

Obsah

1	ÚVOD K BAKALÁRSKEJ PRÁCI	9
2	ÚVOD K INTERNETOVÝM INFORMAČNÝM SYSTÉMOM	10
2.1	KATALÓGY.....	10
2.2	VYHLADÁVANIE NA INTERNETE.....	11
2.2.1	<i>Optimalizácia pre vyhľadávače - SEO</i>	11
2.2.2	<i>Kľúčové faktory</i>	12
3	ZOZNAM POUŽITÝCH TECHNOLOGÍI.....	13
3.1	HYPertext MARKUP LANGUAGE	13
3.1.1	<i>História jazyka HTML</i>	13
3.1.2	<i>Syntax HTML</i>	13
3.1.3	<i>Štruktúra dokumentu HTML</i>	14
3.2	CASCADING STYLE SHEET	15
3.3	SKRIPTOVACÍ PROGRAMOVACÍ JAZYK PHP	16
3.3.1	<i>História</i>	16
3.3.2	<i>Syntax PHP</i>	16
3.3.3	<i>Príkazy</i>	16
3.3.4	<i>Premenné</i>	16
3.4	DATABÁZOVÝ SYSTÉM MYSQL.....	17
3.4.1	<i>Popis MySql</i>	17
3.4.2	<i>Implementácia MySql</i>	17
3.5	APACHE HTTP SERVER 2	18
3.5.1	<i>Typológia webového serveru</i>	18

4	RIEŠENIE.....	19
4.1	ANALÝZA PROBLÉMU	19
4.2	NÁVRH RIEŠENIA.....	20
4.3	ŠTRUKTÚRA DATABÁZE.....	21
4.4	ARCHITEKTÚRA PROGRAMU	22
5	IMPLEMENTÁCIA UŽÍVATEĽSKEJ ČASTI.....	23
5.1	STATICKÁ STRÁNKA.....	23
5.2	REGISTRÁCIA FIRIEM	24
5.3	VYHLÁDÁVANIE V KATALÓGU	25
5.4	PREHLIADANIE FIRIEM.....	27
5.5	ADMINISTRAČNÁ ČASŤ	29
5.6	ŠTRUKTÚRA PRE VEĽKÚ ZÁŤAŽ	30
5.6.1	<i>Štruktúra bez použitia tabuľky katalogindex</i>	<i>30</i>
5.6.2	<i>Štruktúra s použitím tabuľky katalogindex</i>	<i>31</i>
5.6.3	<i>Pred ukladanie výstupov</i>	<i>32</i>
6	ZÁVER	33
6.1	DOSIAHNUTÉ VÝSLEDKY.....	33
6.2	VLASTNÝ PRÍNOS	34
6.3	ĎALŠÍ VÝVOJ.....	34
7	ZOZNAM POUŽITÝCH ZDROJOV	35
8	ZOZNAM PRÍLOH	36

1 Úvod k bakalárskej práci

Tému „Návrh a implementácia štruktúry vysoko vyťažných informačných systémov s webovým rozhraním“ som si vybral predovšetkým z dôvodu jej zaujímavosti, atraktivity a potenciálu vo využití v praxi. V dnešnej dobe je táto téma aj externou prácou pre firmy zaoberajúcimi sa výrobou internetových katalógov a prezentácií rôznych druhov a prevedení.

Informačný systém riešený v mojej práci si kladie za cieľ vytvoriť ucelený zoznam veľkého množstva záznamov o firmách, správu dát a rozšíriteľnosť funkcií systému podľa potrieb užívateľa.

Je určený pre široký záber užívateľov internetu rôznych vekových kategórií hľadajúcich bežné praktické informácie o firmách ale i obchodným partnerom poskytujúc základné kontaktné informácie a má za cieľ ušetriť čas a námahu pri výbere z konkurencie na úrovni oboru činnosti a teritória.

Systém by mal spĺňať základné funkčné požiadavky ako sú rýchlosť zobrazovania výsledkov s ohľadom na technické vybavenie užívateľa, dostupnosť, jednoduchosť, prehľadnosť, intuitívne ovládanie, taktiež technické a provozové požiadavky ako maximálna automatizovanosť, spoľahlivosť, minimum hardwarových rozšírení a v neposlednom rade primerané investičné náklady.

Navrhnuté optimalizačné metódy budem demonštrovať na katalógu firiem, ktorý má potenciál i využitie k zobrazovaniu a správe veľkého množstva dát širokej verejnosti a tým i nadmernému počtu užívateľov.

Praktický prínos vidím v komerčnom využití tejto technológie. Užívateľ internetu má možnosť vyhľadávať vhodne zatriedené informácie o firmách do jedného celku. Môže tak získať údaje ako adresu, telefónne čísla poprípade webové stránky firiem bez zdĺhavého skúmania rôzne vyzerajúcich stránok.

Ďalší prínos je pre samotné firmy, ktoré tak získajú vlastnú vizitku bez potreby akejkoľvek investície do vývoja vlastných stránok alebo k lepšej prezentácii vlastných stránok. Nehovoriac o zvýšení šance danú firmu nájsť na internete a tým zvýšením dopytu po službách.

Kapitola 2 sa zaoberá internetovými informačnými systémami, ako sú katalógy, vyhľadávaniu na internete a optimalizáciou pre vyhľadávacie roboty *SEO* [1], [11]. V nasledujúcich kapitolách sú zhrnuté použité technológie, návrh riešenia a implementáciu systému. V záverečnej kapitole sú prezentované konkrétne výstupy a zhodnotené dosiahnuté výsledky.

2 Úvod k internetovým informačným systémom

Informačné systémy slúžia k zbieraniu, udržovaniu, spracovaniu a poskytovaniu informácií a dát. Internetový informačný systém slúži k zdieľaniu dát užívateľom internetu. V dnešnej dobe je internet celosvetovou záležitosťou a preto je nutné aby informačné systémy boli prístupné komukoľvek a odkiaľkoľvek. S tým je úzko spojená aj náročnosť na prevádzku a údržbu.

Za konkrétny príklad informačného systému bol zvolený internetový katalóg zameraný na prezentáciu firiem, živnostníkov. Rozhodol som sa z dôvodu vývoja katalógových informačných systémov pre firmy. Ako príklad uvádzam katalóg MAXportal.cz.

2.1 Katalógy

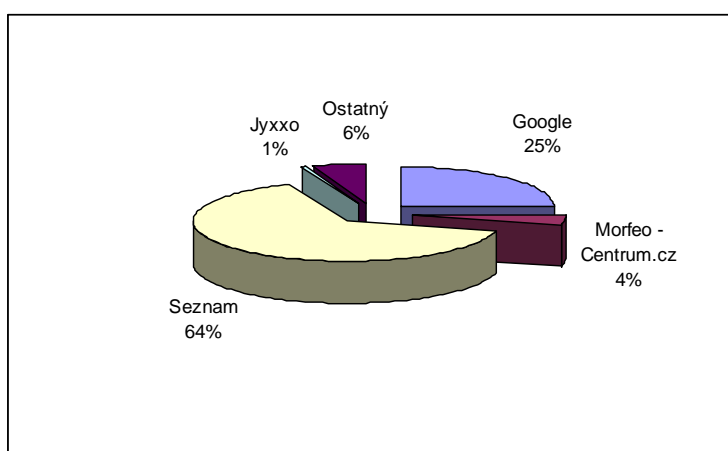
Jedným z nástrojov na vyhľadávanie z dát zverejnených na internete je využiť rôzne odborové a tematické katalógy ktoré spracovávajú a triedia dáta do kategórií. Rozdiel oproti internetovým vyhľadávateľom je v množstve spracovaných dát, druhu dát a v samotnom získavaní dát. Vyhľadávače spracovávajú v drvivej väčšine stránky ktoré si samé nájdu pomocou odkazov na samotných stránkach. Katalógy fungujú na niekoľkých princípoch v závislosti na druhu katalógu. V prevažnej väčšine sa jedná o ručne vkladanie dát samotnými vlastníkmi dát, a tým si zaistili zvýšenie možnosti nájdenia. Samotné dáta bývajú v kvalitnejších katalógoch triedené a kontrolované odbornými pracovníkmi a tým sa zaručila pravovernosť a presnosť informácií.

Štandardne sa jedná o ucelený zoznam rôznych celkov s rovnakým zameraním, významom alebo účelom. Ako príklad si možno predstaviť zoznam poštových smerovacích čísel. Každé miesto je brané ako samostatný celok, ku ktorému patrí číslo – poštové smerovacie číslo. Význam každého miesta je rovnaký, má jednoznačne informovať o umiestnení danej lokality v primárne určených hlavných lokalitách. To znamená usporiadané zmysluplné informácie.

Historicky siaha význam slova až do obdobia antiky, kde znamenalo výčet mien veršovanou skladbou.

2.2 Vyhľadávanie na internete

Ako som už naznačil, pre vyhľadávanie na internete slúžia internetové vyhľadávače. Fungujú na princípe prechádzania stránok robotom, tzv. „botmi“, ktoré zaznamenávajú obsah stránok do vlastných databáz, kde sa zatriedia tak aby po zadaní vyhľadávacieho slova vrátili čo najrelevantnejšie stránky v čo najkratšom čase. Medzi najväčšie z nich vo svete patrí vyhľadávanie od firmy Google, Yahoo a na českom internete Seznam, Morfeo a Jyxxo. Zastúpenie na českom trhu je zobrazené na **Obr. 2.2.1**.



Obr. 2.2.1: Zastúpenie vyhľadávačov na českom trhu

2.2.1 Optimalizácia pre vyhľadávače - SEO

Pre úspešnosť nájdenia stránky je potrebné optimalizovať, pretože vyhľadávače určujú relevanciu stránok pomocou rôznych postupov a pravidiel k danému vyhľadávanému výrazu. Pre dosiahnutie kvalitnej webovej prezentácie je nutné bezpodmienečne dodržiavať tieto pravidlá.

Obor zaoberajúci sa optimalizáciou pre vyhľadávače sa nazýva *SEO* [1], [11], z anglického *Search Engine Optimization*. Má za úlohu vyhodnocovať konkrétnu stránku a vhodne vybrať metódu ktorou sa dosiahne čo možno na najlepšiu pozíciu vo výsledkoch vyhľadávača na hľadaný výraz dominantný k účelu stránky. Tým sa zaručí základná funkcia každej stránky, a to aby bola navštevovaná.

2.2.2 Kľúčové faktory

V tejto kapitole uvediem základné, všeobecné známe zásady *SEO* overené tvorbou internetových stránok v praxi.

- 1) Správny výber **kľúčových slov**, ktoré zaručia predné pozície vo vyhľadávaní. Tieto slová musia odrážať cieľ webovej stránky a musia byť zamerané na správny druh užívateľa. Ideálne sú hľadané slová s malou konkurenciou. Ako príklad uvediem stránky inštalatéra v Nitre, pre ktorého optimalizácia na slovo *inštalatér* nemá význam pokiaľ bude na predných pozíciách vyhľadávača, kde budú hľadať zákazníci z Košíc a veľká konkurencia na toto slovo. Preto je najlepšia kombinácia slov *inštalatér Nitra*.
- 2) **Obsah**. Na obsahu závisí najviac. Je dôležité mať zaujímavý a zrozumiteľný obsah pre užívateľa, ako potencionálneho zákazníka a taktiež pre vyhľadávací robot. Je nutné mať vlastný, originálny text.
- 3) Veľmi dôležitým faktorom je **titulka stránky**. Často je zhodná s podstránkami alebo je vyplnená nezmyselnými znakmi, v najhoršom nevyplnená vôbec.
- 4) Taktiež je potrebné vyplňovať **popis stránky** (description). Jedná sa o krátke zhrnutie informácií. Tento krátky text sa často vyskytuje u výsledkov vyhľadávania a užívateľovi tak dá lepší obraz o stránke.
- 5) **Prístupnosť robotom**. I nad dokonalosť vyhľadávacích robotov, ktoré v súčasnosti prehľadávajú internet, je potrebné neklásť prekážky. Ak je na stránkach flashový prvok, rámeček, alebo obrázok, je nutné vyplniť tomuto elementu textovú alternatívu.
- 6) Každá stránka musí mať pevnú a nemennú adresu. Pokiaľ je nutné túto adresu meniť, alebo nastala zmena v štruktúre webu, stránky so starou adresou treba presmerovať na novú adresu a vyhnúť sa tým potenciálnej duplicitě obsahu.

3 Zoznam použitých technológií

3.1 Hypertext Markup Language

Webové rozhranie pre komunikáciu užívateľa s aplikáciou je vytvorené pomocou jazyka Hypertext Markup Language, HTML [2], v preklade „značkovací jazyk hypertextu“ je jeden z jazykov slúžiacich na tvorbu internetových stránok, zobrazovaniu informácií na internete.

3.1.1 História jazyka HTML

Bol navrhnutý okolo roku 1990 programátorom Tim Berners-Lee pre výskumný ústav vo Švajčiarsku. Zároveň bol navrhnutý i protokol na prenos HTML, nazvaný HTTP (Hypertext Transfer Protocol). Behom niekoľkých pár rokov nastal obrovský rozvoj webu, z toho dôvodu vznikli štandardy HTML.

3.1.2 Syntax HTML

HTML sa skladá z prostého textu a HTML značiek - tagov. Každý tag v sebe nesie nejakú vlastnosť, ktorou upravuje výstupný vzhľad dokumentu. Tagy možno rozdeliť na párové a nepárové. Párové tagy ovplyvňujú text vložený medzi. Napríklad značka symbolizuje tučné písmo. Finálny zápis vyzerá nasledovne:

```
<b>Tučné písmo</b> ,
```

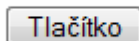
a výsledok potom:

Tučné písmo

Ďalej existujú nepárové značky, jedná sa o rôzne HTML prvky v dokumente ako sú napríklad tlačítka alebo obrázky. Príklad nepárovej značky „tlačítko“:

```
<input type="button" name="button1" value="Tlačítko"> ,
```

Výsledok je zobrazený na **Obr. 4.1**:



Obr. 4.1: HTML tlačítko

Čo sa týka veľkosti textu HTML tagov, tak na nej nezáleží, to znamená že zápis „“ je to isté ako zápis „“.

3.1.3 Štruktúra dokumentu HTML

Štruktúra HTML dokumentu musí spĺňovať presne pravidlá:

- Definícia typu dokumentu DTD
- Hlavný – koreňový element „<html>“
- Hlavičky – určujú kódovanie dokumentu, jeho popis, kľúčové slová. Nachádzajú sa v párovej značke „<head>“
- Telo dokumentu – vyznačené párovým tagom „<body>“

Príklad jednoduchého HTML dokumentu:

```
<!DOCTYPEHTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Názov stránky</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="keywords" content="Kľúčové slová" />
    <meta name="description" content="Popis stránky" />
  </head>
  <body>
    <h1>Nadpis stránky</h1>
    <p>Text v dokumente s <a href="/" title="odkaz">odkazom</a>. </p>
  </body>
</html>
```


3.2 Cascading Style Sheet

Moderný trend v návrhu HTML stránok striktno oddeľuje štruktúru webu od jej vzhľadu. Pre definíciu vzhľadu slúži *Cascading Style Sheet*, ďalej len *CSS* [3], v preklade znamená *kaskádové štýly*. Je navrhnutý pre definíciu značiek *HTML* [2] a ich vlastností. Jedná sa o neodmysliteľný prvok v práci a návrhu internetových stránok a v mojej práci ho použijem pre definíciu HTML výstupu pre užívateľa.

CSS bolo navrhnuté konzorciom W3C. Najdôležitejšou úlohou *CSS* je oddeliť definíciu vzhľadu dokumentu od jeho obsahu a štruktúry a docieľenie tak veľkej prehľadnosti a lepšej udržovateľnosti. Zmena vzhľadu mala za následok zmenu všetkých stránok dokumentu. Preto sa odobrali vlastnosti vzhľadu značkám *HTML* a bolo vytvorené *CSS*.

Pod pojmom *CSS* sa predstavuje zväčša externý súbor uchovávajúci iba definície vlastností značiek *HTML*. Tieto definície avšak môžu byť umiestnené i v samotnom *HTML* kóde, ale tomu sa snaží vyhnúť z dôvodu popierania jednej zo základných vlastností *CSS*, oddelenia štýlov od obsahu a dokumentu, a preto sa tento spôsob berie ako krok späť.

Syntax jazyka je tvorená selektorom, ktorý určuje na ktoré prvky *HTML* sa vlastnosti vzťahujú, a definíciou vlastností uzavretých do zložených zátvoriek nasledovne:

```
H1
{
  Color: #ff0000;
  Font-size: 16px;
  Margin: 10px;
}
```

Tento zápis znamená, že prvku *HTML* H1, nadpisu, je nadefinovaná farba v hexadecimálnom kóde ff0000, čo značí červenú farbu, veľkosť písma je zmenená na 16px a oddelenie nadpisu od okolitých prvkov je 10px.

3.3 Skriptovací programovací jazyk PHP

Hypertextový preprocesor, skrátene *PHP* [4] sa používa pre programovanie dynamických stránok. *PHP* je serverový jazyk, čiže všetky skripty sa vykonávajú na servery. Výsledok je vygenerovaný *HTML kód*, ktorý sa odosiela užívateľovi do prehliadača. Z toho dôvodu je *PHP* nezávislé na konfigurácii jednotlivých užívateľských počítačoch, ale iba na servery kde sa nachádza stránka. *PHP* je OpenSource – voľne šíriteľný kód, nezávislý na platforme a schopný fungovať na ktoromkoľvek servery.

3.3.1 História

Tvorcom *PHP* bol okolo roku 1994 Rasmus Lerdorf prepísaním *Perl* do jazyka *C*. V ďalších rokoch prešlo *PHP* vývojom a niekoľkými verziami, predovšetkým s vylepšeným objektovo orientovaným programovaním.

3.3.2 Syntax PHP

Ako som sa už zmienil, *PHP* vznikol prepísaním *Perl* do jazyka *C*. Tým vznikla syntax veľmi podobná práve syntaxi *C*, *Perl* a vďaka podpore objektov i *Java*.

Pre názornosť uvádzam jednoduchý skript s výpisom textu „Ahoj svet!“ :

```
<?php
    print(“Ahoj svet!”);
?>
```

3.3.3 Príkazy

Príkazy je nutné vpisovať medzi značky vyhradzujúce skript v *PHP* a pre oddelenie príkazov sa pridáva na koniec každého príkazu dvojbodka.

3.3.4 Premenné

Špecifikum *PHP* je, že každá premenná začína znakom doláru „\$“. *PHP* je „case-sensitive“, čo znamená že závisí na veľkosti písmen v názve premennej a v názve premennej nesmie byť na prvom mieste číslica.

3.4 Databázový systém MySql

Pre svoju prácu bolo treba vybrať databázu, pomocou ktorej sa budú ukladať, uchovávať a vyberať dáta. Ako najvhodnejšie riešenie som vybral databázový systém MySql [5]. Hlavné kritéria boli dostupnosť, jednoduchosť a rýchlosť v spracovaní informácií.

3.4.1 Popis MySql

MySql je multiplatformový databázový systém. SQL v názve znamená, že využíva štruktúrovaný dotazovací jazyk. Pre komunikáciu s databázou slúžia dotazy štruktúrované podľa kľúčových slov. Základná štruktúra sa delí na niekoľko príkazov:

- INSERT – týmto príkazom sa vkladajú dáta do databáze
- UPDATE – slúži pre úpravu už vložených záznamov
- SELECT – výber informácií
- DELETE – mazanie

Toto sú štyri základné a najviac používané príkazy. SQL obsahuje ešte ďalšie príkazy ktoré už nie sú tak podstatné pre moju prácu a tak si ich dovoľím vynechať.

MySql patrí do skupiny relačných databáz. V praxi to znamená, že dáta sú usporiadané do relácií – tabuliek, nad ktorými sú definované operácie. Tieto operácie sú spracované práve jazykom SQL.

3.4.2 Implementácia MySql

Databázový systém MySql je taktiež ako PHP [4] OpenSource, multiplatformný software, a preto môže byť použitý na takmer kdekoľvek. Vzhľadom na to, že patrí medzi voľne šíriteľný software, je MySql najrozšírenejší medzi databázovými systémami.

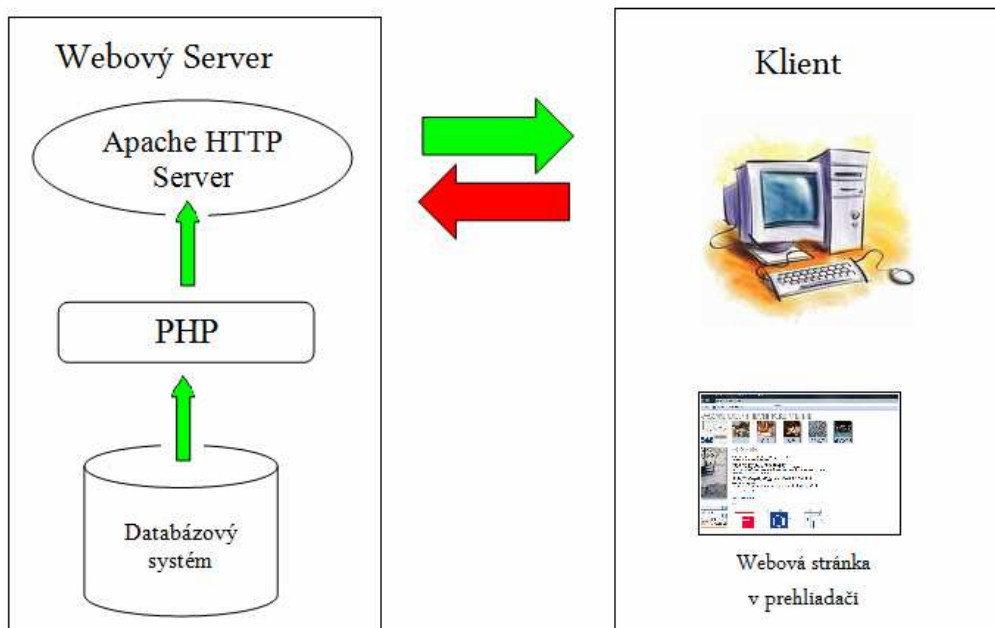
Dnes sa bežne používa ku konfigurácii webových serverov kombinácia Apache [6], PHP a MySql. Preto som si zvolil pre moju prácu práve MySql a PHP a hlavne pre výbornú kompatibilitu s bezproblémový chod ktorý je podčiarknutý OpenSource licenciou.

3.5 Apache HTTP Server 2

Pre funkčnosť internetového informačného systému je potrebný HTTP server. Najrozšírenejší a najdostupnejší je OpenSource Apache HTTP Server 2 [6]. Začiatky siahajú do roku 1993 na Illinoiskú univerzitu. Neskôr okolo roku 1998 bol vývoj pozastavený. Avšak vývoj sa opäť rozbehol vďaka správcom serverov ktorý si vytvárali vlastné rozšírenia. Tieto rozšírenia boli zastrešené skupinou Apache Group a prepísaním pôvodného kódu do Apache2.

3.5.1 Typológia webového serveru

Webový server funguje na princípe softwaru uloženom na jednom počítači, nazývanom server. Tento software slúži na komunikáciu s ostatnými počítačmi v sieti a vybavuje ich požiadavky. Pre tvorbu webových stránok nestačí len samotný server, a preto je server rozšírený o moduly ako skriptovací jazyk PHP [4] a databázový systém MySQL [5]. Celková konfigurácia je popísaná na **Obr. 3.5.1.1**.



Obr. 3.5.1.1: Komunikácia klient – server

4 Riešenie

V tejto kapitole sa budem zaoberať analýzou, vlastným riešením a tvorbou aplikácie. Štruktúru vysoko vyt'ažených informačných systémov som sa rozhodol demonštrovať na internetovom katalógu firiem. Firmy v katalógu sú rozdelené do kategórií v ktorých je možné rôzne zobrazenie firiem, vyhľadávanie a možnosť pravidelnej aktualizácie.

Informačný systém bude rozdelený do dvoch vrstiev, jedna užívateľská a druhá administratívna. V administratívnej vrstve bude mať prístup len administrátor a bude v nej môcť upravovať záznamy o firmách a ďalej bude môcť editovať kategórie.

4.1 Analýza problému

Mojim cieľom je navrhnúť informačný systém pre poskytovanie informácií veľkému počtu užívateľov s dôrazom na minimalizáciu hardwarových nárokov. Taktiež musíme dbať na optimalizáciu pre vyhľadávače a validitu HTML kódu [2].

Požiadavky pre takýto informačný systém sú **dostupnosť**, systém musí byť prístupný užívateľom, to znamená neklásť prekážky. Vychádzam pritom z faktu, že nie všetci užívatelia internetu sú rovnakí a nie všetci majú rovnako dobré technické vybavenie. Preto treba dbať o správnu a postačujúcu funkčnosť stránky pri **vypnutej grafike**, **vypnutom skriptovaní** ako je JavaScript, Java Applety a podobne, pri rôznych veľkostiach okna prehliadača alebo pri vypnutí kaskádových štýlov.

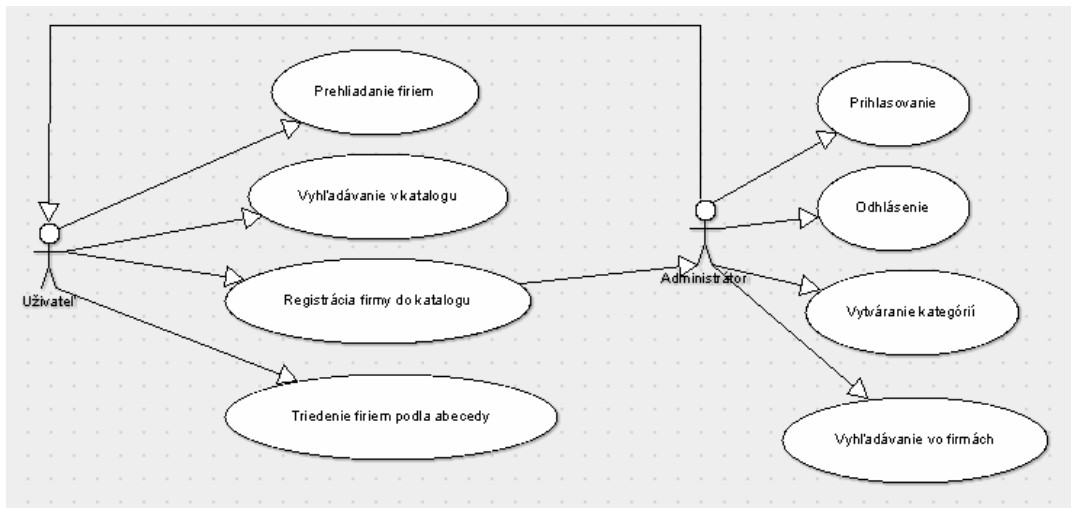
Ďalšou dôležitou požiadavkou je **rýchlosť vykonávania skriptov**. Skript na načítanie jednej stránky musí trvať desatiny sekúnd. S tým úzko súvisia aj **hardwarové nároky**. Pri vyššom časovom intervale je s pribúdajúcim množstvom dotazov záťaž na server príliš veľká, a preto je nevyhnutné zamerať sa na maximalizáciu rýchlosti skriptov a tým minimalizovať hardwarové nároky.

Vyhľadávanie v týchto informáciách je taktiež dôležité pre dobrú výpovednú hodnotu. Pre môj projekt vytvorím jednoduché ale postačujúce vyhľadávanie.

Optimalizácia pre vyhľadávače je veľmi široká téma ktorú nemožno pri návrhu akejkoľvek internetovej stránke pominúť pokiaľ má splňovať svoju najzákladnejšiu úlohu – **návštevnosť**. Ja sa v mojej práci zameriam na základnú optimalizáciu, týkajúcu sa štruktúry HTML kódu a správnosť postupov tvorby kvalitných stránok. S tým úzko súvisí i **validita** stránky a jej HTML kódu podľa známych štandardov.

4.2 Návrh riešenia

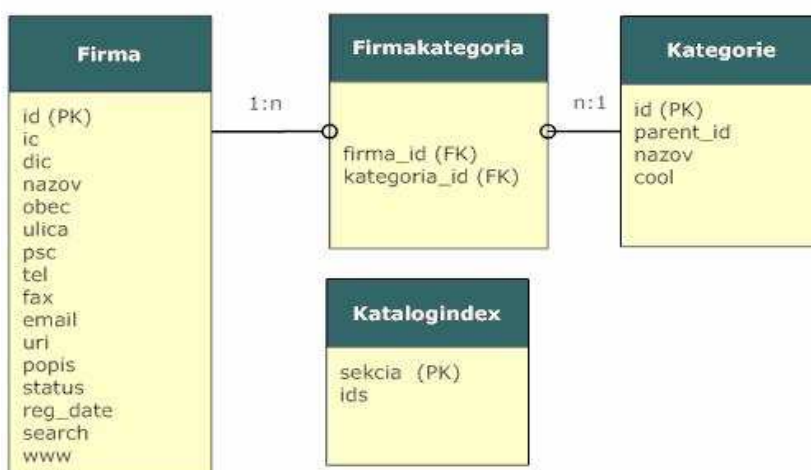
Informačný systém má za úlohu bežné použitie, z toho dôvodu som vytvoril *diagram prípadov užitia* [7] pre prehľadnosť a lepšiu predstavu o systéme.



Obr. 4.2.1: Diagram prípadov užitia

Podľa *diagramu prípadov užitia* je prehľadné rozdelenie informačného systému spolu s možnosťami pre užívateľa.

Pre správu dát bol použitý databázový systém. Tento databázový systém uchováva všetky dáta o firmách a kategóriách v celom katalogu, a má za úlohu rýchly a jednoduchý prístup v dynamických skriptoch ktoré generujú požadovaný výsledok. Štruktúra je zobrazená pomocou *E-R diagramu* na **obrázku 4.2.2**:



Obr.: 4.2.2: Štruktúra databázových tabuliek v E-R diagrame

4.3 Štruktúra databáze

Systém sa skladá z databáze ktorá plní dôležitú súčasť webového informačného systému z dôvodu ukladania dát nevyhnutných pre chod celého katalógu. Štruktúra vychádza z *E-R Diagramu*.

Základná kostra databáze katalógu je uvedená v **tabuľke 4.3.1**.

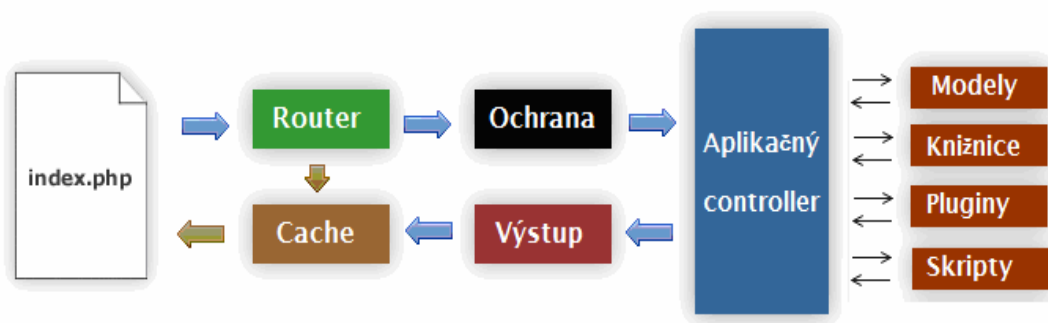
Názov tabuľky	Stĺpec	Typ	Nulový	Extra	Kľúč
firma	id	INT(11)	NIE	auto increment	Primárny kľúč
	ic	VARCHAR(50)	NIE		
	dic	VARCHAR(50)	NIE		
	nazov	VARCHAR(255)	NIE		
	obec	VARCHAR(50)	NIE		
	ulica	VARCHAR(50)	NIE		
	psc	VARCHAR(6)	NIE		
	tel	VARCHAR(20)	NIE		
	fax	VARCHAR(20)	NIE		
	email	VARCHAR(50)	NIE		
	www	VARCHAR(255)	NIE		
	uri	VARCHAR(255)	NIE		
	popis	TEXT	ANO		
	status	INT(1)	NIE		
	reg_date	INT(11)	NIE		
search	TEXT	ANO			
firkategoria	firma_id	INT(11)	NIE		Cudzí kľúč
	kategoria_id	INT(11)	NIE		
kategorie	id	INT(11)	NIE	auto increment	Primárny kľúč
	parent_id	INT(11)	ANO		
	nazov	VARCHAR(255)	NIE		
	cool	VARCHAR(255)	NIE		
katalogindex	sekcia	VARCHAR(10)	NIE		Primárny kľúč
	ids	LONGTEXT	NIE		

Tabuľka 4.3.1: Tabuľka databázovej štruktúry

4.4 Architektúra programu

Základným kameňom architektúry programu je rámec *CodeIgniter* [8], voľne šíriteľný za použitia návrhového vzoru *Model-View-Controller* naprogramovaný v PHP [4]. Jedná sa o objektovú štruktúru, ktorá oddeľuje dáta a ich spracovanie od ich zobrazenia. *Model* predstavuje dátové úložisko, obstaráva a získava dáta ktoré poskytuje controlleru ktorý daný model zavola a následne tieto dáta presúva do *view*, kde sa dáta upravujú pre finálny výstup.

Architektúra rámca *CodeIgniter* je znázornená na **obrázku 4.4.1**:



Obr. 4.4.1.: Architektúra rámca CodeIgniter

Index.php je základný controller, inicializuje základné prostriedky k fungovaniu rámca. *Router* vyhodnocuje HTTP požiadavku, ak existuje pred uložený súbor (*cache*), tak vráti tento ako výstup. *Aplikčný controller* načíta modely, knihnice, pluginy a ostatné prostriedky a na záver view vytvára výstup do prehliadača, a ak je nastavené cachovanie, uloží sa tento výstup.

5 Implementácia užívateľskej časti

5.1 Statická stránka

Internetové stránky určené pre užívateľov sú navrhnuté pomocou jazyka HTML a štandardu HTML 4.01 Transitional [2], [9]. Vzhľad dokumentu je nadefinovaný pomocou kaskádových štýlov CSS.

Základná statická stránka z ktorej vychádzajú ostatné stránky je tvorená hlavičkou, obsahom tvoreným vlastným textom stránky a pätičky. Táto stránka je tvorená CSS nadefinovanými v externom súbore.

Hlavička stránky obsahuje *hlavičku HTML* kódu obsahujúcu *meta tagy*. *Meta tagy* sú veľmi dôležitou súčasťou každej stránky a mal by byť na ňu kladený patričný dôraz pokiaľ chceme stránku správne navrhnuť z hľadiska SEO. V hlavičke sa taktiež nachádza cesta k externému CSS súboru.

Celá stránka je uzatvorená do jedného prvku `<div>`, ktorý je nadefinovaný v CSS pomocou selektora *content*.

Selektor *content* je má v sebe nadefinovanú pevnú šírku, ktorú som zvolil na veľkosť *800px* z dôvodu lepšieho zobrazovania na monitoroch s nízkym rozlíšením. Taktiež pre lepšiu zobrazovaciu schopnosť je stránka vertikálne vycentrovaná na stred a na *10px* od horného a dolného okraja. To som docielil vlastnosťou *margin: 10px auto*.

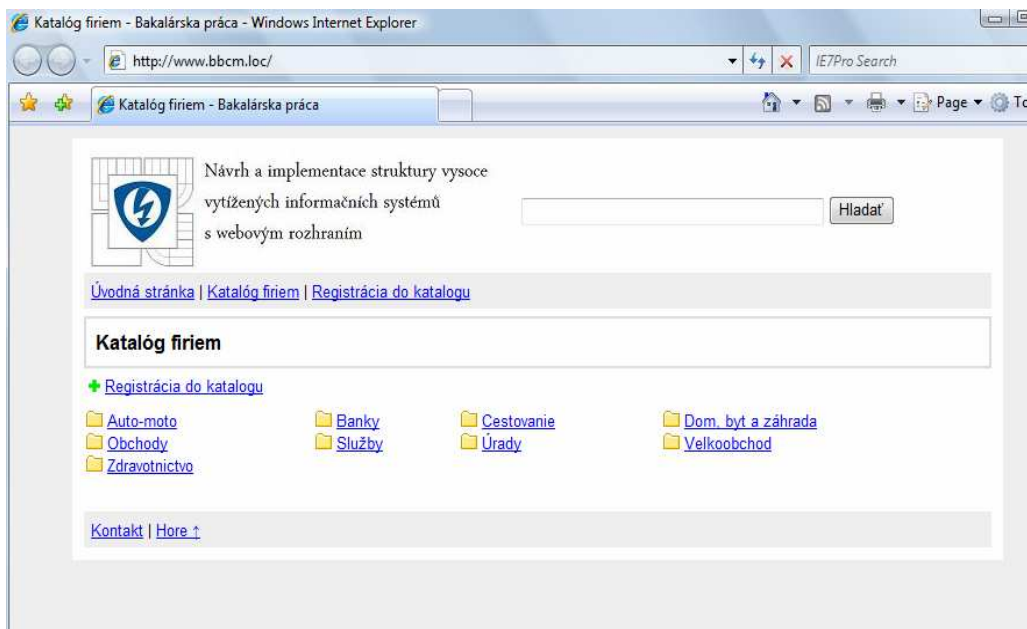
Pre väčšiu prehľadnosť bol typ písma zvolený na *font-family: arial, helvetica, sans-serif* a veľkosť písma na *80%*. Percentá preto, aby bolo zaručené zväčšovanie vo všetkých prehliadačoch a tým zvýšená dostupnosť ľuďom s horším zrakom.

Na konci stránky je umiestnená hlavička, kde sa nachádzajú kontaktné informácie poprípade copyright.

Táto statická stránka bola vytvorená tak, aby zodpovedala najnovším štandardom v tvorbe webových stránok. Je užívateľsky jednoduchá a prehľadná, čo sú základné požiadavky, a navrhnutá pre najlepšiu výpovednú hodnotu stránok.

Každá ďalšia stránka je zložená z rovnakej hlavičky a pätičky, ktoré sú vložené na začiatok popřípade nakoniec stránky.

Výsledný vzhľad hlavnej stránky je zobrazený na obrázku 5.1.1:



Obr. 5.1.1: Vzhľad úvodnej stránky

5.2 Registrácia firiem

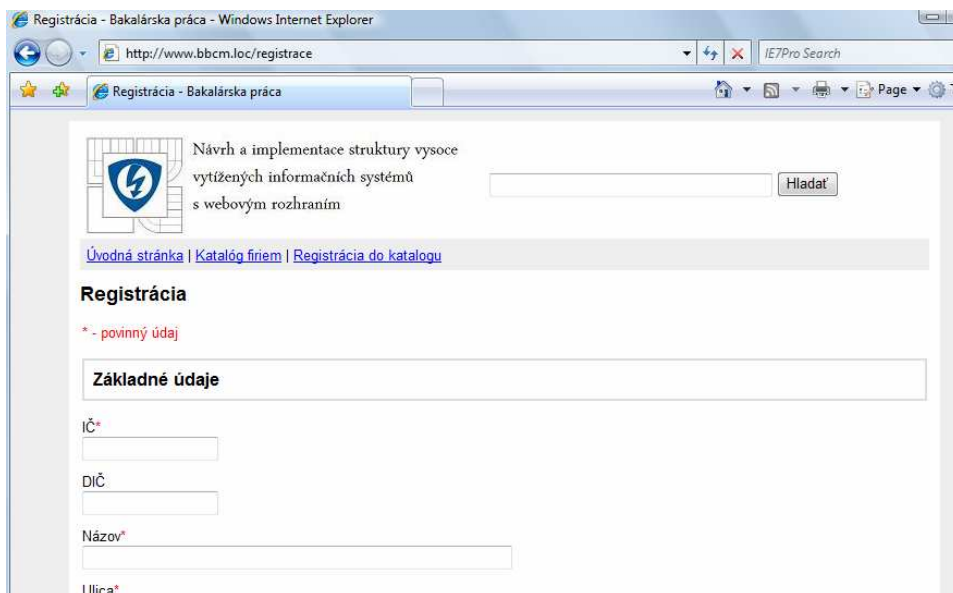
Registrácia firiem je dôležitá časť z hľadiska správy katalogu a preto som ju zaradil do mojej práce.

Registrácia pozostáva z HTML formulára obsahujúceho povinné položky (IČ, názov, ulicu, obec, poštové smerovacie číslo, popis činnosti, kategórie, telefón, email a položku pre opísanie textu z obrázku) a nepovinné položky (DIČ, fax a webové stránky firmy).

Po vyplnení údajov je formulár odoslaný na spracovanie. O spracovanie formulára sa stará *controller* **registrace** rámca *Codeigniter* ktorý má cestu `/system/application/controllers/`. Tento controller skontroluje či boli odoslané dáta pomocou metódy POST. Ak áno, tak nastane kontrola správnosti vyplnených položiek formulára. Ak objaví chybu, vyplní premennú `$error` aj s patričným chybovým hlásením, nedovolí firmu registrovať a vráti späť registračný formulár avšak už s pred vyplnenými údajmi od užívateľa.

Ak sú splnené podmienky pre registráciu, dáta v poli \$_POST sú odoslané modelu *firma* ktorý sa nachádza v `/system/application/models/`. Jedná sa o triedu obsahujúcu metódy na prácu s tabuľkou *firma*, ako napríklad ukladanie (add), úpravu (edit) alebo mazanie (delete). Táto trieda následne uloží dáta do databáze a uloží firmu so statusom ktorý mu neumožňuje zobrazovať sa, do doby, pokiaľ túto registráciu neskontroluje administrátor.

Registračný formulár je zobrazený na obrázku 5.2.1:



Obr. 5.2.1: Registračný formulár

5.3 Vyhľadávanie v katalógu

Vyhľadávanie funguje na princípe generovania špeciálneho stĺpca tabuľky *firma*, `search`. Tento stĺpec je vytvorený pri registrácii alebo upravený pri editácii informácií o firme. Sú v ňom vložené reťazce popisujúce daný záznam a jeho zaradenie do štruktúry.

O vytvorenie adekvátneho reťazca sa stará metóda `addSearch` modelu *Firma*. Táto metóda si podľa predaného parametra `$id` vyberie z databáze údaje (IČ, DIČ, popis, názov, uri, obec, ulicu, poštové smerovacie číslo a www stránky) o firme, ktoré oddelí pomlčkou a priradí k reťazcu `$search`. K reťazcu `$search` sa priradí aj stĺpec `cool` z tabuľky *kategoria* ktorý je vybraný pomocou nasledujúceho dotazu:

```
SELECT cool FROM kategorie WHERE id IN (SELECT kategoria_id FROM firmakategoria WHERE firma_id = '.$id.')
```

Jedná sa o zložený dotaz z dvoch SQL dotazov, kde v prvom dotaze vyberáme požadovaný stĺpec *cool* z výsledku druhého dotazu, v ktorom vyberáme všetky kategórie, konkrétne ich primárne kľúče *kategoria_id*, podľa ktorého nájdeme k danej firme jej správnu kategóriu.

Výsledok ktorý nám vráti tento dotaz je pole *\$kat*, obsahujúce všetky *cool* stĺpce zo všetkých kategórií, a preto musíme použiť cyklus *foreach* pre priradenie do požadovaného reťazca *\$search* nasledovne:

```
foreach($kat as $v)
{
    $search .= '-'.$v['cool'];
}
```

Po týchto krokoch máme v premennej *\$search* požadovaný reťazec. Pre uloženie do databáze je potrebné odstrániť diakritiku a malé a veľké písmená. K tomuto účelu som vytvoril metódu *MakeCool* modelu *Functions*. Jej kód vyzerá nasledovne:

```
function MakeCool($str=" ")
{
    return trim(preg_replace('~[^\p{L}\p{N}]~', '-', strtolower($this->Cz2Ascii(
    preg_replace( '~[^\p{L}\p{N}]~u', '-', trim($str))))),'-');
```

Vyhľadávanie prebieha tak, že hľadaný výraz je odoslaný controlleru *redirect* a jeho metóde *search*, kde prebieha vytvorenie url adresy pre dané vyhľadávanie. Url adresa pozostáva z orezaného hľadaného reťazca o diakritiku pomocou hore zmienenej funkcie *\$this->Functions->MakeCool* a zakódovaného výrazu do hexakódu, pomocou ktorého je akákoľvek url adresa vždy správne interpretovaná.

Po vytvorení správnej url adresy je stránka presmerovaná práve na túto url adresu, do controlleru *search*. Tu sa url adresa znova dekoduje. Po dekodovaní sa vytvorí SQL dotaz do databáze.

Samotný SQL dotaz pozostáva z slova, vytvoreného použitím metódy *MakeCool* na hľadanom výraze. Následne sa takto pripravené slovo rozdelí na niekoľko častí pomocou funkcie *explode()* :

```
$this->data['wordQ'] = $this->Functions->MakeCool($qString);  
$qarray = array_unique(explode('-', $this->data['wordQ']));
```

Pre pole parciálnych slov sa v cykle *foreach* vytvorí dynamický SQL dotaz.

```
foreach($qarray as $v)  
{  
    $sql.=' search LIKE "%'.substr(trim($v), 0, strlen($v)-floor(strlen($v)/5)).%" AND';  
}
```

Pre väčšiu presnosť a minimalizácie vplyvu skloňovania je každé parciálne slovo orezané pomocou funkcie *substr* o pätinu svojej dĺžky:

```
substr(trim($v), 0, strlen($v)-floor(strlen($v)/5))
```

Výsledný SQL dotaz vyzerá po doplnení dynamicky generovaného dotazu s obojstranným *LIKE* nasledovne:

```
SELECT SQL_CALC_FOUND_ROWS DISTINCT(firma.id), firma.*  
FROM firma WHERE status <> 0 AND ('.$sql.')  
LIMIT '$od.', '$this->data['itemsPerPage'];
```

5.4 Prehliadanie firiem

Pod pojmom prehliadanie firiem chcem spresniť spôsob, ako možno zobrazovať veľké množstvo dát s ohľadom na minimalizáciu záťaže v pomere k množstvu informácií vo webovej prezentácii.

Prehliadanie firiem vykonáva controller *catalog*. Z url štruktúry vyberiem parametre. Ak sa žiadny parameter za controllerom nenachádza, načítam z databáze posledné vložené firmy. Je to z dôvodu zlepšenia *SEO* [1], [11], hlavne rýchlosti načítania nových informácií vyhľadávacími robotmi.

Pokiaľ je v url ďalší parameter, pokúsim sa tento parameter vyhľadať medzi kategóriami v stĺpci *cool*. Ak sa nájde zhoda, pokračujem nájdením primárnych kľúčov firiem v tabuľke *catalogindex*. Táto tabuľka má špeciálny význam vo vysoko vyťažených informačných systémoch. Funguje na princípe pred vyplnenia primárnych kľúčov firiem k danému zaraďovaciemu prvku, v mojom prípade sa jedná o kategóriu.

Ak by sme uvažovali napríklad o lokalitách tak by tam bol počet riadkov navýšený o počet lokalít za predpokladu že v každej kategórii je minimálne jedna firma. O pre vyplňanie tejto tabuľky sa stará metóda *indexovanieKat* controlleru *cron*. Spúšťa sa v určitých časových intervaloch, ideálne v nočných hodinách, kedy je záťaž najmenšia.

Jedna z ďalších výhod tabuľky *catalogindex* je že každý riadok je schopný držať poradie primárnych kľúčov, čo má veľké výhody ak je triedenie výsledkov netriviálne. Je to napríklad pri pravidelnej rotácii, ale na tú sa v mojom projekte nebudem zameriavať.

Po takto získaných primárnych kľúčoch spracujem pole vybraním iba tých firiem, ktoré potrebujem vzhľadom na limitovanie počtu firiem na stránku, v mojej práci to je 10 firiem na jednu stránku katalógu. Následne vytvorím dynamický SQL dotaz pre výber potrebných informácií o firmách z databáze:

```
SELECT SQL_CALC_FOUND_ROWS(firma.id), firma.* FROM firma
WHERE id IN (.$ids.) AND status <> 0
ORDER BY reg_date '.$order.' LIMIT '.$od.', '.$this->data['itemsPerPage']
```

kde premenná *\$ids* obsahuje primárne kľúče firiem oddelených čiarkou, *\$order* je spôsob triedenia, vzostupne (*ASC*) alebo zostupne (*DESC*), *\$od* je poradie od ktorého sa má začať vyberať a *\$this->data['itemsPerPage']* obsahuje maximálny počet firiem, ktoré sa vyberajú, v mojom prípade to je 10.

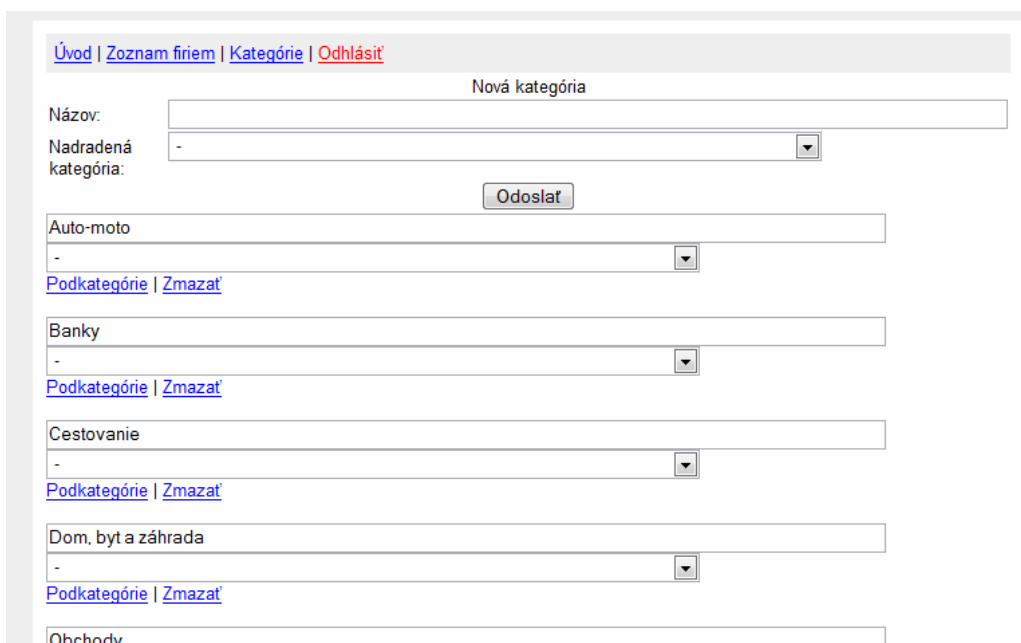
Samotný výstup sa nachádza v */system/application/views/vypis.php*, kde je krátky popis firmy vypísaný pomocou HTML zoznamu spolu s odkazom na kompletne informácie.

5.5 Administračná časť

Administračná časť obsahuje minimum pre správu celého katalógu. Je prístupná len administrátorovi po prihlásení. Obsahuje údaje o posledných zaregistrovaných firmách, základné akcie k úprave ako povolenie, zakázanie alebo zmazanie firmy.

Administrátor môže vyhľadávať v databáze a upravovať firmy. Má k dispozícii editáciu kategórií katalógu. Ako poslednú možnosť má odhlásenie zo systému.

Na obrázku 5.5.1 je zobrazená administračná časť slúžiaca na úpravu kategórií.



The screenshot shows a web interface for managing categories. At the top, there is a navigation bar with links: [Úvod](#), [Zoznam firiem](#), [Kategórie](#), and [Odhlásiť](#). Below this, the page title is "Nová kategória". The form includes a text input for "Názov:" and a dropdown menu for "Nadradená kategória:". A "Odoslať" button is positioned below the dropdown. Below the form, there is a list of existing categories, each with a text input, a dropdown menu, and two links: "Podkategórie" and "Zmazať". The categories listed are "Auto-moto", "Banky", "Cestovanie", "Dom, byt a záhrada", and "Obchodv".

Obr. 5.5.1: Úprava kategórií v administračnej časti

5.6 Štruktúra pre veľkú záťaž

Štruktúra pre veľkú záťaž bola navrhnutá tak, aby systém generoval výstupné stránky v čo možno najkratších časoch. Do úvahy som musel zobrať objem dát v databáze, množstvo firiem ako základných entít a počet dotazov na systém za jednotku času. Základný problém spočíva v efektívnom vyberaní firiem podľa zaradenia podľa rozdeľovacích kritérií, v mojej práci je jediné, rozdelenie do kategórie.

Pre rozhodnutie o riešení, aké som zvolil, som vykonal testy pre rôzne varianty štruktúr. Testoval som štruktúru bez indexovacej tabuľky *katalogindex*, s použitím tabuľky *katalogindex* a nakoniec s použitím pred ukladania výstupov.

5.6.1 Štruktúra bez použitia tabuľky *katalogindex*

Základný problém, zaradenie do kategórie je riešené triviálnym spôsobom, akým funguje drvivá väčšina malých aplikácií. Z princípu sa jedná o použitie tabuliek *firma*, ako dáta o firmách, tabuľky *kategoria*, a *firkakategoria*, určujúca vzťah medzi firmou a kategóriou.

Po zadaní kategórie pomocou parametru v url adrese, systém vyhledá hodnoty primárnych kľúčov firiem z tabuľky *firkakategoria*. Následne vyberá dáta z tabuľky *firma*, ktoré vyhovujú výsledku po hľadaní z tabuľky *firkakategoria*.

Tento spôsob je účinný pri malých počtoch riadkov v tabuľke *firkakategoria*, ako som zistil meraním záťaže prejavujúcej sa na čase generovania výstupu vzhľadom k počtu záznamov databázy. Z **grafu 5.6.1.1** je vidieť, že priebeh času závislý na počte záznamov je popísaný exponenciálnou funkciou. Ďalšou nevýhodou je triedenie firiem podľa požiadaviek užívateľa, ktoré je obmedzené len na údaje obsiahnuté v informáciách o firme.



Graf 5.6.1.1: Meranie času v závislosti na počte záznamov bez použitia tabuľky *katalogindex*

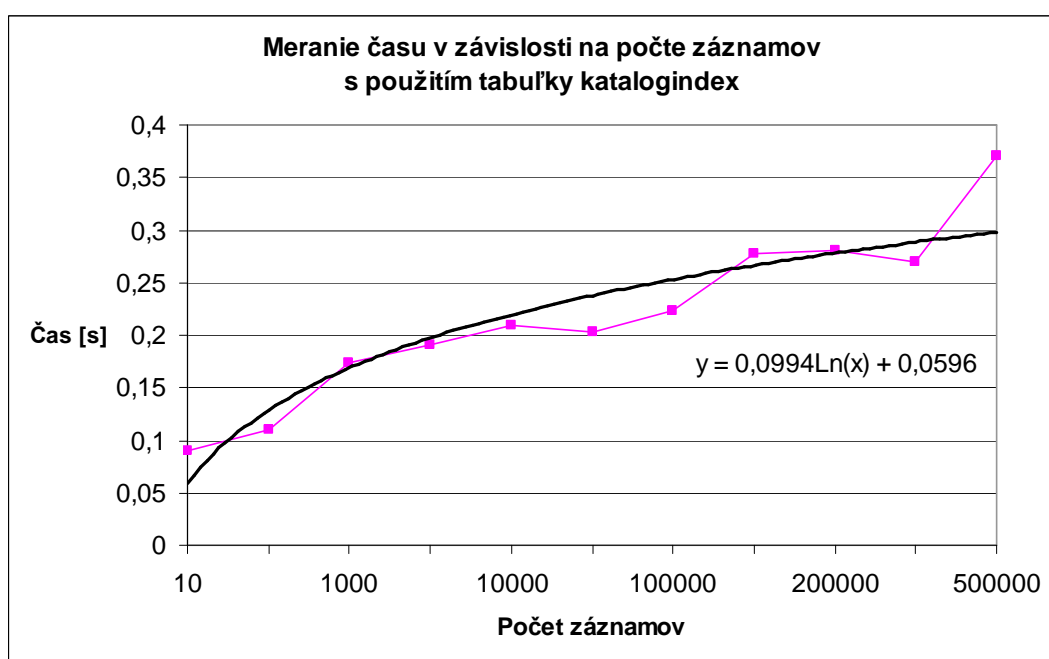
5.6.2 Štruktúra s použitím tabuľky katalogindex

V tejto štruktúre je použitá pomocná tabuľka katalogindex, ktorá uchováva dopredu uložené primárne kľúče firiem v kategóriách.

Po zadaní dotazu do systému, je vyhodnotené zaradenie, kategória, v ktorej užívateľ prehliada firmy. Systém vyberie z databáze už konkrétne primárne kľúče firiem. Tie sa následne bez potreby ďalších dotazov do databáze spracuje už pomocou PHP [4] skriptu, a z databáze vyberie už informácie konkrétnych firiem. Týmto spôsobom je zložitosť a počet dotazov minimalizovaný a výsledná záťaž optimalizovaná na minimálny potrebný čas vykonania skriptu.

Efektívita použitia štruktúry s tabuľkou katalogindex spočíva aj v možnosti rôzne triediť primárne kľúče a tým znížiť potrebu náročných skriptov v čase zadania dotazu užívateľom.

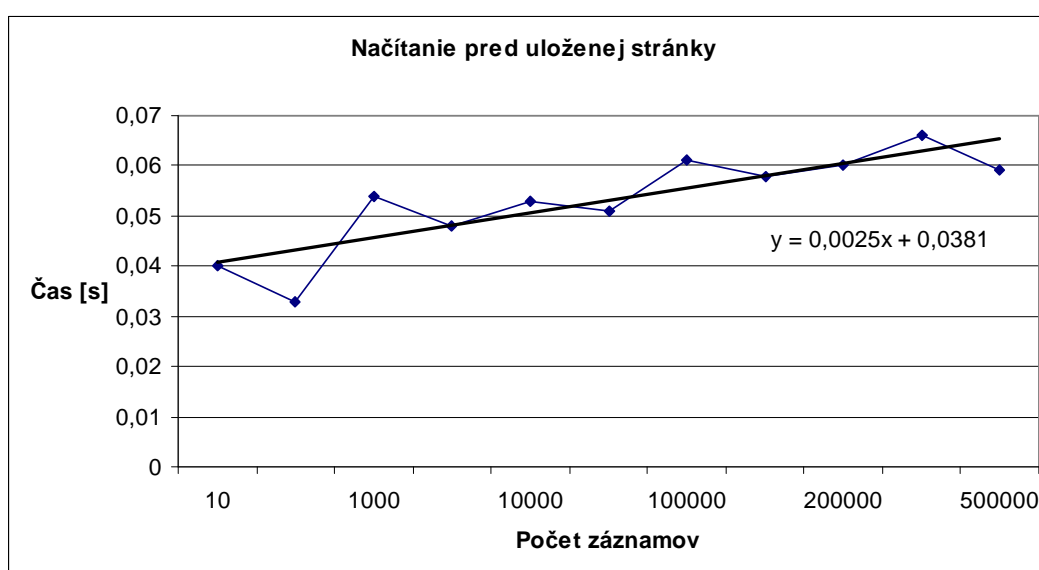
Z grafu 5.6.2.1 je vidieť, že čas vygenerovania sa pohybuje i pri počte 500 000 záznamov, pod 0,4 hranicou sekundy, priebeh závislosti ma vlastnosti logaritmickej funkcie, čím je splnená podmienka pre efektívny chod systému.



Graf 5.6.2.1: Meranie času v závislosti na počte záznamov s použitím tabuľky katalogindex

5.6.3 Pred ukladanie výstupov

Pred ukladanie výstupov, cache, slúži ako doplnenie štruktúr popísaných v kapitolách 5.6.1 a 5.6.2. Je to z dôvodu potreby vykonať úkony pre získanie výstupnej stránky prvý krát. Po vygenerovaní tejto stránky je zdrojový kód uložený na diskový priestor serveru. Po ďalšom požiadavku od užívateľa na takto spracovanú stránku systém vráti výstup z disku a tak sa prestáva správať dynamicky, ale ako statický systém. Čas potrebný na načítanie pred uloženej stránky neprekročí desatinu sekundy a je prakticky nezávislý na počte záznamov ako je vidieť na **grafe 5.6.3.1**, ktorý je popísaný lineárnou funkciou so smernicou blížiacou sa k nule.



Graf 5.6.3.1: Načítanie pred uloženej stránky

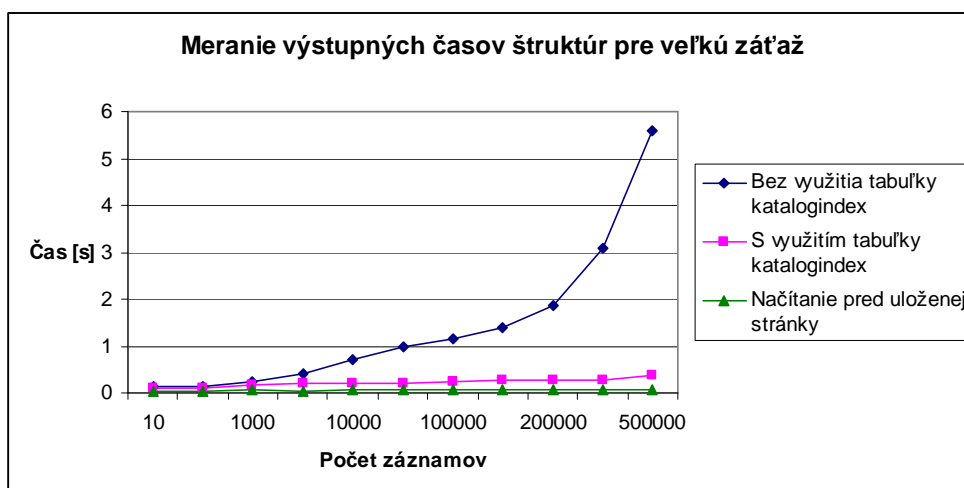
6 Záver

6.1 Dosiiahnuté výsledky

Výsledok mojej bakalárskej práce je informačný systém so štruktúrou pre vysokú záťaž v prevedení katalógu firiem. Význam má v návrhu podobných systémov k zobrazovaniu veľkého množstva elementárnych informačných celkov roztriedených do blokov a kategórií.

Základné výhody sú v efektívnom využití dostupných technológií a prepracovanej správe dát. Užívateľ tak získa praktický nástroj pre vyhľadávanie a získanie informácií o firmách a firmy získajú zákazníka s vynaložením minimálneho úsilia.

Výsledok práce je rozbor návrhov informačných systémov. Použitelným výsledkom je štruktúra s využitím indexovacej tabuľky katalogindex spolu s pred ukladaním výstupných internetových stránok, u ktorých boli overené minimálne požiadavky a záťaž systému v porovnaní s bežne používanými návrhmi. Závislosť výstupných časov na počte záznamov testovaných riešení je vynesena do **grafu 6.1.1**:



Graf 6.1.1: Meranie výstupných časov štruktúr pre veľkú záťaž

6.2 Vlastný prínos

Hlavným prínosom je aplikácia, ktorá sa využíva vo firmách s veľkým prezentujúcimi sa v komerčnej sfére internetu. Aplikácia je hodnotná v jednoduchosti používania užívateľov a optimalizácie pre vyhľadávače.

Overil som a technicky predviedol nápady ako taký informačný systém má fungovať v praxi, čo sa mi podarilo. Získal som zaujímavé poznatky a závery, ktoré využijem v ďalšej práci na podobných projektoch v danom odvetví.

V priebehu tvorby bakalárskej práce som sa bližšie zoznámil s tvorbou internetových aplikácií pomocou PHP [4], vytváraním HTML [10] stránok validných podľa dostupných štandardov a návrhom vzhľadu za pomoci kaskádových štýlov [3]. Pre uchovávanie a spracovanie dát som použil databázový systém MySQL [5].

6.3 Ďalší vývoj

Pre ďalší vývoj je zaujímavá možnosť rozšírenia systému. V mojej práci som použil jednoduchý model katalógu firiem zatriedených podľa kategórií. Tento model by sa dal použiť i v kombinácii firma a geografické zaradenie, a kategória a geografické zaradenie k získaniu lepšieho informačného prínosu pre užívateľa.

Taktiež je možné rozšírenie o triedenie firiem podľa abecedy, poprípade iných kritérií ako napríklad hodnotenie a to vzostupne alebo zostupne.

7 Zoznam použitých zdrojov

- [1] Wikipedia. Search Engine Optimization. [cit. 2008-18-05]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Search_Engine_Optimization>
- [2] W3C Recommendation. HTML 4.01 Specification. [cit. 2008-18-05]. Dostupné z WWW: <<http://www.w3.org/TR/html4/>>
- [3] PROKOP, Marek. *CSS Kaskádové styly pro webdesignéry*. [s.l.] : Mobil Media, a.s., 2003. 288 s. ISBN 80-86593-35-5.
- [4] GILMORE, W. Jason. *PHP a MySQL 5 : Kompendium znalostí pro začátečníky i profesionály*. Překlad RNDr. Jan Pokorný. 1. přeprac. vyd. Brno : Zoner Press, 2007. 864 s. ISBN 80-86815-53-6.
- [5] WELLING, Luke, THOMPSON, Laura. *MySQL : Průvodce základy databázového systému*. Přeložil Jan Gregor. Brno : CP Books, a.s., 2005. 256 s. ISBN 80-251-0671-3.
- [6] KABIR, Mohammed J. *Apache Server 2 : kompletní příručka administrátora*. Brno : Computer Press, a.s., 2004. 724 s. ISBN 80-251-0319-6.
- [7] COCKBURN, Alistair. *Use Cases : Jak efektivně modelovat aplikace*. 1. vyd. Brno : Computer Press, 2005. 264 s. ISBN 80-251-0721-3.
- [8] CodeIgniter. Model – View - Controller PHP Framework. Dostupné z WWW: <<http://www.codeigniter.com>>
- [9] *HTML 4.01 / XHTML 1.0 Reference* [online]. c1999-2007 [cit. 2007-01-01]. Dostupný z WWW: <http://www.w3schools.com/tags/ref_byfunc.asp>.
- [10] BURGET, R.: Tvorba webových stránek, Brno, přednášky 2006
- [11] NĚMEC, Robert. *SEO : optimalizace webů pro vyhledávače*. 1. vyd. Brno : Computer Press, 2006. 120 s.

8 Zoznam príloh

Príloha 1. CD so zdrojovými textami programu.