Yu Lin

Online store based on MAGENTO E-COMMERCE

Technology and Communication

2015
ABSTRACT

Author: Yu Lin
Title: Online store based on Magento E-Commerce
Language: English
Pages: 
Name of Supervisor: Smail Menani

E-commerce application is an important part of web applications and Magento is the most popular e-commerce development platform.

This bachelor's thesis is made for Fusion Oy. The purpose of this project is to enable e-commerce for Fusion Oy business by implementing the Magento e-commerce framework.

The biggest challenge is to make Magento local. The tasks involved in this project, includes the study of Magento module development, Itella posti web service, Paytrail web service. The transport module has been made to use Itella posti shipment service. The shipment module has been made to use online payment service of Nordic region banks such as Nordea bank and Danske bank.

KEYWORDS:

Responsive design, Magento, E-commerce, localization
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>URL</td>
<td>Universal Resource Locator</td>
</tr>
<tr>
<td>XML-RPC</td>
<td>XML Remote Procedure Call</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>B2C</td>
<td>business to customer</td>
</tr>
<tr>
<td>B2B</td>
<td>business to business</td>
</tr>
<tr>
<td>CMS</td>
<td>Content Management System</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>MVC</td>
<td>Model-View-Controller</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENT

I want to express my gratitude to everyone who supported me in the final thesis.

I am thankful for my Partner, Tefu Lin. He is my best friend in life. We have a pleasant experience of project cooperation.

I am sincerely grateful to my wife for her encouragement, She was always there cheering me up and stood by me through the good times and bad.

I would also like to thank all my colleagues in Soprano. I feel very lucky to work with a group of creative developers. They helped me to develop my background in frontend, backend, agile development, etc.

In addition, I express my warm thanks to Dr. Smail Menani for his support, guidance and valuable advices to improve this work.
# TABLE OF CONTENTS

1  INTRODUCTION .................................................................................................... 6

2  MAGENTO OVERVIEW ............................................................................................. 7
   2.1  WHAT IS MAGENTO.......................................................................................... 7
   2.2  FEATURES......................................................................................................... 9
       2.2.1  Responsive design.................................................................................... 9
       2.2.2  Powerful admin panel.............................................................................. 10
       2.2.3  WEB service............................................................................................ 16
       2.2.4  Multi websites & Multi stores ................................................................. 21
       2.2.5  Cache and index management................................................................. 22
       2.2.6  Enterprise edition VS Community edition.............................................. 24
       2.2.7  Import and Export.................................................................................... 25

3  REQUIREMENT ANALYSIS....................................................................................... 26
   3.1  REQUIREMENT.................................................................................................. 26
       THE FOLLOWING ARE THE REQUIREMENTS OF THE WEB-SHOP APPLICATION: .......... 26
   3.2  ANALYSIS......................................................................................................... 26

4  IMPLEMENTATION..................................................................................................... 27
   4.1  MAGENTO INSTALLATION ............................................................................ 27
   4.2  FRONTEND DESIGN.......................................................................................... 32
   4.3  CONFIGURATION.............................................................................................. 34
   4.4  MODULE DEVELOPMENT................................................................................ 38
1 INTRODUCTION

With the popularity of e-commerce, a large number of traditional industries are turning to e-commerce to develop new markets, such as clothing, pharmaceutical, construction, etc.

The demand prompted competition, with a large number of e-commerce platforms comes out. It is very important to understand which platform is more suitable for consumer and also easy for developers to implement.

Magento is an eBay company; it is also the most popular open source e-commerce platform. Magento gives the consumer convenient shopping process, a variety number of payment methods, and the best user experience. On the other hand, modular design, rich web service interface and open source code also give developers a very good technical support.
2 MAGENTO OVERVIEW

2.1 What is Magento

Magento is a rapidly growing e-commerce system. It is based on PHP and MySQL, the system has a modular architecture and rich functionality. It offers lots of API to make third-party software integration much easier.

Magento is also an open source software, which Fully comply with the open source OSL 3.0 protocol. Magento development teams update the version almost every week, which brings the encouragement from many supporters.

Magento has attracted a large number of small and medium size enterprises. /1/ 

Figure 2.1 Most popular e-commerce software (Top 100K sites), April 2014.

Especially in the European market, Magento has become the first choice for many small and medium size enterprises. In the USA, China, Japan, There are a lot of
famous B2C, C2C platforms, such as Taobao, Amazon, and eBay. But in Europe market, consumption is relatively decentralized. The prosperity of small and medium size enterprises promote the development of open source e-commerce software. Unlike those giant e-commerce platforms, Magento can make e-commerce website more personalized in promotion rules, shopping patterns etc.

### Top ranking Internet-retailers in Europe

This ranking is based on 2013 turnover figures.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name company</th>
<th>Country of origin</th>
<th>Online turnover in mio € in Europe</th>
<th>Growth 2013/2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Amazon</td>
<td>USA</td>
<td>19,050</td>
<td>14%</td>
</tr>
<tr>
<td>2</td>
<td>Otto</td>
<td>Germany</td>
<td>6,145</td>
<td>8%</td>
</tr>
<tr>
<td>3</td>
<td>Apple</td>
<td>Luxembourg</td>
<td>2,944</td>
<td>9%</td>
</tr>
<tr>
<td>4</td>
<td>Tesco</td>
<td>UK</td>
<td>3,200</td>
<td>19%</td>
</tr>
<tr>
<td>5</td>
<td>Home Retail Group</td>
<td>UK</td>
<td>2,641</td>
<td>0%</td>
</tr>
<tr>
<td>6</td>
<td>Kering (Redcats)</td>
<td>France</td>
<td>2,400</td>
<td>21%</td>
</tr>
<tr>
<td>7</td>
<td>Dixons</td>
<td>UK</td>
<td>1,662</td>
<td>16%</td>
</tr>
<tr>
<td>8</td>
<td>CDiscount</td>
<td>France</td>
<td>1,782</td>
<td>2%</td>
</tr>
<tr>
<td>9</td>
<td>Zalando</td>
<td>Germany</td>
<td>1,762</td>
<td>52%</td>
</tr>
<tr>
<td>10</td>
<td>Vente-Privee</td>
<td>France</td>
<td>1,600</td>
<td>23%</td>
</tr>
</tbody>
</table>

Figure 2.2 Top ranking internet-retailers in Europe
2.2 Features

2.2.1 Responsive design

The newest Magento Enterprise and Community edition is released with a responsive theme. By 2015, mobile Internet data traffic will surpass PC traffic side. With the rapid development of mobile and tablet, responsive design has been an essential element of the website. Following are the main features:

- Native app VS Responsive design
  1. Native need more programming skills (Objective-C, Java, .NET)
  2. Native app has strict and complex rules to publish, Responsive design websites only open a browser and create a shortcut

- Magento layout is a flexible grid layout
  The web pages are arranged by a grid rule to standardize the layout, it brings a lot of convenience to the web front-end development.

![Grid layouts in different resolution](image)

Figure 2.3 Grid layouts in different resolution

- Meta tag definitions
  1. Use viewport meta tag to control the layout of the mobile browser
2. Full Screen when open through shortcut

```html
<meta name="viewport" content="width=device-width, initial-scale=1,maximum-scale=1" />
```

3. Hide Mobile Status Bar

```html
<meta name="apple-mobile-web-app-capable" content="yes" />
<meta name="apple-mobile-web-app-status-bar-style" content="blank" />
```

• **CSS Media Queries**

In CSS, it can be set up different types of media conditions, different media types of expressions used to check whether the media subject matches certain conditions, if the media type meets the appropriate conditions, it will call the corresponding stylesheet.

```css
/* for 240 px width screen */
@media only screen and (max-device-width:240px){
  selector{ ... }
}

/* for 320px width screen */
@media only screen and (min-device-width:241px) and (max-device-width:320px){
  selector{ ... }
}

/* for 480 px width screen */
@media only screen and (min-device-width:321px)and (max-device-width:480px){
  selector{ ... }
}
```

Figure 2.4 Media query code with different resolution

2.2.2 **Powerful admin panel**

• Login Page

During Magento installation, admin credential will be asked to fill in.
Figure 2.5 Admin panel log in page

- **Dashboard Page**
  Dashboard page gives brief information of most viewed products, new customers, new orders, sales amount and etc.

Figure 2.6 Admin panel dashboard page

- **Sales**
  Sales section is an extremely important part of Magento, particularly order management, invoice management, Tax configuration.
1. Order

Order section has a list of orders; those orders could be in different stats. For example processing, canceled, it is also possible to create custom statuses for orders. Admin can create orders for specific customers.
2. Invoice

Invoice section has a list of invoice, those invoices can be printed, added comment. The administrator can choose to send the invoice to the customer by email.

3. Tax configuration
Tax section can add product, customer tax classes. It can add different tax rates for different countries and zones.

![Admin panel tax configuration page](image1)

Figure 2.10 Admin panel tax configuration page

- Catalog
  Catalog section is about product related function. It includes product management, order management, attribute management.

![Admin panel catalog management page](image2)

Figure 2.11 Admin panel catalog management page

1. Product
   Product section has a list of products; those products could be enabled or disabled. The product can be added, updated, deleted, duplicated.
2. Category

Category section has a list of categories, it can add categories, assign products to category.
2.2.3 Web service

Magento API provides three different ways to use web service

- XML-RPC
  XML-RPC can only use a limited number of data types and some simple data structures.
  1. Create API to use with XML-RPC
2. Define current XML-RPC role scope, the scope here means resource access privilege, different role scope could have different a resource access privilege, for example, define a salesman role which can only access product resource and customer resource.

3. Demo code of how to invoke XML-RPC in Magento
To use XML-RPC, load the following URL into your XML-RPC client:

http://magentohost/api/xmlrpc/

where magentohost is the domain for your Magento host.

The following PHP example shows how to make XML-RPC calls:

```php
$client = new Zend_JsonRpc_Client('http://magentohost/api/xmlrpc/');

// If somestuff requires api authentification,
// we should get session token
$session = $client->call('login', array('apiUser', 'apiKey'));

$client->call('call', array($session, 'somestuff.method', array('arg1', 'arg2', 'arg3')));
$client->call('call', array($session, 'somestuff.method', 'arg1'));
$client->call('call', array($session, 'somestuff.method'));
$client->multiCall(array($session,
    array('somestuff.method', 'arg1'),
    array('somestuff.method', array('arg1', 'arg2'))),
    array('somestuff.method')));
```

// If you don't need the session anymore
$client->call('endSession', array($session));

The XML-RPC only supports the version 1 of the Magento API.

Figure 2.16 How to invoke XML-RPC in Magento

- SOAP V1 and V2
  1. Create API and define role scope for the use of soap is the same as XML-RPC scope
  2. Difference between SOAP V1 and V2
    SOAP v1 do not have all methods included in WSDL, WSDL file only include some simple methods like login. SOAP v2 have all the soap methods included in WSDL, which is much easier for third-party software to understand what method they can invoke.
  3. Use SOAP V1 and V2 to delete a product
Figure 2.17 How to invoke SOAP to delete a product in Magento

- Restful API
  1. Create REST API token

Figure 2.18 How to create REST API token in Magento

2. REST API request response process
3. Use REST API to get a customer

Example:

GET http://magentohost/api/rest/customers/2

Response Body:

```xml
<?xml version="1.0"?>
<Magento_api>
  <entity_id>2</entity_id>
  <website_id>2</website_id>
  <email>last@example.com</email>
  <group_id>1</group_id>
  <created_at>2012-03-22 14:15:54</created_at>
  <disable_auto_group_change></disable_auto_group_change>
  <created_in>Default Store View</created_in>
  <firstname>John</firstname>
  <lastname>Doe</lastname>
  <last_logged_in>2012-03-22 14:15:54</last_logged_in>
</Magento_api>
```

Figure 2.19 Rest API request response process

Figure 2.20 Use REST API to get a customer in Magento
2.2.4 Multi websites & Multi stores

With Magento, it is able to manage multiple websites and stores by one Magento installation. This allows the owner to manage different web shops with multiple domains.

- Website

A site can contain one or more stores, these stores are sharing the same customer information, order information and shopping cart information.

- Store

The shops can be established by a variety of different ways but need to be reminded that if they belong to the same site, then they will share certain information.

- Store Views

Store interface is mainly applied in the use of different languages.

For example, in one case, the customer needs a B2C and B2B sites, each site has different shops, B2C there are two shops are selling cosmetics and electronic products, each store requires a different language (Finnish, English, Chinese).
2.2.5 Cache and index management

E-commerce stores a large number of categories, products, orders, and customers. High performance is a very important requirement. Cache design and index design are essential functions.

- Cache management
  1. Cache management page in admin panel

![Cache management page in admin panel](image-url)
2. Magento cache is based on Zend framework, the cache file in Magento will have a default prefix ‘mage’.

Figure 2.23 Cache management folder

- Index management
  1. Index management page in admin panel

Figure 2.24 Index management page in admin panel

2. Sometimes too many products in the queue are waiting to index, index in admin panel may have no response. Magento offers another way which is index by shell scripts.

Figure 2.25 Use command to index product, category, and stock.
2.2.6 Enterprise edition VS Community edition

For small and medium size enterprise, basic functions of Magento edition are enough to create and manage a user-friendly online shop. However, if a company has a larger scale, and need to dramatically increase sales, Magento Enterprise Edition would be a good option.

Below is the list about the difference between enterprise and community edition.

<table>
<thead>
<tr>
<th>Function</th>
<th>Enterprise</th>
<th>Community</th>
</tr>
</thead>
<tbody>
<tr>
<td>Senior Administrator role and restrictions of</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>operation of licensing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Record administrator behavior</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>Gift Cards (real and virtual)</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>The customer credit limit</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>High-performance and full-page caching</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>Reward Points System</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>BridgeConnect module, make integration with other ERP</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td></td>
<td>systems easier</td>
<td></td>
</tr>
<tr>
<td>Rich Merchandizing Suite (RMS)</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>Strengthen data encryption</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>Improved content management system</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>Content staging and merging. Support for on-demand</td>
<td>Included</td>
<td>Not Included</td>
</tr>
<tr>
<td>and scheduled merger and content</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimized index without affecting the customer's shopping experience</td>
<td>Included</td>
<td>Not Included</td>
</tr>
</tbody>
</table>

### 2.2.7 Import and Export

Magento has import/export function. Include customer and product import/export

- Product and customer import

![Admin panel product and customer import page](image)

- Product and customer export

![Admin panel product and customer export page](image)
3 REQUIREMENT ANALYSIS

3.1 Requirement

The following are the requirements of the web shop application:

1. Theme
2. Product distribution service
3. Ordering and reservation service
4. Inventory service
5. Customer services
6. Marketing services
7. Payment services

3.2 Analysis

- Theme
  Responsive design is a popular design approach, float elements will be widely used in CSS. Considering the support for tablet and mobile phones, we need use CSS media library, websites will have good layouts in different devices. Website need compatible with IE 10+, Chrome, Firefox, and Safari.

- Product distribution service
  Website need includes two shipping methods. One is “pick up at store”, the other one is Itella smart post. With Itella smartpost, it can search pick up points by postcode or city name. The shipping fee is configurable.

- Ordering and reservation service
  Products in the shopping cart can be checkout immediately or put in a wish list for checkout later.
• Inventory service
Product inventory can be added, updated. If a product out of stock, it will hide from the website.

• Customer services
Customers can register, login. It is also possible buy product without registration.

• Marketing services
  1. Customers can read receive the latest news about new products, discount products.
  2. Add social media links to promote the website, like Facebook, Twitter.
  3. Promotion rules like free shipping if order over 40€.

• Payment services
Website needs support PayPal, visa card, and master card. Support Nordic region bank payments, such as Nordea bank, Danske bank.

• Store configuration
  1. Multi-store, one in Helsinki, the other in Espoo.
  3. Possible to add CMS pages like contact us, company info, slider, etc.

4 IMPLEMENTATION

4.1 Magento installation

• Modify etc/hosts, Add “127.0.0.1 magento19.local.fi”
• Modify etc/apache2/extra/httpd-vhost.conf
Figure 4.1 Virtual host configuration in local environment

- Restart apache server
  Server need to restart to make virtual host configuration works.

- Create database, add sample data

Figure 4.2 Create database, add sample data

- Localization
Figure 4.3 Choose website language, language also can be change later

- Set folder permission

  Consider that there will be product image and other images upload to media folder. Media folder needs writable permission.

Figure 4.4 Folder permission warning page

- Configuration
Figure 4.5 Database configuration page, create admin account page

- Folder structure
Code folder includes three types of code, Magento core, community module code, and customized module code. Design folder includes all templates, divided by admin panel template and frontend template. Etc folder includes database connection configuration and brief information of all modules.

Locale includes all translation packages. Skin folder includes admin panel CSS, images, and also frontend CSS, images.

Figure 4.6 Project code folder structure
4.2 Frontend design

Make a Magento theme need use following available tools:

1. Templates (HTML content)
2. Layout (control the main content and sidebar position)
3. Block (make widget can be reused in different places)
4. Skin (images, CSS and block-specific JavaScript)

In this application, template, layout, and the block will use the Magento default theme, but CSS has been rewritten to make it responsive.

Figure 4.7 Home page

Figure 4.8 Product page
Figure 4.9 Shopping cart page

Figure 4.10 Checkout page
4.3 Configuration

Figure 4.11 Multi store configuration

<table>
<thead>
<tr>
<th>Vehicle Store</th>
<th>Stores Name</th>
<th>Store View Name</th>
<th>Website Store</th>
<th>Store Name</th>
<th>Store View Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anchor vehicles</td>
<td>Countries (Root Category</td>
<td>Default Category)</td>
<td>China</td>
<td>(code: cn)</td>
<td>China</td>
</tr>
<tr>
<td></td>
<td>(code: cn)</td>
<td></td>
<td></td>
<td>(code: cn)</td>
<td>China</td>
</tr>
<tr>
<td></td>
<td>(code: cn)</td>
<td></td>
<td></td>
<td>(code: cn)</td>
<td>China</td>
</tr>
<tr>
<td>Max.Vehicle</td>
<td>Manitoba (Root Category</td>
<td>Default Category)</td>
<td>English</td>
<td>(code: en)</td>
<td>English</td>
</tr>
<tr>
<td></td>
<td>(code: en)</td>
<td></td>
<td></td>
<td>(code: en)</td>
<td>English</td>
</tr>
<tr>
<td></td>
<td>(code: en)</td>
<td></td>
<td></td>
<td>(code: en)</td>
<td>English</td>
</tr>
</tbody>
</table>

Figure 4.12 Email configuration

<table>
<thead>
<tr>
<th>Store Email Addresses</th>
<th>General Contact</th>
<th>Sales Representative</th>
<th>Customer Support</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sender Name</td>
<td>Fusion</td>
<td>Fusion Support</td>
</tr>
<tr>
<td></td>
<td>Sender Email</td>
<td><a href="mailto:Fusion@gmail.com">Fusion@gmail.com</a></td>
<td><a href="mailto:fusion@gmail.com">fusion@gmail.com</a></td>
</tr>
<tr>
<td></td>
<td>[STORE VIEW]</td>
<td>[STORE VIEW]</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[STORE VIEW]</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[STORE VIEW]</td>
<td>[STORE VIEW]</td>
</tr>
</tbody>
</table>
### Currency Setup

<table>
<thead>
<tr>
<th>Currency Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Currency</td>
</tr>
</tbody>
</table>

Figure 4.13 Currency configuration

### Catalog Price Rules

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Get up to 50% off on Candle holders</td>
</tr>
<tr>
<td>5</td>
<td>25% off on Cardigan</td>
</tr>
<tr>
<td>9</td>
<td>45% off on Boyfriend Jean</td>
</tr>
<tr>
<td>6</td>
<td>Get Briefcase for $170 less</td>
</tr>
<tr>
<td>8</td>
<td>Get Travel wallet for $150</td>
</tr>
<tr>
<td>10</td>
<td>Get Vase at $80 less than the original price</td>
</tr>
</tbody>
</table>

Figure 4.14 Promotion rule configuration
Figure 4.15 Customer group configuration

<table>
<thead>
<tr>
<th>ID</th>
<th>Group Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>General</td>
</tr>
<tr>
<td>0</td>
<td>NOT LOGGED IN</td>
</tr>
<tr>
<td>5</td>
<td>Private Sales Member</td>
</tr>
<tr>
<td>4</td>
<td>VIP Member</td>
</tr>
<tr>
<td>2</td>
<td>Wholesale</td>
</tr>
</tbody>
</table>

Figure 4.16 Shipment methods configuration
### Payment Methods

Not sure what PayPal payment method to use? Click [here](#) to learn more.

- PointPayment
- Merchant Location
- Paytrail / S-Pankki
- Paytrail / Nordea Tillirto
- Paytrail / Kiarna Laskulla
- Paytrail / Netposti
- Paytrail / Paypal
- Paytrail / Aktia
- Paytrail / POP Pankki
- Paytrail / Säästöpankki
- Paytrail / Luottokunta
- Paytrail / Visa
- Paytrail / JCB
- Paytrail / Collector
- Paytrail / Jousto
- Paytrail / Dinara Club
- Paytrail / American Express
- Paytrail / Säästöpankki, Aktia, Nooa
- Paytrail / MasterCard
- Paytrail / Maestro
- Paytrail / Kiarna Osamaksuila
- Paytrail / Alandsbanken
- Paytrail

---

### Manage Pages

<table>
<thead>
<tr>
<th>Title</th>
<th>URL Key</th>
<th>Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Us</td>
<td>about-magento-demo-store</td>
<td>1 column</td>
</tr>
<tr>
<td>Customer Service</td>
<td>customer-service</td>
<td>3 columns</td>
</tr>
<tr>
<td>Enable Cookies</td>
<td>enable-cookies</td>
<td>1 column</td>
</tr>
<tr>
<td>Home page</td>
<td>home</td>
<td>2 columns with right bar</td>
</tr>
<tr>
<td>Home page</td>
<td>home</td>
<td>1 column</td>
</tr>
<tr>
<td>404 Not Found</td>
<td>no-route</td>
<td>2 columns with right bar</td>
</tr>
<tr>
<td>Privacy Policy</td>
<td>privacy-policy-cookie-restriction-mode</td>
<td>1 column</td>
</tr>
</tbody>
</table>

---

Figure 4.17 Payment methods configuration

Figure 4.18 CMS page configuration
4.4 Module Development

The module is the core of the Magento. Any action on the site, whether it is in the foreground or background operation is implemented by the module. Modules can be thought as a container that can contain these items below: setting, database schema, rendering object, utility helpers, and data model or action controller. A module can contain all of these six can contain only a few of them, or even only one. All modules can be disabled or enabled. Each module can also create an XML file in the etc folder to save their configuration information.

4.4.1 Itella smartpost

The idea of this module is let customer can use itella post service in shopping process.

- It should have below features
  - Display pickup point base on postcode or shipping address
  - Search pickup point base on postcode or shipping address
  - Shipping price is configurable, price can be fixed or calculated base on product weight
  - Set free shipping price
- Create folder structure.

![SmartPost folder structure](image)

Figure 4.19 Itella smart post module folder structure
• In model, add itella web service, and method to fetch data from web service API.

```php
protected $_code = 'itellaSmartPost';
protected $_productCode = '2106';
protected $_shippingLabelCode = 'itellaSmartPost';
CONST REST_URI = 'https://ohjelmat.posti.fi';
CONST PICKUP_URI = '/pup/v1/pickuppoints';
```

Figure 4.20 Itella smart post web service url

• Request parcel pick up point by postcode or city name

```php
protected function _getPointsByRest($requestData) {
  // Make the REST request
  $client = new Zend_Rest_Client( self::REST_URI );
  $result = $client->restGet( self::PICKUP_URI, $requestData ); // Zend_Http_Response object
  if ($result->isSuccessful()) {
    if (count($result->getBody()) > 0) {
      $points = $result->getBody();
    } else {
      // In case REST client does not return any points we can use CURL.
      // Page loading is slow with this method. +/-
      $points = $this->getPointsByUrl($requestData);
    }
    $points = Zend_Serialize::decode( $points );
  } else {
    // This->getPointsByUrl($requestData);
    // Mage::throwException(Mage::helper('smartpost')->__('Can not receive locations via REST, fallback to CURL'));
    Mage::throwException(Mage::helper('smartpost')->__('Can not receive locations via REST'));
  }
  //...
  if (isset($result->PickUpPointExternalResponse->Error->ErrorCode)) {
    Mage::throwException($result->FetchLocationSearchResponse->Error->ErrorMessage);
  }
  if (count($points) == 0) {
    return null;
  }
  return $points;
}
```

Figure 4.21 Part code of get itella smart post pick up points

• In system.xml we add code to display fields in admin panel
- Calculate price base on product weight.

```
<price translate="label comment">
  <label>Price</label>
  <comment>If weight steps is used, use multiple semicolon (;) separated values that matches with weight step</comment>
  <frontend_type;text</frontend_type>
  <sort_order>40</sort_order>
  <show_in_default>1</show_in_default>
  <show_in_website>1</show_in_website>
  <show_in_store>1</show_in_store>
</price>
```

Figure 4.22 display calculate price base on weight in admin panel

- Set free shipping price

```
<free_shipping_enable translate="label">
  <label>Free shipping with minimum order amount</label>
  <frontend_type;select</frontend_type>
  <source_model;adminhtml/system_config_source_enabledisable</source_model>
  <sort_order;60</sort_order>
  <show_in_default>1</show_in_default>
  <show_in_website>1</show_in_website>
  <show_in_store>1</show_in_store>
</free_shipping_enable>

<free_shipping_subtotal translate="label">
  <label>Minimum order amount for free shipping</label>
  <frontend_type;text</frontend_type>
  <sort_order;70</sort_order>
  <show_in_default>1</show_in_default>
  <show_in_website>1</show_in_website>
  <show_in_store>1</show_in_store>
</free_shipping_subtotal>
```

Figure 4.23 display free shipping price field in admin panel

- After config in system.xml, In Magento admin panel will get automatically generated layout
4.4.2 Paytrail

The idea of this module is let customer can use various banking services in the Nordic area. Paytrail (Suomen Verkkomaksut) web service contains payments to all Finnish banks, major credit and debit card payments. API document can be found at [http://docs.paytrail.com/en/](http://docs.paytrail.com/en/).
• In the model, add paytrail web service, and method to fetch data from web service API.
• In the controller, add redirect action, because transaction will not finish in Magento, it will first redirect to paytrail, use paytrail security connection to finish the transaction.
- In system.xml we add code to display fields in admin panel, use paytrail service need apply account first, fill merchant id in Magento admin panel to active paytrail service.

```php
<?php

public function redirectAction()
{
    try
    {
        $svmPayment = Mage::getSingleton('sva/svmPayment');
        if(Mage::getSingleton('checkout/session')->getLastRealOrderId()) {
            $order = Mage::getModel('sales/order');
            $order->loadByIncrementId(Mage::getSingleton('checkout/session')->getLastRealOrderId());
            if ($svmPayment->getEmailStatus() == 0) {
                $order->sendNewOrderEmail()->addStatusHistoryComment(
                    Mage::helper('svm')->__('Order confirmation sent.'), " ".
                    Mage::helper('svm')->__('Client redirected to the payment service')
                )
                ->setIsCustomerNotified(true)
                ->save();
            } else {
                $order->addStatusHistoryComment(Mage::helper('svm')->__('Client redirected to the payment service'))
                    ->save();
            }
            $quote = Mage::getModel('sales/quote')->load($order->getQuoteId());
            $quote->setIsActive(true)->save();
            Mage::getSingleton('checkout/session')->setQuoteId($quote);
            Mage::getSingleton('checkout/cart')->setQuoteId($quote)->save();
        }
    }
}
```
• After configuring in system.xml, the layout will automatically generate in Magento admin panel.

<table>
<thead>
<tr>
<th>Paytrail</th>
<th>Value</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Yes</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>Title</td>
<td>Verkkopankki- tai korttimaakku (Paytrail)</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>Merchant ID</td>
<td>13466</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>Secure Key</td>
<td>******</td>
<td>[WEBSITE]</td>
</tr>
<tr>
<td>Interface language</td>
<td>Finnish</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>New order email is sent</td>
<td>When payment is successfully confirmed</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>New invoice email is sent</td>
<td>When payment is successfully confirmed</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>Show Checkout Image</td>
<td>Yes</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>Checkout Image Columns</td>
<td>12</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>Horizontal Display of Checkout Image</td>
<td>Yes</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>New Order Status</td>
<td>Pending</td>
<td>[WEBSITE]</td>
</tr>
<tr>
<td>Send order details to Paytrail</td>
<td>No</td>
<td>[STORE VIEW]</td>
</tr>
<tr>
<td>Automatic pending order cancellation</td>
<td>No</td>
<td>[GLOBAL]</td>
</tr>
<tr>
<td>Sort Order</td>
<td></td>
<td>[WEBSITE]</td>
</tr>
<tr>
<td>Debug mode</td>
<td>Yes</td>
<td>[STORE VIEW]</td>
</tr>
</tbody>
</table>

Figure 4.29 Paytrail fields in admin panel

Figure 4.30 Paytrail in website frontend
From module development, it is obvious to know that Magento module is based on MVC structure. Magento also provides some predefined component. Magento xml defined the structure of modules.

5 TESTING

5.1 Browser compatibility test


5.2 Magento shopping process

![Figure 5.1 Registration success](image1)

![Figure 5.2 Add products to shopping cart success](image2)
Figure 5.3 Checkout page include account info, shipping method, payment method and order info.

Figure 5.4 Redirect to payment page success
Figure 5.5 Use paytrail sandbox account credential to test payment

Your order has been received.
Thank you for your purchase!
Your order # is: 145000011.
You will receive an order confirmation email with details of your order and a link to track its progress.
Click here to print a copy of your order confirmation.

CONTINUE SHOPPING

Figure 5.6 Get order confirmation

MY DASHBOARD
Hello, yu lin!
From your My Account Dashboard you have the ability to view a snapshot of your recent account activity and update your account information. Select a link below to view or edit information.

RECENT ORDERS  VIEW ALL

<table>
<thead>
<tr>
<th>ORDER #</th>
<th>DATE</th>
<th>SHIP TO</th>
<th>ORDER TOTAL</th>
<th>STATUS</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>145000011</td>
<td>3/29/2015</td>
<td>yu lin</td>
<td>€195.00</td>
<td>Processing</td>
<td>VIEW ORDER</td>
</tr>
<tr>
<td>145000010</td>
<td>3/29/2015</td>
<td>yu lin</td>
<td>€195.00</td>
<td>Canceled</td>
<td>VIEW ORDER</td>
</tr>
</tbody>
</table>

Figure 5.7 View order in my account dashboard
6 CONCLUSIONS

This project is a good practice on Magento, it is able to fully meet customer demand for e-commerce, customers can register, log in, view products, add to cart, checkout, review orders and so on.

By responsive design, it offers a good user experience in different devices.

This application not only offers international payment methods like PayPal, Visa, Master Card, and shipment methods like DHL, UPS. Through the development of modules, delivery methods like Itella smart post, payment methods like Nordea bank, Danske bank have been integrated with this application. Those modules achieve localization needs.

The Magento structure is based on Zend framework. Its MVC structure has a lower degree of coupling between modules and makes module development is quite convenient.
7 WHAT CAN BE IMPROVED

• Performance issues

Magento server requirements are relatively high, when product categories and product number goes higher, might need to use redis or memcache applications to help improve performance.

• Learning cost of Magento is high

Customers reflect that Magento admin panel is powerful but complex to use, takes a long time to learn how to use it.

• Subsequent module updates and Magento version release update

Most of the current e-commerce platform update fast, Magento is no exception, Magento update twice a year approximately, the corresponding module also needs to be updated.

• Promotional rule

Magento offers some basic promotional rules, such as ‘by 100 return 10’, ‘buy three get one free’, ‘coupon codes’. But customers may have more complex marketing strategies, which need to do more module development.
8 REFERENCES

/1/ April 2014 ecommerce Survey

<URL: http://tomrobertshaw.net/2014/04/april-2014-ecommerce-survey/>

/2/ All major internet retailers in Europe

<URL: http://www.retail-index.com/ERetail.aspx/>

/3/ Magento REST API

<URL: http://www.magentocommerce.com/api/rest/authentication/oauth_authentication.html/>