

System Requirements Specification for:  
DBIS Course Project SRS  
ACADSTODAY  
Guide: S. Sudarshan

By:  
Ashish Tajane (09005027)  
Gaurav Choudhary (09005026)  
Saurabh Bhola (09005022)  
Shikhar Paliwal (09005029)

October 1, 2011

# Chapter 1

## Introduction

### 1.1 AcadsToday: Overall Description

AcadsToday is a web interface to help students get more familiar to the courses they are taking during thier curriculum. It can be used in any general institute where there are courses and instructors. Every student will have his own login which is his identity during all the interactions with the website. All the data will be stored and maintained in a common database on the server.

It can help students in numerous ways. Every general course will have rating and reviews on it so that one can choose between two based upon his/her interests. One may contribute by uploading course related material onto the website and help the current and future takers of the course. Uploads include daily lecture notes, ebooks, research papers e.t.c. related to a particular course. Every upload will have its rating too to benefit good materials over others. Now coming to the instructor, it has similar features related to it too like comments and rating. All the above data will be will be for long term purposes i.e. guiding purposes.

Coming to the current updates about the course and instructor, it has a news section wherein one can post any academic or student related news, which will then be displayed on the walls of other students(users). Key implementation is the tagging option with the news post, for e.g. we will have choice of tagging any number of department, course and instructor depending upon to whom is the news more relavant to. It would also help in news filtering. By default it displays every institute news on the user wall(latest first), but one may choose to filter by department or course or instructor depinding upon his/her concerns. These are the short term purpose features of the project.

Apart from filtering news we are implementing search on user, course, instructor and course material which would display the relavant links (regarding the search keyword) to information and reading materials uploaded by others, in order of their rating (most rated first). This would help the user get to useful information very fast.

Finally, to avoid spamming we include institute LDAP authentication during registration in Acadstoday.

### 1.2 Definitions

1. **Department:** Division of a university devoted to a particular academic discipline.
2. **Course:** It is the study of a particular topic within a wider subject area. A typical course has lectures and exams and is offered by a particular department.
3. **Instructor:** A person who belongs to a specific department and teaches one or more courses.

4. **Pre-Requisite Course:** A course X is a pre-requisite of a course Y if one has to pass the course X before taking course Y.
5. **User:** A student registered in the university pursuing a degree in some department with valid Login ID.
6. **Upload:** Any kind of study material that is uploaded by the user for sharing with other users. This may include ebooks, lectures as ppt/pdf/jpg/doc files etc.
7. **Project:** A supplementary, long-term educational assignment necessitating personal initiative, undertaken by an individual student or a group of students under an Instructor.
8. **Project Guide:** Supervisor of a particular project.
9. **Comments:** Remarks made by a User on the shared information/course/instructor for others to read.
10. **Rating:** A position assigned on a scale based on comparative assessment. This is done for a Course or Instructor.
11. **News:** Information about the latest updates in the institute and in particular, the courses a student is taking in current semester.

## 1.3 References

Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. *Database Systems and Concepts*. McGraw Hill, 1996

## 1.4 Overview of Developer's Responsibility

The following responsibilities come under developer's purview:

1. The software shall be developed entirely by us. Anyone who wishes to add further modules to the code or to tailor it for their own needs is welcome to do so.
2. The developer is responsible for the development of User Interface which will use an underlying database structure, also to be developed by the developer.
3. The developer is also responsible for training the users to use the interface, although this would not be necessary considering the simplicity of the interface.
4. The developer will provide some sample uploads and news etc. but the eventual database will be built by the User.

## Chapter 2

# General Description

### 2.1 Product Perspective

AcadsToday is web-based interface designed to facilitate academics among students(user). Its common database will be pre-build and will rarely undergo changes but the rest of the data will constantly grow as users interacts with it. Other than this no existing data is used.

### 2.2 Product Functions Overview

1. User Sign-in and Sign-up (Password verification and confirmation)
2. Updating his/her profile (eg. taking a course, mainting takes relation)
3. News Posting
4. Average Rating Calculation
5. Uploading and link generation
6. Project Registration
7. Commenting
8. Following (notification based on entities being followed)

### 2.3 User Characteristics

As previously mentioned users here are students of the institute. As in no special training is required for a user in today's world. Also user will to keep updating his/her profile to keep the website up-to-date while his/her stay in the institute.

### 2.4 General Constraints

Course Project.

No cost involved.

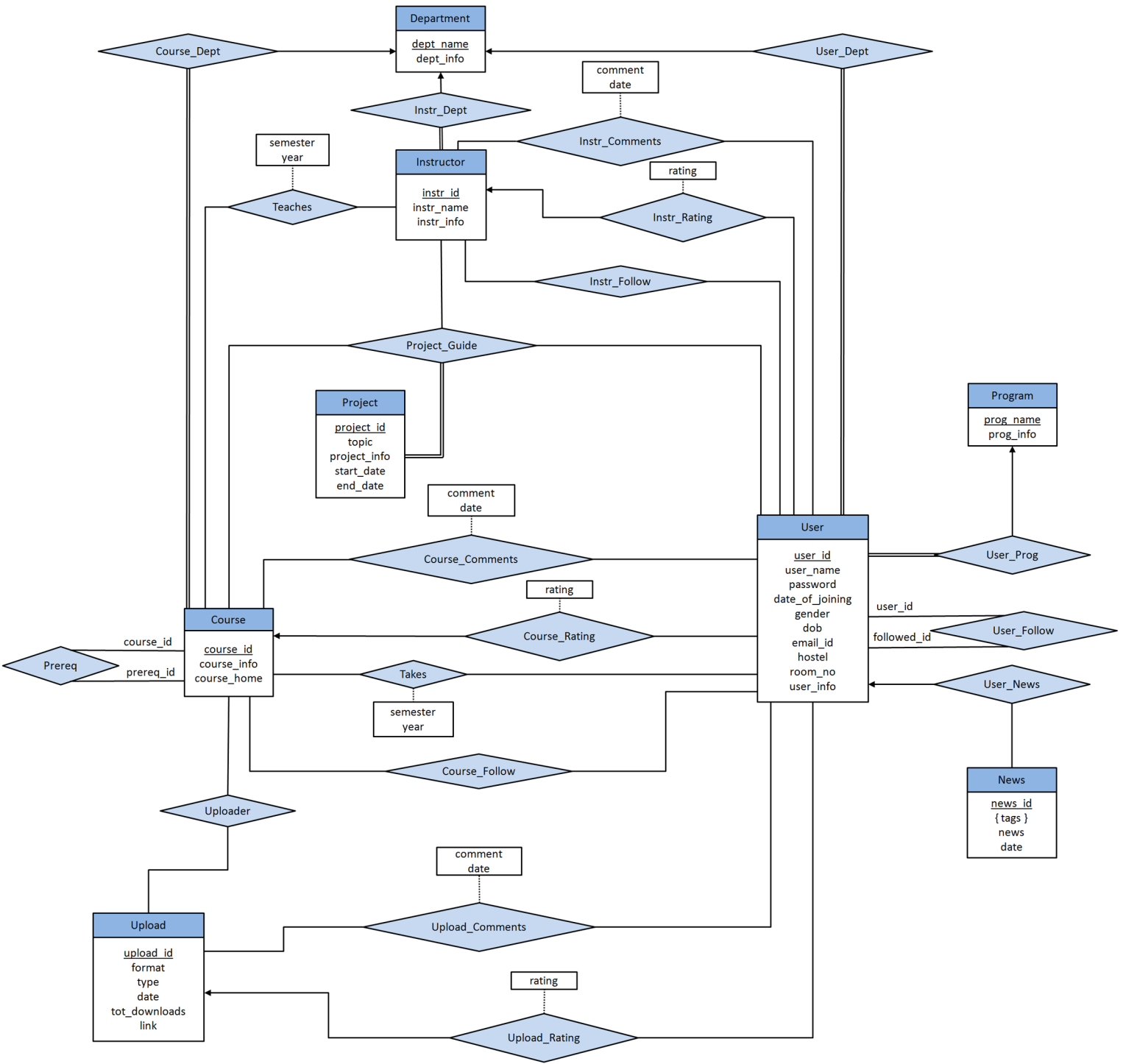
This project is to be completed , including coding, testing and loading of the database, by November 15th 2011. A prototype demo is due in mid-October 2011.

## Chapter 3

# Information Description

### 3.1 Entities and Relationships

1. User
2. Course
3. Instructor
4. Department
5. Project
6. Upload
7. News
8. Program
9. User\_Dept [Relation without a table]
10. Course\_Dept [Relation without a table]
11. Instr\_Dept [Relation without a table]
12. Prereq
13. Takes
14. Teaches
15. Project\_Guide
16. User\_Prog [Relation without a table]
17. User\_News [Relation without a table]
18. User\_Follow
19. Instr\_Follow
20. Course\_Follow
21. Instr\_Comments
22. Course\_Comments
23. Upload\_Comments
24. Inst\_Rating
25. Course\_Rating
26. Upload\_Rating
27. Uploader



E-R Diagram

## 3.2 Data Dictionary

### 3.2.1 A common database

Following is the list all the basic tables and their description which makes the default data for AcadsToday. Changes to this data is very rare.

#### Department

List of all departments in the institute.

dept_name	varchar(50)	primary key	Name of the department
dept_info	text		Relevant information about the department

\*primary key(dept\_name)

#### Course

List of all courses in the institute.

course_id	varchar(10)	primary key	Id of the course
course_name	varchar(25)	not null	Name of the course
course_info	text		Description of the course
dept_name	varchar(50)	foreign-ref (Department)	Name of the department of the course

\*primary key(course\_id)

#### Instructor

List of all instructors of all departments.

inst_id	char(5)	primary key	Id of the instructor
inst_name	varchar(25)	not null	Name of the instructor
inst_info	text		Relevant information about the instructor
dept_name	varchar(50)	foreign-ref (Department)	Name of the department of the instructor

\*\*primary key(inst\_id)

#### Teaches

Relation between Instructor and Course describing all instructor-course pair wherein instructor teaches one , none or more courses.

inst_id	char(5)	foreign-ref (Instructor)	Id of the instructor
course_id	varchar(10)	foreign-ref (Course)	Id of the course taught by the instructor
semester	varchar(10)	Spring/Autumn	Semester in which the course was taught by the instructor
year	int(4)	not null	year in which the course was taught by the instructor

\*\*primary key(inst\_id, course\_id)

### Prereq

Relation between two courses where second course is the pre-requisite of the first course.

course_id	varchar(10)	foreign-ref(Course)	Id of the course
prereq_id	varchar(10)	foreign-ref(Course)	Pre-requisite of the course

**\*\*primary key(course\_id, prereq\_id)**

### Program

Degree Programs in the institue depending upon what it offers.

Prog_name	varchar(10)	foreign-ref(Course)	Id of the course
Prog_info	varchar(10)	foreign-ref(Course)	Pre-requisite of the course

**\*\*primary key(Prog\_name)**

## 3.2.2 User Description

Following is the list all the user dependent tables. They grow as user interacts with the web interface of AcadsToday.

### User

This table contains detail about the registered users. To ensure authenticity of a user, while registration, LDAP Id & password will be checked. Unique user\_id will be provided to all the users. Hence user\_id will be the primary key of the table. Dept\_name, gender, Dob, email\_id, Hostel, room\_no, user\_info etc will store the basic information of any user.

user_id	varchar(15)	primary key	Id of the user
user_name	varchar(20)	not null	Name of the user
password	varchar(15)	min 6 chars	Password for the user account
dept_name	varchar(50)	foreign-ref (Department)	Name of the department of the user
gender	varchar(8)	Male/Female	Gender of the user
dob	char(10)	valid	Date of birth
user_info	text		Additional information about the user

**\*\*primary key(user\_id)**

### Takes

This relates user to all the courses he/she is registered into till today.

user_id	varchar(15)	foreign-ref (User)	Id of the user
course_id	varchar(10)	foreign-ref (Course)	Id of the course undertaken by the user
semester	varchar(10)	Spring/Autumn	Semester in which the course was taken by the user
year	int(4)	not null	year in which the course was taken by the user

**\*\*primary key(user\_id, course\_id)**



## Upload

This relation contains information about any uploaded material. A unique upload\_id has been assigned to each material.

upload_id	varchar(15)	primary key	Id of the upload
format	varchar(10)	ppt/pdf/doc/jpg/others	Format of the file uploaded
type	varchar(10)	notes/lectures/ebook/exam-paper/research-paper/assignment/others	Purpose of the file uploaded
date	char(19)	valid timestamp	Timestamp of upload
tot_downloads	int(10)		Total downloads yet
link	text	not null	link to uploaded file on the server

**\*\*primary key(upload\_id)**

## Uploader

User who has uploaded and the course for which it is associated with is present in this relation.

upload_id	varchar(15)	primary key	Id of the upload
user_id	varchar(15)	foreign-ref (User)	Id of the user who has uploaded
course_id	varchar(10)	foreign-ref (Course)	Id of the course related to the upload

## Project

Details of all the ongoing/completed projects are stored in this relation.

project_id	char(5)	primary key	Id of the project
topic	varchar(100)	not null	Topic of the project
project_info	text		Description of the project
start_date	char(10)	valid start date of the project	Date of commencement of the project
end_date	char(10)	valid end date [can be null in case project is ongoing]	End date of the project

**\*\*primary key(project\_id)**

## Project\_Guide

The relates project with the user, instructor and course it is associated with.

project_id	char(5)	primary key	Id of the project
user_id	varchar(15)	foreign-ref (User)	Id of the user who has done/taken the project
inst_id	char(5)	foreign-ref (Instructor) [can be null]	Id of the instructor under which project has been taken
course_id	varchar(10)	foreign-ref (Course) [can be null]	Id of the course to which project is related to

**\*\*primary key(project\_id, user\_id, inst\_id)**

## Upload\_Comments

There is a feature to comment on the uploaded material. Details of any comment received is stored in this table.

upload_id	varchar(15)	foreign-ref (Upload)	Id of the upload
user_id	varchar(15)	foreign-ref (User)	Id of the user who has commented on the upload
comment	text	not null	Comment
date	char(19)	valid timestamp	Timestamp of the comment

### Upload\_Rating

To distinguish between various uploaded material related to same topic, feature of rating the material has been included.

upload_id	varchar(15)	foreign-ref (Upload) + unique	Id of the upload
user_id	varchar(15)	foreign-ref (User)	Id of the user who has rated the upload
rating	int	[1,5]	Rating

**\*\*primary key(upload\_id, user\_id)**

### Course\_Comments

Similarly commenting can be done on the courses as well

course_id	varchar(10)	foreign-ref (Course)	Id of the course
user_id	varchar(15)	foreign-ref (User)	Id of the user who has commented on the course
comment	text	not null	Comment
date	char(19)	valid timestamp	Timestamp of the comment

### Course\_Rating

Course rating can help any user to decide which course to take among list of courses. Rating can be done on the scale of 1 to 5 by any user who has completed that course.

course_id	varchar(10)	foreign-ref (Course) + unique	Id of the course
user_id	varchar(15)	foreign-ref (User)	Id of the user who has rated the course
rating	int	[1,5]	Rating

**\*\*primary key(course\_id, user\_id)**

### Instr\_Comments

This relation contains info of comment received by any instructor.

inst_id	char(5)	foreign-ref (Instructor)	Id of the instructor
user_id	varchar(15)	foreign-ref (User)	Id of the user who has commented on the instructor
comment	text	not null	Comment
date	char(19)	valid timestamp	Timestamp of the comment

### Instr\_Rating

Similarly Instructor can be rated as well.

inst_id	char(5)	foreign-ref (Instructor) + unique	Id of the instructor
user_id	varchar(15)	foreign-ref (User)	Id of the user who has rated the instructor
rating	int	[1,5]	Rating

**\*\*primary key(inst\_id, user\_id)**

### 3.2.3 Inter-User Features

As the name suggests, this part is to implement the features involving inter-user interaction regarding information shared on the pages. For e.g., while downloading file X, the user will be informed that the users who uploaded file X also uploaded file Y, which may be useful for to the user, as he/she would not have to make a search.

At present, we plan to implement this in a trivla manner, say if for a file X download, we would find the uploads made by the user who uploaded the file X, and display their list. We can also display Most Downloaded File, Most Rated Course(in terms of number) etc. for inter-user interaction.

There is a lot more scope for the development and enhancement of this part of the project which we would try to do.

### 3.2.4 Academic News

This part is to inform the students about the latest updates in the institute, and in particular, in the courses that they attend in the current semester. The key implementation of this part of the project is the Tagging Option associated with it. For e.g., Course Instructor X of a course with course\_id HS-3ZZ announces in the class that there would be a quiz in the next class. Now the user who uploads this news will have an option of tagging the news with course\_id, inst\_name, etc. (This feature of tagging may be extended to the reader of the news too).

By default, every news posted will be displayed on every users wall (the latest being on the top), but the user may filter the news on the basis of the tags, like course-id or inst\_name.

Thus the news will be implemented using the News Table.

#### News

News table which species the publisher and news along with the date & tags of the news.

news_id	char(5)	foreign-ref (User)	Id of the user who has rated the instructor
user_id	varchar(15)	foreign-ref (User)	Id of the user who has rated the instructor
news	text	not null	News by the user
tags	text		Multivalued attrib containing comma seperated values of the tags; to be used while filtering
date	char(19)	valid timestamp	Timestamp of the comment

**\*\*primary key(news\_id)**

Also we plan to consider the News upload databse for the current semester only. After each semester, the databse can be deleted for the news in the next semester.

Apart from the filtering on the basis of Tags, the filtering can also be done according to the month or semester . This will help the user to keep track of the semester news.

### 3.2.5 Follow

A user can follow any course, instructor or any other user. Any updates from these entities will only be displayed on the user news wall.

### User\_Follow

A user following another user.

user_id	varchar(15)	foreign-ref (User)	Id of the user
followed_id	varchar(15)	foreign-ref (User)	Id of the user being followed

**\*\*primary key(user\_id, followed\_id)**

### Instr\_Follow

A user following an instructor.

user_id	varchar(15)	foreign-ref (User)	Id of the user
inst_id	char(5)	foreign-ref (Instructor)	Id of the instructor being followed

**\*\*primary key(user\_id, inst\_id)**

### Course\_Follow

A user following a particular course.

user_id	varchar(15)	foreign-ref (User)	Id of the user
course_id	varchar(10)	foreign-ref (Course)	Id of the course being followed

**\*\*primary key(user\_id, inst\_id)**

## 3.2.6 Search

How tedious it would be to go into depths searching for a file that has been uploaded or suppose the ebooks of a course, or to find who is the course instructor for a particular course, or what ratings has a course/file/instructor earned. For easing the job, we plan to implement the Search option in our project. This will be similar to the Query Search made on the database. Just enter your query and the relevant search table you want and your result will be displayed.

For e.g.,

(1) I want to know if the ebook of Database System Concept by S. Sudarshan is available. I would make a query Search by entering Database System Concepts by S. Sudarshan and search in Upload\_id.

(2) I want to know if the midsem paper of HS-301 is available, again I will make a query search by entering "midsem.Hs-301" and search in the Upload\_id.

The Uploads etc. should be named in a certain manner only for the search result to be correct.

## 3.3 Data Flow

Basically our whole project can be divided into the following sections, each of them having specific functions of their own:

1) **Database Section:** It consists the entire database in form of the tables as described by the ER diagram. Parts of it are entitled to be updated by the authentic Users.

2) **Authorization Section:** It is responsible for authenticating the users who can access the database and update it or use it for some purpose.

3) **Searching Section:** This is responsible for providing results to the Query Search made by a user.

## Chapter 4

# Functional Requirements

### 4.1 Sign In and Sign Up

The first screen for all users (students) is the Sign In screen. If a user has not already registered, he/she can call up the Sign Up screen. New users can thus sign up and proceed to the sign in screen. While signing up, users will fill in their personal information like name, date of birth, gender etc. and academic information like program and department of study etc. and residential information like hostel and room no.

### 4.2 Taking a Course

A user can take a course from already floating courses from that particular semester only and only if he/ she has completed all prerequisites the course has if any.

### 4.3 News Posting

A user can post a news for other users to read with multiple tags so that the search of news regarding a particular topic is completed efficiently and news content is managed in a better way.

### 4.4 Upload

A user can upload notes or other study / reference material like e-books for a particular course. Other users can download the material.

### 4.5 Project Registration

A user can take a project from available projects. A project may or may not be associated with a course and it may or may not have an instructor as a guide. So, a user can simply take a project out of the curriculum without any guidance of any instructor out of sheer interest in the subject.

### 4.6 Rating

A user can rate a course, an instructor or an uploaded material only once.

## **4.7 Comments**

A user can comment on a course, an instructor or an uploaded material many times. Nested comments feature is not supported. Also, a user may not tag other users in comments.

## **4.8 Following**

A user can follow an instructor, a course or another user :

### **4.8.1 Instructor**

The user following a particular instructor will get notifications if any other user comment on that instructor.

### **4.8.2 Course**

The user following a particular course will get notifications if any other user comment on that course or upload any material for that course.

### **4.8.3 User**

The user following another user will get notifications if the other user comment on any instructor, course, uploaded material or he/she upload a material for a course.

## Chapter 5

# External Interface Requirements

### 5.1 User Interfaces

The web (WWW) is used as the interface.

### 5.2 Hardware Interfaces

No special hardware interface.

### 5.3 Software Interfaces

MySQL 5.5.16

PHP 5.3.3

Apache Tomcat 7.0

## Chapter 6

# Performance Requirements

### 6.1 Capacity Requirements

Considering that this software is being used in any institute like IIT Bombay, here are the average figures:

Type	No.	Multiplicity	Total	Remarks
User	5000		5000	
Instructor	200	5	100	5 Courses/Instructor
Courses	500	60	3,00,000	10 semester/user and 8 courses/semester
PreReq	500	2	1000	2 prereq/course
Project	500	2	1000	500 project and 2 user/project
Comments	5000	50	2,50,000	5000 user and 50 comments/user
Rating	5000	60	3,00,000	5000 user and 60 rating/user
Upload	5000	10	50,000	10 upload/user
News	5000	5	25,000	5 news/user

Total Tuple = 10L.

Each tuple = 5 Attribute(avg.) = 250 Byte (Assuming 50Byte/attr.)

Total Size = 2500L Byte.= 300 MB.

### 6.2 Throughput

Assuming each User log in once daily, transactions are being done for 5 users each minute.

For 4 transactions per user

Number of transactions per min = 20



## Chapter 7

# Design Constraints

### 7.1 Standards Compliance

Not applicable.

### 7.2 Hardware Limitations

Number of Users = 5000

Uploads per User = 10

Total upload = 50,000

Size of each upload = 2 MB

Hence we require a space of about 100 GB to store the uploaded material.

## Chapter 8

# Validation Criteria

We will upload some sample Upload materials and have sufficient database for testing the following cases:

1. A User (with valid Login Id as already defined) can search for a document within the documents uploaded already.
2. A User is able to upload a document for sharing with other users. Test if the document gets uploaded and the database is updated with the necessary details associated with the concerned document.
3. A User is able to search for Courses/instructors etc.
4. A User is able to see the ratings of a Course/Instructor/Upload as well as Comments by other Users. Also, the User must be able to comment and/or rate the Course/Instructor/Upload.
5. A User is able to upload a news. Test if this news appears on the "walls" of the relevant Users (relevance to a news may vary).