Design and Accomplishment of BS-based Supervisory System for Student Assignment Online

Xiao Wang
Information Engineering Department
Tianjin Bohai Vocational Technology College
TianJin, P.R.China
adawangyuada@163.com

Jiashan Wang
Information Engineering Department
Tianjin Bohai Vocational Technology College
TianJin, P.R.China
18622632016@163.com

***Abstracts*—Computer education has been developed into education of information technology with the rapid development in information technology represented by computer and Internet in the world today. Schools employ a load of approaches to investigate the efficiency in computer and information technology education, and assignment setting is one of those powerful approaches. The particularity of this approach is apparent compared with other traditional curriculums, especially assignment submission and management online. Traditional paths for assignment submission are not compatible with programming IT courses and are exposed to numerous shortcomings. Therefore the establishment of a set of Internet-based supervisory system which features convenience, speed, applicability, security and reliability for assignment management is pretty important.**

This thesis conducts a briefing on the general process development for web-based supervisory system for assignment management, totally specifying in details for programming completed in three phases named system analysis, system design and system implementation. Computer supervisory system for assignment management is inserted the latest B/S (browser/server) structure of widespread popularity, which functions in handing in assignment of IT courses to teachers in the cyber space for teachers' review and download on the Internet. Some aiding functions are also replenished adding to the submission core, assignment deletion, checking of system assistance, comment affixed and checking of registration roll list, just name a few. In realizing functions aforesaid, the foreground resorts to ASP (Active Server Pages), which is launched by Microsoft with nature agreed to dynamic server pages on the Internet. The background applies SQL Server 2000 data database management system that features high safeguarding security and comparatively more satisfactory function integrated to the Internet.

Keyword- supervisory system for assignment management, ASP, SQL Server 2000, B/S structure

Rapid development in information technology in nowadays urges everyone to learn and apply information technology. Schools employ a number of approaches for investigating teaching feedback on computer programming courses, and assignment is one of those methods featuring effect and quantitative applicability. Compared to the traditional subjects, computer courses are definitely distinctive from the way they are submitted and handed in. Traditional paths for assignment submission applied in computer courses will be disadvantageous, and the establishment of a web-based supervisory system for assignment management featuring convenience, speed, applicability, security and reliability is quite to the note.

All in all, it is necessary to probe into a set an effective and practicable supervisory system for assignment management. By implementation of such system, teachers and students are provided with platform inbuilt with service targeted to teaching and learning methods, which also in return provides practical information to the division of teaching management.

I. TECHNOLOGIES APPLIED IN THE SYSTEM

Supervisory system for computer assignment management introduces the latest B/S structure (Browser/Server) of widespread popularity recently. In realizing functions aforesaid, the foreground resorts to ASP (Active Server Pages), which is launched by Microsoft with nature agreed to dynamic server pages on the Internet. The background applies SQL Server 2000 data database management system that features high safeguarding security and comparatively more satisfactory function integrated to the Internet.

II. ANALYSIS OF SYSTEM DEMANDS

The team for the projected topic has successfully collected all-directional ideas and opinions from the office of teaching affairs, all departments and partial teaching faculties, and decided to simplify some of the functions embedded in the web-based supervisory system for computer assignment submission given real difficulties in limited R&D competency and resources. We shift our attention to the practicability of functions trying to complete project development as scheduled and apply this system in reality within the shortest intervals.

The supervisory system for computer assignment management is composed three subsystems namely the administrator subsystem, the teacher subsystem and the student subsystem. Primary functions designed for the system include:

A. The student subsystem

The student subsystem is implemented in reality after realizing such functions as checking teachers' assignment, assignment submission, message written at departure, checking of assignment marks, review on teachers' commentary note and password change etc;

B. The teacher subsystem

The teacher subsystem is implemented in reality after realizing such functions as assignment setting, notification post, assignment appraisal and password change etc;

C. The administrator subsystem

The administrator subsystem is implemented in reality after realizing such functions as backstage database management, adding, modifying and deleting student information, teacher information, class information and assignment information etc. in convenience.

The supervisory system for assignment management functions in seven modules that satisfy the most fundamental requirements for system management, assignment setting, assignment submission, assignment appraisal, notification post, massage written at departure, student-checked assignment mark and commentary note affixed to assignment. Each functional module is further refined as follows:

1) System management: user account login and account maintenance included;

2) Assignment setting: teachers are enabled to publicize online the assigned tasks in a moment;

3) Assignment appraisal: teachers are enabled to conduct a review on assignment submission, check content of the assignment for appraisal with commentary note or grade affixed;

4) Assignment submission: students are enabled to hand in their finished tasks by uploading;

5) Message written at departure: students are enabled to inform teachers about the confusing part for specified explanation;

6) Notification post: teachers are enabled to present students assignment description and remark on assignment fulfillment online;

7) Assignment mark and commentary note checked by students: students are enabled to check mark and commentary note matched to each assignment by teachers respectively;

8) Other non-functional requirements: well-developed system functions for use, nice and elegant look on the user interface for operation and introduction of design modularity for increasing code reuse rate.

III. SYSTEM DESIGN

The system is divided into 12 modules according to the analysis of requirements and design for module separately is introduced as follows:

A. User login

Administrators, students and teachers are not allowed to access this system unless they type in the username, password and select the role accordingly on the login interface. The system posts their data typed in to the receiving procedure with due fulfillment in necessary information after pressing the button for submission. The procedure takes some time to determine some specific mode for verification of usernames in line with user roles for identifying the right password. Three modes for verification are included, and they are separately administrator verification, student verification and student verification. A prompt dialog box pops out saying “please re-login for username-password mismatch” once the username is not compatible with the password. It rebounds back to the login interface for data filling after the user clicks OK. Contrarily, if the username and password match, role information of the user and individual data are automatically acquired by the system for recoding in Session before it enters the management platform accordingly. Three platform presets are provided to different user roles specified as administrator platform for operation, teacher platform for operation and student platform for operation.

B. The college administrative module

The college administrative module is contained in the administrative subsystem, including the four functions of adding, modifying, and deleting of basic college information and academic course configuration for the college. The administrator must first clicks the button for adding supplements of college information, accesses the interface for adding information and clicks information submission after fulfillment. The system automatically conveys the college information into data. The administrator enters the college list once the supplementary information is filled up, and the interface presents all colleges on a list. Three buttons are followed after each blank targeted at separate college for course configuration, edition and deletion. Apparently the processing for course configuration must be executed after course maintenance since sometimes the course information is not available, the administrator is therefore not allowed to conduct course configuration. Once any unintended error is discovered in college information by the administrator, the problem is to be solved by another type-in of college information by entering the interface for information editing after pressing the button. The modified college information is once again to be kept in the database after pressing the button for submission. This time the data is stored with due updates.The system rebounds to the interface of college overview list after storing the basic college information. In case that the administrator types in extra information unnecessary due to some cause, please press the delete button following the college information for erasing. It is actually not a genuine act of deleting college information, but revising the field value deleted by making tag field updates out of security. The act of pressing a delete button is a crucial step and each time the user reveals his intent, the system produces a prompt dialog box questioning the user for final confirmation. If it is pre-conspired, the deleting procedure proceeds, or the system returns to the list. This method also applies for preventing button mistakenly pressed. Each class has make arrangements for their curriculums, which are usually fixed and rarely changed with year. Out of that we insert course configuration for each class of each college, which is proved to save mass manual force every year for course configuration compared to the past. The administrator is urged for unified configuration of courses according to requirements each college has formerly handed in, after all procedures for course database maintenance are done. Press the button for course configuration first for access to the interface, where a dozen of lists elaborating course configuration requirements of all classes in every college are presented. Click the button for adding courses and enter the interface, select courses required and fill in respective description and save it in the database. The last step is to return to the course configuration page.

C. *The course management module*

The course management module includes major procedures of course adding, modifying and deleting.

D. *The teacher management module*

The teacher management module includes major procedures of course adding, modifying and deleting.

E. *The class management module*

The class management module includes major procedures of course adding, modifying, deleting and teaching plan configuration.

F. *The student management module*

The student management module includes major procedures of course adding, modifying and deleting.

G. *The notification management module*

The notification management module is subordinated to the teacher subsystem including major procedures of course adding, modifying and deleting.

H. *The assignment management module*

The assignment management module is contained in the teacher subsystem, including five functions specified assignment setting, assignment modifying, assignment deleting, checking of assignment fulfillment and assignment appraisal. The teacher clicks the button for assignment setting and enters the interface, types in and submits the information for saving it in the database. Return to the assignment overview list once the act of adding is done. If there is any mistake discovered in the supplementary information by the teacher, click the button following the piece accordingly for editing, enter the page for assignment information editing and retype in. Press the submission button and save the revised information to the database. Update the original assignment information and rebound to the notification list. In case any assignment is required to be deleted, click the delete button following the blank of assignment information for enacting the order straightforwardly. Click the button following separate assignment and enter the page designed with lists for checking of assignment fulfillment. This page presents a list of student registration roll for reference of assignment fulfillment targeted to each student. A download button is also provided for taking down their uploading in defining the real fulfillment. After checking out, insert commentary note in separate blank for student assignment and click submit after it is done. Save the information to the database and return to the assignment list.

I. *The message reply module includes*

The message reply module includes primary functions of message checking and reply

The teacher is enabled to make replies to students upon receiving the message by pressing the reply button, type in replies and click the submit button and save replies to the database.

J. *The notification checking module*

The notification checking module is contained in the student subsystem. All notifications posted by teachers concerning learning course are accessible to students once they log in using student number and password. Click any notification required for detailed review on the notification page.

K. *The student assignment module*

The student assignment module is also available. The student clicks "my assignment" and enters the page, where a list of assignments will be presented in front. Click the required assignment for details and click the download button in the page of assignment details for individual need. Click the upload button and pass individual assignment onto the server. Click the submit button to save the file in the database after uploading.

L. *Student voice module*

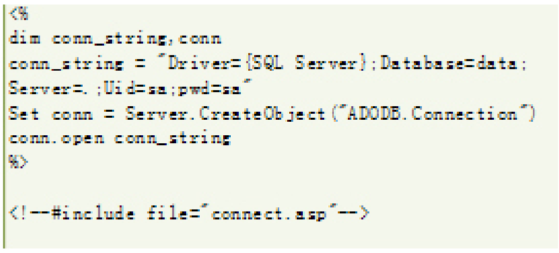
The student message at departure page is entered through my message to realize good contact between students and teaching faculties. Click the button for adding a message and enter the page. Leave a message to the selected teacher and click the submit button. Thus the message is saved in the database.

IV. SYSTEM ACCOMPLISHMENT

Part of dominant long codes are selected and presented hereunder in avoidance of unnecessary elaborates on other codes.

A. *Connection string presets in the universal database*

Each page is practically conducted presets of database connection string due to frequent database connection in different pages of the system. Therefore it is reasonable to reset the data connection string once the data source is removed from the original base. Actually a better choice is to confine all series of connection strings to the file named "connect.asp" and add codes in as follows:



```
<%
dim conn_string,conn
conn_string = "Driver={SQL Server};Database=data;
Server=.;Uid=sa;pwd=sa"
Set conn = Server.CreateObject("ADODB.Connection")
conn.open conn_string
%>

<!--#include file="connect.asp"-->
```

B. *Encrypted MD5 files*

All login passwords in this system are MD5 encrypted out of security. We choose some of codes in this thesis due to the verbose length.

```
Private Function ConvertToWordArray(sMessage)
    Dim lMessageLength
    Dim lNumberOfWords
    Dim lWordArray()
    Dim lBytePosition
    Dim lByteCount
    Dim lWordCount

    Const MODULUS_BITS = 512
    Const CONGRUENT_BITS = 448

    lMessageLength = Len(sMessage)

    lNumberOfWords = (((lMessageLength + ((MODULUS_BITS - CONGRUENT_BITS) \
BITS_TO_A_BYTE)) \ (MODULUS_BITS \ BITS_TO_A_BYTE)) + 1) * (MODULUS_BITS \
BITS_TO_A_WORD)
    ReDim lWordArray(lNumberOfWords - 1)

    lBytePosition = 0
    lByteCount = 0
    Do Until lByteCount >= lMessageLength
        lWordCount = lByteCount \ BYTES_TO_A_WORD
        lBytePosition = (lByteCount Mod BYTES_TO_A_WORD) * BITS_TO_A_BYTE
        lWordArray(lWordCount) = lWordArray(lWordCount) Or LShift(Asc(Mid(sMessage,
lByteCount + 1, 1)), lBytePosition)
        lByteCount = lByteCount + 1
    Loop

    lWordCount = lByteCount \ BYTES_TO_A_WORD
    lBytePosition = (lByteCount Mod BYTES_TO_A_WORD) * BITS_TO_A_BYTE

    lWordArray(lWordCount) = lWordArray(lWordCount) Or LShift(&H80, lBytePosition)

    lWordArray(lNumberOfWords - 2) = LShift(lMessageLength, 3)
    lWordArray(lNumberOfWords - 1) = RShift(lMessageLength, 29)

    ConvertToWordArray = lWordArray
End Function
```

Introduction for use of MD5 encryption: md5 (str)

C. *Applet document uploading*

Document uploading in assignment setting for students and assignment submission employ asp applet uploading. I choose some of the codes out of many.

```
Public function SaveAs(FullPath)
    dim dr,ErrorChar,i
    SaveAs=1
    if trim(fullpath)=″″ or FileSize=0 or FileStart=0 or
FileName=″″ then exit function
    if FileStart=0 or right(fullpath,1)=″/″ then exit
function
    set dr=CreateObject(″Adodb.Stream″)
    dr.Mode=3
    dr.Type=1
    dr.Open
    upfile_5xSoft_Stream.position=FileStart-1
    upfile_5xSoft_Stream.copyto dr,FileSize
    dr.SaveToFile FullPath,2
    dr.Close
    set dr=nothing
    SaveAs=0
  end function
```

Instruction for use is described as follows:

```
set file=upload.file(″file″)
formPath=″../up_files/″
       If  file.filesize>0  Then
              fileExt=lcase(right(file.filename,4))
              If  fileext=″.rar″ Or  fileext=″.doc″  Or
fileext=″.zip″ Or fileext=″.docx″ Then
                     filename=file.FileName

       filename=upload.form(″id″)&year(now)&month(now)&day(now)
&hour(now)&minute(now)&second(now)&fileExt
                     file.SaveAs
Server.mappath(formPath&filename)
              End  If
       End  If
```

V. A SUMMARY OF THE SYSTEM

This system avails secured and reliable assignment submission and checking for the convenience of students and teachers. Still the system is to be further consummated and extended due to a lack of time in terms of unperfected running interface, a lack of openness and uncompleted presets for registration procedure for students and teachers. This system is tested in normal running if less than 200 people are engaged, or otherwise the speed for transmission is markedly slowed down. Additionally, the procedure readability is a little defective on the account of a lack of comprehensive meditation on system analysis and system design. Some functional modules are added to the system in days to come to improvise more functions within, for better and convenience experience for teachers and students.

REFERENCES

[1] Carl Zetie, et al. Practical User Interface Design-Making GUIs Work[M]. USA: McGraw-Hill Book Company, 1995.

[2] Kevin A Schneider, James R Cordy. Abstract User Interfaces: A Mo-del and Notation to Support Plasticity in Interactive Systems[C] .Proc. DSVIS 2001-6th International Conference on Document Analy-sis and Recognition, Glasgow, 2001.40-59.