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
- **RegisterManager**: The place where the advisor controls the courses and students information.
- **CourseManager**: The place where the courses are creating.
- **StudentManager**: The place where the students are creating randomly.
- **Main** : Main class of design
- **Java** : 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 write on transcript file.
- The system must create transcript file.
- The system must create random students’ names.
- Advisors should control the Prerequisite 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.

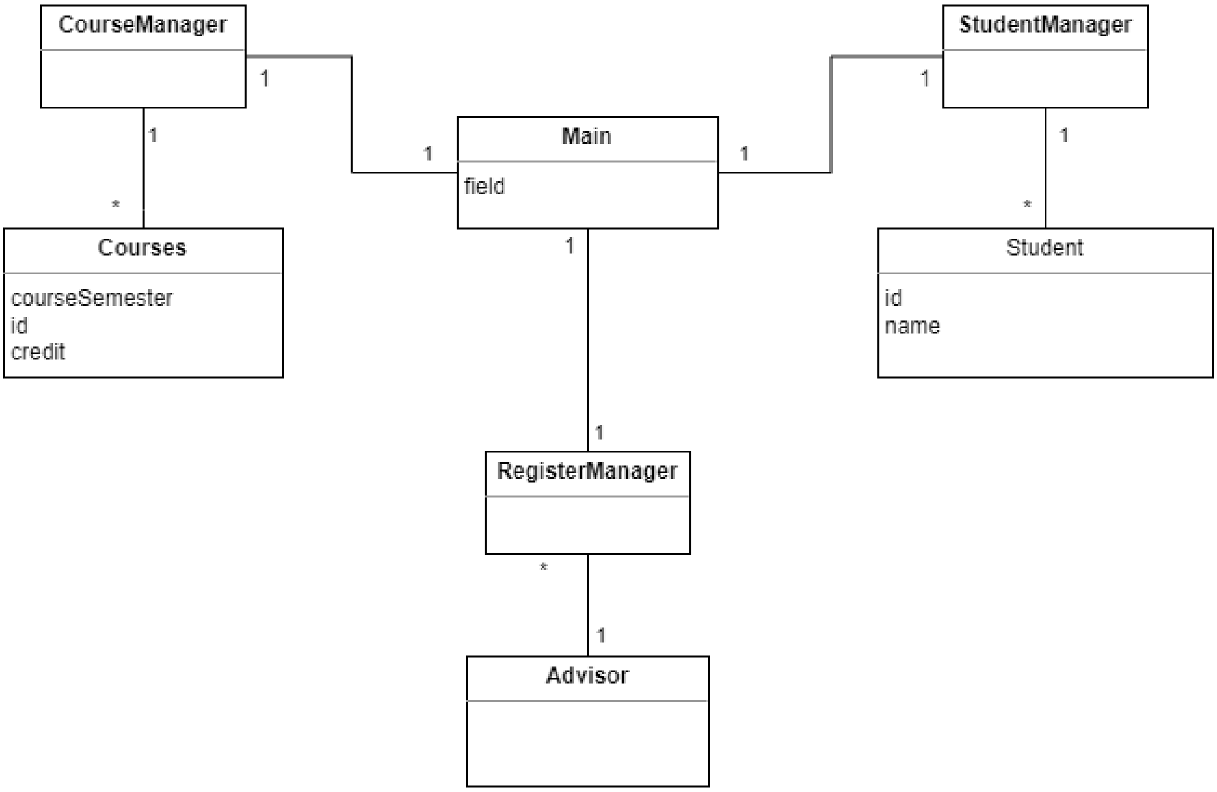
Security

- 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


CourseManager
StudentManager
1
1
Main
field
1
1
1
1
*
*
Courses
courseSemester
id
credit
Student
id
name
1
1
RegisterManager
*
1
Advisor