

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

MP3 E-COMMERCE APPLICATION

DONG YANG

A MAJOR REPORT
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE
CONCORDIA UNIVERSITY
MONTREAL, QUEBEC, CANADA

DECEMBER 2001

© DONG YANG, 2001



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-68489-X

Canada

Abstract

MP3 E-Commerce Application

Dong Yang

As an e-commerce application, the MP3 online shopping cart presents a general structure of a Web application. It is made up of a Web server, a server side application, and a database as backend. Also, it includes a simple search system, an online payment processing and secure transaction clearing.

This project will emphasize on setting up a Web server with SSL (Secure Sockets Layer Protocol) model, shopping cart design, database design, and secure online transaction handling. This shopping cart is built in the Apache Web Server with SSL model. Using the PHP4 program the server side application. The MySQL is used as its database.

Acknowledgements

I would like to thank Dr. Bipin C. Desai, my supervisor, for his continuous guidance, prompt support and valuable suggestions. Without his dedication and patience, this major report would not have been possible.

I am also grateful to Donglin Chen, who has rich working experience, for contributing many ideas to my work. His help has made the project more enjoyable and realistic.

Finally, I would like to thank my wife, Rong Hu, for her endless support.

Table of Contents

CHAPTER 1 INTRODUCTION	3
1.1 MP3 and MP3 Sites	3
1.2 MP3 Shopping Cart.....	4
CHAPTER 2 SETTING A WEB SERVER AND DEVELOPMENT ENVIRONMENT	7
2.1 Choosing development platform.....	7
2.2 Choosing Web server	8
2.3 Choosing server side language and database	15
2.4 Setting SSL into the Web Server	17
CHAPTER 3 DESIGN A MP3 ONLINE SHOPPING CART	24
3.1 Database Design.....	24
3.2 Shopping cart Design.....	30
3.2.1 The user view Design.....	30
3.2.2 The administrator view Design.....	38
CHAPTER 4 IMPLEMENTING THE MP3 ONLINE SHOPPING CART	41
4.1 Online catalog	42
4.2 User authentication	48
4.3 Shopping cart	52
4.4 Online payment integration.....	55
4.5 Administration	58
CHAPTER 5 CONCLUSION AND FUTURE WORK	62
5.1 Conclusion	62
5.2 Future work.....	62

CHAPTER 6	REFERENCE	63
-----------	-----------------	----

Chapter 1 Introduction

E-Commerce, Electronic Commerce, is the buying and selling of products and services by businesses and consumers over the internet. Typically there are three types of E-commerce transactions: B2B (business to business) (i.e. www.Cisco.com), B2C (business to consumer) (i.e. retailers), and C2C (consumer to consumer) (i.e. www.eBay.com). This project presents a B2C E-commerce application – MP3 online shopping cart.

1.1 MP3 and MP3 Sites

Normally, the music is stored as tracks in the CD-ROM or wave files. The wave file is standard the audio signal format. MP3 standards for MPEG1 layer 3. Normally, if a high quantity 12 minutes length music is stored as .wav file on the media, its size will be about 60M. After compression, this music can be saved as MP3 format and its size is only 3.5M. Besides it can keep the original music's quality. For example, the high quality original stereo music has several features that include 44 bits, 170 kb/s speed. In the MP3 format, it almost reaches these original indexes so that MP3 music could keep the high quality stereo music. [1]

Since MP3 music format was introduced several years ago, it makes it possible for the high quality and full length music to be transferred over the Internet. There are a huge number of Web sites distributing the MP3 music, including the big commercial sites such as www.napster.com, www.mp3.com, and www.audiosurge.com, etc. and many individual web sites. They are distributing almost all types of music. Anyone can find any music or song he loves on the Internet.

When the MP3 music is distributing wildly through the Internet, the copyright issue is becoming more and more important. MP3 itself is simply a file format. It is the content embodied in an MP3 file that makes it legal or illegal. Everyone can remember the lawsuit about Napster's copyright issue [2]. Since Napster.com appears on the Internet, people can freely download a variety of music. This makes most people happy, but it also leads to a Web law issue – copyright issue. This kind of lawsuit will be going on in the MP3 free world. All these issues make a legal MP3 site – an MP3 online shopping system necessary.

1.2 MP3 Shopping Cart

Shopping carts are essential to online store automation. A shopping cart provides customers with a "virtual" shopping solution where they can browse items for sell and place the ones they wish to purchase in their "shopping cart" until they are ready to "checkout".

The customer can view the contents of their cart, remove a specific product, or add an item to their cart at anytime by clicking on a link or button. Then when they are ready to purchase their items, they simply go to the "checkout" and pay for their purchases via a secure order form.

The shopping cart integrates electronic mail, Electronic Funds Transfer (EFT), Electronic Data Interchange (EDI), Internet, and similar techniques into a comprehensive, electronic-based system encompassing business functions such as grant application, procurement, payment, supply management, transportation and facility operations. [3]

A normal shopping will include all the tools needed to set up and operate a fully featured online store. It is made up of a Web Server (e.g. Apache, Apache Cocoon, IIS...), Server side programming (e.g. ASP, PHP, Perl, JSP...), and a Database as backend (e.g. MySQL, Oracle...). In this project, Apache, PHP4, and MySQL were used on the Linux platform.

For the PHP shopping cart, it will utilize PHP and other scripting technology (e.g. HTML, JavaScript), making it compatible with virtually every server and browser platform on the web. The following features should be included in a general shopping cart:

- Displays Items, Quantity, Item Sub Total, and Order Total in real time.
- Supports fractional unit pricing (pricing per unit of measurement).
- Products can have style menus, such as size, color, style, etc.
- Supports multiple shipping models (single, tiered, and aggregate).
- Supports standard, second day, overnight, and international shipping.
- Supports unlimited number of items and catalog pages.
- Includes search engine for unlimited product searches.
- Validates all entries, including credit card number and expiration date.
- Displays customized thank you page after order is submitted.
- Sends automatic e-mail order confirmation to customer and webmaster.
- Customers can store their profile to be used for future orders.
- Products can be added, modified and deleted from Database.
- Product pages can be designed and laid out in any fashion.
- Functions can be customized to specifications.

In this project, however, the shopping cart will hold the MP3 music. Some features have to be changed and some special features are needed. For example:

- Downloading and listening music sample will be allowed. It is very important, because customers will have the feeling that they were shopping in the real music store.
- Customers may download the MP3 music, which they bought, after payment since MP3 music is transferred over Internet.
- The quantity columns will be eliminated, since the MP3 music is stored in the file.
- The shipping choice will be eliminated. Customers will download the MP3 music file immediately after payment.
- The online payment transaction application, which can authorize transaction immediately, will be needed.

Although this MP3 shopping cart won't be used as practical shopping, it includes all the modules of the shopping system. The detailed design and implementation will be discussed later. An announcement about the music copyright is given in the Policy page.

In this report, the development of MP3 shopping cart road map will be presented. In Section 2, the setting of the shopping cart environment will be discussed, especially the SSL (Secure Sockets Layer). In Section 3, the database design and shopping cart module design will be covered. In Section 4, detailed module implementation will be presented. The PHP4's language special features will be covered too. The conclusion and future work on MP3 online shopping cart will be given in Section 5.

Chapter 2 Setting a Web server and development environment

The first step of building MP3 shopping system is environment development. Recently, there are many web servers can be chosen, such as IIS from Microsoft, Apache from the Apache Software Foundation, etc. Obviously, each of them has its own advantages. There are also several types of platform, such as Linux, Windows NT, and Solaris. As the server side programming language, there are PHP, Perl, and JSP etc. For the database engine, there are MySQL, MS-SQL, Oracle, etc. First, a Web server is built using these parts. Then a SSL server is set up to provide to secure transactions. Finally, the MP3 shopping cart can be deployed.

In this project, the MP3 shopping cart is built on Linux, Apache, PHP and MySQL with the SSL (Secure Sockets Layer protocol) model. The detailed information on installation of these can be easily found in their own manual. This section only discusses how to choose the system environment, and why choose them.

2.1 Choosing development platform

Whereas Windows operating system has the largest market share in PC computers, Linux has been accepted by PC's users. There are several differences between these two operating systems. On some aspects, Windows 2000 does better (i.e. GUI) and on the other aspects Linux does better (i.e. security).

In the case of Linux, since it is a UNIX-based kernel, it has reliability and security that comes with UNIX. It is open source so that bugs are fixed rapidly. One huge advantage

Linux has is that it is free. Windows requires licensing. Linux is getting to the point where it is getting close to compatibility with Windows platforms. It has large memory support now.[4]

On the contrary, the Windows has some mature aspects to it, where Linux is still getting up to speed. Linux, although it is based upon the 30-year-old basic operating principles of UNIX, has still only been around about 10 to 12 years as an operating system itself. With that in mind, there are a lot of ease-of-use features in Windows and administrative things in Windows that far outpace Linux today. Those more user-friendly aspects make Windows much more attractive for widespread deployment on desktops and servers.

In the project, the purpose is to build a Web Server for the shopping cart. This server should have high security and reliability. Everyone can remember, since Web entered into people's lives, the security has been a serious problem in the Windows NT (even in Windows 2000). Recently, The Red Code worm virus has attacked thousands IIS Web servers.[4] This trouble is enough for us to consider whether we use Windows and IIS to build the Web server. According to the shopping cart's need, Linux is a better choice than Windows.

2.2 Choosing Web server

There are two major Web servers on the market. They are Internet Information Server (IIS) from Microsoft and Apache Web Server from Apache Software Foundation. While IIS with Windows NT provides a friendly user interface for uses, the Apache with Linux is providing a free way to build a reliable Web server. With these two products, making a

decision is not easy. We have to compare them in detail so that we can choose one of them based on its advantages and the goal of the Web server. Here, the comparison will be focused on their Standards Support, Web Server Prerequisites, Development Model, Administration Considerations, Log Considerations, Process/Thread Considerations, and Security Considerations.

Standard Support

Standards support is very important in many government contracts and procurements. The following table 2.1 summarizes the standards support implemented by Apache and IIS [5].

Features	Apache	IIS	Comments
HTTP 1.1	Yes	Yes	
W3C's extended log format	Yes	Yes	
Simple Network Management Protocol (SNMP)	No	Yes	A commercial plug-in SNMP module is available for Apache from Covalent that provides real-time management information for server access statistics, activity, load and utilization as well as on-the-fly configuration changes. More information is available at

			http://www.covalent.net/products/snmp/ .
Internet Server Application Program Interface (ISAPI)	Yes	Yes	ISAPI extension modules are written by third parties and available as part of the Apache distribution. Apache supports ISAPI extensions but it does not support ISAPI filters.
WebDAV	Yes	Yes	Part of Apache 2.0 functionality. WebDAV, which stands for Web Distributed Authoring and Versioning, is a standard under development by W3C, for Web-based collaborative document development.
SSL	Restricted	Yes	A module is available for Apache that provides strong cryptography for the Apache 1.3 Web server SSL 2/3 and TLS 1 (Transport Layer Security 1) protocols. However, this module can be used only outside the United States for free. In the United States, you can use it for noncommercial purposes for free if you use RSAREF (because of various patents held by RSA).

Table 2.1 Comparison between Apache and IIS in the basic support

Web Server Prerequisites

A major difference between Apache and IIS is the prerequisites. While Apache is available for many UNIX versions, Linux, Windows and OS/2, IIS can run only in the Windows environment. What's more, the latest version of IIS will run only on Windows 2000 server. The table 2.2 shows a comparison between Apache and IIS.

Feature	Apache	IIS
OS dependency	Unix, Linux, Windows, OS/2	Windows
Hardware platform	Wide range of hardware supported by the different operating systems. including Intel and SPARC.	Those supported by Windows.

Table 2.2 Comparison between Apache and IIS in the Web server prerequisites

Development Model

Besides the fact that Apache is free if the user downloads the software, the other major difference between IIS and Apache is that the source for Apache is freely available. The free source model has increased the development of Apache functions through many volunteer programmers (and recently from IBM), resulting in a fairly rapid pace of functional enhancements. In addition, Apache is modular in structure, permitting Apache users to pick and choose modules to fit their requirements [6]. By comparison, user's ability to customize IIS functionality is limited. Apache's freely available source has also produced fairly quick bug fixes. Overall, Apache's development model has resulted in a robust, reliable Web server.

On the other hand, IIS is typical of many Microsoft Windows-based products. It supports the Windows GUI and provides integration with other Windows mechanisms, such as Active Directory, MS SQL Server and Windows Security services. IIS also leverages the benefits of COM+. On the downside, IIS requires additional Microsoft products, such as Windows 2000 or Active Directory [5]. This dependency causes a worry of potential lock in to Microsoft products and strategy for some companies.

Administration Considerations

Both Apache and IIS provide many built-in administration functions. These functions include:

- Web server administration using scripts for commonly performed functions.
- Support for multiple hardware virtual servers using separate IP addresses.
- The ability to manage multiple Web servers as one server.
- Support for multiple software virtual servers using host headers.

Unlike Apache, IIS provides GUI support for server administration. One of the projects under the auspices of the Apache Software Foundation is to develop a graphical configuration tool for Apache.

Besides administrative GUI support, IIS also provides the ability to administer one or more IIS Web servers from a Web browser. IIS also provides some common administration functions available with other Microsoft products, such as a setup wizard and unattended/remote setup and maintenance. IIS even provides integration with

Windows operating system functions such as directory services, backup and restore of IIS configuration information and the Windows performance monitor (to view IIS performance data).

Log Considerations

Both Apache and IIS can write to multiple logs using standard and extended log formats (from the W3C) and can even support custom logging [5]. They permit logging to text files. Both of them also let you automatically close an aging log file and open a new file based on elapsed time.

Here again, IIS provides better integration with Windows OS facilities. For example, IIS can write file security and application events to the Windows event log, which can then be viewed using standard Windows tools, such as Event Viewer. IIS also provides logging to any ODBC database. In fact, multiple IIS servers can log to a single ODBC database. Logging to an ODBC database in Apache is possible, but only through an add-on [5]. Likewise, modules are available outside the official Apache distribution that provides log file analysis, data distribution and so on.

Process/Thread Considerations

Both IIS and Apache provide isolation between different Web client access requests. Apache, in its current standard distribution, forks processes in response to incoming Web client requests. This approach tends to be more resource-intensive than spawning threads

within a process, which is the approach adopted by IIS. However, thread support will be one of the main enhancements to arrive with Apache 2.0.

Security Considerations

Both Apache and IIS support basic security features out of the box. These include:

- Restrict access by domain name
- Restrict access by user
- Restrict access by group
- Restrict access by directory and/or file
- Restrict access by IP address
- Basic HTTP user authorization with clear-text passwords

However, IIS provides the ability to integrate Web server security with Windows security features. Moreover, Windows 2000 can store its ACLs in Active Directory and support authentication using digital certificates (such as those that conform to the X.509 certificate standards) [6]. IIS can then leverage these security mechanisms directly.

Comparing Apache and IIS, Apache is a secure alternative to IIS. First, Apache doesn't install a lot of extra programs. A default Apache doesn't install any Apache modules (extensions) at all - just a basic web server. By default, Windows NT and IIS install seven external Dynamic Link Library (DLL) files plus FrontPage server extensions. Every one of these eight components has had security updates since Windows NT was shipped. Second, Apache components, if they are installed, run as privileged user, so if a buffer overflow occurs, damage is minimal. Conversely, Microsoft IIS allows system-level

access, thereby potentially granting root (super user) permission. Any user, even a remote one, who has root permission can access, change, and delete any file anywhere on the system. Finally, Apache gets all of its configurations from a single file, httpd.conf. Microsoft IIS gathers configuration data from several files.

For the Web server security reason, the Apache Web server is used in the shopping cart system.

2.3 Choosing server side language and database

After setting the Web server for shopping cart, choosing the server side language and database is another difficult issue because there are a couple of languages available for developing Web project, such as PHP, Perl, and Java servlets. After comparing these, there are enough reasons to choose PHP and MySQL.

PHP is a server side language designed specifically for the Web. It is freely provided in www.php.net. PHP is more competitive than others on the following aspects.

- **High performance:** When developers focus on language performance, developers can find PHP very efficient. Using a single inexpensive server, developers can serve millions of hits per day.
- **Interfaces to many different database systems:** It has native connections available to many database systems. In addition to MySQL, developers can directly connect to PostgreSQL, mSQL, Oracle, dbm, filePro, Hyperwave, Infomix, InterBase, and Sybase, among others. Furthermore, developers can connect to any database that provides an ODBC driver in the Microsoft's products.

- Built-in libraries for many common Web tasks: Because PHP was designed for use on the Web, it has many built-in functions for performing many useful Web-related tasks. You can generate GIF image on-the-fly, connect to other network services, send email, work with cookies, and generate PDF documents, all with just a few lines of codes.
- Low cost: It is free. Anyone can download the latest version at any time from www.php.net
- Ease of learning and use: PHP's syntax is based on other programming languages, primarily C and Perl. Developers can use PHP almost immediately, if developers have knowledge on C or Perl, or C-like languages such as C++, Java.
- Portability: PHP is available for many different operating systems. It can work with free UNIX like operating system such as Linux, commercial UNIX version such as Solaris, or different version of Microsoft Windows. Developers' code will usually work without any modification on the different system running PHP.
- Availability of source code: PHP provides free source code so that you can modify or add something into the language if developers want. The constant peer-review and the world-wide test team allows the PHP team to create a production quality PHP engine much faster than any closed source company could.

For the database, MySQL is a good choice to develop a Web application. Comparing it with other main type databases (such as PostgreSQL, Microsoft SQL Server, and Oracle), it has many strengths. They are as follows.

- High performance: MySQL is undeniably fast.

- Low cost: It is available at no cost, under an Open Source license, or at low cost under a commercial license if required for developers' application.
- Ease of use: Most modern databases use SQL. If you have used another RDMBS, developers should have no trouble adapting to MySQL. It is also easier to set up than many similar products.
- Portability: MySQL can be used on many different UNIX systems as well as under Microsoft Windows.

When PHP integrates with MySQL, it is simply language. There are lots of PHP tools to manage and maintain MySQL databases [7]. The feature set is complete compared to other databases. For example, there are very useful functions like `mysql_insert_id` (Get the id generated from the previous INSERT operation) and `mysql_affected_rows` (Get number of affected rows in previous MySQL operation) which are not available for other database products.

For MP3 shopping cart, first thing must be considered is Web server security. Then the low cost must be also considered. Therefore, PHP and MySQL combo is very good choice with their obvious strengths.

2.4 Setting SSL into the Web Server

For a shopping cart, it must provide an online secure transaction. This allows developer to implement a transaction between shopper and a user securely from end to end. In the MP3 shopping cart, for example, there are a couple of modules that need high security, such as the Registration module, in which user inputs private information and stores in

database and the check out module, in which the shopping cart shows user's private information. Consequently, using SSL (Secure Sockets Layer protocol) in the Web server is the most important.

The Secure Sockets Layer protocol suite was originally designed by Netscape to facilitate secure communication between Web server and Web browsers. It has since been adopted as the unofficial standard method for browsers and servers to exchange sensitive information.

A typical Web application, which is built by Linux, Apache, PHP, and MySQL, is shown in figure 2.1.

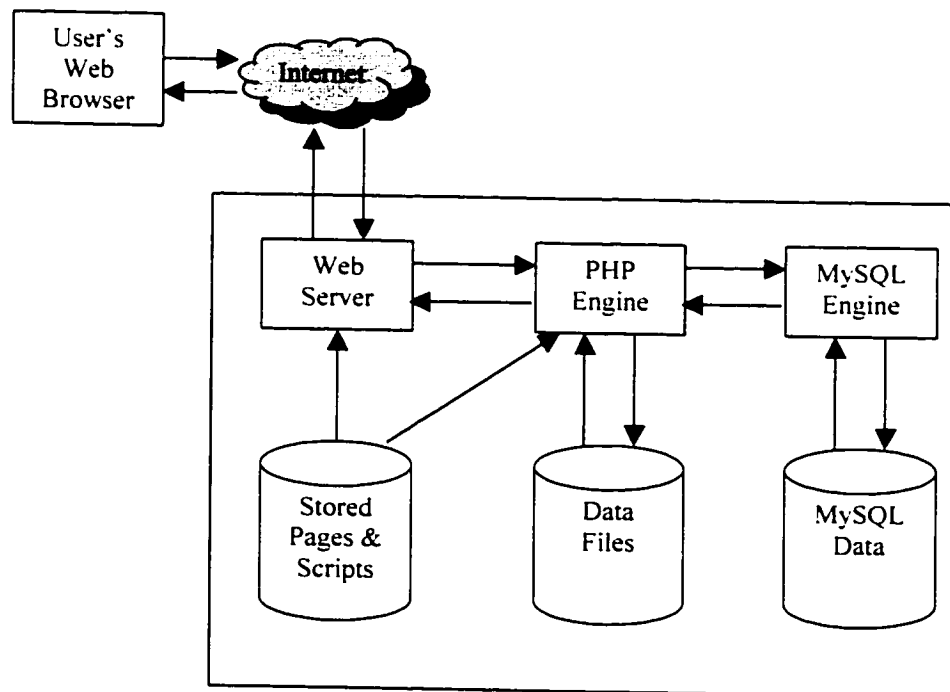


Figure 2.1 User information is stored or processed in a typical Web application

The details of each transaction occurring in the system will vary, depending both on the system design and on the user data and actions that triggered the transaction. As shown in

figure 2.1, the Web application system consists of three parts. They are the user's machine, the Internet, and the Web server. Obviously, in the shopping cart system, we can do nothing in the user's machine and Internet for secure transaction. Fortunately, both Netscape and Internet Explorer support SSL, and most Web servers include SSL functionality, or can accept it as an add-on module. We just put the SSL module to the Web server so that we can protect user's private information.

Networking protocol and software that implements them are usually arranged as a stack of layer. Its structure is illustrated in figure 2.2.

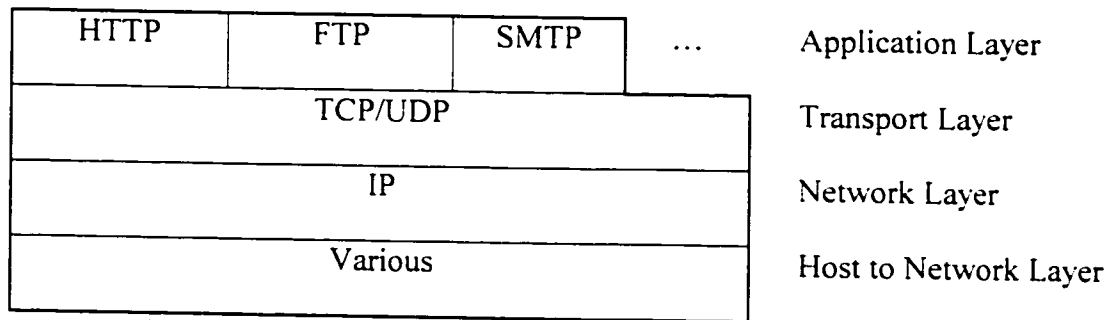
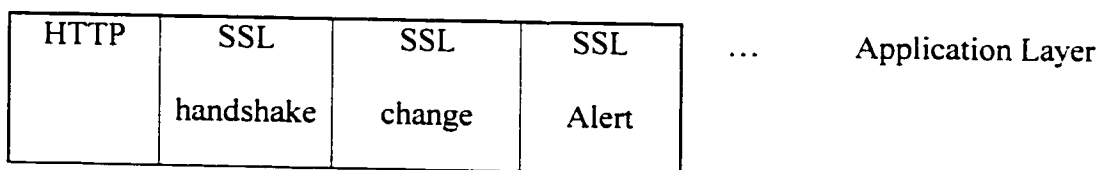


Figure 2.2 the protocol stack used by an application layer protocol

Each layer can pass data to the layer above or below, and request services of the layer above or below. When using HTTP transfer information, the HTTP protocol calls on the TCP, which in turn relies on the IP. This protocol in turn needs an appropriate protocol for the network hardware being used to take packets of data and send them as an electrical signal to the destination.

After adding the SSL as additional transparent layer to this model, the stack of layer will change to figure 2.3 [7].



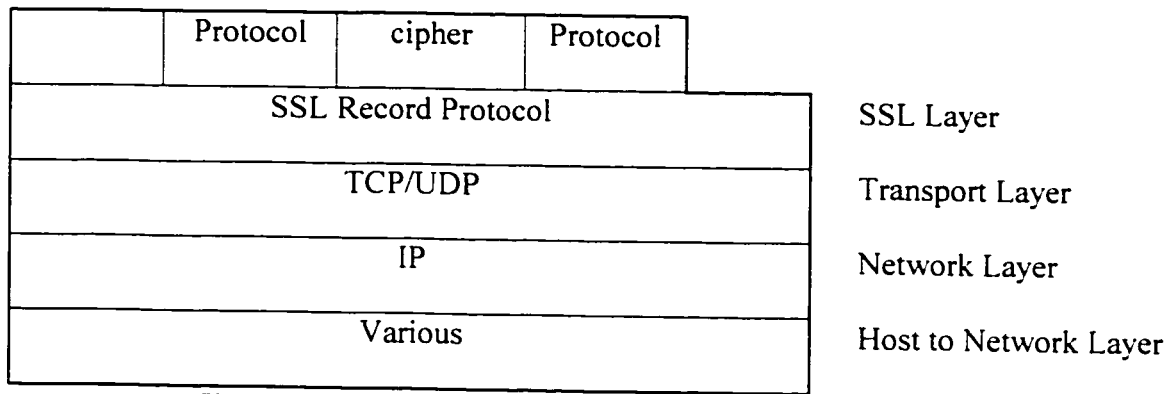


Figure 2.3 SSL adds an additional layer to the protocol stack

As figure 2.3 shows, the SSL layer exists between the transport layer and the application layer. It modifies the data from HTTP application before giving it to the transport layer to send it to its destination. The SSL is theoretically capable of providing a secure transaction environment for protocols other than HTTP. The SSL layer provides the same interface to protocols above it as the underlying transport layer. It then transparently deals with handshaking, encryption, and decryption.

When data is sent over an SSL connection, the following steps occur:

1. It is broken into manageable packets.
2. Each packet is (optionally) compressed.
3. Each packet has a message authentication code (MAC) calculated using a hashing algorithm.
4. The MAC and compressed data are combined and encrypted.
5. The encrypted packets are combined with header information and sent to the network.

In the user side, the typical web pages' links is http://. When their data will be handled as the secure transaction, users will see the web page link to https://. After adding the SSL into Web server, the Web application system processing is also changed to figure 2.4.

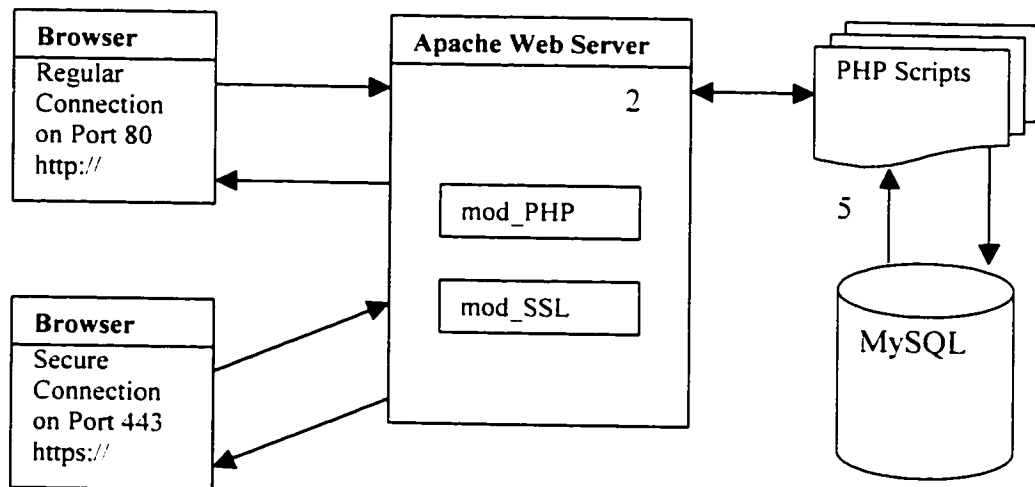


Figure 2.4 Web Application processing after adding the SSL into Web server

Figure 2.4 illustrates a Web application processing with SSL model. Here, the MP3 shopping cart will be used as an example to explain this processing.

1. The user clicks on a link to MP3 shopping cart from his web browser; his web browser sends the request for <http://dumbo.concordia.ca/shopsnd/topsale.php>.
2. Apache server at Dumbo gets this request for topsale.php. It knows that .php files are handled by PHP preprocessor (mod_PHP). Because mod_PHP has been specified in the Apache configuration file, it tells PHP to deal with this request.
3. topsale.php is a PHPscript that contains commands. One of these commands is to open a connection to a database and grab data. PHP handles the connection to the database, and interprets the SQL calls to extract data from the database.

4. The database gets the connection requests from the PHP interpreter, and processes the request. The request could be something like a simple select statement, or update statement.
5. The database then sends the response and results back to PHP interpreter.
6. Apache sends the PHP information back to the user's web browser, as the response to his request. The user now sees a web page containing information from a database.

However, some links in the MP3 shopping cart are in secure connection because those pages contain the information that need protection, such as Registration, Login, change password, update user's profile, and Checkout, etc. For example, if the user click on Registration, his web browser will send the request for https://dumbo.concordia.ca/shopsnd/member_register.php, then the process would be different.

7. Every request and response is encrypted and decrypted at both ends. That is, the web browser connects to Apache server, obtains its public encryption key, encrypts the request and sends it over.
8. The Apache server sees the request decrypts it and authenticates it. It processes the file, encrypts it and sends it over. The web browser then decrypts it with the server's key. Since the connections are encrypted, different ports are used. Port 80 is used in the non-secure connection, while Port 443 is used in the secure connection.

Although the Web server, Server side language, database, and SSL are combined, the real secure online transaction also includes other items that have to be done, such as secure storage, web page organization, and online payment strategy. These context designs are discussed in next chapter.

Chapter 3 Design a MP3 online shopping cart

The term online shopping cart is used to describe a specific online shopping mechanism. As figure 2.4 illustrates, a real shopping cart system should include server side application and MySQL besides Apache server. Combined PHP scripts and HTML make shopping cart's GUI and functions in the server side program. The database stores the music catalog and others data in different tables. The following two sub-sections will present the database design and shopping cart design.

3.1 Database Design

MySQL, as back end of the MP3 shopping cart, stores the music online catalog and user's information. The online catalog keeps all the information of the music. Users can search music in the database. The shopping cart allows a user to register; therefore, a database is needed to keep the user's information so that users can purchase their music without inputting the private information again. Moreover, it should also keep the transaction information and the administrator's information, such as username, password.

In order to implement all modules in the shopping cart, four tables will be created. They are:

- **musicshop:** It stores the MP3 music information, such as title, artist, subject, genre, etc..
- **transaction:** It stores the transaction information, such as username, music id, purchase date.
- **userinfo:** It keeps the user's private information, such as name, address, phone number, username, password, etc.

- administrator: It stores the administrator's username and password.

musicshop table

Unlike the normal shopping cart (e.g. book store), MP3 shopping cart needs to deal with issue of storing the MP3 file, which is binary file. This issue should be considered when creating the musicshop table. When facing this problem, there are two solutions.

One solution is to store the MP3 binary data into the database directly. It is the easiest solution. Like most database, MySQL has the ability of storing binary data. However, it makes a lot of trouble for the shopping cart. Although the size of an MP3 file is much smaller than the original .wav file, it is at least 3MB. A music shop generally keeps thousands of various music CDs (One CD has average 50 – 60 minutes music.). While compressing this huge number music to MP3 files, the total size will be over 50 GB. This won't be accepted by MySQL. On the other hand, since users will download the full length MP3 after payment processing, if the database stores the MP3 binary data, users will have to retrieve huge amount data from database while occupying bandwidth of Web server. These lead to limited number of active users. Therefore, this solution is not suitable for MP3 online shopping system.

Another solution is that MP3 files are stored in a separate. The database only keeps the reference of the each MP3 file. The reference is the exact path of the MP3 file in the server. Typically, the path is not over 100 characters; therefore, the MySQL database has enough ability to deal with the MP3 online store. The other aspect, when users download

MP3 music. they only interact with Web server instead of database. This strategy is shown in figure 3.1.

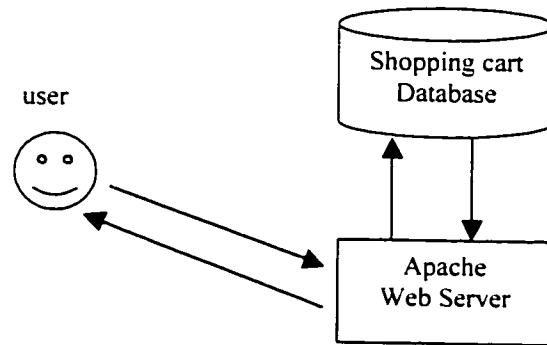


Figure 3.1 The user download the MP3

1. The user sends a query to Web server.
2. PHP connects to database and get the result of the query – a reference to MP3 file.
3. The user downloads the MP3 file.

The musicshop table is created as in figure 3.2.

Field	Type	Null	Key	Default	Extra
id	int(12) unsigned		PRI	NULL	auto_increment
music_type	varchar(30)	YES		NULL	
title	varchar(60)	YES		NULL	
sub_title	varchar(60)	YES		NULL	
artist	varchar(40)	YES		NULL	
subject	text	YES		NULL	
sample_file_reference	varchar(80)	YES		NULL	

full_file_reference	varchar(80)	YES		NULL	
image_reference	varchar(80)	YES		NULL	
free	varchar(4)	YES		NULL	
sale_count	int(7)	YES		0	

Figure 3.2 table musicshop

The sample_file_reference and full_file_reference are the URL for the MP3 sample file and full length file (as discussed previously). The image_reference is a URL for the music CD cover image. The music_type and artist store the genre of music and artist's name. The subject keeps the description of the music, for example, music publishing background or artists' introduction. The id, free and sale_count are special fields, used for system process. The id, as the primary key of table, not only speeds the system retrieving data from database, but also simplify the parameter transmitted among pages. The free field keeps a flag value, "yes" or "no" so that system can know which music can be provided in the Free music page. The sale_count is integer field and it is filled default value "0". According to this value, the system can publish the music of top sale to the Top sale page. The Top sale and Free music pages will be introduced later.

transaction table

The transaction table is used to store the users' transaction. It is created as shown figure 3.3.

Field	Type	Null	Key	Default	Extra
b_id	int(11)		PRI	NULL	auto_increment
music_id	int(12)	YES		NULL	
shopping_data	Date	YES		NULL	

username	varchar(15)	YES		NULL	
----------	-------------	-----	--	------	--

Figure 3.3 table transaction

music_id, shopping_data, and username are used to store the record. This record shows which music is bought by which user in which date. It makes shopper to check the transaction notes easily from the online payment processing company.

userinfo table

The system requires the users to register before they purchase music. Although online payment processing needs user's private information, such as name, address, credit card number, credit card expiry date and others in the purchase stage, users can avoid reentering these information during each shopping. Once they register in system, they only need confirm their private information in the Check out page.

However, everybody knows that if someone gets these information together at the same time, especially credit card number and credit card expiry date, he can purchase anything in any online stores no matter whether this credit card is his or not. So, we must consider how to prevent other people from getting these information together at the same time. On the other hand, the online payment processing application from www.itransact.com is integrated into this project. This application spilt the users' private information into two parts. The general information is obtained from the shopping cart system, and the credit card number and expiry date are entered in the page of www.itransact.com. Therefore, the table userinfo shouldn't keep the information about the credit card.

This table userinfo is created as figure 3.4.

Field	Type	Null	Key	Default	Extra
username	varchar(15)		PRI		
password	varchar(20)	YES		NULL	
pwd_rem_q	varchar(40)	YES		NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
address	varchar(64)	YES		NULL	
city	varchar(32)	YES		NULL	
state	varchar(32)	YES		NULL	
country	varchar(32)	YES		Canada	
postcode	varchar(32)	YES		NULL	
phone	varchar(32)	YES		NULL	
email	varchar(40)	YES		NULL	

Figure 3.4 table userinfo

Although all the fields in the table are allowed null, users cannot leave these fields empty because an error control, which checks if the form is filled completely by users, is in the GUI of shopping cart system.

administrator table

Since there is an administration module in the shopping cart system, the table "administrator" is needed. This last table is administrator. This table is simpler than the

others. It keeps only the administrator's username and password. It is created as show in figure 3.5.

Field	Type	Null	Key	Default	Extra
username	varchar(15)	YES		NULL	
password	varchar(15)	YES		NULL	

Figure 3.5 table administrator

3.2 Shopping cart Design

The MP3 shopping cart is made up of four main parts. They are catalog, shopping cart, online payment and administration. The parts of catalog, shopping cart, and administration include several separate functions. But in the administration part, all the modules interact with users. Therefore, there are two basic views of the shopping cart system: the user view and the administrator view. The shopping cart design will come up with these two system flows.

3.2.1 The user view Design

An MP3 online shopping cart should make the user feel likes shopping in a real music shop while feeling more comfortable. It should have friendly user interface, easy search functions and purchase operations.

In this project, a special function is introduced: it allows a user to listen to any pieces of sample music the user is interested in. When users click the link of the sample music, they can download the sample music file into their computer. Several pieces of the 30-second sample music are provided in the system. This sample music file which is only

400 – 500 bytes is small enough so that users using low speed modem (e.g. 28.8k) can easily download.

Since this is MP3 music shopping mall, this site should provide free MP3 player software to users. These players should include different platform applications. For example, Microsoft Media player and Winamp are in Windows, FreeAmp is in the Linux /UNIX or Windows and XMMS is in Linux. The shopping cart has a clearly displayed notice of music copyright.

A typical procedure followed by users is as follows: users browse the music catalog, listen to sample music, add the music into the user's cart, check out, make online payment using credit card, and download the music. The user's view of the MP3 shopping cart is shown in figure 3.6.

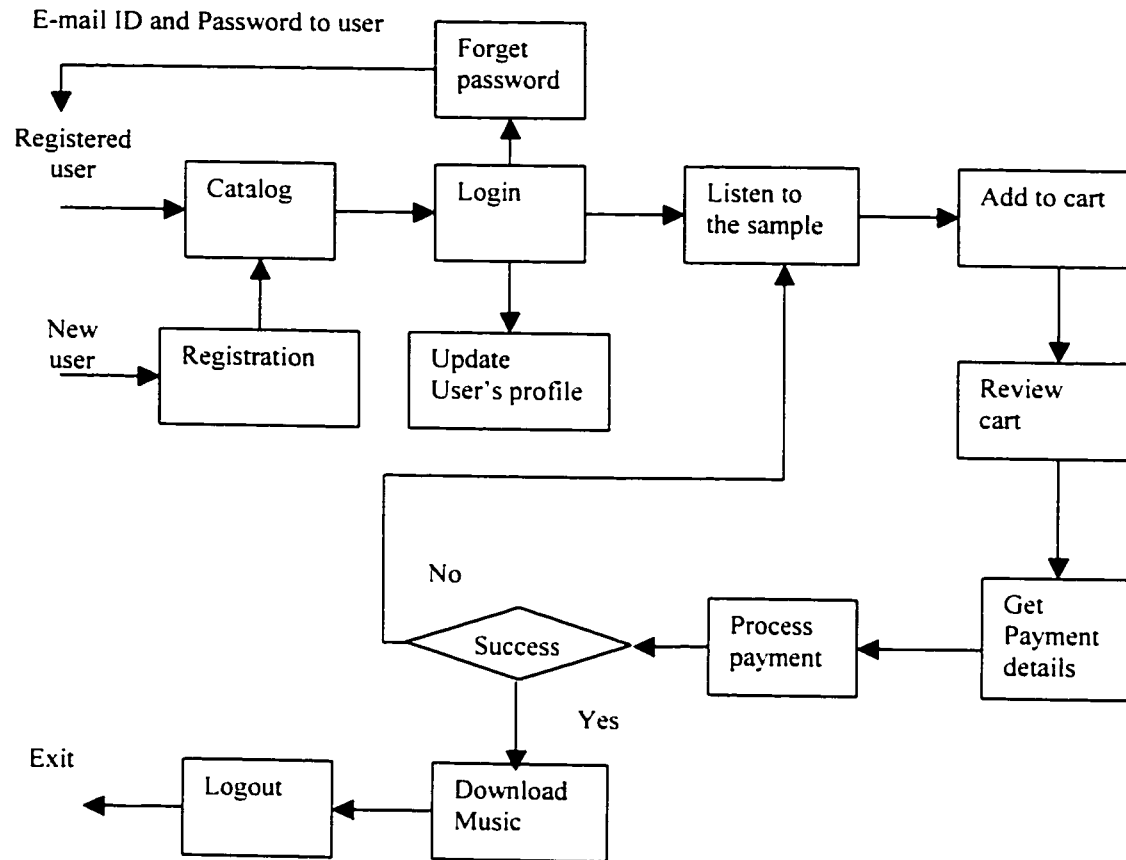


Figure 3.6 The user view of MP3 shopping cart

Figure 3.6 shows the main link between scripts in the user part of the system. A customer will come first to the main page, which provides links to different modules. From there, the customer can get the sample music, purchase music, and get the full length music. In order to simplify the purchase process, the user is required to register before using this system. For the user view, the navigation of site is shown in Figure 3.7.

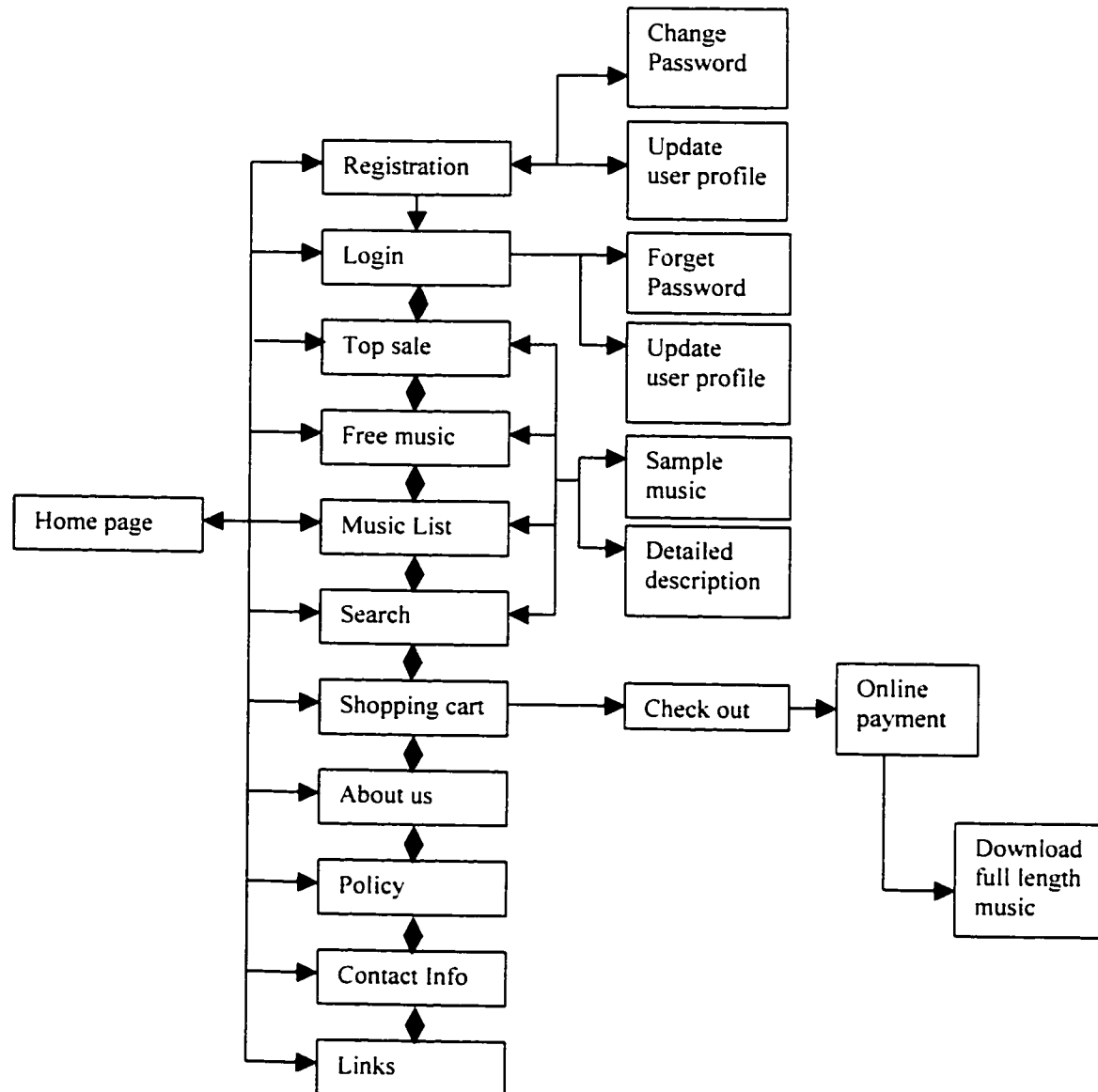


Figure 3.7 The hierarchy of the web pages for the user view

In this illustration, the administration module is omitted because it is for user view. Although there are links from the home page to shopping cart page, the user has to login first. The reason is that user's information is directly input from the database in the check out page. The user deals with the elaborate registration form only once. In the check out

page, she has the option of modifying her private information stored in the database. The system also allows the users to update their profiles if necessary.

For the online shopping cart, the online payment process is a special and important part. The following three are the main transaction clearing mechanisms for it:

- Sign up with a transaction clearing provider. There are many alternatives here depending on the area the user lives in. Some of these will offer real-time clearing, and others won't. Whether the user needs live clearing depends on the service you are offering. If the user is providing a service online, the user will most likely want it; if you are shipping goods, it's less crucial. Either way, you are not responsible for storing credit card numbers.
- Send a credit card number to merchants via encrypted email, for example, by using PGP or GPG. When merchants receive and decrypt the email, merchants can process these transactions manually.
- Store the credit card numbers in the merchant's database. The merchant can process these transactions without any users' inputting and emailing.

The last one is easy to implement in the system. But it is not a good idea. If the system uses this mechanism, you must guarantee that both your web server and database are 100 percent secure. In the past three years, we often hear the news about hackers stealing user's credit card number from some online companies. Recently, most companies have realized the seriousness of the problem is if they are attacked by hacker. So, they have given up this mechanism.

The second mechanism, in fact, is manual processing. The online shopping benefits both the user and the shopper. It provides comfortable and convenient shopping for the user. It also saves the shopper's time and human resource. This transaction mechanism needs manual handling of the transaction, so it doesn't reflect the strength of online shopping cart. Therefore, it is not a good choice.

The first mechanism makes the shopping cart system avoid storing the credit card number. The transaction processing is processed by a third party. It also includes real-time clearing so that it can meet the most online shopping cart's needs. It improves the online transaction's security. Consequently, it is the best choice among these three transaction processing mechanisms. It is applied into this project.

There are several ways to incorporate online transaction into the system. One is adding a module, which is CCVS module in PHP. The detailed information is in the PHP4 manual. Another is using third party application, such as Mainstreet, Cybercash, etc. There are two considerations when choosing which scheme is used in the MP3 online shopping cart.

One is that this transaction processing application must have the PHP API because this shopping cart is built in PHP scripts. And it can provide the free test version since this MP3 shopping cart is only an academic project. Another consideration is that it can process the credit card clearing for the transaction immediately.

MP3 shopping cart is a little bit different from other shopping carts, such as online book store, and online computer shopping mall, etc. In these shopping carts, as users browse an online catalog, users can add items to their shopping carts. When they finish the browsing, users check out of the online store – that is, purchase the items in their cart. This processing is illustrated in figure 3.8.

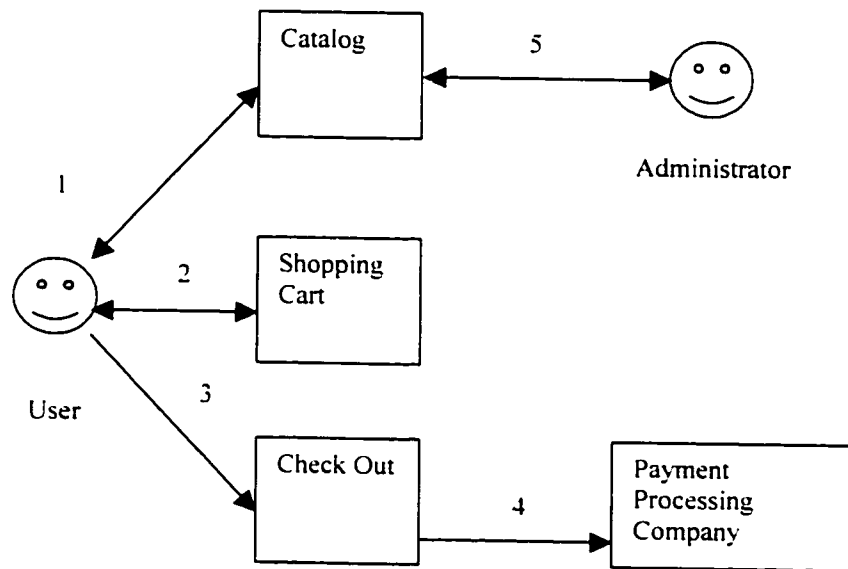


Figure 3.8 The normal shopping cart structure

1. Users browse the products catalog.
2. Users check their cart or add items in their cart.
3. Users check out items from cart, and confirm the purchase.
4. The transaction goes to the payment processing. If it is successful, the user will receive their items in a few days. The purchase is complete.
5. The administrator modifies the online catalog.

The MP3 shopping cart, however, is more like service online. It allows users to listen to the sample music before adding items to their shopping cart. In the purchase stage, unlike normal shopping cart that has at least one day shipping duration, users can download the full length music immediately after purchase. So it needs a real-time clearing. These lead to a specific design of the database, server side modules, and online payment. The figure 3.1 will be changed to figure 3.9

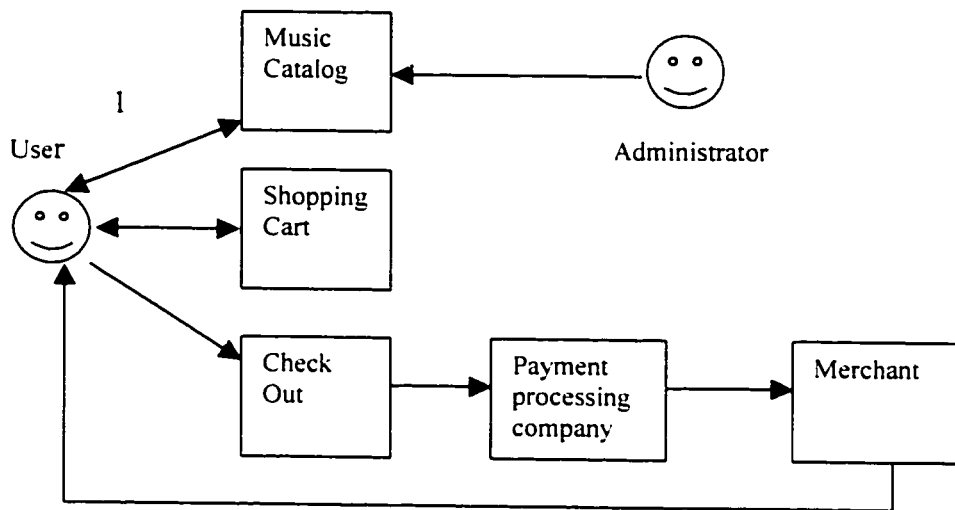


Figure 3.9 The MP3 shopping cart structure

1. The user browses the music catalog and plays the sample of music.
2. The user adds the music to her cart or views items in her cart.
3. The user checks out items from cart and confirms the purchase.
4. The transaction goes to the payment processing. If it is successful, the online payment processing company will email shopper a transaction receipt.
5. If payment processing is successful, the user can download the music he just bought. The purchase is complete.
6. The administrator modifies the online catalog.

Note: If the online payment processing fails (e.g. the user's credit card number is invalid), the user will go back to shopping cart.

The CCVS module (see PHP4 manual) in PHP is easy to integrate into the PHP shopping cart system. It also provides both real-time and normal transaction clearing. Unfortunately, it doesn't provide a free test version for us. If developers want to build a real PHP online shopping cart, CCVS is the best choice.

For this project, we have to choose a third party application and embed MP3 shopping cart system. According to the above considerations, the iTransact (www.iTransact.com) is a suitable application. It has three services on transaction clearing, are include real-time transaction clearing. iTransact has simple API, which is HTML form submitting. This API is almost suitable for any online shopping cart system since most shopping carts are built using a script language and HTML combo. Furthermore, it allows us to create a test merchant account. Using test credit card number, we can test all functions in the MP3 shopping cart. The detailed integration will be presented in the implementation part.

3.2.2 The administrator view Design

Administration is an important part in the shopping cart system. Its functions are only for the administrator. So it has its own GUI. It includes inserting new music into database, deleting the existing music from database, editing information of the music and changing password. The administrator view of the MP3 shopping cart is illustrated as figure 3.10.

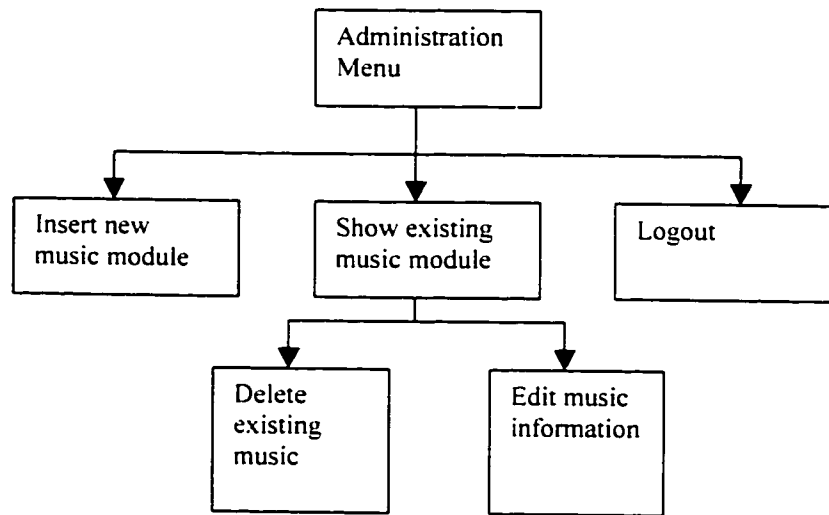


Figure 3.10 The administrator view of MP3 shopping cart

As database design, this project uses the database to store only the music information and the reference of the MP3 files. In this part, we discuss the procedures used by the administrator to put the MP3 files and the MP3 sample files of music in the database.

There are two solutions for this. One is putting a function in the *Insert new music* module. While updating information for new music, the corresponding MP3 and sample music files are uploaded to the corresponding directory. This method also extends to Edit the existing music and Delete the music. The advantages of this method are that administrators can complete their work easily and they can work in anywhere. However, we know that the uploading speed is much slower than downloading speed with HTTP protocol [8]. Normally, the new music is compressed from the new music CD. A CD

includes at least 50—60 minutes of music. The total of MP3 file size will be 30—40M. That is a large amount of data for uploading these MP3 files with the HTTP protocol.

Another solution is to split the job of inserting the new music into two steps. First, the administrator uploads the full length and sample files into the corresponding directory. Then the administrator inserts the information of new music into the database. As the administrator of MP3 shopping cart, the administrator should have the permission of managing the directory tree of web site. In this solution, she can use more robust FTP software for uploading those big size files with FTP protocol. After deploying the MP3 file in the directory tree, she can insert the information of the new music into the database with the *Insert the new music* module in this system. This solution makes administrators complete their work easily.

Figure 3.11 shows the administrator view web page navigation.

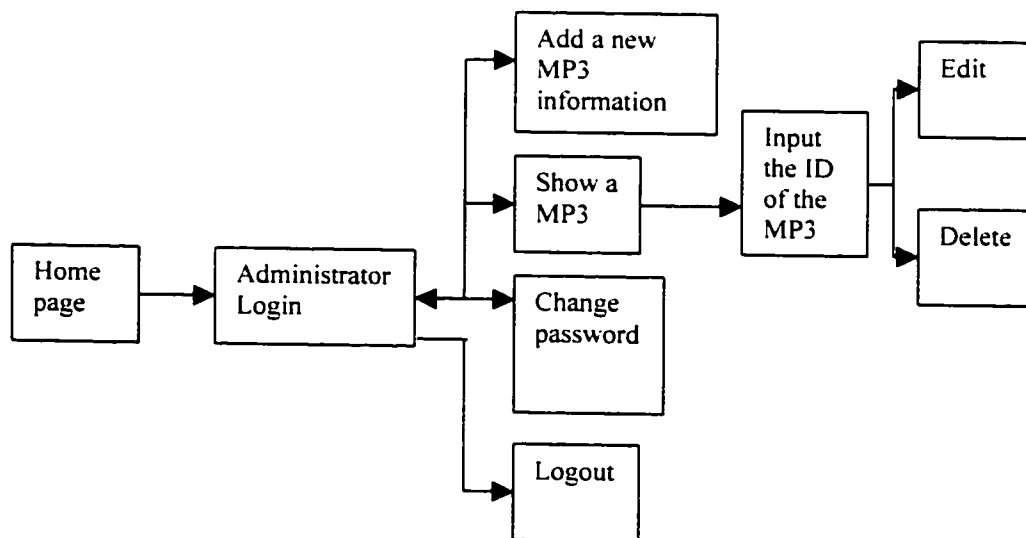


Figure 3.11 The hierarchy of the web pages for the administrator view

Chapter 4 Implementing the MP3 online shopping cart

In this chapter, we present the implementation details of the MP3 shopping system. First, we will introduce the file system in the MP3 shopping cart. The PHP scripts are generally publicly loadable. Since the Web server is configured to parse files with the extension .php using PHP interpreter, outsiders will be able to view only the uninterrupted source. However, if using files with other extension (e.g. .inc files) are placed in the web directories, anybody requesting them will receive the unparsed source. So we should be careful when using other extension. Based on this reason, two kinds of file are used in the project. They are .php files and .inc files. All the source codes with PHP tags are put in .php file. The .inc files contain only HTML tags, such as page header files and footer file.

And for security reason, the session is used in every page. Sessions in PHP are driven by a unique session ID, a cryptographically random number. This session ID is generated by PHP and stored on the client side for the lifetime of a session. It can be either stored on a user's computer in a cookie, or passed along through URLs [7]. The idea of session control is to be able to track a user during a single session on a Web site. By using it, we can easily support logging in a user and showing content according to her authorization level or personal preference. In this system, the `session_start()` function is always in the beginning of every script.

The MP3 shopping system consists of five main code modules as follows:

- Catalog
- User authentication
- Shopping cart
- Online payment integration
- Administration

In order to easily debug and maintain the system, the library of functions is built as functions.php file. It includes all the functions that are used in the system. This makes the source code easy to read.

4.1 Online catalog

Before implementing the online catalog, the user interface must be considered first in the home page. In this shopping cart, a main page is shown in figure 4.1.

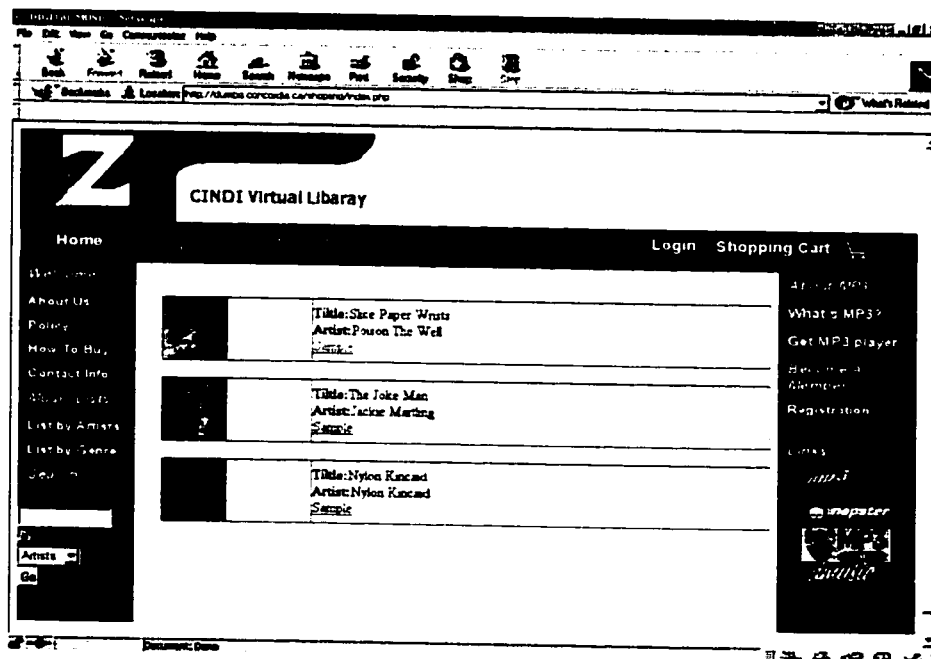


Figure 4.1 Home page of the shopping cart

In this main page, all the modules' links are provided except the administration module. The user can go to any module from here. This style is used for all pages of the catalog, giving GUI of a unified style. This page is divided into header and footer. They are imported from header.inc and footer.inc files. This will avoid changing every page using this style GUI, when the user interface is changed.

From figure 4.1, we can see that there are a variety of music lists provided in shopping cart. They are as follows:

- The top sale music list
- The free music list
- The music list by artists
- The music list by genre
- The music list from the result of searching

In fact, these different lists are obtained from the different queries. In the musicshop table, it has sale_count and free columns. This was introduced in section of database design. Using two queries as

```
select * from musicshop where free = 'yes'    and
```

```
select * from musicshop group by id order by sale_count desc limit 3
```

retrieve data from the database. Because of consideration of space, we limit the number of result to 3 for the top sale query. These queries are passed to the get_detail(\$query) method. This method returns the results of queries.

After getting the results of queries from database, we have to put the result into an array. Here, the `result2array($result)` method is called. It returns a two dimensional array so that we can output the results of queries to screen.

The free music and top sale output is as shown in figures 4.2 and 4.3.

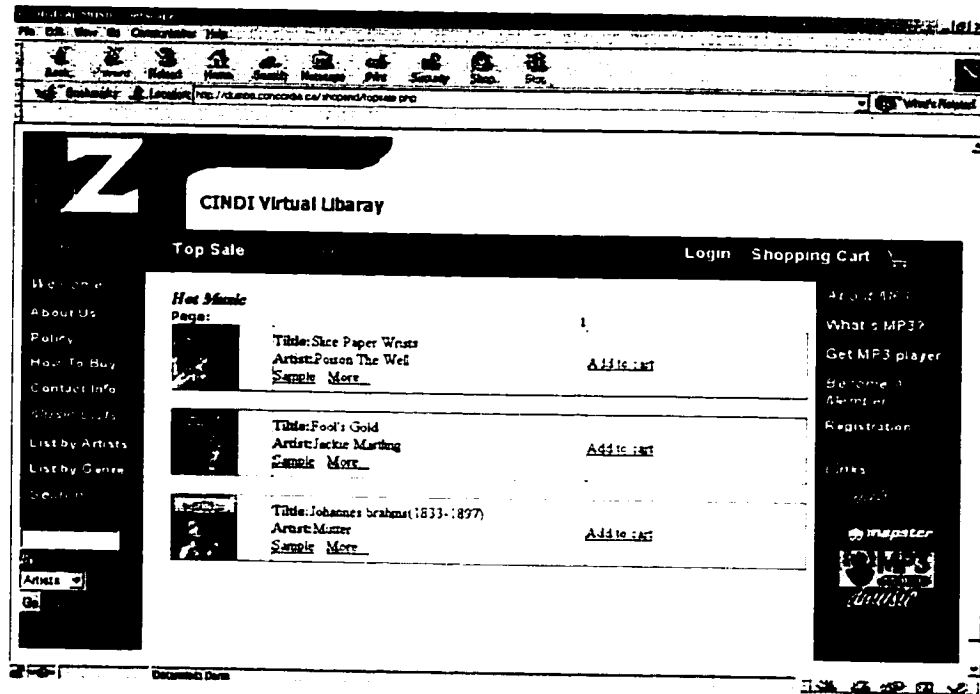


Figure 4.2 Output of top sale

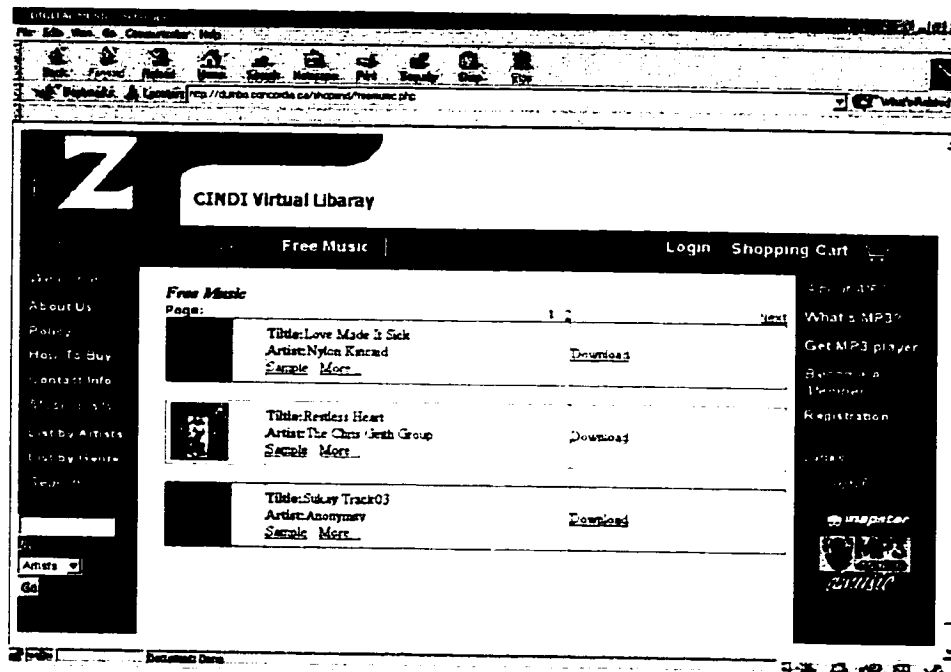


Figure 4.3 Output of free music

The list by artists and list by genre are different from topsale list. They output artists and music genre list first and provide the links to show detailed music information. In these two list the `get_detail($query)` and `result2array($result)` methods will be called twice separately. One is for genre or artists list, one is for detailed music information. The output list by genre is as shown in figure 4.4 and 4.5.

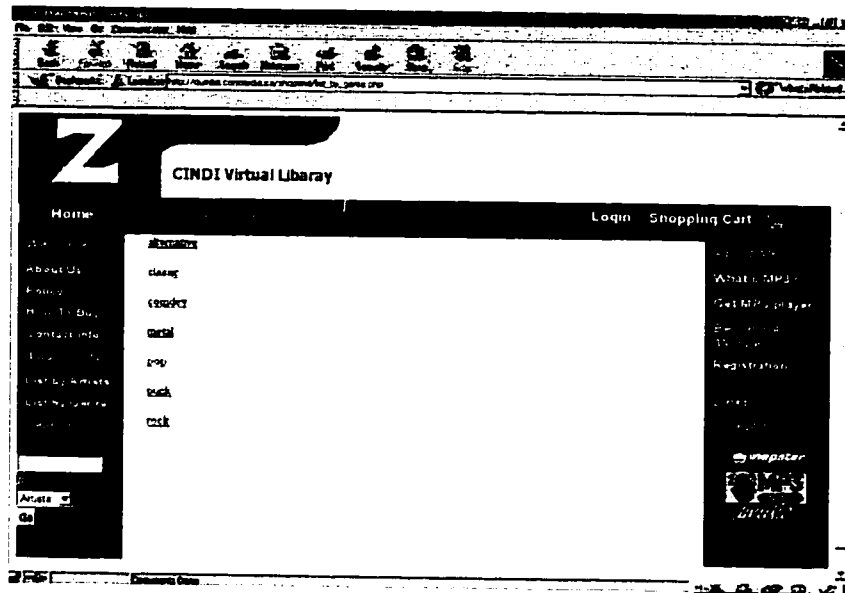


Figure 4.4 list by genre

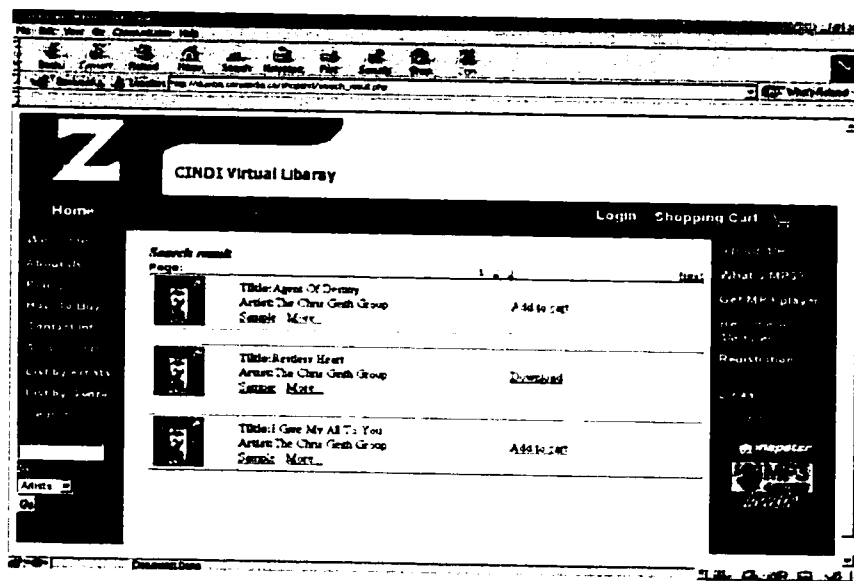


Figure 4.5 The detailed music information

The list by result of searching is similar to other list functions. The `get_detail($query)` and `result2array($result)` methods are reused here. But in the search function, the judgement clause is also used. It judges what query will be passed—artists or genre. Its output screen is as shown in figures 4.6 and 4.7.

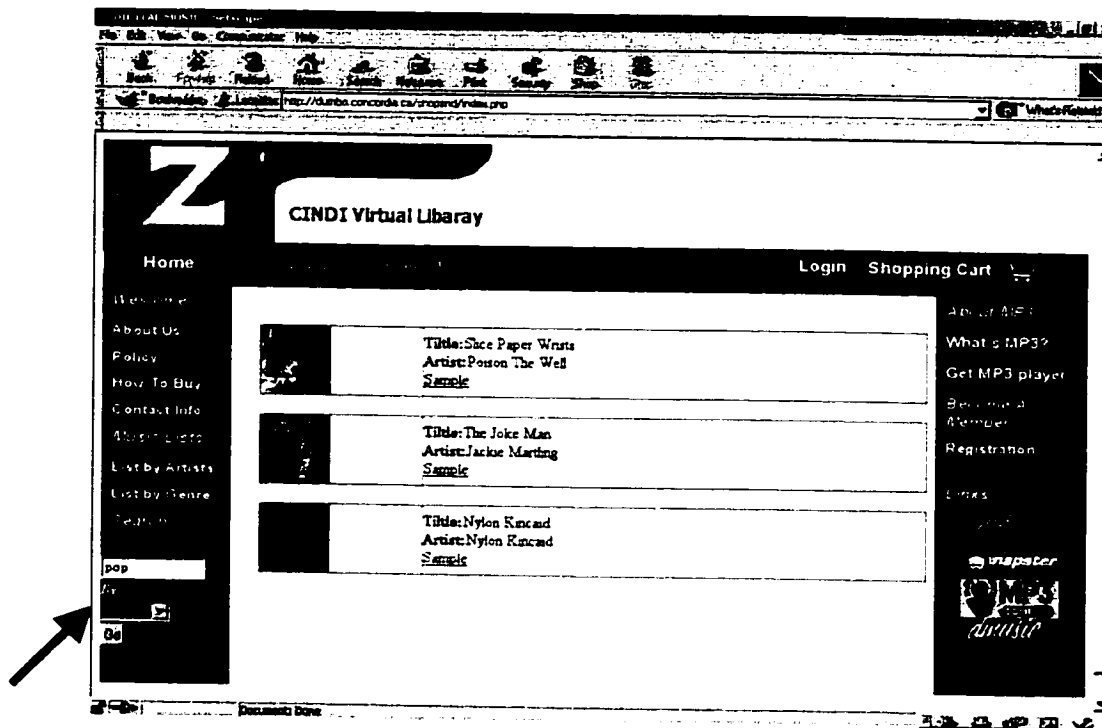


Figure 4.6 Inputting the key word for searching

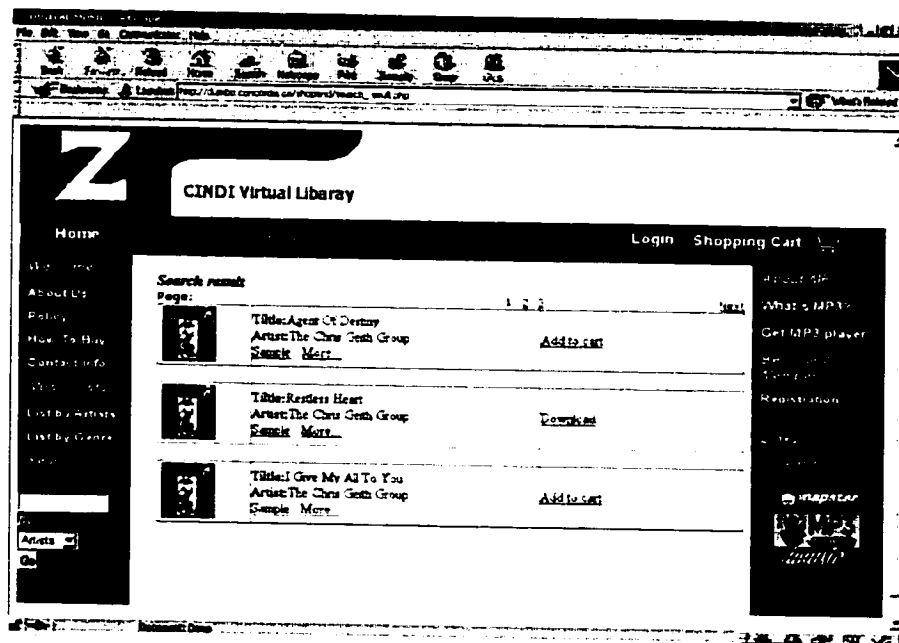
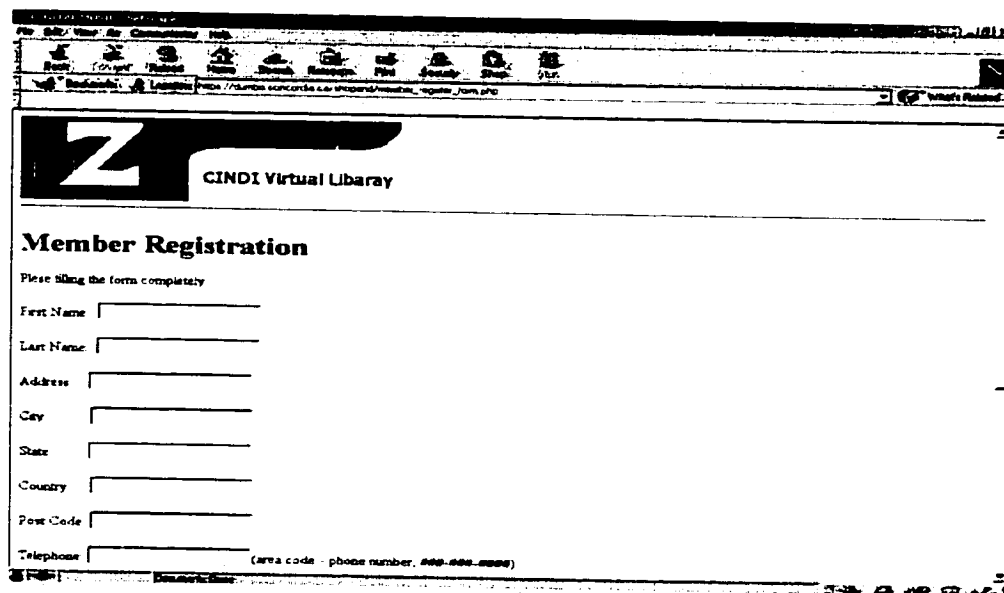


Figure 4.7 The result of searching

4.2 User authentication

There are five main elements to the user authentication module: user registration, update user's profile, login and logout, changing password, and password reminder. From the user authentication module, the session control will start to be used. Except the online catalog module, all the other modules should use session control. The session control will be used in every page. Using session control, the system can decide if the user has permission to access such a page. During registration, log-in, and check out, the SSL server will be used because the user will input or output the user's private information over the Internet. As it was mentioned in chapter 2, links will connect to https:// rather than to http://.

For user registration, the user will be directed to the registration form to fill it. Then the system inserts the data to database. The outputting screen of registration form is as shown figure 4.8.



The screenshot shows a web browser window displaying the 'CINDI Virtual Library' member registration form. The browser's address bar shows 'http://www.cindilibrary.org/registration/registration_form.asp'. The form has a header with the CINDI logo and the text 'CINDI Virtual Library'. Below the header, the title 'Member Registration' is displayed. A note says 'Please filling the form completely'. The form contains several input fields: 'First Name', 'Last Name', 'Address', 'City', 'State', 'Country', 'Post Code', and 'Telephone'. A small note next to the telephone field says '(area code + phone number, 000-000-0000)'. The browser's status bar at the bottom shows 'Document: http://www.cindilibrary.org/registration/registration_form.asp'.

Figure 4.8 User registration form

Behind this form, there are many input error check methods. First, we check if the user fills the form completely. In the database section, we mentioned that the database tables allow a column to be null. But the system doesn't allow empty form. So we use a method `filled_out($form_vars)` to check this error. It expects to pass an array of variables in general, this will be the `$HTTP_POST_VARS` arrays or `$HTTP_GET_VARS`. It will check if they are filled out, and return true if they are and false if they are not.

In the registration form, the email address and phone number are checked by methods `valid_email($email)` and `valid_phonenumber($number)` separately. These two methods use the existing PHP feature function `ereg(string pattern, string string)`. It will return a Boolean value. According to the email address format, `ereg()` can be called as following.

```
ereg("^[a-zA-Z0-9_]+@[a-zA-Z0-9\.-]+\.[a-zA-Z0-9\.-]+-S", $email)
```

The parameter for phone number checking is more simpler than email address checking because we just concern if the input is a number. The method for phone number checking is:

```
ereg("([0-9]{3})-([0-9]{3})-([0-9]{4})", $number).
```

In order to avoid the user input the wrong password because the password is hidden in the form, the registration form includes a repeat password field. The error checking also includes comparing if the input values are the same in both password field and repeat password field.

Before inserting the data into database, two things have to be done. One is to check whether it is taken by others. If it is taken by others, the system outputs a error message to user.

Another is checking whether the user has logged in. In the system, the logged in user's username is kept in the session variable. However, we can not use session in the registration form file. The reason is that if the session is started in the begin of the registration form file, the cache of the web browser of the user can not remember the HTTP_POST_VARS values when the user returns to this page. For example, when the user doesn't fill the form completely and clicks submit button, the system shows the error message that the user has not filled the form. But when the user comes back to registration form and want to continue fill the form, the form is emptied. Therefore, we should check whether the user logs in or not after the user clicks the submit button. If the user has logged in, the system links to the member service page automatically rather than puts the data into the database. The registration form page also provides the link to the member service as shown in figure 4.9.

Figure 4.9 Registration form continue

The updating user's profile function is similar to registration. There are two methods to complete this function. They are `display_profileinform($profile)` that puts values of the result of database into the correspond HTML form, and `update_profile($query)` that send the query to update database.

Implementing the login and logout function is very easy. For login, first, using the `login()` method to check if the username and password are correct. If the `login()` returns value is "true", the username is registered as session variable – `valid_user`. In the beginning of every page, the system just checks the session variable's value; it can know which user is accessing the system. On the contrary, the session variable of `valid_user` is destroyed in the logout function. The login page is shown as figure 4.10.

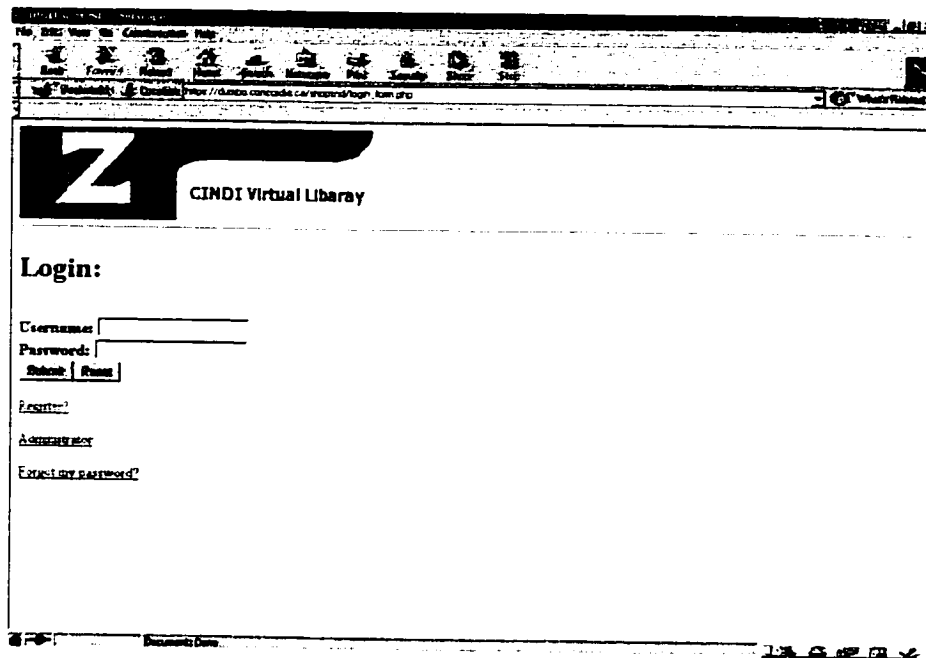


Figure 4.10 The login page

In the login page, there is a link which goes to password reminder. There is a field of `password_rem_q` in the table `userinfo`. When the user fill registration form, she is asked the general question of "What is your mother's name" for password reminder. When the user needs the system to remind her password, she is asked this question again. If her answer is correct, the system will use the PHP's email function to email password to her email box. The code of email part is:

```
mail($email, "Your password.", $msg, $from)
```

The change password function in this module is much similar to update user's profile and it is omitted here.

4.3 Shopping cart

This system needs the user to login before they view the shopping cart. If the user clicks the shopping cart without login, the system will return a page letting the user know that

she has to login first, and it provides useful links to the user in this page. This page is as figure 4.11.

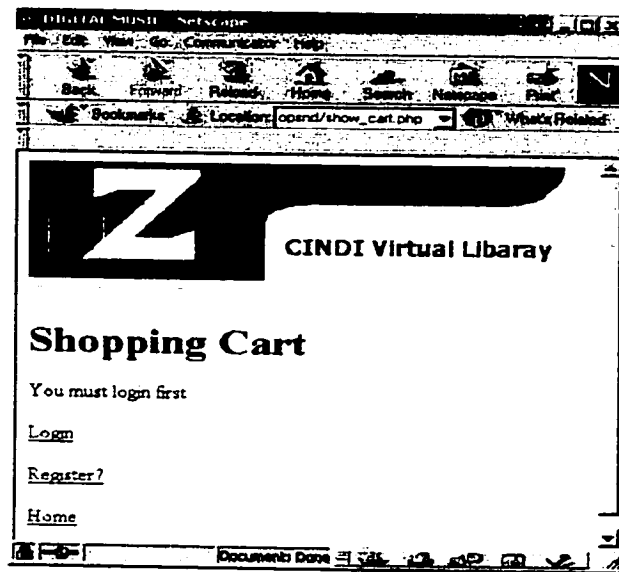


Figure 4.11 The system refuse the user view shopping cart

The shopping cart functionally revolves around a session variable called \$cart. This is an associative array that has music_id as keys and quantity as values, although the quantity is always 1 in the MP3 shopping cart. For example, if you add a pieces of MP3 music to your shopping cart, the array would contain: 3 => 1. That is, the MP3 music with the id 3. When you view your cart, the system will use \$cart array to look up the full details of the items in the database. Another session variable is used to show the total price. It is called \$total_price.

When the user clicks the shopping cart, the system will display the contents of the cart. The system will execute a part of code to judge if the cart is empty. The code is following.

```
If($cart&&array_count_values($cart))  
    Display_cart($cart)
```

Else

```
Echo "<P> Your shopping cart is empty.</P>";
```

As this code, if the cart has some contents, the `Display_cart($cart)` will be called. If the cart is empty, the system just displays the message to the user.

When the user click the Add to cart, the system adds the item to session variable `$cart` first, then it calculates the total price of items in the cart and saves it into the another session variable `$total_price`. Finally, the system executes the `show_cart` function and outputs the screen as figure 4.12 for the user.

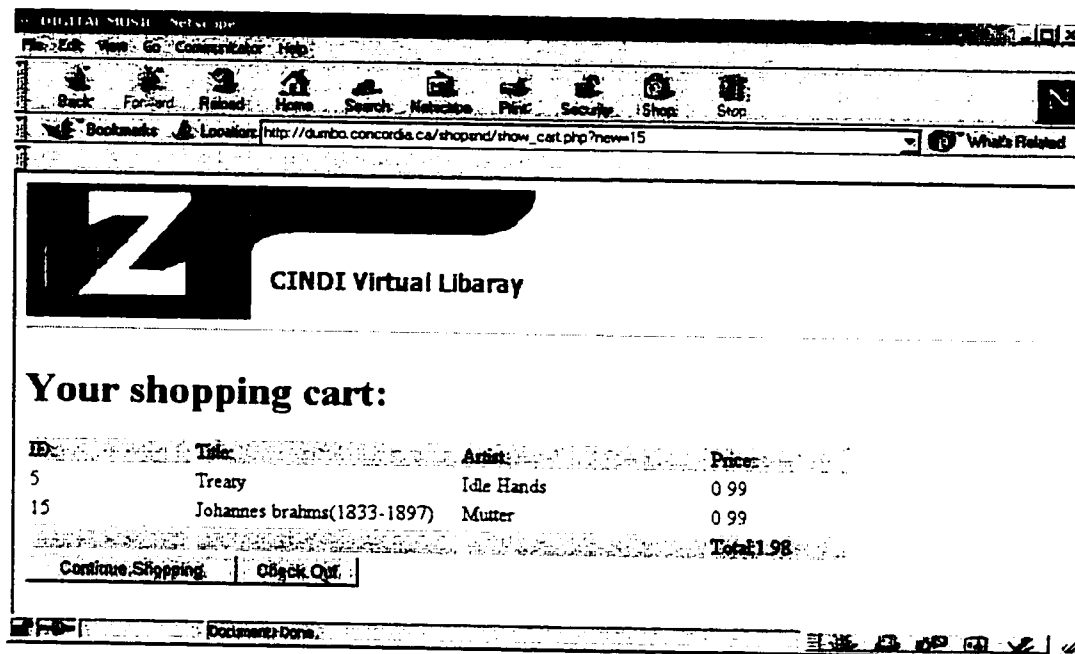


Figure 4.12 View the shopping cart

In this page, the system provides two buttons: continue shopping and check out. When the user clicks continue shopping, the system will return to home page. If the user clicks the check out, the system will executes the online payment processing with the secure connection.

4.4 Online payment integration

As mentioned in shopping cart design section, we choose the iTransact for handling online payment processing. Before integration this application, we must create a test merchant account in www.iTransact.com.

When the user click Check Out, the system only loads the user's private information from database and displays it for confirmation by user. In this page, the system also provides a link to the user to update her profile just like update profile in the user authentication module. This page is show in figure 4.13.

Qty	Description	Title	Image	Price
5	Treaty		Idle Hands	0.99
15	Johannes brahm(1833-1897)		Mutter	0.99
21	a white ceramic...			
				Total: 1.98

Confirm your personal information, if there are some wrong, please
[update your information](#)

First Name: Dong Last Name: Yang
Address: 1160 st mathieu, APT 1405
City: Montreal State: Quebec Zip: H3H 2P4
Country: Canada
Phone Number: 514-989-9530
E Mail Address: ydo8@hotmail.com

[Purchase](#)

Figure 4.13 Check out

After the user clicks the Purchase button in this page, the system goes to the iTransact to process the online payment.

The iTransact provides three models for transaction clearing. We integrate the standard model into the system. Since the iTransact provides the API with HTML format, we just

copy the HTML code and put in our own merchant account ID and password. In the meanwhile, we have to change some values in the form. The details will be seen in the checkout.php file. The iTransact allows the merchant to test the account with the test credit card number 5454545454545454 and any expiry date. This page is shown figure 4.14.

The screenshot shows a web browser window displaying the 'Dong - TEST ACCOUNT SECURE PAYMENT FORM'. The browser's address bar shows the URL 'http://localhost:8080/checkout.php?card=5454545454545454&exp=12/2006'. The page content includes the following text and form elements:

**Dong - TEST ACCOUNT
SECURE PAYMENT FORM**

Please complete your transaction by providing payment information below. All information is submitted and handled securely via Netscape's secure protocol, SSL (Secure Socket Layer).

TO PAY BY CREDIT CARD, ENTER YOUR CARD NUMBER AND EXPIRATION DATE
WHEN YOU ARE FINISHED, CLICK THE SUBMIT SECURELY BUTTON BELOW

Card Number: Exp. Date:

Below the input fields are logos for VISA, MasterCard, and American Express.

This transaction will be processed in REAL-TIME.
Your transaction will complete in 25 seconds to two minutes, depending on network traffic. In some cases, it may take longer. If your browser supports JavaScript, a timer will count up from zero. Please be patient. Your information is being encrypted for your protection.

A timer displays '0:05'.

Submit Securely

At the bottom, a small disclaimer states: 'Are you concerned about using your credit card on the Internet? Dong - TEST ACCOUNT uses private secure technology with a digital SSL certificate. You may view the properties of this page to verify the certificate.'

Figure 4.14 Online payment processing by iTransact

If the credit card number is valid, the iTransact will display the message to the user and provides a link returning to the MP3 shopping cart system. The link was passed as parameter from the MP3 shopping cart to iTransact. This page is shown in figure 4.15.

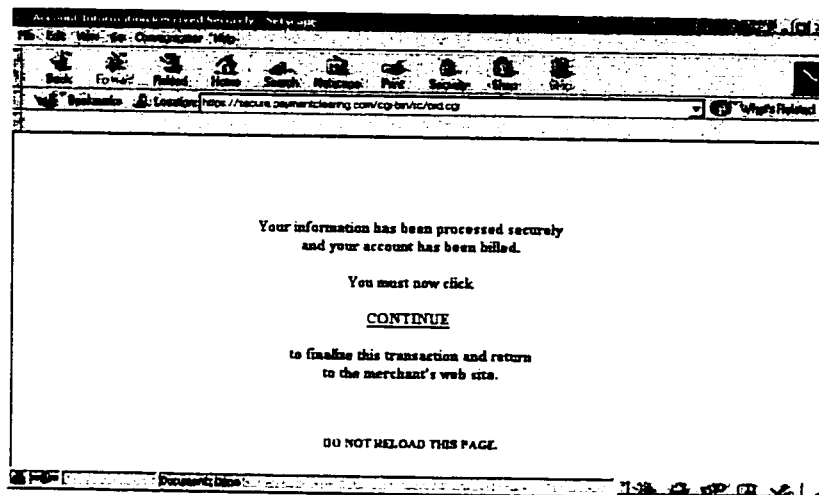


Figure 4.15 The online payment processing is successful

Behind this page, the iTransact will send a notification about this transaction to the merchant. The user will return the download page as figure 4.16 in the MP3 shopping cart system.

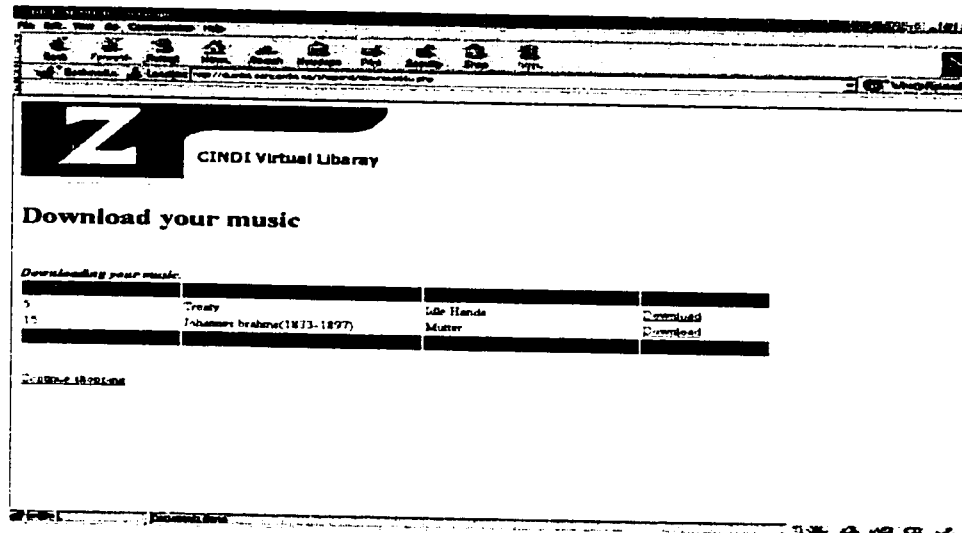


Figure 4.16 The page of download music

When the user leaves this page, she can either click back button in the web browser or click the continue shopping link, the system will destroy the session variables \$cart and \$total_price. It means the user's cart is emptied.

4.5 Administration

Sine the shopping cart system is mainly for the user, the entrance of administration module should not appear in the home page. So it is put in the login page. It likes figure 4.8.

The administration interface is implemented very simply. An administration menu is implemented as figure 4.17.

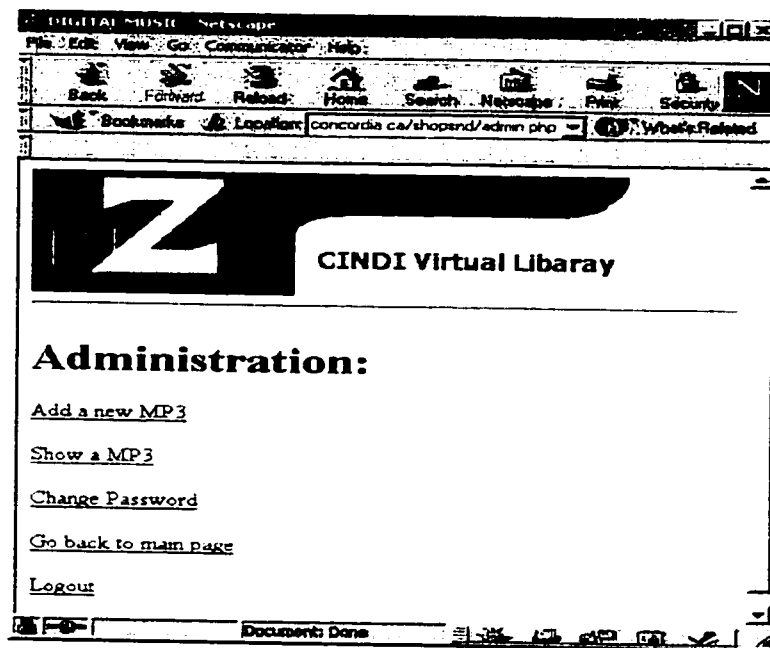


Figure 17 The administration menu

All functions in the administration module are exactly the same as the functions in the user authentication. For example, the *Add a new MP3* function is the same as the *user registration* (it is as shown in figure 18.), and the *show an MP3* function is same as the *update user profile* (it is as shown in figures 19, 20 and 21.). In most functions, we can reuse existing methods to implement.

CINDI Virtual Library

Insert the new MP3

Music Type:

Title:

Sub-title:

Artist:

Subject:

Sample file reference:

Full file reference:

Image reference:

Free music:

Figure 18 Adding a new MP3 by administrator

CINDI Virtual Library

Input a MP3 ID

MP3 ID:

Figure 19 Inputting the mp3's id

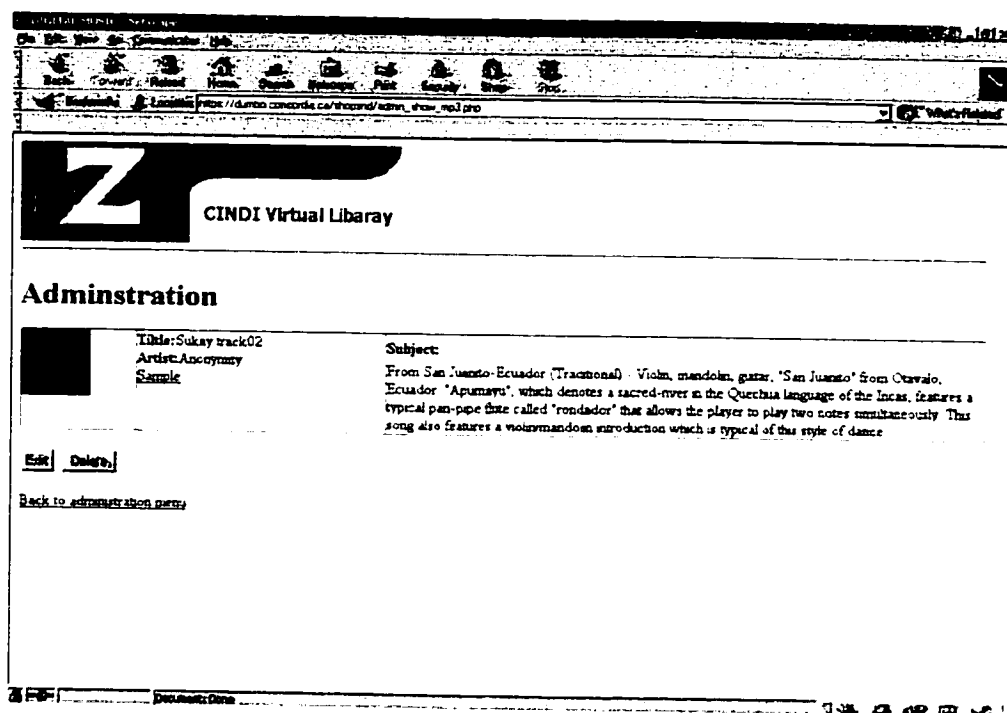


Figure 20 Showing the MP3 for administrator

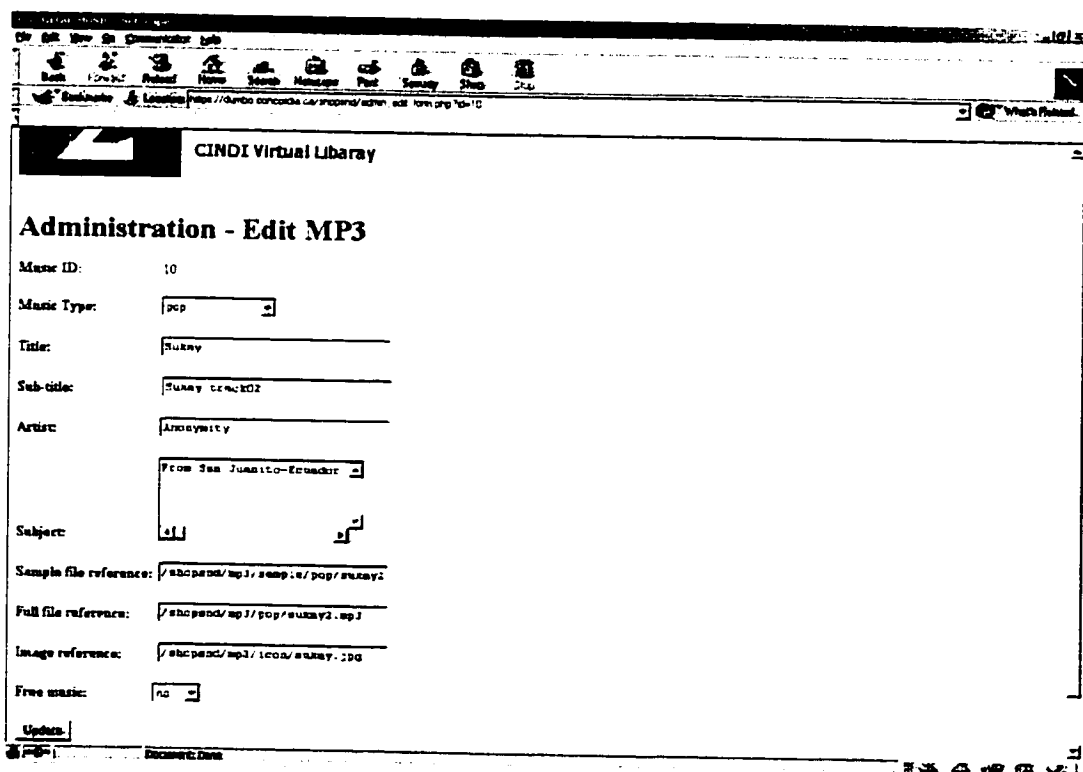


Figure 21 Editing information of music

However, we have to register session variable username of administrator as \$admin_user rather than \$valid_user. The system must distinguish the logged in user between a customer and an administrator, because the pages in administration module should not be accessed by the customer. But the administrator has the permission to access the whole site. Since this part is same as the user authentication part, the implementation is ignored.

Up to now, the shopping cart has been implemented. The more detailed code will be seen in source files.

Chapter 5 Conclusion and future work

5.1 Conclusion

The MP3 online shopping cart presents a way of development of the online shopping cart. As an online application example, it draws a structure of Web application. It implemented an exact free, robust and secure online shopping cart. On the other hand, the detailed comparison among the difference of web application development tools was provided in the project, and the MP3 shopping cart project proved that the PHP/MySQL is a rapidly developed combo on the Web application.

5.2 Future work

On one hand, as the real online music store, the seller wants to let her customers know what new music is available. So the system needs a mailing list function. Besides, the system may provide an area to collect music requests from users.

On the other hand, this application is programmed in PHP combined HTML. Although the HTML tags were inserted to the .php files for GUI, the PHP and HTML can be separated completely. After using CVS in the system, this system can be used to develop general shopping cart, and the HTML code is stored the pure .html file. Others just put their own HTML GUI, and changed a little thing (e.g. the web directory tree) in the configuration file in the system. This system will become their shopping cart whatever it is book store or computer online service.

Chapter 6 Reference

1. Christopher Jones, *MP3 Overview*, in www.webmonkey.com, July, 28, 2000
2. Sue Zeidler, *Metallica to Drop Yale From Napster Lawsuit*, in www.techtv.com/news, April 20, 2000
3. Gary Bollinger and Bharathi Natarajan, *Build an E-Commerce Shopping Cart*, in www.Java-pro.com, June 2000
4. Laura Rutherford, *Linux or Windows 2000?*, in www.notes.net, May 1, 2001
5. Raj Rajagopal, *Comparing Apache and Internet Information Server*, in *Unix World*, December 4, 2000
6. Dave Murphy, *Comparing Microsoft IIS And Apache HTTP Server*, in www.insidereport.com, July 23, 2001
7. Laura Thomson and Luke Welling, *PHP and MySQL Web Development*, published in SAMA
8. Larry L. Peterson, Bruce S. Davie, *Computer Networks: A System Approach*, Published in Morgan Kaufmann, March 2000
9. Tracy Pilcher, *An Open Source E-Commerce Solution*, in Devshed.com, March 28, 2001
10. Ying Zhang, *Building an E-Commerce Site*, in Devshed.com, June 8, 2001
11. Israel Denis Jr., *The Soothingly Seamless Setup of Apache, SSL, MySQL, and PHP*, in Devshed.com, June 7, 2001