Document Management and Exchange System: Supporting Education Process

doi:10.3991/ijet.v5s2.1227

E. Egredzija[1], B. Kovacic2
[1] Information Technology Institute, City of Rijeka, Croatia
[2] University of Rijeka, Rijeka, Croatia

***Abstract*—Development and implementation of new technologies are very important in education. One of the most challenging tasks in the education process is to build efficient and cost-friendly system for content management and exchange. The system has to be reliable, easy manageable and open. Centralized storage, secured access, and ubiquitous client technologies have emerged as best-practice solutions in engineering that kind of services. Users can easily publish or exchange documents and not need to worry about their distribution, storage or technical skills required for efficient document management. The system that will be presented is built on open source technologies and is deployable on all today's popular web software platforms. The web server, the programming language and operating system that are used to build and deploy such a system are all non-proprietary and completely open because our mission was to build system that can be easily extended and not limited by its corporate license. The system uses security mechanisms such as user group access policy, operating system level security (file system) and secured data storage in database.**

Because of the growing need for document management in education process we believe that this project will find its place in practice.

***Index Terms*—content management and exchange, document management, content distribution, open source technologies, content security**

I. INTRODUCTION

Process of education has been improved by newly developed and advanced technologies. Upgrading of communication technologies and especially application of computer in communication facilitated long distance data transfer. It favored the development of various distance learning techniques in which the advantage of electronic media allowed fast and relatively simple distribution of learning material to a great distance. Presently, the development of telecommunication and computer networks, especially of Internet, creates a powerful base for the transfer of vast multimedia information and the education independent of its venue and time [1]. Efficient and cost-friendly system for content management and exchange has crucial importance in educational process.

Basic principles of document management and exchange system are shown in Section II. Development platforms for document management and exchange system is shown in Section III. Further development of document management and exchange system is shown in Section IV. Security of such system is shown in Section V; conclusion in Section is followed by the reference list.

II. SYSTEM REQUIREMENTS

Document management and exchange system by its own purpose has to comply fully with these four main standards:

1. reliability
2. easy manageability
3. openness
4. secured access

A system to be reliable, it must be carefully constructed and design and development tools have to be matured and proved technologies. The data that is stored and managed by this system needs to be 100% available and the underlying technologies have to be fully complied with data storage and management standards. The software stack of the system must be carefully designed and thoroughly tested so that software management routines and client applications can always satisfy all of the organization's constituent request and demands.

The need for easy manageability in software industry is one of its core principles. Without the simplicity and cleanness in mind, software product can easily turn to become just the opposite of cost-effective solution that satisfies organizations needs. When discussing about manageability two aspects emerge: 1) manageability of software from customers point of view (system administration, content management, backup routines, etc.); and 2) manageability from software creator's view because the software needs to be maintained and updated in regular periods of time. So without the cleanly defined management routines, the system is likely to become extremely hard to maintain from both aspects.

Openness is defined as using open and standardized technologies in design and development process. When using such technologies, replacing software components, or extending software package using other open technologies, and engagement of other companies for development process comes with smaller price. Such standardized technologies can be open sourced or not, but with open source comes much greater and effective control.

When dealing with huge amount of data that are published and accessed from different geographical locations and different sort of users, centralized storage of data comes as pretty much only viable solution. All the data is stored on a single storage unit (operating system file system, network file system etc. [4,5]), and is accessed from

services that don't have to be on the same hardware of operating system stack. Centralized storage is much easier to maintain and security mechanisms are more effective.

Data exchange and communication in such a system needs to travel across secured channels and standardized group policy rules and user authentication have to be carefully designed. Administering the system or the "backend" part of document management is done through secured network connections, and secured database access. User queries for documents are done after the successful authentication and pairing user profile with its access privileges (even with anonymous users).

When developing client's side of software package, or in other words, when constructing part of the software product that will be used by end users, it is necessary to survey and conclude which technologies are common to most of potential constituents of software product. Today's such ubiquitous technology is web client software, or web browser applications. It is free and it is easy to learn because of its multi-purpose nature – if a user is capable of browsing the web then he is capable of using custom web application with little or no further education. Users don't have to worry about software installation procedures or content distribution. Their web browser is a presentation technology that gives user a window to service offered by some software product. All the inner logic (business logic) is deployed on centralized server systems and maintained by professionals.

III. DEVELOPMENT PLATFORMS

Guided by the four main standards which are detailed in section describing system requirements we easily came to cleaned-up version of list of the development platforms which are suitable for our system. More precisely, our selection was nothing but a compilation of available technologies. Because of the openness of these technologies, the choice could have been something different, but what is most important, it wouldn't have had any major impact on the final software product. In addition, these technologies are not solely used at design and development phase, but they are building block of the deployment platform. Because of that, the customer is not limited in any way with platform choice when deploying the software product. It is true that our system consist of two main parts: custom software application and software platform on which the application is deployed. The application depends on services of the underlying platform and cannot operate without it. But that platform, or some part of it, can quite easily be replaced with some other option. Once again, that was one of our main goals.

Figure 1 describes all the components of software platform on which the system is built and deployed. Every component can be replaced with some other software (or software package). For example, CPU (Central Processing Unit or the processor) architecture can become x64, operating system Solaris, MySQL database can be replaced with Oracle Express, Apache web server with IIS (Internet Information Services on Windows operating system), etc. The only part that cannot be replaced is the PHP scripting language because the whole custom software application is built with this technology. But, PHP applications can be deployed on all popular web servers and operating systems. Software application is not mentioned in Figure 1 because it is explained in detail in next section.

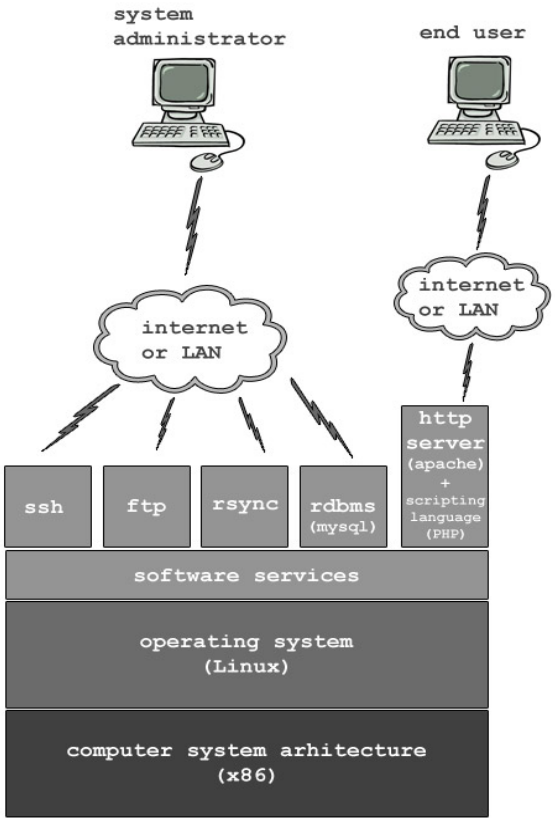

Figure 1. The complete software platform (system without custom software application)

The whole communication within system can be divided in two channels:

Communication between system and system administrator (or content administrator); it can be achieved through several services as seen in Figure 1. Depending of administrator needs or current administration job, secured SSH (Secure Shell) protocol connection can be used to operate directly on system, ftp protocol can be used for file transfer, RSYNC service (file and directory synchronization) can be used for data backup and synchronization, MySQL database client can be used for database administration etc.

Communication between system and end user: it is done through web client browser mediation. Web browser is nothing but presentation service that is displaying results of user query for system content.

It is worth mentioning that the technologies used in software platform are part of software stack called popularly by its acronym LAMP [2]. L stands for Linux operating system, A for Apache web server, M for MySQL RDBMS (Relational database management system) and P for PHP programming language. All the components of this software stack are one of the most prominent open source software solutions and are used by all major academic institutions, government institutions and corporations. Michael Kunze [2] in 1998 popularized the acronym in his articles from a German magazine for computer technology. They wanted to point out that a set of applications that make LAMP may represent a useful alternative

to commercial solutions. Today this software stack is on the forefront of technology solutions for serving dynamic web applications.

IV. DEVELOPMENT OF WEB APPLICATION

In this part of the work we describe the development of web applications that will use the services installed on a server [6]. The development will consist of 4 steps:

1) Design and development of database

The database is used because of flexibility of relational data model and maintenance of user profiles and user privileges data [3]. The relational model is simplified and consists of 2 logical parts: first part is consisted of tables that represent data defining users, user privileges, and available content; second part is for application "front-end" modules and it describes available modules. The database is designed using standard SQL (Structured Query Language) syntax so it can be exported and built with some other RDBMS software.

2) Development of core application and additional modules and components

The application consists of a core part, i.e. the code that initiates the application and sets the environment, and if needed, loads the available modules and components which are the second part. The third part consists of graphical user interface building blocks.

The whole application is designed with object oriented paradigm of software design [7]. Application modularity was one of the primary goals, so the OOP (object-oriented paradigm) was only viable choice. There are several main classes and routines that make the core of application:

2.1. Initialization routines that make the startup of the application. The main purpose of these modules are to define the initial context of the application and to make all of the necessary decisions, primarily which modules are going to be started and which GUI (graphical user interface) components are to be generated and displayed. For example, the initialization routines can decide whether the user is browsing the systems managed content categories or the user has made request for download of data. Also, these routines load all the necessary dynamic configuration data. It's dynamic because its content can be changed during run time.

2.2. The *frontend* class that is the central point of the front part of the web application. The purpose of this class is as follows:
- making connections to database server
- user authentication and definition of user privileges
- initializing the content modules
- generating the HTTP headers (Hypertext Transfer Protocol), initial HTML elements (Hyper Text Markup Language), HTML meta-data, constructing the main "body" part of the web application, initializing and displaying the graphical user interface

2.3. Application content modules and interface components. Modules are classes that define which content is displayed based on the user input, initialization routines and configuration data. Components are interface blocks that make the GUI part of the application. Components are called somewhere from the design templates and are initialized by the *frontend* class. They can be easily turned off or even deleted from system. Naturally, the OOP is used extensively and basically all of the system can be extended with new custom components and content modules.

2.4. Components and modules inherit some functionality of the main *frontend* class so the globally defined routines (application level) can be used by all relevant building blocks of the application (the component or the module level).

3) System administration and content management routines

For system to function normally and effectively it is necessary to install and build the administration tools that are both easy to use and are robust enough to allow fine grained control over the entire system. The system administration can be divided in these tasks:

- Administration at operating system level
- Administration of the web server
- Administration of database server
- Content management

We will explain the content management routines because they are custom made procedures built solely for this system operation. Content management is performed in two ways: by invoking *arhiva* class administration script in the administration directory and/or by inserting/editing data in the database using the appropriate tools.

This calling of the *arhiva* class is done through execution of the PHP script in the UNIX shell (or Windows command prompt) and the results will show in plain text format. This class represents the very essence of the system content management because its main purpose is searching the whole archive and refreshing and/or inserting the content in the database. The database can be seen here as the extension of archive located on the file system: all the available content is documented and described in the database. We couldn’t rely solely on the file system features in the content management process because the system would suffer from performance degradation.

The content management routine searches all the available data stored on the file system at the location that represents physical storage of the system content archive. The results of the search are compared with content stored in the database: existing content is edited if needed and newly added content is documented in the database so the system can locate it more efficiently. At the same time, the user privileges for the content are imposed based on the patterns defined in advanced. Patterns are characters prefixed or suffixed on the directory or file names. Using the patterns adds to the simplicity and efficiency in administration because all the relevant UNIX (or Windows) tools can be used for managing directory and file names.

4) The development and design of application user interface (GUI)

After the skeleton of the application is constructed, the whole interface needs to be designed and all the components are integrated in the single layout.

The technologies used for GUI are standards for web application interface design: HTML, CSS (Cascading Style Sheets), Javascript. All the components are glued together using design templates so newly created design can be easily imported in the system. Web application homepage is shown in Figure 2, and content information and download link are presented in Figure 3.

Figure 2. Web application homepage

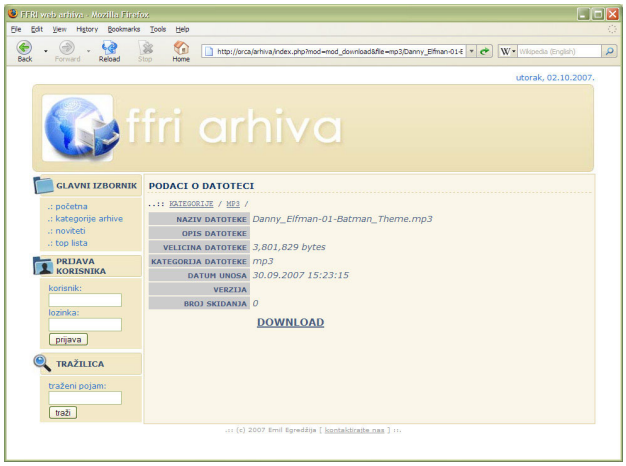

Figure 3. Content information and download of file

V. SECURITY

The system was designed and developed with all the security standards in mind. Security is implemented at various levels in system:

- The physical archive data is stored on the secure file system using user privilege policy
- The network access to the content management system is only allowed through secured communication protocols
- Web server is secured by carefully planned configuration and content isolation
- Database connections are only allowed from the local machine directly, or from another machine through web interfaces (no direct remote access exists)
- User privileges are defined for all the managed content in system
- All non-anonymous user must pass the user authentication process
- The user passwords are stored using cryptographic hash functions.

VI. CONCLUSION

Modern education processes demand the use of efficient and easily administered systems for content management and exchange. We wanted to build the system that can easily be deployed and successfully used in education organizations or other institutions. System for content management and exchange was built on open source technologies and is deployable on all today's popular web software platforms and can be easily extended and not limited by its corporate license. The security in such system has a great importance. Security mechanisms are implemented by user group access policy, operating system level security (file system) and secured data storage in database.

Document management and data exchange has a huge importance in every learning management system (LMS). The system for content management and exchange can be incorporated in every LMS as valuable service and we believe that such system will find its place in practice.

REFERENCES

[1] B. Kovacic, “Application of dialogue realized by distance learning system based on dialogue”, *Proceedings of the International Convention MIPRO 2003*, section CE, Rijeka, Croatia, 2003.

[2] M. Kunze, “LAMP: Freeware Web Publishing System with Database Support”, (http://www.heise.de/ct/english/98/12/230/), 1998.god.

[3] V. Vianu, R.Hull, S. Abiteboul, “Foundation of databases”, Addison Wesley, 1995.

[4] Silberschatz, A., Galvin, P.B., Gagne, G.: “Operating system concepts, 7ed”, John Wiley & Sons, 2004.

[5] Tanenbaum, A.S.; Woodhull, A.S.: “Operating Systems. Design and Implementation, 3ed”, Prentice Hall Software Series, 2006.

[6] S. A. Gabarro, “Web Application Design and Implementation: Apache 2, PHP5, MySQL, JavaScript, and Linux/UNIX”, John Wiley & Sons, 2007.

[7] http://www.php.net

AUTHORS

E. Egredzija is with Information Technology Institute, Information system development division, City of Rijeka, Korzo 16, Rijeka 51000, Croatia, (e-mail: emil.egredzija@rijeka.hr)

B.Kovacic is with University of Rijeka, Department of Informatics, Omladinska 14, Rijeka 51000, Croatia, (e-mail: bkovacic@inf.uniri.hr)

This article was modified from a presentation at The 32nd International Convention MIPRO on information and communication technology, electronics and microelectronics., in 25-29 May 2009, Opatija, Croatia. Manuscript received, 12 February 2010. Published as resubmitted by the authors 1st March 2010.