

System pro hodnocení výhodnosti nákupu

System for an evaluation of the shopping convenience

Tomáš Tureček

Bakalářská práce
2007



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2006/2007

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš TUREČEK**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Systém pro hodnocení výhodnosti nákupu**

Zásady pro vypracování:

1. Seznámení s problematikou systémů pro hodnocení výhodnosti nákupu
2. Seznámení s využívanými technologiemi a stanovení základních funkcí
3. Stanovení struktury evidovaných dat o zboží, stanovení principu získávání dat od internetových obchodníků
4. Návrh struktury databáze
5. Návrh způsobu agregace odpovídajících si položek zboží od obchodníků
6. Návrh uživatelského rozhraní
7. Realizace systému za využití technologií ASP.NET 2, MS SQL 2005, XML
8. Zhodnocení přínosu systémů pro hodnocení výhodnosti nákupu a použitého řešení

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. BILL, Evjen, et al. ASP.NET 2.0 : Programujeme profesionálně. [s.l.] : Computer Press, 2007. 1224 s. ISBN 978-80-251-1473-5.
2. WALTER, Stephen. ASP.NET 2.0 Unleashed . Los Angeles, USA : Sams, 2006. 1992 s. Bk&CD-Rom edition. ISBN 0672328232.
3. ESPOSITO, Dino. XML efektivní programování pro .net. Praha : Grada Publishing, 2004. 596 s. ISBN 80-247-0775-6.
4. YOUNG, Michael J. XML krok za krokem. Miroslav Kučera; Ing. Aleš Thiemel. 1. vyd. Praha : Mobil Media, 2002. 472 s. ISBN 80-7226-170-3.

Vedoucí bakalářské práce:

Ing. Petr Šilhavý

Ústav aplikované informatiky

Datum zadání bakalářské práce:

13. února 2007

Termín odevzdání bakalářské práce:

24. května 2007

Ve Zlíně dne 13. února 2007



prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

System pro hodnocení výhodnosti nákupu je webová aplikace napsaná v ASP.NET2 umožňující uživatelům rychlé nalezení hledaného produktu a porovnání ceny k tomuto produktu agregovaných položek z různých internetových obchodů. Ceníky jsou stahovány z obchodů zaregistrovaných do systému v podobě datových souborů ve formátu XML a získaná data jsou ukládána do databáze Microsoft SQL 2005.

Klíčová slova: Porovnávání cen, internetové obchody, ASP.NET 2, MS SQL 2005, XML

ABSTRACT

The System for an evaluation of the shopping convenience is a web application designed in ASP.NET2, made with the purpose of quick retrieval of a searched product and aggregated items to this product, not least giving a comparison of a price between the shops offering this product. The price lists are downloaded from the registered shops as an XML file and saved into the Microsoft SQL 2005 database.

Keywords: Price comparison, internet shops, ASP.NET 2, MS SQL 2005, XML

Děkuji vedoucímu bakalářské práce, Ing. Petrovi Šilhavému, za odborné vedení, konzultace, rady a připomínky, které mi poskytnul na začátku a během řešení především praktické části této práce.

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně

.....
Podpis diplomanta

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 POUŽITÉ TECHNOLOGIE	10
1.1 ASP.NET.....	10
1.2 SQL DATABÁZE	12
1.3 XML	13
1.4 HTML.....	13
1.5 CSS.....	14
1.6 VÝVOJOVÉ A KONFIGURAČNÍ NÁSTROJE	14
2 ROZBOR ŘEŠENÍ	16
2.1 STANOVENÍ CÍLE PRÁCE.....	16
2.2 STANOVENÍ ZÁKLADNÍCH FUNKCÍ.....	16
2.3 PRINCIP ZÍSKÁVÁNÍ DAT	16
2.4 AGREGACE POLOŽEK	17
2.5 PROPOJENÍ POUŽITÝCH TECHNOLOGIÍ	17
II PRAKTICKÁ ČÁST	19
3 FORMÁT DATOVÉHO SOUBORU	20
3.1 STRUKTURA DATOVÉHO SOUBORU	20
4 DATABÁZE	22
4.1 STRUKTURA DATABÁZE.....	22
4.2 POPIS TABULEK	22
4.3 VYTVOŘENÍ A KONFIGURACE DATABÁZE.....	24
4.4 SQL SCRIPT PRO PŘENOS NA JINÝ SERVER (SQLQUERY.SQL)	25
5 IMPORT CENÍKU	26
5.1 PŘÍPRAVA DATABÁZE A NAČTENÍ CENÍKU (XMLLOADER.CS).....	26
5.2 INDEXACE POLOŽKY (ITEM.CS)	27
5.3 INDEXACE PRODUKTU (ARTICLE.CS).....	28
5.4 INDEXACE VÝROBCE (MANUFACTURER.CS).....	30
5.5 INDEXACE KATEGORIE (CATEGORY.CS).....	30
5.6 KONSTRUKCE A VYKONÁVÁNÍ SQL DOTAZŮ (SQL.CS).....	31

6	UŽIVATELSKÉ ROZHRANÍ	33
6.1	ROZVRŽENÍ	33
6.2	ÚVODNÍ STRANA (HOME.ASPX)	35
6.3	KATALOG PRODUKTŮ (LIST.ASPX).....	36
6.4	DETAIL PRODUKTU (DETAIL.ASPX).....	40
6.5	SEZNAM OBCHODŮ (SHOPS.ASPX)	41
6.6	DETAIL OBCHODU (SHOP.ASPX).....	41
6.7	REGISTRACE OBCHODU (REGISTER.ASPX)	42
6.8	NÁPOVĚDA (HELP.ASPX)	43
6.9	ADMINISTRACE (ADMIN.ASPX).....	43
	ZÁVĚR	45
	ZÁVĚR V ANGLIČTINĚ	46
	SEZNAM POUŽITÉ LITERATURY	47
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	48
	SEZNAM OBRÁZKŮ	49
	SEZNAM TABULEK	50
	SEZNAM PŘÍLOH	51

ÚVOD

Na internetu je nespočetné množství obchodů, které nabízejí nepřehledné množství produktů. Pro zákazníka dnes není nijak složité najít na obchod, který nabízí určitý produkt, ale je stále obtížnější porovnat cenu tohoto produktu v několika obchodech.

Každý den přibývají nové a nové obchody které si navzájem konkurují. Prodejci chtějí zákazníka nalákat do svého obchodu a zejména mají zájem na tom, aby si zákazník jejich obchod zapamatoval nebo nejlépe oblíbil. Proto prodejci dávají výrazné slevy na zboží tak, aby cena produktu zákazníka zaujala. Proto bývají ceny v internetovém obchodě často nižší než v klasickém kamenném obchodě.

Chce-li zákazník najít obchod, který nabízí daný produkt za nejnižší cenu, je to často velmi zdlouhavý proces, při kterém navíc není tato nejnižší cena skutečně nalezena. Pokud by existoval centralizovaný systém shromažďující exportované ceníky internetových obchodů ve formě konzistentních dat, který by navíc agregoval stejné produkty z více obchodů, bylo by pro zákazníka nesmírně jednoduché najít obchod s nejnižší cenou hledaného produktu.

I. TEORETICKÁ ČÁST

1 POUŽITÉ TECHNOLOGIE

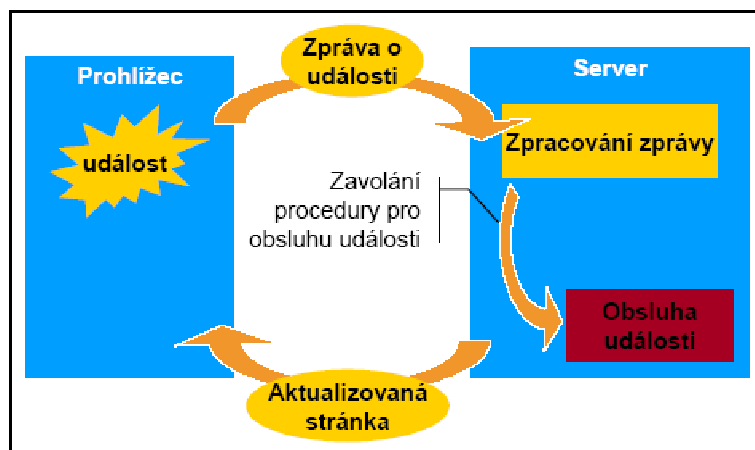
1.1 ASP.NET

1.1.1 Úvod do ASP.NET

ASP.NET je programovací koncepce (Framework) postavená nad obecným běhovým prostředím (Common Language Runtime - CLR). Nasazením na server lze vytvářet silné webové aplikace. První verze ASP.NET 1.x přinesla pro webový vývoj mnohá významná vylepšení, ASP.NET 2.0 zdokonaluje jejich možnosti a přidává nové koncepce v oblasti vývojářské produktivity, administrace, managementu, rozšiřitelnosti a výkonu.

Vývojář může jednoduše přidávat vlastní prvky i jejich chování, aby splnil své požadavky, nové prostředí umožní vytváření aplikací rychleji, s menším rozsahem kódu a úsilím.

ASP.NET zavedlo nový způsob práce s formuláři. Systém sám se nyní stará o komunikaci mezi klientem a serverem a sám si také generuje HTML tagy. Serverové ovládací prvky jsou vlastně prvky stránky (HTML), které nenapsal ani se o ně nestará autor kódu, ale které spravuje ASP.NET.



Obrázek 1. Vztah mezi klientským prohlížečem a serverem s ASP.NET

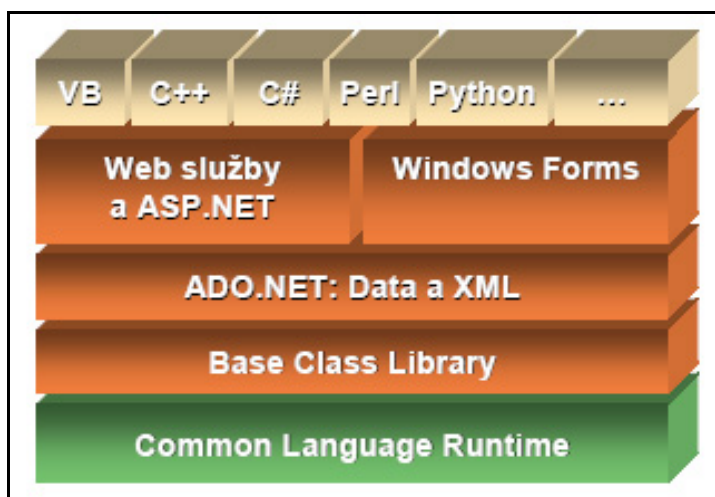
1.1.2 .NET Framework

.NET framework stojí jako nadstavba nad operačním systémem. Obsahuje CLR (Common Language Runtime) zajišťující základní funkcionalitu celého .NET frameworku. Na tomto Common Language Runtime jsou postaveny všechny další základní knihovny a

objekty sloužící například pro přístup k datům (ADO.NET, XML, SQL), k vícevláknovému zpracování aplikací (Threading), k přístupu do sítě a internetu (NET) a k zpracovávání, auditování a monitorování bezpečnosti aplikací (Security).

Tyto knihovny tvoří srdce .NET frameworku a jsou přístupné z jakéhokoli jazyka platformy .NET. Přímou podporou Microsoftem jsou podporované čtyři jazyky: Visual Basic, C++, JScript a nový, velmi perspektivní jazyk C#. Všechny jazyky pracující na platformě .NET však musí splňovat určité předpoklady dané CLS (Common Language Specification), která obsahuje soubory vlastností a požadavků kladených na tyto jazyky.

Pro činnost webových stránek v ASP.NET 2.0 je zapotřebí Microsoft .NET Framework 2.0.



Obrázek 2. Architektura .NET Framework

1.1.3 Jazyk C#

C# je jeden z nejmodernějších jazyků dnešní doby. Byl vytvořen tak, že slučuje přednosti ostatních jazyků. Nejvíce se jazyk C# podobá jazyku Java. Vznikl na základě podobných myšlenkových úvah a snahy řešit podobné problémy. Jeho syntaxe vychází z jazyka C++, provedené změny směřují zvláště ke zjednodušení. Další odlišnosti byly evokovány snahou o jednoduchou podporu komponentně orientovaného vývoje.

1.2 SQL databáze

1.2.1 Co je to databáze?

Databázi si můžeme představit jako prostor, do kterého se ukládají všechny potřebné údaje. Zpracováním a přístupem údajů v databázi bývá pověřen program, kterému se říká DBMS (DataBase Management System), nebo česky SŘBD (Systém Řízení Báze Dat). Veškeré zpracování, ukládání a získávání dat je prováděno DBMS programem. Jakákoli aplikace, která databázi používá vždy přistupuje přes tento DBMS program. Mezi DBMS patří řada programů v nejrůznějších cenových kategoriích. Většina profesionálních a komerčních DBMS nejsou programy nikterak levné a jejich cena se pohybuje ve velmi vysokých částkách. Naštěstí existují v kategorii DBMS i programy, které jsou v nízké cenové relaci, nebo zcela zdarma. Naprostá většina dnešních DBMS staví na tzv. relačním modelu dat. V tomto modelu jsou veškerá data uspořádána do databázových tabulek. Každá tabulka zpravidla uspořádává údaje o určitém objektu. Databázová tabulka je uspořádána do řádků a sloupců.

1.2.2 Jazyk SQL

Přístup k údajům uloženým v databázi obstarává DBMS program. Aby mohly jiné aplikace používat také údaje z databáze, musí DBMS program nabízet rozhraní, pomocí kterého budou moci ostatní aplikace s DBMS spolupracovat. Při práci s DBMS programem se používá osvědčený model klient/server. V roli serveru vystupuje DBMS program nabízející své služby. DBMS program je tak nepřetržitě spuštěn a čeká na požadavky od klientů - jiných aplikací. Na tyto požadavky pak odpovídá a reaguje. Protože DBMS program vystupuje v roli serveru, říká se mu také databázový server.

Pro zápis požadavků na databázový server se dnes nejčastěji používá SQL jazyk. SQL je anglická zkratka znamenající Structured Query Language. Tento jazyk dnes používá naprostá většina databázových serverů. Proto se takovým serverům říká zjednodušeně SQL servery. Samotný SQL obsahuje vše potřebné pro práci s databázemi - ať už příkazy pro práci na vytvoření, rušení, modifikování tabulky, tak i pro práci s údaji samotnými - přidávání, změnu, rušení a vyhledávání údajů. Jazyk SQL má za sebou poměrně dlouhý vývoj. Jeho prototyp by poprvé implementován v roce 1974 společností IBM a dá se říci, že se dodnes stále vyvíjí, hlavně co se týká přidávání nových technologií.

Skutečným faktickým standardem se stal SQL až v roce 1986, kdy se dočkal standardizace díky organizaci ANSI. Od té doby se tedy SQL stal univerzálním nástrojem pro programování databází. Dnes se každý, kdo se dostává do styku s vývojem, či laděním databází, musí seznámit s jazykem SQL, který je prakticky u databází všudypřítomný.

1.3 XML

Historie jazyka XML spadá do poloviny 90. let. Roku 1996 společnost W3C začala pracovat na definování standardu tohoto jazyka. Tento jazyk vychází z obecnějšího značkovacího jazyka SGML (Standard Generalized Markup Language), který ovšem pro svou relativní složitost neměl a nemá velkou šanci na rozšíření. Proto speciální skupina odborníků XWG (XML Working Group) vyvinula standard jazyka XML jakožto zjednodušenou verzi jazyka SGML. Úkolem skupiny XWG kromě definování standardu je také řešení problematiky linkování dokumentů mezi sebou a dále specifikace designu dokumentů napsaných v XML.

1.4 HTML

1.4.1 Co je to HTML?

HTML je formátovací, nikoliv programovací jazyk. Samotné internetové stránky zobrazované prohlížečem jsou jen obyčejné textové soubory, ve kterých je zapsán text obohacený o tagy, které určují význam jednotlivých částí textu. Prohlížeč tyto soubory čte, interpretuje tagy a formátuje výsledný obsah. Podle jednotlivých tagů jsou na stránku prohlížečem kromě textu vkládány ještě další zobrazované objekty jako odkazy, obrázky nebo formuláře.

1.4.2 Historie HTML

Na přelomu 80. a 90. let vznikla v CERNu (evropské středisko atomového výzkumu) první verze jazyka HTML. Původně byl jazyk vyvinut pro rychlé předávání informací mezi vědeckými pracovníky v tomto středisku. Přes odkazy bylo vzájemně provázáno více dokumentů na různých počítačích a pomocí speciálního prohlížeče se tyto dokumenty zobrazovaly.

Jazyk HTML zanedlouho pronikl i mimo vědeckou sféru a začaly se objevovat nové prohlížeče. HTML se neustále vyvíjelo a bylo doplňováno o nové prvky.

Dnes se o standarty HTML stará konsorcium W3C, které doporučuje nové standarty HTML, které specifikují používané tagy. Jazyk HTML se neustále vyvíjí, standardní, dnes používané verze HTML jsou: 3.2, 4, 4.1, a XHTML 1.0, XHTML 2.

1.5 CSS

CSS (Cascading Style Sheets) v českém překladu znamená kaskádové styly. CSS vzniklo v roce 1997 a je součástí HTML od verze 4.

CSS umožňuje definovat vlastnosti určitých elementů jazyka HTML např. barvy textu, pozadí, zarovnání, velikost písma a mnoho dalších vlastností.

CSS se může zapisovat přímo do HTML dokumentu nebo do externího souboru s příponou .css, na který je odkazováno v html stránce patřičným kódem.

1.6 Vývojové a konfigurační nástroje

1.6.1 MS Visual Web Developer 2005 Express Edition

Je to program určený pro vytváření webových stránek. Verze Express doplňuje rodinu Express programů společnosti Microsoft. Tyto programy jsou k dispozici zdarma a jsou určeny nováčkům, nadšencům a studentům. Na druhou stranu postrádají pokročilé funkce, které se uplatňují v komerční sféře.

1.6.2 MS SQL Server 2005 Express Edition

SQL Express je volně stažitelná verze Microsoft SQL Serveru. Nabízí databázové řešení zaměřené na malé aplikace. Na rozdíl od ostatních verzí SQL Serveru, není toto řešení vhodné pro rozsáhlejší projekty, zejména kvůli omezení velikosti a několika pokročilých funkcí.

1.6.3 MS SQL SERVER Configuration Manager

Jedná se o nástroj k základní správě a konfiguraci. Pomocí tohoto nástroje lze zastavovat a spouštět jednotlivé nainstalované instance SQL Serveru a měnit účet, pod

kterým služba běží. V nástroji lze rovněž určovat používané síťové protokoly a jejich nastavení.

1.6.4 MS SQL SERVER Studio Management Express

MS SQL Server Management studio express je grafický nástroj pro správu databáze MS SQL 2005. Jde o verzi odvozenou z SQL Server Management studio, které je dodáváno s plnou verzí MS SQL 2005. Lze ji použít pro správu databází SQL Server Express Edition a všech placených verzí SQL Serveru 2005.

2 ROZBOR ŘEŠENÍ

2.1 Stanovení cíle práce

Cílem práce bylo především splnit všechny body zadání bakalářské práce a požadavky na systém pro porovnávání cen a vytvořit navíc několik vhodných funkcí zejména pro usnadnění vyhledávání produktu a jednoduchou orientaci v katalogu. Systém by měl být po implementaci otestován hlavně na správnost agregace položek ostrými daty od obchodníků a nasazen k ukázkovému provozu na nějaký server s freehostingovým programem pro předvedení výsledků komisi hodnotící celou práci.

2.2 Stanovení základních funkcí

Základní funkcí systému plynoucí ze samotného zadání práce je porovnávání cen výrobků z různých obchodů. Aby mohlo být toto zajištěno, je nutné nejprve navrhnout princip získávání dat od obchodníků a vytvořit strukturu, v jaké budou tato data stahována.

Dále je důležité implementovat import takto stahovaných dat do databáze, která bude mít předem stanovenou strukturu tabulek tak, aby zajišťovala logické uspořádání uchovávaných dat a rychlé vyhledávání informací. Důležitý bod práce je i návrh a realizace agregace odpovídajících si položek z různých obchodů.

Po té by měla přijít na řadu realizace uživatelského rozhraní. Uživatelské rozhraní by mělo být rozděleno na několik stránek a obsahovat prvky jako katalog produktů s filtry kategorií a výrobců a s řazením podle různých kritérií, fulltextové vyhledávání v názvech produktů, výpis stromu kategorií, seznam obchodů a samozřejmě výpis agregovaných položek obchodů ke každému produktu s porovnáním jejich cen. Součástí uživatelského rozhraní bude administrace, která bude zpřístupněna na adrese známé pouze administrátorům případně osobám hodnotícím systém v rámci této práce.

2.3 Princip získávání dat

Ceníky internetových obchodů bývají uchovávány v tabulkách databází na databázových nebo webových serverech. Jejich struktura se mezi jednotlivými obchody mnohdy velmi liší, proto je nutné aby systém shromažďující informace obsažené v cenících internetových obchodů získával tato data v podobě datové struktury s nějakým logickým

uspořádáním, které je shodné pro všechny obchody. Pro popsání této datové struktury byl zvolen jazyk XML. Data jsou tak získávána ve formátu xml stahováním souboru nebo datového proudu.

2.4 Agregace položek

Na začátku práce se nabízely 2 naprosto odlišné varianty řešení, jakým způsobem nebo spíše ve který okamžik agregovat položky.

První varianta spočívala v tom, že při importu ceníku by se data pouze uložila do nějaké tabulky databáze bez jakékoliv náročné indexace všech údajů, čímž by byl import velmi rychlý a jednoduchý i při velkém množství položek v ceníku. Agregace položek měla probíhat pokaždé ve chvíli, kdy se zpracovává dotaz pro výpis seznamu produktů v katalogu uživatelského rozhraní. To by ale znamenalo nadměrné zatížení prostředků téměř při každém zpracování stránky katalogu, což by obzvlášť při větším počtu položek nebylo vhodné. Toto řešení by vytvářelo duplikované záznamy navíc logicky nevhodně uložené, čímž by se zbytečně navyšovala velikost databáze.

Druhá varianta, která byla nakonec zvolena pro implementaci celého systému, vychází z toho, že při importu ceníku jsou data pečlivě indexována. Data jsou logicky roztríděna do tabulek, jsou ignorovány duplikované záznamy a zároveň každá položka je agregována k produktu. Toto je sice velmi náročné a průběh trvá delší dobu, nicméně tato operace je v praxi prováděna maximálně jednou za týden pro každý obchod. Kratší interval by z hlediska aktualizace jen velmi malého počtu položek neměl význam. Při vyhledávání položek seznamu produktů v katalogu tak nedochází k opakovanému zatěžování serveru a díky správnému logickému uspořádání do tabulek se vyhledává jenom v tabulkách, které jsou na daném místě potřebné. Navíc je takto o několik procent zmenšena velikost databáze.

2.5 Propojení použitých technologií

Při výběru vývojových nástrojů byl brán ohled na maximální komfort při vývoji aplikace a na cenovou dostupnost i pro začínající vývojáře. Z toho důvodu byly zvoleny produkty firmy Microsoft z rodiny Express Edition, které jsou zdarma ke stažení na internetu a slouží především pro studentské účely a vývoj jednoduchých aplikací.

Webová aplikace PriceComp byla vytvořena technologií ASP.NET2, z podporovaných programovacích jazyků byl zvolen jazyk C#. Při vývoji bylo použito nástroje Microsoft Visual Web Developer 2005 Express Edition. Jako databázový server byl zvolen Microsoft Sql Server 2005 Express Edition, databáze byla vytvořena a konfigurována editačním nástrojem Microsoft SQL SERVER Management Studio Express. Aplikace byla vyvíjena na počítači s operačním systémem Microsoft Windows XP.

Projekt je rozdělen na několik serverových stránek, které po zpracování na serveru vygenerují text v podobě HTML jazyka, který je na konci zobrazen v prohlížeči uživatele. Výsledná HTML stránka je při zobrazení ještě graficky naformátována pomocí kaskádových stylů CSS, které jsou společné pro každou stránku a jsou importovány ze souboru .css.

II. PRAKTICKÁ ČÁST

3 FORMÁT DATOVÉHO SOUBORU

Pro získávání ceníků jednotlivých obchodů jsou použity soubory ve formátu XML v kódování windows-1250. Tento XML soubor je tvořen kořenovým elementem <SHOP>. V kořenovém elementu jsou vnořeny další elementy, které mohou obsahovat společné informace k položkám (jazyk, měna) a hlavně záznamy s položkami v elementech <SHOPITEM>. Každý element <SHOPITEM> pak obsahuje informace o jednotlivých položkách. Každá informace je opět vnořena uvnitř daného elementu. Příklad datového souboru a význam jednotlivých elementů je popsán níže.

3.1 Struktura datového souboru

3.1.1 Příklad datového souboru

Toto je příklad datového souboru obsahující 2 položky:

```
<?xml version="1.0" encoding="windows-1250" ?>
- <SHOP>
  <ID>www.ahmobile.cz/inshop/</ID>
  <GENERATED>20070507</GENERATED>
  <LANGUAGE>Czech</LANGUAGE>
  <CURRENCY>CZK</CURRENCY>
  <CURRENCY_MARK>Kč</CURRENCY_MARK>
- <SHOPITEM>
  <MANUFACTURER>Nokia</MANUFACTURER>
  <PRODUCT>Nokia 1600</PRODUCT>
  <DESCRIPTION>Mobilní telefon</DESCRIPTION>
  <URL>http://www.vasobchod.cz/mobily/nokia-1600.html</URL>
  <IMGURL>http://www.vasobchod.cz/img/nokia-1600.jpg</IMGURL>
  <PRICE>1500</PRICE>
  <PRICE_VAT>1785</PRICE_VAT>
  <FULLCATEGORYTEXT>Hlavní oddělení|Všechny kategorie|Telefony</FULLCATEGORYTEXT>
</SHOPITEM>
- <SHOPITEM>
  <MANUFACTURER>Nokia</MANUFACTURER>
  <PRODUCT>Nokia 2650</PRODUCT>
  <DESCRIPTION>Mobilní telefon</DESCRIPTION>
  <URL>http://www.vasobchod.cz/mobily/nokia-2650.html</URL>
  <IMGURL>http://www.vasobchod.cz/img/nokia-2650.jpg</IMGURL>
  <PRICE>2000</PRICE>
  <PRICE_VAT>2380</PRICE_VAT>
  <FULLCATEGORYTEXT>Hlavní oddělení|Všechny kategorie|Telefony</FULLCATEGORYTEXT>
</SHOPITEM>
</SHOP>
```

Obrázek 3. Ukázka datového souboru pro import dat

3.1.2 XML elementy

Povinné údaje o každé položce jsou obsaženy v elementech <PRODUCT>, <DESCRIPTION>, <URL> a <PRICE_VAT>. Pokud nejsou tyto elementy u nějaké položky vytvořeny nebo obsahují-li prázdné hodnoty, není pro tuto položku vůbec započat proces indexace a žádná informace o této položce nebude uložena do databáze.

Popis významu jednotlivých elementů je popsán v následující tabulce.

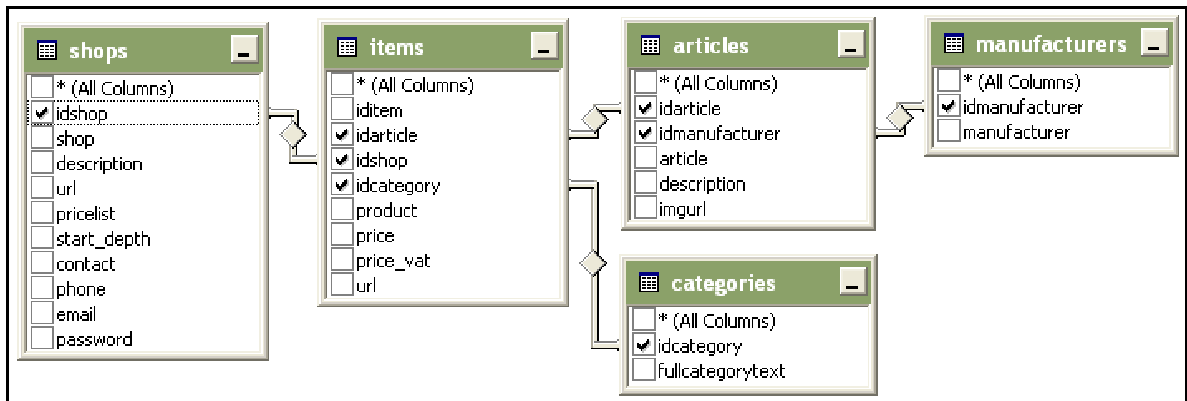
XML element	Popis významu
SHOP	kořenový element celého souboru
SHOPITEM	obsahuje jednotlivé informace o položce
MANUFACTURER	výrobce
PRODUCT	název položky
DESCRIPTION	popis položky
URL	url adresa stránky s detailem položky v obchodě
IMGURL	url adresa obrázku k dané položce v obchodě
PRICE	cena bez DPH
PRICE_VAT	cena včetně DPH
FULLCATEGORYTEXT	kategorie zapsaná cestou ve stromové struktuře

Tabulka 1. Popis jednotlivých elementů XML souboru

4 DATABÁZE

Aplikace byla vyvíjena pod databází Microsoft SQL SERVER 2005 Express Edition. Vlastní databáze obsahuje 5 tabulek se vzájemnými vztahy. Tabulka articles obsahuje navíc fulltextový index na sloupci article, umožňující fulltextové vyhledávání v tomto sloupci.

4.1 Struktura databáze



Obrázek 4. Schéma struktury databáze

4.2 Popis tabulek

4.2.1 Shops

Tato tabulka uchovává informace o každém obchodě. Záznamy zde vznikají při registraci obchodu do systému.

Název sloupce	Datový typ	Popis
idshop	int	identifikátor obchodu
shop	varchar(255)	název obchodu
description	varchar(255)	popis obchodu
url	varchar(255)	internetová adresa obchodu
pricelist	varchar(255)	adresa ceníku pro import
start_depth	int	hodnota počáteční hloubky tvorby kategorií
contact	varchar(50)	jméno kontaktní osoby

phone	varchar(15)	telefonní číslo kontaktní osoby
email	varchar(255)	kontaktní e-mailová adresa
password	varchar(15)	heslo

Tabulka 2. Struktura databázové tabulky shops

4.2.2 Items

Tato tabulka obsahuje informace o položkách. Jsou zde společně uloženy položky všech obchodů systému. Obchod, do kterého každá položka patří je určen identifikátorem ve sloupci idshop.

Název sloupce	Datový typ	Popis
iditem	int	identifikátor položky
idarticle	int	identifikátor artiklu (produktu)
idshop	int	identifikátor obchodu
idcategory	int	identifikátor kategorie
product	varchar(255)	název produktu v obchodě
price	money	cena bez dph
price_vat	money	cena včetně dph
url	varchar(255)	internetová adresa položky v obchodě

Tabulka 3. Struktura databázové tabulky items

4.2.3 Articles

V tabulce articles jsou uloženy záznamy produktů (artiklů), do kterých jsou agregovány položky z tabulky items.

Název sloupce	Datový typ	Popis
idarticle	int	identifikátor artiklu (produktu)
idmanufacturer	int	identifikátor výrobce

article	varchar(255)	název produktu
description	ntext	popis produktu
imgurl	varchar(255)	url adresa obrázku

Tabulka 4. Struktura databázové tabulky articles

4.2.4 Categories

Do tabulky categories se ukládají řetězce obsahující názvy kategorií včetně úplné cesty od kořenové kategorie.

Název sloupce	Datový typ	Popis
idcategory	int	identifikátor kategorie
fullcategorytext	varchar(255)	název kategorie včetně úplné cesty

Tabulka 5. Struktura databázové tabulky categories

4.2.5 Manufacturers

Tato tabulka slouží jako číselník výrobců.

Název sloupce	Datový typ	Popis
idmanufacturer	int	identifikátor výrobce
manufacturer	varchar(50)	název výrobce

Tabulka 6. Struktura databázové tabulky manufacturers

4.3 Vytvoření a konfigurace databáze

V programu Microsoft SQL SERVER Management Studio jsem vytvořil novou databázi. Bylo třeba v databázi povolit fulltextové vyhledávání. Dále jsem pomocí vývojového prostředí Visual Web Developer 2005 Express Edition navrhnul a vytvořil tabulky a nastavil indexy včetně fulltextového nad polem article tabulky articles.

4.4 Sql script pro přenos na jiný server (sqlquery.sql)

Pomocí programu Microsoft SQL SERVER Studio Management byl vytvořen sql script, který slouží pro jednoduchý přenos struktury databáze na jiný počítač. Tento script obsahuje strukturu všech tabulek a klíčů databáze. Po vytvoření databáze a provedení tohoto skriptu se vytvoří celá struktura a databáze je tímto připravena k použití.

5 IMPORT CENÍKU

Importem ceníku je myšleno načtení dat ve formě xml souboru z url adresy a následné začlenění těchto dat pomocí programu do databáze. Xml soubor je buď uložen na internetových stránkách (FTP serveru) obchodu nebo se dynamicky generuje z databáze při každém pokusu o stažení. V případě dynamického generování jsou tato data obsažená v souboru naprosto aktuální, je-li soubor uložen na stránkách, záleží na každém provozovateli obchodu, jak často je soubor aktualizován a tedy jak aktuální jsou data v něm obsažená.

Funkce pro import ceníku jsou logicky uspořádány do tříd a ty jsou uloženy v souborech .cs v adresáři App_Code. Každá třída obsahuje specifické funkce a je uložena v samostatném souboru. Obsah a význam těchto tříd stejně jako jejich funkce a metody jsou podrobně popsány níže.

5.1 Příprava databáze a načtení ceníku (XmlLoader.cs)

Tato třída obsahuje metody a funkce pro stažení ceníku, přípravu databáze před importem, získání url adresy a inicializačních parametrů importu daného obchodu z databáze a vlastní import.

5.1.1 LoadPriceList(string idshop)

Řídící funkce celého procesu importu dat ceníku obchodu do databáze. Jejím parametrem je identifikátor daného obchodu idshop. Je zavolána funkce PriceList s předaným identifikátorem obchodu idshop, která vrátí z databáze url adresu ceníku vybraného obchodu. Po té je ceník ve formě xml souboru stažen z této url adresy.

Před započítím vlastního importu dat, jsou všechna data starého ceníku daného obchodu v databázových tabulkách items a categories smazána, protože budou nahrazeny aktuálnějšími daty.

Začne se procházet struktura staženého xml souboru a pro každý uzavřený xml tag <SHOPITEM> zastupující položku je vytvořena instance třídy Item s patřičnými parametry, která má na starost kontrolu a zaindexování dat této položky.

5.1.2 DbTruncate()

Metoda, která vyprázdní všechny tabulky kromě shops, kde jsou uloženy informace o obchodech zaregistrovaných v systému. Pro vyprázdnění je použit sql příkaz TRUNCATE.

5.1.3 PriceList(string idshop)

Funkce vrací url adresu ceníku obchodu, jehož identifikátor je předán v parametru, z databáze.

5.1.4 StartDepth(string idshop)

Funkce získává z databáze počáteční úroveň tvorby kategorií obchodu, jehož identifikátor je předán v parametru.

5.2 Indexace položky (Item.cs)

Třída Item obsahuje funkce pro ověření unikátnosti položky v obchodě a uložení do databáze. Unikátnost je ověřována podle názvu položky. Obsahuje-li jeden obchod více položek se stejným názvem, je zaindexována pouze první položka v pořadí, ostatní položky se stejným názvem jsou ignorovány.

5.2.1 Item(DataRow row, string idshop, int StartDepth)

Parametr row je záznam obsahující data jedné položky. Tento záznam odpovídá datům mezi xml tagy <SHOPITEM></SHOPITEM> v importovaném xml souboru.

Parametr idshop je unikátní identifikátor obchodu, pod kterým je daný obchod uložen v tabulce shops.

Parametr StartDepth je číslo určující počáteční úroveň zanoření kategorií daného obchodu, od které se mají ukládat kategorie.

Tato funkce napřed přečte všechny potřebné údaje z parametru row a uloží si je do proměnných, aby s nimi mohla dále pracovat. Pokud nejsou všechna povinná data u položky obsažena v proměnných, položka se neindexuje, tzn. funkce je ukončena bez ovlivnění databáze. Pokud jsou potřebné údaje obsaženy, následuje ověření, zda-li je již zaindexována pro tento obchod položka s daným názvem.

Pokud je již položka se stejným názvem v tabulce items obsažena, funkce je opět ukončena bez ovlivnění dat v databázi.

Dále je vytvořena instance třídy Article a zavolána její funkce getIdArticle s patřičnými parametry (viz indexace artiklu), která vrací identifikátor produktu (artiklu) v tabulce articles. Tento identifikátor je přiřazen do proměnné.

Obsahuje-li záznam celé jméno kategorie, je vytvořena instance třídy Category a její funkce getIdCategory s předanými parametry vrací identifikátor kategorie z databázové tabulky categories.

Nakonec je zavolána funkce SqlCommand, jejíž parametr je sql příkaz vytvořený funkcí SqlQuery z návratových hodnot funkcí getNames() a getValues().

5.2.2 GetNames()

Tato funkce vrací pole názvů sloupců v databázi, které budou ovlivněny.

5.2.3 GetValue(DataRow Row, string column)

Funkce vrací hodnotu sloupce column záznamu Row.

5.2.4 GetValues()

Tato funkce vrací pole hodnot, které budou uloženy do sloupců obsažených v getNames().

5.3 Indexace produktu (Article.cs)

Třída Article obsahuje funkce pro ověření unikátnosti artiklu(produktu) v celém systému a uložení do databáze. Tato unikátnost je sice ověřována podle názvu produktu, ne ovšem rovností, ale speciální fulltextovou podmínkou zajišťující optimální agregaci položky na produkt.

5.3.1 Article(string article, string description, string man, string img)

Parametr article je předávaný název položky, u kterého má být testována agregace na produkt. Parametr description je předávaný popis položky. Parametr man je název výrobce a img obsahuje url adresu obrázku přiřazenému k produktu.

Tato funkce přečte všechny potřebné údaje z parametrů row a uloží si je do proměnných, aby s nimi mohly dále pracovat ostatní funkce třídy.

5.3.2 GetIdArticle()

Funkce napřed ověřuje, zda-li je již zaindexován v systému produkt s daným názvem. Toto ověření striktně porovnává rovnost. Pokud je nalezen záznam v tabulce articles, funkce vrátí identifikátor nalezeného záznamu.

Pokud není nalezen žádný záznam, spustí se znovu hledání produktu, tentokrát ne produktu s úplně stejným názvem, ale produktu s co nejpodobnějším názvem. Toto hledání probíhá pomocí sql fulltextové podmínky, která je sestavena na základě názvu produktu. Tato podmínka často najde více vyhovujících záznamů, proto jsou záznamy seřazeny podle proměnné RANK v každém záznamu a na konci je vrácen záznam s největší hodnotou proměnné RANK. Tuto proměnnou vytváří automaticky ms sql fulltextový engine u každého záznamu při použití funkce CONTAINSTABLE nebo FREETEXTTABLE v sql dotazu, proměnná nabývá hodnot od 0 do 1000 a reprezentuje celkovou shodnost daného záznamu s fulltextovou podmínkou. Je-li na konci fulltextového vyhledávání vrácen záznam, funkce getIdArticle vrátí identifikátor tohoto záznamu.

Není-li ani po této operaci nalezen odpovídající záznam v tabulce articles, pokračuje se dále v tvorbě nového produktu. Je vytvořena instance třídy Manufacturer a zavolána její funkce getIdManufacturer s parametrem udávající název výrobce (viz Manufacturer), která vrací identifikátor daného výrobce v tabulce manufacturers. Nakonec je vytvořen v tabulce articles nový záznam a funkce getIdArticle vrátí identifikátor tohoto nově vytvořeného záznamu.

5.3.3 Fulltext(string text)

Funkce Fulltext transformuje parametr text, což je název produktu na fulltextovou sql vyhledávací podmínku.

5.3.4 GetNames()

Tato funkce vrací pole názvů sloupců v databázi, které budou ovlivněny.

5.3.5 GetValues()

Tato funkce vrací pole hodnot, které budou uloženy do sloupců obsažených v getNames().

5.4 Indexace výrobce (Manufacturer.cs)

Třída Manufacturer obsahuje funkce pro ověření unikátnosti výrobce v rámci celého systému a uložení do databáze.

5.4.1 Manufacturer(string Name)

Funkce Manufacturer pouze přiřadí parametry do proměnných uvnitř třídy. Jediný, konstruktoru předávaný, parametr Name obsahuje název výrobce.

5.4.2 GetIdManufacturer()

Funkce getIdManufacturer nejprve testuje, zda-li již v tabulce manufacturers existuje záznam s daným názvem výrobce. Pokud ano, funkce vrátí identifikátor nalezeného záznamu s hledaným názvem výrobce, v opačném případě je pomocí funkce SqlCommand vytvořen v tabulce nový záznam a funkce getIdManufacturer nakonec vrací identifikátor tohoto nově vytvořeného záznamu.

5.4.3 GetNames()

Tato funkce vrací pole názvů sloupců v databázi, které budou ovlivněny.

5.4.4 GetValues()

Tato funkce vrací pole hodnot, které budou uloženy do sloupců obsažených v getNames().

5.5 Indexace kategorie (Category.cs)

Třída Category obsahuje funkce pro ověření unikátnosti kategorie v rámci celého systému a uložení do databáze.

5.5.1 Category(string Full, int Depth)

Konstruktor třídy Category, tato funkce pouze přiřadí parametry do proměnných uvnitř třídy.

Předávaný parametr Full je úplný název kategorie ve stromové struktuře. Je to řetězec získaný z xml tagu <FULLCATEGORYTEXT> importovaného souboru.

Celočíselný parametr Depth je počáteční adresář stromové struktury kategorie, od kterého se má text dané kategorie použít.

5.5.2 GetIdCategory()

Funkce getIdCategory napřed upraví řetězec fullcategorytext tak, aby začínal takovým adresářem, který je vzdálen od začátku o hodnotu parametru Depth. Po té se testuje, zda-li již v tabulce categories existuje záznam s daným názvem kategorie. Pokud ano, funkce vrátí identifikátor nalezeného záznamu, v opačném případě je pomocí funkce SqlCommand vytvořen v tabulce nový záznam a funkce getIdCategory vrátí identifikátor tohoto nově vytvořeného záznamu.

5.5.3 GetNames()

Tato funkce vrací pole názvů sloupců v databázi, které budou ovlivněny.

5.5.4 GetValues()

Tato funkce vrací pole hodnot, které budou uloženy do sloupců obsažených v getNames().

5.6 Konstrukce a vykonávání SQL dotazů (Sql.cs)

Třída Sql obsahuje podpůrné funkce pro generování a vykonávání dotazů a pro kontrolu vkládaných dat, aby nedošlo k vytvoření sql dotazu se špatnou syntaxí. Z této třídy dědí všechny ostatní třídy uvedené výše. To znamená že následující funkce jsou instancím těchto tříd dostupné.

5.6.1 CheckChars(string chars)

Funkce přepisuje nevyhovující znaky parametru chars tak, aby nedošlo k vytvoření sql dotazu se špatnou syntaxí a vrací takto upravený řetězec. Tato funkce se používá před generováním dotazu ve kterém se ukládají textové data do databáze.

5.6.2 SqlQuery(string Table, string[] Names, string[] Values)

Návratová hodnota této funkce je text sql dotazu, který vkládá hodnoty řetězců pole Values do sloupců se jmény předanými polem Names do databázové tabulky s názvem předaným parametrem Table.

5.6.3 SqlCommand(string Cmd)

Vykonání sql dotazu předaného parametrem Cmd. Pokud jde o dotaz, který přidává záznam do tabulky v databázi, návratová hodnota je identifikátor takto vloženého nového záznamu.

5.6.4 LastID(string Cmd)

Parametrem je text vkládacího sql dotazu (insert), funkce upraví tento text tak, že na konec přidá sql příkaz, který slouží k získání identifikátoru naposledy vloženého záznamu. Nakonec je vykonána výše uvedená funkce SqlCommand, která vrátí zmíněný identifikátor vloženého záznamu.

6 UŽIVATELSKÉ ROZHRAŇÍ

6.1 Rozvržení

Program je logicky rozčleněn podle obsahu každé stránky uživatelského rozhraní, které generuje systém. Funkce a samotný kód každé stránky jsou uloženy v souborech .aspx v adresáři Asp. Každá stránka je generována právě jedním z těchto souborů.

6.1.1 Použité serverové ovládací prvky ASP.NET2

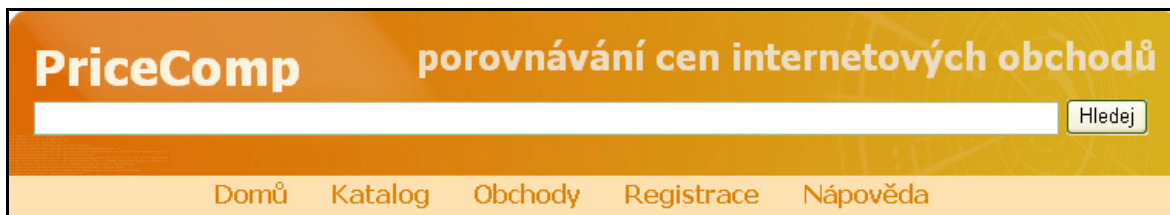
Pro získávání dat z databáze je na každé stránce vložen jeden nebo více serverových ovládacích prvků `asp:SqlDataSource`, jejichž sql dotazy jsou upravovány funkcemi a metodami obsaženými v jednotlivých souborech uživatelského rozhraní (.aspx). Pro výpis a formátování dat, které vrací serverové ovládací prvky `asp:SqlDataSource` jsou použity ovládací prvky `asp:Repeater`. Tyto prvky prochází každý záznam vrácených dat a nakládají s ním podle stanoveného algoritmu, který obsahují.

6.1.2 Vkládání obsahu do šablony

Problém vkládání obsahu do společné šablony všech stránek, je realizován ve verzi Asp.Net2 pomocí master pages. V master stránce je zapsán kód, který má být stejný na všech stránkách a ostatní stránky dodávají pouze obsah. Master stránka pomocí ovládacího prvku `asp:ContentPlaceHolder` určí jedno nebo více míst, kam je možné vkládat obsah a ostatní stránky pomocí prvku `asp:Content` vloží na master stránku právě svůj obsah.

V případě aplikace PriceComp obsahuje Master stránka záhlaví a zápatí celkového vzhladu. V záhlaví je umístěn název projektu s mottem ve formě víceúrovňových nadpisů, textové pole pro fulltextové vyhledávání s tlačítkem hledej (viz. fulltextové vyhledávání) a menu pro přehlednou navigaci obsahující přímé odkazy na všechny stránky projektu. V zápatí je vloženo opět menu, tentokrát menším písmem a dodatek k projektu s autorským označením. Obrázek pozadí (background) v záhlaví stránek byl vytvořen v grafickém programu Adobe Photoshop CS.

Menu je definováno v souboru Web.sitemap. Pomocí jazyka xml se zde definuje struktura menu a jeho jednotlivé prvky. Každý prvek obsahuje název prvku, popis a url adresu, na kterou povede výsledný vygenerovaný odkaz.



Obrázek 5. Záhloví stránek



Obrázek 6. Zápatic stránek

6.1.3 Jednotný vzhled

Jednotný vzhled všech stránek projektu je docílen formátováním pomocí kaskádových stylů (CSS). Kaskádové styly jsou uloženy v samostatném souboru PriceComp.css, což umožňuje jejich aplikaci na každou stránku projektu a globální úpravy vzhledu všech stránek na jednom místě.

```
body
{
    font-family: Tahoma, Verdana, Helvetica, Helvetica-Narrow, sans-serif;
    margin: 10px;
    font-size: 13px;
    text-align: center;
    line-height: 130%;
}

#main
{
    width: 760px;
    text-align: left;
}

a
{
    color: #1f1fb6;
    text-decoration: underline;
}
```

Obrázek 7. Ukázka formátování pomocí CSS stylů

6.2 Úvodní strana (Home.aspx)

Úvodní strana celého systému. Obsahuje strom všech hlavních kategorií a jejich podkategorií a výpis několika náhodně vybraných produktů s obrázky.

6.2.1 Výpis hlavních kategorií

Vypsání kategorií slouží jako přímý odkaz do katalogu produktů s daným parametrem category. Pro získání kategorií z databáze je na stránce vložena komponenta SqlDataSource, jejíž sql dotaz generuje funkce TreeGen. Zobrazení hlavních kategorií probíhá pomocí funkce RootCategory, která testuje přechody na následující hlavní kategorii.



Obrázek 8. Výpis hlavních kategorií na úvodní stránce

6.2.2 Náhodný výpis produktů

Na hlavní straně vpravo je výpis deseti náhodných produktů. Každý záznam výpisu obsahuje náhled obrázku a název produktu. Jednotlivé záznamy slouží jako odkazy pro rychlý přechod na stránku s detailem produktu.



Obrázek 9. Náhodný výpis produktů s obrázkem

6.3 Katalog produktů (List.aspx)

6.3.1 Používané parametry

Nejdůležitější funkcí katalogu je zobrazení výpisu produktů. Tento výpis není nikterak náhodný, řídí se vyhledávacími parametry zadanými uživatelem pomocí odkazů na stránce a formuláře pro vyhledávání, které jsou předávány přes url adresu za jménem souboru volané stránky. Pomocí funkce GetLoad jsou parametry v aplikaci přečteny.

Parametr v url adrese	Význam parametru
search	řetězec pro fulltextové vyhledávání
category	kategorie, ve které má být započato vyhledávání produktů
manufacturer	identifikátor zvoleného výrobce
orderby	sloupec v databázové tabulce použitý při řazení výpisu
direction	směr řazení výpisu (vzestupně, sestupně)
page	číslo zobrazované strany výpisu produktů

Tabulka 7. Parametry katalogu produktů předávané přes url adresu stránky

6.3.2 Fulltextové vyhledávání

Fulltextové vyhledávání je pro uživatele nejjednodušší a nejrychlejší způsob nalezení hledaného produktu.

Základní princip fulltextového vyhledávání lze snadno pochopit pomocí analogie s běžnou odbornou knihou, která obsahuje rejstřík. Představme si tlustou knihu o velkém množství stran a úkol najít všechny zmínky o nějaké osobě. Jednou z možností je čtení od první do poslední stránky. Byla by to však nejspíš práce na několik dní. Daleko rychleji budeme hotovi, když se podíváme do rejstříku, ve kterém je požadovaná informace uložena přesně ve tvaru, v jakém ji potřebujeme. Fulltextový klíč sloupce v tabulce je v podstatě rejstřík všech slov v tomto sloupci obsažených.

Katalog produktů disponuje fulltextovým vyhledáváním které probíhá ve fulltextovém klíči sloupce article tabulky articles. Samotná fulltextová funkce vyhledává takové záznamy, kde se vyskytují všechny slova zadaného řetězce search.

původní text: <i>nokia 1600 stříbrná</i> text ve výsledné podmínce fulltextového vyhledávání sql datazu: <i>"nokia*" AND "1600*" AND "stříbrná*"</i>

Obrázek 10. Příklad transformace zadaného textu pro sql dotaz

6.3.3 Navigace

Navigace zobrazuje úroveň zanoření zobrazované kategorie. Každá úroveň je zároveň odkaz na danou kategorii což umožňuje přehlednou orientaci v kategoriích a snadnou práci s katalogem.

[Vše](#) > [GPS - navigace](#) > [GPS SW PRO PDA](#)

Obrázek 11. Navigace stromem kategorií v katalogu

6.3.4 Kategorie

Systém uchovává informace o kategoriích ve stromové struktuře, kterou interpretuje ve výpisu produktů v katalogu.

Na levé straně katalogu produktů je seznam kategorií, které jsou přímo podřazené kategorii zadané v parametru category a zároveň obsahují produkty, jejichž název splňuje podmínku fulltextového vyhledávání.



Obrázek 12. Výpis kategorií v katalogu

6.3.5 Výrobci

Pod výpisem kategorií jsou vypsaní všichni výrobci produktů splňujících zadané vyhledávací kritéria předané přes parametry url adresy. Seznam výrobců obsahuje takové výrobce, kteří vyrábějí produkty nalezené a zobrazované v seznamu produktů. Výpis výrobců je seřazen vzestupně podle názvu.



Obrázek 13. Výpis výrobců v katalogu



6.3.6 Seznam produktů

Generování veškerých dotazů do databáze na základě vstupní parametrů je prováděno v metodě `SqlDataSource_Init`. Tato metoda je volána při inicializaci jedné z komponent `Repeater`, které prochází každý záznam vrácených dat z databáze komponentou `SqlDataSource`.

Na základě vstupních parametrů předaných přes url adresu metoda `SqlDataSource_Init` vytvoří vyhledávací parametry, které jsou aplikovány do vyhledávacího sql dotazu. Dotaz je omezen maximálním počtem vrácených záznamů kvůli

omezenému počtu vypisovaných produktů. Pokud by dotaz neobsahoval toto omezení, vypisovaly by se na stránce najednou i všechny produkty z databáze, kterých může být i několik set tisíc. Zároveň je sql dotaz doplněn o počet záznamů které mají být přeskočeny kvůli umožnění stránkování celého výpisu.

Výpis produktů které vyhovují vyhledávacím parametrům je vypisován do html tabulky. Záznamy tabulky obsahují popis daného produktu a název produktu, který je zároveň odkazem na stranu s detailem produktu.

řadit podle: názevu data přidání		směr: vzestupně sestupně
	<p>Acer c510 Acer c510 PDA s dotykovým displejem, GPS anténou a operačním systémem Microsoft Windows Mobile 5.0. Je vybaven otočnou BlueTooth a slotem na karty SD/MMC. procesor: Samsung S3C2442 300 MHz paměť: 64 MB SDRAM, 128 MB Flash ROM displej: TFT 2,8" 240 x 320, 65.000 barev, dotykový, podsvícený (se zvýšeným jasem), transflexní slot pro paměťové karty: SD/MMC karty s podporou SDIO operační systém: Micr...</p>	
	<p>Acer c530 Acer c530 PDA s dotykovým displejem, GPS anténou a operačním systémem Microsoft Windows Mobile 5.0. Je vybaven otočnou BlueTooth, Wi-Fi a slotem na karty SD/MMC. procesor: Samsung S3C2442 300 MHz paměť: 64 MB SDRAM, 128 MB Flash ROM displej: TFT 2,8" 240 x 320, 65.000 barev, dotykový, podsvícený (se zvýšeným jasem), transflexní slot pro paměťové karty: SD/MMC karty s podporou SDIO operační systé...</p>	
Strana 1 / 1		

Obrázek 14. Výpis produktů v katalogu

6.3.7 Řazení

Každý tabulkový výpis na internetu obsahující větší počet záznamů by měl uživateli nabídnout možnost řazení podle kritérií, které tyto záznamy obsahují. Je tím dosaženo přehlednějšího zobrazování informací a uživatel se tam dříve dostane k hledané informaci.

Výpis seznamu produktů lze řadit podle názvu nebo data přidání produktu do databáze. Řazení podle těchto kritérií je navíc možno nastavit vzestupně nebo sestupně. Parametry, podle kterých je řazení seznamu produktů koncipováno jsou předány přes url adresu. Na základě těchto parametrů je upraven dotaz do databáze, vše spočívá v modifikaci klauzule ORDER BY daného sql příkazu.

6.3.8 Stránkování


Výpis produktů je kvůli možnému obrovskému počtu záznamů stránkovan. Ovládání stránkování se nachází přímo pod tímto výpisem. Je zde zobrazeno číslo aktuálně zobrazené strany a počet stran tohoto výpisu. Zároveň jsou zde odkazy pro přechod na následující případně předchozí stranu. Číslo strany je předáváno opět pře url adresu jako parametr, na základě kterého je upravován sql dotaz.

6.4 Detail produktu (Detail.aspx)

Obsahuje informace o vybraném produktu a seznam k tomuto produktu agregovaných položek obchodů seřazených podle ceny za položku. Číslo produktu se předává pomocí parametru article v url adrese.

6.4.1 Informace o produktu

Zobrazované informace o vybraném produktu jsou název, popis a obrázek pokud je u produktu obsažen.

<p>ASUS P525</p> <p>ASUS P525 je elegantní a malé PDA s dotykovým displejem a telefonem s operačním systémem Microsoft Windows Mobile 5.0 Phone Edition. Je vybaven numerickou klávesnicí, GSM, GPRS, Bluetooth, Wi-Fi, 2 MPix fotoaparát a slot na karty miniSD. procesor: Intel PXA 270 416MHz paměť: 64 MB hlavní paměti SDRAM pro spuštění aplikací, 128 MB paměti flash ROM displej: TFT QVGA 2.8", rozlišení 240 x 320, 64.000 barev, dotykový slot pro paměťové karty: miniSD operační systém: Microsoft Windows Mobile 5.0 Phone Edition síť: GSM 900 / GSM 1800 / GSM 1900, GPRS Class 10 komunikace: Bluetooth v2.0, Wi-Fi 802.11b, miniUSB port v1.1, IrDA numerická klávesnice – skládá se z 5 x 4 tlačítek (12 scodných s klávesnicí mobilních telefonů, 8</p>	
--	---

Obrázek 15. Detailní informace o produktu

6.4.2 Seznam agregovaných položek

Seznam agregovaných položek k vybranému produktu se vypisuje do html tabulky. Záznamy jsou seřazené podle ceny s dph a každý záznam obsahuje následující informace o položce:

obchod - název obchodu a zároveň odkaz na stranu s detailem obchodu

produkt - název produktu a přímý odkaz na stranu s danou položkou v obchodě

cena - cena s dph, podle které je celý seznam seřazený

odkaz - stejný odkaz jako v poli produkt, ale s intuitivním popisem „do obchodu“

Obchod	Produkt	Cena vč. DPH	Odkaz
ahmobile	ASUS P525	13689,- Kč	do obchodu
sunnysoft.cz	Asus P525	14256,- Kč	do obchodu

Obrázek 16. Seznam agregovaných položek k produktu

6.5 Seznam obchodů (Shops.aspx)

Seznam obchodů je tabulkový výpis všech obchodů zaregistrovaných do systému. Z těchto obchodů jsou importovány ceníky, jejichž obsah se indexuje do tabulek databáze systému. Každý záznam vypisovaného seznamu obsahuje název a popis obchodu. Název obchodu je zároveň odkaz na stránku s detailním popisem obchodu (Shop.aspx). Předává se parametr obsahující identifikátor obchodu.

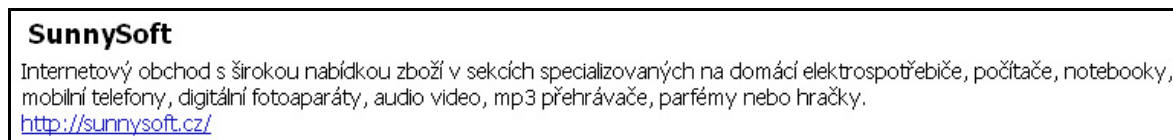
Seznam obchodů	
Název obchodu	Popis
Testovací obchod	Největší specializovaný sportovní eshop. Hokejová výstroj, squash, tenis, badminton, kolečkových bruslí a in-line, nosiče, snowboard, fotbal, sportovní výživa. Garance nejnižší ceny.
AhMobile	Internetový obchod s kamennou prodejnou v Zábavním Centru Černý Most. Prodáváme vše pro Nintendo DS, Nintendo Wii, Nintendo GameCube, GameBoy, PlayStation 2, PlayStation 3, PlayStation Portable, počítačové hry a RC modely.
SunnySoft	Internetový obchod s širokou nabídkou zboží v sekcích specializovaných na domácí elektrospotřebiče, počítače, notebooky, mobilní telefony, digitální fotoaparáty, audio video, mp3 přehrávače, parfémů nebo hračky.
GsMobil.cz	mobilní telefony
mobilprofik.com	mobilní telefony

Obrázek 17. Výpis registrovaných obchodů do systému

6.6 Detail obchodu (Shop.aspx)

V detailním zobrazení obchodu se vypisuje název a popis obchodu a internetová adresa. Text adresy slouží jako přímý odkaz na úvodní stránku zobrazovaného internetového obchodu.

Tato stránka je volána vždy s parametrem nesoucím identifikátor obchodu. Podle tohoto parametru jsou vybrána z databáze data odpovídající obchodu s daným identifikátorem.



Obrázek 18. Detail vybraného obchodu

6.7 Registrace obchodu (Register.aspx)

6.7.1 Formulář registrace

Tato strana slouží k registraci nového obchodu do systému. Obsahuje formulář s textovými poli pro vyplnění údajů a tlačítko odeslat, po jehož stisku se kontroluje správnost všech zadaných údajů.

Vyplňte prosím všechny povinné údaje	
* povinné údaje	
* Název obchodu:	<input type="text"/>
* Popis:	<input type="text"/>
* WWW adresa:	<input type="text"/>
* Adresa datového souboru:	<input type="text"/>
* Kontaktní osoba:	<input type="text"/>
Telefon:	<input type="text"/>
* Email:	<input type="text"/>
* Heslo:	<input type="password"/>
* Heslo znovu:	<input type="password"/>
<input type="button" value="Uložit"/>	

Obrázek 19. Registrace obchodu do systému

6.7.2 Kontrola a uložení dat

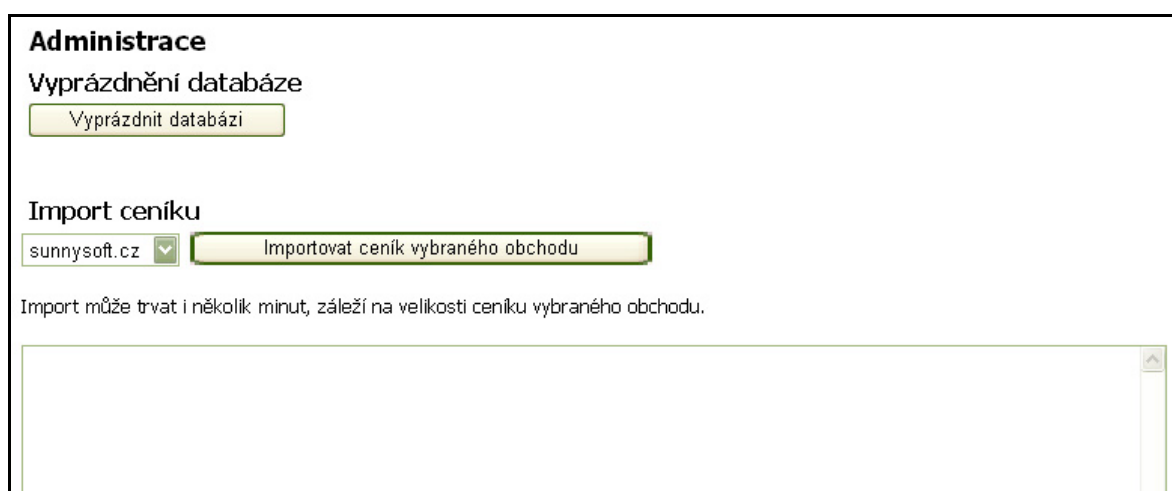
Při odeslání formuláře se kontroluje, zda-li jsou zadány všechny povinné údaje a jestli se shoduje heslo s heslem pro potvrzení. Pokud validace najde špatně zadaný údaj, vypíše se nad formulář chybová hláška s názvem pole, které bylo špatně vyplněno. Jsou-li všechny údaje správně vyplněny a validace nenajde žádnou chybu, uloží se záznam do databáze, konkrétně do tabulky obchodů (shops).

6.8 Nápověda (Help.aspx)

Stránka s nápovědou obsahuje popis datového souboru pro import položek. Je zde zobrazen příklad XML souboru, popis všech jednotlivých elementů a seznam povinných elementů položky, viz. formát datového souboru.

6.9 Administrace (Admin.aspx)

Administrační část uživatelského rozhraní slouží pro manuální obsluhu systému administrátorem. Administrátor má k dispozici funkci pro smazání databáze a funkci pro import ceníku obchodu, který si zvolí. Po provedení každé funkce systém informuje administrátora o správnosti provedení této funkce a o počtu ovlivněných řádků v databázi. Tyto informace se vypisují v textovém poli pod tlačítka administrátorských funkcí.



The screenshot shows the 'Administrace' (Administration) section of the application. It features two main functional areas:

- Vyprázdnění databáze** (Empty database): A single button labeled 'Vyprázdnit databázi' (Empty database).
- Import ceníku** (Import price list): A dropdown menu currently showing 'sunnysoft.cz' and a button labeled 'Importovat ceník vybraného obchodu' (Import price list of selected store).

Below these controls, a text label states: 'Import může trvat i několik minut, záleží na velikosti ceníku vybraného obchodu.' (Import may take several minutes, depending on the size of the selected store's price list). At the bottom of the interface is a large, empty text area for displaying system messages or status updates.

Obrázek 20. Administrační rozhraní

6.9.1 Vyprázdnění databáze

Vyprázdnění databáze v tomto případě znamená vyprázdnění všech tabulek a indexů mimo tabulku shops, ve které jsou uloženy informace o obchodech včetně url adresy ceníku každého obchodu, ze které se stahují data pro import.

Tato funkce slouží především k dokonalému otestování agregace položek. Při každé úpravě fulltextové podmínky agregační funkce bylo totiž potřeba vyprázdnit databázi, znovu importovat ceníky několika obchodů a porovnáním vzniklých produktů a k nim agregovaných položek vyhodnotit optimálnost testované podmínky.

6.9.2 Import ceníku vybraného obchodu

Výběrové pole (selectbox) na stránce obsahuje seznam všech obchodů. Díky selectboxu je možno označit právě jeden obchod. Stisknutím tlačítka „importovat ceník vybraného obchodu“ je pak započata operace importu ceníku obchodu, který je označen ve výběrovém poli.

ZÁVĚR

Vytvořený systém pro hodnocení výhodnosti nákupu by měl uživatelům poskytnout jednoduché nalezení hledaného produktu a k němu ceny z různých obchodů při minimální náročnosti na ovládání. Při nasazení systému do oblasti s nejprodávanějším sortimentem zboží, by při větším množství obchodů uživatelé aplikaci ocenili a systém by tak brzy získal oblibu.

Nejsložitějším bodem práce byla implementace indexace dat importovaných ceníků. Tyto ceníky totiž mohou obsahovat velké množství dat a algoritmus indexace pak běží delší dobu, čímž zatěžuje počítač (server), na kterém je spuštěn. Stěžejní proto bylo vybrání vhodné fulltextové podmínky sql dotazu při agregaci položek tak, aby byly položky agregovány s přípustnou mírou odchylky v názvu položky a aby podmínka nebyla nadbytečně složitá a nedocházelo tím tak ke zbytečnému zatěžování prostředků.

Funkce pro agregaci položek byla od počátku vyvíjena způsobem, aby se při různých nasazeních systému dala podle počtu uchovávaných záznamů v databázi a v neposlední řadě i podle druhu sortimentu zboží upravit citlivost této funkce.

Jak implementace importu ceníku, tak celé uživatelské rozhraní je navrženo tak, aby byl systém jednoduše rozšiřitelný. Nejpříhodnější rozšíření by mohlo být zavedení hodnocení evidovaných produktů a obchodů uživateli. Toto hodnocení by mohlo probíhat formou udělování známek z nějaké stupnice, například 1-5. Průměrná známka hodnocení každého produktu by pak mohla sloužit i jako jedno z řadících kritérií výpisu seznamu produktů. Při větším množství zaregistrovaných obchodů by bylo vhodné vytvořit zautomatizování importu ceníků. Ceníky by se automaticky stahovaly pravidelně po několikadenních intervalech, případně každý obchod by měl individuálně určenou periodu stahování podle aktualizace ceníku obchodníkem.

ZÁVĚR V ANGLIČTINĚ

The System for an evaluation of the shopping convenience was set up to offer its users an easy way to find out a product and its prices at any shops. The main stress was put on the simply use. The use of this application as a virtual shopping centre with wide range of wares at many shops would be a positive for the users and the system might become popular.

The most difficult issue was the implementation of indexing data of imported price lists. These price lists may contain a great amount of data, then an algorithm is running longer which causes a ballast to the server. The central task was to choose the suitable full-text condition of the sql query on condition that the aggregation of the item names has a following rule: there is a tolerable deviation in the name of item and, at the same time, the condition is not uselessly complicated and does not occupy a computer too much.

The function for the aggregation of items was compiled with the guarantee of the ability to set its own sensitivity. Depending on the usage of this system, by the sensitivity we mean a possibility of settings according to the amount of entries and the sort of products.

Both the implementation of the price list data and the user interface is designed with the possibility of further enlargement. It might be a practical addition, when the introduction of rating of products and shops takes place. This rating could be defined by giving a mark (e.g. 1 - 5), the average mark of a certain product could serve as a priority for sorting products. In the case of registering greater amount of shops, it would be suitable to create an automatic update of price list. A price list could be automatically downloaded at regular intervals or each shop could have a certain interval of updating, set by the seller.

SEZNAM POUŽITÉ LITERATURY

- [1] BILL, Evjen, et al. ASP.NET 2.0 : Programujeme profesionálně. [s.l.] : Computer Press, 2007. 1224 s. ISBN 978-80-251-1473-5.
- [2] WALTER, Stephen. ASP.NET 2.0 Unleashed . Los Angeles, USA : Sams, 2006. 1992 s. Bk&CD-Rom edition. ISBN 0672328232.
- [3] Sharp, J., Jagger, J., Microsoft Visual C#.NET Step by Step. ISBN 0-7356-1909-3
- [4] Liberty, J.: Programming C# 3rd Edition, O'Reilly 2003. ISBN 0-596-00489-3.
- [5] ESPOSITO, Dino. XML efektivní programování pro .net. Praha : Grada Publishing, 2004. 596 s. ISBN 80-247-0775-6.
- [6] YOUNG, Michael J. XML krok za krokem. Miroslav Kučera; Ing. Aleš Thiemel. 1. vyd. Praha : Mobil Media, 2002. 472 s. ISBN 80-7226-170-3.
- [7] PROKOPOVÁ, Zdenka, Ing. CSc. Databázové systémy, přednášky, Zlín 2005.
- [8] Adobe Photoshop CS – oficiální výukový kurz. 1. vyd. Praha: Soft Press 2004. ISBN 80-86497-71-2.
- [9] PROKOP, M.: CSS: Kaskádové styly pro webdesignéry. Mobil Média, 2003. ISBN 80-251-0487-7.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ANSI	American National Standards Institute
ASP	Active Server Pages
CLR	Common Language Runtime
CSS	Cascading Style Sheets
DBMS	DataBase Management System
HTML	HyperText Markup Language
SGML	Standard Generalized MarkUp Language
SQL	Structured Query Language
SŘBD	System Řízení Báze Dat
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
XWG	XML Working Group

SEZNAM OBRÁZKŮ

Obrázek 1. Vztah mezi klientským prohlížečem a serverem s ASP.NET	10
Obrázek 2. Architektura .NET Framework.....	11
Obrázek 3. Ukázka datového souboru pro import dat	20
Obrázek 4. Schéma struktury databáze	22
Obrázek 5. Záhloví stránek	34
Obrázek 6. Zápatí stránek	34
Obrázek 7. Ukázka formátování pomocí css stylů.....	34
Obrázek 8. Výpis hlavních kategorií na úvodní stránce	35
Obrázek 9. Náhodný výpis produktů s obrázky	36
Obrázek 10. Příklad transformace zadaného textu pro sql dotaz.....	37
Obrázek 11. Navigace stromem kategorií v katalogu	37
Obrázek 12. Výpis kategorií v katalogu.....	38
Obrázek 13. Výpis výrobců v katalogu.....	38
Obrázek 14. Výpis produktů v katalogu	39
Obrázek 15. Detailní informace o produktu	40
Obrázek 16. Seznam agregovaných položek k produktu	41
Obrázek 17. Výpis registrovaných obchodů do systému	41
Obrázek 18. Detail vybraného obchodu.....	42
Obrázek 19. Registrace obchodu do systému	42
Obrázek 20. Administrační rozhraní.....	43

SEZNAM TABULEK

Tabulka 1. Popis jednotlivých elementů XML souboru	21
Tabulka 2. Struktura databázové tabulky shops.....	23
Tabulka 3. Struktura databázové tabulky items	23
Tabulka 4. Struktura databázové tabulky articles	24
Tabulka 5. Struktura databázové tabulky categories.....	24
Tabulka 6. Struktura databázové tabulky manufacturers	24
Tabulka 7. Parametry katalogu produktů předávané přes url adresu stránky	36

SEZNAM PŘÍLOH

P 1 : Disk CD s aplikací a bakalářskou prací