

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

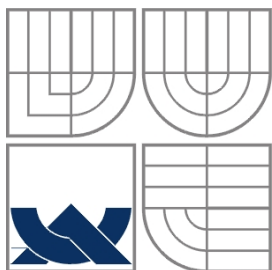
MULTIUŽIVATELSKÝ SYSTÉM PRO PODPORU
ZNOVUVYUŽITÍ MATERIÁLŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

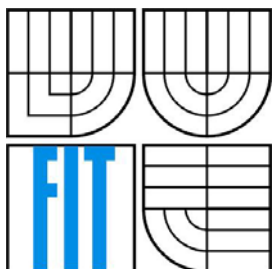
AUTOR PRÁCE
AUTHOR

Petr Kolarik

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MULTIUŽIVATELSKÝ SYSTÉM PRO PODPORU ZNOVUVYUŽITÍ MATERIÁLŮ

MULTI-ACCESS SYSTEMS FOR SUPPORT OF RECOVERABLE MATERIALS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Petr Kolarik

VEDOUCÍ PRÁCE

SUPERVISOR

Mgr. Zdeněk Martínek

BRNO 2007

Abstrakt

Tento text je dokumentací k multiuživatelskému systému pro podporu znovuvyužití materiálů. Rozebírá možnosti struktury v závislosti na funkční specifikaci systému, jeho implementaci pomocí PHP za využití databáze MySQL. Dále rozebírá postup tvorby systému od ER diagramu přes diagram užití až po samotné programování. Tato práce ukazuje jak navrhnout webový informační systém umožňující uživateli definovat víceúrovňové osobní pohledy na obsažená data.

Tento projekt by mohl být základem pro komerční projekt, který může prověřit vhodnost použití navržené struktury jednotlivých částí systému.

Klíčová slova

PHP, CSS, HTML, MySQL, ER diagram, diagram užití, JavaScript, inzertní systém, databáze, Apache, engine, OLAP (Online Analytical Processing), XHTML, SGML, ASP

Abstract

This text is documentation for multi-access system, which supports recoverable materials. It deals with structure possibilities according to functional system specification and its implementation through the PHP together with using MySQL database system. It analyses a progress of system creation from ER diagram through use-case diagram to programming itself. This work shows how to design web advertisement system which enables an user to define personal multi-level views on data. This project might have been as basis for commerce project, which can check up usability designed structure of individual parts.

Keywords

PHP, CSS, HTML, MySQL, ER diagram, use-case diagram, JavaScript, advertisement system, database, Apache, engine, OLAP (Online Analytical Processing), XHTML, SGML, ASP

Citace

Petr Kolarik: Multiuživatelský systém pro podporu znovuvyužití materiálů, diplomová práce, Brno, FIT VUT v Brně, 2007

Multiuživatelský systém pro podporu znovuvyužití materiálů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Mgr. Zdeňka Martínka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Kolarik
30.04.2007

Poděkování

Tímto děkuji Mgr. Zdeňkovi Martínkovi a všem přispěvatelům a diskutujícím na diskusních fórech za jejich pomoc, kterou mi při řešení projektu poskytli.

© Petr Kolarik, 2007.

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

Obsah

Obsah.....	1
1 Úvod	3
2 Implementační prostředky	5
2.1 HTML (Hypertext Markup Language).....	5
2.2 CSS (Cascading Style Sheet)	6
2.3 Jazyk PHP	7
2.4 Databázový systém MySQL.....	8
2.5 Xampp	9
2.6 JavaScript	9
3 Analýza	11
3.1 Registrace	11
3.2 Přihlášení	12
3.3 Inzerát.....	12
3.3.1 Povinné položky.....	12
3.3.2 Rozšíření inzerátu	13
3.4 Navigační menu.....	14
3.4.1 Hlavní stromová struktura.....	15
3.4.2 Vedlejší stromová struktura	16
3.5 Vyhledávání	17
3.5.1 Parametry vyhledávání.....	17
4 Implementace	20
4.1 ER diagram.....	20
4.2 Tabulky a jejich význam	21
4.2.1 Uživatelé	21
4.2.2 Kategorie.....	22
4.2.3 Rozšíření inzerátu	23
4.2.4 Inzeráty	24
4.2.5 Číselník kategorií.....	25
4.3 Diagram užití.....	26
4.4 Struktura zdrojových kódů	27
4.4.1 Hlavní stránka systému	27
4.4.2 Registrace uživatele	29
4.4.3 Přihlášení a odhlášení uživatele	29
4.4.4 Administrace	29

4.4.5	Inzerát	33
4.4.6	Stránka vyhledávání.....	36
5	Závěr.....	39
5.1	Zhodnocení výsledků	39
5.2	Vlastní přínos	39
5.3	Návaznost.....	39
	Literatura	40
	Seznam příloh	41

1 Úvod

Inzertní systém umožňuje uživatelům přehledným a rychlým způsobem nabídnout své zboží různého druhu a tím oslovit velké množství potenciálních zákazníků. Takovýto systém je v podstatě databáze těchto nabídek, označovaných jako inzeráty. V dnešní době se lze s nabídkou zboží setkat v inzertních denících, inzertní systém je pak elektronickou obdobou těchto deníků a je prezentován prostřednictvím internetu.

Při hledání na internetu lze nalézt spoustu inzertních systémů odlišných zaměřením, přehledností a z ní vyplývající kvalitou. Jedna z nejčastějších oblastí pro inzertní systém jsou nemovitosti. Pokud se podíváme na jejich funkční zaměření, pak získáme v podstatě dvě varianty, které jsou z velké části ovlivněny počtem příspěvů neboli velikostí používané databáze příspěvků (inzerátů). První z nich, jenž obsahuje rozsáhlou databázi inzerátů, je zaměřena především na kvalitní vyhledávací engine s možností stupňovitěho vyhledávání. Pro uživatele to znamená, že může zadávat více zpřesňujících parametrů pro vyhledávání svého požadavku. Na druhou stranu zde chybí jakékoliv navigační menu s možností rychlého zobrazení požadované kategorie inzerátů. Na principu kombinace navigačního menu a vyhledávacího engine je založena druhá varianta. Ta se hojně používá pro menší systémy, jejichž databázové servery nepracují s tak velkým množstvím dat jako v prvním případě a tudíž není problém zobrazit dané množství inzerátů kategorií vyšší úrovně. Tento druhý případ bývá často přehlednější a pro uživatele příjemnější k používání. Přehledně tak může vidět, pomocí navigačního menu, strukturu kategorií inzerátů a nebo použít vyhledávání pro vypsání inzerátů s konkrétními parametry.

Multiuživatelský systém pro podporu znovuvyužití materiálů využívá přehledného navigačního menu se stromovou strukturou i rozšířeného vyhledávání.

Intenzivní nasazení informačních technologií do rozličných odvětví lidské činnosti přináší sebou jeden zákonitý jev, kterým je shromažďování velkého množství údajů. Moderní databázové servery umožňují nejen bezpečnou a rychlou práci s takovým množstvím údajů, ale umožňují nám z těchto údajů získat i informace. Taková transformace převodu na informace a následně na poznatky prostřednictvím objevování se nazývá Business Intelligence. Poznatky můžeme poté efektivně využít například v procesu rozhodování. Proto moderní databázové servery obsahují rozsáhlou podporu pro budování datových skladů (data warehouse), analýzy OLAP (OnLine Analytical Processing) a dolování dat (data mining).

Pro přiblížení jedné z těchto technologií je zvolena OLAP, která je typická pro analýzu velkého množství údajů. Výsledkem takové analýzy jsou souhrny a reporty, které slouží uživateli jako podklady pro jeho rozhodování, ať už v oblasti řízení firmy nebo ekonomických a technologických procesů. V případě multiuživatelského systému pro

podporu znovuvyužití materiálů bychom mohli nejlépe využít zmiňované technologie k analýze všech inzerátů a jejich vlastníků. A to tím způsobem, že z dostupných údajů o materiálech vytvoříme přehled zastoupení jednotlivých materiálů, jejich nabídek a poptávek v závislosti na čase či majiteli. Avšak díky použití nekomerčního produktu MySQL, který nemá podporu pro vytváření těchto analýz OLAP, bylo využito pouze myšlenky získání kvalitního přehledu nabídky a poptávky jednotlivých materiálů.

V této práci se nejdříve seznámíte s implementačními prostředky, které byly v průběhu použity k vytvoření jednotlivých částí nebo i celého projektu. Ať už se jedná o webový či databázový server, software pro implementaci CSS, PHP jazyka, JavaScriptu. Následuje analýza nejdůležitějších částí samotného multiuživatelského systému, mezi něž patří navigace, možnosti inzerátů, vyhledávání a další. Analýza obsahuje funkční aspekty, způsob použití a jejich možnosti. Dále nechybí implementace projektu. Obsahuje popis implementace použité databáze prostřednictvím ER diagramu, podrobného popisu všech tabulek databáze a přehledného diagramu užití. A dále postupy a popis hlavních částí systému, použitých funkcí.

Tato práce je návazností na Ročníkový projekt, ve kterém byl navrhnout a částečně realizován inzertní systém s triviálním vyhledáváním a navigačním menu stromové struktury. Následný Semestrální projekt ukázal především možnosti využití analýzy OLAP v komerčních databázových serverech (např. MS SQL) v takovémto inzertním systému. Zároveň byl proveden návrh rozšířeného vyhledávání, inspirovaný právě analýzou OLAP a její základní myšlenkou.

2 Implementační prostředky

V poslední době se při tvorbě internetových stránek uplatňuje filozofie rozdělování zdrojových kódů do různých souborů podle toho, jaký má daný kód význam. Internetová stránka by tedy měla být v ideálním případě rozložena jednak do souboru, který obsahuje pouze HTML kód, jenž určuje strukturu HTML dokumentu a pojmenovává jeho jednotlivé části, pokud možno bez jakékoliv definice vzhledu jednotlivých oddílů stránky. Dále do dokumentu, jenž naopak definuje vzhled nejrůznějších prvků nebo celých skupin pomocí kaskádových stylů CSS, které jsou využívány ve stále hojnější míře a konečně, je-li to potřeba, pak také do dokumentu, který obsahuje deklaraci chování stránky a tedy všechny používané funkce. Kompatibilita, nebo spíše nekompatibilita dnešních prohlížečů se stanovenými standardy však nutí člověka při tvorbě stránek porušovat toto jinak takřka ideální rozdělování, sloužící především ke zpřehlednění kódů a jednoduchosti jeho úprav. Avšak dnes už existují různé programy, které umožňují pomocí jednoduché syntaxe zpřehlednění kódu např. Smarty [6]. V poslední době se do podvědomí, mezi vývojáři webových aplikací, dostává také tzv. PEAR (PHP Extension and Application Repository). Jedná se o framework společně s distribučním systémem knihoven a rozšířeními pro aplikace psané v jazyce PHP [7].

Pokud se dnes rozhodneme začít vytvářet dynamické internetové stránky, jistě začneme pátrat po nejnovějších a nejvhodnějších databázových systémech, webovém serveru či různých nástrojích. Chceme-li se držet na úrovni nekomerčního software, pak nalezneme webový server Apache, multiplatformní databázi MySQL a z nástrojů např. PhpMyAdmin. PhpMyAdmin je často používaný nástroj napsaný v jazyce PHP umožňující jednoduchou správu obsahu databáze MySQL, jako je vytváření tabulek, jejich modifikaci, odstraňování a celou řadou dalších úkonů, prostřednictvím webového rozhraní.

Nyní máme dvě možnosti jak zvolený software nainstalovat. Buď jednotlivě program po programu a nebo komplexně jako jeden balíček. Dnes je stále populárnější instalace balíčků, obsahující potřebné přednastavené součásti společně s množstvím dalších nástrojů, které nám ulehčují práci. Mezi neznámější bezesporu patří PHPtriad, Wamp, Xampp aj.

2.1 HTML (Hypertext Markup Language)

HTML je zkratka z anglického HyperText Markup Language, značkovací jazyk pro hypertext [8]. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci stránek na Internetu. Jazyk je podmnožinou dříve vyvinutého rozsáhlého

univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

Značky můžeme z hlediska významu rozdělit na tři základní skupiny:

- Strukturální značky jsou ty, které rozvrhují strukturu dokumentu. Příkladem jsou odstavce (<p>), nadpisy (<h1>, <h2>). Dodávají dokumentu formu.
- Popisné (sémantické) značky popisují povahu obsahu elementu. Příklad nadpis (<title>) nebo adresa (<address>). Současný trend je orientován právě na sémantické značky, které usnadňují automatizované zpracovávání dokumentů a vyhledávání informací v záplavě dokumentů na webu. Vyvrcholením této snahy je v současné době jazyk XML.
- Stylistické značky určují vzhled elementu při zobrazení. Typickým příkladem je značka pro tučné písmo (). Tento druh značek se nedoporučuje používat, trendem je používání kaskádových stylů oddělených od obsahu dokumentu. Problém je především v tom, že tyto značky jsou orientovány na obrazovku monitoru, problémy mohou nastat při používání dokumentu jiným způsobem - alternativní prohlížeče pro postižené (čtečky pro slepce), v mobilních zařízeních a podobně. Kaskádové styly umožňují definovat rozdílné zobrazení pro různá zařízení.

Ve velice krátkém časovém intervalu byl široce rozšířen a přijat. S postupem doby se potřeby uživatelů rozšiřovaly a nebo i částečně měnily. Z tohoto důvodu byly zapotřebí verze specifikací jazyka HTML, která dospěla až do verze 4.0x, ve které již pravděpodobně zůstane zakonzervován, poněvadž bude snaha o nahrazení technologií XHTML, která z HTML vychází XHTML řeší nedostatky, které měla a nebylo je možné vyřešit bez zásadnějšího zásahu do struktury.

2.2 CSS (Cascading Style Sheet)

CSS [9] vzniklo někdy kolem roku 1997. Původní myšlenkou jejich vzniku byl záměr, aby HTML kód nesl jen a pouze informační hodnotu a způsob vykreslení byl definován odděleně. Jeho důsledkem mělo a má být i do budoucna především lepší vyhlídka pro indexování vyhledávacími roboty typu Gogole, seznam, morfeo a další. Je to kolekce metod pro grafickou úpravu webových stránek. Ta zkratka znamená Cascading Style Sheets, česky "kaskádové styly". Kaskádové, protože se na sebe mohou vrstvit definice stylu, ale platí jenom ta poslední.

Již existuje návrh CSS 2 obsahující vylepšené a složitější formy stylů, které ale v nejrozšířenějším prohlížeči Internet Exploreru v mnoha případech nefungují.

CSS zatím nepodporují mobilní zařízení, například prohlížeče Ipaqů. I v nich ale lze text přečíst. CSS nepodporují ani velmi staré prohlížeče (Explorer 2 a Navigator 3), které se však už nikde nevyskytují, a textové prohlížeče (lynx, links) - v takových se text nezformátuje (zůstává čitelný).

Pro představu, zde je několik možností použití kaskádových stylů:

- nastavit libovolnou a přesnou velikost písma, prokládání, kapitálky
- udělat odsazení prvního řádku odstavce, zvětšit řádkování
- zrušit nebo zvětšit prázdný prostor po odstavci
- automaticky formátovat nadpisy (například je všechny udělat zelené)
- zvýrazňovat odkazy po přejetí myši
- udělat automaticky grafické odrážky
- určité části textu zprůhlednit nebo nezobrazit

2.3 Jazyk PHP

Vzhledem k tomu, že všechny stránky celého systému pracují s proměnnými, databázi a dalšími prostředky, které samotné HTML nenabízí, byl k jeho vytvoření využit dnes nejběžnější skriptovací jazyk PHP [10], který teprve slouží ke generování HTML kódu jednotlivých stránek. Pro tuto práci byl použit ve verzi 5.1.1. běžící na serveru localhost, obsahující plnou podporu relací, které jsou v tomto projektu použity k ověření správnosti přihlašovacích údajů jednotlivých uživatelů. Hlavní předností jazyka PHP je jeho volná dostupnost.

PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, „PHP: Hypertextový preprocessor“, původně Personal Home Page) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což je velmi výhodné pro tvorbu webových aplikací. PHP lze ovšem také použít i k tvorbě konzolových a desktopových aplikací.

PHP skripty jsou prováděny na straně serveru, k uživateli je přenášen až výsledek jejich činnosti. Syntaxe jazyka kombinuje hned několik programovacích jazyků (Perl, C, Pascal a Java). PHP je nezávislý na platformě, skripty fungují bez úprav na mnoha různých operačních systémech. Obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových serverů (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL), podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP, ...)

PHP se stalo velmi oblíbeným především díky jednoduchosti použití a tomu, že kombinuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu

v syntaxi. V kombinaci s databázovým serverem (především s MySQL nebo PostgreSQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací. Díky velmi častému nasazení na serverech se vžila zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP nebo Perl.

S verzí PHP 5 se výrazně zlepšil přístup k objektově orientovanému programování podobný Javě.

2.4 Databázový systém MySQL

Jako systém řízení báze dat (SŘBD) nebylo zvoleno MySQL, které dnes také patří k nejběžnějším a které je rovněž zdarma ke stažení na internetu [11].

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace MySQL, PHP a Apache jako základní software webového serveru.

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, triggerly, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet.

Přehled podporovaných vlastností:

- cizí klíče (od verze 3.23 podporovány v tabulkách typu InnoDB)
- transakce (od verze 3.23 podporovány v tabulkách typu InnoDB)
- podpora různých znakových sad a časových pásem v datech (od verze 4.1)
- poddotazy (od verze 4.1)
- uložené procedury (od verze 5.0)
- triggerly (od verze 5.0)
- pohledy (od verze 5.0)

MySQL nabízí několik typů databázových tabulek (storage engine), které se liší svými možnostmi, použitím a způsobem ukládání dat do souborů:

- MyISAM - nejpoužívanější, bez podpory transakcí
- InnoDB - podpora transakcí
- BerkeleyDB (BDB)

- MEMORY - práci s daty v paměti
- NDB Cluster - uložisko pro clusterované databáze (od verze 5.0)
- ARCHIVE - komprimované tabulky, bez podpory indexů
- CSV - ukládání dat v prostých textových souborech

2.5 Xampp

Balík předkonfigurovaných instalací PHP, Apache, MySQL a dalších. Apache 2, PHP 5, MYSQL 5, phpMyAdmin ve výborně sestavené instalaci, příklad navazuje na již dříve publikovaný popis instalace PHPTriad, Sokkit, Wamp - Apache, MySQL, který je již však svou konfigurací za horizontem použitelnosti.

XAMMP [12] vám již dá mnohem větší možnosti při práci, jak svou konfigurací vhodnou pro instalaci moderních publikačních systémů, tak jistě pokračováním filosofie, protože jednoduše nepoužíváte FTP, ani vzdálený online server. Výhodou, je jistě i to, že nikdo tak nevidí chyby, protože je můžete v klidu opravit, nejste prostě na online serveru, ale na tzv. localhostu. Můžete tedy pracovat i bez připojení k internetu, což může jistě být přínosné ba často profesně přímo nutné.

Přehled součástí základního balíčku verze 1.5.1:

- Apache 2.2.0, phpMyAdmin 2.7.0 pl1
- MySQL 5.0.18, SQLite 2.8.15, ADOdb 4.68
- PHP 5.1.1, PHP 4.4.1-pl1, PEAR, PHP-Switch win32 1.0
- XAMPP Control Version 2.1, XAMPP Security 1.0, OpenSSL 0.9.8a
- Mercury Mail Transport System v4.01b, FileZilla FTP Server 0.9.11
- Webalizer 2.01-10, Zend Optimizer 2.6.0, eAccelerator 0.9.4-rc1
- FPDF 1.5.3

2.6 JavaScript

JavaScript [13] je multiplatformní, objektově orientovaný skriptovací jazyk. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta), na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z

toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele.

Existují i jiné jazyky klientských skriptů, například VBScript. Jsou ale tak málo používané, že když se dnes mluví o "skriptech", myslí se tím JavaScripty.

JavaScript je často zaměňován s Javou. Java je samostatný programovací jazyk. Má s JavaScriptem pouze podobnou syntaxi ale nic víc. Jednalo se spíše o marketingový tah firmy Netscape, která využila komerčního úspěchu platformy JAVA a pojmenovala tak skriptovací jazyk, kterým bylo možné implementovat některé interaktivní vlastnosti na straně klienta.

JavaScript je jazyk:

- Interpretovaný - nemusí se kompilovat, ale jen částečně z tzv. byte kódu.
- Objektový - využívá objektů prohlížeče a zabudovaných objektů
- Závislý na prohlížeči - funguje ale ve většině prohlížečů
- Case sensitivní - záleží na velikosti písem v kódu
- Syntaxí podobný jazykům C, Java atd.

Omezení jazyka JavaScript:

- JavaScript funguje pouze v prohlížeči. Nemá žádné jiné použití v žádné jiné oblasti.
- Uživatel může JavaScript zakázat. Nelze se tedy z pohledu programátora spolehnout na jeho dostupnost ze strany uživatele.
- Existují různé odlišné verze jazyka i prohlížečů, což vede k častým chybám.
- Neumí přistupovat k souborům (kromě cookies) ani k žádným systémovým objektům. Toto omezení je záměr pro zajištění bezpečnosti.
- Neumí žádná data uložit (kromě cookies). Cookies jsou nadstavba nad protokolem HTTP, která dodává do protokolu možnost sledování stavu. Protokol HTTP je samotný bezstavový.

3 Analýza

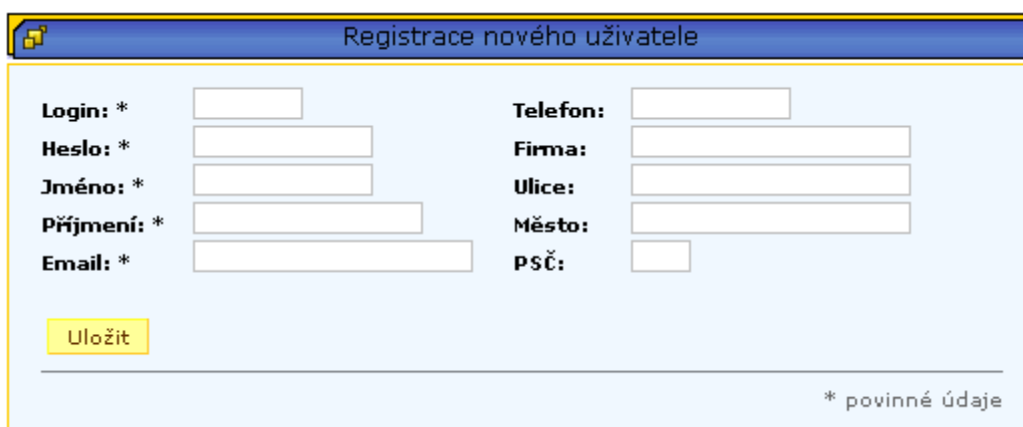
Důležitým kritériem při vytváření systému je snadné a přehledné ovládání a orientace. Systém, který je určen pro běžného uživatele internetu, by neměl být příliš komplikovaný a složitý na ovládání. To by navíc nemělo být naprosto odlišné od klasických aplikací. Dále by všechny potřebné informace měly být snadno a rychle dostupné.

Hlavním úkolem je přehledné fungování multiuživatelského systému jako webové aplikace. Jelikož je použito navigačního menu se stromovou strukturou, získává uživatel jednoznačný přehled o všech dostupných kategoriích inzerátů, které systém obsahuje.

Uživatel má několik možností, jak systém používat a to v závislosti na typu uživatele. V dalších částech bude popisován přístup běžného uživatele. V případě jiného přístupu bude uvedeno o jaký přístup se jedná. Po vstupu na stránky informačního systému má uživatel možnost několika akcí.

3.1 Registrace

Pokud vstoupil uživatel na stránky inzertního systému a nemá v úmyslu přispět do systému podáním vlastního inzerátu, je mu nabídnuta možnost vyhledávání inzerátů a procházení stromové struktury kategorií navigačního menu. V opačném případě má možnost registrace pomocí formuláře registrace.



Registrace nového uživatele

Login: *	<input type="text"/>	Telefon:	<input type="text"/>
Heslo: *	<input type="text"/>	Firma:	<input type="text"/>
Jméno: *	<input type="text"/>	Ulice:	<input type="text"/>
Příjmení: *	<input type="text"/>	Město:	<input type="text"/>
Email: *	<input type="text"/>	PSČ:	<input type="text"/>

* povinné údaje

Obrázek 3.1.1

Formulář vybízí uživatele k zadání osobních údajů, jako jsou *login*, *heslo*, *jméno*, *příjmení* a kontaktní *email*, které jsou povinné a dále *telefon*, *firma*, *ulice*, *město* a *psč*, které jsou volitelné. Po odeslání dat systém ověřuje správné zadání povinných údajů. Položku *login* kontroluje s existujícími daty v databázi a pokud již uživatel s daným *loginem* existuje,

objeví se příslušné upozornění o nutnosti výběru jiného. Totéž platí i v případě *emailu*, avšak tam se očekává, že uživatel svoji emailovou adresu zná a narozdíl od *loginu* je *email* unikátní. Také se prověřuje správný formát *emailu*. Pokud ověření proběhne úspěšně, uživatel je přidán do databáze a je mu umožněn přístup do celého systému.

3.2 Přihlášení

Pokud není uživatel registrovaný, nemůže se do systému přihlásit. Po úspěšné registraci, kdy jsou jeho údaje zapsány do databáze, se smí pomocí svého *loginu* a *hesla* přihlásit do multiuživatelského systému. I v tomto případě dochází k ověřování existence *loginu* a příslušného *hesla* v databázi.

Přihlášený uživatel tak získává, narozdíl od nepřihlášeného, více akcí, které může v rámci systému provádět.

3.3 Inzerát

Už z názvu inzertní systém je zřetelné, že celá aplikace má úkol prezentovat inzeráty uživatelů. Vkládat inzeráty může pouze registrovaný uživatel, který je přihlášen do systému. Za odkazem Vložení inzerátu se skrývá formulář k vyplnění údajů o inzerátu.

3.3.1 Povinné položky

Formulář obsahuje povinné a nepovinné položky. Mezi povinné patří *Kategorie* pro inzerát, což je kategorie hlavního stromu, do které bude inzerát vložen. Dále *Nadpis* a *Popis* – neboli stručná charakteristika nabízeného (požadovaného) zboží.

Další povinnou položkou je v sekci *Osobní údaje Email* (obr. 3.3.1.1). U osobních údajů má uživatel na výběr. Pokud se data, které chce přidat k inzerátu, shodují s těmi, které uvedl při registraci, pak stačí zaškrtnout checkbox Použít osobní údaje z registrace. V opačném případě může vyplnit údaje pro inzerát dle svého uvážení, s tím, že *Email* je povinný.

Osobní údaje	
(použít os. údaje z registrace)	<input type="checkbox"/>
Jméno:	<input type="text"/>
Příjmení:	<input type="text"/>
Firma:	<input type="text"/>
Ulice:	<input type="text"/>
Město:	<input type="text"/>
PSČ:	<input type="text"/>
Email: *	<input type="text"/>
Telefon:	<input type="text"/>

Obrázek 3.3.1.1

3.3.2 Rozšíření inzerátu

Důležitou součástí inzerátu je kromě kontaktních údajů a popisu, také rozšíření inzerátu. Pod tímto pojmem si můžeme představit vlastnosti materiálů, které jsou předmětem inzerátu.

Rozšíření inzerátu: vyberte údaje které chcete vyplnit	
Tloušťka	2 <input type="button" value="v"/> <input type="text"/> <input type="text"/>
Šířka	1 <input type="button" value="v"/> <input type="text"/>
Délka	3 <input type="button" value="v"/> <input type="text"/> <input type="text"/> <input type="text"/>
Průměr	1 <input type="button" value="v"/> 2 <input type="button" value="v"/> 3 <input type="button" value="v"/> <input type="text"/>
Typ inzerátu	Jiné <input type="button" value="v"/>
Množství	<input type="text"/>

Obrázek 3.3.2.1

3.3.2.1 Přesně definovaná položka (kategorie)

Mezi rozšířením by se měla také objevit položka *Typ inzerátu*, která sice není označena jako povinná, ale její hodnota musí být vybrána. To je ošetřeno tím, že pokud uživatel opomene zvolit *Typ inzerátu*, je implicitně nastavena první hodnota výběru.

Poprvé zde narážíme na fakt, že kategorie vedlejšího stromu musí mít přesný název. Je to dáno tím, že se velké množství dat generuje a tudíž je velmi těžké určit, zda se jedná o položku, která je povinná, nepovinná či zda má nějakou specifickou funkci. Proto se v kódu vyskytuje ověření názvu generované položky rozšíření inzerátu, které tudíž klade nároky na přesně definovaný název položky, přesněji *kategorie*, vedlejších stromových struktur. Konkrétně se jedná právě o *Typ inzerátu*.

Zbytek rozšíření inzerátu je zcela nepovinné. Každá položka rozšíření v podstatě odpovídá jedné z kategorií vedlejšího stromu. Což znamená, že se inzerát objeví v každé kategorii odpovídající dané položce rozšíření. Pro představu, rozšíření obsahuje upřesňující data o zboží např. šířku, průměr, hmotnost a je jen na uživateli, které z nich chce vyplnit resp. které má k dispozici.

Množství položek rozšíření závisí na nastavení parametru kategorií. Admin může u každé kategorie vedlejšího stromu nastavit, zda a jak se bude daná kategorie (jako položka) zobrazovat v inzerátu.

Po odeslání formuláře je opět provedeno ověření povinných údajů tzn. jestli jsou zadány položky *Kategorie pro inzerát*, *Nadpis*, *Popis a Email* (u emailu je kontrolována správnost formátu).

Každý uživatel může vytvořit relativně neomezené množství inzerátů. Pokud se stane inzerát neaktuální nebo ho bude chtít uživatel odstranit, pak stačí kliknout v nadpisovém řádku na smazat a po potvrzení je inzerát odstraněn z databáze. Administrátor má možnost smazat jakýkoliv inzerát, běžný uživatel pouze ty jejichž je vlastníkem.

3.4 Navigační menu

Nedílnou součástí většiny inzertních systému bývá také navigační menu. Napomáhá k snadnému a intuitivnímu procházení stránek. I zde tento prvek nechybí, nicméně za běžným vzhledem se skrývá navigační menu, které má svojí funkčností a vlastnostmi netradiční charakter.

Hlavní myšlenkou bylo vytvoření menu klasické stromové struktury. Avšak nyní je umožněno vytvoření více stromů v navigačním menu s různou funkčností. Tyto stromové struktury jsou rozděleny na dvě části (obr. 3.4.1).



Obrázek 3.4.1

3.4.1 Hlavní stromová struktura

První z nich můžeme pojmenovat jako hlavní stromová struktura a obsahuje všechny materiály, které mohou být, v rámci tohoto multiuživatelského systému, nabízeny k inzerci. Pokud bychom sečetli všechny inzeráty obsažené v prvním stromu, pak získáme číslo odpovídající celkovému počtu všech inzerátů v systému. Což znamená, že každý jednotlivý inzerát spadá pouze a jen do jedné kategorie tohoto stromu. Dá se říct, že se jedná o klasickou strukturu a princip pro menu inzertního systému.

Kategorie hlavního stromu mají nastavitelné parametry (obr. 3.4.1.1). Název, popis kategorie a velikost buňky je pro hlavní i vedlejší kategorie stejný. Rozdílným parametrem je typ rozměrů kategorie, který určuje, jaké rozměry se budou pro danou kategorii používat popř. vyhledávat. To je úzce spjato i s dalšími dvěma parametry kategorie pro vyhledávání a pro výběr. První z nich nastavuje, zda se bude kategorie objevovat ve vyhledávání a jestli se zobrazí implicitně jako prvek vstupního textového pole nebo jako prvek pro výběr.

Obrázek 3.4.1.1

3.4.2 Vedlejší stromová struktura

Do druhé části spadají všechny ostatní stromové struktury, tedy 2. až N-tá. Každá vedlejší struktura v podstatě zastupuje jednu vlastnost nebo soubor vlastností materiálů hlavní struktury. Jak můžeme vidět z obr. 3.4.1, mezi takové vlastnosti patří např. rozměry (tloušťka, šířka, délka, průměr), množství, jakost. Malou výjimkou je strom s kořenovým názvem *Typ inzerátu*, který sice nevyjadřuje žádnou vlastnost materiálu, ale spíše typ inzerátu. Další odlišnost máme v tom, že jeden konkrétní inzerát může být obsažen ve více kategoriích vedlejších stromů. Záleží na tom, kolik je v inzerátu uvedeno informací, vlastností o materiálu.

Kategorie vedlejšího stromu mají následující nastavitelné parametry (obr. 3.4.2.1). Název, popis kategorie a velikost buňky je pro hlavní i vedlejší kategorie stejný. Zobrazit v rozšíření inzerátu zkrátka umožňuje, aby se daná kategorie zobrazila jako položka rozšíření inzerátu. Zajímavým je parametr přidávání více hodnot. Ten určuje, zda se při vkládání inzerátu může do stejnojmenné položky rozšíření uložit více hodnot. Pokud chceme do inzerátu nabídnout materiály, u nichž uvádíme např. průměr, avšak jediná hodnota je nedostačující, pak je vhodné tuto kategorii nastavit, tak aby uživatel mohl vložit více jak jednu (obr. 3.3.2.1). Kategorie pro vyhledávání funguje stejně jako u hlavní stromové struktury.

Poslední dvě zatím nezmíněné jsou kategorie pro zadání hodnoty a pro výběr. Zdálo by se, že nastavením obou parametrů na hodnotu „Ano“ dojde vzájemnému vyloučení, ale není tomu tak. Chceme-li, aby se u této položky rozšíření zobrazil prvek pro vstupní pole, nastavíme kategorii pro zadání hodnoty na hodnotu „Ano“. Pokud však chceme zobrazit

položku rozšíření jako prvek výběru, pak nastavíme kategorii pro výběr na hodnotu „Ano“. Sporná situace, kdy oba parametry jsou nastaveny na hodnotu „Ano“, je vyřešena tak, že se zůstane platná hodnota parametru kategorie pro zadání hodnoty a druhá se ignoruje. Důvodem, ponechání možnosti zvolit obě hodnoty „Ano“, je její využití ve vyhledávání. Tam platí prakticky totéž jako u rozšíření inzerátu s tím rozdílem, že v rozšíření by bylo zbytečné umožnit u položky zadávat a ještě vybírat hodnotu z nabídky. Kdežto u vyhledávání je tato volba vítána. Více o možnostech vyhledávání v další kapitole.

Vytvoření nové kategorie

Kategorie umístění : Průměr

Kategorie:

Popis:

Vel.buňky:

Přidávání více hodnot: Ano Ne

Zobrazit v roz. inzerátu: Ano Ne

Kategorie pro zadání hodnoty: Ano Ne

Kategorie pro vyhledávání: Ano Ne

Kategorie pro výběr: Ano Ne

Obrázek 3.4.2.1

3.5 Vyhledávání

Možnost vyhledávání inzerátů podle vybraných parametrů je umožněna všem uživatelům tzn. i těm, kteří se nezaregistrovali do systému. Vyhledávání je pro přehlednost umístěno hned na úvodní stránce a na prvním, u přihlášeného uživatele na druhém, místě v menu. To proto, aby nemusel uživatel složitě hledat v navigačním panelu položku pro vyhledávání. Výhodnější než současné umístění se nabízí hned v záhlaví každé stránky, čímž by byla možnost vyhledávání inzerátů neustále k dispozici.

3.5.1 Parametry vyhledávání

Jaké jsou možné parametry? Jelikož je počet těchto parametrů velmi variabilní a je možné ho měnit, uvedl bych především ty základní (obr. 3.5.1.1). První skupinku tvoří jednotlivé stromové struktury a jejich kategorie a podkategorie. Budeme předpokládat, že náš systém je tvořen ze dvou stromových struktur, zkráceně stromů. První strom se dá pojmenovat jako

hlavní struktura, která obsahuje kategorie značící např. rozdělení podle materiálu. Označení hlavní struktura také znamená to, že každý inzerát, vložený do inzertního systému, se nachází v tomto hlavním stromě. U druhého (popř. následujících) stromu tomu tak být nemusí, jelikož kategorie stromu jsou zároveň položky rozšíření inzerátu (např. rozměry materiálu – průměr, délka apod.) Další skupinku mohou tvořit jednotlivé specifikace inzerátu jako například typ. Typem inzerátu rozumíme zdali se jedná o nabídku k prodeji, koupi, výměně nebo jiné.

Vyhledávání v inzerátech

Najít:

Materiály Typ inzerátu

Rozměry: A B C

Počet nalezených inzerátů splňující zadané parametry : 2

ID:2 Pásky Koupě

Petr Kolarik • U Věže 277/3, 46001 Liberec
 +420732702040 • gobi.zr@centrum.cz

pásky různých rozměrů

- Tloušťka: 14, 5 • Šířka: 180, 120, 100 • Délka: 1000, 1500

ID:3 Tyče Koupě

Petr Kolarik • U Věže 277/3, 46001 Liberec
 +420732702040 • gobi.zr@centrum.cz

tyče o dvou rozměrech

- Délka: 800 • Průměr: 15, 32

Obrázek 3.5.1.1

3.5.1.1 Základní (povinné) parametry

Pod pojmem základní parametry rozumíme nutné položky vyhledávání, které by se měly v tomto multiuživatelském systému objevit. Jelikož se jedná o systém na podporu znovuvyužití materiálů je tím pádem nejvhodnější oblast uplatnění strojírenský průmysl resp. hutní materiály. Mezi základní parametry hutních materiálů pro vyhledávání jsou:

- Výběr z kategorií materiálů.
- Typ inzerátu (povinná u většiny inzertních systémů)
- Rozměry materiálů (šířka, délka, tloušťka, průměr, atd.)

3.5.1.2 Volitelné (doporučené) parametry

Použití volných parametrů závisí pouze na administrátorovi. Může zvolit prakticky kteroukoli kategorii. Nicméně pro multiuživatelský systém pro podporu znovuvyužití především hutního materiálu lze doporučit následující kategorie:

Jakost – stupeň splnění požadavků souborem inherentních znaků, tj. soubor trvalých znaků produktu (definice ISO 9000) [14]; jedná se o jednoznačnou identifikaci produktu, která usnadňuje rozhodování o vhodnosti použití daného materiálu ke konkrétnímu procesu znovuvyužití

Množství – určení počtu kusů nabízeného, poptávaného zboží

Cena – určení ceny zboží, ať už celkové nebo za kus

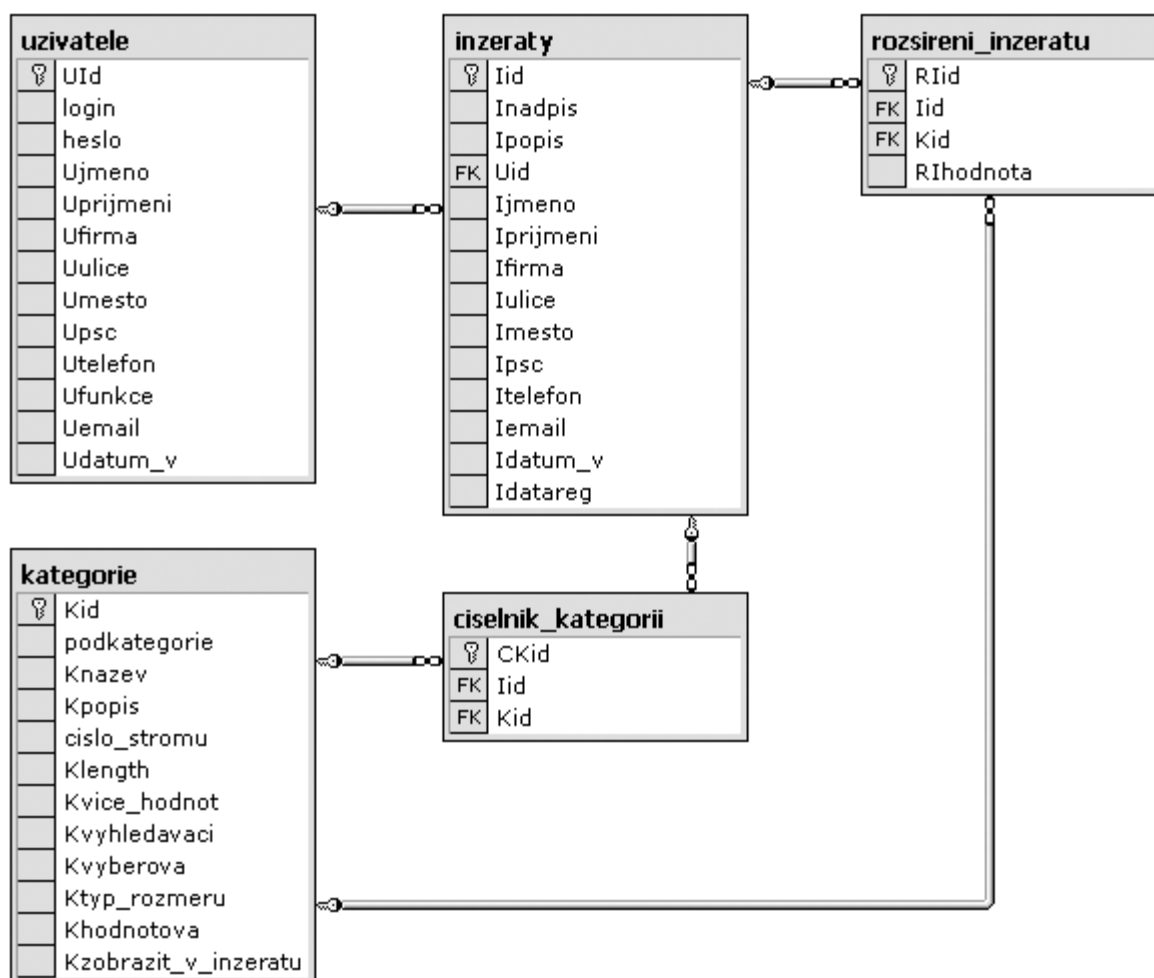
Hmotnost – určení hmotnosti zboží, ať už celkové nebo za kus; může to posloužit v případě, kdy nám jde především o informace kvantitativního charakteru nebo když nelze vyjádřit množství

Nedílnou součástí každého vyhledávání je samozřejmě hledání v databázi inzerátů na základě uživatelem zadaného slova nebo slovního spojení (fulltextové vyhledávání). Po zadání slova se zjišťuje zda se daný řetězec nachází v databázi a poté se vypíše všechny inzeráty, které tento hledaný řetězec obsahují.

4 Implementace

4.1 ER diagram

Důležitou součástí návrhu celého systému je ER diagram (obr. 4.1.1), který definuje tabulky a tím i jejich vztahy. Diagram je tvořen 5 tabulkami. Pro přehlednost nejsou uvedeny typy sloupců a obsah tabulek je rozebrán podrobně v další kapitole.



Obrázek 4.1.1

Tabulkami v diagramu jsou *uzivatele*, *inzeraty*, *kategorie*, *rozsireni_inzeratu* a *ciselnik_kategorii*.

Tabulka *uzivatele* podává informace o uživateli. Každý uživatel může mít až N vložených inzerátů do systému. Tato tabulka také obsahuje sloupec *Ufunkce*, který nám říká, zda se jedná o administrátora nebo běžného uživatele. Vztah mezi entitami uživatel,

kontaktní údaje a inzerát je takový, že každý uživatel má uloženy osobní kontaktní údaje, které by měly být unikátní, kdežto kontaktní údaje uvedené u inzerátu být unikátní nemusí. *Ciselnik_kategorie* obsahuje data o tom, do které kategorie daný inzerát patří. Každý inzerát může obsahovat jedno nebo více rozšíření inzerátu. Poslední zakreslený vztah je mezi *kategorie* a *rozsireni_inzeratu* – naznačuje, že kategorie může být rovna položce rozšíření inzerátu.

4.2 Tabulky a jejich význam

Každá z definovaných tabulek má svůj specifický význam, který si nyní popíšeme.

4.2.1 Uživatelé

Tabulka *uzivatele* (viz. tabulka 4.2.1) obsahuje podrobné informace o uživateli, který je registrovaný v inzertním systému. Povinné údaje, které musí být uvedeny v každém záznamu této tabulky, jsou *login*, *heslo*, *Uemail* a *Ufunkce*.

Název sloupce	Typ
Uid	int(6)
login	varchar(10)
heslo	varchar(15)
Ujmeno	varchar(20)
Uprijmeni	varchar(20)
Ufirma	varchar(40)
Uulice	varchar(40)
Umesto	varchar(40)
Upsc	varchar(5)
Utelefon	varchar(20)
Ufunkce	varchar(5)
Uemail	varchar(40)
Udatum_v	datetime

Tabulka 4.2.1

- **Uid** – primární klíč, autoincrement
- **login** – unikátní (přihlašovací jméno)
- **heslo** – délka minimálně 5 znaků
- **Ujmeno, Uprijmeni, Ufirma, Uulice, Umesto, Upsc, Utelefon** – osobní údaje uživatele. Nepovinné.

- *Ufunkce* – určuje jaké akce jsou danému uživateli povoleny. Na výběr jsou dvě role – *uživatel* a *admin*.
- *Uemail* – unikátní a s jednoznačně daným formátem. Povinný.
- *Udatum_v* – unikátní, datum uložení záznamu o uživateli do tabulky ve formátu (dd.mm.yyyy hh:mm:ss)

4.2.2 Kategorie

Tabulka *kategorie* (viz tabulka 4.2.2) obsahuje informace o všech kategoriích v inzertním systému.

Každá má nadřazenou kategorii vyjma té první každé stromové struktury. Jako další informaci můžeme přidat popisek neboli informace o dané sekci inzerátů. Neméně podstatnou položkou je *cislo_stromu*, které určuje do jaké stromové struktury daná kategorie patří. Posledních sedm sloupců si popíšeme podrobněji později.

Název sloupce	Typ
Kid	int(6)
podkategorie	int(6)
Knazev	varchar(40)
Kpopis	text
cislo_stromu	char(3)
Klength	int(3)
Kvice_hodnot	char(3)
Kvyhledavaci	int(3)
Kvyberova	char(3)
Ktyp_rozmeru	varchar(5)
Khodnotova	char(3)
Kzobrazit_v_inzeratu	int(3)

Tabulka 3.2.2

- *Kid* – primární klíč, autoincrement
- *podkategorie* – index nadřazené kategorie. Pokud se jedná o kořenovou kategorii, pak je index roven 0.
- *Knazev* – název kategorie; povinný
- *Kpopis* – stručná charakteristika dané kategorie. Nepovinný.
- *cislo_stromu* – označení do které stromové struktury kategorie patří. První strom označujeme jako hlavní, ostatní stromy jako vedlejší
- *Klength* – určuje jaká je max. délka řetězce, kterou lze zadat při vyplňování položky rozšíření inzerátu jejíž jméno je rovno názvu kategorie

- ***Kvice_hodnot*** – určuje, zda je možné pro danou kategorii (dále položku) uložit více hodnot. Implicitně nastaveno na hodnotu „ne“. Pokud je nastaveno na „Ano“, pak se objeví u této položky, u formuláře pro vložení inzerátu v části rozšíření inzerátu, nabídka pro výběr počtu zadávaných hodnot.
- ***Kvyhledavaci*** – určuje, zda se daná kategorie objeví ve vyhledávání. To, jak se zobrazí ve vyhledávání, závisí také na dalších parametrech kategorie a také do jaké stromové struktury kategorie patří. Implicitně nastaveno na „ne“ což odpovídá hodnotě 0. Pokud nastavíme „Ano“, pak je přiřazena kategorii číselná hodnota označující, na kterém místě ve vyhledávání se tato kategorie zobrazí. Pozice pro zobrazení se dá následně přenastavit v administraci pod nabídkou Nastavené kat. pro vyhledávání.
- ***Kvyberova*** – určuje, zda se bude položka zobrazovat jako typ ovládacího prvku `<select></select>` s přímými podkategoriemi jako `<option></option>` či nikoli. Implicitně nastaveno na hodnotu „ne“.
- ***Ktyp_rozmeru*** – nastavuje typ rozměru, který se bude brát v úvahu pro inzeráty v této kategorii. Stěžejní se stává informace při vyhledávání, kdy se podle vybrané kategorie z hlavního stromu zobrazí nabídka parametrem určených rozměrů. Mezi typy rozměru patří:
 - *AxBxC* – implicitní typ rozměru, což znamená *tloušťka x šířka x délka*.
 - *TxSxD* – označuje rozměr *tloušťka x šířka x délka*. Obvykle se tento typ uvádí u např. plechů, pásů, tyče čtvercové a ploché.
 - *PxTxD* – označuje rozměr *průměr x tloušťka x délka*. Obvykle se tento typ uvádí u trubek
 - *PxD* – označuje rozměr *průměr x délka*. Obvykle se tento typ uvádí u tyčí kruhového průřezu
- ***Khodnotova*** – určuje, zda se bude položka zobrazovat jako typ ovládacího prvku `<input type=text .../>` či nikoli. Implicitně nastaveno na hodnotu „ne“.
- ***Kzobrazit_v_inzeratu*** – určuje, zda se kategorie bude zobrazovat jako položka ve formuláři pro vložení inzerátu. Implicitně nastaveno na „ne“, což odpovídá přidělené hodnotě 0. Pokud je zvoleno „Ano“, pak se přidělí číselná hodnota (=unikátní, nenulová).

4.2.3 Rozšíření inzerátu

Tabulka *rozsireni_inzeratu* (viz tabulka 4.2.3) obsahuje informace o položkách rozšíření inzerátu. Kromě již zmíněné položky rozšíření *Typ inzerátu* (viz. odst. 3.3.1.1) není

podmínkou, aby inzerát obsahoval jakékoliv rozšíření. Avšak pokud má uživatel k dispozici informace o zboží, které se shodují s nějakou položkou rozšíření, je vhodné ji vyplnit. Pomáhá to konkretizovat nabídku či poptávku daného zboží a tak má uživatel, který vyhledává v multiuživatelském systému, přesný pohled na daný inzerát.

Název sloupce	Typ
RIid	int(6)
Iid	int(6)
Kid	int(6)
RIhodnota	varchar(40)

Tabulka 3.2.3

- **RIid** – primární klíč, autoincrement
- **Iid** – index inzerátu, kterého se položka rozšíření týká
- **Kid** – index kategorie odpovídající položky rozšíření
- **RIhodnota** – uživatelem zadaná hodnota pro položku rozšíření

4.2.4 Inzeráty

Tabulka *inzeraty* (viz tabulka 4.2.4) obsahuje veškeré informace o vloženém inzerátu.

Název sloupce	Typ
Iid	int(6)
Inadpis	varchar(60)
Ipopis	varchar(200)
Uid	int(6)
Ijmeno	varchar(20)
Iprijmeni	varchar(20)
Ifirma	varchar(40)
Iulice	varchar(40)
Imesto	varchar(40)
Ipsc	varchar(5)
Itelefon	varchar(20)
Iemail	varchar(40)
Idatum_v	datetime
Idatareg	char(3)

Tabulka 3.2.4

- **Iid** – primární klíč, autoincrement
- **Inadpis** – název inzerátu. Povinný.
- **Ipopis** – upřesňující informace o zboží, které je předmětem inzerátu. Povinný.

- **Uid** – identifikační číslo uživatele, který daný inzerát vložil – vlastník.
- **Ijmeno, Iprijmeni, Ifirma, Iulice, Imesto, Ipsc, Itelefon, Iemail** – jsou kontaktní údaje na osobu nebo firmu, která daný inzerát zastupuje, tyto data se mohou (ale nemusí) shodovat s osobními údaji uživatele, který inzerát vložil.
- **Idatum_v** – unikátní, datum uložení záznamu o inzerátu do tabulky ve formátu (dd.mm.yyyy hh:mm:ss)
- **Idatareg** – určuje, zda se kontaktní údaje shodují s kontaktními údaji uživatele, který inzerát zadal

4.2.5 Číselník kategorií

Tabulka *ciselnik_kategorii* (viz tabulka 4.2.5) obsahuje informace o tom do jaké kategorie byl inzerát uložen resp. jaké inzeráty daná kategorie obsahuje.

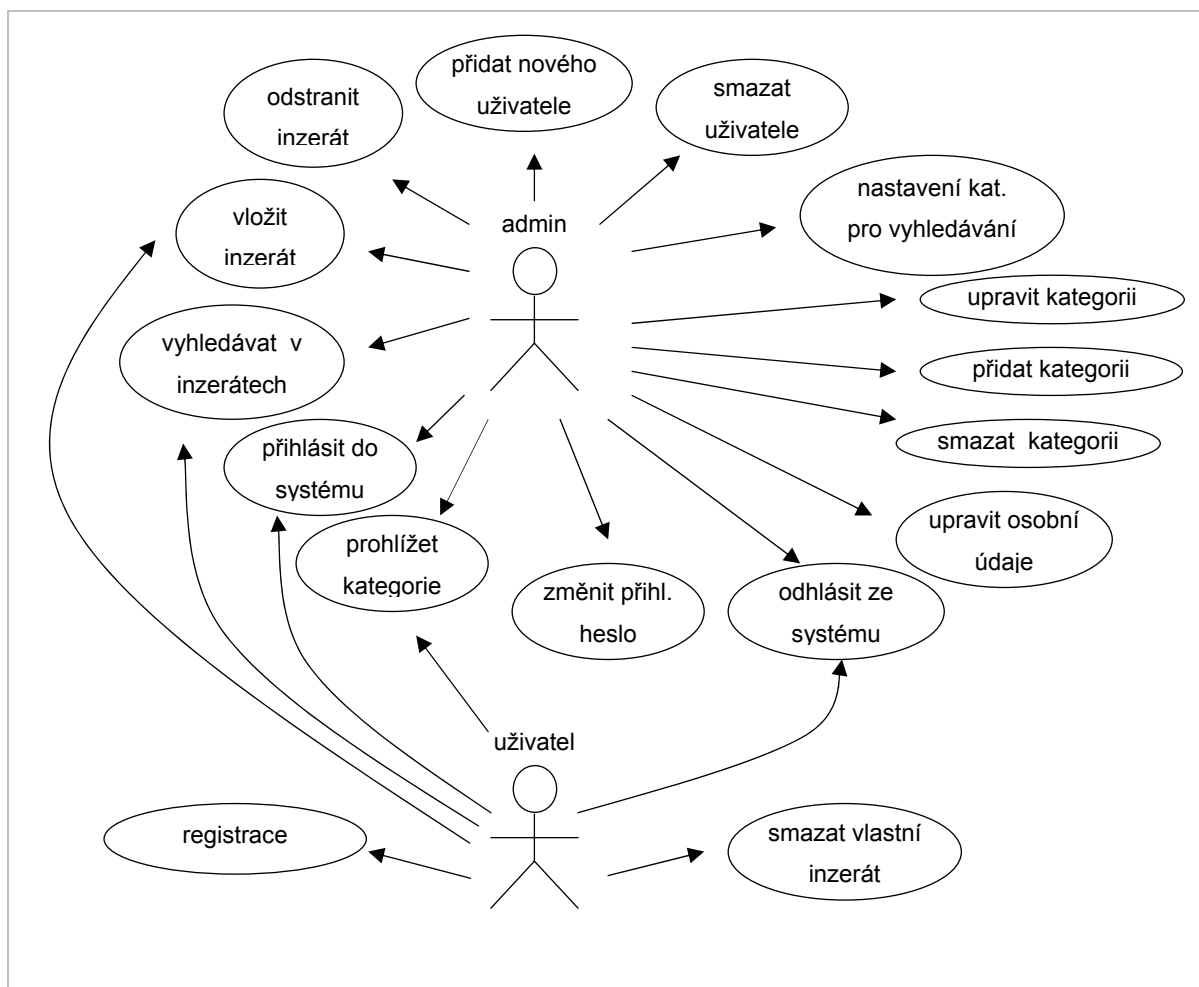
Název sloupce	Typ
CKid	int(6)
Iid	int(6)
Kid	int(6)

Tabulka 3.2.5

- **CKid** – primární klíč, autoincrement
- **Iid** – index inzerátu vložený do zvolené kategorie
- **Kid** – index kategorie, do které byl vložen inzerát

4.3 Diagram užití

Diagram užití má názorně a efektivně zobrazovat akce, které jednotliví uživatelé mohou provádět. Jelikož do systému vstupují uživatelé dvojího typu, jsou v diagramu (obr. 4.3.1) znázorněny oba. Prvním z nich je administrátor (admin), který může provádět téměř všechny hlavní akce jako běžný uživatel. Navíc může přidat a smazat nového uživatele, přidávat, editovat a smazat jednotlivé kategorie a odstranit libovolný inzerát ze systému. Běžnému uživateli je povolena registrace, přihlášení a odhlášení z/do systému. Drobná odlišnost oproti administrátorovi je v mazání inzerátů, kdy běžný uživatel může smazat pouze inzeráty, které do systému vložil.



Obrázek 4.3.1

4.4 Struktura zdrojových kódů

Z programátorského hlediska je projekt tvořen sadou PHP skriptů, které jsou umístěny v jednom adresáři. Ten obsahuje podadresář images s použitými obrázky, dále se v hlavním adresáři vyskytuje soubor s definicemi kaskádových CSS stylů a soubory obsahujícími funkce JavaScriptu. Mezi PHP skripty jsou jednak soubory, které určují samotné stránky, a jednak doplňkové skripty, které obsahují strukturu HTML dokumentu, nebo inicializaci spojení s databází a různé použité funkce. Díky těmto skriptům je změna základní struktury HTML dokumentu, menu nebo přihlašovacích údajů k databázi pouze otázkou změny jednoho konkrétního PHP souboru.

4.4.1 Hlavní stránka systému

4.4.1.1 Seznámení se session v PHP

Přihlašování do systému je realizováno prostřednictvím PHP sessions [15]. Session je k podobným účelům v jazyku PHP implementována. Usnadňují také definici uživatelských účtů a kontrolu přístupových práv jednotlivých skupin uživatelů.

Použití v praxi

Session neboli relace umožňuje přesnou identifikaci uživatele a pohyb na serveru. Za pomoci session jsou naprogramovány všechny web-based mailové servery. Má tedy použití všude tam, kde je zapotřebí monitorovat nebo ověřovat uživatele, například pomocí jeho IP adresy. Používá se také pro možnost vlastní úpravy vzhledu (customizing) např. na vyhledávacích nebo katalogizačních serverech, kdy po vstupu na stránku server ověří vaši identitu a zobrazí informace, novinky nebo zprávy přesně podle vašeho dřívějšího nastavení. Nutno podotknout, že session jsou implementovány do PHP až od verze 4.0.

V našem případě je tedy využití session ideální, jelikož potřebujeme ověřovat uživatele a monitorovat jeho určité akce např. při vyhledávání. Tehdy je zapotřebí sledovat jeho volby vyhledávání.

Princip session

Session pracuje na jednoduchém principu. Každý uživatel, který vstoupí na stránku obdrží jednoznačné číslo, budeme ho nazývat ID, resp. SID (Session ID). Často to nebývá pouze číslo, ale vyskytují se v něm i znaky, např. a3c54cd96d712021d54587d5a4b4c. Tímto číslem se tedy bude uživatel identifikovat, tj. ve všech odkazech je zapotřebí předávat tento parametr. PHP implicitně předává SID v proměnné PHPSESSID. Viz. sekce [Session] a položka session.name v souboru php.ini. Ukládání čísla relace v URL je jeden ze způsobů,

PHP nabízí kromě toho také ukládání do cookie. Session demonstruje svou sílu především možností předávání de facto neomezeného počtu proměnných. Na stránkách tedy nemusíme používat tagy typu:

```
<input type="hidden" name="promenna" value="hodnota">
```

Veškerý parametr, který je předáván je pouze SID. Pokud uživatel navštíví stránku, kde používáte session, PHP automaticky (v případě, že je nastavena v souboru php.ini položka `session.auto_start =1`) otestuje existenci relace. Pokud neexistuje jsou všechny proměnné a relace vytvořeny. V našem případě jsme toto provedli programově a to pomocí funkce `session_start()` hned na prvním řádku hlavní stránky.

Každá relace má svou životnost. Relace končí, pokud uživatel opustí korektně stránku (například se odhlásí), nebo vyprší její časový limit. Na serveru musí běžet démon, který kontroluje aktivní relace a maže již neaktivní. Relace, které nejsou již aktivní se nazývají garbage (smetí).

4.4.1.2 Struktura hlavní stránky

Hlavní struktura HTML stránek je vytvořené zejména pomocí oddílů DIV. Formátování těchto oddílů pomocí stylů CSS je velmi snadné a v poslední době je tento přístup v řadě případů upřednostňován.

Hlavní stránku systému tvoří skript `index.php`, což je základní stavební kámen každé stránky. Při otevření této stránky nejprve načteme pomocí příkazu `require "../func.php"`; skript obsahující hlavní funkce systému. Následně se skript pokusí přihlásit k databázi, po úspěšném připojení zjišťuje, jestli se uživatel pokusil přihlásit. Pokud ano, ověří jeho login a heslo a zahájí sezení (session) a zobrazí jméno a příjmení na levé straně v záhlaví.

Menu

V levé části stránky je umístěno navigační menu stromové struktury. Navigační menu je vytvořeno pomocí JavaScriptu, jehož funkce jsou uloženy v souboru `DynamicTree.js`, který je načten v hlavičce indexového souboru. Tyto funkce umožňující vytvořit stromovou strukturu, které využívá cookies pro ukládání informace o stavu složek a podsložek menu tzn. zda je složka otevřená („rozkliknutá“) či zavřená. Definice vzhledu navigačního menu nalezneme v souboru pro kaskádové styly CSS s názvem `DynamicTree.css`, ve kterém jsou zároveň uloženy všechny ostatní definice vzhledu celého systému. Tento script je volně ke stažení na internetu [16]. Obsahuje dva scriptové soubory `DynamicTree.js`, `DynamicTree.css`.

Jelikož jsou všechny stromové struktury s kategoriemi generovány z databáze, volá se funkce `menu()` ;

Rekurzivní funkce, která zobrazí menu generovaného z databáze

V db nejprve vyhledá v tabulce kategorie všechny kořenové kategorie hlavních resp. vedlejších stromových struktur, tzn. ty, které nejsou ničí podkategorií (hodnota ve sloupci *podkategorie* = '0'). Poté se rekurzivně zanořuje a vypisuje podkategorie až po tzv. listové kategorie stromů.

Zobrazení stránky

Po načtení navigačního menu se následně volá funkce `ukazstranku()` ;

Ověřuje zda stránka, kterou chce uživatel zobrazit, existuje. Pokud neexistuje, přesměruje skript uživatele na stránku *notfound.php*, kde je vypsáno oznámení o nenalezení stránky. V opačném případě načte požadovanou stránku.

Výhodou takového zobrazení je fakt, že se stránka načte do rámu mezi tagy `<div></div>` a tak se nám nemusí při každém přechodu na další stránku načítat celá hlavička, patička, menu a grafika.

4.4.2 Registrace uživatele

Stránka *reg.php* obsahuje formulář pro registraci nového uživatele (obr. 4.4.2.1). Po jeho vyplnění a odeslání ho zpracuje interní skript. Ten po připojení k databázi porovná údaj emailu zaslaný metodou POST s údajem v databázi, zároveň provede ověření správnosti zadání povinných údajů a pokud nedošlo k chybě, uloží relační proměnné s hodnotami loginu, hesla, jména, příjmení, emailu a dalších vyplněných položek. Nakonec se zobrazí patřičné oznámení, že došlo k úspěšné registraci.

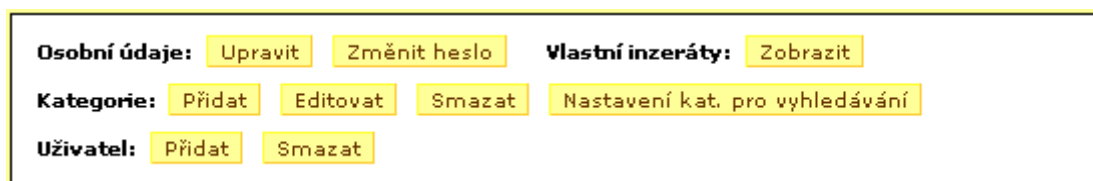
4.4.3 Přihlášení a odhlášení uživatele

Po úspěšné registraci může uživatel přistoupit k přihlášení do multiuživatelského systému pomocí přihlašovacího formuláře na stránce s názvem *prihlaseni.php*. Ten obsahuje položky *login* a *heslo*. Po vyplnění obou údajů a odeslání se interní skript připojí k databázi a ověří, jestli se zadaný *login* shoduje s nějakou hodnotou z tabulky *uzivatele*. Pokud ano, pak ještě porovná příslušné *heslo*, se již zadaným. Proběhne-li vše v pořádku, uživatel je přihlášen, což se projeví zobrazení jména a příjmení v levé části záhlaví, do systému. Klikne-li v menu na odkaz s názvem *Odhlášení*, je mu zobrazena stránka *odhlaseni.php* s oznámením o úspěšném odhlášení ze systému. Tím je sezení (session) daného uživatele ukončena.

4.4.4 Administrace

Každý přihlášený uživatel má v menu odkaz Administrace. Podle toho, jestli se jedná o administrátora nebo běžného uživatele, se zobrazí položky akcí, které lze provádět. Administrátor může v tomto multiuživatelském systému provádět, mimo akcí běžného

uživatele, následující akce (obr. 4.4.4.1). Pouze akce na prvním řádku může provádět i běžný uživatel. Operace týkající se kategorií a uživatelů smí vykonávat pouze uživatel patřící do skupiny admin. S nimi jsou spojeny následující stránky.



Obrázek 4.4.4.1

4.4.4.1 Přidat uživatele – `add_user.php`

Stránka přidání uživatele je téměř totožná se stránkou registrace nového uživatele *reg.php* (viz. kapitola 4.4.2) s tím rozdílem, že v tomto formuláři přibyla rolovací nabídka pro sloupec *funkce* tabulky *uzivatele*. To proto, aby administrátor mohl vybrat, zda vkládá běžného uživatele nebo sobě rovného admina.

4.4.4.2 Smazat uživatele – `del_user.php`

Stránka smazání uživatele nabízí formulář pro vyhledání hledaného(-ných) uživatele(-ů) (obr. 4.4.4.2.1). Obsahuje jeden prvek textového pole pro zadání informace, podle které chceme uživatele pro smazání hledat. A dále prvek pro výběr sloupce z tabulky *uzivatele*, ve které se bude vyhledávat. Výsledek hledání se nám zobrazí pomocí tří resp. čtyř sloupců tabulky. Pokaždé se nám vypíše sloupec *Uid*, *příjmení* a *jméno*. Pokud hledáme výraz z odlišného sloupce než jsou tyto tři, pak se vypíše ještě i tento. Z výsledku poté pomocí checkboxu zaškrtneme ty, které chceme smazat a potvrdíme to příslušným tlačítkem.

Poprvé se zde setkáváme s použitím tzv. souladu (shodou) se vzorem. Jedná se o podmínku, která je užitečná tehdy, potřebujeme-li nalézt hodnoty podobné zadanému řetězci vzoru nebo chceme-li jako kritérium použít jen částečnou informaci [17].

Podmínka přebírá hodnotu výrazu a pomocí predikátu LIKE (podobný) testuje, zda daná hodnota odpovídá definovanému vzorovému řetězci. Řetězec vzoru může být tvořen libovolnou logickou kombinací normálních znaků řetězců a dvou speciálních zástupných znaků:

znak procent (%) – Tento zástupný znak představuje nulový nebo větší počet libovolných normálních znaků.

znak podtržení (_) – Tento zástupný znak představuje libovolný jediný regulérní znak.

Způsob definování řetězce vzoru určuje, které hodnoty se převezmou z hodnotového výrazu.

V našem případě jsme použili vzor řetězce `,%hledaný_řetězec%`, který říká, že znakový řetězec může mít libovolnou délku, ale musí obsahovat „hledaný_řetězec“.

Obrázek 4.4.4.2.1

4.4.4.3 Přidat kategorii – insert_ka.php

Zde je nejprve potřeba vybrat nadřazenou kategorii pro vkládanou kategorii. Položky tohoto výběru jsou načteny pomocí rekurzivní funkce `fseznam_kat()` při níž dochází k postupnému zanořování do struktur kategorií a následný výpis. Na tomto výběru je použita událost JavaScriptu `onblur`, která se vyvolá v okamžiku, kdy uživatel provede výběr kategorie. Provede se volání metody `form.submit()`. To se používá pro odesílání dat z formuláře namísto odesílacího tlačítka. Ovšem neplést si to s událostí `onsubmit`, která narozdíl od předchozí může zrušit vlastní odesílání dat v případě, že výsledek obsluhy události bude `false`. Čehož se využívá ve skriptech, které kontrolují kompletnost a správnost dat před vlastním odesláním (nedopadne-li kontrola správně, vrátí se hodnota `false` a uživatel musí obsah formuláře dopsat, případně poopravit). V tomto případě by bylo použití události `onsubmit` zbytečné.

Po odeslání se zobrazí formulář pro vyplnění názvu kategorie a jejího popisku. Skript se poté připojí k databázi a ověřuje, zda se zadaný název v tabulce nevyskytuje. Pokud ano, vypíše upozornění o nutnosti zadat nový název. Následují položky (dále parametry), jejichž hodnotu nastavujeme pomocí ovládacího prvku `radiobutton` na hodnotu „Ano“ nebo „Ne“ nebo zadáním do textového pole a jsou pro všechny kategorie stejné:

- Kategorie pro vyhledávání
- Kategorie pro výběr
- Velikost buňky

Další zadávané parametry kategorie závisí na tom, je-li nadřazená kategorie vybrána z hlavní nebo vedlejší stromové struktury. U hlavního stromu můžeme nastavit tento parametr:

- Typ rozměrů kategorie
- Mezi nastavitelné parametry vedlejšího stromu patří:
- Zobrazit v rozšíření inzerátu
- Přidávání více hodnot
- Kategorie pro zadání hodnoty

4.4.4.4 Editovat kategorii – edit_kat.php

Upravením kategorie je myšleno, možnost změnit jakýkoliv parametr. Nejprve se musí vybrat kategorie kterou má být editována. Stejně jako u přidání kategorie i zde se pomocí funkce `fseznam_kat()` naplní ovládací prvek pro výběr. Po odeslání změněných dat provede skript kontrolu připojením k databázi a zjištěním, zda není nový název v příslušné tabulce databáze již obsažen. Opět je to ošetřeno vypsáním informace o úspěchu či neúspěchu akce.

4.4.4.5 Smazat kategorii – delete_kat.php.

Nejprve se musí vybrat kategorie, která má být smazána. Stejně jako u přidání kategorie i zde se pomocí funkce `fseznam_kat()` naplní ovládací prvek pro výběr. Při pokusu o smazání kategorie se skript připojí k databázi a zjistí zda má zvolená kategorie vlastní podkategorie a zda neobsahuje inzeráty. Pokud tomu tak není, je smazána. V opačném případě je vypsáno upozornění, že ji nelze smazat z důvodu existujících podkategorií nebo inzerátů.

4.4.4.6 Nastavení kategorií pro vyhledávání – set_searchin.php

Na této stránce můžeme nastavit pořadí kategorií, v jakém se budou zobrazovat na vyhledávací stránce. Jak již bylo popsáno v kapitole Vyhledávání, je zde několik kategorií, jejichž pozice by měli být neměnné (obr. 4.4.4.6.1). Jedná se o kategorie *Materiály*, *Typ inzerátu*, *Tloušťka*, *Šířka*, *Délka* a *Průměr*. Umístění ostatních může být libovolné.

Pro zobrazení tohoto seznamu kategorií pro vyhledávání je volána funkce `vypis_vyhledavaci_kat()`. Ta prochází tabulku kategorie a vybírá ty, jenž mají hodnotu parametru *Kvyhledavaci* rozdílnou od nuly (což znamená nastaven parametr na „Ano“). Vzestupně podle hodnoty je pak vypíše do formuláře.

Pořadí kategorií pro vyhledávání	
Materiály	<input type="text" value="1"/>
Typ inzerátu	<input type="text" value="2"/>
Tloušťka	<input type="text" value="3"/>
Šířka	<input type="text" value="4"/>
Délka	<input type="text" value="5"/>
Průměr	<input type="text" value="6"/>
Jakost	<input type="text" value="7"/>

pozn. kategorie řazené odshora dolů (prvních 6 by mělo být neměnných)

Obrázek 4.4.4.6.1

4.4.4.7 Osobní údaje – `edit_my_details.php`

Stránka `edit_my_details.php` má dvě části. V první lze upravovat veškeré osobní údaje kromě hesla. To je umožněno v druhém formuláři pro změnu hesla. K tomu abychom změnili heslo, potřebujeme zadat heslo staré a poté opakovat zadání nově zvoleného. Po odeslání dat dochází k sérii ověření, zda byly údaje zapsány korektně. Jakákoliv nekorektnost je vypsána do formuláře příslušnou chybovou hláškou.

4.4.4.8 Zobrazení vlastních inzerátů – `show_my_in.php`

Aby měl uživatel přehled o svých vložených inzerátech, je součástí menu administrace taky položka pro zobrazení vlastních inzerátů. Zde se poprvé objevuje volání funkce `vypis_i()`.

Podrobnější popis této funkce bude popsán později.

4.4.5 Inzerát

V této kapitole si rozebereme vkládání inzerátu a poté zobrazování inzerátu v multiuživatelském systému.

4.4.5.1 Vložení inzerátu

Skript pro vložení nového inzerátu nese název `insert_in.php`. Všechny části tohoto formuláře, kromě jedné, jsou pevně dané. Část rozšíření inzerátu je totiž celá generována z databáze.

Základní údaje

Nejprve musíme vybrat kategorii pro inzerát. Jednotlivé položky výběru jsou vygenerovány pomocí rekurzivní funkce `cesta_kat()`, která negeneruje pouze názvy jednotlivých kategorií, ale ukládá celou cestu. Pro uživatele je to přehlednější, když má možnost vidět přesné umístění kategorií. Další vyplnitelnou a zároveň povinnou položkou je nadpis a popis inzerátu.

Osobní údaje

V této části formuláře máme na výběr, zda vyplníme nové údaje o majiteli nabízeného zboží nebo použijeme osobní údaje přihlášeného uživatele. K tomu nám poslouží `checkbox` pro použití osobních údajů z registrace. Po odeslání dat z formuláře se poté kontroluje zaškrtnutí, podle kterého se rozhodne, zda se mají načíst osobní údaje zadavatele inzerátu a nebo použít nové uživatelem zadané informace o kontaktu.

Rozšíření inzerátu

Jak již bylo napsáno, celá tato část je generována. Jednotlivé položky v podstatě odpovídají kategoriím vedlejších stromů (stromových struktur). Jaké kategorie se zobrazí v rozšíření,

závisí na parametru *Kzobrazit_v_inzeratu*. Dotazem na databázi jsou vyhledány všechny kategorie vedlejších stromových struktur s nenulovou hodnotou tohoto parametru a zahrnuty do výběru pro zobrazení. Tímto výběrem máme zajištěno pouze vypsání názvu položky rozšíření nikoli však, jako uživatel, možnost zvolit hodnotu položky.

Způsob zobrazení nám totiž zajišťují další dva resp tři parametry kategorie. A to *Kvyberova*, *Khodnotova* a k druhé uvedené doplňková *Kvice_hodnot*. Ve skriptu se ověří, zda je *Khodnotova*='Ano', pokud dostaneme true, zobrazí se u dané položky vstupní pole typu text. Doplňková hodnota parametru *Kvice_hodnot*='Ano' určí, že je možné zadat více hodnot pro danou položku. Z obr. 3.3.2.1 je vidět, že si uživatel zvolí, pomocí výběru, počet hodnot, které chce zadat. V tomto výběrovém prvku je prostřednictvím JavaScriptu volána událost *onchange*, která při každé změně hodnoty prvku volá funkci *testSelect()*. Funkce testuje jaká hodnota byla vybrána a tudíž kolik vstupních polí typu text má zobrazit. Samotné zobrazení má na starosti funkce *echoDisplay()*, která nadefinuje atribut *style="display:block"* nebo *style="display:none"*. Abychom předešli zobrazování obecné velikosti(*size*) a délky(*maxlength*) vstupního pole, je u kategorií ještě jeden parametr, pomocí kterého nastavíme námi požadovanou velikost resp. délku. Jedná se o *Klength*. Zobrazení prvku výběru u položky rozšíření je podmíněno hodnotou parametru *Kvyberova*="Ano" a zároveň *Khodnotova*="Ne". Za této situace je zavolána funkce *vypis_vyberove_kat()* vypisující přímé podkategorie položky do výběru.

Po vyplnění a odeslání formuláře provádí interní skript ověření správnosti zadání povinných údajů. Prvním z nich je výběr *kategorie*, další jsou *nadpis*, *popis* a *email*. *Email* se ale kontroluje pouze tehdy, pokud nebylo zaškrtnuto tlačítko použít osobní údaje z registrace. Následně se skript připojí k databázi a uloží všechny uživatelem zadané údaje do tabulky *inzerat*.

4.4.5.2 Zobrazení inzerátů

Inzeráty se nám vypisují na více místech multiuživatelského systému a způsob se příliš nemění. Proto nyní rozebereme zobrazování především na stránce *display.php*.

Pokud si uživatel prochází systém, přesněji kategorie, pomocí menu, vždy se mu zobrazí stránka s přímými podkategoriemi a inzeráty spadající do této kategorie a jejich všech podkategoriích.

Cesta kategorie s počtem inzerátů

Aktuální cesta našeho procházení kategoriemi se nám pokaždé zobrazuje v horním řádku stránky. Tento výpis nám zajišťuje funkce *ukaz()*. Abychom věděli kolik inzerátů se v aktuálním vybraném podstromu vyskytuje, je volána funkce *zobraz_inzeraty()*, která zjistí počet inzerátů zvolené kategorie a jejich podkategorií a hned za výpis cesty výpíše

jejich počet. U této funkce máme čtyři vstupní parametry. Vstupními parametry jsou *zobraz_in* (identifikační číslo zvolené kategorie), *strom* (číslo stromu zvolené kategorie) a *inzeraty* (pole inzerátů, které byly nalezeny v již prohledaných podkategoriích). Způsob vyhledávání inzerátu je závislý na parametru čísla stromu, protože hlavní i vedlejší stromová struktura je tvořena odlišně a i ukládání inzerátu do nich se liší. Nalezené inzeráty se ukládají do proměnné *inzeraty*, v podobě identifikačních čísel, která se později využívají k samotnému výpisu inzerátu.

Výpis podkategorií a inzerátů

Pro vypsání přímých podkategorií slouží triviální funkce `zobraz_pod()` s parametrem aktuálně vybrané kategorie. Zobrazí se, pomocí kaskádových stylů, jako složka s názvem a případným popisem. Pod tímto výpisem už následuje vypsání všech inzerátů patřících do podstromu s vybranou kategorií jako kořenovou.

V souvislosti se zahrnutím stromu pro *typ inzerátu* do vedlejší stromové struktury a tím pádem i do rozšíření inzerátu byla nutnost identifikovat mezi rozšířeními inzerátu právě položku *typ inzerátu*, jelikož se jedná o povinný údaj. Vyskytuje-li se v kategorii inzerát, pak musí být v tabulce *rozsireni_inzeratu* záznam o typu inzerátu. Ten může spadat do podkategorií *prodej*, *nákup*, *výměna* a *jiné*. Proto je spuštěn SQL dotaz na určení, v kterých z těchto podkategorií je nějaký inzerát. Seznam těchto, maximálně čtyř, kategorií je následně použit jako druhý vstupní parametr pro funkci na vypsání inzerátu (viz níže). Pokud tedy pole proměnné *inzeraty* není prázdné, provede se setřídění podle identifikačního čísla inzerátu funkcí `RSort()`. A poté je pro každý prvek pole volána funkce `vypis_i()`.

Funkce `vypis_i()` má dva vstupní parametry. Prvním z nich je hodnota prvku pole *inzeraty* a druhým již zmíněný výsledný seznam kategorií předchozího SQL dotazu.

V úvodu funkce se zjišťuje, zda je uživatel přihlášený a zároveň vlastníkem inzerátu. V důsledku toho se případně objeví v hlavičce inzerátu ikona odpadkového koše s odkazem, pro možnost smazání vlastního inzerátu. Po kliknutí na tento odkaz se zobrazí stránka *delete_in.php*. Ta vypíše vybraný inzerát s dotazem, jestli chce uživatel provést smazání. Pokud ano, skript přistoupí k databázi a smaže údaje jak z tabulky *inzerat* tak i z tabulky *ciselnik_kategorii* a vypíše oznámení, že došlo ke smazání. V opačném případě je vypsána hláška o zachování inzerátu.

Následně se pro každý prvek druhého parametru prověřuje, zda inzerát spadá do této kategorie. Pokud ano, ukončí se prověřování a přejde se na vypsání hlavičky inzerátu. Pro vypsání kontaktních údajů a informací z rozšíření se volají funkce `exist_data_in()` a `exist_roz_in()`. V obou funkcích jde v podstatě jen o způsob vypsání údajů podle dostupných dat.

4.4.6 Stránka vyhledávání

První soubor, jehož načtení prohlížeč vyžaduje, je *index.php*. Ten zajistí zobrazení úvodní stránky umožňující uživateli vyhledávání, jedná se o stránku *searchin.php*.

Na stránce se mohou objevit dva typy nabídek pro zadání parametrů vyhledávání. První z nich je výběr ze seznamu a druhý vstupní textové pole.

4.4.6.1 Výběr ze seznamu

Na úvod si SQL dotazem zjistíme počet kategorií, které se mají objevit ve vyhledávání jako výběr ze seznamu (parametry kategorie *Kvyhledavaci*, *Kvyberova* nastaveny na „Ano“). Vytvoříme příslušný počet session proměnných, do kterých při každém odeslání formuláře ukládáme hodnoty z použitých výběrů.

Zobrazení seznamů

Jak již bylo popsáno v kapitole 3.5, tak mezi povinné položky vyhledávání a s tím spojené kategorie systému jsou: Materiály (kořenová kategorie hlavní stromové struktury), Typ inzerátu (s podkategoriemi Prodej, Nákup, Výměna, Jiné), Rozměry (s podkategoriemi Tloušťka, Šířka, Délka, Průměr).

Výpis prvních dvou a také všech nepovinných (pokud jsou) zajišťuje funkce `vypis_na_pozici()`. Vstupní parametry funkce jsou pole proměnné obsahující potřebné údaje kategorie a pozice, na kterou se kategorie výběru vypisuje (to je určující pro ukládání stavu výběru do session). Ve funkci se opět rozlišuje, zda se jedná o hlavní nebo vedlejší stromovou strukturu, nicméně podstata funkčnosti je stejná. Pomocí volání funkce `fseznam_kat()` se nám vypíše výběr a při každé změně aktuálně vybrané hodnoty se, pomocí JavaScriptu, odešle formulář.

U výběrů týkajících se rozměrů se vypisují vždy tři seznamy pro výběr a zároveň s nimi i vstupní textová pole pro zadání hodnoty. A to z důvodu, že každý materiál má stěžejní vždy maximálně tři rozměry, podle kterých se standardně dají vyhledat. V jakých rozměrech se pro zvolenou kategorii bude vyhledávat nám určuje parametr kategorie *Ktyp_rozmeru*, který se při každé změně vybrané kategorie uloží do session proměnné *styp_rozmeru*. Proměnná *styp_rozmeru* slouží poté jako první vstupní parametr funkce `typ_rozmeru()`. Dále jsou jako parametry kategorie jednotlivých rozměrů společně s jejími údaji. V této funkci se tedy, prostřednictvím parametru *styp_rozmeru*, určí, jaká kategorie rozměru odpovídá určitému výběru a následně se uvnitř zavolá již známá funkce `vypis_na_pozici()`.

Vyhledání inzerátů

Pokud uživatel provede výběr z nějakého seznamu, odešle se tedy formulář s vybranými daty. Ty se uloží do session proměnných *skategorie* a následně se s nimi provádí hledání inzerátů splňujících všechny tyto parametry.

Pro každou session proměnnou *skategorie* je volána funkce `vyhledej_inzeraty()`. Vstupními parametry jsou proměnné: *skategorie*(identifikační číslo kategorie), číslo stromu dané kategorie, *inzeraty*(do této proměnné se ukládají nalezené inzeráty pro daný výběr) a *pole_inzeratu*(obsahuje inzeráty, které splnily předchozí vyhledávání). Tato funkce je také volána rekurzivně, aby došlo k projití celého podstromu vybrané kategorie. Nejdříve se podle čísla stromu určí způsob procházení a ukládání nalezených inzerátů. Pokud se jedná o vyhledávání prvního výběru, je provedeno standardní vyhledání všech inzerátů spadající do daného podstromu kategorie. Pro další prohledávání se bere v potaz výsledek předchozího, uloženého v proměnné *pole_inzeratu*. Tento způsob je výhodnější, než způsob kdy, provádíme vyhledávání pro každý výběr s následným zjišťováním průniku těchto výsledků. Ve většině případů je to i časově méně náročné, protože první vyhledání nám v podstatě určí maximální počet výsledků, od kterého se už v dalším vyhledávání jen odebírá. Pokud tedy např. v druhém vyhledání je proměnná *pole_inzeratu* prázdná, již se pro další výběry dále neprohledává.

4.4.6.2 Vstupní textové pole

Pro výpis tohoto prvku vstupního textového pole nám obstarává opět funkce `vypis_na_pozici()`.

Jak již bylo zmíněno, první použití vstupního textového pole jsme zaznamenali u zadávání rozměrů pro vyhledávání. Kromě seznamů pro výběr zde máme i tři textová pole, pro zadání vlastních hledaných hodnot. Po odeslání formuláře se první část zpracování liší od zpracování dalších vstupních textových polí. Je to především dáno jistou svázaností s funkčností týkající se položek pro rozměry. Nicméně výsledkem prvotního zpracování těchto dat je uložení nalezených inzerátů do session proměnné *sinzerat*, se kterou se dále pracuje už stejně.

Jelikož zpracování hodnot z výběrů předchází zpracováním hodnot z textových polí, musíme vzít výsledek z výběrů v potaz(uloženo v proměnné *pole_inzeratu*). Máme-li v session proměnné *sinzerat* nějaký záznam a zároveň záznam v proměnné *pole_inzeratu*, provádíme pro každou hodnotu *sinzerat* postupný průnik. Je to principiálně stejné jako u vyhledávání u výběrů ze seznamu. Je-li proměnná *pole_inzeratu* prázdná, provede se postupný průnik vzájemně s hodnotami polí session proměnné *sinzerat*.

Výslednou množinu inzerátů splňujících všechny zadané parametry pro vyhledávání následně vypíšeme pomocí již známé funkce `vypis_i()`.

5 Závěr

5.1 Zhodnocení výsledků

Výsledkem práce na diplomovém projektu je multiuživatelský systém pro podporu znovuvyužití materiálů. Do jaké míry se podařilo vytvořit vyhovující strukturu je jen na praktickém otestování. Nicméně, tento systém, s možností vytvářet neomezené množství stromových struktur, by měl vyhovovat právě těmto společnostem, zabývajícím se znovuvyužitím materiálů. Ať už se jedná o recyklaci skla, plastů nebo například ve strojírenství kovů a s ním souvisejících materiálů. V těchto oblastech je důležitým faktorem i přesnost údajů o jednotlivých materiálech. Proto bylo vytvořeno přehledné a typizované vyhledávání, které by mělo svojí funkčností naplnit očekávání oboru znalých uživatelů.

5.2 Vlastní přínos

Vlastním přínosem diplomové práce je získaný přehled o strukturách inzertních systémů, jednotlivých částí a jejich významu. Na počátku této práce jsem získal spoustu cenných informací z oblasti Business Intelligence, datových skladů, analýzy OLAP (OnLine Analytical Processing) a dolování dat (data mining), které mi následně pomohly při navrhování a realizaci vyhledávání v multiuživatelském systému. Snahou bylo také vytvořit obecnější návrh, který by, při konkrétní realizaci, nevyžadoval přílišné zásahy do kódu.

5.3 Návaznost

Návazná práce na tomto projektu může být především v konkrétní realizaci projektu pro podporu znovuvyužití materiálů, především v oblasti strojírenství, kde je velmi mnoho parametrů a specifikací pro jednotlivé materiály. Právě konkrétní zaměření na tuto oblast může prověřit vhodnost použití stromových struktur menu a způsob netriviálního vyhledávání.

Literatura

- [1] Kosek Jiří, PHP Tvorba interaktivních internetových aplikací, GRADA, Praha, 1998.
- [2] Hernandez, M., J., Viescas, J., L., Myslíme v jazyku SQL tvorba dotazů, GRADA, Praha, 2004.
- [3] Lacko, L., Datové sklady analýza OLAP a dolování dat, Computer Press, Brno, 2003.
- [4] Václavek, P., JavaScript, Computer Press, 2006.
- [5] Huseby, S., H., Zranitelný kód, Computer Press, Brno, 2006.
- [6] Šablonovací system Smarty
<http://www.abclinuxu.cz/serialy/smarty-sablonovaci-system>, <http://smarty.php.net>
- [7] PEAR, <http://www.abclinuxu.cz/serialy/pear>
- [8] HTML, <http://cs.wikipedia.org/wiki/HTML>
- [9] CSS, <http://www.jakpsatweb.cz/css/css-uvod.html>
- [10] PHP jazyk, <http://php.cz>
- [11] MySQL, <http://dev.mysql.com/>
- [12] XAMPP, <http://www.dk-studio.net/eblog/1158429601-xampp-apache-2-php-5-mysql-5.php>
- [13] JavaScript, <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>
- [14] ISO slovník, <http://www.eiso.cz/informacni-servis/eiso-slovník/>
- [15] Session v PHP, <http://interval.cz/clanky/pracujeme-se-session-v-php/>
- [16] DynamicTree, <http://www.hotscripts.com/search/15240344.html>
- [17] Syntaxe shody se vzorem, Hernandez, M., J., Viescas, J., L., Myslíme v jazyku SQL tvorba dotazů, GRADA, Praha, 2004, s. 136-137.

Seznam příloh

CD

Na přiloženém CD jsou všechny zdrojové soubory včetně součástí grafického návrhu multiuživatelského systému. Dále zde naleznete vyexportované tabulky použité databáze a instalační balíček Xampp.