Software Requirements Specification

for

Management Information System for Blended MOOCs

By

The MOOCs Summer Intern Team, June 2015

Coordinators

Contributors

1. Aditya Kalkonde
2. Amanpreet Kang
3. Anjay Abhishek
4. Ankit Kumar
5. Ankit Raonka
6. Anushka Jejani
7. Apoorva Agarwal
8. Arinjoy Basak
9. Ashwani Pandey
10. Chirram Kumar
11. Dappu Rajesh
12. Deepak Gupta
13. Devesh Kumar
14. Dewang Palav
15. Dhiraj Gandhi
16. Divyanshu Agarwal
17. Dora Dileepkumar
18. Ekta Sharma
19. Gunjan Kulkarni
20. Harshada Kumbhare
21. Kamal Dev Ram
22. Ketaki Kinage
23. Madhupriya Ravishankar
24. Manisha Yadav
25. Meenakshi Kumari
26. Mohsin Mohammad
27. Muhammed Zubairuddin
28. Nihar Yeolekar
29. Nikhita Begani
30. Nitish Deo
31. Parth Patel
32. Prathamesh Dahale
33. Raviprakash Bajpai
34. Sagar Agarwal
35. Samridhi
36. Shantanu Kumar

37.Shashank Kapoor
38.Sneha Chauhan
39.Sonali Kher
40.Subrata Chattopadhyay

Project Mentors

- **Tushar Sharma**
- **Aditi Deshpande**
- **Mitali Nayak**

Principal Investigator

Dr. D. B. Phatak

Table of Contents

Table of Images

19.

1. Introduction

This document is to describe the specifications of the dashboard for IITBombayX partners who agreed to run Blended MOOCs(Massive Open Online Courses) in their institutions. The document gives a detailed description of the database schema involved in our design of this platform and also highlights the dashboard design to be presented to the participating institutes and what are the various functionalities that can be provided to different roles involved in it.

1.1 Purpose

- To provide complete and concise description about Blended MOOC by IITBombayX.
- To provide the nature of project and product to be delivered clearly.
- Finally to convert the idea into a prototype and further into working model.

1.2 Scope

The MOOC platform hosted by IITBombayX is expected to effectively assist the participating institutes in encompassing the courses hosted on IITBombayX as part of their curriculum to be used for assessing their students' performance. The objective of this project is to make the platform accessible to the complete administrative staff of the participating institutes to evaluate the performance of their students based upon the reports that it provides course-wise or week-wise according to the enrollment of the students.

1.3 Description of Idea

Every year a large number of students enroll into engineering courses in different institutions across the globe. Every college has its own curriculum, teaching methodology and exams.The quality of education is a matter of concern. MOOCs provide easily accessible education and that too from best colleges and their best teachers. But these MOOCs also have the limitation that they have no alternative for practical sessions of laboratories. To overcome these issues we have come up with the idea of blended MOOCs. The idea behind blended MOOC is that the score obtained in MOOC course has to be added to the score of that candidate in his/her institute. This makes sure that students take these courses seriously. In the classes the teachers should engage students in group discussions and problem solving. And whenever a lab session is required,it is conducted in the institute itself.

1.4 Definitions and Abbreviations and Acronyms

User	: Someone who interacts with the web portal
Admin	: Administrator- the person who maintains the entire web application
HOI	:Head of Institute – The person with highest status in Institute
Program Coordinator	: Person who manages all the courses
Course Coordinator	: Person who supervises the activity of professors
Associate Teacher	: Person who manages and teaches the student
Student	: Person who enrolls for a course
MOOC	:Massive Open Online Courses- An edX platform that works to help educational institutions, businesses and teachers easily build and host courses for the world to take.
Web Portal	: A web application which present special facilities for analyzing performance of student

2. System Requirements

2.1 Institute Registration:

2.1.1.Description and Priority:

An institute must be registered on IIT BombayX before accessing the online courses under MOOC.

Priority : High

2.1.2 Stimulus and Response:

Precondition: Head of Institute sends a formal email to IITBombayX admin regarding the partnership with IITBombayX.

1. Once the email (proposal) is verified by admin, an email will be sent to the head of the institute with the institute registration form link.

2. Head of the institute will fill up the institute details along with his own details and the details of the program coordinator .

3. After verification by admin, institute account will become active.

Post condition: Institute gets registered with IIT Bombay X and the teachers and students of institute can easily access the courses.

2.2 Registration of user

2.2.1 Description and priority:

In order to access the functionalities of the system, the user first needs to fill up the registration form.

Priority: High

2.2.2 Stimulus and Response:
Precondition: User accesses home page of IIT Bombay X and clicks register.

1. User fills the registration form and clicks ‘submit’ button.
2. If username already exist or if required field is not filled or if information is wrong show error and submission fails.

3. If no errors, then confirmation mail goes to user email.

4. User has to click on the activation link in the email to activate account.

5. Then user get registered and his account is created.

Post condition: The user is registered and gets his account interface.

2.3 User Login

2.3.1 Description and priority:

This system feature provides the user with a page to log into IITBombayX .
Priority: High

2.3.2 Stimulus and Response:
Precondition: User in home page, clicked ‘login’

1. The user is prompted to enter his username and password.

2. The information is sent to server and validated from MySQL database.

3. If user credentials aren’t correct,error message is displayed.

Post condition: The user is redirected to his homepage.

2.4 Enrolling a student in the course

2.4.1 Description and priority:
To access the course content user must enroll himself in the course .
Priority: High

2.4.2 Stimulus and Response:
Precondition: User must be registered and logged in.
1. User selects a course from a list of available options.
2. Course gets added to the dashboard.
3. User gets enrolled in the course.
Postcondition: Quizzes along with the assignments for the respective course becomes available to the user.

2.5 View courses

2.5.1 Description and priority:

Provide the user with a page to view courses and to view activities Associate with each course. There are different views for a particular user, a student can view activities of only those courses for which he has enrolled. Also Associate teachers and course coordinators can monitor the activities of their respective courses.

Priority: High

2.5.2 Stimulus and Response:
Precondition: User must be registered in IITBombayX.
1. User clicks on a course tab and selects a particular course .
Postcondition: The course page and Associated activities are displayed.

2.6 Manage courses for institutes

2.6.1 Description and priority:
Courses can be added , updated as per the need.
Priority: High

2.6.2 Stimulus and Response:
Precondition: User must have the authority to edit the courses.
1. For adding/updating a course, the Programme Coordinator must choose the appropriate

course and modify the list of added courses.
2. Programme Coordinator has the authority to remove the courses in which his institute has enroll.
3. Courses are modified.
Postcondition: Quizzes along with the assignments for the respective course becomes available to the user.

2.7 Assign grades for quizzes

2.7.1 Description and priority:
This system feature provides a way of evaluating a student's performance by assigning him grades for the quizzes.
Priority: Medium

2.7.2 Stimulus and Response:
Precondition: Student must take the quiz .
1. Admin must create the scoring mechanism .
2. The questions will be evaluated and scores will be added depending on the correctness of the questions attempted.
Postcondition: Monitoring the performance of the student depending on results of the quiz.

2.8 View grades for quizzes

2.8.1 Description and priority:
This system feature provides tools to monitor students performance .
Priority: Medium

2.8.2 Stimulus and Response:
Precondition: Student quiz results must be saved.
1. User clicks on a tab which contains the grading results for all the students from respective institutes.
Postcondition: Performance analysis can be done based on grades acquired by the students.

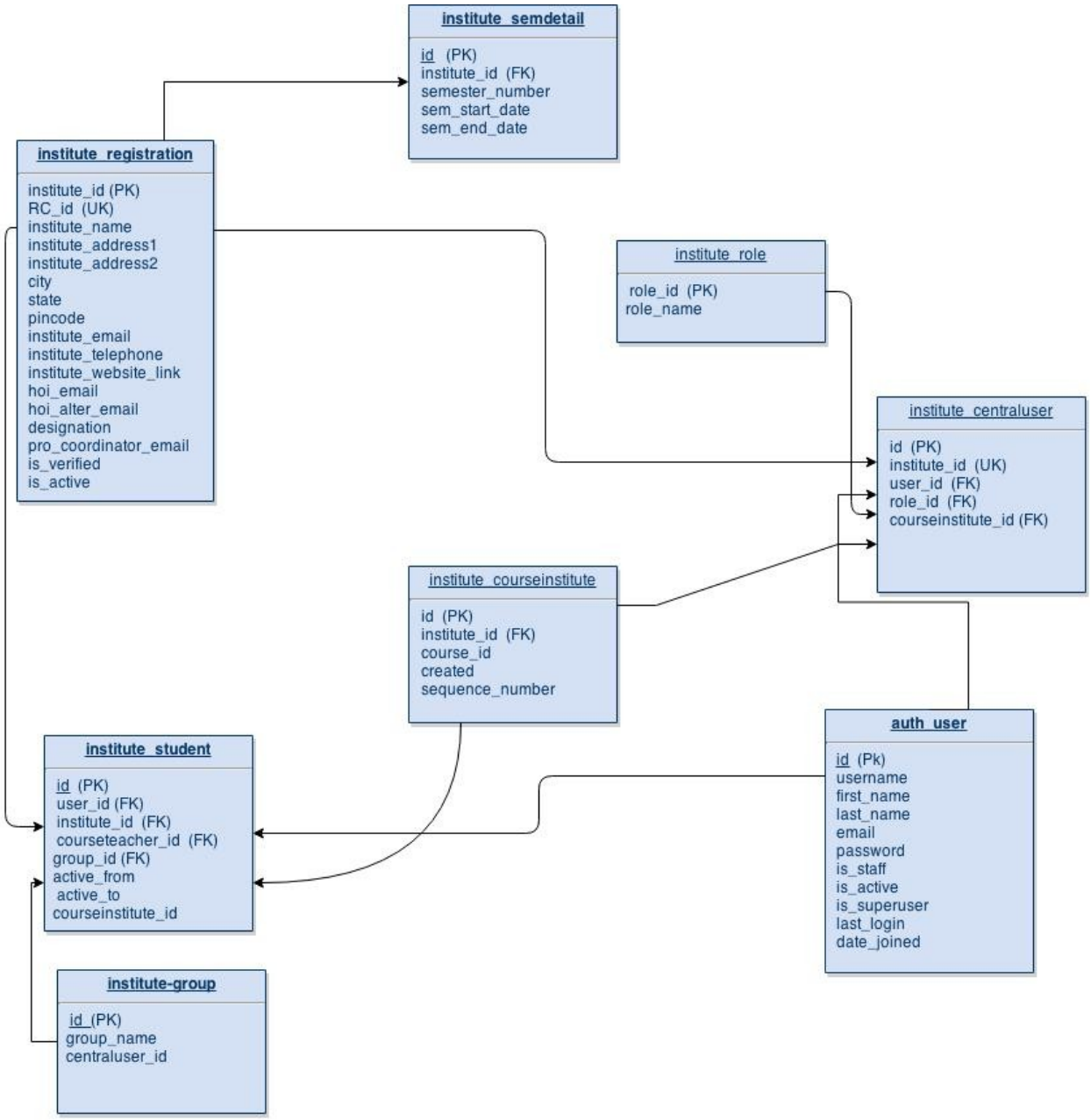
2.9 Performance analysis

2.9.1 Description and priority:
This system feature provides tools to calculate overall grades of the students .

Priority: Medium

2.9.2 Stimulus and Response:
Precondition: Student must have attempted a quiz/assignment .
1. User needs to click on the MIS reports tab available on the dashboard to view the performance of the students.
Post condition: Student performance report becomes available and his progress can be observed.

3. Database schema



Description of Database Schema

1. auth_User

auth_user

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
username	varchar	NO	UNI	NULL	
first_name	varchar	NO		NULL	
last_name	varchar	NO		NULL	
email	varchar	NO		NULL	
password	varchar	NO		NULL	
is_staff	tinyint	NO		NULL	
is_active	tinyint	NO		NULL	
is_superuser	tinyint	NO		NULL	
last_login	datetime	NO		NULL	
date_joined	datetime	NO		NULL	

This table has been taken from the edXapp database. It contains the information of all those users who have registered at the IITBombayX website. We are assuming here that all the users of our new application will have to compulsorily register at the IITBombayX website. The details of the different fields in this table can be found in the edX documentation.

- id: Primary key, and the value typically used in URLs that reference the user. A user has the same value for id here as they do in the MongoDB database's users collection. Foreign keys referencing auth_user.id will often be named user_id, but are sometimes named student_id.

- username: The unique username for a user in the edX system. It can contain alphanumerics and the special characters shown within the brackets: [_ @ + - .]. The username is the only user-provided information that other users can currently see. EdX has never allowed users to change usernames, but may do so in the future.
- first_name: Not used; a user's full name is stored in auth_userprofile.name instead.
- last_name: Not used; a user's full name is stored in auth_userprofile.name instead.
- email: The user's email address, which is the primary mechanism users use to log in. This value is optional by default in Django, but is required by edX. This value must be unique to each user and is never shown to other users.
- password: A hashed version of the user's password. Depending on when the password was last set, this will either be a SHA1 hash or PBKDF2 with SHA256 (Django 1.3 uses the former and 1.4 the latter).
- is_staff: Most users have a 0 for this field. Set to 1 if the user is a staff member of IITBombayX, with corresponding elevated privileges that cut across courses. It does not indicate that the person is a member of the course staff for any given course. Generally, users with this flag set to 1 are either IITBombayX program managers responsible for course delivery, or IITBombayX developers who need access for testing and debugging purposes. Users who have is_staff = 1 have instructor privileges on all courses and can see additional debug information on the Instructor tab. Note: This designation has no bearing on a user's role in the discussion forums, and confers no elevated privileges there.
- is_active: This value is 1 if the user has clicked on the activation link that was sent to them when they created their account, and 0 otherwise. Users who have is_active = 0 generally cannot log into the system. However, when users first create an account, they are automatically logged in even though they have not yet activated the account. This is to let them experience the site immediately without having to check their email. A message displays on the dashboard to remind users to check their email and activate their accounts when they have time. When they log out, they cannot log back in again until activation is complete. However, because IITBombayX sessions last a long time, it is possible for someone to use the site as a student for days without being "active". Once is_active is set to 1, it is only set back to 0 if the user is banned (which is a very rare, manual operation).
- is_superuser: Controls access to django_admin views. Set to 1 (true) only for site admins. 0 for almost everybody. History: Only the earliest developers of the system have this set to 1, and it is no longer really used in the code base.
- last_login: A datetime of the user's last login. Should not be used as a proxy for activity, since people can use the site all the time and go days between logging in and out.

- date_joined: Date that the account was created.

Note: This is not the date that the user activated the account.

2.institute_registration:

institute_registration

Field	Type	Null	Key	Default	Extra
institute_id	int	NO	PRI	NULL	auto_increment
RC_id	int	NO	UNI	NULL	
institute_name	varchar	NO		NULL	
institute_address1	longtext	NO		NULL	
institute_address2	longtext	NO		NULL	
city	longtext	NO		NULL	
state	longtext	NO		NULL	
pincode	int	NO		NULL	
institute_email	varchar	NO		NULL	
institute_telephone	varchar(13)	NO		NULL	
institute_website_link	longtext	YES		NULL	
hoi_email	varchar	NO		NULL	
hoi_alter_email	varchar	NO		NULL	
designation	varchar	NO		NULL	
pro_coordinator_email	varchar	NO		NULL	
is_verified	tinyint(1)	NO		NULL	
is_active	tinyint(1)	NO		NULL	

This table contains the information regarding all the institutes which have registered along with the details of the head of the institute.The different fields here are:

- institute_id : Primary key, an auto incremented value to uniquely identify the Institutes.
- RC_id : Stands for remote center ID. It is a unique ID provided to the remote centers by IIT Bombay. It will be null for the institutes which are not remote centers.
- institute_name : Name of the institute.

- institute_address1 : Address of the institute .(part1)
- institute_address2 : Address of the institute .(part2)
- city : It shall contain the city in which the institute is located.
- state : It is the state in which the institute is located.
- pincode : The area pin code where the institute is located.
- institute_email : Email id of the institute for the purpose of contact.
- institute_telephone : The official telephone number of the head of the institute.
- institute_website_link : Website of the Institute, if present. If not, it can have null values.
- hoi_email : The email id of the institute head.
- hoi_alter_email : The alternate email id of the institute head.
- designation : The designation of the head of the institute e.g. Director, vice chancellor etc.
- pro_coordinator_email : The email id of the program coordinator.
- is_verified: This specifies whether the admin has verified the institute or not. If it is 1, it means the institute is verified.
- is_active: This field indicates if a institute is active with the Blended MOOC program.If it is set to 1 ,it means a institute is following this agreement else if it is 0 ,the institute disapproves.

3. institute_student:

institute_student

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
user_id	int	NO	FOR	NULL	
institute_id	int	NO	FOR	NULL	
group_id	int	NO	FOR	NULL	
active_from	datetime	NO		NULL	
active_to	datetime	NO		NULL	
courseteacher_id	int	NO	FOR	NULL	

The institutes to which each student belongs shall be stored here. The students are only supposed to register on the IITBombayX website. The institutes shall be assigned to him by the program coordinator of the institute. They shall provide the list of email ids of the students of their institute which we shall use to map with the auth_user table.

- Id: It is the primary key. Generated with respect to the table.
- user_id: A foreign key from the auth_user table.
- institute_id: It is the foreign key from the institute_registration table.
- group_id: A foreign key from the institute_group table.
- courseteacher_id: A foreign key from the institute_courseteacher.
- active_from: Start time of the course as enrolled by the student.
- active_to: End time of the course.

4.institute_semdetail:

institute_semdetail

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
semester_number	int	NO		NULL	
institute_id	int	NO	FOR	NULL	
sem_start_date	date	NO		NULL	
sem_end_date	date	NO		NULL	

This table contains the details about the semester of each institute.

- Id: It is the primary key. Generated with respect to the table.
- institute_id: A unique key from institute_registration table.
- sem_id: A foreign key from the institute_semester table.
- sem_start_date: The starting date of the semester per institute.
- sem_end_date: The end date of the semester per institute.

5. institute_role:

institute_role

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
role_name	varchar	NO		NULL	

This table defines the roles of staff in an institute. It has predefined values. It has following fields:

- id : Primary key which is auto incremented.
- role : This field contains designation of faculties like coordinator, associate_teacher etc.

6. institute_group:

institute_group

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
group_name	varchar	NO		NULL	
courseteacher_id	Int	NO	FOR	NULL	

This table contains the details of a group of students under a particular course teacher.

- id : Primary key which is auto incremented.
- group_name: Name of the group.
- courseteacher_id: The id of the course teachers are provided per group.

7.institute_centraluser:

institute_centraluser

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
user_id	int	NO	FOR	NULL	
institute_id	int	NO	FOR	NULL	
role_id	int	NO	FOR	NULL	
courseinstitute_id	int	YES	FOR	NULL	

This table consists of details regarding each role(eg. HOI, coordinator,etc.) of the staff in the institute.

- id : Primary key which is auto incremented.
- institute_id:A foreign key from institute_registration table.
- user_id: A foreign key from auth_user table.
- role_id: A foreign key from institute_role table.
- courseinstitute_id : A foreign key from the institute_course_institute table.

8.institute_courseinstitute:

institute_courseinstitute

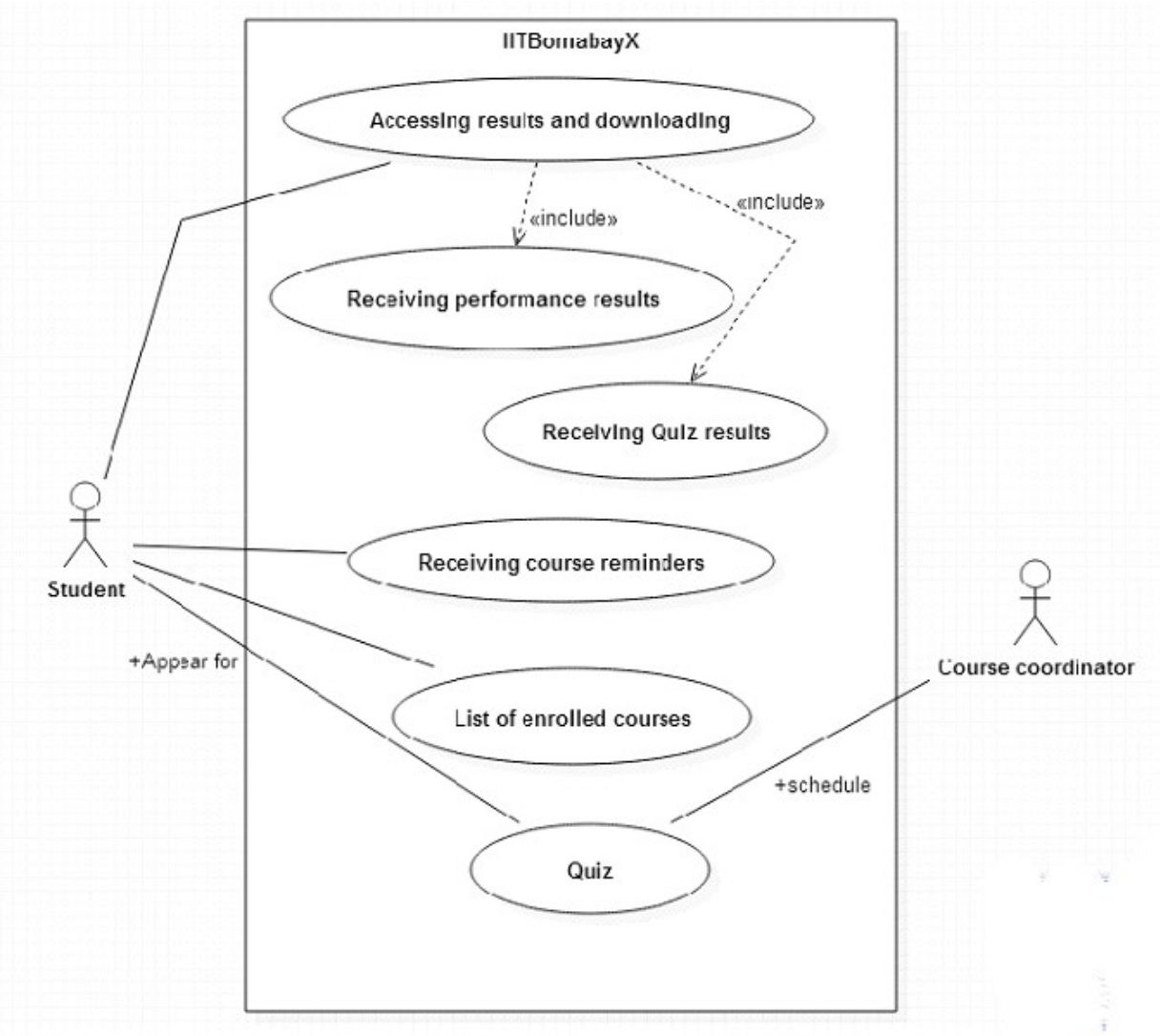
Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
course_id	int	NO		NULL	
institute_id	int	NO	FOR	NULL	
created	date	NO		NULL	
sequence_number	tinyint(1)	NO		NULL	

This table contains the mapping of institute to a particular course taken in a particular year. This table will be mostly used to retrieve data such as comparing the performance of a student or a group of students based on previous years.

- id: Primary key which is auto incremented.
- course_id: Id of the course is being mapped in this table.
- institute_id: A foreign key from institute_registration table.
- created: date is stored when the course was created.
- sequence_number: Suppose in a year a course is conducted twice so to keep track of that information we assign a number.

4. Use Case Model

Actor: Student



List of Use cases:

Use case: List enrolled courses.

Purpose: To get the list of the courses the student has enrolled on IITBombayX.

Actors: Student

Precondition: Student should be logged into the system, on his account.

Basic Flow:
1. Student logs into his account and goes to his homepage.
2. The student is then taken to a page showing all the courses he has enrolled for, in a tabular form. This is accessible from his account home page.

Exception Flow: None.

Postcondition: The student is able to see all the courses he is enrolled for.

Use-case:Accessing results and downloading

Purpose:Student can go through their progress and improve there skills if needed. It allows the students to keep records of their performance and accordingly tune up themselves to improve their grades in the course.

Actors:Student

Precondition:
1. The student must have an existing account on IITBombayX.
2. He must have access to the reports at the time of placing request – i.e., there are no ongoing quizzes or examinations during that period.

Basic Flow:
1. The Student selects a particular course from his account page on IITBombayX, and is taken to the page for it.
2. On the course he finds the 'courseware' tab, which takes him to the courseware page . Under this, he finds the 'Progress' tab.
3. This page shows him his current progress in the course and the details for his responses to and his scores in the exercises and quizzes.

Exception Flow:
1. If the student is accessing the results at a time when quizzes are in progress, or the due date for the quizzes has not been reached, he will not be able to see his results, as the current scores of his tests will not be added to it.

Postcondition:
The student will be able to successfully see the results and if permitted by the coordinator.

Use-case:Appearing for Quiz

Purpose:To get the performance of the student about the course

Actors:Student, Course coordinator

Precondition:
1. The student must have an existing account in the system.
2. The student must be enrolled in the course in whose quiz he is trying to participate.
3. The quiz must be active at the time (within due date and after activation date).

Basic Flow:
1. The student goes to his dashboard, and selects the course.
2. He sees the quiz which is scheduled for the course, and clicks on the appropriate link/button to start the quiz..
3. He is taken to a new page which tells him the rules and regulations for the quiz. He must click on the appropriate 'Take me to the quiz' button to start the quiz.
4. There will be questions of the formats such as MCQ, writing answer options, and single choice answers. For the first three types, if the 'Check' button is available, the answers will be evaluated immediately and the user will be shown the results of theanswers, together with explanations, if the user clicks on the 'Show Explanations' button. Further submissions for the quizzes is not allowed, once the 'Check' button is clicked.
5. The 'Check' button will appear only if the quiz is a graded quiz and the current date is past the due date. Otherwise, the user can see the answers through the 'Check' button only for the exercises Associate with the lessons at the time of practicing.

Exception Flow:
1. If the student tries to access the quiz before the start date and time of the quiz, then he will be shown a warning saying he cannot do so.
2. If a student tries to submit answers multiple times after the submission of answers to a quiz, he will see a warning message, preventing him from doing so.
3. Once a quiz is past the due date, the student cannot visit the quiz to change the answer. He can only see the answers and his results.

Post condition:
The student successfully completes his quiz, and presents his answers for grading. If the answers are evaluated by the auto-grader, he can see the results of his answers immediately, and the scores will be added to his progress in his account. He can then view the results of the quiz and the change in his overall score from his own dashboard.

Use-case: Receiving course reminders

Purpose: The student receives various reminder and notifications regarding the courses he has enrolled for, including reminders for course enrollment, quiz dates, release of course materials and exercises.

Actors: Student

Precondition: The student must have a registered account in the IITBombayX MOOC platform, and in order to receive notifications regarding the courses, he must be enrolled in a particular course.

Basic Flow:
1. If the course coordinator wishes to include a student in his course, the student will receive an email in his inbox informing him to enroll in the course within a particular deadline. He will also receive similar notifications in his dashboard.
2. If new material has been released as part of the course, such as new exercises or new questions, the student will receive new notifications in his email regarding the same, and similar notifications in his dashboard.
3. The student will also receive notifications in his email and in his dashboard regarding the scheduling and deadlines of his quizzes.
4. The student simply needs to open his email inbox or his dashboard notifications tab in order to find the alerts of notifications for the same.
5. He then responds to these notifications accordingly.
6. In case the requests are responded to, he will not receive any more notifications regarding the same instance.

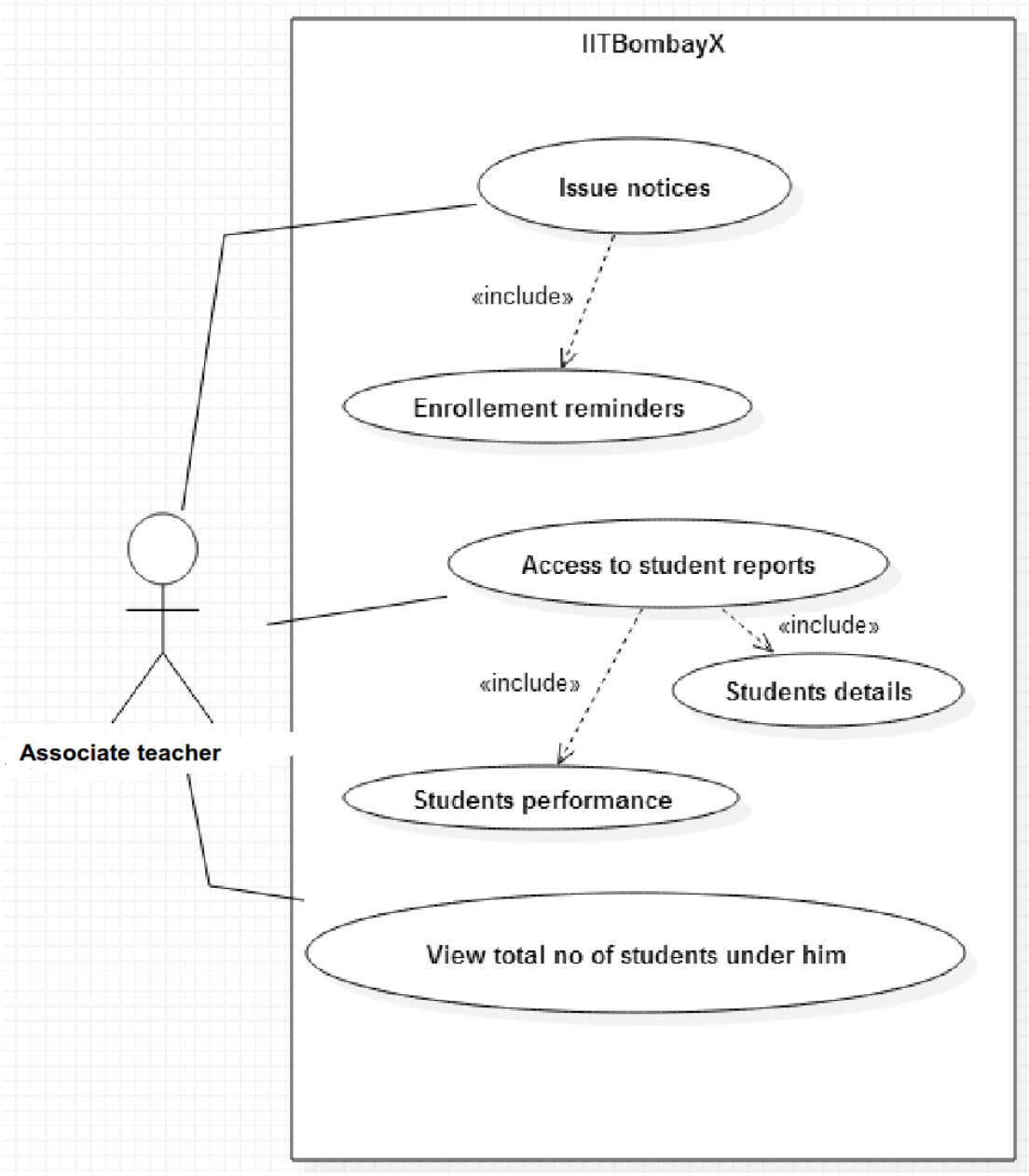
Exception Flow:

1. Due to the servers being clogged up at the time of sending out the notifications, the notifications may not be sent out immediately. In such a case, the notifications will be stored in a queue on the server and will be sent out at the earliest depending on the scheduling of the server processes.

2. The student may receive the notifications but not respond to them immediately. In such a case, he will repeatedly receive notifications from the system for the same (such as course enrollment, quiz alerts, etc.) until the deadlines are reached or they are responded to. After the deadlines are reached, he will stop receiving notifications for the same.

Post condition: The student will receive the notifications for the appropriate reminder at the earliest, and responds to them.

Actor: Associate teacher



List of use cases:

Use-case: Issue Notices

Purpose: The associate teacher will issue notices for the students about the enrollment reminders and quizzes of the course.

Actors: Associate Teacher

Precondition: The Associate teacher should have logged in to the system. He should be aware of the courses allotted to him.

Basic Flow:
1. The coordinator need to login into the system, by clicking the “Login” button.

2. He need to look for the different issues considering the enrollment reminders for the students.

3. This may include reminders for an upcoming quiz of the course he is coordinating, or reminders for enrolling in a course.

4. He will then select the 'Email Facility' link from his dashboard, and for a particular event, he will select the students he wants to send the notification to, and send it.

5. In case of sending course reminders to students who have not joined yet:
    1. The coordinator goes to his dashboard, and clicks on the subject course from among the list of courses he is coordinating.

    2. He sees the course details on his dashboard page, and finds the number of students and the details of students who have signed up for his course, and who have not, based on a list of student emails he had uploaded earlier.

    3. He then clicks on 'Send Enrollment Reminder' link, which sends a reminder to the students via mail and notification to enroll themselves into the course.

Exception Flow:

1. If the coordinator fails to send the enrollment reminders on time either to all the students or to few of them, the students will not be able to get their schedule.

Post condition: The Coordinator has successfully issued the required notices to the students.

Use-case: Access to student reports

Purpose: To get the student details and the student performance reports.

Actors: Associate teacher

Precondition:
1. The Associate teacher has successfully logged into his account on IITBombayX.

2. The Associate teacher has successfully issued the notices to all the students regarding different quizzes and the students have appeared for the quizzes held in the IITBombayX.

Basic Flow:
The Associate teacher goes to his dashboard page, where he clicks on the 'MIS Reports' tab button to go the page for student-wise reports. Here, he has links for downloading the details of performance in the following categories:

1. All students:
    1. The Associate teacher can click on this tab/link to go to the page containing the links for all the students in all the courses enrolled under him, divided by courses, or grouped together.

    2. The page contains a tabular display containing the names of all the students, and a link to their course performances.

    3. There will be an option to download this information in pdf format, which can be used by the Associate teacher.

    4. The Associate teacher will also have the option to download all the student information in an unstructured format, which will include all the details of the course grades, and the quiz results of the students – by clicking on the 'Download CSV button'.

2. Course-wise and semester wise division of students:
    1. The students will be divided into the courses and the semesters they are in (chronologically), and links will be available for each one of them.

    2. On clicking the 'Groupwise' link for MIS Reports, the Associate teacher is taken to a page. The page contains a tabular display containing the details of all the students and the course they are enrolled in, together with the start and end date of the course, and the semester they are taking it in (in the case of the indepedent student, however, this will not be available).

    3. The columns will show the final grades of all the students in their respective courses.

    4. There will be an option to download this information in pdf format, which can be used by the Associate teacher.

    5. The Associate teacher will also have the option to download all the student information in an unstructured format, which will include all the details of the course grades, and the quiz results of the students.

3. Analytical results:
    1. This includes all the analytical information that is provided by the InSight module of the IITBombayX system, to the Associate teacher at the supervisory level. It can be accessed by clicking on the 'Download Reports' tab in the dashboard.

2. This provides information such as course performance graphs (using average course grades as metric for all students or individual course grades for single students) over the weeks, quiz performances over the weeks for the students (same as before, using the quiz grade percentages as metric). These will be available as separate tabs in both views 1 and 2 above.
3. These views and details can be viewed in the browser and dashboard, and downloaded in pdf format.

Exception Flow:
1. If the Associate teacher requests for the reports at a time other than the scheduled time for the corresponding report (hourly, weekly, daily), then he will see a message telling him that the data is not up to date for the generation of reports.

2. The Associate teacher cannot request for the reports in the middle of courses and quizzes. If he does, he will be met with warnings saying that he will not receive up to date information if he tries to do so.

Post condition: The Associate teacher had accessed the student report successfully, and downloaded them if required.

Use case: View all the students enrolled

Purpose: The Associate teacher can view all the students enrolled for a particular course under him.

Precondition: The Associate teacher must be logged into his account on IITBombayX.

Basic Flow:
1. The Associate teacher goes to his dashboard and clicks on the appropriate tab.

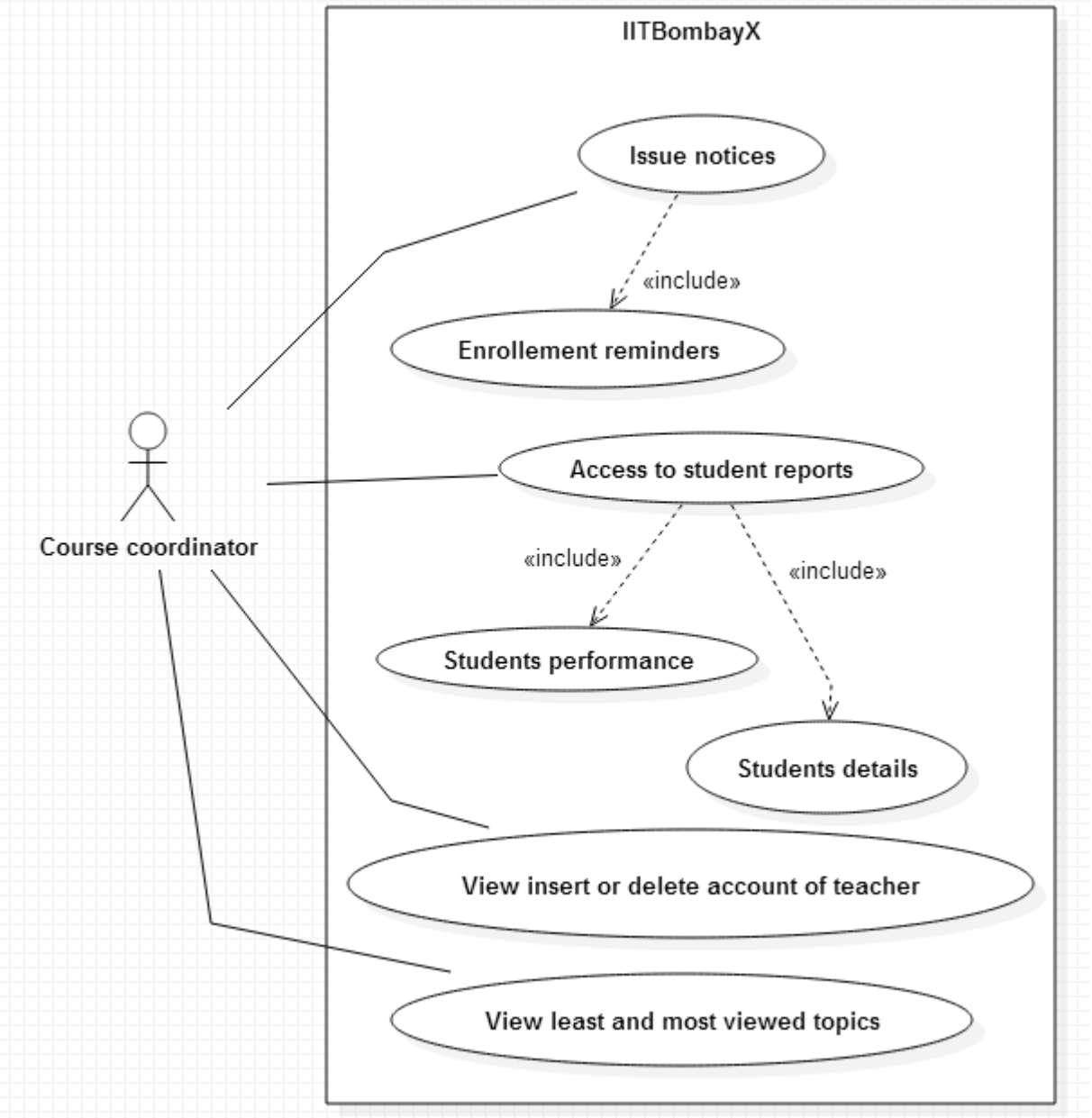
2. He is taken to a page which shows the links: 'All students', and all the courses he is currently overseeing.
1. If he clicks on the 'All students' tab, he will see the list of all the students who are enrolled in the courses under him.

2. If he clicks on any one of the courses links, he will see a list of all the students who are enrolled in that particular course.

Exception Flow:
1. If the course Associate teacher does not have any courses allotted under him, his dashboard page for the same will be empty.

Post condition: The Associated teacher is able to view the list of students enrolled in the course.

Actor: Course Coordinator



List of use cases

Use Case: View the most and least viewed topics

Purpose: Course Coordinator can analyze the topics within a course depending on most viewed and least viewed topics.

Actor: Course Coordinator

Precondition: The Course coordinator must be logged into his account on IITBombayX.

Basic Flow:

1. The course coordinator goes to his dashboard and clicks on the 'Course Analysis' tab under MIS Reports.

2. He is taken to a page which shows the following:

a) Top 5 most popular topics within a course, and

b) Top 5 least popular topics viewed within the course,

Determined by the number of users who visited videos on that topic.

Exception Flow:

1. In the beginning of participation of the institute, when there is not sufficient data to determine trends, the tables will not be displayed.

Post condition: The course coordinator will get to see the tables.

Use Case: Issue Notices

Purpose: The Course Coordinator will issue notices for the students about the enrollment reminders and quizzes of the course.

Actor: Course Coordinator

Precondition: The Course Coordinator should have logged in to the system. He should be aware of the courses allotted to him.

Basic Flow:

1. The Course Coordinator need to login into the system,by clicking the “Login” button.

2. He needs to look for the different issues considering the enrollment reminders for the students.

3. This may include reminders for an upcoming quiz of the course he is coordinating, or reminders for enrolling in a course.

4. He will then select the 'Email facility' tab from his dashboard, and for a particular event, he will select the students he wants to send the notification to, and send it.

5. In case of sending course reminders to students who have not joined yet:

1. The Course Coordinator goes to his dashboard, and clicks on the subject course from among the list of courses he is coordinating.

2. He sees the course details on his dashboard page, and finds the number of students and the details of students who have signed up for his course, and who have not, based on a list of student emails he had uploaded earlier.

3. He then clicks on 'Send Enrollment Reminder' link, which sends a reminder to the students via mail and notification to enroll themselves into the course.

6. Send Link for Registration to the students via a special notification to selected students.

Exception Flow:

1. If the Course Coordinator fails to send the enrollment reminders on time either to all the students or to few of them, the students will not be able to get their schedule.

Post condition: The Course Coordinator has successfully issued the required notices to the students.

Use Case: View, Insert, Update and Delete accounts of the Associate teachers

Purpose: The Course Coordinator can view, insert, update and delete the accounts of the teachers teaching his course.

Actors: Course Coordinator

Precondition: He has to be logged into his account, and must have a course allotted to him.

Basic Flow:

1. Course Coordinator goes to his dashboard, and clicks on the 'Authenticate Associate Teacher' tab.
2. Options are available for the following operations:

- View the Associate teachers currently allotted under his course, through a table.
- Insert a new Associate teacher in the course by sending him an email containing a one-time activation link to create an account for the Associate teacher. This assigns him a username and password.
- Update is used to change the details of the Associate teacher in a particular course, such as changing his course allotted, and personal information.
- Delete an Associate teacher's details from the current list, which deletes his entire details as an Associate teacher.

Exception Flow:

1. During insertion of Associate teachers via email, the dashboard compares the email input with the existing list to see if the person already exists as an Associate teacher. If he does, then there will be a message showing the error, the operation will abort.

   Post condition: The course coordinator is able to view and manipulate the information about the Associate teachers.

Use Case: Access to student reports

Purpose: To get the student details and the student performance reports.

Actors: Course Coordinator

Precondition:

1. The Course Coordinator has successfully logged into his account on IITBombayX.
2. The Course Coordinator has successfully issued the notices to all the students regarding different quizzes and the students have appeared for the quizzes held in the IITBombayX.

Basic Flow:

The Course Coordinator goes to his dashboard page, where he clicks on the 'MIS Reports' tab. Here, he has links for downloading the details of performance in the following categories:

1. All students (link on the Performance Tab):
   1. The Course Coordinator can click on this tab/link to go to the page containing the links for all the students in all the courses enrolled under him, divided by courses, or grouped together.
   2. The page contains a tabular display containing the names of all the students, and a link to their course performances.
   3. There will be an option to download this information in pdf format, which can be used by the course coordinator.
   4. The Course Coordinator will also have the option to download all the student information in an unstructured format, which will include all the details of the course grades, and the quiz results of the students.
2. Course-wise and semester wise division of students (link on the Performance Tab):

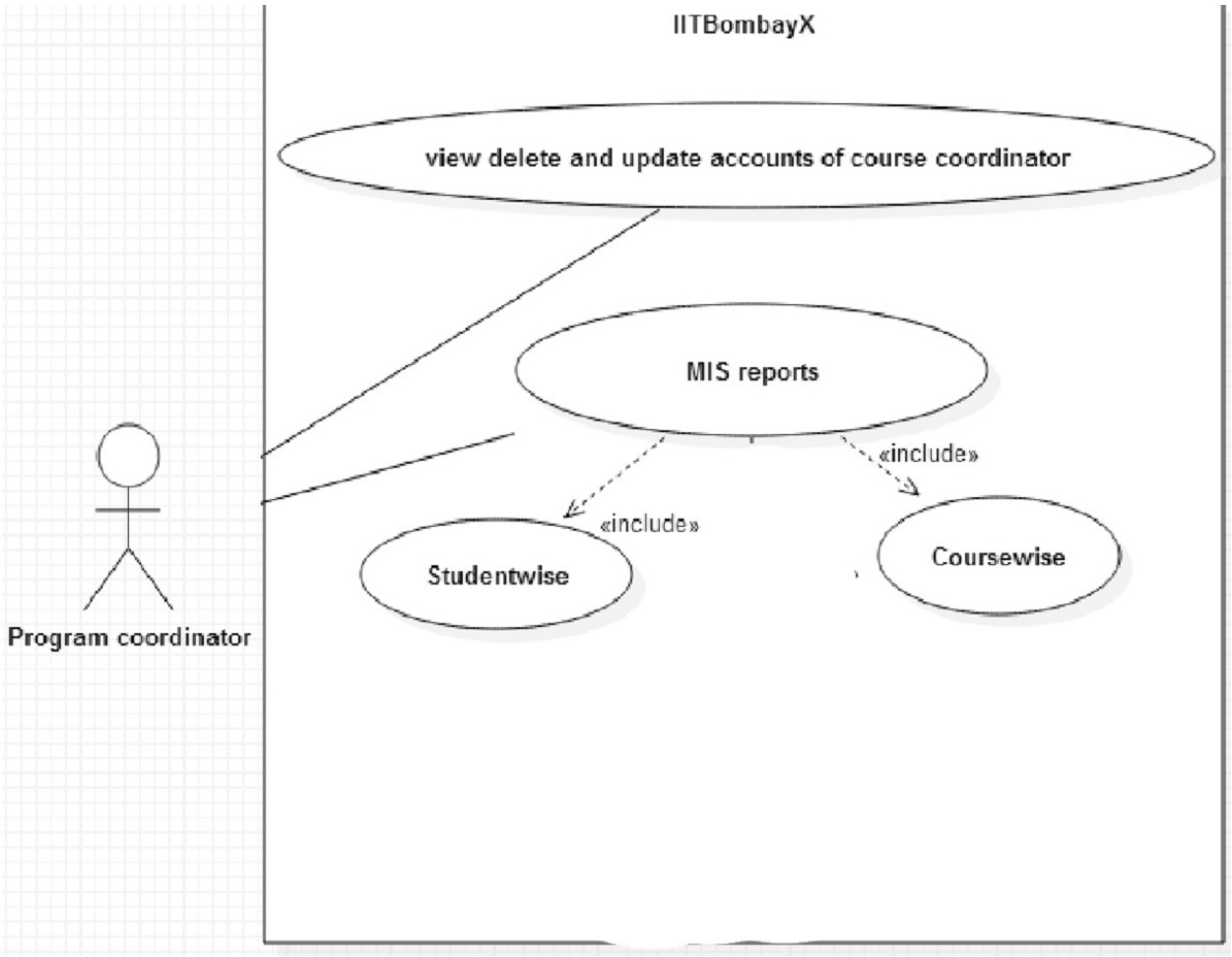
    1. The students will be divided into the courses and the semesters they are in (chronologically), and links will be available for each one of them.
    2. On clicking the link, the Course Coordinator is taken to a page. The page contains a tabular display containing the details of all the students and the course they are enrolled in, together with the start and end date of the course, and the semester they are taking it in (in the case of the indepedent student, however, this will not be available).
    3. The columns will show the final grades of all the students in their respective courses.
    4. There will be an option to download this information in pdf format, which can be used by the course coordinator.
    5. The Course Coordinator will also have the option to download all the student information in an unstructured format, which will include all the details of the course grades, and the quiz results of the students.
3. Analytical results (link on the Course Analysis Tab):
    1. This includes all the analytical information that is provided by the InSight module of the IITBombayX system, to the Associate teacher at the supervisory level.
    2. This provides information such as course performance graphs (using average course grades as metric for all students or individual course grades for single students) over the weeks, quiz performances over the weeks for the students (same as before, using the quiz grade percentages as metric). These will be available as separate tabs in both views 1 and 2 above.
    3. These views and details can be viewed in the browser and dashboard, and downloaded in pdf format.
4. The final results: These must be enabled by the Admin at the IITBombayX level, on the completion of the courses of the students. The admin enables these reports to be downloaded by the Course Coordinator, and the Associate teachers and can further send them to the students in the courses under them.

Exception Flow:

1. If the Associate teacher requests for the reports at a time other than the scheduled time for the corresponding report (hourly, weekly, daily), then he will see a message telling him that the data is not up to date for the generation of reports.
2. The Associate teacher cannot request for the reports in the middle of courses and quizzes. If he does, he will be met with warnings saying that he will not receive up to date information if he tries to do so.

Post condition: The Course Coordinator had accessed the student report successfully, and downloaded them if required.

Actor: Program coordinator



List of use cases:

Use Case: View, Delete, and Update account of course coordinator

Purpose: The Program Coordinator can view, insert, update and delete the accounts of the teachers teaching his course.

Actors: Program Coordinator

Precondition: He has to be logged into his account, and must have a course allotted to him.

Basic Flow:

1. Program Coordinator goes to his dashboard, and clicks on the 'Authenticate Course Coordinator' tab.
2. Options are available for the following operations:

- View the course coordinator currently allotted under his course, through a table.
- Insert a new course coordinator in the course by sending him an email containing a one-time activation link to create an account for the course coordinator. This assigns him a username and password.
- Update is used to change the details of the course coordinator in a particular course, such as changing his course allotted, and personal emotion.
- Delete an course coordinator's details from the current list, which deletes his entire details as an course coordinator.

Exception Flow:

1. During insertion of course coordinator via email, the dashboard compares the email input with the existing list to see if the person already exists as a course coordinator. If he does, then there will be a message showing the error, the operation will abort.

Postcondition: The program coordinator is able to view and manipulate the information about the course coordinator.

Use Case: MIS Reports

Purpose: The program coordinator can view performance reports on a course-wise, course- coordinator-wise, and student-wise manner.

Precondition: The Program Coordinator must be logged into the IITBombayX account.

Actor: Program Coordinator

Basic Flow:
1. The Program Coordinator will go to his dashboard and click on the MIS reports Tab.

2. He can then view the information in the following manner:

1. Student-wise: He will taken to a page which gives a list of all the students under the institute.
    1. For each particular student, he can view the progress report of that student in a particular course by clicking on the corresponding link for the students.

2. Course-wise: He will taken to a page which gives a list of all the courses under the institute.
    1. For each course, there is a link to the list of students under the course, divided by semester, in a tabular form, which also gives the aggregate scores in that course for the student.
    2. If the coordinator clicks on the name of the student, then he can view the performance of the student for that course individually.

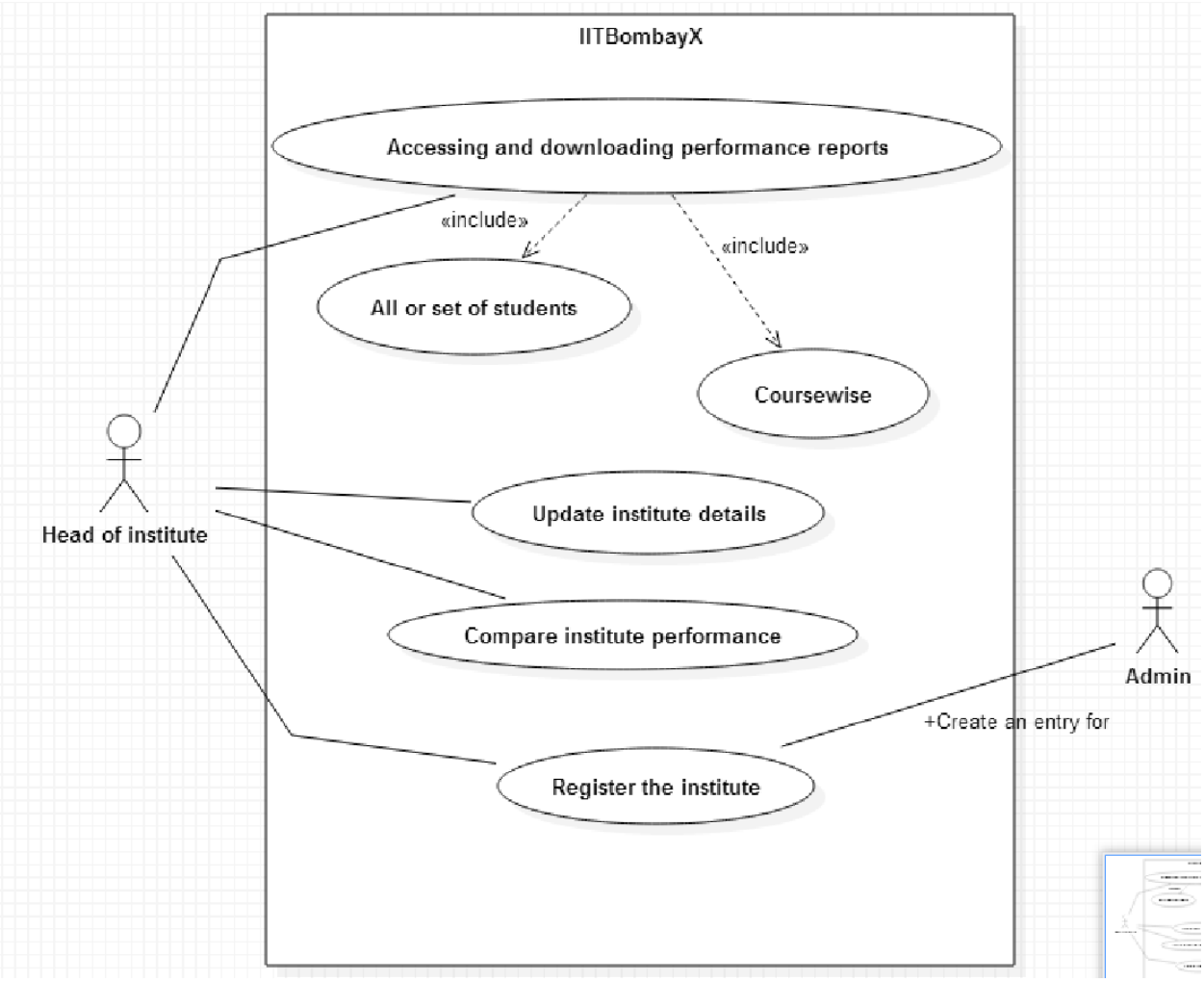
3. Additionally, he can generate reports regarding the information about the students enrolled under the various users and courses.

3. If the coordinator clicks on the name of the student, then he can view the performance of the student for that course individually.

Exception Flow: None

Post condition: The program coordinator is able to navigate across the students in different course and course coordinators, and individually as well.

Actor: Head of Institute



List of use cases:

Use Case: Accessing and downloading performance reports.

Purpose: To get the student details and the student performance reports.

Actors: Head of Institute (HOI)

Precondition:
1. The HOI has successfully logged into his account on IITBombayX.

2. The HOI has successfully issued the notices to all the students regarding different quizzes and the students have appeared for the quizzes held in the IITBombayX.

Basic Flow:

The HOI goes to his dashboard page, where he clicks on the MIS Reports tab. Here, he has links for downloading the details of performance in the following categories:

1. All students (link on the Cumulative Student Performance Tab):

    1. The HOI can click on this tab/link to go to the page containing the links for all the students in all the courses enrolled under him, divided by courses, or grouped together.

    2. The page contains a tabular display containing the names of all the students, and a link to their course performances.

    3. There will be an option to download this information in pdf format, which can be used by the HOI.

    4. The HOI will also have the option to download all the student information in an unstructured format, which will include all the details of the course grades, and the quiz results of the students.

2. Course-wise and semester wise division of students (link on the Coursewise Tab):

    1. The students will be divided into the courses and the semesters they are in (chronologically), and links will be available for each one of them.

    2. On clicking the link, the HOI is taken to a page. The page contains a tabular display containing the details of all the students and the course they are enrolled in, together with the start and end date of the course, and the semester they are taking it in (in the case of the indepedent student, however, this will not be available).

    3. The columns will show the final grades of all the students in their respective courses.

    4. There will be an option to download this information in pdf format, which can be used by the HOI.

    5. The HOI will also have the option to download all the student information in an unstructured format, which will include all the details of the course grades, and the quiz results of the students.

3. Analytical results (link on the Coursewise Tab):

1.This includes all the analytical information that is provided by the InSight module of the IITBombayX system, to the Associate teacher at the supervisory level.

2. This provides information such as course performance graphs (using average course grades as metric for all students or individual course grades for single students) over the weeks, quiz performances over the weeks for the students (same as before, using the quiz grade percentages as metric). These will be available as separate tabs in both views 1 and 2 above.

3. These views and details can be viewed in the browser and dashboard, and downloaded in pdf format.
4. The final results: These must be enabled by the Admin at the IITBombayX level, on the completion of the courses of the students. The admin enables these reports to be downloaded by the HOI, and the Associate teachers and can further send them to the students in the courses under them.

Exception Flow:
1. If the HOI requests for the reports at a time other than the scheduled time for the corresponding report (hourly, weekly, daily), then he will see a message telling him that the data is not up to date for the generation of reports.
2. The HOI cannot request for the reports in the middle of courses and quizzes. If he does, he will be met with warnings saying that he will not receive up to date information if he tries to do so.

Post condition: The HOI had accessed the student report successfully, and downloaded them if required.

Use Case: Update Institute details

Purpose: HOI can update the details of his/her registered institute.

Precondition: HOI must have logged into his account on IIT Bombay X.

Basic Flow:
1. HOI enters the dashboard and click on the appropriate tab in the dashboard.

2. The details may include the institute's address, website, contact details, head of institute details, etc.

Exception Flow:

3. If any wrong information will be provided by the HOI, as in " entering character in a numeric field" , the corresponding error would be generated.

Post condition: The updated information is reflected in the database tables accordingly.

Use Case: Compare Institute Performance

Purpose: The HOI can compare the performance of his institute's students with the progress reports of students of other institutes.

Precondition: HOI must have logged into his account on IIT Bombay X

Basic Flow:
1. HOI can generate a cumulative progress report of his students in a particular course through 'Institutewise' link under the 'MIS Reports' tab, which takes him to page showing his ranking among the other institutes with their publicly shown statistics, while keeping their identity hidden.

2. Accordingly, he can find the progress reports of the students of other institutes and compare them in the above mentioned way.

Exception Flow: None.

Post condition: HOI can analyze the comparison reports between his institute and other institutes in a graphical or tabular form.

Use Case: Register the institutes

Purpose: To register the institute of the director in the IITBombayX platform.

Actors: Head of Institute, Admin

Precondition: The Head of Institute must be logged into the system.

Basic Flow:
1. The Head of Institute logs into the IITBombayX platform and goes to his dashboard from his profile which is currently having only his details).

2. He sends a special email to the IITBombayX administrator regarding the participation of the institute in the programme, to which the admin will send him a form to be filled up online. The HOI fills this form and submits it.

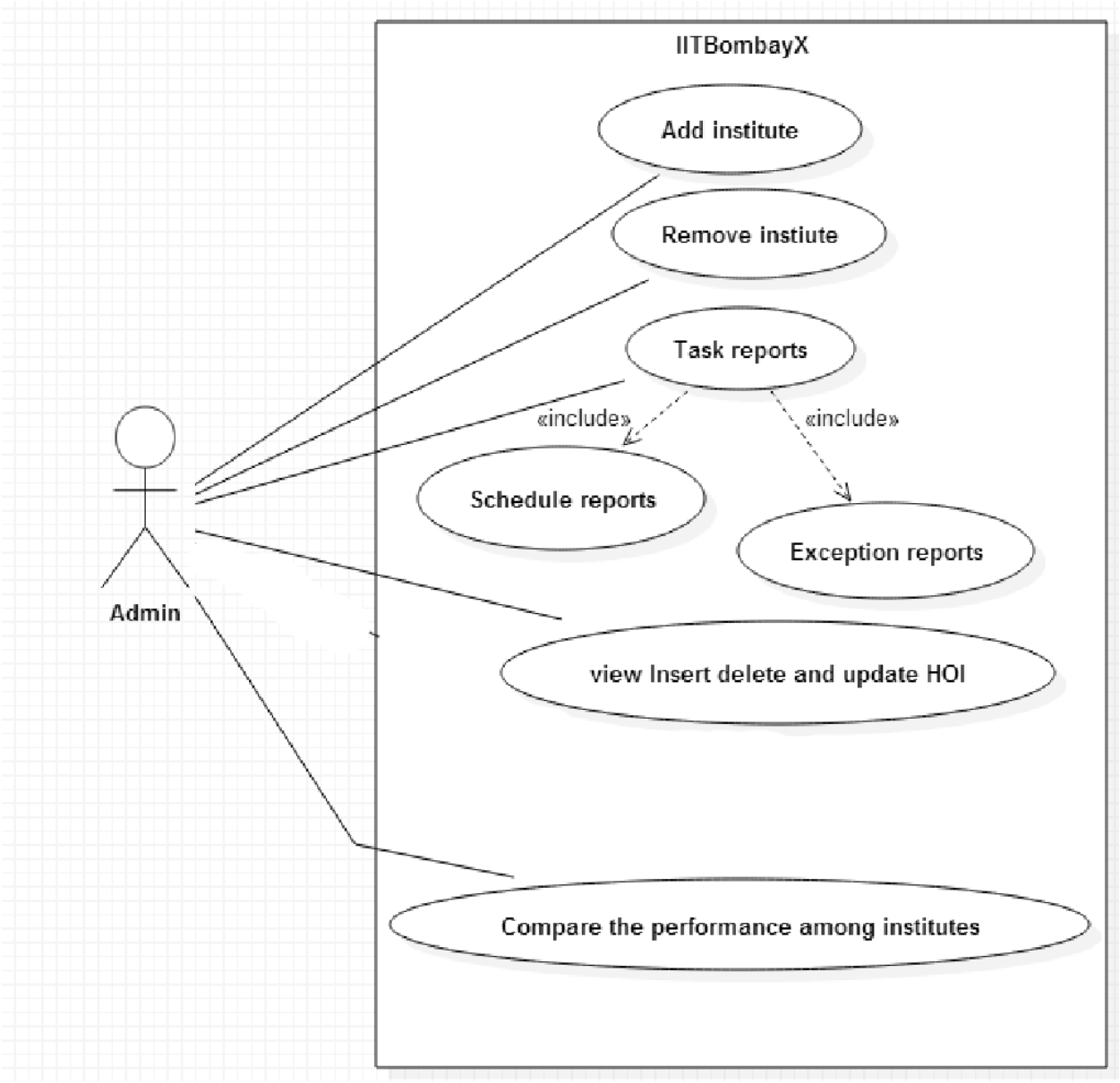
3. The Admin at IITBombayX then performs his use-case roles for creating the entries for the institute.

Exception Flow:

1. The Head of Institute must send the special notification containing the details of the institute to the administrator in order for the institute to be created.

Post condition: Head of Institute have successfully sent the details for the administrator to register the institutes.

Actor: Admin



List of use cases:

Use-case: Add Institute

Purpose: The admin can get different request of the institute, to Associate them with IITBombayX.

Actors:Admin

Precondition: There should be a proper description and a proper details of the institute who all have requested to get associated with the IITBombayX.

Basic Flow:
1. The administrator selects the option for adding a new institute, from his dashboard.

2. The institute needs to fill in the proper details of their institute (see the Head of Institute's use case) in the form given by the admin, and these details are already available with the administrator once the form has been submitted.

3. The administrator provides his proper credentials, and then fires the commands and directives for the institute internally, and initializing other internal tables and making updates as necessary to tables, until the changes are made to incorporate the new institute into IITBombayX.

Exception Flow:
1. If the administrator does not provide the correct credentials, the process of adding institutes will be aborted.

2. If the institute has not provided any details, or the admin tries to add an institute whose details it does not have, then there will be message telling him about the error.

3. If the addition of the institute to the system leads to a problem such as shortage or conflict of resources, the admin will be informed of such events through notifications on his dashboard.

Postcondition: The admin had successfully added the Institute which wants to get Associated with the IITBombayX.

Use Case: Remove Institute

Purpose: The admin can delete the institute Associate with IITBombayX, if some problem occurs for a particular reason.

Actors:Admin

Precondition: The admin need to have a watch on all the institute,he is Associated with, besides being logged into his account.

Basic Flow:
1. The admin needs to login into the system.

2. Having received the request to remove the institute from the system, he goes to his dashboard to select the appropriate 'Remove Institute' link from the options under the Details of Institute tab.

3. In this page, he selects the institutes from the list of institutes on the system, either manually or by a search operation.

4. He clicks on the 'Remove' button/link available to him, and finds a warning message telling him that all the information for this institute will be removed from the system if he continues any further.

5. He provides his credentials (such as password, and key provided by the university director), and then clicks on yes.

6. He is greeted with a diagnostic page wish shows all the details of the removal process, including removal of entries, tables, and internal optimizations if any.

7. Finally, he meets a message telling him whether the system action completed successfully.

Exception Flow:
1. The diagnostic page will tell him any untoward event occurred while removal.

2. If he attempts to remove a institute that does not exist, then he will face an error message.

3. If he fails to provide the key provided by the director of the institute, then the removal process will abort.

Post condition: Admin have successfully removed the institute from the IITbombayX.

Use Case: Tasks Reports

Purpose: To get a schedule report or the exception report

Actors : Admin

Precondition: Admin should be logged into the system.

Basic Flow:

1. The admin should provide the proper task report related to the student.

2. He should make schedule reports or the exception report for the student so that the student will get aware of his performance, and provide the access to download these reports to the individual students only on completion of course, and to the directors of the institutes upon request.

3. He also periodically generates reports regarding the system performance, such as the server usage, total system usage by the students and users, number of accesses to the courses over time, peak usage times, and so on. These can be available as graphs, charts, and reports, and so on.

Exception Flow:
1. If the admin login into the system and the reports he need to generate is not on time, there will be a delay of the whole schedule, which is to take place.
2. If the admin requests for the reports at a time other than the scheduled time for the corresponding report (hourly, weekly, daily), then he will see a message telling him that the data is not up to date for the generation of reports.

Post condition:
The task reports regarding the performance of the students have been successfully generated and are ready to be given to the Head of the institute.

Use Case: View, Insert, Update, and Delete Head of Institute (HOI)

Purpose: The Administrator can insert, delete, view and update the HOI of a particular institute.

Actors: Admin

Precondition: The administrator must be logged into his account on IITBombayX.

Basic Flow:
1. The administrator goes to his account dashboard and clicks on the 'Details of Institute' tab.

2. He will be taken to a page which contains the list of all the participating HOI. The page also contains the search facility to search for HOI by name of institute.

3. Each institute contains the HOI details, and a edit button.

4. On clicking the edit button, the Administrator is goes to a page which allows him to insert, delete, update, or otherwise the view the the HOI allotted to the institute.
1. Insert: The insert function allows a HOI to be added to the list of existing HOI on IITBombayX, based on the request is received from the HOI, and assign the institute to the corresponding HOI.

2. Delete: This allows the removal of HOI from the list.

3. View: Shows the existing details of the HOI.

4. Update: The details of the HOI can be updated by the Administrator via a edit button available next to the individual entries.

Exception Flow:
1. If the same HOI is assigned to another institute, an error message will be shown to the administrator warning about the same.

Post condition:
The HOI details can be viewed and manipulated accordingly by the administrator.

Use case: Compare Institute Performance

Purpose: The admin can compare the performance of one institute students with the progress reports of students of other institutes.

Precondition: Admin must have logged into his account on IIT Bombay X.

Actors: Admin

Basic Flow:
1. Admin can generate a cumulative progress report of students in a particular course in the following ways:

    1. Course-wise: Admin selects any course from the links on the page and then compare the performance among the institutes students based on that course performance.

    2. Semester -wise: Similarly, Admin can select the common courses studied by the students in a particular semester of one institute and generate comparison reports accordingly.

2. Accordingly, he can find the progress reports of the students of other institutes and compare them in the above mentioned ways.

Exception Flow: None.
Post condition: Admin can analyze the comparison reports among all institutes in a graphical or tabular form.

Additional Roles already specified:

At the local level, we additionally have the role already specified as follows:

SysAdmin: He is the technical assistant who handles the entire system. He also assists all the users who access the system.

The other class of users who would be using the IITBombayX MOOCs system would be the people who actually define the course content and structure (MOOCs Course Coordinator), the people who are involved in managing and monitoring the overall activity in the Blended MOOC environment at a global supervisory level (Blended MOOCs Coordinator), and the people who are incharge of managing the system functioning behind the scenes (System administrator). The roles are defined as follows.

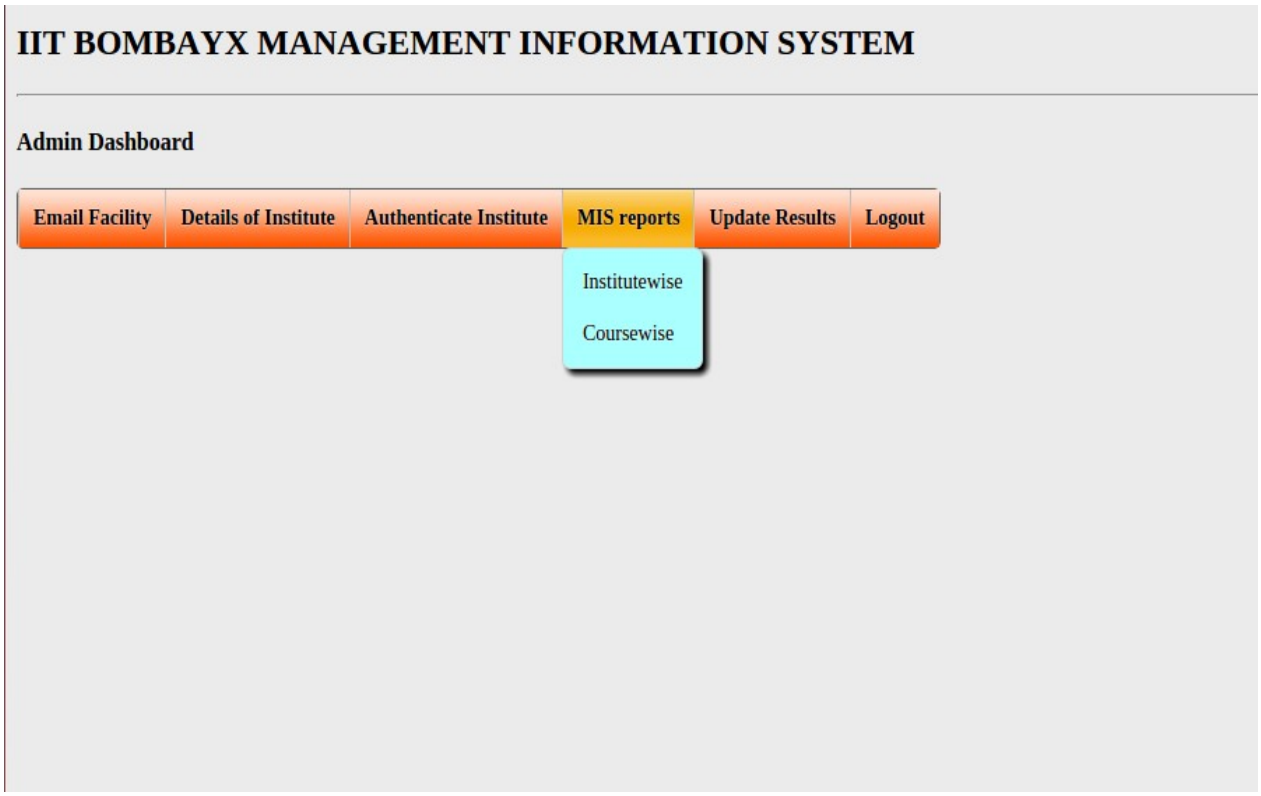
MOOCs Course Coordinator: The MOOCs Course Coordinator is the person who is incharge of creation and management of the individual courses made available on IITBombayX. He can create the course, define the course structure in terms of the week division of the course, and upload the videos. He also creates the practice examples for the students to work out and to grade the students, Additionally, he can communicate with the instructor in charge and incorporate additional problems and material in the course as per the former's suggestion.

Blended MOOCs Coordinator : The Blended MOOCs Course Coordinator is in charge of managing the overall course activities of all the courses on the IITBombayX MOOC system. He can monitor the progress of the institutes on the different courses that are currently active, as well as the performances and activities of the individual students. He can also download and review reports on a scheduled basis for evaluation of performances on course wise, institute wise and other factors, such as enrollment behaviour, demographic patterns, and so on.

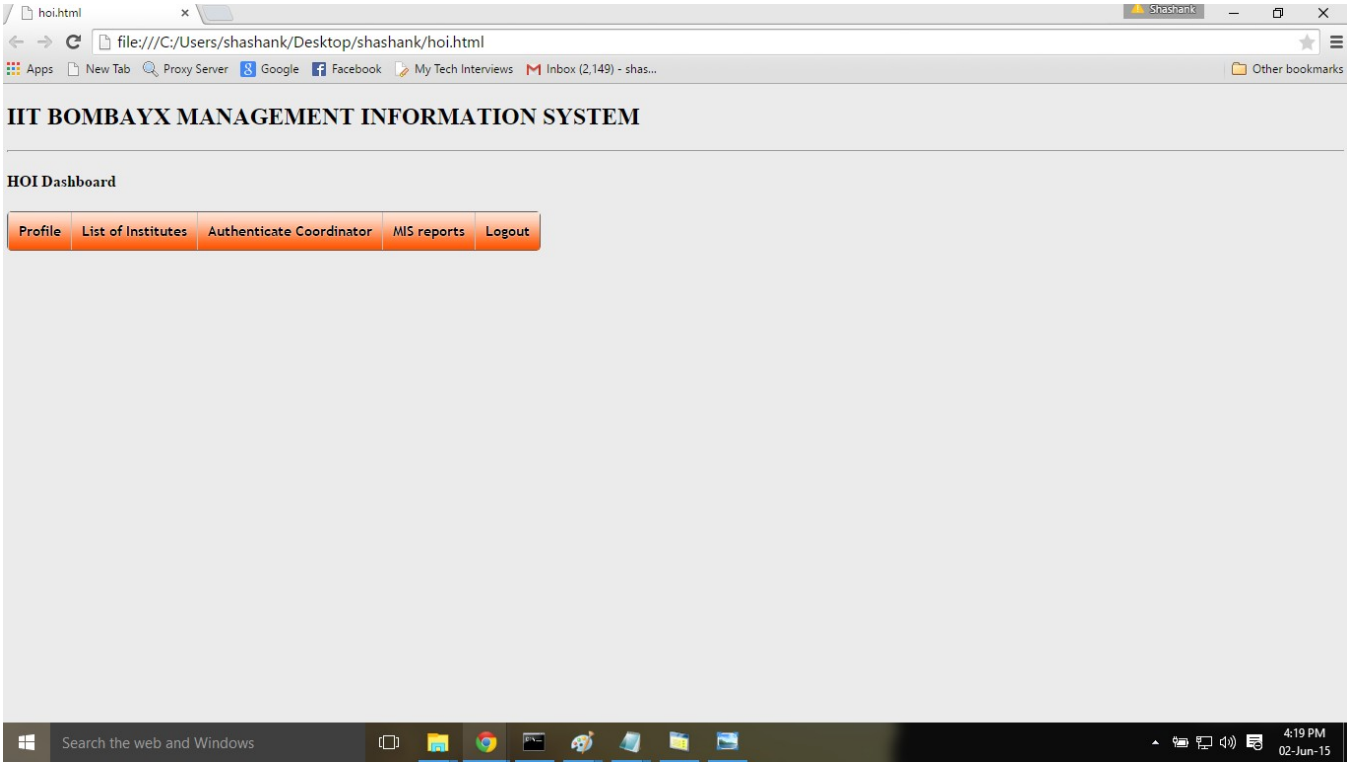
System Administrator : The system administrator is in charge of managing the system performance and monitoring exception cases in system behaviour, such as failures. He occasionally generates reports on features such as disk usage of course contents, the amount of data being stored on the servers, registration and email statistics, and certain exception reports for events such as disk usage exceeding safe limits, or similar cases.

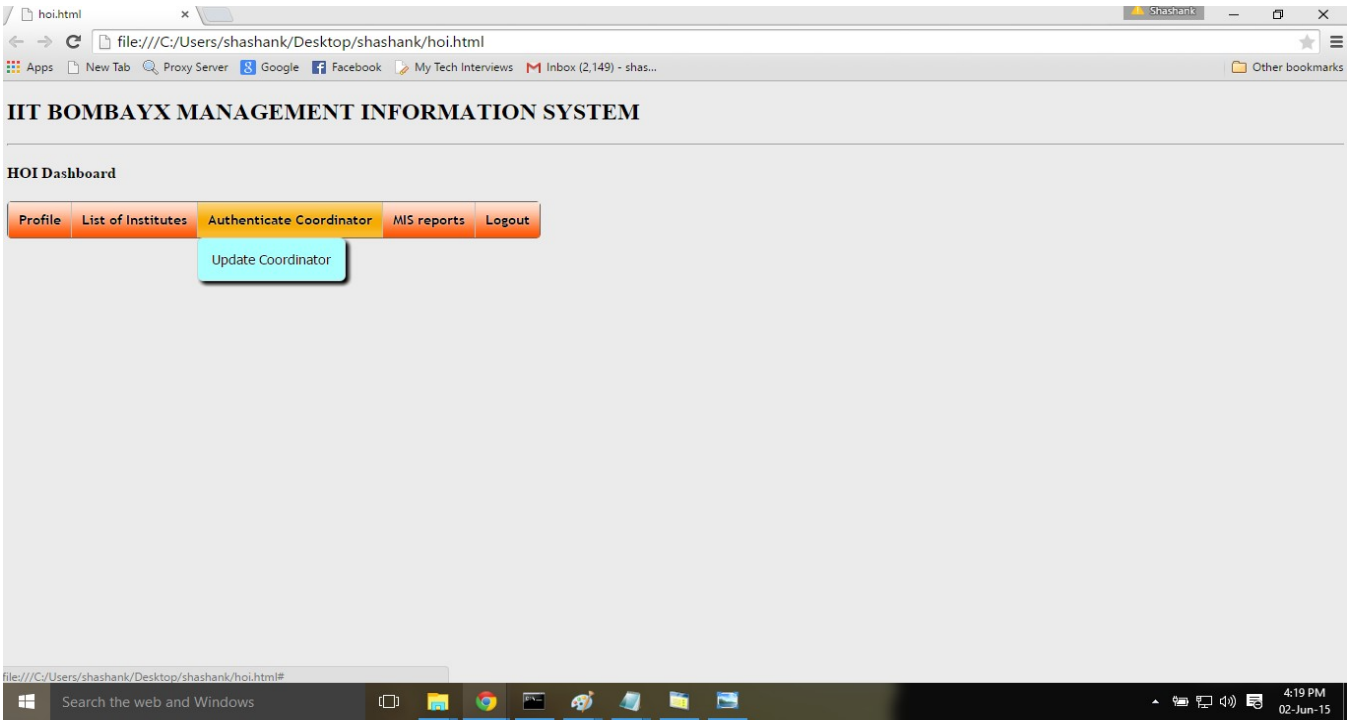
5. Dashboard Design

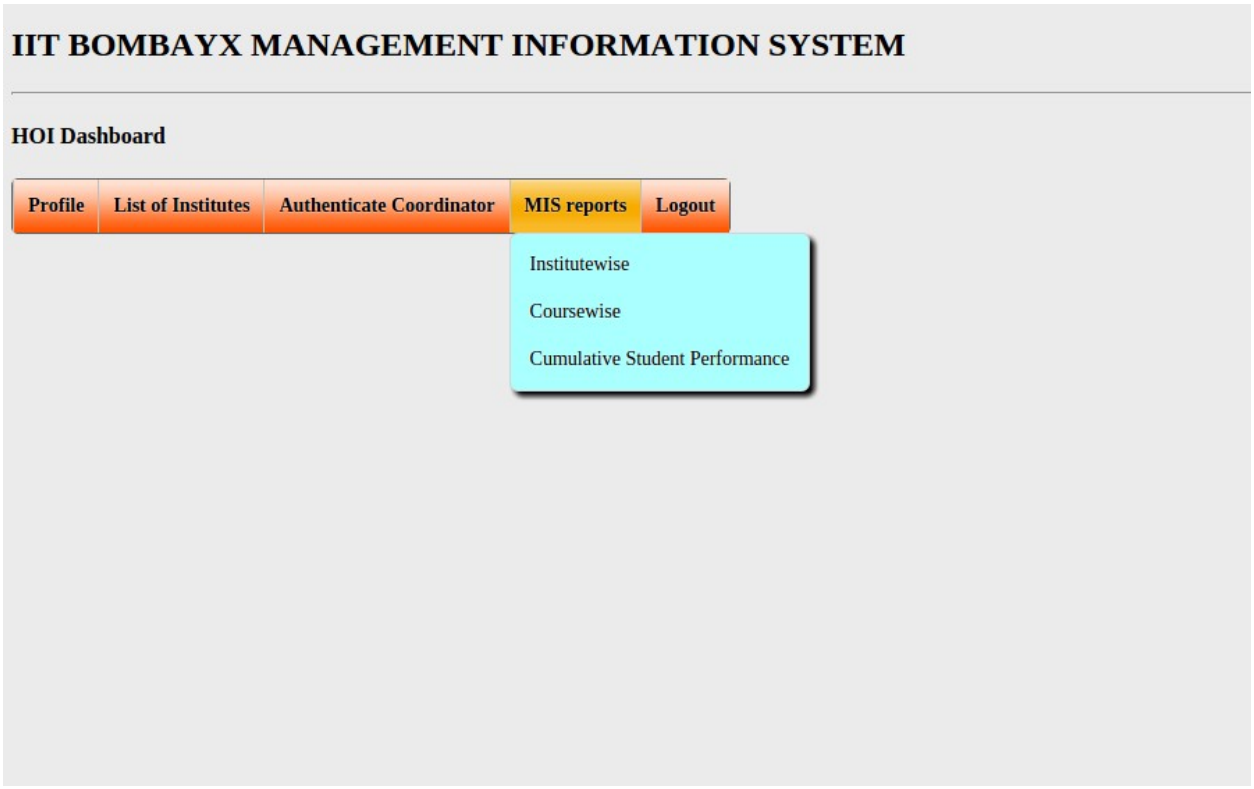
Admin Dashboard



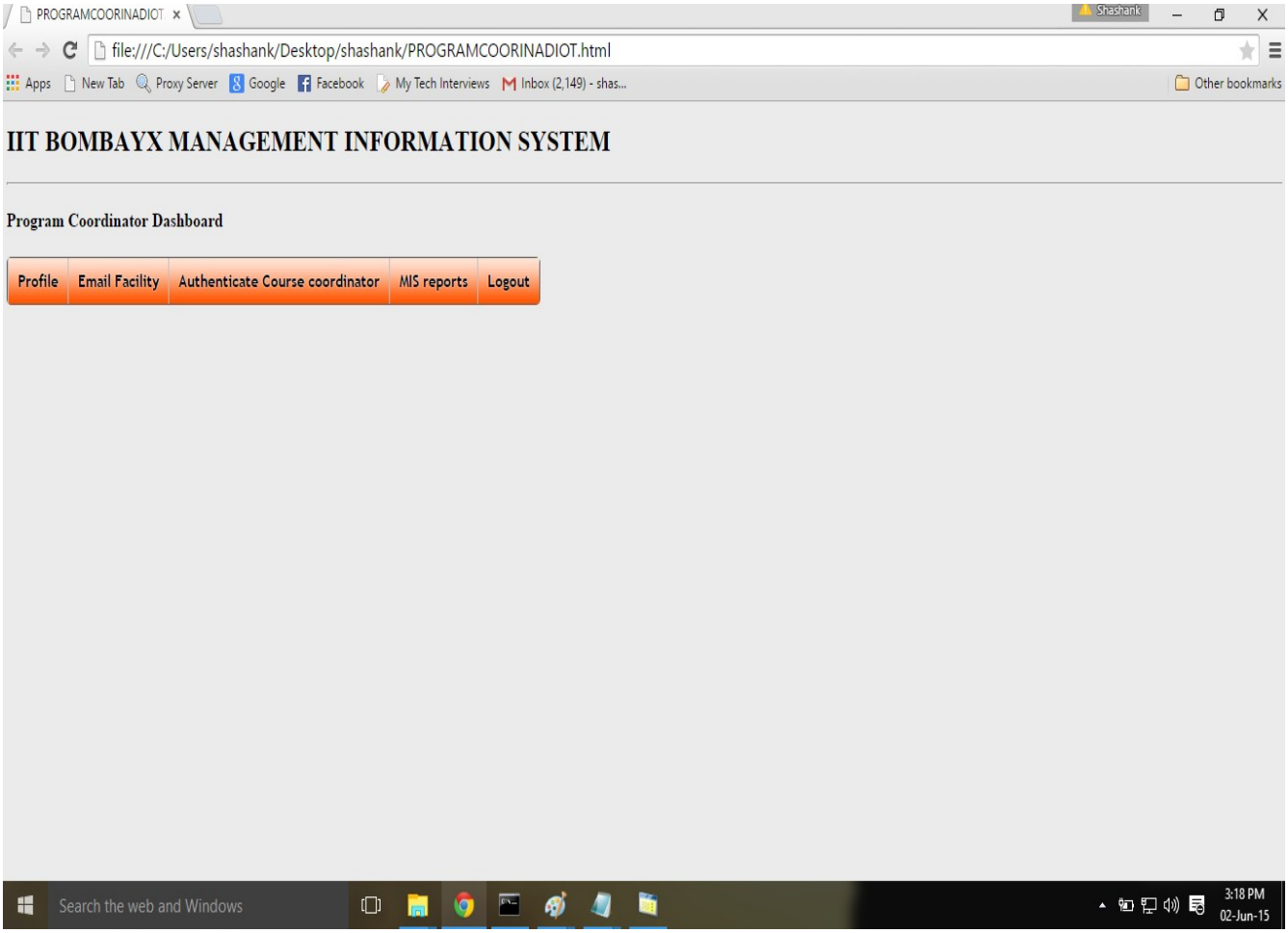
HOI Dashboard

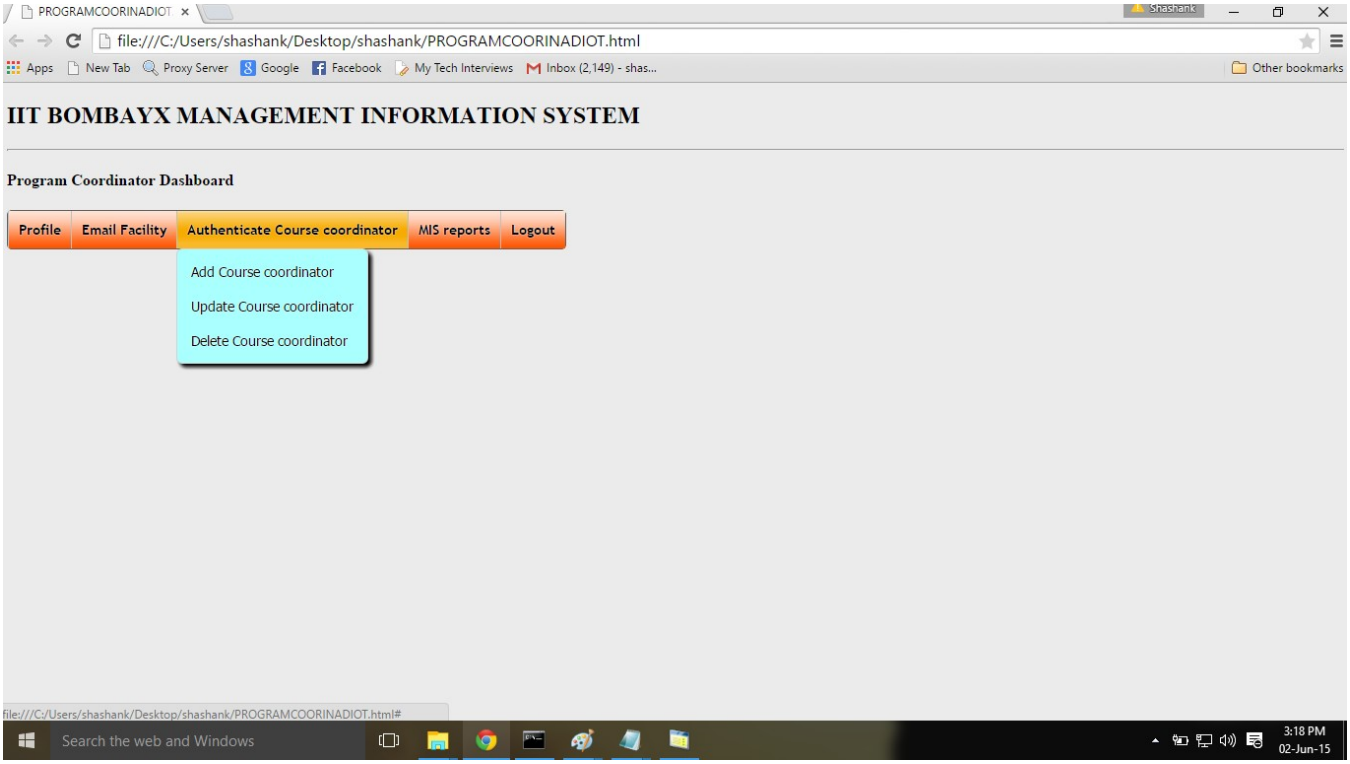


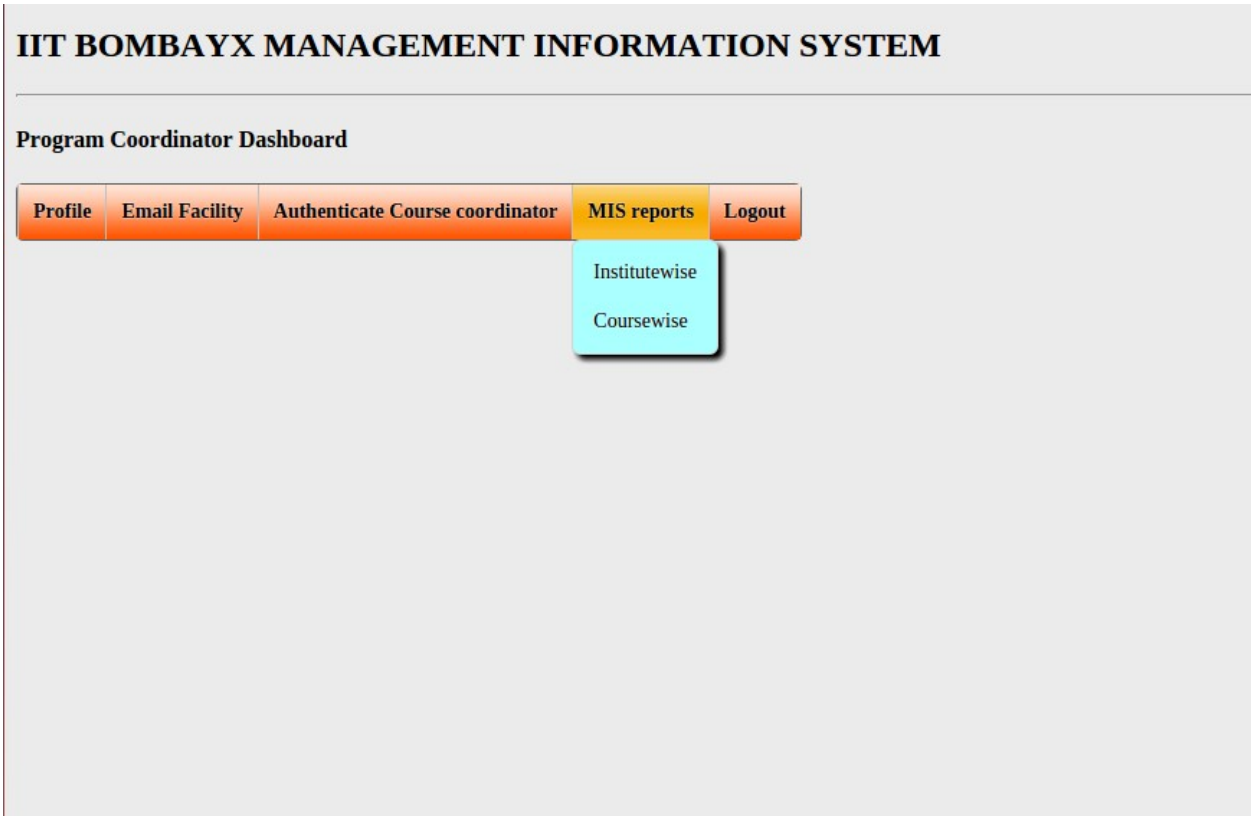



IIT BOMBAYX MANAGEMENT INFORMATION SYSTEM
HOI Dashboard
Profile
List of Institutes
Authenticate Coordinator
MIS reports
Logout
Institutewise
Coursewise
Cumulative Student Performance

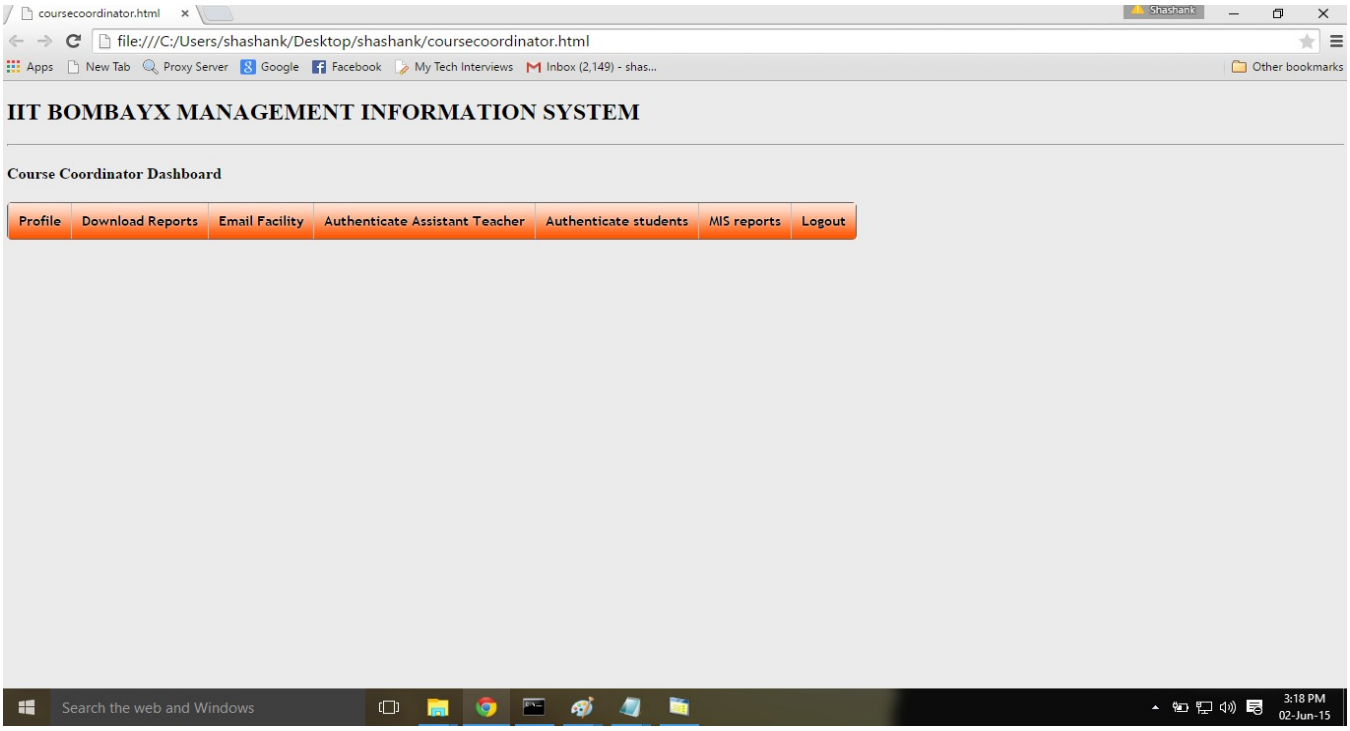
Program Coordinator Dashboard

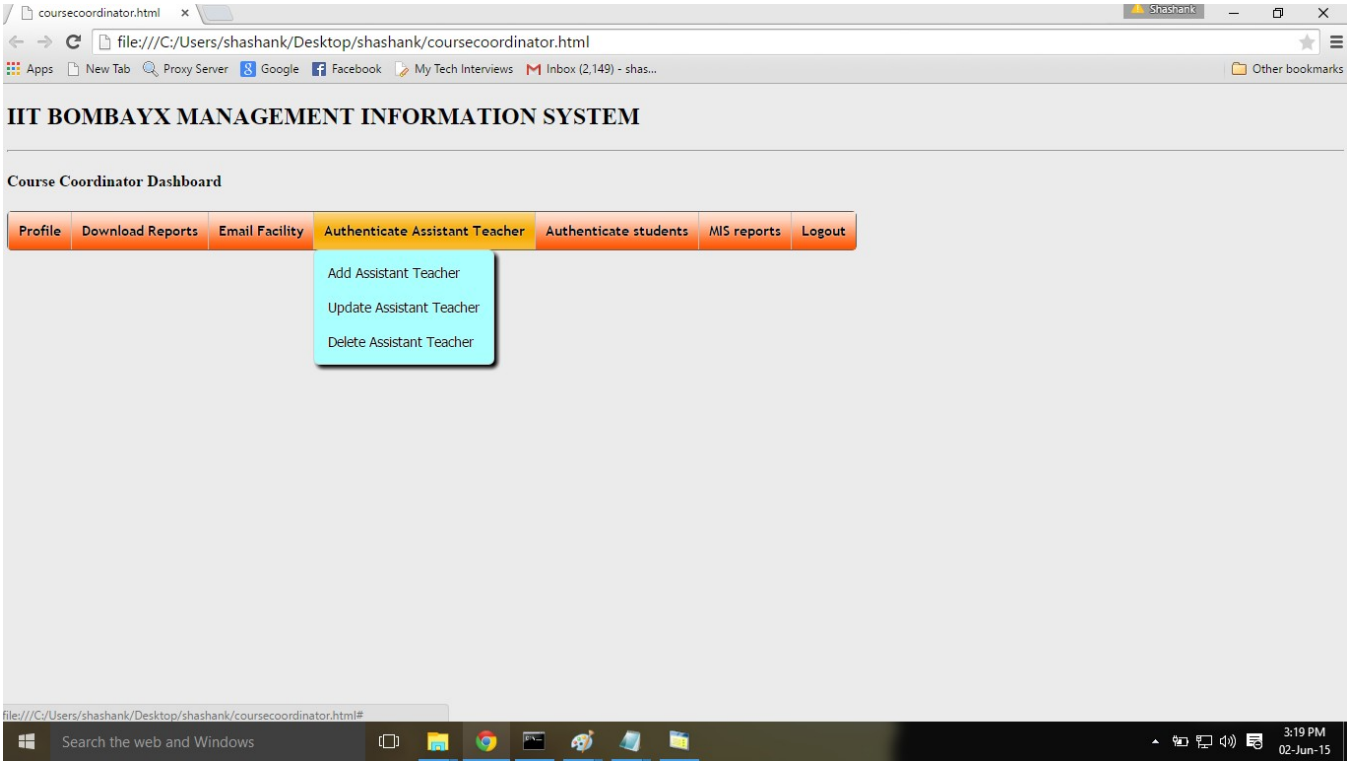


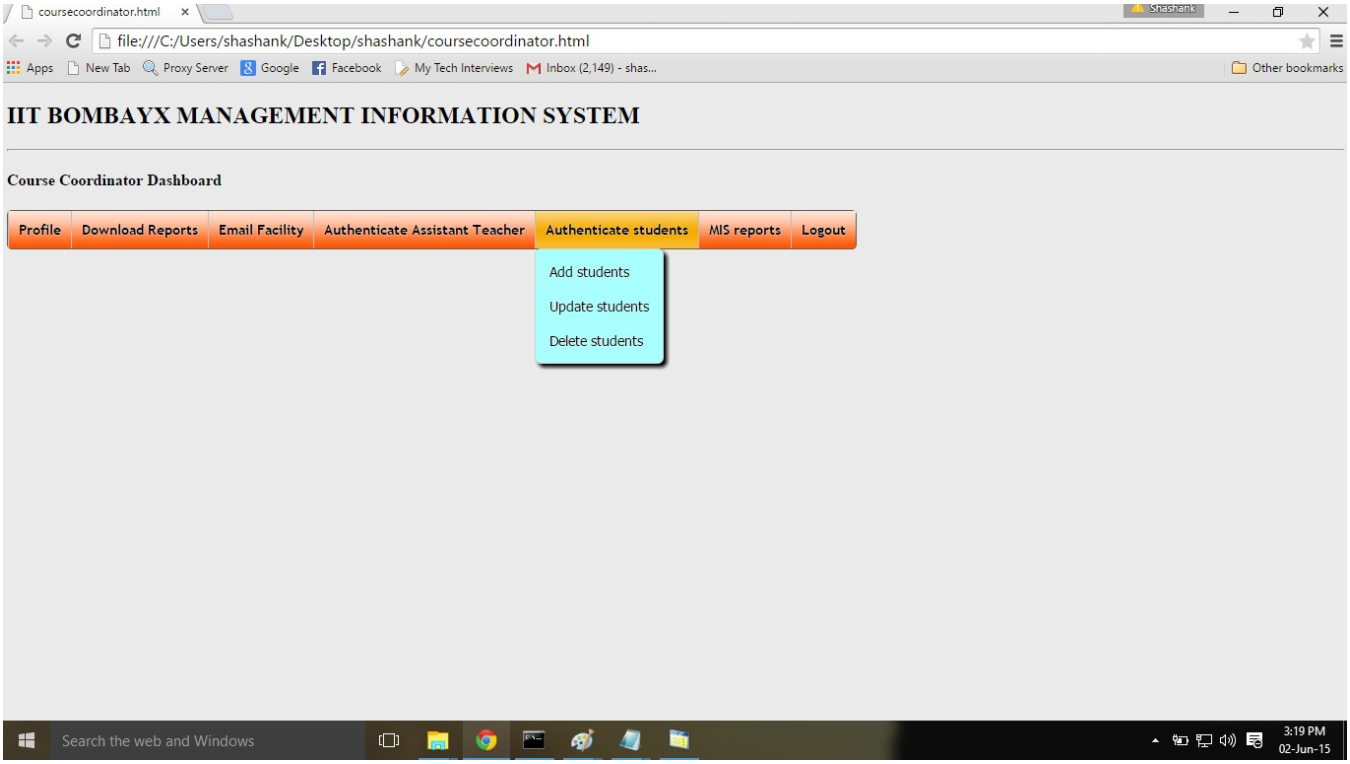

PROGRAMCOORINADIOT
file:///C:/Users/shashank/Desktop/shashank/PROGRAMCOORINADIOT.html
Apps
New Tab
Proxy Server
Google
Facebook
My Tech Interviews
Inbox (2,149) - shas...
Other bookmarks
IIT BOMBAYX MANAGEMENT INFORMATION SYSTEM
Program Coordinator Dashboard
Profile
Email Facility
Authenticate Course coordinator
MIS reports
Logout
Add Course coordinator
Update Course coordinator
Delete Course coordinator
file:///C:/Users/shashank/Desktop/shashank/PROGRAMCOORINADIOT.html#
Search the web and Windows
3:18 PM
02-Jun-15


IIT BOMBAYX MANAGEMENT INFORMATION SYSTEM
Program Coordinator Dashboard
Profile
Email Facility
Authenticate Course coordinator
MIS reports
Logout
Institutewise
Coursewise

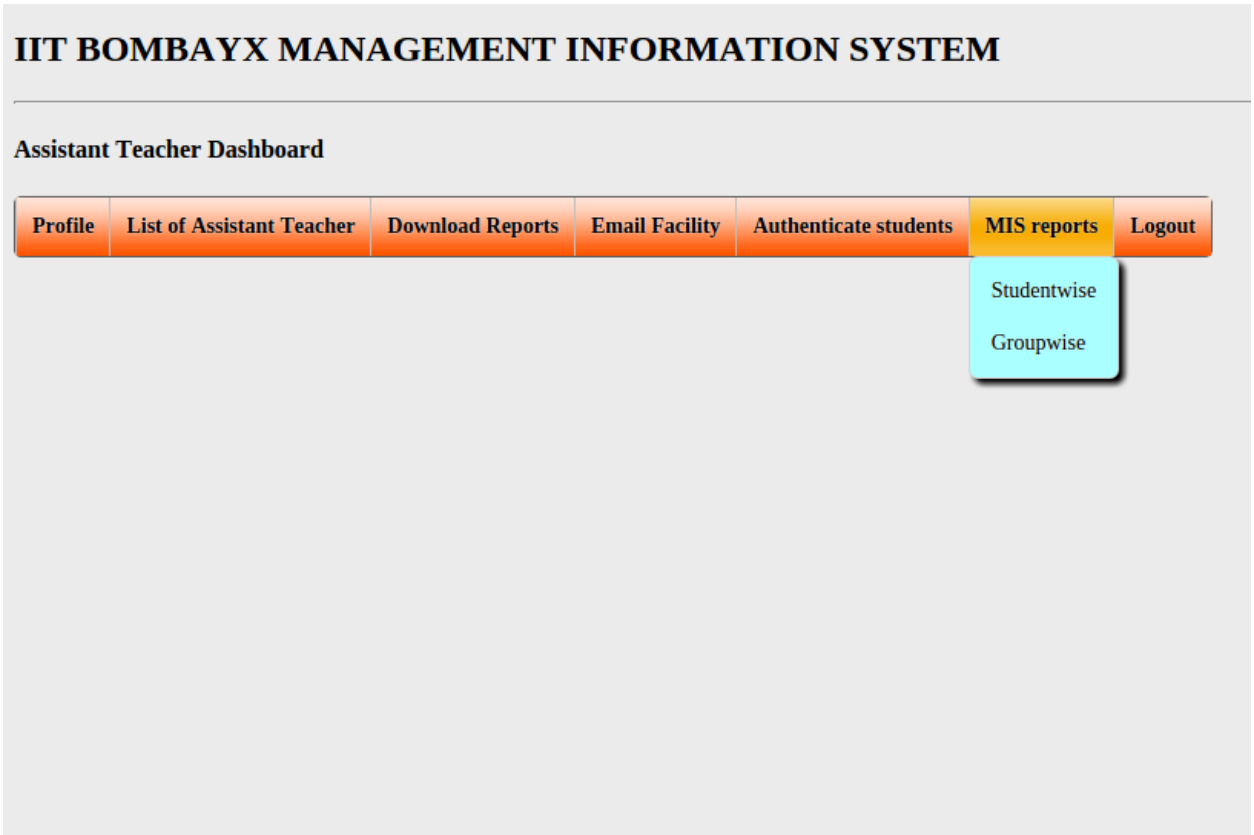
Course Coordinator Dashboard

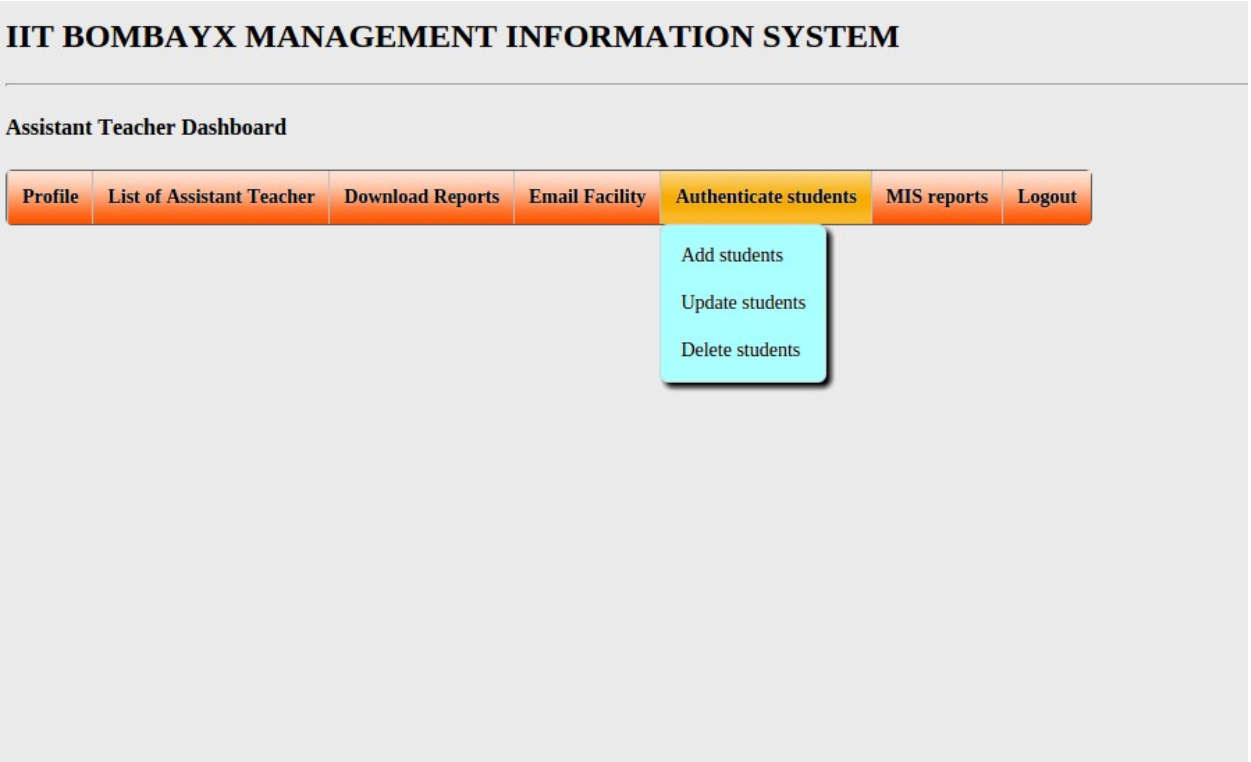




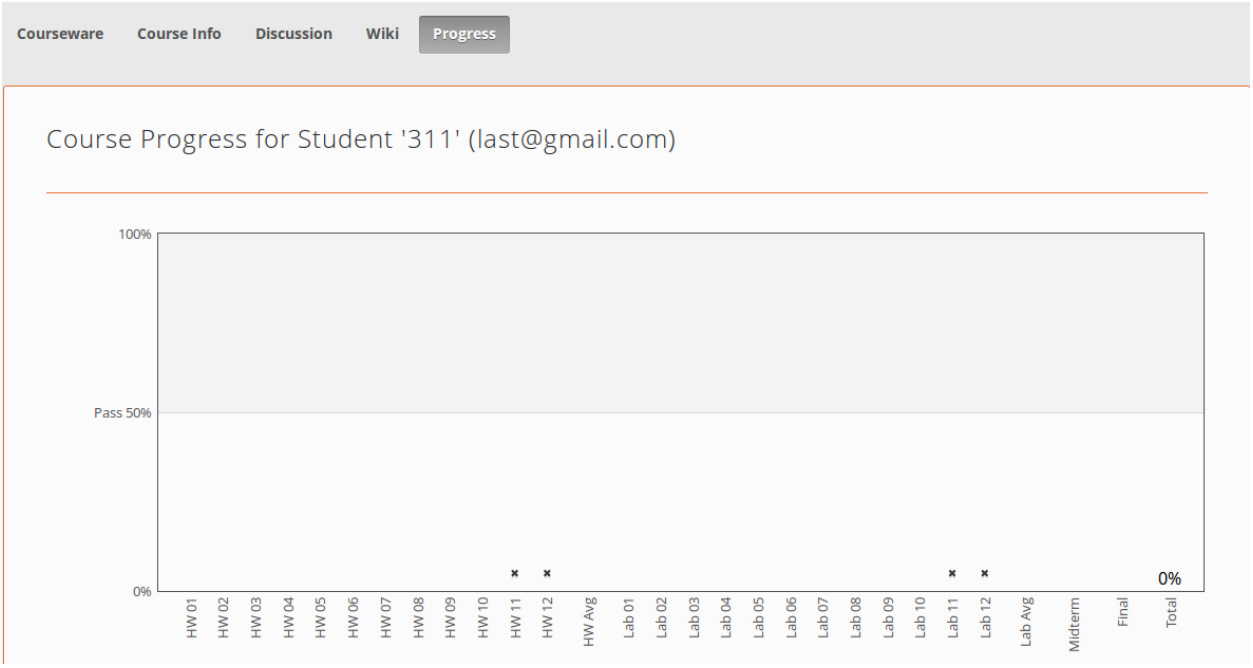


Associate Teacher Dashboard





Student Dashboard



Conclusion:

The above database design may satisfy the various requirement of MIS reports to display the required contents on dashboard of various users. Through this database design we have tried to present an overview that is reliable, easy to implement and expandable in the future.