

Webový e-shop a jeho správa

Jan Turek

Bakalářská práce
2014



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan Turek**
Osobní číslo: **A11173**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Webový e-shop a jeho správa**

Zásady pro vypracování:

1. Vytvořte webový e-shop s možností správy.
2. Zpracujte literární rešerši na dané téma.
3. Navrhněte a vytvořte vzhled prezentace a obchodu.
4. Počítejte s velkým množstvím položek, navrhněte třídění položek do jednotlivých skupin.
5. Aplikaci doplňte o rozšířené vyhledávání, systém optimalizujte pro nejznámější vyhledávače.
6. Při návrhu dbejte na bezpečnost aplikace z hlediska známých útoků.
7. Vytvořený systém umístěte na vhodný webový server.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. PEACOCK, Michael. Programujeme vlastní e-shop v PHP 5. Vyd. 1. Brno: Computer Press, 2011, 334 s. ISBN 978-80-251-3181-7.
2. KOFLER, Michael a Bernd ÖGGL. PHP 5 a MySQL 5: průvodce webového programátora. Vyd. 1. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.
3. LACKO, Ľuboslav. PHP a MySQL: hotová řešení. Brno: CP Books, 2005, 299 s. ISBN 80-251-0397-8.
4. TEAGUE, Jason Cranford. DHTML a CSS pro World Wide Web: praktická vizuální příručka. Praha: SoftPress, 2005. ISBN 80-864-9777-1.
5. ZAKAS, Nicholas C, Jeremy PCPEAK a Joe FAWCETT. Ajax: profesionálně. Vyd. 1. Překlad Jiří Koutný. Brno: Zoner Press, 2007. ISBN 978-80-86815-77-0.
6. SCHLOSSNAGLE, George. Pokročilé programování v PHP 5. Brno: Zoner Press, 2004. ISBN 80-868-1514-5.
7. JQuery [online]. 2014 [cit. 2014-01-30]. Dostupné z: <http://jquery.com/>
8. PHP [online]. 2014 [cit. 2014-01-30]. Dostupné z: <http://www.php.net/>

Vedoucí bakalářské práce:

doc. Ing. Martin Sysel, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

28. února 2014


Termín odevzdání bakalářské práce:

13. června 2014

Ve Zlíně dne 28. února 2014


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ke ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- Že odevzdaná verze diplomové/bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 13. 6. 2014

.....
podpis diplomanta

ABSTRAKT

Vytvořená bakalářská práce „Webový e-shop a jeho správa“ rozšiřuje pole působnosti firem o možnosti nabídnout své zboží díky celosvětové síti Internet. Jedná se o finální řešení internetového obchodu, včetně správy, a umožňuje majiteli nasadit na webový server již hotovou aplikaci. Celá aplikace pracuje online a poskytuje uživateli přívětivé prostředí jak k nákupu, tak i jeho správě. Řešení využívá technologie HTML5, JavaScript, PHP, AJAX a několik frameworků postavených na uvedených technologiích. Díky těmto technologiím byl vytvořen zabezpečený systém z hlediska nejznámějších útoku.

Klíčová slova: E-shop, web, PHP, MySQL, JavaScript, jQuery, Internet, administrace

ABSTRACT

Created bachelor thesis "Web e-shop and its management" broadens the scope of possibilities for companies to offer their products through a global network Internet. This is the final solution of the Internet business, including management, and allows the owner to use the application on a web server. The entire application works online and provides user-friendly environment for both the purchase and its management. The solution leverages HTML5, JavaScript, PHP and several frameworks built on these technologies. With these technologies was created a secure system in terms of the most famous attack.

Keywords: E-shop, web, PHP, MySQL, JavaScript, jQuery, Internet, administration

Primárně bych chtěl poděkovat svému vedoucímu práce, panu doc. Ing. Martinu Syslovi, Ph.D. především za jeho ochotu, výbornou spolupráci a také cenné rady, které jsem v průběhu svojí práce potřeboval jako sůl hesla.

Dále bych chtěl poděkovat svojí rodině a všem blízkým, kteří mne po celou dobu studia podporovali a v případě nouze nade mnou nezlomili hůl.

OBSAH

ÚVOD	10
1 VZHLED INTERNETOVÉHO OBCHODU	12
1.1 POUŽITÉ PROSTŘEDKY	12
1.1.1 Hypertextový značkovací jazyk (HTML)	12
1.1.2 Kaskádové styly (CSS)	13
1.2 GRAFICKÝ NÁVRH APLIKACE.....	13
1.2.1 Grafický nástroj <i>GIMP 2.8</i>	14
1.2.2 Úzký grafický návrh a jeho popis	14
1.2.3 Široký grafický návrh a jeho popis	15
1.2.4 Grafický návrh pro administraci a jeho popis	16
1.3 DALŠÍ POUŽITÉ GRAFICKÉ DOPLŇKY	17
2 TŘÍDĚNÍ POLOŽEK	18
2.1 MOŽNOSTI TŘÍDĚNÍ	19
2.2 TECHNICKÉ ŘEŠENÍ.....	20
2.2.1 Realizace na straně databáze	20
2.2.1.1 Tvorba a editace skupiny	21
2.2.1.2 Zavedení položky do požadované skupiny.....	22
2.2.2 Realizace pomocí serverového skriptovacího jazyka.....	22
2.2.2.1 Třída Group.....	23
2.2.2.2 Třída Menu	23
2.2.2.3 Třída Paging.....	27
2.2.3 Realizace na straně klienta	27
2.2.3.1 Animace menu	27
2.2.3.2 Označování aktuálně procházené skupiny.....	29
2.2.3.3 Zasílání asynchronních požadavků pro změny menu	30
3 VYHLEDÁVÁNÍ A OPTIMALIZACE PRO VYHLEDÁVAČE	32
3.1 VYHLEDÁVÁNÍ V OBCHODU.....	32
3.1.1 Technické řešení vyhledávání	33
3.1.1.1 Realizace na straně databáze.....	33
3.1.1.2 Realizace na straně serveru.....	35
3.1.1.3 Realizace na straně klienta.....	38
3.2 OPTIMALIZACE PRO VYHLEDÁVAČE	40
3.2.1 Vybrané prvky SEO	40
3.2.1.1 Kvalitní a unikátní obsah	40
3.2.1.2 Používání HTML značek podle normovaných předpisů	41
3.2.1.3 Používání titulku, nadpisů a popisů	41
3.2.1.4 Krátká a neměnná URL adresa	42
3.2.1.5 Kanonizační problémy.....	42
3.2.1.6 Budování zpětných odkazů.....	43
3.2.1.7 Korektní používání souboru robots.txt	43
3.2.1.8 Používání description, keywords	44
3.2.1.9 Aktivity na sociálních sítích	44
3.2.2 Přátelské URL odkazy.....	44
3.2.2.1 Technické řešení	45
3.2.3 Open Graph	46

4	ZABEZPEČENÍ APLIKACE	47
4.1	OWASP ASVS	48
4.1.1	Úrovně ASVS.....	48
4.1.1.1	Úroveň 0: Cursorsy	48
4.1.1.2	Úroveň 1: Oportunistic	49
4.1.1.3	Úroveň 2: Standard	49
4.1.1.4	Úroveň 3: Advanced	49
4.1.2	Detaily požadavků ASVS	49
4.1.2.1	V1: Požadavky na ověření autentizace	49
4.1.2.2	V2: Požadavky na ověření správy relací.....	50
4.1.2.3	V3: Požadavky na Access Control List	52
4.1.2.4	V4: Požadavky na validaci vstupů.....	53
4.1.2.5	V5: Požadavky na kryptografické služby	55
4.1.2.6	V6: Požadavky na manipulaci s chybami a logy	55
4.1.2.7	V7: Požadavky na ochranu dat	55
4.1.2.8	V8: Požadavky na zabezpečení komunikace	57
4.1.2.9	V9: Požadavky na zabezpečení HTTP.....	57
4.1.2.10	V10: Požadavky na ověření škodlivých požadavku.....	58
4.1.2.11	V11: Požadavky na ověření obchodní logiky	58
4.1.2.12	V12: Požadavky na zabezpečení souborů a jiných zdrojů	58
4.1.2.13	Požadavky na zabezpečení mobilní platformy.....	59
4.1.3	Popis plnění požadavku ASVS	59
4.1.3.1	V1: Požadavky na ověření autentizace	59
4.1.3.2	V2: Požadavky na ověření správy relací.....	61
4.1.3.3	V3: Požadavky na Access Control list.....	62
4.1.3.4	V4: Požadavky na validaci vstupů.....	63
4.1.3.5	V5: Požadavky na kryptografické služby	65
4.1.3.6	V6: Požadavky na manipulaci s chybami a logy	65
4.1.3.7	V7: Požadavky na ochranu dat	65
4.1.3.8	V8: Požadavky na zabezpečení komunikace	66
4.1.3.9	V9: Požadavky na zabezpečení HTTP.....	66
4.1.3.10	V12: Požadavky na zabezpečení souborů a jiných zdrojů	67
4.2	DALŠÍ ČÁSTI ZABEZPEČENÍ	68
4.2.1	Třída Security	68
4.2.1.1	Popis funkce třídy	69
5	NASAZENÍ APLIKACE	70
5.1	POŽADAVKY NA WEBHOSTINGOVÉ SLUŽBY	70
5.2	PRŮBĚH NASAZENÍ APLIKACE PRO ÚČELY TESTOVÁNÍ NA LOKÁLNÍM SERVERU	71
5.3	PRŮBĚH NASAZENÍ APLIKACE DO OSTRÉHO PROVOZU NA PRONAJATÝ SERVER.....	71
5.3.1	Založení webhostingu a domény	72
5.3.2	Konfigurace služeb.....	72
5.3.2.1	Vytvoření databáze	73
5.3.2.2	Vytvoření servisního emailu	74
5.3.2.3	Povolení FTP pro administrátorskou síť.....	75
5.3.2.4	Povolení přístupu pomocí FTP	76
5.3.3	Nahrání aplikace na server	76
5.3.4	Konfigurace aplikace při prvním spuštění	77

5.3.5 Konfigurace při spuštění aplikaci	77
ZÁVĚR	78
SEZNAM POUŽITÉ LITERATURY	80
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	85
SEZNAM OBRÁZKŮ	87
SEZNAM PŘÍLOH.....	89

ÚVOD

V dnešní uspěchané době plné technologií, které usnadňují lidstvu působení v různých směrech, našly své místo i internetová řešení různých obchodů, kde firmy nabízejí své produkty. Právě široké rozšíření Internetu otevřelo firmám cestu, jak efektivněji nabídnout zboží potenciálním zákazníkům a tím zvýšit svůj zisk.

Nyní jsou k dispozici mnohé řešení e-shopů různých organizací, které poskytují aplikaci zdarma, nebo za nějaký poplatek. Snahou takových řešení je jistá modularita, díky které může firma uspokojit své požadavky. Moduly však často nebývají zcela zdarma a k jejich implementaci povětšinou bývá potřeba ruka programátora.

Navržený systém byl zkonstruován jako celek, ke kterému bude možné instalovat doplňky, jenž mohou být v budoucnu vyvíjeny. Přehledné uživatelské prostředí se snaží zákazníka informovat o veškerém jeho pohybu a vysvětlovat mu aktuálně prováděné akce tak, aby se snadno v obchodu orientoval. Aplikace počítá s větším množstvím položek, které mohou být dosti různorodé, a z tohoto důvodu bylo navrženo třídění do jednotlivých skupin, kde každá skupina může sdružovat až čtyři podskupiny. Mnoho internetových obchodů, při grafickém návrhu opomíjí přehlednost, což vede k neúspěchu celé aplikace. Velice důležitým aspektem pro toto řešení byla právě uživatelská přívětivost, jak z pohledu zákazníka, tak administrátora systému. Z tohoto důvodu obsahuje aplikace velké množství uživatelských doplňků, které toto hledisko realizují, daní je náročnost celé aplikace, která však není nijak kritická.

V systému je možné vyhledávat zboží buďto pouhým zadáním klíčového slova, nebo rozšířeným vyhledáváním, kde je umožněno filtrovat zboží dle specifikovaných kritérií, jako je například rozsah ceny, jestli je položka zlevněná a podobně. S vyhledáváním souvisí taktéž optimalizace pro vyhledávače, což je základem pro úspěšný internetový obchod. Optimalizace částečně spočívá využitím přátelských URL adres, které roboti různých vyhledávacích společností indexují, a také vhodným postavením kostry indexované stránky.

Dalším důležitým aspektem pro systém byla jeho bezpečnost. Snahou bylo případnému útočníkovi co nejvíce zúžit cestu pro útok. Aplikace je zabezpečena z hlediska známých útoků, včetně AJAXových požadavků a do administrátorského rozhraní je dovolen přístup pouze s protokolem HTTPS, kde certifikát poskytuje správce webového serveru. Při nasazování e-shopu je nutné na tuto skutečnost pamatovat, a prověřit si zda poskytovatel webových služeb certifikát poskytuje, popřípadě je nutné si zřídit vlastní.

Aplikace jako celek byla postavená na open source technologiích, a díky tomu se snížily náklady na realizaci. Bylo využito databázového prostředí MySQL, skriptovacího jazyka PHP a klientského jazyka JavaScript. Dále byly využity doplňky, díky kterým se práce na vývoji zjednodušila, a umožnila tak rychleji nasadit složitější funkce, které by v opačném případě zabraly mnoho hodin práce.

1 VZHLED INTERNETOVÉHO OBCHODU

Trendy webdesignu se neustále vyvíjejí turbulentním způsobem. Pokud se navrhne vzhled webové aplikace, dá se předpokládat, že průměrně za rok již bude zastaralý. Současným trendem jsou pastelové barvy a velice strohá grafika, kde primární vzhled utvářejí obrázky s vysokým rozlišením v pozadí, na které jsou naneseny jednoduché elementy nejčastěji ve 2D prostoru. Samozřejmě praktiky jednotlivých webdesignerů se diametrálně liší. Pro tuto aplikaci byl zvolen spíše starší design, při kterém byly použity zaoblené rohy a stíny, ale objevují se zde i prvky současného trendu. Dnešním trendem bývá aplikace responsivního designu, díky kterému se aplikace dokáže přizpůsobit velikosti zobrazovacího zařízení. Pohledy na výše uvedené fakty se mohou různit. [12]

Aplikace byla navržena primárně pro bezdotyková zařízení. U dotykových zařízení je navržený vzhled funkční, ale nedochází k přizpůsobení rozlišení aplikace, k rozlišení displeje.

1.1 Použité prostředky

O grafické zpracování se především zasloužily kaskádové styly (CSS – Cascading Style Sheet) [4], a také rastrový editor *GIMP* 2.8 [9]. Program byl zvolen z důvodu jeho volné publikace a zároveň splňoval všechny nároky na navržený design. Aplikace využívá jazyk HTML5, tím bylo umožněno použít CSS, jenž je k němu vázán.

1.1.1 Hypertextový značkový jazyk (HTML)

Tento jazyk je v současnosti nejpoužívanějším jazykem, který slouží pro publikaci prezentací na Internetu. Využívá jej i řada aplikací, která nejsou určena pro online nasazení, především z důvodu jeho snadné implementace a jednoduchosti. Díky němu lze rozvrhnout základ dokumentů, a o to jak bude následně vypadat, se mohou postarat kaskádové styly.

Bohužel i zde můžeme narazit na nekompatibilitu jednotlivých prohlížečů, což má za následek chybné zobrazení prezentace. Aplikace byla navržena tak, aby se zobrazovala přinejmenším obdobně v různých prohlížečích a nedocházelo k rozbití celého formátování.

V současnosti je připravována verze HTML5, která již lze použít v navrhovaných dokumentech. Verze by se měla stát oficiálním standardem v roce 2014 [10]. Na této verzi byla částečně postavena i navržená aplikace, která je zpětně kompatibilní s původní verzí bez

funkčnosti některých částí, ale vždy se jedná o drobná vylepšení, která při nefunkčnosti nezpůsobuje žádné selhání.

1.1.2 Kaskádové styly (CSS)

S použitím stylů CSS neboli kaskádových stylů lze velmi efektivně a jednoduše nastylovat veškeré dokumenty, které využívají nějaký značkovací jazyk, například HTML. Na jednom centrálním místě se můžou nastavit styly CSS tak, aby ovlivňovaly vzhled značek HTML na jediné webové stránce, nebo v celém rozsahu webu[4].

V současnosti se stala standardem verze CSS3, která byla vyvíjena dlouhá léta. Tento standard ulehčil vývojářům spoustu práce a díky jeho nasazení lze řadu problému řešit kaskádovými styly, nikoli obrázky podporující průhlednost a podobně. Paradoxem je skutečnost, že vývoj webového designu je natolik turbulentní, a tudíž některé novinky, které CSS3 přineslo, jsou v současnosti opět nepotřebné. Na druhou stranu je k dispozici mocný nástroj, který umožňuje prezentaci zdynamizovat a není k tomu potřeba tolik kódu v jazyce JavaScript. U veškerých webových technologií se potýkáme s nekompatibilitou jednotlivých prohlížečů a ani CSS3 se tento nedostatek nevyhnul. Vždy vyjde nějaká novinka, prohlížeče ji nepodporují, a vývojáře tyto nedostatky natolik omezují, že jsou nuceni používat starší způsoby řešení.

1.2 Grafický návrh aplikace

Aplikace se sestává ze tří grafických návrhů (layout), kde první dva jsou vizuálně totožné, ale jejich primárním rozdílem je šířka prezentace. Užší návrh je určen pro stránky, které mají informativní charakter a působí spíše jako prezentace firmy. Druhý návrh slouží pro samotný e-shop, který obsahuje navíc postranní menu, díky kterému aplikace narostla do šířky. Třetí grafický návrh je implementován do administrátorské části aplikace. Veškeré grafické prvky byly umísťované do samostatných vrstev v grafickém prostředí *GIMP 2.8* a následně z využitím HTML5 a CSS3 umístěny a nastylovány do základního vzhledu aplikace jakožto webového dokumentu. Responzivní design nebyl na tuto aplikaci aplikován, jelikož aplikace byla vyvinuta pro bezdotyková zařízení a současné rozlišení displejů takových zařízení je dodatečně velké nato, aby se e-shop zobrazil celý.

1.2.1 Grafický nástroj GIMP 2.8

Jedná se o rastrový grafický editor s částečnou podporou vektorové grafiky. Vyniká možnostmi, které jsou srovnatelné s komerčními programy, jako je například *Adobe Photoshop*. Nesrovnatelnou výhodou je však rozdíl jejich pořizovacích cen, jelikož projekt *GIMP* se nalézá pod licencí „General Public License“ GPL, a tudíž je zdarma. [9]

1.2.2 Úzký grafický návrh a jeho popis



Obr. 1 Rozvržení úzkého grafického návrhu

Do tohoto grafického návrhu (layout) byly implementovány:

1. logo internetového obchodu
2. dvě horizontální menu obsahující možnosti jako běžné prezentace firmy bez internetového obchodu (O firmě, Kontakt, Obchodní podmínky a podob.), dále obsahují možnost registrace a přihlášení, je zde zobrazen aktuálně přihlášený uživatel
3. vyhledávací pole aplikace
4. zde se nalézá aktuální stav košíku
5. prostor vyhrazený pro prezentaci obsahu stránek, využívajících tento layout
6. panel s aktuálně zlevněným zbožím
7. novinky ve firmě a obchodu

8. aktuálně podporované druhy plateb
9. aktuálně podporované způsoby dopravy
10. pata aplikace informující o provozovateli a právech

1.2.3 Široký grafický návrh a jeho popis

The screenshot displays the FOX Kancelářská Technika website. At the top, there is a navigation bar with links for 'O firmě', 'Kariéra', 'Obchodní podmínky', and 'Kontakt'. Below this is a search bar and a shopping cart icon showing 'Váš košík: 0 Kč'. The main navigation menu on the left includes categories like 'Tonerý Cartridge', 'Tisková zařízení', 'Multifunkce', 'Kopírky', 'Tiskárny', 'PC IT', 'Dokumentační technika', 'Zabezpečovací a docházkové systémy', 'Kancelářské potřeby', 'Satelitní technika', 'Elektronika', 'Pokladní systémy', 'Dekorace', and 'Bazar'. The main content area is titled 'Multifunkční stroje' and shows a price range from 5 158 Kč to 86 390 Kč. It includes filters for 'Novinka', 'Výprodej', 'Akční zboží', and 'Bazar'. Below the filters, there are two product cards: 'HP LaserJet Pro M1536dnf' (5 230 Kč bez DPH, 6% discount) and 'HP LaserJet 700 color MFP M775DN' (82 000 Kč bez DPH, 5% discount). A sidebar on the right features 'Akční zboží' with a 'Sleva 11%' offer on 'HP CF283A 83A' toner. The footer contains the text: 'Provozovatel: Copyright © 2014 Jan Turek, Vytvořeno Turekphoan WD. Individuálně navržena webová aplikace.'

Obr. 2 Rozvržení širokého grafického návrhu

V tomto grafickém návrhu byly použity veškeré části uvedené v úzkém grafickém návrhu mimo prostoru, který byl vyhrazen pro prezentaci jednotlivých stránek. Aplikace narostla do šířky a byla rozšířena o části:

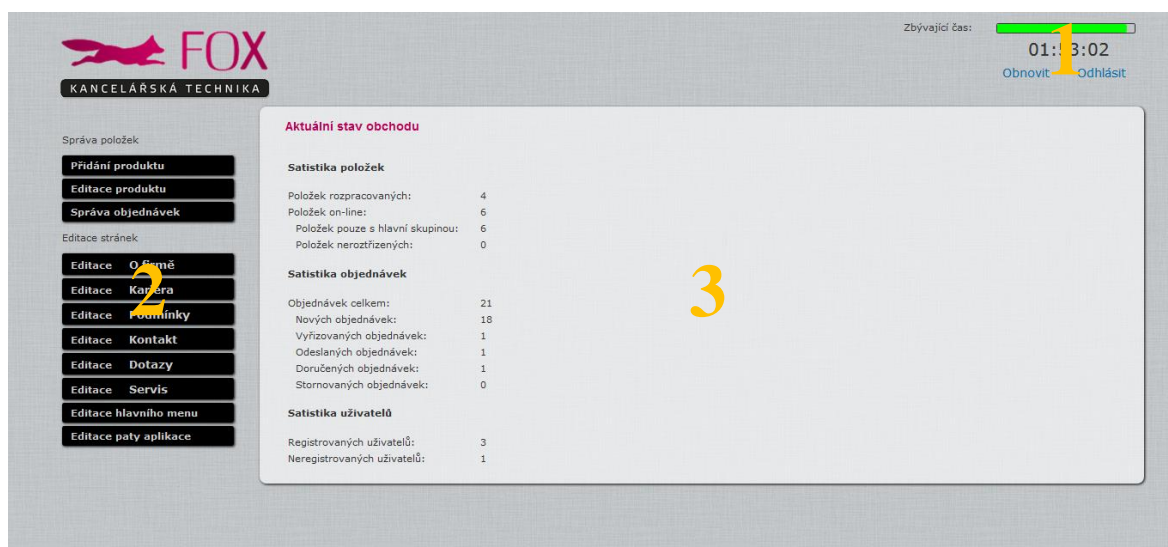
1. jednotlivé skupiny obchodu vyobrazené v hlavním vertikálním menu obchodu
2. filtrování položek, které slouží i jako rozšíření vyhledávání
3. vyobrazené zboží, které odpovídá zvolené skupině a filtru
4. zobrazení počtu položek na stránku

Vzhled, který je uveden na Obr. 2, není výchozí obrazovkou obchodu. Z důvodu optimalizace pro vyhledávače byl zvolen návrh, kde 4. oddíl byl pomyslně rozdělen na tři části, ve

kterých jsou pod sebou umístěny aktuálně probíhající akce, novinky a výprodeje. Jednotlivé položky však vypadají totožně jako v uvedeném návrhu.

1.2.4 Grafický návrh pro administraci a jeho popis

Pro administrátorské rozhraní byl částečně využit projekt „roundcube“, který poskytuje projekt webmail v rámci licence Creative Commons [58]. Z projektu byl použit přihlašovací formulář a pozadí aplikace.



Obr. 3 Rozvržení návrhu pro administraci

V grafickém návrhu samotné administrativní části pozůstalo z projektu „roundcube“ pouze pozadí. Návrh byl celkově rozdělen do třech částí, kde nejvíce měněný obsah je v bloku 3.

Celkové rozvržení pak:

1. informační panel s ukazatelem zbývajících času pro administraci
2. hlavní menu administrativního prostředí
3. oblast pro správu jednotlivých částí aplikace, volených z hlavního menu

Ukazatel zbývajících času je tvořen jednak odpočtem, ale také vizuálním panelem, kterému je v závislosti na zbývajícím času měněna barva.

Pro návrh menu nebylo zvoleno tak dynamické rozhraní jako pro hlavní menu obchodu, jelikož v této části aplikace nebylo počítáno s potomstvem pod jednotlivými položkami. Množství editačních možností tudíž není tak objemné.

Na úvodní straně, která byla uvedena na Obr. 3 Rozvržení návrhu pro administraci Obr. 3, je možné pozorovat základní informace o aktuálním stavu obchodu, čímž se administrátora

snaží upozornit na nové objednávky, popřípadě nutnost provést přeřazení položek pod příslušné skupiny.

1.3 Další použité grafické doplňky

Dále byl v administrativním rozhraní přidán plugin *qtip2* [43], díky kterému jsou řešena dialogová okna podávající informace o editaci, na tomto výkonném tooltipu¹ je postaveno i rozšíření označený jako projekt *jGrowl v1.2.12* [44], který na pravé straně informuje uživatele o stavu provedené akce.

Dalším projektem, který byl do aplikace umístěn, je *jQuery MsgBox* [45]. Jedná se o náhradu standartních dialogových oken „`alert()`“, „`prompt()`“ a podobně, které jsou interními funkcemi JS. Za pomoci tohoto doplňku byla celá aplikace uživatelsky zpřívětivěna.

V aplikaci byl využit editor obrázku a textu pro editaci a tvorbu textů, popřípadě tedy obrázku v částech webové prezentace, a také byl použit pro možnost popisu položek. Jedná se o velmi výkonný a bohatě zdokumentovaný editor *CK Editor* [3]. Díky jeho nasazení do aplikace bylo autorovi ulehčeno nemalé úsilí s tvorbou vlastního editoru, který by byl na podobné úrovni. K tomu modulu byl vytvořen doplněk, díky kterému je možno data z editoru ukládat s využitím AJAX.

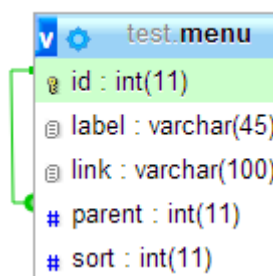
¹ Tooltip – jedná se o velmi krátký popis, zobrazovaný nejčastěji při podržení ukazatele nad objektem

2 TŘÍDĚNÍ POLOŽEK

Jelikož aplikace musí počítat s obrovským množstvím různorodého zboží, bez nějaké možnosti položky třídit, se obchod neobejde. Třízení taktéž souvisí s problematikou optimalizace pro vyhledávače, protože názvy jednotlivých skupin musí obsahovat specifické rysy. Optimalizaci je věnována samostatná kapitola, kde byla tato problematika popsána.

Položky jsou roztrženy do jednotlivých skupin, kde celek veškerých skupin vytváří stromovou strukturu. Metody třídění položek přímo souvisí s generováním menu obchodu, na kterém právě lze stromovou strukturu pozorovat.

Na internetu jsou k dispozici již hotová řešení, která by se dala jednoduše implementovat. Takové menu mají nespornou výhodu v jednoduchosti návrhu, tak i operacemi nad jednotlivými skupinami. Většina současných aplikací takto navržené menu využívá. Z hlediska databáze může být příslušná tabulka realizována následovně.



Obr. 4 Standardní řešení menu, generováno zdrojem [61]

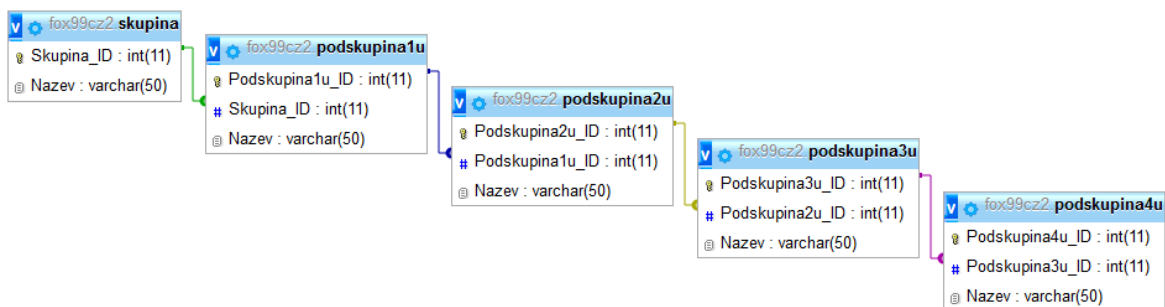
Řešení uvedené na (Obr. 1) je hojně již dlouhou dobu využíváno, toto řešení bývá označováno jako „parent-child relationship“, nebo „parent-child dimension“. Řešení lze nalézt ve zdrojích [25][40].

S takto nadržným modelem je zajištěna konzistence dat, kdy rodič obsahuje cizí klíč potomka. Další výhodou na takto navrženém modelu je možnost přidávat další dimense, a to bez nutnosti měnit již navržený model. Elegantně lze řešit i mazání jednotlivých skupin, jelikož zmíněnému cizímu klíči může být přidán parametr „ON DELETE CASCADE“. Následně pokud přijde požadavek na smazání skupiny, která je rodičem dalších podskupin, provede se automatické mazání v celém rozsahu, a není potřeba situaci dále ošetřovat. Tato skutečnost nebyla ve zdrojích uvedena, ale je možná. Nevýhodou návrhu menu je fakt, že nelze vygenerovat jediným SQL dotazem SELECT, a vždy musí být použit rekursivní algoritmus. Tento problém má negativní dopad na rychlost celé aplikace. Situace by byla řešitel-

ná generováním do externího souboru, odkud by se menu následně zobrazovalo do obchodu. V praxi situace bývá řešena právě tímto způsobem.

Způsobů řešení menu je více, a možnosti jejich zobrazení dokonce ještě více. Situace může být generována pokaždé, nebo do externích souborů, a tyto metodiky se různí.

V původním návrhu problému bylo menu realizováno soustavou 5ti tabulek, které byly vzájemně propojeny cizími klíči, toto řešení bylo příliš složité na obsluhu, a tak byl návrh přepracován.



Obr. 5 Původní databázové rozdělení do skupin, generováno zdrojem [61]

V současném stavu bylo autorem navrženo menu s odlišnou vizí, oproti situaci uvedené na Obr. 4 Standardní řešení menu, generováno zdrojem [61] Byl vytvořen model, s jehož pomocí lze menu vygenerovat jediným SQL dotazem SELECT, a tak ulehčit práci celému systému, daní takového řešení je nutnost ošetřit konzistenci dat manuálně, popřípadě s využitím triggerů (trigger²). Do jisté míry bylo zvoleno toto řešení z důvodu autorovy neznalosti řešeného problému, menu je naprostým unikátem v problematice třídění, je plně funkční a splňuje předpoklady pro nasazení. S navrženým menu bylo potřeba ošetřit i několik dalších problémů, které jsou popsány v kapitole 2.2 Technické řešení.

2.1 Možnosti třídění

Počet skupin, do kterých je možno položky roztřídit, bývá upravováno možnostmi platformy, kde je aplikace umístěna. Každá hlavní skupina umožňuje sdružovat až čtyři úrovně podskupin. Další podskupiny nebyly do obchodu zaneseny z důvodu přehlednosti a systém neumožňuje další dimenzi menu přidat. Předpokládá se, že celkového počtu skupin upravovaného dle možnosti platformy, nebude nikdy dosaženo.


² Trigger neboli spouštěč, je myšleno vykonání dodatečné procedury, při operacích s tabulkou

2.2 Technické řešení

Třídění položek bylo navrženo originálním způsobem. Při návrhu byl kladen důraz na rychlost aplikace, ale tento záměr se částečně kryje s množstvím kódu v JS. Jak již bylo uvedeno v úvodu této kapitoly, jedná se o zcela originální řešení tohoto problému a jeho úskalím je především množství podpůrných systémů pro jeho funkčnost. Ve výsledku však menu funguje na všech platformách, včetně dotykových zařízení, tedy z pohledu obchodníka je navržený systém v pořádku. Hotové řešení by bylo pouze přeneseno do této práce, ale projekt se snaží být co nejvíce originálním. Modul třídění, a následného zobrazení menu využívá veškeré technologie, které byly v tomto projektu použity. Nejvíce vytěžovaným článkem bylo zvoleno PHP.

2.2.1 Realizace na straně databáze

Jednotlivé skupiny jsou ukládány do tabulky „skupiny“. Tabulka abstinuje cizími klíči, a z toho důvodu bylo nutné zajistit konzistenci dat jinými mechanismy, které jsou plně řešeny na straně databáze, avšak ne za pomoci cizích klíčů, ale procedur a triggerů. Dále byly popsány mechanismy operací nad tabulkou, které se snaží nekonzistentnost eliminovat. K dispozici je 5 úrovní skupin, proto tabulka obsahuje sloupce „u1“ až „u5“, namísto identifikátoru rodiče a potomka. Dále tabulka obsahuje sloupec „Pozice“, který slouží pro přemístění skupiny ve výpisu na jiné místo, sloupce „Název“, „Plny_nazev“ a identifikátor skupiny „Skupiny_ID“.



	Skupiny_ID	u1	u2	u3	u4	u5	Pozice	Nazev	Plny_nazev
Skupiny_ID : int(10) unsigned	1	1	0	0	0	0	0	Tonery ...	Tonery a C...
# u1 : int(10) unsigned	12	1	1	0	0	0	0	Inkoust...	Cartridge a ...
# u2 : int(10) unsigned	46	1	1	1	0	0	0	Originální	Originální c...
# u3 : int(10) unsigned	47	1	1	2	0	0	1	Neorigi...	Neoriginální...
# u4 : int(10) unsigned	48	1	1	3	0	0	2	Renovace	Renovovan...
# u5 : int(10) unsigned	13	1	2	0	0	0	1	Lasero...	Tonery a sp...
# Pozice : int(10) unsigned	49	1	2	1	0	0	0	Originální	Originální to...
# Nazev : varchar(40)	50	1	2	2	0	0	1	Neorigi...	Neoriginální...
# Plny_nazev : varchar(60)									

Obr. 6 Struktura a náhled uspořádání dat tabulky „Skupiny“, generováno zdrojem [61]

Z hodnot ve sloupcích „u1“ až „u5“ lze vypočítat chronologickou strukturu dělení skupin. Například ze záznamu 1 je patrné, že se jedná o hlavní skupinu, ze záznamu 13 lze vyčíst, že je potomkem 1. skupiny, a nalézá se na druhé pozici v příslušné úrovni.

2.2.1.1 *Tvorba a editace skupiny*

Při zakládání nové skupiny je vykonán SQL příkaz „INSERT“. Před fyzickým provedením příkazu dochází ke spuštění triggeru („TRIGGER BEFORE INSERT“), který byl pojmenován „CountNewIdandPositionRestrictSameName“, a následně dojde k vložení nové skupiny. Triggerem je upraven mechanismus uložení nové skupiny. Plní funkci automatické inkrementace ve vkládané úrovni skupiny, a současně určuje i číslo nové pozice skupiny. Mechanismus je postaven na aktuálním stavu tabulky, kde se nejprve zjišťuje maximální hodnota ve sloupci příslušné úrovně vkládané skupiny, následně je tato hodnota navýšena o jedničku a připravena pro uložení do sloupce, další jeho činností je výpočet nové pozice vkládané skupiny, kdy se obdobně zjišťuje maximální pozice příslušné úrovně skupiny, ta je taktéž navýšena o jedničku a připravena pro vložení do sloupce „Pozice“, trigger se snaží pokrýt nedostatek MySQL, kde není interně povoleno zakázat vložení záznamu v případě nějaké podmínky. Tento problém byl vyřešen nastavením sloupce „Pozice“ tak, aby neakceptoval hodnotu „NULL“. Pokud se v příslušné úrovni již nalézá skupina se shodným názvem, dojde k nastavení nově vypočtené pozice na hodnotu „NULL“(ASCII 0), která po vykonání příkazu „INSERT“ selže s chybou „(1048) Column 'Pozice' cannot be null“. Tato chyba je následně odchycena skriptovacím jazykem. Problém by nebylo možné vyřešit nastavením unikátního klíče k sloupci „Nazev“, jelikož existují skupiny, které v kombinaci s jinou podskupinou mohou mít název stejný.

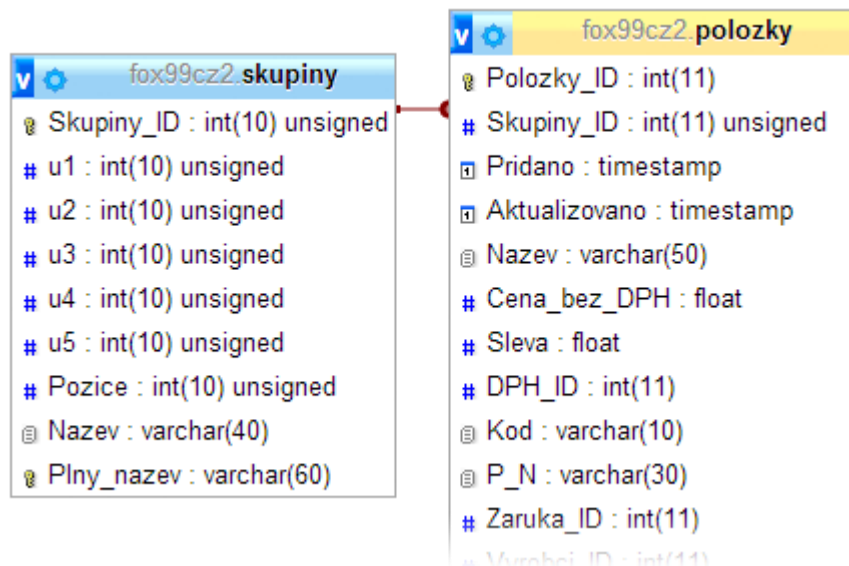
Pokud je skupina editována, je použit SQL příkaz „UPDATE“. K příkazu je navázán trigger s názvem „RestrictSameName“, který před fyzickou aktualizací údajů ověří, jestli již není takto pojmenována jiná skupina. V případě že dojde k nalezení shody, provede se totožná operace jako u vkládání nové skupiny, nastaví se aktualizovaná hodnota sloupce „Pozice“ na „NULL“, čímž dojde k selhání uložení.

Z pohledu databáze bylo nutné ošetřit konzistenci dat, aby před smazání nadřazené skupiny byly smazány veškerí potomci této skupiny. Ve standartním způsobu řešení je kritérium řešeno právě cizími klíči a nastavením příznaku klíče „ON DELETE CASCADE“. Navržené menu touto možností abstinuje, a bylo potřeba mechanismus ošetřit manuálně. Z toho důvodu je zakázáno přímo mazat záznamy v tabulce manuálně, ale musí být použita procedura „SafeDeleteGroup(_Skupiny_ID)“, která ošetří tento nedostatek, smaže veškeré potomstvo a „vrátí“(vyselektuje) identifikátor nadřazené skupiny pro jednodušší orientaci v editaci. Současně procedura přemísťuje položky pod rodičovskou skupinu, a pokud se jedná o

hlavní skupinu, jsou položky označeny jako neroztříděné. Následně je nutno provést jejich přeřazení.

2.2.1.2 Zavedení položky do požadované skupiny

Položky jsou zavedeny do zvolené skupiny tím, že je umístěn do sloupce „Skupiny_ID“ v tabulce „polozky“ cizí klíč.



Obr. 7 Zanesení položky do skupin

Z obrázku (Obr. 7) je patrné, jak je skupinová tabulka provázaná s tabulkou „polozky“. Při výpisu položky je provedeno spojení těchto tabulek a tím je získán přístup na ostatní sloupce tabulky „skupiny“.

2.2.2 Realizace pomocí serverového skriptovacího jazyka

Na straně databáze jsou ošetřovány problémy způsobené nekonzistentním návrhem a jsou zcela odstíněny od řešení problematiky skriptovacím jazykem.

Bez jazyka PHP se při řešení problému třídění položek nelze obejít. Pro aplikaci byly vytvořeny dvě třídy „Group“ a „Menu“.

2.2.2.1 Třída Group

Objekt byl primárně vyvinut jako datový typ, ale posléze bylo vhodné spíše definovat část aplikace, která nese charakteristické rysy třídy ORM³, avšak ne všechny vlastnosti byly implementovány. Snahou byla zajistit částečná persistence dat, kde vytvořením této třídy usnadnilo veškerou práci související s tříděním položek. Pomocí takového objektu bylo možné utvořit šablonu oddílu menu, snáze celé menu generovat, ukládat aktuálně zobrazenou skupinu produktů pro účely pozdějšího zpracování, zobrazení, nebo editace. Podmínky ORM třídy stanoví jednoznačně oddělení objektu od relační databáze. Navržená třída neimplementuje metody, které by zajišťovaly komunikaci s databází. Objekt slouží sice pro operace nad řádkem tabulky „skupiny“, ale komunikace s *MySQL* byla ponechána třídě Menu v plném rozsahu. Důvodem byl fakt, že pro generování menu je vytvářeno větší množství těchto objektů, které však s databází ve skutečnosti vůbec komunikovat nemusí a zabíraly by akorát paměť serveru. Autor si uvědomuje, že PHP bylo vytvořeno s automatickým uvolňováním paměti na konci skriptu [8], ale právě z důvodu většího množství vytvářených objektů byla implementována metoda „close()“, která se snaží uvolnit operační paměť již za chodu. Ve skutečnosti se však jedná o pouhé sdělení, že proměnná již nebude potřebná, a *Garbage Collector* (GC) PHP se postará o uvolnění paměti automaticky při cyklu procesoru, kdy nebude prováděna důležitější operace [8]. Do třídy byly samozřejmě implementovány i další metody. Získávací metody getter⁴ a samozřejmě také nastavovací metody setter⁵. Dále třída obsahuje výčtově metody pro: zjištění úrovně skupiny „getLevel()“, získání identifikátoru rodičovských skupin „getLevelID(\$level)“, označení skupiny jakožto nově vytvářeného objektu, který má být následně zanesen do relační databáze „markUsNew()“, a další. Z důvodu přehlednosti zde tato třída nebyla umístěna, ale lze ji nalézt v příloze (Příloha P I), konkrétně „xampp/htdocs/www/private/classGroup.php“.

2.2.2.2 Třída Menu

Další v pořadí byla vytvořena třída „Menu“. Tento objekt souvisí se zobrazováním a upravitelstvím hlavního menu obchodu, ve kterém byly jednotlivé skupiny vypsané. V objektu

³ Object Relational Mapping – část aplikace, která konvertuje řádek z relační databáze do objektového vyjádření, snahou je jistá persistence dat a zjednodušení vývoje aplikace. [42]

⁴ Getter – jedná se o funkci objektově orientovaného programování, kdy jsou získávány vlastnosti objektů.

⁵ Setter – jedná se o funkci objektově orientovaného programování, kdy jsou nastavovány vlastnosti objektů.

jsou právě používány objekty „Group“, díky kterým byla celá třída zpřehledněna, jelikož umožnily zjednodušit operace, které s tříděním položek souvisí. Dále její metody zpřístupňují aplikaci informace o aktuálně procházené skupině a jejích parametrech. V neposlední řadě jsou třídou „Menu“ zpracovávány přátelské URL odkazy, které identifikují navštívenou skupinu v průběhu nákupu, v této souvislosti ošetřuje bezpečnostní riziko spojené se zadáním chybné URL adresy, která neidentifikuje žádnou skupinu způsobem vyvolání chybového „Status Code“ HTTP protokolu 404, stránka nenalezena [10]. Třída pro svou činnost vyžaduje připojení na databázi, a z toho důvodu dědí metody třídy „Connect“.

```
1  <?php
2  /** Class Menu */
3  class Menu extends Connect{
4      /** @var Skupiny $ActualGroupsObject */
5      protected $ActualGroupsObject;
6      /** Konstruktoem je možno nastavit aktuální skupinu */
7      public function __construct($intSkupinyIDActual = 0,
                                   Array $arrGroupLevels = [],
                                   Array $arrGroupNames = []){}
8
9      /** Nastaví minulou skupinu jako aktivní, definuje potomky */
10     public function wakeActualGroup(){}
11     /** Implementuje menu do obchodu ze souboru /parts/menu.html */
12     public static function printMenu(){}
13     /** Publikuje editované menu do souboru /parts/menu.html */
14     public static function saveMenuIntoWebsite(){}
15     /** Šablona pole menu */
16     public function menuBoxRender(Skupiny $Group,$haveUnderLevel,
                                   Array $levelsNames = [],$hide=true){}
17     /** Šablona editačního pole menu skupiny */
18     public function editableMenuBoxRender(Skupiny $Group,$hide=true){}
19     /** Generuje nové menu a ukládá do temp souboru /parts/menuTemp.html */
20     public function generateMenu(){}
21     /** Generuje editační menu pro administraci */
22     public function generateMenuEditable(){}
23     /** Generuje k aktuální skupině URL */
24     public function getURLActualGroup(){}
25     /** Generuje URL podle identifikátoru skupinu */
26     public function getURLGroupBySkupinyID($skupinyID){}
27     /** Getter aktuální skupiny */
28     public function getActualGroup(){}
29     /** Edituje název a plný název skupiny */
30     public function editActualGroup($nameGroup='', $nameGroupFull=''){}
31     /** Vkládá novou skupinu do databáze kde uID === NULL,
32      * funkce při úspěchu nastaví novou aktuální skupinu */
33     public function addActualGroup($nameGroup='', $nameGroupFull=''){}
34     /** Maže skupiny a všechny její potomky v databázi */
35     public function deleteActualGroup(){}
36     /** Getter aktivní Skupiny_ID */
37     public function getActiveGroupID(){}
38     /** Getter názvu skupiny definové úrovně k aktuální skupině */
39     public function getNameGroup($level){}
40     /** Getter plného názvu aktuální skupiny */
41     public function getFullNameActualGroup(){}
42     /** Dle ID urovni v poli nastaví aktivní skupinu a potomky */
```

```
42     public function setActualGroupFromLevels(Array $arrNewGroupActual =
43                                             []){}
44     /** Dle identifikátoru skupiny nastaví aktuální skupinu a potomky */
45     public function setActualGroupBySkupinyID($skupinyID){}
46     /** Dle URL nastaví aktuální skupinu, při nenalezení chyba 404 */
47     public function setActualGroupFromURL(Array $arrNamesGroups =[],
48                                         \Slim\Slim $SlimApp = null){}
49     /** Zneplatní aktuální skupinu */
50     public function unsetActualGroup($skupinyID){}
51 };
```

Zdrojový text 1 Třída Menu a její metody

Struktura třídění vychází z databázového modelu a třída Menu je určena pro řízení třídění na serverové úrovni. Její metody uvádí Zdrojový text 1 Třída Menu a její metody.

Generování menu je spjato s rekurzivním algoritmem, který musí provádět mnohé operace komparace. Výhodou je použití právě takové struktury databázové tabulky, kde jsou veškerá data týkající se menu získána jediným dotazem, a tak není spojení vytěžováno. Pokud by se pro každého návštěvníka generovalo menu, hardware by byl zbytečně zatěžován a systém by jednou zhavaroval. Pro zrychlení obchodu se menu zanesou do obchodu pouze po dokončení úprav na tomto modulu stiskem tlačítka „Uložit“, na stránce editace hlavního menu v administrativním rozhraní. Takto navržený modul při generování nového menu, respektive vždy po jeho zobrazení v administraci, vytváří kopii generovaného HTML do souboru „/xampp/htdocs/www/parts/menuTemp.html“, který je po úpravách připraven pro nasazení jako hlavní menu obchodu. Díky této koncepci je velice snadné ověřit finální vzhled menu a zamezit případnému výpadku databáze, nebo jiné deformace, což by vedlo k destrukci celého menu, ve výsledku i obchodu. Obecně u veškerých úprav platí pravidlo, že změny by měly být prováděny v době, kdy je předpokládán nízký provoz aplikace. Nastává zde problém související s nekompatibilitou nově generovaného a původního menu. Pokud dojde k editaci menu kdy je smazán, popřípadě editován hlavní název záznamu skupiny, dojde k zneplatnění odkazu v menu obchodu. Tato situace je vyřešena až po zanesení nově generovaného menu do obchodu metodou „SaveMenuIntoWebsite()“ kdy dojde k přepsání původního menu ze souboru „/xampp/htdocs/www/parts/menu.html“ souborem „/xampp/htdocs/www/parts/menuTemp.html“, a tak je nutností vždy po editacích provést vložení nového menu do obchodu. Problém by bylo možné vyřešit několika metodami, které však nebyly aplikovány.

Tato třída implementuje metody, které slouží pro editaci jednotlivých skupin obchodu. Přidání skupiny, editace jejího názvu a plného názvu nejsou nikterak obtížné operace, jež jsou realizovány vždy jediným dotazem na databázi. Na straně *MySQL* jsou spuštěny před

těmito operacemi triggerů, které se postarají o korekci, případně zamítnutí, vkládaných dat. V případě, že se uživatel snaží uložit skupinu pod názvem, jež se v konkrétní úrovni již nachází, je ze strany databáze zpracována chyba 1048. Tato chyba byla popsána v kapitole 2.2.1.1 Tvorba a editace skupiny. Může nastat i chyba s kódem 1062, která v konkrétním případě značí pokus vložení skupiny již s existujícím plným názvem skupiny. PHP tyto chyby odchyťává, a předává ve formě indexů pro další zpracování například JS. Při vkládání se využívá pole úrovní skupin „u1“ až „u5“, kde na přidávané úrovni se nachází textový řetězec „NULL“. Takto zvolený systém má výhodu jednoduchosti následného zpracování požadavku, kde identifikátory skupiny jsou přetypovány na datový typ „integer“ a řetězec „NULL“ je převeden na hodnotu „NULL“, tím jsou připravena data specifikující umístění nové skupiny a nemusí se dále určovat nová pozice. Nevýhodou je fakt, že systém zabírá více paměti a je potřeba vytisknout více kódu. Situace je řešitelná předáváním identifikátoru nadřazené skupiny, kdy by se identifikátory zjistili od rodiče. V tomto případě by pak nejspíše na straně databáze byla vytvořena procedura, která by uvedenou činnost realizovala. Přidávání skupiny realizuje metoda „addActualGroup()“.

U editace skupiny je předáván identifikátor skupiny a nové hodnoty parametrů. Následně je vykonán SQL příkaz „UPDATE“, před kterým dochází ke spuštění vytvořeného triggeru (trigger before update) pojmenovaného „RestrictSameName“, jež zajišťuje totožný systém validace jako u přidávání nové skupiny, s tím rozdílem, že se jedná o pouhou aktualizaci údajů.

Mazání údajů je komplikovanější operací, jelikož navržená databázová tabulka neobsahuje cizí klíč, s jehož pomocí by se elegantně mazání provedlo. Bylo nutné utvořit systém, který by se postaral o konzistentnost dat v případě smazání rodičovské skupiny obsahující potomstvo. Opět je tato činnost ponechávána spíše na straně databáze, konkrétně proceduře „SafeDeleteGroup(_Skupiny_ID)“, a PHP skript pouze zpracovává stav provedení operace, případně nastavení rodičovské skupiny mazané skupiny jako aktivní, pro účely jednodušší orientace při editacích menu.

Veškeré výše uvedené nástroje pro aktualizaci skupin byly spíše realizovány na straně databáze, následující popsané metody byly utvořeny pro zpřístupňování informací o procházených skupinách celé aplikaci, čímž bylo zjednodušeno celé rozhraní obchodu.

Díky těmto metodám je efektivně generována navigace v obchodu, jelikož byly informace o menu částečně persistovány od databáze do úložiště relace session⁶. Navigace úzce souvisí s problematikou SEO⁷, ta byla popsána v kapitole 3.2 Optimalizace pro vyhledávače.

Tato třída je umístěna v souboru „./xampp/htdocs/www/private/classMenu.php“.

2.2.2.3 Třída *Paging*

Jelikož se v obchodu může nalézat nepřeborné množství položek, musí být výstup nějakým způsobem stránkovan. Hlavně z důvodu přehlednosti a především rychlosti aplikace.

Tato činnost je obstarávána třídou *Paging*, která vypočítává limit a offset pro SQL dotazy, a zároveň formátuje odkazy stránek [16]. Nepojednává sice přímo o problematice třídění položek, ale se zobrazením a tříděním má také co dočinění.

Třída byla vytvořena autorem „Mik“ zdroje [16], kde je dostupná její dokumentace, avšak byla poupravena tak, aby bylo možné na další stránky přecházet s pomocí JS technologie AJAX, tím nebyla zbytečně znovu načítána celá stránka, pouze oddíl filtru a výčtu položek. Díky této úpravě bylo ulehčeno serveru.

2.2.3 Realizace na straně klienta

V aplikaci je samotný JavaScript použit spíše zřídka, většina kódu využívající tento jazyk byla napsána s využitím knihovny *jQuery* [7].

Tato část je spojená spíše s uživatelským rozhraním a obstarává funkce.

- Animace menu
- Označování aktuálně procházené skupiny
- Zasílání asynchronních požadavků pro změny menu

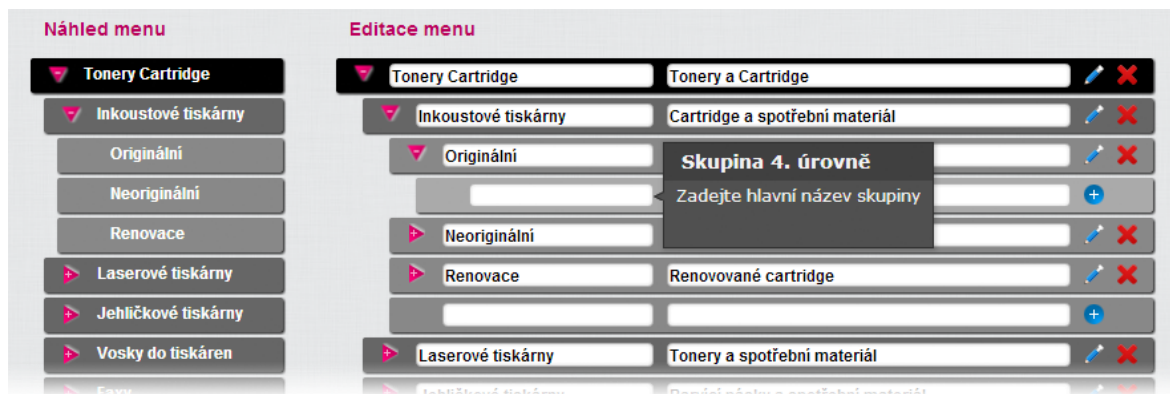
2.2.3.1 *Animace menu*

V internetovém obchodě se může nacházet velké množství skupin. Z tohoto důvodu bylo potřebné navrhnout zobrazení, u kterého by bylo možné schovávat například podúrovně, protože menu může být extrémně dlouhé.

⁶ Session - v překladu relace, je lokálním úložištěm v serveru, které umožňuje předávat data mezi stránkami aplikace. [8]

⁷ SEO – metodika návrhu aplikace, aby byla srozumitelná pro roboty (crawlers) internetových vyhledávačů

Na dnešní poměry bylo zvoleno celkem složité menu s využitím animací „slideUp()“ a „slideDown()“, které zahrnuje *jQuery* [7]. Dále byl využit zásuvný modul (plugin⁸) s názvem *Rotate-jQuery*, díky kterému jsou natačeny šipky při rozbalení menu [12].



Obr. 8 Zpracování menu s možností editace

Veškeré tyto grafické efekty jsou sice výpočetně náročnější, ale i současné počítače s minimální hardwarovou konfigurací vykreslují menu plynule. Skokově je menu vykreslováno pouze na zařízeních typu tablet, chytrý telefon a podobně, kde je hardwarová konfigurace minimální. Po testech, které byly prováděny na jednotkách zařízení různých konfigurací, bylo skokově menu vykreslováno pouze na zařízeních, které přišly jako noviny v roce 2010. Tento průzkum je dosti subjektivní, ale primárně aplikace nebyla vyvinuta pro provoz na takových zařízeních. Chování není vůbec kritické a nezpůsobuje pád prohlížeče. Tedy lze i bez responzivního designu nakupovat s minimálním omezením.

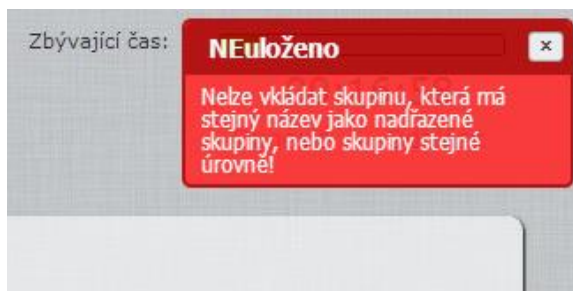
Z pohledu administrace je celá správa zdynamizována, a umožňuje tak velmi přehlednou editaci. Uživatel jen informován o veškerých činnostech modulem *jQuery jGrowl* [44].



Obr. 9 – Ukázka úspěšného uložení editované skupiny modulem *jQuery jGrowl* [44].

⁸ Plugin – modul, který nepracuje samostatně, ale je pouze doplňkem jiné aplikace

V případě chybného zadání dat, které je odchyceno na straně klienta, dochází ke generování objektu modulu *jQuery MsgBox* [45], který o vzniklém problému uživatele podrobně informuje. Pokud je požadavek zjištěn až na serveru, je generován opět objekt *jQuery jGrowl* [44], s rozdílným podbarvením elementu a textovou zprávou chyby, případ lze pozorovat na obrázku (Obr. 10).



Obr. 10 - Ukázka neúspěšného uložení editované skupiny modulem *jQuery jGrowl* [44].

Handlery (handlers⁹), které obstarávají události pro rozbalování a schovávání skupin menu, byly umístěny do souboru „/xampp/htdocs/www/jquery/menu.js“, konkrétně ve funkci „applyHandlersOnMenu()“.

2.2.3.2 Označování aktuálně procházené skupiny

Snahou zrychlit zobrazení menu v obchodu, a snížit tak počet dotazů na databázi, se zne-možnilo znázorňovat aktuálně procházenou skupinu při zobrazení stránky. Tato činnost tedy zbyla na JS.

Při zobrazování stránky server před menu vytiskne skrytý DOM¹⁰ element input, jehož hodnota nese jedinečný identifikátor skupiny. JS po načtení stránky přečte tuto hodnotu, nalezne element obsahující jedinečný identifikátor, rozbalí všechny nadřazené skupiny menu a podbarví aktuální skupinu.

Činnosti je realizována funkcí „markActiveGroup()“, jež byla umístěna do souboru „/xampp/htdocs/www/jquery/menu.js“.

⁹ Handler – v konkrétní případě se jedná o metodu, která slouží pro obstarávání definovaných událostí JS

¹⁰ DOM – rozhraní pro programování aplikací (API), definuje logickou strukturu HTML a XML dokumentů. [10]

2.2.3.3 Zasílání asynchronních požadavků pro změny menu

Celkově jsou vytvořeny tři požadavky, vytvoření, editace a mazání skupiny. Činnost je postavena na technologii AJAX, kterou implementuje *jQuery*.

Místo klasického modelu webových aplikací, kde je za odeslání požadavku na server (a zpracování odpovědi ze serveru) zodpovědný samotný prohlížeč, poskytuje AJAX střední vrstvu pro řízení této komunikace. Engine¹¹ AJAXu je ve skutečnosti nějaký objekt, nebo funkce JS, která je vyvolána vždy, když je potřeba získat informace od serveru, popřípadě je pouze odeslat. Výhodou je, že celá komunikace probíhá asynchronně, tedy další zpracování kódu nečeká na odpověď. [5]

Na server je JS zaslán požadavek vykonat příslušnou akci, server vrátí výsledek ve formátu JSON¹² a JS dále výsledek zpracuje.

S činností přidávání skupiny souvisí i přidání DOM elementů, ve kterém je skupina vykreslena. Celou činnost vložení skupiny obstarává funkce „`add_column()`“. Za předpokladu korektního zápisu názvu skupiny, a také plného názvu skupiny se po kliknutí na tlačítko „Přidat“ provede zaslání požadavku na server. V případě úspěchu je přijat formát JSON, ve kterém jsou umístěny objekty „`newEditableBox`“, „`newBoxView`“ ve formátu HTML a identifikátor nové skupiny. Objekty jsou následně vykresleny na správné místo pomocí *jQuery* a je proveden zápis identifikátoru do skrytého DOM elementu `input`, jež je použit pro označování aktivní skupiny. Následně je volána funkce „`markActiveGroup()`“. Ta provede rozbalení a zvýraznění aktivní skupiny.

U mazání objektu je činnost obdobná, avšak žádné objekty nejsou přidávány, nýbrž odebírány, aby byl zcela využit potenciál technologie AJAX. Opět je zaslán požadavek na server a v případě úspěchu je navrácen identifikátor rodičovské skupiny. Tento identifikátor je zanesen do skrytého DOM elementu `input`, a následně volána funkce „`markActiveGroup()`“, která provede rozbalení potřebné úrovně skupiny.

Obě výše uvedené činnosti jsou provedené jak pro náhled menu, tak i pro menu, kde je umožněna editace.

¹¹ Engine – míněno z pohledu „software engine“, jedná se o výkonné jádro aplikace. [5]

¹² JSON – jedná se o syntaxi zápisu dat, určenou pro výměnu textových dat, podobně jako XML. [5]

S činností pouhé editace souvisí funkce „`edit_column()`“, která taktéž zasílá požadavek na server, a v případě správného vyhodnocení je do náhledu menu zanesen editovaný název skupiny.

Veškeré AJAX požadavky využívané v aplikaci jsou zabezpečeny pomocí tzv. tokenu. Celý mechanismus byl popsán v kapitole 4.2 Další části zabezpečení.

3 VYHLEDÁVÁNÍ A OPTIMALIZACE PRO VYHLEDÁVAČE

Vyhledávání a optimalizace pro vyhledávače jsou části, bez kterých se dnešní e-shop neobejde, obě tyto činnosti spolu úzce souvisí. Je velice důležité, aby byly produkty a veškeré informace o firmě poskytovány do informačních systémů nejpoužívanějších vyhledávačů, a proto je taktéž důležité zajistit optimalizaci aplikace tak, aby byla pro tyto systémy snadno pochopitelná „indexovatelná“. Systém kde je správně zvoleno řešení těchto činností, má slušný předpoklad se umístit ve vyšších sférách vyhledávačů, a tak maximalizovat zisk.

Do aplikace byly obě techniky začleněny, ale pro úspěšný e-shop pouze takové řešení nestačí, dále je podstatný přístup vlastníka e-shopu k problematice marketingu, ale toto není předmětem práce.

3.1 Vyhledávání v obchodu

V současné situaci je možno využít některého z již hotových řešení vyhledávání, nebo lze zkonstruovat vlastní mechanismy, kde je posléze možno specifikovat mnohem více parametrů, popřípadě za nadstandardní služby vyhledávačů zaplatit.

Mezi nejpropracovanější algoritmy vyhledávání patří jednoznačně řešení od společnost *Google* označené jako *Google Custom Search Engine* (dale jen CSE) [47]. Díky funkci CSE je možné zřídit vyhledávací pole a případným dalším nastavením formuláře upravit vzhled stránky s vyhledanými výsledky. Toto řešení má velkou výhodu v naprosté jednoduchosti implementace cizího vyhledávání do vlastní aplikace a usnadňuje tak nemalou práci s tvorbou vlastních algoritmů.

Další možností je využít parazitního formuláře, který bude zasílat metodou GET dotaz na servery společnosti *Google*. Tato možnost byla uvedena *Dušanem Janovským* na stránkách „<http://jakpsatweb.cz>“ [48]. Představuje naprostý základ vyhledávání ve vlastní aplikaci. Předpokladem funkčnosti je, že stránka již byla v minulosti indexována robotem společnosti *Google*. Autor uvádí i další možnosti nastavení přizpůsobení vyhledávače a vyhledávaných výsledků. Nevýhodou těchto řešení jsou zobrazované reklamy, a taktéž žádná možnost zpracovat externí data ve vlastní aplikaci.

Tato společnost dále ve své aplikaci uvádí možnost získat nalezené výsledky formou XML dokumentu, který může být dále zpracován ve vlastní aplikaci. Na první pohled pak vůbec nemusí být k rozeznání od vlastního vyhledávání. Mezi další přednosti patří rozsáhlé možnosti nastavení služby, statistiky a podobně [47]. Bohužel za tuto službu je společností

účtován roční poplatek dle počtu dotazů za toto období, poplatek však není nijak neadekvátní. Pro větší organizace je tato služba dosti lukrativní, jednak z důvodu výhradního postavení mezi vyhledávači, ale také dále z důvodu možnosti nastavení plánovače indexování navržené aplikace, a tím vytěžení co nejvyšší pozice ve vyhledávači nejen této společnosti, ale i jiných organizací využívajících jádra vyhledávání od společnosti *Google*.

Za pomoci českého vyhledávače *Seznam*, lze vyhledávat pouze využitím parazitního formuláře zasílající data metodou GET. Takovým formulářem je umožněno omezené nastavení několika parametrů, jako je například počet nalezených výsledků atd. [49]

Na internetu jsou dostupná mnohá řešení vyhledávání, která souvisí s fulltextovým vyhledáváním v aplikaci, některé metodou frameworků, jiné pomocí externích zdrojů a podobně.

V navržené aplikaci byl zvolen návrh vlastního vyhledávání. Sice veškeré placené služby společnosti *Google* autora přesvědčili o vhodnosti použití metody, avšak při návrhu aplikace byl brán zřetel na ekonomickou stránku, a tak bylo zvoleno vyhledávání vlastní.

3.1.1 Technické řešení vyhledávání

Primárním požadavkem systému byla uživatelská přívětivost, tedy v případě využití řešení od společnosti *Google* by musel být využit formát XML, který by bylo možno dále zpracovávat například pro zobrazení v našeptávači.

V aplikaci bylo implementováno jednoduché fulltextové vyhledávání, které je prováděno spíše heuristickým přístupem.

3.1.1.1 Realizace na straně databáze

MySQL databáze umožňuje fulltextové prohledávání sloupců tabulek s využitím funkcí „MATCH“ a „AGAIN“, na které je aplikován fulltextový klíč, toto pravidlo neplatí pro „BOOLEAN MODE“, ale vyhledávání je bez aplikace klíče pomalejší [11]. Díky této metodice je možno vyhledávat v obchodě dle relevance s využitím přirozeného jazyka („IN NATURAL LANGUAGE MODE“). Tato funkce byla možná, až do verze *MySQL 5.6*, použít pouze pro úložišť „MyISAM¹³“, současná verze umožňuje použít fulltextové vyhledávání na tabulky

¹³ MyISAM – úložisko relační databáze MySQL, s výhodou se používá v aplikacích, kde převažuje náročný výběr dat příkazem SELECT, neumožňuje toliko funkcí co jeho nástupce InnoDB. [11]

s úložištěm „InnoDB¹⁴“. Veškeré tabulky použité v projektu využívají právě úložiště „InnoDB“.

Při návrhu vyhledávání byl nalezen třídílný test, který byl proveden senior konzultantem *Erine Souhrada* [50]. Senior konzultantem byl prováděn test pro možný přechod z úložiště „MyISAM“ na „InnoDB“. Z celého testu vyplynulo, že použití fulltextového vyhledávání je na úložišti „InnoDB“ pomalejší, než na jeho předchůdci. Výsledky vyhledávání taktéž nejsou totožné a je nutné složitěji ošetřit znakovou sadu. S aplikací fulltextového vyhledávání na tomto úložišti je spojeno mnoho problému včetně nutnosti individuálního nastavení databáze u poskytovatele serveru, který požadavek na změnu nemusí akceptovat.

Erine Souhrada doslova uvedl:

„What’s my overall take on InnoDB FTS in MySQL 5.6? I don’t think it’s great, but it’s serviceable. [50]”

Autorem volně přeloženo: *Erine Souhrada* celý test zakončuje se slovy „Nemyslím si, že fulltextové vyhledávání na úložišti InnoDB je dobré, ale je to prospěšné.“

Po pečlivém zvážení veškerých úskalí spojeným s fulltextovým vyhledáváním na úložišti „InnoDB“. Bylo rozhodnuto proces vyhledávání postavit na prosté operaci „LIKE“, která je součástí *MySQL*. V podstatě se taktéž jedná o fulltextové vyhledávání.

Vyhledávání je v případě obchodu prováděno v názvu produktu, a zároveň jeho plného popisu. Uvnitř administrace bylo aplikováno vyhledávání, které umožňuje položce přiřadit související položku, k vyhledávání jsou připojeny ještě záznamy čísla P/N (Sloupec „P_N“), a také kódu ve (sloupci „Kod“).

Pro vyhledávání byly zvoleny právě tyto záznamy, jelikož se v nich nejčastěji nalézají hledaná klíčová slova.

Na straně databáze byly umístěny dvě procedury, jež navrací nalezené výsledky. První procedura byla pojmenována „MIN_MAX_POCTY_SEARCH()“. Tato procedura vyselektuje jednotlivé údaje pro filtr aplikace, následně jsou na straně PHP dopočítány limit a offset pro

¹⁴ InnoDB – úložiště relační databáze MySQL, které je nástupce úložiště MyISAM. Podporuje transakce a je náročnější na systémové prostředky, ale zpřístupňuje mnohé funkce, které MyISAM postrádá

následující proceduru, která byla pojmenována „Select_items()“, ta provede fyzický výběr dat.

3.1.1.2 Realizace na straně serveru

Pro veškeré operace s položkami byla vytvořena třída „Items“. Díky ní jsou veškeré operace s položkami spravovány na jednom místě, což s sebou nese nesporné výhody. Ve třídě byly implementovány jak mechanismy pro samotné vyhledávání, tak i běžné operace nad položkami.

Třída dále dědí vlastnosti od třídy „Menu“, díky čemuž bylo velmi zjednodušeno generování navigace, nastavování aktivní skupiny podle procházené položky, nebo editované v administrativním rozhraní, a celkové ovládání hlavního menu obchodu. Třída „Menu“ dále dědí třídu „Connect“, a tím byl umožněn navržené třídě „Items“ přístup na databázi.

```
1  <?php
2  /** Class Items */
3  class Items extends Menu {
4      /** @var int $PolozkyID */
5      private $PolozkyID;
6      /** Proměnné pro vyhledávání */
7      private $stmt,$arrSearchWords,$canSearch,$word;
8      /** Objekt s daty pro editaci a výpis položky */
9      public $data_item; //objekt
10     /** Lze položky vypsát? */
11     public $vypis;
12     /** Proměnná pro výpis bloku HTML */
13     public $vystup_html;
14     public function __construct($strNameItem = null,
15                                 \Slim\Slim $SlimApp = null,$word = null){
16         /** Definice URL adresy položky */
17         private function pathItem($arrParams = []){}
18         /** Šablona pro tisk položek jako hlavní přehled aplikace */
19         private function itemsToShop(mysqli_result $result){}
20         /** Metoda vytvoří vyrovnací tabulku s ID nalezených položek */
21         private function temporaryTableOfSearchedItems(){}
22         /** Odstraní od produktu obrázek, popřípadě jej smaže z adresáře */
23         public function deleteImageFromItem($obrAssn){}
24         /** Uloží k produktu související položku */
25         public function saveRelatedItem($idRelatedItem){}
26         /** Smaže od produktu související položku */
27         public function deleteRelatedItem($idRelatedItem){}
28         /** Ukládá prosté vlastnosti produktu */
29         public function saveItem($nazev="Neuvedeno",$cena_bez_DPH=0,$sleva=0,
30                                 $DPH=2,$kod=NULL,$P_N=NULL,$zaruka=5,$vyrobce=0,
31                                 $popis_kratky="<p></p>",$ks=0,$tip=0,$novinka=1,
32                                 $vyprodej=0,$domacnosti=0,$male_stredni=0,
33                                 $velke=0,$bazar=0,$vaha_kg=0){}
34     /** Zobrazuje produkty v obchodu, dle příslušných parametrů */
35     public function loadAllItemsFromGroupOrSearch($news=0,$sale=0,
36                                                     $action=0,$bazaar=0,$min=1,$max=1000000000,
37                                                     $order='asc',$page=0,$items_on_page=12){}
```

```
31     /** Získá položky pro pravý panel, které jsou ve slevě */
32     public function itemInSale(){}
33     /** Generuje navigaci obchodu */
34     public function navigate(){}
35     /** Getter Polozky_ID */
36     public function getId(){}
37     /** Získa a zpracuje data pro zobrazení náledu položky v obchodu */
38     public function getItem(){}
39     /** Získá a zpracuje data pro zobrazení editované položky */
40     public function getEditableItem(){}
41     /** Generuje pole položek pro hlavní stránku obchodu*/
42     public function getActionNewsDefaultPageShop(){}
43     /** Metoda poskládá větu tak, aby bylo nalezeno nejvíce položek */
44     public function findBestStatement(){}
45     /** Metoda vyhledává položky včetně "P_N" a "Kod" */
46     public function findBestStatementForAdminItem(){}
47     /** Metoda generuje HTML výsledek vyhledávání, zobrazen jako roleta */
48     public function searchFormHelper(){}
49     /** Metoda generuje HTML nalezených souvisejících položek (admin) */
50     public function searchRelatedItems(){}
51     /** Odstraní instanci třídy Items */
52     public function close(){}
53 };
```

Zdrojový text 2 Třída „Items“ a její metody

V souvislosti s vyhledáváním třída implementuje několik metod, které tento problém postupně řeší.

První krok ve vyhledávání realizují metody „findBestStatement()“ a „findBestStatementForAdminItem()“, z názvů je patrné, že druhá metoda slouží pro hledání v administrativní části aplikace, první metoda je pak využívána pro vyhledávání v samotném obchodě. První metoda provádí hledání pouze v názvu a kompletním popisu položky, druhá metoda pak rozšiřuje vyhledávání o číslo produktu P/N, a také skladový kód.

Dále byl navržen rekurzivní algoritmus, který se snaží sestavit hledanou větu tak, aby bylo vyhledáno co největší množství položek. Systém sice nezohledňuje relevanci, ale uživatel může při vyhledávání slovo, popřípadě větu, blíže specifikovat, a tím lze dohledat požadovaný produkt, pokud jej však obchod obsahuje. Hledání nejvhodnější věty bylo provedeno heuristickým způsobem, kdy je vytvořeno několikanásobné vyhledání s postupnou záměnou pořadí slov ve větě. Pokud je vyhledáváno pouze jediné slovo, algoritmus provede hledání pouze jednou, jelikož není potřeba žádné pořadí slov zaměňovat. Pro vyhledávání byla s výhodou použita třída „mysqli_stmt“, která je součástí driveru „MySQLi“ [8]. Třída umožňuje zrychlit rekurzivní dotazy na databázi metodou vázaných proměnných, kde v konkrétním případě jsou dotazy rekurzivně zasílány uvedeným algoritmem. Systém vždy kontroluje, zda se již s utvořenou větou databáze nedotázal, pokud ano pokusí se opět vygenerovat takový slovosled, který ještě nepoužil, a až poté se dotáže. Tímto způsobem byl

minimalizován počet zasílaných dotazů, a tím zrychlen celý proces hledání. Pro snadnější pochopení tohoto problému zde byla zanesena nejnutnější část zdrojového kódu. Předpokladem je, že v proměnné „\$this->arrSearchWords“ se nalézají slova, kde každé slovo je jedním prvkem pole. Viz Zdrojový text 3.

```
1  <?php
2  if($this->stmt = Connect::$MySqlLi->prepare("... AND (CONVERT(Nazev USING
      utf8) LIKE ?) OR (CONVERT(Popis USING utf8) LIKE ?);")){
3      $foundedItems = 0;
4      $this->stmt->bind_param("ss", $word, $word);
5      $arrAlredyUsedWord = [];
6      $numberOfWords = count($this->arrSearchWords);
7
8      while(count($arrAlredyUsedWord) != $numberOfWords){
9          if($numberOfWords > 1) shuffle($this->arrSearchWords);
10         $search_word = implode("%", $this->arrSearchWords);
11         $word = "%$search_word%";
12         if(!in_array($word, $arrAlredyUsedWord))
13             $arrAlredyUsedWord[] = $word;
14         else
15             continue;
16         $this->stmt->execute();
17         $this->stmt->store_result();
18         if($foundedItems < $this->stmt->num_rows){
19             $foundedItems = $this->stmt->num_rows;
20             $best_word = $word;
21         }
22     }
23     if($word != $best_word){
24         $word = $best_word;
25         $this->stmt->execute();
26         $this->stmt->store_result();
27     }
28 }
```

Zdrojový text 3 Část metody „findBestStatement()“

Vyhledávání je case insensitive¹⁵, a ignoruje diakritiku. Díky tomuto přístupu je umožněno vyhledat více položek s menší restrikcí.

Výše zmíněné metody vytvoří instanci třídy „mysqli_stmt“, která nese data o vyhledaných položkách. Následuje použití dalších metod, v závislosti na zobrazení vyhledaných položek. Položky jsou vyhledány buďto formou roletového našeptávače na každé stránce aplikace, mimo administrativní část, dále je možno provádět vyhledávání na samostatné stránce, kde je umožněno aplikovat i rozšířené vyhledávání metodou volby filtru.

¹⁵ Case insensitive – není dbán důraz na velká a malá písmena

V souvislosti s roletovým našeptávačem je využívána metoda „searchFormHelper()“, která převezme instanci třídy „mysqli_stmt“ včetně nalezených dat a dále data formátuje do HTML, dojde k separaci prvních pěti výsledků, a pokud hledané slovo odpovídá více položkám, je zanesen odkaz na rozšířené vyhledávání. Takto vytvořené formátování je následně umístěno do řetězce, který je již zpracováván dalším skriptem umístěným „/xampp/htdocs/www/ajax/search.php“, kde dochází k jeho převodu do formátu JSON ke kterému jsou přidány další parametry pro zobrazení dat v aplikaci s využitím AJAX.

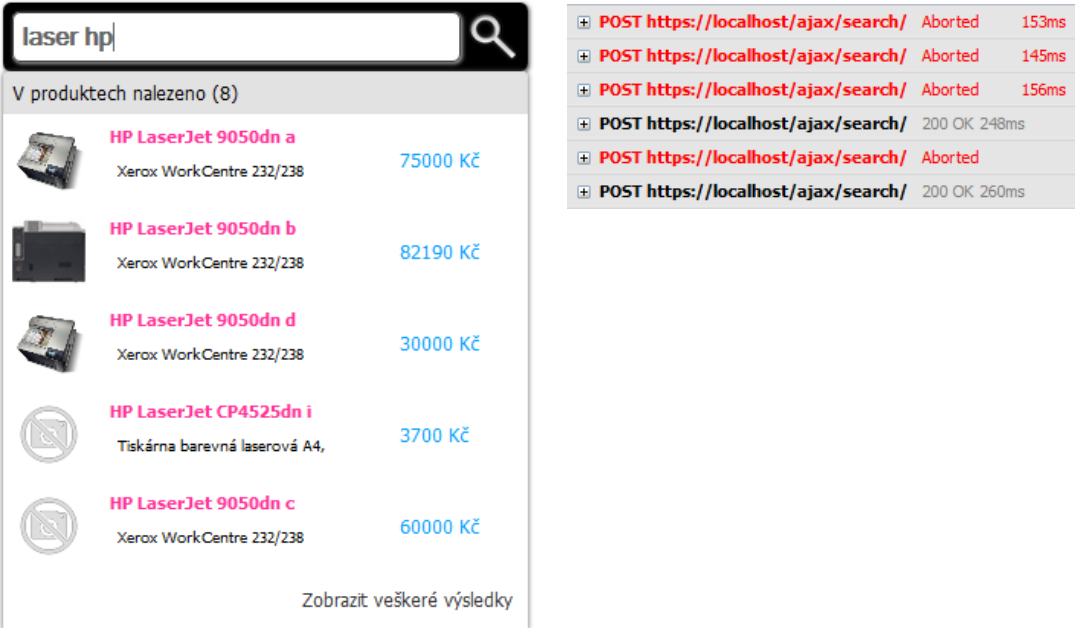
Pokud mají být výsledky vyhledávání zobrazeny v aplikaci na samostatné stránce kde je umožněno provádět filtraci, je po metodě „findBestStatement()“ volána metoda „loadAllItemsFromGroupOrSearch()“, která je primárně určena pro zobrazení nalezených výsledků formou, která byla uvedena na obrázku Obr. 2 Rozvržení širokého grafického návrhu. Metoda je využívána i pro běžné zobrazení produktů ze zvolené skupiny, ale byla použita i pro zobrazení vyhledaných výsledků. V tomto případě je z metody volána ještě metoda „temporaryTableOfSearchedItems()“, která provede separaci identifikátorů nalezených položek a utvoří *MySQL* temp tabulku „find“ s nalezenými identifikátory, následně jsou metodou „loadAllItemsFromGroupOrSearch()“ zaslány dotazy na databázi, kde dochází k detekci skutečnosti, že bylo provedeno vyhledávání, a z tabulky „polozky“ provede s využitím tabulky „find“ separaci nalezených produktů. Opět jsou data zpracována do textového řetězce, který je předáván pro další zpracování buďto klasických GET požadavků na zobrazení stránky obchodu, nebo s pomocí dalšího skriptu „/xampp/htdocs/www/ajax/echo_group.php“, který je používán pro filtrování položek s využitím technologie AJAX.

3.1.1.3 *Realizace na straně klienta*

Jak již bylo zmíněno v aplikaci lze provádět vyhledávání dvěma způsoby, nejprve zde bude osvětleno řešení roletového našeptávače, následně pak provádění rozšířeného vyhledávání s využitím filtrace výsledků.

Modul roletového našeptávače je dostupný ve všech stránkách prezentace, z tohoto důvodu musí být taktéž přístupný na všech stránkách i JS skript, který veškeré požadavky související s vyhledáváním touto metodou zpracovává. Jedná se o handler, který byl aplikován na vyhledávací pole v horní polovině grafického návrhu, konkrétně se jedná o 4. pozici na Obr. 1 Rozvržení úzkého grafického návrhu. Byl umístěn do souboru „/xampp/htdocs/www/jquery/jquery_user.js“. Z tohoto handleru dochází k volání funkce

„searchAjax(search_word)“, kterou je následně požadavek na vyhledávání vykonán. Vyhledávání je prováděno od 2 znaků, kdy dojde k zaslání požadavku. Pokud je zapsán znak mezery, je ignorován do dalšího zápisu jiného znaku než mezery. Systém byl taktéž ošetřen proti generování většího množství požadavku v případě zápisu věty, jelikož je funkce spouštěna vždy po nárůstu počtu znaků hledaného slova. Příklad vyhledávání a průběh požadavku na server lze pozorovat na obrázku Obr. 1.



The screenshot shows a search interface on the left and a network traffic log on the right. The search interface has a search bar with 'laser hp' and a magnifying glass icon. Below the search bar, it says 'V produktech nalezeno (8)'. There are five product listings visible:

- HP LaserJet 9050dn a, Xerox WorkCentre 232/238, 75000 Kč
- HP LaserJet 9050dn b, Xerox WorkCentre 232/238, 82190 Kč
- HP LaserJet 9050dn d, Xerox WorkCentre 232/238, 30000 Kč
- HP LaserJet CP4525dn i, Tiskárna barevná laserová A4, 3700 Kč
- HP LaserJet 9050dn c, Xerox WorkCentre 232/238, 60000 Kč

At the bottom of the search results, there is a link 'Zobrazit veškeré výsledky'. The network traffic log on the right shows a sequence of POST requests to 'https://localhost/ajax/search/':

Request	Status	Response Time
POST https://localhost/ajax/search/	Aborted	153ms
POST https://localhost/ajax/search/	Aborted	145ms
POST https://localhost/ajax/search/	Aborted	156ms
POST https://localhost/ajax/search/	200 OK	248ms
POST https://localhost/ajax/search/	Aborted	
POST https://localhost/ajax/search/	200 OK	260ms

Obr. 11 – Roletový vyhledávací našeptávač a zasílané požadavky na server

Pokud si uživatel bude chtít prohlédnout další nalezené výsledky, popřípadě si položky individuálně filtrovat, může použít rozšířené vyhledávání, které je přístupné po zadání výrazu a kliknutí tlačítka myši na lupu ve vyhledávacím poli, popřípadě pokud je nalezeno více jak pět položek, může využít odkazu na konci našeptávače „Zobrazit veškeré výsledky“. Následně je odkázán na aplikaci obchodu, kde má k dispozici veškeré vyhledané výsledky, a může provádět filtraci dle specifikovaných parametrů filtru.

V této části je JS používán právě pro účely filtrace, kdy je na server zasílán požadavek pro získání dat, které odpovídají požadované filtraci. O veškerou činnost spojenou s filtrací se starají funkce a handlers v souboru „/xampp/htdocs/www/jquery/shop.js“. Zde je problém řešen více funkcemi současně, jelikož bylo potřeba vyřešit změnu URL adresy při změně filtru, dále se používá doplněk *jQuery UI* [60], s jehož pomocí byl řešen horizontální po-

suvník, kterým lze omezit cenové rozpětí zobrazovaných produktů, formátování české měny u posuvníku, samotné zaslání požadavků na server a podobně.

3.2 Optimalizace pro vyhledávače

Někdy taktéž používaná zkratka SEO. Je soubor omezení a příkazů pro internetové aplikace upravující strukturu dokumentů tak, aby byly pro roboty vyhledávacích společností co nejlépe pochopeny, a umístili tak aplikaci do předních pozic ve vyhledávání.

Obecně chybnou snahou je zobrazit vlastní aplikaci na co nejvyšší pozici s nejvyhledávanějšími slovy. Např. je chtěno být na prvních místech, když potenciálním zákazníkem je vyhledáváno slovo „televize“. Tento cíl je však nesprávný! Kdo vyhledává televize, může hledat široké spektrum produktů nebo služeb, které ani zákazníkovi e-shop nemůže nabídnout – např. plasmové televize, 3D televize, novinky o televizích, jak vznikla televize, nebo televizní program. S největší pravděpodobností se získá vyšší návštěvnost, ale přivedené návštěvníky nezaujmou, a tedy odejdou jinam [13].

Správným cílem SEO by mělo být oslovení těch návštěvníků, ze kterých se mohou stát zákazníci. Vysoká návštěvnost tedy není hlavním cílem. Snahou je především cílená návštěvnost. Návštěvníkům se musí prezentovat přesně ty informace, které hledají [13].

3.2.1 Vybrané prvky SEO

Principu optimalizace je více, některé jsou v souladu s etikou, jiné se snaží podvádět roboty a tím vylepšovat své pozice. V této aplikaci byly použity pouze etické principy. V následujících kapitolách budou popsány jednotlivé části SEO a míra jejich plnění, která byla do aplikace zanesena.

3.2.1.1 *Kvalitní a unikátní obsah*

Stránka by měla mít kvalitní a unikátní obsah, pravidelně (v ideálním případě denně) aktualizovaný. [15][14]

Jelikož hlavní stranou byla zvolena úvodní stránka obchodu, kde je především požadovaný proměnlivý obsah, bylo nutné zajistit jeho aktualizaci. Tohoto bylo docíleno tím, že obchod při každé návštěvě generuje několik položek, které jsou označeny jako novinky, slevy a výprodeje, umístěné v příslušných oddílech. Pokaždé co robot navštíví e-shop, obdrží různý obsah.

3.2.1.2 *Používání HTML značek podle normovaných předpisů*

Tvůrcem stránky by měl být používán korektní sémantický i syntaktický zápis značek pro daný jazyk HTML či XHTML – například „h1“ pro nadpis první úrovně, „em“ pro zdůraznění, „strong“ pro silné zdůraznění a podobně. Vyhledávače takovému textu přiřkládají větší význam. Jestliže ale webový tvůrce například definuje nadpis pomocí velikosti písma (font-size), ne vždy jsou roboty stránky aplikace správně roboty pochopeny, tím nedochází ke zjištění důležitosti nadpisu. [15][51]

Dnešní vyhledávací roboti jsou natolik chytrí, že stahují i CSS styly a kontrolují skutečnou nastavenou velikost nadpisů, a podobně [14]. Jestliže je zjištěna snaha nadpis skrýt, nebo je nastavena taková velikost, která by byla v konkrétním případě nesmyslná, tuto skutečnost systémy detekují, hrozí následná penalizace webu za chybnou interpretaci, tedy propad ve vyhledávačích.

Navržená aplikace byla podrobena validátoru W3C [10], kde byly jednotlivé stránky validovány s doctype¹⁶ HTML5. Veškeré stránky byly dle experimentálního validátoru shledány validními. Dále byly v aplikaci použity nadpisy, které mají vhodně upravený vzhled, tak aby splňovaly výše uvedené podmínky. Jednotlivé dokumenty byly upraveny tak, aby v posloupnosti existovaly nadpisy korektních úrovní.

3.2.1.3 *Používání titulku, nadpisů a popisů*

Tvůrce by měl uvádět v titulku objektů (oddíly stránky, obrázky, tabulky a další) konkrétní popis. Například obsahuje-li titulek stránky namísto obecných výrazů typu *Úvodní stránka* klíčová slova, získává stránka další významné plus. [15][51]

V aplikaci byly důležité elementy, které by měly vyhledávače vhodně indexovat, opatřeny těmito parametry. Nejvyšší důraz byl zaměřen na titulek webu, který v případě obchodu obsahuje vždy název procházené skupiny zboží, nebo název zobrazovaného produktu. Hlavní titulek má totiž obrovský význam v indexaci webu. Některé parametry tohoto bodu jsou sdruženy v kapitole 3.2.1.2 Používání HTML značek podle normovaných předpisů.

¹⁶ Doctype – jedná se o uvedení typu dokumentu, aby stránky aplikace byly správně zobrazeny, případně podrobena náležitěmu validátoru

3.2.1.4 *Krátká a neměnná URL adresa*

Uvedení krátké URL adresy podpoří zájem ostatních uživatelů o odkazování na stránku. Příliš dlouhá a lidem nesrozumitelná URL adresa naopak od odkazování spíše odrazuje. Přítomnost klíčového slova v URL adrese může u některých vyhledávacích strojů zvýšit umístění stránky ve výsledcích hledání těchto slov. Naopak se nedoporučuje používat v URL parametr id. Požadavek na neměnnost adres vyplývá z fungování vyhledávacích strojů i z požadavku na budování zpětných odkazů. [15][51]

a) Špatná adresa z hlediska SEO

<http://example.cz/katalog/nabytek/kresla/sede-kreslo-skladaci>

Adresa položky obsahuje strukturu katalogu zboží (nábytek, křesla), při reorganizaci katalogu se URL může změnit.

b) Špatná adresa z hlediska SEO

<http://example.cz/katalog?id=432&what=B603AA60CC0A03EA0EB&kat=jjdd>

V tomto případě URL adresa obsahuje parametr id, a pro člověka je nesrozumitelná.

c) Správná adresa z hlediska SEO

<http://example.cz/katalog/sede-kreslo-skladaci>

Adresa položky obsahuje relevantní klíčová slova, přitom není náchylná ke změnám.

URL adresy uvedené v bodu c) bývají označovány jako přátelské adresy (friendly URL). Obecně je žádoucí, aby byla adresa i v případě bodu c), ještě kratší. Na čím nižší úroveň aplikace je odkazováno, tím je dosahováno lepších výsledků. Systém byl navržen tak, aby splňoval přesně tyto požadavky. Více o tomto bodu bylo popsáno v kapitole 3.2.2 Přátelské URL odkazy.

3.2.1.5 *Kanonizační problémy*

Problémem jsou chápány stránky, které obsahují duplicitní obsah, ale je na ně odkazováno z rozdílných URL adres. Typickým příkladem je „<http://example.com>“ a „<http://www.example.com>“. Za tyto duplicity penalizace ve větším rozsahu nehrozí, avšak stránky jsou hodnoceny hůře. Pokud takové situace nastanou, mělo být v takových případech provedeno přesměrování na jedinou stránku s kódem 301, případně 302. Viz stavové kódy HTML zdroj [10].

I v aplikaci se tyto duplicity nacházejí, ale v těchto situacích je prováděno právě přesměrování ve většině případů s kódem 301 (Moved permanently), nebo kódem 303 (See Other).

V aplikaci byla aplikována promyšlená logika směrování na jisté zdroje, a to oběma metodami GET i POST, zároveň bylo ošetřeno AJAX volání.

3.2.1.6 *Budování zpětných odkazů*

Stránky, na které je odkazováno z jiných zdrojů, získávají výhodnější hodnocení.

O budování zpětných odkazů se musí postarat každý správce aplikace individuálně. Zde by se daly taktéž zmínit reklamy, jež v této činnosti mají výhradní postavení. Pak by se již jednalo o kombinaci praktik SEO a SEM¹⁷ [51]. Bohužel existují i praktiky, které toto budování provádějí neetickou formou, například umístěním odkazů do různých diskusí a podobně. U těchto praktik hrozí penalizace vyhledávačů, tedy je na jednání administrátora, jak se k tomuto problému postaví.

3.2.1.7 *Korektní používání souboru robots.txt*

Jedná se o textový soubor, který musí být umístěn v kořenovém adresáři webu. Tento soubor upravuje činnost robotů vyhledávačů a staví jim do cesty pomyslné mantinely. V praxi si vyhledávače přečtou tento soubor, dle jeho obsahu zjistí odkazy, které robot nemá indexovat. Bohužel praxe je taková, že někteří roboti tento soubor ignorují [51]. Pro aplikaci, ale i jiné systémy je důležité, že tento soubor užívají důležité vyhledávače.

V aplikaci bylo díky této možnosti robotům zakázáno přistupovat na některé adresy, jelikož neobsahují lukrativní obsah, jenž by byl zapotřebí indexovat.

S problematikou těchto robotů souvisí i nekalé praktiky, kdy jsou konstruováni roboti pro získávání například e-mail adres, obecně zajímavých informací a podobně. Bylo by vhodné zakázat přístup všem robotům mimo „googlebot“ a „seznambot“, kteří mají výhradní postavení na české síti. Problémem je fakt, že mnoho robotů nemá toliko slušnosti a přistupuje na aplikaci jakoby z klasického prohlížeče. V tomto případě by bylo možno přístup zakázat pouze formou „blacklist“ ověřování. Vývoj je natolik turbulentní, že ve skutečnosti

¹⁷ SEM – marketingové praktiky sloužící pro zviditelnění aplikace. [51]

se všem nepotřebným robotům zakázat přístup nedá. Z tohoto důvodu nebyly podniknuty kroky pro zmezení přístupu nepotřebným robotům.

3.2.1.8 *Používání description, keywords*

Klíčová slova v meta-tag description nemají vliv na hodnocení webové stránky. Obsah meta-tag *description* je použit u popisu stránky ve výsledcích vyhledávání zejména na Google, pokud tedy vyhledávač nenalezne na stránce vhodnější text (platí zejména pro long-tailové výrazy). Meta-tag description se také zobrazuje jako popis webové stránky při sdílení na sociálních sítích. Hlavním cílem meta-tag description je stručně popsat obsah webové stránky a motivovat uživatele k návštěvě [15].

Meta-tag keywords používá k zjišťování relevance z hlavních vyhledávačů v současné době pouze Bing.

E-shop používá meta-tag description pro všechny indexované stránky stejný. Roboti se snaží vyhledat vhodnější text, a z tohoto důvodu byl také ponechán jednotný popis.

3.2.1.9 *Aktivity na sociálních sítích*

Zejména společnost *Google* klade vyšší důraz na šíření obsahu v rámci sociálních sítí. Pokud je obsah sdílen a komentován v této sféře, dochází k tvorbě vyššího indikátoru kvality, než počet zpětných odkazů. Sociální síť *Google Plus* (G+) přímo ovlivňuje pozice ve vyhledávání ve vyhledávači *Google*. Přihlášený uživatel dostává personalizované výsledky, do kterých se promítá doporučení jeho přátel. [15]

Sociální sítě zásadně ovlivňují postavení webu, více o využití tohoto bodu bylo popsáno v kapitole 3.2.3 Open Graph.

3.2.2 **Přátelské URL odkazy**

Pod tímto pojmem je myšlena lidem srozumitelná URL adresa. Důvodů proč používat tyto odkazy je několik. Z hlediska SEO se jedná o další klíčová slova zobrazená v adrese stránky. Názory na tuto tematiku se sice různí, ale přátelské URL každopádně přispívá k lepšímu umístění ve vyhledávačích, i když povětšinou malým procentem, bohužel na tomto principu bylo SEO postaveno. Důležitějšími faktory přátelských adres jsou uživatelská přívětivost a také se jedná o prvek zabezpečení stránek, protože nelze jednoznačně pochopit strukturu celé aplikace.

3.2.2.1 *Technické řešení*

Vhodné postavení systému tak, aby korektně fungoval přepis na přátelské adresy, by bylo velmi komplikované. Naneštěstí existuje několik mikro frameworků¹⁸, které se zabývají výhradně tímto problémem.

Pro aplikaci byl zvolen mikro framework *Slim v2.4.3* [18]. Díky tomuto výkonnému a spolehlivému modulu se můžou provádět veškeré potřebné změny, které s touto problematikou souvisí.

Pro funkčnost rozšíření bylo potřebné nastavit přesměrování všech požadavků na neexistující soubor, nebo složku na soubor, kde je spuštěno jádro aplikace se směrovacími pravidly. Toto nastavení vyžaduje aktivovaný modul „`mod_rewrite`“ na serveru [17]. Bohužel toto nastavení se v dokumentaci nenalézá, avšak musel být upraven soubor „`.htaccess`“ v kořenovém adresáři aplikace. Viz Zdrojový text 4.

```
1 RewriteEngine On
2 RewriteBase /
3
4 RewriteCond %{REQUEST_FILENAME} !-f
5 RewriteCond %{REQUEST_FILENAME} !-d
6 RewriteRule ^(.*)$ index.php [QSA,L]
```

Zdrojový text 4 – Konfigurace „`.htaccess`“ pro použití *Slim v2.4.3*

Stručně vysvětleno po řádcích, provede se zapnutí jádra „`rewrite`“, nastaví se kořen serveru, pokud se nejedná o složku, anebo o korektní soubor, přesměrují se veškeré požadavky URL adresy na „`index.php`“. V souboru se nachází více předpisů, zde byl uveden pouze nutný obsah.

V souboru „`index.php`“, kde je spuštěn engine *Slim v2.4.3*, bylo nutné utvořit soupis pravidel, která se musí provést při příchodu požadavku s určitým formátem. Popis je obsažen v dokumentaci [18].

V celé aplikaci je pak toto rozšíření přístupné, díky kterému jsou vyvolávány další požadavky na přesměrování, generování chyby 404 – Nenalezeno, a podobně.

¹⁸ Framework – programový modul aplikace, který vykonává její část a podporuje tak její chod

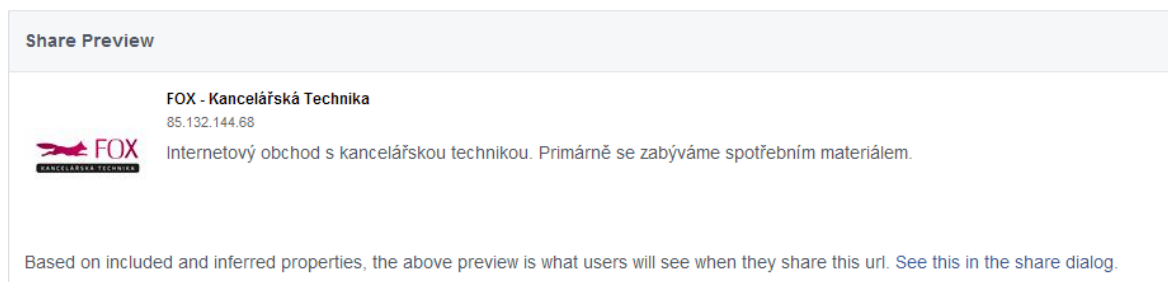
3.2.3 Open Graph

Jedná se o protokol, kterým lze i z poměrně obyčejných stránek vytvořit bohaté objekty v sociálním grafu. Obohacuje stránku o několik meta-tagů, které vyjadřují postavení aplikace v sociálním grafu. Jedná se o další krok k sémantické aplikaci. [19]

Vytvořená aplikace se snaží částečně implementovat tento protokol tak, aby zůstaly jednotlivé stránky validní a současně v souladu s tímto protokolem, tudíž nebyla upravována hlavička DOCTYPE, jak uvádějí některé zdroje. V nastavení aplikace se dá součást povolit, nebo zakázat. Dále je možno tuto část plně personalizovat.

Implementace byla provedena vložením několika meta-tagů s příslušnou hodnotou atributu „property“. Díky těmto vlastnostem bylo známé sociální síti *Facebook* sděleno, o jaký druh zdroje se jedná, dále pak bylo uvedeno několik atributů specifikujících vytvořenou aplikaci. Po implementaci bylo provedeno ladění, které samotná sociální síť poskytuje.

Její výsledek lze pozorovat na obrázku Obr. 12.



Obr. 12 Výsledek implementace *Open Graph* protokolu [19]

Z výsledku je patrná specifikace profilu firmy, kde jsou uvedeny póly její působnosti a nacionály. Takto vytvořený profil bude následně na sociální síti prezentován. Krok umožnil především zobrazit logo aplikace, v opačném případě by byl sociální síti nalezen první vhodný ze zdroje aplikace, což není vhodné.

4 ZABEZPEČENÍ APLIKACE

V průběhu let si čím dál tím větší množství vývojářů začíná uvědomovat bezpečnostní rizika spojená s umístěním aplikaci v síti internet. Poslední studie *SANS institute* z února roku 2014 uvádí, že se neustále zvyšuje počet uživatelů, kteří nechávají testovat své aplikace a dbají tak na jejich zabezpečení [37]. Ve srovnání s průzkumem z roku 2012, jenž prováděla organizace *Ponemon institute* došlo k poměrně velké změně, kdy tehdy mnoho vývojářů nemělo žádné povědomí o existenci nějakých projektů, které se snaží vychovávat bezpečnosti znalé vývojáře a testery. [38]

Neustále se lze setkat s lidmi, kteří vyvíjejí aplikace, ale o možnostech jejich zabezpečení nemají mnoho povědomí, výsledkem jejich práce jsou aplikace, které jsou pro ostrý provoz nevhodné, a brzy tak zanikají. Vývojáři nereagují dostatečně brzy na podmínky útočníku, mnohdy ani nevyhledávají příčinu problému prolomení systému.

Jedním z projektů, který se snaží vychovávat bezpečnosti znalé vývojáře a testery, je komunita *The Open Web Application Security Project* (dále jen *OWASP*). Jedná se o nezávislou neziskovou organizaci, která se zabývá bezpečností webových aplikací s využitím informací od početné komunity uživatelů. Cílem této komunity je poskytnout všem uživatelům informace potřebné k tomu, aby mohli vyvíjet, pořizovat či spravovat aplikace, kterým lze z bezpečnostního hlediska věřit. [20]

Veškeré dokumenty související s možnostmi a principy zabezpečení jsou poskytovány projektem zdarma, a tím je umožněno široké veřejnosti využít poznatky pro svá řešení. Velmi přínosnou vlastností této komunity je fakt, že organizace neklade důraz pouze na technologické aspekty, nýbrž i na jistou uživatelskou přívětivost, tedy se snaží zajistit maximální bezpečnost s co nejmenším omezením uživatele vytvořeného systému. [20]

Organizace *OWASP* za dobu svého působení publikovala několik významných prací. Navržený systém operuje s citlivými daty klientů, a také obsahuje statistické údaje týkající se prodeje, které nesmí být vynášeny mimo tento systém.

V této práci byly využity projekty *OWASP Application Security Verification Standard Project 2013 Beta* (dále jen *ASVS*) [59], a také *OWASP Top Ten – 2013*. [20]

V průběhu návrhu byla využita práce *Lukáše Zemka - Bezpečnost webových aplikací* [39], kde se detailně věnuje zabezpečení webových aplikací s využitím předchozího standardu

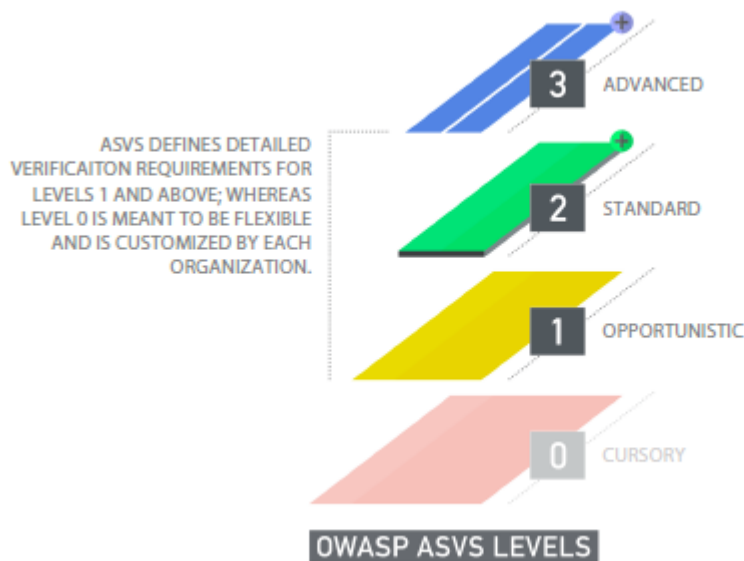
OWASP Top Ten – 2010 [20]. Práce je velmi dobře strukturovaná a poskytla autorovi velice dobré podklady zabezpečení navrženého systému.

4.1 OWASP ASVS

Tento projekt je souborem poznatků, které mohou být použity k ověření míry důvěryhodnosti v oblasti webových aplikací. Soubor poznatků byl rozdělen do čtyř úrovní, kde každá úroveň upravuje stupeň zabezpečení aplikace tak, aby bylo možné posoudit její efektivitu.

4.1.1 Úrovně ASVS

ASVS definuje čtyři úrovně zabezpečení (0 až 3). Čím vyšší číslo úrovně, tím jsou požadavky na bezpečnost vyšší. Aby mohla aplikace získat vyšší úroveň zabezpečení, musí obsahovat požadavky všech nižších úrovní. Úroveň vždy zahrnuje požadavky úrovní nižších. [59]



Obr. 13 Úrovně OWASP ASVS [59]

4.1.1.1 Úroveň 0: Cursory

Na této úrovni nejsou organizací definovány žádné požadavky. Prostor byl organizací vyhrazen pro sestavení individuálních požadavků veřejnosti, která systém vyvíjí. [59]

4.1.1.2 *Úroveň 1: Opportunistic*

Tato úroveň definuje základní zdanitelnost navrženého systému. Úpravy, které jsou v této úrovni ustanoveny, je velmi snadné aplikovat. Pokud je navržený systém upraven dle těchto postupů, tak splňuje základní požadavky na použití bezpečnostních prvků. [59]

4.1.1.3 *Úroveň 2: Standard*

Definuje požadavky na ověření. Úroveň je doporučována pro webové aplikace, kde se vyskytují citlivá data. Ustanovená úroveň bývá využita nejčastěji, jelikož pro většinu současných aplikací je dostačující. [59]

4.1.1.4 *Úroveň 3: Advanced*

Zde jsou ustanoveny požadavky na aplikace, jejichž narušení by mělo za následek ztrátu života. Systémy upravené dle této úrovně jsou chráněny vůči všem zranitelnostem. V této části jsou ustanoveny principy implementace jednotlivých bezpečnostních prvků. [21]

Útoky na takto zabezpečené aplikace provádí profesionálové, jelikož obejít zabezpečení systémů je velmi složitá operace. [59]

4.1.2 **Detaily požadavků ASVS**

Organizací je prováděna výchova vývojářů a testerů formou rozložení aplikačních problémů do samostatných bloků, kde vymezený rozsah bloků slouží pro zabezpečení aplikace na požadované úrovni.

4.1.2.1 *V1: Požadavky na ověření autentizace*

Tyto požadavky byly rozděleny organizací do celkem 22 bodů. Konkrétně pak body **V1.1** až **V1.7** jsou nutné pro splnění první úrovně zabezpečení, body **V1.1** až **V1.20** jsou nutné pro splnění požadavku druhé úrovně a veškeré body jsou potřebné pro splnění požadavku úrovně třetí. [59]

Jako prvním je bod **V1.1**. Zde je požadováno, aby aplikace měla jednoznačně vymezené mantinely, kde má uživatel oprávnění vstoupit. Uvádí se, že skript kde uživatel nemá příslušná oprávnění pro vstup, musí být bezpodmínečně ukončen. [21]

Druhým bodem **V1.2** je upraven mechanismus zobrazování a upravování hesel. V praxi se jedná o zamezení automatického vyplňování formulářového pole hesla a jeho vizuální schování. [22]

Třetí bod **V1.3** upravuje pravidla, která souvisí s autentizací obsahu. Pokud dojde k selhání, nebo například vypršení platnosti autentizačních prvků, musí být bezpodmínečně provedeny takové kroky, aby se útočník nemohl přihlásit. Viz [23]

Čtvrtý bod **V1.4** se zabývá myšlenkou zabezpečení dat, a to jak z pohledu úložiště, tak z pohledu přenosu dat. Pro splnění toho bodu je nutné využít protokolu HTTPS, který využívá SSL/TSL. Tento protokol šifruje komunikaci mezi klientem a serverem, čímž je znemožněno odposlouchávání, dále jednoznačně identifikuje server, tedy je jistota, že klient skutečně komunikuje s dotazovaným serverem, a není podstrčen jiný obsah. [24]

Pátý bod **V1.5** upravuje podmínky pro obnovení uživatelského hesla. [21]

Šestý bod **V1.6** zkoumá operace s hesly uživatelů, konkrétně při přihlašování, resetu a obnovení. Důraz je kladen na jistou distanci od zobrazování podrobnějších informací o procesu autentizace. Konkrétně při chybném přihlášení by neměla být zobrazena informace „uživatelské jméno je chybné“, ale například „autentizace se nezdařila“. Tím je znemožněno určit v které části byla chyba uvedena. [26]

Dalším bodem je **V1.7**. Tato část zahrnuje poznatky související se situací nasazení aplikace do ostrého provozu. Uvádí potřebnou úpravu systému tak, aby neobsahoval ladící informace, hesla, uživatele, objednávky a podobně. [27]

4.1.2.2 **V2: Požadavky na ověření správy relací**

Tyto požadavky byly organizací rozděleny celkem do 15 bodů. Konkrétně pak body **V2.1** až **V2.8** jsou nutné pro zajištění úrovně první. Dále body **V2.1** až **V2.14** jsou potřebné pro splnění požadavků druhé úrovně. Pro třetí úroveň jsou stanoveny všechny předchozí body, ke kterým je přidán poslední bod **V2.8**.

Bod **V2.1** zdůrazňuje nutnost pro správu relací (session) využívat interní funkce programovacího jazyka. Programátor by neměl využívat vlastní funkce. [28]

Bod **V2.2** uvádí nutnost po odhlášení zneplatnit jeho sezení (session). Tato činnost musí být vykonána interní funkcí daného skriptovacího jazyka. Metodika se však může různit v závislosti na složitosti aplikace. [28]

Bod **V2.3** upravuje nutnost sestavit systém tak, aby byl uživatel automaticky odhlášen v případě, že by byla tato činnost opomenuta. [28]

Další bod **V2.4** stanovuje umístění odhlašovacího tlačítka na každé stránce, kde je požadována autorizace.[28]

Pátý bod **V2.5** se zabývá problematikou „Session Hijacking“, jedná se o krádež identifikátoru relace (dále jen SID). Pokud by byl SID odcizen útočníkem a vložen jako cookie do prohlížeče, popřípadě zapsán do URL adresy, mohla by tato osoba vystupovat pod identitou přihlášené osoby. Požadavky uvádějí nutnost SID nezahrnovat do URL adresy, ale do cookies. [2]

Šestý bod **V2.6** je taktéž spojen s operací odhlášení. Vyžaduje řádné vynulování, popřípadě obnovení SID. Činnost je nutná z důvodu známého útoku Session Fixation, kdy je útočníkem před přihlášením uživateli podstrčen jiný SID. Díky tomu že k SID je vázáno přihlášení, bude po provedení autentizace přihlášen jak uživatel, tak i útočník. [29]

Sedmý bod **V2.7** navazuje na bod **V2.6**, ve kterém je stanovena jedna z technik předejití krádeži, a to využíváním příznaku „HttpOnly“, kdy je identifikátor relace přístupný pouze prostřednictvím protokolu HTTP, a nelze s ním operovat v rámci JS. [28]

Následujícím osmým a posledním bodem je požadavek **V2.8**, který stanovuje nutnost používat takzvaný „secure flag“. Tento bod je možné splnit pouze s využitím protokolu HTTPS, kdy jsou cookies¹⁹ s tímto příznakem přenášeny šifrovaným spojením. Taková aplikace musí disponovat bezpečnostním certifikátem k serveru. [30]

Dalším bodem je upravována nutnost po přihlášení provést regeneraci relace, kdy je vygenerován nový SID a zaměněn za původní. Tento krok upravuje bod **V2.9**, jehož provedení je preventivním opatřením proti známému útoku „Session Fixation“, který byl popsán u bodu **V2.6**. [21]

Desátým bodem **V2.10** je upravována doba, kdy by měla být regenerace relace prováděna. Dle organizace je nutností provádět regeneraci vždy, když je prováděn proces re-autentizace. [56]

Dalším bodem **V2.11** je vyžadováno, aby veškeré generované SID byly aplikací ověřovány, zda jsou validní.

¹⁹ Cookie – malé množství dat uložené v prohlížeči klienta, může být zasláno serverem, nebo uloženo JS

Bod **V2.12** stanovuje nutnost ověřit, zda generované SID mají dostatečnou délku a jsou pseudonáhodně generovány [59]. Celkově jestli mají předpoklad odolat útoku.

Další bod v pořadí je **V2.13**. Tento bod upravuje nastavování cesty, pro kterou je relace vytvořena. Dále sděluje nutnost využít i jiný systém potvrzování, než je identifikátor SID. [59]

Čtrnáctým bodem a bodem posledním pro druhou úroveň v této oblasti, je bod **V2.14**. Tento bod vyžaduje ověřování, zda se na jediné relaci nenachází více zařízení [21]. V praxi je nutné ověřit veškeré dostupné prostředky, jako je IP adresa, identifikátor prohlížeče, a podobně.

Absolutně posledním bodem, který je vyžadován pro nejvyšší úroveň zabezpečení, je **V2.15** [21]. Zde je vyžadováno zavést absolutní dobu, která je vymezená pro operace přihlášeného uživatele, bez ohledu na jeho aktivitu.

4.1.2.3 **V3: Požadavky na Access Control List**

Pro body **V3.1** až **V3.4** platí jeden termín, a to Access Control List (ACL). Jedná se o centrální soupis oprávnění. Tento soupis upravuje přístupy ke konkrétním částem aplikace, konkrétním uživatelům. Technicky je ACL realizováno způsobem, kdy před každým vstupem do jistého objektu se provede dotaz na databázi, zjistí se, zda má uživatel potřebná oprávnění pro užití objektu, spuštění funkce, nebo zobrazení stránky a následně je provedena akce dle výrazu podmínky. [31]

Bod **V3.5** deklaruje zabezpečení webové aplikace v rámci adresářové struktury. Nutností je systém zabezpečit natolik, aby nebylo možné procházet umístění souboru mimo vyhrazený prostor. Cílem je zamezit potenciálnímu útočníku možnost získat povědomí o struktuře, prostředí a uživatelích aplikace. S touto problematikou korespondují i přátelské URL adresy, které se snaží adresářovou strukturu skrýt, tato problematika byla popsána v kapitole 3.2.2 Přátelské URL odkazy. [30][18]

V bodech **V3.4** a **V3.6** se objevuje slovo „Zranitelnost“ (Insecure Direct Object References). Tato část byla taktéž projektem OWASP zanesena do žebříčku *OWASP Top 10 – 2013*, konkrétně na čtvrtou pozici nejběžnějších útoků. [20]

Zranitelnost byla definována projektem OWASP jakožto:

“A direct object reference occurs when a developer exposes a reference to an internal

implementation object, such as a file, directory, database record, or key, as a URL or form parameter.” [32]

Přeloženo autorem následně: Aplikace je zranitelná v části, kde je odhalen odkaz na vnitřní implementaci objektu jako souboru, adresáře, databázového záznamu, nebo klíče, jako URL nebo parametr GET.

Jestliže útočník změní hodnotu parametru odkazu na objekt, pro který nemá dostatečné oprávnění, a není vytvořen systém s další autentizací, může takto získat přístup na zakázaný prostor. Tento problém pokrývá ACL.

Dalším bodem je **V3.7**, zde je opět aplikováno pravidlo „Výchozí zakázání přístupu“ (Deny access by default). Přeneseně tento bod upravuje pravidlo, pokud byla navštívena právě tato část, musí aplikace implicitně odebrat přístup. [23]

4.1.2.4 **V4: Požadavky na validaci vstupů**

Tato část upravuje veškeré vstupy nejen do aplikace, ale taky předávání dat mezi jednotlivými vrstvy celého systému. Došlo k vytvoření celkem 15 bodů. Body **V4.1** až **V4.8** jsou potřebné pro splnění první úrovně zabezpečení. Následujícím splněním bodů **V4.9** až **V4.12**, včetně všech předešlých, je upravována druhá úroveň a zbývající body **V4.13** až **V4.15** jsou vyžadovány pro třetí úroveň, samozřejmě včetně všech předešlých.

Bod **V4.1** je vztažen k příslušnému programovacímu jazyku, ve kterém je systém vytvořen. Tímto bodem se důkladněji zabývá *OWASP Testing Guide*. [33]

Bod **V4.2** souvisí s útokem „SQL injection“. Tento typ útoku se stále nachází na první pozici žebříčku nejběžnějších útoků, a obhájil pozici i v nejnovější žebříčku *OWASP Top 10 – 2013*. [20]

Útok „SQL injection“ spočívá v úpravě SQL dotazu tak, že umožní útočnickovi manipulovat s daty v databázi, včetně jejich zobrazení a zneužití. Primárně je využíváno formulářových polí, kde je zanesen škodlivý kód, který v případě neošetření způsobí získání neautentizovaných dat. [6]

Dále tento bod ustanovuje možnosti eliminace tohoto zákeřného útoku. Uvádí se možnosti spojené s ORM třídami a nástroji, popřípadě Doctrine. Další možností je využít SQL dotazy s vázanými proměnnými, které nejsou konstruovány skládáním řetězců, ale princip je

založen na prvotní přípravě dotazu, následně svázáním proměnných s připraveným dotazem za specifikace typu proměnné, a v poslední řadě vykonáním. [34][8]

Bod **V4.3** definuje zranitelnost z pohledu „Cross Site Scripting” (dále jen XSS). Jedná se o útok, kdy je útočníkem do aplikace zanesen cizí skript, jenž je v případě nesprávného ošetření vstupu interpretován prohlížečem. Takto napadeným aplikacím může být měněn vzhled, zamezeno odeslání formulářů a podobně. Nejohroženější skupinou aplikací jsou takové projekty, kde se nacházejí formuláře, nejvíce pak různé diskuse a blogy. [20]

S výhodou lze proti tomuto útoku zabezpečit pouze výstup. Důvodem je interpretace HTML pouze na straně klienta, tedy server nijak neohrozí. Zabezpečení spočívá v nahrazení HTML značek entitami, například namísto značky „<p>“ je vytisknuto „<p>“.

Bodu **V4.4** je věnována problematika „LDAP injection“. Jedná se o adresářovou informační službu, která obvykle uchovává údaje o uživateli. Zranitelnosti lze zneužít podsunutím metaznaků na uživatelském vstupu. Metoda spočívá opět v podsunutí cizího kódu, který je následně spuštěn. Po úspěšném útoku je možno přidávat nebo měnit objekty uvnitř LDAP struktury. Problém lze vyřešit takzvanou whitelist validací, která definuje vzorec vstupních dat, jenž mohou projít validací. [35]

Bod **V4.5** se zaměřuje na útoky typu „OS Command Injection“. Tato metodika je závislá na použití operačního systému serveru, a instalovaného skriptovacího jazyka. V rámci php jsou dostupné funkce „shell_exec()“, „exec()“, „passthru()“ a jiné, veškeré tyto funkce využívají příkazový řádek. S využitím metaznaků je možné upravovat příkazy příkazového řádku, a pokud nebude provedeno opatření pro tyto funkce, může ve výsledku útočník ovládnout celý server. Bod taktéž zmiňuje i útok „Code Injection“, díky němuž lze do zdrojového kódu přidat cizí kód. V minulosti byla zneužívána funkce „eval()“. [36]

Následující bod **V4.6** znázorňuje nutnost validace veškerých vstupních dat, je zde kladen vyšší důraz na ověřování dat způsobem whitelist, než blacklist. Konkrétně je tedy doporučeno spíše vymezit povolený formát dat, než utvářet validaci postavenou na zakazování určitého formátu dat. [20][59]

Pro splnění bodu **V4.7** je vyžadováno, aby validace vstupních dat byla prováděna na straně serveru. Zabezpečení na straně klienta lze jednoduše vypnout a je používáno spíše jako doplněk celé aplikace. [20][59]

Následující bod **V4.8** vyžaduje ošetření veškerých výstupních dat, která nejsou důvěryhodná. V praxi jsou myšleny takové údaje, které zadává uživatel, a jsou dále v aplikaci prezentovány. Z hlediska zabezpečení jsou nedůvěryhodnými daty myšleny HTML značky, CSS styly a URL atributy. Požadavek klade důraz především na kontext, kde je nutné provádět zalomení těchto znaků. [20]

Následující body při návrhu nebyly uváženy, z tohoto důvodu body **V4.9** až **V4.15** zde nebyly uvedeny.

4.1.2.5 **V5: Požadavky na kryptografické služby**

Následující zranitelnost byla organizací rozdělena na 7 částí. Pro splnění první úrovně zabezpečení nemusí být její kroky v aplikaci upraveny. Druhou úroveň zabezpečení upravují body **V5.1** až **V5.4**, pro třetí úroveň byly přidány do celku ještě body **V5.5** až **V5.7**. [20]

V aplikaci nebyly kryptografické služby implementovány. Jediná informace, která je zabezpečena z pohledu tajemství, je uživatelské heslo. Z tohoto důvodu zde jednotlivé body nebyly uvedeny.

4.1.2.6 **V6: Požadavky na manipulaci s chybami a logy**

Tato zranitelnost byla organizací rozdělena do 11 bodů. Ke splnění první úrovně je potřeba realizovat pouze první bod **V6.1**. Následující úroveň pak body **V6.1** až **V6.9**, a pro poslední úroveň je potřebné zajistit všech **V6.1** až **V6.11** bodů zranitelnosti. [20]

Bodem **V6.1** je upravována nutnost zakázání zobrazení veškerých chybových hlášení, které server může zobrazit. Tato oblast se taktéž týká varování. Provedení tohoto kroku je důležité, jelikož by z případného chybového hlášení mohl útoční vyčíst několik informací o serveru, případně část struktury aplikace, popřípadě i použitý Framework. [52]

Další body zabezpečení, které upravují nutnost zpracování chyb bez jejich zobrazení, autentizace při výskytu chyby a podobně v této oblasti zranitelnosti nebyly aplikovány.

4.1.2.7 **V7: Požadavky na ochranu dat**

Tento bod byl organizací rozdělen celkově na 8 bodů. První úroveň zabezpečení vyžaduje plnění bodů **V7.1** a **V7.2**. Pro získání druhé úrovně zabezpečení proti této zranitelnosti jsou vyžadovány body **V7.1** až **V7.4**. A v poslední řadě je organizací upraveno zabezpečení pro třetí úroveň, kdy jsou předpokládány splnit veškeré body.

Bodem **V7.1** je upravována část zranitelnosti tak, aby veškeré formuláře, které zasílají citlivé údaje uživatelů, měly vypnutý caching na straně uživatele. Tento krok bývá prováděn z důvodu zamezení přečtení uživatelových údajů osobě, která by mohla zařízení používat. Tato část z pohledu uživatele příliš vítána není, bohužel je nutná. Řešením je nastavení hlaviček pomocí „header()“ [20]. Tímto bodem je dále upravována podmínka, že veškeré funkce automatického doplňování musí být vypnuta. Postup metody již byl uveden bodě V1.2, pod kapitolou (4.1.2.1 V1: Požadavky na ověření autentizace).

Následujícím bodem **V7.2** je upravována zranitelnost z hlediska metodiky zasílání dat na server. Riziko souvisí s používáním metody GET, při které jsou data zasílána formou parametrů v adrese URL. Tímto přístupem by byl vytvořen odkaz, který by byl napadnutelný útokem zvaným „*Cross-Site Request Forgery*“ (dále jen CSRF), případně XSS, nehledě na to, že takový odkaz lze uložit například jako záložka v prohlížeči, a to v lepším případě. Pro zasílání citlivých dat, nebo provádění akcí je vyžadováno používat metodu POST. Metodou jsou data zasílána v rámci protokolu HTTP/HTTPS, nikoliv však v rámci URL adresy. Tímto přístupem je možno částečně omezit riziko spojené s útokem CSRF i XSS, nýbrž je zasílá v těle protokolu HTTP. [20]

Třetím bodem v pořadí je **V7.3**, který sděluje nutnost ošetřit funkce zálohování, caching nebo dočasné ukládání citlivých údajů na straně klienta, konkrétně například nastavením zaslaných hlaviček dokumentu s uvedením „no-cache“ a „no-store Cache-Control“. [20]

Následujícím bodem **V7.4** jsou upravovány stejné podmínky, jako bylo uvedeno v bodě **V7.3**, s tím rozdílem, že se jedná o mezipaměť popřípadě jiné dočasné úložiště na straně serveru. Veškerá tato data musí být bezpečně odstraněna tak, aby je nebylo možno zneužít. [20]

Dalším bodem je **V7.5**. Tento bod upravuje podmínky jak s přenosem citlivých informací, tak s jejím identifikováním. Je vyžadováno, aby data v systému byla po celou dobu zabezpečena, jak z hlediska přenosu, tak i po dobu kdy jsou uložena. Tím je docíleno, aby s daty nebylo možno operovat navenek aplikace. [20]

Dále jsou upravovány složitější podmínky zabezpečení, které do systému nebyly implementovány.

4.1.2.8 *V8: Požadavky na zabezpečení komunikace*

Tento druh zranitelnosti byl organizací rozčleněn do 9 bodů. Pro splnění první úrovně zabezpečení je vyžadován pouze bod **V8.1**, následně tento bod a navíc body až do **V8.5** jsou nutné pro dosažení druhé úrovně zabezpečení. Následně veškeré body až do **V8.9** jsou nutné pro dosažení nevyššího třetího stupně zabezpečení.

Bodem **V8.1** jsou stanoveny požadavky na certifikát, který je používán v rámci přenosu dat protokolem HTTPS. Tímto bodem je stanovena nutnost vlastnit certifikát, který byl podepsán ověřenou a důvěryhodnou certifikační autoritou. V rámci České Republiky patří mezi nejznámější výčtově „Czechia“, „I. CA“, „TrustPort“, a dále. O možnostech získání podepsaného certifikátu se lze dočíst ve zdroji [54]. [20]

Bod **V8.2** upravuje povinnost zkontrolovat, zda veškerá komunikace se serverem probíhá s využitím SSL/TSL a není možné jakýmkoliv způsobem zvenčí přepnout komunikaci protokolem HTTP. [59]

Následujícími body je dále upravována tato zranitelnost, avšak do aplikace nebyla zanesena.

4.1.2.9 *V9: Požadavky na zabezpečení HTTP*

Následující zranitelnost byla organizací rozčleněna na 4 body. Pro dosažení první úrovně zabezpečení je vyžadováno splnění první třech bodů **V9.1** až **V9.3**. Následně pro dosažení zabezpečení druhé a třetí úrovně jsou vyžadovány veškeré body **V9.1** až **V9.4**.

Bod **V9.1** se zabývá zabezpečením metod protokolu HTTP. Tímto bodem organizace upravuje zakázání metod HTTP protokolu, které nejsou využívány [20]. Nejpoužívanějšími metody jsou GET a POST, ale ve skutečnosti ještě existují i další metody, konkrétně pak PUT, HEAD, DELETE, CONNECT a TRACE. Toto označení se týká pro Linux server, u serveru Windows jsou dostupné i další metody, které je nutno zabezpečit. Aplikace je určena pouze pro Linux servery. Jejich užití může pro navržený systém představovat hrozbu především útokem označovaným jako „Cross Site Tracing“ [55]. V případě tohoto útoku by mohly být odcizeny údaje uživatelů.

Následující bod **V9.2** požaduje, aby veškeré odpovědi ze serveru byly přijaty s bezpečným kódováním UTF-8. [20]

V dalším bodu **V9.3** je zmíněn typ útoku nazývaný „Click Jacking“. Jedná se o druh útoku, kdy je útočníkem na legitimní stránku nanesen další objekt, kterému byla nastavena prů-

hlednost na maximální hodnotu. Poté se návštěvník domnívá, že se skutečně jedná o neškodnou stránku, ale ve skutečnosti kliká na jinou stránku, odkud jsou události přesměrovány do jiné domény. Lze objevit praktiky formou her a podobně. Často se tak útočníci snaží šířit stránky s nevhodným obsahem, popřípadě s reklamou. [20][50]

Bodem **V9.4** bylo ustanoveno, že hlavičky musí být podrobeny testování na přítomnost výhradně ASCII znaků [20]. Nespornou výhodou je fakt, že skriptovací jazyk PHP od verze 5.1.2 provádí kontrolu automaticky, tedy se o tuto zranitelnost vývojáři nemusejí starat. [56][8]

4.1.2.10 *V10: Požadavky na ověření škodlivých požadavku*

Tato zranitelnost byla organizací rozdělena do 11 bodů. Veškeré body se týkají výhradně s třetí úrovní zabezpečení [20]. Z tohoto důvodu při návrhu aplikace nebyly brány na vědomí.

4.1.2.11 *V11: Požadavky na ověření obchodní logiky*

Zranitelnost je organizací rozdělována do 10 kroků. Veškeré kroky jsou vyžadovány pro zabezpečení druhé a třetí úrovně. Po důkladném nastudování veškerých rizik spojených s ignorováním tohoto kroku bylo dojito k závěru, že systém je chráněn z hlediska základní logiky obchodu. Výčtově pak.

- nelze manipulovat s cenou, slevou, ani DPH zboží v průběhu objednávky
- veškeré objednávané zboží je identifikováno, zda skutečně existuje
- obchod nezpracovává žádné údaje týkající se platebních karet a jiných údajů

Dále nebyla tato zranitelnost ověřována, jelikož nebyly splněny předpoklady pro tuto úroveň.

4.1.2.12 *V12: Požadavky na zabezpečení souborů a jiných zdrojů*

Tento bod byl organizací rozdělen do 10 bodů, kde prvních 6 bodů je nutné splnit pro zajištění první úrovně zabezpečení. Veškeré body pak pro splnění druhé a třetí úrovně zabezpečení.

Bodem **V12.1** jsou upraveny mechanismy přesměrování [20]. Pokud by v systému byly aplikovány způsoby přesměrování využívající vstupní parametry, u kterých nedochází k validaci, bylo by možné útočníkem přesměrovat uživatele do jiné aplikace.

V bodě **V12.2** je zmíněn útok nazvaný „Path Traversal“. Jedná se o útok, kdy potenciální útočník může využít absolutních cest, kde provede úpravu, a díky této změně je mu umožněn přístup na soubory nebo adresáře, kde přístup není autentizován. V definici organizace zní, že lze vstoupit do adresářů mimo hlavní kořen aplikace, a tak získat přístup i na konfigurační soubory serveru. [57]

Bod **V12.3** sděluje nutnost veškeré vkládané soubory na server validovat kvalitním a pravidelně aktualizovaným antivirovým programem. [20]

S bodem **V12.4** souvisí pojem „File Inclusion“. Dále bod **V12.5** taktéž s touto problematikou souvisí. Rozdíly však nastávají v přístupu k problému. Bod **V12.4** se zabývá problémem lokálně umístěného a spouštěného cizího skriptu uvnitř aplikace, kdežto bod **V12.5** se zabývá spouštěním skriptu na vzdáleném serveru. K této zranitelnosti byl uveden požadavek na důslednou validaci vstupu. V obou případech může dojít ke krádeži dat ze serveru. [20]

Bod **V12.6** navazuje na bod **V12.5** kde je organizací zmíněn útok „Cross Frame Scripting“ (XFS), kterým je možno díky rámcům „<iframe>“ spouštět škodlivý kód ze vzdáleného místa. Bod taktéž zahrnuje útok „Cross Origin Resource Sharing“. [20][59]

4.1.2.13 *Požadavky na zabezpečení mobilní platformy*

Tento bod byl organizací rozdělen celkově do 26 bodů. Navržená aplikace nebyla navržena pro mobilní platformy, z tohoto důvodu nebyly prováděny testy v této oblasti.

4.1.3 **Popis plnění požadavku ASVS**

V následujících kapitolách bylo rozvedeno řešení zabezpečení, rozčleněné do bloků tak, jak je organizací bezpečnost upravována.

4.1.3.1 *VI: Požadavky na ověření autentizace*

Bod **V1.1** byl splněn. V aplikaci bylo navrženo několik částí, které musí být bezpodmínečně zabezpečeny. Z hlediska „front-end“²⁰ byly zabezpečeny části pro účely administrace vlastních údajů klienta a správu provedených objednávek klientem. Z hlediska „back-

²⁰ Front-End – část aplikace, kde má přístup široká veřejnost

end²¹“ byly zabezpečeny veškeré stránky, jelikož se jedná o část aplikace, kde byl povolen přístup jediné osobě, a to administrátoru celého systému.

Plnění úlohy bylo provedeno přesměrováním uživatele na přihlašovací stránku aplikace. V případě administrativní části se jedná o přihlašovací stránku tohoto modulu.

Bod **V1.2** byl splněn. Na veškerá formulářová pole, kde je vyžadováno heslo, byl aplikován mechanismus, který zamezil zobrazení uživatelských údajů. Z důvodu zrušení podpory HTML tagu „autocomplete=‘off‘“ musela být zcela zrušena operace odeslání samotným formulářem a činnost provedena AJAX voláním.

Bod **V1.3** byl splněn. Pokud dojde k vypršení platnosti přihlášení, nebo jeho selhání, je provedeno bezpečné zničení relace tak, aby ji nemohl útočník použít.

U bodu **V1.4** bývá ve většině aplikací problém, jelikož nepoužívají šifrované spojení protokolem HTTPS. K funkčnosti tohoto protokolu je potřeba zajistit bezpečnostní certifikát, který bohužel není zadarmo. Aplikace byla navržena tak, aby umožňovala nastavit přítomnost certifikátu, a tím zajistit celé spojení šifrované.

Většina solidních firem nabízejících webhostingové služby umožňují v rámci jejich tarifu využít bezpečnostní certifikát registrovaný na jejich doménu, s využitím certifikátu je možno zabezpečit spojení mezi serverem a klientem, ale bude generována hláška „identitu nelze ověřit“. Aplikace musí být umístěna na server firmy, která tuto možnost nabízí. Z tohoto důvodu bylo zabezpečené spojení ponecháno výhradě pro účely administrace, kde administrátora tato hláška nijak neobtěžuje, pouze schválí bezpečnostní výjimku a veškerá komunikace včetně identifikátoru relace (SID) je přenášena šifrovaným spojením. Z hlediska „front-end“ aplikace je použití tohoto certifikátu nemožné, jelikož by každý návštěvník musel schvalovat výjimku a aplikace by nepůsobila seriózně. Spojení mezi „front-end“ aplikací a serverem tedy zůstalo ponecháno protokolu HTTP, ale nastavení lze změnit, více v kapitole (5 Nasazení aplikace). Řešením je zakoupit vlastní certifikát podepsaný certifikační autoritou a zabezpečit tak veškerou komunikaci.

²¹ Back-End – část aplikace, která slouží pro účely administrace kompletního systému, přístup povolen výhradně s oprávněním administrátora

Nastavení zabezpečení „back-end“ aplikace bylo provedeno s využitím souboru „.htaccess“, který je umístěn v kořenovém adresáři aplikace, kde byla zkonstruována podmínka realizující tuto činnost.

Bod **V1.5** byl splněn, jelikož aplikace v současném stavu umožňuje heslo změnit pouze přihlášenému uživateli.

Bod **V1.6** byl splněn. V procesu autentizace nejsou uváděny žádné specifické údaje.

Bod **V1.7** byl splněn. Veškeré ladící informace, které byly v průběhu aplikace využívány, se odstranily.

Z dalších bodů, které souvisí s požadavky na ověření autentizace, a nebyly ve výčtu předchozí kapitoly uvedeny, bylo splněno takzvané „solení“ hesel.

4.1.3.2 **V2: Požadavky na ověření správy relací**

Bod **V2.1** byl splněn. Veškeré operace, které jsou prováděny s údaji související s relací, jsou prováděny interními funkcemi jazyka PHP, nebyly vytvořeny žádné vlastní funkce.

Bod **V2.2** byl splněn. Relace po odhlášení je bezpečně smazána dle doporučení dokumentace PHP. [8]

Bod **V2.3** byl splněn. Veškeré neaktivní relace, které nejsou používány déle jak 24 minut, jsou automaticky zničeny.

Bod **V2.4** byl splněn. Na všech stránkách aplikace je po přihlášení možno provést odhlášení.

Bod **V2.5** byl splněn. Identifikátor relace (SID) není přenášen v URL adrese, ale pomocí cookies. V případě „back-end“ aplikace je session řádně šifrován, a tak je znemožněn odposlech spojení.

Bod **V2.6** byl splněn. Po provedení odhlášení ze systému je veškerá předchozí relace ukončena. Dále je vytvořena nová relace.

Bod **V2.7** byl splněn. Veškerým vytvářen relacím je identifikátoru SID nastavován příznak „HttpOnly“, a tak je znemožněno operovat s cookie využitím například JS.

Bod **V2.8** nebyl zcela splněn. Takzvaný „secure flag“ je nastavován pouze v případě administrativního rozhraní, kde je vyžadováno vyšší zabezpečení oproti „front-end“ aplikaci.

Body **V2.9** a **V2.10** byly splněny. Při každém procesu re-autentizace je prováděno regenerování SID. Regenerace není prováděna v případě AJAX volání. Díky tomuto prvku je předcházeno útoku „Session Fixation“.

Bod **V2.11** byl splněn. Výhradně aplikací jsou generovány identifikátory SID, ty jsou následně regulárním výrazem ověřovány, zda mají správnou délku a obsahují validní znaky.

Bod **V2.12** byl splněn. Nastavením serveru bylo upraveno výchozí nastavení pro mechanismy generování identifikátoru. Došlo k nastavení hash algoritmu, kterým byl zvolen „SHA512“. Tím bylo takřka znemožněno uhodnutí identifikátoru. Dále bylo umožněno generování hash se znaky „-“ a „_“. Díky úpravám se podařilo vytvořit co nejdokonalejší generování v rámci jazyka PHP.

Bod **V2.13** byl splněn. „Front-end“ aplikaci je přidělována jiná relace, než k „back-end“ aplikaci. Nastavení je dáno právě změnou cest, pro které by byly relace vytvářeny. Na systém potvrzování relací dále navazuje vytvořená třída „Security“, díky níž jsou tokenizovány²² veškeré napadnutelné operace. Třída byla dále popsána v kapitole 4.2.1 Třída Security.

Bod **V2.14** byl splněn. Na začátku všech požadavků je ověřováno, zda uživatel navštívil aplikaci pod stejnou IP adresou a identifikátorem prohlížeče, jako v předchozí návštěvě.

Bod **V2.15** nebyl zcela splněn. Absolutní doba přihlášení byla implementována pouze do administrátorské části aplikace, kde má administrátor vymezený čas pro provádění operací.

4.1.3.3 **V3: Požadavky na Access Control list**

Touto oblastí je upravováno celkem 14 bodů. Splnění požadavků **V3.1** až **V3.7** jsou předpokladem pro splnění první úrovně zabezpečení. Následuje plnění bodů **V3.8** až **V3.13**, včetně všech předchozích, oblast je vymezena pro požadavky druhé úrovně zabezpečení. Pro splnění požadavků třetí úrovně musí být splněn i bod **V3.14**.

Do aplikace centrální ACL nebyl zanesen, a tak v systému nedochází k přidělování práv jednotlivým uživatelům. Ve výsledku ale existují dvě úrovně oprávnění. Nejvyšší oprávnění má vlastník aplikace, tomu byl v databázi přidělen identifikátor 0, který jej opravňuje

²² Tokenizace - proces, kdy je generován skrytý řetězec, který je uložen na serveru a současně ve skrytém poli formuláře. Následně je zaslán požadavek na server, kde jsou tyto řetězce porovnány, čímž dochází k procesu autorizace uživatele.

vykonávat úkoly spojené s administrací. Ostatní uživatelé mají běžná práva kdy je jim umožněn přístup do „front-end“ aplikace.

Body **V3.1** až **V3.4** byly splněny. Rozdíl je pouze ve způsobu provádění těchto operací, kdy dochází k ověřování pomocí jazyka PHP a nejsou zasílány požadavky na databázi. Systém byl navržen primárně pro správu jediným uživatelem, z tohoto důvodu je výhodné zbytečně databázi nezatěžovat.

Bod **V3.5** byl splněn. Pomocí souboru „.htaccess“ byl zakázán vstup do adresářů, sloužících pro chod celé aplikace. Díky mikro frameworků *Slim v2.4.3*, došlo ke generování fiktivních odkazů, díky nimž došlo částečně ke schování adresářové struktury.

Body **V3.4** a **V3.6** byly ošetřeny z hlediska zranitelnosti. Veškeré odkazy byly prověřeny, zda jsou řádně zabezpečeny z hlediska zranitelnosti. V systému neexistuje žádný takový odkaz, který by útočník mohl zneužít, a tím obejít proces autorizace.

Bod **V3.7** byl splněn. Jestliže dojde k návštěvě části aplikace, kde je prováděn proces autorizace, a uživateli nebylo uděleno oprávnění, je mu přístup zakázán. Situace je řešena přesměrováním na stránku, kde je umožněno přihlášení.

Další body **V3.8** až **3.14** nebyly v rámci systému uplatněny. Systém neobsahuje natolik citlivé údaje, jako jsou čísla platebních karet, občanských průkazů a podobně. Z tohoto důvodu nebyl na tyto pravidla brán zřetel.

4.1.3.4 **V4: Požadavky na validaci vstupů**

Validace vstupů jsou velice důležité operace, v případě neošetření byť jediného vstupu, dává se útočníkovi prostor pro útok. S tímto tématem souvisí především nejčastější útok „*SQL Injection*“, kterým jsou upravovány, nebo získávány data z databáze. Dalšími známými útoky jsou pak XSS, „*LDAP Injection*“ a velmi nebezpečný CSRF. Autor si při návrhu byl plně vědom těchto útoků a vynasnažil se veškerým zranitelnostem předejít. V následujících odstavcích budou vysvětleny principy použitých zabezpečení.

Bod **V4.1** byl splněn. Konkrétně je tento problém vztažen ke skriptovacímu jazyku PHP. Tak, jako je tomu i u ostatních jazyků, pro jejich funkčnost je vyžadována paměť. Jazyk PHP disponuje propracovanou správou paměti, o kterou se stará GC. Nevyskytuje se zde takový problém, který lze nalézt s problematikou přidělování paměti, jako například u jazyka C. Autor byl s problematikou obeznámen a uvědomuje si, že i v rámci PHP je paměť omezena. Veškeré algoritmy z tohoto důvodu provádějí požadavky na uvolnění nevyužité

části paměti. PHP následně provede uvolnění v době, kdy nebude potřebován výkon pro jinou složitější operaci. Navržené algoritmy nejsou pro využití paměti natolik kritické, aby mohlo dojít k selhání, tak jak je upravováno tímto bodem.

Na následující bod **V4.2** byl brán obzvláště veliký důraz, jelikož se jedná o nejčastější a také nejnebezpečnější útok na webové aplikace v síti internet. Souvisí s útokem „SQL Injection“. Veškeré dotazy, které jsou prováděny směrem k databázi, byly jeden po druhém ověřeny, zda jsou správně ošetřeny proti tomuto útoku. S touto částí stojí za zmínku fakt, že aplikace využívá moderní databázový ovladač „MySQLi“, čímž jsou pokryty veškeré nedostatky výchozího ovladače „MySQL“. Uvnitř aplikace byly využity dvě metody chránění se proti tomuto útoku. Mezi základní se řadí klasický „escape²³“ veškerých řetězců vstupních dat. Dále veškeré číselné údaje jsou ošetřeny funkcí „intval()“ tak, aby bylo na vstupu zajištěno skutečně číslo. Posledním využitým nástrojem bylo využití vázaných proměnných. Tuto možnost umožňuje využít samotný ovladač „MySQLi“, kde jsou operace spojené s dotazy na databázi rozděleny na dílčí problémy, při kterých dochází k validaci vstupu, a až jako poslední krok je prováděno samotné vykonání. Aplikace tedy využívá kombinaci těchto technik, aby se předcházelo útoku. Na nejvíce vytěžované a potenciálně nejnebezpečnější části byl aplikován mechanismus vázaných proměnných. Tedy bod **V4.2**, požadavek ošetření zranitelnosti z hlediska „SQL Injection“, byl splněn.

Bodem **V4.3** je upravována nutnost zabezpečit aplikaci proti útoku „Cross Site Scripting“ (XSS). Jedná se z hlediska všech útoků o nejstarší zranitelnost v oboru [46]. Na veškerých stránkách, kde je interpretován posléze uživatelův vstup, byl výstup ošetřen interní funkcí PHP „htmlspecialchars(string, ENT_QUOTES)“. Tím bylo znemožněno útočnickovi provedení jeho nekalých praktik.

Bodu **V4.4** je splněn. Problematika je věnována „LDAP injection“. V tomto projektu nebyl využit LDAP protokol.

Bod **V4.5** je splněn. Zranitelnost se zaměřuje na útoky typu „OS Command Injection“. Navržená aplikace neinterpretuje žádné serverové funkce, a je proti tomuto útoku chráněna.

²³ Escape – je prováděno „zalomitkování“ nebezpečných znaků. Z manuálu lze vyčíst konkrétní věta, kterou jsou zalomeny znaky: NULL (ASCII 0), \n, \r, \, ', ", a Control-Z. [8]

Bod **V4.6** byl splněn. Veškeré vstupy z formulářů v aplikaci umístěných, především skript zpracovávající údaje z registračního formuláře, jsou validovány a v případě detekce chyby je vstup zamítnut. Tento druh validace byl čistě založen na principu „whitelist“.

Bod **V4.7** byl splněn. Aplikace provádí validaci na straně klienta, která slouží jako první úroveň zabezpečení proti této zranitelnosti. Díky této technice, ze strany seriózních uživatelů obchodu dochází k ulehčení zátěže serveru. V druhé fázi jsou data validována na straně serveru, kde jsou odchyťování spíše útočníci. V případě naleznutí nevalidního vstupu, předají skripty hlášení o nevalidní skutečnosti.

Bod **V4.8** byl splněn. Zranitelnost vyžaduje ošetření veškerých výstupních dat, která nejsou důvěryhodná. Veškerá data jsou podrobena validaci.

Ostatní body této zranitelnosti nebyly brány v potaz.

4.1.3.5 **V5: Požadavky na kryptografické služby**

V aplikaci byl zabezpečen jediný citlivý údaj, kterým je heslo uživatele. Ostatní body upravované organizací OWASP v rámci ASVS v projektu nebyly řešeny.

4.1.3.6 **V6: Požadavky na manipulaci s chybami a logy**

V rámci této zranitelnosti byl splněn pouze jediný bod **V6.1**, který byl splněn provedením úpravy souboru „.user.ini“. V tomto souboru bylo provedeno zakázání zobrazování chyb klauzulí „error_reporting = 0“.

V rámci aplikace není nijak dále manipulováno s chybami, i když by jejich nastavení nebyla složitá operace. Se zapnutím logování výjimek a událostí jsou spojeny komplikace související s nutností vytvořit modul, který by obstarával správu, jelikož by mohl narůstat bez povšimnutí na velikosti. Ovšem pokud by byl takový systém vytvořen, vývojář by měl k dispozici mocný nástroj pro opravení nedostatků aplikace. Což je výhodou.

4.1.3.7 **V7: Požadavky na ochranu dat**

Bod **V7.1** a **V7.3** spolu úzce souvisí a byly splněny. Došlo k nastavení hlaviček veškerých dokumentů tak, aby nebyla možnost citlivé údaje zaznamenat. V této souvislosti byl postup převzat autorem *Paul Underwood*, který na svém blogu dává řešení k dispozici. Viz [53]

Bod **V7.2** upravuje metodiku zaslání dat na server. ASVS stanovil data zasílat výhradně formou POST, aby nemohlo být zneužito URL adresy, která by byla generována formulá-

řem. Díky této úpravě bylo částečně zabráněno útoku CSRF, ale rozhodně se nejedná o finální zabezpečení oproti této hrozbě. CSRF se lze bránit metodou, takzvanou „tokenizací“ všech formulářů. Více k této problematice bylo popsáno v kapitole 4.2.1 Třída Security.

Bod **V7.4** byl splněn. Aplikace v průběhu procesu nákupu, nebo po dobu přihlášení využívá relaci pro dočasné uložení dat. Tato data jsou po ukončení relace bezpečně smazána a tím není povolena žádná operace s využitím dat této relace.

Bod **V7.5** nebyl splněn. Data po dobu přenosu sice mohou být šifrována, ale po uložení do databáze dochází k zabezpečení pouze hesla uživatele, nikoliv veškerých údajů, tedy tento bod nebyl splněn.

Dále následují zranitelnosti bodu **V7**, které při návrhu aplikaci již nebyly zohledněny.

4.1.3.8 **V8: Požadavky na zabezpečení komunikace**

Tento bod je možné splnit pouze s vlastnictvím certifikátu, který je digitálně podepsán ověřenou certifikační autoritou (CA). Právě toto znění bylo naleznuto pro bod **V8.1**. Aplikace používá ve výchozím nastavení zabezpečené připojení pouze v rámci „back-end“ aplikace. Tato část byla zabezpečena cizím certifikátem, a tak používaný certifikát není digitálně podepsán CA. Problém samozřejmě lze vyřešit zakoupením vlastního certifikátu, který by byl ověřenou CA podepsán, a tudíž by byl bod **V8.1** splněn.

Bod **V8.2** nebyl splněn. Tato část vyžaduje, aby veškerá komunikace byla zabezpečena za pomoci SSL/TSL. Tato podmínka je z důvodu využití nepodepsaného CA splněna pouze pro „back-end“ aplikaci.

Následující úpravy zranitelnosti bodu **V8** nebyly při návrhu aplikace řešeny.

4.1.3.9 **V9: Požadavky na zabezpečení HTTP**

Bod **V9.1** byl splněn. U navrženého systému byly explicitně zakázány nepoužívané metody v konfiguraci serveru díky souboru „.htaccess“. Byla provedena úprava, kterou lze pozorovat ze zdrojového textu Zdrojový text 5.

```
1 RewriteCond %{REQUEST_METHOD} TRACE|PUT|DELETE|OPTIONS|CONNECT|HEAD
2 RewriteRule .* - [F]
```

Zdrojový text 5 Obsah souboru „.htaccess“ zajišťující vypnutí nepoužívaných metod

Bod **V9.2** byl splněn. Pomocí funkce „header()“ bylo upraveno kódování v celé aplikaci.

Bod **V9.3** byl splněn. Uvnitř aplikace pomocí funkce „header()“ došlo k nastavení „X-Frame-Options: SAMEORIGIN“. Tímto krokem bylo útočnickům znemožněno použít útok „Click Jacking“. Aplikace využívá rámy v administrativní části aplikace, z tohoto důvodu bylo nastaveno „SAMEORIGIN“. Organizace upozorňuje, že tento krok nemusí být dostatečný, jelikož starší prohlížeče tuto hlavičku neinterpretují.

Bod **V9.4** byl splněn. Jádro PHP provádí automatickou kontrolu hlaviček.

4.1.3.10 *V12: Požadavky na zabezpečení souborů a jiných zdrojů*

Bod **V12.1** byl splněn. V navrženém systému je tato technika praktikována pouze v části obchodu. Zde jsou vstupními parametry názvy jednotlivých skupin, popřípadě název produktu. Tyto vstupy jsou však plně ošetřeny vůči tomuto druhu zranitelnosti.

Bod **V12.2** byl splněn. Aplikace je vůči tomuto útoku plně zabezpečena z důvodu nepoužívání vstupu pro volání příkazů souborového systému. Veškeré složky souborového systému mimo části obrázků a skriptu JS, jsou ošetřeny zakázáním přístupu s pomocí souboru „.htaccess“ a zápisem „Deny from all“.

Bod **V12.3** je závislý na individuálním přístupu administrátora. Systém poskytuje 2 možnosti vložení souboru, kterým musí být obrázek. Je potřebné, aby administrátor k tomuto problému přistupoval zodpovědně, jelikož musí být eliminovány veškerá rizika spojená se zanesením viru do aplikace. Soubor je možno vložit pouze z administrativního rozhraní aplikace. Tudíž nelze garantovat splnění tohoto bodu.

Body **V12.4** a **V12.5** mají souvislost s funkcemi „include()“, „require()“ a jejich zástupci s rozdílnou interpretací vložení přípony „_once“. Uvnitř aplikace jsou výhradně spouštěny skripty, které mají pevně definovanou cestu k souboru, která není možná změnit. Díky tomuto opatření je předpoklad zabezpečení splněn. V systému byla vytvořena výjimka, ale pro tu je nutné oprávnění administrátora.

Bod **V12.6** byl splněn. Aktuálně nastavené zabezpečení, včetně nastavení hlavičky „X-Frame-Options: SAMEORIGIN“, zakazuje přístup z cizího zdroje. Pro útok „Cross Origin Resource Sharing“ nebyla zahrnuta hlavička, která by umožnila tento druh útoku.

Další body **V12** včetně **V13** nebyly při návrhu uváženy.

4.2 Další části zabezpečení

V této kapitole byly uvedeny mechanismy, které jsou prováděny pro zajištění co nejvyššího stupně zabezpečení. Mezi tyto prvky byla zařazena třída „Secure“.

4.2.1 Třída Security

Tato třída byla navržena pro účely „tokenizace“ veškerých formulářů. S pojmem token souvisí obrana proti útoku CSRF. Veškeré skripty zpracovávající data z formulářů, vyžadují pro spuštění účelové části skriptu autorizaci. Dochází k ověření, zda dotaz opravdu přichází ze samotné aplikace, a nikdo se skriptu nesnaží externě zneužít, navíc jsou přijata data pouze s hlavičkou „X-Requested-With:XMLHttpRequest“, tedy hlavičkou s podpisem AJAX.

S třídou taktéž souvisí používání proměnné „HTTP_REFERER“, kde lze nalézt stránku aplikace, odkud uživatel spouští skript. Zasílání proměnná lze bohužel například v prohlizeči Opera vypnout, a pak tato proměnná nemá význam. Bohužel tak jako mnoho systémů, lze i zabezpečení pomocí proměnné „HTTP_REFERER“ obejít, třída však používá tokeny a i v případě zakázání „HTTP_REFERER“ plní další prvek ochrany. V konfigurační části aplikace lze nastavit, zda má být tato proměnná využívána, či nikoliv.

```
1 <?php
2  /** Class Security */
3  class Security{
4      /** @var boolean */
5      protected $authorized;
6      /** @var string */
7      protected $token;
8      /** Čas, po který je token použitelný [s] */
9      private $timeForAcceptedToken = 3600;
10     /** Rozsah maximálního počtu použití tokenu */
11     private $minMaxUsagesToken = [30,40];
12     /** @var array */
13     private $enable_hosts;
14     public function __construct(){
15         /** Kontroluje, jestli má uživatel oprávnění vstoupit na stránku */
16         public function checkAuthorization(){
17             /** Nastaví JSON hlavičku a poskytne "reload"=>true,"success"=>false,
18              "errStr"=>"Unauthorized!" */
19         public function getJSONUnauthorized(){
20             /** Nastaví HTML hlavičku a poskytne text "Unauthorized" */
21         public function getHTMLUnauthorized(){
22             /** Genetuje nový token do session a $this->token, čas vygenerování */
23         private function newToken(){
24             /** Provádí ověření validity tokenu, zda lze ještě použít */
25         private function checkValidityTokenAndUpdate(){
26         /** Getter pro token */
27         public function getToken(){
```

```
27     /** Getter pro IP adresu klienta */
28     public static function getClientIp() {}
29 };
```

Zdrojový text 6 Třída Security a její metody

4.2.1.1 *Popis funkce třídy*

Třída byla implementována do všech stránek aplikace a její funkce je následující. V případě zobrazení stránky metodou GET, tedy v současném návrhu nějaké vizuální prezentace, je do aplikace zanesen skrytý DOM element input, jehož hodnota je třídou vytvořený validní token. Při vznesení AJAX požadavku na zpracování dat serverem je současně s požadavkem a daty odeslán i vygenerovaný token. Následně veškeré PHP skripty, které jsou využívány pro zpracování dat, nebo jejich získání, provádějí na počátku autorizaci, jež je prováděna porovnáním příchozího tokenu s tokenem uloženým v úložišti relace. Ještě před samotným porovnáním je ověřována validita tokenu, ta je vykonávána při vytváření nové instance třídy konstruktorem, kde je volána metoda „checkValidityTokenAndUpdate()“, v případě detekce nevalidního tokenu je generován nový token, čímž dojde k selhání procesu autorizace. Následně je pro zpracování veškerých požadavků potřebné získání tokenu i do prezentace, odkud jsou požadavky vznášeny. Tohoto se docílí prostým obnovením prohlížené stránky.

V třídě je s procesem autorizace, v případě povolení využívání proměnné „HTTP_REFERER“, ověřována skutečnost, zda dotaz skutečně pochází z vytvořené aplikace. Dále je ještě ověřována IP adresa, pro kterou je relace vytvořena, zda se shoduje s IP adresou požadavku. Pokud byt jeden z výše uvedených prvků zabezpečení detekuje selhání, je automaticky generován token nový, čímž dojde k zneplatnění předchozího tokenu v rámci relace.

Politika generování tokenu byla zvolena tak, aby systém byl vhodně zabezpečen. Ve výchozím nastavení byla tokenu nastavena životnost na dvě hodiny „\$timeForAcceptedToken“, a také je generován maximální počet použití tokenu, který je plovoucí v rozmezí stanovené polem „\$minMaxUsagesToken“, ve výchozím nastavení bylo nastaveno rozmezí dané intervalem (30, 40), s tímto návrhem je složitější odchyťávat nově generovaný token.

5 NASAZENÍ APLIKACE

U většiny aplikací, které bývají poskytovány, je velice důležitá pohodlnost v rámci individuálního nastavení, ať již při prvotním uvedení do provozu, nebo případnými změnami.

Při návrhu aplikace byl dbán důraz na personalizaci aplikace, díky této volbě je možno provést veškeré nastavení z jediného souboru, který byl umístěn do aplikace, konkrétně „./xampp/htdocs/www/private/config.php“.

Dále budou popsány jednotlivé kroky, které je potřeba zajistit pro korektní funkci celé aplikace, včetně založení webhostingu²⁴ a domén.

5.1 Požadavky na webhostingové služby

Aplikace byla vyvíjena na odzkoušených technologiích, ve kterých jsou používány nové funkce, jež v předchozích verzích nebyly k dispozici, pro korektní funkci aplikace musí být splněna minimální konfigurace serveru následující. Veškeré verze mohou být novější.

Název	Hodnota
Verze Apache	2.4.4
Verze PHP	5.5.1
Verze MySQL	5.5.28
Minimální webový prostor	50 MB
Minimální prostor MySQL	150 MB
Podpora .htaccess	nutná

Tabulka 1 Minimální konfigurace serveru

Velikost webového prostoru je odvislá na předpokládaném množství položek, především pak jejich obrázků. Obrázky jsou ukládány do adresářové struktury, nikoliv do *MySQL*. Ve výchozím nastavení aplikace lze uložit obrázek o maximální velikosti 4 MB. Každá položka může obsahovat maximálně 4 obrázky. Ve skutečnosti jich však může být až 5, protože dochází k vytváření miniatury hlavního obrázku, aby se celá aplikace urychlila.

Ke korektní funkci celé aplikace je potřebné ověřit, zda poskytovatel webhostingových služeb nabízí k dispozici certifikát, který je potřebný pro administrativní část aplikace. Jestliže takový poskytovatel nenabízí využití svého certifikátu, je nutné zřídit vlastní, po-

²⁴ Webhosting – jedná se o pronájem prostoru na cizím serveru, například pro účely webové prezentace

případě se poohlédnout po jiných nabídkách webhostingu, které budou tento certifikát zpřístupňovat.

Dále musí být poskytovatelem umožněno na databázi spouštět a vytvářet procedury, funkce a triggerly, tato skutečnost musí být taktéž ověřena.

Veškerá podrobnější konfigurace serveru je obsažena již ve zdrojových kódech, tudíž by nemělo být potřeba důkladně server konfigurovat.

5.2 Průběh nasazení aplikace pro účely testování na lokálním serveru

Pokud je vyžadováno aplikaci spustit v servisním módu pro účely testování, nemusí být provedeny postupy z kapitoly 5.3, a lze pouze provést následující kroky.

- Zkopírovat složku „xampp“ z příloženého CD, které je součástí (PŘÍLOHA P I), pod operačním systémem Windows do umístění „C:\“
- Přejít do umístění „C:\xampp\“ zde pokračovat spuštěním programu „xampp-control.exe“
- Provést spuštění modulů *Apache* a *MySQL*
- Povolit výjimku Firewallu pro *Apache* a *MySQL*
- Přejít do prohlížeče a zadat adresu „http://localhost“

Aplikace se automaticky nakonfiguruje a inicializuje pro lokální spuštění. Po této inicializaci lze aplikaci na server přenést, ale bude nutné provést konfiguraci zadáním adresy „http://example.tld/new.php“. Pokud je „čistá“ aplikace rovnou zanesena na server, dojde rovnou k přesměrování na konfigurační stránku. Je však doporučeno do serveru rozbalit obsah archiv „\www.zip“.

5.3 Průběh nasazení aplikace do ostrého provozu na pronajatý server

V této kapitole bude popsán průběh umístění aplikace na webový server a následné zveřejnění. Pro kvalitní webhostingové služby byla vybrána solidní společnost *ONEsolution s.r.o.* (dále jen ONES) [1], se kterou má autor velmi pozitivní zkušenosti. Veškeré uvedené postupy se přibližně shodují s běžným postupem u většiny poskytovatelů, tudíž jednotlivé kroky lze provést i s volbou jiného poskytovatele. Dále prezentované obrázky byly použity z webového profilu společnosti a slouží pouze jako vodítko v procesu zveřejnění aplikace.

5.3.1 Založení webhostingu a domény

Tento proces se nejvíce odlišuje od většiny dílčích kroků, které se spuštěním aplikace souvisí. I když je pojednáváno o stejné problematice, většinou je poskytovateli volena jiná politika přidělování služeb. V případě ONES jsou služby přiděleny v okamžiku schválení objednávky, ještě před uhrazením fakturované částky. Pro tuto webovou aplikaci byl zvolen společností propagovaný tarif *BUSSINES* [1], který plně vyhovuje požadavkům na funkčnost celého systému. Jeho charakteristika a požadované služby byly uvedeny níže.

Název	Hodnota
Tarif	BUSSINES
Verze PHP	5.5.13
Verze MySQL	5.5.28
Webový prostor	10 GB
Prostor MySQL databáze	500 MB
Doba uchování záloh	30 dnů
Multihosting/počet účtů	Ano/3

Tabulka 2 Parametry webhostingu BUSSINES [1]

V tabulce Tabulka 2 byly uvedeny pouze základní parametry produktu firmy ONES. Uživatelem aplikace může být zvolen zcela odlišný tarif u jiného poskytovatele, především závislý na předpokládaném počtu položek, a tím i velikostí webového prostoru.

Co se týká volby domény, byly zvoleny celkově dvě domény druhého řádu s doménou prvního řádu CZ. Konkrétně pak „kvalitnina.plne.cz“ a „fox99.cz“. Z tohoto důvodu byl také potřebný multihostig²⁵. Tento krok však není v procesu nasazení aplikace potřebný.

5.3.2 Konfigurace služeb

Po provedení registrace byly obdrženy přihlašovací údaje do administrativního rozhraní, aplikace phpMyAdmin, a také údaje pro připojení se k serveru, pomocí protokolu FTP²⁶.

Většina poskytovatelů webhostingových služeb neumožňuje vytvářet databáze přímo v prostředí phpMyAdmin [61]. Vytvoření databáze musí být tudíž provedeno v administrativním prostředí společnosti. Jako další potřebný krok je nastavit nějakou vhodnou e-mail adresu, ze které budou následně zasílány servisní informace. Je doporučeno volit adresu

²⁵ Multihosting – možnost sdílet jeden webový prostor více doménami

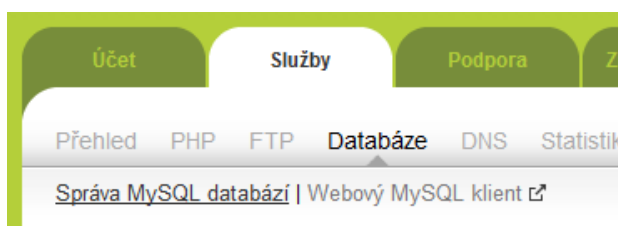
²⁶ File Transfer Protocol – protokol pro přenos souborů mezi počítači s využitím počítačové sítě

typu „robot@example.tld“, jelikož budou moci následní zákazníci tušit, že se jedná o automaticky generovanou zprávu. Dále je nutné provést specifický krok pro webhosting od společnosti ONES, kdy je nutné zanést IP adresu administrátorské sítě do listu povolených adres pro přístup na server v rámci protokolu FTP.

Aby byly splněny výše uvedené kroky, je nutné provést přihlášení do administrativního prostředí, kde budou následně provedeny. Toto prostředí je dostupné z adresy „<https://www.oneadmin.cz/onebit/>“ [1].

5.3.2.1 Vytvoření databáze

Po přihlášení v horní části administrativního rozhraní společnosti existuje karta „Služby“, na kterou je nutné kliknout. Následně je potřebné postupovat na kartu „Databáze“ a „Správa MySQL databází“, na kterou je taktéž nutné kliknout.



Obr. 14 – Správa MySQL databází na serveru společnosti ONES [1]

Následně ve spodní části aplikace je nutné kliknout na „Vytvořit MySQL databázi“, následně dojde k zobrazení dialogového okna, kde je požadováno zadat velikost databáze, popřípadě její popis. Vytvářená databáze by měla být vhodně popsána a následně ze seznamu musí být vybrána nejvyšší přípustná velikost databáze.

Vytvořit MySQL databázi	
*Název databáze:	fox99cz3
Popis:	<input type="text" value="EshopDatabase"/>
*Velikost databáze:	150 MB ▾
*Kódování/Porovnání:	utf8_czech_ci ▾

Obr. 15 Vytvoření databáze pro webovou aplikaci u společnosti ONES [1]

Jelikož se v databázi aplikace nacházejí autorem definované procedury, funkce a triggerů musí být u společnosti ONES vytvořen sekundární *MySQL* účet, kterým bude možno operace provádět. Tohoto se docílí na stejné stránce rozhraní, kde existuje tlačítko „Vytvořit sekundární *MySQL* účet“. Po kliknutí na tlačítko je zobrazeno dialogové okno s výzvou

specifikace účtu. Jako první musí být zvolena vytvořená databáze, což lze provést volbou ze seznamu. Dále v sekci oprávnění musí být zvoleny veškeré parametry mimo „LOCK TABLES“, „CREATE VIEW“ a „SHOW VIEW“, jelikož tyto funkce aplikace nepoužívá. Následně musí být zvolen popis účtu, který by měl být smysluplný, a naposled musí být zvoleno velmi silné heslo. Důrazně je doporučeno využít kombinace číslic, písmen, pomlčky, popřípadě jiného speciálního znaku, aby heslo nebylo uhodnutelné. K posledním kroku musí být akce potvrzena kliknutím na tlačítko „Vytvořit“.

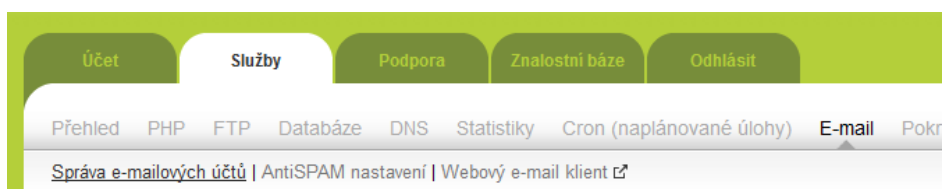
Vytvořit sekundární MySQL účet	
*Login:	fox99.cz.3
*Host:	127.0.0.1
*Databáze:	fox99cz3
Oprávnění:	<input checked="" type="checkbox"/> UPDATE <input checked="" type="checkbox"/> CREATE TMP TABLES <input checked="" type="checkbox"/> DELETE <input type="checkbox"/> LOCK TABLES <input checked="" type="checkbox"/> DROP <input type="checkbox"/> CREATE VIEW <input checked="" type="checkbox"/> INDEX <input type="checkbox"/> SHOW VIEW <input checked="" type="checkbox"/> ALTER <input checked="" type="checkbox"/> FUNCTIONS & TRIGGERS
*Popis:	SecondaryAccount
*Heslo:
*Heslo znovu:
Síla hesla:	1 2 3 4 5

Obr. 16 Příklad vytvoření *MySQL* sekundárního účtu u společnosti ONES [1]

„Název databáze“, „Login“ (přihlašovací jméno pro *MySQL*) a „Heslo“ musí být dobře uschováno, jelikož údaje budou potřebné pro konfiguraci aplikace. Těmito kroky byla dokončena databázová část.

5.3.2.2 Vytvoření servisního emailu

Nyní je potřebné v administrativním rozhraní ONES naleznout v horní části panel „Služby“, následně „E-mail“ a kliknout na „Správa e-mailových účtů“.



Obr. 17 Správa e-mailových účtů u společnosti ONES [1]

Na zobrazené stránce je potřeba kliknout na tlačítko „Vytvořit e-mailový účet“. Následně dojde k zobrazení dialogového okna, kde je nutné zadat do pole název účtu nejlépe slovo

„robot“. Je doporučeno uvést popis účtu, kvóta účtu může být ponechána na 50 MB, jelikož není předpokládána odpověď na tuto adresu. Dále je opět vyžadováno zvolit silné heslo. Po provedení těchto kroků je potřebné operaci potvrdit kliknutím na tlačítko „Vytvořit“.

Vytvořit e-mailový účet	
*Název účtu:	robot @fox99.cz
Popis:	automaticMailer
Kvóta:	50 MB
*Heslo:
*Heslo znovu:
Síla hesla:	1 2 3 4 5

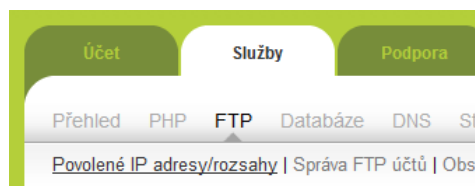
Vytvořit Zavřít

Obr. 18 Vytvoření nového e-mailového účtu robota u společnosti ONES [1]

Díky tomu kroku byl vytvořen účet, pod jehož hlavičkou budou rozesílány informační emaily. Je doporučeno vytvořit ještě jeden hlavní účet s podstatně vyšší kvótou, na který bude nastaveno přeposílání z tohoto účtu.

5.3.2.3 Povolení FTP pro administrátorskou síť

Z důvodu zabezpečení dat na serveru nejsou po vytvoření účtu u ONES povoleny žádné IP adresy, které by měly na server přístup s využitím FTP. Pro nahrání dat na server je nutné vytvořit výjimku pro IP adresu administrátorské sítě. To lze provést kliknutím na kartu „Služby“, následně zvolit „FTP“ a kliknout na „Povolené IP adresy/rozsahy“.



Obr. 19 Správa povolených IP pro přístup na server pomocí FTP u společnosti ONES [1]

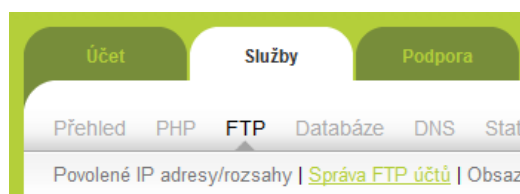
Zde je pak nutno kliknout na „Přidat IP adresu/rozsah“. Dojde k vyvolání dialogového okna kde je potřeba zanást IP adresu administrátorského připojení, pokud se administrátor nalézá v síti, odkud bude prováděna správa, může využít tlačítka „Vložit moji IP adresu“. Nutné je uvést popis výjimky. Celou akci je potřeba potvrdit tlačítkem „Uložit“.

Přidat IP adresu/rozsah	
*Účet:	fox99.cz
*IP adresa/rozsah:	1.2.3.4 <input type="button" value="Vložit moji IP adresu"/>
*Popis:	adminIP
Aktivovat:	<input checked="" type="checkbox"/>

Obr. 20 Vytvoření výjimky pro IP adresu administrátorské sítě

5.3.2.4 Povolení přístupu pomocí FTP

Společnost ONES chápe nutnost dokonale zabezpečit přístup na server pomocí FTP, z tohoto důvodu je ve výchozím nastavení přístup na server zakázán, musí být provedeno odemknutí FTP účtu, jehož údaje byly zaslány na e-mail administrátora. Tak lze učinit z administrace ONES kliknutím na „Služby“, „FTP“ a „Správa FTP účtů“.



Obr. 21 Správa FTP účtu u společnosti ONES [1]

Na této stránce je potřeba kliknout na zámek u primárního účtu, po kliknutí bude uživatel dotázán, zda má být účet skutečně odemčen, akci je potřeba potvrdit. Po dobu 12 hodin je umožněno pracovat na aplikaci s využitím FTP.

Po provedení výše uvedených nastavení byly veškeré služby nakonfigurovány a je možné na server nahrát aplikaci „Webový e-shop a jeho správa“.

5.3.3 Nahrání aplikace na server

Způsobů jak aplikaci na server nahrát, je několik, pro tuto práci poslouží například známý program *Total Commander* [62]. Vytvořená aplikace byla umístěna na DVD, které je součástí práce (viz PŘÍLOHA P I). Na tomto médiu lze nalézt nejen zdrojové kódy, ale i webový server s již nainstalovanou aplikací. Samotná aplikace byla archivována do souboru „\www.zip“. Obsah tohoto archivu je nutné rozbalit do požadovaného adresáře na serveru. Z předchozích kapitol byl získán přístup na server protokolem FTP, nyní je potřebné přístupové údaje zanechat do programu *Total Commander* tak, aby bylo možné aplikaci nahrát. Proces nahrávání trvá několik minut, navržený systém má co do velikosti sice kolem

20MB, ale obsahuje velké množství souborů. Po úspěšném nahrání aplikace je nutné provést její konfiguraci, k čemu budou taktéž potřebné údaje z předešlých kapitol.

5.3.4 Konfigurace aplikace při prvním spuštění

Jakmile je proces kopírování dokončen, je možné přejít k dalšímu kroku, a to provedení základního nastavení aplikace, která vyžaduje zadání informací o databázi, servisním e-mailu pro rozesílání informačních zpráv, povolení, nebo zakázání prvků souvisejících s bezpečností, nastavení nacionálu poskytovatele (administrátora), jako poslední je umožněno provést personalizaci celého systému. Pro jednoduchou konfiguraci byl vytvořen předvyplněný formulář se základní konfigurací navrženého systému, aby každý administrátor pochopil, do kterého vstupního pole je potřeba zapsat jakou hodnotu.

Při prvotním spuštění aplikace zadáním URL adresy „`http(s)://example.tld`“, dojde k vyobrazení konfiguračního nástroje aplikace. U každého parametru, mimo údajů administrátora, je přiřazená definice konstanty, vysvětlení významu jednotlivých konstant bylo zahrnuto do přílohy (PŘÍLOHA P II). Do tohoto nástroje je potřeba zaneš veškeré získané údaje z předchozích kapitol, dále je vyžadováno zadat údaje provozovatele aplikace, který je současně i jediným administrátorem systému. Po vyplnění všech konfiguračních polí je nutné změny odeslat.

Pokud jsou všechny informace v pořádku uloženy, dojde k zobrazení celkem tří zelených rámců informující skutečnost, že příslušné údaje byly v pořádku přijaty. V opačném případě, pokud se zobrazí byť pouze jeden červený rám, bude potřebné konfiguraci změnit.

Po úspěšné konfiguraci dojde k automatickému importu dat na databázi. Následně lze z konfiguračního souboru přejít na hlavní stránku aplikace odkazem „Spustit aplikaci“, který byl umístěn na pravé straně od tlačítka „Uložit“. Tlačítko je přístupné, pouze pokud byla veškerá konfigurace v pořádku přijata! Tato operace provede prvotní inicializaci aplikace a dojde ke smazání konfiguračního prostředí z důvodu bezpečnosti.

Těmito kroky byl dokončen proces nasazení aplikace.

5.3.5 Konfigurace při spuštěné aplikaci

Na již inicializované aplikaci lze provést konfiguraci úpravou obsahu souboru „`/xampp/htdocs/www/private/config.php`“. Zde jsou obsaženy definice konstant, které byly popsány v (PŘÍLOHA P II). Jednotlivou editací obsahu těchto definic je možné provést pozdější konfiguraci.

ZÁVĚR

Veškeré body zadání se podařilo úspěšně splnit v plném rozsahu. Literární rešerše byly individuálně zpracovány ke každé kapitole, čímž je čtenář také uveden do konkrétní problematiky. Jako první byl řešen individuální návrh „front-end“ části e-shopu, kde bylo nutné sestavit vhodný grafický vzhled, jehož zpracování bylo uvedeno v kapitole 1 Vzhled internetového obchodu. V souvislosti s celkovou aplikací bylo nutné zpracovat problematiku třídění položek do jednotlivých skupin, která se objevuje v systému jak z pohledu zákazníka, tak i administrátora v „back-end“ části aplikace. Daný problém byl rozebrán v kapitole 2 Třídění položek. Dále došlo na řešení rozšířeného vyhledávání položek. Navržená aplikace byla vytvořena tak, aby bylo možno provádět vyhledávání ve všech částech prezentace firmy, a současně v administrativním rozhraní zjednodušit operace přidávání souvisejících položek. Při návrhu aplikace bylo také dbáno na optimalizaci pro vyhledávače, díky čemuž bylo docíleno k vytvoření přátelských URL adres v rámci celého systému, a také strukturovaného HTML kódu. Problematika vyhledávání a optimalizace pro vyhledávače byla zpracována v kapitole 3 Vyhledávání a optimalizace pro vyhledávače. Při návrhu systému byly využity poznatky organizace *OWASP*, která stanovila požadavky na zabezpečení, konkrétně upravované projektem *ASVS*. Navržená aplikace byla podrobena penetračnímu testování, při kterém se aplikace také zabezpečovala, aby se docílilo požadované úrovně zabezpečení stanovené organizací. Bezpečnosti v práci bylo věnováno nemalé úsilí, které bylo zdokumentováno v kapitole 4 Zabezpečení aplikace. Aby bylo možné aplikaci prakticky využít, musel být vytvořený systém umístěn na cizí server, kde byla aplikace spuštěna. Celkový postup a současně i návod pro umístění aplikace na webový server byl zdokumentován v kapitole 5 Nasazení aplikace. Při návrhu tohoto bodu se autor zamyslel nad možnostmi využití aplikace více organizacemi a provedl implementaci konfiguračního nástroje, který umožňuje spuštění aplikace co nejlépe zjednodušit a zároveň provést personalizaci systému.

Dále byly do aplikace implementovány moduly, které nebyly předmětem zadání. V aplikaci se autor zabíral webovou kalkulačkou společnosti *Cetelem*, díky které si uživatelé mohou spočítat splátky dražšího zboží. V zadání nebylo konkrétně stanoveno do jaké míry má být prezentace, a celkové uživatelské prostředí, zpracováno. Autor se nechal inspirovat několika známými e-shopy společností a vynosnažil se aplikovat příjemné uživatelské prostředí, které nenásilnou formou uživatele informuje o veškerých prováděných činnostech, a dále se snaží zákazníka směřovat až k provedení objednávky. Do aplikace byl implemen-

tován taktéž modul, jenž na základě Administrativního registru ekonomických subjektů (dále jen ARES), se snaží za právnické osoby a podnikatele po zadání identifikačního čísla (IČ) vyplnití nacionály. Škoda je, že tento registr je dosti pomalý, a vyhledání údajů trvá poměrně dlouho. Pro korektní funkci e-shopu bylo nutné vytvořit i mechanismy vkládání položek do košíku, a také kompletní provedení objednávky. Tyto části řeší třídy „Kosik“ a „Formular“. V aplikaci byl také vytvořen oddíl, ve kterém jsou průběžně zobrazovány aktuální položky nacházející se ve slevě.

Autor plánuje do budoucna v budování aplikace pokračovat. Naplánováno je využití plateb platebními kartami, kdy bude potřené zvýšit úroveň zabezpečení aplikace až na nevyšší úroveň, pro zajištění maximálního bezpečí, kterou projekt *OWASP ASVS* prezentuje. Dále je naplánováno rozčlenit produkty podle bližších specifikací parametrů tak, aby bylo možné například elegantně dohledat alternativy zboží, a ne pouze související položky. V plánu je taktéž rozšířit administrativní možnosti aplikace, kde bude možno upravovat relevanci zobrazování slev, a tím dostat položky, kterých se provozovatelé chtějí zbavit, na přední pozice e-shopu, následně i vyhledávačů po provedení indexace. Systém bude v budoucnu napojen na interní skladový systém firmy, ale prozatím není známo, o jaký systém se bude jednat. Pravdou je, že systém nabízí obrovské množství možností, které by bylo možno vyvinout a následně aplikovat na tento projekt.

SEZNAM POUŽITÉ LITERATURY

- [1] ONESOLUTION S.R.O. *ONEbit* [online]. © 2006 – 2014 [cit. 2014-05-20]. Dostupné z: <http://www.onebit.cz/cz/>
- [2] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: průvodce webového programátora*. Vyd. 1. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.
- [3] CKSOURCE. *CKEditor* [online]. © 2014 [cit. 2014-05-20]. Dostupné z: <http://ckeditor.com/>
- [4] TEAGUE, Jason Cranford. *DHTML a CSS pro World Wide Web: praktická vizuální příručka*. Praha: SoftPress, 2005. ISBN 80-864-9777-1.
- [5] ZAKAS, Nicholas C, Jeremy PCPEAK a Joe FAWCETT. *Ajax: profesionálně*. Vyd. 1. Překlad Jiří Koutný. Brno: Zoner Press, 2007. ISBN 978-80-86815-77-0.
- [6] SCHLOSSNAGLE, George. *Pokročilé programování v PHP 5*. Brno: Zoner Press, 2004. ISBN 80-868-1514-5.
- [7] *Jquery* [online]. 2014 [cit. 2014-01-30]. Dostupné z: <http://jquery.com/>
- [8] *PHP* [online]. 2014 [cit. 2014-01-30]. Dostupné z: <http://www.php.net/>
- [9] *GIMP 2.8* [online]. © 2001-2013 [cit. 2014-05-01]. Dostupné z: <http://www.gimp.org/>
- [10] W3C. *World Wide Web Consortium* [online]. © 2014 [cit. 2014-05-21]. Dostupné z: <http://www.w3.org/>
- [11] ORACLE CORPORATION. *MySQL: The world's most popular open source database* [online]. © 2014 [cit. 2014-05-20]. Dostupné z: <http://www.mysql.com/>
- [12] ON DESIGN. *ONDESIGN.CZ – WEBDESIGN A GRAFIKA* [online]. © 2014 [cit. 2014-05-20]. Dostupné z: <http://www.ondesign.cz/>
- [13] CARDOSO, Leonardo. *Rotate-jQuery* [online]. V0.1.6. 17.1.2013 [cit. 2014-05-20]. Dostupné z: <https://github.com/LeonardoCardoso/Rotate-jQuery>
- [14] SEO LINHART S.R.O. *SEO nástroje* [online]. © 2010 [cit. 2014-05-20]. Dostupné z: www.seonastroje.cz/
- [15] *Wikipedie – otevřená encyklopedie: Search Engine Optimization* [online]. 2014, 8.5.2014 [cit. 2014-05-20]. Dostupné z: http://cs.wikipedia.org/wiki/Search_Engine_Optimization

- [16] Pokročilé stránkování v PHP. *Mikuv weblog* [online]. © 2010 [cit. 2014-05-20]. Dostupné z: <http://mike.treba.cz/pokrocile-strankovani-php/>
- [17] THE APACHE SOFTWARE FOUNDATION. *Apache HTTP Server Version 2.4: Apache Module mod_rewrite* [online]. © 2014 [cit. 2014-05-27]. Dostupné z: http://httpd.apache.org/docs/2.4/mod/mod_rewrite.html
- [18] LOCKHART, Josh. *Slim* [online]. © 2012, 22.9.2013 [cit. 2014-05-20]. Dostupné z: <http://www.slimframework.com/>
- [19] ROUBÍČEK, Aleš. ZDROJÁK.CZ. *Zdroják.cz: Open Graph a jeho nasazení* [online]. [20.5.2014] [cit. 2014-05-20]. Dostupné z: <http://www.zdrojak.cz/clanky/open-graph-a-jeho-nasazeni/>
- [20] OWASP. *Open Web Applications Security Project* [online]. 14 May 2014 [cit. 2014-06-03]. Dostupné z: https://www.owasp.org/index.php/About_OWASP
- [21] OWASP. *Application Security Verification Standard Project* [online]. 2013, 22 Apr 2014 [cit. 2014-06-03]. Dostupné z: https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project#tab=Home
- [22] OWASP. *Code Review Guide V2.0* [online]. 22 Aug 2013 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/Category:OWASP_Code_Review_Project
- [23] OWASP FOUNDATION INC. *Fail securely* [online]. 2009 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/Fail_securely
- [24] OWASP FOUNDATION INC. *Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection (OWASP-EN-002)* [online]. 2014 [cit. 2014-05-20]. Dostupné z: [https://www.owasp.org/index.php/Testing_for_Weak_SSL/TSL_Ciphers,_Insufficient_Transport_Layer_Protection_\(OWASP-EN-002\)](https://www.owasp.org/index.php/Testing_for_Weak_SSL/TSL_Ciphers,_Insufficient_Transport_Layer_Protection_(OWASP-EN-002))
- [25] Microsoft Developer Network: Parent-Child Dimensions [online]. © 2014 [cit. 2014-05-20]. Dostupné z: <http://msdn.microsoft.com/en-us/library/ms174846.aspx>
- [26] OWASP FOUNDATION INC. *Testing for User Enumeration and Guessable User Account (OWASP-AT-002)* [online]. 2012 [cit. 2014-05-20]. Dostupné z: [https://www.owasp.org/index.php/Testing_for_User_Enumeration_and_Guessable_User_Account_\(OWASP-AT-002\)](https://www.owasp.org/index.php/Testing_for_User_Enumeration_and_Guessable_User_Account_(OWASP-AT-002))

- [27] OWASP FOUNDATION INC. *Testing for default credentials (OWASP-AT-003)* [online]. 2013 [cit. 2014-05-20]. Dostupné z: [https://www.owasp.org/index.php/Testing_for_default_credentials_\(OWASP-AT-003\)](https://www.owasp.org/index.php/Testing_for_default_credentials_(OWASP-AT-003))
- [28] SNYDER, Chris, Tom MYER a Michael G SOUTHWELL. *Pro PHP security: from application security principles to the implementation of XSS defenses*. 2nd ed. New York: Distributed to the book trade worldwide by Springer Science Business Media, 2010, xviii, 345 p. Expert's voice in open source. ISBN 14-302-3318-4
- [29] OWASP FOUNDATION INC. *Session fixation* [online]. 2011 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/Session_fixation
- [30] OWASP FOUNDATION INC. *Forced browsing* [online]. 2009 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/Forced_browsing
- [31] OWASP FOUNDATION INC. *Broken Access Control* [online]. 2010 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/Broken_Access_Control
- [32] OWASP FOUNDATION INC. *Insecure Direct Object Reference Prevention Cheat Sheet* [online]. 2014 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet
- [33] OWASP FOUNDATION INC. *OWASP Testing Guide* [online]. 2008 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf
- [34] OWASP FOUNDATION INC. *Use PDO or ORM* [online]. 2014 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet#Use_PDO_or ORM
- [35] OWASP FOUNDATION INC. *File inclusion vulnerability* [online]. 2014 [cit. 2014-05-20]. Dostupné z: [https://www.owasp.org/index.php/Testing_for_LDAP_Injection_\(OWASPDV-006\)](https://www.owasp.org/index.php/Testing_for_LDAP_Injection_(OWASPDV-006))
- [36] OWASP FOUNDATION INC. *Shell Injection* [online]. 2014 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet#Shell_Injection

- [37] SANS. *Survey on Application Security Programs* [online]. 10. 2. 2014 [cit. 2014-05-20]. Dostupné z: <http://software-security.sans.org/blog/2014/02/10/survey-on-application-security-programs-webcast-paper>
- [38] INTERNATIONAL DATA GROUP. *Computer World: Zabezpečení aplikací se musí zlepšit* [online]. 26.3.2012 [cit. 2014-06-04]. Dostupné z: <http://computerworld.cz/securityworld/zabezpeceni-aplikaci-se-musi-zlepsit-44776>
- [39] ZEMEK, Lukáš. *Bezpečnost webových aplikací*. Zlín, 2012. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce doc. Ing. Martin Sysel, Ph.D.
- [40] SANS. *Survey on Application Security Programs* [online]. © 2013 [cit. 2014-05-20]. Dostupné z: <http://wizardinternetsolutions.com/articles/web-programming/dynamic-multilevel-css-menu-php-mysql>
- [41] ORACLE. *MySQL: Using Triggers* [online]. © 2014 [cit. 2014-05-20]. Dostupné z: <http://dev.mysql.com/doc/refman/5.0/en/triggers.html>
- [42] HIBERNATE. *What is Object/Relational Mapping?* [online]. [2013] [cit. 2014-05-20]. Dostupné z: <http://hibernate.org/orm/what-is-an-orm/>
- [43] THOMPSON, Craig. *Qtip2: The powerfull tooltip* [online]. © 2013 [cit. 2014-05-20]. Dostupné z: <http://qtip2.com/>
- [44] LEMOND, Stan. *Jgrowl v1.2.12* [online]. 13.4.2014 [cit. 2014-05-20]. Dostupné z: <https://github.com/stanlemon/jGrowl>
- [45] HAZARD DEVELOPER. *Composide: jQuery MsgBox* [online]. 12 Dec 2013 [cit. 2014-05-20]. Dostupné z: <http://s3.envato.com/files/293712/index.html>
- [46] PEJŠA, Jan. *Co je Cross-site scripting jak mu předcházet*. *Zdrojak.cz* [online]. 5.3.2009 [cit. 2014-05-20]. Dostupné z: <http://www.zdrojak.cz/clanky/co-je-xss-jak-mu-predchazet/>
- [47] GOOGLE. *Google: Custom Search Engine* [online]. © 2014 [cit. 2014-05-20]. Dostupné z: <https://www.google.cz/cse/?hl=cs>
- [48] JANOVSKEÝ, Dušan. *Jak Psát Web: Hledání na vlastním serveru* [online]. 4.5.2014 [cit. 2014-05-20]. Dostupné z: <https://www.google.cz/cse/?hl=cs>
- [49] Seznam.cz [online]. © 1996-2014 [cit. 2014-06-06]. Dostupné z: <http://www.seznam.cz/>

- [50] SOUHRADA, Ernie. *InnoDB Full-text Search in MySQL 5.6*. MySQL Performance Blog [online]. 9.4.2014, č. 3, s. 3 [cit. 2014-05-20]. Dostupné z: <http://www.mysqlperformanceblog.com/2013/07/31/innodb-full-text-search-in-mysql-5-6-part-3/>
- [51] *SeoRadce.cz: otázky a odpovědi ohledně SEO* [online]. [2013] [cit. 2014-05-20]. Dostupné z: <http://www.seoradce.cz/>
- [52] OWASP FOUNDATION INC. *Error Handling* [online]. 2007 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/Error_Handling
- [53] UNDERWOOD, Paul. *Disable HTTP Cache With PHP* [online]. [2014] [cit. 2014-05-20]. Dostupné z: <http://www.paulund.co.uk/disable-http-cache-with-php>
- [54] ZONER SOFTWARE, a.s. *INTERVAL.CZ: Jak si vybrat certifikační autoritu* [online]. 2003 [cit. 2014-05-20]. Dostupné z: <http://interval.cz/clanky/jak-si-vybrat-certifikacni-autoritu/>
- [55] OWASP FOUNDATION INC. *Cross Site Tracing* [online]. 4.2.2013 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/Cross_Site_Tracing
- [56] GOLEM TECHNOLOGIES: *HTTP Response Splitting* [online]. [2013] [cit. 2014-05-20]. Dostupné z: <https://www.golemtechnologies.com/articles/http-response-splitting>
- [57] OWASP FOUNDATION INC. *Path Traversal* [online]. 27.5.2009 [cit. 2014-05-20]. Dostupné z: https://www.owasp.org/index.php/Path_Traversal
- [58] *Roundcube.net: Open Source Webmail Project* [online]. © 2014 [cit. 2014-05-20]. Dostupné z: <http://roundcube.net/>
- [59] OWASP FOUNDATION INC. *OWASP ASVS 2013 Beta _v1.0* [online]. 27.5.2009 [cit. 2014-05-20]. Dostupné z: http://sourceforge.net/projects/owasp/files/ASVS/OWASP%20ASVS%202013%20Beta%20_v1.0.pdf/download
- [60] *Jquery User Interface* [online]. © 2014 [cit. 2014-05-20]. Dostupné z: <http://jqueryui.com/>
- [61] *PhpMyAdmin* [online]. © 2003 - 2014 [cit. 2014-05-20]. Dostupné z: http://www.phpmyadmin.net/home_page/index.php
- [62] *Total Commander* [online]. © 1995-2014, 30.4.2014 [cit. 2014-06-12]. Dostupné z: <http://www.ghisler.com/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASVS	Application Security Verification Standard
CA	Certifikační Autorita
CC	Creative Commons
CSRF	Cross Site Request Forgery
DOM	Document Object Model
FTP	File Transfer Protocol
FTS	Full Text Search
GC	Garbage Collector
GET	Metoda protokolu HTTP
HTML	Hypertextový značkovací jazyk (HyperText Markup Language)
JS	Programovací jazyk zpracovávaný na straně klienta (JavaScript)
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
MySQL	Druh relačního databázového systému
MySQLi	Rozšíření PHP o možnosti komunikace s relační databází MySQL
ONES	ONEsolution s.r.o
ORM	Objektově relační mapování (Object Relational Mapping)
OWASP	Open Web Application Security Project
PHP	Hypertextový preprocesor (Hypertext Preprocessor)
POST	Metoda protokolu HTTP
SEM	Search Engine Marketing
SEO	Search Engine Optimization

SID	Identifikátor relace (Session ID)
SQL	Strukturovaný dotazovací jazyk (Structured Query Language)
SSL	Secure Sockets Layer
TLD	Top Level Domain
TLS	Transport Layer Security
URL	Jednotná adresa zdroje (Uniform Resource Locator)
XSS	Cross Site Scripting

SEZNAM OBRÁZKŮ

Obr. 1 Rozvržení úzkého grafického návrhu	14
Obr. 2 Rozvržení širokého grafického návrhu	15
Obr. 3 Rozvržení návrhu pro administraci.....	16
Obr. 4 Standardní řešení menu, generováno zdrojem [61]	18
Obr. 5 Původní databázové rozdělení do skupin, generováno zdrojem [61]	19
Obr. 6 Struktura a náhled uspořádání dat tabulky „Skupiny“, generováno zdrojem [61]	20
Obr. 7 Zanesení položky do skupin	22
Obr. 8 Zpracování menu s možností editace.....	28
Obr. 9 – Ukázka úspěšného uložení editované skupiny modulem <i>jQuery jGrowl</i> [44].	28
Obr. 10 - Ukázka neúspěšného uložení editované skupiny modulem <i>jQuery jGrowl</i> [44].	29
Obr. 11 – Roletový vyhledávací našeptávač a zasílané požadavky na server	39
Obr. 12 Výsledek implementace <i>Open Graph</i> protokolu [19]	46
Obr. 13 Úroveň OWASP ASVS [59]	48
Obr. 14 – Správa MySQL databází na serveru společnosti ONES [1].....	73
Obr. 15 Vytvoření databáze pro webovou aplikaci u společnosti ONES [1]	73
Obr. 16 Příklad vytvoření <i>MySQL</i> sekundárního účtu u společnosti ONES [1].....	74
Obr. 17 Správa e-mailových účtu u společnosti ONES [1]	74
Obr. 18 Vytvoření nového e-mailového účtu robota u společnosti ONES [1].....	75
Obr. 19 Správa povolených IP pro přístup na server pomocí FTP u společnosti ONES [1]	75
Obr. 20 Vytvoření výjimky pro IP adresu administrátorské sítě	76
Obr. 21 Správa FTP účtu u společnosti ONES [1]	76

SEZNAM TABULEK

Tabulka 1 Minimální konfigurace serveru.....	70
Tabulka 2 Parametry webhostingu BUSSINES [1].....	72

SEZNAM PŘÍLOH

PŘÍLOHA P I: ZDROJOVÉ KÓDY APLIKACE

PŘÍLOHA P II: KONFIGURAČNÍ MOŽNOSTI

PŘÍLOHA P I: ZDROJOVÉ KÓDY APLIKACE

Veškeré zdrojové jsou k dispozici v příloženém DVD.

PŘÍLOHA P II: KONFIGURAČNÍ MOŽNOSTI

Konstanta	Příklad nastavení	Hodnota
SQL_HOST	127.0.0.1	Hostitel databáze, pokud je využívána externí databáze mimo server, musí být uvedena cesta zde.
SQL_USERNAME	fox99.cz.1	Uživatelské jméno pro přihlášení na databázi, je přiděleno poskytovatelem ONES po provedení kroku kapitoly 5.3.2.1. Je nutné aby, tomuto účtu bylo povoleno spouštění procedur, funkcí a triggerů.
SQL_PASSWORD	-	Heslo pro přístup na databázi. Heslo bylo vymyšleno v kapitole 5.3.2.1.
SQL_DBNAME	fox99cz2	Nastavení jména databáze, jméno databáze je přiděleno poskytovatelem ONES po provedení kroku kapitoly 5.3.2.1.
__SMTP_HOST__	smtp.fox99.cz	Adresa serveru odchozí pošty SMTP.
__SMTP_SECURE__	(prázdné), ssl, tsl	Nastavení způsobu šifrování odesílaného emailu.
__SMTP_PORT__	465,25	Port pro server odchozí pošty.
__SMTP_AUTH__	Ano, Ne	Požaduje server pro zasílání emailů autentizaci? Pokud je vyžadována, musí být vyplněny pole __SMTP_USERNAME__ i __SMTP_PASSWORD__
__SMTP_USERNAME__	tester@fox99.cz	Přihlašovací jméno do e-mailového účtu. Toto jméno bylo vytvořeno v kapitole 5.3.2.2.
__SMTP_PASSWORD__	-	Heslo pro přístup do e-mailového účtu.
__NAME_HEADER__	FOX - Kancelářská Technika	Hlavička, se kterou jsou zasílány emaily. Nejlépe je zde uvést jméno firmy, která aplikaci využívá.
__REPLY_TO__	nasefirma@fox99.cz	V případě provedení odpovědi na příchozí email, bude email zaslán zvolenou adresou. Je doporučeno vložit adresu hlavního emailu

__REPLY_TO_NAME__	FOX - Pavel Hájek	Pojmenování e-mailu pro odpověď.
__USE_HTTP_REFERERER_SECURITY__	Ano, Ne	Povolení využívání proměnné HTTP_REFERERER jako doplněk zabezpečení. Je doporučeno ponechat nastavené.
__USE_SSL__	Ano, Ne	Jestliže administrátor, popřípadě provozovatel, disponuje podepsaným certifikátem ověřenou certifikační autoritou, měla by být tato volba zvolena.
__ENABLED_HOSTS__	example.tld	Zde je nutné uvést povolené domény, ze kterých mohou být zpracovávány požadavky. Pokud je na aplikaci směřováno více domén, musí být zde uvedené všechny, oddělené čárkou bez mezer.
__ROOT__	(žádná),beta	Jedná se o deklaraci kořenu aplikace. V konfiguračním prostředí je vyplněn automaticky a není potřeba nic dále řešit.
__MAIN_TITLE__	FOX - Kancelářská Technika	Hlavní titulek obchodu, je doporučeno jej volit v souladu s názvem firmy.
__MAIN_LOGO__	/img/logo.png	Absolutní cesta, kde bylo umístěno logo aplikace.
__DESCRIPTION__	Popis firmy	Zde by měl být uveden krátký popis činnosti firmy. Tento popis bude zobrazen ve vyhledávacích a popřípadě na Facebooku.
__KEYWORDS__	tiskárna, kancelářské, ...	Zde by měla být uvedena klíčová slova oboru činnosti firmy.
__SHORTCUT_ICON__	/img/favicon.gif	Cesta k miniatuře, která bude zobrazena v panelu prohlížeče.
__USE_OPENGRAPH__	Ano, Ne	Volba, zda má být společnost prezentována na sociálních sítích stejně jako ve vyhledávacích.
__LOGO_OG__	/img/fb_logo.jpg	Absolutní cesta k logu firmy, které bude zobrazeno na sociálních sítích. Je vyžadována minimální velikost (200x200)px.
__FACEBOOK_ADMIN_ID__	-	Pokud má aplikace profil na sociální síti Facebook, zde by měl být uveden jeho identifikátor.