

**REPORT SPECIFICATION FOR IITBOMBAYX**  
ARINJOY BASAK, DEWANG PALAV, SAGAR AGARWAL,  
EKTA SHARMA, ASHWANI PANDEY, APOORVA AGARWAL, DEEPAK GUPTA

1ST JUNE, 2015

**CONTENTS**

|   |           |
|---|-----------|
| <b><u>PURPOSE OF DOCUMENT</u></b>                                 | <b>2</b>  |
| <b><u>DESCRIPTION OF IDEA</u></b>                                 | <b>2</b>  |
| <b><u>ADDITIONAL PROBLEM STATEMENTS</u></b>                       | <b>2</b>  |
| <b><u>TABLES</u></b>  | <b>3</b>  |
| <b><u>THE USE CASES</u></b>                                       | <b>9</b>  |
| <b><u>THE STUDENT REGISTRATION AND IDENTIFICATION PROBLEM</u></b> | <b>29</b> |

## **PURPOSE OF DOCUMENT**

- To provide complete and concise description about Blended MOOC by IITBombayX.
- To provide the nature of project and product to be delivered clearly.
- To specify various problems and their remedies in practical implementation.
- Finally to convert the idea into a prototype and further into working model.

## **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 about syllabus, 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 limitation that they have no alternative for practical sessions of laboratories.

To overcome these issues we have overcome 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 into group discussions and problem solving. And whenever a lab session is required, it is conducted in the institute itself.

## **AIMS AND OBJECTIVES**

- To enhance better education worldwide.
- To provide easily accessible education and that too free of cost.
- To bring a change in the conventional system of learning and integrate classroom learning with online learning.
- To augment productivity of students by scientifically designed courses and planned activities undertaken by mutual agreement between IITBombayX and classroom teachers.
- To help students realize their caliber and strong and weak zones by data analysis of student's activities.

## **SOME STATISTICS ON MOOCS**

- Over 6.7 million students were taking at least one MOOC course by the end of year 2011.
- Only 2.6 % institutes have a MOOC, another 9.4% report MOOC in planning stages.
- 71% academic leaders rate that MOOCs are same or superior to the classroom teaching style.

So we see that the number of students taking MOOC has increased many folds during recent years. Thus it is the most appropriate time to implement blended MOOC.

## **ADDITIONAL PROBLEM STATEMENTS**

In addition to the system specification presented, this document also addresses two more problem statements which are not addressed by the existing OpenEdx platform. These problems are especially important in the context of the modern Blended MOOCs system to be implemented through IITBombayX, which will include all of the institutes all over India.

1. We have different roles of users in the platform, such as student, coordinator of course, director of institutes, and administrator of IITBombayX. The existing platform does not cover the representation and use of such a hierarchical system through its databases. In our description of the system, we propose a solution to this problem by introducing two new tables and linking them with the existing tables in the system. This is described in the section on databases used.

2. As per the existing layout of the system, a registered student is not required to fill in the details of his/her institute, since IITBombayX must be available to everyone. However, for a student who will be attending this course as part of the Blended MOOCs model, the coordinator who will want the students to enroll in the course, he must be able to identify the students who are part of their college in the local level. A solution is also proposed to this problem as part of this specification document.

## **TABLES:**

The following are the tables which are included in our system.

### **auth\_user:**

This is the primary authorization table, which has the record of identification fields, entered at time of registration and works as the source for unique data (user\_id, username). Queries access this table to identify individual users.

| Attribute | Type    |
|-----------|---------|
| user_Id   | integer |
| username  | varchar |

The table columns are described as follows:

user\_id : This id is generated as a unique key at the time of registration and is generated for all users irrespective of their role. This id will be used for referencing all other information about user . When user will login this id will be used for getting its complete information .

username : The name of the user of IIT Bombay. This will be the handle which will appear on his dashboard. This will not be visible to any other user . This will be asked from user at time of registration .

These are the primary attributes of the tables that we will be using in our specification. This table will include other details of students.

### **Institute\_table:**

Contains details of the institutes as provided at the time of registration of the institute by the head of the institute.

| Attribute   | Type    |
|-------------|---------|
| name        | Varchar |
| instiute_id | integer |
| state       | varchar |
| city        | varchar |

|         |         |
|---------|---------|
| pincode | integer |
| address | text    |
| website | varchar |

The table columns are described as follows:

**name:** Name of the participating institute. The institute head while registering his/her institute keys in the institute name which then gets updated in the database and an unique college id is generated

**institute\_id:** Auto-generated id for the institute. This institute\_id maps to the college name entered and henceforth is used to associate nodes of hierarchy (students, teachers,etc.) with the particular institute.

**state:** State of the Institute, is from a predefined list of available states and is repeated as the institutes vary in number per state.

**city:** City where the institute is located.

**pin code:** Stores the pin code of the location of the institute. The pin code provides a fairly unique field to an institute and is important for communication and identification purposes.

**address:** Detailed address, gets associated to the user who selects the institute at the beginning.

**website:** Website of the institute, is used to have a communication platform and verify the details provided.

#### **institute\_student:**

The table has activity details of the student mapped to a particular institute. This includes the duration for which the student was active and hence change the permissions to take graded quizzes and also a linking field of associate teacher id to identify the teacher allotted locally to the student.

| <b>Attribute</b>     | <b>Type</b> |
|----------------------|-------------|
| user_id              | integer     |
| institute_id         | integer     |
| course_id            | integer     |
| active_from          | date        |
| active_upto          | date        |
| associate_teacher_id | int         |

The table columns are described as follows:

**user\_id:** Id of user .This will be the foreign key referring to auth\_user(user\_id) .

**institute\_id** :Institute id of user's institute . It is a Foreign Key referring to Institute\_id of Institute\_registration .

**course\_id** : Course id of the course in which user is enrolled .

**active\_from**: Date from which user is enrolled in course .

**active\_to**: Date upto which course is completed .

**associate\_teacher\_id**: Id of immediate level of supervision . This **associate\_teacher\_id** will be a foreign key referring to **user\_id** in **auth\_user** table . This will be used to identify hierarchy level in table .

#### **institute\_centraluser:**

This table contains information about the central user of the institute . Central user will be the one who will be responsible for IIT BOMBAYX course in the particular institute .

This will communicate to IIT BOMBAYX admin on behalf of the institute. He will be the representative of the institute.

| Attribute    | Type    |
|--------------|---------|
| user_id      | integer |
| institute_id | int     |
| role_id      | int     |
| course_id    | int     |

The table columns are described as follows:

**user\_id** : Id of central user .It is foreign key referring to **user\_id** of **auth\_user**.

**institute\_id** : Institute id of corresponding institute . It is foreign key referring to **institute\_id** of **institute\_table** .

**role\_id** : Role id of user. Foreign Key referring to **role\_id** of **roles** table .This will be helpful to identify his role as student , coordinator(course-in-charge), director(head of institute), admin(at IIT BombayX ) .

**course\_id** : Course id of course being monitored by user.

#### **course\_quizzes(mongo):**

Information about course quizzes will be stored in mongo. It stores informations like type of quiz, number of quizzes, the course for which these quizzes are and many other relevant information.

| Attribute  | Type    |
|------------|---------|
| quiz_type  | String  |
| quiz_id    | Integer |
| quiz_count | Integer |
| course_id  | Integer |

The table columns are described as follows:

quiz\_type : The type of the quiz, whether the quiz is graded or ungraded, or whether the quiz is weekly or final.

quiz\_id : Id for the quiz

quiz\_count : The number of participants in the quiz

course\_id : The unique id of the course.

The 'hierarchy' table:

Under the IITBombayX Blended MOOCs scheme, we have multiple students participating in a course who are supervised by a single course coordinator. The coordinators activities and the performances of the students under them, in turn are supervised by the Director or Head of the Institute. In order to determine this hierarchy and represent it in the databases, we propose the inclusion of an additional table denoted by 'hierarchy' in the existing database of IITBombayX.

The attributes of the table are defined as follows:

| Attribute | Type              |
|-----------|-------------------|
| u_id      | integer           |
| g_id      | integer           |
| sub_id    | integer           |
| super_id  | integer           |
| role      | Integer (TINYINT) |

The fields in the diagram indicate the following table columns in the Hierarchy Table scheme:

u\_id : The user id of any user, which is actually the 'id' field in the auth\_user table for any registered user. As defined, it is a unique field.

g\_id : It is a generic field which indicates the section or group number for a group or collection of entities at a lower level that are being governed by or supervised by a higher level entity. This field is a

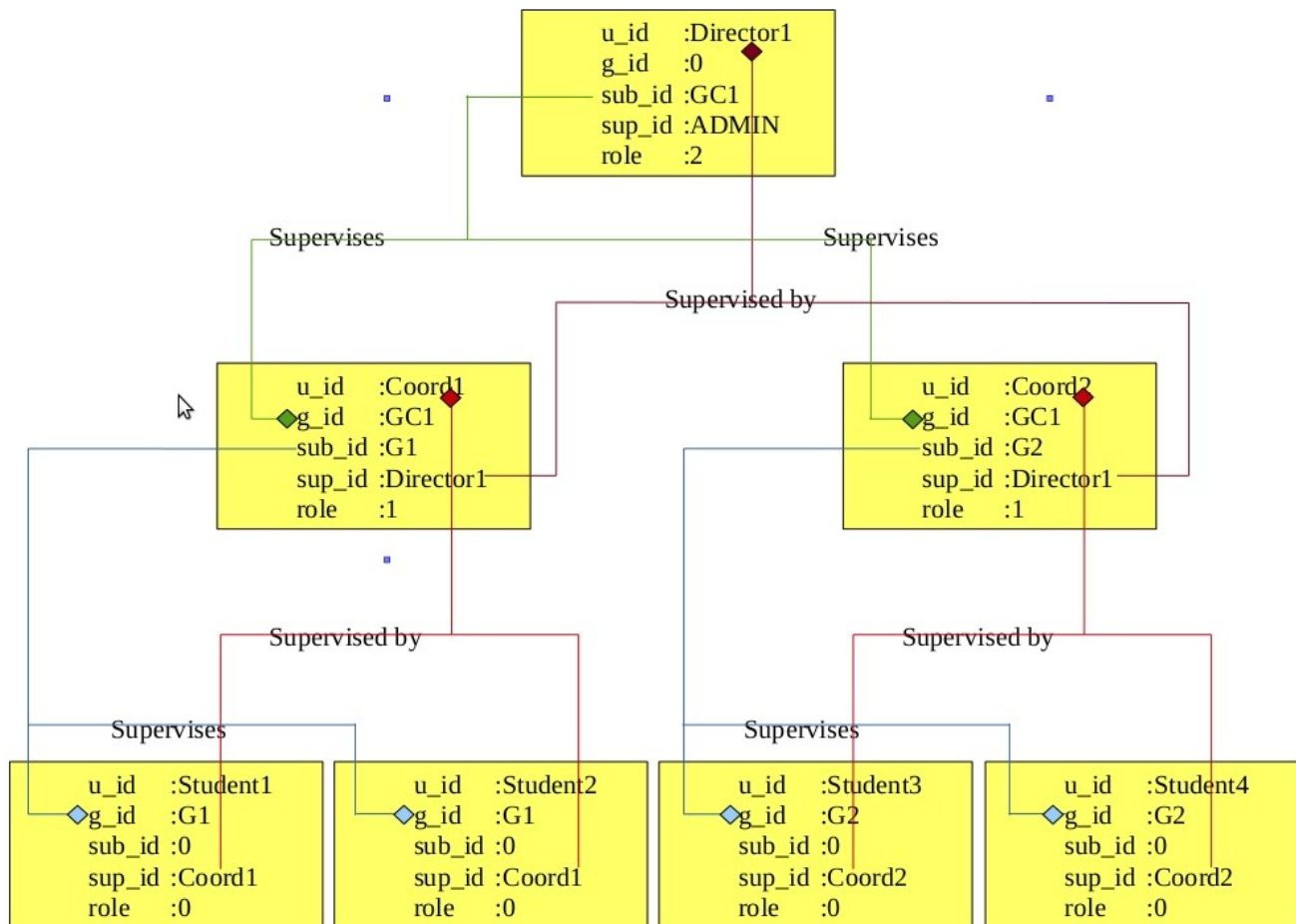
unique field. It is zero (or null) for an entity that has not yet been collected into a section.

sub\_id : For a higher level entity (higher value of level than 0), this indicates the g\_id (group\_id) for the section that is being supervised by the member. For the student, since he will not be supervising any other members, the value of this field will be 0. It will also be zero if an entity of member does not supervise a section or group of lower level entities (say, a coordinator is not being managing a course being taken by a group of students).

super\_id : For a lower level entity (0 or other values of level field), this indicates the u\_id of the member that supervises that member. For the highest level member for an organisation (in this case, the director), the value of this field will be zero. It will also be zero if the entity is not being managed by a higher level entity (say, a student is not being managed by a coordinator yet). By default, this will be set to the u\_id of the ADMIN, because the administrator can access an the entity, if he/she is not being supervised.

role : This is a foreign key to the role\_table, which indicates numerically the roles played by an entity in the heirarchy. This would indicate to the user the roles played by entities in adjacent levels in the hierarchy (say, relation between student and coordinator, and between coordinator and director).

An example of the hierarchy in an institute could be as follows.



The 'role' table:

In order to specify the roles played by different users, who are otherwise indistinguishable based on their user id-s, we propose the creation of an additional table called the 'role table'. The table ultimately associates the roles of the different users of the Blended MOOC system (student, coordinator, and director of the institute) together using numbers. The table is actually proposed as an extension to the existing institute\_role table in IITBombayX, with an aim to replace it while keeping the functionality same.

| Attribute | Type              |
|-----------|-------------------|
| role      | Integer (TINYINT) |
| role_name | varchar           |

The table columns are described as follows:

**role** : This is an integer field, which is only an integer code used to specify the roles of the users in the system. It starts from 0 (indicating student) and can accommodate any level of hierarchy desired. Whenever a new level of hierarchy is added (intermediate or otherwise), we add the entry to this table, and a new value is assigned automatically to the level/role added.

**role\_name** : This a character field, which stores the name of the role corresponding to the integer value, as a string of characters.

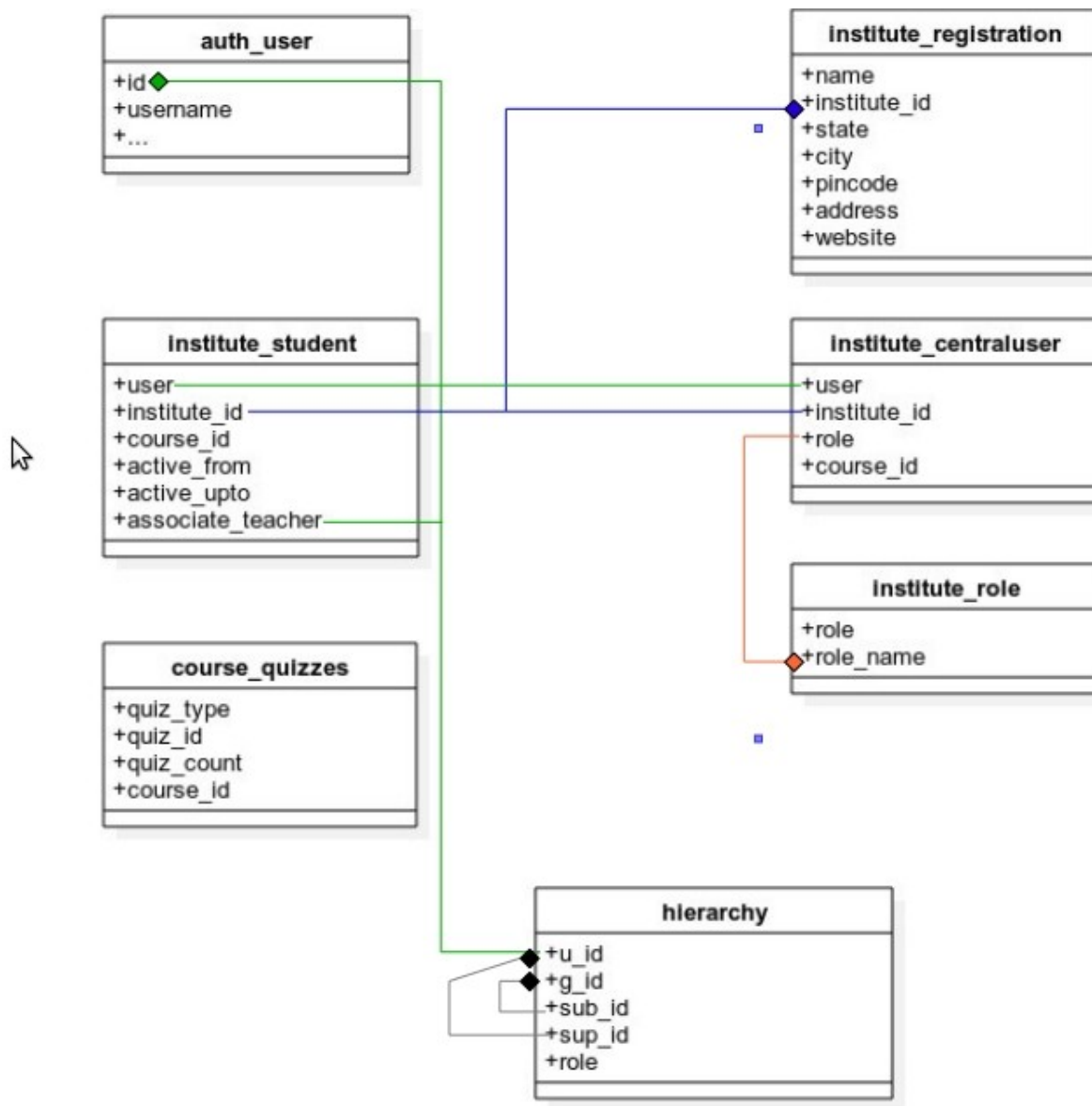
Currently, since we consider a four tier hierarchy, we have the following entries for the insitute\_role table:

| role | role_name                        |
|------|----------------------------------|
| 0    | student                          |
| 1    | coordinator (course in-charge)   |
| 2    | director (head of the institute) |
| 3    | ADMIN (at IITBombayX)            |

It should be noted here that the ADMIN is a special role, which does not indicate a single person or enity, but a host of persons at IITBombayX who are in charge of managing the courses, designing and arranging contents and quizzes, and of course, the administrator who oversees the system performance and troubleshooting.

The following diagram gives a summary description of the tables which are included in the system, together with the linking of the tables via the attributes, which indicate a primary key-foreign key relationship.





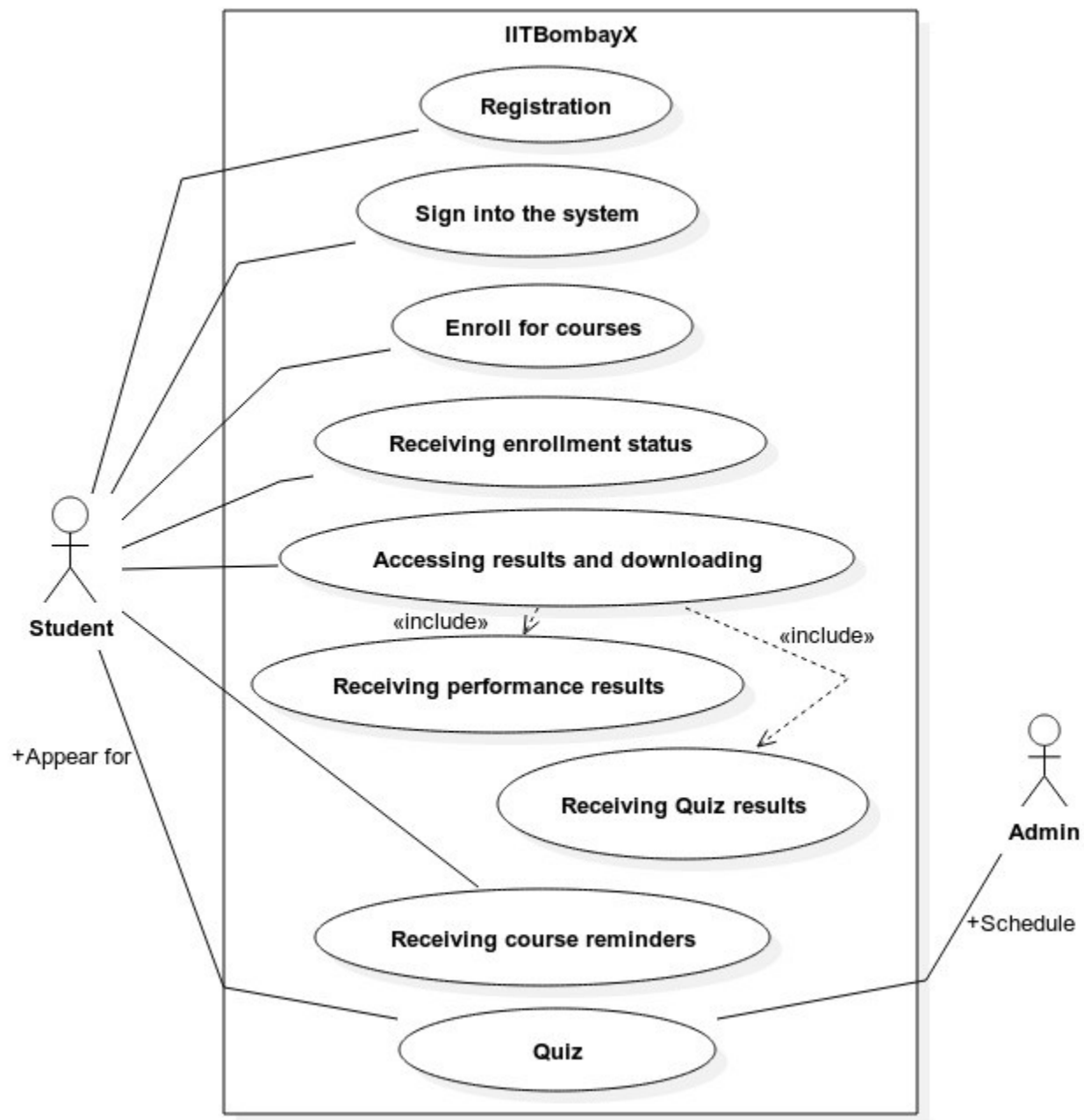
—◆ This relation indicates that the attribute is a foreign key to/ references the target attribute (pointed to by the diamond).

## THE USE CASES

The system behavior in terms of how the users interact with the system is given by the use cases for the system. In the following pages, we describe the use cases for our proposed system.

Use-case Descriptions:

Student:



Student's use cases

1)Registration:

Use-case:Registration

Purpose:Profile of the student is created for accessing the system

Actors:Student

Precondition: None.

#### Basic Flow:

1. The student accesses the IITBombayX website, and clicks on 'Create Account' link.
2. He is presented with a page which asks him to fill up his personal details. These fields include mandatory and not mandatory fields.
3. The student provides the details which include but are not limited to: his full name, the name of his institute, his date of birth, his address, his email id, whether he is a member of some college or not (and therefore, whether, he is an ordinary user or a Blended MOOC participant), and finally his password, which he must confirm.
4. Once he has successfully filled in the details, he clicks on Submit button, and receives a confirmation mail, containing a link to activate his account. This link is valid for only a limited period of time.
5. Once he activates his account, the user account is successfully created, and the user can go on to access the services of the platform.

#### Exception Flow:

- The student does not fill the mandatory fields, or fills the fields with incorrect or inappropriate values.
  1. The submit button will not be activated, and the validation script running on the front end will highlight the corresponding fields.
  2. A corresponding error text will tell the user what the error is:
    1. If the field is not filled, it says - "Please fill the mandatory field"
    2. If a numeric field is filled with text data, it says - "Please input correct data".
    3. If the password is not strong, then it highlights the field, saying that 'The password is not strong', together with the required specifications for the passwords.
    4. The 'repeat password' field does not match the 'input password' field, then it clears the fields, and highlights both the fields, saying 'The passwords do not match'. He needs retype the password, until he is successful.

Postcondition: Student gets a user name and a password for his profile for accessing the system.

#### 2)Enroll for courses:

Use-case: Enroll for courses

Purpose: Student can complete the courses that he applies for as part of his college curriculum, or for his won benefits, and may apply for certificate of performance and completion for the same.

Actors: Student

Precondition: The student needs to have an existing account in the IITBombayX.

#### Basic Flow:

1. The student logs into his account, and searches for the course that he wants to enroll in, using the 'course search'/'list available courses' facility in the website.
2. Once he has found the course he wants to apply for, he enrolls for it by clicking the 'enroll' button, and waits for the enrollment confirmation mail.

#### Exception Flow:

- If the student has not done the registration, then he cannot enroll for the courses.
  1. The student will then be taken to the Registration page, where he must first register himself for an account on IITBombayX.
  2. Later, he can sign into his account, and come back on the page for the course, and enroll himself.

Postcondition: The student will have placed a request to be enrolled into the course of his choice, and will have received an email corresponding to confirmation or otherwise in his inbox.

### 3)Login:

Use-case :Login

Purpose : Getting access to the account.

Actors : Student

Precondition: The student has an existing account on IITBombayX.

Basic Flow:

1. The student need to type in his email-id and password and click on the login button.
2. After clicking on the login button, the student is directed to either his account page, or the last page that he was on, depending upon where he was when he clicked on the “log in” button. If he was on the homepage of the website, he will be taken to his account page-cum-dashboard. If he was browsing some or other course pages, then he will be taken back to the course page, but this time, with access to his account.

Exception Flow:

1. If the student enters invalid email-id or password, there will be a message saying “Invalid email-id or password”.
2. If the student enters email-id and password but he does not have an existing account, there will be a message saying “Account of the email-id does not exist”. The user will then be prompted with the option of registering himself on the website.

Postcondition: The student will be taken to the last page he was on, or his account page, after granting access to his account.

### 4)Receiving enrollment status:

Use-case:Receiving enrollment status

Purpose:To confirm the enrollment of the student for a course.

Actors:Student

Precondition:

1. The student needs to be logged into his account on IITBombayX.
2. The student must have requested for enrollment into a course of his choice.

Basic Flow:

1. If the student has applied for the course, and the limit has not yet reached the maximum number of students for a course, and the course is online for the time, then he will receive a

confirmation mail stating that he has been enrolled for the course, and can start at the date specified.

Exception Flow:

1. If there is the problem or the system failure during the enrollment of the course, the enrollment status will not be received by the student.
2. If the number of students enrolled for a course reaches the upper limit, or for some reason, the course is take offline, the student will be notified regarding the same through the email with which he as registered. This will count as an unsuccessful enrollment.

Postcondition: Successfully enrolled for the course.

5)Accessing results and downloading:

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 goes to the dashboard, and finds the tab/link indicating 'Result'.
2. He is then presented with the options for all the MOOCs he is currently taking, and must select one of them in order to determine which course results or performance he wants to see.
3. The student selects the course by clicking on the appropriate link, and is then presented with a page which gives his overall grade acquired so far in the course, along with links for the quizzes and graded/ungraded exercises he has completed. This can be downloaded as a pdf transcript, but only at the end of the course, when the coordinator issues it for the students.
4. If the student clicks on any of the quizzes that he has taken so far, and all of them are past the due date, then he can view the scores he had secured for that quiz. He may also review the quiz answers, provided the coordinator has granted access to it.

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 and admin (as required), will be able to download a summary report of his performance (a transcript of sorts).

## 6)Receiving course reminders:

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.

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

## 7)Appearing for Quiz:

Use-case:Appearing for Quiz

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

Actors:Student, Admin

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 'start quiz' link/button.
3. He is taken to a new page which tells him the rules and regulations for the quiz. He must click on the '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 the answers, 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 associated with the lessons at the time of practicing.

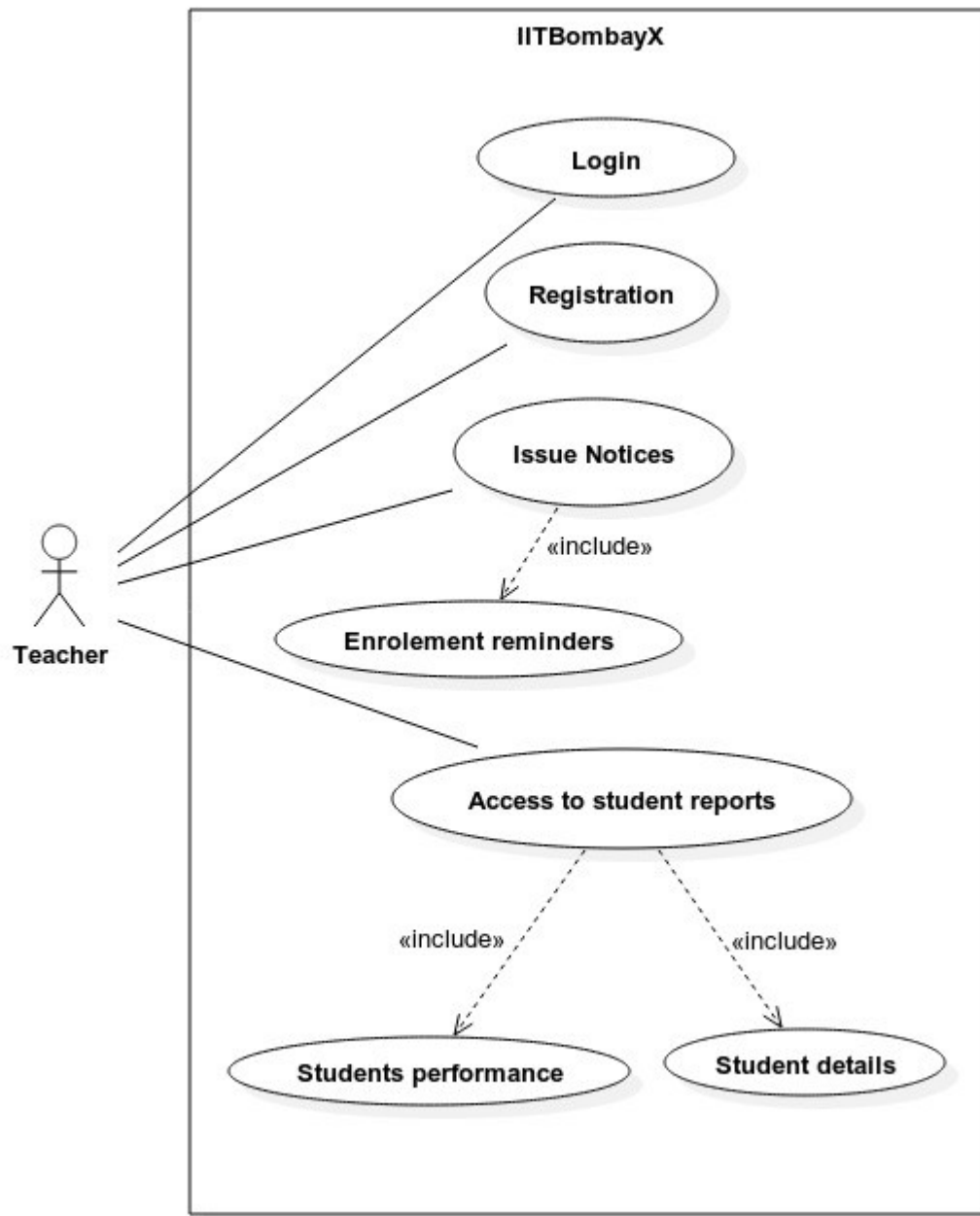
#### Exception Flow:

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.

#### Postcondition:

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.

Coordinator:



Coordinator's use cases

1)Login:

Use-case:Login

Purpose: Getting access to the account.

Actors:Coordinator

Precondition: The Coordinator need to have an account in the IITBombayX.



#### Basic Flow:

1. The coordinator need to type in his email-id and password and click on the login button.
2. After clicking on the login button, the student is directed to either his account page, or the last page that he was on, depending upon where he was when he clicked on the “log in” button. If he was on the homepage of the website, he will be taken to his account page-cum-dashboard. If he was browsing some or other course pages, then he will be taken back to the course page, but this time, with access to his account.

#### Exception Flow:

1. If the coordinator enters invalid email-id or password, there will be a message saying “Invalid email-id or password”.
2. If the coordinator enters email-id and password but he does not have an existing account, there will be a message saying “Account of the email-id does not exist”. The user will then be prompted with the option of registering himself on the website.

Postcondition: The coordinator will get access to the dashboard.

#### 2)Registration:

Use-case:Registration

Purpose:Profile of the coordinator is created for accessing the system

Actors:Coordinator

Precondition: None.

#### Basic Flow:

1. The coordinator accesses the IITBombayX website, and clicks on 'Create Account' link.
2. He is presented with a page which asks him to fill up his personal details. These fields include mandatory and not mandatory fields.
3. The coordinator provides the details which include but are not limited to: his full name, the name of his institute, his date of birth, his address, his email id, his role(a coordinator), and finally his password, which he must confirm.
4. Once he has successfully filled in the details, he clicks on Submit button, and receives a confirmation mail, containing a link to activate his account. This link is valid for only a limited period of time.
5. Once he activates his account, the user account is successfully created, and the user can go on to access the services of the platform.

#### Exception Flow:

- The coordinator does not fill the mandatory fields, or fills the fields with incorrect or inappropriate values.
  1. The submit button will not be activated, and the validation script running on the front end will highlight the corresponding fields.
  2. A corresponding error text will tell the user what the error is:
    1. If the field is not filled, it says - “Please fill the mandatory field”
    2. If a numeric field is filled with text data, it says - “Please input correct data”.
    3. If the password is not strong, then it highlights the field, saying that 'The password is not

strong', together with the required specifications for the passwords.

4. The 'repeat password' field does not match the 'input password' field, then it clears the fields, and highlights both the fields, saying 'The passwords do not match'. He needs retype the password, until he is successful.

Postcondition:Coordinator gets a user name and a password for his profile for accessing the system.

### 3)Issue Notices:

Use-case:Issue Notices

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

Actors:Coordinator

Precondition: The coordinator 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 'send notification' 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.
  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.

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

### 4)Access to student reports:

Use-case:Access to student reports

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

Actors:Coordinator

#### Precondition:

1. The coordinator has successfully logged into his account on IITBombayX.
2. The 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 coordinator goes to his dashboard page, where he has links/ buttons for downloading the details of performance in the following categories:

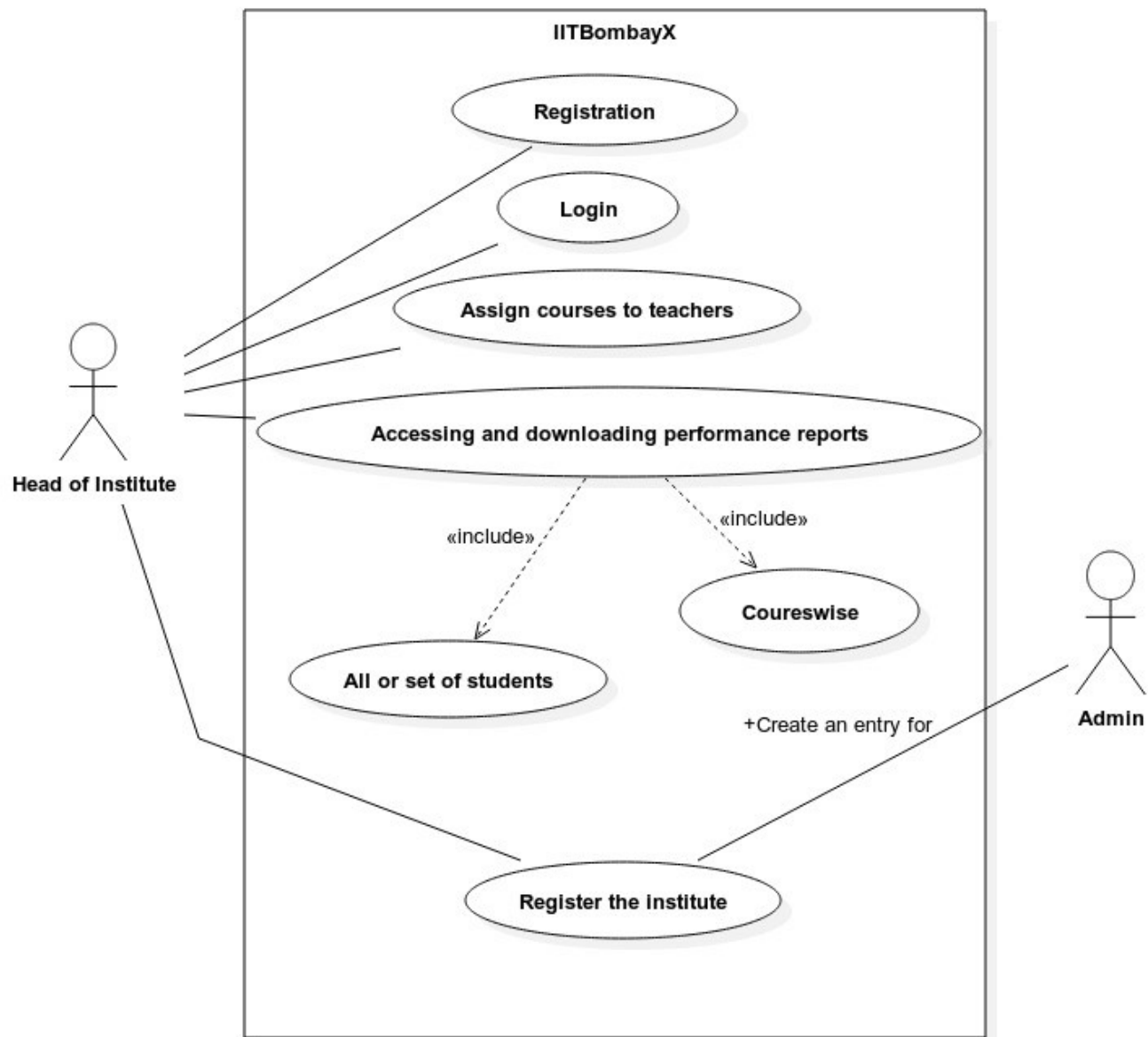
1. All students:
  1. The coordinator can click on this tab/link to go to the page containing the links for all the students in all the courses enrolled under him, divided by courses, or grouped together.
  2. The page contains a tabular display containing the details of all the students and the columns include the union of all the courses they are enrolled in (since all the students are being shown) (eg: if there are students taking courses A and B, and some students taking only course A, then for the latter students, their entry for course A will be NULL).
  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 coordinator.
  5. The 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:
  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 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 generic 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 coordinator.
  5. The 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:
  1. This includes all the analytical information that is provided by the InSight module of the IITBombayX system, to the coordinator 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 coordinators, and the coordinators can further send them to the students in the courses under them.

#### Exception Flow:

1. If the coordinator 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 coordinator 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.

Postcondition: The coordinator had accessed the student report successfully, and downloaded them if required.

Director:



Director's Use Cases

1)Registration:

Use-case:Registration

Purpose: Profile of the director is created for accessing the system

Actors:Director

Precondition: None

Basic Flow:

1. The director accesses the IITBombayX website, and clicks on 'Create Account' link.
2. He is presented with a page which asks him to fill up his personal details. These fields include mandatory and not mandatory fields.
3. Once he has successfully filled in the details, he clicks on Submit button, and the details will go to the administrator.
4. The administrator must receive notification from before hand, in order to authorise the creation of this account. If so, he sends a special confirmation mail, containing a link to activate his account. This link is valid for only a limited period of time.
5. Once he activates his account, the user account is successfully created, and the user can go on to access the services of the platform.

Exception Flow:

- The director does not fill the mandatory fields, or fills the fields with incorrect or inappropriate values.
  1. The submit button will not be activated, and the validation script running on the front end will highlight the corresponding fields.
  2. A corresponding error text will tell the user what the error is:
    1. If the field is not filled, it says - "Please fill the mandatory field"
    2. If a numeric field is filled with text data, it says - "Please input correct data".
    3. If the password is not strong, then it highlights the field, saying that 'The password is not strong', together with the required specifications for the passwords.
    4. The 'repeat password' field does not match the 'input password' field, then it clears the fields, and highlights both the fields, saying 'The passwords do not match'. He needs retype the password, until he is successful.
- If the administrator does not receive the authorisation for the creation of the director account, but he receives the form containing the details of the director, he will NOT authorise the creation of the director account.

Postcondition: director gets a user name and a password for his profile for accessing the system.

2)Login:

Use-case:Login

Purpose: Getting access to the account.

Actors:Head of Institute

Precondition: The director need to have an account in the IITBombayX.

1. Basic Flow: The coordinator need to type his email-Id and password and click on 'Login button'.
2. After clicking on Login button,the director is directed to the dashboard.

Exception Flow:

1. If the director enters the invalid email-Id or password,there will be a message saying “Invalid email-Id or password”.
2. If the coordinator enters email-id and password but he does not have an existing account, there will be a message saying “Account of the email-id does not exist”. The user will then be prompted with the option of registering himself on the website

Postcondition: The director will get access to his dashboard on the system.

3)Assign Courses to teacher:

Use-case:Assign Courses to teacher

Purpose:To distinguish between the allotment of different courses to different Teachers and also, to assign a supervisor for each course.

Actors:Head of Institute

Precondition: The director is logged into the system in his account, and has course coordinators under him, determined by the hierarchy table.

Basic Flow:

1. The director logs into his account and can select the courses from the list of courses available in the IITBombayX system, in the same way as the Student (see student use case).
2. The director then uses a button available on his view of the course page (he has only one button for his view called the 'Apply' button, which his university to enroll students in the course as required) to flag the courses.
3. These courses then can be viewed from his dashboard, where he has a special tab/view for assigning courses, which shows all the coordinators in his university/institute, and all the courses he can govern.
4. In this view, he can select a course, click on an 'add coordinator' link, which gives a temporary section providing the coordinators' names, as well as a search facility to find the coordinators by name.
5. Once he finds the coordinators he wants, he can add them by selcting a check box next to their name in this temporary views.
6. He finishes this by clicking on the Save Changes and Send Notifications button.

Exception Flow:

1. If the director adds a course more than once to his list, he will be warned of this repetition by a message.
2. If he assigns a course twice to the same coordinator, he will be warned of this error.
3. If he assigns too many courses to a coordinator, he will be given a warning but will not be prevented, as this is a matter of the logical infrastructure at the local level of the institute.

Postcondition: The director have successfully assigned the courses to the teachers.

#### 4)Accessing and downloading performance reports:

Use-case:Accessing and downloading performance reports

Purpose:To get a report of all the students or the course wise report of the student

Actors:Head of Institute

Precondition: The director must be logged into his account on the IITBombayX system, and his institute registered.

Basic Flow:

1. The director goes to his dashboard page, where he has links/ buttons for downloading the details of performance in the following categories:
2. All students:
  1. The director can click on this tab/link to go to the page containing the links for all the students in all the courses enrolled under the coordinators under him, divided by courses, coordinators, or grouped together. On clicking, he will be taken to the cooresponding page.
  2. The page contains a tabular display containing the details of all the students and the columns include the union of all the courses they are enrolled in (since all the students are being shown) (eg: if there are students taking courses A and B, and some students taking only course A, then for the latter students, their entry for course A will be NULL).
  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 director.
  5. The director 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. 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 link, the director 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 generic 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 director.
  5. The director 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.
4. Analytical results:
  1. This includes all the analytical information that is provided by the InSight module of the IITBombayX system, to the director 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.
  5. 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 coordinators, and the coordinators can further send them to the students in the courses under them.

Exception Flow:

1. If the director 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 director 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.
3. The director cannot access the final reports of the students before the completion of the courses. If he tries to do so, he will get a message informing him of the same.

Postcondition: Accessing and downloading performance reports ,was successfully done by the director.

5)Register the institutes:

Use-case:Register the institutes

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

Actors:Head of Institute,Admin

Precondition: The director must be logged into the system.

Basic Flow:

1. The director logs into the IITBombayX platform and goes to his dashboard from his profile which is currently having only his details).
2. He then fills in the details of his institute on a form and sends this as a special notification to the administrator at IITBombayX.
3. The Admin at IITBombayX then performs his use-case roles for creating the entries for the institute.

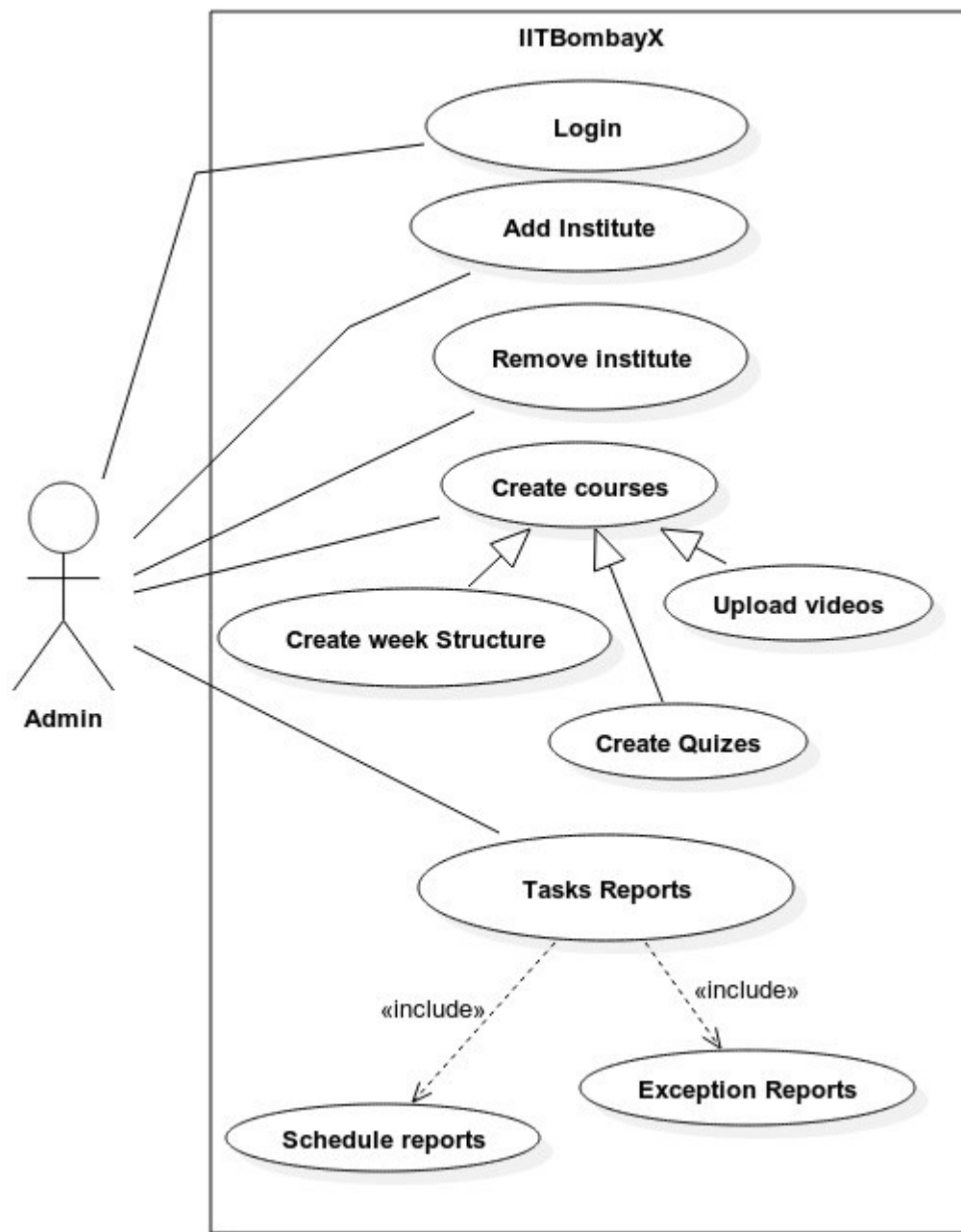
Exception Flow:

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

Postcondition: Director have successfully sent the details for the administrator to register the institutes.



Admin (IITBombayX):



### Admin's Use Cases

1)Login:

Use-case:Login

Purpose: Getting access to the account.

Actors:Admin

Precondition: The Admin need to have an account in the IITBombayX.

Basic Flow:

1. The Admin need to type his email-Id and password and click on 'Login button'.
2. After clicking on Login button,the administrator is directed to the dashboard.

Exception Flow:

1. If the Admin enters the invalid email-Id or password,there will be a message saying “Invalid email-Id or password”.

Postcondition: The Ad min will get access to the dashboard.

2)Add Institute:

Use-case:Add Institute

Purpose: The ad min 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 Director's use case), and these details are already available with the administrator.
3. The administrator provides his proper credentials, and then fires the commands and directives for creating the tables 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

3)Remove Institute:

Use-case:Remove Institute

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

Actors: Admin

Precondition: The admin needs 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 'Remove Institute' link from the options.
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 which 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 an 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.

Postcondition: Admin has successfully removed the institute from the IITBombayX.

#### 4) Create Courses:

Use-case: Create Courses

Purpose: Admin can create week courses structure for the student. He can also create quizzes and upload videos for the student.

Actors: Admin

Precondition: Need to login to the system. It should create different courses by keeping it weekwise for the students.

Basic Flow:

1. The admin needs to login into the system.
2. He should create different courses. This involves:
  - a. Creation of the week structure of the courses.
    - i. The admin (in this case a course creator) will go to his dashboard, and click on

- the 'Course Creator' studio option.
- ii. Once he enters this, he can create a new course through options available to him.
- iii. He can add weeks to his courses to create sub structure in his courses, which appear in the navigation sidebar.
- b. Addition of videos to the courses
  - i. Inside the course studio, once he has selected the course to modify, he can add course videos to that chapter he has selected from the navigation sidebar.
  - ii. These videos are then uploaded via a simple uploader and editor, which also asks for a transcript of the lecture to be created or added later, and a title for the videos.
  - iii. The videos appear in the navigation sidebar.
- c. Addition of exercises and quizzes
  - i. The course creator studio allows the admin to add course exercises at special points in the course same way as video, simply by changing the type of item to be added.
  - ii. In case of course exercises, he simply adds the questions and answers in a special online editor, and clicks on add to course. These can later be deleted from the navigation bar.
  - iii. In case of course exercises, he simply adds the questions and answers in a special online editor, adds the start and stop time stamps, and due date and time of the quiz, and clicks on add to course. These can later be deleted from the navigation bar.
- d. At the end of all editing, he simply clicks on the 'Save Changes' button on the online editor to save the changes to the courses.
- 3. He should inform the Head of the institute about the schedule of the courses via sending a notification, which occurs automatically internally with each 'Save Change' to the courses.

Exception Flow: If the courses have been created successfully but the uploading of the courses fails, the admin will be notified. In case of any errors in the courses in editing, in faults in uploading of errors in exercise formats, or so on – the admin will be notified of the errors accordingly through relevant messages.

Postcondition: Admin have successfully created the courses for the student.

## 5)Tasks Reports:

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.

#### Postcondition:

1. 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.

## **THE STUDENT REGISTRATION AND IDENTIFICATION PROBLEM**

### **PROBLEM STATEMENT**

In the present configuration of Blended MOOCS platform offered by IITBOMBAYX, a registered student is not required to fill in the details of his/her institute, as per the aim of the IITBOMBAYX to be available to everyone. However the grades allotted in this course are, pending further support, ineligible for contribution to the formal grading system of colleges.

- The Teacher teaching a specific subject in a college will have to select the students studying under him, from the database of IITBOMBAYX itself which might as well consist of tens of thousands of students.
- Additionally the teacher should be able to invite the students of his class for enrollment in a specific course, should the student fail to do so himself.
- Both the above task needs to be done without altering the primary system of the IITBOMBAYX.

The above task becomes challenging because the primary system of the IITBOMBAYX should not be altered and basically there is no direct method to link the appropriate students to their respective colleges.

### **PROPOSED SOLUTION**

In order to solve the above problem, the following steps are suggested by the Insight group.

#### **INSTITUTE REGISTRATION**

Every participating Institute of the country would have to be registered with the IITBombayX, which will assign a specific college id to each of them.

#### **STUDENT DATA UPDATE**

In order to link the appropriate students to their respective colleges, without altering the base system, it

is proposed that the students can be made to update the records of their current Institute whenever they login. This will be optional and a link to do so will always be available to the students either on their “Dashboards” or through the use of pop-ups regularly. Both these links will direct to a new page where the student can select his/her Institute from a given drop down list of the participating Institutes.

- The following method of updating the students’ institute details can be done after the registration of the students, and thus will not require altering the base production system of IITBombayX.
- The update is optional and thus will not interfere with the participation of students who wish to study independently.
- As, and when, a student registers his/her Institute, the institute Id of the Respective Institute will be stored in the student’s database.

### TEACHER’S HELPDESK

In order for an Institute to consider the IITBOMBAYX marks in their respective college grading system, there are 2 basic requirements.

- The student, registered on IITBOMBAYX, must enroll himself for the concerned SUBJECT Course, which the guest Institute wishes to grade its student in.
- The Subject Teacher, in the guest Institute, must identify all the students under him/her from the large database of the entire MOOCS students.

Additionally we would want that if a few students of the class forget to register for a given course, in which they will be graded by their respective Institutes, they be given a reminder or an invitation to enroll for the given course.

Thus every Teacher will be given an opportunity to select the list of students studying under him, from the MOOCS database. This will be available on the Teachers dashboard.

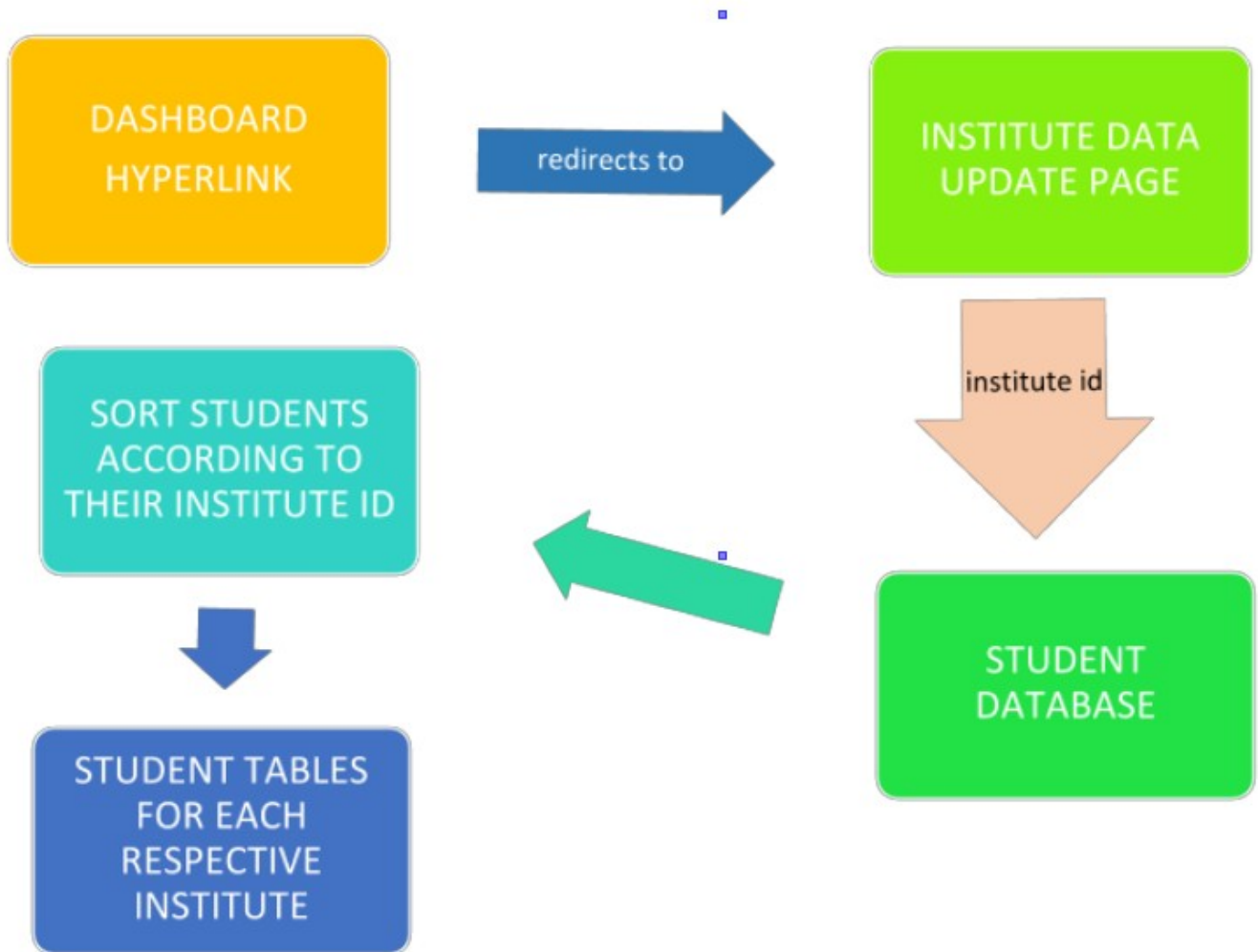
In order to facilitate the teacher in selecting the students under him, we will provide a helpdesk. In this helpdesk we will provide the list of all the students registered under the same college name, using the college id stored in the student’s database. The Teacher can form a list of the students under him/her.

### INVITATION BASED ENROLLMENT

Generally the students would automatically register for the course for which they will be graded, of course it bears marks! But in case the student forgets to register for a specific course then an invitation will be sent to the said student whenever the student’s teacher selects his/ her students. The students selected by a teacher of a particular subject will be searched, in the database, for whether they have enrolled for the same course or not. Those who would not have registered will get an invitation on their dashboard to join the course.

The following diagram is a schematic diagram of the process flow described above.

## FLOWCHART



(Contd.)

## TEACHER'S DASHBOARD

Provides an interface to the Teachers to manage his courses  
He can oversee his course and students as well



## TEACHER'S HELPDESK

An interactive interface to aid the teacher



## STUDENT SELECTION

The Teacher can shortlist the students presently under him



## STUDENT LIST

The Teacher can be provided a list of all the students registered to the same institute.



## REGISTRATION

The shortlisted students are registered for the given course under the concerned teacher.



## INVITATION

If a students forgets to enroll for a subject in which a teacher registers him, he will get an invitation to do the same.