

Institute of Information Technology

# **Course Management System**

Software Requirements Specification

# **Course Management System for IIT**

## **Submitted to:**

Amit Seal Amit  
Lecturer  
Institute of Information Technology  
University of Dhaka

Dr. Md. Mahbubul Alam Joarder  
Professor  
Institute of Information Technology  
University of Dhaka

## **Submitted by:**

MD. Mehedi Hasan (BSSE 0607)

Shubho Chandro Roy (BSSE 0619)

Nur Mohammad (BSSE 0624)

Farzana Akter (BSSE 0627)

Tanvir Ahassan (BSSE 0629)

Khairul Araf (BSSE 0631)

Rubama Yesmin (BSSE 0632)

Tanvir Ahmed Siddiki (BSSE 0639)

## **Submission Date:**

17<sup>th</sup> November, 2016

**Institute of Information Technology**

**University of Dhaka**  
**Letter of Transmittal**

17<sup>th</sup> November, 2016

**Amit Seal Amit**

Lecturer

Institute of Information Technology

University of Dhaka

**Subject: Submission of documentation on “Course Management System”.**

Dear Sir,

With due respect, we are submitting the documentation on the above project you assigned to us. In this project document, we have to give our best effort albeit there might be some shortcomings. We would be highly obliged if you consider those from excusable point.

Yours sincerely,

MD. Mehedi Hasan (BSSE 0607)

Shubho Chandro Roy (BSSE 0619)

Nur Mohammad (BSSE 0624)

Farzana Akter (BSSE 0627)

Tanvir Ahassan (BSSE 0629)

Khairul Araf (BSSE 0631)

Rubama Yesmin (BSSE 0632)

Tanvir Ahmed Siddiki (BSSE 0639)

## Acknowledgement

We are highly indebted for getting such a tremendous opportunity to prepare the report on **Course Management System for IIT**. We would like to thank whole-heartedly our course teacher, Amit Seal Amit, Lecturer, Institute of Information Technology, University of Dhaka, for giving us guideline about how can we prepare this project documentation. In completing this paper, we have collected various important data and information from stakeholders. We are thankful to all of the works cited.

## **Abstract**

The study is made for Course Management System. The scope of study is to analyze the Course Management System for IIT and to know drawbacks of it. Also, enable to design SRS of this system. The object of this study is to develop a SRS (software requirements and specification) of the Course Management System. It also helps this study reveals that overall Course Management System for IIT.

# Table of Contents

Chapter 1: Introduction .....	9
1.1 Purpose .....	9
1.2 Intended Audience .....	9
Chapter 2: Inception .....	11
2.1 Introduction .....	11
2.1.1 List of Stakeholders .....	11
2.1.2 Recognizing multiple view point .....	12
2.1.3 Working towards collaboration .....	13
2.1.4 Requirements questionnaire .....	14
2.2 Conclusion .....	15
Chapter 3: Elicitation.....	16
3.1 Introduction .....	16
3.2 Eliciting requirements .....	16
3.2.1 Collaborative Requirements Gathering .....	16
3.2.2 Quality Function Deployment (QFD).....	17
3.2.3 Usage Scenario .....	18
3.3 Elicitation Work Product .....	19
3.4 Conclusion .....	19
Chapter 4: Scenario Based Modeling.....	20
4.1 Definition of Use case .....	20
4.2 Use Case Diagrams .....	21
4.2.1 System Description from Level-0 use case: .....	21
4.2.2 System description from level-1 use case diagram: .....	26
4.3 Activity & Swim lane Diagrams .....	37
.....	37
Chapter 5: Data Model.....	71
5.1 Data Modeling Concept .....	71
5.2 Data Objects.....	71
5.2.1 Grammatical parsing (Noun identify) .....	71
5.2.2 Identify Data Objects .....	73

5.2.3 Data Object Relation .....	75
5.2.4 E-R Diagram.....	76
5.3.5 Schema Form (Tables).....	77
Chapter 6: Class Based Model.....	78
6.1 Class Based Modeling Concept .....	78
6.1.1 Identifying Analysis Class .....	78
Chapter 7: Flow Oriented Model .....	86
Introduction .....	86
1.1 Data Flow Diagram.....	86
Chapter 8: Behavioral Model .....	91
8.1 State Transition Diagram .....	91

# Table of Figures

Figure 1: Level-0 Use case diagram.....	25
Figure 2: Level-1 Use case diagram.....	30
Figure 5: Level-1.3 Use Case Diagram.....	33
Figure 6: Level-1.4 Use Case Diagram.....	34
Figure 7: Activity Diagram for Sign Up.....	<b>Error! Bookmark not defined.</b>
Figure 9: Activity for Sign In .....	37
Figure 14: Activity diagram for Admission Process .....	41
Figure 15: Swim Lane for Admission Process .....	42
Figure 16: Activity diagram for Course allocation .....	43
Figure 17: Swim lane diagram for Course allocation .....	44
Figure 18: Activity diagram for Course List.....	45
Figure 19: Swim lane diagram for Course List .....	46
Figure 20: Activity diagram for Marks entry and Tabulation sheet formation .....	47
Figure 21: Swim lane diagram for Marks entry and Tabulation Sheet formation.....	48
Figure 22: Activity diagram for Verification and Result Publicationn.....	49
Figure 23: Swim lane diagram for Verification and Result Publicationn .....	50
Figure 24: E-R Diagram.....	76
Figure 25: Data Schema .....	<b>Error! Bookmark not defined.</b>
Figure 26: CRC Diagram .....	<b>Error! Bookmark not defined.</b>
Figure 27: Level-0 DFD .....	87
Figure 28: Level-1 DFD .....	88
Figure 29: Level-2.1 DFD for Authentication .....	88
Figure 30: Level-2.2 DFD for Tabulation Sheet Verification .....	89
Figure 31: Level-2.3 DFD for Result Card Verification .....	90
Figure 32: State Transition Diagram for Authentication .....	92
Figure 33: State Transition Diagram for Database .....	93
Figure 34: State Transition Diagram for Principal.....	93
Figure 35: State Transition Diagram for Teacher.....	94
Figure 38: State Transition Diagram for Student.....	<b>Error! Bookmark not defined.</b>



# Chapter 1: Introduction



This chapter is intended to specify the purpose of this document and the intended audiences of it.

## 1.1 Purpose

This document is the Software Requirement Specification (SRS) for the Course Management System (CMS) for IIT. It contains functional, non-functional and support requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered, and organized by topic. The SRS serves as official means of communicating user requirements to the developer and provides a common reference point for both the developer team and stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

## 1.2 Intended Audience

This SRS is intended for several audiences including the customers as well as the project managers, designers, developers, and testers.

-  The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
-  The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.

- ✚ The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- ✚ The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created software that will fulfill all of the customer's documented requirements.
- ✚ The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.





# Chapter 2: Inception

In this chapter, the Inception part of the SRS will be discussed briefly.

## 2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how does a software project get started and what is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project. At project inception, we establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired, and the effectiveness of preliminary communication and collaborations between the other stakeholders and the software team.

To establish the groundwork, we have worked with the following factors related to the inception phases:

-  List of stakeholders
-  Recognizing multiple viewpoints
-  Working towards collaboration
-  Requirements questionnaire

### 2.1.1 List of Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited (Chapter 3) is created. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: “Whom else do you think I should talk to?”

To identify the stakeholders, we consulted with our honorable teacher, Amit Seal Amit and asked him following questions:

- ✚ Who will be using the project outcomes?
- ✚ Who has resources I need to get the project done?
- ✚ Whose work will my project affect? (During the project and also once the project is completed).
- ✚ Who will be the administrator of the system?

Concluding thoughts on Stakeholders thoughts we identified the following stakeholders for our automated Course Management System for IIT.

**Admin:** System administrator will use the system for adding and removing teacher to the system

**Head of Course committee:** Head of the Course committee will use the system for adding and removing course. He/she will also allocate and de-allocate teacher to courses.

**Teacher:** Teacher take different courses of different batch and they will use the system for entering marks into marks distribution page.

**Student:** Students will use the system for discussion and view, download their results.

**Student staff:** Student staff will use the system for adding student to the system and removing student from the system.

## 2.1.2 Recognizing multiple view point

Different stakeholders achieve different benefits. Consequently, each of them has a different view of the system. So, we have to recognize the requirements from multiple points of view, as well as multiple views of requirements. Assumptions are given below:

### **Admin's view point:**

- ✚ An automated course management system
- ✚ Better student and teacher information system
- ✚ Online result system

**Head of the Course committee's view point:**

- ✚ Better course managing
- ✚ Error free system
- ✚ Restrict access to functionality of the system
- ✚ Maintain a database of all the course related information on the system

**Teacher's view point:**

- ✚ Error free system
- ✚ Maintaining a dynamic marks distribution
- ✚ Restrict marks visibility to others
- ✚ Better marks entry/update system
- ✚ Controlling the system allocation

**Student's view point:**

- ✚ Friendly user interface of the system
- ✚ Easy marks viewable system
- ✚ Discussion forum that will help them to study well

**Student staff's view point:**

- ✚ Better student information management
- ✚ Restrict unauthorized access
- ✚ Information security

## 2.1.3 Working towards collaboration

Every stakeholder has their own requirements. We followed following steps to merge these requirements we-

- identify the common and conflicting requirements
- Categorize the requirements

- Take priority points for each requirement from stakeholders and on the basis of this voting prioritize the requirements
- Make final decision about the requirements

#### **Common requirements:**

- ✓ User friendly and efficient system
- ✓ Information Management
- ✓ Restrict unauthorized access
- ✓ Authentication
- ✓ Database containing detailed information about users, course and results

**Final requirements:** We finalized following requirements for the system by categorizing and prioritizing the requirements.

- ✚ Error free system (Maximum 5% error may be considerable)
- ✚ Allow the users login and logout
- ✚ Restrict access to functionality of the system based upon user roles
- ✚ User friendly and efficient system
- ✚ Secure way to communicate with script section, result section, grade sheet section and departments
- ✚ Automated entry of student information
- ✚ Central database contains all result related information about every students

## **2.1.4 Requirements questionnaire**

We set our first set of context-free questions focuses on the customer and other stakeholders, overall project goals and benefits. The questions are mentioned above. These questions helped us to identify all stakeholders, measurable benefit of the successful implementation and possible alternatives to custom software development. Next set of question helped us to gain a better understanding of problem and allows the customer to voice his or her perception about the solution. The final set of question focused on the effectiveness of the communication activity itself.

## 2.2 Conclusion

Inception phase helped us to establish basic understanding about the Course Management System of IIT, identify the people who will be benefited if this system becomes automated, define the nature of the Course Management System software and establish a preliminary communication with our stakeholders.

In our project, we have established a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stake-holders and the software team. More studies and communication will help both side (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

# Chapter 3: Elicitation





After discussing on inception part, we need to keep focus on the elicitation part. So this chapter specifies the elicitation part.

## 3.1 Introduction

Requirements elicitation is a part of requirement engineering that is the practice of gathering requirements from the users, customers, and other stakeholders. We have faced many problems like understanding the problems, problems of making questions for the stakeholders, problems of less communication with the stakeholders for time limitation, problems of volatility. Though it is not too easy to gather requirements within a very short time, we have surpassed these problems in an organized and systematic manner.

## 3.2 Eliciting requirements

We have seen Question and Answer (Q&A) approach in the previous chapter where the inception phase of requirement engineering has been described. The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. We have finished the following tasks for eliciting requirements-

-  Collaborative Requirements Gathering
-  Quality Function Deployment
-  Usage Scenarios
-  Elicitation work products

### 3.2.1 Collaborative Requirements Gathering

Actually, we met with many stakeholders in the inception phase such as our honorable teacher of IIT, Amit Seal Amit who handles the total course management process, Head of the course



committee who deals with the course management of IIT, Staff Student who manages all the student management processes. These meetings created an indecisive state for us to elicit the requirements. To solve this problem, we have met with the stakeholders (who are acting a vital rule in the whole admission process) again to elicit the requirements. A slightly different scenario from these approaches has been found. We had to complete the following steps to do it.

- ✚ The meetings were conducted with the admin, head of course committee, teacher and the students of IIT, they were questioned about their requirements and expectations from the automated Course Management System.
- ✚ They were asked about the problems they are facing with the current manual system.
- ✚ At last we selected our final requirement list from the meetings.

### 3.2.2 Quality Function Deployment (QFD)

Quality Function Deployment (QFD) is the methods or quality management technique that translates the needs of the customer into technical requirements for the software. Ultimately the goal of QFD is to translate often subjective quality criteria into objective ones that can be quantified and measured and which can then be used to design and manufacture the product. It is a method for maximizing customer satisfaction from the software engineering process. So we have followed this methodology to identify the requirements for the project. The requirements, which are given below, are identified successfully by the QFD-

#### 3.2.2.1 *Normal requirements*

Normal requirements consist of objectives and goals that are stated during the meeting with the customers. Normal requirements of our project are-

- ✚ Restrict unauthorized access
- ✚ Course management system
- ✚ Teacher information management system
- ✚ Dynamic teacher and student allocation
- ✚ Dynamic marks distribution
- ✚ Dynamically generated marks sheet

- ✚ Easily marks editable system
- ✚ Setting deadline of marks entry
- ✚ Restrict view ability of marks
- ✚ Discussion forum
- ✚ User friendly interface

### 3.2.2.2 *Expected requirements*

These requirements are implicit to the system and may be so fundamental that the customer does not explicitly state them. Their absence will be a cause for dissatisfaction.

- ✚ Maintain a database of all items to keep the information of the students
- ✚ Security issue will be ensured
- ✚ The system would be allowing the user to login based on assigned login ID and password
- ✚ If anyone forget password, show the recovery procedure
- ✚ The user interface of the system shall be easy to use and shall make use of drop-down boxes, radio buttons, and other selectable fields wherever possible instead of fields that require the user to type in data

### 3.2.2.3 *Exciting requirements*

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present.

- ✚ Good graphical presentation
- ✚ Download data as CSV
- ✚ Backup all information of the system

Because of time limitations we could not add exciting requirements in the system.

## 3.2.3 Usage Scenario

There is a student staff who enlist the students. She/he can add a new student providing email address and name of the student. She/he also have the authority to remove any student from the system. Teachers are added by the system admin.

Head of Course committee opens courses of the system. Each course has unique course code, course name and course description. When a course is added, head of course committee have to allocate a teacher to that course. He/she can change that allocated teacher of any course. Any course can be removed by him/her.






When a teacher is allocated, he/she assigns the students to that course. Then he/she must provide the marks distribution of that course. He/she can distribute total 100 marks as he wishes. Each distribution has distribution criteria and weight. Total weight of distribution must be 100.

Head of Course committee set the time of marks entry. A start date and a end date of entering marks is set to each of the courses. A teacher can enter/update marks within the fixed time limit. He/she sets the permission of viewing the marks. A student can view the marks if the marks is viewable.

There is discussion forum for each course. Student/teacher can ask question and answer to the question that has been asked at the forum.

### 3.3 Elicitation Work Product

At first we have to know that the output of the Elicitation task may vary because of the dependency on size of the system or product to be built. Here, the Elicitation work product includes:

-  Making a statement of our requirements for the Course Management System.
-  Making a bounded statement of scope for our system.
-  Making a list of users, and other stakeholders who participated in the requirements elicitation.
-  A set of usage scenarios that provide insight into the use of the system.
-  Description of the system's technical environment.

### 3.4 Conclusion

Elicitation phase helped us to understand about the problems of our scopes of the system. This phase also helped us to identify the requirements, negotiate different approaches and specify a preliminary set of solution requirements in an atmosphere that is conducive to the accomplishment of the goal.

# Chapter 4: Scenario Based Modeling

This chapter describes the scenario based model for the post exam control system.

## 4.1 Definition of Use case

A use case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a use case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A use case diagram simply describes a story using corresponding actors, who perform important role in the story and makes the story understandable for the users.

The first step in writing a use case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using system.

### **Primary Actor:**

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

### **Secondary Actor:**

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

## 4.2 Use Case Diagrams

Use case diagrams give the non-technical view of overall system.

### 4.2.1 System Description from Level-0 use case:

After analyzing the user story, we found five actors who will directly use the system as a system operator. Primary actors are those who will play action and get a reply from the system whereas secondary actors only produce or consume information.

Following are the actors of Course Management System –

- Admin (primary)
- Student staff (primary)
- Head of Course committee (primary)
- Teacher (primary)
- Student (primary)

#### **Student staff**

Action1: Opens login page

Reply1: Please fill up email id and password

Action2: Enter email id and password

Reply2: Login successful

Action3: Open student list

Reply3: Student list opened

Action4: Add new student

Reply4: Enter student name and email id

Action5: Enter student name and email id

Reply5: Student is added

Action6: Remove student

Reply6: Enter email id

Action7: Enter email id

Reply7: Student is removed

Action8: Click on logout

Reply8: Logged out

## **Teacher**

Action1: Opens login page

Reply1: Please fill up email id and password

Action2: Enter email id and password

Reply2: Login successful

Action3: Open course allocation page

Reply3: Course allocation page opened

Action4: Enter student email id and search

Reply4: View student list

Action5: Select student

Reply5: Students are selected

Action6: Click on distributed marks

Reply6: Enter marks and weight

Action7: Enter marks and weight

Action8: View distributed marks list

Reply8: Show list

Action9: Number shit generate

Reply9: Show number shit

Action10: Enter student's marks

Reply10: Successfully entry marks

Action11: Click on logout

Reply11: Logged out

### **System admin**

Action1: Opens login page

Reply1: Please fill up email id and password

Action2: Enter email id and password

Reply2: Login successful

Action3: Open teacher list

Reply3: Teacher list is opened

Action4: Add teacher

Reply4: Enter name and email id

Action5: Enter teacher name and email id

Reply5: Teacher is added

Action6: Click on teacher list

Reply6: Teacher list is viewed

Action7: Click on remove teacher

Reply7: Enter email id

Action8: Enter teacher email id

Reply8: Teacher is removed

Action9: Click on logout

Reply9: Logged out

### **Course Committee**

Action1: Opens login page

Reply1: Please fill up email id and password

Action2: Enter email id and password

Reply2: Login successful

Action3: Open course list

Reply3: Course list opened

Action4: Add new course

Reply4: Enter course name, course code and description

Action5: Enter course name, course code and description

Reply5: Select teacher

Action6: Click on select teacher and select teacher

Reply6: teacher selected

Action7: Click on remove course

Reply7: Enter course code number

Action7: Enter course code number

Reply7: Course removed

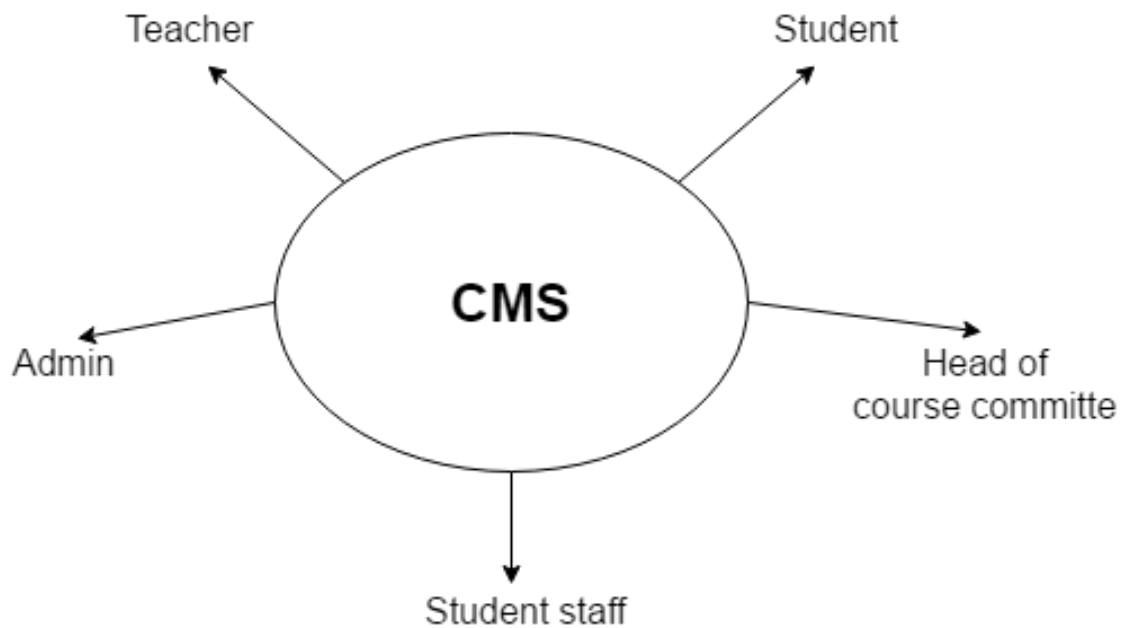
Action8: Click on logout

Reply8: Logged out



#### 4.2.1.1 Level-0 Use Case Diagram

In this level of use case diagram describes the overall system and the actors interacting with the system. Here in our course management system we have five actors interacting with the system.



**Figure 1: Level-0 Use case diagram**

## 4.2.2 System description from level-1 use case diagram:

The actors of course management system have to play different actions and system will reply according to these actions –

### **Principal:**

Action1: Enters signup.

Reply1: Please fill up the required information.

Action2: Enters the information.

Reply2: Registration successful.

Action3: Enters username and password.

Reply3: Login successful.

Action4: Verify tabulation sheets.

Reply4: verified.

Action4: Verify result card.

Reply4: verified.

Action4: Allocates teachers.

Reply4: allocated.

### **Head of Account Section:**

Action1: Enters signup.

Reply1: Please fill up the required information.

Action2: Enters the information.

Reply2: Registration successful.

Action3: Enters username and password.

Reply3: Login successful.

Action4: Verify student admission payment.

Reply4: Verified.

Action5: Confirm student admission.

Reply5: Confirmed.

### **Head of Result Section:**

Action1: Enters signup.

Reply1: Please fill up the required information.

Action2: Enters the information.

Reply2: Registration successful.

Action3: Enters username and password.

Reply3: Login successful.

Action4: Generate tabulation sheet.

Reply4: Generated.

Action5: Generate result card.

Reply5: Generated.

### **Teacher:**

Action1: Enters signup.

Reply1: Please fill up the required information.

Action2: Enters the information (including exam process: single/dual).

Reply2: Registration successful.

Action3: Enters username and password.

Reply3: Login successful.

Action4: Enter marks into number sheet.

Reply4: Entered successfully.

**Student:**

Action1: Enters signup.

Reply1: Please fill up the required information.

Action2: Enters the information (including exam process: single/dual).

Reply2: Registration successful.

Action3: Enters username and password.

Reply3: Login successful.

Action4: Fills up admission form.

Reply4: Successfully filled up.

Action5: Download result.

Reply5: Download completed.

**Student**

Action1: Opens login page

Reply1: Please fill up email id and password

Action2: Enter email id and password

Reply2: Login successful

Action3: Click enroll course

Reply3: View allocated course

Action4: Click view marks

Reply4: Select course

Action5: Select course

Reply5: View marks

Action6: Click on discussion forum

Reply6: Select course

Action7: Click on question and write a question

Reply7: Submit

Action7: Click submit

Reply7: Question submitted

Action8: Write answer

Reply8: Write something

Action9: Submit answer

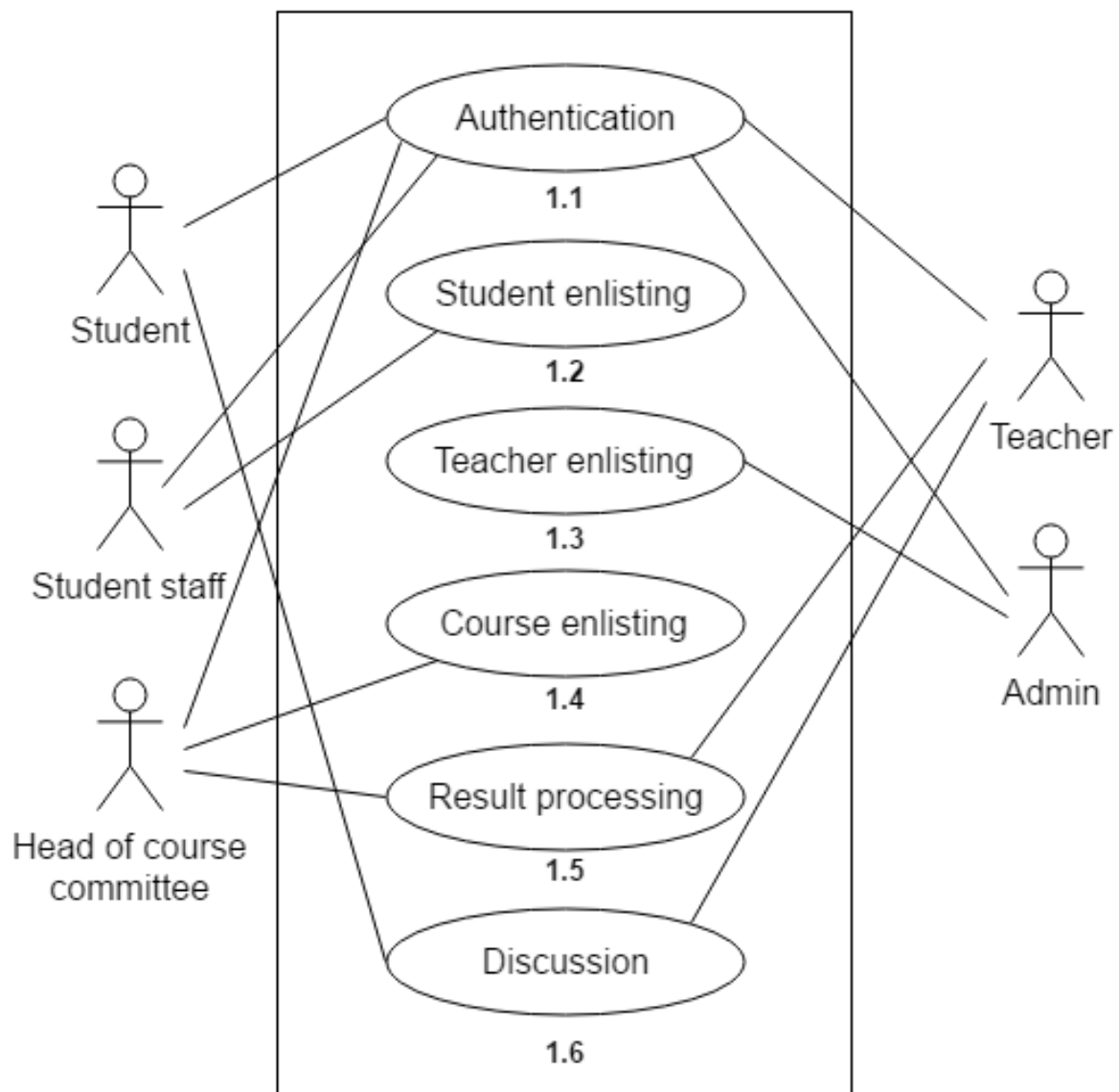
Reply9: Answer submitted

Action10: Click on logout

Reply10: Logged out

#### 4.2.2.1 Level-1 Use Case Diagram

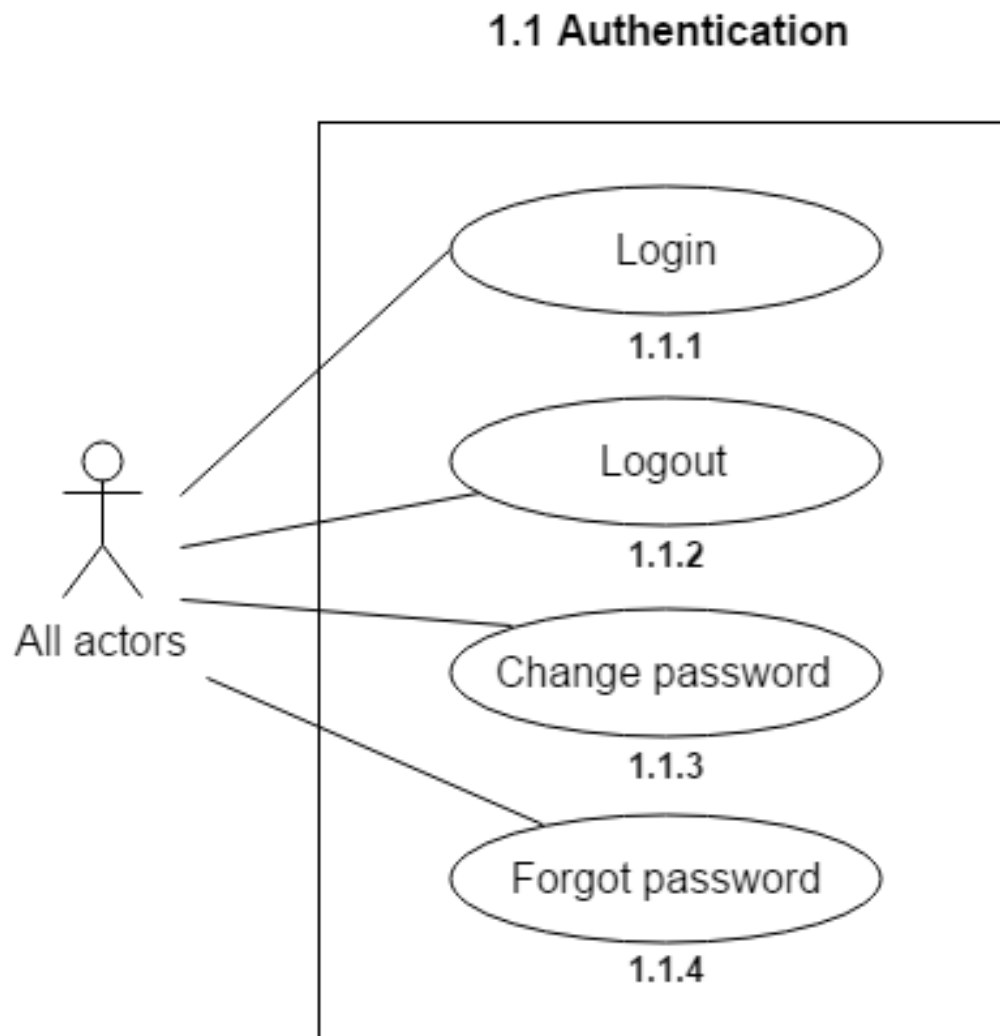
Level-1 use case where total system is divided into its subsystems which elaborately described in section 4.2.2.



**Figure 2: Level-1 Use case diagram**

#### 4.2.2.1.1 Subsystems of Level-1.1 Use Case Diagram

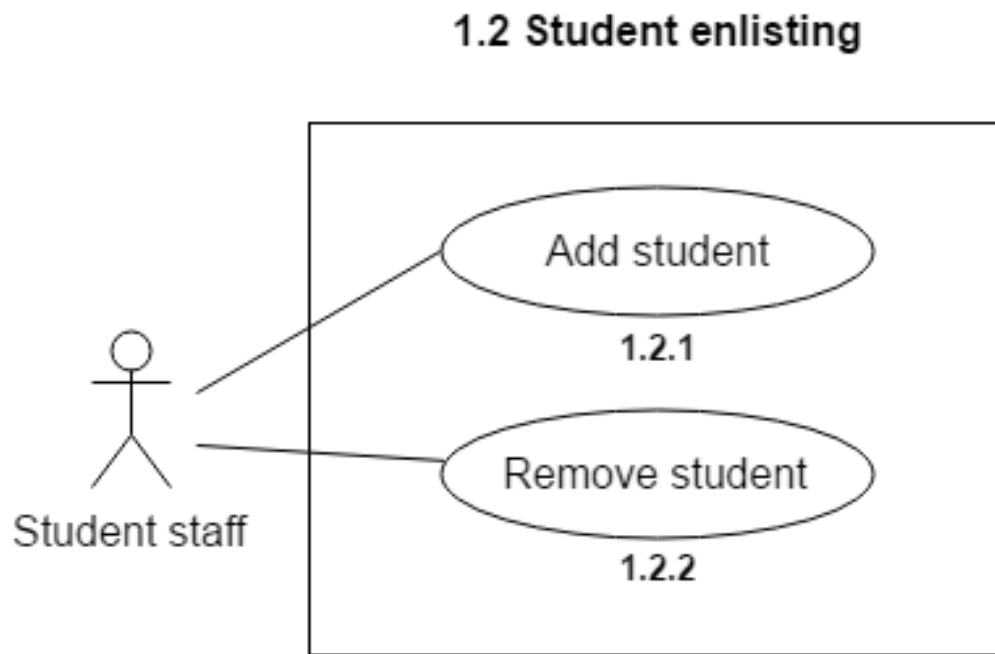
Subsystems of subsystem 1.1 of level-1 use case diagram. System description described in the section 4.2.2.



**Figure 3: Level-1.1 Use Case Diagram**

#### 4.2.2.1.2 Subsystems of Level-1.2 Use Case Diagram

Subsystems of subsystem 1.2 of level-1 use case diagram. System description described in the section 4.2.2.

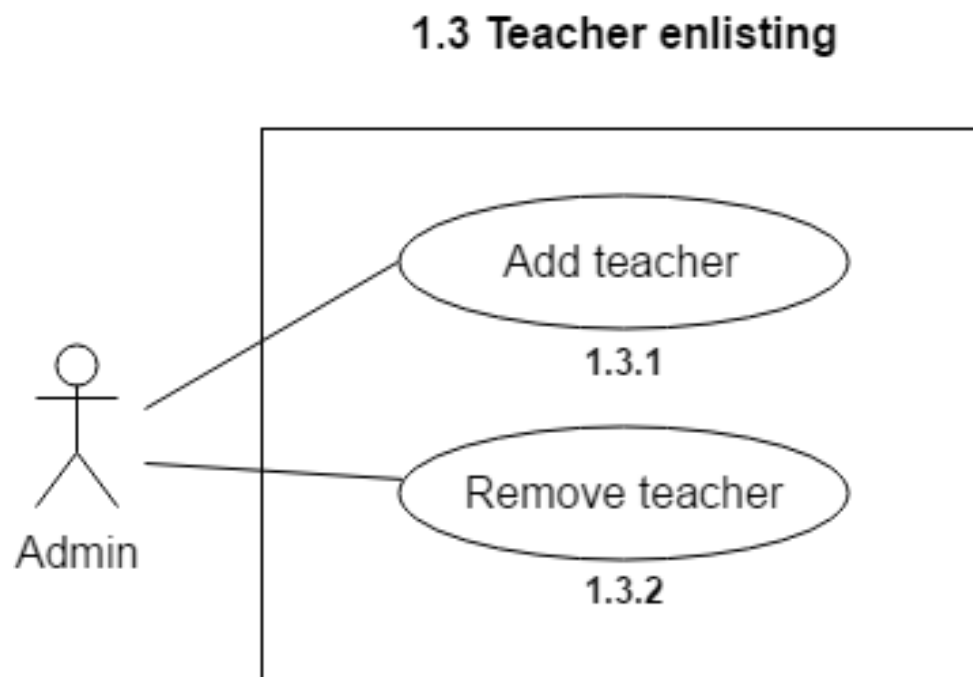


**Figure 4: Level-1.2 Use Case Diagram**



#### 4.2.2.1.3 Subsystems of Level-1.3 Use Case Diagram

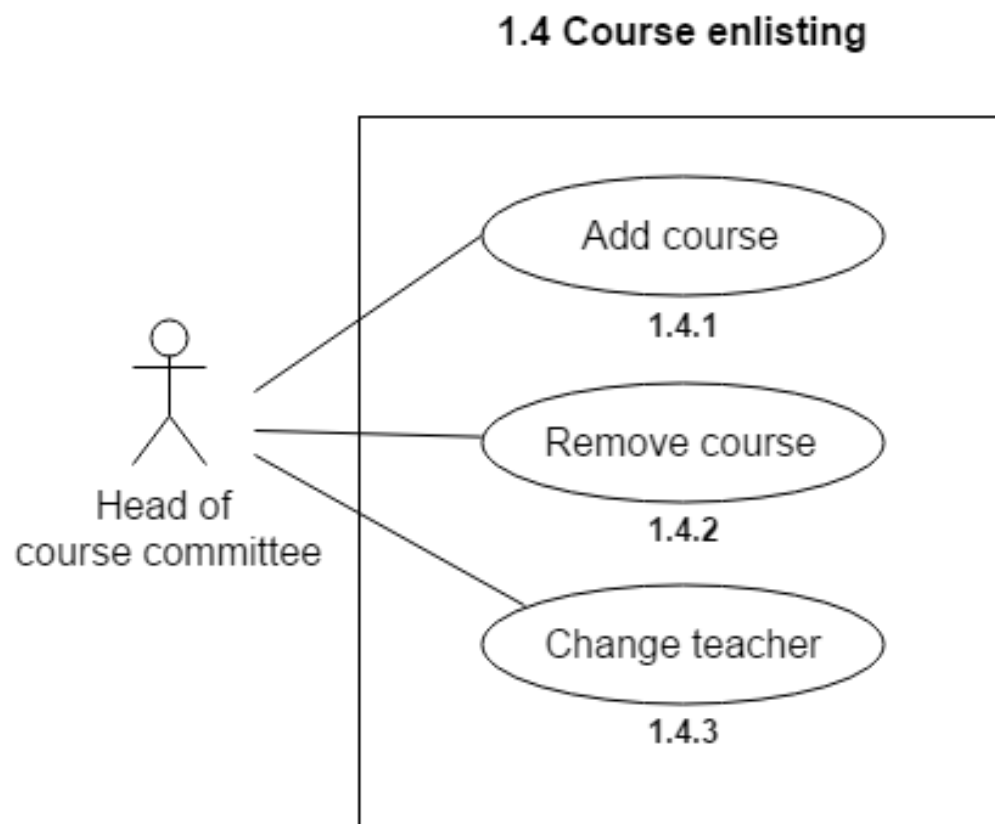
Subsystems of subsystem 1.3 of level-1 use case diagram. System description described in the section 4.2.2.



***Figure 5: Level-1.3 Use Case Diagram***

#### 4.2.2.1.4 Subsystems of Level-1.4 Use Case Diagram

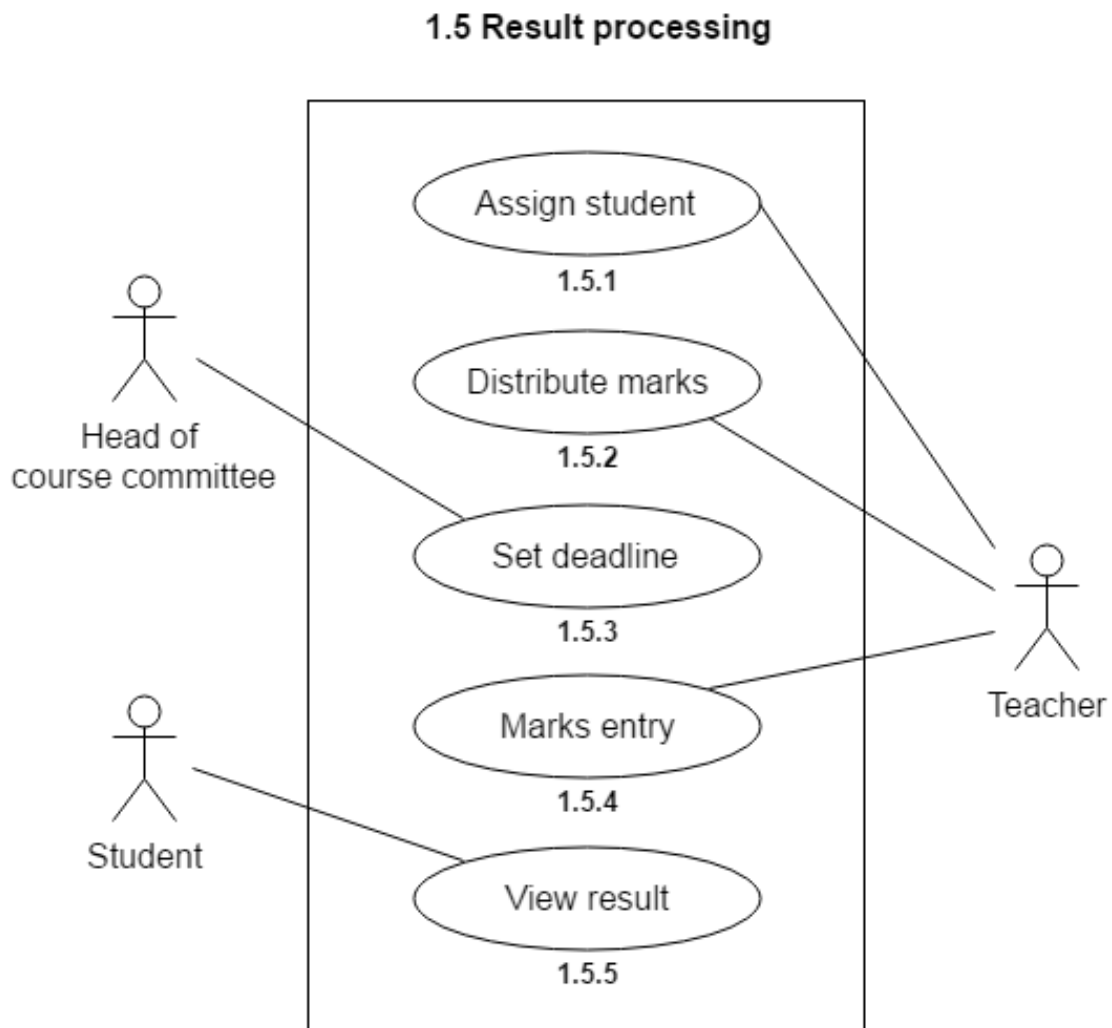
Subsystems of subsystem 1.4 of level-1 use case diagram. System description described in the section 4.2.2.



**Figure 6: Level-1.4 Use Case Diagram**

#### 4.2.2.1.5 Subsystems of Level-1.5 Use Case Diagram

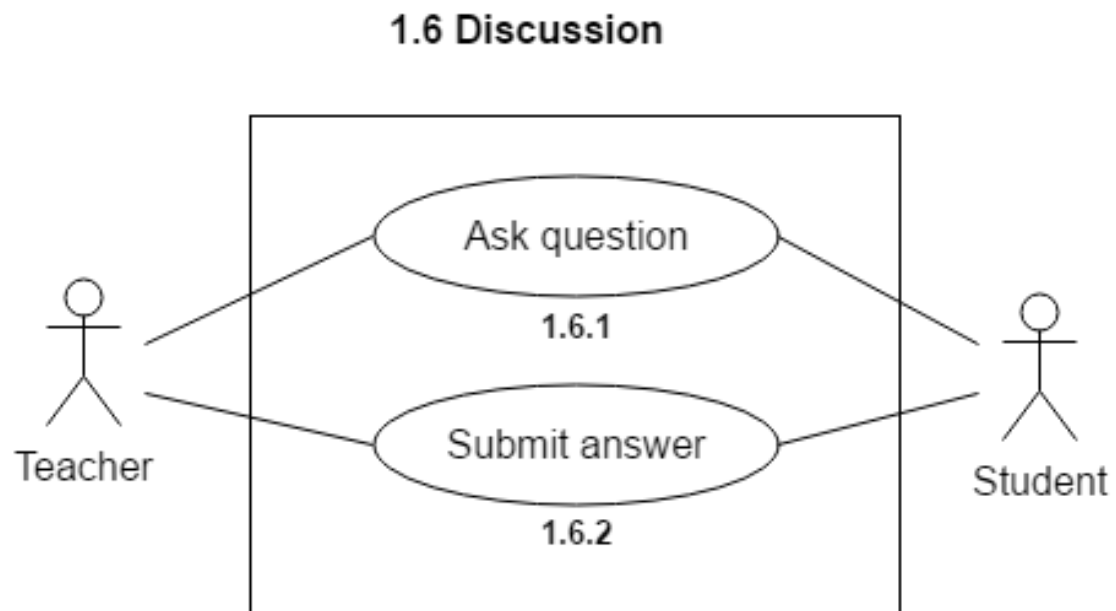
Subsystems of subsystem 1.5 of level-1 use case diagram. System description described in the section 4.2.2.



**Figure 7: Level-1.5 Use Case Diagram**

#### 4.2.2.1.6 Subsystems of Level-1.6 Use Case Diagram

Subsystems of subsystem 1.6 of level-1 use case diagram. System description described in the section 4.2.2.



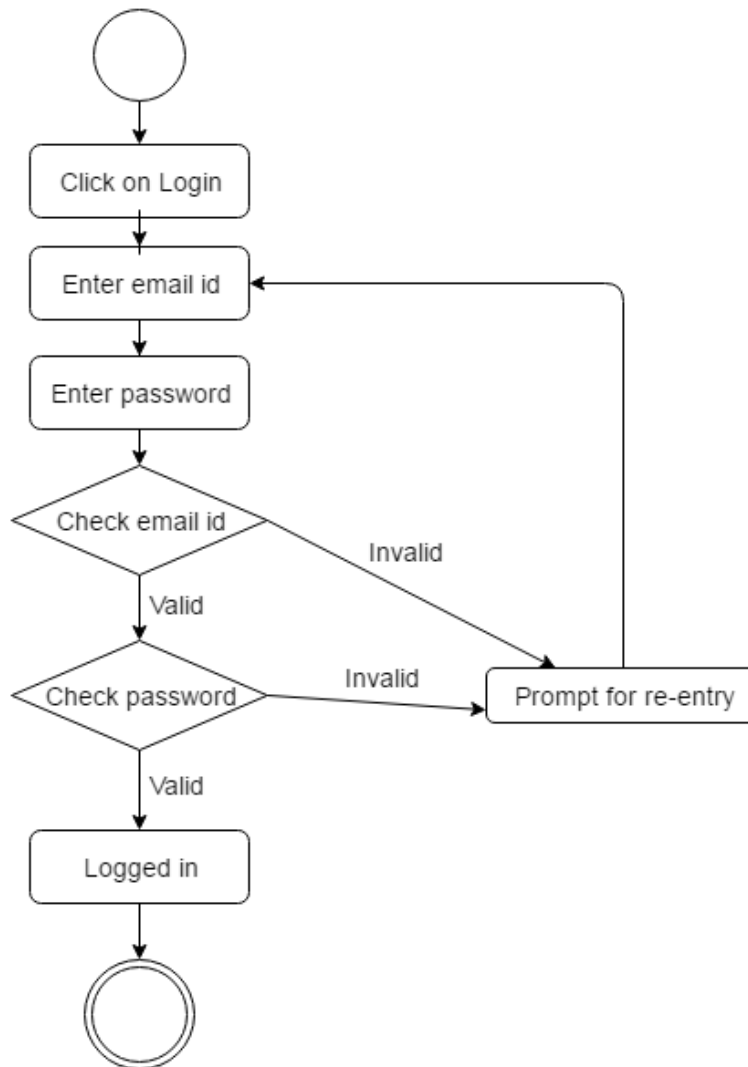
**Figure 8: Level-1.6 Use Case Diagram**

## 4.3 Activity & Swim lane Diagrams

**Activity diagram** shows the technical view of the system for every use case from which we can understand how the system actually works and how the actors interact with the system.

Here is the activity diagram for our Login use case.

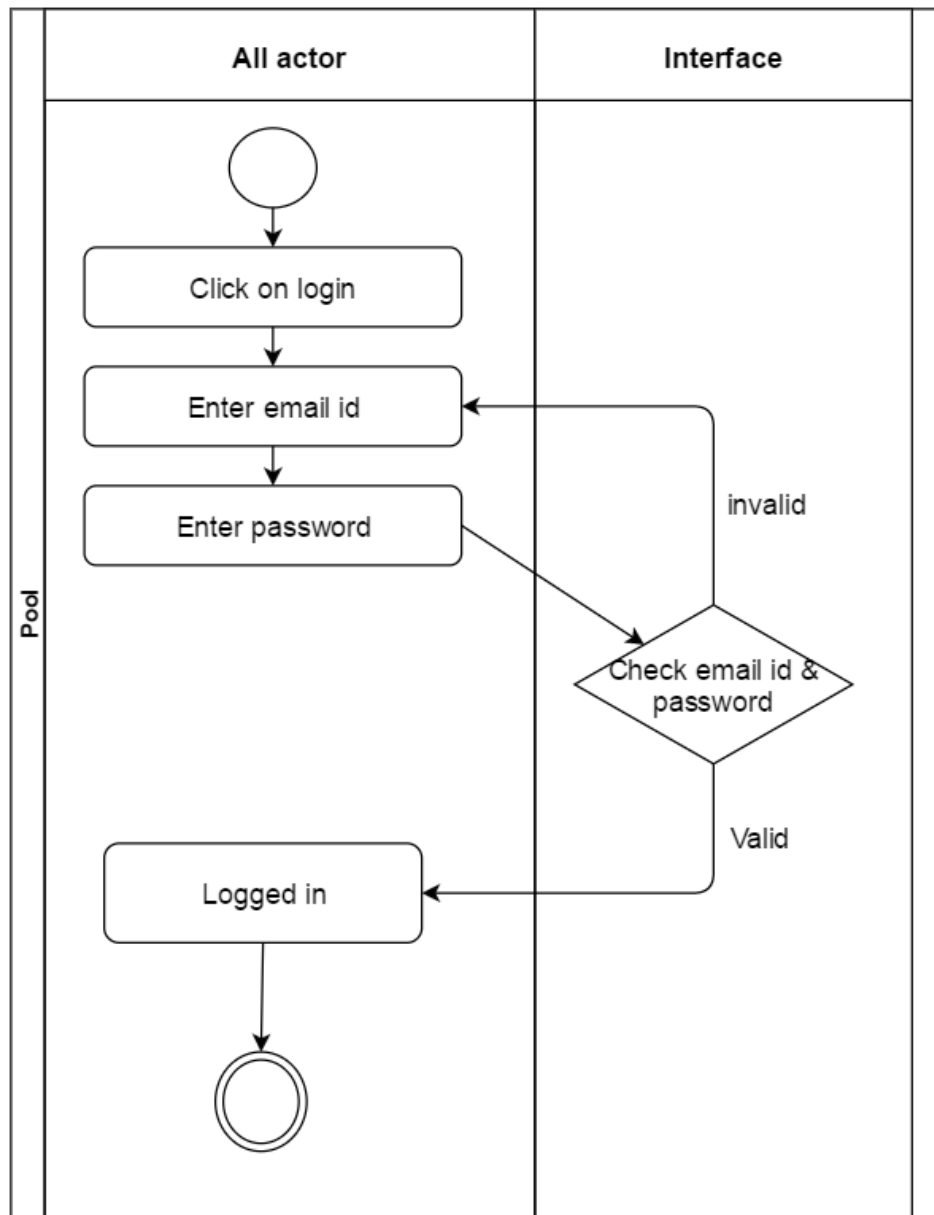
### Usecase 1: Login



**Figure 7: Activity for Login**

**Swim lane** diagram of a specific activity diagram shows the responsibilities of each actor dividing them into lanes. From this diagram we can improve our understanding about how the system works and which actors play what role.

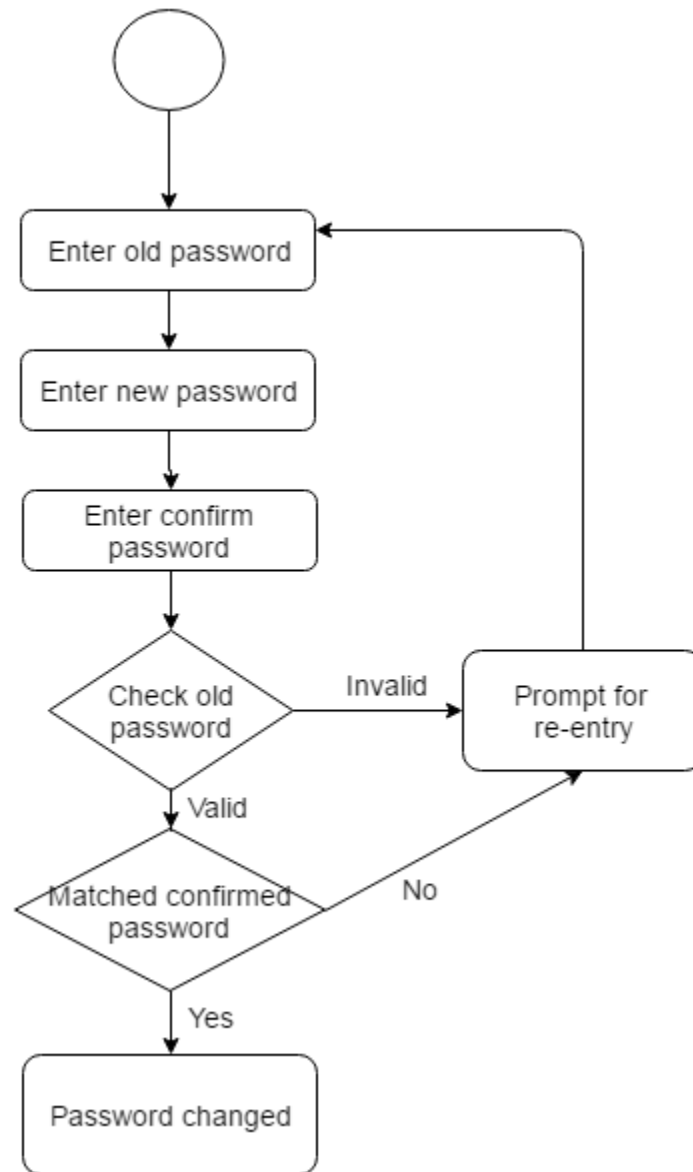
Following is the swim lane diagram of use case Login.



**Figure 10: Swim Lane for Login**

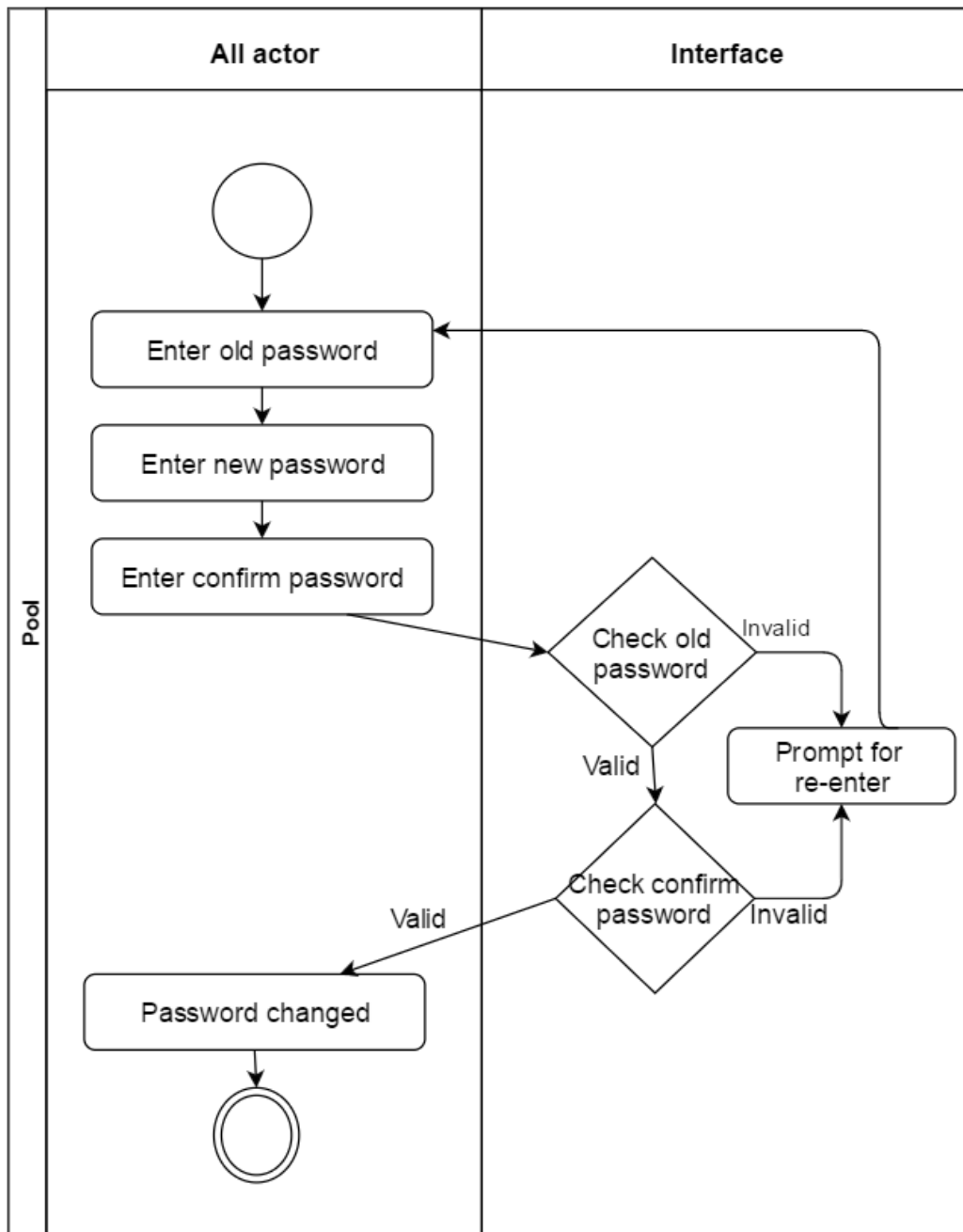
Following is the activity diagram of use case Change password.

## Usecase 2: Change password



**Figure 11: Activity for Change password**

Following is the swim lane diagram of use case Change password.

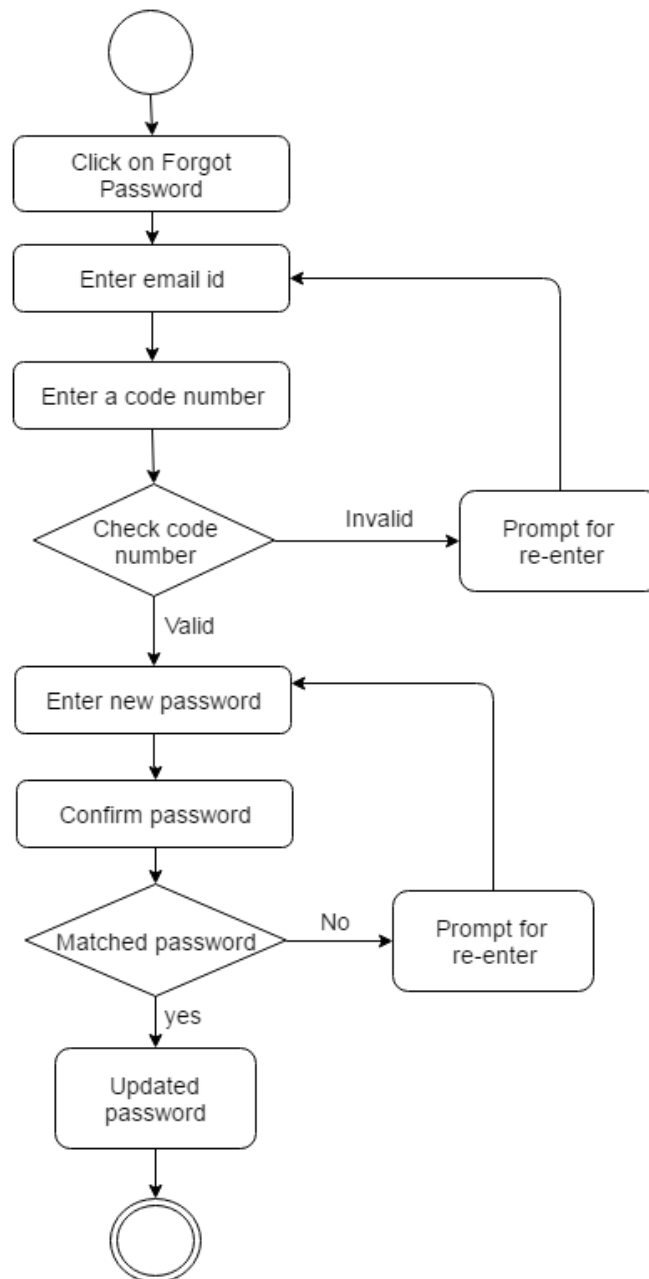


**Figure 8: Swim Lane for Change password**



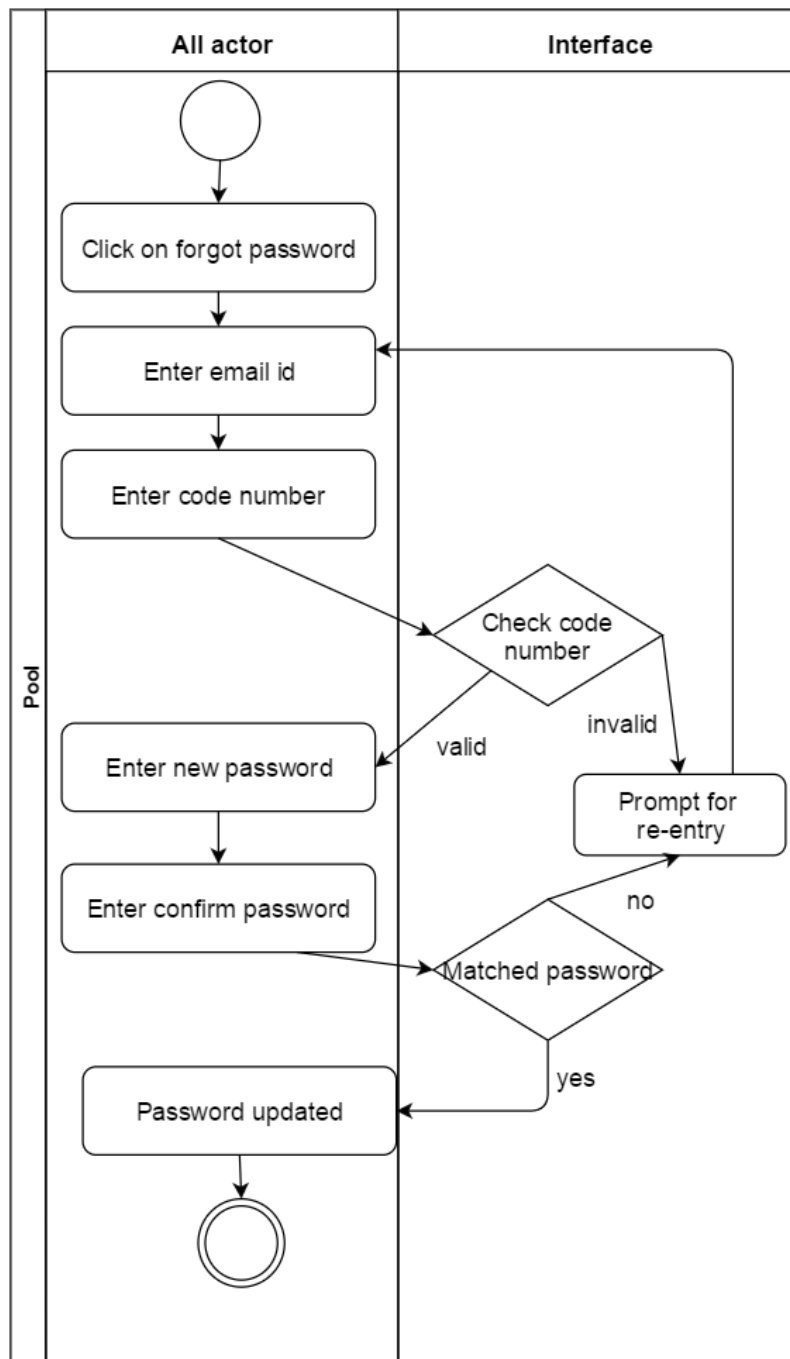
Following is the activity diagram of use case Forgot password.

### Usecase 3: Forgot password



**Figure 14: Activity diagram for Forgot password**

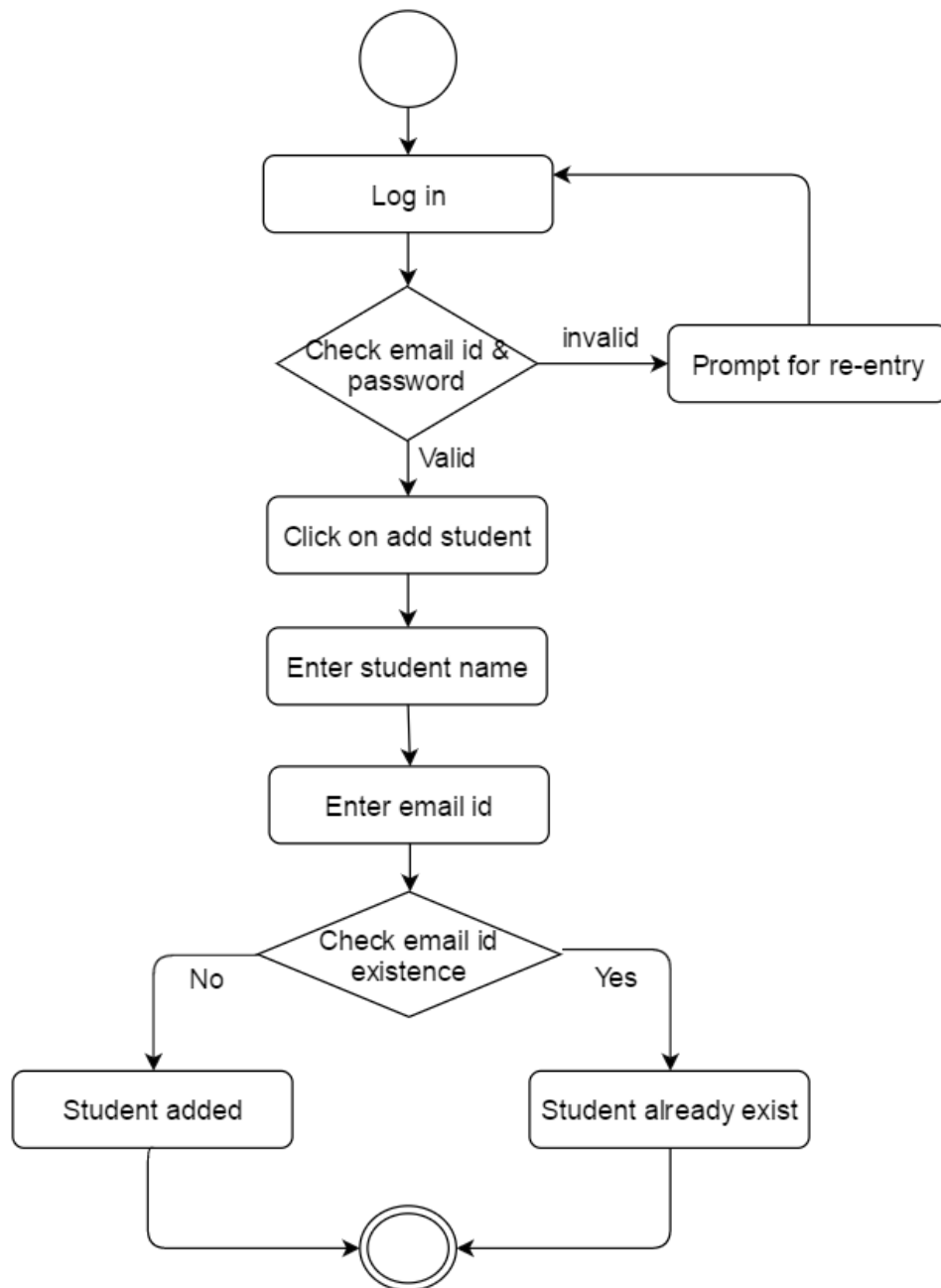
Following is the swim lane diagram of use case Forgot password.



**Figure 15: Swim Lane for Forgot password**

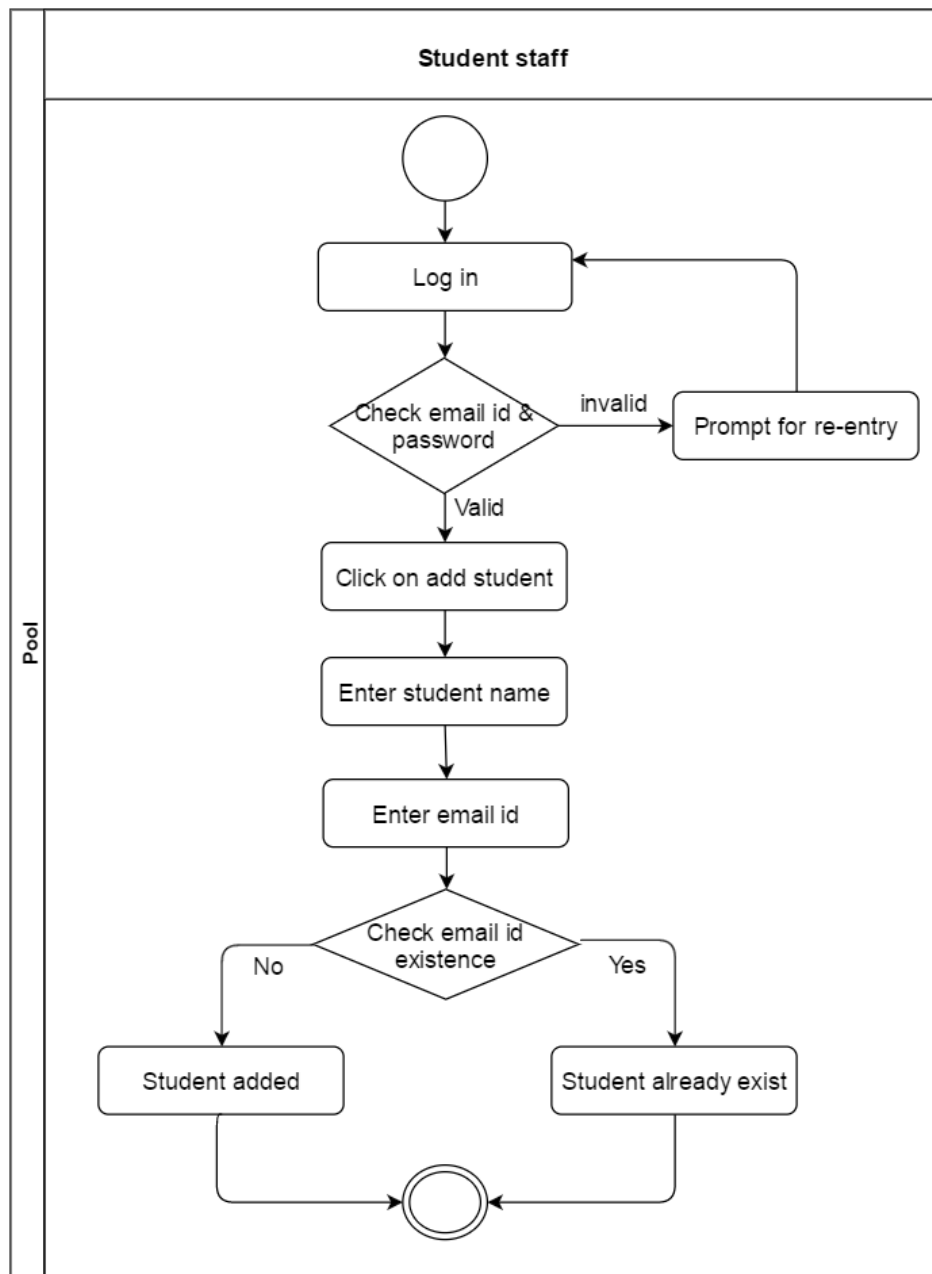
Following is the activity diagram of use case Add student.

#### Usecase 4: Add student



**Figure 16: Activity for Add student**

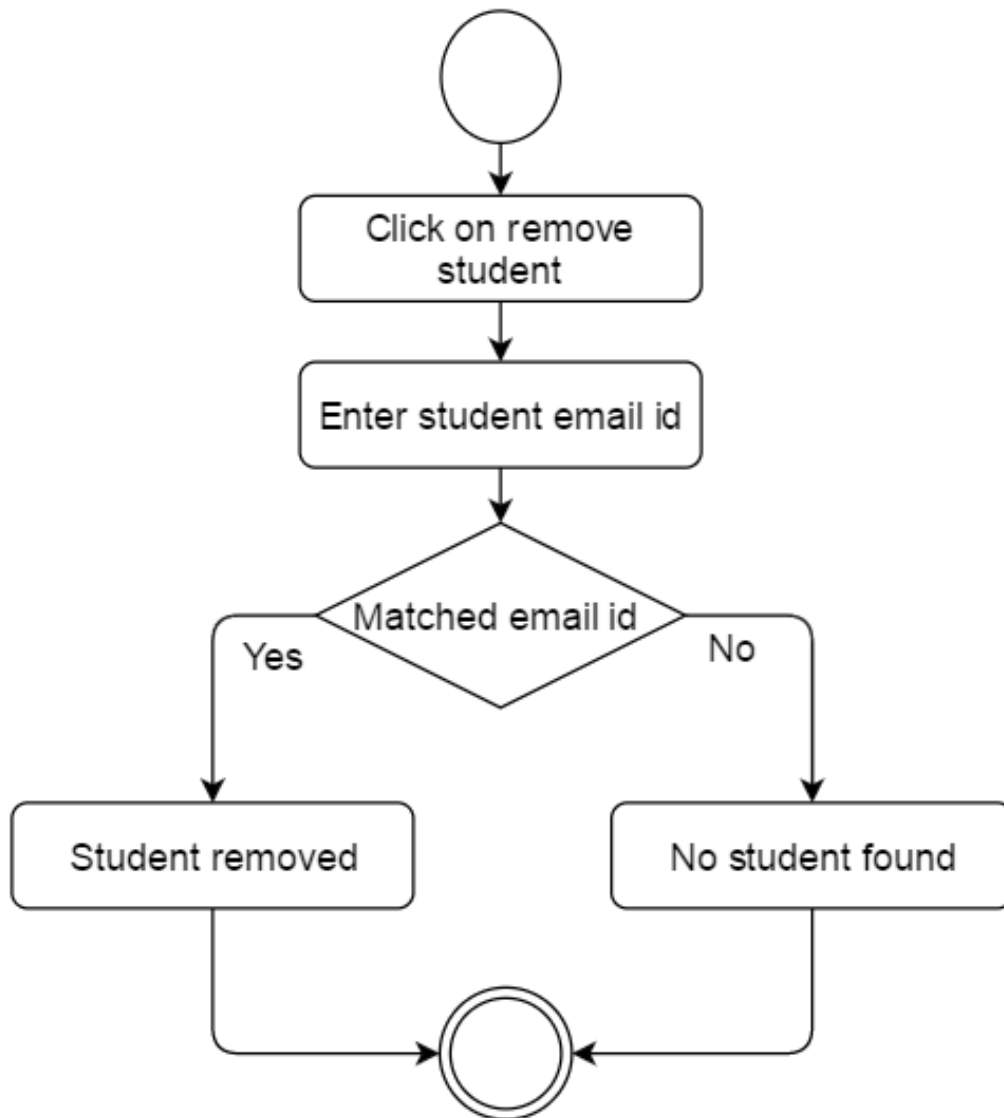
Following is the swim lane diagram of use case Add student.



**Figure 17: Swim Lane for Add student**

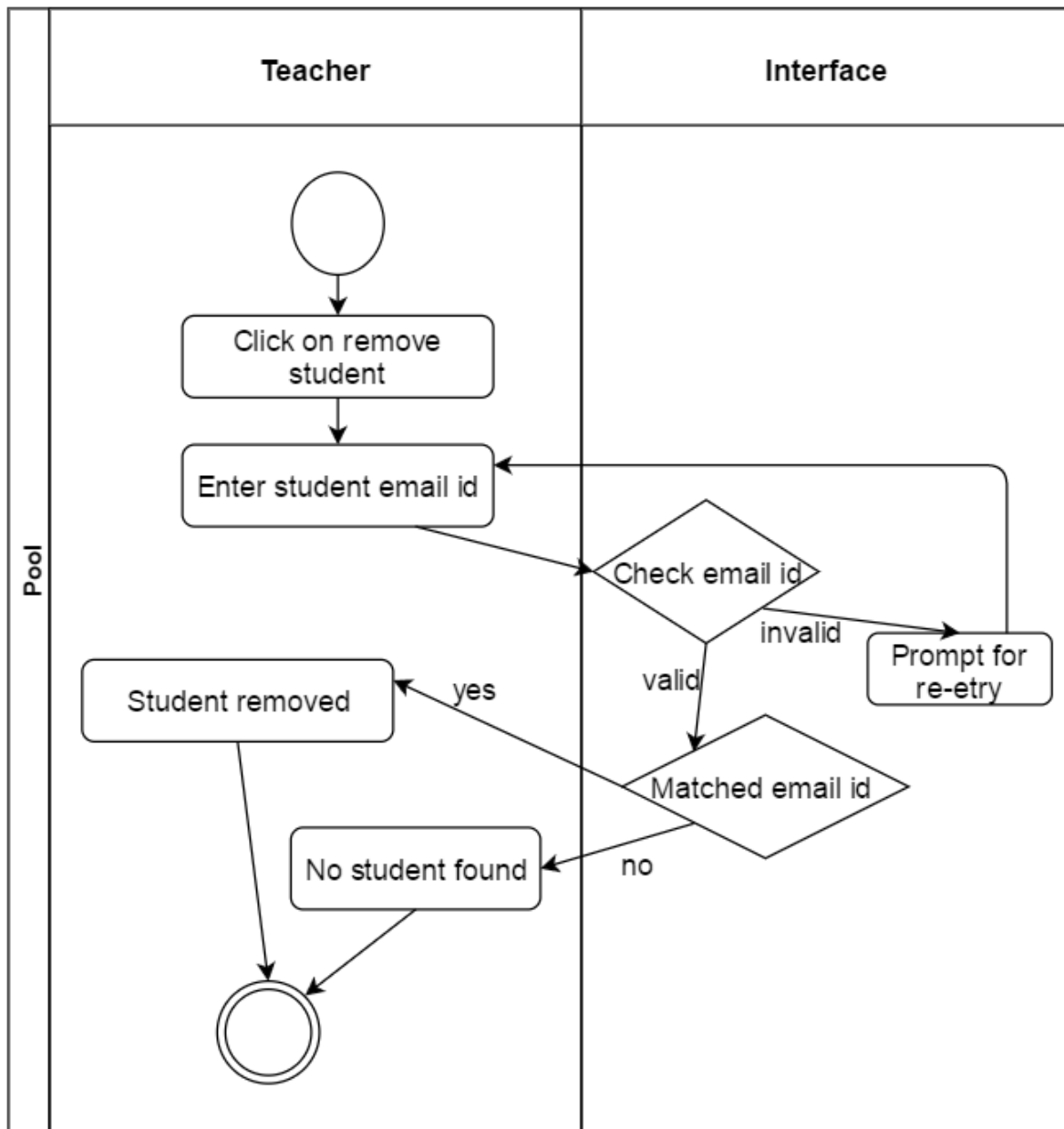
Following is the activity diagram of use case Remove student.

### Usecase 5: Remove student



**Figure 18: Activity for Remove student**

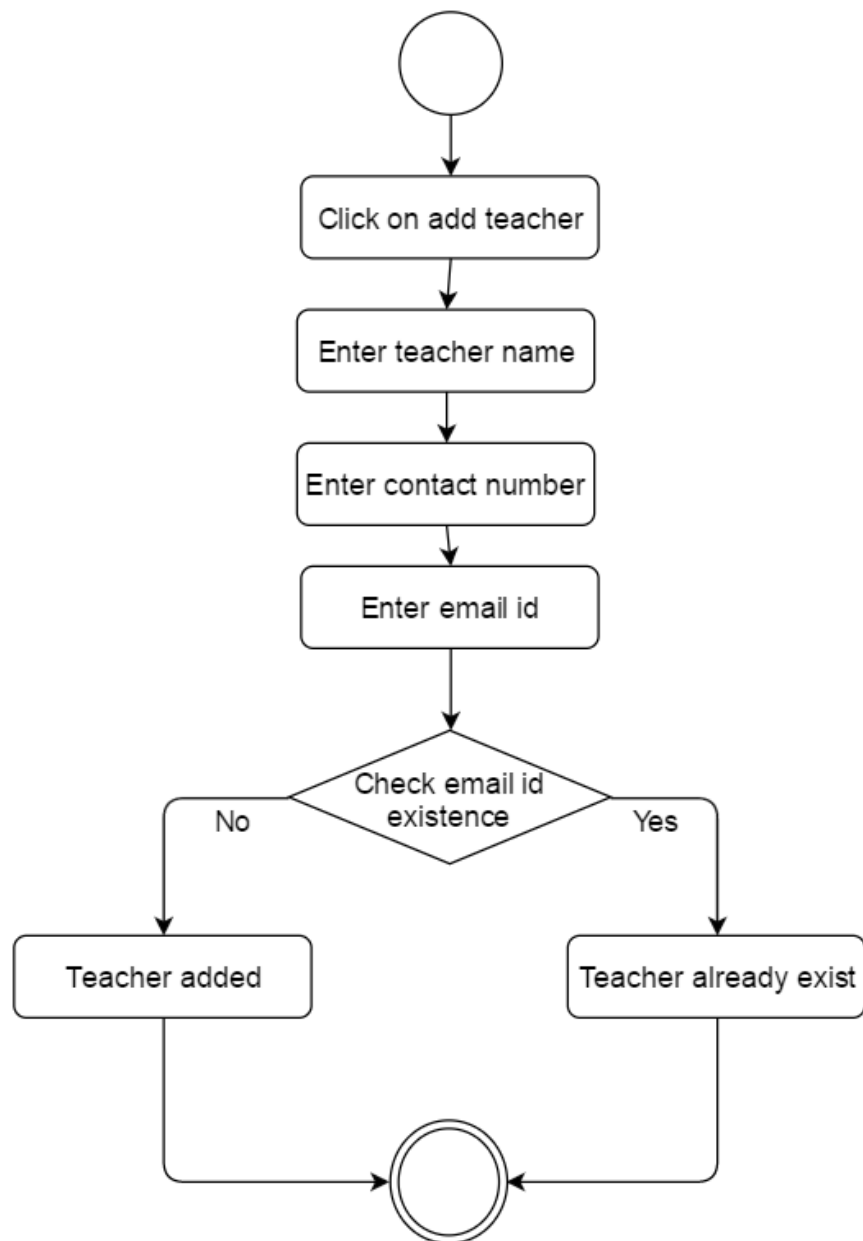
Following is the swim lane diagram of use case Remove student.



**Figure 19: Swim Lane for Remove student**

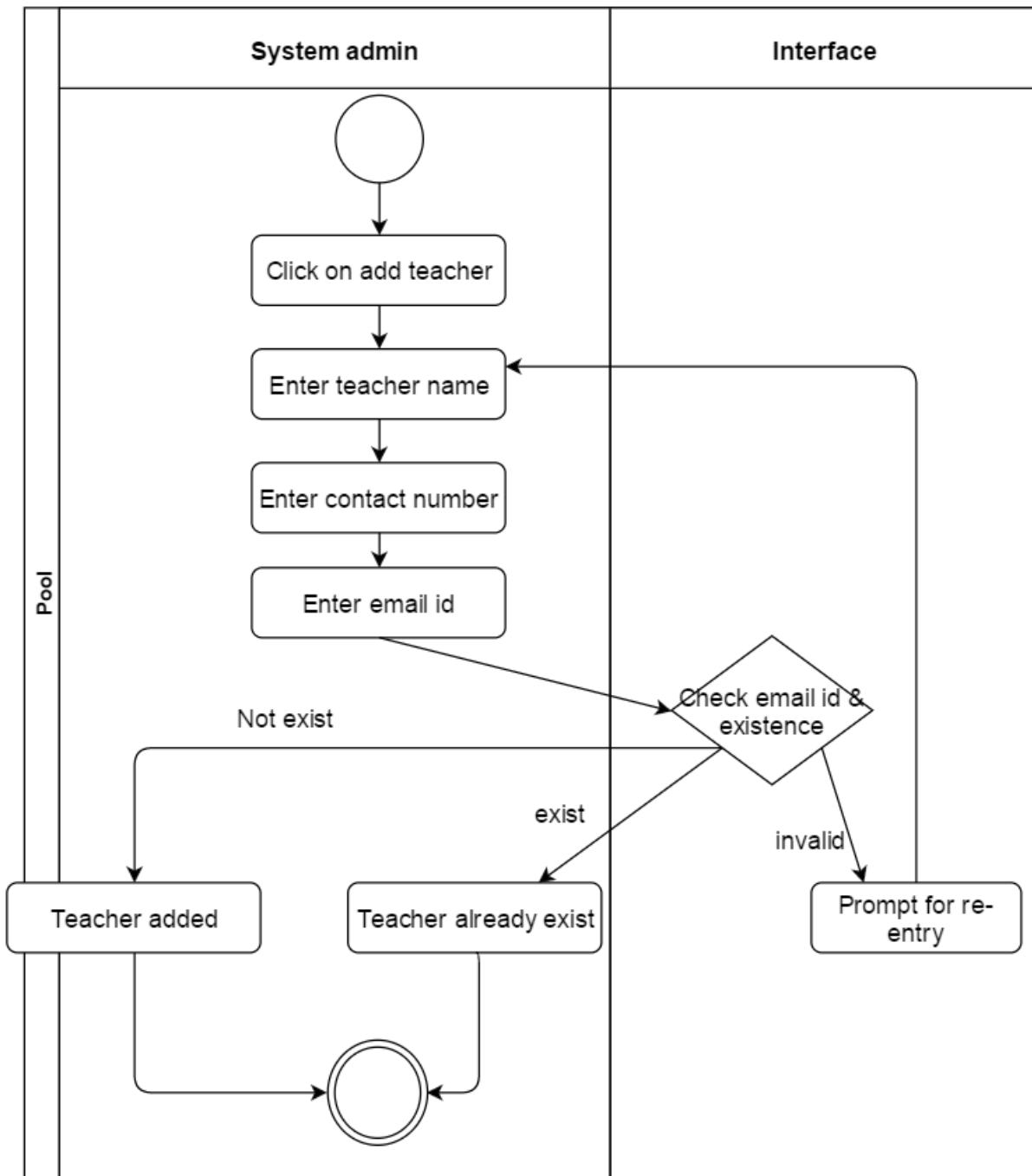
Following is the activity diagram of use case Add teacher.

### Usecase 6: Add teacher



**Figure 20: Activity for Add teacher**

Following is the swim lane diagram of use case Add teacher.

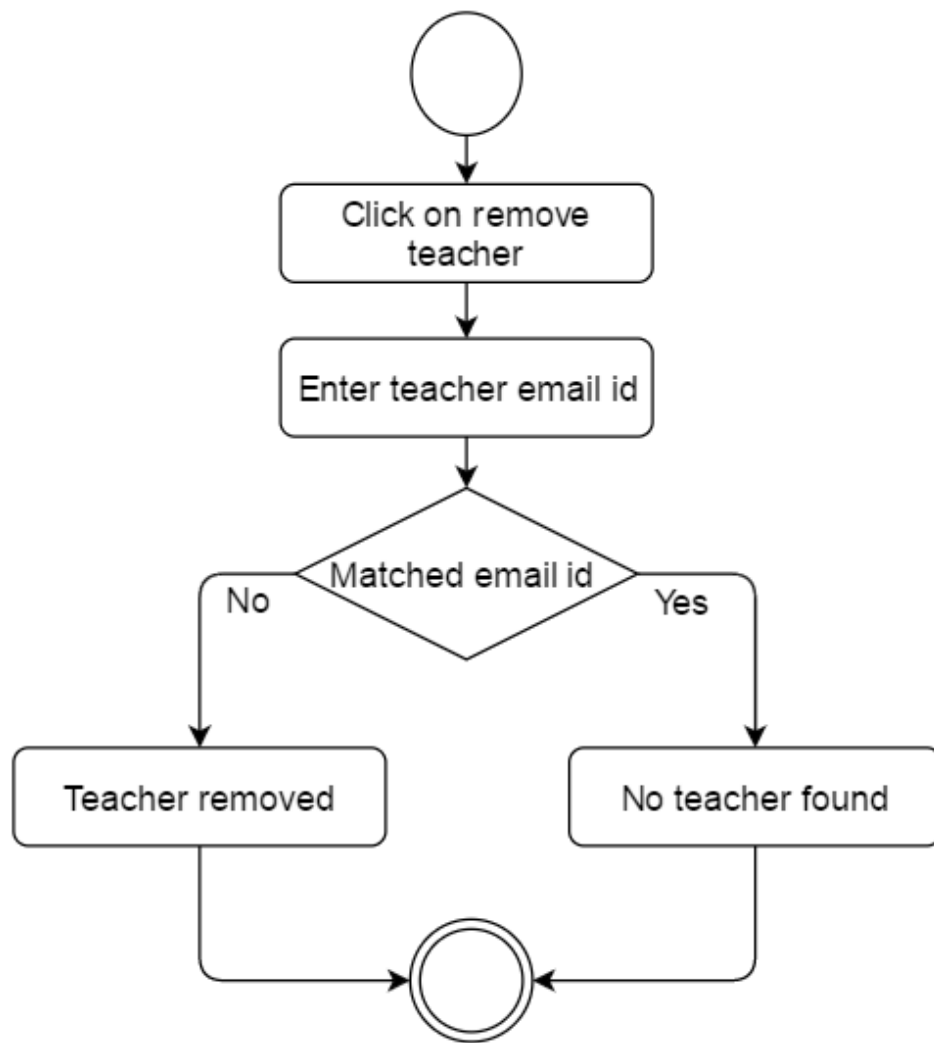


**Figure 21: Swim lane for Add teacher**



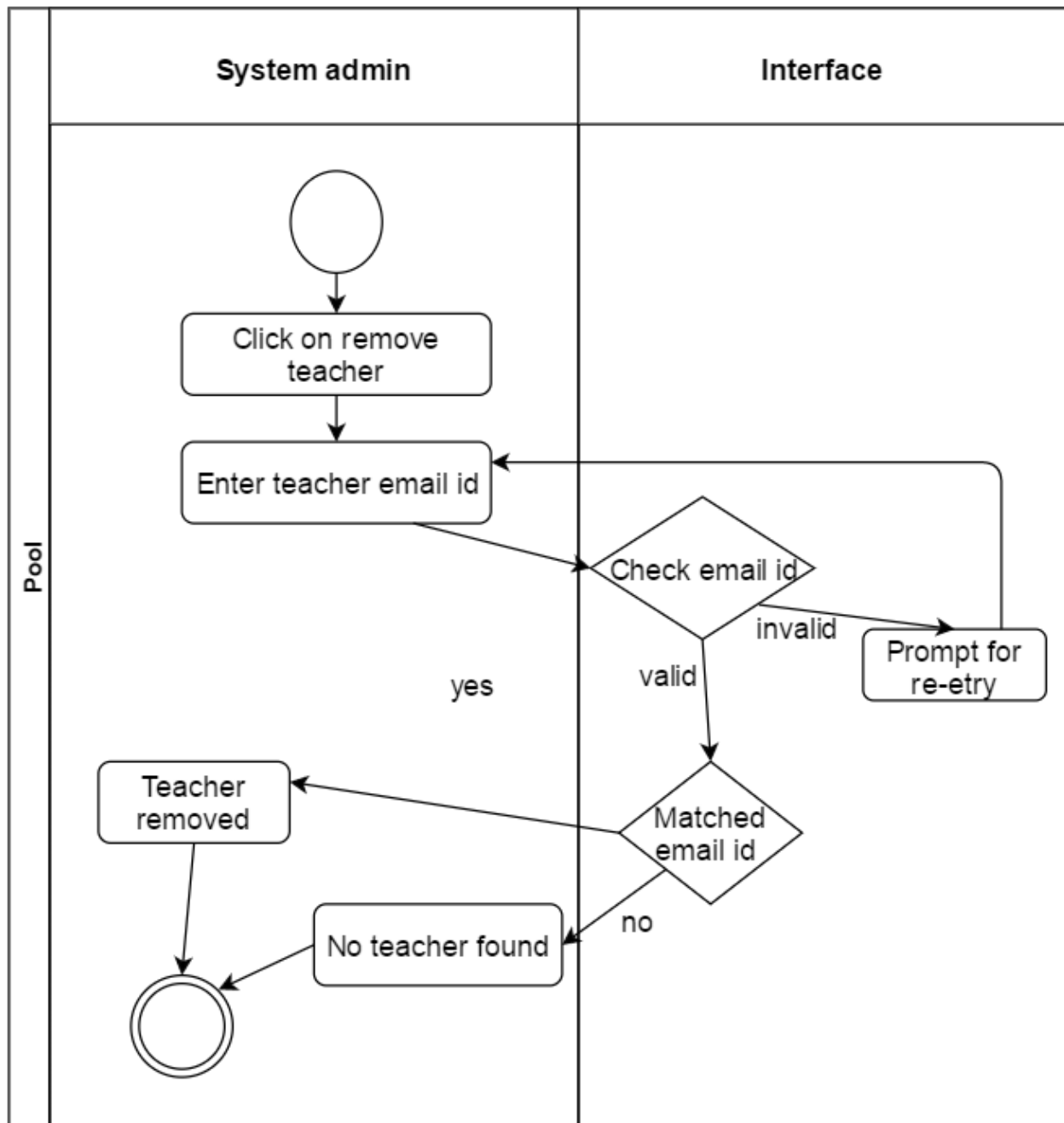
Following is the activity diagram of use case Remove teacher.

### Usecase 7: Remove teacher



*Figure 22: Activity for Remove teacher*

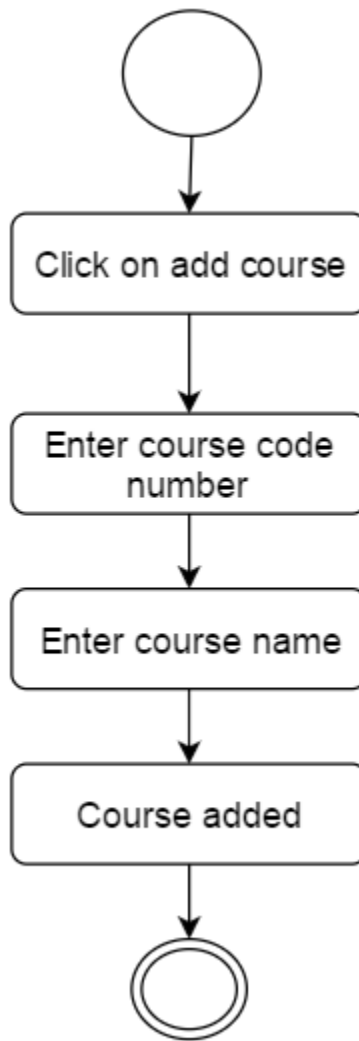
Following is the swim lane diagram of use case Remove teacher.



*Figure 23: Swim lane for Remove teacher*

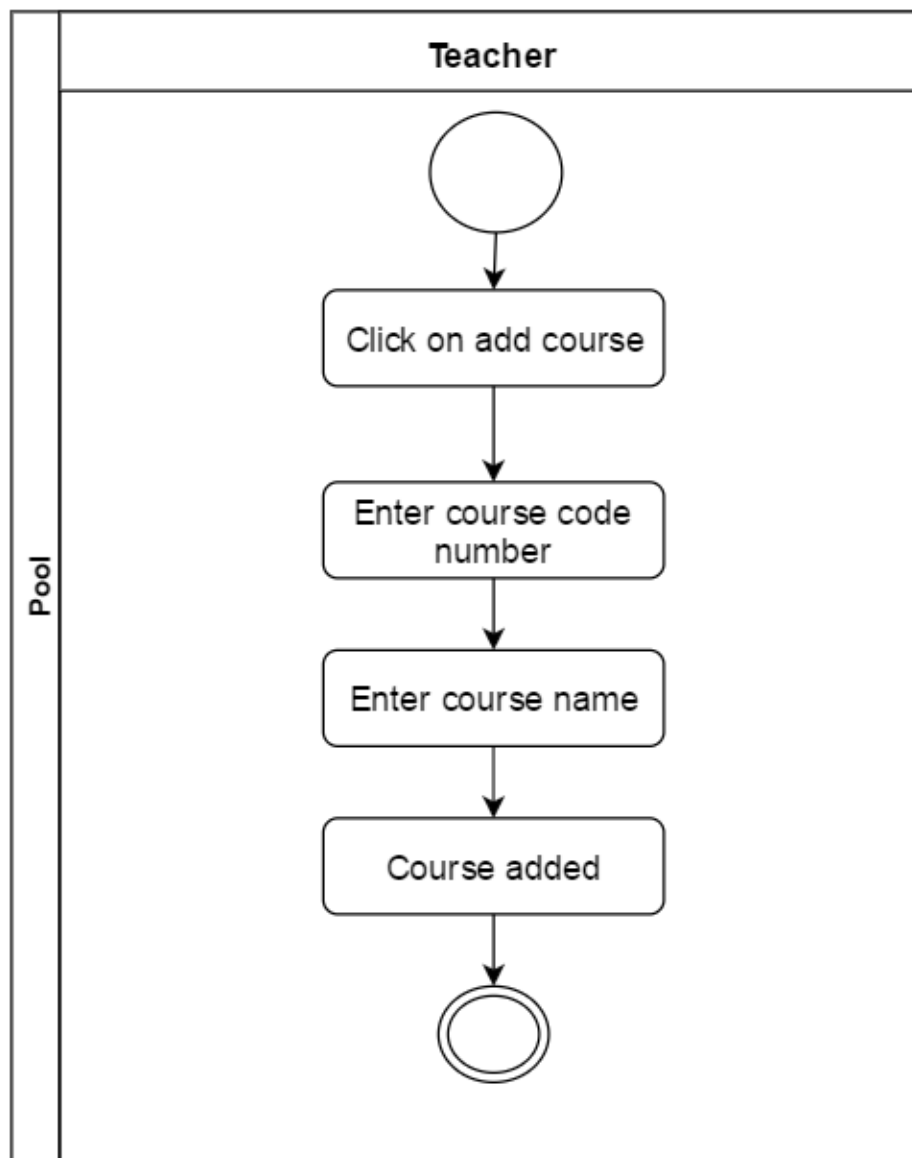
Following is the activity diagram of use case Add course.

### Usecase 8: Add course



*Figure 24: Activity for Add course*

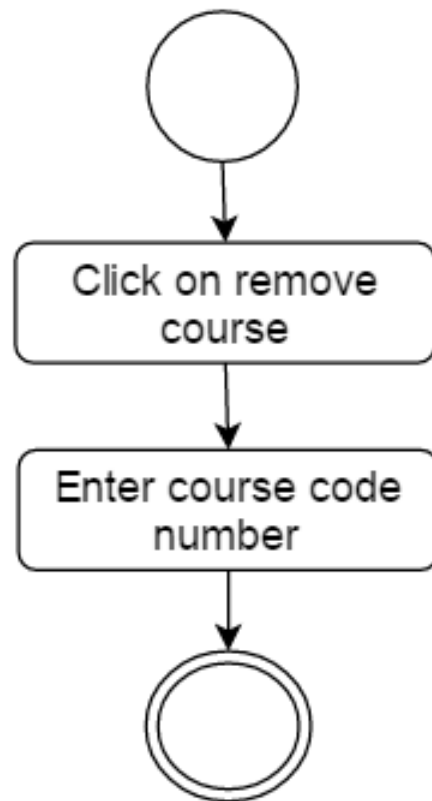
Following is the swim lane diagram of use case Add course.



**Figure 25: Swim Lane for Add course**

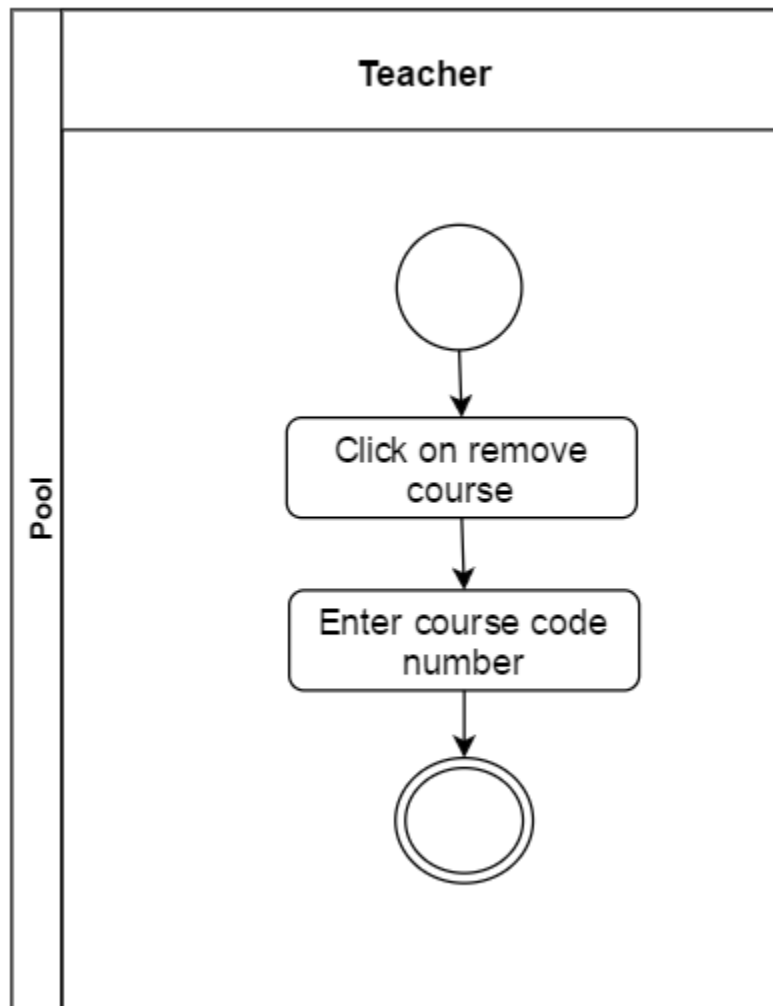
Following is the activity diagram of use case Remove course.

### Usecase 8: Remove course



*Figure 26: Activity for Remove course*

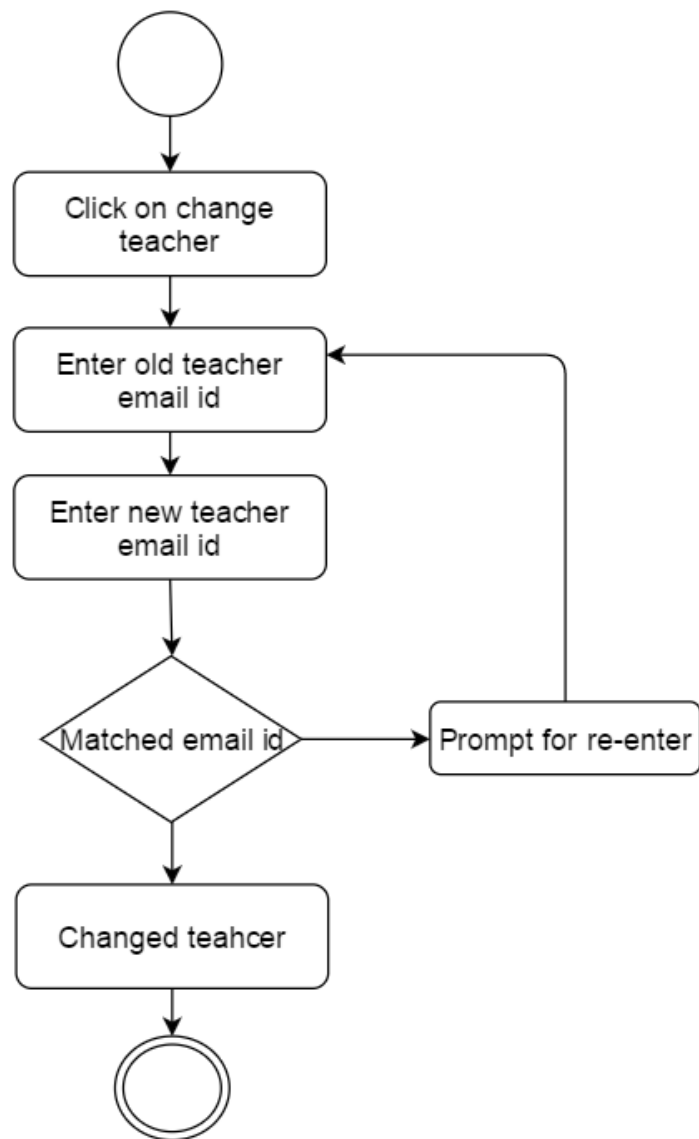
Following is the swim lane diagram of use case Remove course.



**Figure 27: Swim Lane for Remove course**

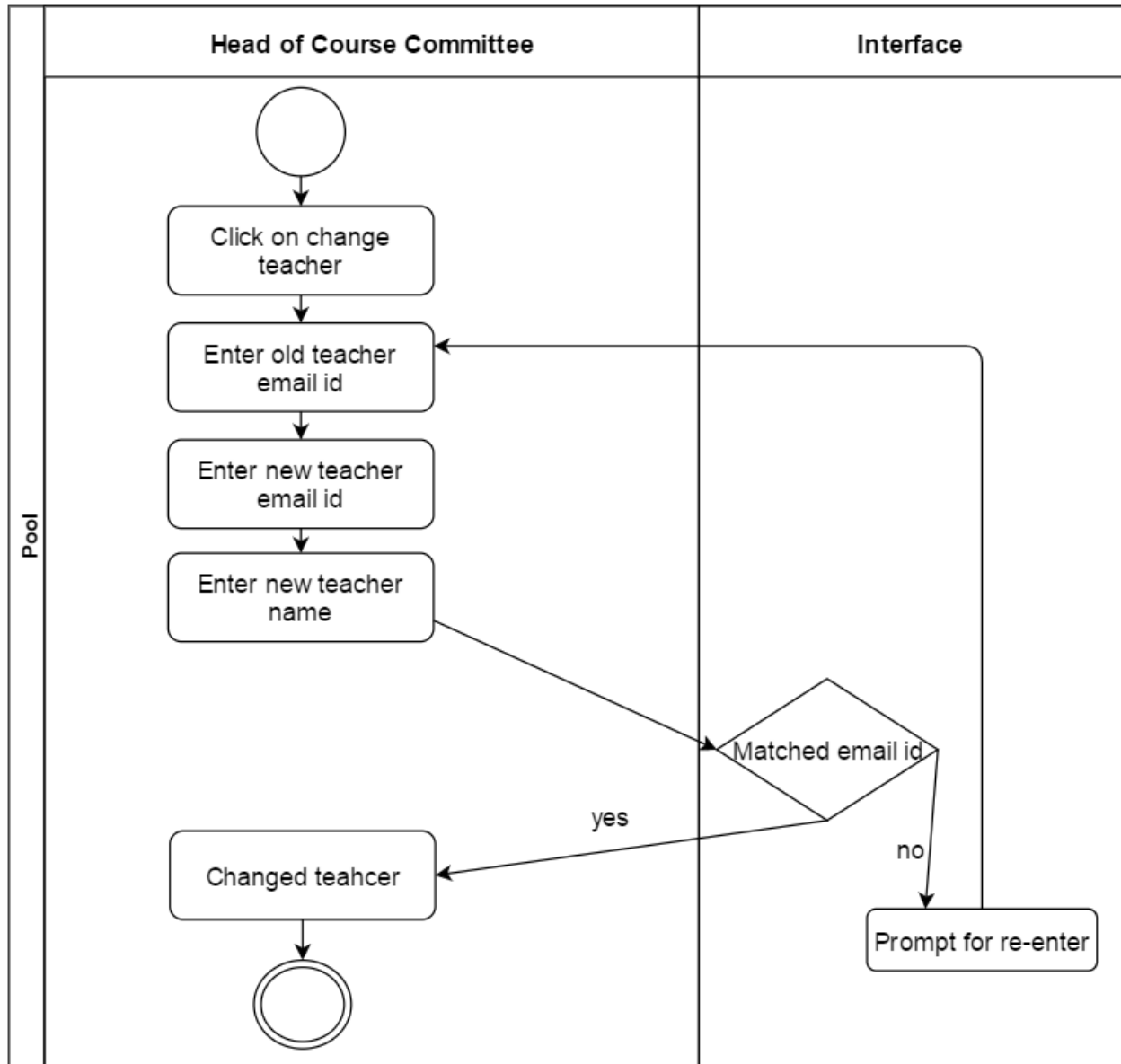
Following is the activity diagram of use case Change teacher.

### Usecase 9: Change teacher



**Figure 28: Activity for Change teacher**

Following is the swim lane diagram of use case Change teacher.

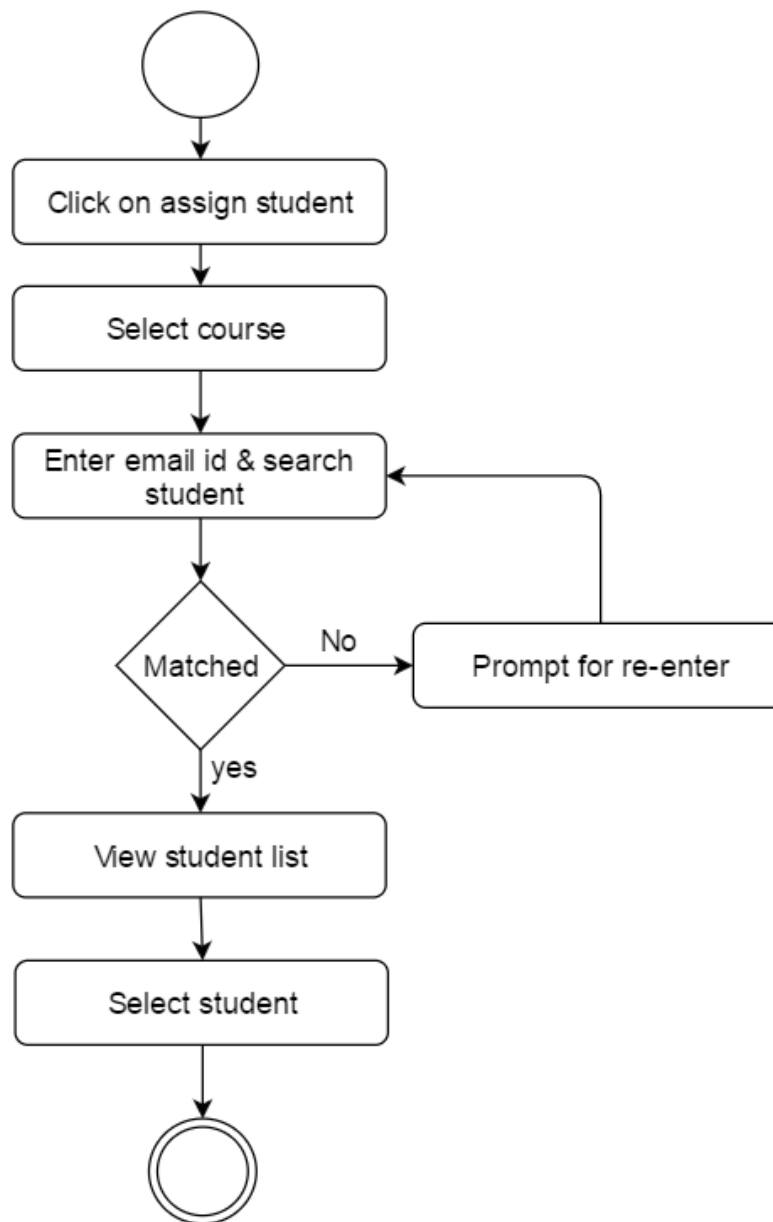


**Figure 29: Swim Lane for Change teacher**



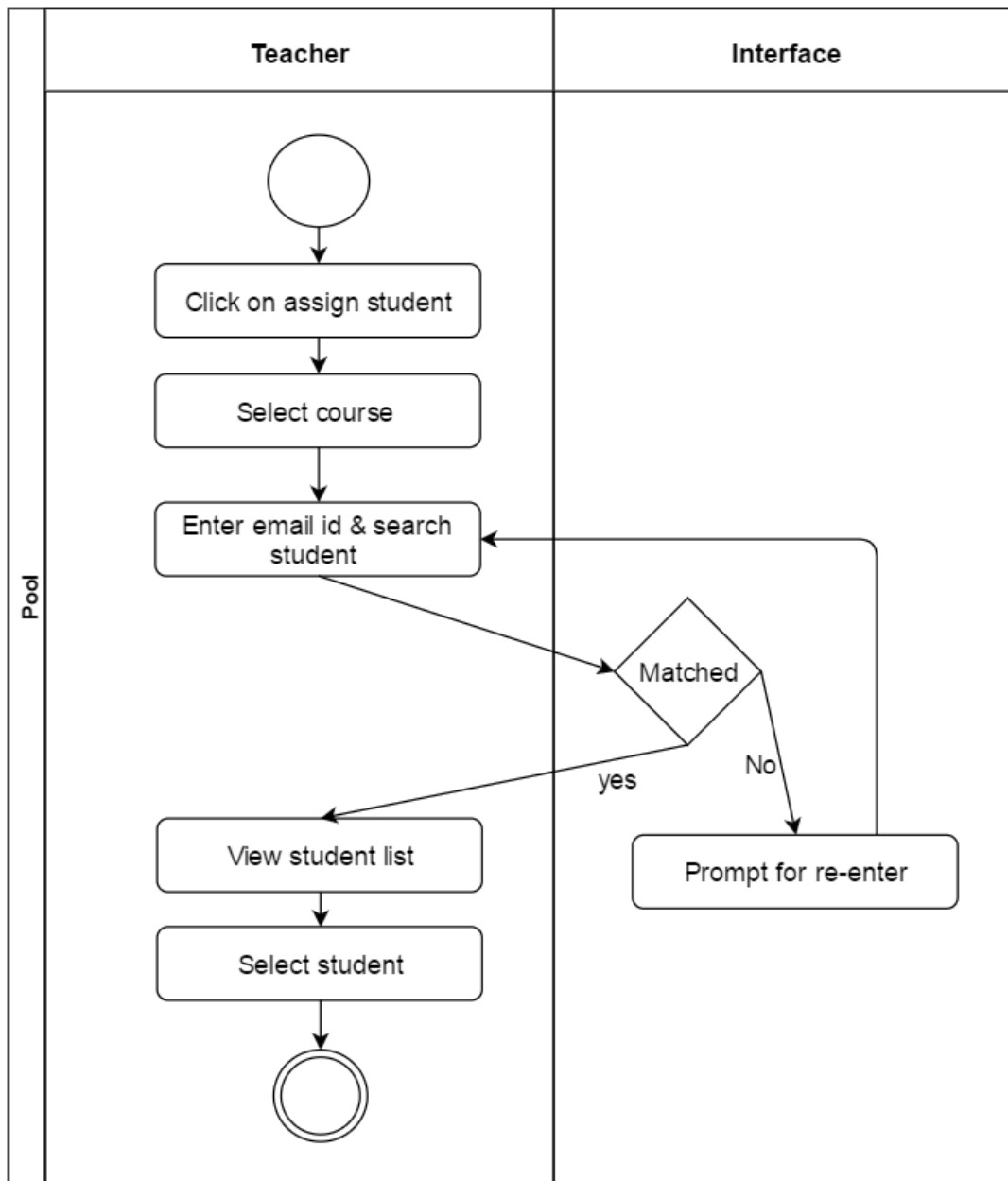
Following is the activity diagram of use case Assign student.

### Usecase 10: Assign student



**Figure 30: Activity for Assign student**

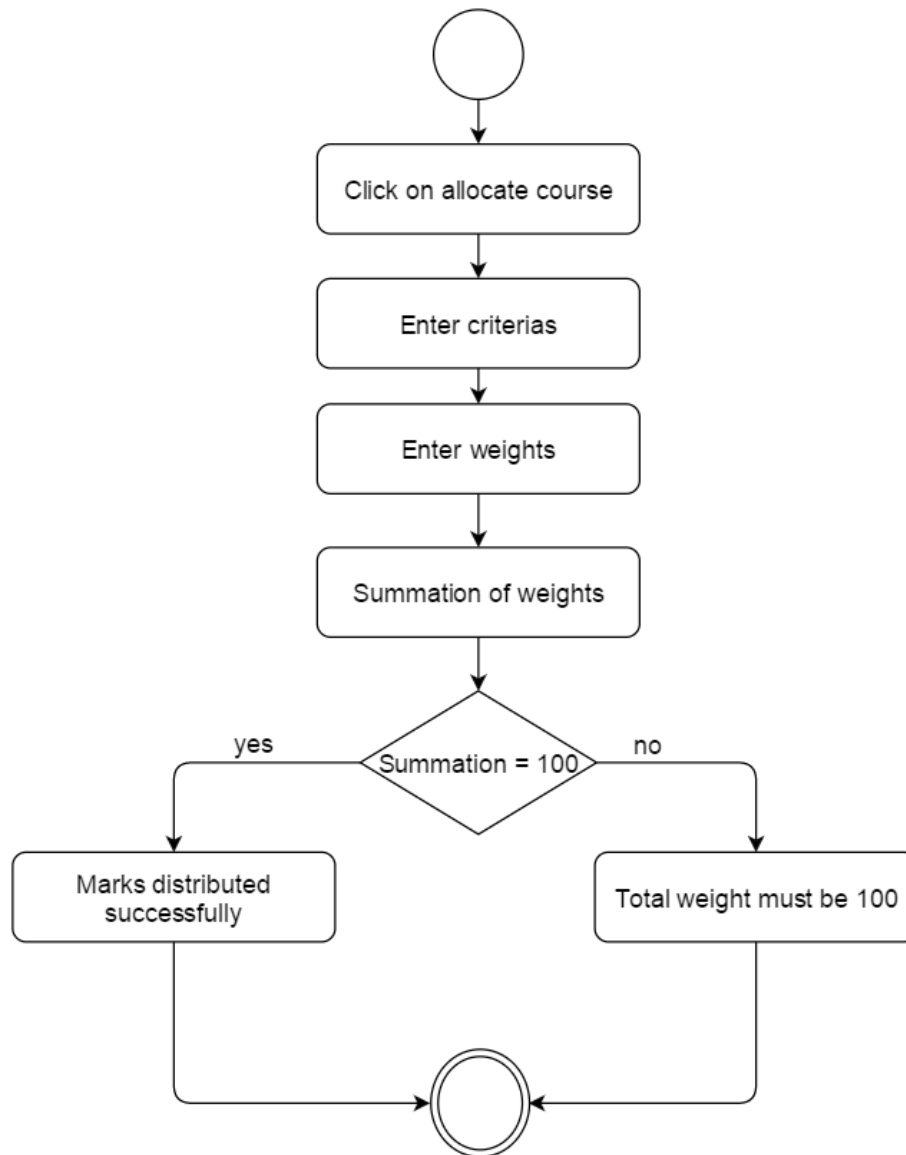
Following is the swim lane diagram of use case Assign student.



**Figure 31: Swim Lane for Assign student**

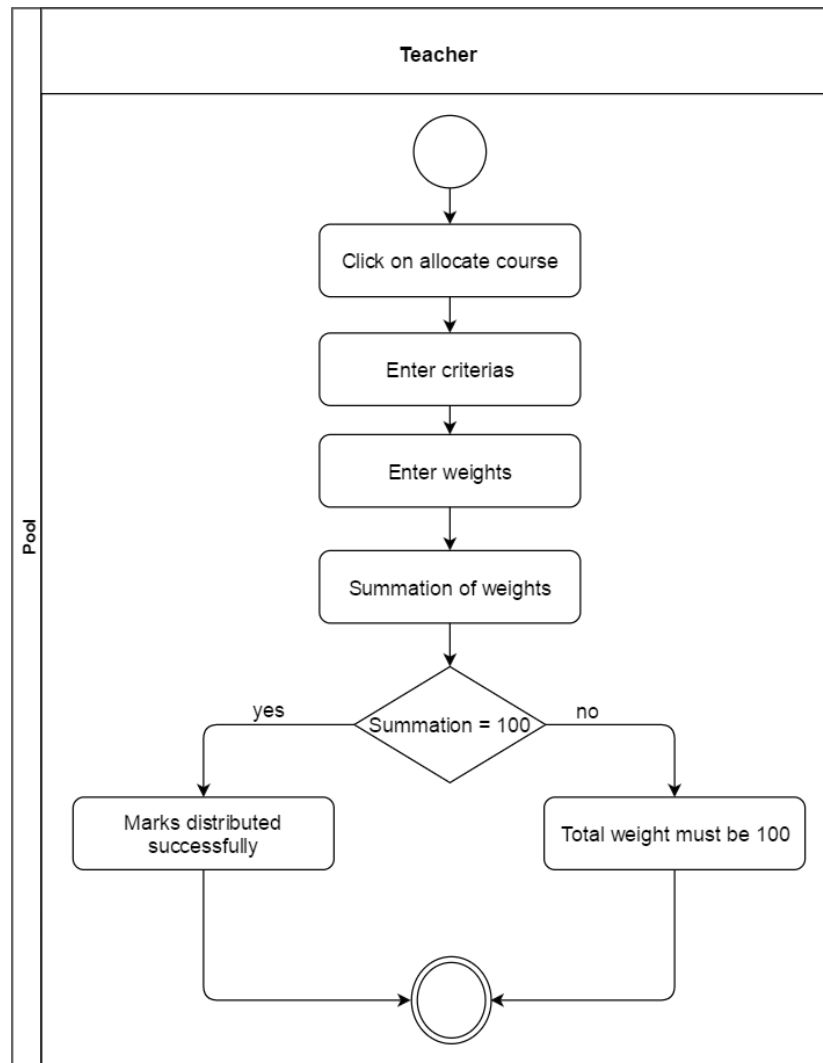
Following is the activity diagram of use case Distribute marks.

### Usecase 11: Distribute marks



**Figure 32: Activity for Distribute marks**

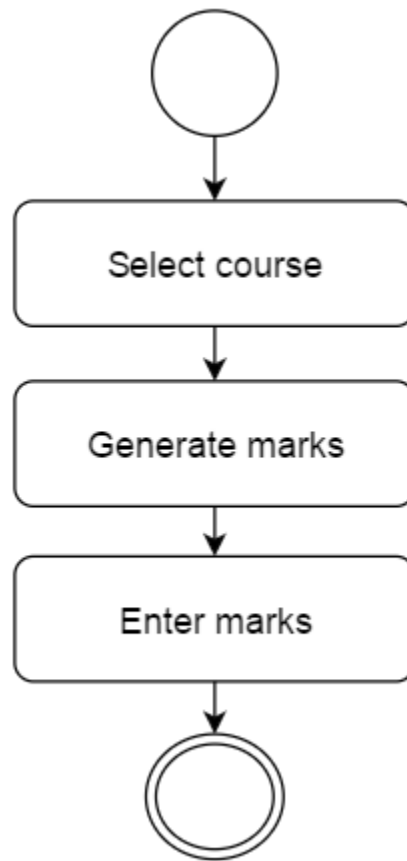
Following is the swim lane diagram of use case Distribute marks.



**Figure 32: Swim Lane for Distribute marks**

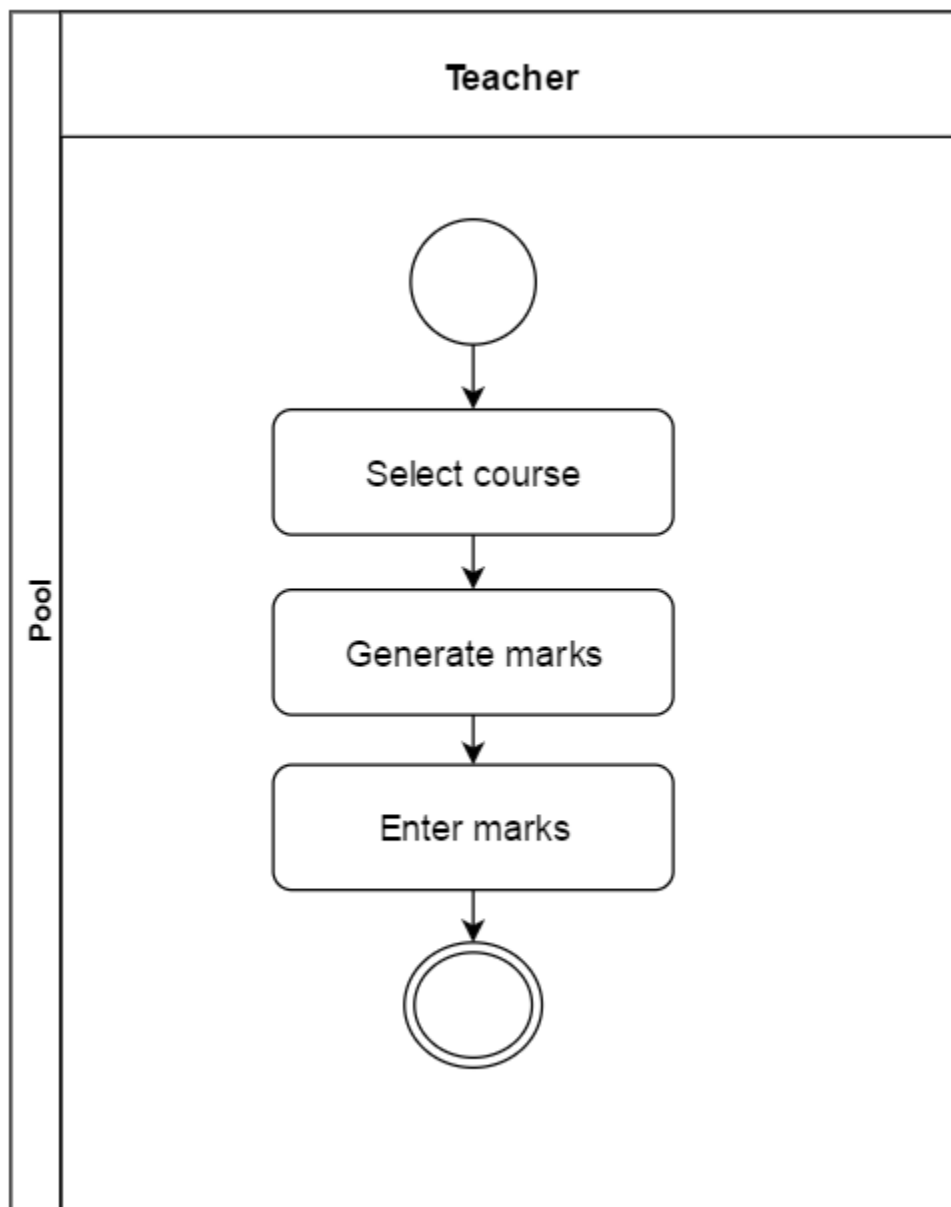
Following is the activity diagram of use case Marks entry.

**Usecase 12: Marks entry**



*Figure 32: Activity for Marks entry*

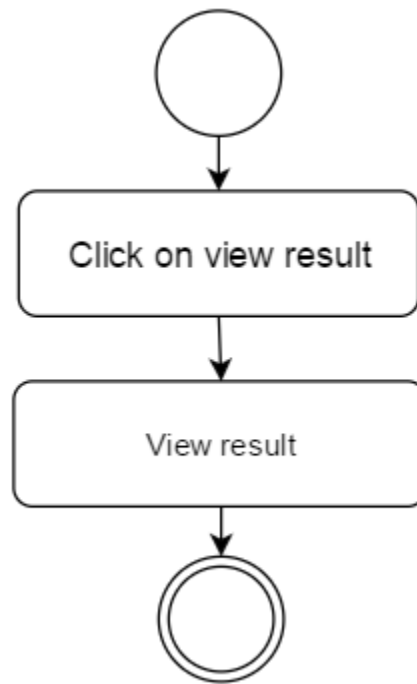
Following is the swim lane diagram of use case Distribute marks.



**Figure 32: Swim Lane for Marks entry**

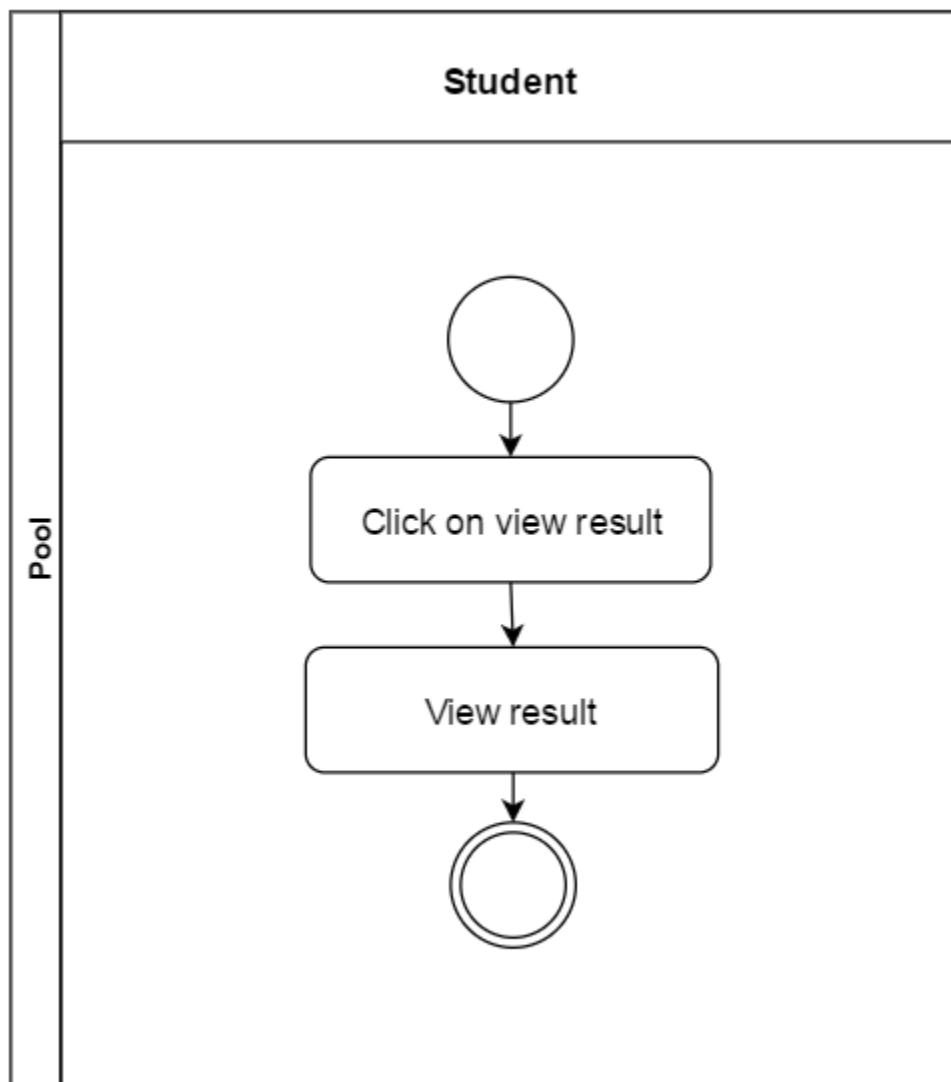
Following is the activity diagram of use case Marks entry.

### Usecase 13: View result



*Figure 32: Activity for View result*

Following is the swim lane diagram of use case View result.

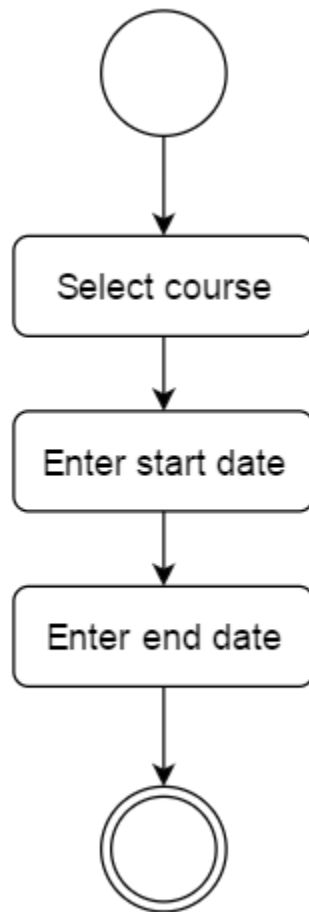


**Figure 32: Swim Lane for View result**



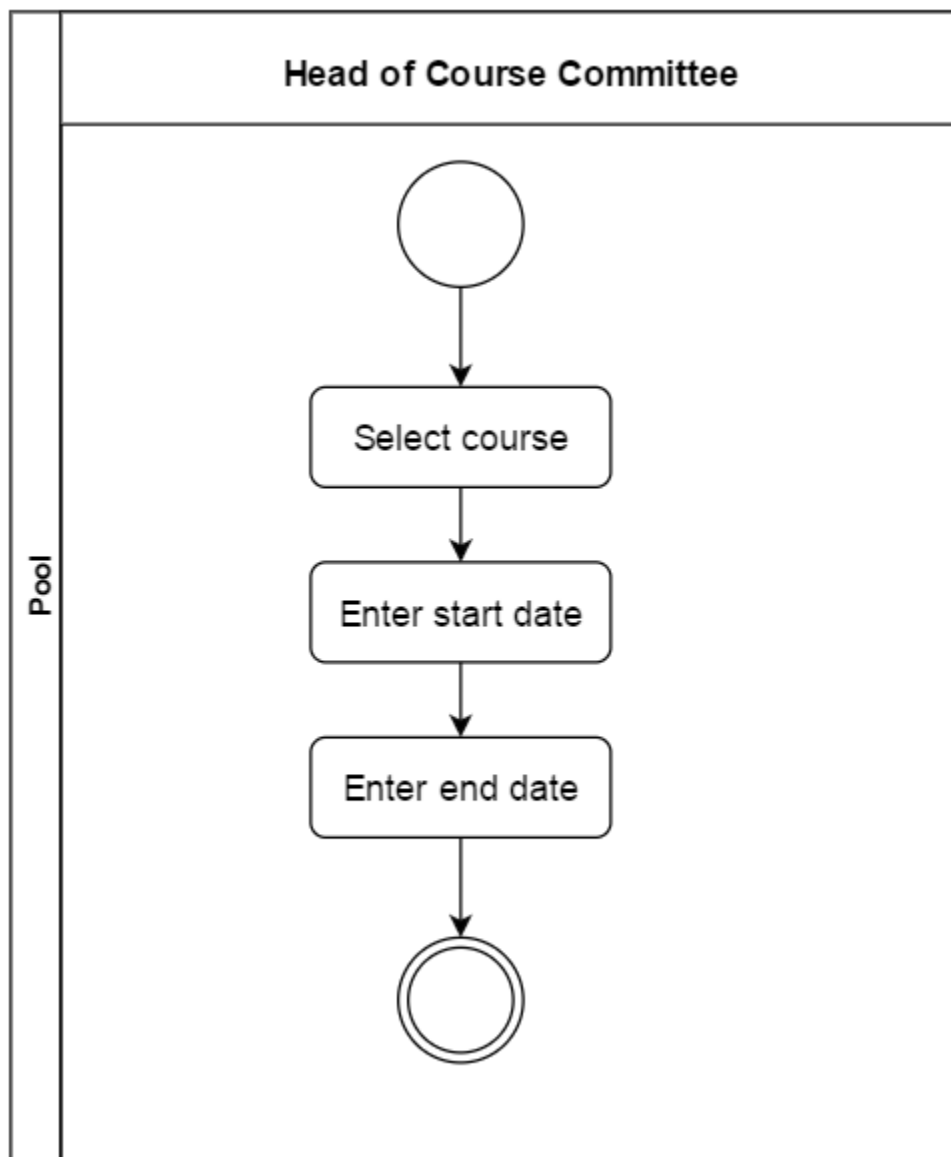
Following is the activity diagram of use case Set deadline.

#### **Usecase 14: Set deadline**



*Figure 32: Activity for Set deadline*

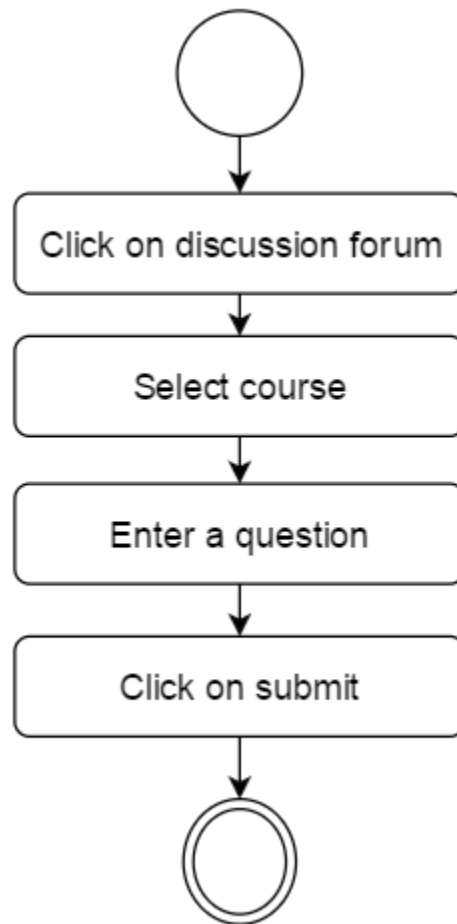
Following is the swim lane diagram of use case Set deadline.



**Figure 32: Swim Lane for Set deadline**

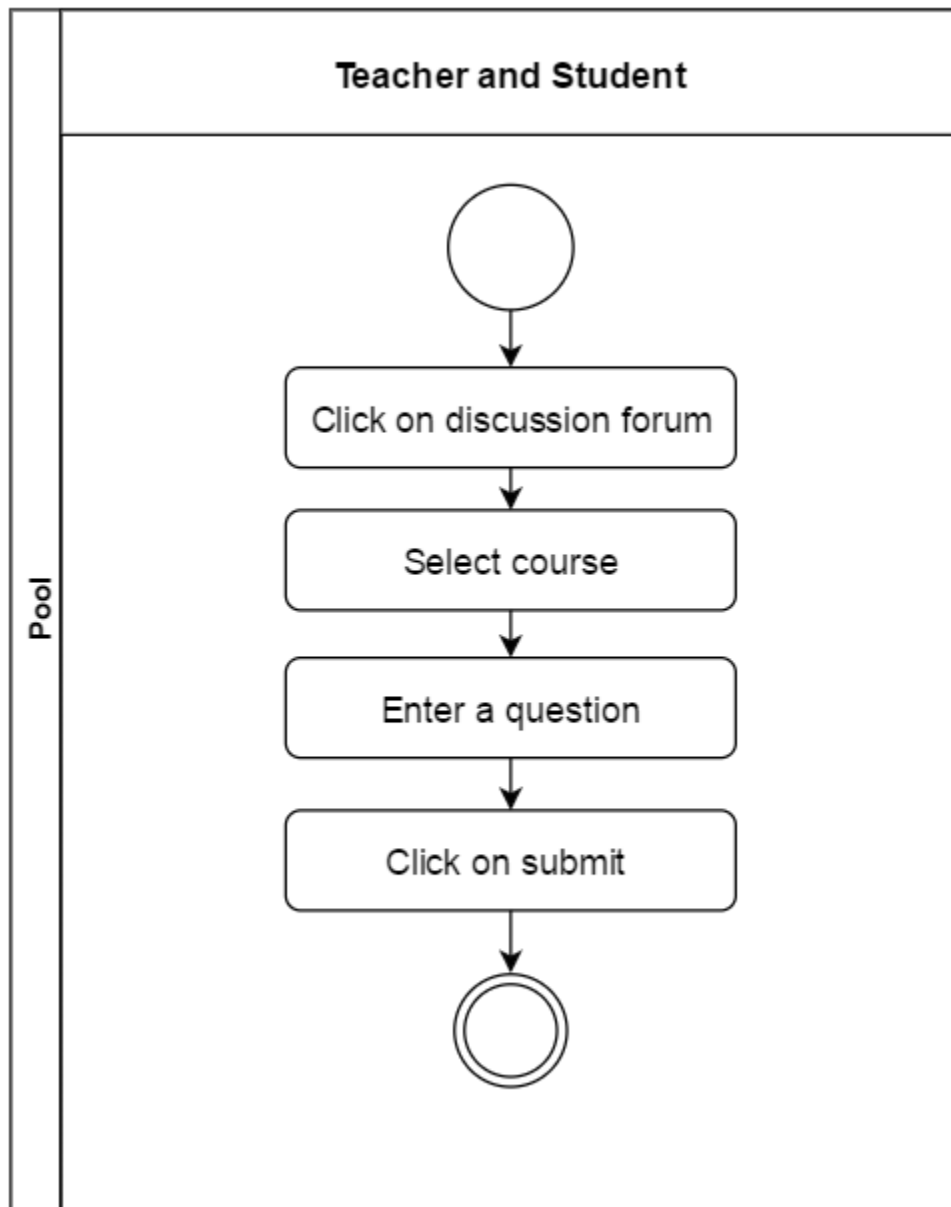
Following is the activity diagram of use case Ask question.

#### Usecase 14: Ask question



*Figure 32: Activity for Ask question*

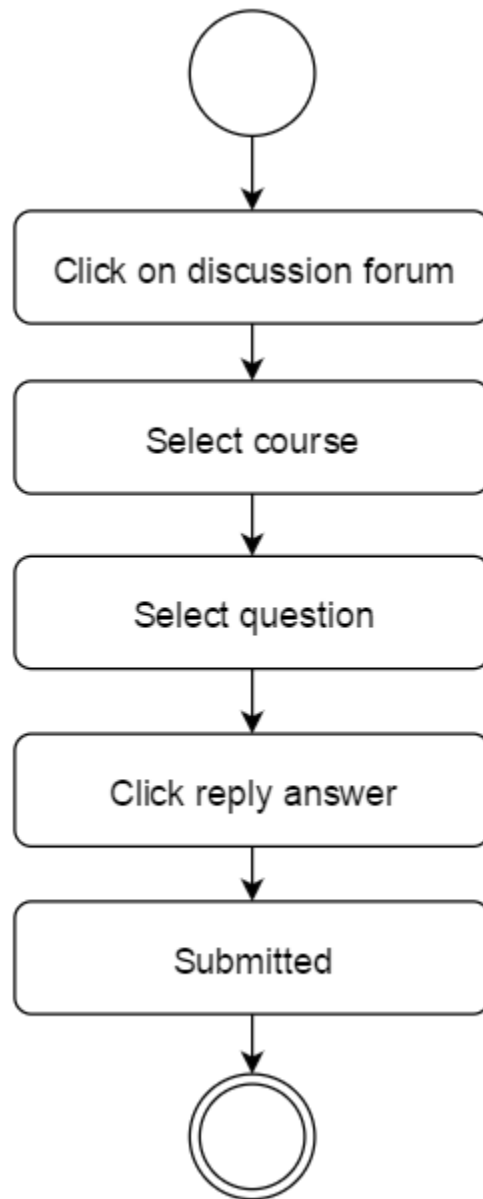
Following is the swim lane diagram of use case Ask question.



**Figure 32: Swim Lane for Ask question**

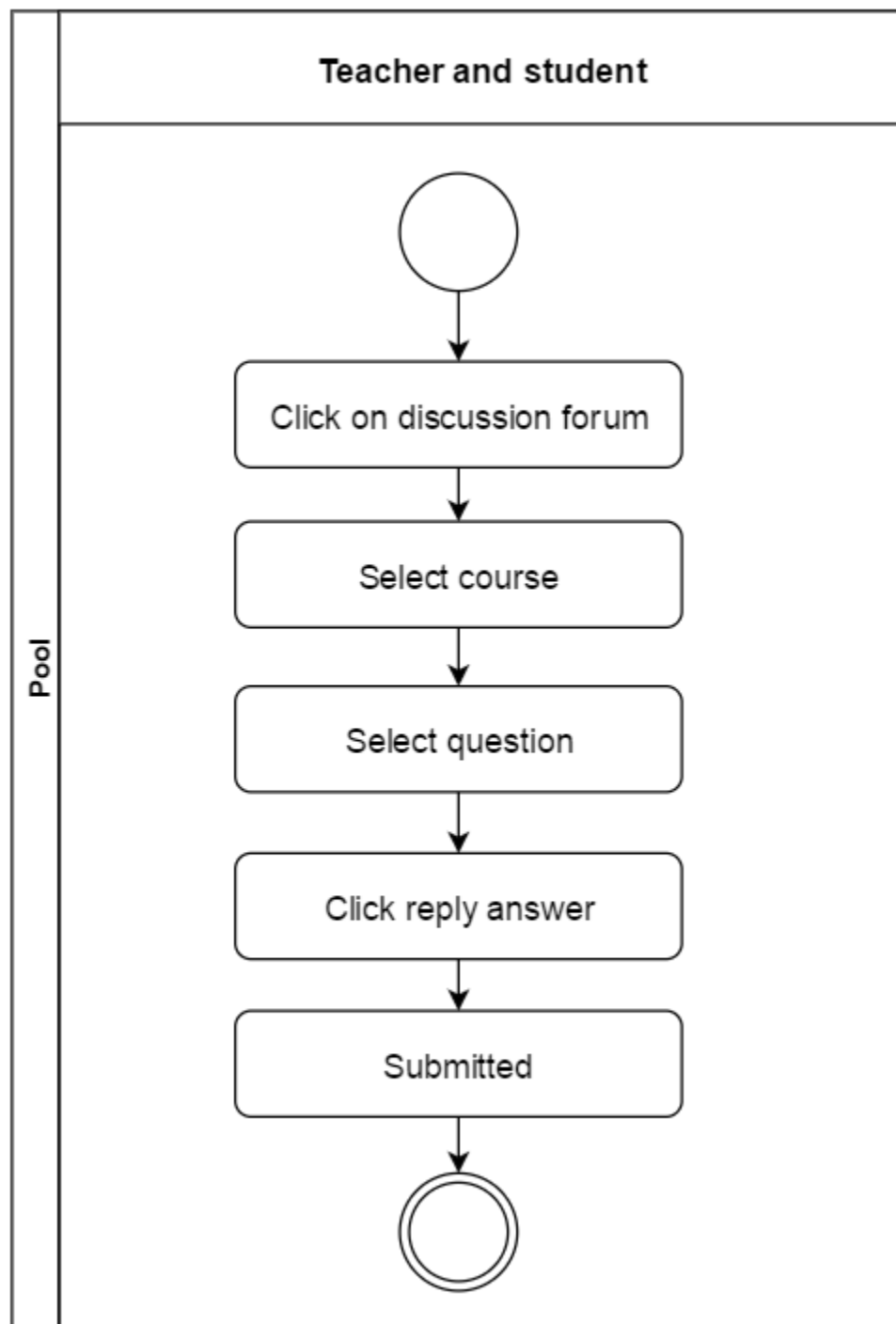
Following is the activity diagram of use case Submit answer.

**Usecase 15: Submit answer**



*Figure 32: Activity for Submit answer*

Following is the swim lane diagram of use case Submit answer.



**Figure 32: Swim Lane for Submit answer**

# Chapter 5: Data Model

In this chapter, we will discuss about the data models of our system.

## 5.1 Data Modeling Concept

If software requirements include the need to create, extend or interface with a data base or if complex data structures must be constructed and manipulated, the software team choose to create data model as part of overall requirements modeling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information that how the data objects are entered, stored, transformed and produced within the system.

## 5.2 Data Objects

A data object is representation of composite information that must be understood by software. Here, composite information means that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

### 5.2.1 Grammatical parsing (Noun identify)

We identified all the nouns whether they are in problem space or in solution space from our usage scenario and categorized them according to their attributes.

<b>Nouns</b>	<b>Problem space/solution space</b>	<b>Attributes</b>
1. Student	S	2,3
2. Name	S	
3. Email	S	
4. Teacher	S	2,3
5. Admin	S	2,3
6. Head of Course Committee	S	2,3
7. Student Staff	S	2,3
8. Course	S	8,9,10,12,13
9. Course Code	S	
10. Course Description	S	
11. Deadline	P	
12. Start Date	S	
13. End Date	S	
14. Mark	S	
15. Marks	S	14
16. Distribution	S	17, 19, 20
17. Distribution Criteria	S	
18. Number Sheet	P	
19. Distribution Weight	S	
20. Viewability	S	
21. Forum	P	
22. Discussion	P	
23. Question	S	25
24. Answer	S	26
25. Question body	S	
26. Answer body	S	



### 5.2.2 Identify Data Objects

Nouns having attributes are selected as data object. Those who doesn't have any attributes have covered under the data objects.

**Data Object:** Admin

**Attributes:**

- Name
- Email

**Data Object:** Head of Course Committee

**Attributes:**

- Name
- Email

**Data Object:** Teacher

**Attributes:**

- Name
- Email

**Data Object:** Student Staff

**Attributes:**

- Name
- Email

**Data Object:** Student

**Attributes:**

- Name

- Email

**Data Object:** Marks

**Attributes:**

- Mark

**Data Object:** Course

**Attributes:**

- Course Code
- Course Name
- Course description

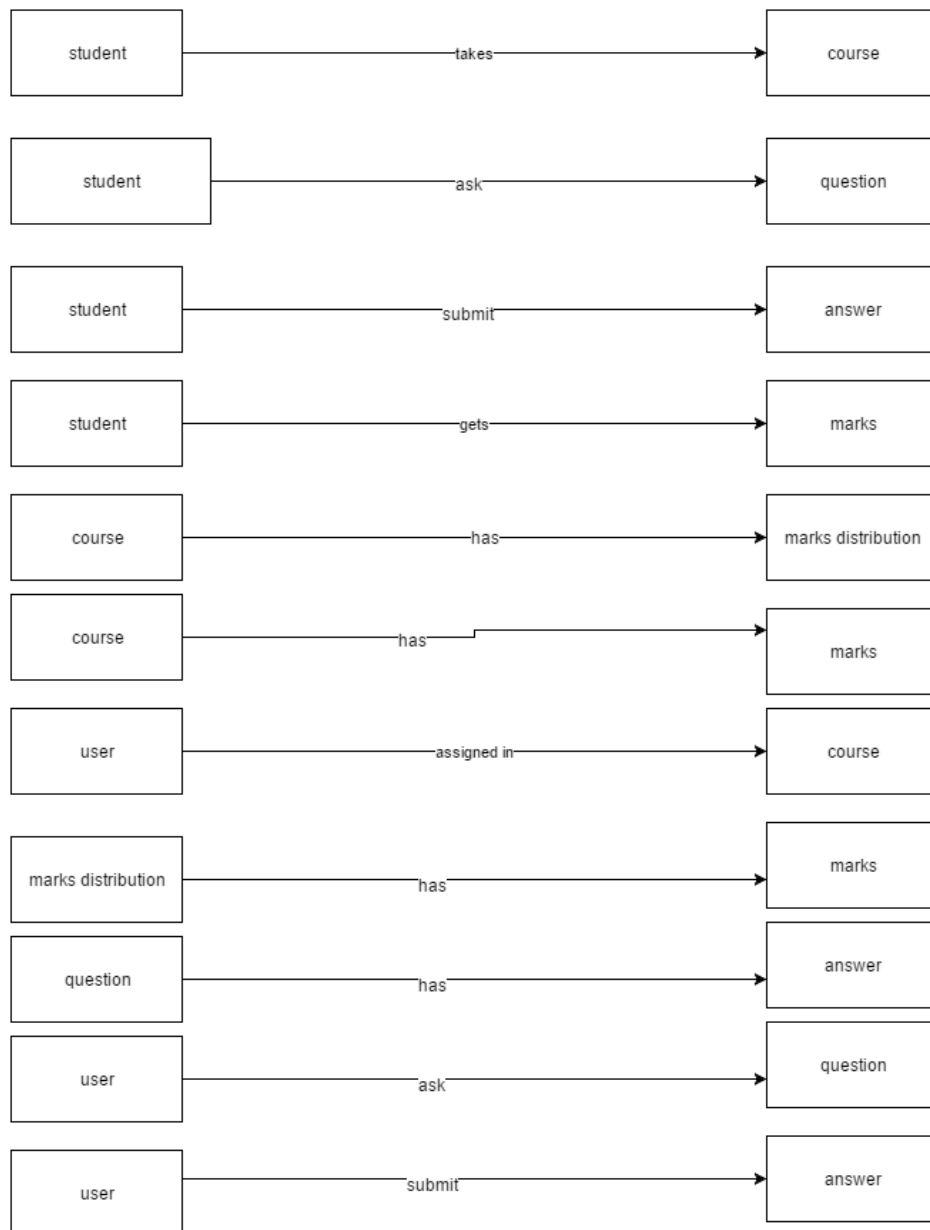
Here, we found that Admin, Teacher, Head of Course Committee, Student Staff have same attributes. So we can consider a new data object named “User” which will hold the attributes of these data objects.

**Data Object:** User

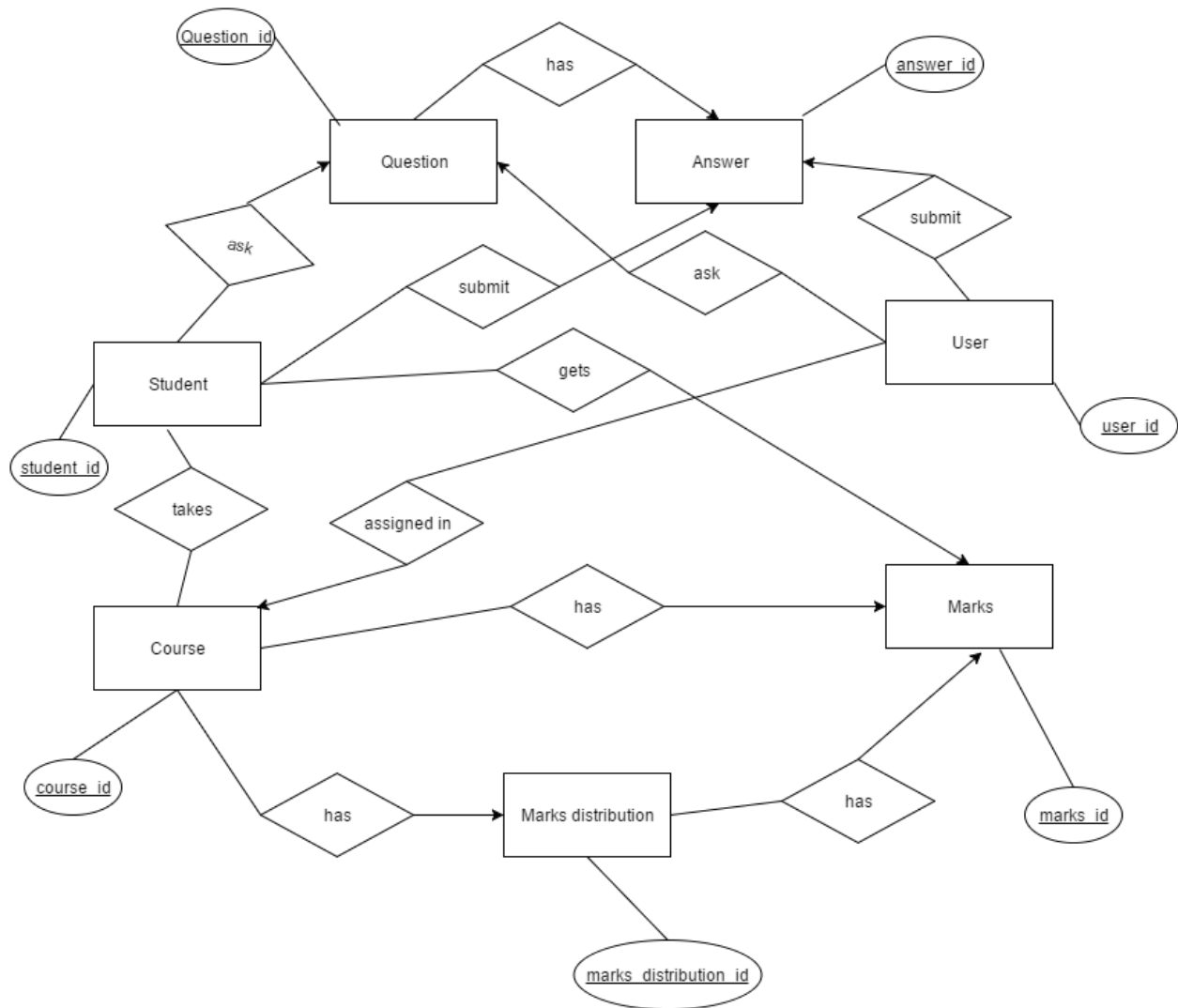
**Attributes:**

- Name
- Email

### 5.2.3 Data Object Relation



## 5.2.4 E-R Diagram



**Figure 24: E-R Diagram**

### 5.3.5 Schema Form (Tables)

User	
<u>User Id</u>	Number(10)
Name	Text(30)
Email	Text(20)
Password	Text(20)
Contact	Number(11)

Course	
<u>Course Id</u>	Number(20)
<u>Course code</u>	Text(7)
Course name	Text(30)
Course description	Text(1000)
<u>User ID</u>	Number(10)

Student	
<u>Student Id</u>	Number(10)
Name	Text(30)
Email	Text(30)
Password	Text(30)
Contact	Number(11)

Student Allocation	
<u>Allocation Id</u>	Number(10)
Student Id	Number(10)
<u>Course Id</u>	Number(10)

Marks Distribution	
<u>Distribution Id</u>	Number(5)
Distribution criteria	Text(20)
Distribution weight	Number(3)
<u>Course code</u>	Text(7)

Marks	
<u>Marks ID</u>	Number(5)
Mark	Number(6)
<u>Course Code</u>	Text(7)
<u>Student Id</u>	Number(10)
<u>Distribution ID</u>	Number(5)

Question	
<u>Question Id</u>	Number(10)
Question body	Text(100)
<u>User Id</u>	Number(10)

Answer	
<u>Answer Id</u>	Number(10)
Answer body	Text(1000)
Question Id	Number(10)
<u>User Id</u>	Number(10)

# Chapter 6: Class Based Model

This Chapter is intended to describe class based modeling of Course management system.

## 6.1 Class Based Modeling Concept

Class-based modeling represents the objects that the system will manipulate, the operations that will applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

### 6.1.1 Identifying Analysis Class

To identify our analysis class, we firstly grammatically parsed all the nouns and then categorized them according to general classification and selection criteria.

Following are the steps we used to analysis the classes for our system.

**Step-1:** Grammatical parsing (noun identifying) and categorizing using general classification:

**Step-2:** Selection Criteria:

1. Retained information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Identifying potential classes based on their general classification and select as classes by applying selection criteria upon potential classes.

## Class Acceptance

Potential class	General classification	Selection criteria	Accepted/Rejected
Student stuff	Role	Multiple attributes Essential requirements	Accepted
Student	Role	Multiple attributes Common attributes Common operations Essential requirements	Accepted
Email	Things	Retained information	Rejected
Password	Things	Retained information	Rejected
System admin	Role	Multiple attributes Needed services Essential requirements	Rejected
Teacher	Role	Multiple attributes Common attributes Common operations Essential requirements	Accepted
Head of course committee	Role	Multiple attributes Essential requirements	Rejected
Course	Things, event	Multiple attributes Common attributes Common operations	Accepted
Course code	Things	Retained information	Rejected

Course name	Things	Retained information	Rejected
Course description	Things	Retained information	Rejected
Mark distribution	Event	Multiple attributes Needed services	Accepted
Criteria	Things	Retained information	Rejected
Weight	Things	Retained information	Rejected
Time of marks entry	Things	Retained information	Rejected
Start date	Things	Retained information	Rejected
End date	Things	Retained information	Rejected
Mark	Things	Retained information	Rejected
Time limit	Things	Retained information	Rejected
Question	Things	Retained information	Rejected
Answer	Things	Retained information	Rejected
Forum	Organizational unit, Structures	Needed services	Rejected

## Accepted Classes

- Student\_staff
- Student
- System\_admin
- Teacher
- Head\_of\_course\_committee



## Attribute Identification

Class name	Attributes
Student_stuff	Name Email Password
Student	Name Email Password
System_admin	Name Email Password
Teacher	Name Email Password Course Mark_distribution
Head_of_course_committee	Name Email Password

## Verb Identification

Verb	Remark
enlist students	No
add student	Yes
have authority	No
opens courses	No
add course	Yes
allocate teacher	Yes
change teacher	Yes
remove teacher	Yes
assign students	Yes
provide marks distribution	Yes
set the time of marks entry	Yes
enter marks	Yes
update marks	Yes
set permission	Yes
view marks	Yes

ask question	Yes
answer	Yes

## Methods

Class name	Methods
Student_stuff	add_student() remove_student()
Student	view_marks() ask_question() answer()
System_admin	add_teacher() remove_teacher()
Teacher	assign_students() enter_marks() update_marks() set_permission() distribute_marks() ask_question() answer()
Head_of_course_committee	add_course() Allocate_teacher() Change_teacher() Remove_course() Provide_mark_distribution() Set_marks_entry_time()

## Class cards

Student_stuff	
Attribute	Method
Name Email Password	add_student () remove_student ()
Responsibility	Collaborative class

1. Adds student 2. Removes student	Student
---------------------------------------	---------

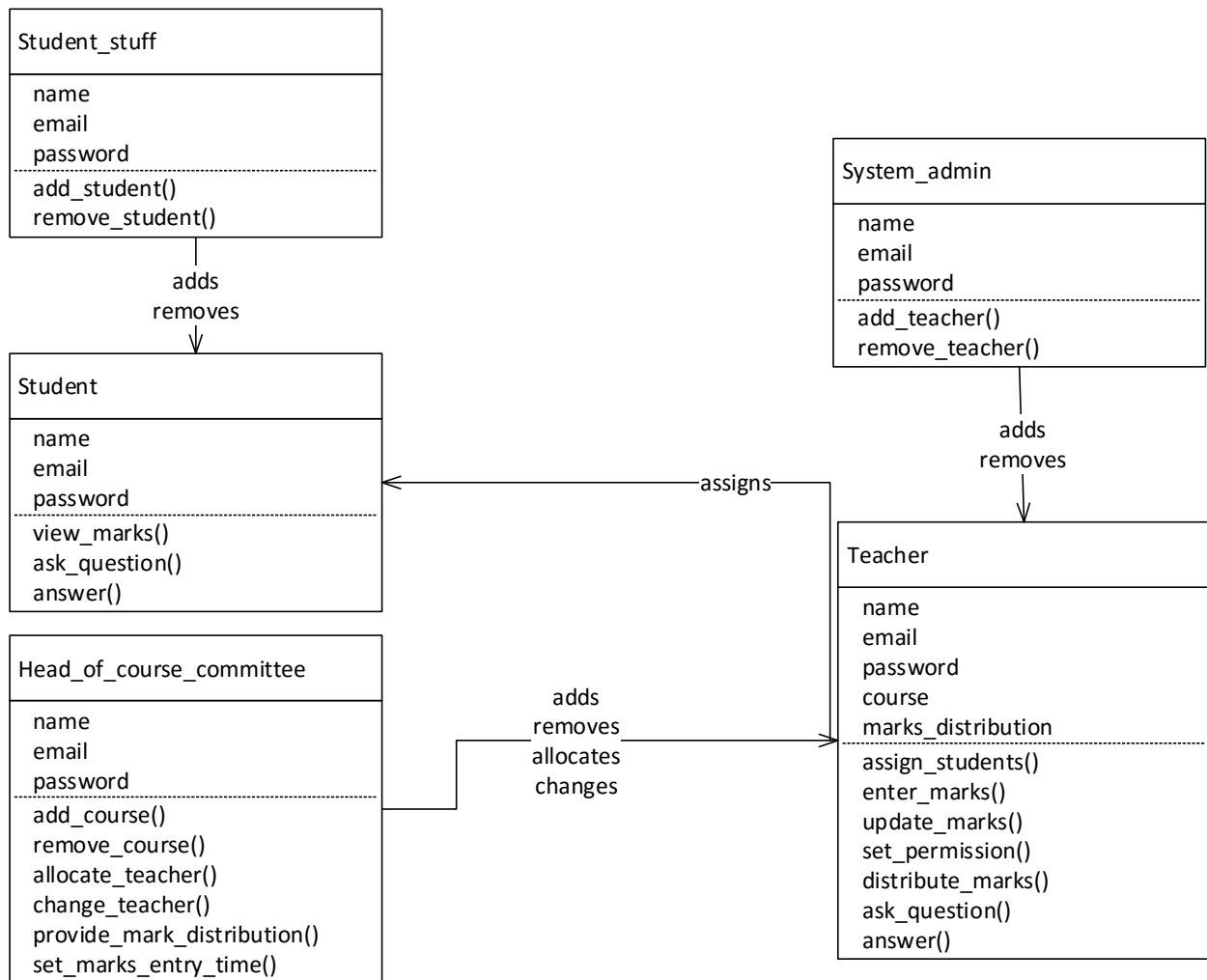
Student	
Attribute	Method
Name Email Password	view_marks() ask_question() answer()
Responsibility	Collaborative class
1.Views marks from sheet 2. Asks question 3. Gives answer	Student_stuff Teacher

System_admin	
Attribute	Method
Name Email Password	add_teacher () remove_teacher()
Responsibility	Collaborative class
1. Adds teacher 2. Removes teacher	Teacher

Teacher	
Attribute	Method
Name Email Password Course Mark_distribution	assign_students() enter_marks() update_marks() set_permission() distribute_marks() ask_question() answer()
Responsibility	Collaborative class
1. Enters marks 2. Assigns students 3. Updates marks 4. Distributes marks 5. Asks Question 6. Assigns students	Head_of_course_committee System_admin Student

Head_of_course_committee	
Attribute	Method
Name Email Password	add_course () Allocate_teacher () Change_teacher () Remove_course () Provide_mark_distribution() Set_marks_entry_time()
Responsibility	Collaborative class
1. Add course 2. Change teachers 3. Allocate teacher 4. Provide marks distribution 5. Add marks 6. Remove marks 7. Set marks entry time	Teacher

## Class Responsibility Collaboration (CRC)



**Figure: CRC diagram of Course Management System**

# Chapter 7: Flow Oriented Model

This chapter focuses on the flow oriented modeling.

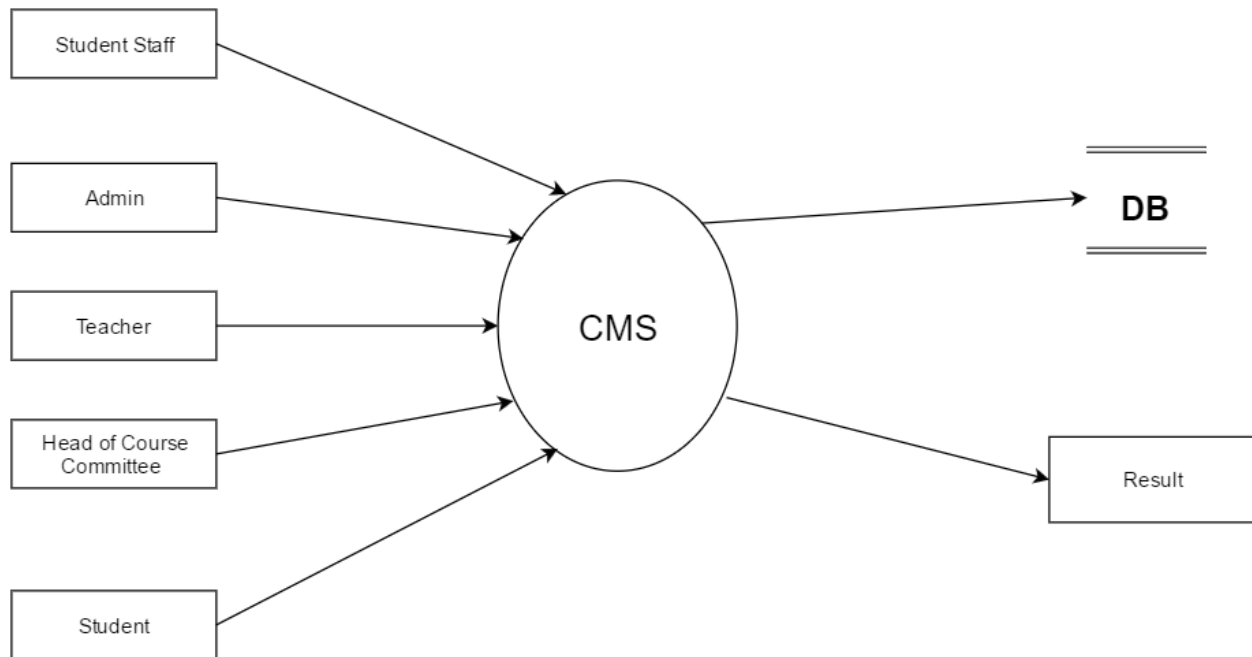
## Introduction

Although data flow-oriented modeling is perceived as an outdated technique by some software engineers, it continues to be one of the most widely used requirements analysis notations in use today. It provides additional insight into system requirements and flow.

### 1.1 Data Flow Diagram

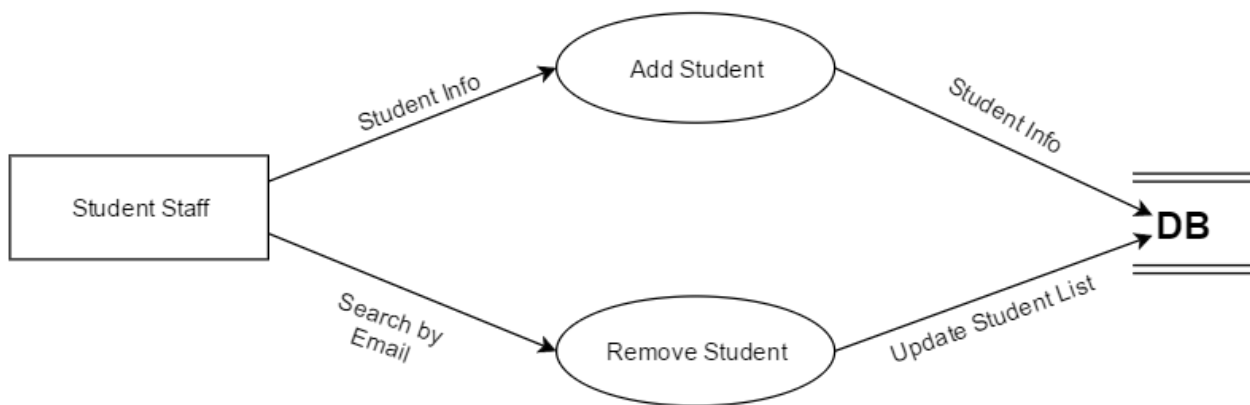
The DFD takes an input-process-output view of a system. In the figures, data objects are represented by labeled arrows and transformations are represented by circles.

**Level-0 DFD:** Level-0 DFD describing the overall system's input and output data.



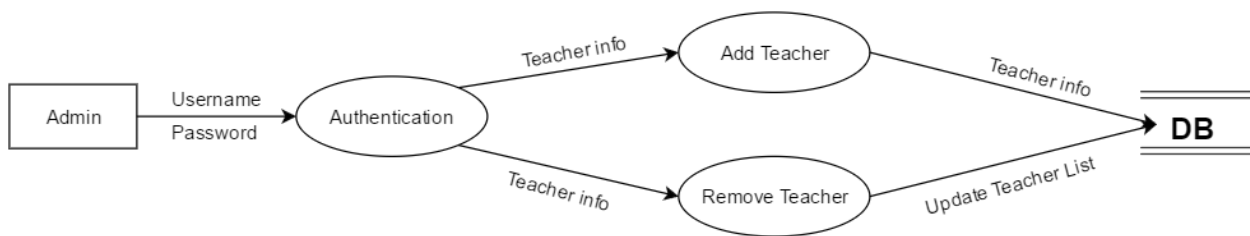
**Figure 27: Level-0 DFD**

**Level-1.1 DFD:** This level-1.1 DFD derived from level-0 DFD.



**Figure 28: Level-1.1 DFD for Student**

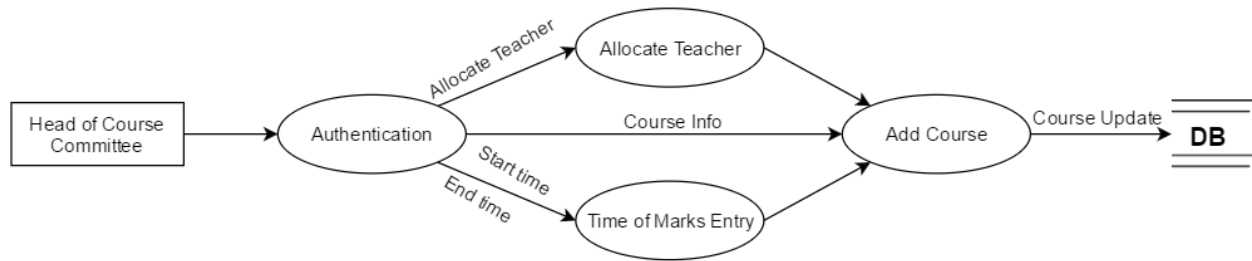
**Level-1.2 DFD:** This level-1.2 DFD derived from level-0 DFD



**Figure 29: Level-1.2 DFD for Admin**

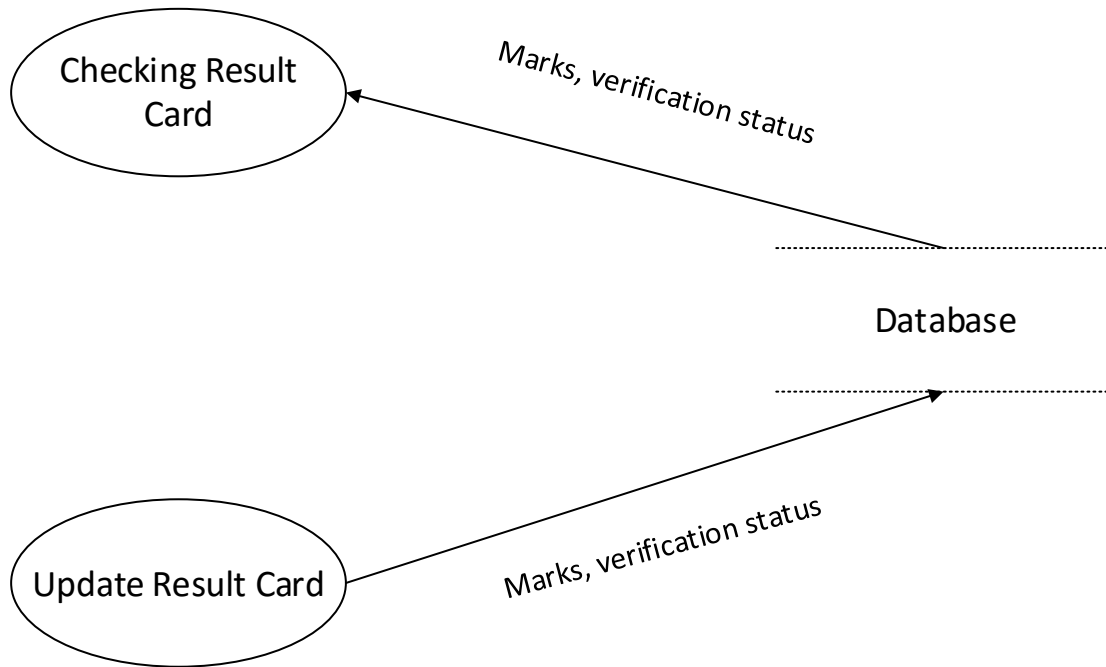


**Level-1.3 DFD:** This level-1.3 DFD derived from level-0 DFD



**Figure 30: Level-1.3 DFD for Head of course committee**

**Level-2.3 DFD:** The process Result Card Verification can be divided into two sub process which derived into level-2.3 DFD.



**Figure 31: Level-2.3 DFD for Result Card Verification**

# Chapter 8: Behavioral Model

The behavioral model indicates how software will respond to external events.

## 8.1 State Transition Diagram

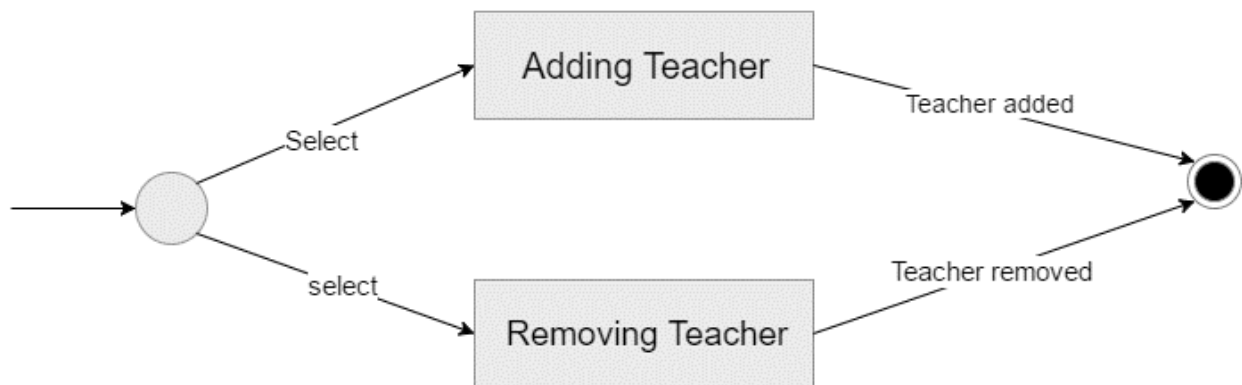
State diagram represents active states for each class the events (triggers). For this we identified all the events, their initiators and collaborators.

Count	Event	Initiator	Collaborator
1	Adding Courses	Head_of_Course_Committee	-
2	Allocating teachers	Head_of_Course_Committee	Teacher
3	Changing teachers	Head_of_Course_Committee	Teacher
4	Removing course	Head_of_Course_Committee	-
5	Providing marks distribution	Head_of_Course_Committee	-
6	Setting marks entry time	Head_of_Course_Committee	-
7	Assign students	Teacher	Student
8	Entering marks	Teacher	-
9	Updating marks	Teacher	-
10	Setting permission to view result	Teacher	-
11	Distributing marks	Teacher	-
12	Asking question	Teacher	Student
13	Answering question	Teacher	Student
14	Adding teacher	System Admin	Teacher

15	Removing teacher	System Admin	Teacher
16	Viewing marks	Student	Teacher
17	Asking Question	Student	Teacher
18	Answering question	Student	Teacher
19	Adding students	Student_staff	Student
20	Removing students	Student_staff	Student

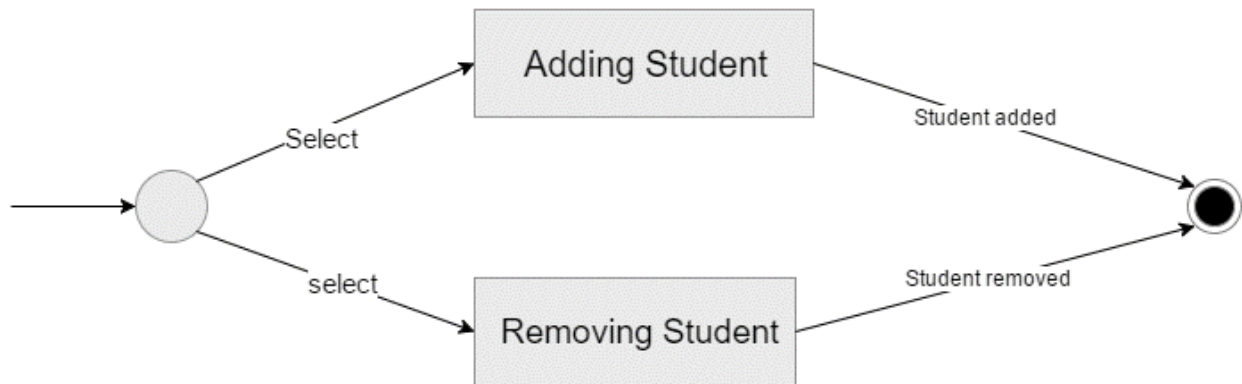
State Transition Diagram represents active states for each class and the events (triggers) that cause changes between these active states. Here we have provided diagram for each of the actors.

### System admin



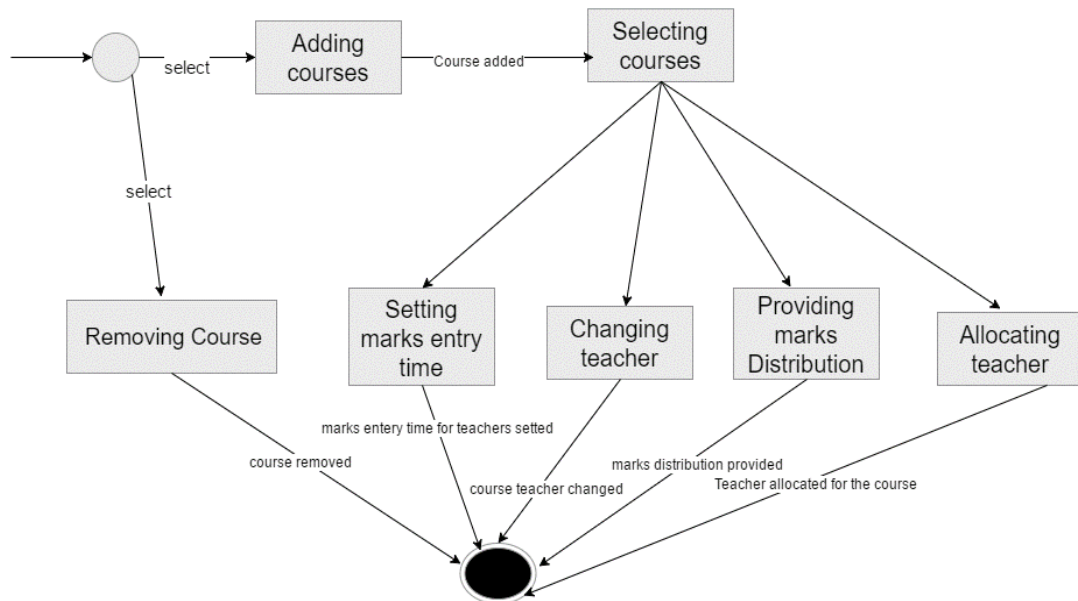
**Figure 32: State Transition Diagram for System admin**

## Student staff



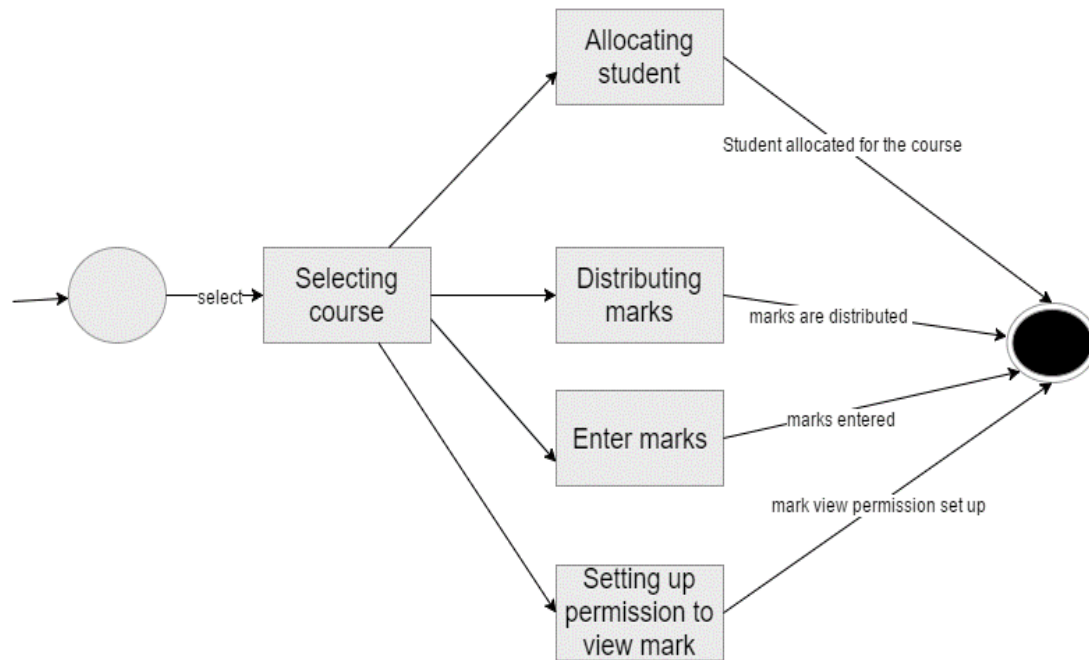
**Figure 33: State Transition Diagram for Student staff**

## Head of course committee



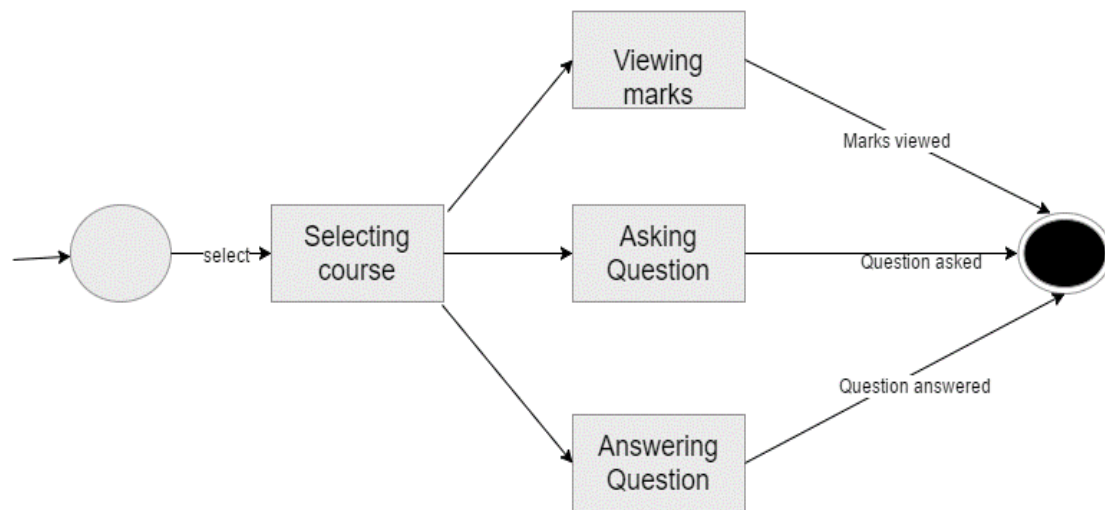
**Figure 34: State Transition Diagram for Head of course committee**

## Teacher



**Figure 35: State Transition Diagram for Teacher**

## Student



**Figure 36: State Transition Diagram for Student**

## 8.2 Sequence Diagram

