INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

COURSE MANAGER

Danhui You

A Major Report

In

The Department of Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

April 2003

0-612-78002-3

Canada

ABSTRACT

Course Manager

Danhui You

Course Manager is a web-based application for managing course information, instructor and marker information, student and group information, course marked entities, demo schedules and reservations, and grade information during the whole process of an undergraduate or graduate course. It also provides functionality for students to upload assignments, to conduct peer reviews, and to take online assessments; for markers to grade assignments and projects; and for instructors to schedule assignments and demos, and to prepare online assessments. The Course Manager system distinguishes five categories of users: administrator, instructor, marker, student and project group. Each user category is associated with a specific user access privilege and can only access the functionality within its access privilege

The Course Manager system is designed and implemented with a typical three-tier client-server architecture, which consists of the user interface tier on the client side, and the processing management tier and the database management tier on the server side. The user interface tier is made up of HTML pages on the client side, which control the presentation logic of the application. The processing management tier uses PHP as the scripting language. The PHP scripts are responsible for the application logic of the system. The database management tier uses MySQL as the DBMS, which controls the data logic of the system.

iii

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF FIGURES

1   Introduction

1.1   Course Manager System

Course Manager is a web-based application for managing the following information

during the process of an undergraduate or graduate course:

1) Course information, including course descriptions, sessions, section and course

   options;

2) Instructor and marker information;

3) Student and group information;

4) Course marked entities, including assignments, projects, topics, assessments and

   exams;

5) Progress report meeting and demo schedules and reservations;

6) Grade information;

7) Login information.

The application also provides functionality for students to upload assignments and

examinations, to conduct peer reviews, and to take online assessments; for markers to

grade assignments and projects; and for instructors to schedule assignments, meetings

and demos, and to prepare online assessments.

The application has been deployed on the dumbo server at Concordia University and put

into production for the following course sections since its initial version in 1999:

- COMP353

- SOEN282

- COMP553

Enhancement has been made to the application according the users' feedback.

The application can be found at:

https://dumbo.concordia.ca/crsmgr

1.2   Architecture of Course Manager System

Course Manager is designed and implemented with a typical three-tier client-server

architecture [1]. The three-tier architecture (also known as the multi-tier architecture)

was introduced in the 1990s. It consists of the user interface tier on the client side, and

the processing management tier and the database management tier on the server side.

Compared to two-tier architecture, which has the user interface on the client side and the

database management on the server side and splits the processing tasks between the client

and the server, the three-tier architecture introduces a middle tier to execute business

logic and rules, and to perform queuing, application execution and database staging. The

three-tier architecture provides increased performance, flexibility, maintainability,

reusability and scalability, while hiding the complexity of distributed processing from the

user. It can accommodate hundreds of users (as compared to only 100 users with the

two-tier architecture) and has become a popular choice for web-based applications.

In the Course Manager system (Figure 1), the user interface tier consists of HTML pages

on the client side, which can be accessed by the user with a web browser. The HTML

pages control the presentation logic of the application. The processing management tier

2

of the application uses PHP as the scripting language [2]. Like other dynamic server

pages such as ASP and JSP, PHP scripts are processed by the web server [3]. These PHP

scripts are responsible for the application logic of the system. The database management

tier of the application uses MySQL [4] as the DBMS [5], which controls the data logic of

the Course Manager system.

**User Interface
Layer: HTML Pages**

**Processing Management
Layer: PHP Scripts**

**Database Management
Layer: MySQL Database**
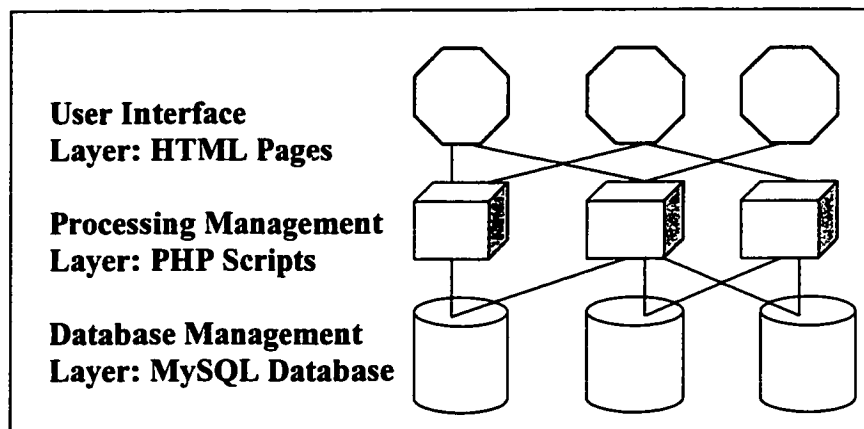
Figure 1 Architecture of Course Manager System

When the user of the Course Manager system requests a page, the web browser

communicates with the Apache web server through the HTTP protocol. The web server

runs PHP scripts based on user requirements and returns processing results in a plain

HTML page to the web browser. If the request requires data from the database, the PHP

scripts will interact with the backend MySQL DBMS. Since the PHP scripts return only

the query result rather than transfer the total file, network traffic is considerably reduced.

1.3   PHP and MySQL

3

PHP scripts play the same roles as any other CGI programs, such as collecting form data, generating dynamic page contents, and maintaining the session. PHP supports a massive number of databases, including Informix, Oracle, Sybase, Solid, MySQL, and PostgreSQL, as well as the ubiquitous ODBC.

MySQL is a small, compact relational database management system ideal for small - and not so small – applications. In addition to supporting standard SQL (ANSI), it compiles on a number of platforms and has multithreading abilities on Unix servers, which make for great performance. MySQL can also be run as a service on Windows NT and as a normal process in Windows 95/98 machines. MySQL also provides multiple APIs for C, C++, Java, Perl, PHP and Python. MySQL server handles concurrency, provides fast access, and guarantees that only authorized users can obtain access. Its privilege and password system is very flexible and secure, and allows host-based verification. Passwords are secure because all password traffic is encrypted when connected with a server.

In addition to being free, the PHP-MySQL combination is also cross-platform, which means you can develop in Windows and serve on a Unix platform. Also, PHP can be run as an external CGI process, a stand-alone script interpreter, or an embedded Apache module. PHP supports a host of other features right at the technological edge of Internet development. These include authentication, XML, dynamic image creation, WDDX, shared memory support, and dynamic PDF document creation.

1.4　Organization of Report

The following chapters describe the Course Manager system in different aspects:

Chapter 2 introduces different user access privileges for the system and the functionality for each of them. It also discusses how the application manages the user access privileges.

Chapter 3 describes the functionality provided by the system in detail and presents the processing and navigation inside each functionality.

Chapter 4 describes the database tables used in the application and the relationships among them.

Chapter 5 concludes the report with some features that can be added to the Course Manager system in the future.

2 User Access Privilege Management

The Course Manager system distinguishes five categories of users: administrator, instructor, marker, student and project group. Each user category is associated with a specific user access privilege and can only access the functionality within its access privilege. When a user is added to the system, his user category is determined and his access privilege is assigned. The user access privilege is a part of the user account information and is stored in the same database table as the user name and password. If a user belongs to different user categories, he will have separate user accounts for each of his user categories. For example, a student who is a member of a project group may have an individual student account and a group account as well. However, he cannot use his group account to access his student information and vise versa.

All the functionality of the Course Manager system is password-protected. The application identifies the access privilege for the user at the time of login. Except the administrator, who has access to all the information in the Course Manager system, the other users can only access the course sections related to their user accounts. For these users, the application will also query the database to get their course section information. The user's identification and course section information is propagated throughout the navigation so that the user can only access the functionality within his access privilege and the course sections related to their accounts.

The administrator has access to most of the functionality provided by the system:

- View, add, modify or delete course descriptions;

- View, add, modify or delete course sessions;

- View, add, modify or delete course sections;

- View, add, modify or delete professors;

- Assign professors to course sections;

- View, add, modify or delete email confirmations for user account creations;

- Change course options for displaying final grades;

- View, add, modify or delete markers;

- View, add, modify or delete students;

- View, add, modify or delete course marked entities;

- View, add, modify or delete topics for course marked entities;

- View, add, modify or delete course groups;

- View, add or delete group members;

- View or discard peer review results;

- View, add, modify or delete time slots for demos;

- Reserve or cancel time slots for demos;

- Choose topics for course marked entities;

- View uploaded assignments;

- View, assign or modify course grades;

- View, assign or modify final letter grades;

- View, add, modify or delete assessments, assessment questions and assessment answers;

- Change anyone's password;

- Get help information.

The professor has access to the following functionality for the course sections that he teaches:

- Change course options for displaying final grades;
- View, add, modify or delete markers;
- View, add, modify or delete students;
- View, add, modify or delete course marked entities;
- View, add, modify or delete topics for course marked entities;
- View, add, modify or delete course groups;
- View, add or delete group members;
- View or discard peer review results;
- View, add, modify or delete time slots for demos;
- View, assign or modify course grades;
- View, assign or modify final letter grades;
- View, add, modify or delete assessments, assessment questions and assessment answers;
- Change his own password;
- Get help information.

The marker assigned to a course section has access to the following functionality for the course section:

- View reserved time slots for demos;

- View uploaded assignments;

- View, assign or modify course grades;

- Change his own password;

- Get help information.

The student registered for a course section has access to the following functionality for the course section:

- View course grades;

- Conduct peer reviews for his group members;

- Upload his assignments or delete his uploaded assignments before the due date;

- Take online assessments;

- Change his own password;

- Get help information.

The project group in a course section has access to the following functionality for the course section:

- Reserve time slots for demos;

- Choose topics for group projects;

- Upload group assignments or delete uploaded group assignments;

- Get help information.

All the pages in the Course Manager system include the gen_include page, which authenticates the user at the login time and propagates the user's access privilege when

9

the user navigates to other pages. The authentication process queries the DB_USER

table, which contains all the account information. The application will also retrieve the

user's course section information by querying the tables in Figure 2. If the user has

several related course sections, the application will display the select_course page (Figure

3) for the user to select a course section. If the user has only one course section, the

application will skip the select_course page and pass the course section information to the

next page. If the user has no related course sections, the application will display an error

message and terminate the navigation.

<<RelationalTable>>
COURSE_DESC

%oCOURSE_DESC_ID : INT
⊘COURSE_NUMBER : VARCHAR
⊘COURSE_NAME : VARCHAR

COURSE_DESC_ID = COURSE_DESC_ID

<<RelationalTable>>
COURSE_SESSION

%oCOURSE_SESSION_ID : INT
⊘COURSE_DESC_ID : INT
⊘SESSION : VARCHAR
⊘YEAR : INT

<<RelationalTable>>
COURSE_STUDENT

%oCOURSE_STUDENT_ID : INT
⊘STUDENT_ID : INT
⊘FIRST_NAME : VARCHAR
⊘LAST_NAME : VARCHAR
⊘EMAIL : VARCHAR
⊘USER_ID : INT

COURSE_STUDENT_ID = MARK_ASSIGNEE_ID

COURSE_SESSION_ID = COURSE_SESSION_ID

<<RelationalTable>>
COURSE

%oCOURSE_ID : INT
⊘SECTION : VARCHAR
⊘COURSE_SESSION_ID : INT
⊘SHOW_FINAL_GRADES : INT

COURSE_ID = COURSE_ID

<<RelationalTable>>
MARK_ASSIGNEE

%oMARK_ASSIGNEE_ID : INT
⊘COURSE_ID : INT

COURSE_ID = COURSE_ID

<<RelationalTable>>
COURSE_PROF

%oCOURSE_PROF_ID : INT
⊘COURSE_ID : INT
⊘PROFESSOR_ID : INT

PROFESSOR_ID = PROFESSOR_ID

COURSE_GROUP_ID = MARK_ASSIGNEE_ID

<<RelationalTable>>
COURSE_GROUP

%oCOURSE_GROUP_ID : INT
⊘NAME : VARCHAR
⊘EMAIL : VARCHAR
⊘USER_ID : INT

<<RelationalTable>>
PROFESSOR

%oPROFESSOR_ID : INT
⊘FIRST_NAME : VARCHAR
⊘LAST_NAME : VARCHAR
⊘EMAIL : VARCHAR
⊘USER_ID : INT

COURSE_ID = COURSE_ID

<<RelationalTable>>
COURSE_MARKER

%oCOURSE_MARKER_ID : INT
⊘FIRST_NAME : VARCHAR
⊘LAST_NAME : VARCHAR
⊘EMAIL : VARCHAR
⊘USER_ID : INT
⊘COURSE_ID : INT

Figure 2 Tables Queried for User's Course Section Information

11

Figure 3 Page to Select Course Section

3 Functionality Description and Navigation Design

3.1 Course Descriptions (for Administrator)

A course description includes a course number and a course name, such as COMP353

Database System. The administrator can insert new course descriptions, or view, modify

or delete existing course descriptions (Figure 4).
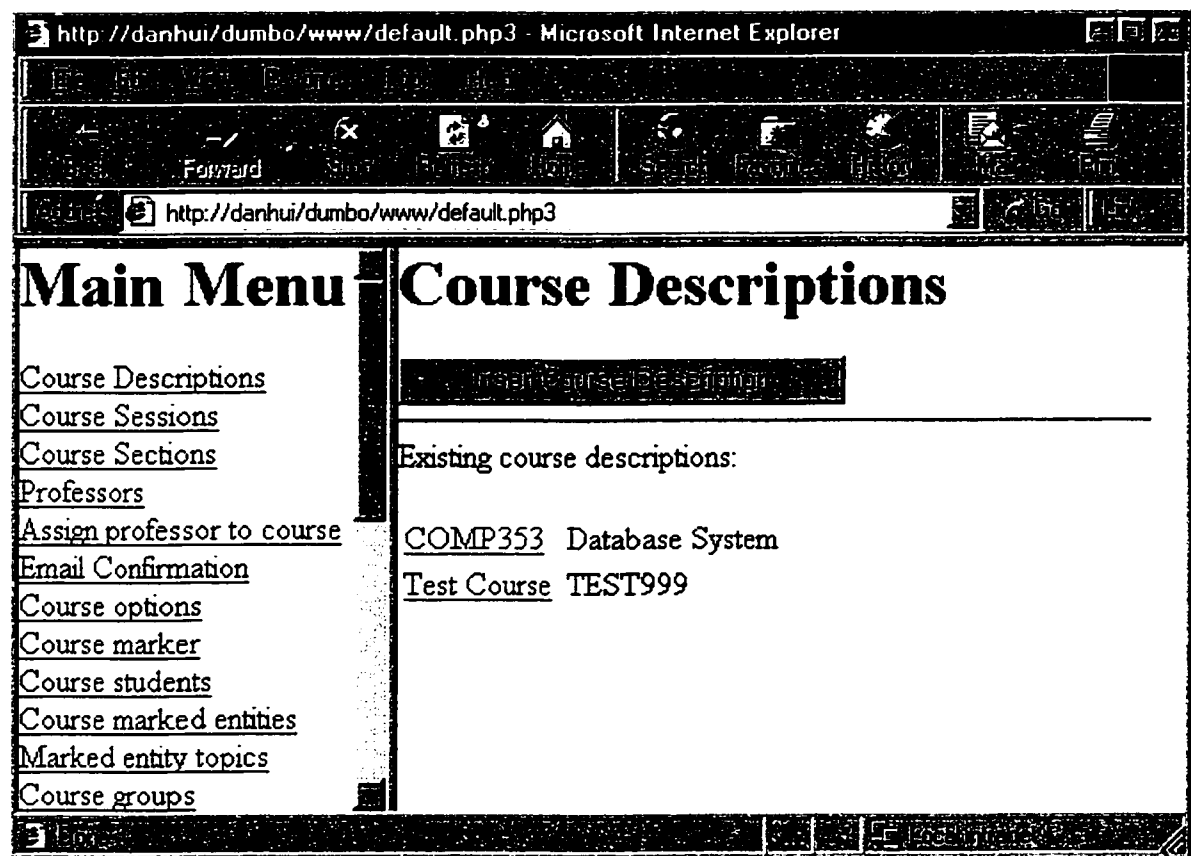
Figure 4 Entry Page for Course Descriptions

When a new course description is added or an existing course number is modified, the

application checks if the new course number already exists in the system and does not

allow the same course to be added twice. When modifying or deleting an existing course,

the application prevents the course description which has been referenced by a course

session to be deleted. In this case, the user has to remove the course session first and then

remove the course description.

3.1.1 Page: course_desc_list

Display (Figure 4)
Existing course descriptions

Input
Selected existing course description

Actions
Insert new course description → Page: course_desc (new course description)
Modify existing course descriptions → Page: course_desc (existing course description)

DB Tables
COURSE_DESC

3.1.2 Page: course_desc

Display (Figure 5)
- **New course description**
  None
- **Existing course description**
  Previous course number
  Previous course name

Input
- **New course description**
  Course number
  Course name
- **Existing course description**
  New course number
  New course name

Actions
- **New course description**
  Insert → Page: course_desc_list
  Cancel → Page: course_desc_list
- **Existing course description**

14

Update → Page: course_desc_list
Delete → Page: course_desc_list
Cancel → Page: course_desc_list

DB Tables
COURSE_DESC
COURSE_SESSION (for referenced course descriptions)

Figure 5 Page to Modify or Delete Course Description

3.2   Course Sessions (for Administrator)

The Course Manager system provides services for the administrator to insert new course sessions (fall/winter/summer and year) for a course, or to view, modify or delete existing course sessions. When a new course session is added or an existing course session is modified, the application checks if the new course session already exists for the course and does not allow the same course session to be added twice. When modifying or deleting an existing course session, the application prevents the course session which has been referenced by a course section to be deleted. In this case, the user has to remove the course section first and then remove the course session.

3.2.1   Page: course_session_list

Display
Existing course sessions

Input
Selected existing course session

Actions
Create new course session → Page: course_session (new course session)
Modify existing course sessions → Page: course_session (existing course session)

DB Tables
COURSE_DESC
COURSE_SESSION

3.2.2   Page: course_session

Display
* **New course session**
   Available course names
   Available sessions
   Available years
* **Existing course session**

Available course names
Available sessions
Available years
Previous course name
Previous session
Previous year

Input
- **New course session**
  Course name
  Session
  Year
- **Existing course session**
  New course name
  New session
  New year

Actions
- **New course session**
  Insert → Page: course_session_list
  Cancel → Page: course_session_list
- **Existing course session**
  Update → Page: course_session_list
  Delete → Page: course_session_list
  Cancel → Page: course_session_list

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE (for referenced course sections)

```
              <<RelationalTable>>
               COURSE_DESC
  %COURSE_DESC_ID : INT
  ◇COURSE_NUMBER : VARCHAR
  ◇COURSE_NAME : VARCHAR
```

```
COURSE_DESC_ID = COURSE_DESC_ID
```

```
              <<RelationalTable>>
              COURSE_SESSION
  %COURSE_SESSION_ID : INT
  ◇COURSE_DESC_ID : INT
  ◇SESSION : VARCHAR
  ◇YEAR : INT
```

```
COURSE_SESSION_ID = COURSE_SESSION_ID
```

```
              <<RelationalTable>>
                 COURSE
  %COURSE_ID : INT
  ◇SECTION : VARCHAR
  ◇COURSE_SESSION_ID : INT
  ◇SHOW_FINAL_GRADES : INT
```
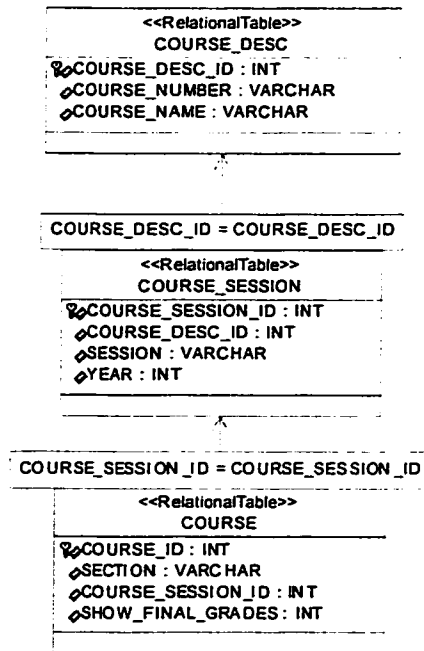
Figure 6 Tables for Course Sessions

18

3.3 Course Sections (for Administrator)

This functionality is for the administrator to insert new course sections for a course session, or to view, modify or delete existing course sections. When a new course section is added or an existing course section is modified, the application checks if the new course section already exists for the course session and does not allow the same course section to be added twice. When modifying or deleting an existing course section, the application prevents the course section with students registered to be deleted. In this case, the user has to remove the students first and then remove the course section.

3.3.1 Page: select_course_session

Display
Available course names
Available course sessions
Available course years

Input
Selected course name
Selected course session
Selected course year

Actions
Submit → Page: course_section_list

DB Tables
COURSE_DESC
COURSE_SESSION

3.3.2 Page: course_section_list

Display
Selected course name
Selected course session
Selected course year
Existing course sections

Input

Selected existing course section

Actions

Create new course section → Page: course_section (new course section)

Modify existing course sections → Page: course_section (existing course section)

DB Tables

COURSE_DESC
COURSE_SESSION
COURSE

3.3.3   Page: course_section

Display

- **New course section**
  None
- **Existing course section**
  Previous course section

Input

- **New course section**
  Course section
- **Existing course section**
  New course section

Actions

- **New course section**
  Insert → Page: course_section_list
  Cancel → Page: course_section_list
- **Existing course section**
  Update → Page: course_section_list
  Delete → Page: course_section_list
  Cancel → Page: course_section_list

DB Tables

COURSE
COURSE_PROF
COURSE_MARKED_ENTITY (for referenced courses)
MARK_ASSIGNEE (for referenced courses)

<<RelationalTable>>
COURSE_PROF

🔑COURSE_PROF_ID : INT
◇COURSE_ID : INT
◇PROFESSOR_ID : INT

COURSE_ID = COURSE_ID

<<RelationalTable>>
COURSE

🔑COURSE_ID : INT
◇SECTION : VARCHAR
◇COURSE_SESSION_ID : INT
◇SHOW_FINAL_GRADES : INT

COURSE_ID = COURSE_ID

<<RelationalTable>>
MARK_ASSIGNEE

🔑MARK_ASSIGNEE_ID : INT
◇COURSE_ID : INT

COURSE_ID = COURSE_ID

<<RelationalTable>>
COURSE_MARKED_ENTITY

🔑COURSE_MKD_ENT_ID : INT
◇MAX_MARK : FLOAT
◇NAME : VARCHAR
◇WEIGHT : FLOAT
◇SHARED_TOPICS : CHAR
◇COURSE_ID : INT
◇DEADLINE : DATE

Figure 7 Tables for Course Sections

3.4 Professors (for Administrator)

The administrator can use this functionality to insert new professors into the system with the professors' first and last names and email addresses, or to view, modify or delete existing professors. When a new professor is added to the system, a professor user account is created with the professor's last name as the basis for the user name. If a same user name already exists in the system, it will be padded with 1's until it is unique. Once the user name is created, it cannot be changed. The user name and password will be emailed to the professor at the specified email address. The user may change his password if necessary. When a new professor is added or an existing professor is modified, the application checks if a professor with the same name already exists in the system and does not allow the same professor to be added twice. When modifying or deleting an existing professor, the application warns the user if the professor is assigned to a course in the system.

3.4.1 Page: professor_list

Display
Existing professors

Input
Selected existing professor

Actions
Insert new professor → Page: professor (new professor)
Modify existing professors → Page: professor (existing professor)

DB Tables
PROFESSOR

3.4.2   Page: professor

Display
- **New professor**
  None
- **Existing professor**
  Professor user name
  Previous professor first name
  Previous professor last name
  Previous professor email

Input
- **New professor**
  Professor first name
  Professor last name
  Professor email
- **Existing professor**
  New professor first name
  New professor last name
  New professor email

Actions
- **New professor**
  Insert → Page: professor_list
  Cancel → Page: professor_list
- **Existing professor**
  Update → Page: professor_list
  Delete → Page: professor_list
  Cancel → Page: professor_list

DB Tables
COURSE_PROF
PROFESSOR
DB_USER
EMAIL_CONF

```
          <<RelaionalTable>>
           COURSE_PROF
%oCOURSE_PROF_ID : INT
 øCOURSE_ID : INT
 øPROFESSOR_ID : INT


PROFESSOR_ID = PROFESSOR_ID
```

```
          <<RelaionalTable>>                    <<RelaionalTable>>
            PROFESSOR                              EMAIL_CONF
%oPROFESSOR_ID : INT              %oEMAIL_ID : INT
 øFIRST_NAME : VARCHAR            øMESSAGE_TYPE : VARCHAR
 øLAST_NAME : VARCHAR            øSUBJECT : VARCHAR
 øEMAIL : VARCHAR                 øMESSAGE : VARCHAR
 øUSER_ID : INT                   øREPLY_TO : VARCHAR
                                  øACTIVATION : INT

    USER_ID = USER_ID
```

```
          <<RelaionalTable>>
             DB_USER
%oUSER_ID : INT
 øUSER_NAME : VARCHAR
 øPASSWORD : VARCHAR
 øACCESS_PRIV : CHAR
```

Figure 8 Tables for Professors

3.5 Assign Professor to Course Sections (for Administrator)

This functionality allows the administrator to assign a professor to a course section (Figure 9). A professor can be assigned to many course sections but a course section can have only one professor assigned to it.

Figure 9 Page to Assign Professor to Course Section

3.5.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name

Selected course session
Selected course year
Selected course section

Actions
Submit → Page: select_course_prof

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.5.2   Page: select_course_prof

Display (Figure 9)
Selected course name
Selected course session
Selected course year
Selected course section
Available professors
Previous professor (for changing previously assigned professor)

Input
Professor (for assigning professor to course)
New professor (for changing previously assigned professor)

Actions
Submit → Page: home
Back → Page: select_course

DB Tables
PROFESSOR
COURSE_PROF

3.6 Email Confirmation (for Administrator)

When a user account is created for a professor, a marker, a student or a project group, an email confirmation is sent to the user at the specified address. The email contains the user name and password and a short message. This functionality is for the administrator to create new confirmation messages, or to view, edit or delete existing messages. Currently there are four kinds of email confirmation messages in the system, for the professor, the marker, the student and the project group respectively.

3.6.1 Page: email_conf_list

Display
Existing email messages

Input
Selected email message

Actions
Create new email message → Page: email_conf (new email message)
Modify existing email messages → Page: email_conf (existing email message)

DB Tables
EMAIL_CONF

3.6.2 Page: email_conf

Display
- **New email message**
  None
- **Existing email message**
  Previous message type
  Previous subject
  Previous message
  Previous reply address

Input

- **New email message**
  Message type
  Subject
  Message
  Reply address
- **Existing email message**
  New message type
  New subject
  New message
  New reply address

Actions
- **New email message**
  Insert → Page: email_conf_list
  Cancel → Page: email_conf_list
- **Existing email message**
  Update → Page: email_conf_list
  Delete → Page: email_conf_iist
  Cancel → Page: email_conf_list

DB Tables
EMAIL_CONF

3.7 Course Options (for Administrator and Professor)

The functionality is for the administrator or professors to specify whether to show the final numerical grades or not when the course grades are displayed (Figure 10).

Figure 10 Page to Set Course Option

3.7.1 Page: select_course

Display
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year

Selected course section

Actions
Submit → Page: set_course_options

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.7.2   Page: set_course_options

Display (Figure 10)
Selected course name
Selected course session
Selected course year
Selected course section
Available course options
Previous course option (for changing previously set course option)

Input
Course option (for setting option for course)
New course option (for changing previously set course option)

Actions
Submit → Page: home
Back → Page: select_course

DB Tables
COURSE

3.8 Course Marker (for Administrator and Professor)

The administrator and the professor can use this functionality to add new markers for a course section with the markers' first and last names and email addresses, or to view, modify or delete existing markers. When a new marker is added to the system, a marker user account is created with the marker's last name as the basis for the user name. If an identical user name already exists in the system, it will be padded with 1's until it is unique. Once the user name is created, it cannot be changed. The user name and password will be emailed to the marker at the specified email address. The user may change his password if necessary. When a new marker is added or an existing marker is modified, the application checks if a marker with the same name has already been assigned to the course section and does not allow the same marker to be added twice for the same section. When a marker is deleted, the information about his assignment to the course section is deleted as well. A course section may have several markers assigned to it.

3.8.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions

Submit → Page: course_marker_list

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.8.2   Page: course_marker_list

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing markers

Input
Selected existing marker

Actions
Insert new marker → Page: course_marker (new marker)
Modify existing markers → Page: course_marker (existing marker)

DB Tables
COURSE_MARKER

3.8.3   Page: course_marker

Display
- **New marker**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
- **Existing marker**
  Selected course name
  Selected course session

Selected course year
Selected course section
Marker user name
Previous marker first name
Previous marker last name
Previous marker email

Input
- **New marker**
  Marker first name
  Marker last name
  Marker email
- **Existing marker**
  New marker first name
  New marker last name
  New marker email

Actions
- **New marker**
  Insert → Page: course_marker_list
  Cancel → Page: course_marker_list
- **Existing marker**
  Update → Page: course_marker_list
  Delete → Page: course_marker_list
  Cancel → Page: course_marker_list

DB Tables
COURSE_MARKER
DB_USER
EMAIL_CONF

```
    <<RelationalTable>>
     COURSE_MARKER
┌─────────────────────────────┐
│%COURSE_MARKER_ID : INT      │
│◇FIRST_NAME : VARCHAR        │
│◇LAST_NAME : VARCHAR         │
│◇EMAIL : VARCHAR             │
│◇USER_ID : INT               │
│◇COURSE_ID : INT             │
└─────────────────────────────┘
```

```
                                    <<RelationalTable>>
                                       EMAIL_CONF
                              ┌─────────────────────────────┐
                              │%EMAIL_ID : INT              │
                              │◇MESSAGE_TYPE : VARCHAR      │
                              │◇SUBJECT : VARCHAR           │
USER_ID = USER_ID             │◇MESSAGE : VARCHAR           │
                              │◇REPLY_TO : VARCHAR          │
                              │◇ACTIVATION : INT            │
                              └─────────────────────────────┘
```

```
    <<RelationalTable>>
        DB_USER
┌─────────────────────────────┐
│%USER_ID : INT               │
│◇USER_NAME : VARCHAR         │
│◇PASSWORD : VARCHAR          │
│◇ACCESS_PRIV : CHAR          │
└─────────────────────────────┘
```
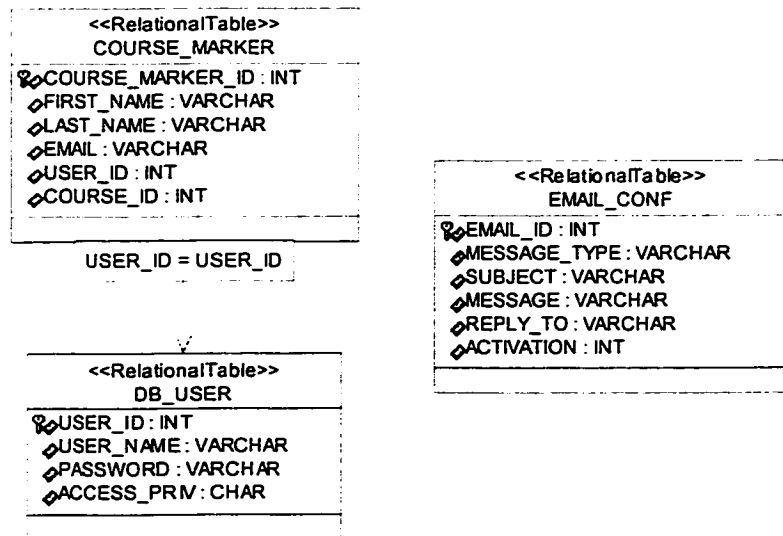
Figure 11 Tables for Markers

34

3.9　Course Students (for Administrator and Professor)

The administrator and the professor can use this functionality to add new students for a course section with the students' IDs, first and last names and email addresses, or to view, modify or delete existing students (Figure 12).

Figure 12 Page for Course Students

The students can be added to a course section in two ways: by entering from the form and by uploading from a comma-separated-value (CSV) file (Figure 13). When a new student is added to a course section, a student user account is created with the student's ID as the basis for the user name. If a same user name already exists in the system, it will be padded with 1's until it is unique. Once the user name is created, it cannot be

changed. The user name and password will be emailed to the student at the specified email address. The user may change his password if necessary. When a new student is added or an existing student is modified, the application checks if a student with the same ID has already been added to the course section and does not allow the same student to be added twice for the same section. When a student is deleted from a course section, his group membership and marks in the section are also deleted.
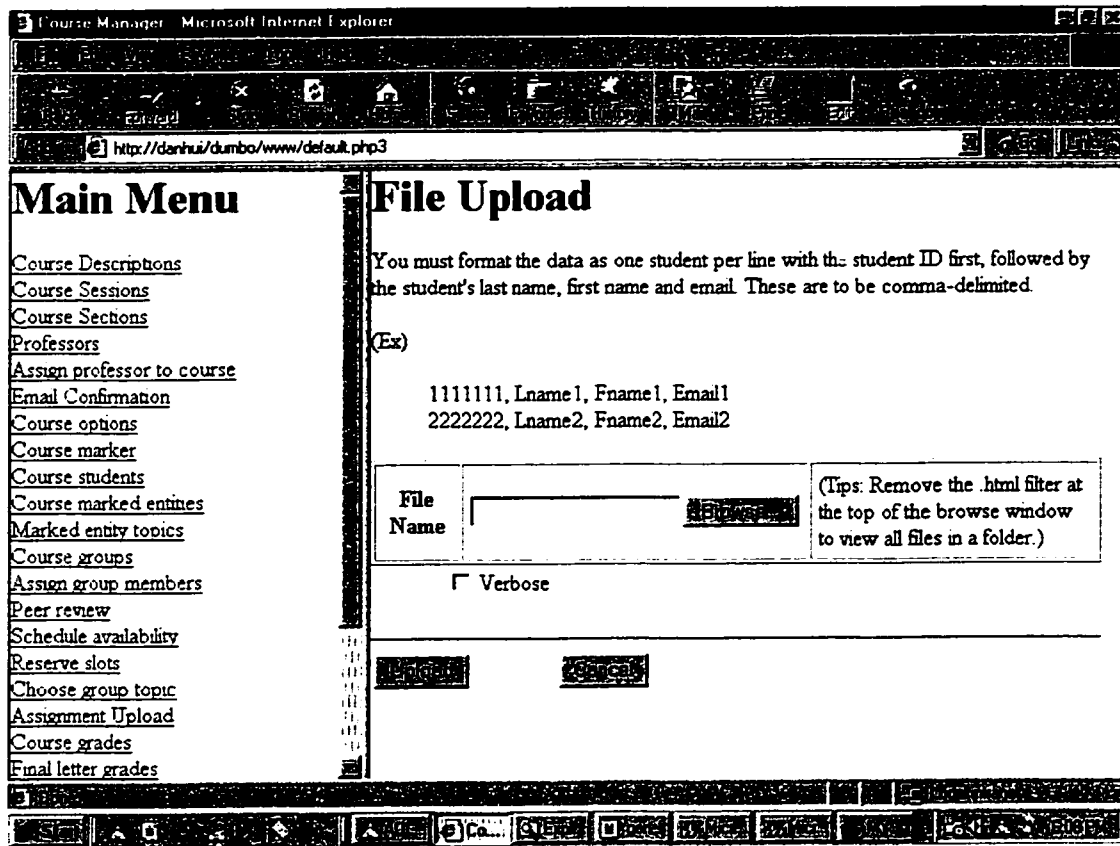
Figure 13 Page to Add Students from a File

3.9.1   Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years

Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: course_student_list

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.9.2 Page: course_student_list

Display (Figure 12)
Selected course name
Selected course session
Selected course year
Selected course section
Existing students

Input
Selected existing student

Actions
Insert new student → Page: course_student (new student)
Modify existing students → Page: course_student (existing student)
File upload (batch student insert) → Page: file_selection_student

DB Tables
COURSE_STUDENT
MARK_ASSIGNEE

3.9.3 Page: course_student

Display
- **New student**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
- **Existing student**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
  Previous student ID
  Previous student first name
  Previous student last name
  Previous student email

Input
- **New student**
  Student ID
  Student first name
  Student last name
  Student email
- **Existing student**
  New student ID
  New student first name
  New student last name
  New student email

Actions
- **New student**
  Insert → Page: course_student_list
  Cancel → Page: course_student_list
- **Existing student**
  Update → Page: course_student_list
  Delete → Page: course_student_list
  Cancel → Page: course_student_list

DB Tables
COURSE_DESC
EMAIL_CONF
DB_USER
MKD_ENT_GRADE
GROUP_MEMBER

COURSE_STUDENT
MARK_ASSIGNEE

```
        <<RelationalTable>>
          COURSE_DESC
    ⚷COURSE_DESC_ID : INT
    ⚷COURSE_NUMBER : VARCHAR
    ⚷COURSE_NAME : VARCHAR


        <<RelationalTable>>
          EMAIL_CONF
    ⚷EMAIL_ID : INT
    ⚷MESSAGE_TYPE : VARCHAR
    ⚷SUBJECT : VARCHAR
    ⚷MESSAGE : VARCHAR
    ⚷REPLY_TO : VARCHAR
    ⚷ACTIVATION : INT


        <<RelationalTable>>
           DB_USER
    ⚷USER_ID : INT
    ⚷USER_NAME : VARCHAR      ⟵ USER_ID = USER_ID
    ⚷PASSWORD : VARCHAR
    ⚷ACCESS_PRIV : CHAR
```

```
        <<RelationalTable>>
          MKD_ENT_GRADE
    ⚷MKD_ENT_GRADE_ID : INT
    ⚷MARK : FLOAT
    ⚷COURSE_MKD_ENT_ID : INT
    ⚷MARK_ASSIGNEE_ID : INT
    ⚷NOTES : TEXT
```

MARK_ASSIGNEE_ID = MARK_ASSIGNEE_ID

```
        <<RelationalTable>>
          MARK_ASSIGNEE
    ⚷MARK_ASSIGNEE_ID : INT
    ⚷COURSE_ID : INT
```

COURSE_STUDENT_ID = MARK_ASSIGNEE_ID

```
        <<RelationalTable>>
          COURSE_STUDENT
    ⚷COURSE_STUDENT_ID : INT
    ⚷STUDENT_ID : INT
    ⚷FIRST_NAME : VARCHAR
    ⚷LAST_NAME : VARCHAR
    ⚷EMAIL : VARCHAR
    ⚷USER_ID : INT
```

COURSE_STUDENT_ID = COURSE_STUDENT_ID

```
        <<RelationalTable>>
          GROUP_MEMBER
    ⚷COURSE_GROUP_ID : INT
    ⚷COURSE_STUDENT_ID : INT
```

Figure 14 Tables for Course Students

3.9.4 Page: file_selection_student

Display (Figure 13)
Upload status (with verbose selection)

Input
File path and name
Verbose selection

39

Actions
Browse → Dialog Box: choose_file
Upload (with verbose selection) → Page: file_selection_student
Upload (without verbose selection) → Page: course_student_list
Cancel → Page: course_student_list

DB Tables
COURSE_DESC
EMAIL_CONF
DB_USER
COURSE_STUDENT
MARK_ASSIGNEE

3.10 Course Marked Entities (for Administrator and Professor)

A course marked entity refers to an entity that can be given a grade, such as an assignment, a lab demo, a project, an exam. A marked entity has a name, a weight, a maximum mark and an optional deadline. A marked entity can have a number of associated topics, in case it is a report or essay on a topic. When a marked entity is defined, the definer should indicate whether the marked entity allows students to share topics or not. The administrator and the professor can add new marked entities for a course section, or to view, modify or delete existing marked entities (Figure 15). When a new marked entity is added or an existing marked entity is modified, the application checks if a marked entity with the same name has already been added to the course section and does not allow the same marked entity to be added twice for the same section. A marked entity cannot be deleted if the entity has been referenced, for example, a student has been given a mark for that entity or topics have been assigned to it.

3.10.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: course_mkd_ent_list

DB Tables
```
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)
```

3.10.2 Page: course_mkd_ent_list

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing marked entities

Input
Selected existing marked entity

Actions
Insert new marked entity → Page: course_mkd_ent (new marked entity)
Modify existing marked entities → Page: course_mkd_ent (existing marked entity)

DB Tables
```
COURSE_MARKED_ENTITY
COURSE
```

3.10.3 Page: course_mkd_ent

Display (Figure 15)
- **New marked entity**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
- **Existing marked entity**
  Selected course name
  Selected course session

Selected course year
Selected course section
Previous marked entity name
Previous marked entity weight
Previous marked entity maximum mark
Previous marked entity share topics selection
Previous marked entity deadline

Figure 15 Page to Modfiy or Delete Marked Entity

Input
- **New marked entity**
  Marked entity name
  Marked entity weight
  Marked entity maximum mark
  Marked entity share topics selection
  Marked entity deadline
  Assign topics selection
- **Existing marked entity**

New marked entity name
New marked entity weight
New marked entity maximum mark
New marked entity share topics selection
New marked entity deadline
Assign topics selection

Actions

- **New marked entity**
  Insert (without assign topics selection) → Page: course_mkd_ent_list
  Insert (with assign topics selection) → Page: mkd_ent_topic_list
  Cancel → Page: course_mkd_ent_list

- **Existing marked entity**
  Update (without assign topics selection) → Page: course_mkd_ent_list
  Update (with assign topics selection) → Page: mkd_ent_topic_list
  Delete → Page: course_mkd_ent_list
  Cancel → Page: course_mkd_ent_list

DB Tables

COURSE_MARKED_ENTITY
COURSE
PROJECT_TOPIC (for referenced marked entities)
MKD_ENT_GRADE (for referenced marked entities)
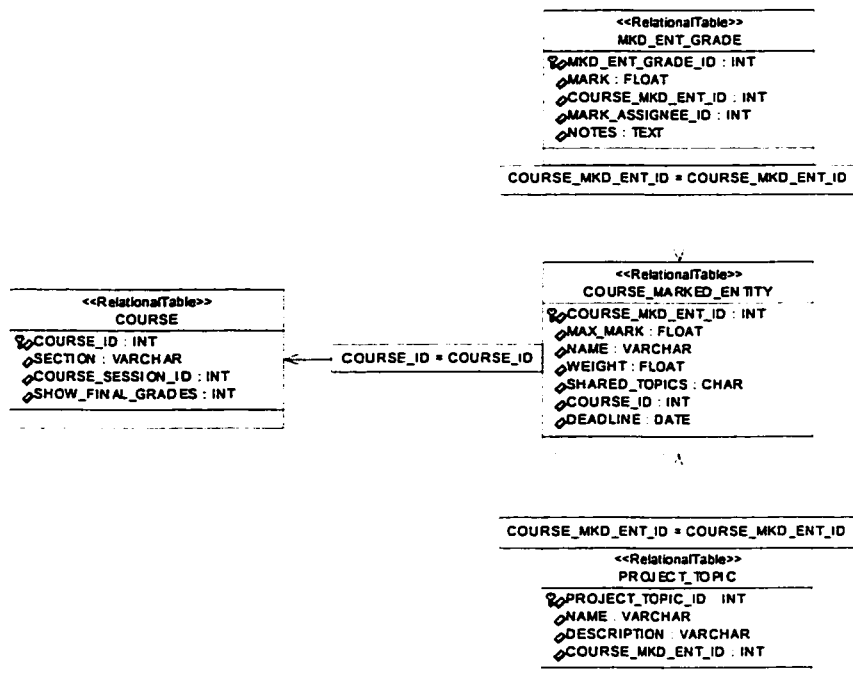
Figure 16 Tables for Marked Entities

3.10.4 Page: mkd_ent_topic_list

See **Marked Entity Topics**.

3.11 Marked Entity Topics (for Administrator and Professor)

The administrator and the professor can add new topics for a marked entity, or to view, modify or delete existing topics. When a new topic is added or an existing topic is modified, the application checks if a topic with the same name has already been added to the marked entity and does not allow the same topic to be added twice for the same marked entity. A topic cannot be deleted if it has already been chosen by project groups.

3.11.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: course_mkd_ent_list

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.11.2 Page: course_mkd_ent_list

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing marked entities

Input
Selected existing marked entity

Actions
Modify existing marked entities → Page: mkd_ent_topic_list

DB Tables
COURSE_MARKED_ENTITY
COURSE

3.11.3 Page: mkd_ent_topic_list

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing topics

Input
Selected topic

Actions
Insert new topic → Page: mkd_ent_topic (new topic)
Modify existing topics → Page: mkd_ent_topic (existing topic)

DB Tables
COURSE_MARKED_ENTITY
PROJECT_TOPIC

3.11.4 Page: mkd_ent_topic

Display
• **New topic**

Selected course name
Selected course session
Selected course year
Selected course section

- **Existing topic**
Selected course name
Selected course session
Selected course year
Selected course section
Previous topic name
Previous topic description

Input
- **New topic**
Topic name
Topic description
- **Existing topic**
New topic name
New topic description

Actions
- **New topic**
Insert → Page: mkd_ent_topic_list
Cancel → Page: mkd_ent_topic_list
- **Existing topic**
Update → Page: mkd_ent_topic_list
Delete → Page: mkd_ent_topic_list
Cancel → Page: mkd_ent_topic_list

DB Tables
PROJECT_TOPIC
PROJECT_ASSIGNED_GROUP (for referenced topics)

```
           <<RelationalTable>>
            PROJECT_TOPIC
  ⚷PROJECT_TOPIC_ID : INT
   ◇NAME : VARCHAR
   ◇DESCRIPTION : VARCHAR
   ◇COURSE_MKD_ENT_ID : INT
```

PROJECT_TOPIC_ID = PROJECT_TOPIC_ID

```
           <<RelationalTable>>
        PROJECT_ASSIGNED_GROUP
  ⚷PROJ_ASSIGNED_GROUP_ID : INT
   ◇COURSE_GROUP_ID : INT
   ◇PROJECT_TOPIC_ID : INT
```

Figure 17 Tables for Topics

3.12 Course Groups (for Administrator and Professor)

The administrator and the professor can use this functionality to add new groups for a course section with the groups' names and email addresses, or to view, modify or delete existing groups. The groups can be added to a course section in two ways: by entering from the form and by uploading from a comma-separated-value (CSV) file. Group members can be added to a group at the same time when the group is created or later. When a new group is added to a course section, a group user account is created with the group name as the basis for the user name. If a same user name already exists in the system, it will be padded with 1's until it is unique. Once the user name is created, it cannot be changed. The user name and password will be emailed to the group at the specified email address. The user may change his password if necessary. When a new group is added or an existing group is modified, the application checks if a group with the same name has already been added to the course section and does not allow the same group to be added twice for the same section. A group cannot be deleted from a course section if it is referenced.

3.12.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: course_group_list

DB Tables
```
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)
```

3.12.2 Page: course_group_list

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing groups

Input
Selected existing group

Actions
Insert new group → Page: course_group (new group)
Modify existing groups → Page: course_group (existing group)
File upload (batch group insert) → Page: file_selection

DB Tables
```
COURSE_GROUP
MARK_ASSIGNEE
COURSE_STUDENT
GROUP_MEMBER
```

3.12.3 Page: course_group

Display
● **New group**
   Selected course name

Selected course session
Selected course year
Selected course section
- **Existing group**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
  User name for selected group
  Previous group name
  Previous group email

Input
- **New group**
  Group name
  Group email
  Assign group members selection
- **Existing group**
  New group name
  New group email
  Assign group members selection

Actions
- **New group**
  Insert (without assign group members selection) → Page: course_group_list
  Insert (with assign group members selection) → Page: group_members
  Cancel → Page: course_group_list
- **Existing group**
  Update (without assign group members selection) → Page: course_group_list
  Update (with assign group members selection) → Page: group_members
  Delete → Page: course_group_list
  Cancel → Page: course_group_list

DB Tables
COURSE_GROUP
DB_USER
GROUP_MEMBER
EMAIL_CONF
MARK_ASSIGNEE
PROJECT_ASSIGNED_GROUP (for referenced groups)

3.12.4 Page: file_selection

Display

Upload status (with verbose selection)

Input
File path and name
Verbose selection
Course groups with students selection

Actions
Browse → Dialog Box: choose_file
Upload (with verbose selection) → Page: file_selection
Upload (without verbose selection) → Page: course_group_list
Cancel → Page: course_group_list

DB Tables
COURSE_GROUP
DB_USER
GROUP_MEMBER
COURSE_STUDENT
MARK_ASSIGNEE
EMAIL_CONF
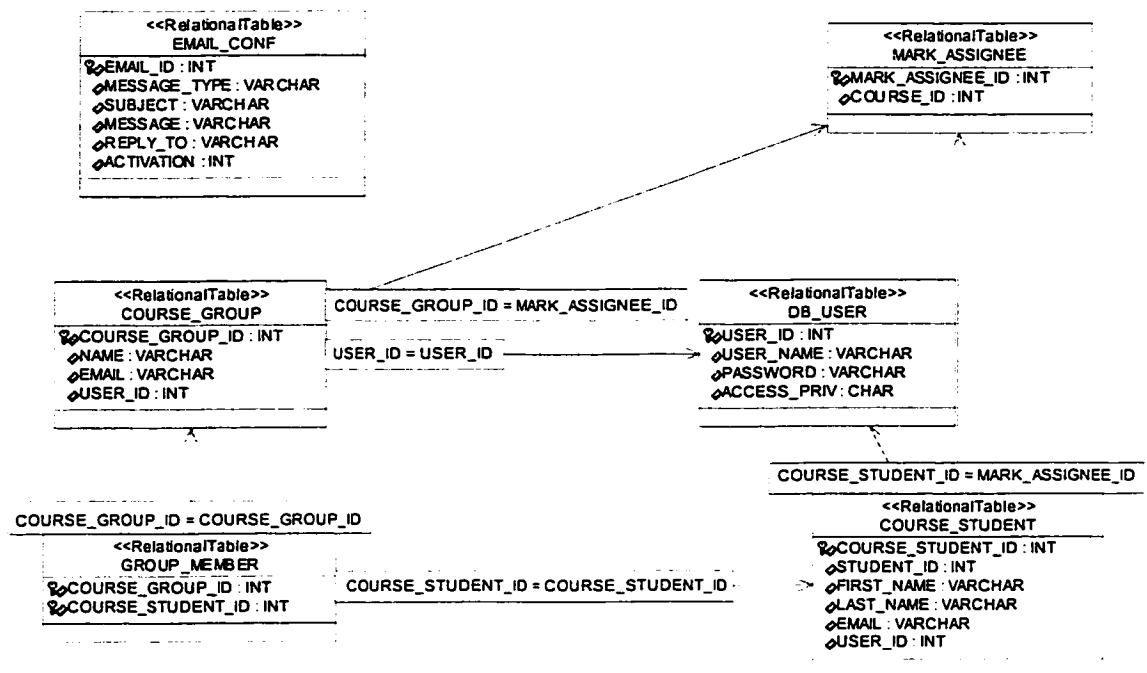
Figure 18 Tables for Course Groups

3.12.5 Page: group_members

See **Assign Group Members**.

3.13 Assign Group Members (for Administrator and Professor)

The administrator and the professor can add or remove members from a group (Figure 19). To add a student to a group, the student should be in the course section for the group. When a mark is assigned to a group for a marked entity, the mark is given to all the members in the group, although it is possible that members of the same group have different marks for the same marked entity. If a student is assigned to a group after the group received a mark, the new member will not have the mark until the mark is assigned specifically to him.
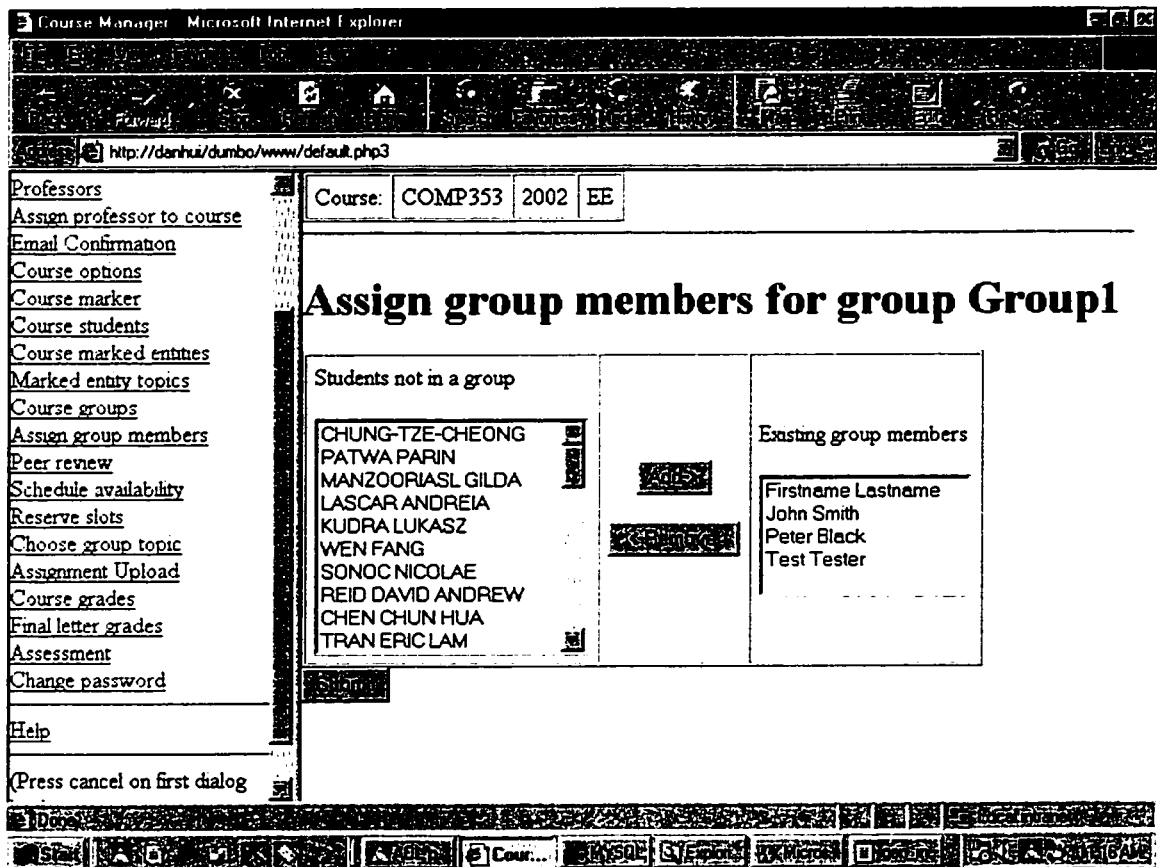
Figure 19 Page to Assign Group Members

3.13.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: course_group_list

DB Tables
```
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)
```

3.13.2 Page: course_group_list

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing groups

Input
Selected existing group

Actions
Modify existing groups → Page: group_members

DB Tables

COURSE_GROUP
MARK_ASSIGNEE
COURSE_STUDENT
GROUP_MEMBER

3.13.3 Page: group_members

Display (Figure 19)
Selected course name
Selected course session
Selected course year
Selected course section
Students not in a group
Existing group members

Input
Add or remove group members

Actions
Summit → Page: home

DB Tables
COURSE_GROUP
GROUP_MEMBER
COURSE_STUDENT
MARK_ASSIGNEE

3.14 Peer Review (for Administrator and Professor)

The administrator and the professor can use this functionality to view the peer review scores and comments given by students to their group members (Figure 20). The administrator and the professor can also decide whether a score should be discarded or not. When a score is discarded, it will not be used any more. However, it is still stored in the database.

Peer review for group Group1

Click on the score to see the reviewer and comments and to decide whether to discard it or not

Student ID	Student Name	Peer Review Scores	Average Score
1111111	John Smith	83 84 85	85
2222222	Peter Black	80 82 84	84
3333333	Test Tester	81 82 83	82.5
4444444	Firstname Lastname	81 83 84	83.5

Figure 20 Page to Display Peer Review Results

3.14.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions

58

Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: course_group_list

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.14.2 Page: course_group_list

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing groups

Input
Selected existing group

Actions
View peer review results for selected group → Page: peer_review

DB Tables
COURSE_GROUP
MARK_ASSIGNEE
COURSE_STUDENT
GROUP_MEMBER

3.14.3 Page: peer_review

Display (Figure 20)
Selected course name
Selected course session
Selected course year
Selected course section
Peer review scores for students in selected group
Peer review comments for students in selected group

Input
Peer review score selection

Actions
OK → Page: peer_review
Cancel → Page: peer_review

DB Tables
PEER_REVIEW
COURSE_GROUP
GROUP_MEMBER
COURSE_STUDENT
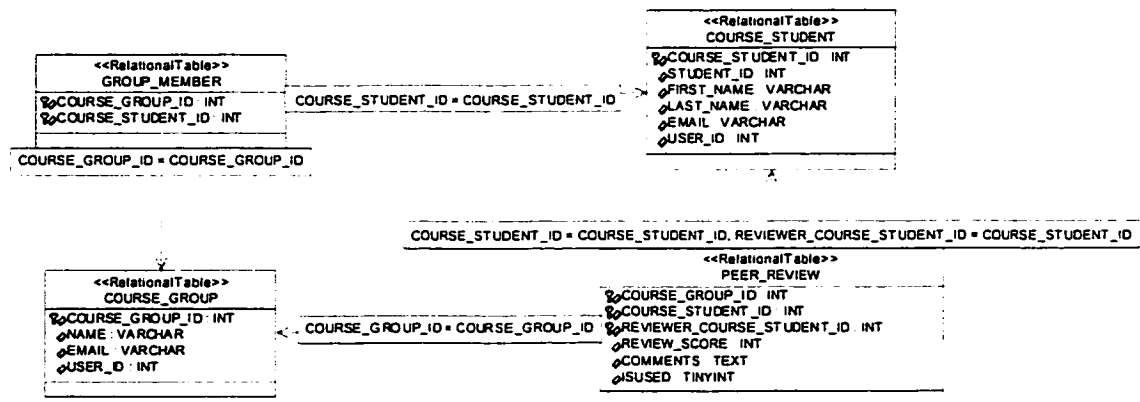
Figure 21 Tables for Peer Review

3.15 Peer Review (for Student)

A student can view the peer review scores from his group member but cannot see who

assigned them. He can also assign peer review scores and add comments for his group

members.

Figure 22 Page to Conduct Peer Review

3.15.1 Page: course_group_list

Display
Student course name
Student course session
Student course year
Student course section
Student groups

Input
Selected group

Actions
Peer review → Page: peer_review_student

DB Tables
COURSE_GROUP
MARK_ASSIGNEE
COURSE_STUDENT
GROUP_MEMBER

3.15.2 Page: peer_review_student

Display (Figure 22)
Student course name
Student course session
Student course year
Student course section
Peer review scores for the student
Peer review scores for other members in the group

Input
Peer review scores for other members in the group
Peer review comments for other members in the group

Actions
Assign peer review score with comments → Page: peer_review_student

DB Tables
PEER_REVIEW
COURSE_GROUP
GROUP_MEMBER
COURSE_STUDENT

3.16 Schedule Availability (for Administrator and Professor)

The administrator and the profess can use this functionality to view, add, modify or delete

time slots for demos (Figure 23). Time slots cannot conflict with each other.

Figure 23 Page to Schedule Availability

3.16.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year

Selected course section

Actions
Submit → Page: month_time_slot_list

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.16.2 Page: month_time_slot_list

Display (Figure 23)
Selected course name
Selected course session
Selected course year
Selected course section
Calendar
Existing time slots

Input
Selected existing time slot

Actions
Navigate through months → Page: month_time_slot_list
Select a day → Page: day_time_slot_list
Modify existing time slots → Page: time_slot

DB Tables
MEETING_TIME_SLOT

3.16.3 Page: day_time_slot_list

Display
Selected course name
Selected course session
Selected course year

Selected course section
Existing time slots

Input
Selected time slot

Actions
Insert new time slot → Page: time_slot (new time slot)
Modify existing time slots → Page: time_slot (existing time slot)

DB Tables
MEETING_TIME_SLOT

3.16.4 Page: time_slot

Display
- **New time slot**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
- **Existing time slot**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
  Previous start time
  Previous end time
  Previous purpose

Input
- **New time slot**
  Start time
  End time
  Purpose
- **Existing time slot**
  New start time
  New end time
  New purpose

Actions
- **New time slot**
  Insert → Page: day_time_slot_list
  Cancel → Page: day_time_slot_list

- **Existing time slot**
  Update → Page: day_time_slot_list
  Delete → Page: day_time_slot_list
  Cancel → Page: day_time_slot_list

DB Tables
SCHED_MEETING
MEETING_TIME_SLOT

3.17 Reserve Slots (for Administrator, Marker and Project Group)

The administrator can reserve or cancel any time slot (Figure 24). The marker can view

all the reserved time slots for his course section. The project group can reserve or cancel

a time slot for his group.

Figure 24 Page to Reserve Time Slot

3.17.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years

Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: group_schedules

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.17.2 Page: group_schedules

Display
Selected course name
Selected course session
Selected course year
Selected course section
Calendar
Existing time slots
Reserved time slots

Input
Selected time slot
Selected purpose

Actions
Navigate through months → Page: group_schedules
Change purpose → Page: group_schedules
Reserve existing time slots → Page: reserve_slot
Cancel reserved time slots → Page: reserve_slot

DB Tables

```
MARK_ASSIGNEE
COURSE_GROUP
SCHED_MEETING
MEETING_TIME_SLOT
```

```
            <<RelationalTable>>
            MEETING_TIME_SLOT
    ─────────────────────────────────
    %TIME_SLOT_ID : INT
    ◇MEETING_DATE : DATE
    ◇START_TIME : TIME
    ◇END_TIME : TIME
    ◇COURSE_ID : INT
    ◇PURPOSE : CHAR
```

TIME_SLOT_ID = TIME_SLOT_ID

```
            <<RelationalTable>>
            SCHED_MEETING
    ─────────────────────────────────
    %TIME_SLOT_ID : INT
    ◇COURSE_GROUP_ID : INT
```

COURSE_GROUP_ID = COURSE_GROUP_ID

```
            <<RelationalTable>>
            COURSE_GROUP
    ─────────────────────────────────
    %COURSE_GROUP_ID : INT
    ◇NAME : VARCHAR
    ◇EMAIL : VARCHAR
    ◇USER_ID : INT
```

COURSE_GROUP_ID = MARK_ASSIGNEE_ID

```
            <<RelationalTable>>
            MARK_ASSIGNEE
    ─────────────────────────────────
    %MARK_ASSIGNEE_ID : INT
    ◇COURSE_ID : INT
```
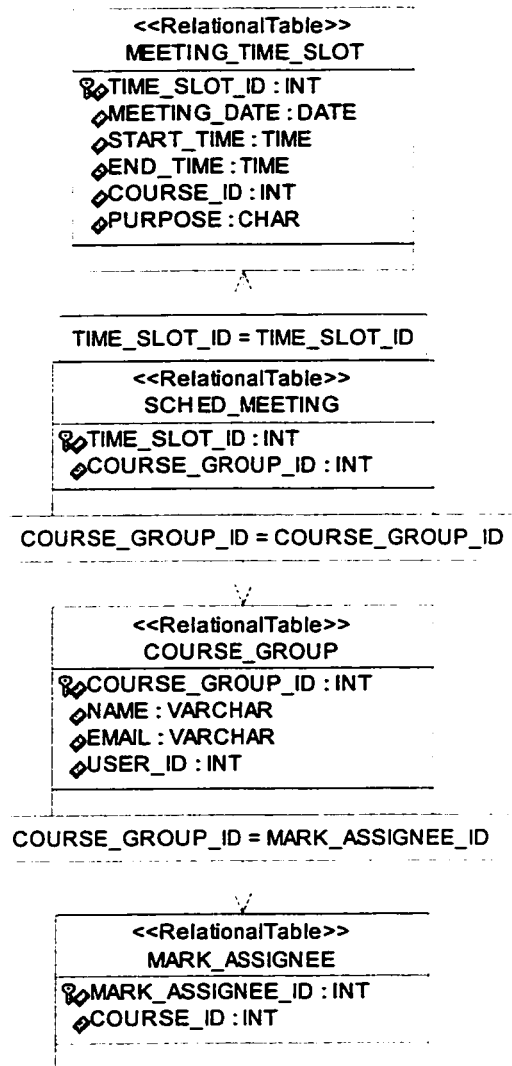
Figure 25 Tables for Time Slots

3.17.3 Page: reserve_slot

Display (Figure 24)
Selected course name
Selected course session
Selected course year

Selected course section
Start time
End time
Purpose

Input
Selected group

Actions
Reserve time slot → Page: group_schedules
Cancel reserve time slot → Page: group_schedules
Back → Page: group_schedules

DB Tables
MEETING_TIME_SLOT
SCHED_MEETING
COURSE_GROUP
MARK_ASSIGNEE

3.18 Choose Group Topic (for Administrator and Project Group)

The administrator can use this functionality to assign or change topics of marked entities

for any project groups (Figure 26). The project group can use this functionality to select

or de-select topics for themselves. If a marked entity allows its topics to be shared,

several groups can select the same topic. Otherwise, a group can only choose from the

topics which have not been assigned to other groups.

Figure 26 Page to Assign Topics

3.18.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: topic_assignments

DB Tables
```
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)
```

3.18.2 Page: topic_assignments

Display (Figure 26)
Selected course name
Selected course session
Selected course year
Selected course section
Existing topic assignments

Input
Selected topic assignment

Actions
Insert new topic assignment → Page: select_project_topic (new topic assignment)
Modify existing topic assignments → Page: select_project_topic (existing topic assignment)

DB Tables
MARK_ASSIGNEE
COURSE_GROUP
COURSE_MARKED_ENTITY
PROJECT_ASSIGNED_GROUP
PROJECT_TOPIC
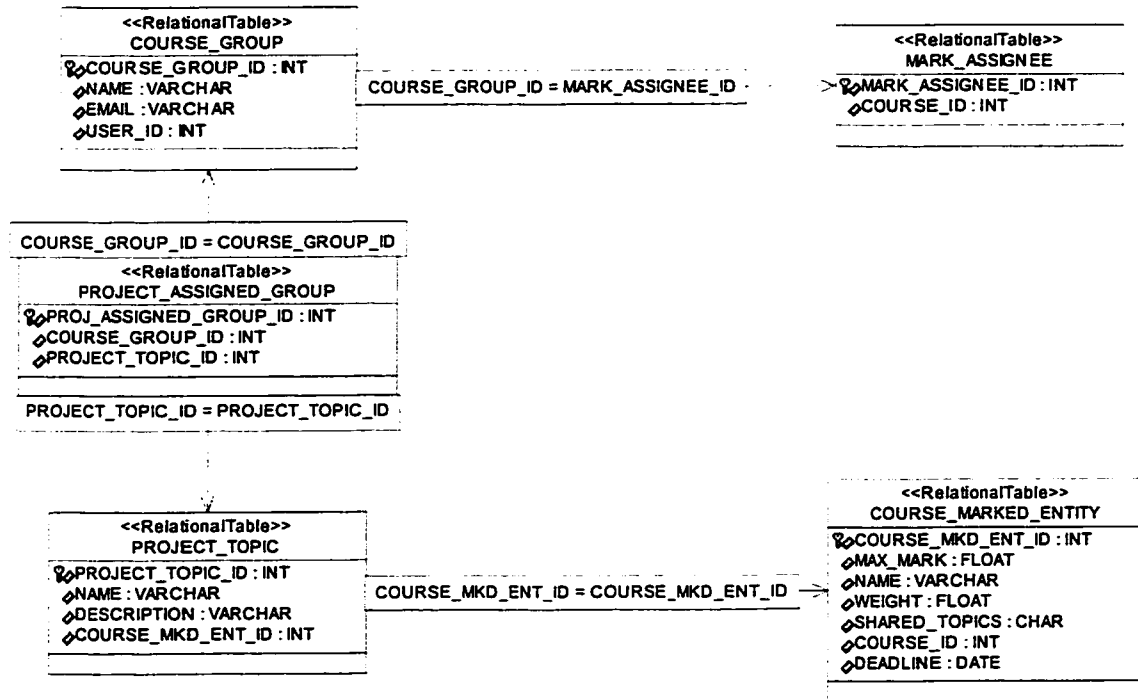
Figure 27 Tables for Group Topics

3.18.3 Page: select_project_topic

Display
- **New topic assignment**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
  Available topics
- **Existing topic assignment**
  Selected course name
  Selected course session
  Selected course year
  Selected course section

73

Available topics
Previous topic assignment

Input
- **New topic assignment**
  Topic assignment
- **Existing topic assignment**
  New topic assignment

Actions
- **New topic assignment**
  Choose → Page: topic_assignments
  Back → Page: topic_assignments
- **Existing topic assignment**
  Update topic → Page: topic_assignments
  De-select topic → Page: topic_assignments
  Back → Page: topic_assignments

DB Tables
PROJECT_TOPIC
PROJECT_ASSIGNED_GROUP
COURSE_MARKED_ENTITY

3.19 Assignment Upload (for Administrator, Marker, Student and Project Group)

The administrator can view all the uploaded assignments. The marker can view the uploaded assignments for his assigned course section. The student and the project group can upload their assignments or delete uploaded assignments for themselves (Figure 28). As a part of the security considerations, the Course Manager system generates the MD5 signature for each assignment upload and stores the information in the database.

Figure 28 Page to Upload Assignments

3.19.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: assignment_list

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.19.2 Page: assignment_list

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing assignments

Input
Selected assignment

Actions
Upload assignment → Page: assignment_upload

DB Tables

COURSE_STUDENT
COURSE_MARKED_ENTITY
COURSE_GROUP
MARK_ASSIGNEE
MKD_ENT_FILE
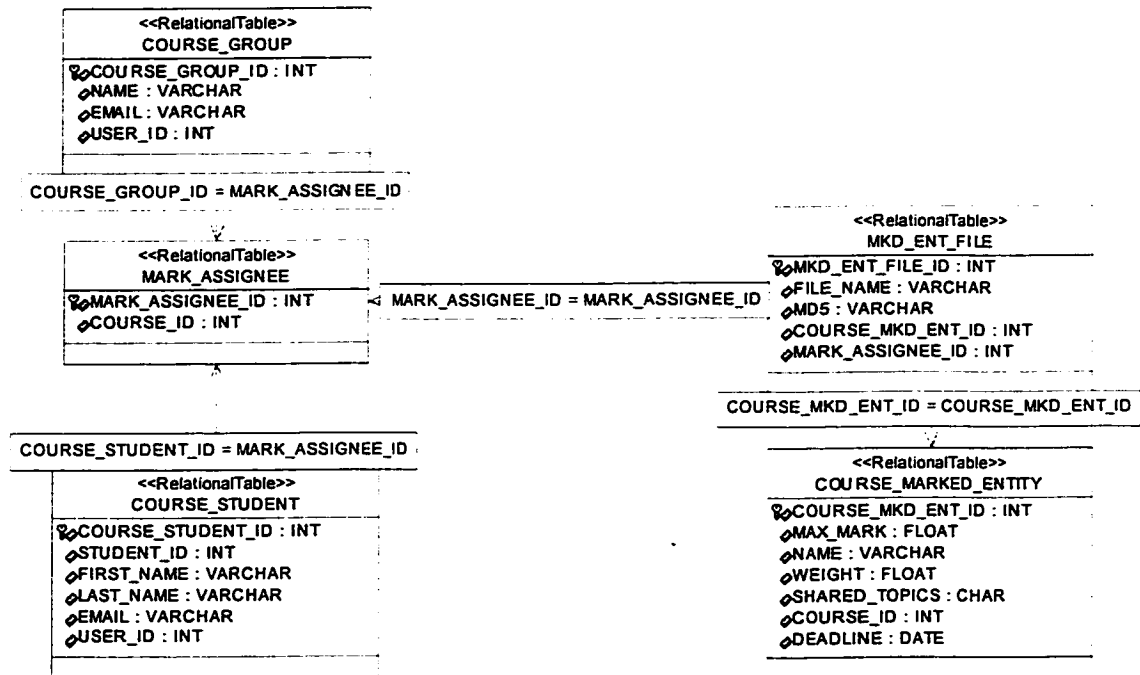
Figure 29 Tables for Assignment Upload

3.19.3 Page: assignment_upload

Display (Figure 28)
Selected course name
Selected course session
Selected course year
Selected course section
Existing uploaded assignments
Upload status (with verbose selection)

Input
Selected existing uploaded assignment
File path and name
Verbose selection

Actions
Browse → Dialog Box: choose_file

View uploaded assignment → Display uploaded assignment
Upload (with verbose selection) → Page: assignment_upload
Upload (without verbose selection) → Page: assignment_list
Delete → Page: assignment_upload
Back → Page: assignment_list

DB Tables
MKD_ENT_FILE
COURSE_DESC
COURSE_SESSION
COURSE

3.20 Course Grades (for Administrator, Professor and Marker)

The administrator can view, assign and modify marks for all the students and project groups (Figure 30). The professor and the marker can view, assign and modify marks for their assigned course sections. Marks can be assigned to an individual student or to a project group (Figure 31). Marks assigned a project group will be applied to all the current group members. If students in a project group have been assigned different marks for a group assignment, the message "See Student Marks" will be shown for group marks. Comments may be added when a mark is assigned.

Figure 30 Page to View Marks

Figure 31 Page to Assign Group Marks

3.20.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: mkd_ent_marks_list

DB Tables

```
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)
```

3.20.2 Page: mkd_ent_marks_list

Include
marks_gen_inc

Display (Figure 30)
Selected course name
Selected course session
Selected course year
Selected course section
Existing marked entites
Existing marks
Existing markers' comments
Selected display preference

Input
Selected marked entites
Selected display preference

Actions
Change display preference → Page: mkd_ent_marks_list
Assign marks → Page: assign_mkd_ent_marks

DB Tables
```
COURSE_MARKED_ENTITY
MKD_ENT_GRADE
MARK_ASSIGNEE
COURSE
COURSE_STUDENT
GROUP_MEMBER
COURSE_GROUP
```

<<RelationalTable>>
COURSE

%COURSE_ID : INT
◇SECTION : VARCHAR
◇COURSE_SESSION_ID : INT
◇SHOW_FINAL_GRADES : INT

COURSE_ID = COURSE_ID

<<RelationalTable>>
COURSE_MARKED_ENTITY

%COURSE_MKD_ENT_ID : INT
◇MAX_MARK : FLOAT
◇NAME : VARCHAR
◇WEIGHT : FLOAT
◇SHARED_TOPICS : CHAR
◇COURSE_ID : INT
◇DEADLINE : DATE

COURSE_MKD_ENT_ID = COURSE_MKD_ENT_ID

COURSE_ID = COURSE_ID

<<RelationalTable>>
MARK_ASSIGNEE

%MARK_ASSIGNEE_ID : INT
◇COURSE_ID : INT

MARK_ASSIGNEE_ID = MARK_ASSIGNEE_ID

<<RelationalTable>>
MKD_ENT_GRADE

%MKD_ENT_GRADE_ID : INT
◇MARK : FLOAT
◇COURSE_MKD_ENT_ID : INT
◇MARK_ASSIGNEE_ID : INT
◇NOTES : TEXT

COURSE_GROUP_ID = MARK_ASSIGNEE_ID

<<RelationalTable>>
COURSE_GROUP

%COURSE_GROUP_ID : INT
◇NAME : VARCHAR
◇EMAIL : VARCHAR
◇USER_ID : INT

COURSE_STUDENT_ID = MARK_ASSIGNEE_ID

<<RelationalTable>>
COURSE_STUDENT

%COURSE_STUDENT_ID : INT
◇STUDENT_ID : INT
◇FIRST_NAME : VARCHAR
◇LAST_NAME : VARCHAR
◇EMAIL : VARCHAR
◇USER_ID : INT

COURSE_GROUP_ID = COURSE_GROUP_ID

<<RelationalTable>>
GROUP_MEMBER

%COURSE_GROUP_ID : INT
%COURSE_STUDENT_ID : INT

COURSE_STUDENT_ID = COURSE_STUDENT_ID

Figure 32 Tables for Course Marks

3.20.3 Page: assign_mkd_ent_marks

Include
marks_gen_inc

Display (Figure 31)
Selected course name
Selected course session
Selected course year
Selected course section
Existing students to assign marks
Existing groups to assign marks
Existing marks
Existing markers' comments

Selected display preference

Input
Selected student to assign marks
Marks for groups
Comments for groups
Selected display preference

Actions
Change display preference → assign_mkd_ent_marks
Submit → Page: mkd_ent_marks_list
Cancel → Page: mkd_ent_marks_list
Assign marks to selected student → Page: note

DB Tables
MKD_ENT_GRADE
MARK_ASSIGNEE
COURSE_STUDENT
GROUP_MEMBER
COURSE_GROUP

3.20.4 Page: note

Include
`marks_gen_inc`

Display
Existing marks
Existing markers' comments

Input
Marks for student
Comments for student

Actions
Submit → Page: assign_mkd_ent_marks
Cancel → Page: assign_mkd_ent_marks

DB Tables
MKD_ENT_GRADE
MARK_ASSIGNEE
COURSE_STUDENT
GROUP_MEMBER
COURSE_GROUP

3.21 Course Grades (for Student)

The student can view marks for himself or for the whole course section. When marks are displayed for the whole section, the student names are not displayed. The student can also view the comments for his marks.

3.21.1 Page: select_mark_view

Include
marks_gen_inc

Display
Selected course name
Selected course session
Selected course year
Selected course section

Input
Selected view grade preference

Actions
View my grades → Page: studentmark
View class grades → Page: classmark

DB Tables
None

3.21.2 Page: studentmark

Include
marks_gen_inc

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing marked entites
Existing marks
Existing markers' comments

Input
None

Actions
Back → Page: select_mark_view

DB Tables
```
COURSE_MARKED_ENTITY
MKD_ENT_GRADE
MARK_ASSIGNEE
COURSE
COURSE_STUDENT
GROUP_MEMBER
COURSE_GROUP
```

3.21.3 Page: classmark

Include
```
marks_gen_inc
```

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing marked entites
Existing marks
Existing markers' comments

Input
None

Actions
Back → Page: select_mark_view

DB Tables
```
COURSE_MARKED_ENTITY
MKD_ENT_GRADE
MARK_ASSIGNEE
COURSE
COURSE_STUDENT
GROUP_MEMBER
COURSE_GROUP
```

3.22 Final Letter Grades (for Administrator and Professor)

The administrator and the professor can use this functionality to assign or modify

students' final letter grades according to the students' final marks (Figure 33).

Figure 33 Page to Assign Final Letter Grades
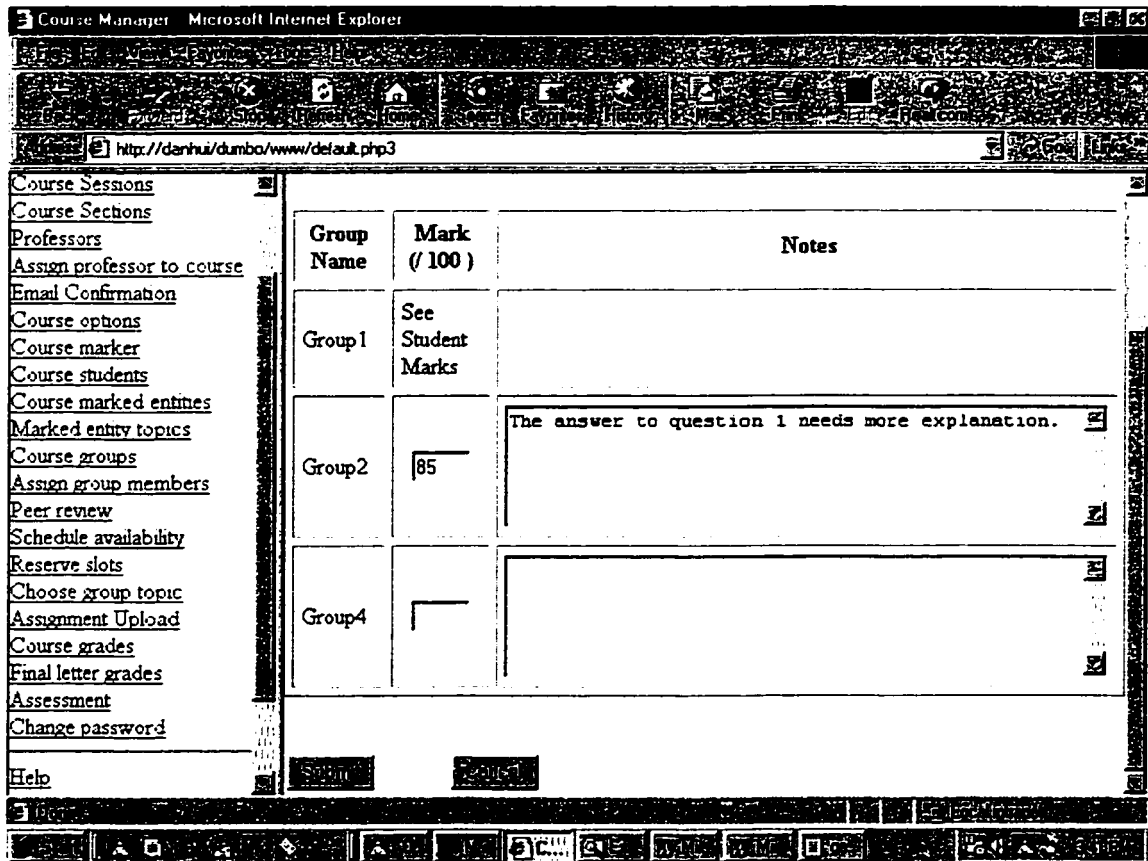
3.22.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session

Selected course year
Selected course section

Actions
Submit → Page: letter_grades_list

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

3.22.2 Page: letter_grades_list

Include
marks_gen_inc

Display (Figure 33)
Selected course name
Selected course session
Selected course year
Selected course section
Existing final marks
Existing letter grades

Input
Letter grades

Actions
Assign letter grades → Page: letter_grades_list

DB Tables
LETTER_GRADE
COURSE_MARKED_ENTITY
MKD_ENT_GRADE
MARK_ASSIGNEE
COURSE
COURSE_STUDENT
GROUP_MEMBER

87

<<RelationalTable>>
MKD_ENT_GRADE

- MKD_ENT_GRADE_ID : INT
- MARK : FLOAT
- COURSE_MKD_ENT_ID : INT
- MARK_ASSIGNEE_ID : INT
- NOTES : TEXT

COURSE_MKD_ENT_ID = COURSE_MKD_ENT_ID

<<RelationalTable>>
MARK_ASSIGNEE

- MARK_ASSIGNEE_ID : INT
- COURSE_ID : INT

MARK_ASSIGNEE_ID = MARK_ASSIGNEE_ID

COURSE_ID = COURSE_ID

COURSE_STUDENT_ID = MARK_ASSIGNEE_ID

<<RelationalTable>>
COURSE

- COURSE_ID : INT
- SECTION : VARCHAR
- COURSE_SESSION_ID : INT
- SHOW_FINAL_GRADES : INT

<<RelationalTable>>
COURSE_STUDENT

- COURSE_STUDENT_ID : INT
- STUDENT_ID : INT
- FIRST_NAME : VARCHAR
- LAST_NAME : VARCHAR
- EMAIL : VARCHAR
- USER_ID : INT

<<RelationalTable>>
COURSE_MARKED_ENTITY

- COURSE_MKD_ENT_ID : INT
- MAX_MARK : FLOAT
- NAME : VARCHAR
- WEIGHT : FLOAT
- SHARED_TOPICS : CHAR
- COURSE_ID : INT
- DEADLINE : DATE

COURSE_ID = COURSE_ID

COURSE_STUDENT_ID = COURSE_STUDENT_ID

COURSE_STUDENT_ID = COURSE_STUDENT_ID

<<RelationalTable>>
GROUP_MEMBER

- COURSE_GROUP_ID : INT
- COURSE_STUDENT_ID : INT

<<RelationalTable>>
LETTER_GRADE

- COURSE_STUDENT_ID : INT
- LETTER_GRADE : VARCHAR

Figure 34 Tables for Final Letter Grades

3.23 Assessment (for Administrator and Professor)

This functionality is provided for the administrator and the professor to create, modify or delete assessments (Figure 35), and to add, modify or remove questions and answers for assessments (Figure 36). An assessment is a kind of course marked entity, so it has a maximum mark and a weight in the final mark. The start date and the end date of an assessment define a period, during which students can take the assessment. An assessment also has a type and a duration. Currently the Course Manager system supports only multiple-choice assessments. When an assessment is created, the number of questions to be used in the assessment is specified. More questions may be provided than the specified number. The questions to be used in each assessment are selected randomly from the whole question set of the assessment. The order in which the questions and answers are presented in each assessment is also selected randomly by the system.

3.23.1 Page: select_course

Display (Figure 3)
Available course names
Available course sessions
Available course years
Available course sections

Input
Selected course name
Selected course session
Selected course year
Selected course section

Actions
Submit → Page: exam_admin

DB Tables
COURSE_DESC
COURSE_SESSION
COURSE
COURSE_PROF (for instructor)
PROFESSOR (for instructor)
COURSE_MARKER (for marker)
COURSE_GROUP (for group)
COURSE_STUDENT (for student)
MARK_ASSIGNEE (for group, student)

Course Manager - Microsoft Internet Explorer

http://danhui/dumbo/www/default.php3

Professors
Assign professor to course
Email Confirmation
Course options
Course marker
Course students
Course marked entities
Marked entity topics
Course groups
Assign group members
Peer review
Schedule availability
Reserve slots
Choose group topic
Assignment Upload
Course grades
Final letter grades
Assessment
Change password

Help

| Course: | COMP353 | 2002 | EE |

Assessment Title	Assessment Test 1
Weight (%)	25
Maximum Mark	100
Start Date	2003-02-16 (yyyy-mm-dd)
End Date	2003-03-28 (yyyy-mm-dd)
Type	Multiple Choice
Duration in Minutes	1
Number of Questions	3

☐ Assign questions to assessment

Update   Delete   Cancel

Figure 35 Page to Modify or Delete Assessment

90
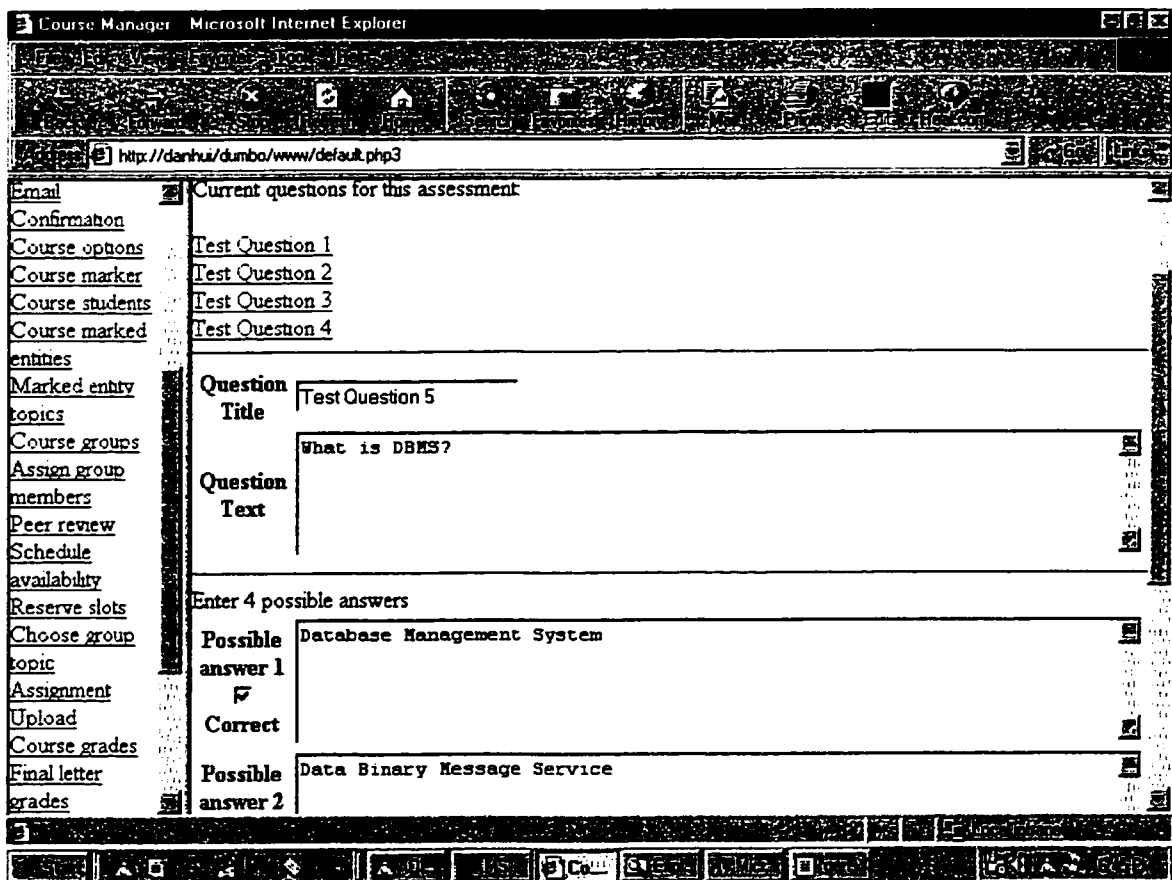
Figure 36 Page to Add Question and Answer Choices

3.23.2 Page: exam_admin

Display
Selected course name
Selected course session
Selected course year
Selected course section
Existing assessments

Input
Selected existing assessment

Actions
Insert new assessment → Page: course_assessment (new assessment)
Modify existing assessments → Page: course_assessment (existing assessment)

DB Tables
ASSESSMENT

91

3.23.3 Page: course_assessment

Display (Figure 35)
- **New assessment**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
- **Existing assessment**
  Selected course name
  Selected course session
  Selected course year
  Selected course section
  Previous assessment title
  Previous assessment weight
  Previous assessment maximum mark
  Previous assessment start date
  Previous assessment end date
  Previous assessment type
  Previous assessment duration
  Previous number of questions

Input
- **New assessment**
  Assessment title
  Assessment weight
  Assessment maximum mark
  Assessment start date
  Assessment end date
  Assessment type
  Assessment duration
  Number of questions
  Assign questions selection
- **Existing assessment**
  New assessment title
  New assessment weight
  New assessment maximum mark
  New assessment start date
  New assessment end date
  New assessment type
  New assessment duration
  New number of questions
  Assign questions selection

Actions

- **New assessment**
  Insert (without assign questions selection) → Page: exam_admin
  Insert (with assign questions selection) → Page: questions_list
  Cancel → Page: exam_admin

- **Existing assessment**
  Update (without assign questions selection) → Page: exam_admin
  Update (with assign questions selection) → Page: questions_list
  Delete → Page: exam_admin
  Cancel → Page: exam_admin

DB Tables
ASSESSMENT
COURSE_MARKED_ENTITY

3.23.4 Page: questions_list

Display (Figure 36)
Selected course name
Selected course session
Selected course year
Selected course section
Existing questions

Input
Question title
Question text
Question answers
Correct answer
Selected question

Actions
Insert new question → Page: questions_list
Cancel → Page: questions_list
Modify existing questions → Page: question_edit

DB Tables
ASSESSMENT
QUESTION
ANSWER

3.23.5 Page: question_edit

Display
Selected course name
Selected course session
Selected course year
Selected course section
Previous question title
Previous question text
Previous question answers
Previous correct answer

Input
New question title
New question text
New question answers
New correct answer

Actions
Update → Page: questions_list
Delete → Page: questions_list
Cancel → Page: questions_list

DB Tables
QUESTION
ANSWER

3.24   Assessment (for Student)

The student can use this functionality to take an assessment between the starting date and the ending date of the assessment (Figure 37). Once the student starts an assessment, the timer starts running and the student has to finish the assessment within the time limit of the assessment. The student commits his answer to each question given by the Course Manager system, which is randomly selected from a question set. The answer cannot be changed after it is committed. If the assessment is interrupted, for example the student closes the browser accidentally, he can resume the assessment as long as he has not exceeded the time limit. The system keeps track of the answered questions and the used time for each student. When time is up, the student will get a notice and can no longer take the assessment. The application calculates the assessment score based on the number of correct answers within the time limit. After the student finishes an assessment normally (he answered all the questions) or abnormally (he exceeded the time limit), he can view his score.

3.24.1   Page: select_assessment

Display
Student course names
Student course sessions
Student course years
Student course sections
Available assessments

Input
Selected assessment

Actions
Select assessment → Page: take_assessment

DB Tables
ASSESSMENT
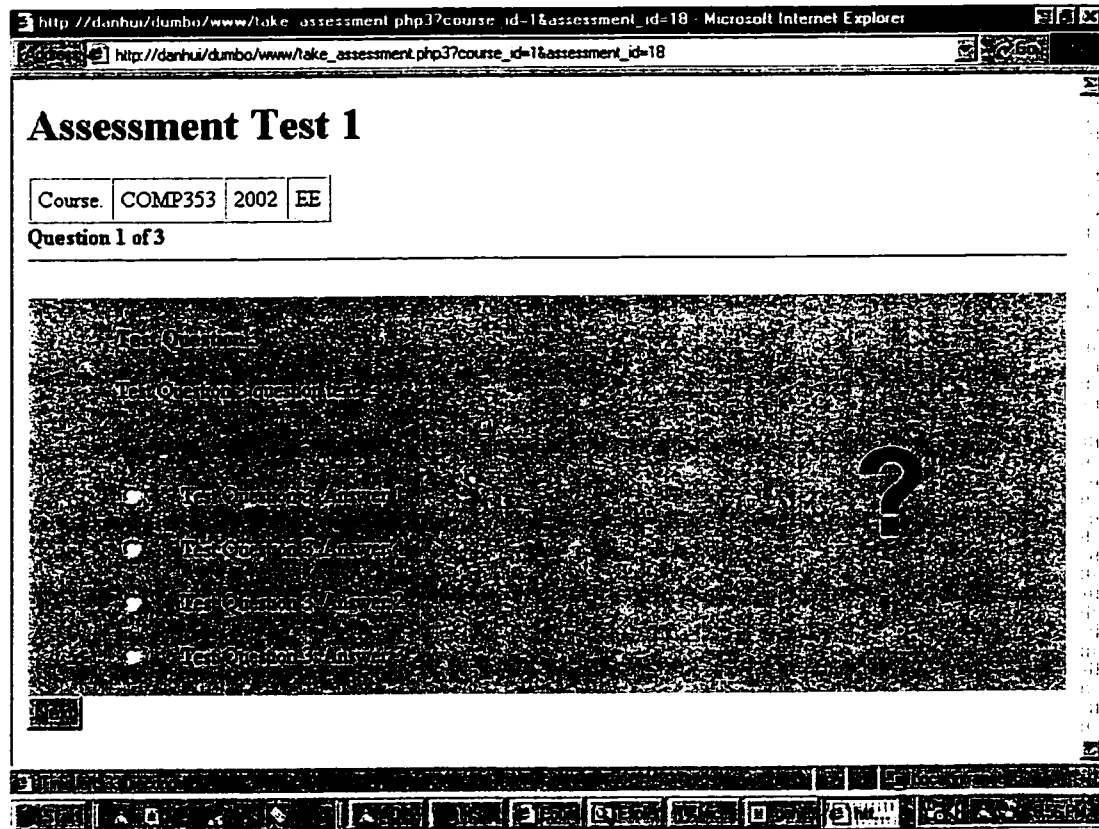
Figure 37 Page to Take Assessment

3.24.2 Page: take_assessment

Display (Figure 37)
Student course names
Student course sessions
Student course years
Student course sections
Total number of questions
Number of questions taken
Current question title
Current question text
Current multiple choices
Time left
Assessment score

Input

Selected answer

Actions
Next question → Page: take_assessment
Time out → Page: take_assessment

DB Tables
ASSESSMENT
USER_ASSESSMENT
USER_ANSWER
ANSWER
QUESTION
COURSE_STUDENT
MKD_ENT_GRADE
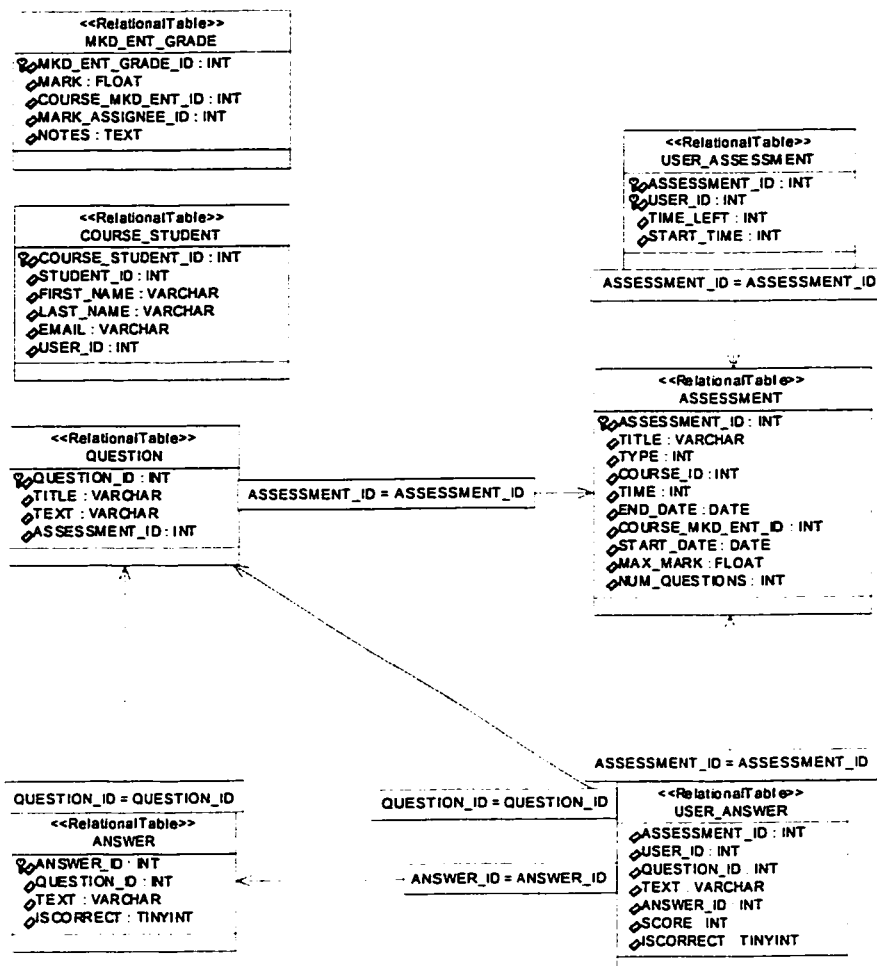
Figure 38 Tables for Assessments

3.25 Change Password

The user can change his password after his user account is created in the Course Manager

system.

3.25.1 Page: change_password

Display
Available user names

Input
Selected user name
New password

Actions
Change password → Page: home

DB Tables
DB_USER

3.26 Help

The user can get online help for the functionality which he can access.

3.26.1 Page: admhelp (profhelp, markerhelp, studenthelp)

Display
Help information for functionality within the user's access privileges

Input
None

Actions
None

DB Tables
None

3.27 Login & Logout

The Course Manager system authenticates each user when he logs in.

3.27.1 Page: default

Include
main_menu
home

Display
Links to functionality

Input
Selected functionality

Actions
Select functionality → Entry page for selected functionality

DB Tables
DB_USER

3.27.2 Page: logoff

Display
None

Input
None

Actions
None

DB Tables
None

4   Database Design

This chapter describes how data are managed in the Course Manager system. Figure 39 gives an overview of all the database tables used in the system. The following tables describe the data fields and types in detail.

Table: **ANSWER**

This table stores all the answer choices for each question used in the assessments. ISCORRECT is used to indicate whether or not the answer is the correct choice for the question.

Field	Type	Null	Key	Default	Extra
ANSWER_ID	int(11)		PRI	NULL	auto_increment
QUESTION_ID	int(11)			0	
TEXT	text				
ISCORRECT	tinyint(1)			0	

Table: **ASSESSMENT**

This table contains information about the assessments.

Field	Type	Null	Key	Default	Extra
ASSESSMENT_ID	int(11)		PRI	NULL	auto_increment
TITLE	varchar(50)				
TYPE	int(11)			0	
COURSE_ID	int(11)			0	
TIME	int(11)	YES		NULL	
END_DATE	date	YES		NULL	
COURSE_MKD_ENT_ID	int(11)	YES		NULL	
START_DATE	date	YES		NULL	
MAX_MARK	float(10,2)	YES		NULL	
NUM_QUESTIONS	int(11)			0	

Figure 39 Database Design

Table: **COURSE**

This table stores course information including course ID, section and session ID.

SHOW_FINAL_GRADES is used to indicate whether the final marks should calculated

when displaying the grades. (COURSE_SESSION_ID,SECTION) must be unique.

Field	Type	Null	Key	Default	Extra
COURSE_ID	int(11)		PRI	0	
SECTION	varchar(10)				
COURSE_SESSION_ID	int(11)		MUL	0	
SHOW_FINAL_GRADES	int(11)			0	

Table: **COURSE_DESC**

This table contains the course numbers and course names for all the courses.

Field	Type	Null	Key	Default	Extra
COURSE_DESC_ID	int(11)		PRI	0	
COURSE_NUMBER	varchar(15)		UNI		
COURSE_NAME	varchar(50)				

Table: **COURSE_GROUP**

This table contains information about project groups.

Field	Type	Null	Key	Default	Extra
COURSE_GROUP_ID	int(11)		PRI	0	
NAME	varchar(50)				
EMAIL	varchar(50)	YES		NULL	
USER_ID	int(11)			0	

Table: **COURSE_MARKED_ENTITY**

This table contains information for all the course marked entities, such as assignments,

projects, demos and exams. The entity may have a deadline. WEIGHT is the percentage

that the marked entity takes in the final mark. SHARED_TOPICS is used to indicate whether the entity allows topics to be shared among project groups.

Field	Type	Null	Key	Default	Extra
COURSE_MKD_ENT_ID	int(11)		PRI	0	
MAX_MARK	float(10,2)			0.00	
NAME	varchar(30)				
WEIGHT	float(5,2)			0.00	
SHARED_TOPICS	char(1)				
COURSE_ID	int(11)	YES		NULL	
DEADLINE	date	YES		NULL	

Table: **COURSE_MARKER**

This table stores information about course markers and their assigned courses.

(COURSE_ID,USER_ID) must be unique.

Field	Type	Null	Key	Default	Extra
COURSE_MARKER_ID	int(11)		PRI	0	
FIRST_NAME	varchar(50)				
LAST_NAME	varchar(50)				
EMAIL	varchar(50)	YES		NULL	
USER_ID	int(11)			0	
COURSE_ID	int(11)		MUL	0	

Table: **COURSE_PROF**

This table describes the relationships between courses and professors.

Field	Type	Null	Key	Default	Extra
COURSE_PROF_ID	int(11)		PRI	0	
COURSE_ID	int(11)		UNI	0	
PROFESSOR_ID	int(11)			0	

Table: **COURSE_SESSION**

This table stores information about course sessions. SESSION can be Fall, Winter or Summer. (COURSE_DESC_ID,SESSION,YEAR) must be unique.

Field	Type	Null	Key	Default	Extra
COURSE_SESSION_ID	int(11)		PRI	0	
COURSE_DESC_ID	int(11)		MUL	0	
SESSION	varchar(10)				
YEAR	int(11)			0	

Table: **COURSE_STUDENT**

This table stores information for all the students.

Field	Type	Null	Key	Default	Extra
COURSE_STUDENT_ID	int(11)		PRI	0	
STUDENT_ID	int(11)			0	
FIRST_NAME	varchar(30)	YES		NULL	
LAST_NAME	varchar(30)	YES		NULL	
EMAIL	varchar(50)	YES		NULL	
USER_ID	int(11)			0	

Table: **DB_USER**

This table contains the user account information for the Course Manager system.

ACCESS_PRIV uses one character to indicates whether the user is administrator,

professor, marker, student or project group.

Field	Type	Null	Key	Default	Extra
USER_ID	int(11)		PRI	0	
USER_NAME	varchar(30)		UNI		
PASSWORD	varchar(30)				
ACCESS_PRIV	char(1)				

Table: **EMAIL_CONF**

This table contains information about the emails sent to the users when their accounts are

created in the Course Manager system.

Field	Type	Null	Key	Default	Extra
EMAIL_ID	int(11)		PRI	0	
MESSAGE_TYPE	varchar(30)				
SUBJECT	varchar(30)				
MESSAGE	varchar(250)				
REPLY_TO	varchar(30)	YES		NULL	
ACTIVATION	int(1)	YES		NULL	

Table: **GROUP_MEMBER**

This table defines the relationships between students and their project groups.

Field	Type	Null	Key	Default	Extra
COURSE_GROUP_ID	int(11)		PRI	0	
COURSE_STUDENT_ID	int(11)		PRI	0	

Table: **LETTER_GRADE**

This table is used to store the letter grades for students.

Field	Type	Null	Key	Default	Extra
COURSE_STUDENT_ID	int(11)		PRI	0	
LETTER_GRADE	varchar(30)				

Table: **LINK_COURSE_MARKED_ENTITY**

This is a legacy table which describes the relationships between courses and marked

entities.

Field	Type	Null	Key	Default	Extra
COURSE_MKD_ENT_ID	int(11)		PRI	0	
MARKED_ENTITY_ID	int(11)			0	
COURSE_ID	int(11)			0	

Table: **MARK_ASSIGNEE**

A mark assignee is either a student or a project group, and MARK_ASSIGNEE_ID refers

to COURSE_STUDENT_ID and COURSE_GROUP_ID respectively. This table defines

the relationships between courses and students or project groups.

Field	Type	Null	Key	Default	Extra
MARK_ASSIGNEE_ID	int(11)		PRI	0	
COURSE_ID	int(11)			0	

Table: **MEETING_TIME_SLOT**

This table contains information about the meeting time slots.

Field	Type	Null	Key	Default	Extra
TIME_SLOT_ID	int(11)		PRI	0	
MEETING_DATE	date			0000-00-00	
START_TIME	time			00:00:00	
END_TIME	time			00:00:00	
COURSE_ID	int(11)			0	
PURPOSE	char(1)				

Table: **MKD_ENT_FILE**

This table stores information about the uploaded assignments or projects, including the

MD5 signatures calculated on them.

Field	Type	Null	Key	Default	Extra
MKD_ENT_FILE_ID	int(11)		PRI	NULL	auto_increment
FILE_NAME	varchar(80)				
MD5	varchar(250)	YES		NULL	
COURSE_MKD_ENT_ID	int(11)	YES		NULL	
MARK_ASSIGNEE_ID	int(11)	YES		NULL	

Table: **MKD_ENT_GRADE**

This table stores the marks and markers' comments for the marked entities of the students or projects groups. : (COURSE_MKD_ENT_ID,MARK_ASSIGNEE_ID) must be unique.

Field	Type	Null	Key	Default	Extra
MKD_ENT_GRADE_ID	int(11)		PRI	NULL	auto_increment
MARK	float(5,2)			0.00	
COURSE_MKD_ENT_ID	int(11)		MUL	0	
MARK_ASSIGNEE_ID	int(11)			0	
NOTES	text	YES		NULL	

Table: **PEER_REVIEW**

This table stores the peer review scores and comments for students who have a project group.

Field	Type	Null	Key	Default	Extra
COURSE_GROUP_ID	int(11)		PRI	0	
COURSE_STUDENT_ID	int(11)		PRI	0	
REVIEWER_COURSE_STUDENT_ID	int(11)		PRI	0	
REVIEW_SCORE	int(3)			0	
COMMENTS	text	YES		NULL	
ISUSED	tinyint(1)			1	

Table: **PROFESSOR**

This table stores information about the professors.

Field	Type	Null	Key	Default	Extra
PROFESSOR_ID	int(11)		PRI	0	
FIRST_NAME	varchar(50)				
LAST_NAME	varchar(50)				
EMAIL	varchar(50)	YES		NULL	
USER_ID	int(11)			0	

Table: **PROJECT_ASSIGNED_GROUP**

This table defines the relationships between project groups and project topics which they

have chosen. (COURSE_GROUP_ID,PROJECT_TOPIC_ID) must be unique.

Field	Type	Null	Key	Default	Extra
PROJ_ASSIGNED_ GRP_ID	int(11)		PRI	0	
COURSE_GROUP_ID	int(11)		MUL	0	
PROJECT_TOPIC_ID	int(11)			0	

Table: **PROJECT_TOPIC**

This table stores information about the project topics and defines the relationships

between project topics and course marked entities.

Field	Type	Null	Key	Default	Extra
PROJECT_TOPIC_ID	int(11)		PRI	0	
NAME	varchar(50)				
DESCRIPTION	varchar(255)	YES		NULL	
COURSE_MKD_ENT_ID	int(11)	YES		NULL	

Table: **QUESTION**

This table stores information about the questions used in assessments.

Field	Type	Null	Key	Default	Extra
QUESTION_ID	int(11)		PRI	NULL	auto_increment
TITLE	varchar(50)				
TEXT	text				
ASSESSMENT_ID	int(11)			0	

Table: **SCHED_MEETING**

This table defines the relationships between meeting time slots and project groups who

have reserved them.

Field	Type	Null	Key	Default	Extra
TIME_SLOT_ID	int(11)		PRI	0	
COURSE_GROUP_ID	int(11)			0	

Table: **USER_ANSWER**

This table stores the user's answer to each question in an assessment.

Field	Type	Null	Key	Default	Extra
ASSESSMENT_ID	int(11)			0	
USER_ID	int(11)			0	
QUESTION_ID	int(11)			0	
TEXT	text				
ANSWER_ID	int(11)			0	
SCORE	int(11)	YES		NULL	
ISCORRECT	tinyint(1)			0	

Table: **USER_ASSESSMENT**

This table keeps track of the time that a user has spent in an assessment.

Field	Type	Null	Key	Default	Extra
ASSESSMENT_ID	int(11)		PRI	0	
USER_ID	int(11)		PRI	0	
TIME_LEFT	int(11)			0	
START_TIME	int(11)			0	

5   Contributions and Future Work

The contributions of this major report to the Course Manger system are to design and implement the Peer Review, Final Letter Grades features; and to improve and upgrade the Assignment Upload, Course Students, Course Groups, Course Grades, Assessment and Email Confirmation features.

The following features can be added to the Course Manager system in the future:

1. Provide the functionality for the administrator and the professor to perform a backup of the system and restore it for the administration, both partially and completely (unload old sessions/years and reload them on demand). Allow a method to choose different sessions/years to unload and reload.

2. Provide the facility to perform course evaluation. The students should have absolute confidentiality.

3. Add an election system for class groups/committees to manage the following information:

   - List of groups/committees;

   - Number of members in each group/committee;

   - Candidate list;

   - Voter list;

   - Confidentiality in voting.

4. Find a solution to the PHP defect of sending batch emails with "nobody" as the sender.

5. Provide an interface for the professor to see the details of all the students and to send emails to one or more students with a single click.

6 References

[1]   Client/Server Software Architecture,
       http://www.sei.cmu.edu/str/descriptions/clientserver_body.html

[2]   PHP Documents, http://www.php.net/docs.php

[3]   PHP Manual, http://www.phpbuilder.com/manual

[4]   MySQL Documents, http://www.mysql.com/documentation

[5]   Bipin C. Desai, *An Introduction to Database Systems,* West St Paul, 1990.