SOFTWARE DESIGN DOCUMENT

Paathshaala Website

Version 1.0

Prepared by : 1. Abhishek Bansal (190001001)
2. Arastu (190001006)
3. Dipin Garg(190001013)
4. Harsh Kushwaha(190001018)
5. Jainil Shah(190001020)
6. Prasheel Kumar Tiwari(190004025)

Submitted to : Dr. Puneet Gupta
Assistant Professor

May 6, 2021

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to give details of the implementation of the Paathshaala Software described in the Paathshaala Requirements Specification document. The Paathshaala Software is designed to facilitate easy work-flow and exchange of information between the professors and the students. The primary audiences of this document are the software developers.

1.2 Scope

This document describes the implementation details of the Paathshaala Software. The software will consist of two parts. First part will contain implementation for the student functionalities. The second part will contain implementation for the faculty functionalities.

1.3 Overview

This document will describe software's data design, architecture design, interface design, and procedural design. With the help of UML diagrams, design of the system and subsystems/modules will be explained visually in order to help the programmer to understand all information stated in this document correctly and easily.

1.4 Definitions,Acronyms,Abbreviations

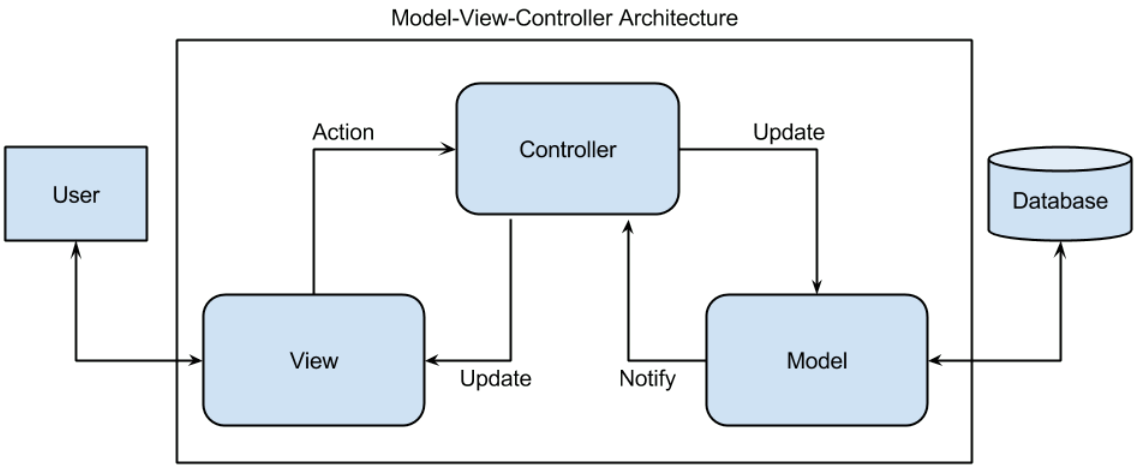
- **Compulsory Course**- These are the courses which are mandatory for the students to complete. The students are automatically added to these courses for enrollment and do not have the choice to leave these courses.
- **Optional Course**- It is not mandatory for the students to complete these courses. Students will not be automatically added in these courses and will have to request for enrollment in these courses.
- **Assignment**- This is a method to evaluate the students where the question can contain text and attachments. The submissions for assignments contains attachments.
- **Quiz**- This is similar to assignment, but the questions can be of MCQ type(single correct or multiple correct) or subjective type(answer is typed in the text box).
- **Discussion Forum**- Here the students enrolled in a course can discuss with faculty and other students problems regarding that course. Each course has an independent discussion forum.

2 Design Overview

2.1 Description of problem

Generally the course materials are difficult to manage without any organisation and can get confusing if not managed properly. Keeping track of assignment deadlines can get tough for the students and maintaining the submissions such that it easy to check and upload the grades can get hectic for the faculty. Also, currently the courses are moving towards online classroom setting. This software will help in organising the course materials, assignments and also make it easy for the faculty to view and grade submissions. So, using this software navigating between tasks will become easy and efficient and will reduce the workload of the users.

2.2 Architecture



Our Flask Web-Application uses an architecture called the MVC architecture. The Model-View-Controller architectural pattern (MVC) divides an interactive application into three components.

- The model(SQLAlchemy) contains the core functionality and data.
- Views(Jinja2) display information to the user.
- Controllers(Flask) handle user input. Views and controllers together comprise the user interface. A change-propagation mechanism ensures consistency between the user interface and the model.

The model component contains the functional core of the application. It encapsulates the appropriate data, and exports procedures that perform application-specific processing. Controllers call these procedures on behalf of the user. The model also provides functions to access its data that are used by view components to acquire the data to be displayed.

View components display information to the user. A view obtains the data from the model. There can be multiple views of the model. Each view has an associated controller component. Controllers receive input, usually as events that encode mouse movement, activation of mouse buttons, or keyboard input. Events are translated to service requests for the model or the view. The user interacts with the system solely through controllers.

2.3 Use Cases

The use case diagram for the software is as shown below. The functionality exclusive to the students include-

- The students can view their assignments. This will show all the assignments assigned to the student along with their status. Here the student can select a pending assignment and submit a response.
- The students can view the quizzes scheduled in the courses which they are enrolled in. During the duration of the quiz they can enter the quiz page and give the quiz.
- The students will receive enrollment invitations for compulsory courses and ones in which their enrollment request have been accepted. They have to accept these to get enrolled in the course.
- The students can also view all the courses. Here they can sent enrollment request to get enrolled in an optional course which is not compulsory.
- Each students will have their own time table which will store their enrolled courses' lectures in tabular form.
- Students can view the course material for the courses which they are enrolled in.

The functionality common to both student and faculty include-

- Both the student and faculty have to login before they can use the software. If they are not registered, they have to register before logging in.
- Both student, faculty can view and post in discussion forums for courses which they are enrolled in or are currently teaching respectively.

The functionality exclusive to faculty include-

- The faculty can view and modify the lectures for the courses which they are currently teaching.
- Faculty can view and modify the students which are currently enrolled in courses which they are currently teaching.
- The faculty can create new assignments and quizzes for courses which they are currently teaching. They can also view and grade students' submissions for previous assignments or quizzes. Faculty will also have option to make grades of an assignment or quiz visible to the students.
- The faculty can upload course material for courses which they are currently teaching.
- The faculty can create new courses which they will be teaching.

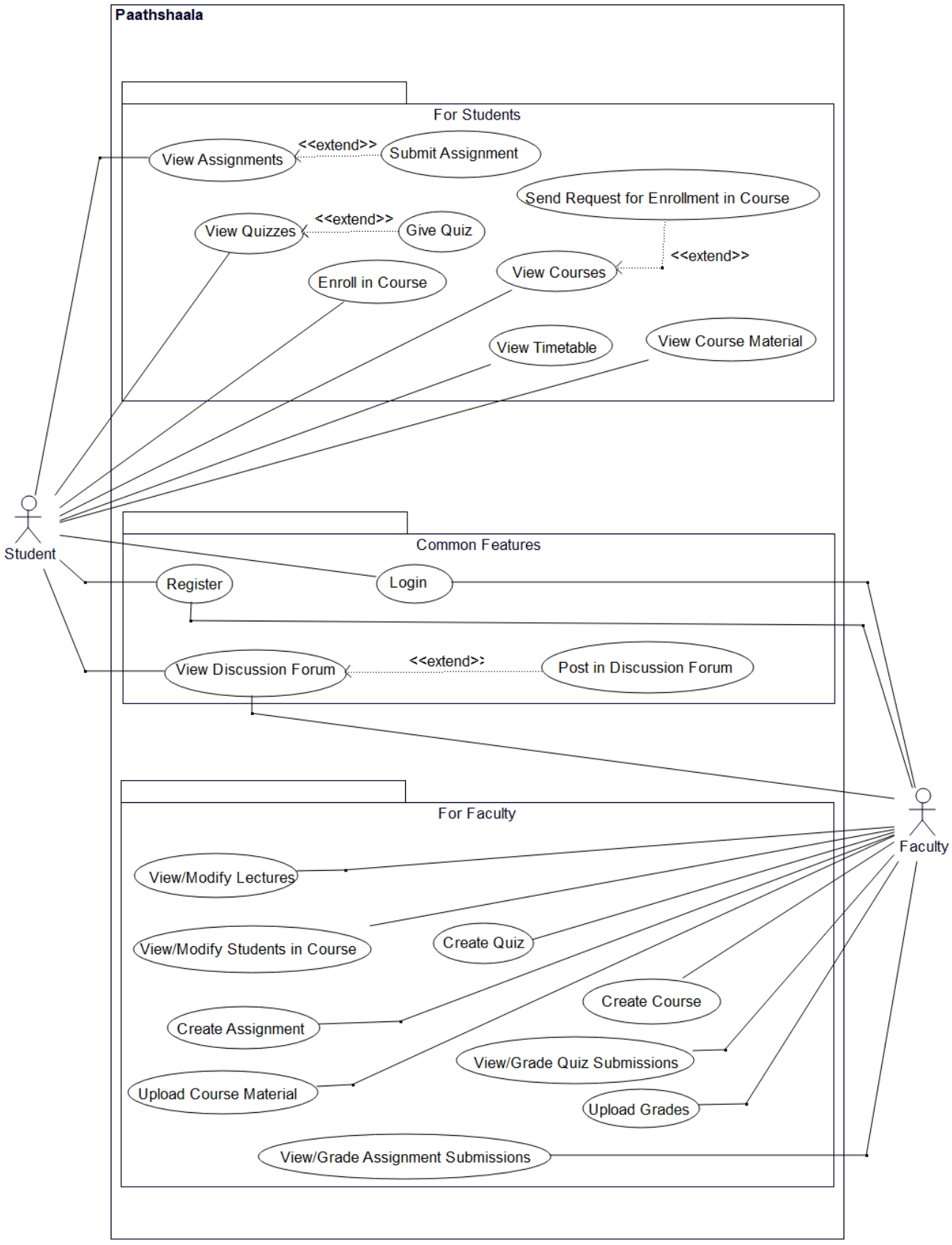


Figure 2.1: Use Case Diagram for Paathshaala

3 Class Details

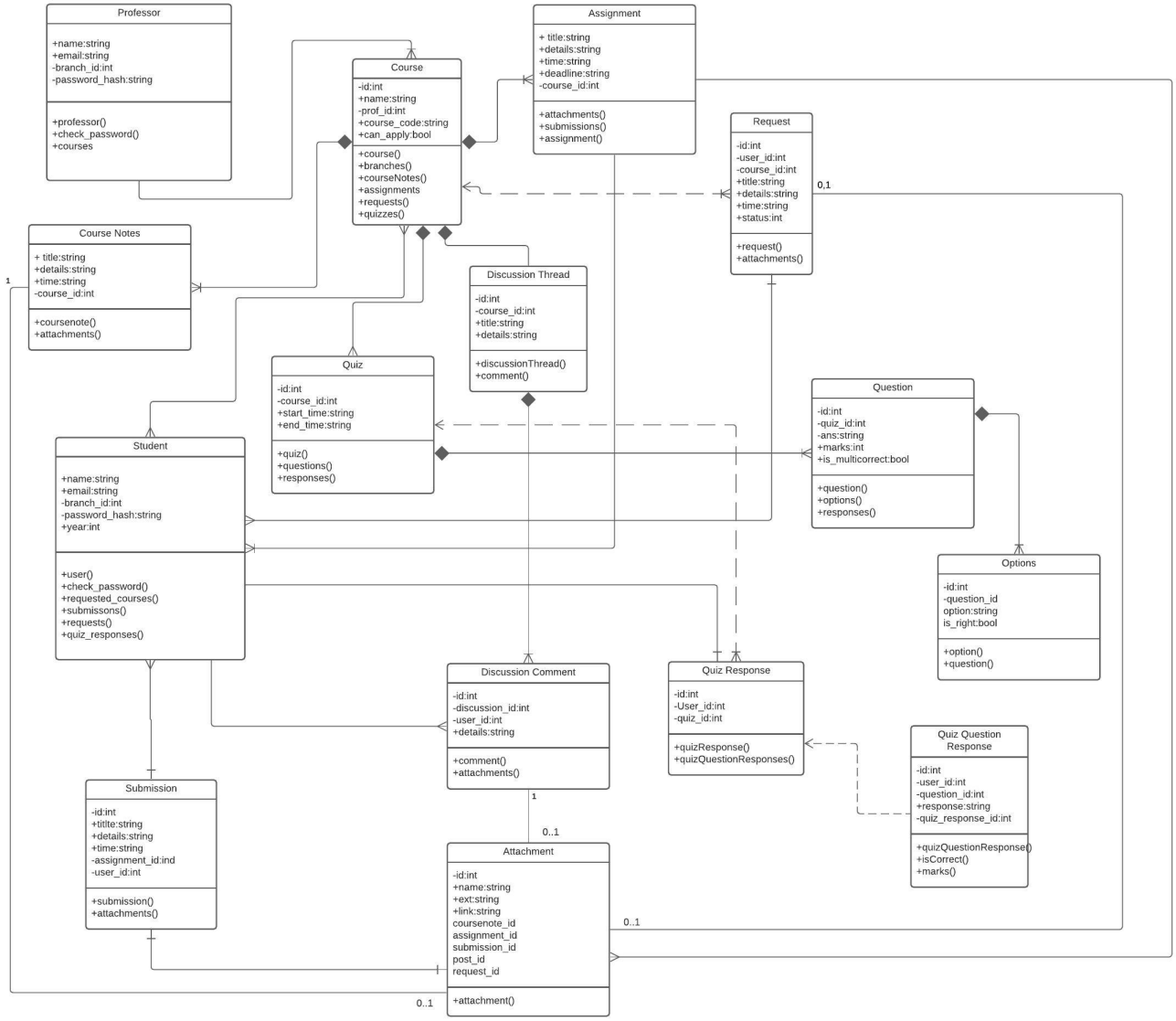


Figure 3.1: Class Diagram

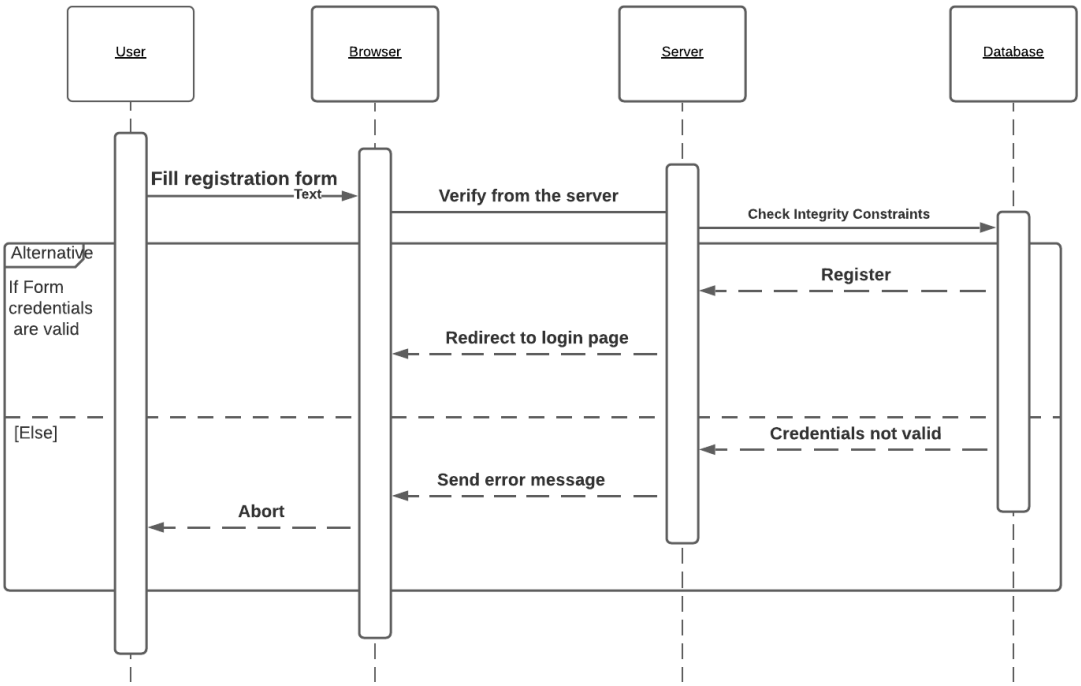
The classes in the class Diagram are Professor, Student, Course, Course Notes, Assignment, Request, Quiz, User, Submission, Attachment, Discussion Thread, Discussion Comment, Question, Options, Quiz Response, Quiz Question Response. Relationships among these classes are:

- Professor has a one-to-many relationship with Courses, because a professor can teach multiple courses.

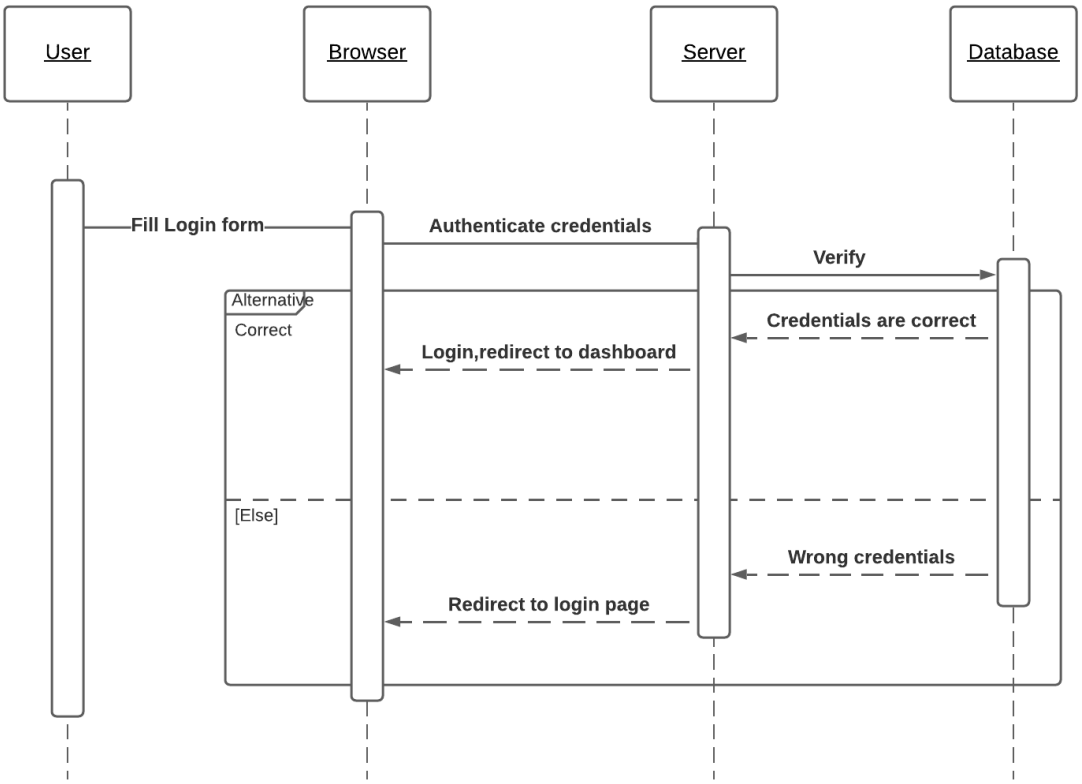
- Course has one-to-many relationship with Course Notes, because each course can have multiple course notes attached with it. Also their is composition relation between them, because course notes cannot exist independently.
- Course has one-to-many relationship with Assignment, and quiz bot because each course can have multiple assignments. Also their is composition relation between them, because assignments or quizzes cannot exist independently.
- Course again has one-to-many relationship with Request class, because their can be multiple requests to enroll in that course. Also since request is dependent on Course, so their is dependency relation between them.
- Course has one-to-one relationship with Discussion thread because their can only be one discussion thread associated with each course. Also their is a composition relation between them, because a course is comprised of Discussion Thread.
- Course and Student has many-to-many relationship with each other, because each course can have many students and a single student can be associated with multiple courses.
- Assignment has one-to-many relation with Student, because a single assignment can be assigned to multiple students.
- Student has one-to-many relationship with discussion comment, because a single student can post multiple discussion comments.
- Submission has one-to-many relationship with Student because a single submission can be made by only one student.
- Attachment has one-to-one relationship with Submission, Discussion Comment, Course Notes, Request, and Assignment classes since each submission can have only 1 attachment. In case of relation with Request, Discussion Comment and Course Notes classes, partial participation is allowed while in others, total participation of attachment is there.
- Discussion Comment has many-to-one relationship with Discussion Thread, because a discussion thread can have multiple discussion comments.Also their is a composition relation their, because each discussion thread is composed of these discussion comments.
- Student has association relation with Quiz Response.
- Student has many-to-one relation with Request class because every student can make multiple request for enrollement in courses.
- Quiz Response has dependence relation with Quiz Class.
- Since a Quiz is composed of many questions, so Quiz has one-to-many relation with Question Class and also their is a composition relation between them.
- Student can also give a quiz, so it has association relation with Quiz Response.
- A Question will have multiple choices to select from, and so there is a one-to-many relation from Question to Options. There is a composition relation too, because each question is comprised of these options.

3.1 Authentication

1. Registration



2. Login



The flask application stores hashed versions of user passwords, which is matched each time a user tries to login. The authenticated user is then authorized to access the protected routes using cookies and sessions.

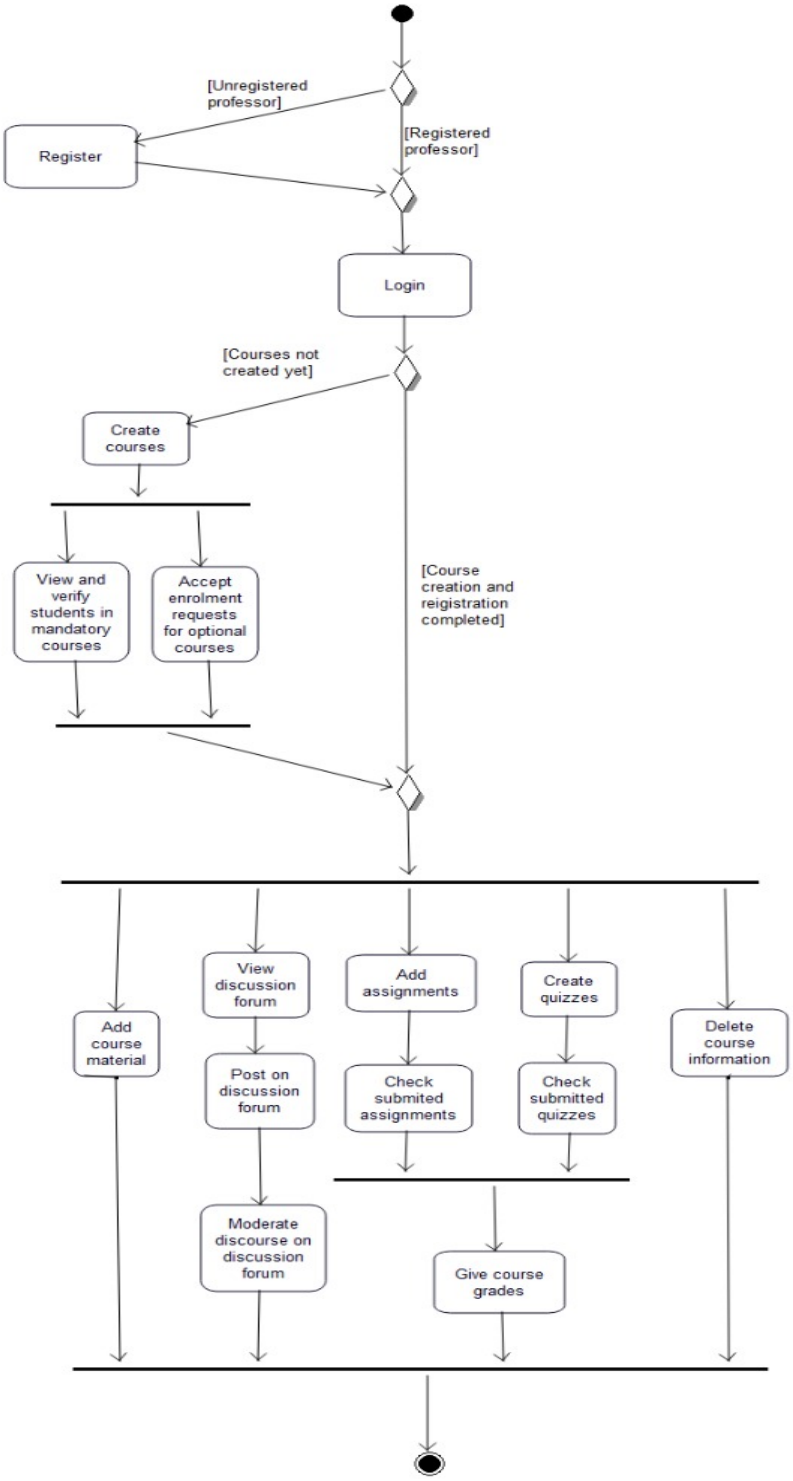
4 Professor Interface

4.1 Functions Performed

Professors can:

1. Register themselves
2. Login themselves
3. Create Courses
4. Add students to courses
5. Edit course information
6. View student applications to join course
7. Upload course material
8. Upload timed assignments
9. View submitted assignments of students and grade them
10. Create Quizzes for courses
11. View Students responses for quizzes
12. Post announcements in discussion thread of a course
13. View posts of students in discussion thread

4.2 Activity Diagram



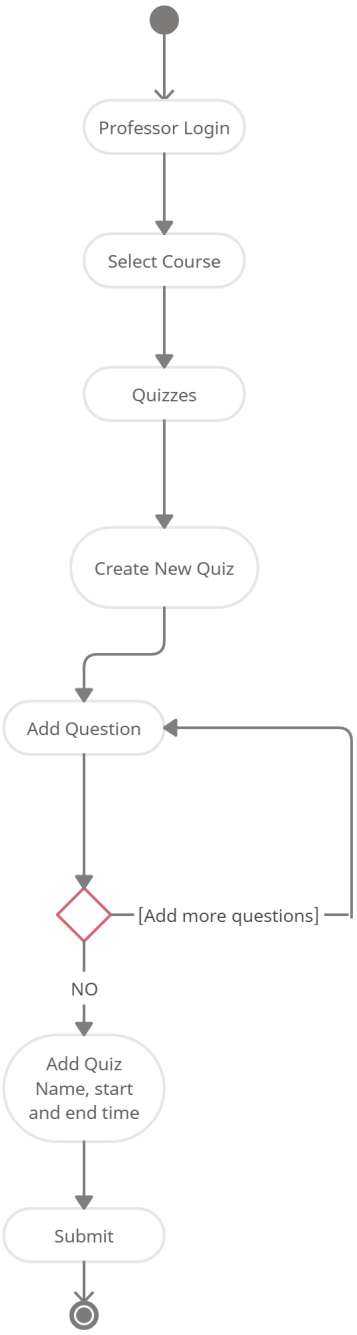
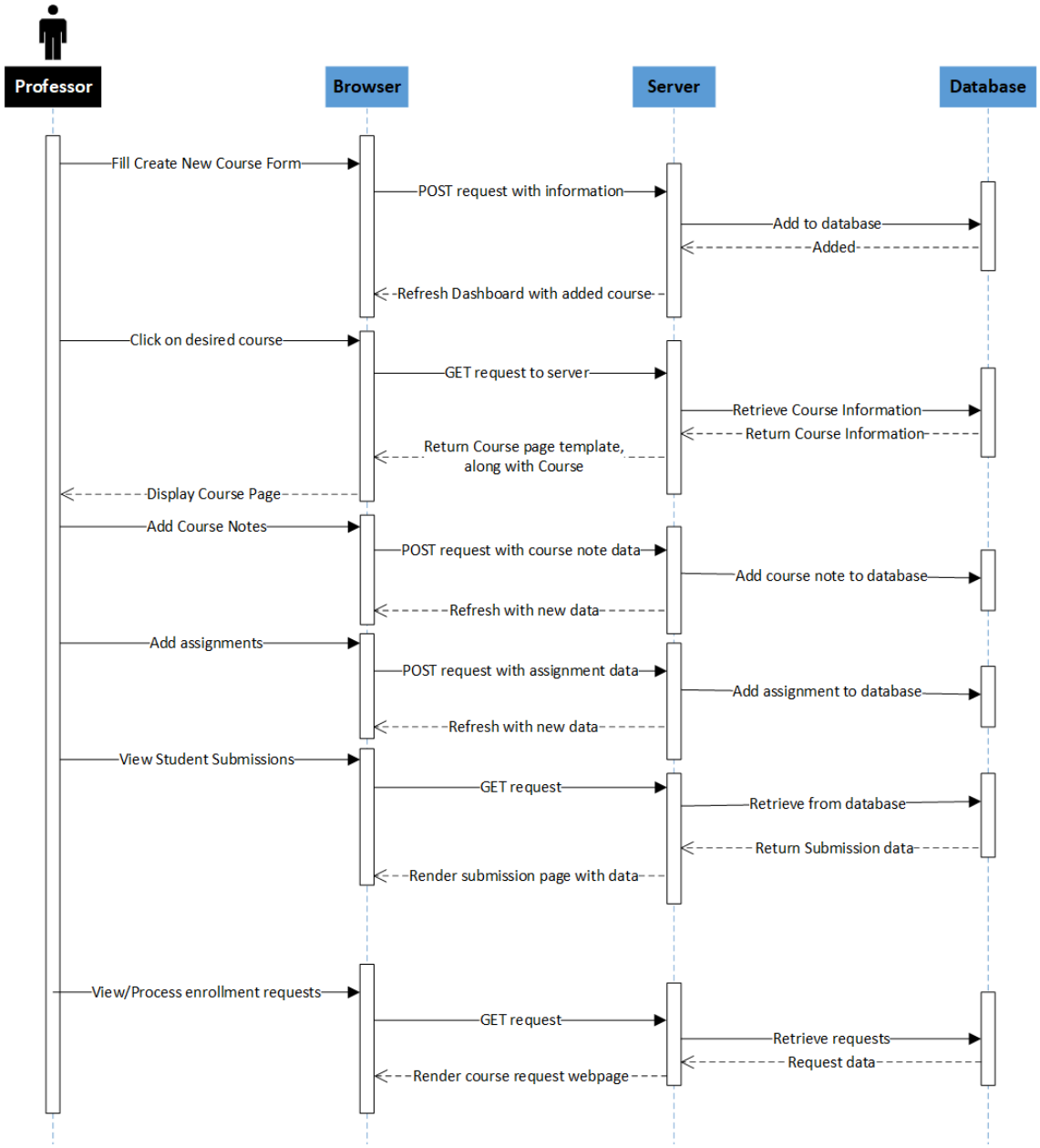


Figure 4.1: Professor Quiz Creation Diagram

To create a quiz for a course, Professors have to login and then go to quizzes section of that course, and click on create new quiz. Then they can add as many questions as they want and when they are finished, they have to enter the quiz name, start and end time of the quiz and finally click on submit to create the quiz.

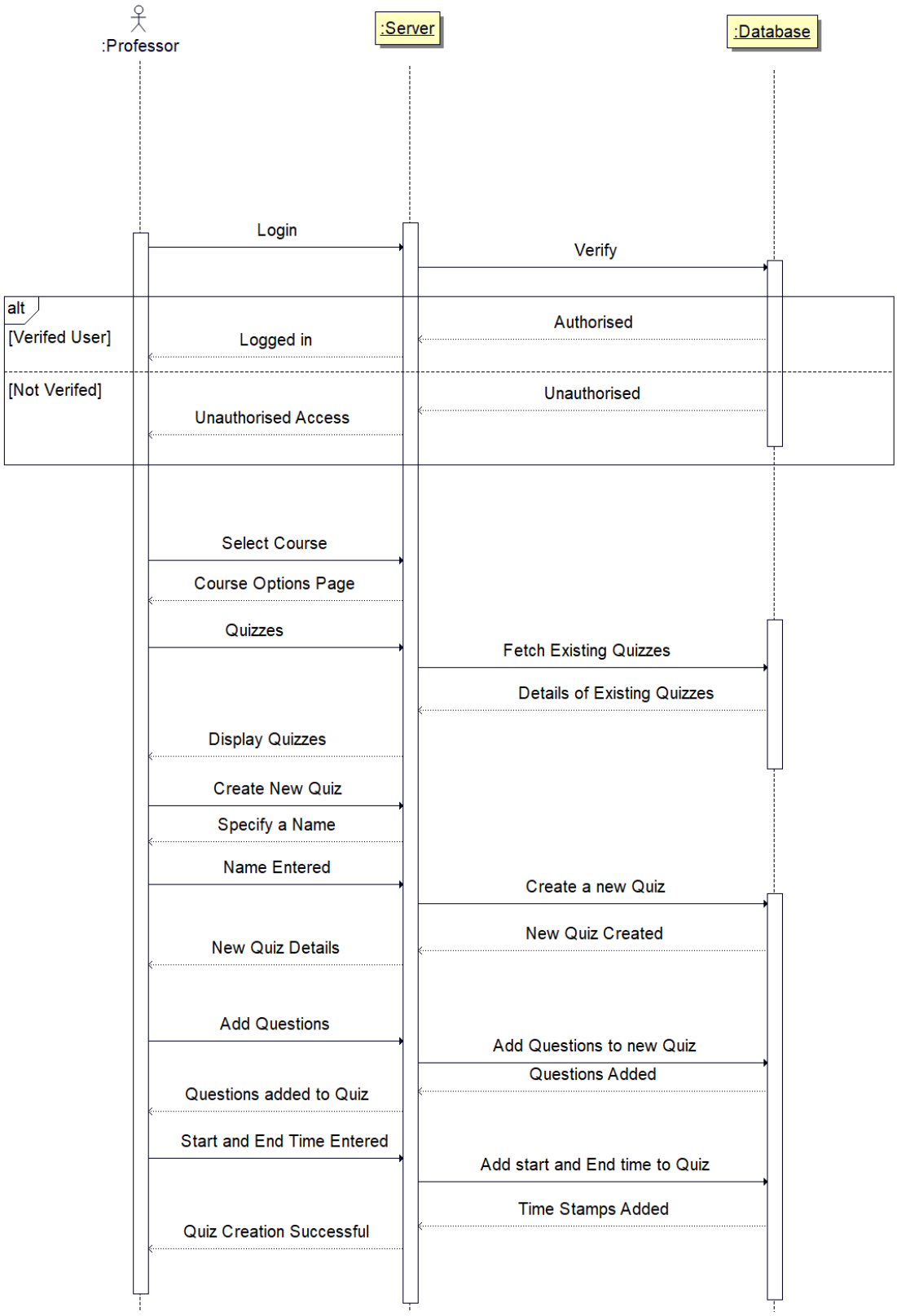
4.3 Sequence Diagrams

1. General Activities of Professor



Upon authorization, professors can do a wide array of daily administrative tasks, such as creating new courses,adding assignments etc. Each of these requests trigger a suitable GET/POST request to the flask server which controls the flow of information and database operations that follow, to cause a change(depending on the trigger input) in database and display it to the professor.

2. Create Quiz



Professor can create quiz using the create new quiz option on the quiz menu of the course portal. A new entry is added to the Quiz table. Then questions entered by the user are stored in the Questions table and options are stored in the options table. These two are linked by the Question ID.

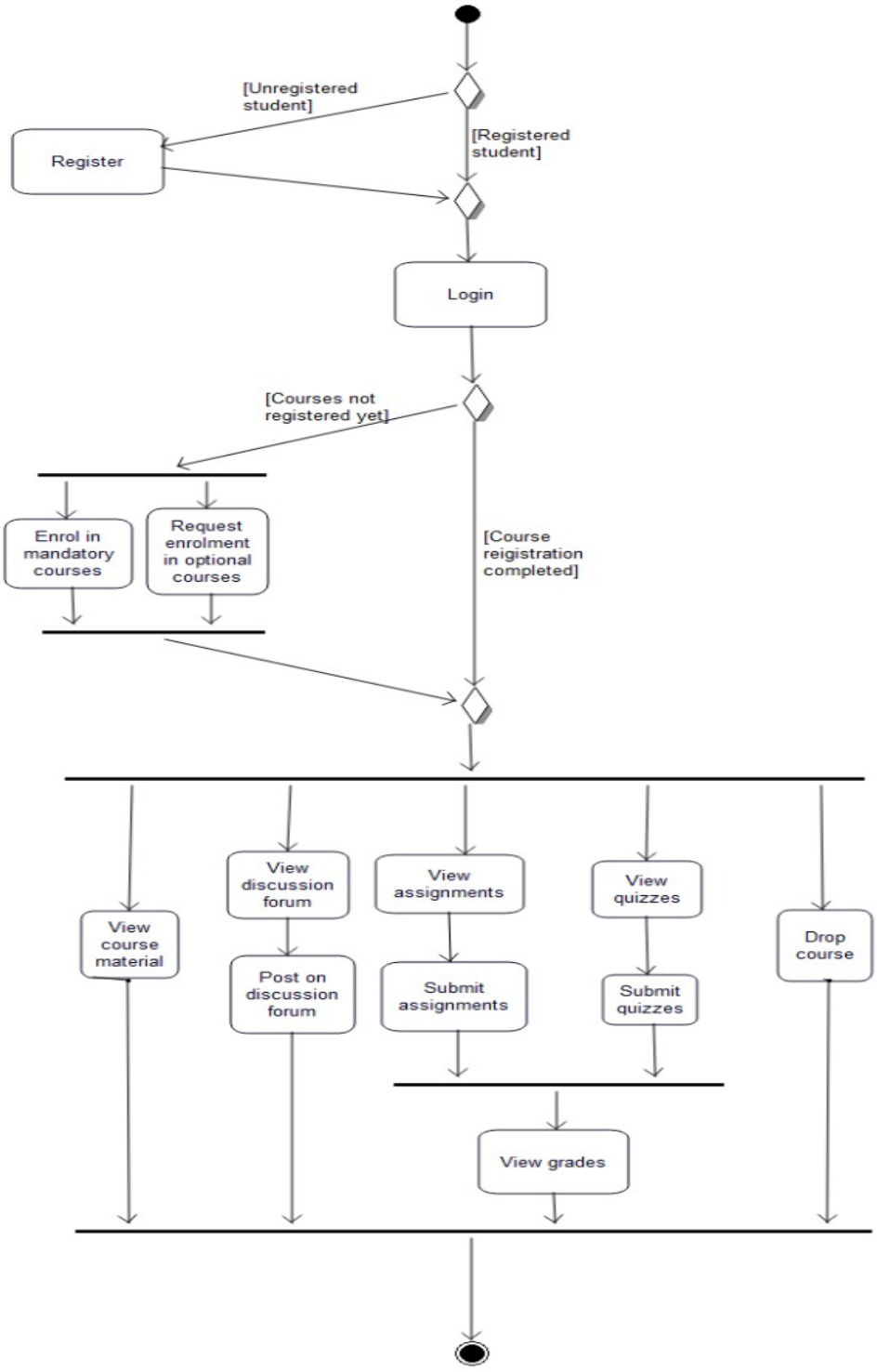
5 Student Interface

5.1 Functions Performed

Students can:

1. View Courses enrolled in
2. Request Courses to be enrolled in
3. View Course information
4. View course material uploaded by the teacher
5. View assignments
6. Submit Assignments before time
7. View grades for assignments
8. Give quizzes set by profs
9. View marks for their response in a quiz
10. View posts of students in discussion thread
11. Post in discussion thread of courses they are enrolled in.

5.2 Activity Diagram



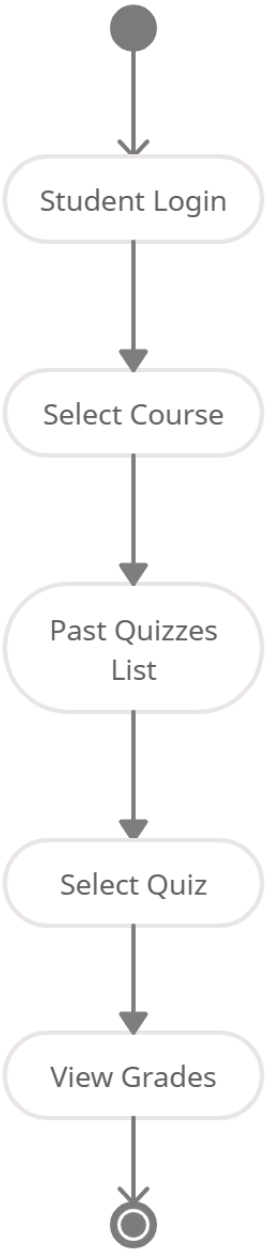


Figure 5.1: Student View Grades

To view their grades, students can select the respective course and then go to past quizzes list, where they can see all the quizzes which they have given, for this course and click on View Grades button for the corresponding quiz.

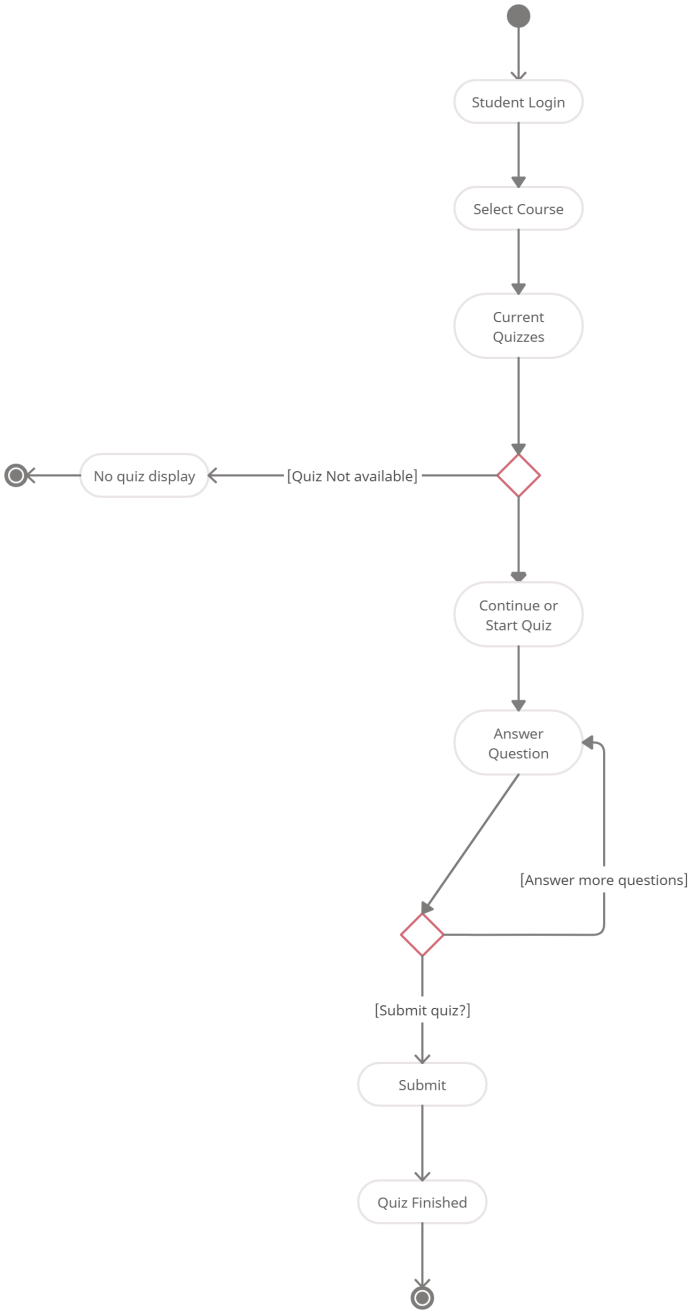
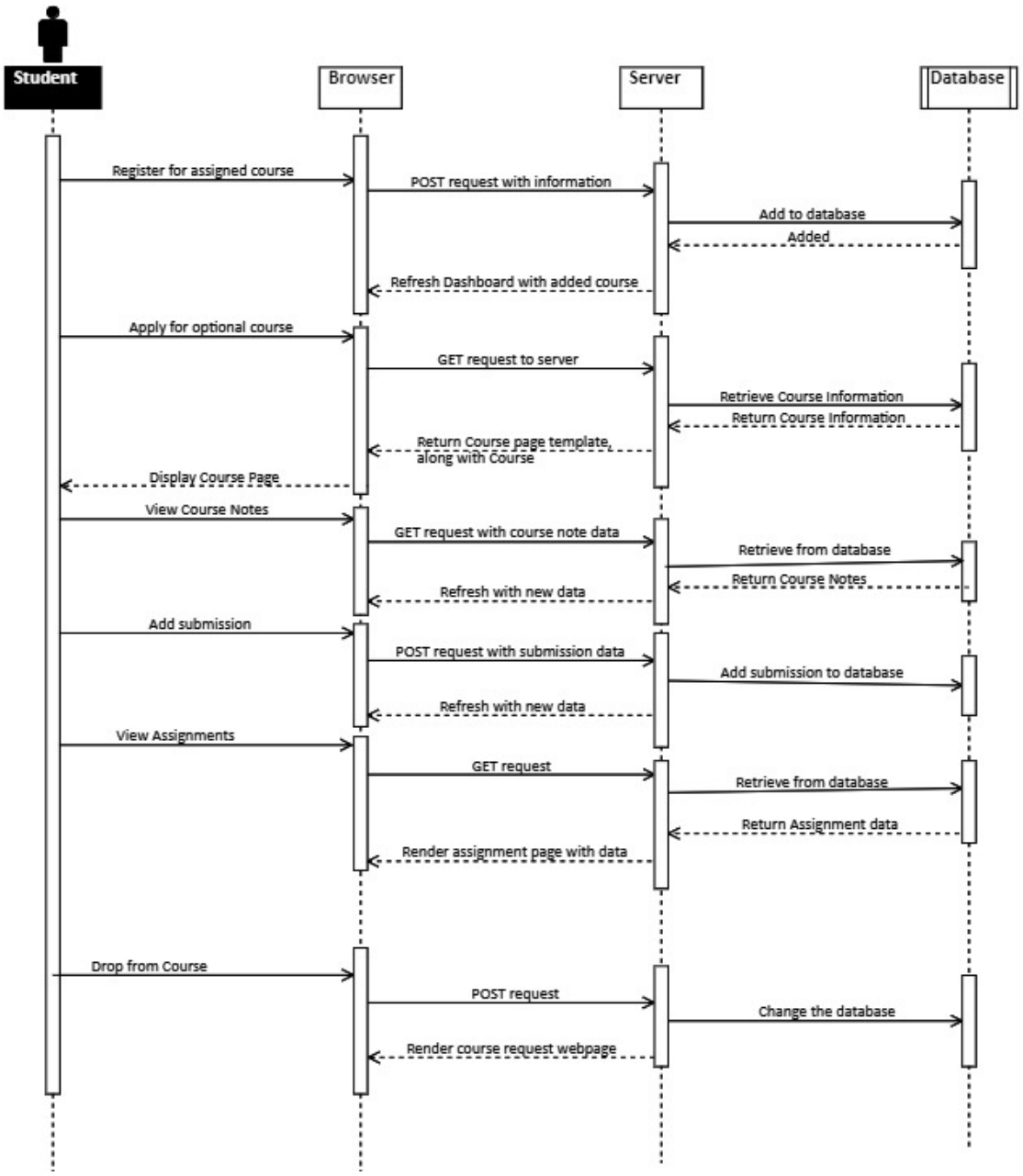


Figure 5.2: Student give Quiz

To give a quiz, students have to select the respective course and the go to current quizzes. If their are no quizzes currently, then the message will appear and they can just quit. If however there are quizzes, then they take it. They can answer questions till the time they want and if they have completed, they can submit the quiz.They will be redirected to past quizzes list where they can view their grades.

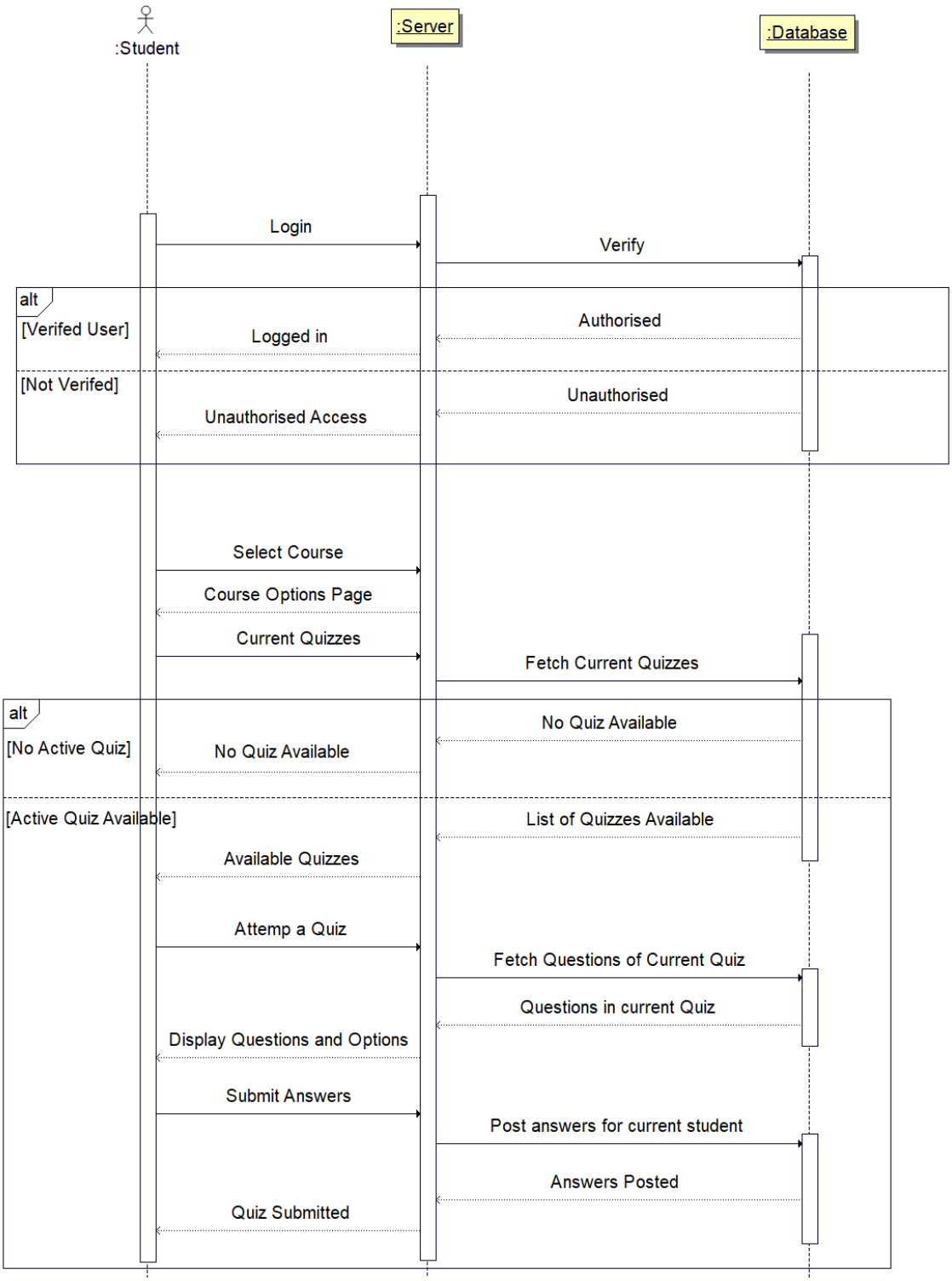
5.3 Sequence Diagrams

1. **General Activities:**



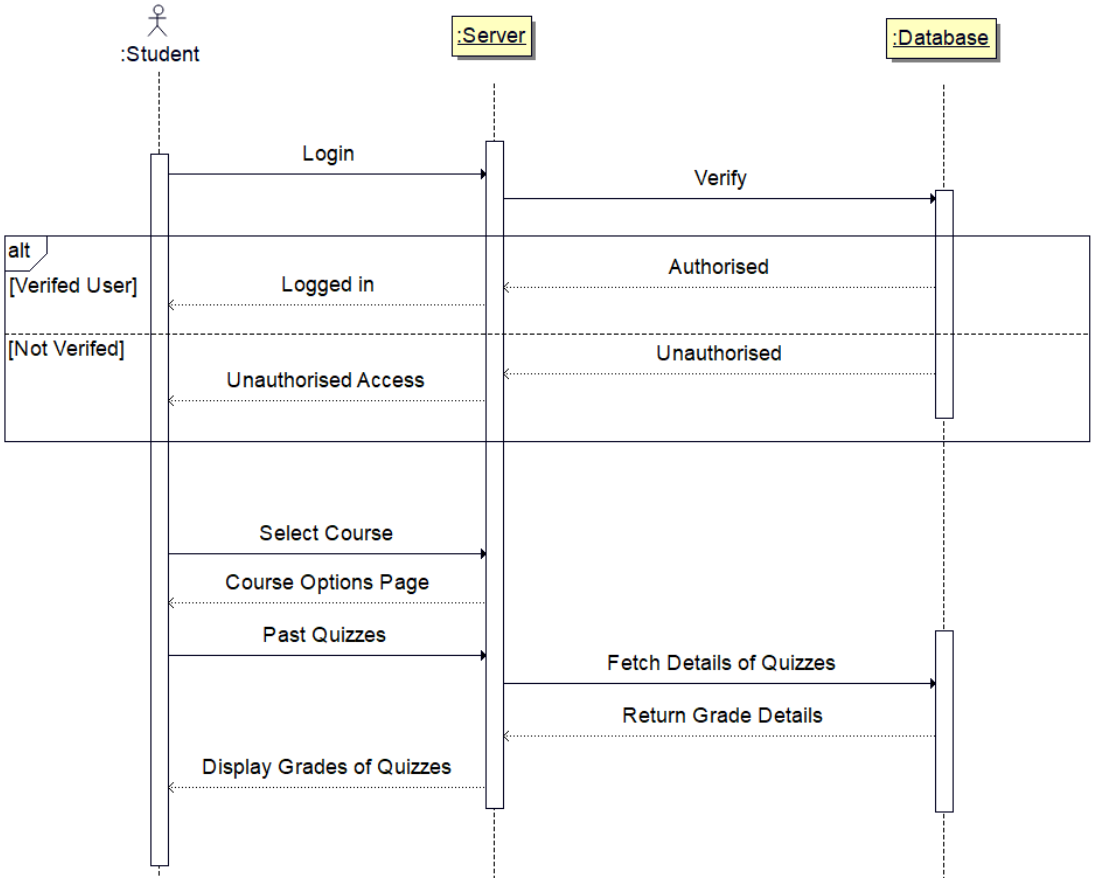
Upon authorization, students can do a wide array of tasks, such as submitting assignments,adding discussion posts, etc. Each of these requests trigger a suitable GET/POST request to the flask server which controls the flow of information and database operations that follow, to cause a change(depending on the trigger input) in database and display it to the professor.

2. **Attempting the quiz:**



Students login is checked by matching the credentials with the stored user details. If verified then student proceeds on to select course and checks for current quizzes. This data is fetched through the Quizzes table. If there exists some quiz with start time as current time then it is displayed along with start button. The Students responses are Stored in Quiz Responses table and evaluated.

3. **View the grades of Quiz:**



Students grades are fetched through the Quiz Question Response Table. This is displayed to student using the past quizzes option on the course dashboard.

6 Discussion Forum Interface

6.1 Activity Diagram

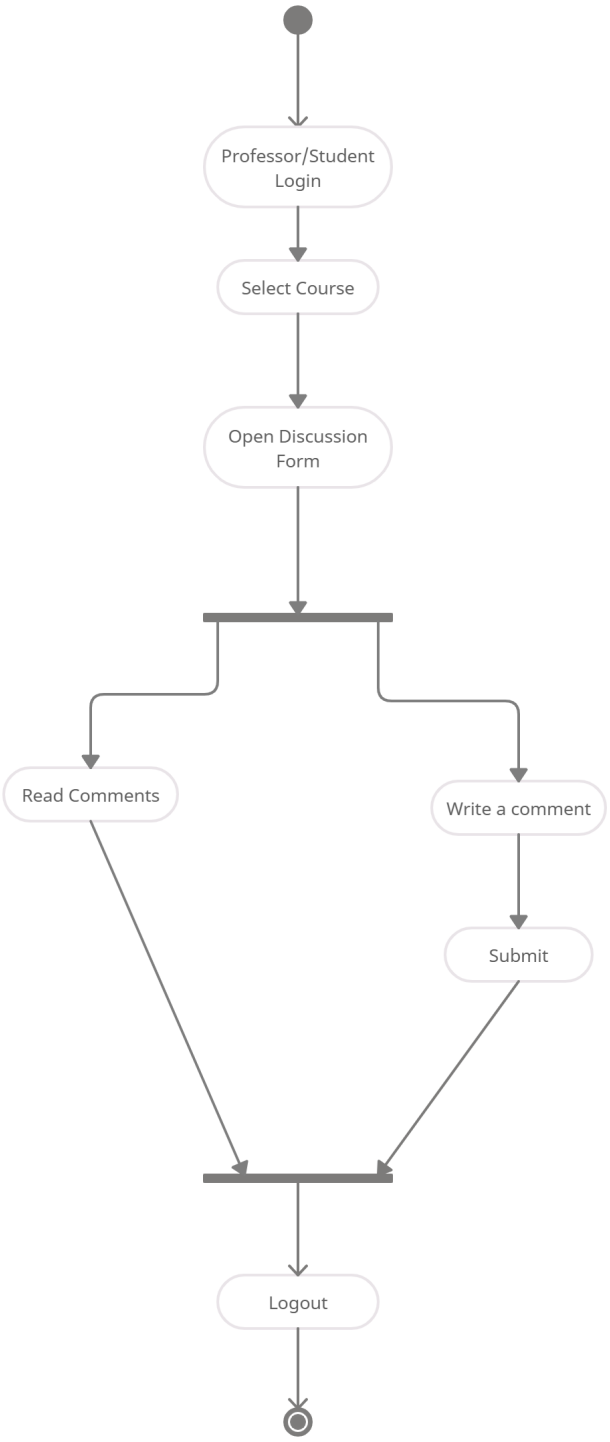
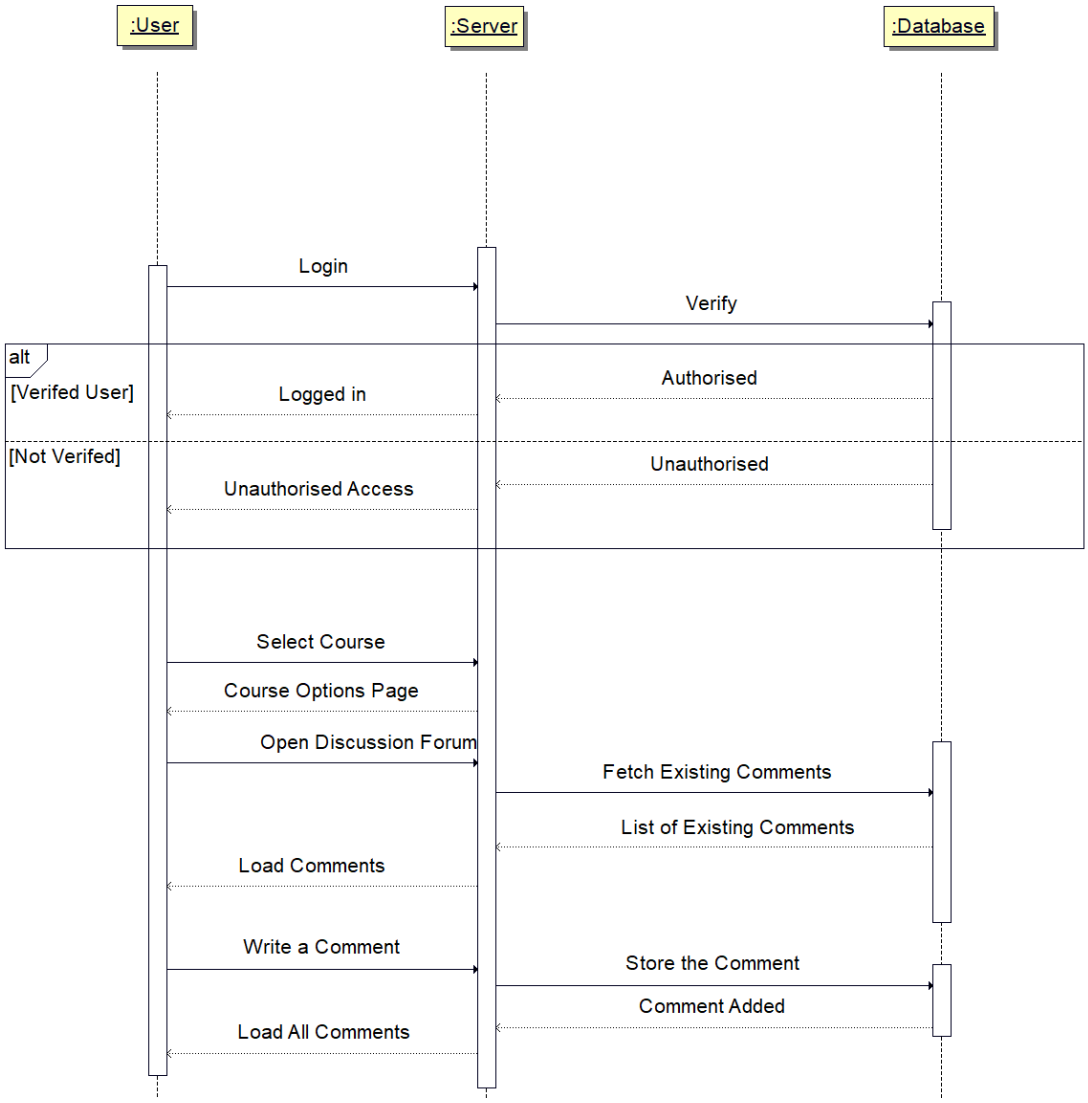


Figure 6.1: Student/Professor view discussion thread

Students and Professors can look at the existing comments on the discussion thread of the course.They can also write a comment on the discussion thread and submit.

6.2 Sequence Diagram



Here User refers to both Professor and student. Both, Professor and student can login and read the comments on the discussion forum of the course. These are fetched from the Discussion Thread table.