

Pokročilé techniky v Oracle – speciální funkce pro multimediální data

Advanced techniques in Oracle - Special functions for multimedia data

Bc. Matúš Varačka

Diplomová práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Matuš Varačka**
Osobní číslo: **A11451**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Počítačové a komunikační systémy**
Forma studia: **prezenční**

Téma práce: **Pokročilé techniky v ORACLE – speciální funkce pro multimediální data**

Zásady pro vypracování:

1. Analyzujte problematiku a zpracujte literární rešerši na dané téma.
2. Navrhněte strukturu databáze – datového tržiště.
3. Popište nástroje pro přípravu multimediálních dat a naplnění základních datových struktur.
4. Popište možnosti a nástroje pro speciální vyhledávání multimediálních dat.
5. Implementujte navržené postupy do vytvořené aplikace.
6. Proveďte testování a vyhodnocení aplikace.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. RIORDAN, R. M. Vytváříme relační databázové aplikace. Praha : Computer Press, 2000. 280 s. ISBN 80-7226-360-9.
2. KYTE, Thomas. Oracle: návrh a tvorba aplikací. Vyd. 1. Brno: CP Books, 2005, 694 s. ISBN 8025105695.
3. LACKO, L'uboslav. Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle. Vyd. . Brno: Computer Press, 2003, 486 s. ISBN 8072269690.
4. LACKO, L'uboslav. Oracle: správa, programování a použití databázového systému. Vyd. 1. Praha: Computer Press, 2002, xiv, 464 s. ISBN 80-7226-699-3.
5. VRÁNA, Jakub. 1001 tipů a triků pro PHP. Vyd. 1. Brno: Computer Press, 2010, 456 s. ISBN 978-80-251-2940-1.
6. GUTMANS, Andi, Stig S?ther BAKKEN a Derick RETHANS. Mistrovství v PHP 5. Vyd. 1. Brno: CP Books, 2005, 655 s. ISBN 80-251-0799-x.
7. BRYLA, Bob a Kevin LONEY. Mistrovství v Oracle Database 11g. Vyd. 1. Brno: Computer Press, 2009, 700 s. ISBN 978-80-251-2189-4.
8. PROCHÁZKA, David. Oracle: průvodce správou, využitím a programováním nad databázovým systémem. 1. vyd. Praha: Grada, 2009, 168 s. ISBN 978-80-247-2762-2.

Vedoucí diplomové práce:

doc. Ing. Zdenka Prokopová, CSc.

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

26. února 2013

Termín odevzdání diplomové práce:

31. května 2013

Ve Zlíně dne 26. února 2013


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Karel Vlček, CSc.
ředitel ústavu

ABSTRAKT

Hlavným cieľom diplomovej práce bolo navrhnúť a realizovať špeciálne nástroje pre prácu s multimediálnymi dátami v systéme Oracle. Navrhnuté nástroje sú implementované v aplikácii, ktorá bude slúžiť pre prezentáciu hudobných skupín. Aplikácia je vytvorená v PHP a komunikuje s databázovým systémom Oracle, v ktorom sú uložené informácie o jednotlivých kapelách a ich skladbách. Diplomová práca vo svojej teoretickej časti opisuje výhody systému Oracle a približuje čitateľovi jednotlivé pojmy týkajúce sa návrhu aplikácie. Praktická časť opisuje jednotlivé funkcie ktoré sa v aplikácii nachádzajú. Taktiež oboznamuje čitateľa s postupom, ktorý viedol k vytvoreniu tejto aplikácie. Prácu dopĺňajú krátke ukážky zdrojového kódu, ktoré boli v práci použité.

Kľúčová slova: Oracle, databáza, PHP, internetové stránky, ORDImage, ORDAudio

ABSTRACT

The main aim of the thesis was to design and implement a special tools for working with multimedia data in Oracle. The proposed instruments are implemented in an application that will be used for the presentation of musical groups. The application is developed in PHP and communicates with the Oracle database system, which stores information about individual bands and their songs. Master thesis in its theoretical part describes the benefits of Oracle and presents the reader with the various concepts of design applications. The practical part describes the different features that are found in applications. It also introduces the reader to the procedure which led to the creation of this application. Work added little bits of source code that were used in the work.

Keywords: Oracle, database, PHP, websites, ORDImage, ORDAudio

Ďakujem vedúcej mojej diplomovej práce doc. Ing. Zdenke Prokopovej, CSc. z Ústavu počítačových a komunikačných systémov za odbornú pomoc. Ďalej ďakujem svojej rodine, svojej priateľke a kamarátom za poskytnutú podporu.

Motto: "Neverím na šťastie, na ktoré musíš čakať, kým do života ti konečne ono vstúpi, pokiaľ ideš za niečím, vytrvale s nadšením, to more sa pred tebou rozostúpi." (Majk Spirit)

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1. POUŽITÉ METÓDY A NÁSTROJE	11
1.1 RELAČNÁ DATABÁZA	11
1.1.1 Relačné podmienky	12
1.1.2 Integrita a integritné obmedzenia.....	13
1.1.3 Relačné vzťahy medzi tabuľkami.....	13
1.1.4 Normalizácia dát	14
1.2 SQL.....	15
1.2.1 Vlastnosti a výhody SQL.....	17
1.2.2 Druhy príkazov SQL.....	18
1.2.3 Druhy uskutočnenia príkazov	19
1.2.4 Obmedzenia dát	19
1.2.4.1 Obmedzenie NOT NULL.....	20
1.2.4.2 Obmedzenie UNIQUE	20
1.2.4.3 Obmedzenie PRIMARY KEY.....	20
1.2.4.4 Cudzí kľúč (FOREIGN KEY).....	21
1.2.4.5 Obmedzenie CHECK.....	21
1.2.5 Vytvorenie pohľadov SQL.....	21
1.3 ORACLE.....	22
1.3.1 Edičné rady.....	22
1.3.2 Technológia Oracle Text.....	23
1.3.3 Multimédia v Oracle.....	24
1.3.4 Práca s multimédiami	25
1.3.5 Vyhľadávanie obrázku	26
2. POUŽITÉ PROGRAMY	27
2.1 SQL DEVELOPER.....	27
2.2 APACHE	27
2.3 PSPAD.....	27
2.4 GITHUB	27
II PRAKTICKÁ ČÁST.....	28
3. PRÍPRAVA NA VÝVOJ APLIKÁCIE	29
3.1 DOTAZNÍK A JEHO VÝSLEDKY	29
3.2 INŠTALÁCIA A PREPOJENIE.....	30
4. NÁVRH APLIKÁCIE.....	32

4.1	POŽIADAVKY NA FUNKČNOSŤ	32
4.2	NÁVRH DATABÁZY	32
4.3	VZHEAD STRÁNKY	34
5.	VÝVOJ APLIKÁCIE.....	37
5.1	APLIKAČNÁ ČASŤ	37
5.1.1	Prihlásenie.....	37
5.1.2	Registrácia	38
5.1.3	Blog.....	39
5.1.4	Hodnotenie skupiny a blogu	40
5.1.5	Vyhľadávanie.....	41
5.1.6	Oakordovanie	41
5.1.7	Správy	43
5.2	DATABÁZA	44
5.2.1	Vytvorenie tabuliek	44
5.2.2	Trigger a sequence	45
5.2.3	Stránkovanie a obmedzenie príkazu Select	45
5.2.4	Editovanie profilu	46
5.3	PRÁCA S OBRÁZKOM	47
5.3.1	Vytvorenie a zmazanie priečinka	47
5.3.2	Procedúry	47
5.3.3	Úprava obrázkov	48
5.3.4	Process copy.....	50
5.3.4.1	Vytvorenie náhľadu	51
5.3.5	Podobnosť obrázkov.....	52
5.4	PRÁCA S HUDBOU	54
6.	SÚČASNÁ PODOBA A BUDÚCNOSŤ	55
6.1	FUNKCIONALITA.....	55
6.2	MOŽNOSTI ROZŠÍRENIA	56
	ZÁVĚR	57
	ZÁVĚR V ANGLIČTINĚ.....	59
	SEZNAM POUŽITÉ LITERATURY.....	61
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	62
	SEZNAM OBRÁZKŮ.....	63
	SEZNAM TABULEK	64

ÚVOD

V dnešnej dobe, kedy sa snažíme všetko digitalizovať, existuje množstvo databázových systémov, ktoré ponúkajú možnosť na uskladnenie.

Digitalizáciou šetríme priestor, uľahčujeme vyhľadávanie v dokumentoch zefektívňujeme archivovanie a často aj šetríme čas. Veľkou výhodou je poskytnúť rovnaký dokument v rovnakom čase viacerým osobám prostredníctvom internetu. Toto by sa nezaobišlo bez databázových systémov, do ktorých sa tieto údaje ukladajú.

Jedným z predstaviteľov databázových systémov je nepochybne systém Oracle. Jeho neodmysliteľnou výhodou, okrem toho že je dostupný aj vo voľnej licencií je to, že umožňuje ukladať aj rôzne typy multimediálnych dát, ku ktorým sa správa ako k bežným uloženým dátam.

Výhodou sú taktiež špeciálne nástroje, pomocou ktorých je možné tieto multimediálne dáta upravovať a pracovať s nimi.

Preto som sa rozhodol vytvoriť aplikáciu, ktorá bude slúžiť pre hudobné skupiny. V tejto aplikácii si budú mať hudobné skupiny vytvoriť svoj profil, využívať funkcie bežného portálu no zároveň im bude poskytnuté dátové úložisko pre ich skladby. Na vývoj aplikácie použijem databázový systém Oracle a poukážem hlavne na jeho výhody a špeciálne nástroje pri práci s multimédiami.

Predkladaná diplomová práca je prepojená s diplomovou prácou „Pokročilé techniky v ORACLE – úložisko multimediálnych dát“

TEORETICKÁ ČÁST

1. POUŽITÉ METÓDY A NÁSTROJE

Pri návrhu tejto aplikácie je nesmierne dôležitá voľba správnych programovacích nástrojov. Zvolené programovacie nástroje musia spĺňať všetky nároky, ktoré sa počas vývoja aplikácie môžu vyskytnúť. Programátor musí mať predstavu o tom, čo všetko bude aplikácia obnášať a na základe týchto poznatkov prebieha voľba nástrojov. Pri návrhu tejto aplikácie bolo jasné, že bude pozostávať z hodnôt, ktoré budú zadávať samotný užívateľ, rovnako ako aj z hodnôt, ktorý boli preddefinované. Všetky tieto hodnoty sa môžu neustále meniť. Preto bude táto aplikácia pozostávať z databáze, kde sa budú tieto hodnoty ukladať a odkiaľ sa budú načítavať. Keďže sa bude jednať o úložisko, kde sa budú nachádzať aj multimediálne dáta, bola zvolená databáza Oracle, ktorá prácu s multimediálnymi dátami bez problémov zvláda. Ako rozhranie medzi databázou a užívateľom bude použitý serverovo orientovaný skriptovací jazyk PHP.

1.1 Relačná databáza

Na úvod by som priblížil pojem databáza, od ktorého sa odvíja hlavná podstata práce.

Databáza je súbor dát, ktoré sú organizované v štruktúrovanom formáte, ktorý je definovaný ako metadáta popisujúce jeho štruktúru. Metadáta môžeme považovať za informácie o uložených dátach, ktoré definujú, ako sú dáta v databáze uložené.

Všetky dáta, ktoré máme uložené v osobnom počítači, sú uložené v nejakých logických a fyzických štruktúrach.

Relácia v matematickom pojatí sa dá znázorniť ako dvojrozmerné pole, v databázovom pojatí sa relácia označuje pojmom tabuľka.

Dáta v databáze sú uložené v tabuľkách. Každý riadok v tabuľke predstavuje jeden záznam. Jednotlivé stĺpce odpovedajú jednotlivým atribútom, ktoré sú v databáze zadané. Každý takýto atribút má priradený svoj názov, dátový typ a môže mať definované ďalšie vlastnosti. Priesečníkom medzi atribútom (stĺpcom) a záznamom (riadkom) je prvok, v ktorom sa nachádza konkrétna hodnota pre konkrétny údaj. Z matematického hľadiska by sa záznam dal popísať ako vektor jednotlivých hodnôt.

Pre správne fungovanie relačnej databázy by mal byť každý riadok identifikovaný pomocou primárneho kľúča.

Primárny kľúč je jednou z hodnôt pre každý záznam, ktorý musí byť jedinečný. Môže ísť o číselný alebo znakový identifikátor. Na základe tejto hodnoty bude celý záznam charakteristický a dostupný k vyhľadaniu. Môže ísť napríklad o identifikačné číslo záznamu, ktorý je pre každý záznam jedinečný.

Ako príklad si môžeme uviesť tabuľku *Skupina*, kde budú atribúty ID_skupiny, Názov_skupiny a Žáner. V jednotlivých záznamoch budú údaje o konkrétnej skupine.

ID_skupiny	Názov_skupiny	Žáner
1	Deep Purple	rock
2	Lucie	rock

Tab. č. 1 – Ukážka tabuľky *Skupina*

Pojem relačná databáza sa dá zjednodušene vysvetliť ako sada nástrojov, ktorá je určená pre efektívne a spoľahlivé ukladanie dát a taktiež pre manipuláciu s nimi.

Pri relačných databázach platia tieto vlastnosti:

- homogenita stĺpcov
- každý údaj je atomickou položkou
- na poradí riadkov nezáleží
- na poradí stĺpcov nezáleží
- každý riadok tabuľky je jednoznačne identifikovateľný hodnotami jedného alebo viacerých atribútov [1, str. 23]

1.1.1 Relačné podmienky

Aby mohla existovať a správne fungovať relačná databáza, je dôležité dodržať určité podmienky. Medzi minimálne podmienky, ktoré treba dodržať patria tieto:

- všetky dáta v databáze musia byť uložené v tabuľkách
- fyzická štruktúra dát a ich uloženia sú nezávislé a od užívateľa oddelené
- predpokladá sa existencia databázového jazyka, ktorý by umožňoval realizovať minimálne operácie s dátami. Medzi tieto operácie patrí selektovanie, spojenie, projekcia a obmedzenia

Okrem týchto minimálnych podmienok sú dané aj pravidlá, ktoré definoval Dr. E. F. Codd. Ide o 12 pravidiel pre relačné databázové systémy. Z týchto pravidiel spomeniem niektoré, ako napríklad:

- dáta v relačnej databáze musia byť reprezentované explicitne na logickej úrovni
- dáta musia byť prístupné pomocou kombinácie názvu tabuľky, názvu stĺpca a hodnoty primárneho kľúča
- databázový jazyk musí byť jednoduchý a užívateľsky prívetivý, pričom musí umožňovať prístup k dátam ako aj manipuláciu s nimi [2, strana 14]

1.1.2 Integrita a integritné obmedzenia

Integrita databázy znamená, že dáta, ktoré sú v nej uložené, musia byť konzistentné. To znamená, že je možné zadávať len dáta, ktoré do databázy patria a taktiež nie je možné, aby sa niektoré dáta stratili.

K zaisteniu integrity sa používajú integritné obmedzenia, ktoré zabránia strate či poškodeniu dát pri ich akejkolvek úprave. Integritné obmedzenia taktiež zabezpečia, že sa v databáze nebudú nachádzať dáta, ktoré tam nepatria. Napr. v prípade, že vymažeme niektorý záznam, odstránia sa všetky záznamy, ktoré sa k tomuto záznamu viažu. [1, str. 24]

1.1.3 Relačné vzťahy medzi tabuľkami

Dôležitou súčasťou všetkých relačných databáz je ich vzájomné prepojenie. Takéto prepojenie sa taktiež nazýva vzťahom alebo reláciou medzi tabuľkami. Vzťahy prepojujú tabuľky zmysluplne a zaisťujú integritu dát. Zabezpečujú, aby zmena, ktorá sa vykoná v jednej tabuľke nemala negatívny dopad na dáta v inom vzťahu.

Poznáme tri základné vzťahy:

- vzťah jedna k jednej (1:1)
- vzťah jednej k viacerým (1:N)
- vzťah viacerých k viacerým (N:M)

Vzt'ah 1:1

Je to vzt'ah medzi dvoma tabuľkami, kedy záznam z prvej tabuľky je viazaný iba k jednému záznamu druhej tabuľky a naopak, záznam z druhej tabuľky je viazaný iba k jednému záznamu z prvej tabuľky.

Vzt'ah 1:N

Vzt'ah medzi dvoma tabuľkami, kedy záznam z prvej tabuľky je viazaný k jednej alebo viacerým záznamom z druhej tabuľky, ale záznam v druhej tabuľky je viazaný iba k jednému záznamu z prvej tabuľky.

Vzt'ah N:M

Vzt'ah, kedy záznam v prvej tabuľke je viazaný k jednému alebo viacerým záznamom v druhej tabuľke, a taktiež záznam z druhej tabuľky je viazaný k jednej alebo viacerým záznamom z prvej tabuľky. [3, str. 27]

1.1.4 Normalizácia dát

Proces, ktorý vedie k odstráneniu redundancie (duplicit), zefektívneniu práce s tabuľkami a zjednodušeniu databázy.

Tabuľka sa môže podľa normalizácie dát zaradiť do jednej z piatich normálových foriem. Z hľadiska normalizácie sa dá povedať, že čím vyššiu formu normálnosti tabuľka dosiahne, tým je práca s ňou ľahšia.

Nultá normálová forma (0NF)

Ak sa tabuľka nachádza v nulte normálovej forme, je z hľadiska dotazov takmer neriešiteľná. Dotazy v nej sa dajú len s problémami vykonávať. Tabuľka sa v tejto forme nachádza vtedy, ak aspoň jedno z jej polí pozostáva z viacerých hodnôt a teda nie je atomické. Príkladom môže byť, tabuľka, v ktorej sú uložené meno, priezvisko a adresa. Nultá forma je dosiahnutá vtedy, ak pole adresa je spojením viacerých údajov, ako napríklad adresa, číslo domu a mesto. V tomto prípade je dotazovanie sa v tabuľke neriešiteľné.

Prvá normálová forma (1NF)

Prvá normálová forma je splnená vtedy, ak všetky polia v tabuľke sú už ďalej nedeliteľné. To znamená, že pre každý údaj je v tabuľke vytvorený zvlášť atribút. Pri aplikovaní na

predchádzajúci príklad by to znamenalo, že tabuľka by bola doplnená o atribúty ulica, číslo domu a mesto.

Druhá normálová forma (2NF)

Tabuľka spĺňa druhú normálovú formu vtedy, ak je splnená prvá normálová forma a zároveň všetky atribúty konkrétneho záznamu sú úplne závislé na primárnom kľúči daného záznamu. Druhá forma zasahuje iba tabuľky, ktorých primárny kľúč je tvorený viac ako jedným poľom (zložený primárny kľúč). Ak má tabuľka primárny kľúč tvorený iba jedným poľom, táto podmienka je splnená automaticky.

Tretia normálová forma (3NF)

Táto forma je splnená, ak sa tabuľka nachádza v druhej normálovej forme a súčasne v nej neexistujú žiadne závislosti nekľúčových atribútov tabuľky.

Štvrtá normálová forma (4NF)

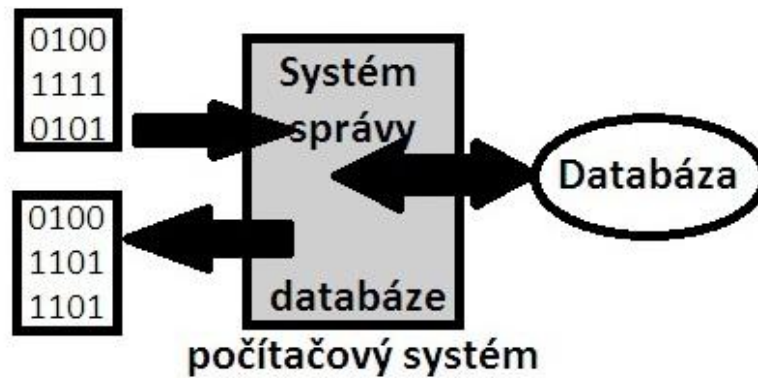
Štvrtá forma sa dosiahne vtedy, ak sa tabuľka nachádza v tretej forme a popritom aj tabuľka popisuje len jednu jedínú súvislosť alebo fakt, t.j. polia tvoriace zložený primárny kľúč musia byť nezávislé.

Piata normálová forma (5NF)

Aby sa tabuľka dostala do piatej normálovej formy, musí byť splnená štvrtá normálna forma a zároveň primárny kľúč tvorený z troch a viacerých polí nesmie obsahovať cyklické závislosti. [2, str. 19]

1.2 SQL

Nástrojom pre správu relačných databázových systémov je jazyk SQL. Spoločne tvoria jednu z najdôležitejších základných technológií počítačového priemyslu. Jazyk SQL je nástroj, pomocou ktorého môžeme organizovať, riadiť a získavať dáta, ktoré sú uložené v databáze. SQL znamená Structured Query Language. Ide teda o počítačový jazyk, ktorý používame pre komunikáciu s relačnou databázou.



Obrázok č. 1 – Použitie jazyka SQL pre prístup k databáze

Jazyk SQL nie je len dotazovací jazyk, ale spĺňa aj mnoho ďalších funkcií, ako napríklad:

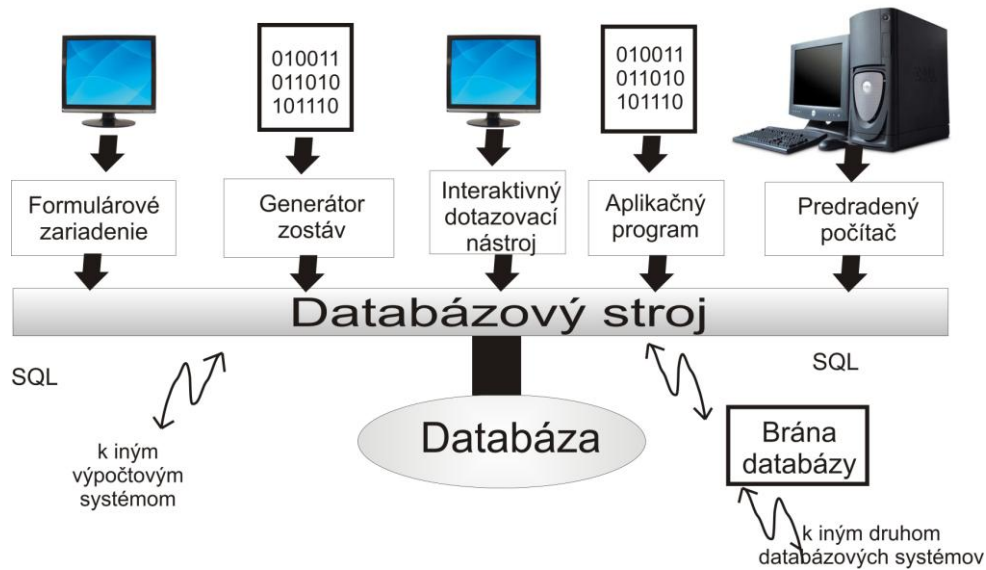
- Definícia dát
- Získavanie dát
- Manipulácia s dátami
- Riadenie prístupu
- Integrita dát

Práve pre tieto všetky funkcie, ktoré SQL obsahuje, sa dá nazvať ako komplexný jazyk pre riadenie a ovplyvňovanie databázového systému.

Samotný jazyk SQL nie je produkt, ktorý by sa dal zakúpiť. Ide o integrovanú súčasť databázového systému, kde slúži ako jazyk a zároveň nástroj pre komunikáciu s ním.

[4, str. 8]

1.2.1 Vlastnosti a výhody SQL



Obrázok č. 2 – Databázový stroj

Na obrázku č. 2 je znázornený „databázový stroj“, ktorý predstavuje srdce databázového riadiaceho systému. Je zodpovedný za členenie, ukladanie a získavanie dát.

SQL spĺňa mnohé funkcie. Vďaka tomu sa dá povedať, že SQL je:

- Interaktívny dotazovací jazyk. Užívateľ zadáva príkazy SQL pomocou iného programu, pomocou ktorých sú mu zobrazené získané dáta na obrazovke
- Databázový programovací jazyk. Programátori môžu začleniť SQL príkazy do svojich vlastných aplikačných programov. Vďaka tomuto získajú dáta a prístup k nim pre svoje programy.
- Jazyk typu klient/server. Ide o veľmi populárnu architektúru, kde programy, ktoré sú spustené na osobných počítačoch používajú jazyk SQL pre komunikáciu s databázovými servermi, ktoré uchovávajú zdieľané dáta.
- Jazyk pre prístup k dátam na internete. Internetové webové servery, ktoré pracujú so spoločnými dátami, používajú jazyk SQL na prácu s nimi a pre prístup k spoločnej databáze.

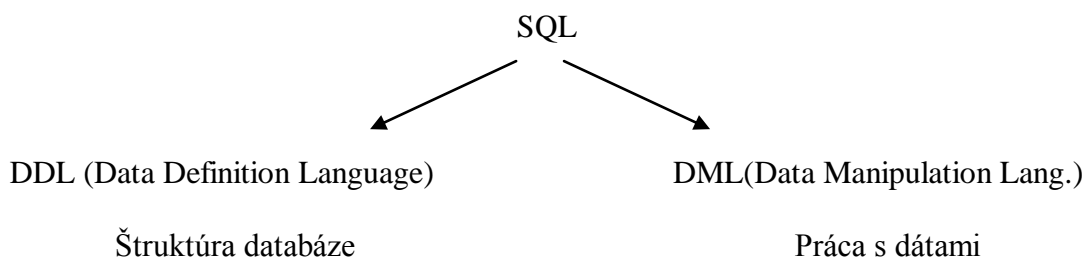
Okrem spomenutých funkcií, ktorými jazyk SQL disponuje, má aj ďalšiu radu výhod.

Z množstva výhod spomeniem aspoň niektoré, a to:

- nezávislosť na predajcovi
- prenositeľnosť medzi počítačovými systémami
- štandardy jazyka SQL
- vysokoúrovňová štruktúra podobná angličtine
- viacnásobné pohľady na dáta
- kompletný databázový jazyk
- architektúra klient/server
- dynamické definovanie dát
- rozšíriteľnosť a objektová technológia [4, str. 30]

1.2.2 Druhy príkazov SQL

Keďže jazyk SQL môžeme považovať za kompletný jazyk, pomocou ktorého môžeme riadiť dáta, zabezpečovať tieto objekty, vytvárať a upravovať tabuľky, obsahuje množstvo príkazov. Tieto príkazy by sa dali kategorizovať na tie, ktoré sa starajú o samotné dáta a tie, ktoré pracujú so štruktúrou databázy.



Obrázok č. 3 – Rozdelenie SQL

Jazyk definície dát

Príkazy, ktoré kategorizačne patria do sekcie DDL sa používajú pre vytvorenie, úpravu alebo samotné odstránenie databázových objektov. Medzi tieto databázové objekty patria tabuľky, pohľady, schémy či trigger. Najčastejšie používanými príkazmi DDL sekcie sú príkazy CREATE, ALTER A DROP.

Jazyk pre manipuláciu s dátami

Do tejto kategórie patri príkazy, ktoré sa používajú k pridávaniu, vyhľadávaniu, úpravám alebo odstraňovaniu dát, ktoré sú v databáze uložené. Hlavnými príkazmi tejto sekcie sú príkazy SELECT, INSERT, UPDATE A DELETE.

1.2.3 Druhy uskutočnenia príkazov

Štandard, ktorý špecifikuje možnosti pre používanie jazyka, špecifikuje taktiež aj spôsoby, ako sa môže SQL príkaz uskutočniť. Definuje štyri metódy uskutočnenia:

- **Priame uskutočnenie.** Pomocou priameho uskutočnenia je umožnená komunikácia priamo z užívateľského rozhrania aplikácie, ako napríklad SQL Plus Worksheet v databáze Oracle. Výsledky týchto dotaz sú spracované a vracané takmer okamžite. Rýchlosť určuje taktiež procesor a obmedzenie databázy. Ide o rýchly spôsob kontroly dát a overenia pripojenia.
- **Integrovaný jazyk SQL.** Pri tejto metóde sú zadané príkazy integrované do hostiteľského programovacieho jazyka. Tieto príkazy môžu byť vložené napríklad do aplikácie vytvorenej v programovacom jazyku C.
- **Pripojovanie modulov.** Táto metóda umožňuje vytvárať bloky príkazov, ktoré sú oddelené od hostiteľského programovacieho jazyka.
- **Rozhranie CLI (Call-level interface).** V tejto metóde je možné vykonávať príkazy SQL pomocou rozhrania predávaním príkazov SQL, ktoré sú predávané ako hodnoty parametrov podprogramov. [3, str. 33]

1.2.4 Obmedzenia dát

Ako bolo v časti o relačných databázach spomenuté, je dôležité, aby dáta boli konzistentné. To má za dôsledok obmedzenie integrity. V jazyku SQL môžeme definovať niekoľko typov obmedzení, ktorým dosiahneme, aby uložené dáta boli správne a konzistentné.

1.2.4.1 Obmedzenie *NOT NULL*

Hodnota *null*, ktorá sa môže vyskytovať v databáze, neznamená hodnotu rovnú nule. Táto hodnota sa udáva v prípade, že hodnota neexistuje prípadne že nebola definovaná. Ide teda o údaj, že dátová hodnota chýba.

Obmedzenie *NOT NULL* môže byť nastavené pre konkrétne atribúty a môžeme ním zadefinovať tabuľke, kde je zadanie určitej hodnoty potrebné. Týmto vie tabuľka rozpoznať, či môže byť záznam s prijatou prázdnu hodnotou na konkrétnu pozíciu prijatý a či nie. V prípade, že niektorý z atribútov nie je nastavený na obmedzenie *NOT NULL* a bude do neho zapísaná prázdna hodnota, bude sa v tomto poli nachádzať hodnota *null*.

1.2.4.2 Obmedzenie *UNIQUE*

Pre obmedzenie jednoznačnosti môžeme použiť dvojicu obmedzení a to *UNIQUE* alebo *PRIMARY KEY*. Obmedzenie *UNIQUE* umožňuje žiadať, aby stĺpec alebo skupina stĺpcov, pre ktoré bolo toto obmedzenie definované, obsahovali iba jednoznačné, unikátne hodnoty. Toto obmedzenie nám zabezpečí, že v celej tabuľke sa bude hodnota poľa, pre ktorú bolo obmedzenie zadané, vyskytovať len jedenkrát. Je to napr. možnosť, ako môže programátor zabrániť tomu, aby sa v tabuľke vyskytovali rovnaké prihlasovacie mená viac krát. Ak nastaví atribút *prihlasovacie meno* na hodnotu *UNIQUE*, bude mať istotu, že s rovnakým prihlasovacím menom nebudú môcť vystupovať viaceré osoby.

1.2.4.3 Obmedzenie *PRIMARY KEY*

Obmedzením *PRIMARY KEY* môžeme dosiahnuť jednoznačnosť údajov. Toto obmedzenie, rovnako ako *UNIQUE* zabraňuje duplicitným údajom v tabuľke. Hlavný rozdiel je v dvoch funkciách, ktoré sa naň vzťahujú.

- Stĺpec, ktorý bol definovaný obmedzením *PRIMARY KEY*, nemôže obsahovať hodnotu *null*. Aj keď toto obmedzenie nebude zadefinované. Každý záznam musí mať na poste atribútu označenom *PRIMARY KEY* konkrétnu, určitú a jedinečnú hodnotu.
- Pre každú tabuľku je možné definovať iba jedno toto obmedzenie. [3, str. 86]

1.2.4.4 Cudzí kľúč (FOREIGN KEY)

Predchádzajúce obmedzenia sa týkali zabezpečenia integrity dát. Cudzíe kľúče sa však zaoberajú tým, ako dáta jednej tabuľky súvisia s dátami v druhej tabuľke. Sú známe ako referenčné obmedzenia.

Referenčná integrita bráni tomu, aby zmena dát v jednej tabuľke mala negatívny vplyv na druhú tabuľku, ktorá je s prvou prepojená. V prípade, ak by sme z jednej tabuľky odstránili či zmenili dáta, ktoré sú prepojené a odkazujú sa do druhej tabuľky, narušila by sa referenčná integrita. Cudzí kľúč bráni teda tomu, aby nebola v tabuľke vykonaná činnosť, ktorá by ovplyvnila dáta označené týmto kľúčom. [3, str. 95]

1.2.4.5 Obmedzenie CHECK

Toto obmedzenie si definuje samotný administrátor. Popisuje obmedzenie, ktoré môže byť dané pre tabuľku, stĺpec či tvrdenia. Obmedzenie slúži pre konkrétne zadanie hodnôt, ktoré môžu byť do daného poľa zadané. Môže nadefinovať rozsah hodnôt (napr. 10-100), zoznam hodnôt (jazz, blues, rock, pop) alebo mnoho ďalších podmienok, ktoré budú pri vkladaní hodnôt kontrolované.

1.2.5 Vytvorenie pohľadov SQL

Tabuľky databáze možno rozdeliť do štyroch kategórií. Sú to kategórie:

- Trvalé základné tabuľky
- Globálne dočasné tabuľky
- Vytvorené miestne tabuľky
- Deklarované miestne dočasné tabuľky

Dáta SQL databáze sú skutočne uložené len v trvalej základnej tabuľke. Nie vždy je však užitočné alebo bezpečné vypisovať všetky údaje, ktoré sa v tejto tabuľke nachádzajú. Z tohto dôvodu je SQL podporovaný vytvorením pohľadov.

Pohľad je virtuálna tabuľka, ktorej definícia je uložená ako objekt schémy. V týchto pohľadoch nie sú uložené dáta. Existuje iba definícia tabuľky. Táto definícia určuje, ktoré informácie z jednej alebo viacerých tabuliek majú byť zobrazené. Po vytvorení pohľadu nie je žiaden problém ho vyvolať použitím jeho názvu v dotaze, ako by išlo o základnú tabuľku.

Pohľad je určitým spôsobom prechod medzi trvalou základnou tabuľkou, v ktorej sú uložené dáta a odvodenou tabuľkou, v ktorej sa zobrazia výsledky. Výsledky, ktorú sú zobrazené po volaní pohľadu nie sú nikde uložené. Ide iba o odvodenie zo základných existujúcich tabuliek.

Hlavnou výhodou pohľadov je, že môžeme definovať rôzne zložité dotazy a uložiť ich v definícii pohľadu. Pri zavolaní pohľadu bude tento dotaz z definície načítaný a vykonaný. [4, str. 233]

1.3 Oracle

Oracle je systém riadenia dát v databáze. Databáza taktiež poskytuje zabezpečenie proti nepovolaným vstupom a k prístupu k citlivým informáciám. Databáza sa skladá z dvoch kategórií, a to databázové a nedatabázové súbory. Medzi tie databázové patria dáta a metadáta, nedatabázové obsahujú inicializačné parametre, protokolovacie informácie a podobne. [5, str. 30]

Oracle je moderný a multiplatformný databázový systém, ktorý má veľmi pokročilé možnosti spracovania dát. Taktiež disponuje aj vysokým výkonom a ľahkou škálovateľnosťou.

Tento systém vyvíja firma Oracle Corporation, čo je najväčšia, firmám dodávajúca software, spoločnosť na svete. Jej ročný obrat je okolo 10,2 miliardy USD a okrem podnikových aplikácií ponúka aj aplikačný server, či nástroje pre podnikovú spoluprácu.

V roku 1994 bola založená Česká pobočka spoločnosti za účelom priblížiť služby a produkty tejto firmy aj zákazníkom Českej republiky. [1, str. 33]

1.3.1 Edičné rady

V súčasnosti spoločnosť Oracle disponuje 4 hlavnými edíciami. Na webovej stránke spoločnosti, www.oracle.com, sú na stiahnutie dostupné verzie Oracle Database 11g Express Edition a Oracle Database 11g Enterprise/Standard Edition.

V prípade edície 11g Express Edition ide o bezplatnú verziu, ktorú je možné stiahnuť z webovej stránky spoločnosti. Aj napriek bezplatnosti ponúka množstvo pokročilých funkcií. Je určená hlavne pre malé firmy a začínajúcich vývojárov. V tejto edícii je plne dostupná podpora XML, práca s priestorovými dátami alebo takzvané fulltextové

vyhľadávanie. Ponúka až 11 GB pre uchovávanie užívateľských dát a využije maximálne 1 GB z pamäte RAM a 1 CPU. Vďaka menšej veľkosti je rýchlejšie dostupná ku stiahnutiu a taktiež jednoduchšia k inštalácii.

Verzia 11g Standard Edition disponuje väčším množstvom funkcií a je určená hlavne pre tímové spolupráce, pre správu dát, ktoré sa nachádzajú na hranici jednotlivých oddelení a pracovných skupín. Táto edícia môže byť použitá od malých firiem s jednoduchou štruktúrou až po veľmi rozšírené a členité štruktúry veľkých firiem. [5, str. 80]

1.3.2 Technológia Oracle Text

Ako bolo spomenuté, databázová platforma Oracle má široké spektrum možností. Jednou zo špeciálnych nástrojov je Oracle Text. Je určený predovšetkým k vyhľadávaniu textu v rozsiahlych textových poliach. Toto vyhľadanie je možné aj pomocou klauzule *WHERE*, no jej zostavenie môže byť niekedy veľmi náročné.

Veľké množstvo dát je uložených práve vo forme textu. Môže ísť napríklad o komentáre či texty piesní. Balíček s názvom Oracle Text umožňuje tieto rozsiahle textové dáta indexovať. Indexovanie prebieha pomocou metódy CTXCAT. Týmto sa značne uľahčí vyhľadanie v textových dátach. Príkladom môže byť vyhľadanie v kuchárskej knihe, kde názov jedla pozostáva s viacerých položiek. Ide o názov hlavného jedla, prílohu, názov omáčky a podobne. Na základe metódy Oracle Text je vyhľadanie aj v takto rozsiahlom textovom poli jednoduché.

Základný syntax indexovania popíšem na nasledujúcom príklade.

Vytvoríme tabuľku, ktorá bude popisovať kuchársku knihu.

```
1 CREATE TABLE jedlo (  
2 Id_jedla NUMBER PRIMARY KEY,  
3 Nazov_jedla VARCHAR2(150)  
4 );
```

Na stĺpec, kde je zadaný názov jedla, vytvoríme index CTXCAT.

```
5 CATSEARCH  
6 (  
7 [schema.] stlpec,  
8 text_query VARCHAR2,  
9 structured_query VARCHAR2,
```

```
10 )
```

```
11 RETURN NUMBER;
```

V tomto zápise *text_query* predstavuje podmienku pre vyhľadávanie a *structured_query* umožňuje využiť štandardný syntax SQL, ako napríklad triedenie.

V prípade, že z tabuľky jedlo chceme nájsť záznam, ktorý obsahuje ako súčasť jedla slovo vajce, (napríklad zeleninový šalát, kuracie mäso, kukurica, vajce), môže samotné vyhľadávanie vyzerat' nasledovne:

```
12 SELECT * FROM jedlo WHERE CATSEARCH(nazov_jedla, 'vajce', NULL) 0;
```

Pre aplikovanie príkazu na vyhľadávanie viacerých slov, napríklad vajce a mäso, stačí do príkazu doplniť tento názov.

```
13 SELECT * FROM jedlo WHERE CATSEARCH(nazov_jedla, 'mäso vajce', NULL) 0;
```

Oracle Text okrem tejto funkcie ponúka ďalšie množstvo funkcií na prácu s textom, ako napríklad indexovanie slov, vyhľadávanie počtu výskytu zvolených slov v texte a podobne. [2, str. 417]

1.3.3 Multimédia v Oracle

Prvé databázové systémy umožňovali ukladať a vedeli pracovať len so základnými dátovými typmi, ako sú text či čísla. Oracle však toho dokáže omnoho viac. Dokáže spracovať okrem základných štruktúrovaných dát aj multimediálne dáta.

Práca s multimediálnymi dátami prebieha pomocou komponentu *InterMedia*, ktorý je licenčne súčasťou všetkých edícií Oracle. Argumenty, ktoré sú určené pre štruktúrované ale aj neštruktúrované dáta sú rovnaké. Či už ide o prácu s dokumentmi, XML alebo práve s multimediálnymi dátami.

Na princípe uloženia multimediálnych dát sú založené napríklad realitné kancelárie, ktoré ukladajú fotky priestorov, fotoalbumy na sociálnych sieťach a podobne.

Hlavným prínosom jednotného úložiska pre štruktúrované ale aj neštruktúrované dáta je v tom, že môžeme s nimi pracovať rovnakým spôsobom. Jednotnosť sa prejavuje hlavne v nasledujúcich funkciách:

- Jednotné API – SQL, sa používa pre prístup ako k štruktúrovaným, tak aj k neštruktúrovaným dátam. V jednom dotaze je preto možné kombinovať dotazy pre štruktúrované alebo aj multimediálne dáta.
- Bezpečnosť – je dôležité, aby multimediálne dáta boli rovnako zabezpečené ako tie so štruktúrou. Zabraňuje to tomu, aby sa napríklad k fotoalbumu užívateľa dostal niekto cudzí.
- Správa – jednotný prístup k obom druhom dát je veľmi užitočný pri zálohovaní a obnove dát. Multimediálne dáta sa zálohujú súčasne so zálohou celej databázy.

1.3.4 Práca s multimédiami

Keďže sa jedná o neštruktúrované dáta, pre prácu s nimi sú na platforme Oracle dostupné tri základné metódy.

- ORDImage, ktorý slúži na prácu s obrázkami. Je schopný pracovať so širokým spektrom rôznych formátov, ako napríklad JPEG, GIF, TIFF, PNG a podobne
- ORDAudio, je určený pre prácu s audio záznamami. Medzi podporované formáty patrí MPEG, WAV, AIFF a podobne
- ORDVideo, pomocou ktorého môžeme pracovať a pristupovať k multimediálnym formátom, kde podporuje napríklad QuickTime, AVI, MPEG a ďalšie

Pri všetkých vymenovaných metódach je práca so súbormi rovnaká. Rozdiel je iba v niektorých špecifických metódach. Multimediálne objektové typy sa dajú použiť pri definícii tabuliek alebo aj pri SQL príkazoch.

Datový typ nemá na starosti len uloženie súboru, ale zabezpečuje taktiež prístup k nim metadatám. Pomocou tohto prístupu je napríklad možné zistiť veľkosť videa, rozlíšenie a počet pixelov obrázku alebo typ kompresie.

Ďalšou výhodou je, že pomocou týchto dátových typov môžeme multimediálne súbory upravovať, ako napríklad konverzovať formáty, meniť veľkosť, či rôzne výrezy v prípade obrázkov.

Keďže databázový server nie je zodpovedný za zobrazenie dát, vstup a aj výstup dát je nutné implementovať vo vlastnej aplikácii, ktorá bude k týmto dátam pristupovať.

1.3.5 Vyhľadávanie obrázku

Oracle obsahuje taktiež špeciálne nástroje, kde si môžeme uviesť ako príklad pre vyhľadávanie multimedialných dát. V tomto prípade pôjde o vyhľadávanie obrázku na základe jeho obsahu.

Ako už bolo spomenuté, špeciálne dátové typy určené na prácu s multimédiami dokážu pristupovať aj k metadátam. Na tejto skutočnosti je založené aj vyhľadávanie na základe obsahu obrázku. Aj keď databáza nedokáže obrázky zobrazit', dokáže pristupovať a tým pádom porovnávať ich obsah.

Platforma Oracle pristupuje k obrázku a dokáže si vytvoriť jeho signatúru – popis. Tento popis potom porovnáva s ostatnými obrázkami, na základe ktorej dokáže nájsť rovnaký či podobný obrázok.

Databáza používa k popisu tieto charakteristiky:

- farbu (rozloženie farieb v obrázku)
- textúru (jednoduché opakujúce sa vzory)
- tvar (plocha, ktorá je vyplnená rovnakou farbou)
- rozloženie týchto charakteristík v priestore

Porovnávanie obrázkov prebieha v celku. Nie je preto možné, aby systém rozpoznal rovnakú osobu na dvoch rozličných obrázkoch. Dokáže iba rozpoznať dva rovnaké obrázky ako celky, nie rovnaké objekty na nich.

Pri vyhľadávaní je možné špecifikovať váhu zhody. Týmto dosiahneme nájdenie keď nie úplne rovnakých, tak podobných obrázkov.

Vyhľadávanie je taktiež možné zrýchliť pridelením indexu nad vytvoreným popisom obrázku. [6]

2. POUŽITÉ PROGRAMY

K vytvoreniu webovej aplikácie som používal niekoľko programov, z ktorých každý bol použitý na inú časť práce. Rozhodol som sa použiť programy SQL Developer, Apache, PSPad a GitHub, nakoľko spĺňali všetky moje požiadavky a sú voľne dostupné.

2.1 SQL Developer

SQL Developer je voľne šíriteľné vývojové prostredie, ktoré zjednodušuje vývoj a správu databáze Oracle. SQL Developer je väčšinou súčasťou verzií Oracle, ktoré sú k dispozícii na stiahnutie. SQL Developer ponúka kompletný vývoj PL/SQL aplikácií, listu pre spúšťanie dotazov a skriptov, konzolu pre správu databázy a kompletné modelovanie dát.

2.2 Apache

Apache je webový server, ktorý bol pôvodne napísaný pre UNIX. Dnes je jedným z najrozšírenejších webových serverov. Jeho ovládanie nie je cez grafické rozhranie, ale cez konfiguračné súbory. Inštalácia Apache môže byť pre niekoho pomerne zložitá, no po úspešnom nainštalovaní sa stáva veľmi kvalitným webovým serverom.

2.3 PSPad

Voľne šíriteľný univerzálny editor pre MS Windows. Je vhodný pre vytváranie webových stránok, keďže obsahuje veľké množstvo unikátnych funkcií, ktoré ušetria veľa času. Ďalšími výhodami je, že dokáže zachytávať a rozdeľovať výstupy kompilácie.

2.4 GitHub

Projekt bol založený na práci dvoch študentov, ktorí pracovali na projekte súčasne. Použitím GitHub je možné zdrojový kód zdieľať, čo uľahčí komunikáciu pri zložitejších projektoch, na ktorých pracuje väčší počet ľudí.

GitHub je webová hostingová služba pre projekty vývoja softvéru. Po založení účtu je užívateľovi poskytnuté úložisko, na ktorom môže so svojimi spolupracovníkmi zdieľať zdrojový kód. Tento kód je možné ukladať do priečinka na lokálnom disku určenom pre GitHub, čo zabezpečuje automatické aktualizovanie upraveného kódu.

PRAKTICKÁ ČÁST

3. PRÍPRAVA NA VÝVOJ APLIKÁCIE

Pred samotným vývojom aplikácie bolo nutné si stanoviť, aké funkcie od aplikácie očakávame a pre koho bude určená. Na základe týchto zistení bolo možné rozhodnúť sa pre správny softvér a návrh celej aplikácie.

3.1 Dotazník a jeho výsledky

Pred začiatkom vývoja som si navrhol funkcie, pre ktoré by mala byť aplikácia určená. Keďže sa poznám s ľuďmi, ktorí sa pohybujú v hudobných kruhoch, rozhodol som sa na nich obrátiť s dotazníkom, v ktorom som ich žiadal o ich návrhy na funkcie, ktoré by v podobnej aplikácii chceli, využívali a ktoré im na už existujúcich aplikáciách chýbajú. Dotazníkom som oslovil hudobníkov, členov hudobných skupín, organizátorov hudobných festivalov ale aj hudobníkov amatérov.

Dotazník sa skladal z dvoch častí. Prvá časť bola zameraná na okomentovanie a číselné ohodnotenie dôležitosti mnou navrhnutých funkcií. V druhej časti bol priestor pre doplnenie ďalších funkcií, ktoré opýtaným chýbajú a ktoré by radi v novej aplikácii využívali.

Takmer všetky z navrhnutých funkcií boli ohodnotené kladne, z čoho vyplývalo, že aplikácia s navrhnutou funkcionalitou by bola užitočná a našla by si svojich návštevníkov.

Na obrázku č. 4 sú zobrazené odpovede na funkcie „Oblíbené kapely a ich novinky“ a „Chat skupín“. Na prvú funkciu boli všetky odpovede kladné. Na druhú zobrazenú funkciu boli odpovede nerozhodné, na základe čoho som sa rozhodol túto funkciu, chat skupín do aplikácie nezaradiť.



Obrázok č. 4 – Grafické znázornenie výsledkov dotazníka

Z druhej časti, kde bol priestor na návrh vlastných funkcií, bola najviac spomínaná funkcia na oakordovanie skladieb a preto som sa rozhodol túto funkciu použiť a začleniť do návrhu.

Na základe týchto zistení som si spísal všetky funkcie, ktoré by mala aplikácia obsahovať a bol som pripravený na samotný návrh. Dotazník obsahoval 19 otázok a prieskumu sa zúčastnilo 17 osôb z hudobnej oblasti.

3.2 Inštalácia a prepojenie

Pred samotným vývojom je potrebné zvoliť správny softvér a nástroje, pomocou ktorých bude vývoj aplikácie realizovaný. Z dôvodu, že aplikácia bude pracovať aj s multimediálnymi dátami, som sa rozhodol použiť databázový systém Oracle.

Komunikácia medzi databázou a užívateľom bude realizovaná pomocou webového rozhrania, ktoré som vytvoril v kombinácii PHP a HTML. Aby mohli tieto nástroje spolu komunikovať na lokálnom počítači, je potrebné nainštalovať Apache. Inštalácia bola pomerne náročná, keďže bolo problematické nájsť správnu kombináciu verzií Apache, PHP a Oracle, ktoré by medzi sebou bez problémou komunikovali.

Výsledná kombinácia spočívala v nasledujúcich verziách:

- Oracle Standard Edition 11g Release 2, verzia 11.2.0.1
- Apache, verzia 2.2.19
- PHP verzia 5.4.12, 32 bit

Po nainštalovaní daných verzií je potrebné tieto prostredia medzi sebou prepojiť. Preto je inštalácia doplnená o úpravu niektorých konfiguračných súborov, ako napríklad php.ini.

Po úspešnom nainštalovaní a spustení všetkých prostredí, je možné overiť funkčnosť zadaním skriptu

```
14 <?php
15 echo phpinfo();
16 ?>
```

Následne bude po zavolaní tejto stránky v tvare localhost/test.php zobrazený výpis o funkčnosti všetkých súčastí, čím overíme komunikáciu medzi nimi.

localhost/test.php

PHP Version 5.4.12 

System	Windows NT METKO-PC 6.1 build 7600 (Windows 7 Business Edition) i586
Build Date	Feb 19 2013 21:22:17
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	<pre> cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo" </pre>
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\dev\php\php.ini
Scan this dir	(none)

Obrázok č. 5 – Výpis stránky test.php

4. NÁVRH APLIKÁCIE

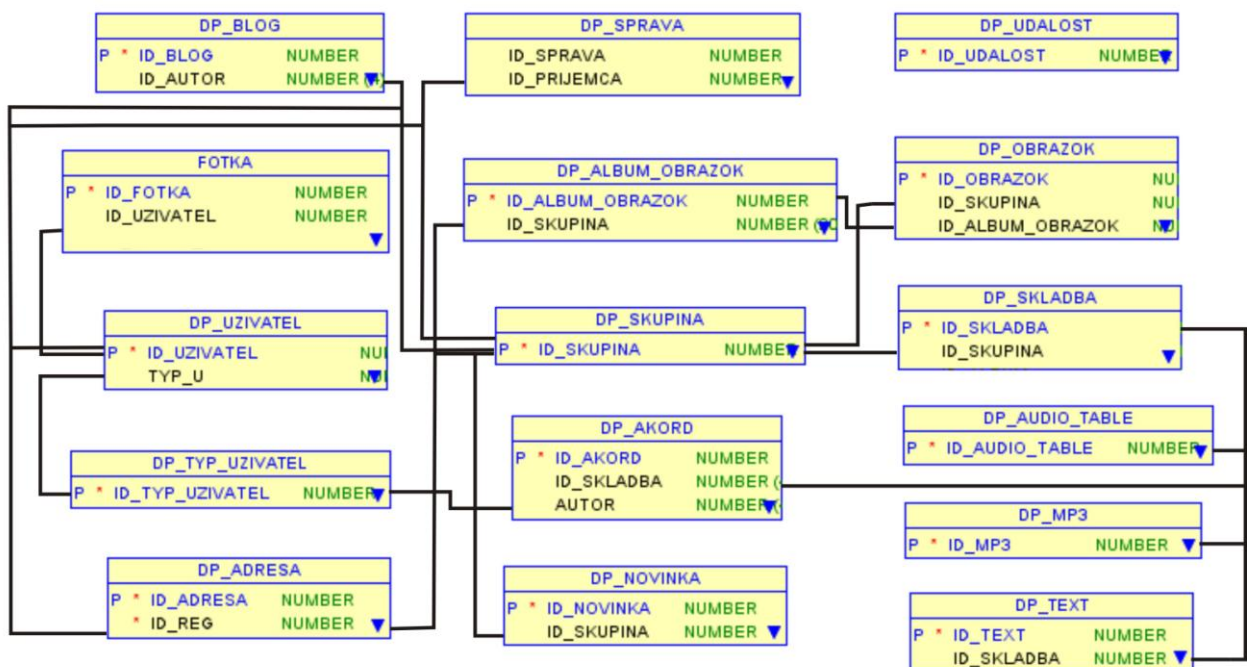
Návrh aplikácie prebiehal v niekoľkých krokoch. V prvom kroku som si zoskupil všetky funkcie, ktoré by mala aplikácia mať. Na základe týchto funkcií som navrhol databázu, do ktorej som začlenil všetky potrebné tabuľky potrebné na obsiahnutie všetkých funkcií. Posledným krokom v počiatočnej fáze návrhu bol vzhľad a rozloženie jednotlivých sekcií aplikácie.

4.1 Požiadavky na funkčnosť

Pri návrhu funkčnosti aplikácie som vychádzal z funkcií, ktoré boli použité v už existujúcich aplikáciách, no považoval som ich za prínosné a výhodné. Inšpiráciou mi boli dobre fungujúce existujúce aplikácie. Rozhodol som sa použiť vybrané funkcie z rôznych aplikácií, zjednotiť a doplniť o množstvo ďalších, ktoré som navrhol sám alebo získal z výsledkov dotazníka.

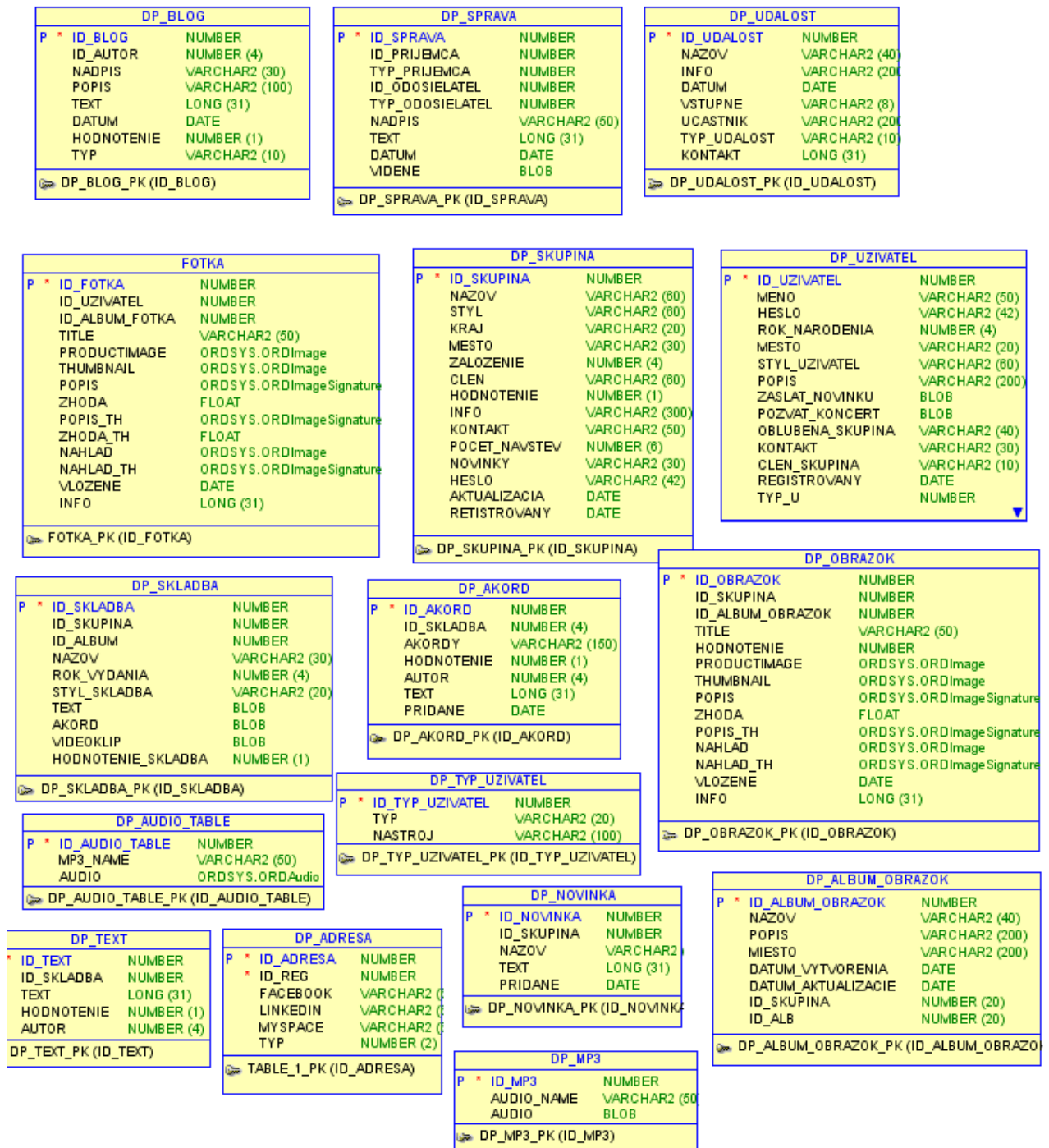
4.2 Návrh databázy

Návrh databázy je nesmierne dôležitý. Je potrebné si dôkladne premyslieť návrh tabuliek a ich naplnenie.



Obrázok č. 6 – Schéma prepojenia tabuliek

Aby databáza spĺňala všetky predpoklady pre správne fungovanie, je potrebné dodržiavať určité pravidlá a taktiež normálové formy návrhu databáze. Štruktúra by mala byť prehľadná s jednoduchým a jednoznačným vyhľadávaním. Každá tabuľka by mala mať tzv. identifikátor, ktorý reprezentuje celý záznam tabuľky.



Obrázok č. 7 – Štruktúra tabuliek

Pri návrhu štruktúry som vychádzal zo štyroch základných tabuliek a to *DP_SKUPINA*, *DP_UZIVATEL*, *DP_SKLADBA* a *DP_OBRAZOK*. Zvyšné tabuľky sú naviazané na tieto základné, ktoré tvoria hlavnú kostru celej štruktúry.

V tabuľke *DP_SKUPINA* sa nachádzajú informácie o každej registrovanej skupine. Každá skupina má svoju diskografiu, skladby, ktoré sú uložené v tabuľke *DP_SKLADBA*. Na túto tabuľku sú prepojené tie, ktoré obsahujú rozširujúce informácie o danej skladbe. Je to napríklad tabuľka *DP_MP3* a *DP_AUDIO_TABLE*, ktorá obsahuje uložené skladby v dátovom type *BLOB* a *ORDAudio*. Súvisiacimi tabuľkami sú aj tabuľky *DP_TEXT* a *DP_AKORD*, v ktorých môže prihlásený užívateľ dopĺňať a rozširovať informácie o danej skladbe.

Informácie o registrovanom užívateľovi sú uložené v tabuľke *DP_UZIVATEL*, doplnenou o tabuľku *DP_TYP_UZIVATEL*, ktorá obsahuje konkrétny typ užívateľa.

Obrázky a fotky kapiel sú uložené v tabuľke *DP_OBRAZOK* v dátovom type *ORDImage*. Tieto obrázky sú rozšírené o informácie o obrázku, ktoré sa nachádzajú v dátovom type *ORDImageSignature*. Každý záznam o obrázku obsahuje aj informáciu o albume, do ktorého daný obrázok patrí. Informácie o albumoch sú v tabuľke *DP_ALBUM_OBRAZOK*.

Keďže fotka, ktorá patrí registrovanému užívateľovi obsahuje odlišné informácie od obrázku, ktorý patrí registrovanej skupine, nachádza sa v zvlášť tabuľke s názvom *DP_FOTKA*. Rozdelenie do dvoch tabuliek taktiež zabraňuje kolíznosti identifikačného čísla užívateľa, ktorému fotka respektíve obrázok patrí.

4.3 Vzhľad stránky

Veľmi dôležitou súčasťou dobrej aplikácie je samozrejme aj jej vzhľad. Už na prvý dojem užívateľ vie, či pri zobrazení webovej stránky mal pozitívny dojem a teda či na nej ostane, alebo naopak.

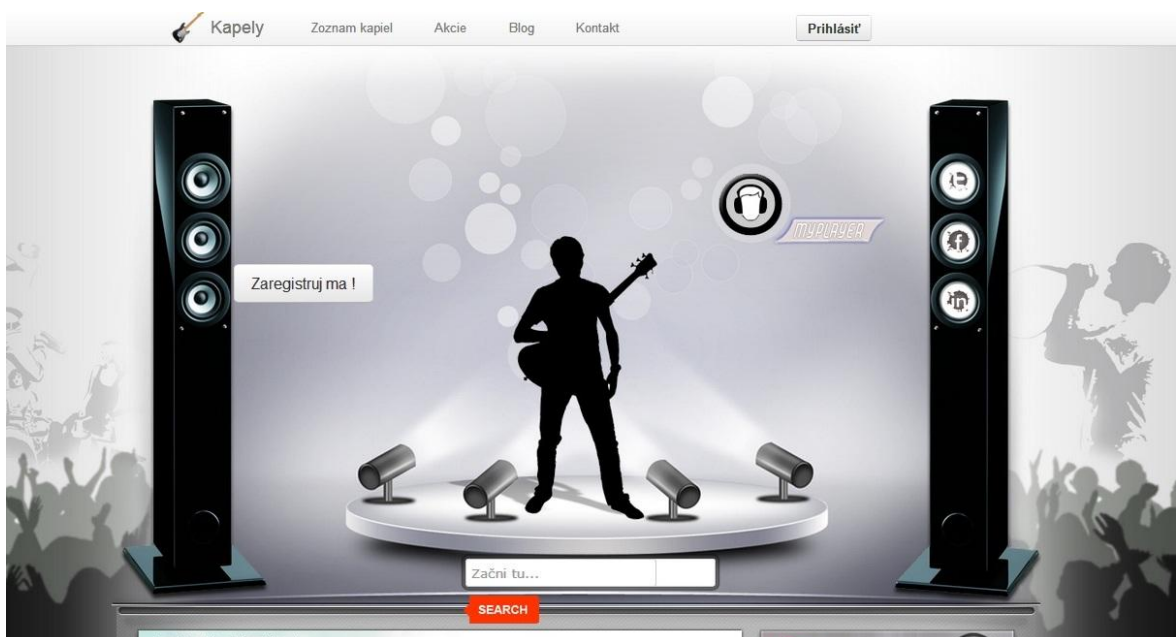
Dizajn tejto aplikácie má viacero základných zobrazení, ktoré sú tématicky a farebne navazujúce a prepojené. Rovnako aj pozadie aplikácie je zrealizované tak, aby sa prispôbovalo veľkosti stránky a tématicky dopĺňalo celkový dizajn.



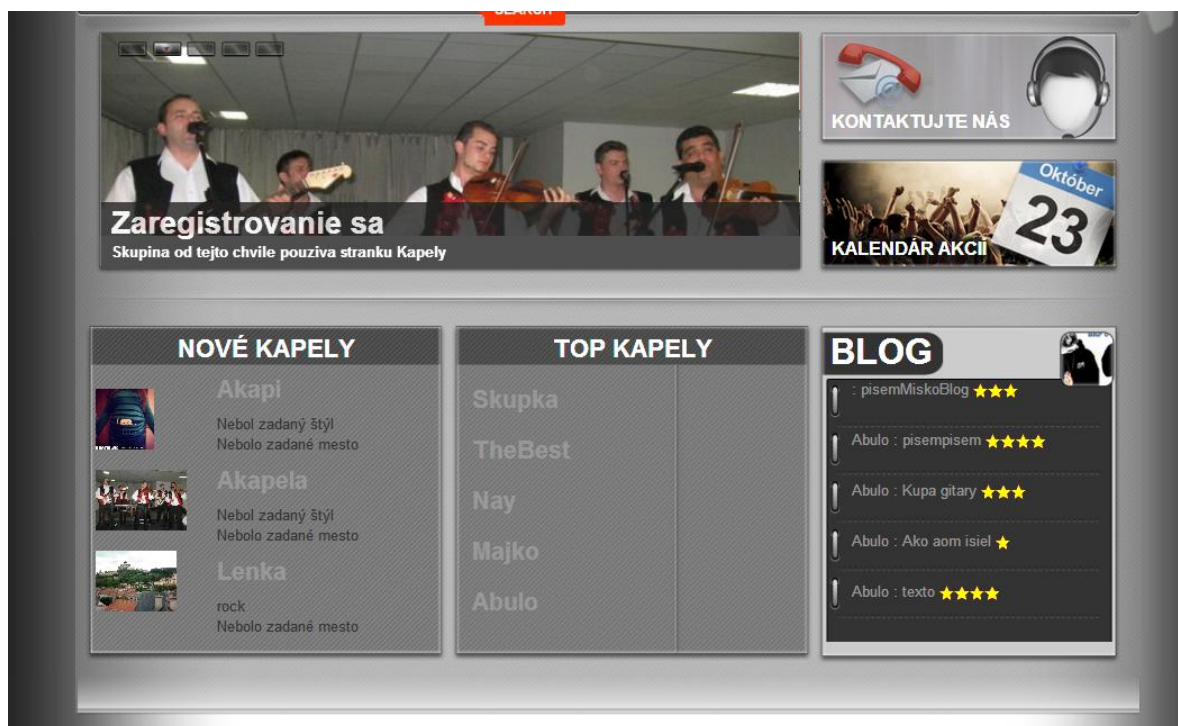
Obrázok č. 8 – Vzhľad aplikácie

Pri návrhu rozmiestnenia jednotlivých okien sa bral ohľad na prítelivosť k užívateľovi. Podstatou bolo, aby rozmiestnenie bolo logické, intuitívne ovládateľné a aby sa k užívateľovi dostali všetky potrebné informácie.

Základné zobrazenie by sa dalo vizuálne rozdeliť na dve časti a to vrchnú a spodnú časť.



Obrázok č. 9 – Vrchná časť hlavnej stránky



Obrázok č. 10 – Spodná časť hlavnej stránky

Vrchná časť obsahuje ovládacie menu, odkazy na sociálne siete a možnosti pre prihlásenie sa a registráciu. Taktiež sa tam nachádza vstupné pole, ktoré je určené pre rýchle vyhľadávanie. Vyhľadávanie prebieha v troch častiach a to v názve kapiel, v názve užívateľov a v názve skladieb.

Spodná časť je vizuálne rozdelená do niekoľko okien, v ktorých sa zobrazujú novinky skupín spoločne s ich fotografiami, novo registrované kapely, rebríček najlepšie hodnotených kapiel a najnovšie pridané blogy. Všetky tieto zoznamy sú interaktívne a po kliknutí na konkrétnu položku sa zobrazí.

Do vizuálne rozdelenej spodnej časti boli umiestnené aj kontaktné údaje či kalendár akcií, medzi ktoré patria koncerty, festivaly a podobne.

5. VÝVOJ APLIKÁCIE

Celá aplikácia komunikuje s databázou Oracle. Takmer každý zobrazený text, ktorý môže užívateľ v aplikácii vidieť, je výsledkom niektorého z príkazov na výpis z databáze. Všetky informácie sú uložené v databáze a na základe žiadostí užívateľa sú postupne zobrazované na obrazovku.

5.1 Aplikačná časť

Aby bola aplikácia plne ovládateľná, aby ňou mohol užívateľ prechádzať a aby mu boli zobrazované tie informácie, ktoré požaduje, je dôležité vytvoriť funkčný zdrojový kód.

5.1.1 Prihlásenie

Prihlásenie prináša týmto užívateľom niekoľko výhod. Bežný, neprihlásený užívateľ môže túto aplikáciu používať len pasívne. Nemôže ju nijakým spôsobom upravovať či rozširovať. Prihlásenie je možné ako užívateľ alebo ako skupina. Prihlásený užívateľ má narozdiel od toho neprihláseného nasledujúce výhody:

- možnosť akordovať skladby
- možnosť hodnotenia blogov, kapiel či skladieb
- písanie blogu
- a komunikácia s ostatnými užívateľmi prostredníctvom správ

Prihlásenie je zabezpečené pomocou *SESSION*, v ktorej sú uchované informácie o tom, či užívateľ je prihlásený a následne niektoré základné, identifikačné údaje konkrétneho užívateľa.

Pri prihlásení je užívateľ povinný zadať heslo, ktoré je uložené v zakódovanej podobe pomocou MD5. Pri každom prihlásení, sa zadané heslo prevedie rovnakou šifrovacou metódou a zadané heslá sa porovnávajú. Po zadaní prihlasovacích údajov sa kontroluje, či daný užívateľ vôbec existuje, či prihlasovný má účet vedený na skupinu alebo konkrétnu osobu a či sa zhodujú zadané heslá. V prípade, že všetky podmienky pre prihlásenie boli splnené, je užívateľ prihlásený a môže využívať všetky výhody tohto stavu.

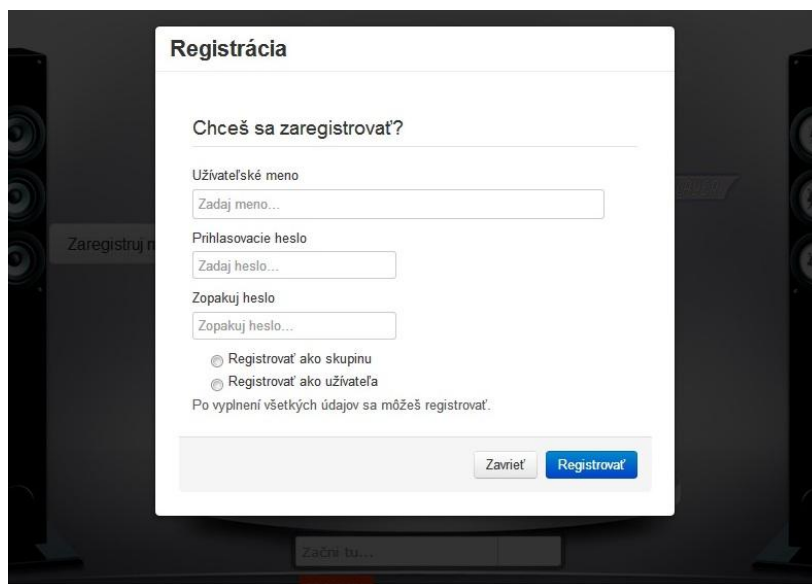
5.1.2 Registrácia

Pred samotným prihlásením je nutná registrácia. Pri registrácii má užívateľ opäť možnosť, vybrať si typ registrácie. Na výber sú dve možnosti a to registrovať sa ako skupina alebo ako užívateľ. Aby bolo garantované, že užívateľ vyberie iba jeden typ, je na túto možnosť použitá funkcia s typom *radio*. Voľba *unchecked* zabezpečuje neoznačenie defaultnej hodnoty, čo slúži ako ochrana proti nepozornosti pri registrácii s predvyplnenou hodnotou.

```
17 <input type="radio" name="reg_type" value = "skupina" unchecked>
```

Pri registrácii sa zadáva užívateľské meno, duplicitne heslo a voľba, o aký typ registrácie ide.

Počas registrácie sa kontroluje, či užívateľ so zadaným menom bude unikátny a teda či sa rovnaké meno už v databáze nenachádza. Taktiež sa kontroluje dĺžka zadaného mena, ktorá musí byť dlhšia ako 3 znaky. Následne sa kontroluje zhoda oboch zadaných hesiel. V prípade, že niektorá z podmienok nie je splnená, bude o tom užívateľ informovaný výpisom na obrazovku spoločne s popisom toho, prečo registrácia nebola umožnená. Po úspešnom zadaní všetkých údajov sa všetky tieto hodnoty vložia do databázy, konkrétne do tabuľky určenej pre registrovaných užívateľov.



The image shows a registration form titled "Registrácia". At the top, it asks "Chceš sa zaregistrovať?". Below this are three input fields: "Užívateľské meno" (User name) with a placeholder "Zadaj meno...", "Prihlasovacie heslo" (Password) with a placeholder "Zadaj heslo...", and "Zopakuj heslo" (Repeat password) with a placeholder "Zopakuj heslo...". There are two radio buttons: "Registrovať ako skupinu" (selected) and "Registrovať ako užívateľa". Below the radio buttons, it says "Po vyplnení všetkých údajov sa môžeš registrovať.". At the bottom right, there are two buttons: "Zavrieť" (disabled) and "Registrovať" (active).

Obrázok č. 11 – Registračný formulár

5.1.3 Blog

Jednou z výhod, ktorú má registrovaný a teda aj prihlásený užívateľ je písanie blogu. Prihlásený užívateľ, či už typu užívateľ alebo skupina má možnosť písať si svoj vlastný blog a zdieľať ho s ostatnými. Taktiež má možnosť prečítať si blog od niekoho iného, ktorý môže zároveň aj ohodnotiť na škále od 1 do 5. Odkazy na najnovšie blogy sú zobrazované na hlavnej stránke, aby užívateľ získal prehľad a bol informovaný, keď sa vytvorí nový blog.

Blog sa skladá z troch častí a to:

- zobrazenie skupiny blogov,
- čítanie konkrétneho blogu
- a písanie blogu.

Pri zobrazení skupiny blogov sa rozlišuje, či užívateľ žiada zobrazit' blogy konkrétnej skupiny alebo toto obmedzenie zanedbal a žiada o zobrazenie všetkých dostupných blogov.

DOSTUPNÉ BLOGY K PREČÍTANIU			
AUTOR	NAZOV	DATUM	
Misko	Korupcia a ci nie?	pridané dňa: 21.05.2013	Prečítať
Abulo	Zmenili sme priestory	pridané dňa: 06.05.2013	Prečítať
Abulo	Kupa gitary	pridané dňa: 05.05.2013	Prečítať
Abulo	Ako aom isiel	pridané dňa: 04.05.2013	Prečítať
Abulo	Ako to povedat...	pridané dňa: 04.05.2013	Prečítať
Kusocek	Koncert v Nitre	pridané dňa: 04.05.2013	Prečítať
Kusocek	Boli sme tam	pridané dňa: 03.05.2013	Prečítať
Kusocek	Keby bolo takto	pridané dňa: 02.05.2013	Prečítať
Abraka	Novy clen	pridané dňa: 01.05.2013	Prečítať

Obrázok č. 12 – Zobrazenie zoznamu dostupných blogov

V zozname blogov sa nachádza meno autora, názov blogu a dátum, kedy bol tento blog vytvorený. Ku každému blogu je možnosť *Prečítať*, čím sa dosiahne zobrazenie konkrétneho blogu. Aby sa pri kliknutí na ikonu *Prečítať* zobrazil konkrétny blog, som dosiahol nasledovnou časťou kódu, ktorá sa nachádza vo vnútri cyklu pre výpis blogov z databáze.

```
18 <a href="citac_blog.php?id_blog_sk='.$ID_BLOG.'">Prečítať</a>
```

Ako je možné vidieť, odkaz je smerovaný na ďalšiu stránku s názvom *citat_blog.php*, ktorá je určená na zobrazenie konkrétneho blogu k prečítaniu. Tento odkaz je doplnený o názov parametra (*id_blog_sk*), v ktorom sa prenáša hodnota zvoleného blogu v podobe jeho ID čísla, ktoré je získané z databázy.

5.1.4 Hodnotenie skupiny a blogu

Ako bolo spomínané, každý prihlásený užívateľ má možnosť ohodnotiť blog na stupnici od 1 do 5. Toto platí aj pre hodnotenie skupín. Na základe tohto hodnotenia je na úvodnej stránke zobrazený rebríček pre top kapely. Taktiež toto hodnotenie slúži aj ako odozva či odporúčanie pre ostatných užívateľov. Vďaka tomu sa môže užívateľ rozhodnúť prečítať si daný blog alebo vypočítať si niektorú z dobre hodnotených kapiel.

V prípade, že užívateľ zvolil kliknutím jednu z piatich hviezdíčiek symbolizujúcich stupeň ohodnotenia, je táto hodnota číselne znázornená pri ikone potvrdzujúcej hodnotenie.

Po odoslaní sa z databázy načíta aktuálna hodnota a počet hlasovaní. Vďaka týmto hodnotám sa vypočíta aktuálna hodnota ktorá sa zapíše do databázy na rovnakú pozíciu vďaka príkazu UPDATE. Tento príkaz vyzerá nasledovne

```

19     $hodnotenie_nove = "".$hodnoty." / ".$pocet_hlasovani."";
20     $prikaz = oci_parse($databaza,"UPDATE DP_SKUPINA SET HODNOTENIE =
    '$hodnotenie_nove' WHERE ID_SKUPINA = '$id_skupina'");
21     oci_execute($prikaz, OCI_DEFAULT);
22     oci_commit($databaza);

```



Obrázok č. 13 – Hodnotenie skupiny

5.1.5 Vyhľadávanie

Základné vyhľadávanie textu je umiestnené na úvodnej stránke. Zadaný výraz sa vyhľadáva medzi názvami skupín, menami účastníkov a názvami skladieb. Toto vyhľadávanie nemusí byť full-textové, to znamená, že nemusí byť zadaný presný názov, ale postačí časť slova. Toto vyhľadávanie má podmienku, že zadaný výraz musí mať minimálne 3 znaky. Po zadaní časti slova budú prehľadávané jednotlivé tabuľky a v tomto zotriedenom poradí budú aj zobrazované výsledky. V prípade, že sa hľadaný záznam nenašiel a ani sa postupnosť znakov nenachádza v niektorej z prehľadávaných tabuliek, bude o tom užívateľ informovaný. Časť kódu, ktorá popisuje vyhľadávanie aj časti textu je nasledovná:

```
23 $zadany_vyraz = $_POST["zadany_vyraz "];
24 $prikaz = oci_parse($databaza, "SELECT ID_SKUPINA, NAZOV FROM
    DP_SKUPINA WHERE NAZOV LIKE '%$ zadany_vyraz %' ");
```

Nájdené a vypísané hodnoty sú interaktívne, čo znamená, že slúžia ako odkaz na danú položku.

5.1.6 Oakordovanie

Oakordovanie je jedna z funkcií, ktorá mala veľký úspech v prieskume, ktorý som vykonal pred samotným návrhom aplikácie. Oakordovanie dáva priestor prihláseným užívateľom, aby pridali akordy k už existujúcej skladbe niektorej zo zaregistrovaných skupín. Táto možnosť je obmedzená tým, že akordy na danú skladbu nemôžu už existovať. To znamená, že v prípade, že daná skladba už je oakordovaná a následne tieto akordy boli skupinou a teda majiteľom skladby odsúhlasené, užívateľ už nemá možnosť túto skladbu znova oakordovať.

Postup pri akordovaní je nasledovný:

- Užívateľ si zvolí skladbu danej kapely, ktorú chce oakordovať
- V zozname skladieb je zobrazené, či táto skladba už bola oakordovaná alebo nie. V prípade že bola, užívateľ má možnosť si tieto akordy pozrieť. V opačnom prípade má možnosť túto skladbu oakordovať sám.
- Ak sa rozhodne skladbu oakordovať, vyhľadá sa v databáze text k danej skladbe. Ak sa nenachádza, musí ho užívateľ napísať sám.

- Po zadání textu společně s místem na akordy, které vyznačí symbolem X, je poskytnutý náhled textu spolu so vstupnými poliami určenými pre zadávanie akordov.
- V prípade, že užívateľ zadal všetky akordy, bude táto jeho verzia uložená do databázy a zároveň sa skupine, ktorá skladbu vlastní zašle správa o jej oakordovaní.
- Skupina má možnosť si oakordovanú verziu pozrieť a ak s návrhom súhlasí, potvrdí ju. Potvrdením sa skladba dostáva do stavu, kedy ju už ďalší užívatelia nemôžu oakordovať. O potvrdení je oboznámení aj autor oakordovania automatickou správou.
- Ak však s oakordovanou verziou skupina nesúhlasí, môže tento návrh zamietnuť a skladbu bude možné aj naďalej oakordovať. O zamietnutí bude taktiež autor informovaný automatickou správou.
- V čase, kedy skupina oakordovanie nepotvrdila ale ani nezamietla, je skladbu naďalej možné oakordovať, pričom sa skupine zobrazia všetky vytvorené verzie.

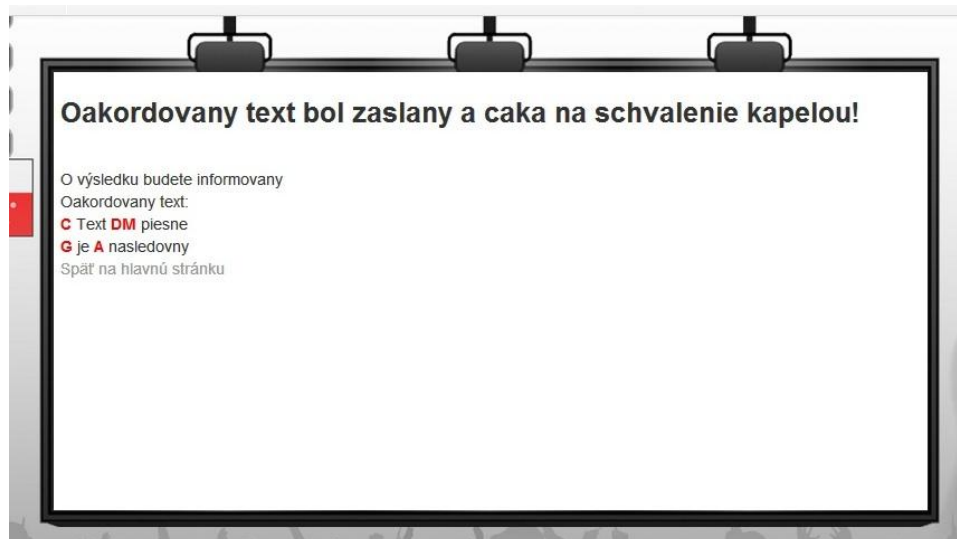
Zadajte text piesne v nasledujúcej forme. Priestor, kde sa bude nachádzať akord oznacte X, ukončenie riadka oznacte R.
Príklad: X Text pie X sne. R X nový X riadok

X text X piesne R X je X nasledovny

text piesne
 je nasledovny

Kliknite pre uloženie zadanych akordov

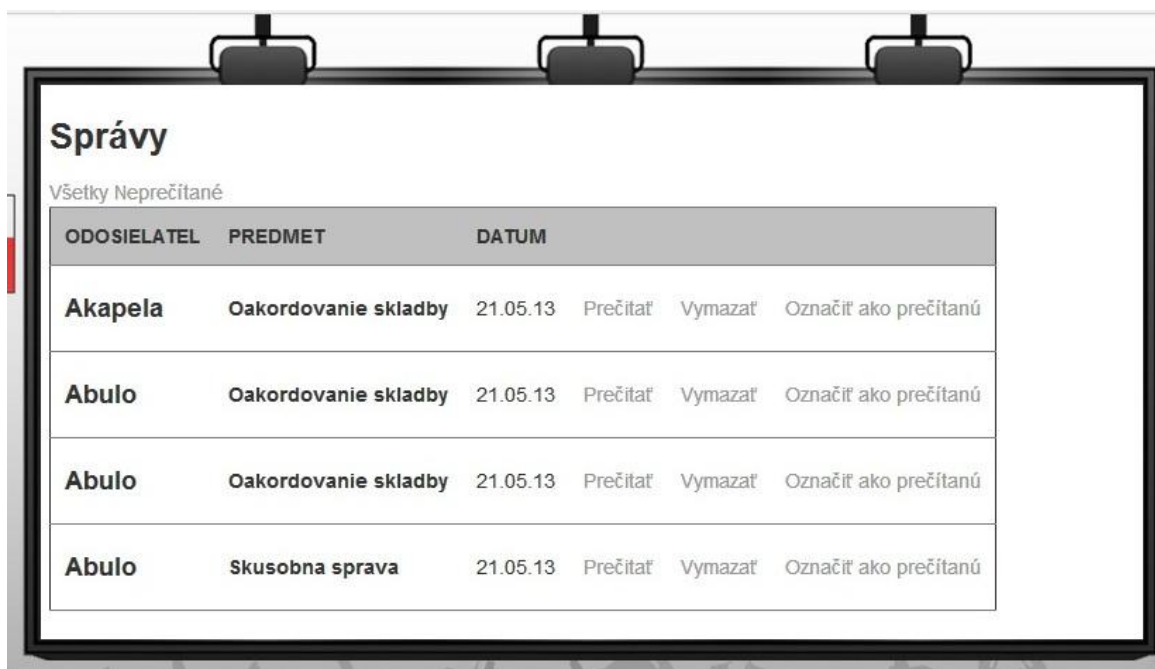
Obrázok č. 14 – Zadávanie textu pre oakordovanie



Obrázok č. 15 – Oakordovaný text

5.1.7 Správy

Medzi prihlásenými užívateľmi je možnosť zasielania si správ. Tieto správy prichádzajú skupinám napríklad pri oakordovaní skladby. Skupina obdrží správu, že užívateľ oakordoval ich skladbu a na základe tejto správy môžu túto verziu potvrdiť, ale zamietnuť. V oboch prípadoch následne obdrží správu informujúcu o stave aj užívateľ.



Obrázok č. 16 – Schránka správ

Správy je možné prečítať, vymazať alebo označiť ako prečítané. Všetky správy sa nebudú zobrazovať v prípade, že zvolíme možnosť zobraziť len neprečítané správy.

```
25 if(isset($_GET["neprecitane"])){
26     $prikaz = oci_parse($database,"SELECT FROM DP_SPRAVA WHERE
    VIDENE = '0' ORDER BY ID_SPRAVA DESC");
27 }
```

Vyššie uvedený kód hovorí, že ak užívateľ vyberie možnosť zobrazenia len neprečítaných správ, selektovacím príkazom sa budú vyhľadávať iba správy, ktoré majú atribut *VIDENE* s hodnotou rovnou nule, čo znamená, že tieto správy boli už zobrazené.

5.2 Databáza

Aby mohli vytvorené PHP funkcie získavať rôzne informácie potrebné k funkčnosti aplikácie, musí byť vytvorená databáza, ktorá bude slúžiť ako úložisko pre tieto hodnoty.

5.2.1 Vytvorenie tabuliek

Tabuľky v *SQL* prostredí tvoria príkazom *CREATE*.

Príkladom vytvorenia takejto tabuľky môže byť tabuľka určená pre ukladanie správ.

```
28 CREATE TABLE DP_SPRAVA
29     (   ID_SPRAVA NUMBER,
30     ID_PRIJEMCA NUMBER,
31     TYP_PRIJEMCA NUMBER,
32     ID_ODOSIELATEL NUMBER,
33     TYP_ODOSIELATEL NUMBER,
34     NADPIS VARCHAR2(50 BYTE),
35     TEXT LONG,
36     DATUM DATE,
37     VIDENE NUMBER,
38     constraint DP_SPRAVA_PK PRIMARY KEY(ID_SPRAVA));
39 commit;
```

5.2.2 Trigger a sequence

Po vytvorení tabuliek je vhodné nastaviť niektorý atribút na auto increment, čo znamená, že každý nový záznam bude mať navýšenú hodnotu od predchádzajúcej na danom atribúte. Môže ísť napríklad o identifikačné číslo, ktoré sa s každým pridaným záznamom navýši. Narozdiel od MySQL, systém ORACLE nemá priamu funkciu auto increment. Preto sa táto funkcia nahrádza sekvenciou a triggerom.

Každá sekvencia aj trigger majú názov. V sekvencii sa udáva, čo sa má vykonať. V našom prípade budeme v sekvencii nastavovať počiatočnú hodnotu počítania a inkrementáciu o hodnotu 1.

Trigger je automaticky volaný príkaz, ktorý sa vykonáva za určitých podmienok. Nastavenie triggeru ktorý som používal obsahoval volanie sekvencie, ktorá vykonávala inkrementáciu a toto volanie prebieha pri každom vložení nového záznamu do konkrétnej tabuľky.

Ukážka vytvorenia auto inkrementácie na tabuľku *DP_SPRAVA*.

```
40 CREATE SEQUENCE dp_sprava_sequence START WITH 1 INCREMENT BY 1;

41 CREATE OR REPLACE TRIGGER dp_sprava_trigger
42 BEFORE INSERT
43 ON DP_SPRAVA
44 REFERENCING NEW AS NEW
45 FOR EACH ROW
46 BEGIN
47 SELECT dp_sprava_sequence.nextval INTO :NEW.ID_SPRAVA FROM dual;
48 END;
```

5.2.3 Stránkovanie a obmedzenie príkazu Select

Pri zobrazení fotoalbumov, zoznamu skupín alebo stránok, kde je veľa záznamov, je potrebné stránku rozdeliť na podstránky, ktoré sú číselne rozdelené. Vďaka tomuto môžeme zobrazovať na jednej stránke (napríklad zoznam.php) aj veľký počet záznamov.



Obrázok č. 17 - Stránkovanie

Toto rozdelenie som dosiahol ikonami, ktoré určujú, ktorá podstránka má byť zobrazená a na základe tejto informácie sa rozhoduje, ktorá časť zdrojového kódu sa vykoná a teda ktorá časť bude zobrazená.

Na začiatku je potrebné zistiť celkový počet záznamov a podeliť ho počtom záznamov na jednej stránke. Stránkovanie som zabezpečil tak, aby v prípade, že sa zoznam dostane na koniec, t.j. na poslednú možnú podstránku, ikony vykonávajúce posun stránky budú neaktívne a po kliknutí na ne nebude vykonaná žiadna akcia.

Keďže na každej podstránke majú byť zobrazené iba niektoré záznamy, je potrebné obmedziť aj výber pomocou príkazu *SELECT*.

Rovnako ako Oracle nemá priamu funkciu na auto inkrementáciu, tak isto nemá priamu funkciu ani na obmedzenie počtu záznamov pri výbere príkazom *SELECT*.

Toto obmedzenie sa mi podarilo dosiahnuť nasledujúcim príkazom, ktorý v prvom kroku počíta záznamy a v druhom kroku vyberá iba taký počet záznamov, ktorý spĺňa obmedzenie dané v podmienke.

```
49 $prikaz = oci_parse($databaza,"SELECT * FROM (SELECT q.*, rownum
    rn FROM (SELECT * FROM DP_SKUPINA ORDER BY NAZOV) q) WHERE rn
    BETWEEN $idmin AND $idmax");
```

5.2.4 Editovanie profilu

Po úspešnej registrácii a následnom prihlásení ma každý užívateľ možnosť vyplnenia svojho profilu. V prípade, že už svoj profil vyplnil, môže tým istým spôsobom upravovať už existujúce informácie.

Editácia prebieha načítaním všetkých uložených informácií, ktoré sú zobrazené ako predvyplnená hodnota vo vstupnom poli formulára. Ak sa užívateľ rozhodne niektorú z hodnôt zmeniť či vymazať, stačí túto hodnotu upraviť vo vstupnom poli a uložiť. Po uložení sa automaticky všetky pôvodné ale aj zmenené hodnoty uložia do tabuľky pomocou príkazu *UPDATE*. Časť zdrojového kódu na zobrazenie a zmenu štýlu skupiny.

```
50 $prikaz = oci_parse($databaza,"SELECT STYL FROM DP_SKUPINA WHERE
    ID_SKUPINA = '$id_skupina' ");
51 $styl = oci_result($prikaz, "STYL");
52 echo '<label for="styl">Styl</label><input name="styl" type = "text"
    value= '.$styl.'>';
```

5.3 Práca s obrázkom

Dôležitou časťou aplikácie, ktorá je dôkazom širokého využitia systému Oracle je práca s obrázkom. Užívateľ si môže nahrať svoju profilovú fotku, vytvárať fotoalbumy a podobne. Profilovú fotku je možné po samotnom nahratí aj upravovať. Všetky úpravy vykonáva systém Oracle.

5.3.1 Vytvorenie a zmazanie priečinka

Každý užívateľ má vytvorené vlastný adresár, do ktorého sa ukladajú jeho fotky. Tento priečinok je pomenovaný identifikačným číslom užívateľa. V prípade, že užívateľ má aj fotoalbumy, vytvoria sa ďalšie zložky, ktoré sú pomenované identifikačným číslom albumu. Na základe tohto je jasné, že štruktúra adresára je množstvo priečinkov pomenovaných číslami. Je tým zaistená istá forma bezpečnosti, že aj keby sa niekto k obrázkom dostal, nevie, komu patria ktoré. Krátka časť takéhoto kódu vyzerá nasledovne:

```
53 $upload_dir = "../obrazok_dir/$id_skupina/$id_album/ ";  
54 $result = mkdir($upload_dir, 0755);
```

V prípade, že užívateľ sa rozhodne fotoalbum zmazať, budú všetky záznamy týkajúce sa albumu a obrázkov z databázy vymazané, rovnako ako sa vyprázdni a následne vymaže aj adresár.

5.3.2 Procedúry

Po tom, ako je obrázok vložený do príslušného adresára, je potrebné vložiť tento obrázok aj do databázy. K tomuto kroku som použil procedúru. Procedúra je postupnosť SQL príkazov, ktorá je vytvorená v Oracli. Túto procedúru v danom momente volám z PHP skriptu. V PHP skripte odovzdvám tejto procedúre určité parametre, ktoré budú v SQL potrebné pre jej vykonanie. Každá procedúra obsahuje názov, hodnoty s ktorými bude pracovať a postupnosť príkazov.

Procedúra, ktorú som vytvoril pre vkladanie obrázku do databázy.

```
55 create or replace procedure insertovanie (id_o number, id_s number,  
    id_a number, nazov long, popisok long) as  
56 begin  
57   ...postupnosť príkazov, ktoré sa majú vykonať...  
58 end;  
59 end;
```

5.3.3 Úprava obrázkov

Jednou z veľkých výhod systému Oracle, ktorá ma viedla k jeho výberu, je schopnosť pracovať s multimedialnými dátami. Okrem toho, že vie tieto dáta ukladať, umožňuje aj ich jednoduchú úpravu.

Keďže veľkosť zobrazovanej profilovej fotky má špecifický rozmer a pomer strán, môže si užívateľ pri nahrávaní túto fotku upraviť tak, ako mu najviac vyhovuje.

Po zvolení novej profilovej fotky, sa užívateľovi táto fotka zobrazí ako miniatúra v ľavom hornom rohu. Na pravej strane sú zobrazené informácie o tejto pôvodnej fotke, ktoré sú získané prostredníctvom *ORDImageSignature*. Pod touto pôvodnou fotkou sa nachádza časť určená na úpravu fotky. Užívateľ si môže vybrať jednu z preddefinovaných funkcií, ako je rotácia, zrkadlenie či prevrátenie fotky. Okrem preddefinovaných má možnosť zadať vlastné hodnoty do vstupného formulára a tieto zmeny budú vykonané so zadanými hodnotami. Tieto voľby je možné aj vzájomne kombinovať.

Jednou z možností je aj automatický orez, kde je preddefinovaný pomer strán na základe ktorého sa fotka automaticky oreže.

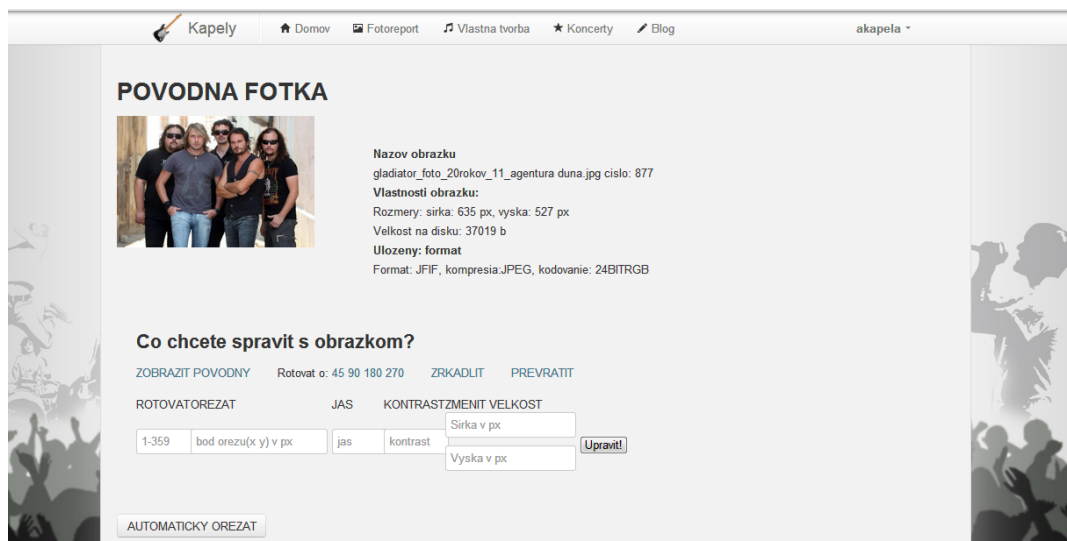
Užívateľ má možnosť vykonať nasledujúce zmeny:

- Preddefinované
 - Rotácia fotky (45°, 90°, 180°, 270°)
 - Horizontálne alebo vertikálne zrkadlenie fotky
 - Zobrazenie pôvodnej fotky
- Voliteľné
 - rotácia fotky (1 ° - 359 °)
 - orezať fotku (zadaním veľkosti orezu v pixeloch)
 - upraviť jas
 - upraviť kontrast
 - zmeniť veľkosť (zadaním veľkosti obrázku v pixeloch)

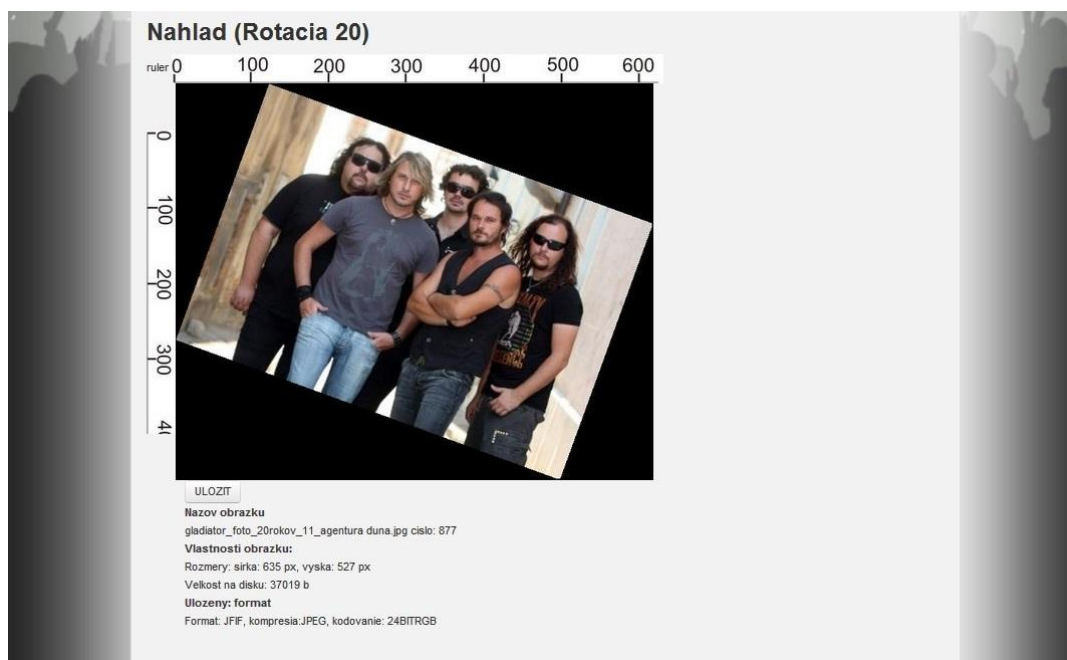
Po vykonaní niektorej zo zmien sa v spodnej časti zobrazí náhľad upravenej fotky spoločne s jej vlastnosťami. Vykonaním každej zmeny sa z PHP skriptu automaticky zavolá

procedúra, ktorá slúži na úpravu fotky a jej následne exportovanie do adresára, ktorý je určený na ukladanie fotiek užívateľa. To znamená, že každá zmena na fotke je okamžite vykonaná aj na fotke uloženej v priečinku a vyexportovaná v upravenej podobe.

Po uložení týchto zmien bude fotka automaticky uložená, vytvorí sa jej náhľad a zmena sa odzrkadlí vo všetkých častiach, kde sa fotka zobrazuje.



Obrázok č. 18 – Úprava profilovej fotky – vrchná časť



Obrázok č. 19 - Úprava profilovej fotky – spodná časť

Úprava fotiek prebieha pomocou špeciálnej funkcie systému Oracle s názvom *PROCESS COPY*.

5.3.4 Process copy

Funkcia *process copy* je jedným zo špeciálnych nástrojov systému Oracle, ktorý slúži napríklad na úpravu obrázkov. Celý proces úpravy fotky prebieha v troch krokoch.

Pre každý krok som vytvoril procedúru, ktorá sa automaticky volá z PHP skriptu. Pre každú procedúru, ktorú som vytvoril, som taktiež nadefinoval, ktoré parametre sa budú z PHP skriptu posielat'.

V prvom kroku sa musí uložit' do databázy pôvodná fotka. Toto uloženie vykonávam pomocou automatického volania procedúry s názvom *insertovanie*. V tejto procedúre sa z PHP skriptu prepošlú potrebné informácie o obrázku, ako je jej identifikačné číslo fotky, identifikačné číslo užívateľa ktorému fotka patrí, identifikačné číslo fotoalbumu a názov fotky. Tieto hodnoty spracované v procedúre, ktorá sa vykonáva v SQL v prostredí Oracle.

V druhom kroku sa vykonáva samotná úprava nahranej fotky. Úprava opäť prebieha prostredníctvom procedúry pomenovanej *upravovanie*. Parametre, ktoré sa tejto procedúre odovzdávajú sú identifikačné číslo fotky, ktorú žiadame upraviť a funkcia, ktorá bude na zmenu fotky aplikovaná.

Posledným krokom je exportovanie upravenej fotky do priečinku, v ktorom sú uložené fotky užívateľa. Procedúra *exportovanie* obsahuje okrem identifikačného čísla fotky aj názov pre novo vytvorenú fotku.

Pri úprave fotiek som zrkadlenie fotky dosiahol vytvorením funkcie *process copy* v nasledujúcom tvare:

```
60 funkcia = 'mirror';  
61 povodny_obrazok.processCopy(funkcia, upraveny_obrazok);
```

Funkcia, ktorá sa má na fotku aplikovať sa posielala v parametroch procedúry. Vďaka tejto vlastnosti môžem funkciu *processCopy* nadefinovať dynamicky a to tak, že zadenovanie konkrétnej funkcie nahradím premennou s názvom *funkcia*, do ktorej sa pred zavolaním *processCopy* priradí konkrétny názov funkcie, ktorý sa získa z parametrov procedúry

.

Ukážka zdrojového kódu, v ktorom sa definuje a volá procedúra pre úpravu obrázku:

```
62 //volanie procedúry a odovzdanie parametrov z PHP  
63 $prikaz = oci_parse($db, "call upravovanie('$id', '$funkciaa')");
```

```
64 //definovanie a vykonanie procedúry v Oracle
65 create or replace procedure upravovanie (id_o number, funkcia long)
66 as
67 begin
68     DECLARE
69         obj_1 ORDSYS.ORDImage;
70         obj_2 ORDSYS.ORDImage;
71     BEGIN
72     SELECT productimage, thumbnail INTO obj_1, obj_2 FROM dp_obrazok
73     WHERE id_obrazok = id_o FOR UPDATE;
74     obj_1.processCopy(funkcia, obj_2);
75     UPDATE dp_obrazok SET thumbnail = obj_2 WHERE id_obrazok=id_o;
76     COMMIT;
77     EXCEPTION
78     WHEN ORDSYS.ORDImageExceptions.NULL_DESTINATION THEN
79     DBMS_OUTPUT.PUT_LINE('The destination is null');
80     WHEN ORDSYS.ORDImageExceptions.DATA_NOT_LOCAL THEN
81     DBMS_OUTPUT.PUT_LINE('Data is not local');
82     WHEN ORDSYS.ORDImageExceptions.NULL_LOCAL_DATA THEN
83     DBMS_OUTPUT.PUT_LINE('dest.source.localData attribute is null');
84     COMMIT;
85     END;
86     END;
```

5.3.4.1 Vytvorenie náhľadu

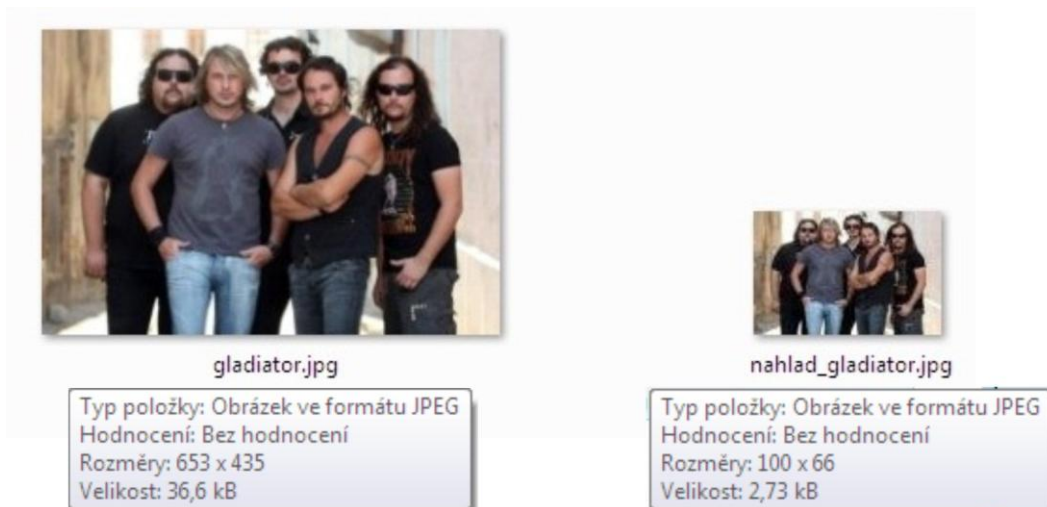
Keďže uložené môžu mať pomerne veľkú veľkosť, môže ich načítanie trvať dlhšiu dobu. Toto je veľmi nevýhodné pri zobrazeniach ako napríklad *zoznam kapiel*, kde stačí oveľa menšia veľkosť zobrazovanej fotografie. Kôli tomuto faktoru je aplikácia nastavená tak, že pri každom uložení novej fotky, sa automaticky vytvorí jej kópia, ktorá má maximálnu veľkosť jednej strany 100 pixelov. Vytvorená kópia sa uloží sa pod rovnakým názvom pred ktorý sa doplní „*nahlad_*“. Výsledný názov kópie fotky, ktorá je určená na náhľad je „*nahlad_povodnynazov*“. Tento obrázok má v porovnaní s pôvodnou fotografiou veľmi malú veľkosť, čo urýchli načítanie fotografie určenej pre náhľad.

Vytvorenie tohto náhľadu je pomocou nasledovnej funkcie

```
86 povodny_obrazok.processCopy(maxScale=100 100', obrazok_nahlad);
```

```
87 // a po ďalších úpravách exportovanie do tvaru
```

```
88 Obrazok_nahlad.export(ctx, 'file', 'cesta_k_priecinku', novy_nazov);
```



Obrázok č. 20 – Automatické vytvorenie náhľadu

Na obrázku č. 20 je zobrazený obsah adresára, ktorý patrí užívateľovi. V adresári sa nachádza pôvodná fotografia s názvom *gladiator.jpg*, ktorá má pôvodnú veľkosť. K tejto fotografii sa automaticky vytvorila kópia s maximálnou veľkosťou jednej strany 100 pixelov, ktorá sa bude načítavať pri zobrazovaní náhľadu. Táto fotografia má automaticky vytvorený názov *uprava_gladiator.jpg* a ako je možné vidieť, veľkosť zmenšenej fotografie je oveľa menšia, čo zabezpečí rýchlejšie načítanie.

5.3.5 Podobnosť obrázkov

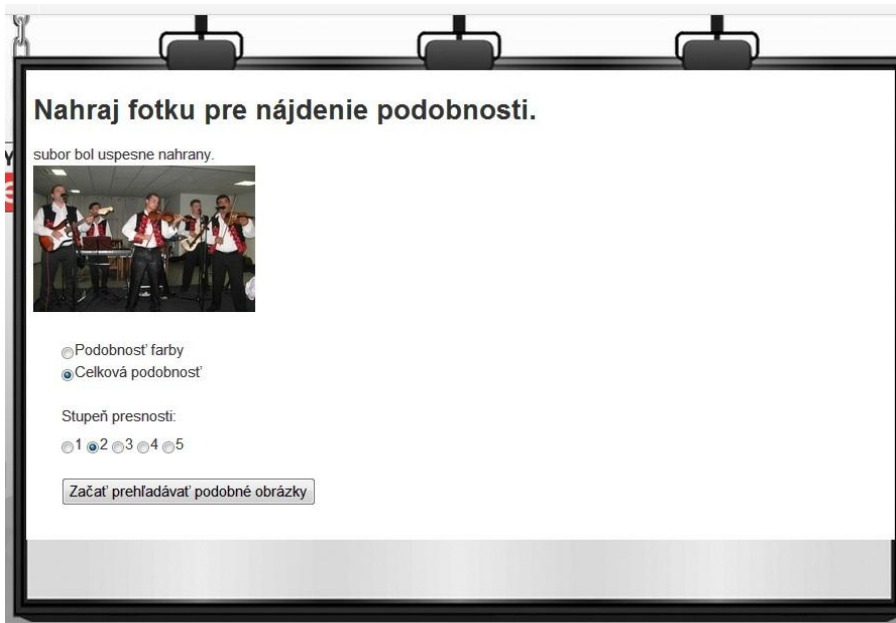
Jedným zo zaujímavých nápadov, ktorý sa objavil v názoroch v dotazníku, bolo vytvoriť vyhľadávanie na základe podobnosti obrázku. Rozhodol som sa túto funkciu v aplikácii ponúknuť.

Účelom funkcie je nahráť hľadaný obrázok, na základe ktorého sa bude vyhľadávať podobnosť tohto obrázku s obrázkami uloženými v databáze. Ak užívateľ vlastní obrázok z koncertu, či z titulnej fotky albumu no nevie názov skupiny, môže skúsiť tento obrázok nahráť a na základe podobnosti obrázkov sa mu zobrazia všetky, ktoré sú zadanému obrázku podobné.

Užívateľ má možnosť zvoliť stupeň presnosti na stupnici od 1 do 5, kde stupeň 1 znamená najvyššiu presnosť a teda najvyššiu zhodu. Taktiež si môže vybrať formu podobnosti, kde si

môže vybrať vyhľadávanie na základe podobnosti farby obrázku alebo celková podobnosť obrázku. Celková podobnosť porovnáva farbu, tvar, umiestnenie tvarov a povrch obrázku.

Po dokončení vyhľadávania sú užívateľovi zobrazené všetky obrázky, ktoré spĺňajú zvolené kritéria.



Obrázok č. 21 – Vyhľadávanie podobnosti obrázkov – nahranie fotky



Obrázok č. 22 – Vyhľadávanie podobnosti obrázkov - výsledok

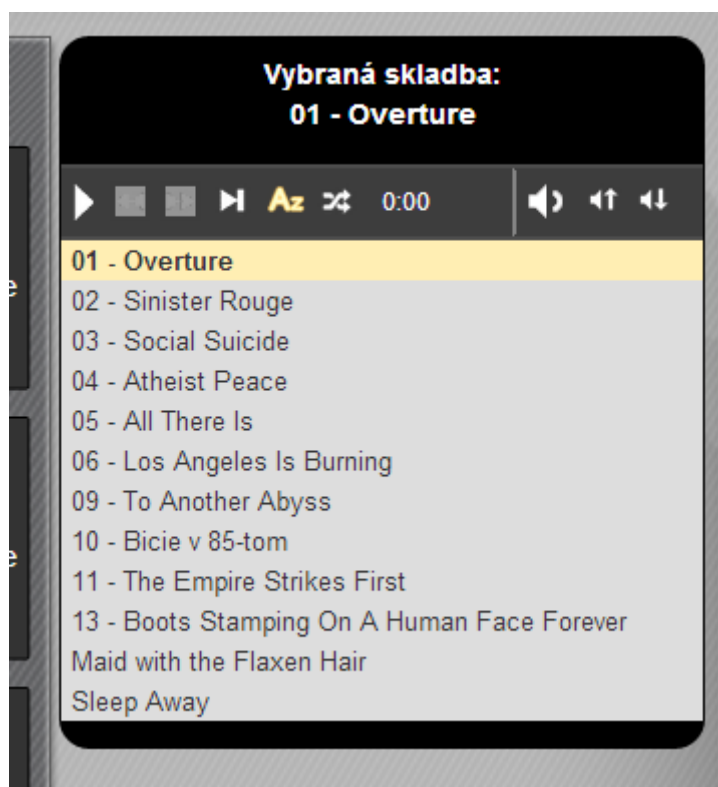
5.4 Práca s hudbou

Okrem práce s obrázkami zvláda systém Oracle aj prácu s inými multimédiami typmi, napríklad práca s hudbou. Podobne ako je obrázok uložený v dátovom type *ORDImage*, hudba sa ukladá do dátového typu *ORDAudio*.

Pre ukladanie hudby je vytvorená v databáze tabuľka *DP_MP3* a *DP_AUDIO_TABLE*.

Pri zobrazení stránky konkrétnej kapely, sa v pravej časti nachádza hudobný prehrávač, ktorý dokáže prehrať skladby uložené kapelou do daného adresára. Tieto skladby je možné prehrať v poradí, v akom boli uložené alebo s využitím funkcie *random*, ktorá prehrá skladby v náhodnom poradí.

Pri nahrávaní skladieb sa kontroluje celková veľkosť a formát nahrávanej skladby. Zabráňuje sa tým nahrávaniu nepodporovaných formátov ako aj nahrávaniu skladieb s väčšou veľkosťou ako sa povoľuje.



Obrázok č. 23 – Prehrávač hudby

6. SÚČASNÁ PODOBA A BUDÚCNOSŤ

6.1 Funkcionalita

Aplikácia ponúka široké spektrum možností, ktoré môže užívateľ využiť. Tieto funkcie sa dajú rozdeliť na tie, ktoré sú umožnené vykonávať bežnému užívateľovi, registrovanému užívateľovi a registrovanej skupine.

➤ **Bežný užívateľ**

- prezeranie profilov kapiel
- zobrazenie fotiek kapiel
- počúvanie hudby

➤ **Registrovaný užívateľ**

- funkcie bežného užívateľa +
- oakordovanie skladieb
- hodnotenie kapiel a blogu
- komentovanie kapiel
- písanie blogu
- schránka správ
- vytvorenie profilu spoločne s profilovou fotkou
- úprava profilovej fotky

➤ **Skupina**

- funkcie registrovaného užívateľa +
- pridávanie skladieb
- vytváranie fotoalbumov
- pridelovanie práv na doplnenie skladieb užívateľom

6.2 Možnosti rozšírenia

Keďže návrh databázy prebiehal s ohľadom na neskoršie rozšírenie či doplnenie funkcií, je možné túto aplikáciu doplniť o ďalšie funkcie, ktoré sa časom môžu ukázať ako žiaduce.

Doplnením tejto aplikácie môžu byť funkcie na:

- predaj hudobných nástrojov
- zverejňovanie inzerátov na člena kapely
- zasielanie informačnej správy v prípade vytvorenia novej udalosti
- nahrávanie videoklipov
- vytváranie a zverejňovanie statusov a podobne.

Tieto funkcie zatiaľ neboli do aplikácie zahrnuté, no v budúcnosti je táto aplikácia plne rozšíriteľná a preto je možné ju doplniť o ďalšie množstvo užitočných funkcií.

Niektoré z týchto funkcií sa môžu ukázať ako vyhovujúce počas používania aplikácie viacerými užívateľmi a následnej odozvy z ich strany.

ZÁVĚR

Hlavným cieľom diplomovej práce bolo na aplikácii pre hudobné skupiny, ktorá bude slúžiť ako multimedialne úložisko dát, predviesť použitie špeciálnych nástrojov pre prácu s multimedialnymi dátami. Táto aplikácia mala využívať databázový systém Oracle a mala poukazovať na široké využitie a uplnatnenie špeciálnych nástrojov, ktoré Oracle obsahuje. Vzhľad aplikácie mal pôsobiť moderne a pritom si zachovať dôraz na jednoduchosť a intuitívnosť ovládania. Špeciálne nástroje mali byť k dispozícii pomocou jednoduchého ovládania.

Keďže údaje a informácie v aplikácii sa môžu meniť každou chvíľou, takmer všetky údaje ktoré sú užívateľovi zobrazené sú získané z databázy. Výnimkou sú časti textu, ktorý je nemenný ako napríklad menu aplikácie. V databáze sú často uložené veľmi citlivé dáta, preto bol kladený dôraz aj na zabezpečenie celej aplikácie.

Aplikácia ponúka priestor pre rôzne typy užívateľov. Užívatelia sa delia na bežného, neregistrovaného užívateľa, registrovaného užívateľa a registrovanú skupinu. Pre každý typ užívateľa sú nastavené určité práva a obmedzenia. Bežný, neregistrovaný užívateľ môže aplikáciu využívať iba ako pozorovateľ. Nemá žiadne práva na upravovanie aplikácie či údajov v nej. Je to aj zabezpečenie proti rôznym robotom, ktoré by mohli napríklad vkladať komentáre či hodnotenia skupiny.

Hlavný dôraz bol kladený na prácu s obrázkom, kde užívateľ môže tento obrázok pomocou špeciálnych nástrojov Oracle upravovať a ukladať. Medzi funkcie na úpravu obrázku patrí napríklad upravenie jasu, kontrastu, orezanie, zmena veľkosti či rotácia obrázku. Všetky tieto funkcie sú pomocou systému Oracle. Upravený obrázok je následne automaticky zobrazený vo všetkých častiach, kde sa nachádza. Užívateľ má taktiež možnosť prehľadávať databázu s uloženými obrázkami a vyhľadať spomedzi nich tie, ktoré odpovedajú užívateľom zvolenej miere podobnosti s obrázkom, ktorý nahral. Vďaka tomu môže užívateľ vyhľadať informácie o skupine na základe podobnosti s obrázkom, ktorý vlastnil a ktorý sa taktiež nachádza medzi obrázkami skupiny. Taktiež som sa snažil prispôsobiť zdrojový kód tak, aby bol všeobecnejšie použiteľný. To znamená, aby boli využívané jednotlivé časti kódu pre viaceré operácie.

V teoretickej časti práce vysvetlujem a opisujem konkrétne pojmy, ktoré s aplikáciou súvisia. Taktiež čitateľovi približujem jednotlivé metódy a funkcie systému Oracle.

V praktickej časti popisujem jednotlivé funkcie, ktoré boli v aplikácii vytvorené. Opis funkcií je doplnený o obrázky a taktiež krátke ukážky kódu, ktoré by mohli celú problematiku priblížiť.

Výsledkom práce je funkčná aplikácia, ktorá ponúka priestor pre hudobné skupiny ako aj pre hudobných fanúšikov. Keďže pre hudobné skupiny existuje niekoľko portálov, snažil som sa využiť niektoré funkcie, ktoré sa na týchto portáloch nachádzajú a zjednotiť ich do jednej aplikácie.

Verím, že táto práca pomôže čitateľovi, ktorý sa rozhodne pre vytvorenie podobnej aplikácie ako inšpirácia ale taktiež, že rozšíri obzory a priblíži systém Oracle a jeho funkcie aj bežnému čitateľovi.

ZÁVĚR V ANGLIČTINĚ

The main aim of the thesis was the application for musical groups, which will serve as a repository of multimedia data demonstrate the use of special tools of Oracle for working with multimedia data. This application should use the Oracle database system and should point to the widespread use special tools that Oracle has. Appearance of the application should looks modern, while maintaining its emphasis on simplicity and intuitive controls. Special tools are made available through a simple control. As data and information in the application may change at any moment, almost all the data that is displayed to the user are obtained from the database. Exceptions are part of the text, which is fixed as the application menu. The databases are often very sensitive stored data, so the focus was also on the security of the entire application.

The application offers space for different types of users. Users are divided into ordinary, non-registered users, registered users and registered group. For each type of user to set specific rights and restrictions. Normal, non-registered users can use the application only as an observer. He has no right to modify applications or data in it. It is also secure against various robots, which could for example upload comments or user groups.

The main focus was to work with the image where the user can this file using special tools Oracle edit and save. Among editing functions include, for example adjust brightness, contrast, crop, resize or rotate the image. All these functions are using Oracle. The modified image is then automatically displayed in all the parts where it is located. The user also has the option to search a database of stored images and search among them those that correspond to the chosen degree of similarity with the image you uploaded. This helps the user find information about the group on the basis of similarities with the image that owned and which also contains the pictures of the group. I also tried to adjust the source code to make it more generally applicable. This means that different parts of the code used for several operations.

In the theoretical part of the thesis explains and describes the specific terms that relate to the application. I also introduce the reader to the different methods and features of Oracle.

The practical part describes the various features that were created in the application. Functional description is accompanied by pictures and also little bits of code that could bring the whole issue.

The result is a functional application that provides space for musical groups as well as for music fans. Since the bands there are several portals, I tried to use some of the features that are found on these portals and unite them into one application.

I believe that this work will help the reader who decides to create a similar application as an inspiration but also to expand their horizons and bring the Oracle system and its functions and normal readers.

SEZNAM POUŽITÉ LITERATURY

- [1] PROCHÁZKA, David. Oracle: průvodce správou, využitím a programováním nad databázovým systémem. 1. vyd. Praha: Grada, 2009, 168 s. ISBN 978-80-247-2762-2
- [2] LACKO, Luboslav. Oracle: správa, programování a použití databázového systému. 2., dopl. vyd. Brno: Computer Press, 2007, 576 s. ISBN 978-80-251-1490-2.
- [3] SHELDON, Robert a Ludvík ROUBÍČEK. SQL: začínáme programovat. 1. vyd. Praha: Grada, 2005, 499 s. ISBN 8024709996.
- [4] GROFF, James R a Paul N WEINBERG. SQL: kompletní průvodce. Vyd. 1. Brno: CP Books, 2005, 936 s. ISBN 80-251-0369-2.
- [5] LONEY, Kevin a Bob BRYLA. *Mistrovství v Oracle Database 10g*. Vyd. 1. Brno: Computer Press, 2006, 700 s. ISBN 80-251-1277-2.
- [6] KRCH, David. Oracle a multimédia?. In: *Www.dbsvet.cz* [online]. 2005 [cit. 2013-04-25]. Dostupné z: <http://www.dbsvet.cz/view.php?cisloclanku=2005072902>
- [7] Set Up Git. *GitHub* [online]. 2013 [cit. 2013-05-22]. Dostupné z: <https://help.github.com/articles/set-up-git>
- [8] ORDImage and ORDImageSignature. *Oracle® interMedia Reference* [online]. 2005 [cit. 2013-05-22]. Dostupné z: http://docs.oracle.com/cd/B19306_01/appdev.102/b14297/ch_imgref.htm#BABFEBIF
- [9] Oracle Multimedia ORDImage Object Type. *Oracle® interMedia Reference* [online]. 2010 [cit. 2013-05-22]. Dostupné z: http://docs.oracle.com/cd/E11882_01/appdev.112/e10776/ch_imgref.htm

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PHP Hypertext Preprocessor, pôvodne Personal Home Page

HTML Hypertext Markup Language

SQL Structured Query Language – dotazovací jazyk

WWW World Wide Web

ID Identifikačné číslo

PX Označenie jednotky pixel

SEZNAM OBRÁZKŮ

<i>Obrázok č. 1 – Použitie jazyka SQL pre prístup k databáze</i>	16
<i>Obrázok č. 2 – Databázový stroj.....</i>	17
<i>Obrázok č. 3 – Rozdelenie SQL</i>	18
<i>Obrázok č. 4 – Grafické znázornenie výsledkov dotazníka</i>	29
<i>Obrázok č. 5 – Výpis stránky test.php</i>	31
<i>Obrázok č. 6 – Schéma prepojenia tabuliek</i>	32
<i>Obrázok č. 7 – Štruktúra tabuliek</i>	33
<i>Obrázok č. 8 – Vzhľad aplikácie</i>	35
<i>Obrázok č. 9 – Vrchná časť hlavnej stránky.....</i>	35
<i>Obrázok č. 10 – Spodná časť hlavnej stránky.....</i>	36
<i>Obrázok č. 11 – Registračný formulár.....</i>	38
<i>Obrázok č. 12 – Zobrazenie zoznamu dostupných blogov.....</i>	39
<i>Obrázok č. 13 – Hodnotenie skupiny</i>	40
<i>Obrázok č. 14 – Zadávanie textu pre oakordovanie</i>	42
<i>Obrázok č. 15 – Oakordovaný text.....</i>	43
<i>Obrázok č. 16 – Schránka správ</i>	43
<i>Obrázok č. 17 - Stránkovanie</i>	45
<i>Obrázok č. 18 – Úprava profilovej fotky – vrchná časť.....</i>	49
<i>Obrázok č. 19 - Úprava profilovej fotky – spodná časť.....</i>	49
<i>Obrázok č. 20 – Automatické vytvorenie náhľadu</i>	52
<i>Obrázok č. 21 – Vyhľadávanie podobnosti obrázkov – nahranie fotky</i>	53
<i>Obrázok č. 22 – Vyhľadávanie podobnosti obrázkov - výsledok</i>	53
<i>Obrázok č. 23 – Prehrávač hudby.....</i>	54

SEZNAM TABULEK

Tabulka č. 1 – Ukážka tabulky Skupina Chyba! Záložka není definována.