

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

E-SHOP PRO PRODEJ STAVEBNÍCH MATERIÁLŮ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

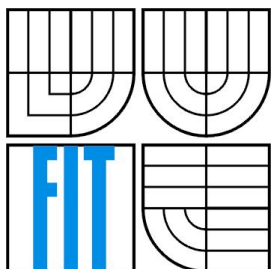
AUTOR PRÁCE
AUTHOR

FRANTIŠEK CZIBOR

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

E-SHOP PRO PRODEJ STAVEBNÍCH MATERIÁLŮ

E-SHOP FOR SELLING OF BUILDING MATERIALS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

FRANTIŠEK CZIBOR

VEDOUCÍ PRÁCE
SUPERVISOR

ING. ŠÁRKA KVĚTOŇOVÁ

BRNO 2007

Zadání bakalářské práce

Řešitel: **Czibor František**
Obor: Informační technologie
Téma: **E-shop pro prodej stavebních materiálů**
Kategorie: Elektronický obchod

Pokyny:

1. Seznamte se s požadavky kladenými na tvorbu dynamických webových stránek.
2. Podrobně analyzujte požadavky na internetový obchod pro prodej stavebních materiálů.
3. Seznamte se s databázovým systémem MySQL a skriptovacím jazykem HTML, PHP, JavaScript a s možnostmi jejich využití při implementaci systému.
4. Systém realizujte a jeho funkčnost ověřte na vhodném vzorku testovacích dat.
5. Zhodnoťte dosažené výsledky a diskutujte další možný rozvoj systému.

Literatura:

- Williams, H. E., Lane, D.: PHP a MySQL - Vytváříme webové databázové aplikace. Computer Press, 2002, 552 s. ISBN 8072267604
- Kosek, J.: HTML, tvorba dokonalých www stránek. Praha: Grada Publishing, 1998, 291 s. ISBN 80-7169-608-0
- PHP: Hypertext Preprocessor. Dostupné na: www.php.net
- DeLisle, M.: PHPMyAdmin - efektivní správa MySQL. Brno: Zoner Press, 270 s. ISBN 8086815099
- Ullman, L.: PHP a MySQL. Computer Press, 2004, 536 s. ISBN 8025100634

Při obhajobě semestrální části projektu je požadováno:

- Body 1-3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Květoňová Šárka, Ing., UIFS FIT VUT**
Konzultant: Štefánik Štefan, PIPO
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, B. S. Těchova 2



doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **František Czibor**
Id studenta: 84433
Bytem: Mužla 403, 943 52 Mužla
Narozen: 19. 02. 1985, Nové Zámky
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: E-shop pro prodej stavebních materiálů
Vedoucí/školitel VŠKP: Květoňová Šárka, Ing.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracování díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.


Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel



.....

Autor

Abstrakt

Tato bakalářská práce se zabývá tématem vytváření informačních systémů. Po jejím úvodu do informačních systémů shrnuje jednotlivé kroky, postupy, při jejich vytváření. Cílem bylo vytvořit po důkladné analýze detailní návrh, pak postoupit k implementování jednotlivých částí systému, a vytvořit funkční informační systém pro internetový obchod. Systém byl implementován pomocí technologií PHP, relační databáze MySQL a skriptovacího jazyka JavaScript.

Klíčová slova

Informační systém, elektronický obchod, databáze, uživatelské rozhraní, UML, PHP, MySQL, JavaScript, HTML, diagram případu užití, ER diagram, kardinalita vztahu, primární klíč, cizí klíč

Abstract

The bachelor's thesis deals with creation of information systems. After the brief introduction to the information systems, the thesis summarises the processes of their creation. After carrying out a thorough analysis, the aim was to form a detailed proposal of an information system, which was followed by the implementation of its particular parts, and then finalised by creation of a functional information system for internet shopping. The system was implemented by the following technologies: PHP, a relational database MySQL, and JavaScript.

Keywords

Information system, electronic business, database, user interface, UML, PHP, MySQL, JavaScript, HTML, use case diagram, ER diagram, cardinality relation, primary key, foreign key

Citace

František Czibor: E-shop pro prodej stavebních materiálů, bakalářská práce, Brno, FIT VUT v Brně, 2007

E-shop pro prodej stavebních materiálů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Šárky Květoňové
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení
Datum

Poděkování

Na tomto místě bych chtěl poděkovat mému vedoucímu Ing. Šárce Květoňové za odborné vedení, za mnoho užitečných rad, připomínek, za konzultace, které mi pomohly při řešení bakalářské práce.

© František Czibor, 2007.

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	1
1 Úvod.....	3
2 Informační systém.....	4
2.1 Co jsou informační systémy	4
2.2 Architektura informačních systémů	4
2.3 Vytváření informačních systémů.....	5
2.3.1 Životní cyklus informačního systému.....	5
2.4 Elektronický obchod	8
2.4.1 Význam pojmu elektronický obchod.....	9
2.4.2 Výhody a nevýhody nakupování na internetu.....	9
3 Implementační prostředky.....	11
3.1 Jazyk UML.....	11
3.2 Databáze MySQL.....	11
3.3 Serverové skriptování PHP	12
3.4 XHTML s použitím kaskádových stylů CSS a JavaScript.....	12
4 Analýza	14
4.1 Prohlášení o cílech uživatele	14
4.2 Požadavky na internetový obchod.....	15
4.2.1 Jednotlivé funkce systému	16
4.3 Modely navrhovaného systému.....	17
4.3.1 Use Case diagram systému	17
4.3.2 ER diagram systému	19
5 Implementace	21
5.1 Tvorba databází a tabulek.....	21
5.2 Implementované funkce systému	23
5.2.1 Registrace uživatele	23
5.2.2 Zapomenuté heslo	24
5.2.3 Přihlášení do systému a odhlášení ze systému.....	25
5.2.4 Funkce a rozdělení uživatelského rozhraní	27
5.2.5 Nabídka zboží – výpis produktů	30
5.2.6 Funkce košíku	31
5.2.7 Funkce pokladna	33
5.2.8 Diskuze – názory k produktům	34
5.2.9 Zákaznický profil.....	36

5.2.10	Administrátorské rozhraní	38
6	Další rozvoj systému	43
7	Závěr	44
	Literatura	45
	Seznam příloh	46

1 Úvod

V dnešní době, kdy internet hraje významnou úlohu v životě člověka, je elektronické obchodování velice důležitou, nedílnou součástí úspěšných firem a společností, prodávajících své produkty. Když člověk jednou vyzkouší pohodlí nákupu přes internet, tak i příště bude bez pochyby sahat po této možnosti nakupování. Tak se postupně vyvíjejí i e-shopsy, s tím i informační systémy, které mají jako jeden ze svých úkolů pomáhat a zjednodušovat lidem jejich činnosti.

Druhá kapitola jako úvodní, nás seznámí co vlastně můžeme chápat pod pojem informační systém, jak ho definovat. Jsou zde popsány i důležité pojmy související s informačním systémem a jak vypadá schéma informačního systému. Dále jsou uvedeny vrstvy architektury informačních systémů, a popsány jakými etapami projde jeho vytváření, a jeho životní cyklus z různých pohledů. Kapitola obsahuje i stručný přehled historie informačních systémů. Závěrem jsem se zmínil i o nejdůležitějších věcech souvisejících s elektronickým obchodováním.

Pro implementace celého systému jsou použity různé implementační nástroje a prostředí. Tyto nástroje jsou popsány s výhodami a nevýhodami v třetí kapitole. Implementační prostředí jsem se rozhodl rozdělit dle architektury informačních nástrojů. Na různých vrstvách jsou použity různé implementační nástroje, jazyky. Na datové vrstvě databázový systém MySQL, na aplikační vrstvě technologie PHP, a na klientské vrstvě jazyk XHTML s kaskádovými styly a JavaScript.

Ve čtvrtém kapitole je čtenář seznámen s analýzou a návrhem daného systému. V první části jsou shrnuty požadavky na systém, co by měl obsahovat, jaké funkce by měl mít. Jsou zde uvedena prohlášení o cílech, které jsou stručným a obecným popisem představ o tom, co by měl systém dělat. Z obchodního hlediska je prohlášení o cílech základním materiálem vznikajícím u uživatele, kde zcela neformálně popíše, co hodlá automatizovaně pomocí informačního systému zpracovávat. V druhé části se kapitola zabývá návrhem systému, a modely použitými při návrhu. Jsou zde rozepsány Use Case diagram a ER diagram navrhovaného systému.

Významná část technické správy sestává z implementace systému. V této kapitole jsou popsány implementace jednotlivých částí systému. Popis implementace jsem se snažil rozčlenit dle vrstvy architektury IS. Začal jsem s implementací databázové struktury systému. Poté, v druhé části kapitoly jsem popsal jednotlivé implementované části systému. Je to rozčlešené na popisy implementace jednotlivých funkcí v systému, v kterých nejprve popíšu implementace uživatelského rozhraní, a pak způsob spojení uživatelské úrovně s aplikační úrovní a jak jsou na této úrovni zpracována data z databáze. Mezitím popíšu i použité techniky a řešení, zabezpečení systému a databáze.

V předposlední kapitole jsem shrnul možné rozšíření systému do budoucna, jak by mohly být ještě jednotlivé funkce systému zlepšeny.

2 Informační systém

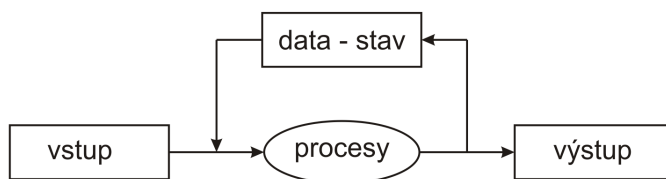
Kapitola obsahuje základní informace o informačních systémech, jejich návrhu, architektuře a implementaci. Po obecném popisu informačních systémů, seznámíme čtenáře s architekturou IS, která je základem správného pochopení jejich fungování. Pak se z kapitoly dozvíme jak se tyto systémy vytvářejí a jak vznikají, jaký je jejich životní cyklus z různých pohledů. Další podkapitola je jemným přehledem historie informačních systémů, a závěrem jsem zmínil základní záležitosti týkající se elektronických obchodů.

2.1 Co jsou informační systémy

Informační systémy (IS) jsou systémy pro sběr, udržování, zpracování a poskytování informací a dat. Příkladem jednoduchého informačního systému může být např. telefonní seznam. Určitě je potřebné aby jsme tato data s kterými IS pracuje, mohli někde na spolehlivém místě ukládat. Na to slouží databáze, které úzce souvisejí, a tvoří nedělitelnou část informačních systémů.

Informacemi rozumíme data, po jejich interpretaci s uživatelem. Tedy různá data z databáze informačních systémů, slouží pro interpretace uživatelem jako informace pro ně. Data jsou jakákoli vyjádření skutečnosti, schopné přenosu, uchování, interpretace či zpracování. Důležitý je ještě pojem znalost, související s informačním systémem. Znalosti jsou informace pro jejich zařazení do souvislostí. (zdroj: literatura [2])

Obecný systém se skládá ze vstupní a výstupní části, mezi kterými probíhá transformace těchto zdrojů. Informační systém (Obrázek 2.1) je velice podobný, taky se skládá ze vstupní a výstupní části, kudy se do systému data vstupují, resp. získávají informace. Mezi vstupem a výstupem je transformační část, která typicky provádí nad daty různé operace. Probíhají zde procesy.



Obrázek 2.1: Schéma informačního systému

(zdroj: literatura [2] Prof. Ing. Tomáš Hruška, CSc – IIS Studijní opora)

2.2 Architektura informačních systémů

Velmi důležitým hlediskem je architektura IS. Dělí se na 3 vrstvy, které jsou následující:

- klientská vrstva – která slouží jako interakce s uživatelem. Je to vlastně uživatelské rozhraní systému, přes kterou uživatel komunikuje se systémem.

- aplikační vrstva – rozumíme vlastní aplikace, která řeší různé funkce celého systému, operace nad daty, uložení v databázích, podle vstupních údajů a parametrů od uživatelů přes uživatelské rozhraní. Dále řeší i vlastní propojení mezi uživatelským rozhraním a databázovým systémem, propojení se sítí a po případě celým světem. Zabezpečuje bezpečné fungování systému a také různé kontroly.
- datová vrstva – jsou vlastní data uložená v databázích. Na této úrovni jsou řešeny vlastní přístupy k datům, realizované pomocí SŘBD (Systém Řízení Báze Dat)

Většina systémů se implementuje jako tzv. Data Warehouses (DW), což je architektura obvykle založená na SŘBD, jež transformuje operativní data do jiné podoby, u které se bere ohled například na čas a rychlost následných dotazů. Tato data se nemění, mohou se transformovat z více zdrojů a jsou aktualizována v časových intervalech. Nad nimi se dělají statistiky či analýza. To je poslední fáze OLAP (Online Analytical Processing).

Opakem DW jsou OLTP (Online Transaction Processing Systems), které jsou přirovnávány k „výrobě“ podniku, DW pak ke „skladování“ výrobků, následně OLAP systémy jsou pak jakýmsi „prodejem“. Je zřejmé, že OLAP systémy jsou rozšířením OLTP systémů, také jejich návrh je složitější. Je u nich použita tzv. multidimenzionální architektura.

2.3 Vytváření informačních systémů

Postup vytváření informačních systémů má několik úrovní. Jsou čtyři základní fáze vzniku informačních systémů které jsou následovné:

- formulace požadavků na úrovni fyzického systému – tj. vytvoření požadavků na systém, což znamená že specifikujeme komunikační prostředí
- datová a funkční analýza – analyzujeme požadavky na systém, vytvoříme konceptuální schéma
- návrh informačního systému – shrnuje převod konceptuálního schématu na databázový model, definice procesů
- implementace informačního systému – znamená vlastní implementace procesů, komunikace klient server a uživatelského rozhraní

Po implementaci, a vlastním provozu ještě můžeme definovat jednu úroveň, která by shrnovala další vylepšení, a modifikace na základě zkušeností v prvních fázích provozu.

2.3.1 Životní cyklus informačního systému

Životní cyklus IS je v produktovém chápání tvořen životními fázemi popisujícími jeho realizace jako produktu od začátku (výzkum) do konce (pokles, zánik produktu). Počet vývojových fází IS je podle různých autorů různý. Většina autorů nepoužívá kompletní marketingový šestifázový vývoj, ale v obvyklé nejstručnější podobě uvádí čtyři fáze (zahájení, růst, dospělost, ústup). Jiní autoři

upřednostňují nikoliv produktové hledisko, ale životní cyklus vývoje IS přísně odvozují od vlastního vývoje produktu (vlastní softwarové aplikace), nikoliv jeho setrvání na trhu. Jako příklad tohoto přístupu uvádíme práce a následující publikované vývojové fáze IS dle Kendalla: (zdroj: lit. [4])

- identifikace problémů, možností a cílů (stanovení strategie vedoucí k realizaci tvorby IS),
- definování informačních potřeb (proč má produkt vzniknout a jaké informační potřeby uspokojí)
- analýza systémových potřeb (zkoumáme systémové potřeby nezbytné pro tvorbu vlastního IS)
- návrh doporučeného systému (navrhujeme, jak bude systém řešen v dané technologii a jaké postupy budou uplatněny),
- vývoj a dokumentace softwaru,
- testování a zavádění softwaru,
- údržba a hodnocení systému.

Nutné je zdůraznit, že vlastní důkladná studie výše uvedených zdrojů prokázala, že neexistuje jednotný přístup k definování a chápání základního pojmu životní cyklus vývoje IS. Někteří autoři jasně svazují životní cyklus IS s metodikou IS. Dokonce do představy životního cyklu IS odvíjí a k ní váží veškerý obsah metodiky IS. Zároveň zužují pohled na životní cyklus IS pouze na životní cyklus vývoje IS. Dle J. Poláka v literatuře [3] *Umění systémového návrhu* má pak vývojový cyklus fáze zadání, analýzy, návrhu, implementace, testování a provozu. Fáze zadání a analýzy pak dohromady označuje tento autor jako stádium expanze, protože při nich dochází ke hromadění informací potřebných pro vytvoření aplikace. Stádium expanze končí rozpracováním analytického konceptuálního modelu, který na logické úrovni reprezentuje požadované zadání a formálně popisuje řešený problém. Zbývající fáze od návrhu k provozu se označuje jako stádium konsolidace. Je to proto, že se v těchto etapách model, který je produktem předchozí expanze, postupně stává fungujícím produktem, programem.

Životní cyklus IS a metody vývoj IS

Z hlediska analýzy vývoje informačních systémů lze v kontextu životního cyklu vymezit konkrétní metody a přístupy k problematice vývoje informačních systémů.

Metoda tunel

Její princip je velmi jednoduchý. Se spuštěním projektu se postupuje do neznáma – tunelu. Podle momentální situace vedení projektu řídí jeho postup a vytváří silný tlak na realizaci projektu. Úkolem je nalézt, identifikovat konec tunelu a projekt zdárně ukončit. Projekt je řízen pouze operativně a nikoliv s koncepcí. Pracovníci zapojení do projektu většinou řeší úkol po celou dobu životního cyklu projektu. Z hlediska řízení projektu je projekt řízen v průřezové oblasti úkolů. Na obrázku 2.2 je graficky znázorněná metoda tunel. (zdroj: literatura [9])



Obrázek 2.2: metoda řízení projektu tunel

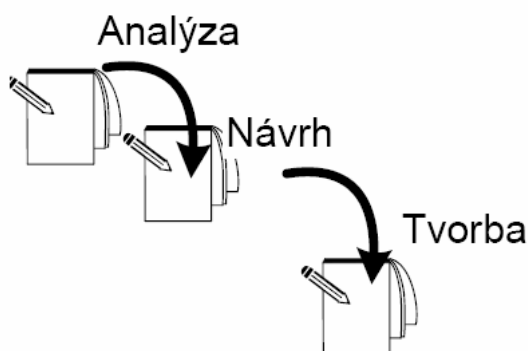
(zdroj: literatura [9] Vladimír Krajčák – Životní cyklus projektu IS)

Vyhodnocení metody:

- chybí koncepce vývoje a není možná opakovatelnost dobrých výsledků,
- nemožní jakákoliv predikce v projektu,
- neexistuje opakovatelných postupů (tj. obecná neznalost, kdy má řešitel provádět danou aktivitu),
- nízká transparence výsledků (nelogičnosti, chyby, nepřehlednost),
- vysoký stupeň chaosu ve vývoji,
- vznik softwarových nedodělků,
- nízká kvalita výstupu projektu – aplikačního softwaru

Metoda vodopád

V této metodě je obsažena následující koncepce vývoje IS a životního cyklu projektu. Metoda je velmi pečlivě zpracována mnoha autory, osobně jsem čerpal z práce Iljy Kravala: *Zásady pro řízení projektů tvorby IS*. (zdroj: literatura I5) U tohoto autora si cením jeho projektový pohled. Projektem je



Obrázek 2.3: Metoda řízení projektu typu vodopád

(zdroj: literatura [9] Vladimír Krajčák – Životní cyklus projektu IS)

dáno rozdělení na jednotlivé části vývoje IS, které jsou dobře definovány a diskrétně rozděleny do konkrétních časových úseků a etap vývoje. Jednotlivé etapy jsou ukončeny milníky, které vyhodnocují realizaci kompletní etapy. Výstupy z těchto etap jsou natolik závazné, že zpravidla neumožňují další podstatné změny v průběhu řešení. Vznikají tak postupně analytické dokumenty, design dokumenty a nakonec dokumenty kódování (zdrojový kód). Protože se dokumenty předávají z jedné fáze do druhé odshora dolů, nazýváme tuto metodu vodopád. Tento vodopádový model je znázorněna na obrázku 2.3.

Vyhodnocení metody

Je v celku pochopitelné, že tato metoda je mnohem lepší, než předešlá chaotická metoda tunel, ale má také své nevýhody. Největší nevýhodou je podle mě její těžkopádnost a obtížné řízení lidských zdrojů. Metoda je zatížena i dalším nedostatkem. Nejdříve musí vzniknout celá analýza, potom celý design a poté celé kódování. Uvedená metoda také naráží při své aplikovatelnosti na překážky vyplývající z faktu, že je nutné postupně precizovat navrhovaný informační systém. Mnohdy tento fakt vyplývá i ze záměru manažera projektu ovlivňovat postoje budoucích uživatelů. Mnohých změn mohou dostat i požadavky uživatelů. Nejde jen o prostý fakt, že analytik neprovede kvalitně analýzu požadavků na systém, ale můžeme připustit i skutečnost, že předmět analýzy je doposud natolik nový a neprozkoumaný, že již dopředu předpokládáme poznávání reality a požadavků po částech v časovém vývoji. (zdroj: literatura [9])

Metoda přírůstkového postupu (iteračně – inkrementální metoda)

Hlavní myšlenka této metody je velmi prostá. Vedoucí projektu organizuje vývoj systému tak, aby rozložil složité procesy na menší a jednodušší problémy a vlastně tak řešil vývoj IS prostřednictvím menších projektů. Každý ze zmiňovaných menších projektů je považován za postup, přírůstek, iteraci.

Přírůstkem se rozumí ucelená a relativně uzavřená část systému (subsystém), kterou lze samostatně navrhnout, implementovat a uvést do provozu, přičemž funkčnost dříve dokončené části systému zůstane zachována. Podstatou této metody je tedy provedení všech kroků v jedné relativně malé iteraci v jedné relativně malé části systému. Systém je pak vyvíjen postupně po jednotlivých přírůstcích tak, že každý přírůstek probíhá svým vlastním životním cyklem. Tímto postupem se může docílit, že některé pevné segmenty jsou rychle zpracovány až do podoby implementace, přičemž některé jiné části systému zůstávají v úvodních stupních vývoje IS. Metoda je vlastně složení několika vodopádů částí systémů.

Vyhodnocení metody

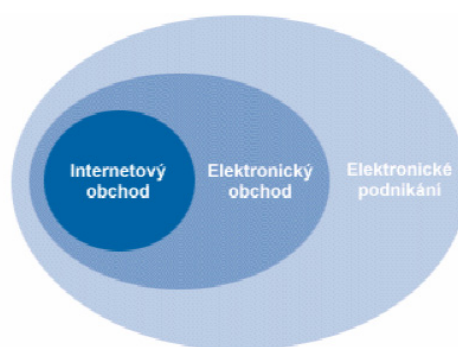
Uvedená metoda je velmi vhodná v objektovém orientovaném prostředí. Základní nepochopení metody spočívá v tom, že rozšiřování systému v inkrementaci nespočívá pouze v přidávání prvků a vazeb, ale v přidávání funkcionalit objektů spolu s novým stále konzistentními vnitřními stavy objektů. (zdroj: literatura [9])

2.4 Elektronický obchod

Pro internetové obchody existuje několik různých pojmenování. Elektronický obchod, virtuální obchod, internetový obchod, on-line obchod nebo e-shop. Avšak tyto názvy nejsou úplně jednotné a ekvivalentní. (zdroj: literatura [10])

2.4.1 Význam pojmu elektronický obchod

Elektronickým obchodem nazýváme takovou realizaci obchodních procesů, která je uskutečněna s využitím elektronických komunikačních prostředků, v dnešní době zejména prostřednictvím internetu. Pro elektronický obchod se vedle internetu využívá celá řada dalších elektronických komunikačních prostředků, například tzv. sítí s přidanou hodnotou (Value Added Network – VAN), hlasových i datových služeb. Pojem elektronický obchod se proto nerovná pojmu internetový obchod, jak se někdy nepřesně uvádí. Elektronický obchod – Electronic Commerce (E – Commerce) je součástí širší oblasti elektronického podnikání – Electronic Business (E – Business), jež představuje využití elektronických komunikačních prostředků ve všech aspektech podnikatelské činnosti. Znázornění souvislosti těchto pojmů vidíme na obrázku 2.4.



Obrázek 2.4: Schéma vztahu mezi pojmy elektronické podnikání, elektronický obchod a internetový obchod

(zdroj: literatura [9] Ing. Radek Foxlík – Elektronický obchod)

2.4.2 Výhody a nevýhody nakupování na internetu

Internetové obchody u nás fungují již několik let. Počet uživatelů internetu (a tedy i potenciálních zákazníků) dosáhl potřebného kritického množství, padají psychické bariéry z nákupů ve „virtuálních obchodech“, vyřešeny jsou i některé otázky zabezpečení plateb a obchodního styku vůbec. V České i Slovenské republice již existují firmy, které prostřednictvím elektronického obchodu realizují významnou, ne-li hlavní část svého obrátu. U určitých druhů zboží již začíná klasický obchodní model zaostávat. Elektronický obchod nabízí zákazníkům mnohem širší nabídku než „kamenný obchod“ a navíc s přidanou hodnotou. Například dodání až do domu (kanceláře), výhodnější cena, větší dostupnost informací o zboží a podobně.

S rostoucím rozvojem elektronického bankovníctví si stále více zákazníků oblíbilo tento způsob nakupování.

Výhody ze strany zákazníka jsou nasnadě:

- **Ušetřený čas.** Do virtuálního obchodu může zákazník přijít třeba v sobotu o půlnoci. Elektronický obchod nikdy nezavírá.

- **Nezanedbatelná finanční úspora.** Protože obecně platí, že elektronické obchody své zboží neprodávají za ceny vyšší než obchody kamenné (úspora při nákupu na internetu se v průměru pohybuje mezi deseti až patnácti procenty).
- **Možnost dohledání všech dostupných informací o zboží.** Na rozdíl od kamenných obchodů, kdy je zákazník odkázán na informace od zaměstnanců (mnohdy nedostatečně proškolených či neochotných)
- **Pohodlnost při nákupu.** Při elektronickém nakupování člověk snadno u počítači vyřizuje mnohdy nepříjemné nakupování, shánění zboží apod.

Nevýhody:

- V mnohých případech stále nedokonalý způsob placení (obvykle zaslání zboží na dobírku nebo převodem z účtu)
- Nemožnost si zboží hned po nákupu odnést a způsob doručování zboží (kurýrní služby)
- Není také možné si zboží prohlédnout, osahat, vyzkoušet, přivonět...
- Anonymita prodeje

U elektronického obchodu je třeba se zmínit ještě o dvou základních druzích elektronického obchodu.

B2B – Business to Business – Tento druh elektronického obchodování se používá v distribučních a prodejních sítích, ve kterých mohou mezi sebou komunikovat výrobci, pobočky, distributoři, velkoobchody, dealeři, nebo obchodní zástupci. Základní rozdíl mezi tímto druhem elektronického obchodu a internetovým obchodem typu B2C je v tom, že prodávající (výrobce, distributor, velkoobchod, apod.) zná předem nakupujícího.

B2C – Business to Consumer – Tento druh elektronického obchodování je zaměřen na prodej koncovým zákazníkům – spotřebitelům. Je to vlastně obdoba klasického "kamenného" obchodu na Internetu.

Tato kapitola je jako úvodní kapitola mé práce, v které je definováno co vlastně informační systém je, co znamenají důležité pojmy s ním související. Byl zde uveden stručný popis architektury informačních systémů, postupy při vytváření IS, vlastní životný cyklus IS z různých pohledů, a také stručný přehled historií informačních systémů, a na závěr stručný přehled internetových obchodů.

3 Implementační prostředky

Implementační prostředky můžeme rozdělit dle jednotlivých architektur informačního systému, jak je to popsáno v kapitole 2.2. Databázová vrstva je implementována do technologií MySQL. Na aplikační vrstvě jsou používány technologie jako webový server Apache, pro přístup datům v databázích MySQL a pro dynamičnost webových stránek skriptovací jazyk PHP. Na vrstvě klientské pro zobrazení dat a komunikace s uživatelem jsem používal technologie XHTML, s použitím kaskádových stylů CSS a skriptovací jazyk JavaScript.

Implementace systému ještě předcházela návrh. Při návrhu informačního systému a v databázové struktuře jsem použil vhodné technologie UML, v které jsem vytvořil i příslušné diagramy.

3.1 Jazyk UML

Během vývoje metodik softwarového inženýrství bylo navrženo mnoho různých víceméně formálních jazyků a prezentačních technik, které slouží pro popis softwarových produktů od konceptuálního modelu až po jeho implementaci. Nová generace metodik softwarového inženýrství se snaží sjednotit užitečné vlastnosti různých metodik a integrovat je do nějaké společné sady. Jedním z nejdále popracovaných přístupů je tzv. unifikovaný modelovací jazyk UML (Unified Modeling Language).

Při komunikaci mezi zadavatelem a řešitelem vznikají různé modely navrhovaného systému. Požívané modely mohou být různých typů. Obecně lze model softwarového systému definovat jako souhrn informací, které jsou určitým způsobem strukturovány. Model by měl obsahovat veškeré shromážděné informace o systému.

Statickou strukturu systému vyjadřují diagramy tříd, diagramy spolupráce, diagramy komponent a diagram nasazení. Funkční stránku popisují model jednání, diagramy aktivit, scénáře událostí a diagramy spolupráce. Dynamickou stránku dokumentují stavové diagramy, scénáře událostí, diagramy spolupráce a diagramy aktivit.

Při návrhu systému jsem vytvořil diagram modely jednání (use case diagram) a diagram tříd – class diagrams (ER diagram). Tyto diagramy jsou uvedeny a popsány v kapitole 4 analýza.

3.2 Databáze MySQL

MySQL je relační 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. Vychází z deklarativního programovacího jazyka SQL (Structured Query Language).

Díky své licenci a rychlosti je v poslední době téměř nejoblíbenějším systémem. MySQL je malý, rychlý a jednoduchý databázový systém. Databáze MySQL má některá omezení, která obsahují jiné databázové systémy, např. robustní Oracle. Právě díky tomu dosahuje vynikající rychlosti.

Databáze MySQL je jeden z prvních hojně rozšířených systémů. Práce s tímto systémem se dá využít v C, C++, Java, Perl, PHP, Python, Tcl, Visual Basic nebo .NET.

Pro jednoduchou správu MySQL databází se používá nástroj PhpMyAdmin. PhpMyAdmin je Open Source program napsaný v PHP, který umožňuje zálohování, vytváření tabulek, vkládání, editaci a mazání záznamů v tabulkách, vytváření databází apod. PhpMyAdmin je pokročilý nástroj pro kompletní správu MySQL systému přes webové rozhraní.

3.3 Serverové skriptování PHP

PHP Hypertext Preprocessor je skriptovací programovací jazyk běžící na straně serveru (server-side), určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, což je velmi výhodné pro tvorbu webových aplikací. PHP lze ovšem také použít i k tvorbě konzolových a desktopových aplikací. PHP je Open Source.

Počátky PHP spadají do roku 1994 kdy jej vytvořil Rasmus Lerdorf. PHP 3.0 vytvořil Andi Gutmans a Zeev Suraski. PHP – Hypertext Preprocessor. PHP 4.0 bylo oficiálně uvolněné v roce 2000. Je znatelně výkonnější než PHP 3, podporuje více web serverů, systém sessions, buffering výstupu atd. V červnu 2003 byla oficiálně uvolněna beta verze PHP 5.

PHP skripty jsou prováděny na straně serveru, k uživateli je přenášen až výsledek jejich činnosti. Syntaxe jazyka kombinuje hned několik programovacích jazyků (Perl, C, Pascal a Java). PHP je nezávislý na platformě, skripty fungují bez úprav na mnoha různých operačních systémech. Obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových serverů (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL), podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP, ...)

Při implementaci mého systému jsem použil verzi PHP 4.

3.4 XHTML s použitím kaskádových stylů CSS a JavaScript

XHTML nebo HTML ? V čem je vlastně rozdíl ? HTML je zkratka z anglického Hyper Text Markup Language, značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci stránek na Internetu. Jazyk je podmnožinou dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

XHTML (zkratka anglického extensible hypertext markup language – „rozšiřitelný značkovací jazyk pro hypertext“) je značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW vyvinutý konsorciem W3C. Je následníkem jazyka HTML, jehož vývoj byl ukončen, a na rozdíl od svého předchůdce se jedná o aplikaci XML.

XHTML má velice přísná a zároveň jednoduchá pravidla. Díky tomu mohou počítače XML a tedy i XHTML velmi snadno automatizovaně zpracovávat. Časem se dá ale očekávat, že stránky vytvořené v XHTML budou lépe indexované a tím pádem i více navštěvované.

Dnes jako nástupce již zastaralého jazyka HTML je moderní značkovací jazyk XHTML, který existuje ve dvou verzích. První je XHTML 1.0, dělicí se ještě na tři další varianty:

- XHTML 1.0 Strict – čistě strukturální značkování, neobsahuje žádné značky spojené s formátováním vzhledu
- XHTML 1.0 Transitional – povoluje atributy pro formátování textu a odkazů v elementu body a některé další atributy
- XHTML 1.0 Frameset – používá se při použití rámců pro rozdělení okna prohlížeče na dvě nebo více částí

Kaskádové styly CSS

Kaskádové styly, známé také pod zkratkou CSS (z anglického Cascading class Sheets) jsou moderním jazykem umožňujícím účinné formátování stránek psaných v jazycích HTML, XHTML či XML. Jazyk byl navržen standardizační organizací W3C.

Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak.

JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. Na rozdíl od serverových programovacích jazyků (například PHP) sloužících ke generování kódu samotné stránky, JavaScript běží na straně klienta, tedy v prohlížeči až po stažení do vašeho počítače. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

Kapitola popisuje použité implementační nástroje, seřazené dle architektury informačních systémů. Pro jednotlivé vrstvy architektury je použit jiný implementační jazyk, nástroj. Datová vrstva naší systému, tedy vlastní databáze je implementována pomocí databázového serveru MySQL. Na aplikační vrstvě, pro přístup k databázi, a pro dynamičnost stránek je použita technologie PHP. Na klientské vrstvě, pro komunikace mezi uživatelem a systémem jsou použity technologie jako XHTML, s kaskádovými styly CSS a JavaScriptem. (zdroj: literatura [5], [14], [15])

4 Analýza

V kapitole popíšu analýzu navrhovaného systému, pod kterou shrnuji požadavky na systém, prohlášení o cílech a modely navrhovaného systému.

Analýzu systému jsem začal prostudováním různých již existujících internetových obchodů, na různých tématech. Těmi jsem se inspiroval, a získal jsem informace jak vlastně má vypadat internetový obchod, jaké požadavky musí plnit. Hlavním cílem bylo se seznámit se strukturou internetových obchodů a jejich funkcemi. Získal jsem dobré informace, které jsem mohl užitečně použít i v mém systému, ale byly i záporné příklady, které sloužili jako poučení, abych tak různé věci nedělal.

4.1 Prohlášení o cílech uživatele

Po konzultace s uživatelem, budoucím majitelem systému vznikly tyto popisy představy o tom co by měl systém dělat:

„Hlavním úkolem systému pro prodej stavebních materiálů je nabídnout na prodej zboží přes internet. Bylo by dobré kdy by měl přehledné a jednoduché uživatelské rozhraní, aby se i nezkušení uživatelé vyznali v nakupování. Popřípadě i pomocník. Potřeba úplného seznamu nabídnutého zboží, dle všech kategorií. Možnost přihlášení uživatelům, a tak získat nějaké, předem zadané výhody pro určené uživatele. Aby přihlášení uživatelé mohli prohlížet stav vyřizování svých objednávek. Obsah košíku mohou podat na objednání. Zaregistrovaní uživatelé by měli mít možnost přístupu ke všem ostatním službám, ke kterým mají přístup i nezaregistrovaní uživatelé, a které systém nabízí.

Systém by měl nabízet zvýraznění, nebo i oddělení akcí do jiné sekce. Na toto zboží, s výhodnějšími cenami opoutat pozornost nakupujícího. Dále při průběhu nakupování zobrazit stav košíku, a aktuální cenovou hodnotu nákupu. Kdyby se zákazník rozhodl jinak, vyprázdnit celý obsah košíku, nebo jednotlivé položky. Po odeslání objednávky oznámení přijetí objednávek prostřednictvím emailu. Emailem může být zákazník obeznámen i např. při registraci, nebo při jiných zvláštních záležitostech. Nezanedbatelné je poslání emailu v situaci, kdy uživatel zapomene své heslo. V tomto případě systém vygeneruje nové heslo, a pošle ho zákazníkovi na zadanou emailovou adresu. Při požádání o zadání emailové adresy, systém ověřuje, jestli opravdu existuje zákazník v našem systému kterému odpovídá zadaná emailová adresa.

Bylo by dobré zvýšit návštěvnost systému přidáním příspěvků k jednotlivým druhům zboží. Přidat příspěvek by měl mít možnost zaregistrovaný zákazník. To by zvýšilo návštěvnost, zákazníci by mohli diskutovat o svých dobrých popřípadě zlých zkušenostech. Z tohoto důvodu by systém měl mít jeden typ uživatelů, kteří by sloužili jako správci diskuze.

Dalším typem uživatelů, kteří se vyskytují v systému by byl typ správce. Jemu by systém umožnil úplně jiné možnosti, myslíme zde, že jemu by neumožnil např. nakupování. Správce musí mít uživatelské rozhraní systému, který umožňuje vyřizování objednávek, nastavení stavu vyřizování objednávek. Musí mít přehled o svých zaregistrovaných zákaznících. Může nastavovat nebo i změnit údaje jednotlivých uživatelů a slevy při nakupování. Dle potřeby tyto uživatele i smazat. Musí mít možnost editovat jednotlivé zboží. Editovat všechny údaje o existujících typech zboží, což zahrnuje i informace o cenách. Má možnost přidávat nové zboží, nebo smazat již existující. Správce taky nastavuje a edituje akce.“

Tedy shrnuté obecné prohlášení o cílech:

- základní funkce elektronického obchodu
- registrace uživatelů
- autentizace uživatelů
- úplný seznam nabídnutého sortimentu
- nakupování do košíku
- objednání obsahu košíku
- průběžná kontrola stavu objednávek
- oznámení jednotlivých akcí prostřednictvím emailu
- diskuze o jednotlivých typech zboží, přidat připomínky
- obdržet zprávu o objednávkách pro správce
- přehled všech objednávek pro správce
- přehled svých objednávek, a jejich stav vyřizování pro zákazníky
- přehled o všech zákaznících pro správce, a editace jejich slev
- úprava zboží a akcí

4.2 Požadavky na internetový obchod

Internetový obchod by měl nabídnout dostatečný výběr zboží, který je dobře přehledný, aby uživatel snadno našel co hledá. Systém by taky měl mít jednoduché uživatelské rozhraní, přihlášení do, a odhlášení ze systému, při kterém by bylo umožněno snadné nakupování. Z hlediska správce je systém již trochu složitější. Správce má mít pod kontrolou mnoho věcí jako např. objednávky, zboží – úprava nabídnutého zboží apod.

4.2.1 Jednotlivé funkce systému

V této podkapitole jsem se snažil shrnout funkce systému, které bude systém umožňovat. Tyto funkce rozdělím dle aktérů, kteří mají přístup k systému, tedy neregistrovaný uživatel, registrovaný uživatel (zákazník) a správce systému.

Neregistrovaný uživatel:

Neregistrovaný uživatel může při návštěvě elektronického obchodu:

- Zobrazit zboží, hledat, vybírat ze zboží
- Vkládat zboží do košíku, ale nemůže to objednat
- Zobrazit akce
- Přečíst novinky
- Prohlédnout ceny
- Zaregistrovat se

Registrovaný uživatel:

Registrovaný uživatel může při návštěvě elektronického obchodu:

- Přihlásit se do, a odhlásit ze systému
- Všechno co je povoleno i neregistrovaným zákazníkům
- Poslat k objednávce obsah košíku – i neregistrovaný zákazník mohl přidat zboží do košíku, registrovaný to může i objednat
- Prohlédnout své objednávky, a stav jejich vyřízení – může prohlédnout své starší objednávky, které jsou označeny stavem vyřízení
- Psát do diskuze – psát názory, a připomínky o jednotlivých druzích zboží, nebo jen obecně
- Projevit zájem o nějaké zboží – zákazník může projevit zájem o zboží které potřebuje dané chvíli, aby to správce pokusil zabezpečit

Správce systému:

Správce systému má největší práva, má pod kontrolou hlavní části systému:

- Spravuje a vyřizuje objednávky – postupně se objevují správci nové objednávky, a ty má za úkol vyřídit, a označit stavem vyřízení
- Filtrace objednávek – hledání mezi objednávkami
- Správa zboží, což zahrnuje přidání nových kategorií, do kategorií nových zboží, smazat neaktuální zboží
- Správa uživatelů – přidání nových zákazníků, popřípadě smazání zákazníků
- Správa novinek a akcí
- Správa diskuze

- Pokusí se zabezpečit požadované zboží

V systému se vyskytuje ještě jeden typ uživatelů, tento je jen zvláštním typem registrovaných uživatelů, kteří jsou pověřeni správou diskuze, mohou pravidlům neodpovídající připomínky smazat, nebo např. vytvořit nové kategorie.

4.3 Modely navrhovaného systému

Při návrhu systému je třeba použít několik diagramů, dle kterých bude probíhat implementace systému. Pomocí těchto diagramů (které můžeme nazvat modelem systému) určujeme požadavky na systém, jakou roli mají plnit, jaké tabulky má mít databáze. Tyto modely jsou velice důležité, a téměř nezbytné při návrhu a následné implementaci systému.

Nejčastěji se pro modelování informačních systémů používá jazyk UML. Já jsem taky při návrhu použil jazyk UML. Pomocí tohoto jazyku byli vytvořeny diagramy USE CASE a ER diagram, pomocí kterých byl systém implementován. Seznámení s tímto technologií jsem uváděl v kapitole 3, implementační nástroje, kde čtenář najde stručný úvod do jazyku UML.

V kapitole jsem rozepsal návrh systému pomocí diagramů USE CASE a ER.

4.3.1 Use Case diagram systému

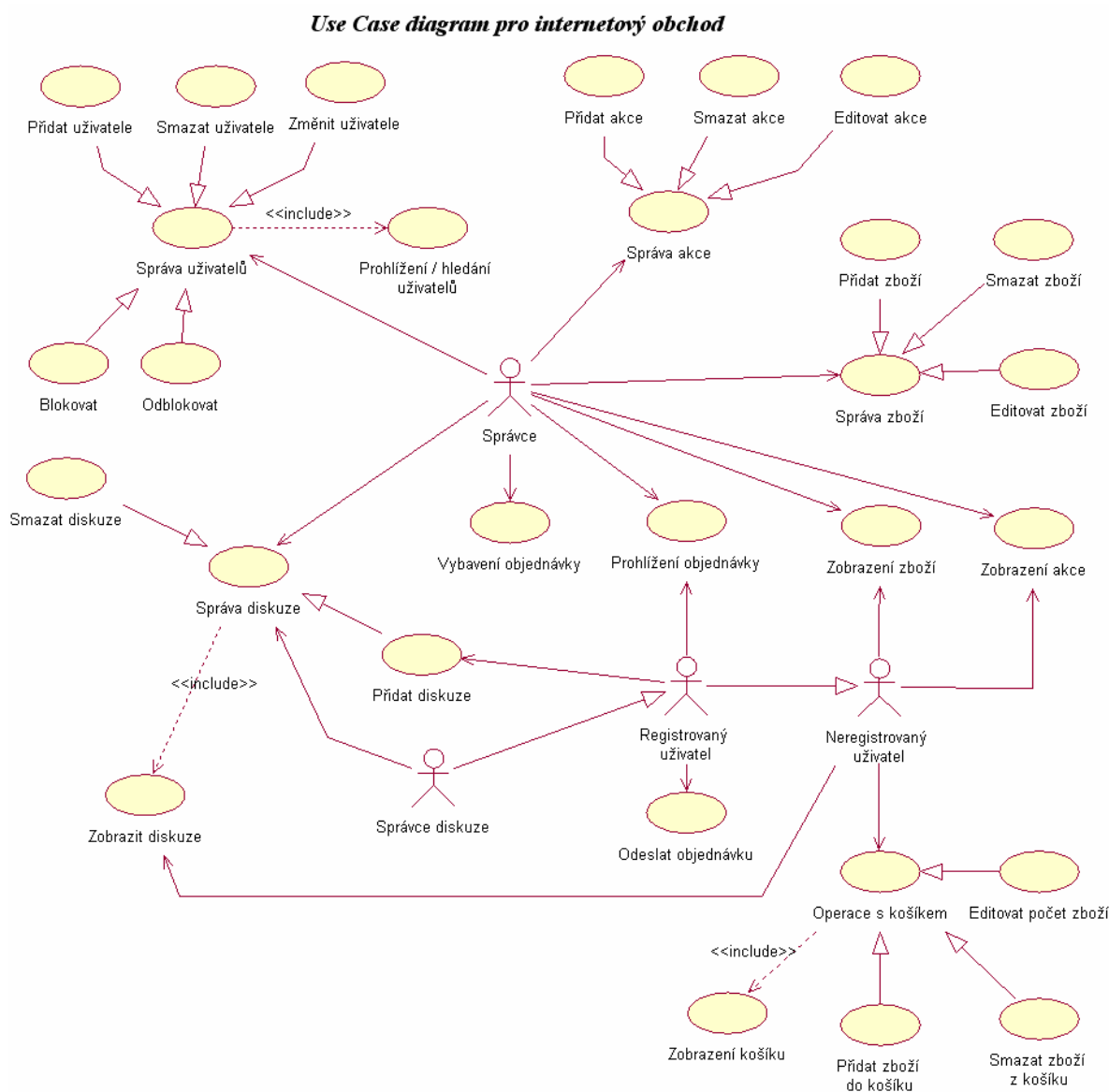
Také diagram případu užití patří mezi modely popisující funkcionalitu systému, vlastně jak bude systém fungovat. Právě proto je tento typ diagramu vhodný pro komunikace s uživatelem. Slouží pro základní vymezení hranice mezi systémem a jeho okolím. Komponentami tohoto model jsou:

- aktér (actor) – uživatelská role nebo spolupracující systém
- hranice systému (system boundary) – vymezení hranice systému
- případ použití (use case) – dokumentace události, na kterou musí systém reagovat
- komunikace – vazba mezi aktérem a případem použití (aktér komunikuje se systémem na daném případě)

Při vytváření diagramu případu užití je třeba brát v úvahu, že existují tzv. sekundární aktéři, tj. uživatelské role nebo spolupracující systémy, pro něž není systém přímo určen, ale které jsou pro jeho činnost nutné. Případy, kdy chceme vyznačit směr komunikace, značíme orientovanými šipkami. Mezi případy použití mohou existovat vztahy. Pokud chceme explicitně vyjádřit akt, že takový vztah existuje, používáme stereotypy, standardně <<include>> - pokud jeden případ zahrnuje případ jiný (např. autentizaci), nebo <<extend>> - pokud nějaký případ rozšiřuje chování jiného (je zde možnost volby)

Na obrázku 3.1 vidíme diagram případu užití našeho systému. V našem diagramu se vyskytují 4 aktéři. První, pokud to seřadíme dle pravomocí vzestupně, je neregistrovaný uživatel. To je návštěvník internetového obchodu, který se nezaregistruje, nebo se jenom nepřihlásí do systému.

Takoví uživatelé mohou využívat operace s košíkem, ale nemohou obsah košíku odeslat k objednání. Mohou jenom prohlížet zboží, popřípadě akce, novinky a diskuze, také mohou vkládat zboží do košíku, ale pouze do něj vkládat, nakupovat ne. Registrovaný uživatel může dělat všechno co může nezaregistrovaný – nepřihlášený uživatel, a navíc může obsah košíku i odeslat k objednání. Tyto objednávky lze později prohlížet, kontrolovat stav jejich vyřizování. Přihlášení taky mají možnost psát diskuze. Jeden jednoduchý registrovaný uživatel je správce diskuze, který jenom spravuje



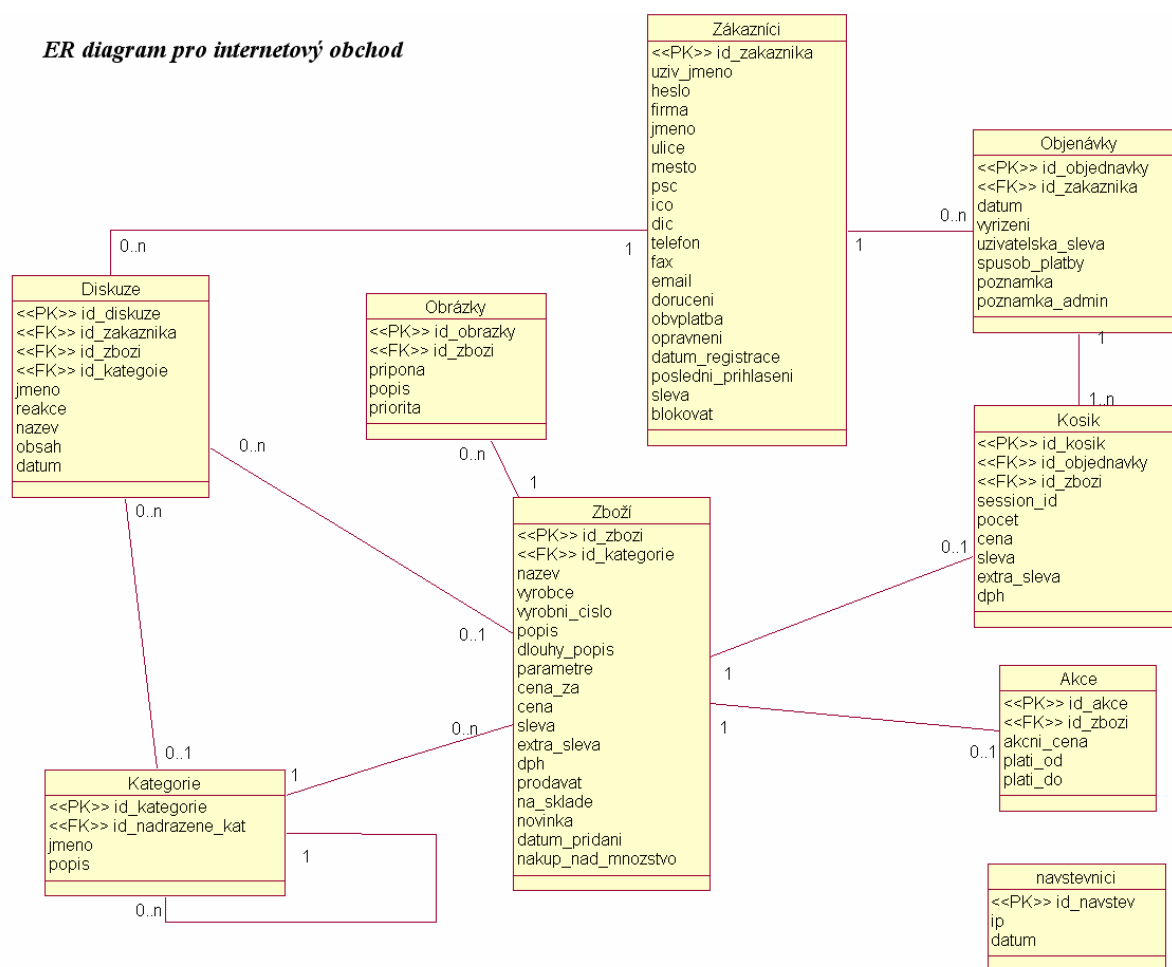
Obrázek 3.1: Use Case diagram pro internetový obchod

diskusní skupiny, popřípadě maže připomínky neodpovídající pravidlům. Posledním přístupujícím aktérem systému je správce systému, který má nejvyšší práva, hlavně vyřizuje objednávky. K němu se dostanou všechny objednávky zaregistrovaných zákazníků. Druhým nejdůležitějším úkolem správce je správa zboží. Upravuje, vkládá popř. maže zboží či skupiny z databáze. Má dohled nad všemi zaregistrovanými zákazníky, může je blokovat, přidat nové, popřípadě některé smazat.

4.3.2 ER diagram systému

Klasickým způsobem prezentace modelovaného světa je E-R (entitně – relační) model, pracující s pojmy entita (objekt), vztah (relationship), atribut (vlastnost). Jeho základem je převedení komplexních struktur modelované skutečnosti do dvourozměrných tabulek a nalezení vztahů mezi nimi. Právě snadná transformovatelnost do tabulek ho činí vhodným pro relační databáze. Základní komponentou modelu je E-R diagram, který se používá pro vizuální reprezentaci dat. ER diagram reprezentuje logické vztahy mezi entitami nebo objekty. Má nezastupitelnou úlohu nejen při návrhu databázových aplikací, ale i při jejich optimalizaci a odstraňování chyb. Efektivní datový model přesně a úplně popisuje a vyhovuje nárokům zadání a je použitelný pro tvůrce databáze, eliminuje redundanci dat a je nezávislý na hardwaru a softwaru.

ER digramy jsou také nazvané diagramem tříd. Vztahy mezi třídami (asociace) vyznačují možné vazby mezi objekty. konce vztahů mohou být ohodnoceny rolí – role označují jakou roli objekt ve vztahu hraje. Pro zachycení kardinality a volitelnosti vztahů se používá notace N..M, kde N a M může být číslo nebo *, samotná * znamená totéž jako 0..*. Pokud je zapotřebí si o vztahu něco pamatovat, používá se tzv. přidružené třídy (atributy vztahů).



Obrázek 3.2: ER diagram pro internetový obchod

Speciální druhy vztahů představují agregace, kompozice a generalizace:

agregace (aggregation) – je druh vztahu, kdy jedna třída je součástí jiné třídy – vztah typu celek/část,

kompozice (composition) – je silnější druh agregace – u kompozice je část přímo závislá na svém celku, zaniká se smazáním celku a nemůže být součástí více než jednoho celku,

generalizace (generalization) – druh vztahu, kdy jedna třída je zobecněním vlastností jiné třídy (jiných tříd) – vztah typu nadtyp/podtyp, generalizace/specializace.

Na obrázku 3.2 vidíme ER diagram našeho systému, který obsahuje 9 entit, které jsou následující:

Primárním klíčem entity uživatel je `id_uzivatele`, který jednoznačně určuje daného uživatele unikátního v systému. Dále obsahuje atributy související s daným zákazníkem, který je v systému zaregistrovaný, jako např. jeho adresa, email, telefon atd. Entita uživatel je ve vztahu s objednávkou s kardinalitou 1 : 0..n, což znamená, že jeden uživatel může mít i několik objednávek, ale však nemusí mít ani jednu. S entitou diskuze je entita uživatel také ve vztahu s kardinalitou 1 : 0..n. Tedy jeden uživatel může mít připsán až několik připomínek do diskuzí, ale nemusí mít ani jednu.

Entita zboží reprezentuje tabulku v relační databázi ve které bude umístěno všechno nabídnuté zboží elektronického obchodu. Zboží je také potřeba unikátně identifikovat, na to slouží primární klíč `id_zbozi`. Obsahuje i cizí klíč, pomocí kterého je zařazen do kategorie. Tedy je ve vztahu s tabulkou kategorie, která určuje jednotlivé kategorie zboží. Kardinalita vztahu je 1 : 0..n, protože zboží musí patřit právě do jedné kategorie, ale do jedné kategorie nemusí být zařazen ani jeden druh zboží, avšak druhů zboží může být několik.

Entita diskuze reprezentuje tabulku, v které budou uloženy jednotlivé diskusní připomínky. Kardinalita vztahu se zbožím je 0..n : 0..1, protože se to může týkat jednoho druhu zboží avšak nemusí. Taky ta připomínka se může týkat jedné skupiny zboží, avšak nemusí, a proto je kardinalita vztahu i s entitou kategorie 0..n : 0..1.

Atribut `id_kosik` entity košík jednoznačně určuje jedno objednané zboží z dané objednávky. Proto je ve vztahu se zbožím s kardinalitou 1 : 1..0, a s objednávkou 1 : 1..n.

Entita obrázky obsahuje informace o obrázcích zboží, entita akce je vlastně výtažek ze zboží, které je právě v akci. Entita návštěvníci slouží pro vedení statistiky o návštěvnosti.

Táto kapitola se týká analýzy a návrhu informačního systému – elektronický obchod. Jsou zde shrnuty požadavky kladené na elektronický obchod, které by informační systém měl obsahovat, a funkce které musí provádět. Kapitola obsahuje i prohlášení o cílech, stručný a obecný popis představy o tom, co by měl systém dělat, a jak by měl fungovat. Z obchodního hlediska je prohlášení o cílech základním materiálem vznikajícím u uživatele, kde zcela neformálně popíše, co hodlá automatizované pomocí informačního systému zpracovávat. Poslední část kapitoly se týká čistého návrhu a jejich modelů. Hlavně se to týká diagramu ER a use case, které jsou podrobně diskutovány i na daný navrhovaný systém.

5 Implementace

Následující kapitola zabývá detailním popisem implementace jednotlivých částí systému. Snažil jsem se zde popsat, jak jsem řešil funkční části systému, jaké techniky a algoritmy jsem přitom používal. Kapitola začíná popisem implementaci databázové struktury systému na úrovni databázové vrstvy. Pak pokračuje popisem řešení aplikační a klientské vrstvy, kde popisují operace s daty uloženými v databázi, implementace uživatelského rozhraní, spojení klientské a aplikační vrstvy, zabezpečení systému a databáze.

5.1 Tvorba databází a tabulek

Databázová struktura systému je implementována v databázovém systému MySQL. (Popis tohoto systému viz. kap. 3 – Implementační prostředky) Struktura databáze se skládá z osmi databázových tabulek, což odpovídá návrhu databáze v ER diagramu. Dále i jednotlivé vztahy, reference a atributy entit jsou implementovány dle tohoto diagramu. Jednotlivé entity v diagramu chápeme jako jednu samostatnou tabulku databázové struktury. Vztahy mezi tabulkami jsou řešeny pomocí cizích klíčů.

Indexy, klíče a integritní omezení

S návrhem a tvorbou databáze se úzce pojí dva další pojmy: indexy a klíče. Index je způsobem, jímž databázový systém žádáme, aby dohlížel na hodnoty ve specifickém sloupci nebo v kombinaci sloupců. Výsledkem takového požadavku je zvýšený výkon při načítání záznamů, přestože se může při přidávání nebo úpravě záznamů poněkud snížit.

Klíč je ve světě databází považován za integrální součást procesu normalizace a používá se při návrhu složitějších databází. Existují dva typy klíčů: primární a cizí. Každá tabulka v mé implementaci obsahuje jeden primární klíč a ten je obvykle spojen jako cizí klíč s jinou tabulkou. Primární klíč tabulky je umělým způsobem odkazování na záznam a musí se řídit třemi pravidly:

- Musí vždy obsahovat hodnotu.
- Jeho hodnota se nesmí nikdy změnit.
- Jeho hodnota musí být v celé tabulce jedinečná.

Při vytváření tabulek pomocí příkazu `CREATE TABLE`, definujeme které sloupce tabulce chceme určit jako index nebo klíč. To provádíme pomocí modifikátorů `INDEX`, `PRIMARY KEY` a `FOREIGN KEY` na konci deklarace tabulky.

Při vytváření tabulek ještě potřebujeme zabezpečit určité referenční integritní omezení. Při určení cizího klíče a reference na druhou tabulku, jsem používal klauzule `ON DELETE` který specifikuje co se má stát při `DELETE` nadřazeném záznamu – `CASCADE` smaže všechny podřízené záznamy. Tak například při smazání jednoho produktu z tabulky zboží, SŘBD zabezpečuje smazání

záznamu které jsou referenční na tento záznam, jako např. všechny obrázky produktu z tabulky obrázků, všechny příspěvky k produktu z tabulky diskuze, a také všechny akce příslušného produktu z tabulky akce. Tato klauzule (ON DELETE) funguje jenom u databázových systémů MySQL typu větší než 4.0. Z tohoto důvodu při testování na školním serveru jsem řešil toto integritní omezení manuálním testováním a vymazáním referenčních záznamů.

V implementaci jsem použil všude při primárním klíče modifikátor AUTO_INCREMENT, kterým žádáme systém aby si sám ve sloupci generoval unikátní číselné hodnoty.

CHAR versus VARCHAR

Oba datové typy slouží k ukládání řetězců a lze v nich nastavit maximální délku. Liší se tím, že vše, co je uloženo ve sloupci typu CHAR, je vždy uloženo jako řetězec s délkou sloupce (řetězec bude automaticky doplněn mezerami). Naproti tomu hodnoty ukládané do sloupce typu VARCHAR jsou vždy jen tak dlouhé, jako kdyby byly uloženy samostatně.

Z toho vyplývá, že:

- Sloupec datového typu VARCHAR zabírá obvykle méně místa na disku.
- Nepoužíváme-li tabulku typu InnoDB, jsou sloupce typu CHAR rychlejší než sloupce typu VARCHAR.

Rozdíly v obsazování místa na disku nebo v rychlosti jsou většinou prakticky nepostřehnutelné. Existuje však ještě třetí, i když menší rozdíl. Databázový systém MySQL osekává nadbytečné mezery z hodnot získaných ze sloupce typu CHAR, nebo vkládaných do sloupce VARCHAR.

V implementaci databázové struktury tohoto systému jsem používal pro řetězce všude VARCHAR, kromě uložených hesel, kde délka řetězce bude vždy stejná.

Dále jsem ještě používal datové typy INT – celé číslo, DECIMAL – číslo typu double uložené jako řetězec pro dosažení čísla s pevnou desetinnou čárkou, TEXT – sloupec s maximální délkou 65535 znaků a pro datum jsem používal datový typ DATETIME.

Hodnoty NULL a implicitní hodnoty

V databázích lze určit, zda může daný sloupec obsahovat prázdné hodnoty, nebo zda můžeme pro daný sloupec definovat implicitní hodnotu.

Hodnota NULL (prázdná hodnota) vyjadřuje ve světě databází, stejně jako ve světě programování, pouze chybějící nebo prázdná data. Chceme-li zajistit, aby měl záznam v daném sloupci v každém případě hodnotu, přidáme do popisu typu sloupce atribut NOT NULL.

Při tvorbě nové tabulky, můžeme specifikovat rovněž implicitní hodnotu s modifikátorem DEFAULT. Používal jsem to v případech, kdy většina záznamů obsahovala v dotyčném sloupci stejnou hodnotu, a tak nastavením implicitní hodnoty ušetřím nutnost vkládání stejné hodnoty do všech řádků.

5.2 Implementované funkce systému

V kapitole popíšu jednotlivé implementované funkce systému, na úrovni aplikační vrstvy – jak pracuje systém s datem v databázi, a také i na uživatelské vrstvě – jak je řešena komunikace s uživatelem systému. Aplikační skripty, které manipulují a zpracovávají data z databáze, jsou implementovány v jazyce PHP. (popis tohoto jazyka viz. kap. 3 – Implementační prostředky)

5.2.1 Registrace uživatele

Uživatelé mohou používat náš systém bez přihlášení, nebo po přihlášení. Bez přihlášení nemohou objednat žádné zboží, pouze mohou zboží vkládat do košíku, a manipulovat s košíkem. Po přihlášení, uživatelé získají výrazné vylepšení možností, a vůbec mohou i nakupovat, tedy objednat produkty z košíku, tj. navštívit pokladnu. Na základě toho, jim mohou být uděleny i uživatelské slevy, pokud tak rozhodne administrátor. K tomu aby se uživatel mohl přihlásit, musí se zaregistrovat v našem systému. Registraci uživatel najde po navštívení systému v přihlašovacím bloku, pod přihlašovací kolonkou. Po kliknutí na odkaz „Vytvořit účet“, se před uživatelem objeví formulář, který se člení na tři části:

- Údaje o zákazníkovi (obrázek 5.1 a)
- Informace o platiteli (obrázek 5.1 b)
- Odeslání registrace (obrázek 5.1 c)

V části údaje o zákazníkovi jsou všechny údaje povinné, a v části informace o platiteli jsou i nepovinné údaje. K úspěšné registraci je třeba souhlasit i s obchodními podmínkami na konci stránky v části odeslat registraci. Povinné údaje jsou označeny hvězdičkou.

The screenshot shows a form titled "Informácia o platiteľovi". It contains several input fields: "Firma", "Krstné meno a priezvisko:*" (with an asterisk), "Adresa:*" (with an asterisk), "Mesto:*" (with an asterisk), "PSČ:*" (with an asterisk), "Telefón:", "Fax:", "IČO:", "DIČ:", and "Obvyklá platba:" with a dropdown menu showing "do predu". There is also a text area for "Poznámka o doručení:". At the bottom, there is a note: "Poznámka: Do kolonky 'Poznámka o doručení' môžete zapisať napr. inú doručovaciu adresu."

Obrázek 5.1 b): Formulář pro registrace –
informace o platiteli

The screenshot shows a form titled "Údaje o zákazníkovi" with a note "(* - povinné)". It contains four input fields: "Užívateľské meno:*" (with an asterisk), "E-mail:*" (with an asterisk), "Heslo:*" (with an asterisk), and "Potvrďte heslo:*" (with an asterisk).

Obrázek 5.1 a): Formulář pro registrace –
údaje o zákazníkovi

The screenshot shows a form titled "Odoslať registráciu". It contains a checkbox with the text "Súhlasím s obchodnými podmienkami (Obchodné podmienky)*:" and a button labeled "Odoslať registráciu".

Obrázek 5.1 c): Formulář pro registrace –
odeslání registrace

Z pohledu aplikační části registrace, v okamžiku odeslání formuláře systém kontroluje vyplněné údaje. Pomocí regulárních výrazů se testují jednotlivá políčka formuláře. V první řadě testují či byly zadány povinné údaje, a testují jejich správnost, jestli obsahují jenom dovolené znaky, příslušné požadované údaje. Např. telefon obsahuje jenom čísla, uživatelské jméno jenom charaktery anglické abecedy bez mezery. Zvláštní případ tvoří emailová adresa, která má mít určitý formát, proto to uvádím jako příklad:

Regulární výraz: `"^[[:alnum:]][a-z0-9_.-]*@[a-z0-9.-]+\.[a-z]{2,4}$"`

určuje: řetězec – elektronická adresa musí začínat písmenem nebo číslicí a musí pokračovat určitou kombinací číslic, písmen, podtržitek, teček a pomlček. Elektronická adresa musí obsahovat symbol zavináče (@), za nímž bude následovat kombinace písmen, číslic, teček a pomlček. Nakonec bude uvedena tečka a dvou až čtyřmístný řetězec (com, edu, cz, info). Pokud se nějaký řetězec neshoduje s povinným předpisem, nebo povinný údaj nebyl vyplněn, uživatel je upozorněn na nesprávně vyplněný formulář.

Po úspěšné kontrole zadaných řetězců, a vyplnění povinných údajů, se dostáváme ke kontrole „významu“ údajů. To znamená, že testujeme či uživatelem zadané uživatelské jméno nebo email již neobsahuje náš systém. Uživatelské jméno musí být jedinečné, protože přihlášení do systému probíhá dle uživatelského jména a hesla. Email musí být jedinečný, protože kdyby uživatel zapomněl heslo, tak v části systému, kde je uživateli vygenerováno nové heslo, je po uživateli požadováno zadání emailové adresy, na kterou budou poslány nové registrační údaje.

Pokud všechny kontroly proběhly úspěšně, je uživatel se zadanými údaji přidán do databáze, a je obeznámen o úspěšné registraci prostřednictvím emailu, ve kterém jsou mu poslány i registrační údaje. Poté se uživatel může přihlásit do systému.

5.2.2 Zapomenuté heslo

V dnešní době kdy má uživatel možnost se zaregistrovat na mnohých webových serverech, aby mohl využívat různé výhody, poskytované různými servery, často se stává, že zapomene každodenně nepoužívané přihlašovací údaje. Právě proto náš systém umožňuje poslat uživateli své přihlašovací údaje, tj. vygenerovat nové heslo, a tak umožnit přihlášení do systému.

Uživatel navštíví náš server a chce se přihlásit, ale právě tehdy na to přijde, že zapomněl např. heslo, ještě na místě v přihlašovacím bloku (který se nachází v horním levém rohu uživatelského rozhraní) se nachází odkaz na část systému, který umožní vygenerování nového hesla.

Po kliknutí na odkaz, se uživateli objeví formulář, kde je požádán o zadání emailové adresy, kterou zadal při registraci. Aby zapomenuté přihlašovací údaje zákazníka mohly být obnoveny, musí si vlastně pamatovat alespoň zadanou emailovou adresu. To je potřebné proto, aby ji systém mohl nějak ztotožnit, resp. vyhledat jeho údaje z databáze.

Po zadání emailové adresy a odeslání formuláře, systém opět kontroluje správnost zadané emailové adresy, pomocí regulárních výrazů, jako to bylo při registraci. V případě nesprávnosti, je uživatel upozorněn, a žádán o opakované zadání údajů. Po úspěšné syntaktické kontrole údajů, systém vyhledá zákazníka v databázi, dle zadané emailové adresy. Pokud se to nepovede, tak zadaná emailová adresa není zaregistrovaná v naší databázi, a to je oznámeno i uživateli. Po úspěšném vykonání vyhledávacího dotazu, systém vygeneruje nové, náhodné heslo, tím přepíše existující heslo v databázi, a poté pošle přihlašovací údaje (včetně nového hesla) zákazníkovi emailem. Vygenerovat nové heslo je potřebné, protože systém k zabezpečení hesel používá vysoce bezpečnostně rozšířenou hašovací funkci MD5, které vytvoří kontrolní součet o velikosti 128 bitů, z libovolné délky. Z tohoto kontrolního součtu – hašového kódu, však zpětně dekodovat není možné. Tedy není možné žádným způsobem získat zpátky uživatelem dříve zadané heslo.

Vygenerování nového hesla

Vygenerování jsem řešil následujícím způsobem: Vytvořím jednoduché pole, do kterého vkládám všechny znaky, které můžu objevit v heslech, tj. malé znaky anglické abecedy, a všechna čísla. Poté vygeneruji náhodné číslo, a tím indexuji předem vytvořené pole. Tak získám jeden náhodný znak, a z osmi náhodných znaků poskládám nové, náhodné heslo. Na vygenerování náhodného čísla používám funkci `mt_rand`, což slouží na generování lepšího náhodného čísla. Řada generátorů náhodných čísel ze starších knihoven `libc` má podezřelé nebo neznámé charakteristiky a je pomalá. (`mt_rand` je průměrně čtyřikrát rychlejší) PHP standardně používá ve funkci `rand` generátor náhodných čísel knihovny `libc`. Funkce `mt_rand` tuto funkci plně nahrazuje. Používá generátor náhodných čísel známých charakteristik Mersenne Twister, který generuje náhodná čísla použitelná pro různé obory kryptografie. Tento generátor má periodu $2^{19937} - 1$ a má rovnoměrné rozložení až do 623 rozměrů. Navíc je rychlejší než většina ostatních generátorů. Jeho nevýhodou je větší složitost kódu než u jednoduchých kongruentních generátorů.

5.2.3 Přihlášení do systému a odhlášení ze systému

Návštěvníci systému mohou rozhodovat jestli se přihlásit do systému nebo používat systém bez přihlášení. Bez přihlášení uživatelé mají omezené možnosti, a nemohou využívat řadu užitečných funkcí. Tito uživatelé mohou využívat funkce košíku, ale objednat nemůžou nic. Uživatelé kteří se zaregistrují a používají náš systém po přihlášení, stávají se zákazníkem našeho internetového obchodu. Mají možnost objednat obsah košíku, kontrolovat stav svých dřívějších objednávek. Pokud administrátor systému, tedy vlastní prodejce rozhodně tak, můžou zaregistrovaní zákazníci využívat i uživatelské slevy, udělené přímo jen konkrétním zákazníkům. Implementace uživatelských slev jsou popsány níže.

Uživatelé kteří se rozhodnou pro přihlášení, přihlašovací blok najdou v levém horním rohu uživatelského rozhraní, a to vždy, bez ohledu na to kde nacházejí v systému. (obrázek 5.2) Přihlášení

přes tento přihlašovací blok, probíhá bez toho, aby se změnila lokalizace nacházení uživatele v systému. Po zadání přihlašovacích údajů a odeslání tohoto formuláře je volána funkce „přihlásit“. V této funkci je na začátku implementována kontrola zadaných údajů. Kontrolujeme či vůbec byly zadané nějaké údaje, či pole formuláře náhodou nejsou prázdné. Pokud bylo zadáno nějaké jméno a heslo, tak systém se pomocí vyhledávacího dotazu SELECT pokusí najít příslušného zákazníka z databáze. Při neúspěchu funkce vrátí chybový kód, a dle těchto chybových kódů skript login.php vypíše chybovou správu, ve které je uživatel informován, že jméno a heslo nesouhlasí s žádným uloženým v databázi. Když se uživatele podařilo najít, tak se nastaví uživatelské relace.



Obrázek 5.2: Přihlášení do systému

Uživatelské relace

Uživatelská relace je vlastně způsob, který umožní aby si „server pamatoval na uživatele“, vlastně pamatoval co uživatel již navštívil během jedné relace. Při uživatelských relacích jsou data uložena na serveru, nikoli ve webovém prohlížeči (jak to je v případě cookie), a identifikátor této relace používá k vyhledání záznamů o konkrétním uživateli (data relace).

Při práci s uživatelskými relacemi je nejdůležitější skutečnost, že každá stránka, která uživatelskou relaci používá, musí začínat voláním funkce `session_start()`. Tato funkce sdělí skriptovacímu stroji PHP, aby zahájil novou uživatelskou relaci nebo připojil stránku k existující relaci. Při prvním užití funkce `session_start()` se funkce pokusí odeslat soubor cookie s názvem `PHPSESSID` (název relace) a hodnotu, což je 32 písmenný, hexadecimálně vyjádřený, identifikátor relace. Protože je to pokus o odeslání souboru cookie, musí být funkce `session_start()` volána ještě před odesláním jakéhokoli obsahu do webového prohlížeče. Po spuštění relace lze hodnoty registrovat v superglobálním poli `$_SESSION`. Do tohoto pole jsem uložil hodnoty: identifikační číslo zákazníka, firma, jméno, uživatelské jméno, sleva a pokud je nastaveno i oprávnění, tedy nejdůležitější údaje o přihlášeném zákazníkovi.

Identifikace správce, a správce diskuze

Právě v této chvíli je systémem identifikováno, či se přihlásí jednoduchý zákazník, zákazník s oprávněním pro správu diskuze, nebo správce systému. Tato identifikace probíhá dle speciálního sloupce – opraveni, tabulky zákazníci v databáze. Tento sloupec má implicitně hodnotu `NULL`, ale pokud to tak není, a hodnota tohoto sloupce se rovná označením klíčového slova `spravce`, nebo `diskuze` tak je v předem zmíněném poli nastaven i hodnota `spravce` nebo `diskuze`. Pak se všude,

v celém systému kde zobrazení něčeho závisí na tom, kdo je přihlášen, kontrolují či jsou nastaveny tyto hodnoty superglobálního pole `$_SESSION`.

Uživatelské slevy

V úvodu této kapitoly, jsem se zmínil pár slovy o uživatelských slevách. Tyto uživatelské slevy se týkají konkrétního nebo konkrétních zákazníků s různými hodnotami slev. Hodnoty slev u jednotlivých zákazníků jsou uloženy v databázi. Nastavuje to správce, dle svého rozhodnutí, ve správcovské části, pod hlavním menu zákazníci, po kliknutí na daného zákazníka. Zde se může prohlédnout i ostatní údaje zákazníků, popřípadě nebo při potřebě to i změnit.

Ve chvíli kdy se zákazník přihlásí je kromě ostatních, nastavována i tato hodnota z databáze u daného zákazníka do superglobálního pole `$_SESSION`. Při výpisu ceny u produktů, pokud je přihlášen nějaký zákazník, je jednoduše z původně vypočtené ceny produktu odečtená sleva nastavená v poli `$_SESSION`, tedy sleva daného přihlášeného zákazníka. A právě to v důsledku způsobí, že po přihlášení určitého zákazníka, se produkty objevují levnější, než u nepřihlášených zákazníků. Ale tuto slevu nestačí odečítat jenom při zobrazení ceny produktu, třeba ji také odečítat i při výpisu košíku, v pokladně u výpisu objednávky, a u dřívějších objednávek zákazníka.

V těchto zmiňovaných případech by nebyl problém odečítat pomocí `$_SESSION` pole, jak to bylo v předchozích případech. Řešitelné by to bylo i u výpisu objednávek u správce systému, protože víme o kterého zákazníka se jedná u jednotlivých objednávek, ale problém by nastával, kdyby při tom správce změnil uživatelské slevy u jednotlivých zákazníků. Jev, kdy by zákazník objednal něco s určitou slevou, poté mu tuto slevu správce odebral, a na konečné faktuře by se mu tam objevila větší cena, by nebyl příjemný. Proto nemůžeme vždy počítat s aktuální uživatelskou slevou, musíme někdy uložit uživatelskou slevu aktuální při objednání zboží. A právě proto je potřeba do databázové tabulky objednávky uložit i uživatelskou slevu, a při výpisu objednávky odečítat tuto slevu.

Odhlášení ze systému řeší skript `logout.php`. Při odhlášení je potřeba vymazat celé uživatelské relace na serveru, nestačí vymazat jenom cookie, na straně klienta. Uživatelské relace totiž mají na straně klienta uloženy soubory cookie, ale data jsou na serveru. Pro odstranění jedné proměnné relace můžeme použít funkci `unset()`. Ale my chceme vymazat celé pole, všechny proměnné relace, a tak provádíme příkaz `$_SESSION = array();` což nám vynuluje celé pole. Poté je ještě třeba vymazat všechna data relace ze serveru, to provádíme příkazem `session_destroy()`. A na konec můžeme vymazat i cookie na straně klienta.

5.2.4 Funkce a rozdělení uživatelského rozhraní

Jednou z nejdůležitějších částí v informačních systémech je rozhraní umožňující komunikaci s uživatelem. Uživatelské rozhraní vlastně řeší komunikace mezi uživatelem a systémem. Přes toto jsou uživatelé informováni o stavu systému, v našem případě výpisu různých druhů zboží, kategorií, stavu košíku, odeslaných objednávkách atd. Správce také přes nějaké uživatelské rozhraní (což vypadá

trochu jinak než zákaznické) edituje a přidává zboží, vyřizuje objednávky, a nastavuje různé části internetového obchodu. Tyto funkční prvky je třeba nějak zorganizovat, umístit, tedy je třeba nějak rozdělit uživatelská rozhraní.

V našem systému se uživatelské rozhraní pro zákazníky skládá ze záhlaví, levé části, pravé části, zápatí a střední „obsahové“ části. V záhlaví je umístěno horní menu, obsahující stav košíku, a odkaz na pokladnu. V případě že košík obsahuje nějaké zboží, je aktivován jako odkaz, a po kliknutí na něj se před uživatelem objeví obsah košíku. (viz. obrázek 5.3) Pokladna je vlastně místo, kde přihlášení zákazníci mohou objednat obsah košíku. Pokladna je přístupná jenom přihlášeným zákazníkům. Na přelomu záhlaví a levé části najde uživatel přihlašovací blok, který mu umožní přihlásit, nebo po kliknutí na zde umístěný odkaz zaregistrovat v registrační části systému.

V levé části, pod přihlašovacím blokem je umístěna nabídka internetového obchodu, e-shop menu. Zde může uživatel dle jednotlivých kategorií vybírat a dívat se na produkty. Při kliknutí na některé kategorie v e-shop menu se uživateli objeví podkategorie, pokud jsme vybrali hlavní kategorie. V obsahové části se objeví seznam produktů, s obrázky, patřícími do vybrané kategorie.



Obrázek 5.3: Vzhľad uživatelského rozhraní systému

Dále ještě je na levé straně umístěna akční nabídka. Pravá strana je dynamická v závislosti na přihlášení uživatele. Jsou zde odkazy na obecné informace, na obchodní podmínky, na nápovědu. Také je zde umístěna vyhledávací kolonka, která umožní uživateli hledat produkty dle různých klíčových slov. Pokud je přihlášen zákazník, objevuje se zde i zákaznický profil. V zákaznickém profilu jsou umístěny odkazy na seznam všech objednávek zákazníka, kde může kontrolovat jejich stav. V zákaznickém profilu může také uživatel změnit své údaje, zadané při registraci. Ještě třeba

zmínit umístění novinek, které jsou také v pravé části uživatelského rozhraní. V administrátorském rozhraní se vůbec neobjevuje pravá část. S administrátorským uživatelským rozhraním jsem se zabýval ve zvláštní kapitole „administrátorské rozhraní“. Ve spodní části stránky, tedy v patě jsou odkazy na validatory.

Z ohledu implementace aplikační vrstvy je systém souborově rozdělen na 3 hlavní části. Na stránce, tedy v uživatelském rozhraní jsou části, které u různých funkčních částí systému zůstávají stejné, tedy nejsou změněny stále. Také jsou v záhlaví, levá a pravá část, a zápětí systému. Tyto části uživatelského rozhraní můžeme implementovat v oddělených souborech, a jednoduše vestavět do hlavních skriptů systému. Tak jsem implementoval záhlaví.php a pata.php. V souboru záhlaví jsem implementoval u záhlaví i levou část uživatelského rozhraní. V patě jsem řešil pravou část, a zápětí systému.

Stav košíku

Při implementaci záhlaví je přihlašovací formulář jednoduše vepsán do přihlašovacího bloku. Podrobný popis přihlašování je popsán v kapitole 5.2.3 Přihlášení do systému, a odhlášení ze systému. Výpis stavu košíku začíná získáním session id – identifikátoru relace. Celé fungování a implementace relací je v internetovém obchodě jednou z klíčových funkcí systému, která je nepostradatelná. Je to vlastně způsob umožňující serveru pamatovat si na uživatele, jak je to podrobně popsáno v předem zmíněné kapitole 5.2.3. Zde opět potřebujeme informace o košíku, což je jednoduchá databázová tabulka navázaná na identifikátory relace. V tomto případě chceme získat informace o košíku v aktuální relaci, přesněji počet položek, a celkovou cenu. Vyhledávacím dotazem hledáme v tabulce košík položky odpovídající aktuální relaci, jejíž id jsme již získali. V případě, že jsme nenašli žádnou položku košíku na tuto relaci, vypíšeme stav prázdného košíku, v jiném případě spočítáme získané položky, vypočítáme celkovou cenu položek v košíku, a tyto hodnoty vypíšeme do záhlaví. (viz. obrázek 5.4)

A horizontal notification bar with a dark background and light text. It contains the text: "Váš košík obsahuje 2 položky v celkové ceně: 36,267.60 SKK | Pokladna | Ako nakupovať?".

Obrázek 5.4: Horní menu – záhlaví pro zákazníka

Menu nabídky produktů

V souboru záhlaví následuje výpis nabídka zboží. Zde vypíšeme hlavní kategorie, které získáme dotazem do databázové tabulky kategorie, ale s podmínkou že nemají nadřazené kategorie, což nám zabezpečí získání jenom hlavních kategorií. Tyto kategorie zobrazíme jako menu. Při výběru určité kategorie uživatelem, se zobrazí podmenu, které se sestává z podkategorií určité kategorie. Tento jev je implementován předáním identifikačního čísla pomocí GET parametrů, a při výpisu jednotlivých hlavních kategorií je kontrolováno či nebylo předáno právě jeho id. Pokud to bylo, tak opět sáhneme do databáze a získáme z tabulky kategorie název kategorií kterým nadřazenou kategorií

je právě id hlavní kategorie jejíž id bylo předáno. Pokud žádnou takovou tabulku nenajdeme, znamená to, že hlavní kategorie nemá žádnou podkategorii.

Pro akce je v databázové struktuře implementovaná samostatná tabulka, která obsahuje id zboží, které je právě v akci, cenu takového akčního zboží a platnost. Při výpisu akcí, pomocí vyhledávacího dotazu, spojíme tabulku akce s tabulkou zboží, ale jediné s podmínkou souladu s aktuálním datem a platností. Tím získáme potřebné informace o akčních produktech, které vypíšeme v akční nabídce.

O tom, či produkt je novinka, nás informuje jeden atribut v tabulce zboží. Při výpisu novinek, v souboru záhlaví, jenom hledáme produkty nesoucí tyto informace, a podobným způsobem jako při akcích je vypíšeme.

5.2.5 Nabídka zboží – výpis produktů

V informačních systémech, jejichž hlavním úkolem je nabízet různé druhy zboží, je velice důležitá implementace, řešení a umístění nabídky s produkty. Mělo by to být umístěné na lehce přístupném, a na první pohled viditelném místě uživatelského rozhraní, aby přicházející uživatelé, kteří mají hlavní cíl nakoupit nebo získat informace o produktech, nabídku co nejnázorněji našli. Právě proto jsem se rozhodl umístit ji na lehce viditelném místě, v horní části levé strany uživatelského rozhraní. Jak je to celé vlastně implementováno je popsáno v předchozí kapitole.

Při vybrání nějaké kategorie uživatelem, v hlavní obsahové části se objeví seznam produktů patřících do této kategorie, resp. podkategorie. Jednotlivé produkty jsou zobrazeny v tabulkové podobě s názvem, malým obrázkem (který je po nahrání administrátorem na server převzorkovaný – zmenšený), a krátkým popisem produktu. Pro usnadnění práce uživatele je možno vybrat počet zobrazených produktů na stránce. Tato volba je umístěná nahoře, nad seznamem.

Z ohledu aplikační implementace je na začátku sestrojen vyhledávací dotaz dle různých podmínek. Při vybrání kategorie a podkategorie, je informace o nich, tedy jejich identifikační číslo předáno pomocí GET proměnných. Právě tyto proměnné kat. a podkat. slouží jako podmínky, důsledku čeho získáme jenom produkty patřící do určitých kategorií. Podmínkami dotazu můžou být i hledané výrazy, které se uživatel snaží najít. Také je při sestavení dotazu kontrolováno, či nebyla předána informace o tom, že je třeba vypsát aktuální nabídku, nebo novinky. V případě, že to bylo požadováno, je dotaz modifikován, aby vyhledal aktuality nebo novinky.

Před vlastním výpisem je uskutečněn výpočet počtu stránek, dle vybraného limitu zobrazení. Výpočet je vlastně velice jednoduchý, jenom spočítáme kolik záznamů podařilo najít, a pokud je to větší než dovolený počet zobrazení, tak počet stránek bude počet záznamů děleno limitem zobrazení, po zaokrouhlení nahoru. GET parametr „s“, předané při přechodu mezi jednotlivými seznamy, určuje začátek výpisu, pokud to nebylo zadané, výpis začínáme od prvního záznamu. Po určení těchto parametrů připojíme k dotazu klauzule LIMIT. Po provádění dotazu je výpis odkazu na předchozí, na

další stránky, dle potřeby. Pomocí implementace cyklu while, v kterém každém cyklu z výsledku dotazu načteme do asociativního pole další řádek, vypíšeme jednotlivé nalezené produkty.

Uživatel může daný produkt vybrat kliknutím na název nebo na obrázek produktu. Tedy mu zobrazí detailní výpis produktu, který sestává ze třech částí:

- detailní informace o produktu (obrázek 5.5 a)
- nakupovací zóna (obrázek 5.5 b)
- diskusní zóna k produktu (obrázek 5.5 c)

novinka

Semmerock CARAT style / platňa brúsená, mramorovaná, 3,5 cm-s fázou / platňa (40/40/3,5 cm)

Výrobné číslo: carat000006

Výrobca: CARAT

Popis

Pôvab platní CARATstyle spočíva v detaile: jednotlivé platne vykazujú čo do farebnej intenzity a rozdielného rozmiestnenia farieb odtienky, ktoré zosilňujú vyslovene živý charakter tohoto novodobého platňového programu. V dvoch atraktívnych farebných tónoch predstavuje CARATstyle svoj pôsobivo variabilný vzhľad a podmaňuje opticky pôvabnými vzormi a príbehmi. Zušľachtené a trvanlivé povrchy v kombinácii s presvedčivým designom prepozičujú platniam CARATstyle ich elegantný zjav

Parametry:

Spotreba : 6,1 kg/m² * Hmotnosť : 13,5 kg/ks * Množstvo na palete : 120 ks = 19,7 m² = 1620 kg

Na sklade:

Na sklade je množstvo: **14554 ks**

Keď potrebujete objednať väčší množstvo, kontaktujte prosím nás [TU a napíšte email](#).

Cena:

bez DPH: **220.40 SKK/ ks** s DPH: **262.28 SKK/ ks**

Obrázek 5.5 a): Detailní výpis produktu

Kúpiť tovar

Výrobné číslo	Cena	Množstvo
carat000006	262.28 SKK/ks	<input type="text"/> ks

[do košíku](#)

Obrázek 5.5 b): Detailní výpis produktu – nakupovací zóna

Pridať názor môže len zaregistrovaný, prihlásení zákazník. Prihláste sa, alebo zaregistrujte!

Názory		
test	<input type="checkbox"/> nový	20.04.07 15:30
└ RE: test	<input type="checkbox"/> nový	20.04.07 15:30
└ RE: test	<input type="checkbox"/> nový	20.04.07 15:31
Ahoj!	<input type="checkbox"/> nový	20.04.07 15:31
└ RE: Ahoj!	<input type="checkbox"/> nový	20.04.07 15:31

[Zobrazit všetky](#) [Zobrazit označené](#) [Pridať nový](#)

Obrázek 5.5 c): Detailní výpis produktu – diskusní zóna

Detailní informace obsahují výrobce, základní a detailní popis, parametry produktu, informace o množství na skladě a cenu produktu. Nakupovací zóna, umožňuje uživateli vkládat určité množství produktů do košíku. V diskusní zóně mohou uživatelé napsat své názory na produkty, nebo reagovat na ostatní názory. Implementace nákupní zóny je rozepsaná v následující kapitole Funkce košík, a diskusní zóna je popsána ve zvláštní kapitole Diskuze k produktům.

5.2.6 Funkce košíku

V internetových obchodech je asi nejdůležitější a taktéž nepostradatelná funkce košíku. Košík v internetových obchodech vypadá trochu jinak, než obvyklý nákupní košík, který lidé téměř každý den používají. Proti tomu význam a funkci má stejnou, umožnit uživateli nakupovat, nějakým způsobem vybírat zboží které chce nakoupit, a nějak „odnést“ k „pokladně“. Košík je implementován na základě uživatelských relací, které umožňují serveru pamatovat na uživatele, jak jsem to podrobně rozebíral v kapitole 5.2.3 v sekci uživatelská relace. Má to klíčovou roli v implementaci košíku. Košík je vlastně jednoduchá databázová tabulka, navázána na identifikační číslo uživatelské relace, uložená jako jeden atribut této tabulky.

O stavu košíku má uživatel neustálé informace, které jsou vypsané v horním „stavovém“ řádku. Zde nachází počet položek, které obsahuje košík, a celková hodnota košíku. Do košíku se může uživatel podrobně podívat po kliknutí na odkaz košíku, také v tomtéž horním řádku, který je aktivní jenom když košík obsahuje nějakou položku. Do košíku se uživatel také dostane po vkládání zboží do košíku v nakupovací zóně, tedy po kliknutí na tlačítko „do košíku“. Implementoval jsem to tak proto, aby zákazník hned mohl kontrolovat svůj košík, v jakém stavu ho má a kolik toho do něj již vložil. Pak se může rozhodnout, či bude pokračovat v nákupu, nebo pokračovat cestou k „pokladně“.

Vkládání zboží do košíku

Nákupní zóna u produktů vypadá jednoduše, vidí zde zákazníka vlastně výrobní číslo a cenu produktu, který právě vkládá do košíku, a kolonku formuláře do kterého uvádí požadované množství. Po stisknutí tlačítka systém zpracovává požadavek, a po správnosti požadavku uživatele přesměruje do košíku.

Na aplikační úrovni systém zpracovává požadavek obslužením formuláře, ve kterém uživatel uvedl požadované množství. Správnost množství je regulárním výrazem kontrolována, popřípadě nesprávnosti je uživatel upozorněn na nesprávnost požadovaného množství. Poté, systém též kontroluje množství na skladě, či vůbec máme z daného produktu na skladě, a či požadované množství na skladě není náhodou větší než jaké dokážeme zaručit. V případě vyhození podmínek by jsme měli vložit produkt „do košíku“, do databázové tabulky, ale nevíme jakou cenu tam máme dát. Jestli je produkt v akci nebo jestli platí extra sleva. Extra sleva znamená, sleva při nakoupení nad určité množství, tzv. množstevní sleva. Systém umožňuje touto slevu nastavit pro správce, která může určit, že nad jaké množství platí, a jak velká daná sleva je. Právě v této chvíli je třeba tyto údaje zkontrolovat. Dotazujeme tabulku akce, jestli se v ní nenachází id tohoto produktu. Pokud ano, vkládáme do košíku akční cenu. Údaje o extra slevě máme zachovány v tabulce zboží, takže pomocí již získaných údajů kontrolujeme či platí tato sleva, pokud ano, nastavíme příslušnou velikost v košíku. Zůstává ještě otázka, jestli košík již náhodou neobsahuje tento produkt. V případě že to již obsahuje, jenom zvýšíme nakupované množství.

Samozřejmě všechny dosavadní manipulace s košíkem probíhaly jenom v rámci určité uživatelské relace, což nám určilo identifikační číslo relace. Toto identifikační číslo i teď ukládáme do košíku, spolu ostatními údaji, protože je to velice důležitý atribut, dle kterého bude i později identifikována položka košíku.

Zobrazení košíku

Košík se zákazníkovi objeví jako jedna tabulka obsahující potřebné údaje o vybraných produktech. Řádkově jsou odděleny názvem produktu, pod nímž jsou sloupcově vyjmenovány jednotlivé údaje o nich. V posledním sloupci je zabudována možnost vymazat položku z košíku, nebo vyprázdnit celý obsah košíku. (viz obrázek 5.6)

Při návštěvě košíku uživatelem, je prvním krokem získání identifikačního čísla uživatelské relace. Dle toho zkusíme najít položky košíku. V databázovém dotazu spojíme tabulky košík a zboží, dle určitých podmínek, mezitím i dle identifikačního čísla uživatelské relace. Pokud jsme našli v databázi nějaké položky košíku na určité relaci, tak po výpisu v hlavičce tabulky vypíšeme jednotlivé položky v cyklu while, v kterém načteme jeden řádek do asociativního pole z výsledku dotazu, a to vypíšeme. Pokud jsme nenašli žádné položky, vypíšeme stav prázdného košíku.

Množství produktu v košíku je zobrazené v jednom poli formuláře, aby uživatel mohl změnit množství ještě i v košíku. Při změně zde systém opět kontroluje, či objednané množství nepřekročí množství na skladě, nebo či množství odpovídá nebo neodpovídá extra slevě, podobně jako to bylo v nakupovací zóně v předchozích detailně popsáno a vysvětleno.

Váš nákupní košík obsahuje:

Jed.cena bez DPH	Extra sleva v % **	Zákl.sleva v %	Množstvo *	Cena bez DPH v SKK	DPH v %	Cena s DPH v SKK	Vymaž
Produkt: Semmelrock CARATexclusiv / platňa brúsená a pieskovaná, 3,5cm s fázou/ platňa(40/40/3,5 cm)							
243.00	0.00	5.00	<input type="text" value="23"/> ks	5,309.55	19.00	6,318.36	X
Produkt: Semmelrock CARATexclusiv/doplňkové prvky/dekoračný pás(40/6/3,5 cm)							
204.00	0.00	5.00	<input type="text" value="122"/> ks	23,643.60	19.00	28,135.88	X
Cena celkom:				28,953.15	19.0	34,454.25	X

◀ Späť nakupovať
▶ Pokladňa

Obrázek 5.6: Zobrazení obsahu košíku

Sleva při objednání nad určité množství produktu – extra sleva

Systém je implementován tak, že správci umožní při úpravě nebo při přidání nového produktu, určit množství, nad které bude platit určitá sleva. Například, pokud zákazník objedná z určité dlažby nad 500 ks, tak je mu cena celého nákupu snížena 10 %-ní slevou.

Při nákupu systém kontroluje, jestli objednané množství neodpovídá této slevě, jak to bylo rozepsáno v předchozích odstavcích. Sleva je uchovaná v databázové tabulce košík. Rozhodl jsem se tak, protože extra sleva musí platit na celé objednané množství, a ne jenom na cenu např. za ks. Takže to není možné odčítat hned při přidání produktu do košíku. Taky nemůže být připojena ani k objednávce, tedy do databázové tabulky objednávka, protože jsou pro různé produkty různé. Tak při výpisu obsahu košíku, nebo při výpisu objednávky je to odečtené po vynásobení cen počtem objednaného množství.

5.2.7 Funkce pokladna

Vlastní význam pokladny z klasických obchodů asi zná každý. Pokladna v našem internetovém obchodě má podobný význam, ale odlišné funkce. V klasické pokladně zákazník zaplatí vybrané

zboží, ale v našem obchodě toto místo vlastně zákazníkovi umožní objednat předem naplněný obsah košíku.

V pokladně systém sestaví tzv. předběžnou objednávku, který slouží pro zákazníka aby překontroloval údaje uvedené na objednávce. Jsou mu zde vypsány údaje o objednateli, způsob platby a předmět objednávky. V dolní části je zobrazen prostor, do kterého zákazník může psát poznámku k objednávce. Zákazník si zde ještě stále může rozmyslet objednávku a vrátit se do košíku nebo k nakupování. Změnit své údaje uvedené na předběžném objednávce může v zákaznickém profilu. Ale pokud uživatel stiskne tlačítko objednat, stává se objednávka platná a závazná, bude uložena do databáze, a čeká na vyřízení.

Na aplikační úrovni, zpracování údajů systémem je řešeno obsluhou formuláře, která je poslána ke zpracování ve chvíli stisknutí tlačítka objednat uživatelem. Obsluha formuláře sestojí dotaz na vkládání nové objednávky, před kterým získá potřebné údaje které budou vloženy do databázové tabulky objednávky. Po úspěšném vložení objednávky do databáze, ještě musíme upravit položky košíku na tuto relaci. To znamená, že nastavíme v těchto položkách košíku, že patří k nově vložené objednávce. Jako reference slouží identifikační číslo objednávky. Při přijetí objednávky je správci poslán email, čímž je obeznámen, že obchod dostal novou objednávku.

Úspěšné přijetí objednávky je oznámeno zákazníkovi, a objeví se mu tady možnost vytisknout objednávku. Tato možnost sestaví objednávku bez dekoračních a barevných prvků stránky, aby si ji uživatel mohl vytisknout.

Uživatelská sleva je také vložena do databázové tabulky objednávky ve chvíli odeslání objednávky. Je to z důvodu, že uživatelskou slevu zákazníka je třeba někde uchovat, která byla platná ve chvíli objednání, nemůžeme stále odkazovat na aktuální, nastavené v tabulce zákazníci. (Myslím tedy při výpise starších, nebo právě vyřizujících se objednávkách.) Protože správce klidně může předem udělené uživatelské slevy zrušit. Uživatelské slevy jsem podrobně rozepsal v kapitole 5.2.3 v sekci Uživatelské slevy.

5.2.8 Diskuze – názory k produktům

V informačních systémech fungujících jako internetový obchod, není nejdůležitější funkcí diskusní zóna. Ale uživatelé, používající náš systém, kteří tak dobře neznají produkty které chtějí nakoupit, nebo které náš obchod nabízí, by určitě dobře ocenili, kdyby mohli o nich číst názory jiných spotřebitelů. Právě proto jsem se rozhodl implementovat do našeho systému něco podobného jako diskusní fórum, kde mají uživatelé možnost vyjádřit své názory, nebo reagovat na jiný názor, tedy diskutovat o daném produktu. Psát do diskusního fóra mají možnost jenom zaregistrovaní zákazníci, nezaregistrovaní uživatelé mohou jenom číst názory.

Diskusní zóna se nachází u výpisu detailů produktu, pod nakupovací zónou. Pro každý produkt je implementováno zvláštní diskusní fórum. Zde zákazník najde výpis seznamu názorů, které

zákazníci připsali k produktu. Tento seznam tvoří název názoru, jméno uživatele který názor přidal, a datum přidání. Je tady ještě kolonka, kde může uživatel označit které názory chce zobrazit. Uživatel může označit a zobrazovat jenom určité názory, nebo vybrat možnost zobrazit všechny názory. (viz. obrázek 5.7 a))

Po kliknutí na tlačítko přidat nový názor se před uživatelem objeví formulář obsahující předem vyplněné jméno uživatele, které samozřejmě může změnit, a položky které musí vyplnit, název a text názoru. Po stisknutí tlačítka odeslat systém zpracovává přidaný názor, a při správném vyplnění formuláře přidá diskuzi do databáze.



Obrázek 5.7 a): Zobrazené příspěvky – diskuze produktům



Obrázek 5.7 b): Reakce na příspěvky

Na aplikační úrovni zpracování formuláře začíná kontrolou vyplněných údajů. Po kontrole jsou tady dvě funkce, pro úpravy textu přidaného názoru. Velice důležitá je kontrola html znaků. V případě, že by uživatel přidal názor který obsahuje různé html znaky nebo kód, při zobrazení názoru by nám to možné způsobilo rozpad námi navržené struktury stránky. Pro chránění proti tomuto případu používám funkce htmlspecialchars, která převádí zvláštní znaky na HTML entity. Některé znaky mají v html zvláštní význam, a pokud si mají zachovat běžný význam, měly by být reprezentovány html entitami. Jedná se o překlad těchto znaků:

- '&' (ampersand) se stává '&'
- '"' (dvojitá uvozovka) se stává '"'
- "'" (jednoduchá uvozovka) se stává '''
- '<' (menší než) se stává '<'
- '>' (větší než) se stává '>'

Druhá kontrola kterou provádím je kontrola příliš dlouhých slov. Pokud by uživatel zadal takové dlouhé slovo, které se nevejde při výpisu na místo námi určené, taky by mohl způsobit rozpad stránky. Na to jsem implementoval funkci skratit_dlouhy, ve které jednoduše spočítám počet znaků

předaného řetězce, a kontroluji či nebyla nalezena mezera, nebo nebyl překročen dovolený limit délky slov a v případě potřeby přidám mezery.

Při úspěšné kontrole zadaných údajů, systém přidá nový názor do databázové tabulky diskuze, a přesměruje uživatele zpět na stránku produktu.

Výpis označených nebo všech příspěvků začíná vyhledáním názorů z databáze, které jsou „hlavními“ názory, tzn. nejsou reakcemi na jiný názor. Rozlišování jsou implementována na databázové úrovni, kde databázová tabulka diskuze má atribut reakce nastavená na hodnotu NULL, pokud je to „hlavní“ názor, a pokud je to reakce obsahuje identifikační číslo diskuze, na který navazuje jako reakce. Výpis jednotlivých názorů sestává z názvu příspěvku, jména uživatele který příspěvek přidal, z data přidání a z vlastního obsahu. Výpis reakcí je řešen rekurzivním voláním funkce diskuze_reakce, který po vyhledání v databázi vypíše jednotlivé reakce na daný příspěvek. Větší odsazení jednotlivých reakcí je řešeno předáním úrovně při volání této funkce, a tato úroveň určí danou velikost odsazení.

Tlačítko reagovat, při všech názorech, slouží uživateli na přidání reakcí na určitý názor. Pokud uživatel na to klikne, zobrazí mu ten jediný příspěvek na který reaguje, a pod tím stejný formulář jako při přidání nového příspěvku, jenom s rozdílem, že v tomto případě má předem vyplněný i název reakce, který nemůže změnit (viz. Obrázek 5.7 b)). Ve zpracování formuláře je rozdíl, že k databázovému dotazu INSERT připojíme i atribut reakce, nastavený na identifikační číslo názoru na který uživatel reaguje.

5.2.9 Zákaznický profil

Náš systém umožňuje zaregistrovaným a přihlášeným zákazníkům přístup do zákaznického profilu. Zákaznický profil sestává ze dvou částí. První je uváděný pod názvem moje objednávky. Zde najdou zákazníci všechny své odeslané objednávky, mohou se na ně podívat a hlavně kontrolovat jejich stav vyřízení. Druhou částí zákaznického profilu je „můj profil“, kde zákazníci mohou prohlížet a editovat své údaje, zadané při registraci.

» Home » Moje objednávky Na stránce zobrazíme záznamov.

Všetky objednávky - 19

Zobrazíme len objednávky

| 1 | **2** | [Nasledujúca >>](#)

Variabilný s. rast. / kles.	Dátum rast. / kles.	Stav rast. / kles.	Poznámka
0000000056	24.04.2007 20:54:11	nová	Objednavka prijata zo dn...
0000000055	24.04.2007 10:53:43	nová	Objednavka prijata zo dn...
0000000054	24.04.2007 10:32:10	nová	Objednavka prijata zo dn...
0000000040	17.04.2007 10:28:10	zásielka odoslaná	Objednavka prijata dna 1...
0000000038	13.04.2007 22:30:19	objednávka sa rieši	Objednavka prijata dna 1...
0000000037	13.04.2007 21:55:17	zásielka odoslaná	Objednavka prijata dna 1...
0000000035	12.04.2007 23:29:36	vyriešená	Objednavka prijata dna 1...
0000000034	12.04.2007 23:16:40	objednávka sa rieši	Objednavka prijata dna 1...
0000000028	11.04.2007 18:06:36	objednávka sa rieši	Objednavka prijata dna 1...
0000000027	11.04.2007 18:04:20	objednávka sa rieši	Objednavka prijata dna 1...

| 1 | **2** | [Nasledujúca >>](#)

Obrázek 5.8: Seznam všech objednávek zákazníka

Zákaznický profil najdou uživatelé v pravém bloku uživatelského rozhraní. Po vybrání volby „moje objednávky“ se před uživatelem objeví seznam všech dříve odeslaných objednávek. Tento seznam obsahuje variabilní symboly všech objednávek, datum odeslání objednávky, stav vyřízení a poznámka kterou připsal správce ke konkrétní objednávce. To je znázorněno i na obrázku 5.8. Zmíněný seznam se objeví pro uživatele v tabulkové podobě, kde jednotlivé řádky tedy jednotlivé objednávky jsou označeny jinými barvami. To není jenom kvůli designu, ale znamenají právě stav vyřízení jednotlivých objednávek. Barva bílá, tedy bez označení znamená vyřešenou objednávku, přičemž světle zelená znamená novou objednávku, světle červená právě řešenou objednávku a světle modrá znamená že zásilka již byla odeslaná. V horní části nad tabulkou objednávek jsou různé možnosti, kde může uživatel vybrat jestli chce zobrazit objednávky jenom aktuální, nové, vyřešené, odeslané nebo objednávky které jsou právě řešené. Je tady i filtr, kolik objednávek chceme najednou zobrazit na jedné stránce. Zákazníci mohou i hledat ve svých objednávkách dle různých klíčových slov. Zobrazené objednávky je možné seřadit dle určitých polí, a to dle variabilního symbolu, data a vzestupném nebo sestupném pořadí.

Z implementačního hlediska na úrovni aplikační vrstvy, při návštěvě této stránky zákazníkem systém vyhledá z databáze objednávky na danou relaci. To znamená že v superglobálním poli SESSION máme uložené identifikační číslo přihlášeného zákazníka, a právě to využíváme jako podmínku vyhledávacího dotazu. Tento dotaz je však ovlivněn mnoha jinými parametry. Tím myslím, že pokud uživatel vybere jenom určitý typ objednávek (nový, řešený, odeslaný, vyřešený), je tato podmínka také připojená k databázovému dotazu, aby systém hledal jenom určitý typ objednávek. Při změně počtu zobrazených objednávek je k dotazu připojena klauzule LIMIT, která určuje od které položky a do které má vypsát záznamy. Způsob implementace, vypočítání počtu stránek, určení hodnoty limitu, jsou stejné jako ty, které jsem rozepsal v kapitole 5.2.5 Nabídka zboží – výpis produktů. Informace o výběru typu objednávky, nebo i podmínky seřazení, jsou předány pomocí parametrů GET. Kontrolou těchto parametrů při sestavení dotazu jsou přidány i části klauzule WHERE a jsou určeny klauzule ORDER BY. Po zobrazení odkazů na další části seznamu objednávek – pokud je to třeba, jsou pomocí cyklu while vybírány jednotlivé řádky výsledku dotazu do asociativního pole, a postupně vypsány.

Po kliknutí na určitou objednávku systém vyhledá sestrojí a zobrazí zákazníkovi vybranou objednávku. Databázový dotaz spojí čtyři tabulky, košík, zboží, objednávky, zákazníci a vybírá z nich potřebné údaje. K tabulce objednávka je potřebná tabulka košík, která určí jaké zboží a v jakém množství je v objednávkách, tabulka zboží je potřebná, aby jsme znali ostatní parametry produktu, pak je potřebná ještě tabulka zákazníci, abychom mohli vypsát údaje o objednavatelích.

Druhá část zákaznického profilu je můj profil, kde jsou zákazníkovi vypsány jeho údaje, které si může prohlédnout a změnit. Je to vlastně předem vyplněný formulář, s údaji zákazníka které jsou získány z databáze. V dolní části je tady i možnost změnit heslo, kde uživatel může změnit své předem určené přihlašovací heslo.

Z aplikačního hlediska je to vlastně obsluha dvou formulářů, první s údaji zákazníka a druhý pro změnu hesla. Při stisknutí tlačítka změnit v prvním formuláři, systém zpracovává zadané, popřípadě změněné údaje a kontroluje je pomocí regulárních výrazů, velice podobným způsobem jak to bylo rozepsáno v kapitole 5.2.1 Registrace uživatele. Po úspěšné syntaktické kontrole systém opět testuje či nehrozí kolize duplicity u údajů uživatelské jméno a email. Pokud všechny tyto údaje byly úspěšné sestrojí dotaz UPDATE pomocí kterého změní údaje zákazníka v databázi. Při změně hesla je po kontrole zadaných údajů jenom přepsáno heslo příkazem UPDATE v databázi pomocí hašovací funkce MD5.

5.2.10 Administrátorské rozhraní

V každém informačním systému je implementována administrátorská část, která slouží pro administrativu a nastavení různých parametrů systému. V našem internetovém obchodě má administrátorská část klíčovou roli, kde správce vyřizuje objednávky, naplní zboží, nahrává obrázky, upravuje kategorie, prohlíží zákazníky popřípadě může upravit i jejich údaje, zadává různé akce, nastavuje novinky, zadává a určuje velikost různých slev, tedy má největší vliv na celý systém.

Po přihlášení se administrátorovi objeví i trochu jiné uživatelské rozhraní, není mu zobrazena pravá část ve které pro zákazníky byly zobrazeny novinky, uživatelský profil, vyhledávání a různé informace. Rozhodl jsem se tak jednak proto, že pro správce tyto věci nejsou tak potřebné, a zadruhé tak máme větší prostor na zobrazení různých tabulek, které obsahují mnoho údajů. Má úplně odlišné horní menu (viz. obrázek 5.9), zboží se mu objevují úplně jinak a má i barevné rozlišení od „normálního“ uživatelského rozhraní systému.



Obrázek 5.9: Horní menu pro správce

Objednávky

Správce je po přihlášení ihned přesměrován do oddílu systému v kterém jsou vypsané všechny objednávky. Vypadá to velice podobným způsobem jako pro jednotlivé zákazníky, v zákaznickém profilu „moje objednávky“, jak naznačuje i obrázek 5.8, s rozdílem, že pro správce jsou zde vypsané všechny objednávky které systém zpracovává, které jsou v databázi uchovány. Administrátor může taky nastavit požadavek zobrazení pouze určitých objednávek, aktuálních, nových, řešených, objednávky s odeslanou zásilkou nebo jenom vyřešené objednávky. Taky může nastavit limit na zobrazení počtu objednávek na jedné stránce. Administrátor má zde možnost hledat ve všech nebo ve vybraných objednávkách dle různých klíčových slov.

Aplikační zpracování skriptu je velmi podobné již rozepsanému v kapitole 5.2.9 Zákaznický profil. Volba různé podmínky zobrazení (zobrazení určitých typů objednávek) při zpracování předá o vybraném stavu informace pomocí GET parametrů, dle kterých je databázový vyhledávací dotaz přizpůsoben. Stejným způsobem je upravený i v případě vybrání určitých limitů, v dotazu pomocí

klauzule LIMIT, rozepsaná implementace použití techniky je v kapitole 5.2.5 Nabídka zboží – výpis produktů. Hledání je řešeno také pomocí GET parametrů, kterým je předán hledaný výraz, a podle toho upraven – sestrojen vyhledávací dotaz.

Po vybraní jedné položky ze seznamu objednávek, vidí před sebou administrátor prostor pro zadávání textu, kam může napsat poznámku k objednávce, která se pak objeví po přihlášení zákazníkovi. Pod tou může nastavit stav objednávky, pomocí rolovací listiny a potvrdit tyto změněné údaje tlačítkem ok. Po stisknutí tlačítka systém zpracovává zadanou poznámku a změní stav objednávky na vybraný stav. Je zde ještě výpis zákazníka nebo firmy, který objednávku poslal, po kliknutí na ně je přesměrován do zákaznické části, kde může prohlédnout údaje o něm, nastavit zákaznické slevy nebo jiné údaje. Podrobný popis zákaznické části viz. níže. V dolní části je vlastní výpis celé objednávky a všech detailů, které správce též může vytisknout.

Kategorie

Další z možností pro administrátora je pod názvem kategorie. Umožňují mu přehled kategorií, do kterých jsou zařazeny produkty našeho internetového obchodu. Zde může i přidat nové kategorie. Po stisknutí tlačítka přidat kategorie se mu objeví jednoduchý formulář, do kterého má vyplnit název, a popřípadě popis kategorie. Název nadřazené kategorie je závislý na tom, kde zvolil volbu přidat nové kategorie. Pokud při výpisu seznamu kategorií vybere možnost přidat kategorie, je automaticky nová kategorie hlavní a přidána do databáze. Pokud klikne na nově vytvořené kategorie a zde vybere možnost přidat kategorie, je kategorie zařazena jako podkategorie vybrané kategorie. Po výběru již existující kategorie objeví v předem vyplněném formuláři údaje o kategorii, které může změnit. Pod

Editovat kategorii "Štandard"


ID kategorie: 11

ID nadřazené kategorie: 10
(při hlavní kategorie o)

Název kategorie:


Patří do kategorie: **Vel'koplošné platne**

Popis kategorie:



Produkty v kategorií "Štandard"

ID produktu	Výrobné číslo	Název	Výrobca	Cena bez DPH	Zľava	Na sklade	Predávať
41	semme050505c	Semmelrock Záhradné platne, 5 cm, s fázou, (50/50/5 cm)		99,00 SKK/ks	0,00%	18757	ano
43	semme04040378	Semmelrock Picola - platňa z vymývateľného betónu, 3,7 cm, (40/40/3,7 cm)	Picola	74,60 SKK/ks	0,00%	12200	ano



Obrázek 5.10: Zobrazení dané kategorie

tímto formulářem je seznam kategorií, pokud do toho patří kategorie, nebo seznam zboží pokud do toho patří produkty (viz. obrázek 5.10). Odtud může být přesměrován na produkty.

Produkty

Jeden z hlavních úkolů pro správce je naplňování a úprava zboží. Zde je pro ně možnost produkty, kde je seznam všech produktů, ve kterém může hledat, filtrovat podle kategorií a seřadit. Při implementaci těchto funkcí je opět použito předání GET parametrů a podle těch je sestrojen dotaz. Po kliknutí na některé ze seznamu, se administrátorovi objeví rozpis všech detailů a všech vlastností daného produktu. K tomuto rozpisu se dostane administrátor i přes eshop menu v levé části stránky. To je vlastně předem vyplněný formulář s údaji o produktu, který může editovat. V dolní části tohoto formuláře jsou volby jako např. prodávat, přidat do akční nabídky nebo zobrazit jako novinku. Při výběru akční nabídky zobrazí další část formuláře, do které má správce zadávat akční cenu a od jakého a do jakého data akce platí.

Systémové zpracování tohoto formuláře začíná kontrolou všech zadaných výrazů, pomocí regulárních výrazů a jsou zde opět použity funkce `htmlspecialchars` a `skratit_dlouhy`. Význam těchto funkcí jsem rozepsal v kapitole 5.2.8 Diskuze – názory k produktům. Po úspěšné kontrole všech parametrů se sestrojí příkaz na přidání (protože ten stejný formulář se objeví i při přidání nového produktu, jen nebude předem vyplněný) nebo při editaci produktu. Po úspěšném vykonání dotazu je produkt přidán nebo editován v databázi.

Akce na databázové úrovni jsou implementovány zvláštní tabulkou, která pro identifikaci obsahuje identifikační číslo zboží, akční cenu a platnost. Nová položka se do této tabulky dostane při editaci nebo při přidání nové položky do tabulky zboží. Systém při zpracování přidání nebo editaci zboží kontroluje, jestli nebylo vyškrtuto políčko přidat do akční nabídky. Pokud to správce vyškrtl, kontrolujeme syntaktickou správnost zadaných dat a zadané akční ceny. Po syntaktické správnosti je kontrolován význam zadaných informací, např. či datum kterým platnost začíná není náhodou větší než kterým končí. Při úspěšné kontrole je při zpracování formuláře akce přidána do databáze.

Další kratší formulář pro administrátora slouží pro přidání obrázků k vytvořenému produktu. Po výběru obrázku systém kontroluje či byl vybrán opravdu obrázek a či velikost nepřekročí 1,5 MB. Větší obrázek systém neumožňuje nahrát. Při výběru možnosti „použít jako prioritní“ systém sáhne do databázové tabulky obrázky, a kontroluje jestli není již nastaven prioritní obrázek produktu, tzn. zda není atribut prioritně nastaven na hodnotu 1, při daném identifikačním čísle. Pokud tomu tak není, umožní další zpracování formuláře.

V dalším kroku zpracování získáme příponu obrázku rozdělením řetězce jména souboru dle tečky. Získanou příponu spolu s ostatními údaji o obrázku, tedy identifikačním číslem zboží, popisem a informacemi o prioritě, uložíme do databáze. Po úspěšném přidání získáme id přenosu, což je identifikační číslo obrázku v databázi a dle toho vytvoříme nový název obrázku, pod kterým bude umístěn na našem serveru. (např. `image_28.jpg`, `small_image_28.jpg`) Po sestrojení adresářové cesty

a jmeně obrázku zmenšíme velikost obrázku na danou velikost, protože kdyby správce nahrál obrázek s větším rozlišením, tak by uživatel při zobrazení produktu musel čekat dokud se ten velký obrázek systémem nenačte. Proto pomocí funkce `image_shrink_size` získáme nové rozměry obrázku pro zmenšení. Tato funkce má jako vstupní parametr název souboru, maximální šířku a maximální výšku výsledného obrázku a vrátí pole s výslednou šířkou a výškou. Pak pomocí další funkce `image_resize` převzorkují obrázek na získané hodnoty šířky a výšky. Tato funkce má vstupní parametry název zmenšovaného souboru, název výsledného souboru, šířku a výšku výsledného obrázku a vrátí `true` po úspěšném zmenšení a nahrávání obrázku. Při úspěchu všech předchozích akcí můžeme nahrát i veliký obrázek. Při neúspěchu některých operací popsanych v předchozích krocích, vymažeme i na začátku přidané informace o obrázku z databáze, aby to bylo v souladu s obrázky na serveru.

Pod těmito formuláři se administrátorovi objeví primární obrázek přidaný k produktu. Po kliknutí na primární obrázek systém zobrazí i ostatní obrázky k produktu. Zde je může správce smazat nebo zobrazit v plné velikosti po kliknutí na daný obrázek. Zobrazení těchto obrázků je možné i pro zákazníky, také po kliknutí na prioritní obrázek u produktu. Mohou si také prohlédnout obrázek v plné velikosti, narozdíl od správce je jenom, že v tomto případě oni to nemůžou obrázek smazat.

Pod tím je, ještě stále na té stejné stránce, zobrazeno i pro administrátora diskusní fórum k produktu. Má všechny oprávnění a možnosti, které měli zákazníci, popsané v kapitole 5.2.8 Diskuze – názory k produktům, rozšířený s možností smazat diskusní příspěvky, což zákazníci nemohli.

Akce a novinky

Pro správce jsou v horním menu další možnosti pro úpravy zboží, pomocí kterých má stálý přehled o novinkách a akcích. Po vybrání volby akce jsou mu zde rozepsány všechny aktuální a neaktuální, jenom naplánované nebo vypršelé akce. Po zaškrtnutí volby zobrazit jenom aktuální, systém zobrazí pouze právě platné akce. Seznam může dle všech vypsaných parametrů seřadit. Zde je i možnost smazat akce. Po výběru je přesměrován do oddílu produktu. Implementace je podobná jako jsem již několikrát zmínil v předchozích odstavcích, pomocí parametrů `get`. Podle těch je sestrojen vyhledávací dotaz. V sekci novinky si může správce prohlédnout všechny aktuální novinky, po případě je smazat.

» Zákazníci Na stránce zobrazíme záznamov.

Přehled zaregistrovaných zákazníků / celkový počet: 8

správce diskuzie

ID zákazníka	Firma	Kontaktní osoba	Email	Počet objednávek	Dátum registrácie
rast. / kles.	rast. / kles.	rast. / kles.	rast. / kles.	rast. / kles.	rast. / kles.
13		tomi	tmeszaros@seznam.cz	0	17.04.2007 23:44:23
15		Odon	odon@seznam.cz	0	19.04.2007 21:56:21
16		Jozef	jozef.laba@seznam.cz	6	19.04.2007 22:01:42
1	ALBA - Novy s.r.o	Karel Nový	k.novy@seznam.cz	19	31.03.2007 11:38:57
4	ALSECOO	Jana Mala	j.mala@seznam.cz	0	31.03.2007 11:38:57
2	ASTEEL	Jozef Horváth	m.kovar@seznam.cz	9	31.03.2007 11:38:57
10	Czibor feri	feri s.r.o	czibor@seznam.cz	1	31.03.2007 12:05:49
3	Spojmat s.r.o	Ladislav Vesely	l.vesely@seznam.cz	4	31.03.2007 11:38:57

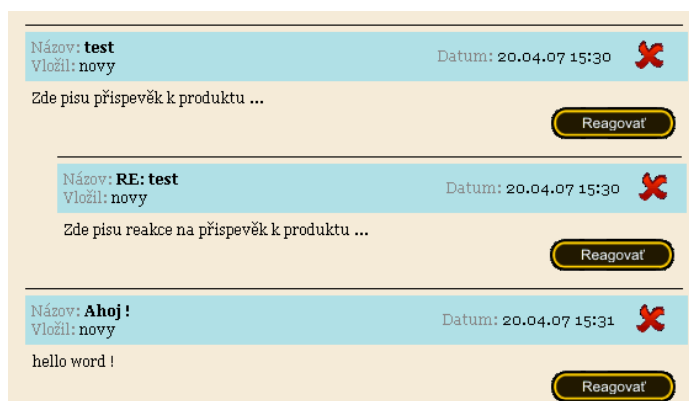
Obrázek 5.11: Přehled zákazníků

Sekce zákazníci

V sekci zákazníci správce může vidět jaké zaregistrované zákazníky má náš internetový obchod. Seznam obsahuje nedůležitější údaje o nich a zároveň i počet jejich objednávek. Dle všech parametrů může seřadit a může i hledat v databázi zákazníků. V seznamu má správce diskuze barevné odlišení (viz. obrázek 5.11). Po vybrání jedné položky se před správcem objeví výpis, který sestává ze tří částí. První část obsahuje údaje o zákazníkovi, který popřípadě může změnit, zde může zadat / změnit i slevu pro zákazníka nebo může zákazníka vymazat. V druhé části může změnit jeho přihlašovací heslo. V třetí části má výpis všech objednávek daného zákazníka, který vypadá podobným způsobem jako všechny objednávky systému.

Správce diskusních příspěvků

Systém umožňuje ještě jeden typ přihlašovacích účtů, který je správcem diskusních příspěvků. To je pouze jednoduchý zákaznický účet, který má všechny práva a možnosti jaké měli zákazníci. Tyto práva jsou zde rozšířena o možnost vymazání diskusních příspěvků. Diskuze jednotlivých uživatelů vidí stejně jako ostatní, ale objevuje se mu zde tlačítko vymazat, kterým vymaže jeden příspěvek, nebo celou část diskuze. (viz. obrázek 5.12)



Obrázek 5.12: Diskuze pro správce diskuze

V této kapitole jsem se zabýval vlastní implementací systému. Snažil jsem zde rozepsat celou implementaci systému na databázové, aplikační a taky na uživatelské úrovni. Vysvětloval jsem jak jsou řešeny jednotlivé funkční části systému a jak jsou tyto tři úrovně na sebe navázány. Snažil jsem se zde popsat i použité techniky, a techniky pomocí kterých systém pracuje s daty uloženými v databázi, pak to zpracovává na aplikační úrovni, a vlastní implementace uživatelského rozhraní systému. Zmínil jsem se o zpracování dat systémem, který zadal uživatel, o zabezpečení systému a databáze. Kapitola je strukturována způsobem, kdy je na začátku je popis implementace databázové struktury, a v další části kapitoly jsou rozděleny jednotlivé implementované funkce systému do několika podkapitol.

6 Další rozvoj systému

Po implementaci každého informačního systému vznikají motivace a nápady na další rozvoj, další rozšíření systému. Tím jsem se zabýval i v kapitole 2.3.1 Životní cyklus informačního systému. V poslední fázi životního cyklu vývoje IS, tedy ve fázi užití, po plánování, analýze, návrhu a implementace opět vznikají impulzy pro vylepšení systému a proto se tento cyklus neustále opakuje. Právě tyto impulzy bych rád shrnul v této předposlední kapitole mé bakalářské práce.

Prvním z možných rozšíření systému by mohlo být rozšíření možnosti vyhledávání. Teď je v systému implementováno jednoduché vyhledávání, které vyhledává dle zadaných klíčových slov ve všech kategoriích mezi všemi druhy zboží a ve všech attributech zboží. Při mnohem větší skladové zásobě nabízených produktů našeho internetového obchodu, by to určitě způsobilo nějaké těžkosti určitým zákazníkům, kteří spěchají a chtějí rychle najít produkty, které hledají. Představil bych rozšířenou vyhledávací část systému, v které by zákazníci mohli určit v jakých kategoriích, jaké typy produktů s jakými parametry atd. hledají, a přizpůsobit k tomu i vyhledávání v databázi

Dalším takovým rozšířením by mohla být úprava nabídky zboží. Tato možnost úzce souvisí s předchozími možnostmi, jen ne na vyhledávací úrovni. Myslím lepší rozpis nabízeného zboží. Teď jsou zařazeny pouze do kategorií dle různých typů stavebních materiálů. Zařazení do různých kategorií by mohlo být rozšířeno i dle jiných kritérií, např. dle výrobců, ne jenom podle typu materiálu. To by možná ulehčilo a zjednodušilo přehled pro zákazníka nad nabízenými produkty.

Zde bych ještě zařadil možné rozšíření administrátorské části systému. Tím myslím lepší správu systému, zboží a kategorií. Práci administrátora by určitě ulehčilo, kdyby nemusel každý produkt jednotlivě přidávat do systému. Dobrá by byla nějaká možnost systému, který by umožňoval přidat produkty hromadně. Představoval bych si to nějakým importováním z často použitých databázových programů nebo tabulkových editorů, v kterých by správce měl prodávané zboží. Určitě by ještě správce ocenil nějaké rozšířené statistiky, který by ukázaly jaké produkty jsou např. nejprodávanější.

Tyto možnosti rozšíření implementace plánuji do budoucna určitě realizovat, pokud by tento systém dobře uspěl a úspěšně fungoval pro uživatele který požadavky vkládal, pro kterého tento systém vznikl.

7 Závěr

Navrhovaný a implementovaný systém který jsem vytvořil jako bakalářskou práci splňuje všechny požadavky uvedené v zadání, a také splňuje podavky vznikající u uživatele, pro kterého byl internetový obchod vytvořen. Systém poskytuje příjemné, jednoduché a přehledné uživatelské rozhraní pro zákazníka, kde může elektronicky nakupovat, tedy objednat produkty. Může ve svém profilu zkontrolovat stav svých objednávek. Ve správcovské části systém pro administrátora umožňuje správu celého systému, jako např. správu objednávek, zboží, kategorií a zadávání různých slev. Systém při nejdůležitějších událostech zasílá informace emailem pro zákazníka a pro administrátora systému. Myslím tím události registrace nového zákazníka, odeslání nové objednávky a zvláště důležitý je v případě zapomenutí hesla.

Pomocí této bakalářské práce jsem získal mnoho nových znalostí v implementaci a vlastně v celém cyklu vytvoření informačních systémů. Na úrovni databáze s vytvořením složitých SQL dotazů a příkazů jsem rozšířil mé znalosti v oblasti obsluhy databáze. Dále jsem se při implementaci seznámil s mnoha novými a zajímavými možnostmi skriptovacího jazyka PHP o kterých jsem dosud ani nevěděl. Při tvorbě webového uživatelského rozhraní jsem se naučil pokročilé techniky kaskádových stylů, jako vytváření layoutů a formátování prvků webového rozhraní. Taky jsem získal mnoho zkušenosti se skriptovacím jazykem JavaScript na straně klienta.

Uživatelské rozhraní systému bylo testováno pod různými prohlížeči. Pod těmito prohlížeči se vyskytly různé problémy, z důvodu nekompatibility a odlišného interpretování kódu prohlížeči. Tyto chyby byli úspěšně opraveny, implementace modifikována tak aby to vyhovělo všem testovaným prohlížečům a odpovídal i validátorům.

Systém byl testován i z hlediska použití, různými typy uživatelů, či ovládání systému zvládnou i méně zkušené uživatele. Během dosavadních testů nebyly zjištěny problémy. Vlastní testování systému probíhá ještě i teď, protože ještě nejsou vyřízeny odeslané objednávky. Zjišťování správné funkčnosti bude probíhat dokud systém nebude instalován na provozní server a umístěn do provozu, kde již firma bude vyřizovat objednávky.

S ohledem na to, že na systému by ještě mohly být upraveny, rozšířeny jednotlivé funkce, právě které jsem uvedl v kapitole 6, a které by dostatečně rozšířily systém, myslím si, že i tak v dostatečné míře odpovídá vlastním cílem, a dá se použít jako internetový obchod pro prodej stavebních materiálů.

Literatura

- [1] Williams, H. E., Lane, D.: *PHP a MySQL – Vytváříme webové databázové aplikace*. Computer Press, 2002, 552 s. ISBN 8072267604
- [2] Prof. Ing. Hruška, T. CSc.: *Pojem informačního systému*. – studijní opora, Brno 2006
- [3] Polák, J., Merunka, V., Carda, A.: *Umění systémového návrhu*. Grada, Praha 2003, ISBN 80-247-0424-02
- [4] Kendall, K.E.: *System Analysis and Design*. Prentice Hall, 1991, ISBN 01-31454-55-2
- [5] Kosek, J.: *HTML, tvorba dokonalých www stránek*. Praha: Grada Publishing, 1998, 291 s. ISBN 80-7169-608-0
- [6] Ullman, L.: *PHP a MySQL*. Brno: Computer Press, 2004, 536 s. ISBN 8025100634
- [7] Budd, A., Moll, C., Collison, S.: *CSS filtry, hacky a pokročilé postupy*. Brno: Zoner Press, 2007, ISBN 978-80-86815-54-1
- [8] Mihule, T.: *Internetový obchod pomocí PHP a MySQL*. Praha: NEWSLETTER, 2003, ISBN: 80-86394-97-2
- [9] Krajčík, V.: *Životní cyklus projektů informačních systémů*. Dokument dostupný na URL: http://portal.vsp.cz/files/casopis/zivotni_cyklus.pdf, 15.03.2007
- [10] Ing. Foulík, R.: *Elektronický obchod*. Dokument dostupný na URL: www.e-studio.cz/dokumenty/elektronicky_obchod.pdf, 15.03.2007
- [11] The PHP Group: *PHP: Hypertext Preprocessor*. Dokument dostupný na URL: www.php.net, 21.04.2007
- [12] Janovský, D.: *O tvorbě internetových stránek*. Dokument dostupný na URL: <http://www.jakpsatweb.cz/>, 21.04.2007
- [13] MySQL AB: *MySQL Documentation*. Dokument dostupný na URL: <http://www.mysql.com/>, 21.04.2007
- [14] Větrovská P.: *Tvorba WWW stránek*. Dokument dostupný na URL: <http://www.webtvorba.cz/>, 17.03.2007
- [15] Kraval, I.: *Zásady pro řízení projektů tvorby IS*. Dokument dostupný na URL: <http://www.objects.cz>, 17.03.2007
- [16] *Wikipedie otevřená encyklopedie*. Dokument dostupný na URL: <http://cs.wikipedia.org/>, 17.03.2007

Seznam příloh

Příloha 1. CD