

Usability improvements on a Metadata Server for Video on Demand based on Free Software.

Diploma thesis

By

De Vos Laurens

Submitted by

Giró Xavier



Departament de Teoria del Senyal i Comunicacions
Universitat Politècnica de Catalunya

July 16, 2009

Abstract

My name is Laurens De Vos and I am an Belgium Student, coming from KAHO Sint-Lieven Rabot of Ghent. This final project encompasses the design of a Video on Demand system and is created in the laboratory of the Department of Signal Theory and Communications on the University of Industrial Engineering of Terrassa (EUETIT)

An first version has already been released by David Vera. Unfortunately there have been missing some services in his project. The final system must provide a management interface for the system administrator to easily upload videos and edit their metadata, and also the possibility to consume the video content from a diversity of user interfaces, such as a web browser, a Set Top Box or, in the future, a mobile device such as an iPhone.

After a thorough study of the structure of an video file the different video definition is not an issue anymore. Once the program, FFmpeg, has been installed and an specific codec selection has been made the generation of an video can be done by using the right code. This technology can be explored further for a thumbnail generation.

Subsequently, a validation in terms of both integration with existing standards and technical performance has been realized, fulfilling the most of the initial expectations.

Content

Chapter 1: Introduction

Chapter 2: State of the Art

| | |
|---|-----------|
| 1. Applications..... | 8 |
| 1.1. Video on Demand (VoD) | 9 |
| 1.2. Live Streaming (broadcast) | 12 |
| 2. Server..... | 14 |
| 2.1 MySQL..... | 14 |
| 2.2 Digital Asset Management (DAM) | 15 |
| 2.2.1 Akamai | 16 |
| 2.2.2 Fedora Commons..... | 16 |
| 2.3 Server languages | 17 |
| 3 Transmission..... | 18 |
| 3.1 Coding – Formats | 18 |
| 3.1.1 Video codec..... | 19 |
| 3.1.2 Audio codec | 20 |
| 3.1.3 Containers..... | 22 |
| 3.2 Metadata | 25 |
| 3.2.1 MPEG-7..... | 25 |
| 3.2.2 MPEG-21..... | 25 |
| 3.2.3 Dublin Core..... | 26 |
| 3.3 Streaming | 26 |
| 3.3.1 RTSP..... | 27 |
| 3.3.2 IGMP..... | 27 |
| 4 Client..... | 28 |
| 4.1 Devices..... | 28 |
| 4.1.1 Mobile..... | 28 |
| 4.1.2 PC s..... | 29 |
| 4.1.3 Set Top Boxes & TVs..... | 31 |
| 4.2 Client languages..... | 32 |

chapter 4: Installation

| | |
|---|-----------|
| 1. Server Software | 37 |
| 1.1 XAMPP | 37 |
| 1.2 FFmpeg..... | 39 |
| 2. Client Software | 41 |
| 2.1 Mozilla Firefox | 41 |
| 2.2 VLC | 42 |
| 2.3 VLC plug-in..... | 42 |
| 3. Configuration of Xampp | 43 |
| 3.1 MySQL database | 44 |
| 3.1.1 MySQL users | 44 |
| 3.1.2 Creation & management of the video database | 46 |
| 3.2 Automatic start of Xampp/Lampp | 51 |
| 3.3 Apache and PHP | 52 |
| 3.3.1 Apache..... | 52 |
| 3.3.2 PHP | 53 |

Chapter 5: Development

| | |
|--|-----------|
| 1. Upload and delete video. | 66 |
| 1.1 Upload video..... | 66 |
| 1.2 Delete video | 68 |
| 2. Add and Erase Topic | 69 |
| 2.1 Add Topic..... | 70 |
| 2.2 Erase Topic | 71 |
| 3. Improvements on design. | 71 |

Chapter 6: Future Work and Conclusion

| | |
|------------------------------|-----------|
| 1. Future Work. | 73 |
| 2. Conclusion. | 74 |

Chapter A: Appendix - Documentation

| | |
|---|-----------|
| A.1 Visitors..... | 76 |
| A.1.1 Navigation..... | 76 |
| A.1.2 Video on demand..... | 79 |
| A.2 Managers / Administrator | 81 |
| A.2.1 Upload video..... | 81 |
| A.2.2 Delete video | 83 |
| A.2.3 Add new Topic | 83 |
| A.2.4 Delete Topic..... | 84 |

Chapter B: Appendix - Database Figures

Chapter C: Appendix - List of Figures

Chapter D: Appendix - Bibliography

Chapter E: Appendix - Acronyms

Chapter 1

Introduction

This project was carried out in laboratory of the Department of Signal Theory and Communications (TSC) on the University of Industrial Engineering of Terrassa (EUETIT) during the period February 2009 to July 2009. This thesis project was supervised by Professor Xavier Giró.

To get a better look at this thesis is to be know that the Department of Signal Theory and Communications (TSC) of EUETIT school recently acquired a HD IPTV receiver. This receiver is capable of requesting contents from a video server accessed through an IP network. The goal of this thesis is to update an existing video on demand server designed for web browsing to provide service to the mentioned IPTV Set Top Box (STB). The existing system was implemented in a previous bachelor thesis by David Vera during Fall semester 2007. During the system enhancement, I have worked together with my colleague Alexander Schmidt, dividing the tasks in two topics. The first topic focuses on the client side of STB, while this thesis is based on improving the existing video server with new capabilities that improve its usability and provides access to the database through a STB.

The following graphic gives you an complete view of the two thesis, both at the client and at the server side.

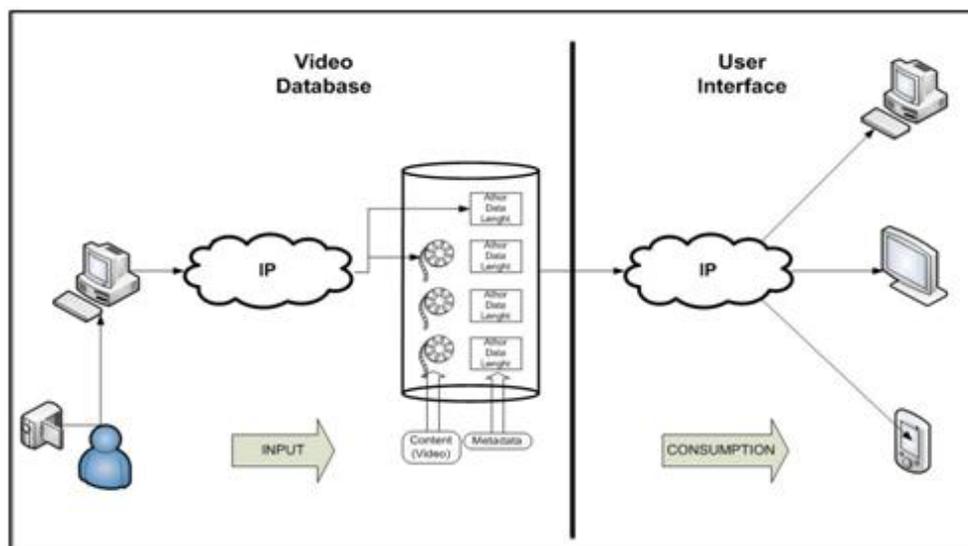


Figure 1: Complete view of Video on Demand.

The study case for the project focuses on video on demand service for the contents generated by the TSC department at the EUETIT school. The final system must provide a management interface for the system administrator to easily upload videos and edit their metadata, and also the possibility to consume the video content from a diversity of user interfaces, such as a web browser, a STB or, in the future, a mobile device such as an iPhone.

My part of the project, focused on the video server, consists of simplifying and improving the ingest of new content on the database by launching a set of video processing algorithms that adapt the content to the different types of interfaces. The basic idea is to offer scalability to the video service by automatically generating different versions of the content adapted to the consumption device. Three cases are defined: high definition (large screens), standard definition (traditional TV displays) and mobile devices (small screens). The improvements also generate automatically the thumbnails from a video that allow a more intuitive navigation through the archive. Important to know is that this project had to be based on free software. Means that every person is able to install each program that will be used in this project.

Another goal of this project is to clean up, comment and document the previous version of the PHP code developed by David Vera. It was detected that the code structure was too messy and inefficient for a further development. This thesis proposes a clear structure of the code based on modularity that must allow its growth in future thesis and related projects.

I would like to thank various peoples who helped me during my project. Albert Márquez, administrator of the TSC lab at the EUETIT, who helped me to start David Vera's project; Xavier Giró for his dedication to this project and of course I want to thank my colleague Alexander Schmidt for his friendship and for the beautiful cooperation.

Chapter 2

State of the Art

First of all it is necessary to make a reference to David Vera's¹ thesis for the State of the Art. His chapter , State of the Art, is structured so that each section explains the technology that is been used currently in the process of each element for the video on demand system. In this thesis, the State of the Art consists an explanation of each different format that can be used for a video transmission. Also each different language codes and programs that has been used in this project.

1. Applications

The last few years streaming has been evolved so much that he became one of a home media. The streaming video will not only be used to play online. There are other ways to take advantage of its features that facilitate or enhance the work in different areas.

If we talk about services you can see it as a functionality which receives video content from producers and other sources. Afterwards, these contents are encoded and, depending on which distribution method is going to be used, either broadcast it or store it in an acquisition database for Video on Demand.

It has become fashionable to pages where you can view videos of all types on demand. Pages where anyone can upload any video for later play on the net or playing from the server were a user can select a video clip that is in their archive.

i.e. there is a database where the content is stored on a disk and then we have also a video streaming server as well as a web server that provides an interface for administration.

When we have a database where the content is stored on a disk you can see that David Vera's project is based on this way. We can call this also "Video on Demand".

To explain the both –Video on Demand – Live Streaming- we will take each of them under the loop.

¹ Tesis David Vera: Servidor Web de Video Sota Demanda Baset en el Videolan.

1.1. Video on Demand (VoD)

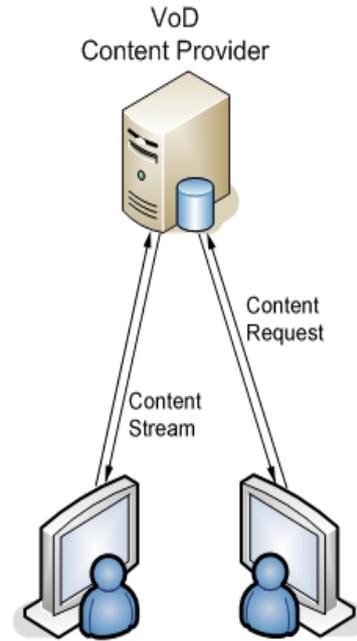
In the transmission of video on demand do not need the full download of the file for enjoy it. People waiting to save the file is completely transmitted.

Video on demand should not be confused with the video in real time. Although second is based on multimedia playback while downloading, is differs in that the information is not previously stored on a server.

Audio and video are encoded at the time, was broadcast on the Internet and displayed and listen in real time. This is what happens to the video lectures and broadcasts on direct. In this case the server is in charge of controlling the transmission.

Needless to say, currently the most popular system for video on demand is the Flash Video. It is a system that allows you to place your video on a

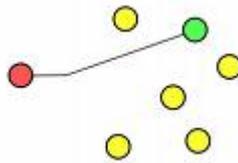
website very easily and a format suitable for display to raise it virtually to anyone. Pages YouTube, Google Video or MySpace using this system for the reproduction of their videos.



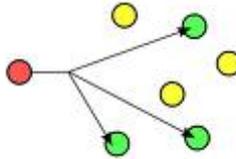
To generate a stream of video on demand it is needed to work with a streaming server. A streaming server is more than just a file server. It is a server that handles the delivery of real-time data. Is the server responsible for negotiating with the player of the end, use more speed suitable for the delivery of data. The server and client maintain a constant communications during the streaming process, so the server can respond to any problems suffered by the network or any demand generated by the client. Apart from managing the speed, the streaming server must maintain an interactivity with the user. Functions as Pause, Stop, ffwd, rwd, etc...

There are also three modes of data transfer:

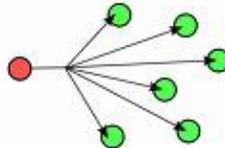
- Unicast: consists in sending information from a sender to one receiver. The communication is from a single host to another single host.



- Multicast: is sending information on a network to multiple recipients simultaneously.



- Broadcast: sending information to all nodes on a network.



They are responsible for interpreting the data streaming to play properly. Players can be used as a connector in a Web browser or a software application on your hard drive.

To receive and execute files streaming over the Internet, the client only must have a codec and a player. This player shall, if necessary, make the request to the file server. The process is as follows: the player begins to receive the file and builds the buffer to begin the storing of information. When the buffer is filled with a small part of the file, the player starts the video display and continues after downloading. The system is synchronized so that the files are available during the download. When the transfer is complete you just have to play it. When a connection is suffering a decline uses the speed information that exists in the buffer, so that they can support around this fall. If communication is cut too long, the buffer remains empty and the file playback is possible to make a reset signal.

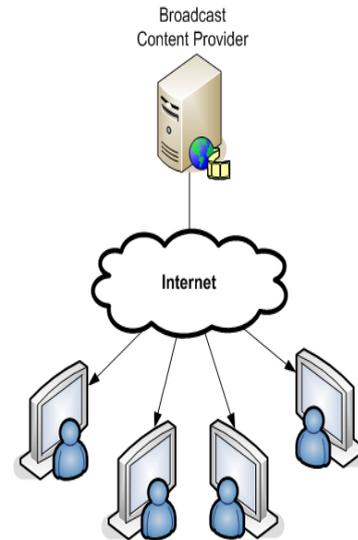
Below you have a list of players you can currently find:

- VLC: It is the player who presents VideoLan. You can play different formats audio and video as well as DVD's, CD's and various streaming protocols. It is a player who can work on most platforms for versions Windows, Mac OS X, BSD, Solaris and Linux. VLC uses a large number of bookstores codec. Many of these libraries are provided by libavcodec, the library of codecs from FFmpeg. Aside from player can perform other functions such as streaming server or transcoding, among others.
- Real player: Player is developed by RealNetworks that can play different types of multimedia formats including MP3, MPEG-4, QuickTime, Windows Media and many other versions of formats RealVideo and RealAudio. The latest version is 10.5, able to work on operating systems like Linux, Palm OS, Mac OS X, Windows, Windows Mobile or Symbian OS.
- Windows Media Player: It is the media player developed by Microsoft. Currently the 11 is the last existing. Allows the playback of various formats such as Audio CD, DVD-Video, DVD Audio, WMA, WMV, MP3, MPG, AVI and others, always the availability of appropriate codecs.
- QuickTime Player: It is the standard media player developed by Apple. It is available for different operating systems like GNU / Linux, Windows and Mac OS X.
- Flash Player: Multimedia application created and distributed by Adobe Systems. Use a language called ActionScript, which can be used to display Flash Video from a SWF file.

1.2. Live Streaming (broadcast)

Broadcast TV - system where TV content is pushed from the Content Provider (CP) on real time to all the users that want to watch it. Usually the broadcasted content from all the several CPs remains on the Service Providers (SP) network, and only the program the customer selects is sent to the user equipment.

Broadcast TV takes advantage from multicast network technologies to send the content to the users. In addition, we can see this behavior as the closest to the classic TV.



There are many solutions that fit this task, some of them are based on the typical real time streaming protocol (RTSP), others are based on the World famous BitTorrent technology.

The situations that the broadcaster will have to face are the following:

- There is only one receiver (Unicast).
- There are multiple receivers (Multicast).
- Each receiver also send the data that receives to other receivers (Torrentcast).

Obviously for each situation there are different ways to approach the problem. The main decision that a broadcaster has to deal with is to decide which technology fits better for his necessity.

Here are some examples of software that can currently find:

- VLC: VLC is the VideoLAN solution for streaming current use, among other functions. It is the most important project that has developed by VideoLAN. Can transmitted flux in unicast and multicast dissemination. The accepted formats are MPEG's (up to version 4), video formats for DVD, satellite channels, digital video TDT or live video on the net. Supported platforms for Windows, Linux, Mac and even for Set Top Box.

- Helix Server: Is your server streaming Real Networks that reaches in free version. Able to deliver a flow rate RealAudio, RealVideo, Windows Media, Quick Time, MP3, MPEG-4 or 3GPP online users on this server. Supported also by the various Windows operating systems, Linux and Solaris.
- Darwin Streaming Server: It is the open source version of the QuickTime Streaming Server "that Apple introduced, described in the following section of software. Server capable of working under platforms like Windows and Solaris.
- Windows Media Services: It is the latest version of the service package that offers Microsoft to install a their servers. Allows users to manage a flow stream from its Common Operating System Windows Server 2003 or Windows Server 2008.
- QuickTime Streaming Server: It is the application that Apple has launched to compete in the world streaming. It is a part of the software Mac OS X Server, an operating system based on the work a network server. A server very adaptable to work with any format due to its flexibility it offers. Methods can work with unicast and multicast, depending on user preferences.
- Flash Media Server: This is the version that Adobe Systems as a video server streaming high quality. Currently the market is Flash Media Server 2.0.4 is available for Linux and Windows Server 2003.

2. Server

A database is a data set that belong to the same context, systematically stored for later use. In this sense, a library can be considered a database mainly composed of texts and documents to printed on paper. At present, due to technological development in fields such as computers and electronics, most of the database are electronically, providing a great variety of solutions to the problem of storing data. In computer systems there are management database (DBMS), which allow store and subsequently access the data quickly and structured. The management database is a kind of very specific software dedicated to serve as an interface between the database and the user applications that use it. Is composed of a data definition language, a language for data manipulation and a query language.

The semantic meaning of a database where the metadata is indexed is explained in the next lines. In computing, represents the action of ordering records through indexes. A more slang to describe it would be stored together with its features that can describe (index). An example could be very simple: a library could keep their books, classifying them according to gender. Within this classification is User who will use the library to find the book you want to read. First choose a gender, for example comedy, and in the list of books that will choose an author for reduce the number of units to choose from. Databases often store their content indexes based on some previously defined parameters. Indexing provides flexibility in searching the translates into faster time to show results.

The search engine Google for example, stored in an index references to all content index pages, a record number of approximately 8,168 million pages. Without this system of indexing the queries take hours to display results.

2.1 MySQL²

It is a management system relational database and multi-user with more than six million installations. On the one hand is offered under the GNU GPL (GNU General Public License), but companies who want to incorporate it in private can buy license products at a company who permitting the use. It is developed for the most part in ANSI C.

² www.mysql.com

2.2 Digital Asset Management (DAM)

The term “digital asset management” also refers to the protocol for downloading, renaming, backing up, rating, grouping, archiving, optimizing, maintaining, thinning, and exporting files such as digital photographs, animations, videos and music. There are two primary types of DAM software: browsers and cataloging software. A browser reads information from a file but does not store it separately. Cataloging software stores information in its own separate file.



The asset is detailed by its metadata. Metadata is the description of the asset and the description depth can depending on the needs of the system, creator, or user.

For example, metadata can describe the description of what is in the package, the encoding and decoding process (e.g. MPEG2), the ownership,...

There exist some predefined standards and template for metadata such as Dublin Core, MPEG-21 which are explained later in this chapter.

In cases of systems that contain large size asset essences, such as MPEG 2 and JPEG2000, for the cases of images and video, there are usually related "proxy" copies of the essence. A proxy copy is a lower resolution representation of the essence that can be used as a reference in order to reduce the overall bandwidth requirements of the DAM system infrastructure. It can be generated and retained at the time of ingestion of the asset to the essence, or at least it can be generated by using transcoders.

Below there are some examples of existed systems:

- Brand asset management systems, with a focus on facilitation of content re-use within large organizations. Here the content is largely marketing- or sales-related, for example, product imagery, logos, marketing collateral or fonts, to give a few examples.
- Library asset management systems, with a focus on storage and retrieval of large amounts of infrequently changing media assets, for example in video or photo archiving.
- Production asset management systems, with a focus on storage, organization and revision control of frequently changing digital assets, for example in digital media production.
- Digital supply chain services, pushing digital content out to digital retailers (e.g. music, videos and games).

2.2.1 Akamai³

If you use the Internet for anything some reason for example; to download music or software, check the headlines, book a flight you have probably used Akamai's services without even knowing it. They play a critical role in getting content from providers to consumers.



Akamai helps worldwide companies to sell, inform, entertain, market, deliver software, as well as many others online. Some of the companies are like Audi, Fujitsu,... They have a service, called “Edge Control”, were they gain insight into worldwide Internet conditions and access to tools to manage their online business.

Important to know is when a customer (e.g. company) use an Akamai platform the IP address will point to an Akamai server and not to the customer’s server. Also each customer will receive content form the closest Akamai server which lead to a good connection. By this two point they will have faster download times and less vulnerability to network congestion or outages.

2.2.2 Fedora Commons⁴

This is my last example of a Digital Asset Management architecture. Fedora Commons was originally an organization that providing technologies to create, manage, publish, share and preserve digital content in the form of digital objects. As example the content can represent metadata of an digital object.



The Fedora Repository is an easy open source software and can be used to support any digital content type. Fedora is being used for a lot of different examples like digital preservation, open access publishing, document management, archives, and a lot more.

To give you an idea of which types of organizations who are using Fedora are museums and cultural organizations, government agencies, medical centers and libraries, broadcasting and media, ...

³ www.akamai.com

⁴ www.fedora-commons.org

2.3 Server languages

The language of the server side are those languages that are recognized, interpreted and executed by a server and sent to the client in a format that is understandable. Some examples are:

-PHP

PHP allows you to include their small code snippets into the HTML page and perform certain actions easily and effectively without having to program entirely in a language other than HTML. For another part, PHP offers also a lot of functions for the operation of database without complications.

-ASP

The types of servers used in this language are those that work with the operating system of the family of Windows NT. To write ASP pages use a scripting language, which are placed in the same web page with HTML.

-PERL

It is a language that adopts some features of C. Its beginnings as programming language Administration computers and has become a tool that is used in programming web pages, database, bioinformatics or can be used for artificial intelligence.

-AJAX

It is a language for creating interactive web applications. This way there is a possibility to make changes to the page without cluttering him.

3 Transmission

3.1 Coding – Formats

Before reading further you have to know that different video data is generated by audio- and video decoders. Those two decoders are combined to generate a fixed or current file. The container format (or the process of multiplexing) defines how to combine them.

The following graphic gives you an complete view of a video file.

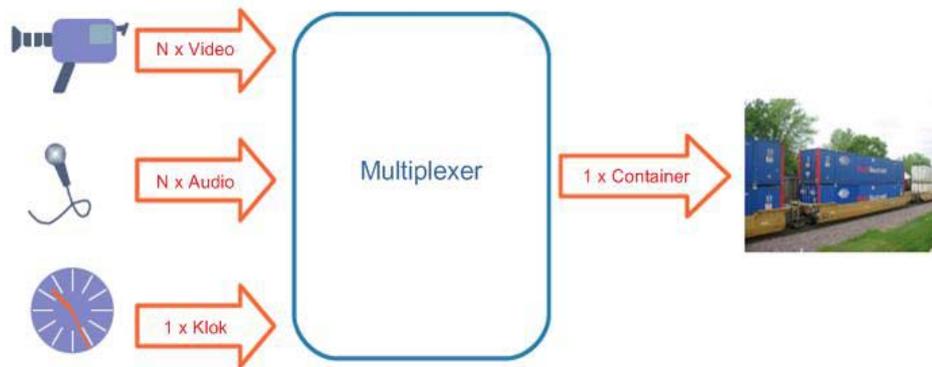


Figure 2: Video file.

Not all combinations of audio- and video codecs can multiplex all container formats. To help you to figure out which combinations are possible figure 3 shows a table of formats found and made by Videolan.

| | PS | TS | Ogg | ASF | MP4 | MOV | MPMPEG | Raw |
|-------------------------|----|----|-----|-----|-----|-----|----------|-----|
| Video formats | | | | | | | | |
| MPEG-1 video | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| MPEG-2 video | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| MPEG-4 video | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| DivX 1/2/3 video | ✗ | ✓1 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| WMV 1/2 | ✗ | ✓1 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| H/I 263 | ✗ | ✓1 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| MJPEG | ✗ | ✓1 | ✓ | ✓ | ✗ | ✗ | SVN only | ✗ |
| Theora | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| H.264/MPEG-4 AVC | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Audio formats | | | | | | | | |
| MPEG Layer 1/2/3 audio | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| AC3 (i.e. AS2) | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| MPEG-4 audio (i.e. AAC) | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Vorbis/Speex | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| FLAC | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |

Figure 3: Format table.⁵

⁵ www.videolan.org/codecs

3.1.1 Video codec

As you can see in the table you have plenty of choice. I will explain them one by one for you.

MPEG (Moving Picture Expert Group)

This is the work of some 350 members from various industries and universities. The working group has developed coding standards, audio and video creation and other auxiliary rules, standards and metadata, among others. These are the parts of standards for video:

- MPEG-1 (part 2): video codec for no interlocking signals(progressive).
- MPEG-2 (part 2) or H.263: video codec for broadcast quality television. Used for TV services by satellite, television signals and digital cable for DVD video discs.
- MPEG-4 (part 10), H.264 or AVC: It is a standard that defines a codec High compression video. The aim of the project was to create a standard capable of providing a good quality image with size report significantly lower than previous standards (MPEG-2 part2) no more than increase the complexity of its design.

Divx 1/2/3 video

This product is created by DivX, Inc. (formerly DivXNetworks, Inc.). The Divx codec has become popular due to its ability to compress lengthy video segments into small sizes while maintaining relatively high visual quality.

WMV 1/2

Those are the generic names given to the set of compression algorithms based on a group of video technologies developed by Microsoft. Those formats are almost always combined with the sound of Windows Media Audio formats.

H-263

Used primarily for videoconferencing, videotelephony, and internet video. H.263 represented a significant step forward in standardized compression capability for progressive scan video. Especially at low bit rates, it could provide a substantial improvement in the bitrate needed to reach a given level of fidelity.

MJPEG

MJPEG is a file containing a sequence of unconnected JPEG files.

Theora (Vorbis)⁶

It is a free video codec developed by the Xiph.org Foundation as part of Ogg project. Based on the video encoding with losses usually inside the Ogg container, combined with the Vorbis audio format. Know that in the latest version of Firefox, 3.5 Beta, contains this video codec.

H-264/MPEG-4 AVC

MPEG-4 Part 10 (a technically aligned standard with the ITU-T's H.264 and often also referred to as AVC). This emerging new standard is the current state of the art of ITU-T and MPEG standardized compression technology, and is rapidly gaining adoption into a wide variety of applications. It contains a number of significant advances in compression capability, and it has recently been adopted into a number of company products, including for example the XBOX 360, PlayStation Portable, iPod, iPhone, the Nero Digital product suite, Mac OS X v10.4, as well as HD DVD/Blue-ray Disc.

3.1.2 Audio codec

MPEG layer 1/2/3 audio

- MPEG-1 Audio Layer 1, commonly abbreviated to MP1, is one of three audio codecs included in the MPEG-1 standard. MP1 uses a comparatively simple sub-band coding, using 32 sub-bands. MPEG-1 layer I was also used by the Digital Compact Cassette format, in the form of the PASC audio compression codec.
- MPEG-1 Audio Layer 2 (MP2, sometimes incorrectly called Musicam) is a lossy audio codec defined by ISO/IEC 11172-3. While MP3 is much more popular for PC and internet applications, MP2 remains a dominant standard for audio broadcasting.
- MPEG-1 Audio Layer 3, more commonly referred to as MP3, is a digital audio encoding format using a form of lossy data compression. It is a common audio format for consumer audio storage, as well as the factor of digital audio compression for the transfer and playback of music on digital audio players. MP3 is an audio-specific format that was designed by the Moving Picture Experts Group.

⁶ www.vorbis.com

AC3 (i.e. A52)

This component was originally written by Shepmaster. A/52 (A52) is essentially the same thing as AC-3 (AC3), which is Dolby Digital. Dolby Digital is sometimes referred to as "DD", as in "DD 5.1" (for "Dolby Digital for 5.1 channels"). Dolby's Audio Coding-3 (AC-3) coding and compression technology was used by Dolby to create "Dolby Digital". Dolby Digital is the brand name used for the AC-3 digital audio encoding system. First used in 1995 for movie theaters, Dolby Digital is now also considered the standard for high quality audio for home theaters.

The "AC3" name (without the hyphen) is often used to refer to any audio encoding which can be used for DVD encoding to match Dolby Digital's specification, but is not specifically licensed by Dolby.

MPEG-4 audio (i.e. AAC)

MPEG-4 (part 3) Audio consists of various components are mostly programmed in a language called SAOL. Based on the compression signs of perceptual audio coding including some variations of Advanced Audio Coding (AAC). AAC stands for Advanced Audio Coding it is a digital audio lossy compressed format. It was designed to replace MP3. For a single impulse per second (bitrate) and a single MP3 file size, AAC is more stable and has better quality, so producing more crystalline. This is a format used for many applications such as Ahead Nero, iTunes or Winamp, among others.

Vorbis/Speex⁷

Free audio encoder compression loss. It is part of the Ogg project and is called Ogg Vorbis. Allows scaling on a variety of bitrates. The quality / bitrate (CD audio, 16/24 bit) is on the same level as MPEG-2. Know that in the latest version of Firefox, 3.5 Beta, contains this audio codec.

FLAC

Free Lossless Audio Codec (FLAC) is a file format for lossless audio data compression. FLAC reduces bandwidth and storage requirements without sacrificing the integrity of the audio source. A digital audio recording (such as a CD track) encoded to FLAC can be decompressed into an identical copy of the audio data. Audio sources encoded to FLAC are typically reduced in a size of 40 to 50 percent (46% according to their own comparison).

⁷ www.vorbis.com

3.1.3 Containers

The audio and video files are not stored or never transferred separately. Others are housed in containers called multimedia files. A media container is a type of computer file that stores information Video, audio, subtitles, chapters, and metadata information by using a synchronization preset format in its specification.

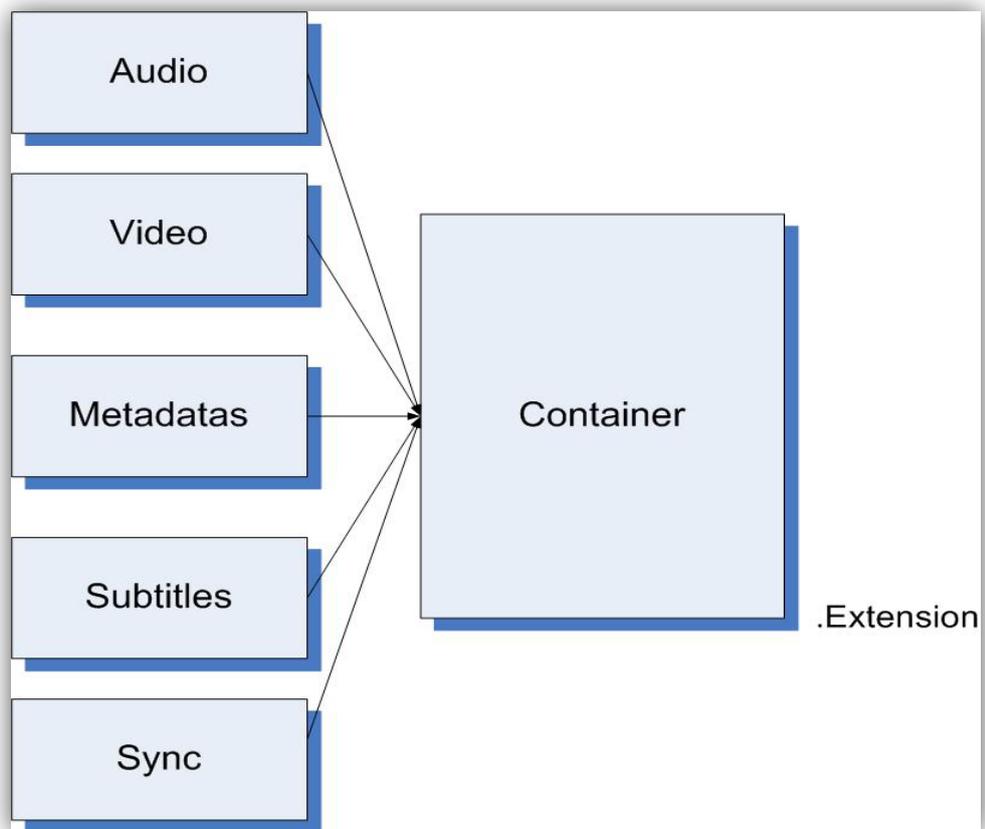


Figure 4: Container Multimedia

For the Playing video file container, operates on a first splitter (splitter) who knows the pattern of the container and separates (demultiplexing) the audio tracks and video. Once separated, each is interpreted by the decoder.

Examples of containers are:

MPEG

- MPEG-1 (part 1): More conegudament appointed as PS (Program Stream). Also transported to other formats.
- MPEG-2 (part 1): known as TS (Transport Stream). This version, the PS is designed to hold more for the current multimedia DVD's.
- MPEG-4 (part 12): has been developed by the MPEG committee as a part of the MPEG-4 standard. The format is a base format for media file formats. Many other file formats are based upon the MPEG-4 part 12 standard. The file format is object-oriented. A presentation, a combination of one or more motion sequences and audio, is described via objects, called boxes. A presentation can consist of one or multiple files. One file describes the structure of the presentation. The other files, if used, contain the media data or other information. This information and the media data can also be integrated in the presentation file. The file format can contain any file type and supports download and streaming.
- MPEG-4 (part 14): the official name is MP4. Designed to carry audio and video (especially if they are defined in MPEG), but also supports other data streams such as subtitles or images.

Container Flash Video (FLV)

FLV is a video container format and as such is similar in some ways to AVI. It's primary purpose is to facilitate video on the Internet more easily than other proprietary formats that require the installation of proprietary software with a lower installed base than the Macromedia Flash Player. Since Flash is a cross-platform and FLV video streams well, it lets web sites present video that is easily viewable by visitors with no extra effort. FLV Flash videos are usually embedded in Flash SWF files that are used to control the video with typical video player controls like play, stop, pause, and seek. Notable FLV users include YouTube, Google Video, and Reuters.

Ogg⁸

Ogg is a stream oriented container, meaning it can be written and read in one pass, making it a natural fit for internet streaming and use in processing pipelines. This stream orientation is the major design difference over other file-based container formats. This encoding format contains the video codec Theora and the audio codec Vorbis. Like mentioned by the both codec the latest version of Firefox, 3.5 Beta, includes those codecs which means that this version supports this format.

AVI (Audio Video Interleave)

It was defined by the Microsoft Video for Windows technology in 1992. Later it was improved by the extension of the group OpenDML by the company Matrox. These extensions are known as AVI 2.0.

ASF (Advanced Streaming Format)

Container format developed by Microsoft specifically designed to the streaming. The formats that generally the ASF are Windows Media Video (WMV) and Windows Media Audio (WMA).

RealMedia

Container format developed by RealNetworks. Its extension is ". Rm. Normally contains the formats, including RealNetworks, RealAudio and RealVideo, and is mainly used for streaming on the Internet.

MOV

The MOV container has been developed by Apple in 1991. The format served as a base for the MP4-container format. The video streams can be in any format supported by the Quicktime codec manager, such as MPEG-4 and Sorensen. On the other hand, the audio streams can be in any format supported by the sound manager and coreaudio, such as AIFF, WAV and MP3. A MOV container is build up out of atoms. These atoms are hierarchically structured and are either parents or contain media or data. MOV containers contain a timeline, separated from the media streams. The timeline allows editing of MOV-files without the need of copying media streams. MOV containers are mainly used on Apple's operating system Mac OS X.

⁸ www.vorbis.com

3.2 Metadata

Literally "metadata" means data that describes other data. In general, a group of metadata is a set of data called a resource. The concept of metadata is analogous to the index to locate objects, but metadata helps to locate other data.

The most frequent use of metadata is also working with search engines. Using additional results are more precise, and the user saves leaks complementary manuals.

Here are some examples of metadata standards more familiar:

3.2.1 MPEG-7 [6]:

It consists of a standard representation of audiovisual information that allows content description (metadata) to:

- Keywords
- Semantic meaning (who, what, when, where)
- Meaning structure (shapes, colors, textures, movements, sounds).

The first version was adopted in July 2001 and currently the latest version approved by the ISO date October 2004. MPEG-7 looks for ways to link the elements of audiovisual content, find and collect the information that the user needs and identify and protect the rights of the content.

3.2.2 MPEG-21 [10]:

Its purpose is to distinguish who are the participants in a market digital goods where possible data are binaries. The basis is defined digital objects. These are resources that are exchanged in the digital market established by the MPEG-21 standard.

3.2.3 *Dublin Core:*

Dublin Core [11] is an RDF-based standard that represents a metadata element set intended to facilitate the discovery of electronic resources. There have been many papers that discussed the applicability of Dublin Core to non-textual documents such as images, audio and video. They have primarily focused on extensions to the core elements through the use of subelements and schemes specific to audiovisual data. Dublin Core is currently used as a metadata standard in many TC archives.

In this context, one has also to mention which is the Meta-data Dictionary SMPTE (Society of Motion Picture and Television Engineers). It is a big collection of registered names and data types, developed mostly for the television and video industry that forms the SMPTE membership. Its hierarchical structure allows expansion and mechanisms for data formatting in TV and video-signals, and provides a common method of implementation. Most metadata are media specific attributes, such as timing information. The SMPTE web site⁹ contains the standards documents.

3.3 Streaming

The protocols define the transaction that is used to establish a connection and transmit the media from the server and client. i.e. streaming protocols made streaming media feasible by transmitting data from the host server over the Internet to the client player or end listener.

More recent protocols such as the Real Time Streaming Protocol (RTSP) and Internet Group Management Protocol (IGMP) are making the transmission of data even more efficient.

This thesis represents a Video on Demand service. As protocol for Video on Demand services the common choice is RTSP. On the other hand, for live TV broadcast, multicasting with Internet Group Management Protocol “version 2” (IGMPv2) [2] is the option.

⁹ www.smpete.org

3.3.1 *RTSP [1]*

RTSP is an application-level protocol for control over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips. This protocol is intended to control multiple data delivery sessions, provide a means for choosing delivery channels such as User Datagram Protocol (UDP), multicast UDP and TCP, and provide a means for choosing delivery mechanisms based upon Real Time Protocol (RTP).

3.3.2 *IGMP*

Internet Group management protocol (IGMP), a multicasting protocol in the internet protocols family, is used by IP hosts to report their host group memberships to any immediately neighboring multicast routers. Routers that are members of multicast groups are expected to behave as hosts as well as routers, and may even respond to their own queries. IGMP may also be used between routers, but such cases is not really common. IGMP is an integral part of IP. Hence, it is required to be implemented by all hosts wishing to receive IP multicast. IGMP messages are encapsulated in IP datagram's, with an IP protocol number of two.

4 Client

4.1 Devices

4.1.1 Mobile

This is a large group that is still growing. Mobile devices allow you to communicate with others or get information at any time and anywhere. Those starts from mobile cell phones to palmtops. The next two points are examples from mobile devices. Do not forget those are just a small part from big group.



- iPhone is the big widescreen iPod whose screen resolution is 320x480 which is bigger than the video iPod's . After charging the battery, you can play high-quality movie up to 7 hours and enjoy music up to 16 hours. I.e. you are able to watch four movies without charging again.

Now you can ask you the question “Which file formats is supported by iPhone?”. Many mobile phones supports 3GP, 3G2 formats, but iPhone [12] does not. There is no doubt that iPhone will support H.264, MPEG-4 in .mp4, .mov, .m4v formats, and MP3, protected AAC, AAC, Audible, Apple lossless in .aac, .mp3, .m4a files. The native video size of iPhone is 480x320. Just as the 5gen video iPod, iPhone need to sync the media via iTunes software.

- Android [14] is a software platform for mobile devices, powered by the Linux Kernel, initially developed by Google and later the Open Handset Alliance. Android delivers a complete set of software for mobile devices: an operating system, middleware and key mobile applications.



To give an example of one of their mobiles is the T-mobile G1 [13]. The G1 is the first Android-based smartphone. It is slightly bigger and a tad heavier than the iPhone. The device is 4.60 inches tall, 2.16 inches wide, 0.62 inches deep and weighs 5.6 ounces. Compare that to the iPhone, which is shorter, thinner and lighter: the iPhone is 4.5 inches tall, 2.4 inches wide, 0.48 inches deep and weighs 4.7 ounces. G1's display comes in at

3.17 inches vs. iPhone's 3.5 inch screen. Further both devices, iPhone and T-Mobile G1, have regular functions. For example; camera, real web browsing, 3G network and Wi-Fi access,...

4.1.2 PC s

First of all we need software to watch a video. The group of this software is called "video players" and are capable to read different video formats on a computer. Because there is a large group of video players, each with their own layout and specifications, it is up to the user himself to choose one or several video players. There is the possibility that not every video player will support every video format. For some examples of players there is a reference to the beginning of this chapter under the topic "Video on Demand".

Beside the video players it is very important to provide access from a regular web browser with a video player plug-in. A plug-in is used to provide a certain, usually very specific, function "on demand". Applications, for example a web browser or an email client, support plugins for many reasons. Some of the main reasons include: to reduce the size of an application - to separate source code from an application because of incompatible software licenses.

Plug-ins depend on the services provided by the host application and do not usually work by themselves. They differ from extensions, which modify or add to existing functionality. Plug-ins rely on the host application's user interface and have a well-defined boundary to their possible set of actions. Extensions have fewer restrictions on their actions, and may provide their own user-interfaces. Figure below is an example of a plug-in framework.

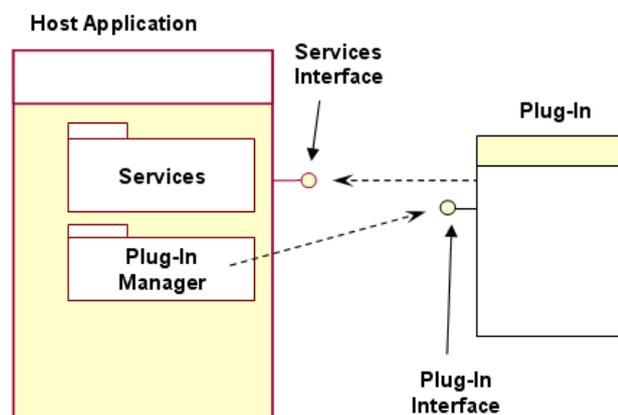


Figure 5: Plug-in framework

Let us take a loop under the video player and plug-ins for different web browsers. Below you find a list of existing plug-ins.

- Adobe Flash Player: Flash Player is a cross-platform browser plug-in that delivers breakthrough Web experiences to over 99% of Internet users and is created by Adobe.
- Quick Time Player: QuickTime Player is an easy-to-use application for playing, interacting with or viewing video, audio, VR or graphics files. This plug-in is made by Apple.
- VLC: VLC media player is a highly portable multimedia player and multimedia framework capable of reading most audio and video formats (MPEG-2, MPEG-4, H.264, DivX, MPEG-1, mp3, OGG, AAC,...) as well as DVDs, Audio CDs VCDs, and various streaming protocols.
- Real Player: RealPlayer enables your computer to play streaming RealVideo and RealAudio. This plug-in is made by Real Networks.
- Windows Media Player: Windows Media Player lets you play streaming audio, video, animations, and multimedia presentations on the web and is created by Microsoft.

4.1.3 Set Top Boxes & TVs:

A Set Top Box is a device who receives an incoming signal. One of his functions is to convert the incoming signal to an useful signal for a television. An example of an signal should be one who is coming from a Digital Video Broadcast (DVB). Other functions of a Set Top Box is to display analogue television or other video content from the internet. Like mentioned in the introduction chapter is this project linked to the project made by Alexander Schmidt [17]. He worked on the interface for a STB from the company ADB called type: “HD-3800/TW”.



This STB from ADB is for example an advanced High Definition IPTV Set Top Box with home networking capabilities. The STB is also compatible with High Definition (HD) and Standard Definition (SD) television transmissions. About the formats this STB includes MPEG2, MPEG4/H.264 and VC-1 Advanced Video Coding (AVC).

If we talk about a standard for Digital Television (DTV) we can say that those support display formats that are relatively consistent with legacy analog TV formats. Standard Definition Television (SDTV) is also distinguished from High Definition Television (HDTV). Specifically, SDTV specifies two formats which you can see in the table below.

| Vertical Lines | Horizontal Pixels | Aspect Ratio | Refresh Rate (fps)* |
|----------------|-------------------|--------------|---------------------|
| 480 | 704 | 4:3,16:9 | 24p, 30p,60p ,60i** |
| 480 | 640 | 4:3 | 24p, 30p,60p ,60i** |

* fps = frames per second ** I = interlaced, p= progressive

The transmission of signal quality in DTV have the issues to manifest in artifacts such as blocking or tiling and stuttering. The Advanced Television Systems Committee (ATSC) specifies MPEG2 compression and a transport subsystem. On the other hand, the audio compression is based on the AC-3 specifications from Dolby Digital.

SDTV standards were developed by the Grand Alliance and reviewed, tested and documented by the ATSC at the request of the United States Federal Communications Commission (FCC).

4.2 Client languages

The browser is an application that is able to interpret the orders received in the form of HTML code and display it in content. When a user clicks on a hypertext link, in reality what happens is a set of requests from a resident web server (a computer that is constantly connected to a network) which is sent and interpreted by the browser (the client). However, if the page you requested is not an HTML file, the browser is unable to interpret it and should make use of an external application.

The languages of the client side are those that can directly be "digested" by the browser and require no pre-treatment.

-HTML

Is one of the languages on the server side and also extended to the Internet. I.e. HTML is the language with which web pages are written. These can be seen by a user using an application called a browser. One could say that HTML is the language used by browsers to display pages to the user. This language allows us to combine text, sounds and images. Also allows the introduction of references to other pages through links called links. We did not think that the web would become a recreational area with a multimedia, so the HTML was not created to respond to all possible uses and all the groups of people used in the future. However if you have incorporate changes over time, these are the standards of HTML and currently the latest version is HTML 4.01, presented in September 2001. A HTML file is simply a text. So for programming in HTML, we need a text editor such as OpenOffice, Microsoft Word or a simple notebook.

-CSS

The HTML is limited to the application form to a document. The solution was the introduction of the CSS pages. Their mode of operation is defined in a special syntax, the presentation format can be applied to:

- An entire website (all pages).
- An HTML document or a fragment of it (head, body or even the full page)
- A specific label, you can set up different styles for a single.

-JavaScript

The great advantage of JavaScript is not required that a development kit for their programming, and compiling scripts or make them into documents external to the HTML, JavaScript language is written in HTML.

Requirements

The present project provides a technical to a set of requirements defined by the Signal Theory and Communications (TSC) Department of the UPC in the EUETIT. During their research and teaching activity the department generates video content that wants to be published and available on the Internet. Examples of such content are demos and thesis defenses from the research side, and tutorials and lectures from the teaching side. The final system must provide a management interface for the system administrator to easily upload videos and edit their metadata, and also the possibility to consume the video content from a diversity of user interfaces, such as a web browser, a STB or, in the future, a mobile device such as an iPhone.

The design of the video server presented in this thesis aims at providing solutions to the following requirements:

- Debug the existing platform when editing and deleting database entries.
- Debug the existing platform to ensure the deletion of the actual content from disk when an entry is deleted from the database.
- Clean and comment the source code organization of the existing platform.
- Make the video server public on the Internet.
- When possible, use only free formats. If this is impossible, standard formats must be used instead, but never use proprietary formats.
- Configure the platform to provide remote database access to the Set Top Box interface on ADB.
- Set up a video streaming server capable of sending the content to the ADB Set Top Box.
- Offer scalability to the video service by automatically generating different version of the content adapted to the consumption device. Three cases are defined: high definition (large screens), standard definition (traditional TV monitors) and mobile devices (small screens).
- Generate automatically the thumbnails from a keyframe of the video that allow a more intuitive navigation through the video archive.

Chapter 4

Installation

This chapter explains which software has been used for this project. Note that the operating system for the facilities is Ubuntu version 9.04 created by Linux.

The following figure shows each software element in the process:

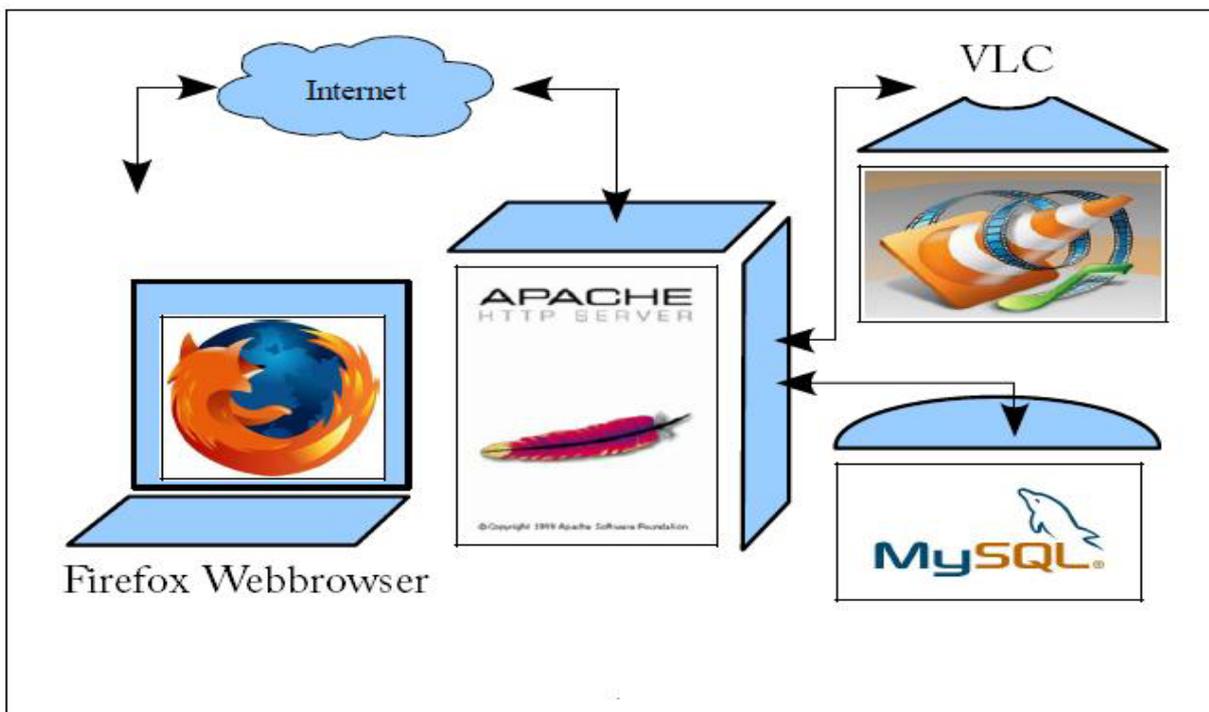


Figure 6: Software structure.

Because the VLC plugin only works on Mozilla browser Firefox, users can use the service only if they are using Firefox Webbrowser.

The Web Server Apache contains the element that will serve the requests of users for the various functions that service offers. For any modification of the database from the web portal, Apache communicates with MySQL, who manages the metadata of the video. By video service demand Apache communicates with the use of VLC for video playing.

In this part you will find two types of software: Server Software and Client Software. Beside the two types of software there is also an explanation for the configuration of XAMPP.

1. Server Software

The first important issue was to find a motor or video sever that was capable of transmitting video over the network. I would like to make a reference to David Vera's thesis. He worked with the video player VLC.

1.1 XAMPP

It is a platform independent software free, mainly consisting in database MySQL, the Apache Web server and interpreters for scripting language: PHP and Perl. The name comes from the acronym of X (for any different operating systems), Apache, MySQL, PHP and Perl. The program is released under the GNU license and acts as a free web server, easy to use and able to interpret dynamic pages. XAMPP is currently available for Windows, Linux, Solaris and Mac OS X.



The installation of XAMPP [15] on a linux platform is quite simple when you follow the next steps one by one. In total you have a four step installation - step 1 = Download – step 2 = Installation – step 3 = Start – Step 4 = Test. You can find this method on the website of XAMPP as well.

A good thing to know is that in the past this software was called LAMPP but to avoid misconceptions they renamed it to »XAMPP for Linux«.

- Step 1 = Download

Go to the Xampp¹⁰ website and link yourself to Xampp for Linux. First step is to click on the link below so you download the latest version of xampp.

¹⁰ www.apache.org

- Step 2 = Installation

After a perfect download it is time to open a terminal in Linux and write the following commands;

- First you have to login as the system administrator root;
su
- Extract the downloaded file to folder /opt;
tar xvfz xampp-linux-1.7.1.tar.gz -C /opt

- Step 3 = Start

To start xampp you have to use the next command;
/opt/lampp/lampp start

- Step 4 = Test

To check yourself on mistakes you can type the url at a web browser

http://localhost

Now you should see the start page of XAMPP containing some links to check the status of the installed software and some small programming examples.

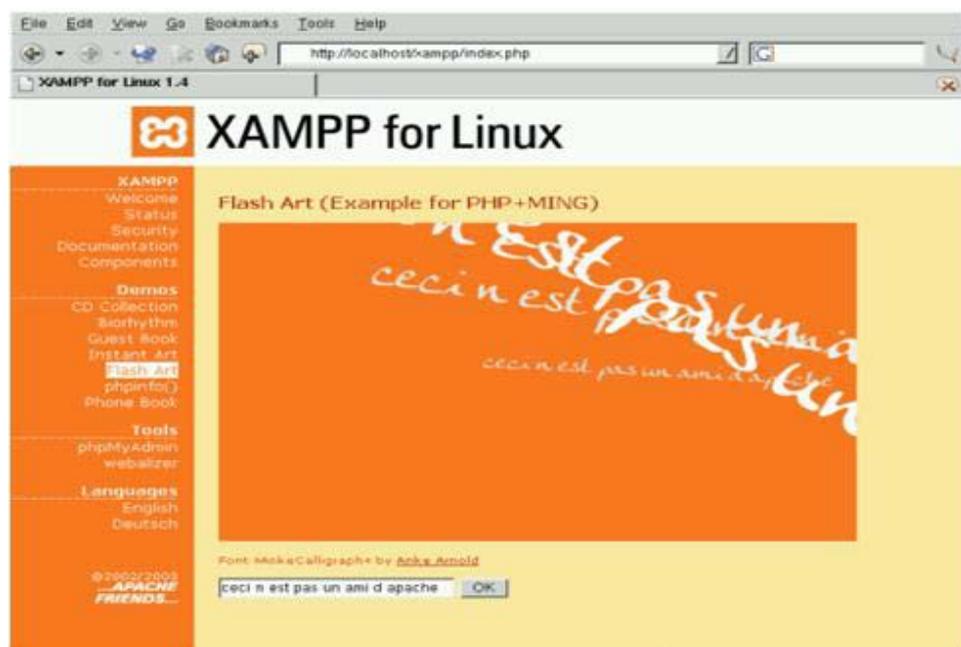


Figure 7: Start page of Xampp.

1.2 FFmpeg

To record, convert and stream digital audio and video in numerous formats we are using FFmpeg [16] which is a computer program. FFmpeg is a command line tool that is composed of a collection of free software / open source libraries. It includes libavcodec, an audio/video codec library used by several other projects, and libavformat, an audio/video container mux and demux library.



FFmpeg is developed under Linux, but it can be compiled under most operating systems, including Apple Inc. Mac OS X, Microsoft Windows and AmigaOS.

As you know you can use this program to convert a video. Now we also will use this software to generate the thumbnails of a video. Thumbnails are reduced-size versions of pictures, used to help in recognizing and organizing the different videos.

For the transcoding and transcaling of the videos in this project it is not enough to install the regular system of FFmpeg. i.e. as on the top comment this software includes libavcodec library and libavformat but not the x264 library package that we need in this project. X264 is a free software library for encoding video streams into the H.264/MPEG-4 AVC format. It is released under the terms of the GNU.

In the lines below you find a great working command list to install the completed library package [17] and the program ffmpeg (notice, this is only for a Linux platform).

- If FFmpeg is already installed on your computer you need to unstill x264, libx264-dev and ffmpeg. Open a terminal and run the following command lines:

```
sudo apt-get purge ffmpeg x264 libx264-dev
```

- Now we will find all the necessary packages and FFmpeg software:

```
sudo apt-get update  
sudo apt-get install build-essential subversion git-core checkinstall yasm  
texi2html libfaac-dev libfaad-dev libmp3lame
```

- Get the most current source files from the official x264 git repository, compile, and install them:

```
cd  
git clone git://git.videolan.org/x264.git  
cd x264  
./configure  
make  
sudo checkinstall -fstrans=no -install=yes -pkgname=x264 -pkgversion  
"1:0.svn`date +%Y%m%d` -0.0ubuntu1" -default
```

- Finally you need to get the most current source files from the official FFmpeg svn, compile, and install as well:

```
cd  
svn checkout svn://svn.ffmpeg.org/ffmpeg/trunk ffmpeg  
cd ffmpeg  
./configure -enable-gpl -enable-postproc -enable-pthreads -enable-libfaac -  
enable-libfaac -enable-libfaad -enable-libmp3lame -enable-libtheora  
make
```

2. Client Software

Before a user can use the services of this project he needs to install two elements: the Mozilla Firefox browser and the VLC plug-in for this browser.

2.1 Mozilla Firefox

It is a web browser with a graphical user interface developed by the Mozilla Corporation and a large number external volunteers. Firefox started as a derivative of Mozilla Application Suite, which ended up replacing Mozilla as the flagship product of the Mozilla project. For the installation in Linux, just type the command below in a terminal window;

- Download the Firefox file from the Firefox website. Second, install the file to /opt/firefox:

```
sudo tar -jxvf firefox-3.0.tar.bz2 -C /opt  
rm firefox-3.0.tar.bz2
```

- Link to your plugins and remove totem-mozilla as it doesn't seem to work with Firefox 1.5.x or 2.x:

```
sudo mv /opt/firefox/plugins /opt/firefox/plugins.bak  
sudo ln -s /usr/lib/firefox/plugins /opt/firefox/plugins  
sudo rm /usr/lib/firefox/plugins/libtotem_mozilla.*
```

- To ensure it is used as the default version, modify the symbolic link in /usr/bin:

```
sudo dpkg-divert --divert /usr/bin/firefox.ubuntu --rename  
/usr/bin/firefox  
sudo ln -s /opt/firefox/firefox /usr/bin/firefox
```

This is may be not the easiest way to install Firefox Mozilla. But because of the upgrade version of Ubuntu 9.04 there were some problem with the normal way of installation. The normal way should be to use the default package installation under Ubuntu. For the installation in another operating systems (e.g. windows) there is a reference to David Vera's thesis for Firefox in windows in the chapter "Installations".

2.2 VLC

VLC [7] is a video player and audio will play a variety of formats. Its latest version also includes a streaming video server which is used for shooting video on demand. For installation of VLC on a Linux platform you need to run the next commands in a terminal.



- To find the video player type next command;
sudo apt-cache search vlc
- To install the video player vlc;
sudo apt-get install vlc

The second issue was how to store all the videos and which program is used as a Web server. There is a software package that addresses all these needs, called XAMPP and others among includes Apache (web server) and MySQL (Manager database).

2.3 VLC plug-in

The connector uses VLC to play videos directly in the websites without the need to open the application. Installing the connector is done in Linux. Implement again some commands in a terminal window to install the plugin.

- To find the vlc plugin:
sudo apt-cache search mozilla-plugin-vlc
- To intall the vlc plugin:
sudo apt-get install mozilla-plugin-vlc

Pay attention, at the last moment of this improvement project we upgraded Ubuntu 8.10 to Ubuntu 9.04. Because this is a new application of Lunix we figured out that there are problems with the mozilla-plugin-vlc. You will find a refer in the Chapter “Future Work” to solve this problem later.

3. Configuration of Xampp

Once we have installed the software on the server mentioned in the previous pages we need to fit the settings for a good use of the project. I would like to refer this part to David Vera's thesis [18] where he explains the work carried out with XAMPP for Windows. We will explain what has been changed during the period of this project.

As explained in the installation of XAMPP software, once installed, the programs can use Apache, PHP and MySQL, which are those that will be needed. To work with XAMPP you know that, once installed, XAMPP puts the Apache engine on to work on the localhost computer so that from any browser installed on your PC should have access to the following URL: <http://localhost/> . By doing so, the installation will be checked.

Finally, this means that Apache will be available for any external HTTP request at IP address linked to the PC.

Below you can find the interface XAMPP.

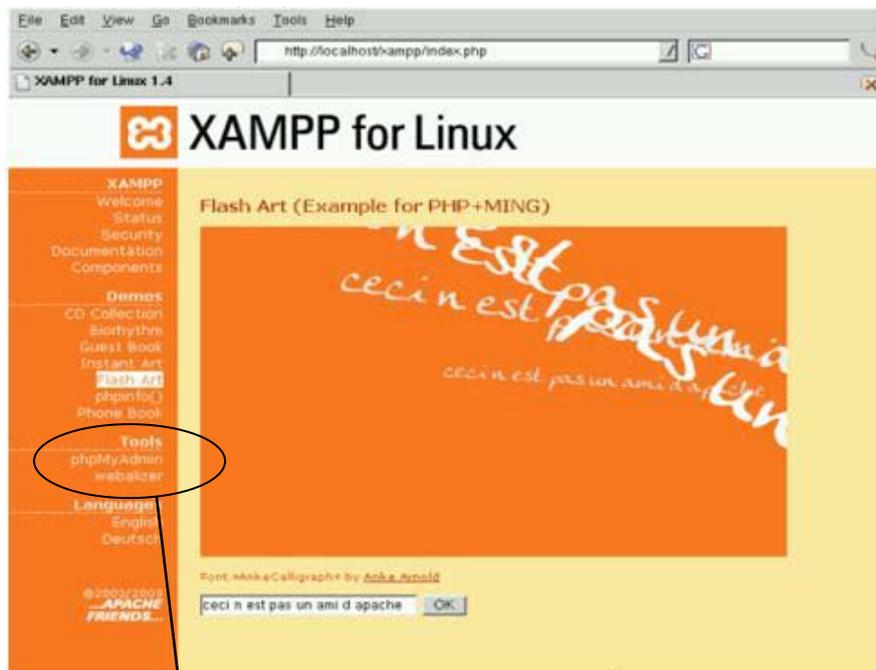


Figure 8: Interface of Xampp.

Figure 9 shows a part of this interface. In the submenu of the website are links to different services XAMPP provides. In this case the link is "phpMyAdmin" which manages all of the database.

3.1 MySQL database

To access the MySQL database figure 10 shows the interface of the before mentioned "phpMyAdmin" which is the graphical interface of the MySQL server:

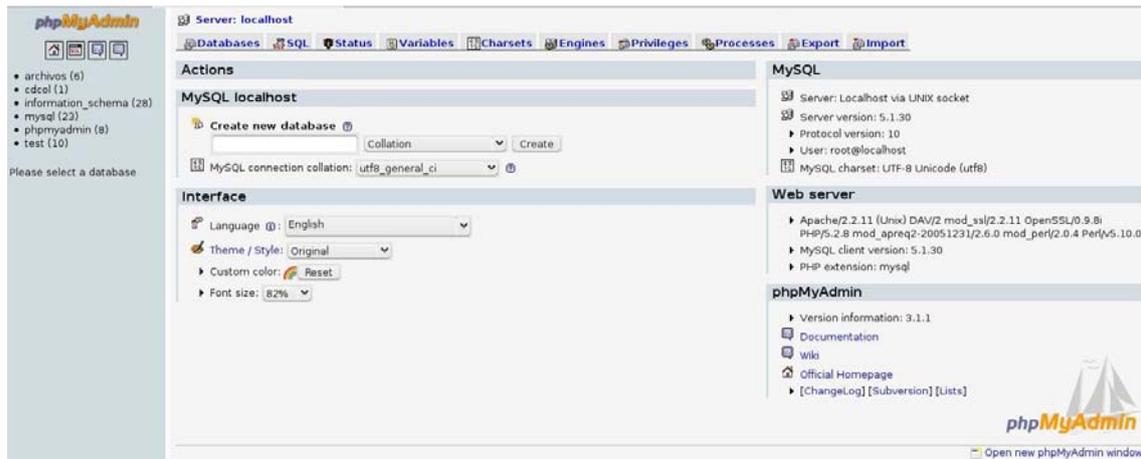


Figure 9: Interface of MySQL server

From this interface you can create database tables and users with their appropriate permissions and this all without the need to write SQL code. I.e. a very useful tool to manage databases. In this project 5 tables were created in the database. It is worth mentioning you should always use lowercase letters without accent.

To go on we continue with how to create an “MySQL users with phpMyAdmin” and “Creation & management of the database”.

3.1.1 MySQL users

The creation of the users who have access to the database can be determined by giving different permissions for different users to access the base data. This means not everyone can access the content. In this case we created the users "Laurens", "Alex" and "Albert". They have all the permissions for the database.

The previous management is performed with the following steps:

- Under the title of "localhost" is a blue link called "Privileges". This leads to the web page below. The table represents all the users created. Users are placed in each row and to change their options you should click the icon that is on the end of each row (half of a body with a stick before.)

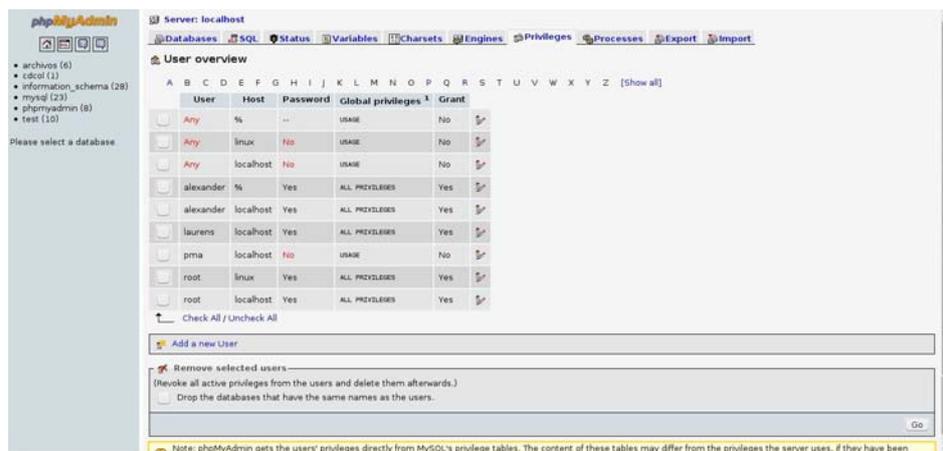


Figure 10: Privileges – User creation.

- Creating new users is very intuitive. You must click the "Add new user", which leads to the following:

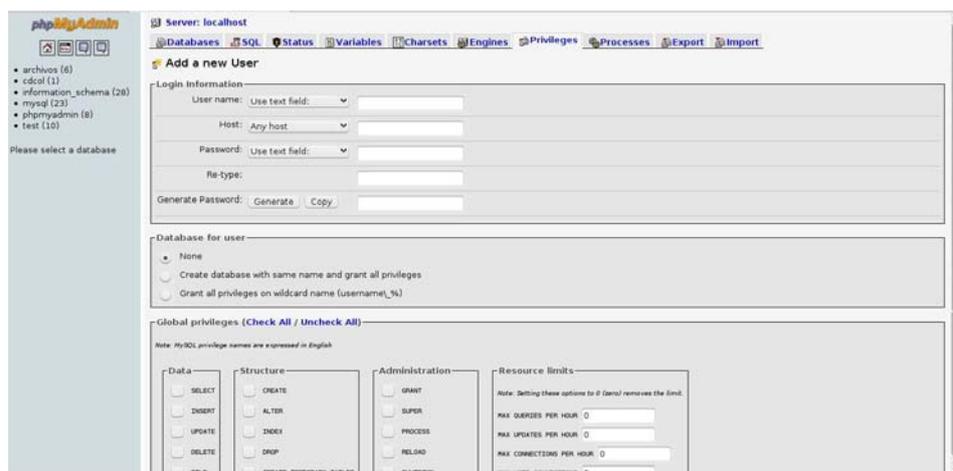


Figure 11: Privileges - Add new User.

This is where you can create a user. Thus to access the database from the PHP code (we will see later) they won't have any problems. The section "Resource Limits", that can be seen in the photograph above, was left fixed at 0 for no limit set.

- A very important step is to have the privileges for the users database that is created in MySQL. This step is performed when the users are created. Let us go to the table of the figure 12 and click on the icon, at the end of the row, for each user (discussed above). You enter a series of submenus where you have to change the parameters, and you can see the blue box at the following figure.

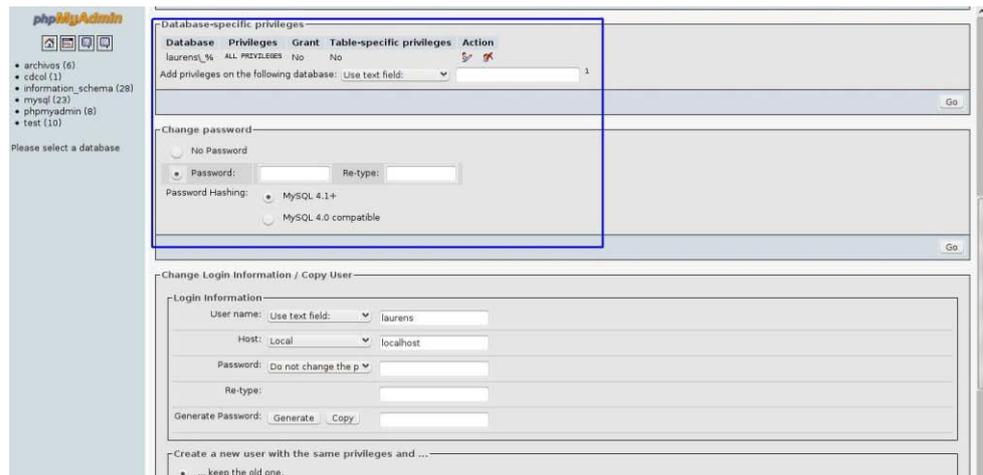


Figure 12: Privileges – Parameters.

3.1.2 Creation & management of the video database

The database of David Vera’s project contained two tables. One, “usuarios”, to store the information of the users and the second table, “videos”, was to store the information of the video. Some examples of the stored information of a video on David Vera’s project were the id-number of the video, name, manager (who uploaded the video), three different extensions (low, standard, high), date, three different url’s (low, standard, high), description and topic (in which part of the university; docents, students, pfc,...).

To see the different requirements of my thesis compared with David Vera’s thesis I refer to two chapters in the both thesis. In the next pages below we will explain the new structure of this project. The first step of this process is creating a database.

- If you look at the phpMyAdmin interface you will find next to the subject “permissions” the subject “ database”. This one will take you to the creation of management process on the database. The result after clicking the link can be seen in the following figure.



Figure 13: Management process on the database.

To create a new database you must fill in the text field (see blue box in the figure above), by entering the name and the organization and finally click the create button. The attribute of "Planning" refers to the encoding type for the database. i.e. it is noted that the application takes no account of the lowercase or uppercase accents, among others. This field is not very important but has been given the value of "Latin1_general_cs" which means Multilingual (Western Europe). Later they will ask for the creation of tables on the database. Filling text fields "name and number of fields so you will create the first table in the database.

- The last step is to fill the attributes of the tables as you can see in figure below.

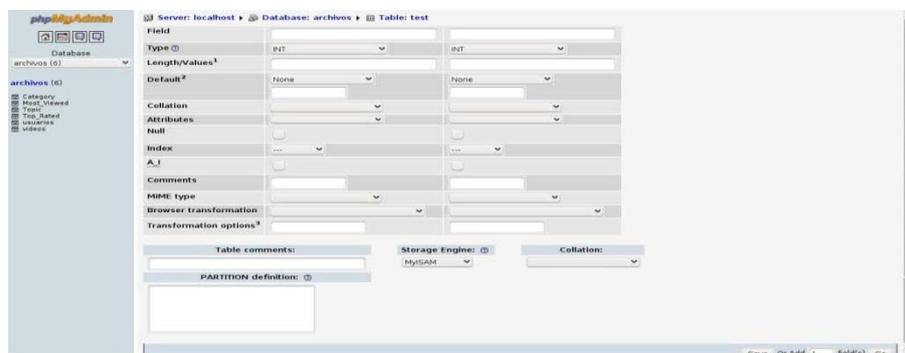


Figure 14: Attributes of the tables.

The attributes of the tables contains "field", "type", "Length / Values," "Planning" and "Extra".

In the installation we had an access denied error in MySQL. i.e. we were not able to access the database anymore. This problem can be solved by changing the root password in the config.inc.php file. Below we will go step by step to work this out;

- The directory of the file on my computer;

```
/opt/lampp/phpmyadmin/config.inc.php
```

- After finding this I didn't have the permission to edit the file, so I changed the execution permission on the file by this next command;

```
chmod 777 /opt/lampp/phpadmin/config.inc.php
```

- Now you have to open the file and change the root password.

```
find $cfg['Servers'][$i]['password'] = ";  
add your password $cfg['Server'][$i]['password'] = 'new_password';
```

The difference between this project and David Vera's one is the structure of the database.

This project contains six tables; users, videos, category, topic, most viewed, top rated. In the chapter B, appendix "Database figures" you will find figure 16 "whole structure of this project" with a description of each table explained below.

Lets have a closer look at the first table "users". The part will contain the information of each user with all the permissions.

- Id = Each user gets an identity number, easy to see how much users the database has.
- User = Contains the name of the user.
- Password = Each user has a personal password to access the database.

Second table "videos" contains the information of all the uploaded videos on the database.

- Id = Unique identifier for each item in the table. Is of type bigint () (the bigint range is 0 to 18446744073709551615). The (100) of bigint means length values, which was considered to 100 for service that is given over to a value well enough to work with margin. The "id" attribute has the "AUTO_INCREMENT", which means every time you introduce a new element to the table the "id" and will automatically create a "Id" for every single video. This helps the administrator to identify videos if there are a large number of them. This unique identifier also has the attribute of being "primary key" to MySQL knows that this field will only record in the table. In the case that you have two videos with the same name, same date, the same type or any field you won't have a problem because you will never have two videos with the same id.
- Name = This is the field of each record which will display the name of the video. This field and other fields that are less "description", will have the same attributes. The attribute "varchar (100)" means that this field will accept all types of chains characters with a maximum length of 100. This ensures there won't be errors for the length of the records because it is very difficult to put a name to a Video that surpass the 100 characters.

- Author = As a function of the service in this project is to upload videos in the database (usually professors), it was thought to create a field "manager" to indicate who had uploaded the video provided. Only find the names of teachers who are there on the table "User", which will be covered later.
- Date = This represent the date of each uploaded video is recorded or the date of your last update. The format is dd / mm / yyyy, for example: 02/01/2008.
- Description = This field indicates the description of the contents of a video on the database. It has the attribute "type" to "varchar (500)" to support the maximum length of 500 characters. This value is chosen because the description of the contents of a video did not by believe necessary to use more than 500 characters.
- TopicId = This identifier is important for the interface that is made by Alexander Schmidt. Via this way Alexander knows in each topic course their different videos.
- UrlHigh, UrlStd, UrlLow = Those are the directories for the three different video definitions. For example UrlHigh : http://[IP-address]/videos/high/[Id_number video]. [extension of video].
- ThumbnailUrl = This directory is the place where the thumbnail (keyframe) is located of each uploaded video. For example: http://[IPaddress]/thumbnails/[Id_number_video].MPEG.

The other tables contains some of the same fields as those in the table "videos". The following lines gives you an explanation of the rest of the tables.

- Category = This part is to show all the categories on the interfaces otherwise you have to look into the code to find those and this is not a good way.
- Topic = With this table we are able to show all the topics (for example; Machine learning, Image Processing, Audio Processing, Signal & Systems, Content delivery and management,...)
- Top Rated = Every client who watch the videos will be able to rate each of them. This gives the result that one video will have the highest rating (top rated).
- Most Viewed = This will give the five most viewed videos on the database. You can compare it with the most viewed videos on Youtube.

To show you from each table on the database with their contained fields a represented figure. See chapter B, appendix "Database figures".

3.2 Automatic start of Xampp/Lampp

We realized that when the server starts, lampp is not starting up by default. The solution for this problem is to generate a script for the initialization on startup. This problem can be solved by following the next steps;

- We created a script on my Desktop called 'runlampp.sh'(attached to this message). The content of the script is;

```
sudo /opt/lampp/lampp start
```

- After this we copied the script to '/etc/init.d' by this command;

```
sudo cp /home/laurens/Desktop/runlampp.sh /etc/init.d
```

- Then we had to change the execution permission on the file by this command;

```
chmod 755 /etc/init.d/runlampp.sh
```

- Finally we just have to configure the startup launch of the script with update-rc.d. This application creates symbolic links from several run levels to the script.

```
sudo update-rc.d runlampp.sh defaults
```

3.3 Apache and PHP

3.3.1 Apache

Apache [8] is the web server component of the popular LAMP web server application stack, alongside Linux, MySQL, and the PHP/Perl/Python (and now also Ruby) programming languages. This part will be very useful to upload a large number of video files to the server, an important service that is designed to give users this project.

To enable this option you should access the "php.ini" that lies in the XAMPP folder (or lamp folder). In Linux, if you have installed the program to the folder root/opt will find this file specifically to the route:

opt/lamp/htdocs/etc/php.ini

"Php.ini" is the configuration of PHP, which can be accessed from a simple text editor (eg Texteditor). Specifically, you must change three parameters in this document for this project.

- `Max_execution_time = 3600` // parameter that defines the maximum run time r for each script in seconds. Is used when uploading a large file to the server ,after a time X , the transfer will not be canceled. It has been alleged that a file can't take more than 3600 seconds = 1 hour because of the bandwidth of the network of UPC (where it will load videos).
- `Upload_max_filesize = 1000m` // parameter that defines the maximum size PHP upload allowed. In this case becoming a 1000 Mb = 1 Gb as users who use this service will not be expected to use this size as for a large video.
- `Post_max_size = 1000m` // parameter that defines the maximum file size that the POST method of PHP code (which is used to develop the site.) In this case it is given a maximum of 1000 Mb = 1 Gb, as the value permit maximum file size in PHP.

3.3.2 PHP

PHP was the language chosen to build the web portal. It is needed for a dynamic page (for the management database) and XAMPP package which works with the PHP code. To view the results of work with PHP, you should put websites in the folder "htdocs" which is inside the XAMPP folder:

opt/lampp/htdocs/projecte/

For this project, specifically, we created a folder named "projecte" which keeps track of all the progress that is made. In this folder there is another called "videos" and "thumbnails" directories. These contain all the videos and each thumbnail of a video. As discussed above in the section "Creation & management of the database" data is classified into three video categories in folders according to its definition (High, Standard and low):

opt/lampp/htdocs/projecte/videos
&
opt/lampp/htdocs/projecte/thumbnails

Beside the two folders containing "projecte" are all the PHP files that have been used by this thesis. At this point I would like to refer to David Vera's thesis because in the next pages below we will explain the improvements to his project.

Figure 13 shows a diagram of all the PHP files in this improved project. Below represents all the PHP files from David Vera's project:

*Body.php – borrar.php – cerca.php – comprovamod.php – form1.php –
form2.php – form3.php – funciones.php – login.php – logout.php –
modificar.php – nosu.php – pujarnova.php – videos.php*

One the goals during this thesis was to improve David Vera’s project. This improvement is completed by adding some new services to the project. Each of the new services will be explained in the next chapter “Development”. Below you can find a structure of all the PHP files that this project contains. As you see have each of the service their on PHP files which contains the code that is written in PHP and HTML. Like already mentioned, the files will be located in the directory; *opt/lampp/htdocs/projecte/*

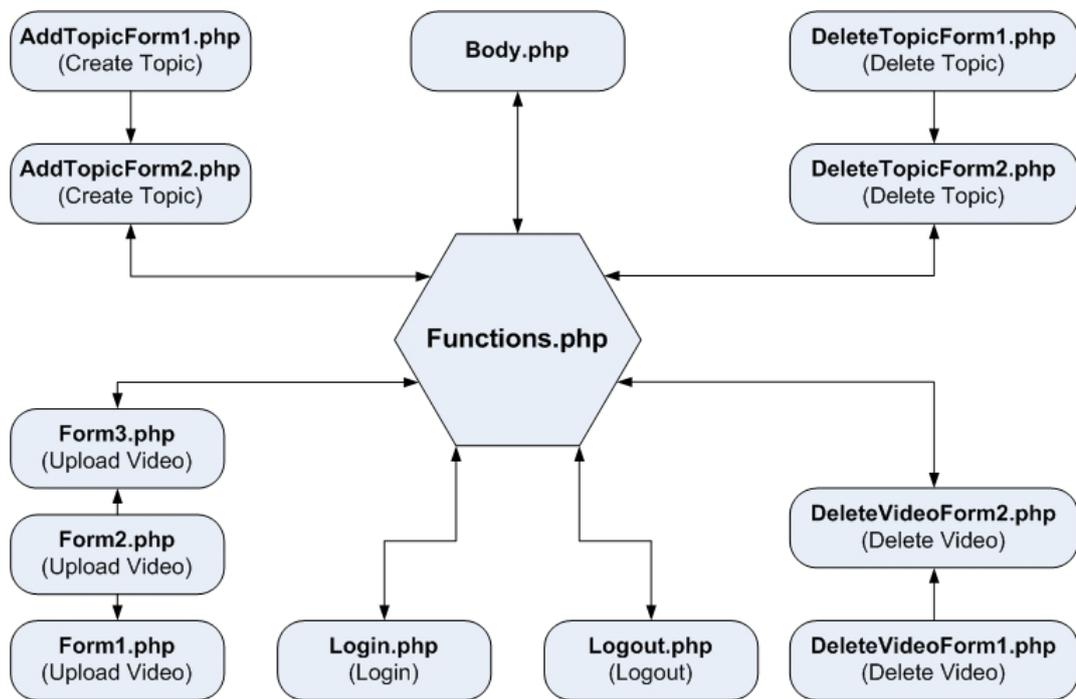


Figure 15: PHP-file structure.

This chapter explains in more details the working of the successful improvements during this project.

The first improvements were; implementing comments to each command, translating Catalan texts to English, deleting unnecessary PHP files and functions (e.g. change video,..) Next subsections explain each requirement for the PHP code.

1. Thumbnail generation and upload.

The goal of this issue was to generate a thumbnail from each uploaded video and locate them in the following directory:

opt/lampp/htdocs/projete/thumbnails

The basic code to generate a thumbnail is shown in figure 24 which also explains which is the inputfile and outgoing file.

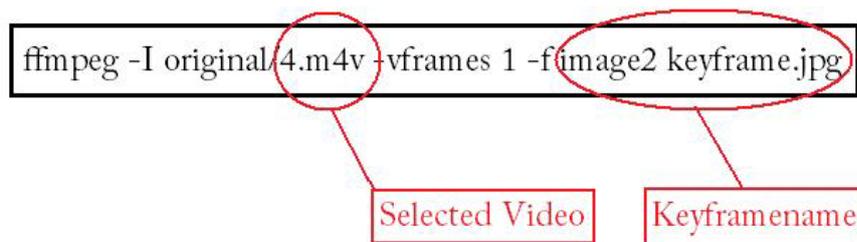


Figure 16: Thumbnail generation

Next you have to implement this PHP code in the form3.php file of this project. It is important is to use the right directory every time.

```
// path to ffmpeg
$ffmpeg = "/usr/local/bin/ffmpeg";
// the input video file
$video = "/opt/lampp/htdocs/projecte/videos/" . $def . "/" . $id . "." . $ext;
// where the keyframe/thumbnail will be saved
$thumbnail = "/opt/lampp/htdocs/projecte/thumbnails/" . $id . ".jpg";
// default time to get the keyframe/thumbnail
$second = 1;
// generate the thumbnail
$thumbcmd = "ffmpeg -i $video -deinterlace -an -ss $second -t 00:00:01 -r 1 -y -
vcodec mjpeg -f mjpeg -s 400x300 $thumbnail";

echo $thumbcmd;
exec($thumbcmd);
//Give the uploaded file the full permissions, so the file can be deleted by any user.
chmod ($thumbnail, 0777);
// done
echo 'done!';
```

Now that we saved each thumbnail in the right directory it is time to upload them on the server. This is a little part of html code in PHP code for the Low definition version. The size of the uploaded picture you can fix with “width = 200”.

```
echo("<TD>". $id . "</TD><TD>". $name . "</TD><TD>". $professor . "</TD><TD>". $data .
</TD><TD>". $desc . "</TD><TD> <p><a href= videos.php?name=". $id . "&url=
"Low">Play<a/></p><p><a href=". $urlLow . ">Download<a/></p></TD> <TD> No
</TD><TD> No </TD> <td>". $Topic . "</td><TD></TD> ");
```

The picture below gives you a complete view of the result.



Figure 17: Thumbnails

2. Automatic generation of different video scales

The next goal is to generate two video formats. Starting from a High definition video and go to a Standard and Low definition video.

The High definition video I get from our HDD lab camera, EVERIO JVC "GZ-HD7". With this camera you have the possibility to choose between two extensions for your video. The first is .TOD and the second .MOD. Those extensions are generated by the makers of the HDD JVC camera.

To create a Standard definition video I picked- which you can see in the photograph below- the "MPEG layer 1/2/3 audio" codec and the "MPEG-2" video codec. Follow the two frameworks on the figure 26 and the following type of container "TS" (Transport stream) has become.

| | PS | TS | Ogg | ASF | MP4 | MOV | MPMJPEG | Raw |
|-------------------------|----|----|-----|-----|-----|-----|----------|-----|
| Video formats | | | | | | | | |
| MPEG-1 video | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| MPEG-2 video | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| MPEG-4 video | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| DivX 1/2/3 video | ✗ | ✓1 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| WMV 1/2 | ✗ | ✓1 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| H/1 263 | ✗ | ✓1 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| MJPEG | ✗ | ✓1 | ✓ | ✓ | ✗ | ✗ | SVN only | ✗ |
| Theora | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| H.264/MPEG-4 AVC | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Audio formats | | | | | | | | |
| MPEG Layer 1/2/3 audio | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| AC3 (i.e. AS2) | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| MPEG-4 audio (i.e. AAC) | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Vorbis/Speex | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| FLAC | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |

Figure 18: Standard format¹¹.

¹¹ www.videolan.org/codecs

For the next conversion, the Low definition video, the audio codec "MPEG-4" and video codec "H.264/MPEG-4" are been used. Idem as by the Standard definition you can see in figure 27 which container you become; "MP4".

| | PS | TS | Ogg | ASF | MP4 | MOV | MPMPEG | Raw |
|-------------------------|----|----|-----|-----|-----|-----|----------|-----|
| Video formats | | | | | | | | |
| MPEG-1 video | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| MPEG-2 video | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| MPEG-4 video | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| DivX 1/2/3 video | ✗ | ✓1 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| WMV 1/2 | ✗ | ✓1 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| H/1 263 | ✗ | ✓1 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| MDPEG | ✗ | ✓1 | ✓ | ✓ | ✗ | ✗ | SVN only | ✗ |
| Theora | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| H.264/MPEG-4 AVC | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Audio formats | | | | | | | | |
| MPEG Layer 1/2/3 audio | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| AC3 (i.e. AS2) | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| MPEG-4 audio (i.e. AAC) | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Vorbis/Speex | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| FLAC | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |

Figure 19: Low format¹².

To make the conversion possible we have to use the program FFmpeg. You have to know that there are a lot of different ways to convert a video. Next command line is the instruction to convert a video . Under the code you find a list with an explanation of the parameters. By switching them you can change the parameters of the video.

```
ffmpeg -y -i mov60.TOD -acodec libfaac -ar 44100 -ab 96k -coder ac -me full -me range 16 -
sunq 5 -sc_threshold 40 -vcodec libx264 -s 1280x544 -b 1600k -cmp +chroma -partitions
+parti4x4+part8x8 -i_qfactor 0.71 -keyint_min 25 -b_strategy 1 -g 250 -r 20 low.mp4
```

List of parameters:

- y = overwrite the output file if it already exists
- i mov60.TOD = the input file
- acodec libfaac = using the aac audio codec
- ar 44100 = the audio sampling rate
- ab 96k = the audio bitrate
- vcodec libx264 = using the x264 codec
- s 1280x540 = the size of the output file
- b 1600k = the bitrate of the output video
- g 250 = frequency of keyframes
- r 20 = the frame rate
- low.mp4 = the output file

¹² www.videolan.org/codecs

By this generation of the three different definition videos we need a structure. To explain this more specifically, when you upload a high definition video this program will generate automatically an standard definition and an low definition video. Every definition has his own folder (directory, like I mentioned before). Now code has been created which can only generate a low definition video but not a high definition when you upload an standard definition video. And of course for uploading an low definition video this project copies the file only to a low definition folder without generating the high and standard definition video. Important to know; each file you upload you have to give read & write permissions cause otherwise you are not able to delete the video in the future. This can be done by using the next command line in a Terminal of Ubuntu.

```
chmod 777 [file_name]
```

The issue was to implement this into each code where we upload a file; creation of thumbnail, upload of videos, picture upload of a new topic,... There is one function in PHP who can do this but first you have to put the directoryname of the file in a variable.

```
$video = "/opt/lampp/htdocs/projecte/videos/high/34.TOD  
chmod ($video, 0777);
```

Below it is given an example of the generation of the different formats when you upload an High definition video.

```
<?php  
// If the video is completely uploaded to the server...  
if (is_uploaded_file($_HTTP_POST_FILES['tmp_file']['tmp_name'])) {  
  
    // the input video file  
    $video="/opt/lampp/htdocs/projecte/videos/".$def."/".$id.". ".$ext;  
    // path to ffmpeg  
    $ffmpeg="/usr/local/bin/ffmpeg";  
  
    // Convert video  
    // If the implemented video is a high definition version, then generate standard and low  
    definition  
    if($def == "high") {  
        // copy selected video and save it in the selected directory  
        copy($_HTTP_POST_FILES['tmp_file']['tmp_name'], $video );  
        //Give the uploaded file the full permissions, so the file can be deleted by every user.  
        chmod ($video, 0777);  
  
        // Where the Standard definition video will be saved  
        $standdefinition="/opt/lampp/htdocs/projecte/videos/standard/".$id.".mpg";  
        // generate standard definition video  
        $defstandcmd="$ffmpeg -i $video -acodec libmp3lame -vcodec mpeg2video  
        $standdefinition";
```

```
exec ($defstandcmd);
//Give the uploaded file the full permissions, so the file can be deleted by every user.
chmod ($standdefinition, 0777);

// Where the low definition video will be saved
$lowdefinition="/opt/lampp/htdocs/projecte/videos/low/"$id.".mp4";
// generate low definition video
$deflowcmd = "ffmpeg -i $video -acodec libfaac -vcodec libx264 $lowdefinition";
exec ($deflowcmd);
//Give the uploaded file the full permissions, so the file can be deleted by every user.
chmod ($lowdefinition, 0777);
}
}
?>
```

To give a registered user the possibility to upload those videos new PHP functions were implemented. The special part of this issue was to work with all the data of the database to select the right category. For example if you select the category “Courses” you will find in the next step all the possible topic under the selected category. This all by calling the data of each category from the database. By working on it this way you are able to add a new topic without changing something in the code.

3. Delete videos and thumbnails

David Vera's part was to give a user the possibility to delete an uploaded video. For this issue he has used the function "unlink". By working this way you delete the URL link on the database but not the video file in the directory in which it is located.

Below you see the improved code to delete the different version URLs (high, standard and low) as well as the files in their directories. As example you will find the code to delete a high definition video. When we select a high definition video to delete the system has also to delete the other two definitions (standard and low). The code is written so that it will delete the three definition videos as well as the thumbnail of the selected video. To delete a thumbnail we will work on the same way as deleting a video. Both, "thumbnailUrl" and thumbnail file in his directory, will be deleted.

```
// check if they are deleted otherwise show message of unsuccessful deleting
if ((unlink($DeleteHigh)) && (unlink($DeleteStd)) && (unlink($DeleteLow)) &&
(unlink($DeleteThumbnail))) {

    // Delete selected row in the database.
    MYSQL_QUERY("DELETE from videos WHERE id='$id'", $link);

    // show message of succesful deleted video
    ?> <font class="text"><p align="center" > The videos are deleted. </p><p
align="center"> <a href=" DeleteVideo1.php"> Delete another video. </a> </p>
</font></center> <?

}
else {

    // show message if the mission " delete videos" failed
    ?> <font class="text"><p align="center" >Sorry, failed in video deleting.<a href=
DeleteVideo1.php> Back to delete video. </a> </p> </font></center> <?

}
```

4. Add new Topic

This part is implemented in two PHP files , AddTopic.php and Add TopicForm2.php, and gives you the possibility to create an new topic under a selected category. The code is built in such a way that it will check- if you field in a new for a new topic, you did not use the same name of an already existed topic, and that there are not more than 5 topic under one category. Finally you can select a picture for the topic as well as implement a description.

To add an new topic we started with writing a html code to create an new window who ask you for the name of the new topic, description and finally the picture you want to upload for the topic. This code is written in the file AddTopic.php and is linked to AddTopicForm2.php. Means if you fill each field correctly he will go to the next PHP file. This file, AddTopicForm2.php, will add all metadata of the topic on the database and upload the picture in his directory:

/opt/lampp/htdocs/projecte/thumbnails/.

Html code to insert all the information about the new topic and the picture;

```
<!-- Define a form to generate a new topic -->
<form enctype="multipart/form-data" action="AddTopicForm2.php" method="POST"
name="ftp" id="form_ftp" >
<table align="center" class = 'tablar' >
<tr><td class = 'modo1' colspan=2 align="center" > Add new topic </td></tr>

<!-- Select a category for the topic -->
<tr><th > Select category:</th><td align="left" class = 'modo1'>
<select name="NewSelectedCategoryId"> <option value="1"> Courses <option value="2">
Thesis <option value="3"> Campus </select></td></tr>

<!-- Enter title and description -->
<tr><th > Title of new topic:</th><td align="left" class = 'modo1'>
<input name = "NewNameTopic" type="text" size="25" maxlength="20" value = ""></td></tr>
<tr><th > Topic description: </th> <td class = "modo1" >
<textarea name="TopicDescription" rows="7" cols="43" value= ""> </textarea> </td></tr>

<!-- Choose a file on disk as a picture -->
<tr> <th > Select picture:</th><td class = "modo1" >
<input name="userfile" type="file"/></td></tr>

<!-- Button to submit the data -->
<tr><td class="modo1"></td><td align="center" >
<input name="Submit" type="submit" class="modo1" id="upload" value="Create new topic"
/><INPUT type="reset"></td></tr>
</table></form>
```

Php code to add all metadata of the new topic on the database and upload the picture;

```
if(move_uploaded_file ($_FILES['userfile']['tmp_name'], $Picture)){

    //Give the uploaded file the full permissions, so the file can be deleted by every user.
    chmod ($Picture, 0777);

    // Update the data of the topic on the database
    MYSQL_QUERY("update Topic set NameTopic='$NewNameTopic',
    DescriptionTopic='$NewDescriptionTopic', ThumbnailUrl='$ThumbnailUrl' where
    CategoryId = '$NewSelectedCategoryId' and TopicId = '$TopicId'");

    ?> <font class="text"><p align="center" >New topic <?php echo $NewNameTopic; ?> has
    been created.<a href= AddTopic.php> Back to add new topic </a> </p> </font></center>
    <?
}
else {
    echo "Failed to upload file Contact Site admin to fix the problem";
}
```

5. Delete a topic

First you have to fill in the name of the topic you want to delete. When this is done it will check to see if the topic already exists. When it exists the next step is to delete the picture in its located directory and reset all the data of the topic on the database by overwriting it with a zero. This service contains two PHP files; DeleteTopic1.php and DeleteTopic2.php. The first file will ask for the name of the topic you want to delete so we can get the information about that topic of the database in the second PHP file. The PHP code is almost the same as the code for deleting a video. The differences are only the parameters.

This chapter represents the final result which are created during this thesis. The differences between this upgraded project and David Vera's project are; upload and deleting part for a video, Add and erase part for a topic under a category and finally the improvements on design.

1. Upload and delete video.

This topic will go about the services upload or delete a video and what are the possibilities in this functions.

1.1 Upload video.

One of the requirements in this project was to create an service that automatically generate three different versions of a video. The three cases are defined as; high definition (large screens), standard definition (traditional TV monitors) and low definition (mobile devices/ small screens). The next requirement was to generate automatically the thumbnails from a keyframe of the video that allow a more intuitive navigation though the video archive.

As explained the working and code in the chapter "Installation" it is time to talk about the procedure to upload a video.

This upload service contains three forms. By working on this way there is the possibility to check information of already existed video. This is important because two videos with the same Id are not allowed. The second check is when you selected a category to get all the available topics under that category and represent them in the second form.

First form is to enter the video name, select the category and select the definition of the file(video). Second form will ask to implement a description of the video , select one of the choices of the topics under your selected category (see first form) and finally there is a browse window to load up the file. The last form will generate the three definition versions as well as the thumbnail of the movie and show the message "The file has been uploaded successfully".

If there is an error between all those actions- for example; did not fill in a name, error in uploading the video,... - each error will be visible on the screen and there is always a possibility to return to the first form “Back to upload a video”.

Figure 20: First form upload video.

Figure 21: Second form upload video.

After an successful upload all the uploaded videos will be represented, in order of their Id number, under the Home tab. Each video shows his name, date, description, definitions urls, selected category, selected topic and finally the thumbnail of the video.

| Home | | | | | | | | | | |
|------|----------------|---------|------------|-------------------------------------|------------------|------------------|------------------|----------|---------------------|-----------|
| Id | Title of video | User | Date | Description | Low | Standard | High | Category | Topic | Thumbnail |
| 63 | culture day | laurens | 28-05-2009 | video of culture day | Play Download | Play Download | Play Download | Courses | Signals and Systems | |
| 56 | Video test | laurens | 26-05-2009 | Video made by Laurens and Alexander | Play Download | Play Download | Play Download | Thesis | IPTV | |
| 55 | test | laurens | 26-05-2009 | test video | Play Download | Play Download | Play Download | Courses | Machine learning | |

Figure 22: Home tab

1.2 Delete video:

To remove a video there is the delete video service. In this service all the definition will be erased from the disk as well as the thumbnail of the selected video. The delete service is based on two forms; DeleteVideo1.php and DeleteVideo2.php. The first form create a field to enter the id number of the supposed deleted video. Second form is almost the same as the second from of upload video. On this moment all the definitions as well as the thumbnail will be first deleted from their location on the disk and second all the information on the database will be erased.



| Delete Video | |
|--|---|
| ENTER THE ID OF THE VIDEO YOU WANT TO DELETE: | <input type="text"/> |
| | <input type="button" value="Delete video"/> |
| | <input type="button" value="Herinitialiseren"/> |

Figure 23: Delete video (form: DeleteVideo1.php).

2. Add and Erase Topic.

This topic will go about the services add or erase a topic on the database and what are the possibilities with this functions.

Let us begin with the whole category and topic structure that starts with the three fixed categories; “Courses”, “Thesis” and “Campus”. Each of them have the possibility to contain maximum five different topics. Figure below give an example of a whole Category/Topic structure.

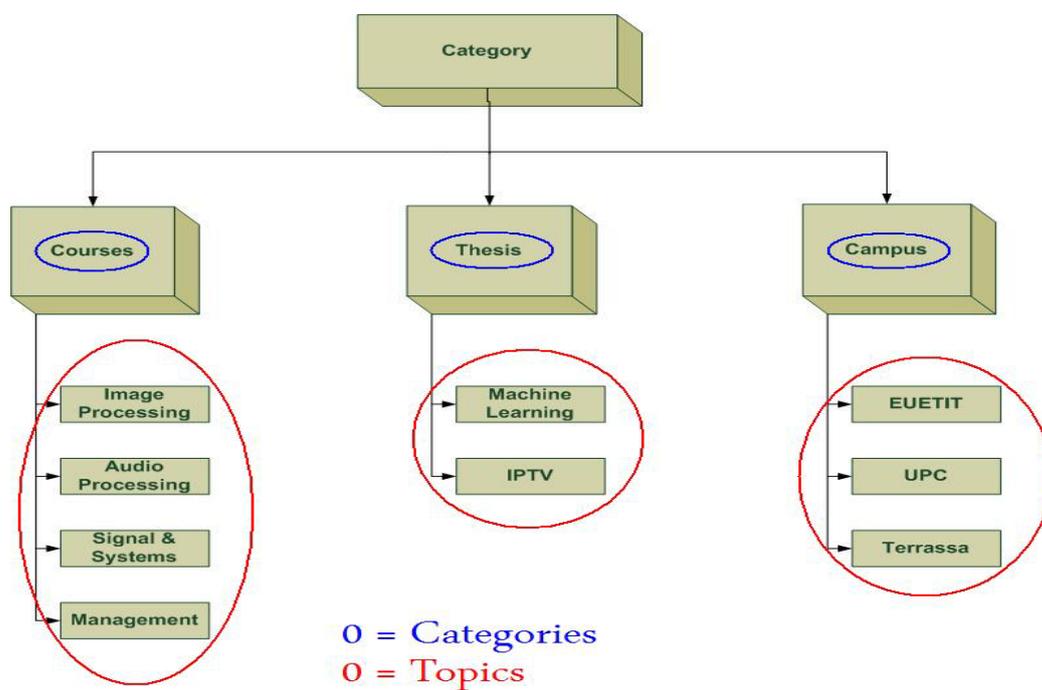


Figure 24: Example of a Category/Topic structure

2.1 Add Topic.

This service has two forms; AddTopic1.php and AddTopic2.php. Like the previous services the last form is to create or delete the necessary objects. So every information for a new topic has to be implemented in the first form. This starts with the new name of the topic, select under which category the new topic has to be located, add a description and finally there is a possibility to upload a picture that represent the topic. By uploading this picture the graphic layout of several interfaces will be nicer. For example; as new topic “Signal & Systems” the picture below satisfied the layout of this topic.

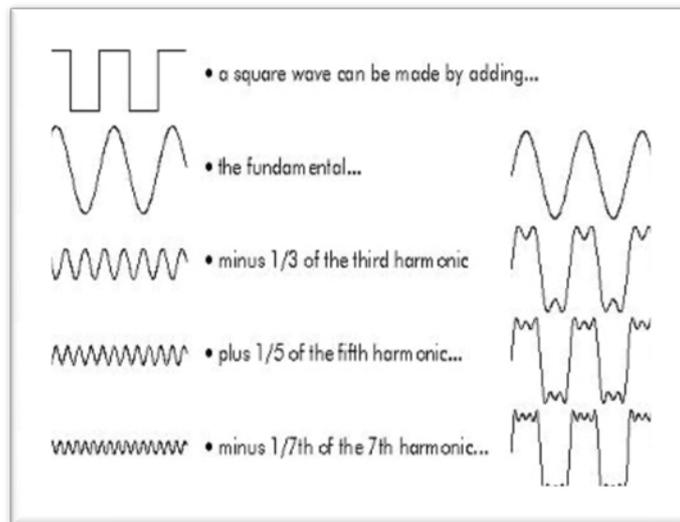


Figure 25: Example Topic picture.

By adding a new topic the system will always check if the topic, by name, already exist and check if the limit is not reached of the maximum topics under a category. Next figure is an example of the first form of the add topic service.

Figure 26: Add Topic (First form: Addtopic1.php).

2.2 Erase Topic.

This last selection of service gives the possibility to delete one topic. By erasing a topic all the data of the selected topic will be erased from the database and his existed picture will be deleted from his directory.

The principle is exactly the same as the delete video service. It exists two forms; DeleteTopic1.php and DeleteTopic2.php. The figure below is the layout of the first form who ask the name of the topic.



Figure 27: Delete Topic (First form: DeleteTopic1.php).

3. Improvements on design.

During this project the language has been changed from Catalan, that was written by David Vera, to English. The translation starts from the comments in the code to the error messages.

Beside the translation there is an upgraded version of David Vera's thesis, chapter "Study Cases". The name has been changed in "Document". This text file is regular to a help file. Each guest or user will find on the web interface a link to this document. Specific, this link will stand under the tab "About". In this tab there is a complete description about the web interface and the makers of this project.

Future Work and Conclusion

1. Future Work.

For the improvement of this project they could make the following changes and improvements:

- Content ingest from a client application. i.e. multiple ingest of videos.
- By uploading a video, generate several different thumbnails from a video and give the user the possibility to select one of the several thumbnails. E.g. The user is able to select which thumbnail represents the video in the archive.
- More tests on different Web browsers. For example; Internet Explorer, Konqueror (Linux), Google chrome, Opera,...
- Allow the use of an external user management. It could be share accounts, for example, the TSC Moodle.
- Improve the “Upload video” service by creating a system that is implemented in this service, called “Auto detect definition”. The goal is to recognize automatic the definition of the video the user wants to upload. In this moment the user have to do this manual.
- Implement a service where clients or users can give each uploaded video a rate. Create a new tab called “Top Rated” which represents a list of the videos with the highest rate. Next to “Top Rated” there is a possibility to create a service called “Most Viewed” which shows all the videos that has been watched most by clients in general.

2. Conclusion.

After analyzing the existed project of David Vera, we can assert that this improved project have solved some future work pointed by David Vera. Beside one require we successfully realized all of them during this thesis.

The final system provides a management interface for the system administrator to easily upload videos and edit their metadata. There is also the possibility to consume the video content from a diversity of user interfaces, such as a web browser or a Set Top Box.

In this video service there is an automatically generating different version of the content adapted to the consumption device. Three cases are defined: high definition (large screens), standard definition (traditional TV monitors) and mobile devices (small screens). The improvements also generate automatically the thumbnails from a keyframe of the video that allow a more intuitive navigation though the video archive.

Important to know, one of the requirements was to use only free software and like you can notice form the title of this thesis is this project only based on free software. This means that everybody is able to install every program that is been used in this project. The free programs that we used are for example; Ubuntu from Linux, Firefox Mozilla, XAMPP, FFmpeg and the video player VLC.

During this thesis my personal opinion about Video on Demand services has changed a lot. Comparing with my background studies “Electronics Engineering” I can proudly make the conclusion that I am able to program in PHP and HTML language. Last, but not least, was the most interesting part of this thesis using the program “FFmpeg” to generate the three different video definitions as well as the thumbnail. This was the first time that I came in contact with video formats which gave me a great experience with the different available video or audio codecs.

Appendix: Documentation

This document “Documentation” gives a global view of the whole service. It is made as a guide for a user or a client that leads them thru this project. There is also a direct link on the website who refer to this document. Clients and users can find the link under the tab “About” which give them information about the history and makers of this service.

The service has two types of users: visitors and managers or administrators. They are classified as they have the privileges to access the data.

A.1 Visitors

Visitors are all these people who can enjoy the services that have nothing to do with the modification of the metadata from the database. The different tasks can performed are:

A.1.1 Navigation

As explained in previous sections, the service that this project be is achieved through a web portal. This is stored in a computer that located in the laboratory of the Department of Signal Theory and Communications Terrassa, which performs the task of server.

After entering the url into a web browser you will see the following interface:



Figure 28: Web interface.

Below we explain the design of this figure and each tabs you find.

The portal is divided into three sections, the top, the middle and bottom. At the top is the logo of the TSC Terrassa and a search engine. The search engine looks for all the videos based on their title. Respond in the bottom section with a search that displays all the video titles which begin with characters you've typed in the text field.

The average index contains Portal, represented by the tabs:



Figure 29: Tabs design web interface

- Home: Under this tab you will find all the uploaded video with their information. The contents of the information is; Title of the video, User, Date, Description, the three Url's of the videos (low, standard and high), Category, Topic and finally the thumbnail of each video.

| Home | | | | | | | | | | |
|------|----------------|---------|------------|-------------------------------------|--|--|--|----------|---------------------|---|
| Id | Title of video | User | Date | Description | Low | Standard | High | Category | Topic | Thumbnail |
| 63 | culture day | laurens | 28-05-2009 | video of culture day | Play Download | Play Download | Play Download | Courses | Signals and Systems |  |
| 56 | Video test | laurens | 26-05-2009 | Video made by Laurens and Alexander | Play Download | Play Download | Play Download | Thesis | IPTV |  |
| 55 | test | laurens | 26-05-2009 | test video | Play Download | Play Download | Play Download | Courses | Machine learning |  |

Figure 30: Tab Home.

- Courses, Thesis, Campus: Those are the three different categories. Under each category tab you will see all the uploaded videos of the selected category. The information you will find are; Title of the video, User, Date, Different Url's, the topic name and finally the thumbnail of the video.

| Courses | | | | | | | | |
|---------|----------------|---------|------------|--|--|--|---------------------|---|
| Id | Title of video | User | Date | Low | Standard | High | Topic | Thumbnail |
| 63 | culture day | laurens | 28-05-2009 | Play Download | Play Download | Play Download | Signals and Systems |  |
| 55 | test | laurens | 26-05-2009 | Play Download | Play Download | Play Download | Machine learning |  |

Figure 31: Tab Courses.

- Webcam: Here you can see the result of the project by Angela Abad (Servers running audio and video based formats and free software) which is playing a stream of video captured from a webcam transmitted in real time.
- Administration: section that users may enter type "Managers."
- Credits: presented in this section, the creator of this site and give thanks former of which designers could extract information from their jobs.

A.1.2 Video on demand

To play videos you just have to click on the link "Play" that exists in each row of the tables of the videos that you can find under the tabs “Home”, “Courses”, “Thesis”, “Campus” or do a search function with the search engine.

| Home | | | | | | | | | | |
|------|----------------|---------|------------|-------------------------------------|--|--|--|----------|---------------------|---|
| Id | Title of video | User | Date | Description | Low | Standard | High | Category | Topic | Thumbnail |
| 63 | culture day | laurens | 28-05-2009 | video of culture day | Play Download | Play Download | Play Download | Courses | Signals and Systems |  |
| 56 | Video test | laurens | 26-05-2009 | Video made by Laurens and Alexander | Play Download | Play Download | Play Download | Thesis | IPTV |  |
| 55 | test | laurens | 26-05-2009 | test video | Play Download | Play Download | Play Download | Courses | Machine learning |  |

Figure 32: Tab Home.

When you click on the link “Play” there will find the window below.



Figure 33: Tab video.

The table presents the video's title as a header. These elements are:

- The connector of VLC (black part of the table) that will be displayed on the Video.
- The description of the video corresponds to the "Description" of the table videos database.
- Controls of the video on demand are at the bottom: "Play" (play video), "Pause" (break the transmission of video and if you to push the play continues on the transmission was paused), Stop (stops broadcast video but can not continue where it has stopped) "Fullscreen" (full screen) and "Mute" (silence).

Under the table above can be found:

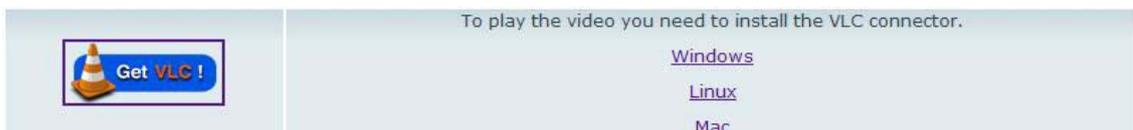


Figure 34: VLC connector.

This table can find the links that refer to the discharges of VLC connector for the three operating systems Windows, Linux and Mac.

A.2 Managers / Administrator

These are users who have assigned a user name and password for the administrator. This data is stored in the table "users" within the database "Archives". A part of the services that users can use the "Visitor" the "Managers" can perform other functions. The "managers" have to access different services by clicking on the "Administrator" and fill in a form of access.



The image shows a login form titled "Administrator". It contains two input fields: "USER:" and "PASSWORD:". Below these fields is a "Login" button.

Figure 35: Login.

After completing the login form will be a table for the selection of service (left table):



The image shows a menu titled "You are logged in as : laurens". The menu items are: "Upload video", "Delete video", "Add new topic", "Delete topic", and "Exit".

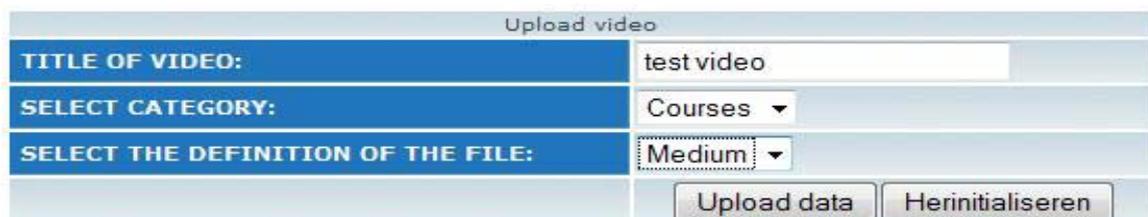
Figure 36: Services.

The table with the selection of service is the table that appears on every page of the service management. This provides access to a service without need to return to a menu. This table is located at top right corner of the lower section of the website.

We will explain each selection of service and starts with the first one, Upload video.

A.2.1 Upload video.

By using this service you must click on the "Upload Video" . After that you will see the next window that you have to fill in with all the necessary information. As example I insert the name "test video" in the table "Title of video".



The image shows a form titled "Upload video". It contains three input fields: "TITLE OF VIDEO:" with the value "test video", "SELECT CATEGORY:" with the value "Courses", and "SELECT THE DEFINITION OF THE FILE:" with the value "Medium". Below these fields are two buttons: "Upload data" and "Herinitialiseren".

Figure 37: Upload video-Form1.

The "Manager" will fill in the three fields, that can be seen in the figure above, to move to the following form. If the "Video Title" is not populated, you can not transfer a file.

It must be said that the forms are divided into two steps because it was initially consider restrict upload a video if the user enters a title and video existed in the database. Currently no checking whether the new title already exists two or more videos can have the same name but will never have the same id. The next step is to fill in the "DESCRIPTION OF THE VIDEO" and indicate the file to upload:



| You selected as Category: Courses | |
|-----------------------------------|-----------------------------------|
| SELECT TOPIC: | Content delivery and management ▾ |
| VIDEO DESCRIPTION: | This is a test video. |
| SELECT FILE: | <input type="text"/> Bladeren... |
| Upload video Herinitialiseren | |

Figure 38: Upload video-Form2.

The service has been enabled to upload a maximum of a 1Gb memory. We could see that, logically, the requires a transmission time. In the time the file is uploading you can't click on anther service. This is very important, cause otherwise the transfer will be closed. Depending of the transfer, correct transfer, failed transfer, wrong parameter you will get an message when the video is completely uploaded.

A.2.2 Delete video.

To remove a video you must click on "Delete Video" which can be seen in the selection table of service.

The first form will ask you for the id number of the video you want to delete.



Figure 39: Delete video.

If you insert a right id number the video will be deleted from his located directory and as well as thumbnail. Next all the data of the video on the database will be erased.

A.2.3 Add new Topic.

This will give you the opportunity to create a new topic under a selected category. The categories are fixed, so you can choose about three categories which are "Courses", "Thesis", "Campus". By adding a new topic you have to fill in one form with the necessary information.

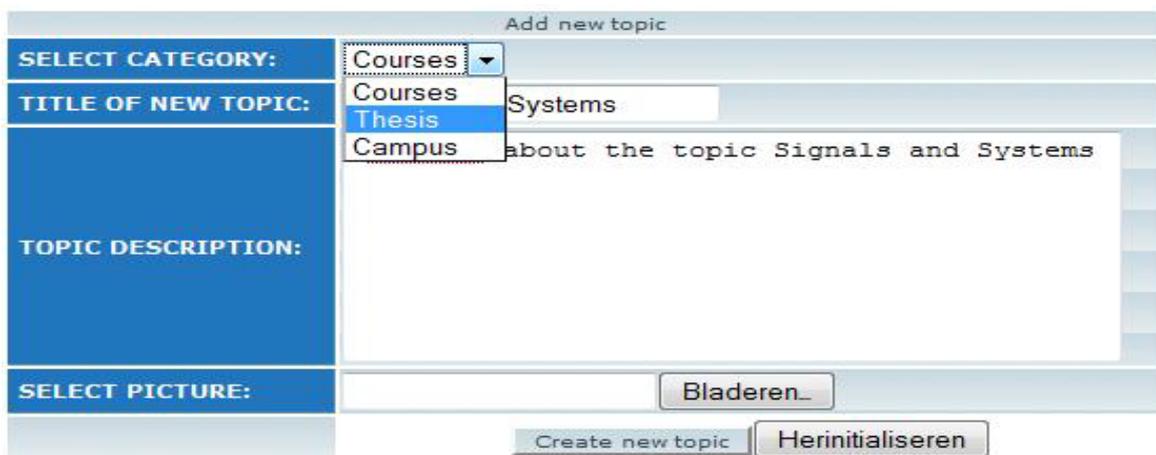


Figure 40: Add new Topic.

In the figure above we show you that you have the possibility to select first one of the categories.

Next you have to insert a name for the topic who you want to create. After that you have to fill in a Description of the topic and the finally step is to select an picture of the topic. For example, you can upload for the topic “Signals and Systems” the next picture:

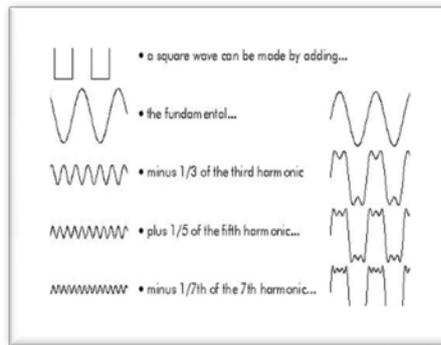


Figure 41: Example Topic picture.

If you followed every step you will get a message with the description of the correct creation.

A.2.4 Delete Topic.

In this last selection of service you have to possibility to delete one topic. This because we fixed the total topic of each category to five. I.e. you can create maximum five topic under one category. To delete a topic you have to fill in the correct name of the topic. Like you can see in the figure below. By deleting a topic you will erase all the data of the topic on the database and delete the picture of the topic on his located directory.

Figure 42: Delete Topic.

Appendix: Database Figures

Figure 43: Complete database structure.

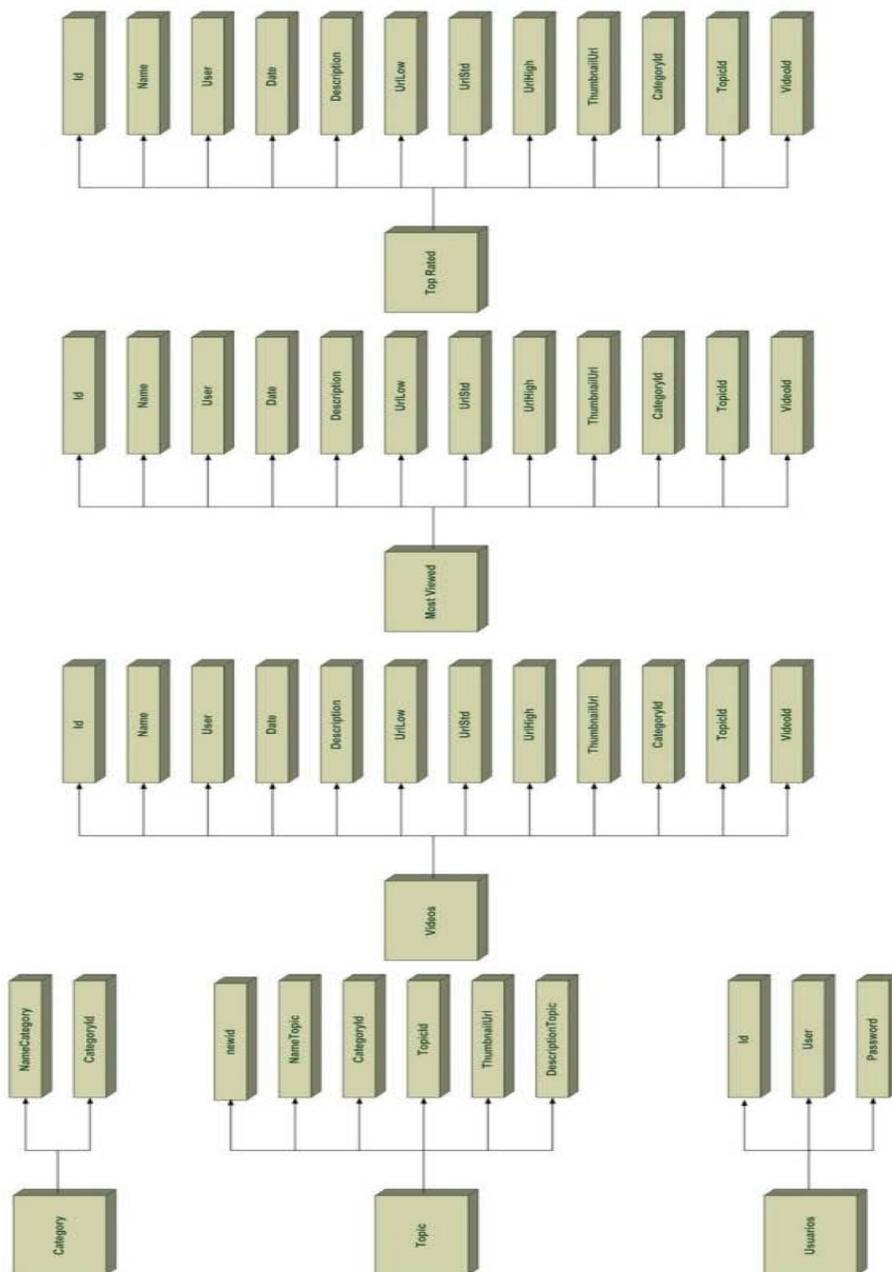


Figure 44: Table “users”.

| | Field | Type | Collation | Attributes | Null | Default | Extra | Action |
|--------------------------|-----------------|--------------|-----------------|------------|------|---------|----------------|---|
| <input type="checkbox"/> | id | bigint(100) | | | No | None | auto_increment |        |
| <input type="checkbox"/> | gestor | varchar(100) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | password | varchar(100) | utf8_unicode_ci | | No | None | |        |

Figure 45: Table “videos”.

| | Field | Type | Collation | Attributes | Null | Default | Extra | Action |
|--------------------------|---------------------|--------------|-------------------|------------|------|---------|----------------|---|
| <input type="checkbox"/> | id | bigint(100) | | | No | None | auto_increment |        |
| <input type="checkbox"/> | name | varchar(100) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | user | varchar(100) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | date | varchar(100) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | Description | varchar(500) | latin1_swedish_ci | | No | None | |        |
| <input type="checkbox"/> | urlLow | varchar(100) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | urlStd | varchar(100) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | urlHigh | varchar(100) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | ThumbnailUrl | varchar(100) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | NameCategory | varchar(40) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | NameTopic | varchar(40) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | Videoid | int(50) | | | No | None | |        |

Figure 46: Table “Category”.

| | Field | Type | Collation | Attributes | Null | Default | Extra | Action |
|--------------------------|---------------------|-------------|-----------------|------------|------|---------|-------|---|
| <input type="checkbox"/> | NameCategory | varchar(40) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | CategoryId | int(11) | | | No | None | |        |

Figure 47: Table “Topic”.

| | Field | Type | Collation | Attributes | Null | Default | Extra | Action |
|--------------------------|-------------------------|--------------|-----------------|------------|------|---------|----------------|---|
| <input type="checkbox"/> | newid | bigint(100) | | | No | None | auto_increment |        |
| <input type="checkbox"/> | NameTopic | varchar(40) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | CategoryId | int(11) | | | No | None | |        |
| <input type="checkbox"/> | TopicId | int(50) | | | No | None | |        |
| <input type="checkbox"/> | ThumbnailUrl | varchar(100) | utf8_unicode_ci | | No | None | |        |
| <input type="checkbox"/> | DescriptionTopic | varchar(100) | utf8_unicode_ci | | No | None | |        |

Figure 48: Table “Top Rated”.

| | Field | Type | Collation | Attributes | Null | Default | Extra | Action | | | | | | |
|--------------------------|---------------------|--------------|-------------------|------------|------|---------|----------------|--------|--|--|--|--|--|--|
| <input type="checkbox"/> | id | bigint(100) | | | No | None | auto_increment | | | | | | | |
| <input type="checkbox"/> | name | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | user | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | date | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | Description | varchar(500) | latin1_swedish_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | urlLow | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | urlStd | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | urlHigh | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | ThumbnailUrl | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | NameCategory | varchar(40) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | NameTopic | varchar(40) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | Videoid | int(50) | | | No | None | | | | | | | | |

Figure 49: Table “Most Viewed”.

| | Field | Type | Collation | Attributes | Null | Default | Extra | Action | | | | | | |
|--------------------------|---------------------|--------------|-------------------|------------|------|---------|----------------|--------|--|--|--|--|--|--|
| <input type="checkbox"/> | id | bigint(100) | | | No | None | auto_increment | | | | | | | |
| <input type="checkbox"/> | name | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | user | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | date | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | Description | varchar(500) | latin1_swedish_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | urlLow | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | urlStd | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | urlHigh | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | ThumbnailUrl | varchar(100) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | NameCategory | varchar(40) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | NameTopic | varchar(40) | utf8_unicode_ci | | No | None | | | | | | | | |
| <input type="checkbox"/> | Videoid | int(50) | | | No | None | | | | | | | | |

Appendix: List of Figures

| | |
|--|----|
| <i>Figure 1: Complete view of Video on Demand.</i> | 5 |
| <i>Figure 2: Video file.</i> | 18 |
| <i>Figure 3: Format table.</i> | 18 |
| <i>Figure 4: Container Multimedia</i> | 22 |
| <i>Figure 5: Plug-in framework</i> | 29 |
| <i>Figure 6: Software structure.</i> | 36 |
| <i>Figure 7: Start page of Xampp.</i> | 38 |
| <i>Figure 8: Interface of Xampp.</i> | 43 |
| <i>Figure 9: Interface of MySQL server</i> | 44 |
| <i>Figure 10: Privileges – User creation.</i> | 45 |
| <i>Figure 11: Privileges - Add new User.</i> | 45 |
| <i>Figure 12: Privileges – Parameters.</i> | 46 |
| <i>Figure 13: Management process on the database.</i> | 47 |
| <i>Figure 14: Attributes of the tables.</i> | 47 |
| <i>Figure 15: PHP-file structure.</i> | 54 |
| <i>Figure 16: Thumbnail generation</i> | 56 |
| <i>Figure 17: Thumbnails</i> | 57 |
| <i>Figure 18: Standard format.</i> | 58 |
| <i>Figure 19: Low format.</i> | 59 |
| <i>Figure 20: First form upload video.</i> | 67 |
| <i>Figure 21: Second form upload video.</i> | 67 |
| <i>Figure 22: Home tab.</i> | 67 |
| <i>Figure 23: Delete video (form: DeleteVideo1.php).</i> | 68 |
| <i>Figure 24: Example of a Category/Topic structure</i> | 69 |
| <i>Figure 25: Example Topic picture.</i> | 70 |
| <i>Figure 26: Add Topic (First form: Addtopic1.php).</i> | 70 |
| <i>Figure 27: Delete Topic (First form: DeleteTopic1.php).</i> | 71 |
| <i>Figure 28: Web interface.</i> | 76 |
| <i>Figure 29: Tabs design web interface</i> | 77 |
| <i>Figure 30: Tab Home.</i> | 77 |
| <i>Figure 31: Tab Courses.</i> | 78 |
| <i>Figure 32: Tab Home.</i> | 79 |
| <i>Figure 33: Tab video.</i> | 80 |
| <i>Figure 34: VLC connector.</i> | 80 |
| <i>Figure 35: Login.</i> | 81 |
| <i>Figure 36: Services.</i> | 81 |

Figure 37: Upload video-Form1. 81
Figure 38: Upload video-Form2. 82
Figure 39: Delete video...... 83
Figure 40: Add new Topic...... 83
Figure 41: Example Topic picture. 84
Figure 42: Delete Topic. 84
Figure 43: Complete database structure. 85
Figure 44: Table “users”..... 86
Figure 45: Table “videos”..... 86
Figure 46: Table “Category”..... 86
Figure 47: Table “Topic”..... 86
Figure 48: Table “Top Rated”..... 87
Figure 49: Table “Most Viewed”..... 87

Appendix: Bibliography

- [1] IETF IGMP v2,
“<http://tools.ietf.org/html/rfc2236>”.
- [2] IETF RTSP,
“<http://tools.ietf.org/html/rfc2326>”.
- [3] Metadata definition,
“<http://en.wikipedia.org/wiki/Metadata>”.
- [4] Xiph.org foundation,
“<http://www.xiph.org/>”.
- [5] David Vera (2007). *Servidor Web de Video sota Demand Basat en el Videolan*.
PFC carried at UPC-EUETIT.
- [6] Emili Bonilla (2007). *Real-Time Metadata for IPTV Systems*.
PFC carried at UPC-EUETIT.
- [7] Videolan VLC
“<http://www.videolan.org>”
- [8] Apache Commons Configuration
“<http://www.commons.apache.org/configuration>”
- [9] Wes Simpson (2005). *Video over IP: A Practical Guide to Technology and Applications*, Focal Press.
- [10] Emili Bonilla (2007). *Real-Time Metadata for IPTV Systems : State of the Art*.
PFC carried at UPC-EUETIT.
- [11] Dublin Core
“<http://www.dublincore.org>”
- [12] iPhone specifications
“<http://www.apple.com/iphone/specs.html>”

- [13] T-Mobile G1 specifications
“<http://www.htc.com/www/product/g1/specification.html>”

- [14] Andriod
“<http://www.andriod.com>”

- [15] Apache
“<http://www.apache.org>”

- [16] FFmpeg
“<http://www.ffmpeg.org>”

- [17] Alexander Schmidt (2009). *Graphical User Interface for Video on Demand Navigation from an IPTV Set Top Box*. PFC carried at UPC-EUETIT.

- [18] David Vera (2007). *Servidor Web de Video sota Demand Basat en el Videolan : Configuració de XAMPP*. PFC carried at UPC-EUETIT.

Chapter E

Appendix: Acronyms

A

AAC.....Advanced Audio Coding
ATSC.....Advanced Television Systems Committee
AVC.....Advanced Video Coding

B

BTV.....Broadcast TV

C

CP.....Content Provider

D

DBMS.....Database Management System
DD.....Dobly Digital
DTV.....Digital Television
DVB.....Digital Video Broadcast

E

EUETIT.....University of Industrial Engineering of Terrassa

F

FCC.....Federal Communications Commission
FLAC.....Free Lossless Audio Codec

G

GNU GPL.....GNU General Public License

H

HDD.....High Definition Display
HDTV.....High Definition Television

I

IGMP..... Internet Group Management Protocol
ITU-TInternational Telecommunication Union

M

MPEG..... Moving Picture Expert Group

P

PS.....Program Stream

R

RDF Resource Description Framework
RTP Real Time Protocol
RTSPReal Time Streaming Protocol

S

SAOL.....Structured Audio Orchestra Language
SDTV..... Standard Definition Television
SMPTE..... Society of Motion Picture and Television Engineers
SP Service Provider
STBSet Top Box

T

TSTransport Stream
TSCTheory and Communications

U

UDP..... User Datagram Protocol