

# 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

# Introduction

- 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

# System Use Cases

- 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.

# System Use Cases Cont..

Use Cases	Actors
Login the system	Student
Logout the system	Instructor
Download files	
Upload files	
Post tests/quizzes	
Take test/quizzes	
Access Chat Room	
Create Project Meeting Room (PMR)	
Access Project Meeting Room (PMR)	
Post messages	
Retrieve messages	



# Use Case: Take Tests/Quizzes

Use Case	Take Tests/Quizzes
Actors	Student
Type	Primary
Description	The student takes a multiple choice test/quiz, submits his/her answers, and is given a score back.

## Use Case: Post Messages

Use Case	Post Messages
Actors	Student
Type	Primary
Description	The student posts a message in a forum that can be read by all other members of the class.

## Use Case: Create PMR

Use Case	Create PMR
Actors	Instructor
Type	Primary
Description	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.

## Use Case: Download files while in PMR

Use Case	Download files while in PMR
Actors	Student
Type	Secondary
Description	While in the PMR, the student has permission to access and/or download files that have been uploaded by other members in the group.

## Use Case: Post messages while in PMR

Use Case	Post messages while in PMR
Actors	Student
Type	Secondary
Description	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.

# Logout from the System Actions and Responses

Use Case	Logout from the system
Actors	Users (Students/Instructor)
Purpose	Primary
Overview	Users log out their user accounts

## Logout from the System: Actions and Responses Cont..

### Events

Actor Action	System Response
1. This use case begins when a user hit the button exit in the system.	
	2. Displays a confirmation message to the user. The system stores the current time of day as end time
	3. Logout the user.

## Post Tests/Quizzes

Use Case	Post tests/quizzes
Actors	Instructor
Purpose	To upload Tests/Quizzes online from Instructor account
Overview	The instructor uses the given interface to post questions, multiple answer choices, and a solution key to practice tests/quizzes.



## Post Tests/Quizzes Cont..

### Events

Actor Action	System Response
1. This begins when the instructor upload the tests/quizzes file to the test area of the system. Instructor can also set the timer.	
	2. The system transfers the files to testarea, and wait for a signal from student site to take the test.

# 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)

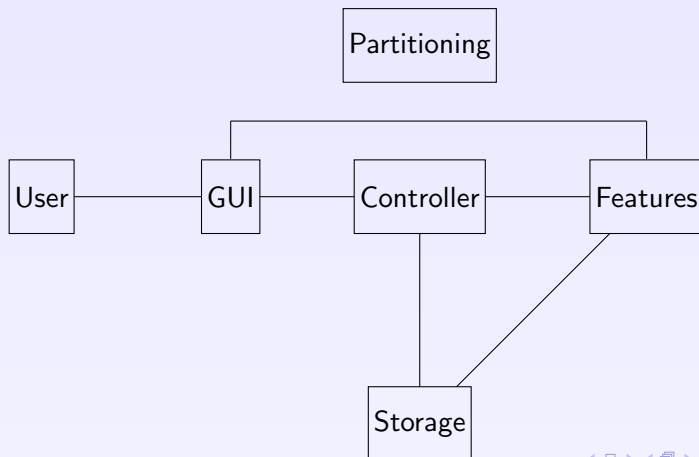
## Input

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

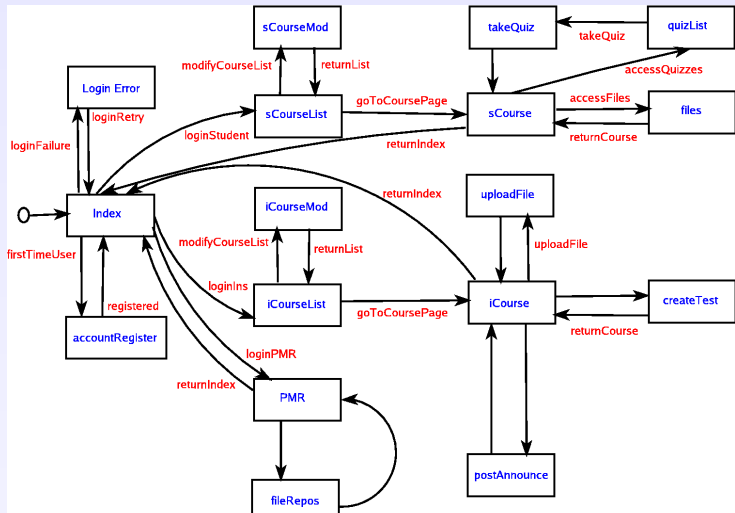
## Expected Output

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.

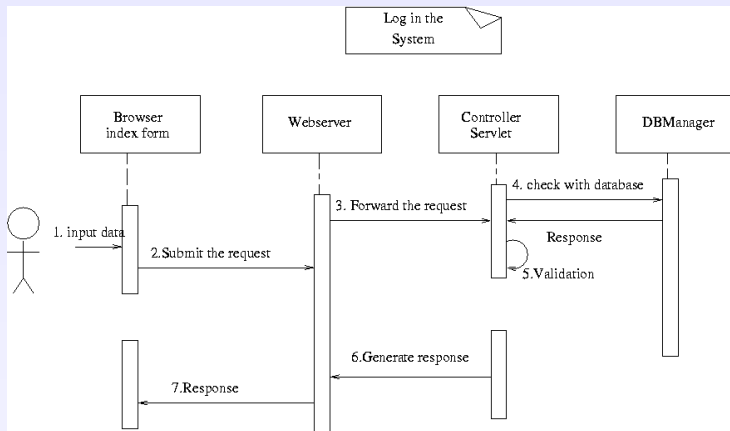
# Partitioning



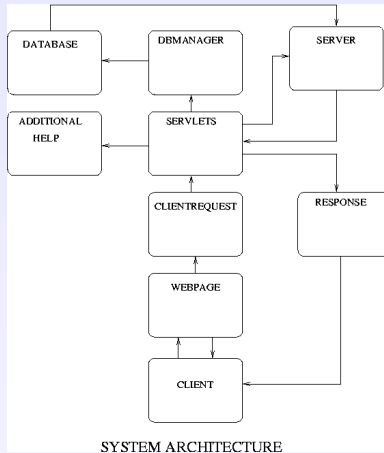
# Flow Chart



# System Sequence Diagram



# System Architecture





# 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(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(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(quizld, 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(quizld, courseld, 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)

# 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)
- 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)
- 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(courseId, pmrId, pmrName)
- PMRGroup(userId, pmrId, lastVisit)
- PMRFile(pmrId, userId, version, commitTime, filename, filepath, commitDate)

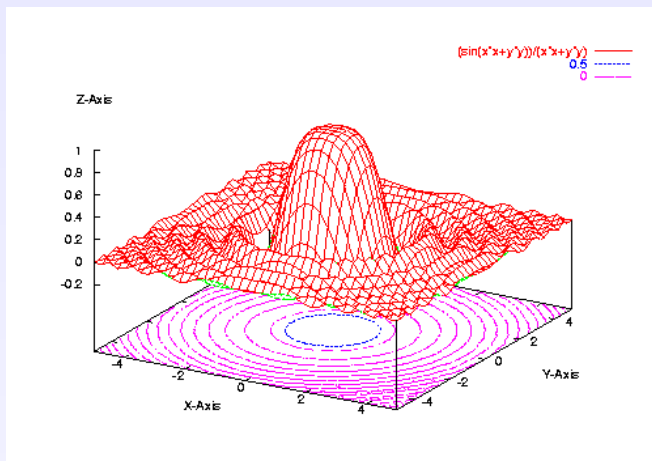
# Technologies Used

- Java.
- Java Servlet.
- HTML
- Oracle 8i.

# Concluding Remarks

- 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



## Use Case: Login to the system

<b>Use Case</b>	<b>Login to the system</b>
Actors	User (Student/Instructor)
Type	Primary
Description	Users login to the system with their passwords and usernames



## Use Case: Logout from the system

<b>Use Case</b>	<b>Logout from the system</b>
Actors	User (Student/Instructor)
Type	Primary
Description	Users logout their user accounts.

# Use Case: Upload Files

Use Case	Upload files
Actors	Instructor
Type	Primary
Description	The instructor uploads any course-related materials that can be downloaded by either the students in the class.

## Use Case: Download Files

Use Case	Download files
Actors	Instructor
Type	Primary
Description	The instructor has access to view and/or download files that have been uploaded by the students in the class.

## Use Case: Download Files

Use Case	Download files
Actors	Student
Type	Primary
Description	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.

## Use Case: Post Messages, Instructor

<b>Use Case</b>	<b>Post messages</b>
Actors	Instructor
Type	Primary
Description	The instructor posts a message in a forum that can be read by all other members of the class.

## Use Case: Retrieve Messages, Student

<b>Use Case</b>	<b>Retrieve messages</b>
Actors	Student
Type	Primary
Description	The student retrieves/views any messages that has been posted in the forum.

# Use Case: Retrieve Messages, Instructor

<b>Use Case</b>	<b>Retrieve messages</b>
Actors	Instructor
Type	Primary
Description	instructor retrieves/views any messages that has been posted in the forum.

## Use Case: Post Tests/Quizzes

Use Case	Post tests/quizzes
Actors	Instructor
Type	Primary
Description	The instructor uses the given interface to post questions, multiple answer choices, and a solution key to practice tests/quizzes.



## Use Case: Access PMR

Use Case	Access PMR
Actors	Student
Type	Primary
Description	The student enters the group and has access to files and messages that are only accessible exclusively by other members in the same group.

## Use Case: Access chat room while in PMR

Use Case	Access chat room while in PMR
Actors	Student
Type	Secondary
Description	While in the PMR, the student has permission to access a real-time chat room to communicate online with other group members.

# References

- Stanford University's Course Work -  
<http://getcoursework.stanford.edu/>
- Moodle Open Source Course Management System -  
<http://moodle.org/>
- PGF User Guide -  
<http://www.ctan.org/tex-archive/graphics/pgf/>
- UML Use Case Diagram -  
<http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>
- UML Class Diagram -  
<http://www.agilemodeling.com/artifacts/classDiagram.htm>