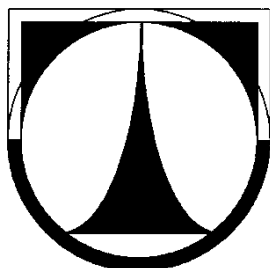


TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY, INFORMATIKY A MEZIOBOROVÝCH STUDIÍ



Ústav nových technologií a aplikované informatiky

E-shop konfigurator pro CMS Joomla

E-shop configuration tool for Joomla CMS

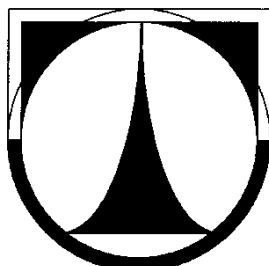
Bakalářská práce

Karel Bartůněk

Květen 2011

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY, INFORMATIKY A MEZIOBOROVÝCH STUDIÍ



Ústav nových technologií a aplikované informatiky

Studijní program

B2612 Elektrotechnika a informatika

Obor Informatika a Logistika

E-shop konfigurator pro CMS Joomla

E-shop configuration tool for Joomla CMS

Bakalářská práce

Karel Bartůněk

Vedoucí bakalářské práce: Mgr. Jiří Vraný, Ph.D.

Počet stran: 33

Počet obrázků:

Počet tabulek: 0

Počet příloh: 1

Místo pro oficiální zadání BP

Téma

E-shop konfigurátor pro CMS Joomla

Anotace

Bakalářská práce se zabývá problematikou návrhu a vývoje komponent pro redakční systém Joomla. Na úvod jsou popsány základní vlastnosti, možnosti a historie tohoto systému. Dále je pozornost věnována implementaci návrhového vzoru Model-View-Controller, využití technologií XHTML, PHP, MySQL a JavaScript.

Následně jsou tyto znalosti a technologie využity k vytvoření vlastní, uživatelsky přívětivé komponenty, která bude moci být nasazena v e-shopech založených na CMS Joomla, doplněných o komponentu VirtueMart poskytující funkce e-shopu a která slouží jako dynamický vyhledávač produktů v e-shopech.

Theme

E-shop configuration tool for Joomla CMS

Annotation

The Bachelor thesis deals with the issue of design and development of components for Joomla Content Management System. The introduction describes the basic features, possibilities and history of this system. Further attention is paid to the implementation of Model-View-Controller design pattern as well as to the use of XHTML, PHP, MySQL and JavaScript technologies.

Consequently, this knowledge and technologies are used for creation of an own, user-friendly component, which is to be employed in Joomla CMS-based e-shops, enhanced with VirtueMart component that provides e-shop functions and that serves as a dynamic search engine of products in e-shops.

Klíčová slova

Joomla, VirtueMart, komponenta, modul, e-shop.

Key words

Joomla, VirtueMart, component, module, e-shop.

Zpracovatel: TU v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií
Ústav nových technologií a aplikované informatiky
Dokončeno: 2011

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce.

V dne

.....

podpis

Poděkování

Na tomto místě bych rád poděkoval Mgr. Jiřímu Vranému, Ph.D. za vedení mé bakalářské práce.

Dále bych rád poděkoval především rodičům a všem mým blízkým, kteří mě po celou dobu studia na FM TUL podporovali a byli mi po dobu studia pevnou oporou.

Obsah

Klíčová slova	5
Přehled použitých zkratk, symbolů	11
1. Úvod	12
1.1. Cíle bakalářské práce	12
2. CMS Joomla!	13
2.1. Historie systému Joomla!	13
2.2. Licence Joomla!	13
2.3. Požadavky systému a vytvořené aplikace	13
3. Použité technologie	14
3.1. XHTML.....	14
3.2. JavaScript	14
3.3. MySQL.....	14
3.4. AJAX (Asynchronous JavaScript and XML).....	14
3.5. jQuery	14
3.6. PHP.....	15
4. Rozšíření pro CMS Joomla!	15
4.1. Komponenta	15
4.2. Modul.....	15
4.3. Plug-in	15
5. Model-View-Controller	15
5.1. Model.....	16
5.2. View	16
5.3. Controller.....	16
6. Vývoj komponenty	17

6.1.	Adresářová struktura	17
6.2.	Souborový systém	17
6.3.	Databáze	19
6.4.	jQuery v administraci.....	20
6.5.	Komentáře	21
7.	Vývoj modulu	21
7.1.	Úprava <head>.....	24
8.	Administrační část (back-end)	25
9.	Uživatelská část (front-end).....	27
9.1.	Prezentace front-endu.....	29
10.	Instalace	29
10.1.	Uživatel.....	29
10.2.	Vývojář	29
11.	Závěr	31
12.	Zdroje informací	32
Příloha – A obsah přiloženého CD		33

Seznam obrázků

Obrázek 1: Schéma softwarové architektury MVC	16
Obrázek 2: Ukázka adresářové struktury	17
Obrázek 3: Ukázka obsahu souboru index.html.....	19
Obrázek 4: Struktura tabulky #__qc_templates	19
Obrázek 5: Tabulka #__qc_categories	19
Obrázek 6:Tvorba tabulek	20
Obrázek 7: konfliktní mód	20
Obrázek 8: nekonfliktní mód	21
Obrázek 9: Ukázka komentáře	21
Obrázek 10: XML soubor s parametry a výstup v administraci.....	22
Obrázek 11: SQL v konfiguračním souboru	22
Obrázek 12: Vložení souboru s češtinou	22
Obrázek 13: Jazykový soubor	23
Obrázek 14: struktura adresáře modulu	23
Obrázek 15: Vložení odkazů na soubory	24
Obrázek 16: Sekce <head>	25
Obrázek 17: Hlavní stránka administračního rozhraní Joomla!.....	25
Obrázek 18: Hlavní stránka komponenty	26
Obrázek 19: Úprava šablony	26
Obrázek 20: Chybová hláška	27
Obrázek 21: Uživatelské rozhraní.....	28
Obrázek 22: Instalace rozšíření	29
Obrázek 23: Ukázka instalačního XML souboru	30

Přehled použitých zkratk, symbolů

AJAX	Asynchronous JavaScript and XML	Asynchronní JavaScript a XML
CMS	Central Management System	System pro správu obsahu
GNU GPL	GNU General Public License	Všeobecná veřejná licence
PHP	Hypertext Preprocessor	Hypertextový preprocesor
HTML	HyperText Markup Language	
CSS	Cascading Style Sheets	Kaskádové styly

1. Úvod

Bakalářská práce se zabývá tvorbou komponenty a modulu pro jeden z nejvíce používaných systémů pro tvorbu a správu obsahu (CMS - Content Management System) profesionálních internetových či intranetových stránek – systém Joomla.

Tento systém si získal velké množství příznivců a počet instalací díky své propracovanosti. Základní instalace obsahuje funkce jako SEO-friendly URL, caching, indexaci stránek, RSS, tisknutelné verze stránek, zobrazování novinek, blogy, diskusní fóra, hlasování, kalendář, vyhledávání v rámci webserveru a také například podporuje změny vzhledu. Pokud nám chybí nějaká specifická funkce, je možné doinstalovat či naprogramovat rozšiřující komponentu, která danou funkčnost doplní. Základní systém můžeme rozšířit například o fotogalerii, internetový obchod, možnost vícejazyčné verze webu a tak dále.

1.1. Cíle bakalářské práce

Cílem této bakalářské práce je seznámit se s objektově orientovaným programováním komponent pro CMS Joomla!. Obecně bude popsán vývoj vlastní komponenty a modulu od návrhu, přes testování až po její samotnou implementaci. Podrobněji bude popsána má komponenta, která bude umožňovat dynamickou tvorbu šablon pro vyhledávání zboží v internetovém obchodě na základě zvolených parametrů a bude testována na reálném a zaběhnutém e-shopu s výpočetní technikou.

2. CMS Joomla!

2.1. Historie systému Joomla!

Samotná Joomla vznikla ze svého předchůdce – redakčního systému Mambo. Mambo bylo vyvíjeno v od roku 2000 a od té doby získalo mnoho ocenění jako je například “Best Open Source Solution” či “Best of Show - Total Industry Solution”. V roce 2005 však vnitřní rozepře ve Výboru pro řízení Mambo způsobily odchod základu vývojářského týmu (tzv. Core Team) a tento tým následně začíná vystupovat pod názvem Open Source Matters. V září roku 2005 tedy vychází verze systému Joomla! 1.0 pod hlavičkou OSM. Od té doby Joomla! prošla mnoha fázemi vývoje, od verze 1.0 se za šest let vývoje dostala k verzi 1.6, která vyšla v lednu 2011.

2.2. Licence Joomla!

Systém je licencován pod GNU GPL, je šířen zdarma a práva vlastní společnost Open Source Matters. Licence GNU GPL nám umožňuje používat systém zdarma jak pro osobní účely, tak je možnost jej nasadit i v komerční sféře, můžeme jej dále upravovat, kopírovat či distribuovat a to i za poplatek. V případě, že upravený systém poskytneme k užívání dalšímu subjektu, je nutné uveřejnit jeho zdrojové kódy. Pokud upravujeme systém pouze pro vlastní použití, licence nám dává právo úpravy nezveřejňovat.

V této práci vznikne rozšíření systému – komponenta a modul. I přesto, že oba budou využívat vnitřní funkce frameworku Joomla, tak na takto vytvořené rozšíření se již licence Joomla nevztahuje a určit si ji může vývojář sám.

2.3. Požadavky systému a vytvořené aplikace

Joomla! je vytvořena skriptovacím programovacím jazykem PHP. Ke svému běhu tedy potřebuje podporu PHP, databázový systém MySQL a webový server Apache. Při vývoji mé komponenty a modulu pracuji s Joomla verze 1.5.20. Jelikož se jedná o komponentu, která má umožňovat vyhledávání zboží na základě zvolených parametrů, je nutné mít nainstalovanu komponentu VirtueMart, poskytující funkce e-shopu. Tu jsem měl při vývoji nainstalovanu ve verzi 1.1.5. Na uživatelské stanici je nutné mít nainstalovaný moderní

internetový prohlížeč (např. Chrome, Opera, Firefox, IE apod.), povolený JavaScript, aplikace totiž využívá AJAX (Asynchronous JavaScript and XML) a funkce knihovny jQuery verze 1.6.0.

3. Použité technologie

3.1. XHTML

XHTML je značkový jazyk pro tvorbu hypertextových dokumentů v prostředí WWW vyvinutý mezinárodním konsorciem W3C.

3.2. JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, používaný jako interpretovaný programovací jazyk pro WWW stránky.

3.3. MySQL

MySQL je multiplatformní relační databázový systém. Komunikace s ním probíhá pomocí jazyka SQL.

3.4. AJAX (Asynchronous JavaScript and XML)

Cílem bylo vytvořit interaktivní komponentu, proto jsem zvolil tuto technologii, díky které je možné měnit obsah stránky aniž by se musela celá znovu načítat. Umožňuje okamžitou komunikaci mezi prohlížečem a serverem a tak je docíleno průběžného obnovování výsledků vyhledávání mezi zbožím.

3.5. jQuery

jQuery je JavaScriptová knihovna (sada funkcí) uvolněná pod duální licenci MIT a GPL, která umožňuje a velice ulehčuje interakci mezi JavaScriptem a HTML. Má velké množství funkcí, umožní nám jednoduše pracovat s objekty DOM, manipulovat s CSS, vytvářet animace, reagovat na události atd. V mé aplikaci je jQuery využito k práci s AJAX, umožnilo mi vytvořit posuvník s cenou díky utilitě Slider (obrázek), či vykreslit strom kategorií pomocí utility Treeview.

3.6. PHP

PHP je skriptovací programovací jazyk. Nejčastěji se používá při tvorbě internetových aplikací s technologiemi HTML, XHTML. Skripty jsou prováděny na straně serveru.

4. Rozšíření pro CMS Joomla!

Joomla! umožňuje rozšířit svou funkčnost pomocí třech typů rozšíření, které mají své specifické vlastnosti – komponenty, moduly, pluginy (v mé aplikaci využívám komponentu pro správu šablon vyhledávání a modul pro výpis vyhledávacího formuláře).

4.1. Komponenta

Komponenta je základní typ rozšíření pro tento systém. Je rozdělena na dvě části, back-end a front-end. Back-end komponenty je přístupný v administraci redakčního systému a slouží administrátorovi například k vytváření, editaci obsahu webových stránek, či může plnit jakoukoliv jinou funkci, která pro ni byla naprogramována. Front-end se stará naopak o zobrazení dat na straně uživatele. Joomla byla navržena tak, že na každé stránce se může zobrazit pouze jedna komponenta.

4.2. Modul

Oproti komponentám je jich možné zobrazit více na jedné stránce. Typickými moduly jsou například menu, ankety nebo také formuláře. Starají se o to, jak a kde se vypíší data vytvořená nějakou komponentou.

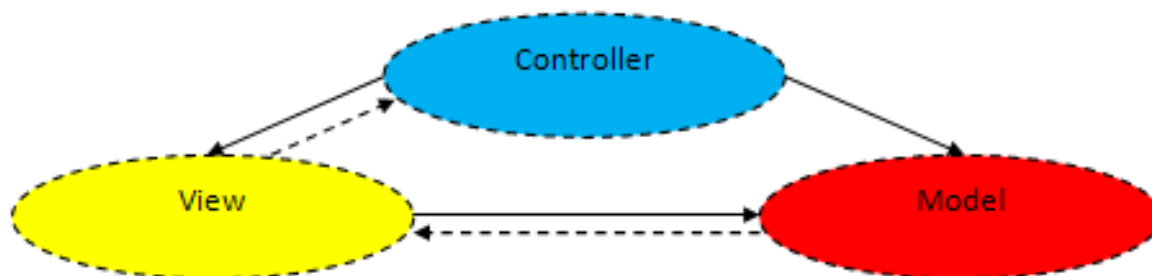
4.3. Plug-in

Plug-in se používá v případě, že potřebujeme vložit na stránku krátký kus zdrojového kódu. Často jsou plug-iny používány k formátování výstupu komponenty nebo modulu, například zvýrazňování klíčových slov.

5. Model-View-Controller

Model-View-Controller (zkráceně MVC) je softwarová architektura používaná ke strukturování zdrojového kódu aplikací, která rozděluje datový model, uživatelské rozhraní a

řídící logiku do tří oddělených částí. Následně je možné upravovat každou část zvlášť s minimálním dopadem na části ostatní. Velkou výhodou využití tohoto vzoru je usnadnění testování výsledného software díky přehlednosti kódu. Využití této struktury je oceňováno hlavně v případě, kdy na určitém projektu pracuje více lidí. V Joomla je MVC vzor implementován použitím tří tříd: JModel, JView a JController.



Obrázek 1: Schéma softwarové architektury MVC

5.1. Model

Model je část komponenty, která zajišťuje přístup k datům a manipulaci s nimi. V mé komponentě bude model například zajišťovat ukládání, editaci a mazání šablon pro vyhledávání.

5.2. View

View má na starost převod dat, která získá z modelu, do podoby vhodné k prezentaci pro uživatele. U webových aplikací se většinou jedná HTML stránku.

5.3. Controller

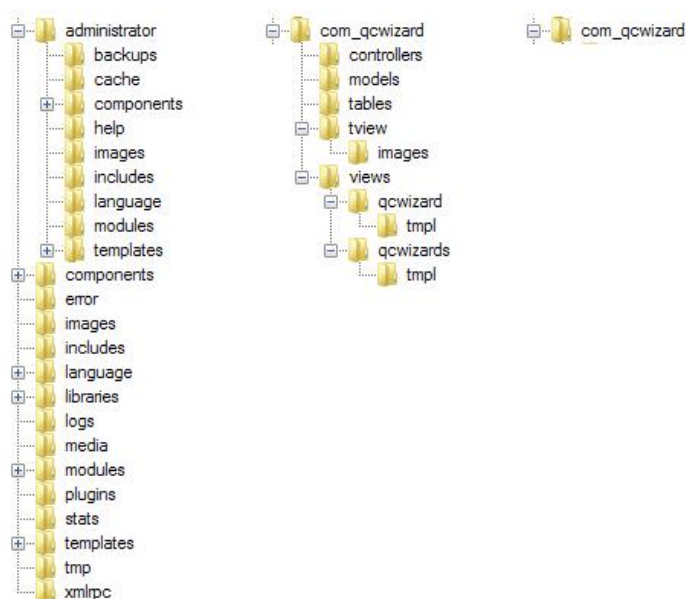
Controller je odpovědný za zpracování akcí uživatele, řídí tok událostí v programu.

6. Vývoj komponenty

6.1. Adresářová struktura

Při vývoji komponenty se musíme řídit pevně daným schématem pojmenovávání jejích adresářů a souborů. Pro každou komponentu je vyžadováno unikátní jméno bez mezer. Pro mou komponentu jsem zvolil název QC Wizard. Celá komponenta je rozdělena do dvou adresářů, jejichž jméno je uvozeno prefixem `com_`, budeme tedy pracovat s adresáři `com_qcwizard`. První z těchto adresářů bude sloužit pro front-end část komponenty. Vytvoříme ho ve složce `components`. Druhý adresář vytvoříme ve složce `administrator/components` a bude nám sloužit pro back-end.

Na obrázku č.2 uvádím adresářovou strukturu Joomla!. První strom je root adresář celého webu, druhé dva jsou námi výše vytvořené adresáře pro back-end a front-end sekci komponenty.



Obrázek 2: Ukázka adresářové struktury

6.2. Souborový systém

Pro mou komponentu jsem musel vytvořit následující seznam souborů, z nichž každý má určitou funkci.

- `/administrator/components/com_qcwizard/controller.php` – základní controller
- `/administrator/components/com_qcwizard/index.html`
- `/administrator/components/com_qcwizard/install.sql` – soubor s MySQL příkazy, díky nimž se vytvoří potřebné tabulky v databázi
- `/administrator/components/com_qcwizard/qcwizard.php` – vstupní bod komponenty
- `/administrator/components/com_qcwizard/uninstall.sql` – soubor s MySQL příkazy, které zajistí odstranění dat a tabulek při odinstalování komponenty ze systému
- `/administrator/components/com_qcwizard/controllers/index.html`
- `/administrator/components/com_qcwizard/controllers/qcwizard.php`
- `/administrator/components/com_qcwizard/models/index.html`
- `/administrator/components/com_qcwizard/models/qcwizard.php` – model, který zajišťuje práci s aktuální šablonou
- `/administrator/components/com_qcwizard/models/qcwizards.php` – tento model obsahuje funkce pro práci se seznamem šablon
- `/administrator/components/com_qcwizard/tables/index.html`
- `/administrator/components/com_qcwizard/tables/qcwizard.php` – obsahuje definici třídy, která představuje tabulku v databázi
- `/administrator/components/com_qcwizard/views/qcwizard/index.html`
- `/administrator/components/com_qcwizard/views/qcwizard/view.html.php`
- `/administrator/components/com_qcwizard/views/qcwizard/tmpl/form.php`
- `/administrator/components/com_qcwizard/views/qcwizard/tmpl/index.html`
- `/administrator/components/com_qcwizard/views/qcwizards/index.html`
- `/administrator/components/com_qcwizard/views/qcwizards/view.html.php`
- `/administrator/components/com_qcwizard/views/qcwizards/tmpl/default.php`
- `/administrator/components/com_qcwizard/views/qcwizards/tmpl/index.html`
- `/components/com_qcwizard/qcwizard.php`

V seznamu se nám vyskytuje několik souborů, které se jmenují `index.html`. Ty mají za úkol se zobrazit, pokud se uživatel pokouší vstoupit do adresáře, do kterého nechceme, aby se dostal. Některé servery jsou totiž nastaveny tak, že vypisují obsah adresáře, což může být

do jisté míry bezpečnostní riziko. Použitím index.html tedy zvyšujeme bezpečnost systému. Soubor obsahuje jediný řádek HTML kódu.

```
<html><body bgcolor="#FFFFFF"></body></html>
```

Obrázek 3: Ukázka obsahu souboru index.html

Soubory s názvem view.html.php, soubor form.php a default.php se starají o výpis dat, která zpracovává příslušný model.

Instalační balíček komponenty obsahuje také soubory ve složce /treeview se skripty a styly pro vypsání dynamického stromu kategorií pomocí JavaScriptu.

6.3. Databáze

Systém Joomla využívá databázový systém MySQL. Při vývoji mé komponenty jsem ho tedy též použil. Databázi využívám pro uložení seznamu šablon a jejich vlastností. V databázi nainstalovaného systému Joomla jsem vytvořil dvě tabulky. V první z nich (#__qc_templates) ukládám jména šablon s jejich ID.

	Sloupec	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Extra	Akce							
☐	<u>id</u>	int(11)			Ne		auto_increment								
☐	name	varchar(25)	utf8_general_ci		Ne										

Obrázek 4: Struktura tabulky #__qc_templates

Druhá tabulka (#__qc_categories) slouží k uložení informací o tom, jaké kategorie produktů souvisí s jakou šablonou pro vyhledávání.

	Sloupec	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Extra	Akce							
☐	<u>id</u>	int(11)			Ne		auto_increment								
☐	<u>wizard_id</u>	int(11)			Ne										
☐	<u>category_id</u>	int(11)			Ne										
☐	<u>parent_id</u>	int(11)			Ne										

Obrázek 5: Tabulka #__qc_categories

Automatická instalace za nás vyřeší i vytvoření těchto dvou tabulek v databázi. Do instalačního balíčku musíme přidat textový soubor, který se bude jmenovat install.sql a jeho obsahem budou SQL příkazy pomocí nichž se tabulky vytvoří.

```
CREATE TABLE IF NOT EXISTS `#__qc_categories` (  
  `id` int(11) NOT NULL auto_increment,  
  `wizard_id` int(11) NOT NULL COMMENT 'číslo průvodce',  
  `category_id` int(11) NOT NULL COMMENT 'číslo kategorie, zařazené do průvodce',  
  `parent_id` int(11) NOT NULL COMMENT 'číslo rodiče zařazené kategorie',  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;  
  
CREATE TABLE IF NOT EXISTS `#__qc_templates` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(25) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

Obrázek 6:Tvorba tabulek

6.4. jQuery v administraci

Pro výpis stromu kategorií jsem se rozhodl používat JavaScriptovou knihovnu jQuery. Nastal ale konflikt s jinou knihovnou, která je používána administrací Joomla! – knihovnou Mootools. Proto je nutné pracovat s jQuery v nekonfliktním módu. To zajistíme vložením řádku

```
jQuery.noConflict();
```

Následně používáme místo syntaxe

```
$(document).ready(function(){  
  $("#browser").treeview({  
    collapsed: true,  
    animated: 250,  
    persist: "cookie"  
  });  
});
```

Obrázek 7: konfliktní mód

novou syntaxí, kde nahrazujeme znak \$ řetězcem jQuery:

```
jQuery(document).ready(function() {
    jQuery("#browser").treeview({
        collapsed: true,
        animated: 250,
        persist: "cookie"
    });
});
```

Obrázek 8: nekonfliktní mód

6.5. Komentáře

Pro lepší orientaci a hlavně do budoucna k úpravám zdrojových kódů byly okomentovány některé důležité funkce.

```
/**
 * Metoda ukládající záznam
 *
 * @access public
 * @return boolean True při úspěchu
 */
```

Obrázek 9: Ukázka komentáře

7. Vývoj modulu

O vypsání vyhledávacího formuláře se mi postará jednoduchý modul. Vytvořil jsem si nejzákladnější modul, který se skládá z konfiguračního souboru XML a z PHP souboru s kontrolerem. V konfiguračním XML souboru jsou uloženy informace o modulu. Můj modul má možnost nastavit si několik vlastností, které ovlivňují výsledky vyhledávání produktů v e-shopu.

```
<params>
  <param type="radio" name="sort" label="SORT" description="Umožňuje zvolit defaultní řazení" default="a">
    <option value="a">Od nejlevnějšího</option>
    <option value="b">Od nejdražšího</option>
    <option value="c">Dle názvu</option>
  </param>
  <param type="radio" name="count" label="COUNT" description="Umožňuje zvolit počet produktů ve výpisu" default="20">
    <option value="10">10</option>
    <option value="20">20</option>
    <option value="50">50</option>
  </param>
```

Parametry

▼ Parametry modulu

Přípona třídy modulu	
Zvolit šablonu	Notebooky ▼
Zkratka měny	Kč
Počet sloupců	3
Řazení výsledků	☒ Od nejlevnějšího ☐ Od nejdražšího ☐ Dle názvu
Počet zobrazených	☒ 10 ☐ 20 ☐ 50
Vyhledávat od ceny	0
Vyhledávat do ceny	100000
Povolit ve vyhledávání	☒ Ano ☐ Ne
Je skladem	2-3d.gif
Nedostupný	not_available.gif

Obrázek 10: XML soubor s parametry a výstup v administraci

Všechny parametry jsou napevno uloženy v XML, tak tu nyní vzniká problém, jak načítat seznam šablon vyhledávání, který vytváříme v komponentě. Joomla! však podporuje vkládání SQL dotazů do konfiguračního souboru a tak můžeme dynamicky načítat aktuální seznam šablon.

```
<param name="parent"
      type="sql"
      default="id"
      label="SELECTCOM"
      description="SELECTCOM2"
      query="SELECT id, name FROM #__qc_templates"
      key_field="id"
      value_field="name"
/>
```

Obrázek 11: SQL v konfiguračním souboru

V případě, že bychom chtěli vytvářet multijazyčnou verzi modulu, je možné do balíku připojit soubory s danou lokalizací.

```
<languages>
  <!-- Any language files included with the module -->
  <language tag="cs-CZ">cs-CZ.mod_qcwizard.ini</language>
</languages>
```

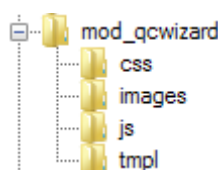
Obrázek 12: Vložení souboru s češtinou

Daný jazyk modulu se poté řídí nastavením jazyka celé administrace. Níže na obrázku je předvedena struktura jazykového souboru. Každý řádek představuje jeden výraz a je složen z klíče a překladu.

```
CURRENCY=Zkratka měny
COLUMNS=Počet sloupců
SORT=Řazení výsledků
MIN=Vyhledávat od ceny
MAX=Vyhledávat do ceny
MAXIMAL=Nastavuje se max. cena, do které se vyhledává
MINIMAL=Nastavuje se min. cena, od které se vyhledává
PARENT=ID hlavní kategorie
SKLADEM=Je skladem
SKLADEM2=Označení v DB, že položka je skladem
NEDOSTUPNY=Nedostupný
NEDOSTUPNY2=Označení v DB, že položka je nedostupná
DOSTUPNOST2=Povolit ve vyhledávání
```

Obrázek 13: Jazykový soubor

Při vývoji modulu se musíme řídit podobnými pravidly jako při vývoji komponenty. Jako u komponent platí, že hlavní adresář je ve tvaru mod_jmenomodulu (v mém případě mod_qcwizard).



Obrázek 14: struktura adresáře modulu

Souborový systém modulu se skládá z:

- /modules/mod_qcwizard/helper.php – soubor s pomocnými funkcemi
- /modules/mod_qcwizard/index.html
- /modules/mod_qcwizard/mod_qcwizard.php – vstupní bod modulu
- /modules/mod_qcwizard/mod_qcwizard.xml – konfigurační soubor
- /modules/mod_qcwizard/qcwizard.css – soubor se styly
- /modules/mod_qcwizard/css/showLoading.css – soubor se styly
- /modules/mod_qcwizard/js/jquery.cookie.js – JavaScriptová knihovna
- /modules/mod_qcwizard/js/jquery.showLoading.min.js – JavaScriptová knihovna
- /modules/mod_qcwizard/tmpl/default.php – soubor s HTML
- /modules/mod_qcwizard/tmpl/index.html

7.1. Úprava <head>

Abychom mohli používat knihovnu jQuery a dodatečné kaskádové styly, je nutné vložit odkaz do sekce <head>. Samozřejmě by šlo vložit odkazy například do šablony, ale to by správce e-shopu musel zasahovat do zdrojového kódu. Proto Joomla! obsahuje možnost vložit tam tyto odkazy automaticky. Musíme začít tak, že vložíme řádek:

```
$document =& JFactory::getDocument();
```

Dále máme možnost již vkládat odkazy na soubory.

- `$document->addStyleSheet();` - umožňuje do hlavičky vložit odkaz na kaskádové styly
- `$document->addScript();` - vloží odkaz na JavaScriptový soubor
- `$document->addScriptDeclaration();` - vloží JavaScript

```
$document =& JFactory::getDocument();
$document->addStyleSheet("components/com_gcwizard/treeview/jquery.treeview.css");
$document->addStyleSheet("components/com_gcwizard/treeview/screen.css");
$document->addScript("components/com_gcwizard/treeview/jquery.js");
$document->addScript("components/com_gcwizard/treeview/jquery.noconflict.js");
$document->addScript("components/com_gcwizard/treeview/jquery.cookie.js");
$document->addScript("components/com_gcwizard/treeview/jquery.treeview.js");
$document->addScriptDeclaration('jQuery(document).ready(function() {

    jQuery("#browser").treeview({
        collapsed: true,
        animated: 250,
        persist: "cookie"
    });
});});
```

Obrázek 15: Vložení odkazů na soubory

```

<script type="text/javascript" src="/includes/js/joomla.js"></script>
<script type="text/javascript" src="/media/system/js/mootools.js"></script>
<script type="text/javascript" src="components/com_qcwizard/treeview/jquery.js"></script>
<script type="text/javascript" src="components/com_qcwizard/treeview/jquery.noconflict.js"></script>
<script type="text/javascript" src="components/com_qcwizard/treeview/jquery.cookie.js"></script>
<script type="text/javascript" src="components/com_qcwizard/treeview/jquery.treeview.js"></script>
<script type="text/javascript">
jQuery(document).ready(function(){

    jQuery("#browser").treeview({
        collapsed: true,
        animated: 250,
        persist: "cookie"

    });

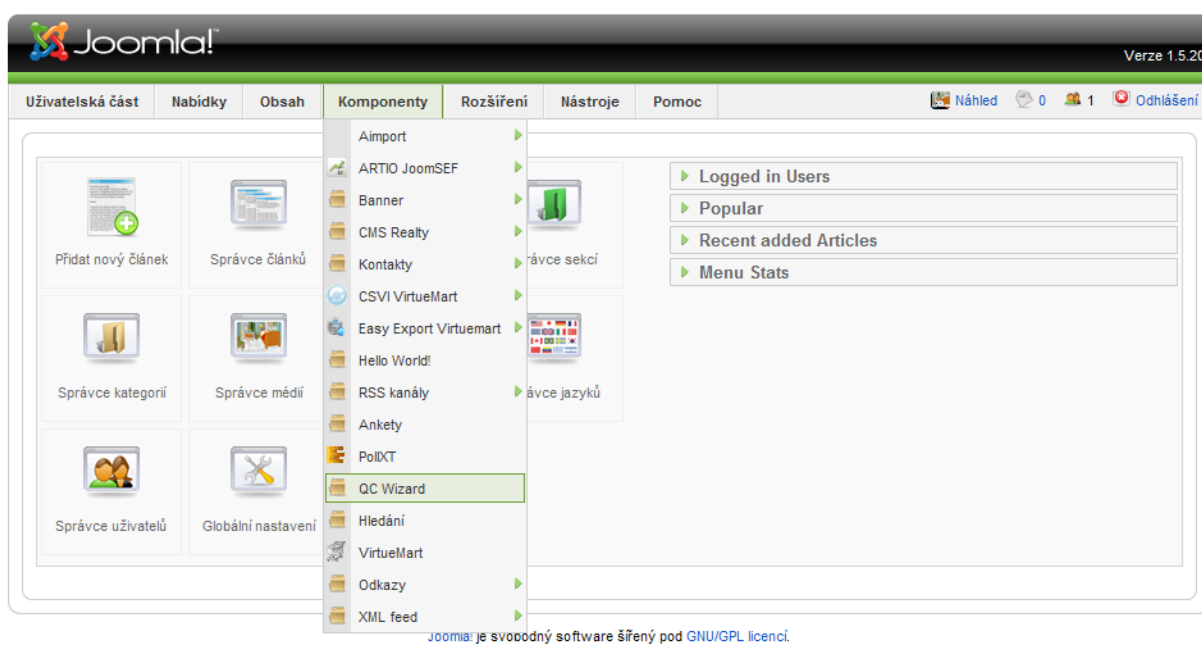
});
</script>

```

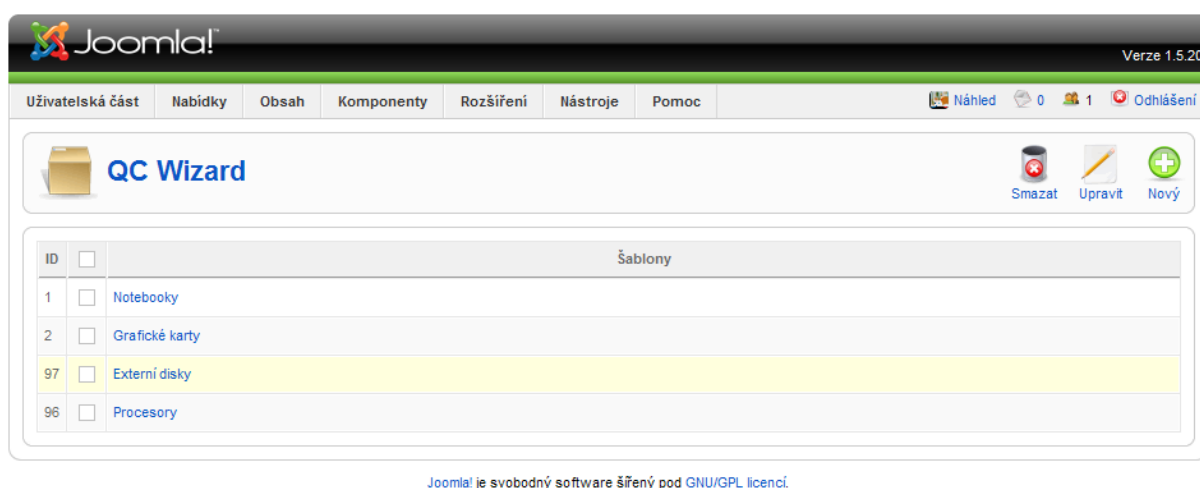
Obrázek 16: Sekce <head>

8. Administrační část (back-end)

Administrační část systému Joomla! umožňuje správcům webu spravovat jeho obsah. Obsahuje řadu funkcí jako přidávání článků, tvorbu anket, správu produktů v e-shopu atd. V případě mé komponenty se budou editovat šablony pro vyhledávání. Na obrázku č.17 je hlavní stránka administračního rozhraní Joomla. V horní části je hlavní menu, přes které se dá dostat ke všem funkcím systému. Naší komponentu najdeme v sekci Komponenty pod názvem QC Wizard.



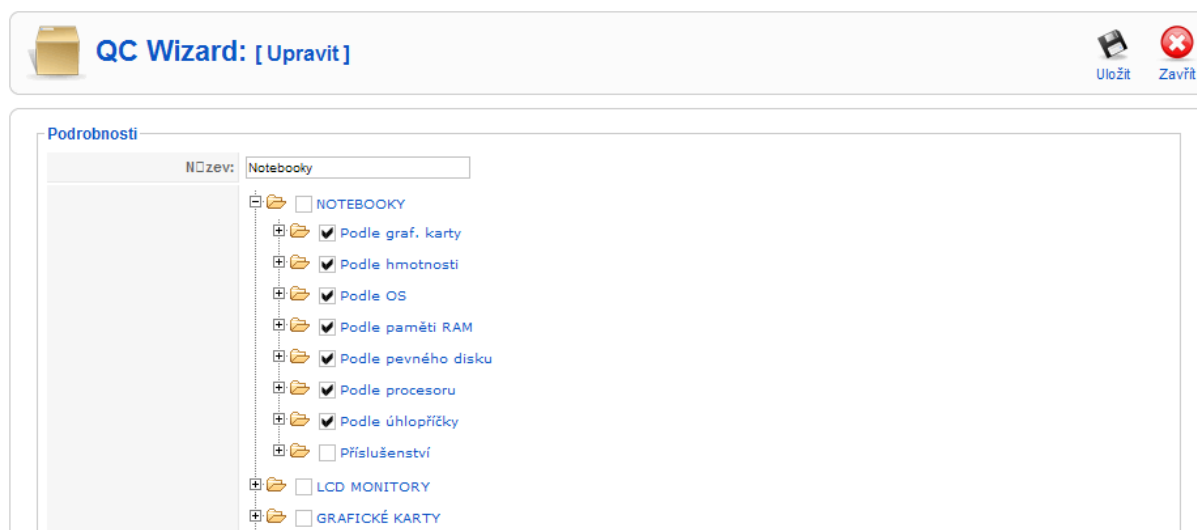
Obrázek 17: Hlavní stránka administračního rozhraní Joomla



Obrázek 18: Hlavní stránka komponenty

Po kliknutí na položku QC Wizard se nám otevře editační prostředí pro šablony. V tabulce vidíme všechny již vytvořené šablony, pomocí tlačítek v horním pravém rohu však můžeme přidat další, editovat stávající a nebo smazat jednu či více šablon najednou.

Jelikož byla komponenta vyvíjena a testována na e-shopu, který se zabývá prodejem výpočetní techniky, vytvořil jsem 4 testovací šablony pro vyhledávání (Notebooky, Grafické karty, Externí disky, Procesory).

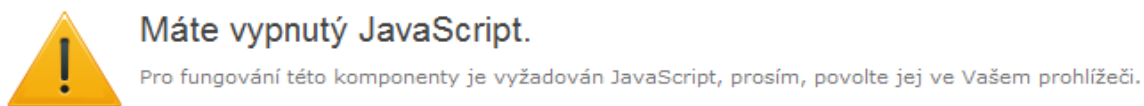


Obrázek 19: Úprava šablony

Na obrázku č.19 je již editační formulář jediné kategorie. Zvolil jsem kategorii Notebooky, kde se po jejím rozkliku vypíší její podkategorie, z kterých následně můžeme sestavit šablonu pro vyhledávání. Podle obrázku je nyní označeno sedm podkategorií, které se následně využijí jako filtr pro vyhledávání.

9. Uživatelská část (front-end)

Uživateli mé komponenty jsou mimo majitelů e-shopu hlavně jejich zákazníci. Ta jim dodává větší komfort při elektronickém nakupování. Je určena především pro ty zákazníky, kteří vědí, jaké má produkt splňovat parametry, jelikož komponenta umožňuje vyhledávání v nepřeberném množství produktů právě na základě různých parametrů. Všem dodává ještě větší komfort okamžité zobrazení výsledků vyhledávání díky technologii AJAX. Jak jsem uvedl v úvodu práce, je očekáván na uživatelské stanici moderní prohlížeč a zapnutý JavaScript. V případě jeho absence není uživateli umožněno vyhledávání a je zobrazena chybová hláška s upozorněním.



Obrázek 20: Chybová hláška

Pokud je splněna podmínka zapnutého JavaScriptu, nic nebrání vyhledávání zboží. Na následujícím obrázku č.21 je snímek uživatelského rozhraní.

Seznam prvků uživatelského rozhraní:

- Uživatel si může pomocí posuvníku vybrat rozmezí ceny, v jaké se bude výsledný produkt vyskytovat. Posuvník je realizován pomocí jQuery utility jménem Slider.
- K určení, podle jaké logiky se budou vypisovat produkty, slouží HTML prvek Select. Produkty můžeme řadit od nejlevnějších, nejdražších, podle názvu.
- Počet zobrazovaných produktů ovlivníme HTML prvkem Select, který nabízí možnost 10, 20, 50 produktů na jednu stránku.
- V případě, že zákazník bude chtít nakupovat pouze z produktů, které má obchod skladem, může zakliknout možnost Skladem.

Dále se nám zobrazí kategorie a jejich prvky, podle kterých budeme

- vyhledávat. V případě zakliknutí nějakého prvku, se nám pro lepší přehled podbarví název kategorie.

- Pokud se zákazník rozmyslí a bude chtít začít nové vyhledávání, slouží k resetování formuláře tlačítko Nové vyhledávání.
- V případě, že by průběh SQL dotazu trval déle, je po dobu jeho trvání zobrazen grafický progressbar.

Průvodce notebooky

Cena: 14331 Kč - 79220 Kč

Od nejdražšího 10

Skladem ☒

PODLE ÚHLOPŘÍČKY	PODLE PROCESORU	PODLE PEVNÉHO DISKU
☐ 15,4 palců ☐ 18palců a více ☐ Do 11palců ☒ Od 11 do 15,4palců ☐ Od 15,4 do 17palců	☐ AMD Athlon 64 ☐ AMD Athlon 64 X2 Dual-Core ☐ AMD Athlon II Dual-Core ☐ AMD Athlon II Neo ☐ AMD Athlon II Neo Dual-Core	☐ 120GB ☐ 128GB ☐ 160GB ☐ 16GB ☐ 250GB
PODLE PAMĚTI RAM	PODLE OS	PODLE HMOTNOSTI
☐ 1024MB ☐ 2048MB ☐ 3072MB ☒ 4096MB ☐ 6144MB	☐ Win7 Home Premium ☐ Win7 Pro ☐ Win7 Starter ☐ Windows VHB ☐ Windows VHP	☐ Do 1kg ☐ Od 1 do 2kg ☐ Od 2 do 3kg ☐ Od 3 do 4kg ☐ Více než 4kg
PODLE GRAF. KARTY		
☐ AMD Radeon HD 6250 ☐ AMD Radeon HD 6310 ☐ AMD Radeon HD 6370M ☐ AMD Radeon HD 6470M ☐ AMD Radeon HD 6490M		

Nové vyhledávání

Nalezeno celkem 28 položek.

0 | 1 | 2 |

Jméno	Cena	Náhled	Dostupnost
<u>Lenovo T410s i5-540M/4GB/128GB SSD / DVD RW / nVidia/14.1" WXGA+ touch / BT / 3G / cam / W7P64</u>	46477 Kč		Skladem
<u>HP NB 8440p i7-640M 14.0 LED HD+AG 4GB 160(SSD) DVD RW BT WF FP 3G cam W7P</u>	41497 Kč		Skladem

Obrázek 21: Uživatelské rozhraní

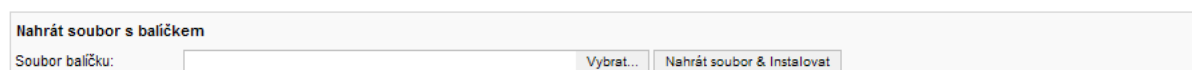
9.1. Prezence front-endu

Vyzkoušet možnosti vyhledávání je možné na e-shopu [pcprovsechny.cz](http://www.pcprovsechny.cz) s reálnými produkty. Formulář pro vyhledávání mezi notebooky je následující: <http://www.pcprovsechny.cz/konfigurator>.

10. Instalace

10.1. Uživatel

Aby nám Joomla! usnadnila práci při instalaci rozšíření, obsahuje instalačního průvodce. Pokud chceme nainstalovat modul, komponentu či třeba nový design webu, instalace nám zabere jen pár sekund. Vývojáři uvolňují svá díla zazipovaných souborech. V administraci Joomla! pak stačí v sekci Instalace vybrat daný ZIP soubor a kliknout na tlačítko Instalovat. Vše již zařídí Joomla a my budeme moci začít používat dané rozšíření.



Obrázek 22: Instalace rozšíření

Jediné, co bude muset správce e-shopu udělat po instalaci, je nakopírování souboru `qcwizard.php`, který se nachází v adresáři `/administrator/components/com_qcwizard/tmpl` do adresáře `/templates/system`. Bohužel nainstalování tohoto souboru do adresáře `/templates/system` instalátor Joomla! neumí.

10.2. Vývojář

Aby uživatelé mohli jednoduše na pár kliknutí nainstalovat dané rozšíření, vývojář musí dodržet jistá pravidla. Proto se pro instalaci vytváří balíček souborů, který je následně zazipován. Instalace se řídí podle souboru XML (obrázek č.23), který obsahuje všechna důležitá data. Platí při tom, že XML soubor v balíčku musí mít název ve tvaru (v mém případě) `mod_qcwizard.xml`. XML soubor obsahuje všechny informace o rozšíření, jméno, licenci, informační text, seznam všech souborů v administrátorské i uživatelské části atd.

```

<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.5.0">
  <!-- Name of the Module -->
  <name>QC Wizard</name>

  <!-- Name of the Author -->
  <author>Karel Bartůněk</author>

  <!-- Version Date of the Module -->
  <creationDate>2011-05-22</creationDate>

  <!-- Module version number -->
  <version>1.0.0</version>

  <files>
    <filename module="mod_qcwizard">mod_qcwizard.php</filename>
    <filename>index.html</filename>
    <filename>helper.php</filename>
    <filename>tmpl/default.php</filename>
    <filename>tmpl/index.html</filename>
  </files>

  <languages>
    <!-- Any language files included with the module -->
    <language tag="cs-CZ">cs-CZ.mod_qcwizard.ini</language>
  </languages>

```

Obrázek 23: Ukázka instalačního XML souboru

11. Závěr

Díky této bakalářské práci vznikla univerzální komponenta s modulem, které je možné nainstalovat na jakýkoliv e-shop postavený na systému Joomla!+Virtuemart, jelikož byla dodržována pravidla softwarové architektury Model-View-Controller s použitím funkcí frameworku Joomla!.

V práci byl stručně popsán redakční systém Joomla!, vývoj komponenty a modulu, následně bylo představeno jak je nasadit do provozu na internetový obchod a detailně popsána funkcionality.

Pro dosažení kvalitních výsledků bylo vše projednáváno s provozovateli internetového obchodu pcprovsechny.cz.

Vývoj této komponenty by měl v budoucnu směřovat k jejímu přepracování pro nové verze systému Joomla! (1.6+) a rozšiřování funkcionality podle nových požadavků provozovatelů zmíněného internetového obchodu.

12. Zdroje informací

[1] LEBLANC Joseph. Learning Joomla! 1.5 Extension Development.

Birmingham: Packt Publishing Ltd., 2007. ISBN 978-1-847191-30-4

[2] Joomla Framework API documentation [online].

URL: < <http://docs.joomla.org/Framework> >.

[3] Joomla component development [online].

URL: < http://docs.joomla.org/Component_Development >.

[4] Dokumentace k PHP [online].

URL: < <http://www.php.net/docs.php> >.

Příloha – A obsah přiloženého CD

Přiložené CD obsahuje adresáře s daty:

- Sources – Kompletní zdrojové kódy
- Archive – Instalační balíčky modulu a komponenty
- Document – Bakalářská práce ve formátu PDF