

**Univerzita Pardubice
Fakulta elektrotechniky a informatiky**

Univerzální šablony pro produkt In-Shop 4 společnosti ZONER

Tomáš Havránek

**Bakalářská práce
2008**

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Katedra informačních technologií
Akademický rok: 2007/2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš HAVRÁNEK**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**

Název tématu: **Univerzální šablony pro produkt In-Shop 4 společnosti Zoner.**

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

V teoretické části aplikace budou zhodnoceny přístupy při tvorbě šablon pro univerzální systémy. Dále budou přestaveny a zhodnoceny principy tvorby přístupných stránek s důrazem kladeným na efektivní rozložení funkčních prvků. V implementační části bude vytvořena sada univerzálních šablon pro produkt In-Shop4 společnosti ZONER. Při tvorbě šablon bude dbáno na optimální rozložení funkčních prvků a na validitu vůči požadovaným standardům.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

LEE, J., WARE, B. Open Source - vývoj webových aplikací (Linux, Apache, MySQL, PHP a Perl). Computer press, 2003.

Gilmore W.J, Velká kniha PHP 5 a MySQL, Zoner press, 2005

Vedoucí bakalářské práce:

Ing. Lukáš Čegan

Ústav elektrotechniky a informatiky

Datum zadání bakalářské práce:

30. listopadu 2007

Termín odevzdání bakalářské práce:

16. května 2008



A handwritten signature in blue ink, which appears to be "Simeon Karamazov".

doc. Ing. Simeon Karamazov, Dr.

děkan

V Pardubicích dne 29. dubna 2008

SOUHRN

Tato bakalářská práce se zabývá tvorbou univerzálních šablon pro produkt Zoner In-Shop 4. Její první část je věnována obecně technologiím a možnostem e-shopů. Ve druhé části je práce zaměřena na tvorbu šablon pro In-Shop, konkrétně šablon určených pro mobilní zařízení.

KLÍČOVÁ SLOVA

In-Shop, šablona, mobilní web, e-shop

TITLE

Universal templates for In-Shop 4 product of Zoner company.

ABSTRACT

This bachelor work deals with creation of universal templates for Zoner In-Shop 4. First part of work generally describes technologies which e-shops uses and its possibilities. Second part is aimed to concrete creation of In-Shop templates for mobile devices.

KEYWORDS

In-Shop, template, mobile web, e-shop

OBSAH

1	ÚVOD	9
2	E-SHOP	10
2.1	TECHNOLOGIE VYUŽÍVANÉ E-SHOPY	11
2.1.1	HTML (Hypertext Markup Language)	11
2.1.2	XHTML (Extensible Hypertext Markup Language)	12
2.1.3	CSS (Cascading Style Sheets).....	12
2.1.4	PHP (Hypertext Preprocessor)	14
2.1.5	ASP (Active Server Pages)	15
2.1.6	ASP.NET.....	15
2.1.7	JSP (JavaServer Pages)	15
2.1.8	Webové servery	16
2.1.9	Databáze, databázové servery.....	16
3	E-SHOPY NA TRHU	18
3.1	ŘEŠENÍ NA MÍRU	18
3.2	KRABICOVÉ ŘEŠENÍ	18
3.3	VOLNĚ DOSTUPNÁ ŘEŠENÍ.....	19
3.4	KONKRÉTNÍ KOMERČNÍ ŘEŠENÍ.....	20
3.4.1	Zoner In-Shop 4.....	20
3.4.2	oXySHOP	20
3.4.3	KarsaV2.....	20
3.5	KONKRÉTNÍ NEKOMERČNÍ ŘEŠENÍ.....	20
3.5.1	osCommerce.....	20
3.5.2	Quick.Cart.....	21
3.6	ZHODNOCENÍ DOSTUPNÝCH ŘEŠENÍ NA TRHU	21
4	MOBILNÍ ZAŘÍZENÍ A WEBOVÉ STRÁNKY	23
4.1	STANDARDY MOBILNÍHO WEBU	24
4.1.1	XHTML Basic 1.0, XHTML Mobile Profile 1.2.....	24
4.1.2	CSS Mobile Profile 2.0.....	24
4.2	OMEZENÍ MOBILNÍCH ZAŘÍZENÍ VE ZOBRAZENÍ WEBU	24
4.3	ZPŘÍSTUPNĚNÍ WEBOVÝCH STRÁNEK PRO MOBILNÍ ZAŘÍZENÍ	25
4.3.1	Zpřístupnění vytvořením CSS stylů pro různá zařízení.....	25
4.3.2	Zpřístupnění vytvořením (webových) šablon pro různá zařízení.....	26
4.4	WEBDESIGN VHODNÝ PRO MOBILNÍ STRÁNKY	28
4.4.1	Layout stránek.....	28
4.4.2	Rozmístění prvků na stránce.....	28
4.4.3	Čeho se vyvarovat	29
4.4.4	Obsah stránek.....	30

5	ŠABLONY ZONER IN-SHOP 4	31
5.1	CO JSOU ŠABLONY V IN-SHOPU	32
5.2	PRINCIP ZPRACOVÁNÍ ŠABLON	33
5.3	TVORBA ŠABLON	33
5.4	STANDARDNÍ ŠABLONY V MOBILNÍM PROHLÍŽEČI.....	34
6	ŠABLONY ZONER IN-SHOP 4 URČENÉ PRO MOBILNÍ ZAŘÍZENÍ	36
6.1	LAYOUT	36
6.1.1	<i>Hlava</i>	36
6.1.2	<i>Hlavní blok</i>	37
6.1.3	<i>Blok kategorie a vyhledávání</i>	37
6.1.4	<i>Blok navigace</i>	37
6.1.5	<i>Pata</i>	37
6.1.6	<i>Blok přihlášeného zákazníka</i>	37
6.2	IMPLEMENTACE	38
6.2.1	<i>Šablona Layout.ascx</i>	38
6.2.2	<i>Šablona MobileNavigation.ascx</i>	40
6.2.3	<i>Problém kotev a tagu base</i>	40
6.2.4	<i>Stylopisy</i>	41
6.3	POROVNÁNÍ STANDARDNÍCH A MOBILNÍCH ŠABLON	41
7	ZÁVĚR.....	43

SEZNAM OBRÁZKŮ A TABULEK

OBRÁZEK 1: PŘÍSTUP KLIENTŮ Z RŮZNÝCH ZAŘÍZENÍ NA WEBOVÝ SERVER	23
OBRÁZEK 2: ZONER IN-SHOP MANAGER 4 - NASTAVENÍ VZHLEDU PRODEJNY	32
OBRÁZEK 3: MOŽNÉ ZPŮSOBY ROZVRŽENÍ STRÁNEK E-SHOPU ZONER IN-SHOP 4	32
OBRÁZEK 4: STRUKTURA ADRESÁŘE INSHOP/LAYOUT	34
OBRÁZEK 5: VYKRESLENÍ STANDARDNÍ ŠABLONY V PROHLÍŽEČI MOZILLA FIREFOX	34
OBRÁZEK 6: VYKRESLENÍ STANDARDNÍ ŠABLONY V PROHLÍŽEČI MS INTERNET EXPLORER MOBILE... ..	35
OBRÁZEK 7: ROZVRŽENÍ STRÁNKY E-SHOPU UPRAVENÉ PRO MOBILNÍ ZAŘÍZENÍ	37
OBRÁZEK 8: BLOK NAVIGACE ZOBRAZENÝ V PROHLÍŽEČI	40
TABULKA 1: POROVNÁNÍ VYBRANÝCH ŘEŠENÍ PRO E-SHOP	22
TABULKA 2: VÝSLEDKY TESTU PROHLÍŽEČŮ VE VYKRESLOVÁNÍ STRÁNKY	26

SEZNAM ZKRATEK

ASP	active server pages (aktivní serverové stránky)
BSD	Berkeley software distribution (Berkeley softwarová distribuce)
CSS	cascading style sheets (tabulky kaskádových stylů)
DTD	document type definition (definice typu dokumentu)
GPL	general public licence (všeobecná veřejná licence)
HTML	hypertext markup language (hypertextový značkovací jazyk)
HTTP	hypertext transfer protocol (protokol pro přenos hypertextu)
IIS	internet information services (informační služby internetu)
JSP	javaserver pages (javaserver stránky)
LGPL	lesser general public licence (menší všeobecná veřejná licence)
MP	mobile profile (mobilní profil)
PDA	personal digital assistant (osobní digitální pomocník)
PHP	hypertext preprocessor (hypertextový preprocesor)
SQL	structured query language (strukturovaný dotazovací jazyk)
XHTML	extensible hypertext markup language (rozšiřitelný hypertextový značkovací jazyk)
XML	extensible markup language (rozšiřitelný značkovací jazyk)

1 ÚVOD

Žijeme v době, kterou doslova ovládá Internet. Do prostředí Internetu se přesouvá stále více služeb, které dříve nebylo možné používat „bez kontaktně“, jako je to možné dnes. Pro příklad uvedu Internetové bankovníctví, elektronické obchody nebo systémy studijní agendy. Také dosažitelnost Internetu je dnes na vysoké úrovni, stává se standardem, že v kavárnách, restauracích, letištních i nádražních halách jsou instalované WiFi hotspoty, přes které se lze zdarma bezdrátově připojit k Internetu.

I uživatelé mobilních zařízení jako jsou mobilní telefony a přenosné počítače typu PDA jejichž popularita a prodeje v posledních letech rostou, chtějí služeb Internetu využívat naplno. Bohužel jejich zařízení si se standardním webem poradí jen ztěžka.

Tato bakalářská práce se zabývá tvorbou univerzálních šablon pro aplikaci In-Shop 4 firmy Zoner, která slouží k tvorbě a provozování elektronického obchodu. Jejím cílem je úprava šablon, která přizpůsobí stránky e-shopů tak, aby bylo možné jej bezproblémově používat i v mobilních zařízeních.

V následující kapitole je vysvětlen pojem e-shop a popsány některé technologie používané e-shopy. Třetí kapitola poskytuje stručný přehled dostupných řešení elektronických obchodů na trhu. Ve čtvrté kapitole jsou popsána omezení mobilních zařízení pro mobilní web a přístup vhodný pro tvorbu mobilního webu. Další kapitola se zabývá standardními šablonami In-Shop 4. Šestá kapitola popisuje návrh a implementaci šablon vhodných pro zobrazení v mobilních zařízeních.

2 E-SHOP

E-shop, virtuální nebo také elektronický obchod, je obchod, který je možné nalézt na Internetu a provést v něm nákup stejně jako v obchodě kamenném. Tento může existovat i ve fyzické podobě, ale není to nezbytně nutné.

Virtuální obchod je webovou aplikací, která umožňuje obchodní styk mezi prodávajícím a kupujícím. Jedná se v podstatě o virtuální katalog zboží nabízeného obchodníkem. Katalog lze volně procházet, sledovat parametry zboží a případně vybrané zboží vložit do virtuálního košíku. Košík v elektronickém obchodě v podstatě supluje klasický nákupní košík známý z každé samoobsluhy. To znamená, že je jakýmsi dočasným úložištěm výrobků, které si zákazník vybral a později je může závazně objednat. Z košíku lze samozřejmě zboží vrátit pokud si jeho nákup zákazník během procházení obchodu rozmyslí, nebo najde zboží jemu více vyhovující. Ve chvíli, kdy kupující dokončí výběr, přistoupí k virtuální pokladně, u které dokončí objednávku potvrzením výběru zboží, výběrem způsobu platby a dodání, případně udáním doručovací adresy, kam má být vyřízená objednávka zaslána. Takto zadaná objednávka je následně automaticky e-shopem zaslána obchodníkovi k jejímu vyřízení.

K vytvoření takto pracujícího obchodu jsou zapotřebí technologie, které podporují dynamické generování HTML stránek a zachování stavů mezi jednotlivými akcemi uživatelů. Značkovací jazyk HTML, ve kterém je většina dnešních webových stránek vystavěna, dynamické generování nenabízí a protokol HTTP¹, používaný pro výměnu hypertextových dokumentů ve formátu HTML (s rozšířením MIME² lze přenášet jakýkoliv soubor), je koncipován jako bezstavový. Z těchto důvodů je nutné při tvorbě obchodu použít jiné prostředky splňující požadavky popsané výše. Těmito prostředky jsou například PHP, ASP, ASP.NET, JSP.

¹ Hypertext Transfer Protocol – internetový protokol, verze 1.1 definovaná předpisem RFC 2616

² Multipurpose Internet Mail Extensions – internetový standard, který rozšiřuje standard RFC 822 o multimediální možnosti posílaných zpráv, popsany doporučením RFC 1341

2.1 TECHNOLOGIE VYUŽÍVANÉ E-SHOPY

Vlastní virtuální obchod je v podstatě soubor webových stránek, které jsou generovány na webovém serveru s pomocí nějaké technologie na serveru implementované. Každý e-shop také využívá databázi, ve které jsou uložena data potřebná k jeho provozu. V této kapitole bych chtěl některé základní technologie využívané pro tvorbu a provoz elektronických obchodů představit a popsat.

2.1.1 HTML (Hypertext Markup Language)

HTML v aktuální verzi 4.01 je značkovací jazyk odvozený od univerzálního značkovacího metajazyka SGML (Standard Generalized Markup Language). HTML se používá k tvorbě (popisu struktury) stránek WWW (World Wide Web). Tyto dokumenty jsou složeny z prvků (elementů), které jsou obaleny tzv. tagy nebo-li značkami, jež jim dávají určitý význam. Značky se uzavírají mezi ostré závorky, mohou být párové nebo nepárové a mohou mít své atributy, pomocí nichž jsou ovlivňovány další vlastnosti prvků. Pokud je tag párový, potom uzavírací značka je shodná s počáteční, pouze před názvem obsahuje znak lomítka. Atributy značek nejsou povinné a jsou uvedeny hned za značkou oddělené mezerou. Párový tag pro odstavec s atributy obalující prvek potom může vypadat takto:

```
<p color="black" align="left">text prvku</p>
```

Struktura HTML dokumentu začíná deklarací DTD (Document Type Definiton), ve které je uveden odkaz na definici určité verze HTML používané v dokumentu. Příklad deklarace DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Následuje tag `<html>` obalující celý dokument a ten je uvnitř dále členěn na hlavičku `<head>` `</head>` a tělo dokumentu `<body>` `</body>`. Hlavička obsahuje metadata týkající se celého dokumentu, která určují např. kódování, název dokumentu, jazyk, stylovací soubory. Tělo obsahuje samotný popis prvků dokumentu, pomocí kterého jsou následně zobrazeny webovým prohlížečem. Dokument je ukončen uzavíracím tagem `</html>`.

Některé základní tagy HTML:

- `<h1>`, `<h2>` až `<h6>` nadpisy 1. až 6. úrovně (párové),
- `<p>` odstavec textu (párový),
- `
` zalomení řádku (nepárový),
- `<table>` tabulka (párový),
- `<tr>` řádek tabulky (párový),
- `<td>`, `<th>` buňka tabulky, th - hlavičková buňka (párový),
- `<div>` obecný kontejner, může obalit libovolný obsah, používá se v kombinaci s kaskádovými styly pro vytvoření grafického layoutu dokumentu, zalomí řádek před i za sebou (párový),
- `` obecný kontejner, na rozdíl od `<div>` nezalamuje řádky (párový).

2.1.2 XHTML (Extensible Hypertext Markup Language)

Extensible Hypertext Markup Language je značkovací jazyk, jež se měl stát nástupcem HTML. Vyvinut byl přepracováním HTML tak, aby vyhovoval standardům XML dokumentů a byl zpětně kompatibilní. Aktuální verze XHTML je 1.1.

XHTML je oproti HTML přísnější, vyžaduje ukončování všech tagů, párových i nepárových. Párové tagy ukončujeme stejně jako v HTML, tedy `<p>odstavec</p>` a v případě nepárového vložení mezery a lomítka za značku. Takový tag potom vypadá takto: `
`. XHTML také nenabízí žádnou možnost formátování. Pro definování vzhledu dokumentu se v kombinaci s XHTML používají CSS. Dalším rozdílem je case-sensitivita XHTML a z toho plynoucí povinnost psát všechny značky a atributy malým písmem. XHTML také vyžaduje, aby hodnoty atributů byly uvedeny v uvozovkách. Dokumenty začínají XML deklarací, ve které je uvedena verze XML a kódování dokumentu, např.:

```
<?xml version="1.0" encoding="windows-1250" ?>
```

Deklarace XML se nemusí použít, pokud je stránka kódována v UTF-8. Následuje deklarace DTD a po té již struktura známá z HTML.

2.1.3 CSS (Cascading Style Sheets)

Pro formátování webových stránek se používají CSS. Cascading Style Sheets přeloženo do češtiny znamená tabulky kaskádových stylů nebo zkráceně kaskádové styly. Současná verze CSS je 2.1. Formátováním ve smyslu CSS je myšleno např. přiřazení barvy textu, odsazení okrajů odstavce, definice pozadí či ohraničení

prvku nebo určení pozice prvku. Kaskádové styly nabízejí větší možnosti formátování než HTML, umožňují také jednodušší změnu stylu dokumentu v porovnání s úpravou HTML. S využitím kaskádových stylů je možné oddělit obsah dokumentu od jeho stylu, což v kombinaci s HTML přináší možnost jednodušší správy webových stránek a v XHTML dokonce jedinou možnost stylování dokumentu.

Definice stylu se skládá ze dvou částí: selektoru (název elementu, třídy, identifikátoru) a deklarace, jež je uzavřena ve složených závorkách. Selektor tedy označuje element (např. `<p>`), třídu nebo identifikátor, pro který je deklarace stylu určena.

```
selektor { vlastnost: hodnota }
```

Kaskádové styly lze použít třemi způsoby. Můžeme je vepsat přímo k tagům, které chceme ovlivnit s využitím atributu *style*. Jako hodnotu atributu *style* píšeme deklaraci stylu, například:

```
<p style="color: red">cerveny text</p>
```

Text v takto deklarovaném odstavci bude mít červenou barvu. Další možností je použít *stylopis*, což je definice stylů přímo v dokumentu. *Stylopis* se vkládá mezi tagy `<style>` `</style>` a styly v něm se definují standardně. Definice odstavců s červeným textem pak vypadá takto:

```
<style>
  p { color: red }
</style>
```

Poslední možností je využití externího souboru kaskádových stylů. Jedná se o soubor s příponou *.css*, kde jsou definovány styly používané ve webové stránce. Styly se v souborech definují stejně jako ve *stylopisu* s tím rozdílem, že se nepoužívá tag `<style>`. Soubor se připojuje k webové stránce HTML tagem `<link />`, který se umísťuje do hlavičky dokumentu. Příklad připojení souboru *style.css*:

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

Selektor, určující element, nastavuje jeho vlastnosti ve všech výskytech. Například `p { color: red }` (`p` je selektor, `color: red` je deklarace vlastnosti) nastaví všem odstavcům v dokumentu červenou barvu textu. Pokud potřebujeme u několika elementů stejného typu jiné vlastnosti použijeme tzv. třídu elementu. Název třídy se od ele-

mentu odděluje tečkou. Takovýto selektor určuje vlastnosti onoho elementu, který má navíc jako jeden atribut *class*, jehož hodnota odpovídá názvu třídy v selektoru. Třída může být také definována i bez návaznosti na určitý element, potom je selektor pouze název třídy s tečkou na začátku a ovlivňuje formát kteréhokoliv prvku odpovídající třídy. Obdobně pokud selektor začíná znakem #, ovlivňuje vlastnosti prvku s identifikátorem (atributem *id*), kteréhož hodnota se shoduje se selektorem bez křížku. Použitím identifikátorů a tříd lze dosáhnout různého vzhledu u elementů stejného typu, například:

```
h1.zlutý { color: yellow }
h1.modrý { color: blue }
```

S pomocí tohoto stylopisu můžeme správným nastavením třídy elementu vytvořit dva různé nadpisy první úrovně. Jeden se žlutým a druhý s modrým textem. Selektory i deklarace je možno spojovat, tzn., že pro několik různých selektorů můžeme mít deklarované stejné vlastnosti a deklarovat můžeme i více než jednu vlastnost. V takovém případě oddělujeme selektory čárkou a deklarace středníkem jako zde:

```
h1,h2,h3,h4,h5,h6 {
  color: red;
  background-color: green;
  text-decoration: underline;
}
```

Takto definovaný styl vykreslí nadpisy v dokumentu červeným potrženým písmem se zeleným pozadím. Vlastnosti elementů definované pomocí CSS se také v dokumentu dědí od rodičovských elementů, ale pouze v případě, kdy nejsou předefinované. Tedy, pokud v dokumentu zanořím do nadpisu 1. úrovně (element `<h1>`), jehož styl jsem definoval výše, element `<i>` (kurzíva) i text tohoto prvku bude červený, podtržený se zeleným pozadím a navíc psaný kurzívou.

2.1.4 PHP (Hypertext Preprocessor)

PHP je skriptovací programovací jazyk primárně zaměřený na tvorbu dynamicky generovaných webových stránek. Jeho syntaxe je převzata z C, Perlu a Javy. Skripty se vepisují přímo do HTML (XHTML) dokumentu a jsou vykonávány na straně serveru, následně jsou klientovi odeslány jen výsledky jejich činnosti ve formě webové stránky. PHP také umožňuje připojení různých databázových serverů, práci se soubor-

ry na serveru, zachování stavu mezi jednotlivými požadavky klienta vytvářením tzv. sessions (sezení). Ještě bych vyzdvihl platformní a serverovou nezávislost PHP a to, že je Open Source. Nejnovější verze je 5.2.5.

2.1.5 ASP (Active Server Pages)

ASP je skriptovací technologie nezávislá na programovacím jazyce, která byla vyvinuta firmou Microsoft. Nejčastěji používané jazyky v ASP jsou VBScript (obdoba Visual Basic) a JScript (Microsoftem upravený JavaScript). Pomocí ASP je možné podobně jako v PHP tvořit dynamické webové stránky, tzn., že skripty se vepisují přímo do HTML kódu. S ASP je také možné využívat relační databáze, vytvářet sessions a další. K dosažení plné funkčnosti ASP je nutné použít webový server Microsoft IIS (Internet Information Services), což omezuje na využití serverových operačních systémů společnosti Microsoft.

2.1.6 ASP.NET

ASP.NET je moderní technologie společnosti Microsoft a byla vyvinuta jako nástupce ASP. ASP.NET je součástí platformy .NET Framework (aktuálně ve verzi 3.5), to je jakási nadstavba operačního systému. Tato nadstavba obsahuje balík knihoven s jednotným prostředím, který nabízí programátorům objektové třídy pro využití ve svých programech nezávisle na programovacím jazyce. Proto lze s .NET resp. ASP.NET využít libovolný vyšší programovací jazyk uzpůsobený pro .NET. Microsoft pro .NET vyvíjí dva hlavní jazyky: VB.NET (Visual Basic.NET) a C#. Tyto jsou pro tvorbu aplikací v ASP.NET nejvhodnější. S technologií ASP.NET lze vytvářet webové aplikace, které jsou rychlejší, protože programované části jsou předkompilované ve formě dynamických knihoven a na rozdíl od skriptovacích jazyků se nemusí stránky vždy znovu parsovat. V ASP.NET je také možné oddělit prezentační vrstvu (X/HTML) stránek od programové části. Toto oddělení pomůže zpřehlednit stránky a následně zjednoduší jejich úpravu nebo odstraňování chyb. ASP.NET lze nasadit na server Microsoft IIS, částečně ho lze provozovat i na serveru Apache s rozšířením Mono.

2.1.7 JSP (JavaServer Pages)

JavaServer Pages je technologie určená pro tvorbu dynamických webových aplikací. JSP navrhla a vytvořila společnost Sun Microsystems jako součást platformy Java, resp. dílčí platformy Java EE 5 (Java Enterprise Edition, dříve J2EE), která je

určená pro vývoj rozsáhlých aplikací. Platforma Java zastřešuje použití programovacího jazyka Java v oblastech vývoje a provozu různých aplikací. Tím pádem se webové aplikace JSP programují v Javě. Programování je podobné jako u skriptovacích jazyků typu PHP, programové části se vepisují přímo do dokumentů společně s (X)HTML. Ale na rozdíl od skriptovacích jazyků jsou při prvním použití dokumenty JSP automaticky převedeny na zdrojový kód servletu, který následně vyřizuje požadavky směřující na dokumenty JSP. Díky této kompilaci jsou stránky generovány mnohem rychleji ve srovnání se skriptovacími jazyky. Servlety se také starají o to, aby po modifikaci JSP stránky došlo k vytvoření nového zdrojového kódu servletu. Díky tomu se nemusí programátor starat o kompilaci, čímž je mu ulehčen vývoj. Servlety jsou v podstatě programové komponenty vykonávané na straně serveru, které reagují na požadavky klientů například generováním webových stránek. Protože JSP je vystavěno na technologii Java, je platformně i serverově nezávislé. Jedinou podmínkou je podpora JSP ze strany serveru.

2.1.8 Webové servery

Technologie, které jsem popsal výše v textu, potřebují ke svému běhu webserver. Webový server je program nebo služba běžící v popředí nebo pozadí operačního systému. Tento program vyřizuje HTTP požadavky klientů odesláním webové stránky. Mezi nejpoužívanější webservery patří Apache a Microsoft IIS.

Apache HTTP server (aktuálně 2.2.8) je multiplatformní open source webový server. Tento server je nabízen zdarma a pro svoji platformní nezávislost je nerozšířenějším webserverem internetu. Je pro něj vyvíjeno množství modulů, které rozšiřují jeho schopnosti. Například již jmenované Mono, které umožní na serveru využít technologii ASP.NET nebo moduly Tomcat či Geronimo, po jejichž připojení lze použít JSP.

Druhým nejrozšířenějším serverem je Microsoft Internet Information Services (IIS). Poslední verze nese označení 7. Tento server nabízí plnou podporu ASP.NET, ovšem je limitován možností nasadit jej jen na počítače s operačním systémem Windows.

2.1.9 Databáze, databázové servery

Databáze je úložiště dat a soubor nástrojů, které zajišťují ukládání a manipulaci s těmito daty. Virtuální obchod pro svůj chod potřebuje uchovávat určitá data, kte-

rými jsou např. informace o uživateli, nabízených produktech nebo objednávkách. Jako skladiště dat nejčastěji využívají právě databáze. Pro přístup k datům v databázi se využívá databázového serveru. Databázový server je program nabízející prostředky pro práci s daty v databázi i pro přístup k nim.

Některé nejpoužívanější databáze pro webové aplikace:

- MySQL 5.0.51a multiplatformní systém šířený pod GPL³ i komerční licenci, snadno implementovatelný, výkonný, velmi rozšířený, nejčastěji používaný v kombinaci s Apache a PHP
- PostgreSQL 8.3.1 multiplatformní databázový open source systém nabízený pod BSD⁴ licenci, podpora standardu SQL 92/99
- Oracle Database 11g multiplatformní systém, podpora standardu SQL 92, několik verzí – placené i zdarma
- Microsoft SQL Server 2005 pouze pro platformu Windows, možnost získat placené i neplacenou verzi

³ General Public Licence – licence pro svobodný software. <http://www.gnu.org/licenses/gpl.html>

⁴ Berkeley Software Distribution – licence pro svobodný software.
<http://www.xfree86.org/3.3.6/COPYRIGHT2.html#6>

3 E-SHOPY NA TRHU

Dnes existuje několik způsobů, jakými je možné získat a používat aplikaci e-shop. V této části práce tyto způsoby zmíním a následně popíši několik komerčních aplikací e-shopů nabízených na českém trhu a také některé, už nejen české, dostupné zdarma. Zaměřit bych se chtěl hlavně na technologie využívané e-shopy, možnost tvorby šablon, možnost propojení obchodu k účetním nebo jiným firemním aplikacím.

3.1 ŘEŠENÍ NA MÍRU

Jednou z možných alternativ je nechat si vytvořit e-shop přesně podle požadavků podnikatele od firmy, která se zabývá vývojem webových aplikací. Takovéto řešení nabízí zákazníkovi možnost rozhodovat o tom, co všechno by měl e-shop umět, rozhodovat o implementaci všech prvků, kterými by se mohl chtít odlišit od konkurence. Těmito prvky může být originální design e-shopu oproti šablonovitému jiných řešení, způsob ovládání e-shopu, napojení na specifické informační systémy, které používá zákazník a jiné zákaznické požadavky na e-shop. Všechny výhody takového řešení jsou vykoupeny vysokou cenou, kvalitní na míru postavený e-shop s technickou podporou vyjde zákazníka na několik desítek tisíc korun. A navíc zákazník nebude mít e-shop k dispozici okamžitě, ale až ve chvíli kdy bude aplikace vytvořena. Tvorba kvalitního e-shopu zabere z pravidla 1 až 2 měsíce práce. Dalším problémem, který nastává je hosting e-shopu, toto může zařídit firma, která e-shop dodala, nebo si zákazník zřídí vlastní hosting. Hosting pro takovouto aplikaci je samozřejmě placenou službou, řádově stovky až tisíce korun měsíčně. Teprve po nasazení e-shopu na server bude moci zákazník e-shop začít provozovat.

3.2 KRABICOVÉ ŘEŠENÍ

V tomto případě se jedná o již vytvořená řešení, která uspokojí většinu zákazníků. Tyto prodejny nabízejí vysokou modulárnost, vysokou úroveň customizace, také velké množství modulů a funkcí, aby uspokojily co nejvyšší množství potenciálních zákazníků. Ve většině případů také poskytují možnost úpravy šablon vzhledu e-

shopu, aby bylo možné ještě více přizpůsobit e-shop představám zákazníka. Většina těchto „krabicových“ řešení se nekupuje jako celek, nýbrž se pronajímá licence na jejich užívání, zpravidla se tato licence pronajímá na měsíc. Ceny měsíčního pronájmu se pohybují od 150 Kč do 4000 Kč včetně hostingu aplikace. Výhodou takového řešení jsou nižší zřizovací náklady, téměř okamžité nasazení do provozu, kvalitní podpora, pokračující vývoj aplikace a to, že v měsíčním pronájmu jsou započítány i náklady na hosting, o který se již zákazník nemusí starat. Nevýhodou může být určitá šablonovitost e-shopů stejného řešení, nemožnost použití jiných funkcí než těch nabízených e-shopem.

3.3 VOLNĚ DOSTUPNÁ ŘEŠENÍ

Řešení, která je možné získat a šířit volně (ovšem nikoliv podmíněně) a často také s otevřeným zdrojovým kódem, nabízejí možnost jak získat e-shop bez vynaložení finančních prostředků a téměř okamžitě. Toto jsou hlavní z mála výhod, které taková řešení nabízí. Mezi nevýhody patří jednotné grafické vzezření obchodů, které bez zásahu do zdrojového kódu jde jen velmi těžko změnit. Drtivá většina těchto aplikací nemá nativní podporu českého jazyka a ani neumožňuje změnu textů prodejny, jedinou možností se znovu stává zásah do zdrojového kódu aplikace. Samozřejmě lze na Internetu dohledat šablony, které změní vzhled i jazyk těchto e-shopů, ale většinou jsou tyto šablony zpoplatněné. Navíc jen těžko se napojí takovýto e-shop na jiné informační systémy, které by chtěl podnikatel používat. Podnikatel také musí řešit otázku hostingu a zprovoznění aplikace na serveru.

Tyto aplikace lze získat pod několika druhy licencí, nejčastěji licencemi GPL, LGPL. GPL (General Public Licence) je licence pro svobodný software. Takový software lze bezplatně používat, upravovat i šířit, ale pouze pod podmínkou, že bude šířen zdarma (případně za cenu vynaloženou na distribuční kanál). Zdrojové kódy musí být s touto licencí k dispozici bezplatně. Pokud je takový software dále šířen (i upravený), musí být poskytován pod stejnou licencí, tedy GPL. LGPL (Lesser GPL) je obdobou licence GPL s tím rozdílem, že software šířený pod touto licencí lze zakomponovat i do jiného software, který není šířen pod licencí GPL.

3.4 KONKRÉTNÍ KOMERČNÍ ŘEŠENÍ

3.4.1 Zoner In-Shop 4

Vysoce modulární řešení společnosti Zoner. S pomocí administračního programu Zoner In-Shop Manager lze nastavovat parametry obchodu, jako je např. výběr šablon, zobrazované prvky, texty používané v prodejně a jiné. E-shop také nabízí možnost úpravy šablon, díky kterým lze vytvořit originální obchod nebo si jej přizpůsobit svým požadavkům. Produkt je možné provozovat jen na serverech dodaných firmou Zoner. Nenáročné řešení určené široké škále zákazníků.

3.4.2 oXySHOP

Multiplatformní aplikace vyvinutá firmou oXy Online v prostředí Java 1.6. Jedná se o aplikaci, která je vytvářena zákazníkovi „na míru“, tzn., že je provedena projektová analýza a návrh řešení, implementace, včetně grafického designu a napojení e-shopu na ostatní informační systémy zákazníka. Vytvořená aplikace je dodávána včetně hostingu serveru. Toto řešení je určené hlavně pro střední a velké virtuální obchody.

3.4.3 KarsaV2

KarsaV2 je aplikace e-shop společnosti ExpertShop, nabízená jako „krabicové řešení“. Tento e-shop je možné provozovat jak na serverech společnosti, tak i na vlastních. Vzhled vlastního obchodu lze upravovat s pomocí manageru vzhledu, který je vytvořen jako webová aplikace dostupná na serveru, kde je obchod umístěn. V manageru vzhledu lze zvolit různá barevná schémata nebo předpřipravené šablony. Samotná aplikace také nabízí API⁵, pomocí kterého lze vytvořit zcela nový obchod, případně upravit některé jeho části. E-shop je určený pro začínající i již zaběhnuté obchodníky.

3.5 KONKRÉTNÍ NEKOMERČNÍ ŘEŠENÍ

3.5.1 osCommerce

OsCommerce je open source systém pro tvorbu a provozování e-shopu. Ve verzi 2.2 je systém vystavěn na kombinaci databáze MySQL (3.23 nebo novější), skriptovacího jazyka PHP (verze 4.2.1 a vyšší) a serveru Apache nebo Microsoft IIS. E-shop je

⁵ Application Programming Interface – rozhraní pro programování aplikace, soubor procedur a funkcí knihovny nebo jádra, které může programátor používat

šířen pod GPL licencí a je určen pro začínající nebo neprofesionální obchodníky. Obchod je ovládán přes webové rozhraní, změnu vzhledu nativně nepodporuje. Přesto je možné design změnit. Lze využít komerční šablony určené pro osCommerce, šablony vyrobené na zakázku nebo jej upravit svépomocí. Češtinu základní aplikace také nepodporuje, ale na Internetu jsou volně k dostání moduly, které ji počestí.

3.5.2 Quick.Cart

Aplikace Quick.Cart je vystavěná ve skriptovacím jazyku PHP. Zajímavá je hlavně tím že nevyužívá relační databáze, nýbrž tzv. flat files databázi. Tato databáze je součástí aplikace a pro ukládání jednotlivých tabulek používá soubory, do kterých data ukládá jako čistý text. Využití této technologie je nejvíce limitujícím prvkem produktu. Nelze totiž data z tabulek řadit, což toto řešení určuje pro použití s opravdu malými obchody nebo jen pro vyzkoušení. Jinak Quick.Cart obsahuje všechny důležité prvky, které by měl elektronický obchod mít, případně je možné další funkčnost dodat pomocí plug-inů. Samozřejmostí je možnost vlastních úprav šablon vzhledu nebo nákup již předpřipravených šablon.

3.6 ZHODNOCENÍ DOSTUPNÝCH ŘEŠENÍ NA TRHU

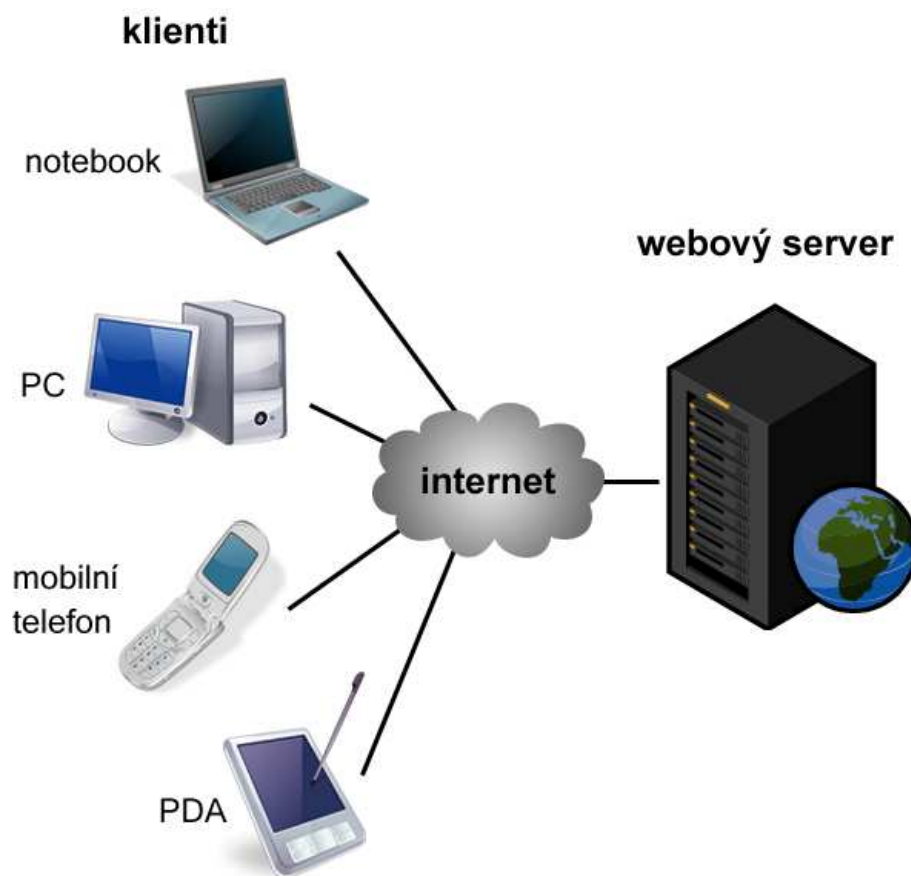
Všechny výše zmíněná řešení nabízejí podnikateli možnost, vytvořit virtuální obchod. Ovšem otázkou zůstává, jaké úrovně bude takový obchod dosahovat a kolik bude jeho pořízení stát peněz. Pro začínající prodejce bych doporučil řešení osCommerce, Quick.Cart, ale lépe Karsa V2 nebo Zoner In-Shop 4, zejména z důvodu nižších zřizovacích nákladů. Pro obchodníky, kteří to myslí s podnikáním na Internetu opravdu vážně je jednoznačně vhodný Zoner In-Shop 4. A pro opravdu velké, již fungující obchody, jejichž majitelé by chtěli přejít na jinou, výkonnější aplikaci virtuálního obchodu, lze doporučit na míru jejich požadavkům postavený oXySHOP, za který ale obchodník zaplatí relativně vysokou částku. Některé další informace týkající se zde zmíněných řešení jsou k nalezení v tabulce číslo 1. V tabulce jsou uvedeny technologie na nichž jsou aplikace vystavěny, podporované platformy operačního systému, na kterém běží server s aplikací, zda e-shopy mají implementovanou podporu šablon vzhledu, případně jiné možnosti úpravy designu a jak mají vytvořené administrační rozhraní.

Tabulka 1: Porovnání vybraných řešení pro e-shop

obchod	technologie	platforma serveru	šablony	administrace
In-Shop 4	ASP.NET 2.0, Microsoft IIS 6.0, MS SQL Server 2005	Windows	ano, možnost vytvoření vlast- ních	off-line deskto- pová aplikace
oXySHOP	JSP, Apache + Tomcat, Sybase 12.5	multiplatformní, primárně Linux	ne, dodávány včetně designu na klíč	off-line deskto- pová aplikace
KarsaV2	ASP.NET 1.1, Microsoft IIS 6.0, MS SQL Server 2000	Windows	ano, možnost úpravy nebo vytvoření vlastní	on-line přes web i off-line deskto- pový klient
osCommerce	PHP 4, Apache nebo Microsoft IIS, MySQL 3.23	multiplatformní	ne, komerční rozšíření, vlastní úprava designu	on-line přes webový formulář
Quick.Cart	PHP 4, server s podporou PHP	multiplatformní	ano, možnost vytvoření vlast- ních, komerční	on-line přes webový formulář

4 MOBILNÍ ZAŘÍZENÍ A WEBOVÉ STRÁNKY

Velké množství mobilních zařízení dnes disponuje technologiemi umožňujícími připojení do sítě Internet. Pokud jsou tato zařízení vybavena i prohlížečem webových stránek, je možné z nich přistupovat na webové servery a stránky, jak ukazuje obrázek 1. To, zda se webovou stránku podaří vykreslit korektně, závisí na několika faktorech. V první řadě úspěch závisí na samotném zařízení a na software, který je k prohlížení webových stránek na zařízení k dispozici. Důležitým faktorem je, jaké webové standardy tento prohlížeč podporuje nebo s kterými dokáže alespoň částečně pracovat. Neméně důležité je i to, jakým způsobem jsou napsány webové stránky, které chceme prohlížet. Zda se jedná o stránky přímo určené mobilním zařízením, nebo stránky primárně určené standardním zařízením, ale napsané s ohledem i na mobilní, a nebo stránky vhodné čistě pro klasické zobrazení ve webovém prohlížeči PC. V této části práce se zaměřím právě na standardy vhodné pro mobilní stránky, omezení mobilních zařízení a řešení těchto omezení.



Obrázek 1: Přístup klientů z různých zařízení na webový server

Ještě než přejdu ke konkrétnímu popisu bylo by vhodné si definovat pojem mobilní zařízení. Mobilním zařízením v této práci jsou myšleny v první řadě kapesní počítače typu PDA nebo handheld a druhotně také mobilní telefony. Naopak notebooky a laptopy jsou považovány za standardní zařízení.

4.1 STANDARDY MOBILNÍHO WEBU

Moderní mobilní webové dokumenty jsou postaveny na základech několika standardů. A tyto standardy jsou odvozeny od základních webových, kterými jsou HTML, XHTML a CSS.

4.1.1 XHTML Basic 1.0, XHTML Mobile Profile 1.2

XHTML Basic je podmnožina standardního XHTML upravená pro použití s mobilními zařízeními. Obsahuje mnohem méně tagů a atributů. XHTML MP vychází z XHTML Basic a rozšiřuje jej o některé tagy pro formuláře, formátování textu, přidává podporu CSS stylování, dále také podporu skriptování a atributy událostí. A v poslední verzi XHTML MP 1.2 přidává podporu tagů button a legend a atribut inputmode pro tagy input a textarea.

4.1.2 CSS Mobile Profile 2.0

Vychází ze standardu CSS 2.1 a odstraňuje ty prvky CSS 2.1, které není možné, nebo velmi těžce možné využít na mobilní platformě. Společně s XHTML MP umožňuje vytvořit webové dokumenty určené pro mobilní zařízení, které mohou obsahovat grafický design oproti standardní čistě textové formě nestylovaných dokumentů.

4.2 OMEZENÍ MOBILNÍCH ZAŘÍZENÍ VE ZOBRAZENÍ WEBU

I když jsou současná mobilní zařízení vysoce vyspělá, stále mají některá omezení, která jim brání k plnohodnotnému prohlížení standardního webu. Tím největším omezením je rozlišení jejich displeje. U kapesních počítačů je dnešním standardem rozlišení 480 x 640 pixelů a u mobilních telefonů 240 x 320. Takto nízké rozlišení nestačí ke korektnímu zobrazení složitějších webových stránek, jako jsou např. stránky internetových obchodů nebo jiné různé stránky s vícesloupcovým layoutem. Problém může také nastat u webových aplikací, jejichž odkazy se otvírají v nových oknech. S tímto si nemusí prohlížeče mobilních zařízení poradit nebo to může uživa-

teli ztížit navigaci po stránkách. Dalším omezením je velikost vnitřní paměti u mobilních telefonů, které jí nemusí mít dostatek pro načtení celé stránky. Stránka se pak načte neúplná a mohla by chybět část důležitého obsahu. Další v řadě je zde problém s různou úrovní podpory standardů pro (mobilní) web, kdy různé prohlížeče jinak reprodukuje stejné stránky nebo neznají některé jejich prvky, z čehož plyne nepoužitelnost některých dokumentů v určitých prohlížečích. Řešením je instalace jiného prohlížeče na zařízení (např. Opera Mini), což současné mobilní telefony i handheldy umožňují.

4.3 ZPŘÍSTUPNĚNÍ WEBOVÝCH STRÁNEK PRO MOBILNÍ ZAŘÍZENÍ

Budeme-li chtít standardní webovou prezentaci zpřístupnit pro mobilním zařízení, máme několik způsobů jak toho docílit. První možností je vytvořit CSS styl, jež upraví layout stránky tak, aby byla vhodná pro mobilní zařízení a nastavit mu příznak pro použití s takovým zařízením. Dalším způsobem je vytvořit šablonu stránek, nebo stránky přímo určené pro mobilní zařízení, které se budou načítat při požadavku z těchto zařízení.

4.3.1 Zpřístupnění vytvořením CSS stylů pro různá zařízení

Principem této úpravy webové stránky je ve vytvoření 2 stylopisů kaskádových stylů, které jsou v hlavičce webové stránky k ní přilinkované s různým parametrem *media*. První stylopis určený pro stylování stránky na standardním zařízení je přilinkován s parametrem *media*="screen" a měl by se načítat, když je stránka otevřena v normálním webovém prohlížeči. Druhý styl napsaný pro stylování stránky v mobilním prohlížeči se přilinkuje s parametrem *media*="handheld". Takto vytvořená stránka by se měla v prohlížečích různých zařízení zobrazovat jinak nastylovaná. Hlavička stránky se stylopisy pro různá média může vypadat takto:

```
<head>
  <title>Univerzální stránka</title>
  <link rel="stylesheet" type="text/css" media="screen"
    href="normalni.css">
  <link rel="stylesheet" type="text/css" media="handheld"
    href="mobilni.css" />
</head>
```

Definovat pro jaké zařízení je styl určen lze i u tagu `<style>` s atributem `media`, nebo pomocí direktiv `@import` a `@media`, za které uvedeme typ zařízení. Naneštěstí většina mobilních prohlížečů tyto definice CSS stylů ignorují a používají styl pro ně nevhodný nebo dokonce oba dva najednou.

Své předchozí tvrzení jsem ověřil otestováním několika prohlížečů pro mobilní zařízení na testovací webové stránce, jež se nachází na url adrese <http://www.htmldog.com/test/handheld.html>. Na této stránce jsou různě definované CSS styly, podle všech výše zmíněných definic, z toho jednou pro standardní zařízení `screen` a jednou pro zařízení `handheld`. Tato testovací stránka vykreslená v prohlížeči je rozdělená na dvě ohraničené části, které leží nad sebou. V horním obdélníku je část textu nastylována styly pro zařízení `screen` a ve spodním obdélníku pro zařízení `handheld`. Pokud prohlížeč styly aplikuje, objeví se na stránce místo stylovaného textu zelený obdélníček. V případě kdy styly nepoužije, vypíše se červený text `not`. Pro testování jsem použil 4 mobilní prohlížeče, konkrétně MS Internet Explorer Mobile 6.12, Opera Mobile 8.56, Mozilla Minimo 0.2 a NetFront Browser 3.5. Z těchto prohlížečů testem prošel pouze jediný – Opera Mobile, ostatní stránku nezobrazili správně. Výsledky testu jsou vypsány v tabulce 2, která obsahuje pro dokreslení i výsledky desktopových prohlížečů Mozilla Firefox 2.0.0.14 a MS Internet Explorer 7.0.6001.18000. Screenshoty obrazovek všech prohlížečů tak, jak vykreslili testovací stránku jsou v příloze A.

Tabulka 2: Výsledky testu prohlížečů ve vykreslování stránky

Prohlížeč	Styl pro media="handheld"	Styl pro media="screen"	Uspěl v testu
MS Internet Explorer Mobile 6.12	vykreslil vše (4/4)	vykreslil vše (4/4)	NE
Opera Mobile 8.65	vykreslil vše (4/4)	nevykreslil nic (0/4)	ANO
Mozilla Minimo 0.2	nevykreslil nic (0/4)	vykreslil vše (4/4)	NE
NetFront Browser 3.5	nevykreslil nic (0/4)	vykreslil vše (4/4)	NE
Mozilla Firefox 2.0.0.14	nevykreslil nic (0/4)	vykreslil vše (4/4)	ANO
MS Internet Explorer 7.0.6001.18000	nevykreslil nic (0/4)	vykreslil většinu (3/4)	ČÁSTEČNĚ

4.3.2 Zpřístupnění vytvořením (webových) šablon pro různá zařízení

Druhou možností jak přizpůsobit webovou prezentaci pro mobilní zařízení je vytvořit pro tato zařízení vlastní šablonu, nebo pro ně uzpůsobené stránky, na které bude po připojení k webovému serveru uživatel přesměrován.

Toto řešení předpokládá, že minimálně do úvodní webové stránky bude implementován skript nebo rozhodovací blok, který vyhodnotí jaké zařízení (prohlížeč) vyslalo požadavek na webový dokument a podle výsledku jej buď přesměruje na mobilní verzi webu, či použije vhodné (mobilní) šablony, nebo jej nechá pokračovat na standardní web.

Z výše uvedeného vyplývá nutnost použití funkce, která rozhodne o jaké zařízení se jedná. Takováto funkce je k dispozici pouze v prostředí ASP.NET a její název je *IsMobile*, nicméně některé prohlížeče na PDA zařízeních identifikuje jako prohlížeče normálních počítačů. Proto je nutné takovouto funkci napsat, nebo použít funkci, kterou již někdo vytvořil před námi. Jedinou možností, jak ve funkci rozhodnout o jaké zařízení se jedná, je jeho identifikace pomocí údajů, které prohlížeč zařízení posílá v HTTP požadavku na webový dokument. Funkce bude také nutně potřebovat soubor identifikačních hodnot prohlížečů (zařízení), pro jejich správné rozpoznání. Tento soubor musí být dostatečně rozsáhlý, aby funkce byla schopna identifikovat co možná největší počet zařízení. Funkce porovná data z prohlížeče klienta s tímto souborem dat a pokud dojde k nějaké shodě s údaji ze souboru, bude výsledek funkce pravda a zařízení bude identifikováno jako mobilní. Z výsledku funkce se rozhodne jakou šablonu či stránku použít.

Příklad funkce pro identifikaci mobilního zařízení napsané v jazyce PHP:

```
function IsMobileDevice()  
{  
    $ret=ERegI(QuoteMeta("UP.LINK|MIDP|UP.BROWSER|NOKIA|MOT|SEC-".  
        "|WAP|ERICSSON|SAMSUNG|SIE-|PHONE|PANASONIC".  
        "|MITSU|LG|PORTALMMM|BLACKBERRY|SYMBIAN|PHILIPS".  
        "|SENDO|KLONDIKE|SAGEM|MOBILE|ALCATEL|SONY".  
        "|WINDOWS CE|PDA"), $_SERVER["HTTP_USER_AGENT"]);  
    return $ret;  
};
```

Ošetření přístupu na webové stránky tímto způsobem oproti definici media v kaskádových stylech, je ve výsledku vhodnější řešení. Protože pokud použijeme ve funkci dostatečně rozsáhlý a aktuální soubor identifikačních údajů prohlížečů a zařízení, jsme schopni maximálně minimalizovat počet zobrazení webových stránek, které nejsou pro zařízení určeny.

4.4 WEBDESIGN VHODNÝ PRO MOBILNÍ STRÁNKY

Jak bylo v předchozím oddíle napsáno, jednou z možností zpřístupnění webu je vytvoření dalších stránek nebo šablon vhodných přímo pro mobilní zařízení. V tomto oddíle naznačím, jak by měly stránky určené pro mobilní zařízení vypadat, jakých doporučení se držet a kterých nešvarů se vyvarovat.

4.4.1 Layout stránek

Layout nebo-li rozvržení stránky by mělo být co nejjednodušší a to hlavně z důvodu nízkého rozlišení displejů mobilních zařízení. Na displejích s nízkým rozlišením nelze dost dobře vytvořit vícesloupcové rozložení prvků (např. navigace a obsah ve vlastním sloupci vedle sebe), tedy stránky určené pro tato zařízení by měly mít pouze jeden sloupec tzv. dlouhou nudli. Do sloupce se musí vejít celá stránka. Navíc by měla být omezena nutnost posouvání stránky v horizontálním směru na minimum, nejlépe úplně odstraněna a stránka by měla být rolovatelná jen ve svislém směru. To hlavně z důvodu, že pokud je uživatel nucen posouvat stránku horizontálně, může se na stránce lehce ztratit. A také proto, že čtení textu, který si musí čtenář posunovat do stran, je obtížné a nepříjemné.

Je vhodné dodržovat správnou sémantiku dokumentu. A to proto, aby byl dokument správně strukturován i v případě, kdy bude zobrazen v zařízení, které nepodporuje kaskádové styly nebo je jejich použití zakázané. Dodržováním sémantiky jazyka je myšleno správné používání úrovní nadpisů, řazení textů do odstavců namísto neutrálních kontejnerů, které nic neříkají o logické struktuře obsahu. Pokud bude dokument sémanticky správný, neměl by být problém jej správně zobrazit na jakémkoliv zařízení.

4.4.2 Rozmístění prvků na stránce

Stránka by měla na svém začátku obsahovat hlavičku, která identifikuje ji i web, aby se mohl uživatel rychle zorientovat, kde se vlastně nachází. Za hlavičkou by měl následovat hlavní blok s informacemi, které jsou nejdůležitějším obsahem aktuální stránky. To z důvodu, že čtenář může rychle pochopit, co je obsahem stránky a zda tento obsah odpovídá tomu, co hledal, a to bez dlouhého zdržování rolováním stránky přes nedůležitý obsah. Proto by před vlastním obsahem hlavního bloku mělo být minimum ostatních rušivých prvků jako jsou např. reklamy nebo s aktuální stránkou nesouvisející texty. Za hlavním blokem by již měla být umístěna navigace, na kterou

by mohla na začátku stránky odkazovat kotva pro její rychlejší dosažení. Za sekci navigace je možné vložit ostatní prvky stránky, jakými mohou být např. patička. Pokud web obsahuje sekci hledání, lze ji umístit na začátek stránky pod hlavičku, nebo do hlavičky. Stejně tak ji lze umístit před navigaci, nebo až za ní. Umístění sekce hledání je závislé od toho, jak často bude tato funkce na webu využívána, čím častěji o to by měla být výš ve struktuře stránky, aby byla pro uživatele rychleji dosažitelná. Je dobré jednotlivé sekce stránky od sebe oddělit, proto aby bylo možné je lehce rozlišit. Pro takovéto oddělení lze využít elementů vodorovná čára nebo nadpis, který může být s použitím kaskádových stylů barevně odlišen od ostatního textu.

Pro rychlejší dostupnost některých prvků stránky je také vhodné na začátek stránky vložit několik kotev, které na ně budou odkazovat. Těchto kotev by nemělo být mnoho, aby neodsouvaly hlavní informace stránky pod vykreslenou část stránky a nenutily tím zbytečně uživatele k rolování. Nejlépe by se kotvy měly vejít na jeden až dva řádky. Pokud stránky obsahují vstupní textová pole, je vhodné jejich popis a vlastní textové pole umístit na samostatné řádky pod sebe. To z důvodu nedostatečné šířky displeje mobilních zařízení, na kterém by se nemuselo vykreslit celé pole nebo by bylo schované za okrajem stránky. Stejným způsobem je vhodné uspořádat tlačítka, pokud jich je ve stránce na jednom místě více.

Prvky stránky by se měly zarovnávat na levý okraj stránky, rozhodně ne doprostřed nebo vpravo. Prvky umístěné na jiný než levý okraj, by se nemusely na nedostatečně širokém displeji zobrazit, a protože by bylo pro jejich dosažení nutné posouvat stránku doprava, mohlo by se stát, že je uživatel přehlédne. U obrázků se doporučuje uváďet jejich velikost, aby si prohlížeč ve vykreslené části stránky vytvořil prostor pro takové obrázky, před tím než je načte. A po jejich načtení již nemusel měnit délku stránky, umístění ostatních prvků nebo posouvat text, který by v tu chvíli už uživatel mohl číst.

4.4.3 Čeho se vyvarovat

Není vhodné snažit se skládat prvky stránky vedle sebe nebo vytvářet více sloupcový layout, protože většina mobilních zařízení má natolik malý displej, že se je stejně nepodaří vedle sebe vykreslit. Nevhodné je také použití rámu (které jsou zapovězeny i v normálním webu), protože jim některé mobilní prohlížeče nemusí rozumět a jsou celkově problematické. Pro mobilní web jsou rozhodně nevhodné tabulky, ať jako

nástroj pro vytvoření layoutu nebo jako normální tabulku. V první řadě mnoho prohlížečů neumí tabulky správně vykreslit, druhotně pokud se nevejdou na displej a nutí uživatele k vodorovnému rolování, čímž mu znesnadňují prohlížení stránky a ničí její celkové rozvržení. Další věci, které bychom se měli vyvarovat, jsou příliš velké obrázky (rozměrově i datově) a to z důvodu jejich dlouhého načítání a také proto, že mnoho uživatelů za připojení platí podle množství přenesených dat. Rovněž mnoho různých obrázků na stránce není vhodné, protože brzdí její načítání, neboť prohlížeč musí pro každý z nich udělat jeden HTTP požadavek. Nedoporučuje se používání absolutních délkových jednotek pro formátování obsahu stránky, ale vhodnější jsou procentuální nebo relativní jednotky jako jsou *em*, *ex*, *small*, *medium*, *bold* a jiné. Také není vhodné do stránek implementovat vyskakující okna, která se nemusí zobrazit vůbec, protože některé prohlížeče neumí zobrazit více webových stránek najednou, nebo se zobrazí místo stránky původní. Ani automatické obnovování stránek nelze doporučit, alespoň pokud na to uživatel není upozorněn a nemá možnost refresh zakázat nebo si vybrat jinou stránku, která se automaticky znovu načítat nebude.

4.4.4 Obsah stránek

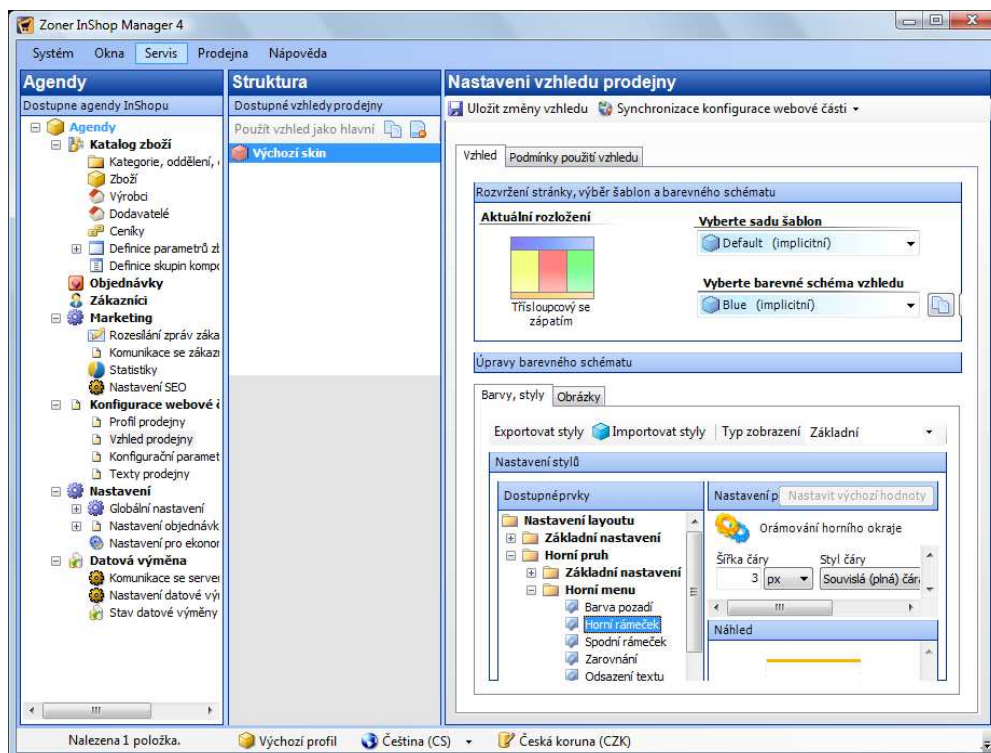
Nejvhodnější obsah pro mobilní webové stránky je text, který může být formátován pomocí kaskádových stylů a také může být. V dnešní době již nejsou zapovězeny ani obrázky, ale pouze v rozumné míře, jak bylo popsáno výše. Přestože první mobilní telefony již obsahují prohlížeče s podporou technologie Flash nebo do nich přehrávač Flash Lite doinstalovat, Flash zatím není pro implementaci do mobilních webových stránek vhodný. Stejně tak video a audio se přímo nehodí pro zakomponování do mobilního webu, hlavně z důvodu náročnosti na přenosovou rychlost připojení k Internetu, kterým disponuje zařízení a z důvodu omezené velikosti paměti, kterou zařízení disponují.

5 ŠABLONY ZONER IN-SHOP 4

Mým úkolem v této bakalářské práci bylo vytvoření univerzálních šablon pro aplikaci In-Shop 4 společnosti Zoner. Firma Zoner se zabývá vývojem a distribucí software, dále také poskytováním služeb internetu a dokonce i vydává odbornou počítačovou literaturu ve vlastním vydavatelství. Společnost sídlí v Brně a má své zastoupení na Slovensku, v Maďarsku, Japonsku a USA a zaměstnává více než 70 zaměstnanců. Šablony, které jsem tvořil, by měli sloužit pro zobrazení stránek e-shopu v mobilních zařízeních, pro která jsou uzpůsobeny. Jejich účelem je upravit vzhled e-shopu tak, aby rozložení a úprava stránek byla vhodná pro vykreslení v mobilních prohlížečích, dále ulehčit zákazníkovi orientaci v e-shopu a zrychlit načítání jeho stránek. Tyto šablony by měly sloužit zákazníkům e-shopů postavených na platformě Zoner In-Shop 4, kteří na jejich webové stránky budou přistupovat z mobilních zařízení, jakými jsou kapesní počítače nebo mobilní telefony.

Produkt Zoner In-Shop 4 nabízí každému provozovateli e-shopu možnost přizpůsobit si jeho design tak, aby maximálně vyhovoval jeho požadavkům. Správný design je pro e-shop, který chce být úspěšný, velice důležitý. Musí jej totiž odlišit od jiných e-shopů a zanechat v zákaznících pocit jistoty a důvěry. Také je nutné design vhodně přizpůsobit druhu zboží, které se v obchodě prodává a hlavně představám a očekáváním zákazníků, na které se e-shop především orientuje. To znamená, že jinak by měl vypadat e-shop s náradím určený pro kutily a jinak obchod s kosmetikou zaměřený na ženy. Naopak pokud nebude design e-shopu na dostatečně vysoké úrovni nebo nebude vhodně korespondovat s nabízeným sortimentem, může v potenciálních zákaznících vyvolat nedůvěru nebo pochybnost o spolehlivosti obchodu.

Pro úpravu a vytvoření specifického designu nabízí In-Shop 4 dvě možnosti. První možnost je úprava vzhledu e-shopu s pomocí programu Zoner In-Shop Manager, který slouží k administraci prodejny a úpravám jejího vzhledu. Jak rozhraní pro úpravu vzhledu v aplikaci In-Shop Manager vypadá, je vidět na obrázku číslo 2. Druhou mnohem více sofistikovanější možností je upravit sadu šablon, které In-Shop používá při generování webových stránek e-shopu.



Obrázek 2: Zoner In-Shop Manager 4 - nastavení vzhledu prodejny

Standardní šablony aplikace In-Shop nabízejí 4 barevná schémata a 6 možných způsobů rozvržení stránky, tyto způsoby rozvržení jsou: třísloupcový se zápatím, třísloupcový, levý dvousloupkový, levý dvousloupkový se zápatím, pravý dvousloupkový, pravý dvousloupkový se zápatím a pro názornost jsou k vidění na obrázku číslo 3.



Obrázek 3: Možné způsoby rozvržení stránek e-shopu Zoner In-Shop 4

5.1 CO JSOU ŠABLONY V IN-SHOPU

Jako šablona je v In-Shopu označován soubor s příponou .ascx, který server používá při generování webové stránky e-shopu, kterou si vyžádal klient. Takovýto soubor obsahuje XHTML tagy a navíc ještě tagy In-Shop, které server při zpracování šablony nahrazuje odpovídajícími hodnotami, XHTML prvky stránek nebo vkládá další šablony. In-Shop tag potom může být nahrazen zdrojovým kódem například webového formuláře, hypertextového odkazu na jinou stránku e-shopu, obrázku nebo nadpisu sekce aj.. Zoner In-Shop 4 má takovýchto šablon přibližně 120.

5.2 PRINCIP ZPRACOVÁNÍ ŠABLON

Základní šablonou celé kolekce je soubor s názvem `Layout.ascx`. Tento soubor je první, který se načítá při požadavku na stránku e-shopu a jsou v něm uchována základní data o struktuře jeho webových stránek. Všechny ostatní šablony použité k zobrazení stránky, se vkládají do této šablony, nebo do šablon do ní vložené.

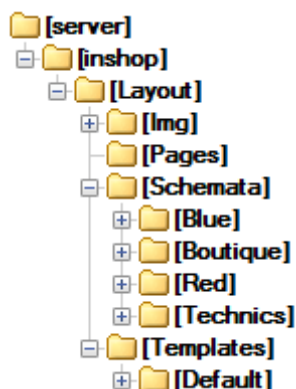
Šablony samy o sobě nemohou být zobrazeny, protože obsahují i tagy, kterým prohlížeče nerozumí. Pro vytvoření výstupu, který již budou prohlížeče schopné reprodukovat, je nutné šablony zpracovat. Šablony zpracovává skript `shop.aspx`, který v šablonách nahrazuje In-Shop tagy adekvátními daty vhodnými pro webový prohlížeč.

Princip zpracování šablon je tedy takový, že pokaždé když na webový server e-shopu přijde požadavek na stránku, tak server volá skript `shop.aspx` s parametry požadavku. Skript `shop.aspx` následně načte a zpracuje šablonu `layout.ascx` a ostatní šablony, které jsou určeny parametry klienta požadavku. Zpracovaný výstup ze šablon potom server pošle zpátky klientovi jako webovou stránku.

5.3 TVORBA ŠABLON

Spíše než vytváření vlastních šablon od prázdného stolu, je vhodnější tvořit vlastní šablony úpravou standardních. Tento postup má mnoho výhod. Asi tou největší výhodou je, že v době, kdy k šablonám není dokončená dokumentace, nám umožní se s šablonami seznámit, pochopit jak pracují a jak se používají tagy. Touto cestou jsem se při tvorbě šablon pro mobilní zařízení vydal i já.

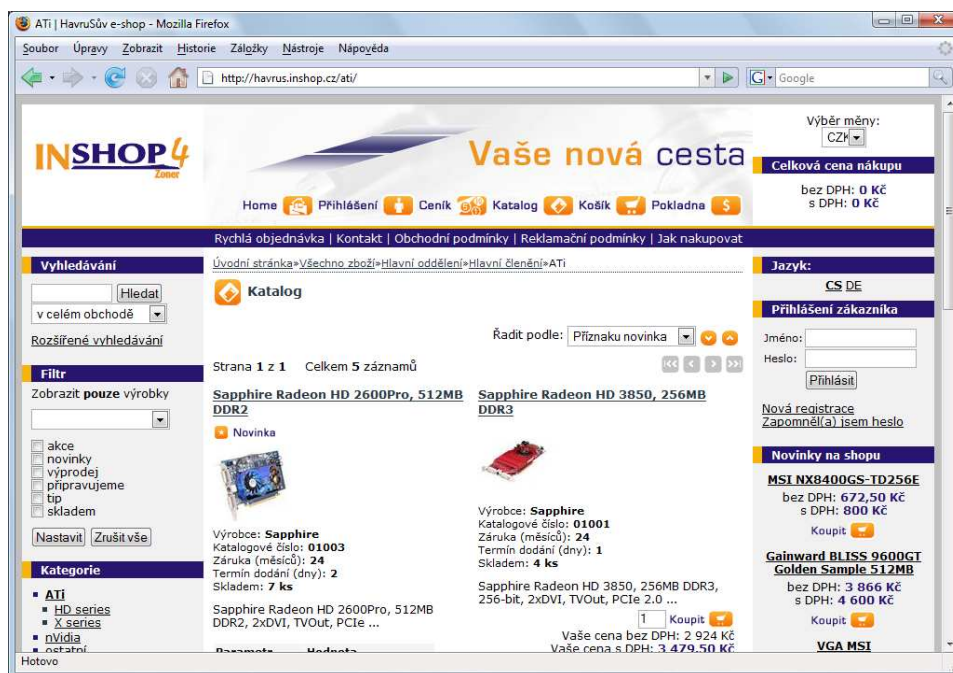
Důležité při tvorbě šablon je také vědět, kam vytvořené šablony nahrát, aby je bylo možné serverem zpracovat. Standardní šablony jsou uloženy na serveru v adresáři `inshop/layout/templates/default`. Pokud chceme ale používat naše šablony, musíme je nahrát na server do adresáře `inshop/layout/pages`. Když má server nějakou šablonu zpracovat nejdříve ji hledá v adresáři `inshop/layout/pages` a pokud ji nenalezne, použije standardní šablonu z adresáře `inshop/layout/templates/default`. Dobré je také vědět, že v adresáři `inshop/layout` se nacházejí i adresáře, ve kterých jsou uloženy obrázky, stylotypy a grafika pro různá barevná schémata. Struktura adresáře je znázorněna na obrázku 4.



Obrázek 4: Struktura adresáře inshop/layout

5.4 STANDARDNÍ ŠABLONY V MOBILNÍM PROHLÍŽEČI

Na obrázcích 5 a 6 je zobrazena stejná stránka e-shopu. Na prvním obrázku jak ji vykreslí standardní prohlížeč Mozilla Firefox a na druhém jak se vykreslí v mobilním prohlížeči MS Internet Explorer Mobile. Z obrázků je patrné, že prohlížeč v mobilním zařízení není schopen stránku správně vykreslit z důvodu malé šířky displeje. Stránka je nevhodně uspořádaná, příliš dlouhá a tabulky jsou špatně vykreslené. Orientace na stránce je velmi složitá, protože uživatel není schopen odhadnout, kde se co na stránce nachází. A protože hlavní blok stránky je vykreslen v jejím prostředku, bude mít uživatel problém zjistit, na jaké stránce se nalézá a co je jejím obsahem. Navíc stránka obsahuje až příliš velké množství obrázků, které nepříjemně zdržují její nahrání do prohlížeče.



Obrázek 5: Vykreslení standardní šablony v prohlížeči Mozilla Firefox



Obrázek 6: Vykreslení standardní šablony v prohlížeči MS Internet Explorer Mobile

Z předchozího textu je tedy zřejmé, že standardní šablony pro užití na mobilních zařízeních přímo vhodné nejsou. Ne že by nebylo možné v takovém e-shopu nakoupit nebo například zjistit informace o stavu svojí objednávky, ale bylo by to pro zákazníka zdlouhavé a nepříjemné. Většina potenciálních zákazníků by stejně po prvním načtení stránek do prohlížeče jejich mobilního zařízení odešla hledat někam jinam, neboť při pohledu na rozbitý, pro mobilní zařízení nevhodný layout obchodu, by se ani nechtěli pokoušet o nákup v něm.

6 ŠABLONY ZONER IN-SHOP 4 URČENÉ PRO MOBILNÍ ZAŘÍZENÍ

V současné době, kdy roste popularita handheldů a mobilní telefony s barevným displejem a integrovaným internetovým prohlížečem jsou standardem, by byla škoda odepřít jejich majitelům právo využívat možností Internetu naplno v těchto zařízeních tím, že naše webové stránky nebudou vhodné pro zobrazení v nich. Navíc je podle mého názoru v mobilních zařízeních budoucnost, protože se začínají objevovat ve více a více odvětvích lidské činnosti. Proto bychom se při tvorbě webových aplikací a prezentací měli více věnovat i uživatelům s mobilními zařízeními. V této části své bakalářské práci budu psát o šablonách Zoner In-Shop 4 vhodných pro zobrazení e-shopu na mobilních zařízeních, které jsem vytvořil, jakým způsobem jsem je navrhl, implementoval a jak vypadá konečný výsledek.

6.1 LAYOUT

Jak jsem již psal v kapitole o layoutu stránek pro mobilní weby, nejvhodnějším rozvržením stránky je jednosloupcový layout s prvky stránky, které jsou rozloženy po celé výšce stránky. Toto rozvržení jsem se rozhodl použít i pro šablony, které jsem tvořil.

Nejvýše na stránce e-shopu je umístěna hlava, následovaná hlavním blokem, pod ním se nachází sekce kategorií a vyhledávání výrobků, další v pořadí je blok navigace v e-shopu a poslední sekci je patička stránky. Pro lepší představu je celkové rozvržení stránky vyobrazeno na obrázku číslo 7 níže za textem.

6.1.1 Hlava

Prvním blokem stránky je hlava, ta obsahuje obrázek loga obchodu, formulář pro výběr měny, která se bude zobrazovat v eshopu, dále odkazy pro přepnutí jazykové mutace e-shopu a kotvu (odkaz) pro rychlé přeskočení obsahu stránky na blok navigace.

6.1.2 Hlavní blok

Hlava je následovaná hlavním blokem, ve kterém je zobrazován obsah aktuálních stránek e-shopu. Tzn. pokud zákazník otevře z bloku navigace např. katalog, ceník nebo obchodní podmínky obchodu, po nahrání nové stránky se v hlavním bloku zobrazí právě tyto prvky.

6.1.3 Blok kategorie a vyhledávání

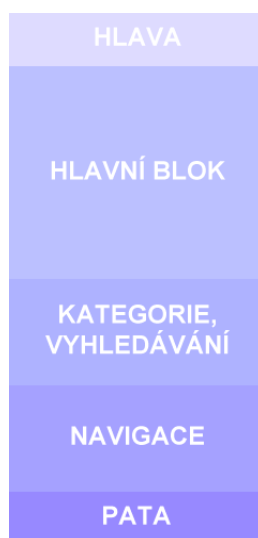
V tomto bloku jsou umístěny dvě sekce. První je sekce kategorií, ve které se zobrazuje strom kategorie zboží, členění zboží a další oddělení obchodu, podle toho jak jsou nastaveny v In-Shop Manageru. Druhou sekcí tohoto bloku je formulář vyhledávání zboží v e-shopu.

6.1.4 Blok navigace

Předposledním blokem v rozvržení stránky je navigace, na kterou z hlavy stránky odkazuje kotva pro její rychlejší dosažení. Odkazy z této sekce se po jejich použití budou zobrazovat v hlavním bloku stránky a jedná se o všechny odkazy pro navigaci v e-shopu, např: ceník, katalog, pokladna, košík, obchodní podmínky, atd.

6.1.5 Pata

Za navigací se nachází poslední sekce stránky, kterou je pata stránky. V této sekci je umístěna informace o aktualizaci e-shopu a copyright stránek.



Obrázek 7: Rozvržení stránky e-shopu upravené pro mobilní zařízení

6.1.6 Blok přihlášeného zákazníka

Tento blok v rozvržení stránky nebude viditelný do doby, než se registrovaný zákazník přihlásí. Po provedení přihlášení bude tento oddíl zobrazen mezi bloky kategorie

a vyhledávání a blokem navigace. Oddíl obsahuje další navigaci určenou registrovanému zákazníkovi. Odkazy z této sekce jsou určeny pro změnu informací o zákazníkovi, pro získání informací o objednávkách aj.

6.2 IMPLEMENTACE

Ve chvíli, kdy máme připravený návrh layoutu, nastal čas jej aplikovat na šablony. První šablonou, kterou je nutné upravit a která rozvržení stránek e-shopu nejvíce ovlivní, je šablona layout.ascx a po této i ostatní do ní vkládané.

6.2.1 Šablona Layout.ascx

Šablona layout.ascx je tou, ve které jsou uchovány informace o rozvržení stránek a ta, která se načítá při každém požadavku na stránku e-shopu a určuje velké množství vlastností stránky. Kód šablony začíná standardní XML deklarací, za kterou následuje deklarace doctype. V původních šablonách je v doctype deklarovaný jazyk XHTML ve verzi 1.0 Strict. Pro mobilní zařízení je vhodnější dokument ve verzi XHTML určené právě jim. V šablonách jsem tedy deklaroval jazyk XHTML Mobile Profile verze 1.2, a to hlavně z důvodu nejširší nabídky použitelných tagů a také proto, že povoluje použití tagu <button /> (na rozdíl od jiných deklarací pro mobilní zařízení), který je hojně použit ve formulářích na e-shopu a jeho nahrazení úpravou šablon není možné. DTD definice a hlavička použitá v šabloně layout.ascx:

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.2//EN"
"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile12.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs">
  <head id="LayoutHead" runat=server>
    <title><shop:GetText TextID="ShopName" runat=server/></title>
    <link rel="stylesheet" type="text/css"
      href="/inshop/layout/pages/mobile.css" />
    <meta http-equiv="Content-Type" content="text/html;
      cha-set=utf-8" />
    <meta name="author" content="Autor šablony: Tomáš Havránek,
      autor aplikace: Zoner software, s.r.o. " />
    <meta name="keywords" content=" " />
    <meta name="description" content=" " />
  </head>
```

Hlavička šablony již obsahuje i jeden speciální tag aplikace Zoner In-Shop. Tyto tagy začínají slovem *shop*, za kterým je dvojtečka a po ní následuje název tagu určující, jakou funkci tag plní. Tyto tagy také mohou obsahovat další parametry, které budou jejich funkci upřesňovat a zapisují se jako parametry normálního XHTML tagu. Každý tento speciální In-Shop tag musí navíc obsahovat parametr *runat=server*, jinak nebude serverem zpracován. Příklad In-Shop tagu, který je použit pro vložení názvu e-shopu do titulku stránky, je zde:

```
<title><shop:GetText TextID="ShopName" runat=server/></title>
```

V těle šablony jsou umístěny rozhodovací bloky, které podle nastavení e-shopu určují, zda mají být některé části stránek vloženy nebo ne. Jako například u výše zmíněného bloku přihlášeného zákazníka, jehož šablona se do stránky připojí pouze pokud je přihlášen. Tyto bloky jsou uvozeny tagy *<shop:test></shop:test>*, jejichž parametry slouží k určení podmínky, která se má vyhodnotit. Uvnitř těchto rozhodovacích bloků jsou vloženy další tagy, které obalují část kódu, jenž se má vykonat při splnění a nesplnění podmínky. Tag *shop:successtemplate* uvozuje kód, který se vykoná, když je podmínka splněná, jinak kód uvozuje tag *shop:alternativetemplate*. Ukázka rozhodovacího bloku, který vypisuje různý nadpis katalogu, podle toho, zda se jedná o katalog výsledku hledání, nebo normální zobrazení katalogu, je zde:

```
<shop:Test Expression="shop:IsSearchingResult" runat=server>
  <SuccessTemplate>
    <shop:GetText TextId="Catalogue_Searching" runat=server />
  </SuccessTemplate>
  <AlternativeTemplate>
    <shop:GetText TextId="Catalogue_Title" runat=server />
  </AlternativeTemplate>
</shop:Test>
```

Tuto šablonu, jakož i ostatní upravované, jsem se snažil co nejvíce přizpůsobit pro vhodné zapojení do navrženého layoutu. Odstranil jsem co nejvíce obrázků, které by zvětšovaly objem přenesených dat nutných k načtení stránky. Také jsem upravil některé možnosti zobrazení. Například šablona katalogu již neumožňuje zobrazovat obrázky zboží v plné velikosti, nýbrž pokud jsou obrázky povoleny, ať už velké, nebo náhledové, budou zobrazeny jen ty náhledové. Uživatel si samozřejmě může velký obrázek zobrazit, ale až přímo ze stránky s detailním popisem zboží. Dalším,

co jsem ze šablon odstraňoval, byly tabulky, které jsem nahradil jejich rozepsáním do textu. U všech šablon jsem odstranil přebytečné bílé znaky, které jen zvětšují jejich velikost a prodlužují načítání stránek. Tímto a odstraněním některých prvků šablon, jsem velikost několika z nich zmenšil až o 20 KB a celkově jejich úpravou zrychlit načítání stránek e-shopu.

6.2.2 Šablona MobileNavigation.ascx

Šablona MobileNavigation.ascx je šablona, která není standardní součástí balíku šablon Zoner In-Shop 4. Tuto šablonu jsem navrhl pro použití v kolekci šablon určených mobilním zařízením. Je využívána k zobrazení bloku navigace. Šablona v sobě soustřeďuje všechny hlavní odkazy na stránky e-shopu, čímž by měl být usnadněn pohyb v obchodě.

Šablona je vkládána do stránek e-shopu v šabloně Layout.asxc. Jak blok navigace vypadá v e-shopu je zobrazeno na obrázku 8.



Obrázek 8: Blok navigace zobrazený v prohlížeči

6.2.3 Problém kotev a tagu base

Aplikace In-Shop do hlaviček generovaných stránek vkládá tag `<base />`, který určuje základní umístění webu. Toto umístění je pak použito pro správné směrování relativních odkazů na stránce. Aplikace In-Shop relativní odkazy používá a bez definice tohoto tagu ji nelze používat. Bohužel pokud je základní umístění definováno, nelze pro pohyb na stránce použít kotvy. Protože potom místo po aktuální stránce, odkazují na adresu základního umístění.

Tento problém jsem vyřešil implementováním jednoduché Javascript funkce, kterou jsem pojmenoval *anchor*. Tato funkce pomocí změny hodnoty parametru *hash* objektu *location* mění část url adresy za znakem #. Pomocí této změny posunuje stránku na pozici kotvy, jejíž název jsme přiřadili parametru *hash*. Jak tato funkce vypadá, je zde:

```
<script language="javascript" type="text/javascript">
<!--
function anchor(wher)
{
    document.location.hash=wher;
}
-->
</script>
```

Funkce je volána v odkazu, který má posunout stránku. A je volána pomocí přednastavené hodnoty parametru *href*. Hodnota parametru má následující tvar: *Javascript:anchor('název_kotvy');*. Příklad odkazu, jež využívá tuto funkci, je uveden níže:

```
<a href="Javascript:anchor('down');">Skoč dolu</a>
```

6.2.4 Stylopisy

Pro mobilní šablony jsem samozřejmě vytvořil i CSS stylopisy, přesněji jsem vytvořil 4 stylopisy, každý pro jedno barevné schéma, které je standardní součástí aplikace Zoner In-Shop. Tyto stylopisy jsem tvořil hlavně s ohledem, aby jejich použití napomáhalo zpřehlednění struktury stránek, aby texty byly vhodně formátované a dobře čitelné. Snažil jsem se stylopisy vytvořit tak, aby jejich velikost byla co nejmenší. Také jsem dbal na dodržení standardu CSS Mobile Profile 2.

6.3 POROVNÁNÍ STANDARDNÍCH A MOBILNÍCH ŠABLON

S využitím vytvořených šablon oproti standardním jsou stránky e-shopu mnohem přehlednější a lepší pro čtení na mobilních zařízeních. Také navigace po nich se velmi zjednodušila a stránky se staly použitelnějšími. Bohužel v současné době není možné použít oba druhy šablon v e-shopu najednou. Nelze je použít, protože aplikace In-Shop zatím nemá implementován nějaký modul, který by zjistil, s jakým zaří-

zením se uživatel k e-shopu připojuje a podle identifikace zařízení by byly použity správné šablony. Porovnání výsledku práce a standardní šablony je vidět na obrázcích v příloze B.

7 ZÁVĚR

Cílem bakalářské práce bylo vytvořit šablony pro aplikaci Zoner In-Shop 4, které vhodně přizpůsobí stránky aplikace tak, aby ji bylo možné používat i v prohlížečích mobilních zařízení. Navržené řešení spočívá ve vytvoření sady šablon, která se bude používat pro generování kódu webových stránek při přístupu na stránky z mobilního zařízení.

Výsledkem práce je právě taková sada šablon, s pomocí které lze vytvářet webové stránky e-shopu, jež budou vhodné pro zobrazení na mobilních zařízení. Pro implementaci šablon do aplikace In-Shop 4, je nutné ještě upravit její programový kód tak, aby mohla používat 2 sady šablon najednou a rozhodovat jakou sadu šablon použít. V práci jsem také ověřil, že samotné přilinkování kaskádového stylu vytvořeného pro mobilní zařízení a nastavení jeho použití s nimi, není vhodným způsobem řešení pro zpřístupnění stránek z těchto zařízení.

POUŽITÁ LITERATURA

- [1] HLADKÝ, R. *Nástroje pro on-line obchodování na Internetu*. [s.l.] : [s.n.], 2002 tisk. 13 s. Dostupný z WWW: <<http://dsp2002.pef.czu.cz/pdf/dsp-128.pdf>>.
- [2] *HyperText Markup Language - Wikipedie, otevřená encyklopedie* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Html>>.
- [3] *Extensible HyperText Markup Language - Wikipedie, otevřená encyklopedie* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/XHTML>>.
- [4] *Cascading Style Sheets - Wikipedie, otevřená encyklopedie* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Cascading_Style_Sheets>.
- [5] *PHP - Wikipedie, otevřená encyklopedie* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/PHP>>.
- [6] *Active Server Pages - Wikipedia, the free encyclopedia* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Active_Server_Pages>.
- [7] *ASP.NET - Wikipedie, otevřená encyklopedie* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/ASP.NET>>.
- [8] BRANICKÝ, Marek. *JavaServer Pages pro všechny* [online]. 2002 [cit. 2008-05-11]. Dostupný z WWW: <<http://interval.cz/clanky/javaserver-pages-pro-vsechny/>>.
- [9] *Webový server - Wikipedie, otevřená encyklopedie* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Webov%C3%BD_server>.
- [10] *Internet Information Services - Wikipedia, the free encyclopedia* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Internet_Information_Services>.
- [11] *Apache HTTP Server - Wikipedia, the free encyclopedia* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Apache_http>.
- [12] *Database - Wikipedia, the free encyclopedia* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Database>>.

- [13] *MySQL - Wikipedie, otevřená encyklopedie* [online]. 2008 [cit. 2008-05-11].
Dostupný z WWW: <<http://cs.wikipedia.org/wiki/MySQL>>.
- [14] *PostgreSQL - Wikipedia, the free encyclopedia* [online]. 2008 [cit. 2008-05-11].
Dostupný z WWW: <<http://en.wikipedia.org/wiki/PostgreSQL>>.
- [15] *Oracle Database - Wikipedia, the free encyclopedia* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Oracle_database>.
- [16] *Microsoft SQL Server - Wikipedia, the free encyclopedia* [online]. 2008 [cit. 2008-05-11]. Dostupný z WWW:
<http://en.wikipedia.org/wiki/Microsoft_SQL_Server>.
- [17] KRUPKA, Marcel. *Zakládáme e-shop: aplikace, hosting, volba sortimentu - LUPA* [online]. c1998-2008 [cit. 2008-05-11]. Dostupný z WWW:
<<http://www.lupa.cz/clanky/zakladame-e-shop-aplikace-hosting-volba-sortimentu/>>.
- [18] *Softwarové licence -- PC Svět* [online]. c2000-2004 [cit. 2008-05-11]. Dostupný z WWW: <<http://www.pcsvet.cz/art/article.php?id=4616>>.
- [19] *Stránky o svobodném software* [online]. c2004 [cit. 2008-05-11]. Dostupný z WWW: <<http://www.gnu.cz/article/30/>>.
- [20] BRAŠNA, Jan. *Mobilní web prakticky* [online]. c2005 [cit. 2008-05-11]. Dostupný z WWW: <<http://www.janbrasna.com/clanky/mobilni-web-prakticky/>>.
- [21] BRAŠNA, Jan. *Mobilní standardy* [online]. c2005 [cit. 2008-05-11]. Dostupný z WWW: <<http://www.janbrasna.com/clanky/mobilni-standardy/>>.
- [22] RABIN, Jo. *Comparison of XHTML Mobile Profile and XHTML Basic* [online]. c2007 [cit. 2008-05-11]. Dostupný z WWW:
<<http://dev.mobi/article/comparison-xhtml-mobile-profile-and-xhtml-basic>>.
- [23] RABIN, Jo. *Comparison of CSS 2.1, CSS MP, WCSS and CSS Level 1* [online]. 2007 [cit. 2008-05-11]. Dostupný z WWW:
<<http://dev.mobi/content/comparison-css-21-css-mp-wcss-and-css-level-1>>.
- [24] KOKOŠKA, Lukáš. *Jak skládat stránku pro mobilní web* [online]. 10.03.2008 [cit. 2008-05-11]. Dostupný z WWW: <<http://www.mobilniweb.info/jak-skladat-stranku-pro-mobilni-web/22>>.
- [25] KOKOŠKA, Lukáš. *Pár pravidel pro mobilní web* [online]. 31.03.2008 [cit. 2008-05-11]. Dostupný z WWW: <<http://www.mobilniweb.info/par-pravidel-pro-mobilni-web/30>>.

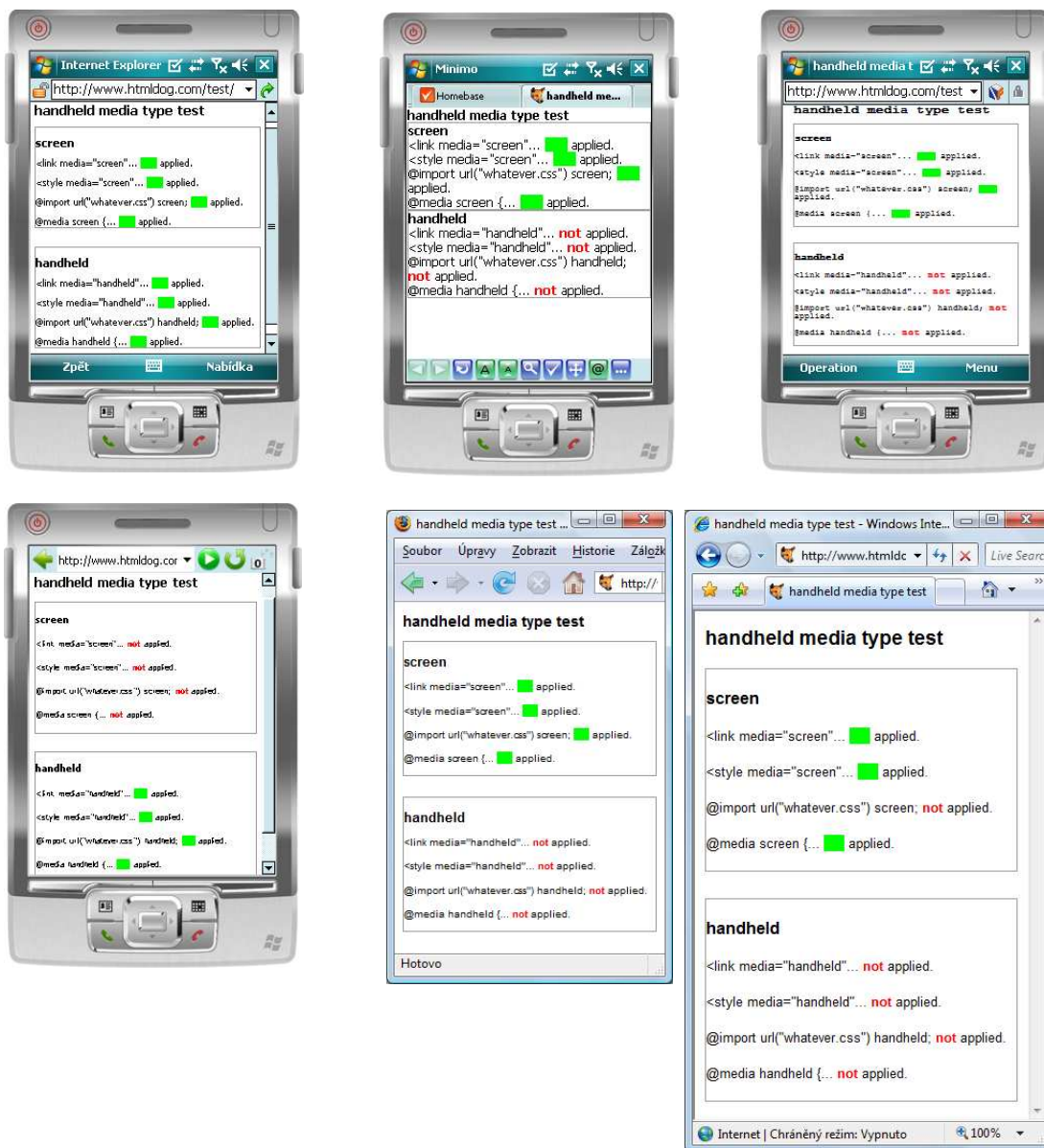
- [26] GRIFFITHS, Patrick. *HTML and CSS for Mobiles* [online]. c2006 [cit. 2008-05-11]. Dostupný z WWW: <<http://www.htmldog.com/ptg/archives/000055.php>>.
- [27] STANÍČEK, Petr. *Praktická příručka CSS, 5. díl - Média* [online]. c2002 [cit. 2008-05-11]. Dostupný z WWW: <http://www.webtip.cz/art/wt_tech_html/wt_cssserial_005.html>.
- [28] BEDNAŘÍK, Bohumír. *Identifikace mobilních prohlížečů* [online]. [2004] [cit. 2008-05-11]. Dostupný z WWW: <<http://bobocop.cz/identifikace-mobilnich-prohlizecu.php>>.
- [29] SCHUBERT, Svante, BERJON, Robin. *CSS Mobile Profile 2.0* [online]. c2007 [cit. 2008-05-11]. Dostupný z WWW: <<http://www.w3.org/TR/css-mobile/>>.
- [30] RABIN, Jo, MCCATHIENEVILE, Charles. *Mobile Web Best Practices 1.0* [online]. c2005-2006 [cit. 2008-05-11]. Dostupný z WWW: <<http://www.w3.org/TR/mobile-bp/>>.

SEZNAM PŘÍLOH

Příloha A: Výsledky testu prohlížečů ve vykreslování stránky s CSS stylem a parametrem media

Příloha B: Porovnání zobrazení e-shopu mobilní a standardní šablony

PŘÍLOHA A: VÝSLEDKY TESTU PROHLÍZEČŮ VE VYKRESLOVÁNÍ STRÁNKY S CSS STYLEM A PARAMETREM MEDIA



Prohlížeče na obrázcích zleva nahoře: MS Internet Explorer Mobile 6.12, Mozilla Minimo 0.2, NetFront Browser 3.5, Opera Mobile 8.65, MS Internet Explorer 7.0.6001.18000, Mozilla Firefox 2.0.0.14.

PŘÍLOHA B: POROVNÁNÍ ZOBRAZENÍ E-SHOPU MOBILNÍ A SANDARDNÍ ŠABLONY



Jazyk: [CS](#) [DE](#)
Výběr měny:

CZK ▾

[Skočit na navigaci](#)

[Úvodní stránka](#)»[Všechno zboží](#)»[Hlavní oddělení](#)»[Hlavní členění](#)»[ATI](#)»[Sapphire](#)
Radeon HD 3850, 256MB DDR3

**Sapphire Radeon HD 3850,
256MB DDR3**

[Dotaz na výrobek](#)

[Našli jste lepší cenu?](#)

[Doporučit výrobek](#)



Výrobce: Sapphire
Katalogové číslo: 01001
Záruka (měsíců): 24
Termín dodání (dny): 1
Skladem: 4 ks

Sapphire Radeon HD 3850, 256MB DDR3,
256-bit, 2xDVI, TVOut, PCIe 2.0

Výrobce čipu: ATI
Verze čipu: Radeon HD3850
Kapacita paměti: 256 MB
Typ paměti: DDR3
Sběrnice: PCI-Express
Frekvence čipu: 668 MHz
Frekvence paměti: 1656 MHz
Šíře sběrnice: 256 bit
Chladič čipu: Aktivní (s ventilátorem)
Televizní výstup: Ano
DVI výstup: Ano
HDMI výstup: Ne
VIVO: Ne

1 [Koupit](#)

Vaše cena bez DPH: 2 924 Kč

Vaše cena s DPH: 3 479,50 Kč

Kategorie

- [ATI](#)
 - [HD series](#)
 - [X series](#)
- [nVidia](#)
- [ostatní](#)

Vyhledávání

v celém obchodě ▾

• [Rozšířené vyhledávání](#)

Navigace v e-shopu

[Titulní stránka](#)
[Přihlášení](#)
[Ceník](#)
[Katalog](#)
[Nákupní košík](#)
[Pokladna](#)
[Filtr](#)
[Novinky na shopu](#)
[Výprodeje](#)
[Rychlá objednávka](#)
[Kontakt](#)
[Obchodní podmínky](#)
[Reklamační podmínky](#)
[Jak nakupovat](#)

Aktualizováno

27.4.2008

Copyright © 2007 ZONER software, s.r.o.,
www.zoner.cz

Copyright © 2007 DPS Technology s.r.o.,
www.dps.cz

Provozováno na systému Zoner INSHOP4.,
www.zoner-inshop.cz

Obrázek: e-shop vykreslený s použitím mobilní šablony

INSHOP4
Zoner

Vaše nová cesta
[Titulní stránka](#) [Přihlášení](#) [Ceník](#) [Katalog](#)
[Nákupní košík](#) [Pokladna](#)

Vyber měny:
 CZK ▼
Celková cena nákupu
 bez DPH: 0 Kč
 s DPH: 0 Kč

Rychlá objednávka
 • [Kontakt](#)
 • [Obchodní podmínky](#)
 • [Reklamační podmínky](#)
 • [Ask nákupcevi](#)

Vyhledávání

 v celém obchodě ▼
 • [Rozšířené vyhledávání](#)

Filter
 Zobrazit **pouze výrobky**
 ▼
 akce
 novinky
 výprodej
 připravujeme
 tip
 skladem

Kategorie
 • [ATI](#)
 • [DVIDIA](#)
 • [ostatní](#)

Výprodeje
 MSI NX6200AY-TD128LF

 s DPH: 660 Kč
 ASUS FAX1050/TD/256M/A

 s DPH: 720 Kč
 Aktualizováno
 27.4.2008

[Úvodní stránka](#) » [Všechno zboží](#) » [Hlavní oddělení](#) » [Hlavní členění](#) » [ATI](#) » [Sapphire](#)
 Radeon HD 3850, 256MB DDR3
Sapphire Radeon HD 3850, 256MB DDR3

[Dotaz na výrobek](#)
[Našli jste lepší cenu?](#)
[Doporučit výrobek](#)



Výrobce: Sapphire
 Katalogové číslo: 01001
 Záruka (měsíců): 24
 Termín dodání (dny): 1
 Skladem: 4 ks
 Sapphire Radeon HD 3850, 256MB DDR3, 256-bit, 2xDVI, TVOut, PCIe 2.0

Výrobce čipu: ATI
 Verze čipu: Radeon HD3850
 Kapacita paměti: 256 MB
 Typ paměti: DDR3
 Sběrnice: PCI-Express
 Frekvence čipu: 668 MHz
 Frekvence paměti: 1656 MHz
 Šíře sběrnice: 256 bit
 Chladič čipu: Aktivní (s ventilátorem)
 Televizní výstup: Ano
 DVI výstup: Ano
 HDMI výstup: Ne
 VIVO: Ne

1

Vaše cena bez DPH: 2 924 Kč
 Vaše cena s DPH: 3 479,50 Kč

Jazyk:
 CS DE

Přihlášení zákazníka
 Jméno:
 Heslo:

• [Nová registrace](#)
 • [Zapomněli\(a\) jsem heslo](#)

Novinky na shonu
 MSI NX8400GS-TD256E
 bez DPH: 672,50 Kč
 s DPH: 800 Kč

 Gainward BLT55 9600GT Golden Sample 512MB
 bez DPH: 3 866 Kč
 s DPH: 4 600 Kč

 VSA MSI NX8800GT-TZDS12E-OC
 bez DPH: 4 370 Kč
 s DPH: 5 200 Kč

Rychlá objednávka
 • [Kontakt](#)
 • [Obchodní podmínky](#)
 • [Reklamační podmínky](#)
 • [Ask nákupcevi](#)

Copyright © 2007 ZONER software, s.r.o.,
www.zoner.cz
 Copyright © 2007 DPS Technology s.r.o.,
www.dps.cz
 Provozováno na systému Zoner INSHOP4.,
www.zoner-inshop.cz

Obrázek: e-shop vykreslený s použitím standardní šablony