

# **A User-Centered Open Source Video Browser**

By

**Sharaelna Binti A. Rahim**

Dissertation submitted in partial fulfillment of  
The requirements for the  
Bachelor of Technology (Hons)  
(Business Information System)

JULY 2006

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

t  
LB  
1028.75

.S571  
2006

1) Interactive Video  
2) IT IS -- Thesis

**CERTIFICATION OF APPROVAL**

**A User-Centered Open Source Video Browser**

By

Sharaelna Binti A. Rahim

A project dissertation submitted to the  
Technology Program

Universiti Teknologi PETRONAS

In partial fulfillment of the requirement for the  
BACHELOR OF TECHNOLOGY (Hons)  
(BUSINESS INFORMATION SYSTEM)

Approved by



(Mr. Nordin Bin Zakaria)

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

July 2006

## **CERTIFICATION OF ORIGINALITY**

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and the original work contained herein have not been undertaken or done by unspecified sources or persons.



---

**SHARAE LNA BINTI A. RAHIM**

# TABLE OF CONTENTS

## CERTIFICATION OF ORIGINALITY

LIST OF TABLES.....	i
LIST OF FIGURES.....	i
ABSTRACT.....	ii
ACKNOWLEDGEMENT.....	iii

## CHAPTER 1: INTRODUCTION

1 Background of Study .....	1
1.1 Problem Statement.....	2
1.1.1 How do users know what is there?.....	2
1.1.2 How users find anything there? .....	3
1.1.3 How can users find what they want?.....	3
1.2 Objectives and Scope of Study .....	3
1.2.1 Objectives.....	3
1.2.2 Scope of the Study.....	3

## CHAPTER 2: LITERATURE REVIEW AND THEORY

2 THE CONCEPT OF VIDEO BROWSING .....	5
2.1 Video Browsing Techniques .....	6
2.2 Playback System on Video Browser.....	7
2.3 Multimedia Database System .....	7
2.4 Keyframe-Based Video Browsing .....	8

## CHAPTER 3: METHODOLOGY

3 PROJECT DEVELOPMENT.....	13
3.1 Setting the Configuration Files for PHP.....	14
3.2 Database Design .....	15
3.3 Interface Design .....	18

3.3.1	Uploading Video Files Page .....	18
3.3.2	Interface of playing the video .....	21
3.3.3	Interface of searching the video.....	22

**CHAPTER 4: RESULT AND DISCUSSION**

4	DISCUSSIONS and EXPECTED RESULTS .....	23
4.1	Solutions to the Problem Statements .....	23
4.1.1	How do users know what is there?.....	23
4.1.2	How users find anything there?.....	24
4.1.3	How can users find what they want? .....	25
4.2	Results .....	25

**CHAPTER 5: CONCLUSION AND RECOMMENDATION**

5.1	Lesson Learnt.....	28
5.1.1	Gaining New Knowledge .....	28
5.1.2	Improve Programming Skills.....	28
5.2	Recommendations or Suggestion .....	29
5.2.1	Initial Difficulty in Understanding How to Use .....	29
5.2.2	Questions on How Well Keyframes Represent Content .....	29
5.3	Area of Further Study .....	30

<b>BIBLIOGRAPHY.....</b>	<b>31</b>
<b>REFERENCES .....</b>	<b>31</b>
<b>APPENDICES.....</b>	<b>32</b>

## LIST OF TABLES

Table 4.1: Comparative findings of some existing video browsers.....	26
--	----

## LIST OF FIGURES

Figure 2.1: Multimedia Database Systems.....	8
Figure 2.2: Video Abstraction Levels.....	9
Figure 2.3: Keyframes selected from video to be displayed on a browsing interface.....	10
Figure 3.1 (a): Before – System Default php.ini.....	14
Figure 3.1 (b): After – Setting up the php.ini to allow the video to be uploaded.....	14
Figure 3.2: Attributes that will describe the video to be stored into database.....	15
Figure 3.3 (a): Creating the folders to store the video clips (upload folder).....	17
Figure 3.3 (b): All the uploaded video files are stored inside the upload folder.....	17
Figure 3.4: Interface to upload the video into the database.....	18
Figure 3.5: Interface to display and play the video clip based on user’s preferences.....	21
Figure 3.6: Interface to search video based on the keyword entered by the user.....	22
Figure 4.1: An example of video classification from Grouper.com.....	23
Figure 4.2: Enables the user to choose the Category (i.e. Radiobutton - Music or Comedy) that corresponds to their uploaded video.....	24

## **ABSTRACT**

The main focus of this paper is developing a video browser in a basis concept of User Interface Software Technology (UIST). The study presents a framework for designing video content browser in a keyframe-based that is used such as in digital video libraries or streaming media application. It includes the process of designing and developing the video browser for an enhancement of the existing video browsers such as Fischlar System, Open Video Project, Google Video, Informedia System, and YouTube to provide an easy to use system of video browsing. The comparative findings of existing system technologies will be able to investigate and identify the effects of providing context and interactivity that are superior and support the video browser system. Also included is the identification of problems in video browsing from the user's perspectives.

## **ACKNOWLEDGEMENT**

The support of my beloved family, FYP Supervisor Mr. Nordin Bin Zakaria, and enthusiastic friends are gratefully acknowledged. They are always there for me from the beginning of the project until the implementation stage. They had given their advice, opinion, and full support to give self motivation to me so that I will be able to finish up developing the system whenever I feel like I don't have the ability to do so.

A million thanks to them that keep on supporting and helping me in making the project successful. Thank you very much.



# **CHAPTER 1**

## **INTRODUCTION**

### **1 BACKGROUND OF STUDY**

The main concept of the study is a user-centered open source video browser. The study is to investigate the effects of providing the context (i.e. the use of video information) and interactivity (i.e. user interface of video browser) which are superior for both performance and user perceptions. It is a representation of video browser on video segmentation that is dividing the video stream into a set of meaningful segments or shots. The video data is displayed into keyframes to allow the user to browse their desired video content. Most of the background study is based on the research done by Hyowon Lee and Alan F. Smeaton [3].

Besides, the study will compare or review the existing video browsers in terms of weaknesses and strength including its context and interactivity based on users' perspectives to obtain new interface ideas in a more systematic way. The comparative of existing systems include Fischlar System, Open Video Project, Google Video, Infromedia System, and YouTube. Obtaining usage information from these systems allow developing some of the desirable features in future interfaces to digital video libraries.

## **1.1 Problem Statement**

The users may retrieve video information from any video storage without having any knowledge of how and where all the data objects can be found. The issue is where the users should know the allocation of the video in the storage area which definitely difficult to achieve the result because of a lot of arguments involved such as security issue and database performance issue.

Besides, the accuracy video content retrieval based on the user's query preferences also being put into consideration since the browsing is a term used to describe many types of user's behavior. For example, the user searches the video collection through video browser by beginning to enter a set of search terms or by selecting a portion of the collection to browse. Then, the system will respond and display a list of videos corresponding to the user's query or selection. At this point, the user must browse through the list to select their desired video that they would like to view or download.

Based on the research, there are three (3) common questions that came out from the users which should be analyzed prior to developing the video browser. These common problems should not be neglected in order to assure a user-friendly user interface and reliability of the system. All three problems are stated as follows:

### **1.1.1 How do users know what is there?**

Users need to gain familiarity with browsing the video information throughout the entire video database. In other words, the users need "recognition" instead of "recall" for the video content which concerns the query tools inside the user interface and efficiency of Video Database Management System (VDBMS). But, it is not easy for the users to learn in browsing the video content at video browser by viewing one shot at a time. Conversational method can be used as an alternative of browsing whereby the method is similar to the librarian who can show the user the content through a collection of stories.

### **1.1.2 How users find anything there?**

The issue in this problem is related to the neglecting user's familiarity of the video browser interface design. In other words, the video browser should not have to be complex and put many widgets or multimedia elements. The purpose is to avoid user's difficulties and confusion when accessing the browser, thus provide the user with easy to use of the system.

### **1.1.3 How can users find what they want?**

As video is a complex data object, finding the appropriate shot or keyframe might have many combinations of interaction modes.

## **1.2 Objectives and Scope of Study**

### **1.2.1 Objectives**

- Understand the concept of video browser.
- Identify the problems related to the video browsing application.
- Understand video indexing.
- Analyze and identify different types of video content browsing.
- Compare existing browser interfaces for video browser improvement.

### **1.2.2 Scope of the Study**

The scope of study encompasses the basis concept of user-centered open source video browser. The study presents a framework for designing video content browser that is based on keyframes basis that is used in digital video libraries or streaming media application. Besides, it includes the process of designing and developing the video browser for enhancement of the existing video browsers to provide an easy to use of system. The enhancement will refer to the comparative findings from the existing video browsers: Fischlar System, Open Video Project, Google Video, Informedia System, and YouTube. The findings will be able to investigate and

identify the effects of providing context and interactivity that are superior and support video browser system.

Furthermore, the scope of study involves identification of the problems related to the video browser applications such as digital video library and streaming media application. The identified problems are referred to users' perspectives because they are the end-users who will use the system.

## **CHAPTER 2**

### **LITERATURE REVIEW and/or THEORY**

#### **2 THE CONCEPT OF VIDEO BROWSING**

According to M. Wildemuth, Russell, Ward, Marchionini, and Sanghee Oh (2006), browsing is a term that is used to describe many types of behavior. The scope encompasses an entire information-seeking episode which can be defined as behavior that occurs as a person is scanning the list of results retrieved from an information system. This type of browsing is often referred to as scanning [5].

Generally speaking, browsing is the traditional method of finding an image in a collection when the images have been manually sorted or categorized. However, it has been shown that browsing is an effective method even when the browsing structure has been automatically generated [1].

Therefore, video browser is developed that will apply with the concepts stated earlier and methodologies that already gathered from the research done. The system is more focusing on the user interface of video representation that allows the user to browse video content via sets of keyframes extracted by the system [3].

Video browser is one of the broad areas of digital video which is designed for video content browsers based on browsing keyframes. The consideration involves human computer interaction (HCI) and video browsing usability plus with its system environment.

There are three characteristics identified by Newman [4] in order to implement the system which listed as in the next page:

- i) *The object itself* – video as composite sequence which comprises of video, audio and text, thus with scenes and frames.
- ii) *The characteristics of the object* – include recording format, video duration, frames size and rate, and compression techniques.
- iii) *The attributes that describe the video and its relationship with other database objects* – such as title, director, date of production, and production studio.

## **2.1 Video Browsing Techniques**

Video browser is a user-interface interactively between users and the browser itself. Video browser allows the users to browse, retrieve, upload, and search the video files based on the user's preferences and view segments. Basically, there are different techniques on video browsing identified by A. Rowe, S. Boreczky, and A. Eads (1994) [2]:

- a) Bibliographic data
  - Information about the video such as title, abstract, subject, and genre.
  - Individuals involved in the video such as producer, director, and cast.
  - It is a traditional method where the video data will be stored with many keywords that describe about the video content itself.
- b) Structural data
  - A scene that is composed of a sequence of shots and a segment is composed of a sequence of scenes.
- c) Content data
  - Users want to retrieve videos based on their content.
  - Keyframes will represent the content of the video.
  - Represent key images in a video in a set of keyframes.
  - Many researches are still figuring out on how to implement the content-based video retrieval as a whole in different areas of technology advancement.

From the findings (Hyowon Lee and F. Smeaton), most systems use a textual querying interface and only few systems use the visual query interface whereby this area should be more focused for future development on browsing the video contents [3].

## 2.2 Playback System on Video Browser

Besides, most systems use keyframe-based that represents each frame of video images viewed as thumbnails or filmstrips and the most important interface feature in video browsing applications is playback. The making of video browser more closely related to the video player in terms of user interaction as a whole where browsing and playback should not really be separated. The playback is a video player such as Windows Media Player and RealPlayer. Playback should be understood as one part of browsing [3].

## 2.3 Multimedia Database System

Rest assured that video is one part of media objects or presentable data in the multimedia system and identify the architecture of multimedia database system (see *Figure 2.1*). Initially, multimedia database systems contain the graphical images, video clips and sound files. These objects will hold infinite quantities of multimedia data that can be used to support applications such as movies on demand, architectural design, educational systems, and video on demand (VOD) systems.

*Figure 2.1* shown in the previous page is the basic architecture of multimedia database systems whereby video is located inside the video database management system (VDBMS). This is where the video files are stored and retrieved when users query for video browsing based on the textual-based, structural-based or content-based. Each layer of multimedia database systems is described as below:

- i) First layer of the architecture is *Database Management Layer* whereby there are four different of databases which are text DBMS, image DBMS, audio DBMS, and video DBMS that are functioning to each form of media. Each media has its own database storage.

- ii) Second layer shows *Multimedia Object Management and Query Processing Layer* where the integration of media will contribute to multimedia documents. This layer will operate on integrating various dimensions of multiple media objects such as space, time, and semantics.
- iii) The final or third layer is *Interactive Layer* or presentation layer that presents user interfaces for the system to display or play video and audio whenever users make a media request.

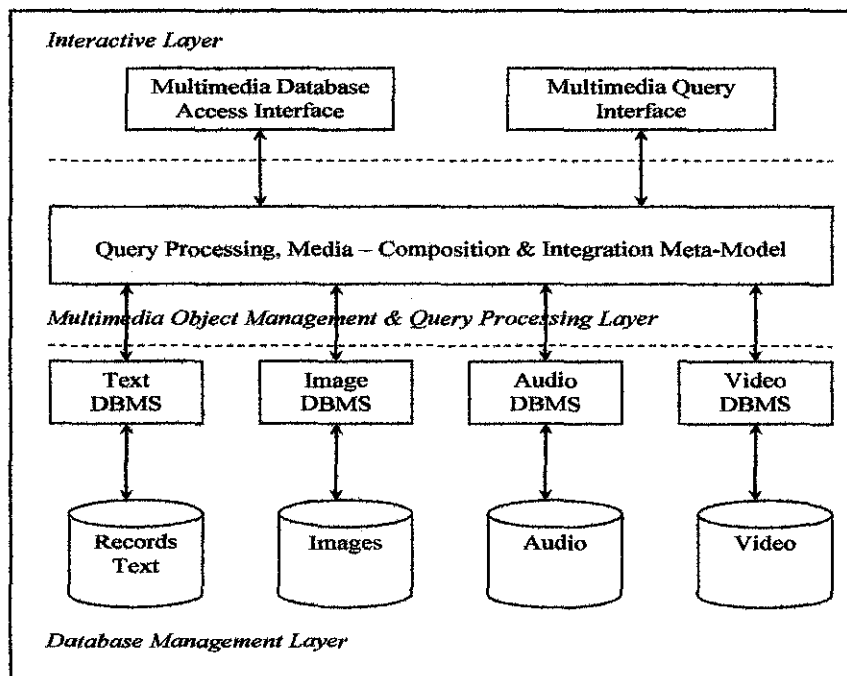


Figure 2.1: Multimedia Database Systems

## 2.4 Keyframe-Based Video Browsing

This section is very important because keyframe-based will be applied in video browser design framework. Initially, video content is abstracted to identify and extract a set of still images or keyframes from the video. Figure shown in the next page is the different levels of abstraction for video indexing (see *Figure 2.2*) [3].

At the lowest level, the video comprises a series of frames whereby each frame is an image that might display as a thumbnail image or filmstrips. Frames are grouped into shots that contain a continuous uninterrupted sequence of frames from the video



contents or recorded by the camera. Sequences of shots are combined to form a scene.

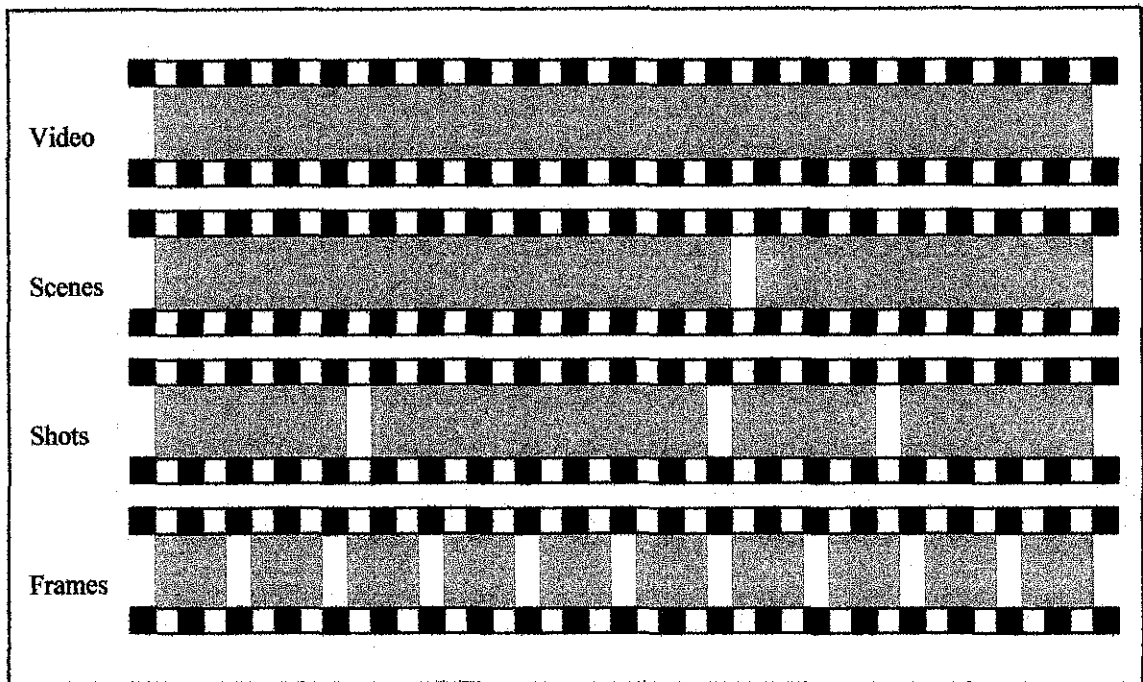


Figure 2.2: Video Abstraction Levels

In addition, each scene has the boundaries defined as the start and end of a shot which usually determine from the variations in the color distribution of the frames, editing effect that change the shot such as fade, wipe, and hard cut, and also based on the appearance and disappearance of an object. There are several factors that need to be considered in determining the boundaries between scenes which are:

- i) The *color content of successive frames* is likely similar when the frames form part of the same scene. Shots from the same scene are likely to share a common background and be shot with the same lighting conditions such as day and night scenes, outdoor and indoor scenes.
- ii) A *dialogue between two actors* will form part of the same scene even the color content of the frames is different. For example, two actors involved in a telephone conversation.
- iii) The *audio content* usually change significantly when a change of the scene rather than during a scene. For example, during a conversation of two actors, the dialogue is not usually synchronized exactly with the switching camera focus between two actors.

Frames that can represent video well are referred to as keyframes, and can be used to turn the temporal content (time duration) into static images that can be glanced. The effectiveness and usefulness of browsing such keyframes are evident in many places. For example, a movie poster that shows some snapshots of action scenes, a video or DVD tape cover showing interesting images from the video and Web-based video streaming which shows attractive still images taken from the movie.

Keyframe-based browsing is similar to the standard feature of "thumbnail browsing" in image retrieval interfaces, which show a number of small images before showing a larger single image if a user requests it. Indeed, it is difficult to find any image retrieval system that does not provide a thumbnail browsing interface. From the research done, it is important to have thumbnail or keyframe browsing feature in the video browser to allow the user to browse quickly through a large number of videos efficiently. The efficiency of displaying keyframes from video is similar to the efficiency in image browsing interfaces, but there is one crucial difference which keyframes are more than a set of pictures because there is a clear idea of time progression among the keyframe images. This is illustrated in *Figure 2.3*.

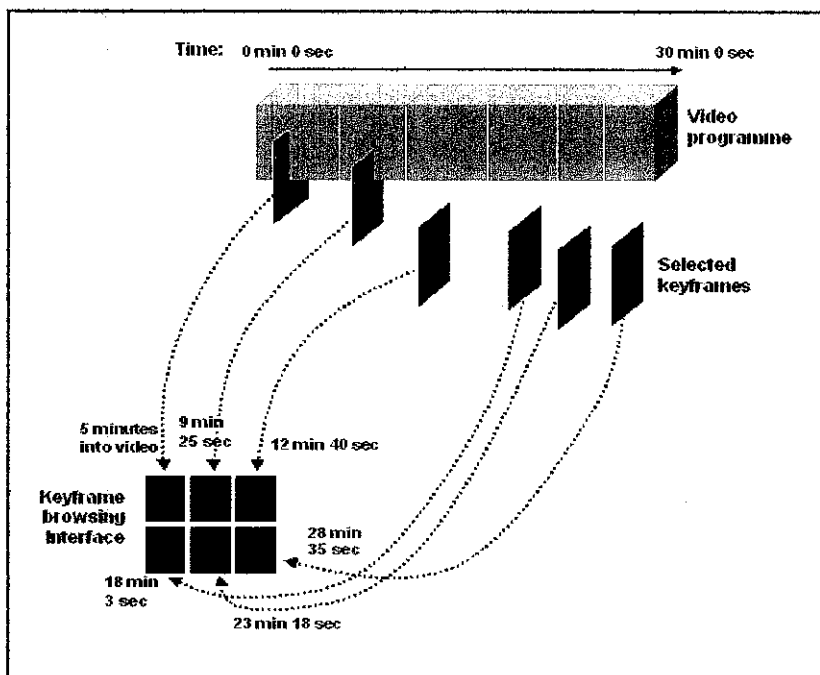


Figure 2.3: Keyframes selected from video to be displayed on a browsing interface.

Keyframe-based browsing can be categorized in different ways that are listed below:

- *Storyboard*: showing a set of miniaturized keyframes spatially on the screen in chronological order, allowing quick browsing. This is also termed as "keyframe lists" or "filmstrips".
- *Slide show*: flipping a set of keyframes one by one on the screen.
- *Hierarchically arranged browser*: showing different levels of detail hierarchically, the user browsing keyframes in a drill-down manner, which is especially suitable when the content itself is a structured video such as news or magazine programmes.

As an idea to identify and understand different user's information-seeking behavior, different systems of video browsing support different interface features to cater different users, tasks, and domains. Inside the Fischlar Digital Video Library frameworks (see *Appendix 2-1* and *Appendix 2-2*), the researchers have divided the interface into several elements supporting different user actions. All the interface elements identified are listed as follows [3]:

i) Interfaces for browsing and selecting video programs

In the Fischlár system, its main interface organizes video items in different folders for easily sorted, filtered and browsed. The searching through a digital video library is often done in queries on some form of metadata from the video such as title, date, or description.

ii) Interfaces for querying video content

User specifies a query in terms of actual video content attributes. There are two methods which are manually pre-indexed and automatically indexed video collections. For manually indexed, user can select different content attributes such as person, events and locations selectable from a list. However, for automatically indexed, user can query either by drawing objects with certain shapes and colors on a canvas by specifying motion or selecting example video clips from the screen. Specifying

motion in a query is useful when user is looking for a video clip where something is moving from one spatial location to another. For this project, we will focus on manually indexed on the video data during user searching.

iii) Interfaces for browsing video content

The search results of video content are displayed on the keyframe-based video browsing. **The keyframes will represent the content of the video.** Video content is abstracted to identify and extract a set of still images or keyframes from the video which is done automatically. Initially, video is composed of a sequence of still images, or frames that are continuously changing at a speed of 25 frames per second [3]. In other words, the video data will be displayed frame-by-frame to allow the user for browsing the content of video data. This kind of video retrieval interfaces is similar to Fast Forward/Rewind on a home VCR machine or some selective keyframe browsing on a digital video library.

iv) Interfaces for watching video content

Even though watching or video playback may or may not be the final goal of a user's video searching task, however, playback is often the last stage of the user's interaction with the video library, for instance, and interfaces support are very common in the form of video player software such as RealPlayer, Windows Media Player, and Apple Quicktime Player. These players basically have the same playback interface with buttons for play/pause/stop/rewind, volume control, and a timeline bar which indicates current point of playback for overall video clip.

## CHAPTER 3

### METHODOLOGY or PROJECT WORK

#### 3 PROJECT DEVELOPMENT

The idea of the project is to develop an open source video browser to serve an even broader audience or users (see *Appendix 3-2* to *Appendix 3-11*). So, the video browser is developed by using open source software which is PHP. The development of this project is based on System Development Life Cycle which is described as follows:

1. *Planning* – Do a research about an open source video browser and its main elements which emphasizes the user interface and its output to the users. This stage is already done on Final Year Project Part A.
2. *Analyzing* – Identify the system requirement based on the research done which prioritizes user's perspectives.
3. *Designing* – Create user interface for the video browser.
4. *Implementing* – Put on production that allows the end users to use the video browser after passing the testing stage.

The idea of designing the interfaces for video browser will include an interface to **upload, display the searching results, and play the video clip** based on user's preferences. Three (3) functions specified are very crucial to be implemented so that the objectives and scope of study can be achieved.

### 3.1 Setting the Configuration Files for PHP

Figure 3.1(a) and Figure 3.1(b) shown below are the guidelines to set up the php.ini in the configuration files. These guidelines are very crucial to be implemented in order to allow the video to be uploaded and stored into the file server.

```

;::::::::::::::::::;
; Resource Limits ;
;::::::::::::::::::;

max_execution_time = 30      ; Maximum execution time of each script, in seconds
max_input_time = 60         ; Maximum amount of time each script may spend parsing request data
memory_limit = 8M           ; Maximum amount of memory a script may consume (8MB)

;::::::::::::::::::;
; File Uploads ;
;::::::::::::::::::;

; Whether to allow HTTP file uploads.
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system default if not
; specified).
upload_tmp_dir =

; Maximum allowed size for uploaded files.
upload_max_filesize = 2M

```

Figure 3.1 (a): Before – System Default php.ini

```

;::::::::::::::::::;
; Resource Limits ;
;::::::::::::::::::;

max_execution_time = 30      ; Maximum execution time of each script, in seconds
max_input_time = 60         ; Maximum amount of time each script may spend parsing request data
memory_limit = 20M          ; Maximum amount of memory a script may consume (8MB)

post_max_size = 100M

;::::::::::::::::::;
; File Uploads ;
;::::::::::::::::::;

; Whether to allow HTTP file uploads.
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system default if not
; specified).
upload_tmp_dir = "C:/wamp/www/videobrowser/upload/"

; Maximum allowed size for uploaded files.
upload_max_filesize = 100M

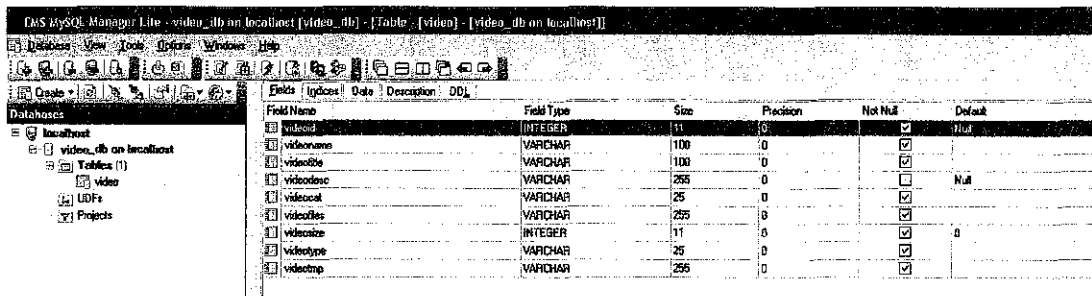
```

Figure 3.1 (b): After – Setting up the php.ini to allow the video to be uploaded.

### 3.2 Database Design

This project involves the development of video database by using software called EMS MySQL Manager Lite which will be linked to PHP MySQL (see *Figure 3.2*). All the video information is stored into database and will be retrieved through the video browser. All the video data is stored into database called as PHP MySQL. There are four attributes on Video table created which are *video id*, *name*, *title*, *description*, *category*, *file*, *size*, *video type*, and *temporary file*. The SQL command to create the video table is shown below:

```
CREATE TABLE `video` (  
  `videoid` int(11) NOT NULL auto_increment,  
  `videoname` varchar(100) NOT NULL default '',  
  `videotitle` varchar(100) NOT NULL default '',  
  `videodesc` varchar(255) default NULL,  
  `videocat` varchar(25) NOT NULL default '',  
  `videofiles` varchar(255) NOT NULL default '',  
  `videosize` int(11) NOT NULL default '0',  
  `videotype` varchar(25) NOT NULL default '',  
  `videotmp` varchar(255) NOT NULL default '',  
  PRIMARY KEY (`videoid`)  
) TYPE=MyISAM
```



Field Name	Field Type	Size	Precision	Not Null	Default
videoid	INTEGER	11	0	<input checked="" type="checkbox"/>	Null
videoname	VARCHAR	100	0	<input checked="" type="checkbox"/>	
videotitle	VARCHAR	100	0	<input checked="" type="checkbox"/>	
videodesc	VARCHAR	255	0	<input checked="" type="checkbox"/>	Null
videocat	VARCHAR	25	0	<input checked="" type="checkbox"/>	
videofiles	VARCHAR	255	0	<input checked="" type="checkbox"/>	
videosize	INTEGER	11	0	<input checked="" type="checkbox"/>	0
videotype	VARCHAR	25	0	<input checked="" type="checkbox"/>	
videotmp	VARCHAR	255	0	<input checked="" type="checkbox"/>	

Figure 3.2: Attributes that will describe the video to be stored into database

Each attribute is described as follows:

- **videoid:** Video ID as a primary key which will generate in increment automatically when a new data is inserted.
- **videoname:** Indicates the name of the author or artist, for example *Dewa*.
- **videotitle:** Title of the video which is related to the video name (videoname) such as *Sedang Ingin Bercinta*.
- **videodesc:** Describe in details about the video related to the video name (videoname) and video title (videotitle) will be stored or uploaded into database.
- **videocat:** To categorize the video information into specific category such as *Entertainment* and *Music* for easier used in searching process.
- **videofiles:** Upload or store the video files in various types of video format such as .wmv, .mpeg or .mpg. The datatype for this video is BLOB (Binary Large Object) (*Figure 3.2*) that specifically for large size of video whereby the video itself will go into a temporary location on the Web server or localhost (see *Figure 3.3(a)* and *Figure 3.3(b)*).
- **videosize:** Show the size of the video in bytes when the video is uploaded. The system will allow the user to upload the video less than 104857600 bytes.
- **videotype:** Indicates the Multipurpose Internet Mail Extensions (MIME) type for the uploaded video file which is `video/x-ms-wmv` and `video/mpeg`.
- **videotmp:** Name of the video file as stored in the temporary directory on the Web server



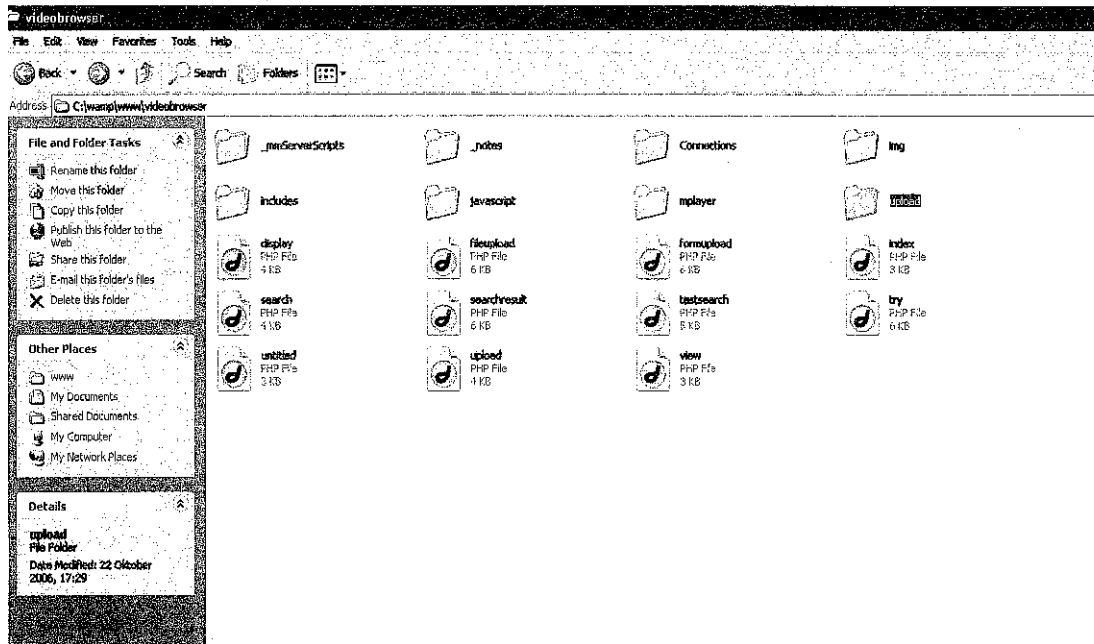


Figure 3.3 (a): Creating the folders to store the video clips (upload folder).

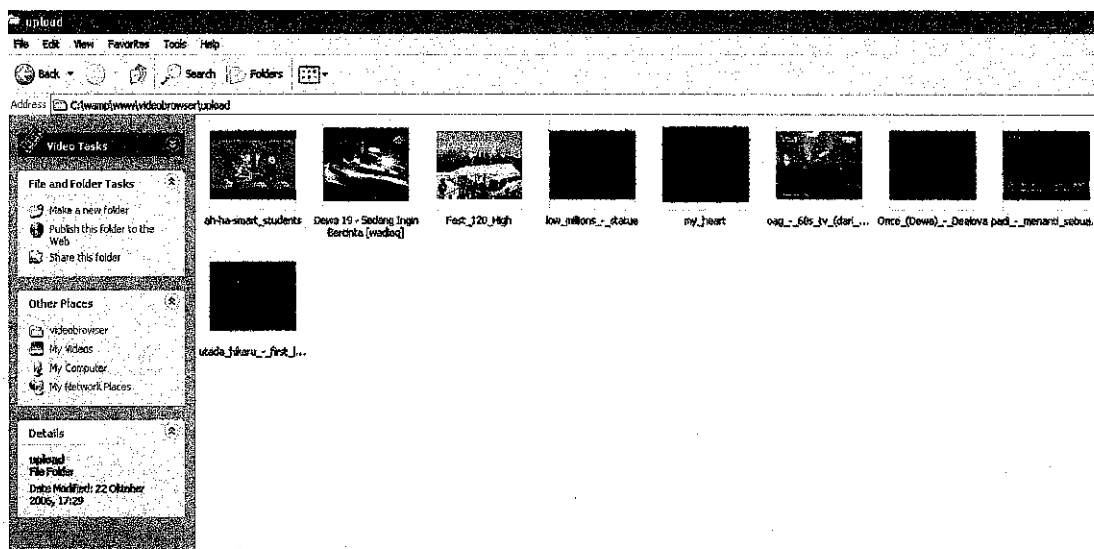


Figure 3.3 (b): All the uploaded video files are stored inside the upload folder.

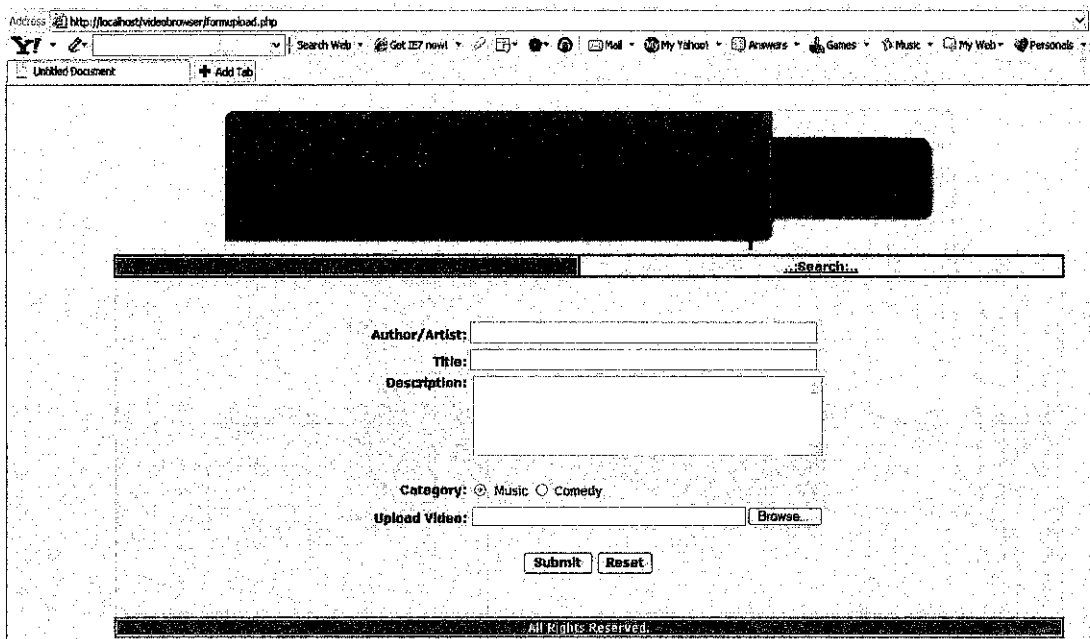
In addition, the video format for this video browser is standardized with the use of `video/x-ms-wmv` (.wmv format) and `video/mpeg` (.mpg and .mpeg formats) in order to ensure the consistency and integrity of accessing the video on the web and video database. If we do not standardize the video format especially in the video streaming technology, the database will become slower than expected and affect the accessibility of the videos on the web.

### 3.3 Interface Design

User interface shown in the next page (*Figure 3.4 and Figure 3.5*) is a page for uploading and viewing the video respectively. The video browser is developed by using Macromedia Dreamweaver MX 2004 and SWiSHmax for flash interactive design. Besides, this development includes all main elements that are crucial in the video browser which are listed as follows:

- i) Interfaces for browsing and selecting video programs
- ii) Interfaces for querying video content
- iii) Interfaces for browsing video content
- iv) Interfaces for watching video content

#### 3.3.1 Uploading Video Files Page



The image shows a screenshot of a web browser window. The address bar displays the URL: `http://localhost/videobrowser/formupload.php`. The browser's toolbar includes a search bar, a "Search Web" button, and several icons for navigation and search. The main content area of the browser shows a form for uploading a video. The form includes a large black rectangular area at the top, likely a video player or a placeholder for a video thumbnail. Below this is a search bar with the text "...Search...". The form fields are as follows:

- Author/Artist:** A text input field.
- Title:** A text input field.
- Description:** A larger text input field.
- Category:** A radio button selection with two options:  Music and  Comedy.
- Upload Video:** A text input field followed by a "Browse..." button.
- Submit** and **Reset** buttons are located at the bottom of the form.

At the bottom of the browser window, there is a footer that reads "All Rights Reserved."

Figure 3.4: Interface to upload the video into the database.

This is an application that enables users to upload files (see *Appendix 3-3*). It is important to know how PHP handles uploaded files prior to understand how the upload works. In this page have a form like the following:

- In the `<form>` tag must set the attribute `enctype="multipart/form-data"` to let the server know that a file is coming along with the regular form information. The full coding using `<form>` tag is as follows:

```
<form action="formupload.php" method="post"
enctype="multipart/form-data" name="form1">
```

- Sets the maximum size file in the form field that can be uploaded. This is a hidden field and is shown here as:

```
<input type="hidden" name="MAX_FILE_SIZE"
value="2000000000000000000">
```

The name of this form field must be `MAX_FILE_SIZE`. The value is the maximum size (stated in bytes) of files that allow people to upload.

- Need an input of type file as shown below:

```
<input name="userfile" type="file" id="userfile">
```

Can choose any name for the file and will be used to access the file from the receiving PHP script.

When a file is specified and this form is submitted, PHP automatically creates a few variables. They are stated as follows:

- `$_FILES['userfile']['tmp_name']` – The name of the file as stored in the temporary directory on the Web server.
- `$_FILES['userfile']['name']` – The name of the file as it was on the user's machine.
- `$_FILES['userfile']['size']` – The size of the file in bytes.
- `$_FILES['userfile']['type']` – The Multipurpose Internet Mail Extensions (MIME) type of the file (see *Appendix 3-1*), for example `video/mpeg` or `video/quicktime`.

The video is stored in the temporary directory specified in the `php.ini` file. If no temporary directory is specified in `php.ini`, the operating system's default temporary directory is used.

When allowing file uploads to users, this open up to the denial-of-service (DoS) attacks. Someone could send many multi-megabyte files to the system simultaneously if not careful, which could bring the machine to a crashing halt. There are two things can overcome this problem which stated as in the next page:

1. As stated above, put a hidden form field before `<input name="userfile">`. The hidden form should look as shown below:

```
<input type="hidden" name="MAX_FILE_SIZE" value="104857600">
```

This is reasonable first step and can help in stopping someone who didn't know we have set a size limit. However, this measure will not stop anyone with stronger malicious intent. A person can look into the source code of the page and make the needed changes.

2. Another security measure is altering the `php.ini` file inside the configuration files. It contains the `upload_max_filesize` item that can be set to a number that is reasonable. By default, `php.ini` will allow 2MB uploads. By making the change in the `php.ini` file will prevent the attacker from altering it.

### 3.3.2 Interface of playing the video

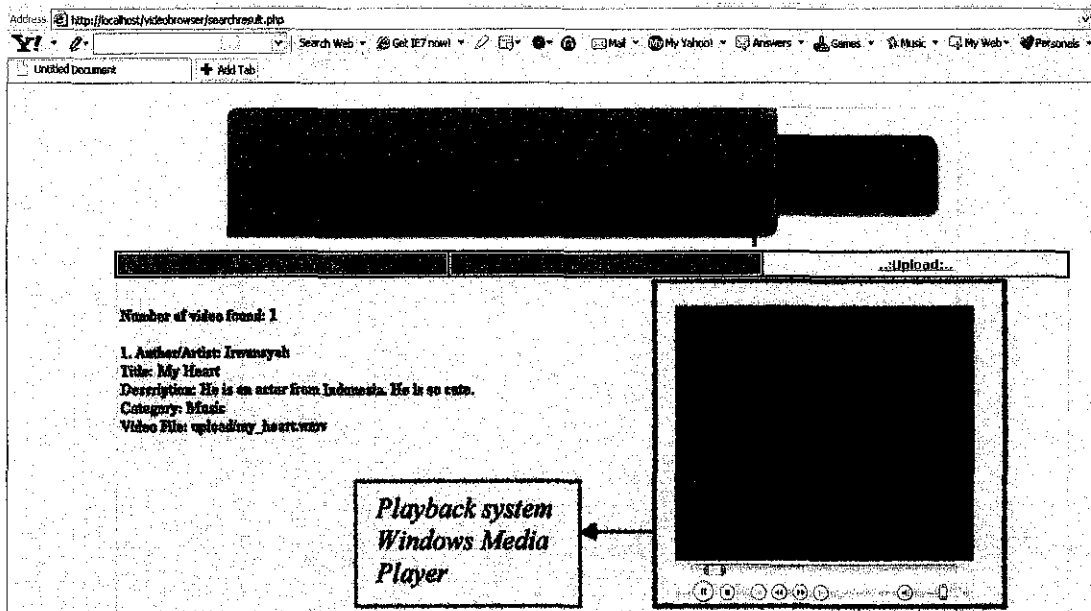


Figure 3.5: Interface to display and play the video clip based on user's preferences.

Figure 3.5 at the above shows the playback feature in the video browser. The player used is Windows Media Player. This interface will allow the user to watch video based on their selected video clip. It is the same like video streaming technology. Besides, the playback will display video information such as video name (author/artist), title, description, category or genre, and the path name of video file.

### 3.3.3 Interface of searching the video

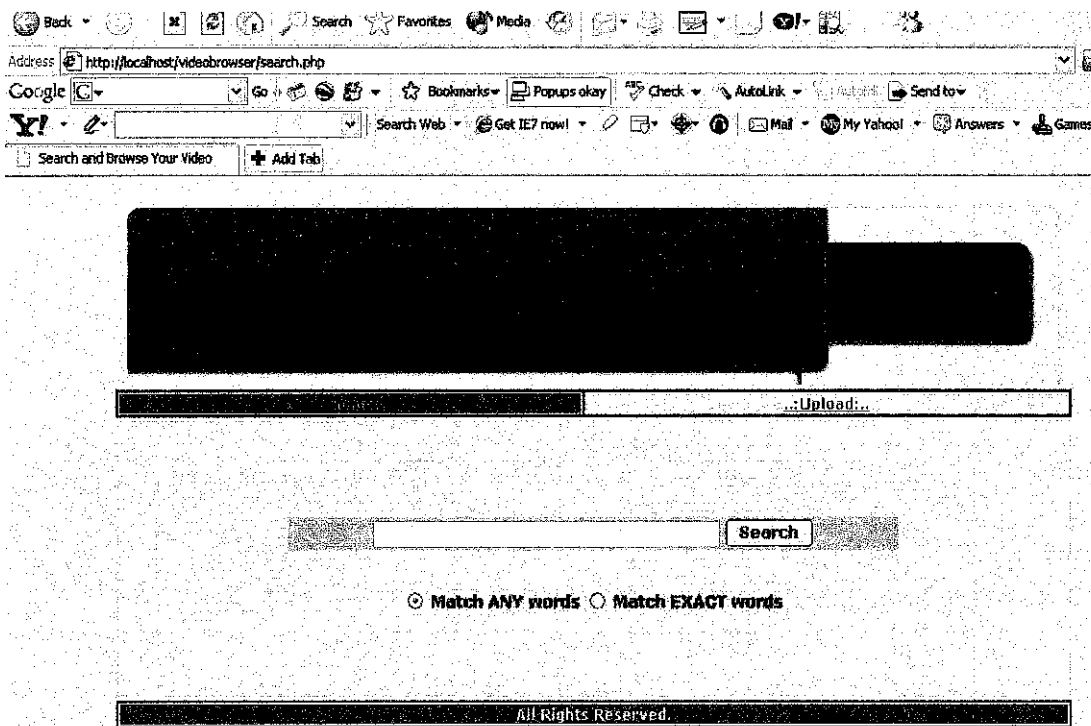


Figure 3.6: Interface to search video based on the keyword entered by the user.

When the user enter a keyword for example “Dewa” in the text field and press the Search button, then the system will search the video based on video name or title. The system will retrieve the video information at searchresult.php page and play the video.

Please refer to Appendix 3-8 and Appendix 3-9.

# CHAPTER 4

## RESULTS and DISCUSSIONS

### 4 DISCUSSIONS and RESULTS

#### 4.1 Solutions to the Problem Statements

##### 4.1.1 How do users know what is there?

This can be done by classifying or categorizing the video data into specific category or genre such as Comedy and Music. By categorizing the video data, the user will be able to search based on user's preferences. *Figure 4.1* and *Figure 4.2* below, are an example of categorization that should be provided on the video browser application:

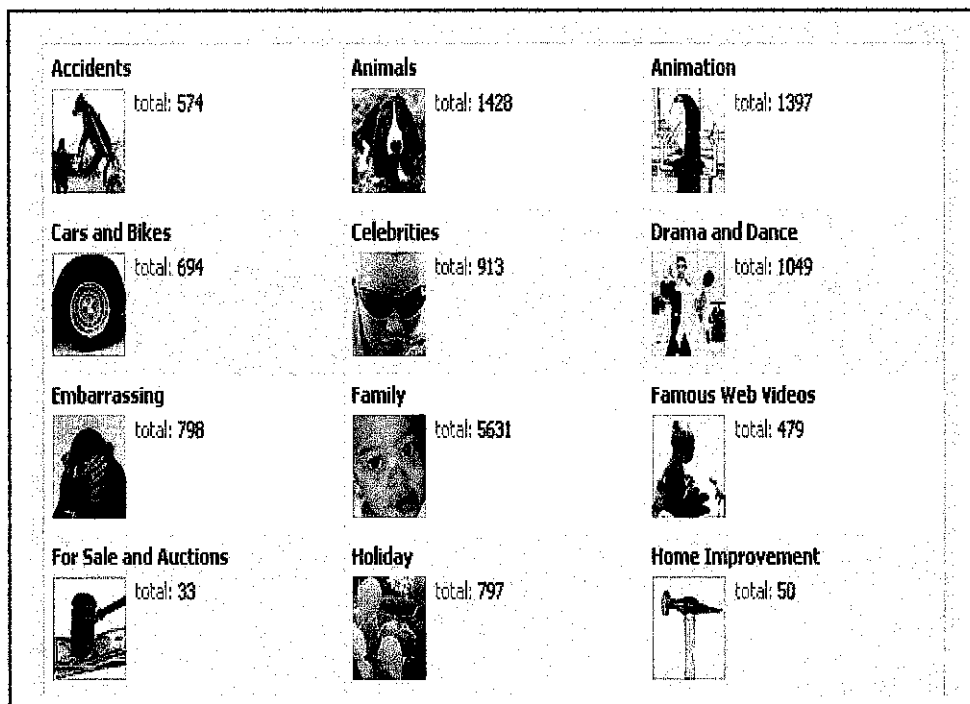


Figure 4.1: An example of video classification from Grouper.com.

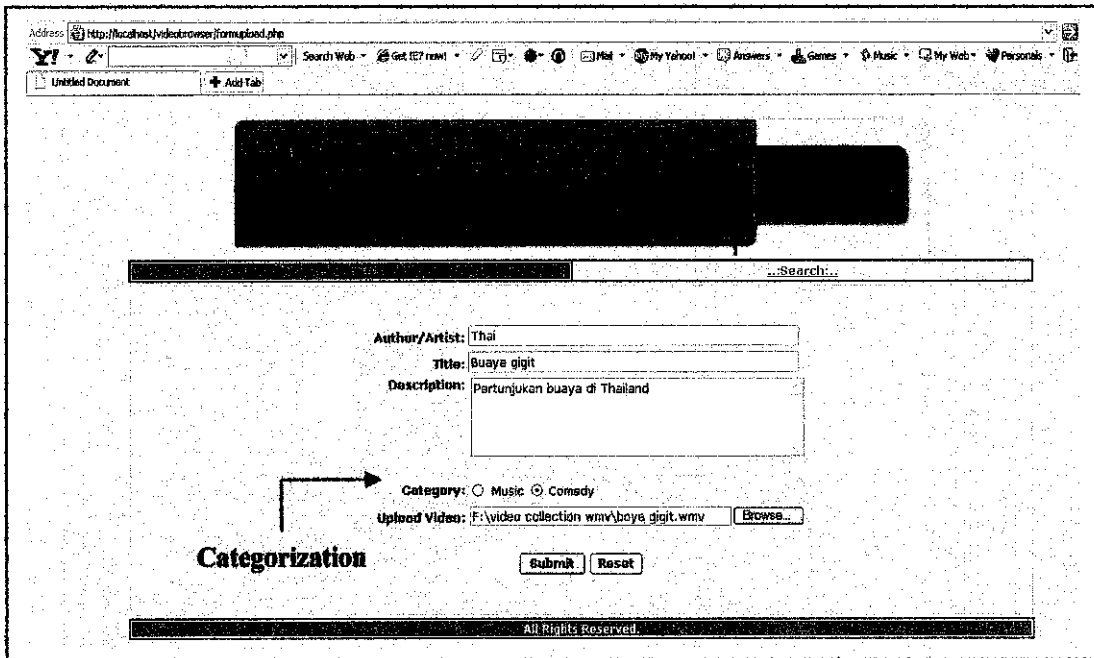


Figure 4.2: Enables the user to choose the Category (i.e. Radiobutton - Music or Comedy) that corresponds to their uploaded video.

#### 4.1.2 How users find anything there?

Based on the findings, the author has suggested that the video content should be displayed in keyframe basis with the use of video description effectively. The keyframe-based video browsing is applying video image content (in .jpeg or .gif formats) on the browser which allows the user to visualize and find what they want. With the use of video description that will describe the video, the user will be able to interactively browse the whole content of a video clip very quickly and efficiently. Thus, ensures the efficiency of video context and interactivity in the video browser.

Other method is querying in textual-based whereby each video contains many keywords that describe the video content itself. For example, with the use of search tools that will guide the user to query based on user's preferences. It is a conventional method that effective in retrieving the desired video content.

Storing the video records (footage) that combine with some level of classification can help the video database to find appropriate material. Query tool is used which will refer to scene, shot, and video time code. In this system, footage can be tagged



with complex annotations and key frames that will facilitate indirectly and directly browsing by topic and visual frame.

### 4.1.3 How can users find what they want?

When user has seen and knows something exists, user can identify it from a log of storyboards where they can ask for it by shot or by frame number. Reassured that this problem is difficult to handle and find its solution because the video searching process might not satisfy what the user want. This is due to very large database and depends on the speed of network.

## 4.2 Results

The author has done an analysis towards comparing some existing video browsers in different aspects. Table 4.1 shown in the next page indicates the differentiation of some existing video browsers from the aspects of:

- **Playback system** – allowing the user to play the selected video clip on video player.
- **Keyframe-Based** – displaying the video contents in a keyframe basis or frame-by-frame.
- **Keyframe type** – include storyboard, slide show, or hierarchically display.
- **Query tool** – provide search tool to the user for searching video process.
- **Allow upload** – provide upload page to give the user to upload the video into the database.
- **Open source** – indicate either the video browser is developed using open source software or not.

Video Browser	Playback System (Yes/No)	Keyframe-Based (Yes/No)	Keyframe Type(s)	Provide Query Tool (Yes/No)	Allow Upload (Yes/No)	Open Source (Yes/No)
<b>Fischlar System</b>	Browsing interaction quite separate from playback.	Yes	Storyboard	Yes	No	No
<b>Open Video Project</b> (see <i>Appendix 4.1 and 4.2</i> )	Yes	Yes	Storyboard and slide show	Yes.	No	Yes
<b>Google Video</b>	Yes	Yes	Storyboard	Yes	Yes	Yes
<b>Informedia Digital Library</b>	Yes	Yes	Storyboard	Yes.	No	No
<b>YouTube</b>	Yes	Yes	Storyboard and slide show.	Yes	Yes	Yes

Table 4.1: Comparative findings of some existing video browsers.

Based on the observation done on Table 4.1, it can be seen that:

- All systems provide query tools that allow the user to query for their desired video content either by textual query or visual query. This is one of the interface elements that is important to be included in the video browser.
- All systems that have been observed provide playback system such as Windows Media Player or RealPlayer as a most important interface feature.
- All systems use keyframe(s) as their video browsing method.
- There are two systems (Open Video Project and YouTube) are using both storyboard and slide show as keyframe type which ensure the interactivity and context of browsing interaction on the video browser. Others are using only storyboard to display the keyframe.
- Only two systems (Google Video and YouTube) allow the user to upload a video into the database through the video browser.
- Only three systems (Open Video Project, Google Video, and YouTube) are developed by using open source software i.e. PHP. As a consequence, it will

allow any community to do a research related to the video browser applications.

Throughout the results, the video browser that the author has been developed will include all main kinds all interface elements or features from the user's perspectives. The system will include all elements stated in the Table 4.1. Furthermore, the system is expected to display the video content in a keyframe basis and use the video information to describe each video displayed to assure the efficiency of interactivity and context.

This is to ensure that the system that has been developed will be different from the existing video browsers, hence meet or achieve the goals of the study. Later, the system will be tested by the selected number of end users so that they can give their comments and make final results to finalize the product or video browser being develop.

Effective video browsing depends on appropriate video representations. Human computer interaction (HCI) is important for the developers to develop the video browser based on the user's specifications or requirements. Different users have different taste, tasks, and domains in browsing the video. It is important for the users to know how they can assure the video is there in the storage, how they can find it, and how they can get their desired video files in the storage. At the end of the project also, user will be able to identify, understand, and familiar with video browsing interface elements provided in the video browser.

## **CHAPTER 5**

### **CONCLUSION and RECOMMENDATIONS**

#### **5.1 Lesson Learnt**

##### **5.1.1 Gaining New Knowledge**

From this study, the author has gained a lot of new knowledge that specifically in multimedia terms since it is normally used in Information Technology (IT) perspectives. The multimedia architecture and framework do not only include and concern the multimedia elements on the user interface such as widgets, playback system, and video browsing method, but also concerns about the knowledge management, data mining, and database management system.

At first, it is hard for the author to understand in depth about the multimedia framework and architecture from the topic chosen. Through the consultation and frequently ask questions from the FYP Supervisor, Mr. Nordin Bin Zakaria, then any difficulties or problems occurred have been already solved. In addition with frequent reading of many research papers make the author has more understanding and become enthusiastic in learning process and also developing the video browser.

##### **5.1.2 Improve Programming Skills**

This FYP involves web-based programming development. The development of video browser in a keyframe basis is an advanced learning in PHP and MySQL web development. Throughout the learning and developing the system, the author able to maximize and improve programming skills in a web-based applications. The author will find out by herself through accessing the websites and searching any forums that specifically tell about the experiencing problems or difficulties such as

uploading the video issue and displaying the video in a keyframe-based. These kind of problems are new things that the author has faced for the first time. Besides, the author will find someone who can help and has more experience in PHP programming. It is also important to encourage self learning in order to excel self motivation and improve skills especially in communication skills and computer skills.

## **5.2 Recommendations or Suggestion**

### **5.2.1 Initial Difficulty in Understanding How to Use**

As the video browser is used by many different users for different tasks or preferences, therefore it is important to build the browser that is familiar for the users to browse, query, and view the video contents. The users need recognition but not recall the content because this situation may lead to the deficiency on browsing the video contents.

Therefore, initial learning time is worth to be considered especially on the different video browsers. An initial idea is to avoid confusion to the users during the video browsing or querying the video files.

### **5.2.2 Questions on How Well Keyframes Represent Content**

As new browsers were being implemented, people were often curious about how the keyframes were selected. A brief explanation on automatically detecting camera shots and selecting one frame from each camera shot was given, and people were impressed by the "smart" technique which the system uses. However, the ways keyframes are selected, and especially the way selected keyframes are arranged in multiple layer browsers, are sometimes questioned. More research findings should be implemented on the keyframe-based or specifically content-based information retrieval (CBIR) for future development on browsing the video contents at high resolution so that it will become more effective and consistent to standardize with textual-based information retrieval.

### **5.3 Area of Further Study**

As the querying methods consist of textual query and visual query, the author would like to further study to apply the visual query. Based on the research, the visual query or content-based data retrieval have greater accuracy to retrieve the media contents such as images and videos from the user searching or browsing process. The content-based video retrieval is much easier for the users to recognize and visualize the desired video contents rather than recalls something that unsure or do not know to find the video data. Therefore, the author would like to prove by differentiating which query-based has higher accuracy of data retrieval system which is either textual query or visual query.

## BIBLIOGRAPHY

[1] Brad Bulger, Jay Greenspan, and David Wall. (2004). *MySQL/PHP Database Applications*. Wiley 2<sup>nd</sup> Edition.

## REFERENCES

[1] Alex May. (January 16, 2004). *A Web-based Image and Video Browser for Content-based Retrieval*. Outsourcing Report.

[2] Lawrence A. Rowe, John S. Boreczky, and Charles A. Eads. (February 1994). *Indexes for User Access to Large Video Databases*. University of California Berkeley.

[3] Hyowon Lee and Alan F. Smeaton. (May 21, 2002). *Designing the User Interface for the Físchlár Digital Video Library*. Centre for Digital Video Processing, Dublin City University.

[4] John L. Newman. *Multimedia Database Systems*.

[5] Barbara M. Wildemuth, Terrell Russell, T.J. Ward, Gary Marchionini, and Sanghee Oh. (2006). *The Influence of Context and Interactivity on Video Browsing*. Open Video Project, School of Information and Library Science, University of North Carolina at Chapel Hill.

## **APPENDICES**



## APPENDIX 2-1

### Físchlár's user interface: recording

<p>TV programmes for today and tomorrow : displayable by channel, genre or recommendation</p>	<p>Recording requested programmes in red</p>	<p>User preference indication icons (thumbs up/down)</p>
<p>The screenshot shows the Físchlár interface with a dark background. On the left is a sidebar with two columns of filters: 'Recommendations' and 'Recommendations'. The main area displays a 'Calendar: Tomorrow' for 'BIRIC 2 Tomorrow'. Programs listed include '10:00 Tomorrow's News', '10:30 Tomorrow', '11:30 Healthweek High', '12:00 Working Lunch', '12:30 Parks for Parks', and '14:00 Parks for Parks'. Some program titles are highlighted in bright yellow, while others are in gray. The bottom bar contains a 'RECORD PLAY' button, a 'TV CHANNELS' list, a 'TV CHANNELS' button, and a 'MY ACCOUNT' button.</p>		
<p>Recordable programmes in bright yellow: a user clicks on this title to request recording</p>	<p>Overlapping programmes with other channel in gray</p>	

## APPENDIX 2-2

### Físchlár's user interface: browsing/playback

List of recorded TV programmes can be filtered by channel and genre, ordered by date, length and title

Browsing Interface allows keyframe-based content browsing of a recorded TV programme selected from the list on the left



Browser icons panel allows the users to select one of the browsing interfaces

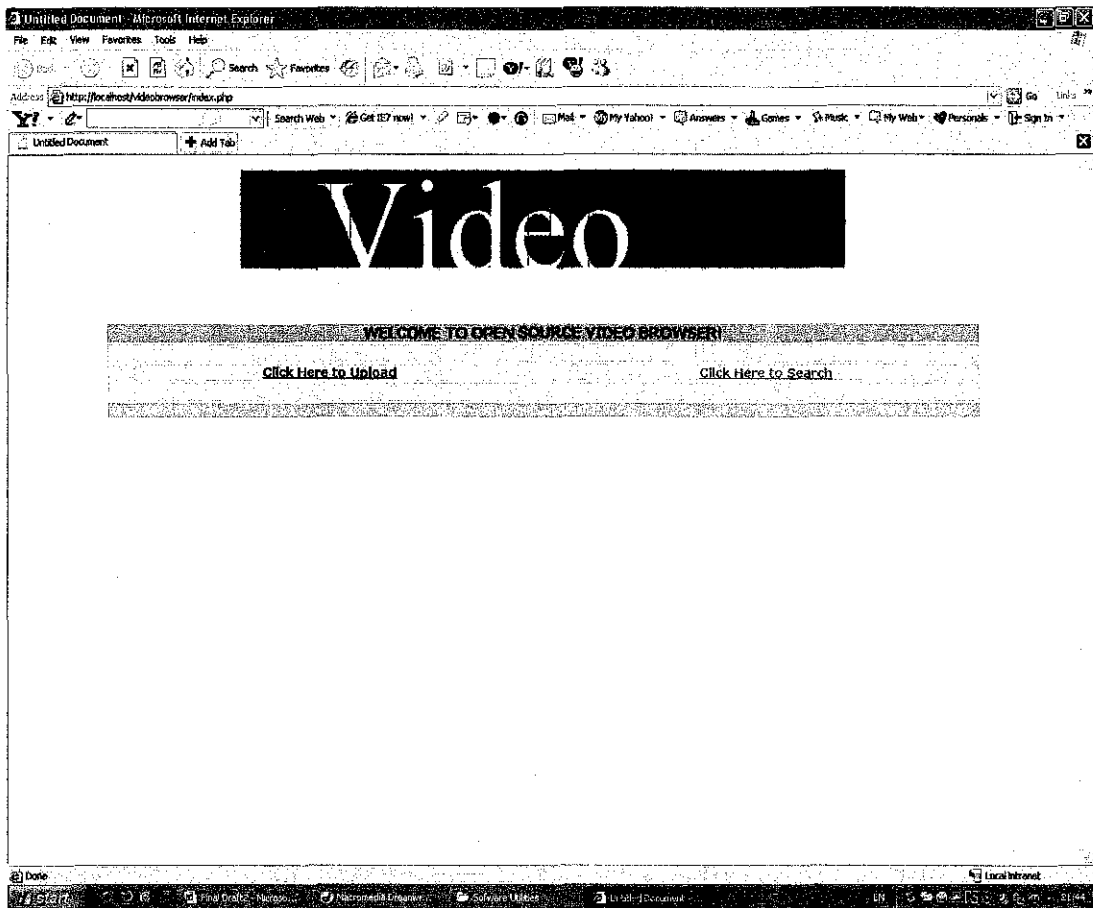
Player pops up on top of browser screen when a user clicks on a keyframe, with full-screen mode as option

**APPENDIX 3-1**  
**MIME Types for Media Content Type**

<b>Type/sub-type</b>	<b>Extension</b>
video/mpeg	mp2
video/mpeg	mpa
video/mpeg	mpe
video/mpeg	mpeg
video/mpeg	mpg
video/mpeg	mpv2
video/quicktime	mov
video/quicktime	qt
video/x-la-asf	lsf
video/x-la-asf	lsx
video/x-ms-asf	asf
video/x-ms-asf	asr
video/x-ms-asf	asx
video/x-msvideo	avi
video/x-sgi-movie	movie

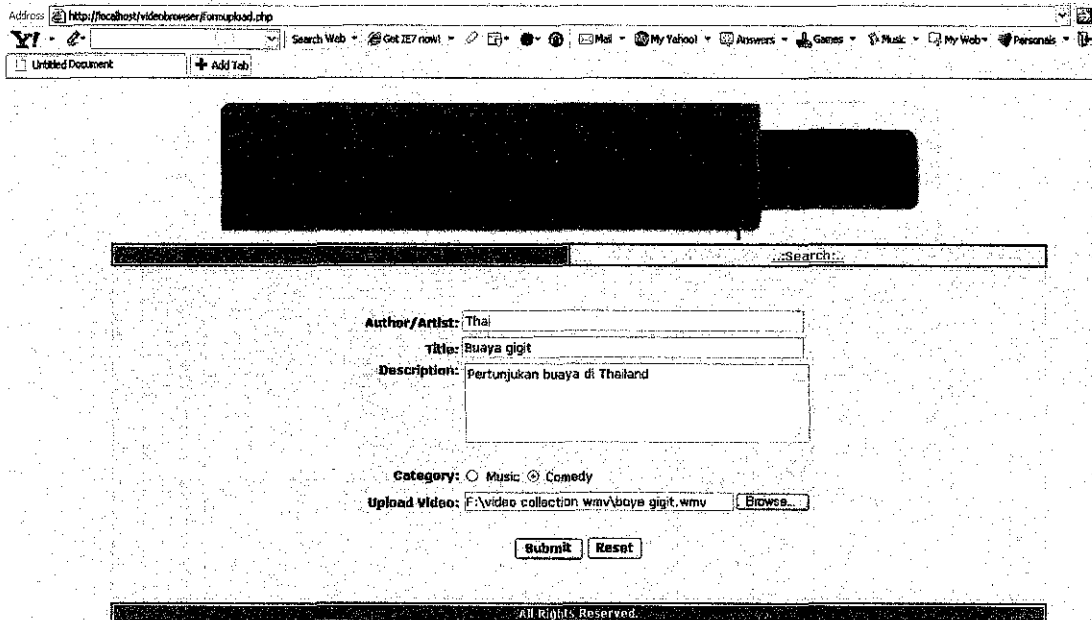
# APPENDIX 3-2

## Video Browser: Main Page



## APPENDIX 3-3

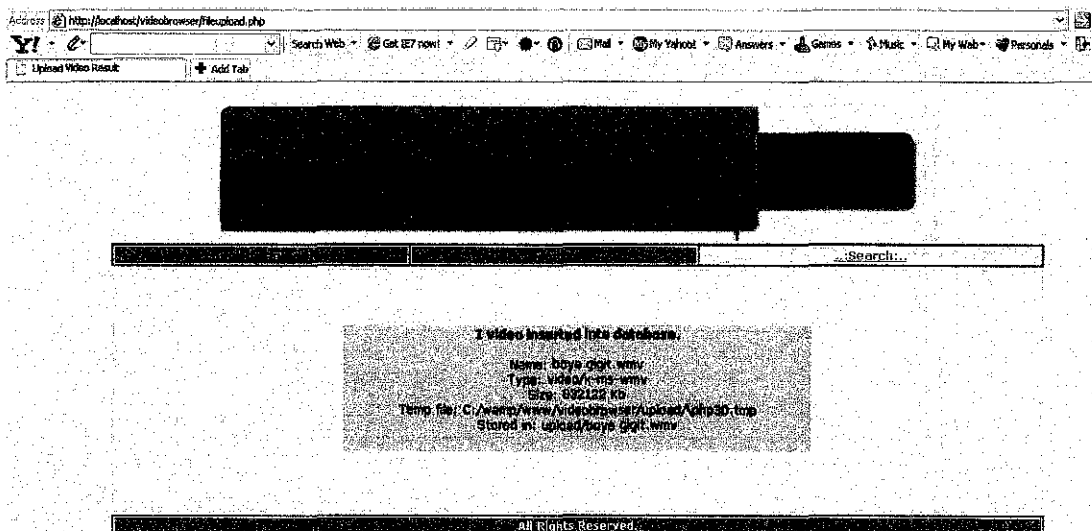
### Video Browser: Upload Page



The screenshot shows a web browser window with the address bar containing "http://localhost/videobrowser/fileupload.php". The browser's address bar also shows "Search Web" and "Get IE7 now!". The browser's tab bar shows "Untitled Document" and "Add Tab". The main content area of the page is mostly obscured by a large black redaction box. Below the redaction, there is a search bar with the text "Search:". Below the search bar, there are several form fields: "Author/Artist: Thai", "Title: Buaya gigit", and "Description: Pertunjukan buaya di Thailand". Below these fields, there is a "Category:" label with radio buttons for "Music" and "Comedy", where "Comedy" is selected. Below the category selection, there is an "Upload video:" label followed by a text input field containing "F:\video collection\wmv\buaya gigit.wmv" and a "Browse..." button. At the bottom of the form, there are "Submit" and "Reset" buttons. At the very bottom of the page, there is a footer that says "All Rights Reserved."

## APPENDIX 3-4

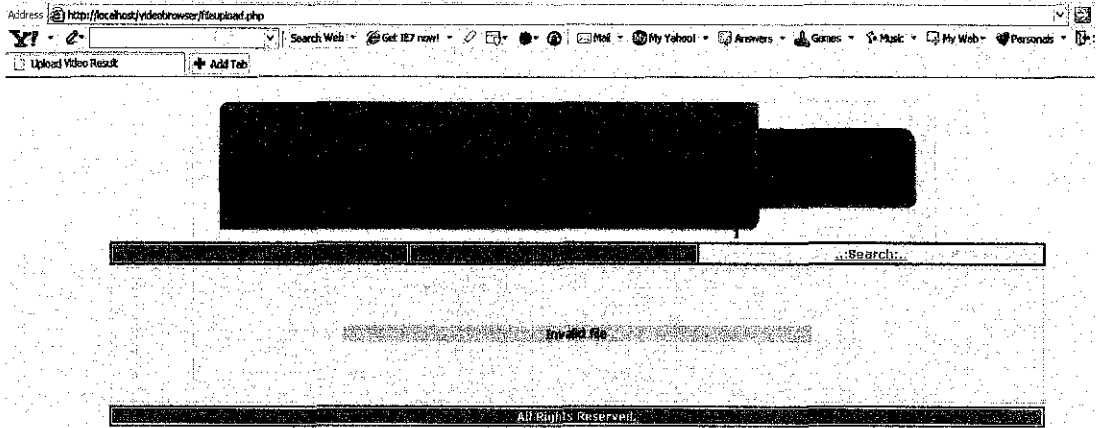
### Video Browser: Results Page when upload video is succeeded



The screenshot shows a web browser window with the address bar containing "http://localhost/videobrowser/fileupload.php". The browser's address bar also shows "Search Web" and "Get IE7 now!". The browser's tab bar shows "Upload Video Result" and "Add Tab". The main content area of the page is mostly obscured by a large black redaction box. Below the redaction, there is a search bar with the text "Search:". Below the search bar, there is a message box that says "1 video inserted into database:". Below the message box, there is a list of video details: "Name: buaya\_gigit.wmv", "Type: video/m-ms-wmv", "Size: 992122 KB", "Temp file: C:/wamp/www/videobrowser/upload/0h93D.tmp", and "Stored in: upload/buaya\_gigit.wmv". At the bottom of the page, there is a footer that says "All Rights Reserved."

## APPENDIX 3-5

### Video Browser: Results Page when upload video is not succeeded



## APPENDIX 3-6

**Video Browser: List all uploaded video into database in a grid view.**

videoid	videoname	videotitle	videodesc	videocat
1	Irwansyah	My Heart	He is an actor from Indonesia. He is so cute.	Music
2	Once	Dealova	Indonesia singer who is also has a band called Dewa19.	Music
3	Padi	Menanti Sebuah Jawaban	Indonesia band whereby this song is a soundtrack for film Ungu Violet.	Music
4	DAG	60's TV	Malaysian band lead by Radhi DAG.	Music
5	Dewa	Sedang Ingin Bercinta	Indonesian band.	Music
6	akha	Smart Student	Malaysia comedians.	Comedy
7	Low Millions	Status	English song.	Music
8	Utada Hikaru	First Love	Japanese Song.	Music
9	Pepsi	Ad World Cup	Pepsi commercial ad for World Cup.	Comedy
10	Thai	Buaya gigit	Pertunjukkan buaya di Thailand	Comedy

## APPENDIX 3-7

**Video Browser: Selected video in a form video**

**Object**

video\_db on localhost [vid...]

video

**General**

Refresh

Table properties

Table Editor options

Restore default size

**Data Management**

Commit transaction

Rollback transaction

**Explorer**

**Fields (9)**

- videoid [INT(11)]
- videoname [VARCHAR(100)]
- videotitle [VARCHAR(100)]
- videodesc [VARCHAR(255)]
- videocat [VARCHAR(25)]
- videofiles [VARCHAR(255)]
- videosize [INT(11)]
- videotype [VARCHAR(25)]
- videotmp [VARCHAR(255)]

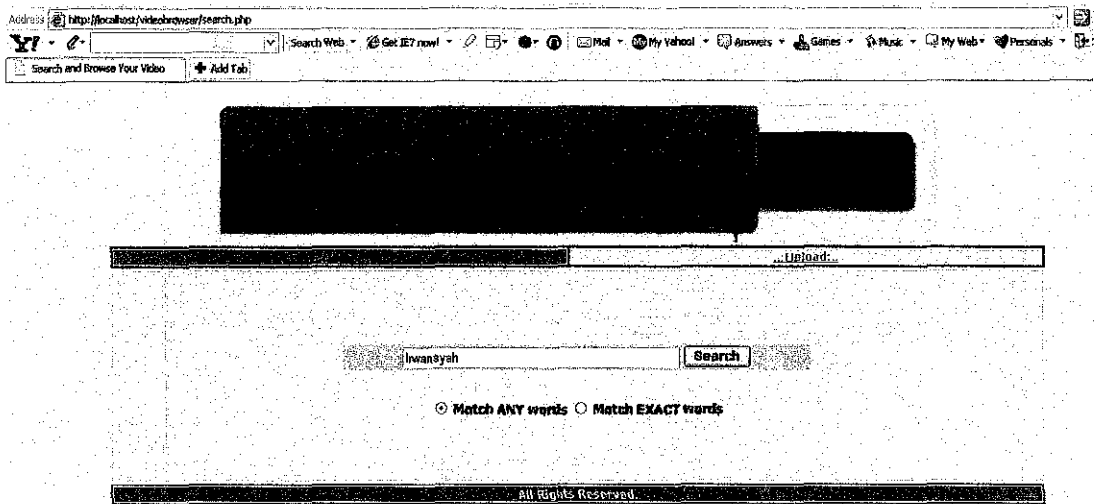
**Indices (1)**

- PRIMARY [videoid]

Fields	Data
videoid [INTEGER(11)]	9
videoname [VARCHAR(100)]	Pepsi
videotitle [VARCHAR(100)]	Ad World Cup
videodesc [VARCHAR(255)]	Pepsi commercial ad for World Cup.
videocat [VARCHAR(25)]	Comedy
videofiles [VARCHAR(255)]	upload/Feat_T20_High.wmv
videosize [INTEGER(11)]	6970034
videotype [VARCHAR(25)]	video/x-ms-wmv
videotmp [VARCHAR(255)]	C:/wamp/www/videobrowser/upload/jhp8.tmp

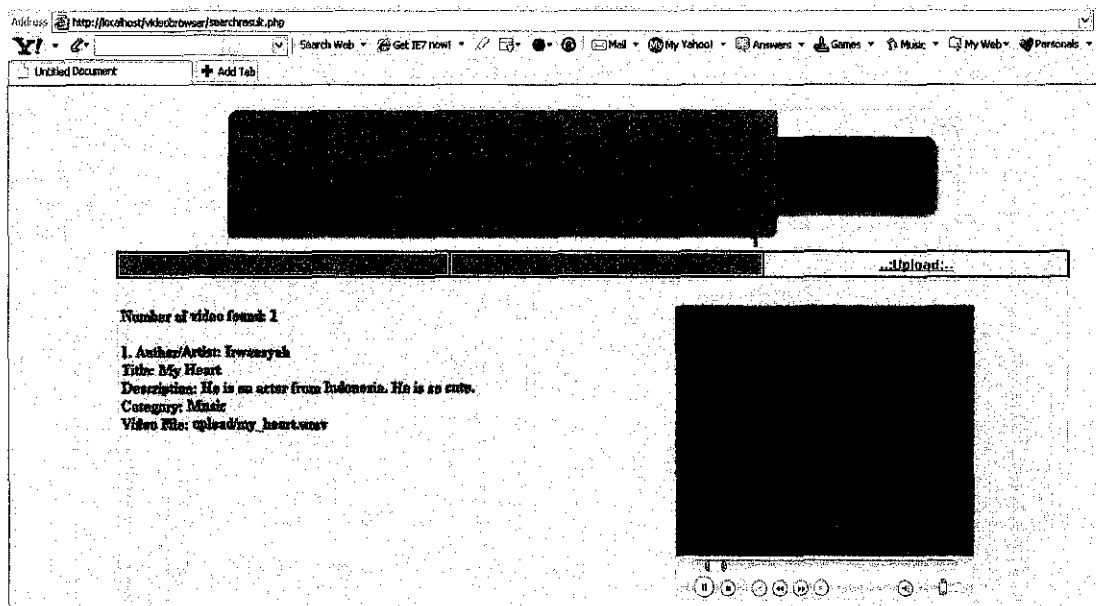
## APPENDIX 3-8

### Video Browser: Search Page



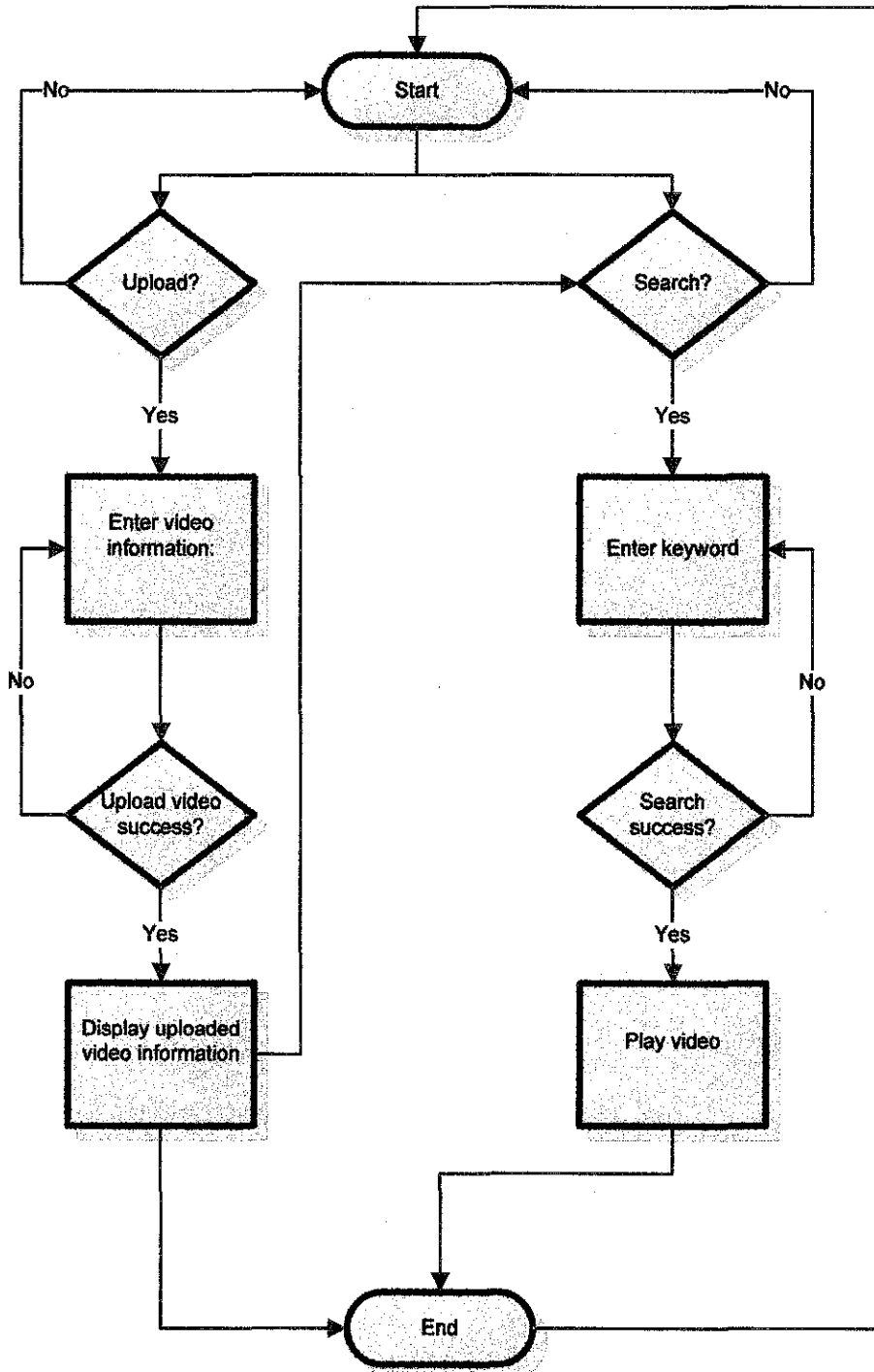
## APPENDIX 3-9

### Video Browser: Results Page when video information is found



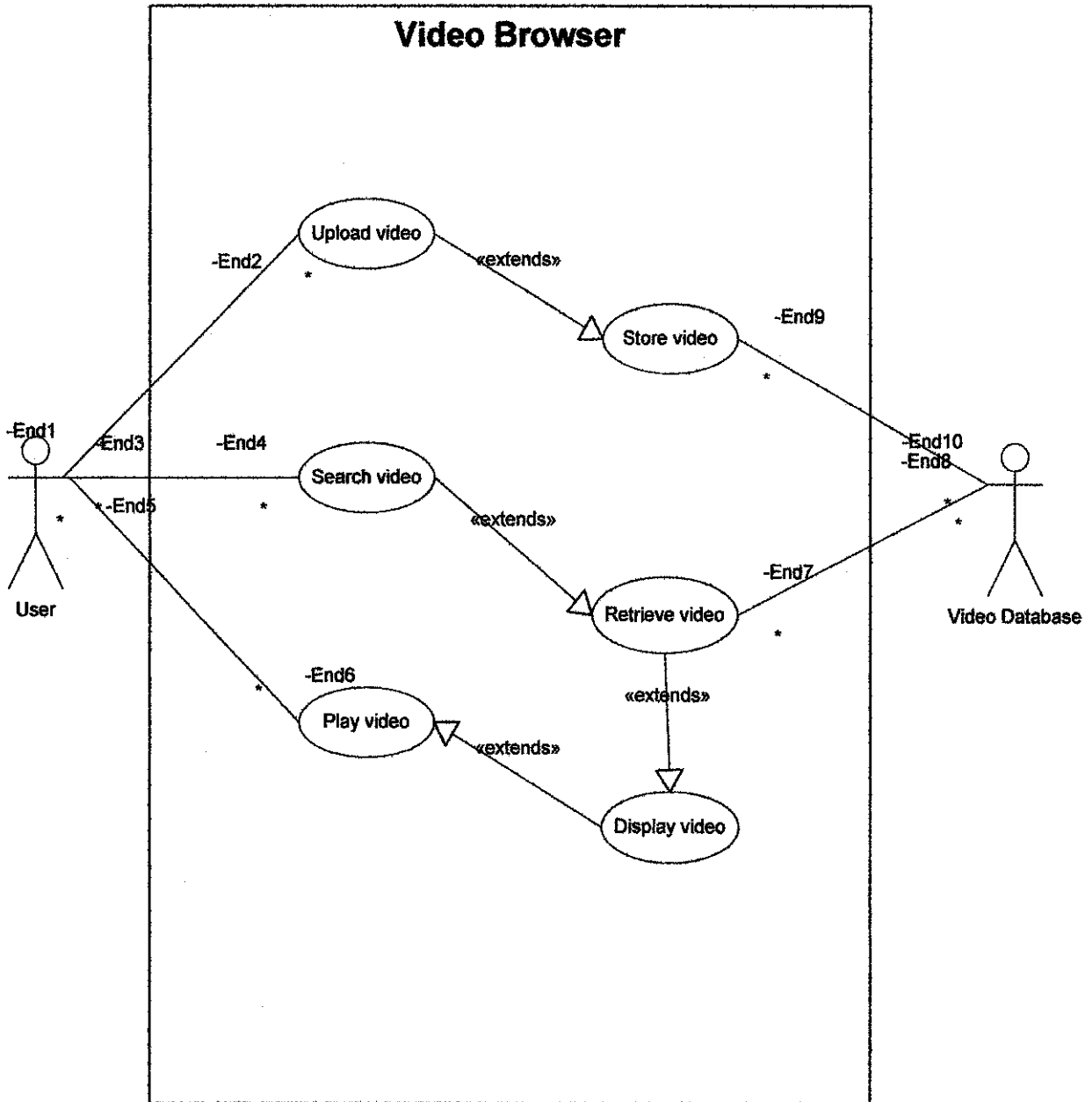


**APPENDIX 3-10**  
**Flow Chart Diagram for Video Browser**



# APPENDIX 3-11

## Use Case Diagram for Video Browser



# APPENDIX 4-1

## Open Video Project: Main Page

<b>Search</b> <input style="width: 100px;" type="text"/> <input type="button" value="Search"/> <a href="#">Detailed Search</a> <small>ex. "water" or "space shuttle"</small>		<b>Project New</b> <ul style="list-style-type: none"> <li>• Educators, take our survey!</li> <li>• ACM Video Finding Aid Available</li> <li>• NASA Video Finding Aid Available</li> <li>• Teachers using video?</li> </ul> <a href="#">more</a>									
<b>Browse</b>		<b>Featured Video</b> <ul style="list-style-type: none"> <li>• <b>New</b></li> <li> <b>NASACONNECT - MMOU - Expanding Universe Activity</b>  <ul style="list-style-type: none"> <li>• NASA Connect - FoF - Computer Simulation</li> <li>• NASA Connect - GoE - Mars Exploration and Water</li> <li>• NASA Connect - GoE - Elliptical Orbit Activity</li> </ul> <a href="#">more</a> </li> <li>• <b>Popular</b></li> <li> <b>Atom Bomb - Joe Bonica's Movie of the Month</b>  <ul style="list-style-type: none"> <li>• Classic Television Commercials (Part II)</li> <li>• NASA 25th Anniversary Show, Segment 01</li> <li>• NASA 25th Anniversary Show, Segment 07</li> </ul> <a href="#">more</a> </li> </ul>									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Genre</b></td> <td style="padding: 2px;"><b>Duration</b></td> </tr> <tr> <td style="padding: 2px;"> <ul style="list-style-type: none"> <li>• Documentary [494]</li> <li>• Educational [1147]</li> <li>• Ephemeral [2094]</li> <li>• Historical [187]</li> <li>• Lecture [33]</li> <li>• Other [6]</li> <li>• Public Service [17]</li> </ul> </td> <td style="padding: 2px;"> <ul style="list-style-type: none"> <li>• Less than 1 minute [263]</li> <li>• 1 to 2 minutes [440]</li> <li>• 2 to 5 minutes [745]</li> <li>• 5 to 10 minutes [698]</li> <li>• More than 10 minutes [1520]</li> </ul> </td> </tr> <tr> <td style="padding: 2px;"><b>Color</b></td> <td style="padding: 2px;"><b>Sound</b></td> </tr> <tr> <td style="padding: 2px;"> <ul style="list-style-type: none"> <li>• In color [1988]</li> <li>• In black &amp; white [1979]</li> </ul> </td> <td style="padding: 2px;"> <ul style="list-style-type: none"> <li>• With sound [3570]</li> <li>• Silent [397]</li> </ul> </td> </tr> </table>	<b>Genre</b>	<b>Duration</b>	<ul style="list-style-type: none"> <li>• Documentary [494]</li> <li>• Educational [1147]</li> <li>• Ephemeral [2094]</li> <li>• Historical [187]</li> <li>• Lecture [33]</li> <li>• Other [6]</li> <li>• Public Service [17]</li> </ul>	<ul style="list-style-type: none"> <li>• Less than 1 minute [263]</li> <li>• 1 to 2 minutes [440]</li> <li>• 2 to 5 minutes [745]</li> <li>• 5 to 10 minutes [698]</li> <li>• More than 10 minutes [1520]</li> </ul>	<b>Color</b>	<b>Sound</b>	<ul style="list-style-type: none"> <li>• In color [1988]</li> <li>• In black &amp; white [1979]</li> </ul>	<ul style="list-style-type: none"> <li>• With sound [3570]</li> <li>• Silent [397]</li> </ul>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Collections</b></td> </tr> <tr> <td style="padding: 2px;"> <ul style="list-style-type: none"> <li>• University of Maryland HCIL Open House Video Reports</li> <li>• The Informedia Project at Carnegie Mellon University</li> <li>• Internet Moving Images Archive</li> <li>• 2001 TREC Video Retrieval Test Collection</li> <li>• CHI Video Retrospective</li> <li>• Digital Himalaya Project</li> <li>• NASA K-16 Science Education Programs</li> <li>• William R. Farris Collection</li> </ul> </td> </tr> </table>	<b>Collections</b>	<ul style="list-style-type: none"> <li>• University of Maryland HCIL Open House Video Reports</li> <li>• The Informedia Project at Carnegie Mellon University</li> <li>• Internet Moving Images Archive</li> <li>• 2001 TREC Video Retrieval Test Collection</li> <li>• CHI Video Retrospective</li> <li>• Digital Himalaya Project</li> <li>• NASA K-16 Science Education Programs</li> <li>• William R. Farris Collection</li> </ul>
<b>Genre</b>	<b>Duration</b>										
<ul style="list-style-type: none"> <li>• Documentary [494]</li> <li>• Educational [1147]</li> <li>• Ephemeral [2094]</li> <li>• Historical [187]</li> <li>• Lecture [33]</li> <li>• Other [6]</li> <li>• Public Service [17]</li> </ul>	<ul style="list-style-type: none"> <li>• Less than 1 minute [263]</li> <li>• 1 to 2 minutes [440]</li> <li>• 2 to 5 minutes [745]</li> <li>• 5 to 10 minutes [698]</li> <li>• More than 10 minutes [1520]</li> </ul>										
<b>Color</b>	<b>Sound</b>										
<ul style="list-style-type: none"> <li>• In color [1988]</li> <li>• In black &amp; white [1979]</li> </ul>	<ul style="list-style-type: none"> <li>• With sound [3570]</li> <li>• Silent [397]</li> </ul>										
<b>Collections</b>											
<ul style="list-style-type: none"> <li>• University of Maryland HCIL Open House Video Reports</li> <li>• The Informedia Project at Carnegie Mellon University</li> <li>• Internet Moving Images Archive</li> <li>• 2001 TREC Video Retrieval Test Collection</li> <li>• CHI Video Retrospective</li> <li>• Digital Himalaya Project</li> <li>• NASA K-16 Science Education Programs</li> <li>• William R. Farris Collection</li> </ul>											
<b>Special Collection Spotlight</b> <small>none</small>											

## APPENDIX 4-2

### Open Video Project: Display video data in keyframe basis

The screenshot shows a web interface for a video search project. On the left is a 'Modify Search' sidebar with various filters. The main area displays search results for 'water', showing 136 videos found on page 1. The results are sorted by Relevance and 10 results are shown per page. Each result includes a keyframe image, the video title, segment information, year, genre, popularity, and download count.

**Modify Search**

Search: All fields  
for: water

Genre: Documentary

Duration: -AnyDuration-

Format: -AnyFormat-

Color:  Color  B&W  Either

Sound:  Sound  Silent  Either

**Search**

Detailed Search

Page 1 Search Results (136 videos found)

Layout: [Icons] sort by: Relevance Results per page: 10

<b>The Miracles of Water, segment 09 of 11</b> 1986 • Documentary • Popularity (downloads): 408	<b>How Water Won the West, segment 05 of 8</b> 1986 • Documentary • Popularity (downloads): 365	<b>How Water Won the West, segment 03 of 8</b> 1986 • Documentary • Popularity (downloads): 298	<b>The Miracle of Water, segment 01 of 11</b> 1986 • Documentary • Popularity (downloads): 607
<b>The Miracle of Water, segment 10 of 11</b> 1986 • Documentary • Popularity (downloads): 281	<b>The Rio Grande, segment 08 of 9</b> 1986 • Documentary • Popularity (downloads): 259	<b>How Water Won the West, segment 01 of 8</b> 1986 • Documentary • Popularity (downloads): 458	<b>The Rio Grande, segment 07 of 9</b> 1986 • Documentary • Popularity (downloads): 289