**Student Registration System**

## Vision

In this project we are gonna build a registration system for the university. Mainly it serves to students and of course lecturers. With the help of this system we are aiming for automation and preventing this whole registration process from manual human errors.

## Description

In this project we will design a registration system for our school. Student Registration System (SRS) as a kind of management information system can not only record the information for student registration each term quickly and efficiently, but also do statistics on the students' basic information, registration information and give results of the analysis. The background of the subject is analyzed in the project firstly, and the technologies used in the development are introduced. Then the implementation of the system is given in detail. The upgrade and expansion of the system are prospected in the end of the project.

## Requirements

### Functional Requirements

1. Knowing student information

* Students will have semester information.
* Students will have advisor.
* Students will have passed classes.
* Students will have failed classes.
* Students will have total completed credits.
* Students will have transcripts.

2. Knowing lecturer information

* Lecturers will have classes which they will give.
* Lecturers will have students.
* Lecturers can be advisors.

3. Knowing course information

* Courses will have credits.
* Courses will have pre-requisite courses.
* Courses will have available semester
* Courses will have available hours.
* Courses will have required completed credits.

4. Course enrollment

* Checking if student can take the selected courses.
  + If student can take the course, advisor will approve, and course will be added to student’s schedule.
  + If student cannot take the course, inform the student with error messages.

### 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 separate JSON files.
4. Program should be implemented with JAVA.
5. Output will be mapped.

## Business Rules

1. Register for a new course, student must pass it prerequisite courses.
2. Register to Technical Elective courses a must be have more than blocker credits.
3. Register to a course, that course’s lecture hour can’t be collide with another courses lecture hour.
4. Register a course, that course’s quota must not be exceeded.

## Use Case

Enrolling to a course

* Student select their courses according to semester’s curriculum.
* Student sends of selected courses to their advisor.
* Advisor check student’s selected course list.
* Advisor approves student’s courses.
  + If selected course’s quota is full therefore student cannot take that course.
  + Selected courses’ lecture hours collides with each other. Advisor will approve the course that repeating or from lower semester.
  + Student haven’t passed pre-requisite of selected course.
  + Student haven’t completed enough credits for selected course.
  + Student course list contains technical elective course, although he/she already took 2 TEs in FALL semester.
  + Student course list contains technical elective course, although he/she already took 3 TEs in SPRING semester.
  + Student course list contains faculty technical elective course in FALL semester, and he/she is not graduating this semester.

**!!!** If any of the problems occur which listed above student can’t take the class and of course advisor won’t approve that course, informs the student, and drops that course from student’s course list.

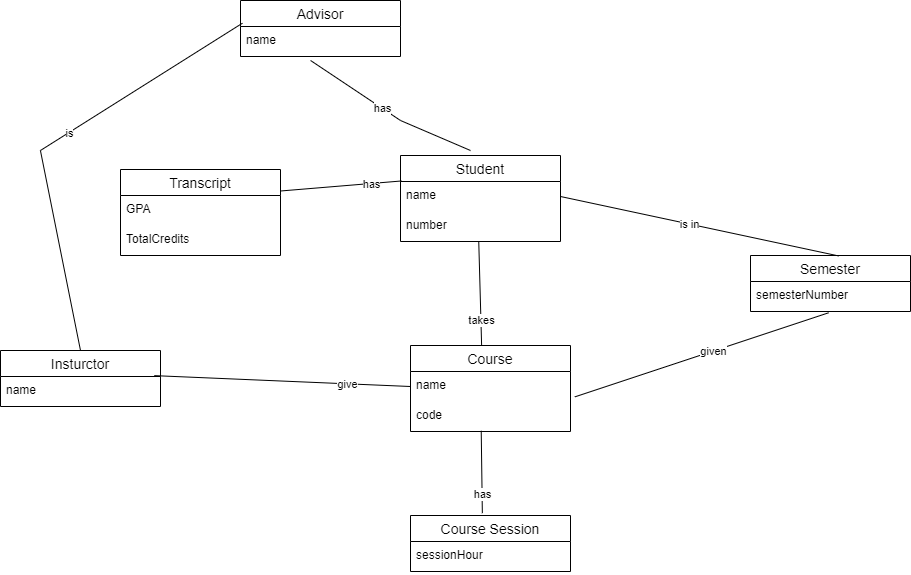
* Approved courses added to student’s time schedule.

## Iteration Plan

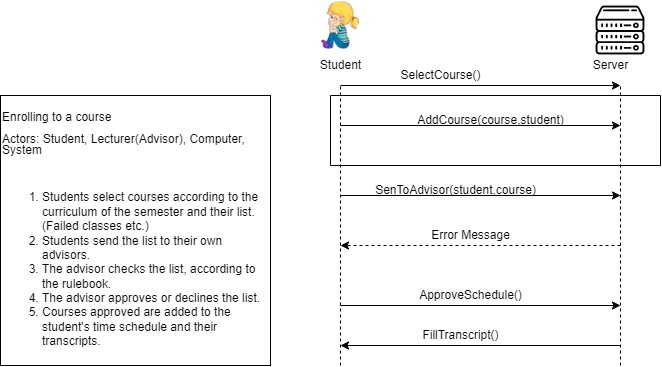
First Iteration: On first iteration, we will implement the core functionalities of the software such as.

1. Requirement analysis and creation of the diagrams about the current student registration system.
2. Implementing a student registration system.
3. Implementing credits and prerequisite system.
4. Implementing advisor and approval system.
5. Implementing a log tracing system.
6. Implementing a course registration limitations.

## Domain Model



## System Sequence Diagram



## Glossary

* Advisor: Actor who approves or disapproves student’s course selection list.
* Course: Lessons students must have passed to graduate.
* Credits: Weight impression of the lecture.
* Curriculum: Overall content of the course.
* FTE Courses: Faculty Technical Elective courses.
* Java: Backend Programming language.
* JSON File: Standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays.
* Log: Storing sequential data.
* NTE Course: Non-Technical Elective courses.
* Pre-requisite courses: Courses required to be past in order to get the connected course.
* Schedule: Student’s courses in a weekly plan.
* Session: Lecture hours available for same lesson.
* Semester: A half year term in a school.
* Student: A person formally engaged in learning, especially one enrolled in a school or college.
* TE Course: Technical Elective courses.
* Transcript: Lecture records of the student.
* UE Course: University Elective courses.