Chenjia Lai

Front-end Development and Multi-language Implementation on Magento Platform

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

15 May 2015

Abstract

Author(s)	Chenjia Lai
Title	Front-end Development and Multi-language Implementation on Magento Platform
Number of Pages	31 pages + 4 appendices
Date	15 May 2015
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Web programming
Instructor(s)	Dr. Tero Nurminen, principal lecturer

Magento is a smart platform for E-commerce solutions. Aseanic Trading Oy plans to develop a website based on the Magento platform. The purpose of this project is to develop an E-commerce platform for middle and small companies, to improve product management, and to increase a company's profit.

This project is developed based on the Magento platform. The backend of the Magento platform uses PHP language, while the frontend uses DIV, CSS and jQuery. Also, MySQL is used as the database supported by Magento as default. The project focuses on the frontend UI design, multiple language switching and payment method integration.

The results show that the frontend UI has been designed. The project was also implemented multi-language switching successfully. Moreover, the payment method integrated has been done.

More functions could be developed in future work to synchronize inventory between the website and the Internal system.

Keywords | Magento, CSS, DIV, jQuery, E-commerce

Contents

ABBREVIATIONS

API	Application Programming Interface
CMS	Content Management System
CSS	Cascading Style Sheet
CSV	Comma-Separated Values
DIV	DIVision
HTML	Hyper Text Markup Language
JS	JavaScript
LAMP	Linux-Apache-MySQL-PHP
MVC	Model-View-Controller
OSL	Open Software License
UI	User Interface
WAMP	Windows-Apache-MySQL-PHP
XML	Extensible Markup Language

1 Introduction

An e-commerce platform becomes more and more important in business. It is used for example for promotion, shopping. Online shopping has been explosively increased since 2004. Most companies started their business in an e-commerce platform in order to occupy the market. [1]

Aseanic Trading Oy is an Asian product retailer that runs department stores and warehouse stores in Helsinki, Espoo, Vantaa and Tampere. It decided to build an online shop to provide services in Finland.

The aims of the project are to develop a platform for market brand building, to advertise promotions, and to do shopping online. Because the products in the Asian shop are important for Asian people who are immigrants and students in Finland, the website should include different languages such as English, Finnish, Chinese, Vietnamese and Thais. Another important issue is to provide high-level security for payment. Under the circumstances, Aseanic Trade Oy has made a contract with POINT which is a professional payment gateway in Finland. POINT guarantees high-level security for the payment API (Application Programming Interface).

The website development is based on the Magento platform. Other functions or modules, such as the front-end UI (User Interface) design, which are developed by CSS (Cascading Style Sheet), DIV (DIVision) and jQuery, are extended into the core of Magento. UI (User interface) design and multi-language are implemented by CSV (Comma-Separated Values) files.

2 Background and market analysis

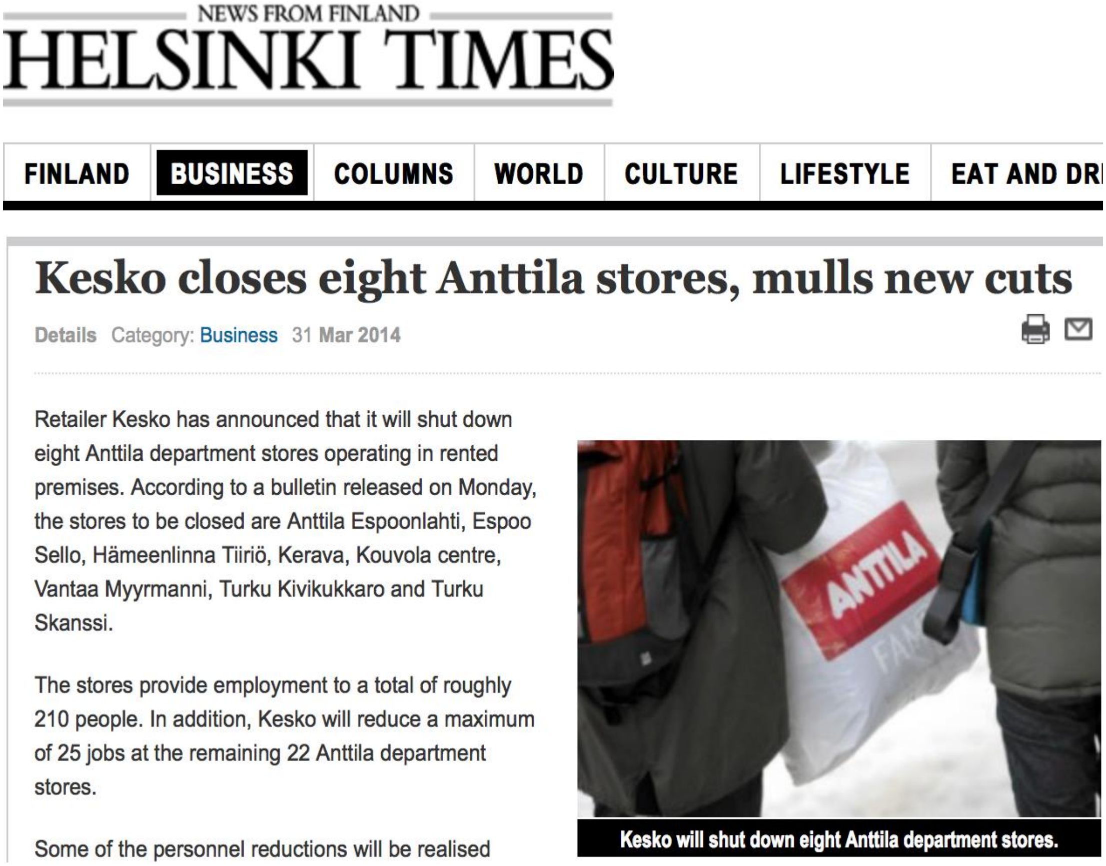
In Finland, one of the high-tech countries in the world, most new technologies, such as smart-phones and the Internet, have been applied to people's lives. According to the latest information, 90% of the population whose ages are 16 to 74 have access to the Internet, while 49% of them access the Internet by smartphones.

Online shopping did not start in Finland as early as other Nordic countries. However, Finnish people are adapting to it rapidly. Approximately two thirds of the Finnish population shopped online during 2011 and the number is growing every year. Online shopping is continuing to be more popular among young adults than any other age groups. Two-thirds of people aged 25 to 34, while one-tenth of those aged 65 to 74, did online shopping on the Internet during the past three months. The market as a whole has been growing at a rate of around 13 – 18 % annually for several years.

The turnover of e-commerce in Finland was approximately € 10.1 billion in 2011. 87% of it is domestic commerce and 13% is foreign commerce. Among it, 47% is from goods sale, 52% from services and 1% from the content. [1]

Many companies have carried out e-commerce projects in recent years to improve efficiency and reduce costs. In the past two years, several famous Finnish companies, such as Anttila and Musta Pörssi, have closed shops and paid more attention to e-commerce platform. In this situation, the e-commerce platform is impacting on traditional retailers.

Anttila closed the shop in Sello in November 2014, and has paid attention to the online platform. [2]



Kesko closes eight Anttila stores, mulls new cuts

Details Category: Business 31 Mar 2014

Retailer Kesko has announced that it will shut down eight Anttila department stores operating in rented premises. According to a bulletin released on Monday, the stores to be closed are Anttila Espoonlahti, Espoo Sello, Hämeenlinna Tiiriö, Kerava, Kouvola centre, Vantaa Myyrmanni, Turku Kivikukkaro and Turku Skanssi.

The stores provide employment to a total of roughly 210 people. In addition, Kesko will reduce a maximum of 25 jobs at the remaining 22 Anttila department stores.

Some of the personnel reductions will be realised

Kesko will shut down eight Anttila department stores.

Figure 1. Anttila closing its shop in Espoo. Copied from[2]

As one of the biggest Asian product retailers in Finland, Aseanic Trading Oy decided to build an e-commerce platform to expand its market in northern Finland. There were several reasons considered by the company for this, and they will be explained in the following of sections.

2.1 Potential customers

Aseanic Trading Oy is successful in the capital region of Finland. However, there is a large potential market beyond that. Before the project started, the potential of the market was unknown. First of all, population research, customer estimation and market analysis are necessary.

Figure 2 shows the most important immigration groups in Finland, which is provided by the Finnish population registration office.

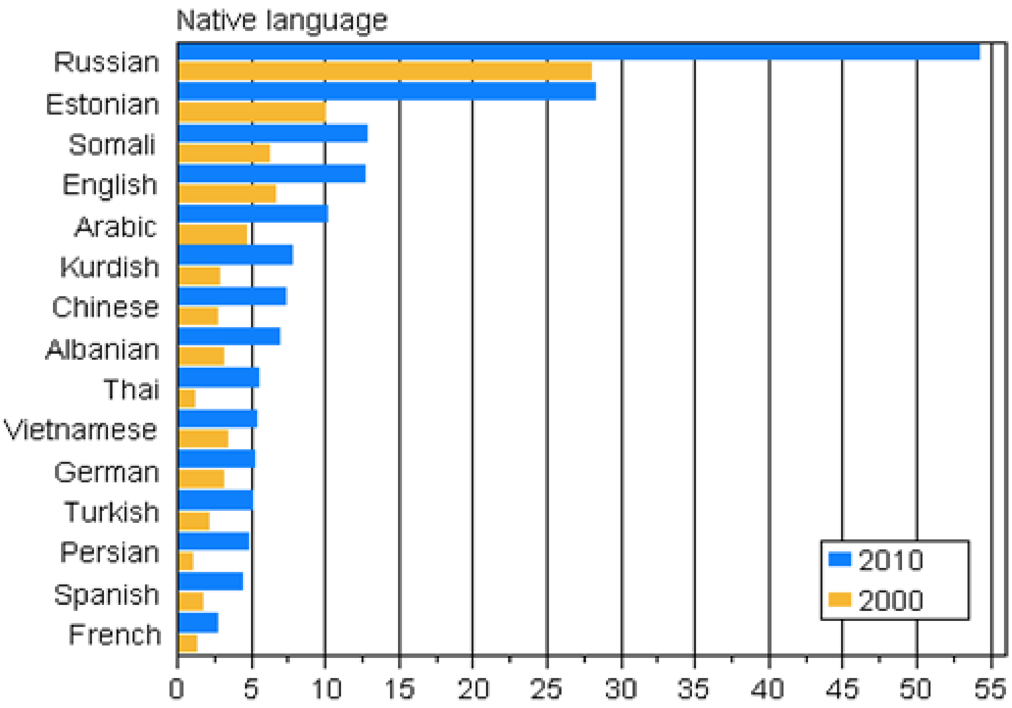


Figure 2. The largest groups by native language 2000 and 2010. Copied from [3]

The data shown in Figure 2 is used to estimate our potential customers in Finland. As we know, the native language is one of the important indicators to show people's original nations. Figure 2 shows that there are big groups of Vietnamese, Thais and Chinese origin in Finland.

Table 1 shows the populations of Vietnames, Thais and Chinese in Finland. The data help to estimate the potential of the market.

Table 1. Population of Vietnames, Thais and Chinese in Finland. Analysis from Figure 3

Nations	Population
Vietnamese	6000
Thais	6500
Chinese	7500

The Important question is how many people live outside the capital region of Finland. We have to calculate and estimate it by a reliable, logical, accurate formula.

Our potential customers are Vietnamese, Thais and Chinese with a total number of 20,000. Generally, Aseanic Trading Oy achieves 350 bills per day, 50% of which are from Vietnamese, 15% from Thais, and 5% from Chinese. Assuming that two people share one bill on average, there should be 700 people shopping in the shop per day. There is another Asian product retailer, Vii Voan, which is the main competitor of Aseanic Trading Oy. Assuming that Vii Voan has 10% more customers than Aseanic Trading Oy, there should be around 1470 people shopping in the capital region of Finland. Therefore, the number of our potential customers is 18530.

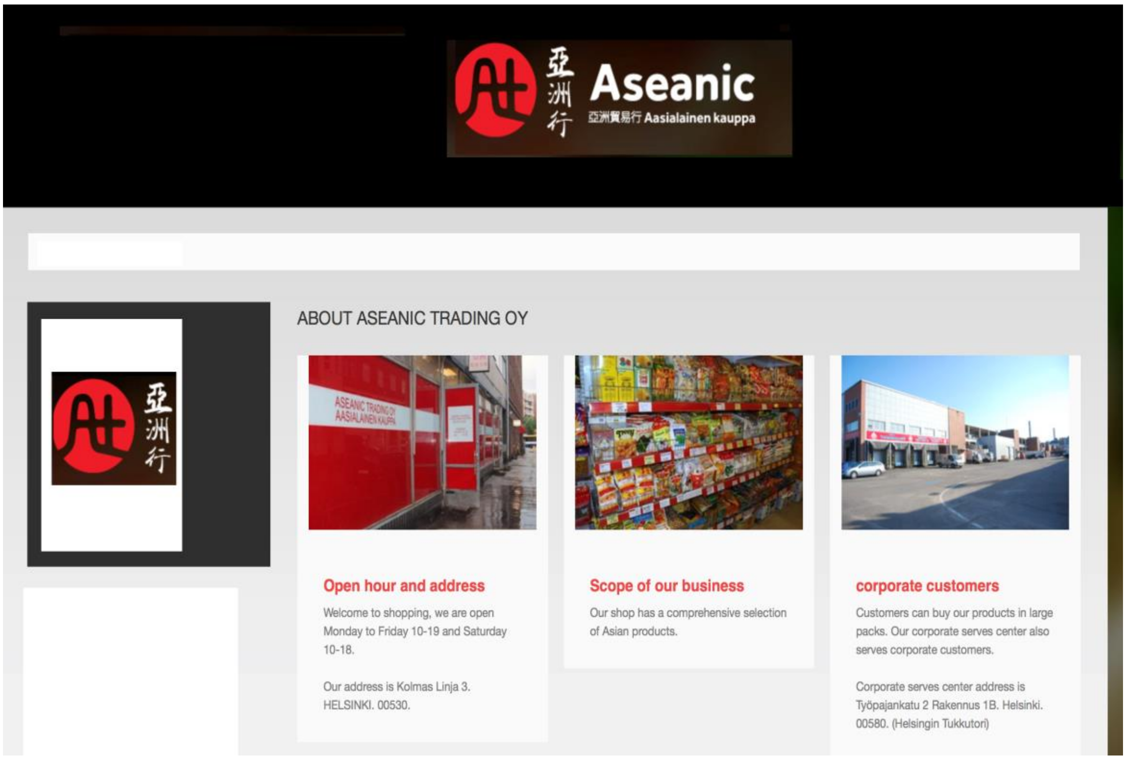
2.2 Building the brand and communicating with customers

One of the aims of the e-commerce platform is building our brand smarter and promoting the products. Another goal is to communicate with customers. That means feedback is very important for retailers. The company orders new products and does advertisement according to customers' requirements. A website could be a good platform for spread of food culture, which includes recipes explaining how to cook traditional Asian food. The website is not only a shopping tool, but also a tool to spread different cultures. We can sell our products as packages that are mentioned in the recipes. This kind of one-step shopping makes customers convenient to choose the right products and save time and money.

2.3 Response to competition

Before the project started, another task was to compare different competitors' websites that include UI, functions, payment gateway and language layout. There are two important competitors. One of them is Vii Voan which was mentioned before. Another one is DFH. DFH is a Chinese product retailer which a business ranges including Chinese people only in Helsinki. It started online shopping at the end of 2013.

Vii Voan is not interested in online shopping. I tried to search Vii Voan online. However, there was no website for this company. Therefore, this project compared with Aseanic Trading Oy's and DFH's websites only.



ABOUT ASEANIC TRADING OY

Open hour and address

Welcome to shopping, we are open Monday to Friday 10-19 and Saturday 10-18.

Our address is Kolmas Linja 3. HELSINKI. 00530.

Scope of our business

Our shop has a comprehensive selection of Asian products.

corporate customers

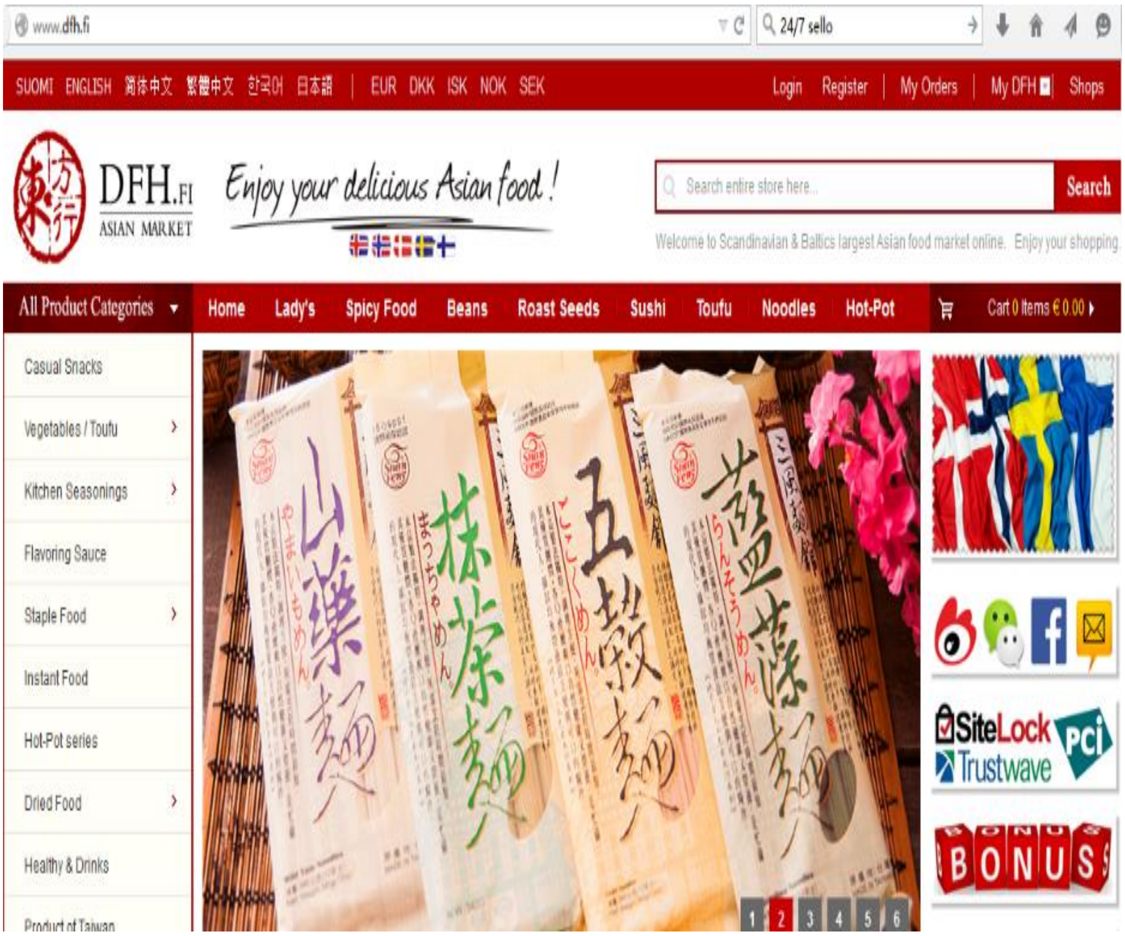
Customers can buy our products in large packs. Our corporate serves center also serves corporate customers.

Corporate serves center address is Työpajankatu 2 Rakennus 1B. Helsinki. 00580. (Helsingin Tukkutori)

Figure 3. Aseanic Trading Oy website before development Copied from [5]

Fingure 3 shows the Aseanic's website which was a static website before. It was based on HTML (Hyper Text Markup Language) and JavaScript technologies. The website contained only some pictures, information and descriptions. However, DFH has been built an online shop already.

Figure 4 is the home page of the DFH online shop. It integrated the multiple-language switching function. That is easy to use by different Asian people who come from different countries. This is an important point for customers who do the shopping online. Most people prefer to use their mother tongue when they read instruction or information. DFH's website integrated a payment module which was supported by Paytrail. Paytrail is a payment gateway the same as POINT, which accepts net bank transactions, credit cards and debit cards.



Welcome to Scandinavian & Baltics largest Asian food market online. Enjoy your shopping.

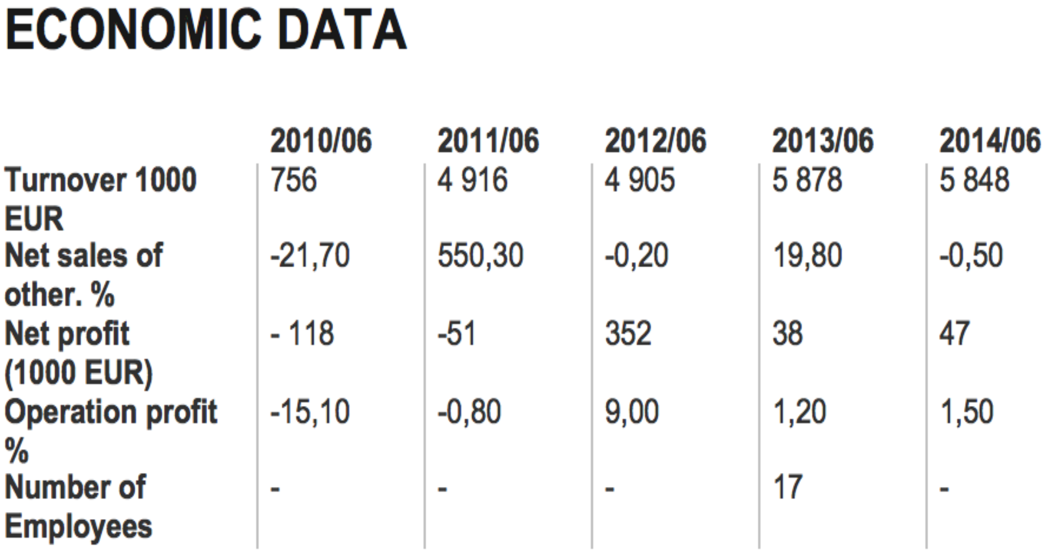
Figure 4. DFH online shop Copied from[4]

As the biggest of Asian food retailers in Finland, Aseanic Trading Oy has to face this competition. However, it has felt high pressure in the e-commerce platform, because DFH has more experience than Aseanic.

2.4 New profit

As a business owner, one always has to think about improving the profit. Among Asian food retailers, there is only one company which tries to develop new service line with the e-commerce platform. Figure 5 shows the Aseanic Trading Oy fiscal year report provided by the financial condition office in Finland.

In traditional retail, the profit cannot rise sharply within two or three years. As figure 5 shows, the turnover of 2014 was 584,800 euros. Compared with 2013, the turnover is almost similar. Section 2.1 discussed the new markets which are in the middle and northern Finland. Building an e-commerce platform is a good solution to capture that market.



ECONOMIC DATA

	2010/06	2011/06	2012/06	2013/06	2014/06
Turnover 1000 EUR	756	4 916	4 905	5 878	5 848
Net sales of other. %	-21,70	550,30	-0,20	19,80	-0,50
Net profit (1000 EUR)	- 118	-51	352	38	47
Operation profit %	-15,10	-0,80	9,00	1,20	1,50
Number of Employees	-	-	-	17	-

Figure 5. Aseanic fiscal year [5]

3 Magento Platform

3.1 Introduction to Magento

As the business owners, the most important thing is to choose the right E-commerce platform for their projects. This paper shows how Magento could be one of the most well-known and widely used e-commerce platform in the world.

First of all, the Magento platform provides versatile content management in the system. For this reason, the page editor can be used by the shop owner or developer to build and customize their website easily and fast. This contents management is user-friendly with the WYSIWYG (what you see is what you get) interface. It helps the shop owner or developer who is not experienced with coding with web programming. [6]

Secondly, the Magento platform is developed by easy-to-use configuration with mobile devices. It is easy to develop the website seamlessly viewable on a small-size smartphone, a large-size smartphone which is more than 6 inches, and a tablet. This accommodates a consumer-shopping trend that continues to move towards mobile usage.

Another advantage is powerful back-end support. The Magento platform can support up to 500,000 products on one site, and handle more than 80,000 orders per hour. That means this website can be used not only for small and middle size e-commerce companies but is also suitable for large enterprises. With Magento, one can easily add prompts to browse related options on the product and check out pages, enhancing user experience and increasing sales potential. [6]

The fourth advantage is easy integration with third-party software. Magento connects the shop owner’s e-commerce platform experience with Ebay ,Paypal, Google shpping, Quickbooks and more. The platform also allows shop owners to add Google Analytics and other such third part integrations to their site.

The fifth reason is most important for shop owners and customers. Online shopping is a virtual environment. How to keep it with high security is very important. The Magento platform keeps the website safe and manages internal access with an option to customize multiple levels of security permissions. PCI data security comes as a standard, Login screens are CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart)–equipped, and a secondary password prompt includes extra defense against unwanted breaches.

For those reason, Magento is an open-source e-commerce platform which is built using the Zend framework. The first version called Bento was published on March 31, 2008 by Varien. This platform uses the MVC (Model-View-Controller) pattern and XML (Extensible Markup Language) configuration files. The core of the code was developed by the PHP programming language and employed the MySQL relational as the database. [6]

There are some different versions in the Magento platform, such as Magento Community Edition, Magento Enterprise Edition and Magent go. [6]

Magento Community Edition is an open-source content management system, which is distributed under the Open Software License (OSL 3.0). OSL 3.0 allows developers to access the Magento Community Edition source code. Developers can modify and customize the software. After development, they can distribute their own derivative works under the OSL 3.0 license. This will lead to a larger community of developers enhancing Magento in ways that are available to all. [6]

For this reason, developers can touch the core of this system with the source code and extend it with personal functions. Generally speaking, the original source code of Magento is software which is provided with complete functions as an online shop. It includes basic functions which are registration, searching, shopping cart and amount calculation. It also shows powerful back-end management for the website, such as adding or deleting products with different regulars, management CMS pages and calculation delivery methods. Developers can extend advanced features and develop more functions customized based on the original source code. The source codes of the features are packaged as .zip files and installed into the Magento platform with “load extended” option which is provided by the back-end management system. In other words, developers modify the source code as a second development which customizes functions and user interface.

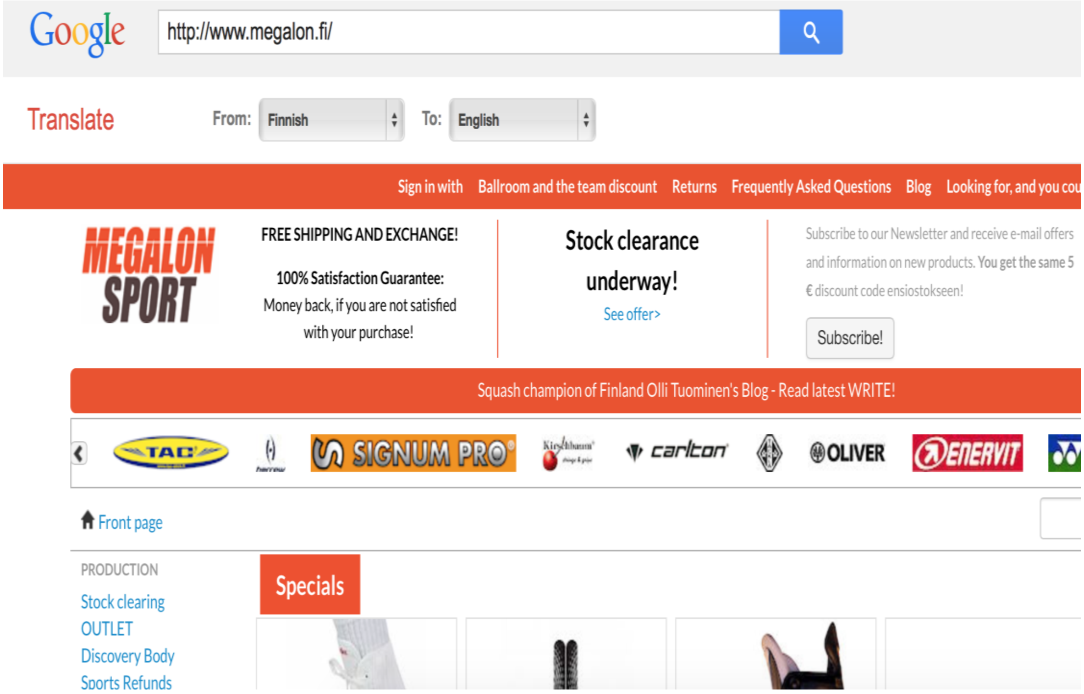
Mgento Enterprise Edition is an advanced version based on Magento Community Edition. Both of them have the same core files. However, Enterprise Edition is not free, because there are more features and functionality provided in this version. Compared with Magento CE version, Enterprise Edition can download third-part feature packages, which are published in Magento marketing for free. Another difference is providing warranties, such as installation introduction, website configuration and technical supporting, by Magento specialists and helpdesk. Magento Enterprise Edition is a design for middle to large size business with a revenue in excess of more than 15 million dollars and 25000 visitors per month. [7]

Magento go is a solution which is supported by Magento based on the cloud. It is a commercial edition for small companies whose products are under 10000 totally. This is a special version, whose package includes several services, such as developing of UI, providing a server and configuring the environment. Customers just need to upload products to the website as the instruction shows. [7]

3.2 Advantage of development with Magento

The Magento is an advanced platform for an e-commerce solution all over the world. Especially, Magento CE is free to use. Developers can modify the source code with PHP, HTML, XML and CSS files to customize functions, themes and features. Developers do not need to write the whole of function codes for an online shopping website, but instead they can modify some features which are required by customers. For example, there are different checkout companies in the world which offer API differently. Developers develop a suitable payment method module which is required and integrate it into the website.

Another reason for developing with the Magento platform is that Aseanic Trading Oy is an Asian food retailer which business is targeted at to different customers who comes from different countries in Asia. A multi-language display is very important. There wre two different solutions considered before I choose Magento. The first solution is to build two or three websites and publish them in different servers. Customers switching languages would jump into another website which is independent.



Google http://www.megalon.fi/

Translate From: Finnish To: English

Sign in with Ballroom and the team discount Returns Frequently Asked Questions Blog Looking for, and you cou

MEGALON SPORT

FREE SHIPPING AND EXCHANGE!

100% Satisfaction Guarantee: Money back, if you are not satisfied with your purchase!

Stock clearance underway!
See offer>

Subscribe to our Newsletter and receive e-mail offers and information on new products. You get the same 5 € discount code ensiostokseen!

Subscribe!

Squash champion of Finland Olli Tuominen's Blog - Read latest WRITE!

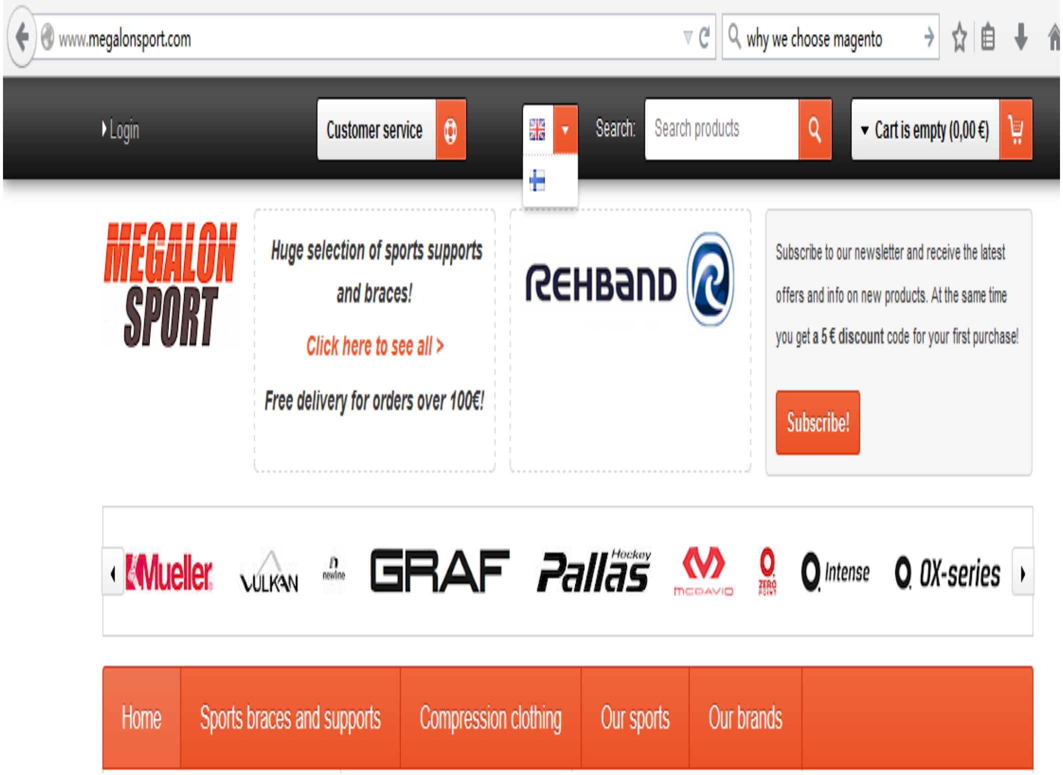
TAC SIGNUM PRO carlton OLIVER ENERVIT

Front page

PRODUCTION
Stock clearing
OUTLET
Discovery Body
Sports Refunds

Specials

Figure 6. Example of language switching on a Finnish website Copied from [6]



www.megalonsport.com why we choose magento

Login Customer service Search: Search products Cart is empty (0,00 €)

MEGALON SPORT

Huge selection of sports supports and braces!
Click here to see all >
Free delivery for orders over 100€!

REHBAND

Subscribe to our newsletter and receive the latest offers and info on new products. At the same time you get a 5 € discount code for your first purchase!

Subscribe!

Mueller VULKAN GRAF Pallas McDAVID Intense OX-series

Home Sports braces and supports Compression clothing Our sports Our brands

Figure 7. Example of language switching on an English website Copied from[7]

This is a typical solution to switch languages between websites. As the figures 6 and 7 show, customers visit www.megalon.fi first and switch into an English version when they click British flag logo. The website jumps into www.megalonsport.fi which is another domain instead of the previous one. However, there is a problem. Customer information, product information and shopping records cannot be synchronized. For example: customers do the shopping at www.megalon.fi with two products, and after that, switching to English version. However, the shopping cart will be empty at www.megalonsport.fi as it for a new visitor. There is another solution which is to develop websites in different languages independently, but to upload those websites into the same server.

Figure 8 shows an implementation for that and tells us how it is working. The user visits the website with a default domain name www.aseanic.fi. The language switching function loads the current address and redirects to the same pages, which display another language when the user clicks the language-switching icons.

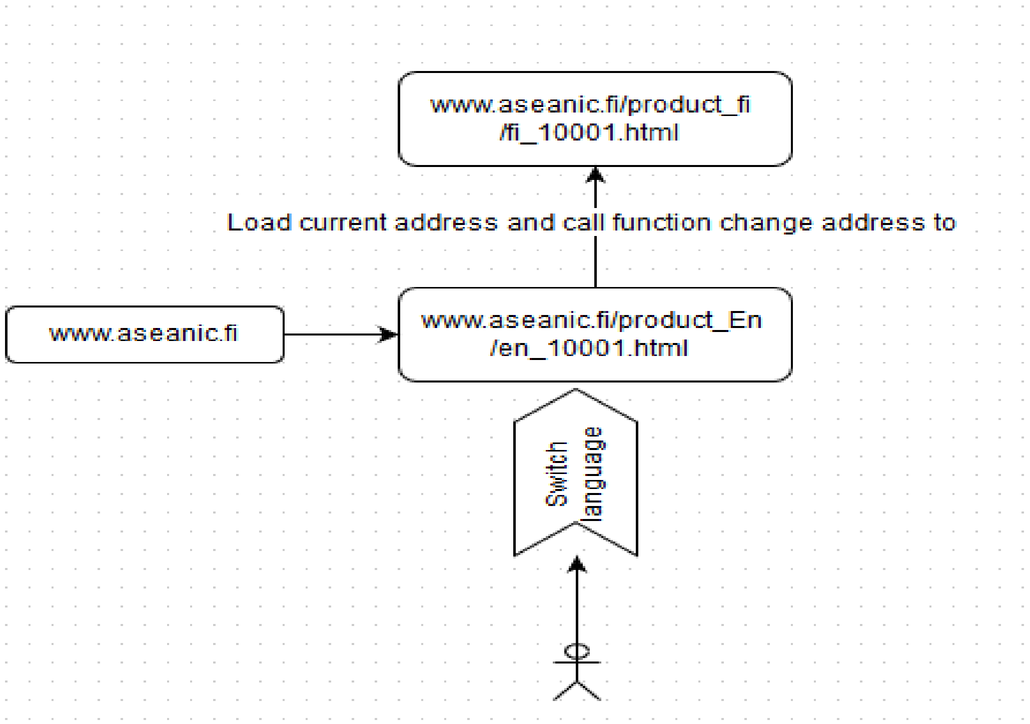

www.aseanic.fi/product_fi
/fi_10001.html
Load current address and call function change address to
www.aseanic.fi
www.aseanic.fi/product_En
/en_10001.html
Switch language

Figure 8. Language switching implementation by redirection

However, it is hard to implement the second solution which mentioned before, because developer must develop the same website in different languages many times. Therefore, all the pages must be rewritten with a different language. That is impossible if more and more languages are integrated in the website.

Magento's language switching function is easy to modify with the "Language switching" module. The "language switching" module will match the "language package" automatically when programmers develop and upload them to server.

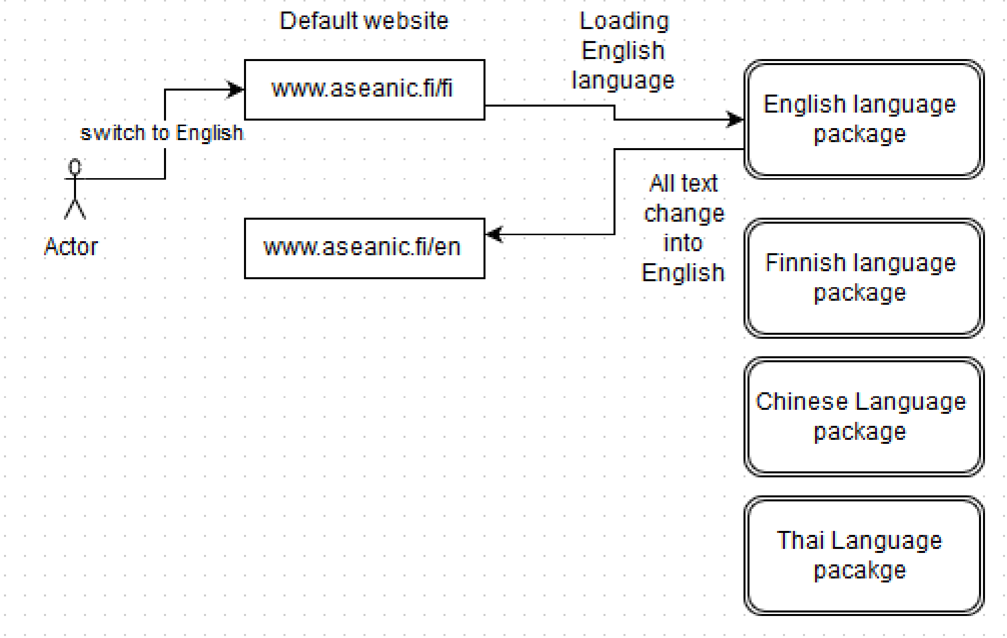


Figure 9. Magento language switching module

Developers do not need to develop the website many times. Therefore, Magento platform improves the development efficiently.

4 UI design and implementation

4.1 UI design

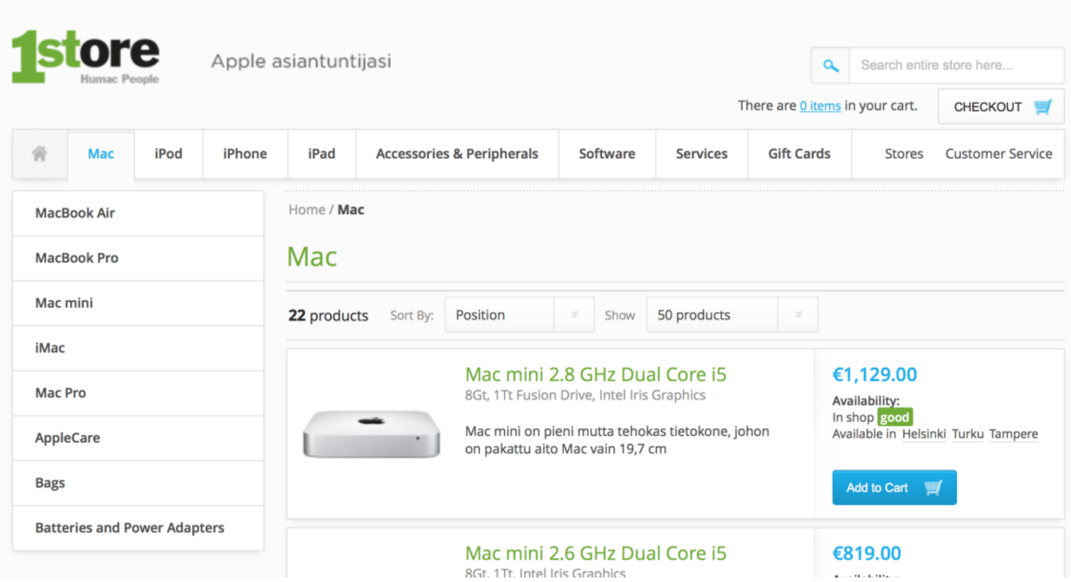
The user interface is one of the most important developments in website modification. The UI design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. The UI brings together concepts from interaction design, visual design, and information architecture design. [8]

There are four elements which should be considered before design. The first one is input control, such as buttons, text fields and checkboxes. Another one is navigational components which are search field, pagination and sliders. The third one is information components, for example, icons, notifications and message boxes, and the last element is containers which depend on what kind of function or diagram is inside.

The best user interface should be easy to understand and intuitive. Keeping the interface simple is very important to users. In other words, using labels or icons is better than text messages. The user will feel more comfortable and understand the usage more quickly. Page layout is next step which should be considered. UI design is to plan a relationship with every item on the page. How and where should those elements be placed on the pages? What color? What size? Generally speaking, the UI is the first impression the user gets while visiting the website. User experience is the first thing in UI development. [8]

In my project and analysis, the B2C e-commerce website includes the following parts:

1. Logos and login/registration on top of the pages
2. Promotion slide in the main part of the page.
3. Products list at the right side or left side
4. Products in the middle of the page with pictures and descriptions.
5. Information box at the bottom of the pages.



There are 0 items in your cart.

Mac mini 2.8 GHz Dual Core i5
8Gt, 1Tt Fusion Drive, Intel Iris Graphics
Mac mini on pieni mutta tehokas tietokone, johon on pakattu aito Mac vain 19,7 cm
€1,129.00

Mac mini 2.6 GHz Dual Core i5
8Gt, 1Tt, Intel Iris Graphics
€819.00

Figure 10 1store online shop's home page Copied from [9]

Fingure 10 is a typical design for an online shopping website. In this case, a common layout and elements make it easy for a user to use and get familiar with our website quickly.

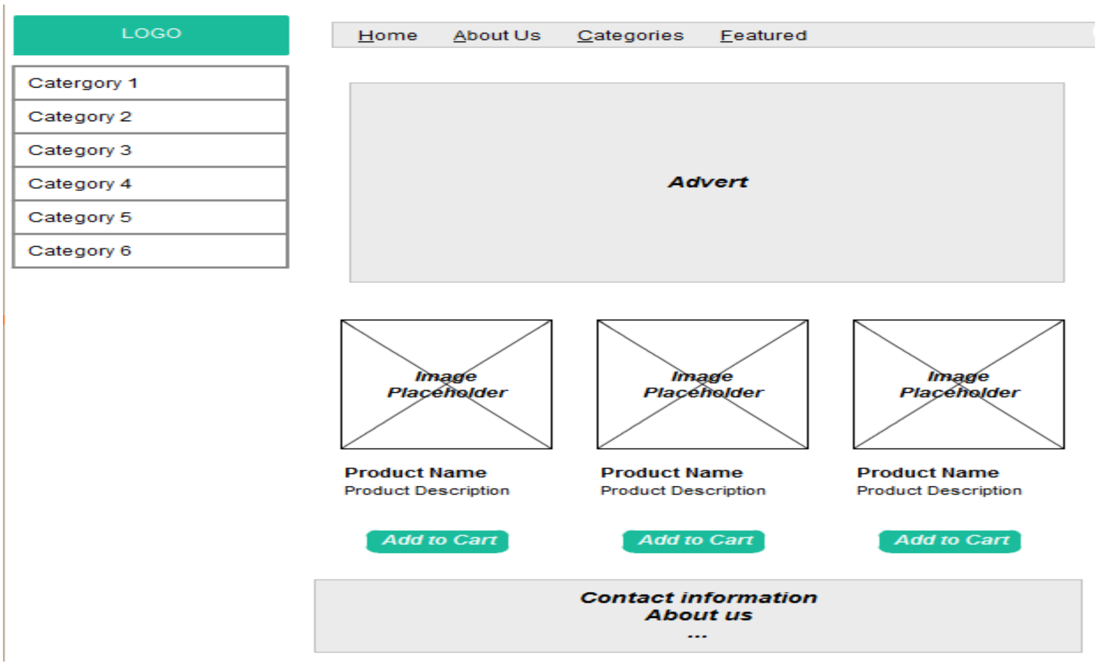


Figure 11. UI design diagram for a new Aseanic Trading Oy website

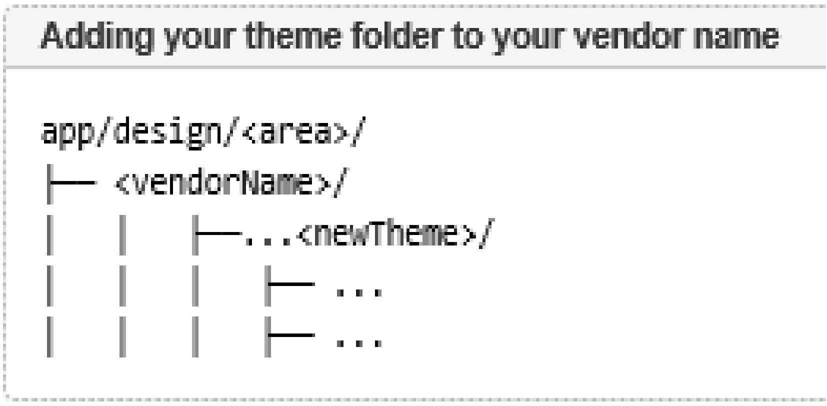
Figure 11 is the main layout of the home page. The category list is on the left side of the home page. On top of the body, there is a navigation bar which is placed in the middle. An advertisement slide window is below that. The product detail layout is in the middle of the body and the footer of the page at the bottom of it.

4.2 DIV AND CSS

Generally, shop owners or programmers are allowed by the Magento CE version to modify the code customizing layout. They develop web page different with CSS, by not only changing colors and components, which are integrated in the original source code, but also by replacing images which are added by the developer. As the instruction shows, programmers edit HTML generated by existing modules with the PHP programming language to adjust PHTML template files. While PHP coding is involved, it is typically copying and pasting small fragments of the PHP code from existing template files into a new template file with differently structured HTML. This can be useful if the existing HTML generated does not have sufficient CSS class names or HTML elements to achieve the presentation change which one would like to achieve. [10]

Magento also allows to modify structurally with the source code which is moving or adding functions. Finally, developers develop new customized functions as an extended module. For example, a user interface module is used in the layout and the payment method module is used in the checkout online.

In the UI development part, the programmer focuses on how to develop a new UI for the website. First of all, the theme file is created at “/app/design/<area>/” in the Magento directory. Figure 12 is a tree list of the theme architecture. It lists different theme files which are integrated in the system. Magento manages to multiple the themes in the back-end platform.

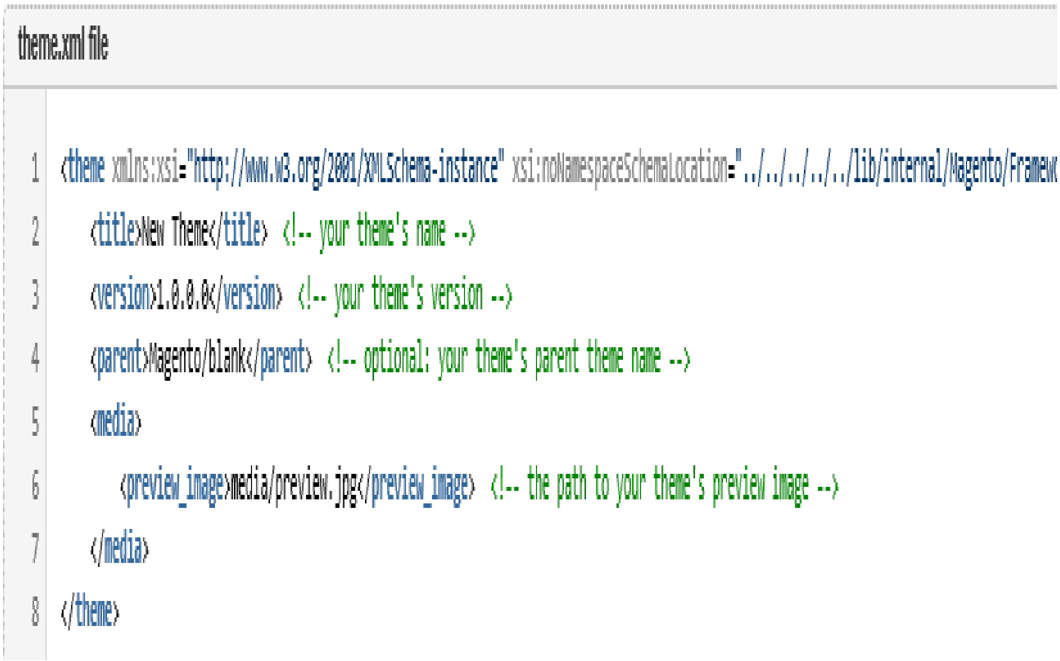


Adding your theme folder to your vendor name

```
app/design/<area>/
├── <vendorName>/
│   │   ├──...<newTheme>/
│   │   │   ├── ...
│   │   │   ├── ...
```

Figure 12. Theme file created.

After that, the theme files, which have been created before, must be declared in theme.xml file as figure 13 shows. Theme.xml file contains theme declaration, such as name, version information, images and title.



theme.xml file

```
<theme xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../../../../lib/internal/Magento/Framew
    <title>New Theme</title> <!-- your theme's name -->
    <version>1.0.0.0</version> <!-- your theme's version -->
    <parent>Magento/blank</parent> <!-- optional: your theme's parent theme name -->
    <media>
        <preview_image>media/preview.jpg</preview_image> <!-- the path to your theme's preview image -->
    </media>
</theme>
```

Figure 13. Declaration of theme

That is the first step to develop a new UI in the Magento platform. The next step is to build a template file with the DIV, CSS and JavaScript technologies.

DIV is DIVision which is used to divide a web page into different layers and parts. The Division tag is used to specify a section within an HTML document. Anything from text to images can be placed within a DIV. Compared between DIV and table in HTML file, DIV is similar to tables. However it is customizable with CSS and loads faster than tables. [11]

CSS is Cascading Style Sheets which is used to mark up of pages. While most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology which is used by most websites to create visually engaging webpages. [12]

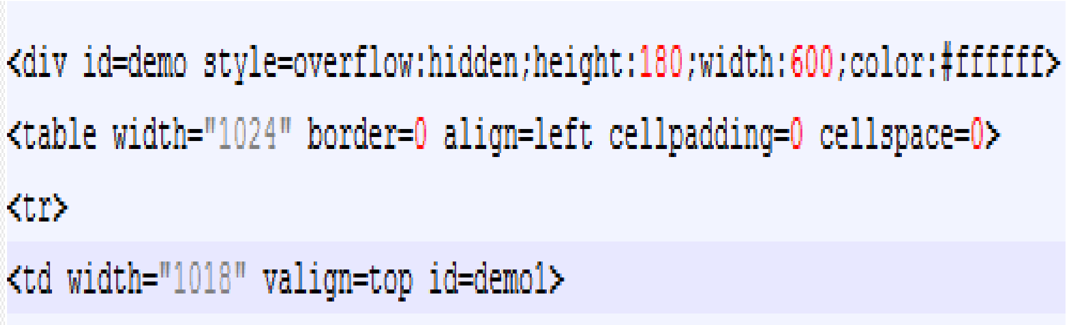
Generally, DIV divides pages into several areas, and CSS gives those sections different styles.

4.3 jQuery in Magento

In front-end development, JavaScript is a dynamic programming language, which is used to add more functions to the webpages. It provides smoother appearance browsers to users. For example: customers visit websites by a PC and mobile phones. However, there are different resolutions between PC and mobile phones. In order to achieve a better browsing effect, programming first checks what kind of device resolution is used. Therefore, the website will be selected to be used as a mobile version or desktop version automatically.

Generally, jQuery is widely in use for front-end development. It is the library of JavaScript, which is designed to help programmers developing various functions with simple code. JQuery is APIs, which are packaged in respective functions. In this way, programmers use the library instead of writting code one by one. Compared jQuery with normal JavaScript code, which are applied in our case with the same function, jQuery is more efficient.

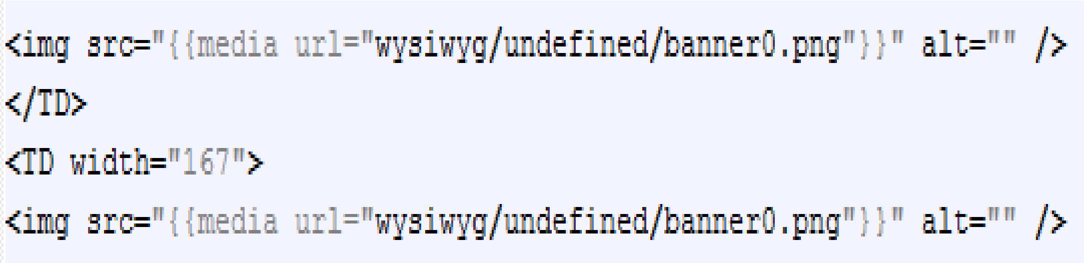
Here is an application in my development as Figure 14 shows. The website will load the picture from the right side across to the left. First of all, the developer builds a table and fills the picture inside.



```
<div id=demo style=overflow:hidden;height:180;width:600;color:#ffffff>
<table width="1024" border=0 align=left cellpadding=0 cellspace=0>
<tr>
<td width="1018" valign=top id=demo1>
```

Figure 14. Implementation insert function with normal JS code – Built a table

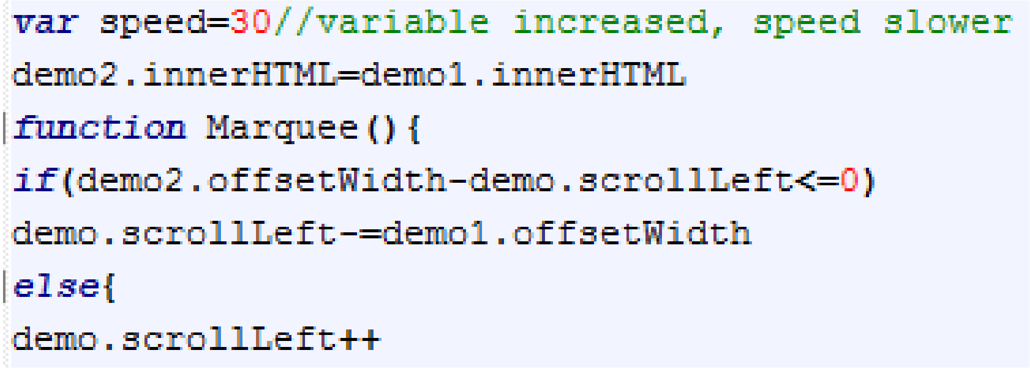
Figure 15 is implementation with the JavaScript code. In this example, the picture's name and path is given by the JavaScript code, so that programming knows where and which picture will be loaded.



```
<img src="{{media url="wysiwyg/undefined/banner0.png"}}" alt="" />
</TD>
<TD width="167">
<img src="{{media url="wysiwyg/undefined/banner0.png"}}" alt="" />
```

Figure 15. Implementation insert function with normal JS code – Picture path and name

The last step is to define the action for the picture, which is inserted in the table. The programmer writes the code for effect and speed when the picture is loaded to the web page. In this case, the picture is loaded from the right side to the left with a slow motion.

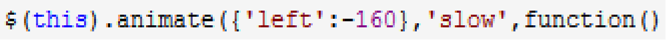


```
var speed=30//variable increased, speed slower
demo2.innerHTML=demo1.innerHTML
function Marquee(){
if(demo2.offsetWidth-demo.scrollLeft<=0)
demo.scrollLeft-=demo1.offsetWidth
else{
demo.scrollLeft++
```

Figure 16. Implement insert function with normal JS code – Actions

Figure 16 is the sample code to implement a loading picture from right to left. It is intricate programming.

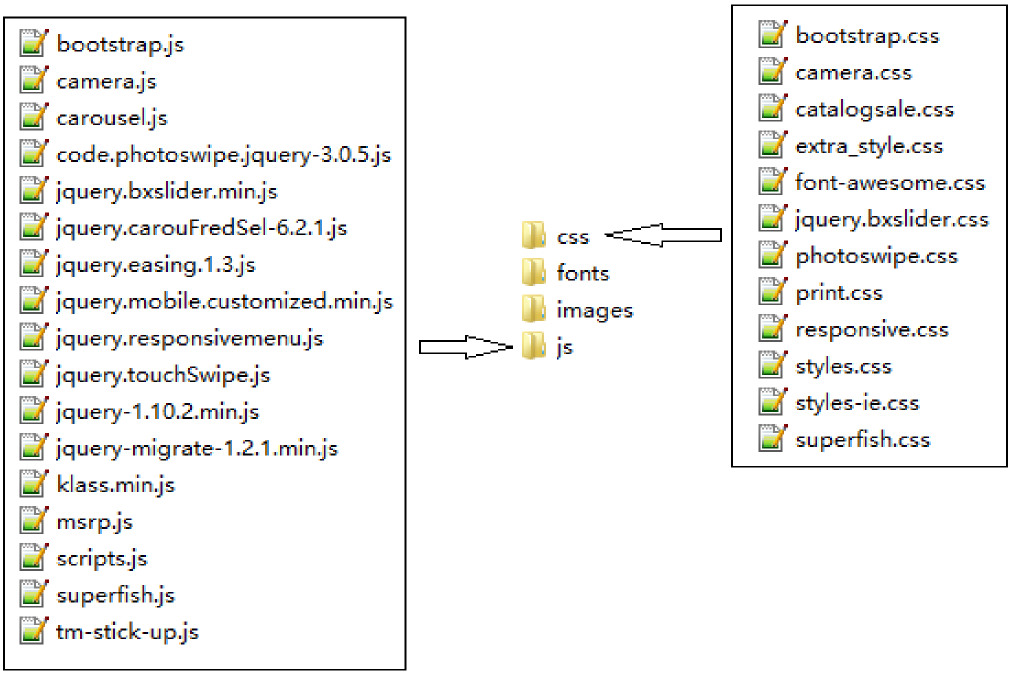
Let us check how JQuery implements the same function. Figure 17 shows that there is only line of code.



```
$(this).animate({'left':-160},'slow',function()
```

Figure 17. Implementation inserts function with jQuery

By comparing the above with different ways for front-end development, we will find that jQuery is more efficient and simpler in our development. In this situation, the programmer develops all functions with jQuery and uploads them in “js” folder, which is ready to package and extend to the Magento system.



bootstrap.js
camera.js
carousel.js
code.photoswipe.jquery-3.0.5.js
jquery.bxslider.min.js
jquery.carouFredSel-6.2.1.js
jquery.easing.1.3.js
jquery.mobile.customized.min.js
jquery.responsivemenu.js
jquery.touchSwipe.js
jquery-1.10.2.min.js
jquery-migrate-1.2.1.min.js
klass.min.js
msrp.js
scripts.js
superfish.js
tm-stick-up.js

css
fonts
images
js

bootstrap.css
camera.css
catalogsale.css
extra_style.css
font-awesome.css
jquery.bxslider.css
photoswipe.css
print.css
responsive.css
styles.css
styles-ie.css
superfish.css

Figure 18. JavaScript function and CSS files ready to package

4.4 Add blocks into pages

Magento likes a modular development platform. Functions or blocks could be developed independently as individual modules. In this section, we will discuss how Magento loads blocks into the page.

Figure 19 shows that there are several containers one the page. Those containers register into the Magento system, which is shown in Figure 20. Containers load a web page template, which includes pictures, text messages and videos inside. All template files are developed by div and CSS. Here we insert the block, which is named “Banner”.

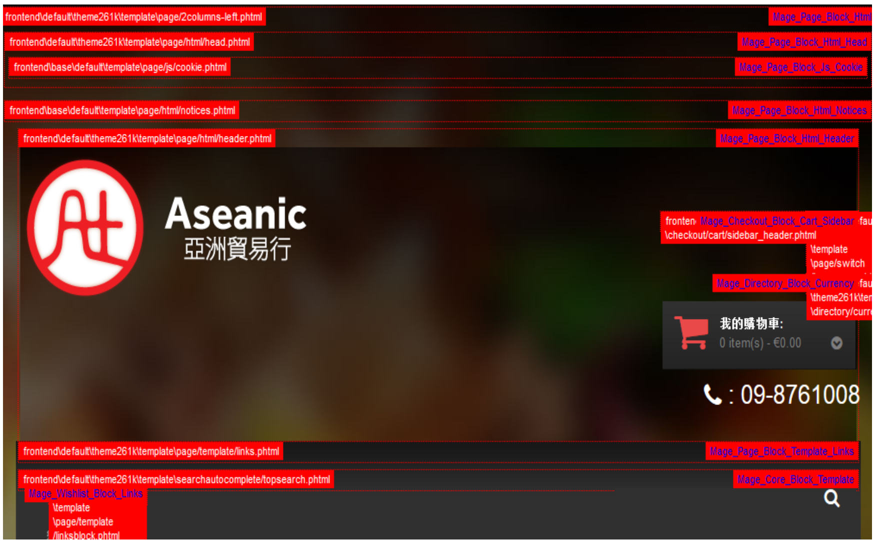

frontend\default\theme261k\template\page/2columns-left.phtml
Mage_Page_Block_Html
frontend\default\theme261k\template\page/html/head.phtml
Mage_Page_Block_Html_Head
frontend\base\default\template\page/js/cookie.phtml
Mage_Page_Block_Js_Cookie
frontend\base\default\template\page/html/notices.phtml
Mage_Page_Block_Html_Notices
frontend\default\theme261k\template\page/html/header.phtml
Mage_Page_Block_Html_Header
Aseanic
亞洲貿易行
fronten Mage_Checkout_Block_Cart_Sidebar fau
\checkout/cart/sidebar_header.phtml
\template
\page/switch
Mage_Directory_Block_Currency fau
\theme261k\ten
\directory/curr
我的購物車:
0 item(s) - €0.00
: 09-8761008
frontend\default\theme261k\template\page/template/links.phtml
Mage_Page_Block_Template_Links
frontend\default\theme261k\template\searchautocomplete/topsearch.phtml
Mage_Core_Block_Template
Mage_Wishlist_Block_Links
\template
\page/template
/linksblock.phtml

Figure 19. Different containers on the page

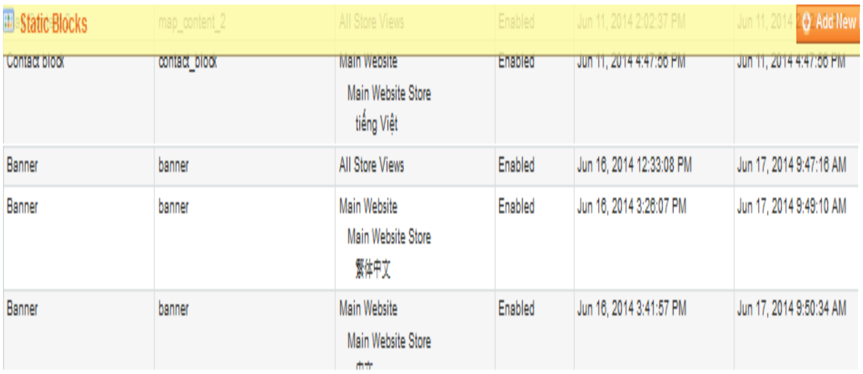
<?php echo $this->getLayout()->

createBlock('cms/block')->

setBlockId('banner')->toHtml() ?>

Listing 1. Example code of load static block in Magento

Listing 1 is an example code in the main page file that loads the “banner” block. This code gets a layout template with the “getLayout()” function, which is defined by Magento first. That places containers in the right place on main page. Function “createBlock()” initializes a static block which one will load. After that, the “setBlocId()” selects block by the ID name which has been registered in the system before. The last step is that programming loads the block which was selected in the previous steps into the HTML template.



	map_content_2	All Store Views	Enabled	Jun 11, 2014 2:02:37 PM	Jun 11, 2014
Contact block	contact_block	Main Website Main Website Store tiếng Việt	Enabled	Jun 11, 2014 4:47:56 PM	Jun 11, 2014 4:47:56 PM
Banner	banner	All Store Views	Enabled	Jun 16, 2014 12:33:08 PM	Jun 17, 2014 9:47:16 AM
Banner	banner	Main Website Main Website Store 繁体中文	Enabled	Jun 16, 2014 3:26:07 PM	Jun 17, 2014 9:49:10 AM
Banner	banner	Main Website Main Website Store 中文	Enabled	Jun 16, 2014 3:41:57 PM	Jun 17, 2014 9:50:34 AM

Figure 20. Banner static block

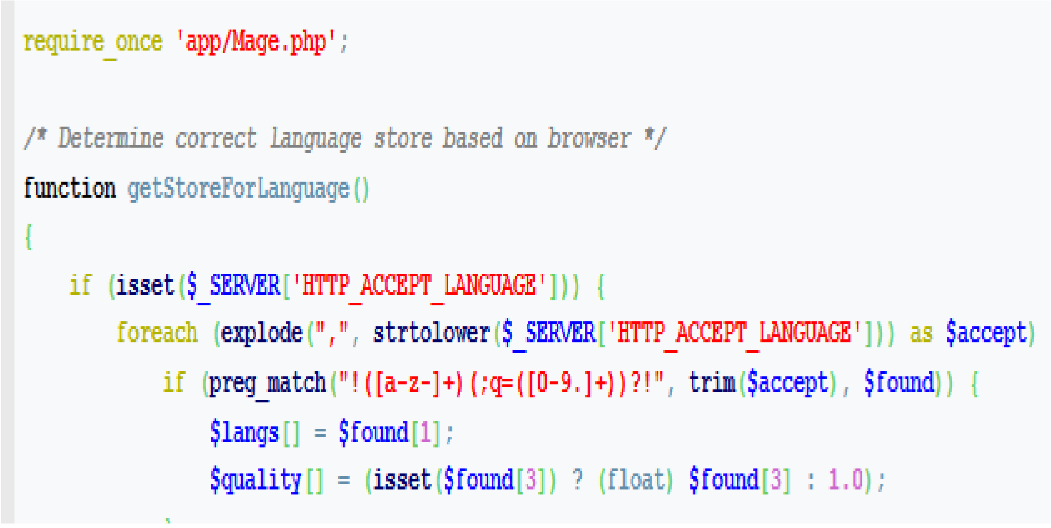
5 Multi-language module

5.1 CSV file

We have discussed the possibilities of how to implement multiple-language switching in the previous chapter. Magento provides the API to make it easy. In the Magento platform, language packages are an independent module which can be extended inside. Language packages are developed by CSV files. CSV is comma-separated values file. The CSV file consists of any number of records, separated by line breaks of some kind; each record consists of fields, separated by some other character or string, most commonly a literal comma or tab. Usually, all records have an identical sequence of fields. [14]

5.2 Multi-language module implementation

In the case of this project, Magento platform language is English. Language packages build mapping with English to transferred languages, like: “log in” is mapping with “kirjaudu sisään”. Developers translate all the keywords which are used on the website and build them into CSV files. Next, they develop the PHP code in the “language switching module” to integrate it with the Magento platform.



```
require_once 'app/Mage.php';

/* Determine correct language store based on browser */
function getStoreForLanguage()
{
    if (isset($_SERVER['HTTP_ACCEPT_LANGUAGE'])) {
        foreach (explode(",", strtolower($_SERVER['HTTP_ACCEPT_LANGUAGE'])) as $accept)
            if (preg_match("!([a-z-]+)(;q=([0-9.]+))?!", trim($accept), $found)) {
                $langs[] = $found[1];
                $quality[] = (isset($found[3]) ? (float) $found[3] : 1.0);
```

Figure 21. Load language package with CSV file

In this way, the Magento website could speak different languages while language packages have been uploaded into the system. Customers switch the languages in all pages and do not worry about the data lost.

6 Payment method developments

6.1 Research and data analysis

Magento is an excellence platform, which has an integrated perfect payment method such as Paypal and Authorize.net. However, that is not enough for all users who want to start online shopping in different countries. For example, in China, there are few people shopping with Paypal online. In our case, we have to consider that our customers who come from different counties of Asian do not use Paypal at all.

Table 2. Feedback of customer with Paypay payment method

Type	Used Paypal	Heard but not use	Never heard
Customer Number	3	79	10

The research data showed that our customers are not interested in Paypal when they do some shopping online. As the developer, I have to think about how to make it easy for our customers to use. In this case, Aseanic hopes to develop a method to accept debit cards, credit cards and netbanks. This is a good solution for customers who want to shop online, because most of them use netbank or debit cards in Finland. In this situation, Aseanic decide to make a contract with a third-part checkout company as a payment gateway. Therefore, customers pay their orders to checkout company first when the website confirms transactions. Aseanic Trading Oy will get the payment after one day.

6.2 Implementation for payment module

The Magento platform is a smart system, which is accepted by the developer to add or edit source code by extended functions. It is similar to inheritance in oriented object programming. Developers modify files, which are named system.xml, config.xml, and build template for a new payment method as a module.

First of all, system.xml files must be created in Magento “etc” folder that to give the name and label of the module.

```
<pay translate="label" module="Payment_Module">
<label>Aseanic payment method</label>
```

Listing 2. Example code declare payment module

That is the first step to define the name of the module. Magento will configure the new payment module with the name “Payent _Module” and display “Aseanic payment method” when the customer go to the payment page. After that, config.xml is created to define the default values for this module.

```
<model> Payment_Module / Payment_Module </model>
 <order_status>On</order_status>
<title>Aseanic payment method </title>
```

Listing 3. Example code with config.xml file

The next step is to create the class for the payment method which extends from “extends Mage_Payment_Model_Method_Abstract”. All the Magento payment functions extend from payment architectures, which are integrated by the Magento platform. Programmers develop more functions. However they must be based on those architectures. Here we created a new module whose name is “Aseanic_pay_model” but have to extend with the existing model which is integrated in the core of Magento.

```
        classAseanic_pay_modelextends
        Mage_Payment_Model_Method_Abstract
        {
        protected $_code = Payment_Module;
…
```

Listing 4. Example code with extended class

7 Environment and Testing

7.1 Testing environment

Magento platform is developed with the PHP language and we have to build environments for it running and testing. There are three environments that have to install: PHP IDE, database server and server for the website. The programmers could choose different software that depends on what system they have. WAMP is a web service solution package working on the Windows platform. WAMP means Apache server, MySQL server and PHP environment running on the Windows platform. In this case, the WAMP environment was installed in the PHPstudy which integrated all the IDE in the package.

As the Aseanic website is a software application, testing tasks should be done before publishing it on the Internet. As for the functions which were developed with extent modules, tests would focus on UI testing, language switching testing, shopping function testing and payment method testing.

7.2 UI Testing

The target of UI testing is to check that all the pages are running well on the browser. The first thing is to visit all the pages that will check DIV, CSS code right or not. DIV and CSS codes are used to load containers into the page and give it style. Because of the complex CSS file in Magento, programmers should be careful when they define the container in the pages. Sometimes, in development, more than one functions have a combination relationship which does some operation with one container and the browser cannot understand which operation should be done. In this situation, the tester will visit each of the webpages to check the layout is like pre-designed or not.

For another reason, the source code will be displayed in different style on different browsers. Another possibility is that the browser version is different. Compatibility must be considered in the testing process. In general, website testing will choose popular browsers such as: Internet explorer, Firefox, Safari or Chrome. Three different versions of the same browser should be chosen to test which could make testing reliable and accurate.

7.3 Language switching testing

The goal of language switching testing is to verify it the language package work or not. Magento’s default language is English. The testing process is to switch into a different language on the website and check that all the text has been translated or not. Sometimes, a problem occurs that the text was still an English word which cannot be translated, but if switched into another language, the text has been translated. The reason for this situation is that the Magento system loads the language package in CSV files. Keywords are mapped each other one by one. However, assuming that one of the keyword is mapped twice in the CSV file, the system cannot load both words. The system will set this keyword to an unavailable status and load the default language instead. Programmers will fix this issue when they delete one of them.

7.4 Shopping function testing

The objective of shopping function testing includes the following tasks:

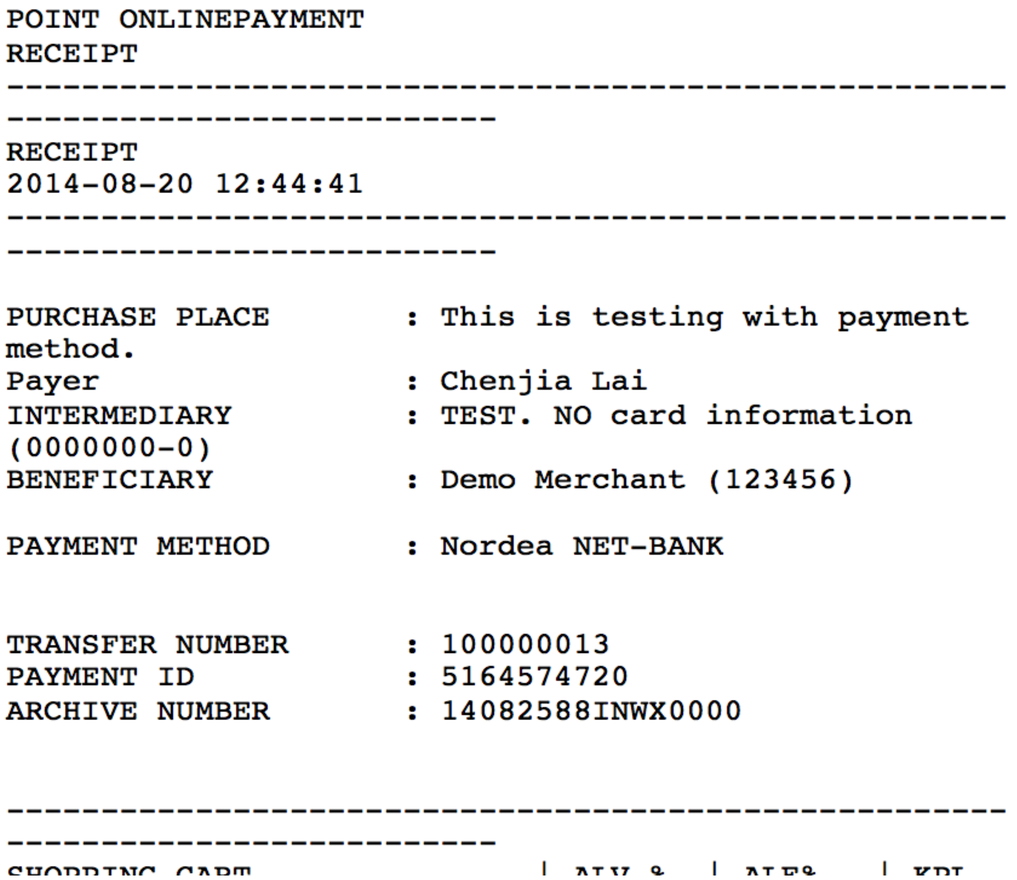
1. The product could be selected or not
2. The price calculation function works correct or not
3. The payment method works or not

Because the developer has changed the core of the source code in the Magento platform, it might affect the integration files in the system. This testing includes visiting the website as customers, selecting the products and checking out. Because of limited time and resources, it was impossible to test lots times. In this case, the website had be tested by three persons and each of them does shopping 50 times.

7.5 Payment method testing

Payment method testing is most important in this project, as we know that a company or a customer will lose money if this function does not work well. In this case, we chose a third-part company as the payment method gateway and we had to test this method with POINT together. At the beginning of testing, we informed POINT and asked for a testing account and IP to start this process. After that, we configured the private key and public key which we got from POINT on the website with the administer access right. At the same time, our IP was promised to access POINT TESTING ENVIRONMENT. That meant the payment node had been opened by POINT.

As the POINT's instruction shows, page will be redirected to the "POINT PAYMENT page, when the customer clicks the checkout button on website. There are several options on the screen such as: pay by credit card, pay by debit card, pay by Nordea netbank and pay by Sampo netbank. In the testing environment, we do not need to pay the order with our money since only virtual payment occurs. The net bank password could be 1234 and the account as default with any words. If the order is paid successfully, the customer will get an E-mail from POINT to show the information as in figure 28.



```
POINT ONLINEPAYMENT
RECEIPT
-----------------------------------------------------
--------------------------
RECEIPT
2014-08-20 12:44:41
-----------------------------------------------------
--------------------------

PURCHASE PLACE       : This is testing with payment
method.
Payer                : Chenjia Lai
INTERMEDIARY         : TEST. NO card information
(0000000-0)
BENEFICIARY          : Demo Merchant (123456)

PAYMENT METHOD       : Nordea NET-BANK


TRANSFER NUMBER      : 100000013
PAYMENT ID           : 5164574720
ARCHIVE NUMBER       : 14082588INWX0000


-----------------------------------------------------
--------------------------
SHOPPING CART               | ALV %  | ALE%    | KPL
```

Figure 28. Part of payment confirmation in testing environment

In this case, the tester will test all the payment methods that include different banks in Finland. That will ensure that the payment method works perfect.

8 Result and discussion

The result of testing shows some possible problems, like capability with different browsers and multi-language switching function cannot work perfect. The programmer fixes those issues to make customer experience better.

The development with the Magento platform is more efficient. Generally speaking, software development is to design architecture first, and then to develop features. Each of features must be tested many times to ensure the reliability of the software. That is a problem where the testing time is much longer than in development. Sometimes, new developments have to wait for the previous feature.

9 Conclusions

The final task of the project was to build a website for a company who wanted to start an e-shop on the Internet. The website has now been used in the Aseanic Trading Oy half year.

The result of this development reached the expectations. The customized UI was perfect and the language switching function worked very well. However, because of the limited time and schedule, there are more modules that could not be implemented in this version, such as calculation for the total price if more than 28 euros or not. That is the next promotion for customers who have bought something at 28 euros that the delivery is free. The company is considering adding more functions to the website in future versions.

REFERENCES

1. E-Commerce and online payment trends in Finland [online]. Maksuturva Oy.
   URL:https://www.maksuturva.fi/en/suomen-maksuturva-oy/news/e-commerce-and-online-payment-trends-in-finland/
   Accessed 13 April 2015

2. Kesko will shut down eight anttila sotres 8 [online]. HELSINKI TIMES
   http://www.helsinkitimes.fi/business/10026-kesko-closes-eight-anttila-stores-mulls-new-cuts.html
   Aceeseed 13 April 2015

3. The largest groups by native language 2000and 2010 [online]. Statistics Finland.
   URL:http://www.stat.fi/til/vaerak/2010/vaerak_2010_2011-03-18_kuv_003_en.html
   Accessed 15 April 2015

4. DFH website user interface [online]. DHF.FI
   URL:http://www.dfh.fi/
   Accessed 15 April 2015

5. Aseanic trading oy taloustiedot [online]. Taloussanomat
   http://yritys.taloussanomat.fi/y/aseanic-trading-oy/helsinki/0977659-3/
   Accessed 17 April 2015

6. Magento license FAQ [online]. Magento INC.
   https://magento.com/legal/licensing
   Accessed 17 April 2015

7. About Magento [online]. Stackoverflow.com
   http://stackoverflow.com/tags/magento/info
   Accessed 17 April 2015

8. User interface design basics [online]. Usability.gov
   http://www.usability.gov/what-and-why/user-interface-design.html
   Accessed 17 April 2015

9. User interface and layout [online]. 1store oy
   http://www.1store.fi/
   Accessed 17 April 2015

10. Frontend developer’s guide [online]. Magento INC.
    https://wiki.magento.com/display/MAGE2DOC/Frontend+Developer%27s+Guide
    Accessed 19 April 2015

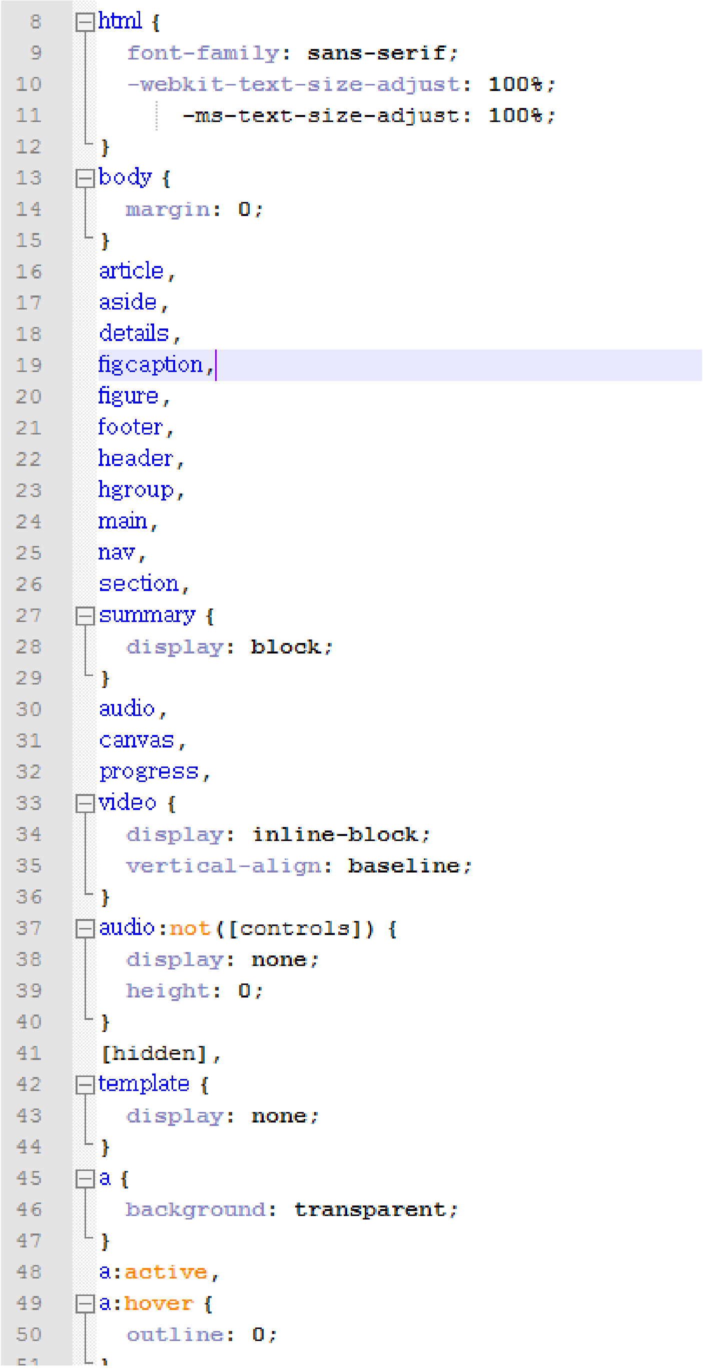
11. CSS and DIV tutorial [online]. GOSTATS.com
    http://gostats.com/resources/css-div-tutorial.html
    Accessed 19 April 2015

12. Web-based mobile apps of the future using HTML5, CSS and JavaScript [online]. Htmlgoodies. By Scott Clark
    http://www.htmlgoodies.com/beyond/article.php/3893911/Web-based-Mobile-Apps-of-the-Future-Using-HTML-5-CSS-and-JavaScript.htm
    Accessed 19 April 2015

13. Explain for CSV file [online]. wikipedia
    http://en.wikipedia.org/wiki/Comma-separated_values
    Accessed 20 April 2015

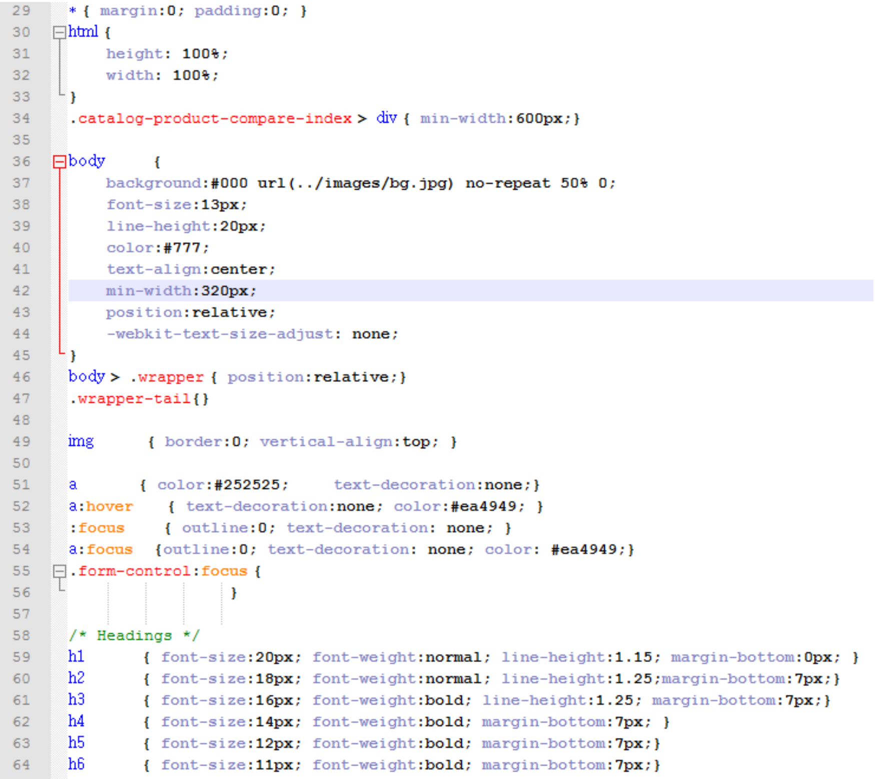
Appendices:

Appendix 1: Bootstrap.css file code



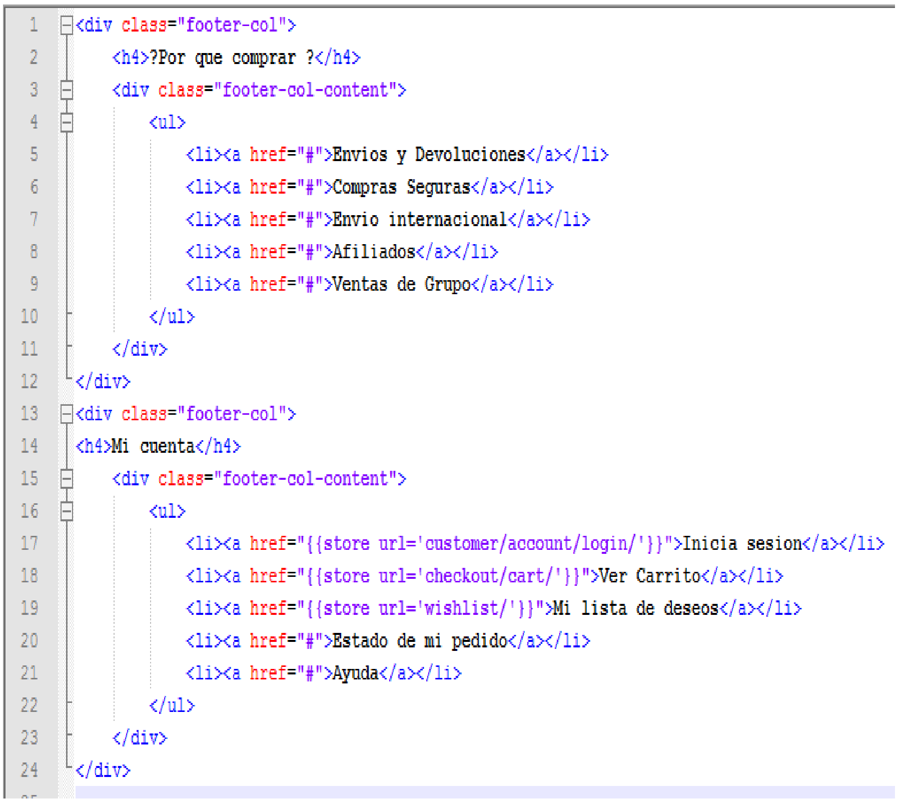
```
html {
  font-family: sans-serif;
  -webkit-text-size-adjust: 100%;
      -ms-text-size-adjust: 100%;
}
body {
  margin: 0;
}
article,
aside,
details,
figcaption,
figure,
footer,
header,
hgroup,
main,
nav,
section,
summary {
  display: block;
}
audio,
canvas,
progress,
video {
  display: inline-block;
  vertical-align: baseline;
}
audio:not([controls]) {
  display: none;
  height: 0;
}
[hidden],
template {
  display: none;
}
a {
  background: transparent;
}
a:active,
a:hover {
  outline: 0;
}
```

Appendix 2: Part of CSS for main layout code



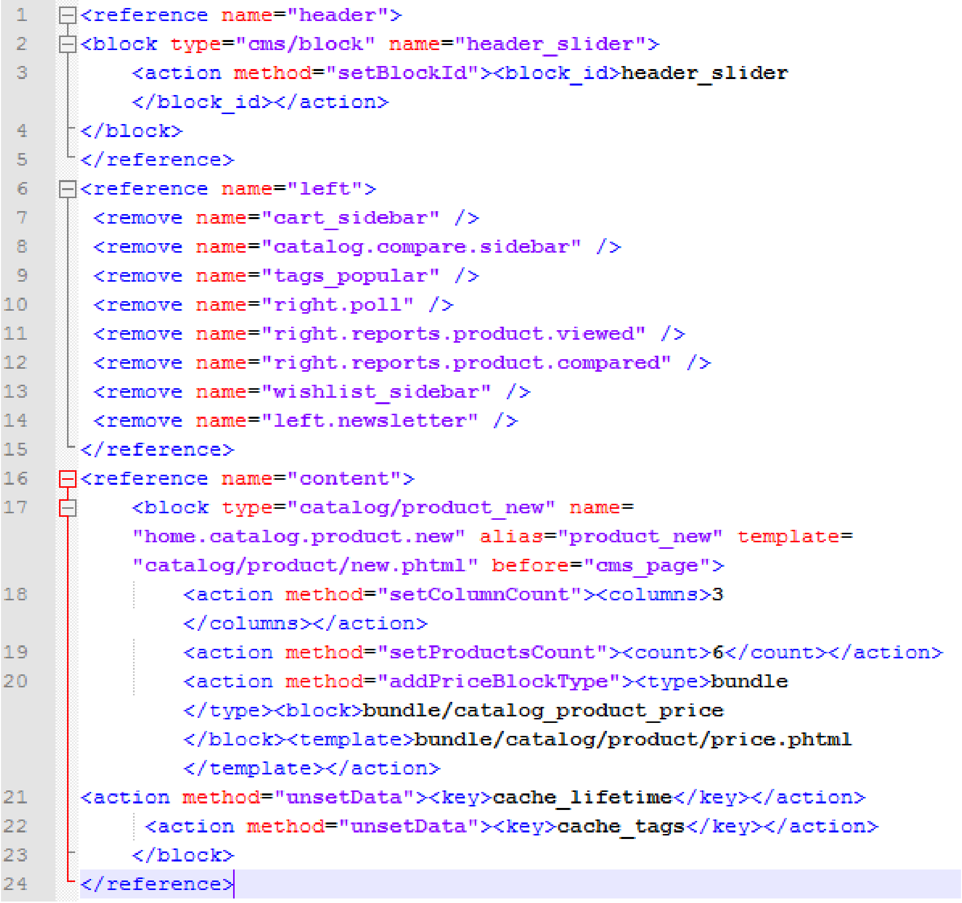
```css
* { margin:0; padding:0; }
html {
    height: 100%;
    width: 100%;
}
.catalog-product-compare-index > div { min-width:600px;}

body      {
    background:#000 url(../images/bg.jpg) no-repeat 50% 0;
    font-size:13px;
    line-height:20px;
    color:#777;
    text-align:center;
    min-width:320px;
    position:relative;
    -webkit-text-size-adjust: none;
}
body > .wrapper { position:relative;}
.wrapper-tail{}

img       { border:0; vertical-align:top; }

a        { color:#252525;     text-decoration:none;}
a:hover     { text-decoration:none; color:#ea4949; }
:focus      { outline:0; text-decoration: none; }
a:focus   {outline:0; text-decoration: none; color: #ea4949;}
.form-control:focus {
                 }

/* Headings */
h1       { font-size:20px; font-weight:normal; line-height:1.15; margin-bottom:0px; }
h2       { font-size:18px; font-weight:normal; line-height:1.25;margin-bottom:7px;}
h3       { font-size:16px; font-weight:bold; line-height:1.25; margin-bottom:7px;}
h4       { font-size:14px; font-weight:bold; margin-bottom:7px; }
h5       { font-size:12px; font-weight:bold; margin-bottom:7px;}
h6       { font-size:11px; font-weight:bold; margin-bottom:7px;}
```

Appendix 3: Example template for footer of the website code



```
<div class="footer-col">
    <h4>?Por que comprar ?</h4>
    <div class="footer-col-content">
        <ul>
            <li><a href="#">Envios y Devoluciones</a></li>
            <li><a href="#">Compras Seguras</a></li>
            <li><a href="#">Envio internacional</a></li>
            <li><a href="#">Afiliados</a></li>
            <li><a href="#">Ventas de Grupo</a></li>
        </ul>
    </div>
</div>
<div class="footer-col">
<h4>Mi cuenta</h4>
    <div class="footer-col-content">
        <ul>
            <li><a href="{{store url='customer/account/login/'}}">Inicia sesion</a></li>
            <li><a href="{{store url='checkout/cart/'}}">Ver Carrito</a></li>
            <li><a href="{{store url='wishlist/'}}">Mi lista de deseos</a></li>
            <li><a href="#">Estado de mi pedido</a></li>
            <li><a href="#">Ayuda</a></li>
        </ul>
    </div>
</div>
```

Appendix 4: Home page template example code



```
<reference name="header">
<block type="cms/block" name="header_slider">
    <action method="setBlockId"><block_id>header_slider
    </block_id></action>
</block>
</reference>
<reference name="left">
 <remove name="cart_sidebar" />
 <remove name="catalog.compare.sidebar" />
 <remove name="tags_popular" />
 <remove name="right.poll" />
 <remove name="right.reports.product.viewed" />
 <remove name="right.reports.product.compared" />
 <remove name="wishlist_sidebar" />
 <remove name="left.newsletter" />
</reference>
<reference name="content">
    <block type="catalog/product_new" name=
    "home.catalog.product.new" alias="product_new" template=
    "catalog/product/new.phtml" before="cms_page">
        <action method="setColumnCount"><columns>3
        </columns></action>
        <action method="setProductsCount"><count>6</count></action>
        <action method="addPriceBlockType"><type>bundle
        </type><block>bundle/catalog_product_price
        </block><template>bundle/catalog/product/price.phtml
        </template></action>
<action method="unsetData"><key>cache_lifetime</key></action>
     <action method="unsetData"><key>cache_tags</key></action>
    </block>
</reference>
```