*A blue and white logo

Description automatically generated with low confidenceGroup members:*

Lê Thanh Phương Nam - ITITWE19025

Đặng Khải Dương - ITITWE19010

Đinh Bình Thanh Thông - ITITWE19027

**Group Report Lab 6**

**Topic 1: The New Student Enrollment Process**

**Topic 2: The student stand-downs/suspensions/ exclusions/expulsions process**

**Topic 3: The Graduation Application Process**

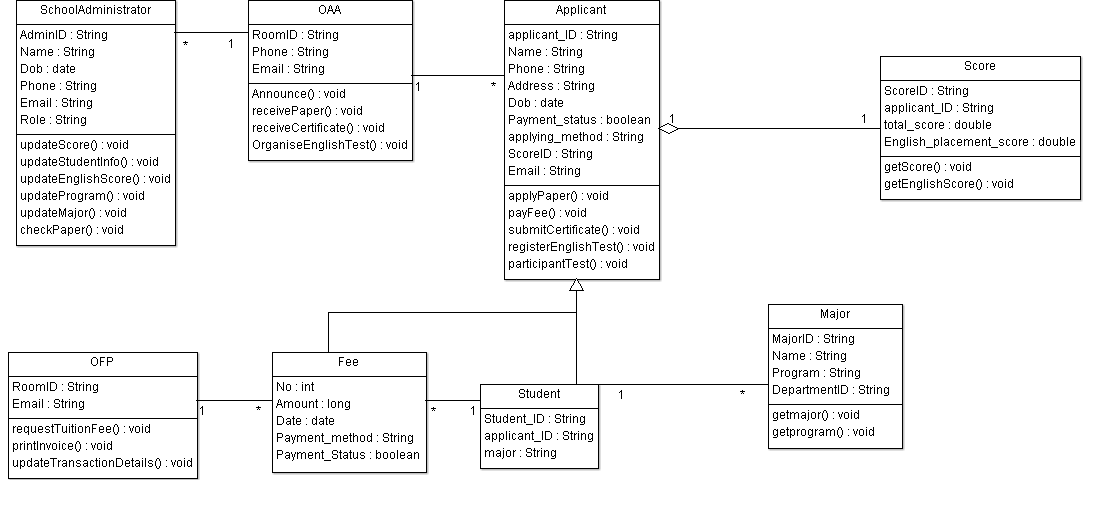
**Topic 4: The Course Registration Process.**

**Topic 5: The Tuition Fee Payment Process**

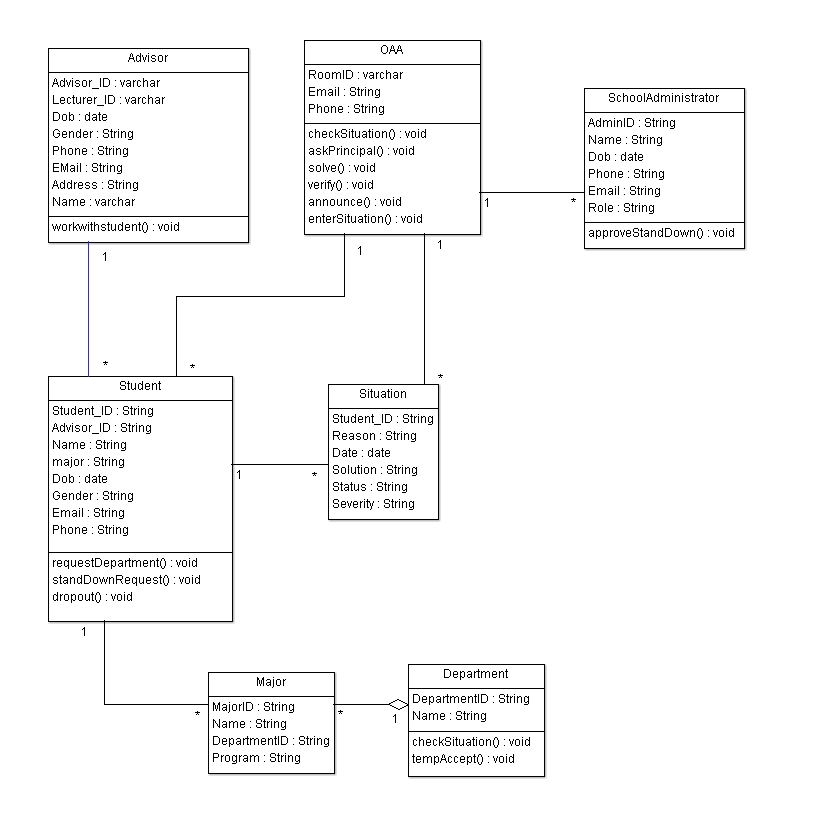
**Topic 6: The Grading Process for students**

1. **Class Diagram:**

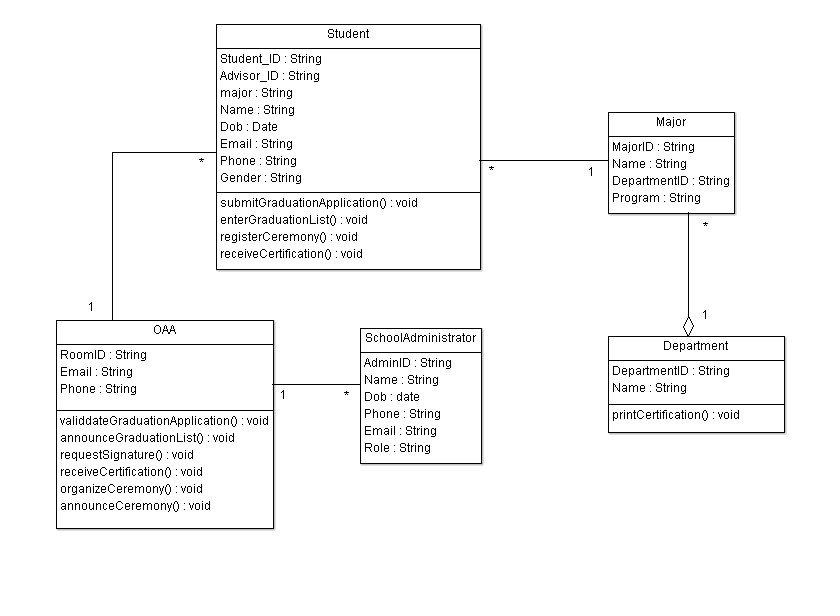
**Topic 1:**

****

**Topic 2:**

****

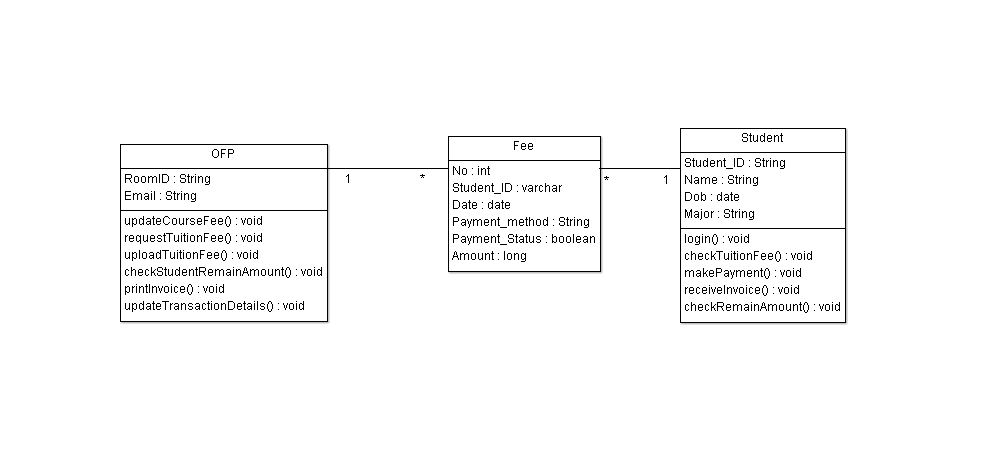
**Topic 3:**

****

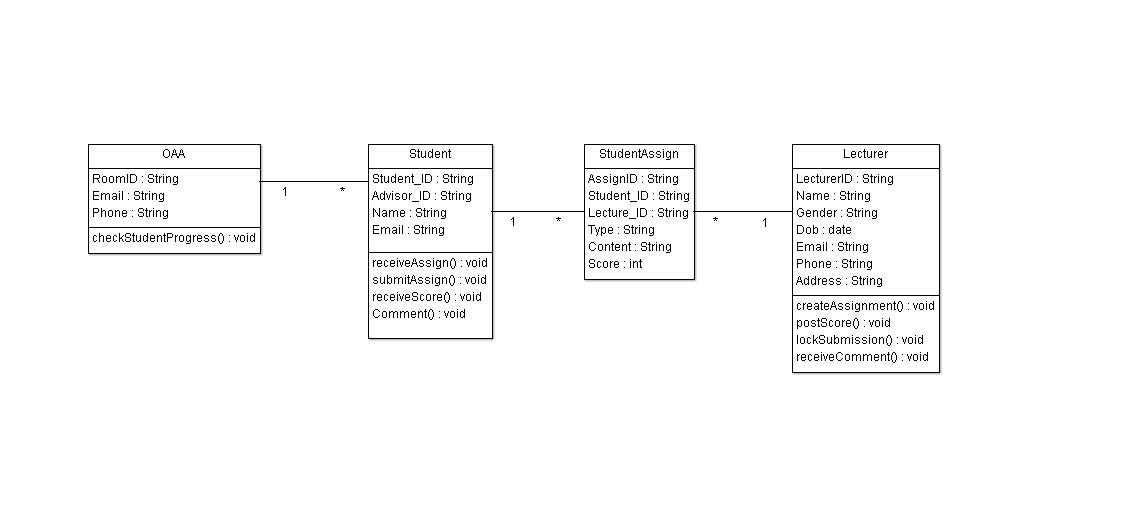
**Topic 4: A picture containing timeline

Description automatically generated**

**Topic 5:**

****

**Topic 6:**

****

1. **Relationships between Entities in processes**

**Analysis:**

From each topic's separate class diagram in section 1. The relationships between the entities (classes) can be examined as follows:

**Topic 1:**

* OAA- Applicant is a one-to-many relationship since OAA has many SchoolAdministrator who work with multiple applicants at the same time. Therefore, OAA-SchoolAdministrator is a one-to-many relationship
* Applicant-Score is a one-to-one relationship because each applicant can only apply with one score and vice versa In this case, the score is an employee because its components, such as the total score or the English entrance score, are attributes, and the score itself has the matriculate ID to distinguish it from the scores of other applicants.
* Student is a one-to-many relationship between the major entity and the entity inherited from the Applicant. In this case, the major entity is (\*) because it contains the program containing the course and the student ID to identify which student belongs to the major. Because one student can have as many majors as they want, the major entity is (\*).
* Fee entity acts as a connector between the OFP and the Student, and there will be a one-to-many relationship between the OFP entity and the Fee entity, as well as a one-to-many relationship between the Student and the Fee entity. The reason is the same as in topic 4.

**Topic 2:**

* Advisor-Student is a one-to-many relationship because many students are grounded, one advisor can overlook many students.
* Student-Situation is a one-to-many relationship since students can perform many actions, resulting in many situations to be concerned about.
* The relationship between the Student and the Department appears once more in this process, but this time the Major entity appears. The Major entity, in this case, would be the link between the Student and the Department, with the Student-Major relationship being one-to-one and the Major-Department relationship being one-to-many, with the department being the single entity because one Department has many majors. The majors in the IT department, for example, are Network Engineering, Computer Engineering, Data Science, and so on.
* OAA-Situation has a one-to-many relationship with the school administrators because OAA is the only entity in this case and is responsible for all of the situations that occur.
* OAA also supervises many school administrators, so OAA has a one-to-many relationship with the school administrators. In this case, the school administrators are also considered an entity.

**Topic 3:**

* OAA-Student is a one-to-many relationship with the OAA being a single entity because it handles all student applications.
* The Major-Student has a one-to-many relationship and The Department-Major has a one-to-many relationship. The reason is the same as topic 2.
* Finally, OAA-SchoolAdministrator has a one-to-many relationship. The reason is the same as topic 1.

**Topic 4:**

* Lecturer-Department is a one-to-many relationship with the, in this case, the department is the one. This is the same reason as Topic 2, but unlike students, the lecture has a direct relationship with the department.
* The Major-Student has a one-to-many relationship and The Department-Major has a one-to-many relationship. The reason is the same as topic 2.
* One major can have many courses, and in this case, the Course can be considered an entity, and the Major entity and the Course entity have a one-to-many relationship.
* To access the course, we must first register for it, and in this case, the registration can also be considered an entity, so the Course and the CourseRegistration have a one-to-many relationship. However, because the CourseRegistration entity is embedded in the relationship between the Course and the Student, which is already established through the Major connector entity, it will not be directly involved in the system.
* Finally, the OAA and the students have a one-to-many relationship because the OAA is in charge of all students "registration."

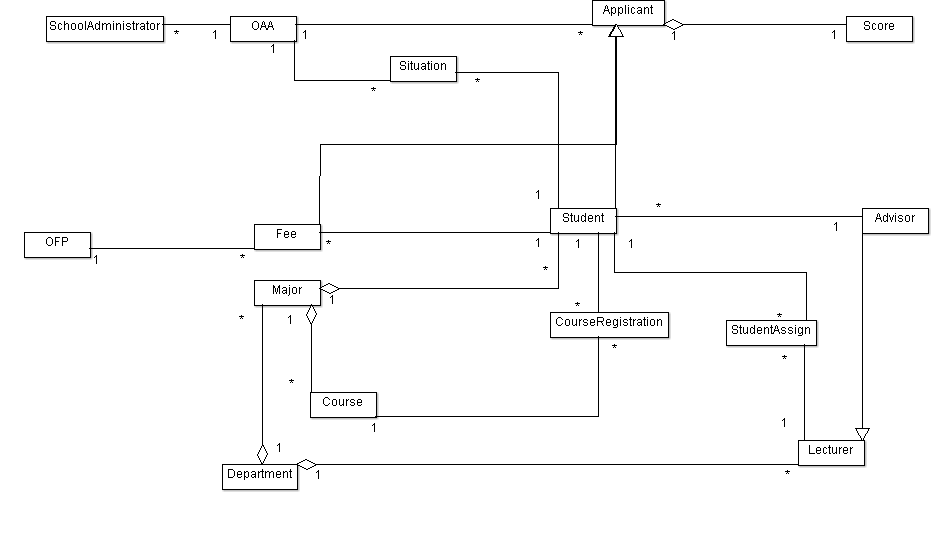
**Topic 5:**

* The Student will primarily work with the OFP in this process, but the Student does not belong to the OFP, so the Fee entity acts as a connector between the OFP and the Student, and there will be a one-to-many relationship between the OFP entity and the Fee entity, as well as a one-to-many relationship between the student and the fee entity. Because it requires the “studentID” as a unique attribute to distinguish its ownership and the date when the fee is sent in among the students in the system, the fee can be considered an entity in this case.

**Topic 6:**

* OAA-Student is one-to-many relationships
* Because the Student will work with the Lecturer but does not belong to the Lecturer, a connector will be required. As a result, Student's Assign will serve as a link between students and lecturers. Therefore, The Lecturer-StudentAssign has a one-to-many relationship and The Student-StudentAssign has a one-to-many relationship.

**Based on the analysis above, we get the relationship system in the 6 processes as follows**

****

1. **Database designs**

The normal form of the relationship system has the 2NF because all entities have a primary key (1NF) and non-key attributes rely on the key attribute from which the database is retrieved. However, to achieve 3NF, we would need to further normalize the system. To accomplish this, we must make the non-key attribute directly dependent on the key attribute.

**Based on the relationship system and the normalization, we can come up with this database:**

Because all entities have a primary key (1NF) and non-key attributes rely on the key attribute from which the database is retrieved, the normal form of the relationship system has the 2NF. However, to achieve 3NF, the system would need to be further normalized. To achieve this, the non-key attribute must be directly dependent on the key attribute.

Qr code

Description automatically generated