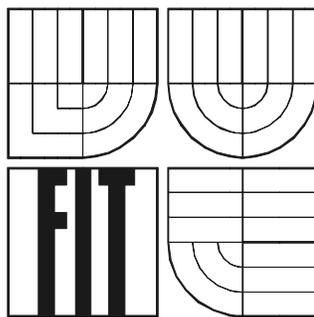


BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY



On-Line Advertisement System

BSc Thesis

2007

Andrej Zito

On-Line Advertisement system

© Andrej Zito, 2005.

This work was created as a school piece on Brno Univeristy of Technology, Faculty of Information Technology. The work is protected by a copyright law and its use without authorization from the author is illegal, except for cases defined by the law.

Statement

I hereby state that this work has been created by me under the supervision of Ing. Petr Blatný.
All publications used are listed in the References chapter.

.....
Andrej Zito
2007.1.23

Acknowledgements

I'd like to express my gratitude toward my supervisor Petr Blatny for a professional help and feedback leading to complete this work.

Abstrakt

Tato práce se zabývá návrhem a realizací on-line inzertního systému. Cílem bylo navrhnout a vytvořit funkční aplikaci poskytující základní úlohy pro koncové uživatele umožňující jim přidávat, přehlížet a vyhledávat inzeráty, registrovat se, posílat příspěvky a soukromé vzkazy. Systém byl implementován za použití následujících technologií: XHTML, PHP, MySQL, CSS, JavaScript, DOM and AJAX.

Klíčová slova

Inzerce, inzeráty, informační systém, webová aplikace, XHTML, PHP, MySQL, CSS, JavaScript, DOM, AJAX, databáze, klient, server

Abstract

This work deals with a concept and realization of on-line advertising information system. The goal was to design and create a functional application providing basic tasks for end users allowing them to add, browse and search advertisements, register, send comments and private messages. The system was implemented using the following technologies: XHTML, PHP, MySQL, CSS, JavaScript, DOM and AJAX.

Key Words

Advertising, advertisement, information system, web application, XHTML, PHP, MySQL, CSS, JavaScript, DOM, AJAX, database, client, server

Table of Contents

Table of Contents.....	9
1 Introduction.....	11
1.1 Report guide.....	12
2 Mission	13
3 Current status	14
3.1 Getting familiar.....	14
3.2 Detailed concept and AJAX	14
3.3 Core functionality	14
3.4 Cross-platform application	15
3.5 Authorization and data protection.....	15
3.6 Usability.....	16
4 Concept.....	17
4.1 Use-case diagram.....	17
4.1.1 Non-registered user.....	17
4.1.2 Registered user.....	18
4.1.3 Administrator	18
4.2 ER diagram	19
5 Technologies used.....	21
5.1 PHP.....	22
5.2 MySQL	23
5.3 XHTML	24
5.4 CSS	25
5.5 JavaScript.....	27
5.6 DOM.....	30

5.7	AJAX	32
6	Usability.....	35
6.1	What makes a website or piece of software usable?.....	35
6.2	Why is Usability Important?.....	35
6.3	How Do You Achieve a High Level of Usability?	36
6.4	Usability tips	36
7	Implementation phases	40
7.1	Design	40
7.2	XHTML templates	40
7.3	MySQL database and sample data.....	40
7.4	Scripts	45
8	Installation and requirements.....	48
8.1	1 st steps.....	48
9	Summary.....	49
10	References.....	50

1 Introduction

According to TechWeb site a definition of information system is as following:

A business application of the computer. It is made up of the database, application programs and manual and machine procedures. It also encompasses the computer systems that do the processing.

Processing the Data

The database stores the subjects of the business (master files) and its activities (transaction files). The application programs provide the data entry, updating, query and report processing.

The Procedures

The manual procedures document how data are obtained for input and how the system's output is distributed. Machine procedures instruct the computer how to perform scheduled activities, in which the output of one program is automatically fed into another.

Transaction Processing

The daily work is the online, interactive processing of the business transactions and updating of customer, inventory and vendor files (master files).

Batch Processing

At the end of the day or other period, programs print reports and update files that were not updated on a daily basis. Periodically, files must be updated for routine maintenance such as adding and deleting employees and making changes to product descriptions.

This report tries to cover a specific type of information systems: advertising information system. After giving this term to Google approximately 164 000000 results are found. However after trying the first ones one will find out that the results have mainly to do with advertising in the sense of marketing usually connected to business companies.

In this case the advertising system targets public and individuals giving them the opportunity to post their needs and offers on the internet. Popular server www.ebay.com is also some kind of an advertising system but is more considered as an auction system. A good example of advertising system I tried to create is www.craigslist.org or some of the Czech servers www.nejinzerce.cz or www.bazos.cz.

My reason for choosing this project was that I'm interested in this area and in web-based information systems in general.

1.1 Report guide

Chapter 2 “**Mission**” lists goals that were set at the beginning of project. Some of them were specified by the faculty, part of them was designed by me.

Chapter 3 “**Current status**” discusses the current status of goals from Chapter 2.

In the chapter 4 “**Concept**” is my result of conceptual model that was an important input for technical implementation. I have used Visio 2007 Beta to create all schemas in this chapter.

First part of chapter 5 “**Implementation**” gives an overview, including pros and cons of tools that have been chosen for this task. Most of the content was taken from [2], [3], [4], [5], [6], [7], [8], [9] and [10].

Chapter 6 “**Usability**” summarizes basics of the usability requirements and tips.

7th chapter “**Implementation phases**” serves as a timeline of the implementation on practical level and was created from my own notes through the project.

Chapter 8 “**Installation and requirements**” was written based on my own experiences gained during the project.

2 Mission

The following requirements were set at the beginning:

- Get familiar with web-based advertisement systems.
- Get familiar with PHP, HTML, AJAX (Asynchronous JavaScript and XML), MySQL and other useful tools.
- Create a detailed concept of the advertisement system. Use AJAX for a client communication.
- Realize a functional prototype of the advertisement system.
- Prototype must be a cross-platform application.
- Authorized access and data protection should be implemented.
- Clear and easy-to-use user interface respecting usability elements.

3 Current status

3.1 Getting familiar

As a regular internet user from time to time I had a chance to use web advertising systems to buy or sell some stuff. This gave me a basic overview of what functions should be in place. During this project I actively tried to use and explore these systems in more detail.

Some of the technologies specified by requirements were already know to me, some of them not. All are listed in the chapter 5.

3.2 Detailed concept and AJAX

An IS concept was created during Term Project (ISP). However it was modified a little bit through the implementation process.

AJAX technology was used to refresh the number of new messages without reloading the web page. The script is coded in 2 files, refresh.js and refresh_msg.php. More details are provided in chapter 7.

3.3 Core functionality

Basic functions were implemented and are ready to use. These include:

- Login of a registered user
- Registration of a new user
- Adding a new advertisement, including a picture
- Browsing and searching the database of ads
- Discussion threads for each ad
- Private messaging among registered users
- Modification of personal data
- Setting up a feed (however no ads are being sent at the moment)

Tasks for administrator have not yet been implemented.

3.4 Cross-platform application

The information system was designed and implemented as a web-based application. This allows users to use the system from any hardware device (PC, Mac, PDA, Smartphone) running any operating system (Windows, Mac OS, Linux) with a web browser that supports XHTML and CSS technology.

According to several analyzes the requirement above shouldn't be a problem for the most of internet users and thus improves the probability of finding more regular users.

3.5 Authorization and data protection

Part of the core functions are only available to registered users. Every new user can register him using the registration form and providing necessary data (login, password, email). The password each user chooses is encrypted using MySQL PASSWORD function.

Once the process is over authorization can take place on the login page. After entering login and password the data are cross-checked with the user table in database. If a match is found the authorization was successful and the user id is stored using PHP sessions. This will be destroyed only by logging out or closing the web browser window.

3.6 Usability

The site followed some basic rules and standards people are used to:

- Logo is placed in the top-left corner and also acts as a link to home page
- Menu with categories and subcategories is listed on the left side
- Utilities can be located in the top-right corner
- The specific content of each page is wrapped in a box entitled by the current page showing clear visual hierarchy
- The top-most bar shows the current position anywhere in the system helping the user not to lose orientation

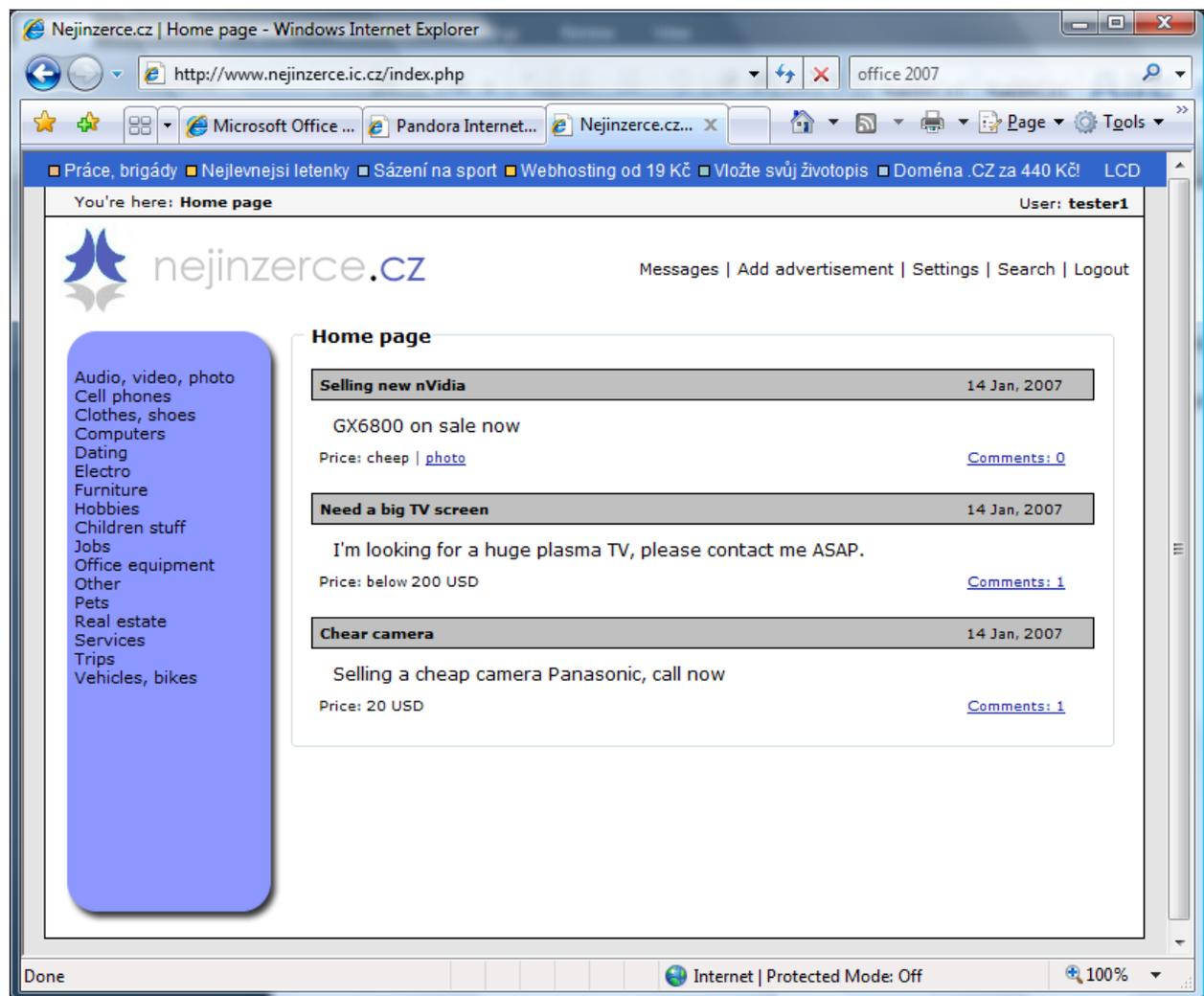


Figure 3.1

4 Concept

4.1 Use-case diagram

Based on the system function specifications 3 types of actors have been formed:

4.1.1 Non-registered user

Everyone who hasn't authorized himself is a non-registered user by default. He still can do some tasks in the system, i.e. add a new advertisement, browse-search the database and add comments. By successfully filling in a registration form the user can promote his status to a registered user.

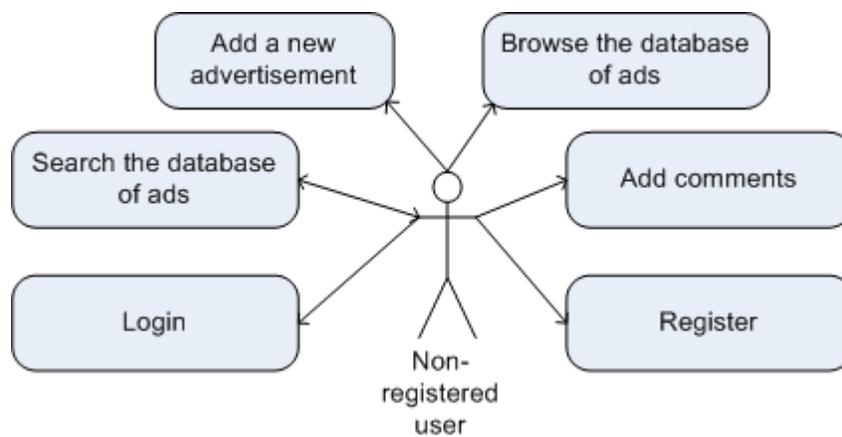


Figure 4.1

4.1.2 Registered user

After a login process the user enters the system as a registered user having the option to perform most of the tasks offered. In addition to tasks of non-registered user he can send private messages to other registered users within the application and change personal information. Last but not least this kind of a user can subscribe to receive new advertisements via email on a regular basis.

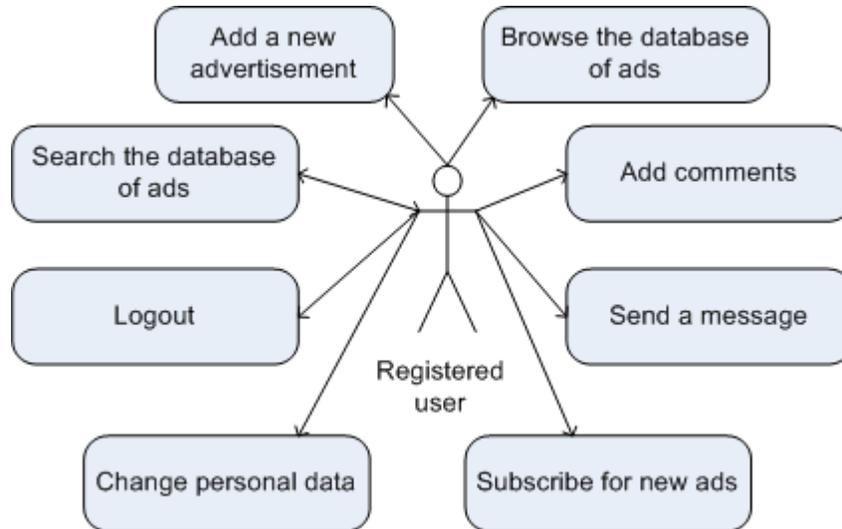


Figure 4.2

4.1.3 Administrator

The ultimate user with no restrictions. He's expected to optimize and make the database safe. He can also block user accounts and delete ads/comments with explicit content.

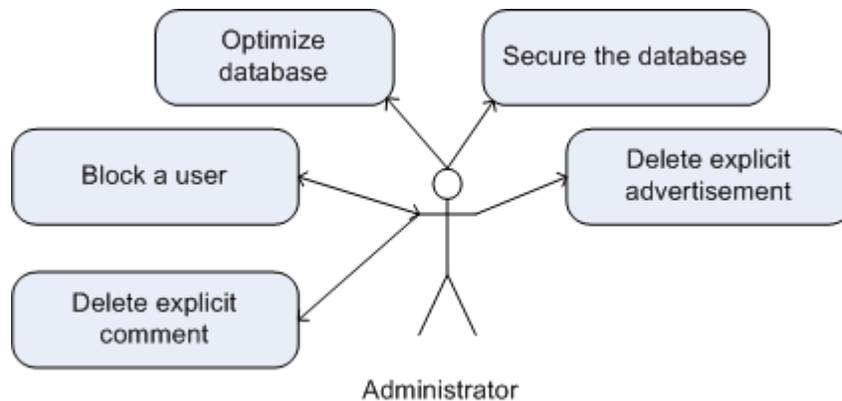


Figure 4.3

4.2 ER diagram

Entity-relation diagram presents the relationship between entities in the whole system. The following entities have been designed during the time:

User

Entity represents all actors in the system. Each user needs to have at least a unique login, password and email address. Other personal information like name, surname and contact (telephone 1 or 2, icq, msn) are optional. The user entity can send and receive messages, adds advertisements, posts comments and chooses a setting.

Advertisement

This entity is described by a heading, text, price, region, date of creation. Each ad is created by some user and belongs to a certain category. It may contain an uploaded photo and comments posted by people.

Category

Each category has a name and a parent category used to build a tree structure of all categories. Advertisement may be posted only to a category without any child that has 5 digits – ending with 0 mean an offer, ending with 1 means a demand. One category per user can be chosen in settings and all new ads from the category will be automatically sent to user's inbox on a regular basis.

Message

Message is holding a subject, text, date of creation and a status. Messages are sent among registered users in the system.

Comment

Comment stores a heading, text and date of creation. It was sent by a specific user in a discussion thread to a specific advertisement.

Setting

User has the ability to subscribe to an automated process and receive new ads from a chosen category into private mailbox (mail stored in user entity) on a regular basis. One user can watch only one category at once.

Photo

Last entity is used to store location of photos uploaded to a server and copied to photos folder. For each advertisement only one photo is allowed.

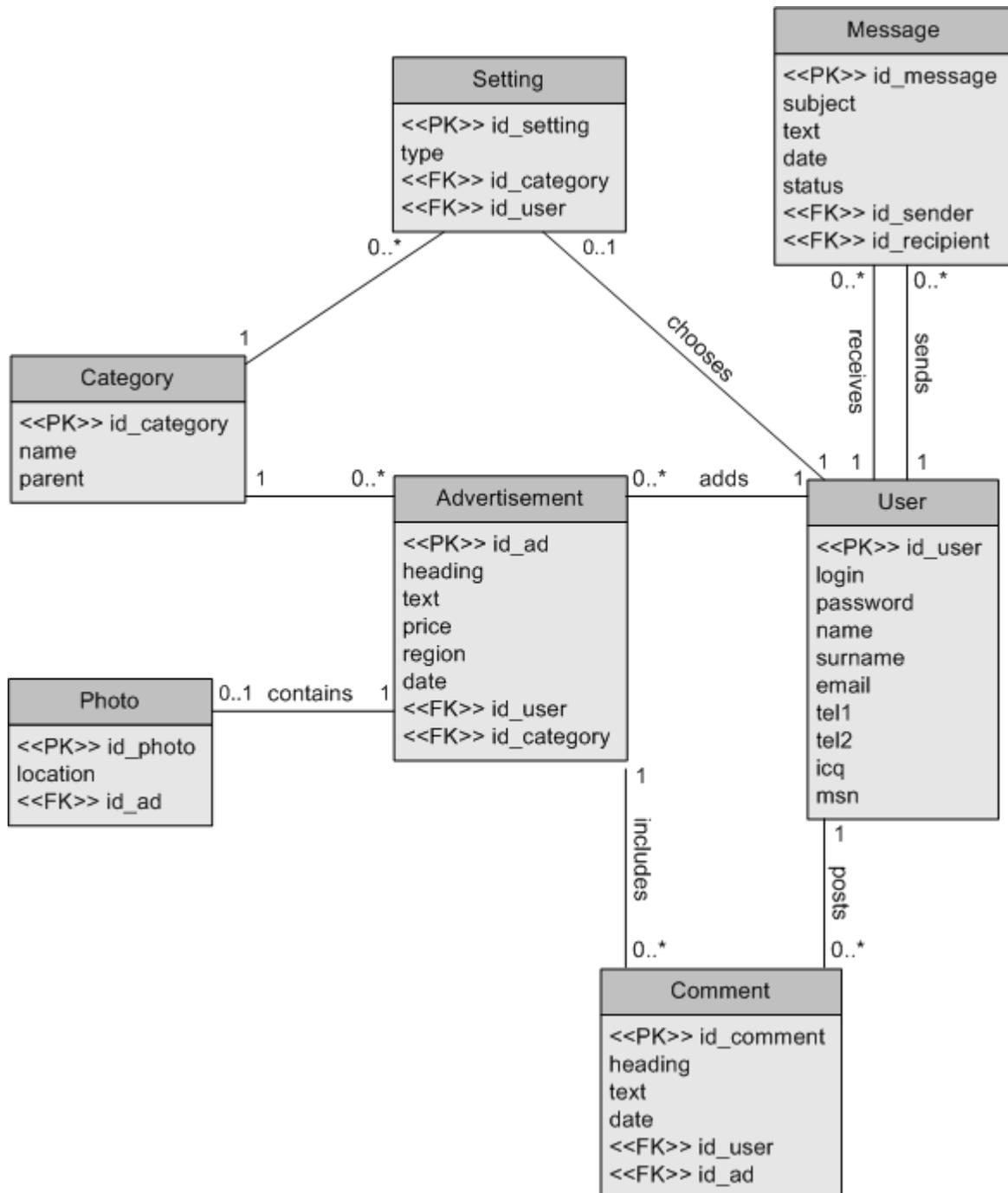


Figure 4.4

5 Technologies used

This project was implemented using the following technologies:

- PHP
- MySQL
- XHTML
- CSS
- JavaScript
- DOM
- AJAX

The reason for choosing the first 4 technologies was that I already had some experiences with them. Moreover PHP and MySQL server are Open Source and are free to use. I took the opportunity of this project and tried to improve my knowledge in this area. The use of remaining 3 technologies was set by project specification and they were completely new for me.

The rest of this chapter gives an overview of technologies used.

5.1 PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is an open-source server-side scripting language for creating dynamic Web pages for e-commerce and other Web applications. A dynamic Web page is a page that interacts with the user, so that each user visiting the page sees customized information. Dynamic Web applications are prevalent in commercial (e-commerce) sites, where the content displayed is generated from information accessed in a database or other external source.

PHP offers a simple and universal solution for easy-to-program dynamic Web pages. The intuitive interface allows programmers to embed PHP commands right in the HTML page. PHP's syntax is similar to that of C and Perl, making it easy to learn for anyone with basic programming skills. Its elegant design makes PHP significantly easier to maintain and update than comparable scripts in other languages.

PHP offers excellent connectivity to most of the common databases (including Oracle, Sybase, MySQL, ODBC and many others), and offers integration with various external libraries, which allow the developer to do anything from generating PDF documents to parsing XML. Another key advantage of PHP, when compared to other scripting languages such as ASP or ColdFusion, is that it is open-source and cross-platform, suitable for today's heterogeneous network environments.

Pros

- Easy combination with (X)HTML
- Great level of support and information
- Relatively easy to use
- Many good function libraries
- Easy accessibility
- Price – free to use
- Great communication with databases

Cons

- Server-side scripting – client must contact and transfer data between the server
- Sometimes problems with code safety

5.2 MySQL

Database management system

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

Relational database management system

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

Open Source

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.

Works in client/server or embedded systems

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

5.3 XHTML

The Extensible HyperText Markup Language, or XHTML, is a markup language that has the same depth of expression as HTML, but a stricter syntax. Whereas HTML is an application of SGML, a very flexible markup language, XHTML is an application of XML, a more restrictive subset of SGML. Because they need to be well-formed (syntactically correct), XHTML documents allow for automated processing to be performed using a standard XML library—unlike HTML, which requires a relatively complex, lenient, and generally custom parser (though an SGML parser library could possibly be used). XHTML can be thought of as the intersection of HTML and XML in many respects, since it is a reformulation of HTML in XML. XHTML 1.0 became a World Wide Web Consortium (W3C) Recommendation on January 26, 2000. XHTML 1.1 became a W3C recommendation May 31, 2001.

Motivation

The need for a more strict version of HTML was felt primarily because World Wide Web content now needs to be delivered to many devices (like mobile devices) apart from traditional computers, where extra resources cannot be devoted to support the additional complexity of HTML syntax.

Another goal for XHTML and XML was to reduce the demands on parsers and user-agents in general. With HTML, user-agents increasingly took on the burden of “correcting” errant documents. Instead XML requires user-agents to fail when encountering malformed XML. This means an XHTML browser can theoretically be faster and made to run more easily on miniaturized devices than a comparable HTML browser. The recommendation for browsers to post an error rather than attempt to render malformed content should help eliminate malformed content. Even when authors do not validate code, and simply test against an XML browser, errors will be revealed.

An especially useful feature XHTML inherits from its XML underpinnings is XML namespaces. With namespaces, authors or communities of authors can define their own XML elements, attributes and content models to mix within XHTML documents. This is similar to the semantic flexibility of the ‘class’ attribute from HTML, but with much more power. Some W3C XML namespaces/schema that can be mixed with XHTML include MathML for semantic math markup and Scalable Vector Graphics for markup of vector graphics.

5.4 CSS

What is CSS?

- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements
- Styles are normally stored in Style Sheets
- Styles were added to HTML 4.0 to solve a problem
- External Style Sheets can save you a lot of work
- External Style Sheets are stored in CSS files
- Multiple style definitions will cascade into one

Styles Solve a Common Problem

HTML tags were originally designed to define the content of a document. They were supposed to say "This is a header", "This is a paragraph", "This is a table", by using tags like `<h1>`, `<p>`, `<table>`, and so on. The layout of the document was supposed to be taken care of by the browser, without using any formatting tags.

As the two major browsers - Netscape and Internet Explorer - continued to add new HTML tags and attributes (like the `` tag and the color attribute) to the original HTML specification, it became more and more difficult to create Web sites where the content of HTML documents was clearly separated from the document's presentation layout. To solve this problem, the World Wide Web Consortium (W3C) - the nonprofit, standard setting consortium, responsible for standardizing HTML - created STYLES in addition to HTML 4.0.

Style Sheets Can Save a Lot of Work

Styles sheets define HOW HTML elements are to be displayed, just like the font tag and the color attribute in HTML 3.2. Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in your Web, just by editing one single CSS document!

CSS is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once. As a Web developer you can define a style for each HTML element

and apply it to as many Web pages as you want. To make a global change, simply change the style, and all elements in the Web are updated automatically.

Multiple Styles Will Cascade Into One

Style sheets allow style information to be specified in many ways. Styles can be specified inside a single HTML element, inside the <head> element of an HTML page, or in an external CSS file. Even multiple external style sheets can be referenced inside a single HTML document.

Cascading Order

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:

1. Browser default
2. External style sheet
3. Internal style sheet (inside the <head> tag)
4. Inline style (inside an HTML element)

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style declared inside the <head> tag, in an external style sheet, or in a browser (a default value).

5.5 JavaScript

JavaScript is the name of Netscape Communications Corporation's implementation of the ECMAScript standard, a scripting language based on the concept of prototype-based programming. The language is best known for its use in websites (as client-side JavaScript), but is also used to enable scripting access to objects embedded in other applications.

Despite the name, JavaScript is only distantly related to the Java programming language, the main similarity being their common debt to the C syntax. Semantically, JavaScript has far more in common with the Self programming language.

JavaScript is a prototype-based scripting language with a syntax loosely based on C. Like C, the language has no input or output constructs of its own. Where C relies on standard I/O libraries, a JavaScript engine relies on a host environment into which it is embedded. There are many such host environment applications, of which web technologies are the best-known examples. These are examined first.

One major use of web-based JavaScript is to write functions that are embedded in or included from HTML pages and interact with the Document Object Model (DOM) of the page to perform tasks not possible in HTML alone. Some common examples of this usage follow:

- Opening or popping up a new window with programmatic control over the size, position and 'look' of the new window (i.e. whether or not the menus, toolbars etc are visible).
- Validation of web form input values to make sure that they will be accepted before they are submitted to the server.
- Changing images as the mouse cursor moves over them: This effect is often used to draw the user's attention to important links displayed as graphical elements.

The DOM interfaces in various browsers differ and don't always match the W3C DOM standards. Rather than write different variants of a JavaScript function for each of the many browsers in common use today, it is usually possible, by carefully following the W3C DOM Level 1 or 2 standards, to provide the required functionality in a standards-compliant way that most browsers will execute correctly. Care must always be taken to ensure that the web page degrades gracefully and so is still usable by any user who:

- has JavaScript execution disabled - for example as a security precaution
- has a browser that does not understand the JavaScript - for example on a PDA or mobile phone
- is visually or otherwise disabled and may be using an unusual browser, a speech browser or may have selected extreme text magnification

Other examples of JavaScript interacting with a web page's DOM have been called DHTML and SPA.

A different example of the use of JavaScript in web pages is to make calls to web and web-service servers after the page has loaded, depending upon user actions. These calls can obtain new information, which further JavaScript can merge with the existing page's DOM so that it is displayed. This is the basis of Ajax programming. PnP JavaScript design pattern was adopted gradually after commonly use of Ajax to reduce JavaScript maintenance cost.

Pros

- JavaScript is for the most part simple to learn and has some practical effects for the webmaster or web designer.
- JavaScript is client-side meaning they are executed by your browser (the client) and reduces the demand of the website's server.
- JavaScript is very fast because code's function runs immediately on the user's computer instead of having to send information to the server and back to the user for an answer.
- JavaScript language having a wide variety of neat tricks displays nicely with other languages like, PHP and can be used in a large assortment of applications.
- JavaScript can be inserted into any web page regardless of the file extension (ex: .html .php .shtml etc)

Cons

- Because of the wide-open nature of the Internet, security is an important issue. The JavaScript language executes immediately onto the users' computer and in some unfortunate cases it can be exploited for malicious purposes, like delivering viruses and adding advertising programs onto ones computer (adware). Which is one main reason that users tend to turn off JavaScript in their Internet options settings from running on their computers while surfing the net.
- Search engines can't read JavaScript navigation links no more than it can read text in an image. Some of them do read JavaScript on pages. It is, however, next to impossible for a search engine robot spider to follow links generated by JavaScript. What this means to you as a webmaster is that if you're relying on the SE robots to index other pages of your website based on following java links presented on your submitted page (homepage). It will most likely never find the other pages, due to your JavaScript navigation links. So, ultimately you would have to submit each individual page to the search engines. And if you have tons of pages this could take allot of time. You can create a sitemap page to be submitted to search engines also, which is a good idea anyways.
- JavaScript is sometimes read or understood differently with different browsers. When building your website and implementing JavaScript on your web pages. It is a good idea as standard procedure to test your web pages with several browsers for its functionality so you can make adjustments to clean up bad or needless use of scripts and other codes that look a little strange in your browser window.

5.6 DOM

The Document Object Model (DOM) is an application programming interface (API) for valid HTML and well-formed XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model, with a few exceptions - in particular, the DOM interfaces for the XML internal and external subsets have not yet been specified.

The DOM is a programming API for documents. It is based on an object structure that closely resembles the structure of the documents it models. For instance, consider this table, taken from an HTML document:

```
<TABLE>
<TBODY>
<TR>
<TD>Shady Grove</TD>
<TD>Aeolian</TD>
</TR>
<TR>
<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>
</TR>
</TBODY>
</TABLE>
```

A graphical representation of the DOM of the example table is:

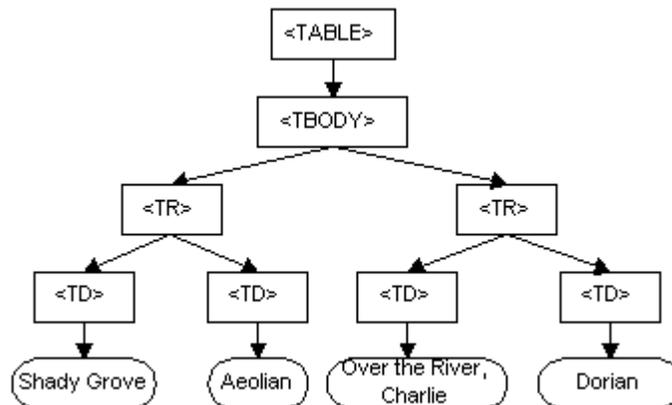


Figure 5.1

In the DOM, documents have a logical structure which is very much like a tree; to be more precise, which is like a "forest" or "grove", which can contain more than one tree. Each document contains zero or one doctype nodes, one root element node, and zero or more comments or processing instructions; the root element serves as the root of the element tree for the document. However, the DOM does not specify that documents must be implemented as a tree or a grove, nor does it specify how the relationships among objects be implemented. The DOM is a logical model that may be implemented in any convenient manner.

In this specification, we use the term structure model to describe the tree-like representation of a document. We also use the term "tree" when referring to the arrangement of those information items which can be reached by using "tree-walking" methods; (this does not include attributes). One important property of DOM structure models is structural isomorphism: if any two Document Object Model implementations are used to create a representation of the same document, they will create the same structure model, in accordance with the XML Information Set [Infoset].

The name "Document Object Model" was chosen because it is an "object model" in the traditional object oriented design sense: documents are modeled using objects, and the model encompasses not only the structure of a document, but also the behavior of a document and the objects of which it is composed. In other words, the nodes in the above diagram do not represent a data structure, they represent objects, which have functions and identity. As an object model, the DOM identifies:

- the interfaces and objects used to represent and manipulate a document
- the semantics of these interfaces and objects - including both behavior and attributes
- the relationships and collaborations among these interfaces and objects

5.7 AJAX

Ajax, shorthand for Asynchronous JavaScript and XML, is a web development technique for creating interactive web applications. The intent is to make web pages feel more responsive by exchanging small amounts of data with the server behind the scenes, so that the entire web page does not have to be reloaded each time the user requests a change. This is meant to increase the web page's interactivity, speed, and usability.

The Ajax technique uses a combination of:

- XHTML (or HTML) and CSS, for marking up and styling information.
- The DOM accessed with a client-side scripting language, especially ECMAScript implementations such as JavaScript and JScript, to dynamically display and interact with the information presented.
- The XMLHttpRequest object is used to exchange data asynchronously with the web server. In some Ajax frameworks and in certain situations, an IFrame object is used instead of the XMLHttpRequest object to exchange data with the web server, and in other implementations, dynamically added <script> tags may be used.
- XML is sometimes used as the format for transferring data between the server and client, although any format will work, including preformatted HTML, plain text, JSON and even EBML. These files may be created dynamically by some form of server-side scripting.

Pros

- **Bandwidth utilization:** by generating the HTML locally within the browser, and only bringing down JavaScript calls and the actual data, Ajax web pages can appear to load relatively quickly since the payload coming down is much smaller in size. An example of this technique is a large result set where multiple pages of data exist. With Ajax, the HTML of the page, e.g., a table control and related TD and TR tags can be produced locally in the browser and not brought down with the first page of the document. In addition to "load on demand" of contents, some web based applications load stubs of event handlers and then load the functions on the fly. This technique significantly cuts down the bandwidth consumption for web applications that have complex logic and functionality.

- **User interface:** the most obvious reason for using Ajax is an improvement to the user experience. Pages using Ajax behave more like a standalone application than a typical web page. Clicking on links that cause the entire page to refresh feels like a "heavy" operation. With Ajax, the page often can be updated dynamically, allowing a faster response to the user's interaction. While the full potential of Ajax has yet to be determined, some believe it will prove to be an important technology, helping making the web even more interactive and popular than it currently is.
- **Separation of Data, Format, Style, and Function:** a less specific benefit of the AJAX approach is that it tends to encourage programmers to clearly separate the methods and formats used for the different aspects of information delivery via the web. Although AJAX can appear to be a jumble of languages and techniques, and programmers are free to adopt and adapt whatever works for them, they are generally propelled by the development motif itself to adopt separation between: (1) the raw data or content to be delivered - which is normally embedded in XML and sometimes derived from a server-side database; (2) the format or structure of the webpage - which is almost always built in HTML (or better, XHTML) and is then reflected and made available to dynamic manipulation in the DOM; (3) the style elements of the webpage - everything from fonts to picture placement - are derived by reference to embedded or referenced CSS; and (4) the functionality of the web page is provided by a combination of (A) JavaScript on the client browser (also called DHTML), (B) Standard HTTP and XMLHttpRequest for client-to-server communication, and (C) server-side scripting and/or programs utilizing any suitable language preferred by the programmer to receive the client's specific requests and respond appropriately.

Cons

- **Browser integration:** the dynamically created page does not register itself with the browser history engine, so triggering the "Back" function of the users' browser might not bring the desired result. Developers have implemented various solutions to this problem. These solutions can involve using invisible IFRAMEs to invoke changes that populate the history used by a browser's back button. Google Maps, for example, performs searches in an invisible IFRAME and then pulls results back into an element on the visible web page. The World Wide Web Consortium (W3C) did not include an IFrame element in its XHTML 1.1 Recommendation; the Consortium recommends the object element instead. Another issue is that dynamic web page updates make it difficult for a user to bookmark a particular state of the application. Solutions to this problem exist, many of which use the URL fragment

identifier to keep track of, and allow users to return to, the application in a given state. This is possible because many browsers allow JavaScript to update the fragment identifier of the URL dynamically, so that Ajax applications can maintain it as the user changes the application's state. This solution also improves back-button support. It is, however, not a complete solution.

- **Response-time concerns:** network latency — or the interval between user request and server response — needs to be considered carefully during Ajax development. Without clear feedback to the user, smart preloading of data and proper handling of the XMLHttpRequest object, users might experience delay in the interface of the web application, something which users might not expect or understand. Additionally, when an entire page is rendered there is a brief moment of re-adjustment for the eye when the content changes. The lack of this re-adjustment with smaller portions of the screen changing makes the latency more apparent. The use of visual feedback (such as throbbers) to alert the user of background activity and/or preloading of content and data are often suggested solutions to these latency issues. In general the potential impact of latency has not been "solved" by any of the open source Ajax toolkits and frameworks available today, such as the effect of latency variance over time.
- **Search Engine Optimization:** websites that use Ajax to load data which should be indexed by search engines must be careful to provide equivalent data at a public, linked URL and in a format that the search engine can read, as search engines do not generally execute the JavaScript code required for Ajax functionality. This problem is not specific to Ajax, as the same issue occurs with sites that provide dynamic data as a full-page refresh in response to, e.g., a form submit (the general problem is sometimes called the hidden web).
- **JavaScript reliability:** Ajax relies on JavaScript, which may be implemented differently by different browsers or versions of a particular browser. Because of this, sites that use JavaScript may need to be tested in multiple browsers to check for compatibility issues. It's not uncommon to see a JavaScript code written twice, a part for IE, a part for Mozilla compatibles.

6 Usability

Usability addresses the relationship between tools and their users. In order for a tool to be effective, it must allow intended users to accomplish their tasks in the best way possible. The same principle applies to computers, websites, and other software. In order for these systems to work, their users must be able to employ them effectively.

6.1 What makes a website or piece of software usable?

Usability depends on a number of factors including how well the functionality fits user needs, how well the flow through the application fits user tasks, and how well the response of the application fits user expectations. We can learn to be better user interface designers by learning design principles and design guidelines. But even the most insightful designer can only create a highly-usable system through a process that involves getting information from people who actually use the system.

Usability is the quality of a system that makes it easy to learn, easy to use, easy to remember, error tolerant, and subjectively pleasing.

6.2 Why is Usability Important?

From the user's perspective usability is important because it can make the difference between performing a task accurately and completely or not, and enjoying the process or being frustrated. From the developer's perspective usability is important because it can mean the difference between the success and failure of a system. From a management point of view, software with poor usability can reduce the productivity of the workforce to a level of performance worse than without the system. In all cases, lack of usability can cost time and effort, and can greatly determine the success or failure of a system. Given a choice, people will tend to buy systems that are more user-friendly.

6.3 How Do You Achieve a High Level of Usability?

The key principle for maximizing usability is to employ iterative design, which progressively refines the design through evaluation from the early stages of design. The evaluation steps enable the designers and developers to incorporate user and client feedback until the system reaches an acceptable level of usability.

The preferred method for ensuring usability is to test actual users on a working system. Achieving a high level of usability requires focusing design efforts on the intended end-user of the system. There are many ways to determine who the primary users are, how they work, and what tasks they must accomplish. However, clients' schedules and budgets can sometimes prevent this ideal approach. Some alternative methods include user testing on system prototypes, a usability inspection conducted by experts, and cognitive modeling.

6.4 Usability tips

1. **Design a clear and simple navigation system.** According to Web usability expert, Jakob Nielsen, a good navigation system should answer three questions:
 1. Where am I?
 2. Where have I been?
 3. Where can I go?

Your site's navigation system will answer all three questions if you're careful to include these basic elements:

- **Keep it consistent.** The navigation system should be in the same place on every page and have the same format. Visitors will get confused and frustrated if links appear and disappear unpredictably. Consider using Server Side Includes for your main navigation system to make certain the navigation system stays consistent.
- **Use appropriate text inside links.** Don't make your visitors guess where a link is going to take them. Visitors should be able to anticipate a link's destination by reading the text in the link or on the navigation button. This isn't the time to be cute or obscure - visitors don't have

the time or patience for it. If there's any question about a link's destination, clarify the issue with a TITLE attribute that explains exactly where the link goes.

- **Use CSS to emphasize text links.** Some designers dislike underlined text links inside page content - although visitors expect to be able to click on underlined text. If you decide to remove this important visual navigation clue, style your links with CSS to replace underlining with another, consistent visual technique like a background color, different font, or text color that indicates a hyperlink.
- **Always include text links.** You can create some great looking menus using JavaScript or other scripting language, but never rely completely on a dynamic menu system. Some users may have problems using a mouse to navigate through the menu and others may be listening to the page using a screen reader. Every page should have basic text links that link to all major sections of the site.
- **Add a text-based site map.** Large or complex sites should always have a text-based site map in addition to text links. Every page should contain a text link to the site map. Lost visitors will use it to find their way, while search engines spiders will have reliable access to all your pages.
- **Include a home page link inside your main navigation system.** Visitors may enter your site via an internal page, but hopefully they'll want to head for the home page next.
- **Site logo links to home page.** Most sites include their logo somewhere at the top of every page - generally in the top, left-hand corner. Visitors expect this logo to be a link to your site's home page. They'll often go there before looking for the home link in the navigation system.
- **Include a site search box.** A robust site search feature helps visitors quickly locate the information they want. Make the search box prominent and be sure that it searches all of your site - and only your site. We've run across far too many Web sites that include a "Search the Web" search box on their home page. The result? Visitors hardly get to the site before the search function sends them to another site!

2. **Keep the content clear and simple.** You may attract visitors with an eye-catching design, but content is what keeps them at the site and encourages them to return. Content is also the best way to boost your site in search engine rankings.

Always keep search engines in mind when you write content, but remember that your ultimate audience is human visitors. Present your content with humans in mind.

- **Don't save the best for last.** Place your most important content high on the page. Think of a newspaper: the top story is always prominently displayed above the fold. Check your page display at in a number of different screen resolutions to make sure that your most important content is visible when the page loads.
- **Make page content easy to scan.** You'll spend hours - maybe days - writing your page content and it's really annoying to think that visitors may read less than half of it. Format your content so that it's easy to scan. Emphasize important points (or product characteristics) with a combination of header tags, bold type, color, or lists.
- **Avoid using text inside images whenever possible.** Text in images is invisible to search engine spiders and to visitors who may have images turned off in their browsers or who use assistive technologies like screen readers.
- **Add ALT and TITLE attributes to all images.** Each image should have a descriptive ALT attribute and TITLE attribute associated with it - particularly images that are also links to other pages. That way, they can quickly jump to the page they're interested in without having to wait for the entire page to load.
- **Contrast, contrast, contrast!** Be careful with background images and colors because they can obscure the text content on the page. Make sure you have a good reason to deviate from the successful dark text on a light background model. Visitors can't buy your products if they can't read the content.

3. **Support your brand.** A good brand creates or reinforces a user's impression of the site. When your site is strongly branded, that means that visitors will think of you first when they go shopping for your product or service.

Branding on a Web site takes time, effort, and close attention to page design and layout.

- **Keep colors and typefaces consistent.** Visitors should never click on an internal link in your site and wonder if they've left your Web site. Choose your colors and fonts carefully and use them consistently throughout the site.
- **Keep page layout consistent.** Use a Web site template to enforce a uniform page structure. Visitors should be able to predict the location of important page elements after visiting just one page in your site.
- **Custom error page.** Create a useful custom error page that helps visitors if they should click on a broken internal link or type a URL incorrectly. The custom error page should reflect the site's overall color, type, and layout structure as much as possible and provide useful links to help visitors find what they're looking for.
- **Create a good tagline and use it on every page.** A good tagline clearly and concisely explains your "value proposition" or what makes your site stand out from competing sites. It should be memorable and reinforce your brand in one quick phrase.

7 Implementation phases

7.1 Design

Page was designed in Adobe Photoshop application. 4 colors are present:

- Black – standard color for text
- White – background color. With the combination of black content provides good contrast making the text easy to read
- Grey – mainly used to visually separate hierarchy of the site
- Blue – last of the cool colors used in order to enrich the look

The design was created as a compromise between functionality and look. Very few graphic images have been used which should lead to boosting the speed and minimize data transfer between server and client. On the other hand I tried to make the user interface appealing.

7.2 XHTML templates

Once the design was finalized it was necessary to transform it from a .PSD file into a XHTML document.

Width of the main frame was set to 760px and therefore made the layout fixed. Most of the internet users have screen resolution at least 800x600 which allows them to see the content without having to scroll vertically.

Using CSS style sheet the visual aspect has been successfully separated from the functional side. This makes the code clearer, easier to understand and navigate through. Several classes and identifications were defined making the look of all pages consistent.

7.3 MySQL database and sample data

PHP scripting usually gathers, stores and modifies data in a database. Therefore it was important to create a database with tables corresponding to entities specified in ER diagram first.

Users

Table stores information of all actors in the system. Each user must have a unique login that is used during for authorization process followed by a password that is encrypted using MySQL's PASSWORD function. Personal information like name, surname and contact (email, telephone, icq, msn number) are present as well.

```
CREATE TABLE users (  
    id_user MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    login VARCHAR(30) NOT NULL,  
    password VARCHAR(20) NOT NULL,  
    name VARCHAR(30),  
    surname VARCHAR(30),  
    email VARCHAR(40) NOT NULL,  
    tel1 VARCHAR (30),  
    tel2 VARCHAR (30),  
    icq VARCHAR (9),  
    msn VARCHAR (40),  
    PRIMARY KEY (id_user)  
);
```

Advertisements

This table contains main data described by a header, content, price, region, date of creation. Each ad is created by some user and belongs to certain category.

```
CREATE TABLE advertisements (  
    id_ad MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    category SMALLINT UNSIGNED NOT NULL,  
    heading VARCHAR(40) NOT NULL,  
    text MEDIUMTEXT NOT NULL,  
    price VARCHAR(30) NOT NULL,  
    region SMALLINT UNSIGNED NOT NULL,  
    date TIMESTAMP NOT NULL,  
    id_user MEDIUMINT UNSIGNED NOT NULL,  
    PRIMARY KEY (id_ad));
```

Categories

Filled with 635 rows it represents a tree structure of categories. Each entry has a parent category which is used to track the path back to the root node and also to easily list ads from subcategories. Only a node without any child can make a connection to an advertisement and is represented by 5 digits – ending with 0 mean an offer, ending with 1 means a demand. The system has been created quite fast using Excel and formulas.

```
CREATE TABLE categories (  
    id_category SMALLINT NOT NULL AUTO_INCREMENT,  
    name VARCHAR(30) NOT NULL,  
    parent SMALLINT,  
    PRIMARY KEY (id_category)  
);
```

Messages

Hold all the private messages registered user may send among each others. A sender submits a form to a specific recipient including a subject and body. The date is set automatically using MySQL's function NOW(). By default the status of a message is set to 0 meaning it hasn't been read yet. Once the recipient goes to Inbox all messages with status 0 will be updated to status 1 – read message.

```
CREATE TABLE messages (  
    id_message MEDIUMINT NOT NULL AUTO_INCREMENT,  
    subject VARCHAR (40) NOT NULL,  
    text MEDIUMTEXT NOT NULL,  
    date TIMESTAMP NOT NULL,  
    status SMALLINT UNSIGNED NOT NULL,  
    id_sender MEDIUMINT UNSIGNED NOT NULL,  
    id_recipient MEDIUMINT UNSIGNED NOT NULL,  
    PRIMARY KEY (id_message)  
);
```

Comments

Every comment stores a header, main text and was sent by a specific user in a discussion thread to a specific advertisement. Again, the date of creation is automatically generated by NOW() function.

```
CREATE TABLE comments (  
    id_comment MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    heading VARCHAR(40) NOT NULL,  
    text MEDIUMTEXT NOT NULL,  
    date TIMESTAMP NOT NULL,  
    id_ad MEDIUMINT UNSIGNED NOT NULL,  
    id_user MEDIUMINT UNSIGNED NOT NULL,  
    PRIMARY KEY (id_comment)  
);
```

Settings

User has the ability to subscribe to an automated process and receive new ads from a chosen category into private mailbox (mail information stored in Users table) on a regular basis. One user can watch only one category at once. The type field has 4 values:

Daily

Every other day

Twice a week

Weekly

```
CREATE TABLE settings (  
    id_setting MEDIUMINT NOT NULL AUTO_INCREMENT,  
    category SMALLINT UNSIGNED NOT NULL,  
    type SMALLINT UNSIGNED NOT NULL,  
    id_user MEDIUMINT UNSIGNED NOT NULL,  
    PRIMARY KEY (id_setting)  
);
```

Photos

Last table is used to store location of photos uploaded to a server and copied to photos folder. For each advertisement only one photo is allowed.

```
CREATE TABLE photos (  
    id_photo MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    location VARCHAR(100) NOT NULL,  
    id_ad MEDIUMINT UNSIGNED NOT NULL,  
    PRIMARY KEY (id_photo)  
);
```

7.4 Scripts

Add.php

The script can be divided into 2 nodes:

Showing a form to enter a new advertisement. This will be shown in case a user enters the page for the first time or in case of any reason the advertisement couldn't be added (problem with DB, invalid input data etc.). The list of categories which a user can add his ad to is generated automatically from the DB.

Processing input data. First the input is validated, empty fields are being checked. In case of an error a variable \$error_msg is populated with information and \$error variable is set to TRUE. If there was no error a connection to DB is made and the advertisement gets inserted. If a photo was sent along with the advertisement, it is copied into photo folder and renamed with the ID of the new advertisement for easier processing.

If the user hasn't been logged yet the advertisement will be added under anonymous user, who had ID 7 in the database.

Db.php

Contains defined constants holding the critical information needed to connect to a MySQL database. A connection and a database choose is attempted and error messages appear in case of a failure.

Discussion.php

This script allows users to post their comments for each advertisement in the system. The text of a specific ad is shown at the top. Below previous comments from users are being displayed, followed by a form to enter a new comment. In case of a success new comment is directly displayed. The only return information shown is in case of fields missing.

Include.php + footer.php

Basic scripts generating templates included in all pages. Include.php produces utilities menu (Login, Register, Messages etc.) based on the current user status. Left menu is also created in this script showing local navigation of selected category.

Footer.php completes the structure of a valid XHTML document and is always embedded in calling scripts at the end.

Index.php

Home page that welcomes every visitor. Because the home page is a specific category with a NULL parent all advertisements in the system are displayed here. For better orientation only 5 ads are displayed per page, remaining pages act as a link at the bottom.

The structure of advertisement is as following:

Title in bold and a date of creation are at the top on a silver background

Text itself follows, intended from the left side

Ad ends with a line of Price, Photo if there's any and number of comments that serves as a link to discussion

Login.php

Contains 2 fields necessary to authorize a user: login and password. Once the login process has been finished a user SESSION variable is set containing user's ID. This is preserved in the system unless a logout script is called

Logout.php

As wrote above this file destroys the SESSION variable and therefore transforms a registered user into a non-registered one.

Msg.php

Script offering a messaging service among registered users. According to the action GET variable user can:

Compose a new message to another user. This requires recipients' login, subject and a text

Read messages received in inbox

See sent and deleted messages

When sending a message a new row is inserted in a database with a state set to 0 meaning the message has not been read. After clicking Inbox and reading the email the status is changed to 1.

Refresh.js + refresh_msg.php

Implementation of AJAX technology is present in these 2 scripts.

JavaScript file defines 2 functions refresh and process. Process function is called from within the refresh function after a state change was received. In case of a success and new messages the process function changes Messages link under utilities menu showing the number of new messages in bold.

Refresh function transfers data to PHP script that in return delivers number of new messages.

The process of communication between JS and PHP files takes place every 30 seconds. Thanks to AJAX technology a registered user can be easily informed of a new received message without having to reload page manually.

Register.php

Form enabling a non-registered to user to register himself. Some of the input fields are necessary (for example login, password) and must be filled in, otherwise the user cannot proceed.

When a new user is being entered into a db, his password is encrypted with the help of MySQL PASSWORD function.

Search.php

Provides a very basic search engine that looks for specified text in a title or in the text field. If at least one advertisement is found it will be displayed the same way as on the homepage. In this case the pages are not listed.

Settings.php

Here a user can change his personal data entered during registration. Additionally one can subscribe to receive new ads via email on a regular basis.

8 Installation and requirements

Advertising information system created can run on any web server with database server MySQL and PHP server installed. It was created and tested on Apache server 1.3.33, MySQL 4.1.9 and PHP 4.3.10 which come in a self-installation package EasyPHP 1.8.

Settings of the servers were left as default. The only thing to mention is that PHP setting `register_globals` should be set to Off for higher security.

Server applications mentioned above can run on a great variety of operating systems. These combinations have well-known acronyms, for instance WAMP which I used refers to Windows-Apache-MySQL-PHP. Other popular are: LAMP (Linux), MAMP (Macintosh), NAMP (Novel NetWare), BAMP (BSD) etc.

8.1 1st steps

1. Copy folder structure with all script into a root folder on a web server
2. Make the photo folder accessible for everyone to be able to upload pictures. This can be done by modifying access right on a server by command `CHMOD 777`
3. Execute SQL scripts on MySQL server. This will create category structure and also add `anonymous_user` with ID 7 that is used to represent non-registered users.
4. Modify `db.php` under include folder to reflect the needs of your MySQL server. The `user-password-database host-database` must be changed.

9 Summary

The goal of this thesis was to create a functional on-line advertising information system. The system as is is ready to provide basic tasks for end users. Unlike other advertising systems in Czech Republic it was being developed with minimum graphic elements that would slow the transfer from server to client, however with effort to keep the design appealing and easy to use. For these reasons I think the application has a chance among end users.

In order to evaluate the usability level I've asked some of my colleagues and friends to try out the site without previous information. The feedback given to me was very positive and almost everyone managed to perform all basic tasks successfully.

Of course the system has some defects at the moment and there's still a lot to do to make it better. I'd like to continue working on it in the future and improve following areas: convert procedural scripts into object-oriented, implement functions for administrator and enwide possibilities for users (RSS, design customization), increase the security and input data validation, add advanced searching and custom sorting of advertisements.

A lot of time was dedicated to study technologies. I believe it was a time well spent and the knowledge gained will help me find a good job in web industry. Additionally I have deepened my knowledge from programming and IS modeling.

10 References

- [1] *Information system Definition* [online]. URL: <<http://www.techweb.com/encyclopedia/defineterm.jhtml?term=information%20system&x=&y=>> [cit. 2007-1-11].
- [2] *About PHP* [online]. URL: <<http://www.zend.com/zend/aboutphp.php>> [cit. 2007-1-5].
- [3] *What is MySQL?* [online]. URL: <<http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html>> [cit. 2007-1-5].
- [4] *XHTML* [online]. URL: <<http://en.wikipedia.org/wiki/XHTML>> [cit. 2007-1-5].
- [5] *Introduction to CSS* [online]. <URL: http://www.w3schools.com/css/css_intro.asp> [cit. 2007-1-5].
- [6] *JavaScript* [online]. URL: <<http://en.wikipedia.org/wiki/JavaScript>> [cit. 2007-1-5].
- [7] *What is the Document Object Model?* [online]. URL: <<http://www.w3.org/TR/DOM-Level-2-Core/introduction.html>> [cit. 2007-1-5].
- [8] *Ajax (programming)* [online]. URL: <[http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))> [cit. 2007-1-5].
- [9] *Introduction to Usability* [online]. URL: <<http://www.usabilityfirst.com/intro/index.txt>> [cit. 2007-1-6].
- [10] *Design Tip: Web Site Usability Checklist* [online]. URL: <http://www.netmechanic.com/news/vol7/design_no4.htm> [cit. 2007-1-6].
- [11] KOSEK, J. *Tvorba interaktivních internetových aplikací*. 1. vydání. Praha: Grada Publishing, 1998. ISBN 80-7169-373-1.
- [12] ULLMAN, L. *PHP a MySQL: názorný průvodce tvorbou dynamických WWW stránek*. 1. vydání. Praha: Computer Press, 2004. ISBN 80-251-0063-4.
- [13] CAMPBELL, M. *Web design garage*. 1st edition. Michigan: Pearson Education, 2005. ISBN 0-13-148199-1.
- [14] KRUG, S. *Web design: nenuťte uživatele přemýšlet!* 1. vydání. Brno: Computer Press, 2003. ISBN 80-7226-892-9.