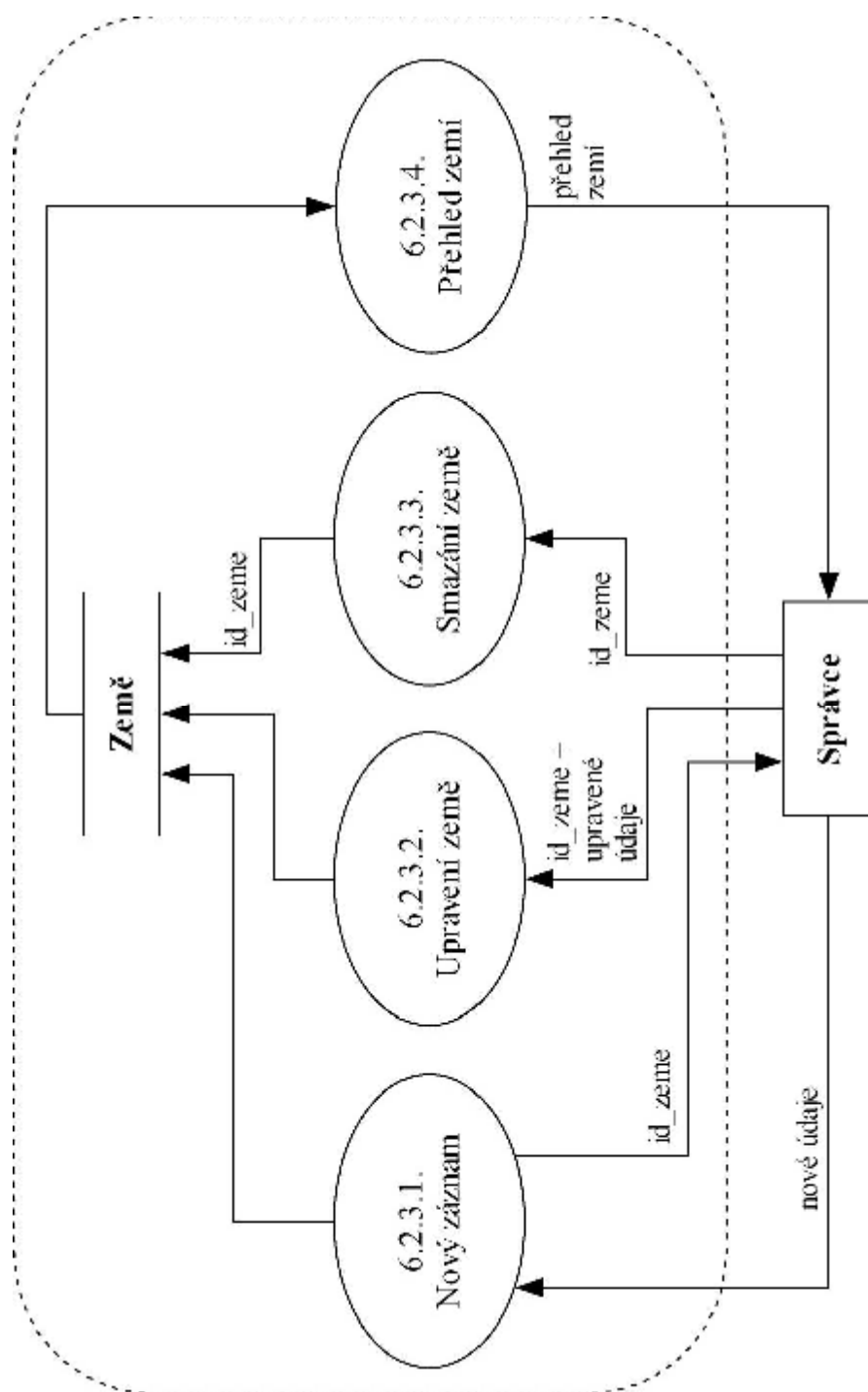
Obr. A-22 DFD 6.2. Číselníky



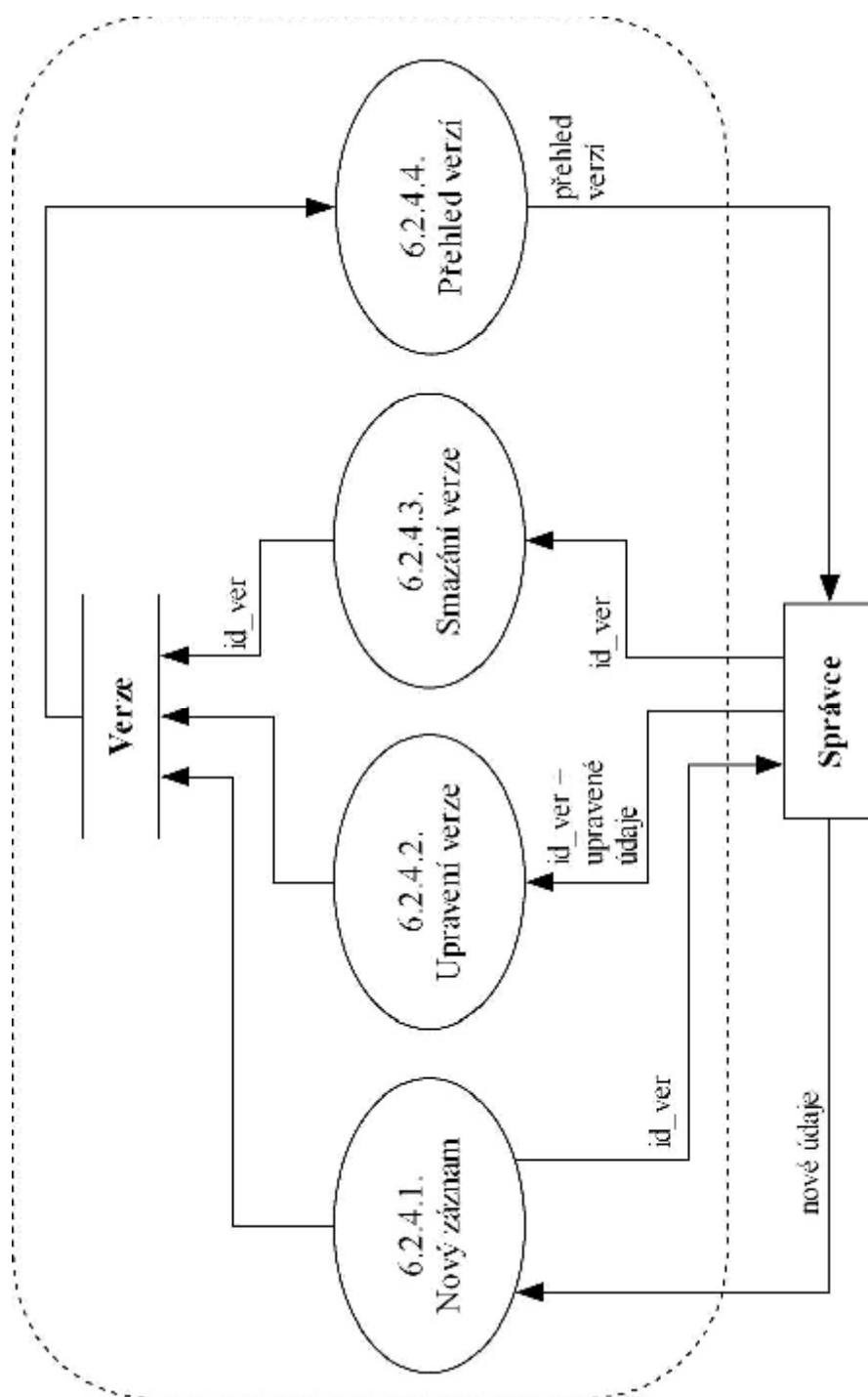
**Obr. A-23** DFD 6.2.1. Evidence kateder



**Obr. A-24** DFD 6.2.2. Evidence oborů

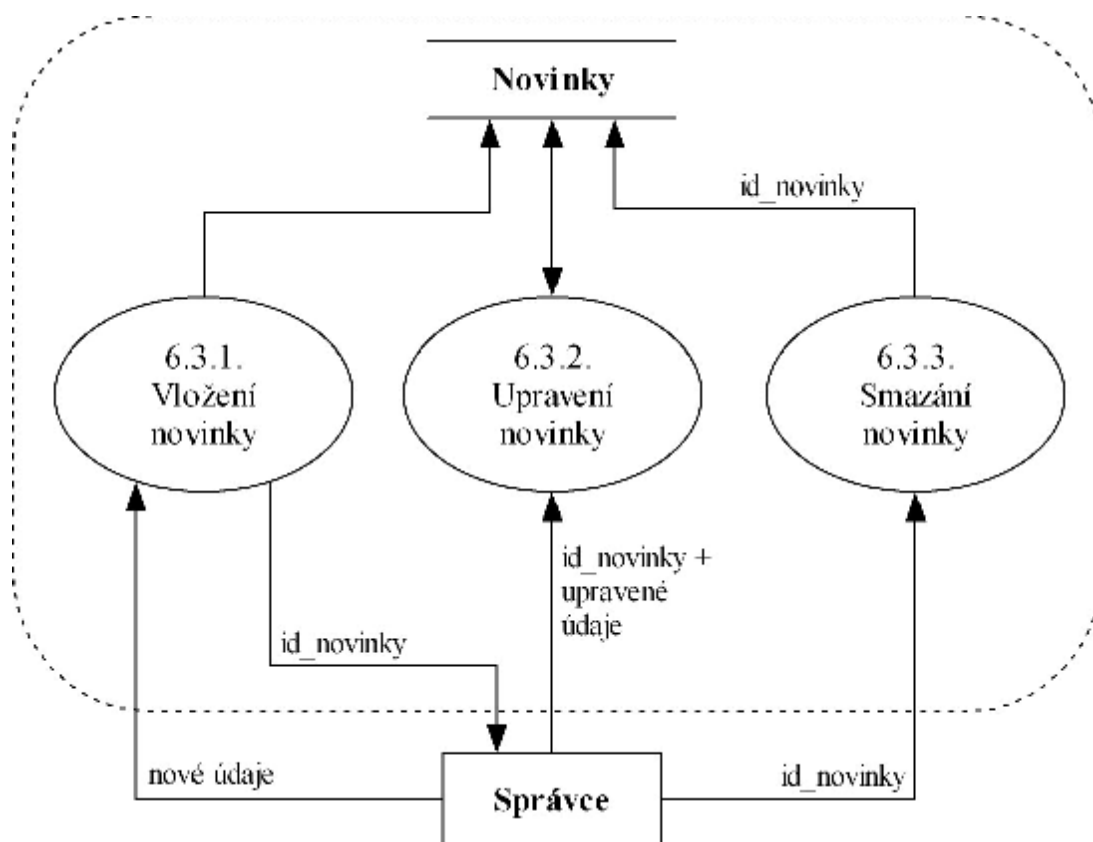


**Obr. A-25** DFD 6.2.3. Evidence zemí původu

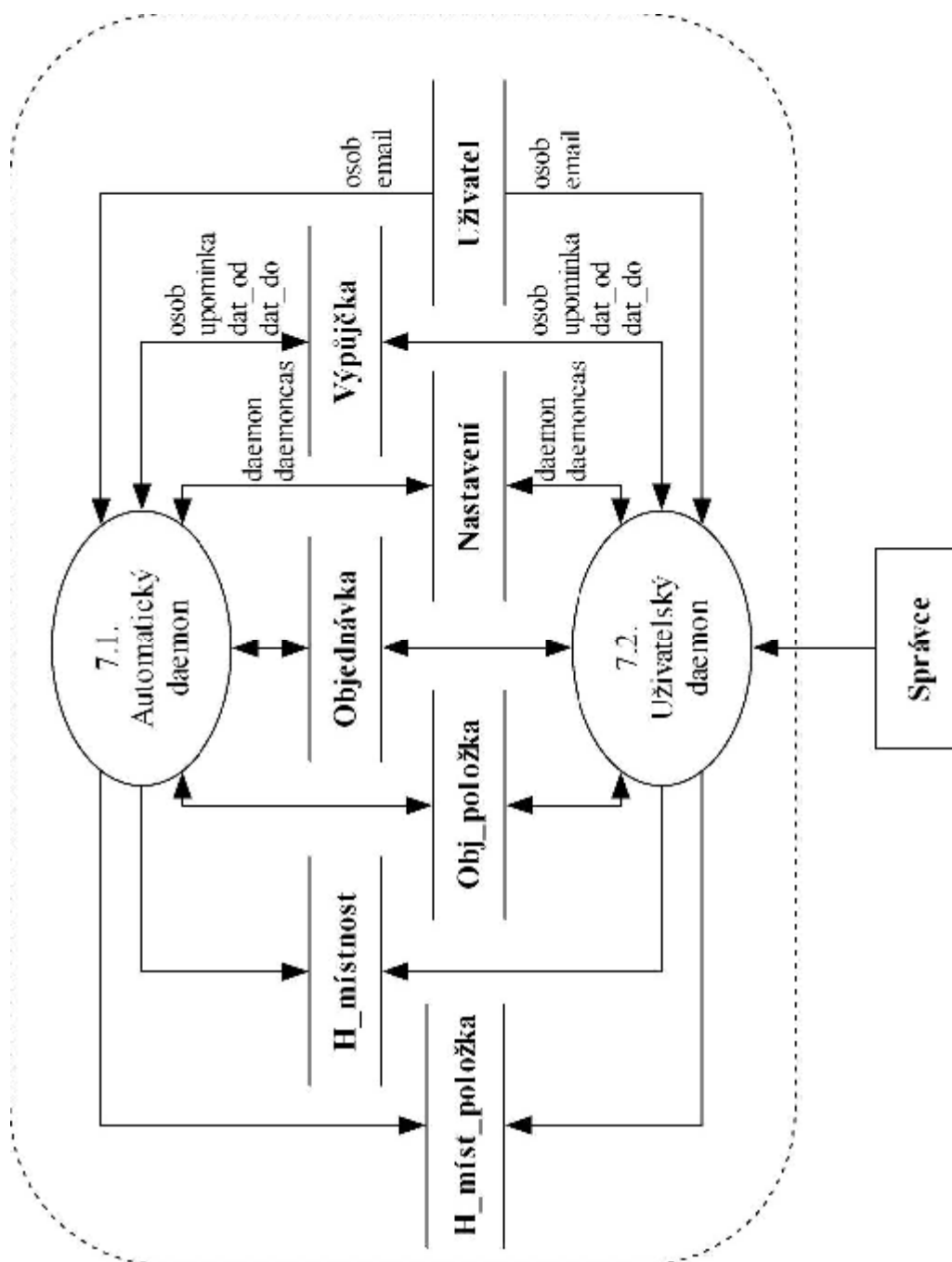


**Obr. A-26** DFD 6.2.4. Evidence jazykových verzí





Obr. A-27 DFD 6.3. Novinky



Obr. A-28 DFD 7. Daemon

## Příloha B – obecný popis souborů

Tato příloha obsahuje obecný popis souborů aplikace, neslouží k jejich detailnímu popisu. Pro získání větších znalostí o souborech je vhodné nahlédnout do programátorské dokumentace.

### Hlavní adresář (tzv. „root“):

- *daemon.php* – tento soubor slouží ke spuštění daemona. Daemon je vyvoláván v pravidelných intervalech pro provedení nutných akcí nad databází.
- *changelog.txt* – soubor obsahující přehled změn ve verzích Informačního systému Audiovizuálního centra VŠB-TUO.
- *index.php* – inicializační skript aplikace, přesměrovává pouze na soubor *portal.php*.
- *install.php* – instalační soubor, který kontroluje existenci souboru s údaji pro vstup do databáze. Neexistuje-li tento soubor, je vyvolán instalační proces. V opačném případě je oznámeno, že aplikace je již nainstalována, a nabídnuta možnost dalšího postupu.
- *portal.php* – soubor, který vytvoří session na serveru, provede kontrolu na existenci souboru s údaji pro vstup do databáze. Pak buď vyvolá instalační proces, nebo vytvoří instanci třídy s rozhraním a vyvolá její spuštění.

### Adresář „include“:

- *appversion.inc* – soubor, který obsahuje definici s číslem verze aplikace.
- *Branch.inc* – soubor, který implementuje třídu pro obsluhu oborů; výpis, vkládání, aktualizaci, mazání dat o oboru v databázi.
- *Cassette.inc* – soubor, který implementuje třídu pro obsluhu kazet; výpis, vkládání, aktualizaci, mazání dat o kazetách v databázi.
- *CommonThings.inc* – soubor, který implementuje třídu, kde se nachází často používané metody, které se v jednotlivých implementacích tříd nemění. Tato třída bude rozšiřována ostatními třídami.
- *Country.inc* – soubor, který implementuje třídu pro obsluhu země původu, výpis, vkládání, aktualizaci, mazání dat o zemi původu v databázi.
- *Daemon.inc* – tento soubor implementuje třídu, která se stará o tyto automatické akce: smazání nepotvrzených objednávek; upozornění na vyzvednutí objednávky; vymazání nevyzvednutých objednávek; přesunutí objednávky místnosti, která je už prošlá, do historie; upozornění na vrácení výpůjčky; zaslání upomínky. Akce jsou prováděny ve

dvou režimech: uživatelský (vyvoláno uživatelem); automatický (vyvoláno systémem přes soubor *daemon.php*).

- *Database.inc* – soubor, který implementuje třídu starající se o komunikaci s MySQL databází.
- *dbinclude.inc* – soubor, který obsahuje informace pro přístup do MySQL databáze. Vytváří se až při instalaci aplikace.
- *Department.inc* – soubor, který implementuje třídu pro obsluhu katedry; výpis, vkládání, aktualizaci, mazání dat o katedře.
- *Gui.inc* – soubor, který obsahuje implementaci třídy s rozhraním mezi uživatelem a systémem.
- *History.inc* – soubor, který implementuje třídu pro obsluhu kompletní historie; výpisy, vkládání apod.
- *Lend.inc* – soubor, který implementuje třídu pro obsluhu výpůjčky; výpis, vkládání, aktualizaci, mazání dat o výpůjčce v databázi.
- *LendItem.inc* – soubor, který obsahuje definici třídy položky výpůjčky a práci s touto položkou; přidávání a odebírání z databáze.
- *Messages.inc* – soubor, který obstarává ukládání chyb a zpráv do session a práci s nimi.
- *News.inc* – soubor, který implementuje třídu pro obsluhu novinek; výpis, vkládání, aktualizaci, mazání dat o novince v databázi.
- *Option.inc* – soubor, který implementuje třídu pro obsluhu nastavení aplikace; výpis hodnot nastavení z databáze a její aktualizace.
- *Order.inc* – soubor, který implementuje třídu pro obsluhu objednávek; výpis, vkládání, aktualizaci, mazání dat o objednávkách v databázi.
- *OrderItem.inc* – soubor, který obsahuje definici třídy položky objednávky a práci s touto položkou; přidávání a odebírání z databáze.
- *Page.inc* – soubor, který obsahuje implementaci třídy s následujícími možnostmi: načítání šablony; vyhledávání, mazání a nahrazování bloků v šablonách; vkládání chybových hlášek a zpráv do šablon; ukládání aktuální URL do session, načítání poslední URL ze session; zobrazení stránky ve vybraném jazyce.
- *Place.inc* – soubor, který implementuje třídu pro obsluhu umístění pořadů na videokazetě; výpis, vkládání, aktualizaci, mazání dat o umístěných pořadech v databázi.
- *Program.inc* – soubor, který implementuje třídu pro obsluhu pořadů; výpis, vkládání, aktualizaci, mazání dat o pořadech v databázi.
- *SendMail.inc* – soubor, který obsahuje třídu pro posílání e-mailů systémem uživateli.
- *User.inc* – soubor, který implementuje třídu pro obsluhu uživatele; výpis, vkládání, aktualizaci, mazání dat o uživateli v databázi.
- *Version.inc* – soubor, který implementuje třídu pro obsluhu jazykové verze programu; výpis, vkládání, aktualizaci, mazání dat o jazykové verzi v databázi.

### Adresář „template“:

- *install.tpl* – šablona, která slouží k instalačnímu procesu aplikace.
- *mainpage.cassette.tpl* – šablona, která je potřebná k zobrazení informací pro procesy vkládání, editování, zobrazování a sestavování přehledů ohledně videokazet.
- *mainpage.history.tpl* – šablona slouží pro sestavování webových stránek týkajících se historie výpůjček a promítání v místnostech.
- *mainpage.lend.tpl* – tato šablona umožňuje sestavit dialogy týkající se informací ohledně výpůjček videokazet.
- *mainpage.login.tpl* – šablona slouží k zobrazení informací pro přihlášení uživatele do aplikace.
- *mainpage.messages.tpl* – šablona umožňuje zobrazení chybové hlášky nebo zprávy, která nastala v průběhu nějakého kroku aplikace.
- *mainpage.news.tpl* – šablona slouží k sestavení dialogů pro zobrazení, vkládání, editování a smazání novinky z aplikace.
- *mainpage.option.tpl* – šablona umožňuje provádět akce kolem nastavení aplikace díky zobrazení informací pomocí této šablony.
- *mainpage.order.tpl* – šablona pro sestavení dialogů týkajících se akcí kolem objednávek.
- *mainpage.person.tpl* – šablona sestavuje dialogy akcí pro manipulaci s osobními údaji.
- *mainpage.place.tpl* – šablona sestavuje dialogy při vkládání pořadů na videokazetu.
- *mainpage.program.tpl* – tato šablona umožňuje sestavení webové stránky s akcemi týkající se pořadů.
- *mainpage.tables.tpl* – šablona je potřebná k sestavení dialogů pro práci s číselníky, tzn. sestavují se přehledy, dialogy o vkládání, editování a mazání.
- *mainpage.user.tpl* – šablona pro sestavování dialogů a práci s uživateli informačního systému.
- *menu.login.tpl* – šablona pro sestavení přihlašovacího menu.
- *menu.menu.tpl* – šablona pro sestavení menu privátní části aplikace.
- *page.tpl* – šablona kostry celé aplikace.
- *cze.xml* – český jazykový balíček v XML formátu.
- *eng.xml* – anglický jazykový balíček v XML formátu.
- *javascript.js* – soubor obsahující funkce pro vykonávání akcí na straně klienta.
- *style.css* – soubor s kaskádovým stylem aplikace.
- adresář „*template/images*“ – v adresáři jsou uloženy obrázky, které slouží k přehlednějšímu vzhledu aplikace.



# Přiložené CD

Přiložené CD obsahuje:

- text diplomové práce v MS Word;
- uživatelskou a programátorskou dokumentaci v MS Word;
- zdrojové texty aplikace IS Audiovizuálního centra VŠB-TUO;
- použité programové prostředky (MySQL Database Server, Apache HTTP Server, PHP, MySQL Control Center, PSpad, phpMyAdmin).

Veškeré programy umístěné na tomto CD jsou volně přístupné na internetu a jejich použití není v rozporu s licenčními podmínkami.