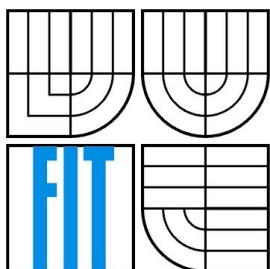


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

WEBOVÝ PORTÁL SKLADOVÝCH ZÁSOB

WEB PORTAL OF THE GOODS STORE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. TOMÁŠ OBRÁTIL

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MARTIN DRAHANSKÝ PH.D.

BRNO 2008

Abstrakt

Tato diplomová práce popisuje webový portál firmy ZZM spol. s r.o., který má zlepšit dostupnost zboží a služeb zákazníkům firmy a umožnit vyhodnocování odběru a oběhu zboží v jednotlivých oblastech spravovaných firmou ZZM. Projekt využívá technologií PHP a MySQL. V projektu jsou použity technologie pro zabezpečení přístupu, autentizaci uživatelů a session pro sledování pohybu a získávání informací o zákazníkovi, jakožto i technologie pracující na bázi OLAP pro vyhodnocování získaných informací a následnou optimalizaci distribuce zboží. Webový portál bude samostatná aplikace komunikující se stávajícím interním systémem firmy ZZM na základě komunikačního protokolu PDK verze 6.

Klíčová slova

webový portál, PHP, MySQL, ZZM, podpora marketingu, session, OLAP.

Abstract

This master thesis presents ZZM spol. s r.o.'s web portal of good store, which should improve availability of goods and services to ZZM's customers and make a good way to evaluation of taking and running the goods in a separate regions managing by ZZM. Project make use of PHP and MySQL technology. Application includes technology for autentization, security and session for following behaviour and obtaining information about customers. For better decision making aggregated data will be present in OLAP technology. Web portal will be independent application communicating with actual internal system of firm ZZM based on communication protocol PDK version 6.

Keywords

Web portal, PHP, MySQL, ZZM, support of marketing, session, OLAP.

Citace

Obrátil Tomáš: Webový portál skladových zásob. Brno, 2008, diplomová práce, FIT VUT v Brně.

Webový portál skladových zásob

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením

Ing. Martina Drahanského, Ph.D.

Další informace mi poskytli Ing. Tomáš Drahanský z firmy ZZM spol. s r.o. a Martin Pokorný z firmy Cronax spol. s r.o.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Obrátil
19.5.2008

Poděkování

Tímto bych chtěl poděkovat Ing. Martinu Drahanskému, Ph.D., Ing. Tomáši Drahanskému z firmy ZZM spol. s r.o. a Martinu Pokornému z firmy Cronax spol. s r.o. za jejich ochotu a čas, který mi věnovali při konzultacích mé diplomové práce, a za všechnu pomoc, které se mi dostalo.

© Tomáš Obrátil, 2008.

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

Obsah:

Obsah:.....	6
1 Úvod	8
2 Specifikace požadavků na webový portál firmou ZZM.....	10
2.1 Funkční požadavky	10
2.1.1 Obecná sekce	10
2.1.2 Sekce pro přihlášené.....	10
2.1.3 Sekce pro zaměstnance firmy ZZM	11
3 Současný stav	12
4 Použitelné technologie.....	13
4.1 Výběr vhodného programovacího jazyka.....	13
4.2 Výběr vhodného databázového serveru	14
5 Technologie pro tvorbu grafické stránky webu	16
6 Technický popis webového portálu.....	17
6.1 Webová aplikace.....	17
6.2 Databázový server.....	20
6.3 Přenosový protokol	22
6.3.1 Session.....	22
6.3.2 Protokol pro přenos informací mezi aplikací a serverem Neptun	23
7 Optimalizace OLAP	24
7.1.1 3NF.....	24
7.1.2 Presentace dat	25
7.1.3 Agregace.....	25
7.1.4 Přehled vlastností kontingenční tabulky.....	26
7.1.5 Přehled vlastností dynamické tabulky	26
7.1.6 Server pro OLAP.....	27
7.1.7 Získání dat	28
8 Implementace	29
8.1 Implementace aplikace.....	30
8.1.1 Konvence v pojmenovávání souborů	30
8.1.2 Funkce mysqlsession.....	30
8.1.3 Funkce login	30
8.1.4 Třída katalog.....	31

8.1.5	Třída vyhledávač.....	32
8.1.6	Třída leták.....	32
8.1.7	Třída databáze.....	33
8.1.8	Třída objednávka.....	33
8.1.9	Třída uživatel.....	34
8.1.10	Třída update.....	35
8.1.11	Třída košík.....	36
8.1.12	Třída olap.....	37
8.1.13	Funkce patička a hlavička.....	39
8.2	Ukázka operací v tabulce OLAP.....	39
9	Testování aplikace.....	42
10	Závěr.....	43
	Literatura.....	45
	Přílohy.....	46

1 Úvod

Ve své diplomové práci jsem se zabýval návrhem webového portálu skladových zásob. Toto zadání vzešlo z požadavků průmyslového zadavatele, firmy ZZM spol. s r.o.¹, který tento portál a s ním spojené služby hodlá využívat k efektivnímu rozjezdu a prodeji nového sortimentu náradí, který by rád zařadil do svého obchodního portfolia. Služba má nejen sloužit pro rychlejší a efektivnější spojení se svými klienty, ale také umožnit firmě kontrolovat, vyhodnocovat a sledovat oběh zboží, aby následně mohly být prováděny zásahy a optimalizace oběhu zboží.

Firma ZZM se snaží tímto krokem přispět ke spokojenosti svých klientů a nabídnout jim novou a pohodlnou službu, jak jednoduše a snadno objednávat nové zboží přes webové rozhraní a zároveň jak zkvalitňovat svoje nabízené služby a marketingové strategie, při jejichž rozhodování by jim měl pomoci nový systém pro podporu rozhodování OLAP, který umožní vyhodnocení prodejů zboží podle různých kategorií.

Projekt by měl být malý až středně velký, měl by obsluhovat kolem dvou tisíc zákazníků a obsahovat asi pět tisíc položek náradí. Systém by měl komunikovat se stávajícím interním systémem Neptun, který pro firmu ZZM spravuje a vyvíjí outsourcingová firma Cronax spol. s r.o.². Snadné použití této služby by mělo být zajištěno na straně klienta jen nutností používání nějakého dostupného webového prohlížeče, jako je např. Internet Explorer, Mozilla, Opera nebo další, které jsou dnes volně a snadno dostupné na každém PC. Pro budoucí rozvoj projektu se předpokládá, že k tomuto webovému rozhraní by mohl přibýt i nainstalovaný *think client* na straně zákazníka, který by umožňoval částečnou práci v off-line módu.

V první části dokumentu je popsáno konkrétní zadání, které vyplynulo z rozhovorů se zástupci firem ZZM a Cronax. V dalším oddíle jsou popsány možnosti a volba programovacího jazyka a také volba vhodné databáze, která bude v projektu použita s ohledem na dohodnuté konkrétní zadání. Následuje popis rozdělení na popis samotné aplikace, kde se uvádí její vnitřní rozdělení a způsob komunikace s dalšími systémy. Následuje popis zvolené databáze, její struktury a její schéma. Další odstavec slouží k představení optimalizací, které jsou řešeny technologií OLAP (*On-Line Analytical Processing*) pro snadnější rozhodování managementu firmy. V neposlední řadě je uveden komunikační protokol a informace o způsobu použití session, které zajistí předávání informací mezi jednotlivými stránkami.

¹ dále jen ZZM

² dále jen Cronax

Za touto teoretickou částí se nachází popis samotného řešení diplomové práce, popis důležitých částí systému a některá praktická řešení daného problému.

Jako poslední je závěr, který shrnuje zvolené technologie, problémy s praktickou implementací a návaznost této práce na budoucí možné rozšíření a další projekty, které by měly stavět na tomto modelu a prostředcích zde uvedených.

Cílem této práce je naplnit požadavky, které jsou na systém kladeny ze strany zadavatele, a zároveň splnit zadání, která mi byla uložena pro mou diplomovou práci. Na základě všech těchto požadavků vyvinout funkční aplikaci, kterou by bylo možno nasadit do praktického provozu ke spokojenosti zadavatele. Aplikaci navrhnout tak, aby umožnila budoucí rozšiřování a přidávání dalších částí a funkčních celků k již hotovému jádru aplikace.

2 Specifikace požadavků na webový portál firmou ZZM

2.1 Funkční požadavky

Systém musí umožňovat tři sekce. Jednu pro přihlášené klienty (sekce pro přihlášené), druhou pro nepřihlášené potenciální zákazníky (obecná sekce) a třetí pro zaměstnance firmy ZZM.

2.1.1 Obecná sekce

V obecné sekci bude všem poskytnuta možnost nahlížet do katalogu náradí, ovšem bez podrobných informací jako je cena pro zákazníka, dostupnost a další.

Také se zde budou nacházet volně přístupné informace jako jsou kontakty, úvod o firmě ZZM a odkazy na její další produktové sekce.

Dále pro ty, kteří by se chtěli stát partnery ZZM, bude připraven dotazník, který vyplněný bude předán obchodnímu zástupci firmy (email) v daném regionu pro možnost kontaktovat potenciálního zákazníka.

2.1.2 Sekce pro přihlášené

Musí obsahovat možnost přihlášení pod jednoznačným Id³ klienta a jeho heslem. Klient bude identifikován po celou dobu jeho pohybu po stránkách náradí firmy ZZM.

O klientovi budou shromažďovány informace během jeho připojení na stránky ZZM (co si prohlížel, co nakoupil, kdy se připojil, z kterého kraje (města) provedl objednávku (bude určováno pomocí PSČ, které má klient uvedeno u svých provozoven, od kterých přišla objednávka) a další vhodné informace, které pomohou zajistit lepší rozhodování a marketing firmy ZZM.

Klient bude mít možnost vidět katalog zboží se všemi informacemi o něm. Např. cena, která bude brána z obecného sazebníku, pokud klient pro tuto položku nemá svoji zvláštní cenu.

Klientovi bude umožněno zboží si vložit do košíku, ze kterého bude následně vygenerována objednávka.

³ Id - jednoznačný identifikátor

Vyhledávání požadovaného zboží bude možno dvojím způsobem:

- 1) podle sekcí (směsi, zednické nářadí, nářadí pro fasádníky...)
- 2) pomocí vyhledávače (podle kódu, jména, fulltextem)

Z nákupního košíku bude poté vygenerovaná objednávka - nutno se rozhodnout, zda prvně nezávazná objednávka, která jen potvrdí množství, které je firma ZZM schopna pokrýt, a pak teprve potvrzení takové objednávky, nebo se přímo vygeneruje závazná objednávka a následně se klient jen rozhodne, co udělá s položkami, které nemohou být vykryty z aktuálního množství na skladě.

Po odeslání závazné objednávky bude její potvrzení zobrazeno jak na obrazovce, tak na uvedený email.

2.1.3 Sekce pro zaměstnance firmy ZZM

Měla by být přístupná jen pro zaměstnance firmy a měla by umožňovat následující funkce:

- Umožní výpis informací o jednotlivých zákaznících.
- Umožní výpis kumulovaných informací pro jednotlivé zboží, jeho prodejnost, návaznost prodejnosti zboží na jednotlivé klienty a data prodeje.
- Umožní zadávat popis a obrázky jednotlivých produktů.

3 Současný stav

Aplikace by měla být samostatnou aplikací komunikující se systémem Neptun. Jelikož dochází k vzájemné komunikaci a aplikace Nářadí je jakousi podporou nebo nástavbou pro server Neptun, je nutné vycházet z již zavedených schémat a některých způsobů uložení dat. To se týká zejména práce s názvy jednotlivých výrobků, které už samy v sobě obsahují podrobnější informace. Ty jsou ale z názvu již dále systémově nedělitelné. Stejně tak je nutné dodržení již zavedeného komunikačního formátu pro snadnou kompatibilitu se serverem. Některé věci bylo nutné zjednodušit kvůli snadnějšímu chodu aplikace. Mezi tyto věci například patří výpočet ceny, která je v interním systému počítána trojstupňově, a je zcela zbytečné ji v aplikaci Nářadí implementovat tímto způsobem. Proto bude cena pro aplikaci Nářadí předpočítána a samotná aplikace jen nalezne odpovídající cenu pro daný výrobek, popřípadě pro daný výrobek a konkrétního uživatele. Je nutné také respektovat systém vytváření identifikátorů zboží. Tento identifikátor má stromovou strukturu nestejně hloubky větví a předem dané syntaxe.

Interní systém v současné době neobsahuje podrobnější slovní popisy jednotlivého zboží ani související obrázky. Tyto popisy by měli být získány po přeložení německého katalogu do češtiny třetí firmou, která by měla zajistit veřejnou část. Ta nebude vyžadovat přihlášení od uživatele. Tato data by se posléze měla připojit i do aplikace Nářadí, i když hlavní část aplikace by měla být textová a spíše orientovaná na nutná data a na rychlost možného nákupu.

4 Použitelné technologie

4.1 Výběr vhodného programovacího jazyka

Pro zvolený druh aplikace je možné buď využít skriptovaných jazyků jako jsou jazyky PHP, PERL a další, nebo využít rodinu jazyků, které vyžadují kompilaci, ať už do mezikódu nebo úplnou, jako jsou C++, C#, Java, .NET, ASP.NET nebo mnoho dalších. Z hlediska výběru mezi těmito jazyky je nutné vybrat možnost, která je široce podporována webhostingovými firmami. S tímto rozhodováním je spojena i nutná volba zda si vybrat jazyk, který běží jen na komerčně licencovaných serverech nebo pod licenci GNU jako svobodný software nebo jako open source. S touto volbou je spojena případná nutnost platit poplatky za licence. Jelikož by zvolená aplikace měla být určena pro webové rozhraní, pak se nejvíce nabízí jazyky Java, ASP.NET nebo PHP, které jsou z hlavní části pro tvorbu těchto aplikací tvořeny.

Jazyk ASP.NET je produkt firmy Microsoft a s tím spojená nutnost užití některého jejich serveru, protože tento jazyk je nepřenositelný a závislý na této architektuře. Tedy jako vhodnější se jeví jeden z kandidátů Java nebo PHP, kteří jsou volně dostupní a platformě přenositelní, navíc podpora ze strany webhostingových firem je podstatně větší.

Jazyk Java přenositelnost zaručuje tím, že zdrojový kód je při vývoji přeložen do spustitelného mezikódu (bytecode), který pak lze spouštět pomocí nainstalovaného runtime prostředí (*Java Virtual Machine*) přímo na různých typech počítačů či technických zařízeních bez nutnosti nového překladu.

Protože však tato přenositelnost není a nemůže být, hlavně z technickým příčin (rozhraní) 100 procentní, vyvinulo se několik edicí Javy, lišících se drobnými rozdíly a rozšířeními [1].

Jazyk PHP je také nezávislý na platformě, skripty fungují bez úprav na mnoha různých operačních systémech. Je to dáno tím, že pro tento skriptovací jazyk se využívá interpreta (nejčastěji serveru apache), který bez problémů můžeme využívat, jak na platformě Linux tak Windows.

Z tohoto důvodu se jako vhodnější jeví jazyk PHP, který sice není tak robustní jako jazyk Java, avšak nedochází k problémům s přenosem kódu mezi jednotlivými virtuálními prostředím, jako u jazyka Java, navíc jeho výpočetní síla je pro zamýšlený projekt zcela dostačující. PHP je jednodušší jazyk než jazyk Java, který se snaží do sebe začleňovat nové trendy a postupy z mnoha dalších programovacích jazyků. Je zpětně kompatibilní a umožňuje využívat vlastností více programovacích jazyků a tak nechává vývojáři částečnou svobodu v syntaxi. Navíc verze

PHP 5 v sobě již zahrnuje plnou podporu objektově orientovaného přístupu, podobného jazyku Java.

Oba dva jazyky jsou prováděny na straně serveru a na stranu klienta je přenášen pouze výsledek ve formě html. Tím je zajištěno, že klient nemusí mít nainstalovaný žádný speciální software, protože mu stačí běžně dostupné webové prohlížeče. Navíc je přenesen výpočetní výkon na stranu serveru, čímž není klientův systém nijak výpočetně zatěžován.

Jazyk PHP nabízí velké množství funkcí zajišťujících jak práci s dostupnými databázemi, tak tvorbu obrázků, grafů, zprávu sezení, autentizační funkce a spousta dalších potřebných rozšíření, které je možné v mé práci využít. Navíc je tento jazyk široce podporován, existuje o něm velké množství dostupné literatury, ukázkových aplikací a tutoriálů. Z hlediska nasazení na serverech Cronax se jeví také jako vhodný, neboť je plně podporován ve verzi 5. Jazyk také nabízí celou řadu dostupných rozšíření, prostřednictvím PEAR (*PHP Extension and Application Repository*) [2], který rozšiřuje nebo vylepšuje paletu funkcí, které nabízí samotné PHP. Prostřednictvím tohoto rozšíření jsou například dostupné moduly DB nebo ADODB, které nabízejí abstraktní rozhraní mezi PHP a databází, čímž dojde k nezávislosti zdrojových kódů PHP na jednotlivých typech databází. Tyto moduly podporují širokou paletu dostupných databází, jak komerčních tak freeware. Dále je zde dostupnost modulů pro autentizaci uživatele a ověřování hesel, vylepšené odesílání pošty, moduly pro code cache a spousta dalších.

4.2 Výběr vhodného databázového serveru

Databázových systémů je na trhu celá řada, od robustních a drahých systémů jako je např. Oracle, Access až po freewarové databázové systémy, jako jsou MySQL, Firebird, PostgreSQL, Berkeley, které sice nejsou tak robustní a neoplývají tolika možnostmi, ale zato jsou mnohem méně náročné na hardwarové prostředky aplikace a jejich pořizovací cena je velice nízká nebo nulová.

Pro webové aplikace jsou nejčastěji nabízeny databázové servery MySQL a PostgreSQL, jejich používání je zdarma.

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci.

MySQL je multiplatformní databáze. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

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

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování a až donedávna nepodporovalo pohledy, trigger, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet [3].

PostgreSQL je plnohodnotným relačním databázovým systémem s otevřeným zdrojovým kódem. Má za sebou více než patnáct let aktivního vývoje a má vynikající pověst pro svou spolehlivost a bezpečnost. Běží na všech rozšířených operačních systémech včetně Linuxu, UNIXů (AIX, BSD, HP-UX, SGI-IRIX, Mac OS X, Solaris, Tru64) a Windows. Plně podporuje cizí klíče, operace JOIN, pohledy, spouště a uložené procedury. Obsahuje většinu SQL92 a SQL99 datových typů, např. INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL a TIMESTAMP.

PostgreSQL je šířen pod BSD licenci, která je nejliberálnější ze všech open source licencí. Tato licence umožňuje neomezené používání, modifikaci a distribuci PostgreSQL. PostgreSQL je možno šířit se zdrojovými kódy nebo bez nich, zdarma nebo komerčně.

PostgreSQL umožňuje běh uložených procedur napsaných v několika programovacích jazycích, v Perlu, v Python, v jazyku C nebo v speciálním PL/pgSQL - jazyku vycházejícím z PL/SQL firmy Oracle. Existují PostgreSQL varianty JDBC, ODBC, dbExpress, Open Office, PHP, .NET Perl nativních rozhraní. K PostgreSQL existuje překladač Embedded SQL pro C a C++.

Rozhodování mezi těmito dvěma databázovými servery není lehké, protože když hledáte nějaké srovnání nebo názory lidí pracujících s těmito nástroji, tak co člověk to názor. Pro svou práci jsem si nakonec vybral databázový server MySQL, protože existuje nepřehledné množství dokumentace, různých tutoriálů, návodů a praktických řešení, které se dají použít jako zdroj vynikajících informací. Navíc většina publikací je uváděna právě v tomto spojení PHP - MySQL [4][5].

Rychlost i funkční vybavenost tohoto produktu zcela postačují pro chystaný projekt. Tento produkt je navíc podporován ze strany webhostingového partnera firmy ZZM a tudíž je možné ho plně využít.

5 Technologie pro tvorbu grafické stánky webu

Jelikož aplikace je vyvíjena pro WWW (*World Wide Web*), tak jako nejvhodnější použitelná technologie pro grafické ztvárnění webu, se jeví použití kaskádových stylů - css a použití javascriptu. Grafická úprava připravovaných stránek bude udělána na základě materiálů, které dodá grafické studio, pod něhož spadá tvorba celé mediální kampaně na výrobky ze sekce nářadí firmy ZZM. Grafický návrh dodaný grafickým studiem, by měl graficky sjednotit budoucí kampaň probíhající v jednotlivých médiích s webovým portálem, aby úspěšněji dosahoval marketingových cílů firmy a vystupoval v jednotném stylu s kampaní. Kaskádové styly budou muset být částečně přizpůsobeny pro jednotlivé prohlížeče, neboť zejména Internet Explorer jinak interpretuje význam kaskádových stylů a to zejména v oblasti vnějších a vnitřních okrajů. Neovlivitelnou částí aplikace budou roletová menu, která přebírají nastavení systému a jsou kaskádovými styly neovlivitelná. V horní části stránek bude hlavička s informacemi o firmě ZZM, která bude tvořena statickým obrázkem. Tvary tabulek do kterých se bude vypisovat budou také tvořeny pomocí stylů.

6 Technický popis webového portálu

6.1 Webová aplikace

Webová aplikace bude sestavena z jednotlivých tříd, které budou obsluhovat jednotlivé části aplikace, jako je obsluha přihlášení, vygenerování session⁴, komunikace s databází, vykreslování webových stránek, komunikace se systémem Neptun, grafické zobrazení získaných dat o prodeji různých druhů zboží a další podpůrné třídy pracující na úpravě databáze během její nečinnosti. Termínem vykreslování se myslí zápis kódu do standardu html, který posléze bude nastylizován příslušným kaskádovým stylem.

V aplikaci budou jedny z nejdůležitějších tříd `katalog` a `kosik`. Třída `katalog` bude umět načíst z databáze veškeré informace o náradí. Třída bude mít také metodu pro získání ceny, která se bude skládat ze základu a daňové části. Každý výrobek bude mít svou základní cenu, která je stejná pro všechny klienty s výjimkou těch, kteří mají stanovenou svoji vlastní cenu. Stanovování cen se bude uskutečňovat v systému Neptun. V aplikaci dojde jen ke zjištění, zda je pro daného klienta stanovena základní cena, nebo má svoji vlastní. Na základě toho, jestli je klient plátcem DPH, či nikoliv, mu bude v seznamu předkládána cena s DPH, nebo bez DPH, což klientům umožní snáze se orientovat v nabídce a rychleji se rozhodovat podle svých potřeb. Přidávání obrázků bude automatické. Podle výrobního čísla zboží se zkontroluje, zda je obrázek s daným číslem v adresáři s obrázky, a pokud ano, bude přidán ke kartě zboží, kde budou podrobnější informace o konkrétním druhu výrobku. V průběhu samotného hledání obrázky použity nebudou, tak jako tomu bývá u klasických e-shopů, protože by tento systém měl sloužit hlavně uživatelům, kteří si potřebují vyřídit nákup a orientují se převážně podle textových značek.

Pro nepřihlášené zákazníky nebo ty, jenž si chtějí procházet zboží jako na klasickém e-shopu, bude sloužit další aplikace, která je na stránkách `www.triuso.cz`. Zde jsou uvedeny výrobky, které jsou i v papírovém katalogu náradí. Náradí z webu `triuso.cz` lze spárovat v jednotlivých aplikacích podle jména nebo výrobního čísla. K tomuto účelu bude v mojí aplikaci udělán vyhledávač, který dokáže vyhledávat jak podle stromové struktury zboží, tak podle jejího výrobního čísla nebo plně fulltextově.

⁵ session - sezení (relace). V textu bude užíváno originálního anglického názvu `session`.

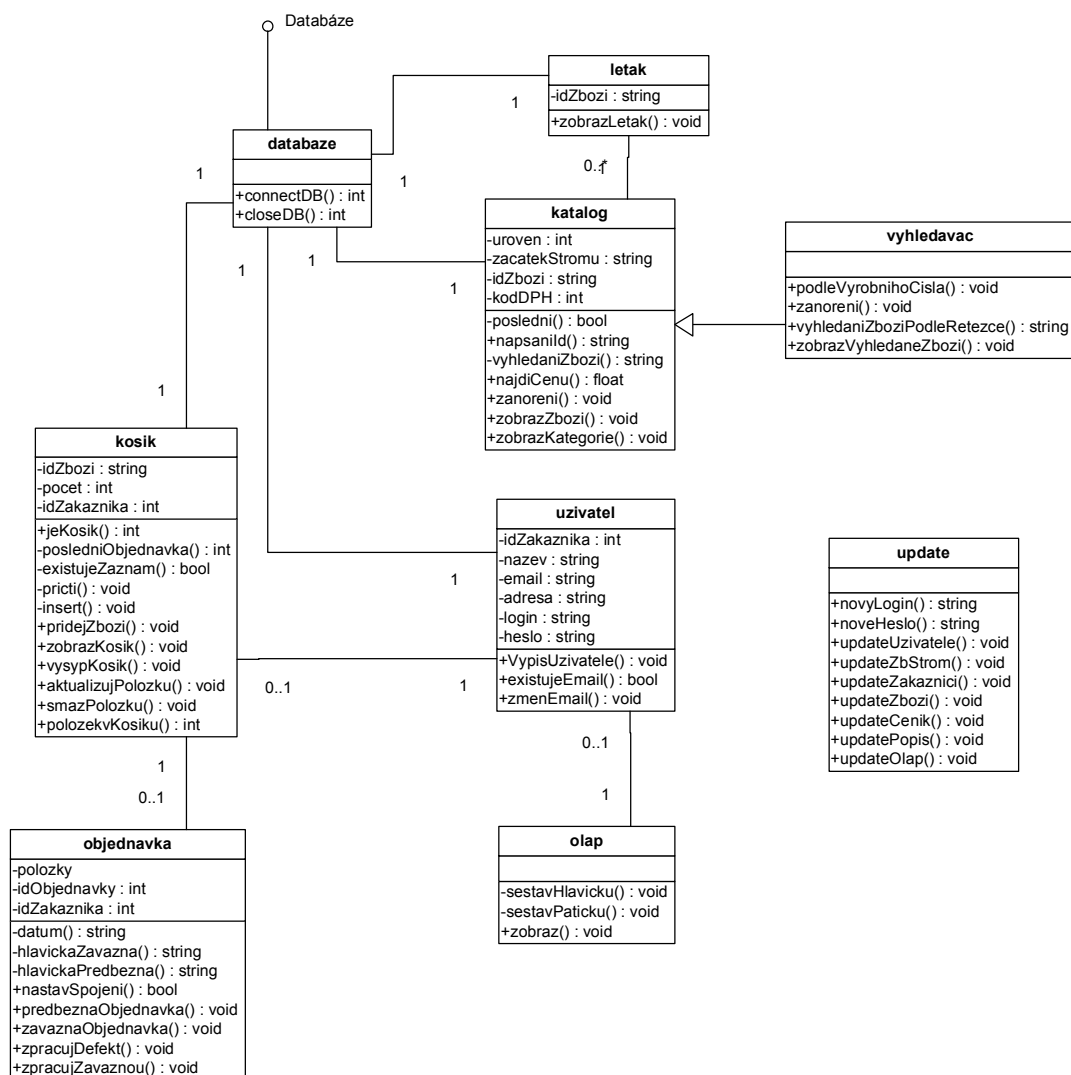
Aplikace bude také zajišťovat vygenerování objednávky a její odeslání internímu systému, který ji potvrdí. Pokud by došlo k situaci, že na skladě nebude dostatečné množství položek, bude klient dotázán, zda si přeje objednávku stornovat nebo zda si objedná zboží, které je vykryté, nebo si nechá objednat i zboží, které na skladě není, a to mu bude doručeno v nejbližší možné dodací lhůtě. Po potvrzení všech těchto kroků bude klientovi zobrazena informace o potvrzení objednávky a potvrzená objednávka mu bude poslána emailem a zapsána do historie nákupů. Sestavení potvrzení objednávky a její následné odeslání na kontaktní email zajistí nástroje php pro komunikaci pomocí emailu.

Popis komunikačního protokolu mezi aplikací a serverem Neptun následuje v odstavci 4.3.2 „Protokol pro přenos informací mezi aplikací a serverem Neptun“.

Aplikace bude obsahovat také třídu `letak`, která bude sloužit k vykreslení jednotlivých karet zboží. Na těchto kartách budou zobrazeny podrobnější informace o zboží, jako jsou jeho jméno, výrobní číslo, cena, DPH, rozměry, ale také, pokud jsou známy, podrobnější informace pro daný produkt nebo jeho obrázek. Tyto karty se budou vždy otevírat v novém okně, aby se klientovi nestalo, že si omylem zavře aplikaci a bude se muset znovu vyhledávat jednotlivé zboží.

Veškeré objednávky, které klient provede budou zaznamenány do jeho košíku. Košík je perzistentní a přetrvává i po dobu, kdy je klient odhlášen, což byl jeden z požadavků zadavatele. Košík se vyprázdní buď tím, že se odešle objednávka, která po potvrzení košík sama vyprázdní, nebo tím, že klient sám zvolí možnost vyprázdnění košíku. Práci s košíkem bude zajišťovat třída `kosik`, která bude obsahovat jak metody na vykreslení košíku, tak také metody pro jeho správu a editaci. V košíku bude možno jak měnit jednotlivé množství zboží, tak i mazat celé položky. Součástí zobrazení obsahu košíku bude i cena pro jednotlivé položky a celková cena celého nákupu, která bude opět udávána s DPH, nebo bez něj, podle toho, zda je klient plátcem DPH, či nikoliv.

Třída `update` bude sloužit k indexování, třídění a různým úpravám databáze během slabého vytížení serveru zejména v nočních hodinách, kdy server nebude příliš zatížen. Její převážná funkce bude spočívat v aktualizaci jednotlivých informací dodávaných z interního systému Neptun. Mezi tyto informace patří informace o klientech, zboží, cena jednotlivých položek a strom zboží. Aplikace také bude mít třídu `databaze`, která bude zajišťovat připojení k databázi.



Obr. 6.1: Diagram aplikace. Třída `update` slouží jako nezávislá třída, spouštěná externím programem, je určena pro úpravu databáze.

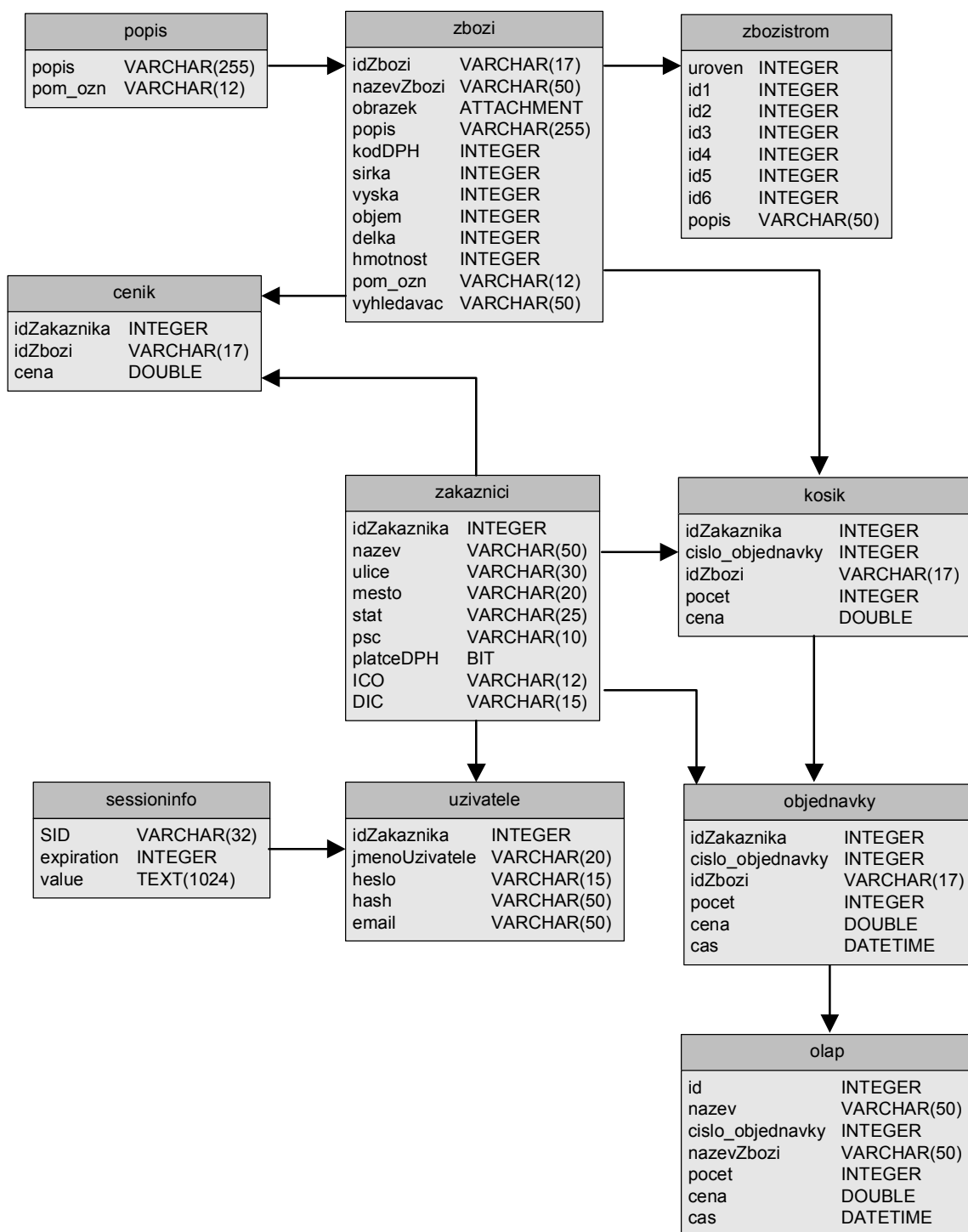
Problémy, které vzniknou interpretací HTML kódu a kaskádových stylů v jednotlivých prohlížečích, je možné obecně obejít nastavením různých hodnot okrajů a hodnot jednotlivých atributů pro jednotlivé prohlížeče v kaskádových stylech. Nejčastější problém vzniká právě mezi zobrazením v prohlížeči Internet Explorer 6 a prohlížeči Mozilla a Opera. Prohlížeče Opera a Mozilla se chovají v převážné většině podobně a více ctí standard. Zobrazované informace budou testovány pro základní a nejvíce rozšířené prohlížeče, kterými jsou v dnešní době Internet Explorer 6, Internet Explorer 7, Mozilla a Opera. Šířka sloupce, do kterého budou data vykreslována, bude pravděpodobně odvozena od šířky stále nejčastěji používaného rozlišení, které je podle serveru e-komerce.cz 1024 × 768px [6].

6.2 Databázový server

Databázový server MySQL bude sloužit jako úložiště informací o produktech. Informace a produkty budou v databázi rozděleny do tabulek. Data související s nářadím a se všemi informacemi o klientech, se budou do databáze plnit periodicky z interního informačního systému Neptun. Zbylé informace jako je doba strávená na jednotlivých stránkách, co si klient koupil. Ke komunikaci s databází bude sloužit třída `database`, která zajistí persistentní spojení s databází. Dotazy na databázi budou kladeny pomocí klasických php metod jako je `mysql_query()`.

Databáze bude obsahovat tabulky, jak je uvedeno na obrázku 1 a bude rozdělena na dvě pomyslné části.

Jedna část databáze bude obsahovat informace o zboží, jako je jeho skladové číslo, rozměry, specifikace výrobku, odkaz na obrázek. Ceny pro jednotlivé výrobky se stanoví z ceníku, který jako klíč obsahuje jak název položky, tak id zákazníka, pokud má stanovenou speciální cenu jen pro sebe. Pokud nebude nalezeno spojení identifikátoru zboží a konkrétního přihlášeného zákazníka, pak bude cena pro výrobek stanovena podle normálního sazebníku, který je společný pro všechny klienty. Výsledná cena se v zobrazení v aplikaci ještě upravuje podle toho jestli je zákazník plátce DPH či nikoliv. Pokud není, tak je cena uvedena s DPH. Kód DPH je uváděn u jednotlivého zboží.



Obr. 6.2. Databázové schéma aplikace.

Druhá sekce v databázi bude obsahovat tabulky týkající se jednotlivých klientů. Bude obsahovat název jejich firmy, její sídlo, její provozovnu. Pokud bude více provozoven, ty pak budou uvedeny v samostatné tabulce, která bude svázána s tabulkou Klient pomocí

jednoznačného klientova identifikačního čísla. Tyto všechny informace budou periodicky plněny z interního systému Neptun, který firma ZZM používá pro svoje skladové hospodářství. Systém Neptun je také používán pro uchovávání informací o jednotlivých zákaznících, jejich speciálních cenících, tak i pro výpočet jednotlivých cen výrobků.

Podle identifikačního čísla klienta budou také uchovávány informace o jeho historii přihlášení a jeho nákupech. Pro zaměstnance firmy bude uvedena možnost, kdy si budou moci prohlížet historii jednotlivých klientů, jejich pohyb po stránkách, jejich nákupy a jejich historii přihlášení. Tímto způsobem by se do budoucna mělo umožnit zlepšování postupů, které klienti nejčastěji využívají pro nalezení zboží, které si chtějí koupit.

6.3 Přenosový protokol

Jako přenosový protokol mezi aplikací a internetovým prohlížečem klienta, který bude sloužit pro grafický výstup aplikace, byl zvolen protokol HTTP⁵ verze 1.1, který je široce podporován v síti www. HTTP je bezstavový protokol, proto bude nutné použít session pro uchování informací o pohybu klienta po stránkách firmy. Dalším protokolem použitým v aplikaci je protokol pro přenos informací mezi aplikací a současným interním serverem Neptun, který firma ZZM používá.

6.3.1 Session

Jako session se v IT rozumí trvajícím síťové spojení mezi klientem a serverem, zahrnující výměnu většího množství paketů.

V případě použití protokolů, které žádnou podporu pro sessions nemají (UDP)⁶ nebo kde spojení typicky trvá velmi krátkou dobu (HTTP), jsou session udržovány přímo aplikačním programem a k tomu nutné informace jsou vkládány do přenášených dat[6].

Typickým příkladem je použití HTTP cookie k uložení jednoznačného identifikátoru SID - session ID, který se dá do vzájemného vztahu s libovolným počtem jiných prvků dat, ať už je to počet návštěv za měsíc, oblíbená barva pozadí nebo cokoliv, co chcete dát do souvislosti s klientem.

⁵ Hyper Text Transfer Protocol

⁶ User Datagram Protocol - uživatelský datový protokol

Použití cookies není vynutitelné, protože klient nemusí mít cookies vůbec povoleny a i pokud je povoleny má, může dojít ke ztrátě informací v cookies v důsledku odstranění cookies na klientově stroji, popřípadě přihlašování se na jeden účet z více PC.

Jako zpracovatelé session proto budou sloužit vlastní funkce, které mi umožní ukládat session do databáze MySQL, namísto do cookies nebo jen přepisováním URL⁷. To mi umožní rozšířit paletu informací, které budeme moci trvale ukládat o klientech a o jejich chování. Také tím odpadá starost o to, zda má klient zapnuté cookies nebo ne a odstraní se tím i další problém, který vzniká u session přepisováním URL a to že informace není trvalá, ale dojde k její ztrátě po uzavření aplikace. Při přepisování URL také dochází k potenciální bezpečnostní díře, stejně tak jako v případě uložení cookies na straně klienta. K tomuto dochází, jelikož nemá přehled nad tím co se s cookies děje a kdo k nim může přistupovat a libovolně je měnit.

Session nám umožní zachovávat informace z minulých návštěv klienta, informace o tom, jaká má oprávnění k přístupu k aplikaci, jeho identifikátor, kterým bude jednoznačné id zákazníka, zda je plátce DPH či nikoliv a další informace, jako je čas posledního přihlášení a SID⁸. K vygenerování cookies a načtení jednotlivých informací o klientovi dojde při přihlášení klienta do systému v modulu pro přihlašování. Pro uchovávání informací bude sloužit databáze MySQL, kde bude vytvořena tabulka, která bude uchovávat informace o posledním přihlášení (čas, SID), oprávnění klienta a zda je někdo již pod patřičným heslem a jménem přihlášen.

6.3.2 Protokol pro přenos informací mezi aplikací a serverem Neptun

Jedná se o jednoduchý textový protokol, který je odvozený od komunikačního protokolu PDK verze 6. Tento protokol byl vyvinut pro komunikaci v oblasti léčiv, což je další oblast, kterou se firma ZZM zabývá. Z důvodu možné kompatibility v dalších aplikacích se bude tento protokol využívat i pro potřeby komunikace aplikace nářadí s serverem Neptun. Tento protokol řeší, jak má vypadat formát objednávky, defektní list, dodací list a rekapitulaci dodacích listů a vratek. Podrobnosti o přenosovém protokolu jsou uvedeny v přílohách, jako Komunikační formát PDK verze 6[viz. příloha]. Pro účely této aplikace nejsou nutná všechna pole, která komunikační protokol nabízí, protože jeho návrh pochází z oblasti léčiv.

⁷ Uniform Resource Locator - „jednotný lokátor zdrojů“

⁸ Identifikátor sezení

7 Optimalizace OLAP

V současné době se OLAP technologie jeví jako nejslibnější, co se týče podpory při rozhodování manažerských pracovníků, kteří rozhodují a optimalizují oběh zboží na základě historických dat o prodeích a dalších ekonomických transakcích. OLAP je část širšího pojmu s názvem Business Intelligence, který zahrnuje oblasti jako relační reporting a data mining. OLAP je technologie uložení dat v databázi, která umožňuje uspořádat velké objemy dat tak, aby byla data přístupná a srozumitelná uživatelům zabývajícím se analýzou obchodních trendů a výsledků. Také umožňuje pokládat ad-hoc dotazy nad velkými objemy dat. Není nutná stoprocentní správnost získaných dat, ale také jejich rychlost a komplexnost.

Základem jakéhokoliv OLAP systému je koncept OLAP kostky (často také nazývané multidimenzionální kostka nebo hyperkostka). Ta sestává z číselných faktů nazývaných fakta nebo dříve míry, které jsou rozříděné podle dimenzí. Metadata v kostce jsou typicky vytvořena pomocí schématu hvězdy nebo sněhové vločky, které reprezentují uložení dat v relační databázi.

Způsob uložení dat se svým zaměřením liší od běžněji užívaného OLTP (Online Transaction Processing), kde je důraz kladen především na snadné a bezpečné ukládání změn v datech v konkurenčním (mnohauživatelském) prostředí [9].

Základní rozdíly mezi OLAP a OLTP vyplývají z rozdílného použití - u OLAP se jedná o jednorázově nahrávaná data, nad kterými jsou prováděny složité dotazy, u OLTP jsou data průběžně a často modifikována a přidávána a to obvykle mnoha uživateli zároveň:

- OLAP nepoužívá na rozdíl od OLTP normalizované uložení dat v 3NF (bod 7.1.1) formě - data jsou uložena tak, aby umožňovala rychlou realizaci složitých dotazů, časté je zdvojené (redundantní) uložení, které by v případě OLTP komplikovalo provádění změn v datech
- OLAP používá podstatně více indexů než OLTP - opět to souvisí se zaměřením, kde indexy umožňují rychlé provedení složitých dotazů
- OLAP na rozdíl od OLTP často používá předpočítané agregované a odvozené hodnoty

7.1.1 3NF

3NF - (*Third Normal Form*) je soubor doporučení (metodika) pro návrh datové struktury databáze, jehož dodržení vede k optimálnímu využití vlastností systému OLTP při tvorbě databázových aplikací. 3NF obsahuje jako své podmnožiny soubor doporučení 2NF (*Second Normal Form*) a 1NF (*First Normal Form*), ale s ohledem na to, že na tyto podmnožiny již dnes

není prakticky vůbec odkazováno, jsou zde uvedeny základní doporučení 3NF bez ohledu na to, zda patří pouze do 3NF, nebo pouze do 2NF a 3NF, nebo do všech tří souborů:

- Eliminuj duplicitní sloupce v jednotlivých tabulkách.
- Pro každou skupinu dat s jasně vymezeným významem vytvoř zvláštní tabulku, každý řádek opatří unikátním primárním klíčem.
- Obsahem jednotlivých sloupců tabulky by měla být jednoduchá, dále nedělitelná informace.
- Podmnožinu dat se shodnou hodnotou pro určitý sloupec tabulky převed' do samostatné tabulky a spoj s původní tabulkou cizím klíčem.
- Odstraň z tabulky sloupce, které jsou přímo závislé na jiné skupině sloupců tabulky než pouze na primárním klíči.

7.1.2 Prezentace dat

Pro zobrazení dat se v dnešní době ustálil výraz prezentace dat. Prezentace nám zajišťuje zobrazení multidimenzionální kostky do dvou dimenzionální průmětny, ať už je to monitor počítače nebo třeba výstup z tiskárny. Prezentace musí zajistit, aby zobrazení multidimenzionální kostky bylo pokud možno, co nejpřehlednější a i přes to byla stále zachována funkčnost modelu a možnost vykonávat operace, které jsou na něj kladeny. Mezi tyto opera patří *pivotng*, který umožňuje zaměňovat a třídit sloupce klíčů. Další operací, kterou by měl model splňovat je *rollup*, *drilldown*. Pomocí těchto operací lze zakrýváním klíčových sloupců zvyšovat a snižovat agregaci. V neposlední řadě by měl model umožnit všechny sloupce zneviditelnit, případně nastavit filtry – *slice & dice*. Při zneviditelnění všech sloupců, by měl model umožnit vypsání jen plně agregovaných faktů pro všechny dimenze.

7.1.3 Agregace⁹

Jak již bylo řečeno pro komplexní dotazy nad OLAP kostkou by měli produkovat odpovědi v řádu desetin času oproti těm samým dotazům nad OLTP relačními daty. Jedním z nejdůležitějších mechanismů v OLAP, který dovoluje dosahovat těchto výkonů, je užití agregátů. Agregovaná data jsou vytvářena z tabulky faktů změnou granularity specifických dimenzí a agregováním dat podle těchto dimenzí. Počet možných agregátů je dán kombinací všech možných kombinací zobrazení multidimenzionální kostky. Počet pohledů, je-li k počet

⁹ Agregace - spojování, seskupování, shlukování

dimenzí je pak permutací k prvků tedy $k!$. Zde již vidíme, že pro 5 dimenzí je možný počet pohledů na data roven číslu 120.

Kombinací všech možných agregátů a základních dat dostáváme odpověď na každý dotaz, který může být zodpovězen z uložených dat. Vzhledem k potenciálně obrovskému počtu agregátů, které by měly být spočítány, často je pouze dopředu dáno menší množství agregátů, které budou předem plně vypočteny a zbytek bude spočítán až na požádání. Problém rozhodnout, která data mají být agregována (zobrazena) je také znám jako *view selection problem* (problém výběru zobrazení). Problém zobrazení může být ohraničen celkovým množstvím vybraných množin agregátů a jejich časem nutným k aktualizování ze základních dat nebo přímo čtením z agregátů. Cílem výběru zobrazení je minimalizování průměrného času nutného k odpovědi v systémech OLAP.

7.1.4 Přehled vlastností kontingenční tabulky

Řádky i sloupce jsou dimenze. Je možné vytvářet i hierarchické dimenze. Fakta jsou průsečíky dimenzí, fakt je proto v zásadě pouze jediný, více faktů se v průsečíku hůře zobrazuje, což se řeší záložkami nebo více sloupci, což je nepřehledné.

- Dimenze lze přesunovat a zaměňovat – pivoting.
- Zakrýváním hierarchie dimenzí lze zvyšovat a snižovat agregaci – rollup, drilldown.
- Všechny sloupce lze zneviditelnit nebo nevyužívat – slice & dice.

Jak je vidět, tak kontingenční tabulka splňuje všechny operace, které OLAP systém by měl umožňovat. Kontingenční tabulka je prostorově méně náročná, avšak nejsou vidět všechny faktové položky a hierarchie dimenzí může být nepřehledná [10].

Také zorientování se v kontingenční tabulce není tak snadné, protože zvláště pivoting zcela obrací tabulku a převrací zde celé sloupce a řádky mezi sebou. Stejně tak zobrazení více hodnot do jednoho průsečíku sebou nese jisté zneprůhlednění zobrazovaných dat, kde uživatel snadno může ztratit orientaci a možnost rychlého vyhodnocení prezentovaných dat. Zejména z těchto důvodů jsem dal ve své práci přednost dynamické tabulce, jejíž popis následuje v kapitole 7.1.5.

7.1.5 Přehled vlastností dynamické tabulky

Tento druh zobrazení neumožňuje hierarchii dimenzí, tzn. dimenze jsou ploché. Tabulka je rozdělena na sloupce dimenzí a sloupce faktů. Řádek dynamické tabulky tvoří uspořádanou n -tici dimenzních a faktových položek.

- Sloupce klíčů lze zaměňovat a třídit – pivoting.

- Zakrýváním klíčových sloupců lze zvyšovat a snižovat agregaci – rollup, drilldown.
- Všechny sloupce lze zneviditelnit, případně nastavit filtry – slice & dice.

Dynamická tabulka je prostorově náročnější, avšak jsou vidět všechny agregační i dimenzní položky [11].

Dynamická tabulka i přes absenci hierarchie dimenzí působí mnohem přehlednějším dojmem a data jsou zní mnohem lépe čitelná. Navíc je mnohem snadnější sledovat prováděné operace, protože ty nezpůsobují nikdy záměnu řádků a sloupců v prezentaci. Je zde také mnohem lépe vyřešeno zobrazení více faktů pro dané dimenze. Toho je dosaženo tím, že fakta jsou přidána jen jako další sloupce na konec tabulky za sloupce dimenzí a nejsou nijak omezeny počtem, jen velikostí plochy určené pro prezentaci.

Z těchto důvodů jsem si ve své aplikaci zvolil prezentaci dynamickou tabulkou.

7.1.6 Server pro OLAP

Datová skladiště mohou být implementována na standardních nebo rozšířených relačních databázových serverech, nazývaných Relational OLAP (**ROLAP**). Tyto servery předpokládají, že data jsou uložena v relačních databázích a poskytují rozšíření SQL a speciální přístup a implementační metody pro efektivní implementaci multidimenzionálního datového modelu a operací.

Naopak multidimenzionální OLAP (**MOLAP**) servery jsou servery, které přímo ukládají multidimenzionální údaje ve speciálních datových strukturách a implementují OLAP operace nad těmito speciálními datovými strukturami.

Jako aplikační server bude v mojí práci pro analýzu dat použito řešení ROLAP. Použití serverů MOLAP v mojí implementaci používat nebudu, protože objem dat nebude nikdy tak velký, aby se musel zavádět speciální druh serveru, který navíc webhostingový partner firmy ZZM v současné době neposkytuje a do budoucna se o něm ani neuvažuje. Navíc pořízení specializovaného serveru sebou nese další náklady na jeho pořízení, vyškolení údržby a jeho provozování.

V aplikaci nářadí na serveru bude implementováno hvězdicové schéma. Což znamená, že databáze sestává z jedné tabulky hodnot a jedné tabulky pro každou dimenzi. Každá n-tice v tabulce hodnot sestává z ukazatele (cizí klíč – často je pro efektivitu užíván generovaný klíč) do každé dimenze, který poskytuje jeho multidimenzionální souřadnice. Každá tabulka dimenze sestává ze sloupců, které odpovídají atributům dimenze. Hvězdicové schéma neposkytuje explicitně podporu pro hierarchii atributů.

Další možností kterou by bylo možno použít pro implementaci databázového schématu, by bylo schéma sněhové vločky. Schéma sněhové vločky poskytuje zjemnění hvězdicového schématu tak, že hierarchie dimenzí je explicitně reprezentována normalizováním tabulek dimenzí. To vede k výhodné údržbě tabulek dimenzí. Nicméně nenormalizovaná struktura dimenzionální tabulky ve hvězdicovém schématu může být mnohem vhodnější pro procházení dimenzemi. Toto schéma v našem případě, stejně nevyužijeme, protože dynamická tabulka, která byla zvolena k reprezentaci dat, hierarchické zanoření dimenzí nepodporuje.

7.1.7 Získání dat

Jako podklad pro prezentaci dat bude ve výsledné aplikaci použita tabulka `olap`, která bude získávat data z uzavřených objednávek, jejíž data budou v tabulce `objednavky`. Tyto data budou muset projít úpravou, aby co nejlépe vyhovovala rychlé a snadné prezentaci v dynamické tabulce. Jako jeden z kroků bude nahrazení číselných zkratk, která budou použita v tabulce `objednavky`, za plnohodnotná jména, snadno čitelná pro uživatele. Tohoto nahrazení bude použito, jak u jmen firem, tak u jmen zboží. Toto zpracování značně urychlí výsledné zobrazování, protože nebude nutné provádět žádné složité dotazy na operační tabulky pomocí prvků `join` při každém zobrazení prezentace. Úpravou také projde sloupec s datem uvedené transakce. Ten bude v tabulce objednávky uložen v plném formátu a to jak datum, tak i hodina, minuta a sekunda. Této přesnosti není v OLAP prezentaci potřeba a proto čas bude upraven jen na zobrazení data, proběhlé transakce. Sloupec faktů `cena`, bude agregován jako počet kusů zboží krát jeho jednotková cena.

K úpravě tabulky `olap` bude docházet periodicky jednou denně a to vždy v noci, kdy budou inkrementálně přidána nová data, která přibyla minulý den. Tento způsob je dostatečný pro možnosti rozhodování a také nijak nezatěžuje server během pracovního dne, v době největšího provozu. Úprava dat pro OLAP bude součástí skriptu `update`, který periodicky zajišťuje všechny potřebné operace pro údržbu databáze v nočních hodinách.

8 Implementace

Implementace projektu vychází z teoretického úvodu. Jako programovací jazyk je použit jazyk PHP verze 5.2.5 a jako aplikační databáze je použita MySQL verze 5.0.26. Pro komunikaci se serverem Neptun je použito komunikačního protokolu PDK verze 6. Při implementaci bylo nutné vycházet z již stávajících schémat a některých zavedených popisů, které jsou používány v interním systému Neptun. K výměně dat dochází nahráním speciálních tabulek do mojí aplikační databáze, ze kterých jsou data transformována do tabulek, které přímo využívá moje aplikace. K tomu transformování dochází v době, kdy je server nejméně vytižen v nočních hodinách a tím hrozí malé riziko přílišného zatížení serveru. Stejně tak bylo nutné se držet zavedené konvence u stromu zboží. Hierarchie zboží je stromová, s nestejnou délkou jednotlivých větví. Délka větví je obecně hloubky 4 až 6, což si vynutilo kontrolní mechanismy, které kontrolují, zda nalezený prvek ve stromě je již jako list nebo jen další uzel stromu.

Při implementaci jsem využíval objektového programování. Také jsem ve své práci využil dědičnosti tříd k rozšíření stávající třídy a modifikování její metody. Tento postup byl zvolen u třídy `vyhledavac`, která rozšiřuje vlastnosti třídy `katalog`.

V projektu jsem s úspěchem využil vlastní implementace zacházení s cookies. Toto řešení zvyšuje bezpečnost aplikace, neboť neumožňuje klientovi přistupovat k vlastním cookies, které se nacházejí na serveru. Stejně tak pro klienta není možné cookies jakkoliv modifikovat. Osvědčil se mi způsob psaní všech názvů vkládaných souborů do jednoho speciálního souboru, který je pak už jako jediný vkládán do scriptů využívajících dané třídy. Toto řešení zabraňuje tomu, že při změně názvu nějaké třídy zapomeneme někde změnit její jméno uvnitř vykonávacího scriptu.

Při psaní jsem se rozhodl, že budu používat zápis jen v PHP kódu, nikoliv kombinaci PHP a HTML. Při potřebě zápisu nepřerušovaného víceřádkového html kódu jsem využil zápisu pomocí dokumentového stylu u funkce `echo`. Jedná se o použití syntaxe `echo <<<END text v HTML kódu END;` Ukončovací `end` se musí nacházet na samostatném řádku a nesmí mu předcházet žádná mezera ani tabulátor. Pro přehlednost kódu jsem využíval odsazení vnořených funkcí. Také jsem využíval automatickou úpravu zarovnání kódu, kterou nabízí PHP editor. Při psaní mi byly neocenitelnou pomůckou takzvané `templates`, do kterých je možno si naprogramovat často opakovaný kód, který je posléze přístupný přes textové zkratky. Tento postup mi ušetřil spoustu času a zbytečného vypisování.

Programování jsem prováděl na lokálním počítači, který byl nastaven stejně jako server, což mi umožnilo programovat i bez přístupu na internet.

8.1 Implementace aplikace

8.1.1 Konvence v pojmenovávání souborů

Při pojmenování souborů byla dodržována tato konvence. Všechna písmena v pojmenování souborů budou malá. Názvy všech tříd začínají písmenem *x*, např. `xkosik.php` nebo `xvyhledavac.php`. Tyto scripty obsahují jen definici třídy a její metody. Volání této třídy zajišťuje jiný script, obvykle pojmenovaný stejně, jako třída avšak bez písmene *x* na začátku jména souboru. Například soubor volající metody z třídy `kosik` ze souboru `xkosik.php`, se bude jmenovat `kosik.php`. Tento postup umožňuje se rychle orientovat v názvech souborů a již na první pohled by mělo být z názvu souboru patrná jeho funkce. Veškeré obrázky zboží jsou pojmenovány podle výrobního čísla zboží a jsou v samostatném adresáři nazvaném `produkt_img`. tento adresář ještě obsahuje složku `tmb`, která obsahuje náhledy jednotlivých obrázků. Veškeré obrázky zboží mají příponu `.jpg`

8.1.2 Funkce `mysqlsession`

V rámci této sady funkcí je implementováno vlastní zacházení s cookies, které jsou tímto ukládány nikoliv na straně klienta, ale na straně serveru do speciální tabulky v databázi. Musí být implementováno přesně šest funkcí i když by některá měla být prázdná. Tyto funkce jsou `session_open`, `session_close`, `session_select`, `session_write`, `session_destroy` a `session_garbage_collect`. Tyto funkce nám nahradí stávající obsluhu `session`. Uvnitř těchto funkcí si můžeme definovat libovolné chování `session`. K uvedení těchto funkcí do logiky `php` slouží funkce

`session_set_save_handler()`. Implementace byla přebrána a upravena z [12]. Za zmínku stojí jenom změna uvnitř funkcí, kde místo na klienta je dotaz prováděn do databáze. A při vytvoření `session` kontrola zda je vytvořeno trvalé spojení k databázi a pokud ne, tak jeho vytvoření. O kódování a dekodování vlastního obsahu `session` se už stará samo `php`.

8.1.3 Funkce `login`

Na začátku každé stránky je funkce, která kontroluje zda je uživatel přihlášen. Pokud je uživatel přihlášen, pak se normálně pokračuje ve vykonávání dalšího kódu. Pokud však uživatel přihlášen není, tak se mu nabídne stránka pro přihlášení. Důvod proč nemusí být uživatel přihlášen je ten, že právě na stránku vstoupil, došlo k odhlášení v důsledku delší nečinnosti uživatele nebo k jeho

odhlášení ze systému pomocí tlačítka odhlásit. Kontrola zda je uživatel přihlášen je na základě vytvořeného cookies a naplněného identifikátoru uživatele. Z vloženého hesla a jména se vygeneruje jednosměrnou funkcí SHA1¹⁰ [13] hash, který je porovnán s hashem uloženým v databázi. Jestliže je hash totožný, pak je nastaven identifikátor uživatele a zda je plátce daně či nikoliv do cookies a uživatel je vpuštěn dále do systému. Po zadání špatných údajů je uživatel upozorněn, na špatné jméno nebo heslo a je vyzván k opětovnému vložení přihlašovacích údajů. Po úspěšném vložení je klient buďto vrácen na stránku na které se nacházel, pokud došlo k jeho odhlášení v důsledku nečinnosti nebo je odeslán na úvodní stránku systému.

Pokud klient přistupuje na stránky poprvé, pak je také dotázán na kontaktní email, který se uloží k jeho profilu. Tento krok je nutný z důvodu zajištění aktuální kontaktní emailové adresy pro tento objednávkový systém, na které si klient přeje komunikovat.

8.1.4 Třída katalog

Implementuje veškerou práci s katalogem zboží, jako je práce se sestavením katalogu ze stromu zboží, a navigace v něm. Při procházení stromem generuje navigační lištu s hloubkou zanoření a možností opětovného návratu do kterékoliv úrovně. Protože strom může mít nestejnou délku větví, bylo při práci s ním nutno kontrolovat zda je daný prvek uzlem stromu nebo jeho listem. Kontrolu jsem vyřešil dotazem na databázi zda aktuální prvek je již v hierarchii poslední nebo se za ním nachází nějaký další prvek. Pokud ano, pak je prvek uzlem a jednotlivé prvky úrovně dané větve se vypisují jako odkazy na další nižší úroveň stromu. Pokud se zjistí, že prvek je již konkrétní zboží, tzn. za prvkem již není další nižší úroveň, pak je prvek vypsán jako konkrétní zboží s polem pro objednání konkrétního výrobku a s možností prohlédnutí si karty zboží. Protože popis stromu se liší od id zboží v tom, že id zboží obsahuje vždy dvojčíferné číslo, na rozdíl od stromu, tak bylo nutno implementovat metodu, která umožňuje ze zjištěné cesty stromem sestavit identifikátor zboží a toto zboží posléze vypsát.

Speciální metodou je najít ceny, která není v databázi přímo u karet jednotlivého zboží, ale v samostatném ceníku. Tento ceník zohledňuje jak obecné ceny, tak ceny speciální pro jednotlivé vybrané klienty. Pokud se zjistí, že existuje složený klíč z id zboží a identifikátoru uživatele, pak je použito této ceny, pokud takováto cena neexistuje, pak je použita cena obecná pro daný výrobek. Pokud klient není plátcem DPH, pak je mu cena vynásobena s DPH a uváděna

¹⁰ SHA hashovací funkce je pátou kryptografickou hasovací funkcí vydanou NSA (*National Security Agency*). Byl publikován v NIST (*National Institute of Standards and Technology*) jako *U.S. Federal Information Processing Standard*

včetně DPH. Jestliže klient je plátcem DPH a tudíž je pro něj výhodnější vidět cenu bez DPH, pak je mu zobrazena bez DPH.

8.1.5 Třída vyhledávač

Třída vyhledávač rozšiřuje třídu katalog a zavádí do ní nové metody. Jsou to metody pro vyhledávání podle výrobního čísla a fulltextové vyhledávání. Vyhledání podle výrobního čísla jen provede dotaz do databáze a podle zadaného výrobního čísla najde konkrétní výrobek a jeho parametry. Tento způsob by měl být využíván zejména klienty, kteří si našli dané zboží v katalogu nebo na stránkách `www.triuso.cz`.

Třída upravuje metodu vypsání zanoření pro vyhledávání podle výrobního čísla, protože výsledek vyhledání nám vrátí o jednu úroveň navíc, která je nežádoucí ve výpise.

Pro fulltextové vyhledávání jsem zkoušel nejdříve použít fulltextový index přímo v databázi a práci s ním. Bohužel práce s tímto indexem, i přes to, že ve specifikaci bylo napsáno, že je *case insensitive*¹¹, se mi nezdařila. Neboť podle tohoto kritéria nebylo možno správně vyhledávat slova s velkými písmeny. Další problém se kterým se nedalo v tomto případě nic dělat byla čeština. Jestliže nebylo zadáno správné české označení, přesně tak, jak je v databázi, pak vyhledávání selhalo. Stejně tak se mi nepodařilo vyhledávání jen částí slov. Proto jsem si pro fulltextové vyhledávání navrhl vlastní metodu, pro kterou bylo nutno vytvořit podporu přímo v databázi a to přidáním redundatního sloupce u zboží. Tento sloupec obsahuje speciálně upravený obsah názvu zboží. Původní název zboží je zbaven všech nežádoucích znaků, poté je odstraněna čeština a nakonec jsou všechna písmena převedena na malá. Stejný postup je potom zvolen i u úpravy dotazu. Tato metoda si poradí jak s velkými a malými písmeny, tak i z češtinou nebo její absencí, protože tvar hledaného slova je vždy převeden na jednotný formát. K vytvoření tohoto sloupce pro vyhledávání, dochází při aktualizaci dat ze systému Neptun v nočních hodinách, kde nám případné zpracování většího množství dat nevadí.

8.1.6 Třída leták

Třída obsahuje jedinou metodu `zobrazLeak()`, která slouží zobrazení informací o zboží, jeho popisu a k zobrazení obrázku. Pro vypsání ceny je použita metoda ze třídy `katalog`. Obrázek je zobrazen jen pokud nějaký obrázek k danému zboží existuje. Jeho nalezení probíhá pomocí sestavení jména obrázku z cesty do adresáře obrázků a jeho vlastního jména, které se skládá

¹¹ Nerozlišuje velká a malá písmena

z výrobního čísla a koncovky jpg. Stejně tak, pokud není v rámci karty zboží uveden nějaký popis, tak se dotyčná položka popis vůbec nezobrazuje.

8.1.7 Třída databáze

Ve třídě databáze jsou jen jedna metoda pro vytvoření spojení na databázi, která vytvoří persistentní spojení s databází. Persistentní spojení je zde vytvářeno také proto, že spojení je využíváno také metodami cookies, kde je nezbytné, aby bylo spojení neustále otevřeno. V tomto důsledku nemá cenu implementovat metodu, která by spojení uzavírala, protože persistentní spojení nelze uzavřít po dobu běhu aplikace.

8.1.8 Třída objednávka

Tato třída slouží k vytvoření objednávky a ke komunikaci se serverem Neptun. Pro komunikaci se serverem je využito formátu PDK verze 6. Tento formát stanoví, jak má které pole vypadat. Jde o čistě textovou formu výměny dat. Třída obsahuje metodu pro sestavení hlaviček jednotlivých formátů pro komunikaci. Formát může být buď předběžná objednávka nebo závazná objednávka. To zda jde o závaznou objednávku nebo o předběžnou objednávku se hlavičce frází TEST. Pokud je zpráva odeslána s polem TEST, pak systém Neptun ví, že se jedná o předběžnou objednávku, na kterou reaguje tím, že odešle kolik kterého zboží bylo v objednávce vykryto a které případně je vykryto z části, či vůbec. Tyto informace jsou zobrazeny klientovi, který se na jejich základě rozhodne co bude dále se svou objednávkou dělat.

Možnosti jsou následující:

1. Objedná jen zboží, které je vykryté.
2. Objednání veškerého zboží. To, které není momentálně na skladě, klientovi bude dodáno v co nejkratší možné době.
3. Zruší celou objednávku a neobjedná si nic. tato operace vede k navrácení se do košíku, tak jak byl před provedením předběžné objednávky

Po té co bude objednávka kladně vyřízena se provede uložení položek do tabulky proběhlých objednávek, ze kterých se posléze generuje dynamická tabulka pro podporu rozhodování. Zpráva o vyřízení objednávky také bude zaslána uživateli na jeho kontaktní email, který si vyplnil při prvním přihlášení.

Celý systém výměny dat se serverem probíhá přes FTP (*File Transfer Protocol*), kde objednávka nebo předběžná objednávka je na server nahrána v podobě souboru s patřičnou příponou podle specifikace v komunikačním formátu PDK a poté je obsah FTP prostoru kontrolován, zda se v něm nalézají vygenerované soubory s odpověďmi. Přípona pro předběžnou i

závaznou objednávkou je vždy obj. Záleží jestli je v hlavičce uvedeno pole check, které říká, že se jedná o předběžnou objednávku. Pokud toto pole uvedeno není, pak se jedná o závaznou objednávku. Pokud na server byla zaslána předběžná objednávka, pak je vygenerován soubor se stejným jménem, ale příponou def. Jestliže se jednalo o závaznou objednávku, pak je serverem vygenerován soubor s příponou dod. Poté co je správný soubor nalezen, je dešifrován z jeho textové podoby a s příslušnými možnostmi předveden uživateli. Toto předvedení je formou tabulky, která nám ukazuje zda je zboží možno objednat v plné výši, nebo zda ne. Pokud není možné zboží objednat v plné výši, je ve sloupci poznámka napsán důvod, pro který objednání není možné.

Při předběžné objednávce je na straně serveru Neptun jenom zkontrolováno množství dostupného zboží na skladě, nikoliv jeho zablokování. Proto se může stát, že i když předběžná objednávka byla vykryta celá, tak výsledná objednávka může být z části nevykryta. Toto je riziko se kterým musí uživatel počítat a pokud možno dobu mezi objednáním zboží po potvrzení předběžnou objednávkou a jejím vlastním nákupem příliš neprotahovat. Tomuto by se v budoucnu dalo částečně zamezit tím, že pokud by předběžná objednávka byla zcela pokryta ustoupilo by se od kroku jejího potvrzení a rovnou by se vygenerovalo plné potvrzení objednávky i s jejím začleněním do systému Neptun. Je předpoklad, že když bude mít klient informace o tom, že je jeho poptávka plně uspokojena, tak stejně tuto předběžnou objednávku odešle jako závaznou.

V současné době je implementováno odeslání předběžné objednávky a přijetí odpovědi na ni. Odpověď je interpretována a zobrazena klientovi k dalšímu posouzení. Vygenerování objednávky je nachystáno, ale protože není v současné době implementován protějšek na straně serveru Neptun, je jen provedeno překopírování dat z košíku do tabulky `objednavka`, aby bylo možno testovat chování aplikace a zobrazení dat pro podporu rozhodování pomocí OLAP.

Komunikace se serverem Neptun přes FTP, se jeví v současné době, jako dostatečně rychlá a spolehlivá. při selhání komunikace je klient informován o tomto selhání a je vyzván buď k zrušení transakce nebo k jejímu opakování.

8.1.9 Třída uživatel

Tato třída se stará o výpis informací o uživateli po jeho přihlášení. Mezi tyto informace patří uživatelské jméno, adresa, PSČ, IČ, DIČ, a email určený pro komunikaci s portálem náradí. Také obsahuje metodu, která kontroluje při přihlášení, zda uživatel má zadán komunikační mail. Pokud tomu tak není, pak zařídí, aby byl do systému nový email přidán. Email je kontrolován na formální správnost emailové adresy podle specifikace RFC2822 [14]. V tomto případě není

ověřována živost emailové adresy klienta. Kontrola probíhá pomocí regulárního výrazu, který email ověří, zda je správně zapsán syntakticky.

8.1.10 Třída update

Tato třída se stará o veškeré práce na pozadí, které se provádějí v nočních hodinách. Jsou to věci jako je natahování aktuálních dat ze systému Neptun, jejich doplnění a úprava vhodná pro používání v mé aplikaci. Obnova dat probíhá tak, že systém Neptun také v nočních hodinách do databáze aplikace Nářadí exportuje do předem definovaných tabulek potřebná data. Tyto tabulky mají speciální název a jsou používány jenom jako rozhraní mezi aplikací Nářadí a systémem Neptun. Některé tabulky jsou v současné době jen zkopírovány do jiné tabulky, přesně tak jak jsou nachystány ze systému Neptun. To jsou data z tabulek `cenik`, `zakaznici`, `zboziStrom`.

V tabulce `uzivatele` jsou uložena hesla a loginy jednotlivých uživatelů. Tabulka `uzivatele` a `zakaznici` má vztah 1:1. Toto řešení jsem zvolil proto, aby bylo možné data z Neptunu jen okopírovat a spárovat s daty pro přihlášení v aplikaci Nářadí. To zaručuje, že i když se změní název firmy, pořád klientovi zůstávají ty samé přihlašovací údaje. Jelikož jsou noví uživatelé na straně Neptunu přidáváni nakonec seznamu podle id, což je zajištěno auto incrementem položky `id` u zákazníků, tak je možné snadno přidávat nové uživatele i do mé aplikace. Stačí jen vzít nejvyšší `id` zákazníka uloženého v tabulce `uzivatele` a poté vzít všechny nové zákazníky z tabulky `zakaznici`, vytvořit jim nové přihlašovací údaje a uložit je do tabulky `uzivatele`. Heslo i login jsou generovány náhodně. Login je generován jako číselný kód o velikosti osmi znaků. Heslo je také osmi znakové a je generováno jak z malých a velkých písmen, tak i z číslic. Tato hesla by měla splňovat podmínky pro bezpečná hesla, která nejdou prolomit slovníkovou metodou. Jejich délka osmi znaků je také v obvyklých mezích pro hesla. V tabulce `uzivatele` je také uložen hash z loginu a hesla, který se při přihlašování kontroluje se zasláným hashem, který je vytvořen na základě přihlašovacího formuláře.

Pro tabulku `zbozi` je nutné ještě doplnit popis zboží a sloupec pro vyhledávání. Popis zboží pochází z webu `www.triuso.cz`, který další firma zpracovala podle katalogu Nářadí, když ho překládala z německého originálu. Vytvoření sloupce pro vyhledávání funguje na principu, který je popsán v teoretickém úvodu. Nejprve jsou veškerá data zbavena nebezpečných znaků, poté je odstraněna čeština a nakonec jsou všechny velké znaky převedeny na malé. V tomto standardním tvaru je název uložen do sloupce `vyhledavac`, který se používá k fulltextovému vyhledávání. V tomto bodě je možné do pole pro vyhledávání také v budoucnu

uložit popis výrobku, ve kterém by se případně mohlo také vyhledávat. Tato metoda pracuje spolehlivě, jak s velkými písmeny, tak i s diakritikou.

Třída také obsahuje metodu pro předzpracování dat pro systém OLAP. Postup úpravy dat je popsán v kapitole 7.1.7. Protože přidávání dat do tabulky olap je jen inkrementální a pracuje jen nad daty, která vznikla od minulého zpracování, tak doba nutná k provedení těchto úprav není nikterak velká a je možno script provádět bez problémů v celku. V budoucnu pokud by se ukázalo, že doba nutná pro úpravu dat se mnohonásobně zvětšuje, je možno provádění této metody přenést do samostatného scriptu nebo nastavit pro tento scrip delší možnou dobu zpracování, než jaká je implicitně nastavena v PHP pro zpracování jednoho scriptu. V současné době je tato doba nastavena na 30 sekund. Druhou možností je nastavení 30 sekundového limitu pro každou prováděnou metodu nebo dokonce i v jejím průběhu. Tento postup nám zaručí, že každá metoda nebo dokonce její část bude mít na své provedení 30 sekund. Tento způsob prodlužování chodu scriptu však nefunguje pokud je zapnutý safe mod. V současné době je na serverech firmy ZZM vypnut.

Metody třídy je možné v budoucnu jakkoliv měnit, pokud se zjistí, že je nutné data přicházející ze systému Neptun jakkoliv modifikovat nebo jsou vyžadovány jiné úpravy dat před použitím v OLAP.

8.1.11 Třída košík

Třída košík se stará o veškerá data, která se týkají košíku a manipulaci s nimi. Při vkládání zboží do košíku zajišťuje jeho správné vložení. Data v košíku jsou organizována podle čísla objednávky, id zákazníka a id zboží. To zjistí jestli je v současné době pro daného klienta vytvořen košík, tzn. jestli je v košíku nějaká položka s klientovým id a tím pádem i odpovídajícím číslem objednávky. Pokud ne, tak se podívá do minulých objednávek, která objednávka byla poslední a košíku přidělí číslo o jedno vyšší. Pokud je toto vůbec první objednávka, pak je jí přiděleno číslo jedna. Pokud již v košíku je rozdělena nějaká objednávka, pak se zjišťuje jestli existuje i položka s konkrétním id zboží. Pokud ano, tak je její počet zvýšen o objednaný počet kusů. Pokud není, pak je vytvořena nová položka košíku pro daného zákazníka.

Metoda `zobrazKosik()`, se stará o zobrazení košíku do podoby editovatelné tabulky, kde můžeme editovat počet jednotlivého zboží. Stejně tak můžeme některé položky z košíku odstranit. Odstranění položky z košíku lze provést dvěma způsoby. První způsob je, že zatrheme položku „smaž položku“ a dáme aktualizovat košík, nebo zvolíme počet kusů roven nule. Košík

také uvádí u každé položky její cenu za kus, cenu celkovou a DPH. V posledním řádku je možné nalézt celkovou cenu za celý obsah košíku a celkové množství zboží.

K vyprázdnění košíku může dojít dvěma způsoby. Jeden je vysypání košíku pomocí tlačítka „Vyprázdni košík“, kdy dojde k úplnému zrušení všech položek v košíku. Druhý způsob jak dojde k vyprázdnění košíku je ten, že se provede objednávka zboží a tím se automaticky z košíku provedený nákup odstraní. Data v košíku zůstávají perzistentní i po odhlášení uživatele od aplikace, takže může si vybrat část nákupu poté se odhlásit a při příštím nákupu znovu pokračovat. Zde spatřuji velikou výhodu oproti řešení, kdy jsou položky udržovány jen v cookies, které by musely být trvalé, což zase zabraňuje automatickému odhlášení klienta od systému v případě dlouhé nečinnosti.

Košík má také metodu `polozekvKosiku()`, která nám vrací, kolik je položek v košíku. Tohoto se využívá u tlačítka košík při výběru zboží, aby byl klient informován kolik položek již v košíku má. A aby mohl sledovat, jak jeho výběr byl přidán do košíku.

8.1.12 Třída `olap`

Tato třída zajišťuje zobrazování informací pro podporu rozhodování v podobě dynamické tabulky. Obsahuje univerzální metodu `zobraz`, která vygeneruje dynamickou tabulku pro vložený počet sloupců dimenzí a faktů. K tomuto využívá také metodu `sestavHlavicku()`, která je schopná sestavit z jmen sloupců, které mají být ve výběru, požadovanou hlavičku tabulky pro daný případ. Tabulka je navržena tak, aby umožňovala operace `pivoting`, snižování a zvyšování agregace a nastavení filtrů `slice and dice`.

Při prvním přístupu ke scriptu, je inicializováno prvotní nastavení tabulky, do její plné podoby. Od této chvíle si tabulka sama uchovává a předává kontext mezi jednotlivými voláními sebe sama. Součástí kontextu je, který filtr byl vybrán v minulých zobrazeních tabulky, aby nový vybraný filtr zachovával kontext a vybíral data jen z těch polí tabulky, které byly momentálně zobrazeny. Pro uchování, zda sloupec je sloupcem faktů nebo dimenzí, slouží podtržítka před jménem sloupce s fakty. Tohoto rozlišení je zapotřebí pro možnosti nastavení filtrů, barvy záhlaví tabulky a možnosti `pivotingu` jednotlivých dimenzí. `Pivoting` je možný jen pro dimenze nikoliv pro fakta. Stejně tak je barevně odlišeno pozadí hlavičky tabulky pro dimenze a pro data. Samostatnou barvu má buňka pro řádky. Tento sloupec je zobrazován stále a není možné jej vypnout.

Jestliže jsou v tabulce zrušeny všechny dimenze, pak jsou vytisknuty celkové souhrny pro všechny položky nacházející se v tabulce `olap`. Tabulka má v tomto případě jen hlavičku, jeden řádek s celkovými součty pro počet položek a cenu a je ukončena patičkou. Souhrny

zobrazené v tomto jediném řádku musí mít stejnou hodnotu jako agregáty zobrazené v patičce tabulky při jejím maximálním zobrazení.

Cena výrobků je uváděna jako cena za jeden výrobek krát počet zakoupených výrobků, tudíž obsahuje souhrnou cenu za daný objem stejného zboží, koupeného v jedné objednávce daného klienta.

Konstrukce tabulky funguje na základě pole, do kterého jsou dána všechna jména sloupců, které se mají zobrazit. Přidání těchto jmen je možné dvěma způsoby. První způsob je při prvním použití scriptu, nebo při vrácení všech hodnot pole pomocí tlačítka „obnovit“, kdy je pole nastaveno na implicitní hodnoty. Nebo druhý způsob je, když je script volán již z formuláře, který je v dynamické tabulce a tím přenáší již hodnoty uvnitř pole `POST`, ze kterého jsou hodnoty do pole načteny. Dále je pak toto pole rozděleno na dvě menší, jedno, které obsahuje jen dimenze a druhé, které obsahuje jen fakta. V hlavičce je jinak obsluženo pole pro dimenze a jinak pro fakta. V metodě `zobraz()` je potom pro jednotlivé dimenze kontrolováno, zda se jedná o stejnou položku jako minule, nebo zda je již nová položka. Dokud se jedná o stálé n-tice položek dimenze, pak jsou fakta neustále agregována. Jakmile se najde nová kombinace položek dimenzí, pak je minulé n-tice položek dimenzí vytisknuta a s ní i příslušné hodnoty agregovaných faktů. Tento přístup nám umožňuje pracovat s tabulkou i když dopředu nevíme kolik bude mít sloupců dimenzí a kolik sloupců faktů. Ve scriptu bylo nutno ošetřit některé zvláštní případy, jako jsou absence všech dimenzí a jen jeden řádek tabulky, neboť zde nedochází k žádnému porovnání. V tomto případě je nutné načtené hodnoty rovnou vypsát na výstup. Stejně tak pokud nemáme žádné dimenze, pak je nutné agregovat data přes celý obsah a vypsát jen závěrečné agregáty. Pro správné zobrazení posledního řádku musíme po vypsání poslední různé hodnoty zajistit vypsání zbytkových dat, která jsou nachystána k dalšímu porovnání. Tyto data stačí na závěr jen vypsát a poté je možno přistoupit k uzavření tabulky patičkou.

Celá koncepce tohoto scriptu je navržena tak, že práce pro konkrétní data, která by se měla zobrazovat v dynamické tabulce jsou nastavena v samotné scriptu a poté už je jen volána třída `olap` a její metoda `zobraz()`. To nám v budoucnu umožňuje kdykoliv snadno přidat prezentaci nových dat, bez nutnosti zásahu do třídy `olap`, která je zcela abstraktní. Jen při vytváření tabulky v databázi je nutné dodržet konvence pro pojmenování sloupců. Sloupce se jmény dimenzí nesmí začínat znakem podtržítka, kdežto sloupce faktů tímto znakem začínat musí.

V patičce tabulky jsou zobrazeny jednotlivé souhrnné údaje. Ty se týkají souhrnného počtu zobrazených řádků, celkového součtu položek veškerého zobrazeného zboží, celkové ceny všech uvedených výrobků a v poslední řadě je to zobrazení průměrné ceny na jeden výrobek.

8.1.13 Funkce patička a hlavička

Tyto funkce slouží k vygenerování patičky a hlavičky html stránek. Funkci `hlavicka()` se jen předá, jaký titulek se má u stránky vygenerovat a ta se postará o vložení celého html kódu hlavičky. Obsahem hlavičky je také horní lišta, která se vypisuje na každou stránku zobrazenou uživateli. Součástí této lišty je tlačítko „odhlásit“, které umožňuje odhlášení uživatele. V tomto záhlaví se také nachází informace o firmě ZMZ, její logo a kontaktní email. Tento panel by měl také obsahovat logo firmy triuso.

Funkce `hlavicka` a `paticka` jsou zde proto, aby bylo snadné hlavičky a patičky stránek snadno měnit a měli jednotný styl a při případné změně nebylo nutné upravovat kód na každé stránce. Všechny funkce jsou obsaženy v souboru `funkce.php`, který sdružuje funkce, které se používají velmi často v různých skriptech.

Velice podobný je i účel souboru `soubory.php`, který obsahuje vložení veškerých nezbytných knihoven a funkcí, které se využívají v aplikaci a není nutné je všechny pořád všude vypisovat. Tímto krokem dojde sice k možnému nadbytečnému vložení některých částí kódu, který se zrovna v dané části aplikace nepoužívá, ale na druhou stranu nemusíme řešit neustálé hlídání si, zda jsme vložili všechny funkce a třídy, které budeme potřebovat.

8.2 Ukázka operací v tabulce OLAP

Obrázek č. 8.2.3. nám ukazuje OLAP tabulku se dvěma sloupci dimenzí a dvěma sloupci faktů. Sloupce dimenzí jsou v šedé barvě a sloupce faktů jsou fialové. V záhlaví tabulky jsou vidět pole pro zneviditelnění sloupců jednotlivých dimenzí, nastavení filtrů *slice and dice* a šipky pro *pivoting* dimenzí.

V zápatí tabulky je vidět počet vypsaných řádků, celkový součet všech položek, celková cena a její průměr na jednu položku.

Na obrázku 8.2.4. můžeme vidět provedení operace *pivoting*, kdy dojde k záměně pořadí sloupců `Firma` a `číslo objednávky`. Můžeme si povšimnout automatické změny řazení sloupců, protože sloupce jsou řazeny podle důležitosti a to zleva doprava.

Obrázek 8.2.5. nám ukazuje zvýšení agregace oproti obrázku 8.2.3. Zvýšením agregace nám ubylo řádků tabulky. To je dáno tím, že stejné n-tice tabulky, jsou agregovány a vypisovány jen do jednoho řádku tabulky.

Obrázek 8.2.6. ukazuje možnost nastavení filtrů, které zajišťují funkce *slice and dice*. A vybírají jen žádanou hodnotu ze sloupce dané dimenze.

Řádek	Firma	č.obj.	Počet	Cena
1	J.O.M. spol. s r.o.	1	16	1946
2	J.O.M. spol. s r.o.	2	2	426
3	J.O.M. spol. s r.o.	3	4	1016
4	J.O.M. spol. s r.o.	4	1	120
5	J.O.M. spol. s r.o.	5	34	5621.82
6	J.O.M. spol. s r.o.	6	9	1618
7	J.O.M. spol. s r.o.	7	6	981
8	J.O.M. spol. s r.o.	8	4	508.2
9	J.O.M. spol. s r.o.	9	3	522.1
10	J.O.M. spol. s r.o.	10	133	12369
11	Phmr. Milan Gajdušek	1	20	6128.5
12	Phmr. Milan Gajdušek	2	3	2095.59
13	Phmr. Milan Gajdušek	3	11	2291.94
13			246	Σ 35644.15 Ø 144,89

Obr. 8.2.3. OLAP tabulka

Řádek	č.obj.	Firma	Počet	Cena
1	1	J.O.M. spol. s r.o.	16	1946
2	1	Phmr. Milan Gajdušek	20	6128.5
3	2	J.O.M. spol. s r.o.	2	426
4	2	Phmr. Milan Gajdušek	3	2095.59
5	3	J.O.M. spol. s r.o.	4	1016
6	3	Phmr. Milan Gajdušek	11	2291.94
7	4	J.O.M. spol. s r.o.	1	120
8	5	J.O.M. spol. s r.o.	34	5621.82
9	6	J.O.M. spol. s r.o.	9	1618
10	7	J.O.M. spol. s r.o.	6	981
11	8	J.O.M. spol. s r.o.	4	508.2
12	9	J.O.M. spol. s r.o.	3	522.1
13	10	J.O.M. spol. s r.o.	133	12369
13			246	Σ 35644.15 Ø 144,89

Obr. 8.2.4. Pivoting dimenzí Firma a číslo objednávky.

Řádek	Firma	Počet	Cena
1	J.O.M. spol. s r.o.	212	25128.12
2	Phmr. Milan Gajdušek	34	10516.03
2		246	Σ 35644.15 Ø 144,89

Obr. 8.2.5. Agregace hodnot pro dimenzi Firma.

Řádek	Firma	č.obj.	Počet	Cena
1	J.O.M. spol. s r.o.	1	16	1946
2	J.O.M. spol. s r.o.	2	2	426
3	J.O.M. spol. s r.o.	3	4	1016
4	J.O.M. spol. s r.o.	4	1	120
5	J.O.M. spol. s r.o.	5	34	5621.82
6	J.O.M. spol. s r.o.	6	9	1618
7	J.O.M. spol. s r.o.	7	6	981
8	J.O.M. spol. s r.o.	8	4	508.2
9	J.O.M. spol. s r.o.	8	3	522.1
10	J.O.M. spol. s r.o.	9	133	12369
11	Phmr. Milan Gajdušek	10	1	6128.5
12	Phmr. Milan Gajdušek	1	3	2095.59
13	Phmr. Milan Gajdušek	2	11	2291.94
13			246	Σ 35644.15 Ø 144,89

Obr. 8.2.6. Možnost nastavení filtrů pro jednotlivé dimenze.

9 Testování aplikace

Aplikace byla testována v průběhu vývoje a to postupem od zdola nahoru. Tzn. nejdříve byly testovány jednotlivé moduly a jejich chování. Tyto moduly byly posléze začleňovány do větších celků, kde bylo testováno, zda celky správně spolu komunikují a chovají se správně ve vzájemné interakci. Některé části aplikace jsou zatím jen v testovací fázi. Touto částí je zejména spojení se serverem Neptun, protože na serveru Neptun ještě není vytvořen plnohodnotný modul, který by splňoval všechny požadavky, kladené na spojení s tímto serverem. Tím i data získaná z prodejů jsou jen testovací a jsou uměle vytvářena, aby bylo možno část aplikace zabývající se optimalizacemi a projekcí faktů o prodejích testovat a prezentovat. Modul spojení na serveru Neptun v současné době odpovídá jen na zaslání předběžné objednávky a vygenerování její odpovědi. Podpora pro testování závazné objednávky zatím nebyla napsána.

Při testování aplikace bylo zjištěno, že na testovacím stroji na rozdíl od serveru funguje příkaz `escapeshellcmd()`, který by měl být běžně dostupný. Scripty využívající tento příkaz byly normálně prováděné, jen nedocházelo k ošetření vstupů a mohlo by tímto dojít k bezpečnostní díře v aplikaci. Pokud by nedošlo k odhalení tohoto problému, pak by veškeré vstupy z formulářů mohli představovat potenciální možnost útoku pomocí *SQL injection*¹², kterou je možné ukrást data z databáze, nebo v horším případě získat až administrátorskou identitu a pod tou se přihlašovat do systému.

¹² SQL injection je technika napadení databázové vrstvy programu vsunutím (odtud „injection“) kódu přes neošetřený vstup a vykonání vlastního, samozřejmě pozměněného, SQL dotazu

10 Závěr

Tato diplomová práce mi byla velkým přínosem, protože mi bylo umožněno pracovat na reálné aplikaci. Asi jedním z největších problémů při tvorbě, je nutnost komunikace s více lidmi a čekání na výsledky jejich práce, což prodlužuje dobu, která je nutná k vytváření aplikace. Při psaní práce jsem si dobře uvědomil, že priority práce se v čase mění a je nutné na ně reagovat. Zajímavé bylo také nahlédnutí do reálně fungující firmy a jejich potřeb. Jako velice přínosný mi byl předmět „Pokročilé informační systémy“ s Prof. Ing. Tomášem Hruškou, CSc. Bohužel tento předmět je vyučován až v posledním semestru magisterského studia a proto některé podnětné nápady nebylo již možno do aplikace zahrnout. Doufám, že mi ale pomohou v dalších projektech podobného typu.

Aplikace je v současné době stále v testovací fázi a na jejím rozvoji by se mělo pokračovat i po skončení této diplomové práce. Aplikace by se dále měla rozšiřovat, jak množstvím informací, které budou k dispozici pro rozhodování, tak i přidáváním další funkčnosti jakou by mohlo být hlídání zaplacení faktur nebo nabízení expertních informací k jednotlivému druhu zboží. Tímto by chtěla firma ZZM ještě více přispět ke spokojenosti zákazníků, protože v dnešní době je čím dál tím těžší sledovat poslední trendy a nabídka produktů je natolik velká a v některých směrech specifická, že bez pomoci je velice těžké se v ní orientovat. S přidáváním funkcností, které budou mít nějaký měřitelný charakter, je možné také rozšiřovat množství informací, které budou sloužit jako podpora pro rozhodování managementu firmy.

V aplikaci Nářadí jsem splnil všechny cíle, které byly zadány pro mou diplomovou práci. V současné době je vytvořena funkční aplikace, která umožňuje výběr, prohlížení zboží a manipulaci s ním v košíku. Do aplikace byly také zabudovány nástroje pro vyhodnocení oběhu zboží v podobě OLAP, aby bylo možné následně provést jeho optimalizace. Doufám, že tato část bude pro firmu ZZM přínosem a v budoucnu povede ještě k lepšímu ekonomickému růstu firmy. Projekt je řešen modulově a proto přidávání dalších modulů a rozšiřování funkčnosti by v budoucnu mělo být snadné.

Použité technologie jsou v dnešní době natolik rozšířené, že případný přechod k jinému webhostingovému partnerovi by neměl být problémem. Stejně tak PHP i MySQL jsou v dnešní době živými projekty a neustále se pracuje na jejich zdokonalování a rozšiřování. Tento trend by měl zajistit, že aplikaci bude možno dále využívat a rozšiřovat, což je také jedním z důležitých požadavků kladených na současné aplikace. Výběr technologií, které byli použity v aplikaci se mi zatím jeví jako bezproblémový a dostatečně rychlý. Spolehlivost a bezproblémový chod ale

ukáže až testování více uživatelů současně. Zde se také projeví případné nedostatky v rychlosti aplikace i použitých technologií.

Literatura

- [1] Hatina, Petr: Programování v jazyku Java (1) - Úvod, 9.7.2004
URL http://www.linuxsoft.cz/article.php?id_article=244, (prosinec 2007)
- [2] Wiesemann, Mark, a jiní. PEAR - PHP Extension and Application Repository,
URL <http://pear.php.net/> (prosinec 2007)
- [3] MySQL, 30. 1. 2008, URL <http://cs.wikipedia.org/wiki/Mysql> (leden 2007)
- [4] Gilmore, W Jason: Velká kniha PHP MySQL 5, Zoner Press ZR617, 2005,
ISBN 80-86815-53-6
- [5] Lacko, Jaroslav: PHP a MySQL Hotová řešení, CP Books, a.s. 2005 ISBN 80-251-0397-8
- [6] Sklar, David: PHP 5 - moduly, rozšíření a akcelerátory, Zoner Press ZR424, 2005,
ISBN 80-86815-19-6 strana 25
- [7] Kozák, David, Navrcholu.cz: Rozlišení 1024x768 stále dominuje, 17. února 2006
URL <http://www.e-komerce.cz/ec/ec.nsf/0/2BEC48A2429E0376C1257117006A69D0>
(leden 2007)
- [8] Session, 18.11.2007, URL <http://cs.wikipedia.org/wiki/Session> (prosinec 2007)
- [9] Wiley, John, & Sons: OLAP Solutions: Building Multidimensional Information Systems,
2nd Edition. ISBN 978-0471149316.
- [10] Hruška, Tomáš: opora pro předmět Pokročilé informační systémy - Technologie OLAP -
kontingeneční tabulka str.80, FIT VUT Brno 2004
- [11] Hruška, Tomáš: opora pro předmět Pokročilé informační systémy - Technologie OLAP -
dynamická tabulka str.78, FIT VUT Brno 2004
- [12] Gilmore, W Jason: Velká kniha PHP MySQL 5, kapitola 18 zpracovatelé sezení, Zoner
Press ZR617, 2005, ISBN 80-86815-53-6, s. 431 - 450
- [13] Secure hash standard, 17 April 1995, URL <http://www.itl.nist.gov/fipspubs/fip180-1.htm>
(květen 2008)
- [14] Network Working Group, Internet Message Format, April 2001, URL
<http://tools.ietf.org/rfc/rfc2822.txt> (květen 2008)

Přílohy

Příloha 1 *Screenshot* z aplikace Nářadí

Příloha 2 *Screenshot* z aplikace Nářadí

Příloha 3 *Screenshot* z aplikace Nářadí

Příloha 4 *Screenshot* z aplikace Nářadí

Příloha 5 Komunikační formát PDK verze 6

Zobraz kategorie

Základní úroveň > Hrabla > Hrabla

Název zboží	DPH	Cena bez DPH		
Shrnovadlo plastové 50/120cm s držadlem	19 %	175,00 Kč	1	<input type="button" value="objednat"/>
Lopata hliník 23x26cm, násada 68cm s T rukojetí	19 %	312,00 Kč	1	<input type="button" value="objednat"/>
Lopata hliník 33x40cm, násada 130cm	19 %	426,00 Kč	1	<input type="button" value="objednat"/>
Lopata hliník 33x40cm, násada 74cm s D rukojetí	19 %	0,00 Kč	1	<input type="button" value="objednat"/>
Lopata hranatá s vyšším okrajem 470mm, nás.130cm	19 %	0,00 Kč	1	<input type="button" value="objednat"/>
Lopata hranatá s vyšším okrajem 580mm, nás.130cm	19 %	0,00 Kč	1	<input type="button" value="objednat"/>
Lopata na sníh nerez, násada 135cm, bez hrany	19 %	0,00 Kč	1	<input type="button" value="objednat"/>
Lopata na sníh nerez, násada 135cm, s hranou	19 %	0,00 Kč	1	<input type="button" value="objednat"/>
Shrnovadlo hliníkové 50/135cm, jemný lem, Triuso	19 %	500,00 Kč	1	<input type="button" value="objednat"/>
Shrnovadlo plastové 50/135cm, Al lem, Alu násada	19 %	372,00 Kč	1	<input type="button" value="objednat"/>
Shrnovadlo dřevěné 55/135cm Al lem Kobold	19 %	285,00 Kč	1	<input type="button" value="objednat"/>
Shrnovadlo hliníkové 50/135cm, Al lem, Kobolt	19 %	302,00 Kč	1	<input type="button" value="objednat"/>
Lopatka na posyp	19 %	62,00 Kč	1	<input type="button" value="objednat"/>

Příloha 1 *Screenshot* z aplikace Nářadí – Zobrazení výpisu zboží s možností objednání.

Zobraz kategorie

Hladítka
Hrabla
Latě
Lžice
Metry
Michani
Nářadí
Obkladac
Ostatni
Pokryvac
Regály
Sádrokarton
Topora
Zahrada, dům
zámečnik, zednik, čepice, šály rukavice
Oblečení, boty
Malování

Příloha 2 *Screenshot* z aplikace Nářadí - Zobrazení stromu zboží.

Karta Zboží

Název Zboží:	Sekera 1250g/700mm lakované jasanové toporo
Výrobní číslo:	AE1250A
DPH:	19 %
Cena:	228,00 Kč
Šířka balení:	0 mm
Výška balení:	0 mm
Objem balení:	0
Délka balení:	0 mm
Hmotnost balení:	1790 g

Popis:
s lakovaným jasanovým topůrkem, TÜV-kontrola



Příloha 3 *Screenshot* z aplikace Nářadí - Zobrazení karty zboží. Zobrazení je uvedeno v plné formě s popisem a obrázkem.

Váš nákupní košík

Smaž položku	Množství	Název zboží	DPH	Cena za jednotku bez DPH	Celkem
<input type="checkbox"/>	1	Hladítko dřevěné s brusným potahem, hrubost 16, na	19 %	268,00 Kč	268,00 Kč
<input type="checkbox"/>	1	Sekera 1250g/700mm, lakované jasanové toporo	19 %	345,00 Kč	345,00 Kč
<input type="checkbox"/>	1	Sekera 600g/360mm, lakované jasanové toporo	19 %	194,00 Kč	194,00 Kč
<input type="checkbox"/>	1	Sekera 600g/360mm, lakované jasanové toporo	19 %	180,00 Kč	180,00 Kč
<input type="checkbox"/>	1	Sekera 600g/380mm, lakované jasanové toporo	19 %	132,00 Kč	132,00 Kč
5 Položek					1 119,00 Kč

Příloha 4 Screenshot z aplikace Nářadí - Nákupní košík se zbožím určeným k objednání.

Příloha 5 Komunikační formát PDK verze 6.

Pharmdata s.r.o

Komunikační formát 6

Komunikačního formát PDK verze 6

revize k 9.10.2007

Komunikační rozhraní řeší:

- časový nesoulad upgrade číselníku PDK u distributorů a v lékárnách
- možnost průběžné aktualizace kódů PDK v lékárnách.

Obecná pravidla pro formát komunikačních souborů:

Všechny soubory dále uvedené jsou v textové řádkové formě s proměnnou délkou polí, jako Kódová stránka je čeština CP 852 LATIN 2. Numerické údaje používají jako desetinnou tečku Každé pole typu Datum je jednotně uvedeno v textové formě ve formátu rok - 4 místa, měsíc - 2 místa, den - 2 místa (RRRRMMDD) např. 16.květen roku 2006 je : 20060516.

Povinné pole

Objednávka

Název souboru obsahuje číslo objednávky s příponou ".OBJ". Pole označené jako nepovinné není nutné uvádět.

hlavička:

číslo verze	3	číslo verze komunikačního formátu objednávky (tato verze je 6)	Ano
Pharmdata kód odběratele	7	pokud je vyplněn má prioritu před následujícím údajem	Ne
kód odběratele	16	kód, pod kterým je odběratel registrován u dodavatele	Ano pokud není předchozí údaj
kód dodavatele	16	kód, pod kterým je dodavatel registrován u odběratele (např. IČO, DIČ)	Ano
číslo objednávky	12	číslo objednávky odběratele	Ano
datum	8	datum vystavení objednávky	Ano
příznak testu	4	text "TEST"	Ne
datum dodání	12	požadovaný okamžik dodání buď jen jako datum nebo datum a čas - formát RRRRMMDD nebo RRRRMMDDHHII (míněno rok,měsíc,den,hodina,minuta).	Ne
místo dodání	20	kódové nebo textové označení bližšího určení místa dodání (např. u velké nemocnice - dodávka přímo na kliniku)	Ne

druh objednávky	1	0 nebo neuvedeno=běžná, 3=kompensační, 5=transferová. Kompensační objednávka je objednávka vystavená kupujícím na zboží zdarma od prodávajícího. Důvod kompenzace je předmětem vzájemné dohody mezi prodávajícím a kupujícím.	Ne
transfer firma	16	kód firmy která zadala transferovou objednávku, povinné u transferové objednávky	viz vlevo
transfer zastupce	16	kód zástupce který zadal transferovou objednávku, povinné u transferové objednávky	viz vlevo
id akce	16	Identifikační kód akce (transferu)	Ne

položky:

druh kódu	1		Ano
kód	16		Ano
množství	12,2		Ano

textová část:

Textová část objednávky je na začátku řádku označena slovem "TEXT". Počínaje další řádkou do konce souboru následuje libovolný text (informace dodavateli).

druh kódu:

- 0 APA
- 1 SÚKL - pozor 7 znaků
- 2 EAN
- 3 PDK
- 4 VZP PZT
- 8 čárový kód
- 9 kód dodavatele

Zboží s neurčeným druhem kódu nebude při objednávce akceptováno.

Pokud bude zásilka v hlavičce označena textovou informací „TEST“, je objednávka považována za testovací. Bude provedeno potvrzení zboží (odeslán defektní list), ale nebude realizována dodávka zboží. Pokud textový údaj „TEST“ nebude uveden, jedná se o normální zakázku. Podrobný popis zpracování objednávky je uveden u defektního listu.

Příklad souboru:

Objednávka : 0005541.obj

6|1600200|010-45316490|0005541|200605101530|TEST|20060512||||||

0|0000280|150.00|

1|0051621|100.00|

3|6895873000126|10.00|

4|32521|13.00|

9|0071499|50.00|

1|0000105|5.00|

TEXT

Pošlete nám, prosím, nový katalog zboží.

Děkuji Nováková

Defektní list

Název souboru obsahuje číslo objednávky s příponou ".DEF". Pole označené jako nepovinné není nutně uvádět.

hlavička:

číslo verze	3	Číslo verze formátu defektního listu	Ano
Pharmdata kód odběratele (PDZ)	7	pokud existuje má prioritu před následujícím údajem	viz objednávka
kód odběratele	16	Kód, pod kterým je odběratel registrován u dodavatele	
kód dodavatele	16	Kód, pod kterým je dodavatel registrován u odběratele (např. IČO, DIČ)	Ano
číslo objednávky u odběratele	12	Pořadové číslo objednávky, převzato z objednávky	Ano
číslo objednávky u dodavatele	12	Důležité pro zpětné dotazy s velkoobchodem	Ano
datum	8	Datum vystavení defektního listu	Ano
status	3	Zpracování na straně velkoobchodu	Ne
datum dodání	12	požadované datum (popř. datum a čas) dodání - převzato z objednávky	podle objednávky
místo dodání	20	převzato z objednávky	podle objednávky
druh objednávky	1	0 nebo neuvedeno=běžná, 5=transferová - převzato z objednávky	podle objednávky

transfer firma	16	kód firmy která zadala transferovou objednávku, povinné u transferové objednávky - převzato z objednávky	podle objednávky
transfer zastupce	16	kód zástupce který zadal transferovou objednávku, povinné u transferové objednávky - převzato z objednávky	podle objednávky
id akce	16	Identifikační kód akce (transferu) - převzato z objednávky	podle objednávky

položky:

druh kódu	1	Převzato z objednávky	Ano
kód	16	Převzato z objednávky	Ano
nepotvrzené množství	12,2		Ano
kód defektu	3		Ne
text defektu	50	Povinné při vyplněném kódu defektu	viz vlevo
druh kódu 2	1	Podmíněně povinné	viz vlevo
kód_2	16	Podmíněně povinné	viz vlevo

textová část:

Textová část defektního listu je na začátku řádku označena slovem "TEXT". Počínaje další řádkou do konce souboru následuje libovolný text (informace dodavatele).

druh kódu:

- 0 APA
- 1 SÚKL - pozor 7 znaků
- 2 EAN
- 3 PDK
- 4 VZP PZT
- 8 čárový kód
- 9 kód dodavatele

Zboží s neurčeným druhem kódu nebude při objednávce akceptováno.

Status / text statusů:

001 Potvrzeno - vždy, když proběhne zpracování u dodavatele a odběratel je existující a nezablokovaný, potom v položkách defektního listu mohou být kódy defektu dle tabulky, přičemž i celá objednávka může být defektní.

002 Zablokovaný odběratel – všechny položky objednávky jsou uvedeny v defektním listu a mají kód defektu 002.
--

003 Neznámý odběratel – všechny položky objednávky jsou uvedeny v defektním listu a mají kód defektu 001.
--

Kód / text defektů:

001 Vadné číslo odběratele
002 Tento odběratel je blokován
003 Tento výrobek neexistuje
004 Tento výrobek není v prodejním skladě zaveden
005 Tento výrobek je pro prodej blokován
006 Tento výrobek není k dispozici
007 Tento výrobek není na skladě v dostatečném množství
008 Nebylo přiřazeno číslo defektu
009 Není k dispozici na této pobočce - objednejte na jiné pobočce
010 Není k dispozici - objednejte v centrálním skladu
011 Výrobek byl vyřazen z registrace
012 Krátkodobý výpadek dodávek
013 Dlouhodobý výpadek dodávek
015 Není v číselníku PDK
016 Položku nelze určit z důvodů duplicity objednáčích kódů
017 Nekorektní množství
020 Neplatný druh objednávky
021 Neplatný kód transferové akce
022 Neplatný kód zadavatele transferové objednávky
061 Tento výrobek dodavatel nevyrábí
062 Tento výrobek je výběhový / doprodej
063 Tento výrobek je na zakázku - objednávku uplatněte telefonicky
064 Tento výrobek je privátní sortiment
099 Tento výrobek má nový kód
100 Nabídka náhradního přípravku (tento řádek se může vyskytnout i vícekrát k jedné poptávané položce pokud je nabízeno více náhrad). V textu defektu je název nabízené náhrady, v polích DRUH KODU 2 a KOD_2 je uveden PDK nebo Apa kód náhrady. Nenahrazuje samotné odmítnutí, to musí být uvedeno před tímto defektem.

Defektní list pro testovací objednávku nemusí obsahovat pravdivé informace ohledně potvrzeného nebo nepotvrzeného množství, vždy však musí obsahovat korektní informace týkající se kódů.

U testovací i ostré objednávky distributor v defektním listu odpoví vždy kódem dle objednávky.

Pokud kódem v objednávce nebude kód PDK, bude kód PDK uveden v položce KOD_2.

U přípravku, u něhož není potvrzován defekt a není pro komunikaci použit kód PDK, je nutno uvést defektní řádek s nulovým nepotvrzeným množstvím a s opravným kódem PDK. Kód defektu v tomto případě bude 099 – Tento výrobek má nový kód.

Zpracování odpovědi na elektronickou objednávku na straně lékárenského SW musí být schopno oba způsoby (jak u defektní, tak u nedefektní položky) opravy resp. doplnění kódu PDK v lékárně zachytit a zaznamenat tak, aby byl takto získaný kód PDK použitelný při dalším objednávání.

Pokud budou obě strany (SW na straně lékárně i distributora) fungovat popsaným způsobem, provede se napárování aktivního sortimentu v lékárně na číselník PDK pouhým odesláním testovacích objednávek na všechny zapojené distributory, neboť na všechny dodavatelské kódy bude odpovězeno současně i kódem PDK.

Příklad souboru :

Defektní list : 0005541.def

```
6|160200||010-45316490|0005541|0012589|200605101530||20060512|||||
1|0051621|50.00|007| Tento výrobek není na skladě v dostatečném množství|3|4013054001622|
3|0071499|10.00|007| Tento výrobek není na skladě v dostatečném množství|||
3|6895873000126|5.00|007| Tento výrobek není na skladě v dostatečném množství|||
4|32521|3.00|007| Tento výrobek není na skladě v dostatečném množství|3|4122629|
1|0000011|0.00|099| Tento výrobek má nový kód|3| 8584005128706|
1|0000105|5.00|011| Produkt vyřazen z registrace|3|0001058|
1|0000105|5.00|100| Dexamethazon tbl.20x0.5mg(blistr) Léčiva|3|8594739018167|
```

TEXT

Nabídka dne:

Dodací list

Název souboru obsahuje číslo objednávky lékárně nebo dodacího listu s příponou ".DOD". Pole označené jako nepovinné není nutné uvádět.

hlavička:

Číslo verze	3	Číslo verze formátu dodacího listu	Ano
kód dodavatele	16		Ano

číslo objednávky u odběratele	12	Převzato z objednávky	Ano
Číslo dodacího listu	12	Číslo dodacího listu dodavatele	Ano
Datum	8		Ano
IČO	8	IČO odběratele	Ano
Počet položek dodacího listu	8	Počet řádků zboží	Ano
Součet ve všech sazbách bez DPH	10,2	Celková cena bez DPH	Ano
Součet ve všech sazbách s DPH	10,2	Celková cena včetně DPH	Ano
Součet ve snížené sazbě bez DPH	10,2		Ano
Součet ve snížené sazbě vč. DPH	10,2		Ano
Součet v základní sazbě bez DPH	10,2		Ano
Součet v základní sazbě vč. DPH	10,2		Ano
datum dodání	12	požadované datum (datum a čas) dodání - převzato z objednávky	podle objednávky
místo dodání	20	převzato z objednávky	podle objednávky
druh objednávky	1	0 nebo neuvedeno=běžná, 5=transferová - převzato z objednávky	podle objednávky
transfer firma	16	kód firmy která zadala transferovou objednávku, povinné u transferové objednávky - převzato z objednávky	podle objednávky
transfer zastupce	16	kód zástupce který zadal transferovou objednávku, povinné u transferové objednávky - převzato z objednávky	podle objednávky
id akce	16	Identifikační kód akce (transferu) - převzato z objednávky	podle objednávky

položky:

kód zboží	16	Kód zboží PDK	Ano
-----------	----	---------------	------------

Množství	12,2	Dodané (potvrzené) množství	Ano
cena výrobce	12,2	Povinný u regulovaných přípravků, všechny ceny jsou za MJ	viz vlevo
nákupní cena bez DPH	12,2		Ano
nákupní cena s DPH	12,2		Ano
DPH v procentech	4,1		Ano
Prodejní cena	12,2	Doporučená prodejní cena lékárny s DPH	Ne
Sarže	20		Ne
Exspirace	8		Ne
kód APA	7	Podmíněně povinný viz poznámku níže.	viz vlevo
Název	50		Ne
Čárový kód	16		Ne

textová část:

Textová část dodacího listu je na začátku řádku označená slovem "TEXT". Počínaje další řádkou do konce souboru následuje libovolný text (informace odběrateli). Na začátku této oblasti je možno uvádět i certifikáty surovin v následujícím tvaru, a to i vícenásobně každý na zvláštním řádku:

CERTIF_SUR|PDK kód suroviny|certifikát|

Vyhledem k velmi malému podílu surovin v oběhu zboží není na toto zavedeno zvláštní pole ve struktuře dodacího listu.

Na dodacím listu je u položek použit pouze kód PDK. Aby byly odstraněny problémy s časovou synchronizací upgrade číselníku PDK v lékárnách a u distributora, je doplněn polem Kód_APA, které musí být vyplněno, je-li v kódu PDK uveden kód EAN. Tímto způsobem obdrží lékárna oba kódy, které mohou být v lékárně v položce PDK uvedeny.

Nepovinným polem je název přípravku, který má usnadnit dohledávání nespárovaných přípravků a zakládání nových přípravků, přijatých v lékárně na základě telefonických objednávek.

Příklad souboru**Dodacího list : 0005612.dod**

4|010-5316490|0005541|0005612|20060512|12345678|3|5029.20|5960.35|1029.20|1080.35| 4000.00| 4880.00|20060512|||||

4013054001622|150.00|6.00|6.64|6.97|5.0|8.60|||0118332|||

6905218880090|5.00|6.00|6.64|6.97|5.0|8.60|||0142621|Essentiale balzam 3.5g|

4122629|4.0||1000.00|1220.00|22.0|1375.00|S123|20010624|||||

8595142303703|2.00||189.90|225.98|19.0||8524561||0698457|Glycerolum 85% 1200g Dr.Kulich|

8590724313125|8.00||86.20|102.58|19.0||45600123||0342205|Ethacridini lactas monohydricus ČL 2002 100g Tamda|

TEXT

CERTIF_SUR|8595142303703|01548/0805/158|

CERTIF_SUR|8590724313125|00453/0705/124|

Katalog bude v další dodávce.

Rekapitulace dodacích listů a vratek

"Rekapitulace..." je úplný výčet dodacích listů a vratek spadajících pod jednu fakturu. Součtové položky sčítají odpovídající údaje jednotlivých dodacích listů.

Název souboru obsahuje číslo této faktury nebo jeho zkrácení (proměnnou část) a příponu ".SBD". Pole označené jako nepovinné není nutné uvádět. Finanční údaje u vratky jsou uváděny záporně. Věta hlavičky má 27 polí, věta dodacího listu (popř. vratky) má 10 polí.

hlavička:

číslo verze	3	číslo verze formátu sběrného dokladu	Ano
Pharmdata kód odběratele (lékárny)	7	pokud existuje má prioritu před následujícím údajem	Ne
kód odběratele	16	kód, pod kterým je odběratel registrován u dodavatele	Ano pokud není předchozí údaj
kód dodavatele	16	kód, pod kterým je dodavatel registrován u odběratele (např. IČO, DIČ)	Ano
středisko odběratele	10	interní upřesňující údaj o odběrateli	NE
středisko dodavatele	10	interní upřesňující údaj o dodavateli	NE
číslo faktury	16		Ano
datum vystavení faktury	8		Ano
datum zdanitelného plnění faktury	8		Ano
datum splatnosti faktury	8		Ano
variabilní symbol faktury	16		Ano
specifický symbol faktury	16		NE
konstantní symbol faktury	4		NE

ISO kód měny	3	CZK pro Kč	Ano
číslo bankovního účtu dodavatele	16	pro tuzemskou banku (holé číslo účtu bez kódu banky)	Ano pro tuzemskou banku
kód banky dodavatele	16	4 místný numerický kód - jen pro tuzemskou banku	Ano pro tuzemskou banku
IBAN dodavatele	30		Ano pro zahr.banku
SWIFT kód dodavatele	11		Ano pro zahr.banku
procento základní sazby dph	4,1		Ano
Součet základů daně v základní sazbě	12,2	může být i záporná je-li faktura dobropis	Ano
Součet DPH v základní sazbě	12,2	může být i záporná je-li faktura dobropis	Ano
procento snížené sazby DPH	4,1		Ano
Součet základů daně ve snížené sazbě	12,2	může být i záporná je-li faktura dobropis	Ano
Součet DPH v snížené sazbě	12,2	může být i záporná je-li faktura dobropis	Ano
Částka osvobozená od DPH	10,2	může být i záporná je-li faktura dobropis	Ano
Součet základů a daně ve všech sazbách celkem	12,2	může být i záporná je-li faktura dobropis	Ano
počet dodacích listů	3	počet dodacích listů ve fakturě	Ano
počet vratek	3	počet vratek (reklamací) ve fakturě	Ano
položky:			
druh dokladu	1	D znamená dodací list, "V" vratku	Ano
číslo dokladu od dodavatele	16	číslo dodacího listu nebo vratky vydané dodavatelem	Ano

číslo dokladu od odběratele	16	číslo objednávky v případě dodacího listu číslo vratky (reklamací) vydané odběratelem	Ano
datum vystavení	8	datum vystavení dodacího listu nebo vratky u dodavatele	Ano
Součet ve všech sazbách bez DPH	10,2	Celková cena bez DPH - identický údaj jako na formátu dodacím listu - v případě vratky záporné	Ano
Součet ve všech sazbách s DPH	10,2	Celková cena včetně DPH - identický údaj jako na formátu dodacím listu - v případě vratky záporné	Ano
Součet ve snížené sazbě bez DPH	10,2	identický údaj jako na formátu dodacím listu - v případě vratky záporné	Ano
Součet ve snížené sazbě vč. DPH	10,2	identický údaj jako na formátu dodacím listu - v případě vratky záporné	Ano
Součet v základní sazbě bez DPH	10,2	identický údaj jako na formátu dodacím listu - v případě vratky záporné	Ano
Součet v základní sazbě vč. DPH	10,2	identický údaj jako na formátu dodacím listu - v případě vratky záporné	Ano

Příklad souboru sběrného dokladu **50015489.SBD** (první řádek zde uměle zalomen):

```
6|1600200|51480205|02|0010|1250015489|20050905|20050831|20051005|1250015489|0008|CZK|150054756|0300||19.0|10000.00|1900.00|5.0
|20005.00|1000.25|0.00|32905.00|3|1|
D|50211234|512008|20050822|15000.00|16450.00|10000.00|10500.00|5000.00|5950.00|
D|50211354|512018|20050825|11000.00|12250.00|6000.00|6300.00|5000.00|5950.00|
V|65001368|35018|20050827|-1000.00|-1050.00|-1000.00|-1050.00|0.00|0.00|
D|50211458|513108|20050828|5005.00|5255.25|5005.00|5255.25|0.00|0.00|
```