

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: M 2612 – Elektrotechnika a informatika

Studijní obor: 3902T005 – Automatické řízení a inženýrská
informatika

Generátor šablon obecného internetového obchodu

Common online store templates generator

Diplomová práce

Autor: **Lukáš Veselý**

Vedoucí DP práce: Mgr. Jiří Vraný

Konzultant:

V Liberci 30. 6. 2006

Originál zadání

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/200 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že souhlasím s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum: 30. června 2006

Podpis:

Poděkování

Je mojí milou povinností poděkovat vedoucímu diplomové práce, panu Mgr. Jiřímu Vranému, za vstřícnost při tvorbě zadání a trpělivost s mojí osobou při dlouhém vývoji projektu.

Dále patří můj dík přátelům Ondřeji Čechovi, Markovi Duškovi a Josefu Pajskrovi za neocenitelné rady při vývoji webové aplikace.

Abstrakt

Internetový obchod je v dnešní době již nedílnou součástí komerční sféry. Firmy si nechávají vytvářet vlastní internetové obchody, a proto by bylo výhodné mít nástroj pro tvorbu různých obchodů tak, aby se dali použít pro široké spektrum zboží.

V první řadě je potřeba se seznámit s aktuálními trendy z této oblasti a zvládnout všechny nezbytné technologie pro vývoj internetových obchodů. To znamená zvládnout XHTML, CSS, jazyk PHP a v neposlední řadě jazyk relačních databází SQL. Následuje realizace projektu, což obnáší vytvoření vlastního obecného internetového obchodu a aplikaci, která bude tento obchod generovat a připravit ho k provozu.

Před vlastním vývojem obchodu bylo třeba kvalitně navrhnout databázi pro ukládání dat tak, aby se snadno spravovala a byla flexibilní. K tomuto účelu jsem použil pravidla normalizace databáze. Tvorbu vzhledu internetového obchodu jsem provedl pomocí kaskádních stylů CSS, struktura dokumentů je tvořena pomocí XHTML a logická část je napsána jazykem PHP. Přístup k datům zajišťuje jazyk PHP spolu s SQL. Obecnost obchodu je zajištěna použitím datatypů, které umožňují popis libovolného zboží. Vyvinul jsem také jednoduchý HTML editor, který umístí do obchodu HTML dokumenty zprostředkovávající informace zákazníkům. Posledním krokem byla tvorba aplikace, která umožňuje vygenerovat různé vizuální varianty obchodu a umístí je na webový server. Tuto aplikaci může používat i člověk neznalý všech výše uvedených technologií.

Cíle stanovené v zadání označuji za splněné, úspěšně jsem splnil všechny cíle. Největším problémem při práci se ukázala tvorba vzhledu obchodu, neboť různé prohlížeče interpretují CSS jinak. Zejména Internet Explorer zpracovává CSS chybně. Diskutovat lze také vhodnost obecného řešení oproti řešení konkrétnímu. Konkrétní řešení bude vždy výhodnější pro daný druh zboží, obecné řešení je zase možné použít beze změn téměř všude.

Abstract

Online business became a stable part of the commercial domain over the last few years. Companies are hiring web developers to create online stores for

their business, so it is profitable to have a tool for creating various online stores, which would offer different types of wares.

Firstly, it is necessary to become familiar with current trends from this area and master all needed technologies for web applications development. This means learning XHTML, CSS, PHP language, relational databases language SQL and even more. Next step is the creation of the applications, which includes development of the common online store and an application that will generate such an online store and make it ready to use right away.

Before the development could start, a very good concept of the database for storing data was needed, so the database is flexible and easy to maintain. I used normalization rules to achieve this task. The design part of the store is made by cascading style sheets CSS, documents structure by XHTML and the logical part is written in PHP. Access to data is realized using PHP and SQL languages. The application can be used for stores for any type of business. This is ensured by the use of so called datatypes. I have developed an easy HTML editor that will place HTML documents right on to the store's website, providing customers with useful information. The last step was developing an application that generates different visual interfaces of the website and places them on a web server. This application can be used by a person unskilled in all technologies described above.

I consider all goals of my diploma thesis successfully completed. The biggest problems were caused by different interpretations of CSS by various web browsers, especially Internet Explorer which often fails to process CSS correctly. The program I have created can be used by many types of online stores, however a specific design will always suit a specific business clients needs better than a template designed to be used by many.

Obsah

Originál zadání	2
Prohlášení	3
Poděkování	4
Abstrakt	5
Abstract	5
Obsah	7
Úvod	9
1 Teoretické základy práce	10
1.1 Interakce webového serveru a klienta	10
1.2 Databázové systémy	10
1.2.1 Implementační datové modely	11
1.2.2 Normalizace databáze	11
1.2.3 Normální formy	12
1.2.4 Výhody a nevýhody normalizace	13
1.3 Technologie a postupy tvorby webových systémů	13
1.3.1 HTML	13
1.3.2 XML	14
1.3.3 XHTML	14
1.3.4 CSS	14
1.3.5 PHP	14
1.3.6 SQL	15
1.3.7 Web design	15
2 Prostředky k realizaci projektu	16
2.1 Diskuze webového serveru	16
2.2 Výběr databázového serveru	16
2.3 Programové nástroje pro realizaci projektu	16
2.3.1 Delphi	16
2.3.2 RichView komponenty pro Delphi	17
2.3.3 PSPad	17
2.3.4 CASE Studio	17
2.3.5 MySQL-Front	18
2.3.6 Rozšíření PHP PEAR	18
3 Realizace projektu	18
3.1 Definice rozsahu	18
3.2 Analýza projektu	19
3.2.1 Funkční analýza	19
3.2.2 Výsledky funkční analýzy	21
3.2.3 Datová analýza	23
3.2.4 Výsledky datové analýzy	24
3.3 Systémové zázemí	26
3.4 Pravidla vývoje obchodu	26
3.5 Zajištění obecnosti obchodu – datatypy	27
3.6 Grafický návrh internetového obchodu	28
3.7 Uživatelská část	31
3.7.1 Registrace zákazníka a změna údajů	31
3.7.2 Přihlášení do systému	33
3.7.3 Horní menu informací obchodu	33

3.7.4	Kategorie produktů	34
3.7.5	Vyhledávání produktů	35
3.7.6	Seznam produktů	35
3.7.7	Detail produktu	36
3.7.8	Nákupní košík.....	37
3.7.9	Vytvoření objednávky	37
3.8	Část správce systému	38
3.8.1	Přihlášení a změna hesla	38
3.8.2	Správa horního menu informací	39
3.8.3	Správa kategorií produktů	41
3.8.4	Správa produktů	42
3.8.5	Objednávky	43
3.8.6	Seznam zákazníků	45
3.8.7	Ostatní nástroje	45
3.9	Generátor internetového obchodu.....	46
3.9.1	Design	46
3.9.2	Nastavení	47
3.9.3	Dokončení tvorby obchodu.....	48
3.10	Práce s databází pomocí SQL a PHP	49
Závěr		50
Seznam literatury		51

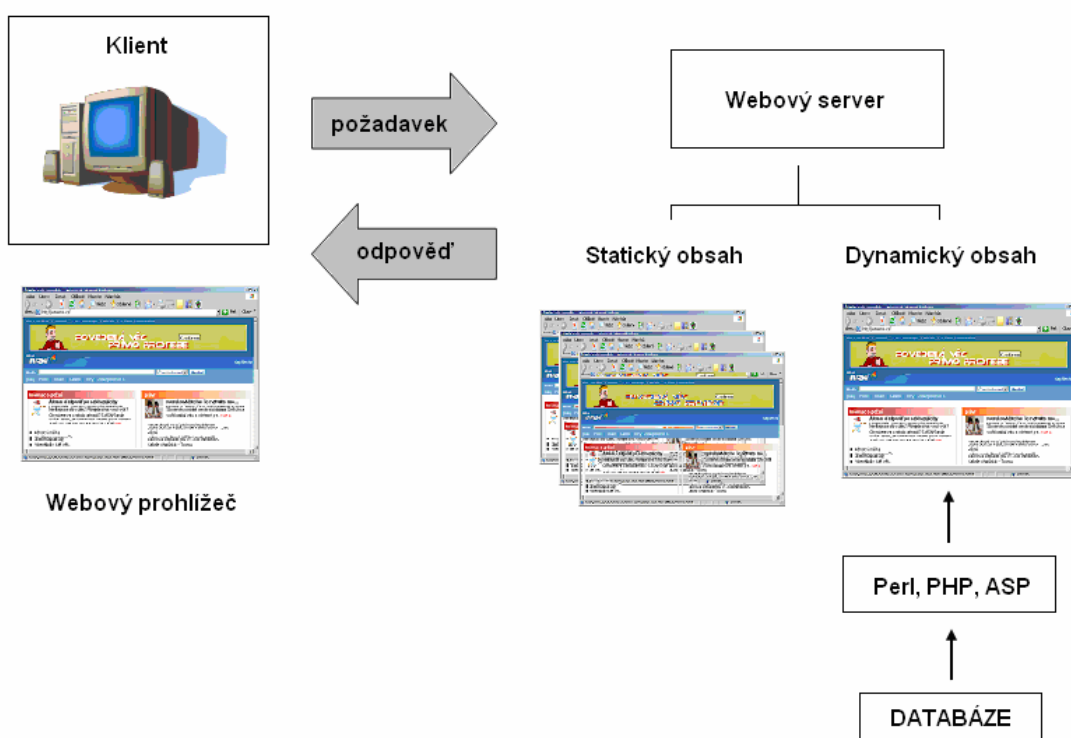
Úvod

O internetovém obchodování v dnešní době již slyšel zřejmě každý a velké množství lidí se s ním setkalo v praxi. Počet osob využívajících pravidelně možnosti nákupu zboží v internetovém obchodě neustále roste. Internetový obchod má jistě budoucnost, a proto jsem vymyslel návrh zadání vytvořit internetový obchod jako diplomovou práci. Můj návrh neobsahoval pouze vytvoření obchodu, ale stanovil jsem si za úkol vytvořit takový internetový obchod, který bude možno nasadit na široké spektrum sortimentu. Součástí projektu je navíc úkol naprogramovat v některém vyšším programovacím jazyce aplikaci, která umožní vygenerovat obchod v různých vizuálních provedeních a připravit ho k okamžitému provozu. Tato aplikace tedy umožní uživateli, který neovládá žádné technologie vytváření webových stránek, vytvořit si vlastní internetový obchod dle svých představ a umístit ho na webový server. Pro realizaci tohoto projektu bylo potřeba se dostatečně seznámit s technologiemi vývoje webových aplikací, což zahrnuje XML, XHTML, CSS, a technologiemi umožňující využití databází, jako jsou PHP, SQL a další.

1 Teoretické základy práce

1.1 Interakce webového serveru a klienta

Webový server, který je připojen k počítačové síti přijímá od klientského počítače požadavky ve tvaru HTTP a následovně mu vrací odpověď v podobě HTML dokumentu, či dokumentu jiného formátu (obrázek, text). Server může odesílat buď tzv. statický obsah v podobě neměnného datového souboru, nebo tzv. dynamický obsah, kdy se obsah mění podle klientových požadavků za využití databází a skriptovacích jazyků (např. PHP), které se spouštějí na straně serveru ještě před stažením z Internetu. Skriptovací jazyky jako JavaScript se naopak spouští na straně klienta.



Obr. 1 Vztah mezi klientem a serverem

1.2 Databázové systémy

Databáze slouží k uspořádávání dat dle různých kritérií, podle kterých je můžeme potom třídit, vyzvedávat, či měnit. Databázový systém je tvořen dvěma částmi, tzv. bází dat, která představuje souhrn potřebných informací, a programem SŘBD (Systém Řzení Baze Dat, Data Base Management System,

DBMS), který data spravuje (MySQL, Oracle, MS Access a další). SŘBD má prostředky k popisu dat, jejich definici a hlavně k vytvoření vlastní databáze. K vytvoření a definici databáze slouží jazyk typu DDL (Data Definition Language). Přístup k již existujícím databázím umožňuje jazyk typu DML (Data Manipulation Language). Data musí odpovídat vlastnostem popisovaného objektu, což se nazývá integrita dat a systém musí zajistit, že nebude nikdy porušena. Díky SŘBD se programátor nemusí starat o organizaci dat v bázi a přistupuje k nim pomocí jazyka strukturovaných dotazů SQL (Structured Query Language).

1.2.1 Implementační datové modely

Existuje několik datových modelů. Hierarchický, ve kterém se pracuje s daty ve stromové struktuře a jsou v něm realizovány vztahy 1:N. Síťový je jeho variací a lze zde vyjadřovat i vztahy M:N, jeho realizace je však náročnější. Oba tyto modely umožňují vysokou efektivitu zpracování, nevýhoda spočívá v nesnadnosti změnit strukturu stromů a jejich vazby.

Tento problém řeší relační model. Relační model je sestavený z řady tabulek a jejich sloupce se váží na sloupce v jiných tabulkách. Vznikne tak závislost mezi propojenými poli, což nám umožňuje provádět dotazy. Toto je bohužel na úkor snížení efektivity zpracování databáze.

Objektově orientovaný model dat nabízí lepší popis objektů reálného světa. Vedle vlastností využívá i principů objektově orientovaného programování (OOP), pracuje s metodami, třídami, děděním.

1.2.2 Normalizace databáze

Normalizace je proces uspořádání dat v databázi. To znamená vytváření tabulek a definování vztahů mezi tabulkami na základě pravidel, která jsou stanovena tak, aby zabezpečovala data a vytvořila flexibilnější databázi. Toho je dosaženo zejména eliminací dvou faktorů: redundantních dat a funkčních nezávislostí na primárním klíči.

Redundantní data zbytečně zabírají diskovou kapacitu a působí problémy při správě databáze. Jestliže potřebujeme změnit data, která existují na více než jednom místě, pak tato data musejí být změněna vždy přesně stejným způsobem ve všech lokacích.

Funkční nezávislost dat na primárním klíči si můžeme například představit takto: mějme tabulku zákazníků a v ní atribut adresa zákazníka. Tento atribut je funkčně závislý na daném zákazníkovi. Pokud ale bude v tabulce zákazníků plat zákazníkova obchodníka, pak tento atribut není funkčně závislý na zákazníkovi. Tento atribut patří do tabulky s obchodníky a proto by tam měl být přesunut. Funkční nezávislost může způsobit problémy při přístupu k datům, jelikož cesta k datům může být neúplná, nebo úplně chybět.

1.2.3 Normální formy

Existuje několik pravidel pro normalizaci databáze, která se nazývají normální formy. Základní normální formy jsou tři – první normální forma, druhá normální forma a třetí normální forma. Pokud jsou splněny podmínky pro první normální formu, pak říkáme, že databáze je v první normální formě a obdobně se zbylými formami. Existují další stupně normalizace, ale pro většinu aplikací si vystačíme s těmito třemi základními:

První normální forma:

- Eliminuje opakující se atributy v tabulkách.
- Množiny příbuzných dat rozděluje do samostatných tabulek.
- Každou množinu příbuzných dat identifikuje primárním klíčem.

Druhá normální forma:

- Pro množiny hodnot, které se vyskytují ve více záznamech vytvoří oddělené tabulky.
- Tyto tabulky pak prováže se záznamy pomocí cizích klíčů.

Třetí normální forma:

- Přesune atributy, které nejsou závislé na primárním klíči do samostatných tabulek.

Databáze v třetí normální formě nemusí být vždy nejpraktičtějším řešením. Pokud máme kupříkladu tabulku zákazníků s jejich adresami, pak bychom museli pro město a PSČ definovat vlastní tabulku, neboť více zákazníků může bydlet ve stejném městě. Problémem je velmi velký počet tabulek, které mohou neúměrně zatěžovat systém.

1.2.4 Výhody a nevýhody normalizace

Mezi nejvýznamnější výhody normalizace patří:

- lepší struktura databáze
- redukce nadbytečných dat
- konzistence dat v databázi
- větší flexibilita databáze
- lepší zabezpečení databáze

Normalizace má bohužel i své stinné stránky. Největší nevýhodou normalizované databáze může být podstatné snížení výkonu z důvodu velkého množství tabulek. Každé zaslání dotazu totiž představuje pro systém určitou zátěž a pro získání potřebných informací z databáze je často třeba vybírat data z několika tabulek.

1.3 Technologie a postupy tvorby webových systémů

1.3.1 HTML

HTML (HyperText Markup Language) je značkovací jazyk se svojí přesnou syntaxí a vlastními značkami (tagy), které jsou většinou párové a určují vzhled textu, tabulek, obrázků a podobně. HTML používá princip hypertextu, tedy propojování více dokumentů pomocí odkazů. Textový soubor obsahující HTML využívá celé osmibitové ASCII, tedy včetně znaků národních abeced. Má příponou *.html nebo *.htm, může být napsán v jednoduchém textovém editoru jako je např. Poznámkový blok, ale pro zjednodušení práce se používají editory jako Homesite či PSPad, které umožňují funkci náhledu (WYSIWYG – What You See Is What You Get). Program, který tento text dekóduje a interpretuje jej do podoby webové stránky se nazývá internetový prohlížeč. Standardně s operačním systémem Windows je dodáván Internet Explorer, ale pro některé své nedostatky se používají i jiné, volně dostupné, prohlížeče (Firefox, Netscape, Opera). Korektnost tagů je důležitá pro správné zobrazení v každém prohlížeči a lze ji zkontrolovat tzv. validováním.



Obr. 2 Loga prohlížečů

1.3.2 XML

XML (EXtensible Markup Language) je stejně jako HTML značkovací jazyk, ale nemá vlastní značky. Nepracuje s formou dokumentu, ale s jejím obsahem. Jedná se o univerzální formát pro ukládání dat bez problému přenositelnosti mezi mnoha aplikacemi různých druhů na různých operačních systémech. K popisu dat používá DTD (Document Type Definition), nebo XML Schéma.

1.3.3 XHTML

XHTML (EXtensible HyperText Markup Language) rozšiřuje HTML jazyk, odstraňuje jeho nedostatky a má za cíl jej v budoucnu plně nahradit. Je také mnohem více striktní co se validity kódu týče a přehlednější. XHTML je vlastně HTML definované jako XML aplikace.

1.3.4 CSS

Kaskádní styly (Cascading Style Sheets) definují jak zobrazit HTML elementy a umožňují zároveň graficky mnohem více, než samotné HTML, a v dnešní době jej prakticky nahrazují. Obvykle se ukládají do samostatného textového souboru s příponou *.css a připojují se do hlavičky HTML dokumentu. Stránka má tak oddělen svůj vzhled a strukturu, což při pozdějších úpravách designu stránky umožňuje, že stačí změnit pouze údaj v souboru se styly. CSS usnadňují údržbu grafiky webové stránky, neboť jsou mnohem flexibilnější. Jejich další výhodou je i větší kompatibilita s webovými prohlížeči. Kaskádováním se myslí, že může být zadáno více stylů pro jeden text, přičemž nenastává konflikt, protože každý styl má jinou prioritu, kterou určují pravidla kaskádování. Vedle formátování HTML stránky lze použít i k formátování XML dokumentů.

1.3.5 PHP

Skriptovací jazyk PHP (Hypertext Preprocessor, původně Personal Home Page) je určený pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, takže skripty jsou pak prováděny na straně serveru. Jeho syntaxe je kombinací několika

programovacích jazyků (Perl, C, Pascal a Java). PHP je nezávislé na platformě a obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k několika databázovým serverům a podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP, ...).

1.3.6 SQL

Tento jazyk je jedním z dotazovacích jazyků a umožňuje formulovat požadavky uživatelů při vybírání informací z databáze. Neurčuje přesně jak se má úkon provést, pouze popisuje co se má provést a jaká data jsou potřeba, patří tedy mezi neprocedurální jazyky. Na rozdíl od ostatních jazyků je SQL tvořeno pouze několika výkonnými příkazy, které umožňují data definovat a aktualizovat.

1.3.7 Web design

Ovládání jakéhokoliv webu, nejenom internetového obchodu, by mělo být pro uživatele intuitivní, a proto je třeba dodržovat několik zásad. Právě především u internetových obchodů je otázka designu na místě, obchodník chce své produkty prodat, potřebuje zákazníka nějakým způsobem nalákat a přesvědčit ho, proč nakupovat právě u něj. Význam prvků obchodu by měl být zřejmý na první pohled. Zákazník nesmí být odrazen mnoha rušivými efekty (např. velké blikající bannery) nebo celkovou nepřehledností, která může způsobit, že potenciálnímu zákazníkovi dojde trpělivost a raději vyhledá jiný obchod.

Panel s navigací musí být jasně viditelný a přehledný, barevná sladěnost nesmí působit rušivě, ale ani nudně, měla by uživatele zaujmout a proto se do určité míry může objevit i lehká disharmonie. Dbáme též na zvyklosti co se týče umístění panelů a názvů některých položek, což zákazníkovi velmi usnadní orientaci v našem obchodu. Pokud si neví rady, mělo by mu být co nejrychleji umožněno použít vyhledávání. Důležitá je rovněž stručnost, která však nesmí způsobit nesrozumitelnost. Pokud je struktura nabízených výrobků široká, řadíme ji do vhodně zvolených a pojmenovaných kategorií a dbáme, aby byl zákazník stále informován o tom, v které části stromu se nachází a měl možnost okamžitého opuštění kategorie či přeskokování na nadřazenější

kategorie. U loga se může objevit krátký a výstižný slogan, který usnadní uložení informace o obchodě do zákaznickovy paměti.

2 Prostředky k realizaci projektu

2.1 Diskuze webového serveru

Pro vývoj své diplomové práce jsem vybral webový server Apache (verze 1.3) a to hned z několika důvodů. Apache server je open source aplikace a je tedy zdarma. Má velmi dobrou podporu jazyka PHP a databázového serveru MySQL, je do velké míry bezpečný a má výbornou dokumentaci.

2.2 Výběr databázového serveru

Jelikož jsem pracoval s webovým serverem Apache, bylo téměř povinností zvolit za databázový systém MySQL, který s Apachem velmi dobře spolupracuje. MySQL je navíc pro nekomerční účely zdarma, je podporován většinou operačních systémů, má krátké doby odezvy, podporu českého jazyka a mnoho dalších kvalit.

2.3 Programové nástroje pro realizaci projektu

2.3.1 Delphi

Delphi jsou integrované grafické vývojové prostředí určené pro tvorbu aplikací v jazyce Object Pascal (objektové nastavbě Pascal). Obsahuje systém RAD (Rapid Application Development) umožňující vizuální návrh grafického uživatelského rozhraní, na jehož základě se automaticky vytváří kostra zdrojového kódu, což výrazně urychluje vývoj aplikace. Programování v Delphi je tak z velké části založeno na použití komponent, což je malý program, který vykonává určitou činnost – například zobrazuje text, komunikuje s databází atd. Jejich předností oproti některým konkurenčním produktům jsou knihovny komponent, které jsou dodávány přímo s Delphi, případně se dají stáhnout z internetu, nebo si lze vytvořit své vlastní.

2.3.2 RichView komponenty pro Delphi

Tyto shareware komponenty pro vývojové prostředí Delphi slouží k prohlížení, editaci a tisky hypertextových dokumentů. V dokumentech je možno formátovat text, vkládat obrázky, tabulky, či seznamy.

2.3.3 PSPad

Tento volně šiřitelný editor je určený programátorům nejrůznějších programovacích jazyků. Program kombinuje dohromady jednoduchost, přehlednost a rychlost psaní kódu.

Mezi jeho užitečné vlastností vlastnosti patří:

- práce se soubory v různých formátech a kódováních
- automaticky barevně zvýrazňuje syntaxi
- pracuje s více projekty i s více dokumenty současně
- obsahuje konverzi češtiny, HEXA editor, editor TopStyle Lite (pro editaci CSS)
- umožňuje export včetně zvýraznění RTF, HTML, XHTML a dalších
- obsahuje šablony pro HTML, PHP, JavaScript, MySQL a další
- náhled v IE nebo v Mozille

2.3.4 CASE Studio

Profesionální software, díky němuž lze omezit chyby ještě před započítím vlastní psaní kódu, umožňuje vizuálně navrhovat relační diagramy databázových struktur a nabízí:

- přehledné zobrazení ER diagramů (Entity Relationship Diagrams)
- podporu pro různé databáze
- převod fyzických struktur databáze na SQL skripty
- zpětné načtení struktury (reverse engineering), kdy se vytvoří model struktury z databáze
- definované šablony
- generování detailní HTML a RTF dokumentace, export do XML formátu a další

2.3.5 MySQL-Front

MySQL-Front je grafický nástroj pro správu MySQL databází. Jeho výhodou oproti nástrojům postaveným na PHP a HTML je jeho rychlost, neboť nemusí načítat HTML stránky, jde o klasickou aplikaci.

MySQL-Front zobrazuje data přehledně v tabulkách s proměnnou šířkou. Velkou předností této aplikace je její jednoduchost. Vše je dostupné po pouze několika kliknutích myši, což činí správu databáze, importování a editování dat velmi jednoduchým. Program má navíc vestavěný SQL editor pro přímé vkládání SQL příkazů.

2.3.6 Rozšíření PHP PEAR

PEAR je zkratka pro PHP Extension and Application Repository, což znamená Rozšíření PHP a Zdroj Aplikací. Jedná se o open source knihovnu hotových funkčních celků pro jazyk PHP.

3 Realizace projektu

3.1 Definice rozsahu

Podrobná definice problémů a požadavků na systém je mnohem důležitější, než se na první pohled může zdát. Při podcenění této fáze vývoje aplikace může později dojít k chybám, které se velmi těžko odstraňují.

Předtím, než je možné vytvořit aplikaci generující internetový obchod, je zřejmě třeba naprogramovat samotný obecný internetový obchod. To obnáší v první řadě kvalitní návrh databáze pro ukládání dat obchodu a to tak, aby se dala ukládat strukturně různá data. Následně je třeba vypracovat zákaznickou a správcovskou část systému. Zákaznická část představuje vlastní obchod, kde je nabízeno zboží a toto zboží zákazník nakupuje. Tato část by měla být co nejvíce intuitivní a jednoduchá, neboť zákazník, který se nebude v obchodě orientovat, zřejmě rychle odejde. Část správce obchodu dává správci kontrolu nad strukturou obchodu, zbožím v obchodě a ostatním nastavením. Dále musí obsahovat nástroj, kterým bude zboží prodáváno zákazníkům. Nakonec je potřeba vyřešit vzhled obchodu, což je jedna z nejdůležitějších vlastností webových aplikací. Každý člověk se raději vrací na stránky vzhledově příjemnější, než na stránky, které jsou nehezké.

Druhá část úkolu projektu je vytvoření aplikace generující internetový obchod s možností vizuálních změn. Jak ale realizovat možnost vytvářet různé vizuální varianty systému? Nejlepším a nejkorektnějším řešením je použití kaskádních stylů. Změnou hodnot atributů jednotlivých elementů dosáhneme požadované vizuální variability systému. Aplikace musí dále vyřešit počáteční nastavení obchodu a databáze a nakonec způsob jakým bude obchod přenesen na webový server. Zde se samo nabízí řešení pomocí ftp, které je pro tento účel nejběžnější.

3.2 Analýza projektu

Při vývoji složitějších systémů je třeba výstižným způsobem popsat realitu. K tomu nám slouží model, který nám také pomůže při komunikaci mezi programátorem systému a zadavatelem.

Při vývoji informačních systémů se zpravidla vytvářejí dva typy základních modelů: model funkční a datový. Funkční model ukazuje funkce v systému a vazby mezi nimi. Datový model představuje data, která bude systém spravovat a vazby mezi nimi. V dnešní době se nejčastěji používají modely relační a objektové, přičemž pro obchodní informační systém je nejvýhodnější relační datový model.

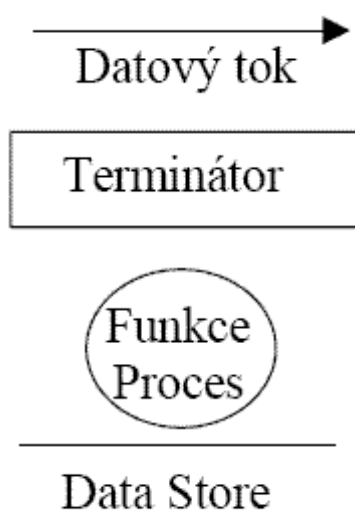
Výsledkem analýzy je fyzický datový model, podle něhož vytvoříme fyzickou implementaci dat – databázi v prostředí databázového serveru.

3.2.1 Funkční analýza

Pomocí funkční analýzy vytvoříme model systému jako množinu vzájemně provázaných funkcí. Často se k tomuto účelu používají DFD (Data Flow Diagram) známé také pod názvem De Marcovy diagramy. Výsledkem je vyjádření systému pomocí procesů propojených datovými toky.

DFD je prostředek jak graficky znázornit funkční model systému. Model pak vyjadřuje toky dat a jejich transformace, obsahuje popis funkcí, které provádějí transformaci dat. Není zde ovšem žádná časová závislost, ale pouze grafické znázornění závislostí mezi jednotlivými procesy v systému.

DFD ke složen ze čtyř základních prvků:



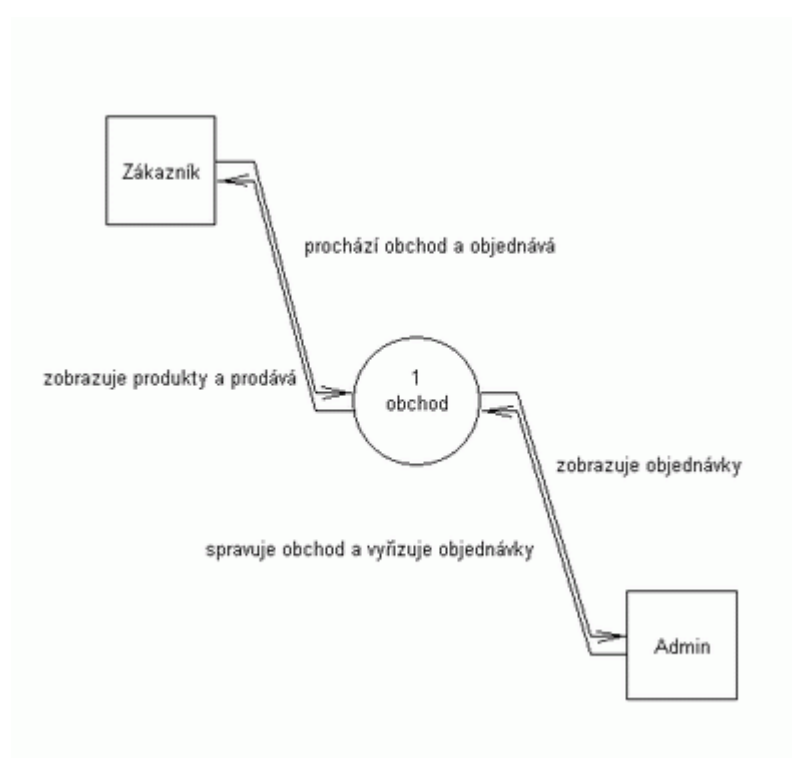
označuje směr přenosu dat – je orientovaný, pojmenovaný a strukturovaný

prvky z okolí systému, jsou zdrojem, nebo cílem datových toků, nepatří do systému

funkce, proces je transformace dat, při které vstupní data jsou transformována na nějaká data výstupní

je označení pro libovolný systém uchování dat, pasivní prvek, data v něm nevznikají

DFD je víceúrovňový diagram. První úroveň se nazývá kontextový digram, vymezuje hranice celého systému a způsob jakým bude komunikovat s okolím. Nižší úrovně postupně top-down rozkladem (rozklad shora dolů) detailněji popisují funkčnost procesů z vyšších úrovní. Kontextový diagram našeho systému je na obrázku 3 a druhá úroveň DFD je znázorněna na obrázku 4.



Obr. 3 Kontextový diagram

objednávku z obsahu košíku. Terminátor admin pak má na starosti správu obchodu, což obnáší úpravu kategorií, produktů, informací pro uživatele, vyřizování objednávek a další nastavení obchodu.

Druhá úroveň (obrázek 4) detailně popisuje vnitřní funkci celého systému z pohledu zákazníka i správce systému.

Ze schématu je vidět, že *zákazník* má na výběr několik akcí. V první řadě má možnost prohlížet informace obchodu pomocí horního menu (proces 1.3). Data pro horní menu jsou uložena v datastoru *topmenu_data*, což je očíslovaný adresář na webovém serveru obsahující HTML dokument.

Aby *zákazník* mohl být identifikován a mohl tedy nakupovat, je třeba se nejprve zaregistrovat (proces 1.7). Tento proces slouží též ke změně údajů zákazníka a tyto údaje uloží do datastoru *customers* (tabulka v databázi). Po registraci je možno se přihlásit do systému (proces 1.1), čímž dojde k nastavení hodnot v datastoru *sessions* a zpřístupní uživateli košík. Datastor *sessions* představuje proměnné sezení webového serveru.

Zákazník může procházet kategoriemi, případně vyhledávat a prohlížet produkty (proces 1.2). Data kategorií a produktů jsou uložena v datastorech *categories* a *products* (tabulky v databázi). Pokud je *zákazník* přihlášen, má možnost vkládat, případně odebírat produkty z košíku (proces 1.4). Informace o produktu do košíku jsou uložena v datastoru *products* (tabulka databáze) a samotný obsah košíku pak v datastoru *cart*. Po naplnění košíku produkty se zpřístupní akce vytvoření objednávky (proces 1.5). Nová objednávka je vytvořena na základě obsahu košíku a uložena do objednávek – datastor *orders* (tabulka v databázi).

Pro přístup do administrátorského rozhraní je třeba se přihlásit (proces 1.1), heslo je porovnáno s heslem v datastoru *admin* (tabulka v databázi). Poté může *admin* spravovat obchod, což obnáší změny v horním menu (proces 1.10) a přidávání, mazání, změny kategorií a produktů (proces 1.6 a proces 1.11). Procesy aktualizují obsahy datastorů *topmenu_data*, *categories* a *products*.

Druhou důležitou úlohou uživatele *admin* je práce s objednávkami. *Admin* prohlíží objednávky (proces 1.13), které může buď smazat (proces 1.13), nebo vyřídit (proces 1.14). Data pro objednávky jsou uložena v datastoru *orders*. Po vyřízení objednávky se příslušná objednávka vymaže.

V poslední řadě má admin k dispozici zobrazení registrovaných zákazníků (proces 1.12), které vybírá z datastoru *customers* a změnu administrátorského hesla (proces 1.9), které je uloženo v datastoru *admin*.

3.2.3 Datová analýza

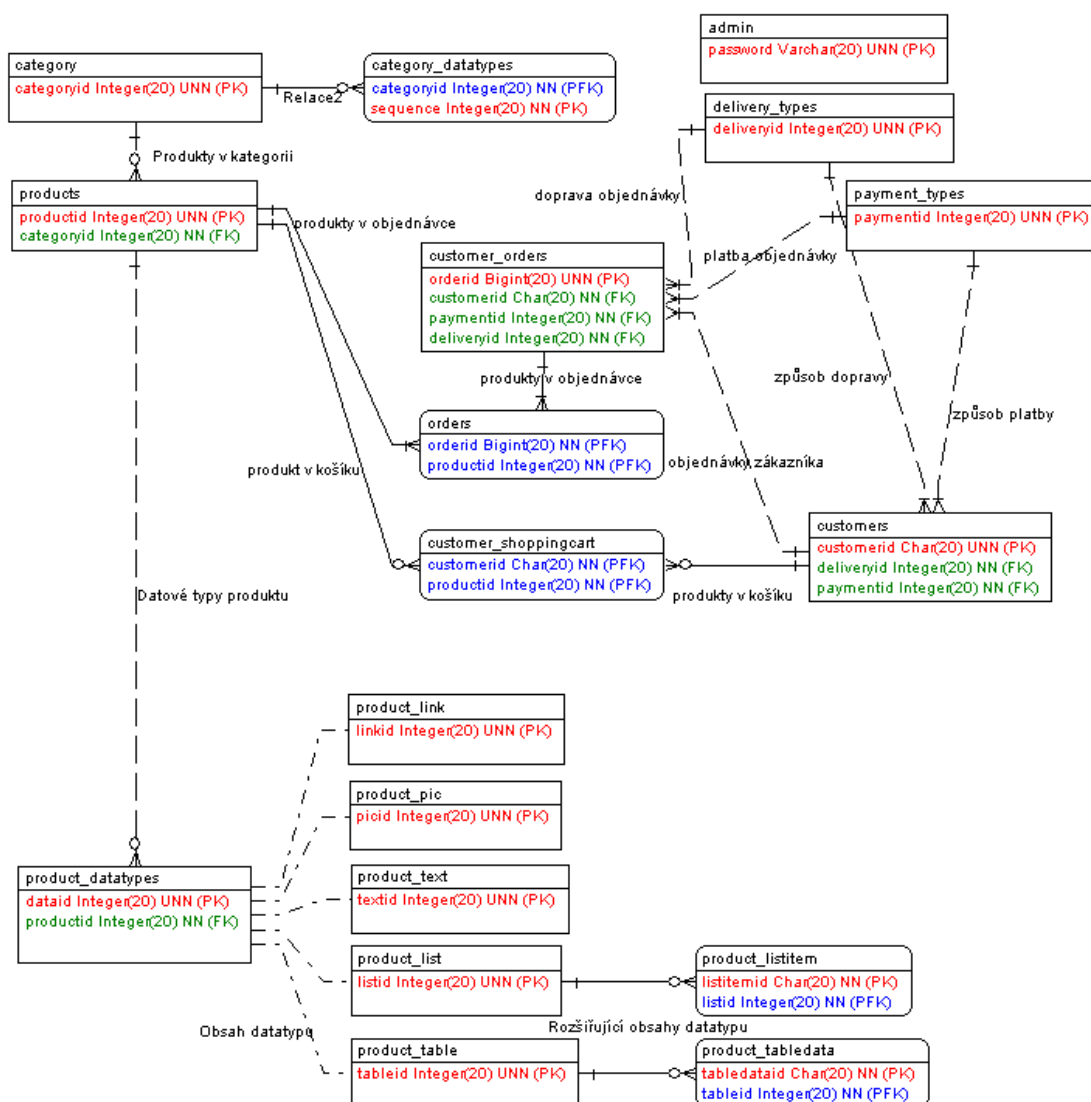
Úkolem datové analýzy je popsat vztahy mezi daty v systému a jejich strukturou. Výsledkem budiž grafický relační model, pro jehož realizaci použijeme ERD (entitně relační diagram).

ER diagram obsahuje dva typy prvků – entity a relace. Entity se zpravidla zobrazují jako obdélníky, které obsahují jméno entity. Entita reprezentuje množinu objektů v reálném světě, jejichž data potřebujeme ukládat. Entita je charakterizována jejími prvky – atributy. Entity jsou mezi sebou propojeny relacemi, které reprezentují množinu vazeb mezi entitami. Relace mají navíc definovanou *kardinalitu*, což je vyjádření mocnosti vztahu entit. Rozlišují se tři základní mocnosti:

- 1:1 – k jedné instanci entity A existuje jedna instance entity B.
- 1:N – k jedné instanci entity A existuje více instancí entity B.
- M:N – k jedné instance entity A existuje více instance entity B a zároveň k jedné instance entity B existuje instance entity A. Tuto relaci nelze v databázi realizovat a proto se tato relace nahrazuje entitou, která vztah M:N převede na dvě relace 1:N.

Druhou vlastností relací je *volitelnost*. Volitelnost určuje, jestli je výskyt entity v relaci povinný, nebo ne.

Datový model nám tedy popíše vztahy mezi entitami bez popisu jakýchkoliv funkcí, které jsou nutné pro manipulaci s daty.



Obr. 5 ER diagram internetového obchodu

3.2.4 Výsledky datové analýzy

Z provedené datové analýzy dostáváme logický datový model v podobě ER diagramu. V našem ER diagramu odpovídá každé entitě jedna fyzická tabulka v databázi. V entitách jsou pro větší přehlednost zobrazeny pouze atributy, které jsou primárním, nebo cizím klíčem.

V entitě *category* jsou uloženy kategorie pro produkty, záznamy jsou rozlišitelné pomocí primárního klíče *categoryid*. Tato entita je svázána relací 1:N s entitou *category_datatypes*, která obsahuje seznam výchozích datotypů produktů pro danou kategorii. Kategorie nemusí obsahovat žádné výchozí datatypy. Entita *category_datatypes* má za primární klíč dvojici cizí primární klíč *categoryid* z entity *category* a primární klíč *sequence*. Atribut *sequence* zde

zajišťuje uložení pořadí datatypů a *categoryid* propojuje obě entity. Pro každý datatyp dané kategorie existuje jedinečné pořadí a proto kombinace těchto dvou klíčů zajišťuje jedinečnost záznamu.

Kategorizace produktů uložených v entitě *product* je tedy určena entitou *category* a vzájemnou relací 1:N. Propojení je realizováno pomocí cizího klíče entity *produkt* – atributu *categoryid*. Záznamy v entitě *product* jsou identifikovatelné pomocí primárního klíče *productid*.

Každý produkt může kromě svých povinných atributů obsahovat také neomezený počet datatypů (pojem datatyp je vysvětlen v kapitole 3.5 Zajištění obecnosti obchodu – datatypy). Entita *product_datatypes* obsahuje informaci jaké typy datatypu produkt obsahuje. Záznamy jsou jednoznačně určeny primárním klíčem *dataid* a relace 1:N s entitou *product* je zajištěna pomocí cizího klíče *productid*. Vlastní obsah datatypu je uložen v entitách *product_link*, *product_pic*, *product_text*, *product_list* a *product_table*. Tyto entity jsou v relaci 1:1 s entitou *product_datatypes*, relaci zajišťuje primární klíč každé entity, jehož hodnota odpovídá primárnímu klíči entity *product_datatypes*. Zde by se mohlo zdát, že entita *product_datatypes* je zde zbytečná, ale není tomu tak. Jednotlivé datatypy obsahují různá data reprezentovaná v databázi různými datovými typy, a proto je není možno uložit do jedné tabulky, aniž by pokaždé zůstávala většina atributů prázdných. Entita *product_list* respektive *product_tabledata* je navíc v relaci 1:N s entitou *product_listitem* respektive *product_tabledata*, který zajištěn cizím klíčem *listid* respektive *tableid*. V těchto dvou entitách jsou uložena další data datatypu. Tato data jsou v oddělené entitě, jelikož jednomu datatypu může náležet více záznamů těchto entit (kupříkladu tabulka jako datatyp a buňky tabulky jako záznam). Záznamy jsou jednoznačně identifikovatelné dvojicí klíčů *listid* a *listitemid* respektive *tableid* a *tabledataid*, kde *listitemid* respektive *tabledataid* zajišťují uložení pořadí datatypů.

Nyní se zaměříme na entitu *customers*. V ní jsou uloženy informace o zákaznících a každý zákazník je identifikován primárním klíčem *customerid*. Entita je v relaci N:1 s entitami *delivery_types* respektive *payment_types*, které definují výchozí způsob doručení objednávky respektive platby za zboží pro zákazníka. Propojení tabulek zajišťuje cizí klíč *deliveryid* respektive *paymentid*.

Každý uživatel má svůj nákupní košík, v něm naskládané produkty, které chce koupit. Jedná se tedy o relaci M:N, který je třeba nahradit entitou a

dvěmi relacemi 1:N. Touto entitou je entita *customer_shoppingcart*, která je v relaci N:1 k entitám *product* a *customers*. Fyzické propojení je realizováno pomocí cizích klíčů *customerid* a *productid*, které zároveň tvoří primární klíč.

Z obsahu košíku vytváří uživatel objednávky, informace o objednávce jsou uloženy v entitě *customer_orders*, která je v relaci N:1 k entitě *customers*. Propojení zajišťuje cizí klíč *customerid*, objednávka je unikátní díky primárnímu klíči *orderid*. Pro určení způsobu doručení respektive platby je entita v relaci N:1 s entitou *delivery_types* respektive *payment_types*. Objednávka může obsahovat více různých produktů a určitý produkt může být ve více objednávkách, jde tudíž opět o relaci typu M:N, kterou je potřeba nahradit, což dělá entita *orders*. Entita *orders* je v relaci N:1 s entitou *product* respektive *customer_orders*. Fyzicky jsou tabulky propojeny cizími klíči *productid* respektive *customer orderid*.

Poslední entita *admin* není v relaci s žádnou jinou entitou, protože obsahuje pouze heslo pro administrátora, které nemá žádnou vazbu na ostatní data.

Výsledný SQL skript pro generaci tabulek databáze je možno nalézt v příloze.

3.3 Systémové zázemí

Pro funkci internetového obchodu je samozřejmě nutné umístit systém na nějaký webový server. Není důležité jaký konkrétní webový server vybereme (Apache, MS IIS a další), je ovšem nezbytné mít přístup na tento webový server do místa uložení obchodu pomocí protokolu ftp. Na tomto serveru musí dále pracovat PHP stroj (nejlépe verze 5 a vyšší).

Druhý požadavek se týká databáze, do níž se budou ukládat data obchodu. Databázový systém musí být MySQL (nejméně verze 4.1) a v něm je třeba mít vytvořenu databázi s plnými přístupovými právy.

3.4 Pravidla vývoje obchodu

Před započítím práce na internetovém obchodu jsem si stanovil několik pravidel, která mi v delším časovém horizontu usnadní práci a případné pozdější změny. Jedná se zejména o tato pravidla:

- opětovné využívání kódu

- členění zdrojového kódu
- oddělení logiky od obsahu

Skupina problémů, které může vývojář řešit je omezená a je tedy velmi pravděpodobné, že kód který píše, byl již někým napsán. Velmi často programátor dokonce píše kód, který je přímo funkcí v knihovně programovacího jazyka. Vyplatí se tedy před psaním kódu prohlédnout manuál, zda neobsahuje nějakou funkci, která provádí to, co potřebujeme. Složitější úlohy většinou v knihovnách nebývají, což ovšem neznamená, že řešení již neexistuje. Práce v Open Source komunitě má tu výhodu, že vývojáři mezi sebou sdílí aplikační komponenty, které jsou většinou zdarma. Pokud nám daná komponenta přesně nevyhovuje, máme možnost si ji navíc upravit podle vlastní potřeby. Já jsem této možnosti využil při programování odesílání elektronické pošty. PHP sice nabízí sadu základních funkcí, ale ty byly nevyhovující a proto jsem využil možnost použít knihovny Pear, jmenovitě PearMail.

Členění zdrojového kódu do funkcí a rozdělení příbuzných funkcí do oddělených souborů je výhodné zejména proto, že:

- kód je čitelnější a srozumitelnější
- kód se snadno opětovně využívá a redukuje se jeho redundance
- usnadňuje týmovou práci

Oddělením logiky od obsahu znamená oddělení kódu PHP a HTML, případně XHTML. Toto pravidlo se ukáže důležitým v případě, že je potřeba změnit zásadně design. Měnit se bude v tomto případě jen část obsahující obsah a logika zůstane beze změn. Já jsem pravidlo zrealizoval tak, že do zobrazovaného dokumentu jsou vkládány jen funkce, které generují HTML kód, a funkce potřebné pro získání parametrů do funkcí s HTML výstupem. Navíc všechny funkce generující nějaký HTML výstup jsem soustředil do jedné knihovny *output_fns.php*.

3.5 Zajištění obecnosti obchodu – datatypy

Zásadní otázkou bylo, jak zajistit obecnost internetového obchodu. Obecný obchod je takový, který může teoreticky obsahovat jakýkoliv druh zboží. Je tedy potřeba vypracovat způsob, jak uložit do databáze informace o úplně rozdílných produktech. Jelikož ve webové prezentaci je jen několik struktur pomocí kterých se dají interpretovat data, vzal jsem nejdůležitější z nich a

nazval je datatypy. Jedná se celkově o pět datatypů a jsou to datatypy hypertextový odkaz, text, seznam, tabulka a obrázek. Každý produkt může obsahovat libovolné množství těchto datatypů, přičemž se ukládá i jejich vzájemné pořadí v rámci jednoho produktu pro účel zobrazování. Tímto způsobem je možné definovat strukturu dat libovolného produktu.

Definovat datatypy pro každý produkt zvlášť by ovšem nebylo nejlepší řešení, protože budou s největší pravděpodobností existovat skupiny produktů se stejnou strukturou dat a tyto produkty logicky rozdělíme do kategorií. Definici druhů datatypů produktu jsem přesunul tedy do kategorií

3.6 Grafický návrh internetového obchodu

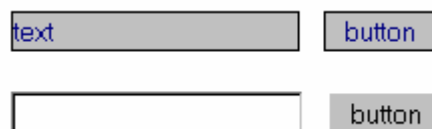
Dle zásady oddělení struktury a vzhledu webového dokumentu se veškeré vizuální vlastnosti dokumentu popisují kaskádními styly zatímco struktura je popsána pomocí HTML. Já jsem CSS styly použil na změnu vzhledu vstupních formulářových elementů (vstupní textová pole a tlačítka), formátování textu a zejména pozicování elementů v dokumentu a nastavení barev v nich.

Na obrázku 6 jsou zobrazeny formulářové elementy upravené pomocí stylů a stejné elementy se standardním vzhledem. Následuje výpis stylů pro upravené prvky:

```
input.styled
{
    border: 1px solid black;
    background-color: Silver;
    Color: Navy;
}




input[type="submit"].styled
{
    border: 1px solid black;
    background-color: Silver;
    Color: Navy;
}
```



Obr. 6 Formulář. elementy

Příklady formátování textu jsou uvedeny níže. CSS ovšem nabízí mnohem širší škálu možností pro formátování, na obrázku je zobrazeno pouze základní formátování. Příslušné styly jsou uvedeny níže.

```

span.style1
{
    font-family: Tempus Sans ITC;
}

span.style2
{
    color: Blue;
}

span.style3
{
    font-weight: bold;
}

span.style4
{
    font-size: 20pt;
}

```

Styl 1
 Styl 2
 Styl 3
 Styl 4

Obr. 7 Formátování textu

Pozicování elementů v dokumentu je možno realizovat pomocí tabulek. Tento přístup ale není vůbec vhodný a nedoporučuje se již používat, neboť takovéto pozicování vyžaduje velké množství tabulek, kód dokumentu se pak stává nepřehledným a navíc může při použití mnoha tabulek dojít k nepředvídatelnému chování internetového prohlížeče. Mnohem lepší metodou je kaskádními styly nastavit způsob zobrazení a pozici kontejnerů (HTML tag <div>) vůči okolním elementům. Tyto kontejnery obsahují vlastní prvky dokumentu, které se zobrazují.

V mé aplikaci jsem pro základní rozvržení vzhledu dokumentu použil šest kontejnerů. Jsou to kontejnery pojmenované *main*, *top*, *middle*, *menu*, *content* a *bottom*. Kontejner *main* je hlavním kontejnerem a obsahuje všech zbylých pět. Jeho styly mají hodnoty nastaveny tak, aby byl horizontálně zarovnán na střed. Kontejner *top* je první potomek kontejneru *main*, a proto je zobrazen v horní oblasti. Následuje kontejner *middle*, který je rodičem kontejnerů *menu*, *content* a *bottom*. Kontejnery *menu* (oblast menu) a *content* (oblast obsahu) jsou nastaveny jako plovoucí vlevo respektive vpravo, čímž dostaneme jejich zobrazení vedle sebe. Kontejner *bottom* ruší obtékání obsahu okolo kontejnerů *menu* a *content* a vymezuje spodní oblast. Výsledek je zobrazen na obrázku 8.



Obr. 8 Vymezení základních oblastí

Jak je patrné z obrázku 8, viditelné jsou pouze kontejnery *top*, *menu*, *content* a *bottom*, zbylé dva slouží pouze k realizaci pozicování.

Jeden z mnoha problémů se kterými jsem se zde potýkal, bylo vytvořit styly tak, aby celá oblast menu měla určitou barvu a oblast obsahu barvu jinou. Problém tkví v tom, že oba kontejnery mají proměnný obsah a tím pádem mění svoji výšku, přičemž není způsob jak pouze nastavením stylů kontejnerů zajistit, aby měli vždy shodnou výšku. Pokud tedy nastavím barvu pozadí obou kontejnerů, pak pod nižším z nich bude zobrazena barva pozadí jejich rodičovského kontejneru *middle*. Standardním řešením tohoto problému je vytvoření obrázku o výšce jeden pixel a šířce odpovídající šířce levého kontejneru. Tento obrázek se nastaví jako pozadí rodičovského kontejneru s vertikálním opakováním. Barva pravého sloupce se nastaví pomocí barvy pozadí rodičovského kontejneru. Toto řešení jsem použil ve svém projektu.

Dále je třeba zmínit podporu kaskádních stylů v internetových prohlížečích. Největší obtíže působí paradoxně nejrozšířenější prohlížeč Microsoft Internet Explorer. Tento prohlížeč totiž ignoruje velké množství specifikací a různé hodnoty vlastností stylů interpretuje nesprávně a odlišně, narozdíl od prohlížečů Mozilla Firefox, nebo Opera. Toto způsobuje obrovské těžkosti při návrhu vzhledu internetových stránek, neboť design stránek může být ve Firefoxu zobrazen korektně a v Internet Exploreru úplně chybně. Internet Explorer navíc obsahuje řadu známých chyb. Já jsem se nejvíce setkával s chybou, kdy při použití plovoucích prvků je obtékající text zobrazen tak, jakoby měl na straně u plovoucího prvku 3 pixely široký okraj. Jediným možným řešením této situace je často definice jiných stylů pro Internet Explorer a zbylé prohlížeče. Toto řešení jsem použil ve svém projektu. Existují pak dva externí soubory s kaskádními styly, přičemž první se aplikují styly pro všechny prohlížeče a pak se aplikují styly pouze pro Internet Explorer, které přepíší vybrané hodnoty stylů předchozích. Je třeba, aby se druhý soubor stylů

aplikoval pouze v Internet Exploreru, což je zařízeno následujícím řádkem v hlavičce HTML dokumentu:

```
<!--[if IE ]><link href="./styles/styles_ie.css" type="text/css"
rel="stylesheet" /><![endif]-->
```

Celý řádek je zakomentován, proto jej prohlížeče ignorují. Internet Explorer ho ovšem rozpozná, neboť se jedná o podmínkový komentář – specialitu Internet Exploreru, a řádek interpretuje.

Nakonec této kapitoly uvádím ukázkou vzhledu internetového obchodu:

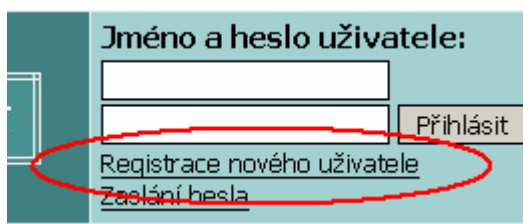


Obr. 9 Vzhled stránky s produkty

3.7 Uživatelská část

3.7.1 Registrace zákazníka a změna údajů

Registrace zákazníka je přístupná přes odkaz (obrázek 10) ve formuláři pro přihlášení v pravém horním rohu stránky, jak je vidět na obrázku 9.



Obr. 10 Registrace zákazníka

Po kliknutí na tento odkaz se zobrazí formulář pro zadání údajů o zákazníkovi, viz. obrázek 11.

Údaje o firmě, nebo soukromé osobě	
Přihlašovací jméno*:	<input type="text"/>
Heslo*:	<input type="password"/>
Heslo (podruhé)*:	<input type="password"/>
Jméno firmy/Jméno + příjmení*:	<input type="text"/>
IČO:	<input type="text"/>
DIČ:	<input type="text"/>
Adresa*:	<input type="text"/>
Město*:	<input type="text"/>
PSČ*:	<input type="text"/>
Telefon*:	<input type="text"/>
Fax:	<input type="text"/>
Email*:	<input type="text"/>

Doručovací adresa(pokud se liší):	
Adresa:	<input type="text"/>
Město:	<input type="text"/>
PSČ:	<input type="text"/>

Nastavení	
Výchozí způsob dopravy:	<input type="text"/>
Výchozí způsob platby:	<input type="text"/>

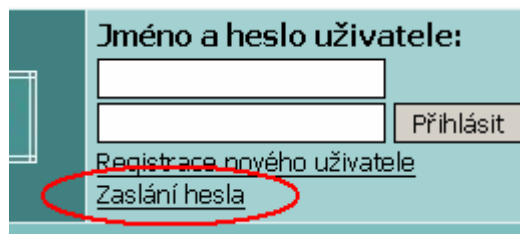
Vytvořit

Obr. 11 Formulář pro zadání údajů zákazníka

Údaje označené hvězdičkou jsou povinné, část doručovací adresa se vyplňuje jen v případě, že bývá odlišná od adresy zákazníka. V tabulce nastavení se nastavují výchozí způsoby doručení zboží a platby.

Po odeslání formuláře na server se data uloží do databáze a zákazníkovi je přidělen jedinečný kód. Heslo zákazníka je zakódováno funkce PHP crypt() a uloženo do databáze. Takto zakódované heslo nelze již rozkódovat, ale výsledek kryptovací funkce crypt() pro daný vstup je vždy stejný, a proto nám umožní hesla později porovnávat.

Pro zapomnětlivé uživatele je v přihlašovacím okně odkaz (obrázek 12), který vede na formulář pro resetování a zaslání nového hesla na zadanou emailovou adresu. Systém se pokusí uživatelem zadanou adresu vyhledat v databázi v tabulce uživatelů a pokud ji nalezne, heslo změní a zašle na tuto adresu.



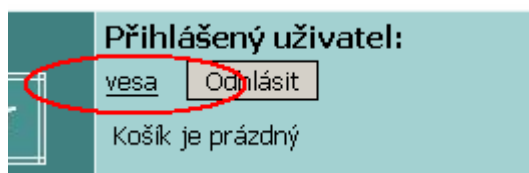
The screenshot shows a login form titled "Jméno a heslo uživatele:". It contains two input fields for username and password, followed by a "Přihlásit" button. Below the input fields, there are two links: "Registrace nového uživatele" and "Zaslání hesla". The "Zaslání hesla" link is circled in red.

Obr. 12 Zákazník zapomněl heslo

3.7.2 Přihlášení do systému

Aby se zákazník mohl přihlásit, musí být již v systému registrován. Pro akci přihlášení slouží formulář v pravém horním rohu stránky. Po odeslání přihlašovacího jména a hesla se nejprve vyhledává uživatel. Pokud existuje, zakóduje se odeslané heslo a porovná s heslem v databázi. Pokud jsou si hesla rovna, dojde k nastavení proměnné sezení *sessions*. Pokud je zákazník přihlášen, může používat nástroj košík a měnit své údaje.

Změna údajů zákazníka se provádí přes odkaz (obrázek 13), který vede na téměř totožný formulář jako při registraci. Způsob práce a uložení je také stejný.



The screenshot shows a user profile page titled "Přihlášený uživatel:". It contains a text input field with the name "vesa" and an "Odlásit" button. Below the input field, there is a message "Košík je prázdný". The "Odlásit" button is circled in red.

Obr. 13 Modifikace údajů zákazníkem

3.7.3 Horní menu informací obchodu

Horní menu (obrázek 14) informací nabízí zákazníkovi různé informace vytvořené správcem systému. Menu obsahuje seznam položek, které jsou odkazy a po kliknutí na ně se zobrazí příslušný obsah.



The screenshot shows a horizontal navigation menu with four items: "Vítejte", "O nás", "Speciální akce", and a partially visible "Kontakt" on the right. Each item is enclosed in a box with a vertical separator on the right.

Obr. 14 Horní menu

3.7.4 Kategorie produktů

Kategorie produktů slouží k rozdělení produktů do skupin. Kategorie se dají procházet buď kliknutím na jméno požadované kategorie, nebo lze využít rozbalovacích a zabalovacích prvků. Použití rozbalovacího prvku má stejný účinek jako kliknutí na jméno kategorie – kategorie se rozbalí. Naproti tomu kliknutí na zabalovací prvek zabalí kategorii a aktuální kategorií se stane kategorií nadřazená. Příklad kategorií je na obrázku 15.



Obr. 15 Menu kategorií

Kategorizace produktů pak funguje tak, že nadřazenější kategorie vlastní produkty všech koncových kategorií, které jsou jejími potomky. Pokud tedy postupujeme od vyšší kategorie k nižší, zužuje se nám tím výběr produktů a postupně se dostáváme k hledanému produktu.

Tím, že vybere nějakou kategorii, dojde ke změně proměnné sezení *categoryid*. Tato proměnná je řídicí hodnotou pro zobrazování produktů v dané kategorii.

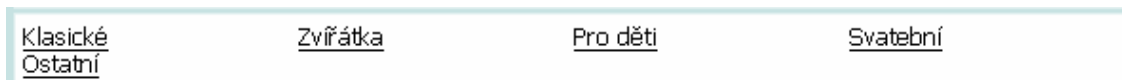
Pro snadnější orientaci v kategoriích obchod nabízí další dva prvky umístěné v obsahové oblasti obchodu. Jsou to prvek navigační panel, často nazývaný jako drobky (obrázek 16), a panel zobrazující kategorie (obrázek 17), které jsou prvními potomky aktuální kategorie.

Pomocí navigačního panelu – drobků se můžeme okamžitě dostat do všech jeho rodičovských kategorií. Ukazuje nám vlastně cestu od kategorie v nejvyšší vrstvě ke kategorii aktuální.



Obr. 16 Drobky

Druhý prvek nám umožňuje postupovat níže ve struktuře kategorií, ale nedostaneme se pomocí něj zpět. Jeho největší výhodou je přehledné zobrazení jemu přímo podřízených kategorií.



Obr. 17 Panel podřízených kategorií

3.7.5 Vyhledávání produktů

Systém nabízí jednoduché vyhledávání produktů podle části jména, nebo jedinečného kódu produktu. Vyhledávání podle jména zobrazí všechny produkty, jejichž jméno obsahuje zadaný řetězec. Vyhledání produktu dle jeho kódu pak zobrazí tento produkt za předpokladu, že existuje. Formulář pro zadání kritérií vyhledávání je na obrázku 18.

Obr. 18 Formulář vyhledání produktů

3.7.6 Seznam produktů

Seznam produktů je umístěn v hlavní části obchodu a množina produktů je určena výběrem kategorie respektive proměnnou sezení *categoryid*. Nad vlastním seznamem se nachází panel nastavení zobrazení seznamu produktů (obrázek 19).

Obr. 19 Panel nastavení zobrazení seznamu produktů

Vstupní prvky panelu nastavení zobrazení umožňují zvolit režim výpisu s obrázky (obrázek 20), nebo rychlý výpis v tabulce (obrázek 21).




Dále je možné zvolit řazení produktů buď podle jména, nebo podle ceny produktu. Posledním prvkem je počet najednou zobrazovaných položek. Pokud je produktů v kategorii více, než je hodnota zobrazovaných položek, přistupujeme k dalším stránkám pomocí čísel stránek umístěných v levé části panelu.

Oba režimy výpisu produktů obsahují název produktu, cenu a ikonku košíku pro možnost vložení produktu do košíku. Výpis s obrázky obsahuje ikonku produktu a stručný popis, zatímco rychlý výpis má navíc kód produktu a

počet kusů skladem. Klepnutím na jméno produktu (případně ikonku produktu) se zobrazí detail produktu.

1	výpis: s obrázky	řazení: dle názvu	položek: 9	Ok
				
Auto	Beruška	Medvěd		
200,- Kč bez DPH	238,- Kč s DPH	190,- Kč bez DPH	200,- Kč s DPH	170,- Kč bez DPH
Pro milovníky dopravních prostředků.	Roztomilý dort pro malé i velké slečny.	Méd'á potěší všechny děti. xxxxxxxxxxxxxxxxxxxxxxxxxxxx		



Obr. 20 Výpis seznamu produktů s obrázky

Kód	Název	Cena bez DPH	Cena s DPH	Skladem	
1	<u>Auto</u>	200,- Kč	238,- Kč	0	
2	<u>Beruška</u>	190,- Kč	200,- Kč	0	
3	<u>Medvěd</u>	170,- Kč	179,- Kč	0	

Obr. 21 Rychlý výpis seznamu produktů

3.7.7 Detail produktu

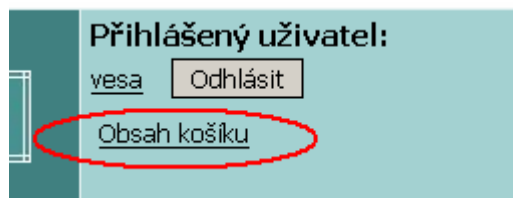
Kliknutím na produkt v seznamu produktů se dostaneme na detailní zobrazení produktu (obrázek 22), který obsahuje všechny atributy daného produktu a zobrazuje obsahy všech datatypů produktu v daném pořadí. I zde je umístěna ikonka košíku stejně jako u seznamu produktů.

Auto	
	Kód: 1 Skladem: 0 Cena bez DPH: 200,- Kč Cena s DPH 19%: 238,- Kč 
Pro milovníky dopravních prostředků.	
Popis	
Polovina kulatého dortu na smetánkovém hřebetu. Kombinace světlého a kakového korpusu s vanilkovým a čokoládovým krémem.	
Přízdoba	
• číslo	
Složení	
<ul style="list-style-type: none"> • mouka • cukr • kypřidlo • vejce • máslo • pudink • kakao • čokoláda 	

Obr. 22 Detailní pohled na produkt

3.7.8 Nákupní košík

Produkty do košíku může vkládat pouze přihlášený uživatel a činí tak pomocí ikonky košíku v seznamu produktů, nebo detailu produktu. Do košíku samotného lze vstoupit přes odkaz umístěný v přihlašovací oblasti (obrázek 23).



Obr. 23 Nákupní košík

Zde vidíme v přehledné tabulce produkty v košíku a jejich důležité atributy (obrázek 24). Produkty můžeme z košíku odstraňovat buď výběrem, nebo kompletním vysypáním. Pokud již máme košík plný, vytvoříme objednávku. Předtím je ale třeba zvolit požadovaný způsob platby a dopravy a případně upravit množství jednotlivých položek. Po odeslání objednávky se košík automaticky vyprázdní.

	Kód	Název produktu	Množství	Skladem	Cena/kus	DPH
<input type="checkbox"/>	1	Auto	<input type="text" value="1"/>	0	200,- Kč	19%
<input type="checkbox"/>	3	Medvěd	<input type="text" value="1"/>	0	170,- Kč	5%

Způsob dopravy:

Způsob platby:

Obr. 24 Obsah košíku

3.7.9 Vytvoření objednávky

Obsah objednávky je vytvořen podle produktů, které jsou v košíku. V objednávce jsou zobrazena stejná data jako u obsahu košíku, obsahuje však navíc součet cen jednotlivých produktů s ohledem na množství spolu s cenou za doručení zboží. Nezbytným krokem před odesláním objednávky je uvést adresu, město a PSČ na které bude zásilka doručena. Tyto hodnoty jsou již přednastaveny na základě informací o daném zákazníkovi v databázi. Zákazník má také možnost připojit k objednávce poznámku. Vzhled objednávky vidíme na obrázku 25.

Objednávka

Kód	Název produktu	Množství	Skladem	Cena/kus	DPH	Celkem s DPH
1	Auto	1	0	200,- Kč	19%	238
3	Medvěd	1	0	170,- Kč	5%	179
Poštovné:						50,- Kč
Celkem bez DPH:						420,- Kč
Celkem s DPH:						474,- Kč

Způsob dopravy: poštou
Způsob platby: kartou

Adresa pro doručení:

Adresa: Mšenská 48

Město: Jablonec nad Nisou

PSČ: 46604

Poznámka k objednávce:

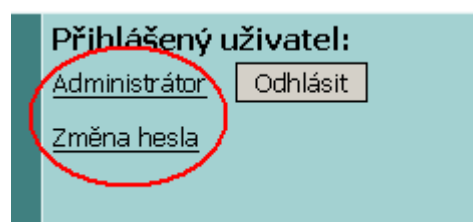
Objednat

Obr. 25 Objednávka

3.8 Část správce systému

3.8.1 Přihlášení a změna hesla

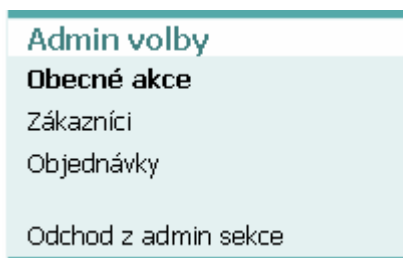
Pro přístup k veškerým nástrojům správce je nezbytné se přihlásit jako uživatel admin (přednastavené heslo v novém obchodě je také admin). Přihlášení probíhá stejným způsobem jako u zákazníků. Po přihlášení se zobrazí odkaz vedoucí na formulář pro změnu hesla správce obchodu a odkaz, který vede do správcovské sekce (obrázek 26).



Obr. 26 Odkazy pro změnu hesla a přístup do sekce správce

Zde musím podotknout, že ne všechny nástroje správce se nacházejí ve správcovské sekci. Prvky pro správu kategorií a produktů jsou zobrazeny přímo u menu kategorií respektive seznamu produktů. Důvodem je vyšší přehlednost při zásazích do kategorií a produktů a ušetření práce, která by musela být vynaložena pro tvorbu obdobného zobrazení produktů a kategorií v sekci správce.

Členění správcovské části je zajištěno pomocí menu Admin volby, které se skládá z položek Obecné akce, Zákazníci, Objednávky a Odchod z admin sekce. Toto menu je vidět na obrázku 27.



Obr. 27 Menu Admin volby

3.8.2 Správa horního menu informací

Horní menu informací slouží k zobrazení libovolného dokumentu s nějakou informativní hodnotou pro zákazníky. Pro tvorbu takového obsahu jsem vytvořil v programovacím jazyce Delphi velmi jednoduchý editor HTML dokumentů. Při vývoji této aplikace jsem použil pro nekomerční účely volně šiřitelné komponenty RichView. Nevýhoda použití těchto komponent byla absence dokumentace, proto jsem musel vycházet z přiložených vzorových řešení, ale i přesto byla práce s komponentami dosti obtížná. Druhou nevýhodou bylo to, že vytvořený HTML kód neodpovídal specifikaci XHTML, a proto jsem musel vytvořit filtr, pomocí kterého aplikace vygeneruje validní XHTML kód (skutečně validním se stává až po zobrazení v systému obchodu, neboť postrádá vše co není uzavřeno mezi tagy <body> a </body> včetně těchto tagů).

Aplikace je velmi podobná produktu Wordpad, nebudu proto popisovat podrobně její funkci, ale pouze shrnu v bodech její možnosti.

- aplikace umožňuje běžné formátování textu
- do dokumentu je možné vložit číslovaný, nebo nečíslovaný seznam, tabulky, nebo obrázky
- aplikace umí vytvářet hypertextové odkazy
- dokumenty je možné ukládat ve formátu .rvf a zpětně je otevírat
- aplikace obsahuje nástroj pro vložení vytvořeného dokumentu do horního menu internetového obchodu

Poslední zmíněnou funkci je třeba podrobněji rozebrat, jelikož je speciální pro tuto aplikaci a zásadní pro vytvoření nových položek horního

menu. Funkce se nachází v hlavním menu Soubor, položka Upload na server. Po odklepnutí se zobrazí okno pro zadání údajů potřebných pro ftp připojení a pole pro název položky v horním menu (obrázek 28). Klepnutím na tlačítko OK se spustí proces nahrávání dat na webový server.

Upload dokumentu na server

FTP server: 192.168.1.10 : 21

Login: admin Heslo: heslo

Adresář na FTP: ./DP/html/

Název položky: Vítejte

OK Zrušit

Obr. 28 Okno aplikace pro zadání údajů nové položky

Fyzické uložení dat položek horního menu je v adresáři ./html/ vzhledem ke kořenovému adresáři celého systému. Každá položka je přitom uložena v samostatném adresáři, jehož název je číslo pořadí položky v menu. Pro druhou položku zleva jsou její data tedy v adresáři ./html/2/. Adresář již obsahuje samotná data a to konkrétně soubor data.html, což je zobrazovaný dokument, soubor name.txt, který obsahuje jméno položky a nakonec případné obrázky pro dokument data.html.

Správu horního menu doplňují nástroje správce obchodu pod položkou Obecné akce (obrázek 29). Správce může měnit pořadí položek, mazat položky a upravovat jména položek. Změna pořadí je zrealizována prohozením čísel adresářů a upravením cest k obrázkům v souboru data.html. Smázání položky smaže příslušný adresář a zbytek adresářů přečísluje, změna názvu položky provede změnu souboru name.txt.

Úprava horního menu

Vítejte	<	Smazat	>
O nás	<	Smazat	>
Speciální akce	<	Smazat	>

Uložit

Obr. 29 Nástroj pro modifikaci horního menu

3.8.3 Správa kategorií produktů

Kategorie produktů je stromová struktura, kde její kořen je nultá kategorie, která se nezobrazuje.

Každá kategorie může mít nastaveny dva příznaky – koncová kategorie a slepá kategorie. Slepá kategorie může být jen v první hladině a nemůže obsahovat žádné produkty, ani další kategorie. Je to vlastně pouze pomocný vizuální prvek, který slouží jako mezinadpis. Koncová kategorie může obsahovat produkty, zatímco kategorie bez obou příznaků obsahuje další kategorie.

Správce má možnost vytvářet, modifikovat, mazat kategorie a měnit jejich vzájemné pořadí v rámci jedné hladiny, což vidíme na obrázku 30. Tyto jednotlivé prvky se zobrazují pouze je-li akce přípustná. Novou kategorii je možno vytvořit pouze v případě, pokud není aktuální kategorie koncová, nastavení příznaku slepá je možné jen pokud bude nová kategorie v první hladině, příznak koncová je možno nastavit vždy. Editace kategorie je možná vždy, ale nastavení příznaku koncová je možné jen, když je kategorie prázdná a nemá žádné potomky. Příznak slepá je navíc omezen pouze na kategorie v první hladině. Smazání kategorie je přípustné jen v případě, že neobsahuje žádné produkty ani další kategorie.

The image shows a vertical stack of four form sections for managing categories. Each section has a title bar with a teal background and a light blue border. The first section, 'Nová kategorie:', contains a text input for 'Jméno:', two checkboxes for 'koncová' and 'slepá', and a 'Vytvořit' button. The second section, 'Editace kategorie:', contains a text input for 'Jméno:' with the value 'Tradiční', two checkboxes for 'koncová' and 'slepá', and a 'Změnit' button. The third section, 'Smazat kategorii', contains a single 'Smazat' button. The fourth section, 'Posun kategorie', contains two buttons: 'Posunout nahoru' and 'Posunout dolů'.

Nová kategorie:	
Jméno:	<input type="text"/>
<input type="checkbox"/> koncová	<input type="checkbox"/> slepá
<input type="button" value="Vytvořit"/>	

Editace kategorie:	
Jméno:	<input type="text" value="Tradiční"/>
<input type="checkbox"/> koncová	<input type="checkbox"/> slepá
<input type="button" value="Změnit"/>	

Smazat kategorii	
<input type="button" value="Smazat"/>	

Posun kategorie	
<input type="button" value="Posunout nahoru"/>	
<input type="button" value="Posunout dolů"/>	

Obr. 30 Formuláře pro vytváření a modifikaci kategorií

Po odeslání formuláře pro novou kategorii se dostaneme na stránku umožňující editaci datotypů produktů, které budou do dané kategorie patřit (obrázek 31). Výchozí nastavení se převezme z rodičovské kategorie, protože je zde předpoklad, že datatypy potomků a rodičů budou obdobné. Datatypy je zde možno přidávat, mazat a měnit jejich vzájemné pořadí.

Úprava datových typů kategorie			
Text	Smazat	^	v
Hypertextový odkaz	Smazat	^	v
Text	Smazat	^	v
Obrázek	Smazat	^	v
Nový dat. typ: Hypertextový odkaz ▼ Přidat			
Vytvořit kategorii			

Obr. 31 Úprava datotypů nové kategorie

3.8.4 Správa produktů

Přihlášení správce způsobí zobrazení formulářových prvků v oblasti seznamu produktů. Formuláře v panelu nad seznamem produktů (obrázek 32) slouží pro vkládání nových a hromadné mazání existujících produktů.

☐
Přidat nový
Smazat vybrané

Obr. 32 Tlačítka pro akce hromadný výmaz produktů a vložení nového

Hromadný výmaz produktů vymaže všechny produkty, které mají zaškrtnutý element checkbox. Poklepáním na checkbox, který se nachází nalevo od tlačítek se vyberou všechny produkty na dané stránce.

Stisk tlačítka pro nový produkt přenese správce na formulář zadávání informací o produktu a jeho datatypech (obrázek 33), které jsou definované aktuální kategorií. Do této kategorie bude také produkt umístěn. Formulář má navíc vlastnost, že dědí data od nejnižšího produktu v aktuální kategorii. Toto může urychlit zadávání produktů, které se jen velmi málo liší.

Nový produkt

Nový produkt - hlavní údaje

Název: Medvěd

Cena: 170 Kč

DPH: 5%

Popis: Méd'a potěší všechny děti.

Ikona: Browse...

Text

Nadpis: Popis

Hodnota: Kulatý dort s vanilkovou náplní s ušima z kakaového korpusu a náplně.

Seznam

Nadpis: Přídoba

Počet řádků: 1

Seznam

Nadpis: Složení

Počet řádků: 9

Pokračovat

Obr. 33 Zadávání dat pro nový produkt

Na obrázku 33 vidíme také dva datatypy produktu. První je typu text a druhý je typu Obrázek. Všechny datatypy mají navíc vlastnost Nadpis, kterým je možné nadepsat data obsažená v datatypu. Nadpis může být prázdný.

3.8.5 Objednávky

V menu Admin volby je umístěna položka Objednávky, která zobrazí všechny objednávky odeslané v určitém časovém rozmezí (obrázek 34). Tento časový interval je možné libovolně nastavit pomocí vstupních polí v záhlaví tabulky. Dále je možné nastavit řazení objednávek podle datumu odeslání, nebo jména zákazníka. Hlavním obsahem okna jsou pak samotné objednávky, které mají svůj identifikační unikátní kód, datum odeslání a jméno zákazníka, kterému objednávka patří.

Seznam objednávek

řazení: datum od: datum do:

Kód	Datum objednávky	Zákazník	
1	02.07.2006	Lukáš Veselý	<input type="button" value="Smazat"/> <input type="button" value="Zpracovat"/>

Obr. 34 Seznam objednávek

Objednávku je možné vymazat, nebo postoupit dalšímu zpracování. Dalším zpracováním se rozumí zobrazení všech nezbytných detailů pro fyzické odeslání objednávky (obrázek 35). Správce má možnost odeslat informativní email po dokončení zpracování objednávky a v něm specifikovat poznámku. Tento email obsahuje seznam a ceny odeslaného zboží a další informace. Tlačítko pro dokončení objednávky je ovšem přístupné pouze v případě, že všechny produkty v objednávce jsou skladem v požadovaném množství. Po dokončení objednávky je objednávka vyřazena ze systému.

Objednávka

Informace o objednávce

Objednávající: Lukáš Veselý
Datum objednávky: 02.07.2006
Způsob doručení: poštou
Způsob platby: kartou

Doručovací údaje:

Adresa: Mšenská 48
Město: Jablonec nad Nisou
PSČ: 46604

Kód	Název produktu	Množství	Skladem	Cena	Sazba DPH
1	<u>Auto</u>	1	0	200,- Kč	19%
3	<u>Medvěd</u>	1	0	170,- Kč	5%

Poznámka pro zákazníka:

Poslat email: ☒

Obr. 35 Objednávka připravená pro fyzické odeslání zboží

3.8.6 Seznam zákazníků

Další položkou menu Admin volby jsou Zákazníci. Tato volba zobrazí správci seznam zákazníků podle počátečních písmen jmen tak, jak jsou uloženi v databázi (obrázek 36). První stránka platí pro zákazníky s jiným počátečním písmenem, než jsou znaky abecedy a poslední stránka zobrazí kompletní přehled zákazníků. Po kliknutí na jméno zákazníka se zobrazí needitovatelné informace o zákazníkovi tak, jak je zadal při registraci.

Druhou možností jak zobrazit detaily určitého zákazníka je zadání jeho unikátního kódu do vyhledávacího pole Kód Zákazníka a formulář odeslat.

Seznam zákazníků

Kód zákazníka:

[.](#) [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#) [vše](#)

Kód	Jméno zákazníka
1	Lukáš Veselý

Obr. 36 Výpis zákazníků

3.8.7 Ostatní nástroje

Zbývá popsat ostatní nástroje správce. Ty jsou umístěny pod položkou Admin volby Obecné. Zde se také nachází nástroj pro modifikaci horního menu, což bylo popsáno v kapitole 3.8.2.

Jedná se o nástroj změny reklamního obrázku (obrázek 37) v horní oblasti obchodu a definování způsobu platby a doručení (obrázek 38), které obchod nabízí. U změny obrázku je nutné dát velký pozor na rozměry obrázku, pokud bude nový obrázek velikostně výrazně větší, než starý, může dojít k rozhození vzhledu obchodu. Je proto doporučeno používat pokaždé obrázky stejných rozměrů. Platby a doručení je třeba specifikovat ještě před zpřístupněním obchodu zákazníkům, protože jinak je nebudou moci uvádět ve svých objednávkách. Platby a doručení navíc není možné smazat, neboť by mohlo dojít k nežádoucím změnám v objednávkách.

Změna reklamního obrázku

Soubor (*.jpg):

Obr. 37 Změna obrázku reklamy

Přehled způsobů doručení

v ceně: - Kč

v ceně: - Kč

Přehled způsobů platby

Obr. 38 Přidávání a modifikace způsobů platby a doručení

3.9 Generátor internetového obchodu

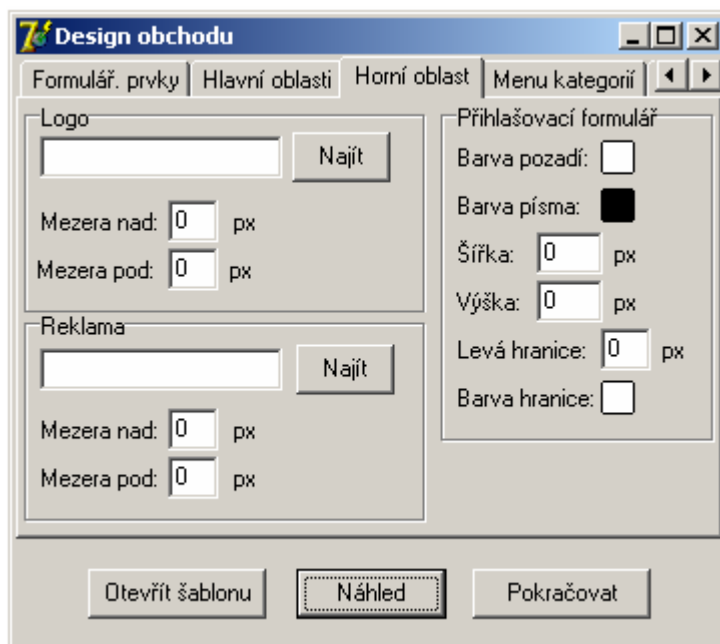
3.9.1 Design

První část aplikace generátor se týká vzhledové variability výsledného obchodu. Vizuální vlastnosti jsou rozděleny do osmi záložek podle příbuznosti vlastností v nich. Jedná se o záložky:

- formulářové prvky
- hlavní oblasti
- horní oblast
- menu kategorií
- seznam produktů
- detail produktu
- tabulky a nadpisy v obsahové části

Ve většině skupin jde pouze o změny barvy písma, hranice a pozadí. Skupina horní oblasti (obrázek 39) je poněkud složitější, protože jsou v ní tři prvky, které se navzájem ovlivňují a to obrázek loga, obrázek reklamy a rámec přihlašovacího formuláře. Program sám vybírá nejvyšší z těchto prvků a jeho výšku za výšku rodičovského kontejneru. Nejlepším řešením je načíst oba obrázky, nechat rozměry přihlašovacího formuláře nulové a použít funkci náhled, která otevře výchozí internetový prohlížeč a zobrazí v něm vzhled stránky podle nastavení v aplikaci. Pokud formulář přesahuje rodičovský rámec,

je třeba zadat rozměry jeho rozměry a zvětšovat je do té doby, dokud nebude design v pořádku.



Obr. 39 Okno vzhledu obchodu, záložka horní oblast

Tato část generátoru nabízí funkci otevřít uloženou šablonu a náhled. První funkce je zřejmá a funkce náhled jsme zmínili výše. Slouží k promítnutí aktuálního nastavení do vzhledu stránky a zobrazení v internetovém prohlížeči. Zde je třeba podotknout, že tento náhled je pouze statická stránka a nemá žádnou funkčnost. Náhled je realizován tak, že aplikace má k dispozici několik šablon html dokumentů, které mají připojenu tabulku stylů. Hodnoty stylů, které se dají aplikací nastavovat, jsou většinou označovány způsobem \$JménoTřídyNázevAtributu. Aplikace pak prohledává šablonu styles_sbl.css a nahrazuje tyto značky hodnotami podle nastavení vzhledu v aplikaci. Výsledkem je soubor stylů styles.css. Stiskem tlačítka pokračovat se dostaneme do části nastavení.

3.9.2 Nastavení

Část aplikace nastavení slouží k nastavení údajů pro přístup do MySQL databáze, připojení k webovému serveru pomocí protokolu ftp a nastavení konstant obchodu. Tyto úlohy mají stejné řešení jako uložení hodnot aplikace pro vzhled obchodu do souboru se styly. Výsledkem je soubor constants.php a db.sql. První soubor obsahuje konstanty obchodu, zatímco druhý soubor je skriptem pro vytvoření tabulek v databázi pro internetový obchod. Okno dále

umožňuje vytvořit výchozí datatypy pro základní neviditelnou kategorii. Tyto datatypy budou přednastaveny při tvorbě nových kategorií v první hladině a můžou tedy mírně urychlit práci. Okno aplikace je na obrázku 40.

The screenshot shows a settings window titled 'Nastavení'. It is divided into several sections. On the left, there is a 'MySql nastavení' section with input fields for 'Hostitel', 'Přihl. jméno', 'Heslo', and 'Databáze'. Below this is the 'Nastavení obchodu' section with fields for 'Title' and 'Currency'. At the bottom left is the 'Údaje o provozovateli' section with fields for 'Jméno', 'Adresa', 'Město', 'PSČ', 'Telefon', and 'E-mail'. On the right side, there is an 'FTP nastavení' section with fields for 'Hostitel', 'Port' (which has the value '21'), 'Přihl. jméno', 'Heslo', and 'Adresář'. Below the FTP section is a 'Základní datové typy' section containing two buttons: 'Přidat typ' and 'Odebrat typ'. At the very bottom of the window, there are two buttons: 'Zpět' and 'Pokračovat'.

Obr. 40 Okno nastavení obchodu

3.9.3 Dokončení tvorby obchodu

Dokončení tvorby obchodu se uskuteční stiskem tlačítka upload. Předtím je ovšem vhodné využít možnost uložení šablony tlačítkem uložit. Aplikace se pomocí ftp protokolu připojí na webový server a umístí tam obsah adresáře data – zde je uložen php kód internetového obchodu. Dále umístí na web server vygenerované soubory, zejména soubor kaskádních stylů, konstant a sql skript vytvářející tabulky. Tento sql skript je nutné spustit, k čemuž slouží php skript initialize_db.php. Stačí tedy standardním způsobem zadat jméno skriptu s jeho kompletní cestou, které je závislá na url adrese webového serveru a dalším členění webového prostoru, do internetového prohlížeče.

3.10 Práce s databází pomocí SQL a PHP

Jako příklad ukládání dat do databáze uvádím funkci pro vytvoření objednávky zákazníkem.

```
function create_order($customerid, $productids, $amounts, $note, $deliveryid,
                    $paymentid, $delivery_address, $delivery_city, $delivery_psc)
{
    //vytvorí v DB záznamy o objednávce
    $date = getdate();
    $datetime = $date['year']."-".$date['mon']."-".$date['mday']." ".$date['hours'].
        ":".$date['minutes'].":".$date['seconds'];

    $conn = db_start();
    $query = "insert into customer_orders(customerid, orderdate, ordernote,
        deliveryid, paymentid, delivery_address, delivery_city, delivery_psc)
        values('".$customerid."', \"\".\"$datetime.\"\", \"\".\"$note.\"\", \"\".\"$deliveryid.\"\",
        \"\".\"$paymentid.\"\", \"\".\"$delivery_address.\"\", \"\".\"$delivery_city.\"\", \"\".\"$delivery_psc.\"\"");
    $result = mysql_query($query);
    if(!$result)
    {
        db_rollback();
        return false;
    }
    $orderid = mysql_insert_id();

    for($i = 0; $i < sizeof($productids); $i++)
    {
        $query = "insert into orders(orderid, productid, amount) values('".$orderid.
            "\", \"\".\"$productids[$i].\"\", \"\".\"$amounts[$i].\"\"";
        $result = mysql_query($query);
        if(!$result)
        {
            db_rollback();
            return false;
        }
    }

    db_commit();
    return $orderid;
}
```

Funkce *create_order* nejprve zjistí aktuální datum. Zahájí transakci a vloží do tabulky *customer_orders* záznam o objednávce. Tím dojde k automatickému vytvoření hodnoty pole *orderid*, protože toto pole je nastaveno na AUTO INCREMENT. Pole *customerid* je přitom cizí klíč ukazující do tabulky *customers* a *orderid* je primární klíč tabulky. Hodnotu nového *orderid* zjistí funkce PHP *mysql_insert_id()*. Tuto hodnotu budeme potřebovat na propojení tabulky *customer_orders* s tabulkou *orders*, která obsahuje záznamy o produktech v databázi. Následuje cyklus pro vložení právě těchto záznamů o produktech v objednávce do databáze *orders*. Pokud nedošlo v průběhu funkce k chybě, ukončí se transakce a data se uloží do databáze.

Závěr

Úkol seznámit se s aktuálními trendy vývoje webových prezentací a technologiemi umožňujícími využití databází jsem zvládl dobře. Největší potíže při tvorbě práce mi způsobovaly kaskádní styly. Nebylo to ovšem neschopností pochopit kaskádní styly, ale problém byl zapříčiněn chybnou interpretací kaskádních stylů Internet Explorerem. Styly funkční ve Firefoxu se zobrazovali v Internet Exploreru chybně a musel jsem často hledat náhradní řešení. V této chvíli jsem pochopil, že programování webových aplikací vyžaduje velkou míru praxe a komplexní znalost všech standardních specifikací a postupů.

Nicméně se mi podařilo úspěšně vyvinout internetový obchod schopný nasazení do praxe. Navíc tento obchod může teoreticky pojmout kterýkoliv produkt díky použití datatypů produktu. Zde je na místě diskuze, zda je výhodné vyvíjet obecná řešení. Obecné řešení je složitější na logiku kódu a zároveň paradoxně méně možností oproti řešení pro konkrétní zadání. Jeho výhodou je možnost nasadit ho na velké množství různých obchodů. Vzhledovou stránku systému jsem vyřešil pomocí kaskádních stylů a tento vzhled je zobrazen téměř stejně v prohlížečích Internet Explorer, Mozilla Firefox i Opera.

Dále jsem naprogramoval aplikaci vytvářející jednoduché html dokumenty, které jsou ukládány na webový server a zobrazovány na požádání zákazníků. Jelikož se jedná o html dokument, jeho obsah může být velmi různorodý.

Posledním krokem diplomové práce bylo vyvinout aplikaci, která bude generovat obchod tak, že umožní vytvořit různé vzhledové varianty obchodu a umístí je na server. Požadavek na tuto aplikaci byl, aby ji mohl používat člověk, který neví nic o tom, jak se takovéto webové aplikace vyvíjejí. Vyřešení této úlohy hodnotím jako úspěšné.

Seznam literatury

- [1] STANÍČEK, Petr: *CSS Kompletní průvodce*. 2. vyd. Brno: Computer Press, 2003. 178 s. ISBN 80-7226-872-4.
- [2] *Full Web Building References*. [online]. <<http://www.w3schools.com>>.
- [3] WELLING, Luke, THOMSONOVÁ, Laura: *PHP a MySQL – rozvoj webových aplikací*. 2. vyd. Praha: SOFTPRESS, 2003. 910 s. ISBN 80-86497-60-7.
- [4] LISCHNER, Ray: *Delphi v kostce*. 1. vyd. Praha: Computer Press, 2000. 550 s. ISBN 80-7226-361-7.
- [5] CÍSAŘOVÁ, Klára: *Studijní materiály k předmětu RDB*. [online]. TU Liberec, 2003. <<http://www.fm.vslib.cz/~ksi/cz/mater/rdb/Texty.zip>>.
- [6] KOSEK, Jiří: *PHP – Tvorba interaktivních internetových aplikací*. 1. vyd. Praha, 1998. 492 s. ISBN 80-7169-373-1.
- [7] SVOBODA, Luděk a kol.: *1001 tipů a triků pro Delphi*. 2. vyd. Praha: Computer Press, 2002. 390 s. ISBN 80-7226-529-8.
- [8] HLAVENKA, Jiří a kol.: *Vytváříme WWW stránky*. 4. vyd. Praha Computer Press, 2000. 520 s. ISBN 80-7226-293-9.
- [9] KRUG, Steve: *Web design – Nenutíte uživatele přemýšlet!* 1. vyd. Praha: Computer Press, 2003. 144 s. ISBN 80-7226-892-9.
- [10] STEPHENS, R. K., PLEW, R. R.: *Naučte SQL za 21 dní*. 1. vyd. Praha: Computer Press, 2004. 581 s. ISBN 80-7226-870-8.