Blackboard, a course database system, is widely used in the University of North Carolina at Chapel Hill. It provides many technical supports and services to registered courses. Two of its most important functions are remote video viewing through Internet and comment posting on discussion forum. The current Blackboard system, however, is inconvenient for users to control them simultaneously. This paper describes the design and development of a database-driven and web-based digital video application system -- a novel working prototype which integrates the function of discussion forum and video/audio application based on specified needs for a community of users.

Headings:

Database—Management—System

Information System—Design

Interface Design

Web database
DESIGN AND IMPLEMENTATION OF A DATABASE-DRIVEN DIGITAL VIDEO APPLICATION SYSTEM FOR THE SCHOOL OF EDUCATION OF UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

By
Li Chen

A master’s paper submitted to the faculty of the School of Information and Library Science of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in Information Science.

Chapel Hill, North Carolina
December 2003

Approved by:

Advisor
Table of Contents

1. Introduction .................................................................1

2. Related Works .............................................................2

3. System Analysis ...........................................................3
   3.1 User Description ......................................................3
   3.2 Risk Assessments and Managements .............................9
   3.3 Business Requirements and Rules ...............................10
   3.4 Technology Decision/Justifications ............................10

4. System Design .............................................................12
   4.1 System Architecture Design ......................................12
   4.2 Database Design and Implementation ..........................14
   4.3 Interface Design and Implementation ..........................20
   4.4 System Design Idea and Technology Challenges ............28

5. Future Works ..............................................................33

6. Conclusion .................................................................35

7. Acknowledgements ......................................................35

8. Reference .................................................................36

9. Appendices ..................................................................37
1. Introduction

Blackboard, a course database system, is widely used in the University of North Carolina at Chapel Hill. It provides a variety of technical supports and services to registered courses. One of its functions is that it offers live remote viewing and on-demand replay of course materials using streaming video and audio over the Internet. Blackboard also allows registered users to post their comments on its discussion forum. However, the above two functions are separated, so they cannot be used simultaneously by one user. However, several courses supported by the Blackboard system and provided by the School of Education of the University of North Carolina at Chapel Hill, crave to integrate the functions of discussion forum and video/audio applications and provide users with easy control them both.

The purpose of this project is to create a discussion forum application system with the function of controlling the start time of a streaming video/audio that meets the special needs of the courses offered by the School of Education. It means that this application system is first a discussion forum, what users can control a streaming video/audio while posting comments and reading comments at the same time. A back-end MYSQL database is used to store users’ information, course information, video information, and comments. Web interfaces provide users at
different levels with different permissions to enter this system and allow all users to submit comments for a special video on line.

This paper introduces related works of similar projects in other research institutes in section two. Section three presents a system analysis of this particular system. It contains user description, user analysis, risk assessments and management, requirements and rules, and technology decision. In section four, the paper focuses on the system architecture design, database design, and interface design. In section five, some further developments of this system are briefly discussed.

2. Related Works

Many research groups and companies are engaged in developing systems to produce lecture internet broadcasting by using streaming video technology. One group concentrates on a lecture browser based on student needs (Rowe, Harley, Pletcher, and Lawrence, 2001). Another group focuses on using streaming video to provide unclassified courses in a web-based course (Semrau, 2001). Others provide archive lecture streaming video to their students (Convera, 2003).

The Lecture browser system (Rowe, Harley, Pletcher, and Lawrence, 2001) is different from the system discussed in this paper. It is a web-based player to present users the lecture stream, presentation slides, an index to the slide titles, and a keyword search interface. This system automatically changes slides when
streaming video moves to the next topic, and it also provides forward and backward functions to control streaming video. The index function allows users to locate a special topic and play the streaming video and slides on that topic. The keyword search function supports search for selected words within the lecture archive.

An article published by California State University (Semrau, 2000) presents a system that uses a phone line to connect to a particular website. The website stores some lecture video used for some online courses. This group uses manual methods to edit all videos. They also provide a separate “chat room” between students and instructors for interactivity.

Convera Company working with California State University announced that they have developed a system that provides video archives of courses. Users can automatically capture video and browse course summaries. They can also search for precise streaming video clips using text or image clues. In addition, they can track voice sound by using this system. This system also provides catalog content using metadata and annotations.

3. System Analysis

3.1 User Description

3.1.1 Current users
At the present state, the target users of the video application system of this system can be divided into two groups: administrators and students. Administrators are the users who have permissions to manage this system. Currently, only the database administrator who develops this system in the School of Education and Terri Buckner, who supports this system, are in the administrative user group. Several students will be added to the system to test all functions of the system.

3.1.2 Expected changes in user groups

Because of the development of this system, future user groups will be different from the current user groups fundamentally. First, we need some professional technical personal to manage the whole system, including the database and software. So only this person has the permission to access the whole system. Second, the function of instructors to use this system to assist their on line courses will expand. The instructor group contains faculty and teaching assistants. Third, students taking courses on this system will become the major user group. Finally, we will let guests visit some assigned course discussion forums.

3.1.3 Characteristics of Target users

Students:

- **Attitudes**
  Students do not like to spend time learning new systems. They expect and welcome a system that is easy to learn and use.

- **Computer literacy**
Students should have basic knowledge about computers and the Internet.

- **Video Player**
  Students should only use real player as the video player, because the system is designed for the real player client.

- **Physical system**
  Students can use the system in computer labs and at home.

**Instructors:**

- **Attitudes**
  Instructors have a major interest in video application and expect to improve their educational methods by using new technology.

- **Computer literacy**
  They should be familiar with the Internet.

- **Video Player**
  They should only use real player as the video player.

- **Physical system**
  They can use this system in their office or at home.

**Administers:**

- **Attitudes**
  They are interested in web database programming and streaming video.
  They also want to explore new methods by helping educators use new technology.

- **Computer literacy**
They should be familiar with UNIX, MYSQL server, PHP, and JavaScript programming languages.

- **Video Player**
  They should only use real player as the video player.

- **Physical system**
  They can work from his office or at home through FTP or F-security SSH transfer/client or other remote transfer software.

**Guests:**

- **Attitudes**
  They are interested in finding useful information about a specific course or related research in combining multimedia with on line courses.

- **Computer literacy**
  They should have basic knowledge about computers and Internet.

- **Video Player**
  They should only use real player as the video player.

- **Physical system**
  They can visit the system web pages anywhere.

**3.1.4 User Responsibilities**

This system currently provides user tasks for students and administers. These user tasks are used as the guidelines for designing system functionalities and features.
Students’ Responsibilities:

After a student logs in to this system, first he/she should select a course that he/she is registered in; second he/she should select a video discussion forum that he/she wants to participate in; then he/she can view all comments posted on this forum, and he/she can also post his/her comments to the forum.

Administrator’s Responsibilities:

a. Add student information:
   Create an account for a student and add student information into the database, such as the student’s name, phone number, email address, password.

b. Update student information:
   Change a student’s information as needed. For example, help the student change his/her password.

c. Delete student information:
   Delete a student’s information when the student is no longer registered according to the instructor’s notification.

d. Add employee information:
   Create an account for an employee and input employee information to the database. Employee includes administrators, faculty, and teaching assistants.

e. Update employee information:
Update an employee’s information as needed.

f. Delete employee information:
Delete an employee’s information if the employee does not use this system any more.

g. Add course information:
Add course information to the database, such as which year a course is available at this system, its section, and its instructor’s name.

h. Update course information:
Update course information if there are any changes or any error.

i. Delete course information:
Delete course information if a course is no longer registered to this system.

j. Add video information:
Add video information to the database such as video name and the length of a video.

k. Update video information:
Update video information if there are any errors or changes.

l. Delete video information:
Delete video information if a video is not used by any courses registered in this system.

m. Assign courses to students:
Since this system will be used for several courses, and each course will contain different students who register the course, the administrator can assign permission to the students to access their enrolled course.
n. Assign videos to courses:

Each course discussion forum will include different videos, so the administrator will assign videos to each course discussion forum according to the instructor’s notification.

o. Delete video form course:

If an instructor does not want to use a video in his/her course discussion forum, he/she can ask administrator delete it from the course video list.

p. View comments:

The administrator can monitor every course discussion forum and view all posted comments.

3.2 Risk Assessments and Management

In this project, the following describes the risks which might occur in the design and implementation:

- Because of program errors files become corrupt and result in loss of data.
- Files are accidentally or deliberately delete or changed mistakenly by administrators or overwritten with old version files.
- Server might be down, so the system is inaccessible.
- Database moves from one server to another server so that it makes the system unreachable.
- The system is accessed by unauthorized individuals such as hackers or crackers.

Based on the above possible risks, Preclusion approaches as follow are taken:
• Backup all programming files, database files, and interface files in another separate computer and network system.

• Store the master files in another account.

• Print out and store the most updated files in a different place.

• Copy all these files to a disk.

• Use PHP session with user name and password to control the access to this system.

3.3 Business Requirements and Rules

The following is a description of business requirements and rules about the digital video application system:

• The capacity of the system must be large enough to store all data.

• The database must enforce certain security standards. User information must be kept confidential.

• The database must allow concurrent data manipulation, such that at least 10-20 users can simultaneously log into the system.

• Rules and access permissions must be set based on user responsibilities. User roles must be associated with login/password values such that these permissions are enabled on logging into the system.

• Only the administrator can access all system, add, update, and delete data stored in the system.

3.4 Technology Decision/Justifications
3.4.1 Personal Home Pages (PHP)

Since this system requires a lot of programming, Personal Home Pages (PHP) is chosen as the middleware to connect the database and the web. Reasons are listed below for choosing PHP:

- PHP is designed for creating robust and reliable dynamic Web pages and is open-source and free. Almost all scripts are free and supported by a forum so that help is conveniently available.
- PHP runs on the server side, so it is invisible to the user.
- PHP is an extensible language because it can integrate with other component models such as Java and C++ objects.
- PHP is easy to use since it brings many built in features.
- PHP supports all major platforms such as UNIX and Windows, and its features support most popular databases.

3.4.2 MYSQL Database

The MYSQL database package is chosen to create the relational database management in digital video application for the following reasons:

- MYSQL is often used with the popular open-source language—PHP.
- MYSQL is a very fast database system. Because MYSQL was originally designed for the purpose of querying data at a fast pace. It is a true multi-threaded database server that excels at retrieving information.
- MYSQL is very functional.
• MYSQL is open-source released under the GUN Public License. From this point; any individual can practice coding without any monetary compensation.

• UNIX on which MYSQL runs is more reliable than other systems, like Windows.

3.4.3  JavaScript

JavaScript is one of the most popular client-side scripting languages. It can also be used on the server side and supported by Internet Explorer. In addition, JavaScript is a powerful language because it can create objects and functions at run time. JavaScript runs on the user’s browser rather than on the server side. This point is very important for this system because one of major goals of this system is to play streaming video on line. Therefore, JavaScript is used in the PHP pages of the web interface to deal with controlling video playing and ensure valid data input.

4  System Description

4.1 System Architecture Design

Figure 1 shows the architecture of the digital video application system. This system contains three servers: a web server, a MYSQL database server, and a streaming video server. The web server is used for storing web pages with HTML
files embedded with PHP script and JavaScript. The web server does the work of translating the PHP script, inserts, deletes, and updates data that are stored in the MYSQL server. All videos that will be used in this system are stored in a streaming media server. Clients run a program on their computers, called a client player. It means the streaming server sends the video to clients. This system adopts commercial software from Real Networks for the streaming media server and the client player.

*Figure 1: Digital Video Application System Architecture*
4.2 Database Design and Implementation

The data stored in the database contain user information, course information and information related to videos. Information related videos include video description and comments posted for the video. The whole database includes eight tables. Two tables store user information: the Student table and the Employee table; three tables store course information: the Course table, the CourseToken table, and the CourseVideo table; three tables store video information: the Video table, the Body table, and the Header table. Figure 2 shows the database schema diagram of the digital video application system and this schema is implemented in MYSQL database system.
Figure 2: Database Schema Diagram

The Employee table (table 1) contains the following information: employee name, phone number, email address, username, password, and user type. Each employee in this system is assigned a username and a password for logging in to this system so that the system can prevent from unauthorized access. Each employee is also assigned
to a user type in order to control the employee’s permission to access. For example, administrator can access the whole system, but a teaching assistant can only access a related course for which he/she is registered.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>eID</td>
<td>The employee identifier number. <em>Primary Key</em></td>
<td>Int(11)</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>fname</td>
<td>The employee’s first name</td>
<td>Varchar(30)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>lname</td>
<td>The employee’s last name</td>
<td>Varchar(30)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>phone</td>
<td>The employee’s phone number</td>
<td>Varchar(20)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>email</td>
<td>The employee’s email address</td>
<td>Varchar(50)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>username</td>
<td>The employee’s user name for login in</td>
<td>Varchar(8)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>usertype</td>
<td>The employee’s role (TA, faculty, or administrator)</td>
<td>Varchar(20)</td>
<td>yes</td>
<td>null</td>
</tr>
</tbody>
</table>

Table 1: *employee table description*

Table 2 presents the information about the Student table. The Student table includes student name, phone number, email address, username, and password. The username and password are used to log in to the system.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>studentID</td>
<td>The student’s identifier number. <em>Primary key</em></td>
<td>Int(11)</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>sfname</td>
<td>The student’s first name</td>
<td>Varchar(30)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>sname</td>
<td>The student’s last name</td>
<td>Varchar(30)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>phone</td>
<td>The student’s phone number</td>
<td>Varchar(20)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>email</td>
<td>The student’s email address</td>
<td>Varchar(50)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>username</td>
<td>The student’s user name for login in</td>
<td>Varchar(10)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
<td>Type</td>
<td>Null</td>
<td>Default</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------</td>
<td>----------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>courseID</td>
<td>The course identifier number. <em>Primary key</em></td>
<td>Int(5)</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>cName</td>
<td>The course’s name</td>
<td>Varchar(255)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>cYear</td>
<td>The course available year.</td>
<td>Varchar(6)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>semester</td>
<td>The course available semester</td>
<td>Varchar(20)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>section</td>
<td>The course section number</td>
<td>Char(3)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>eID</td>
<td>The course’s instructor. <em>Foreign key references the table employee</em></td>
<td>Int(11)</td>
<td>Not null</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 2: student table description**

As shown in table 3, the Course table contains information about courses, such as course name, course section number, which year and semester the course is available, and who is the course instructor.

The Coursetable table (table 4) is a relationship table that represents the relationship between the course and the student. It means that this table provides information of which course(s) the student taken. So this table only has two fields: courseID and studentID.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>courseID</td>
<td>The course identifier number. <em>Combined primary. Foreign key references the table course</em></td>
<td>Int(5)</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>studentID</td>
<td>The student’s identifier number. Combined primary key. Foreign key references the table student.</td>
<td>Int(11)</td>
<td>Not null</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 4: course token table description**

Table 5 is also a relationship table. It represents which course contains what videos. Since this system will generate a forum for each video contained in each course, The forumid is created in the table. Therefore, Coursevideo table has three fields: forumid, videoed, and courseID.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>forumid</td>
<td>Course discussion forum identifier number. Primary key Auto increment number</td>
<td>Int(10) unsigned</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>videoID</td>
<td>The video identifier number. Foreign key references the table video</td>
<td>Int(3)</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>courseID</td>
<td>The course identifier number. Foreign key references the table course.</td>
<td>Int(5)</td>
<td>Not null</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 5: course video table description**

The Video table (table 6) stores video information, such as video title, length, and link. Since this system will store all video at separate streaming video server, the link field can record the URL of a video.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>videoID</td>
<td>The video identifier number. Primary key</td>
<td>Int(3)</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>title</td>
<td>The video title</td>
<td>Varchar(255)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
<td>Type</td>
<td>Null</td>
<td>Default</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------</td>
<td>-------------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>parent</td>
<td>The postid of the parent comment</td>
<td>Int(11)</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>poster</td>
<td>The author of the comment</td>
<td>Varchar(10)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>title</td>
<td>The title of the comment</td>
<td>Varchar(20)</td>
<td>Not null</td>
<td></td>
</tr>
<tr>
<td>children</td>
<td>Number for represent whether the comment has replied comments</td>
<td>Int(11)</td>
<td>Not null</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6: video table description

Both table Header and Body are used to represent the information about posted comments. According to the requirement of the Education School, this system should create a tree structure to show all replied comments together with a parent comment. This system will do many small queries to build the tree structure. But because the type of comment’s body (message) is text, this will slow down the information retrieval process. In order to avoid this problem, the comment’s message and video start time are separated from the Header table. The Header table contains the following information: postid of the parent comment if it has parent (parent), the author of the comment (poster), comment title (title), the date and time of the comment was posted (posted), whether this comment has replied comments (children), how many chat are used within one application (area), the comment belongs to which discussion forum (forumid), and its postid in this forum (findex).
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>posted</td>
<td>The date and the time that the comment was posted.</td>
<td>datetime</td>
<td>Not null</td>
<td>0000-00-00 00:00:00</td>
</tr>
<tr>
<td>postid</td>
<td>The identifier number for each comment. <em>Primary key auto increment number.</em></td>
<td>Int(10)</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>forumid</td>
<td>The forum identifier number. <em>Foreign key references the table courvideo.</em></td>
<td>Int(5)</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>findex</td>
<td>The index number for each comment in its posted forum.</td>
<td>Int(5)</td>
<td>yes</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 7: header table description*

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>postid</td>
<td>The identifier number for each comment. <em>Primary key.</em> <em>Foreign key references the table header.</em></td>
<td>Int(10)</td>
<td>Not null</td>
<td>0</td>
</tr>
<tr>
<td>message</td>
<td>The body of the comment.</td>
<td>text</td>
<td>yes</td>
<td>null</td>
</tr>
<tr>
<td>videotime</td>
<td>The user input video start time.</td>
<td>Varchar(20)</td>
<td>yes</td>
<td>null</td>
</tr>
</tbody>
</table>

*Table 8: body table description*

### 4.3 Interface Design and Implementation

According to different users with different permissions to access this system, the interface design section could be divided into three parts: interface design for administrator, interface design for student, and interface design for the discussion forum. The following is the description for each section in detail.
First of all, the Log In page (Appendix 1) is described since this is the first page shown to anyone who wants to use this system. The key purpose for creating the log in page is to prevent unauthorized access to this system. Different users input their own user names and passwords to enter different interfaces respectively.

4.3.1 Interface design and implementation for administer

According to requirements of the Education School, for this system, three catalogs for the administrator are designed: Account Management, Course Management, and Video Management. When an administrator uses his/her user name and password to log in this system, the first page is the menu page (appendix 2) that shows three catalogs, with the functions to view a discussion forum, and log out.

4.3.1.1 Account management

There are four functions in the account management section: create faculty/TA account, modify faculty/TA account, create student account, and modify student account.

- Create faculty/TA account

In the Employee Registration page (appendix 3), an employee’s information such as first name, middle name, last name, phone number, email address, password, and user type should be recorded. Since this system can automatically use the input email address as the user’s user name, users do not input his/her user name. The system also lists four user types at a drop down
list to let user select. Employee ID as the primary key for employee table will be automatically added by the system. After click the “Go” button, the Employee Registration Confirmation page (appendix 4) will show all input information to the user.

- Modify faculty/TA account

From the point view of security, currently this system only allows users to modify their own information, except the employee ID. As shown in appendix 5, users can re-enter their own information if they find errors or their personal information is changed. After clicking the “Do Update” button, the updated information will be presented in the Update Employee Information Confirmation page (appendix 6).

- Create student account

The administrator can create an account for a student as shown in appendix 7. After inputting a student’s first name, middle name, last name, phone number, email address, and password, and clicking the “Register” button, a student account is created. Student ID will also be added by the system as the primary key for the Student table. Then the Student Registration Confirmation page (appendix 8) lists all input data.

- Modify student account

Since the system contains many student’s accounts, if we want to modify one of the student’s accounts, page one of the Update Student Information (appendix 9) will show all students’ names in a drop down list. After a student name is selected, page two of the Update Student Information (appendix 10)
will present all information of the student. The administrator can change the information as needed, but he/she cannot change the student ID. After clicking the “Do Update” button, the Update Student Information Confirmation page (appendix 11) will list all modified information.

4.3.1.2 Course Management

The Course Management section provides three functions: create new course, modify course, and add students for course.

- Create new course
  According to system needs, course information includes course name, course open year, course open semester, and faculty name in the Add Course page (appendix 12). After clicking the “add” button, all information about the course will be listed at the Add Course Confirmation page (appendix 13). The system automatically creates a course ID for a newly added course.

- Modify course
  Because the system contains several courses, if the administer wants to modify course information, the page of View All Course (appendix 14) is the first page for this function. This page displays a course table that lists all course information contained in the system with “Delete” and “Edit” links in the same row. If the administer wants to delete a course, he/she simply clicks the Delete link in the same course row, the course will then be deleted from the Course table (appendix 15). When the “Edit” link from a course list is clicked, the Modify Course page (appendix 16) will list all the course information. Except course ID, other information can be changed. After
clicking the “submit” button, updated course information is listed in the Update Course Confirmation page (appendix 17).

- Add student for course

It is possible that a student takes several courses in the system at the same time, so student names and course names are listed in two drop down lists (appendix 18). According to course registration information, the administrator can add any student to any course in which he/she is enrolled. If adding a student to a course is successful, a confirmation page (appendix 19) will appear.

4.3.1.3 Video Management

The Video management section also contains three functions: add new video, modify video, and add video to existing courses.

- Add new video

The system needs information about the video title, total time of video and video link for a new video (appendix 20). When the “Add” button is clicked, a confirmation page of Add New Video (appendix 21) displays all information about the video. The video ID is automatically added by the system.

- Modify video

Like Modify Course page, Modify Video page (appendix 22) is a table that lists all data for all videos. And this information can be edited or deleted. Simply clicking a “delete” link of a video will delete the video from the database (appendix 23). If the administrator wants to modify video information, he/she only needs to click “edit” link in the same row with the
video information. The Update Video page (appendix 24) will then display all information about the video. Except video ID, other data can be modified as needed. Then the Update Video Confirmation page (appendix 25) will display all information about the video after changes.

- Add video to existing courses

Because the system allows one course to contain several different videos, and one video can also be added to different courses, all video titles are listed in a drop down list, and information about course name, course open year, course open semester, and course section is listed in a course drop down list (appendix 26). The administrator can add any video to any course. After clicking the “Submit Query” button, a confirmation page of Adding Video to Existing Course will display (appendix 27). At the same time the forum ID will be automatically added by the system while a discussion forum is created with the newly added video. For example, if the system has three courses and every course contains four videos, then twelve discussion forums are generated by the system.

Since design and implementation of the View Forum in the administrator’s section are very similar to that of the student part, they will be discussed in the parts of interface design and implementation for students.

4.3.2 Interface design and implementation for student
The major purpose for students using the system is to post or view comments in the system, so the student pages are very simple. After the student logs in, the system will display all course names that the student is registered in according to the student’s user name and password (appendix 30). Based on the selection of the course, the system presents all videos included in the course (appendix 31). Then the student can select a video discussion forum to enter. In order to control the whole system access, the system only allows a student to access the discussion forum in which he/she is registered. This is different from that of the administrator who can view all courses (appendix 28), and select any video contained in the course (appendix 29).

4.3.3. Interface design and implementation for discussion forum

All users can access one or more discussion forums, so the forums are designed separately. Although the system contains many different discussion forums, the interface and function design are similar. Every discussion forum has View the Tree of Comments page, Post New Comment page, Display a Comment page, and Reply to an Existing Comment page.

- View the tree of comments

According to the requirement of the Education School, the system uses a tree structure to display all posted comments in a video discussion forum (appendix 32). This page contains a comment table with tree structure and a video player for playing the video. There is a control panel below the video player so that the user can play, stop, reward, and forward the video. The
comment table includes three buttons: New Post, Expand, and Collapse. When the “Expand” button is clicked, all replied comments with their parent comments will be displayed to the user. At the same time, there is a “-” sign displayed before all parent comments. If a “-” button is clicked, all replied comments of the parent comment will disappear from the page. Unlike the “Expand” button, the “Collapse” button is used to hide all replied comments from the page, regardless of their parent comments. Simultaneously, a “+” button will be displayed in front of every parent comment that has replied comments. When the user clicks a “+” button, the next level replied comments will be displayed as shown in appendix 33. This design can help user conveniently search the comments they want to view.

- Post new comment

A user can use the “New Post” button to post a new comment for the video. As shown as appendix 34, the Post New Comment page contains a post form and a video player. The post form has three blank fields: Your Name, Message Title, and Video Start Time. The system will automatically fill in Your Name field with the user name. The default title “New post” is filled in the Message Title field. Both Your Name and Message Title can be changed as needed. But the Video Start Time field is empty and it must be filled in with the time. Below the three fields, there is a text form used to type comments. The video player with control panel plays the video on the right side of the page. All fields should be filled in, otherwise, an error message
will display below the table. It means that the comment failed to be posted in
the discussion forum.

- Display a comment

A user can click any comment title listed on the View the Tree of Comments
page, then the Display a Comment page (appendix 35) shows the specific
comment with the posted time, video start time, author of the comment, and
comment title on the left side of the page. Reply and Index buttons are also
listed on the left side. The Index button is used to return to the View of the
Tree of Comment page, and the Reply button is designed to post reply
comments for the currently displayed comment. On the right side, a video
player is playing a video. Below the video player, there is a Play at Special
Time button used to play the video at the assigned special time listed on the
left side of the table. All replied comments of the current comment are also
listed on the page.

- Reply to an existing comment

The Reply to an Existing Comment page (appendix 36) is very similar to the
post new comment page. There are only two differences: (1) the default
message title is “re:” with its parent message title; (2) all the parent comments
are displayed in the text form.

4.4 System Design Idea and Technology Challenges

4.4.1 Design Idea
According to the needs of the Education School, the current system contains three sections: administrator section, student section, and course discussion forum section.

Since the Education School requires that this system be used by several courses and each course can contain one or many videos, different users have different permissions to access the system. Accordingly, three parts are developed in the database: user account, course, and video. In this way the system can add users to courses and add videos to courses conveniently as needed. Because of different access permissions, currently access for two level users is developed based on the user name and password.

Right now, the system is designed to generate one discussion forum for each video in every course, so a forumid is set as a primary key for each video in every course. For example, the system contains two courses: EDU 110 and EDU120. Course EDU 110 includes videos: Green World, Blue Sea, and Gone With the Wind; and EUD 120 has videos: Green World, Gone With the Wind, and Red Mountain. The system will generate six forumids. Thus the system will contain six course discussion forums, regardless of duplicated videos in different courses.
Although all users are divided in two groups with different access permissions, all users should be able to view course discussion forum(s), so the discussion forum is designed as the public section for all users.

4.4.2 Technology Challenges

Many challenges emerged during the design process. A few of the most difficult challenges was how to develop a tree structure to show all comments, how to play a video according to a user-assigned start time, how to generate dynamic numbers for the discussion forum, and how to place a comment back to its posted forum.

- View comments with a tree structure

In order to accurately represent the relationship between the original comment and the replied comments, it is necessary to adopt multiple threads to display all comments posted in a discussion forum. However, how to record the relationship among comments in a database is a difficult question. Besides Post ID used as the primary key in the Header table, Parent and Children fields are added. The Parent field is used to record a comment post ID if this comment is a parent for one or many replied comments. The Children field is used to show whether a comment has replied comments or not (children). For example, as shown in table 9 the Header table has a comment whose post ID is 3 and the comment has three replied comments whose post ID are 5, 6, 7 respectively. At this time, the
children field in the Header table for comment 3 is 1, which means the comment 3 has replied comments. And its parent field is 0 since comment 3 is an initial comment without parent. While comment 5 and 6 are replied comment for comment 3, their parent filed is comment 3 with post ID “3”. Because there is no replied comment, their children fields are “0”. However, although comment 7 is also a replied comment for comment 3, it has its own replied comment: comment 8. So its parent field is “3” and children field is “1”; and parent field of comment 8 is “7” and children field of comment 8 is “0” due to no replied comments. So parent field contains an identified value (postid) and children a Boolean value (children or not). This design is implemented by coding PHP and SQL statements to insert and update the necessary data. In the PHP script, a tree node is created to execute SQL statements.

<table>
<thead>
<tr>
<th>Post ID</th>
<th>parent</th>
<th>children</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 9: example for header table (post Id, parent, and children)*

- Play a video at a dynamic start time
Playing a video at a dynamic start time is the core task in the system and it was also the most difficult challenges during the system development. First, SMIL is used to control the video start time since the SMIL language has the functions to manipulate video start time and end time. However, SMIL language can only generate static files but not dynamic file to control playing video and it can not be embedded in a PHP script, so SMIL research is transferred to HTML + Time research. HTML + Time works well with PHP script. Some built in functions in HTML + Time are used and are embedded with PHP scripts to develop the system which can control the start time for playing a video. HTML + Time can only control AVI format video, however the Education School hopes that the system can maneuver RAM format videos which can be played by both real player and media player. Therefore, JavaScript was added with some built in functions in real player to operate the start time for playing a video. Finally, a comment message is displayed with its start time assigned by the user on the left of the View Comments interface and a button is added under the real player on the right of the same interface. This design only needs the user to click the button and trigger the button event. Then PHP coding working with SQL statements will obtain a dynamic start time from the database and send the start time to the JavaScript function to operate the start time of a playing video.

- Generate dynamic number of the discussion forum and place a comment back to its forum position
The system would contain multiple discussion forums because one discussion forum is created for each video in every course. Since the number of the course and video in the system is dynamic, it is not possible to create a table for each discussion forum. So it is difficult to allocate all posted comments into their posted discussion forums. Forum ID is designed for the purpose to classify comments into different forums. For instance, the system can display all comments with forum ID “3” by executing SQL select statements.

However, another problem appears since all comments with the same forum ID have discrete post IDs. So it is difficult to list all comments with the same forum ID in order and correctly represent the relationships among initial comments with their replied comments. The “Findex” field is added in the Header table to deal with this problem. The default value of “findex” is 0 since “0” is used as the first index for all first comments in every discussion forum, and “1” as the second index for all second posted comments, and so on. Therefore, the system can use forum id to select all comments posted in a specific discussion forum, then according to their findex to display them in a tree structure.

5 Future Works
Currently, this system will be used as a basic video application discussion forum since most essential features and functionalities are completed to let users post or view comments as well as play video or control playing video at the same time. However, in the future, more features and functionalities will be added to the system. This section will simply discuss these features and functionalities as future work.

It will be better for the system if we add a function for automatic time records for users when they find some interesting points and want to post their comments. Currently, the system asks the user to enter the start time manually. If we add an automatic time record function for users, this function can conveniently help them finish their postings without turning attentions away from the content. Another point we should consider is that the system could add the function that makes the users post multiple comments and start time at one time. Currently, the system only allows the users to post one comment with a start time.

As mentioned in section three, “future user discussion”, the system could adopt four user levels. So the system should distinguish administrator from instructors and grant their different permissions to access the system. Since we do not discuss functions for different user groups in detail, we currently developed all functions for administrator who can access the whole system. In the future, we should only assign some functions for instructors or guests.
Some advanced functions were also discussed during the system development. For example, the system could display multiple video discussion forums at the same time. There are two suggestions for this point: one is that the system can show all video discussion forums for one course; the other is that the system can display multiple discussion forums based on user needs.

6 Conclusion

This project developed a novel working prototype based on specified needs for a community of users. Several technical challenges are faced and overcome.

7 Acknowledgements

Thanks to Terri Buckner, Gary Marchionini, and Xiangming Mu.
8 Reference


http://www.acm.org/chapters/los_angeles/Archives/laacm0005.html

9 Appendices:

Appendix 1: Log in Page

Appendix 2: Menu page
Appendix 3: Employee registration page

Appendix 4: Employee registration confirmation page
Appendix 5: modify employee page

Appendix 6: Update employee information confirmation page
Appendix 7: student registration page

Appendix 8: student registration confirmation page
Appendix 9: update student information page one

Appendix 10: update student information page two
Appendix 11: update student information confirmation page

Appendix 12: add course page
Appendix 13: add course confirmation page

Add New Course confirmation
The following is the new course information that has been inserted:
course ID: 11
Course name: inls392
Course open year: 2003
Course open semester: fall
Section: 001
Professor ID: 16

Appendix 14: view all course page

View All Courses

Course Table:

<table>
<thead>
<tr>
<th>courseID</th>
<th>name</th>
<th>year</th>
<th>semester</th>
<th>section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>Delete</td>
<td>1</td>
<td>256</td>
<td>spring</td>
</tr>
<tr>
<td>Edit</td>
<td>Delete</td>
<td>2</td>
<td>201</td>
<td>fall</td>
</tr>
</tbody>
</table>
Appendix 15: delete course confirmation page

Appendix 16: modify course page
Appendix 17: update course confirmation page

The following was updated
CourseID: 1
Course name: inls258
Year: 2003
Semester: spring
Section: 01
eID: 2

Appendix 18: add student for course page
Appendix 19: add student for course confirmation page

Thanks! Student register course successful! staff page

Appendix 20: add new video page

School of Education
University of North Carolina at Chapel Hill

Item with "*" is required.

*Video Title:
*Video Length: 15 min
*Video Link: location/movies/red.ram

Menu  View video  View courses  Logout
Appendix 21: add new video confirmation page

The following is the new video information that has been inserted:
Video ID: 4
Title: red
Length: 16 min
Link: www.ils.unc.edu/~chenl/education/movie/red.ram

Appendix 22: modify video page

Video Table:

<table>
<thead>
<tr>
<th>videoID</th>
<th>Title</th>
<th>Length (min)</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>Delete</td>
<td>red</td>
<td>16</td>
</tr>
<tr>
<td>Edit</td>
<td>Delete</td>
<td>test2</td>
<td>11</td>
</tr>
<tr>
<td>Edit</td>
<td>Delete</td>
<td>green</td>
<td>120</td>
</tr>
</tbody>
</table>
Appendix 23: delete video confirmation page

Appendix 24: update video page
Appendix 25: update video confirmation page

The following was updated

VideoID: 3
Title: green
Length: 12mins
Link: www.ils.unc.edu/~chenl/education/movie/green.ram

Appendix 26: add video to existing course page
Appendix 27: add video to existing course confirmation page

Appendix 28: administrator view forum page one:
Appendix 29: administrator view forum page two

Appendix 30: student page one
Appendix 31: student page two

Appendix 32: view the tree of comment page
Appendix 33: view the tree of comment page (2)

Appendix 34: post new comment page
Appendix 35: display a comment page

This point is very interesting. Please view it.

Appendix 36: Reply an existing comment page
Appendix 37: log out page

School of Education
University of North Carolina at Chapel Hill

You have successfully logout.
Click [here](http://its.unc.edu/~che/education/logout.php) to login again.