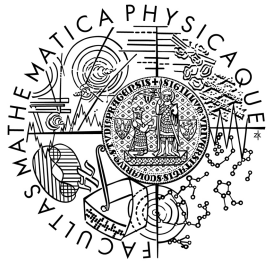
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Vítězslav Čížek

Cenový vyhledávač

Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: Mgr. Jiří Kocanda
Studijní program: Informatika, Správa počítačových systémů

2007

Poděkování
Chtěl bych poděkovat vedoucímu svého ročníkového projektu a bakalářské práce Mgr. Jiřímu Kocandovi za jeho podněty a připomínky. Dále Martinu Srkalovi, za jeho rady ohledně typografickém systému LaTeX a Kateřině Dufkové za poznámky k designu aplikace.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 22.5.2007 Vítězslav Čížek

Obsah

Název práce: Cenový vyhledávač
Autor: Vítězslav Čížek
Katedra (ústav): Ústav formální a aplikované lingvistiky
Vedoucí bakalářské práce: Mgr. Jiří Kocanda
E-mail vedoucího: kocanda@atlas.cz

Abstrakt: Předložená práce se zabývá problematikou zpracování velkého množství informací na Internetu. Tato úloha nabývá v současnosti na významu spolu se vzrůstajícím množstvím dostupných informací. Cílem bylo vytvořit aplikaci, která by tento problém řešila pro jednu konkrétní často využívanou oblast a zjednodušila tak uživateli náročnou úlohu analýzy dat. Aplikace je určena pro vyhledávání v datech vyskytujících se v databázích internetových obchodů. Použité technologie byly zvoleny tak, aby maximálně usnadňovaly použití koncovým uživatelům a zároveň umožňovaly nezávislost programu na použitém operačním systému.

Klíčová slova: web, vyhledávání, webové obchody, e-commerce

Title: Price Explorer
Author: Vítězslav Čížek
Department: Institute of Formal and Applied Linguistics
Supervisor: Mgr. Jiří Kocanda
Supervisor's e-mail address: kocanda@atlas.cz

Abstract: The present work is focused on the problem of processing huge amount of information on the Internet. This task's importance is constantly growing in present days along with the increasing amount of available information. The goal was to develop an application that would solve this problem for one concrete, frequently used area and simplify the user the time-consuming task of data analysis. The program is designed for searching data occurring in on-line shop databases. The technologies used in this application were chosen in a way to satisfy two conditions: to facilitate usage to the end-user and at the same time to enable program's platform independence.

Keywords: web, search engine, on-line shop, e-commerce

Kapitola 1

Úvod

Problematika zpracování velkého množství informací nabývá v současnosti na významu spolu se vzrůstajícím množstvím informací dostupných na Internetu. Z této široké oblasti jsem se rozhodl ve své bakalářské práci zaměřit na vyhledávání v datech vyskytujících se v databázích internetových obchodů. Na tomto poli se prozatím nevyskytuje mnoho aplikací umožňující zpracovávat data napříč internetovými obchody. Cílem mé práce je vytvořit aplikaci, která by tento problém řešila a zjednodušila tak uživateli náročnou úlohu analýzy tohoto typu dat.

Vytvořenou aplikaci jsem pojmenoval Cenové vyhledávání. Program slouží k současnému vyhledávání ve více webových obchodech a zpracované výsledky zobrazuje uživateli. Přihlášeným uživatelům jsou navíc k dispozici další užitečné funkce, které nejsou přístupné uživatelům bez registrace. Jde například o možnost prohlížet historii, opakovat minulá hledání či monitorovat vybrané kusy zboží. O změně ceny monitorovaného zboží je uživatel informován při dalším přihlášení nebo e-mailem. Administrátor aplikace má přístup do speciálního rozhraní, kde může spravovat registrované uživatele a prohlížet statistiky práce uživatelů.

Cenové vyhledávání je webová aplikace vyvinutá pro umístění na serveru. Výhodou tohoto druhu aplikace jsou především minimální nároky na uživatele a jeho systém, protože k používání takovéto aplikace není nutná ani instalace, ani žádný dodatečný software. Jediným požadavkem na uživatele je připojení k internetu a libovolný webový prohlížeč. Použité technologie byly zvoleny tak, aby maximálně usnadňovaly použití koncovým uživatelům

a zároveň umožňovaly nezávislost programu na použitém operačním systému.

Kapitola 2

Technologie použité k vývoji aplikace

Následuje podrobný popis technologií, které byly použity k vývoji aplikace, jejich srovnání s obdobnými řešeními a důvody k jejich zvolení.

2.1 Java Servlety a JSP

Java Servlety a JSP[5] (Java Server Pages) jsou technologie od společnosti Sun Microsystems. Slouží mimo jiné k vytváření dynamických webových stránek na straně serveru. Aplikace v JSP a servlety jsou vytvářeny v jazyku Java, který je přenositelný mezi operačními systémy, neboť je překládán do bytecode. Ten je posléze interpretován Java Virtual Machine. Servlety jsou standardní třídy jazyka Java odvozené od třídy HttpServlet. JSP je java kód včleněný do statické webové stránky prováděný serverem. Ve skutečnosti není pro server mezi JSP a servlety žádný rozdíl, neboť JSP jsou interně reprezentovány jako servlety.
Nyní uvedu některé jejich výhody oproti jiným technologiím.

CGI

Common Gateway Interface je standard pro spouštění externích programů uvnitř webového serveru. Ve srovnání se Servlety má však tento protokol několik nevýhod. Při spuštění více kopií servletu je v paměti pouze jedna instance java třídy a jednotlivé přístupy jsou obsluhovány lightweight vlákny,

zatímco u CGI je pro každý HTPP požadavek (request) spuštěn jeden systémový proces, což má za následek větší zatížení serveru. Servlet navíc oproti CGI programům zůstává v paměti a nemusí být při dalším přístupu znovu načítán.

ASP

Active Server Pages je technologie firmy Microsoft umožňující dynamické vytváření webových stránek na straně serveru. Dynamický obsah se generuje zpravidla v jazyce VBScript, který je méně vhodný na vytváření komplexních aplikací než jazyk java. Nevýhodou je také portabilita. Na rozdíl od servletů/JSP jsou ASP vázána pouze na platformu Microsoft Windows a webový server IIS.

PHP

PHP je skriptovací jazyk se syntaxí podobnou jazykům C nebo Perl. V současné době patří mezi nejpopulárnější řešení na serveru, zejména pro svou jednoduchost a otevřenost. Mezi jeho nevýhody patří dynamické typování a absence nativní podpory UNICODE a jmenných prostorů. Vývoj rozsáhlejších projektů je v něm složitější na správu.

SSI

Server side includes je, jak už vyplývá přímo z názvu, technologie pro vkládání kódu do statických stránek na straně serveru. Není však vhodná pro složité operace jako je například zpracovávání uživatelských dat nebo spolupráce s databází. Další nevýhodou je rychlost, protože server musí procházet kód stránky při každém přístupu.

2.2 Apache Tomcat

Webový server Apache Tomcat[4] implementuje Servlety a JSP. Je vyvíjen Apache Software Foundation, která ho poskytuje zdarma i se zdrojovými kódy pod Apache licencí. Není platformově závislý, což z něj činí oblíbené řešení pro hostování Java aplikací.

2.3 MySQL

MySQL[3] je multiplatformní relační databáze šířená pod dvojí licencí. Je k dispozici buď zdarma pod licencí GNU/GPL, nebo jako MySQL Enterprise pro komerční užívání. Dnes patří mezi nejčastěji nasazované databáze na webových serverech. Vyznačuje se dobrou stabilitou, pravidelným vydáváním nových verzí a bezpečnostních záplat.

Kapitola 3

Instalace aplikace

3.1 Požadavky na instalaci aplikace

Aplikace Cenový vyhledávač je vytvořena v Java Servletech a JSP pro Javu verze 5 a využívá databázi MySQL. Proto ji lze umístit pouze na takový server, který tyto technologie podporuje. Pro provoz aplikace je nezbytný webový server podporující Java Servlety a JSP. K vývoji byl použit server Apache Tomcat, který je šířen zdarma, není závislý na jedné platformě a vždy podporuje nejnovější verze servletů a JSP. K provozu je ale samozřejmě možné použít libovolný jiný java server co splňuje zadané požadavky, například ServletExec nebo JRun. Pro ukládání dat používá aplikace databázi MySQL, která je též k dispozici zdarma a funguje na více platformách. Vývoj byl prováděn na verzi 5, ale program byl úspěšně testován i na starších verzích. Co se týče Java Virtual Machine, je pro provoz striktně vyžadována Java verze 5, jinak je nutná rekompilace celé aplikace.

3.2 Doporučená konfigurace

Softwarové požadavky

Doporučuje se Java verze 5.0, databáze MySQL verze 5.37 nebo vyšší a webový server Apache Tomcat alespoň ve verzi 5.5.20.

Hardwarové požadavky

Hardwarové požadavky na instalaci se drobně liší v závislosti na použitém operačním systému. Nicméně zde uvedené hodnoty by měly být dostačující pro všechny operační systémy.
Java JRE vyžaduje na instalaci alespon 100MB volného diskového prostoru a na běh 64MB operační paměti. MySQL zabírá 30MB a Tomcat 20MB místa na disku, obě aplikace se vejdou do 16MB paměti. Za předpokladu, že s aplikací bude pracovat 500 uživatelů, nebude velikost databáze přesahovat 50MB. V součtu se tedy doporučuje alespoň 200MB volného místa na disku a 128MB operační paměti.

Nastavení aplikace

Velký vliv na výkonnost aplikace mají nastavení v *konfiguračním souboru hledání/sledování* (viz kap. 5.1). Hodnota *obnovovat* se doporučuje nastavit na více než 24 hodin, aby nedocházelo ke zbytečnému stahování dat, *mazat* nastavit na hodnotu okolo jednoho týdne a *timeout* přibližne na 10 vteřin.

3.3 Instalace na systémech UNIX

Aplikace je dodávána s instalačním skriptem *unix-install.sh*, který usnadňuje instalaci na systémech typu UNIX. Tento skript má formát otevřeného textu, takže je možné ho kdykoli upravit aby svojí funkcí vyhovoval uživateli. Je napsán pro Bourne Shell, který je standardně dodáván s UNIXovými systémy, ale může být interpretován i s ním kompatibilními příkazovými interprety, jako Korn Shell nebo populární BASH. Skript je interaktivní, ptá se uživatele na informace potřebné k instalaci aplikace. Nejprve je nutné mu zadat adresář kam se bude aplikace instalovat. Ten bývá zpravidla umístěn v adresářovém stromu webového serveru na kterém bude program běžet, neboť je nezbytné aby server měl přístup k adresáři aplikace. Instalační skript se dále zeptá na cestu ke *centrálním konfiguračnímu souboru*, který je načítán při startu aplikace a obsahuje informace o dalších konfiguračních souborech. Poté je třeba zadat adresu databázového serveru a heslo pro připojení, aby bylo možné vytvořit databázi používanou aplikací k uložení dat. Závěrem zadá uživatel heslo, které bude nastaveno pro administrátorský účet. Po zadání těchto údajů instalační program provede potřebná nastavení, vytvoří databázi a do cílového adresáře nainstaluje aplikaci.

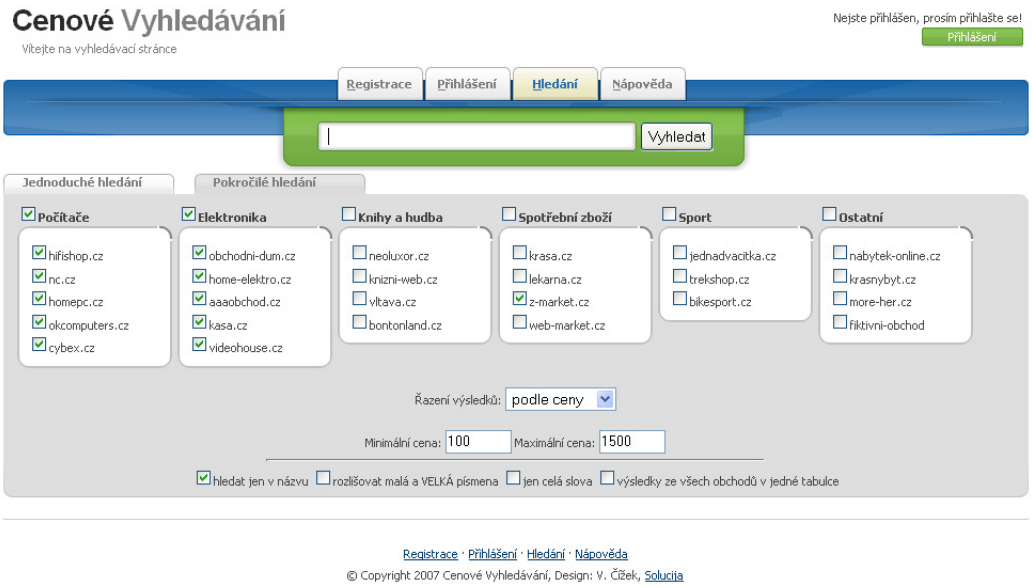
Na ostatních operačních systémech je třeba provést uvedené kroky manuálně.

3.4 Instalace na ostatních systémech

Adresář s aplikací je nutné nakopírovat do adresáře serveru, kde jsou umístěny servletové webové aplikace. Ten je závislý na konkrétním použitém serveru. Například pro server Apache Tomcat jde o podadresář webapps v kořenovém adresáři Tomcatu. Dále je nutné v souboru *web.xml*, který popisuje mapování servletů, v inicializačním parametru servletu *daemon* uvést cestu k *centrálnímu konfiguračnímu souboru*. Dále je nutné vytvořit databázi do které bude aplikace ukládat svoje data. K tomu lze použít s aplikací dodávaný SQL skript *empty.sql*. Před jeho spuštěním v databázi je však vhodné ho upravit - změnit heslo pro administrátorský účet aplikace, jinak bude použito heslo přednastavené. Skript vytvoří prázdnou databázi neobsahující žádná data a účet administrátora. Nakonec je ještě nutné restartovat webový server, aby došlo k načtení aplikace.

Kapitola 4

Uživatelská dokumentace



Obrázek 4.1: Screenshot stránky rozšířeného vyhledávání

Aplikace Cenové vyhledávání byla vytvořena, aby pomohla zjednodušit a zrychlit orientaci v nabídkách internetových obchodů. V této kapitole je popsána práce s aplikací z pohledu běžného uživatele. Jsou zde nastíněny důvody k registraci a popsány všechny funkce programu, které může využít přihlášený nebo nepřihlášený návštěvník stránky.

4.1 Registrace a přihlášení

Aplikaci Cenové vyhledávání lze používat bez registrace a přihlášení. Po přihlášení získá uživatel další výhody, které by nebylo možno realizovat bez jednoznačné identifikace. Jde především o možnost sledovat zboží a jeho cenu a obdržet e-mail automaticky v případě, že došlo k nějaké změně. Další možností je zobrazení minulých hledání uživatele.

Registrace i přihlášení do systému jsou řešeny standardně. Před prvním přihlášením je třeba provést registraci, jejíž formulář se zobrazí, pokud uživatel z úvodní stránky zvolí odkaz *Registrace* v dolním textovém menu. V tomto formuláři uživatel zvolí své nové uživatelské jméno, heslo a e-mailovou adresu, na kterou si přeje nechat zasílat e-maily s výsledky hledání. Uživatelské jméno ani heslo nesmí obsahovat znaky s čárkami nebo háčky, může obsahovat malá i velká písmena anglické abecedy, číslice a interpunkční znaky. Pokud není zvolené uživatelské jméno zaregistrováno jiným uživatelem, je registrace potvrzena, v opačném případě si musí uživatel vybrat jiné. Uživatelé se registrují pouze jednou, případnou změnu hesla nebo e-mailové adresy lze provést kdykoli později po přihlášení zvolením odkazu *Opravit údaje*.

4.2 Funkce dostupné všem uživatelům

Po zadání adresy Cenového vyhledávání do internetového prohlížeče se všem uživatelům zobrazí stejná úvodní stránka. Uživateli poskytuje několik možností - přihlášení a registraci zmíněnou v předchozím odstavci, zobrazení nápovědy a vyhledávání zboží bez přihlášení.

Vyhledávání a jeho parametry

Zahájení vyhledávání zboží je velmi jednoduché. Do textového okna v centrální části stránky je třeba zadat text, který co nejpřesněji vystihuje hledané zboží, a dále stačí zaškrtnout kde vyhledávat a vše potvrdit stiskem tlačítka *Vyhledat*. Vyhledávací stránka existuje ve dvou variantách - jednoduché a pokročilé. V jednoduché variantě je možné vybírat pouze kategorie obchodů, ve kterých se bude vyhledávat, a nejsou k dispozici přepínače. V pokročilé variantě (viz obrázek 4) je možné kromě zaškrtnutí celé kategorie zvolit i konkrétní obchody, ve kterých se má hledat. V zaškrtávání či odškrtávání

velkého množství obchodů pomohou tlačítka u kategorií, která zároveň zaškrtnou či odškrtnou všechny obchody v dané kategorii.

V obou variantách je k dispozici vysunovací nabídka pro výběr kritéria řazení výsledků. V pokročilé variantě navíc pro zadání doplňujících parametrů hledání slouží další zaškrtávací políčka - *hledat jen v názvu*, *rozlišovat malá a VELKÁ písmena*, *celá slova* a *výsledky ze všech obchodů v jedné tabulce*. Zaškrtnutím první volby omezíte množinu nalezených zboží na ty, které mají hledaný výraz v názvu. Pokud ho necháte nezaškrtlé, bude výraz vyhledáván i v popisu zboží, což může zvláště u obecnějších a často se vyskytujících slov vést k velkému množství výsledků, které s hledaným zbožím souvisí pouze volně. Zvolením volby *rozlišovat malá a VELKÁ písmena* zajistíte, že při vyhledávání bude brán ohled na velká a malá písmena. Zaškrtnutím volby *celá slova* zajistíte, že vyhledávána budou pouze celá slova a nikoli jejich části. Vybráním volby *výsledky ze všech obchodů v jedné tabulce* zajistíte výpis všech výsledků do jediné tabulky, pokud volba zůstane nezaškrtnuta, bude ve výsledku pro každý obchod samostatná tabulka.

Ještě je třeba se zmínit podrobněji o samotném vyhledávacím řetězci. Ten musí být alespoň 3 znaky dlouhý (tuto podmínku vyžadují některé podporované obchody), může obsahovat české znaky a může jít o více slov.

Vyhledávací algoritmus

Ve výchozím nastavení vyhledávání se hledá v názvu a popisu produktu, jsou hledány i části slov, nejsou rozlišována malá a velká písmena, není nastaveno žádné cenové omezení a výsledky jsou řazeny vzestupně podle ceny.

Pro jeden konkrétní obchod probíhá vyhledávání následujícím způsobem. Všechna slova zadaná do vyhledávacího pole jsou nejprve převedena do kódování, které daný obchod podporuje. Pak je ze všech těchto slov vytvořen jeden vyhledávací dotaz pro obchod, jednotlivá slova jsou spojena operátorem konjunkce (ve výsledku se musí objevit všechna). Tento vyhledávací dotaz je odeslán obchodu. Proces se opakuje pro každý zvolený obchod.

Po obdržení odpovědí z obchodu jsou na dílčí výsledky aplikovány další omezující podmínky. Nalezené produkty jsou seřazeny podle zvoleného kritéria a zobrazeny uživateli.

4.3 Funkce dostupné přihlášeným uživatelům

Po přihlášení je uživateli zobrazena úvodní obrazovka pro přihlášené uživatele, z které jsou dostupné další funkce. Jednou z nich je změna registračních údajů dostupná pod odkazem *Opravit údaje*. Přechod na vyhledávací stránku je možný zvolením odkazu *Hledání*.

Sledování zboží

Po zvolení odkazu *Sledování* z úvodní stránky po přihlášení je uživateli zobrazena stránka se stavem každého zboží, které dal v minulosti sledovat. Pokud chce přihlášený uživatel dát sledovat nové zboží, stačí když na stránce s výsledky vyhledávání stiskne tlačítko *Sledovat* u zboží, které si přeje monitorovat. Na případnou změnu stavu bude uživatel upozorněn zasláním emailu na adresu zadanou při registraci a upozorněním na úvodní stránce po svém dalším přihlášení.

Historie hledání

Po zvolení odkazu *Historie hledání* z úvodní stránky po přihlášení je uživateli zobrazena stránka s tabulkou shrnující co, kdy a v kterých obchodech uživatel v minulosti vyhledával. Historie je zobrazena jako hypertextové odkazy, vedoucí na výsledky minulých vyhledávání.

Kapitola 5

Administrátorská dokumentace

Součástí aplikace Cenové vyhledávání je administrátorské rozhraní. Tato kapitola má pomoci administrátorovi v práci s tímto rozhraním. Obsahuje také popis formátu konfiguračních souborů aplikace. Uživatelské jméno pro přihlášení do administrátorského rozhraní je *admin*.

5.1 Konfigurace aplikace

Aplikace je dodávána s počáteční konfigurací, která podporuje 25 obchodů z České republiky, vybraných záměrně z různých oblastí. Při vývoji aplikace bylo ovšem pamatováno na fakt, že provozovatel může chtít tuto počáteční množinu změnit, rozšířit, či se zaměřit jen na určitou část trhu, proto mu byla ponechána možnost nakonfigurovat aplikaci podle svých konkrétních požadavků. Tuto konfiguraci je možné provést úpravou konfiguračních souborů obchodů a parsování, jejichž význam a formát bude podrobně zmíněn dále.

Všechny konfigurační soubory jsou v běžném textovém formátu v kódování *utf-8*, a lze je tedy snadno editovat libovolným textovým editorem. V souborech je možné používat komentáře, komentářový řádek musí začínat znakem lomítko (/) a aplikací bude zcela ignorován. Prázdné řádky jsou taktéž ignorovány.

Konfigurační soubor hledání/sledování

V tomto konfiguračním souboru jsou umístěny parametry programu související s dobou odpovědi programu a cachováním. V souboru se vyskytují řádky tvaru:

```
nazev-hodnoty=hodnota
```

Následující tabulka shrnuje možné hodnoty:

Název hodnoty	Význam
obnovovat	Doba v minutách, po které budou obnovovány záznamy v cachi
mazat	Doba v minutách, po které budou mazány záznamy v cachi
timeout	Doba v sekundách, do které bude aplikace čekat na odpovědi od obchodů

Tyto hodnoty lze prohlížet a měnit i po přihlášení administrátora do webového rozhraní, viz podkapitola 5.2. Největší efekt má vhodné nastavení hodnoty *timeout*, protože odpovídá maximální době, kterou budou uživatelé čekat na odpověď aplikace na výsledky hledání. Její příliš nízká hodnota může zapříčinit, že téměř žádné obchody nestihnou zaslat svůj výsledek a uživatel pak neobdrží uspokojivé výsledky.

Konfigurační soubor obchodů

Konfigurační soubor obchodů obsahuje informace nutné k sestavení vyhledávacích dotazů pro jednotlivé podporované obchody. Každý neprázdný nekomentářový řádek v tomto souboru je považován za záznam s informacemi o jednom podporovaném obchodu. Na tomto řádku se musí vyskytovat přesně šest hodnot oddělených navzájem libovolným počtem bílých znaků (mezer a tabelátorů). Formát každého řádku je následující:

```
jmeno-obchodu prefix-dotazu doplneni-pro-vice-stranek \
typ-strankovani prefix-infostranek kodovani
```

Hodnota *jmeno-obchodu* udává jméno obchodu, které bude používáno k identifikaci obchodu v aplikaci a také ve vyhledávacím formuláři aplikace přístupném uživatelům. Hodnota *prefix-dotazu* obsahuje asi nejdůležitější

informaci o obchodu, totiž prefix vyhledávacího dotazu pro tento obchod. Musí jít o platnou www adresu vyhledávací stránky obchodu bez prefixu *http://*, ve které musí být uvedeny i hodnoty všech GET proměnných nutných k úspěšnému vyhledávání. Musí končit názvem proměnné, sloužící pro zadání vyhledávaného výrazu, a znakem rovnítko (=). Další dvě hodnoty - *doplneni-pro-vice-stranek* a *typ-strankovani* slouží pro kontrolu stránkování u obchodů, které je používají. První hodnota obsahuje jméno GET proměnné udávající počet stránek v daném obchodu a musí končit znakem rovnítko. Druhá hodnota je ještě dále strukturována, její formát je *start/posun*. *Start* je číselná hodnota udávající, zda obchod počítá stránky od nuly nebo od jedničky. *Posun* je číselná hodnota udávající, kolik je třeba přičíst k hodnotě stránkovací GET proměnné pro posun na další stránku. Pokud obchod stránkování nepodporuje, musí být hodnota *typ-strankovani 1/0* a konvence je uvádět *doplneni-pro-vice-stranek xxx=*. Hodnota *prefix-infostranek* udává začátek www adresy stránek s informacemi o zboží pro daný obchod, opět bez *http://*. Pokud obchod neposkytuje stránky s informacemi o jednotlivých výrobcích, musí být hodnotou *prefix-infostranek* znak pomlčka (-). Poslední hodnota *kodovani* udává kódování stránek, které obchod používá. Množina povolených hodnot přesně odpovídá množině kódování, které podporuje implementace Javy na serveru, na kterém aplikace běží.

Příklad obchodu, který používá běžné stránkování s čísly stránek od jedničky:

```
nc www.nc.cz/search/default.asp?SearchType=All&EXPS= \
PgID= 1/1 www.nc.cz/DetailPage.asp?DPG= CP1250
```

Příklad obchodu, který používá stránkování udávající index zboží, od kterého bude vypisováno, začínající od nuly:

```
more-her www.more-her.cz/index.php?stylVypisu=kat&search= \
start= 0/10 www.more-her.cz/ CP1250
```

Příklad obchodu, který nepoužívá stránkování a neposkytuje stránky s informacemi o zboží:

```
z-market www.z-market.cz/browser.php3?category=search&text= \
xxx= 1/0 - 8859_2
```

Konfigurační soubor parsování

Konfigurační soubor parsování obsahuje položky nutné k získání potřebných informací ze stránek se seznamem nalezeného zboží a stránek s doplňujícími informacemi o výrobcích pro jednotlivé podporované obchody.

Konfigurační soubor využívá k parsování vstupní stránky stejné regulární výrazy jako Java, konkrétně její package *java.util.regex*[2]. Jejich znalost je nutná k pochopení následujícího textu. Pro některé často aplikované regulární výrazy ovšem aplikace z důvodů zvýšení rychlosti používá pouze jejich podmnožinu, kdy jediným wild-cardem je posloupnost znaků *(.*?)*, která má význam namatchování nejmenšího nutného (případně i nulového) počtu znaků tak, aby byl uspokojen výraz.

Každý neprázdný nekomentářový řádek v konfiguračním souboru parsování je považován za záznam s informacemi o jednom podporovaném obchodu. Na tomto řádku se musí vyskytovat přesně 22 hodnot oddělených navzájem řetězcem `__` (dvě podtržítka za sebou). Pouze poslední z hodnot se týká parsování stránek s doplňujícími informacemi o zboží, ostatní se týkají parsování stránek se seznamem nalezeného zboží. Formát každého řádku je následující:

```
jmeno-obchodu__top-level-regexp__skupina-stranky__ \
skupina-zbozi__regexp-cislo-stranky__regexp-zbozi__odkaz__ \
nazev__popis__skladem__cena__infostranka-regexp
```

Položka *jmeno-obchodu* má stejný význam jako u konfiguračního souboru obchodů. Položka *top-level-regexp* je regulární výraz používající pouze výše zmíněnou podmnožinu speciálních znaků, který rozdělí html stránku s výsledky vyhledávání na část čísel stránek a část se seznamem nalezeného zboží, které budou dále parsovány. Hodnoty *skupina-stranky* a *skupina-zbozi* musí být číselné hodnoty udávající, pod kterou závorkou *top-level-regexp* je namatchována část s čísly stránek, resp. část se seznamem zboží. Hodnotami *regexp-cislo-stranky* a *regexp-zbozi* jsou opět regulární výrazy, které budou opakovaně použity na příslušné části stránky k postupnému získání všech čísel dostupných stránek (pokud nelze namatchovat přímo celkový počet stran) a všech nalezených zboží. Výsledek regulárního výrazu *regexp-cislo-stranky* musí být namatchován pod první závorku. V případě *regexp-zbozi* jde opět o výraz z výše zmíněné podmnožiny regulárních výrazů.

Následuje pět položek - *odkaz, nazev, popis, skladem* a *cena* - pro zjištění vlastností zboží z části stránky věnované jednomu zboží, která je získána použitím regulárního výrazu *regexp-zbozi*. Všechny tyto položky jsou dále strukturované následujícím způsobem. Jejich formát je:

```
skupina-v-regexpu-zbozi__skupina-v-regexpu-vlastnosti__ \
regexp-vlastnosti
```

Hodnotou položky *skupina-v-regexpu-zbozi* je číslo závorky, pod kterou je v regulárním výrazu *regexp-zbozi* namatchován text dané vlastnosti. Protože ale nemusí být vždy možné přímo v regulárním výrazu *regexp-zbozi* získat přesně to, co je třeba, pro dočištění textu vlastnosti (např. ceny od znaku oddělujících tisíce) je ještě možné aplikovat speciální regulární výraz pro vlastnost, který udává hodnota položky *regexp-vlastnosti*. V tomto výrazu lze použít plné regulární výrazy podle normy Javy. Pod kterou závorkou je v tomto regulárním výrazu namatchován výsledek pak říká hodnota položky *skupina-v-regexpu-vlastnosti*. Je třeba zmínit, že *regexp-vlastnosti* je aplikován opakovaně, dokud něco namatchuje, a jeho výsledky jsou řetězeny. Pokud není dodatečný regulární výraz potřeba, je třeba uvést *.** jako hodnotu *regexp-vlastnosti* a 0 jako hodnotu *skupina-v-regexpu-vlastnosti*. Pokud obchod některou z vlastností u svých zboží neuvádí, je třeba uvést u všech třech podpoložek této vlastnosti znak pomlčka (-).

Poslední položka *infostranka-regexp* udává regulární výraz, který bude použit pro parsování informační stránky o zboží z daného obchodu. Tento regulární výraz (podle normy Javy) má za úkol vyříznout z celé stránky kus, který je při sledování zboží třeba kontrolovat, zda se změnil. Výsledek tohoto regulárního výrazu musí být namatchován pod první závorku.

Mezi oběma konfiguračními soubory platí následující vztah - obchod, pro který existuje záznam v jednom z nich, musí mít záznam i v druhém konfiguračním souboru.

Příklad:

```
obchodni-dum__<div class="find-pager">(.*?)</span>(.*?) \
class="find-results">(.*?)<div class="find-pager"__1__3__ \
<a href="find(.*?)>(\d+) ?</a>__class="show-box-name" \
(.*?)<a href="(.*?)"(.*?)>(.*?)<(.*?) \
class="how-box-desc">(.*?)<(.*?)javascript(.*?) \
```

```
<td class="show-box-price">(.*?)K__3__0__ \
.*__5__0__.*__7__0__.*__9__2__pojem(.*?) \
alt="(.+?)"__10__0__\d+|\.|,__ \
class="detail-ceny-box"(.*?)</div>
```

Na závěr několik tipů, jak zvolit jednotlivé regulární výrazy. Pro internetové obchody, které používají tagy class nebo id je často vhodné zaměřit se na právě tyto tagy a orientovat se v html kódu podle nich. Pro obchody, které je nepoužívají, je většinou možné orientovat se podle samotného textu uvnitř tagů, nejčastěji podle nadpisů. Pro získání čísel stránek u obchodu, který uvádí seznam čísel stránek ve formi odkazu na jednotlivé stránky, je možné použít regulární výraz *>(\d+)<*. Pro získání ceny lze často jako *regexp-vlastnosti* použít výraz *\d+|\.* , resp. *\d|*, pokud obchod používá desetinnou tečku, resp. desetinnou čárku.

Centrální konfigurační soubor

V tomto konfiguračním souboru jsou umístěny parametry programu související s instalací aplikace na webový server a důležité přihlašovací údaje. V souboru se vyskytují řádky tvaru:

```
nazev-hodnoty=hodnota
```

Následující tabulka shrnuje povinné hodnoty:

Název hodnoty	Význam
config_admin	Název konfiguračního souboru hledání/sledování na serveru, včetně cesty
config_obchody	Název konfiguračního souboru obchodů na serveru, včetně cesty
config_parser	Název konfiguračního souboru parsování na serveru, včetně cesty
base_url	Url databáze, ke které se připojovat (ve formátu *server/jméno_databáze*)
base_login	Uživatelské jméno pro připojení k databázi
base_pass	Heslo pro připojení k databázi
admin_pass	Heslo pro přihlášení do administrátorského rozhraní (pro login *admin*)

Tyto hodnoty lze prohlížet a měnit pouze v tomto souboru. Jeho umístění je parametrem souboru *web.xml*, jehož relevantní část může vypadat například takto:

```
<servlet>
    <servlet-name>daemon</servlet-name>
    <servlet-class>PROJEKT.daemon</servlet-class>
    <init-param>
        <param-name>konf</param-name>
        <param-value>
            /opt/tomcat/webapps/PROJEKT/konfigurace.txt
        </param-value>
    </init-param>
    <load-on-startup>0</load-on-startup>
</servlet>
```

5.2 Funkce dostupné administrátorovi z webového rozhraní

Po zadání uživatelského jména admin a platného hesla k tomuto účtu na přihlašovací stránce do aplikace Cenové vyhledávání je uživatel přesměrován na úvodní stránku administrátorského rozhraní. Zde jsou k dispozici statistiky hledání a sledování, které byly aplikací provedeny.

Statistiky hledání a sledování

Administrátorské rozhraní poskytuje informace z následujících čtyř oblastí - sledování, hledání, nejčastěji hledané výrazy a nejčastěji využívané obchody. Pro každou z oblastí je na úvodní stránce administrátorského rozhraní k dispozici odkaz na stránku s podrobnějšími informacemi, na úvodní stránce jsou uvedeny jen souhrnné informace - celkový počet uskutečněných operací pro oblast hledání a sledování, nejčastější vyhledávaný výraz a obchod, ve kterém bylo vyhledáváno nejčastěji.

Po zvolení odkazu *Sledování celkem* bude administrátorovi zobrazena stránka se seznamem každého zboží, které dal některý z registrovaných uživatelů sledovat. Po zvolení odkazu *Hledání celkem* bude zobrazena stránka s tabulkou s údaji co, kdy a ve kterých obchodech bylo uživateli vyhledáváno. Po zvolení odkazu *Nejčastěji hledáno* bude zobrazena stránka s tabulkou uvádějící jednotlivé hledané výrazy spolu s četnostmi jejich vyhledávání, seřazené od nejčastěji hledaného. Po zvolení odkazu *Nejčastější obchod* bude zobrazena stránka s tabulkou uvádějící u jednotlivých obchodů četnosti hledání v nich, opět seřazené od nejčastěji využívaného obchodu.

Do statistik hledání jsou započítána všechna hledání zadaná přihlášenými i nepřihlášenými uživateli. Do statistik sledování se dostanou jen sledování zadaná přihlášenými uživateli, protože nepřihlášení nemají vůbec možnost sledování využít.

Změna nastavení

Změnit nastavení zmíněná v podkapitole 5.1 je možné i z administrátorského rozhraní. Stačí zvolit odkaz *Změnit nastavení* z úvodní stránky administrátorského rozhraní, ve formuláři na další stránce lze vyplnit příslušné hodnoty.

Jsou k dispozici i volby pro vyprázdnění cache (dočasné paměti s výsledky hledání), či pro zastavení všech sledování všech uživatelů.

Správa uživatelů

Po zvolení odkazu *Počet uživatelů* z úvodní stránky administrátorského rozhraní je zobrazen přehled všech registrovaných uživatelů. U každého z nich je uveden údaj o celkovém počtu hledání, které zadal, a počtu jeho aktivních sledování. Stisknutím tlačítka *Vymazat* lze zrušit registraci libovolného uživatele.

Kapitola 6

Programátorská dokumentace

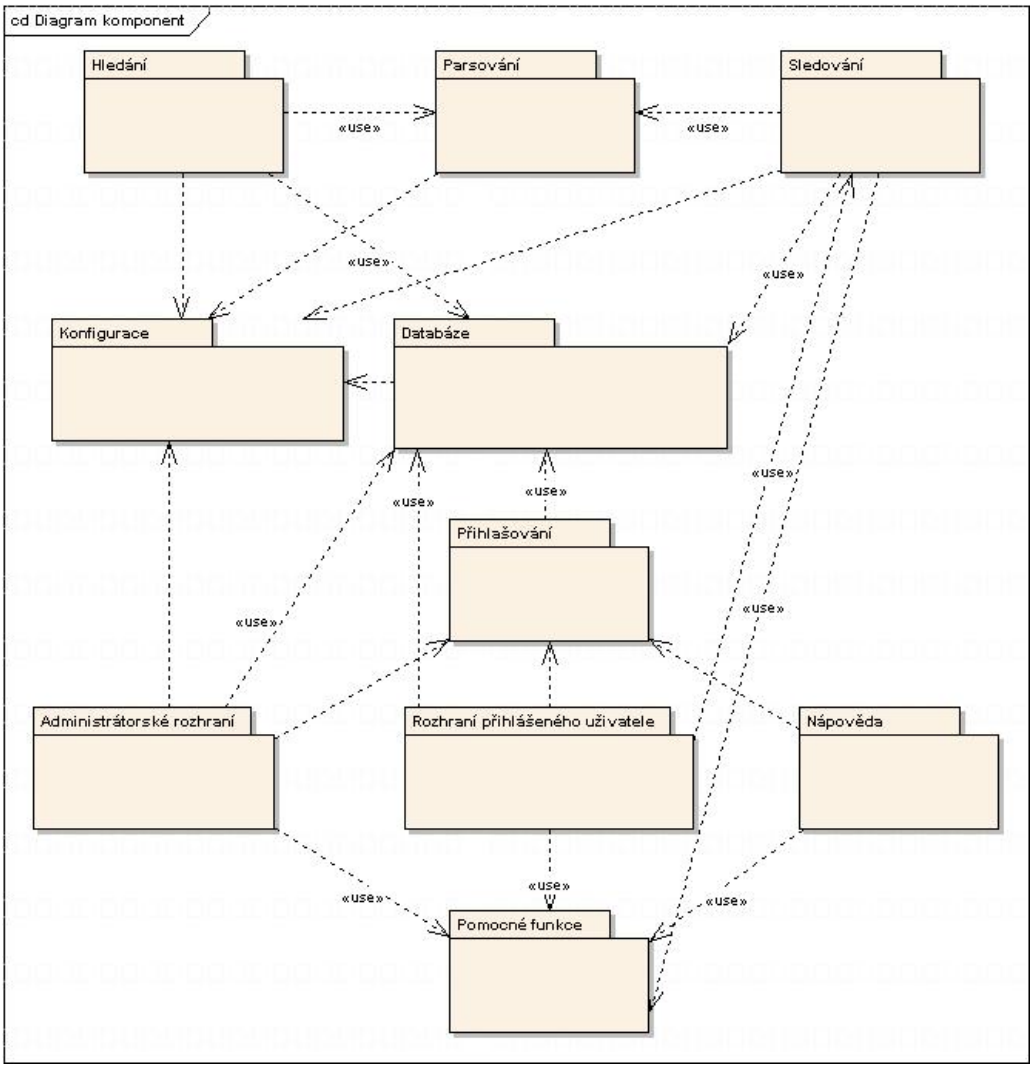
Tato kapitola obsahuje programátorskou dokumentaci k aplikaci Cenové vyhledávání. Nejprve bude uveden přehled funkcí jednotlivých komponent programu, poté budou komponenty rozebrány podrobněji, včetně samotných tříd, ze kterých se skládají.

6.1 Přehled komponent

Aplikace je rozdělena na několik komponent, některé z nich se od zbytku aplikace liší pouze logicky, další navíc používají i jiné technologie. V této podkapitole bude popsána koncepce komponent a jejich vzájemné spolupráce. UML diagram celé aplikace lze vidět na diagramu 6.1.

Protože aplikace poskytuje grafické uživatelské rozhraní, je přirozené, že nezanedbatelná část kódu zajišťuje zobrazení tohoto rozhraní. Mnoho komponent tedy obsahuje třídy pro vytváření GUI souvisejícího s činností komponenty nebo zobrazujícího výsledky činnosti komponenty uživateli.

Asi nejdůležitější komponentou aplikace je komponenta *Hledání*, která koordinuje vyhledávání v jednotlivých obchodech na základě požadavku uživatele. Tato komponenta používá komponentu *Konfigurace* pro sestavení vyhledávacích dotazů pro podporované obchody, komponentu *Pomocných funkcí* pro stažení výsledků z těchto obchodů a komponentu *Parsování* pro získání seznamu nalezených zboží z těchto stránek. Také používá *Databázo-*



Obrázek 6.1: UML diagram package aplikace Cenové vyhledávání

vou komponentu pro cachování výsledků.

Dále aplikace obsahuje několik převážně GUI komponent - *Administrátorské rozhraní*, *Rozhraní přihlášeného uživatele* a čistě GUI komponentu *Nápověda*. Všechny tyto komponenty používají některé metody komponenty *Přihlašování*.

Další důležitou komponentou je komponenta *Sledování*, která zajišťuje sledování vybraného zboží. Komponenta obsahuje GUI část sloužící ke komunikaci s uživatelem ohledně sledování.

Již zmíněná komponenta *Konfigurace* zajišťuje načítání, správu a zpřístupnění konfigurace ostatním částem programu. Jde jednak o konfiguraci související s podporovanými obchody (jejich počet, adresy, kódování, které používají,...), konfiguraci hledání a sledování (timeouty, ...) a konfiguraci instalace na webovém serveru (cesty ke konfiguračním souborům, přihlašovací údaje pro databázi,...).

Komponenta *Pomocných funkcí* kromě zmíněné metody pro stažení webové stránky s výsledky hledání obsahuje i další pomocné metody pro usnadnění výpisu opakujících se částí GUI nebo zaslání emailu.

6.2 Popis komponent

V této kapitole budou postupně popsány komponenty aplikace Cenové vyhledávání. Dokumentaci jednotlivých tříd a jejich metod lze nalézt v Javadoc dokumentaci k programu.

Komponenta Hledání

Tato komponenta zajišťuje vyhledávání zboží podle uživatelem zadaných kritérií. Pro zobrazení vyhledávacího formuláře v jednoduché, resp. pokročilé variantě slouží JSP soubory *index.jsp*, resp. *big_index.jsp*. Zpracování informací z obou těchto variant formuláře má na starosti servlet *start*, který zároveň vypisuje uživateli výsledky v požadavaném formátu a aplikuje volby hledání (case-sensitive vyhledávání, hledání pouze v názvu, hledání všech slov, řazení výsledků). Servlet koordinuje použití cache a získávání odpovědí od podporovaných obchodů. Zpracování odpovědí od jednotlivých obchodů probíhá v samostatných vláknech reprezentovaných třídou *search_thread*, pro synchronizaci mezi těmito vlákny je použita třída *bariera*, implementující známé synchronizační primitivum. Pro ošetření situace, kdy některý z obchodů neodpoví do stanoveného timeoutu, slouží třída *vlakno_killer*.

V získávání použitelných dat z www stránek obchodů pomáhá servletu *start* třída *search*. Ta koordinuje sestavení vyhledávacího dotazu pro obchod (spoluprací s třídou *init* z komponenty *Konfigurace*), stažení www stránky s výsledky z internetu (spoluprací s třídou *html* z komponenty *Pomocných funkcí*) a parsování této stránky (spoluprací s komponentou *Parsování*). Také zajišťuje, aby v případě, že obchod podporuje výpis výsledků hledání do více stránek, byly takto načteny všechny stránky.

Jako struktura pro uložení informací o jednom nalezeném zboží slouží třída *item*.

Komponenta Sledování

Komponenta obsluhuje samotné sledování, včetně jeho zahájení a zrušení. Zahájení sledování poté, co uživatel stiskl tlačítko *Sledovat*, zajišťuje servlet *sledovano*, který zároveň informuje uživatele o úspěšném začátku sledování. Servlet *zrusit* naopak provede zrušení sledování, včetně informování uživatele. Třída *daemon* koordinuje sledování - v pravidelných intervalech (podle nastavení z konfiguračního souboru hledání/sledování) provádí obnovu cache a zároveň i sledování zboží, zda se nezměnila jeho cena.

Komponenta Parsování

Tato komponenta zajišťuje parsování www stránek s nabídkami obchodů. Samotné parsování provádí třída *parser* a využívá k tomu regulární výrazy podle normy Javy, konkrétně její package *java.util.regex*. Pro některé často aplikované regulární výrazy ovšem aplikace z důvodů zvýšení rychlosti používá pouze jejich podmnožinu, jejíž implementace je v souboru *my_regex.java* ve třídách *MyPattern* a *MyMatcher*.

Třídy *MyPattern* a *MyMatcher* jsou obdobou standardních tříd *Pattern* a *Matcher* z package *java.util.regex*. Matchování ale na rozdíl od nich probíhá v lineárním čase vzhledem k délce textu, ve kterém je vyhledáváno a na rozdíl od standardních regulárních výrazů se nebacktrackuje. Formát regulárního výrazu je následující - libovolná posloupnost znaků (neescapovaných), speciální význam má pouze posloupnost znaků *(.*?)*, která dělí celý

výraz na podvýrazy, které jsou brány jako běžný text bez speciálních znaků (i v případě, že by se v něm nějaké speciální znaky podle normy Javy vyskytovaly). Matchování funguje tak, že nejprve je nalezen první podvýraz, pak je za posloupnost *(.*?)* namatchován nejmenší možný počet znaků (může být i nulový) do nalezení dalšího podvýrazu (druhý podvýraz se začíná hledat za koncem vyskytu prvního podvýrazu) a takto se pokračuje pro každý zbývající podvýraz. Pokud jsou nalezeny všechny podvýrazy, je matchování úspěšné.

Samotné parsování www stránky s nabídkou obchodu probíhá takto. Celá stránka je nejprve pomocí regulárního výrazu *top-level-regexp* (pouze podmnožina výrazů) rozdělena na část čísel stránek a část se seznamem nalezeného zboží, které budou dále parsovány. Počet stránek s výsledky hledání je získán pomocí opakované aplikace výrazu *skupina-stranky* (podle normy Javy) na takto získanou část čísel stránek. Informace o nalezených zboží jsou z části stránky se seznamem nalezeného zboží získány pomocí opakované aplikace výrazu *regexp-zbozi* (pouze podmnožina výrazů). Jde o následujících pět položek pro každé zboží - *odkaz, název, popis, skladem* a *cena*. Protože ale nemusí být vždy možné přímo v regulárním výrazu *regexp-zbozi* získat přesně to, co je třeba, pro dočištění textu vlastnosti (např. ceny od znaku oddělujících tisíce) je ještě možné aplikovat speciální regulární výraz *regexp-vlastnosti* pro vlastnost. V tomto výrazu lze použít plné regulární výrazy podle normy Javy. Je třeba zmínit, že *regexp-vlastnosti* je aplikován opakovaně, dokud něco namatchuje, a jeho výsledky jsou řetězeny. Pokud není dodatečný regulární výraz potřeba, je třeba uvést *.** jako hodnotu *regexp-vlastnosti* a 0 jako hodnotu *skupina-v-regexpu-vlastnosti*. Pokud obchod některou z vlastností u svých zboží neuvádí, je třeba uvést u všech třech jeho podpoložek této vlastnosti znak pomlčka (-). Podrobnější informace o formátu konfiguračního souboru parsování lze nalézt v administrátorské dokumentaci.

Komponenta Konfigurace

Tato komponenta soustřeďuje veškerou funkčnost týkající se načítání, ukládání a zpřístupňování konfigurace ostatním částem aplikace. Pro načítání obsahu centrálního konfiguračního souboru slouží třída *konfigurace*, tato třída obsahuje mimo jiné informace o cestách ke zbylým konfiguračním souborům,

a musí být proto inicializována ihned po startu aplikace. Pro načítání a ukládání konfiguračního souboru hledání/sledování slouží třída *conf_admin*. Pro inicializaci informací nutných k vyhledávání slouží třída *init*, která načítá a zpřístupňuje data z konfiguračních souborů obchodů a parsování. Třída *init* také zajišťuje sestavení vyhledávacích dotazů pro jednotlivé obchody. Jako struktury pro uložení informací o jednom obchodu z konfiguračního souboru obchodů, resp. konfiguračního souboru parsování slouží třída *obchod*, resp. *radek_konfig_parser*.

Komponenta Databáze

Tuto komponentu tvoří jediná třída *dbase* zajišťující přístup do databáze přes JDBC konektor.

Komponenta Administrátorské rozhraní

Zobrazení úvodní stránky administrátorského rozhraní má na starosti třída *admin*. Formulář pro změnu nastavení je zobrazován JSP technologií souborem *admin_web_conf.jsp*. Zpracováním dat z tohoto formuláře a jejich případným zápisem do konfiguračního souboru hledání/sledování se zabývá servlet *admin_change_conf*. Zobrazení všech statistik zajišťuje třída *stats*. Stránku se seznamem registrovaných uživatelů zobrazuje servlet *uzivatele*, rušení uživatelů přístupné ze stránky s tímto seznamem řeší servlet *user_smazat*.

Komponenta Rozhraní přihlášeného uživatele

Zobrazení úvodní stránky rozhraní přihlášeného uživatele má na starosti třída *user*. Formulář pro změnu registračních údajů (pouze hesla a emailu) je zobrazován JSP technologií souborem *user_edit.jsp*. Zpracováním dat z tohoto formuláře a jejich případným zápisem do databáze se zabývá servlet *user_change*. Zobrazení historie hledání zajišťuje třída *history*, zobrazení aktivních sledování zboží pak třída *watch*. Smazání historie zajišťuje servlet *history_clear*.

Komponenta Přihlašování

Tato komponenta obsahuje JSP i servletovou část. JSP část obsahuje dva soubory login.jsp a registrace.jsp, které slouží k zobrazení přihlašovacího, resp. registračního formuláře uživateli. Servletová část obsahuje k nim příslušné třídy *login* a *registrace*, které zajišťují zpracování informací z těchto formulářů, ověření jejich správnosti a případně přihlášení nebo registraci uživatele do systému. Poslední třídou z této komponenty je třída *bye*, která zajistí odhlášení uživatele.

Komponenta Nápověda

Tato komponenta je z programátorského hlediska velmi jednoduchá, jde pouze o dva JSP soubory *napoveda-admin.jsp* a *napoveda-user.jsp*, které zobrazují administrátorskou, resp. uživatelskou dokumentaci ve formě www stránky.

Komponenta Pomocné funkce

Tato komponenta obsahuje několik různorodých tříd. První z nich je třída *filter*, která zajišťuje překódování všech HTTP požadavků (jejichž součástí jsou vstupní data všech formulářů aplikace) do požadovaného kódování *utf-8*. Její metoda *doFilter* je volána Tomcatem pomocí mechanismu filtrů ještě předtím, než tato data dostane jakákoli jiná třída aplikace.

Třída *html* slouží pro usnadnění výpisu opakujích se částí www stránek ostatním částem aplikace. Obsahuje několik konstant pro výpis různých zápatí a dalších částí a metody pro výpis jednotlivých částí stránek (hlavičky, těla stránky,...). Navíc obsahuje metody pro stažení stránky zadané pomocí URL a kódování a načtení této stránky do řetězce.

Třída *mail* slouží pro zasílání upozorňovacích emailů uživatelům při změně stavu sledovaného zboží, nebo administrátorovi při výjimečných stavech aplikace. Třída *util* obsahuje několik metod pro práci s řetězci.

Závěr

Během práce na tomto softwarovém projektu, potažmo bakalářské práci jsem si vyzkoušel několik nových technologií. Ty, které jsem nakonec pro vývoj své aplikace zvolil, se osvědčily. Také jsem musel vyřešit několik technických problémů, mezi nimiž vyniká zvláště nejednota kódování česky psaných webových stránek. Na českých webech se běžně vyskytují tři různá kódování češtiny, což způsobuje problémy tvůrcům podobných aplikací.

Vytvořená aplikace funguje jak bylo na začátku vývoje zamýšleno. Navíc je bez problémů rozšiřitelná, přidání dalších internetových obchodů je snadné a nevyžaduje zvláštní odborné znalosti.

Literatura

[1] Kanisová H., Müller M.: *UML srozumitelně*, Computer Press, 2004

[2] Gosling J., Joy B., Steele G., Bracha G.: *Java language specifiction, 3rd edition*, Prentice Hall, 2005

[3] `http://www.mysql.org`

[4] `http://tomcat.apache.org`

[5] `http://java.sun.com/products/servlet`