Student Registration System

1-) Description

This simulation system design will help our university to see the bottlenecks in registration process such as course capacity, prerequisite courses, and available course sessions etc. With the help of computers, we can implement such system to consider those limitations and avoid mistakes in the process.

This student registration simulation will consider those limitations given above and simulate registering students to their desired courses. This simulation is for undergraduate students to properly enroll courses according to their curriculum. Also, this simulation gives insights to departments to manage each course that students take if there is any necessary condition.

2-) Requirements

Functional Requirements

1. Knowing course information
    - a. Courses will have credits.
    - b. Courses might have pre-requisite courses.
    - c. Courses will have quota value.
    - d. Courses will have available semester
    - e. Courses will have available lecture hours.
    - f. Courses will have available sessions.
    - g. Courses might have required completed credits.

2. Knowing student information
    - a. Students will have a transcript.
    - b. Students will have semester information.
    - c. Students will have advisor.

3. Knowing transcript information
    - a. Transcript will have students passed & failed courses.
    - b. Transcript will have total completed credits of the student.

4. Course enrollment
    a. Checking if student can take the selected courses one-by-one.
        i. If student can take the course, advisor will approve, and course will be added to student's schedule.
        ii. If student cannot take the course, inform the student.

5. Show current courses for each student.

Non-Functional Requirements

1. Each process of registration should be logged.
2. Each process of registration should be prompted to CLI.
3. System should take input as a JSON file.
4. Program should be implemented with Python.

3-) Business Rules

- To register a course, student must pass its pre-requisites.
- To register TE courses and graduation project student must be completed the required credits.
- To register FTE courses, student must be either at SPRING semester or must be at his/her graduation state.
- To register a course, that course's lecture hour can only collide with another course's hour at most for 1 hour.
- To register a course, that course's quota must not be exceeded.

4-) Use Case (Fully Dressed)

Simulate the course enrollment process.

Actor: User.

1. User starts the simulation.
2. Courses will be created.
3. Students will be created one by one.
4. Student will select courses according to semester's curriculum.
5. Student send selected courses to the advisor one by one in order to get their approval.
6. Advisor controls student's selected course.
7. Advisor approves student's courses.
8. Approved course added to student's schedule.

Alternative 7: Failure at approving

7a-) At step 7, Selected course's quota is full therefore student cannot take that course. Advisor won't approve that course, informs the student, and drops that course from student's course list

7b-) At step 7, Selected courses' lecture hours collides with each other. Advisor will approve the course that repeating or from lower semester.

7c-) At step 7, Student haven't passed pre-requisite(s) of selected course. Advisor won't approve that course, informs the student, and drops that course from student's course list.

7d-) At step 7, Student haven't completed enough credits for selected course. Advisor won't approve that course, informs the student, and drops that course from student's course list.

7e-) At step 7, Student course list contains technical elective (TE) course, although he/she already took 2 TEs in FALL semester. Advisor won't approve that course, informs the student, and drops that course from student's course list.

7f-) At step 7, Student course list contains technical elective (TE) course, although he/she already took 3 TEs in SPRING semester. Advisor won't approve that course, informs the student, and drops that course from student's course list.

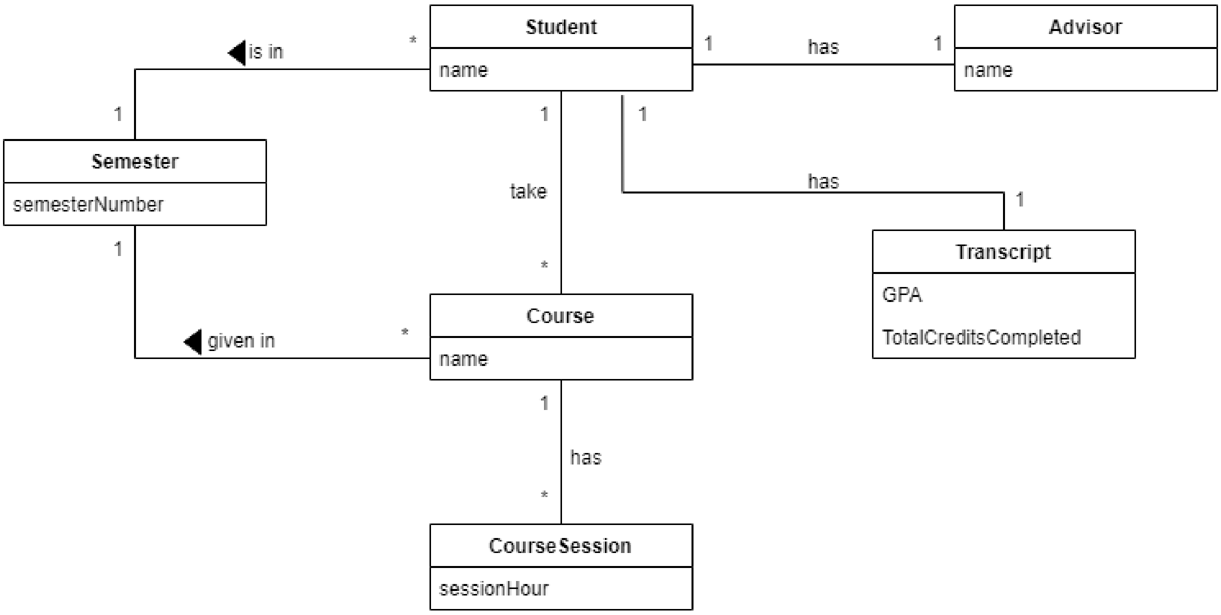
7g-) At step 7, Student course list contains faculty technical elective (FTE) course in FALL semester, and he/she is not graduating this semester. Advisor won't approve that course, informs the student, and drops that course from student's course list.

5-) Project Plan

On this project, we will implement the software such as.

- Requirement analysis about the current student registration system.
- Creating a design from the information that has been gathered from requirement analysis.
- Implementing student registration system.
- Implementing advisor's approval system.
- Implementing semester simulation system.
- Implementing log tracing system.
- Implementing course registration limitations.
- Implementing the unit test.

6-) Domain Model

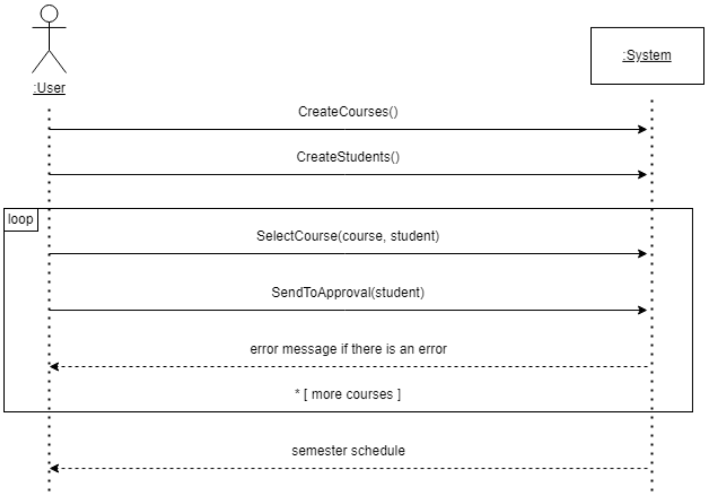


7-) System Sequence Diagram

Simulate the course enrollment process.

Actor: User.

1. User starts the simulation.
2. Courses will be created.
3. Students will be created one by one.
4. Student will select courses according to semester's curriculum.
5. Student send selected courses to the advisor one by one in order to get their approval.
6. Advisor controls student's selected course.
7. Advisor approves student's courses.
8. Approved course added to student's schedule.



8-) Glossary

- Advisor: Actor who approves or disapproves student’s course selection list.
- CLI: Command line interface.
- Course: Lessons students must have passed to graduate.
- Course Session: Different lecture hours available for same lesson.
- Credits: Weight impression of the lecture.
- Curriculum: Overall content of the course.
- FTE Courses: Faculty Technical Elective.
- JSON File: Standardized data storing format.
- Log: Storing sequential data.
- NTE Course: Non-Technical Elective.
- Pre-requisite courses: Courses required to be past in order to get the connected course.
- Python: Programming language.
- Schedule: Student’s courses in a weekly plan.
- Semester: A half year term in a university.
- Student: Main actor of the system.
- TE Course: Technical Elective.
- Transcript: Lecture records of the student.
- UE Course: University Elective.
- Unit: Each meaningful piece of the software.
- Unit Test: Testing the units of the software.