

System Requirements Specification for: DBIS Course Project SRS

Add name here

November 29, 2011

Contents

1	Introduction	2
1.1	Overall Description	2
1.2	Definitions	2
1.3	References	2
1.4	Overview of Developer's Responsibilities	3
2	General Description	4
2.1	Product Perspective	4
2.2	Product Functions Overview	4
2.3	User Characteristics	4
2.4	General Constraints	5
3	Information Description	6
3.1	Entities and Relationships	6
3.1.1	Entities	6
3.1.2	Relations	6
3.2	Data Dictionary	7
3.2.1	Student	7
3.2.2	Instructor	7
3.2.3	Course	8
3.2.4	Prerequisite	8
3.2.5	Offering	8
3.2.6	Classrep	9
3.2.7	Timeslot	9
3.2.8	Project	9
3.2.9	Takes	9
3.2.10	Teaches	10
3.2.11	Takes_Project	10
3.2.12	Feedback	10
3.2.13	Work_Profile	11
3.2.14	Work_Department	11
3.2.15	Takes_Work	11
3.2.16	Review	11
3.2.17	Post	12
3.3	Data Flow	12
4	Functional Requirements	17
4.1	Interfaces	17
4.1.1	Login and Register	17
4.1.2	Home Screen	17
4.1.3	Comprehensive Search	17
4.1.4	Compare Courses	17
4.1.5	Profiles	17
4.1.6	Review	17
4.1.7	Discussion Thread	18

4.1.8	Feedback	18
4.1.9	Weekly Schedule	18
4.2	Internal Functions	18
4.2.1	Storage	18
4.2.2	Feedback Normalisation	18
4.2.3	Rating Computation	18
4.2.4	Work Profiles	19
5	External Interface Requirements	20
5.1	User Interfaces	20
5.2	Hardware Interfaces	20
5.3	Software Interfaces	20
6	Performance Requirements	21
6.1	Estimated number of users	21
6.2	Estimated database size	21
7	Design Constraints and Validation Criteria	22
7.1	Standards Compliance	22
7.2	Hardware Limitations	22
7.3	Validation Criteria	22
7.4	Other Requirements	22

Chapter 1

Introduction

1.1 Overall Description

IIT Bombay offers hundreds of courses across several departments covering a diverse array of sub-disciplines and flavors. This supports a highly flexible curriculum, and allows the student to study exactly what he intends to. Most universities across the world have adopted this ideology of letting the student choose his curriculum.

This, however, also brings with it the dilemma of choosing between several options, each having its own positives and negatives. This decision becomes difficult because the students are usually not well informed about the importance and relevance of different courses.

Our project aims to help students with their decision making when choosing their courses. We provide the student with a comprehensive comparison among different courses based on how each course would impact the student's profile. This is achieved by taking into consideration several parameters related to courses, students, instructors, job profiles and course projects.

1.2 Definitions

1. Offering : An instance of a course run in a particular year and semester. There can be multiple offerings of the same course across different semesters, or even in the same semester eg. Different batches of the same course.
2. Work Profile : A work opportunity offered to students by an organisation eg. Internships.
3. Review : A review for a course written by some student, for the reference of other students.
4. Post : A brief comment by a student sharing some information with those taking a particular offering.
5. Time Slot : A logical entity in the institute's weekly schedule. Each offering runs in a particular time slot, which decides the schedule of its classes.
6. CR, DR, IR : Course Rating, Difficulty Rating, Instructor Rating for an offering of a course. These constitute the feedback given by students, and are used to compare between different courses.

1.3 References

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan. Database System Concepts. McGrawHill, 1996
2. <http://www.courserank.com> - A similar project undertaken by students at Stanford University.

1.4 Overview of Developer's Responsibilities

- Development: The developer is responsible for implementation of the database structure prescribed in this document, as well as the interface for the users of the application, also as stated later in the document.
- Training: The interface is easy to use for most purposes. The administrator needs to be trained to periodically update the database and add data.
- Maintenance: The administrator needs to add new data as and when required.

Chapter 2

General Description

2.1 Product Perspective

The application is related to the institute's existing database of academic information including courses, students, instructors and the like.

It has the following interfaces:

1. Search Interface: To search for relevant courses based on an array of parameters.
2. Comparison Interface: To compare different courses head-to-head on different factors.
3. Feedback Interface: To post a brief feedback about an offering and its instructor, which is used for rating the courses and instructors.
4. Discussion Interface: To post a comment on the discussion thread for an offering, which is visible on the offering's page or on the student's homepage.
5. Profile: To display and edit information about the courses, internships and other details of the student, also for instructors and courses.

2.2 Product Functions Overview

1. Login to the student's profile.
2. Search for relevant courses and offerings.
3. Rate courses and instructors on student feedback, relevance and importance.
4. Display information about courses and instructors in a concise manner.
5. Compare courses with each other.
6. Collect feedback for an offering and its instructor from students.
7. Display information about past projects done in a course.
8. Provide a discussion thread for each offering, where students can post updates and share information.
9. Allow users to write detailed reviews for courses.

2.3 User Characteristics

Our application has 2 types of users:

1. Students: These are the targeted users of the application. All features are designed for easy use by the students, and hence no prior technical training is required.

2. Administrator: Responsible for keeping the database up-to-date with the latest data. Must be proficient with the database and schema design, and must have access to the original data in the institute's database. Knowledge of scripting, though not an absolute necessity, is recommended as it will aid the administrator in adding new information.

2.4 General Constraints

The project is to be completed, including coding, testing and loading of the database, by the end of November 2011. A prototype demo is scheduled for mid October 2011.

Chapter 3

Information Description

3.1 Entities and Relationships

See figures 3.1, 3.2, 3.3 and 3.4 for ER diagrams (at the end of the chapter).

3.1.1 Entities

- Student
- Instructor
- Course
- Offering
- Work_profile
- Time_slot
- Project
- Feedback
- Review
- Posts

3.1.2 Relations

- project_guide
- project_offering
- takes
- takes_project
- takes_work
- classrep
- teaches
- prereq
- course_offering
- schedule
- course_review
- post_relation
- feedback_relation

3.2 Data Dictionary

The Application has the following tables in the schema, which are subsequently defined in detail:

- Student
- Instructor
- Course
- Prerequisite
- Offering
- Classrep
- Timeslot
- Project
- Takes
- Teaches
- Takes_Project
- Feedback
- Work_Profile
- Work_Department
- Takes_Work
- Review
- Post

3.2.1 Student

Attribute	Data Type	Constraints	Description
student_id	varchar(30)	primary key	Student ID
name	varchar(30)	not null	Name
department	varchar(30)	not null	Department
photo	blob		Display Picture
cpi	decimal(4,2)	between 0 and 10	CPI
password	varchar(50)		encrypted user password
cr_mean	decimal(6,4)	between 0 and 5	Mean Course Rating
cr_var	decimal(6,4)	between 0 and 25	Variance of Course Rating
ir_mean	decimal(6,4)	between 0 and 5	Mean Instructor Rating
ir_var	decimal(6,4)	between 0 and 25	Variance of Instructor Rating
dr_mean	decimal(6,4)	between 0 and 5	Mean Difficulty Rating
dr_var	decimal(6,4)	between 0 and 25	Variance of Difficulty Rating
FULLTEXT (student_id, name, department)			

The Student relation stores basic information and some statistics about students. The Student ID is used to identify the student across the schema.

3.2.2 Instructor

Attribute	Data Type	Constraints	Description
instructor_id	varchar(12)	primary key	Instructor ID
name	varchar(30)	not null	Name
department	varchar(30)	not null	Department
rating	decimal(3, 2)		Instructor rating
FULLTEXT (instructor_id, name, department)			

The Instructor relation stores basic information about instructors. The Instructor ID is used to identify the Instructor across the schema.

3.2.3 Course

Attribute	Data Type	Constraints	Description
course_id	varchar(8)	primary key	Course ID
name	varchar(50)	not null	Name
department	varchar(30)	not null	Department
description	varchar(1500)		Course Description
	FULLTEXT (course_id, name, department, description)		

The Course relation stores information about the various courses offered by different departments of the institute. This is different from individual offerings of the course in particular semesters.

3.2.4 Prerequisite

Attribute	Data Type	Constraints	Description
course_id	varchar(8)	not null foreign key references Course	Course ID of target course
prereq_id	varchar(8)	not null foreign key references Course	Course ID of prerequisite course

The Prerequisite relation defines the course hierarchy in terms of prerequisite courses for each course.

3.2.5 Offering

Attribute	Data Type	Constraints	Description
offering_id	int(12)	primary key auto increment	Offering ID
course_id	varchar(8)	not null foreign key references Course	Corresponding Course
year	decimal(4,0)	not null	Year when offered
semester	varchar(6)	not null in {spring, autumn, summer, winter}	Semester when offered
credits	decimal(3,1)	not null	Credit Weightage of offering
venue	varchar(30)		Venue of Lectures/Labs
timeslot_id	decimal(2,0)	not null foreign key references Timeslot	Timeslot Occupied in Schedule
strength	decimal(4,0)		Batch Strength
num_ap	decimal(4,0)		AP grades given
num_aa	decimal(4,0)		AA grades given
num_ab	decimal(4,0)		AB grades given
num_bb	decimal(4,0)		BB grades given
num_bc	decimal(4,0)		BC grades given
num_cc	decimal(4,0)		CC grades given
num_cd	decimal(4,0)		CD grades given
num_dd	decimal(4,0)		DD grades given
num_fr	decimal(4,0)		FR grades given
num_xx	decimal(4,0)		XX grades given

The Offering relation stores information about individual offerings of courses in particular semesters. It also stores grading statistics (which will initially be null) for the offering.

3.2.6 Classrep

Attribute	Data Type	Constraints	Description
student_id	varchar(12)	not null foreign key references Student	Student who is the Class Representative
offering_id	int(12)	not null foreign key references Offering	Offering for which the student is the Class Representative

The Takes_Project relation relates students with the course projects they are undertaking i.e. a student is related to all the projects he/she has done or is doing, and a project is related to all students doing the project.

3.2.7 Timeslot

Attribute	Data Type	Constraints	Description
timeslot_id	decimal(2,0)	not null	Timeslot ID
day	decimal(1,0)	not null between 1 and 7	Day of the week (Monday = 1)
start_time	time	not null	Start Time of the slot
end_time	time	not null	End Time of the slot
primary key (timeslot_id, day, start_time)			

The Takes_Project relation relates students with the course projects they are undertaking i.e. a student is related to all the projects he/she has done or is doing, and a project is related to all students doing the project.

3.2.8 Project

Attribute	Data Type	Constraints	Description
project_id	int(12)	primary key auto increment	Project ID
name	varchar(50)	not null	Project Title
description	varchar(1500)		Project Description
offering_id	int(12)	not null foreign key references Offering	Offering in which project was undertaken
instructor_id	varchar(12)	not null foreign key references Instructor	Instructor under which project was undertaken
(offering_id, instructor_id) in (select * from Teaches)			

The Project relation contains details about course projects taken by students under some instructor in some offering of a course. Each tuple in this relation is connected to the participating students via the Takes_Project relation.

3.2.9 Takes

Attribute	Data Type	Constraints	Description
student_id	varchar(12)	not null foreign key references Student	Student taking the offering
offering_id	int(12)	not null foreign key references Offering	Offering being taken by the student
grade	varchar(3)		Grade obtained by the student in the offering
primary key (student_id, offering_id)			

The Takes relation relates students with the offerings they are enrolled in i.e. a student is related to all the offerings he/she has taken or is taking, and an offering is related to all students enrolled in it.

3.2.10 Teaches

Attribute	Data Type	Constraints	Description
instructor_id	varchar(12)	not null foreign key references Instructor	Instructor teaching in the offering
offering_id	int(12)	not null foreign key references Offering	Offering in which the instructor is teaching

The Teaches relation relates instructors with the offerings they are teaching in i.e. an instructor is related to all the offerings he/she has taught in or is teaching in, and each offering is related to all instructors teaching it.

3.2.11 Takes_Project

Attribute	Data Type	Constraints	Description
student_id	varchar(12)	not null foreign key references Student	Student participating in the project
project_id	int(12)	not null foreign key references Project	Project being participated in by the student
score	decimal(5,2)	between 0 and 100	Score obtained by the student in the project
primary key (student_id, project_id)			

The Takes_Project relation relates students with the course projects they are undertaking i.e. a student is related to all the projects he/she has done or is doing, and a project is related to all students doing the project.

3.2.12 Feedback

Attribute	Data Type	Constraints	Description
student_id	varchar(12)	not null foreign key references Student	Student giving the feedback
offering_id	int(12)	not null foreign key references Offering	Offering for which feedback is being given
instructor_id	varchar(12)	not null foreign key references Instructor	Instructor for which feedback is being given
mytime	timestamp		Time of Submission
course_rating	int(1)	not null between 0 and 5	Course Rating
instructor_rating	int(1)	not null between 0 and 5	Instructor Rating
difficulty_rating	int(1)	not null between 0 and 5	Difficulty Rating
avg_workload	decimal(3,1)	not null	Average Weekly Workload
primary key (student_id, offering_id, instructor_id)			
(offering_id, instructor_id) in (select * from Teaches)			

The Feedback Relation stores all feedback recieved from students. It lies at the core of most of the statistics that the application presents, and is the most crucial relation of the schema.

3.2.13 Work_Profile

Attribute	Data Type	Constraints	Description
work_id	decimal(12,0)	primary key	Work Profile ID
name	varchar(50)	not null	Name of Company/Institution
description	varchar(1500)	not null	Description of the Profile
period	varchar(6)	not null in {summer, winter, spring, autumn}	Period of the Work
length	decimal(2,0)		Length of the Work in weeks
stipend	decimal(7,0)		Monthly Stipend in INR
location	varchar(30)	not null	City, Country of the Work
category	varchar(2)		Category of Work eg. A,B,etc.
FULLTEXT (name, description)			

The Work_Profile relation stores work opportunities including Internships, External Projects, etc. These are associated with students through the Takes_Work relation.

3.2.14 Work_Department

Attribute	Data Type	Constraints	Description
work_id	decimal(12,0)	not null foreign key references Work_Profile	Work Profile allowing the given department
department	varchar(30)	not null	Department eligible for the work profile

The Work_Department relation defines which department students are eligible for a given work profile.

3.2.15 Takes_Work

Attribute	Data Type	Constraints	Description
student_id	varchar(12)	not null foreign key references Student	Student taking the work profile
work_id	decimal(12,0)	not null foreign key references Work_Profile	Work Profile being taken

The Takes_Work relation associates students with the Work Profiles they have worked on.

3.2.16 Review

Attribute	Data Type	Constraints	Description
course_id	varchar(8)	not null foreign key references Course	Course for which review is being written
time	timestamp	not null	Time of Submission of the review
content	varchar(1500)	not null	Content of the Review
author	varchar(30)		Name or Nickname of author of Review
primary key (course_id, time, author)			
FULLTEXT (content)			

The Review relation stores reviews stored by students about different courses. These are displayed along with the corresponding course, and provide a descriptive feedback.

3.2.17 Post

Attribute	Data Type	Constraints	Description
post_id	int(12)	primary key, auto increment	post identifier
parent_post_id	int(12)		container of the post
student_id	varchar(12)	not null foreign key references Student	Student giving the update
offering_id	int(12)	not null foreign key references Offering	Offering for which update is being posted
mytime	timestamp	not null	Time of Submission of Update
content	varchar(500)	not null	Content of the Update
privilege_flag	boolean	not null	Flag whether update is privileged
(student_id, offering_id) in (select student_id, offering_id from takes)			
FULLTEXT (content)			

The Post relation stores updates/comments related to a particular offering being posted by students taking the offering. An update can be privileged, in which case it will be broadcast on the home-screens of all students taking the offering.

3.3 Data Flow

There are an array of features and functionalities (described in more detail in Section 4 on Functional Requirements) provided by the software. However, the major flow of control follows the following request-fetch-process-return order:

1. User selects the feature to be used through the user interface. For example, if the user wants to search for a course by name, the search course module (in the interface) will be used.
2. User enters arguments and other details for the function to be used. For the above example, the user would enter some keywords or a part of the name of the course to be searched for.
3. Once the user submits the request, an appropriate database query is called to return a raw result. For the search function, this will fetch the list of corresponding courses.
4. This raw result is processed by the software, and displayed in a concise and meaningful manner, as is relevant for the particular request. For the above example, this would mean arranging the search results in an order as requested by the user, say by descending order of difficulty.
5. Secondary requests, on these results, can then be made by the user, which will be fetched, processed and returned in a similar fashion. For instance, the user could ask for further detailed information about one of the courses from the search results.

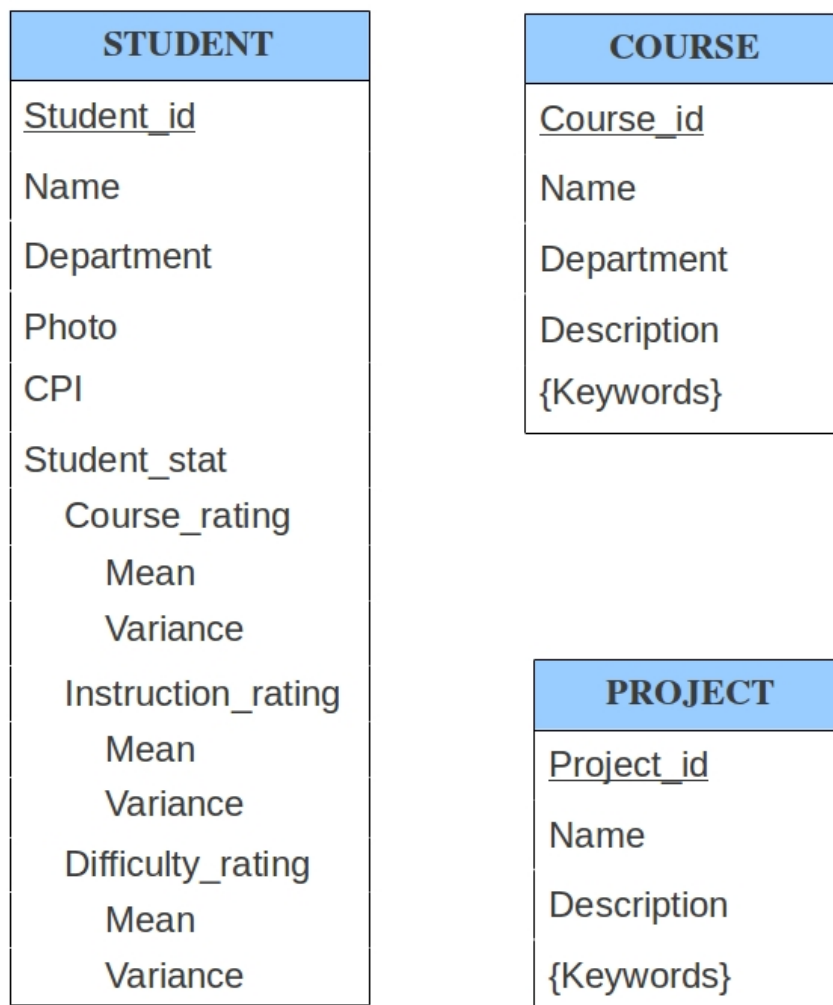


Figure 3.1: Entities and their attributes part 1 : Student, Course, Project.

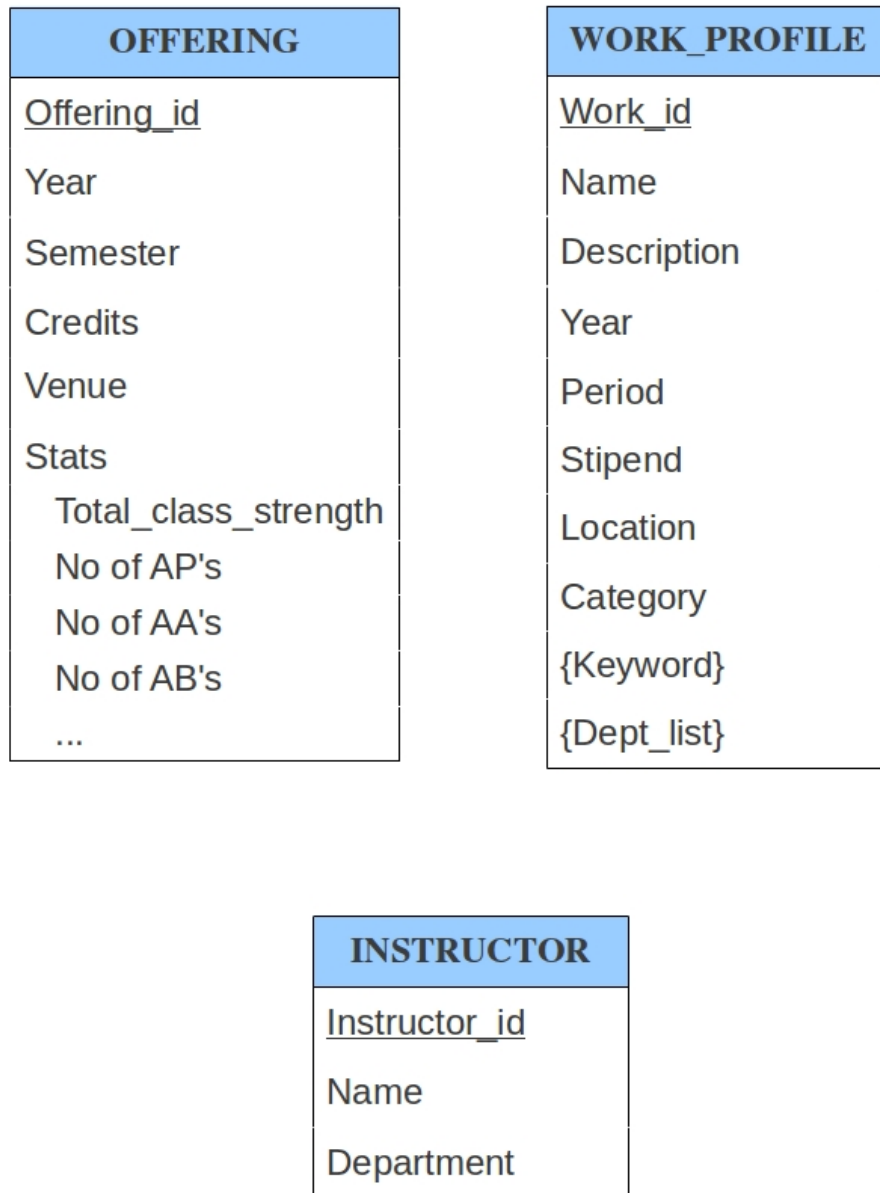


Figure 3.2: Entities and their attributes part 2 : Offering, Work_Profile, Instructor.

REVIEW
<u>Time_</u>
Content
<u>Author_</u>

TIME_SLOT
Time_slot_id
{
Day
Start_time
End_time
}

FEEDBACK
<u>Time_</u>
Course_rating
Instructor_rating
Difficulty_rating
Average_work_load

UPDATES
<u>Time_</u>
Content
Cr_flag

Figure 3.3: Entities and their attributes part 3 : Review, Time_Slot, Feedback, Updates.

Chapter 4

Functional Requirements

4.1 Interfaces

4.1.1 Login and Register

The first page when running the software in logged-out mode. This will provide a Login section for already registered users, and a Registration area for new users willing to use the software. The Registration process will take from the user only the student ID, email ID and password, thus making the registration process very convenient and straight-forward. After a user performs a login, (s)he will be taken to his/her Home Screen (detailed in Section 4.1.2).

4.1.2 Home Screen

The Home Screen is the first page reached by the user after logging in. It is aimed at giving the user easy access to all features of the software. It will display a list of important updates from different courses the user has enrolled in, and links to software features like search, profile, etc.

4.1.3 Comprehensive Search

This interface will be used for searching for entities in the database by its name, department, relevance, rating (based on course content, difficulty, instructors and also weekly workload) and other details. It will also provide options to display the results in a particular sorted order based on these parameters. This will essentially be the main entry-point for the student into the vast pool of courses available for a student to take. It will also link to the detailed pages of those results.

4.1.4 Compare Courses

The Compare Courses interface would be a secondary interface, where user will be able to add courses from the Search Course interface, and then have a head-2-head comparison between different aspects of the courses being compared. This comparison would be displayed in a way that facilitates a direct comparison, and thus helps the student take an informed decision on which course to take.

4.1.5 Profiles

This is a group of similar interfaces, or informative pages. Essentially, each course, offering, instructor, student, project would have a profile page, with detailed information about it. These will be accessible from the results of a search which includes the particular unit whose profile is to be viewed. The purpose of these is to give complete information to the user, and thus facilitate him/her to have a comprehensive idea about the unit. Additionally, the user can edit his/her own profile. He can change the name, password, cpi and other such details.

4.1.6 Review

Reviews are briefly descriptive write-ups about courses. Users will be able to view previous reviews for a particular course, and also write one. These are meant to provide a more subjective peer opinion than what a quantitative result can offer, and thus will be important for a particular class of users.

4.1.7 Discussion Thread

For each course offering, there will be a discussion thread, on which all students enrolled in the offering will be allowed to post comments and updates. This will not only enable dissemination of important notices, but also promote sharing of information. Also, the CR of the offering will have the authority to post privileged updates, which will be displayed directly on the homescreen of all enrolled users, thus useful for posting important and urgent course-related notices.

4.1.8 Feedback

This is another important interface/feature. It will be used by the user to provide feedback for a particular instructor in a particular offering. The feedback would consist of ratings corresponding to Course (Overall), Instructor, Difficulty (Course Content) and Average Weekly Workload. These data, from all students, will form the basis of the way different courses and instructors are rated in comparison with each other. Another feature of this interface is that feedback for a particular offering will be available only for a fixed period of time. Also, the data submitted through this interface will be further processed (refer Sections 4.2.2 and ??) to make it useful and meaningful.

4.1.9 Weekly Schedule

This is a small interface which will show the selected courses arranged and aligned with their respective timeslots. This will enable the student to find a clash-free combination of relevant courses which (s)he intends to take, again helping the student decide on course selection.

4.2 Internal Functions

4.2.1 Storage

The most basic internal feature of the software. This basically means that whenever the user adds or modifies some information, it will be correspondingly added/updated in the database. This will be done automatically, and is different from the manual addition/update of new data done by the Administrator.

4.2.2 Feedback Normalisation

On every feedback submission by a user, the software must perform the following features:

1. Store the raw feedback in the database
2. Update the mean and variance of all feedback given by the particular user
3. Precompute the ratings for instructors, offerings, etc. to reduce overhead at the time of retrieval

Essentially, the software will store the mean and variance of different ratings (course, instructor and difficulty) for each user. This information will be used to normalise the ratings (refer Section 4.2.3) for each student, when computing the overall rating for an instructor or course. This is important because even though the rating feedback is on a scale of 0-5, different users will have different scales of magnitude. For instance, while one user might give a rating of 3 to a normal course and 4 to a good course, another might give a 2 to a normal course and a 5 to a good course. When computing the total rating for the course, the user giving more MORE EXTREME ratings will thus get more weightage, which is undesirable.

4.2.3 Rating Computation

Another important internal function of the software is to compute the ratings for each course and instructor. Whenever a profile page of a course (or instructor) is requested, the software will compute, using all relevant feedback available, an average rating for the course. Importantly, the software will take into account the variation in ratings of different users. Basically, it will use the mean and variance (refer Section 4.2.2) for each type of rating for each user to normalise the feedback given by that user before including it in the computation of the average rating. Finally, this normalised average rating will be provided to the user, thus ensuring an unbiased and neutral comparison between different courses.

4.2.4 Work Profiles

Another interesting and useful feature of the software is that it could potentially take into account the different Work Profiles that are offered to students, and what qualities they seek in potential employees i.e. the software considers the keywords (tags, representing the relevant topics and subjects) associated with the work profile, and matches them to the different courses and projects, thus giving an indication as to how important a course can be to get the student a good work profile. This is one factor which students try to consider when taking a course, but in most cases they are unable to get true information about the importance of a course. This feature attempts to quantify the course's industry importance, helping the student towards building a strong academic portfolio.

Chapter 5

External Interface Requirements

5.1 User Interfaces

The software is used via a website interface.

5.2 Hardware Interfaces

The software does not interface with any special hardware.

5.3 Software Interfaces

- Java 6
- JDBC and servlets
- Oracle Thin JDBC Driver

Chapter 6

Performance Requirements

6.1 Estimated number of users

Considering the number of registered students and the courses run in a semester, we have estimated the number of users in various categories in accordance with our project scale. Note that the data is just an estimate.

User Type	Number of Users	Comments
System Administrator	1	The whole system is maintained by a single system administrators.
Student	2000-2500	These figures are based on the assumption that about 500 students from each batch (1s yr, 2nd yr..) will use our application
Class-rep	600 - 700	There are about 600 - 700 courses run by different departments in a semester. Each course has at least one class-rep.

6.2 Estimated database size

To make the approximate calculations, we make the following assumptions for the space required in database:

- course information : 4kb
- student information : 100kb (including user image)
- other information : 1 kb each for instructor info., internship info. etc.

With the above approximations and the number of various users (as estimated in previous section), the database size can be approximated to 300 MB for one semester.

Chapter 7

Design Constraints and Validation Criteria

7.1 Standards Compliance

Not applicable

7.2 Hardware Limitations

Not applicable

7.3 Validation Criteria

We will load the database with sample data and show the smooth functioning of the various facilities. Correctness checks will be conducted.

7.4 Other Requirements

Not applicable