Requirement Analysis Design

1. Description

The purpose of this project is to create a student course registration system. There will be a system where students can register their courses through the system. Students can easily enroll in the courses and the courses are supervised by the counselees.

2. Developers

- Hüseyin Hasılcı 150119698
- Murat Fidan 150119901
- Süleyman Burak Özcan 150119742
- Enes Burak Karaçuka 150119768
- Mustafa Furkan Yapıcıoğlu 150119724
- Turgay Tumay 150119750
- Hasan Hut 150119764
- Bedirhan Sarıhan 150119692
- Doğukan Kömürcü 150119770

3. Glossary

- **Student** : Someone who interacts with the system.
- **Course** : What students register through the system
- **Advisor** : Someone who check the courses and students
- **RegisterationManager**: The place where the advisor controls the courses and students information.
- **CourseManager**: The place where the courses are creating.
- **FileManager**: Writes transcripts and problems to files.
- **StudentManager**: The place where the students are creating randomly.
- **CourseSection**: Helps assign dates to courses.
- **Main** : Main class of design
- **Python** : A programming language

- **Functional Requirement:** A requirement that the system must be able to do.
- **Non-Functional Requirement**: A requirement that specifies how the system should do it.
- **BYS :** Marmara University Information Management System.
- **Transcript:** The place where students' course information is kept.
- **Prerequisite** : something that is necessary to an end or to the carrying out of a function.
- **JSON** : JSON is a text-based data format that is the lightweight alternative to XML widely used on the Web for data interchange.

4. Functional Requirements

- The system must create random students.
- The system must take the course data on BYS.
- The system must read transcript file.
- The system must writes the problems to the file.
- The system must write on transcript file.
- The system must create transcript file.
- The system must create random students' names.
- Advisors should control the course requirement and for courses.

5. Non - Functional Requirements

Usability

- Outputs and logs of the system must be a proper manner and to be easily understandable.

Flexibility

- When new courses are added, the system should be able to integrate them easily.
- The system should be able to easily integrate new students when they are added.

Performance

- The system should check the courses quickly and determine which courses students can take in a short time.

Reliability

- The project code will be tested to ensure it is functional.

Data Integrity

- All information of courses and students will be kept in json files.

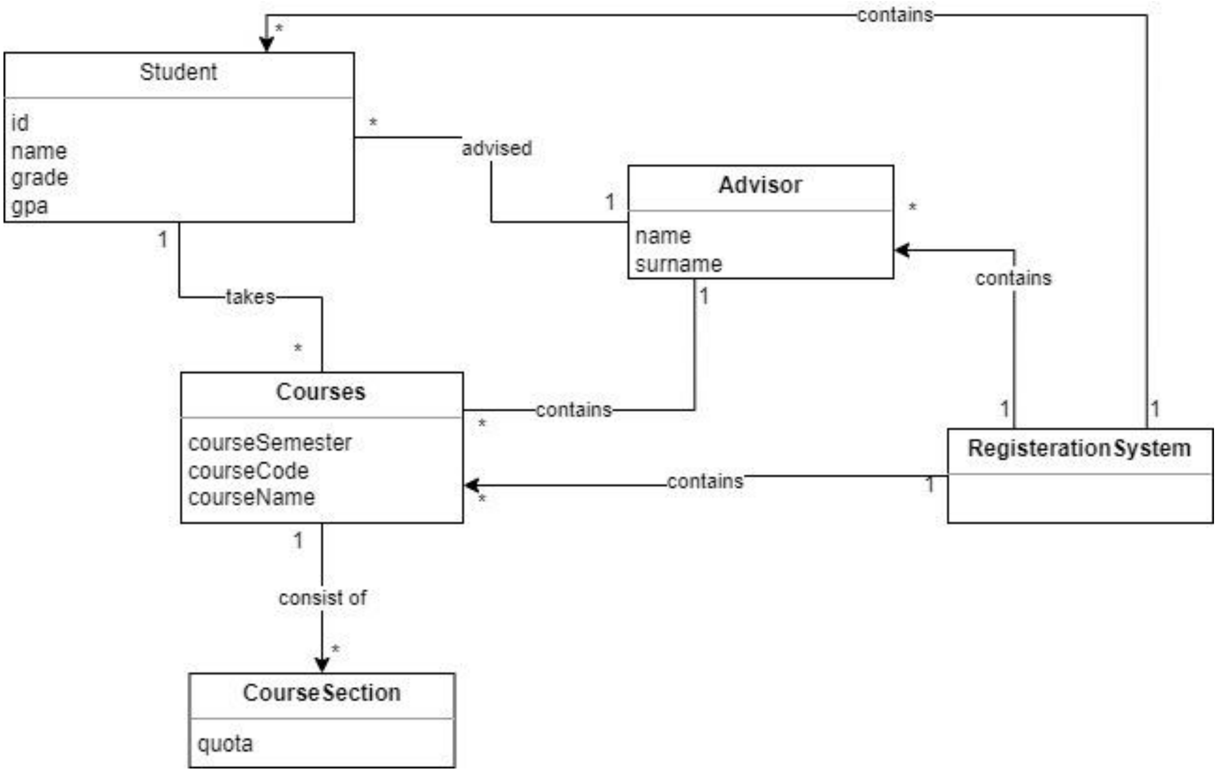
Security

- The information of the students kept in the system is personal and should not be shared. Therefore, there cannot be any outside access to the system.

Maintainability

- If there is any error that may occur in the system, the system determines these errors and writes them to the log file. All errors and outputs can be seen through log files.

6. Domain Model



7. Use Case: Course Registration

Actors: Student, System

1. The system presents the courses that the student can choose during course selection. These courses consist of the courses taken in the previous semester and the courses ofthe semester they are in.
2. The student chooses courses from the courses offered by the system.
3. The system saves the courses chosen by the student and sends them to the Advisor.
4. The Advisor approves the courses chosen by the student.
5. The system adds the approved courses to the student's transcript.
6. The course registration process ends.

Alternative: Course Selection Process

2a. Students can choose as many courses as they want, with a maximum of 10 courses offered by the system.

Alternative: System Check Errors

3a. If the quota of the course chosen by the student is full, the system does not allow the student to take this course. This course is removed from the courses that the student wants toregister for.

3b. If the course chosen by the student has any prerequisites, the system checks whether he/she has passed the prerequisite course successfully before giving the course. The course that does not meet the prerequisite will be removed from the courses that the student wants toregister for.

Alternative: Advisor Check Errors

4a. If the course is a Graduation project, if the student's total credits are less than 165, theAdvisor does not approve the student's course and removes this course from the student'scourses.

4b. If the course is a TE (Technical Elective) course, if the student has two TE courses approved, the advisor does not approve the student to take more TE courses during the semester and removes this course from the student's courses.

4c .If the course is an NTE(Non-Technical Elective) course, if two NTE courses are approved, the student is not approved to take more NTE courses during the semester. Also, the student cannot take the NTE course in the FALL semester. If these errors are received, thestudent will remove this course from their courses.