Contents

1 INTRODUCTION

This chapter is a part of our software requirement specification for the project “Course Management System”. In this chapter, we have focused on the intended audience for our project “Course Management System”

1.1 Purpose

This document describes the Software Requirement Analysis of course Management System. It contains all the functional, non-functional, supporting requirements and expected requirements. At the same time it establishes a requirement’s baseline for the development of the project. 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.

1.2 INTENDED AUDIENCE

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

- The work of the developer team that means the product which they have created can easily be verified by the customer whether it is acceptable to them or not by using this SRS document of the Course Management System.

- Project Managers will be able to plan milestones, delivery date and at the same time ensure whether the developing team is on track or not during the development of the system by using this SRS document.

- 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.

- For developing the system's functionality, this SRS document will be used by the developers. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customers' documented requirements.

- The testers will use this SRS in various phases of testing. They 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.

1.3 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

2 COURSE MANAGEMENT SYSTEM INCEPTION

The inception part of our SRS is briefly discussed in this part :

2.1 INTRODUCTION

Course management system is a Software which will help teachers of all sectors such as school, college, university, training program, coaching to maintain their course in an efficient way. This document is a sample demand list of what the stakeholders want for their anticipated software. Like the various features and interfaces of the software which will help them to ease their work while taking a course.

2.2 INCEPTION OF A COURSE

At first, we have entered into the inception stage. This stage includes, how our project will start and what are the scopes and limitations. The main goal of this phase is to identify the requirements & demands and then establish some sort of mutual understanding between the software team and the customers. They are the intended users for our software. In order to make this phase effective we took the following steps:

- Identifying the client of our project
- Icebreaking
- Identifying the stakeholders of a course
- Identifying the multiple viewpoints of the stakeholders

2.2.1 Identifying the client of our project

At first, we have identified the location from where we will start our expedition. Normally teachers and students will act as a stakeholder. But there are other things related with this as well. So we have to go through a systematic approach in order to identify all stakeholders. But over all students are teachers are expected to be our clients. We have analyzed our requirements with the consent of both of them

2.2.2 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 acceptation of our proposal. We started this phase by talking with the students with context free languages. The students informally expressed their expectations. We also carried out informal meeting with teachers from different institutions. The behavior of both student and teacher was positive and both of them want this change in course management.

2.2.3 IDENTIFYING THE STAKEHOLDERS OF A COURSE

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 institutions that we have visited and teachers we have met indicates that a course has a few stakeholders only. Identification of the stakeholders were done from the information provided by the teachers from various departments of the University of Dhaka. The stakeholders of our system are given below:

- Student
- Teacher

2.2.4 IDENTIFYING THE MULTIPLE VIEWPOINTS OF THE STAKEHOLDER

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:

- Students View Point
    - Smartphone based system.
    - Keeping all class record at a single place.
    - Making result generating process automated.
    - Open platform for discussion.
    - Getting informed about any notice in an efficient way.
    - Maintaining privacy of marks for each students.
    - Submitting assignments online.
- Teacher’s View Point

- Simple Course Management.
- Easy user interface to interact.
- Online Attendance.
- Assignment submission.
- Online quiz taking system.
- Generating templates for script evaluation.
- Storing result against student's profile.
- Generating results automatically from the stored marks.
- Storing class contents.
- Generating improvement curves for students.

2.2.3 Conclusion

Our primary goal is to design a software which will make course management a lot easier and more efficient, not only for university lecturers or professors but also for the whole community of teachers . At the same time, where students can get what information they need or raise questions in any fact related to the course. For these reasons, the software will be designed in such a way that it won't be complicated for both the students and teachers who will use it. The software will be so simple that a teacher who does not have any idea about software, can be able to maintain it without any annoyance. Otherwise it will not meet its goal. That is to make the course management process easy and efficient. The software will be designed in such a way as it takes very little time to manage. To make this software project successful, collaboration with the stakeholders i.e. the students and teachers was a main priority that what they want, how the software will work, how it can be more effective for all.

3 ELICITATION OF CMS

So far we have discussed the Inception phase of our project. Now we need to focus on the Elicitation phase. So this chapter specifies the Elicitation phase.

3.1 INTRODUCTION

Elicitation is a part of requirements engineering that is. We have faced many difficulties, like understanding the problems, making questions for teachers, getting appointments from teachers in spite of their busy schedule, making them understand the power of automation. Despite not being easy to gather requirements within a very short time, we have overcome all problems in a systematic manner. We have done several meetings and finalized the requirements for our software.

3.2 ELICITING REQUIREMENTS

Elicitation phase is mainly combining the elements for problem solving, elaboration, negotiation and specification. Without the collaboration of the stakeholder eliciting would have been really hard. We have finished the following tasks for eliciting requirements-

- Collaborative Solution
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

3.2.1 COLLABORATIVE SOLUTION

We have met with many teachers in the Inception phase . These meetings created an indecisive state and we could identify what the real problem is. To solve this problem, we have met with the stakeholders several times and came up with the solution which eventually helped us to elicit the requirements.

3.2.2 QUALITY FUNCTION DEPLOYMENT

QFD, Quality Function Deployment, is a technique that translates the needs of the customer into technical requirements for software. QFD mainly translates 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 methodology concentrating on maximizing customer satisfaction from the software engineering process. So, we have followed this methodology to identify the requirements for the project. The requirements identified successfully by the QFD are given below

3.2.2.1 NORMAL REQUIREMENTS

The requirements which are normally stated by the customer in various meetings are the normal requirements of a system, In general sense the objectives and goals that are stated for a product or system during meetings with the customer are normal requirements. The presence of these requirements fulfills customers' satisfaction. The normal requirements for our project are given below

- Simple and user-friendly interface
- Smartphone based application
- Easily accessible for all
- Create course and invite students to join
- Online attendance
- Online quiz
- Keep all class records and store contents
- Assignment submission
- Storing marks against student
- Template design for script evaluation
- Automated result generation
- Privacy in result distribution
- Store log off all activities
- Generate student improvement curves

3.2.2.2 EXPECTED REQUIREMENTS

These requirements are intrinsic to the system and may be so elementary that the customer does not explicitly state them. But the absence of this requirements would result in dissatisfaction. The expected requirements are given below

- Creating profile for each student
- Forum for uploading files, post queries & search files
- Authentication
- Scalability
- Access control to teacher

3.2.2.3 EXCITING REQUIREMENTS

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying. These are the exciting features of our project

- Plagiarism check
- Notice board
- Course evaluation bar diagram

3.2.3 USAGE SCENARIO

Course Management system will be a digitalized system to manage a single course . The overall control management of the course will remain upon the course teacher or multiple course teachers. Student will join only and enjoy the features. The features that will be provided are mentioned below:

❖ Register

Initially both teachers and students will have to register. Students need to provide their

- Name
- Email address
- Class roll

- User name
- Password

Teachers will have to provide their

- Name
- Email address
- User name
- Password

The optional credentials for students will be

- Registration number
- Department
- Year(semester).

For teachers, optional credentials will be

- department name
- Designation.

Upon giving these information's account will be generated for student and teacher

❖ Create Course

Teachers will create their own course with course name, course schedule and outline. The teachers will invite the students to join the course sending a random code students email id. If a student provides mandatory information he or she will be added. Otherwise a request for providing necessary information will be sent again. Course will automatically be assigned with an id.

❖ Login & Recovery

Teacher & student can login using username & password. If someone forgets his/her password, it can be reset after a verification through Email/mobile number.

❖ Forum

Forum will be used for query or problem discussion. Both teachers and students are allowed to contribute here by adding comments and uploading files. Comments can be edited deleted and replied also.

❖ Maintain Course

In this part, there will be many subsystems for specific tasks which are given below:

- **Class Lecture Maintenance**

In this subsystem, documents and file related to each lecture can be uploaded and can be downloaded also. Both course teacher and students will be eligible to upload and download files.

- **Online Attendance**

The teacher will give the schedule of class and according to it, an automated code will be generated by the teacher for attendance. The teacher will fix a particular time and open a dash board. Students will have to provide the code. Those students who will provide the code within the fixed time will be counted as present.

- **Class record**

There will be a log book for the classes where all records will be kept.

❖ Students Profile

Students will have a profile based on his given data. Profile will have a profile id. The feature of student profile are:

- **Attendance Count**

When the student will provide a certain code his attendance will be counted against his profile. There will be a counter against everyone's profile which will keep this count.

- **Marks Of students**

The achieved marks by the students on different aspects such as attendance, midterm, lab, quiz , final will be stored against his/her profile.

- **Improvements curves**

This feature will help to see the improvement of a student where a curve will be generated based on the marks given to him. A teacher can view the curve of all students but the student can only see his own improvement curve.

- **Student Log**

Keeps the log of what files have student uploaded. A student will be able to view his own log and search the log as well. Teacher will be able to view and search the log of all students.

❖Upload

Student can upload files in various places. Every uploaded file will get a document id. The ways of upload are given below

- **General Upload**

Students and teachers can upload files in the forum . The student will provide the file and the file will be sent to the teacher for verification. If the teacher permits the file then

the file will be uploaded and the upload log will be kept against the student's profile. Teachers will provide their selected file and the file will be uploaded.

- **Assignment Upload**

Teacher will open a dash board and set a particular time in which the students will upload their particular file. Every assignment will automatically get an id. Teacher will be able to set the time of submission. If the students want to resubmit the assignment than his/her previous file will be replaced. their will an assignment log to keep the record about who has submitted and how many times. Teacher won't be able to submit any kind of file in his own assignment dash board

❖ Exam and Mark distribution

There will be several subsystems in this section, which are described below

- **Online quiz**

Teacher will set fixed time, question, marking and distribution and prepare a digital MCQ script. It will be checked by option making formula. Marks achieved by each students will be stored against their Profile. Each quiz will have an id

- **Midterm**

Midterm can be taken both with physical script or via online quiz. If taken physically given marks by the teacher will be stored against the profile.

- **Assignment evaluation**

Teacher will mark the assignments which have been submitted. And if there are any physical assignments; the teacher will give the marks of that assignment. These will be stored against the student Profile.

- **Plagiarism Checker**

Plagiarism of all the submitted assignment will be checked. If more than the percentage set by the teacher of the files match with each other then we will count it as copy. If copied files are discovered then the a report of plagiarism will be assigned against the students profile.

- **Presentation**

A teacher can take presentation of the students. The student can upload their ppt file against their profile. Teacher will mark the presentations and which will again be stored in the students profile.

- **Lab evaluation & Final**

Teacher will give input of the marks of labs, lab final and term final which will be stored against the students profile.

❖ Script

Teachers will create their own template where they can set their own criteria for each question to judge a script easily. There will be an option for comments or giving feedback. If the teacher wants, he can also store his marks according to his fixed template which will again be attached against the student's profile. Script will also get its own id automatically.

❖ Result making and distribution

The teachers will select criteria of marks such as quiz(if any), midterm(if any), assignment(if any), presentation(if any), Plagiarism(if any mark is to be deducted), lab(if any), lab final(if any), attendance(if any) which are stored against the students profile and also set weights for different criteria's. The process will return the final grade of the student which will be sent to him/her by email. Student can see his own grade only . Teacher can see grade of all students.

❖ Course Evaluation

A bar diagram depending on the final grades of the students will be generated by the system. Both teacher and students can view and download the diagram for further use.

❖ Search

Both student and teacher can search for different items from the forum. The teacher can search the profile of a particular student. The student can search his own log to see his uploaded documents.

❖ Notice Board

There will be a centralized notice board where the teacher will only be able to add notices. The students can only view the notices. Notice have the following features

- **Add File Notice**

The teacher can add a file as notice which will be visible to the Students.

- **Create Notice**

A teacher can create a notice by giving topic, date, time, place, notification time. This created notice will be shown in the notice board. The students can see it and from fixed notification time before the upcoming date notifications of

the notice will be again sent to the students by email on a fixed basis.

❖ Over all control

The overall control of the system will be in hands of the teacher. Teacher can add or remove any student from the course. If the student is removed, he will not be able to further participate in that course. The teacher will not be able to manipulate any marks nor information of the students. The student can change their own information but that has to be approved by the teacher. Student and teacher can log out of the system, any time he/she wants

4 SCENARIO BASED MODELING OF COURSE MANAGEMENT SYSTEM

This chapter describes the Scenario Based Model for the "Course Management System"

4.1 INTRODUCTION

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

4.2 DEFINITION OF USE CASE

Use case defines the 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 roles 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.

Primary Actors: As digital course management system incorporates only teachers and students. So, the system has two primary actor: teacher and student.

Secondary Actor: The system also acts as an internal actors in some cases. Therefore, it is the secondary actor of our system.
Outside system: E-mail.

4.3.1 LEVEL-0 USE CASE DIAGRAM - CMS

Level: 0(CMS)

Use case Name: CMS

Use case Id: 0

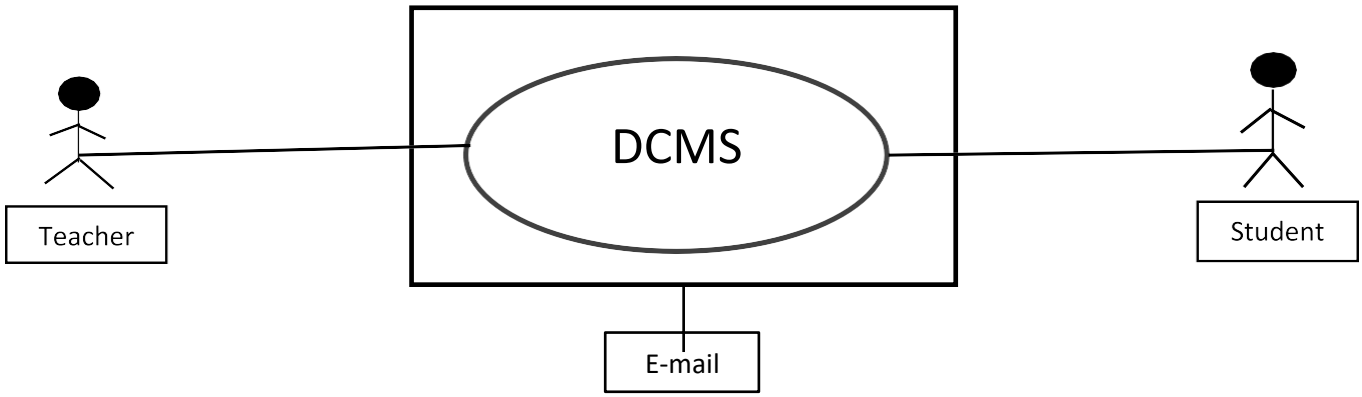


Fig 1: level 0

Primary actor: Teacher,

Student.

Outside System: E-Mail.

Description: This use case shows the low level interaction between system and actors.

4.3.2 LEVEL-1 USE CASE DIAGRAM- CMS SUB SYSTEM

Level: 1:
Use case Name: CMS SUB SYSTEM
Use case Id: 1

Primary actor: Teacher, Student.
Outside System: E-Mail.

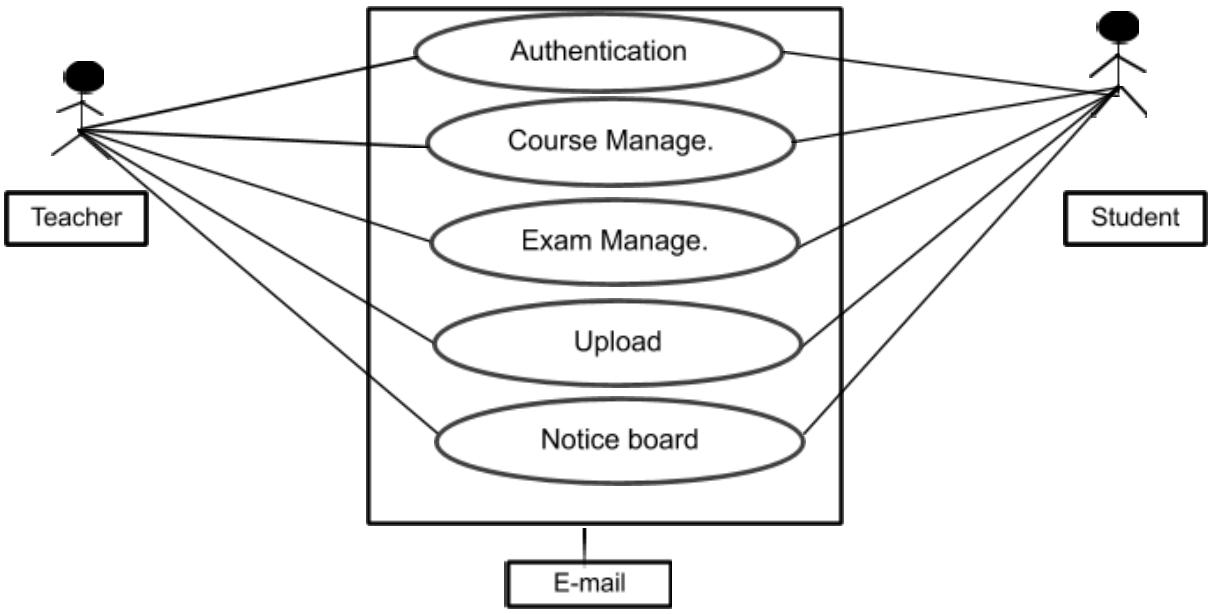


Fig 2: level 1

Description: From this level all the subsystem of the proposed main system and connectivity of those subsystem through actors has been explicit. From this level interaction between actors and subsystem will be clearer. Here, the whole system are divided into six subsystem and E-mail is the outside system in this proposed system

4.3.2.1 LEVEL-1.1 USE CASE DIAGRAM- Authentication

Level: 1.1:
Use case Name: Authentication
Use case Id: 1.1

Primary actor: Teacher, Student.
Outside System: E-Mail.

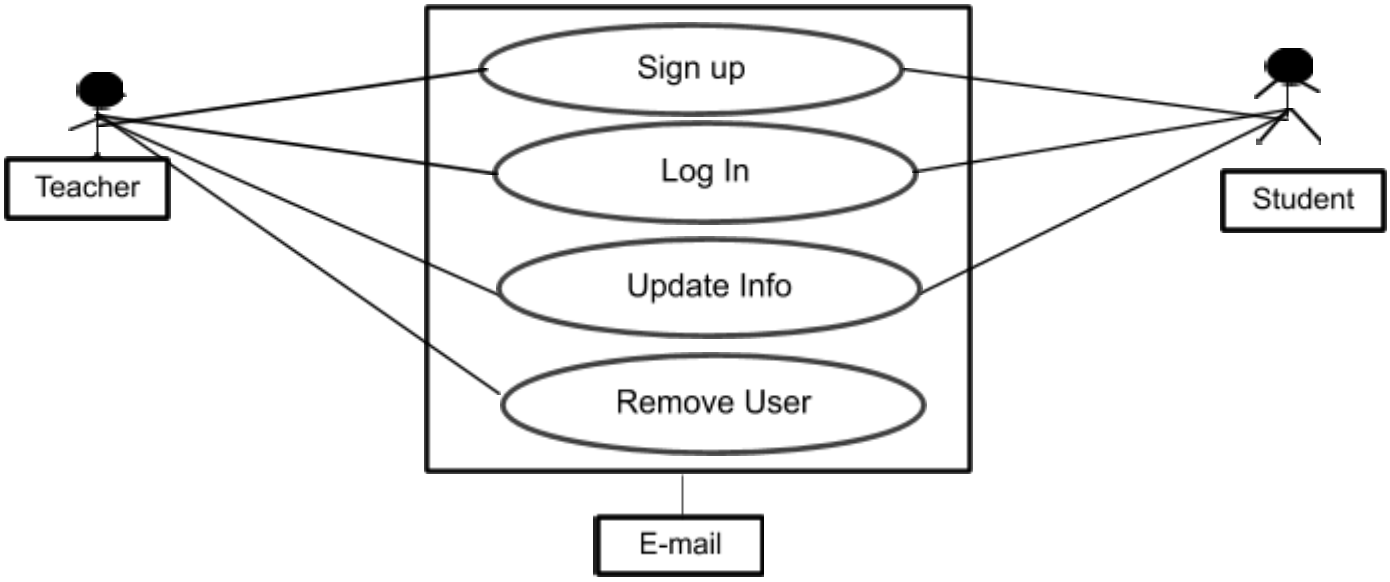


Fig 3: level 1.1

Description: The whole authentication process has been explicit in this level in four more subsystems. They are Registration, log in, update info, remove member.

LEVEL-1.1.1 USE CASE DIAGRAM- Sign Up

Level: 1.1.1:
Use case Name: Sign Up
Use case Id: 1.1.1

Primary actor: Teacher, Student.
Outside System: E-Mail.

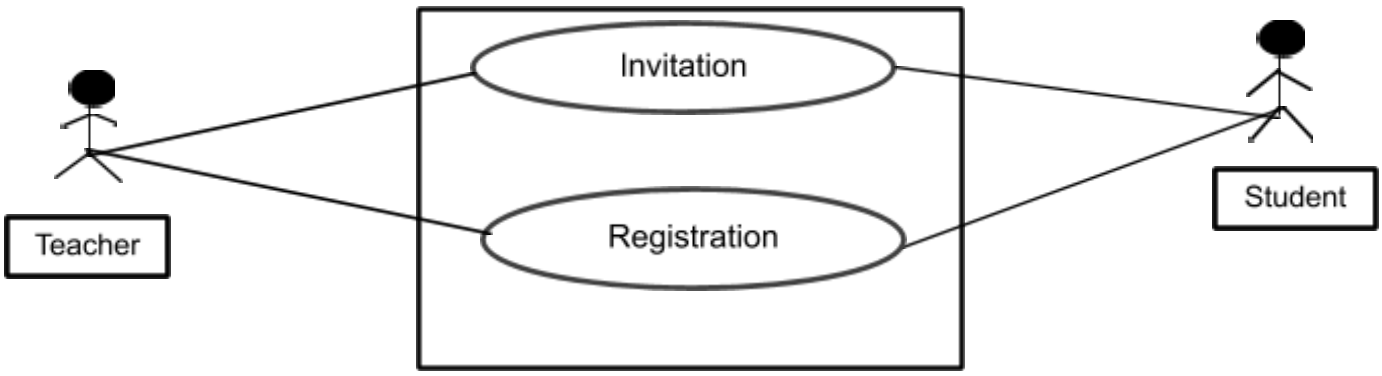


Fig 4: level 1.1.1

Level: 1.1.1.1:
Use case Name: invitation

Primary actor: Teacher, Student.
Outside System: E-Mail.

Description: Course teacher initiates a course through inviting his/her students with an invitation link which will automatically generated by the system when the course instructor open a new course.

Action: Sending invitation link.
Reply: invitation sent.

Level: 1.1.1.2
Use case Name: Registration
Primary actor: Student.
Secondary actor: system
Outside System: X

Action: The students will register to a new course with an invitation link sent by the course instructor.
Reply: Registered as a new member and sent for approval.

LEVEL-1.1.2 USE CASE DIAGRAM- Log in

Level 1.1.2 (Log In):
Use case Name: Log In
Primary actor: Teacher, student.
Secondary actor: system

Action: User provides required info to enroll a course.
Reply: Logged in.

LEVEL-1.1.3 USE CASE DIAGRAM- Update profile info

Level 1.1.3(Update profile info)
Use case Name: Update profile info
Use case Id: 1.1.3
Primary actor: Teacher, student.
Secondary actor: system

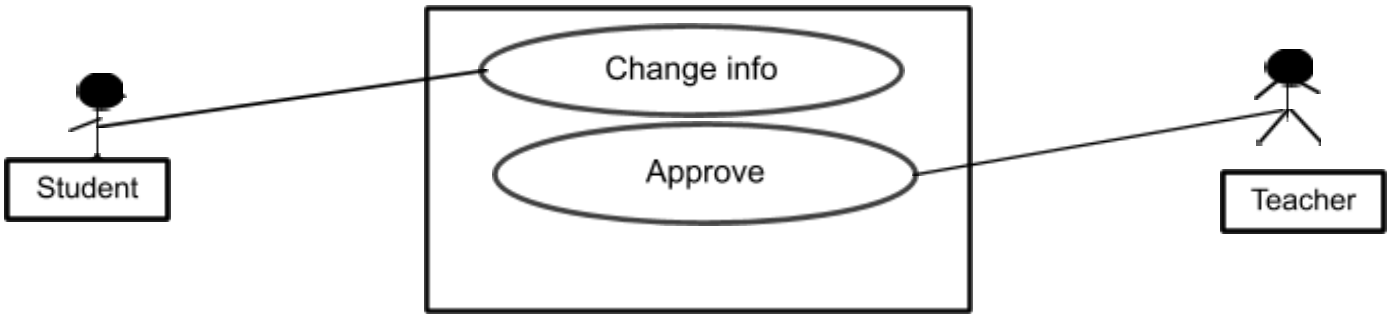


Fig 5: level 1.1.3

Level: 1.1.3.1 (Change info):
Use case Name: Change info
Primary actor: student.

Action: Student may want to change any profile info. That info will be stored for further approval.
Reply: Teacher may approve the change or not

LEVEL-1.1.4 USE CASE DIAGRAM- Remove user

Level: 1.1.4 (Remove user):
Use case Name: Remove user
Primary actor: Teacher.

Action: Course instructor may want to remove a student.
Reply: User Removed

4.3.2.2 LEVEL-1.2 USE CASE DIAGRAM- Course management

Use case Name: Course management
Use case Id: 1.2
Primary actor: Teacher, student.
Secondary actor: System

.

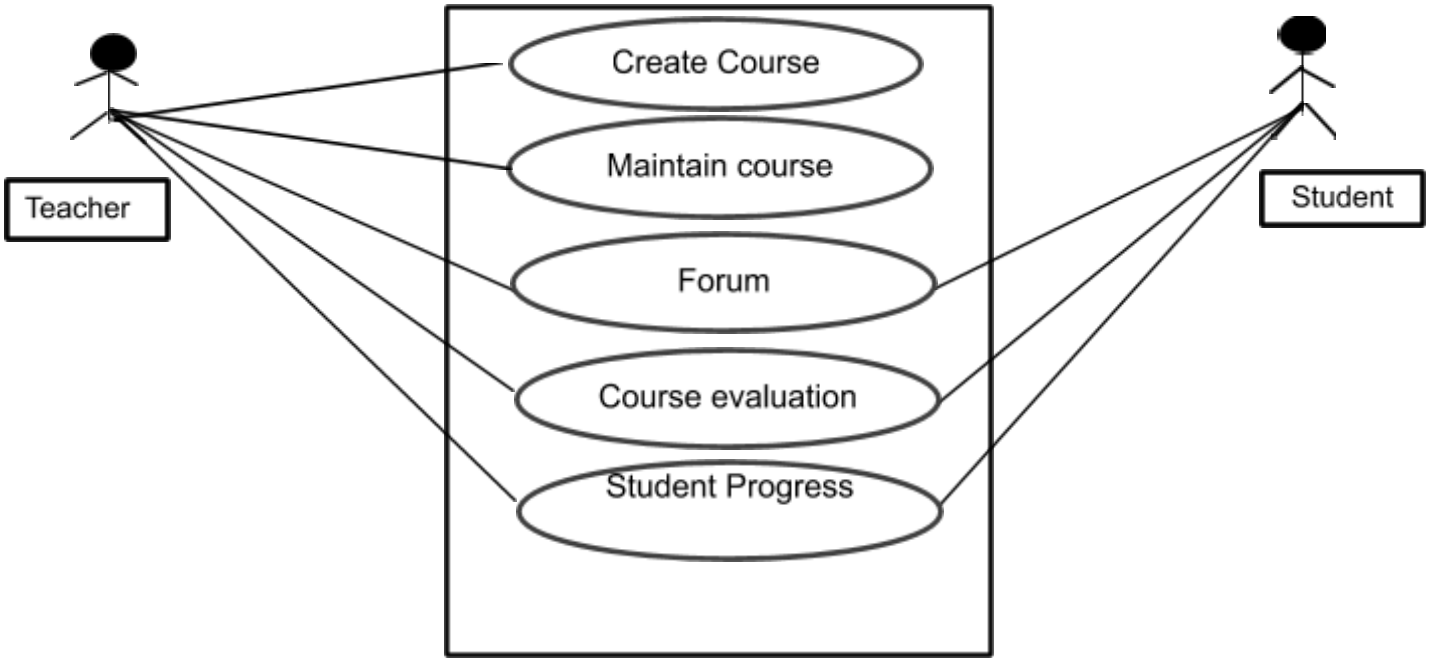


Fig 6: level 1.2

Description: Here, the course management section is been divided into five sub-section. They are Create course, maintain course, forum, course evaluation, student progress evaluation.

LEVEL-1.2.1 USE CASE DIAGRAM- Create course

Level: 1.2.1(Create course)
Use case Name: Create course Use case Id: 1.2.1
Primary actor: Teacher.
Secondary actor: System.

Action: Teacher initiate a course by enrolling the system as a course instructor.
Reply: A new course has been created.

LEVEL-1.2.2 USE CASE DIAGRAM- Maintain course

Level: 1.2.2(Maintain course):
Use case Name: Maintain course

Use case Id: 1.2.2
Primary actor: Teacher, student.
Secondary actor: System.
Description: Divided into three sub process, class lecture, online attendance, class record.

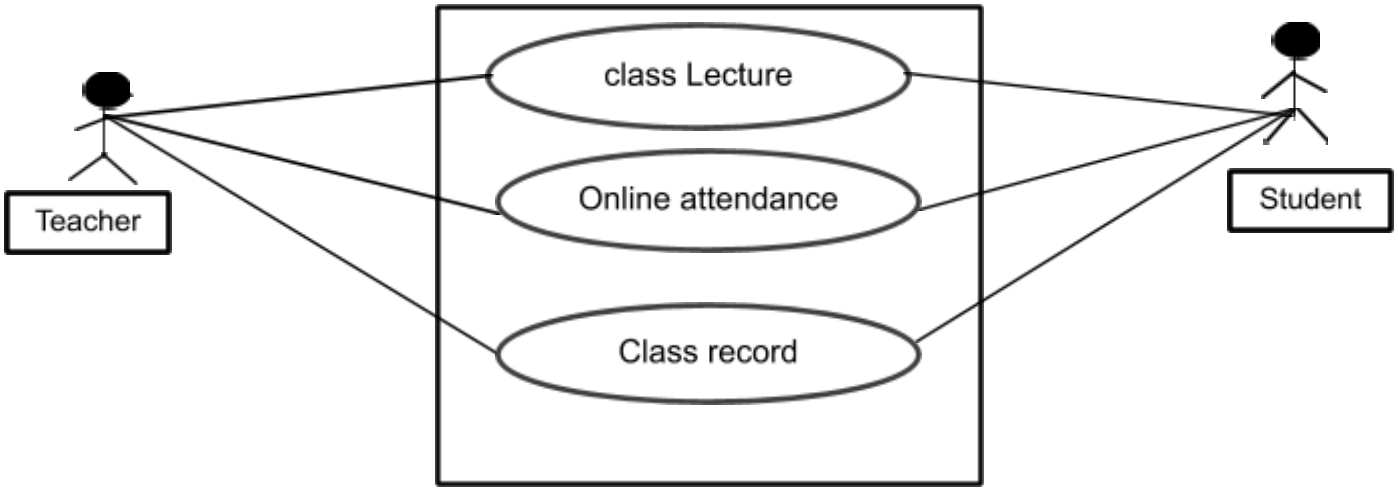


Fig 7: level 1.2.2

Level: 1.2.2.1(class lecture):
Use case Name: class lecture
Use case Id: 1.2.2.1
Primary actor: Teacher, student.

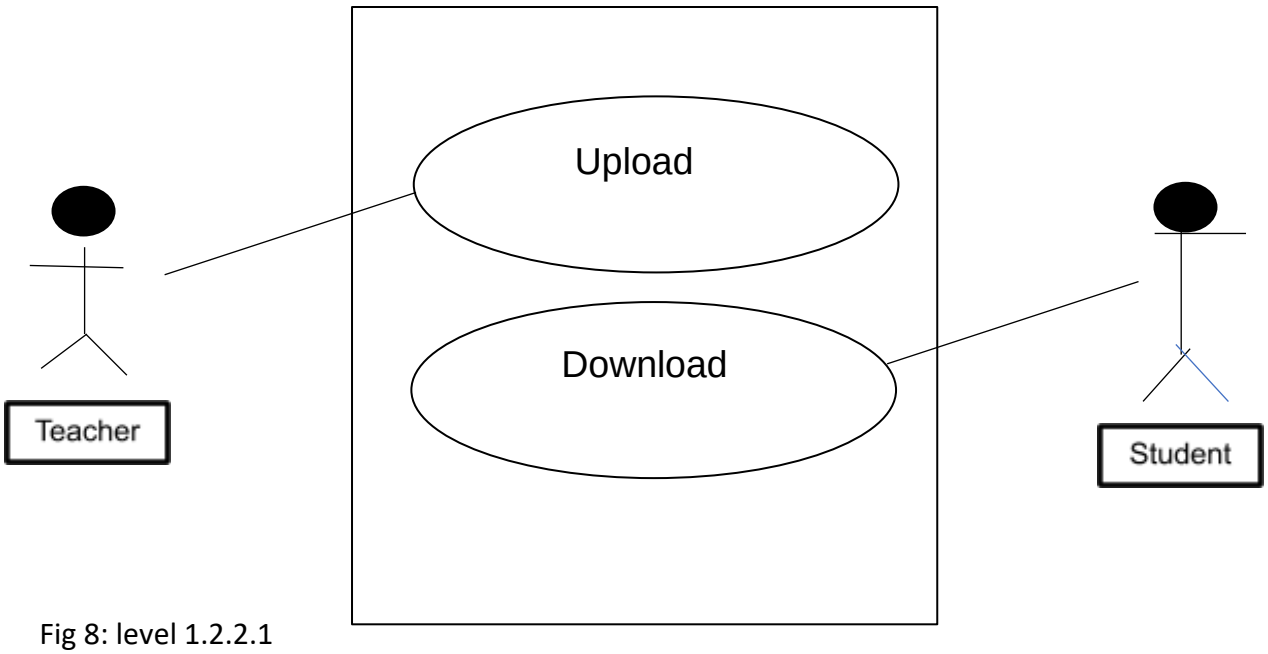


Fig 8: level 1.2.2.1

Action: Teacher will upload Class lecture and the students can only view and download those lectures.

Reply: Class lectures will be uploaded and viewed & downloaded by the teacher and the students respectively

Level: 1.2.2.2(Online attendance):

Use case Name: Online attendance

Primary actor: Teacher, student.

Secondary actor: System.

Action: Teacher will open the attendance system for a certain time and the students will give attendance within that time.

Reply: Attendance will be taken.

Level: 1.2.2.3(Class Record):

Use case Name: Class Record

Primary actor: Teacher.

Secondary actor: System.

Action: A class record will be kept and users may want to view the record.

Reply: Class record has been viewed.

LEVEL-1.2.3 USE CASE DIAGRAM- Forum

Level 1.2.3(Forum):
Use case Name: Forum
Use case Id: 1.2.3
Primary actor: Teacher, student.

Secondary actor: System

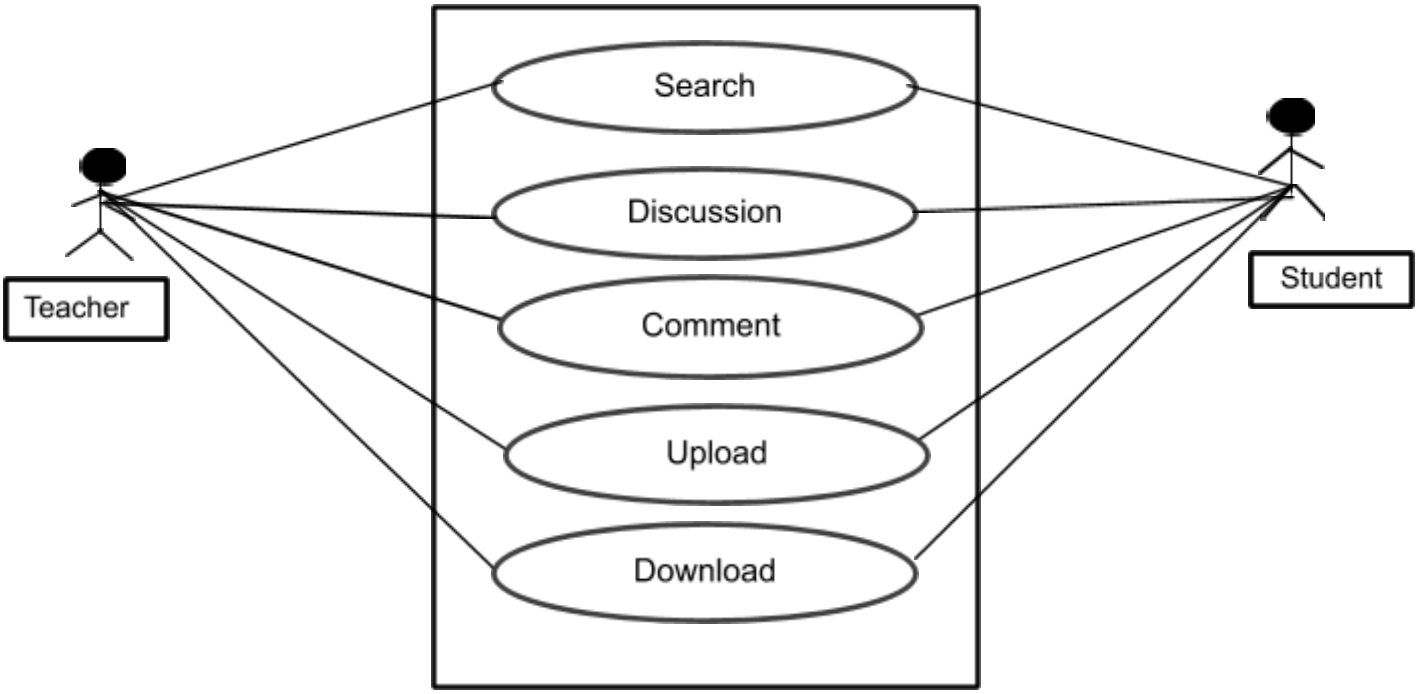


Fig 9: level 1.2.3

Description: An internal forum can be created and all the members of that course including course instructor where anyone can take part in the discussion and can upload and download a file.

Level: 1.2.3.1(Search):

Use case Name: Search

Use case Id: 1.2.3.1

primary actor: student, teacher

Secondary actor: system

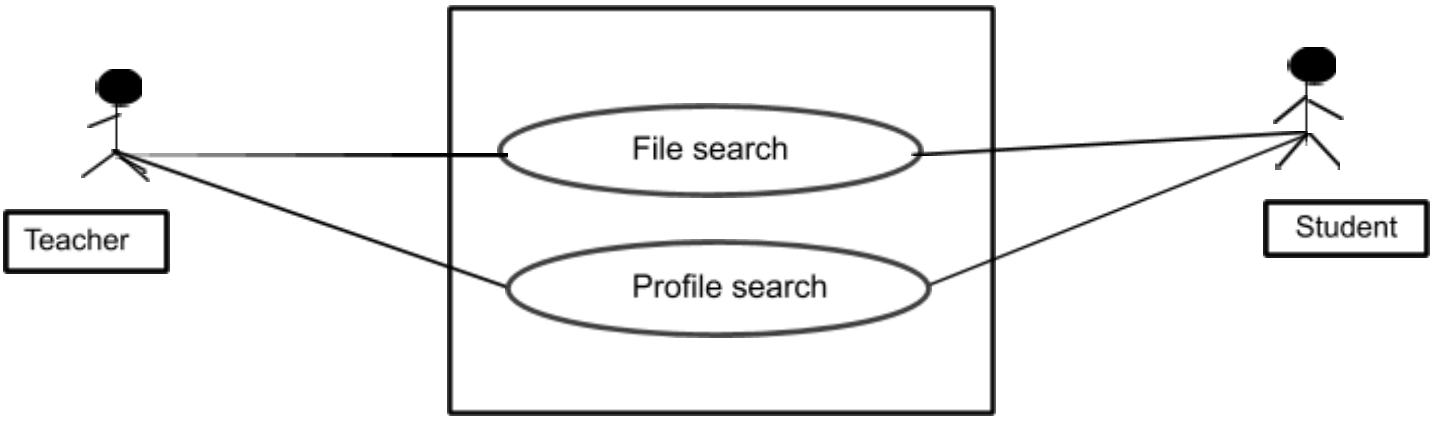


Fig 10: level 1.2.3.1

Level: 1.2.3.1.1(File Search):

Use case Name: file Search

Primary actor: Teacher, student.

Action: Both teacher and student may search a file.

Reply: Viewed the search result.

Level: 1.2.3.1.2(Profile Search):

Use case Name: profile Search

Primary actor: Teacher, student.

Action: Both teacher and student may search a profile.

Reply: Viewed the search result.

Level 1.2.3.2(Discussion):

Use case Name: Discussion

Primary actor: Teacher, student.

Secondary actor: System

Action: Members may want to take part in group discussion.

Reply: Group discussion started.

Level 1.2.3.3(comment):

Use case Name: comment

Primary actor: Teacher, student.

Action: users may want to do comment on certain upload.
Reply: commented.

Level 1.2.3.4(Upload):

Use case Name: Upload

Primary actor: Teacher, student.

Action: Any authorized user may want to upload in forum.
Reply: uploaded.

Level 1.2.3.5(Download)

Use case Name: Download

Primary actor: Teacher, student.

Action: Any authorized user may want to download in forum.

Reply: downloaded

LEVEL-1.2.4 USE CASE DIAGRAM- course evaluation

Level 1.2.4(course evaluation):
Use case Name: course evaluation
Use case Id: 1.2.4
Primary actor: Teacher, student.

Secondary actor: System

Description: Teachers may provide the marks or system may collect data on a regular basis and will generate a statistical representation about the course progress.

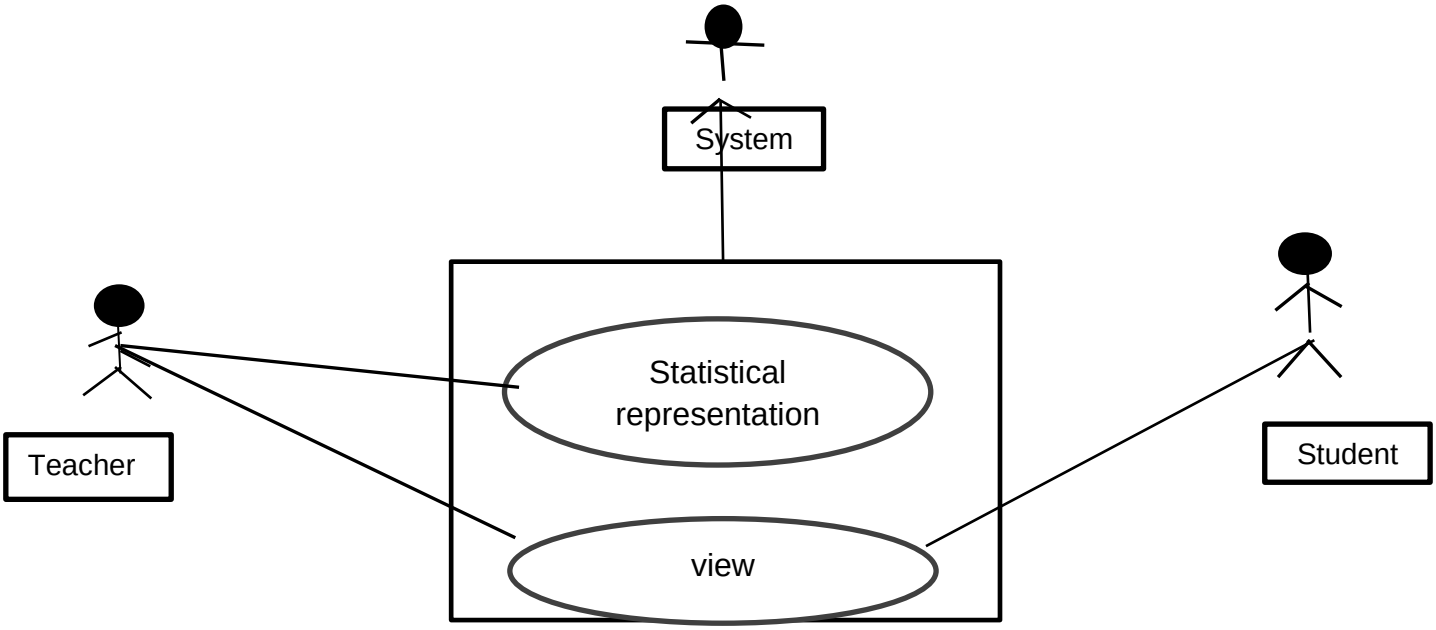


Fig 11: level 1.2.4

Level: 1.2.4.1 (Statistical representation):

Use case Name: Statistical representation

Use case Id: 1.2.4.1

Primary actor: Teacher.

Secondary actor: System

Action: Teacher will provide data for generating a statistical overview about the whole course.

Reply: system will generate and calculate those data.

Level: 1.2.4.2(view):

Use case Name: view

Use case Id: 1.2.4.2

Primary actor: Teacher, student.

Action: any authorized user may want to see the course progress.

Reply: viewed.

LEVEL-1.2.5 USE CASE DIAGRAM- Student progress

Level: 1.2.5(Student progress):

Use case Name: Student progress

Use case Id: 1.2.5

Primary actor: Teacher, student.

Secondary actor: system

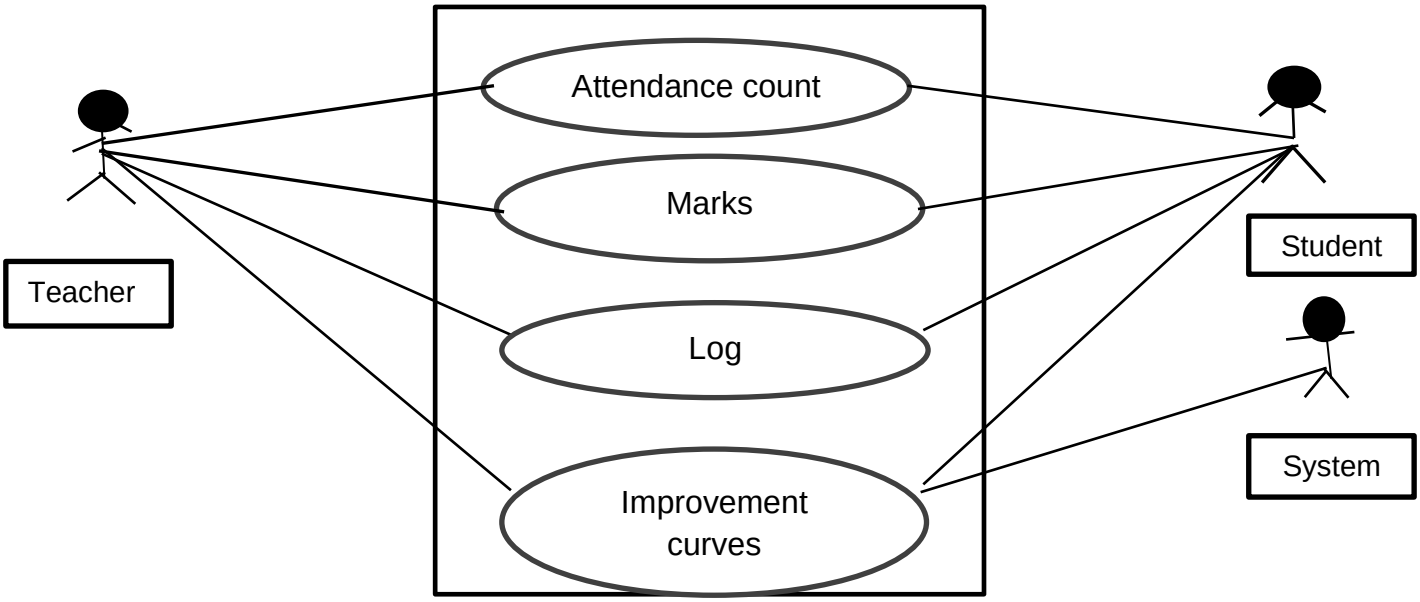


Fig 12: level 1.2.5

Description: The whole student progress process has been divided into four sub process. System will collect data on a regular basis and will generate a progress diagram.

Level 1.2.5.1(attendance count):

Use case Name: attendance count

Primary actor: Teacher, student.

Use case Id: 1.2.5.1

Secondary actor: system

Action: Teacher will initiate a session for submitting attendance and the students will provide their attendance using their id.

Reply: Attendance counted.

Level: 1.2.5.2(Marks):

Use case Name: Marks

Primary actor: Teacher, student.

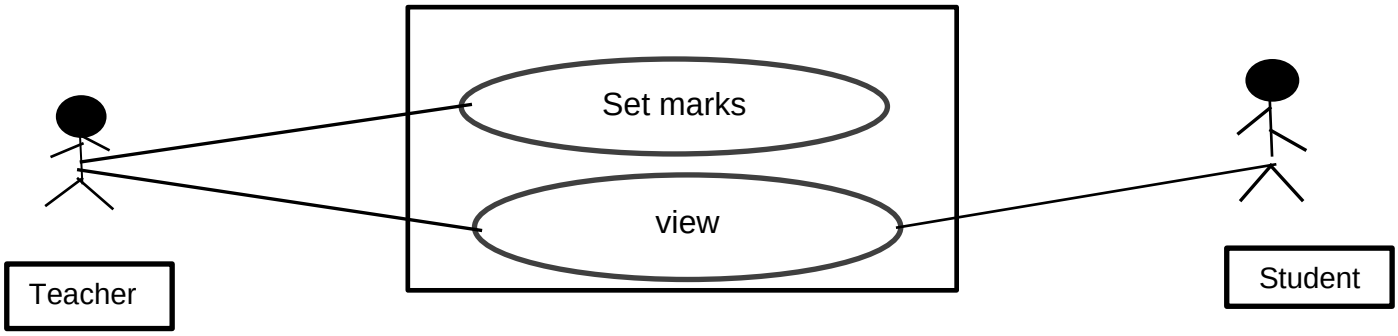


Fig 13: level 1.2.5.2

Level: 1.2.5.2.1(set marks):

Use case Name: set marks

Use case Id: 1.2.5.3.1

Primary actor: Teacher.

Action: Teacher will provide marks.

Reply: the marks will be set against the student profile.

Level: 1.2.5.2.2(view marks):

Use case Name: view marks

Use case Id: 1.2.5.3

Primary actor: Teacher, student.

Action: Both teacher and student may want to view the marks.

Reply: viewed.

Level: 1.2.5.3(Log):

Use case Name: Log

Use case Id: 1.2.5.3

Primary actor: student.

Action: A log should be kept for the student.

Reply: Log kept.

Level: 1.2.5.4(Improvement curves):

Use case Name: Improvement curves

Use case Id: 1.2.5.4

Primary actor: student.

Secondary actor: system.

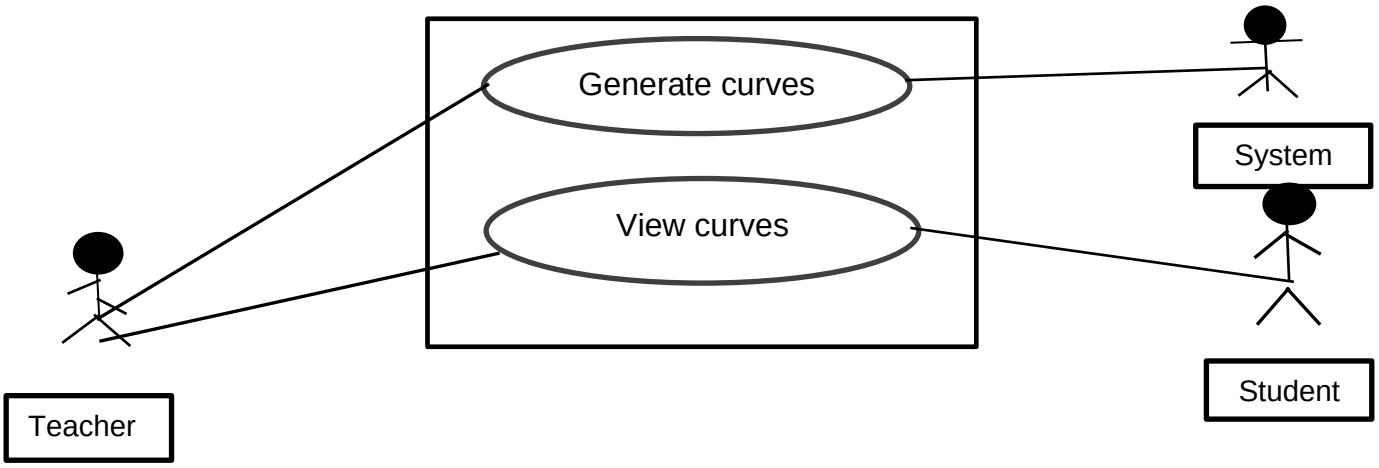


Fig 14: level 1.2.5.4

Level: 1.2.5.4.1(Generate curves):

Use case Name: Generate curves

Use case Id: 1.2.5.4.1

Primary actor: Teacher.

Secondary actor: system

Action: Teacher will provide data to produce an improvement curves.

Reply: Improvement curves generated.

Level: 1.2.5.4.2(View curves):

Use case Name: View curves

Use case Id: 1.2.5.4.2

Primary actor: Teacher, student.

Action: Both teacher and student may want to view the improvement curves.

Reply: Improvement curves viewed.

Level: 1.3(Exam and result management):

Use case Name: Exam and result management

Use case Id: 1.3

Primary actor: Teacher, student.

Description: Whole exam process has been divided into three sub processes. Where teacher can set question and marks & students can only view those and can also review the script paper.

4.3.2.3 LEVEL-1.3 USE CASE DIAGRAM- Exam management

Level: 1.3(Exam and result management):

Use case Name: Exam and result management

Use case Id: 1.3

Primary actor: Teacher, student.

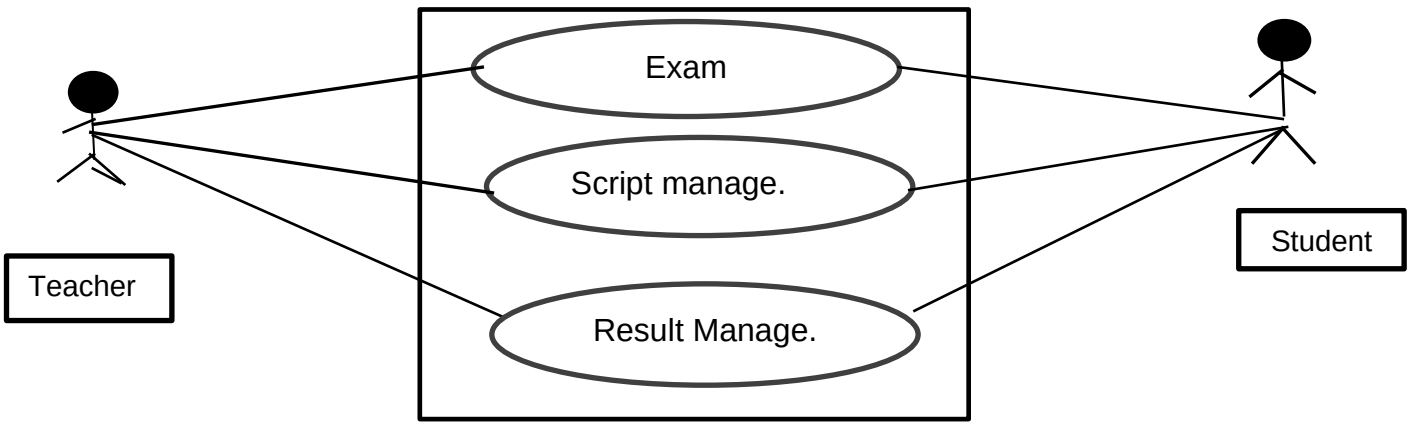


Fig 15: level 1.3

Description: Whole exam process has been divided into three sub processes. Where teacher can set question and marks & students can only view those and can also review the script paper.

Level: 1.3.1(Exam management):

Use case Name: Exam management

Use case Id: 1.3.1

Primary actor: Teacher, student.

Description: In examination section teachers have access on every section but student can only submit assignment and presentation slides.

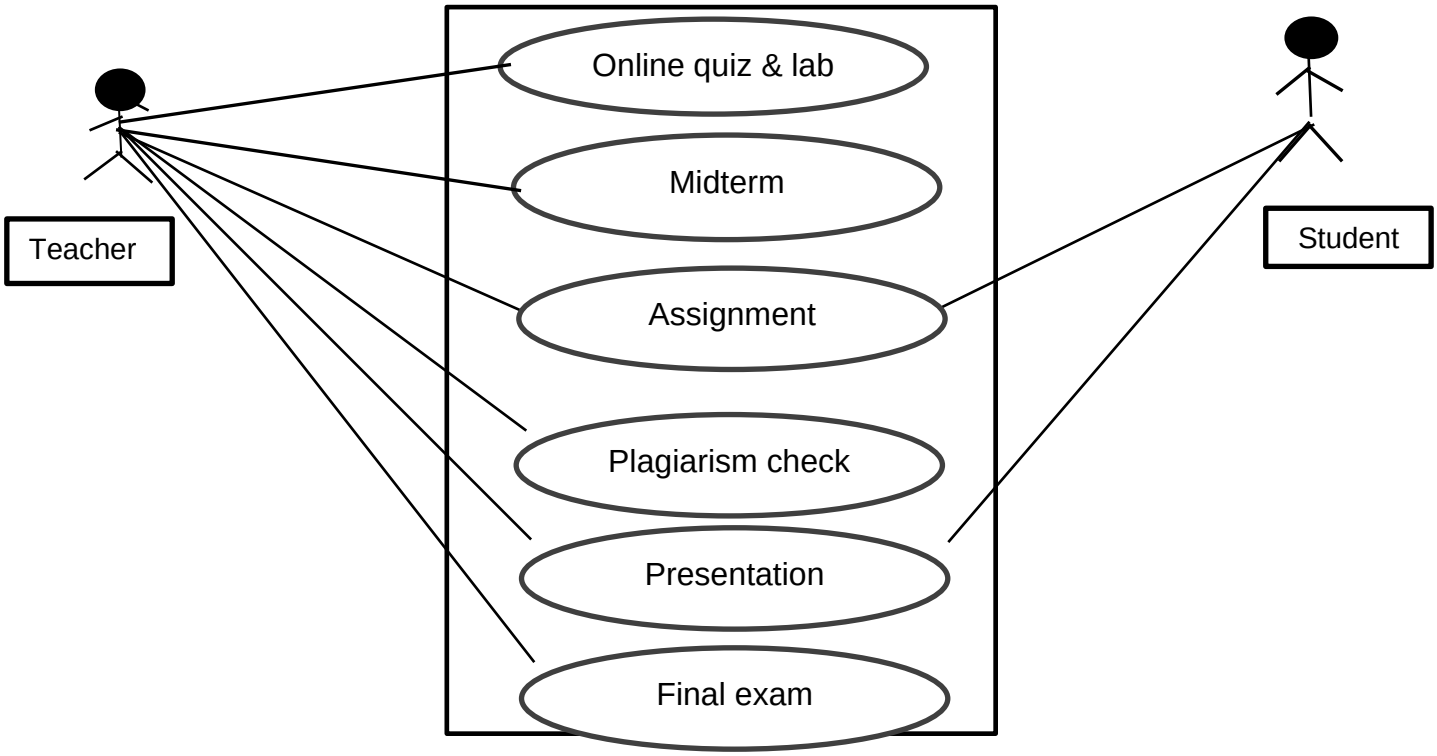


Fig 16: level 1.3.1

Level: 1.3.1.1(Online quiz and lab):

Use case Name: Online quiz and lab

Use case Id: 1.3.1.1

Primary actor: Teacher.

Action: Teacher will set the question and also set the marks after evaluation.

Reply: Question and marks has been set.

Level: 1.3.1.2(Midterm):

Use case Name: Midterm

Use case Id: 1.3.1.2

Primary actor: Teacher.

Action: Teacher will set the question and also set the marks after evaluation.

Reply: Question and marks has been set.

Level: 1.3.1.3(Assignment):

Use case Name: Assignment

Use case Id: 1.3.1.3

Primary actor: Teacher, student.

Action: Student will submit their assignment and teacher will provide marks after evaluation.

Reply: marks has been set.

Level: 1.3.1.4(Plagiarism check):

Use case Name: Plagiarism check

Use case Id: 1.3.1.4

Primary actor: Teacher.

Secondary actor: system

Action: Teacher will initiate the plagiarism check.

Reply: Plagiarism checked and set the result against the student profile.

Level: 1.3.1.5(Presentation):

Use case Name: Presentation

Use case Id: 1.3.1.5

Primary actor: Teacher, student.

Action: Students will upload their presentation slide and teacher will set the marks.

Reply: Done.

Level: 1.3.1.6(Final exam):

Use case Name: Final exam

Use case Id: 1.3.1.6

Primary actor: Teacher, student.

Action: Teacher will set the question and also set the marks after evaluation.

Reply: Question and marks has been set.

Level 1.3.2(Script management):

Use case Name: Script management

Use case Id: 1.3.2

Primary actor: Teacher, student.

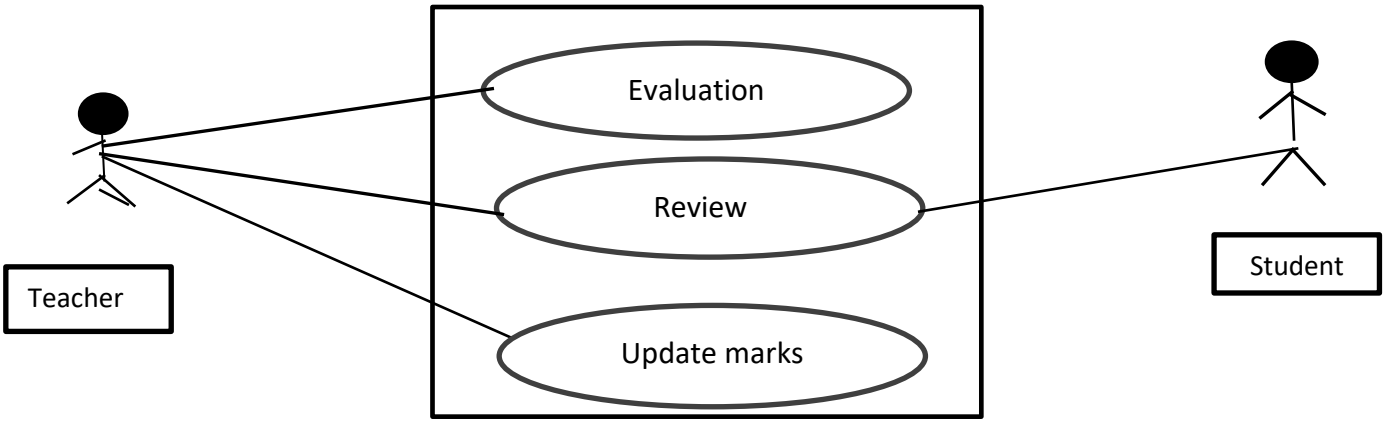


Fig 17: level 1.3.2

Level 1.3.2.1(Evaluation):

Use case Name: Evaluation
Use case Id: 1.3.2.1
Primary actor: Teacher
Action: Teacher will evaluate the script and set a marks.
Reply: Done.

Level 1.3.2.2(Review):

Use case Name: Review
Use case Id: 1.3.2.2
Primary actor: student
Action: Student may want to review any script.
Reply: Done.

Level 1.3.2.3(update marks):

Use case Name: update marks
Use case Id: 1.3.2.3
Primary actor: Teacher

Action: Teacher may want to update any marks.
Reply: Done.

4.3.2.4 LEVEL-1.4 USE CASE DIAGRAM- Upload

Level: 1.4(upload):

Use case Name: upload
Use case Id: 1.4
Primary actor: Teacher, student.
Description: according to the story upload may be two types- general file upload and assignment upload.

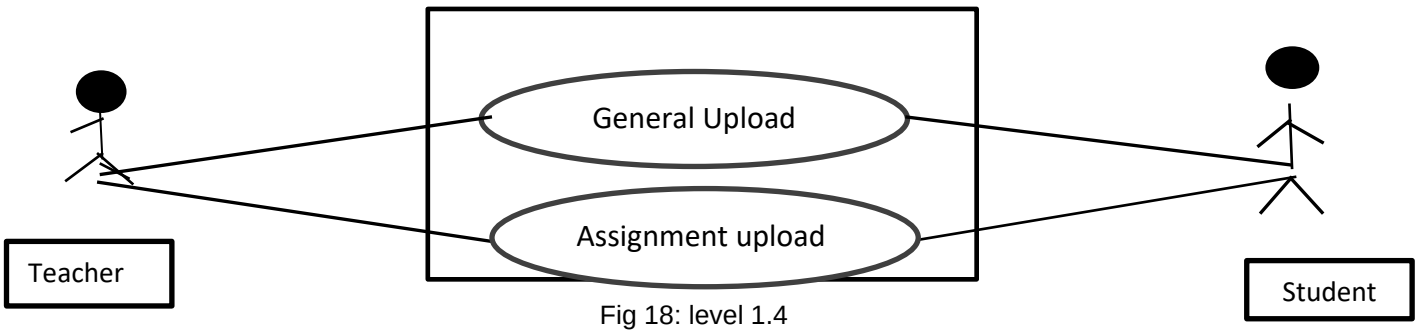


Fig 18: level 1.4

Level: 1.4.1(General Upload):

Use case Name: General upload

Use case Id: 1.4.1

Primary actor: Teacher, student.

Action: Any authorized user may want to upload any doc.

Reply: uploaded.

Level: 1.4.2(Assignment upload):

Use case Name: Assignment upload

Use case Id: 1.4.2

Primary actor: student, teacher.

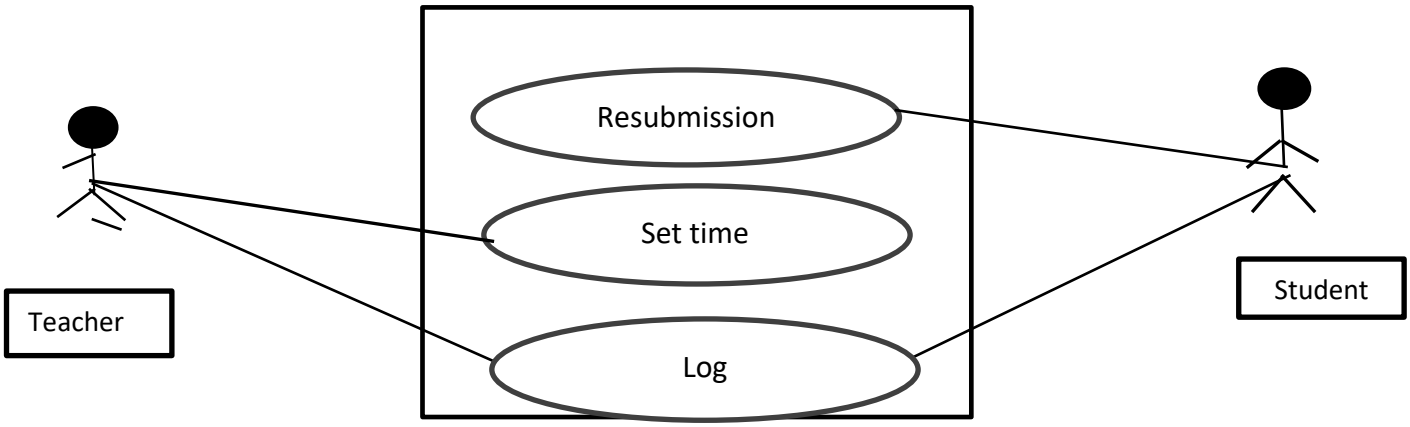


Fig 19: level 1..4.2

Level: 1.4.2.1(Resubmission):

Use case Name: Resubmission

Use case Id: 1.4.2.1

Primary actor: student.

Action: Student may want to resubmit any assignment.

Reply: Resubmitted.

Level: 1.4.2.2(Set time):

Use case Name: Resubmission

Use case Id: 1.4.2.2

Primary actor: Teacher.

Action: Teacher will set a time for the assignment submission and resubmission.

Reply: done.

Level: 1.4.2.3(Log):

Use case Name: Log

Use case Id: 1.4.2.3

Primary actor: Teacher, student.

Action: A log should be kept.

Reply: done.

4.3.2.4 LEVEL-1.5 USE CASE DIAGRAM- Notice Board

Level: 1.5(Notice board):

Use case Name: Notice board

Use case Id: 1.5

Primary actor: Teacher, student.

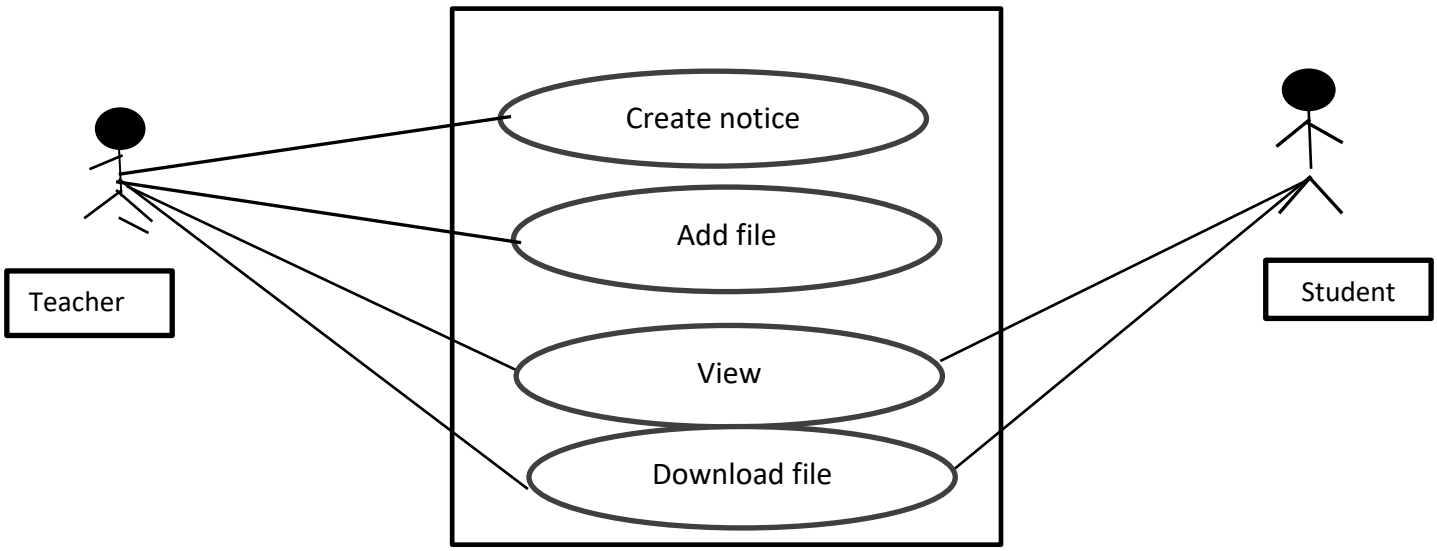


Fig 20: level 1.5

Level: 1.5.1(Create Notice):

Use case Name: Create Notice

Use case Id: 1.5.1

Primary actor: Teacher.

Action: Teacher will create a notice.

Reply: Done.

Level: 1.5.2(Add file):

Use case Name: Add file

Use case Id: 1.5.2

Primary actor: Teacher.

Action: Teacher may add file a notice.

Reply: Done.

Level: 1.5.3(View notice):

Use case Name: View notice

Use case Id: 1.5.3

Primary actor: Teacher, student.

Action: Both teacher and students may want to view a notice

Reply: Viewed

4.4 Activity Diagram Of CMS

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

4.4.1 Authentication

Activity name: Authentication

Activity id: 4.4.1

Version 1.1

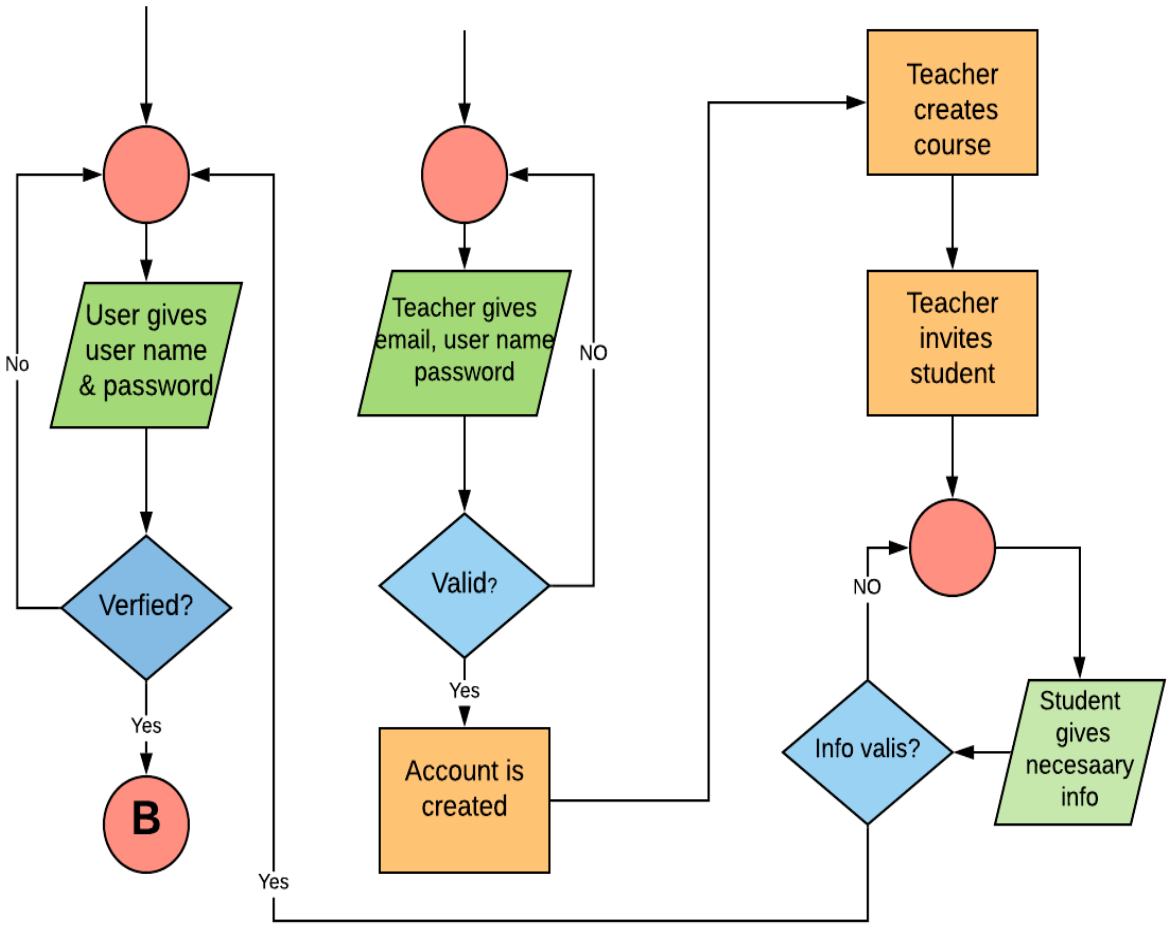


Fig 21: Authentication

4.4.2 Online Attendance

Activity name: Online Attendance

Activity Id: 4.4.2

Version: 1.0

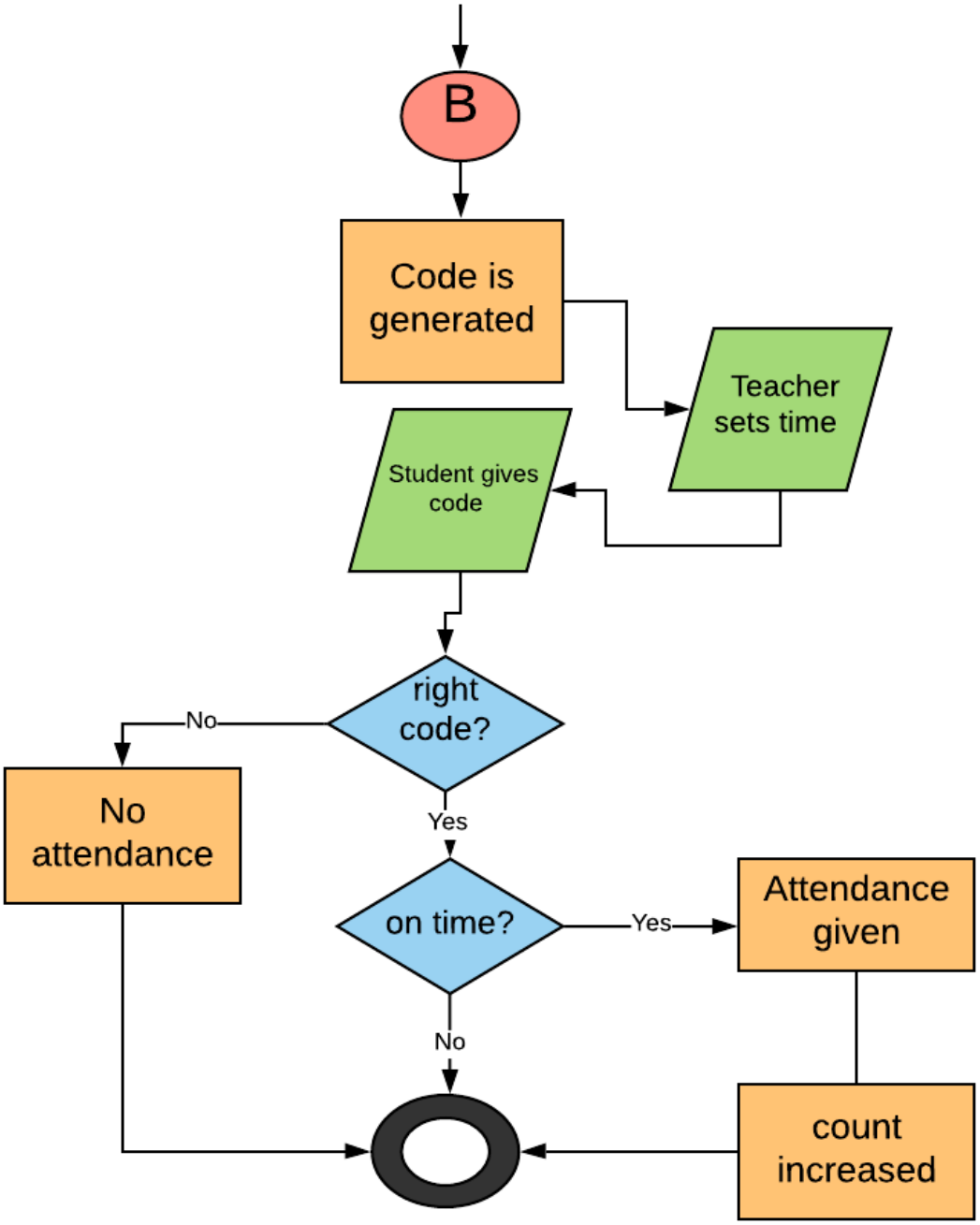


Fig 22: Online Attendance

4.4.3 Forum

Activity name: Forum

Activity Id: 4.4.3

Version: 1.2

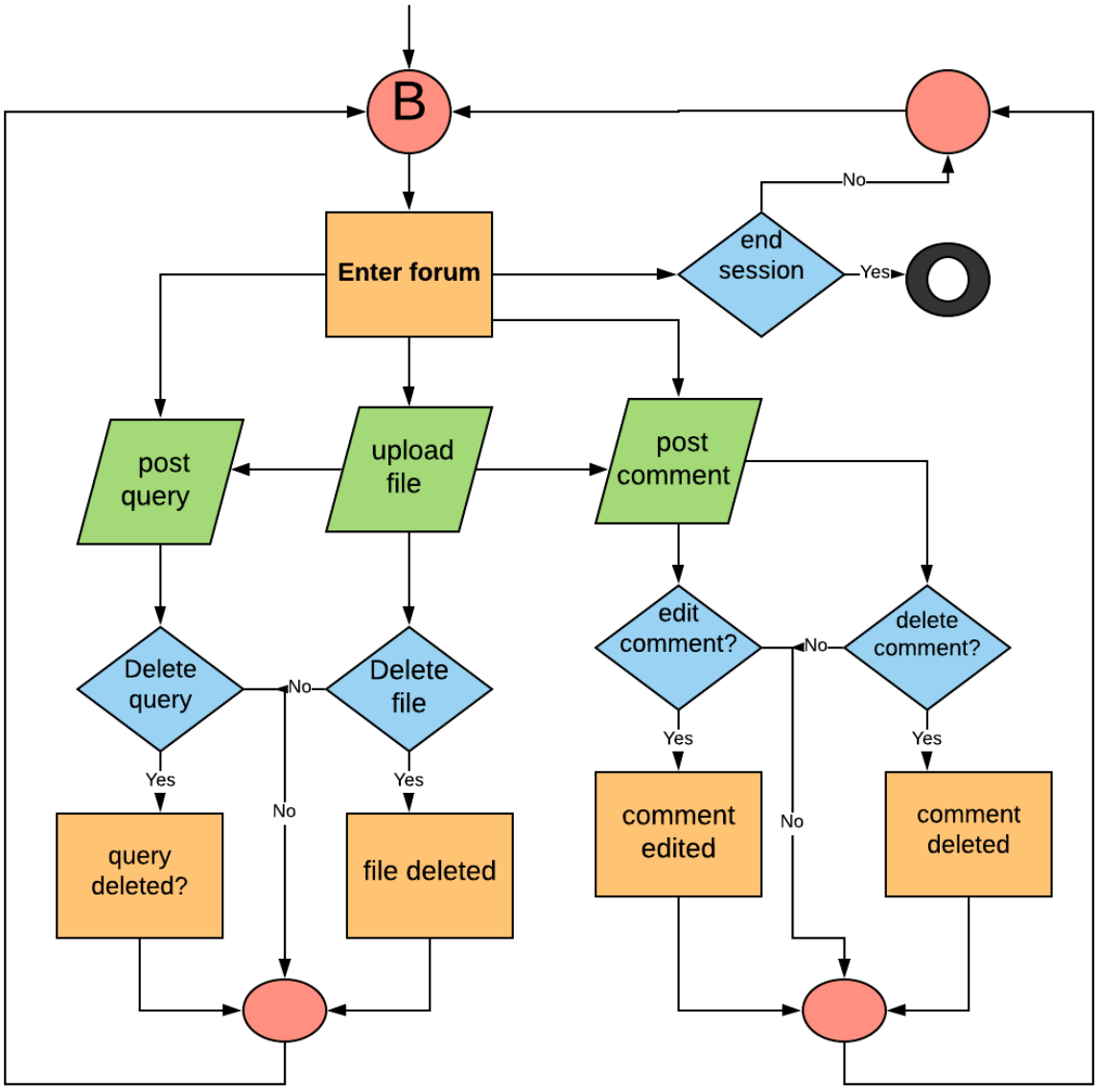


Fig 23: forum

4.4.4 Assignment

Activity name: Forum

Activity Id: 4.4.4

Version: 1.0

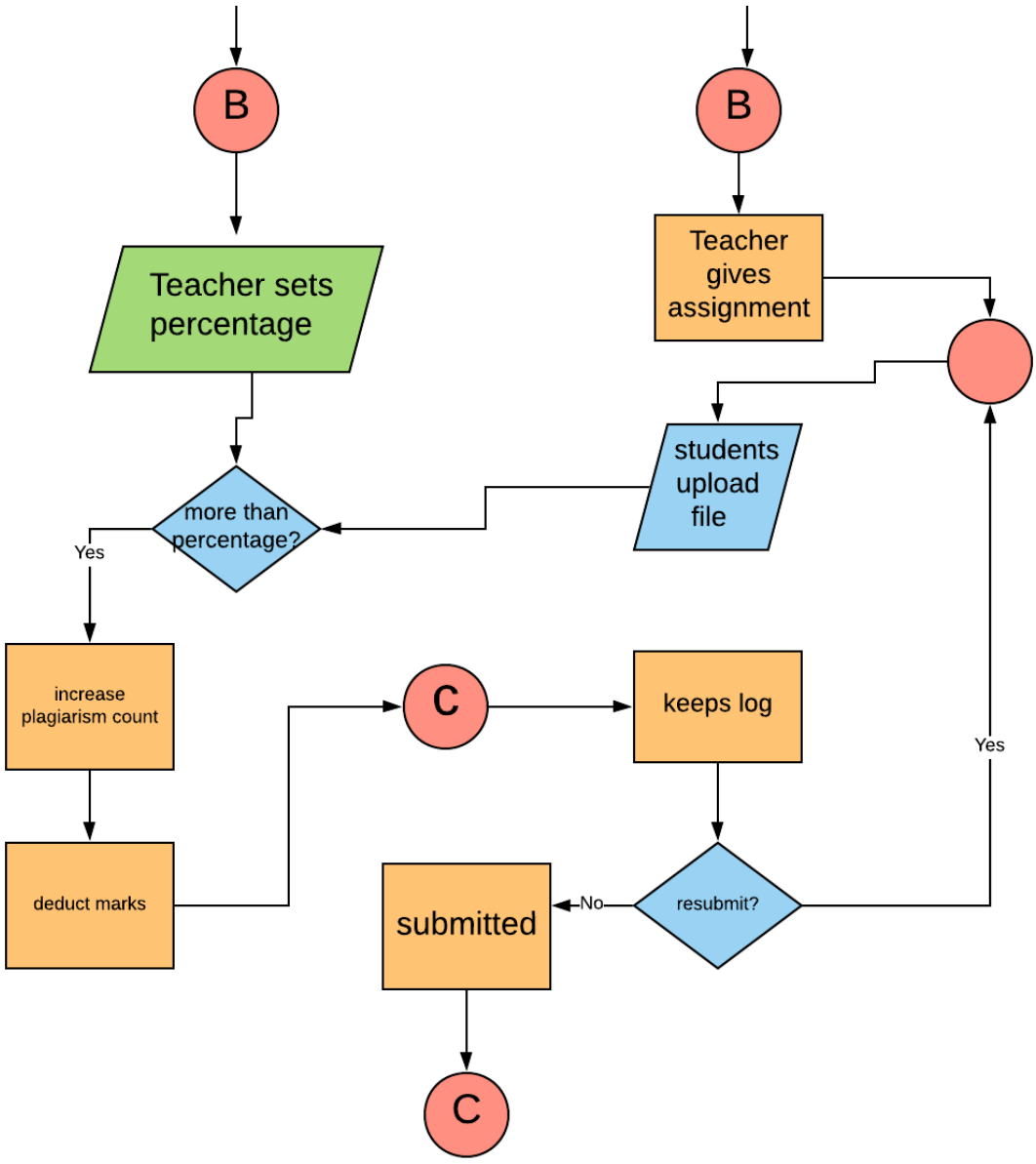


Fig 24: Assignment

4.4.5 Quiz

Activity name: Quiz

Activity Id: 4.4.5

Version: 1.0

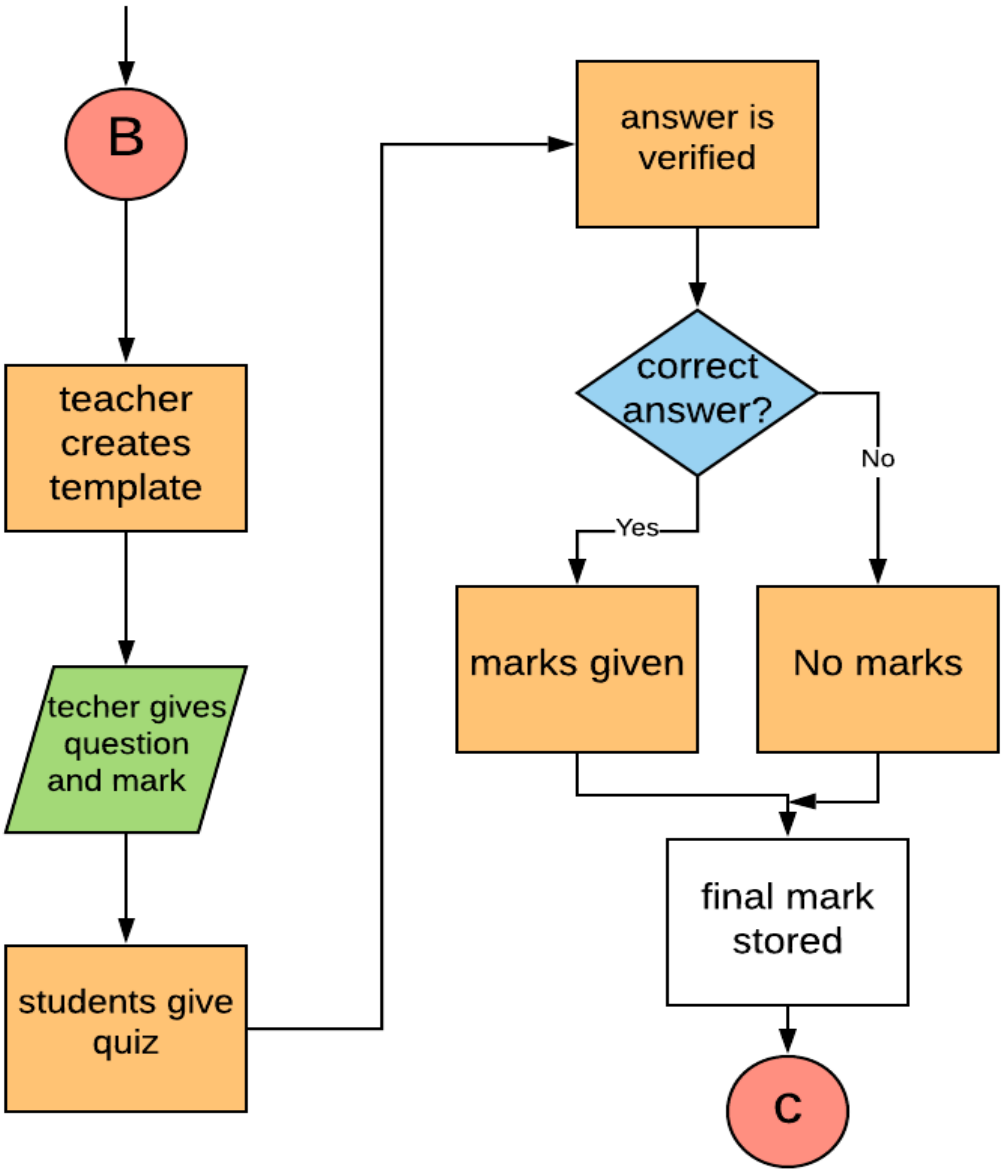


Fig 25: Quiz

4.4.6 Script

Activity name: Script

Activity Id: 4.4.6

Version: 1.0

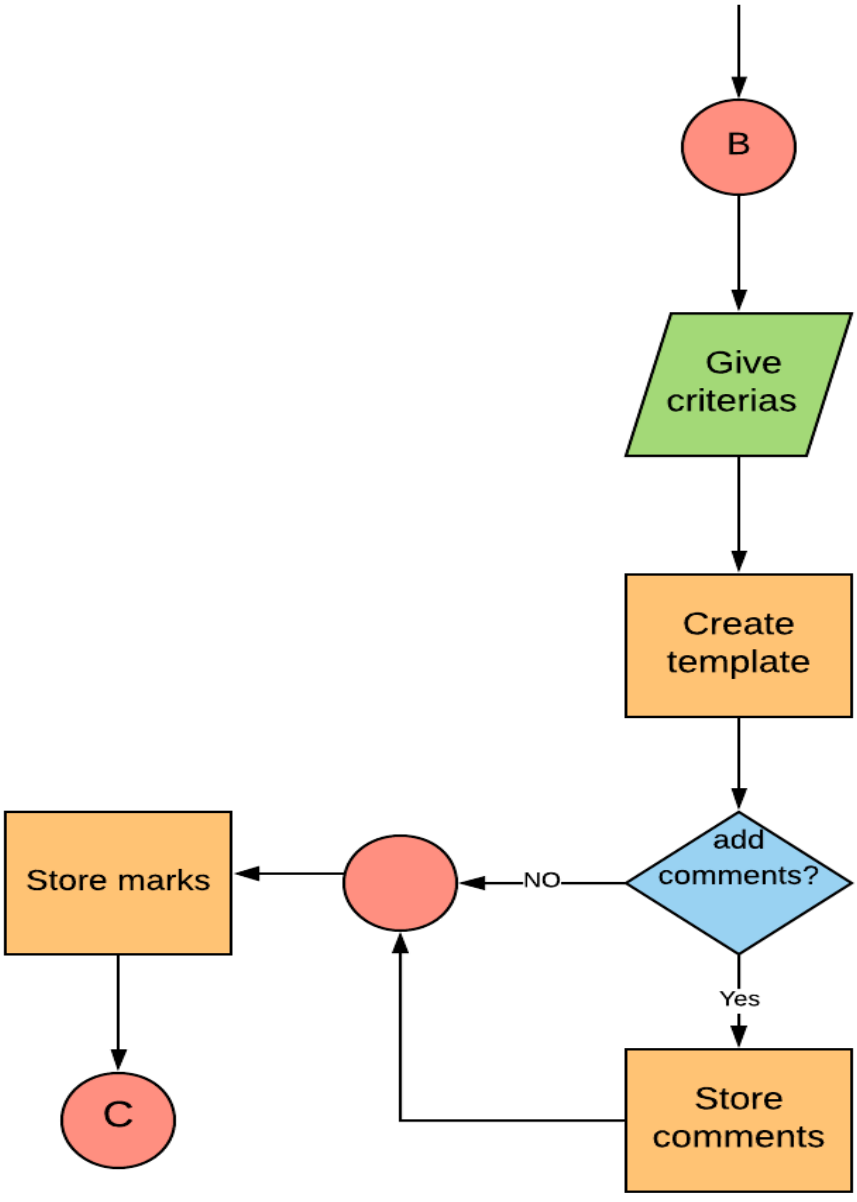


Fig 26: Script

4.4.7 Download

Activity name: Download

Activity Id: 4.4.7

Version: 1.0

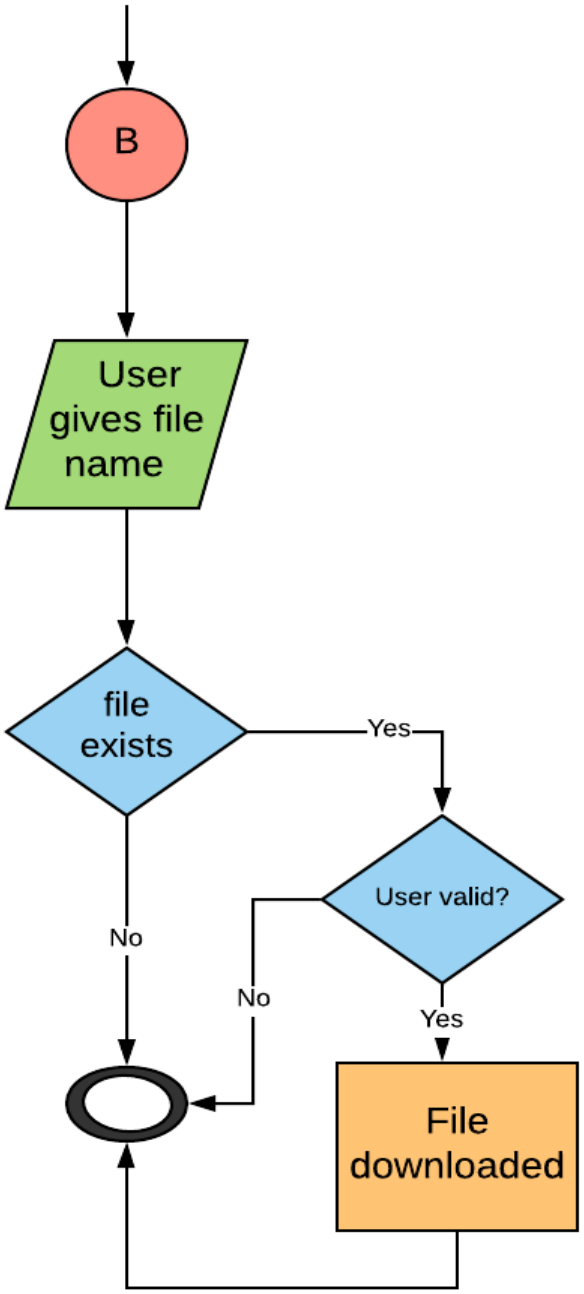


Fig 27: Download

4.4.8 Upload

Activity name: Upload

Activity Id: 4.4.8

Version: 1.0

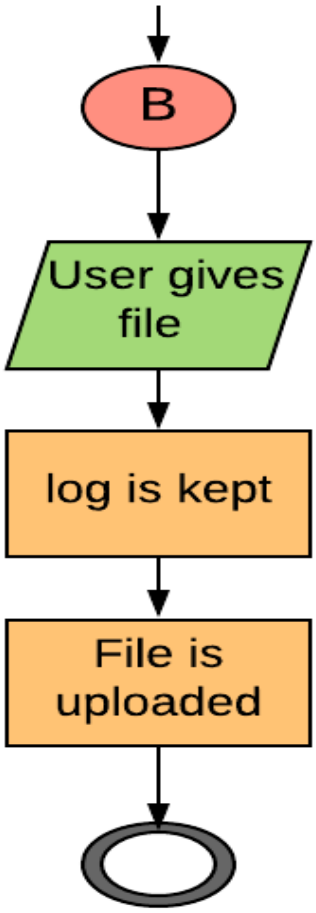


Fig 28: Upload

4.4.9 Course Curve

Activity name: Course carve

Activity Id: 4.4.9

Version: 1.0

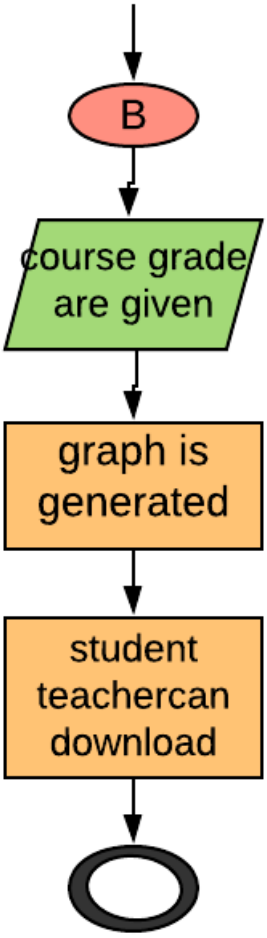


Fig 29. Course carve

4.4.10 Store & Generate Result

Activity name: Result

Activity Id: 4.4.10

Version: 1.2

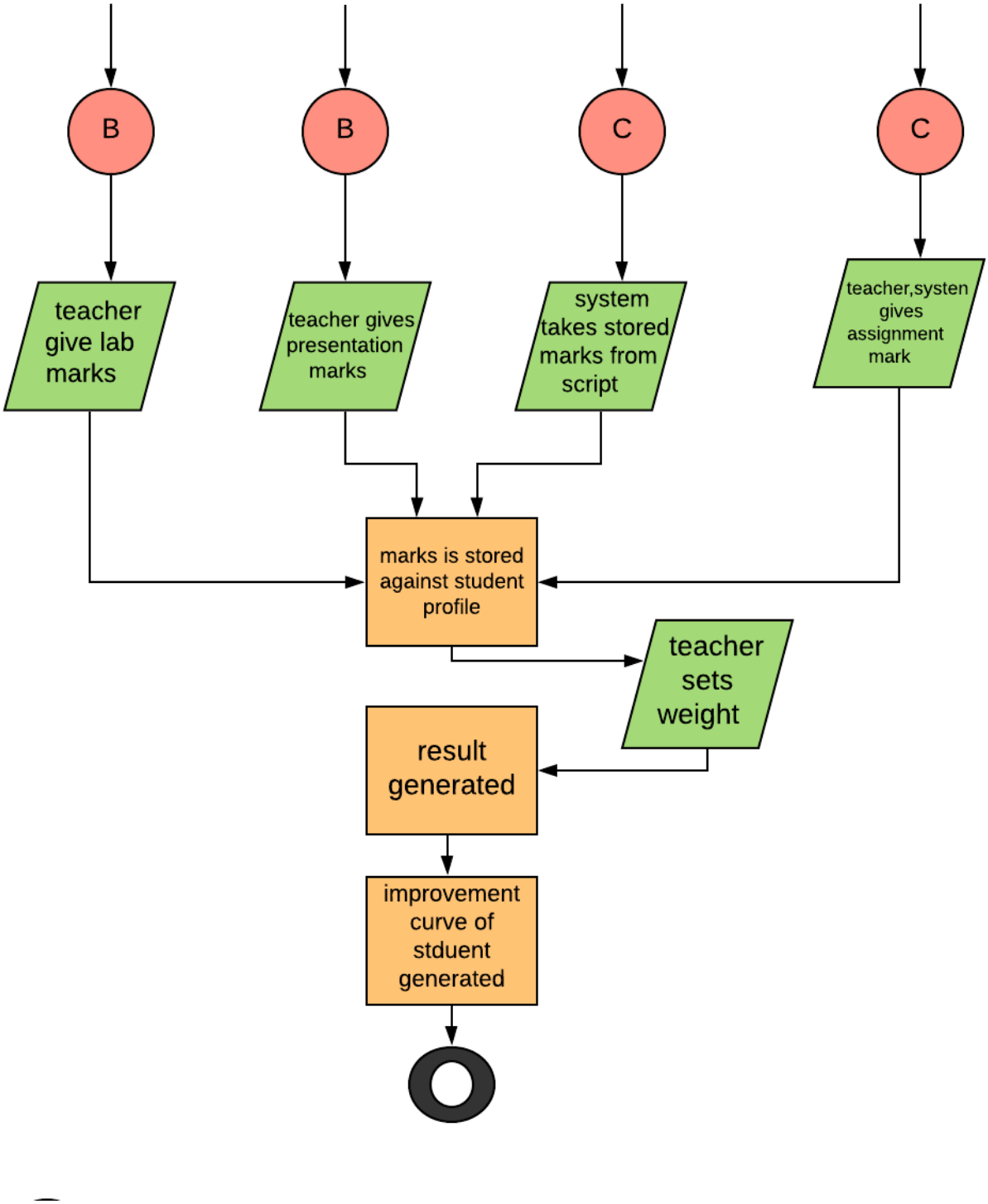


Fig 30: Store & Generate result

4.5 Swimlane Diagram Of CMS

Authentication

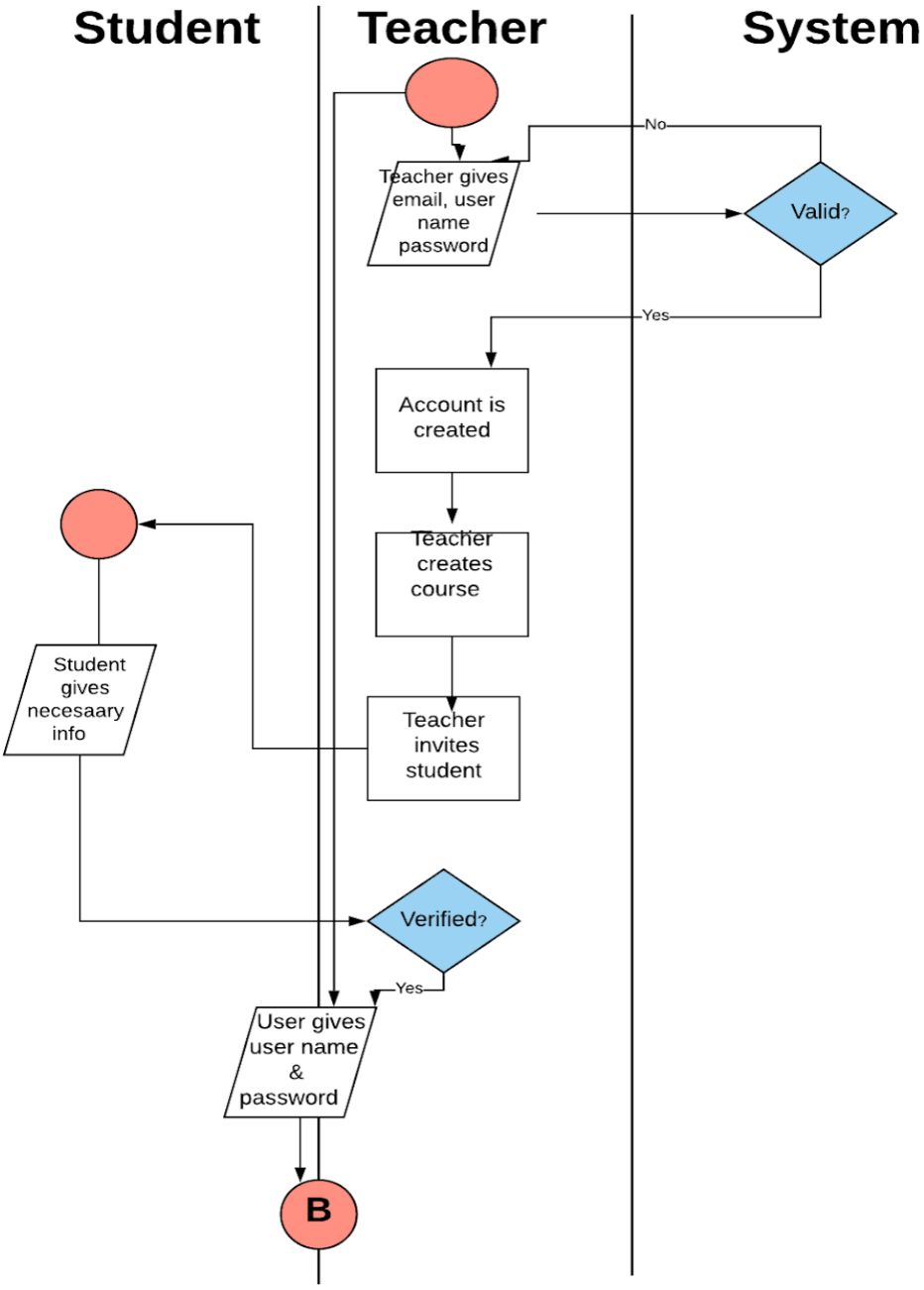


Figure 31: Swimlane diagram for Registration and login

Description: Figure 31 shows the swimlane diagram for Registration process and login

Online Attendance

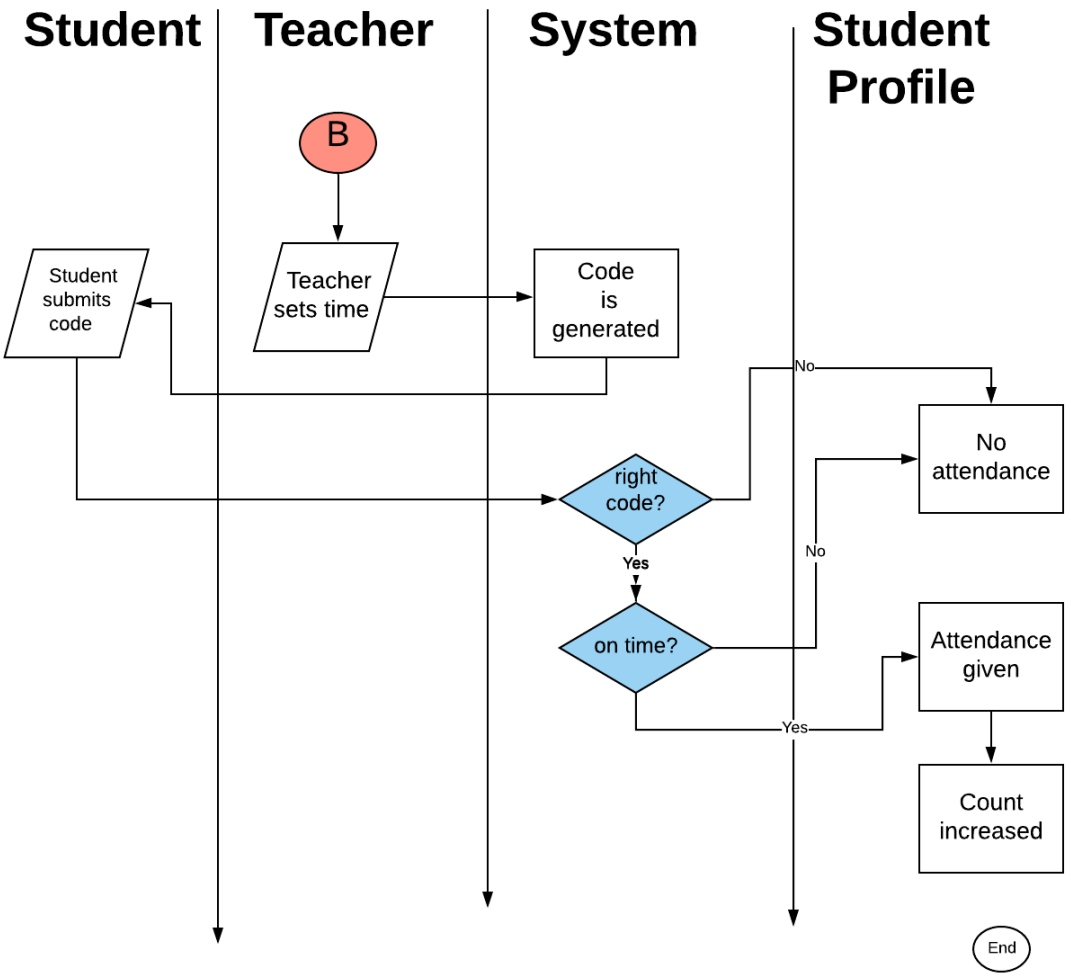


Figure 32: Swimlane diagram for Online Attendance

Description: Figure 32 shows the swimlane diagram for Online Attendance process.

Quiz

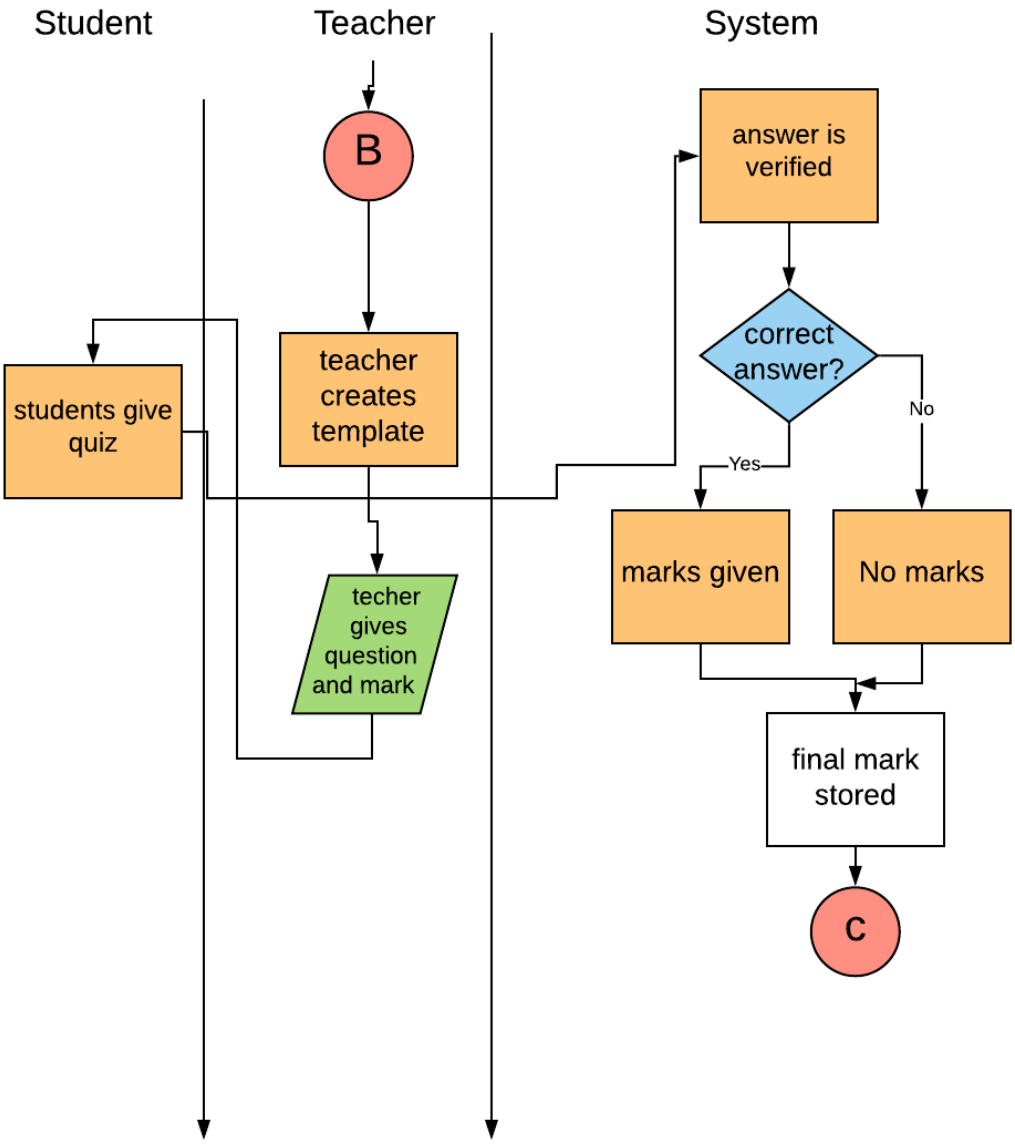


Figure 33: Swimlane diagram for Online Quiz

Description : Figure 33 shows the swimlane diagram for Online Quiz

Script

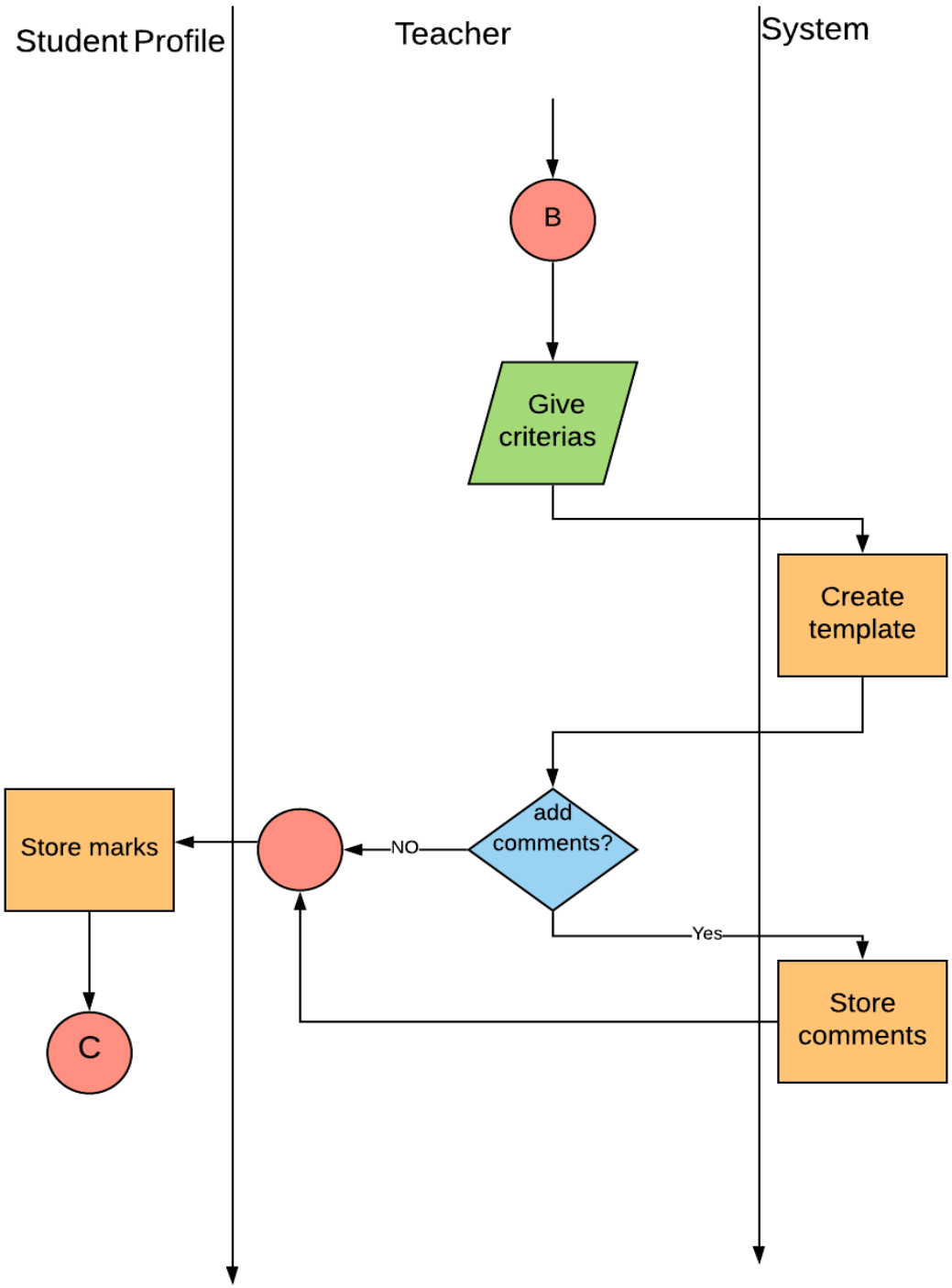


Figure 34: Swimlane diagram for Script Viewing

Description : Figure 34 shows the swimlane diagram for Script Viewing

Assignment

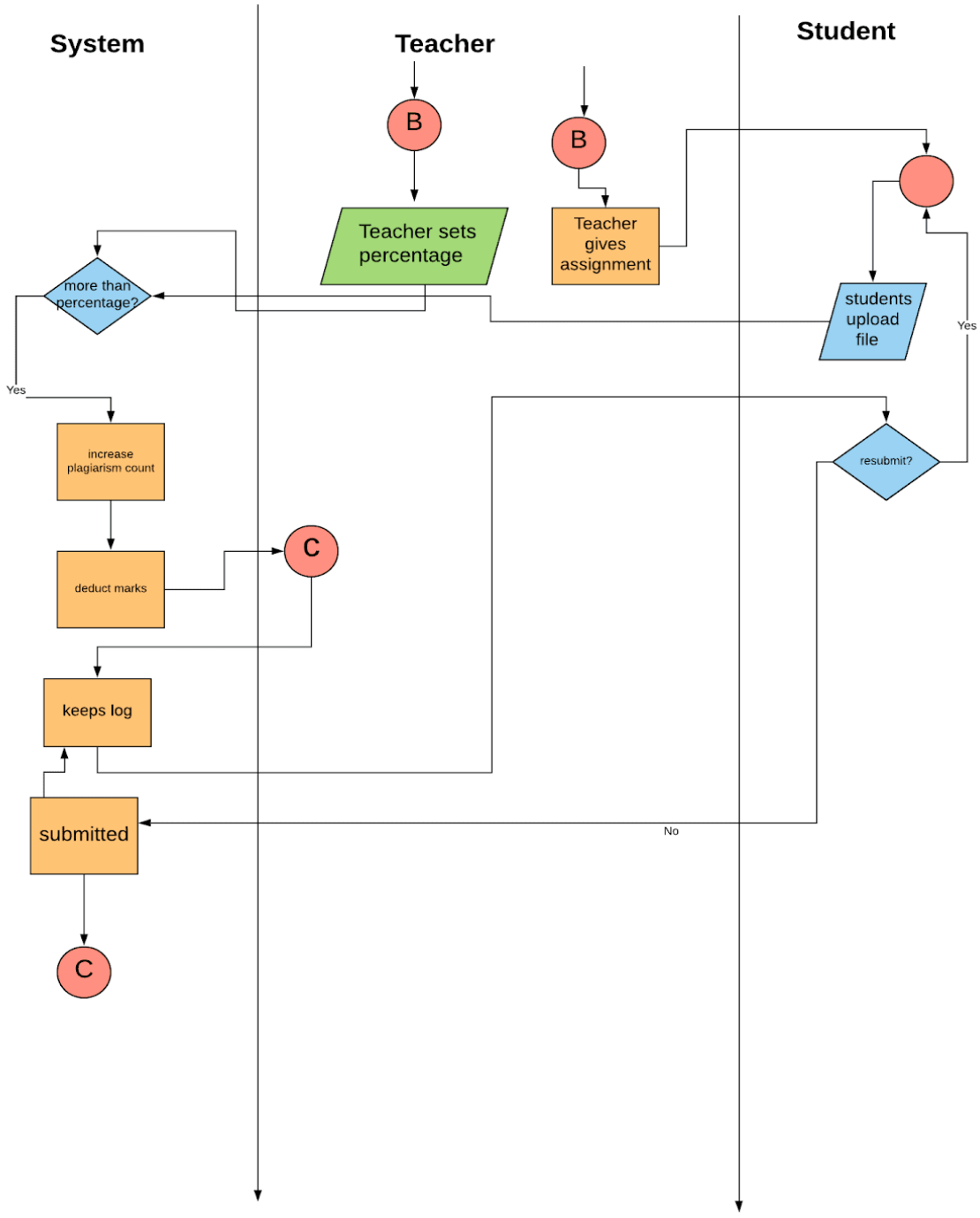


Figure 34: Swimlane diagram for Assignment

Description : Figure 34 shows the swimlane diagram for Assignment

Download

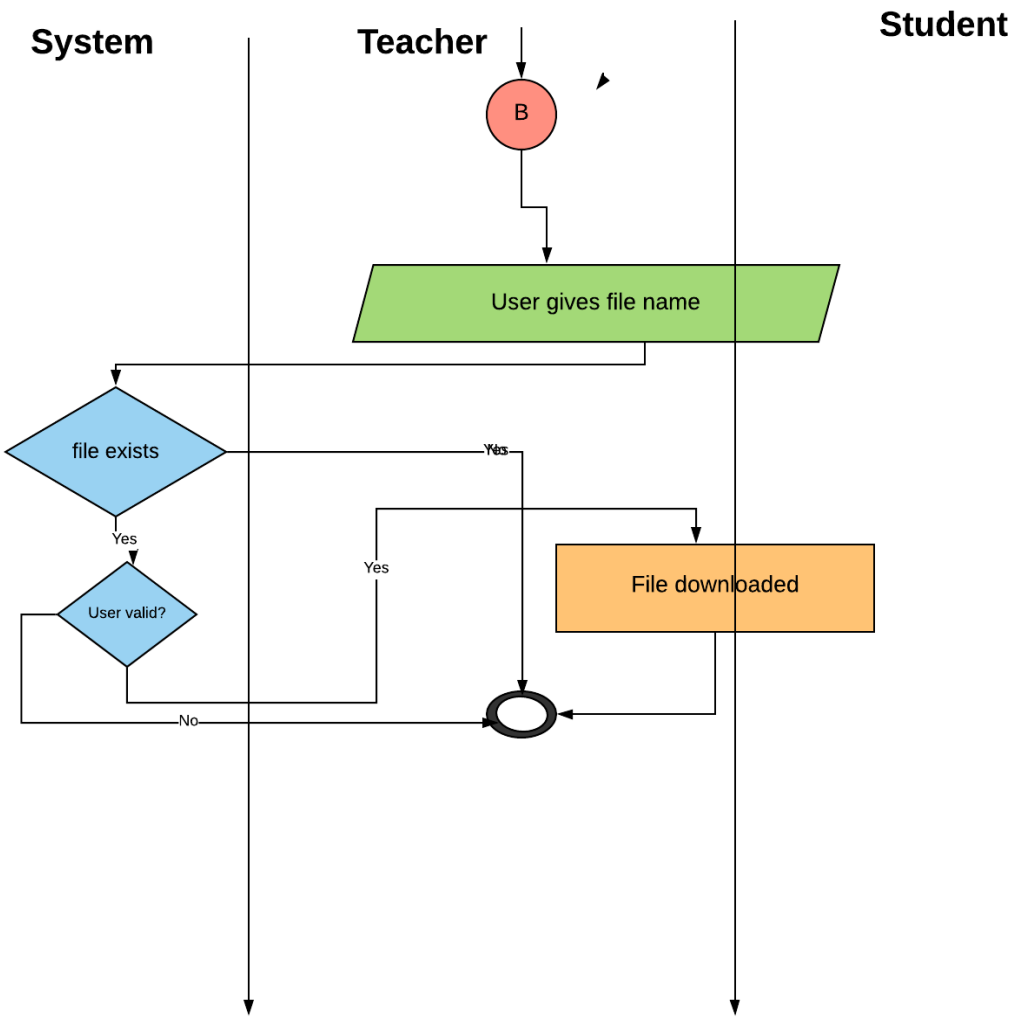


Figure 36: Swimlane diagram for Download

Description : Figure 36 shows the swimlane diagram for Download

Upload

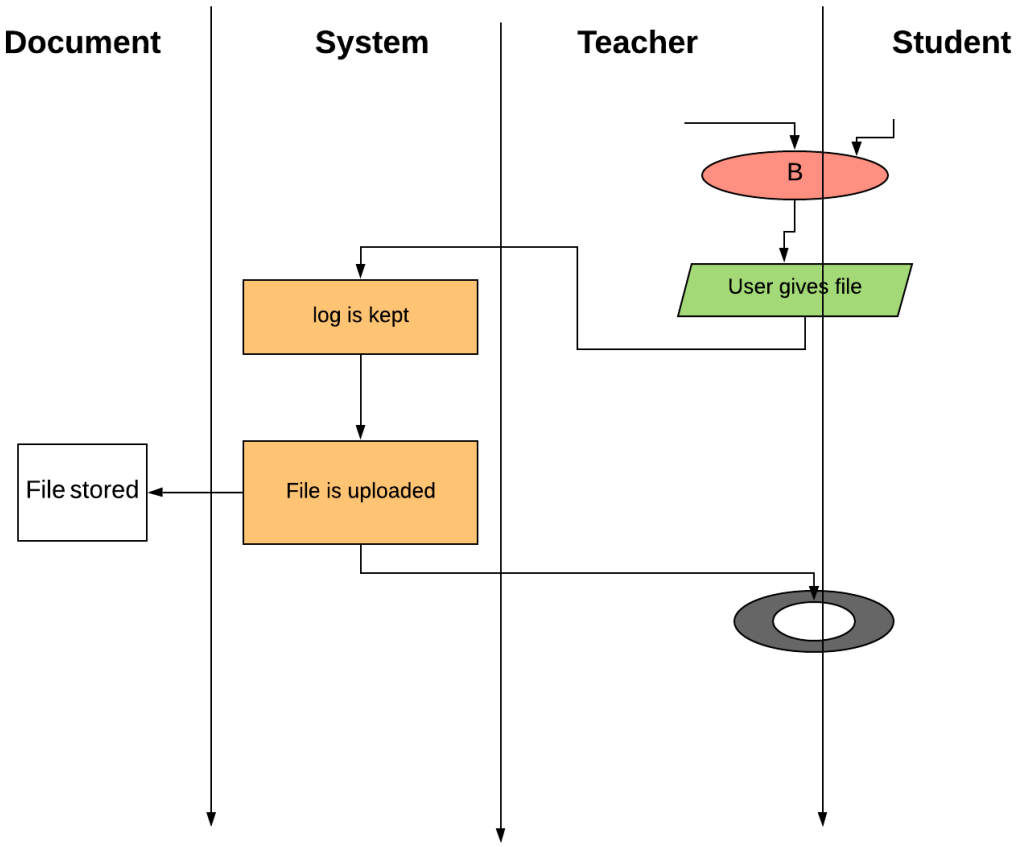


Figure 37: Swimlane diagram for upload

Description : Figure 37 shows the swimlane diagram for Upload

Forum

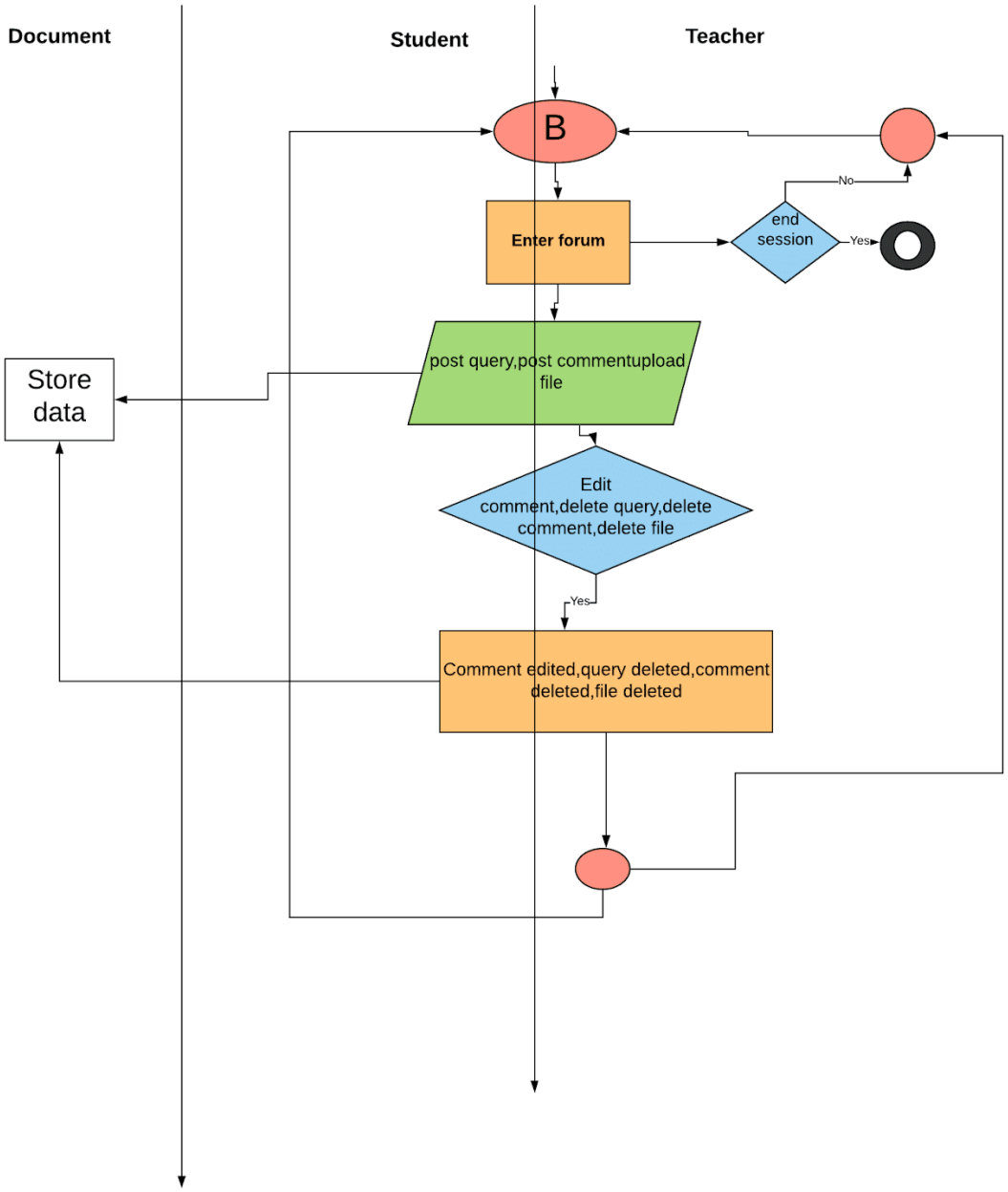


Figure 38: Swimlane diagram for F0rum

Description : Figure 38 shows the swimlane diagram for Forum

Result

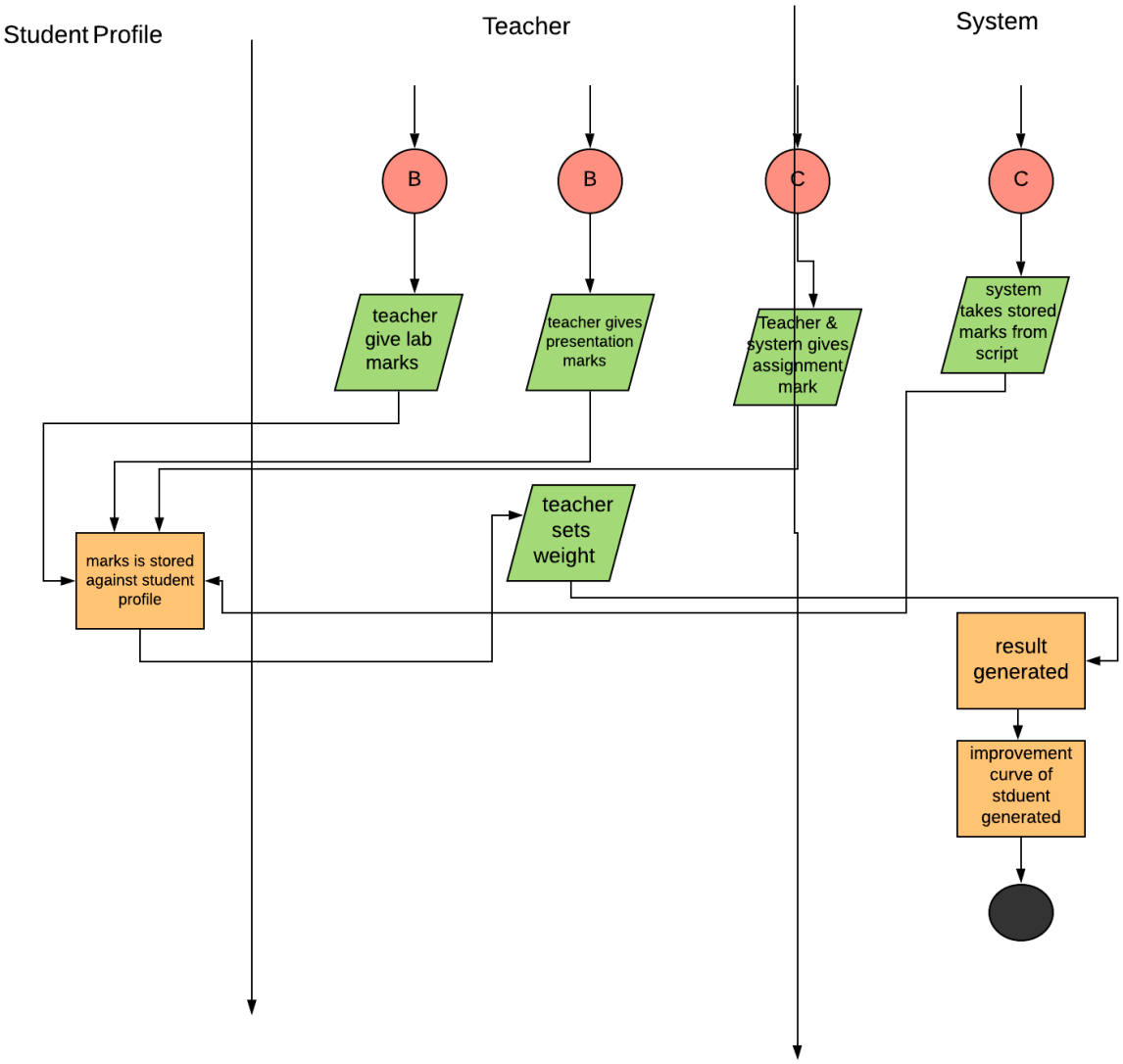


Figure 39: Swimlane diagram for Result

Description : Figure 39 shows the swimlane diagram for result generation

Course Curve

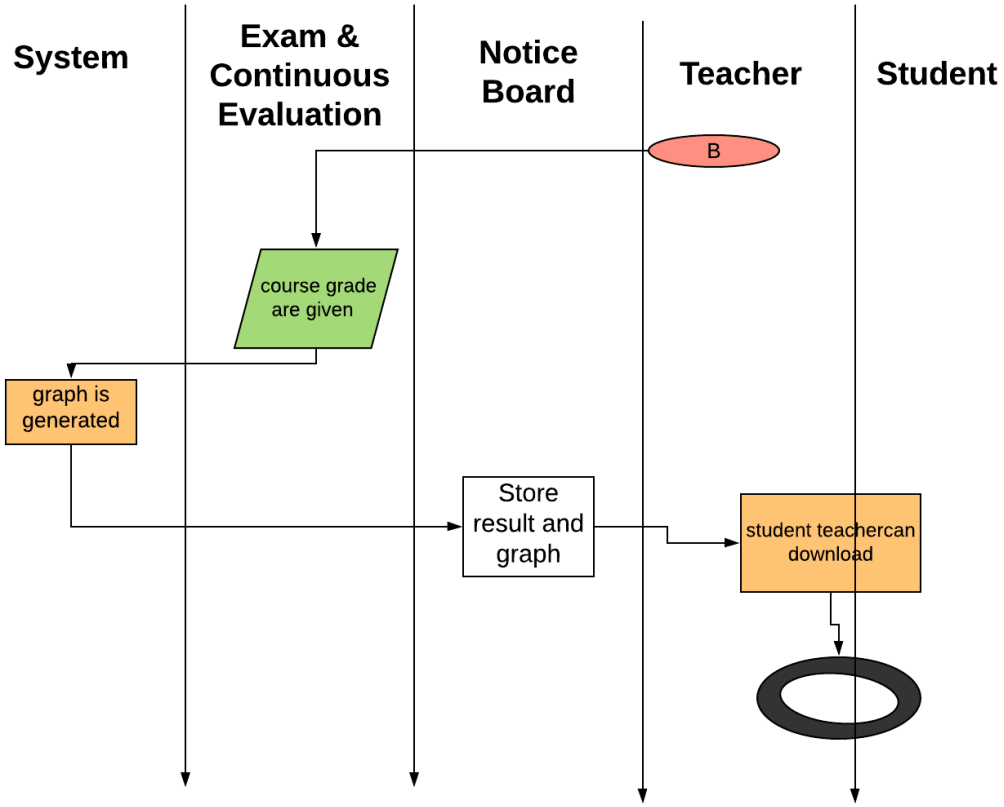


Figure 40: Swimlane diagram for Course Curve

Description : Figure 40 shows the swimlane diagram for Course Curve

5 DATA BASED MODELING OF GMS

5.1 DATA MODELING CONCEPT

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models 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 about how the data objects are entered, stored, transformed and produced within the system.

5.1.1 DATA OBJECTS

A data object is a representation of composite information that must be understood by the software. Here, composite information mean an information 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.

5.1.1.1 Noun identification

We identified all the nouns from our elicitation story. The noun list are given below

Table 1 : Noun List

serial	Noun	Attributes
1	System	x
2	Course	3,4,16,17,18,52
3	Teacher	2,6,8,9,11,12,13,14,15
4	Student	6,7,8,9,11,12,13,14,33

5	features	x
6	Email address	x
7	Class roll	x
8	User name	x
9	Password	x
10	Credentials	x
11	name	x
12	Registration number	x
13	Department name	x
14	year	x
15	Designation	x
16	Course name	x
17	Schedule	x
18	Outline	x
19	Forum	x
20	Query	8,60,62
21	Problem	x
22	Discussion	8.60,62
23	Comment	8,60,62
24	file	8,11,24,60,61,63
25	document	8,11,24,60,61,63
26	Class lecture	8,11,24,60,61,63
27	Attendance	8,28,29
28	code	x
29	date	x
30	Dash board	x
31	Class record	x
32	presents	x
33	profile	8,25,32,41,43

34	marks	x
35	exam	x
36	midterm	8,34,49
37	lab	8,34,49
38	quiz	8,34,44
39	final	8,34,49
40	script	8,34,49
41	Improvement curve	x
42	criteria	x
43	Student log	x
44	question	x
45	presentation	8,34,49
46	assignment	8,34,47,48
47	plagiarism	x
48	Resubmission count	x
49	template	42,23
50	result	8,51
51	grade	x
52	Bar diagram	x
53	Notice board	8,54,55,56,57,58
54	notice	x
55	topic	x
56	date	x
57	Place	x
58	time	x
59	notification	x
60	type	x
61	size	x
62	post(status string)	x

63	id	x

5.1.1.2 Selected initial data objects

Table 3: Initial Data Objects

serial	Noun	Attributes
1	Course	3,4,16,17,18,52,63
2	Teacher	2,6,8,9,11,12,13,14,15
3	Student	6,7,8,9,11,12,13,14,33
4	Query	8,60,62,63
5	Discussion	8.60,62,63
6	Comment	8,60,62,63
7	file	8,11,24,60,61,63
8	document	8,11,24,60,61,63
9	Class lecture	8,11,24,60,61,63
10	Attendance	8,28,29
11	profile	8,22,32,41,43,63
12	midterm	8,34,49,63
13	lab	8,34,49,63
14	quiz	8,34,44,63
15	final	8,34,49,63
16	script	8,34,49,63
17	presentation	8,34,49
18	assignment	8,25,34,47,48,63
19	template	42,23,63
20	result	8,33,51
21	Notice board	2,54,55,56,57,58

5.1.1.2 ANALYSIS

- Here query,discussion,comment are identically equal. So we are going to merge them and create posts
- File , document and class lecture are identically same. So, we are going to merge them and create document
- Midterm,presentation,lab,final and script are identically equal. So merging them we have created script

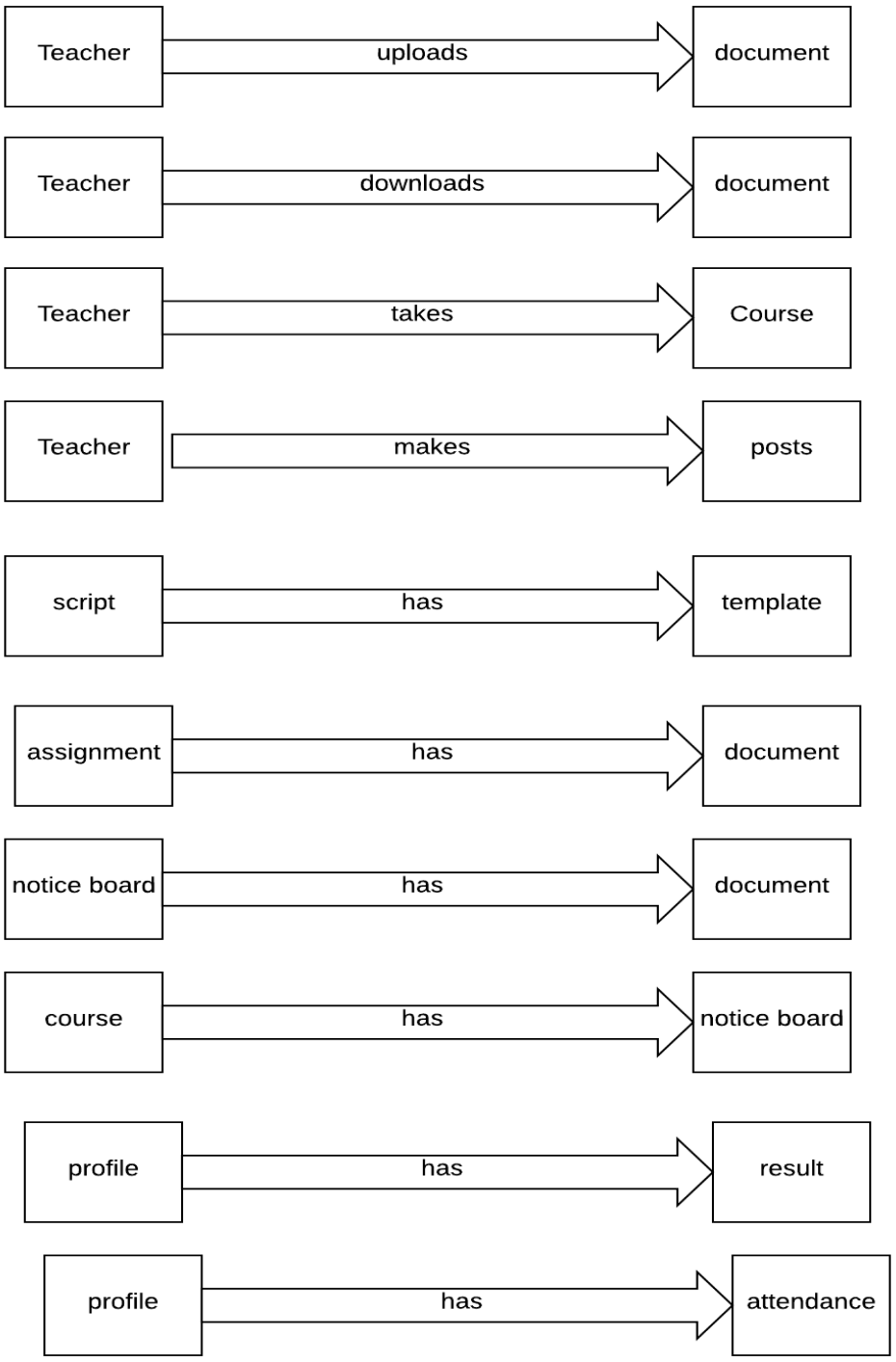
5.1.1.3 Final Data Objects

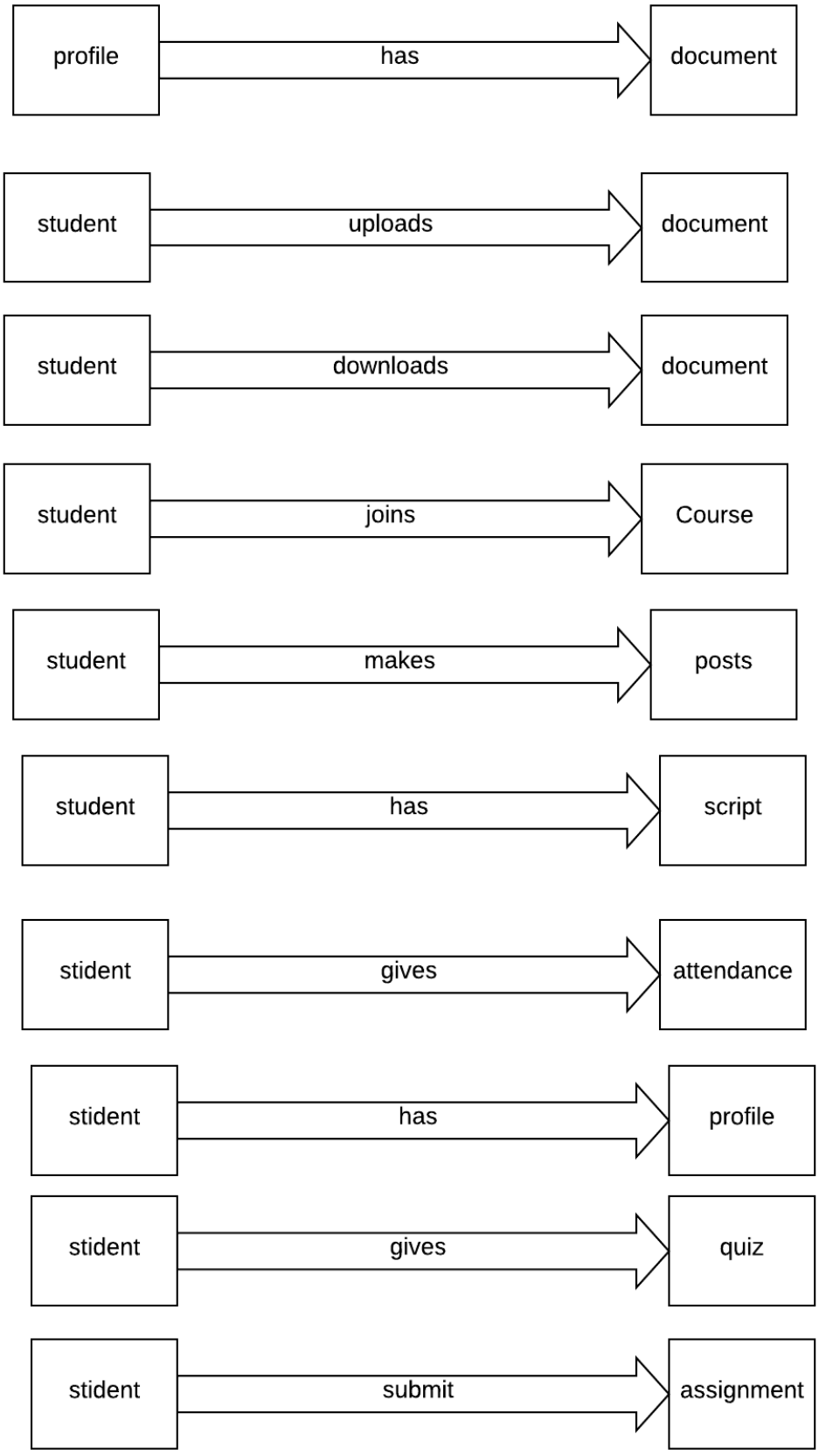
Table 2: Final Data Objects

serial	Noun	Attributes
1	Course	2,3,16,17,18
2	Teacher	2,6,8,9,11,12,13,14,15
3	Student	6,7,8,9,11,12,13,14,33
4	posts	8,60,62,63
5	document	8,11,24,60,61,63
6	Attendance	8,28,29
7	profile	8,22,32,41,43,63
8	quiz	8,34,44,63
9	script	8,34,49,63
10	assignment	8,25,34,47,48,63
11	template	42,23,63
12	result	8,51,33
13	Notice board	2,54,55,56,57,58

5.2 Data Object Relationship

Data objects are connected to one another in different ways.




profile
has
document
student
uploads
document
student
downloads
document
student
joins
Course
student
makes
posts
student
has
script
stident
gives
attendance
stident
has
profile
stident
gives
quiz
stident
submit
assignment

5.3 ER diagram

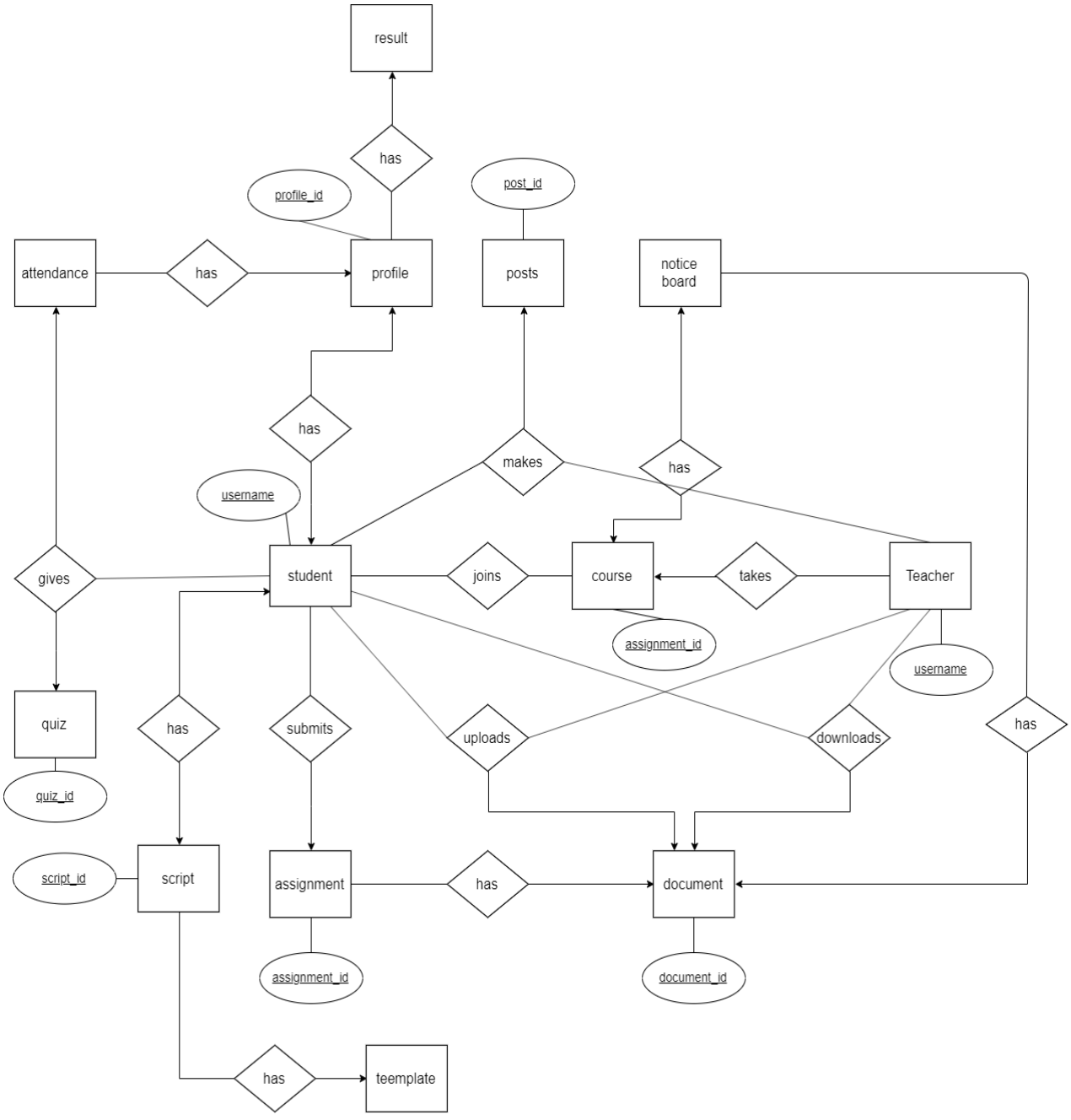


Fig 41: Er Diagram CMS

5.3 Schema Diagram

Table 34: Teacher Schema Diagram

Teacher		
Attributes	Type	Size
name	varchar	20
Email address	Varchar	30
User name	Varchar	30
password	Varchar	30
Department name	Varchar	20
Registration number	Varchar	40
Year	Varchar	20
Designation	Varchar	20
Course ID	Number	NULL

Table 5: Student Schema Diagram

Student		
Attributes	Type	Size
name	varchar	20
Class roll	varchar	10
Email address	Varchar	30
User name	Varchar	30
password	Varchar	30
Department name	Varchar	20
Registration number	Varchar	40
Year	Varchar	20
Profile ID	Number	NULL

Table 6: Course Schema Diagram

Course		
Attributes	Type	Size
Teacher name	varchar	20
Student name	varchar	20
Course name	Varchar	20

Schedule	Varchar	300
Outline	Varchar	200
Id	Number	NULL

Table 7: Document Schema Diagram

Document		
Attributes	Type	Size
Id	Number	NULL
User name	Varchar	20
file	File	NULL
name	Varchar	20
size	Number	NULL
type	Varchar	20

Table 8: Online Attendance Schema Diagram

Attendance		
Attributes	Type	Size
Username	Varchar	20
date	Date	NULL
Code	Varchar	20

Table 9: Student Profile Schema Diagram

Profile		
Attributes	Type	Size
User name	varchar	20
Profile id	Number	NULL
Document id	Number	NULL
Improvement curve	File	NULL
Student log	Varchar	100

Table 11: Assignment Schema Diagram

Assignment		
Attributes	Type	Size
Assignment id	Number	NULL
Document id	Number	NULL
User name	Varchar	20
marks	Number	NULL
plagiarism	Number	NULL
resubmission	Number	NULL

Table 12: Quiz Schema Diagram

Quiz		
Attributes	Type	Size
Quiz id	Number	NULL
Username	varchar	20
questions	varchar	1000
marks	Number	NULL

Table 13: Script Schema Diagram

script		
attribute	type	size
User name	Varchar	20
Template id	Number	NULL
marks	Number	NULL
Script id	Number	NULL

Table 14: Notice Board Schema Diagram

Notice Board		
attributes	Type	size
Course id	Number	NULL
notice	Varchar	100
topic	Varchar	20
Document id	Number	NULL
date	Date	NULL
Place	Varchar	NULL
time	Time	NULL

Table 15: Result Schema Diagram

result		
attribute	type	size
User name	Varchar	20
Profile id	Number	NULL
grade	varchar	2

Table 17: Post Schema Diagram

posts		
Attributes	Type	Size
id	Number	NULL
User name	Varchar	20
type	Varchar	20
size	Number	NULL
post(status string)	Varchar	1000

Table 18: Post Schema Template

Template		
Attribute	Type	Size
Template id	Number	NULL
Criteria	Varchar	100
comments	varchar	100

6 CLASS-BASED MODELING FOR CMS

In this part we have discussed about the classes along with their attributes and methods of our **COURSE MANAGEMENTS SYSTEM**

6.1 CLASS BASED MODELING CONCEPT

Class-based Modeling represents the object. The system manipulates the operations. The elements of the class based model consist of classes and objects, attributes, operations, **class** – responsibility - collaborator (CRS) models.

6.2 GENERAL CLASSIFICATION

To identify the potential classes, we have first selected the nouns from the solution space of the story. We have used seven general characteristics for this. The seven general characteristics are as follows :

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

Table 19: General Classification

serial	Noun	Attributes
1	System	3,4,7
2	Course	3,5
3	Course Teacher	4,5
4	Student	4,5
5	features	x
6	Email address	x
7	Class roll	x

8	User name	x
9	Password	x
10	Credentials	4,7
11	name	x
12	Registration number	x
13	Department name	x
14	year	x
15	Designation	x
16	Course name	x
17	Schedule	x
18	Outline	x
19	Forum	2,3,7
20	Query	2,3
21	Problem	x
22	Discussion	2,3
23	Comment	2,3
24	file	1,2,7
25	document	1,2,7
26	Class lecture	1,2,7
27	Online Attendance	3,4
28	code	x
29	date	x
30	Dash board	x
31	Class record	x
32	presents	x
33	profile	2,3,4,7
34	marks	x
35	exam	2,3,7
36	midterm	x

37	lab	x
38	quiz	3,7
39	final	x
40	script	2,3,7
41	Improvement curve	2
42	criteria	x
43	Student log	x
44	question	x
45	presentation	x
46	assignment	2,3,7
47	plagiarism	2,3
48	Resubmission count	x
49	template	2
50	result	2,3
51	grade	x
52	Bar diagram	x
53	Notice board	2,3,7
54	notice	x
55	topic	x
56	date	x
57	Place	x
58	time	x
59	notification	x
60	type	x
61	size	x
62	post(status string)	x
63	id	x

6.3 SELECTION CRITERIA

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

- Retain information
- Needed services
- Multiple attributes
- Common attributes
- Common operations
- Essential requirements

Table 20: Selection Criteria

NO	Noun	Selection Criteria
1	System	1,2,3,4,5
2	Course	1,2,3,4,5
3	Course Teacher	2,3,4,5
4	Student	2,3,4,5
5	Credentials	1
6	Forum	2,3
7	Query	1
8	Dash board	2
9	Online Attendance	1,2,3
10	Student Profile	1,3,4
11	Exam	1,2,3,4,5
12	Assignment	1,2,4,5
13	Document	2,3,4,6
14	Notice Board	1,2,4,5
15	Script	1,2,3,4,5

16	**Quiz**	**1,2,3,4**
17	**Plagiarism Check**	**1,3,4,5**
18	**Result**	**1,2,3,6**

6.4 Associate Noun and Verb identification

We will now identify the nouns and verbs associated with the potential classes to find out the attributes and methods of each class :

Table 21: Noun Verb Identification

No.	Potential Class	Noun	Verb
1.	System	Log, store, retrieve, Template,course, course teacher, student profile	Keeping log,store uploaded documents, Retrieving marks, create template, Verifying data
2	Course	Course name,course code,course teacher,student, schedule, outline, improvement curve,bar diagram, document,folder	Requesting to resubmit info in failure, creating folder, uploading files, downloading files, generate course evaluation
3	Course teacher	name,Email address, username,password ,name,designation	Create account, login, creating course,entering marks,set template, Inviting students, Adding student, Removing student,

			logout
4	Student	Email address,roll, Username,password, Name,roll,department, year/semester.	Registering, login, providing credentials, submit assignments ,create profile, logout
5	Forum	Query,comment, search, keyword	Query,discussions, search by keywords, posting comments , deleting comments, replying comments
6	Online Attendance	schedule,code, dashboard	Generating automated code, opening dashboard, set fixed time,count present
7	Student profile	Attendance,marks, improvement curve,log	Store marks, create improvement curves, retrieve attendance, search log
8	Exam & continuous evaluation	Quiz,assignment, presentation,lab, script	Get quiz marks ,get presentation marks. get assignment marks, Get lab marks, get script set weights. generate continuous evolution
9	Assign-	Name,	Opening dashboard,

	ment	dashboard, time, resubmission, counts	set time,uploading files, get plagiarism result, resubmit assignment if needed,keep submission count,
10	Document	name, type, size	Get document from system, store document in system, remove document from system
11	Quiz	Fix time,template question,marks,script	Set fix time, set digital MCQ paper, set marks distribution,check quiz,store marks
12	scripts	Midterm,final,criteria, comments.,template	Set template,comment if needed,store marks against profile
13	Notice Board	Notice,file,notification.	Adding file notice, creating notice,giving topic, giving date, giving place,notify users on a fixed basis.
14	Result	marks,grade	get_marks,generate_ result,store_result_

			against_profile
15	Plagiarism check	Percentage, count, document	Check plagiarism, set percentage, keep plagiarism count, marks deduction

6.5 Identify Attributes

We have identified our potential classes, So, now we have identified the classes

Table 22: Identify Attributes

No.	Potential Class	Attributes
1.	System	Template, Profile, document, dashboard
2	Course	Course name, course id, course teacher, Student, schedule, Outline, Document, bar diagram
3	Course teacher	Name, Email address, Username, Password, Designation, Document,

4	Student	Email address, roll, Username, password, Name, department, year/semester. Student profile, Document
5	Forum	Student, Course teacher, Query, comment, keyword, post id
6	Online Attendance	Student, Code, Dashboard, Date
7	Student profile	Online Attendance, Exam & continuous evaluation, improvement curve, log, Presents, Student, Document
8	Exam & continuous evaluation	Student, Student profile, Quiz, lab, Assignment, script

9	Assignment	Name, Dashboard, time, resubmission counts, Assignment id, Documents, Plagiarism Checker
10	Document	Document id, Student, Course Teacher Name, Type,, Size,
11	Quiz	Quiz id, time, template, Question, Marks
12	scripts	Script id, Midterm, Final, criteria, Comments, template
13	Notice Board	Notice, File, Topic, Time, Date,.
14	Result	Exam & continuous evaluation, Grade, Script, Student profile,

		Student
15	Plagiarism check	Student profile, Assignment id, Percentage, Count, document

6.6 Identify Methods of potential Classes

Now , we are going to identify the methods of our potential classes

Table 23: Identify Methods

NO	Noun	Method
1.	System	Keep_log(), Handling_documents(), create_template(), Verify_data()
2	Course	resubmit_in_failure(), check_code(), create_folder(), upload_files(), download_files(), generate_bar_diagram()
3	Course teacher	Register(), login(), invite_students(), add_student(), remove_student, create_course(), enter_marks(), set_template(),

		logout()
4	Student	Register(), login(), provide_credentials(), create_profile(), logout()
5	Forum	Query(), discussions(), search_by_keyword(), comments(), upload_files(), delete_comments(), reply_to_comments()
6	Online Attendance	generate_code(), open_dashboard(), set_fixed_time(), count_present()
7	Student profile	Store_marks(), create_improvement_curves(), retrieve_attendance(), keep_log_of_file(); search_log()
8	Exam & continuous evaluation	Get_quiz_marks(), get_presentation_marks(). get_assignment_marks(), Get_lab_marks(), get_script(), set_weights(), continuous_evolution(), store_in_profile()
9	Assignment	Open_dashboard(), set_time(), get_plagiarism_result(),

		upload_files(), resubmit_assignment(), keep_submission_count(),
10	Document	Get_document_from_system(), store document in system(), remove_doc_from_system(),
11	Quiz	Set_time(), digital_MCQ_paper(), set_marks_distribution(), check_quiz(), store_marks()
12	Script	Set_template(), commentIFneeded(), store_marks_in_profile()
13	Notice Board	Add_file_notice(), create_notice(), give_topic(), givie_date(), give_place(), notify_users()
14	Result	get_marks(), generate_result(), store_result_against_profile()
15	Plagiarism checker	set_percentage(), Check_plagiarism(), keep_plagiarism_count(), marks_deduction()

6.7 Analysis

To find out our final classes, we need to carry out some analysis.

- **System** :

System class is needed for performing automated tasks & user interface.

- **Course** :

Course class will hold most of the necessary functionalities and services which will be controlled by course teacher and used by both teacher and students.

- **Course teacher** :

Though course teacher and student have some common attributes & functionalities, we have kept them as two separate classes for clarifying the story more clearly. Course teacher will have the authority to maintain the overall course.

- **Student** :

Student is a vital user of our Course Management System. Student has many distinguishable functionalities, so it has been kept as a separate class.

- **Forum** :

Forum class will contain the following attributes student,course teacher,query,comment,keyword. Teacher and students can discuss about any topic or add query. Both users can contribute here by commenting and adding files.

- **Online attendance** :

As we have generated an automated attendance system, it will perform some specific tasks such as generating code, taking attendance, verifying codes given by students, storing attendance, so we have kept it as a separate class.

- **Student profile** :

This class will store all the information of a student. System will retrieve all necessary information from student profile and generate results and bar diagram.

- **Exam & continuous evaluation** :

Most of the continuous evaluation parts and exams can be merged as a single class “Exam & continuous evaluation” as they perform some common functionalities.

- **Assignment** :

Assignment has been kept as a class as it needs some distinct tasks to be performed such as submission, resubmission, plagiarism checking, keeping log.

- **Document** :

All the files of this system will be stored in document class. Users will upload, download and view files from document class.

- **Quiz** :

It has been kept as a class because it needs some manual & automated functionalities that are different from other continuous evaluation parts.

- **Script** :

It has been kept as a class because it needs some manual & automated functionalities that are different from other continuous evaluation parts.

- **Notice Board** :

This class will hold the features for creating notice & notifying the users. Teacher will manipulate it with the help of system.

- **Result** :

This class will get marks from student profile, assist the system to generate result automatically and store it against each profile.

- **Plagiarism check** :

This class will verify the assignments using all the previously submitted assignments. Teacher will set necessary criteria and system will check automatically using those criteria.

6.8 Class cards

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

Table 24: System Class card

System	
Attributes	Methods
• Template, • Student Profile, • document, • dashboard	• Keep_log(), • Handling_documents(), • create_template(), • Verify_data()
Responsibility	Collaborator
1. Keeping log, 2. store uploaded documents, 3. Retrieving marks, 4. create template	System, Student, Course teacher, Forum, Student profile, result

Table 25: Course Class card

course	
Attributes	methods
• Course name, • course id, • course teacher, • Student, • schedule, • Outline,	• resubmit_in_failure(), • check_code(), • create_folder(), • upload_files(), • download_files(), • generate_bar_diagram()

• Document, • bar diagram	
Responsibility	Collaborator
1. Inviting students, 2. checking codes, 3. adding student, 4. requesting to resubmit info in failure, 5. creating folder, 6. uploading files, 7. downloading files, 8. generate course evaluation	Course, Student, Course teacher, System, Student profile

Table 26: Course Teacher Class card

Course Teacher	
Attributes	Methods
• Name, • Email address, • Username, • Password, • Designation, • Document,	• resubmit_in_failure(), • check_code(), • create_folder(), • upload_files(), • download_files(), • generate_bar_diagram() • logout
Responsibility	Collaborator
1. Registering, 2. creating course, 3. entering marks, 4. set template	Course teacher, system

Table 27: Student Class card

Student	
Attributes	Method
• Email address, • roll, • Username, • password, • Name, • department, • year/semester. • Student profile, • Document	• Register(), • login(), • provide_credentials(), • create_profile(), • logout()
Responsibility	Collaborator
1. Registering, 2. providing credentials, 3. submit assignments, 4. create profile	Student, Student profile, Assignments

Table 28: Forum Class Card

Forum	
Attributes	Method
• Student, • Course teacher, • Query, • comment, • keyword • post id	• Query(), • discussions(), • search_by_keyword(), • comments(), • upload_files(), • delete_comments(), • reply_to_comments()
Responsibility	Collaborator

1. Query,discussions, 2. adding comments, 3. uploading files, 4. deleting comments, 5. replying comments	Forum Student Course teacher Course System Student profile

Table 29: Online Attendance Class card

Online Attendance	
Attributes	Methods
• Schedule, • Code, • Dashboard, • count	• generate_code(), • open_dashboard(), • set_fixed_time(), • count_present()
Responsibility	Collaborator
1. Generating automated code, 2. opening dashboard, 3. set fixed time, 4. count present	Online Attendance System, Student, Course teacher, Result, Student profile

Table 30: Student Profile Class card

Student profile	
Attributes	Methods
• Online Attendance,	• Store_marks(), • •create_improvement_curves(),

• Exam & continuous evaluation, • improvement curve, • log, • Presents, • Student, • Document	• retrieve_attendance(), • keep_log_of_file(); • search_log()
Responsibility	Collaborator
1. Store marks, 2. create improvement curves, 3. retrieve attendance, 4. search log	Student profile Exams & continuous evaluation, Student profile, Student, Course teacher, System Online attendance result

Table 31: Exam & Continuous evaluation Class Card

Exam & continuous evaluation	
Attributes	Methods
• Student, • Student profile, • Quiz, • lab, • Assignment, • script	• Get_quiz_marks(), • get_presentation_marks(). • get_assignment_marks(), • Get_lab_marks(), • get_script(), • set_weights(), • continuous_evolution(), • store_in_profile()
Responsibility	Collaborator

1. Get quiz marks, 2. get midterm marks, 3. get presentation marks. 4. get assignment marks, 5. Get lab marks, 6. get final marks 7. generate result, 8. generate grade, 9. set weights	Quiz, Script, Student, Course Teacher, Online attendance Exams & continuous evaluation, Assignment

Table 32: Assignment Class Card

Assignment	
Attributes	Methods
• Name, • Dashboard, • time, • resubmission counts, • Assignment id, • Documents, • Plagiarism Checker	• Open_dashboard(), • set_time(), • get_plagiarism_result(), • upload_files(), • resubmit_assignment(), • keep_submission_count(),
Responsibility	Collaborator
1. Opening dashboard, 2. set time, 3. uploading files, 4. resubmit assignments if needed, 5. keep submission count 6. check plagiarism, 7. keep plagiarism count	Assignment Student, System

Table 33: Document Class Card

Document	
Attributes	Methods
• Document id, • Student, • Course Teacher • Name, • Type, • Size,	• Get_docu_from_system(), • store_docu_in_system(), •remove_doc_from_system()
Responsibility	Collaborator
1. Get document from 2. system, store document in system, 3. remove document from system	Student, Course teacher, tuden

Table 34: Quiz Class Card

Quiz	
Attributes	Methods
• Quiz id, • time, • template, • Question, • Marks	• Set_time(), • digital_MCQ_paper(), • set_marks_distribution(), • check_quiz(), • store_marks()
Responsibility	Collaborator

1. Set fix time, 2. set digital MCQ paper, 3. set marks distribution, 4. check quiz,store marks	Quiz Course teacher, Student, Course teacher, Student profile Exam & continuous evaluation

Table 35: Script Class Card

Script	
Attributes	Methods
• Script id, • Midterm, • Final, • criteria, • Comments, • template	• Set_template(), • commentIFneeded(), • store_marks_in_profile()
Responsibility	Collaborator
1. Set template, 2. comment if needed, 3. store marks against profile	Script Course teacher, Student profile, Student System Exam & continuous evaluation Result

Table 36: Notice Board Class Card

Notice Board	
Attributes	Methods
• Notice, • File,	• Add_file_notice(), • create_notice(),

• Topic, • Time, • Date,.	• give_topic(), • givie_date(), • give_place(), • notify_users()
Responsibility	Collaborator
1. Adding file notice, 2. creating notice,giving topic, 3. giving date, 4. giving place, 5. notify users on a fixed basis.	Notice Board Course, Course Teacher, Student

Table 37: Result Class card

Result	
Attributes	Methods
• Exam & continuous evaluation, • Grade, • Script, • Student profile, • Student	• get_marks(), • generate_result(), • store_result_against_profile()
Responsibility	Collaborator
1. Get marks, 2. Generate result, 3. Store result against profile	Result System, Exams & continuous evaluation Quiz Script Course, Course Teacher,

	Student , Student Profile, Assignment

Table 38: Plagiarism Checker Class Card

Plagiarism Checker	
Attributes	**Methods**
• Student profile, • Assignment id, • Percentage, • Count, • document	• set_percentage(), • Check_plagiarism(), • keep_plagiarism_count(), • marks_deduction()
Responsibilities	**Collaborator**
1. Check plagiarism, 2. set percentage, 3. keep plagiarism count, 4. marks deduction	System, Document, Student profie, Course teacher,

6.9 Class Modeling Diagram

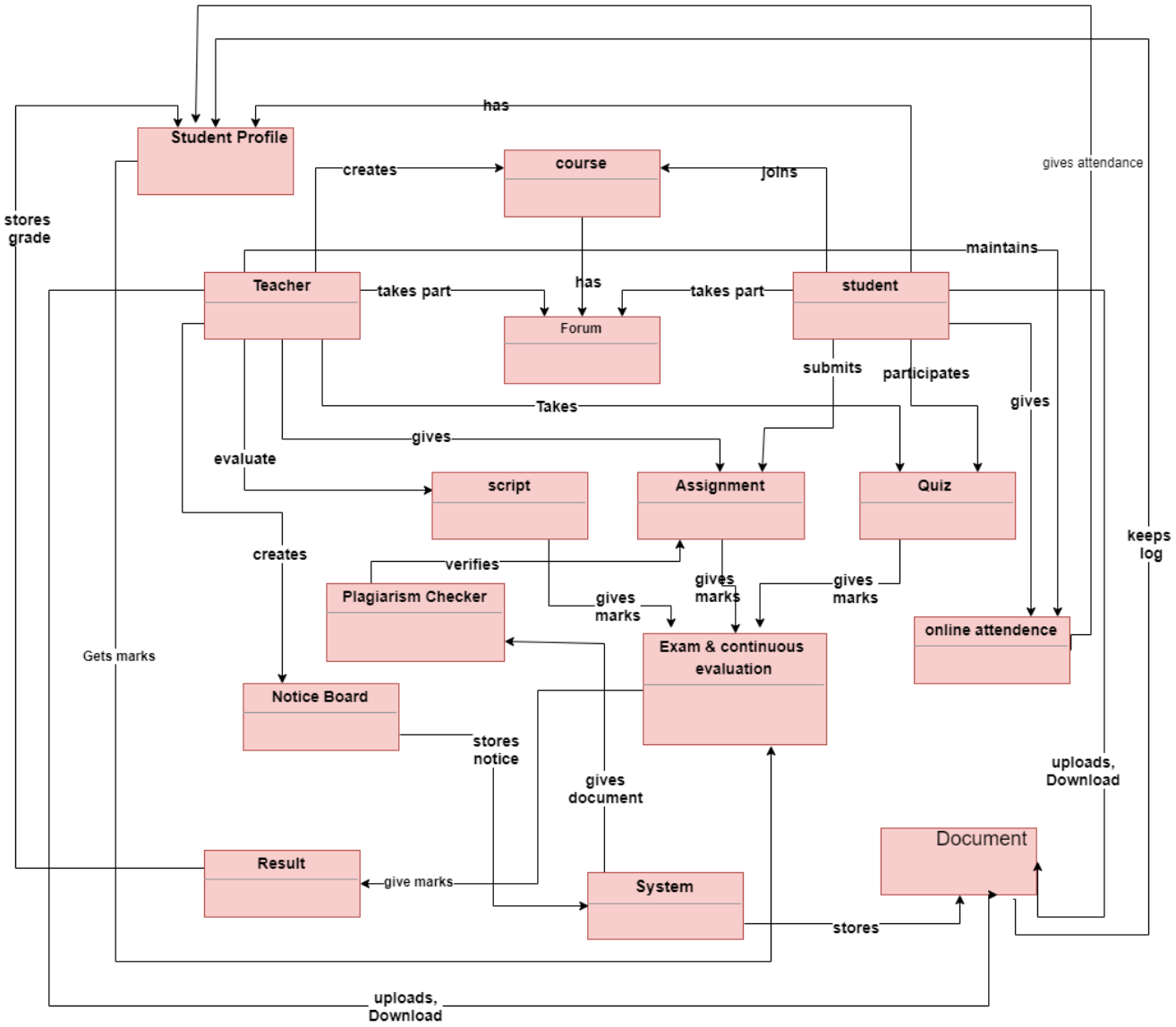


Fig 42: Class Modeling Diagram

Description :

Class diagram is shown in figure 6.9. This diagram shows the interactions/relationships among the classes.

7 BEHAVIORAL MODELING OF CMS

7.1 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.

SL	Events	Initiator	Collaborator	State Name
1	have to do registration	Student	System, teacher	Registering
2	send an invitation link	Teacher	Email	sending_invitation
3	complete their registration process	Student	System	Entering_into_the_link
4	to be verified and approved	System	Student, Teacher	Verifying_info
5	modify own information	Student	Teacher	Modifying_personal_info
6	require to login	Student	System	Logging_in
7	create own course	Teacher	System	Creating_course
8	invite to join the course	Teacher	Email	Inviting
9	provides mandatory information	Student	System, Student profile, Teacher	Providing_Info

10	request for providing necessary information	System	Student	Requesting
11	Upload file in class lecture	Teacher	System, Course	Uploading
12	Download file	Teacher, Student	System, Course	Downloading
13	give the schedule of class	Teacher	Online Attendance	Giving_class_schedule
14	automated code generation	System	Online Attendance, Student profile	Generating_code
15	fix a particular time	teacher	Online Attendance	Fixing_time
16	open a dashboard	Teacher	Online Attendance	Opening_dashboard
17	have to provide the code	Student	Online Attendance	providing_code
18	Class record will be kept in a log book	teacher	System	keeping_class_record
19	Generated a bar diagram	System	Course	generating_bar_diagram
20	View bar diagram	Student, Teacher	Teacher, Student	viewing_bar_diagram
21				
22	Upload General file	Student	Teacher, System	uploading

23	Count attendance	Online Attendance	Student profile, Student, System	counting_attendance
24	Marks will be stored in Student's profile	Student profile	Quiz, Assignment	storing_mark
25	curve will be generated	System	Student profile	generating_curve
26	View curve/log/notice board	Student, Teacher	System	viewing
27	Keeps the log of student's file	System	Student profile	keeping_log
29	open a dash board and set a particular time for upload	Teacher	System, Assignment, Student	opening_dashboard
30	Resubmit assignment	Student	System, Assignment	Resubmission
31	Log of assignment	Assignment	System	keeping_log

32	Set fix time, question, marking distribution	Teacher	Quiz	taking_quiz
33	Prepare a digital MCQ script	Teacher	System, Quiz	preparing_script

34	Checked by option making formula	Teacher	System, Quiz	checking_script()
35	Midterm can be taken with both physical script or via online quiz	Teacher	Script, Online Quiz,Student , System	taking_midterm()
36	Physically given mark will be stored against student's profile	Teacher	Student profile	storing_mark()
37	Assignment submit	Student	System, Assignment	submitting_assignment ()
38	Marking the Assignment	teacher	Assignment, plagiarism checker	marking_assignment()
39	Check plagiarism of all the submitted assignment	System	Assignment, plagiarism Checker	checking_plagiarism()
40	Percentage of plagiarism will be assigned in The assignment	System	Student profile, Plagiarism checker	assigning_plagiarism _percentage()
41	Take presentation	Teacher	Student	Taking_presentation

43	Lab Evolution mark will be stored in student's profile	teacher	Student profile	storing_mark
44	Create a template	Teacher	Script	creating_template
45	Set judging criteria for each question	teacher	Script	setting_template
46	Comment or giving feedback	Teacher	Script	commenting_and_feedback
47	Request for review	Student	System, Teacher, Script	requesting_for_review
48	Comments shows	Script	Student	showing_comments
49	Rechecking Script	Teacher	Student, Script	rechecking
50	Set criteria of marks and weight for final result	Teacher	result	setting_mark_and_weight
51	Generates final grades	System	Result	Generating_result

52	Sent final grades to Students	System	Email, Result	sending_grades
53	Search item from forum/log	Teacher, student	Forum, System, student profile	searching

54	Add file in notice	Teacher	Notice, System	uploading
55	Create notice	Teacher	Notice	creating_notice
57	Notification of the notice will be sent	System	Notice, Student, Email	Sending_notification
58	Remove student	Teacher	System, Course	removing_student
59	Logout	Teacher,student	System	logging_out

7.1.2 Events after analysis:

After some analysis, we can merge some events and states which are of same types. The analyzed data are given below :

SL	Events	Initiator	Collaborator	State Name
1	have to do registration	Teacher	System	Registration
2	create own course	Teacher	System	Creating_course
3	send an invitation link	Teacher	Email	sending_invitation
4	provides mandatory information	Student	System,teacher	providing_info
5	require to login	Student,teacher	System	Logging_in
6	modify own information	Student,Teacher	System	Modifying_personal_info
7	Upload file in class lecture	Teacher	System, Course	Uploading
8	Download file	Teacher, Student	System, Course	Downloading
9	give the schedule of class	Teacher	Online Attendance	Giving_class_schedule
11	automated code generation	System	Online Attendance, Student profile	Generating_code
12	fix a particular time	teacher	Online Attendance	Fixing_time

13	open a dashboard	Teacher	Online Attendance	Opening_dashboard
14	have to provide the code	Student	Online Attendance	providing_code
15	Class record will be kept in a log book	System		keeping_class_record
16	Generate a bar diagram	System	Course	generating_bar_diagram
17	Download bar diagram	Student, Teacher	Student, Teacher	downloading
18	Upload General file	Student	Teacher, System	uploading
19	Count attendance	Online Attendance	Student profile, Student, System	counting_attendance
20	Marks will be stored in Student's profile	Student profile	Quiz, Assignment	storing_mark
21	curve will be generated	System	Student profile	generating_curve
22	View curve/log/notice board/bar diagram	Student, Teacher	System	viewing
23	Keeps the log of student's file	System	Student profile	keeping_log
24	open a dash board and set a particular time for upload	Teacher	System, Assignment, Student	opening_dashboard
25	Resubmit assignment	Student	System, Assignment	Resubmission

26	Keep log of assignment	System	student	keeping_log_assigment

27	Set fix time, question, marking distribution	Teacher	Quiz	taking_quiz
28	Prepare a digital MCQ script	Teacher	System, Quiz	preparing_script
20	Checked by option making formula	Teacher	System, Quiz	checking_script
30	Midterm can be taken with both physical script or via online quiz	Teacher	Script, Online Quiz,Student, System	taking_midterm
31	Physically given mark will be stored against student's profile	Teacher	Student profile	storing_mark
32	Assignment submit	Student	System, Assignment	submitting_assignment
33	Marking the Assignment	teacher	Assignment, plagiarism checker	marking_assignment

34	Check plagiarism of all the submitted assignment	System	Assignment, plagiarism Checker	checking_plagiarism
35	Percentage of plagiarism will be assigned in The assignment	System	Student profile, Plagiarism checker	assigning_plagiarism _percentage
38	Take presentation	Teacher	Student	Taking_presentation
39	Lab Evolution mark will be stored in student's profile	teacher	Student profile	storing_mark
40	Create a template	Teacher	Script	creating_template
41	Set judging criteria for each question	teacher	Script	setting_template
42	Comment or giving feedback	Teacher	Script	commenting_and _feedback
43	Rechecking Script	Teacher	Student, Script	rechecking

44	Generates final grades	System	Result	Generating_result
45	Sent final grades to Students	System	Email, Result	sending_grades
46	Search item from forum/own log	Teacher, student	Forum, System,student profile	searching
47	Add file in notice	Teacher	Notice, System	uploading
48	Create notice	Teacher	Notice	creating_notice
49	Notification of the notice will be sent	System	Notice, Student, Email	Sending_notification
50	Remove student	Teacher	System, Course	removing_student
51	Logout	Teacher,student	System	logging_out

7.1.2 State Transition Diagram:

System :

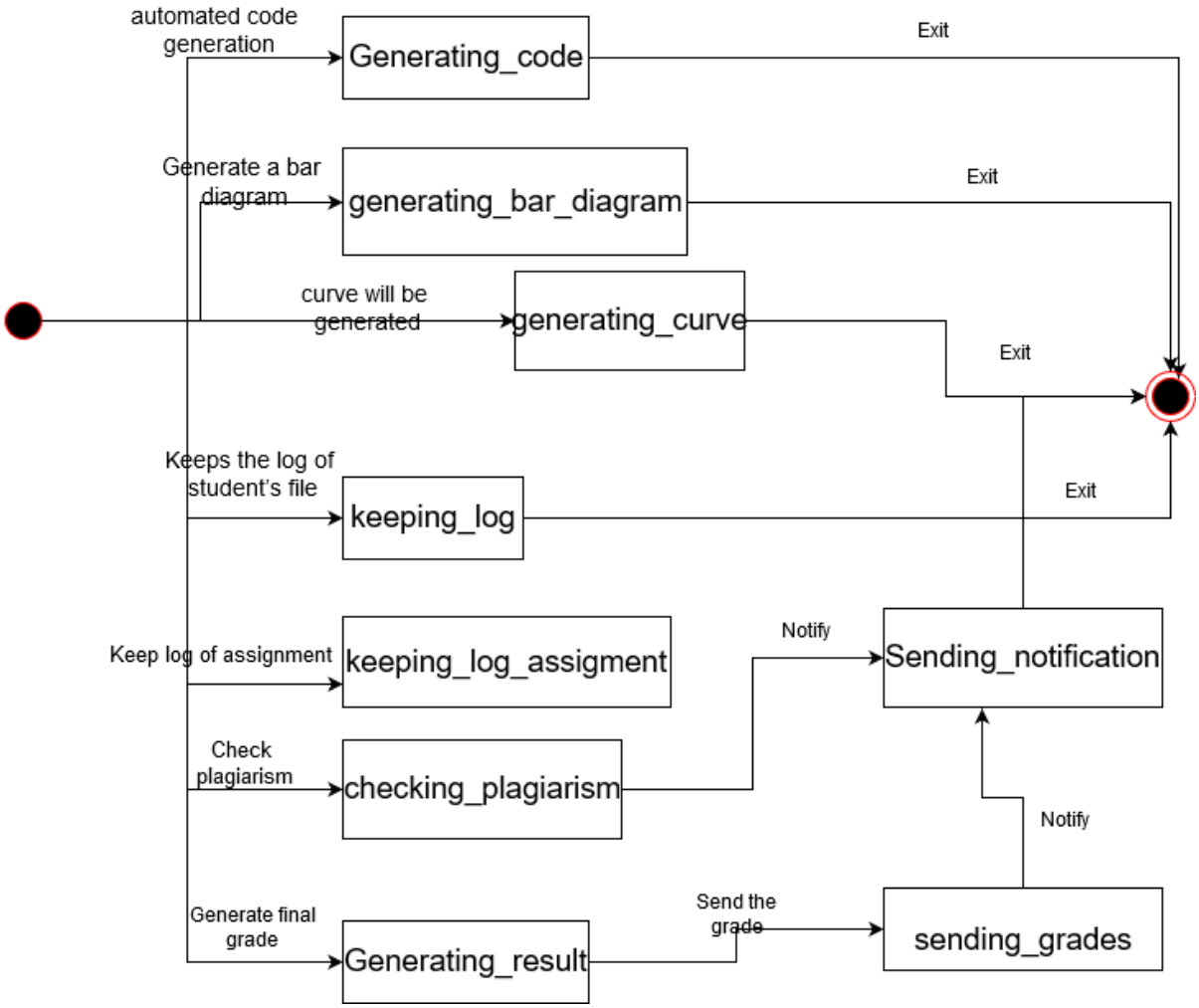


Figure 43 : State transition for System class

Description :

Figure 43 shows state transition diagram of System class. This class mainly performs the automated tasks and user interfaces.

Course Teacher

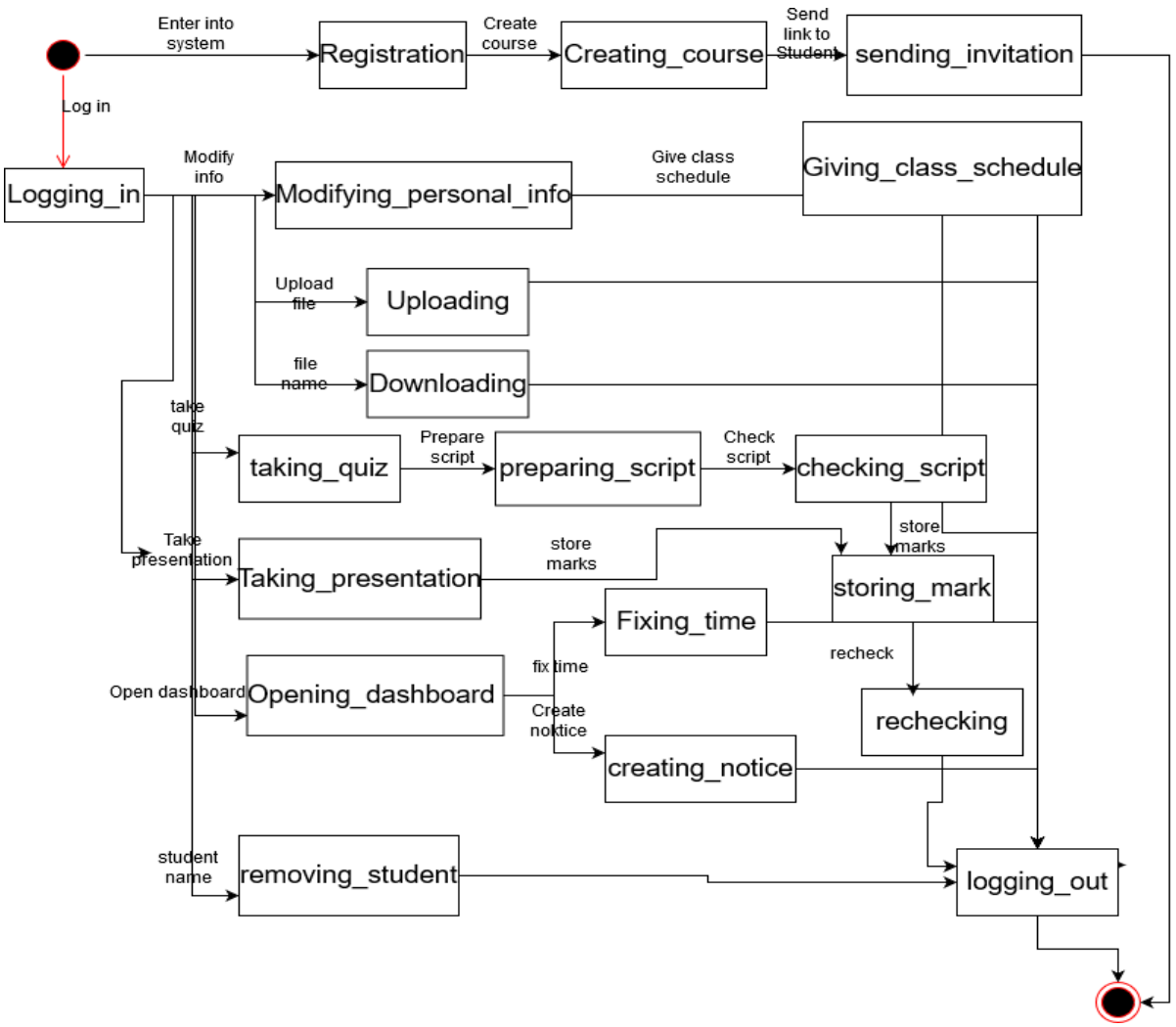


Figure 44: State transition for Course Teacher class

Description:

Figure 44 shows state transition diagram for Teacher class. Teacher is the main manipulator and user of the Course Management System. This diagram shows how the overall system is initiated, manipulated, controlled and used by the course teacher.

Student

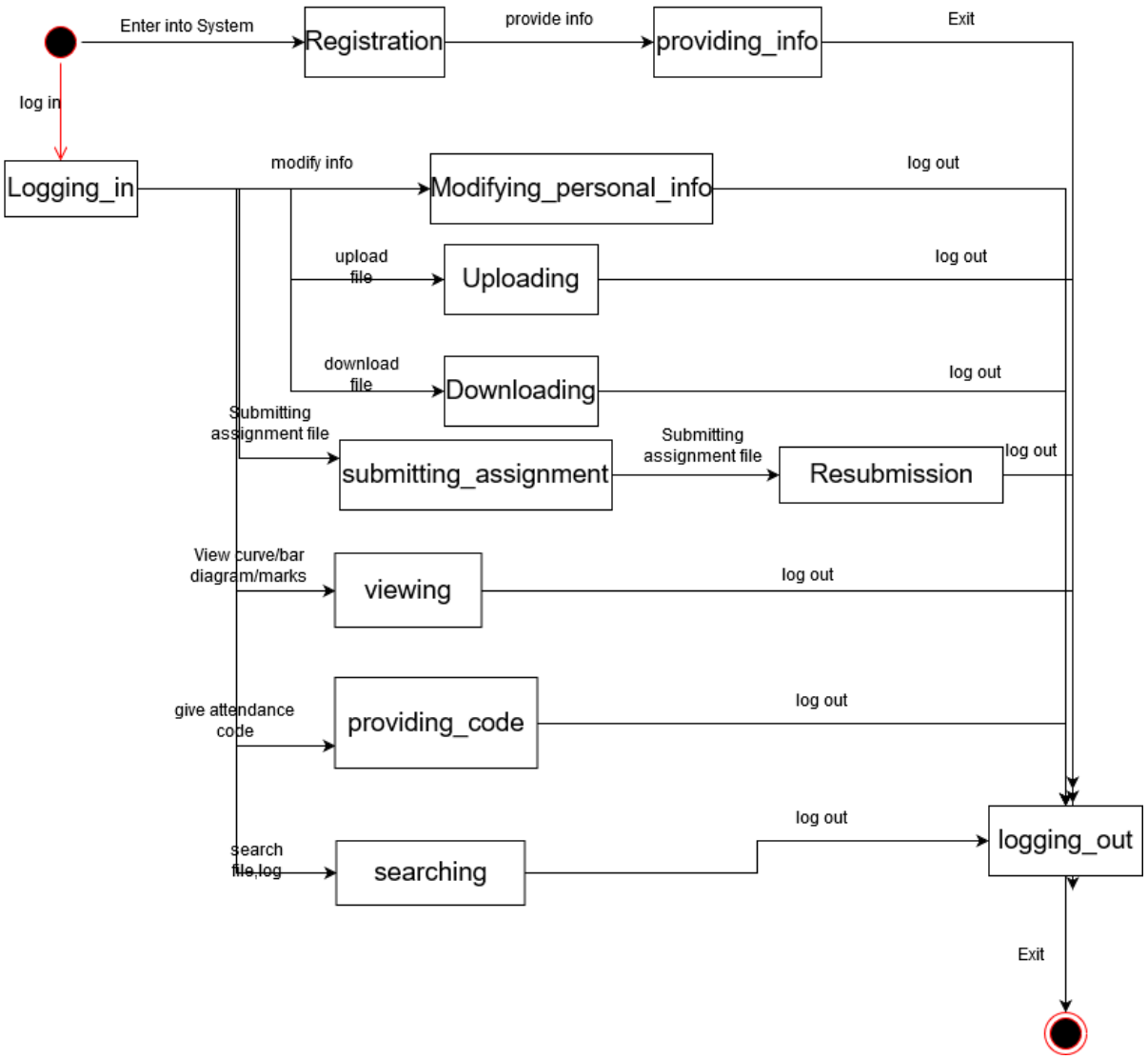


Figure 45 : State transition for Student class

Description :

Figure 45 shows state transition diagram for Student class. Students are the users for whom the Course Management System is mainly designed. This diagram shows which functionalities a student can perform while using the system and also which task can be performed after what.

Forum :

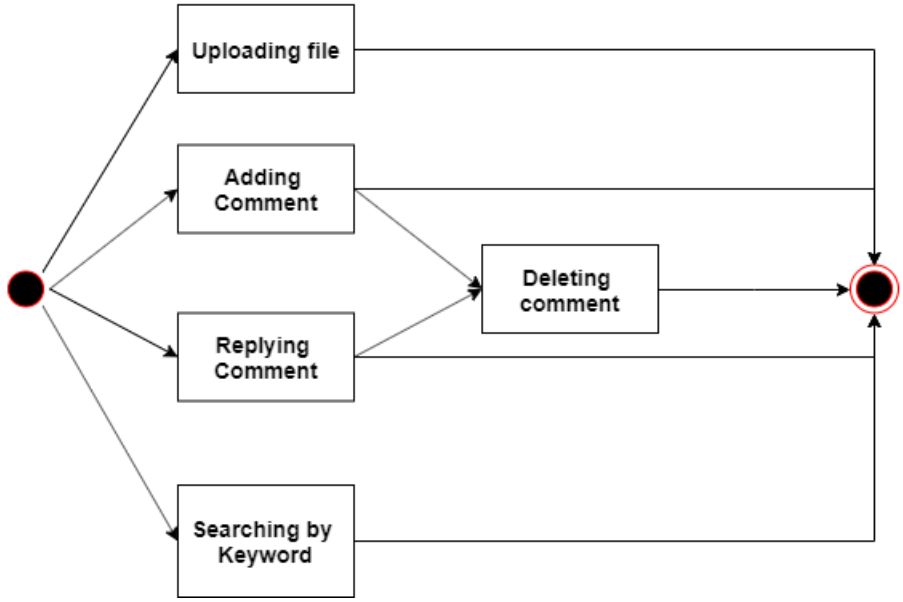


Figure 46 : State transition for Forum class

Description:

Figure 46 shows state transition diagram for Forum class. Forum will be used for query or problem discussion. Both teachers and students are allowed to contribute here by adding comments and uploading files. Comments can be edited deleted and replied also.

Course

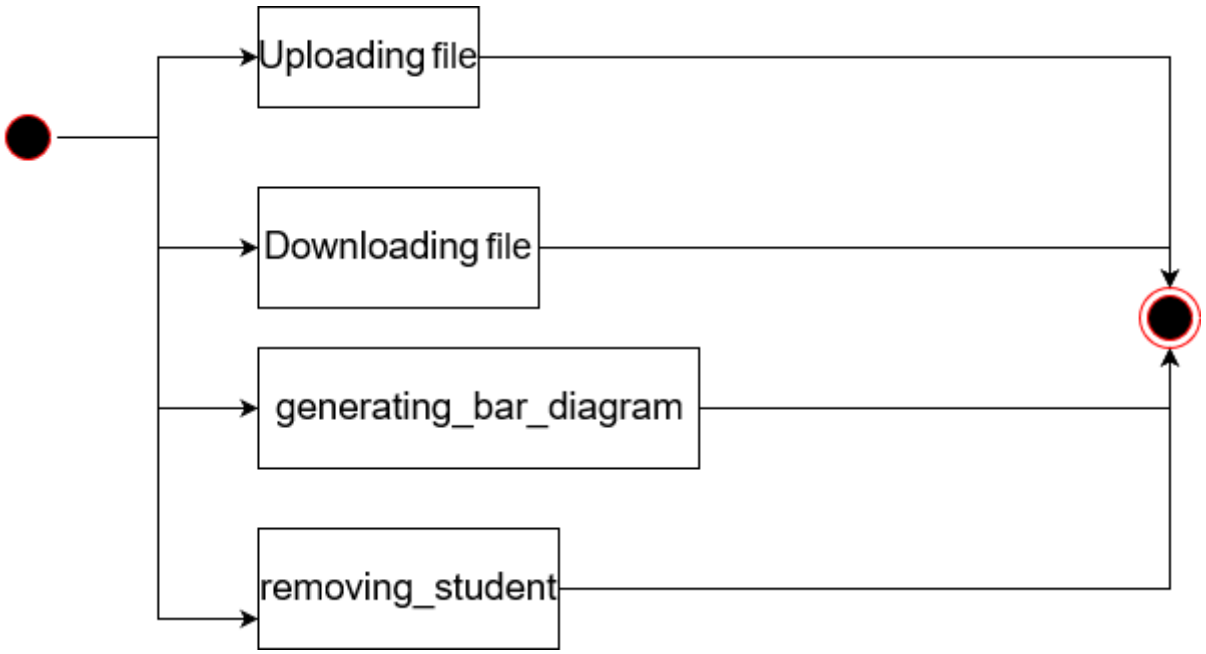


Figure 47: State transition for Course class

Description:

Figure 47 shows state transition diagram for Course class

Student Profile

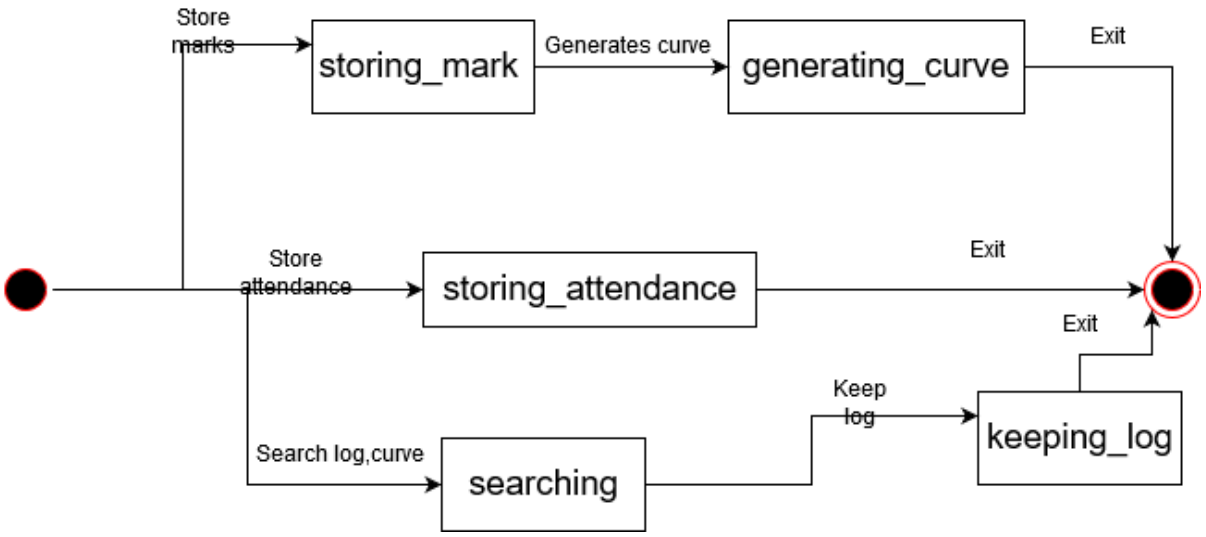


Figure 48: State transition for Student Profile class

Description:

Figure 48 shows state transition diagram for Student Profile class.

Online Attendance :

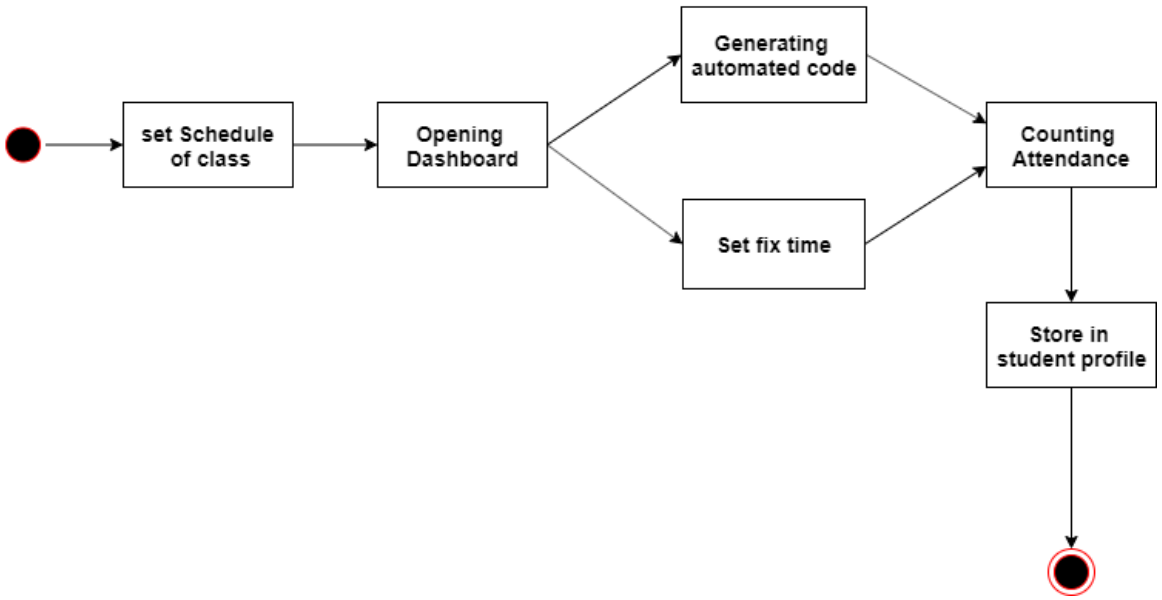


Figure 49: State transition for Online Attendance class

Description:

Figure 49 shows state transition diagram for Online Attendance class. The teacher will give the schedule of classes and according to it, an automated code will be generated by the teacher for attendance. The teacher will fix a particular time and open a dashboard. Students will have to provide the code. Those students who will provide the code within the fixed time will be counted as present.

Assignment :

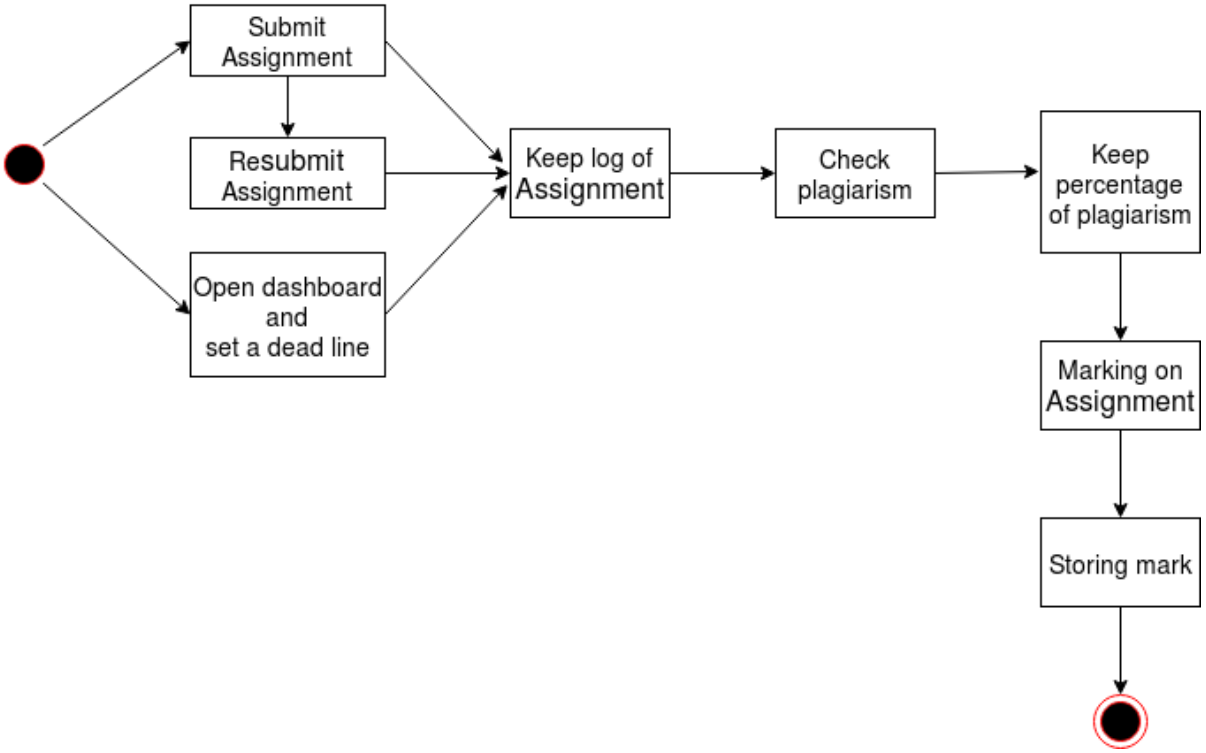


Figure 50: State transition for Assignment class

Description:

Figure 50 shows state transition diagram of Assignment class. In our system, online assignment submission follows the procedures mentioned in the above figure.

plagiarism_checker :

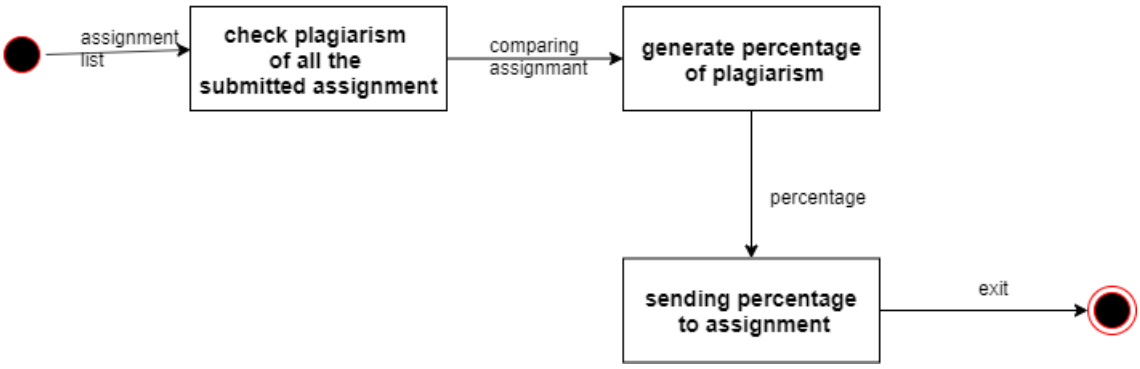


Figure 51: State transition for Plagiarism Checker class

Description:
Figure 51 shows state transition diagram of Plagiarism checker class. Plagiarism checker will verify the submitted assignments comparing with all the assignments of other students. This is a wow factor of our system.

Quiz :

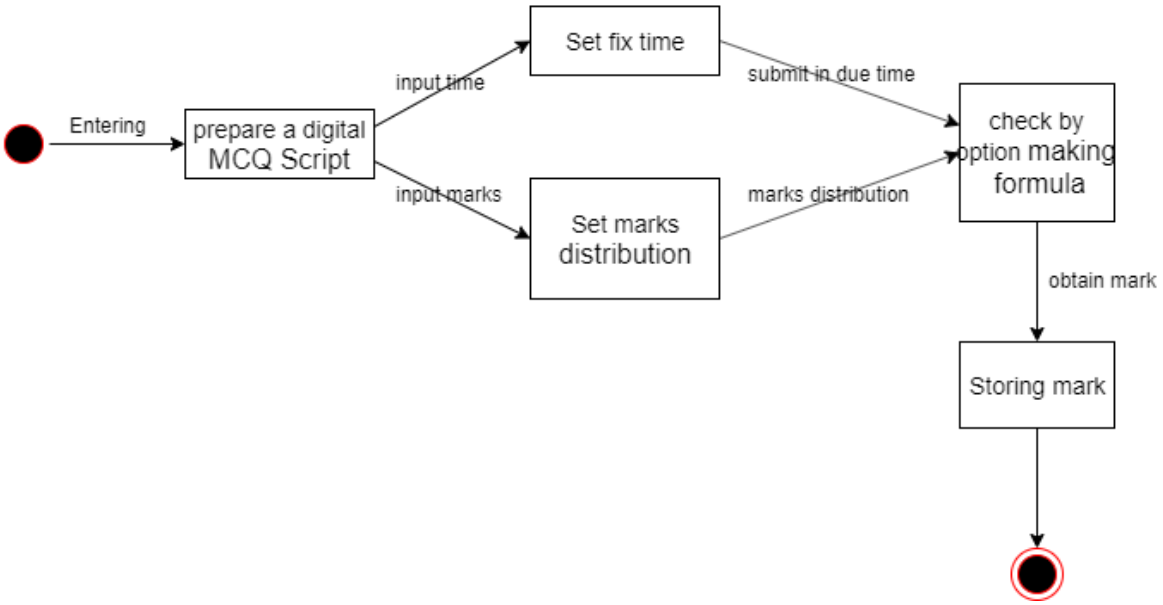


Figure 52: State transition for Quiz class

Description:
Figure 52 shows state transition diagram of Quiz class. Quiz class will go through some manual parts controlled by course teacher and some automated parts generated by system. These are shown in the above figure.

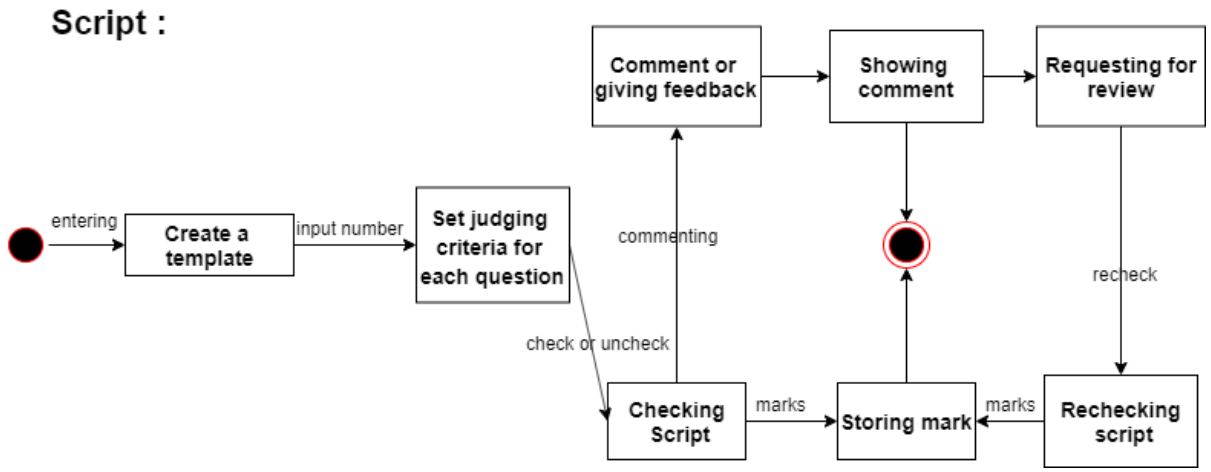


Figure 53: State transition for Script class

Description:
Figure 53 shows state transition diagram of Script class. Script class will go through some manual parts controlled by course teacher and some automated parts generated by system. These are shown in the above figure

Exam and Continuous evolution :

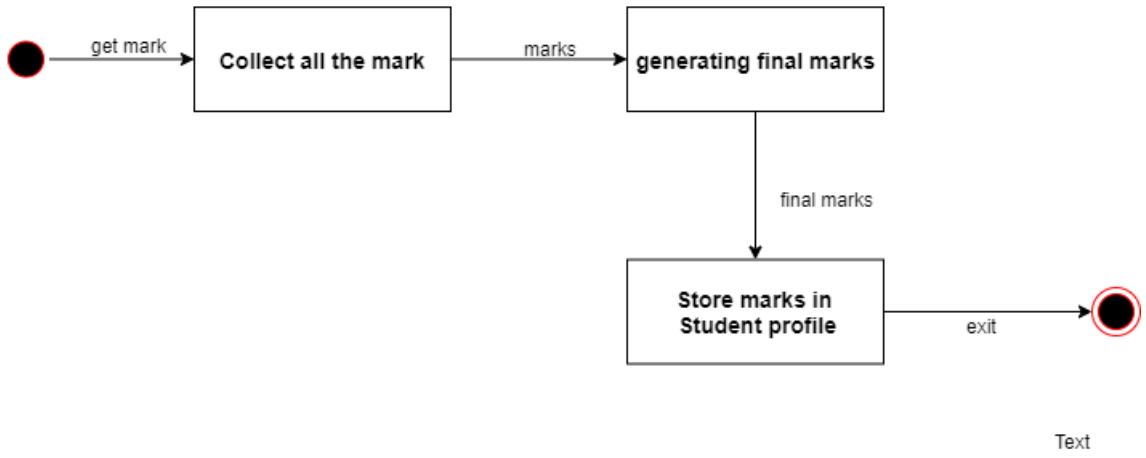


Figure 54: State transition for Exam and Continuous Evaluation class

Description:

Figure 54 shows state transition diagram of Exam and Continuous Evaluation class. All sorts of exams and continuous evaluation parts are subsystems of this class. This class will collect marks from all marking criteria and provide a system for generating result. Finally, it will store the results against students' profile.

Notice board :

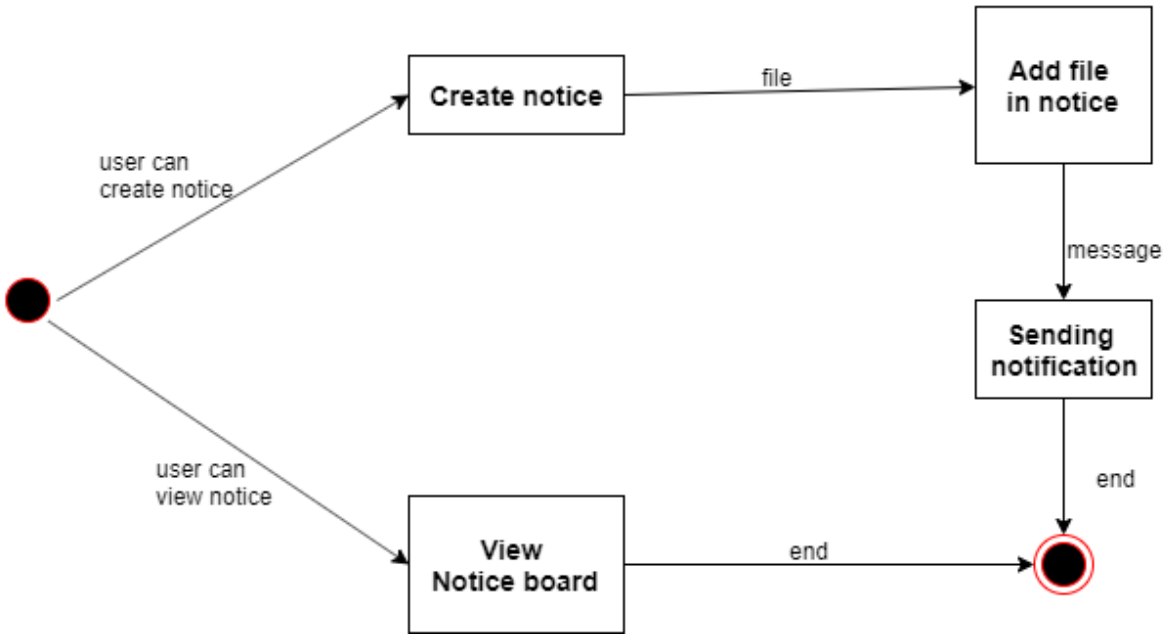


Figure 55: State transition for Notice Board class

Description:

Figure 55 shows state transition diagram of Notice Board class. This shows the functionalities of creating, holding and sending notice.

Result :

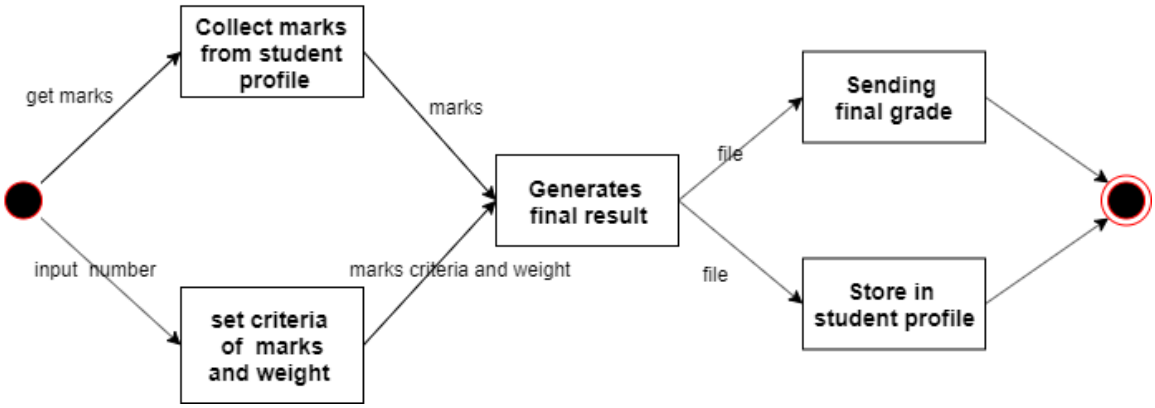


Figure 56 : State transition for Result class

Description:
Figure 56 shows state transition diagram of Result class. Result generation is mostly an automated subsystem which follow the shown steps above.

Document :

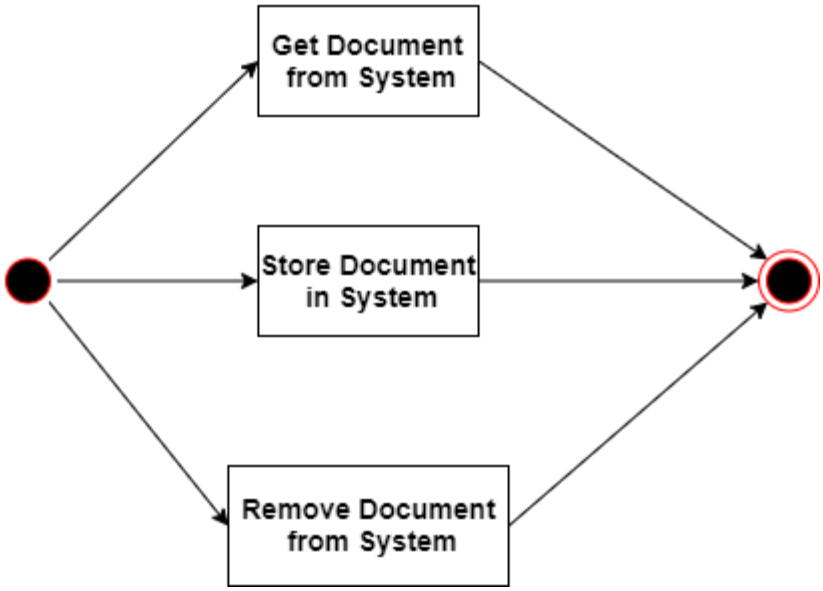


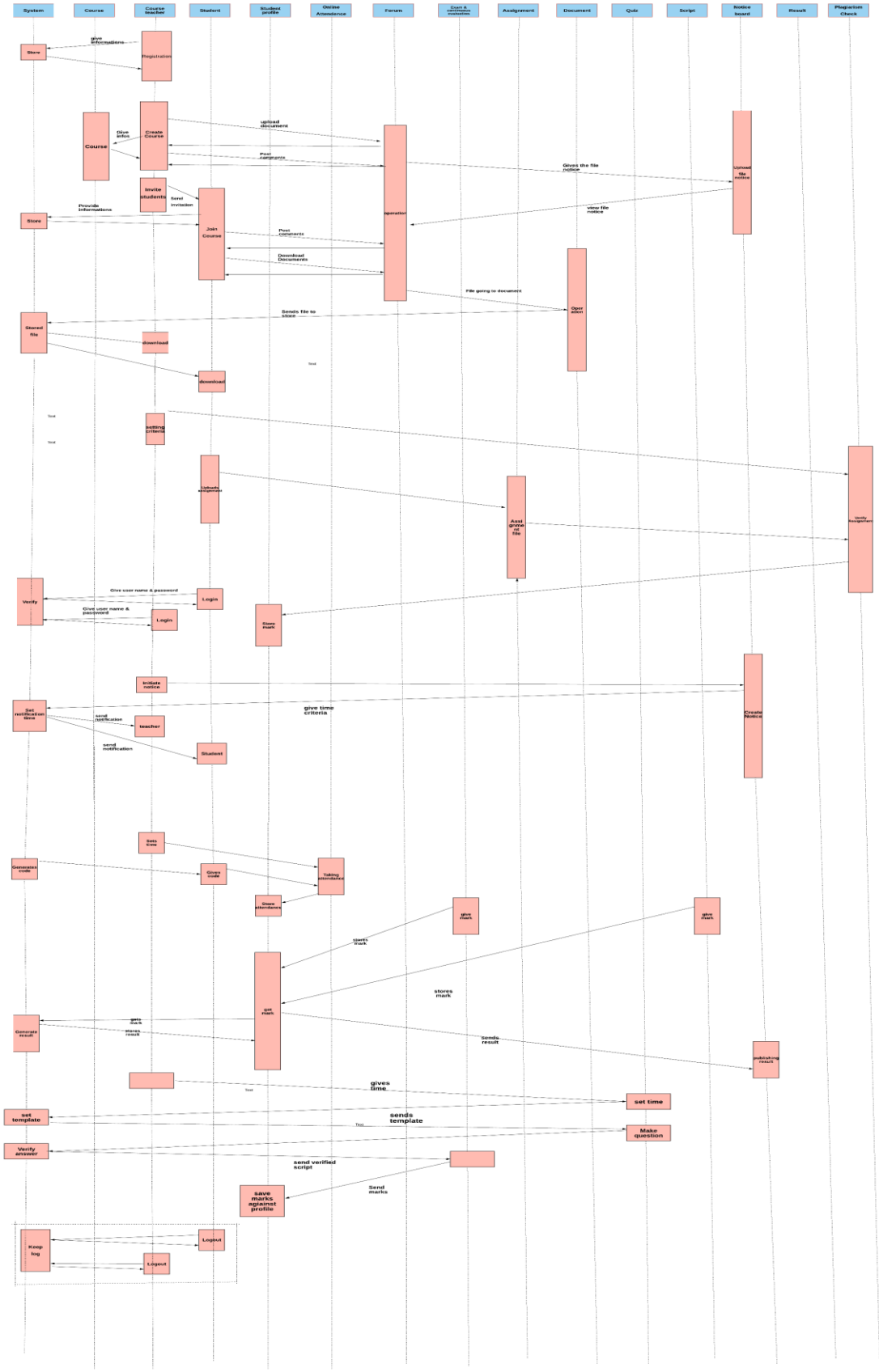
Figure 57 : State transition for Document class

Description :
Figure 57 shows state transition diagram of Document class

7.2 Sequence Diagram

Fig no:58

Name: sequence Diagram


System
Course
Course teacher
Student
Student profile
Online Attendance
Forum
Exam & continuous mainteam
Assignment
Document
Quiz
Script
Notice board
Result
Plagiarism Check
Store
give informations
Registration
Course
Give info
Create Course
upload document
Post assignments
Invite students
Send invasion
Provide informations
Store
Join Course
Post comments
Download Documents
operation
Gives the file notice
Upload file notice
view file notice
Operation
File going to document
Sends file to save
Stored file
download
download
setting criteria
Upload assignment
Assignme nt file
Verify assignment
Verify
Give user name & password
Login
Give user name & password
Login
Store mark
Initiate notice
Create Notice
Set notification time
give time criteria
send notification
teacher
send notification
Student
Sets time
Generate code
Gives code
Taking attendance
Store attendance
give mark
give mark
stores mark
get mark
gets mark
stores result
Generate result
Sends result
publishing result
gives time
set time
set template
sends template
Make question
Verify answer
send verified script
save marks against profile
Send marks
Keep log
Logout
Logout

8 Data Flow Diagram

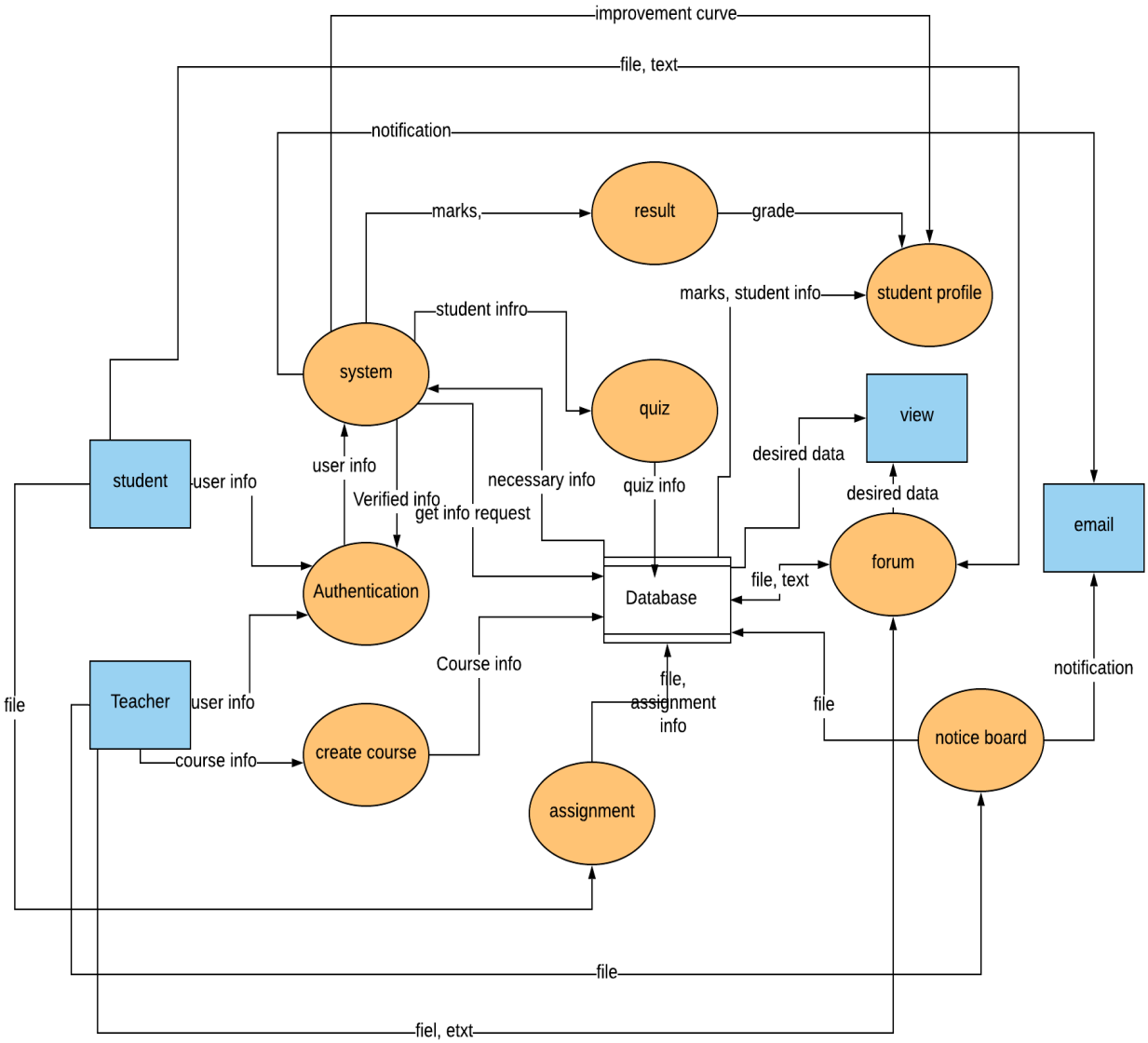


Fig 59 : Data flow Diagram