Developing E-Commerce website using Zend Framework and MySQL database

Nguyen, Duc Huu Tri

2015 Leppävaara

Laurea University of Applied Sciences
Leppävaara

Developing E-Commerce website using Zend Framework and MySQL database

Nguyen Duc Huu Tri
Business Information Technology in
Bachelor’s Thesis

Laurea University of Applied Sciences
Leppävaara
Business Information Technology

Abstract

Nguyen Duc Huu Tri

Developing an E-commerce website using Zend Framework and MySQL database

Year	2015	Pages	37

This thesis concentrates on developing an E-commerce website using Zend framework and MySQL database. Basically, a web development consists of 5 stages: graphic user interface(GUI) design, database design, back-end development, front-end development and search engine optimization (SEO) but only graphic user interface design, database design and back-end developing using Zend framework are focused in this thesis. The outcome of this thesis is the E-commerce website, www.mocaymart.com.

The main methodology in this thesis is qualitative research, which mainly conducts various interviews for requirements collection and analysis. Furthermore, constructive research that is designed to solve problems in the real world is also ultilized.

As a conclusion, the web owner satisfies with www.mocaymart.com and this website is evaluated as the key factor in their start-up plan.

Key words www.mocaymart.com, E-commerce website, Zend framework, Coconut products

Table of Contents

1 Introduction

The invention of the Internet accelerated information exchange and it enables various resources accessed globally. It also makes E-commerce industry more attractive by offering such useful function as shopping cart, online payment etc. that help customers' shopping more convenient. E-commerce can help us reach new potential with our business, making our products and services available to a wider audience (Peacock 2008, 7). Presently, much E-commerce websites like www.amazon.com, www.ebay.com, or www.alibaba.com etc. gain the prominent success and they inspire more and more people start up business in this field.

1.1 Project background

Building an E-commerce website www.mocaymart.com using Zend framework and MySQL database is a part of the business start-up project planed by two Vietnamese. Their business is about online trading coconut products such as coconut jelly, coconut jams, coconut oil, coconut handmade etc. The idea of this start-up project inspired by the expanding E-commerce market in Vietnam, a dynamic economy in the Asia supported by young labour force and huge population size.

Vietnam's E-commerce and Information Technology Agency (VECITA) estimated that the amount of B2C E-commerce sales will rise to more than US$4 billion by 2015 and the current sales are estimated to be around US$2.2 billion, with an average spending of US$120 per capita (Edward & Fabio 2014). VECITA also showed that the Vietnam's online population reached a total of 32.6 million users in 2013 and it represents 35.6% of the total population and according to recent surveys, 92% of Internet users go online every day (Edward & Fabio 2014). Figure 1 and figure 2 provide the broader view toward Vietnam E-commerce market.

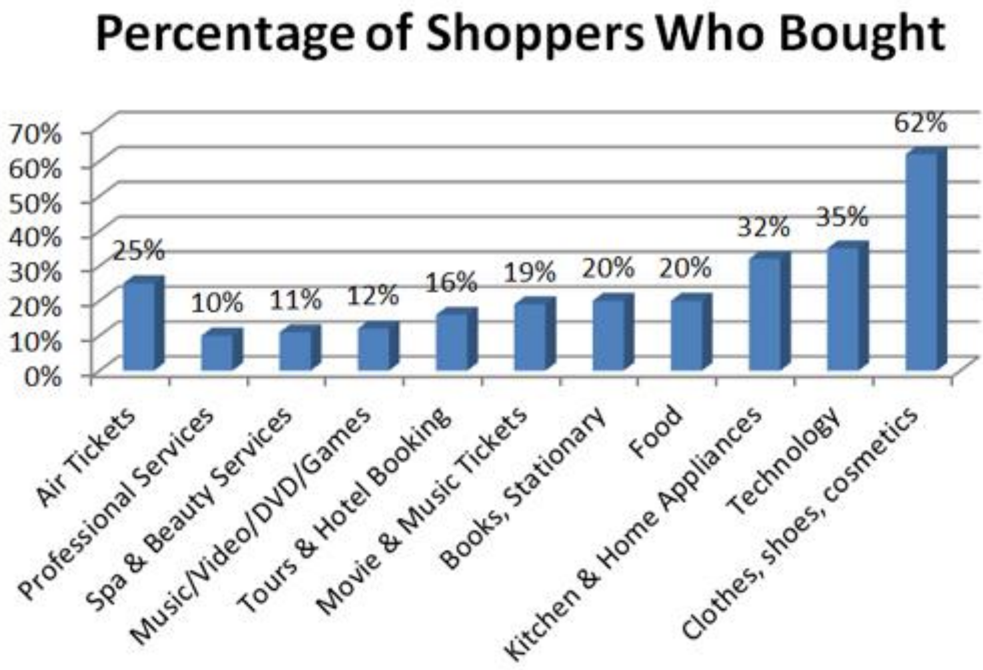


Figure 1: Online shopping product categories in Vietnam 2013

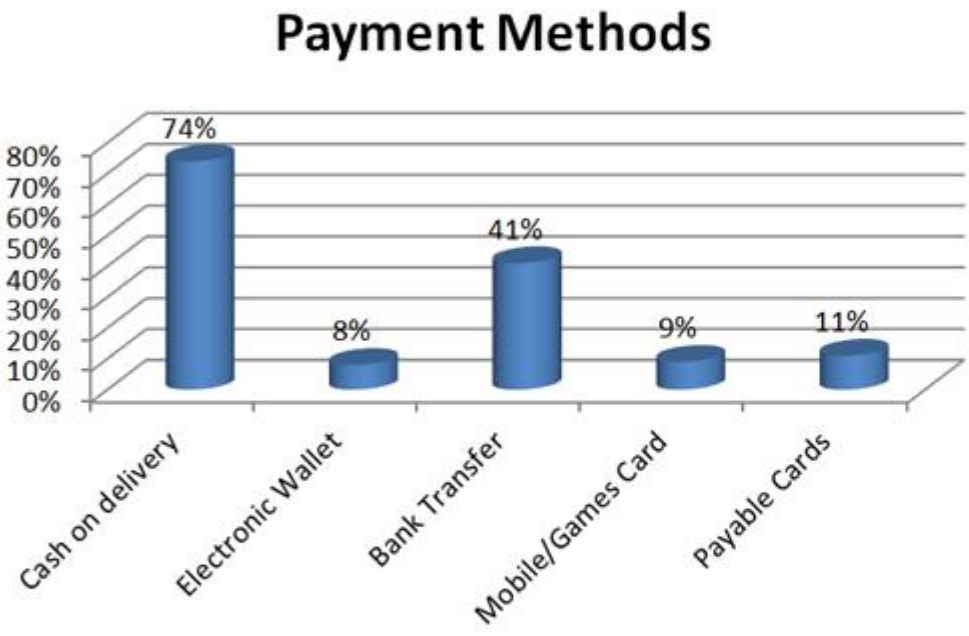


Figure 2: Payment methods of Vietnamese online shoppers in 2013

Above diagrams show that food consists of a considerable ratio in product categories chosen by Vietnamese shopper and the two most popular payment methods are “Cash on delivery” and “Bank Transfer” that are also the main way of payment on www.mocaymart.com website. The website is almost finished and it is published on May 2014, the default language is presently Vietnamese and English and Japanese language will be available soon. It is developed base on the requirement of the customers who is www.mocaymart.com website owner. I am in charge of building the website in all stages: design, database design, front-end development, back-end development, and search engine optimization (SEO). Due to the fact that the vast majority of customers are Vietnamese, the website is hosted on the server located in Vietnam for maximum performance of web surfing.

1.2 Project objective

The main objective of this project is building an E-commerce website which contains user pages and functional admin pages for website contents administration. The website should satisfy the shopper's demand by offering the well-designed graphic user interface and good functions such as shopping cart, products searching, contact forms, ordered product status monitoring etc. This project also gave me a great opportunity to grasp necessary skills of building a website from the scratch and those experiences enable me more brilliant achievement in the future.

1.3 Thesis research method

The main research method in this thesis is qualitative research method that concentrates on conducting the interviews toward web owner as shown in appendix. The website's features such as functions, graphic user interface, and website organization are generated according to requirements that are acquired through various meetings between web developer and web owner.

Furthermore, this thesis also uses constructive research approach that intended to solved the problem in the real world, by that means, to make a contribution to the theory of the discipline in which it is applied (What is a constructive research approach, n.d)

2 Project theoretical background

Basically, developing a website contains 5 phases as following:

- **Graphic user interface design**: To design the web interface, icons, banners etc.
- **Front-end development**: To code the elements that the customer can see and interact directly. Those elements are interpreted and rendered by web browser. The technologies related to front-end developing can be listed as Hyper Text Markup Language (HTML), Cascading Style Sheet (CSS), JavaScript, JQuery, Asynchronous JavaScript and XML (Ajax).

- **Database design**: To plan the data model of a database. In detail, the data tables are design based on the website structure and functionalities.
- **Back-end development**: To code the behind-the-scenes elements, which communicate with the web server and the database server. Hypertext Preprocessor (PHP), MySQL database, Sequence Query Language (SQL) etc. are the technologies used in back-end development.
- **Search Engine Optimization (SEO)**: To make the website friendlier to search engines like Google, Bing, and Yahoo by optimizing HTML tags and conducting link building, blacklists check etc.

However, this thesis concentrates on only 3 phases: the graphical user interface design, database design and back-end development. The theories related to those technologies will be introduced as following items.

2.1 Zend framework

Zend is a web application framework developed by Zend Technologies and it is evaluated as one of the most popular PHP framework. The Model-View-Controller (MVC) architecture of Zend framework makes the website easy to develop, maintain and extend by separating the "Model", "View" and "Controller" layer

2.1.1 Web application framework introduction

A web application framework is a bunch of source code organized into a certain architecture that can be used for rapid development web applications (Porebski, Przystalski & Nowak 2011, p2). Web application framework allows developers reduce development time by providing the pre-coded libraries that support database access (DA), session management, and security management etc. However, frameworks are not the cure for all programming problems (Porebski, Przystalski & Nowak 2011, p3) and it is only suitable for dynamic content like social networking, E-commerce website, news portal etc. because it may not meet the requirement of such large scale websites as www.google.com or www.amazon.com.

Web framework can be developed by different programming languages such as Java, PHP, Python, Ruby etc. Figure 3 reveals the usage of web framework by different programming language on August 2014.

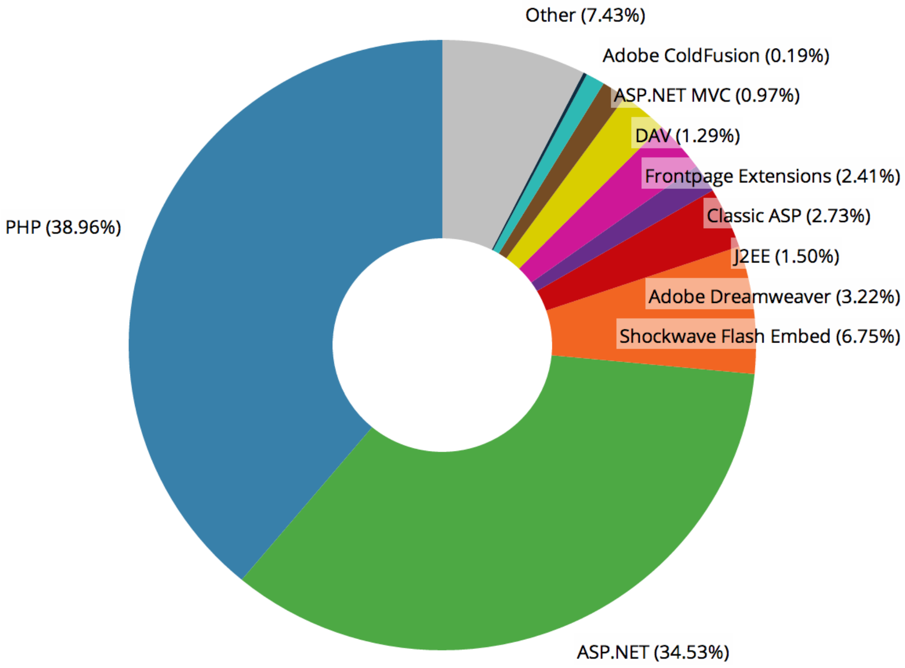


Figure 3: Web application framework usage by programming language on August 2014

According to this chart, PHP and ASP.NET are dominant programming language used in web application framework with the percentage nearly 75%. There are some reasons that make PHP the most popular programming language in web developing.

- PHP is an open source and free to use, easy to learn especially for developers who have background in C and Java. It is able to work on UNIX, Window and Mac environment.
- PHP offer the huge community where the challenge in website developing can be shared and solved. Furthermore, due to its' popularity the PHP developers receive the excellent support PHP usage and configuration from PHP website, support groups, forums etc.
- PHP is fast to develop thus it is the good choice for small and medium size website.

2.1.2 MVC architecture in Zend framework

MVC in Zend framework stands for Model-View-Controller that represents the layers in the design of modern web application. Basically, the codes in web application can be divided into 3 categories: presentation, business logic and data access and it is also the idea of MVC architecture. Further explanations for each layer in MVC are demonstrated in figure 4.

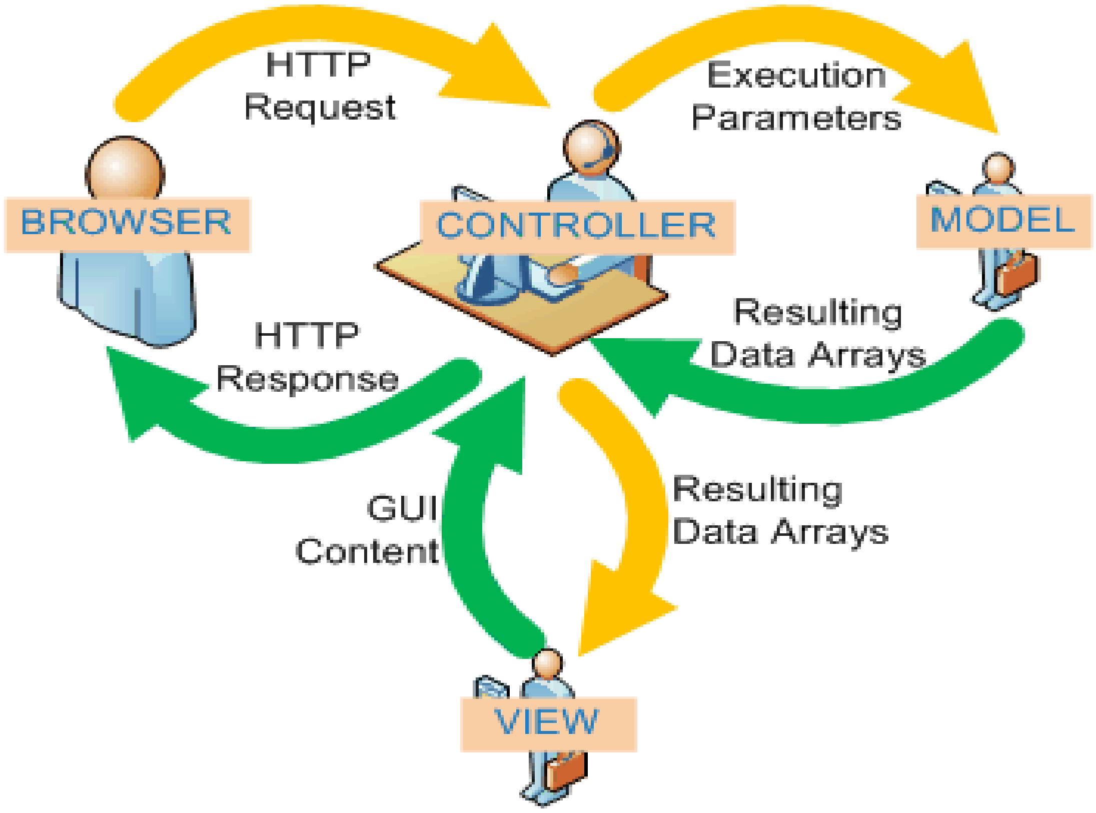


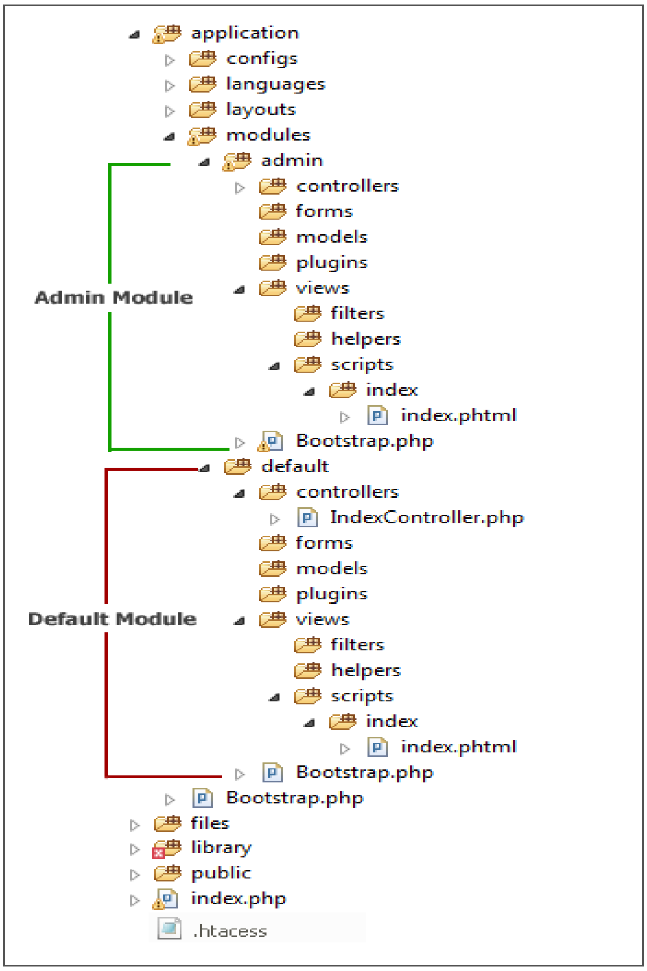
Figure 4: MVC pattern in Web application

- **Controller**: The mission of this layer is controlling and delegating the HTTP requests. For example, when controller receives the request of retrieve customer information, it interprets the request into action based on predefined business logic that executes the database retrieve from Model layer. Consequently, controller passes the data to View layer for presentation.
- **Model**: This layer defines data access routines and some business logic in order to make the connections to the database for data retrieving.
- **View**: The view layer is in charge of creating the user interface that is visible for the web user. For instance, view layer render the data received from the controller in some format defined by Hyper Text Markup Language (HMTL) and Cascading Style Sheet (CSS).

According to the above diagram, view, and model should not communicate with each other and the communications have to be conducted via controller layer. The separating of model, view and controller in web application design makes website coding and maintenance simpler. For example, in the case of web developer want to use the additional database for the new module, he or she just codes a model layer that supports the new database.

2.1.3 Zend framework advantages

As an open-source framework developed by PHP5, Zend framework is evaluated as one of the most popular PHP framework because of its' high ability in building interactive web application. The considerable advantage of Zend framework in comparison with other PHP framework such as CakePHP, Codeigniter, Yii, Symfony is the highly modular Model-View-Controller design that offer many loosely coupled components. The typical folder structure of Zend framework reveals more about the benefit of Zend framework as figure 5.



```
application
  configs
  languages
  layouts
  modules
    admin
      controllers
      forms
      models
      plugins
      views
        filters
        helpers
        scripts
          index
            index.phtml
      Bootstrap.php
    default
      controllers
        IndexController.php
      forms
      models
      plugins
      views
        filters
        helpers
        scripts
          index
            index.phtml
      Bootstrap.php
  Bootstrap.php
files
library
public
index.php
.htacess
```

Figure 5: Web application folder structure in Zend framework

The web application in Zend framework is divided into modules that are responsible for different functions and each module have 3 layers: controller, model, and view. For example, Admin module is designed for the web admin to monitor the website content while default module contains the source codes of the website visible to users. This clear structure allows developer build or maintains a website effortlessly.

2.2 MySQL database

According to website www.w3school.com, MySQL is a database system used on the web and it runs on a database server. A database is the application for storing a collection of data. Data on the database can be created, accessed, managed, searched and replicated by different Application Programming Interfaces (APIs). There are four types of database management system (DBMS) named Hierarchical DBMS, Network DBMS, Relational DBMS (RDBMS) and Object-oriented DBMS, but RDBMS is the most popular in web development because following advantages.

- **Data integrity**: The cross-reference feature in RDBMS ensures the data integrity and duplicated data elimination. Furthermore, it also allows related information be retrieved easily.
- **Efficient and speed**: The data in RDBMS can be updated rapidly and everyone can access the data in RDBMS simultaneously.
- **Security**: RDBMS perform the strict authentication and authorization policy so the access from unauthorized users is eliminated and the users can only see the data related to his or her position.

MySQL is an RDBMS that many businesses choose for the database of their website. It is free to use because of an open-source license. MySQL is a PHP-friendly RDBMS and the combination of those could build the powerful website without considerable effort.

3 Project implementation

Basically, the website is developed according to customer requirements and it has undergone 5 stages: graphic user interface design, database design, back-end developing, font-end developing and search engine optimization (SEO). Nevertheless, only graphic user interfaces design, database design and back-end developing using Zend framework are gone through in this thesis.

3.1 Graphic user interface and database design

The objective of this step is creating the easy-to-use web interface. Additionally, the website organization should meet the customer’s requirement.

3.1.1 Customer requirements

The website is developed for the start-up business in E-commerce field so the customer require some basic functions for it. The details are described as following.

- **Product information portal**: The website should contain the brief and full information about products such as price, discount price, image, name, short description, and full description.
- **Cart functions**: Web users are able to add the product to cart, remove, edit or add the new product to the existing cart.
- **Payment functions**: Users are able to do the payment process after choosing their shopping product. For example, the forms for users fill the delivering address, personal information, payment method etc.
- **Ordered product monitoring**: Users can edit, delete or add new products into paid order.
- **User login**: Users can register account and login as the web members.
- **Product searching**: Users can search the products by name.
- **Contact form**: Users can contact to the web admin by submitting the contact form.
- **Multilingual website (optional)**: The website can be displayed in Vietnamese (default language), English and Japanese.
- **Admin pages**: The website should have admin pages for web contents management

According to above requirements, I generated the samples of the website interface using HTML, CSS and send it to customer for approval. The final web interface and website organization are decided base on meeting between customer who is website owner and me as the website developer.

3.1.2 Sitemap and web interface

The web organization is design based on customer’s requirement. The sitemap of website www.mocaymart.com is demonstrated as figure 6.

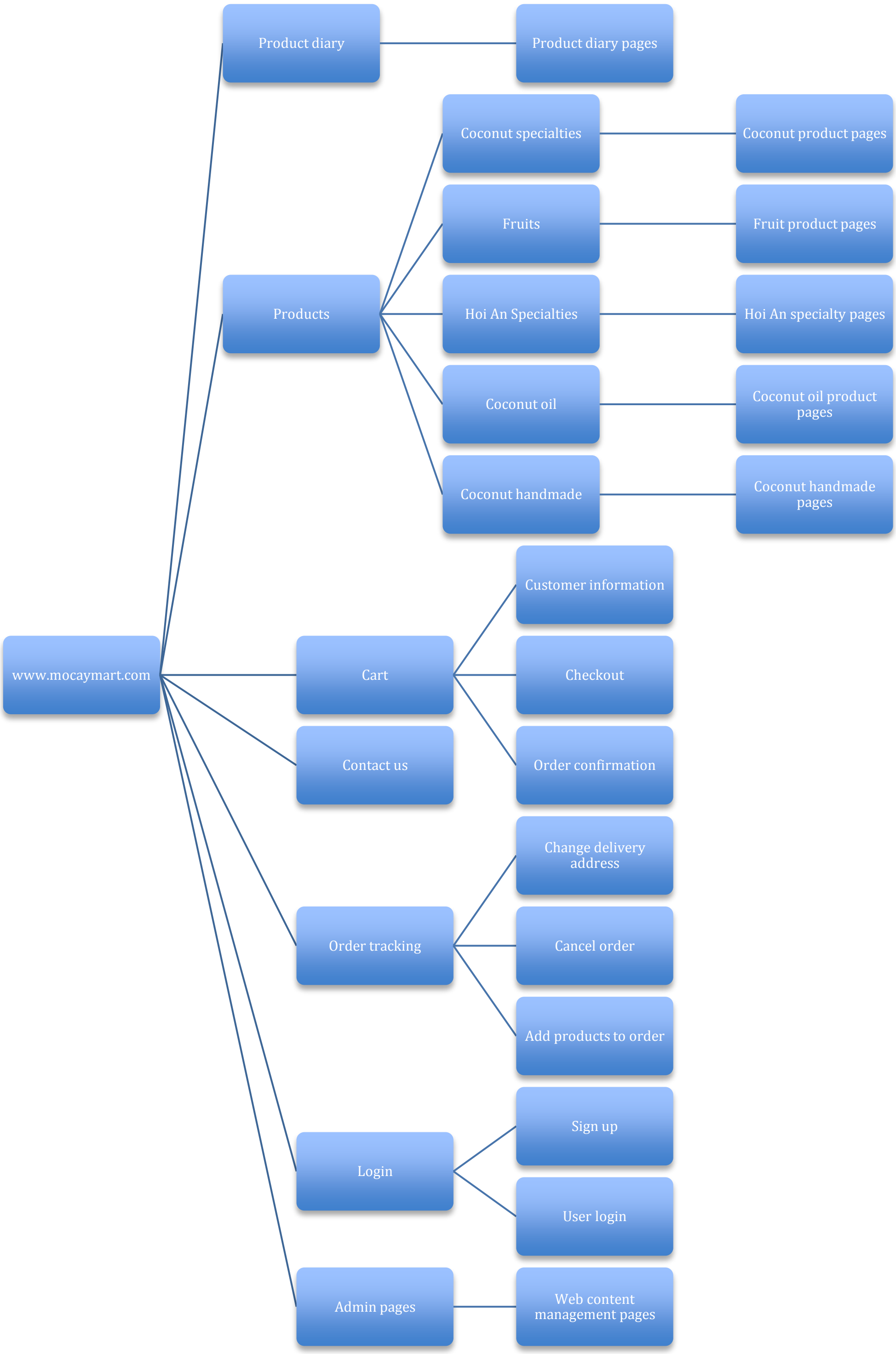


Figure 6: The sitemap of www.mocaymart.com

In general, user page consist of 4 functionalities: Information portal, cart related functions, contact form and user login. The table 1 explains those in more detail.

	Web pages	Description
Information Portal	Product diary pages	Tell the stories inside each product
	Product category pages	Show brief information of products in each category like image, name, price, discount price.
	Product pages	Show full information about products such as full of images, name, price, discount price, in stock status, short description, full description.
Cart functions	Customer information page	Display the form that allow shopper fill personal data for payment process
	Checkout page	Allow shopper choose payment method and delivery time
	Order confirmation	Show the order detail such as amount, delivery time, delivery address, payment method, order number (six digits) etc.
	Order tracking	Show the order status by order number
	Change delivery address	Allow shopper choose different address for delivering
	Cancel order	Shopper can cancel order by email to web admin
	Add products to order	Purchase new products in the same existing order
Contact form	Contact us page	Allow shopper send the message to web admin by filling contact form
User login	Sign up page	Shopper who would like to become the user of www.mocaymart.com website can fill the register form contained username, password and personal data.
	Login page	Users can log into his or her account by provide valid username and password

Table 1: Web pages functionality description

When the web sitemap and web pages description are done, I generated the web interface as following pictures.

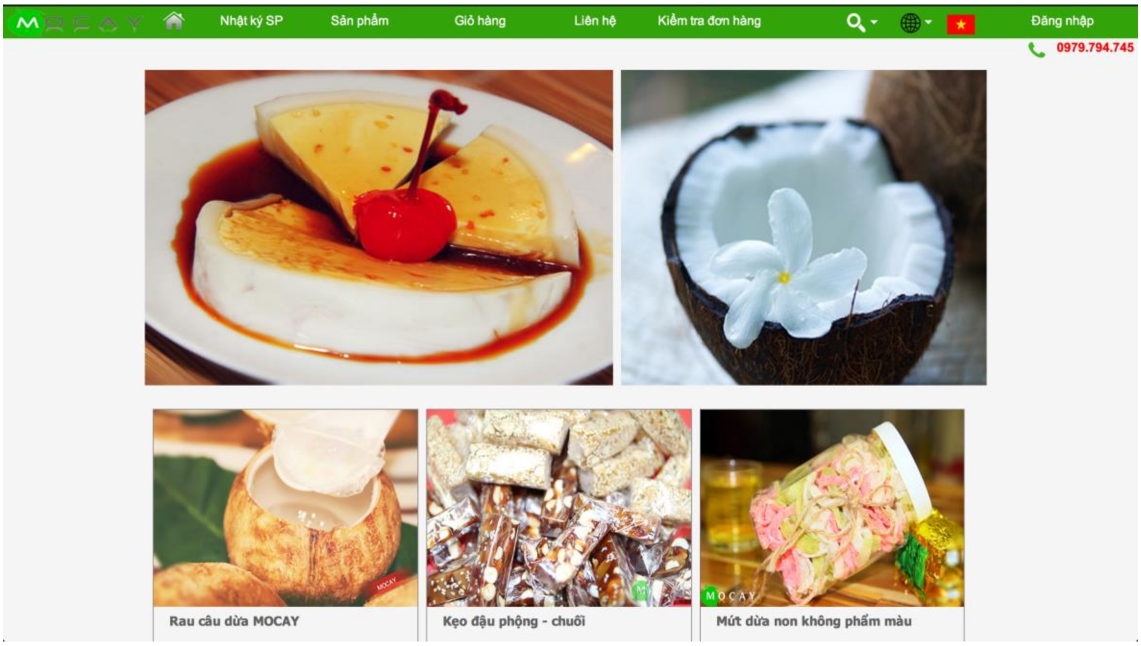
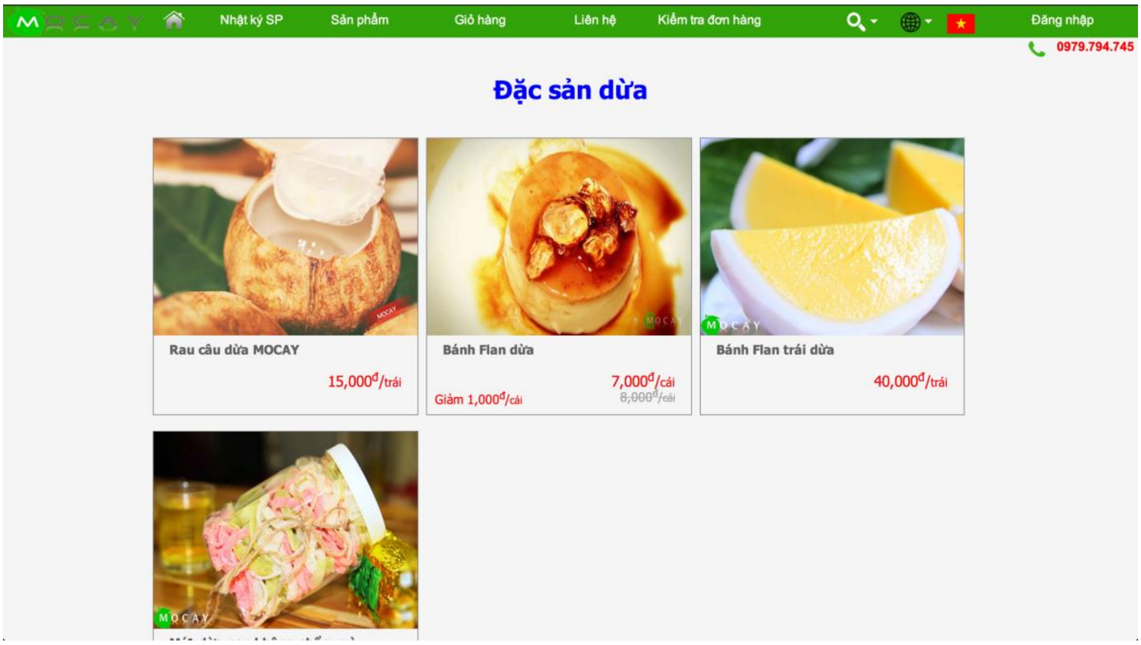


Figure 7: www.mocaymart.com home page



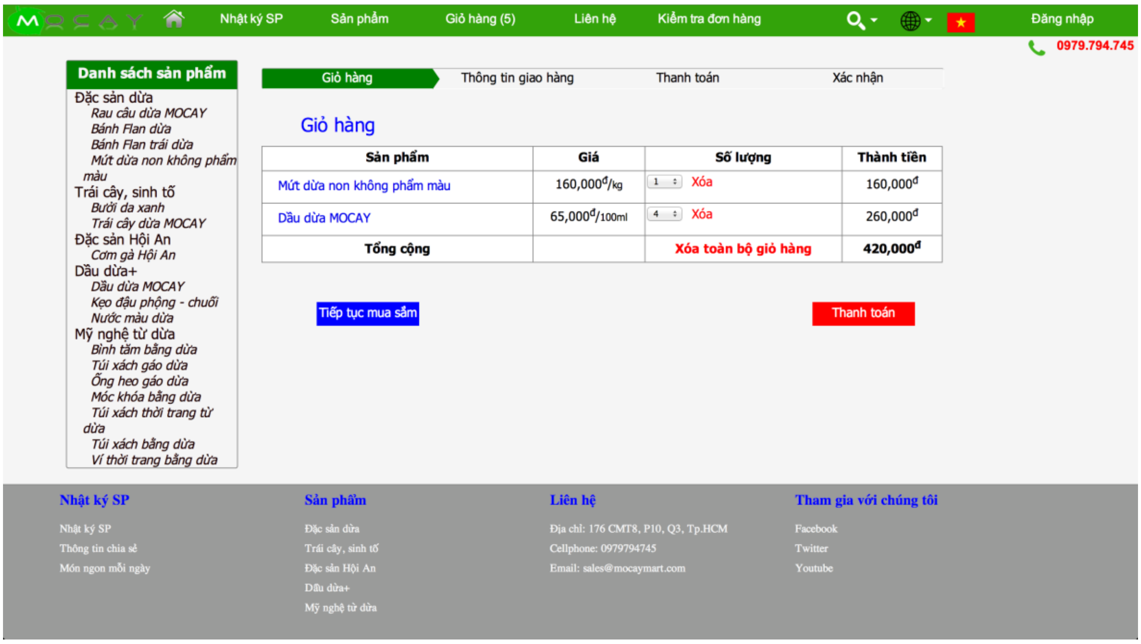
Đặc sản dừa

Rau câu dừa MOCAY
15,000^{đ}/trái

Bánh Flan dừa
7,000^{đ}/cái
~~8,000^{đ}/cái~~
Giảm 1,000^{đ}/cái

Bánh Flan trái dừa
40,000^{đ}/trái

Figure 8: www.mocaymart.com product category page



Danh sách sản phẩm

Đặc sản dừa
- *Rau câu dừa MOCAY*
- *Bánh Flan dừa*
- *Bánh Flan trái dừa*
- *Mứt dừa non không phẩm màu*

Trái cây, sinh tố
- *Bưởi da xanh*
- *Trái cây dừa MOCAY*

Đặc sản Hội An
- *Cơm gà Hội An*

Dầu dừa+
- *Dầu dừa MOCAY*
- *Kẹo đậu phộng - chuối*
- *Nước màu dừa*

Mỹ nghệ từ dừa
- *Bình tăm bằng dừa*
- *Túi xách gáo dừa*
- *Ống heo gáo dừa*
- *Móc khóa bằng dừa*
- *Túi xách thời trang từ dừa*
- *Túi xách bằng dừa*
- *Ví thời trang bằng dừa*

Giỏ hàng

Sản phẩm	Giá	Số lượng	Thành tiền
Mứt dừa non không phẩm màu	160,000^{đ}/kg	1 Xóa	160,000^{đ}
Dầu dừa MOCAY	65,000^{đ}/100ml	4 Xóa	260,000^{đ}
Tổng cộng		**Xóa toàn bộ giỏ hàng**	**420,000^{đ}**

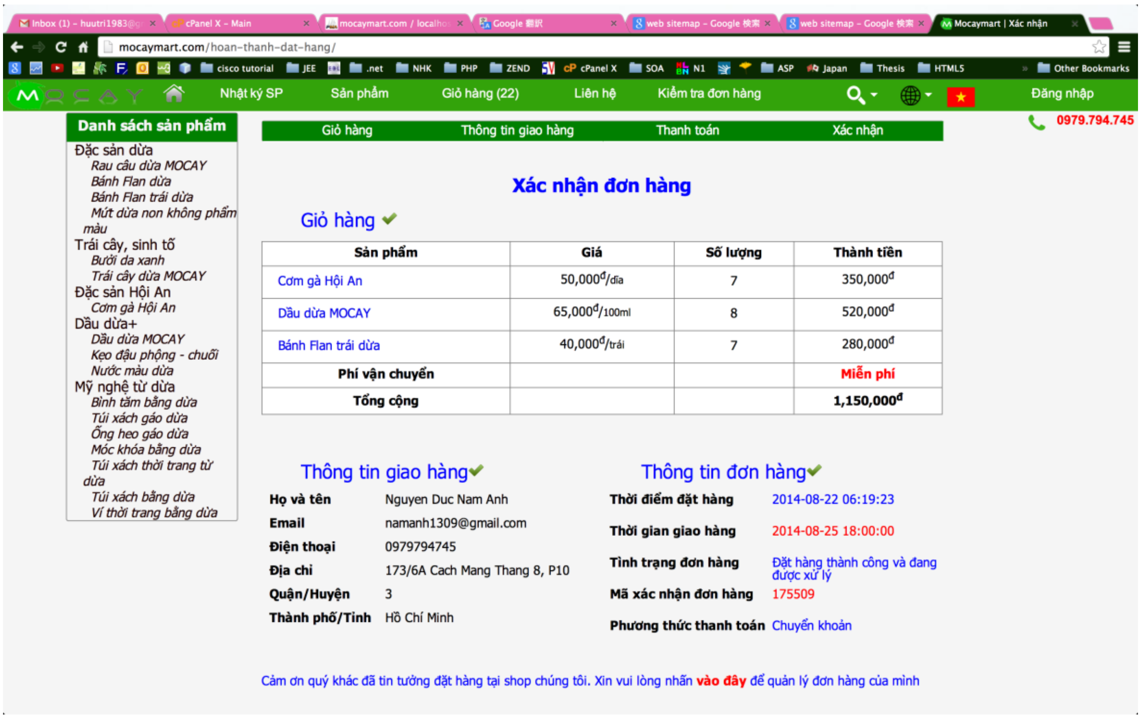
Nhật ký SP
Nhật ký SP
Thông tin chia sẻ
Món ngon mỗi ngày

Sản phẩm
Đặc sản dừa
Trái cây, sinh tố
Đặc sản Hội An
Dầu dừa+
Mỹ nghệ từ dừa

Liên hệ
Địa chỉ: 176 CMT8, P10, Q3, Tp.HCM
Cellphone: 0979794745
Email: sales@mocaymart.com

Tham gia với chúng tôi
Facebook
Twitter
Youtube

Figure 9: www.mocaymart.com shopping cart page



Danh sách sản phẩm

Đặc sản dừa
- *Rau câu dừa MOCAY*
- *Bánh Flan dừa*
- *Bánh Flan trái dừa*
- *Mứt dừa non không phẩm màu*

Trái cây, sinh tố
- *Bưởi da xanh*
- *Trái cây dừa MOCAY*

Đặc sản Hội An
- *Cơm gà Hội An*

Dầu dừa+
- *Dầu dừa MOCAY*
- *Kẹo đậu phộng - chuối*
- *Nước màu dừa*

Mỹ nghệ từ dừa
- *Bình tăm bằng dừa*
- *Túi xách gáo dừa*
- *Ống heo gáo dừa*
- *Móc khóa bằng dừa*
- *Túi xách thời trang từ dừa*
- *Túi xách bằng dừa*
- *Ví thời trang bằng dừa*

Xác nhận đơn hàng

Giỏ hàng ✓

Sản phẩm	Giá	Số lượng	Thành tiền
Cơm gà Hội An	50,000^{đ}/dĩa	7	350,000^{đ}
Dầu dừa MOCAY	65,000^{đ}/100ml	8	520,000^{đ}
Bánh Flan trái dừa	40,000^{đ}/trái	7	280,000^{đ}
Phí vận chuyển			**Miễn phí**
Tổng cộng			**1,150,000^{đ}**

Thông tin giao hàng ✓

Họ và tên	Nguyen Duc Nam Anh
Email	namanh1309@gmail.com
Điện thoại	0979794745
Địa chi	173/6A Cach Mang Thang 8, P10
Quận/Huyện	3
Thành phố/Tỉnh	Hồ Chí Minh

Thông tin đơn hàng ✓

Thời điểm đặt hàng	2014-08-22 06:19:23
Thời gian giao hàng	2014-08-25 18:00:00
Tình trạng đơn hàng	Đặt hàng thành công và đang được xử lý
Mã xác nhận đơn hàng	175509
Phương thức thanh toán	Chuyển khoản

Cảm ơn quý khác đã tin tưởng đặt hàng tại shop chúng tôi. Xin vui lòng nhấn **vào đây** để quản lý đơn hàng của mình

Figure 10: www.mocaymart.com order confirmation page

3.2 Database design

Basically, the database design for www.mocaymart.com website contains two main steps: requirements collection and database design which can be divided into 3 sub-steps named conceptual database design, logical database design, and physical database design.

3.2.1 Requirements collection

This is the vital step in the whole database design process. The requirements for website functions, workflows, use cases and business process website owner should be gathered from website owner so that we can design and implement an appropriate database.

In the case of www.mocaymart.com, the used tool for this stage are activity diagram for describing the activities of online shopper and use case diagram which demonstrate the interaction between website and users such as web admin, online shopper etc. The following activity diagram and use case diagram are generated based on the information of interview toward website owner.

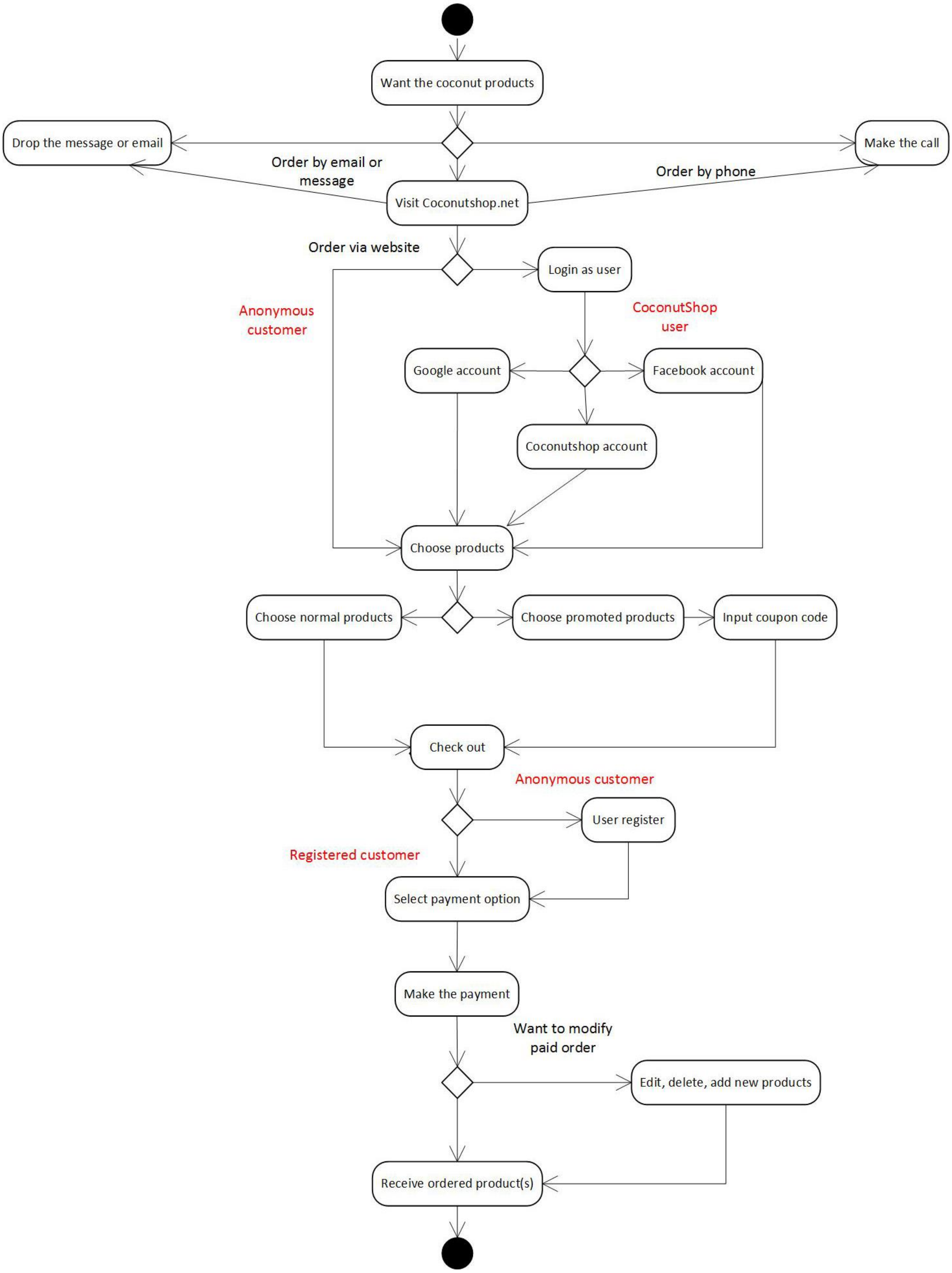


Figure 11: www.mocaymart.com website online shopper activity diagram

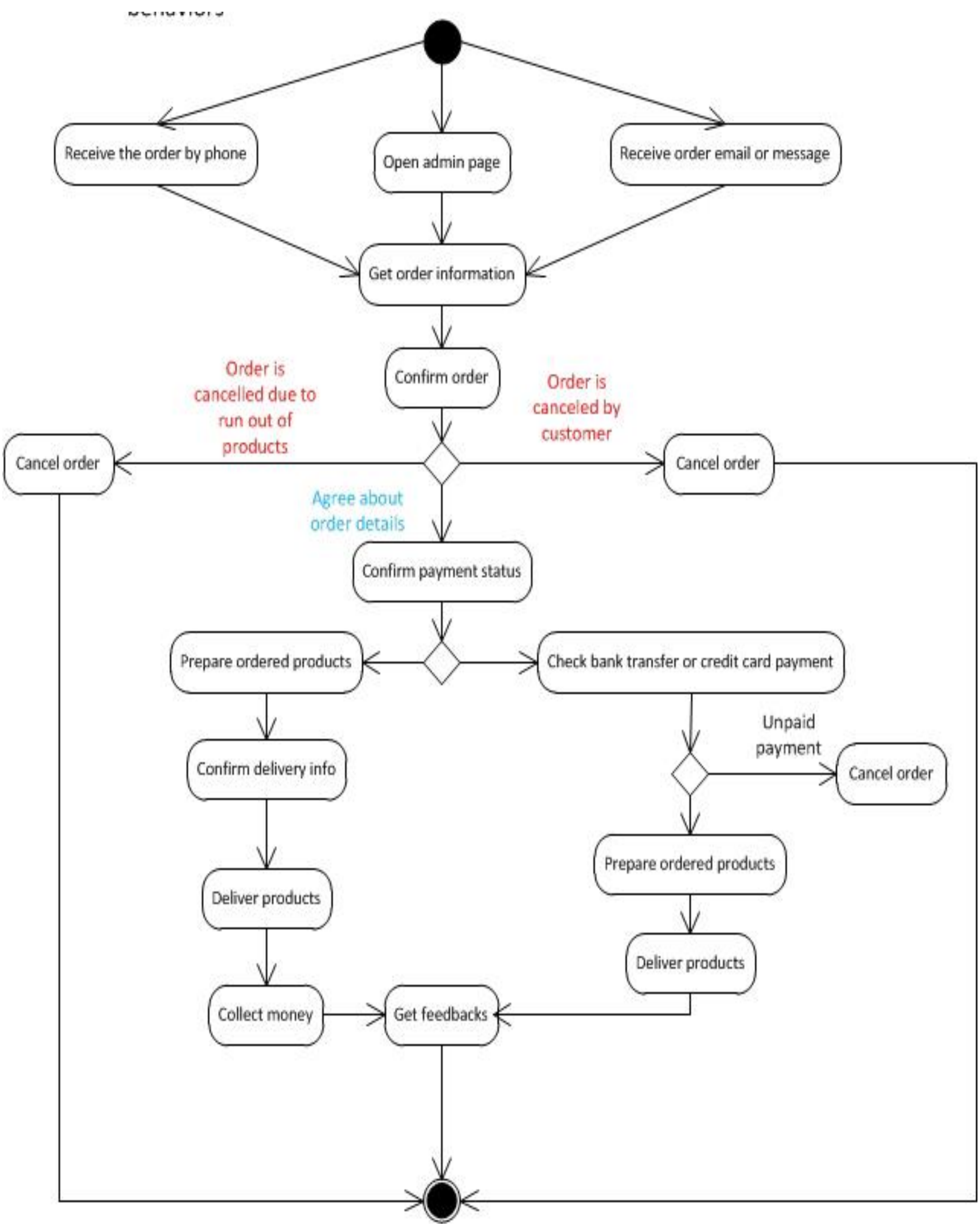


Figure 12: www.mocaymart.com web admin activity diagram

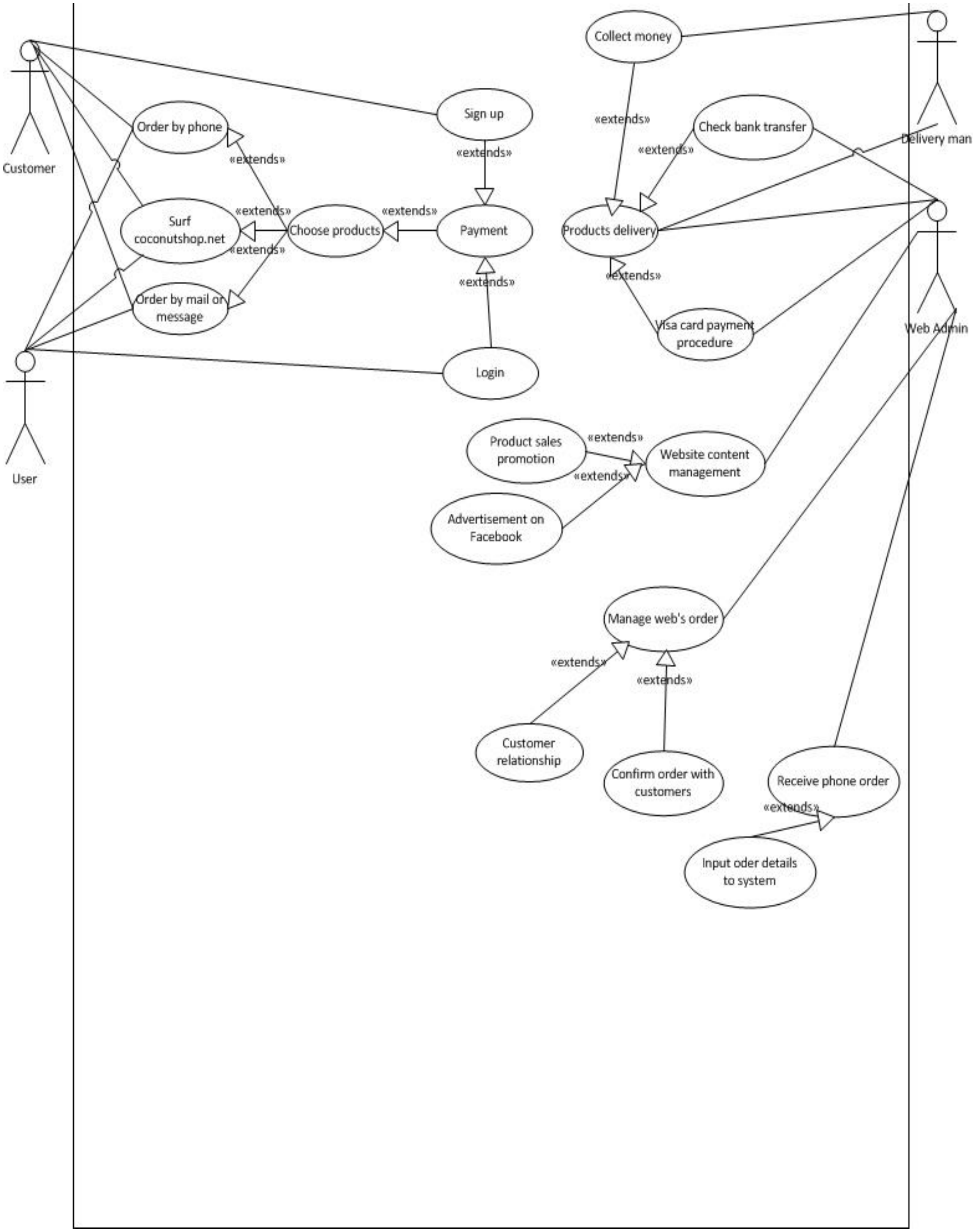


Figure 13: www.mocaymart.com website use case diagram

According to online shopper activity diagram, there are two types of web user that are described as bellow.

- **Anonymous customers**: Those online shoppers purchase products via www.mocaymart.com website without any user account. They merely add desired products to shopping cart; fill the form of personal information, delivery details, and payment method and the products would be delivered after confirmation process. The website does not record their information and they are not the object of some campaigns such as special offer for the regular customer, newsletter service etc.

- **Registered customers**: Those who register the account on www.mocaymart.com by providing the username, password and other personal data. They get some benefits in comparison with anonymous customers like special offers that only are available for registered customers, newsletter service, coupon for discount price etc.

Web admins, on the other hand, are responsible for two main tasks: web content managements such as posting new information about product or updating website data and order management like getting order information, order confirmation, order cancelation etc.

3.2.2 Database design

This step is for translating the data requirements and functional requirements, which are resulted from requirements collection step to the database. The outcome should be the physical database schemas that are implemented on the database server. It is divided into 3 substeps as following:

- **Conceptual design**: The goal of this step is creating the Entity-Relationship (ER) diagram that defines the database entities and their relationship.
- **Logical design**: The result of this step is the set of relation schemas. It is basic for the tables of the database and primary keys; foreign keys should be determined in this phase.
- **Physical design**: The main goal of this phase is database implementation into the database server. Thus, the chosen database management system (DBMS) should be decided. In this phase, the sequence query language (SQL) is used to create the database schema in the database server.

Figure 14 and figure 15 show the extended entity-relationship (EER) diagram and generated database schema, which are the outcomes of logical design and physical design respectively.

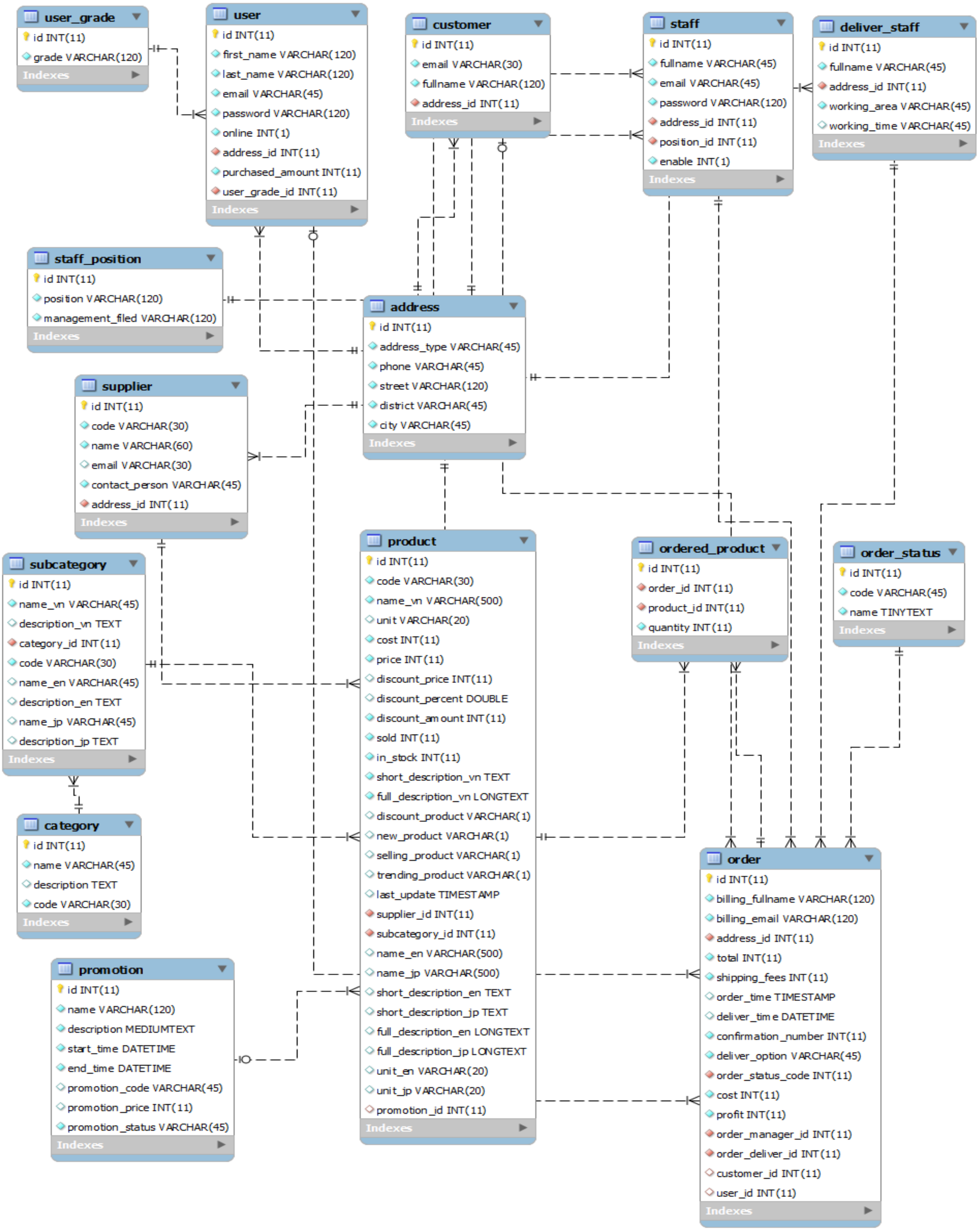


Figure 14: The EER diagram of www.mocaymart.com website database

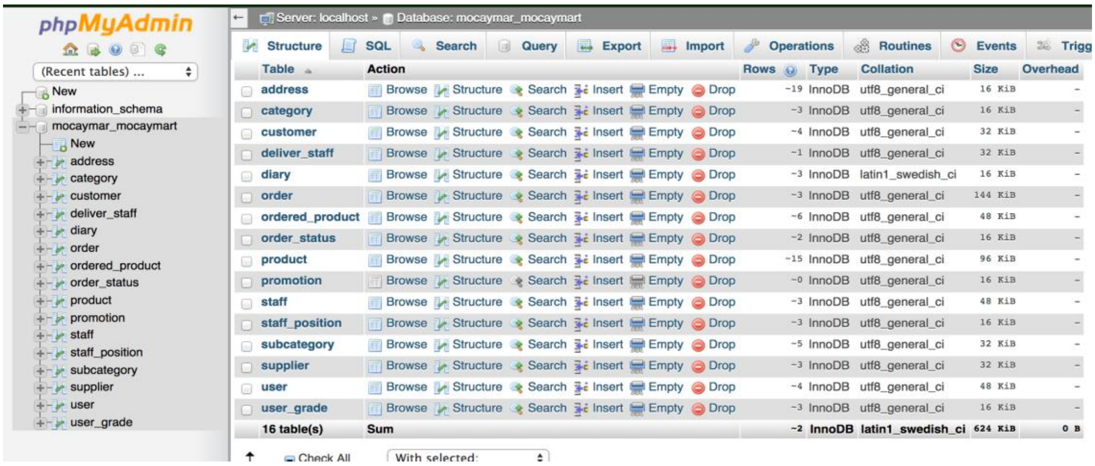
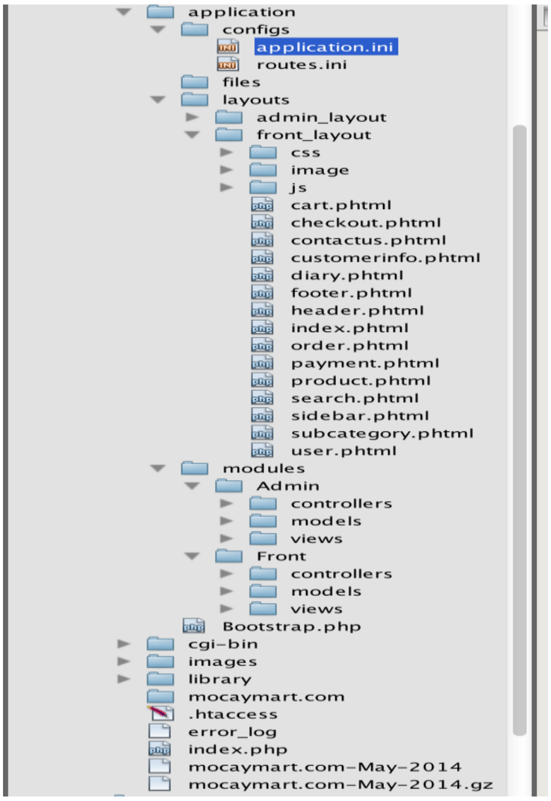


	Table	Action						Rows	Type	Collation	Size	Overhead
□	address	Browse	Structure	Search	Insert	Empty	Drop	~19	InnoDB	utf8_general_ci	16 KiB	-
□	category	Browse	Structure	Search	Insert	Empty	Drop	~3	InnoDB	utf8_general_ci	16 KiB	-
□	customer	Browse	Structure	Search	Insert	Empty	Drop	~4	InnoDB	utf8_general_ci	32 KiB	-
□	deliver_staff	Browse	Structure	Search	Insert	Empty	Drop	~1	InnoDB	utf8_general_ci	32 KiB	-
□	diary	Browse	Structure	Search	Insert	Empty	Drop	~3	InnoDB	latin1_swedish_ci	16 KiB	-
□	order	Browse	Structure	Search	Insert	Empty	Drop	~3	InnoDB	utf8_general_ci	144 KiB	-
□	ordered_product	Browse	Structure	Search	Insert	Empty	Drop	~6	InnoDB	utf8_general_ci	48 KiB	-
□	order_status	Browse	Structure	Search	Insert	Empty	Drop	~2	InnoDB	utf8_general_ci	16 KiB	-
□	product	Browse	Structure	Search	Insert	Empty	Drop	~15	InnoDB	utf8_general_ci	96 KiB	-
□	promotion	Browse	Structure	Search	Insert	Empty	Drop	~0	InnoDB	utf8_general_ci	16 KiB	-
□	staff	Browse	Structure	Search	Insert	Empty	Drop	~3	InnoDB	utf8_general_ci	48 KiB	-
□	staff_position	Browse	Structure	Search	Insert	Empty	Drop	~3	InnoDB	utf8_general_ci	16 KiB	-
□	subcategory	Browse	Structure	Search	Insert	Empty	Drop	~5	InnoDB	utf8_general_ci	32 KiB	-
□	supplier	Browse	Structure	Search	Insert	Empty	Drop	~3	InnoDB	utf8_general_ci	32 KiB	-
□	user	Browse	Structure	Search	Insert	Empty	Drop	~4	InnoDB	utf8_general_ci	48 KiB	-
□	user_grade	Browse	Structure	Search	Insert	Empty	Drop	~3	InnoDB	utf8_general_ci	16 KiB	-
	16 table(s)	Sum						~2	InnoDB	latin1_swedish_ci	624 KiB	0 B

Figure 15: www.mocaymart.com website database schema

3.3 Back-end developing

The www.mocaymart.com website uses Zend framework that is a PHP web application framework as back-end technology. The back-end developing process mainly contains 4 stages: Zend framework configuration, controller layer developing, model layer developing and view layer developing. NetBeans is chosen as integrated development environment (IDE) for the whole project. Figure 16 shows the structure of Zend framework in the case of www.mocaymart.com website.

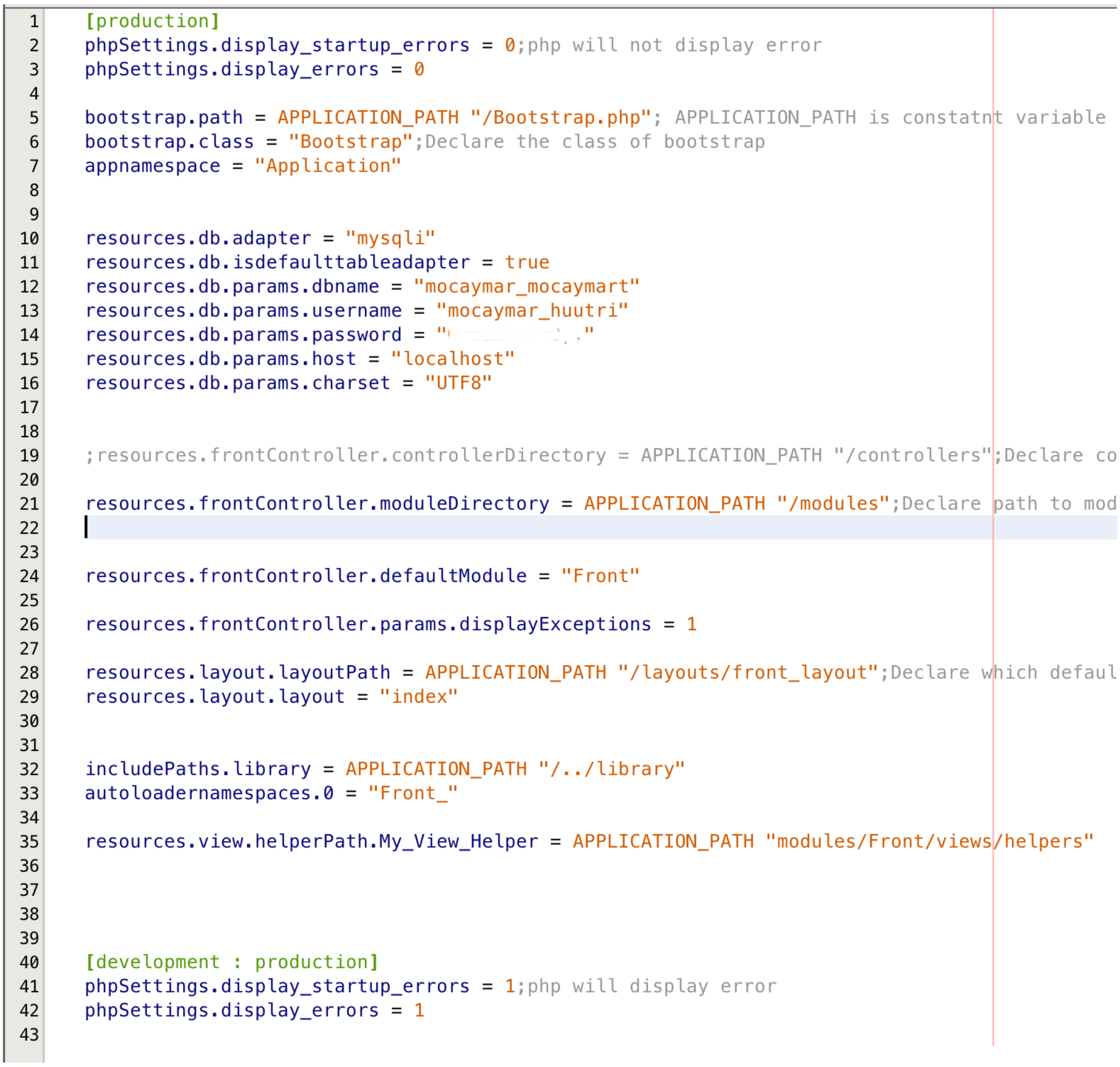


```
application
  configs
    application.ini
    routes.ini
  files
  layouts
    admin_layout
    front_layout
      css
      image
      js
      cart.phtml
      checkout.phtml
      contactus.phtml
      customerinfo.phtml
      diary.phtml
      footer.phtml
      header.phtml
      index.phtml
      order.phtml
      payment.phtml
      product.phtml
      search.phtml
      sidebar.phtml
      subcategory.phtml
      user.phtml
  modules
    Admin
      controllers
      models
      views
    Front
      controllers
      models
      views
  Bootstrap.php
cgi-bin
images
library
mocaymart.com
.htaccess
error_log
index.php
mocaymart.com-May-2014
mocaymart.com-May-2014.gz
```

Figure 16: www.mocaymart.com website back-end folder structure

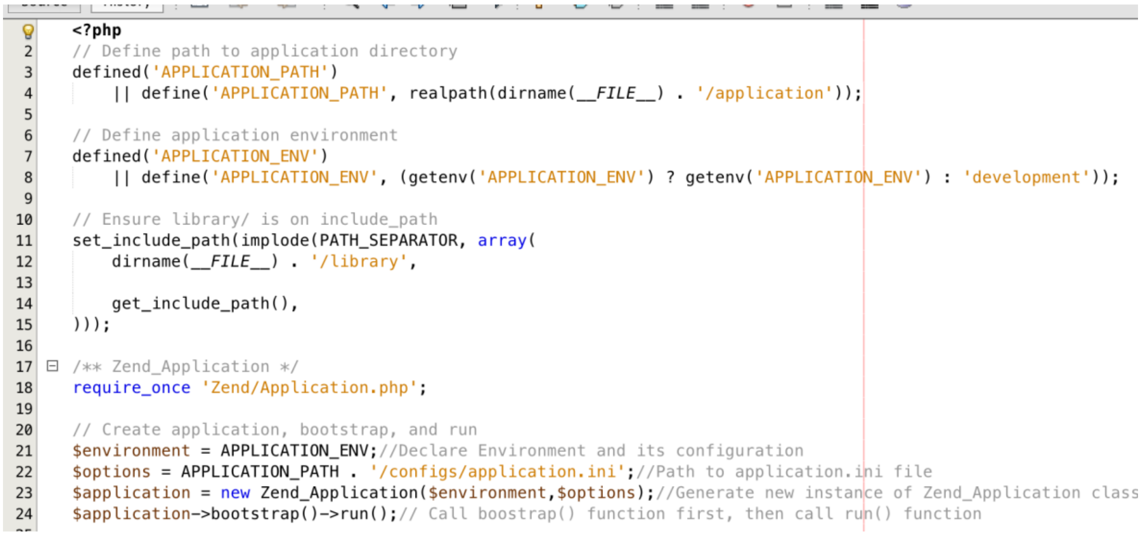
3.3.1 Zend framework configuration

Zend framework requires configuration before it can be used. The configuration is defined in 3 files: “index.php”, “application.ini” and “Boostrap.php”. The “index.php” file in Zend framework play the role of receptionist who receives all requests from the web user and consequently dispatch its proper component for further process. “Boostrap.php” and “application.ini”, on the other hand, contain the information of database connection, the default module, the path to Zend framework library etc. The further explanation of those configuration files can be found in below figures.



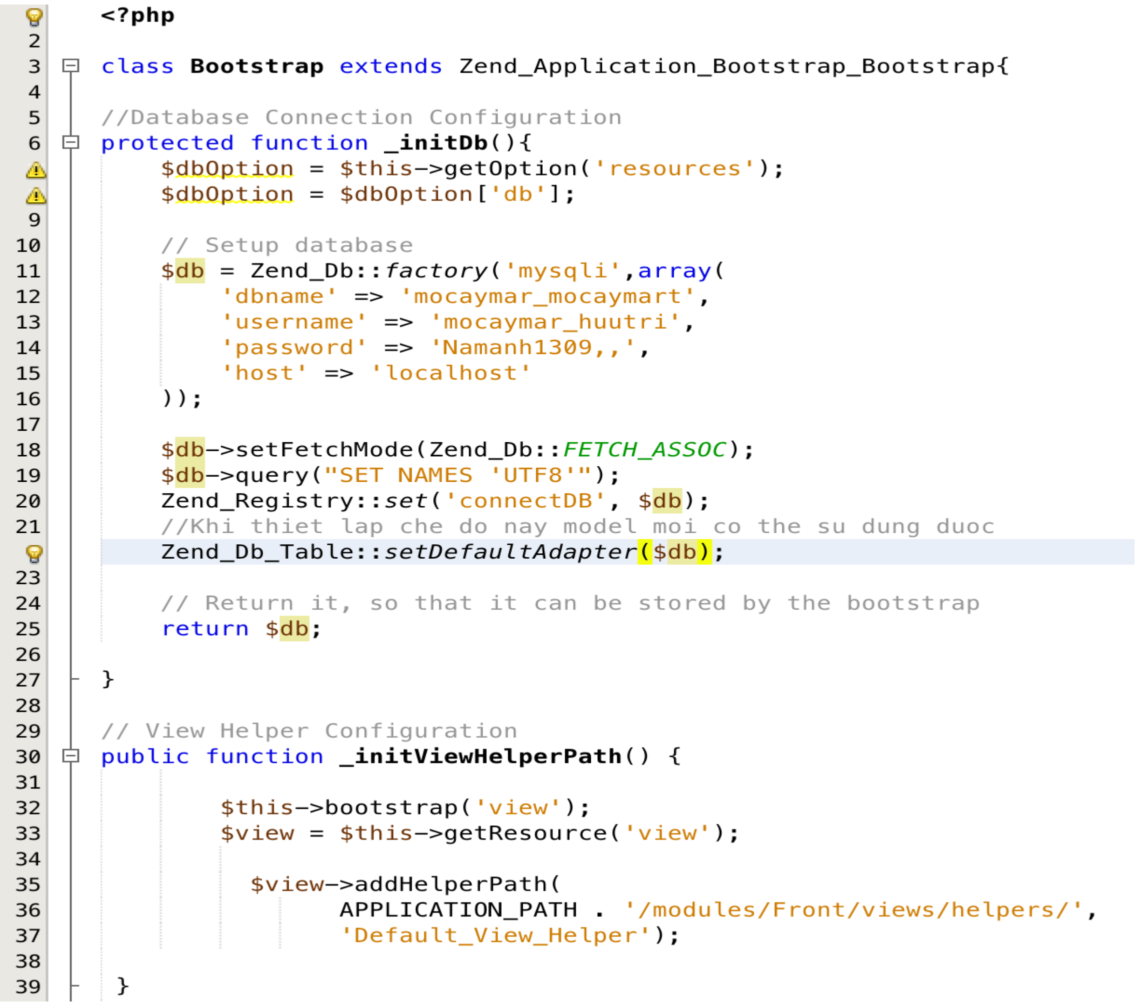
```
[production]
phpSettings.display_startup_errors = 0;php will not display error
phpSettings.display_errors = 0

bootstrap.path = APPLICATION_PATH "/Bootstrap.php"; APPLICATION_PATH is constatnt variable
bootstrap.class = "Bootstrap";Declare the class of bootstrap
appnamespace = "Application"


resources.db.adapter = "mysqli"
resources.db.isdefaulttableadapter = true
resources.db.params.dbname = "mocaymar_mocaymart"
resources.db.params.username = "mocaymar_huutri"
resources.db.params.password = "[redacted]"
resources.db.params.host = "localhost"
resources.db.params.charset = "UTF8"


;resources.frontController.controllerDirectory = APPLICATION_PATH "/controllers";Declare co

resources.frontController.moduleDirectory = APPLICATION_PATH "/modules";Declare path to mod


resources.frontController.defaultModule = "Front"

resources.frontController.params.displayExceptions = 1

resources.layout.layoutPath = APPLICATION_PATH "/layouts/front_layout";Declare which defaul
resources.layout.layout = "index"


includePaths.library = APPLICATION_PATH "/../library"
autoloadernamespaces.0 = "Front_"

resources.view.helperPath.My_View_Helper = APPLICATION_PATH "modules/Front/views/helpers"




[development : production]
phpSettings.display_startup_errors = 1;php will display error
phpSettings.display_errors = 1

```

Figure 17: application.ini file in www.mocaymart.com website



```php
<?php
// Define path to application directory
defined('APPLICATION_PATH')
    || define('APPLICATION_PATH', realpath(dirname(__FILE__) . '/application'));

// Define application environment
defined('APPLICATION_ENV')
    || define('APPLICATION_ENV', (getenv('APPLICATION_ENV') ? getenv('APPLICATION_ENV') : 'development'));

// Ensure library/ is on include_path
set_include_path(implode(PATH_SEPARATOR, array(
    dirname(__FILE__) . '/library',

    get_include_path(),
)));

/** Zend_Application */
require_once 'Zend/Application.php';

// Create application, bootstrap, and run
$environment = APPLICATION_ENV;//Declare Environment and its configuration
$options = APPLICATION_PATH . '/configs/application.ini';//Path to application.ini file
$application = new Zend_Application($environment,$options);//Generate new instance of Zend_Application class
$application->bootstrap()->run();// Call boostrap() function first, then call run() function
```

Figure 18: index.php file in www.mocaymart.com website



```php
<?php

class Bootstrap extends Zend_Application_Bootstrap_Bootstrap{

//Database Connection Configuration
protected function _initDb(){
    $dbOption = $this->getOption('resources');
    $dbOption = $dbOption['db'];

    // Setup database
    $db = Zend_Db::factory('mysqli',array(
        'dbname' => 'mocaymar_mocaymart',
        'username' => 'mocaymar_huutri',
        'password' => 'Namanh1309,,',
        'host' => 'localhost'
    ));

    $db->setFetchMode(Zend_Db::FETCH_ASSOC);
    $db->query("SET NAMES 'UTF8'");
    Zend_Registry::set('connectDB', $db);
    //Khi thiet lap che do nay model moi co the su dung duoc
    Zend_Db_Table::setDefaultAdapter($db);

    // Return it, so that it can be stored by the bootstrap
    return $db;

}

// View Helper Configuration
public function _initViewHelperPath() {

        $this->bootstrap('view');
        $view = $this->getResource('view');

         $view->addHelperPath(
                APPLICATION_PATH . '/modules/Front/views/helpers/',
                'Default_View_Helper');

 }
```

Figure 19: Boostrap.php file in www.mocaymart.com website

3.3.2 Controller layer developing

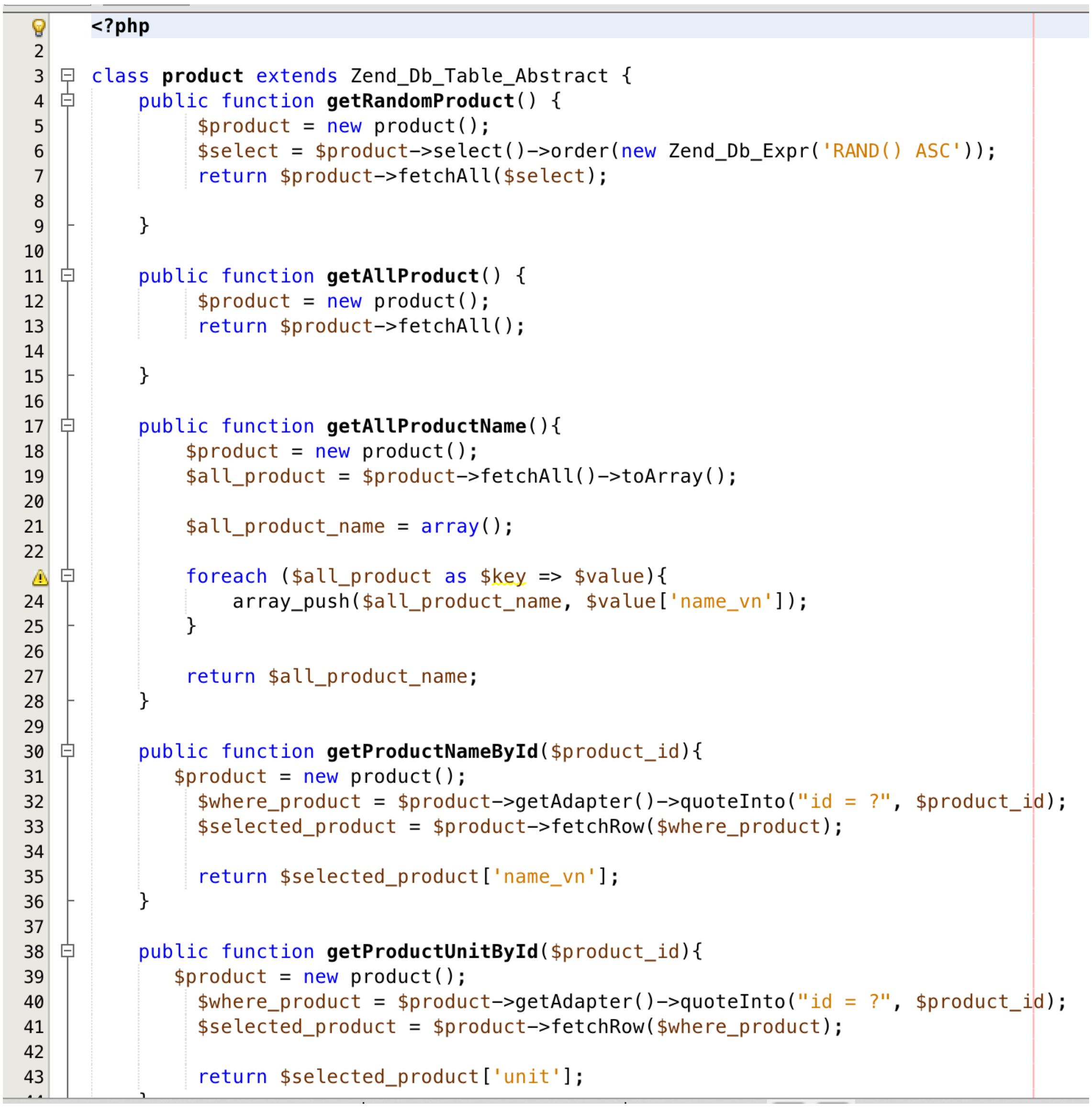
The controller layer in Zend framework process the request received from “index.php” file’s dispatching. Every controller class implements the “Zend_Controller” class that is the backbone of Zend framework’s MVC. The controller layer’s performance is triggered when a web user clicks or types the URL with the structure **http://hostname/module/controller/action/parametes**. The mechanism of the controller layer is described in more detail via the example of “CartController.php” file as the appendix.

This controller belongs to “Front” module and it is responsible for controlling all action related to shopping cart such as “add to cart”, “update item”, “remove cart”. There are 4 public functions in this controller: “indexAction”, “addtocartAction”, “updateitemAction” and “removecartAction” that operate the actions of “open the shopping cart page”, “add the products to shopping cart”, “update the quantity of items in shopping cart” and “remove entire items from the shopping cart” respectively. For instance, when a web user clicks the URL http://www.mocaymart/Front/cart/removecart, firstly the “CartController” in “Front” module would be operated. Consequently, this controller conducts the function “removecartAction” to remove all items from existing shopping cart. This action used the “remove cart” function from the “cart” class in the model layer.

3.3.3 Model layer developing

Model in Zend framework is not only the mean for data access but in reality, it represents a system’s data and behaviour. The model is independent of controller and view in Zend framework (Chapter3. The Model). The model contains the business rules and the behaviour of related data. In back-end developing using Zend framework, model plays two vital roles: keeping all application data during the current request processing so that it can be retrieved and used for the next request and holding the business and constraint of the data. For example, the “CartItem.php” and “Cart.php” are 2 classes in model layer, which retain the data about product id, quantity, and price of each item when web user add new products to shopping cart. Those data would be used in checkout request when the total amount of items in shopping cart is calculated for payment.

In www.mocaymart.com website, each table in the database is represented by one class and the name of table and class should be identical. Those classes inherit from “Zend_Db_Table_Abstract” predefined class, which contain business logic for database connection. Figure 20 shows the example of a class in the model layer.



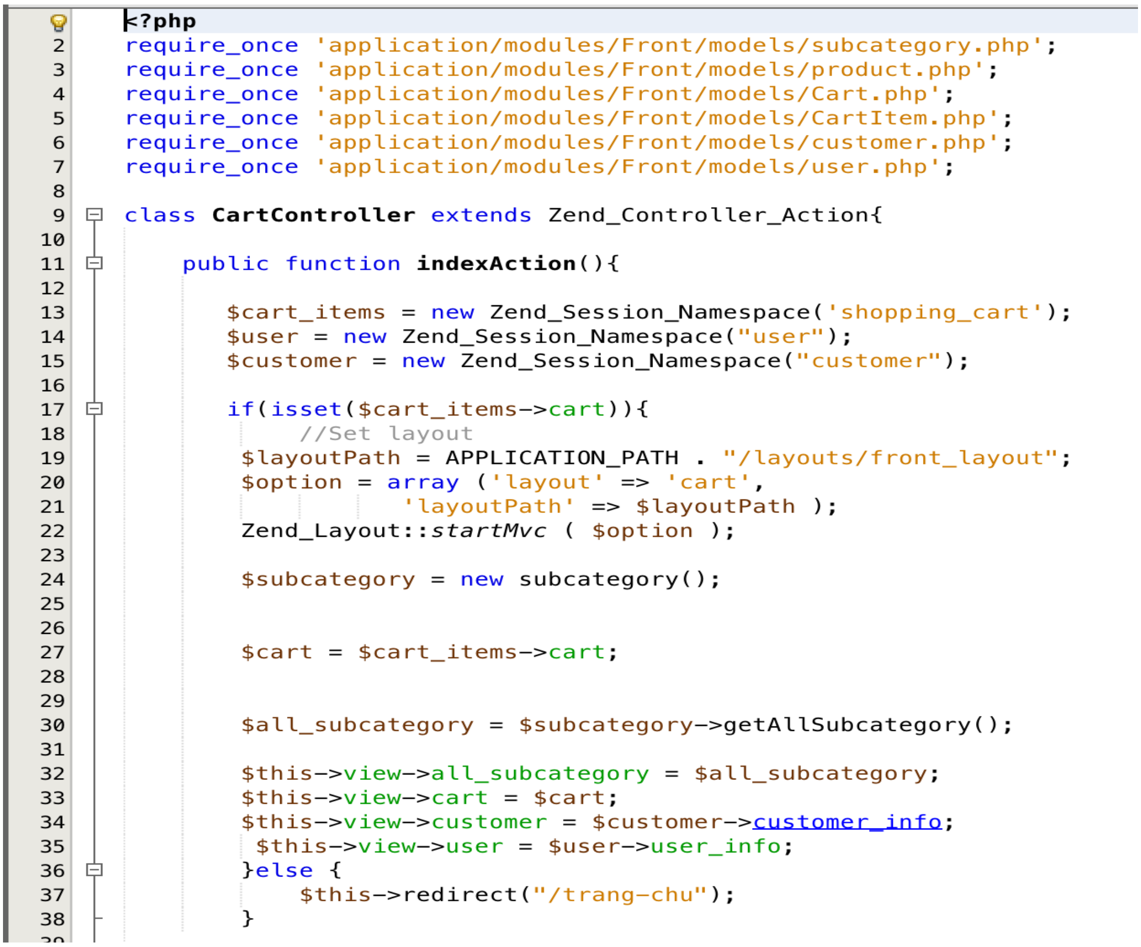
```php
<?php

class product extends Zend_Db_Table_Abstract {
    public function getRandomProduct() {
         $product = new product();
         $select = $product->select()->order(new Zend_Db_Expr('RAND() ASC'));
         return $product->fetchAll($select);

    }

    public function getAllProduct() {
         $product = new product();
         return $product->fetchAll();

    }

    public function getAllProductName(){
        $product = new product();
        $all_product = $product->fetchAll()->toArray();

        $all_product_name = array();

        foreach ($all_product as $key => $value){
            array_push($all_product_name, $value['name_vn']);
        }

        return $all_product_name;
    }

    public function getProductNameById($product_id){
       $product = new product();
         $where_product = $product->getAdapter()->quoteInto("id = ?", $product_id);
         $selected_product = $product->fetchRow($where_product);

         return $selected_product['name_vn'];
    }

    public function getProductUnitById($product_id){
       $product = new product();
         $where_product = $product->getAdapter()->quoteInto("id = ?", $product_id);
         $selected_product = $product->fetchRow($where_product);

         return $selected_product['unit'];
```

Figure 20: Model class in www.mocaymart.com website

The “product” class inherits from “Zend_Db_Table_Abstract” so that it can retrieve the data records of the product table in the database. Furthermore, it also has some functions to manipulate the data retrieved from product table such as “getAllProduct”, “getRandomProduct”, “getAllProductName”, “getProductNameById” etc.

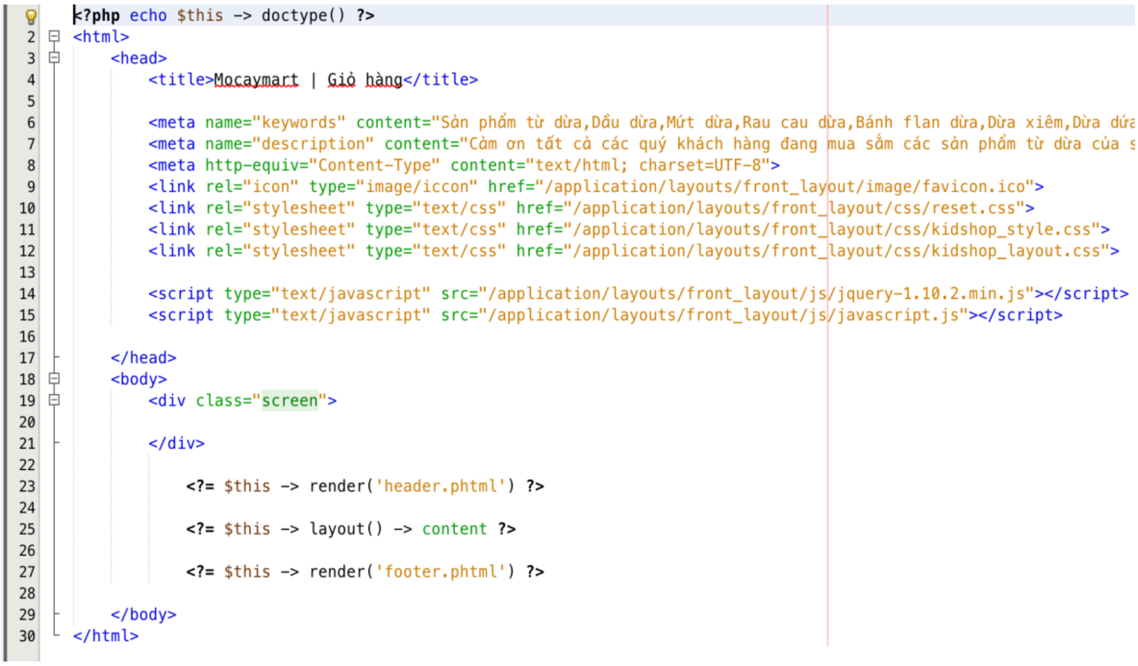
3.3.4 View layer developing

The view layer is in charge of creating user interface for the website. Each page in www.mocaymart.com website has its own layout which is defined in respective controller. The operation of view layer in Zend framework can be understood via following example.



```
<?php
require_once 'application/modules/Front/models/subcategory.php';
require_once 'application/modules/Front/models/product.php';
require_once 'application/modules/Front/models/Cart.php';
require_once 'application/modules/Front/models/CartItem.php';
require_once 'application/modules/Front/models/customer.php';
require_once 'application/modules/Front/models/user.php';

class CartController extends Zend_Controller_Action{

    public function indexAction(){

       $cart_items = new Zend_Session_Namespace('shopping_cart');
       $user = new Zend_Session_Namespace("user");
       $customer = new Zend_Session_Namespace("customer");

       if(isset($cart_items->cart)){
            //Set layout
        $layoutPath = APPLICATION_PATH . "/layouts/front_layout";
        $option = array ('layout' => 'cart',
                     'layoutPath' => $layoutPath );
        Zend_Layout::startMvc ( $option );

        $subcategory = new subcategory();


        $cart = $cart_items->cart;


        $all_subcategory = $subcategory->getAllSubcategory();

        $this->view->all_subcategory = $all_subcategory;
        $this->view->cart = $cart;
        $this->view->customer = $customer->customer_info;
         $this->view->user = $user->user_info;
        }else {
            $this->redirect("/trang-chu");
        }
```

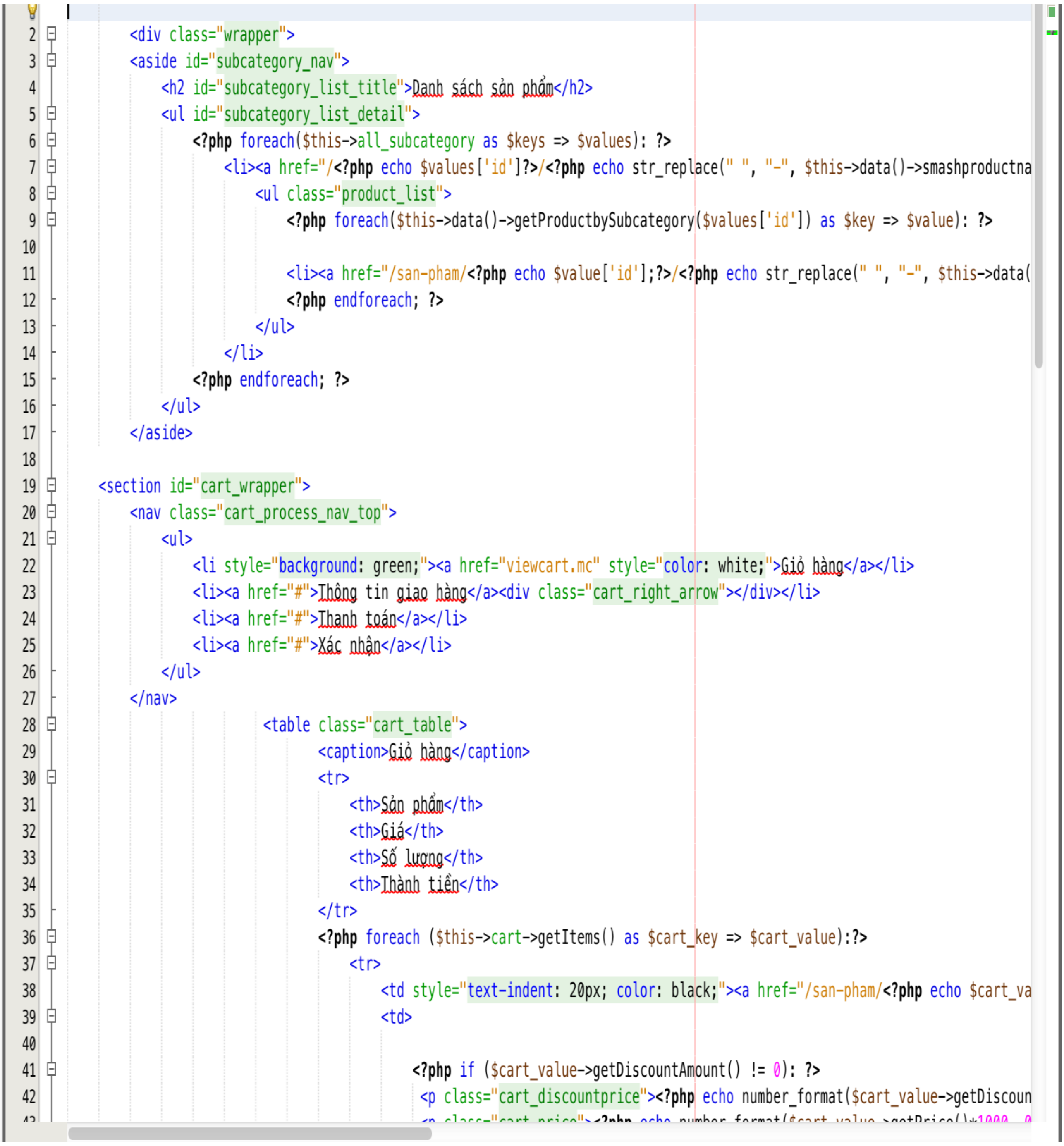
Figure 21: CartController controller in www.mocaymart.com website



```
<?php echo $this -> doctype() ?>
<html>
    <head>
        <title>Mocaymart | Giỏ hàng</title>

        <meta name="keywords" content="Sản phẩm từ dừa,Dầu dừa,Mứt dừa,Rau cau dừa,Bánh flan dừa,Dừa xiêm,Dừa dứa
        <meta name="description" content="Cảm ơn tất cả các quý khách hàng đang mua sắm các sản phẩm từ dừa của s
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link rel="icon" type="image/iccon" href="/application/layouts/front_layout/image/favicon.ico">
        <link rel="stylesheet" type="text/css" href="/application/layouts/front_layout/css/reset.css">
        <link rel="stylesheet" type="text/css" href="/application/layouts/front_layout/css/kidshop_style.css">
        <link rel="stylesheet" type="text/css" href="/application/layouts/front_layout/css/kidshop_layout.css">

        <script type="text/javascript" src="/application/layouts/front_layout/js/jquery-1.10.2.min.js"></script>
        <script type="text/javascript" src="/application/layouts/front_layout/js/javascript.js"></script>

    </head>
    <body>
        <div class="screen">

        </div>

            <?= $this -> render('header.phtml') ?>

            <?= $this -> layout() -> content ?>

            <?= $this -> render('footer.phtml') ?>

    </body>
</html>
```

Figure 22: cart.phtml layout in www.mocaymart.com website

“CartController” defines “cart” as the layout of shopping cart page. This layout is described in “cart.phmtl” file as figure 22. According to this, the shopping cart page has 3 parts: the header and footer part, which renders the “header.phtml” and “footer.phtml” files respectively and the main part, which renders the file as figure 23.



```

        <div class="wrapper">
        <aside id="subcategory_nav">
            <h2 id="subcategory_list_title">Danh sách sản phẩm</h2>
            <ul id="subcategory_list_detail">
                <?php foreach($this->all_subcategory as $keys => $values): ?>
                    <li><a href="/<?php echo $values['id']?>/<?php echo str_replace(" ", "-", $this->data()->smashproductna
                        <ul class="product_list">
                            <?php foreach($this->data()->getProductbySubcategory($values['id']) as $key => $value): ?>

                            <li><a href="/san-pham/<?php echo $value['id'];?>/<?php echo str_replace(" ", "-", $this->data(
                            <?php endforeach; ?>
                        </ul>
                    </li>
                <?php endforeach; ?>
            </ul>
        </aside>

    <section id="cart_wrapper">
        <nav class="cart_process_nav_top">
            <ul>
                <li style="background: green;"><a href="viewcart.mc" style="color: white;">Giỏ hàng</a></li>
                <li><a href="#">Thông tin giao hàng</a><div class="cart_right_arrow"></div></li>
                <li><a href="#">Thanh toán</a></li>
                <li><a href="#">Xác nhận</a></li>
            </ul>
        </nav>
                         <table class="cart_table">
                                <caption>Giỏ hàng</caption>
                                <tr>
                                    <th>Sản phẩm</th>
                                    <th>Giá</th>
                                    <th>Số lượng</th>
                                    <th>Thành tiền</th>
                                </tr>
                                <?php foreach ($this->cart->getItems() as $cart_key => $cart_value):?>
                                    <tr>
                                        <td style="text-indent: 20px; color: black;"><a href="/san-pham/<?php echo $cart_va
                                        <td>

                                            <?php if ($cart_value->getDiscountAmount() != 0): ?>
                                             <p class="cart_discountprice"><?php echo number_format($cart_value->getDiscoun
                                             <p class="cart_price"><?php echo number_format($cart_value->getPrice()*1000, 0
```

Figure 23: index.phtml file of shopping cart in www.mocaymart.com website

“Index.phtml” file in this example is responsible for rendering the main part of shopping cart page. The “CartController” created the “$all_subcategory” object to store all data of subcategory such as subcategory id, subcategory name and passed it to view layer with the name “all_subcategory”. Consequently, “index.phtml” file render information of all subcategory by PHP functions named “foreach” and “echo”.

4 Conclusion

This thesis describes the phases of graphic user interface design, database design and back-end developing in developing process of www.mocaymart.com website. The website is developed based on the requirements of the website owner who planned the start-up business plan in E-commerce field. Those general requirements are analysed in order to acquire the data requirements and functional requirements by which the database is created. Subsequently, back-end developing using Zend framework is operated to create the dynamic content for the website.

Presently, the web user quantity of www.mocaymart.com website is increasing and its’ contribution for the success of web owner’s business is highly evaluated. The functions, that are not available presently such as multilingual pages, login as Google and Facebook accounts etc., would be continuously developed. This website is expected to play the key role in the future’s business development.

References

Peacock, M. 2008. Selling online with Drupal E-Commerce. Packt Publishing.

Porebski, Bartosz; Przystalski, Karol; Nowak, Leszek 2011. Building PHP Applications with Symfony, CakePHP, and Zend Framework. Willey.

W3Schools. Accessed 16th August 2014.
http://www.w3schools.com

Vietnam Online – Understanding Vietnam's E-commerce Market. Accessed 16th August 2014.
http://www.vietnam-briefing.com/news/vietnam-online-understanding-vietnams-e-commerce-market.html/

What is a constructive approach? Accessed 19th January 2015.
http://www.metodix.com/en/sisallys/01_menetelmat/02_metodiartikkelit/lukka_const_research_app/02_mita_konst_tut_tark

Statistics for websites using Framework technologies. Accessed 18th August 2014.
http://trends.builtwith.com/framework

Business Advantages of a Relational Database. Accessed 19th August 2014.
http://www.insidebusiness360.com/index.php/business-advantages-of-a-relational-database-15806/

Main phases of database design. Accessed 26th August 2014.
http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/0303011/1146161367915/1146161680673/1146161874577/1146161968355.html

Zend Framework: Surviving The Deep End, Chapter 3 The Model. Accessed 28th August 2014.
http://www.survivethedeepend.com/zendframeworkbook/en/1.0/the.model

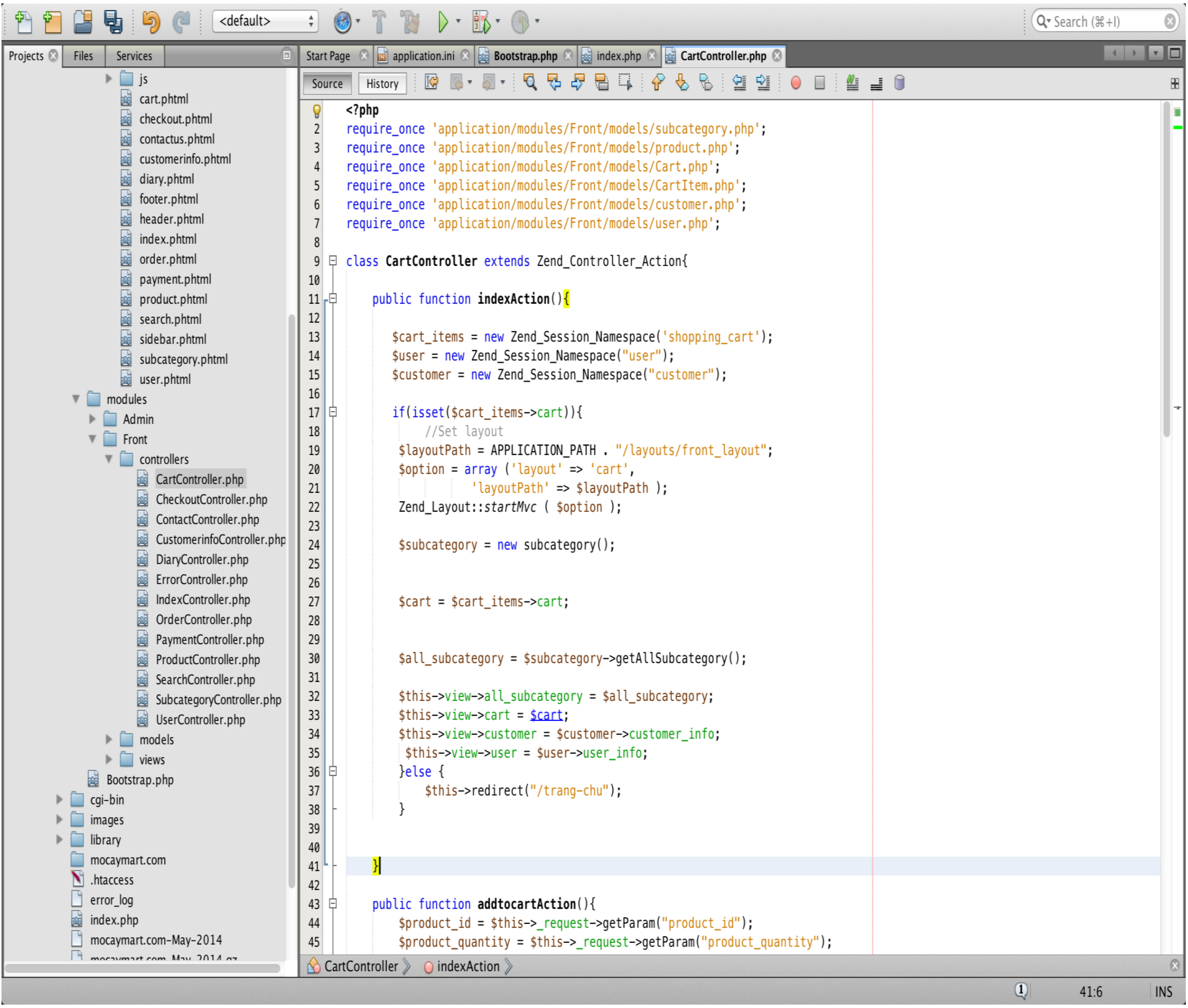
Figures

Tables

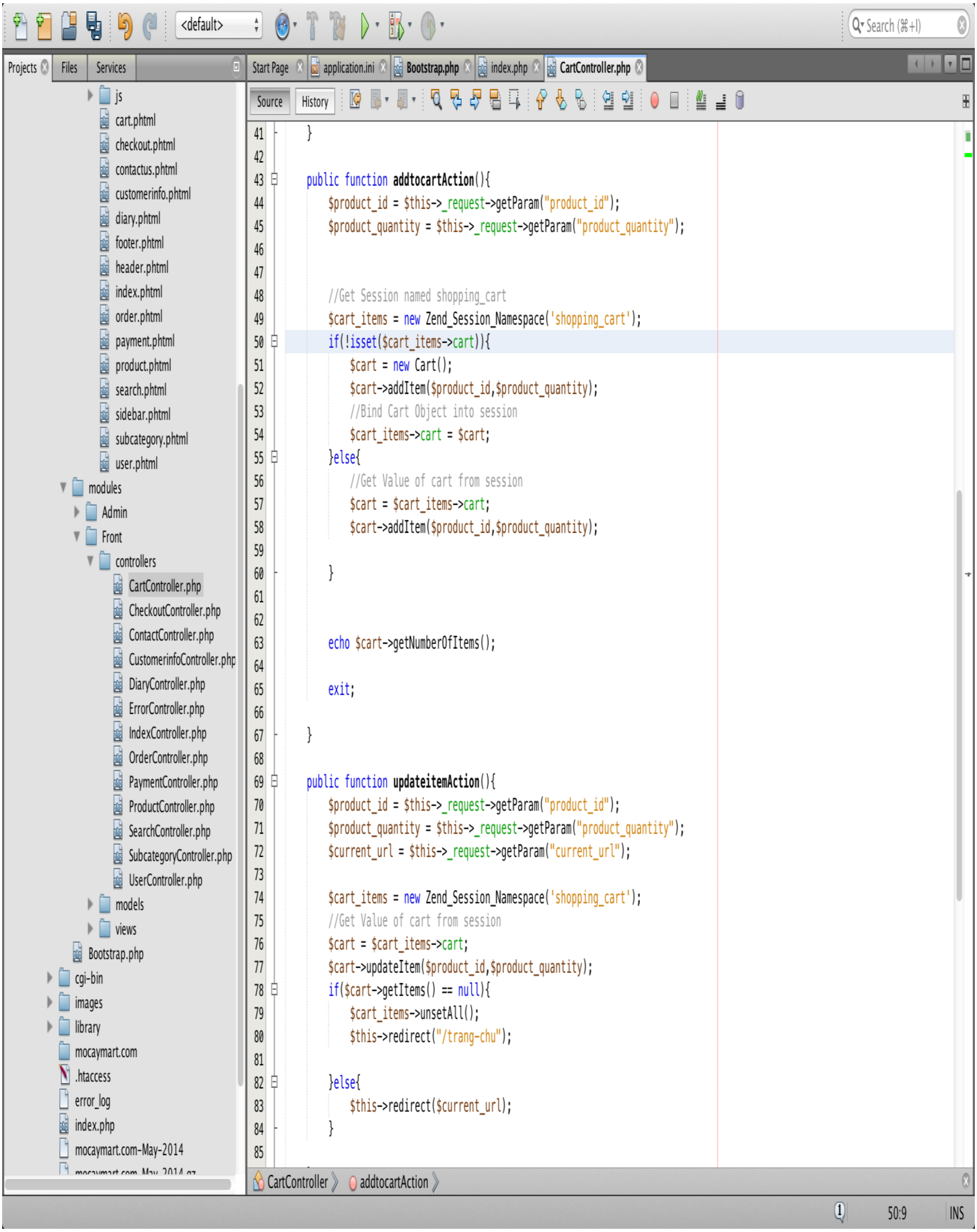
Appendix

Question for www.mocaymart.com web owner

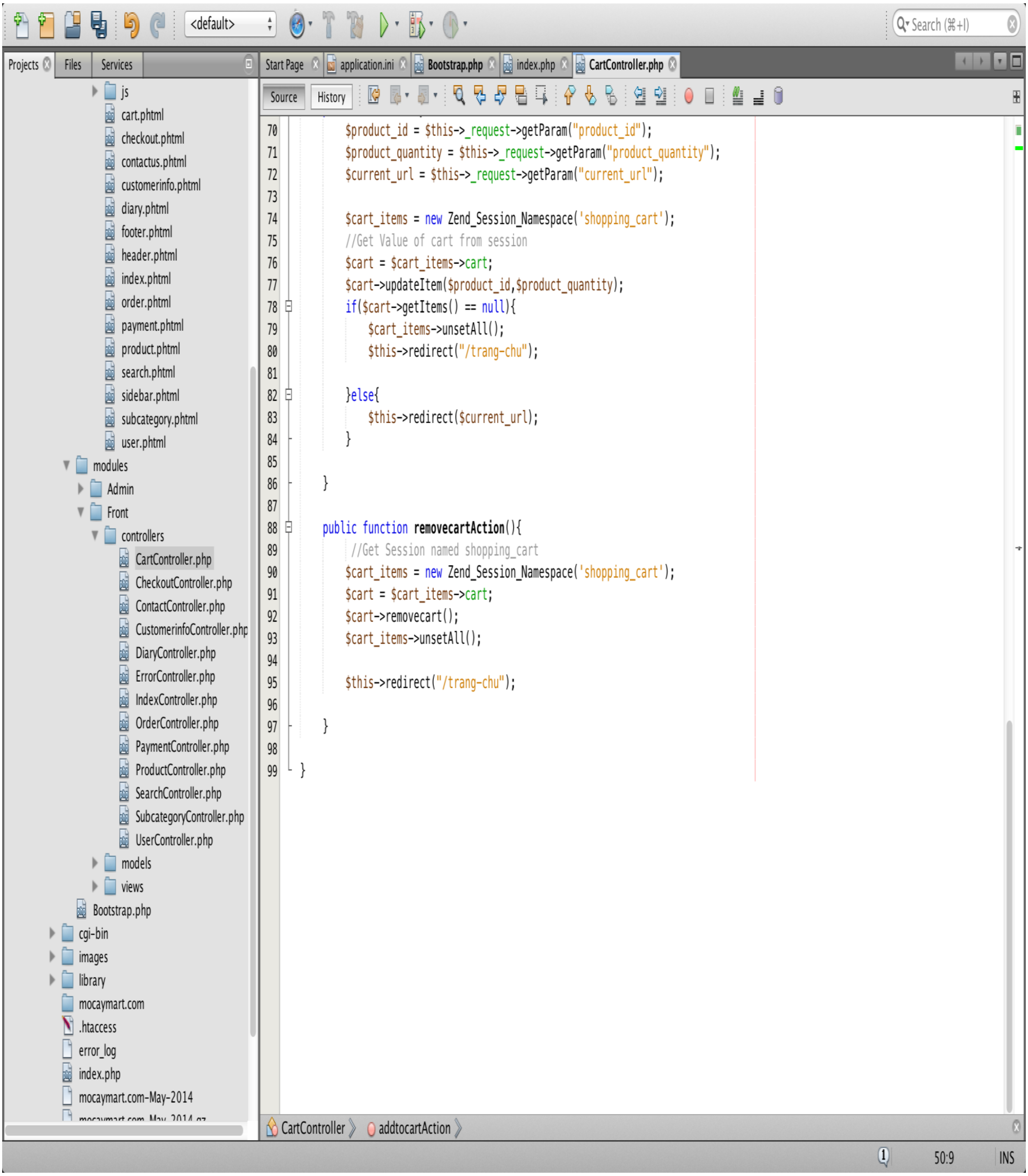
1. Could you please share the idea of your business? Do you think that www.mocaymart.com website play vital role in your business project.
2. Could you please describe the functions you wished in the website?
3. Could you please you share your idea about the graphic user interface of website?
4. Could you please describe the business process of the website such as card functions, payment process, delivery process etc.
5. Could you share the information about the budget for website development and the whole project.
6. Do you have any further development plan for this website in the future.



```
<?php
require_once 'application/modules/Front/models/subcategory.php';
require_once 'application/modules/Front/models/product.php';
require_once 'application/modules/Front/models/Cart.php';
require_once 'application/modules/Front/models/CartItem.php';
require_once 'application/modules/Front/models/customer.php';
require_once 'application/modules/Front/models/user.php';

class CartController extends Zend_Controller_Action{

    public function indexAction(){

       $cart_items = new Zend_Session_Namespace('shopping_cart');
       $user = new Zend_Session_Namespace("user");
       $customer = new Zend_Session_Namespace("customer");

       if(isset($cart_items->cart)){
            //Set layout
        $layoutPath = APPLICATION_PATH . "/layouts/front_layout";
        $option = array ('layout' => 'cart',
                    'layoutPath' => $layoutPath );
        Zend_Layout::startMvc ( $option );

        $subcategory = new subcategory();


        $cart = $cart_items->cart;


        $all_subcategory = $subcategory->getAllSubcategory();

        $this->view->all_subcategory = $all_subcategory;
        $this->view->cart = $cart;
        $this->view->customer = $customer->customer_info;
         $this->view->user = $user->user_info;
        }else {
            $this->redirect("/trang-chu");
        }


    }

    public function addtocartAction(){
        $product_id = $this->_request->getParam("product_id");
        $product_quantity = $this->_request->getParam("product_quantity");
```



```
    }

    public function addtocartAction(){
        $product_id = $this->_request->getParam("product_id");
        $product_quantity = $this->_request->getParam("product_quantity");


        //Get Session named shopping_cart
        $cart_items = new Zend_Session_Namespace('shopping_cart');
        if(!isset($cart_items->cart)){
            $cart = new Cart();
            $cart->addItem($product_id,$product_quantity);
            //Bind Cart Object into session
            $cart_items->cart = $cart;
        }else{
            //Get Value of cart from session
            $cart = $cart_items->cart;
            $cart->addItem($product_id,$product_quantity);

        }



        echo $cart->getNumberOfItems();

        exit;

    }

    public function updateitemAction(){
        $product_id = $this->_request->getParam("product_id");
        $product_quantity = $this->_request->getParam("product_quantity");
        $current_url = $this->_request->getParam("current_url");

        $cart_items = new Zend_Session_Namespace('shopping_cart');
        //Get Value of cart from session
        $cart = $cart_items->cart;
        $cart->updateItem($product_id,$product_quantity);
        if($cart->getItems() == null){
            $cart_items->unsetAll();
            $this->redirect("/trang-chu");

        }else{
            $this->redirect($current_url);
        }

```



```php
        $product_id = $this->_request->getParam("product_id");
        $product_quantity = $this->_request->getParam("product_quantity");
        $current_url = $this->_request->getParam("current_url");

        $cart_items = new Zend_Session_Namespace('shopping_cart');
        //Get Value of cart from session
        $cart = $cart_items->cart;
        $cart->updateItem($product_id,$product_quantity);
        if($cart->getItems() == null){
            $cart_items->unsetAll();
            $this->redirect("/trang-chu");

        }else{
            $this->redirect($current_url);
        }

    }

    public function removecartAction(){
         //Get Session named shopping_cart
        $cart_items = new Zend_Session_Namespace('shopping_cart');
        $cart = $cart_items->cart;
        $cart->removecart();
        $cart_items->unsetAll();

        $this->redirect("/trang-chu");

    }

}
```