\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Software Design and Analysis Project Report**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Group Members:**

* Kashish Fatima 18k-0219
* Laiba Zaheer 18k-0311
* M.Talal Khan 18k-1251

**REQUIREMENT ENGINEERING**

**STAKEHOLDERS**

**Student**

Student who are accessing website to find appropriate tuition. Students are primary business actor and primary stakeholder. Student will give quizzes while learning. Student can report any complain regarding system and teachers. Student will also provide a feedback for each tuition after completion.

**Teacher**

Teacher are primary system actor and primary stakeholder. They will initiate the system by offering courses. Teacher can manage quizzes, view all register student, and receive payment. Teacher will also set their qualification and personal information.

**Admin**

Admin is our secondary stakeholder. Admin will manage whole website, and resolve all queries or complains. Admin will approved courses and offered courses also all the feedbacks are visible to admin and admin can search any teacher and student and remove them.

**BASIC FUNCTIONALITY**

* **Signup:**New student and teacher can register themselves.
* **Login:** Existing students and teachers can login.
* **Courses:** Teacher can add new courses.
* **Course Offered:** Teacher can add new offered courses.
* **Qualification:** Teacher can add and update their qualification.
* **Quizzes:** Teacher can add new quizzes.
* **Complain:** Student and teacher can report any complain regarding system and personnel.
* **Registration:** Student can register themselves into any offered courses.
* **Feedback:** Student will provide with feedback at the end of registration.
* **Search:** Student can search any course and teacher. Admin can search any teacher and student by city, country, and name.

**NON FUNTIONAL REQUIREMENTS**

* **VALID DATE**  
  For a student to registercourse date should not be less than current date.
* **REGISTRATION STATUS**  
  Registration made by a student can view the current status of that registration.
* **MANAGEMENT OF QUERIES**  
  Queries and complains provided to admin.
* **PERFORMANCE**Quick response of teacher for a registration to be accepted or rejected.
* **SECURITY**  
  Teachers and student personal data should be secure.
* **AUTHENTICATION**Student enrolled to system must be authenticated.

**FUNTIONAL REQUIREMENTS**

* **Signup:**

New student and teacher can register themselves.

* **Login:**

Existing students and teachers can login.

* **Add Courses:**

Teacher can add new courses.

* **Add Course Offered:**

Teacher can add new offered courses.

* **Update Qualification:**

Teacher can add and update their qualification.

* **Manage Quizzes:**

Teacher can add new quizzes.

* **Add Complain:**

Student and teacher can report any complain regarding system and personnel.

* **Registration:**

Student can register themselves into any offered courses.

* **Add Feedback:**

Student will provide with feedback at the end of registration.

* **Search:**

Student can search any course and teacher. Admin can search any teacher and student by city, country, and name.

* **Remove:**

Admin can remove student and teacher.

* **Change Status:**

Admin can approve or reject any course and course offered.

**Domain Requirement:**

* Student must be logged in to register a course.
* Courses must be approved before approval of offered courses.
* Course offered must be approved before registration.
* Feedback must be provided for each registration.
* To add any quiz registrations must exist.

**Use Case Narration:**

**NARRATION 1**

**Use Case Name:**

Register Student

**Use Case Description:**

When a student visit website, student can login if student have an account else student will create a new account and then student will be able to register into any courses.

**Input:**

Click “Login” button to login then click “Sign up” button to create new account.

Click “Register” button to register into any course.

**Primary Actor:**

Student

**Other Actor:**

Admin

**Stakeholders:**

1. Student: to login to the system.
2. Admin: to view new register student.

**Pre -conditions:**

Student is not login to system.

**Flow of event:**

1. Student click Register button to register into course.
2. System will direct the student to login page.
3. Student input details to create account.
4. Account is created successfully.
5. Student will be displayed to admin.
6. Student will be register into course.

**Alternative event:**

1.1. Student click login button.

3.1. Student login to system.

**Exceptional event:**

3.1. Student enter incorrect details.

3.2. System will display an error.

**Post condition:**

System will create account and will be directed to course registration page.

**Output:**

Student will be logged in or register in the course.

**NARRATION 2**

**Use Case Name:**

Register a course

**Use Case Description:**

A student can view all the approved courses offered by all the teacher and register to any teacher on the time specified by a teacher.

**Related Use case:**

Includes: login, sendPayment, verifyTeacherAvaibility.

**Input:**

Click “Register” button.

**Primary Actor:**

Student

**Other Actor:**

Teacher, PaymentGateway.

**Stakeholders:**

1. Student: to register in a course.
2. Teacher: view all registration.

**Pre- conditions:**

Student must be registered and logged in.

**Flow of event:**

1. View all course offered.
2. Register in to any course.
3. System will check teacher availability at selected time.
4. System will request payment.
5. Payment is received through payment gateway.
6. Registration will be displayed to teacher.

**Exceptional flow of event:**

3.1. Teacher is not available.

3.2. System will redirect to register page.

4.1. Payment is not received.

4.2. Registration will not be displayed to teacher.

**Post condition:**

Teacher will begin teaching and take quizzes until registration ends.

**Output:**

Registration is completed.

**NARRATION 3**

**Use Case Name:**

Add course offered

**Use Case Description:**

Teacher can choose a course that they desire to teach.

Teacher can add a course that they desire to teach.

**Related Use case:**

Includes: course status.

Extend: View Course offered.

**Input:**

Teacher will click course offered button.

Teacher will input all the required fields.

**Primary Actor:**

Teacher

**Other Actor:**

Admin,Student.

**Stakeholders:**

1. Teacher: Add courses and desired courses to teach.
2. Admin: Approved courses and offered courses.
3. Student: Can register into offered courses.

**Pre- conditions:**

Teacher must be logged in.

**Flow of event:**

1. View all course offered.
2. Select courses.
3. Add details of the selected course.
4. System will send request to admin for approval.
5. Approved courses will be shown to student.

**Alternative event:**

2.1. Teacher add new course.

**Exceptional flow of event:**

2.1. Teacher add already existed course.

2.2. System will display error.

**Post condition:**

Admin will have to approve or reject the requested course and course offered.

**Output:**

Course offered will be added.

**NARRATION 4**

**Use Case Name:**

Update course offered status

**Use Case Description:**

Admin is requested to approve the newly offered course by teacher.

**Related Use case:**

Extend: View all Course offered.

**Input:**

Admin view all offered courses.

**Primary Actor:**

Admin

**Stakeholders:**

1. Admin: To approve offered courses.
2. Student: New offered courses will be available to them.

**Pre- conditions:**

Courses must be offered.

**Flow of event:**

1. View all offered course.
2. Admin will approve offered course.
3. Approved courses will be displayed to student.

**Alternative event:**

* 1. Admin reject the offered course.

**Post condition:**

Approved courses will be displayed to student.

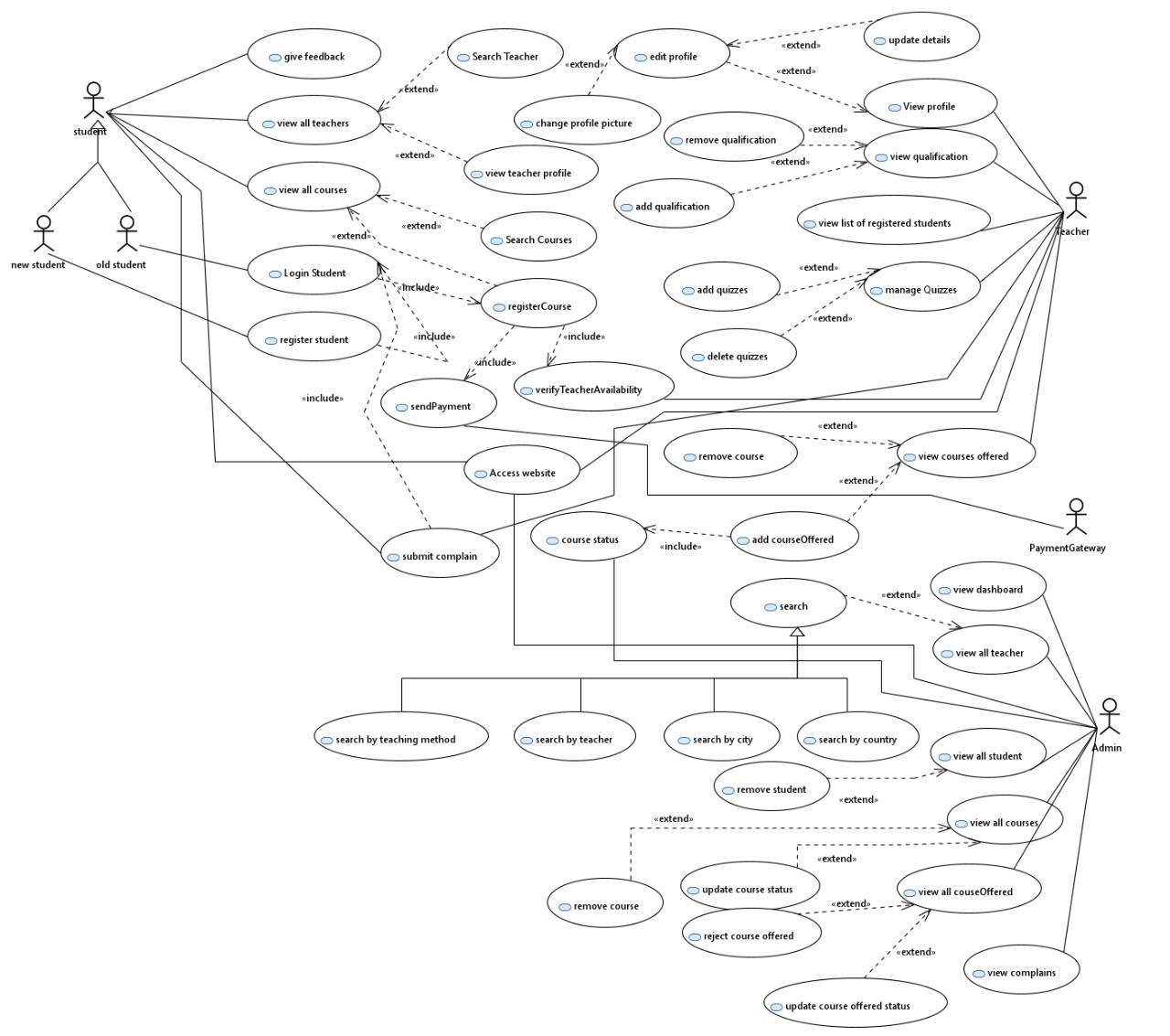
**Output:**

Course offered status will be change.

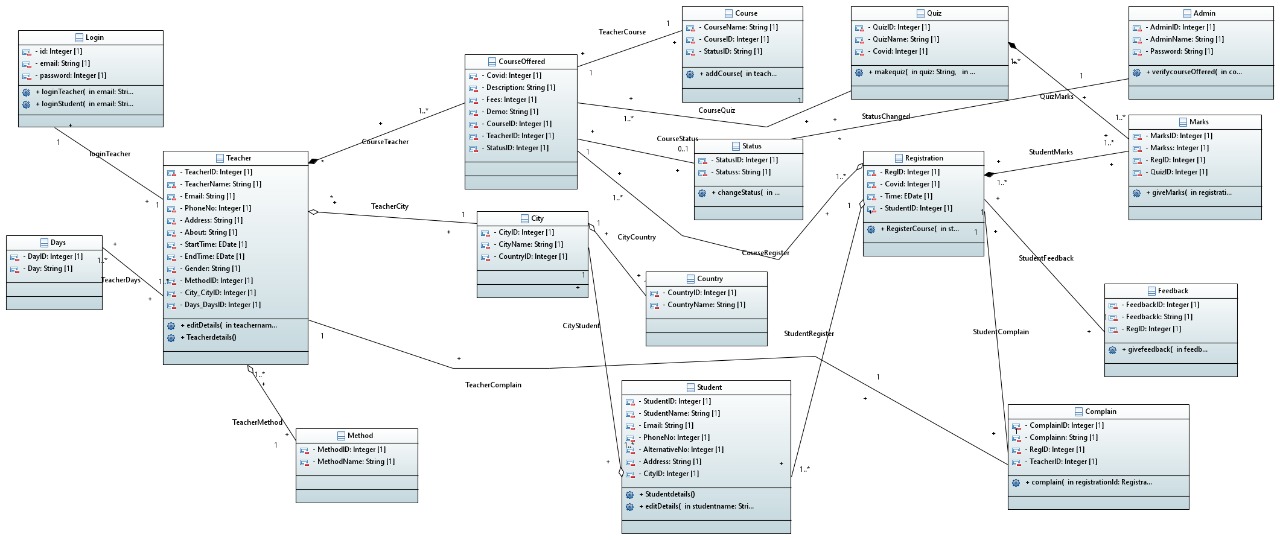
**Entity Class Description:**

* **Student:**Contain all student data.
* **Teacher:**Contain all teacher data and their qualification.
* **Login:** Contain all users’ login id and password.
* **Days:** Contain days for teacher to teach.
* **City:** Contain all cities from which student and teacher belongs.
* **Country:** Contain all countries of all cities.
* **Course offered:** Contain all the offered courses by teacher whether they are approved or not.
* **Course:** Contain all courses.
* **Quiz:** Contain all quizzes taken by all teacher.
* **Marks:** Contain marks of each student in each quizzes.
* **Registration:** Contain all registration in all approved offered course.
* **Feedback:** Contain all feedbacks for all registration.
* **Complain:** Contain all complains by student and teacher.
* **Admin:** Contain admin id and password.
* **Status:** Contain statuses of course.

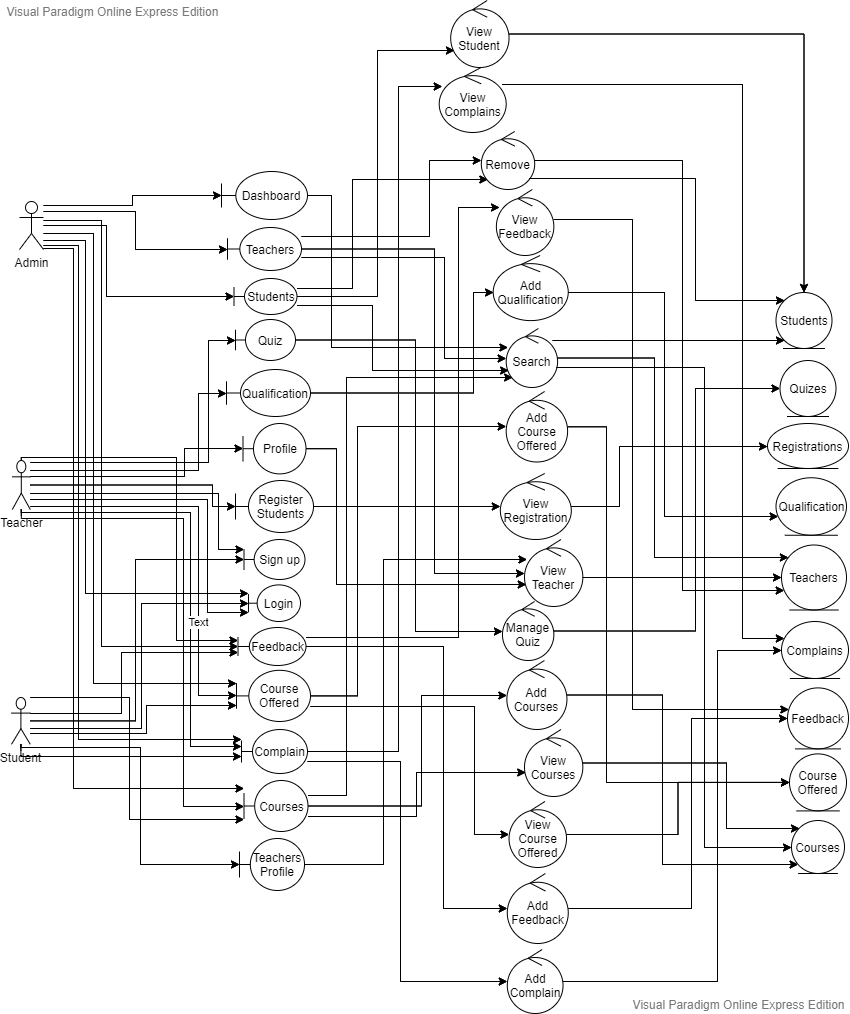
**Use Case Diagram:**



**Class Diagram:**

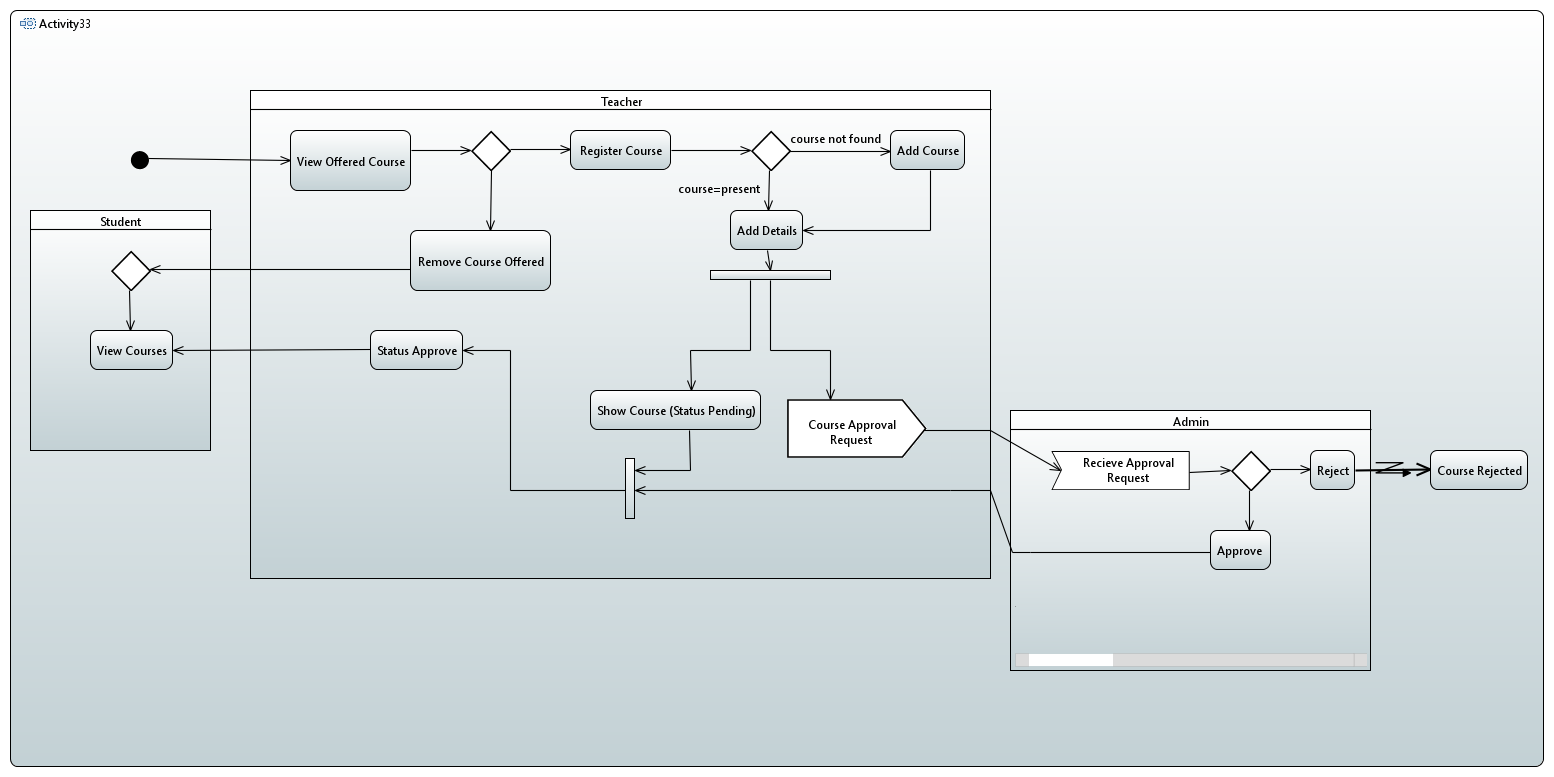
****

**ECB Robustness:**

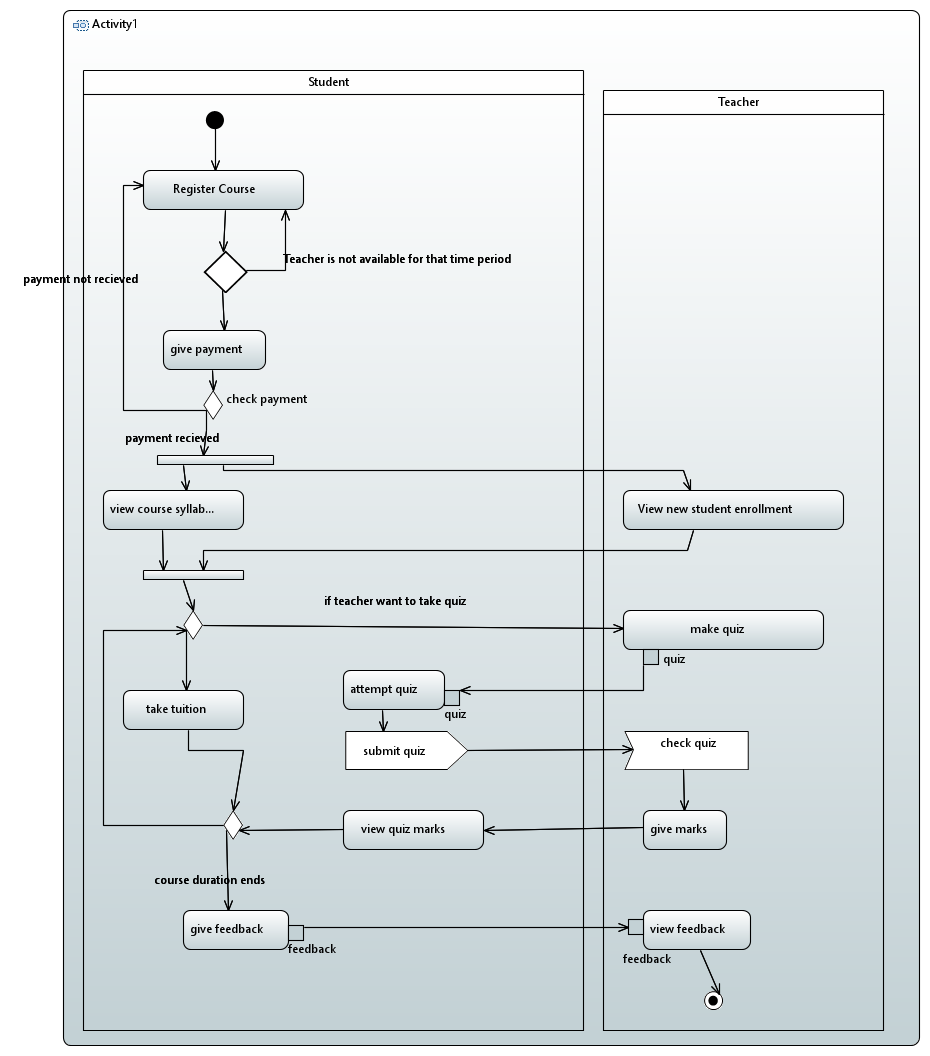
****

**Activity Diagram:**

* Course Offered by teacher.

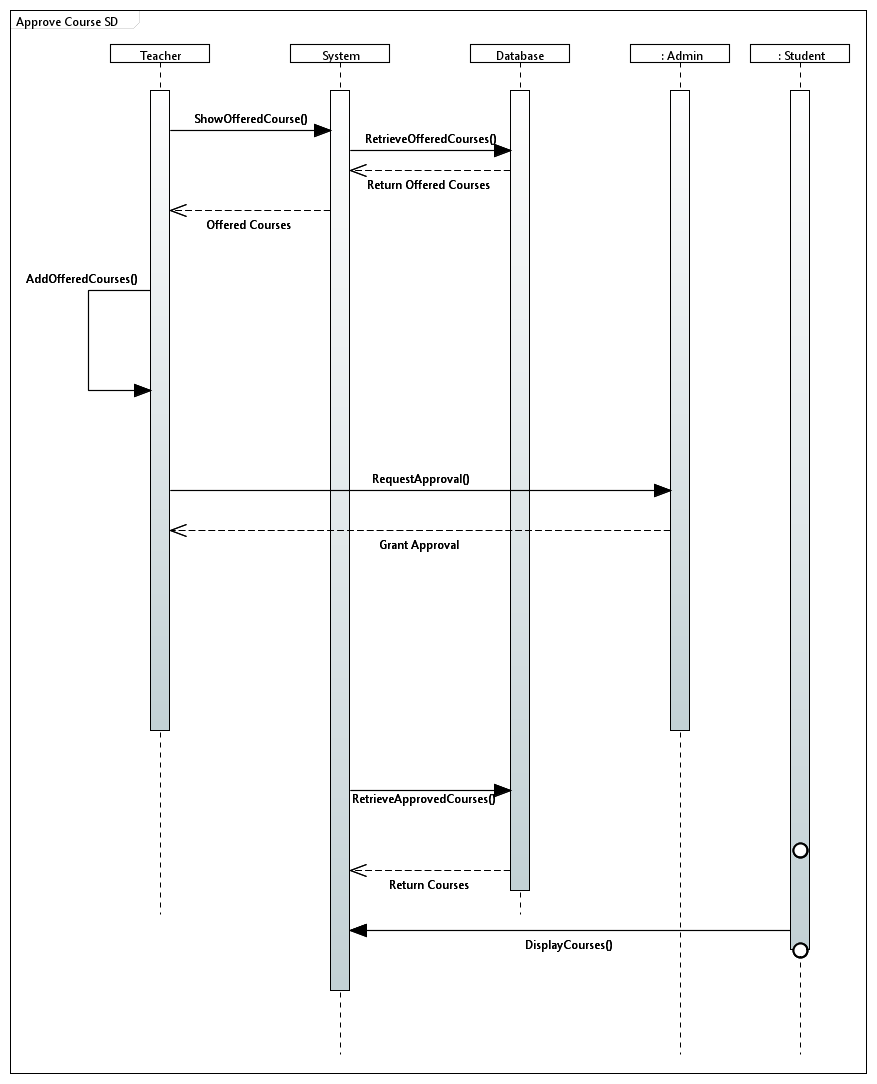


* Registration by student.

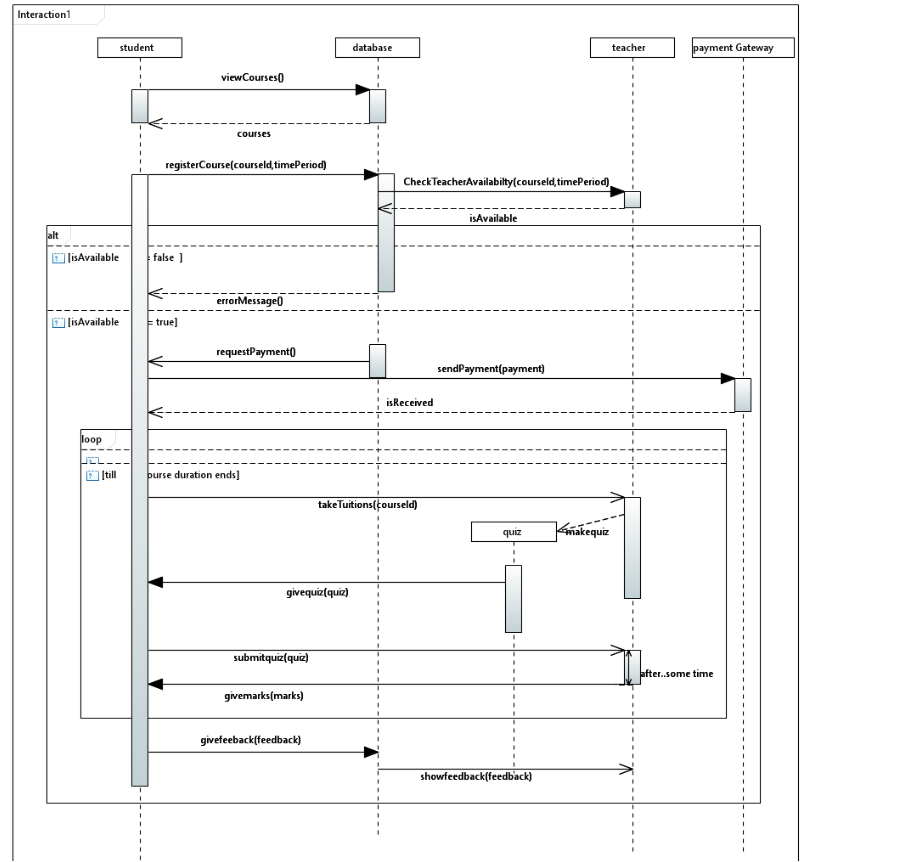


**Sequence Diagram:**

* Course Offered by teacher.

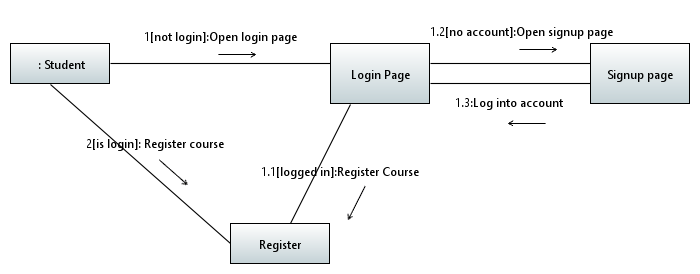


* Registration by student.

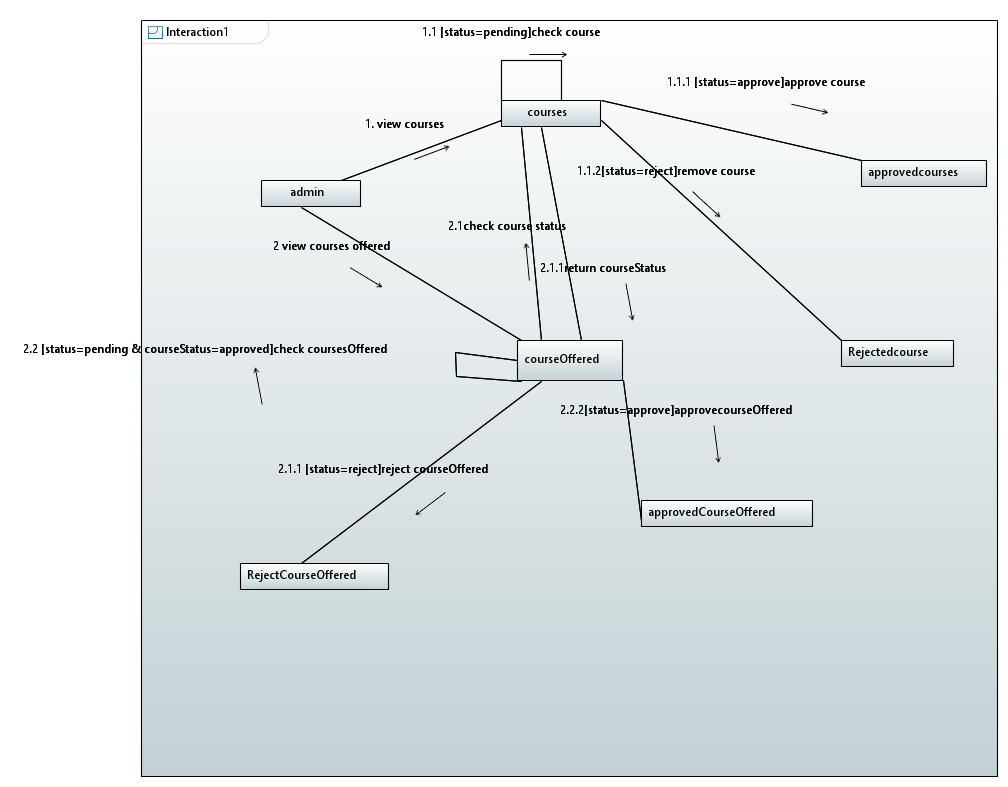


**Collaboration Diagram:**

* Registration requirement.

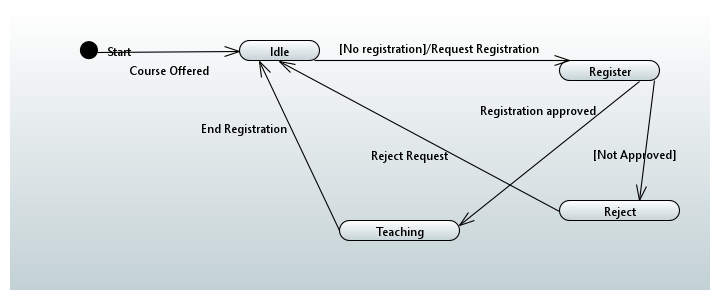
****

* Approval of courses and course offered by Admin.

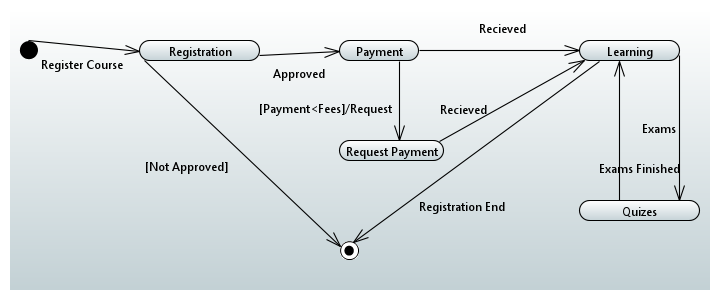
****

**State Diagram:**

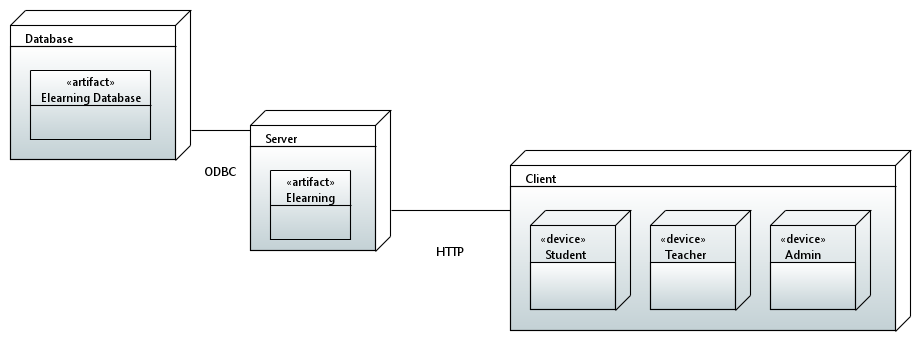
* Teacher state during registration.



* Student state during registration.



**Deployment Diagram:**

****