DESIGN AND DEVELOPMENT OF ELECTRONIC PAYMENT ARCHITECTURE SUPPORTING UMIFIED PROCESSES OVER INTERNET

NC Malik Farhan Haider
PC Jabran Khalil Shahid

DEDICATION

We dedicate our effort to our families who love us and provided us constant motivational force and to the instructors who were always there to help us.

ACKNOWLEDEMENT

We are thankful to Almighty Allah who enabled us to complete this project. We show gratitude to our parents, whose love and care has enabled us to be what we are. Our deepest appreciation is extended to all whose unfeigned help and encouragement made the present work a reality.

We gratefully acknowledge the help and guidance provided by Colonel Raja Iqbal and our project advisor Major Muhammad Saeed. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

We would like to express our gratitude to all faculty members of the Department of Computer Science for their cooperation and healthy academic environment throughout our career at Military College of Signals, Rawalpindi.

DECLARATION

“No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere and neither may it be allowed to be reproduced.”

ABSTRACT

The advent of the internet and World Wide Web has sparked a rapid increase in the Net concepts of shopping online that helps us to develop personalized shopping experiences that put the information your customers need within easy reach.

The goal of this research project is to develop a system to manage the secure, verifiable transfer of value from a payer to a payee in an integrated manner over the Internet .As part of this project, we will be designing and coding the information flows and processes to support this integrated approach. The creation or integration with the financial applications on the backend is also included in the scope of this project.

The focus of this project is on the identification, design and implementation of concepts and solution that can provide solid, reasonable security to support electronic payments. The key areas that will be addressed are cryptography (for confidentiality), web interface (for user access) and Database (for record storage).There are a number of approaches and tools available for all three research goals, and we will be assessing and implementing the most optimal solution based on the minimization of false negatives, false positives and cost, and the maximization of usability and convenience.

To enhance the mechanism of online shopping a different mode of payment method has been developed. The revolution in telecommunication system actually promotes the payment method of purchasing and selling products on online. The evolution of scratch card system in the history of internet

online shopping is divulged as the leading phenomenon. Fixed amount of different cards are published by the company having unique numbers have played a unanimous role especially for those customers who lived in remote areas. The purchasers of these cards are either shopkeepers or general public.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In today’s global market place products and services are like commodities and transaction detail such as speed, security, privacy, automation and payment systems compatibility are becoming more critical. Financial Institutions that adapt and innovate will survive and prosper in this dynamic environment by delivering high value services to their customers.

An efficient payment system plays a central role in country’s infrastructure, both in terms of financial stability and facilitating transactions among consumers. Pakistan is currently running on a paper-based system, primarily cheques, but in recent year’s electronic payment systems have rapidly increased in importance for both retail and wholesale transactions.

The banking industry in Pakistan is undergoing a sea change of activities and is moving into a concentrated approach of networking through automation between banks. It is the foreign and private sector banks which have brought about this change by adopting new technologies and going online (e-banking).

This electronic revolution in banking basically centres on changes in the distribution channels of the financial institutions. The basis for the emergence of the modern electronic distribution channels is the result of the e-Commerce initiatives taken by the State Bank of Pakistan, which is focusing/promoting a robust e-Payment and settlement system in Pakistan’s financial sector.

The availability of online communication infrastructure in banks has created an environment for e-banking, which is based on electronic communication media to engage in business activities without going to bank premises. While it enhances business efficiency & effectiveness, it reduces business cost. It is a significant departure from traditional mode of business, creating new ways of carrying out business. It gives customers the freedom to do banking without consideration to geographical & time barriers. It will save time & travelling cost, increases the volume of business and give quicker return than in traditional mode of business.

1.1 History & Background:

With the development of Internet, a new breed of transaction, online payment has emerged. Online Payments allow people to buy goods and services on the Web. And in the same background every bank is trying to launch different products for its customers like Credit Cards, Debit Cards, ATM Cards, Web Banking, Online Shopping Facilities and many more to come. Recent studies show that more than 80% of internet payments are done using Credit Cards, which makes it the most popular means of payments on the web by far. In parallel, Card Fraud has also grown dramatically on the net, with hackers and criminals targeting the weaker security of these payments. As a result of these fraudulent activities, large numbers of Credit Card numbers have been captured and misused resulting in high level of charge backs for Merchants because of Cardholders claiming that their Credit Cards were used fraudulently.

In order to block the expansion of these fraudulently activities, card associations have developed new models for validating internet payment transaction, hoping to counter fraudulent activities by reinforcing the authentication of the cardholder.

Studies show that most Internet users still refuse to shop over the Web for fear of their Credit Card details being captured and misused. An E-payment system therefore needs to be truly guaranteed to restore this confidence. For this guarantee to be offered, all parties to an electronic transaction need to have some level of assurance that the Cardholder is who he claims he is.

Cardholder Authentication is the driving factor behind the development of our project over the Web, and is a recognized need in the shared payment markets, for Banks, Merchants and indeed the Cardholders themselves.

A number of previous have been made over the course of the development of the shared architecture to offer a secure solution for the payment of goods and services. These attempts include Virtual Card numbers, Secure Electronic Transaction (SET) Credit Card-based systems that involve PIN pad readers being installed at the Cardholder end of the solution and PKI-based systems. All these have failed to reach critical mass because of their lack of simplicity, interoperability as well as high cost of implementation for merchants, issuers, acquirers and Cardholders.

Our project offers a break through in this area, as it proposes a simple infrastructure into which any authentication system can be plugged. It also limits the implementation cost at the merchant site, which has always been identified as a critical factor for any such system to be successful.

1.2.E-Commerce:

Electronic Commerce (E – Commerce) is the use of telecommunications or the Internet to carry out business of any type. E-commerce is not similar to other kinds of commerce. It involves the buying and selling of goods, only in this case, over the INTERNET. E-commerce sites range from a simple Web page highlighting a single item to fully developed on-line catalogs featuring thousands of products .

1.2.1 The Process of E-Commerce:

The process of e-commerce consists of many steps, a few important of which are

1. Attracting customers
   - Advertising, marketing
2. Interacting with customers
   - Catalog, negotiation
3. Handling and managing orders
   - Order capture
   - Payment
   - Transaction
   - Fulfillment (physical good, service good, digital good)
4. React to customer inquiries
   - Customer service
   - Order tracking

1.3.Present Scenario:

At current prices, Internet Payment Systems do not offer a general-purpose method of transferring money. In many cases the recipient of the payment can only be a merchant, and person-to-person payments are not supported.

Currently, Internet Commerce applies primarily to niche markets:

- Where very broad stock is a competitive advantage (CD's, books, musical scores) global mega stores may be favored, or search engines for co-operating chains of smaller stores that are technically able to share their databases.
- Specialized product, where the customers are spread all over the world, but there are very few of them per square kilometers, so they can't be effectively reached by a shop-front. This has the potential to open up for trade a whole range of new connoisseur products which hitherto you simply have not been able to trade in any cost-effective way.
- Electronic Product, such as newspaper searches, legal or medical searches, books distributed in Postscript form, etc

1.4. Existing Methods of purchasing on net:

It seems that online shopping has been discovered. Magazine covers, newspaper articles, TV and radio ads have hawked the promises and perils of shopping the vast electronic mall called the Internet - and you're eager to try out for you.

More and more known and not-so-known merchants are opening shop on the Web each day. If they are a substantial merchant, here's what to look for:

- A secure site that encrypts your credit card for transmission.
- Many ways to contact them as possible - phone number, fax number, street address, email address etc.
- A stated privacy policy that protects your personal information.
- A stated returns policy and satisfaction guarantee.

Does that mean that smaller sites that don't adhere the above should not be shopped? Of course not. Just be sure to get as much information from them as possible before spending your hard-earned dollars at their site.

There has never been a documented case of anyone having his or her credit stolen while buying something off the Web. Even Visa and MasterCard are telling their cardholders that the web is safe and to use their credit cards to buy from a merchant with a secure server. MasterCard has even established their Shop Smart program. At web sites that display the 'Shop Smart' icon, you can be assured that they provide the best available means to safeguard your transactions.

1.5. Threats of Shopping Online:

Many new owners are leery of purchasing items over the Internet. Alarmists would have us believe that credit card numbers are routinely plundered en route to online vendors. In truth, it’s much easier for someone to tap your phone line and listen as you place a call to catalog retailer than it is to intercept and decode credit card data that you send to an online retailer.

It’s far more important to ask “Can I trust this vendor to sell me a quality product and ship it as soon as possible?” Learning to evaluate an online retailer will help you avoid trouble. The first rule of online shopping is to look for vendor names you recognize and trust—vendors that advertise on television and in magazines, for example. You should also take into account the retailer’s web site. Larger, more reputable retailers often have well designed sites since they can afford a dedicated designer and web support staff. Smaller retailers are often identifiable by poor page design (including gaudy background patterns) and slow load times. You can’t always judge a book by its cover, however. Small Mom-and-Pop vendors may have poor design but offer impeccable service. If you’re apprehensive about the retailer, look for its customer service telephone number and give them a call. If there is no phone number listed, look for another vendor.

The next issue to consider is the retailer’s shipping and returns policies. Most vendors charge for shipping, offering you a choice of several shipping methods. In-stock items are usually packaged and sent either overnight, or for two-day or three-day delivery. Return policies vary from retailer to retailer. At a minimum, you should choose a retailer that allows you to return your product for immediately replacement if it is damaged. Try to find an online vendor that offers a satisfaction guarantee. This allows you to return the product if you change your mind or if you simply aren’t satisfied with the item.

Many reputable retailers have their online catalogs tied to their inventory database. This allows the vendor to report the product’s stock status within the product description page. Avoiding orders from retailers that fail to indicate whether or not the product is in stock. You should also shop around if the retailer reports your item is not in stock. After all, you want the product as soon as possible. The item you order may be on backorder, and may take weeks to ship. (The only time you might want to bend this rule is if you’d like to place a pre-order for a newly announced item, such as an iMac.)

The final task is to ensure that the online order form is secure. Although it’s common practice for large online retailers to use a secure order form, smaller vendors may eschew the technology since it is somewhat difficult to implement. As you input your order, look in the bottom left corner of your browser. If the order form is secure, you will see a small, locked padlock icon. This verifies that your credit card number and personal data will be encrypted before it is transferred to the retailer’s server.

1.6. Security Measures:

Aim of implementing of security in our system is to provide the authenticated exchange of information among communicating parties using digital certificates. We will use authentication mechanism to verify that user actually are who they claim are and have the authority to access the resources. The project will provide the reliable medium for the implementation of e-commerce e-government and online transaction. The project is also viable from the national aspect a e-commerce establishment within Pakistan is not as well established as compare to the International market.

1.7. Payment Systems:

The most common mode of transactions are

Cash

- properties: wide accept, convenient, anonymity, untraceability, no buyer transaction cost

Online credit card payment, Smart Cards

- Secure protocols: SSL, SET

Internet payment systems

- Electronic cash, digital wallets

CHAPTER 2

BRIEF DESCRIPTION OF APLICATIONS USED

2.1 INTRODUCTION:

2.1.1 THE CLIENT:

The applications we can develop with MySQL and PHP make use of a single client the Web browser. This is not the only possibility for Internet-based applications. For very sophisticated applications that require more client-side processing or that need to maintain state (we will talk about maintaining state later in the Introduction), a Java applet may be necessary. But unless we're coding something like a real-time chat program, client-side Java is completely unnecessary. So the only client we should be concerned with is the Web browser. The applications will need to render in the browser. As we probably already know, the primary language of browsers is the hypertext markup language or HTML. HTML provides a set of tags that describe how a Web page should look. If we are new to the concept of HTML, get on the Web and read one of the many tutorials out there.

It shouldn't take that much time to learn the basics. Of course, most browsers will accept more than HTML. There are all kinds of plug-ins, including RealPlayer, Flash, and Shockwave. Most browsers also have some level of support for JavaScript, and some of the newer ones can work with XML. But, like most Web developers, we will be taking a lowest-common-denominator approach. We're going to create applications that can be read in any browser. There will be no JavaScript, XML, or anything else that could prevent some users from rendering the pages we serve. HTML it is.

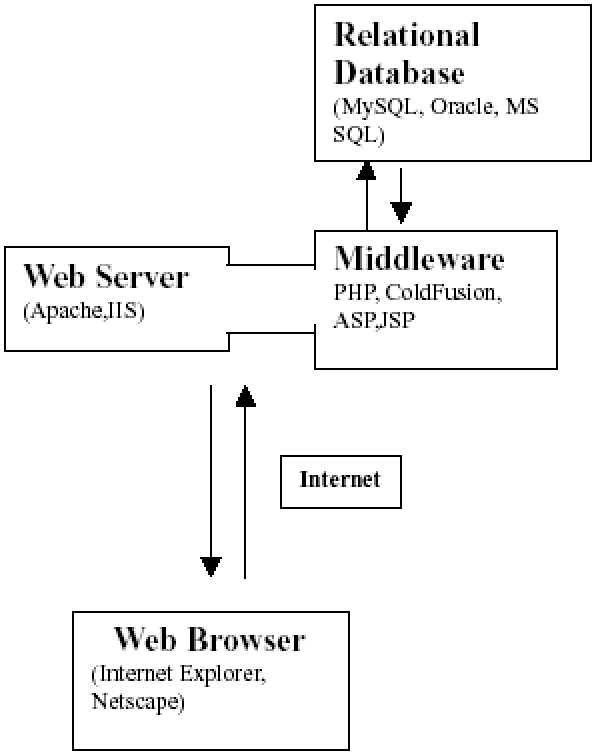
2.1.2 THE SERVER

Almost all of the work of Web applications takes place on the server. A specific application, called a Web server, will be responsible for communicating with the browser.

A relational database server stores whatever information the application requires.Finally, we need a language to broker requests between the Web server and the database server; it will also be used to perform programmatic tasks on the information that comes to and from the Web server. Figure I-1 represents this system. But of course none of this is possible without an operating system. The Web server, programming language, and database server we use must work well with Wer operating system.

2.1.3 OPERATING SYSTEM

There are many operating systems out there. Windows 98 and Macintosh OS are probably the most popular. But that’s hardly the end of it. Circumstances may have forced we to work with some obscure OS for the past few years. We may even be under the impression that wer OS is the best thing going. That’s fine. But if we’re planning on spending a lot of time on the Web and are planning on running applications, we’re best off getting to know either Windows NT/2000 or Unix. These two account for well over 90 percent of all the Web servers on the Web. It is probably easier for we to learn a little NT/2000 or Unix than it is to convince everybody else that the AS/400 is the way to go.



Architecture of Web applications

Which should we use? Well, this is a complex question, and the answer for many will be based partially on religion. In case we’re unaware of it, let’s take a moment to talk about the broad topics in this religious war. If we don’t know what we are talking about, here are the basics. PHP and MySQL belong to a class of software known as *open source*. This means that the source code to the heart of their applications is available to anyone who wants to see it. They make use of an open-source development model, which allows anyone who is interested to participate in the development of the project. In the case of PHP, coders all over the world participate in the development of the language and see no immediate pay for their substantial work. Most of the people who participate are passionate about good software and code for the enjoyment of seeing people like we and me develop with their tools.

This method of development has been around for some time, but it has gained prominence as Linux has become increasingly popular. More often than not, opensource software is free. We can download the application, install it, and use it without getting permission from anyone or paying a dime to anyone. Suffice it to say that Microsoft, Oracle, and other traditional software companies do not make use of this method of development. If we are not an open-source zealot, there are excellent reasons to choose NT/2000. Usually, the thing that steers people towards NT/2000 is inertia. If we or our company has been developing with Microsoft products for years, it is probably going to be easier to stay within that environment. If we have a team of people who know Visual Basic, we are probably going to want to stick with NT/2000. Even if this is the case, there’s nothing to prevent we from developing with PHP and MySQL. Both products run on Windows 95/98 and Windows NT/2000.

But in the real world, almost all PHP/MySQL applications are running off of some version of Unix, whether it be Linux, BSD, Irix, Solaris, HP-UX, or one of the other flavors. If we need to run these on Windows, minor alterations to the PHP scripts may be necessary. Most of the people who created PHP and MySQL are deeply involved with Unix, and most of their development is done on

Unix machines, so it’s not surprising that the software they have created works best on Linux, BSD, and other Unix boxes.

The major advantage of Unix is its inherent stability. Boxes loaded with Linux have been known to run months or years without crashing. Linux and BSD also have the advantage of being free and able to run on standard PC hardware. If we have any old 486, we can load it up with Linux, MySQL, PHP, and Apache and have werself a well-outfitted Web server. We probably wouldn’t want to put this on the Web, where a moderate amount of traffic might overwhelm it, but it will serve nicely as a development server, a place where we can test wer applications.

2.1.4 WEB SERVER

The Web server has what seems to be a fairly straightforward job. It sits there, running on top of wer operating system, listening for requests that somebody on the Web might make, responds to those requests, and serves out the appropriate Web pages. In reality, it is a bit more complicated than that, and because of the 24/7 nature of the Web, stability of the Web server is a major issue. There are many Web servers out there, but two Web servers dominate the market. They are Apache and Microsoft’s Internet Information Server (IIS).

2.1.5 INTERNET INFORMATION SERVER

IIS is deeply tied to the Windows environment and is a key component of Microsoft’s Active Server Pages. If we’ve chosen to go the Microsoft way, we’ll almost certainly end up using IIS. There is a certain amount of integration between the programming language and Web server. At this point, PHP 4 integrates well with IIS. As of this writing, there is some concern about the stability of PHP/IIS under heavy load, but PHP is improving all the time, and by the time we read this there may no longer be a problem.

2.1.6 APACHE

The Apache Web server is the most popular Web server there is. It, like Linux, PHP, and MySQL, is an open-source project. Not surprisingly, Apache works best in Unix environments, but also runs just fine under Windows. Apache makes use of third-party modules. Because it is open source, anyone with the skill can write code that extends the functionality of Apache. PHP will most often run as an Apache extension, known as an Apache module. Apache is a great Web server. It is extremely quick and amazingly stable. The most frequently stated complaint about Apache is that, like many pieces of Unix software, there are limited graphical tools with which we can manipulate the application. We alter Apache by specifying options on the command line or by altering text files. When we come to Apache for the first time, all this can be a bit opaque.

Though Apache works best on Unix systems, there are also versions that run on Windows operating systems. Nobody, not even the Apache developers, recommends that Apache be run on a busy server under Windows. If we have decided to use the Windows platform for serving Web pages, we’re better off using IIS.

But there are conditions under which we’ll be glad Apache does run under Windows. We can run Apache, PHP, and MySQL on a Windows 98 machine and then transfer those applications to Linux with practically no changes to the scripts. This is the easiest way to go if we need to develop locally on Windows but to serve off a Unix/Apache server.

2.1.7 MIDDLEWARE

PHP belongs to a class of languages known as *middleware*. These languages work closely with the Web server to interpret the requests made from the World Wide Web, process these requests, interact with other programs on the server to fulfill the requests, and then indicate to the Web server exactly what to serve to the client’s browser. The middleware is where we’ll be doing the vast majority of wer work. With a little luck, we can have wer Web server up and running without a whole lot of effort. And once it is up and running, we won’t need to fool with it a whole lot. But as we are developing wer applications, we’ll spend a lot of time writing code that makes wer applications work. In addition to PHP, there are several languages that perform similar functions. Some of the more popular choices are ASP, Perl, and ColdFusion.

2.1.8 RELATIONAL DATABASES

Relational Database Management Systems (RDBMSs) provide a great way to store and access complex information. They have been around for quite a while. In fact, they predate the Web, Linux, and Windows NT, so it should be no surprise that there are many RDBMSs to choose from. All of the major databases make use of the Structured Query Language (SQL).

Some of the more popular commercial RDBMSs are Oracle, Sybase, Informix, Microsoft’s SQL Server, and IBM’s db2. In addition to MySQL, there are now two major open-source relational databases. Postgres has been the major alternative to MySQL in the open-source arena for some time. In August 1999, Borland released its Interbase product under an open-source license and allowed free download and use.

2.2 Database Design with MySQL

2.2.1 Features MySQL Does Not Support

MySQL is a polarizing piece of software in the applications development community. It has aspects that many developers like: it's free, it doesn't take up a whole lot in the way of resources, it's very quick, and it's easy to learn compared to packages like Oracle and Sybase. However, MySQL achieves its speediness by doing without features common in other databases, and these shortcomings will keep many from adopting MySQL for their applications. But, for many, the lack of certain features shouldn't be much of a problem.

2.2.2 REFERENTIAL INTEGRITY

. A foreign key is a column that references the primary key of another table in order to maintain a relationship. The Contacts table contains a company_id column, which references the primary key of the Companies table.This column is a foreign key to the Companies table.It's easy enough to create tables with all the columns necessary for primary keys and foreign keys. However, in MySQL foreign keys do not have the significance they have in most database systems. In packages like Oracle, Sybase, or PostGres, tables can be created that explicitly define foreign keys.If the database system is aware of a relationship, it can check to make sure the value being inserted into the foreign key field exists in the referenced table. If it does not, the database system will reject the insert. This is known as referential integrity. To achieve the same effect in MySQL, the application developer must add some extra steps before inserting or updating records. For example, to be ultra-safe, the programmer needs to go through the following steps in order to insert a row in the Contacts table.

1. Get all of the values for company_id in the Companies table.
2. Check to make sure the value for company_id we are going to insert into

the Contacts table exists in the data we retrieved in step 1.
3. If it does, insert values.

The developers of MySQL argue that referential integrity is not necessary and that including it would slow down MySQL. Further, they argue that it is the responsibility of the application interacting with the database to ensure that the inserted data is correct. There is logic to this way of thinking. In general, in these applications, all the possible values are pulled from a database anyway and there’s very little opportunity for errors to creep into the system. For example, using PHP and HTML, the programmer might turn the Companies table into a drop-down box. That way the user can only choose a valid value.

2.2.3 TRANSACTIONS

In relational databases, things change in groups. As shown in a variety of applications, many changes require that rows be updated in several tables concurrently. In some cases, tables may be dropped as part of a series of statements that get the data where it needs to be. An e-commerce site may contain code like the following:

1. Insert customer into the Customers table.
2. Add invoice information into the Invoice table.
3. Remove a quantity of 1 of ordered item from the Items table.

When we’re working with a series of steps like this, there is potential for serious problems. If the operating system crashes or power goes out between steps two and three, the database will contain bad data. To prevent such a state, most sophisticated database systems make use of *transactions*. With transactions, the developer can identify a group of commands. If any one of these commands fails to go through, the whole group of commands is nixed and the database returns to the state it was in before the first command was attempted. This is known a COMMIT/ROLLBACK approach. Either all of the requests are committed to the database, or the database is rolled back to the state it was in prior to the transactions.

2.2.4 STORED PROCEDURES

The big fancy database systems allow for procedural code (something very much like PHP or Perl) to be placed within the database. There are a couple of key advantages to using stored procedures. First, it can reduce that amount of code needed in middleware applications. If MySQL accepted stored procedures, a single PHP command can be sent to the database to query data, do some string manipulation, and then return a value ready to be displayed in wer page. The other major advantage comes from working in an environment where more than one front-end is accessing the same database. Consider a situation where there happens to be a front-end written for the Web and another in Visual C++ accessible on Windows machines. It would be a pain to write all the queries and transactions in two different places. We'd be much better off writing stored procedures and accessing those from wer various applications.

2.3 INTRODUCTION TO PHP

2.3.1 PHP variables

These are variables available through PHP.

2.3.1.1 PHP_SELF

This is the address of the file being run. Usually, the full path is given from the ServerRoot directory, which is very useful when a form is both presented and processed in the same PHP page.

```
<?
        if(isset($submit))
        {
                //do some form processing here
                echo “thanks for the submission”;
        } else {
?>
<form name=myform method=post action=<?=$PHP_SELF?>>
<input type=text name=first_name> first name<br>
<input type=text name=last_name> last name<br>
<input type=submit name=submit value=submit>
</form>
<?  }
?>
```

Keep in mind that PHP_SELF always refers to the name of the script being executed in the URL. So in an include file, PHP_SELF will not refer to the file that has been included. It will refer to the script being run. It’s worth noting that PHP_SELF behaves strangely when PHP is run on Windows or as a CGI module. Make sure to look at phpinfo() to see the value of $PHP_SELF on wer system.

2.3.1.2 HTTP_POST_VARS

This is the array that contains all the variables sent through the POST method, usually through forms. We can access each individual variable as an element in an associative array (for example $PHP_POST_VARS[“myname”]).

2.3.1.3 HTTP_GET_VARS

This is the array that contains all the variables sent through the GET method. We can access each individual variable as an element in an associative array (for example $PHP_GET_VARS[“myname”]).

2.3.1.4 HTTP_COOKIE_VARS

All of the cookies sent to the browser will be readable in this associative array. This includes the session cookie. If we are wondering how wer cookies are behaving, phpinfo() will give we a quick readout of what wer browser is sending to the server.

2.3.2 Apache variables

Apache keeps track of dozens of variables. We can't include a complete list of variables here, as the variables we use will vary depending on wer current setup. Here are some of the ones we might use frequently in wer scripts. As we look at this list and phpinfo(), keep in mind that if we are not getting what we want out of wer Web server variables, we will need to make changes to wer server configuration, not PHP. PHP just passes the information along and cannot alter these variables.

2.3.2.1 DOCUMENT_ROOT

This variable returns the full path to the root of wer Web server. For most Apache users, this directory will be something like /path/to/htdocs. We use this variable throughout to make our applications portable. Take this include statement as an example:

```
include"$DOCUMENT_ROOT/ functions/charset.php";
```

By using the DOCUMENT_ROOT variable instead of an absolute path. If we are using a Web server other than Apache, DOCUMENT_ROOT may not be available. If we set the include_path directive in wer php.ini file, we will not need to worry about specifying any path in wer include statement—PHP will look through all of the directories we specify and try to find the file we indicate.

2.3.2.2 HTTP_REFERER

This variable contains the URL of the page the user viewed prior to the one he or she is currently viewing. Keep in mind when using HTTP_REFERER that not every page request has a referer. If someone types the URL into a browser, or gets to wer page via bookmarks, no referer will be sent. This variable can be used to present customized information. If we had a relationship with another site and wished to serve up a special, customized header for only those referred from that domain we might use a script like this.

```
//check if my user was referred from my_partners_domain.com
if(ereg (“http.*my_partners_domain.com.*” , $HTTP_REFERER))
{
        include’fancy_header.php’;
}
Else
{
        include’normal_header.php’;
}
```

Keep in mind that HTTP_REFERER is notoriously unreliable. Different browsers serve up different HTTP_REFERERs in certain situations. It is also easily spoofed. So we wouldn’t want to use a script like the preceding one to serve any secure information. I worked on a site where HTTP_REFERER was used to determine if a special GIF should be included in the header.

2.3.2.3 HTTP_USER_AGENT

Anyone who has built a Web page knows how important browser detection is. Some browsers will choke on fancy JavaScript, and others require very simple text. The user_agent string is wer key to serving the right content to the right people. A typical user_agent string looks something like this:

```
Mozilla/4.0 (compatible; MSIE 5.01; Windows 98)
```

We can then parse this string to get what we are looking for. We may be interested in PHP’s get_browser() function. Theoretically, this function will determine the capabilities of wer user’s browser so we can find out if wer script can safely serve out, for example, frames or JavaScript. The PHP manual has instructions for installation and use of get_browser(), but I do not recommend using it. Why? Using get_browser() we will be told that both Internet Explorer 5 for the PC and Netscape Navigator 4.01 for the Mac support CSS (cascading stylesheets) and JavaScript. But as anyone with client-side experience knows, writing DHTML that works on both of these browsers is a major task (and a major pain). The information we get from get_browser() can lead to a false sense of security. We’re better off accessing HTTP_USER_AGENT and making decisions based on the specific browser and platform.

2.3.2.4 REMOTE_ADDR

This is the IP address of the user that sent the HTTP request. REMOTE_ADDR is easily spoofed and doesn’t necessarily provide information unique to a user. We might want to use it for tracking, but it should not be used to enforce security.

2.3.2.5 REMOTE_HOST

This is the host machine sending the request. When I dial it up through my ISP (att.net), the REMOTE_HOST looks like this: 119.san-francisco-18-19rs.ca. dial-access.att.net. REMOTE_HOST is often not available.

2.3.2.6 REQUEST_URI

This is pretty much the same as PHP_SELF, except that it contains information in the querystring in addition to the script file name. It contains everything from the root path on.

2.3.2.7 SCRIPT_FILENAME

This variable contains the filesystem’s complete path of the file.

2.4 IMPORTANT PHP4 FUNCTIONS

2.4.1 MySQL API

There are a total of 33 MySQL functions available in PHP. We may find uses for some of the other MySQL functions in wer applications, but we probably won’t use all of them. For the sake of this listing, I’ll break the functions into the set we might use the most, and then the ones that we’re less likely to use extensively.

2.4.2 FREQUENTLY USED MYSQL FUNCTIONS

We will probably end up using the following functions frequently. We may want to dog-ear this page.

2.4.2.1 MYSQL_CONNECT( )

We can’t do anything with MySQL until we make the connection using the following function. int mysql_connect(str host, str username, str password) Most often we will be connecting to MySQL on localhost using a username and password assigned to we, the Web developer. The integer that this function returns to we is a connection identifier. We may need to track the connection identifier and use it with the mysql_db_select() function. It will typically look something like this:

```
$conn = mysql_connect(“localhost”, “username”, “password”) or
die (“Could Not Connect to Database”);
```

If MySQL is not installed on a standard port or if the mysql socket is not located in /tmp/mysql.sock, we can specify the port of socket location in the host string. For example:

```
mysql_connect(“localhost:/usr/local/mysql.sock”,
“username”, “password”);
```

Or, if the MySQL database in sitting on another machine, we can access it with the following

```
mysql_connect(“mymachine.mydomain.com”, “username”,
“password”);
```

We can also specify host, username, and password in the php.ini file. That way we could leave one or more of these arguments empty.

2.4.2.2 MYSQL_PCONNECT( )

The mysql_pconnect() function works exactly like mysql_connect() but with one important difference: The link to MySQL will not close when the script finishes running.

```
int mysql_pconnect(str host, str username, str password)
```

When we use this function the connection remains open, and additional calls to mysql_pconnect() will attempt to use these open connections when they run. This could make our scripts quite a bit faster. It is interesting to note what happens when mysql_pconnect() is run. The first time the script is run, PHP will ask for a connection, and MySQL will open a connection to the database. When that script finishes, the connection remains available. The next time a PHP page is requested, PHP will ask for a connection that is already open. If MySQL has one available, it will grant PHP the open connection. If there are no open connections available, a new connection will be opened.

Establishing a connection with the MySQL database will be about the slowest function in our scripts. If PHP can use a connection that has already been opened, there will be far less overhead in the application. In order for mysql_pconnect() to work, set the following lines in our php.ini file:

```
mysql.allow_persistent = On
mysql.max_persistent = -1; maximum number of ;persistent
links. -1 means no limit
```

Note that these are the defaults. We will probably want to limit the number of persistent connections if we use this method.

2.4.2.3 MYSQL_SELECT_DB( )

The mysql_select_db() function changes the focus to the database we wish to query.

```
int mysql_select_db (string database_name [, int link_identifier])
```

We can include the integer it returns in the mysql_query() function, but it is only really needed if we are connecting to more than one database. The second, optional argument is the link identifier retrieved from the mysql_connect()/ mysql_pconnect() function. It typically looks like this:

```
$db = mysql_select_db(“database_name”) or
die (“Could Not Select Database”);
```

2.4.2.4 MYSQL_QUERY( )

This mysql_query() function is probably the MySQL function that we will use most frequently in our scripts. int mysql_query (string query [, int link_identifier]) This function sends any query that we can put together to MySQL. It is important to understand that this function does not actually return the result of the query. It opens a cursor that points to the result set on MySQL. So if we were to do the following:

```
echo mysql_query(“select * from table”);
```

we would not get a meaningful answer, only the number that identifies the result set. Following mysql_query(), we will need to make use of one of the functions that actually retrieves the data from MySQL (mysql_fetch_row(), mysql_fetch_ array(), mysql_result()).

The optional second argument would be the result of either mysql_connect() or mysql_select_db(). It is typically used as in the following code sample. Note that a query can fail for any number of reasons. It is best to use mysql_error() to find out why the query failed.

```
$result = mysql_query(“select * from db”) or
die (mysql_error() );
```

2.4.2.5 MYSQL_FETCH_ARRAY( )

Once we have retrieved our result from a query, we will (more often than not) use mysql_fetch_array() to retrieve the rows from a query. array mysql_fetch_array (int result [, int result_type]) This function returns an associative array, with names of the select columns as the key. By default, mysql_fetch_array() will return each column in a row twice: the first will have an associative key, the second will have a numeric key. To tell PHP to limit the results to numeric results use MYSQL_NUM as the second argument. To get only the associative keys, use MYSQL_ASSOC as the second argument. This function returns FALSE when there are no rows left to fetch. The following will print the results of a query as a table:

```
$query =(“select * from table_name”);
```

2.4.2.6 MYSQL_FETCH_ROW( )

The mysql_fetch_row() function works almost exactly like mysql_fetch_array(), but it only returns a numeric array of the fetched row.

```
array mysql_fetch_row (int result)
```

2.4.2.7 MYSQL_INSERT_ID( )

Frequently the primary key of a MySQL table will be an auto_increment field. In such cases, after we do an insert query we may need to know the number MySQL assigned to the newly inserted row. int mysql_insert_id ([int link_identifier]) Following method would work equally well for getting the row that was just inserted into the database.

```
mysql_query(“insert into users (fname, lname) values (‘jay’, ‘greenspan’)
or
die (myslq_error());
mysql_query(“select max(user_id) from users”);
```

However, there is no guarantee that this script will return an accurate result. On a busy server, it is possible that an insert (perhaps run by another users accessing the script at nearly the same time) will occur between the time it took for these two queries to run. In such cases, our user will end up with bogus data. mysql_insert_id() returns the value of the auto_increment field associated with the specific copy of the script, so we know the number that it returns is accurate.

2.4.2.8 MYSQL_NUM_ROWS( )

A query can execute successfully, but still return zero rows in the result. This function will tell we exactly how many rows have been returned by a select query.

```
int mysql_num_rows (int result)
```

We might use it in a case like this:

```
$query = “select * from table_name”;
$result = mysql_query($query) or
die( mysql_error() );
if (mysql_num_rows($result) == 0)
{
        echo “Sorry, no results found.”;
} else{
//print results
}
```

2.4.2.9 MYSQL_AFFECTED_ROWS( )

This function is similar to the mysql_num_rows() function, but works for a different set of queries. It returns the number of rows in a table that are affected by an update, insert, or delete query.

```
int mysql_affected_rows ([int link_identifier])
```

This function is excellent for checking that a query we have run has actually accomplished something.

```
$query = “delete from table_name where unique_id = 1”;
$result = mysql_query($query) or
die (mysql_error());
$deleted_rows = mysql_affected_rows();
if ($deleted_rows == 0)
{
echo “no rows removed from the table.”;
} else {
echo “We just removed $deleted_rows row/rows from the database.”;
}
```

2.4.2.10 MYSQL_ERRNO()

If there is a problem with a query, this function will spit out the error number registered with MySQL.

```
int mysql_errno ([int link_identifier])
```

On its own, this isn’t terribly helpful. For the most part, we would only use this if we wished to use custom error handling. Better error messages come from mysql_error() which is discussed next.

2.4.2.11 MYSQL_ERROR()

This function should accompany every mysql_query() we run. string mysql_error ([int link_identifier]) As we can see in code samples we make use of mysql_ error with the die statement. mysql_query(“select * from my_table”) or
die (mysql_error()) Without it, we will only know that our query has failed. We won’t know if we’re searching for a database that doesn’t exist or if we’re trying to insert a string into a numeric field, or have made some sort of syntactical blunder.

2.4.2.12 MYSQL_RESULT()

This function, which grabs the contents of a specific cell, should be used sparingly. mixed mysql_result (int result, int row [, mixed field]) It’s relatively slow and can almost always be replaced by mysql_fetch_array(). However, if we need to grab contents from a single cell it can be convenient. The second argument will always be the number of the row we are accessing. The third can either be the numeric offset of the column or the column name. Here’s an example of how we could use mysql_result(). In this case, we’re running a simple count(), so there is only one value to be accessed. Using mysql_result() is a bit easier than mysql_fetch_array().

```
mysql_connect(“localhost”, “username”, “password”);
mysql_select_db(“test”);
$result = mysql_query(“select count(*) from users”) or
die ( mysql_error() );
echo mysql_result($result,0,0);
```

If we have many rows, or even many columns, that need to be retrieved we should use mysql_fetch_array().

CHAPTER 3

INTRODUCTION TO PROJECT

3.1. AIM

The aim of this project is the study and implementation of electronic payment system on the World Wide Web with special emphasis on security to enable clients to securely and authentically make online transactions using our scratch cards.

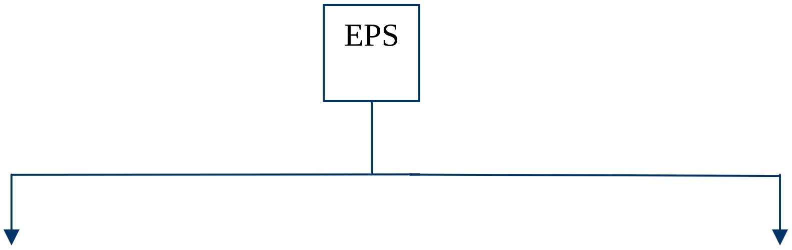
3.2. Goals and Objectives:

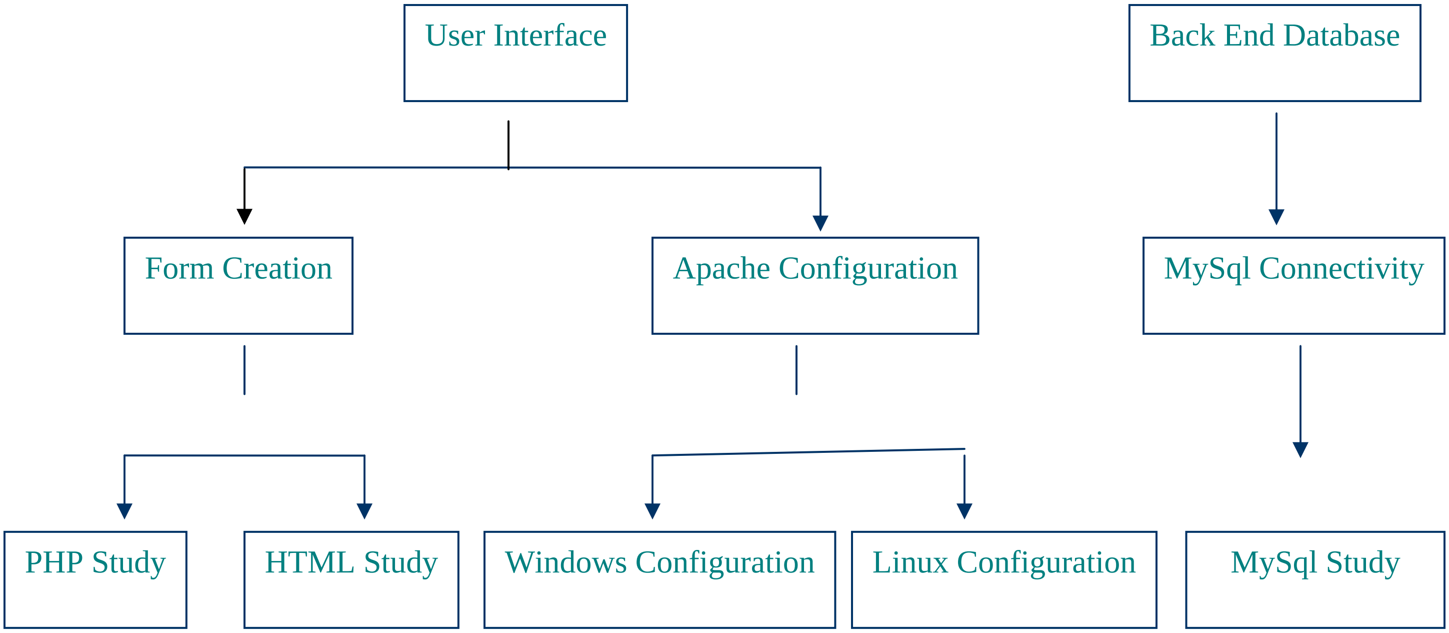
- Bring efficiency in the Payment system in Pakistan's Banking sector in a cost effective manner.
- Propose solutions to enable future growth of e-banking / e-Commerce.
- Build a strategy to bring our product at a technology par which could compete with other such products.
- Identify and enable Key EFT (Electronic Funds Transfer) products.
- Design and development of middle-ware based transaction processing.
- Deposit and payment automation solutions.
- Electronic payment networks design and implementation.
- System Integration.
- Remote banking and other services.
- PIN and Cards management systems.
- Host authorization.

3.3. Challenges:

- Communication infrastructure Build Up.
- Harnessing distributed host application environment.
- Transaction exchange platform setup.
- Developing industry alliances.
- Operational maturity.
- Adoption of an enterprise blue-print for the bank.

3.4. Dependencies:





3.4.1. Real Time Implementation

To make the product a real time business Software application a reasonable amount of space is required over the web .In order to cope the situation we had to acquire 50 MB space over the internet .If the database grows exponentially we have to acquire more space.

3.4.2 Configuration of Web server

Configuration of any web server requires a detailed knowledge of networking, Operating system and the scripting language required to run on the server. Apache is a free ware web server available for non commercial use .The normal installation of Apache web server comes with few DLL files missing like php4nt.dll and others, moreover in the configuration file php modules were required to be added. All this was learnt by time and experience .The nest problem faced was to configure Apache over Linux Platform it was a big challenge as we have had never worked on Linux Platform.

3.4.3. Understanding of PHP and MySql

The curriculum designed for Undergraduate courses does not include modern/new languages so we had to put extra effort in learning the new horizons of PHP and MySql along with our degree courses. PHP is one of the most emerging web application development language specifically designed for security purposes. Mastering the PHP was one of the major issues we had to undertake for our project to progress .Similarly MySql is free ware software for database development, design and monitoring. Learning MySql was another important task to undertake for keeping the user records correct, secure and invulnerable to outside attacks.

3.4.4. Designing of Databases

Database design needs to take special care as incorrectly defined primary, foreign and candidate keys can make the database a mess. Different tables are to be created each with its own primary, foreign and candidate keys. Every time a new table is created, when new vendors/service providers registers with our system.

3.4.5. Integration over Server

The most difficult part of all was to integrate the whole system and run it over the internet. Database connectivity of PHP and HTML forms with MySql Databases was achieved using Cpanel6.Understanding Cpanel was another difficult task confronting us.

3.5. Risk management

Risk management is very important for any type of project. We shall be monitoring the following risk factors. PERT charts and other charts are to be maintained to monitor the risk.

2.5.1. Time

The time allocated to specific tasks and time allocated to overall project completion should be taken into consideration.

2.5.2. Budget

Project should remain within the budget specified and long delays should be avoided as they cost extra amount to be spent.

2.5.3. Applicability

Project should be made keeping in view the demands of the suppliers. It should fulfil all the requirements specified by the suppliers.

2.5.4. Satisfying user requirements.

User should not be engaged in long boring forms rather he should be provided as user friendly environment as possible.

3.6. Security

Transaction made are secure so that the confidence of the user is not shattered. .

The other risks and challenges which require attention are as under:

- Communication infrastructure Build Up.
- Harnessing distributed host application environment.
- Transaction exchange platform setup.
- Developing industry alliances.
- Operational maturity.
- Adoption of an enterprise blue-print for the bank (if possible).

3.7. System Requirement:

System requirements includes software requirements and hardware requirements

3.7.1. Software Requirements:

1 Client Requirements

1.1. Any one of the following platforms.

Red Hat Linux, Ms-Windows 98, Ms-Windows NT, Ms-Windows 2000 etc

1.2. Any one of the browsers

Ms-Internet Explorer, Netscape Navigator, Mozilla etc

2 Server Requirements

2.1 Any one of the following Servers

Apache Web Server, Personal Web Server, Internet information Services 5.0 ,Internet information Services 5.1 (5.1) etc

2.2. Any one of the following platforms

Linux, Ms-Windows 98, Ms-Windows NT Server, Ms-Windows Web Server, Ms-windows 2000, Ms-windows XP.

2.3. Any one of the following browsers

Mozilla, Ms-Internet Explorer, Netscape Navigator etc

3.7.2. Hardware Requirement:

1 .Minimum Requirement

a)	CPU	Pentium-I, Pentium-II
b)	RAM	64 MB
c)	HDD	2 GB
d)	Modem	56 k
e)	Color SVGA 14”	Monitor
f)	Standard Mouse	
g)	Standard Keyboard	

2. Standard Requirement

a)	CPU	Pentium-III
b)	RAM	128 MB
c)	HDD	4.2 GB

d) Modem 56 k

e) Color SVGA 14” Monitor

f) Standard Mouse

g) Standard Keyboard

3.8. PLATFORMS

1) Red Hat Linux
2) Windows 2000 Server

3.9. SOFTWARE USED

The Softwares used are

1) MySql

2) Apache Web Server

3) PHP (HyperText Pre Processor)

4) CPanel6

5) Flash MX

6) Swish 2.0

7) Microsoft FrontPage 2003

8) Rational Rose 2000 Enterprise Edition

9) HTML (Hyper Text MarkUp Language)

10) Web Host Manager

3.9.1. WHY MySql:

Main Features of MySQL

The following are the some of the important characteristics of MySQL:

- Fully multi-threaded using kernel threads. This means it can easily use multiple CPUs if available.

- Compatibility with C, C++, Eiffel, Java, Perl, PHP, Python and Tcl APIs.
- Works on many different platforms.
- Many column types: signed/unsigned integers 1, 2, 3, 4, and 8 bytes long, `FLOAT', `DOUBLE', `CHAR', `VARCHAR', `TEXT', `BLOB',`DATE', `TIME', `DATETIME', `TIMESTAMP', `YEAR', `SET', and `ENUM' types.
- Very fast joins using an optimized one-sweep multi-join.

- Full operator and function support in the `SELECT' and `WHERE' parts of queries. For example:

```
mysql> SELECT CONCAT(first_name, " ", last name) FROM tbl_name
       WHERE income/dependents > 10000 AND age > 30;
```

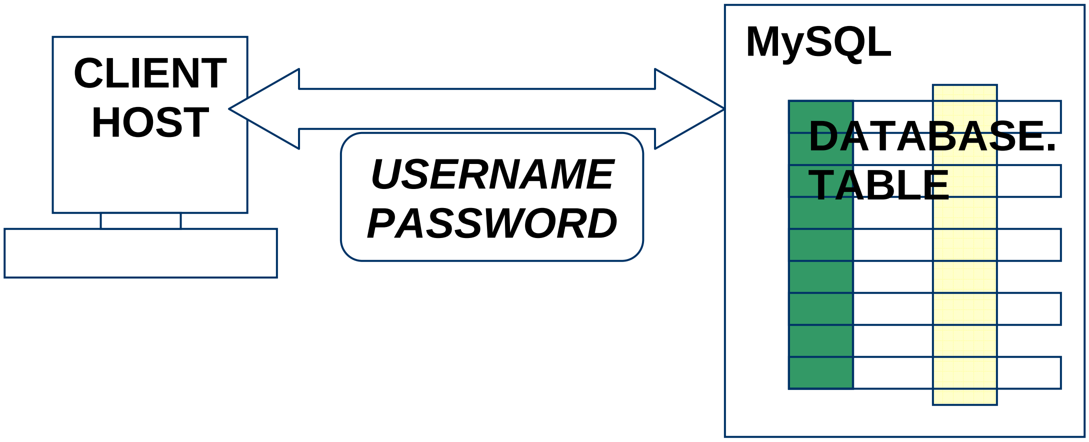
- SQL functions are implemented through a highly optimized class library and should be as fast as possible! Usually there isn't any memory allocation at all after query initialization.

- Full support for SQL `GROUP BY' and `ORDER BY' clauses. Support for group functions (`COUNT()', `COUNT(DISTINCT ...)', `AVG()',`STD()', `SUM()', `MAX()' and `MIN()').

3.9.2.MySQL Security

- A privilege and password system that is very flexible and secure, and allows host-based verification. Passwords are secure because all password traffic is encrypted when you connect to a server.

- Username / password (and optionally client hostname) checked before any commands are accepted;
- Different access for each operation (SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, …)
- Access is allow / deny at a global, database, table or column level



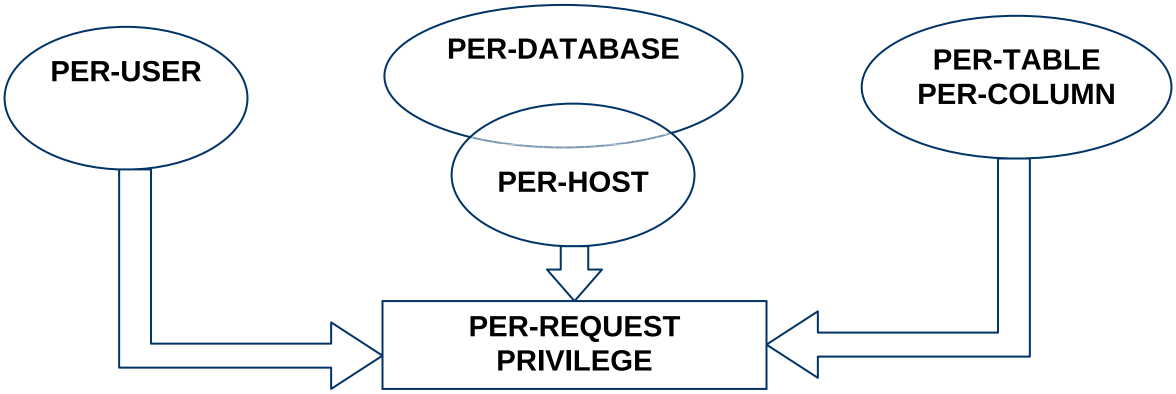
- For given SQL statement, permissions are sum of:
-
    - global 'user' permissions;
    - permissions specific to table or column;
    - database (i.e. 'all tables') restricted by host.

- Very fast B-tree disk tables with index compression.
- Up to 32 indexes per table are allowed. Each index may consist of 1 to 16 columns or parts of columns. The maximum index length is 500 bytes

(this may be changed when compiling MySQL). An index may use a prefix of a `CHAR' or `VARCHAR' field.

- Fixed-length and variable-length records.
- In-memory hash tables which are used as temporary tables.

- Handles large databases. We are using MySQL with some databases that contain 50,000,000 records and we know of users that uses MySQL with 60,000 tables and about 5,000,000,000 rows



- All columns have default values. You can use `INSERT' to insert a subset of a table's columns; those columns that are not explicitly given values are set to their default values.
- A very fast thread-based memory allocation system.
- No memory leaks. MySQL has been tested with Purify, a commercial memory leakage detector.

3.9.3. Database Administration:

- Full support for several different character sets, including ISO-8859-1 (Latin1), big5, ujis, and more. For example, the Scandinavian characters `a*', `a"' and `o"' are allowed in table and column names.

- `DELETE', `INSERT', `REPLACE', and `UPDATE' return the number of rows that were changed (affected). It is possible to return the number of rows matched instead by setting a flag when connecting to the server.

- Function names do not clash with table or column names. For example, `ABS' is a valid column name. The only restriction is that for a function call, no spaces are allowed between the function name and the ` (' that follows it. *Note Reserved words::.
- Clients may connect to the MySQL server using TCP/IP Sockets, Unix Sockets (Unix), or Named Pipes (NT).

Chapter 4

System Model

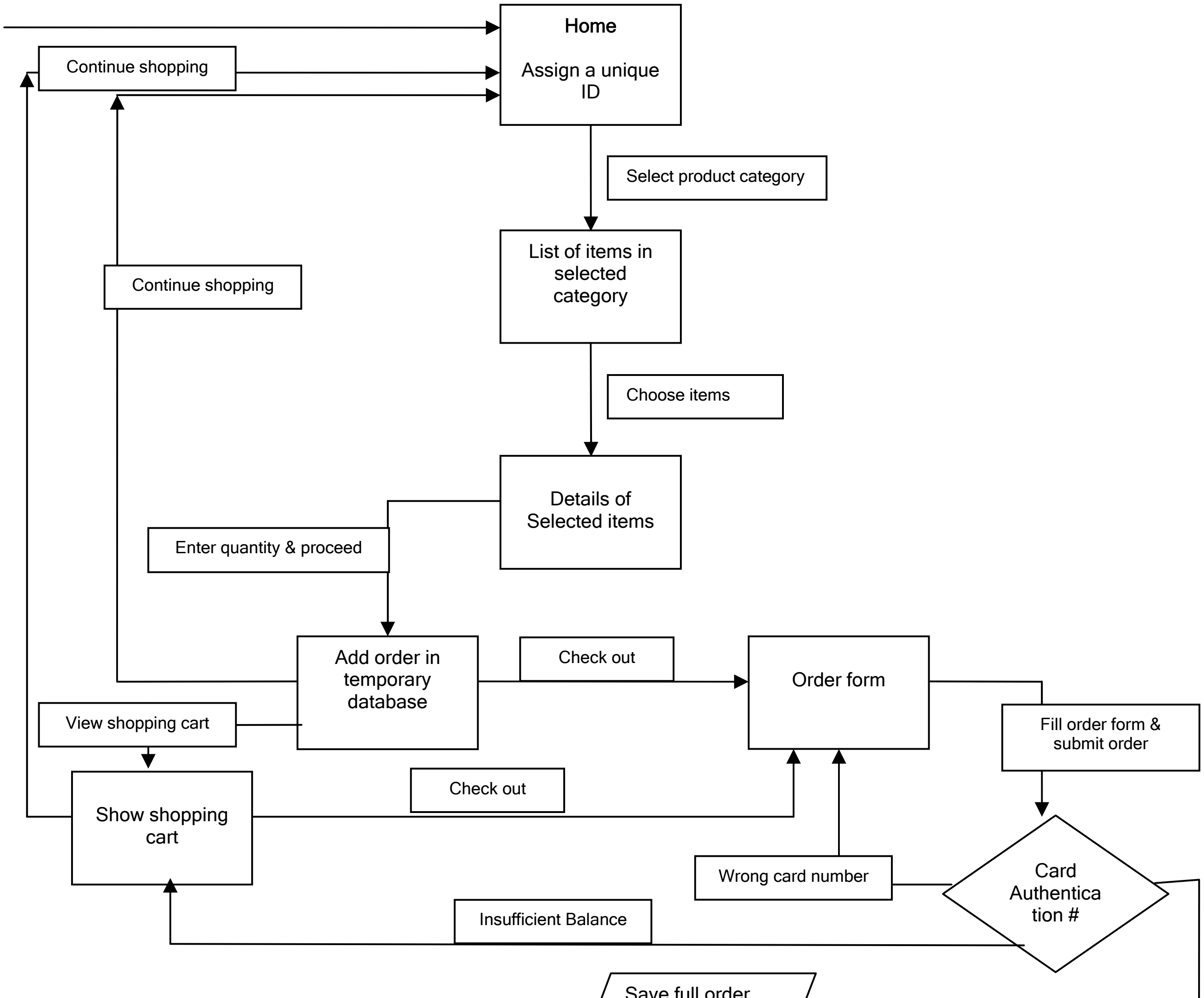
4.1 Introduction

There is a new concept in the modern era of the history of the internet to establish online shopping. There are many methods to purchase or to sell different products either from the departmental stores or computer shops on net such methods are credit card system, visa card system etc., but we (group of

Digital Bazaar online shopping) introduce a new meaningful and convenient way to purchase or sell products on online.

To enhance the mechanism of online shopping a different mode of payment method has been developed. The revolution in telecommunication system actually promotes the payment method of purchasing and selling products on online. The evolution of scratch card system in the history of internet shopping is divulged as the leading phenomenon. Fixed amount of different cards are published by the company having unique numbers have played a unanimous role especially for those customers who lived in remote areas. The purchasers of these cards are either shopkeepers or general public.

4.2 Development



3.3 Deliverables

3.4 Payment method:

4.3.Scratch card numbers:

The scratch cards are available on different shops and the addresses of these shops are mentioned in our site. The basic logic is to generate unique numbers of 11 digits in random order form that are already stored in our database. The scratch card box must be filled by the customer; the site cannot permit to leave the box empty or wrongly filled.

There is an admin section that can be handled by the administrator of the site. All the changes including updates, deletes, insert will have done through admin section. The orders are booked by the customers and the requirement can be fulfilled by the administrator through various means and media. Easy deliveries of these orders are the prima fascia of our company.

4.4. Helping To Establish and Build Relationships with Customers:

Gravity Market makes it easy for your customers to share information with you. And that helps you create a compelling and satisfying customer experience.

Net concepts helps you develop personalized shopping experiences that put the information your customers need within easy reach:

- personalized welcome messages and content

- user-friendly login/out with password reminder
- account information editing (available whether making a purchase or not)
- purchase history reporting
- tax and shipping & handling cost calculation
- auto-fill-in of order forms for existing customers
- separation of delivery and billing addresses
- Up to five delivery addresses per account.

4.5. Banners, Top Stories, View Shopping Cart

Extend your customer experience beyond your site with:

- Check Balance of Yours Scratch Card
- Contact information
- Mention places from where to purchase Digital Bazaar Scratch Cards.

4.6.Creating Customized Shopping Experiences

Net concepts e-commerce solutions are customized to suit your customers' needs and expectations.

We develop browsing options around your product range and categories to maximize customer convenience and recognition. Your customers will be able to review products in a moderated way that lets you gather feedback and enhance product related content.

- Search functions let your customers find exactly what they are looking for fast.

- **Merchandising**

Highlight products daily or with every page refresh with rotating featured products. And maximize exposure to selected items with a specials page accessible from anywhere in your online store.

Got a merchandising idea? Talk to us about custom functionality.

4.7. Secure Shopping Cart Functionality:

Net concepts' e-commerce shopping cart software allows your customers to:

- add and remove products
- add and remove news
- check orders
- view cart contents
- save cart contents for purchase on a return visit

All shopping cart transactions are secured with Secure Socket layer (SSL) encryption of all transmitted data.

4.8. Steps to start shopping online

After seeing all the .com e-commerce advertisements you decided to buy try buying things online. But how do you start? Where to buy? What to look for? How do you find the best place for shopping a particular product in mind? How do you pay for online shopping? Is online shopping safe?

Let us start with payment which is the most important thing in online shopping. If you don’t have a credit card it is time to get one. But we introduce a scratch card system and find it more convenient. You may find others also good depending upon your way of usage. Sending by snail mail is not a very good idea. This is because the shipment of the products you wish to buy will start only when the merchants are sure that they get the money. In this case the shipment will start only after the cheque /DD is enchased which will cause a lot of time delay. Now onwards let us assume you got a credit card or a method for paying the online merchants.

4.8.1 Delivery time/method/charge

Compare the charges and times across the merchants and choose the one which is best for you. Most of the major cities will not have any problem in delivery. If you are not in a city then check, whether the delivery is possible to your place before ordering.

4.8.2. Secure payment

If they are taking scratch card numbers online it has to be through a secure line so that no body else in the internet would intercept and get your scratch card information. Though such interception on your particular transaction is highly unlikely it is better to be on the safer side.

Other general comparisons which you do on normal shopping apply here also like price/discount comparison, brand names etc. Choose the one which is best in all ways for you. Do a lot of window shopping.

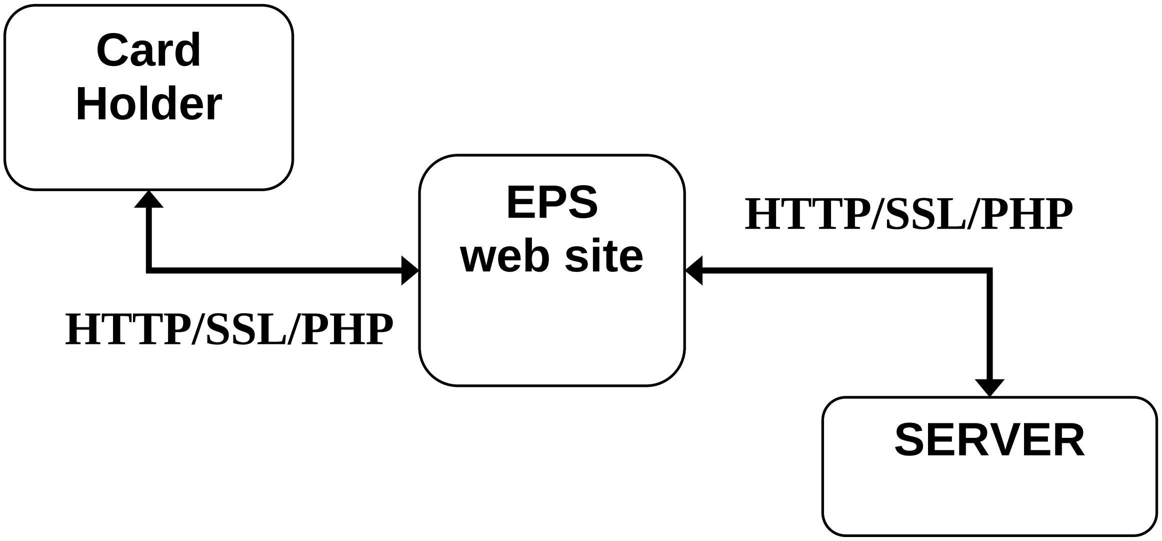
If you have already browsed through a standard e-commerce site you can skip this part. After choosing the website go to the site and locate the item you wanted to purchase. Most of the e-commerce sites have the standard shopping cart. If you see any sign of shopping cart symbol/link nearby click on it to add your item for purchasing. You can browse around the site if you want to look at other products. Generally all the sites should have the price tag immediately next to each item.

Once you decide to buy the item go to their payment page choose the payment method for you. Then verify the total amount you are supposed to pay. If you are paying through scratch details. After this the site usually verifies the validity of scratch card information which will take in a few seconds of time. If this step is completed without any problem the site may offer an order number for tracking. This number can later be used to see at which stage your order is in.

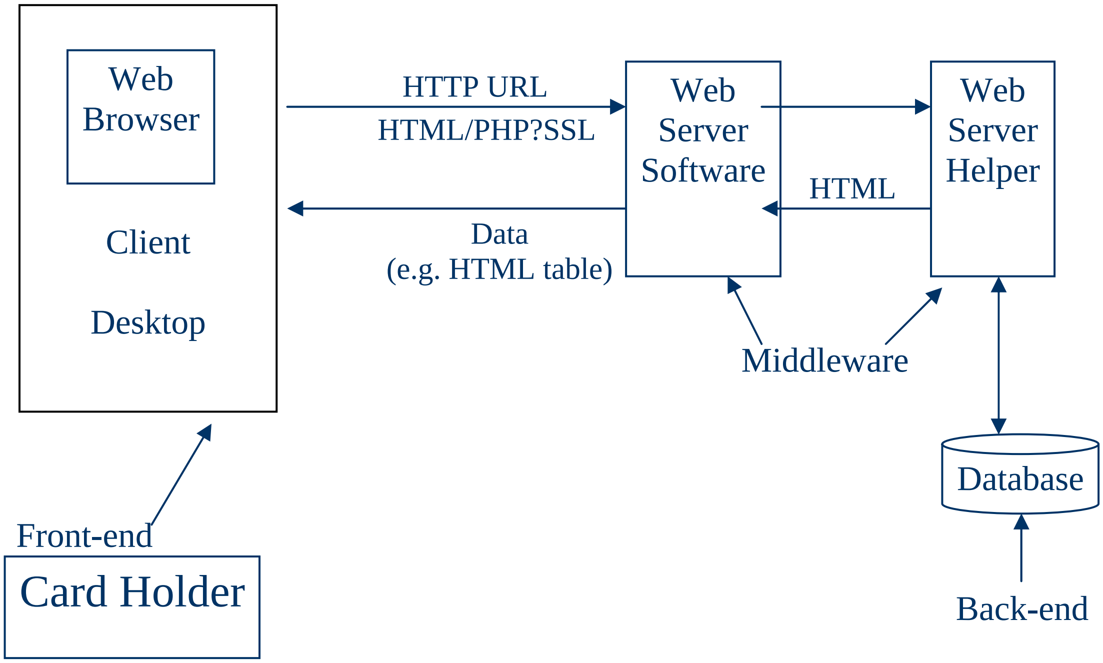
After this just wait for your item to reach home and that's it. You have done your first online shopping. Sometimes it is a good idea to register at a site if you are going to be a regular online shopper. For some sites it is a must to register before you order anything. It may get you some discount or points with prices etc.

4.9.System Architecture

Level 0 Architecture:



Level 1 Architecture



So let's set the scenario. We have a web page that pulls some data out of a database. User requests this page from his browser, the request is sent to the web server which in turn calls a PHP script. The PHP script is executed by the PHP preprocessor; it pulls data from the database. The results are then massaged by the rest of the PHP script and turned into HTML. The final HTML gets sent back to the user's browser.

Got that? Let's look at this step by step:

1. User clicks on a link to from his web browser; his web browser sends a request for link e.g. **http://www.eps.com/default.php**.

2. Apache gets the request for **default.php**. It knows that **.php** files are handled by the PHP preprocessor, so it tells PHP to deal with it.

3. **default.php** is a PHP script that contains commands. One of these commands is to open a connection to a database and grab some data. PHP knows how to talk to the database, so it does its thing.

4. The data comes back from the database, and de.php does something to format the data. Typically this would be to make it look pretty before formatting it into HTML.

4.9.1. The HTML goes back to Apache.

6. Apache sends this back to User’s browser, as the response to his request. User now sees a pretty web page containing some information from a database.

Again, that's not 100% correct but it's enough to understand what goes on :).

Operational Scenarios:

EPS may be used in many ways such as:

1. providing the end user means to transfer his/her money such as utility bills without going to respective offices and saving a lot of time.

2. Helping the end user in personal time management and self-organization.

Business Context

It is full fledge commercial project intended for the real market with wide scope of application in e-commerce. Its main goal is to get the Developers acquainted with Business Oriented Software Engineering basics. The second goal is to help the Developers in learning latest trends and techniques in software field.

CHAPTER 5

OBJECT ORIENTED ANALYSIS AND DESIGN

5.1 Use Case Modeling

The Use Case Diagram models the software. The diagram presents a general outline of the interaction of various external users or actors with the software. The diagram consists of a number of actors and use cases which are initiated by the actors. A Use Case Report is given which outlines the different functionalities of the software and explains how actors interact with the software and how use cases are invoked and processed, as well as the flow of events involved in each use case.

5.1.1 List of Actors

- Administrator
- User (Client)
- PHP Server
- Database Server

5.1.2 Use Case List

- Login (User)
- Register
- Search for Items
- View Account
- Recharge Account
- Store Items
- Transactions
- Login (admin)
- View Transactions
- Ship Items

- Search for User
- View User record
- View Cards Database
- Add new Cards
- Remove Users

5.2 Use Case Report

5.2.1 Description:

The Use Case diagram describes how the external user such as the administrator or client interacts with the software. The administrator may invoke different use cases, which allow the administrator to log into the software, view records and upgrade them. Similarly the user may issue a new certificate or revoke an old one.

5.2.2 Use Cases:

Login: (User)

Brief Description: The Login Use Case allows a registered User to login to the software.

Precondition: The user must be registered user having his/her own ID and user password.

Flow of Events: User types in his or her name and password. The software checks for validation. If the user id is correct the user may logon otherwise the user is prevented from access to the software.

Post Conditions: Once user is logged on, user may use software to access different files and portions of the software.

<u>Register</u>

Brief Description: A new user invokes this use case when he/she uses the software for the first time or want to open another account. It is used to register a new user with our software.

Precondition: The user must be having a scratch card of EPS with a card and a pin number.

Flow of Events: The user enters various fields required to complete the registration. If there is an error connecting to the database the system displays an error message to the user.

Post Conditions: Once the registration has been completed successfully, the Login use case is invoked.

Search For Items

Brief Description: This is used by the user to search/view the records of all the items stored in the database.

Precondition: The user must be logged in to the software.

Flow of Events: The user clicks on various links which shows the items stored in the database. If there is an error connecting to the database the system displays an error message to the user.

Post Conditions: The user can select the items of their choice.

View Account

Brief Description: This is used by the user to view his profile, or all the transactions that he/she has done so far.

Precondition: The user must be logged in to the software.

Flow of Events: The user clicks on various links which shows their records. If there is an error connecting to the database the system displays an error message to the user.

Post Conditions: The user can go to any link from here.

Recharge Account

Brief Description: This is used by the user to recharge his/her account.

Precondition: The user must be logged in to the software. User must have a new scratch card number and pin.

Flow of Events: The user views his/her record and click the recharge account link. The user fills the fields using a new scratch card number and pin and recharges his/her account. If there is an error connecting to the database the system displays an error message to the user.

Post Conditions: The user’s updated profile is displayed.

Store Items

Brief Description: This is used by a user to store items in a shopping cart.

Precondition: The user must be logged in to the software.

Flow of Events: The user can select different items of their choice and place them in their shopping cart. If there is an error connecting to the database the system displays an error message to the user.

Post Conditions: The selected items in the cart will be shipped to the user,

Transactions

Brief Description: This use case allows the user to buy all the selected items that are in their cart.

Precondition: The user must be logged in to the software. Items must be placed in the shopping cart.

Flow of Events: The user selects items of his/her choice. If there is an error connecting to the database the system displays an error message to the user.

Post Conditions: The balance of the user is deducted and the items are shipped to the user later on

<u>Login (Admin)</u>

Brief Description: This is used by the administrator to login to access the software.

Precondition: A login ID and password of an administrator is required.

Flow of Events: The Administrator types in the ID and password. The software checks for validation. If the ID and password are correct the administrator may logon otherwise the software prevents access to the software.

Post Conditions: Once the administrator is logged on, administrator may use the software to access the

<u>View Transactions</u>

Brief Description: This is used by the administrator to view the transactions carried out by the users.

Precondition: An administrator of the system should be logged in.

Flow of Events: The administrator uses the link to view the transactions carried out by the users on the current date of the server. If there is an error connecting to the database the system displays an error message to the administrator.

Post Conditions: The administrator can select the pending transactions and ship the order.

<u>Ship Items</u>

Brief Description: This is used by the administrator to ship the items selected by the users.

Precondition: An administrator of the system should be logged in.

Flow of Events: The administrator selects the transactions after viewing them, and then stores it to the database. If there is an error connecting to the database the system displays an error message to the administrator.

Post Conditions: Once the transactions have been selected, it is inserted into the database.

<u>Search for user:</u>

Brief Description: This is used by the administrator to search for a particular or all of the users of the system.

Precondition: An administrator of the system should be logged in.

Flow of Events: The administrator follows the link to open the search page and enters the login ID of the user to start the search. If there is an error connecting to the database the system displays an error message to the administrator.

Post Conditions: The user's ID is displayed, and the administrator can click on it open the record of the user.

View cards Database

Brief Description: This is used by the administrator to view all the cards of the system stored in the database.

Precondition: An administrator of the system should be logged in.

Flow of Events: The administrator follows the link to view all the cards stored in the database to see which user is using which card, and how many cards are in use. If there is an error connecting to the database the system displays an error message to the administrator.

Add new cards

Brief Description: This is used by the administrator to add new cards to the database.

Precondition: An administrator of the system should be logged in.

Flow of Events: The administrator enters the data required to create new cards in the given text fields. The system checks the entered data. If it is not valid an error message is displayed to the administrator.

Post Conditions: Once the administrator enters all the relevant data the system generates new card numbers and passwords and stores them in the database.

Remove Users

Brief Description: This is used by the administrator to remove the selected users from the system.

Precondition: An administrator of the system should be logged in.

Flow of Events: The administrator follows the link to search for the user and then select the user to delete him/her from the system. If there is an error connecting to the database the system displays an error message to the administrator.

Post Conditions: Once the administrator clicks on the button, the selected user’s profile is deleted from the database.

5.2.3 Use Case Diagrams

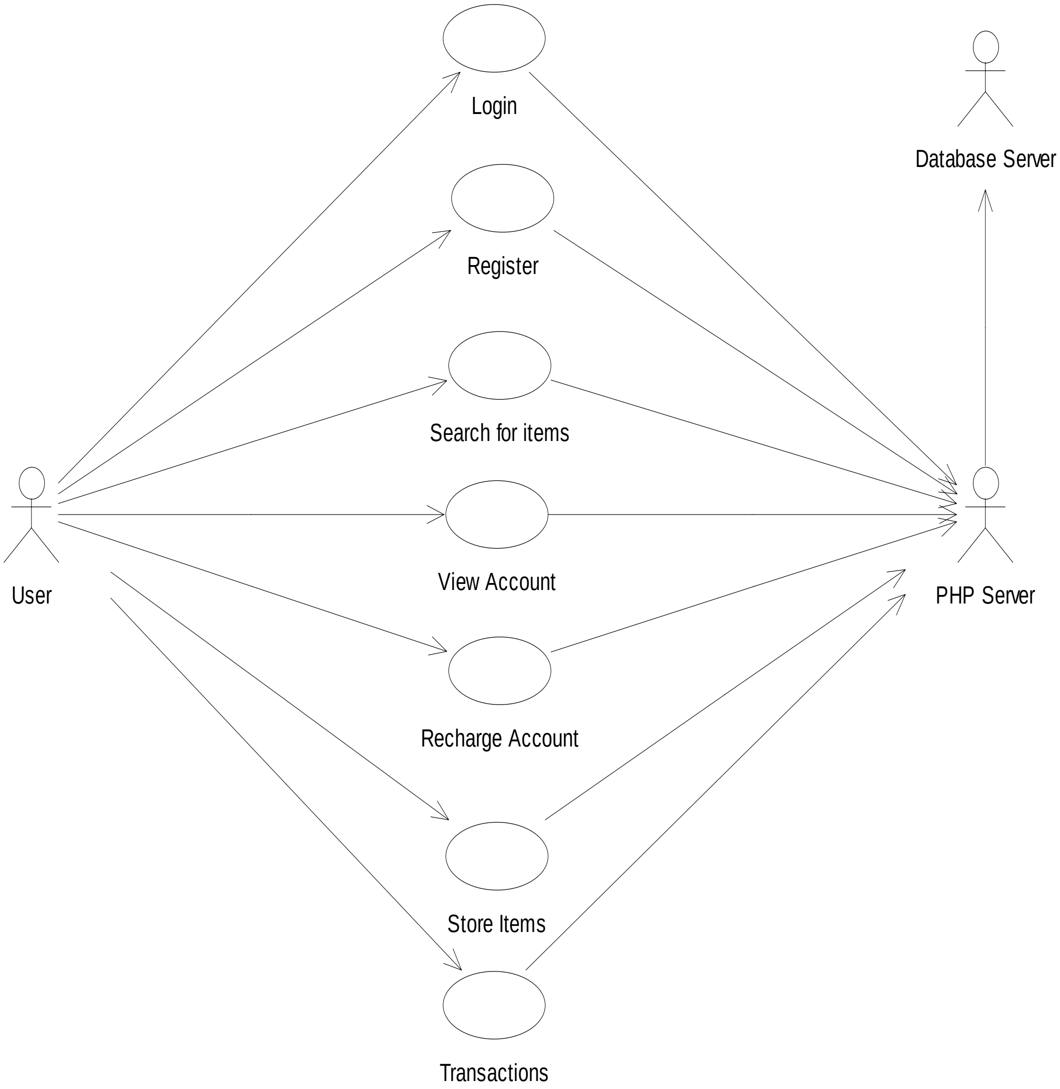


Figure 5-1 User’s Use Case Diagram

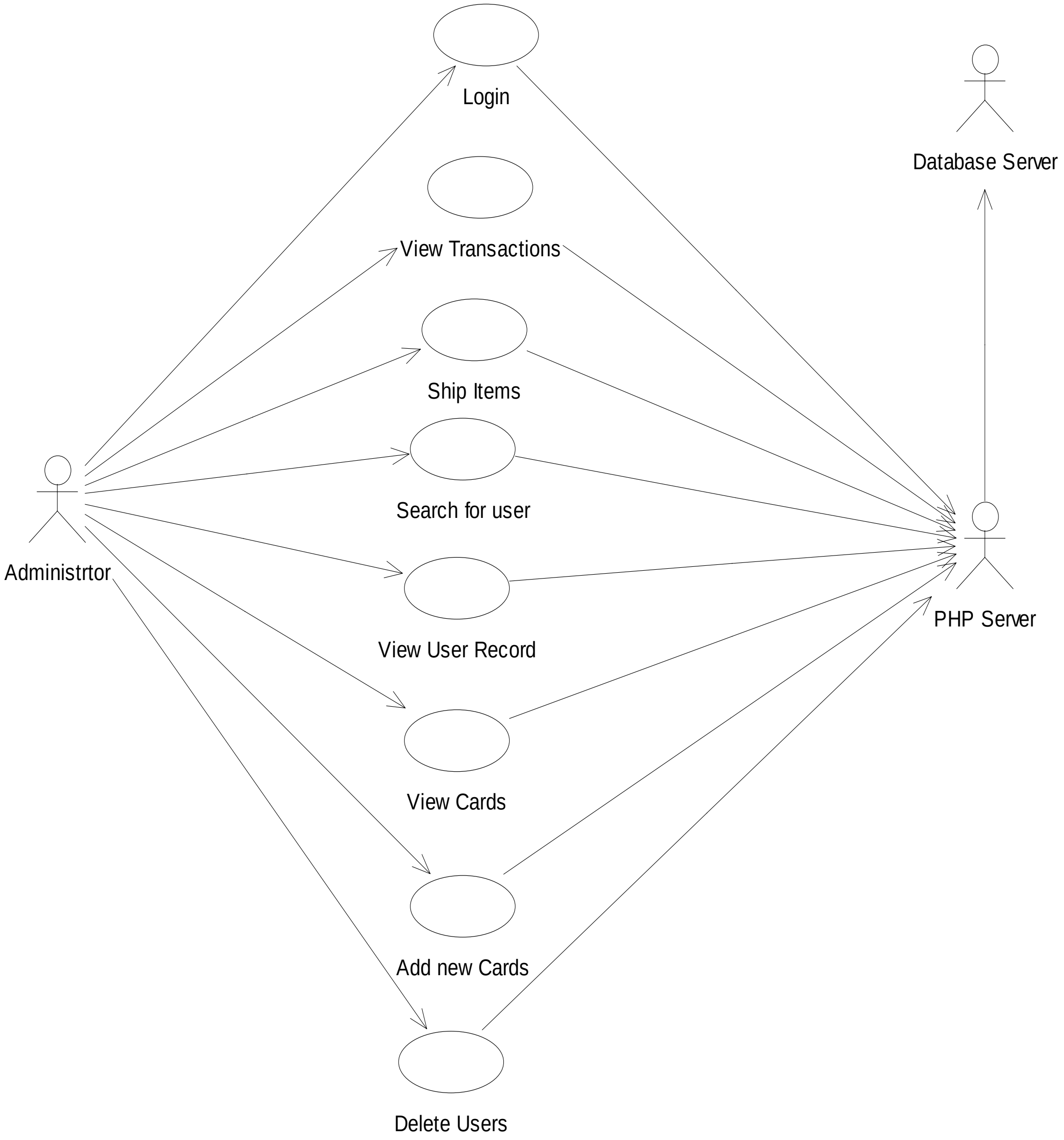


Figure 5-2 Administrator’s Use Case Diagram

5.3 Sequence Diagrams

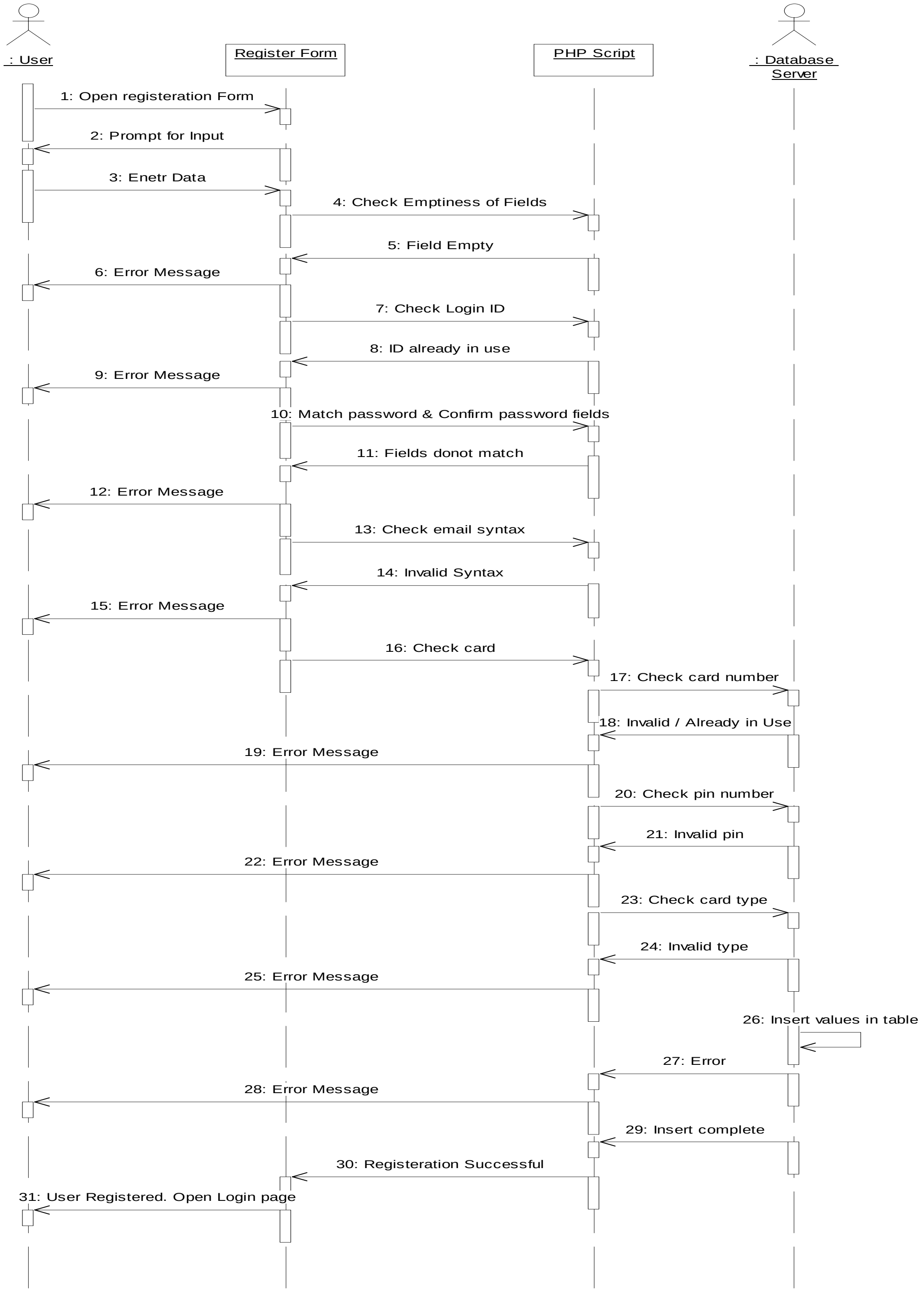


Figure 5-3 User's Registration Sequence Diagram

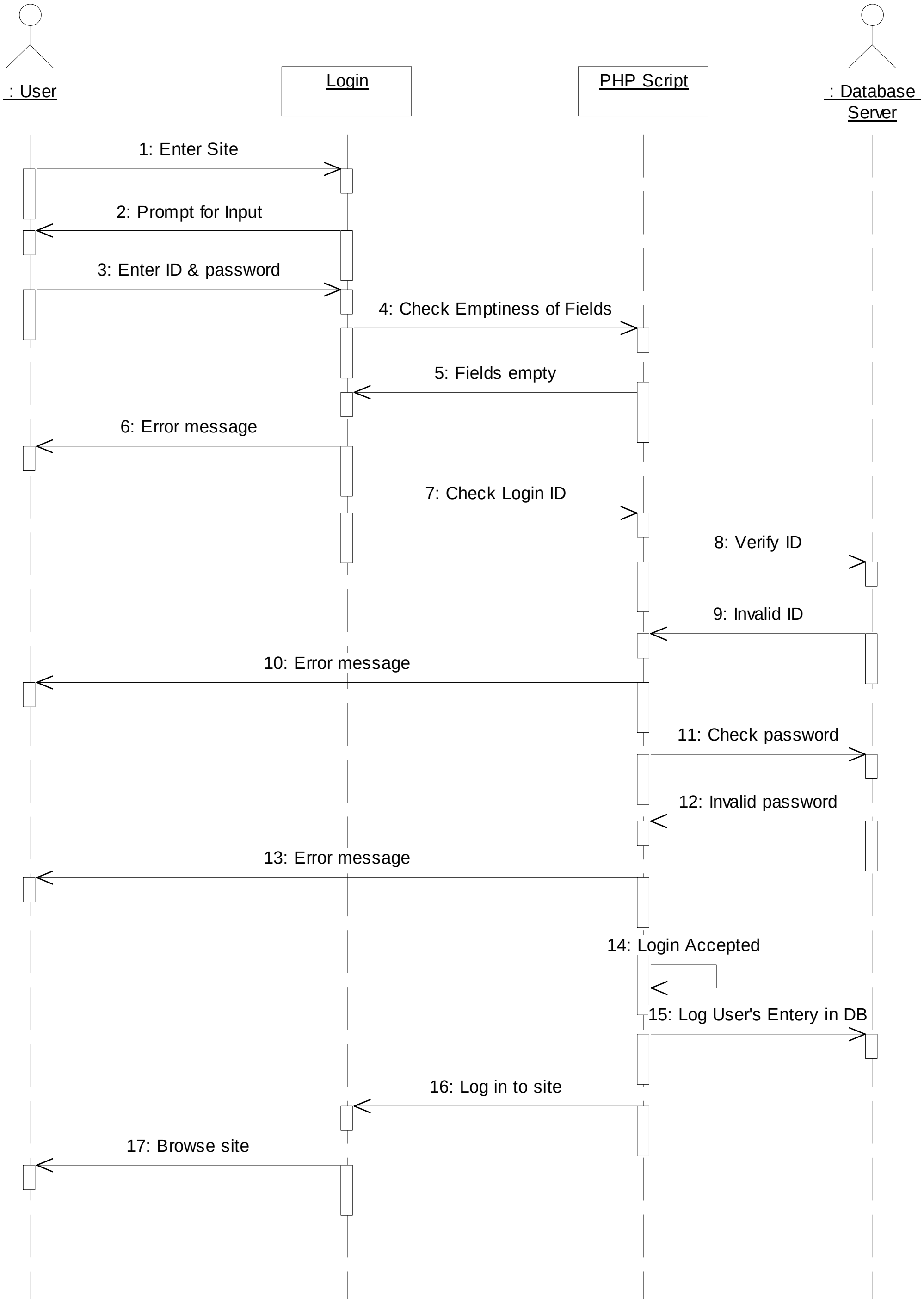


Figure 5-4 User's Login Sequence Diagram

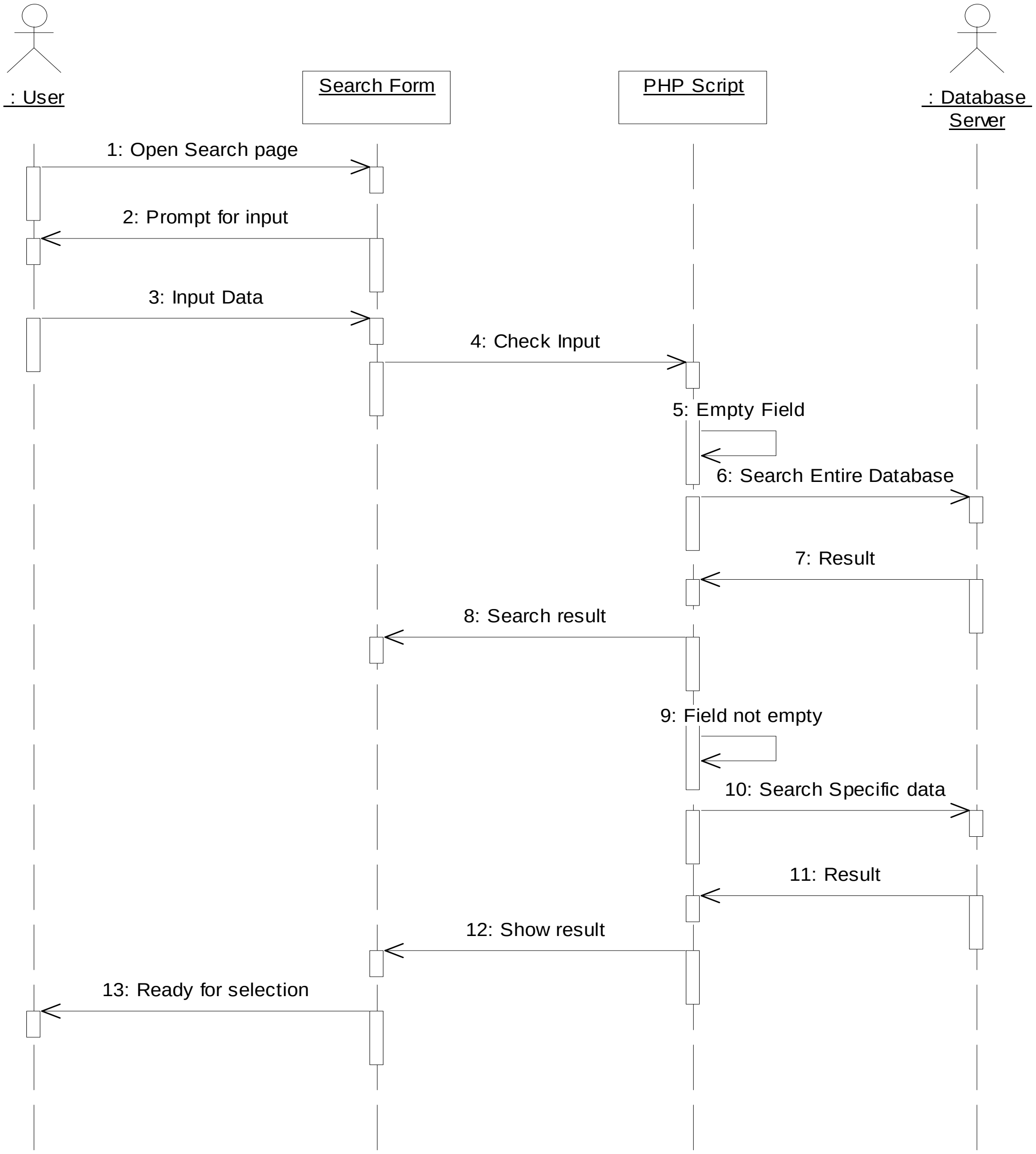


Figure 5-5 User’s Search for Item Sequence Diagram

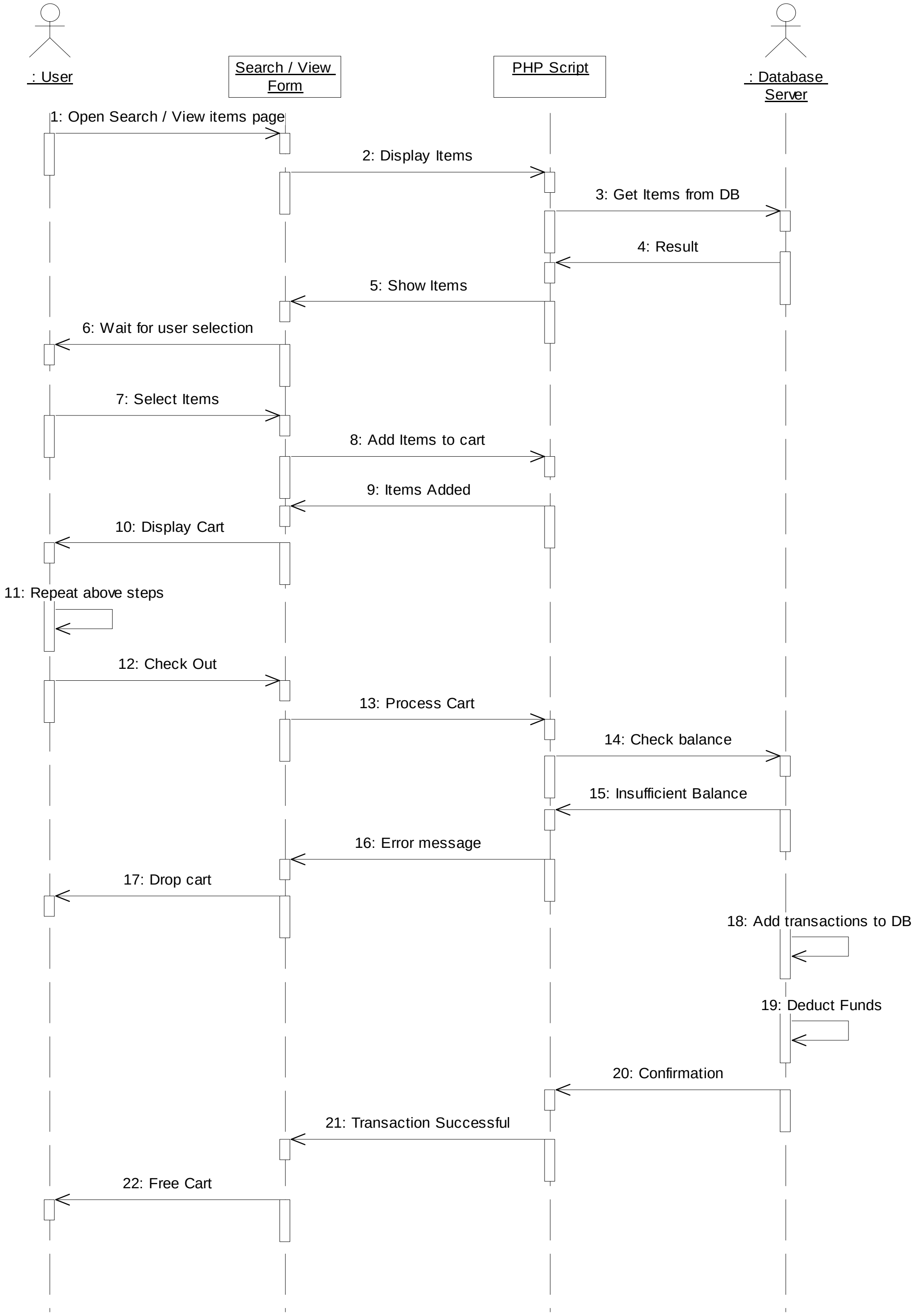


Figure 5-6 User's Transaction Sequence Diagram

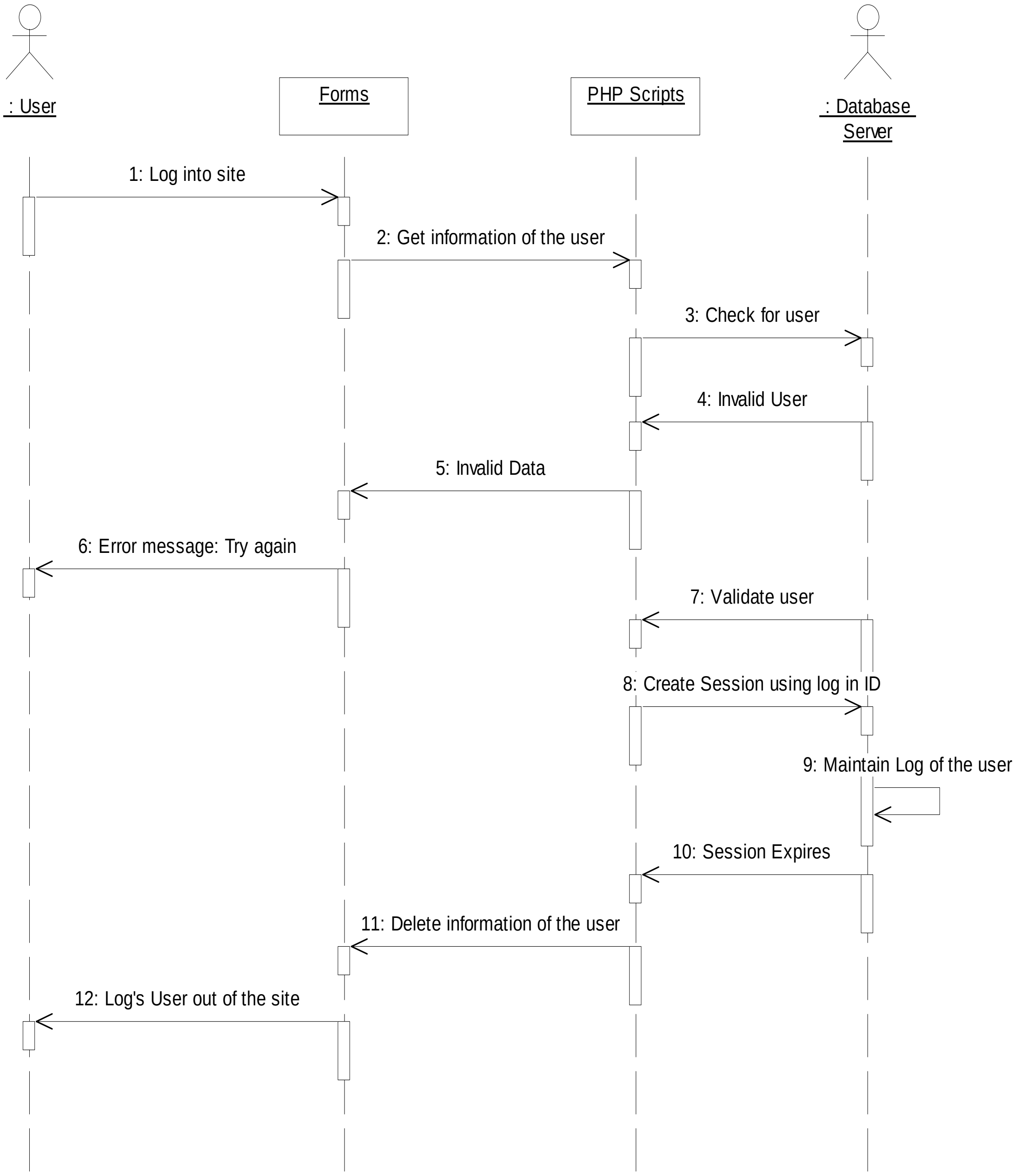

: User
Forms
PHP Scripts
: Database Server
1: Log into site
2: Get information of the user
3: Check for user
4: Invalid User
5: Invalid Data
6: Error message: Try again
7: Validate user
8: Create Session using log in ID
9: Maintain Log of the user
10: Session Expires
11: Delete information of the user
12: Log's User out of the site

Figure 5-7 Session Management Sequence Diagram

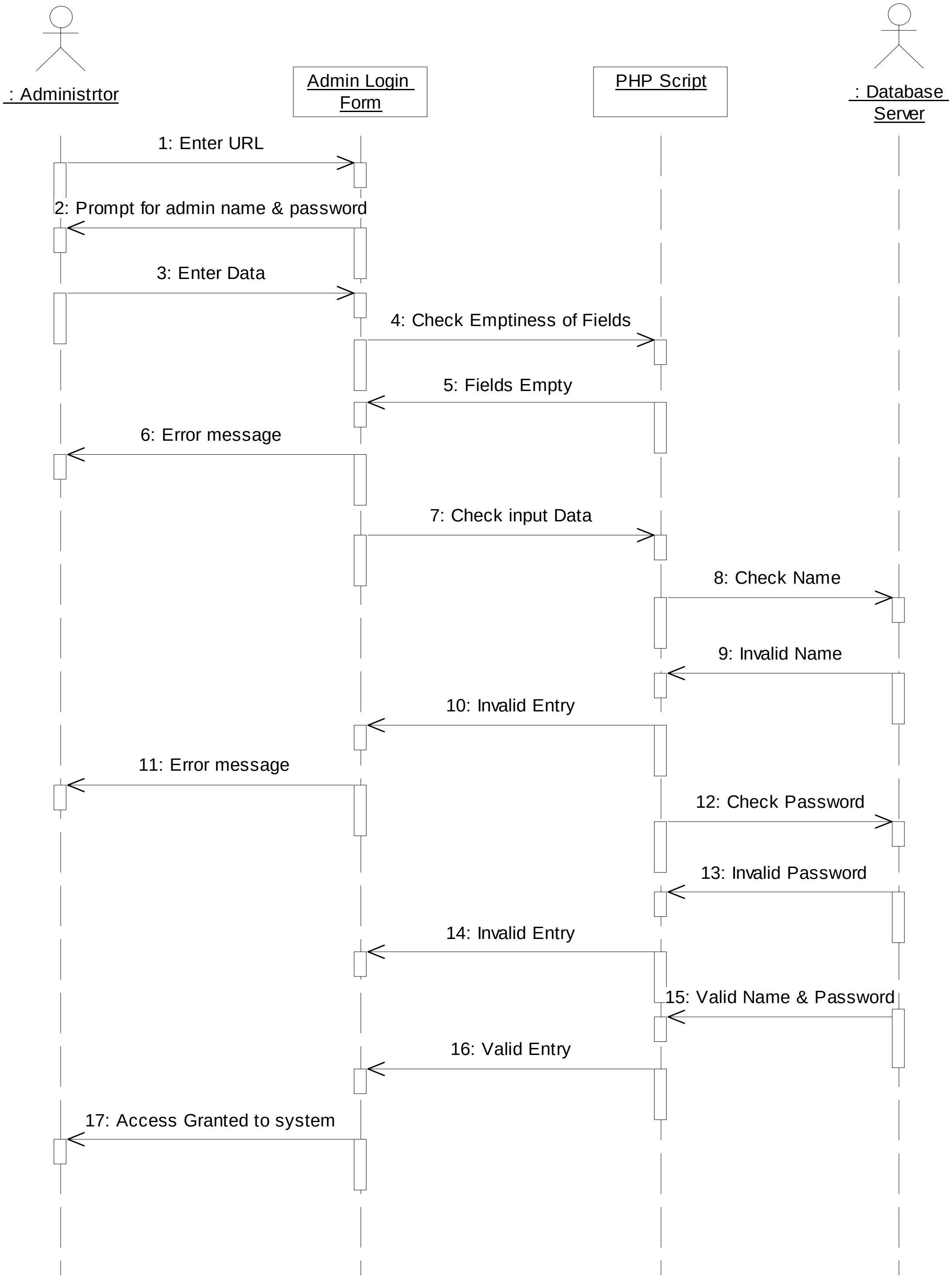


Figure 5-8 Administrators Login Sequence Diagram

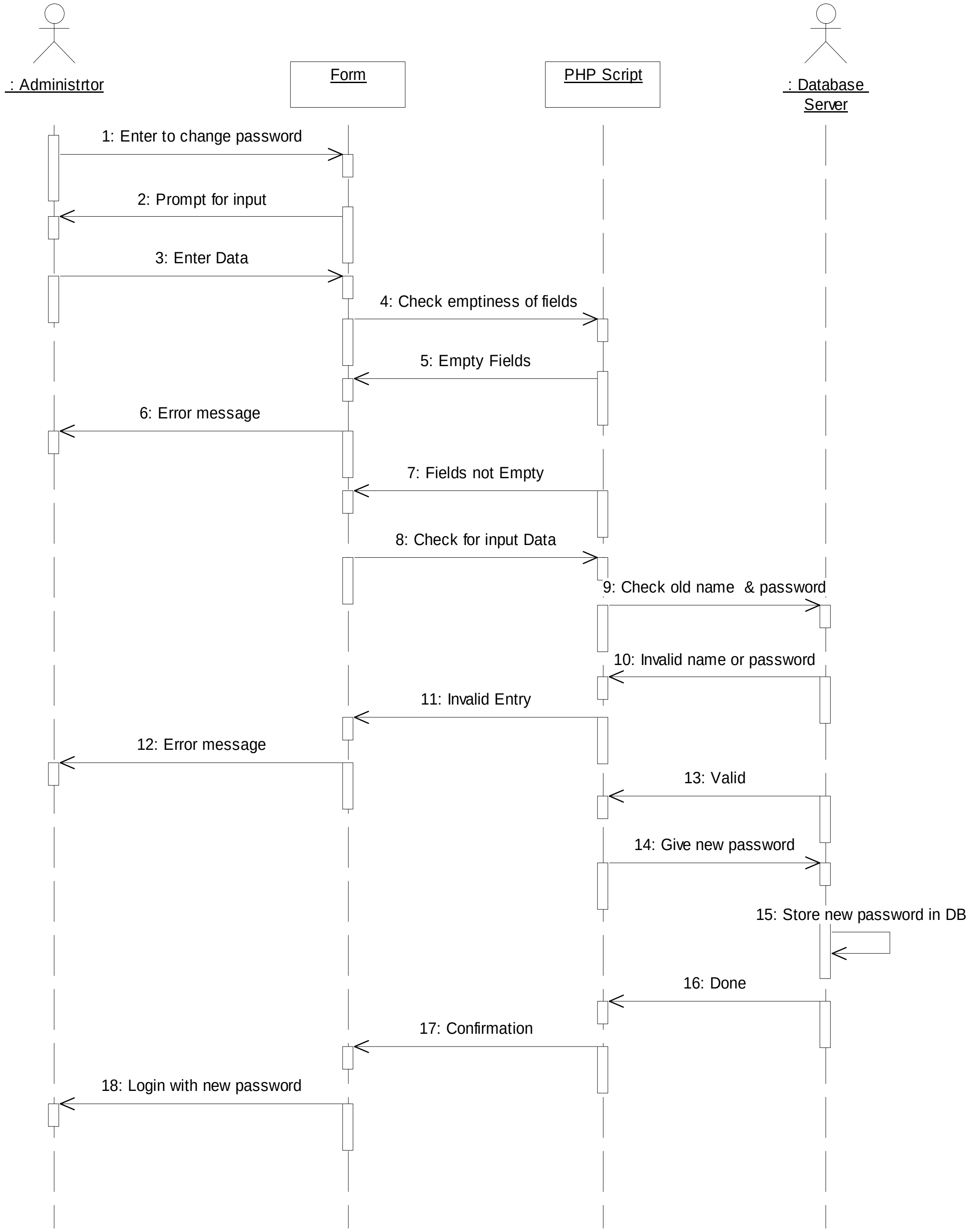


Figure 5-9 Administrators Changing Password Sequence diagram

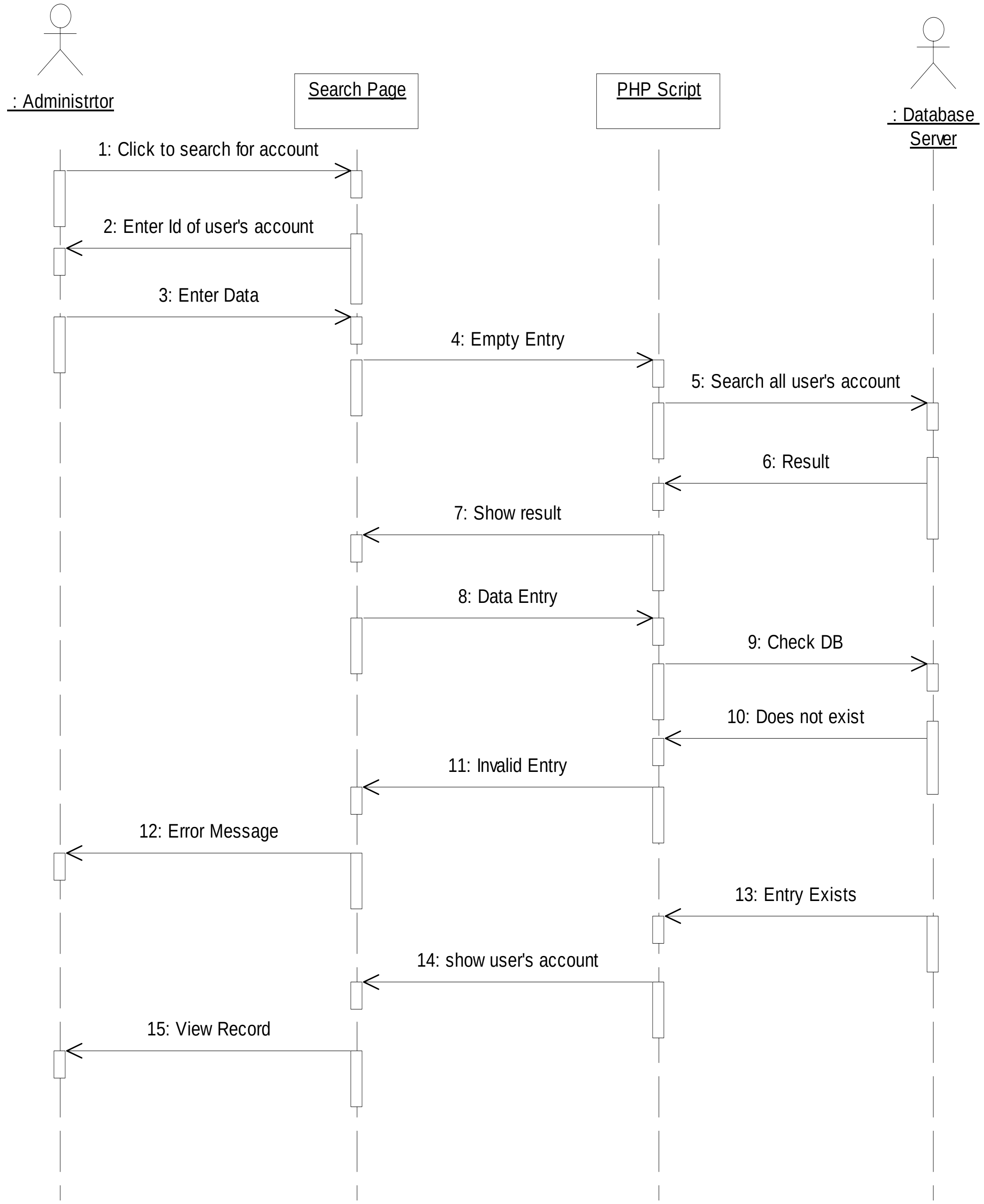


Figure 5-10 Administrators Search User’s Account Sequence diagram

5.4 Collaboration Diagrams

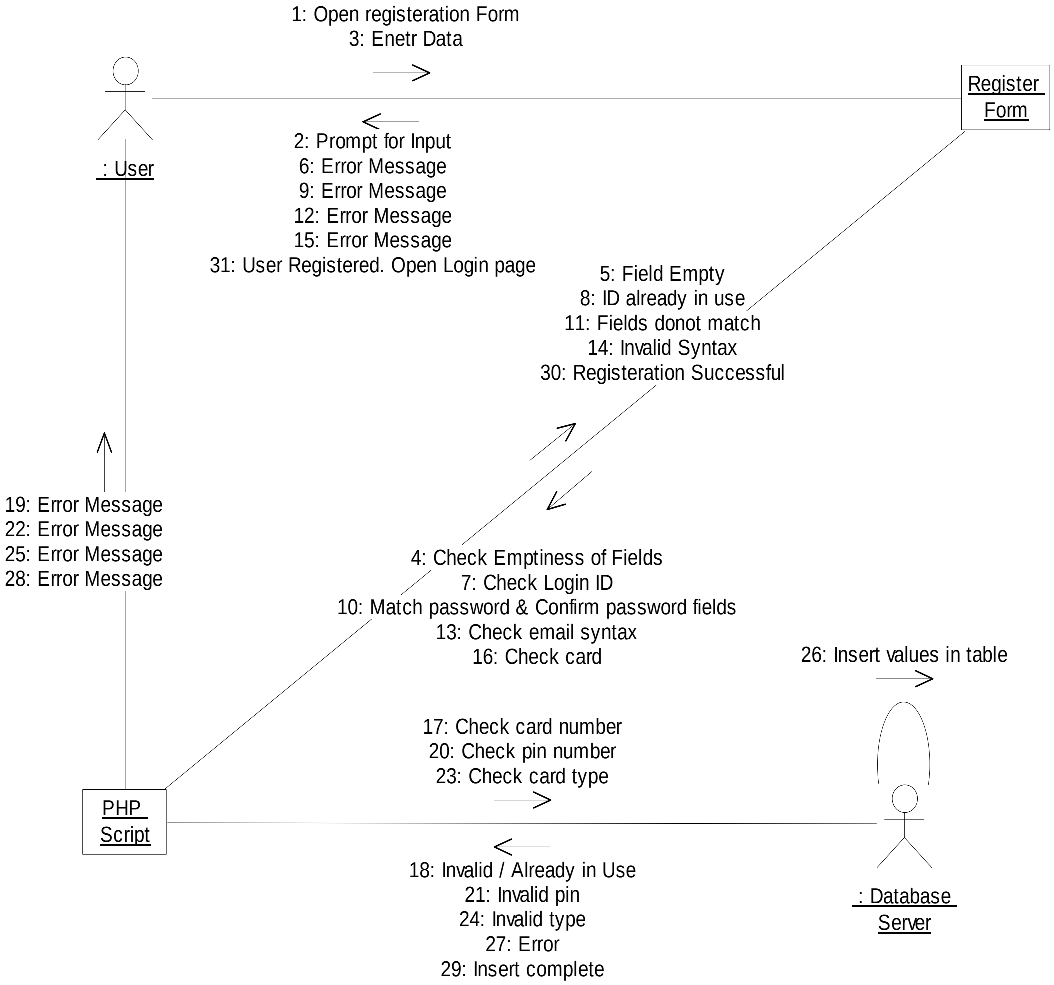


Figure 5-11 User’s Registration Collaboration Diagram

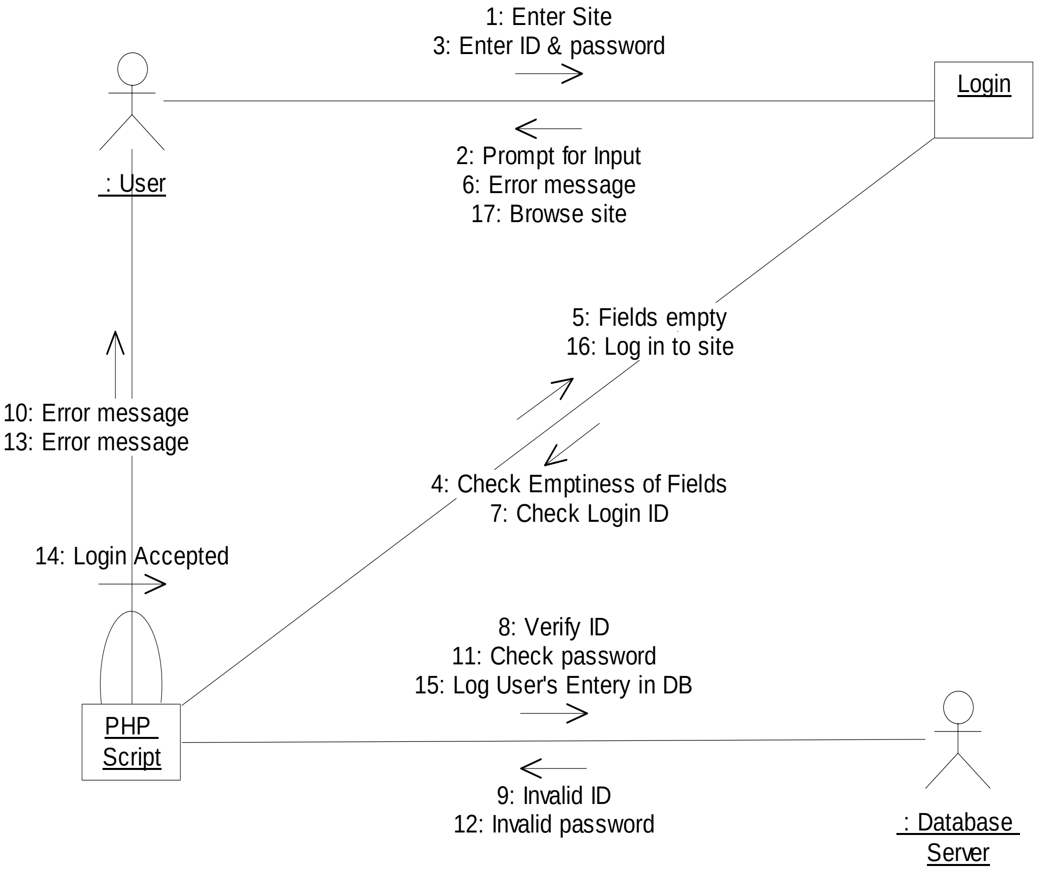


Figure 5-12 User’s Login Collaboration Diagram

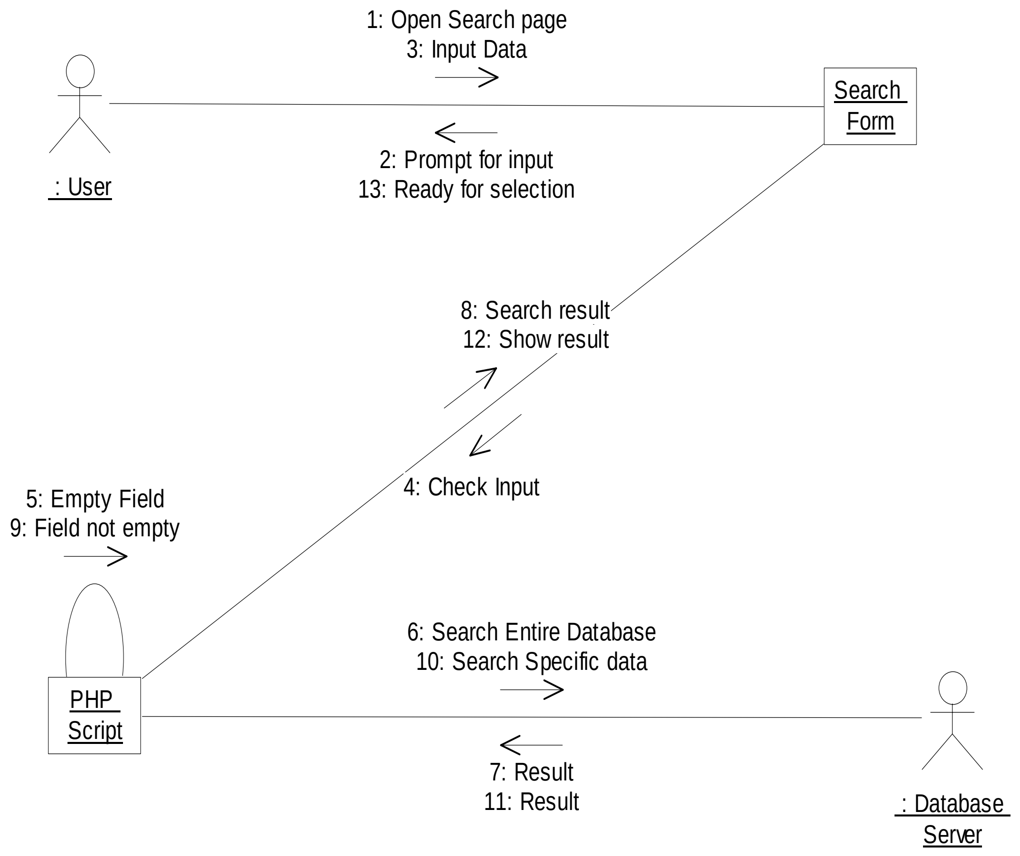


Figure 5-13 User's Search for an item Collaboration Diagram

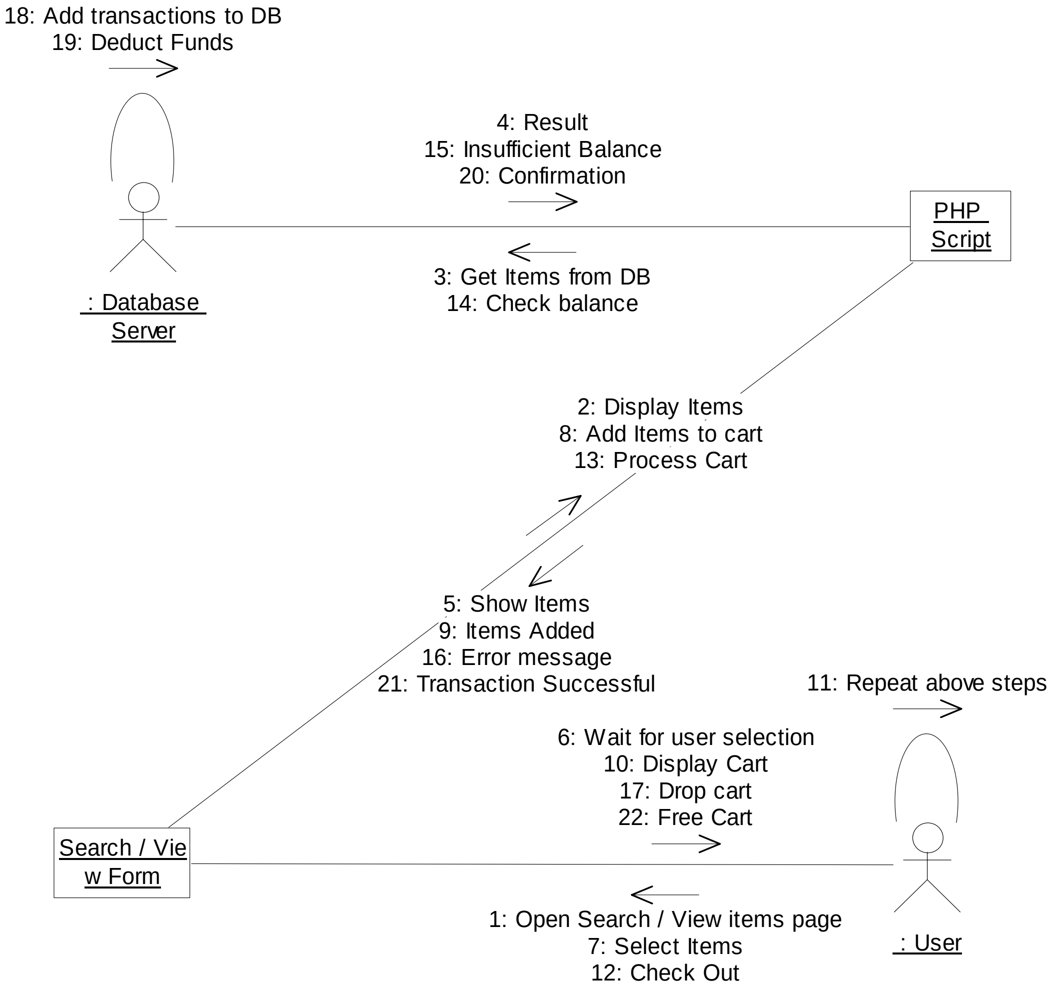


Figure 5-14 User's transaction Collaboration Diagram

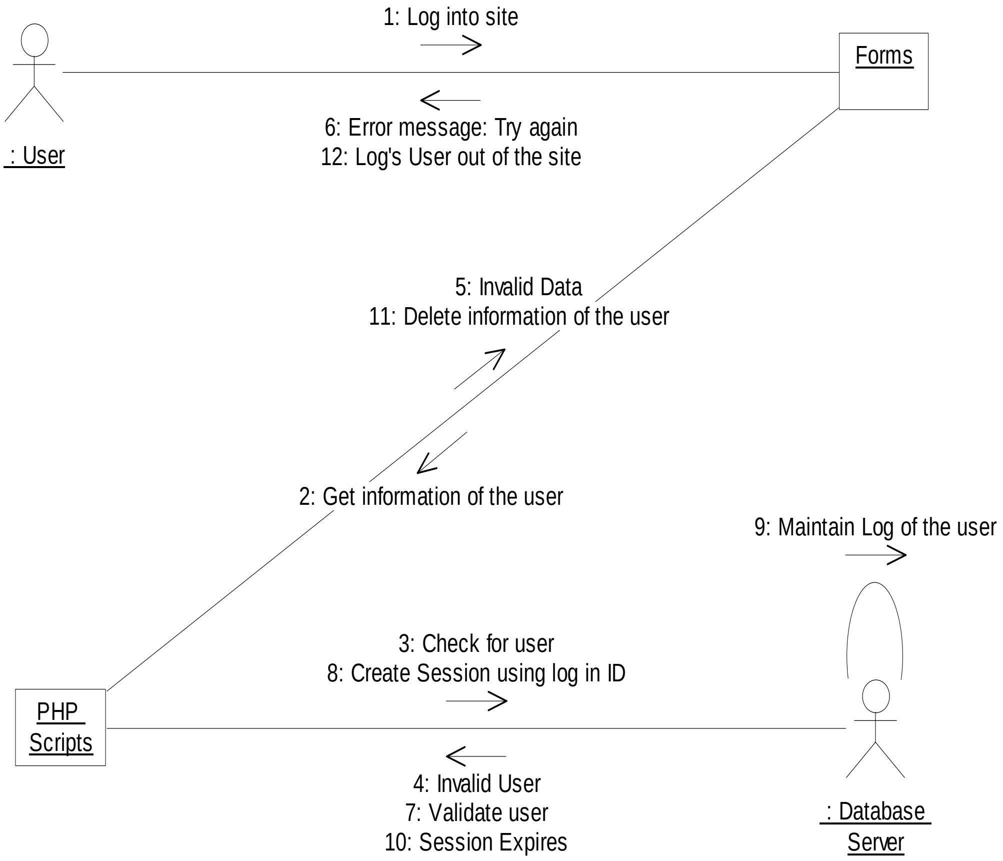

1: Log into site
Forms
6: Error message: Try again
12: Log's User out of the site
: User
5: Invalid Data
11: Delete information of the user
2: Get information of the user
9: Maintain Log of the user
3: Check for user
8: Create Session using log in ID
PHP Scripts
4: Invalid User
7: Validate user
10: Session Expires
: Database Server

Figure 5-15 Session Management Collaboration Diagram

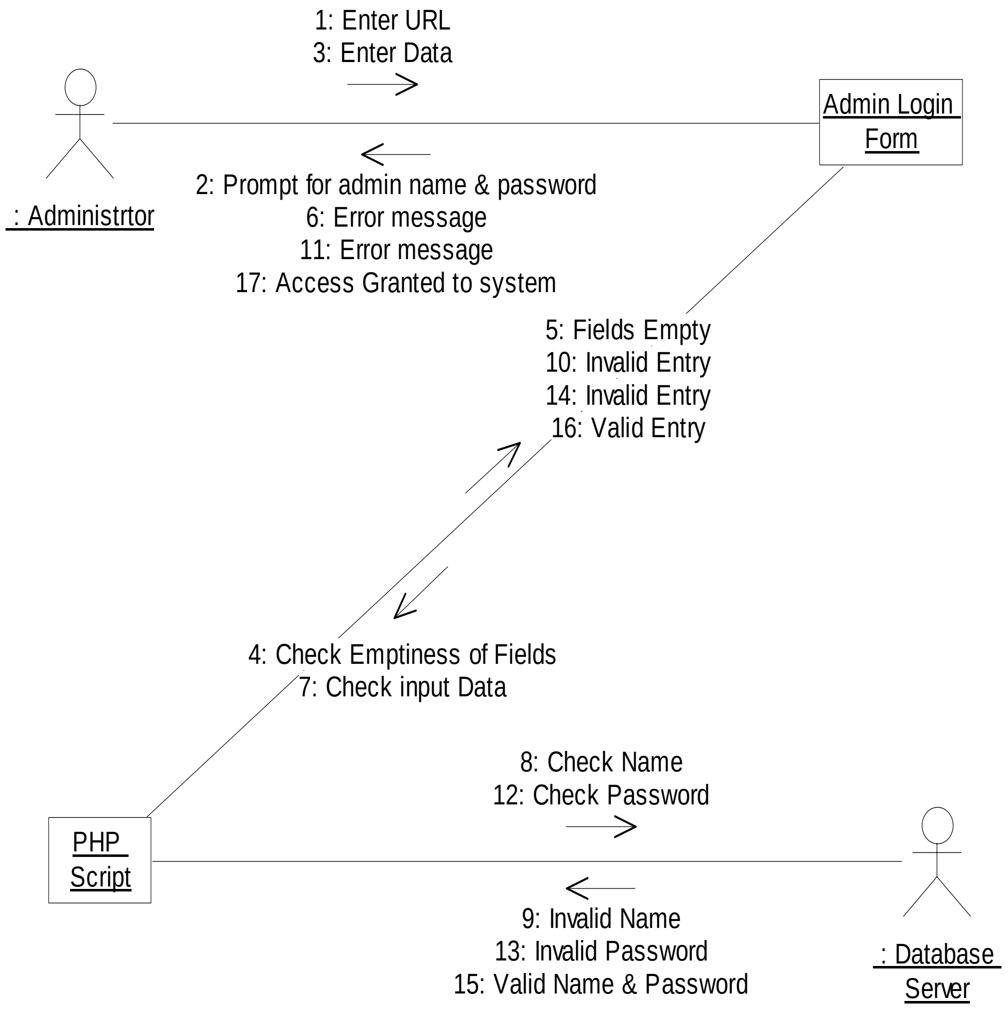


Figure 5-16 Administrators Login Collaboration Diagram

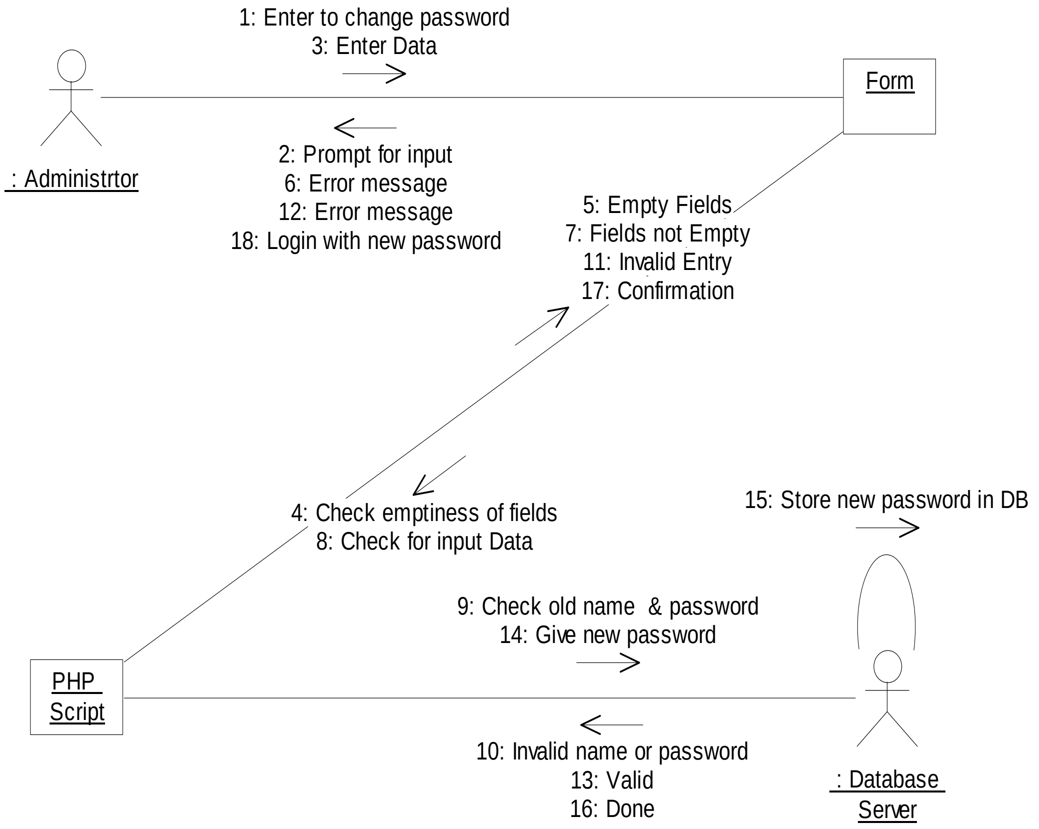


Figure 5-17 Administrator's Changing Password Collaboration Diagram

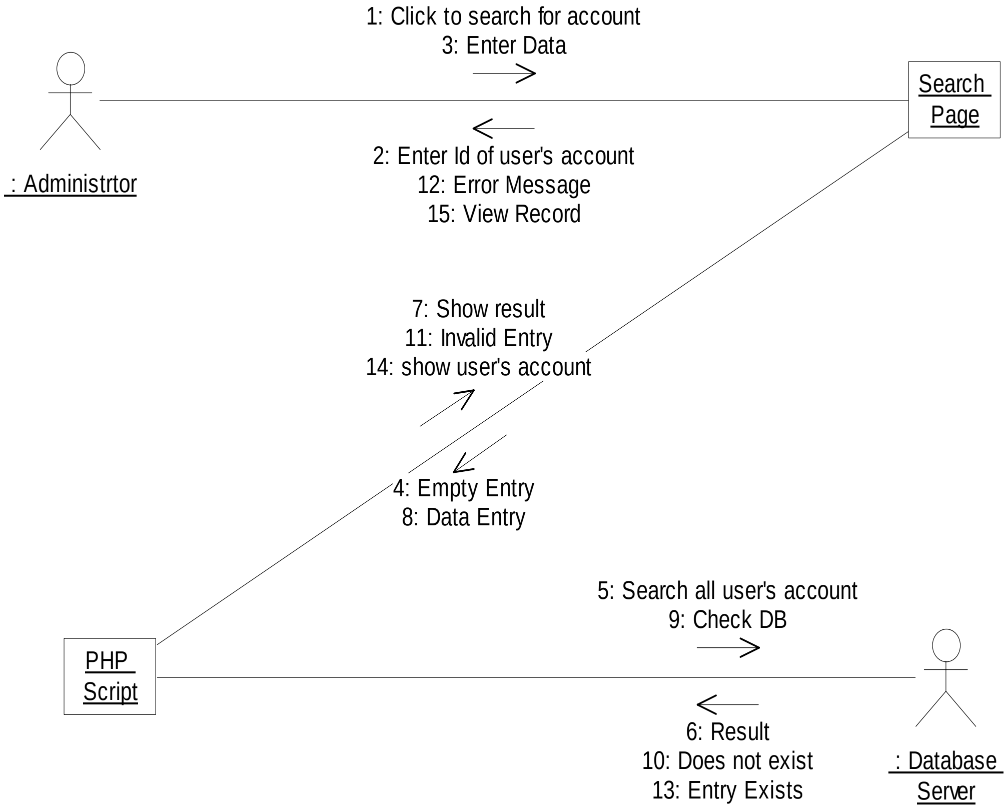


Figure 5-18 Administrator's Search for User Account Collaboration Diagram

5.5 Database Description

MySQL has been used for the designing and creation of the data base. The name of database is epscity_info.

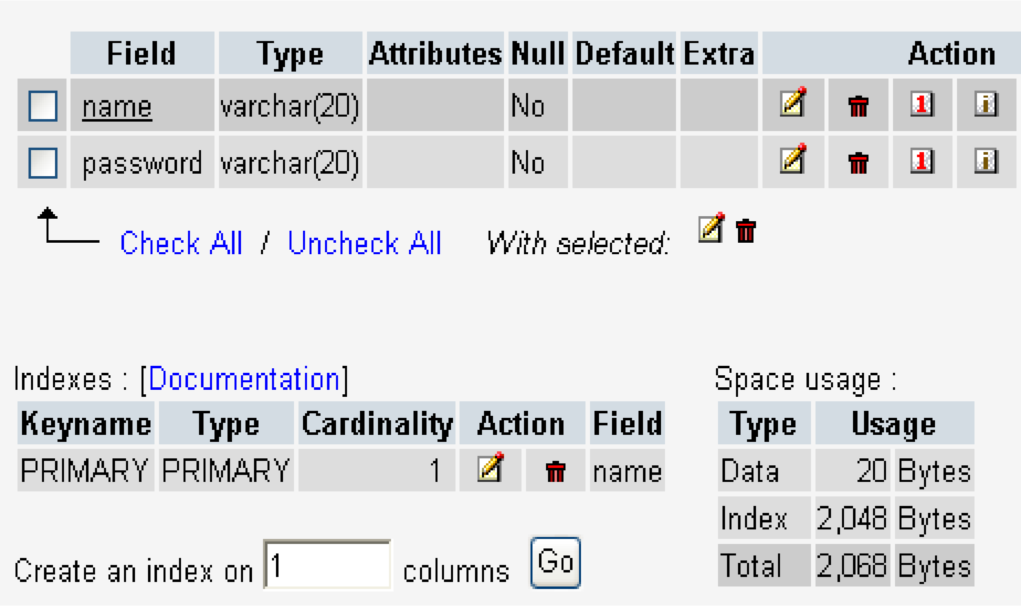
The detail of different tables used in data base is as given below.

(1) admin

This table contains the name and password of all administrators of the system.

The Fields of the table are as under.

- name
- password



	Field	Type	Attributes	Null	Default	Extra	Action			
□	name	varchar(20)		No						
□	password	varchar(20)		No						

Indexes : [Documentation]

Keyname	Type	Cardinality	Action		Field
PRIMARY	PRIMARY	1			name

Create an index on 1 columns Go

Space usage :

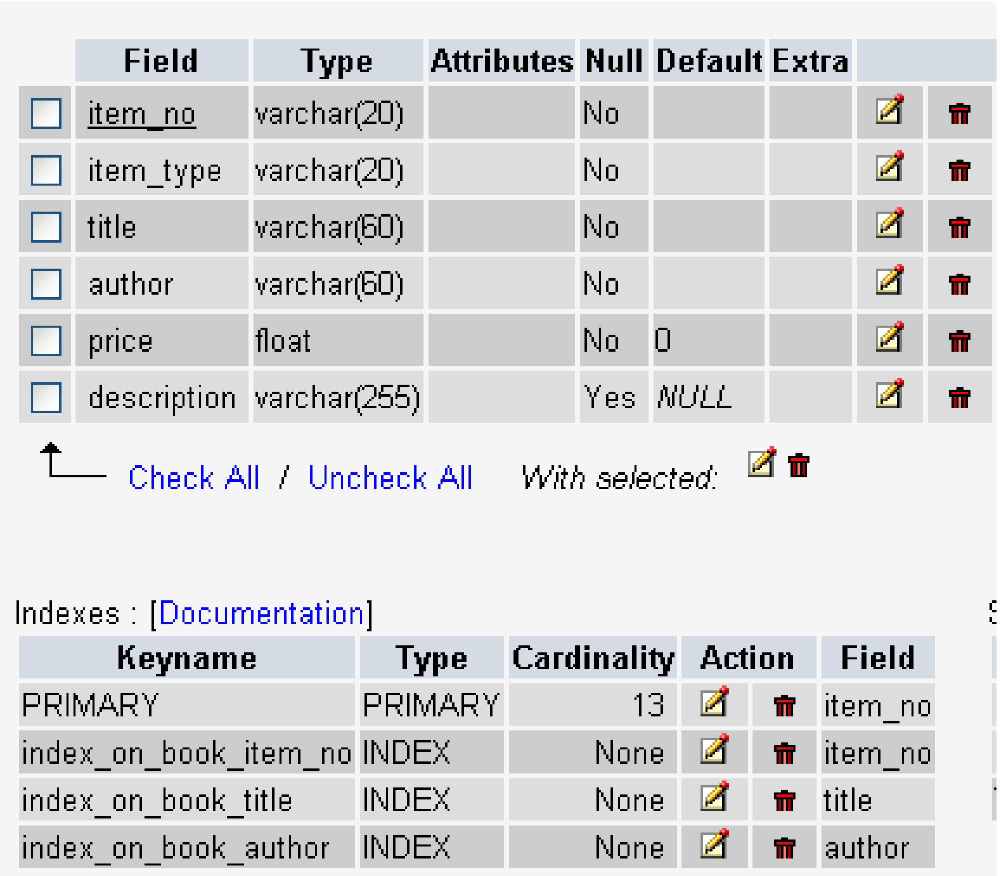
Type	Usage	
Data	20	Bytes
Index	2,048	Bytes
Total	2,068	Bytes

Figure 5-19 admin table

(2) book_shop

This table contains the books that are on display on the system. The fields are as under:

- item_no
- item_type
- title
- author
- price
- description



	Field	Type	Attributes	Null	Default	Extra		
□	item_no	varchar(20)		No			✎	🗑
□	item_type	varchar(20)		No			✎	🗑
□	title	varchar(60)		No			✎	🗑
□	author	varchar(60)		No			✎	🗑
□	price	float		No	0		✎	🗑
□	description	varchar(255)		Yes	*NULL*		✎	🗑

Indexes : [Documentation]

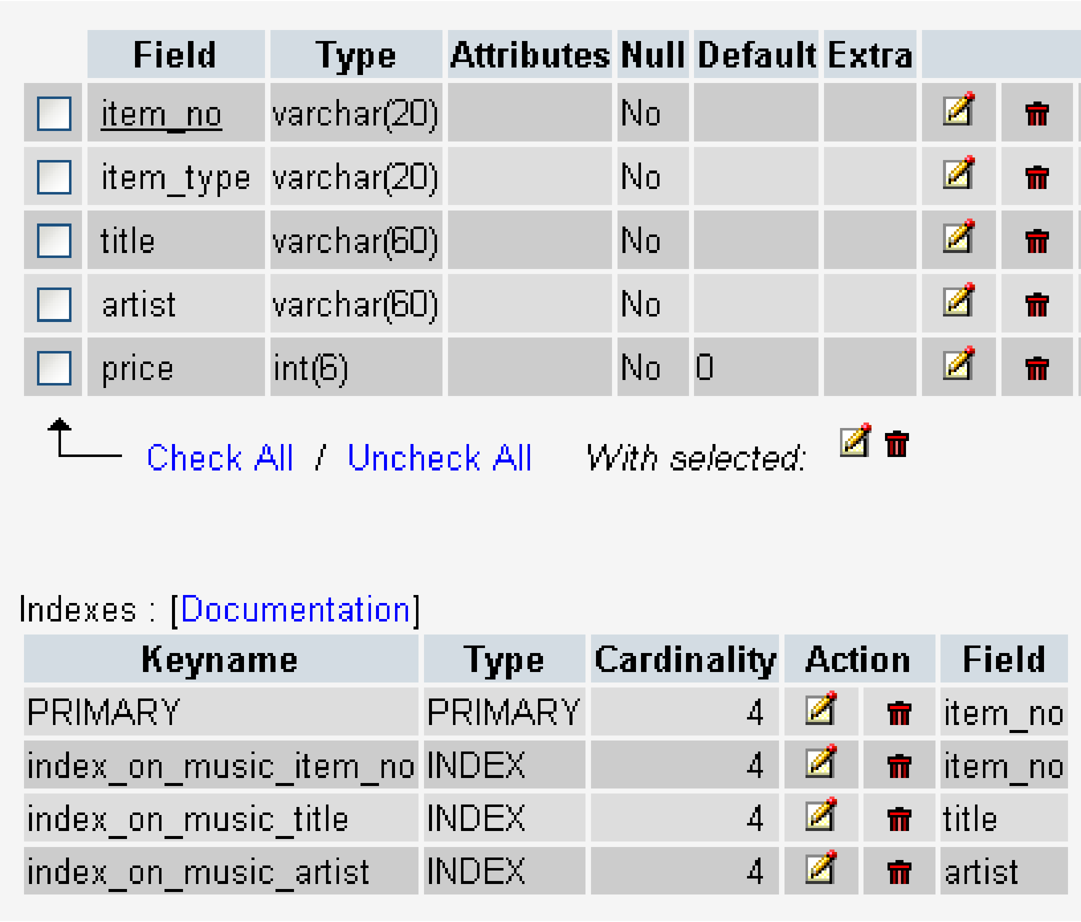
Keyname	Type	Cardinality	Action		Field
PRIMARY	PRIMARY	13	✎	🗑	item_no
index_on_book_item_no	INDEX	None	✎	🗑	item_no
index_on_book_title	INDEX	None	✎	🗑	title
index_on_book_author	INDEX	None	✎	🗑	author

Figure 5-20 book_shop table

(3) music_shop

This table contains the music albums/Dramas/Videos etc that are on display on the system. The fields are as under:

- item_no
- item_type
- title
- artist
- price
- description



	Field	Type	Attributes	Null	Default	Extra		
□	item_no	varchar(20)		No			✎	🗑
□	item_type	varchar(20)		No			✎	🗑
□	title	varchar(60)		No			✎	🗑
□	artist	varchar(60)		No			✎	🗑
□	price	int(6)		No	0		✎	🗑

Check All / Uncheck All *With selected:* ✎ 🗑

Indexes : [Documentation]

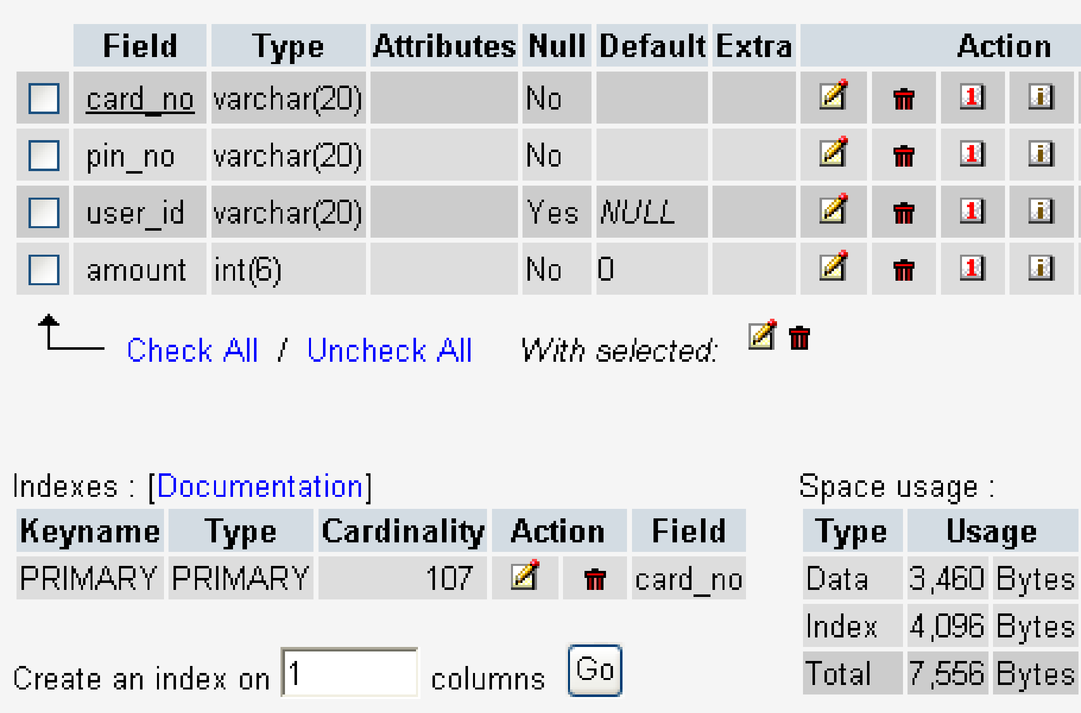
Keyname	Type	Cardinality	Action		Field
PRIMARY	PRIMARY	4	✎	🗑	item_no
index_on_music_item_no	INDEX	4	✎	🗑	item_no
index_on_music_title	INDEX	4	✎	🗑	title
index_on_music_artist	INDEX	4	✎	🗑	artist

Figure 5-21 music_shop table

(4) card

It contains the serial numbers and passwords of all the cards of the system. The fields are as under:

- card_no
- pin_no
- user_id
- amount



	Field	Type	Attributes	Null	Default	Extra	Action			
□	card_no	varchar(20)		No			✎	🗑	①	ⓘ
□	pin_no	varchar(20)		No			✎	🗑	①	ⓘ
□	user_id	varchar(20)		Yes	*NULL*		✎	🗑	①	ⓘ
□	amount	int(6)		No	0		✎	🗑	①	ⓘ

Indexes : [Documentation]

Keyname	Type	Cardinality	Action		Field
PRIMARY	PRIMARY	107	✎	🗑	card_no

Create an index on 1 columns Go

Space usage :

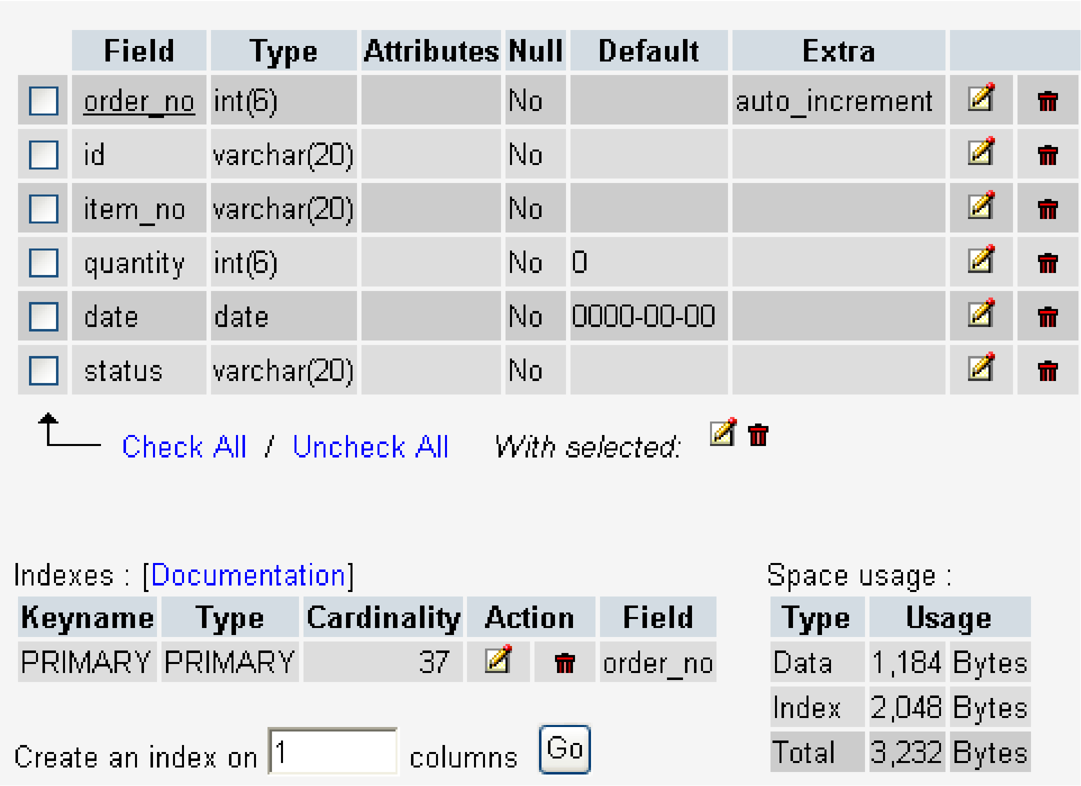
Type	Usage	
Data	3,460	Bytes
Index	4,096	Bytes
Total	7,556	Bytes

Figure 5-22 card table

(5) transaction

It contains the record of all the transactions made so far. The fields are as under:

- order_no
- id
- item_no
- quantity
- date
- status



	Field	Type	Attributes	Null	Default	Extra		
□	order_no	int(6)		No		auto_increment	✎	🗑
□	id	varchar(20)		No			✎	🗑
□	item_no	varchar(20)		No			✎	🗑
□	quantity	int(6)		No	0		✎	🗑
□	date	date		No	0000-00-00		✎	🗑
□	status	varchar(20)		No			✎	🗑

Indexes : [Documentation]

Keyname	Type	Cardinality	Action		Field
PRIMARY	PRIMARY	37	✎	🗑	order_no

Create an index on 1 columns Go

Space usage :

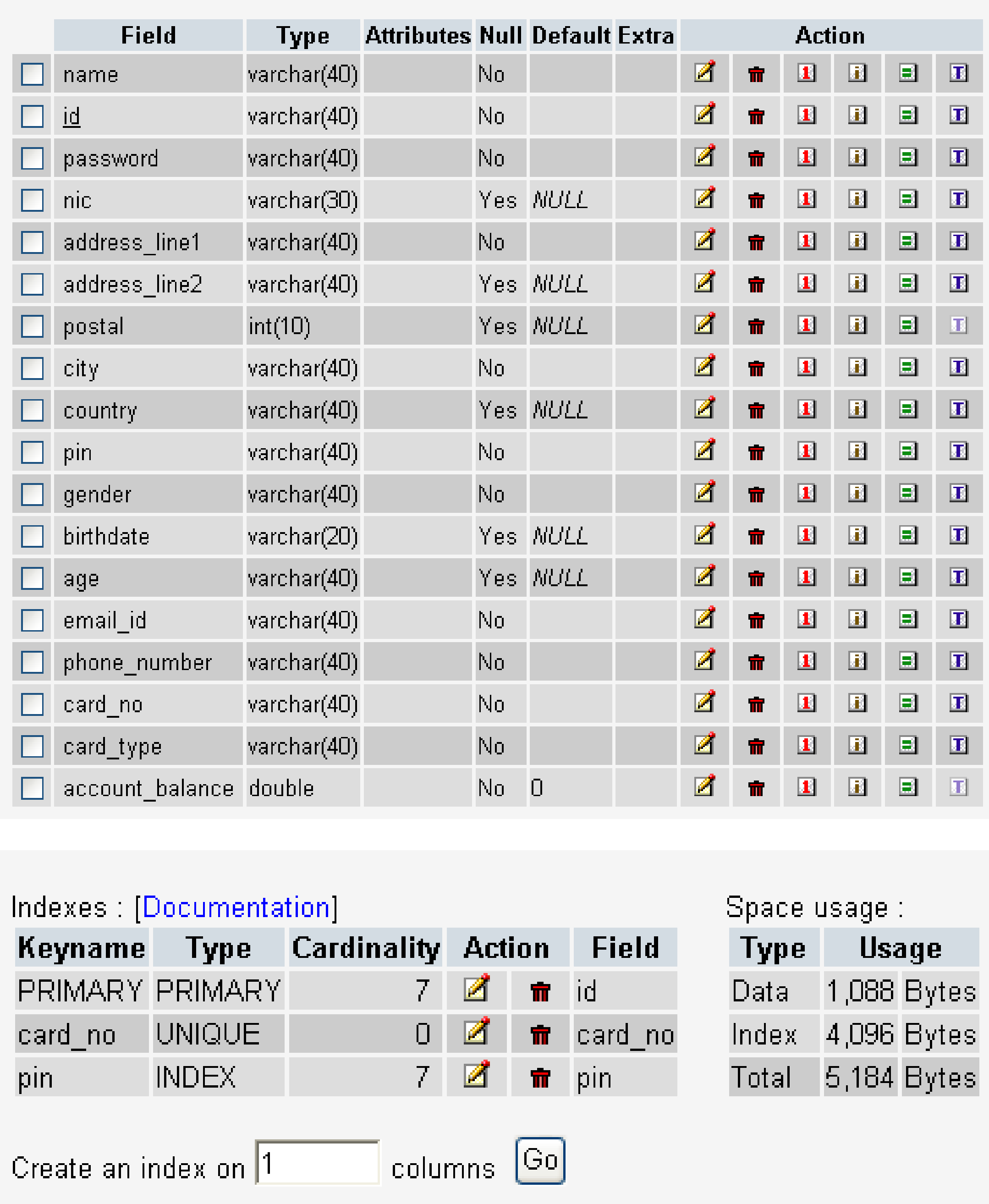
Type	Usage	
Data	1,184	Bytes
Index	2,048	Bytes
Total	3,232	Bytes

Figure 5-23 transaction table

(6) user_profile

It contains profile of all the registered users. The fields of the table are as under:

- name
- id
- password
- nic
- address_line1
- address_line2
- postal
- city
- country
- pin
- gender
- birthdate
- age
- email_id
- phone_no
- card_no
- card_type
- account_balance



	Field	Type	Attributes	Null	Default	Extra	Action					
□	name	varchar(40)		No			✎	🗑	1	i	=	T
□	id	varchar(40)		No			✎	🗑	1	i	=	T
□	password	varchar(40)		No			✎	🗑	1	i	=	T
□	nic	varchar(30)		Yes	*NULL*		✎	🗑	1	i	=	T
□	address_line1	varchar(40)		No			✎	🗑	1	i	=	T
□	address_line2	varchar(40)		Yes	*NULL*		✎	🗑	1	i	=	T
□	postal	int(10)		Yes	*NULL*		✎	🗑	1	i	=	T
□	city	varchar(40)		No			✎	🗑	1	i	=	T
□	country	varchar(40)		Yes	*NULL*		✎	🗑	1	i	=	T
□	pin	varchar(40)		No			✎	🗑	1	i	=	T
□	gender	varchar(40)		No			✎	🗑	1	i	=	T
□	birthdate	varchar(20)		Yes	*NULL*		✎	🗑	1	i	=	T
□	age	varchar(40)		Yes	*NULL*		✎	🗑	1	i	=	T
□	email_id	varchar(40)		No			✎	🗑	1	i	=	T
□	phone_number	varchar(40)		No			✎	🗑	1	i	=	T
□	card_no	varchar(40)		No			✎	🗑	1	i	=	T
□	card_type	varchar(40)		No			✎	🗑	1	i	=	T
□	account_balance	double		No	0		✎	🗑	1	i	=	T

Indexes : [Documentation]

Keyname	Type	Cardinality	Action		Field
PRIMARY	PRIMARY	7	✎	🗑	id
card_no	UNIQUE	0	✎	🗑	card_no
pin	INDEX	7	✎	🗑	pin

Space usage :

Type	Usage	
Data	1,088	Bytes
Index	4,096	Bytes
Total	5,184	Bytes

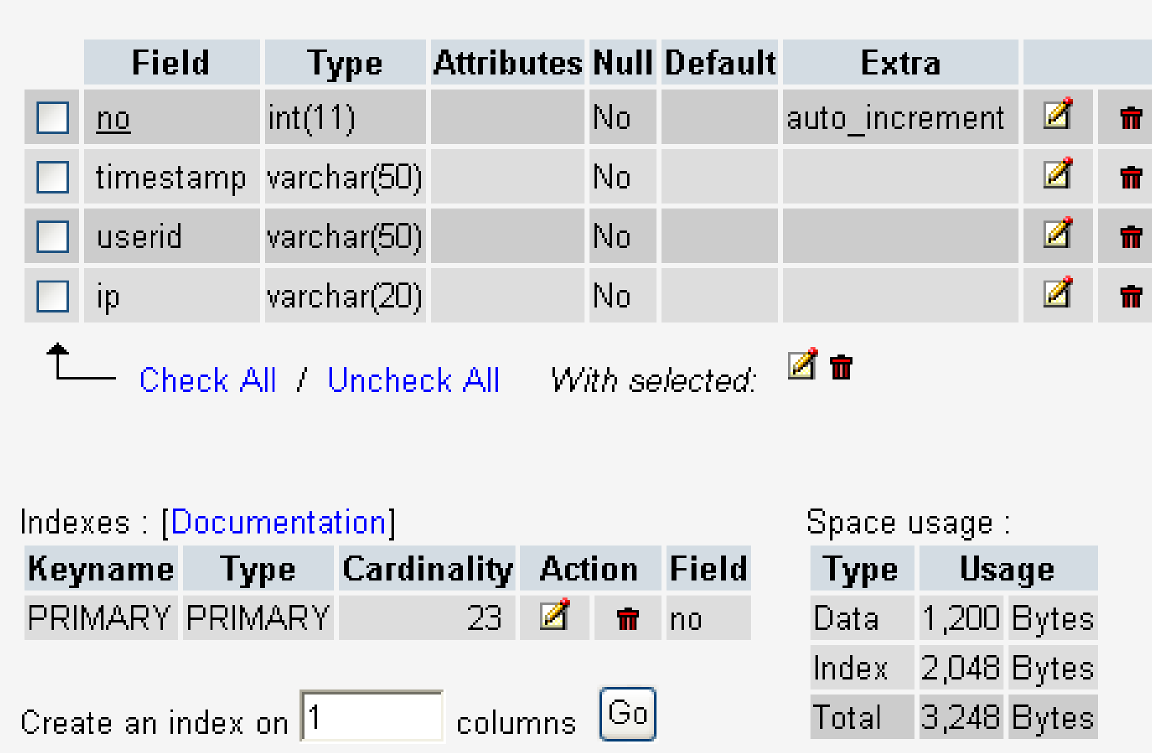
Create an index on 1 columns Go

Figure 5-24 user_profile table

(7) useronline

It contains the log of the user’s who use our system. The fields of the table are as under:

- no
- timestamp
- userid
- ip



	Field	Type	Attributes	Null	Default	Extra		
□	no	int(11)		No		auto_increment	✎	🗑
□	timestamp	varchar(50)		No			✎	🗑
□	userid	varchar(50)		No			✎	🗑
□	ip	varchar(20)		No			✎	🗑

Indexes : [Documentation]

Keyname	Type	Cardinality	Action		Field
PRIMARY	PRIMARY	23	✎	🗑	no

Create an index on 1 columns Go

Space usage :

Type	Usage	
Data	1,200	Bytes
Index	2,048	Bytes
Total	3,248	Bytes

Figure 5-25 useronline table

5.6 Description of files (HTML & PHP)

5.6.1 USER APPLICATION:

1. default.htm:

Main page of the system. The users use this page to login to the system. New users can use the link to go to the register form

2. register.htm:

Registration form for new users. The required fields are to be filled for the system to proceed. Otherwise an error message will be displayed.

3. error.htm:

Displays different error messages to the user.

4. links.htm

Provides the links to different forms of the system.

5. default_authenticated.htm

When the user logs into the site, this form is displayed which contains other forms (both html & php forms)

6. board.htm

This page is the main page of the board of governors of the company.

7. recharge.htm

This form is used to fill in the required data for recharging their account.

8. registration_success.htm

This form is displayed when the registration process is completed.

9. login.php

It is used to validate the user. PHP connects to the database and finds the relevant record of the user. If the user ID or password does not exist then the user is not granted permission to enter the system.

10. register.php

Used to register a new user, The values from register.htm are passed to register.php which connects to the database to check for validity of the new account. If some error occurs the system does not lets the user proceed.

11 account_status.php

Used to display the account status of the logged in user. Displays all the transactions that the user has carried out so far.

12. added_items.php

Used to maintain a shopping cart. Whenever a user selects an item, that item is placed in this cart.

13. book_listing.php

Used to display all the books that are stored in the books database.

14. changing_qty.php

Used to change the quantity of a selected item. By default the quantity is 1.

15. confirm_order.php

Used to do the transactions of the users. Whenever the users check’s out his order is confirmed and the transaction is carried out.

16. display.php

Displays all the items that are placed in the shopping cart by the user.

17. functions.php

Contains all the functions of the system. Every file calls this file.

18. music_listing.php

Used to display all the items that are stored in the music database.

19. logout.php

Used to log a user out of the system. When a user logs out of the system he/she cannot carry out any kind of transactions.

21. register.php

Used to register a new user. Checks for the validity of the entered data and stores information in the database. If there is an error then the user is not registered and hence not allowed to proceed.

22. search_result.php

Displays the result of the item that the user searches.

23. view_account.php

Displays the account information of the logged in user from the database.

24. recharge.php

Used to recharge an account of a logged in user.

5.6.2 ADMINISTRATOR APPLICATION:

1. admin.htm

Login page of the administrator.

2. admin_error.htm

Displayed when there is an error in the login of the administrator.

3. admin_main.htm

Displayed when the administrator is authenticated.

4. add_card.htm

The administrator enters data to add new cards in the database on this page.

5. admin.php

Used to login the administrator. Checks the ID and password of the administrator from the database.

6. delete_all_selected_users.php

Used to delete all the selected users.

7. logout_admin.php

Used to log an administrator out of the system.

8. search_user.php

Used by the administrator to search for a particular user by entering their login ID.

9. ship_order.php

Used by the administrator to ship the items in the transaction database that are pending to the user.

10. transaction_admin.php

The administrator uses this form to display the transactions carried out by all the users on the current date of the server.

11. card.php

Used by the administrator to display all the cards in the database.

12. passgen.php

Generates random numbers for card numbers and passwords and stores the entry in the card database.

13. ship_order.php

Used by the administrator to note down the transactions carried out by the user. The system changes the status of the transaction from pending to shipped in the database.

Note:

All the PHP forms carry out the following operations:

(1) Authenticates the user/administrator using the function authenticateuser() / authenticateadmin(). The values for their parameters are taken from the HTML forms by the POST method.

(2) Connects to the database using the information stored in the common.inc file.

(3) Runs a query on a specific table.

(4) Creates a log of the user's visit.

(5) Transfer various information to other forms.

CHAPTER 6

SECURITY

In the physical world, face-to-face transactions, photo identification and even written signatures offer some protection against fraud. However, the Internet remains relatively anonymous, making it harder to know who is at the other end of the network.

The challenge for the Internet economy is to translate the trust conventions of the physical marketplace and make them work online. Public key infrastructure has become the de facto standard for establishing this trust and executing binding contracts over electronic networks.

PKIs integrate digital certificates, public-key cryptography, and certificate authorities into a total, enterprise-wide network security architecture. A typical enterprise's PKI encompasses the issuance of digital certificates to individual users and servers; end-user enrollment software; integration with corporate certificate directories; tools for managing, renewing, and revoking certificates; and related services and support.

6.1 The Need for Secure Communication

With the astonishing growth of the Internet, businesses are beginning to find new ways to expand their opportunities. With e-commerce on the rise, everyone major business has a web site which provides a variety of services. As more and more credit cards, account numbers, and personal information begins streaming through the Web, it becomes ever more important to adopt some form of data protection.

The most common form of data protection is encryption. For true security on the Web, you need to use some form of encryption and authentication between the browser and the web server to prevent the information between the two from being seen or changed by an unwanted third party. An encrypted transmission is a transmission that contains plain text data which has been mathematically altered so as to be unreadable, but which can be transformed back to the original by a reverse mathematical algorithm.

6.2 Internet Security Issues

All communication over the Internet uses the Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP allows information to be sent from one computer to another through a variety of intermediate computers and separate networks before it reaches its destination.

The great flexibility of TCP/IP has led to its worldwide acceptance as the basic Internet and intranet communications protocol. At the same time, the fact that TCP/IP allows information to pass through intermediate computers makes it possible for a third party to interfere with communications in the following ways:

5.2.1. Eavesdropping. Information remains intact, but its privacy is compromised. For example, someone could learn your credit card number, record a sensitive conversation, or intercept classified information.

5.2.2. Tampering. Information in transit is changed or replaced and then sent on to the recipient. For example, someone could alter an order for goods or change a person's resume.

5.2.3. Impersonation. Information passes to a person who poses as the intended recipient. Impersonation can take two forms:

5.2.4. Spoofing. A person can pretend to be someone else. For example, a person can pretend to have the email address `accounts@citibank.com`, or a computer can identify itself as a site called `www.amazon.com` when it is not. This type of impersonation is known as spoofing.

5.2.5. Misrepresentation. A person or organization can misrepresent itself. For example, suppose the site `www.buyfromme.com` pretends to be a e-commerce store when it is really just a site that takes credit-card payments but never sends any goods.

Normally, users of the many cooperating computers that make up the Internet or other networks don't monitor or interfere with the network traffic that continuously passes through their machines. However, many sensitive personal and business communications over the Internet require precautions that address the threats listed above. Fortunately, a set of well-established techniques and standards known as **public-key cryptography** make it relatively easy to take such precautions.

Public-key cryptography facilitates the following tasks:-

- **Encryption and decryption** allow two communicating parties to disguise information they send to each other. The sender encrypts, or scrambles, information before sending it. The receiver decrypts, or unscrambles, the information after receiving it. While in transit, the encrypted information is unintelligible to an intruder.
- **Tamper detection** allows the recipient of information to verify that it has not been modified in transit. Any attempt to modify data or substitute a false message for a legitimate one will be detected.

- **Authentication** allows the recipient of information to determine its origin that is, to confirm the sender's identity.
- **Non-repudiation** prevents the sender of information from claiming at a later date that the information was never sent.

These issues are addressed through the use of cryptography, or more precisely, public key cryptography, which is discussed in the subsequent chapters.

6.3 The Basics of Cryptography

6.3.1 Encryption and Decryption

Encryption is the process of transforming information so it is unintelligible to anyone but the intended recipient. Decryption is the process of transforming encrypted information so that it is intelligible again. A cryptographic algorithm, also called a cipher, is a mathematical function used for encryption or decryption. In most cases, two related functions are employed, one for encryption and the other for decryption.

Data that can be read and understood without any special measures is called *plaintext* or *cleartex*t. Encrypting plaintext results in unreadable gibberish called *ciphertex*t. You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data.

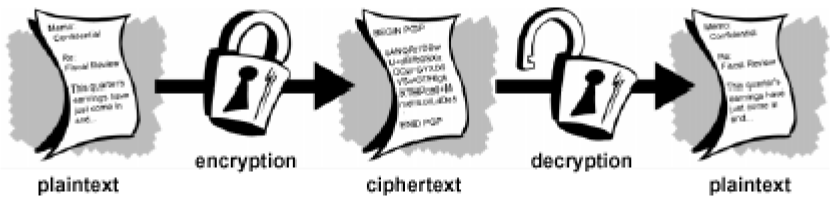


Figure 6-1   Encryption and decryption

6.3.2 How Does Cryptography Work?

A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a *ke*y—a word, number, or phrase—to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key. A cryptographic algorithm, plus all possible keys and all the protocols that make it work comprise a *cryptosyste*m.

6.3.3 Keys

A key is a value that works with a cryptographic algorithm to produce a specific ciphertext. Keys are basically really big numbers. Key size is measured in bits; the number representing a 1024-bit key is huge. In public key cryptography, the bigger the key, the more secure the ciphertext.

Keys are stored in encrypted form. They are usually stored in a special type of file known as a *keystore*, which is a password protected file for the storage of private keys and digital certificates.

6.4 Types of Encryption

With most modern cryptography, the ability to keep encrypted information secret is based not on the cryptographic algorithm but on a number called a key that must be used with the algorithm to produce an encrypted result or to decrypt previously encrypted information. Decryption with the correct key is simple. Decryption without the correct key is very difficult, and in some cases impossible for all practical purposes.

There are basically two types of Encryption and Decryption techniques which are: -

6.4.1 Symmetric-Key Encryption

Symmetric Key Encryption or Shared Key Encryption (Secret key encryption) is the oldest form of encryption in existence. In symmetric key encryption, there is only one key. You encrypt the message, using this key. The person decrypting the message must have the same key i.e. the same key is used for both encryption and decryption. *Figure 1-2* is an illustration of the Symmetric Key Encryption process.

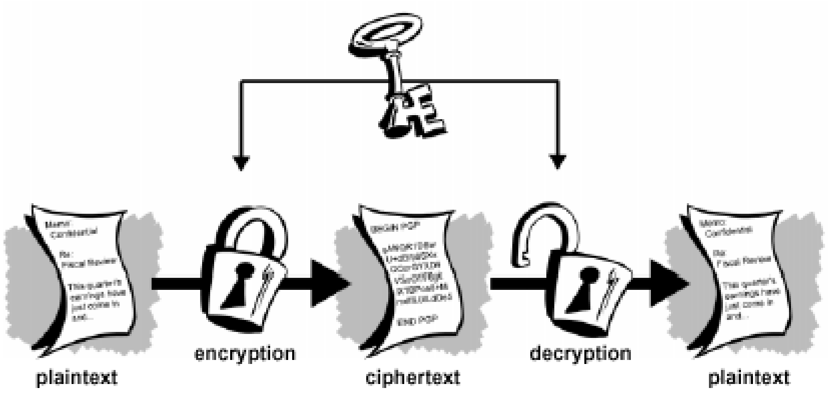


Figure 6-2 Symmetric-Key Encryption

Implementations of symmetric-key encryption can be highly efficient, so that users do not experience any significant time delay as a result of the encryption and decryption. Symmetric-key encryption also provides a degree of authentication, since information encrypted with one symmetric key cannot be decrypted with any other symmetric key. Thus, as long as the symmetric key is kept secret by the two parties using it to encrypt communications, each party can be sure that it is communicating with the other as long as the decrypted messages continue to make sense.

Symmetric-key encryption is effective only if the symmetric key is kept secret by the two parties involved. If anyone else discovers the key, it affects both confidentiality and authentication. A person with an unauthorized symmetric key not only can decrypt messages sent with that key, but can encrypt new messages and send them as if they came from one of the two parties who were originally using the key.

6.4.2 Key Management and Conventional Encryption

Conventional encryption (Symmetric-Key Encryption) has benefits. It is very fast. It is especially useful for encrypting data that is not *going* anywhere. However, conventional encryption alone as a means for transmitting secure data can be quite expensive simply due to the difficulty of secure key distribution. For a sender and recipient to communicate securely using conventional encryption, they must agree upon a key and keep it secret between themselves. The persistent problem with conventional encryption is *key distributio*n: how do you get the key to the recipient without someone intercepting it? This problem is addressed by the second kind of encryption, namely the Public-Key Encryption or Asymmetric Encryption.

6.4.3 Public-Key Encryption

The problems of key distribution are solved by public key cryptography, the concept of which was introduced by Whitfield Diffie and Martin Hellman in 1975. Public Key encryption (Asymmetric encryption) is a cryptographic mechanism that ensures the validity of data as well as who it comes from. The idea behind public key encryption is that every party in the transaction has two keys: a public key and a private key. Each public key is published, and the corresponding private key is kept secret. Data encrypted with your public key can be decrypted only with your private key. Figure shows a simplified view of the way public-key encryption works.

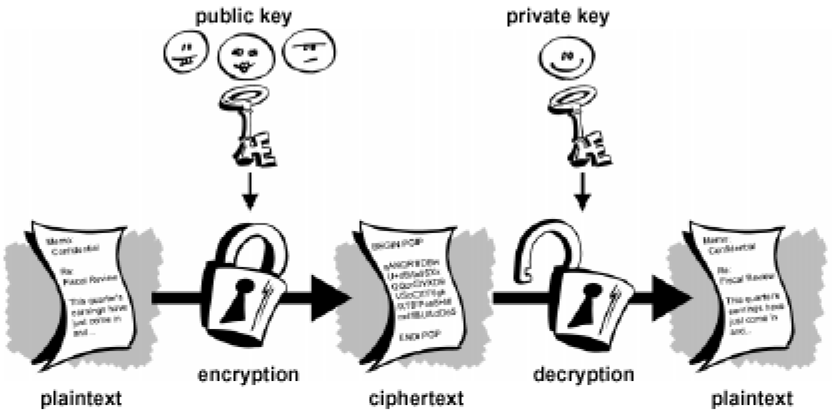


Figure 6-3 Public key encryption

You can freely distribute a public key, and only you will be able to read data encrypted using this key. In general, to send encrypted data to someone, you encrypt the data with that person's public key, and the person receiving the encrypted data decrypts it with the corresponding private key.

It is computationally infeasible to deduce the private key from the public key. Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the information.

Compared with symmetric-key encryption, public-key encryption requires more computation and is therefore not always appropriate for large amounts of data. However, it's possible to use public-key encryption to send a symmetric key, which can then be used to encrypt additional data. This is the approach used by the SSL protocol.

As it happens, the reverse of the scheme shown in Figure 2.2 also works: data encrypted with your private key can be decrypted only with your public key. This would not be a desirable way to encrypt sensitive data because it means that anyone with your public key, which is by definition published, could decrypt the data. Nevertheless, private-key encryption is useful, because it means you can use your private key to sign data with your digital signature; an important requirement for electronic commerce and other commercial applications of cryptography. Client software such as commercial browsers can then use your public key to confirm that the message was signed with your private key and that it hasn't been tampered with since being signed through Digital Signatures.

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. Public-key cryptosystems are RSA (named for its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman), and SHA (Secure Hash Algorithm).

6.4.4 Key Length and Encryption Strength

The strength of encryption is related to the difficulty of discovering the key, which in turn depends on both the cipher used and the length of the key. Encryption strength is often described in terms of the size of the keys used to perform the encryption: in general, longer keys provide stronger encryption. Key length is measured in bits. For example, 128-bit keys for use with the RC4 symmetric-key cipher supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher. Roughly speaking, 128-bit RC4 encryption is 3 x 10^{26} times stronger than 40-bit RC4 encryption.

6.5 Digital Signatures

A major benefit of public key cryptography is that it provides a method for employing *digital signature*s. Digital signatures enable the recipient of information to verify the authenticity of the information's origin, and also verify that the information is intact. Thus, public key digital signatures provide authentication and data integrity. A digital signature also provides non-repudiation, which means that it prevents the sender from claiming that he or she did not actually send the information. These features are every bit as fundamental to cryptography as privacy, if not more.

The manner in which digital signatures are created is illustrated in Figure 1-6. Instead of encrypting information using someone else's public key, you encrypt it with your private key. If the information can be decrypted with your public key, then it must have originated with you.

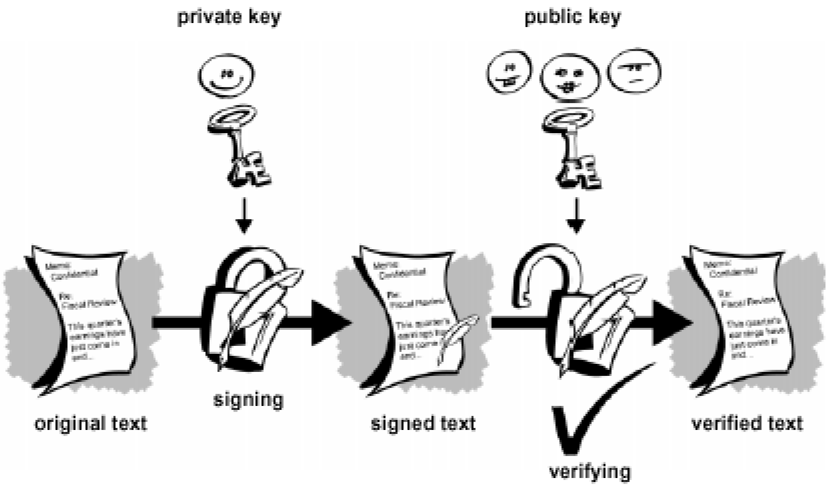


Figure 6-4 Simple digital signatures

6.6 Hash Functions

A one-way hash function takes variable-length input—in this case, a message of any length, even thousands or millions of bits—and produces a fixed-length output; say, 160-bits. The hash function ensures that, if the information is changed in any way—even by just one bit—an entirely different output value is produced.

Public Key Cryptography uses a cryptographically strong hash function on the plaintext the user is signing. This generates a fixed-length data item known as a *message digest.* This digest and the private key are used to create the "digital signature."

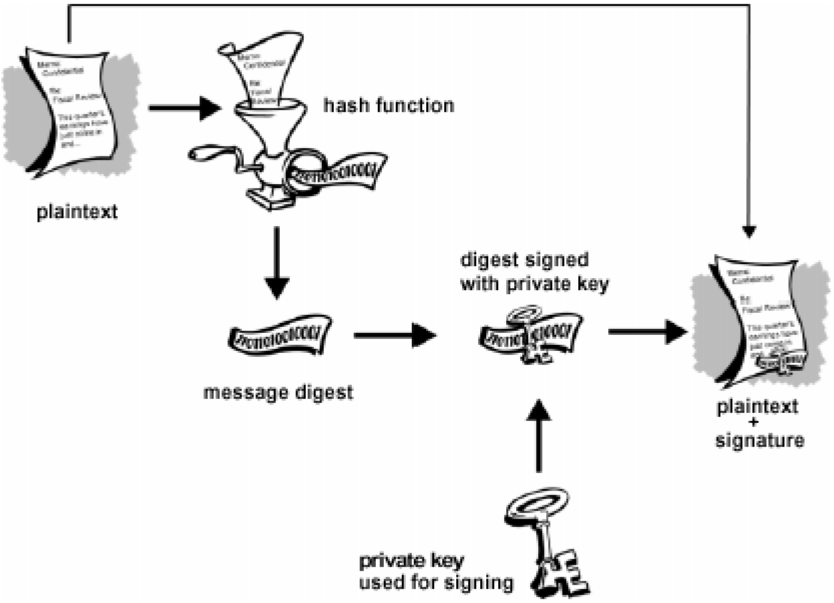


Figure 6-5 Secure digital signatures

As long as a secure hash function is used, there is no way to take someone's signature from one document and attach it to another, or to alter a signed message in any way. The slightest change in a signed document will cause the digital signature verification process to fail.

6.7 Digital Certificates

Digital certificates are the online equivalent of physical proofs of identity, such as passports or driving licenses, helping to identify users communicating across electronic networks. Unique to each individual, a digital certificate may be held on a hard disk, diskette, or, for the ultimate in tamper-proof security, a cryptographic smart card. Digital certificates are the essential element of a PKI. It is issued by a trusted third party, a bank for example, to authenticate the holder.

There are several uses for digital certificates:

1. Establishing secure Internet/intranet connections.
2. Web client authentication.
3. Encrypting and signing secure e-mail
4. Software Publishing

PKI uses a matching key pair, one private and held by the user and one made available on a public directory. Each key performs a one-way transformation of data that can only be reversed by its matching key. The ‘Public’ key is made available to everyone, a bank for example, whilst the ‘Private’ key is kept secret and only accessible by the user. By matching these key pairs and using them to decrypt information, or using them to create a digital signature, a user can be authenticated. Such encryption can be used as proof of identity when using the internet.

6.8 Secure Socket Layer Protocol

The Transmission Control Protocol/Internet Protocol (TCP/IP) governs the transport and routing of data over the Internet. Other protocols, such as the HyperText Transport Protocol (HTTP), Lightweight Directory Access Protocol

(LDAP), or Internet Messaging Access Protocol (IMAP), run "on top of" TCP/IP in the sense that they all use TCP/IP to support typical application tasks such as displaying web pages or running email servers.

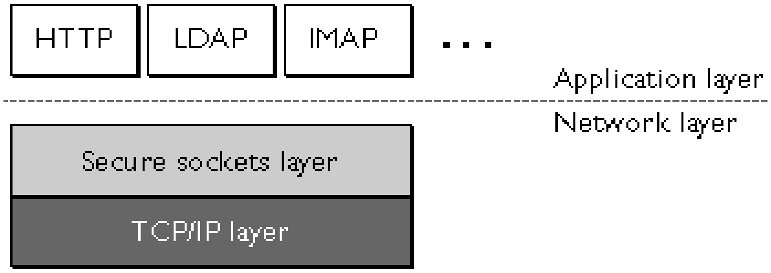


Figure 6-6 SSL runs above TCP/IP and below high-level application protocols

The SSL protocol runs above TCP/IP and below higher-level protocols such as HTTP or IMAP. It uses TCP/IP on behalf of the higher-level protocols, and in the process allows an SSL-enabled server to authenticate itself to an SSL-enabled client, allows the client to authenticate itself to the server, and allows both machines to establish an encrypted connection.

These capabilities address fundamental concerns about communication over the Internet and other TCP/IP networks:

- **SSL server authentication** allows a user to confirm a server's identity. SSL-enabled client software can use standard techniques of public-key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs. This confirmation might be important if the user, for example, is sending a credit card number over the network and wants to check the receiving server's identity.

- **SSL client authentication** allows a server to confirm a user's identity. Using the same techniques as those used for server authentication, SSL-enabled server software can check that a client's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the server's list of trusted CAs. This confirmation might be important if the server, for example, is a bank sending confidential financial information to a customer and wants to check the recipient's identity.
- **An encrypted SSL connection** requires all information sent between a client and a server to be encrypted by the sending software and decrypted by the receiving software, thus providing a high degree of confidentiality. Confidentiality is important for both parties to any private transaction. In addition, all data sent over an encrypted SSL connection is protected with a mechanism for detecting tampering--that is, for automatically determining whether the data has been altered in transit.

The SSL protocol includes two sub-protocols: the SSL record protocol and the SSL handshake protocol. The SSL record protocol defines the format used to transmit data. The SSL handshake protocol involves using the SSL record protocol to exchange a series of messages between an SSL-enabled server and an SSL-enabled client when they first establish an SSL connection. This exchange of messages is designed to facilitate the following actions:

- Authenticate the server to the client.
- Allow the client and server to select the cryptographic algorithms, or ciphers, that they both support.
- Optionally authenticate the client to the server.
- Use public-key encryption techniques to generate shared secrets.
- Establish an encrypted SSL connection.

6.8.1 Ciphers Used with SSL

The SSL protocol supports the use of a variety of different cryptographic algorithms, or ciphers, for use in operations such as authenticating the server and client to each other, transmitting certificates, and establishing session keys. Clients and servers may support different cipher suites, or sets of ciphers, depending on factors such as the version of SSL they support, company policies regarding acceptable encryption strength, and government restrictions on export of SSL-enabled software. Among its other functions, the SSL handshake protocol determines how the server and client negotiate which cipher suites they will use to authenticate each other, to transmit certificates, and to establish session keys.

The cipher suite descriptions that follow refer to these algorithms:

- **DES.** Data Encryption Standard, an encryption algorithm used by the U.S. Government.
- **RSA.** A public-key algorithm for both encryption and authentication. Developed by Rivest, Shamir, and Adleman.
- **RSA key exchange.** A key-exchange algorithm for SSL based on the RSA algorithm.
- **SHA-1.** Secure Hash Algorithm, a hash function used by the U.S. Government.

Key-exchange algorithm RSA key exchange govern the way in which the server and client determine the symmetric keys they will both use during an SSL session. The most commonly used SSL cipher suites use RSA key exchange.

The SSL protocol support overlapping sets of cipher suites. Administrators can enable or disable any of the supported cipher suites for both clients and servers. When a particular client and server exchange information during the SSL handshake, they identify the strongest enabled cipher suites they have in common and use those for the SSL session.

Decisions about which cipher suites a particular organization decides to enable depend on trade-offs among the sensitivity of the data involved, the speed of the cipher, and the applicability of export rules.

6.9: Installing Apache

Get the Win32 version of the Apache web server from one of the mirrors. It is called something like apache_x_y_z_win32.exe. This is a self-extracting archive that contains the Apache base system and sample configuration files.

Don't mix Apache versions 1.3 and 2! It won't work. If you find 1.3.x on modssl.org, you cannot expect it to work with 2.0.x.

Note: You can skip this step and get a full Apache+SSL distribution from modssl.org, as described below. There will be no fancy installation program but you won't need to overwrite the stock Apache files. This is the better way if you are experienced and don't fear editing configuration files (which you will need to do anyway).

Change at least the following parameters in Apache-dir/conf/httpd.conf:
[Replace all occurences of www.my-server.dom with the real domain name!]

- Port 80 to # Port 80 (Comment it out; Port is not necessary, Listen overrides it later.)
- (if **not** in addition to IIS) Listen 80
- Listen 443 (So your server listens on the standard SSL port)
- ServerName **www.my-server.dom**
- (if in addition to IIS) DocumentRoot and the corresponding <Directory some-dir> to your Inetpub\wwwroot

Install the Apache service (NT/2000 only) and start the server. Verify that everything works before proceeding to the SSL installation because this limits the possible errors.

Try **http://www.my-server.dom:443/**. It won't be encrypted yet but if this works then the port configuration (port 443) is right.

6.10: Getting OpenSSL and mod_ssl

Unzip Apache_X-mod_ssl_Y-openssl_Z-WIN32[-i386].zip. it to a new directory. If you need the newest version, you will have to compile it yourself if it is not there.

Copy the files ssleay32.dll and libeay32.dll from the Apache/modssl distribution directory to WINNT\System32.

Copy openssl.cnf to the directory openssl.exe is in.

6.10.1: Creating a test certificate

SERVER CERTIFICATE:

1. Create the key and request:

   ```
   openssl req -new > new.cert.csr
   ```

2. remove the passphrase from the key (optional):

   ```
   openssl rsa -in privkey.pem -out new.cert.key
   ```

3. convert request into signed cert:

   ```
   openssl x509 -in new.cert.csr -out new.cert.cert -req -signkey
   new.cert.key -days 365
   ```

4. The Apache-SSL directives that you need to use the resulting cert are:

   ```
   SSLCertificateFile /path/to/certs/new.cert.cert
   SSLCertificateKeyFile /path/to/certs/new.cert.key
   ```

CA CERTIFICATE:

1. create the key and request:

   ```
   openssl req -new > new.cert.csr
   ```

2. remove the passphrase from the key (optional):

   ```
   openssl rsa -in privkey.pem -out new.cert.key
   ```

3. convert request into signed cert:

   ```
   openssl x509 -in new.cert.csr -out new.cert.cert -req -signkey
   new.cert.key -days 365
   ```

4. The Apache-SSL directives that you need to use the resulting cert are:

   ```
   SSLCertificateFile /path/to/certs/new.cert.cert
   SSLCertificateKeyFile /path/to/certs/new.cert.key
   ```

Step two - sign the client request with the CA key:

```
  openssl x509 -req -in client.cert.csr -out client.cert.cert -signkey my.CA.key -CA
my.CA.cert -CAkey my.CA.key -CAcreateserial -days 365
```

Step three - issue the file 'client.cert.cert' to the requester.

The Apache-SSL directives that you need to validate against this cert are:

```
  SSLCACertificateFile /path/to/certs/my.CA.cert
  SSLVerifyClient 2
```

6.10.2: Configuring Apache and mod_ssl

Copy the executable files (*.exe, *.dll, *.so) from the downloaded apache-mod_ssl distribution over your original Apache installation directory (remember to stop Apache first and DO NOT overwrite your edited config files etc.!).

Find the LoadModule directives in your httpd.conf file and add this after the existing ones, according to the file you have found in the distribution:

```
LoadModule ssl_module modules/ApacheModuleSSL.dll
```

or

```
LoadModule ssl_module modules/ApacheModuleSSL.so
```

or

```
LoadModule ssl_module modules/mod_ssl.so
```

in newer versions.

In newer versions of the distribution, it could also be necessary to add

```
AddModule mod_ssl.c
```

after the AddModule lines that are already in the config file.

Add the following to the end of httpd.conf:

```
# see http://www.modssl.org/docs/2.8/ssl_reference.html for more info
SSLMutex sem
SSLRandomSeed startup builtin
SSLSessionCache none

SSLLog logs/SSL.log
SSLLogLevel info
# You can later change "info" to "warn" if everything is OK

<VirtualHost www.my-server.dom:443>
SSLEngine On
SSLCertificateFile conf/ssl/my-server.cert
SSLCertificateKeyFile conf/ssl/my-server.key
</VirtualHost>
```

Don't forget to call apache with -D SSL if the IfDefine directive is active in the config file!

You might need to use regedit to change the key HKEY_LOCAL_MACHINE\SOFTWARE\Apache Group\Apache\X.Y.Z to the correct number if the apache.exe from modssl.org/contrib is not the same version as the previously installed one. (This seems not to be necessary with recent versions.)

Also, if you use IfDefine directives and start apache as a service, you need to edit the apache command line in the registry (HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Apache2) (I haven't tried this).

Start the server, this time from the command prompt (not as a service) in order to see the error messages that prevent Apache from starting. If everything is OK, (optionally) press CTRL+C to stop the server and start it as a service if you prefer.

CHAPTER 7

INSTALLATION INSTRUCTIONS

Electronic Payment System

7.1. Required Components (Server Side)

1. To install Electronic Payment System on your system you must have the following components which are provided along with the software.

a. Apache web server with mod_SSL.

b. MySQL.

c. PHP

7.1.1 Hardware Requirement:

1 .Minimum Requirement

a)	CPU	Pentium-I, Pentium-II
b)	RAM	64 MB
c)	HDD	2 GB
d)	Modem	56 k
e)	Color SVGA 14”	Monitor
f)	Standard Mouse	
g)	Standard Keyboard	

2. Standard Requirement

a)	CPU	Pentium-III
b)	RAM	128 MB
c)	HDD	4.2 GB
d)	Modem	56 k
e)	Color SVGA 14”	Monitor
f)	Standard Mouse	
g)	Standard Keyboard	

7.1.2 Installation Instructions

You should carryout following steps to install Electronic Payment System.

a. Install **Apache mod_SSL web server** provided in the package. Just double click on the **apache web server** zip file and select where you want to copy apache and it will be installed on your system.

e.g “c:/apache”

b. Place PHP folder in the directory of your choice. Note the path. This would be needed in the configuration file of **Apache mod_SSL web server.**

e.g “c:/php”

c. Install MySQL for databases provided in the package. Chose the path where you want to install MySQL and it will be installed,

e.g “c:/mysql”

.

7.1.3 Configuration of Apache Web Server

For the system to work the Apache Web Server has to be configured as follows.

a. Change directory paths

Edit the file httpd.conf in the folder install_dir/conf and change all the paths to the path where you have stored/installed Apache.

b. Change the port from 8080 to 80

Edit *install_dir*\conf\httpd.conf and change the line that says

<Parameter name="port" value="8080"/> to

<Parameter name="port" value="80"/>**.**

This is optional, but if you have no other server already running on port 80, it will make it easier to enter URLs.

d. Modify “httpd.conf” to set PHP and to ignore existing CLASSPATH.

The server needs to know the location of the main PHP install directory, since apache needs the compiler. But, the compiler is not part of the standard library. Also, most people prefer that apache does not pick up their development CLASSPATH. Otherwise, something might work in a development test case that fails in deployment. If the server's CLASSPATH is only the standard one, the same code will work during development as during deployment. That is, the development code won't run unless you properly place it in the right deployment directories.

Add the following lines to the end of the file install_dir/conf/httpd.conf:

```
LoadModule php4_module M:/Project/PHP/sapi/php4apache.dll
AddModule mod_php4.c
  AddType application/x-httpd-php .php .php3 .phtml
```

7.1.4 Configuration of mod_SSL

a. Generate an SSL server keypair and certificate

The following command generates a keypair that will be used in the SSL connections. Note that the section highlighted should be replaced by the user. This is the distinguished name of the server which is made up of several parts. The first part ***CN=*** is the fully qualified hostname of the machine. The second part ***O=*** is the organization and will be the same as the organization name for the CA root certificate and is usually your company name. The ***C=*** section is simply the 2 letter ISO country code.

C:\> keytool -storetype jks -storepass changeit -alias ssl_server -genkey -keyalg RSA -keysize 1024 -keypass changeit **-dname "CN=compter name , O=organization, C=PK"**

C:\> keytool -storetype jks -storepass changeit –list

Keystore type:jks

Keystore provider: SUN

Your keystore contains 1 entry:

```
ssl_server, Wed Jun 27 20:01:23 EST 2001, keyEntry, Certificate
fingerprint (MD5): 20:1A:9E:B9:1A:F3:2A:DD:0D:C2:9B:28:AA:DA:BE:DB
```

If this command succeeds then a key pair will be generated and saved in a file called **.keystore** in the user’s home directory. It should be noted that this will have the password **changeit.**

b. Configure apache web server for SSL

The standard apache distribution has SSL disabled in the configuration. This should be enabled allowing secure communications between the browser and application when bootstrapping and configuring Keystorm. The configuration file **httpd.conf** which resides in the **apache/conf** directory must be changed. The following section shows how the file should look after configuring SSL.

1. Add port number 443 for SSL.
2. At the end of the file add the following lines

```
LoadModule ssl_module modules/mod_ssl.so

SSLMutex sem
SSLRandomSeed startup builtin
SSLSessionCache none
SSLLog logs/SSL.log
SSLLogLevel info
```

3. You can later change "info" to "warn" if everything is OK

```
<VirtualHost localhost:443>
SSLEngine On
SSLCertificateFile conf/ssl/my-server.cert
SSLCertificateKeyFile conf/ssl/my-server.key
```

c. Test For apache and SSL

Open browser and type URL as

http://localhost/

It will open Apache index.html page than type next URL as

https://localhost/

It will also open same apache index page. If it is alright than now your ready to start Electronic Payment System.

7.1.5 Setting Up database

a. Installing MySQL

Install MySQL from the folder MySQL provided in the package to the directory of your choice. You are now ready to create databases at your leisure or use the existing database provided in the package,

b. Selecting a login name and password

Run winmysqladmin.exe from mysql/bin. A window will appear prompting for a user name and a password. The user name and password you type will be used to connect to the database from PHP.

c. Creating database

To make the database effective and run the application in proper way you have to create MySQL database. To create database follow the following steps:

1. Run command prompt.
2. Go to the directory where you installed MySQL.
3. Write down the following statements:
    a. cd bin

```
b.  mysql.exe

c.  CREATE DATABASE info ;

d.  connect info ;

e.  CREATE TABLE user_profile (

        name VARCHAR(40) NOT NULL,

        user_id VARCHAR(20) NOT NULL,

        password VARCHAR(20) NOT NULL,

        address_line1 VARCHAR(40) NOT NULL,

        address_line2 VARCHAR(40) DEFAULT NULL,

        city VARCHAR(20) NOT NULL,

        country VARCHAR(20) NOT NULL,

        pin VARCHAR(20) NOT NULL,

        gender VARCHAR(20) NOT NULL,

        age VARCHAR(20) NOT NULL,

        email_id VARCHAR(20) NOT NULL,

        phone_number VARCHAR(20) NOT NULL,

        card_no VARCHAR(20) NOT NULL,

        expiry_date VARCHAR(20) NOT NULL,

        card_type VARCHAR(20) NOT NULL,

        account_balance float NOT NULL,

        PRIMARY KEY(user_id));

f. CREATE TABLE book_shop (

        item_no VARCHAR(20) NOT NULL,

        item_type VARCHAR(20) NOT NULL,
        title VARCHAR(60) NOT NULL,
        author VARCHAR(60) NOT NULL,
        price float NOT NULL,
        PRIMARY KEY(item_no));
g. CREATE TABLE music_shop (
        item_no VARCHAR(20) NOT NULL,
        item_type VARCHAR(20) NOT NULL,
        title VARCHAR(60) NOT NULL,
        artist VARCHAR(60) NOT NULL,
        price float NOT NULL,
        PRIMARY KEY(item_no));
h. CREATE TABLE transaction (
        order_no INT NOT NULL primary key auto_increment,
        user_id VARCHAR(20) NOT NULL,
        item_no VARCHAR(20) NOT NULL,
        quantity INT NOT NULL DEFAULT 0,
        date date NOT NULL,
        status VARCHAR(20) NOT NULL);
i.CREATE INDEX index_on_book_item_no ON book_shop(item_no) ;
j. CREATE INDEX index_on_book_title ON book_shop(title) ;
k. CREATE INDEX index_on_book_author ON book_shop(author) ;
l. CREATE INDEX index_on_music_item_no ON music_shop(item_no) ;
m. CREATE INDEX index_on_music_title ON music_shop(title) ;
n. CREATE INDEX index_on_music_artist ON music_shop(artist) ;
```

d. Using original database

You can also use the original database which is provided in the package. Just copy the info folder in the directory …../data from the path where you installed MySQL. This will setup the database in your system.

7.1.6 Launching of Electronic Payment System

Copy “eps” directory/folder from given package and place it in the directory “C:/apache/htdocs”. Now start apache web server as before and open browser and write URL as under

“http://localhost/eps/”

It will open the flash intro of our site.

Press enter to enter the main page of the site. This will open the Login page of the site. New users can login from the link given. Registered users can enter their login ID and password to continue.

Enjoy browsing on the site.

Administrators can open the administration page of the site by entering the following URL

“http://localhost/eps/admin.htm”

7.2. Required Components (Client Side)

To run Electronic Payment System on your system you must have the following components:

1. Any one of the following platforms.

   Red Hat Linux, Ms-Windows 98, Ms-Windows NT, Ms-Windows 2000 etc

2. Any one of the browsers

Ms-Internet Explorer, Netscape Navigator, Mozilla etc

7.2.1 Hardware Requirement:

1 .Minimum Requirement

a)	CPU	Pentium-I, Pentium-II
b)	RAM	64 MB
c)	HDD	2 GB
d)	Modem	56 k
e)	Color SVGA 14”	Monitor
f)	Standard Mouse	
g)	Standard Keyboard	

2. Standard Requirement

a)	CPU	Pentium-III
b)	RAM	128 MB
c)	HDD	4.2 GB
d)	Modem	56 k
e)	Color SVGA 14”	Monitor
f)	Standard Mouse	
g)	Standard Keyboard	

7.2.2 Getting Ready to start

You can run the Electronic Payment System by using the following steps:

1. Open a web browser.
2. Type the following URL in the address bar

“http://SERVERNAME/eps/”

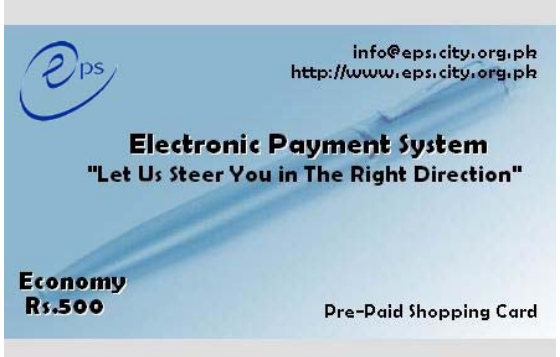
The system will be started on your computer.

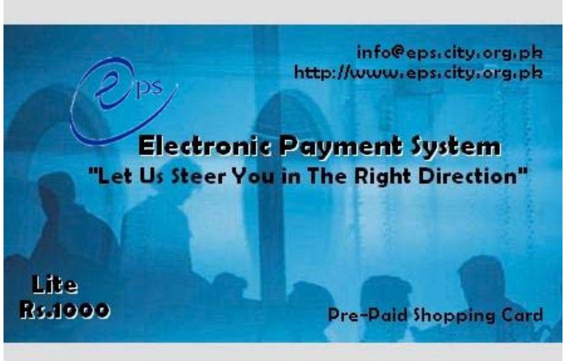
SERVERNAME is the name of the server where the files are kept.

CHAPTER 8

VISUAL INTERFACE

8.1 Scratch Cards


eps
info@eps.city.org.pk
http://www.eps.city.org.pk
Electronic Payment System
"Let Us Steer You in The Right Direction"
Economy
Rs.500
Pre-Paid Shopping Card


eps
info@eps.city.org.pk
http://www.eps.city.org.pk
Electronic Payment System
"Let Us Steer You in The Right Direction"
Lite
Rs.1000
Pre-Paid Shopping Card

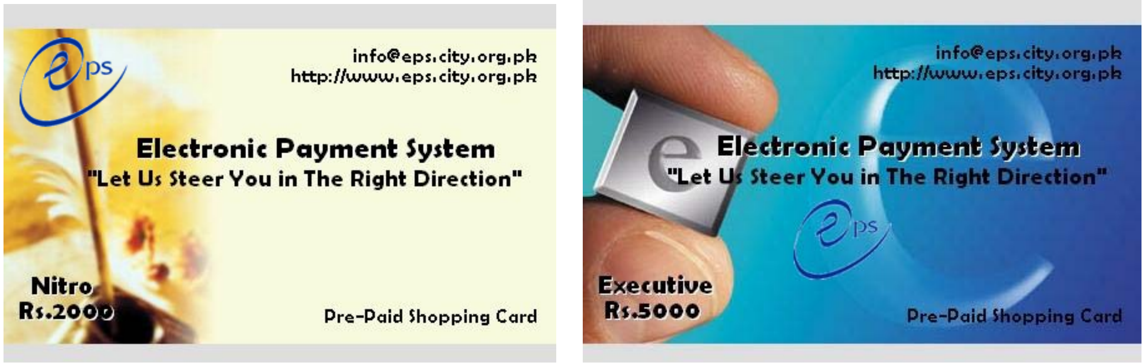
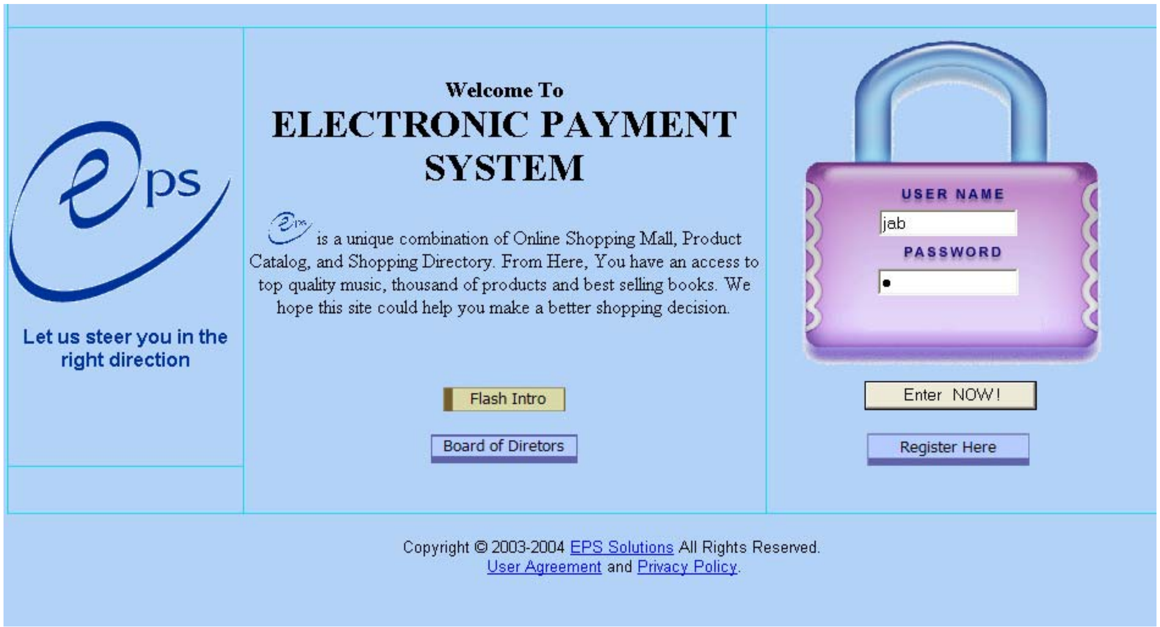

info@eps.city.org.pk
http://www.eps.city.org.pk
Electronic Payment System
"Let Us Steer You in The Right Direction"
Nitro
Rs.2000
Pre-Paid Shopping Card
info@eps.city.org.pk
http://www.eps.city.org.pk
Electronic Payment System
"Let Us Steer You in The Right Direction"
Executive
Rs.5000
Pre-Paid Shopping Card

Figure 8-1 Scratch Cards

8.2 User Interface



Let us steer you in the right direction

Welcome To
ELECTRONIC PAYMENT SYSTEM

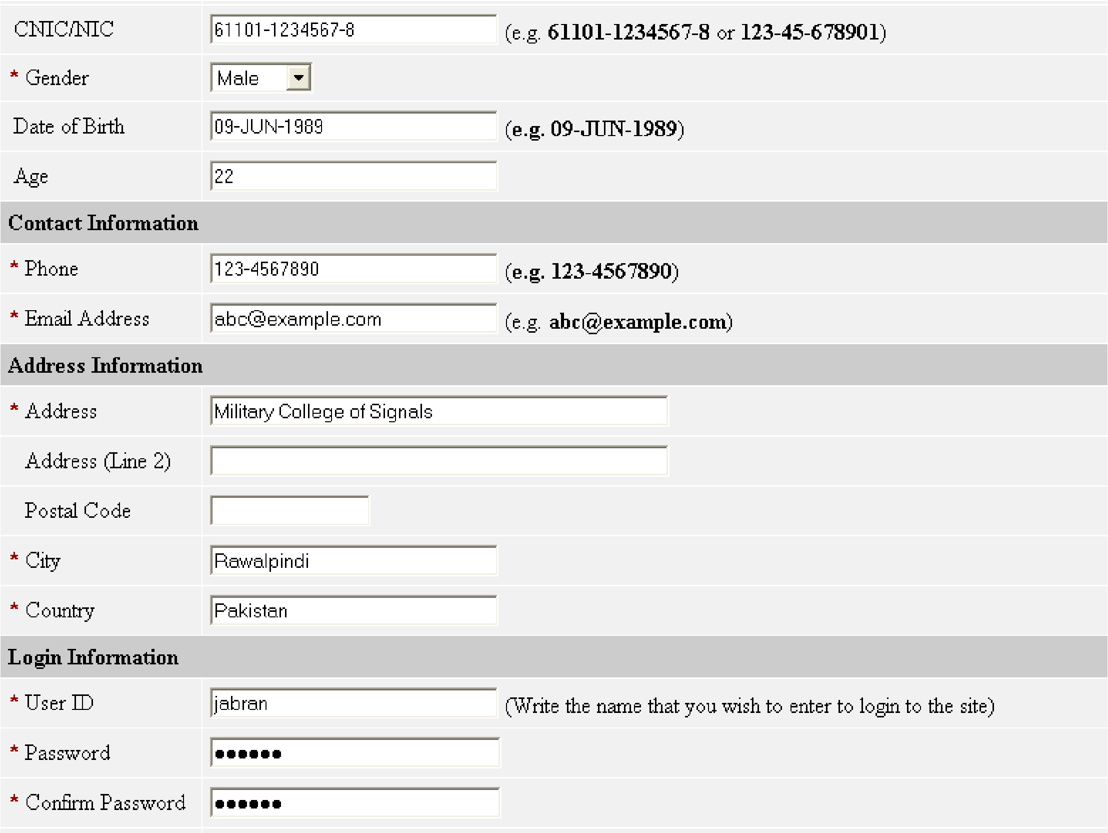
is a unique combination of Online Shopping Mall, Product Catalog, and Shopping Directory. From Here, You have an access to top quality music, thousand of products and best selling books. We hope this site could help you make a better shopping decision.

USER NAME: jab

PASSWORD: •

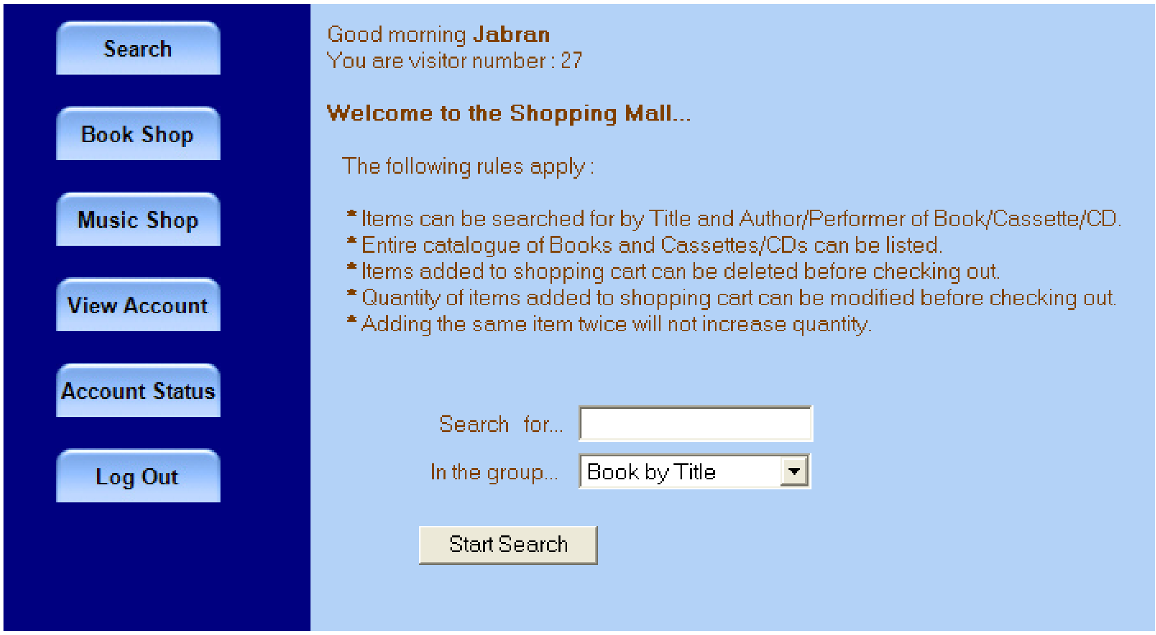
Copyright © 2003-2004 EPS Solutions All Rights Reserved.
User Agreement and Privacy Policy.

Figure 8-2 User's login Form



CNIC/NIC	61101-1234567-8	(e.g. **61101-1234567-8** or **123-45-678901**)
* Gender	Male	
Date of Birth	09-JUN-1989	(**e.g. 09-JUN-1989**)
Age	22	
Contact Information		
* Phone	123-4567890	(**e.g. 123-4567890**)
* Email Address	abc@example.com	(e.g. **abc@example.com**)
Address Information		
* Address	Military College of Signals	
Address (Line 2)		
Postal Code		
* City	Rawalpindi	
* Country	Pakistan	
Login Information		
* User ID	jabran	(Write the name that you wish to enter to login to the site)
* Password	••••••	
* Confirm Password	••••••	

Figure 8-3 User's Registration Form



Good morning **Jabran**
You are visitor number : 27

Welcome to the Shopping Mall...

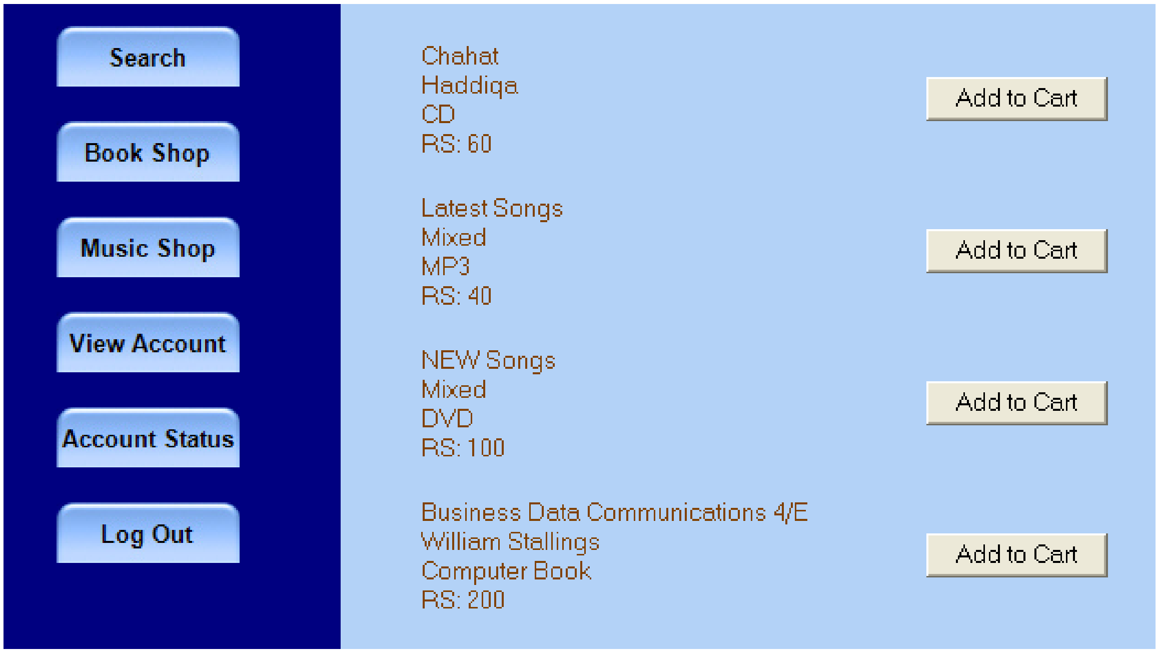
The following rules apply :

* Items can be searched for by Title and Author/Performer of Book/Cassette/CD.
* Entire catalogue of Books and Cassettes/CDs can be listed.
* Items added to shopping cart can be deleted before checking out.
* Quantity of items added to shopping cart can be modified before checking out.
* Adding the same item twice will not increase quantity.

Search for...

In the group... Book by Title

Figure 8-4 user's main search page



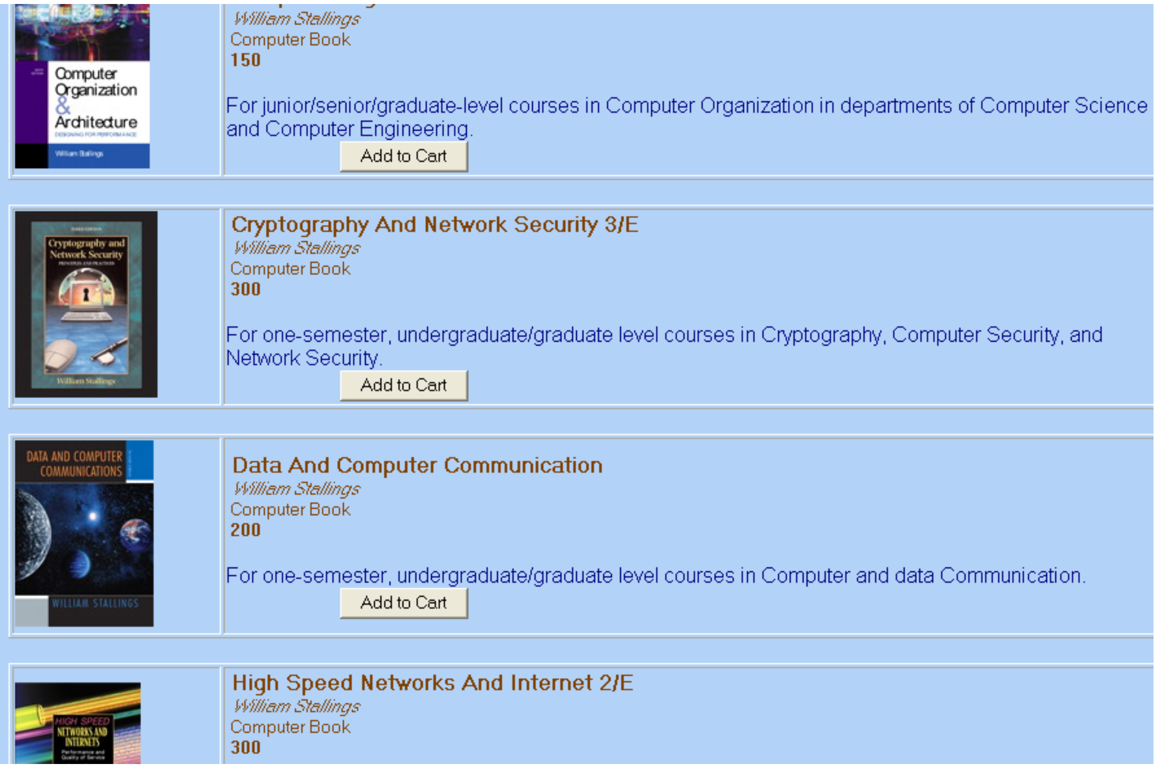
Chahat
Haddiqa
CD
RS: 60

Latest Songs
Mixed
MP3
RS: 40

NEW Songs
Mixed
DVD
RS: 100

Business Data Communications 4/E
William Stallings
Computer Book
RS: 200

Figure 8-5 Searched items page


Computer Organization & Architecture

William Stallings
Computer Book
150

For junior/senior/graduate-level courses in Computer Organization in departments of Computer Science and Computer Engineering.

Cryptography and Network Security
William Stallings

Cryptography And Network Security 3/E
William Stallings
Computer Book
300

For one-semester, undergraduate/graduate level courses in Cryptography, Computer Security, and Network Security.

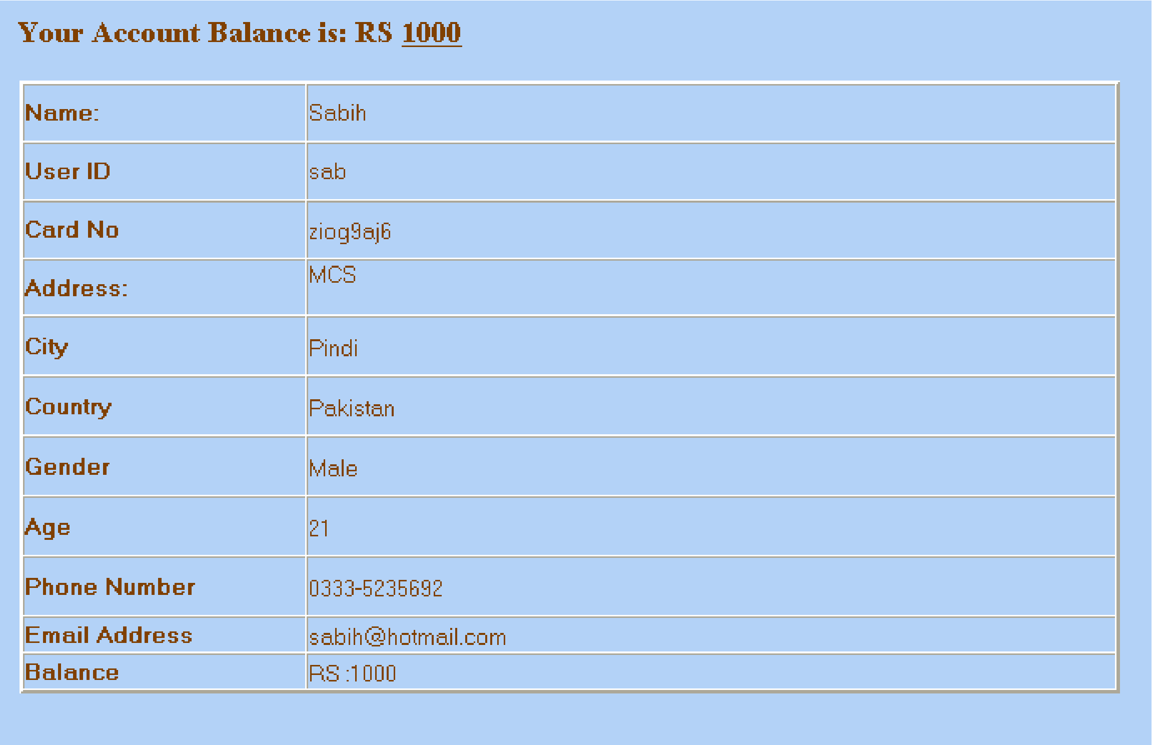
DATA AND COMPUTER COMMUNICATIONS
WILLIAM STALLINGS

Data And Computer Communication
William Stallings
Computer Book
200

For one-semester, undergraduate/graduate level courses in Computer and data Communication.

High Speed Networks And Internet 2/E
William Stallings
Computer Book
300

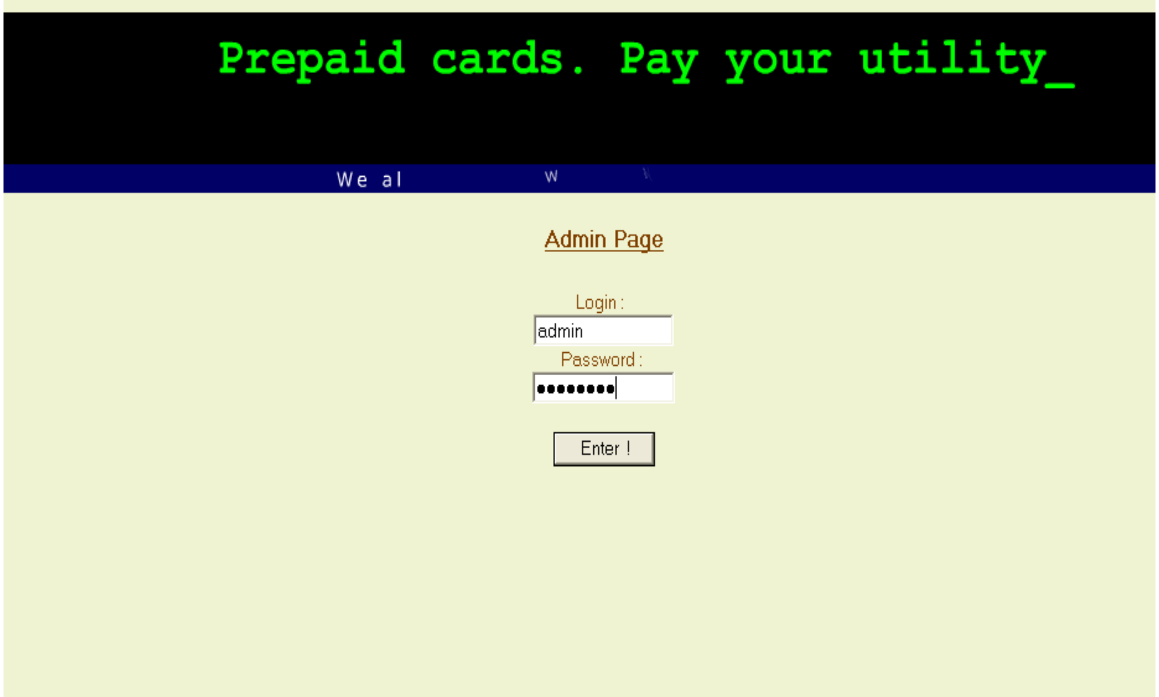
Figure 8-6 Books Listing



Your Account Balance is: RS 1000

Name:	Sabih
User ID	sab
Card No	ziog9aj6
Address:	MCS
City	Pindi
Country	Pakistan
Gender	Male
Age	21
Phone Number	0333-5235692
Email Address	sabih@hotmail.com
Balance	RS :1000

Figure 8-7 Users Account Status

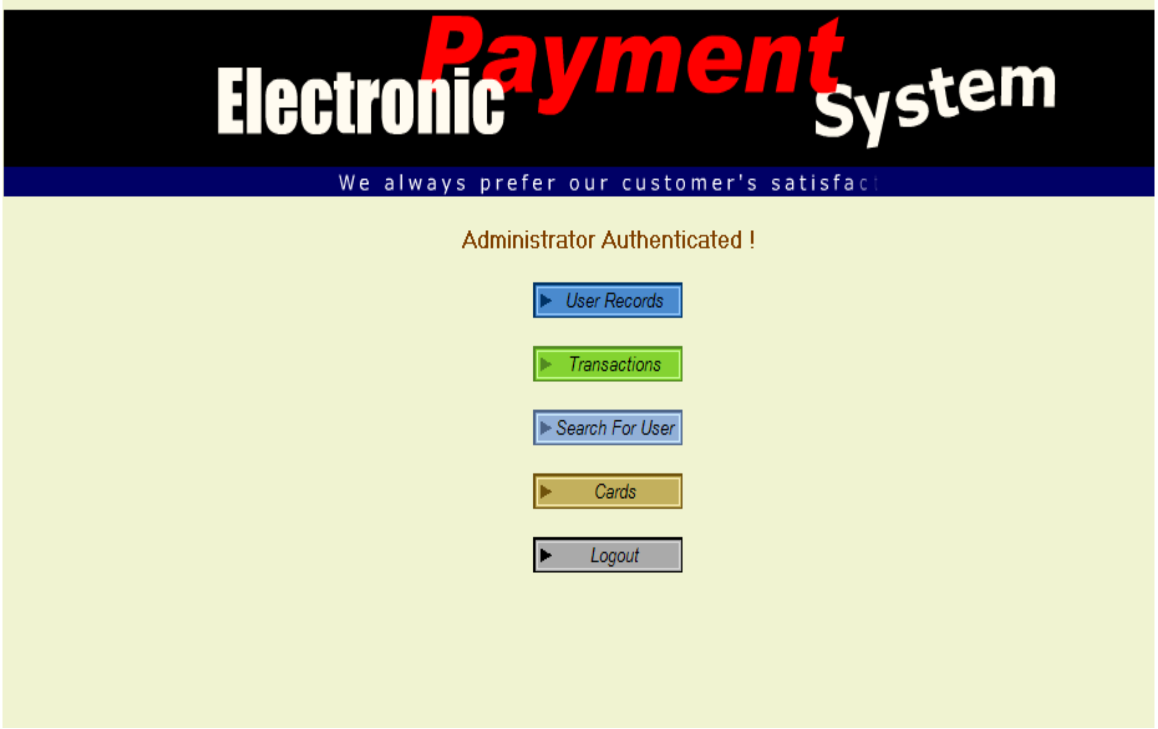


Admin Page

Login : admin

Password : •••••••••

Figure 8-8 Administrators Login Page

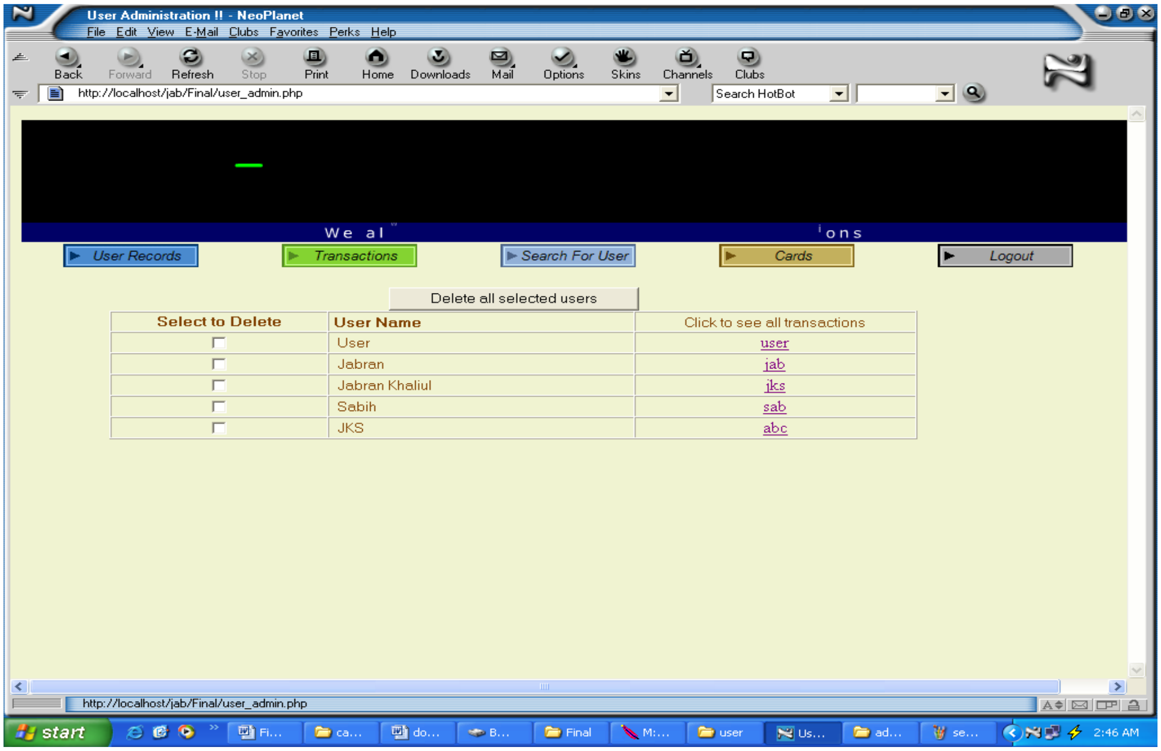


Electronic Payment System

We always prefer our customer's satisfact

Administrator Authenticated !

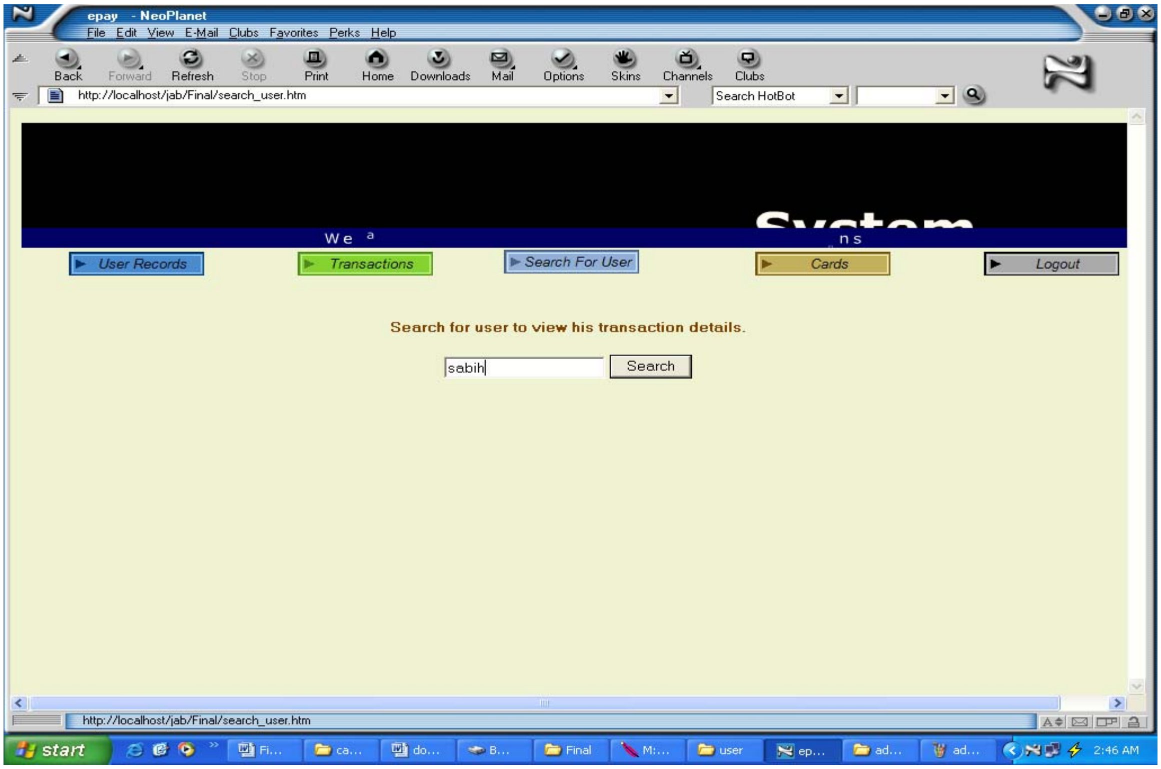
Figure 8-9 Main page of the administrator



Delete all selected users

Select to Delete	User Name	Click to see all transactions
□	User	user
□	Jabran	jab
□	Jabran Khaliul	jks
□	Sabih	sab
□	JKS	abc

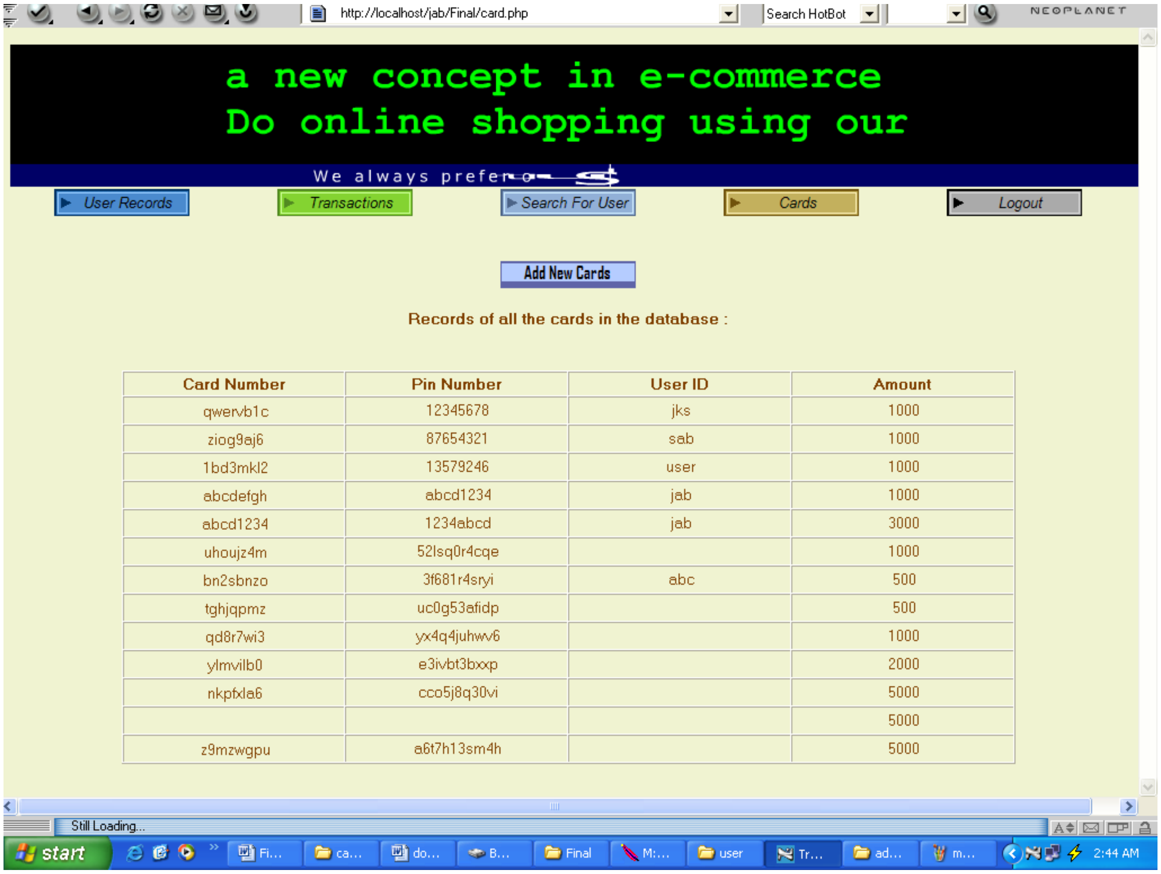
Figure 8-10 Administrator's list all users page


epay - NeoPlanet
File Edit View E-Mail Clubs Favorites Perks Help
Back Forward Refresh Stop Print Home Downloads Mail Options Skins Channels Clubs
http://localhost/jab/Final/search_user.htm
Search HotBot
User Records
Transactions
Search For User
Cards
Logout
Search
http://localhost/jab/Final/search_user.htm
start
2:46 AM

Search for user to view his transaction details.

sabih

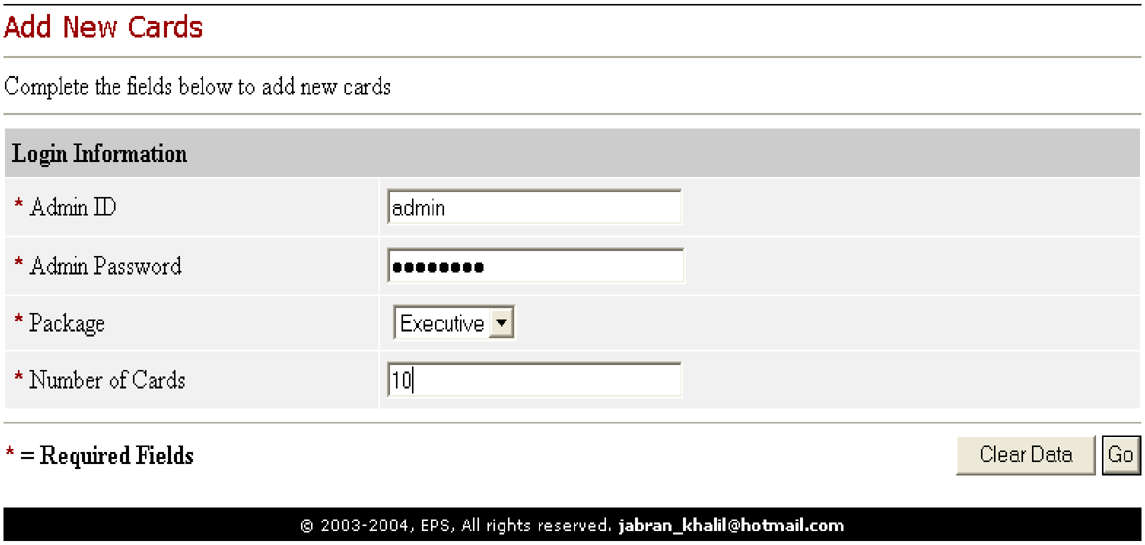
Figure 8-11 Administrator's search for user page



Records of all the cards in the database :

Card Number	Pin Number	User ID	Amount
qwervb1c	12345678	jks	1000
ziog9aj6	87654321	sab	1000
1bd3mkl2	13579246	user	1000
abcdefgh	abcd1234	jab	1000
abcd1234	1234abcd	jab	3000
uhoujz4m	52lsq0r4cqe		1000
bn2sbnzo	3f681r4sryi	abc	500
tghjqpmz	uc0g53afidp		500
qd8r7wi3	yx4q4juhwv6		1000
ylmvilb0	e3ivbt3bxxp		2000
nkpfxla6	cco5j8q30vi		5000
			5000
z9mzwgpu	a6t7h13sm4h		5000

Figure 8-12 Administrator's view all cards page



Add New Cards

Complete the fields below to add new cards

Login Information

* Admin ID	admin
* Admin Password	●●●●●●●●●
* Package	Executive ▾
* Number of Cards	10

* = **Required Fields**

Clear Data
Go
© 2003-2004, EPS, All rights reserved. jabran_khalil@hotmail.com

Figure 8-13 Adding new cards form

Additional Components

As we are using Web Services as a main framework for our software, we have to deeply study their pros and cons so that it's clear why we are choosing Web Services.

Advantages of Web Services

Web Services is a promising new technology that enables applications to "talk" to each other in a distributed environment. Here we discuss some of the features of Web Services that have endeared this new technology to many in the software industry:

1. Based on open standards.

Web Services are based on open, non-proprietary, and freely available standards. Notably, Web Services are based on XML as a data description format and HTTP as a data exchange protocol (although HTTP is not even required). It can be argued that Web Services are an old concept, but it was the creation and evolution of XML and HTTP that have allowed Web Services to be viable.

2. Loosely-coupled, so result in modularity and flexibility.

Loose coupling is one of the fundamental qualities of Web Services. Loose coupling implies that components should know as little as possible about one another, which allows the components to be changed as necessary.

3. Dynamically described, leading to systems that can be upgraded automatically.

In traditional systems if requirements mandate that the new API be used, then often the entire distributed system must be upgraded and fully tested. Such upgrades tend to be expensive and risky, and therefore infrequent. Because Web Services are self-defining and support dynamic discovery, they should solve the problem of this all-or-nothing upgrade problem. Instead of having an API set at design time, each Web Service points to the interface specified in its WSDL file. Consumers of that Web Service know where to find the WSDL file (either because its location was determined at design time, or because it found the file by looking in a UDDI registry).

4. Reduce integration costs.

Each of Web Services' characteristics contributes to their advantages in any integration project. Web Services' reliance on open standards reduces the cost of the required software, and enables integration products from different vendors to interoperate.

5. Internal department-to-department (D2D) Sharing.

Internal department-to-department (D2D) Web services might be appropriate for any function that's even remotely difficult to deploy when the data might be useful to other departments. Web services provide a great way to deploy and maintain software within an organization quickly and efficiently, without many of the risks of accessing Web services outside the firewall. Because D2D Web services are inside your network, the possibility of network trouble causing an outage is much reduced, and levels of volume are much more predictable.

6. Accessibility.

Business services can be completely decentralized and distributed over the Internet and accessed by a wide variety of communications devices. [27]

7. Efficiencies.

Businesses can be released from the burden of complex, slow and expensive software development and focus instead on value added and mission critical tasks. Web services constructed from applications meant for internal use can be easily exposed for external use without changing code. Incremental development using Web services is natural and easy and since Web Services are declared and implemented in a human readable format there is easier bug tracking and fixing. The overall result is risk reduction and more efficient deployability. [27]

8. Simplify Business to Business Integration (B2Bi)

Web Services promise to reduce the cost of integrating systems belonging to two companies just as they promise to reduce the cost of intra-enterprise integration. The use of open standards, loose coupling, and dynamic discovery all enhance Web Services' ability to enable less expensive integration between companies.

Disadvantages of Web services:

Web Services sound so good, but are there any pitfalls to watch out for? Here, we discuss some of the reasons Web Services might not be the ideal tool for the software industry.

1. Immaturity.

The core Web Services standards (SOAP, WSDL, and UDDI) are still in development. Different Web Services implementations are not 100% interoperable. Security, quality of service, billing and metering, transaction support, and composition of Web Services are still on the drawing board. In fact, we are merely in the early days of trying to understand not only the technological requirements for Web Services, but also how the different aspects of Web

Services must interact, and maybe more importantly, the business models that will help propel Web Services to become more than just a niche technology play.

2. Current Web Services are single-vendor specific.

Current major vendor Web Services offerings, such as Microsoft's .Net framework, IBM WebSphere, Sun ONE, and other offerings are these company's Web Services-based models for how computing will be done in the future.

3. Payment Problems.

One of the concepts emerging with the notion of Web Services is the idea that businesses can offer individual or groups of Web Services as a commercial service that can be offered on a per-transaction, subscription, or one-time license fee. The business model of a "Web Service provider" makes sense on the surface, but upon closer inspection, there are serious flaws with this concept. Just as putting up a Web site doesn't guarantee traffic, and traffic doesn't guarantee profitability, the same goes for Web Services.

4. Interoperability.

For Web services architecture to realize its full potential (as an architecture for cross-platform program-to-program communications), interoperability between various vendor's platforms and Web services implementations must be assured. And achieving interoperability is not a short-term thing -- as new W3C specifications and recommendations are released and as new vendors enter the market, more and more interoperability testing will need to take place between broader and broader mixes of vendors and products.

5. Lack of standard language mappings.

There is no portable, standard API for SOAP, so the developer is left in the dark about how to map SOAP data to implementation data and vice versa. If you use a web services product for, say, Java from one vendor and then want to switch to a web service product for Java from another vendor, you get to rewrite all your code, because the new vendor’s toolkit and API are different.

6. Lack of standard services, e.g. events.

There is only one standard service available for Web Services: the UDDI interface service. Various other important services are not specified. There is no notification service (event service), no transaction service, no security service. [2]

7. Lack of performance due to the requirement to parse and transport XML.

The use of XML places a heavy burden on your system. Despite the relative simplicity of XML, there are huge memory and CPU requirements compared to other solutions. Generating and parsing XML documents is a time consuming task that also needs a lot of memory compared to the actual data that is in those documents. Complex schema's or DTDs slow down the processing of the structure and content of the document even more.

8. Security/privacy.

We consider Web services architectural security as consisting of two levels:

1. network-level security
2. content level security

At the network level we see a layer of security protection existing at the "line" level with Secure Sockets Layer (SSL) security already being implemented by numerous enterprises.

On the content security side, the W3C has put much effort into securing XML content. Standards recommendations exist for many of the following security/privacy considerations:

- Protect private data/document *confidentiality*.
- *Authenticate* where data/content has originated and validate its origin.
- Provide only *authorized* users with access to certain types of content.
- Ensure the *data and content integrity* that has been sent between communicating entities.
- Provide for *non-repudiation* (a record that shows what transpired and who/what initiated it such that a transaction can be traced along its route and no aspect of the transaction can be denied).

Reference:

1. **Professional PHP Programming**

   Wrox Press Ltd.

2. **Mastering PHP 4.1**

   By *Jeremy Allen and Charles Hornberger*

3. **Special Edition**

   Using HTML 4 *Sixth Edition*

   By *Holzschlag*

4. **MySQL Reference Manual**

   by Michael Widenius, David Axmark, MySQL AB

5. **Inside Adobe Photoshop 6**

   By Gary David Bouton (Editor), Gary Kubicek, Jim Rich, Al Ward

6. **Flash(tm) 5 Virtual Classroom**

   By Doug Sahlin

7. Swish!

   By Michael Sampson, Michael Chesworth (Illustrator), Jr. Martin Bill,

8. Administering IIS (5)

   By Mitch Tulloch, Patrick Santry

9. Web Sites

   www.wrox.com

   www.mysql.com

   www.php.net

   www.zend.com

   www.webwizguide.com

   www.phparena.net/

   www.micrsoft.com

   http://www.faqts.com/knowledge_base/index.phtml/fid/51

   http://www.microsoft.com/office/

   http://mysqlfront.venturemedia.de/

   www.phpbuilders.com