**Requirements Analysis Diagram**

# Vision:

The main goal of this project is simulate a course registration system. Students can send enroll request to a course, system check for quota and prerequisite courses, also advisor check for other issues (time conflict, the number of courses etc.)

Problem Description:

In today's age, university student information, course information, teacher information, transcripts are very complex. Complexities can be experienced as these structures take up large areas.

# Functional Requirements:

* The system must create random students.
* The system must be able to read from a json file and write through to a json file.
* The system must check for course prerequisites, course quotas…
* The system must allow advisor to check requirements and advisors can deny or allow to take course

# Non-Functional Requirements:

## Usability:

Outputs and system logs should have proper names and must be clearly

understandable

## Flexibility:

Whenever new courses, advisors, students are added, system should integrate them without any bugs.

## 

## Performance:

The system must do its tasks in a proper amount of time. No delays.

## Reliability:

The source code should be tested and does not include any bugs

## Data Integrity:

All data store in json files

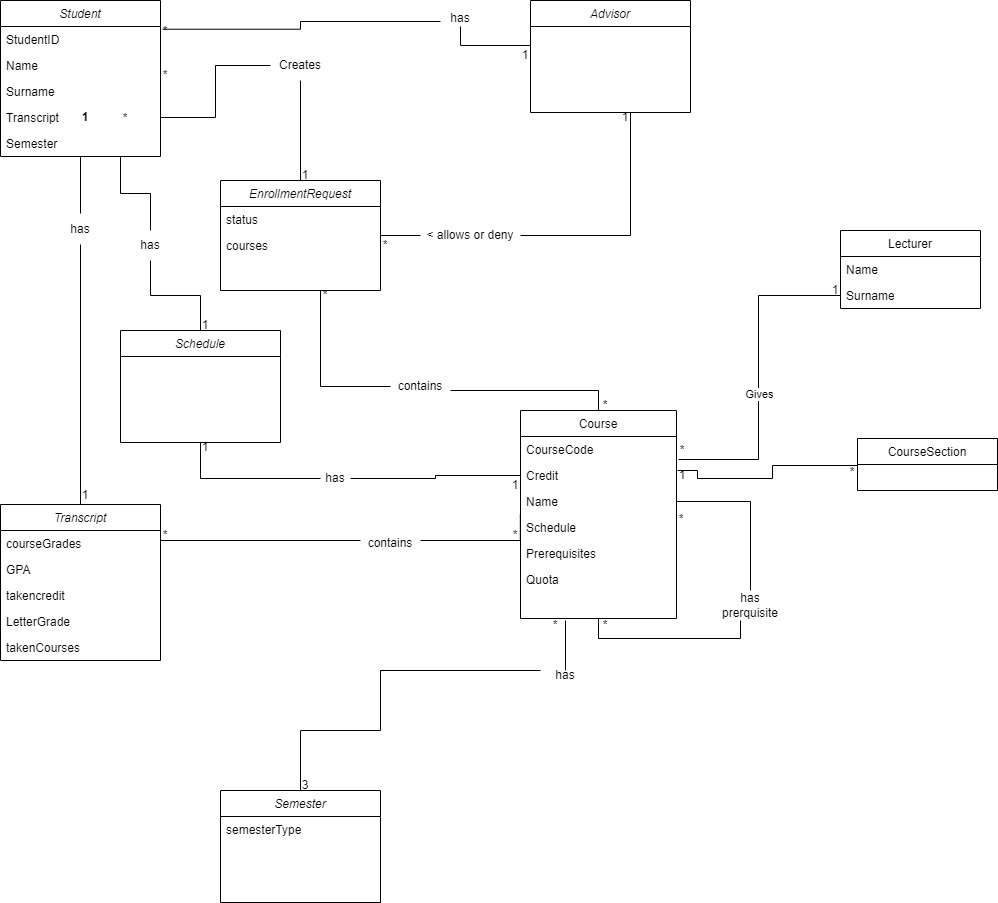
## Maintainability:

The source code must be easy to understand so if any bugs occur devoleper can detect with no effort. Also all possible errors should be logged in a file

# Use Cases:

Actors: Customer

* System will be launched in command line(terminal).
* Randomly created students will select randomly created courses.
* System checks for course quota or other requirements
* All students have a randomly generated advisor
* Randomly created advisors checks every course enrollment requests and generate an output(allow/deny)
* All operations are logged and count of every operation is produced as an output

Domain Model:

Glossary:

* Course: The course is what student registers through BYS, Information Management System.
* Student: Student is a user who is interacting with and within the system.
* Advisor: Advisor is a person who is assigned to students in order to check whether they can take specific courses, regarding on quota, credits, prerequisites etc. Furthermore he/she helps students with other problems regarding university as well.
* RegistrationManager: This is the manager which helps the advisor to control the courses and students’ information.
* CourseManager: This is the manager where the courses are being managed.
* FileManager: This manager writes transcripts and problems to files, as its names suggest, it manages the files.
* StudentManager: In this manager students are being created randomly.
* CourseSection: This helps to assign dates to courses.
* Transcript: The place where students' course information is kept.
* Java : A programming language( with which the project is done).
* Main: This represents the main class of design.
* Functional Requirement: A requirement that the system must be able to fulfill and to do.
* Non-Functional Requirement: A requirement that specifies how the system should do it giving it required instructions.
* BYS : Marmara University Information Management System.
* Prerequisite : something that is necessary, that needs to be accomplished in order to continue, e.g. you need to pass programming 1 before you can take programming 2
* JSON : JSON is a text-based data format that is the lightweight alternative to XML widely used on the Web for data interchange.