

BESTSIDESTORY

Project Work:

Development of the online shop
"Diesen-Samstag.com"
with Magento



Report on project work

Ardalan Ahmed Ismail
IT Specialist
Subject area
Application Development

Practical operation:

Bestsidestory GmbH
Neumarkt 20
D-04109 Leipzig
Germany

Coaches:

Matthias Schnick

Implementation period:

28.09.2011-20.10.2011

1. Context

1.1. Project Environment

The Agency Bestsidestory GmbH specializes in e-commerce, marketing and public relations. It also operates several online stores. One of them is this-Samstag.de.

This-Samstag.de is the online pharmacy for alternative medicine and natural cosmetics. The store carries products from the following areas: naturopathy, homeopathy, biochemistry, and Spagyrik Spenglersan therapy

1.2. Project

The existing online store "Diesen-Samstag.com" to be placed on a Magento shop solution. So far, the online shop based on ePages Strato.

Reasons for the development of a new solution:

- Cost: Magento is an open source software, ePages, however, a commercial software and requires a monthly fee
- Flexibility: Magento can be extended by installing free or self-developed modules, ePages evolutions, however, are proprietary
- ePages has no connection to Enterprise-Resource-Planning systems (ERP system)
- ePages allows the webshop developers have no direct access to the database

1.3. Project Objective

The aim of the project is a professional and convenient web shop for the online pharmacy that provides comprehensive functions for both the owner and the user. Here, on an already existing online shop (Diesen-Samstag.de) is placed.

1.4. Economic Evaluation

The new online shop:

- is easy to use for administrators
- characterized by the use of Magento through great usability and clarity of features in the administration area on better connections to ERP systems (enterprise resource planning system)
- is easier to find in the results of search engines, because this Magento by default already relevant ways to deliver so-called on-page optimization such as speaking URLs, meta tags maintainable, individual page titles, automatic rewrites, and an integrated XML and HTML sitemap for Google
- is based on a technology that has achieved now a global and ever-growing community of developers and, together with the Magento Inc. for the development of the Software Store

These reasons listed prove a cost and time savings for the business owner.

1.5. Changes to the project proposal

- The import of customer data was not possible in the allotted time, as they must be checked by the customer due to data protection reasons
- Creating a completely new template in Magento has taken more time than the planned 10 hours, required for this importing and exporting the database less time than the planned 12 hours
- Instead of the required module to display an extra price for customers, a module was developed to display a percentage price savings to avoid confusion by representations (Figure 1.1) of multiple prices



Figure 1.1 percentage price savings

1.6. Legal Situation

All contents of the "Diesen-Samstag.de" website is provided by Bestsidestory agency and the agency is available to provide the webshop developed in this project to use the Internet. All legal issues incumbent on the customer.

2. Project Planning

2.1. Actual state

The current online shop (Diesen-Samstag.com) is based on ePages and currently is the functionality in the admin area extremely limited:

- The administrator does not have access to the databases, because it is restricted by Strato
- ePages it inflexible and its functionality is determined only by ePages developers
- The online store has no connection to WaWi systems (enterprise resource planning system)
- ePages has no proper export interface

In addition to the many limitations ePages is commercial software and must be rented from a provider.

2.2. Target state

The current online shop is to be replaced on the basis of open source software, it is important to note that the design is not changed fundamentally.

A comparison between Magento and ePages

Magento	ePages
Technology	
PHP	Perl
MySQL	unknown
Preis	
Open Source	8000\$
General	
Functionality can be extended by installing modules	Functionality is determined only by ePages developers
Export options are very flexible and can be automated	No proper export interface available
Access to databases	No access to database
Several ways to connect to the shop ERP systems	No connection to ERP systems
Better access to social media	
Affiliate marketing is possible	Affiliate marketing is not possible

As indicated in the table, Magento has many advantages over e-Pages.

2.3. Alternatives

There is an alternative to Magento and that is also available as open source software OXID - esales.

Both systems support elementary functions:

- Complex products with variants, attributes, volume discounts several (zoom) images
- Accessories products, cross-sell and upsell
- Coupons and Discounts
- Search engine optimization

And here's a comparison between both CMS systems

Magento	OXID- esales
Organization	
Magento Inc. headquartered near Los Angeles, USA approximately 290 employees	OXID - eSales AG headquartered in Freiburg, Germany approximately 60 employees
First version (1.0) was published on 31 Mar 2008 heard since the 06 June 2011 on eBay	First open source version was published in autumn 2008
Features	
complex template system partially high effort for easy change, but by clean system structure	Easy to adjust Templates
timed, various designs or per General Category / page possible	Widespread Smarty Template engine in use
Extension / adaptation by internal and external modules	
Complex module system	Relative Simple module implementation
Tutorial's sparse documentation and examples on blogs, forums and books	Helpful and extensive documentation
Many modules in Magento Connect	Numerous modules under OXID eXchange

Both systems have their strengths and weaknesses.

2.4. Decision

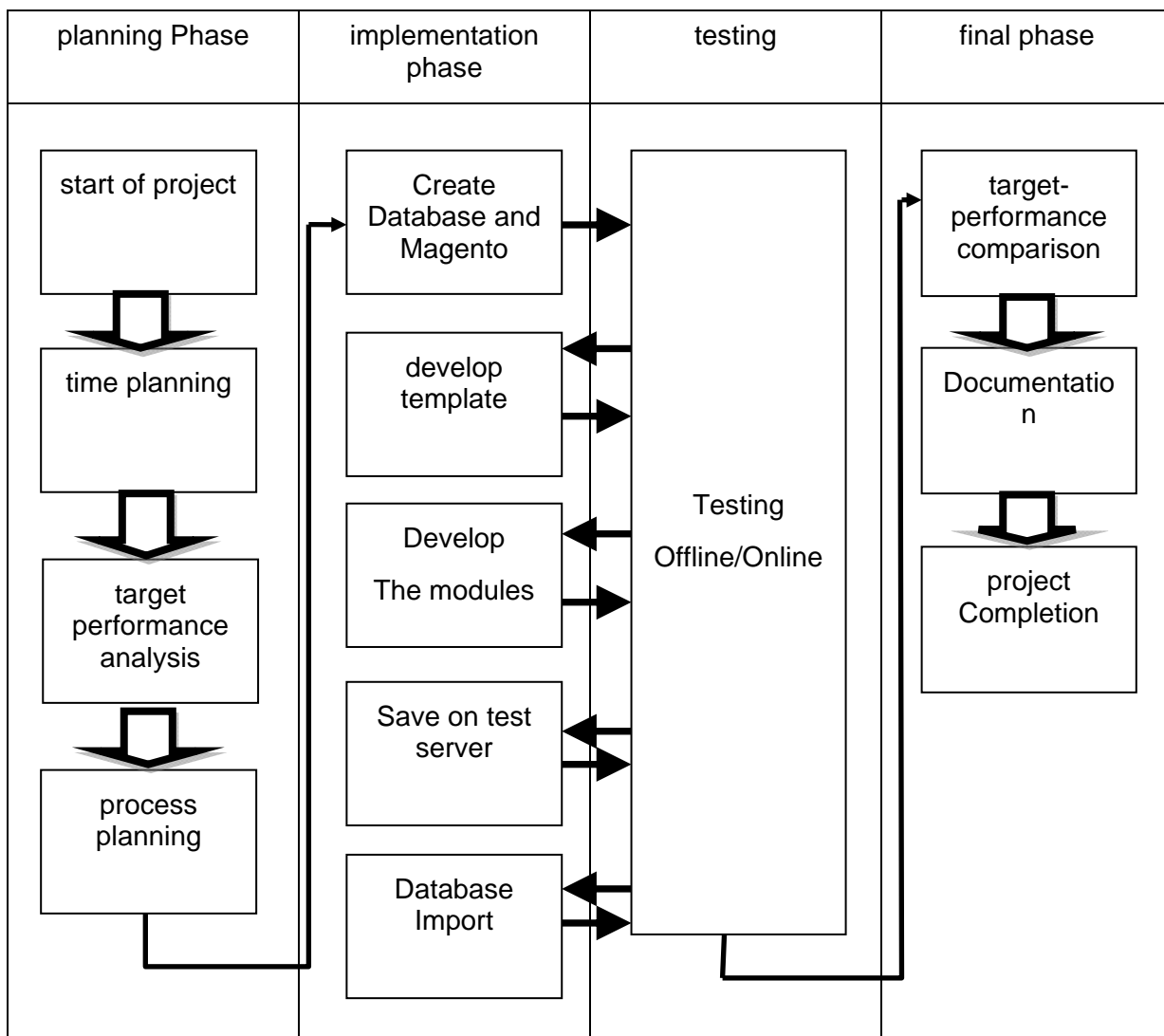
	Magento			OXID - esales	
	Factor	Points	Points x Factor	Points	Points x Factor
Employee experience	5	5	25	3	12
Language	5	3	15	5	25
user-friendly backend	4	5	20	4	20
Adaptability and extensibility	3	3	9	5	15
SEO	4	4	16	3	12
Data export	3	4	12	3	9
Development opportunities	3	4	12	3	9
Organization	2	4	8	2	8
Total sum			119		100

Weighting criteria (Factor of 5 to 1 = very important to less important)

Decision for the highest score Magento.

After the presentation of pre - and disadvantages of the alternatives and a final interview, the client opted for Magento.

2.5. Flow diagram



The test phase is parallel to the implementation phase, as the flawless execution of the previous functions is necessary for the individual to be programmed functions.

2.6. Cost Planning

For the software used is no cost, since Notepad ++, Apache, PHP and MySQL open source software is used.

No special hardware is required, as the project is run locally on any PC using the given software and used the laptop was available, therefore no cost only up costs (rent, electricity, paper, etc.).

There are no personnel costs excluding costs of care.

3. Magento

3.1. Short Description

Magento is a system developed in PHP shop system, it shows very flexible, not only in product categorization, but also in other areas. Magento has its own search engine optimization, an Ajax-supported application interface for front-end and back as well as more substantial analysis functionality.

Magento is a serious solution that will establish itself just in the middle segment of e-commerce

3.2. Why Magento?

Magento has many benefits

- modern e-commerce system
- contemporary arsenal of sales agents and functions
- High modularity
- source openness
- High adaptability to customer requirements
 - Data
 - Function
 - user interface
- Marketplace for extensions
- Dynamic developer community
- A variety of third-party extensions

3.3. Architecture von Magento

The Magento application adheres strictly to the structural design of the Model-View-Controller (MVC) pattern to translate business logic into programming logic. In simple terms, involves the MVC concept that the presentation layer (view) of the data layer (model) is detached and the data flows through the control layer (controller) can be controlled, as the following figure shows.

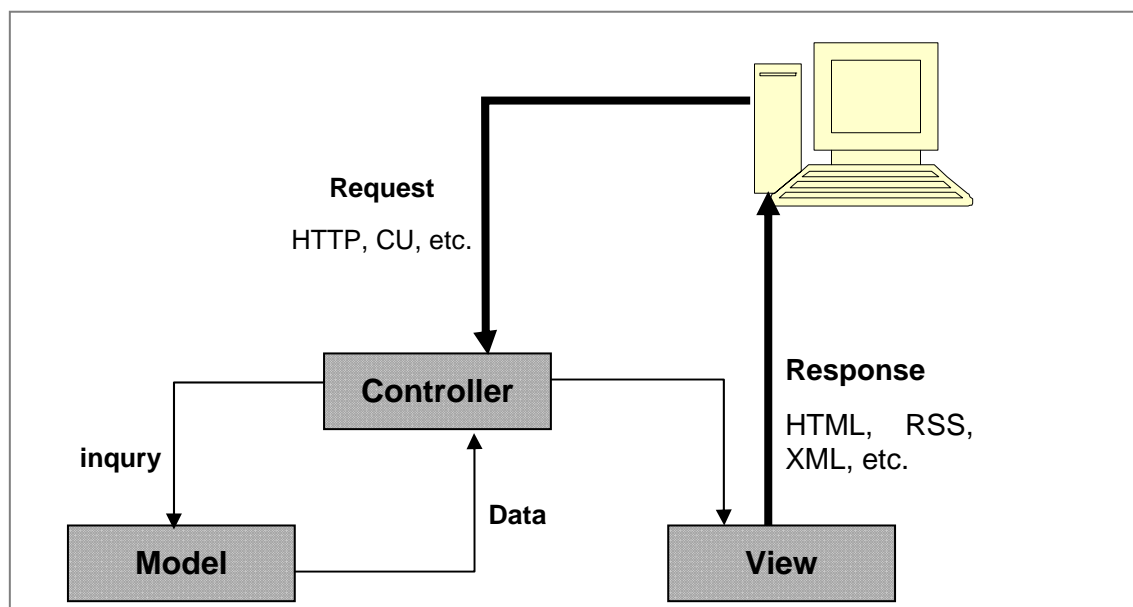


Figure 3.1 Model-View-Controller (MVC)

As a modular template system is built:

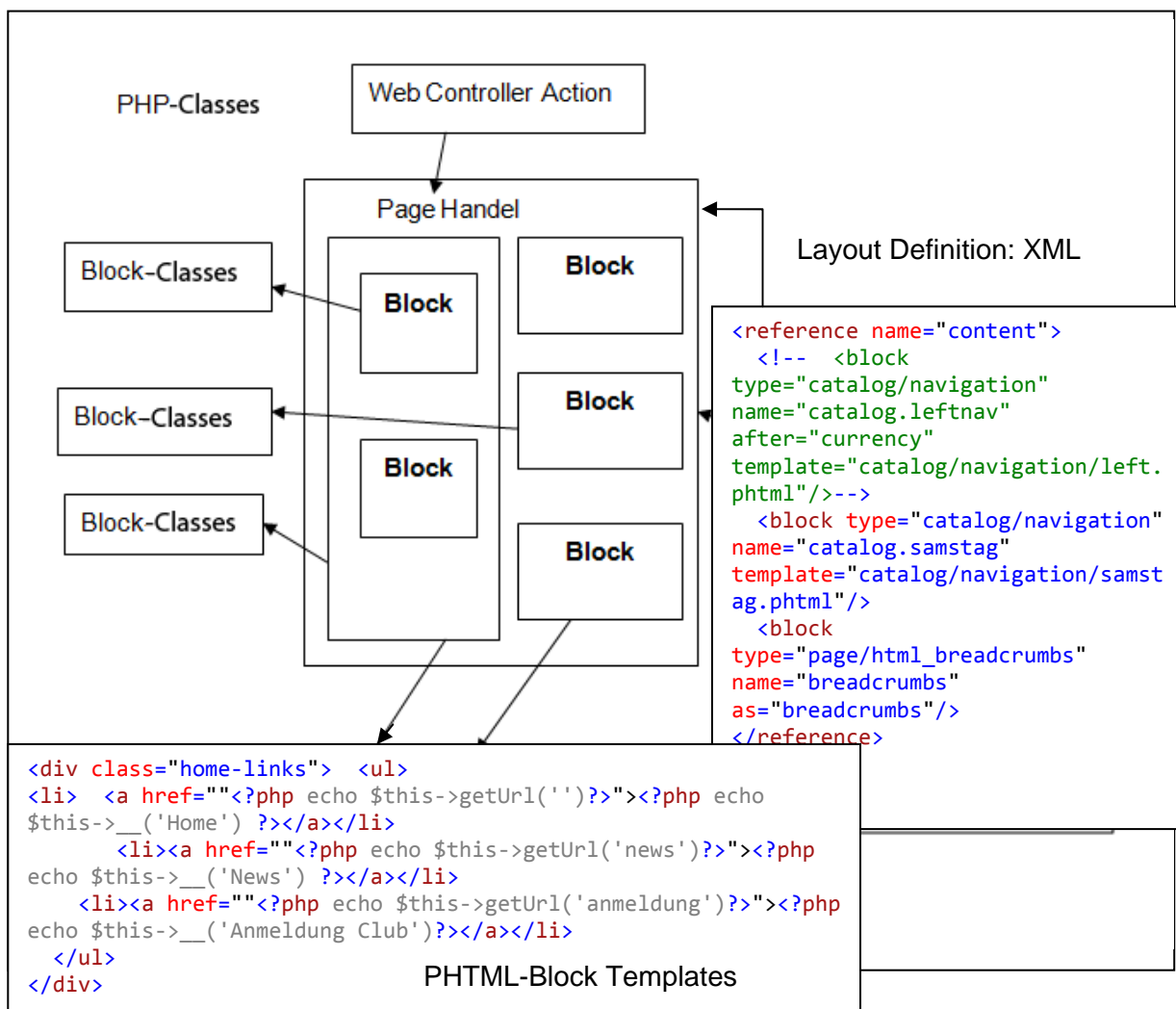


Figure 3.2 a modular template system

PHTML is the extension of a special HTML file (eg index.phtml). Using this extension, the Web server recognizes that this is not a normal HTML file, but that he must first have something to do with it. The server searches these HTML tags

:

<?php (xxxxxxx) ?>

Modular software architecture: Architecture is the art which is currently found in the e-commerce sector. It allows extensive customization and integration with other systems such as inventory management programs, eBay, Amazon, price search engines and payment providers.

4. Project implementation

4.1. Development Environment

The project is being developed on a PC with Windows 7 operating system, since this was directly available. In this Notepad + + and downloaded and installed XAMPP.

4.2. Backend changes

No changes are made in this area.

4.3. Designing the user interface

The user interface (front end) should as far as possible consistent with the existing online store, so that the usual workflow is maintained for the user. To this end, a new template is created under the name **Samstag** and will be integrated in the Magento directory.

When creating the template it must be ensured that the core of Magento structure is not changed. Any change in the core structure can lead to major problems in the next update.

As shown in the architecture figure (Figure 3.2) of Magento, the layout (design) is strictly separated from the functionality.

Legal adjustments: Magento was not designed for the German market, so adjustments must be made for the German market. Changes to be made are:

- Terms and Conditions
- Disclaimer
- Revocation
- Note about shipping costs
- tax rules, tax rates and tax brackets
- Basic pricing (price per set)
- Invoice and delivery note adjustments
- E-mail signature Customize

4.3.1. Creating the template:

Magento brings a complete template with it, but that does not meet the requirements for the shop to be developed.

The changes need to be made:

- in existing products all sides are three columns, but in the Magento Template design varies the page
- Magento has a horizontal menu, is called a vertical drop down menu
- Magento has many additional features we do not need, such as the price comparison between two products, coupon redemption, etc., that are to be deactivated
- in existing online store also a main navigation for the shop categories is a special navigation for news, blog, health issues, etc. It is to be created
- other partner stores will be linked to the new block element, which can be seen on each side
- Each article should be linked to the so-called social media (Facebook, twitter, etc.)
- Integrate a blog (WordPress) to the side with its own database and its own template
- Create a block element and a dynamic page for Bestsellers
- Create a dynamic page for weekly offers
- Create a home page with displaying selected item
- Representation of percentage savings of articles with twice the price (old & new)

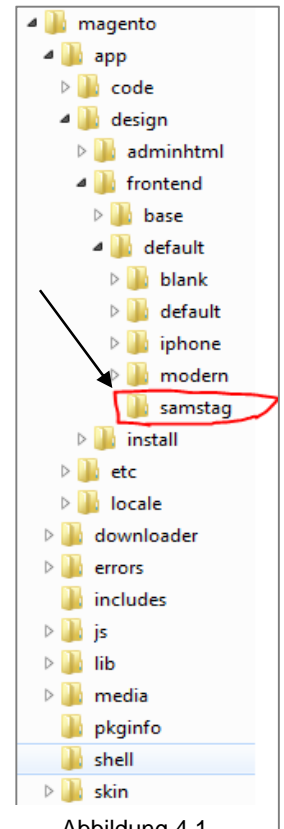


Abbildung 4.1
Magnto Template

Code example - change the existing template:

To create the first vertical navigation the existing navigation must be disabled or deleted. Then a new navigation is created as follows:

1. First Clear block of the existing navigation and create a new block element in the XML file.

```

<reference name="content">
<!-- <block type="catalog/navigation" name="catalog.leftnav" after="currency"
template="catalog/navigation/left.phtml"/>--> ←A
<block type="catalog/navigation" name="catalog.samstag"
template="catalog/navigation/samstag.phtml"/> ←B
<block type="page/html_breadcrumbs" name="breadcrumbs" as="breadcrumbs"/>
</reference>
    
```

Figure 4.2 XML file (catlog.xml)

A → the top navigation block is disabled

B → a new block for vertical navigation is created

Second Creating the new PHTML file for the vertical navigation

```

<div class="vertical-nav-container box base-mini">
  <div class="vertical-nav">
    <!--<div class="head">-->
      <div class="block block-cart">
        <div class="block-title">
          <strong><span><a href="<?php echo $this->
          getUrl('')?>"><?php echo $this->__('Categories') ?>
          </span></strong></a>
        </div><!--End block block-cart-->
        <div class="block-content">
          <!--</div>--><!--End Of head-->

          <h4 class="no-display">
            <?php echo $this->__('Category Navigation:') ?></h4>
            <ul id="nav_vert">
              <?php foreach ($this->getStoreCategories() as $_category): ?>
                <?php echo $this->drawItem($_category) ?>
                <?php endforeach ?>
              </ul>
            </div>
          </div><!--End Of vertical-nav-->
          <?php echo $this->getChildHtml('topLeftLinks') ?>
        </div>
      </div>

```

Figure 4.3 PHTML file(vert nav.phtml)

- Formatting the new block in CSS file

```

#nav_vert ul {margin:0; padding:0;font-size:13px;font-
weight:normal;overflow:hidden;}
#nav_vert a {text-decoration:none;}
#nav_vert li {float:left; width:160px; margin-left:10px;color:#000;padding-
left:5px; }
#nav_vert li ul { display:none;padding-left:0px;font-weight:normal ;}
#nav_vert li ul ul{ display:none; }
#nav_vert li ul ul li{ display:none; }
#nav_vert li:hover ul,#nav_vert li:focus ul { display:block; }
#nav_vert li ul:hover li,#nav_vert li ul:focus li { display:block; }
#nav_vert li ul a:active{display:block;}
#nav_vert ul li a:focus .nav-span,
#nav_vert ul li a:active .nav-span {display:block;border:1px solid #f00;}

```

Figure 4.4 CSS file

Code example - developing a new template:

Development of a second navigation blocks for the shop:

- Create the block in the XML file

```
<reference name="content">
<!-- <block type="catalog/navigation" name="catalog.leftnav" after="currency"
template="catalog/navigation/left.phtml"/>-->
<block type="catalog/navigation" name="catalog.samstag"
template="catalog/navigation/samstag.phtml"/> ←A
<block type="page/html_breadcrumbs" name="breadcrumbs" as="breadcrumbs"/>
</reference>
```

Figure 4.5 XML file (catlog.xml)

A → the block element created

- Write the corresponding PHTML file

```
<div class="home-links">
<ul>
<li><a href="<?php echo $this->getUrl('<'><?php echo $this->__('Home') ?></a></li>
<li><a href="<?php echo $this->getUrl('themen')?>"><?php echo $this->__('Themen') ?></a></li>
<li><a href="<?php echo $this->getUrl('marken')?>"><?php echo $this->__('Marken') ?></a></li>
<li><a href="<?php echo $this->getUrl('methoden')?>"><?php echo $this->__('Methoden')
?></a></li>
<li><a href="<?php echo $this->getUrl('customer/account/login/')?>"><?php echo $this-
>__('Anmeldung Club') ?></a></li>
<li><a href="<?php echo $this->getUrl('news')?>"><?php echo $this->__('News') ?></a></li>
<li><a href="<?php echo $this->getUrl('blog')?>"><?php echo $this->__('Blog') ?></a></li>
</ul>
</div>
```

Figure 4.6 PHTML file(Samstag.phtml)

- The new block formatting in CSS file

```
.home-links{ font-size:14px;padding:3px;width:554px;
background:#f1f1f1;margin-bottom:5px;
height:28px;
}
.home-links li { display:inline;padding-right:5px;border-right:1px solid #aaa;}
.home-links li:last-child{border-right:none;}
```

Figure 4.7 CSS file

Code example - the existing extensions PHTML block:

View the article should be expanded to include a way to navigate between items of the same category can.



The following code is added to the Product View

```
<?php
if ($this->helper('catalog/data')->getCategory()) {
    $categoryname = $this->helper('catalog/data')->getCategory();
} else {
    $categorynames = $this->helper('catalog/data')->getProduct()->getCategoryIds();
    $categoryname = Mage::getModel('catalog/category')->load($categorynames[0]);
}
$productposition = $categoryname->getProductsPosition();
$current_pid = $this->helper('catalog/data')->getProduct()->getId();
// bauen Array von Produkten Positionen
$plist = array();
foreach ($productposition as $pid => $pos) {
    $plist[] = $pid;
}
$currentposition = array_search($current_pid, $plist);
// get link für zum vorherigen Artikel
$previd = isset($plist[$currentposition+1])? $plist[$currentposition+1] : $current_pid;
$product = Mage::getModel('catalog/product')->load($previd);
$prevpos = $currentposition;
while (!$product->isVisibleInCatalog()) {
    $prevpos += 1;
    $nextid = isset($plist[$prevpos])? $plist[$prevpos] : $current_pid;
    $product = Mage::getModel('catalog/product')->load($nextid);
}
$prev_url = $product->getProductUrl();
// get link für das nächste Produkt
$nextid = isset($plist[$currentposition-1])? $plist[$currentposition-1] : $current_pid;
$product = Mage::getModel('catalog/product')->load($nextid);
$nextpos = $currentposition;
while (!$product->isVisibleInCatalog()) {
    $nextpos -= 1;
    $nextid = isset($plist[$nextpos])? $plist[$nextpos] : $current_pid;
    $product = Mage::getModel('catalog/product')->load($nextid);
}
$next_url = $product->getProductUrl();
// get link in der aktuellen Kategorie
$more_url = $categoryname->getUrl();
?>
<div class="produkt_nav">
<a class="vor" href="<?=$prev_url; ?>"><< Vorherige</a>
<!-- <a href="<?=$more_url; ?>">zu Kategorie</a -->
<a class="next" href="<?=$next_url; ?>">Nächste >></a>
</div>
```

4.3.2. the design:

A new folder is created, the:

- CSS Files
- JavaScript Files
- Images

includes.

Then, the original files of Magento will be expanded and changed.

Hint: This procedure is better to put all the files from scratch as because Magento already provides many useful presentations. Additionally, there are numerous templates folder in Images.

To realize the shop but the default configuration, ie the use of a three-column layouts, shop specific colors, fonts and images are included which are responsible for the original design Magento files in the newly created folder (Samstag), and then adjusted adding new files needed.

New files (see The Figure 4.6).

- Samstag.css in Ordner css
- Shop spezifische Bilder in Ordner Images
- JavaScript Datei in der Ordner js

Hint: Magento provides a CSS file for the elderly Versions of Internet Explorer (6 and 7) must be prepared and are therefore not new. All changes take place directly in this file.

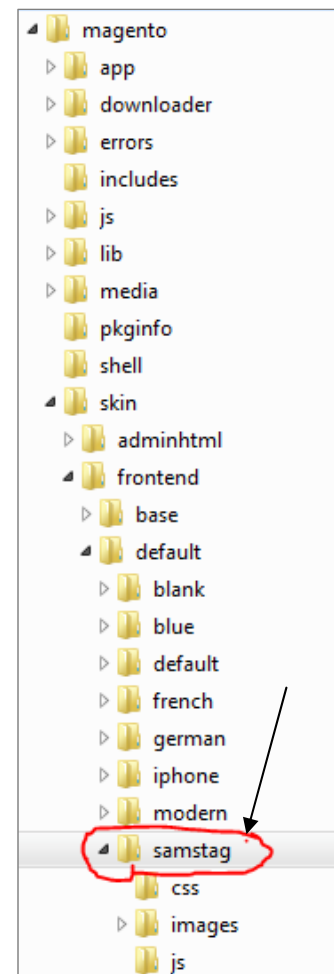


Figure 4.8
Design Folder

4.4. Creating the static pages:

The most important pages to be created first, are:

- Bestseller
- How to purchase
- Delivery
- Terms and Conditions
- Revocation
- Payment methods
- News

Danach werden die Verweise für diese Seiten eingerichtet.

Damit der Benutzer die Möglichkeit hat AGBs und Widerrufsbelegungen zu speichern und zu drucken, werden außerdem entsprechende PDF-Dateien erstellt und unter media/pdf abgelegt. Anschließend müssen diese per Verweis eingebunden werden.

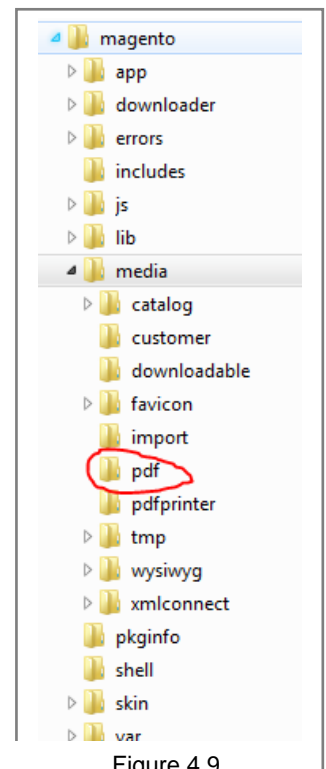


Figure 4.9
PDF Folder

4.5. WordPress Integration (blog)

Upon request, the site operator a separate database for WordPress is created first. After the files of WordPress backbone in a subfolder of Magento's directory structure (magento / blog) must be inserted and installed.

After installing the header of the blog is designed just like the online stores. In addition, a link to the main page in the blog is integrated.

4.6. Data Import:

It was (as originally intended) any PHP file to import the data written, because Magento already has a corresponding function (data flow) features, can easily be imported using this CSV file. Previously, the attributes must however. Adapted within the CSV file to the requirement of Magento's.

Important in the use of Data Flow:

- All necessary images for existing products are in the Magento directory / media / import.
- To import CSV file is located in Magento / var / Import Export.
- decomposition of the CSV file into part file
 - in the original CSV file contains over 2000 items, so that the file is too large for a single import operation
 - with a problem / error within the script the import function is aborted with no error message

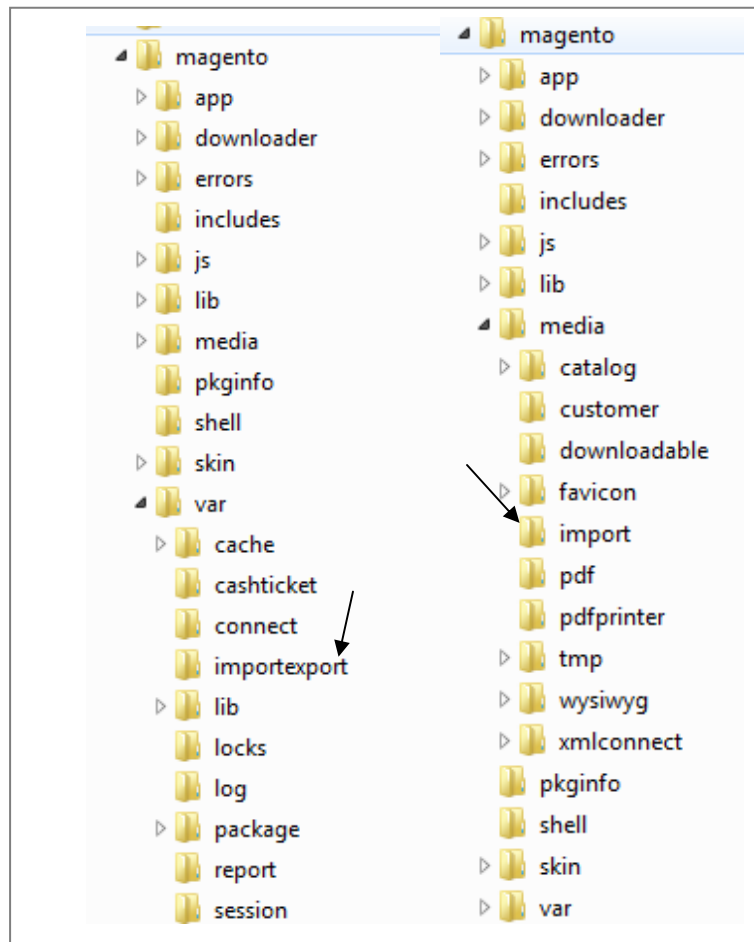


Figure 4.10
Import folder for images and CSV files