Software Requirement Specification and Analysis
SPL-2 Mid presentation 1

E-shcool: An Online Learning Environment

Submitted By

Hasnain Iqbal, BSSE-1106

Ahmed Adnan, BSSE-1131

Supervised By

Dr. Zerina Begum
Professor
Institute of Information Technology
University of Dhaka

Date: 10-3-2022

Introduction

This chapter is a part of our Software Requirement Specification and Analysis for the project “E-shcool: An online Learning Environment”. In this chapter, we have tried to identify the purpose and the intended audiences for our project.

Purpose

E-shcool is an online-based educational platform which will provide a learning environment to the students and ensure the academic experience virtually to both students and teachers. During this pandemic situation, offline educational activities have become tough and unavailable. So, we are trying to solve this problem by developing an effective virtual educational environment. The purpose of this project is to make the major academic events online and automate some manual activities. This project aims to ensure distance learning as well as virtual management of the educational system of IIT(Institute of Information Technology).

Intended Audience

This document is intended for software developers, project managers, marketing staff, users, testers, and documentation writers. The rest of the document contains SRSA (software requirement specification and analysis) starting from product features, all the way to diagrams and plan of implementation. This document is best read serially from top to bottom. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. But as time goes on, our SRS document is expected to evolve as users and developers work together to validate, clarify and expand its contents. The SRS serves as an official medium of expressing users’ requirements to the developer and provides a common reference point for both the developer team and the stakeholder community.

Conclusion

This analysis of the audience helped us to focus on the users who will be using our software requirements analysis. This overall document will help each and every person related to this project that includes users, project managers, designers, developers, testers, stakeholders to have a better idea about the project.

Inception of the Project

Introduction

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved is. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we established a basic understanding of the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team. To establish the groundwork, the following factors have been worked on to the inception phase:

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

Icebreaking

Icebreaking refers to the fact that to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptance of our proposal. We started this by talking with them in context-free languages. Their behavior, responding to our questions or willing to take a change in their shops solely depends on this phase.

Identifying the Stakeholders

Stakeholder refers to any person or group who will be affected directly or indirectly by the system. Stakeholders include end-users who interact with the system and everyone else in an organization who may be affected by its installation. The situations that we have observed indicate that “E-shcool: An Online Learning Environment” has a few stakeholders only. Identification of the stakeholders were done from the information provided by the potential users from IIT (Institute of Information Technology), University of Dhaka. The stakeholders of our system are given below:

- Students
- Teachers
- Office Staffs (who will act as admin)

Identifying Multiple Viewpoints of the Stakeholders

Different stakeholders expect different benefits from the system as every person has his own point of view. So, we have to recognize the requirements from multiple viewpoints. Different viewpoints of the stakeholders about the expected software are given below:

- **Teacher's View Point**

    ➔ Conduct online classes easily.
    ➔ Handle course activities in an efficient manner.
    ➔ Provide course materials in an organized way.
    ➔ A platform to communicate with the students about academics.
    ➔ Take exams, assignments online and evaluate them.

- **Student's View Point**
    ➔ Get notified about online class meetings and easy access to the meetings.
    ➔ Have all the course materials in an organized space..
    ➔ A place to discuss and inquire about course activities.
    ➔ A platform to make the exams and assignments related activities easier.
    ➔ A place to get all the academic notices.

- **Office Staffs (Admin)**
    ➔ Handle all academic data in a systematic way.
    ➔ Have the entire academic system online.

Conclusion

Our primary goal is to make a software that will automate all the activities that happen in a classroom. We expect to create an online learning environment that will satisfy all the requirements and expectations of all the stakeholders involved. For this reason, this software will have a user-friendly interface and the design will be very interactive to give the same experience they get in a physical classroom environment.

Elicitation of E-shcool

Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders. Many difficulties were faced, like understanding the problems, making questions for the stakeholders, limited communication with stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organized and systematic manner.

Eliciting Requirements

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. The following tasks were done for eliciting requirements:

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

Collaborative Requirements Gathering

We have met with some stakeholders in the Inception phase such as the teachers, students, staff. These meetings created an indecisive state for us to elicit the requirements. To solve this problem, we have met with the stakeholders (who are playing a vital role in the whole process) again to elicit the requirements.

Quality Function Deployment (QFD)

Quality function deployment (a model for product development and production) is a focused methodology for carefully listening to the customers or consumers of that particular product and then effectively responding to those identified needs and expectations.

We visited IIT (Institute of Information Technology), University of Dhaka. They expressed their opinions and expectations about an online learning environment/ classroom management system. Their reactions and suggestions have given us a clear view of the product's expected requirements and a possible solution.

Normal requirements

In normal requirements, the motive and perspective or the aim of the project are defined by collecting the statements of the customers. These statements' collection upholds the basic system procedure that is to be followed while turning the system product in an automated manner.

Normal requirements of E-shcool:

- Course enrollment should be automated for the students.
- The class routine should be available for both teachers and students.
- The class should be taken in a secure third-party application.
- The attendance of the students should be automatically calculated.
- There should be a notice board.
- Exams and assignments related activities are needed to be there.
- Automatic result management.
- Class materials are needed to be organized for students to use.

Expected requirements in QFD

In an existing system or service, the main demand of the customer beholds the user friendly objects and system approach. As we recently visited IIT, University of Dhaka, we came across the expectations of the stakeholders classified into three different categories according to their position. They are the teachers, students and office staff.

The expected requirement of E-shcool:

- A course stream to communicate easily.
- Reminders of deadlines.
- Well-managed course material section.
- A search bar.
- Exam management system.
- Result viewing for teachers.

Exciting requirements in QFD

These features go beyond expectations and prove to be satisfying when present. Often exciting requirements enable the scope of innovation in the process or new ways of handling functionalities. Stakeholder satisfaction can be dramatically improved through the implementation of a few exciting requirements. For our project, we have identified some exciting requirements.

Exciting requirements of E-shcool:

- An option to auto-evaluate the written answer scripts.
- A chatbot for ease of inquiries.
- Each student's performance analysis.

Usage Scenario

Abstract

E-shcool is an online-based educational platform which will provide a **learning environment** to the students and ensure the **academic experience virtually** to both students and teachers. This project will provide everything needed to operate and manage a class. This **Web-based** project will manage and automate the academic activities in such a way that all the entities involved (teachers, students, admin) will be provided with the accessibility of the features required in an efficient manner.

The project will have the following features and will work in the following chronological order of the features:

Authentication

The system has an Authentication System. A user can-

- **Log In**
- **Sign Up**

in the authentication phase. There will be 3 types of entities in the authentication phase:

- **Admin**
- **Student**
- **Teacher**

Admin's Authentication: Admins' account and profile will exist from the very beginning so they **don't need to sign up**. They will only log in into their accounts to handle management.

Students' Authentication: Students can open their accounts by providing their name, roll, semester, reg no etc. information and they will have a personal profile in which they can login. The Admin will **verify** the students when they sign up and approve their accounts.

Teachers' Authentication: The **admin will create profiles for the teachers.** The teachers can then log in to their personal profile accounts by providing login information.

Admin's Usage Scenario

The admin will have the following functionalities:

Authenticity Verification: The admin will verify the student accounts and will create the teachers' accounts authenticating their information manually.

Handling Academic Activities: The admin will do the **allotment of the courses** semester wise by following the academic routine. The students will then **automatically be enrolled** into the courses of their respective semesters in their personal profiles. Admin will also **allot courses to the teachers** following the academic routine and the teachers can view their assigned courses in their personal profiles.

While assigning courses into semesters and teachers into those courses, a routine will be automatically generated and the required parts of that routine will be sent to the profiles of the teachers and the students based on their course enrollments.

Handling the Notice Board: The admin has the **full control over the notice board** and will **update the latest academic notices** and information in the notice board which can be seen from the profiles of the students and the teachers.

Database Manipulation: The admin has **full authority over the database** and can add, update and delete user data in it.

Teachers' Usage Scenario

A teacher can login to his/her profile. The teacher will have the following functionalities in the profile:

Assigned Course Tiles: The teacher can see only the courses he/she has been assigned to. Entering the courses, there will be a **course stream** in which both the teachers and the students can post and comment things about the course. A course stream will also contain **ongoing and scheduled meeting links**. There will also be a section for **course materials management** and another section for the **exams and assignments related activities**. The teacher can also see all the **performance, results and marks** of the students of that course in a separate section and give feedback.

Scheduling Meetings: A teacher can **schedule meetings** for the courses by using this module. An online meeting link will be automatically generated and all the students enrolled into that course will be **automatically notified** into their personal profiles and emails.

Materials Management: There will be a materials management section in the profile of the teacher. By using this module, a teacher can **upload and remove class recordings, lecture slides, PDFs** etc. All the uploaded materials will **automatically be organized into folders** according to their **types**. This materials management section can also be entered through the course streams to ensure specificity.

Exams and Assignments Related Activities: There will be a section where a teacher can handle assignments and exam related activities. There will be two sections in this segment:

- **Uploading Assignments and Exams**
- **Evaluation of Exams and Assignments**

Uploading Assignments and Exams: A teacher can **upload assignments** along with a deadline and post things about that assignment. A teacher can also take exams. There will be two ways to take an exam:

- **MCQ based exam:** The teacher will have to create a **Google Form** and upload that form in this section

- **Written exam:** The teacher will **upload question paper**

Evaluation of Exams and assignments: The evaluation of the assignments will be completely manual. But in case of exams, the marks of the MCQ based exams will automatically be added to the students profile. In case of the written exams, there will be two ways-

- **Manual Checking:** A teacher can **manually check** the exam scripts uploaded by the students and add marks.

- **Auto Evaluation by the machine:** The teacher will have to **upload a sample solution.** The machine will first convert the uploaded **handwritten answer scripts** of the students into **typed softcopies using Machine Learning**, then **evaluate** them using the sample solution using **NLP algorithms.** Then the evaluated marks will be added to the student's profile.

Reminder Section: This section will show the teacher all the **upcoming** classes, exams and assignments.

An AI based Chatbot: There will be an **AI based chatbot** that will **answer** some academic questions and will provide some **suggestions** to the teacher.

Search Bar: A search bar will be there for the inquiries.

Notice Board: A teacher can see all the latest academic notices and updates in the notice board.

Students' Usage Scenario

Assigned Course Tiles: A student can only see the courses of the semester he/she is in. Entering the courses, there will be a **course stream** in which both the teachers and the students can post and comment things about the course. A course stream will also contain **ongoing and scheduled online class meeting links**. There will be a section where a student can access the **course materials.** There will be another section for the **exams and assignments related activities.**

Join Ongoing Meetings: There will be a module where a student can see the **ongoing online class meetings** and can join the meeting by simply clicking on that module. The **attendance** will be **automatically counted** from the online meeting and will be added to the profile of that student for that respective course.

Course Materials Section: There will be a section for course materials in the profile of the student. By using this module, a student can **access** and **download** the class recordings, lecture slides, PDFs etc.

Exams and Assignments Related Section: A student can see all the **upcoming exams and assignments** in this section along with the submission deadline. A student can **upload the handwritten solution PDFs** in this section which will be evaluated later. In case of the MCQ based exams, the student can fill up the google forms from here.

Performance and Result Section: A student can see the **marks** of the quizzes, assignments and exams here and there will also be **performance analysis graphs** for the student's in depth understanding.

An AI based Chatbot: There will be an AI based chatbot that will **answer some academic questions** and will provide some **suggestions** to the student.

Search Bar: A search bar will be there for the inquiries.

Notice Board: A student can see all the latest academic notices and updates in the notice board.

Use Case Diagram : E-shcool

What is Use Case Diagram

A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

The Purpose of Use Case Diagram

The reasons why an organization would want to use case diagrams include:

- Represent the goals of systems and users.
- Specify the context a system should be viewed in.
- Specify system requirements.
- Provide a model for the flow of events when it comes to user interactions.
- Provide an outside view of a system.
- Show’s external and internal influences on a system.

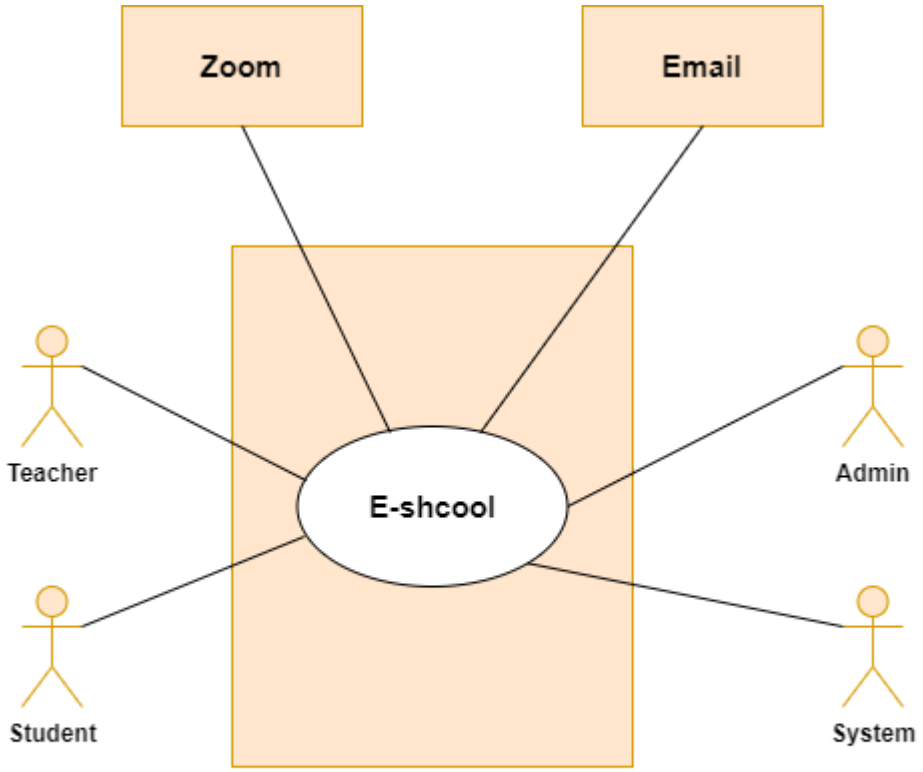
Level - 0

Name: E-shcool: An Online Learning Environment

Primary Actors: Teacher, Student

Secondary Actors: Admin, System

External Entities: Zoom, Email, Google Form



Level 0 - E-shcool: An Online Learning Environment

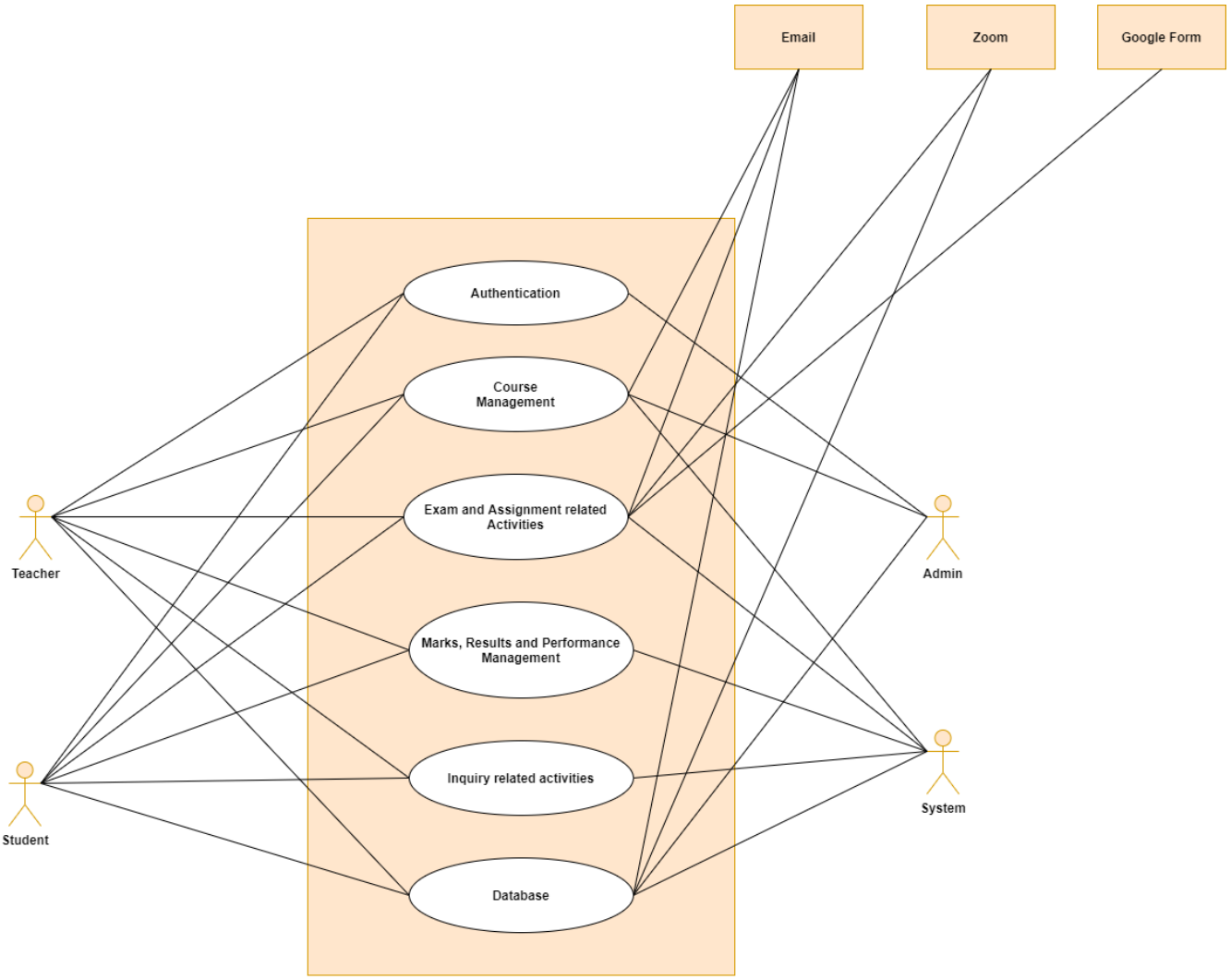
Level - 1

Name : E-shcool: An Online Learning Environment

Primary Actors : Teacher, Student

Secondary Actors : Admin, System

External Entities: Zoom, Email, Google Forms



Level 1 - E-shcool: An Online Learning Environment

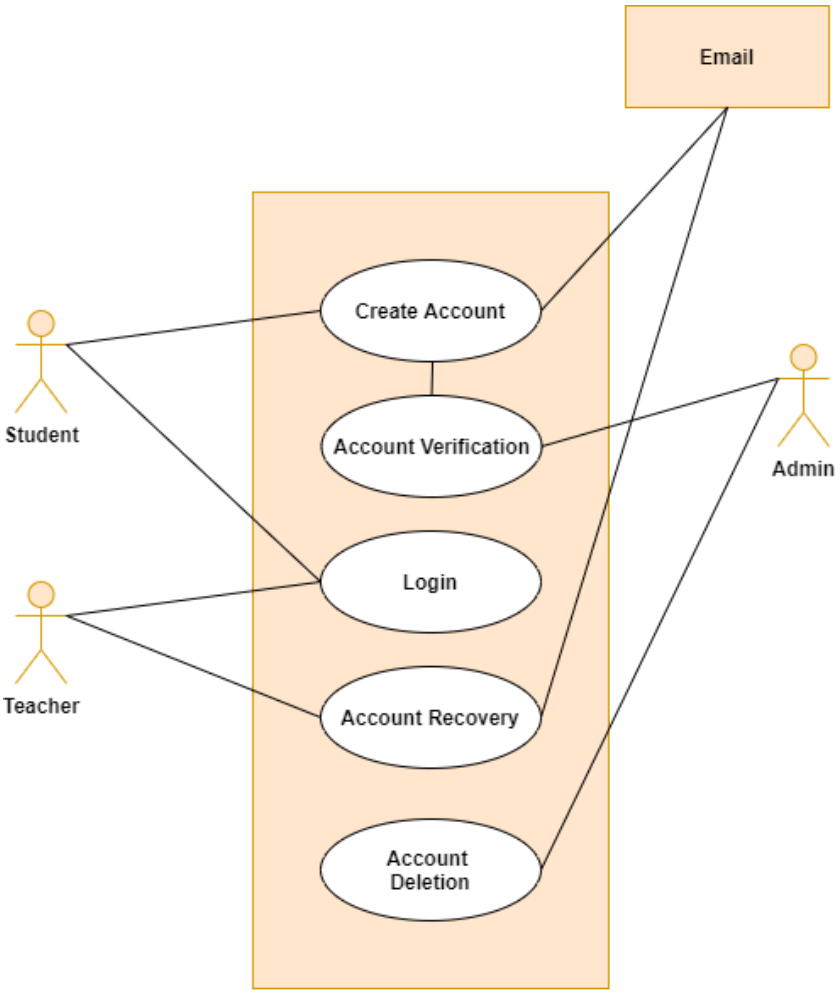
Level - 1.1

Name : Authentication

Primary Actors : Teacher, Student

Secondary Actors : Admin

External Entities: Email



Level 1.1 - Authentication

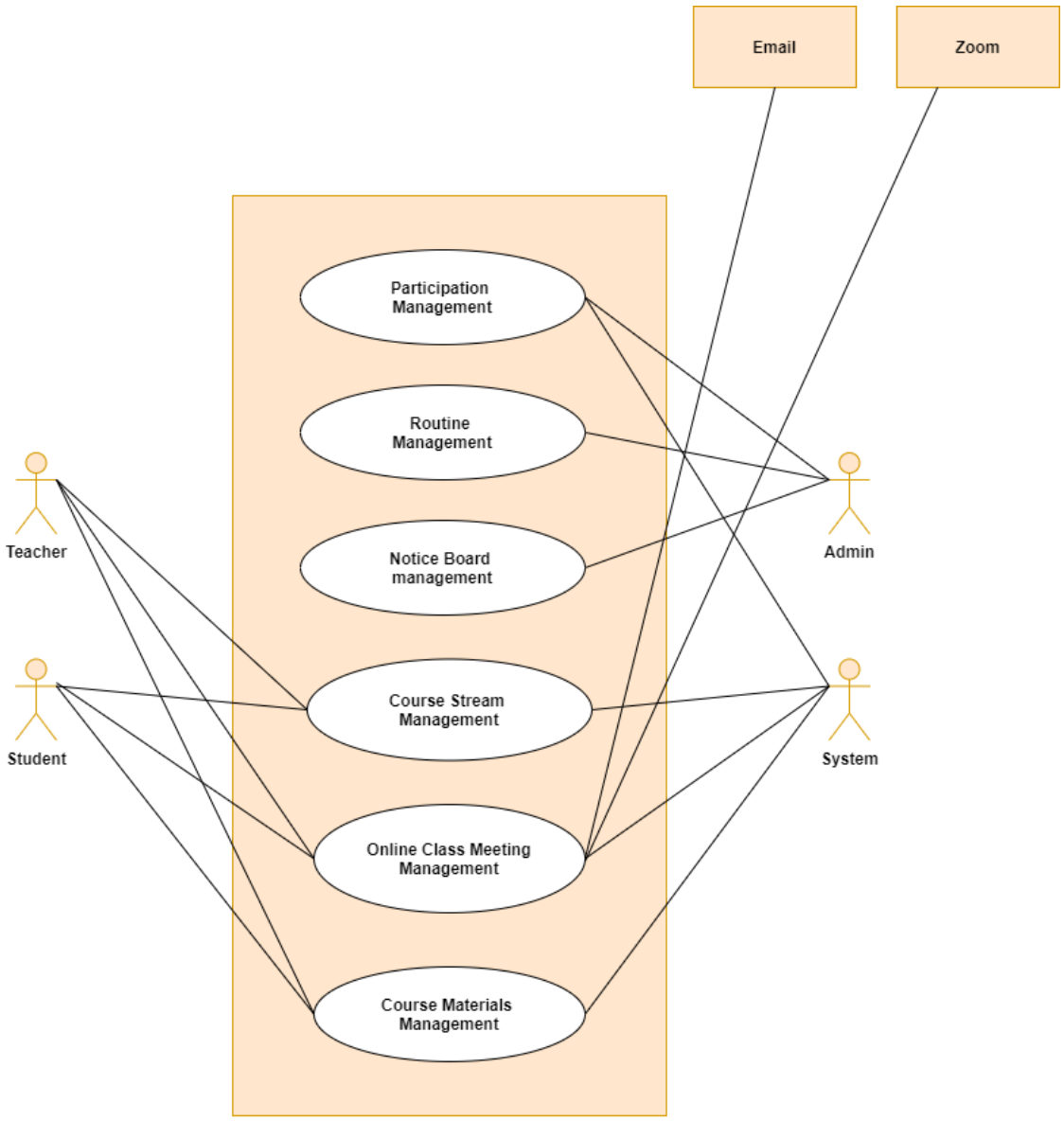
Level - 1.2

Name : Course Management

Primary Actors : Teacher, Student

Secondary Actors : Admin, System

External Entities: Email, Zoom



Level 1.2 - Course Management

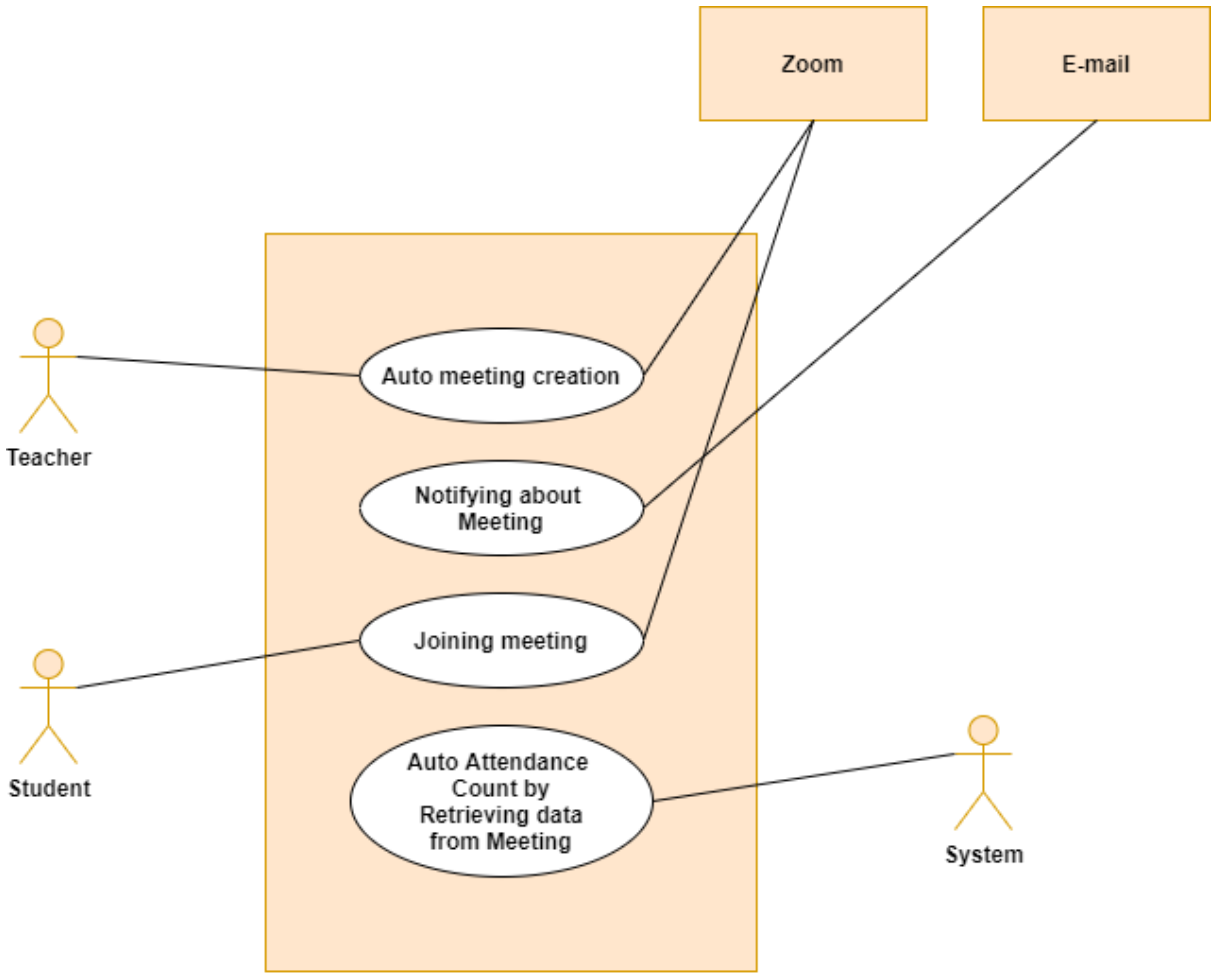
Level - 1.2.5

Name : Online Class Meeting Management

Primary Actors : Teacher, Student

Secondary Actors : System

External Entities: Email, Zoom



Level 1.2.5 - Online Class Meeting Management

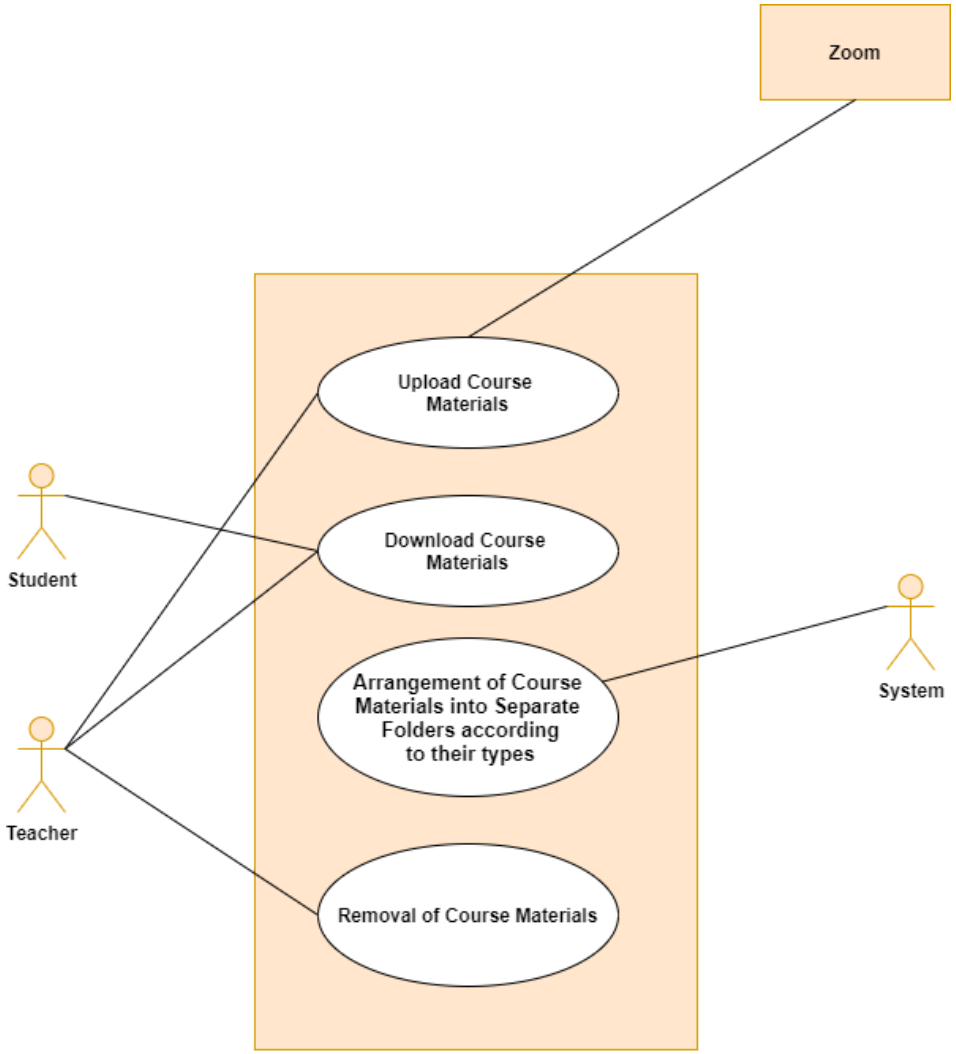
Level - 1.2.6

Name : Course Materials Management

Primary Actors : Teacher, Student

Secondary Actors : System

External Entities: Zoom



Level 1.2.6 - Course Materials Management

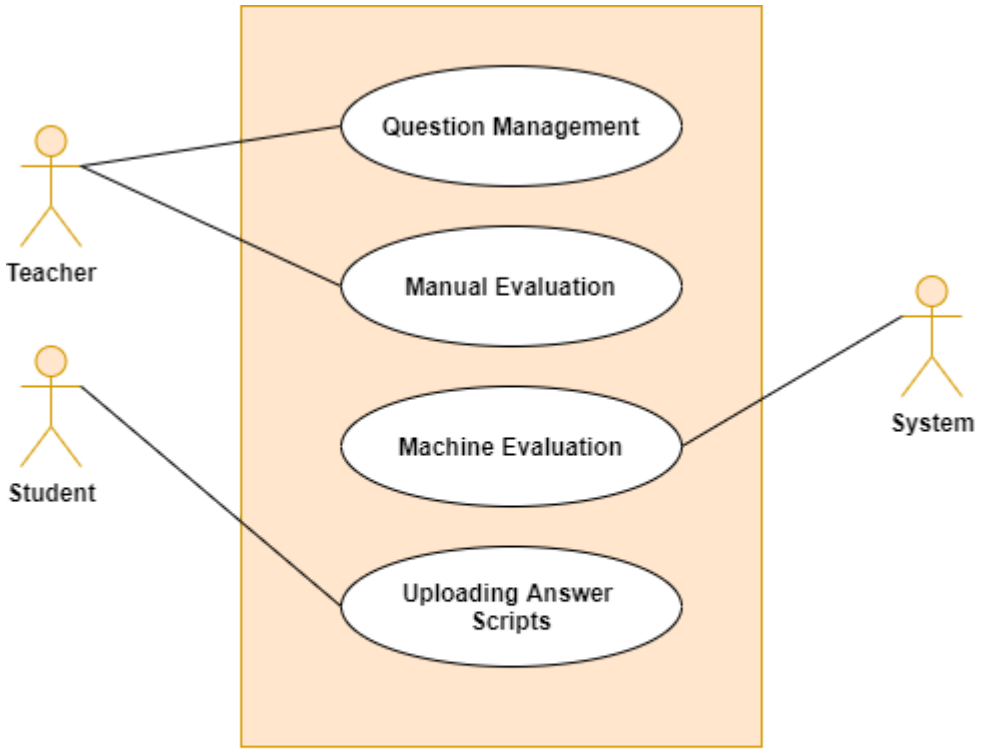
Level - 1.3

Name : Exam & Assignment Related Activities

Primary Actors : Teacher, Student

Secondary Actors : System

External Entities:



Level 1.3 - Exam & Assignment Related Activities

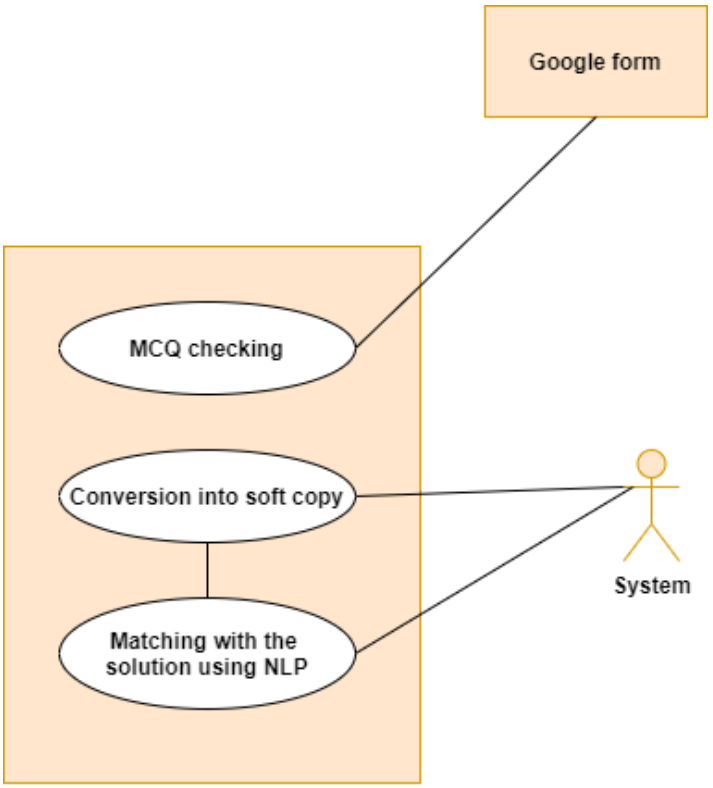
Level - 1.3.3

Name : Machine Evaluation

Primary Actors : N/A

Secondary Actors : System

External Entities: Google Form



Level 1.3.3 - Machine Evaluation

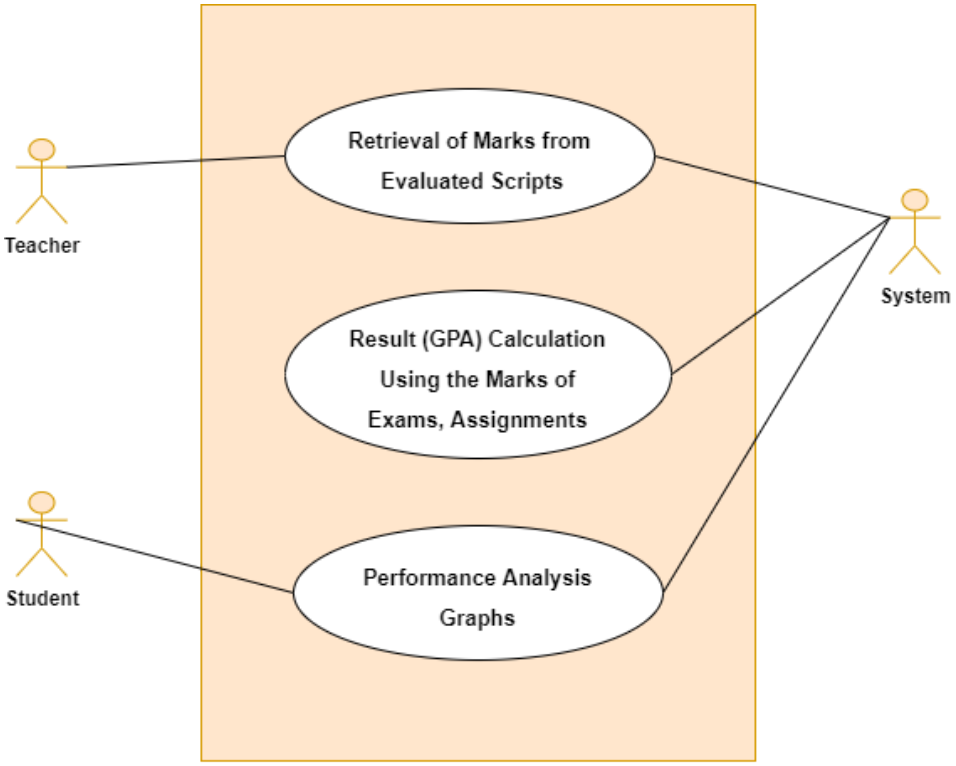
Level - 1.4

Name : Marks, Results & Performance related Activities

Primary Actors : Teacher, Student

Secondary Actors : System

External Entities: N/A



Level 1.4 - Marks, Results and Performance Management

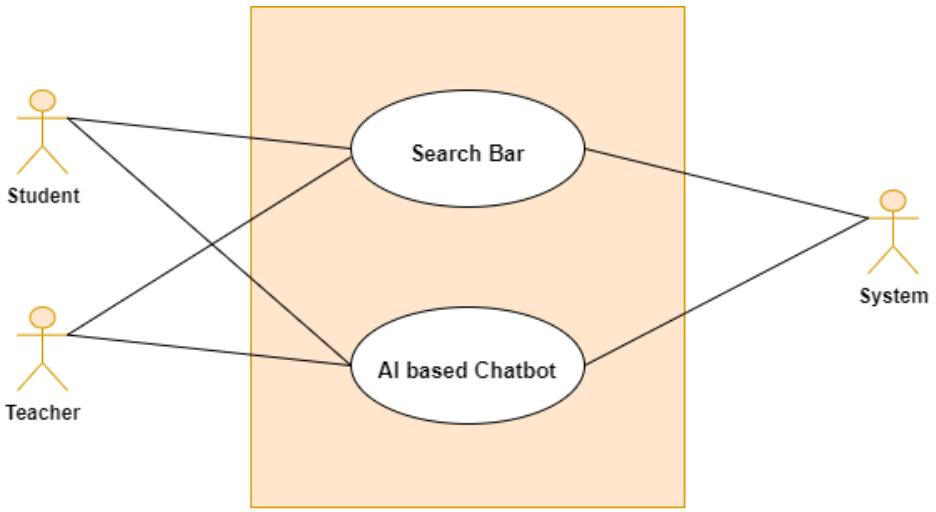
Level - 1.5

Name : Inquiry related Activities

Primary Actors : Teacher, Student

Secondary Actors : System

External Entities: N/A



Level 1.5 - Inquiry Related Activities

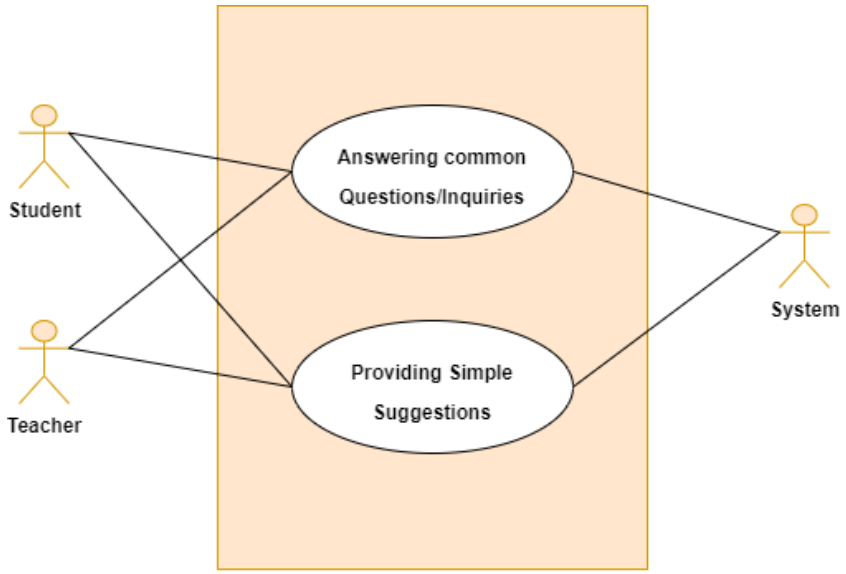
Level - 1.5.2

Name : AI based Chatbot

Primary Actors : Teacher, Student

Secondary Actors : System

External Entities:



Level 1.5.2 - AI based Chatbot

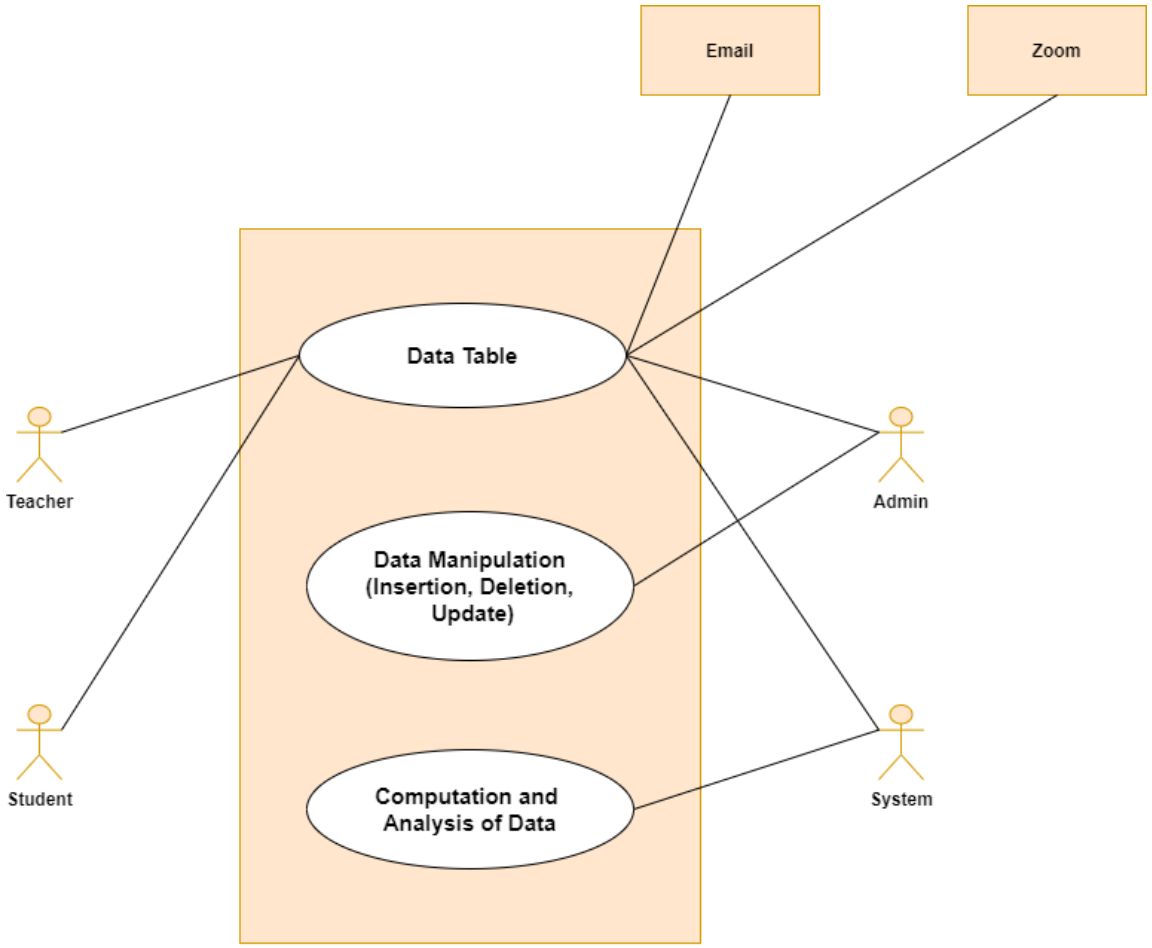
Level - 1.6

Name : Database

Primary Actors : Teacher, Student

Secondary Actors : Admin, System

External Entities: Email, Zoom



Level 1.6 - Database

Activity Diagram : E-Shcool

Activity Diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The basic purpose of activity diagrams is to capture the dynamic behavior of the system.

It focuses on the execution and flow of the behavior of a system instead of implementation. Activity diagrams consist of activities that are made up of actions that apply to behavioral modeling technology.

An activity can be attached to any modeling element to model its behavior. Activity diagrams are used to model-

- Use cases
- Classes
- Interfaces
- Components etc.

Following rules must be followed while developing an activity diagram-

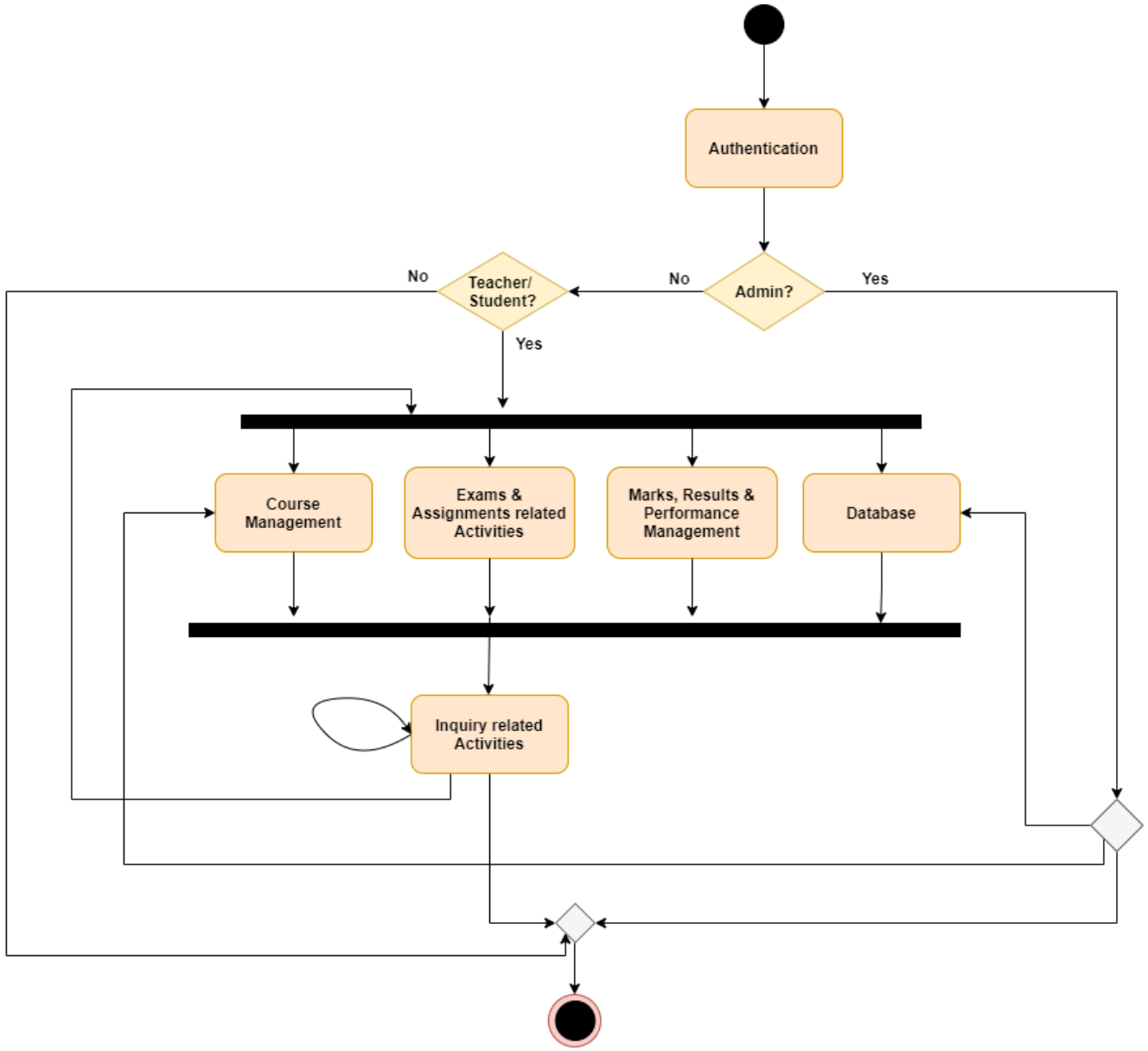
- All activities in the system should be named.
- Activity names should be meaningful.
- Constraints must be identified.
- Activity associations must be known.

To draw an activity diagram, one must understand and explore the entire system. All the elements and entities that are going to be used inside the diagram must be known by the user. The central concept which is nothing but an activity must be clear to the user. After analyzing all activities, these activities should be explored to find various constraints that are applied to activities. If there is such a constraint, then it should be noted before developing an activity diagram.All the activities, conditions, and associations must be known. Once all the necessary things are gathered, then an abstract or a prototype is generated, which is later converted into the actual diagram.

Level-1

Name : E-shcool: An Online Learning Environment

Reference : Use Case Level 1

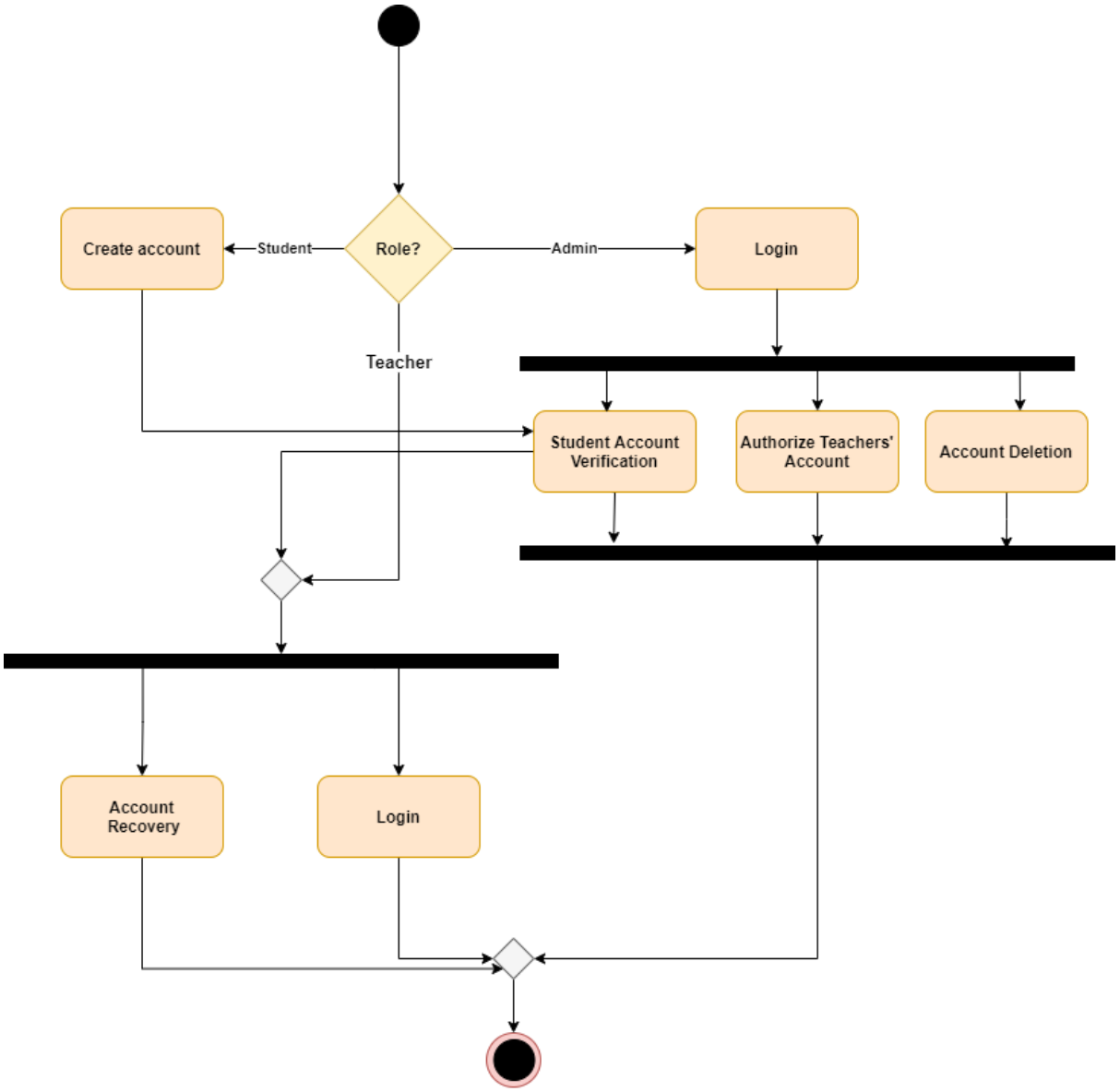


Level 1 - E-shcool: An Online Learning Environment

Level-1.1

Name : Authentication

Reference : Use Case Level 1.1

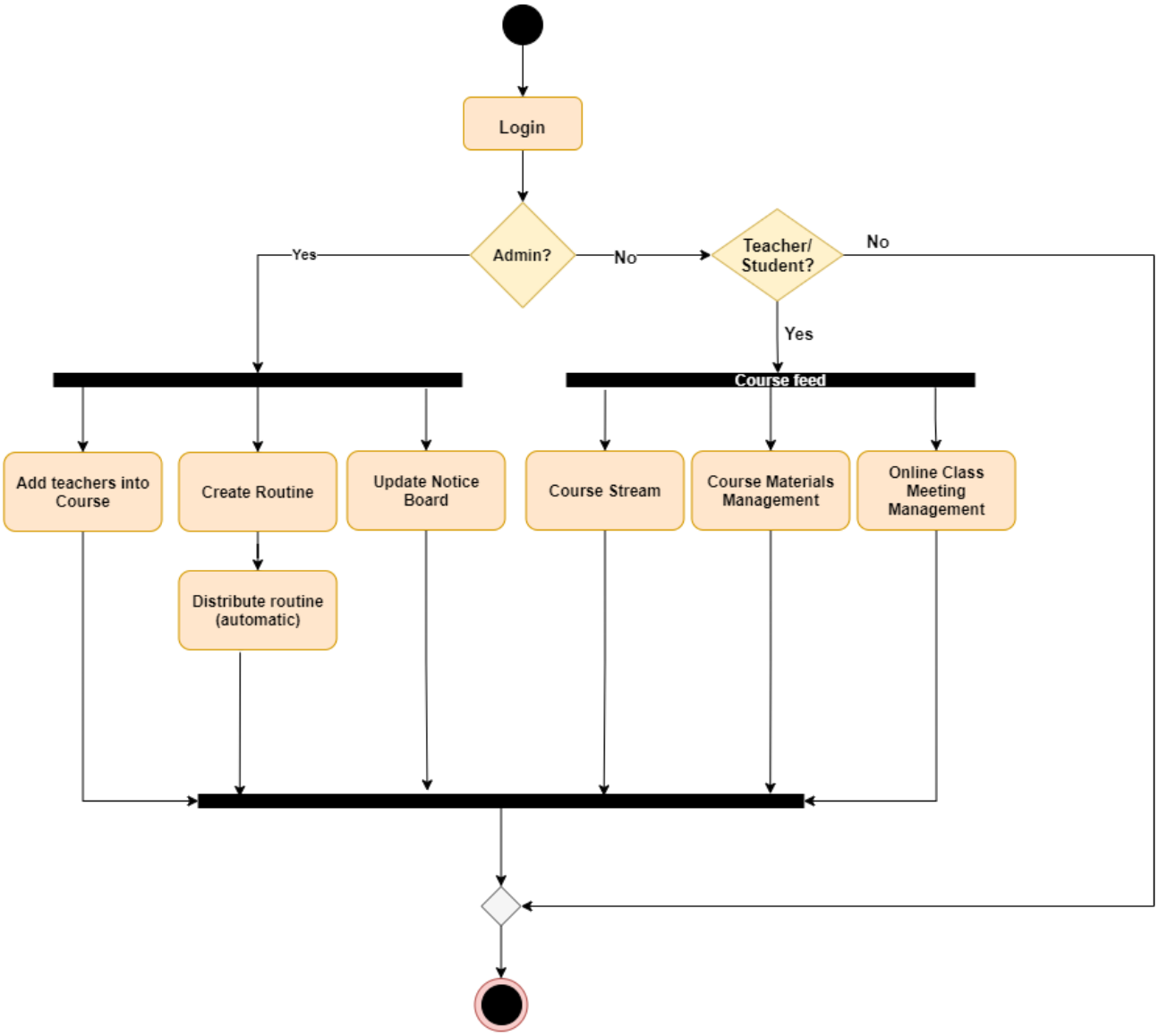


Level 1.1 - Authentication

Level-1.2

Name : Course Management

Reference : Use Case Level 1.2

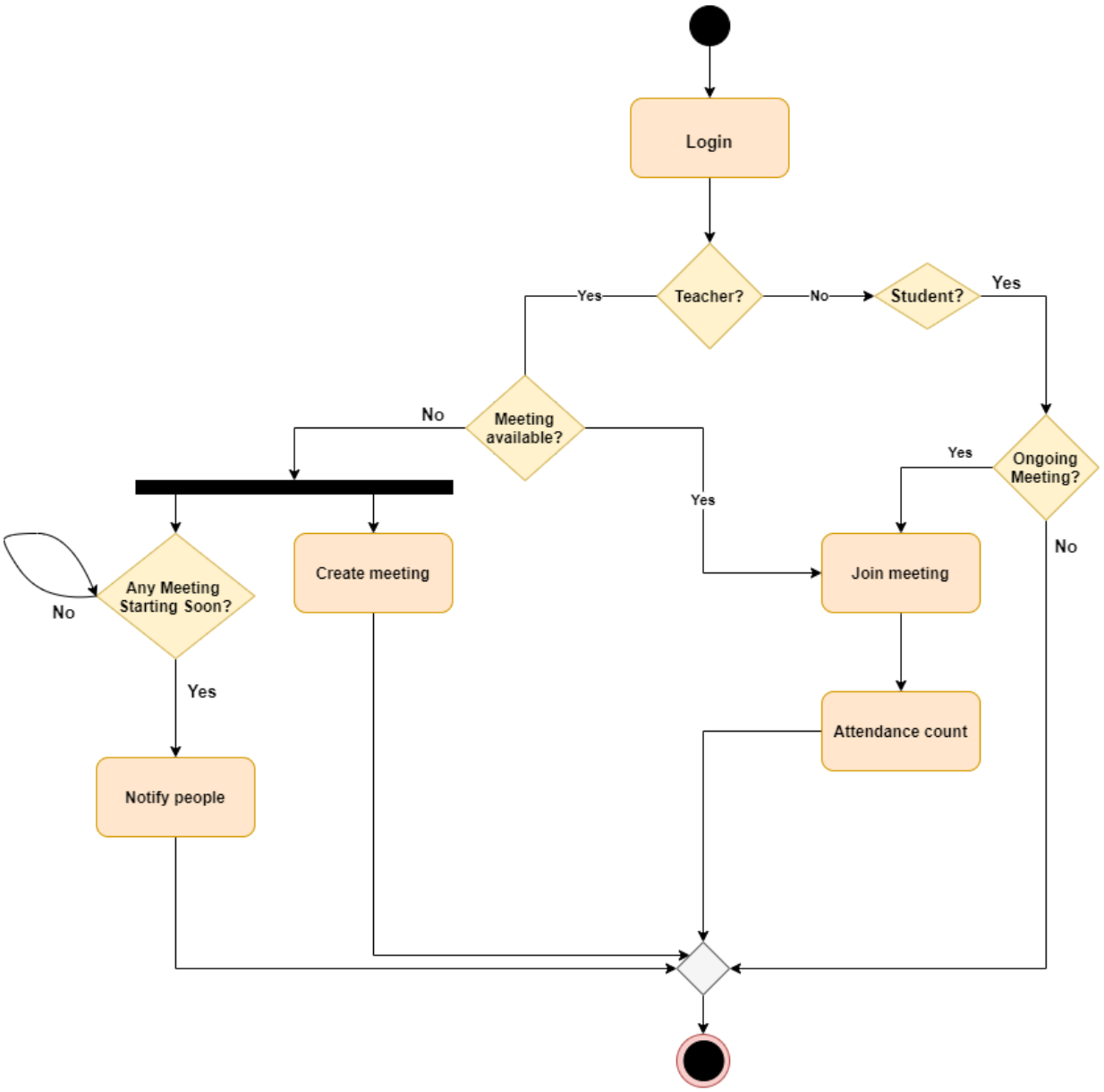


Level 1.2 - Course Management

Level-1.2.5

Name : Online Class Meeting Management

Reference : Use Case Level 1.2

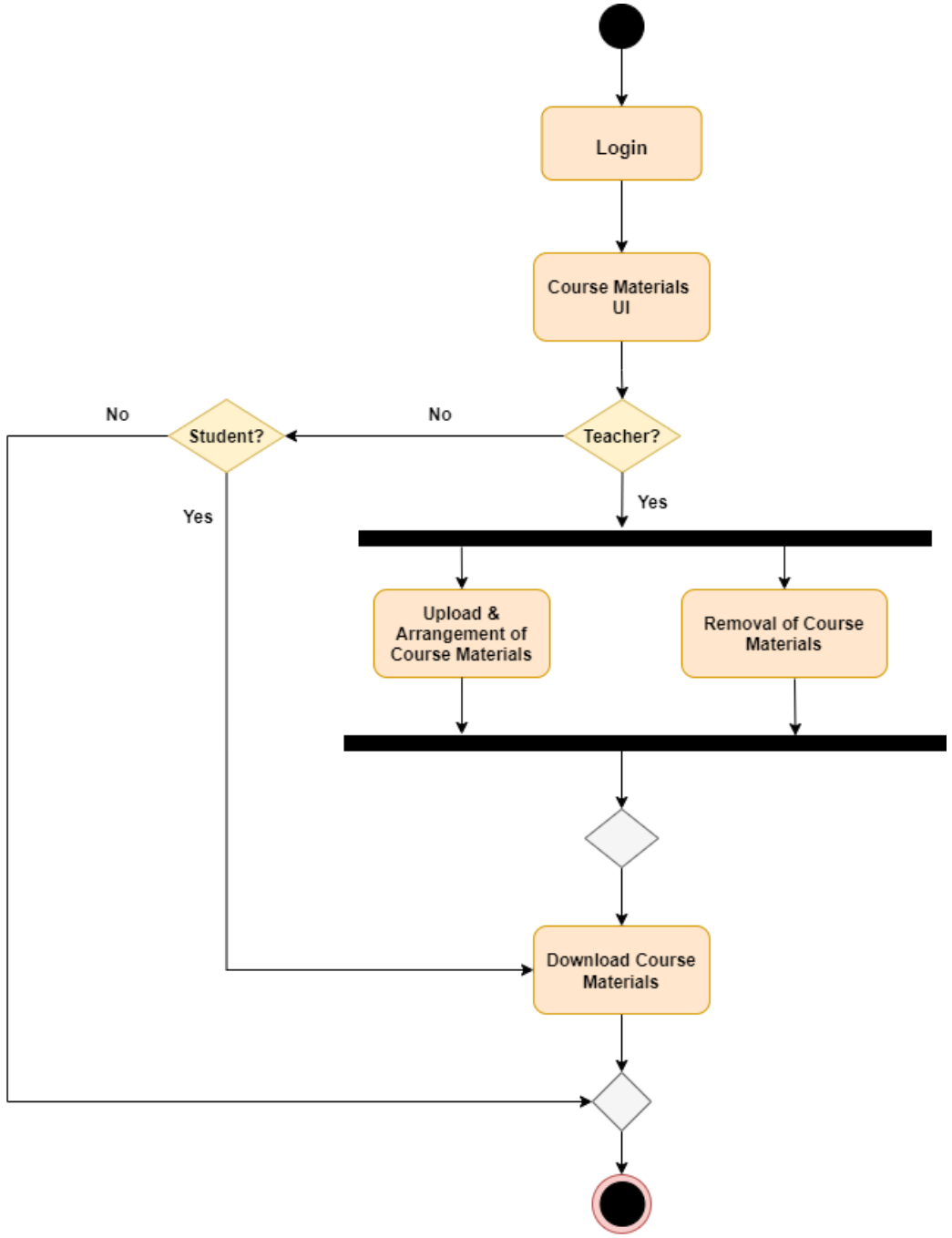


Level 1.2.5 - Online Class Meeting Management

Level-1.2.6

Name : Course Materials Management

Reference : Use Case Level 1.2.6

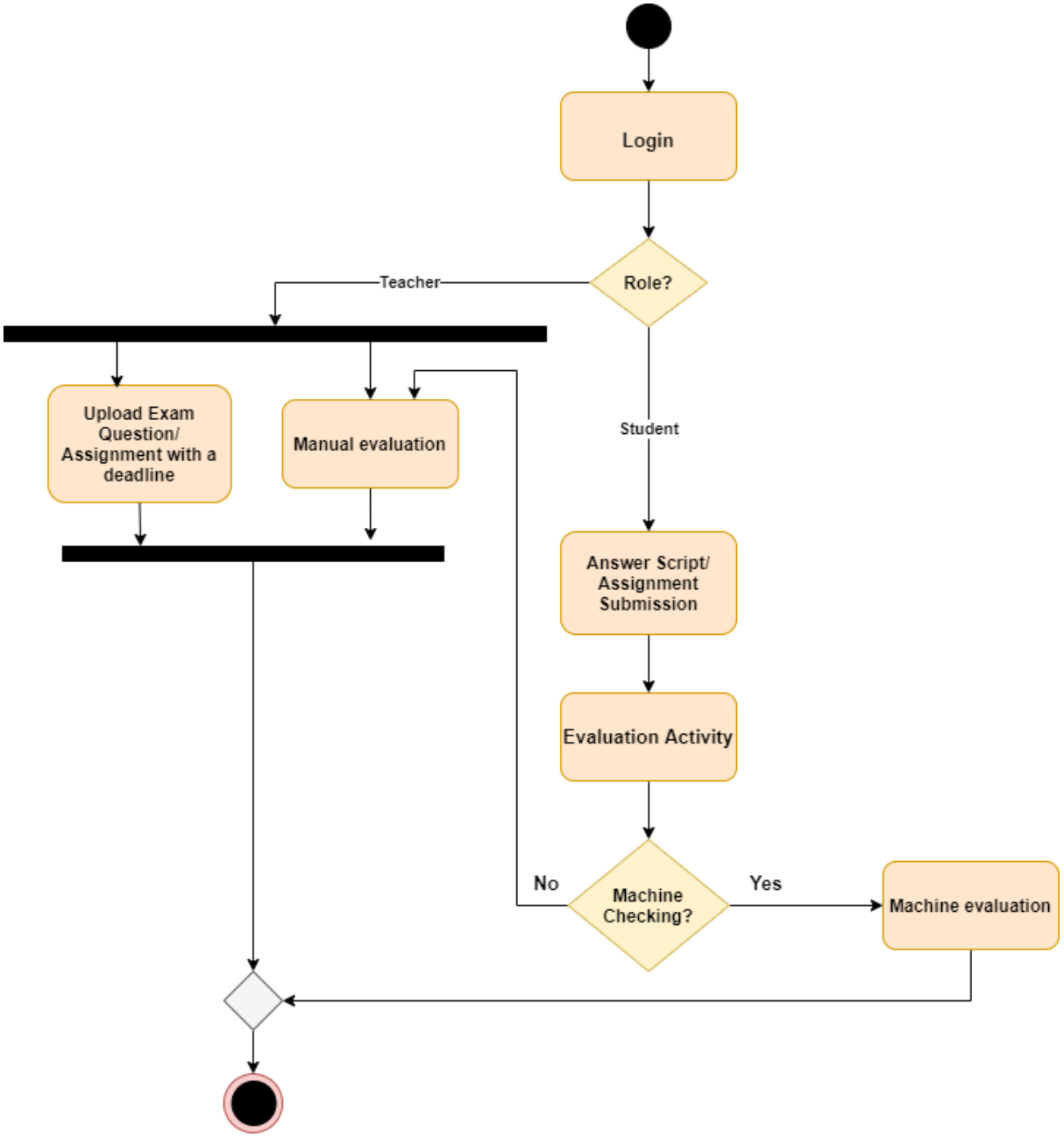


Level 1.2.6 - Course Material Management

Level-1.3

Name : Exam & Assignment related Activities

Reference : Use Case Level 1.3

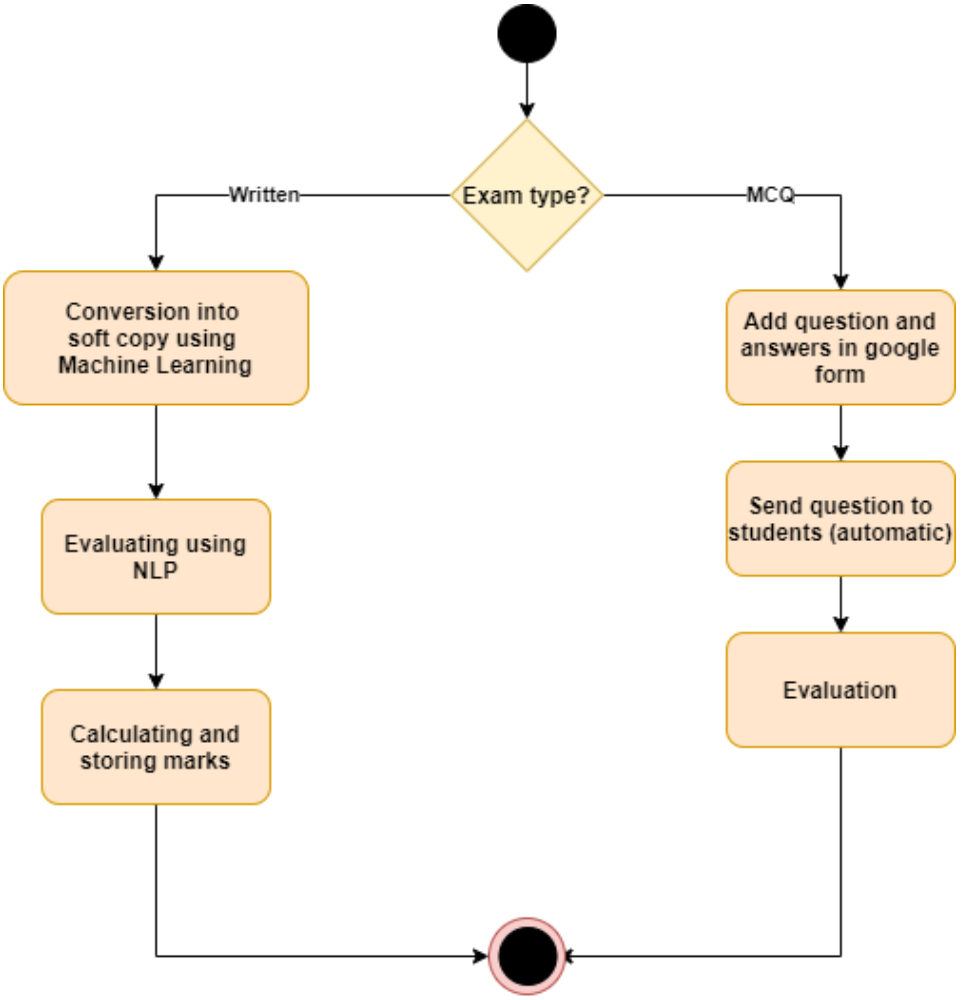


Level 1.3 - Exam & Assignment related Activities

Level-1.3.3

Name : Machine Evaluation

Reference : Use Case Level 1.3.3

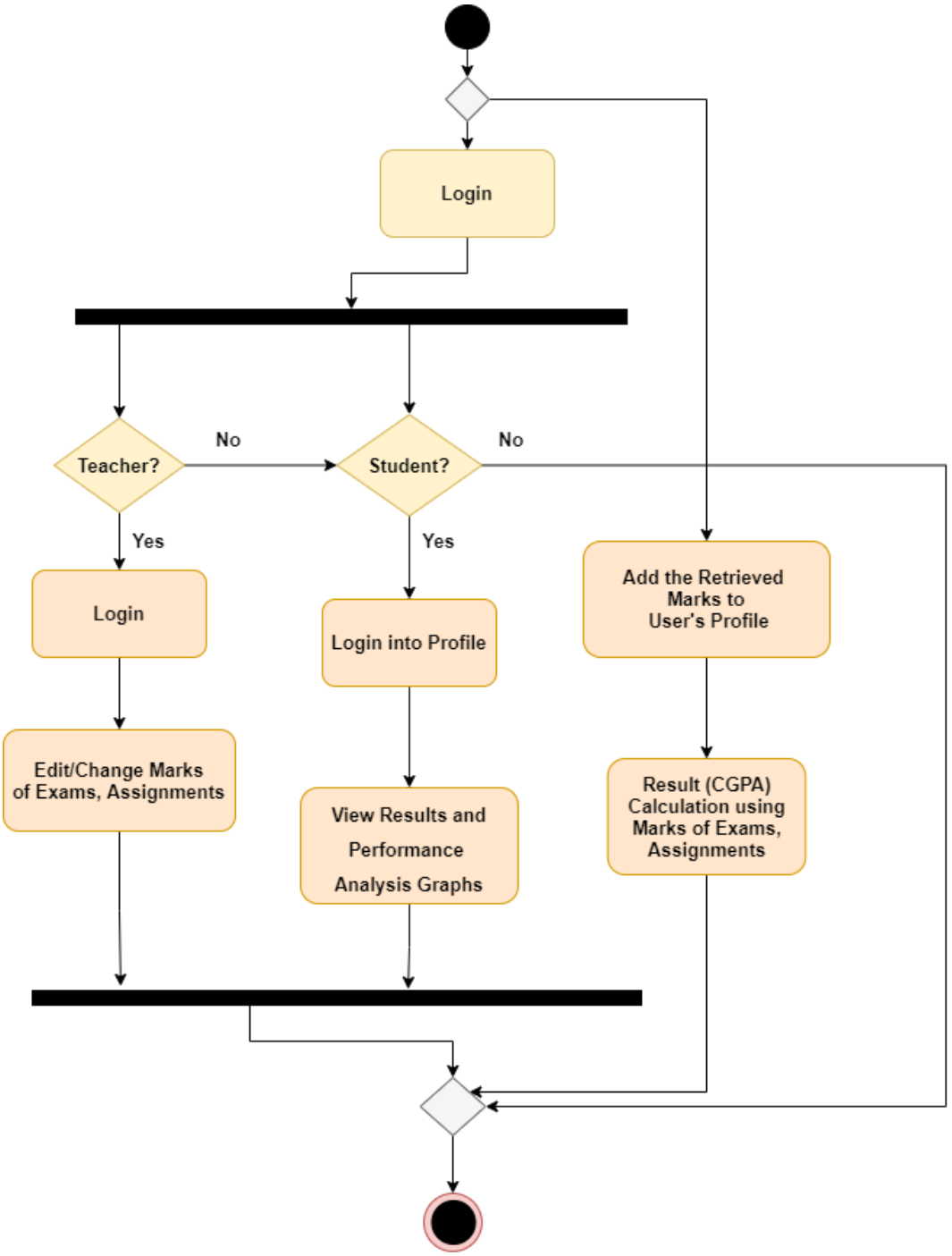


Level 1.3.3 - Machine Evaluation

Level-1.4

Name : Performance, Results & Marks Management

Reference : Use Case Level 1.4

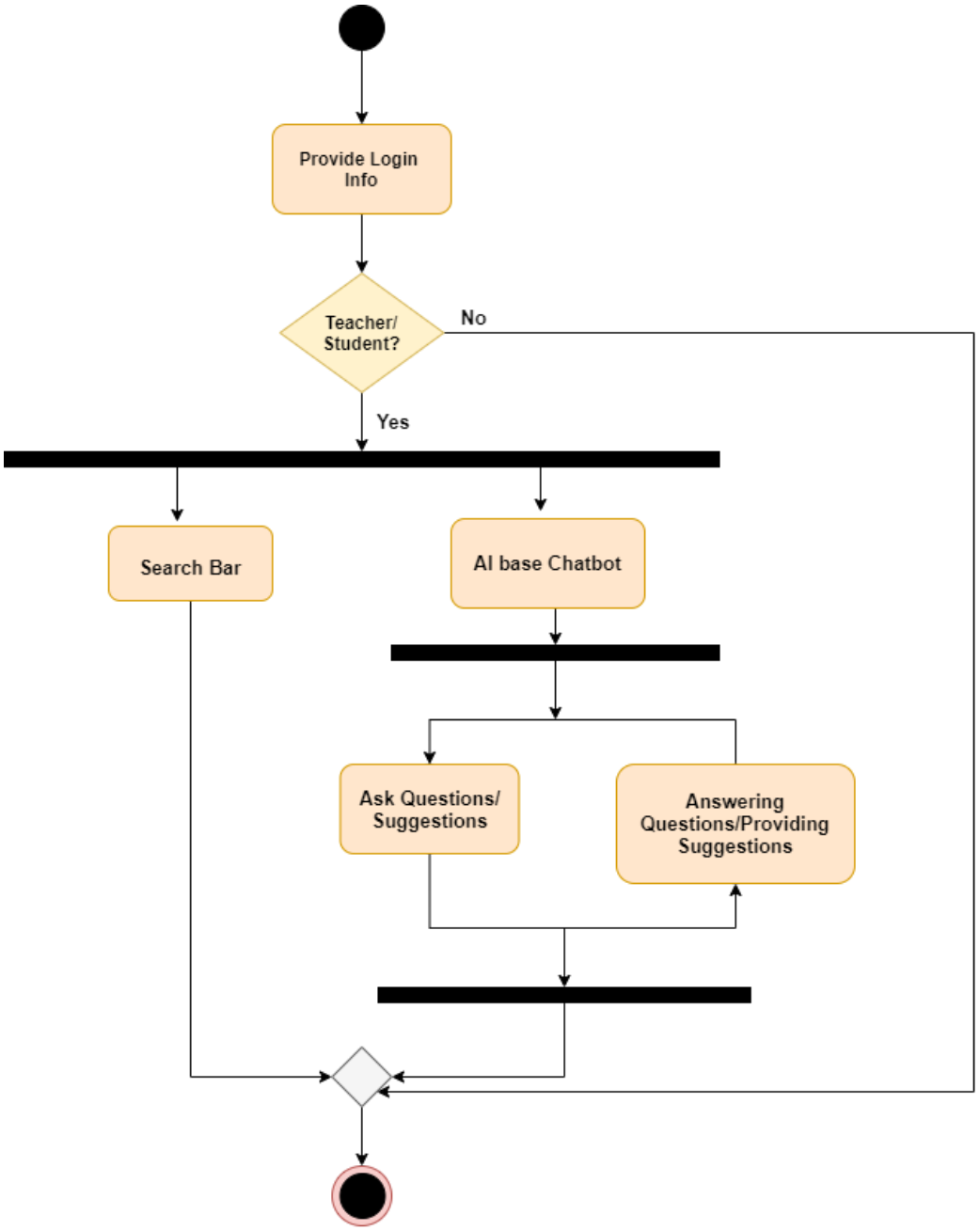


Level 1.4 - Marks, Results & Performance Management

Level-1.5

Name : Inquiry related Activities

Reference : Use Case Level 1.5

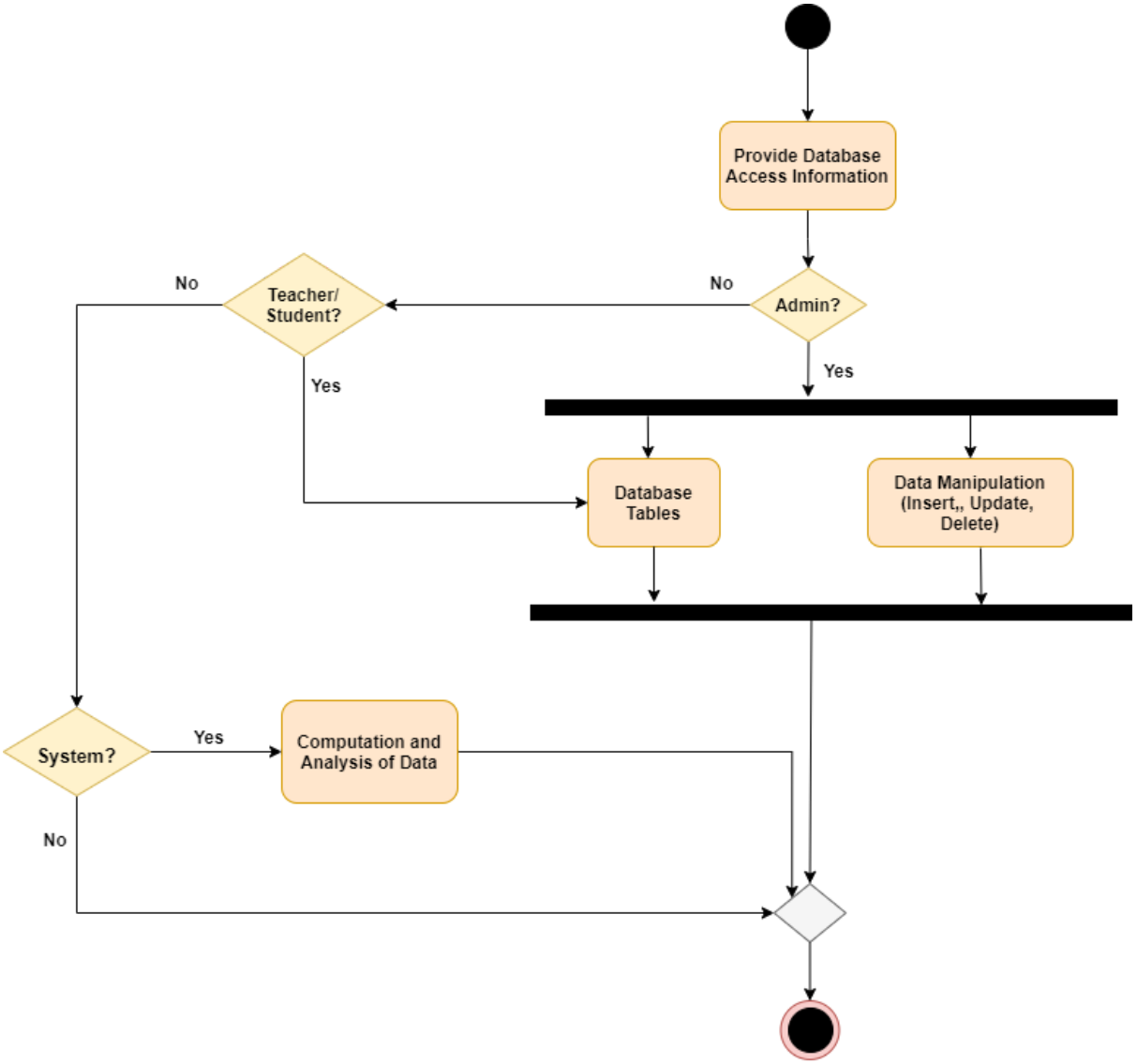


Level 1.5 - Inquiry related Activities

Level-1.6

Name : Database

Reference : Use Case Level 1.6



Level 1.6- Database

Swimlane Diagram: E-shcool

A Swimlane diagram is a type of flowchart that outlines who does what in a given process. Based on the analogy of lanes in a pool, a swimlane diagram places process steps within the horizontal or vertical “swimlanes” of a particular department, work group or employee, thus ensuring clarity and accountability. Highlighting connections and communications between these lanes, it can serve as an indicator of waste, redundancy and inefficiency in a process.

Like any other flowchart, it visualizes a process from beginning to end, using the metaphorical lanes of an actual swimming pool to place the steps of mapping the lanes either vertically or horizontally.

A swim lane diagram is typically used for projects that extend over various departments and distinguish channels according to a specific set of objectives. By organizing the responsibilities in various directions, it can clearly distinguish the objective of each department and individuals inside the team.

Swimlanes (also written as “swim lanes”) represent a valuable element in process flow diagrams (PFDs), as well as in what’s called the Business Process Model and Notation (BPMN) and its software design counterpart – Unified Modeling Language (UML). They introduce parallel (vertical or horizontal) lines to group process steps by actor (which can be a department, work group, employee or even an information system). A swimlane diagram not only spells out processes designated to a specific actor, it also shows how different actors interact to keep a process rolling efficiently.

How to Make a Swimlane Diagram

- Identify the lanes. Then decide what “Actors” are needed to be represented by swimlanes and label them.
- The separation of processes into lanes—either horizontally or vertically—and organizing discrete tasks in sequential order along the other axis.

Add steps. Each step should be connected to the one before it with a line.

The Purpose of Swimlane Diagram

Providing an easy-to-read representation of responsibilities within a process, a swimlane diagram can serve the following purposes:

- To communicate and highlight which process steps or sub-processes are assigned to a particular actor.

- To identify bottlenecks and other inefficiencies, which in turn helps detect redundancies between various lanes, duplicative steps, process delays or capacity constraints that can be later addressed and resolved. All this leads to increased performance and quality.

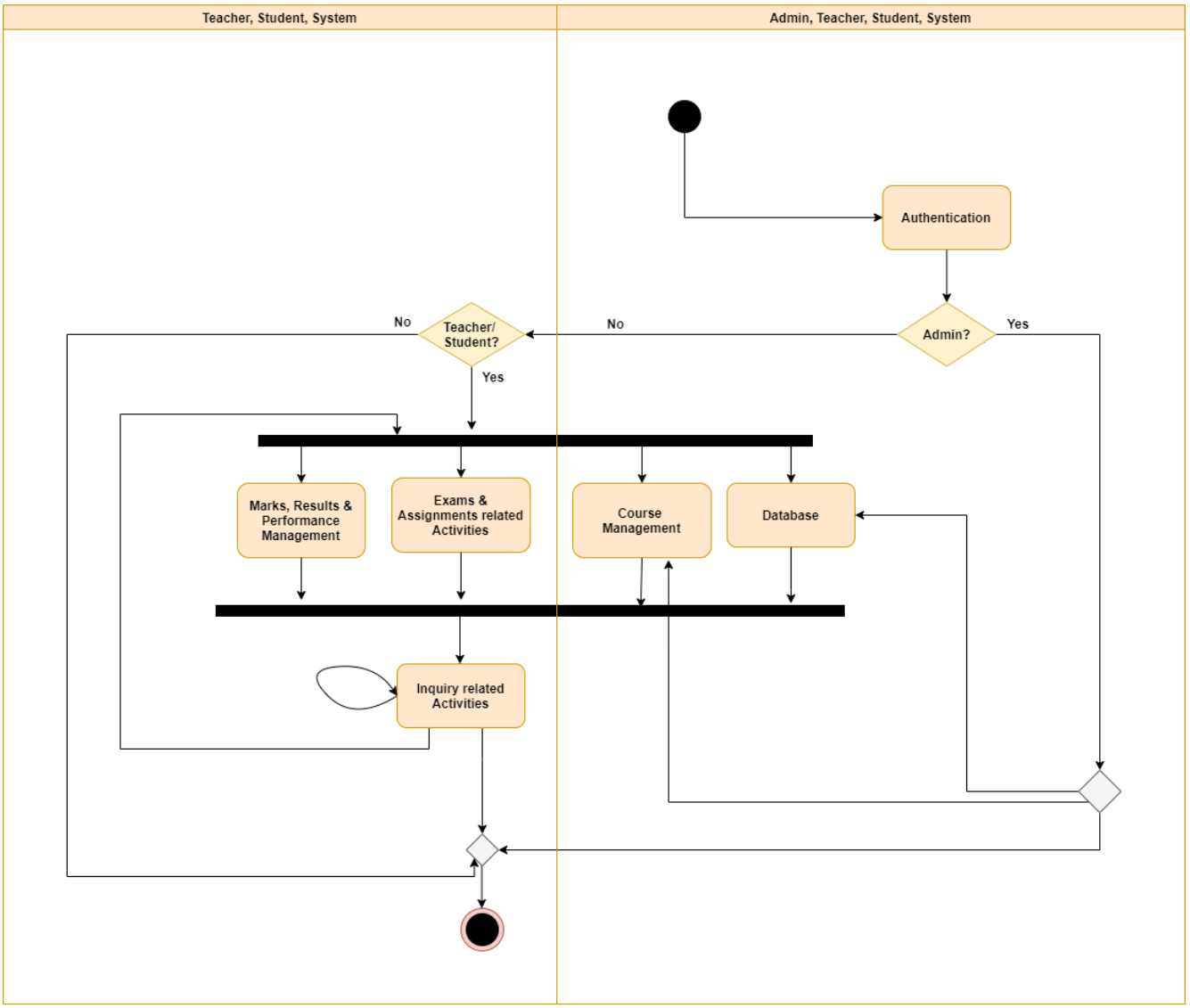
- To better structure a given process and account for evolving circumstances, such as staffing or technology changes.

- To provide a formal model of integrating processes between teams and departments, which results in clearer, more organized workflows on an ongoing basis.

Level-1

Name : E-shcool: An Online Learning Environment

Reference : Use Case & Activity Level 1

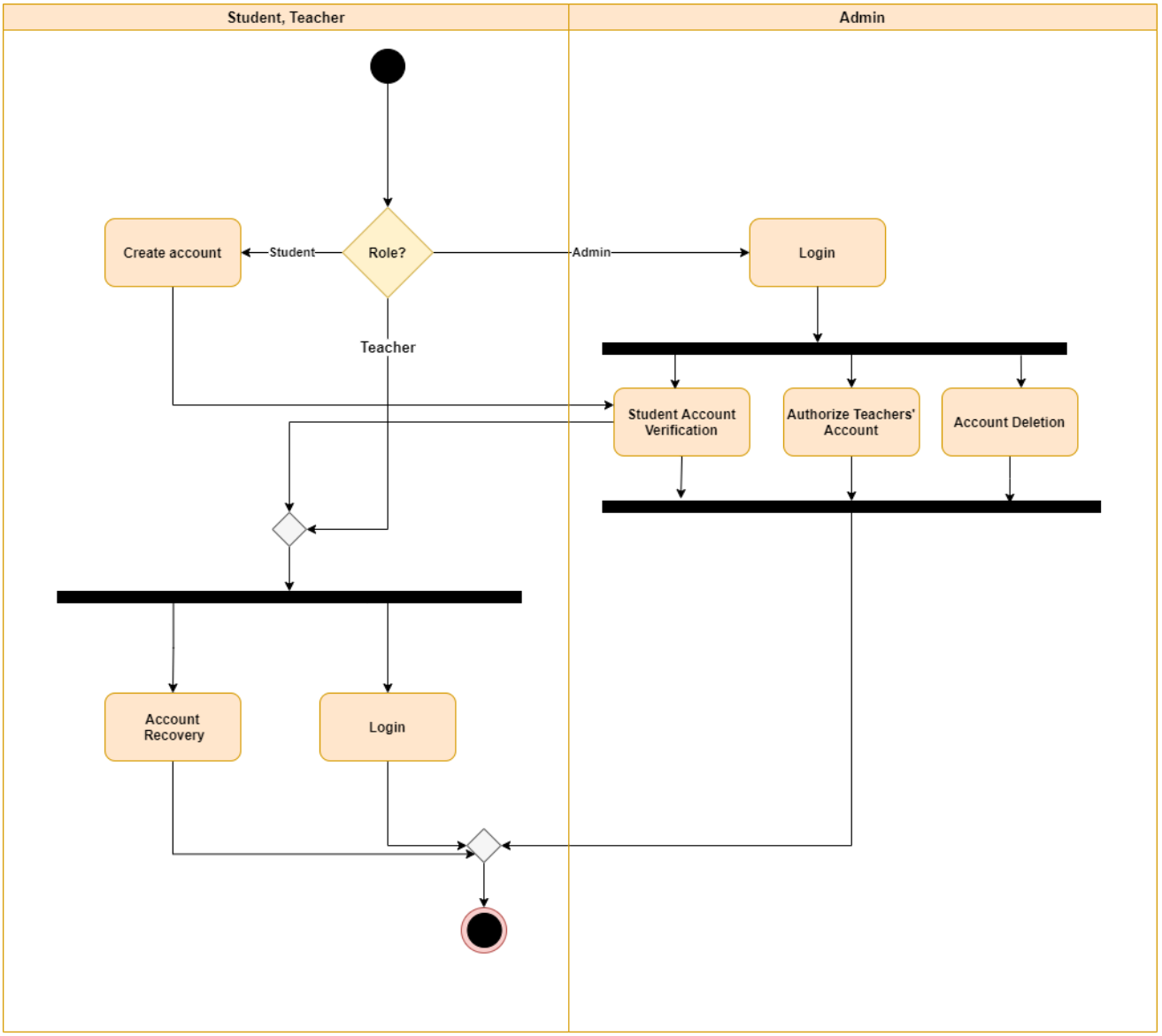


Level 1 - E-shcool: An Online Learning Environment

Level-1.1

Name : Authentication

Reference : Use Case & Activity Level 1.1

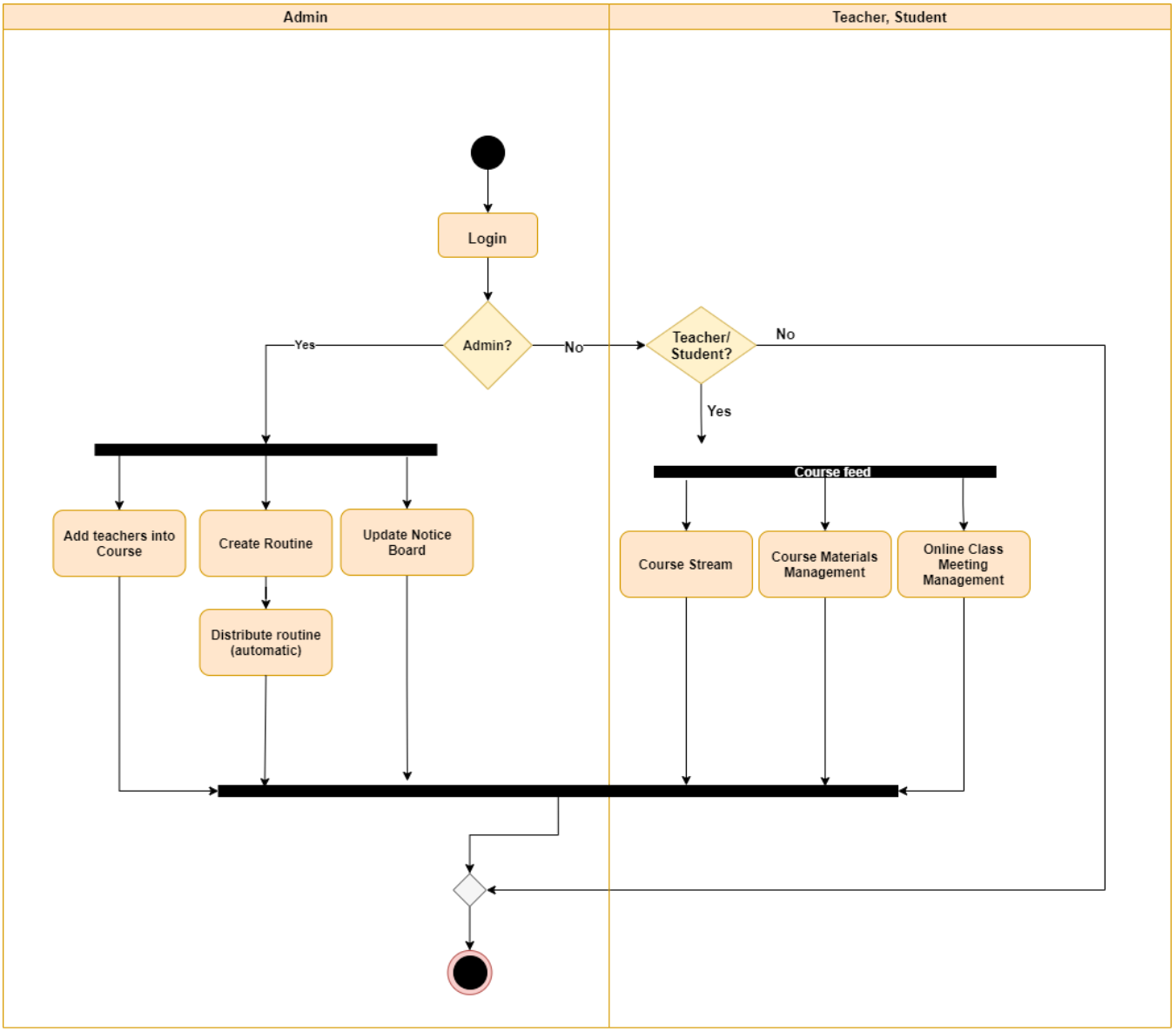


Level 1.1 - Authentication

Level-1.2

Name : Course Management

Reference : Use Case & Activity Level 1.2

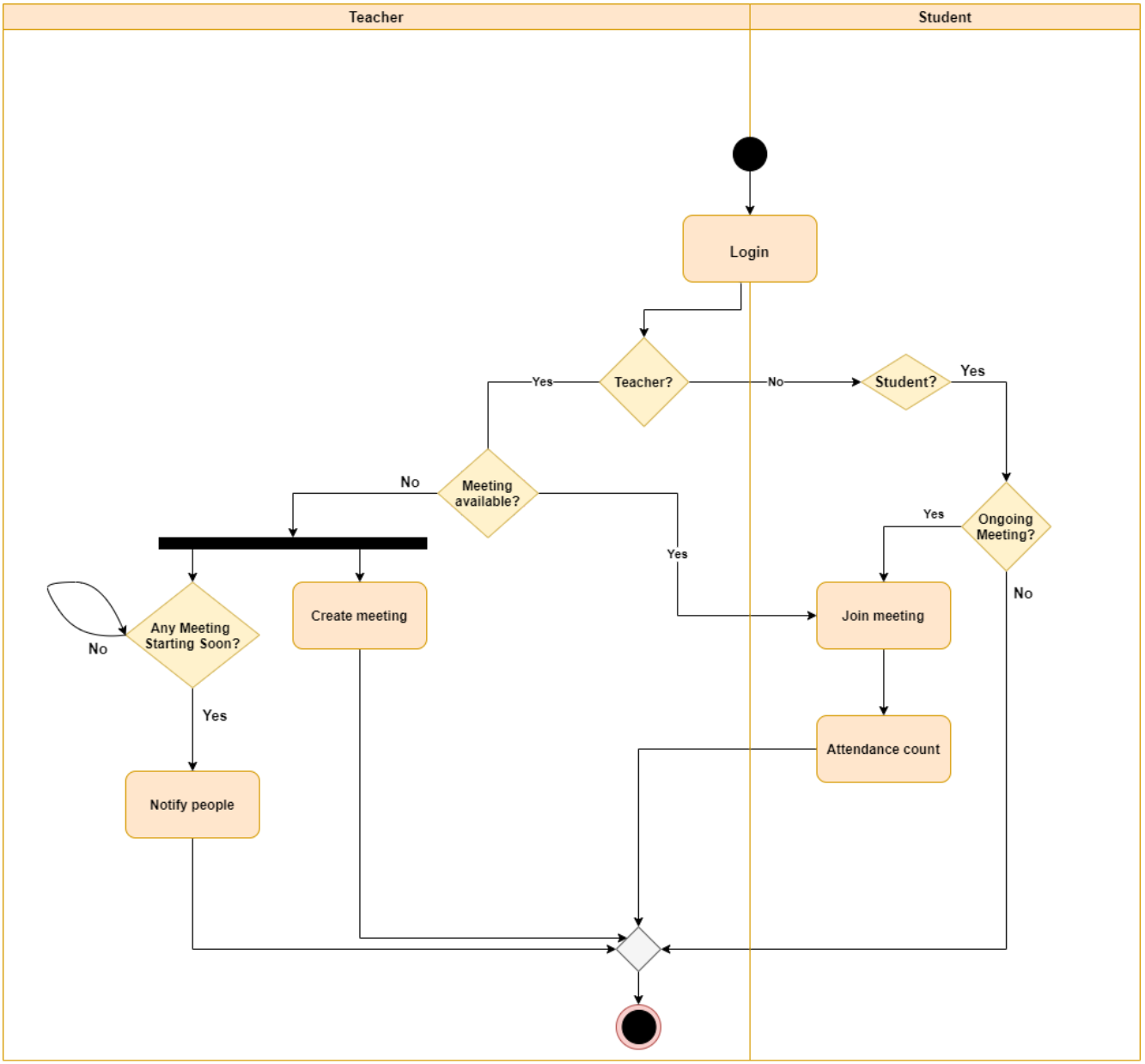


Level 1.2 - Course Management

Level-1.2.5

Name : Online Class Meeting Management

Reference : Use Case & Activity Level 1.2.5

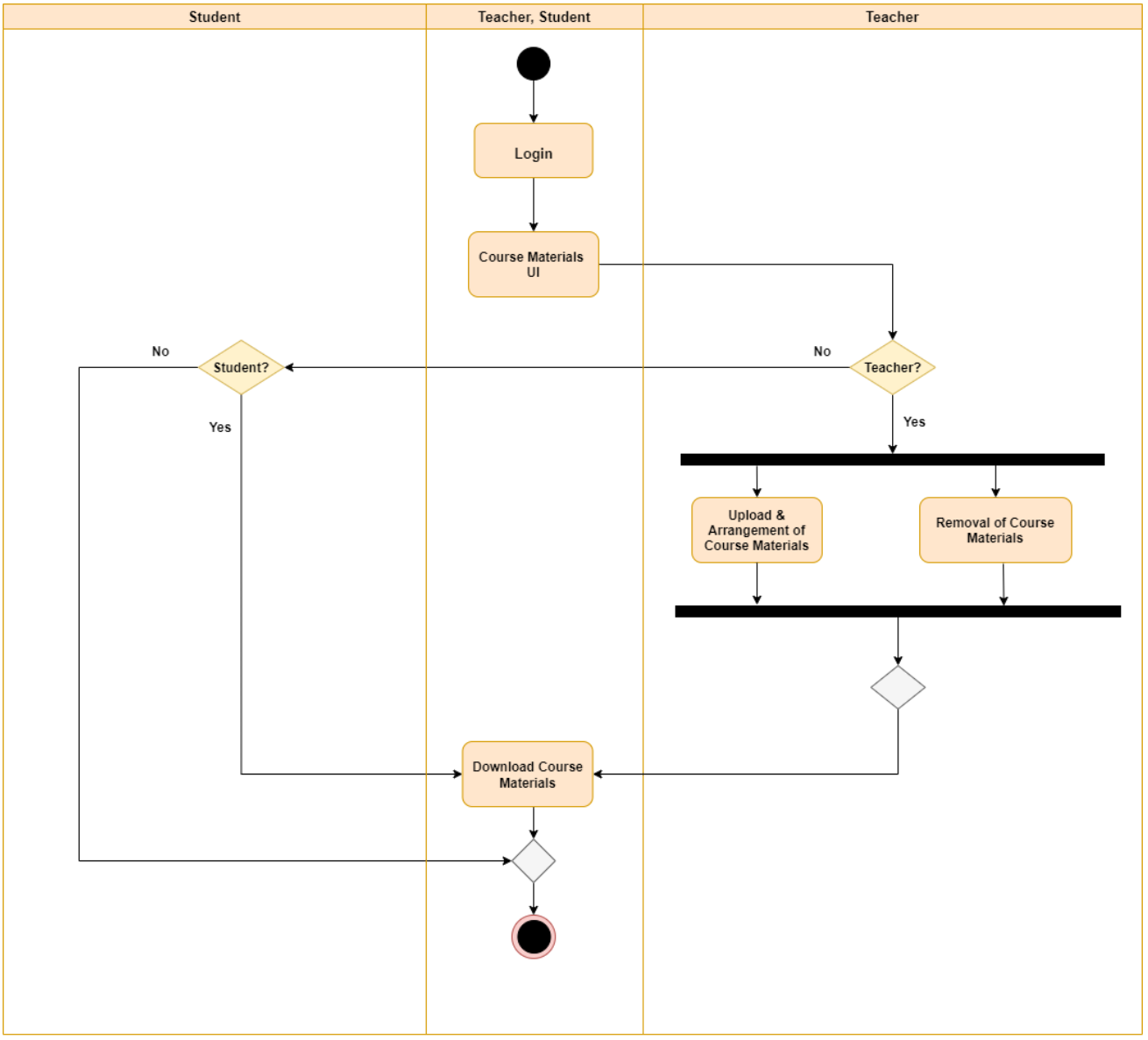


Level 1.2.5 - Online Class Meeting Management

Level-1.2.6

Name : Course Materials Management

Reference : Use Case & Activity Level 1.2.6

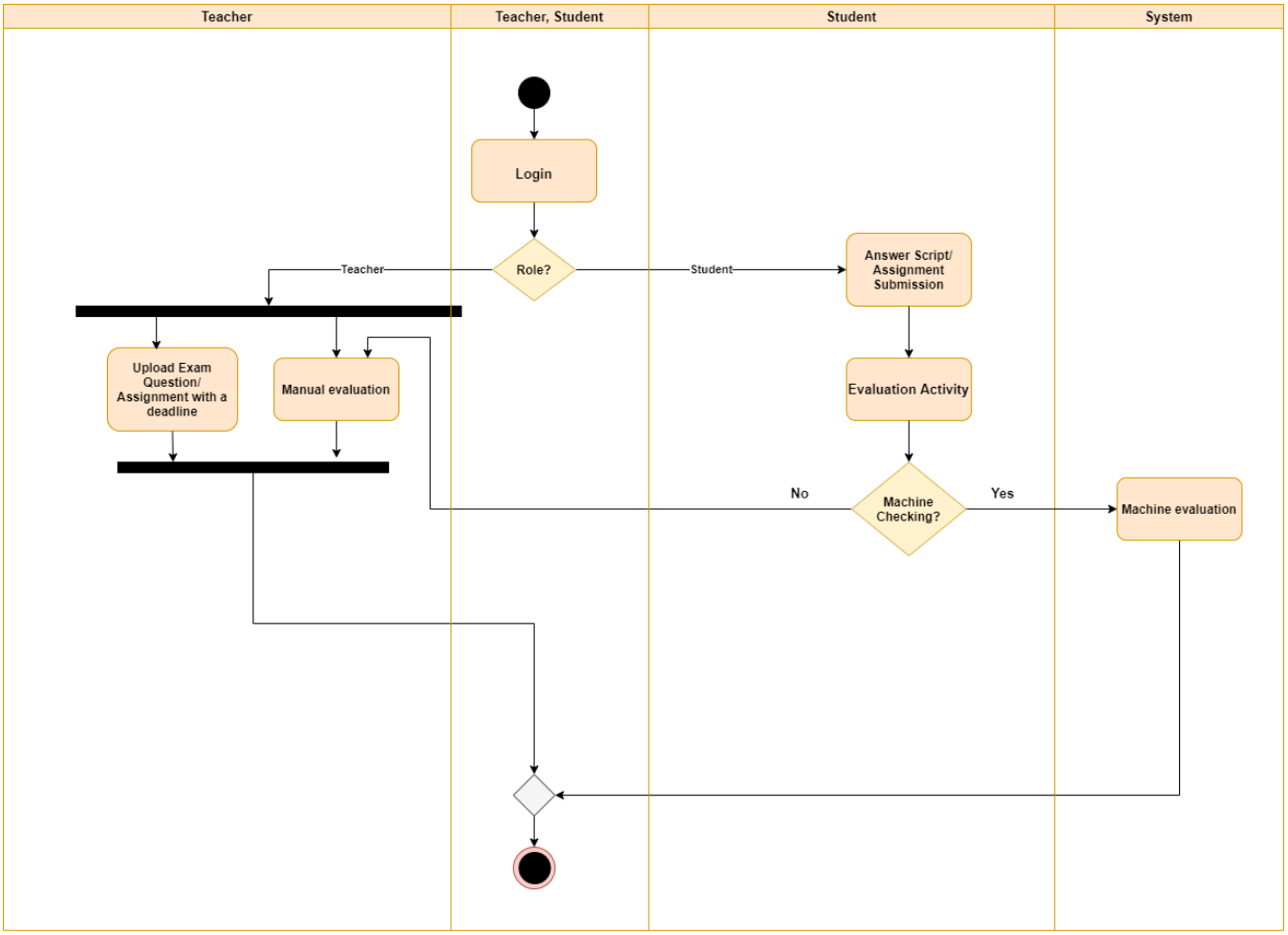


Level 1.2.6 - Course Material Management

Level-1.3

Name : Exam & Assignment related Activities

Reference : Use Case & Activity Level 1.3

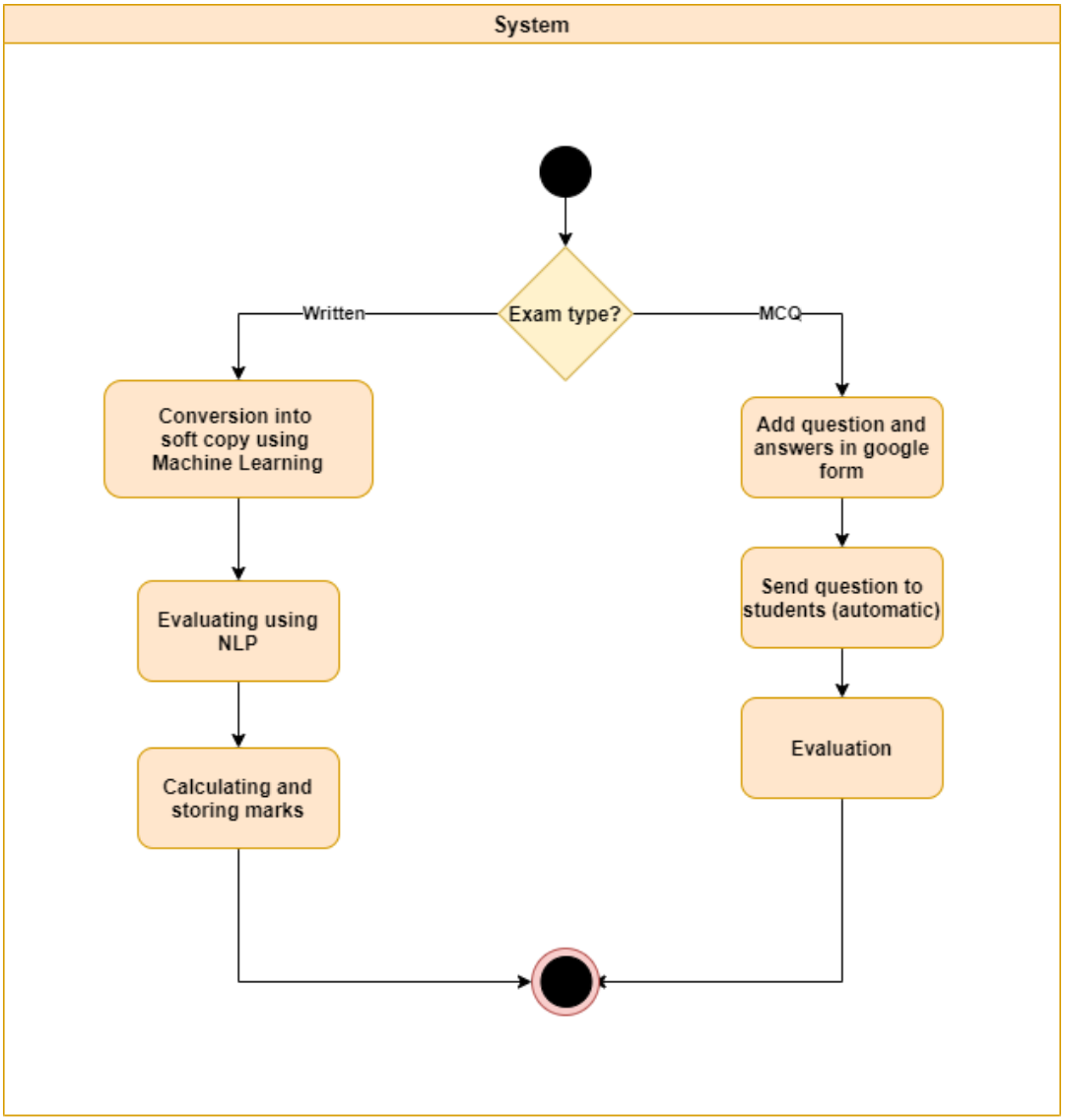


Level 1.3 - Exam & Assignment related Activities

Level-1.3.3

Name : Machine Evaluation

Reference : Use Case & Activity Level 1.3.3

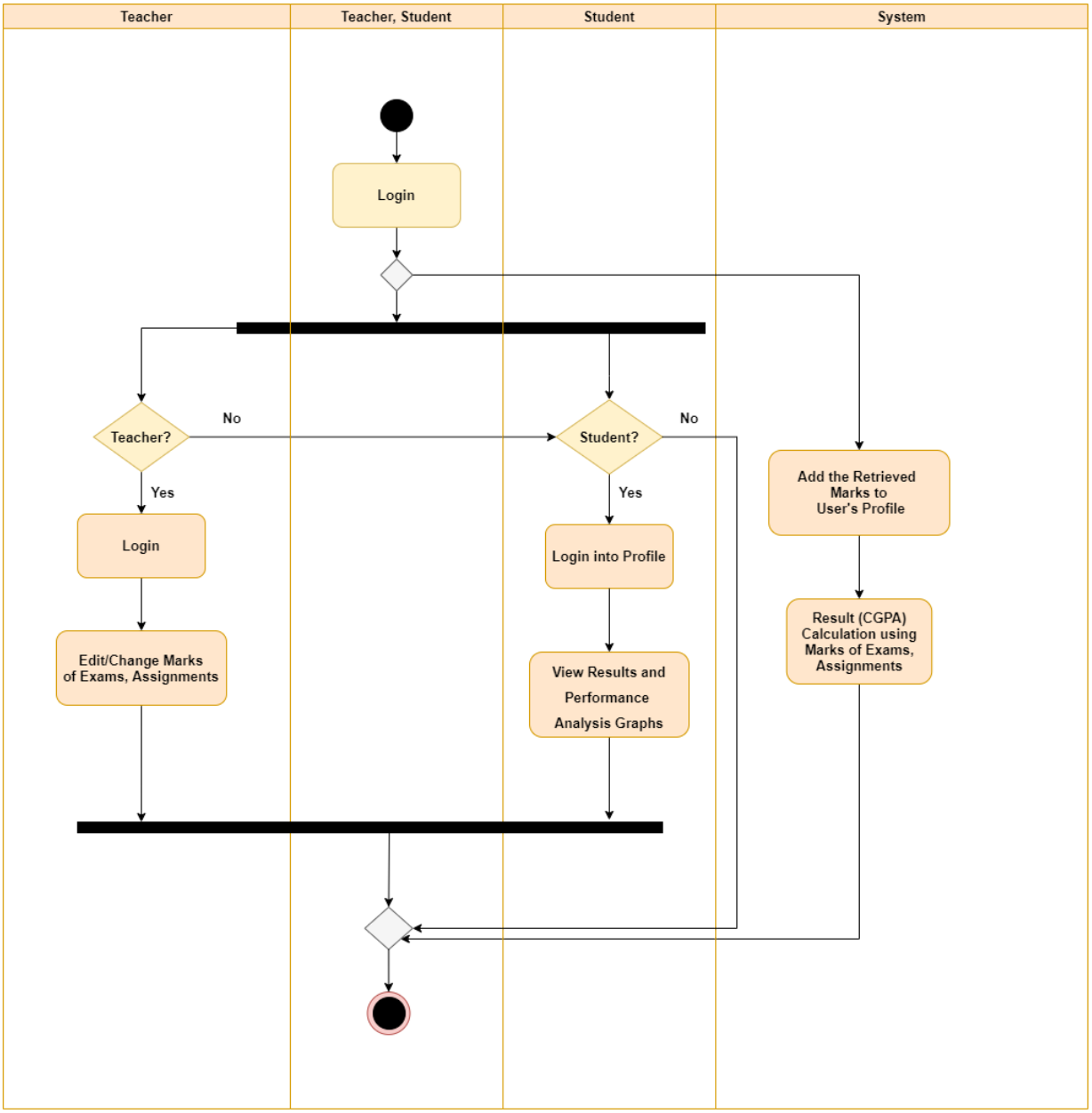


Level 1.3.3 - Machine Evaluation

Level-1.4

Name : Marks, Results & Performance Management

Reference : Use Case & Activity Level 1.4

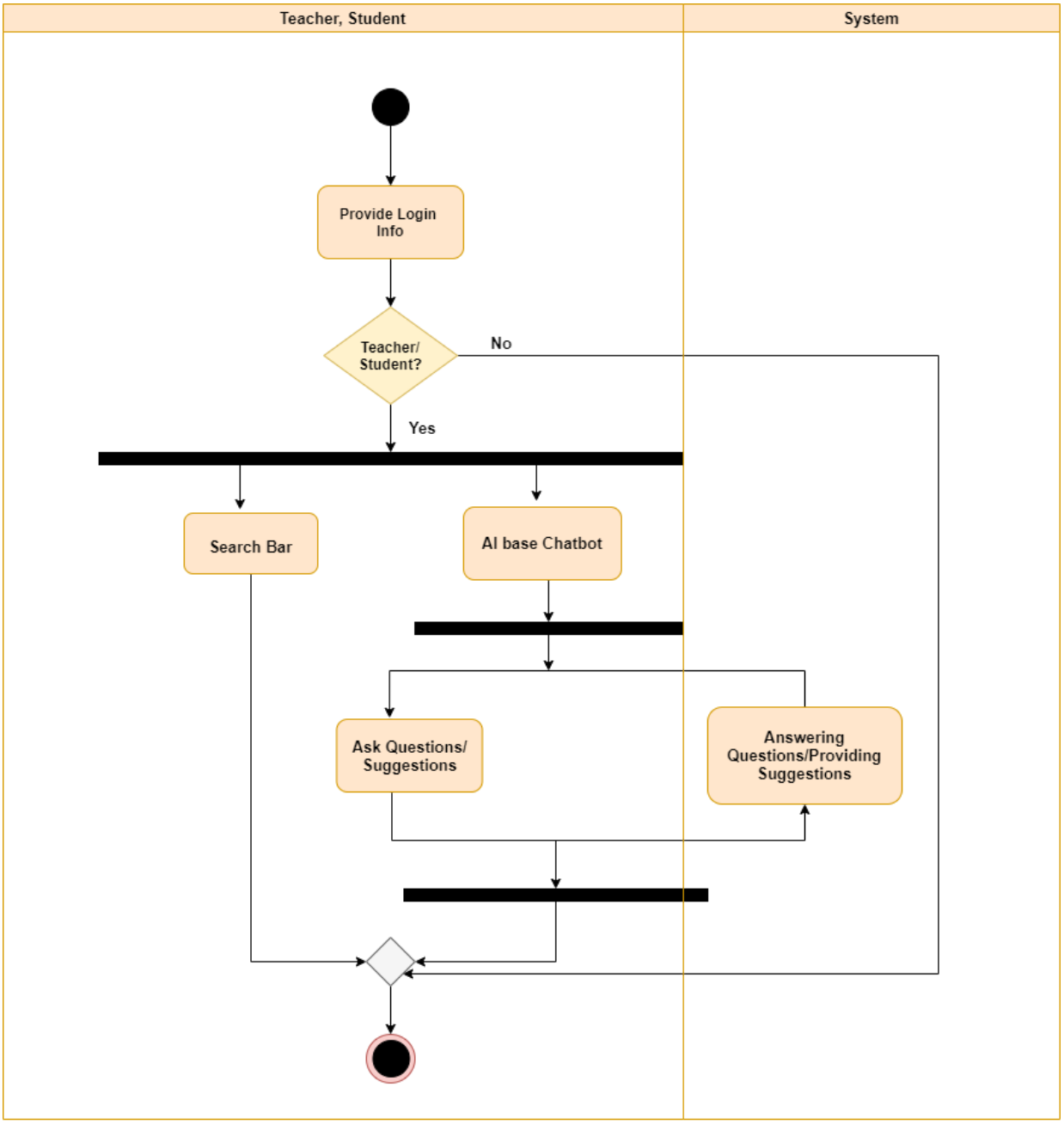


Level 1.4 - Marks, Results & Performance Management

Level-1.5

Name : Inquiry related Activities

Reference : Use Case & Activity Level 1.5

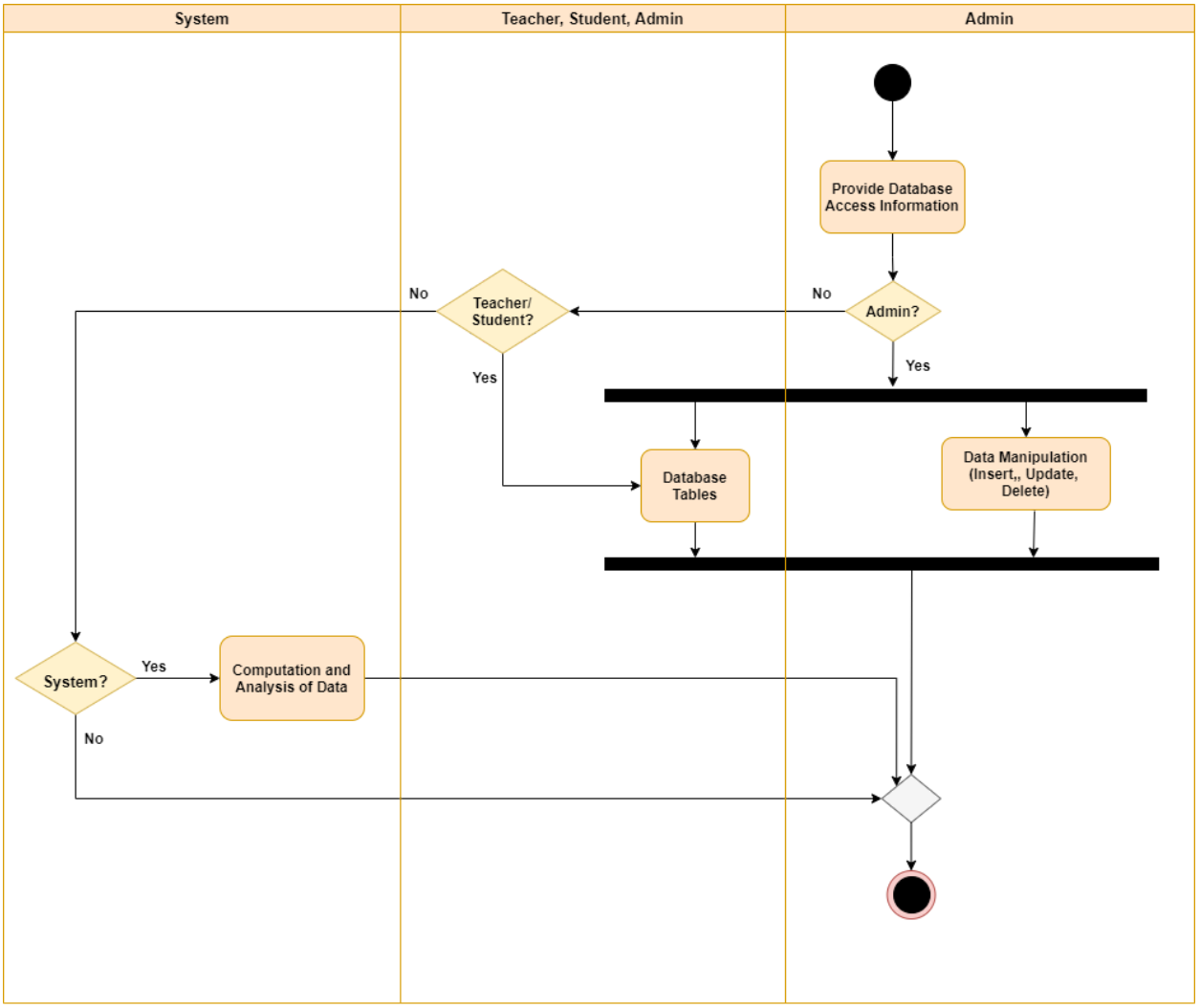


Level 1.5 - Inquiry related Activities

Level-1.6

Name : Database

Reference : Use Case & Activity Level 1.6



Level 1.6- Database

Data Based Modeling : E-shcool

Data Modeling Concept

A Data Model is an organized view of database concepts and their relationships. The purpose of creating a conceptual data model is to establish entities, their attributes, and relationships. In this data modeling level, there is hardly any detail available on the actual database structure.

The 3 basic elements of Data Modeling are-

- Entity: A real-world thing
- Attribute: Characteristics or properties of an entity
- Relationship: Dependency or association between two entities

The entity relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

The main goal of a designing data model is to make certain that data objects offered by the functional team are represented accurately. It should be detailed enough to be used for building the physical database.

Data Objects

A data object is a collection of one or more data points that create meaning as a whole. In other words, it's a storage or container to store data values. More specifically, data objects are usable, functional, and meaningful artifacts whose form and function is to encode data. It can be used in type checking operations. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

Data Object Identification

Noun list

Serial	Noun	Problem/Solution space	Attribute
1	educational platform	p	
2	learning environment	p	
3	academic experience	p	
4	Academic activities	p	
5	**teacher**	s	14,17,20,21
6	**student**	s	14,15,16,17,20,21,36,37,38
7	accessibility	p	
8	features	p	
9	**authentication**	s	17,74,75
10	entities	s	
11	**admin**	s	14,17
12	account	s	
13	profile	s	
14	name	s	
15	roll	s	
16	semester	s	
17	ID	s	
18	information	s	

19	allotment	s	
20	**course**	s	5,6,14,16,17,21,32,34, 35,37,38,40,49,63,71
21	**routine**	s	17,18
22	**notice**	s	17,18
23	Notice board	p	
24	authority	p	
25	database	s	
26	user	s	
27	functionality	p	
28	Course stream	p	
29	**post**	s	17,18,30
30	comment	s	
31	section	p	
32	**material**	s	17,18
33	management	s	
34	**exam**	s	17,18,20,49,53,55,56,7 2
35	**assignment**	s	17,18,20,49,53,55,56,7 2
36	performance	s	
37	**result**	s	6,17,38
38	**marks**	s	20,34,35,71
39	feedback	s	
40	meeting	s	
41	module	p	

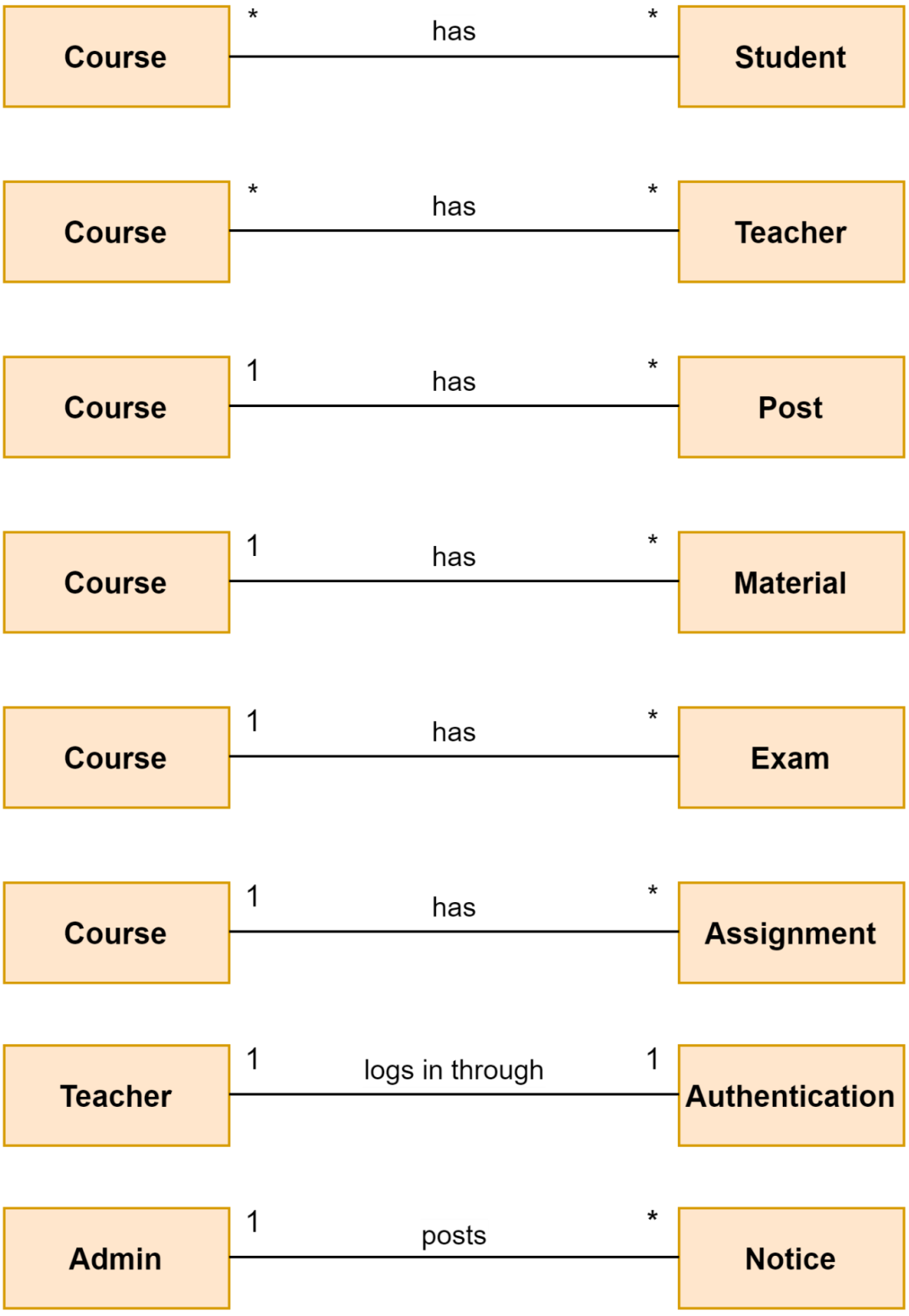
42	notification	s	
43	recording	s	
44	slide	s	
45	pdf	s	
46	folder	s	
47	upload	s	
48	evaluation	s	
49	deadline	s	
50	MCQ exam	s	
51	Written exam	s	
52	Google form	s	
53	Question paper	s	
54	manual	s	
55	scripts	s	
56	Sample solution	s	
57	Handwritten scripts	s	
58	Typed softcopy	s	
59	Machine learning	s	
60	NLP algorithms	s	
61	Reminder section	s	
62	Upcoming deadlines	s	
63	chatbot	s	
64	answer	s	
65	suggestion	s	

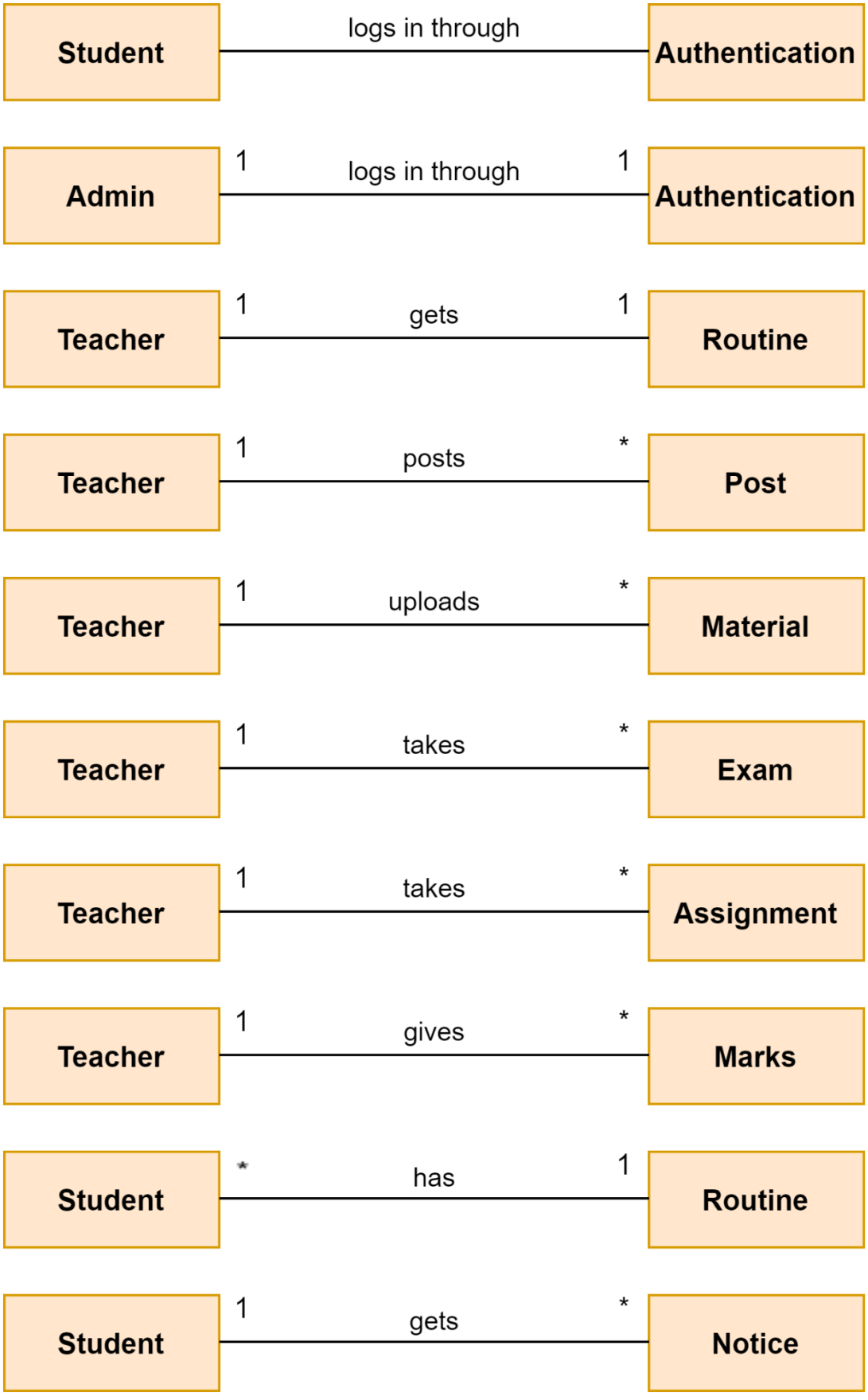
66	class	s	
67	search	s	
68	inquiry	s	
69	Course tile	s	
70	Ongoing meeting	s	
71	attendance	s	
72	submission	s	
73	quiz	s	
74	e-mail	s	
75	password	s	
76	Zoom	s	

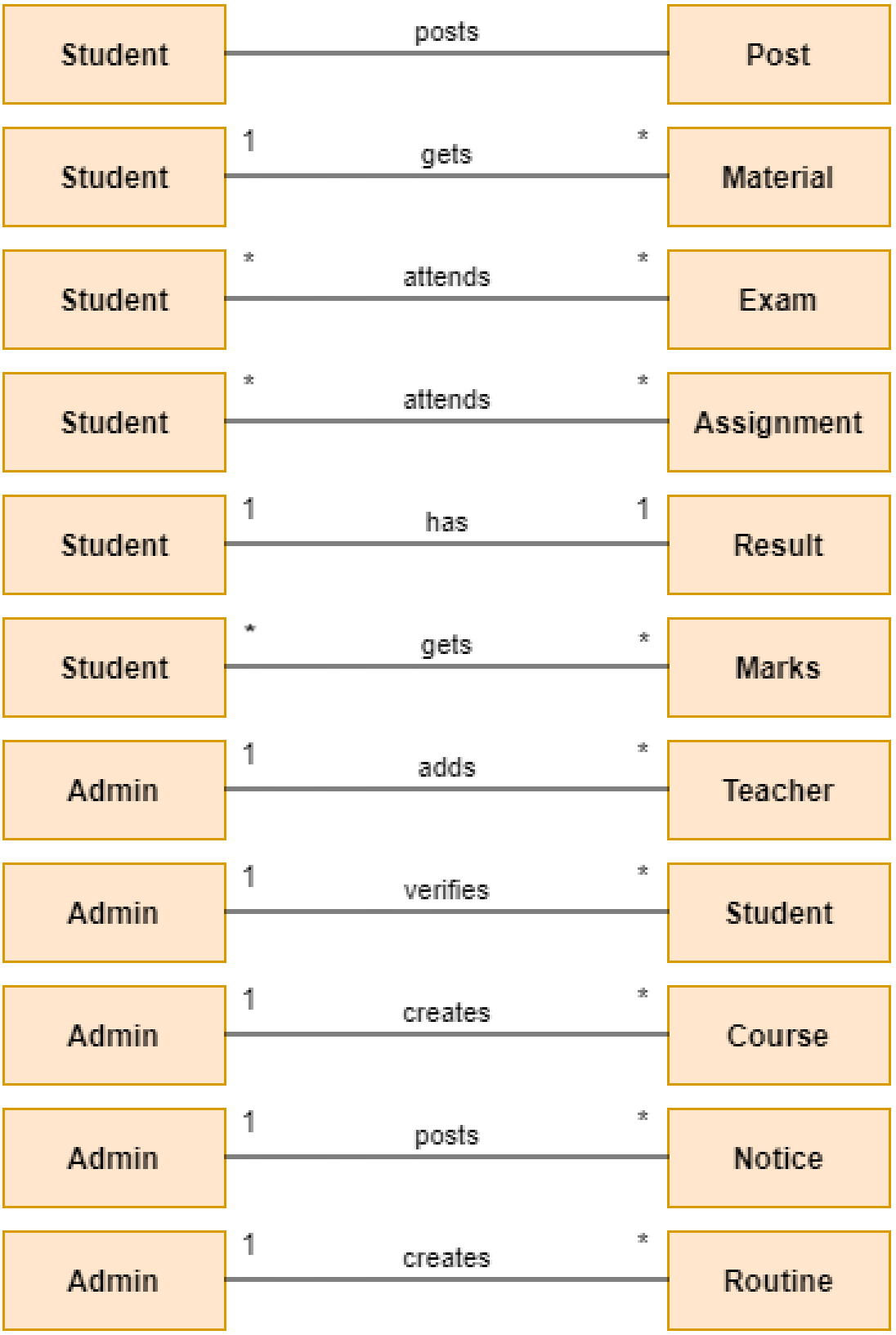
Final Data Objects

1. Teacher
2. Student
3. Admin
4. Authentication
5. Course
6. Routine
7. Notice
8. Post
9. Material
10. Exam
11. Assignment
12. Result
13. Marks

Relationship Between Data Objects




Student
logs in through
Authentication
Admin
1
logs in through
1
Authentication
Teacher
1
gets
1
Routine
Teacher
1
posts
*
Post
Teacher
1
uploads
*
Material
Teacher
1
takes
*
Exam
Teacher
1
takes
*
Assignment
Teacher
1
gives
*
Marks
Student
*
has
1
Routine
Student
1
gets
*
Notice


Student
posts
Post
Student
1
gets
*
Material
Student
*
attends
*
Exam
Student
*
attends
*
Assignment
Student
1
has
1
Result
Student
*
gets
*
Marks
Admin
1
adds
*
Teacher
Admin
1
verifies
*
Student
Admin
1
creates
*
Course
Admin
1
posts
*
Notice
Admin
1
creates
*
Routine

ER Diagram

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

Here are the geometric shapes and their meaning in an E-R Diagram:

Rectangle: Represents Entity sets.

Ellipses: Attributes

Diamonds: Relationship Set

Lines: They link attributes to Entity Sets and Entity sets to Relationship Set

Double Ellipses: Multivalued Attributes

Dashed Ellipses: Derived Attributes

Double Rectangles: Weak Entity Sets

Double Lines: Total participation of an entity in a relationship set.

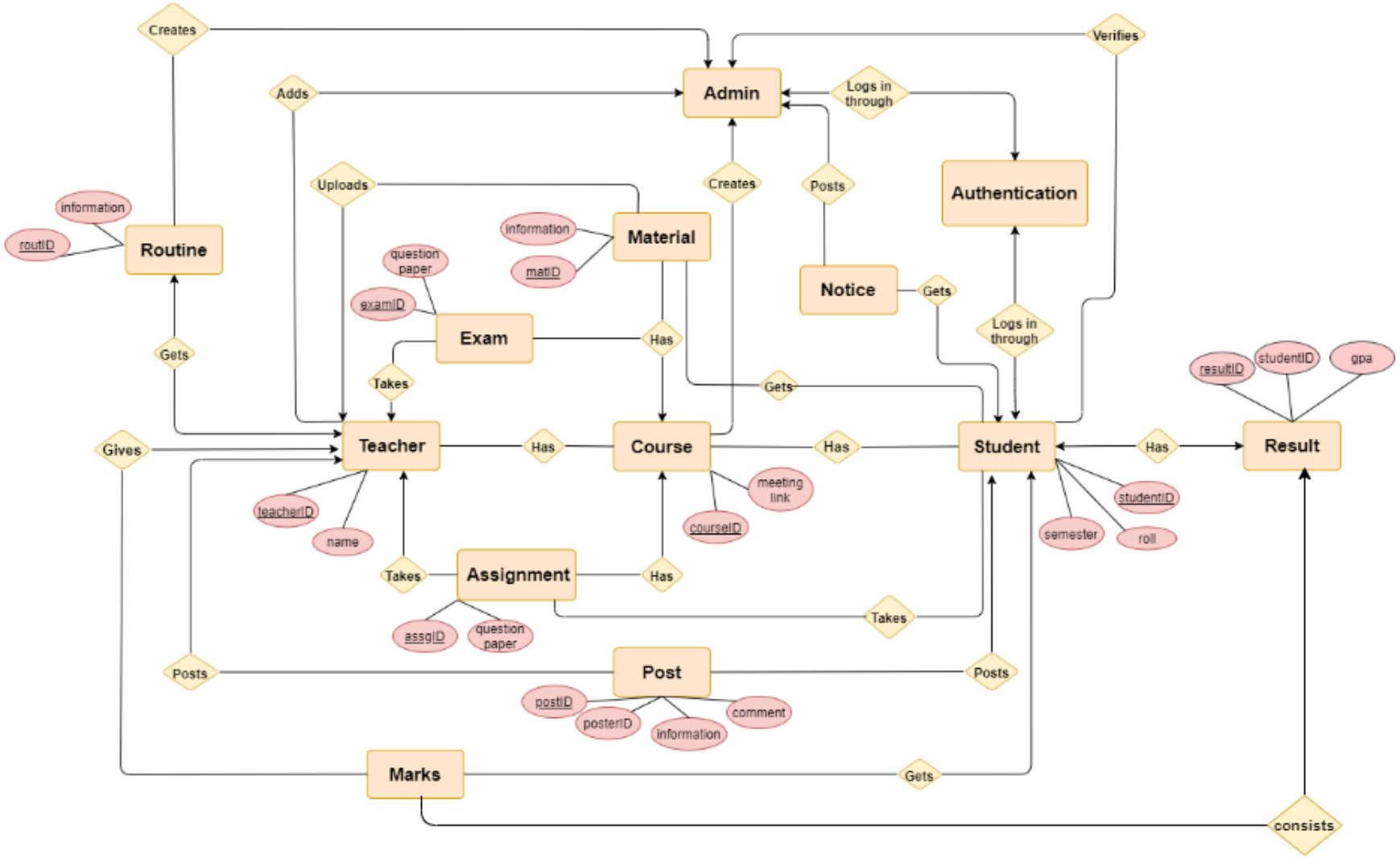


Figure: ER Diagram for all data Object

Schema

Data Object	Attributes	Type	Size
Teacher	teacherID Name	Number Varchar	6 40
Student	studentID Name Roll Semester Performance routID(FK)	Number Varchar Number Number Varchar	6 40 6 6 40
Admin	adminID Name	Number Varchar	6 40
Course	courseID Name Semester Meeting_link adminID (FK)	Number Varchar Number Varchar Number	6 40 6 40 6
Authentication	authID E-mail Password	Number Varchar Varchar	6 40 40
Routine	routID Information adminID(FK)	Number Varchar Number	6 40 6
Notice	notID Information adminID(FK) teacherID(FK) studentID(FK)	Number Varchar Number Number Number	6 40 6 6 6

Post	postID Information Comment posterID(FK) courseID(FK)	Number Varchar Varchar Number Number	6 40 40 6 6
Material	matID Information courseID(FK) teacherID(FK) studentID(FK)	Number Varchar Number Number Number	6 40 6 6 6
Exam	examID Information Deadline Question_paper Sample_solution Scripts studentID(FK) teacherID(FK) courseID(FK)	Number Varchar Varchar Varchar Varchar Varchar Number Number Number	6 40 40 40 40 40 6 6 6
Assignment	assignmentID Information Deadline Question_paper Scripts studentID(FK) teacherID(FK) courseID(FK)	Number Varchar Varchar Varchar Varchar Number Number Number	6 40 40 40 40 6 6 6
Result	resultID studentID(FK) gpa	Number Number Number	6 6 6

Marks	marksID Attendance teacherID(FK) examID(FK) resultID(FK) marks	Number Number Number Number Number Number Number	6 6 6 6 6 6 6
Teaching (Joined)	courseID teacherID	Number Number	6 6
Attending (Joined)	courseID studentID	Number Number	6 6
StudentExam (Joined)	examID studentID	Number Number	6 6
StuAssignment (Joined)	assigID studentID	Number Number	6 6
Marking (joined)	marksID studentID	Number Number	6 6

Class Based Modeling : E-shcool

Class Based Modeling Concept

Class-based modeling identifies classes, attributes and relationships that the system will use. It represents the object. The system manipulates the operations.

The elements of the class based model consist of classes and object, attributes, operations, class – responsibility - collaborator (CRC) models.
Classes are determined using underlining each noun or noun clause and entering it into the simple table. Attributes are the set of data objects that are defining a complete class within the context of the problem. The operations define the behavior of an object.

A class-based model contains-

- **Objects**

  What the system will manipulate

- **Operations** (methods or services)

  What to be applied to the objects to effect the manipulation

- **Relationships** (some hierarchical) between objects

- **Collaborations** between the classes that are defined

Solution Space Noun List

Serial	Noun
1	**teacher**
2	**student**
3	**authentication**
4	**entities**
5	**admin**
6	**account**
7	**profile**
8	**name**
9	**roll**
10	**semester**
11	**ID**
12	**information**
13	**allotment**
14	**course**
15	**routine**
16	**notice**
17	**database**
18	**user**
19	**post**
20	**comment**
21	**material**
22	**management**

23	**exam**
24	**assignment**
25	**performance**
26	**result**
27	**marks**
28	**feedback**
29	**meeting**
30	**notification**
31	**recording**
32	**slide**
33	**pdf**
34	**folder**
35	**upload**
36	**evaluation**
37	**deadline**
38	**MCQ exam**
39	**Written exam**
40	**Google form**
41	**Question paper**
42	**manual**
43	**scripts**
44	**Sample solution**
45	**Handwritten scripts**
46	**Typed softcopy**
47	**Machine learning**

48	**NLP algorithms**
49	**Reminder section**
50	**Upcoming deadlines**
51	**chatbot**
52	**answer**
53	**suggestion**
54	**class**
55	**search**
56	**inquiry**
57	**Course tile**
58	**Ongoing meeting**
59	**attendance**
60	**submission**
61	**quiz**
62	**e-mail**
63	**password**

Verb list

Serial	Verb
1	**provide**
2	**ensure**
3	**operate**
4	**manage**
5	**automate**
6	**have**
7	**handle**
8	**verify**
9	**approve**
10	**create**
11	**follow**
12	**enroll**
13	**allot**
14	**view**
15	**assign**
16	**generate**
17	**send**
18	**control**
19	**update**
20	**see**
21	**add**
22	**update**

23	**delete**
24	**enter**
25	**create**
26	**contain**
27	**give**
28	**schedule**
29	**upload**
30	**remove**
31	**organize**
32	**evaluate**
33	**post**
34	**take**
35	**add**
36	**check**
37	**convert**
38	**show**
39	**join**
40	**count**
41	**access**
42	**download**
43	**submit**

General Classification

Candidate classes were then characterized in seven general classes. The seven general characteristics are as follows :

1. **External entities**
2. **Things**
3. **Events**
4. **Roles**
5. **Organizational units**
6. **Places**
7. **Structures**

Table of General Classification

Serial No.	Noun	General Classification
1	**teacher**	**4, 5, 7**
2	**student**	**4, 5 , 7**
3	**authentication**	**3**
4	**entities**	**2**
5	**admin**	**4, 5, 7**
6	**account**	**2, 4, 5, 7**
7	**profile**	**2, 4, 7**
8	**name**	**2**
9	**roll**	**2**
10	**semester**	**2**

11	**ID**	**2**
12	**information**	**2**
13	**allotment**	**3**
14	**course**	**2, 5, 7**
15	**routine**	**2**
16	**notice**	**3, 5, 7**
17	**database**	**2, 4, 7**
18	**user**	**4**
19	**post**	**3**
20	**comment**	**3**
21	**materials**	**2, 5, 7**
22	**management**	**3**
23	**exam**	**2, 5, 7**
24	**assignment**	**2, 5, 7**
25	**performance**	**3**
26	**result**	**2, 5, 7**
27	**marks**	**2, 5**
28	**feedback**	**3**
29	**meeting**	**3, 5, 7**
30	**notification**	**2, 7**
31	**recording**	**2**
32	**slide**	**2**
33	**pdf**	**2**
34	**folder**	**2**
35	**upload**	**3**

36	**evaluation**	**3**
37	**deadline**	**3**
38	**MCQ exam**	**2**
39	**Written exam**	**2**
40	**Google form**	**1**
41	**Question paper**	**2**
42	**manual**	**3, 7**
43	**scripts**	**2**
44	**Sample solution**	**2**
45	**Handwritten scripts**	**2**
46	**Typed softcopy**	**2**
47	**Machine learning**	**2**
48	**NLP algorithms**	**2**
49	**Reminder section**	**2, 4**
50	**Upcoming deadlines**	**3, 5**
51	**chatbot**	**2, 4, 7**
52	**answer**	**3**
53	**suggestion**	**3**
54	**classroom**	**2, 5, 6**
55	**search**	**3**
56	**inquiry**	**3**
57	**Search bar**	**2, 4, 7**
58	**Ongoing meeting**	**2, 3**
59	**attendance**	**2, 5, 7**
60	**submission**	**3**

61	**quiz**	**2**
62	**e-mail**	**1**
63	**password**	**2**
64	**Zoom**	**1**

Potential to be classes

- Teacher
- Student
- Account
- Profile
- Admin
- Email
- Course
- Notice
- Database
- Materials
- Exam
- Assignment
- Google Form
- Result
- Meeting
- Zoom
- Chatbot
- Classroom
- Search bar
- Attendance

Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria.

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

A candidate class generally becomes a class when it fulfills around three characteristics.

Serial No.	Noun	Selection Criteria
1	**Teacher**	**1, 2, 3, 4, 5, 6**
2	**Student**	**1, 2, 3, 4, 5, 6**
3	**Account**	**1, 2, 3, 4, 5**
4	**Profile**	**1, 2, 3, 4, 5**
5	**Admin**	**2, 3, 4, 5, 6**
6	**Email**	**6**
7	**Course**	**1, 2, 3, 4, 5, 6**
8	**Notice**	**2, 4, 5**
9	**Database**	**1, 2, 3, 4, 5**
10	**Materials**	**1, 3**
11	**Exam**	**1, 2, 3, 4, 5**
12	**Assignment**	**1, 2, 3, 4, 5**

13	**Google Form**	**6**
14	**Result**	**1, 2, 3, 4, 5**
15	**Meeting**	**1**
16	**Zoom**	**6**
17	**Chatbot**	**1, 2, 3, 4, 5**
18	**Classroom**	**1, 2, 3, 4, 5**
19	**Search bar**	**3, 4, 5**
20	**Attendance**	**1**

Selected Classes

- Teacher
- Student
- Account
- Profile
- Admin
- Course
- Result
- Database
- Notice
- Exam
- Assignment
- Chatbot
- Zoom
- Google Form
- Email
- Search bar

Analysis

The classes **Profile** and **Account** do almost the same work. So, we make only **Account** a class. Similarly, **Exam** and **Assignment** classes have almost the same functionality and attributes. Therefore, we have made a single class **Exam_Assignments**. All the external entities share some common features and integrating them into the system may require different procedures. So, we can make a parent class **External_Modules** and we can define the integrating methods Abstract so that we can implement them whenever we inherit the parent class.

Final Selected Classes

- Teacher
- Student
- Account
- Admin
- Course
- Result
- Notice
- Exams_Assignments
- Inquiries
- Database
- External_Modules

Attribute and Method Identification

Teacher	
Attribute	**Method**
- teacherID - teacherName - designation - contactInfo - assignedCourses - assignedClassHours - publications	+ getStudentID() + getTeacherName() + assignCourses() + getCourses() + sharedCourseManagement() + setClassHours() + getContactInfo()

Student	
Attribute	**Method**
- studentID - studentName - batch - roll - registrationNo - enrolledCourses - creditHours -CGPA - email - contactInfo - attendance	+ getStudentID() - setID() + getContactInfo() + getAcademicInfo() + getCourses() + setCourses() + setBatch() + calculateCreditHours() + getCGPA() + calculateAttendance()

Account	
Attribute	**Method**
- accountID - userName - password - accountType - accountStatus	+ signUp() + login() - accountRecovery() + accountRemoval() + getAccountType() + forgetPassword()

Admin	
Attribute	**Method**
- adminID - email - password + academicInfo + academicEvents	- setAdminInfo() + verifyStudentAccount() + createTeacherAccount() + generateRoutine() + assignCourses() + manageNoticeBoard() + editBatch() + editRoutine() + editAcademicInfo()

Course	
Attribute	**Method**
+ courseCode + courseName - courseCredit - courseOutline - courseMaterials - assignedTeachers - onlineClassMeetingLinks - courseAttendanceList - courseMarksList - courseStreamData	+ assignCourseTeachers() + setCourseOutline() + uploadCourseMaterials() + downloadCourseMaterials() + organiseMaterials() + courseStreamManagement() + attendanceCountFromMeeting() + courseStreamManagement()

Result	
Attribute	**Method**
- courseOutline - marks - grade - previousMarks - previousGrades	+getMarks() + calculateCGPA() + integrateCourseOutlineWeights() + createPerformanceGraph() + calculateResult() + getCGPA() + updateResult() + viewPerformanceStats() + getGrades()

Notice	
Attribute	**Method**
+ academicInfo + academicEvents + govtHolidays - deadlines - paymentActivities	+ updateNoticeBoard() + sortNotices() + viewNoticeBoard() + categorizeNotices()

Exams_Assignments	
Attribute	**Method**
- examType - assignmentType - submissionDeadline - duration - evaluationType - marks - questions - sampleSolutions	+ getDeadline() + getExamDetail() + getAssignmentDetail() + createExamAssignment() + uploadQuestion() + uploadSampleSolution() + createMCQExam() + createWrittenExamAssignment() + manualEvaluation() + convertHandwritingToTypedCopies() + machineEvaluation() + setMarks() + getMarks()

Inquiries	
Attribute	**Method**
- inquiryType - inquiryMessage - trainData - responseDataSet	+ parseInquiry() + generateResponse() + generateSuggestion() + search()

Database	
Attribute	**Method**
- studentData - teacherData - courseData - examAssignmentData - resultData - inquiryData - academicData	+ updateData() + parseData() + removeData() + addData() + computation() + compareData()

External_Modules	
Attribute	**Method**
- moduleLinks - moduleTime - moduleDate - notificationMessage - responseFormInfo	+ generateLink() + parseData() + sendNotification() + sendReminder() + handleFormInfo()

CRC Card

Table : CRC card for Teacher

Teacher	
Responsibilities	**Collaborator**
• Maintain teacher's profile • Create online meeting • Upload course materials and outlines • Upload exam papers, assignments • Evaluate exams and assignments	Student Admin Account Course Exam_Assignment Result External_Modules

Table : CRC card for Student

Student	
Responsibilities	**Collaborator**
• Maintain student profile • Join online meetings • Download course materials • Submit exams and assignments • Get online attendance • Get performance evaluation	Teacher Admin Account Course Exam_Assignment Result External_Modules

Table : CRC card for Account

Account	
Responsibilities	**Collaborator**
• Sign up users • Create accounts according to user type • Login users • Removal of accounts • Recover passwords	Admin Teacher Student

Table : CRC card for Admin

Admin	
Responsibilities	**Collaborator**
• Verify student account • Create teacher's profile • Generate routine • Assign courses to students and teachers • Manage Notice Board • Control database	Student Teacher Account Course Database Notice

Table : CRC card for Course

Course	
Responsibilities	**Collaborator**
• Manage course stream • Handle online meeting activities • Calculate course attendance from online meeting • Course materials management • Has exams, assignments • Update course marks according to course outline	Admin Teacher Student Exam_Assignments Result External_Modules

Table : CRC card for Result

Result	
Responsibilities	**Collaborator**
• Get marks from courses' exams and assignments • Calculate grades and CGPA • Generate performance stats • Create performance graphs • Edit result	Teacher Student Course Exam_Assignments

Table : CRC card for Notice

Notice	
Responsibilities	**Collaborator**
• Update notice of academic events and information • Remove notice • Sort and categorize notices according to priorities	Admin Teacher Student

Table : CRC card for Exam_Assignments

Exam_Assignments	
Responsibilities	**Collaborator**
• Upload question paper and sample solution • Create a submission deadline • Submission of solution script • Manual Evaluation • Machine Evaluation • Evaluation marks upload	Teacher Student Course External_Modules Result

Table : CRC card for Inquiries

Inquiries	
Responsibilities	**Collaborator**
• Take inquiry message • Provide answer to the inquiry • Provide suggestions • Search items	Teacher Student

Table : CRC card for Database

Database	
Responsibilities	**Collaborator**
• Store data of all the system • Add, delete, update data • Parse data into specific formats for further activities • Use data for computation, comparison and analysis	Admin Teacher Student Course Exam_Assignments Result External_Modules

Table : CRC card for External_Modules

External_Modules	
Responsibilities	**Collaborator**
• Generate meeting links • Send notifications along with deadlines • Handle user's response forms	Teacher, Student Exam_Assignments Course Result

Class Card

After identifying our final classes we have generated the following class cards-

Class : Teacher	
Attribute	**Method**
- teacherID - teacherName - designation - contactInfo - assignedCourses - assignedClassHours - publications	+ getStudentID() + getTeacherName() + assignCourses() + getCourses() + sharedCourseManagement() + setClassHours() + getContactInfo()
Responsibilities	**Collaborator**
• Maintain teacher's profile • Create online meeting • Upload course materials and outlines • Upload exam papers, assignments • Evaluate exams and assignments	Student Admin Account Course Exam_Assignment Result External_Modules

Class : Student	
Attribute	**Method**
- studentID - studentName - batch - roll - registrationNo - enrolledCourses - creditHours - CGPA - email - contactInfo - attendance - result	+ getStudentID() - setID() + getContactInfo() + getAcademicInfo() + getCourses() + setCourses() + setBatch() + calculateCreditHours() + getCGPA() + calculateAttendance()
Responsibilities	**Collaborator**
• Maintain student profile • Join online meetings • Download course materials • Submit exams and assignments • Get online attendance • Get performance evaluation	Teacher Admin Account Course Exam_Assignment Result External_Modules

Class : Account	
Attribute	**Method**
- accountID - userName - password - accountType - accountStatus	+ signUp() + login() - accountRecovery() + accountRemoval() + getAccountType() - forgetPassword()
Responsibilities	**Collaborator**
• Sign up users • Create accounts according to user type • Login users • Removal of accounts • Recover passwords	Admin Teacher Student

Class : Admin	
Attribute	**Method**
- adminID - email - password + academicInfo + academicEvents	- setAdminInfo() - verifyStudentAccount() - createTeacherAccount() - generateRoutine() - assignCourses() - manageNoticeBoard() - editBatch()

	- editRoutine() - editAcademicInfo()
Responsibilities	**Collaborator**
• Verify student account • Create teacher's profile • Generate routine • Assign courses to students and teachers • Manage Notice Board • Control database	Student Teacher Account Course Database Notice

Class : Course	
Attribute	**Method**
+ courseCode + courseName - courseCredit - courseOutline - courseMaterials - assignedTeachers - onlineClassMeetingLinks - courseAttendanceList - courseMarksList - courseStreamData	- assignCourseTeachers() - setCourseOutline() - uploadCourseMaterials() + downloadCourseMaterials() - organiseMaterials() - courseStreamManagement() - attendanceCountFromMeeting() - courseStreamManagement()

Responsibilities	Collaborator
• Manage course stream • Handle online meeting activities • Calculate course attendance from online meeting • Course materials management • Has exams, assignments • Update marks using outline •	Admin Teacher Student Exam_Assignments Result External_Modules

Class : Result	
Attribute	**Method**
- courseOutline - marks - grade - previousMarks - previousGrades	- getMarks() - calculateCGPA() - integrateCourseOutlineWeights() - createPerformanceGraph() - calculateResult() - getCGPA() - updateResult() - viewPerformanceStats() - getGrades()
Responsibilities	**Collaborator**
• Get marks from courses' exams and assignments • Calculate grades and CGPA • Generate performance stats • Create performance graphs	Teacher Student Course Exam_Assignments

• Edit result	

Class : Notice	
Attribute	**Method**
+ academicInfo + academicEvents + govtHolidays - deadlines - paymentActivities	- updateNoticeBoard() - sortNotices() + viewNoticeBoard() - categorizeNotices()
Responsibilities	**Collaborator**
• Update notice of academic events and information • Remove notice • Sort and categorize notices according to priorities	Admin Teacher Student

Class : Exam_Assignments	
Attribute	**Method**
- examType - assignmentType - submissionDeadline - duration - evaluationType - marks - questions	+ getDeadline() + getExamDetail() + getAssignmentDetail() - createExamAssignment() - uploadQuestion() - uploadSampleSolution() - createMCQExam()

- sampleSolutions	-createWrittenExamAssignment() - manualEvaluation() - convertHandwritingToTypedCopies() - machineEvaluation() - setMarks() + getMarks()
Responsibilities	**Collaborator**
• Upload question paper and sample solution • Create a submission deadline • Submission of solution script • Manual Evaluation • Machine Evaluation • Evaluation marks upload	Teacher Student Course External_Modules Result

Class : Inquiries	
Attribute	**Method**
- inquiryType - inquiryMessage - trainData - responseDataSet	- parseInquiry() - generateResponse() - generateSuggestion() - search()
Responsibilities	**Collaborator**
• Take inquiry message • Provide answer to the inquiry	Teacher Student

• Provide suggestions • Search items	

Class : Database	
Attribute	**Method**
- studentData - teacherData - courseData - examAssignmentData - resultData - inquiryData - academicData	- updateData() - parseData() - removeData() - addData() - computation() - compareData()
Responsibilities	**Collaborator**
• Store data of all the system • Add, delete, update data • Parse data into specific formats for further activities • Use data for computation, comparison and analysis	Admin Teacher Student Course Exam_Assignments Result External_Modules

Class : External_Modules	
Attribute	**Method**
- moduleLinks - moduleTime - moduleDate - notificationMessage - responseFormInfo	- generateLink() - parseData() - sendNotification() - sendReminder() - handleFormInfo()
Responsibilities	**Collaborator**
• Generate meeting links • Send notifications along with deadlines • Handle user's response forms	Teacher, Student Exam_Assignments Course Result

CRC Diagram

ID : 1

Name: Teacher

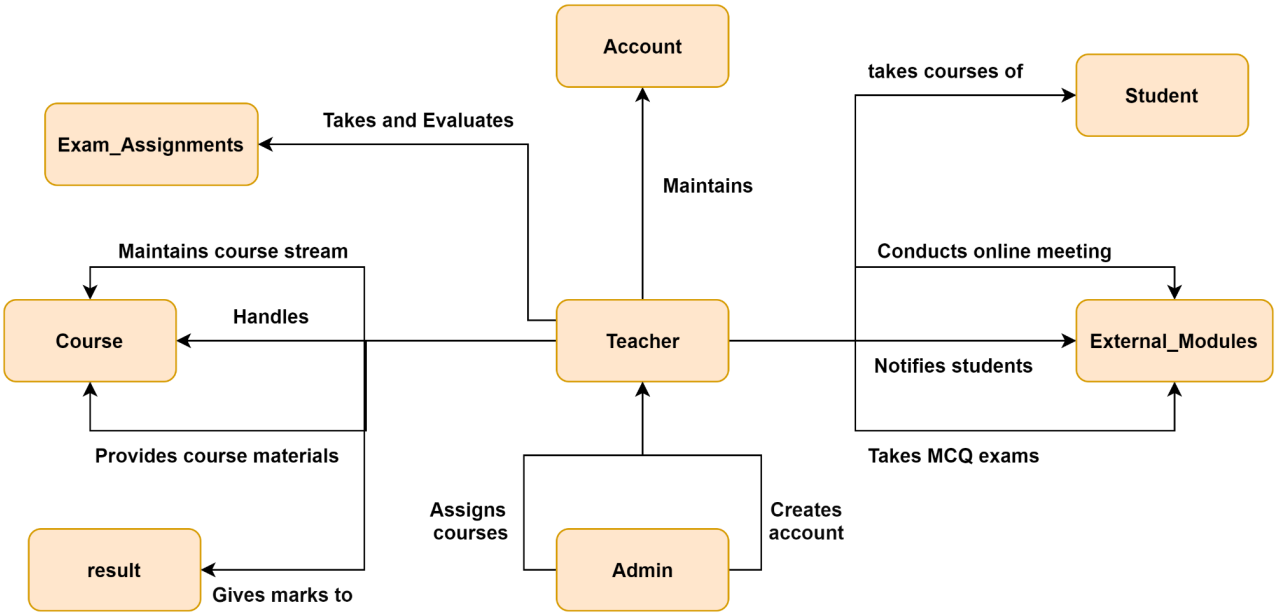


Figure: CRC Diagram for Teacher

ID : 2

Name: Student

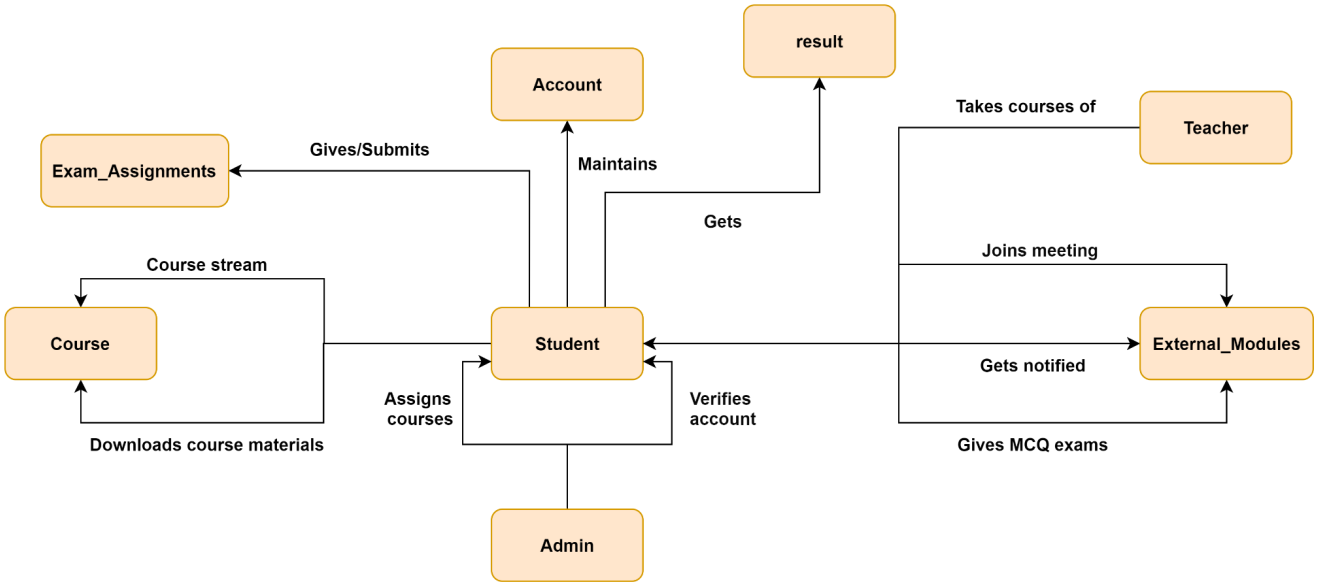


Figure: CRC Diagram for Student

ID : 3

Name: Account

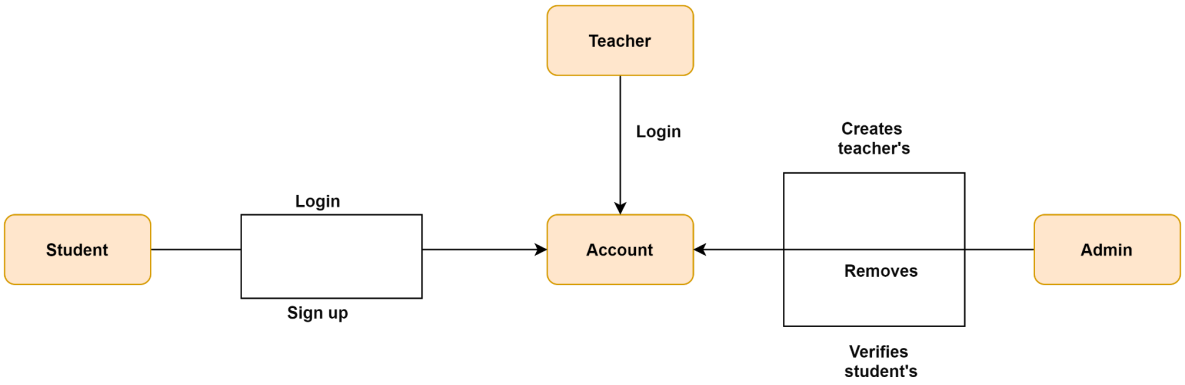


Figure: CRC Diagram for Account

ID : 4

Name: Admin

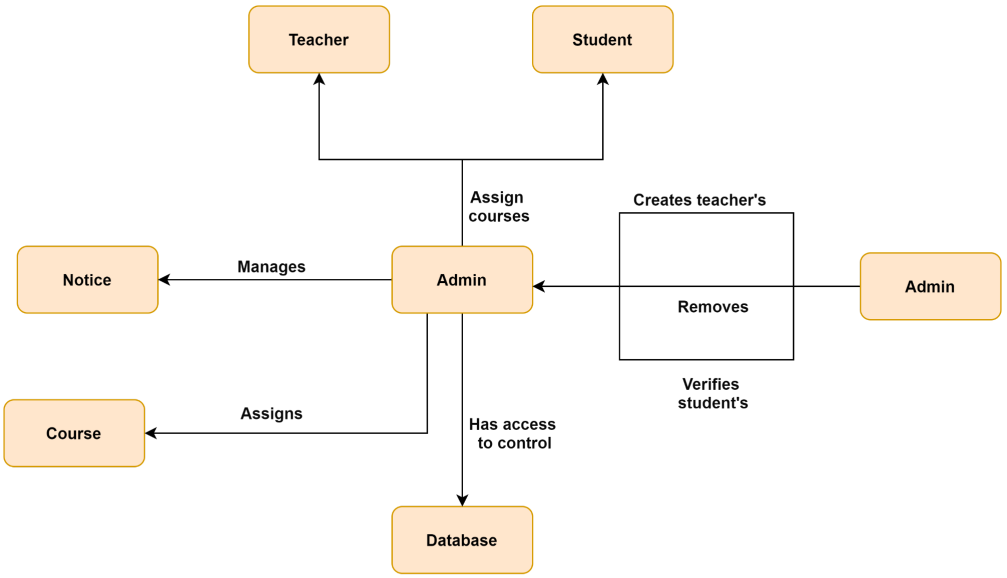


Figure: CRC Diagram for Admin

ID : 5

Name: Course

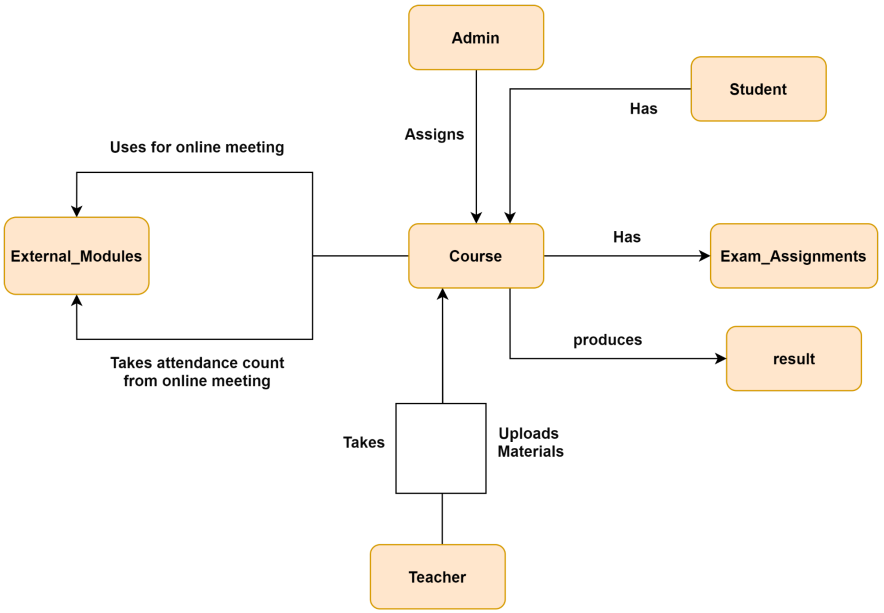


Figure: CRC Diagram for Course

ID : 6

Name: Result

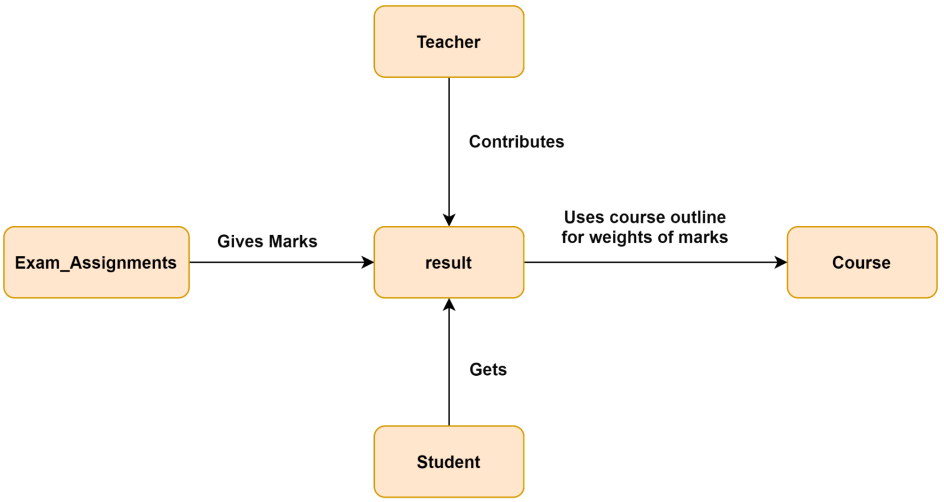


Figure: CRC Diagram for Result

ID : 7

Name: Notice

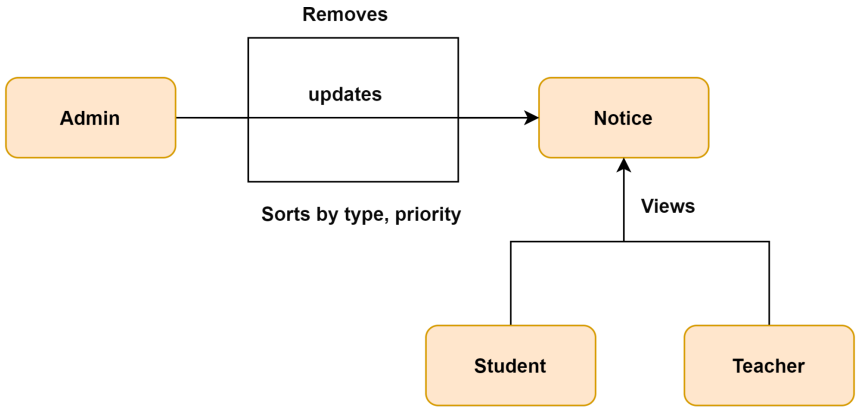


Figure: CRC Diagram for Notice

ID : 8

Name: Exam_Assignments

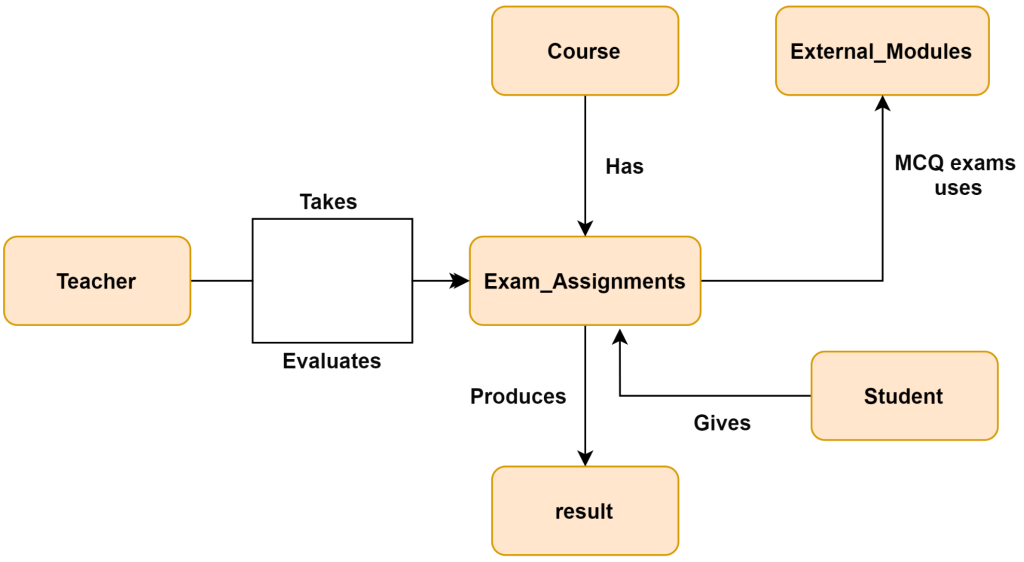


Figure: CRC Diagram for Exam_Assignments

ID : 9

Name: Inquiries

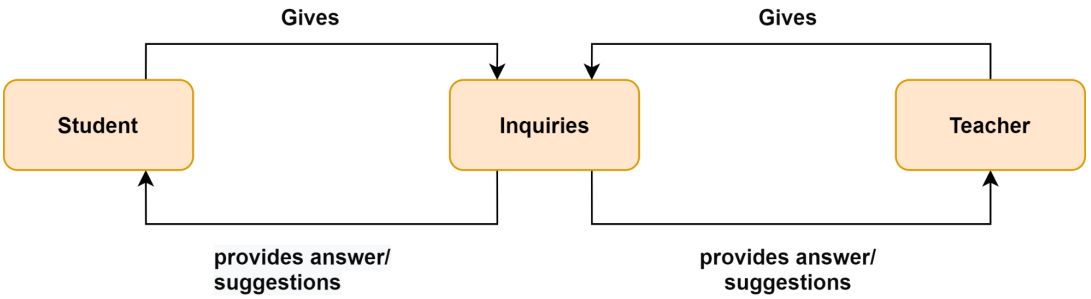


Figure: CRC Diagram for Inquiries

ID : 10

Name: Database

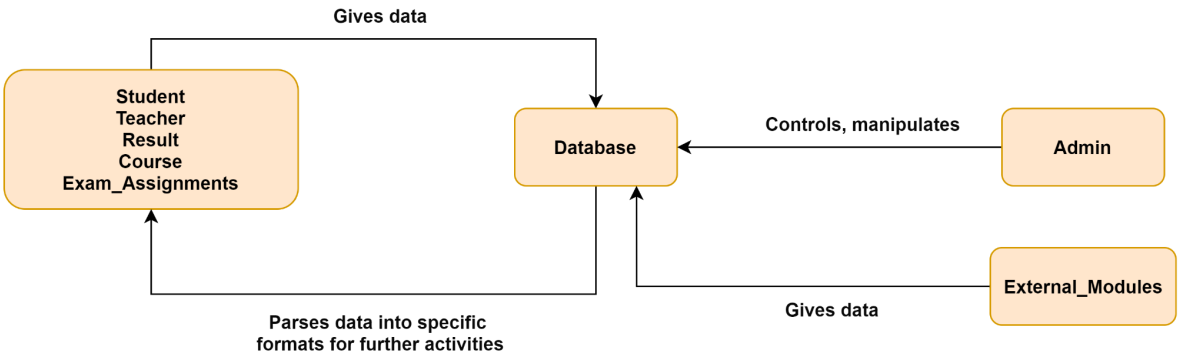


Figure: CRC Diagram for Database

ID : 11

Name: External_Modules

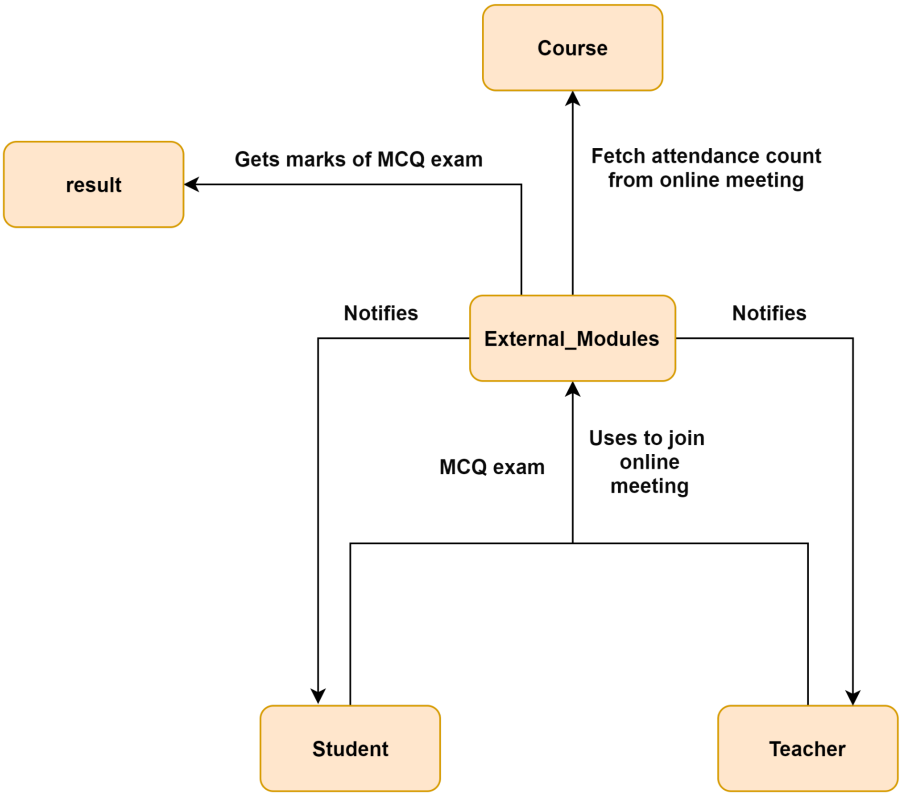


Figure: CRC Diagram for External_Modules

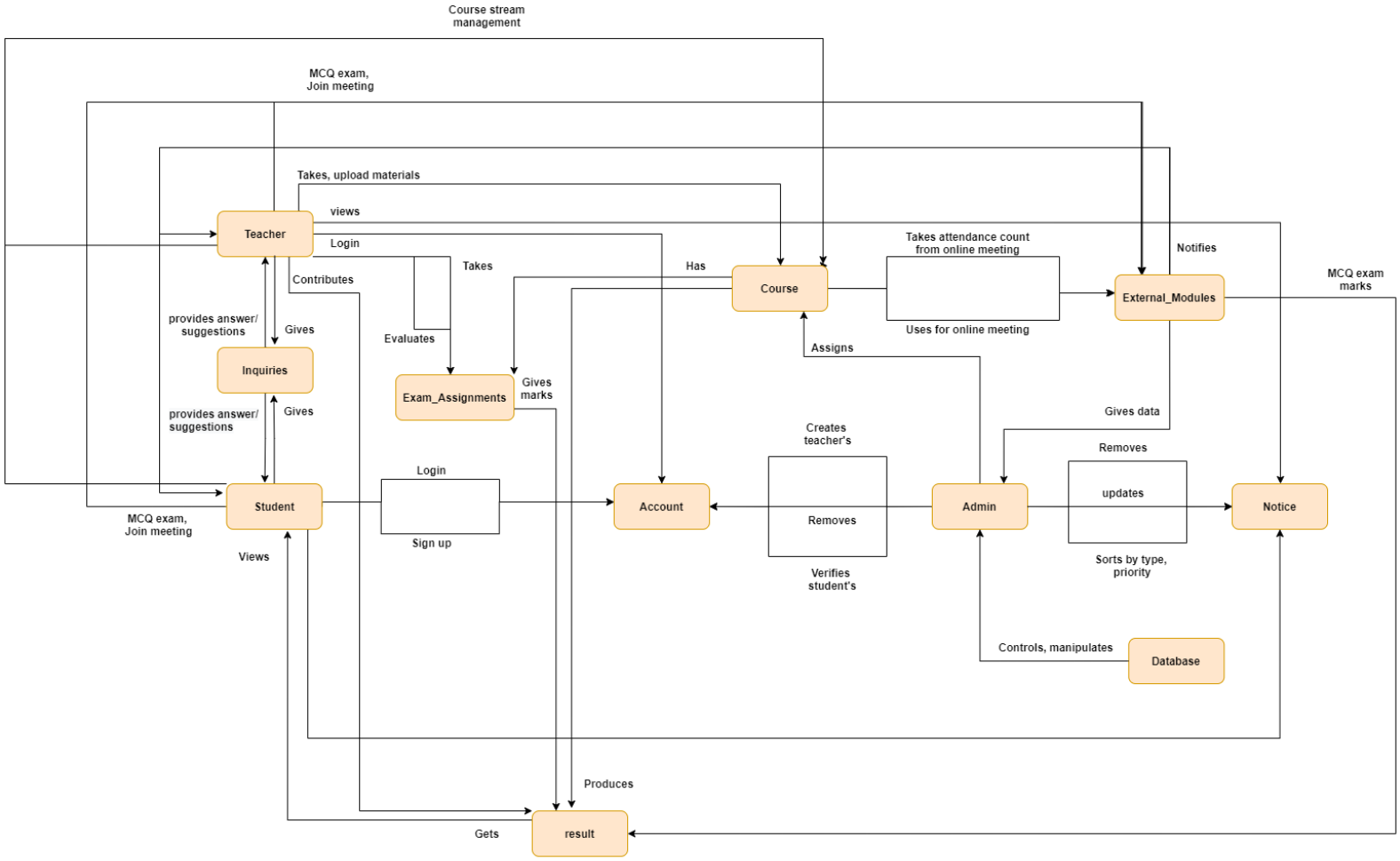


Figure: CRC Diagram for all classes

Behavior Modeling: E-shcool

The Behavior Modeling indicates how the system will behave to external events or stimuli. It is represented as a function of time and event,

It describes interactions between objects. It shows how individual objects collaborate to achieve the behavior of the system as a whole. In UML behavior of a system is shown with the help of use case diagram, sequence diagram and activity diagram.

To create behavioral model following things can be considered-

- Evaluation of all use-cases to fully understand the sequence of interaction within the system.
- Identification of events that drive the interaction sequence and understand how these events relate to specific classes.
- Creating sequence for each use case.
- Building a state diagram for the system.
- Reviewing the behavioral model to verify accuracy and consistency.
- The State Transition diagram provides an intra-class view of the system.
- Construct a Sequence Diagram.
- The Sequence Diagram depicts the inter-class relationship in the chronological order.

Event Table

SL no	Event	State name	Initiator	Collaborator
1	Create Student Account	Create_Stu_Acc	Student	Database
2	Log In	Login	Admin, Student, Teacher	Account, Database
3	Verify Account	Verify_Acc	Admin	Database, Course
4	Create Teacher Account	Create_Tea_Acc	Admin	Database, Teacher
5	Delete Account	Delete_Account	Admin	Database
6	Initiate Course	Initiate_Course	Admin	Course, Teacher, Student
7	Create Routine	Create_Routine	Admin	Teacher, Student, Database
8	Post Notice	Post_Notice	Admin	Notice, Teacher, Student, Database
9	Create Post	Create_Post	Teacher, Student	Course
10	Update/Delete Post	Update/Delete_Post	Teacher, Student	Course
11	Post Comment	Post_Comment	Teacher, Student	Course
12	Create Meeting	Create_Meeting	Teacher	Student, External
13	Send Notification	Send_Notification	External	External, Student
14	Join Meeting	Join_Meeting	Student, Teacher	External, Courses, Database

15	Define Course Outline	Define_Outline	Teacher	Courses, Result, Database
16	Attendance Calculating	Attendance_Calc	External	Student, Result
17	Upload Materials	Upload_Materials	Teacher	Courses, Database
18	Delete Materials	Delete_Materials	Teacher	Courses, Database
19	Download Materials	Download_Materials	Student	Courses, Database
20	Take Exam	Take_Exam	Teacher	Course, Student, Exams_Assignments
21	Take Assignment	Take_Assignment	Teacher	Courses, Student, Exams_Assignments
22	Take MCQ Exam	Take_MCQ	Teacher	External, Student, Course
23	Upload Questions	Upload_Ques	Teacher	Course, Student, Database, Exams_Assignment
24	Upload Sample Answer	Upload_Ans	Teacher	Course, Exams_Assignments
25	Upload Answer Script	Upload_Script	Student	Exams_Assignments, Courses, Database
26	Set Deadlines	Update_Deadline	External, Exams_Assignments	Courses, Student

27	Evaluate Exam	Evaluate_Exam	Teacher	Courses, Exams_Assignments
28	Manual Checking	Manual_Check	Teacher	Marks, Course, Exams_Assignments
29	Machine Checking	Machine_Checking	Teacher	Exams_Assignments, Courses, Marks
30	Search	Search	Teacher, Student	Courses, Materials, Inquiries
31	View Activity	View_Activity	Student	Courses, Marks, Result
32	View Materials	View_Materials	Student	Course, Materials
33	Set Inquiry	Set_Inquiry	Teacher, Student	Courses, Materials, Inquiries
34	Get Response	Get_Response	Inquiries	Teacher, Student, External
35	Review Performances	Review_Performance	Teacher	Courses, Student
36	Post Comments	Post_Comments	Teacher	Course, Student
37	View Marks	View_Marks	Student	Marks
38	View Result	View_Result	Student	Result
39	View Performance Graph	View_Graph	Student	Courses
40	Edit Entries	Edit_Entries	Admin	Database, Student, Teacher

State Transition Diagram

State Transition Diagram represents active states for each class of events (triggers). For this we identified all the events, their initiators and collaborators. In the State Transition Diagram the states are shown in boxed texts, and the transition is represented by arrows. It is also called State Chart or Graph. It is useful in identifying valid transitions.

In the state transition table all the states are listed on the left side, and the events are described on the top. Each cell in the table represents the state of the system after the event has occurred. It is also called the State Table. It is useful in identifying invalid transitions.

Notation

For those not familiar with the notation used for state-transition diagrams, some explanation is in order.

State: A condition during the life of an object in which it satisfies some condition, performs some action, or waits for some event.

Event: An occurrence that may trigger a state transition. Event types include an explicit signal from outside the system, an invocation from inside the system, the passage of a designated period of time, or a designated condition becoming true.

Guard: A boolean expression which, if true, enables an event to cause a transition.

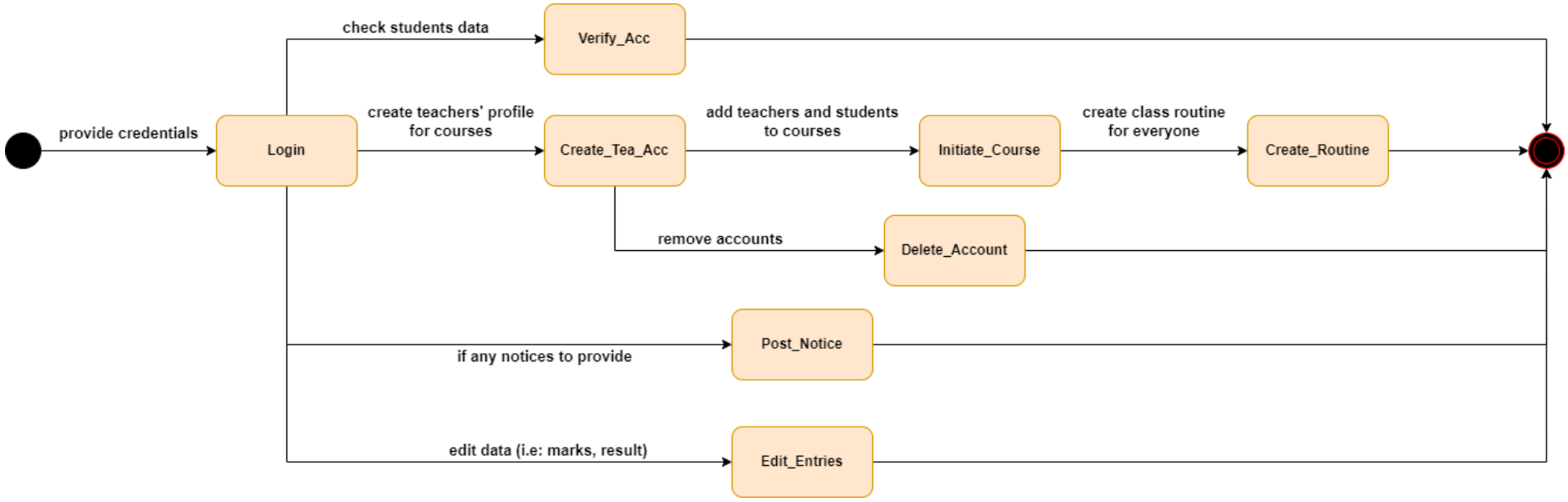
Transition: The change of state within an object.

Action: One or more actions taken by an object in response to a state change.

State Transition Diagram

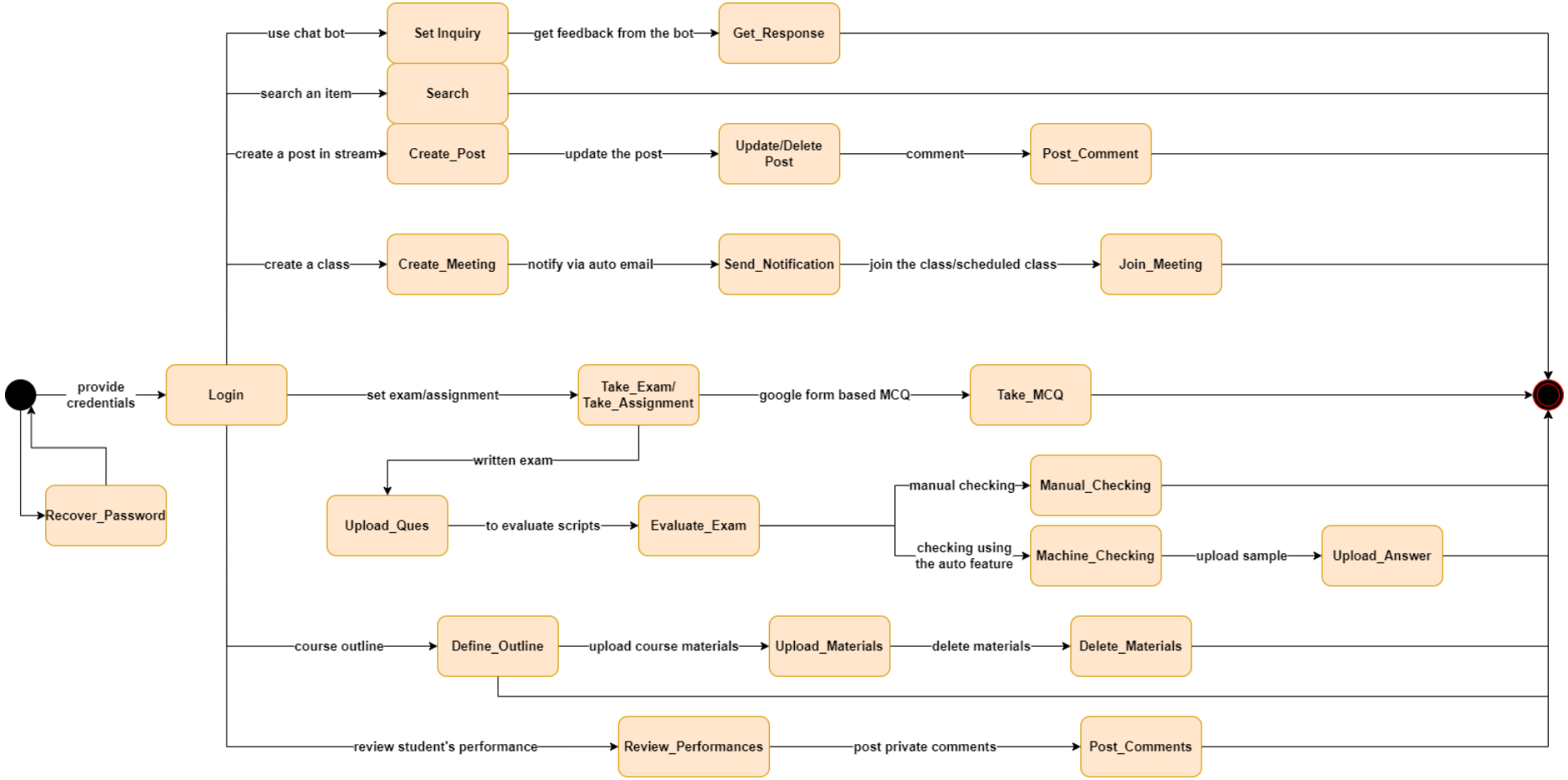
ID: 01

Name: Admin



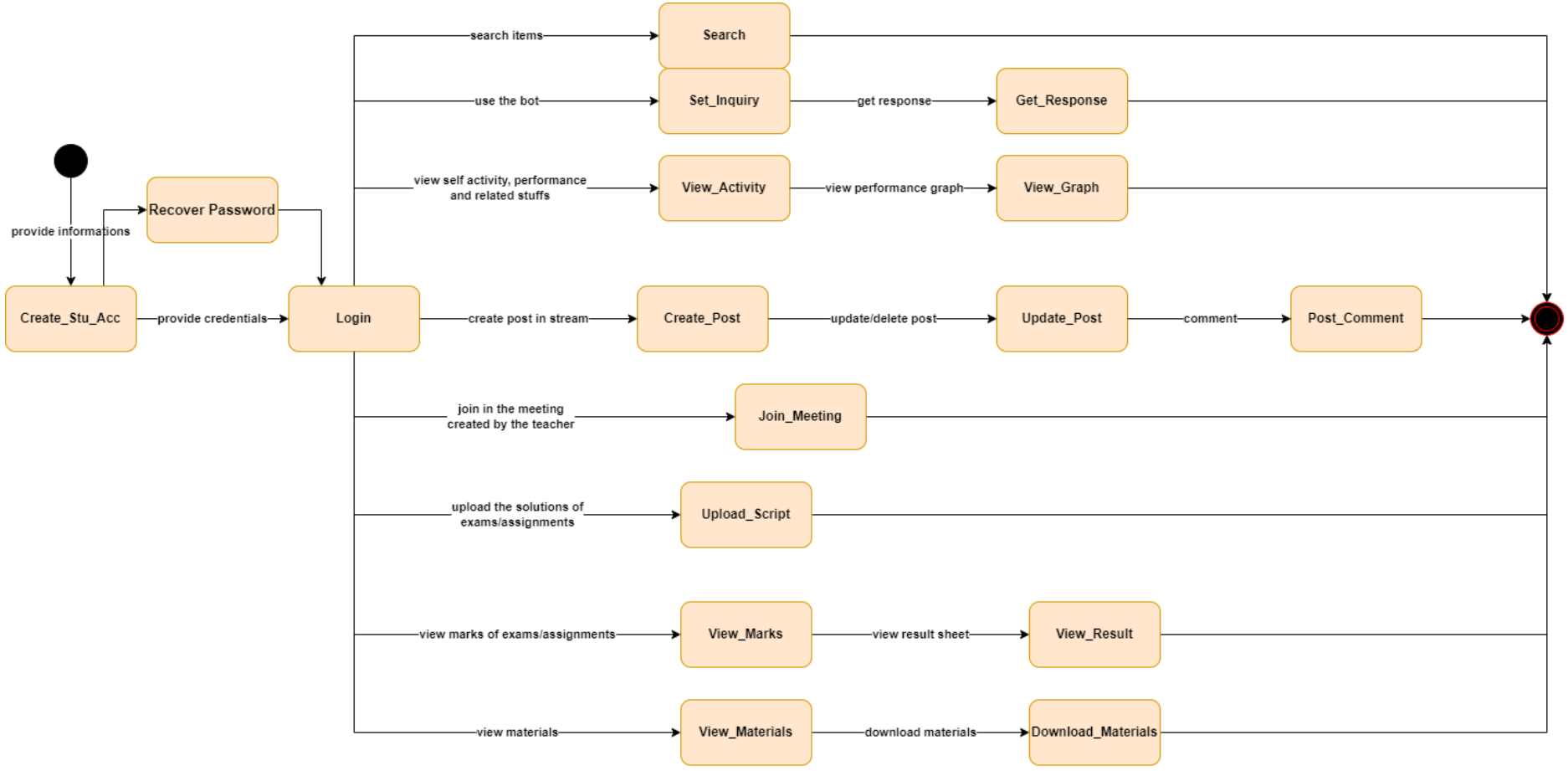
ID: 02

Name: Teacher

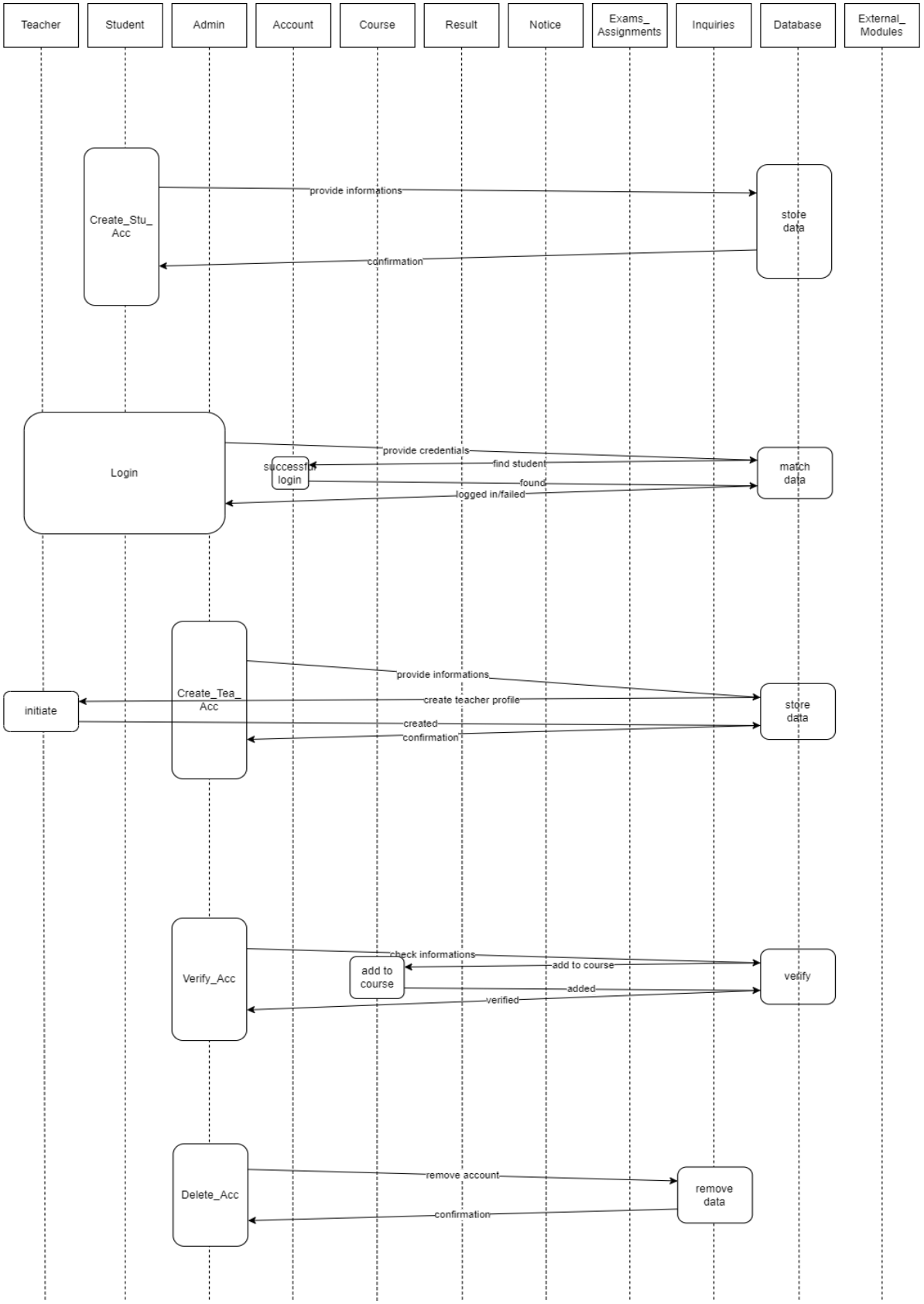


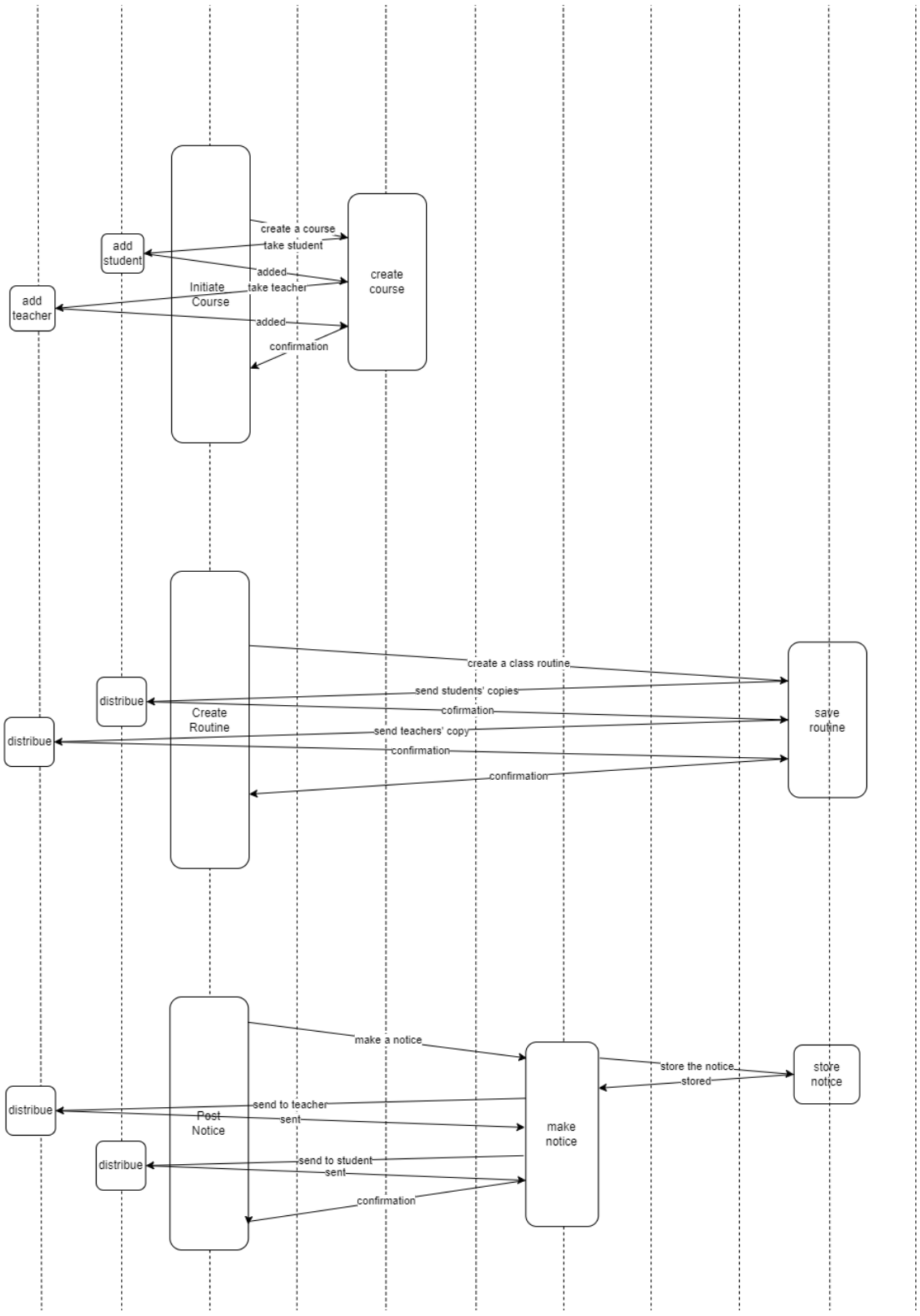
ID: 03

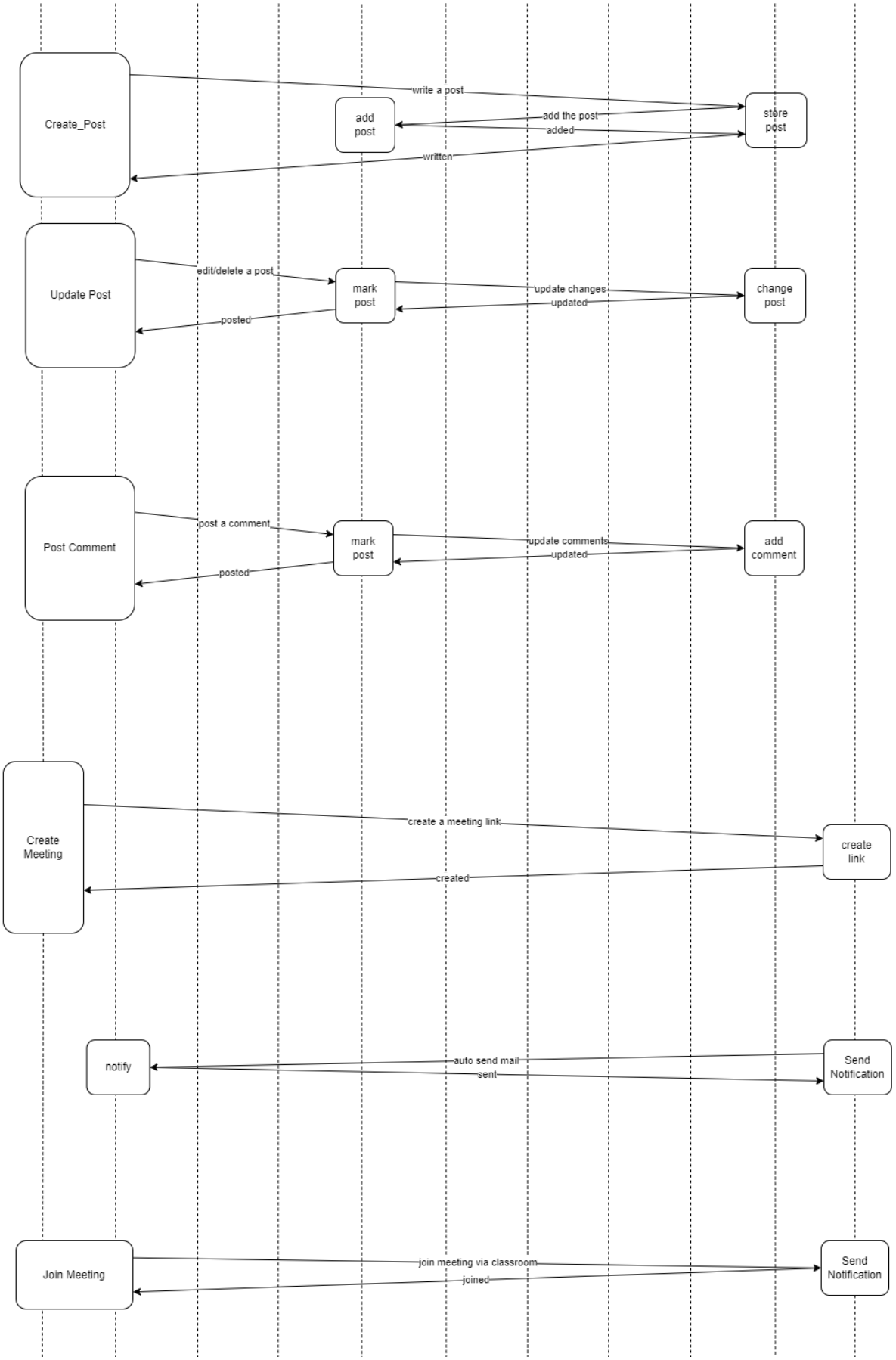
Name: Student

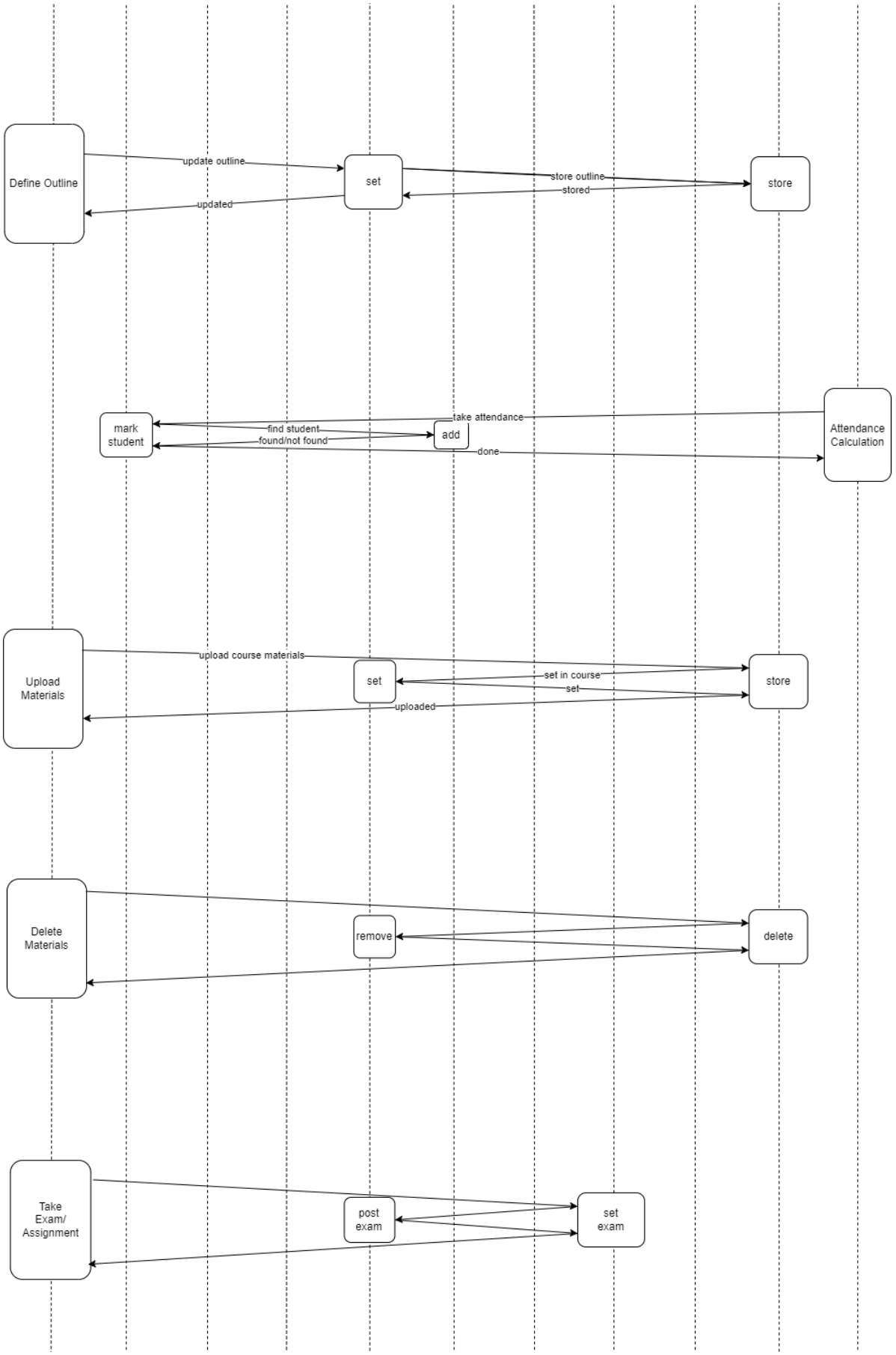


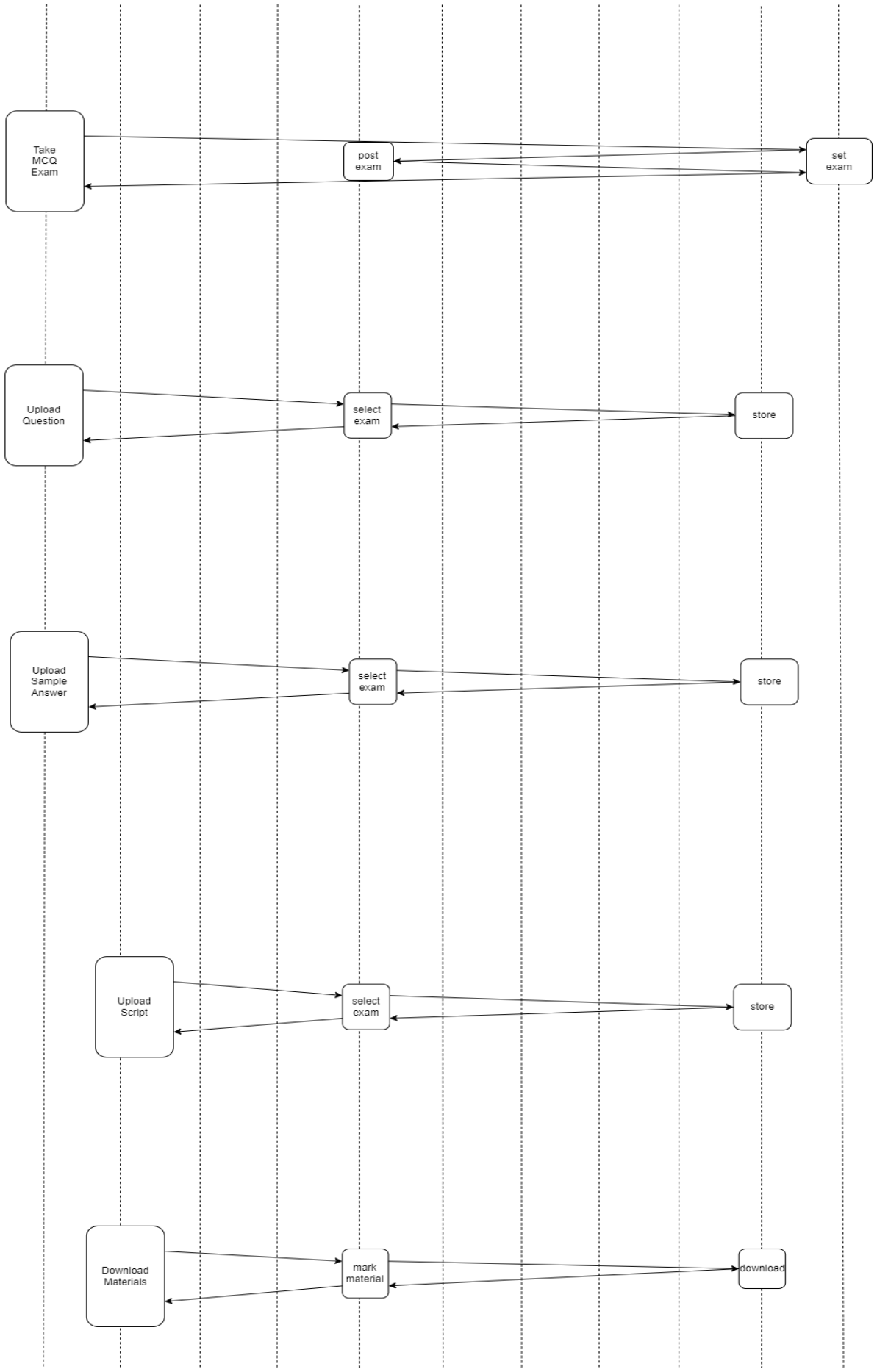
Sequence Diagram

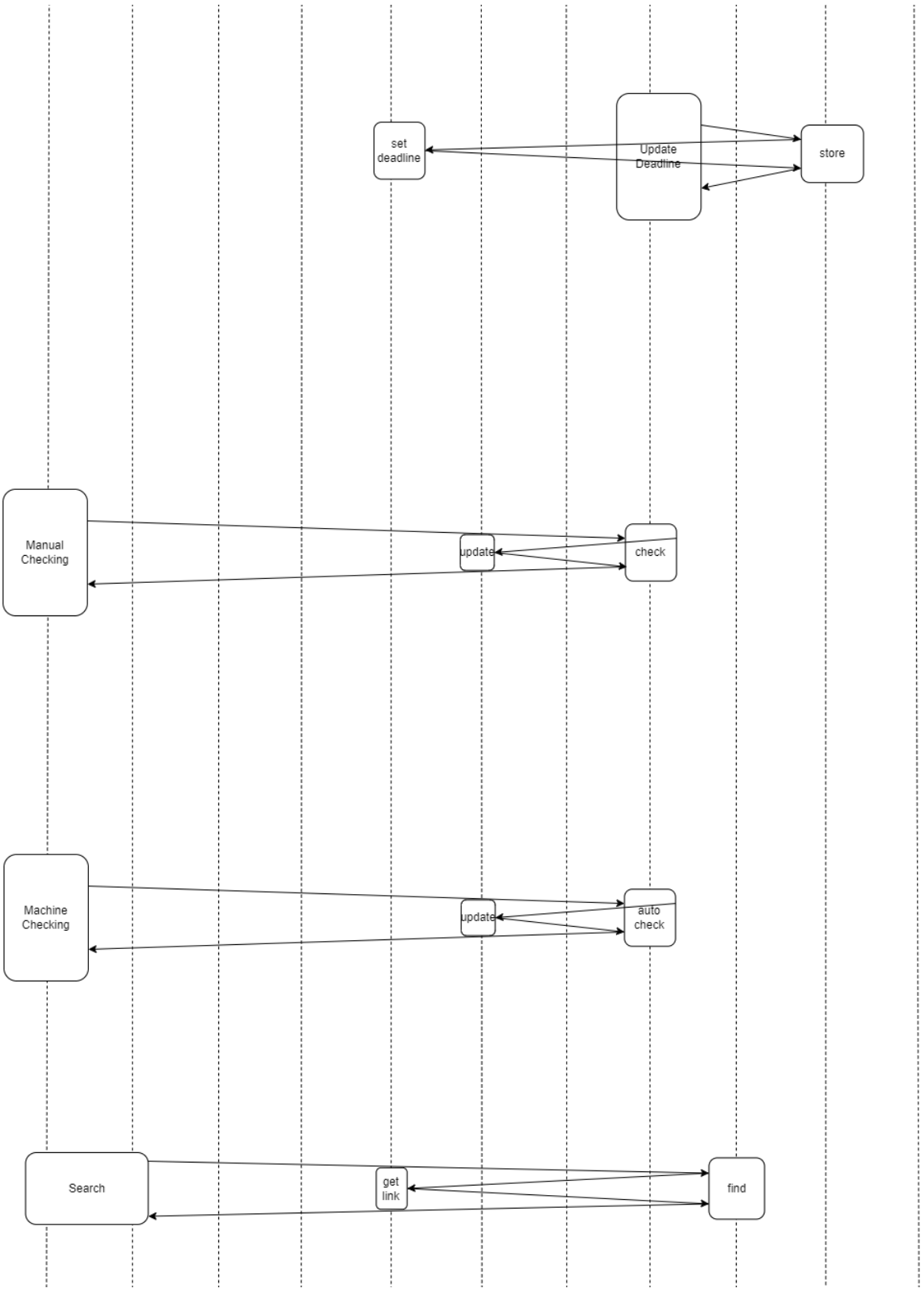


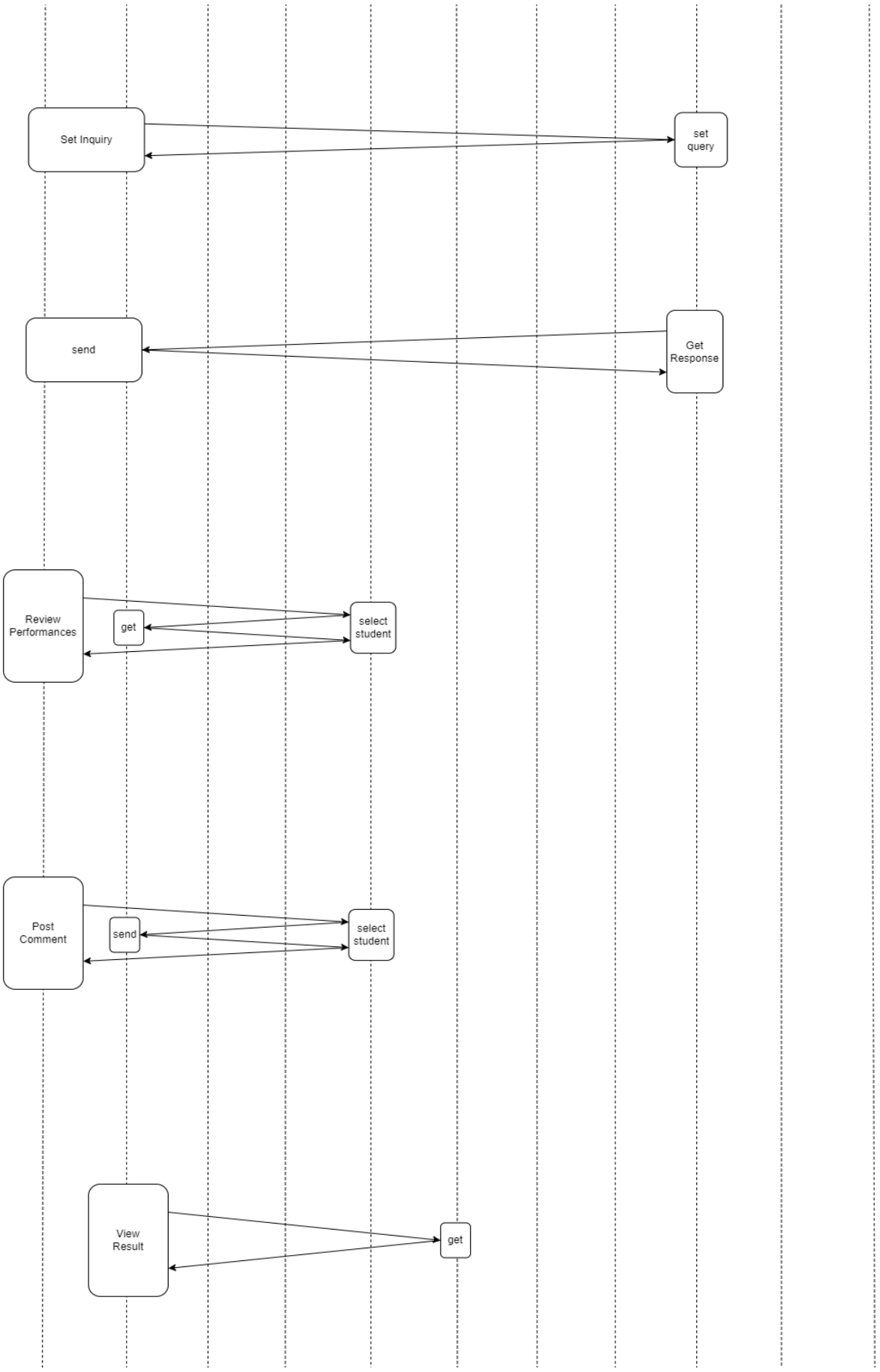

add student
add teacher
Initiate Course
create course
create a course
take student
added
take teacher
added
confirmation
distribue
distribue
Create Routine
save routine
create a class routine
send students' copies
cofirmation
send teachers' copy
confirmation
confirmation
distribue
distribue
Post Notice
make notice
store notice
make a notice
store the notice
stored
send to teacher
sent
send to student
sent
confirmation

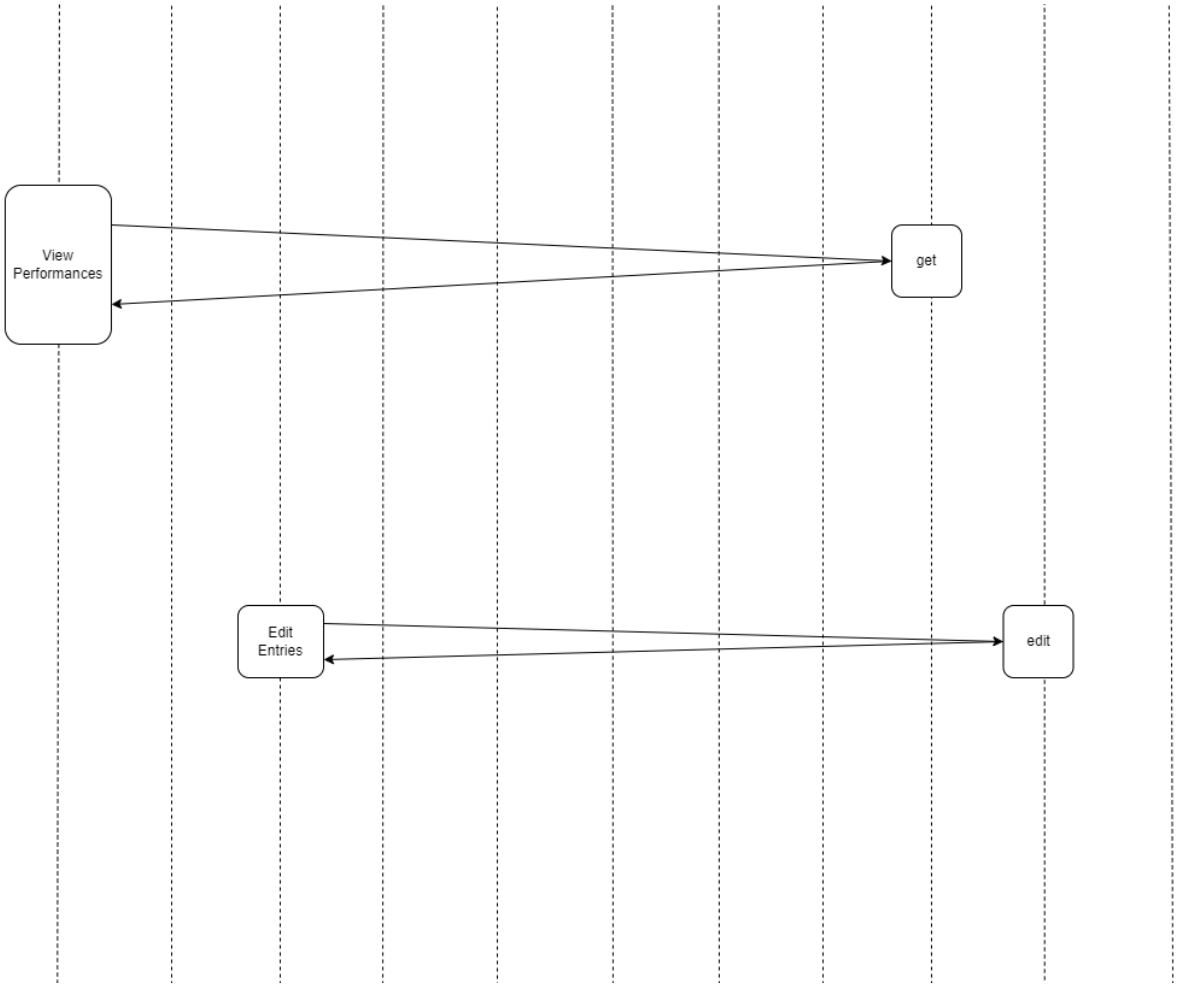

Create_Post
write a post
add post
add the post
added
store post
written
Update Post
edit/delete a post
mark post
update changes
updated
change post
posted
Post Comment
post a comment
mark post
update comments
updated
add comment
posted
Create Meeting
create a meeting link
create link
created
notify
auto send mail
sent
Send Notification
Join Meeting
join meeting via classroom
joined
Send Notification


Define Outline
update outline
set
store outline
store
stored
updated
mark student
take attendance
find student
add
found/not found
done
Attendance Calculation
Upload Materials
upload course materials
set
set in course
store
set
uploaded
Delete Materials
remove
delete
Take Exam/ Assignment
post exam
set exam


Take MCQ Exam
post exam
set exam
Upload Question
select exam
store
Upload Sample Answer
select exam
store
Upload Script
select exam
store
Download Materials
mark material
download


set deadline
Update Deadline
store
Manual Checking
update
check
Machine Checking
update
auto check
Search
get link
find


Set Inquiry
set query
send
Get Response
Review Performances
get
select student
Post Comment
send
select student
View Result
get


View Performances
get
Edit Entries
edit