

Student Registration System

1-) Description

This system design is required in our university since registration process with paper and pen will take too long and it could be hard to determine other limitations 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 system will consider those limitations given above and register students to their desired courses. This system is for undergraduate students to properly enroll courses according to their curriculum. Also, this system gives control to departments in order 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 will 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 will have required completed credits.
- 2.** Knowing student information
 - a.** Students will have passed classes.
 - b.** Students will have failed classes.
 - c.** Students will have semester information.
 - d.** Students will have advisor.
 - e.** Students will have total completed credits.

3. Course enrollment
 - a. Checking if student can take the selected courses.
 - 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.
4. 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 JAVA.

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)

Enroll a course

Actors: Student, Advisor, System.

1. Student will select courses according to semester's curriculum.
2. Student sends list of selected courses to his/her advisor.
3. Advisor controls student's selected course list.
4. Advisor approves student's courses.
5. Approved courses added to student's schedule.

Alternative 4: Failure at approving

4a-) At step 4, 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

4b-) At step 4, Selected courses' lecture hours collides with each other. Advisor will approve the course that repeating or from lower semester.

4c-) At step 4, 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.

4d-) At step 4, 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.

4e-) At step 4, 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.

4f-) At step 4, 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.

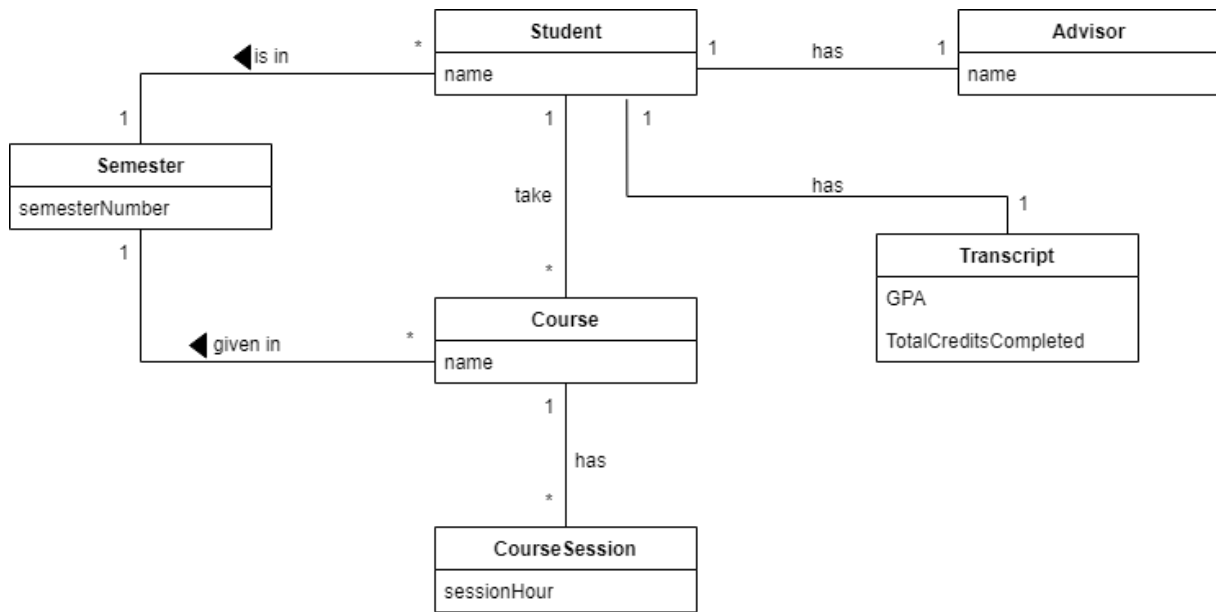
4g-) At step 4, Student course list contains faculty technical elective (FTE) course in FALL semester, and he/she is not graduating this semester. Advisor won't approve that course, informs the student, and drops that course from student's course list.

5-) Iteration Plan

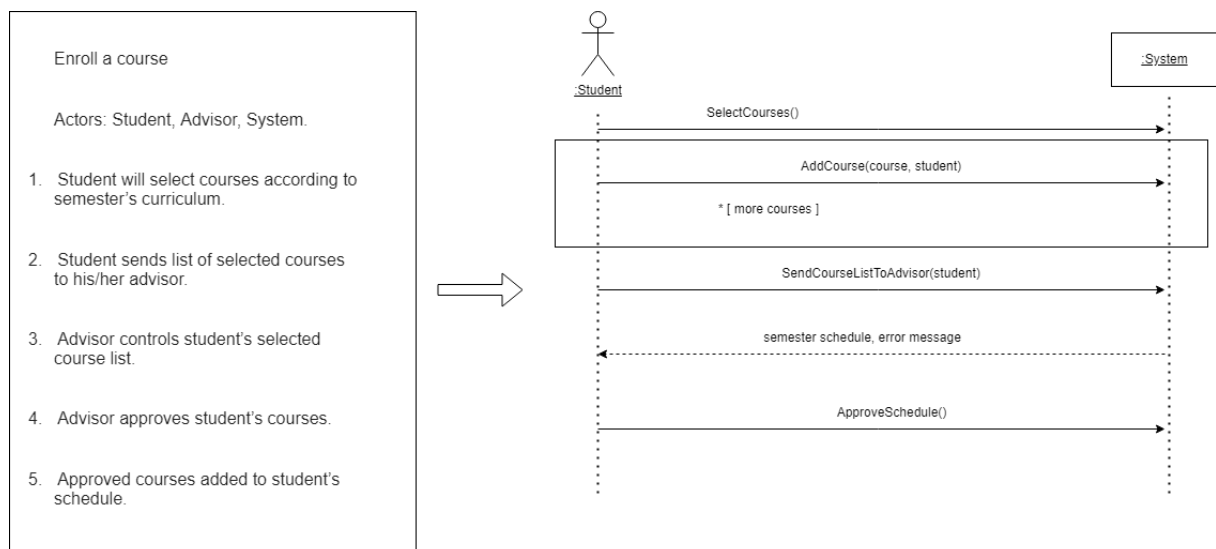
First Iteration: On first iteration, we will implement the core functionalities of the software such as.

- 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.

6-) Domain Model



7-) System Sequence Diagram



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.
- Java: Programming language.
- 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.
- 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.