Haaga-Helia
University of Applied Sciences

E-commerce web application for small business

Minh Binh Chu

Bachelor's Thesis

Degree Program in Business

Information Technology

November 2015

Haaga-Helia
University of Applied Sciences

Abstract

November 11, 2015

Author(s)
Minh Binh Chu

Degree program
Business Information Technology

Report/thesis title E-commerce web application for small business	**Number of pages and appendix pages** **49 + 12**

E-commerce and e-commerce website are among popular terms nowadays. They exist all around the world. People use them with or without awareness. From a developer point of view, study how an e-commerce website works is a compelling, and at the same time, challenging topic.

This thesis report documents the process of developing an e-commerce web application from the beginning until the end. Two main goals of the development were to create a simple e-commerce website but fully functional and to learn more about building web application with PHP and MySQL.

The background theoretical study is a part of the report. It contains useful information about e-commerce as well as the programming languages which were used to develop the e-commerce web application. Moreover, all steps that needed to build the application are explained in details.

In this project, PHP and MySQL were used to build the back-end system. HTML5, CSS3, JavaScript were used to build the front-end system. In additional, AJAX was also handful to connect the front-end with the database when necessary. Because of the aiming to study thoroughly about the code behind an e-commerce web application, this project did not use any frameworks to build the application despite their powerfulness and convenience.

In conclusion, this report serves as a guideline for developers who want to study about e-commerce web application and PHP.

Keywords
E-commerce, PHP, MySQL, HTML, CSS, PayPal

Table of contents

Terms and abbreviations

Ajax	Asynchronous JavaScript and XML
CSS	Cascading Style Sheets
DOM	Document Object Model
HTML	HyperText Markup Language
JavaScript	Object-oriented scripting language
PHP	Server Side scripting language
SQL	Structured Query Language
UI	User Interface
Back-end	The system run behind the user interface
Front-end	The user interface for the application
Hacker	A person who uses computer skills to break into computer systems without authorization
Web shop	Online shopping website
B2A	Business to Administration
B2B	Business to Business
B2C	Business to Consumer
C2A	Consumer to Administration
C2B	Consumer to Business
C2C	Consumer to Consumer

1 Introduction

This project was started from an idea of developing a free e-commerce web application. The target of this application is for small companies that want to have their own customized web shop but do not require complicated functions. This e-commerce web application should be easy to implement and maintain.

This project is also for the learning purpose. Developing an e-commerce web application from the beginning will help to learn faster and deeper about how e-commerce site works, the logic behind the code and the programming languages as well.

1.1 Goals of the project

The ultimate goal of this project is to deliver a ready e-commerce web application with basic functions for testing. The testing environment for payment transactions is PayPal testing environment – Sandbox.

The second goal that set for this project is to learn the necessary programing languages that needed to build an e-commerce web application. For this specific e-commerce web application, the chosen programming languages are PHP, MySQL, HTML5, JavaScript, and Ajax.

1.2 Scope of the project

According to the goals of this project, the e-commerce web application will be set to have the following functions:

For the shop owner:

Owner of the web shop can use the administration (admin) account to login to the admin area of the web shop on the internet browser and manage his online store by:

- Add, edit or delete categories and subcategories
- Add, edit or delete products, pictures and descriptions for each product and also the price
- View, edit or delete orders from customers
- View, edit or delete customer information

For the customer:

Customer can access to the online store on their internet browser and:

- View the products, pictures, descriptions, price
- Add products to cart
- Check out then pay the whole carts by using Paypal

- Check out then reserve the products to pick up at the store or ask the seller to send the products by post.

1.3 Out of scope

Because this is a testing e-commerce web application, no information of the products is actually valid. All product's details, prices, pictures are only for testing purposes.

Matters related to high level security will not be included in this project. For example, how to protect customer information that stores in the database or security protocol to cope with hacker's attacks. This product will use PayPal service to cover the payment transactions. According to that, the security about money transaction is based on PayPal security level.

Other online banking solutions other than PayPal will not be included in this project.

2 Background theory

This chapter consists of the background theories about e-commerce concept, its definition, history and the future trend of e-commerce. The short explanation about HTML, CSS, JavaScript, PHP, MySQL and PayPal will be included in this chapter.

2.1 E-commerce

For many years, when people want to buy something, the only option for them is going to a store or shopping mall. For sellers, when they want to sell their goods, usually they have to find a place to locate their store first, then load it with the products they want to sell and wait for customers. Everything is manually. However, they all changed when the internet appeared and now become the indispensable part of the world. Nowadays, people do not have to set a foot outside but still get the products they want. The seller will not have to worry about locating their store, finding good place while their sales continue to growth annually. These changes and convenience come from a simple word – “e-commerce”.

2.1.1 Definition

“E-commerce is about digitally enabled commercial transactions between and among organizations and individuals (Laudon & Laudon 2012, 399).”

E-commerce nowadays can be divided into six categories:

Business to Business (B2B)
This is the biggest form of e-commerce with both sellers and buyers are businesses. In this type of e-commerce, there is no individual consumer involved. Because this is between companies and companies, the transactions could value billions dollars. (Khurana 2015.) The figure below shows an impressive amount of money was circulated within the e-commerce system in America.

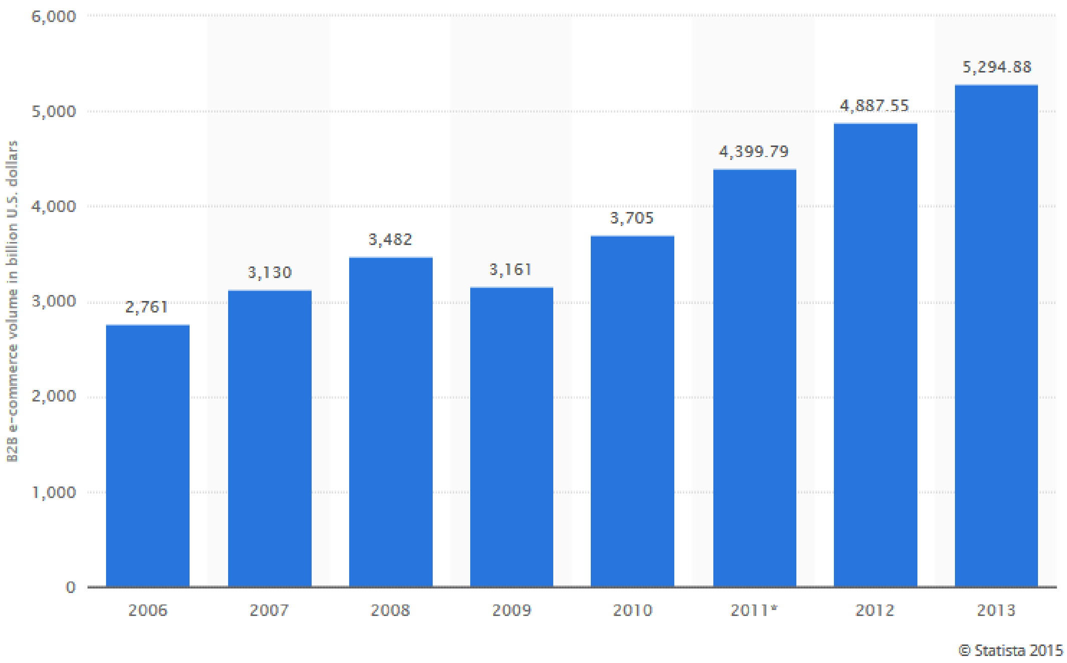


Figure 1. B2B e-commerce volume in the United States from 2006 to 2013 (Statista 2015)

Business to Consumer (B2C)

This is an e-commerce type about online businesses sell products to final consumers. It is related to traditional retail. However, consumers usually have more information available for the products and there is also a widespread idea that the price is cheaper than the traditional retail.

Consumer to Consumer (C2C)

C2C is about all electronic transactions between consumers and consumers. In general, these transactions are carried out on the online platform which is provided by the third party.

Consumer to Business (C2B)

This type of e-commerce is the reverse of traditional commerce. A large number of individuals put their ready services or products on a platform. These services or products are available for purchase by companies that are seeking the exact types of services or products on that platform. For example, a photographer put all of his photos on an online image store. One of these photos was sold to a company which was looking for their website's background image. Another one was bought by a company for their demonstration event.

Business to Administration (B2A)

This type of e-commerce is about all online transactions between companies and public administration. It covers quite large areas such as fiscal, social security, employment, legal documents and register, etc.

Consumer to Administration (C2A)

C2A e-commerce encompasses trading between final consumers and public administration. For example, it could happen when a person orders a health service from a public health care center and pays for it through an online portal.

(Fernandes 2014.)

2.1.2 History

E-commerce has gone a long way to gain its popularity today. These following paragraph will summarize this road into few milestones.

In 1960s, the history of e-commerce began. Electronic Data Interchange (EDI) was introduced at that time and soon became popular. EDI helps to transfer data (include invoices, bills, documents, etc.) between companies' computer systems regardless theirs geography locations. EDI works based on standardizing message format without the need for human intervention. (Rouse 2015.)

In 1969, ARPANET was established and served as a testing base for new networking technologies. ARPANET was used to connect many universities and research centers. (Beal 2015.)

In 1979, the term "teleshopping" was invented by Michael Eldrich - an English inventor and entrepreneur. It means shopping from distance. His system connected a modified television with a computer via the domestic telephone line. (Maroon 2015.)

In 1981, the first business to business online shopping system called Thomson Holidays UK was installed (Benvenuti 2014).

In April 1984, the first comprehensive electronic commerce service was launched in America and Canada. It was called Electronic Mall by CompuServe. (Benvenuti 2014.)

In 1990, the first World Wide Web server was created by Tim Berners Lee. He successfully connected HTTP (Hypertext Transfer Protocol) client and server via the internet. (Maroon 2015.)

In 1994, SSL encrypted appeared with the first third-party payment service for online credit card. Amazon, one of the biggest online store, was also established this year. (Maroon 2015.)

In 1995, Dell and Cisco began to implement commercial transactions widely via the internet. EBay was established in the same year. (Benvenuti 2014.)

From 1995 to 2004 was the golden time of online stores. Many famous e-commerce businesses were established during that time: Yahoo – 1995, Google – 1998, PayPal – 1998, Alibaba – 1999 (Alibaba Group 2014; Engine Directory 2015; Google Company 2014; PayPal About 2015).

In 2004, Payment Card Industry Data Security Standard (PCI DSS) was formed (Maroon 2015).

From 2004 to 2009 was the expansion period for e-commerce with the appearance of mobile commerce.

In 2009, Bitcoin appeared. It was created by Satoshi Nakamoto. However, Satoshi Nakamoto is just a nick name. Until now, the real identity of the Bitcoin's creator is still a mystery. (CNN Money 2014.)

> Bitcoin is a form of digital currency, created and held electronically. No one controls it. Bitcoins aren't printed, like dollars or euros – they're produced by people, and increasingly businesses, running computers all around the world, using software that solves mathematical problems. (CoinDesk 2015.) Bitcoin gained in popularity when its price jumped from around 20.41 U.S. dollars in January 2013 to approximately 1124.76 U.S. dollars in November 2013 (figure 2) (Statista 2015).

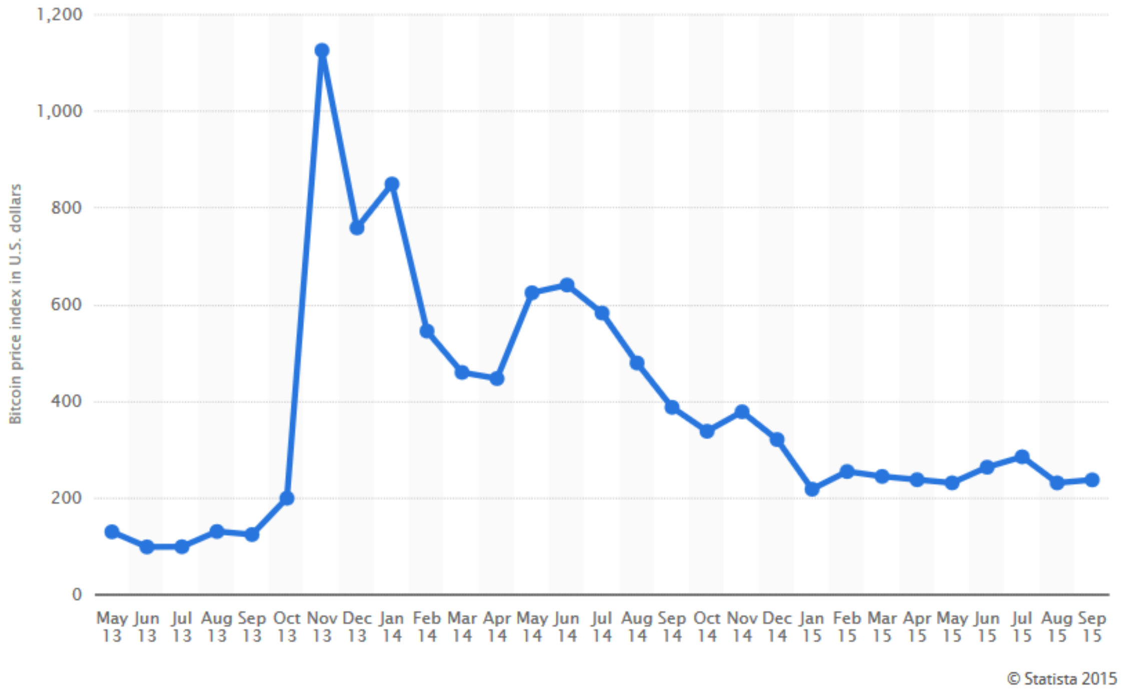


Figure 2. Bitcoin price index from May 2013 to September 2015 (in U.S. dollars) (Statista 2015)

From 2009 to present, e-commerce has been growing incessantly and this trend is forecasted to continue in the future (figure 3).

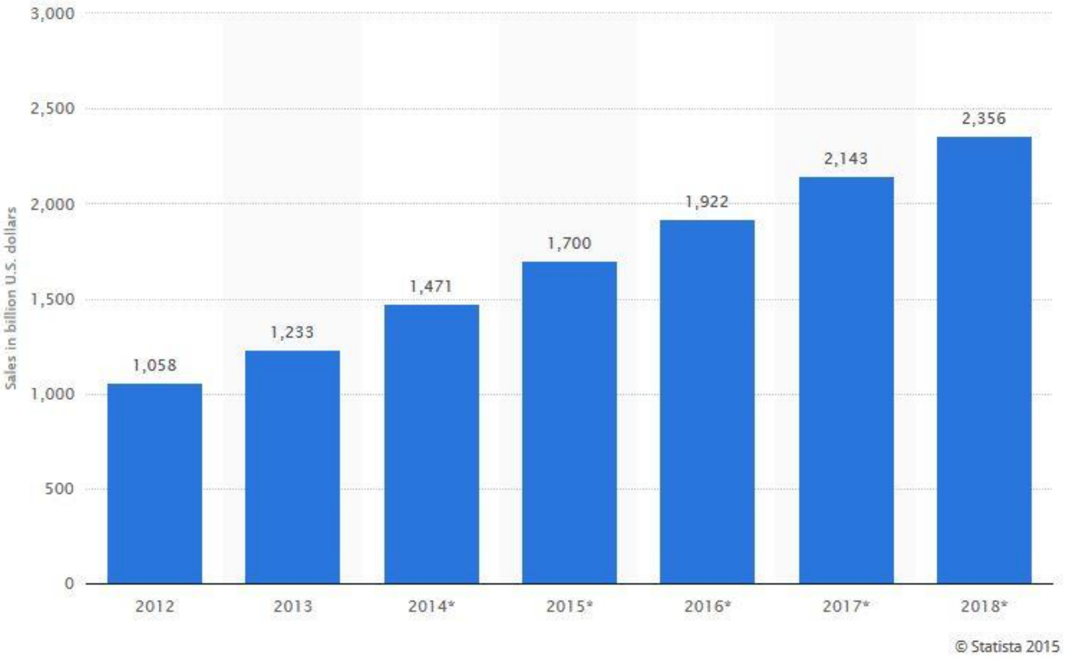


Figure 3. B2C e-commerce sales worldwide from 2012 to 2018 (in billion U.S. dollars) (Statista 2015)

2.1.3 Advantages and risks of e-commerce

Advantages

E-commerce has brought to both businesses and customers many practical benefits.

Reducing cost is one of the most obvious benefits that e-commerce brings to businesses (Felke-Morris 2013, 508). Without having to build a brick store, install shelves or pay the rent for the storage, a company can save a huge amount of money. Building a web shop online can cost money as well; however, compare to a real brick store, the cost of building a web shop is much cheaper. Besides, there are services on the internet that provide a simple web shop for free. With e-commerce, opening a store is now easier than ever.

E-commerce helps shop owners to manage the store as well as the store's data more effectively (Felke-Morris 2013, 508). Depending on the comprehensive level of the e-commerce site, shop owners can easily manage the information of products, categories and customer's data. When customers buy products via the web shop, their information will be automatically added to the system's database. This helps the store to collect customer information more effectively.

A good e-commerce site could increase the customer satisfaction by enhancing the interactive between customers and the store. A web shop can provide the customer with many assistant services. Telling examples are the FAQs page to explain about frequently asked questions or the live chat section to assist customers instantly. (Felke-Morris 2013, 508.) Moreover, based on the customers' information through orders that they have purchased or have searched, the system could even forecast the trend of consuming in the near future. A very good example about this is eBay. For years, they have been doing the predicting consuming trend very well.

To consumers, e-commerce brings the convenience to them (Felke-Morris 2013, 508). They can go shopping 24 hours per day, 7 days per week. Besides, there is almost no geographical boundaries apply to e-commerce. Aiding by the internet, now consumers can shop at any online stores they want, even when they are living in Vietnam and the shop is located in America. This is not only beneficial for consumers but also for businesses. A store that opens all the time and attracts not only the local consumers but also the international ones surely is better than a store with limited opening hours and narrow customers' types.

Last but not least, with e-commerce, consumers can always easily compare the prices and the diversity of products between stores. With one browser and the internet, they can get access to many stores at the same time and check the differences quickly. Based on these comparisons, consumers can choose the most suitable products for their preferences. (Felke-Morris 2013, 508.)

Risks of e-commerce

Despites of many advantages as mentioned above, there are also risks existing along with e-commerce.

Because e-commerce is working based on a technical system, when the system is not received appropriate care, it could lead to the failure of the whole e-commerce site. Both the front-end system and the back-end system have to operate well and corporate smoothly with each other. A web shop with a powerful back-end system but has a poor, unfriendly user interface (UI) will have many difficulties in attracting customers to their store. On the contrary, a good looking online store could also lose their customers if they cannot make their payment easily. In other words, technical issue has always been a critical one that decides the success of an e-commerce site.

Another risk that should be concerned about e-commerce is security (Felke-Morris 2013, 509). Unlike a brick store, a web shop must be prepared to cope with buglers from all over the world. These buglers have another name as hackers. A web shop has to have the sufficient security scheme to protect its system as well as the database. It should has a backup plan to deal with any emergency cases and to keep the system running stable.

The third risk is about privacy (Felke-Morris 2013, 509). When using e-commerce system, consumers have to provide their private information in order to purchase products. More specifically, they will have to provide their name, email, phone number, even credit card information and possibly their address if they want the product to be delivered to their home. These are very sensitive information and related to both security and privacy issues. How will these information be protected and how will these be used? Their information on the database could be used for many purposes without them even notice; it could be used internally for marketing or could be used externally for selling (Felke-Morris 2013, 509). It can even be worse if the information is stolen by hackers and is used for illegal purposes. Consumers should never underestimate this issue.

Shopping on a virtual store means products showed to consumers are also virtual. Consumers will have to make decision based on the photos, the descriptions that provided by the store. When the real product arrives, it could be much different than what they expected. It is the risk that consumers have to face when using the online shopping service. (Felke-Morris 2013, 509.)

2.2 PayPal

PayPal is the name of an online payment service provided by PayPal Holdings, Inc. This company was founded in 1998 in America. In 2002, PayPal was acquired by eBay with a contract worth of 1.5 billion dollars. (Faculty 2015.) In late 2015, PayPal is set to part from eBay as a separate publicly traded company (Statista 2015).

PayPal uses data encryption and anti-fraud technology to make online transactions through its gateway safe and secure. It works as a safe bridge between buyers and sellers. For buyers, they need to link their credit card(s) or bank account(s) with their PayPal account. Later, when they want to buy products on a web shop which accepts PayPal as a payment method, they can pay for the products through PayPal gateway without having to enter their credit card information again. It helps to reduce the risk of revealing the sensitive banking information. The money will be transferred to the PayPal account of the seller then he can transfer this amount of money to his own bank account later if he wants. (Grabianowski & Crawford 2015.) (Figure 4.)

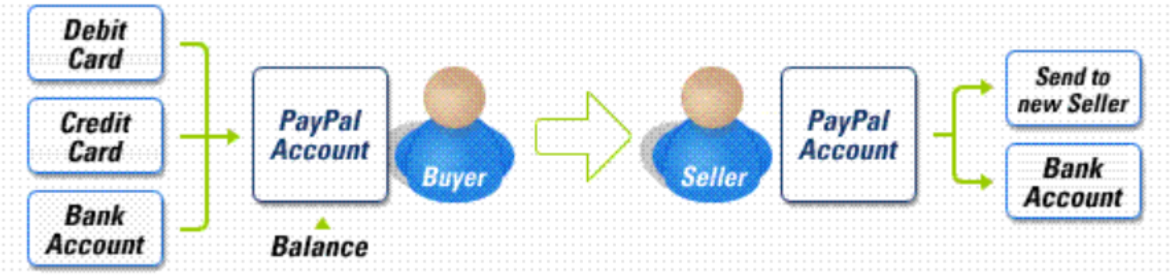


Figure 4. How PayPal works (Faculty 2015)

PayPal provides the following services:

- Send and receive money
- Pay with PayPal balance, bank account, credit card, or debit card
- Shop online

There are four types of PayPal account:

- Personal account
- Premier account
- Business account
- Student account

Personal account is for people who want to shop with PayPal or send money to friends, family as gifts.

Premier account is suitable for part-time business people. It allows to receive money into PayPal account.

Business account is for company or merchant group. It has all the features of premier account and some additional features to help company manage its account better. The business account offers the following services:

- Operate account under company or business name
- Accept debit card, credit card, and bank account payments
- Allow up to 200 employees limited access to business account
- Sign up for PayPal products that meet your business needs
- Set up a separate email address for customer service

Student account is for teenagers from the age 13. This type of account is suitable for parents who want to control their children's account. This account is linked to one of the three accounts above. The parents can track all transactions of the student accounts that linked to their PayPal account. They can create four student accounts under each of their PayPal account.

In 2015, 169 million people use PayPal in 190 countries and in 21 different currencies. The total payment volume of PayPal in 2014 reached 234.64 billion U.S. dollars (figure 5). The revenue was 8.03 billion U.S. dollars with a net income of 419 million U.S. dollars.

(PayPal 2015.)

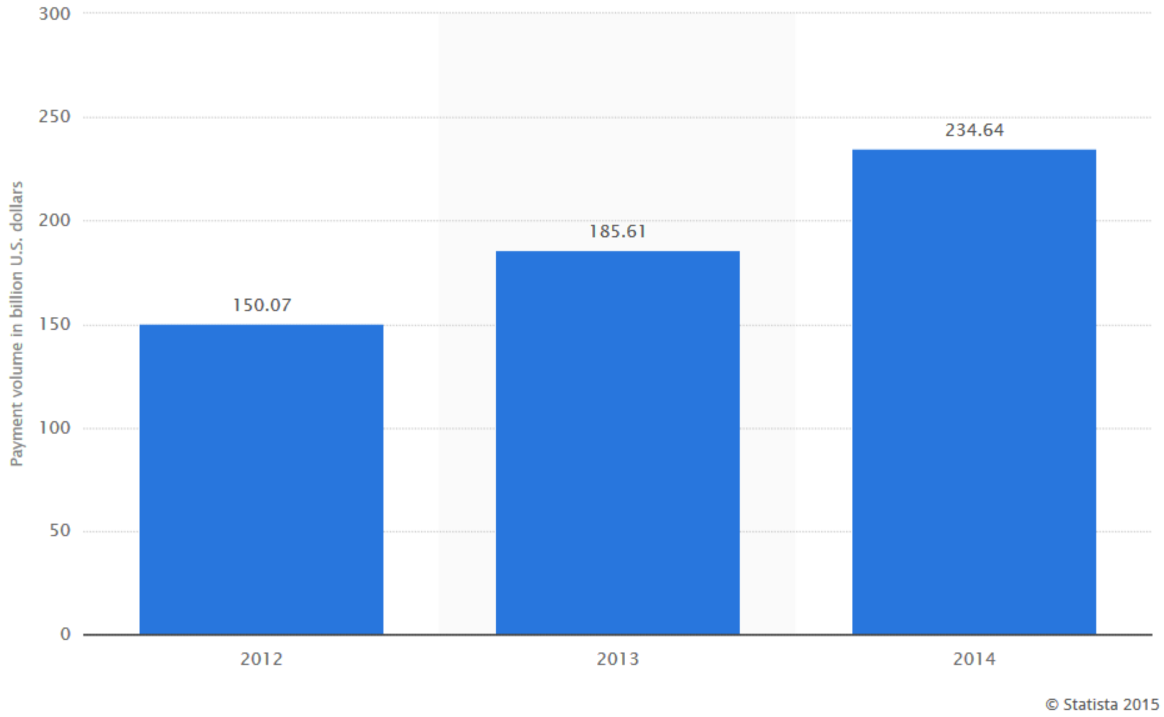


Figure 5. PayPal's annual payment volume from 2012 to 2014 (in billion U.S. dollars) (Statista 2015)

2.3 HTML, XHTML and HTML5

HTML stands for HyperText Markup Language. Markup languages are responsible for directing the browser software on how to display web documents (web pages). The mark-ups are tags, which usually consists of opening and closing tags to display graphics, formatting text. (Richardson & Thies 2013, 21.) Each individual mark-up is also referred as element (Rouse 2015).

HTML is developed by Tim Berners-Lee in 1991, based on SGML (Standard Generalized Markup Language). SGLM is not a document language. It describes how to create a document type definition (DTD). (Richardson & Thies 2013, 21.) HTML standards are created by the World Wide Web Consortium (W3C 2015).

XHTML stands for EXtensible HyperText Markup Language. It is almost the same as HTML, but stricter. In XHTML, every mark-up or tags must be written following the syntax correctly. It was introduced in 2000. (W3C 2015.)

HTML5 is the latest version of HTML. It was first introduced in 2008 and the final recommendation was released on the 28th of October 2014. It has many new features compare to HTML4. For examples: (W3C 2015.)

- New semantic elements like <header>, <footer>, <article>, and <section>

- New form control attributes like number, date, time, calendar, and range
- New graphic elements: <svg> and <canvas>
- New multimedia elements: <audio> and <video>

Because of its new advance features, HTML5 is becoming more and more popular (figure 6).

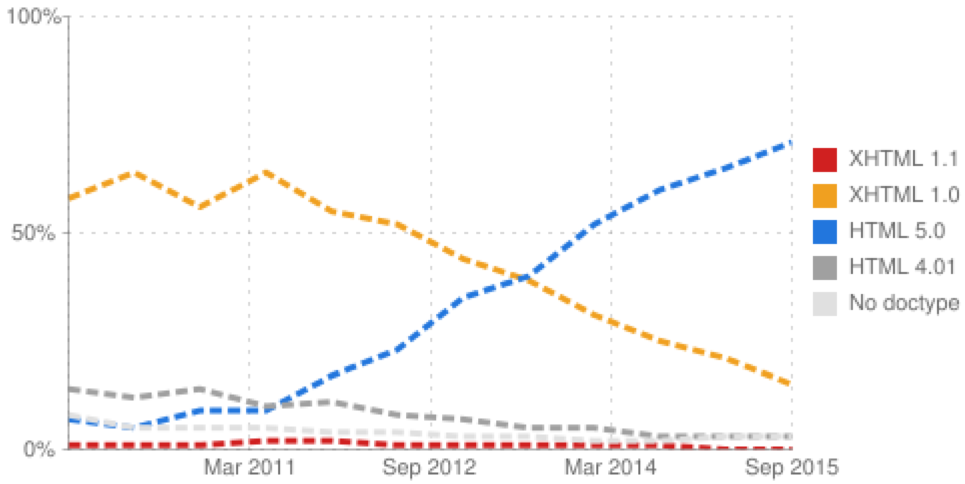
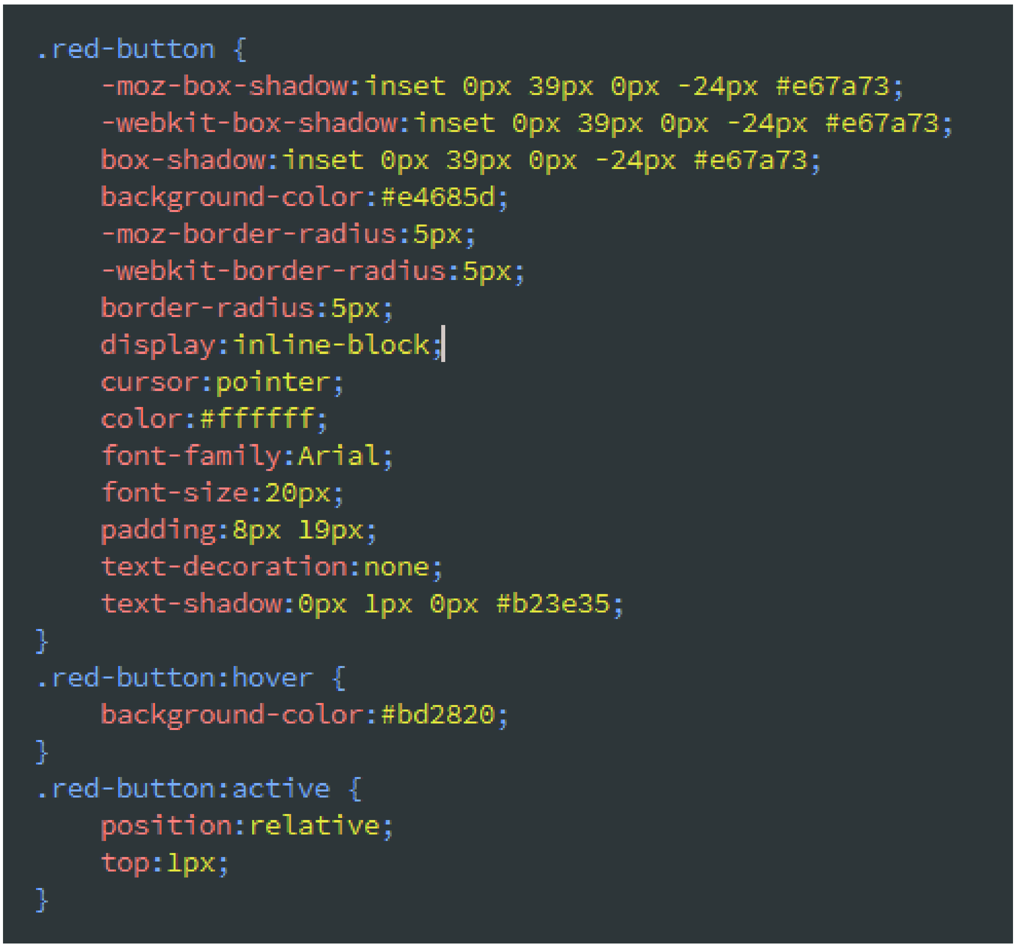


Figure 6. HTML Version Trends (PowerMapper Software 2015)

2.4 CSS

CSS stands for Cascading Style Sheets. It is a language which describes how elements of web page should be presented. CSS helps web page to display better on different size screens. Because CSS can be stored in a different document than HTML document so a CSS file could be used to manage multiple HTML documents. It helps to save time and effort managing web pages. (W3C 2015.)

The following figures illustrate an example of using CSS to modify HTML element. The first figure shows the CSS code for a button element of HTML.



```css
.red-button {
    -moz-box-shadow:inset 0px 39px 0px -24px #e67a73;
    -webkit-box-shadow:inset 0px 39px 0px -24px #e67a73;
    box-shadow:inset 0px 39px 0px -24px #e67a73;
    background-color:#e4685d;
    -moz-border-radius:5px;
    -webkit-border-radius:5px;
    border-radius:5px;
    display:inline-block;
    cursor:pointer;
    color:#ffffff;
    font-family:Arial;
    font-size:20px;
    padding:8px 19px;
    text-decoration:none;
    text-shadow:0px 1px 0px #b23e35;
}
.red-button:hover {
    background-color:#bd2820;
}
.red-button:active {
    position:relative;
    top:1px;
}
```

Figure 7. CSS code for a button (Chu 2015)

When inserting a button in HTML without any CSS style, it will look simple (figure 8).

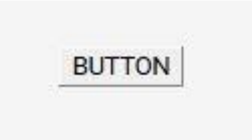


Figure 8. A button without applying the CSS style (Chu 2015)

When the button above had been applied with the CSS code that was shown on the figure 7, its appearance became more appealing and had the hover effect when pointed by the mouse (figure 9).

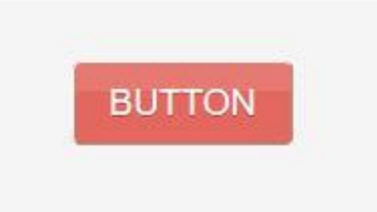


Figure 9. A button was applied with the CSS style (Chu 2015)

2.5 JavaScript, JQuery and AJAX

JavaScript

JavaScript (JS) is a cross-platform, object-oriented scripting language (MDN 2015). It was created in 10 days in May 1995 by Brendan Eich. Its original name was Mocha then was changed to JavaScript in December 1995. (W3C 2015.)

JS can be connected to the objects of its host environment and provides control over them. JS is used in web development to make web pages become more dynamic. JS can control a browser and manipulate the browser's Document Object Model (DOM). JS can also allow an application to communicate with the database. (MDN 2015.)

JQuery

JQuery is a fast, small, and feature-rich JavaScript library. JQuery was started from the suggestion of John Resig in 2005 and then it has grown into the most popular library for JS. (The jQuery Foundation 2015.) It was created with the purpose of making using JS become easier. JQuery simplified many complicated things from JS such as manipulating DOM or AJAX. (Jquery-tutorial.net 2015.)

AJAX (Fote 2013.)

AJAX stands for Asynchronous JavaScript and XML. AJAX is a client-side script which communicates web page and web server/database. It helps web pages interact with server side dynamically via JavaScript.

AJAX allows web pages to make asynchronous calls and callbacks to web server to retrieve or save data to the database. Because of that, the client web browser becomes faster, more responsive and more user-friendly.

2.6 PHP

PHP is a recursive acronym that stands for Hypertext Pre-processor (Richardson & Thies 2013, 201). It was a simple set of Common Gateway Interface (CGI) binaries written in the C programming language by Rasmus Lerdorf in 1994 (The PHP Group 2015).

PHP is one of the most popular server-side languages for web site development because of its flexible and easy-to-use features. PHP code can be placed anywhere in an HTML document. It allows to establish the framework and structure of the page as well as data processing. PHP is used to develop the CMS for web pages because with PHP, they can control the dynamic of the content. (Richardson & Thies 2013, 202.)

Figure 10 shows the popular of PHP programming language in comparison to other server-side programming languages.

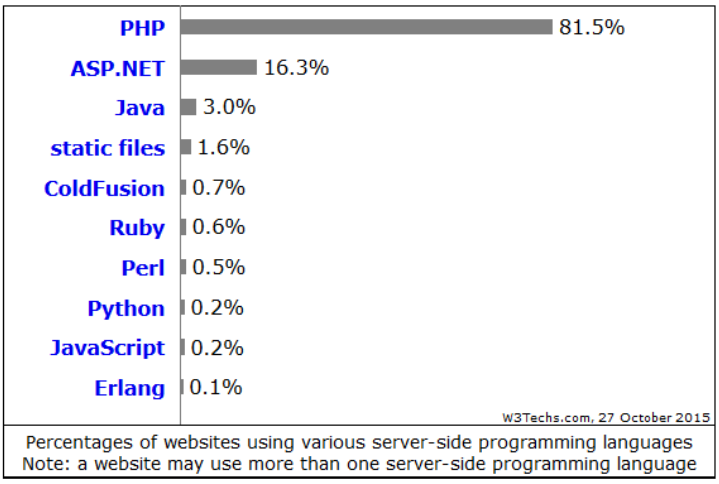


Figure 10. Usage of server-side programming languages for websites (W3Techs 2015)

More specifically, the work flow of PHP is quite simple. When a client's web browser sends a request to the server, the server will pass this request to PHP. PHP will handle the request then generate the corresponded HTML file. This HTML file will be passed back to the server then the server will send this back to the client's web browser. In the processing of PHP, PHP can interact with the database system, the file system or mail server, etc. (Figure 11.)

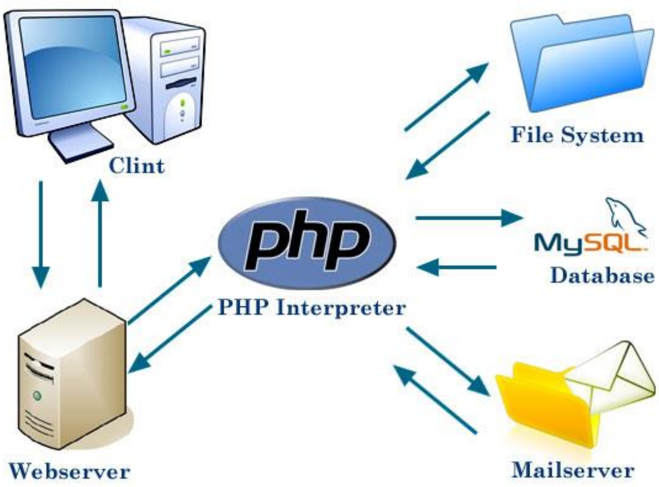


Figure 11. How PHP works (Ejobindia 2014)

2.7 MySQL

SQL (Structured Query Language) is the standard language to manage the data in the relational database management system (RDMS) (QuinStreet Inc 2015).

In the recent years, there are many new database software have appeared in the database market; for example, MongoDB, CouchDB, PostgreSQL, MariaDB. However, MySQL is still the most popular one of 58 per cent of the database market share in April 2014 (figure12). It is an open source database software that can integrate databases with the power of SQL. Any hosting server with MySQL installed will enable the web application to create and manage the database by the server-side programing languages such as PHP. (Richardson & Thies 2013, 207.)

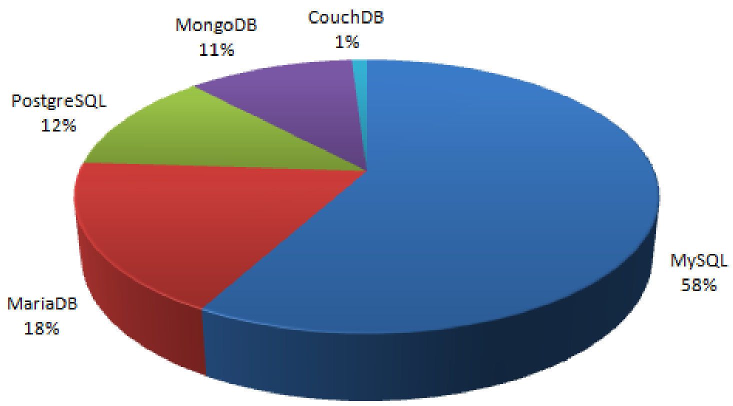


Figure 12. Database market share, April 2014. (Jelastic 2014)

2.8 System development lifecycle

To develop an application, there are steps that developers or project team must follow. First of all is defining the purposes of the application, which answers the question ‘what is this application for’. The following step is finding out the requirements, what is needed to build the application. When the requirements are listed, the project team starts to design a plan of how it is to be implemented. This process is called system analysis and design (SA&D).

In the next step, these designs are implemented by the programming process which produces the actual programming code. Finally, the application will be tested to check to see whether it actually works or not.

The stages that an application have to go through in the transformation from idea into actual product are called system development lifecycle (SDLC). The simplicity and popularity of SDLC is the waterfall lifecycle or waterfall methodology. This method divides the project into sequential stages. It means one stage must be completed before moving to the next one.

(Whiteley 2013, 205.)

In general, the waterfall lifecycle consist of the following stages (figure 13): (TechnologyUK 2015)

- **Requirement analysis:** find out what is needed to develop an application or system
- **System design:** make a plan of how the system is going to be built
- **Implementation:** implement the plan by writing the application with computer programming languages
- **Testing:** check if the system is working, then find out the possible faults and failures and fix them. There are two types of testing, system test and acceptance test. The system test is a technical test by the development team. The acceptance test is deployed by users.
- **Deployment:** After all the tests are done, the final product is released into the market or gone live on the internet.
- **Maintenance:** to fix the problems which occur during the live working process of the products and enhance the products quality with better version

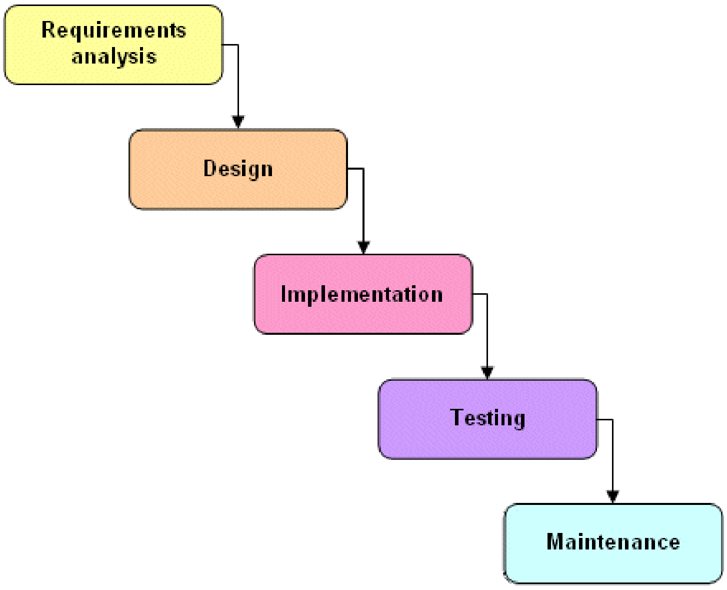


Figure 13. Waterfall lifecycle model (TechnologyUK 2015)

3 E-commerce web application project

This chapter will present thoroughly about the plan of the development process and the actual development process. It will describe how the e-commerce website was built compare to the plan.

3.1 Development plan

Since the goals and scope had been clearly defined for this project, the Waterfall methodology was chosen to apply to the development process.

The development process of this project consists of the following periods:

a) Analyze and define the requirements for a basic web shop
b) Design the architecture for the back-end and front-end systems based on the requirements
c) Build the back-end system and then the front-end system for the web shop
d) Test the web shop with some use cases

Because of the learning objective, no framework both for front-end and back-end system will be applied. Building the application from the scratch with pure code is the best efficient way to learn a programing language and see how the code actually works.

3.2 Development process

The development process starts by defining the requirements for the web shop and then a blue print for the e-commerce web application will be made.

3.2.1 Requirements

For a basic e-commerce web application, it must has at least these five components:

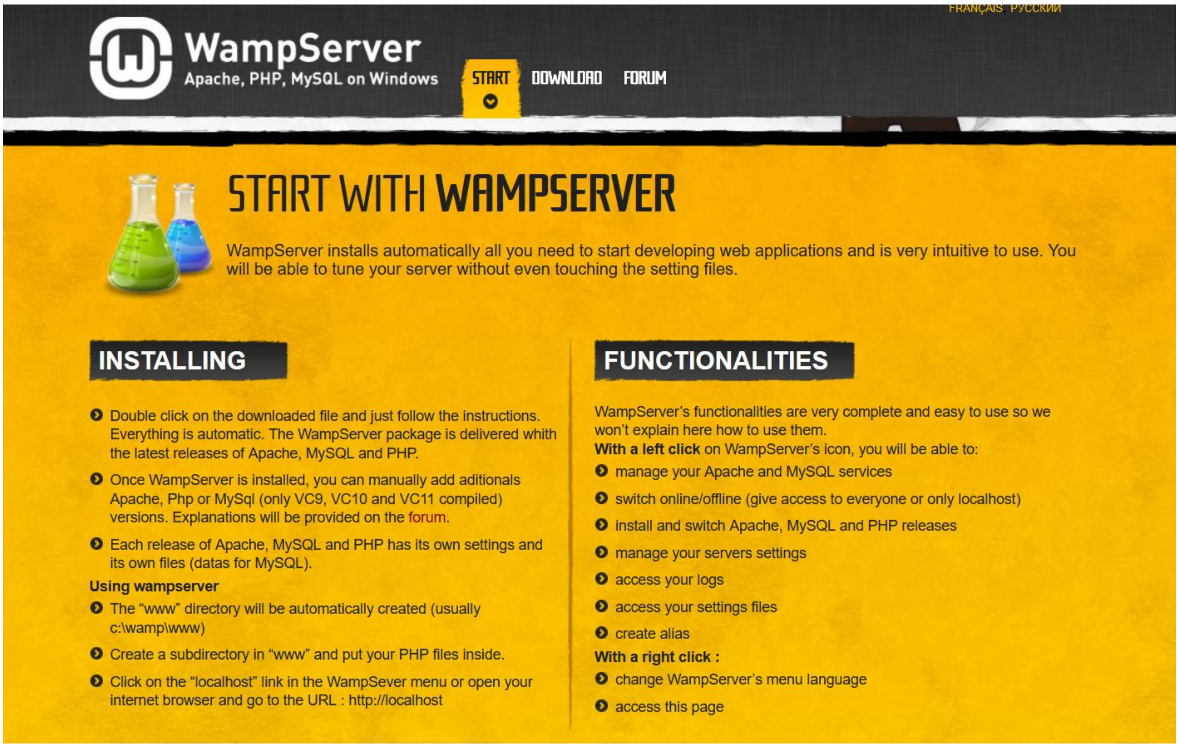
a) A database system to store the data of products, customers, orders, etc.
b) A control center which helps the shop owner to manage all the data of the store
c) A friendly demonstration site to display all the products and its related information (description, price, etc.)
d) A virtual shopping cart for customers to add the chosen products from the demonstration site
e) A payment system that handles the transactions between customers and the web shop

3.2.2 Set up the development environment

In the first period of the project, the e-commerce site was developed and tested on a local host. MySQL, PHP and Apache (a free web server) must be installed in order to run the web shop locally. To reduce time and effort to set up a local host for websites, WampServer (WAMP) was chosen to install on the local computer.

> WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database. (WampServer 2015.)

To install WAMP, it is needed to access the homepage of WAMP and follow the instruction that was clearly written on the home page (figure 14). The code editor that was Dreamweaver and Notepad++



START WITH WAMPSERVER

WampServer installs automatically all you need to start developing web applications and is very intuitive to use. You will be able to tune your server without even touching the setting files.

INSTALLING

- Double click on the downloaded file and just follow the instructions. Everything is automatic. The WampServer package is delivered whith the latest releases of Apache, MySQL and PHP.
- Once WampServer is installed, you can manually add aditionals Apache, Php or MySql (only VC9, VC10 and VC11 compiled) versions. Explanations will be provided on the forum.
- Each release of Apache, MySQL and PHP has its own settings and its own files (datas for MySQL).

Using wampserver

- The "www" directory will be automatically created (usually c:\wamp\www)
- Create a subdirectory in "www" and put your PHP files inside.
- Click on the "localhost" link in the WampSever menu or open your internet browser and go to the URL : http://localhost

FUNCTIONALITIES

WampServer's functionalities are very complete and easy to use so we won't explain here how to use them.

With a left click on WampServer's icon, you will be able to:

- manage your Apache and MySQL services
- switch online/offline (give access to everyone or only localhost)
- install and switch Apache, MySQL and PHP releases
- manage your servers settings
- access your logs
- access your settings files
- create alias

With a right click :

- change WampServer's menu language
- access this page

Figure 14. WampServer's home page (WampServer 2015)

3.2.3 Database design and implementation

The name of the database (DB) for the project is **ecommerce_DB.** In this database, there are seven tables: admin, category, sub_category, products, customers, orders, and ordered_products. (Figure 15.) More details can be viewed from the appendix 1.

- Table **admin** contains of the usernames and passwords for the administration accounts. In the development of this project, only one administration account was created and it acts as the shop owner to manage the store.
- Table **category** consists of the list of main categories for products.
- Table **sub_category** consists of the list of narrowed categories for products. Each sub_category belongs to a specific category.
- Table **products** contains of all necessary information about each product.
- Table **customers** contains of the information of customers when they order or buy the products from the web shop.
- Table **orders** is for storing all the orders made by customers or placed manually by the shop owner.
- Table **ordered_products** contains of the information about products ordered by customers.

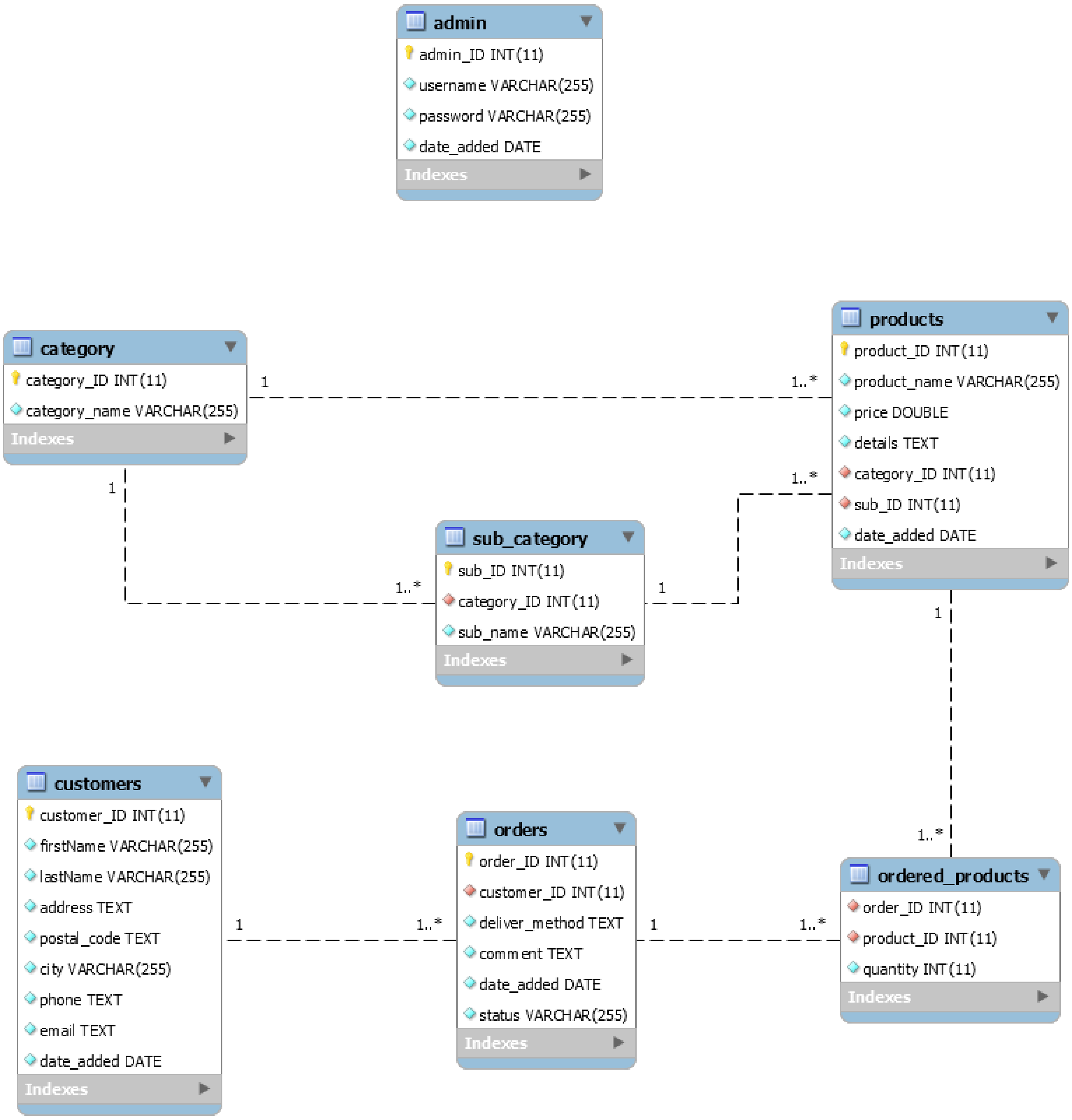


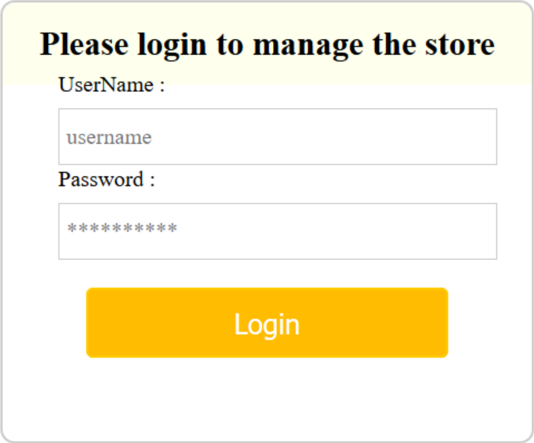
Figure 15. UML diagram of the ecommerce_DB (Chu 2015)

3.2.4 Store management system

When using the e-commerce web application, the shop owner/user will firstly need to login as the administration account (admin) to continue.

Login system for the admin

The login system will ask the user to provide the username and password in order to get access to the store management system (figure 16).



Please login to manage the store

UserName :

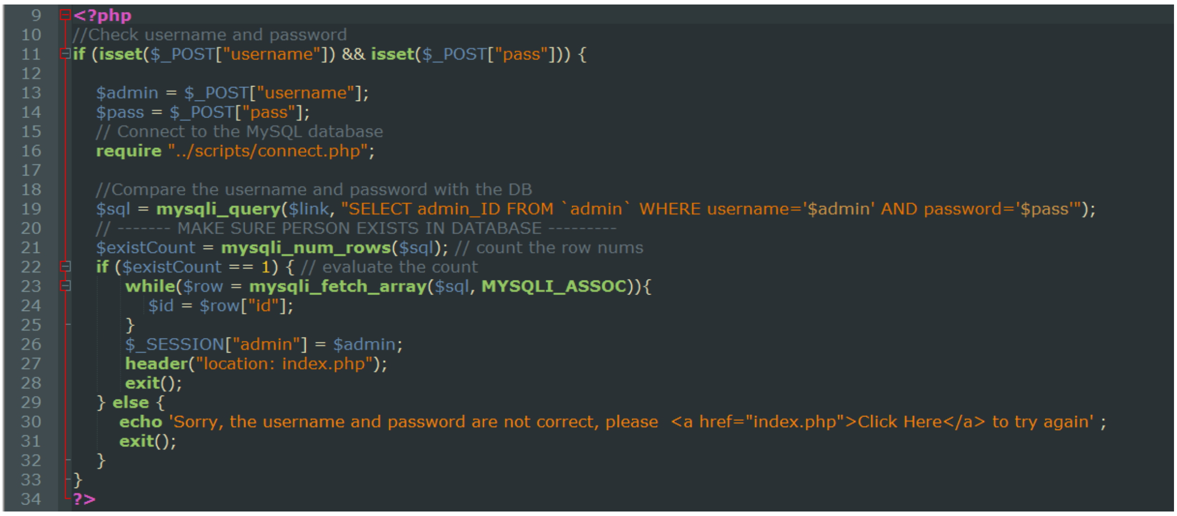
username

Password :

****

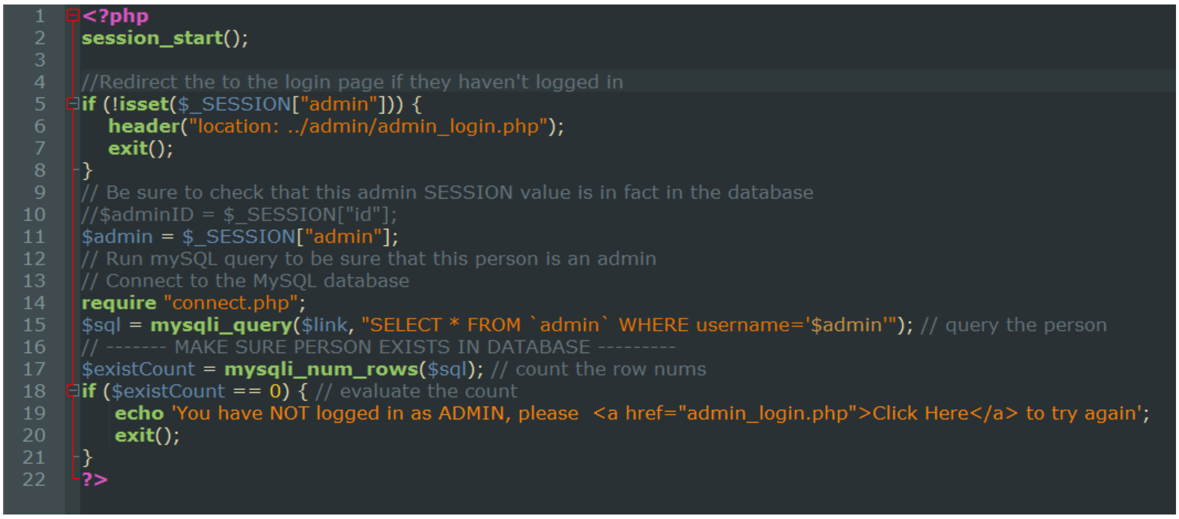
Figure 16. Login screen shows up when accessing to the store management system (Chu 2015)

After providing the username, password and pressing “login”, the login system will compare these information with the username and password in the database. If they are correct, it will create a new session called “admin” then redirect the user to the store management system’s front page. Otherwise, it will ask the user to re-enter the username and password. Every other pages of the store management system will have the PHP code that checks if the session “admin” exists when the user tries to access into these pages. (Figure 17 & 18.)



```php
<?php
//Check username and password
if (isset($_POST["username"]) && isset($_POST["pass"])) {

    $admin = $_POST["username"];
    $pass = $_POST["pass"];
    // Connect to the MySQL database
    require "../scripts/connect.php";

    //Compare the username and password with the DB
    $sql = mysqli_query($link, "SELECT admin_ID FROM `admin` WHERE username='$admin' AND password='$pass'");
    // ------- MAKE SURE PERSON EXISTS IN DATABASE ---------
    $existCount = mysqli_num_rows($sql); // count the row nums
    if ($existCount == 1) { // evaluate the count
        while($row = mysqli_fetch_array($sql, MYSQLI_ASSOC)){
            $id = $row["id"];
        }
        $_SESSION["admin"] = $admin;
        header("location: index.php");
        exit();
    } else {
       echo 'Sorry, the username and password are not correct, please  <a href="index.php">Click Here</a> to try again' ;
       exit();
    }
}
?>
```

Figure 17. PHP code for login system (Chu 2015)



```php
<?php
session_start();

//Redirect the to the login page if they haven't logged in
if (!isset($_SESSION["admin"])) {
    header("location: ../admin/admin_login.php");
    exit();
}
// Be sure to check that this admin SESSION value is in fact in the database
//$adminID = $_SESSION["id"];
$admin = $_SESSION["admin"];
// Run mySQL query to be sure that this person is an admin
// Connect to the MySQL database
require "connect.php";
$sql = mysqli_query($link, "SELECT * FROM `admin` WHERE username='$admin'"); // query the person
// ------- MAKE SURE PERSON EXISTS IN DATABASE ---------
$existCount = mysqli_num_rows($sql); // count the row nums
if ($existCount == 0) { // evaluate the count
     echo 'You have NOT logged in as ADMIN, please  <a href="admin_login.php">Click Here</a> to try again';
     exit();
}
?>
```

Figure 18. PHP code to check user’s session (Chu 2015)

User interface for the store management system

After logging in, the user will be automatically redirected to the front page of the store management system. Herein, the user can decide what to do next with the store.
The user interface of the store management system was modified from the default template provided by Dreamweaver. (Figure 19.)

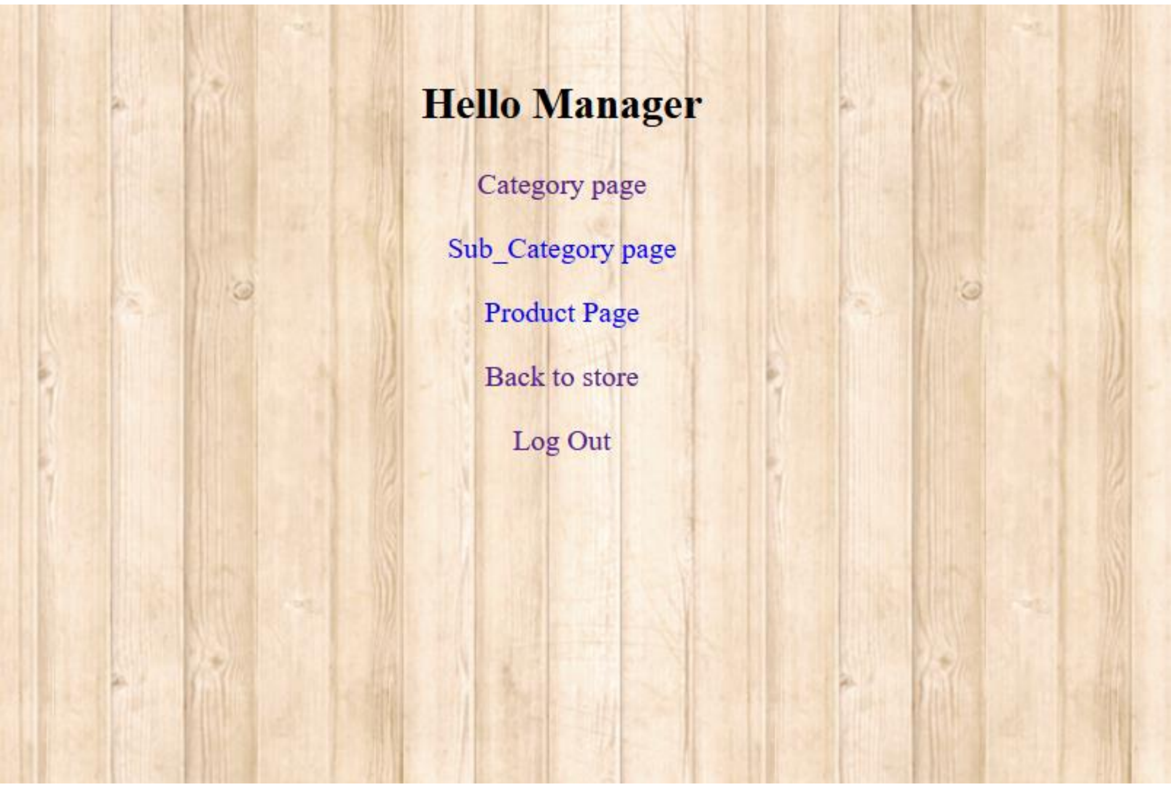
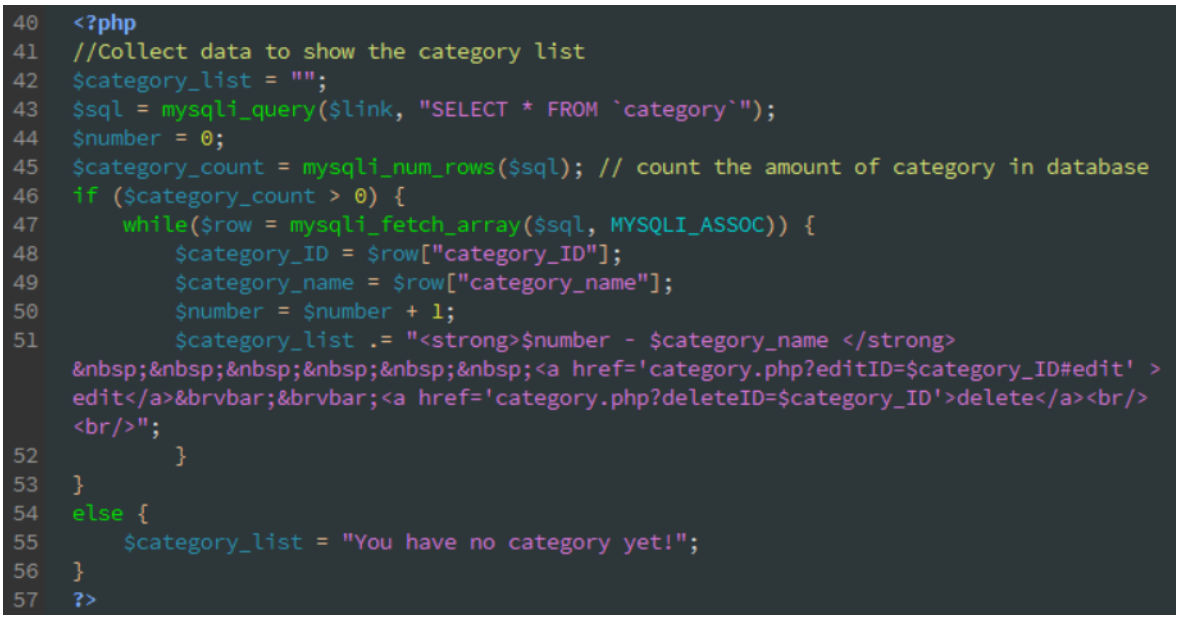


Figure 19. The front page of the store management system (Chu 2015)

Category page

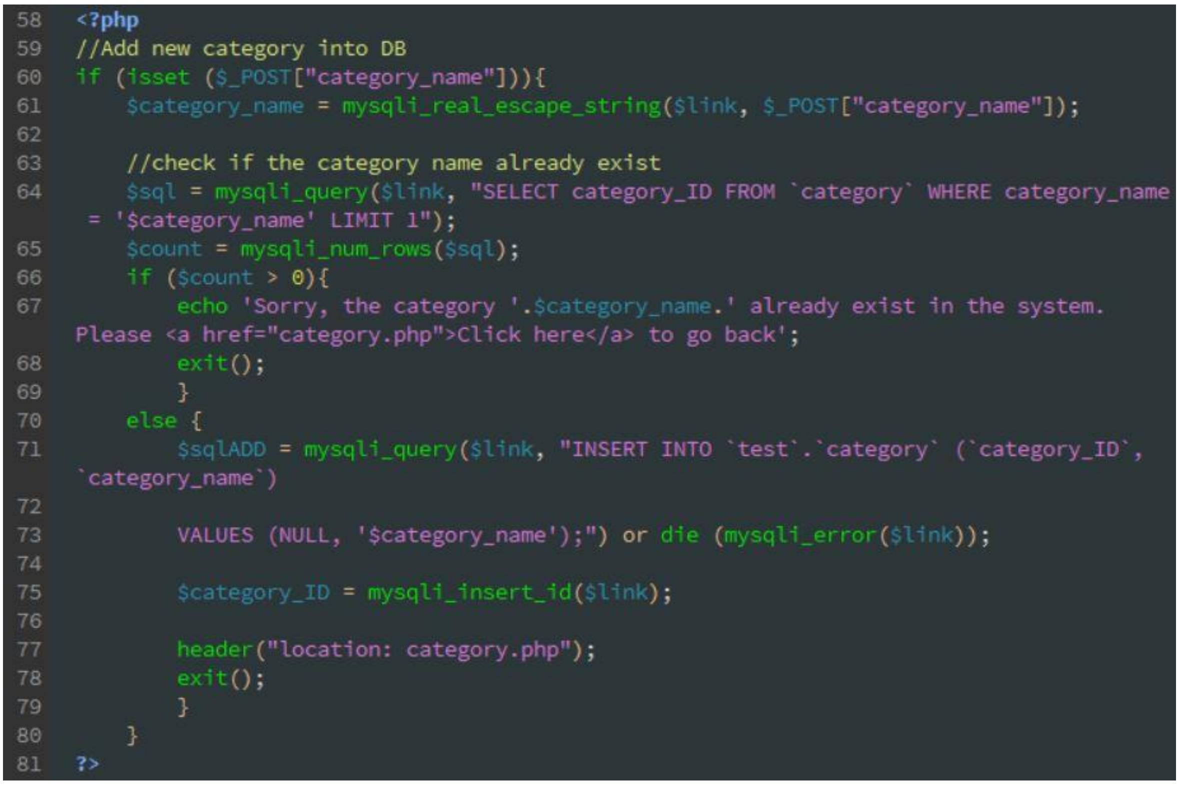
In this page, user can add more categories to the store, view, edit or delete the existing categories. The system will connect to the database, fetch all the categories from it, put them in an array and then return the list in HTML code. When user choose add, edit or delete functions, the corresponded PHP code will be triggered and handle the user's request. (Figure 20.)



```
<?php
//Collect data to show the category list
$category_list = "";
$sql = mysqli_query($link, "SELECT * FROM `category`");
$number = 0;
$category_count = mysqli_num_rows($sql); // count the amount of category in database
if ($category_count > 0) {
    while($row = mysqli_fetch_array($sql, MYSQLI_ASSOC)) {
        $category_ID = $row["category_ID"];
        $category_name = $row["category_name"];
        $number = $number + 1;
        $category_list .= "<strong>$number - $category_name </strong>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href='category.php?editID=$category_ID#edit' >
edit</a>&brvbar;&brvbar;<a href='category.php?deleteID=$category_ID'>delete</a><br/>
<br/>";
        }
}
else {
    $category_list = "You have no category yet!";
}
?>
```

Figure 20. PHP code to collect the categories from database (Chu 2015)

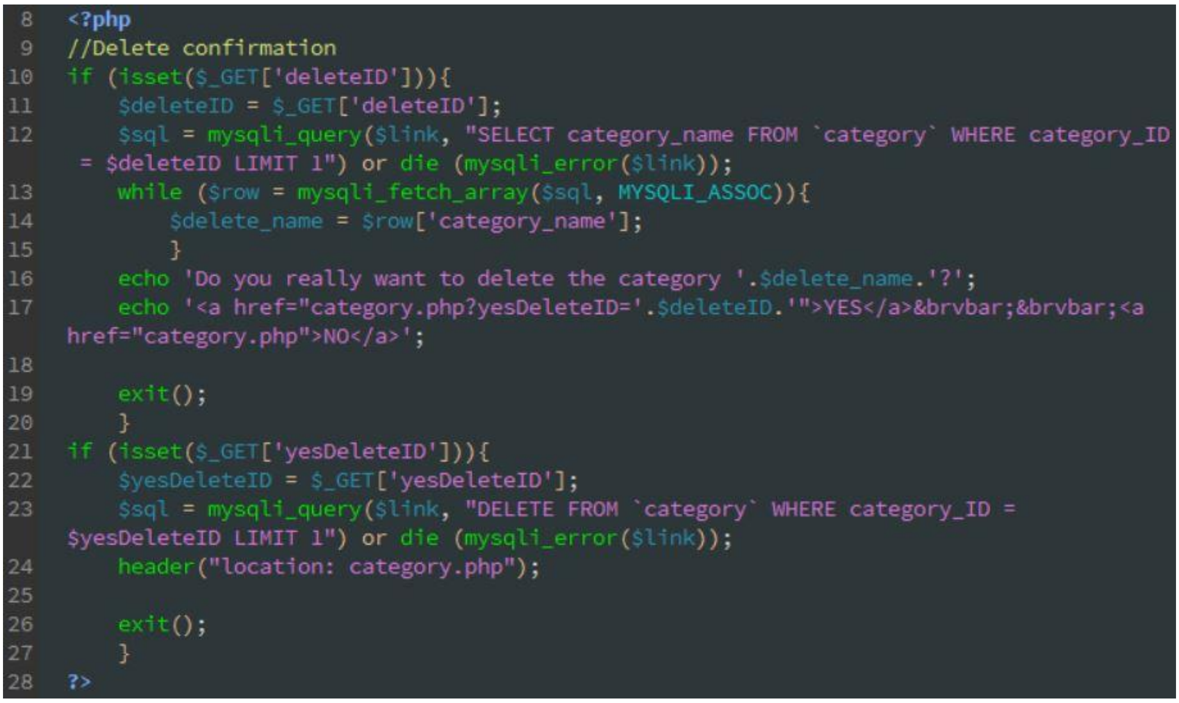
Because the category name should be unique, before adding new category, the system will check if the category that user wanted to add already existed in the database. If there is an exactly similar category, the system will refuse this request and show a warning to the user about the situation. (Figure 21.)



```
<?php
//Add new category into DB
if (isset ($_POST["category_name"])){
    $category_name = mysqli_real_escape_string($link, $_POST["category_name"]);

    //check if the category name already exist
    $sql = mysqli_query($link, "SELECT category_ID FROM `category` WHERE category_name
 = '$category_name' LIMIT 1");
    $count = mysqli_num_rows($sql);
    if ($count > 0){
        echo 'Sorry, the category '.$category_name.' already exist in the system.
Please <a href="category.php">Click here</a> to go back';
        exit();
        }
    else {
        $sqlADD = mysqli_query($link, "INSERT INTO `test`.`category` (`category_ID`,
`category_name`)

        VALUES (NULL, '$category_name');") or die (mysqli_error($link));

        $category_ID = mysqli_insert_id($link);

        header("location: category.php");
        exit();
        }
    }
?>
```

Figure 21. PHP code for adding new category (Chu 2015)

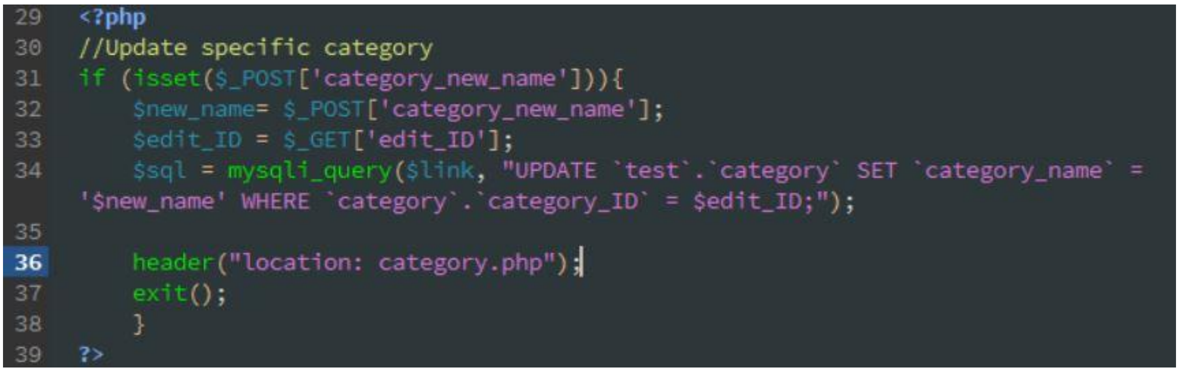
When user wants to delete a specific category, the system will show a confirmation notice to confirm that the user truly wants to delete this category, which avoids a mistaken click (figure 22).



```
<?php
//Delete confirmation
if (isset($_GET['deleteID'])){
    $deleteID = $_GET['deleteID'];
    $sql = mysqli_query($link, "SELECT category_name FROM `category` WHERE category_ID
 = $deleteID LIMIT 1") or die (mysqli_error($link));
    while ($row = mysqli_fetch_array($sql, MYSQLI_ASSOC)){
        $delete_name = $row['category_name'];
        }
    echo 'Do you really want to delete the category '.$delete_name.'?';
    echo '<a href="category.php?yesDeleteID='.$deleteID.'">YES</a>&brvbar;&brvbar;<a
href="category.php">NO</a>';

    exit();
    }
if (isset($_GET['yesDeleteID'])){
    $yesDeleteID = $_GET['yesDeleteID'];
    $sql = mysqli_query($link, "DELETE FROM `category` WHERE category_ID =
$yesDeleteID LIMIT 1") or die (mysqli_error($link));
    header("location: category.php");

    exit();
    }
?>
```

Figure 22. PHP code for deleting a specific category (Chu 2015)

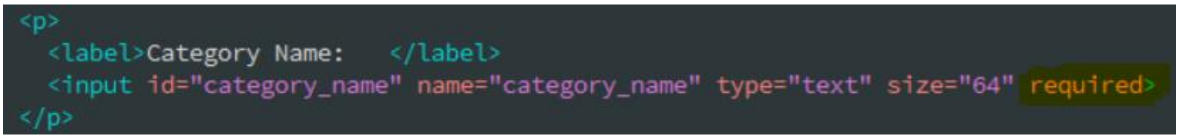
In case user just want to edit an already existed category, the PHP code below will handle that request.



```php
<?php
//Update specific category
if (isset($_POST['category_new_name'])){
    $new_name= $_POST['category_new_name'];
    $edit_ID = $_GET['edit_ID'];
    $sql = mysqli_query($link, "UPDATE `test`.`category` SET `category_name` = '$new_name' WHERE `category`.`category_ID` = $edit_ID;");

    header("location: category.php");
    exit();
    }
?>
```

Figure 23. PHP code for updating a specific category (Chu 2015)

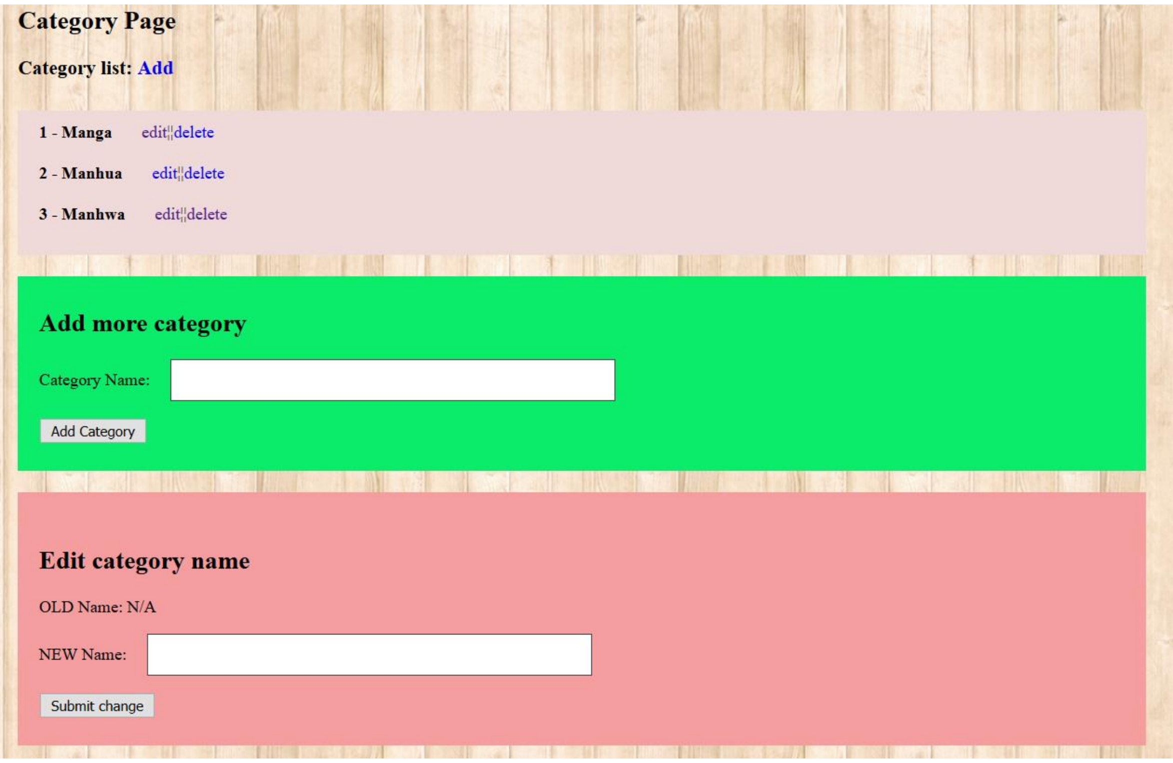
To prevent the user from submitting a blank value, the system uses HTML5 **“required”** attribute for input elements (figure 24).



```html
<p>
  <label>Category Name:   </label>
  <input id="category_name" name="category_name" type="text" size="64" required>
</p>
```

Figure 24. Example of required attribute in HTML5 (Chu 2015)

All functions for this category page are showed in one page. It helps the user to manage the categories easier (figure 25).



Category Page

Category list: Add

1 - Manga edit||delete

2 - Manhua edit||delete

3 - Manhwa edit||delete

Add more category

Category Name:

Edit category name

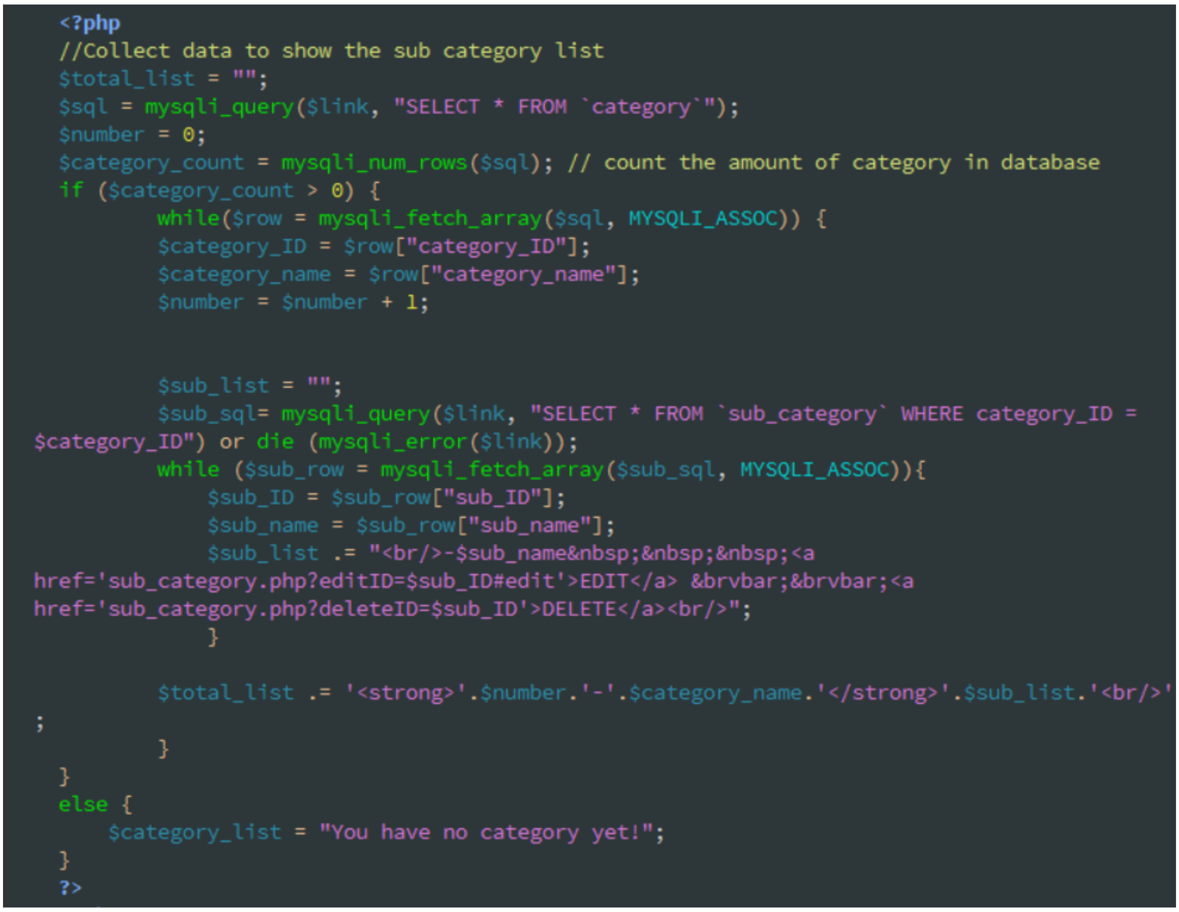
OLD Name: N/A

NEW Name:

Figure 25. UI for Category page (Chu 2015)

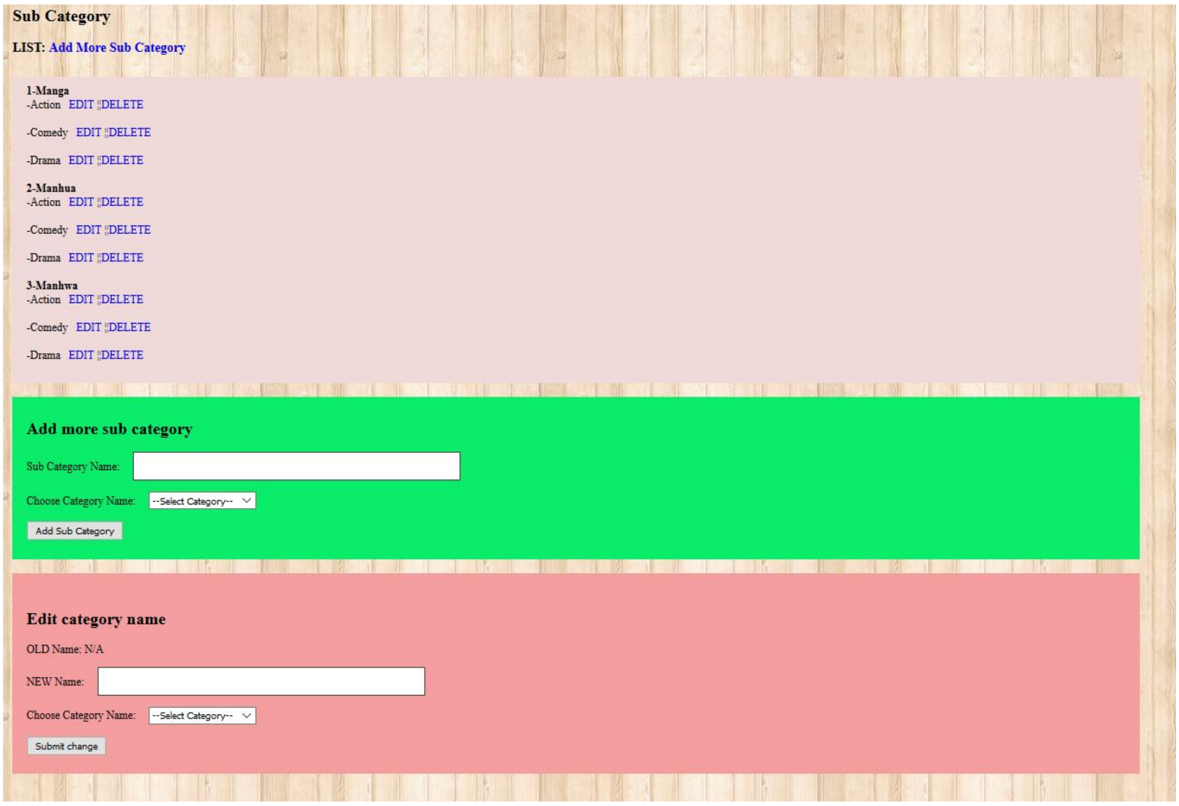
Sub_category page

This page shows all the child categories for each main category and helps the user to manage these sub_categories. All the UI, functions and PHP codes on this page are similar to ‘Category page’. However, the PHP code for showing the sub_category list was modified to show a sub_categories list under each main category. (Figure 26 & 27.)



```
  <?php
  //Collect data to show the sub category list
  $total_list = "";
  $sql = mysqli_query($link, "SELECT * FROM `category`");
  $number = 0;
  $category_count = mysqli_num_rows($sql); // count the amount of category in database
  if ($category_count > 0) {
          while($row = mysqli_fetch_array($sql, MYSQLI_ASSOC)) {
          $category_ID = $row["category_ID"];
          $category_name = $row["category_name"];
          $number = $number + 1;


          $sub_list = "";
          $sub_sql= mysqli_query($link, "SELECT * FROM `sub_category` WHERE category_ID =
$category_ID") or die (mysqli_error($link));
          while ($sub_row = mysqli_fetch_array($sub_sql, MYSQLI_ASSOC)){
              $sub_ID = $sub_row["sub_ID"];
              $sub_name = $sub_row["sub_name"];
              $sub_list .= "<br/>-$sub_name&nbsp;&nbsp;&nbsp;<a
href='sub_category.php?editID=$sub_ID#edit'>EDIT</a> &brvbar;&brvbar;<a
href='sub_category.php?deleteID=$sub_ID'>DELETE</a><br/>";
              }

          $total_list .= '<strong>'.$number.'-'.$category_name.'</strong>'.$sub_list.'<br/>'
;
          }
  }
  else {
      $category_list = "You have no category yet!";
  }
  ?>
```

Figure 26. PHP code for showing sub_category list (Chu 2015)



Sub Category

LIST: Add More Sub Category

1-Manga
-Action EDIT ||DELETE

-Comedy EDIT ||DELETE

-Drama EDIT ||DELETE

2-Manhua
-Action EDIT ||DELETE

-Comedy EDIT ||DELETE

-Drama EDIT ||DELETE

3-Manhwa
-Action EDIT ||DELETE

-Comedy EDIT ||DELETE

-Drama EDIT ||DELETE

Add more sub category

Sub Category Name:

Choose Category Name: --Select Category--

Add Sub Category

Edit category name

OLD Name: N/A

NEW Name:

Choose Category Name: --Select Category--

Submit change

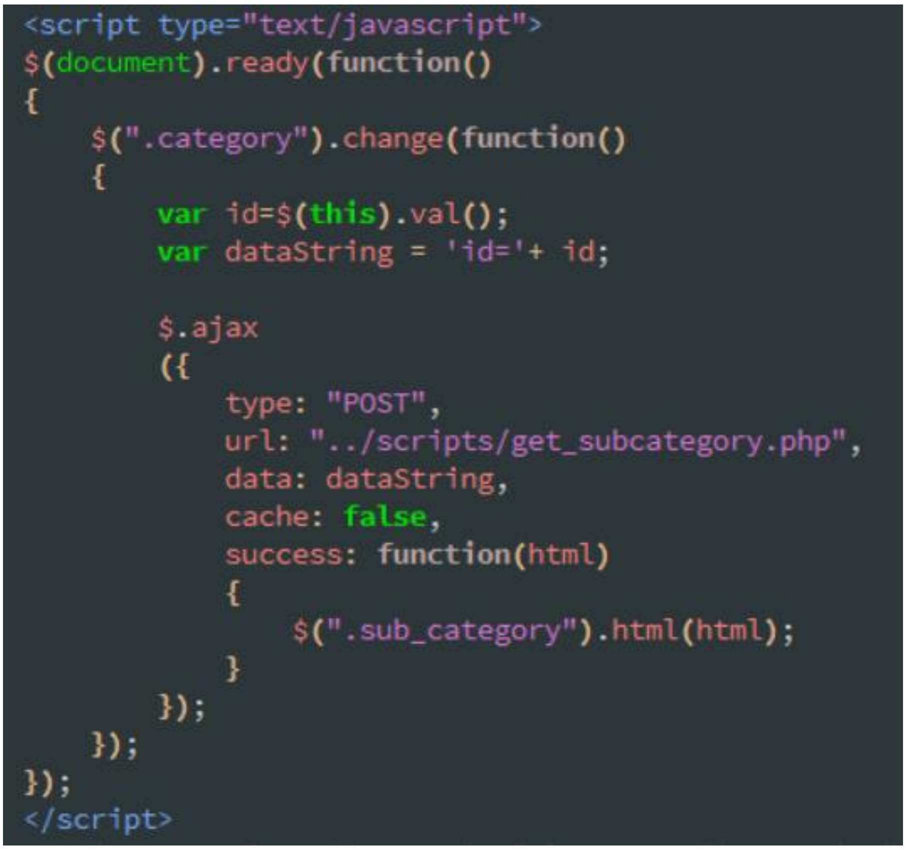
Figure 27. UI for Sub_category page (Chu 2015)

Product page

This page shows all the products of the web shop. It allows users to add new products with related necessary information (name, price, details, category and sub_category).

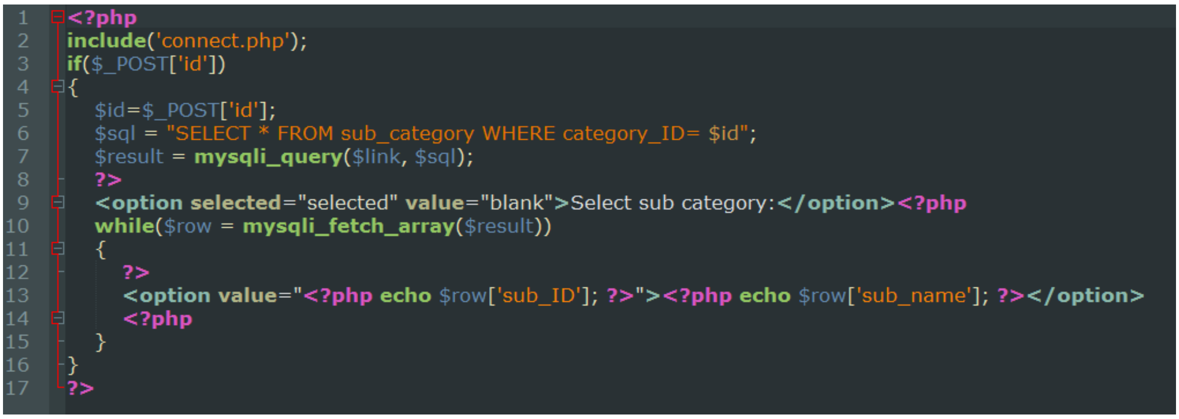
The PHP code for this page is similar, but more complicated than the previous pages because it has to handle more data about products. Each product belongs to a specific sub_category and each sub_category also belongs to a specific main category. When user wants to add new product or edit a product, the system will let the user choose a category first and then show a sub_category selection list based on that.

The system uses AJAX to handle this task. When user selects a category from the dropdown list, AJAX script will get this category's ID and pass it to a PHP function. This function will use the category's ID to get all the sub_categories which belong to that category and return them back as HTML code. (Figure 28 & 29.)



```
<script type="text/javascript">
$(document).ready(function()
{
    $(".category").change(function()
    {
        var id=$(this).val();
        var dataString = 'id='+ id;

        $.ajax
        ({
            type: "POST",
            url: "../scripts/get_subcategory.php",
            data: dataString,
            cache: false,
            success: function(html)
            {
                $(".sub_category").html(html);
            }
        });
    });
});
</script>
```

Figure 28. AJAX code (Chu 2015)



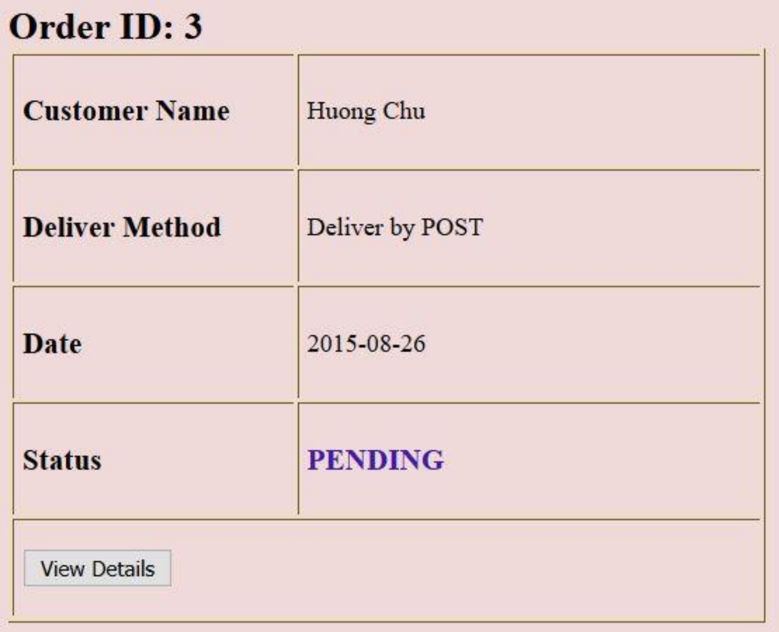
```
<?php
include('connect.php');
if($_POST['id'])
{
    $id=$_POST['id'];
    $sql = "SELECT * FROM sub_category WHERE category_ID= $id";
    $result = mysqli_query($link, $sql);
    ?>
    <option selected="selected" value="blank">Select sub category:</option><?php
    while($row = mysqli_fetch_array($result))
    {
        ?>
        <option value="<?php echo $row['sub_ID']; ?>"><?php echo $row['sub_name']; ?></option>
        <?php
    }
}
?>
```

Figure 29. PHP function to handle the data passed from AJAX (Chu 2015)

Each product has a picture along with it to demonstrate about the product. The picture is stored separately from the database. It is stored in the file system and has a name syntax as: **$product_ID.jpg**. It means the picture has a name that matches the **product_ID** which is unique. This makes the picture's name become unique as well.

Orders page

This page contains all the orders information. At first, the page will show all the summarized orders for a quicker overview. The summarized orders only show selected information for each orders. If the user wants to see more details about one specific order, he can choose to click the **"View Details"** button. The system will redirect the user to the **order_details.php** page. (Figure 30.)



Order ID: 3

Customer Name	Huong Chu
Deliver Method	Deliver by POST
Date	2015-08-26
Status	**PENDING**

Figure 30. A summarized order (Chu 2015)

The **order_details.php** page has the dynamic content. It will base on the order's ID to generate the content that corresponds to a specific order. In this page, the user can see all the information of an order included the customer's information. The system allows the user to update the status of an order or edit the customer's information if needed.

3.2.5 Store demonstration pages and shopping cart

Home page design

This page displays all information about the store. It contains of five main parts:

a) Header: Display the store logo, the cart logo which leads customers to their shopping cart
b) Banner: Display the banner for the store
c) Menu bar: Display the main categories and the list of sub_categories underneath
d) Main Content: Display the products with pictures, prices
e) Footer: Display additional information about the store; for example, about the store, address, phone number, social networks, etc.

Dreamweaver cc 2015 provides ready website templates for blog, email, portfolio and e-commerce. In this project, the template for e-commerce was used as a base template for the whole system. (Figure 31.)

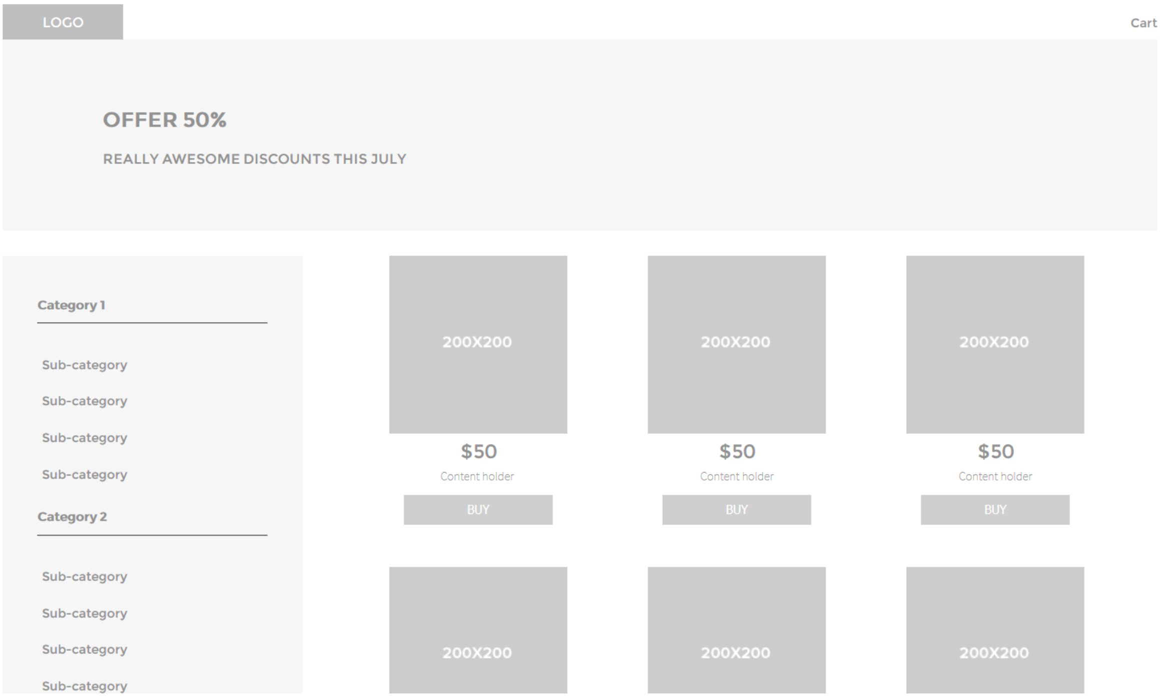
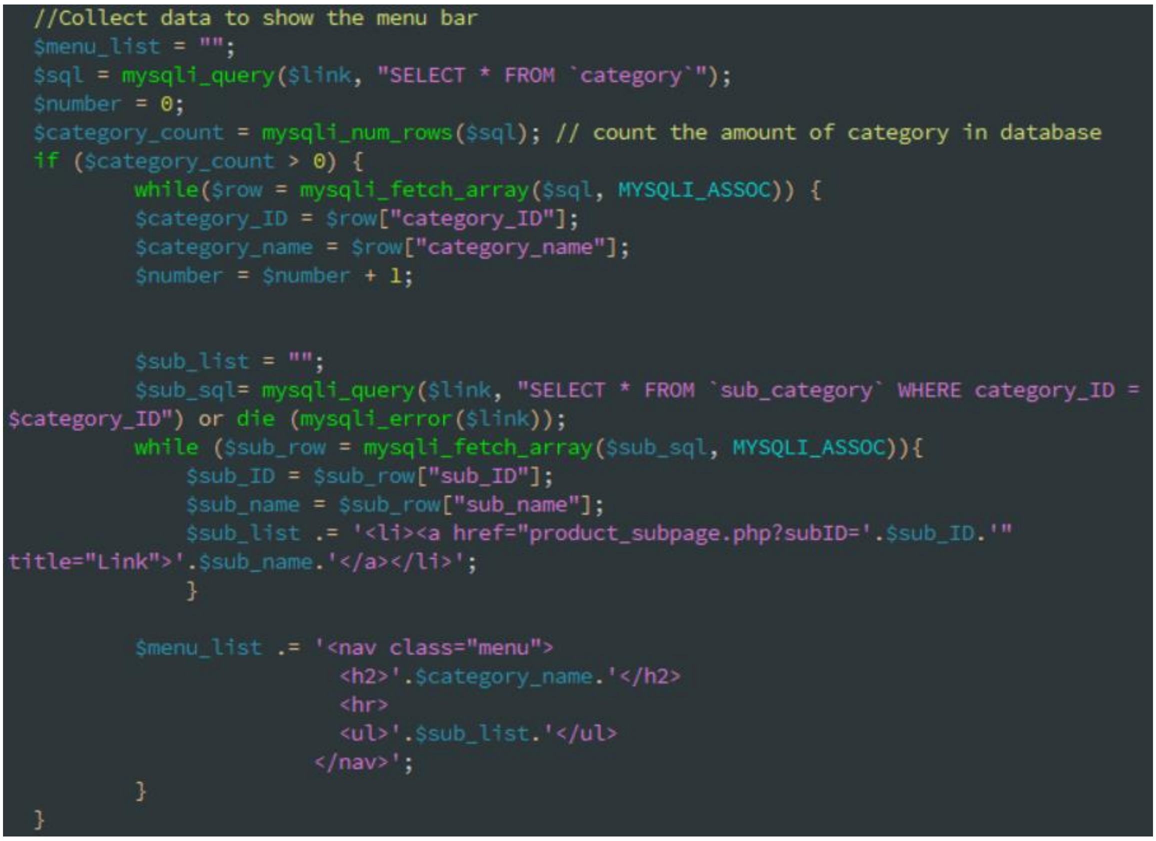


Figure 31. Automatically generated template by Dreamweaver

The menu bar on the left side was built to be dynamical with PHP. It fetches the data from the database and displays the category list according to the data it gets. When the customer chooses to click a sub_category link on the menu bar, he will be redirected to a page named **product_subpage.php**. This page displays all the products which belong to this sub_category. (Figure 32.)



```
  //Collect data to show the menu bar
  $menu_list = "";
  $sql = mysqli_query($link, "SELECT * FROM `category`");
  $number = 0;
  $category_count = mysqli_num_rows($sql); // count the amount of category in database
  if ($category_count > 0) {
          while($row = mysqli_fetch_array($sql, MYSQLI_ASSOC)) {
          $category_ID = $row["category_ID"];
          $category_name = $row["category_name"];
          $number = $number + 1;


          $sub_list = "";
          $sub_sql= mysqli_query($link, "SELECT * FROM `sub_category` WHERE category_ID =
$category_ID") or die (mysqli_error($link));
          while ($sub_row = mysqli_fetch_array($sub_sql, MYSQLI_ASSOC)){
              $sub_ID = $sub_row["sub_ID"];
              $sub_name = $sub_row["sub_name"];
              $sub_list .= '<li><a href="product_subpage.php?subID='.$sub_ID.'"
title="Link">'.$sub_name.'</a></li>';
              }

          $menu_list .= '<nav class="menu">
                          <h2>'.$category_name.'</h2>
                          <hr>
                          <ul>'.$sub_list.'</ul>
                        </nav>';
          }
  }
```

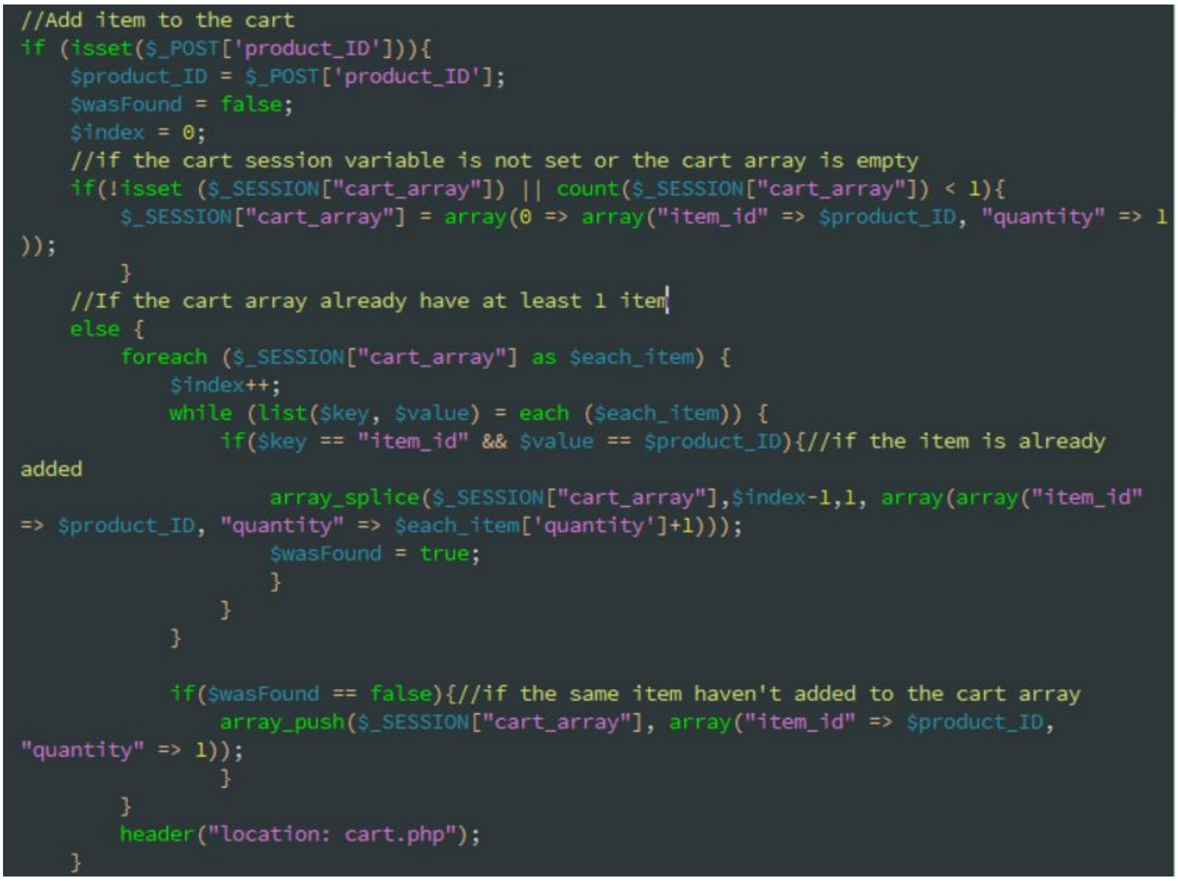
Figure 32. PHP code for showing the menu bar (Chu 2015)

When the customer clicks a product picture, he will be redirected to the **product page** which shows more details about this product. This product page is dynamical. It generates the contents based on the product's ID.

On the main content area, it shows all the products of the store with pagination. The pagination is quite important. It helps the UI look better when there is about hundreds of products on the web store.

Shopping cart

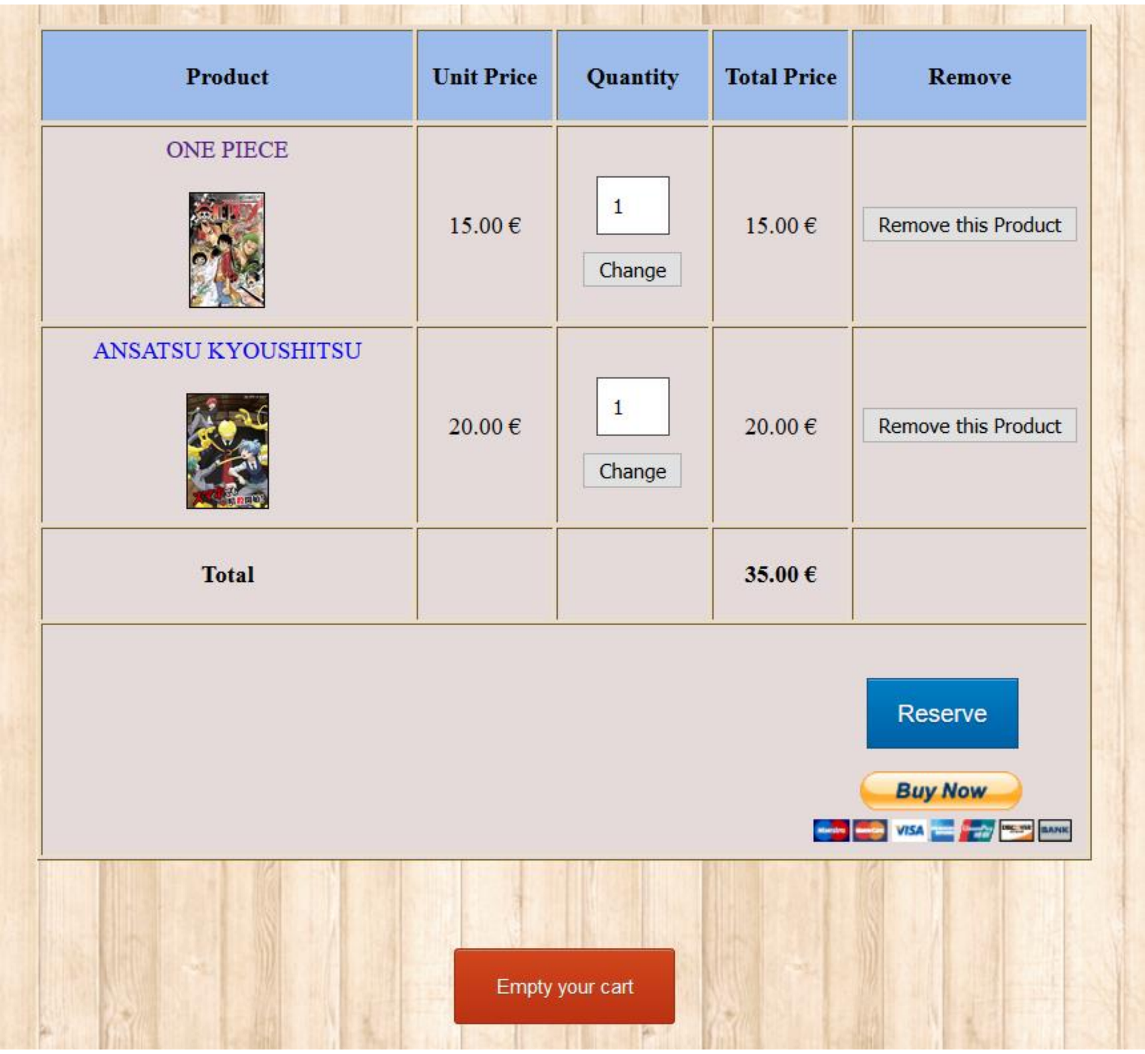
One of the most important features for an e-commerce site is the virtual shopping cart. After choosing a product, the customer can add this product to his shopping cart and then check out if he wants or continue shopping. To build the shopping cart, the system use a php page called **cart.php**. When the customer chooses to add a product into his shopping cart, a session called "cart_array" which contains pairs of key and value will be created. The key is the product's ID and the value is the product's quantity. After that, customer will be redirected to the cart.php page. (Figure 33.)



```
//Add item to the cart
if (isset($_POST['product_ID'])){
    $product_ID = $_POST['product_ID'];
    $wasFound = false;
    $index = 0;
    //if the cart session variable is not set or the cart array is empty
    if(!isset ($_SESSION["cart_array"]) || count($_SESSION["cart_array"]) < 1){
        $_SESSION["cart_array"] = array(0 => array("item_id" => $product_ID, "quantity" => 1
));
        }
    //If the cart array already have at least 1 item
    else {
        foreach ($_SESSION["cart_array"] as $each_item) {
            $index++;
            while (list($key, $value) = each ($each_item)) {
                if($key == "item_id" && $value == $product_ID){//if the item is already
added
                    array_splice($_SESSION["cart_array"],$index-1,1, array(array("item_id"
=> $product_ID, "quantity" => $each_item['quantity']+1)));
                    $wasFound = true;
                    }
                }
            }

            if($wasFound == false){//if the same item haven't added to the cart array
                array_push($_SESSION["cart_array"], array("item_id" => $product_ID,
"quantity" => 1));
                }
        }
        header("location: cart.php");
    }
```

Figure 33. PHP code for creating the cart_array (Chu 2015)

On the cart.php page, the customer can see all of the items that he added into his shopping cart. The system allows customers to manage their shopping cart by changing the quantity of each item, removing a specific item or emptying the whole shopping cart. (Figure 34.)

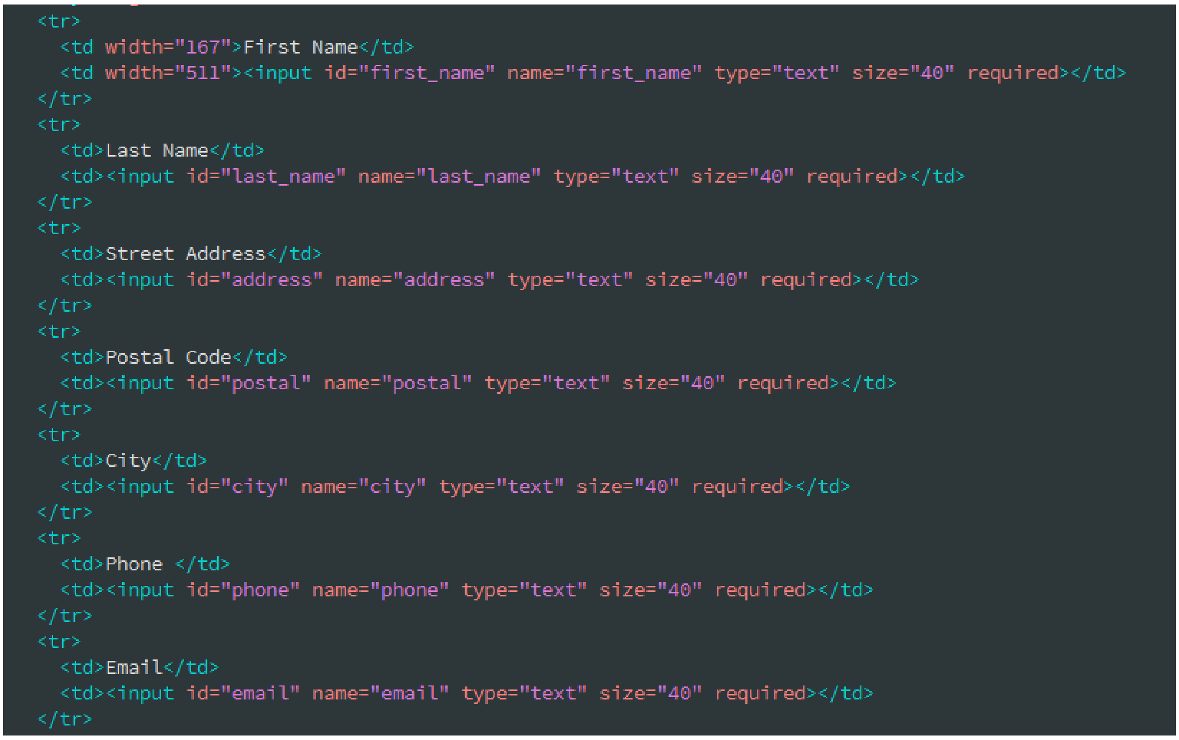


Product	Unit Price	Quantity	Total Price	Remove
ONE PIECE	15.00 €	1 Change	15.00 €	Remove this Product
ANSATSU KYOUSHITSU	20.00 €	1 Change	20.00 €	Remove this Product
Total			35.00 €	

Figure 34. Shopping cart page (Chu 2015)

If the customer wants to check out, the system will provide two options for him. The customer can reserve the products on the shopping cart and choose either to pick up those products at store or have those products delivered to an address by post. The second option is paying for those products immediately using the PayPal gateway.

If the customer chooses to reserve the products on the shopping cart, the system will ask the customer to provide necessary information with a reservation form. After the customer submits the form, the system will automatically place an order into the database. At the same time, a notification email will be sent to the shop owner to announce him the new order. This reservation form uses “required” attribute of HTML5 to validate the input fields. (Figure 35.)



```
<tr>
  <td width="167">First Name</td>
  <td width="511"><input id="first_name" name="first_name" type="text" size="40" required></td>
</tr>
<tr>
  <td>Last Name</td>
  <td><input id="last_name" name="last_name" type="text" size="40" required></td>
</tr>
<tr>
  <td>Street Address</td>
  <td><input id="address" name="address" type="text" size="40" required></td>
</tr>
<tr>
  <td>Postal Code</td>
  <td><input id="postal" name="postal" type="text" size="40" required></td>
</tr>
<tr>
  <td>City</td>
  <td><input id="city" name="city" type="text" size="40" required></td>
</tr>
<tr>
  <td>Phone </td>
  <td><input id="phone" name="phone" type="text" size="40" required></td>
</tr>
<tr>
  <td>Email</td>
  <td><input id="email" name="email" type="text" size="40" required></td>
</tr>
```

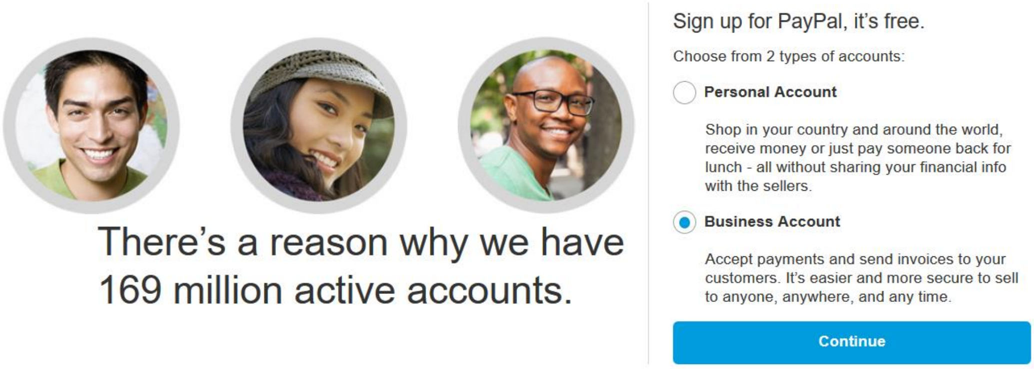
Figure 35. HTML reservation form (Chu 2015)

3.2.6 Payment system

In this project, the e-commerce store uses PayPal gateway as the payment system. PayPal provides a testing environment for e-commerce site called Sandbox. Sandbox helps merchants which use PayPal services to test their payment system. Sandbox has most of the same features as real PayPal services.

> The PayPal Sandbox mirrors the features found on the PayPal production servers. While some PayPal features do not apply to the Sandbox (such as closing an account, issuing monthly statements, storing shipping preferences, and the support of PayPal Shops), the Sandbox has parity with the PayPal API feature set supported by the live environment. This means you can test your PayPal processes and know they will behave the same on the production servers as they do in the Sandbox environment. (PayPal 2015.)

To use Sandbox, the first requirement is a PayPal account. To create a PayPal account, it is instructed to access the URL: https://www.paypal.com/signup/account and then follow the instruction on the page. The account type should be business. (Figure 36.)



There’s a reason why we have 169 million active accounts.

Sign up for PayPal, it’s free.

Choose from 2 types of accounts:

○ **Personal Account**

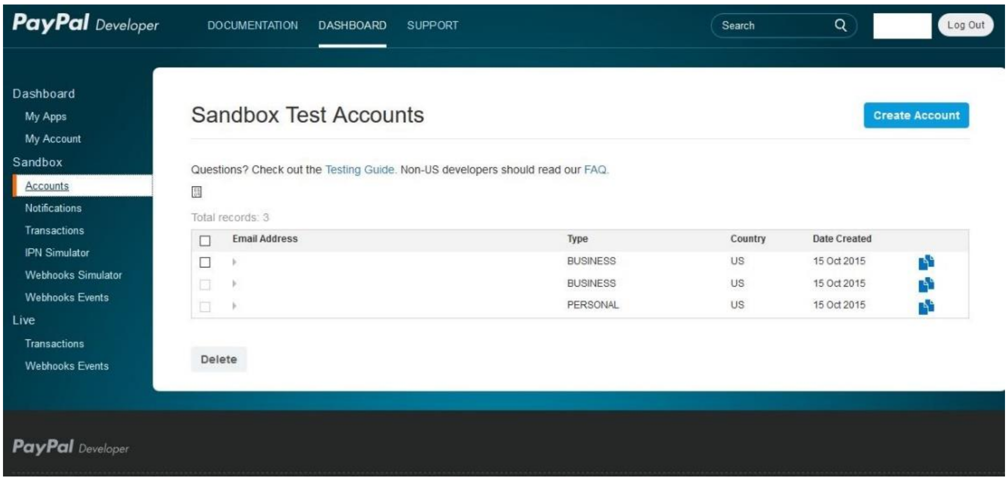
Shop in your country and around the world, receive money or just pay someone back for lunch - all without sharing your financial info with the sellers.

● **Business Account**

Accept payments and send invoices to your customers. It's easier and more secure to sell to anyone, anywhere, and any time.

Figure 36. PayPal’s sign up page (PayPal 2015)

The second requirement is to have at least two test accounts, one test account with the role of seller and the other one with the role of buyer. After creating a PayPal account, it is required to go to PayPal developer zone by follow the URL: https://developer.paypal.com then it is possible to login using the PayPal account. On the “Developer” page, it is needed to click “Dashboard” tab on the top and then navigate to the “account” tab under the “Sandbox” title on the left menu bar. From there, it is possible to create two test accounts for buyer and seller. The buyer account is personal account and the seller account is business account. The balance of test accounts can be set from 1 to 10 million US dollars. (Figure 37.)



Sandbox Test Accounts

Questions? Check out the Testing Guide. Non-US developers should read our FAQ.

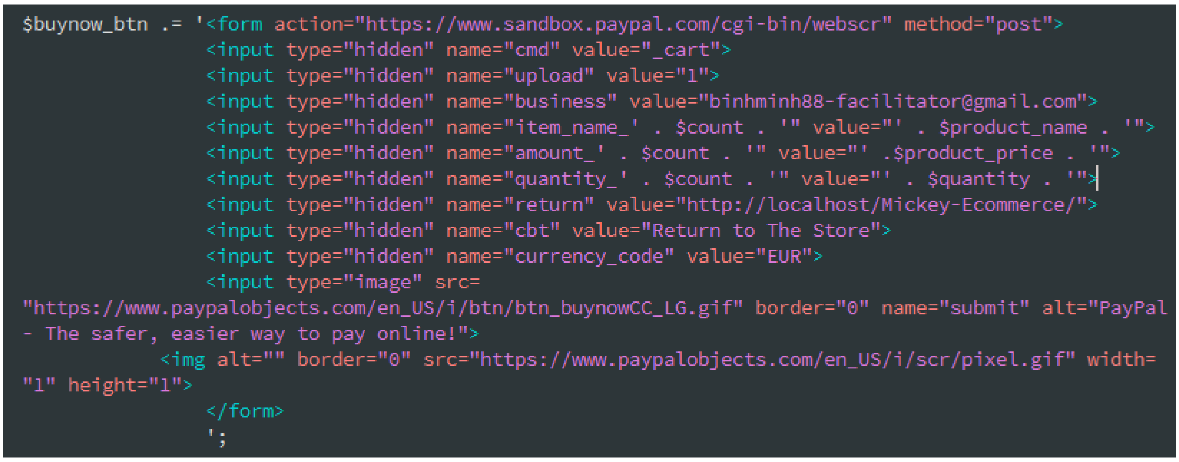
Total records: 3

☐	Email Address	Type	Country	Date Created
☐	▸	BUSINESS	US	15 Oct 2015
☐	▸	BUSINESS	US	15 Oct 2015
☐	▸	PERSONAL	US	15 Oct 2015

Figure 37. Sandbox account (PayPal Developer 2015)

The third requirement is integrating the seller test account with the e-commerce web application. Each time the customer checks out his shopping cart by using PayPal service, the system will pass all the data about products and their prices to the Sandbox server with the target money receiver is the seller test account.

The HTML code for the PayPal button must contain a number of default variables in order to connect with the PayPal server. These variables will be sent to PayPal server when the customer clicks the “Buy Now” button on the shopping cart page. (Figure 38.) More details about the variables can be seen from the appendix 3.



```
$buynow_btn .= '<form action="https://www.sandbox.paypal.com/cgi-bin/webscr" method="post">
                <input type="hidden" name="cmd" value="_cart">
                <input type="hidden" name="upload" value="1">
                <input type="hidden" name="business" value="binhminh88-facilitator@gmail.com">
                <input type="hidden" name="item_name_' . $count . '" value="' . $product_name . '">
                <input type="hidden" name="amount_' . $count . '" value="' .$product_price . '">
                <input type="hidden" name="quantity_' . $count . '" value="' . $quantity . '">
                <input type="hidden" name="return" value="http://localhost/Mickey-Ecommerce/">
                <input type="hidden" name="cbt" value="Return to The Store">
                <input type="hidden" name="currency_code" value="EUR">
                <input type="image" src=
"https://www.paypalobjects.com/en_US/i/btn/btn_buynowCC_LG.gif" border="0" name="submit" alt="PayPal
- The safer, easier way to pay online!">
            <img alt="" border="0" src="https://www.paypalobjects.com/en_US/i/scr/pixel.gif" width=
"1" height="1">
                </form>
                ';
```

Figure 38. PHP code for the PayPal button (Chu 2015)

When the payment system was successfully integrated with Sandbox, the e-commerce web application was ready for testing.

3.3 Test the e-commerce web application

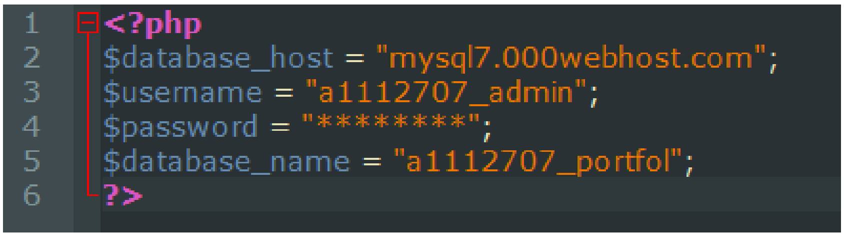
The e-commerce web application was tested through different use cases. These use cases was created based on the goals of the project and the requirements for this application. Based on the back-end and front-end systems that were built in the previous phases, some fiction data was added into the database to build a test web shop. The chosen test web shop is an online comic store. This store sells products which are comic books. All the products on this store are not real and only for testing purpose. The application was installed on a local host which powered by WAMP.

There are two types of actors for the e-commerce web shop’s use cases:

a) **Normal user:** this user is the customer who visits the online store and buys products.
b) **Administrative user:** this user is the web shop owner. The administrative user has the administrative account to manage the store.

3.3.1 Online testing environment setup

A free hosting service provided by Hostinger International Limited was chosen as the testing environment for this project. The first step was registering a free account at the address: https://www.000webhost.com/. With the “Free Hosting” plan, Hostinger supports both PHP and MySQL on their server, which meets the minimum requirements for running the e-commerce web application. The second step was uploading the project to the hosting server. After that, the PHP configuration file needed to be edited in order to connect with the database on a new server (figure 39).



```php
<?php
$database_host = "mysql7.000webhost.com";
$username = "a1112707_admin";
$password = "********";
$database_name = "a1112707_portfol";
?>
```

Figure 39. Configuration PHP code for connecting with the database (Chu 2015)

After successfully setting up the testing environment for the web shop, the URL for the web shop is: http://mickeychu.netii.net/Mickey-Ecommerce/

3.3.2 Use cases for administrative users

Create new categories, sub_categories and products

The table below shows the use case when the shop owner firstly wanted to create new categories, sub_categories and add new products to his store.

Table 1. Use case 001 (Chu 2015)

Element	Description
Use case name	Create new categories, sub_categories and products
Use case ID	001
Author	Chu Binh Minh
Last updated	02.11.2015
Primary actor	Web shop owner
Preconditions	The owner was provided with an administrative account with username and password. He used this account to log in to the store management system.
Main Scenario	The shop owner used the store management system to create some categories, then sub_categories for each category; finally, added some new products with pictures.
Post condition	All the categories, sub_categories and products were created as well as all data was added successfully to the database. The shop owner could see all of his new products and categories on the front page of the store.

Update existed categories, sub_categories and products

The table below shows the use case when the shop owner wanted to change some details of the categories, sub_categories or products that he had added on the use case 001.

Table 2. Use case 002 (Chu 2015)

Element	Description
Use case name	Update existed categories, sub_categories and products
Use case ID	002
Author	Chu Binh Minh
Last updated	02.11.2015
Primary actor	Web shop owner
Preconditions	The owner was provided with an administrative account with username and password. He used this account to log in to the store management system.
Main Scenario	The owner used the store management system to update few random categories, sub_categories and products the he had added before.
Post condition	The selected categories, sub_categories and products were updated successfully and these changes could be seen from the store management system or the shop front page.

Delete existed categories, sub_categories and products

The table below shows the use case when the shop owner wanted to delete some of the categories, sub_categories or products that he had added on the use case 001.

Table 3. Use case 003 (Chu 2015)

Element	Description
Use case name	Delete existed categories, sub_categories and products
Use case ID	003
Author	Chu Binh Minh
Last updated	02.11.2015
Primary actor	Web shop owner
Preconditions	The owner was provided with an administrative account with username and password. He used this account to log in to the store management system.
Main Scenario	The owner used the store management system to delete some random products. If he wanted to delete a sub_category, he could delete all the products that belonged to this category or he could delete the sub_category right away if there was no product belongs to this sub_category. The same condition was applied to delete category.
Post condition	The selected products, sub_categories or category were successfully deleted from the database. The changes could be seen from the store management system or the shop front page.

Manage the orders page

The table below shows the use case when the shop owner wanted to view the orders that the customer had made.

Table 4. Use case 004 (Chu 2015)

Element	Description

Use case name	Manage the orders page
Use case ID	004
Author	Chu Binh Minh
Last updated	02.11.2015
Primary actor	Web shop owner
Preconditions	The owner was provided with an administrative account with username and password. He used this account to log in to the store management system.
Main Scenario	The owner used the store management system to view the order page. He could update the status of a random order and update the customer information for this order.
Post condition	The selected order was updated successfully. The order table and customer table in the database was updated successfully.

3.3.3 Use cases for normal user

View products on the web shop

The table below shows the use case when the customer wanted to visit the online e-commerce website.

Table 5. Use case 005 (Chu 2015)

Element	Description
Use case name	View products on the web shop
Use case ID	005
Author	Chu Binh Minh
Last updated	02.11.2015
Primary actor	Customer
Preconditions	The customer was provided with the URL of the web shop.
Main Scenario	The customer used his internet browser to visit the online web shop and viewed the products on the web shop individually or by sub_categories.
Post condition	The customer successfully saw all the products on the web shop.

Add products to shopping cart and manage the shopping cart

The table below shows the use case when the customer had visited the web shop. He wanted to add products into his virtual shopping cart.

Table 6. Use case 006 (Chu 2015)

Element	Description
Use case name	Add products to shopping cart and manage the shopping cart
Use case ID	006
Author	Chu Binh Minh
Last updated	02.11.2015
Primary actor	Customer
Preconditions	The customer had already visited the web shop.

Main Scenario	The customer chose some products, added them to his shopping cart then managed the shopping cart by adding more products, changing the quantity of random products or emptying the whole shopping cart.
Post condition	All selected products were added successfully to the shopping cart. The customer successfully added more products, changed quantity of products on the shopping cart and emptied the shopping cart.

Check out the shopping cart

The table below shows the use case when the customer already added products into his shopping cart. Then he wanted to check out with the shopping cart.

Table 7. Use case 007 (Chu 2015)

Element	Description
Use case name	Add products to shopping cart and manage the shopping cart
Use case ID	007
Author	Chu Binh Minh
Last updated	02.11.2015
Primary actors	Customer, web shop owner
Preconditions	The customer had already added some products into his shopping cart.
Main Scenario	The customer could choose to reserve these products through the reserving form or pay for his cart using PayPal.
Post condition	**In case the customer chose to reserve the products:** After submitting the reservation form, the system successfully created an order with exact information and sent a notification email to the registered email of the web shop owner. The owner could see the email then checked the new order with order ID on the orders page of the store management system. **In case the customer chose to pay with PayPal service:** Both of the customer and the shop owner received the notification email about the transaction.

The figure below shows all the use cases for the e-commerce web application in this project.

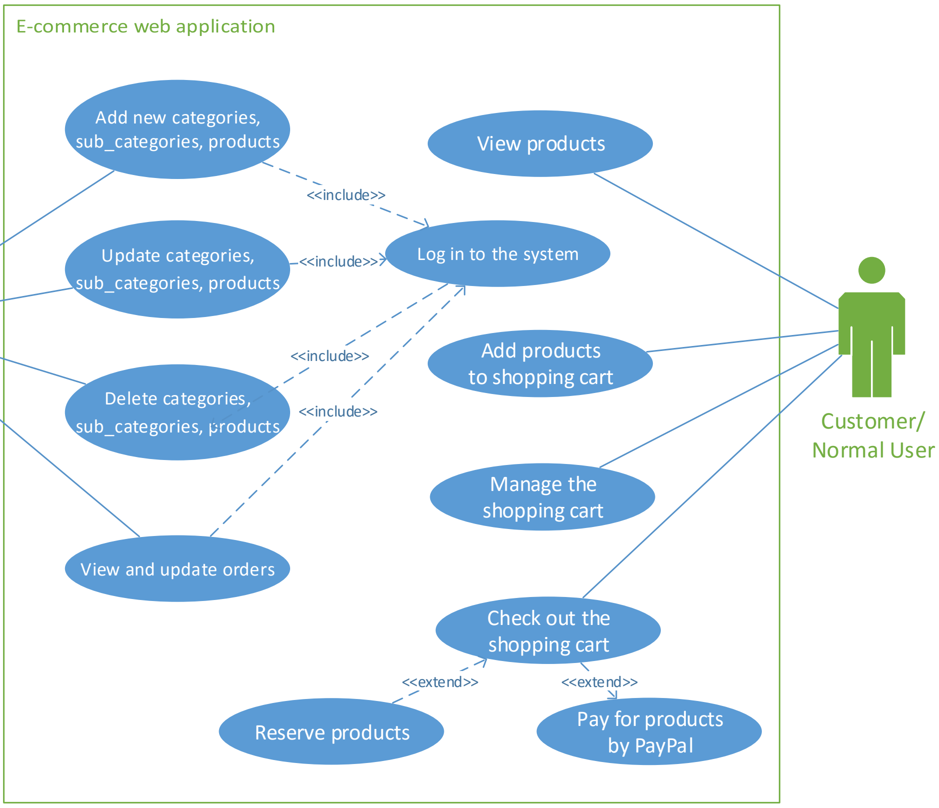
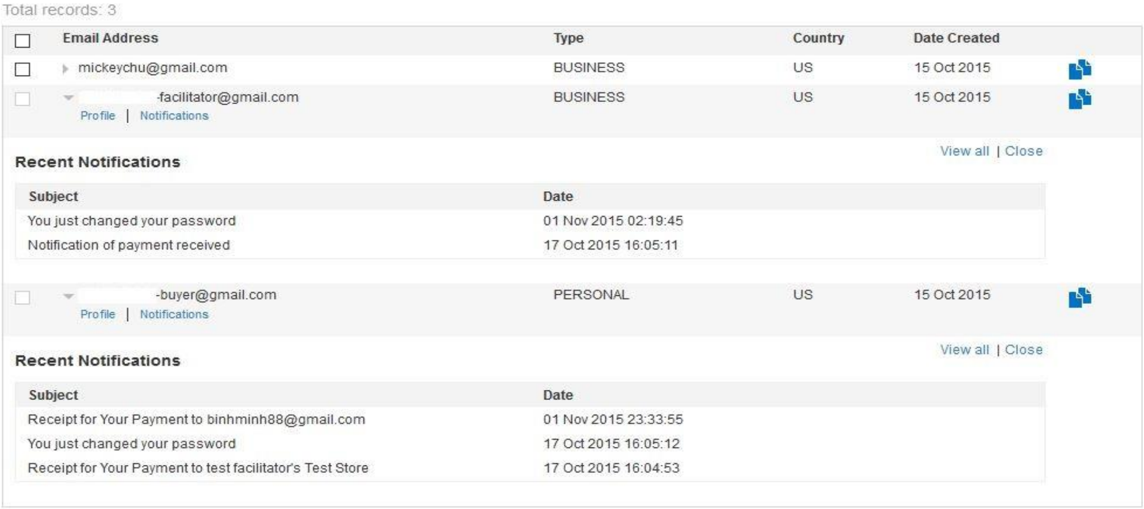

E-commerce web application
Add new categories, sub_categories, products
View products
<<include>>
Update categories, sub_categories, products
<<include>>
Log in to the system
<<include>>
Add products to shopping cart
Delete categories, sub_categories, products
<<include>>
Shop Owner/ Administrative user
Customer/ Normal User
Manage the shopping cart
View and update orders
Check out the shopping cart
<<extend>>
<<extend>>
Reserve products
Pay for products by PayPal

Figure 40. UML use case diagram (Chu 2015)

3.3.4 Result of the test

All the test created through the use cases were successful, they all met the post conditions. Regarding the PayPal service, because all the transactions were made on the Sandbox environment, the notifications about transactions could only be checked on the PayPal developer zone. In the real PayPal service, the notification emails would be sent to the actual emails of buyers and sellers. The figure below shows that the notifications for both buyer and seller can be seen on the same page which is Sandbox test account page.



Total records: 3

	Email Address	Type	Country	Date Created
□	mickeychu@gmail.com	BUSINESS	US	15 Oct 2015
□	[redacted]-facilitator@gmail.com Profile | Notifications	BUSINESS	US	15 Oct 2015

Recent Notifications

View all | Close

Subject	Date
You just changed your password	01 Nov 2015 02:19:45
Notification of payment received	17 Oct 2015 16:05:11

	Email Address	Type	Country	Date Created
□	[redacted]-buyer@gmail.com Profile | Notifications	PERSONAL	US	15 Oct 2015

Recent Notifications

View all | Close

Subject	Date
Receipt for Your Payment to binhminh88@gmail.com	01 Nov 2015 23:33:55
You just changed your password	17 Oct 2015 16:05:12
Receipt for Your Payment to test facilitator's Test Store	17 Oct 2015 16:04:53

Figure 41. Notifications view of Sandbox test accounts (PayPal Developer 2015)

4 Discussion

This chapter discusses the outcome of the project and in comparison with the goals that set from the beginning. It also includes the recommendation for future development.

4.1 Conclusion and evaluation

Conclusion

The two main objectives of this thesis project were to deliver a simple e-commerce web application and to associate with learning.

For the first objective, after almost four months of carrying out this project, the ready e-commerce web shop was born. It has had the basic functions to support the shop owner to promote and sell his products online. In the other hand, the application also met the requirements from customers' points of view about an online shopping store. Finally, the first goal was successfully achieved.

Regarding the background theory part, there were numerous valuable and reliable sources from the Haaga-Helia's library. The book "Web development and design foundations with HTML5" by Felke-Morris, T. (2013) did provide many competent knowledge about e-commerce as well as the background information about developing web application. In the development process, the book "Effortless E-Commerce with PHP and MySQL" by Ullman, L (2014) was the undisputed part. The development plan and implementation was made based on the excellent guide from the book. Throughout the project development process, I have learnt hugely about the protocols, methods and many more necessary factors that needed to develop a web application using mainly PHP and MySQL. The second goal of this project was fully accomplished.

Evaluation

The project was started in August 2015. After three months, the first version of the e-commerce web application was ready to be tested on an online hosting service. From the beginning, the development process was quite slow due to the lack of working experience with developing web application in general and with PHP in particular. However, PHP has a strong community and there are numerous dedicated supporters for it. Their posts and guide were a big help in the development process. The project got back on track and has finally been completed in November 2015.

Despite the risks that might happen during operation, more and more e-commerce sites have been arisen in the past few years. Along with it is the solid growth of digital shoppers. E-commerce has broken many tangible and intangible barriers of traditional commerce and become the intimate factor of modern business. (Figure 42.)

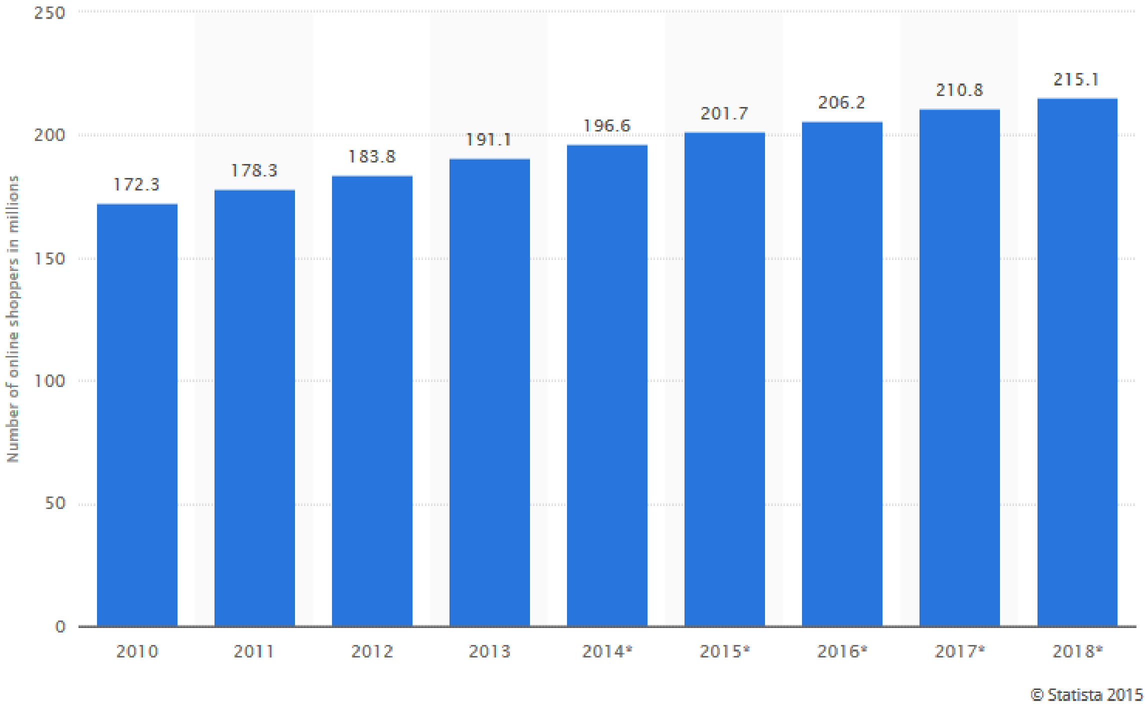


Figure 42. Number of digital shoppers in the United States from 2010 to 2018 (in millions) (Statista 2015)

From another perspective, employees in the IT sectors are now given more job opportunities related to e-commerce. There are especial needs of experts in developing e-commerce applications, not only web application but also mobile application. It is the great chance for future developers or who are interested in the e-commerce field.

4.2 Recommendations for future development

The final product which was the outcome of the thesis project roles as the model system. It could be applied for any type of small size shops. The leftover work for the shop owner is inserting the data about products and then choosing a theme for the front pages of the store.

Nevertheless, there is still much room to develop this e-commerce system further. The first and quite easy task will be adding the search and filter function to the system. Currently, the customer can only filter the products by sub_categories. In the future, developer can add more filter features such as filter products by prices, by languages or by popularity.

Moreover, the integration with PayPal service can be developed further. The system can operate more effectively by using PayPal IPN (Instant Payment Notification). Once it is set up, it will send back information about processed transactions to the e-commerce site's server. It is very useful in many cases; for examples, it could be used to enable instant download from the website after customers pay.

Another feature needed for the future e-commerce web application will be the sign up and login system for customers. With the system, a customer can create an account to have better experience with the online store. Based on this new feature, the customer can add products to the wish list, save them to the shopping cart to continue shopping or view the history of orders that they made with the store.

Last but not least, the whole system can be re-write using PHP framework for the back-end or JavaScript framework for front-end. The target frameworks can be Laravel PHP framework and AngularJS framework. These frameworks have numerous useful packages that are ready to use, which helps reduce the time for development while increasing the security level for the application.

Bibliography

Alibaba Group 2014. History and Milestones. URL: http://www.alibabagroup.com/en/about/history. Accessed: 10 October 2015.

Beal, V. 2015. ARPANET. URL: http://www.webopedia.com/TERM/A/ARPANET.html. Accessed: 22 October 2015.

Benvenuti, S. 2014. A Brief History of e-Commerce. URL: http://www.pertechindustries.com/a-brief-history-of-ecommerce/. Accessed: 22 October 2015.

CNN Money 2014. What is Bitcoin? URL: http://money.cnn.com/infographic/technology/what-is-bitcoin/. Accessed: 10 October 2015.

CoinDesk 2015. What is Bitcoin? URL: http://www.coindesk.com/information/what-is-bitcoin/. Accessed: 10 October 2015.

Engine Directory 2015. History of Yahoo! Search – The Original Search Engine. URL: http://searchenginedirectory.biz/search-engine-histories/history-of-yahoo-search-engine/. Accessed: 10 October 2015.

Faculty 2015. History of PayPal. URL: https://faculty.ist.psu.edu/bagby/432Fall07/T7/history.html. Accessed: 25 October 2015.

Ejobindia 2014. Justify the Reason to Pursue PHP. URL: http://www.ejobindia.com/blog/justify-the-reason-to-pursue-php/. Accessed: 27 October 2015.

Faculty 2015. How PayPal Works. URL: https://faculty.ist.psu.edu/bagby/432Fall07/T7/howWorks.html. Accessed: 25 October 2015.

Felke-Morris, T. 2013. Web Development and Design Foundations with HTML5. 6th ed. Pearson Education. Harlow.

Fernandes, J. 2014. Types of e-commerce. URL: http://bloomidea.com/en/blog/types-e-commerce. Accessed: 22 October 2015.

Fote, W. 2013. What is Ajax and Where is it Used in Technology? URL: http://www.seguetech.com/blog/2013/03/12/what-is-ajax-and-where-is-it-used-in-technoogy. Accessed: 26 October 2015.

Google Company 2014. Our history in depth. URL: https://www.google.com/about/company/history/. Accessed: 3 October 2015.

Grabianowski, E. & Crawford, S. 2015. How PayPal Works. URL: http://money.howstuffworks.com/paypal.htm. Accessed: 25 October 2015.

Helander, A. 2015. Building a WordPress site for Sports Academy of Kouvola. Bachelor’s Thesis. Haaga-Helia University of Applied Sciences. URL: http://urn.fi/URN:NBN:fi:amk-2015052610509. Accessed: 20 October 2015.

Jaakkola, S. 2012. Implementation of Drupal e-commerce website – Case: T:mi Jane Palmu Design. Bachelor’s thesis. Haaga-Helia University of Applied Sciences. Helsinki.

Jelastic 2014. Software Stacks Market Share: April 2014. URL: http://blog.jelastic.com/2014/05/20/software-stacks-market-share-april-2014/. Accessed: 27 October 2015.

Jquery-tutorial.net 2015. What is jQuery? URL: http://www.jquery-tutorial.net/introduction/what-is-jquery/. Accessed: 26 October 2015.

Khurana, A. 2015. Understanding the Different Types of Ecommerce Businesses. URL: http://ecommerce.about.com/od/eCommerce-Basics/a/Types-Of-Ecommerce.htm. Accessed: 22 October 2015.

Laudon, K. & Laudon, J. 2012. Management Information Systems - Managing The Digital Firm. 12th ed. Pearson Education. Harlow.

Le, V. 2014. Developing an activity-based costing system, case company X. Bachelor's thesis. Haaga-Helia University of Applied Sciences. Helsinki.

Marron 2015. The History of E-commerce: Past and Future. URL: http://bytecoiner.org/the-history-of-e-commerce-past-and-future/. Accessed: 22 October 2015.

Mozilla Developer Network 2015. Introduction. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction. Accessed: 26 October 2015.

Munir, F. 2015. Development of Online Product Management System – Case: Next Trading Company. Bachelor's Thesis. Haaga-Helia University of Applied Sciences. URL: http://urn.fi/URN:NBN:fi:amk-2015060312085. Accessed: 02 November 2015.

Neupane, S. 2015. Developing a financial simulation tool, as web application. Bachelor's Thesis. Haaga-Helia University of Applied Sciences. URL: http://urn.fi/URN:NBN:fi:amk-2015101115336. Accessed: 23 October 2015.

PayPal About 2015. Who we are. URL: https://www.paypal.com/us/webapps/mpp/about. Accessed: 3 October 2015.

PayPal 2015. URL: https://www.paypal.com/. Accessed: 25 October 2015.

PayPal 2015. PayPal account types. URL: https://www.paypal.com/us/webapps/helpcenter/helphub/article/?solutionId=FAQ2347. Accessed: 25 October 2015.

PayPal 2015. Adding PayPal Checkout to Your 3rd-party Shopping Cart. URL: https://www.paypal.com/cgi-bin/webscr?cmd=p/pdn/howto_checkout-outside. Accessed: 05 November 2015.

PowerMapper Software 2015. HTML Version Statistics. URL: http://try.powermapper.com/Stats/HtmlVersions. Accessed: 27 October 2015.

Richardson, T. & Thies, C. 2013. Multimedia Web Design And Development – Using Languages to Build Dynamic Web Pages. Mercury Learning And Information. Dulles.

Rouse, M. 2015. E-commerce (electronic commerce or EC) definition. URL: http://searchcio.techtarget.com/definition/e-commerce. Accessed: 22 October 2015.

Rouse, M. 2015. Electronic Data Interchange (EDI) definition. URL: http://searchdatacenter.techtarget.com/definition/EDI. Accessed: 22 October 2015.

Statista 2015. B2B e-commerce volume in the United States from 2006 to 2013. URL: http://www.statista.com/statistics/239366/us-b2b-e-commerce-volume-since-2006/. Accessed: 3 October 2015.

Statista 2015. B2C e-commerce sales worldwide from 2012 to 2018 (in billion U.S. dollars). URL: http://www.statista.com/statistics/261245/b2c-e-commerce-sales-worldwide/. Accessed: 3 October 2015.

Statista 2015. Bitcoin price index from May 2013 to September 2015 (in U.S. dollars). URL: http://www.statista.com/statistics/326707/bitcoin-price-index/. Accessed: 3 October 2015.

Statista 2015. Statistics and facts on Bitcoin. URL: http://www.statista.com/topics/2308/bitcoin/. Accessed: 3 October 2015.

Statista 2015. Statistics and facts about PayPal. URL: www.statista.com/topics/2411/paypal/. Accessed: 25 October 2015.

Statista 2015. PayPal's annual payment volume from 2012 to 2014 (in billion U.S. dollars). URL: http://www.statista.com/statistics/419783/paypals-annual-payment-volume/. Accessed: 25 October 2015.

TechnologyUK 2015. The Waterfall Model. URL: http://www.technologyuk.net/computing/sad/waterfall_model.shtml. Accessed: 03 November 2015.

The jQuery Foundation 2015. URL: https://jquery.com/. Accessed: 26 October 2015.

The PHP Group 2015. History of PHP. URL: http://php.net/manual/en/history.php.php. Accessed: 27 October 2015.

Ullman, L. 2014. Effortless E-Commerce with PHP and MySQL. 2nd ed. Pearson Education. Harlow.

W3school 2015. HTML and XHTML. URL: http://www.w3schools.com/html/html_xhtml.asp. Accessed: 26 October 2015.

W3school 2015. HTML5 Introduction. URL: http://www.w3schools.com/html/html5_intro.asp. Accessed: 26 October 2015.

W3C 2015. HTML & CSS. URL: http://www.w3.org/standards/webdesign/htmlcss. Accessed: 26 October 2015.

W3C 2015. A Short History of JavaScript. URL: https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript. Accessed: 26 October 2015.

W3Techs 2015. Usage of server-side programming languages for websites. URL: http://w3techs.com/technologies/overview/programming_language/all. Accessed: 27 October 2015.

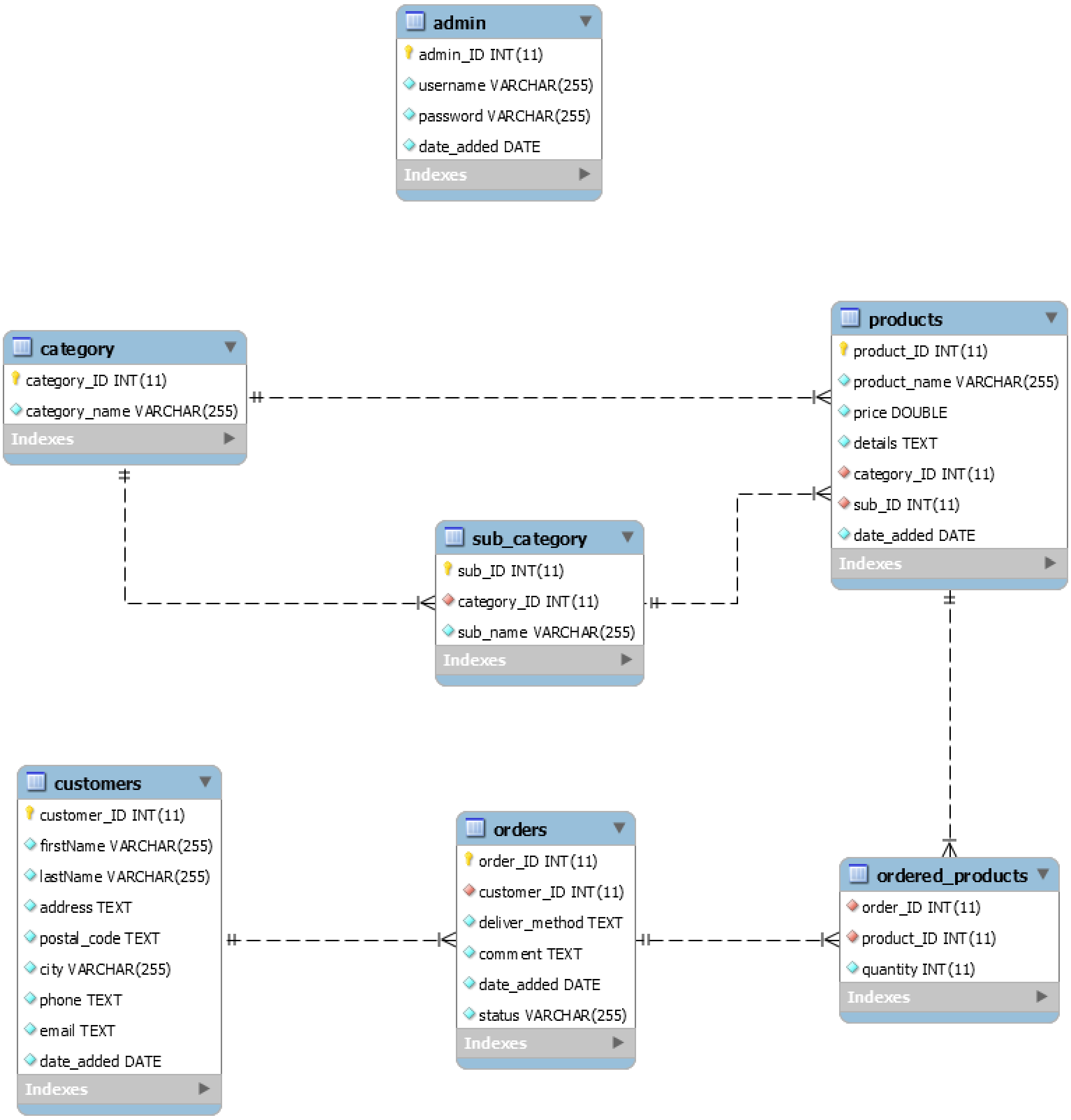
Wampserver 2015. Start with WampServer. URL: http://www.wampserver.com/en/. Accessed: 29 October 2015.

Whiteley, D. 2013. An Introduction to Information Systems. Palgrave Macmillan. Basingstoke.

Appendices

Appendix 1. Relational database

1. Relational database diagram



2. SQL queries for creating tables

Create table “admin” for administrative accounts

```
CREATE TABLE IF NOT EXISTS `ecommerce_db`.`admin` (
  `admin_ID` INT(11) NOT NULL AUTO_INCREMENT,
  `username` VARCHAR(255) NOT NULL,
  `password` VARCHAR(255) NOT NULL,
  `date_added` DATE NOT NULL,
  PRIMARY KEY (`admin_ID`))
ENGINE = InnoDB
AUTO_INCREMENT = 2
```

```
DEFAULT CHARACTER SET = latin1
```

Create table “category”

```
CREATE TABLE IF NOT EXISTS `ecommerce_db`.`category` (
  `category_ID` INT(11) NOT NULL AUTO_INCREMENT,
  `category_name` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`category_ID`),
  INDEX `cateogry_ID` (`category_ID` ASC))
ENGINE = InnoDB
AUTO_INCREMENT = 4
DEFAULT CHARACTER SET = latin1
```

Create table “sub_category”

```
CREATE TABLE IF NOT EXISTS `ecommerce_db`.`sub_category` (
  `sub_ID` INT(11) NOT NULL AUTO_INCREMENT,
  `category_ID` INT(11) NOT NULL,
  `sub_name` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`sub_ID`),
  INDEX `category_ID` (`category_ID` ASC),
  CONSTRAINT `sub_category_ibfk_1`
    FOREIGN KEY (`category_ID`)
    REFERENCES `ecommerce_db`.`category` (`category_ID`))
ENGINE = InnoDB
AUTO_INCREMENT = 10
DEFAULT CHARACTER SET = latin1
```

Create table “products”

```
CREATE TABLE IF NOT EXISTS `ecommerce_db`.`products` (
  `product_ID` INT(11) NOT NULL AUTO_INCREMENT,
  `product_name` VARCHAR(255) NOT NULL,
  `price` DOUBLE NOT NULL,
  `details` TEXT NOT NULL,
  `category_ID` INT(11) NOT NULL,
  `sub_ID` INT(11) NOT NULL,
  `date_added` DATE NOT NULL,
  PRIMARY KEY (`product_ID`),
  INDEX `category_ID` (`category_ID` ASC),
  INDEX `sub_ID` (`sub_ID` ASC),
  CONSTRAINT `products_ibfk_1`
    FOREIGN KEY (`category_ID`)
    REFERENCES `ecommerce_db`.`category` (`category_ID`),
  CONSTRAINT `products_ibfk_2`
    FOREIGN KEY (`sub_ID`)
    REFERENCES `ecommerce_db`.`sub_category` (`sub_ID`))
ENGINE = InnoDB
AUTO_INCREMENT = 19
DEFAULT CHARACTER SET = latin1
```

Create table “customer”

```
CREATE TABLE IF NOT EXISTS `ecommerce_db`.`customer` (
  `customer_ID` INT(11) NOT NULL AUTO_INCREMENT,
  `firstName` VARCHAR(255) NOT NULL,
  `lastName` VARCHAR(255) NOT NULL,
  `address` TEXT NOT NULL,
  `postal_code` TEXT NOT NULL,
  `city` VARCHAR(255) NOT NULL,
  `phone` TEXT NOT NULL,
  `email` TEXT NOT NULL,
  `date_added` DATE NOT NULL,
  PRIMARY KEY (`customer_ID`))
ENGINE = InnoDB
AUTO_INCREMENT = 6
DEFAULT CHARACTER SET = latin1
```

Create table “order”

```
CREATE TABLE IF NOT EXISTS `ecommerce_db`.`order` (
  `order_ID` INT(11) NOT NULL AUTO_INCREMENT,
  `customer_ID` INT(11) NOT NULL,
  `deliver_method` TEXT NOT NULL,
  `comment` TEXT NOT NULL,
  `date_added` DATE NOT NULL,
  `status` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`order_ID`),
  INDEX `customer_ID` (`customer_ID` ASC),
  CONSTRAINT `order_ibfk_1`
    FOREIGN KEY (`customer_ID`)
    REFERENCES `ecommerce_db`.`customer` (`customer_ID`))
ENGINE = InnoDB
AUTO_INCREMENT = 6
DEFAULT CHARACTER SET = latin1
```

Create table “ordered_products”

```
CREATE TABLE IF NOT EXISTS `ecommerce_db`.`ordered_products` (
  `order_ID` INT(11) NOT NULL,
  `product_ID` INT(11) NOT NULL,
  `quantity` INT(11) NOT NULL,
  INDEX `order_ID` (`order_ID` ASC, `product_ID` ASC),
  INDEX `product_ID` (`product_ID` ASC),
  CONSTRAINT `ordered_products_ibfk_1`
    FOREIGN KEY (`order_ID`)
    REFERENCES `ecommerce_db`.`order` (`order_ID`),
  CONSTRAINT `ordered_products_ibfk_2`
    FOREIGN KEY (`product_ID`)
    REFERENCES `ecommerce_db`.`products` (`product_ID`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
```

Appendix 2. Figures from testing the web shop on online hosting service

URL for the testing web shop: http://mickeychu.netii.net/Mickey-Ecommerce

Last accessed: 05 November 2015

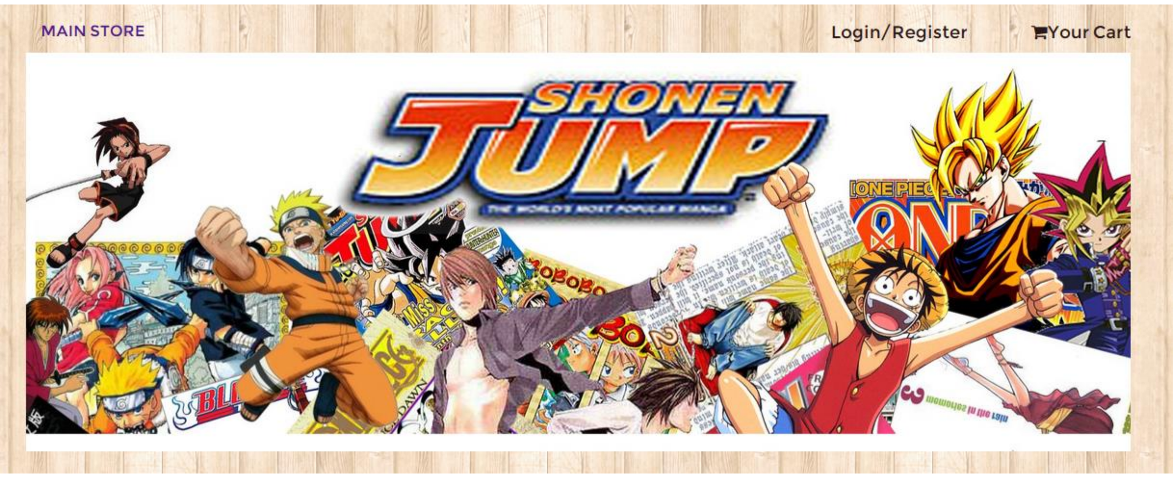


Figure 1. The header of the front page

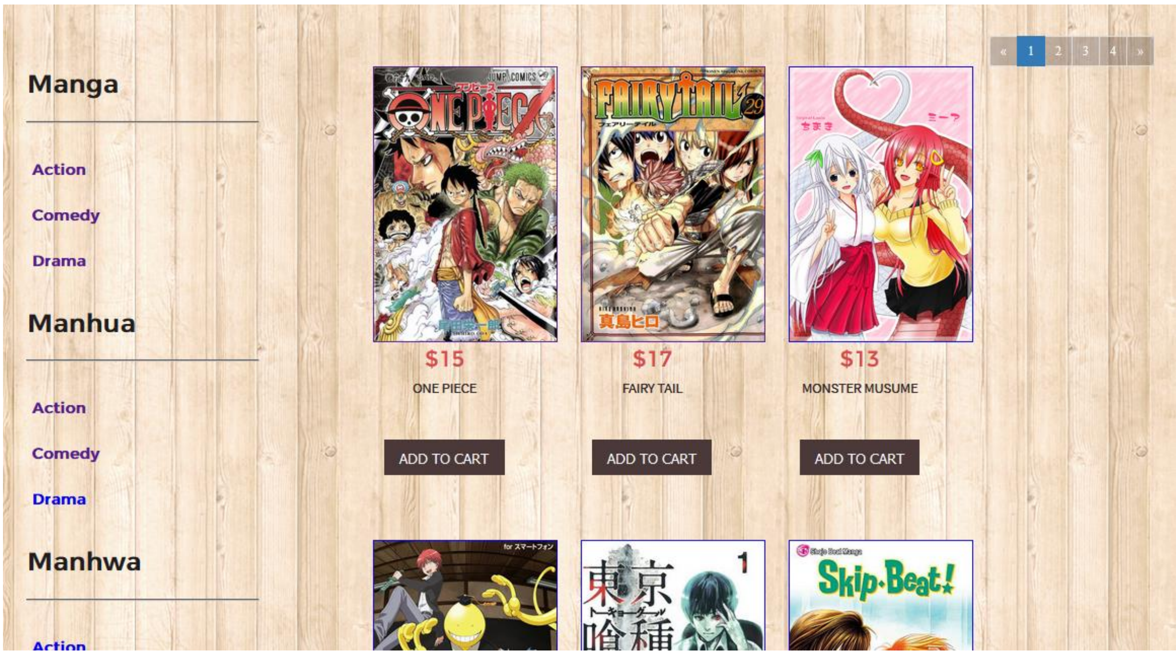


Figure 2. The content display on the front page

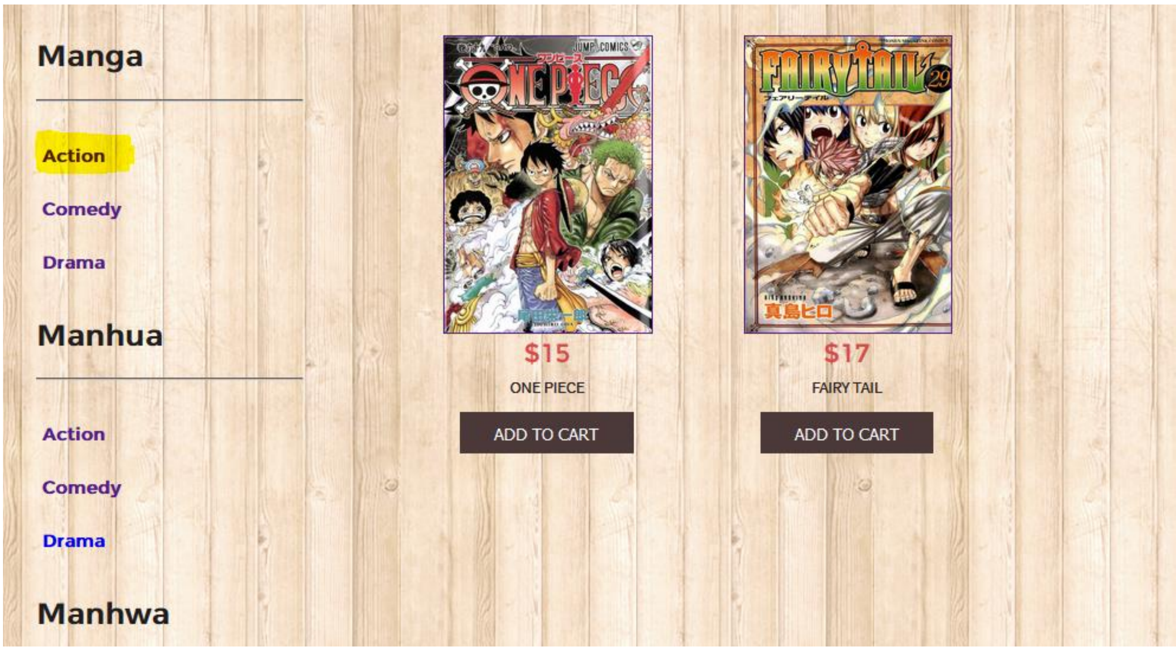
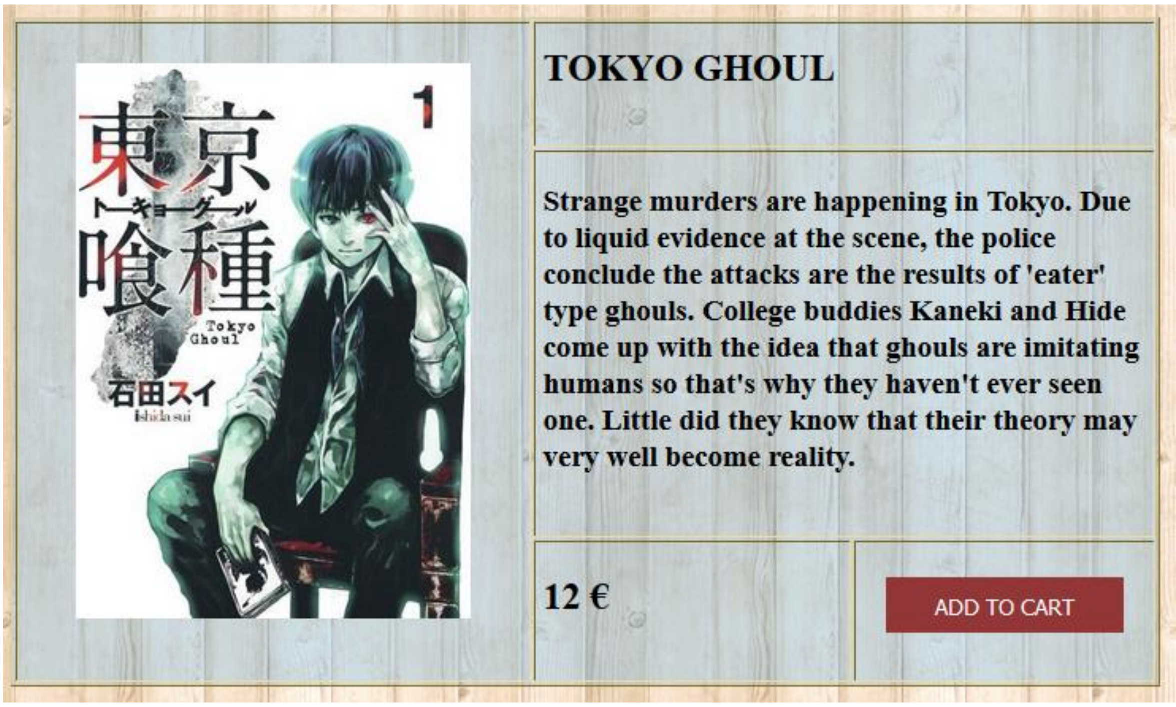


Figure 3. View products with the filter by sub_category “Action” under the main category “Manga”

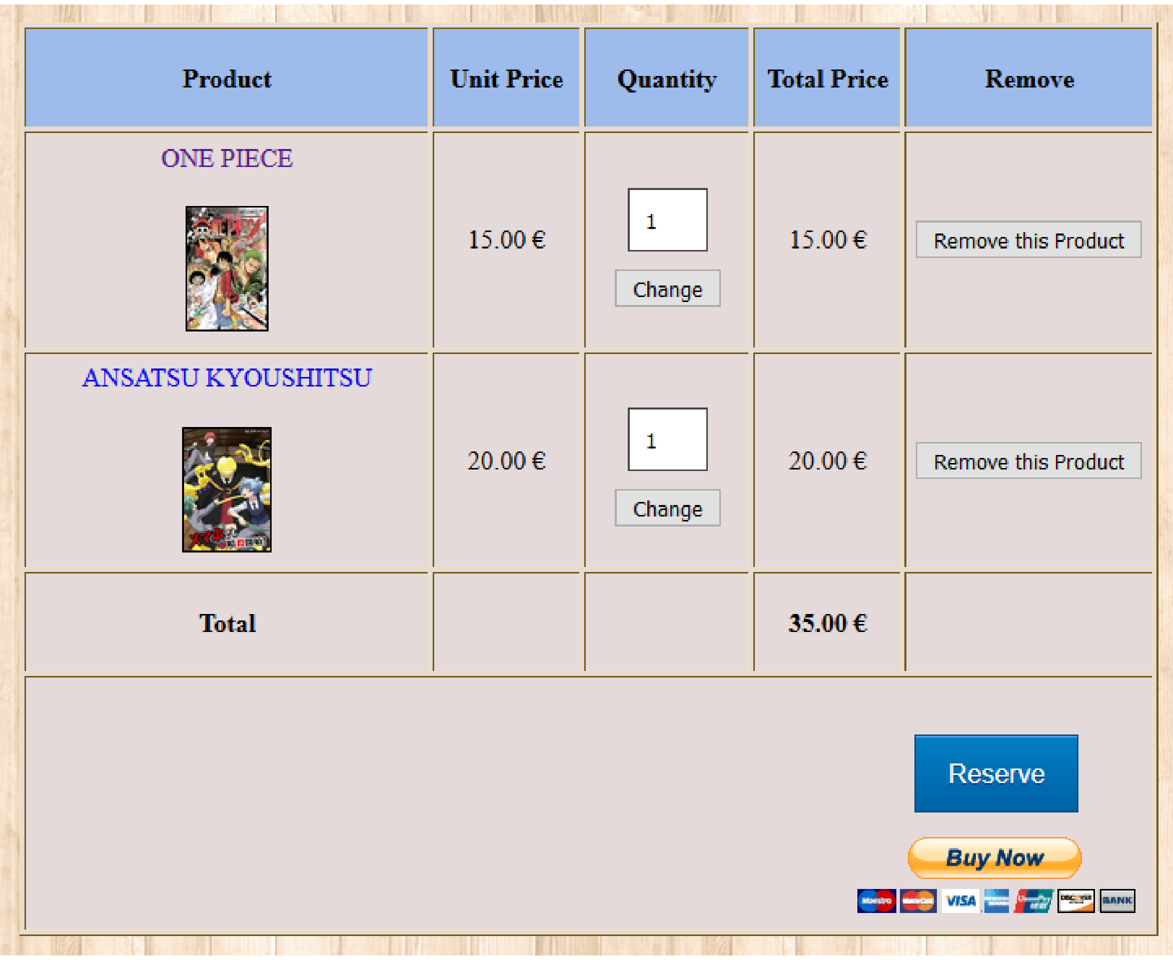


TOKYO GHOUL

Strange murders are happening in Tokyo. Due to liquid evidence at the scene, the police conclude the attacks are the results of 'eater' type ghouls. College buddies Kaneki and Hide come up with the idea that ghouls are imitating humans so that's why they haven't ever seen one. Little did they know that their theory may very well become reality.

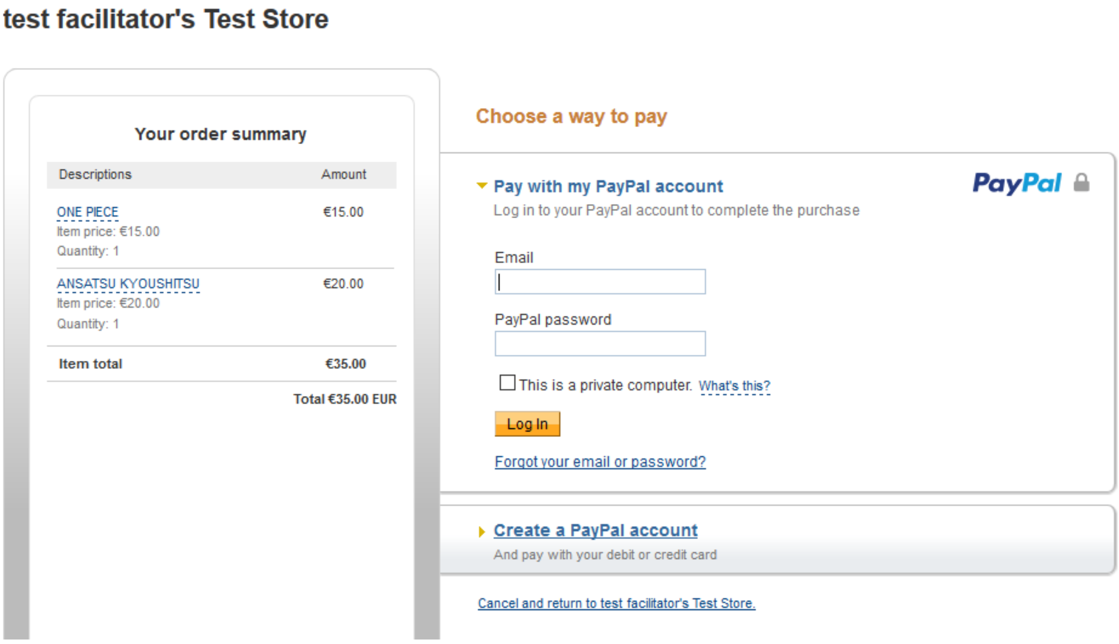
12 €

Figure 4. View details of a product



Product	Unit Price	Quantity	Total Price	Remove
ONE PIECE	15.00 €	1 Change	15.00 €	Remove this Product
ANSATSU KYOUSHITSU	20.00 €	1 Change	20.00 €	Remove this Product
Total			35.00 €	

Figure 5. The shopping cart with some products added.



test facilitator's Test Store

Your order summary

Descriptions	Amount
ONE PIECE Item price: €15.00 Quantity: 1	€15.00
ANSATSU KYOUSHITSU Item price: €20.00 Quantity: 1	€20.00
Item total	€35.00
	Total €35.00 EUR

Choose a way to pay

Pay with my PayPal account

Log in to your PayPal account to complete the purchase

Email

PayPal password

□ This is a private computer. What's this?

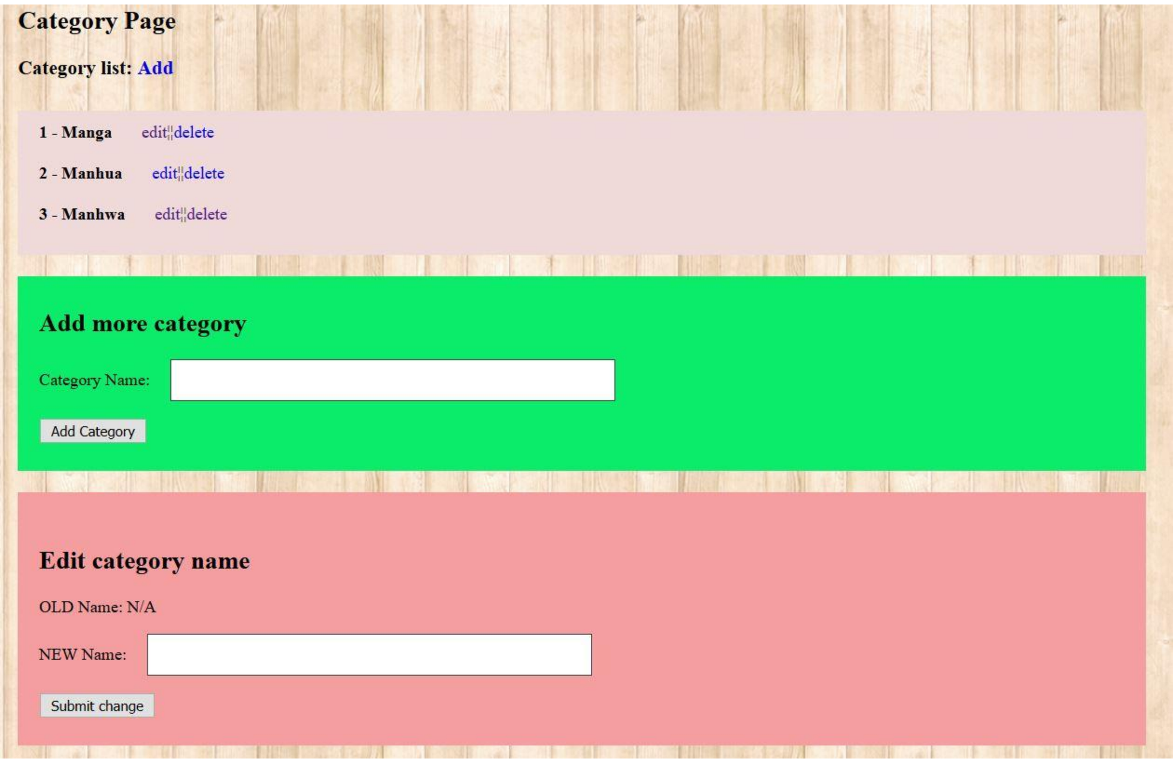
Forgot your email or password?

Create a PayPal account

And pay with your debit or credit card

Cancel and return to test facilitator's Test Store.

Figure 6. Check out the shopping cart using PayPal Sandbox


Category Page

Category list: Add

1 - Manga edit¦¦delete

2 - Manhua edit¦¦delete

3 - Manhwa edit¦¦delete

Add more category

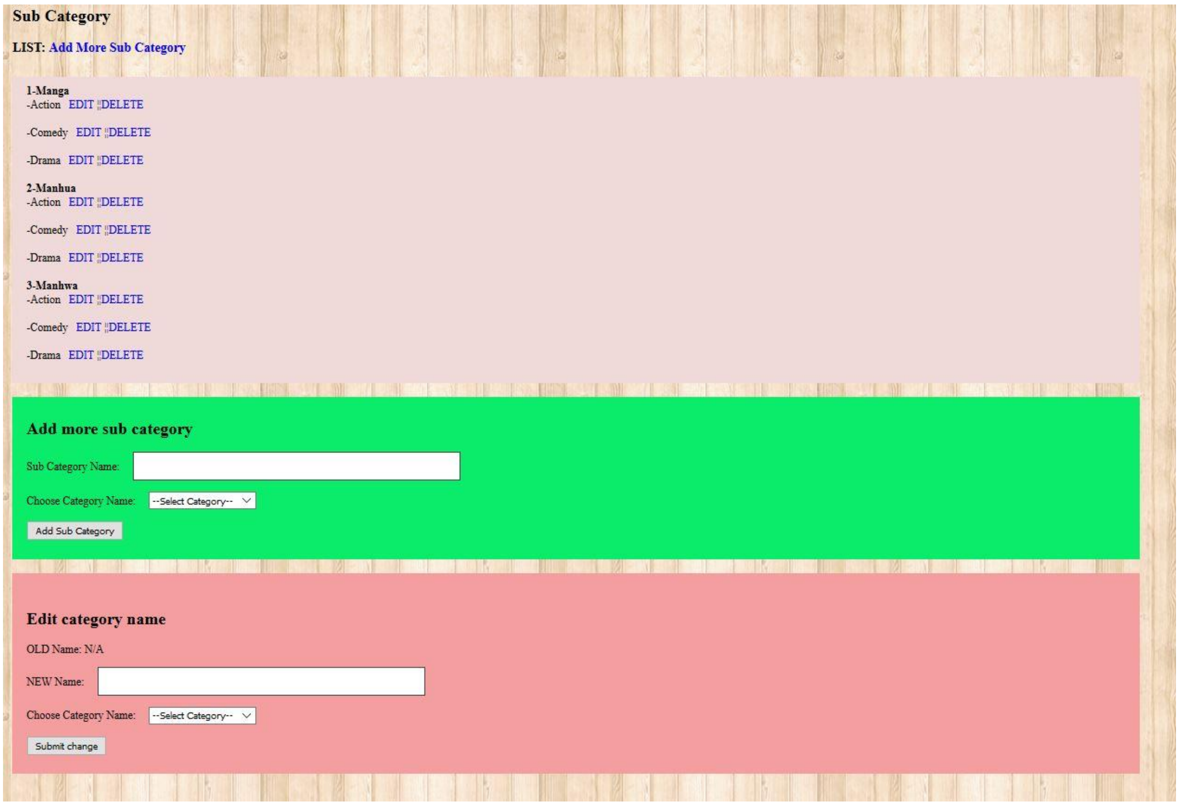
Category Name:

Edit category name

OLD Name: N/A

NEW Name:

Figure 7. Category page on the store management system


Sub Category

LIST: Add More Sub Category

1-Manga
-Action EDIT ¦¦DELETE
-Comedy EDIT ¦¦DELETE
-Drama EDIT ¦¦DELETE

2-Manhua
-Action EDIT ¦¦DELETE
-Comedy EDIT ¦¦DELETE
-Drama EDIT ¦¦DELETE

3-Manhwa
-Action EDIT ¦¦DELETE
-Comedy EDIT ¦¦DELETE
-Drama EDIT ¦¦DELETE

Add more sub category

Sub Category Name:

Choose Category Name: --Select Category--

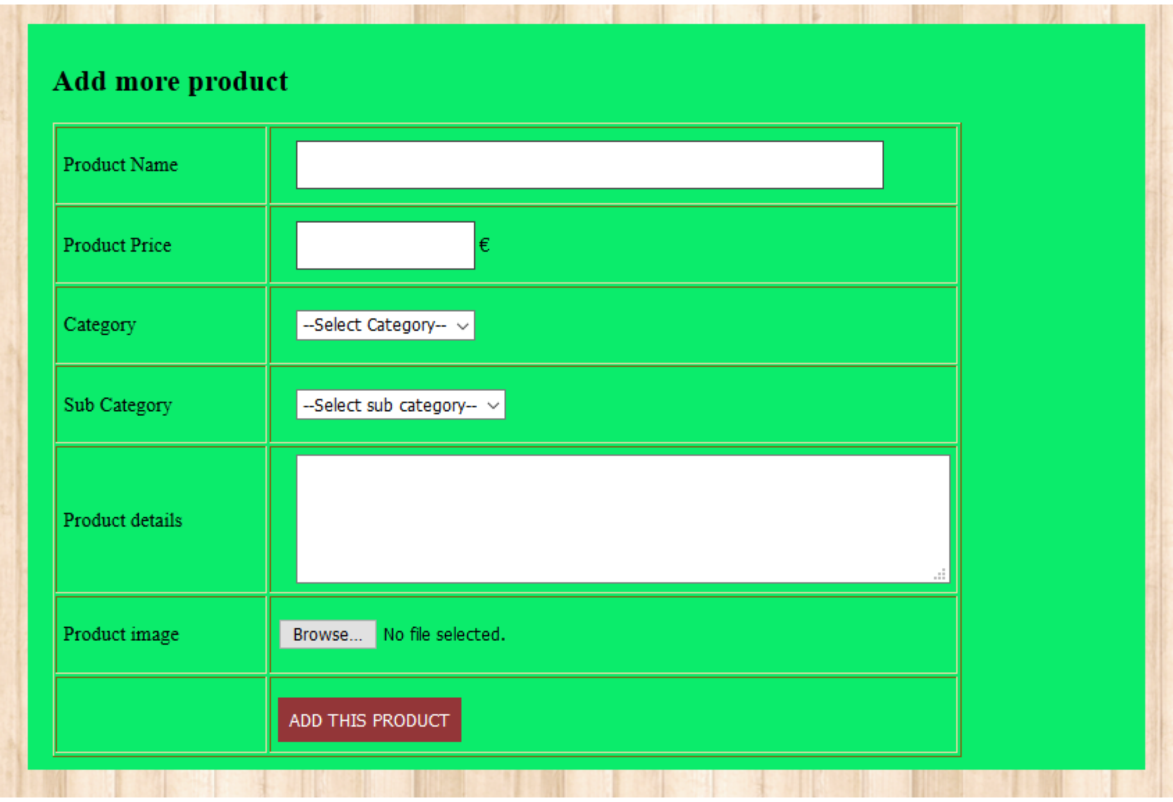
Edit category name

OLD Name: N/A

NEW Name:

Choose Category Name: --Select Category--

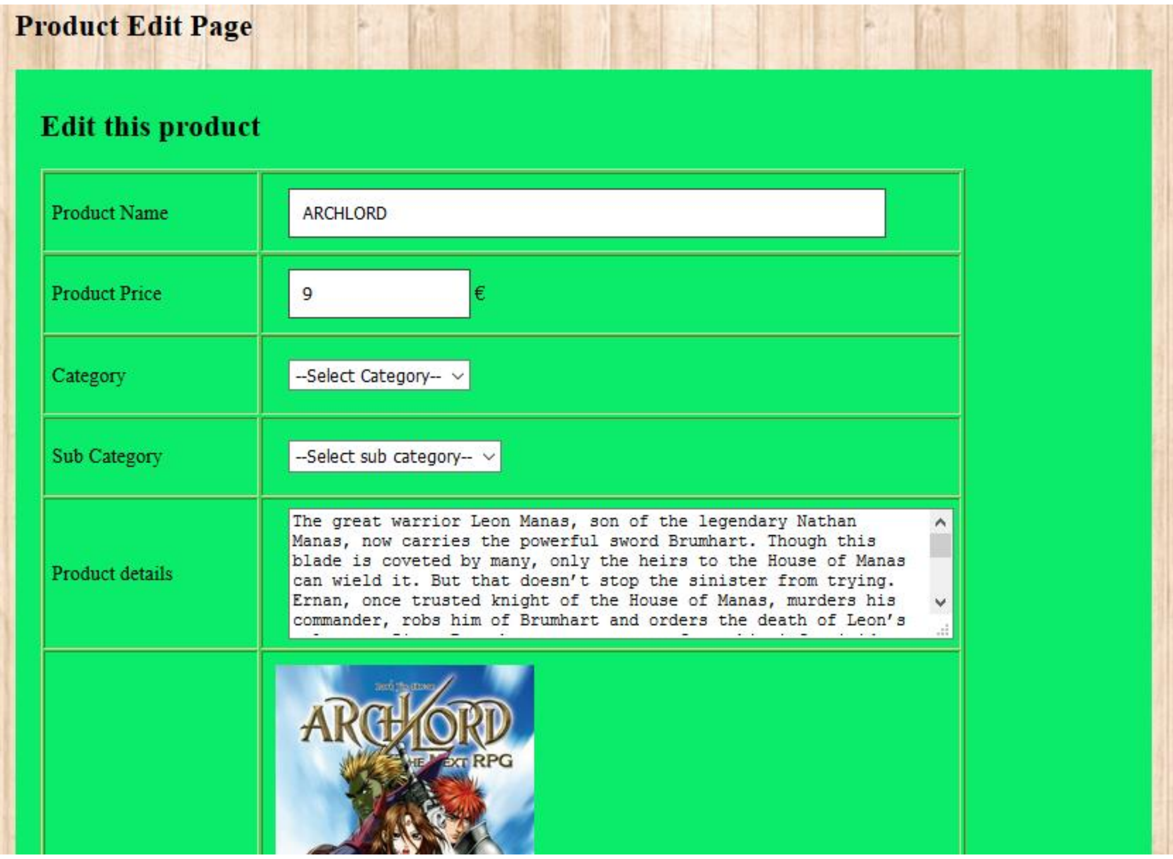
Figure 8. Sub_category page on the store management system



Add more product

Product Name	
Product Price	€
Category	--Select Category--
Sub Category	--Select sub category--
Product details	
Product image	Browse... No file selected.
	ADD THIS PRODUCT

Figure 9. Add product page on the store management system

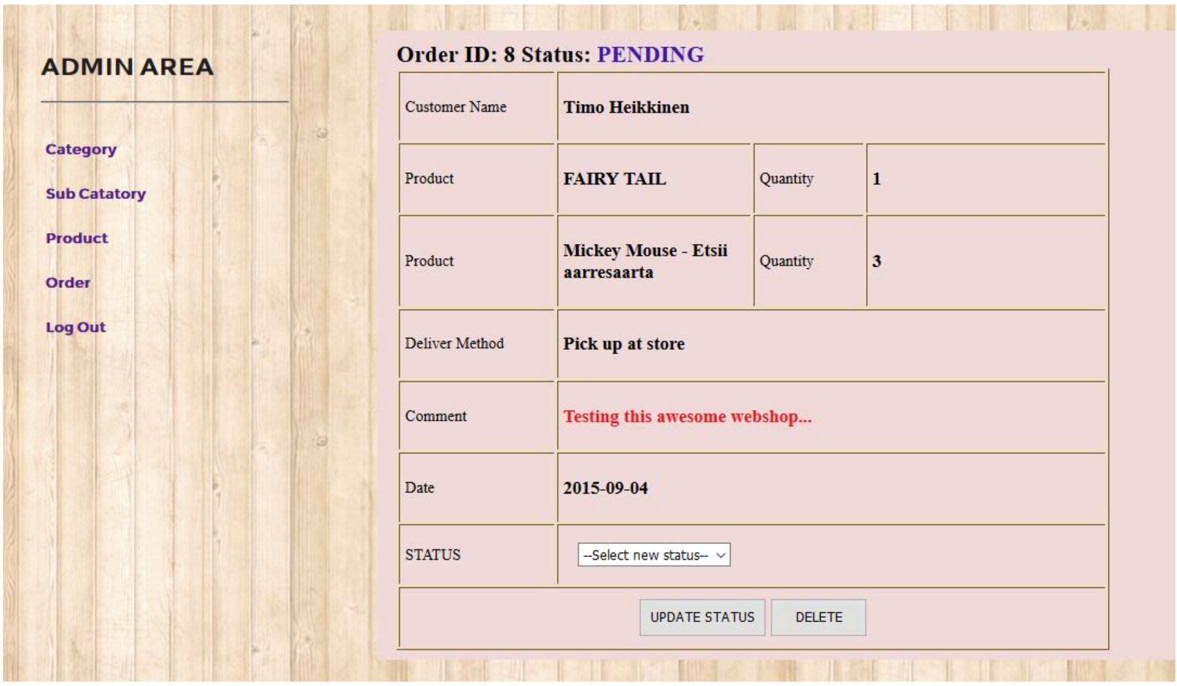


Product Edit Page

Edit this product

Product Name	ARCHLORD
Product Price	9 €
Category	--Select Category--
Sub Category	--Select sub category--
Product details	The great warrior Leon Manas, son of the legendary Nathan Manas, now carries the powerful sword Brumhart. Though this blade is coveted by many, only the heirs to the House of Manas can wield it. But that doesn't stop the sinister from trying. Ernan, once trusted knight of the House of Manas, murders his commander, robs him of Brumhart and orders the death of Leon's

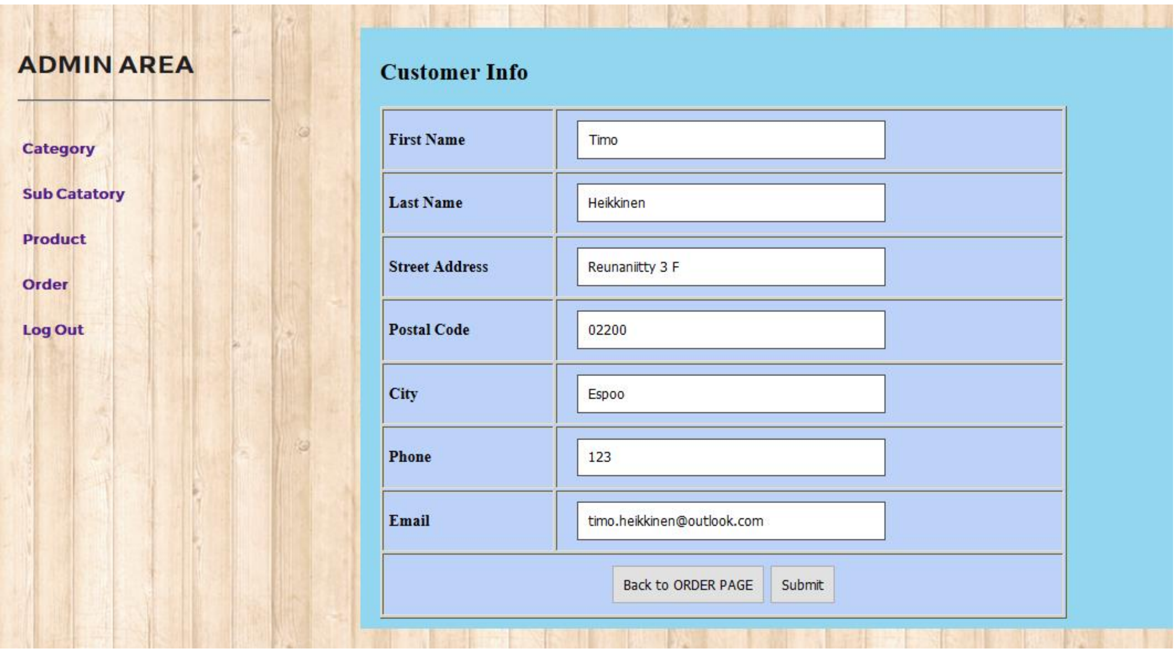
Figure 10. Edit product page on the store management system



Order ID: 8 Status: PENDING

Customer Name	Timo Heikkinen		
Product	FAIRY TAIL	Quantity	1
Product	Mickey Mouse - Etsii aarresaarta	Quantity	3
Deliver Method	Pick up at store		
Comment	Testing this awesome webshop...		
Date	2015-09-04		
STATUS	--Select new status--		

Figure 11. A detailed order view on the store management system



Customer Info

First Name	Timo
Last Name	Heikkinen
Street Address	Reunaniitty 3 F
Postal Code	02200
City	Espoo
Phone	123
Email	timo.heikkinen@outlook.com

Figure 12. Customer edit page on the store management system

Appendix 3. PayPal variables

(PayPal 2015.)

Name	Value
business	Email address on your PayPal account
quantity	Number of items. This will multiply the amount if greater than one
item_name	Name of the item (or a name for the shopping cart). Must be alphanumeric, with a 127character limit
item_number	Optional pass-through variable for you to track payments. Must be alpha-numeric, with a 127 character limit
amount	Price of the item (the total price of all items in the shopping cart)
shipping	The cost of shipping the item
shipping2	The cost of shipping each additional item
handling	The cost of handling
tax	Transaction-based tax value. If present, the value passed here will override any profile tax settings you may have (regardless of the buyer's location).
no_shipping	Shipping address. If set to "1," your customer will not be asked for a shipping address. This is optional; if omitted or set to "0," your customer will be prompted to include a shipping address
cn	Optional label that will appear above the note field (maximum 40 characters)
no_note	Including a note with payment. If set to "1," your customer will not be prompted to include a note. This is optional; if omitted or set to "0," your customer will be prompted to include a note.
on0	First option field name. 64 character limit
os0	First set of option value(s). 200 character limit. "on0" must be defined for "os0" to be recognized.
on1	Second option field name. 64 character limit
os1	Second set of option value(s). 200 character limit. "on1" must be defined for "os1" to be recognized.
custom	Optional pass-through variable that will never be presented to your customer. Can be used to track inventory
invoice	Optional pass-through variable that will never be presented to your customer. Can be used to track invoice numbers
notify_url	Only used with IPN. An internet URL where IPN form posts will be sent

return	An internet URL where your customer will be returned after completing payment
cancel_return	An internet URL where your customer will be returned after cancelling payment
image_url	The internet URL of the 150 X 50 pixel image you would like to use as your logo
cs	Sets the background color of your payment pages. If set to "1," the background color will be black. This is optional; if omitted or set to "0," the background color will be white