



Ekonomická
fakulta
Faculty
of Economics

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Ekonomická fakulta

Katedra aplikované matematiky a informatiky

Bakalářská práce

Doporučovací systém pro vybraný e-shop

Vypracoval: Pavel Vosátka

Vedoucí práce: doc. Ing. Ladislav Beránek, CSc.

České Budějovice 2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Pavel VOSÁTKA**
Osobní číslo: **E12258**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Ekonomická informatika**
Název tématu: **Doporučovací systém pro vybraný e-shop**
Zadávající katedra: **Katedra aplikované matematiky a informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Doporučovací systémy jsou programy, které se danému uživateli pokouší nabízet různé předměty v rámci např. e-shopu, které by pro něj mohly být zajímavé. Tyto systémy mohou být užitečné pro uživatele, který se rozhoduje mezi mnoha takovými objekty a nechce se zabývat procházením informací o všech dostupných objektech.

Cílem práce je navrhnout a realizovat webovou aplikaci, která by na základě své databáze a podle zadání uživatele pomohla vybrat a nabídnout určitý předmět jeho zájmu. Student musí implementovat, navrhnout algoritmus pro doporučovací systém a vytvořit určitou webovou aplikaci. Příkladem aplikace může být web o rybaření, který by obsahoval popis jednotlivých technik rybolovu. Doporučovací systém by pak měl pomoci při výběru vláčecí nástrahy po zadání určitých parametrů.

Metodický postup:

1. Studium odborné literatury, seznámení se s principy doporučovacích systémů.
2. Návrh vhodných technik pro doporučovací systém - algoritmy, implementace.
3. Praktická část - vytvoření vlastní webové aplikace, která bude obsahovat modul pro doporučení.
4. Závěry a obecná doporučení.

Rozsah grafických prací:

Rozsah pracovní zprávy: **40 - 50 stran**

Forma zpracování bakalářské práce: **tištěná**


Seznam odborné literatury:

1. Robin BURKE. Hybrid Web Recommender Systems.
<http://josquin.cs.depaul.edu/twiki/pub/Csc594/ArticleList/burke-hybrid.pdf>
2. HOGAN, B. P. HTML5 a CSS3: výukový kurz webového vývojáře. Vyd. 1. Brno: Computer Press, 2011.
3. HUETER, Geoffrey J., Steven C. QUANDT a Noble H. HUETER. Universal system and method for representing and predicting human behavior patent description.
<http://www.freepatentsonline.com/20090248599.pdf>
4. KUNEGIS, Jerome, Alan SAID a Winfried UMBRATH. The Universal Recommender. September 2009,
<http://adsabs.harvard.edu/abs/2009arXiv0909.3472K>
5. LECKY-THOMPSON, Ed a Steven D. NOWICKI. PHP 6: programujeme profesionálně. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010.
6. SCHAFFER, J. Ben, Dan FRANKOWSKI, Jon HERLOCKER a Shilad SEN. Collaborative filtering recommender systems,
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.4520&rep=rep1&t>


Vedoucí bakalářské práce: **doc. Ing. Ladislav Beránek, CSc.**
Katedra aplikované matematiky a informatiky

Datum zadání bakalářské práce: **7. ledna 2014**

Termín odevzdání bakalářské práce: **15. dubna 2015**


doc. Ing. Ladislav Rolínek, Ph.D.
děkan

JIHOČESKÁ UNIVERZITA
V ČESKÝCH BUDĚJOVICÍCH
EKONOMICKÁ FAKULTA
Studentská 13 (20)
370 05 České Budějovice


prof. RNDr. Pavel Tlustý, CSc.
vedoucí katedry

V Českých Budějovicích dne 26. února 2014

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47 zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to – v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

Dne 1. 9. 2015

.....

Pavel Vosátka

Poděkování

Mé poděkování patří doc. Ing. Ladislavovi Beránkovi, CSc. za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnoval.

OBSAH

1	Úvod.....	9
1.1	Motivace.....	9
1.2	Cíl práce	9
1.3	Obsah kapitol	10
1.4	Užívaná terminologie	10
2	Přehled řešené problematiky.....	12
2.1	Uživatelské preference	12
2.1.1	Vymezení a typologie uživatelských preferencí	12
2.1.2	Jak získat informace o uživatelských preferencích?	13
2.2	Doporučovací systémy	17
2.2.1	Typy doporučovacích systémů	17
2.2.2	Znamé doporučovací systémy	21
2.2.3	Problém studeného startu („Cold-start problem“)	26
2.3	Vývojářské nástroje a ostatní technologie.....	26
2.3.1	Databázový systém Microsoft SQL Server	26
2.3.2	Microsoft Visual Studio.....	29
2.3.3	ASP technologie	29
2.3.4	ASP webhosting.....	30
2.3.5	CSS	30
2.3.6	HTML	30
3	Metodika	32
4	Výsledky a jejich diskuse	33
4.1	Architektura.....	33

4.2	Tabulky	34
4.2.1	Produkt.....	34
4.2.2	Sekce.....	34
4.2.3	Hodnocení	35
4.2.4	Parametr	35
4.2.5	ParametrTyp.....	36
4.2.6	ParametrProdukt	36
4.2.7	SekceParametr	37
4.2.8	Doporučujeme.....	37
4.2.9	Databázové schéma.....	39
4.3	Funkce	40
4.3.1	HodnoceniProduktu	40
4.3.2	CenaProduktu.....	40
4.4	Kontrolery (Controllers).....	41
4.4.1	ProduktsController.....	41
4.4.2	SekcesController	43
4.4.3	SelectorController	44
4.4.4	HomeController	44
4.5	Modely (Models).....	44
4.5.1	Vygenerované třídy.....	44
4.5.2	ProduktModel	45
4.5.3	SekceVypis	45
4.5.4	SelectoryHodnotyInterval	45
4.5.5	SelectoryHodotyVycet.....	45
4.5.6	Selector	46

4.5.7	ProduktyVypis	46
4.6	Pohledy (Views).....	46
4.6.1	VypisProduktu	46
4.6.2	VypisSekci	47
4.6.3	VypisSelectoru.....	48
4.6.4	Produkt.....	49
4.6.5	VypisProduktuJavaScript.....	50
4.6.6	Index	51
4.6.7	Vygenerované pohledy	52
5	Závěr	54
I.	Summary and keywords.....	55
II.	Seznam použitých zdrojů.....	56
III.	Seznam obrázků.....	60
IV.	Seznam zdrojových kódů.....	61
V.	Seznam příloh	62

1 ÚVOD

Tato bakalářská práce se věnuje doporučovacímu systému pro vybraný e-shop. V současné době je zcela běžné, že lidé jsou doma, v zaměstnání i na cestách připojeni k internetu. S tím souvisí i skutečnost, že řada lidí dnes raději nakupuje prostřednictvím e-shopů, než v kamenných obchodech. E-shopů je však velké množství, a tudíž je možné mezi nimi vybírat podle celé řady kritérií. Pro provozovatele e-shopů to pak znamená především velkou konkurenci, od které je potřeba se nějakým způsobem odlišit, aby potenciální zákazník nakoupil právě u konkrétního obchodníka. Stejně tak největší e-shopy v současnosti obsahují tisíce či dokonce desetitisíce položek, které mohou zákazníkovi nabídnout. Na jednu stranu je velkou výhodou e-shopů, že nabízejí širší nabídku zboží než kamenné prodejny, na druhou stranu se nabídka zboží může pro zákazníka stávat nepřehlednou. Právě to je důvodem, proč existují doporučovací systémy, které jsou mezi e-shopy čím dál tím více využívány. Především je nutné klást velký důraz na inovaci a způsoby zjišťování uživatelských preferencí a požadavků.

Doporučovací systémy ovšem nejsou zcela novou záležitostí, neboť mají dlouhou historii. Spolu s rozvojem počítačů a informačních technologií se na počátku 90. let objevilo hned několik technologií, které směřovaly ke zjištění preferencí uživatele a na jejichž základě k nabídce optimálních produktů (Bainbridge, 2004).

1.1 MOTIVACE

Důvodem, proč jsem si zvolil právě toto téma, je skutečnost, že bez doporučvacích systémů se dnes již úspěšný e-shop prakticky neobejde, čímž je toto téma pro internetové obchody stěžejní a velmi aktuální. Mou osobní motivací byl spíše můj životní koníček rybaření, neboť jedním z mých mnoha snů je mít svůj vlastní e-shop s rybářskými potřebami, který bych měl rád plně ve svém vlastním vedení.

1.2 CÍL PRÁCE

Cílem této práce je na základě uživatelských preferencí a charakteristik jednotlivých doporučvacích systémů zvolit optimální doporučovací systém pro vybraný e-shop.

Přínosem práce bude zejména formulace komplexního doporučení pro vybraný e-shop. V obecném rozměru bude přínosem práce také skutečnost, že budou představeny nejvýznamnější doporučovací systémy a budou porovnány jejich charakteristiky za účelem formulace vlastního návrhu. Práce je omezena na vytvoření konkrétní aplikace na jeden e-shop, a tudíž formulované doporučení a návrh se nehodí pro každý e-shop. Veškeré zde uvedené návrhy nelze aplikovat bez toho, aby byly upraveny na míru konkrétnímu e-shopu. Práce je do jisté míry rovněž limitována také předpokládaným rozsahem bakalářské práce, čímž je nutné věnovat se jen vybraným aspektům problematiky.

1.3 OBSAH KAPITOL

Bakalářská práce je standardním způsobem členěna na teoretickou a praktickou část. První kapitolou práce je úvod, druhá kapitola je pak koncipována jako přehled řešené problematiky v odborných zdrojích, kdy je kapitola zaměřena zejména na uživatelské preference (krátkodobé a dlouhodobé), doporučovací systémy (jejich typy a představení známých doporučovacích systémů) a vývojářským nástrojům a ostatním technologiím (databázový systém Microsoft SQL Server, ASP technologie a ASP webhosting). Třetí kapitola je věnována metodice zpracování práce. Čtvrtá kapitola, která tvoří analytickou část práce, je pak zaměřena na výsledky a jejich diskusi (se zaměřením na popis databáze a popis samotné webové aplikace). Poslední pátou kapitolu práce tvoří závěr.

1.4 UŽÍVANÁ TERMINOLOGIE

Z hlediska terminologie je nezbytné vysvětlit zejména pojem doporučovací systémy (anglicky: „*recommender systems*“). Doporučovací systém lze definovat jako automatickou aplikaci, která se vyznačuje tím, že na základě uživateli zpětné vazby doporučuje ke koupi vybrané položky. Taková aplikace umožňuje přesně zacílit zboží na jednotlivé uživatele (potenciální zákazníci), a tím lze pak zvýšit ziskovost e-shopu (Jannach, 2011). Další definice uvádí, že doporučovací systémy jsou personalizovaní informační agenti, kteří poskytují uživateli určité doporučení v podobě návrhů na položky z e-shopu, které mohou být pro uživatele užitečné (Burke, 2007). V oblasti vyhledávání informací, je doporučovací systém definován jako systém, který je schopen nalézt entity v datové sadě, které mohou být zajímavé pro uživatele (Kunegis, Said, & Umbrath, 2009).

Doporučovací systém lze definovat různým způsobem, v zásadě lze shrnout, že se jedná o aplikaci, která uživatelům doporučuje pro ně vhodné produkty z nabídky.

2 PŘEHLED ŘEŠENÉ PROBLEMATIKY

2.1 UŽIVATELSKÉ PREFERENCE

2.1.1 Vymezení a typologie uživatelských preferencí

Velké webové portály, ale i e-shopy dnes obsahují rozsáhlé množství informací. Každý z návštěvníků e-shopu má jiné potřeby a ze zboží a informací o něm ho zajímá pouze vybraná část. Problémem pro e-shop je správně identifikovat potřebu (preferenci) uživatele a zobrazit mu informace, o které má v dané chvíli zájem. O zájmech uživatele nám může více prozradit jeho chování při opakovaných návštěvách webového portálu. Pokud v chování identifikujeme opakující se vzory, můžeme podle nich seskupit podobné uživatele a vytvářet pro ně doporučení (Cockburn & Mckenzie, 2001).

„Uživatelská preference je obvykle definována jako funkce $PU(o): O \times U \rightarrow [0,1]$, která pro konkrétního uživatele U a objekt o z množiny objektů O vrací míru oblíbenosti objektu u uživatele“ (Peška, 2010). Jinak řečeno uživatelská preference bývá nejčastěji chápána jako utility-funkce, která pro konkrétního uživatele a položku vrací míru oblíbenosti položky, kde oblíbenost objektu v případě e-shopů je chápána jako míra ochoty uživatele věc koupit (Peška, 2010).

Uživatelské preference lze členit dle celé řady kritérií, přičemž jedno z nich je jejich členění na preference krátkodobé a dlouhodobé. Krátkodobou preferencí se rozumí taková, která se vztahuje k objektům na základě aktuálního cíle. Pokud to aplikujeme na případ e-shopu, v situaci, kdy má uživatel za aktuální cíl koupit dámské in-line brusle, bude jeho preference těchto bruslí vysoká, zatímco preference pánských in-line bruslí se bude blížit nule navzdory skutečnosti, že uživatel je muž a standardně vyhledává informace o pánských in-line bruslích. Oproti krátkodobé uživatelské preferenci pak stojí preference dlouhodobá. Ta je založena na obecnějších pravidlech, kterými se uživatel obvykle při vyhledávání produktů v e-shopech řídí. Jedná se například o preferenci vybraných kvalitních značek či nižší ceny apod. (Peška, 2010).

Doporučovací systémy se vyznačují tím, že vycházejí právě z preferencí uživatele, proto je jejich identifikace stěžejní otázkou (Hueter, Quandt, & Hueter, 2013). Nepodaří-li se

správně identifikovat preference uživatele, dojde k tomu, že uživatel v e-shopu nenakoupí, nebo nenakoupí tolik produktů, kolik by nakoupil za předpokladů, že by mu byl doporučen další pro něj zajímavý produkt, o kterém se tento uživatel nemusí vůbec dozvědět.

2.1.2 Jak získat informace o uživatelských preferencích?

Zájem o prezentované zboží a k němu souvztažné informace můžeme zjistit porovnáním klíčových slov s modelem uživatele. Musíme však předpokládat, že všechny zobrazené produkty (stránky) uživatele zaujaly. Když víme, jaké produkty uživatele zajímají, můžeme mu doporučit zboží z nabídky e-shopu, které obsahuje příslušná klíčová slova. Klíčová slova můžeme také získat z anotací odkazů, na které uživatel klikl. Kliknutí na odkaz však obecně nemusí vyjadřovat uživatelský zájem o stránku (předmět), na kterou směřuje. Druhým způsobem zjištění zájmu je sledování aktivit uživatele. (Barla & Bieliková, 2009).

Analýza chování uživatele v konkrétním e-shopu

Návštěvníci e-shopu za sebou zanechávají digitální stopu v podobě odkazů, které použili při navigaci mezi stránkami e-shopu. Z každého prohlížení e-shopu je možné získat vektor, jehož složkami jsou jednotlivé stránky e-shopu. Uspořádané jsou podle pořadí, v jakém byly navštíveny. Takto vytvořené posloupnosti nám mohou odhalit aktuální (krátkodobé) preference uživatele, jakož i jeho dlouhodobé návyky (preference) při návštěvách daného e-shopu. Uvedené informace pak dokážeme využít při doporučení odkazů a následně dokumentů, na které vedou. Po analýze dokumentů lze doporučovat konkrétní objekty (produkty e-shopu), které tyto dokumenty obsahují.

Webový portál, na kterém je umístěný e-shop, si je možné představit jako orientovaný graf. Jednotlivé stránky jsou uzly grafu. Mezi stránkami se pohybujeme prostřednictvím hypertextových odkazů, které tvoří orientované hrany grafu. Při používání portálu tak vznikají různé cesty mezi jednotlivými stránkami. O cíli uživatele v rámci konkrétní návštěvy nám může mnoho napovědět již první navštívená stránka e-shopu či první produkt. Na ní si uživatel vybere některý z odkazů, čímž určí cestu v grafu. Každý další použitý odkaz konkretizuje záměr uživatele. V takovém případě mu ve vhodné chvíli můžeme doporučit dokument (produkt) na konci cesty, který ostatní uživatelé považovali za zajímavý. Snížíme mu tím počet odkazů, které musí použít, což uživateli usnadní práci.

Z dlouhodobého hlediska nám analýza posloupností navštívených stránek může prozradit určité zařazení uživatele do jedné z cílových skupin e-shopu. Větší e-shopy často obsahují informace určené různorodým skupinám uživatelů, čemuž je přizpůsobena i navigace. Uživatel si jako první v menu vybere, do které z těchto skupin patří. Když si při každé návštěvě vybere stejnou skupinu, můžeme tuto informaci využít a následně mu přímo doporučit sekce s informacemi a produkty pro jeho skupinu.

V posloupnostech navštívených stránek také můžeme hledat opakující se vzory. Ty nám řeknou více o zvyklostech uživatele při práci s daným e-shopem. Někteří uživatelé více využívají navigaci poskytnutou portálem a pohybují se v kruzích, jiní častěji využívají možnost vrátit se zpět, čímž jejich posloupnosti připomínají schody.

V posloupnostech navštívených stránek identifikujeme tyto základní vzory: (Canter, Rivers, & Storrs, 1985).

1. *Cesta* – posloupnost, v níž se žádná stránka neopakuje;
2. *Kruh* – posloupnost začínající i končící na téže stránce;
3. *Smyčka* – posloupnost procházející už jednou navštívenou stránkou;
4. *Hrot* – posloupnost, ve které se vracíme zpět po stejné trase.

Použitý vzor vypovídá jak o návycích uživatele, tak o jeho aktuálních záměrech. Například větší výskyt hrotů signalizuje, že uživatel hledá konkrétní produkt. Naproti tomu, výskyt smyček a kruhů napovídá, že uživatel se snaží objevit, co se v e-shopu nachází. V případě, že u každého uživatele bude z dlouhodobého hlediska na konkrétním portálu převažovat jeden ze vzorů, můžeme tuto skutečnost využít k rozdělení uživatelů do skupin. V rámci skupin můžeme následně doporučit zajímavé dokumenty a přizpůsobovat navigaci (Milic-Frayling *et al.*, 2004).

Na konkrétních stránkách e-shopu však návštěvníci obvykle provádějí různé akce. Ty lze pak využít k zjištění uživatelského zájmu o prezentované informace či zboží (uživatelských preferencí). Podle typu zájmu lze identifikovat v zásadě tři kategorie akcí. Prvními dvěma typy jsou akce vyjadřující čistě kladný (tisk stránky, přidání do oblíbených položek, zkopírování textu do schránky) nebo záporný (zavření stránky po velmi krátkém čase od zobrazení, zastavení načítání stránky) zájem. Poslední typ představují akce, které mohou

vyjadřovat oba druhy zájmu v závislosti na kontextu, v jakém byly provedeny (čas strávený na stránce, míra pohybu myši, míra rolování stránky apod.) (Holub & Bielíková, 2010).

Akce, jejichž význam závisí na kontextu jejich provedení, porovnáváme s akcemi provedenými na dané stránce e-shopu ostatními uživateli v minulosti. Podle toho také určujeme, zda akce vyjadřuje kladný nebo záporný zájem. Pokud například uživatel strávil na stránce s konkrétním produktem e-shopu nadprůměrně mnoho času oproti ostatním uživatelům, usuzujeme z toho, že produkt uživatele zaujal. Takto určíme zájem o každou viděnou stránku, resp. o každý produkt, který e-shop nabízí a uživatel jej zobrazil (Holub & Bielíková, 2010).

Zájem je možné vyjádřit různými formami, například dvěma hodnotami (stránka s produktem uživatele zaujala/nezaujala) nebo spojitě (reálné číslo na zvolené stupnici odrážející míru zájmu). Takto můžeme určit zájem nejen o samotný produkt (webovou stránku), ale také o objekty na vyšší významové úrovni. Pokud například e-shop informuje o nějakém typu produktů (například produkty na hubnutí), přiřadíme zájem uživatele přímo k produktům na hubnutí. Produkty, které na webu e-shopu zaujaly několik lidí, je dále možné doporučovat dalším uživatelům. Podle zájmu o zhlédnutí produktu lze určit i zájem o produkty e-shopu, které dosud uživatel neviděl s využitím kolaborativního filtrování (pojem bude vysvětlen níže v kapitole 2.2.1).

Zpětná vazba

Pro výpočet vhodnosti konkrétního objektu (produktu e-shopu) pro uživatele, potřebujeme mít k dispozici model uživatele, který je založen na jeho uživatelských preferencích. Model uživatele může být vytvořen z jeho implicitní nebo explicitní zpětné vazby při hodnocení samotných objektů, informací v jeho profilu, nebo různými kombinacemi zmíněných přístupů.

Implicitní získávání uživatelských preferencí

Mnoho systémů personalizovaného doporučení nepožaduje od uživatelů přímo explicitní zpětnou vazbu, aby mohly vytvořit modely uživatele, nebo skupiny. Tento přístup se používá zejména při doporučování jednotlivci, ale našel uplatnění i při doporučování skupině. Implicitní získávání preferencí je realizováno například postupy, které byly

nastíněny výše (analýza pohybu na stránkách e-shopu, délky setrvání na konkrétním produktu, typu navštívených produktů apod.).

Explicitní získávání uživatelských preferencí

Takový způsob získávání údajů o preferencích vyžaduje od uživatelů explicitní zpětnou vazbu, aby mohli být jednotliví uživatelé nebo skupina přiřazeni k určitému modelu, popř., aby model mohl být vůbec vytvořen. Pro získání uživatelských preferencí je proto nutná přímá interakce uživatele se systémem. Typickým příkladem v oblasti e-shopů je třeba systém Heureka, která vyžaduje prostřednictvím zasílaného e-mailu hodnocení e-shopu i produktů, které uživatel nakoupil, a to jak hodnocení v rozmezí 1–5 hvězdiček, tak i hodnocení slovní ve formě komentáře uživatele. Variantou explicitního získávání preferencí je možnost editovat svůj uživatelský profil vytvořen z implicitně získaných údajů. Většina e-shopů totiž umožňuje registraci, některé ji k nákupu dokonce vyžadují, přičemž u registrace je nutné vyplnit základní údaje o uživateli, nicméně v řadě případů je možné vyplnit i doplňující (nepovinné údaje), na jejichž podkladě lze pak identifikovat preference uživatele (Wongchokprasitti & Brusilovsky, 2007).

Zaměření se na negativní zpětnou vazbu uživatele

Tento postup se zaměřuje na eliminování doporučení nevhodných položek. Získávání negativních preferencí uživatelů, například při doporučení k nákupu hudby skupině uživatelů eliminuje z doporučení možnosti, které neodpovídají preferencím kteréhokoliv z uživatelů (Jameson & Smyth, 2007).

Tento postup však může být v určitých situacích kontraproduktivní, protože určité položky doporučení, které jsou negativními preferencemi jednoho uživatele, mohou být preferencemi jiného. V úvahu musíme vzít také množství objektů, ke kterým získáváme zpětnou vazbu. Při doporučení hudby tento aspekt nemusí být natolik klíčový, jelikož uživatel z daného seznamu přiřadí několika nosičům negativní preference. Na druhé straně při doporučení například potravin k nákupu by bylo výhodnější zaměřit se na sbírání pozitivní zpětné vazby, protože jejich počet je většinou mnohem větší. Rozhodování mezi negativní a pozitivní zpětnou vazbou je takto značně závislé na doméně, ve které se doporučení realizuje.

2.2 DOPORUČOVACÍ SYSTÉMY

Po získání preferencí jednotlivých uživatelů – vytvoření modelu uživatele – následuje proces samotného doporučování. Za tímto účelem je potřeba představit jednotlivé doporučovací systémy a jejich typologii.

2.2.1 Typy doporučovacích systémů

Mezi základní typy doporučovacích systémů patří: doporučení založené na obsahu „*Content-based filtering*“, kolaborativní doporučení „*Collaborative filtering*“, doporučení založené na znalostech „*Knowledge-based filtering*“ a hybridní doporučení „*Hybrid recommenders*“, které je obvykle tvořeno kombinací výše uvedených doporučení (Burke, 2007).

Doporučení založené na obsahu („*Content-based filtering*“)

Při doporučení založeném na obsahu je funkce užitečnosti objektu pro uživatele odhadována na základě funkcí užitečností, tzv. podobných objektů. Uživatelé jsou tedy doporučováni objekty s podobnými vlastnostmi jako aktuálně prohlížený objekt. Přístup založený na obsahu tvoří doporučení na základě korelační analýzy objektů, které budou doporučené a objektů, které si uživatelé v minulosti vybrali nebo hodnotili. Tento přístup na jedné straně vyžaduje analýzu obsahu, na druhé straně analýzu uživatelského profilu, který je vybudován pomocí historie jeho interakcí s objekty (Burke, 2007).

Nově přidané objekty jsou ohodnoceny na základě jejich vlastností a podobnosti s již hodnocenými objekty. Možná omezení tohoto přístupu jsou následující (Adomavicius & Tuzhilin, 2005):

- *Omezená analýza obsahu*: při tomto přístupu se doporučují objekty pouze na základě atributů objektů, které jim je nutné explicitně přiřadit. Při doporučení objektů ve formě textu toto omezení není až tak výrazné, na druhé straně například při doporučení hudby nebo videa může být obtížné přiřadit objektům jejich atributy automaticky. Dalším problémem omezené analýzy obsahu je omezená možnost rozlišit kvalitní obsah (například při doporučení produktů při online nakupování), protože pokud jsou objekty reprezentovány stejnou množinou atributů, bude je tato metoda považovat za rovnocenné, bez ohledu na jejich hodnocení uživatelem;

- *Specializace*: při tomto přístupu jsou doporučeny pouze objekty, které maximalizují funkci užitečnosti objektu pro uživatele. Omezení spočívá v omezené možnosti doporučení objektů, které nejsou podobné těm, které uživatel v minulosti hodnotil pozitivně, ale mohly by být pro uživatele zajímavé. Tento problém lze částečně řešit přidáním určité nahodilosti při vybírání objektů;
- *Problém nového uživatele*: pro nového uživatele systém neumí spolehlivě doporučit objekty, jelikož o něm ještě nemá dostatek informací na vybudování profilu preferencí.

Kolaborativní doporučení („Collaborative filtering“)

Kolaborativní metody odhadují funkci užitečnosti objektu na základě funkcí užitečnosti objektů, které byly hodnoceny tzv. podobnými uživateli (Adomavicius & Tuzhilin, 2005). Tento přístup tedy doporučuje objekty na základě preferencí podobných uživatelů. Sbírá uživatelské hodnocení objektů a následně doporučuje objekty na základě principu nejbližšího souseda mezi uživateli. Za nejbližšího souseda bude v tomto případě považován uživatel/uživatelé, jejichž hodnocení se nejvíce podobají těm našim. Při tomto přístupu je možné analýzu obsahu (objektů) zcela vynechat a zaměřit se pouze na doporučení na základě podobných uživatelů. Toto může být přínosné v případech, kdy je analýza obsahu nemožná nebo nedostatečná (Schafer, Frankowski, Herlocker, & Sen, 2007).

Při získávání preferencí podobných uživatelů můžeme uvažovat dva typy vztahů mezi uživateli, které jsou následující: (Burke, 1999)

- *Symetrické vztahy mezi uživateli* – například při doporučení filmů jsou uživateli doporučené ty, které pozitivně hodnotili ostatní uživatelé s podobným vkusem;
- *Asymetrické vztahy* – například při výukovém systému mají často vyšší váhu doporučení od odborníků nebo lepších studentů.

Možná omezení tohoto přístupu jsou pak následující: (Adomavicius & Tuzhilin, 2005)

- *Problém nového uživatele*: podobně jako při doporučení založeném na obsahu, nemá systém dostatek informací o novém uživateli, aby mohl spolehlivě najít podobné uživatele, a tak doporučovat objekty. Možným řešením je použití hybridních přístupů, ve kterých lze dodatečnou informaci o uživateli získávat například z jejich profilů;

- *Problém nového objektu (například produktu v e-shopu)*: přístupy založené na spolupráci se spoléhají výhradně na hodnocení uživatelů, proto nové objekty, které nebyly posuzovány dostatečným počtem uživatelů, nebudou doporučené;
- *Malý počet hodnocení*: úspěšnost doporučení při kolaborativních doporučení závisí na dostupnosti kritické masy uživatelů, která objekt hodnotila. Z toho vyplývá omezení při doporučení objektů, které hodnotil jen malý počet uživatelů, protože tyto nebudou doporučované nikdy, nebo jen zřídka. Dalším problémem je doporučení uživateli, jehož preference jsou velmi odlišné od ostatních uživatelů. I v tomto případě je možným řešením použití hybridních přístupů na hledání podobných uživatelů například na základě informací z jejich profilů.

Doporučení založené na znalostech („Knowledge-based filtering“)

Doporučení založené na znalostech využívá k formulaci doporučení znalosti o uživateli a znalosti o jednotlivých položkách. Tato metoda bývá užívána v těch oblastech, kde je neúčinné obsahové i kolaborativní doporučení. Jde zejména o nákup produktů, které uživatelé nakupují jednorázově, nebo sporadicky, například nákup drahého zboží, jakým je nábytek do bytu nebo dům. V takovém případě není možné získat uživatelský profil z jeho předchozích hodnocení, jelikož ty buď neexistují, nebo jich je málo a pravděpodobně jsou už staršího data a nemusí se shodovat s aktuálními preferencemi uživatele. Proto musíme využít jiné informace a to požadavky uživatele na kupovaný produkt. Systémy generující doporučení na základě znalostí vytvoří doporučení buď na základě podobností těchto požadavků, tzv. *Case-based systems*, nebo podle explicitních pravidel – *Constraint-based systems* (Blanco-Fernández *et al.*, 2008).

Proces doporučení je následující:

1. Uživatel si zvolí požadavky;
2. Systém vygeneruje doporučení na základě požadavků;
3. Pokud systém nenašel žádný vyhovující obsah, uživatel může změnit požadavky.

Možná omezení tohoto přístupu jsou pak následující:

- Uživatel musí sám zadat požadavky, pokud je zadávání požadavků složitě nastavené, může uživatele odradit. Je tak nevhodnější ponechat většinu požadavků jako dobrovolně uváděných, čímž uživatel sám může modifikovat zadání dle toho,

kolik výsledků získá, je-li jich moc, uživatel přidá další konkretizující požadavky, je-li jich naopak málo nebo žádné, může uživatel požadavky ubrat požadavky či změnit.

Hybridní doporučení („Hybrid recommenders“)

Omezení, která vyplývají z použití výše uvedených metod, které jsou založené na obsahu, znalostech či kolaborativním doporučení, lze minimalizovat použitím tzv. hybridních přístupů. Tyto přístupy v sobě spojují výhody obsahového a kolaborativního doporučení a snaží se minimalizovat jejich nevýhody. Metody založené na obsahu a kolaboraci mohou být kombinovány různými způsoby (Burke, 2007).

Způsoby, kterými lze metody kombinovat, jsou zejména následující (Adomavicius & Tuzhilin, 2005):

- *Implementace metod odděleně a následně kombinování jejich výsledků:* při tomto přístupu můžeme použít výstupy z jednotlivých systémů a následně jako výsledné doporučení použít lineární kombinaci hodnocení. Můžeme však použít jednotlivé systémy doporučení zvlášť a v určitém bodě zvolit jeden, jehož doporučení je lepší na základě vyššího uspokojení, nebo podobnosti s předchozími doporučeními;
- *Zahrnutí obsahově založených charakteristik do kolaborativního přístupu:* (Pazzani, 1999) popisuje způsob použití obsahově založených charakteristik v kolaborativních přístupech jako „spolupráci pomocí obsahu“. Přístup spočívá v použití kolaborativního přístupu k doporučení a zároveň zachování uživatelských profilů. Řeší tím problém nedostatku počtu hodnocení, protože není nezbytné, aby daný objekt ohodnotilo určité množství uživatelů, čímž produkt může být doporučován i na základě vysokých hodnot funkce užitečnosti v návaznosti na uživatelský profil;
- *Zahrnutí kolaborativních charakteristik do přístupu řízeného obsahem:* nejpopulárnější přístup v této kategorii je použití technik redukce dimenzionality skupin uživatelských profilů založených na obsahu, což vede k dosažení vyšší výkonnosti systému než při klasickém přístupu založeném na obsahu;

- *Sestrojení sjednocujícího modelu*: tyto přístupy se snaží vytvořit sjednocující model, který by zahrnul charakteristiky obou modelů a následně je zkombinoval do samostatného systému (Adomavicius & Tuzhilin, 2005).

2.2.2 Známé doporučovací systémy

V současnosti existuje poměrně mnoho systémů pro doporučování. V této práci se zaměříme pouze na některé, které jsou vhodné pro e-shopy. Zvoleny byly dva velmi známé systémy Amazon a Etarget a jeden méně známý Gauss Algorithmic. Systémů však existuje významně více (kupříkladu lze uvést systém Pandora), nicméně není možné se zde věnovat všem existujícím systémům.

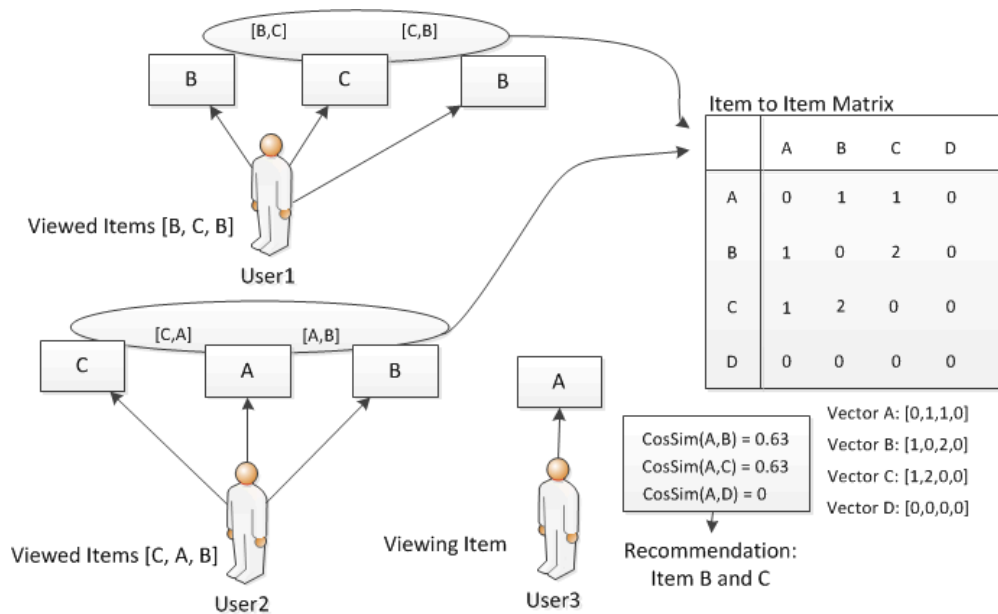
Amazon

Internetový obchod Amazon patří k největším a nejpropracovanějším elektronickým obchodům na celém internetu (v mezinárodním měřítku). Doporučovací systém začal Amazon používat jako jeden z prvních e-shopů na světě a dnes se řadí v kvalitě doporučení ke špičce. To je důvodem, proč je poměrně obtížné získat komplexní informace o doporučovacím systému, v literatuře patrně není dostupné zcela vše, co je relevantní. Mezinárodní internetový obchod Amazon využíval dříve jako doporučovací systém upravenou formu doporučování na základě obsahu, díky čemuž mohl poskytovat milionům zákazníků doporučení v reálném čase, zatímco nejdražší operace určení podobného zboží byly předpočítávány „offline“. Ve fázi předpočítávání se vytvořila tzv. matice podobnosti „Item Similarity matrix“, která popisovala, jak jsou si jednotlivé věci vzájemně podobné (Linden, Smith, & York, 2003). To platí i v současnosti, nicméně současný systém doporučení je hybridní, přičemž kombinuje zejména doporučení na základě obsahu a kolaborativní doporučení (Obrázek 1). Systém dnes při formulaci doporučení zohledňuje celou řadu faktorů, mezi které patří zejména následující (Neumann, 2009):

- 1) Doporučení na bázi toho, co zakoupili jiní spotřebitelé dohromady se sledovaným či do košíku vloženým předmětem;
- 2) Doporučování podobných produktů;
- 3) Doporučování produktů, které systém na základě určitého algoritmu vyhodnotil jako „sousední“ či „doplňkové“ produkty k prohlíženému předmětu;

- 4) Doporučení na základě hodnocení předmětu uživateli (hodnocení 1 až 5 hvězdiček);
- 5) Doporučení na základě podobného profilu uživatele (dle toho, co uživatel hledá, v jaké cenové skupině apod.);
- 6) Vyhledávání stejných či podobných produktů. Zde je nutné vzít v úvahu, že Amazon má významně více uživatelů, než nabízených produktů, což vede k tomu, že je jednodušší vyhledávat shodu či podobu v produktech než v uživatelích;
- 7) Retargeting – uživateli se po velké části internetu opětovně nabízejí produkty z Amazonu, na které se dříve díval či je vyhledal;
- 8) Globální přístup – spíše než sledovat jednotlivé produkty, které uživatel zakoupil, se globální přístup snaží vygenerovat určitá pravidla ze všech předmětů, které uživatel v minulosti zakoupil. Například pokud zákazník v minulosti zakoupil několik knih s romantickými tématy a několik sci-fi knih, globální přístup navrhne doporučit pro příště knihy, které oba tyto prvky kombinují (Neumann, 2009).

Obrázek 1: Doporučovací systém Amazon



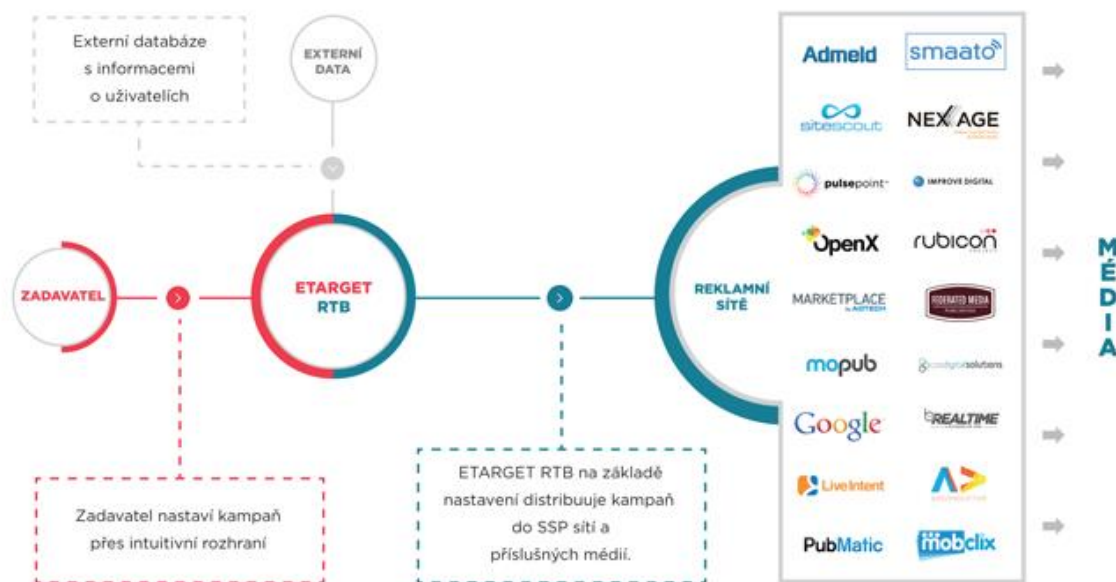
Zdroj: <https://kunuk.wordpress.com/2012/03/04/how-does-the-amazon-recommendation-system-work-analyze-the-algorithm-and-make-a-prototype-that-visualizes-the-algorithm/>

Při generování předpovědi ohodnocení věci uživatelem, jsou pak z matice v reálném čase vybírány věci, které jsou nejpodobnější, a které uživatel již hodnotil. Jelikož počet věcí, které uživatel hodnotil, a jsou zároveň dostatečně podobné jiné věci, je obvykle malý, výpočet ohodnocení je rychlý. Problémem v tomto případě může být velikost matice podobnosti věcí $N \times N$. Při velkém množství produktů dochází k velkým nárokům na dostupnou paměť, proto je obvykle nutné provést optimalizace této matice.

Etarget

Etarget je reklamní (doporučovací) systém, který v současné době působí v devíti zemích Střední a Východní Evropy, kterými jsou následující: Slovensko, Česká republika, Maďarsko, Rakousko, Polsko, Chorvatsko, Srbsko, Rumunsko a Bulharsko. Společnost Etarget byla založena v roce 2002. „Reklamní systém ETARGET RTB zabezpečuje připojení a zobrazování reklam ve více než 60 sítích a statisících webových stránek. V minulosti tyto sítě působily nezávisle, což představovalo ekonomicky neefektivní způsob spravování kampaní“ (ETARGET CZ, 2002). Současný systém používaný pro volbu vhodné reklamy pro uživatele v reklamní síti Etarget je hybridního typu (Obrázek 2).

Obrázek 2: Doporučovací systém ETARGET



Zdroj: <http://www.etaarget.cz/rtbnews/jake-webstranky-jsou-dostupne-diky-etaarget-rtb>

System sestává ze šesti samostatných podsystemů, které jsou následující (ETARGET CZ, 2002):

- 1) Reklama Etarget je cílená zejména behaviorálně, to znamená, že „*reklamní odkazy se uživatelům zobrazují podle toho, co je zajímavá, jaký článek právě čtou, apod.*“ (ETARGET CZ, 2002);
- 2) K dispozici mají e-shopy rovněž geotargeting, což je cílení podle místa, které „*je výhodné hlavně pro inzerenty s lokální působností. Pokud chce e-shop oslovit jen uživatele ve Středočeském kraji, stačí odznačit všechny ostatní regiony*“ (ETARGET CZ, 2002);
- 3) Etarget umožňuje cílit i podle typu zařízení (počítač, smartphone, tablet), na kterém se uživateli reklama zobrazí. Stejně tak si lze přímo ve vlastním kontě vytvořit jednoduchou stránku pro mobilní telefony;
- 4) Nejnovější formou doporučení je Retargeting. Služba Retargeting je tu právě proto, aby se reklamou e-shop připomínal svým uživatelům a potenciálním zákazníkům až do rozhodnutí zde provést nákup či zjistit o nabídce více. Retargeting tak provází potenciální zákazníky téměř na celém českém internetu včetně TOP portálů, jako jsou idnes.cz, blesk.cz a na dalších více jak 1 300 stránkách. Díky této široké síti stránek je tak možné účinně oslovovat více než 80 % Čechů na internetu (ETARGET CZ, 2002);
- 5) System umožňuje i zvolit si čas, ve kterém se má reklama zobrazovat. Tato varianta je výhodná, pokud má zákazník e-shop kontaktovat, lze ji nastavit tak, aby se zobrazovala, když na zákaznické podpoře e-shopu je dostupný jeho pracovník;
- 6) Pokud e-shop má vysledováno, že největší návštěvnost přichází z určitých webů, je vhodné nastavit si tyto weby jako prémiovou stránku, reklama se tak na takových webech zobrazuje více a častěji než jinde, což navyšuje návštěvnost v e-shopu a v konečném důsledku i počet objednávek a zisk (ETARGET CZ, 2002).

Způsob placení za reklamu je pak nastaven tak, že se platí za klik na reklamu, přičemž platby jsou realizovány prostřednictvím aukce (tj., kdo zaplatí nejvíce, zobrazuje se na prvním místě). Maximální výši ceny za klik si nastavuje každý inzerent samostatně (ETARGET CZ, 2002).

Hybridizace těchto podsystémů funguje způsobem „pipeline“. Na počátku se hledá vhodná reklama pro daného uživatele pomocí Retargetingu, v případě že není nalezena žádná vyhovující, pokračuje systém hledáním vhodné reklamy v Behaviorálním podsystému, následně v Kontextuálním (dle zařízení, ze kterého se jedinec připojuje) a pokud ani tam nebyla nalezena vhodná reklama, vybere systém reklamu pomocí podsystému „Weighted selection“, který doporučuje reklamu na základě přiřazených vah, které jsou vypočteny podle určitých kritérií, jako je například cena za klik zákazníka na reklamu.

Gauss Algorithmic

System Gauss Algorithmic je novým českým doporučovacím systémem. Společnost Gauss Algorithmic byla založena v roce 2013 s cílem poskytovat špičkové služby v oblasti analýzy Velkých dat, strojového učení a prediktivní analytiky. Tým společnosti tvoří přední specialisté na vědecké i komerční využití umělé inteligence a zkušení manažeři s dlouholetou praxí v marketingu, finančním řízení i e-commerce (Gauss Algorithmic, 2015).

Jedná se o systém, který při doporučování zboží zákazníkovi e-shopu, vychází z analýzy skutečného chování a potřeb uživatelů, jejich segmentace a personalizace doporučování zboží (Gauss Algorithmic, 2015). Lze jej charakterizovat jako systém hybridního typu, který analyzuje následující aspekty:

- 1) *„Sběr a kultivace dat – historické statistiky návštěvnosti, objednávek, data o zboží a námi doporučené externí zdroje dat;*
- 2) *Analýza historický dat – v datech hledáme a učíme se rozpoznávat vzory chování uživatelů (odsud často používaný výraz behaviorální cílení) s využitím strojového učení. Zároveň analyzujeme zboží, jeho podobnost a ceny;*
- 3) *Nasazení doporučování na web – okamžitě po dokončení analýzy začne umělá inteligence doporučovat zboží na základě typu zákazníka a jeho chování;*
- 4) *Průběžná aktualizace a zlepšování doporučování na základě nových dat – systém se neustále přizpůsobuje změnám a trendům, v případě zapojení externích zdrojů zohledňuje například i počasí nebo nabídku konkurence“* (Gauss Algorithmic, 2015).

2.2.3 Problém studeného startu („Cold-start problem“)

Jedním z problémů kolaborativního i obsahového doporučování je problém studeného startu (tzv. „Cold-start problem“), který je způsoben nedostatečným počtem hodnocení od uživatelů a tudíž není možné dostatečně přesně určit nejbližší okolí uživatele, nebo věci. Obvykle se tento problém vyskytuje zejména v úvodních fázích po spuštění projektu, například internetového obchodu (Pazzani, 1999).

Jedním z řešení tohoto problému, které směřuje k tomu, abychom dokázali zvolit dostatečně velkou skupinu nejbližších sousedů, je brát v úvahu i jiné údaje než například pohlaví, věk, bydliště a další demografické údaje. Jelikož ale nevybíráme podobné uživatele na základě jejich preferencí (resp. ne celá množina sousedů sestává z takových uživatelů) je přesnost doporučení nižší než například u čistě kolaborativní techniky (Pazzani, 1999).

2.3 VÝVOJÁŘSKÉ NÁSTROJE A OSTATNÍ TECHNOLOGIE

2.3.1 Databázový systém Microsoft SQL Server

Databáze existují, protože je zapotřebí měnit data na informace. Data jsou číré a nezpracované skutečnosti. Informace se získávají zpracováním dat na něco užitečného (Gilfillan, 2003). Databáze v dnešní době představují neodmyslitelnou součást nejen průmyslu informačních technologií, ale i průmyslu jako celku. Databáze, ať už chceme nebo ne, dnes využíváme každodenně při bankovních transakcích, cestovních rezervacích, hledání na webových stránkách, hledání zaměstnání nebo při nákupu či jen vyhledávání zboží v e-shopu. Definici pojmu databáze lze najít desítky. Pro účely této práce lze však shrnout, že databáze představuje kolekci vzájemně souvisejících dat, se kterými pracujeme jako s ucelenou jednotkou. Slovem databáze můžeme tedy označit cokoli od jednoduché množiny dat, jakou je například telefonní seznam, až po složité zpracované databáze, jako například databáze produktů v e-shopu. Na práci s takovými daty používáme databázový systém (Price, 2005).

Databázový systém se pak skládá z databáze a systému řízení báze dat (DBMS/SŘBD), který je speciálním software sloužícím pro správu, dodávaný výrobcem databáze. Příkladem databázových systémů jsou třeba: Oracle, MySQL či Microsoft SQL Server.

Pokud se týká databázových systémů, existuje jich celá řada, nicméně v návaznosti na analytickou část práce, zde představíme pouze Microsoft SQL Server. V současnosti je na trhu dostupná verze Microsoft SQL Server 14.

Microsoft SQL Server je relační databázový a analytický systém pro e-obchody, byznys a řešení datových skladů vyvinutý společností Microsoft. Tento systém nabízí celou řadu funkcí, které od moderní databáze požadujeme. „*Počítač, na který budu instalovat MSSQL se stane databázovým serverem. Instalovat na něj budu databázový systém. Těchto databázových systémů může na počítači běžet nezávisle na sobě několik. V případě MSSQL se pak jednotlivé ‚instalace‘ nazývají instance databázového systému. Na jednom databázovém serveru pak může běžet až 16 instancí databázového systému. Tyto instance se navzájem neovlivňují a pracují samostatně. V každé z nich pak může být hostováno libovolný počet databází.*“ (Jecha, 2009).

V současné době je systém Microsoft SQL Server k dispozici hned v několika edicích, jedná se konkrétně o edice: (i) Enterprise pro klíčové firemní aplikace, (ii) Podnik, která zahrnuje „*podniková řešení business intelligence a datové sklady; edice Business Intelligence pro samoobslužná a podniková řešení business intelligence; a edice Standard pro základní databázi a možnosti reportování a provádění analýz*“ (Microsoft, 2015).

Díky své funkci platformy pro správu kriticky důležitých dat, umožňujících dynamický vývoj s rozsáhlou nabídkou funkcí business intelligence, které sahají za rámec běžných relačních dat, tvoří Microsoft SQL Server 2014 základ, na němž mohou malé, střední i velké firmy stavět svou informační infrastrukturu. Ke správě databázových systémů pak slouží nástroj Microsoft SQL Server Management Studio (Stanek, 2013).

Jádro Microsoft SQL Serveru je tvořeno následujícími součástmi (Stanek, 2013):

- *Database Engine Services* – jedná se o součást, která obsahuje jádro celé databáze, notifikační a replikační komponenty. Jádro databáze, což je ve své podstatě vlastní databázový stroj, tvoří základ celého Microsoft SQL Serveru. Replikace je pak odpovědná za distribuci dat do více databází, čímž zvyšuje dostupnost dat a umožňuje rozložení zátěže při čtení na všechny vyhrazené databázové servery;
- *Analysis Services* – jedná se o složku poskytující online analytické zpracování (OLAP) a funkci dolování dat („data-mining“) různým obchodním aplikacím.

Složka analytické služby umožňuje společněm agregovat data z více různých datových zdrojů, například z relačních databází, a umožňují také mnoha způsoby tato data zpracovávat;

- *Integration Services* – jedná se o složku umožňující transformaci a integraci firemních dat díky funkcím pro import a konverzi dat z nejrůznějších datových zdrojů a také umožňují přesouvat data do jednoho či více cílových datových úložišť. „Tyto funkce umožňují slučovat data z nesořaditelných datových zdrojů, načítat data do datových skladů a datových tržišť atd.“ (Stanek, 2013);
- *Reporting Services* – tato složka obsahuje „Report Manager“ a „Report Server.“ Jedná se o komponenty, které vytvářejí „komplexní serverovou platformu pro vytváření, správu a distribuci sestav. Report Server je založen na standardní službě Microsoft Internet Information Services (IIS) a na technologii Microsoft .NET Framework, což umožňuje spojit výhody SQL Serveru a IIS při poskytování a zpracování sestav“ (Stanek, 2013);
- *Service Broker* – jedná se o služby poskytující spolehlivý nástroj pro frontování a zpracování zpráv, který tvoří centrální součásti celé databáze. Fronty je možné použít za účelem postupného zpracování dotazů nebo jiných požadavků a je možno s nimi pracovat na základě dostupnosti veškerých zdrojů. Zprávy pak vytvářejí prostor pro vzájemnou komunikaci různých databázových aplikací. Databázový stroj pracuje s funkcí Service Broker pro doporučování notificačních zpráv;
- *Master Data Services* – jedná se o prostředí pro vytváření obchodních pravidel, která mají následně zajistit kvalitu a přesnost hlavních dat. Obchodní pravidla lze využít ke spuštění obchodních procesů, které řeší problémy s validací dat a řídí toky činností;
- *Data Quality Services* – jedná se o „prostředí pro vytváření úložišť znalostí a metadat, jež napomáhají zlepšovat kvalitu firemních dat. Procesy provádějící údržbu dat mohou upravovat nebo odstraňovat neúplná nebo chybná data. Procesy provádějící porovnávání dat mohou vyhledávat a případně slučovat duplikovaná data“ (Stanek, 2013).

2.3.2 Microsoft Visual Studio

Microsoft Visual Studio je ucelená kolekce nástrojů a služeb, které jsou určeny k vytváření široké škály aplikací nejen pro platformu Microsoft. Tento nástroj sám o sobě nepodporuje žádný programovací jazyk, čímž ponechává na samotném programátorovi, jaké rozšíření ve vztahu k programovacímu jazyku si do nástroje přidá. Aplikace vytvářené prostřednictvím Microsoft Visual Studio mohou být různého typu. Aplikace vytvářené prostřednictvím tohoto nástroje jsou aplikacemi typu klient-server, kdy server je počítač, který obsahuje uloženou databázi, klient je pak počítač, který obsahuje vlastní aplikaci. Microsoft Visual Studio je použitelné mimo jiné na platformě .NET, se kterou bude v této práci dále pracováno. Nástroj také obsahuje celou řadu vlastních designérů pro tvorbu aplikací (například web designer, designer dat, designer mapování, WinForms Designer a další), stejně jako obsahuje celou řadu nástrojů, mezi kterými stojí za zmínku například celá řada průzkumníků (dat, řešení atd.) (Šíma & Vilímek, 2006).

Aplikace, které jsou ve Visual studio vyvíjeny, mohou být určeny pro klientské počítače, web, zařízení či pro cloud. Pokud se týká operačních systémů, lze prostřednictvím Microsoft Visual Studio vytvářet aplikace pro Windows, Android a iOS i moderní webové aplikace a cloudové služby. Aktuálně je k dispozici verze Visual Studio 2015, je k dispozici rovněž verze Visual Studio online, které programátorovi poskytuje sadu cloudových nástrojů pro spolupráci, které fungují s existujícím prostředím IDE nebo editorem (MS Visual Studio, 2015).

2.3.3 ASP technologie

Nástroj ASP.NET je nástrojem, který je optimální k tvorbě webových stránek prostřednictvím HTML5 a CSS3 stejně jako prostřednictvím JAVA Scriptu. Jako u ostatních nástrojů v této práci se jedná o nástroj od společnosti Microsoft. Současný ASP.NET tvoří součást .NET Framework pro tvorbu webových aplikací a služeb. ASP.NET je nástupcem původní technologie ASP („Active Server Pages“). Výhodou tohoto nástroje je možnost vybrat si hned z několika programovacích jazyků. *„Každá stránka v ASP.NET aplikaci obsahuje jednak HTML se serverovými komponentami, tak i kód v pozadí (tzv. code-behind), který ošetřuje události komponent a stará se o funkcionalitu celé stránky. Kromě serverových komponent máme v ASP.NET již*

připravenou funkcionalitu pro většinu nejběžnějších úkonů, které je třeba na webu provádět. Velmi jednoduše do aplikace přidáme správu uživatelů a jejich rolí, profilů a přístupových práv, dále můžeme velmi snadno spravovat strukturu webu pomocí sitemap, můžeme využívat skiny, máme usnadněnou lokalizaci aplikace, veškeré nastavení a konfiguraci aplikace máme na jednom místě, a to v souboru web.config“ (Herceg, 2007).

2.3.4 ASP webhosting

ASP.NET webhosting se v řadě oblastí liší od běžného webhostingu. V současné době je nabízen například prostřednictvím ASPone. Klasický web hosting, který má za cíl pokrýt požadavky aplikace ASP.NET se nabízí celkem v pěti variantách, a to v omezené variantě Free, dále ve variantách Basic, Developer, Enterprise a VIP. Vedle toho je nabízen rovněž specializovaný webhosting zaměřený na Privátní cloud řešení, E-mail hosting, FTP hosting, Microsoft CRM, Exchange hosting a Sharepoint 2013 hosting (ASPone, 2007).

2.3.5 CSS

CSS vzniklo koncem roku 1996. Jedná se o kolekci metod pro grafickou úpravu webových stránek. Zkratka CSS znamená „Cascading Style Sheets,“ v překladu kaskádové styly. Na počátku své existence se jeho grafické využití omezovalo jen na správu písma a barev, v současnosti je však daleko širší. Styly umožňují oddělit strukturu HTML nebo XHTML od vzhledu, čímž získáme přehledný a jednoduchý kód. CSS je možné přesunout do externích souborů, čímž se změní datová velikost a dá se jedním souborem změnit celý styl stránky. CSS také zaručuje stejné grafické vykreslování ve všech prohlížečích (Holzschlag, 2006). V současnosti je aktuální verze CSS3 (Hogan, 2011).

2.3.6 HTML

Zkratka HTML znamená „HyperText Markup Language.“ Tento jazyk byl speciálně vyvinut za účelem tvorby dokumentů na World Wide Web (WWW, Internetu). Jak již z názvu vyplývá, původně byl jazyk HTML vytvořen pro zobrazování textových dokumentů. V relativně krátké době byly jeho původní příkazy doplněny o další, multimediální prvky (grafika, animace, hudba, video a jiné). Takto se jeho hodnota znásobila. Dnes už nikoho, kdo je obeznámen se službami internetu, nepřekvapí, že prakticky ihned je možné získat snímky z meteorologické stanice, obrázky z každé části země nebo poslouchat skladby hudebních skupin všech žánrů. HTML jazyk patří do

skupiny značkových jazyků (markup languages). Na popis WWW stránek využívá značky (příkazy, tagy), které jsou od ostatního textu odděleny lomenými závorkami. Obecně platí, že HTML značky jsou párové, vždy ohraničují úsek textu a ohraničením vyjadřují, kde začíná a kde končí úsek zvýrazněného textu, který bude vyjadřovat odkaz na jinou stránku (Písek, 2006). V současnosti je aktuální nový formát HTML5 (Hogan, 2011).

3 METODIKA

Vzhledem k specifickým rysům zamýšleného e-shopu jsem jako první krok volil druh doporučovacího systému, který by měl obchod využívat. Modul je tvořen pro e-shop se sortimentem rybářských potřeb. V oblasti rybaření nelze předpokládat nijakou větší podobnost uživatelů, jako je tomu například u knih, kdy se pracuje s tím, že uživateli hodnotícímu kupříkladu kladně knihu o sebezpoznání jsou nabídnuty ty knihy, které zakoupil jiný uživatel hodnotící tuto knihu podobně. Dále zde není předpokládán velký počet hodnocení položek, což je dalším problémem kolaborativního doporučení (viz kapitola 2.2.1). Z těchto důvodů, i důvodu větší náročnosti implementace, která by překračovala rámec této práce, jsem se rozhodl tento druh doporučovacího systému nevyužít. Naopak jsou zde využity prvky doporučení založeného na obsahu, kdy jsou uživateli nabízeny položky podobné aktuálně prohlíženému produktu na základě jejich atributů a vzájemné souvislosti (komplementy k aktuálně prohlížené položce). Jedná se tedy o specifické řešení vhodné především pro daný e-shop.

Jako databázový systém byl zvolen Microsoft SQL Server 14 společně s nástrojem správy databáze Microsoft SQL Server Management Studio, jehož licenci mám jakožto student Jihočeské univerzity v Českých Budějovicích zdarma k dispozici díky spolupráci univerzity se společností Microsoft. Dále je zde potřeba upozornit na skutečnost, že veškeré delší texty, jako například popis jednotlivých produktů, jsou demonstrativně vyplněny pseudolatinským textem vygenerovaným pomocí Lorem ipsum.

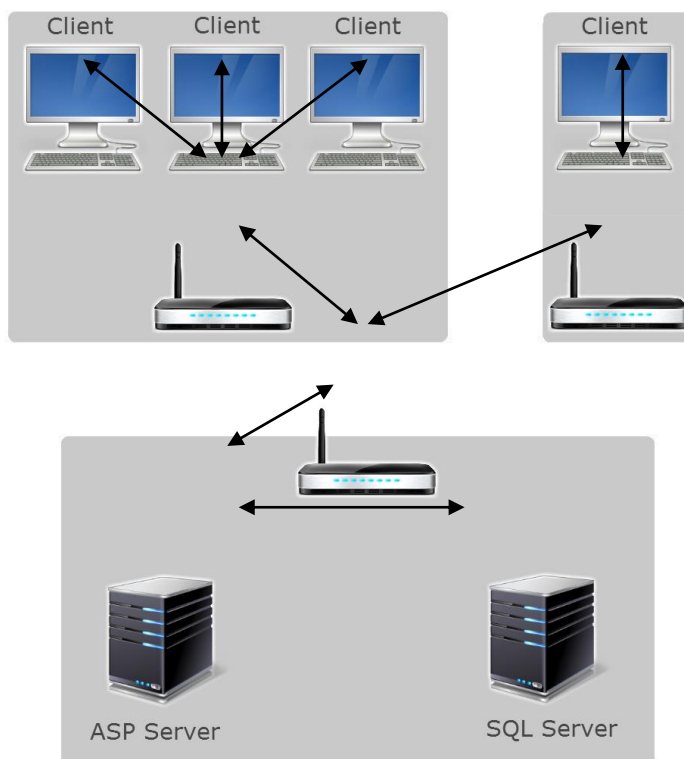
Vývojové prostředí je zde využíváno také od společnosti Microsoft, konkrétně Microsoft Visual Studio Ultimate 2013 s využitím programovacího jazyka C#. Díky vývojovému prostředí od stejné společnosti jako u databázového systému je snadné připojit databázi k prostředí, jelikož obsahuje knihovny pro snadné propojení s Microsoft SQL Serverem. Toto prostředí je velice intuitivní a programovací jazyk C# má velkou škálu příznivců, tudíž i různých tutoriálů a internetových fór s obrovskou komunitou.

4 VÝSLEDKY A JEJICH DISKUSE

4.1 ARCHITEKTURA

Klient odešle skrz webový prohlížeč požadavek na zobrazení stránky. Ten se dostane k ASP serveru za pomoci poskytovatele připojení na obou stranách. ASP server požadavek zpracuje a pošle dotaz na data z SQL serveru, ten dotaz vyhodnotí a získaná data pošle zpět. ASP server přijatá data poskládá do žádoucí podoby a pošle klientovi požadovanou stránku prostřednictvím poskytovatelů internetu (Obrázek 3).

Obrázek 3: Architektura daného systému



Zdroj: Autor

4.2 TABULKY

4.2.1 Produkt

Tabulka Produkt (Obrázek 4) obsahuje datové atributy Id pro uchování unikátního číselného identifikátoru pro jednoznačné určení produktu, Id_sekce obsahující identifikátor sekce z tabulky Sekce, do které je produkt přiřazen, NazevKratky, který se používá při hromadném výpisu produktů, při výpisu názvu detailu produktu se zobrazí hodnota atributu NazevDlouhy, atribut Nahled obsahuje cestu k obrázku produktu, atributy PopisKratky a PopisDlouhy jsou určeny pro popis produktu a posledním atributem je CenaBezDPH, který je určen pro cenu produktu bez daně z přidané hodnoty, aby v případě změny daňové sazby bylo snadné končnou cenu přepočítat.

Obrázek 4: Databázový slovník tabulky Produkt

	Column Name	Data Type	Allow Nulls
PK	Id	int	<input type="checkbox"/>
	Id_sekce	int	<input type="checkbox"/>
	NazevKratky	nvarchar(50)	<input type="checkbox"/>
	NazevDlouhy	nvarchar(200)	<input type="checkbox"/>
	Nahled	nvarchar(100)	<input checked="" type="checkbox"/>
	PopisKratky	nchar(500)	<input checked="" type="checkbox"/>
	PopisDlouhy	nvarchar(4000)	<input checked="" type="checkbox"/>
	CenaBezDPH	float	<input type="checkbox"/>

Zdroj: Autor

4.2.2 Sekce

Jednotlivé sekce jsou uloženy v tabulce Sekce (Obrázek 5). Ta obsahuje datové atributy Id pro jednoznačné určení sekce, Id_nadSekce pro uchování identifikátoru jí nadřazené sekce a atribut Nazev určený pro název sekce. Atribut Id_nadSekce má povoleno obsahovat hodnoty NULL, protože minimálně jedna sekce je na vrcholu hierarchie struktury a žádnou nadřazenou sekci samozřejmě nemá.

Obrázek 5: Databázový slovník tabulky Sekce

	Column Name	Data Type	Allow Nulls
PK	Id	int	<input type="checkbox"/>
	Id_nadSekce	int	<input checked="" type="checkbox"/>
	Nazev	nvarchar(50)	<input type="checkbox"/>

Zdroj: Autor

4.2.3 Hodnocení

Tabulka Hodnoceni (Obrázek 6) slouží pro uchování hodnocení produktů od uživatelů. Protože jeden produkt může obsahovat více hodnocení, jsou ve vztahu 1:N. Atribut Id slouží k jednoznačnému určení hodnocení, Id_produkту určuje pomocí identifikátoru produktu, ke kterému produktu hodnocení patří, Komentar s nepovinnou hodnotou obsahuje komentář k hodnocení, HodnoceniUzivatele udává počet hvězdiček, kterými uživatel produkt ohodnotil, atribut Doporucuje určuje, zda je uživatel s produktem spokojený a doporučuje ho i ostatním uživatelům ke koupi a nakonec atribut DatumHodnoceni obsahuje datum, kdy bylo hodnocení provedeno.

Obrázek 6: Databázový slovník tabulky Hodnoceni

	Column Name	Data Type	Allow Nulls
PK	Id	int	<input type="checkbox"/>
	Id_produkту	int	<input type="checkbox"/>
	Komentar	nvarchar(800)	<input checked="" type="checkbox"/>
	HodnoceniUzivatele	float	<input type="checkbox"/>
	Doporucuje	bit	<input type="checkbox"/>
	DatumHodnoceni	date	<input type="checkbox"/>

Zdroj: Autor

4.2.4 Parametr

K určení parametrů, podle nichž se budou produkty filtrovat, slouží tabulka Parametr (Obrázek 7). Opět obsahuje atribut Id pro jednoznačné určení parametru, Id_ParametrTyp obsahuje identifikátor z tabulky ParametrTyp a určuje, o jaký typ parametru se jedná, a atribut Nazev obsahuje název parametru (například značka, hmotnost či cena).

Obrázek 7: Databázový slovník tabulky Parametr

	Column Name	Data Type	Allow Nulls
PK	Id	int	<input type="checkbox"/>
	Id_ParametrTyp	int	<input type="checkbox"/>
	Nazev	nvarchar(50)	<input type="checkbox"/>

Zdroj: Autor

4.2.5 ParametrTyp

Tato tabulka (Obrázek 8) je pouze pomocná pro tabulku Parametr a obsahuje datový atribut Id, pro jednoznačné určení a Nazev pro název typu parametru. Tabulka obsahuje dva řádky a dá se předpokládat, že jiné hodnoty již obsahovat nebude. První řádek je určen pro výčtový typ a znamená, že uživatel bude vybírat z výčtu hodnot parametru, na základě kterých bude chtít filtrovat produkty. Příkladem je parametr značka, která bude obsahovat jednotlivé názvy výrobců. Druhý řádek je určený pro intervalový typ a je určen pro číselné parametry. Uživatel tedy například zadá, že chce vypsát pouze ty produkty, jejichž cena je v jím nastaveném intervalu.

Obrázek 8: Databázový slovník tabulky ParametrTyp

	Column Name	Data Type	Allow Nulls
PK	Id	int	<input type="checkbox"/>
	Nazev	nvarchar(20)	<input type="checkbox"/>

Zdroj: Autor

4.2.6 ParametrProdukt

Tabulka ParametrProdukt (Obrázek 9) spojuje tabulky Parametr a Produkt ve vztahu M:N a navíc obsahuje hodnotu, která se následně používá pro filtrování produktů. Atribut Id slouží pro jednoznačné určení řádku, Id_produkat obsahuje identifikátor produktu, ke kterému se řádek vztahuje, Id_parametr obsahuje identifikátor parametru, se kterým bude hodnota spárována, a nakonec atributy HodnotaInt a HodnotaString. Tyto dva atributy slouží k uchování hodnoty parametru, ovšem hodnota je vždy pouze v jednom z nich. To, ve kterém atributu je hodnota uchována, záleží na typu parametru. Pokud jde o výčtový typ, hodnota je uložena v atributu HodnotaString, hodnoty intervalových parametrů jsou naopak uloženy v atributu HodnotaInt. Díky rozdělení číselných a nečíselných hodnot dochází ke

snazšímu porovnání a filtrování. A právě proto, že je hodnota parametru obsažena jenom v jednom atributu, musí mít atributy HodnotaString a HodnotaInt povolené neobsahovat žádnou hodnotu.

Obrázek 9: Databázový slovník tabulky ParametrProdukt

	Column Name	Data Type	Allow Nulls
▶	Id	int	<input type="checkbox"/>
	Id_produkut	int	<input type="checkbox"/>
	Id_parametr	int	<input type="checkbox"/>
	HodnotaInt	float	<input checked="" type="checkbox"/>
	HodnotaString	nvarchar(50)	<input checked="" type="checkbox"/>

Zdroj: Autor

4.2.7 SekceParametr

Je zřejmé, že všechny sekce nemohou mít vždy stejné parametry. K propojení sekce a parametru je určena tabulka SekceParametr (Obrázek 10), která obsahuje datový atribut Id pro jednoznačné určení řádku, Id_sekce obsahující identifikátor sekce, Id_parametr obsahující identifikátor parametru a Poradi, určující pořadí výpisu jednotlivých parametrů v dané sekci.

Obrázek 10: Databázový slovník tabulky SekceParametr

	Column Name	Data Type	Allow Nulls
▶	Id	int	<input type="checkbox"/>
	Id_sekce	int	<input type="checkbox"/>
	Id_parametr	int	<input type="checkbox"/>
	Poradi	int	<input checked="" type="checkbox"/>

Zdroj: Autor

4.2.8 Doporučujeme

Tabulka Doporučujeme (Obrázek 11) slouží k doporučení dalších produktů při výpisu detailu produktu. To se děje na základě vzájemně souvisejících sekcí. Je vhodné, aby například při výběru prutu, byly uživateli nabídnuty příbuzné výrobky, jako naviják, vlasec apod.

Sloupec Id slouží pro jednoznačné určení řádku, Id_sekce reprezentuje sekci, ve které je zařazen aktuálně zobrazený produkt a Id_doporucujemeSekce obsahuje identifikátor doporučených sekcí pro danou sekci.

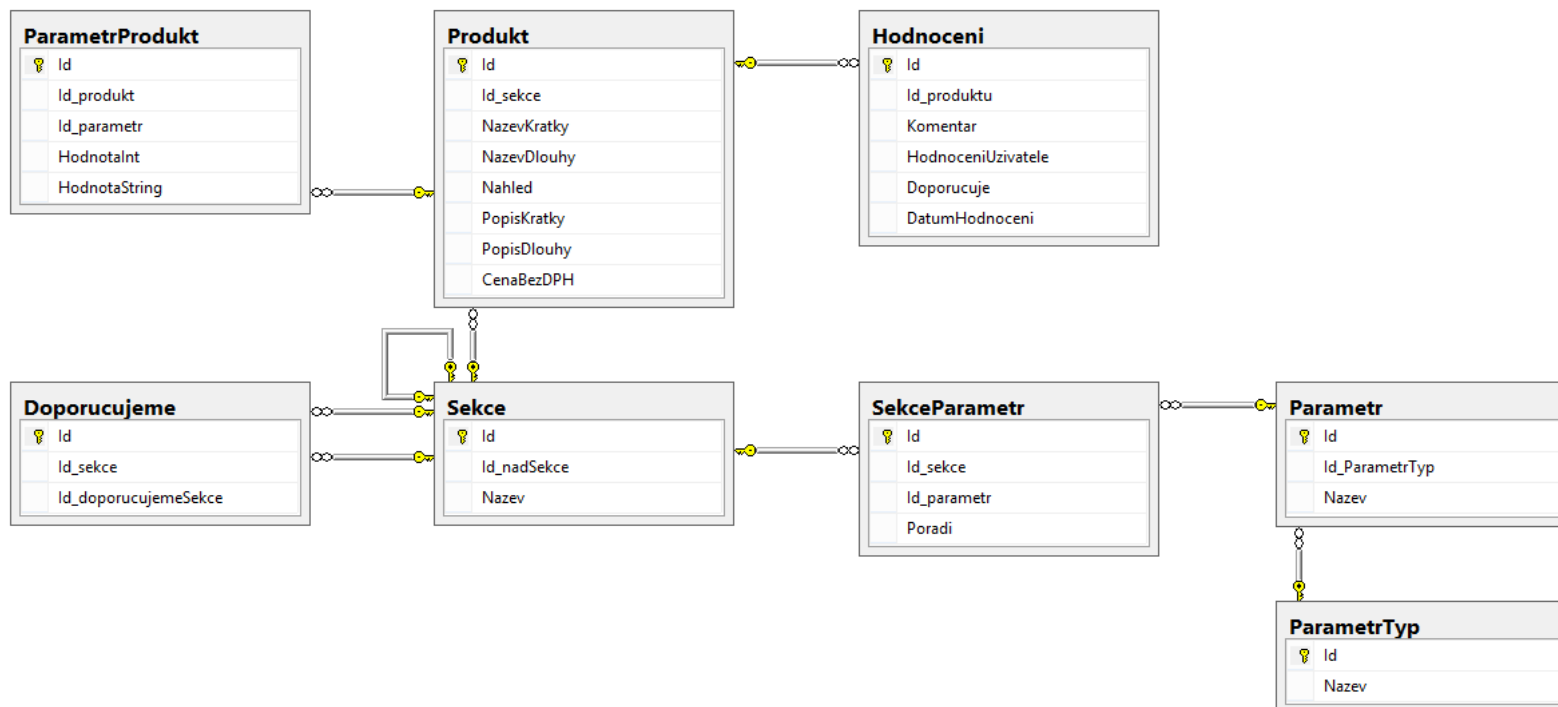
Obrázek 11: Databázový slovník tabulky Doporucujeme

	Column Name	Data Type	Allow Nulls
▶ 🔑	Id	int	<input type="checkbox"/>
	Id_sekce	int	<input type="checkbox"/>
	Id_doporucujemeSekce	int	<input type="checkbox"/>

Zdroj: Autor

4.2.9 Databázové schéma

Obrázek 12: Databázové schéma



Zdroj: Autor

Tabulky jsou vzájemně propojeny na základě primárního a cizího klíče (Obrázek 12). Všechny tabulky jsou provázány ve vztahu 1:N, to znamená, že k jednomu řádku v první tabulce odpovídá více řádků z druhé tabulky. Například k jednomu produktu se vztahuje více hodnocení. Zvláštností je dvojitá kardinalita mezi tabulkami Doporučujeme a Sekce z toho důvodu, že jedna sekce může mít doporučených více sekcí a zároveň tato sekce může být doporučena pro více jiných sekcí. Další netypický vztah je mezi tabulkou Sekce a jí samotnou, protože sekce jsou hierarchicky uspořádány v jedné tabulce.

4.3 FUNKCE

4.3.1 HodnoceniProduktu

Funkce `HodnoceniProduktu` vypočítává průměrné hodnocení produktu na základě udělených počtů hvězdiček uživateli. Jako parametr je předána hodnota `Id` produktu. Dále jsou využity agregační funkce `SUM()` a `COUNT()` k jednoduché matematické operaci. Nejprve se provede celkový součet hodnocení, který je poté dělen jejich počtem. Tato data se samozřejmě vztahují jen k odpovídajícímu produktu. Výsledek je uchován z důvodu přesnosti v datovém typu `float`, který je pro výpis ovšem zaokrouhlen na dvě desetinná čísla.

4.3.2 CenaProduktu

Pro získání ceny produktu se využívá `CenaProduktu`, která přebírá parametr `Id` produktu. Tato hodnota je uložena v tabulce `ParametrProddukt` ve sloupci `HodnotaInt` v řádku, kde hodnota sloupce `Id_parametr` nabývá hodnoty 6 (id parametru `cena`) a zároveň hodnota sloupce `Id_produk` je rovna parametru funkce.

Tato struktura je zvolena tak, aby hodnoty, podle kterých se produkty mohou filtrovat, byly všechny ve stejné tabulce a data byla konzistentní. Algoritmus pro filtrování tedy funguje pro všechny parametry stejně a není nutné dělat výjimky. Další výhodou se projeví v situaci, kdy dojde ke změně sazby daně z přidané hodnoty. Tehdy je možné přepočítat ceny s DPH na základě sloupce `CenaBezDPH` v tabulce `Produkt` jednoduchou změnou číselného koeficientu (viz Ukázka kódu 1).

Ukázka kódu 1: Úprava ceny produktů při změně sazby DPH

```
UPDATE ParametrProdukt SET HodnotaInt = ROUND((SELECT CenaBezDPH FROM Produkt WHERE Id = Id_produk) * 1.21, 0) WHERE Id_parametr = 6
```

Zdroj: Autor

Všechny kontrolery používají proměnnou datového typu `EshopEntities1`. Ta slouží k přístupu aplikace do databáze. Tato proměnná obsahuje třídy, které odpovídají svým názvem a strukturou jednotlivým tabulkám.

4.4 KONTROLERY (CONTROLLERS)

4.4.1 ProduktsController

Základní funkcí kontroleru ProduktsController je získávání a zpracování informací z databáze a předání je příslušným pohledům.

Stěžejní metodou je metoda VsechnyProdukty s parametry datového typu integer a dvěma poli datového typu string. První parametr reprezentuje číselný identifikátor aktuální sekce. Z důvodu výpisu produktů bez omezení sekce, je povoleno, aby hodnota parametru nabývala hodnoty null. Druhým parametrem je pole datového typu string představující podmínky výčtových filtrů, které byly zadány uživatelem. Podmínky intervalových filtrů obsahuje třetí parametr metody, který je opět pole datového typu string. Hlavní funkcí metody je vygenerování SQL dotazu, podle kterého proběhne vyfiltrování všech produktů, které odpovídají zadaným kritériím a zároveň se nacházejí v požadované sekci (Ukázka kódu 2). Generování dotazu začíná tím, že se do proměnné SELECT přiřadí názvy požadovaných sloupců z tabulky. V případě, že identifikátor sekce není roven hodnotě null, k proměnné SELECT se připojí podmínka na výběr pouze těch produktů, které patří do požadované sekce. Následně proběhne vytvoření podmínek, které odpovídají hodnotám v předaných parametrech. Vzhledem tomu, že hodnoty parametrů jednotlivých produktů jsou uloženy v řádcích, nikoli ve sloupcích, není možné produkty vybírat pomocí jednoduché klauzule WHERE, ale je nutné použít klauzuli INTERSECT, čímž dojde k průniku výsledků z jednotlivých dotazů (Obrázek 13). Na základě tohoto dotazu dojde k výběru požadovaných produktů.

Obrázek 13: Grafické znázornění klauzule INTERSECT



Zdroj: Autor

Ukázka kódu 2: Generování podmínek pro SQL dotaz, který provádí výběr produktů splňující kritéria filtrů

```
//vygenerovani podmínek vycetovych selectoru
string[] splity;
int idParametrStary = 0, idParametrNovy;
for (int i = 0; i < checkboxPodminky.Length; i++)
{
    splity = checkboxPodminky[i].Split(new char[] { ',' });
    idParametrNovy = Convert.ToInt32(splity[0]);

    //nastaveni stareho id, aby program pokračoval nasledující podmínkou
    if (i == 0)
        idParametrStary = idParametrNovy;

    if (idParametrStary == idParametrNovy)
    {
        //stejný parametr jako předtím s novou hodnotou
        if (podminka != string.Empty)
            podminka += " OR ";
        podminka += "(Id_parametr = "+splity[0]+" AND HodnotaString = '"
        +splity[1]+"')";
    }
    else
    {
        //jiný parametr - ukončení stareho dotazu a vytvoření noveho
        if (subdotaz != string.Empty)
            subdotaz += " INTERSECT " + subdotazSelect + podminka;
        else
            subdotaz = subdotazSelect + podminka;

        podminka = "(Id_parametr = "+splity[0]+" AND HodnotaString = '"
        +splity[1]+"')";
        idParametrStary = idParametrNovy;
    }
}

if (numberPodminky != null && numberPodminky.Length > 0)
{
    string[] splity1, splity2;
    for (int i = 0; i < numberPodminky.Length - 1; i += 2)
    {
        splity1 = numberPodminky[i].Split(new char[] { ',' });
        splity2 = numberPodminky[i + 1].Split(new char[] { ',' });

        podminka = subdotazSelect + "Id_parametr = " + splity1[0] + " AND
        HodnotaInt >= " + splity1[2] + " AND HodnotaInt <= " + splity2[2];
        if (subdotaz != string.Empty)
            subdotaz += " INTERSECT ";
        subdotaz += podminka;
    }
}
}
```

Zdroj: Autor

Jednou z dalších metod je `VypisProduktuJavaScript` s parametrem datového typu `integer`, přičemž je umožněna i hodnota `null`. V případě, že hodnota parametru neodpovídá hodnotě `null`, metoda načte filtry, které jsou spárovány s aktuální sekcí. Z toho vyplývá, že při základním výpisu produktů, není možné produkty filtrovat podle žádných kritérií.

Nezávisle na hodnotě předaného parametru metoda vždy získá z databáze seznam podsekcí jedné podúrovně. Samotné získání informací o produktech tato metoda nezajišťuje, protože toto provádí metoda `VsechnyProdukty` ve spolupráci s `JavaScriptem`.

Metoda `VypisProduktu` se od té předchozí liší v tom, že nezpracovává informace o aktuální sekci, podsekcích či filtrech. Jejím jediným cílem je výběr produktů z databáze. Pokud je uveden číselný identifikátor sekce, budou vybrány pouze ty produkty, které do této sekce spadají.

Problém studeného startu je zde řešen pomocí metody `ProduktyNejlepsi`. Ta nepřebírá žádné parametry. Používá se při výpisu produktů na hlavní stránce, protože v této situaci není možné zjistit uživateli preference. Proto mu jsou nabídnuty produkty, které mají nejlepší hodnocení v jednotlivých sekcích. Aby byl zajištěn stále stejný vzhled stránky, vybírá se vždy pouze prvních dvanáct produktů z takto sestaveného pořadí.

Pro výpis detailu produktu je použita metoda `Produkt`, které je předán parametr datového typu `integer`. Ten reprezentuje číselný identifikátor produktu v tabulce `Produkt`. Pro ošetření chyby, kdy uživatel umaže identifikátor produktu z URL adresy, je možné předat parametr hodnoty `null`. V tomto případě je zobrazen štítek s nápisem „Produkt nebyl nalezen“. V případě platné hodnoty identifikátoru dojde k získání detailních informací a hodnocení o patřičném produktu. Zároveň jsou získány produkty, které jsou doporučeny ke koupi s tímto produktem. Doporučení je vypočítáno pomocí tabulky `Doporucujeme`, kde jsou k dané sekci přiřazeny sekce příbuzné. Z těchto sekcí jsou vždy vybrány celkem čtyři nejlépe hodnocené produkty.

V tomto kontroleru jsou také metody, které byly vygenerovány `Visual Studiem` a jsou určeny pro správu tabulky `Produkt`. Mezi ně patří metoda `Index`, sloužící pro výpis všech produktů, metoda `Details`, která získává detailní informace o položce. K vytvoření nového produktu je využita metoda `Create`, úpravu produktu zajišťuje metoda `Edit`, smazání položky metoda `Delete` a kontroler obsahuje ještě další dvě pomocné metody `DeleteConfirmed` a `Dispose`.

4.4.2 SekcesController

Účelem tohoto kontroleru je získávání dat o sekcích. Stěžejní metodou je metoda `Sekce`, která jako parametry přebírá číselný identifikátor sekce, který může zároveň nabývat

hodnoty null pro případ, že sekce není specifikována, a dva řetězce s názvy metody a kontroleru. Metoda získá informace o aktuální sekci z databáze, jejích podsekcí a to vše vrátí zpět. Návrátová hodnota je datového typu SekceVypis.

Dále kontroler obsahuje vygenerované metody, které zajišťují správu tabulky Sekce. Jedná se o metody Create, Delete, Details, Edit, Index, DeleteConfirmed a Dispose.

4.4.3 SelectorController

Hlavní metodou tohoto kontroleru je metoda Selectory, která přebírá jako parametr číselný identifikátor reprezentující sekci. Z databáze se získají informace o filtrech, které jsou pro danou sekci specifikovány. Poté je ještě nutné zjistit hodnoty pro jednotlivé filtry – pro výčtový filtr se zjišťují jednotlivé položky, podle kterých může uživatel výpis produktů filtrovat (například výrobce nebo příchut') a pro intervalový filtr se zjišťují minimální a maximální hodnoty, které jsou v dané sekci. Tuto metodu využívá i metoda VypisSelectoru, která pouze vrací pohled se získanými daty od metody Selectory.

4.4.4 HomeController

Kontroler obsahuje pouze jednu metodu – Index. Ta využívá ostatní kontrolery a pomocí metod jejich instancí plní model, který bude následně předán pohledu. Z metody Sekce instance kontroleru SekcesController získá informace o sekcích, které jsou umístěny hierarchicky nejvýše, a z metody ProduktyNejlepsi instance ProduktsController načte data o dvanácti nejlépe hodnocených produktech v jednotlivých kategoriích.

4.5 MODEL (MODELS)

Modely jsou nosiči dat, která jsou naplněna kontrolery a jsou předána pohledům pro jejich zobrazení.

4.5.1 Vygenerované třídy

Vývojové prostředí Microsoft Visual Studio umožňuje vygenerování tříd na základě databázového schématu. Automaticky se tedy vytvoří třídy, které mají shodný název a stejné datové položky, jako obsahuje tabulka. Tímto vzniklo sedm tříd – Hodnoceni, Parametr, ParametrTyp, ParametrProdukt, Produkt, Sekce, SekceParametr.

4.5.2 ProduktModel

Tento model slouží k uchování dat o jednotlivých produktech a jejich výpisu. Z toho důvodu, že je potřeba uchovávat více informací o produktech, než je uvedeno v tabulce Produkt, není možné k tomu to účelu použít model Produkt. Protože jde v podstatě o rozšíření již existujícího modelu, dědí model ProduktModel všechny datové složky a metody z modelu Produkt. Navíc obsahuje vlastnosti NahledObrazek (datového typu string), Cena (datového typu double), HodnoceniPrumer (datového typu double), HodnoceniPrumerWidth (datového typu integer) a Doporucujeme (list datového typu ProduktModel). Náhled obrázku existuje i v modelu Produkt, ale tento model pomocí vlastnosti NahledObrazek počítá se situací, kdy náhled neexistuje a v tom případě vrací cestu k obrázku obsahující přeškrtnutý fotoaparát. Vlastnost HodnoceniPrumerWidth slouží k výpočtu šířky obrázku se žlutou barvou, který se později používá při grafickém znázornění hodnocení produktu (více popsáno v kapitole 4.6.1), na základě hodnoty vlastnosti HodnoceniPrumer. Model ProduktModel také obsahuje doporučené produkty k přikoupení, tento seznam reprezentuje vlastnost Doporucujeme.

4.5.3 SekceVypis

Pro uchovávání informací o aktuální sekci a popřípadě jejích podsekcích je určen model SekceVypis, který později používá pohled VypisSekci (kapitola 4.6.2). Vlastnost AktualniSekce (datového typu Sekce) obsahuje informace o aktuální sekci, případný seznam podsekcí reprezentuje vlastnost PodSekce (list datového typu Sekce) a pro vytvoření správného odkazu na jednotlivé sekce slouží vlastnosti ActionName a ControllerName (oba datového typu string).

4.5.4 SelectoryHodnotyInterval

Tato jednoduchá třída byla vytvořena pro jednodušší práci s modelem Selector (kapitola 4.5.6). Obsahuje celkem tři datové položky – Id_parametr (datového typu integer) pro pozdější generování podmínek SQL dotazu, Minimum a Maximum (oba datového typu double) reprezentující hranice intervalu, ve kterém mohou filtrační podmínky pohybovat.

4.5.5 SelectoryHodotyVycet

Třída SelectoryHodotyVycet reprezentuje jednu položku ve výčtovém filtru – například konkrétního výrobce produktů. Vlastnost Id_parametr (datového typu string) slouží pro

pozdější generování SQL dotazu a vlastnost `HodnotaString` (opět datového typu `string`) nese název položky.

4.5.6 Selector

Nosičem dat jednoho filtru je model `Selector`. Ten obsahuje celkem sedm vlastností a jednu přetíženou metodu. Vlastnost `ID` (datového typu `integer`) je určena pro identifikátor filtru, název je uchován ve vlastnosti `Nazev` (datového typu `string`), `Minimum` a `Maximum` (oba datového typu `double`) reprezentují hranice intervalu intervalového filtru, seznam položek výčtového filtru je uložen ve vlastnosti `Vycet` (list datového typu `SelectorHodnotyVycet`) a ve vlastnostech `Id_parametrTyp` a `Id_parametr` jsou uchovány identifikátory parametru a typu parametru. V případě, že je metodě `PridatHodnoty` předán parametr list datového typu `SelectorHodnotyVycet`, provede se přiřazení hodnoty parametru do vlastnosti `Vycet`. Pokud je předán parametr list datového typu `SelectorHodnotyInterval`, dojde k přiřazení hodnot minima a maxima z první položky v listu do vlastností `Minimum` a `Maximum`. Tento postup se zdá být poměrně neefektivní, ale díky automatizaci SQL dotazů a přebírání jejich hodnot jde o nejjednodušší řešení.

4.5.7 ProduktyVypis

Model `ProduktyVypis` se používá při hromadném výpisu produktů v pohledu `VypisProduktu`. Obsahuje tedy vlastnosti `Produkty` (list datového typu `ProduktModel`) pro jednotlivé produkty, `Sekce` (datového typu `SekceVypis`) pro uchování informací o aktuální sekci a `Selectory` (list datového typu `Selector`) s daty o filtrech, podle kterých je možné výpis produktů filtrovat.

4.6 POHLEDY (VIEWS)









Pohledy zobrazují data získaná z kontrolorů prostřednictvím modelů. Jednotlivé pohledy je možné do sebe vkládat prostřednictvím metody `Partial`.

4.6.1 VypisProduktu

Pomocí tohoto pohledu je zobrazen stručný přehled o produktech. Zobrazena jsou data získaná z modelu datového typu `ProduktModel`. U každé položky je zobrazen krátký název, část krátkého popisu, náhledový obrázek, cena a celkové hodnocení uživateli. Grafické zobrazení hodnocení (Obrázek 14) probíhá na základě překrytí dvou obrázků. Podkladový

obrázek sestává ze žluté barvy a má velikost 1px × 1px. Podle průměru hodnocení uživatelů, které bylo více popsáno v kapitole 4.3.1, se nastaví šířka obrázku. Ten je poté překryt obrázkem s transparentními tvary hvězd a bílým pozadím. Toto zobrazení je umožněno díky rozdílné hodnotě CSS vlastnosti z-index a absolutní pozici obou obrázků. Každý z produktů je zároveň odkazem na detail položky.

Obrázek 14: Ukázka pohledu VypisProduktu

<p>The Complete mix Voorn 2kg</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 146,-</p>	<p>Sonubaits Krill Drilled Boilie</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 196,-</p>	<p>Shakespeare Cypri 1202 Karperhenge</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 1307,-</p>	<p>Shimano Aernos Winkle Picker</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 2679,-</p>
<p>Spro Offshore Pro Reel</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 2091,-</p>	<p>JVS Extreme Nylon Vislijn</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 146,-</p>	<p>Shakespeare Sigma Fly Reel</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 1389,-</p>	<p>Daiwa Yank N Bank Margin Plavačková dělička</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 9801,-</p>

Zdroj: Autor

4.6.2 VypisSekci

Pro zobrazení výpisu sekci slouží pohled VypisSekci (Obrázek 15). Pohledu je předán model datového typu SekceVypis. Pokud je proměnná Model.AktualniSekce.Id rovna hodnotě null, tak to znamená, že je požadavek na zobrazení sekci na nejvyšší hierarchické úrovni. Tomu se tak děje především na úvodní stránce. V případě, že hodnota proměnné Model.AktualniSekce.Id není rovna hodnotě null, jde o výpis příslušných podsekci (Obrázek 16). K tomuto výpisu je zároveň připojen odkaz pro možnost navrácení se o jeden hierarchický stupeň výše. Každý název je zabalen do odkazu přejítí na požadovanou sekci.

Obrázek 15: Ukázka pohledu VypisSekci 1



Zdroj: Autor

Obrázek 16: Ukázka pohledu VypisSekci 2

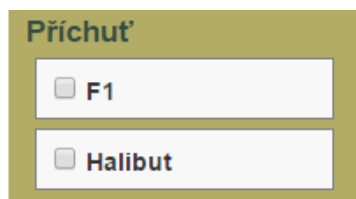


Zdroj: Autor

4.6.3 VypisSelectoru

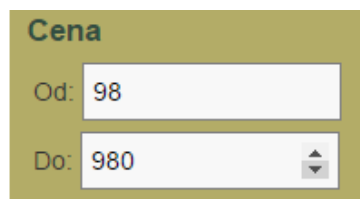
Pohled VypisSelectoru je určen k zobrazení filtrů, pomocí nichž je možné výpis produktů filtrovat. Model je list datového typu Selector. Počet, druh a typ filtrů je závislý na tabulce SekceParametr. Pohled vytvoří HTML formulář, postupně projde list selectorů a každý jednotlivě vypíše. U výčtového filtru je pro každou položku vytvořen HTML element input typu checkbox (Obrázek 17). Aby nebylo nutné kliknout přímo na checkbox, ale zaškrtování fungovalo i při kliknutí na název příležitosti k checkboxu, je samotný checkbox i název zabalen do HTML elementu label. V případě intervalového filtru (Obrázek 18) jsou vygenerovány dva HTML elementy input typu number. Při nastavování desetinných hodnot těchto elementů je nutné, aby byla hodnota zadána s desetinnou tečkou, a zároveň musí být nastaven parametr Step na hodnotu „any“, aby bylo umožněno uživateli měnit hodnotu pomocí kliknutí myši na tlačítka nahoru a dolů.

Obrázek 17: Ukázka pohledu VypisSelectoru – Intervalový filtr



Zdroj: Autor

Obrázek 18: Ukázka pohledu VypisSelectoru – Výčtový filtr



Zdroj: Autor

4.6.4 Produkt

Dalším pohledem, určeným pro zobrazení detailních informací o jednotlivých položkách databáze je pohled Produkt (Obrázek 19). Ten přebírá model datového typu ProduktModel. Jestliže tento model nabývá hodnoty null, je vypsáno upozornění „Produkt nebyl nalezen“. V opačném případě je horní část stránky rozdělena na dva sloupce. V levém sloupci je zobrazena fotografie produktu, v pravém jeho základní údaje. Ty se skládají z dlouhého názvu, krátkého popisu, ceny s HTML elementem input typu button, který je připraven pro možnost přidání položky do nákupního košíku a dále hodnocení produktu. Hodnocení je zde zobrazeno identicky, jako v pohledu VypisProduktu (více popsáno v kapitole 4.6.1). Zbývající část stránky již není rozdělena na sloupce a prvním prvkem je zde dlouhý popis produktu. Dále je zobrazeno již konkrétní hodnocení uživatelů s procentuálním vyjádřením míry jejich spokojenosti a případný komentář. Procentuální hodnota je vypočítána na základě počtu udělených hvězdiček a jeho násobku dvaceti, protože uživatel může vybírat z celkem pěti hvězdiček (jedna hvězdička tedy představuje dvacet procent). V případě, že uživatel položku doporučuje, je nastavena hodnocení třída Doporučuje, která obarvuje text nadpisu na zelenou barvu. V opačném případě je nastavena třída Nedoporučuje, která přiřadí parametru color hodnotu red. Pokud nebyla dosud položka hodnocena, vypíše se informace „Produkt nebyl hodnocen“.

Poslední část pohledu zobrazuje čtyři produkty, které jsou doporučeny k zakoupení společně s tímto produktem. Doporučení je generováno za pomoci tabulky Doporučujeme (viz kapitola 4.2.8). Využívá funkce HTML.Partial, které jsou předány parametry s názvem pohledu a model, který je tomuto pohledu předán. Díky této metodě je do pohledu Produkt vložen pohled VypisProduktu.

Obrázek 19: Ukázka pohledu Produkt



SPICY MEATY METHOD MIX 2KG

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras facilisis nibh lacus, sit amet ultricies lorem commodo et. Curabitur volutpat.

★★★★☆

327,-

Popis produktu

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum sagittis leo nec nibh tincidunt vestibulum. Morbi vitae efficitur eros. Donec dolor justo, maximus sed augue eget, pulvinar ultricies erat. Vivamus sit amet semper metus. Duis iaculis dolor sed libero sagittis.

Hodnocení produktu uživateli

68% - **Doporučuje**

Komentář: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam pulvinar tortor id urna laoreet vulputate. Aenean malesuada elit in semper molestie. Cras venenatis ornare magna eu eleifend. Suspendisse condiment

32% - **Nedoporučuje**

Komentář: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam pulvinar tortor id urna laoreet vulputate. Aenean malesuada elit in semper molestie. Cras venenatis ornare magna eu eleifend. Suspendisse condi

Doporučujeme přikoupit

<p>Sonubaits Krill Drilled Boilie</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 196,-</p>	<p>Sonubaits Mixed Method Boilie</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 131,-</p>	<p>The Complete mix Voorn 2kg</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 146,-</p>	<p>JVS Voer Roach 1kg</p> <p>Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 98,-</p>
---	--	--	--

Zdroj: Autor

4.6.5 VypisProduktuJavaScript

Pohled VypisProduktuJavaScript zobrazuje výpis podsekcí a selectoru pro danou sekci a jí přiřazené produkty. K tomu využívá vložení pohledu VypisSekci a zároveň VypisSelectoru. Tyto pohledy jsou umístěny v levé části stránky pod sebou, vpravo je prostor pro výpis produktů. Na rozdíl od pohledu VypisProduktu, je použita technologie JavaScript (Ukázka kódu 3). Ta přiřazuje každému inputu z pohledu Selector událost po kliknutí a při jejím vyvolání se zavolá metoda VsechnyProdukty z kontroleru

ProduktsController (viz kapitola 4.4.1) a předají se jí příslušné parametry. Tato metoda vrátí vyfiltrované produkty podle přání uživatele a provede jejich výpis. Po vzhledové stránce je výpis totožný s výpisem produktu z pohledu VypisProduktu (Obrázek 14). Technologie JavaScript byla použita z důvodu, že umožňuje dynamické chování stránky bez nutnosti jejího refreshu.

Ukázka kódu 3: Načítání produktů pomocí technologie JavaScript

```
//vytvoreni pole zaskrtnutych checkboxu
var podminkyCheckbox = Array();
var inputCheckbox = $('input[type=checkbox]').each(function(i, value) {
    if ($(this).prop('checked')) {
        podminkyCheckbox.push(Array($(this).attr('name'), $(this).attr('alt'),
$(this).prop('checked')));
    }
});

//vytvoreni pole ciselných inputu
var podminkyNumber = Array();
$('input[type=number]').each( function(i, value) {
    podminkyNumber.push(Array($(this).attr('name'), $(this).attr('alt'),
$(this).prop('value')));
});


//ziskani dat
jQuery.ajaxSettings.traditional = true;
$.ajax({
    url: "@Url.Action("VsechnyProdukty", "Produkts")",
    data: { 'id': id, 'checkboxPodminky': podminkyCheckbox, 'numberPodminky':
podminkyNumber },
    type: "GET",
    dataType: "json",
    success: function (data) {
        vypisDat(data);
    },
    error: function () {
        alert("Při načítání se vyskytla chyba. Zkuste to znovu.");
    }
});
```

Zdroj: Autor













4.6.6 Index

Do pohledu je nejprve vložen úvodní tematický obrázek, který je umístěn ve složce Content/Images. Dále jsou zde vypsány produkty pomocí pohledu VypisProduktu, který přebírá jako parametr list datové typu ProduktModel. Tyto produkty představují řešení problému studeného startu (více v kapitole 4.4.1). Vedle pohledu VypisProduktu je v levém sloupci zobrazen pohled VypisSekci (viz kapitola 4.6.2). Názorné zobrazení rozdělení jednotlivých pohledů je k vidění na obrázku níže (Obrázek 20).

Obrázek 20: Ukázka pohledu Index



Pavel, ten má na háku...

<p>Nástrahy</p> <hr/> <p>Pruty</p> <hr/> <p>Navijáky</p> <hr/> <p>Vlasce</p>	<p>The Complete mix Voorn 2kg Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 146,-</p>	<p>Sonubaits Krill Drilled Boilie Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 196,-</p>	<p>Shakespeare Cypry 1202 Karperhenge Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 1307,-</p>	<p>Shimano Aernos Winkle Picker Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 2679,-</p>
<p>Spro Offshore Pro Reel Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 2091,-</p>	<p>JVS Extreme Nylon Vislijn Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 146,-</p>	<p>Shakespeare Sigma Fly Reel Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 1389,-</p>	<p>Daiwa Yank N Bank Margin Plavačková dělička Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 9801,-</p>	
<p>Fox Rage Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 425,-</p>	<p>Fox FX Rod Tip Protector Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 327,-</p>	<p>Mitchell Avocast 8000 Braid Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 3267,-</p>	<p>X2 Halibut Pellet 20mm 1kg Lorem ipsum dolor sit amet, consectetur</p>  <p>★★★★☆ 163,-</p>	

VypisSekci

VypisProduktu

Zdroj: Autor

4.6.7 Vygenerované pohledy

Vývojové prostředí Microsoft Visual Studio umožňuje vygenerovat pohledy, které slouží pro správu dat databáze. Při generování pohledu je nutné nastavit třídu, pro kterou mají být pohledy určeny. Mechanismus vytvoří celkem pět pohledů (Obrázek 21–Obrázek 25). Index pro hromadný výpis položek, Details pro podrobný výpis položky, Edit pro její úpravu, Create pro vytvoření nové položky a Delete pro smazání stávajících.

Obrázek 21: Ukázka pohledu Delete

Are you sure you want to delete this?

Sekce

Id_nadSekce 2
Nazev Boilies

Delete | Back to List

Zdroj: Autor

Obrázek 23: Ukázka pohledu Create

Sekce

Id_nadSekce
Nazev
Create

Zdroj: Autor

Obrázek 22: Ukázka pohledu Details

Sekce

Id_nadSekce 2
Nazev Boilies

Edit | Back to List

Zdroj: Autor

Obrázek 24: Ukázka pohledu Edit

Sekce

Id_nadSekce
Nazev
Save

Zdroj: Autor

Obrázek 25: Ukázka pohledu Index

Create New

Id_nadSekce	Nazev	
2	Pelety	Edit Details Delete
3	Feederové pruty	Edit Details Delete

Zdroj: Autor

Mimo nástroje správy je vygenerován pohled `_Layout`, který obsahuje základní strukturu HTML stránky a do něj je vkládáno tělo z jejích pohledů. Zde je specifikován název stránky, CSS styly a popřípadě další skripty.

5 ZÁVĚR

Tato bakalářská práce měla za úkol podat přehled metod implementace doporučovacích systémů a představit jejich nejvýznamnější představitele. Dále byla rozebrána problematika uživatelských preferencí. V neposlední řadě bylo cílem navrhnout a implementovat modul doporučení pro daný e-shop, který bude na základě informací poskytnutých uživatelem vybírat a nabízet potenciální položky jeho zájmu, což může přinést užitek jak jemu, tak majiteli obchodu.

Navrhovaný modul obsahuje více způsobů doporučení. Prvním z nich je výpis produktů na hlavní stránce, kdy v tomto případě ještě nelze určit předmět uživatelského zájmu. Nabídnuty mu jsou produkty, které mají nejlepší hodnocení v jednotlivých sekcích, a pro zachování vždy stejného vzhledu stránky je vybráno pouze prvních dvanáct produktů z takto sestaveného pořadí. Hlavní doporučení zde ovšem probíhá při detailním zobrazení konkrétního produktu, kdy jsou uživateli nabídnuty čtyři nejlépe hodnocené produkty ze sekcí příbuzných té, ve které je aktuálně prohlížena položka zařazena. Příbuzné sekce jsou vybírány na základě komplementů, tzn., že se vzájemně doplňují (například k navijákům jsou doporučeny vlasce apod.).

Nasazení modulu u jiných internetových obchodů je ovšem limitováno rozlišnými specifiky, například v případě rozšíření sortimentu nabízeného zboží by byla vhodná implementace korelační analýzy objektů pro automatické přiřazení „sousedních“ či „doplňkových“ produktů k prohlíženému předmětu. V případě omezeného množství nabízeného zboží, jako je tomu u tohoto e-shopu, postačuje ruční přiřazení příbuzných sekcí v databázi. Dalším možným rozšířením je implementace hybridního doporučení.

Aplikace je vytvořena v jazyku C#, implementace je provedena ve vývojovém prostředí Microsoft Visual Studio Ultimate 2013, a jako databázový server byl využit Microsoft SQL Server 14. Aplikace je optimalizována pro webový prohlížeč Google Chrome, který patří v posledních letech mezi nejvíce používané prohlížeče s celosvětovým zastoupením 51,74 % pro červenec roku 2015 (StatCounter Global Stats, 2000). Optimalizace pro zbylé prohlížeče je nad rámec rozsahu bakalářské práce.

I. SUMMARY AND KEYWORDS

The aim of this bachelor thesis is to propose and implement a recommender system for an e-shop, that selects and offers specific objects of users interest based on its database and according to their enter. The theoretical part introduces the issue of user preferences and explains a few basic implementation approaches. Furthermore, the reader learns about developing tools used for implementation the module of recommendation, which uses C# as a programming language. The properties of this system are described in practical part using samples of individual functionalities.

Keywords: Recommender system, user's preferences, e-shop, database

II. SEZNAM POUŽITÝCH ZDROJŮ

Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *Ieee Transactions On Knowledge And Data Engineering*, 17(6).

ASPone,. (2007). Specializovaný hosting – ASPone.cz. *Webhosting, freehosting, virtuální servery* / ASPone. Retrieved August 11, 2015, from <http://www.aspone.cz/cz/Specializovany-hosting/>

Bainbridge, W. S. (2004). *Berkshire encyclopedia of human-computer interaction*. Great Barrington, Mass.: Berkshire Pub. Group.

Barla, M., & Bieliková, M. (2009). On Deriving Tagsonomies: Keyword Relations Coming from Crowd. In *Computational collective intelligence semantic web, social networks and multiagent systems: first international conference, ICCCI 2009, Wroclaw, Poland, October 5–7, 2009 : proceedings* (Online-Ausg., pp. 309–320). Berlin [etc.]: Springer. doi:10.1007/978-3-642-04441-0_27

Blanco-Fernández, Y., Pazos-Arias, J. J., Gil-Solla, A., Ramos-Cabrer, M., López-Nores, M., García-Duque, J., *et al.* (2008). A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. In *Knowledge-Based Systems* (Vol. 21, pp. 305–320). Amsterdam, The Netherlands: Elsevier Science Publishers B. V. doi:10.1016/j.knosys.2007.07.004

Burke, R. (1999). Integrating Knowledge-based and Collaborative-filtering Recommender Systems. In *Artificial intelligence for electronic commerce: papers from the AAAI Workshop ; [.. was held 18 July 1999 in Orlando, Florida]* (pp. 69–72). Menlo Park, Calif: AAAI Press.

Burke, R. (2007). Hybrid Web Recommender Systems. In *The Adaptive Web* (pp. 377–408). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72079-9_12

Canter, D., Rivers, R., & Storrs, G. (1985). Characterizing user navigation through complex data structures. In *Behaviour & Information Technology* (Vol. 4, pp. 93–102). doi:10.1080/01449298508901791

- Cockburn, A., & Mckenzie, B. (2001). What do web users do? An empirical analysis of web use. In *International Journal of Human-Computer Studies* (Vol. 54, pp. 903–922). doi:10.1006/ijhc.2001.0459
- ETARGET CZ, (2002). Etarget – efektivní reklama na internetu. Retrieved August 10, 2015, from <http://www.etablet.cz/#1>
- Gauss Algorithmic, (2015). Inteligentní doporučení. *Analýza velkých dat, strojové učení a prediktivní analytika | Gauss Algorithmic*. Retrieved August 10, 2015, from <http://www.gaussalgo.cz/inteligentni-doporucovani/>
- Gilfillan, I. (2003). *Myslíme v MySQL 4* (1. vyd.st ed.). Praha: Grada.
- Herceg, T. (2007). Úvod do asp.net 2.0. *Největší český web zaměřený na .NET framework*. Retrieved August 11, 2015, from <http://www.dotnetportal.cz/clanek/44/Uvod-do-ASP-NET-2-0>
- Hogan, B. P. (2011). *HTML5 a CSS3: výukový kurz webového vývojáře* (Vyd. 1.). Brno: Computer Press.
- Holzschlag, M. E. (2006). *HTML a CSS: jdi do toho* (1. vyd.st ed.). Praha: Grada.
- Holub, M., & Bieliková, M. (2010). Vplyv vzorov v správání návštěvníkov webového portálu na odporúčania. In *5th Workshop on Intelligent and Knowledge oriented Technologies* (pp. 10–13).
- Hueter, G. J., Quandt, & Hueter, N. H. Universal system and method for representing and predicting human behavior. Retrieved from <http://www.google.com/patents/US8566256?hl=cs>
- Jameson, A., & Smyth, B. (2007). Recommendation to Groups. In *The Adaptive Web* (pp. 596–627). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72079-9_20
- Jannach, D. (2011). *Recommender systems: an introduction* (1. pub.st ed.). New York: Cambridge University Press.

- Jecha, T. (2009). Seznámení a instalace microsoft sql serveru. *Největší český web zaměřený na .NET framework*. Retrieved August 10, 2015, from <http://www.dotnetportal.cz/clanek/140/Seznameni-a-instalace-Microsoft-SQL-Serveru>
- Kunegis, J., Said, A., & Umbrath, W. (2009). The Universal Recommender. *Arxiv Preprint Arxiv:0909.3472*.
- Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations Item-to-Item Collaborative Filtering. In *IEEE Internet computing* (pp. 76–80). Los Alamitos: IEEE Computer Society.
- Microsoft, (2015). Edice systému SQL Server. *Oficiální domovská stránka Microsoft*. Retrieved August 10, 2015, from <http://www.microsoft.com/cs-cz/server-cloud/products/sql-server-editions/overview.aspx>
- Milic-Frayling, N., Jones, R., Rodden, K., Smyth, G., Blackwell, A., & Sommerer, R. (2004). Smartback. In *Proceedings of the 13th conference on World Wide Web - WWW '04* (pp. 63–71). New York, New York, USA: ACM Press. doi:10.1145/988672.988682
- MS Visual Studio,. (2015). Visual Studio – Microsoft Developer Tools. Retrieved August 11, 2015, from <https://www.visualstudio.com>
- Neumann, A. W. (2009). *Recommender systems for information providers: designing customer centric paths to information* (1st ed.st ed.). New York: Springer.
- Pazzani, M. J. (1999). A Framework for Collaborative, Content-Based and Demographic Filtering. In *Artificial Intelligence Review* (Vol. 13, pp. 393–408). Kluwer Academic Publishers. doi:10.1023/A:1006544522159
- Peška, L. (2010). *Uživatelské preference v prostředí prodejních webů* (Diplomová práce). Praha.
- Písek, S. (2006). *HTML: tvorba jednoduchých internetových stránek* (2., aktualiz. a dopl. vyd.nd ed.). Praha: Grada.
- Price, J. (2005). *C#: programování databází* (1. vyd.st ed.). Praha: Grada.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative Filtering Recommender Systems. In *The Adaptive Web* (pp. 291–324). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-72079-9_9

Stanek, W. R. (2013). *Microsoft SQL Server 2012: kapesní rádce administrátora* (1. vyd.st ed.). Brno: Computer Press.

StatCounter Global Stats,. (2000). Top 5 Desktop, Tablet & Console Browsers from July 2014 to July 2015. Retrieved August 17, 2015, from <http://gs.statcounter.com/>

Šíma, F., & Vilímek, D. (2006). *Microsoft Visual Studio .NET: praktické programování krok za krokem* (1. vyd.st ed.). Praha: Grada.

Wongchokprasitti, C., & Brusilovsky, P. (2007). NewsMe: A Case Study for Adaptive News Systems with Open User Model. In *Third International Conference on Autonomic and Autonomous Systems (ICAS'07)* (pp. 69–69). Los Alamitos, CA: IEEE. doi:10.1109/CONIELECOMP.2007.88

III. SEZNAM OBRÁZKŮ

Obrázek 1: Doporučovací systém Amazon	22
Obrázek 2: Doporučovací systém ETARGET	23
Obrázek 3: Architektura daného systému	33
Obrázek 4: Databázový slovník tabulky Produkt	34
Obrázek 5: Databázový slovník tabulky Sekce	35
Obrázek 6: Databázový slovník tabulky Hodnoceni	35
Obrázek 7: Databázový slovník tabulky Parametr	36
Obrázek 8: Databázový slovník tabulky ParametrTyp	36
Obrázek 9: Databázový slovník tabulky ParametrProdukt.....	37
Obrázek 10: Databázový slovník tabulky SekceParametr	37
Obrázek 11: Databázový slovník tabulky Doporucujeme	38
Obrázek 12: Databázové schéma	39
Obrázek 13: Grafické znázornění klauzule INTERSECT	41
Obrázek 14: Ukázka pohledu VypisProduktu	47
Obrázek 15: Ukázka pohledu VypisSekci 1	48
Obrázek 16: Ukázka pohledu VypisSekci 2	48
Obrázek 17: Ukázka pohledu VypisSelectoru – Intervalový filtr.....	48
Obrázek 18: Ukázka pohledu VypisSelectoru – Výčtový filtr	48
Obrázek 19: Ukázka pohledu Produkt	50
Obrázek 20: Ukázka pohledu Index.....	52
Obrázek 21: Ukázka pohledu Delete	53
Obrázek 22: Ukázka pohledu Details	53
Obrázek 23: Ukázka pohledu Create	53
Obrázek 24: Ukázka pohledu Edit	53
Obrázek 25: Ukázka pohledu Index.....	53

IV. SEZNAM ZDROJOVÝCH KÓDŮ

Ukázka kódu 1: Úprava ceny produktů při změně sazby DPH	40
Ukázka kódu 2: Generování podmínek pro SQL dotaz, který provádí výběr produktů splňující kritéria filtrů	42
Ukázka kódu 3: Načítání produktů pomocí technologie JavaScript.....	51

V. SEZNAM PŘÍLOH

Příloha A – Datový nosič

Obsah přiloženého CD

- Zdrojový kód webové aplikace;
- záloha databáze ve formátu .bak;
- tato práce ve formátu PDF.