Course Management System

Narasimha Rao. K
Anil Kumar S.R.
Sheetal Sonare

Computer Science and Engineering
IIT Bombay
narasimha@cse.iitb.ac.in
anil@cse.iitb.ac.in
sheetal@cse.iitb.ac.in

November 28, 2004
Outline

- Introduction
- Requirements
- System Use Cases
- Test Plan and Test Cases
- Flow Chart
- System Sequence Diagram
- System Architecture
- DataBase Tables
- Technologies to be Used
- References
Course Management System (CMS) is a web-based course application used by students and professors.

The objective of this product is to provide interaction between students and teachers.

This product provides
- Online discussions
- Online quizzes
- Automatic group allocation
- Statistical analysis of students performance
- Others
Requirements for Students

Students must be able to:

- Login to the system
- Sign up for access to a course
- Download files within/related to their courses
- Taking quizzes of multiple choice questions
- Access a Project Management Room (PMR) that will be only accessible to group members.
Requirements for Students Cont..

Upon logging into this room they can:

- view what other group members are currently logged into the room as well as email addresses
- access a real-time chat room to communicate online with other group members
- post and get files
- post and get messages
- log out of the project meeting room
- Logout of the whole system.
Requirements for Instructors

Instructors must have all the abilities as the student. Additionally, they can:

- Create a course for their students to access
- Delete a course
- Create an interactive quiz that their students can take by using a simple interface where they provide the questions, choices, and correct answers.
- Create a PMR that can will be accessible only by specified group members
Use Cases describe the sequence of events an actor, using a system, performs to complete a process.

This necessitates identifying the actors as well as the processes they perform.

The following table illustrates the actors identified along with the processes / use cases and sequence diagram they take part in.
<table>
<thead>
<tr>
<th>Use Cases</th>
<th>Actors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login the system</td>
<td>Student</td>
</tr>
<tr>
<td>Logout the system</td>
<td>Instructor</td>
</tr>
<tr>
<td>Download files</td>
<td></td>
</tr>
<tr>
<td>Upload files</td>
<td></td>
</tr>
<tr>
<td>Post tests/quizzes</td>
<td></td>
</tr>
<tr>
<td>Take test/quizzes</td>
<td></td>
</tr>
<tr>
<td>Access Chat Room</td>
<td></td>
</tr>
<tr>
<td>Create Project Meeting Room (PMR)</td>
<td></td>
</tr>
<tr>
<td>Access Project Meeting Room (PMR)</td>
<td></td>
</tr>
<tr>
<td>Post messages</td>
<td></td>
</tr>
<tr>
<td>Retrieve messages</td>
<td></td>
</tr>
</tbody>
</table>
## Use Case: Take Tests/Quizzes

<table>
<thead>
<tr>
<th><strong>Use Case</strong></th>
<th><strong>Take Tests/Quizzes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors</strong></td>
<td>Student</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Primary</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The student takes a multiple choice test/quiz, submits his/her answers, and is given a score back.</td>
</tr>
</tbody>
</table>
# Use Case: Post Messages

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Post Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Student</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>The student posts a message in a forum that can be read by all other members of the class.</td>
</tr>
</tbody>
</table>
### Use Case: Create PMR

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Create PMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Instructor</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>The instructor creates a group for students to work on projects together. In this group, students have exclusive access to files and messages that cannot be accessed by other members of the class that are not in the specified group.</td>
</tr>
</tbody>
</table>
Use Case: Download files while in PMR

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Download files while in PMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Student</td>
</tr>
<tr>
<td>Type</td>
<td>Secondary</td>
</tr>
<tr>
<td>Description</td>
<td>While in the PMR, the student has permission to access and/or download files that have been uploaded by other members in the group.</td>
</tr>
</tbody>
</table>
Use Case: Post messages while in PMR

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Post messages while in PMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Student</td>
</tr>
<tr>
<td>Type</td>
<td>Secondary</td>
</tr>
<tr>
<td>Description</td>
<td>While in the PMR, the student posts the messages in a forum that can be viewed only by members of the same group and not anyone else in the class.</td>
</tr>
</tbody>
</table>
### Logout from the System

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Logout from the system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Users (Students/Instructor)</td>
</tr>
<tr>
<td>Purpose</td>
<td>Primary</td>
</tr>
<tr>
<td>Overview</td>
<td>Users log out their user accounts</td>
</tr>
</tbody>
</table>

Narasimha Rao. K  Anil Kumar S.R.  Sheetal Sonare
Computer Science and Engineering  IIT Bombay  narasimha@cse.iitb.ac.in
### Events

<table>
<thead>
<tr>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. This use case begins when a user hit the button exit in the system.</td>
<td>2. Displays a confirmation message to the user. The system stores the current time of day as end time</td>
</tr>
<tr>
<td></td>
<td>3. Logout the user.</td>
</tr>
</tbody>
</table>
### Post Tests/Quizzes

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Post tests/quizzes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Instructor</td>
</tr>
<tr>
<td>Purpose</td>
<td>To upload Tests/Quizzes online from Instructor account</td>
</tr>
<tr>
<td>Overview</td>
<td>The instructor uses the given interface to post questions, multiple answer choices, and a solution key to practice tests/quizzes.</td>
</tr>
</tbody>
</table>
## Events

<table>
<thead>
<tr>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. This begins when the instructor upload the tests/quizzes file to the test area of the system. Instructor can also set the timer.</td>
<td>2. The system transfers the files to testarea, and wait for a signal from student site to take the test.</td>
</tr>
</tbody>
</table>
Test Plan and Test Cases

- Software fails because of errors. Testing is the process of finding errors.
- Even though we have not yet reached the implementation phase, we can still create test cases based on the required functionalities of our system.
- A test case is made up of inputs to a program, and the expected outputs. Each test case examines a certain part of the system.
Use Case: Post Quizzes (Instructor)

**Input**
The instructor will either enter a question or a possible solution in the text box.

**Expected Output**
Take the input value from the form and compare it with the original text that was entered. The original input and the output from the form should be identical.
Use Case: Post Quizzes (Instructor)

Input
The instructor will either enter a question or a possible solution in the text box.

Expected Output
Take the input value from the form and compare it with the original text that was entered. The original input and the output from the form should be identical.
Use Case: Uploading and Downloading Files (Student and Instructor)

Input
Create a file of some sort that will be used to check the functionality of our uploading and downloading methods.

Expected Output
First upload the file to the site, then later download it and check it for consistency with the original file. The name and contents should be identical. This will be true whether files are uploaded or downloaded while in the regular site or once in the PMR.
Use Case: Uploading and Downloading Files (Student and Instructor)

Input
Create a file of some sort that will be used to check the functionality of our uploading and downloading methods.

Expected Output
First upload the file to the site, then later download it and check it for consistency with the original file. The name and contents should be identical. This will be true whether files are uploaded or downloaded while in the regular site or once in the PMR.
Use Case: Take Tests/Quizzes (Student)

**Input**
The student will provide some input as a response to questions presented.

**Expected Output**
Take the values inputted by the student and compare it to the values taken from the form. The student’s input and the output from the form should be identical.
Use Case: Take Tests/Quizzes (Student)

**Input**
The student will provide some input as a response to questions presented.

**Expected Output**
Take the values inputted by the student and compare it to the values taken from the form. The student’s input and the output from the form should be identical.
Use Case: Access that Room (Student and Instructor)

**Input**
Login to the chat room by providing a user name and password.

**Expected Output**
If the chat room is accessed, a window should pop up to indicate that the proper connection has been established and it should also alert the user when the connection has been broken.
Use Case: Access that Room (Student and Instructor)

Input
Login to the chat room by providing a user name and password.

Expected Output
If the chat room is accessed, a window should pop up to indicate that the proper connection has been established and it should also alert the user when the connection has been broken.
Use Case: Posting and Retrieving Messages (Student and Instructor)

**Input**
create a message to check the functionality of our posting and retrieving methods.

**Expected Output**
First post the message in the forum, then later retrieve it from the site and check it for consistency with the original message. The name and contents should be identical. This will be true whether messages are posted or retrieved while in the regular site or once in the pmr.
Use Case: Posting and Retrieving Messages (Student and Instructor)

Input
create a message to check the functionality of our posting and retrieving methods.

Expected Output
First post the message in the forum, then later retrieve it from the site and check it for consistency with the original message. the name and contents should be identical. this will be true whether messages are posted or retrieved while in the regular site or once in the pmr.
Partitioning

Partitioning

User

GUI

Controller

Features

Storage

Narasimha Rao. K
Anil Kumar S.R.
Sheetal Sonare

Computer Science and Engineering
IIT Bombay

narasimha@cse.iitb.ac.in
anil@cse.iitb.ac.in
sheetal@cse.iitb.ac.in
System Architecture

- DATABASE
- ADDITIONAL HELP
- DBMANAGER
- SERVLETS
- CLIENTREQUEST
- WEBPAGE
- CLIENT
- SERVER
- RESPONSE

SYSTEM ARCHITECTURE
### DataBase Tables

- **CMSUser**(`userId`, `firstName`, `lastName`, `login`, `password`, `email`, `isProf`)
- **CMSCourse**(`courseId`, `userId`, `courseName`)
- **CMSEnroll**(`userId`, `courseId`)
- **CMSFile**(`courseId`, `filename`, `filepath`, `time`, `date`)
- **ProfAnnounce**(`userId`, `courseId`, `time`, `announcement`, `day`)
- **Quiz**(`quizId`, `courseId`, `title`)
DataBase Tables

- CMSUser(userId, firstName, lastName, login, password, email, isProf)
- CMSCourse(courseId, userId, courseName)
- CMSEnroll(userId, courseId)
- CMSFile(courseId, filename, filepath, time, date)
- ProfAnnounce(userId, courseId, time, announcement, day)
- Quiz(quizId, courseId, title)
Course Management System

DataBase Tables

- CMSUser(userId, firstName, lastName, login, password, email, isProf)
- CMSCourse(courseld, userId, courseName)
- CMSEnroll(userId, courseld)
- CMSFile(courseld, filename, filepath, time, date)
- ProfAnnounce(userId, courseld, time, announcement, day)
- Quiz(quizId, courseld, title)
DataBase Tables

- CMSUser(userId, firstName, lastName, login, password, email, isProf)
- CMSCourse(courseld, userId, courseName)
- CMSEnroll(userId, courseld)
- CMSFile(courseld, filename, filepath, time, date)
- ProfAnnounce(userId, courseld, time, announcement, day)
- Quiz(quizId, courseld, title)
Course Management System

DataBase Tables

- CMSUser(userId, firstName, lastName, login, password, email, isProf)
- CMSCourse(courseId, userId, courseName)
- CMSEnroll(userId, courseId)
- CMSFile(courseId, filename, filepath, time, date)
- ProfAnnounce(userId, courseId, time, announcement, day)
- Quiz(quizId, courseId, title)
CMSUser(userId, firstName, lastName, login, password, email, isProf)
CMSCourse(courseId, userId, courseName)
CMSEnroll(userId, courseId)
CMSFile(courseId, filename, filepath, time, date)
ProfAnnounce(userId, courseId, time, announcement, day)
Quiz(quizId, courseId, title)
DataBase Tables Cont..

- Question(quizId, qNum, question)
- Answer(quizId, qNum, answer, isRight)
- QuizResult(userId, quizId, score)
- PMR(courseld, pmrId, pmrName)
- PMRGroup(userId, pmrId, lastVisit)
- PMRFFile(pmrId, userId, version, commitTime, filename, filepath, commitDate)
- Question(quizId, qNum, question)
- Answer(quizId, qNum, answer, isRight)
- QuizResult(userId, quizId, score)
- PMR(courseId, pmrId, pmrName)
- PMRGroup(userId, pmrId, lastVisit)
- PMRFile(pmrId, userId, version, commitTime, filename, filepath, commitDate)
Database Tables Cont..

- Question(quizId, qNum, question)
- Answer(quizId, qNum, answer, isRight)
- QuizResult(userId, quizId, score)
- PMR(courseld, pmrId, pmrName)
- PMRGroup(userId, pmrId, lastVisit)
- PMRFile(pmrId, userId, version, commitTime, filename, filepath, commitDate)
DataBase Tables Cont..

- Question(quizId, qNum, question)
- Answer(quizId, qNum, answer, isRight)
- QuizResult(userId, quizId, score)
- PMR(courseld, pmrId, pmrName)
- PMRGroup(userId, pmrId, lastVisit)
- PMRFFile(pmrId, userId, version, commitTime, filename, filepath, commitDate)
DataBase Tables Cont..

- Question(quizId, qNum, question)
- Answer(quizId, qNum, answer, isRight)
- QuizResult(userId, quizId, score)
- PMR(courseId, pmrId, pmrName)
- PMRGroup(userId, pmrId, lastVisit)
- PMRFile(pmrId, userId, version, commitTime, filename, filepath, commitDate)
Database Tables Cont..

- Question(quizId, qNum, question)
- Answer(quizId, qNum, answer, isRight)
- QuizResult(userId, quizId, score)
- PMR(courseld, pmrId, pmrName)
- PMRGroup(userId, pmrId, lastVisit)
- PMRFile(pmrId, userId, version, commitTime, filename, filepath, commitDate)
Technologies Used

- Java.
- Java Servlet.
- HTML
- Oracle 8i.
We hope that Course Management System encourages the availability of more educational tools.

Course Management System will ensure that instruction and study extends beyond the classroom, and is easily accessible and useable to those involved.
Gnu Plot

\[
\frac{\sin(x + y^2)}{(x + y^2)}
\]

Narasimha Rao. K
Anil Kumar S.R.
Sheetal Sonare
Computer Science and Engineering
IIT Bombay
narasimha@cse.iitb.ac.in
anil@cse.iitb.ac.in
sheetal@cse.iitb.ac.in
## Use Case: Login to the system

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Login to the system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>User (Student/Instructor)</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>Users login to the system with their passwords and usernames</td>
</tr>
</tbody>
</table>
## Use Case: Logout from the system

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Logout from the system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>User (Student/Instructor)</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>Users logout their user accounts.</td>
</tr>
</tbody>
</table>
Use Case: Upload Files

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Upload files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Instructor</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>The instructor uploads any course-related materials that can be downloaded by either the students in the class.</td>
</tr>
<tr>
<td>Use Case</td>
<td>Download files</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Actors</td>
<td>Instructor</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>The instructor has access to view and/or download files that have been uploaded by the students in the class.</td>
</tr>
</tbody>
</table>
## Use Case: Download Files

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Download files</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actors</strong></td>
<td>Student</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Primary</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The student has access to view and/or download files that have been uploaded by either the instructor and/or other students in the class.</td>
</tr>
</tbody>
</table>
Use Case: Post Messages, Instructor

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Post messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Instructor</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>The instructor posts a message in a forum that can be read by all other members of the class.</td>
</tr>
</tbody>
</table>
Use Case: Retrieve Messages, Student

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Retrieve messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Student</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>The student retrieves/views any messages that has been posted in the forum.</td>
</tr>
</tbody>
</table>
Use Case: Retrieve Messages, Instructor

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Retrieve messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Instructor</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>instructor retrieves/views any messages that has been posted in the forum.</td>
</tr>
</tbody>
</table>
## Use Case: Post Tests/Quizzes

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Post tests/quizzes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Instructor</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>The instructor uses the given interface to post questions, multiple answer choices, and a solution key to practice tests/quizzes.</td>
</tr>
</tbody>
</table>
## Use Case: Access PMR

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Access PMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Student</td>
</tr>
<tr>
<td>Type</td>
<td>Primary</td>
</tr>
<tr>
<td>Description</td>
<td>The student enters the group and has access to files and messages that are only accessible exclusively by other members in the same group.</td>
</tr>
</tbody>
</table>
## Use Case: Access chat room while in PMR

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Access chat room while in PMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors</td>
<td>Student</td>
</tr>
<tr>
<td>Type</td>
<td>Secondary</td>
</tr>
<tr>
<td>Description</td>
<td>While in the PMR, the student has permission to access a real-time chat room to communicate online with other group members.</td>
</tr>
</tbody>
</table>
References

- Stanford University’s Course Work - http://getcoursework.stanford.edu/
- Moodle Open Source Course Management System - http://moodle.org/
- UML Class Diagram - http://www.agilemodeling.com/artifacts/classDiagram.htm