Vision

Our purpose for this project is, create a software that provide a service for course registration and tracking courses in a distance education environment. As a result of this project, we aim to develop a software system where students can see the courses they can take based on their transcriptions and be able to follow which courses they should take according to the Marmara University Computer Engineering Department curriculum.

Scope

While creating the software we will demonstrate an object oriented approach to create a student registration system. Also the software will be extensible. When customer want to add or update something on project, the software will extend without changing whole design.
In the system a specific semester will given as an input. The software will simulate a course registration for students from json files for that semester. There is some conditions will be taken into consideration to while register a course listed in functional requirements.

Functional Requirements

Program should read json files given as input.

Each student should have their own separate json file.

According to the specific semester given as input, the program should register all students ,whose json file is created, for that semester.

While students register courses program should consider “Curriculum for the students who registered after Fall 2020”. A student should register courses according to curriculum

While students register courses program should consider “Prerequisite tree for the students who registered in 2020 and later”. Adviser shouldn’t approve register if student didn’t take the prerequisite course(s).

If the hours of the courses to be taken overlap, the adviser should not approve the course(s).

For some specific courses, the adviser should not approve the course(s) if the student has not completed enough credits.

There should be a quota for technical elective, non-technical elective, faculty technical elective courses. If this quota is full, Student should not be able to register to the course(s).

The program should give two file as output. In the first file, the registration process of the student should be output as a json file. This json file should contain preregistration transcript, post-registration transcript and a summary of the problems encountered while registering for the courses.

In the second output, another json file should contain a statistical report of the total problems encountered by the students during course registration. In addition, the file should contain a list of students who had problems, and by looking at the json file of the students in this list, we should be able to see that the student could indeed actually enroll in the course because of the problem.

Nonfunctional Requirements

Usability Requirement:
In order for consumers to use the system, it must have a clear and straightforward design. When the user makes a possible error, it should provide the relevant information.

Performance Requirement:
The system must produce an output in a decent amount of time.

Robustness or Fault-Tolerance Requirements:
When the system is disconnected or frozen it should save all of the processes.

Reliability Requirement:
To assure that project code components are working, they will be tested during the implementation phase.

Stability Requirement:
The system should be secure as in sense that it should be able to work with a variety of inputs.

Supportability Requirement:
Any Java-based platform should be able to run the system.
The system should function without the use of a graphical user interface. It must use command line.

Glossary

Technical Elective course: any course in a technical field of student (field of computer engineering for this case).

Faculty Technic Elective course: courses that concern the student's department (Department of Engineering for this case).

Non-Technical elective course: courses in any subject that the student may choose.

Registration: The action or process of registering or of being registered to courses.

Credit: measure the number of applied hours that are recognized for successful completion of a particular course of study.

Prerequisite Tree: Diagram showing the courses that must be passed in order to take a course.

Curriculum: the subjects comprising a course of study in a school or college
Advisor: faculty employee who checks whether students meet the necessary conditions for enrollment in courses

Transcript: states your name, the institution you studied at, and a list of all courses taken, all grades received and degrees conferred. It should provide full details of the degree class and grades you received in the university

Json file: an open standard file format that uses human-readable text to store and transmit data objects

Stakeholders

Stakeholder

- Murat Can Ganiz

Developers

- Samet Enes Örsdemir
- Ömercan Göktaş
- Berkan Korkmaz
- Ahmet Can Başaran
- Ömer Yiğit Tatlısu
- Sarp Siraç Müftüler
- Muhammed Yusuf Şahin
- Tayyip Erkam Fidan
- Nurefşan Yücel

Use Cases:

Use Case: Complete Course;
Actors: Student, Course;

- A student wants to pass an active course.
- The student must take at least grade D to pass the course.
- If the student got enough grade, than the student can pass the course.
- After checking is completed, active course is going to be added to a passive course list of the student

1. Student gets a random credit from simulaton.
2. Grade will be converted to letter grade form.
3. If the grade is enough to pass the course, student adds the Course to his/her passive courses.
4. Course will be removed from Student's active courses.
5. Letter grade will be added to Student's transcript.
6. Cumulative GPA will be calculated on Student's transcript.

3.a. If the student got better grate than DD, will be moving to the next step.

3.b. If the student could not get enough grade to pass the course, it will not be able to possible for the student to pass the course.

Use Case: Registe a Course;

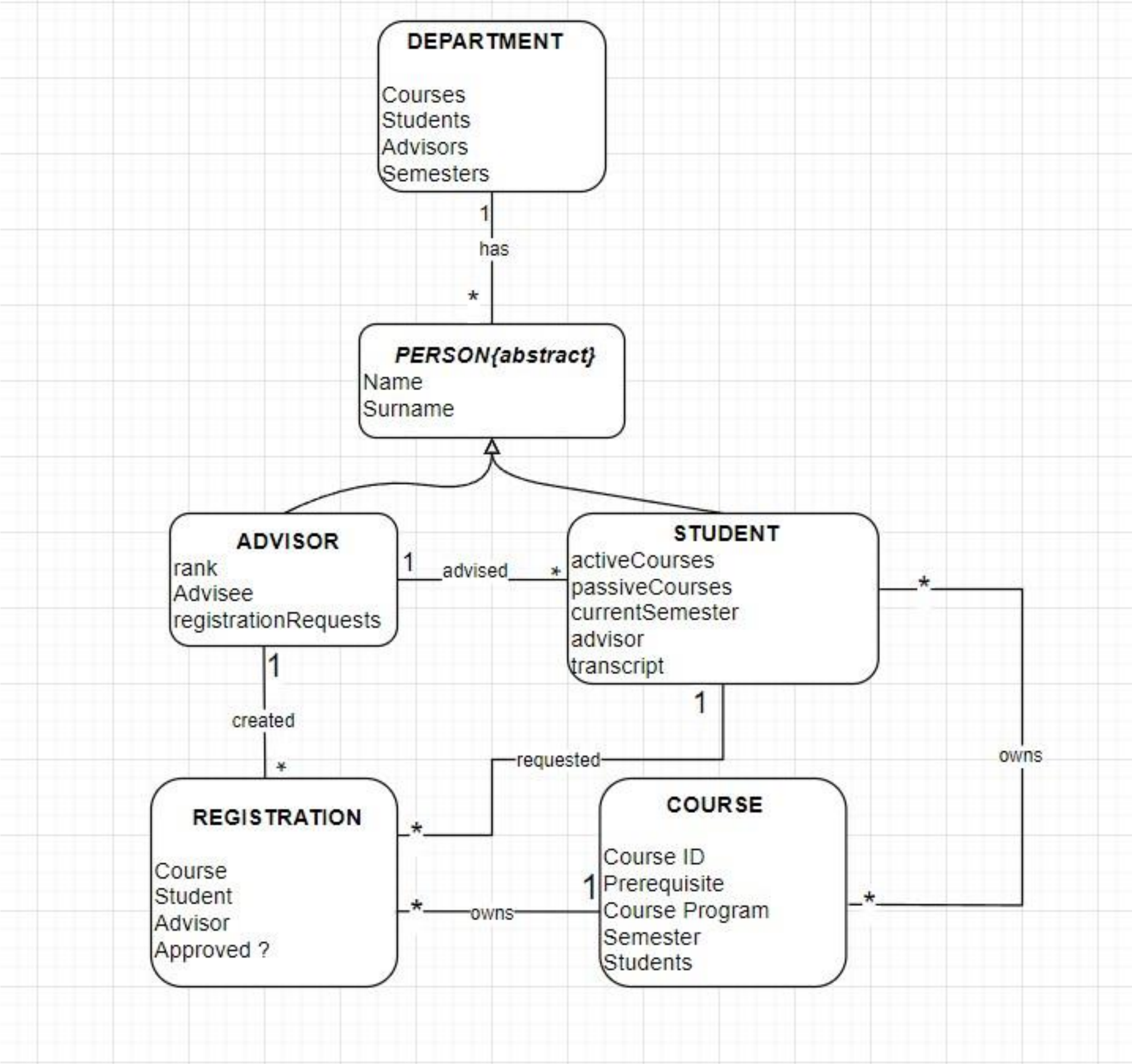
Actors: Student, Advisor ,Course Registeration;

- Student wants to register a Course.
- Student must have enough credit to register.
- Student must pass prerequisites of the course that student wants to register.
- Course quota must not be full.
- Student Week Schedule must be avaliaable to take the Course.

1. The student create a register.
2. Advisor create registration by using information of request.
3. Advisor wants to know course informations.
4. Course sends its informations to Advisor.
5. Advisor checks Course informations and Student informations by using Department rules.
6. If registration is avaliable for rules, Advisor confirms the registration.
7. Course will be added to Student’s active courses.
8. Student will be added to Course’s registered students.
9. Registration log will be added Advisor’s registration list.

6b. If registration is not avaliable, Advisor reject the registration.

Domain Model



System Sequence Diagram