**Requirements Analysis Diagram**

# Vision:

The main goal of this project is to simulate a course registration system. Students can send enrollment request to a course, system checks for quota and whether there are any prerequisite courses. On the other hand, advisor checks for other issues that may occur such as (time conflict (clash), the number of courses etc.)

Problem Description:

In today's age, university student information, course information, teacher information and transcripts are very complex. Complexities can be experienced as these structures take up large areas. A course registration systems should be organized very well and pay attention to the smaller details in order to make it an easier and better experience for the students.

# 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 , etc.
* The system must allow advisor to check requirements and advisors can deny or allow a student to take certain course

# Non-Functional Requirements:

## Usability:

Outputs and system logs should have proper names and must be clearly understandable, easy for users’ use.

## 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 should not include any bugs.

## Data Integrity:

All data should be stored in json files

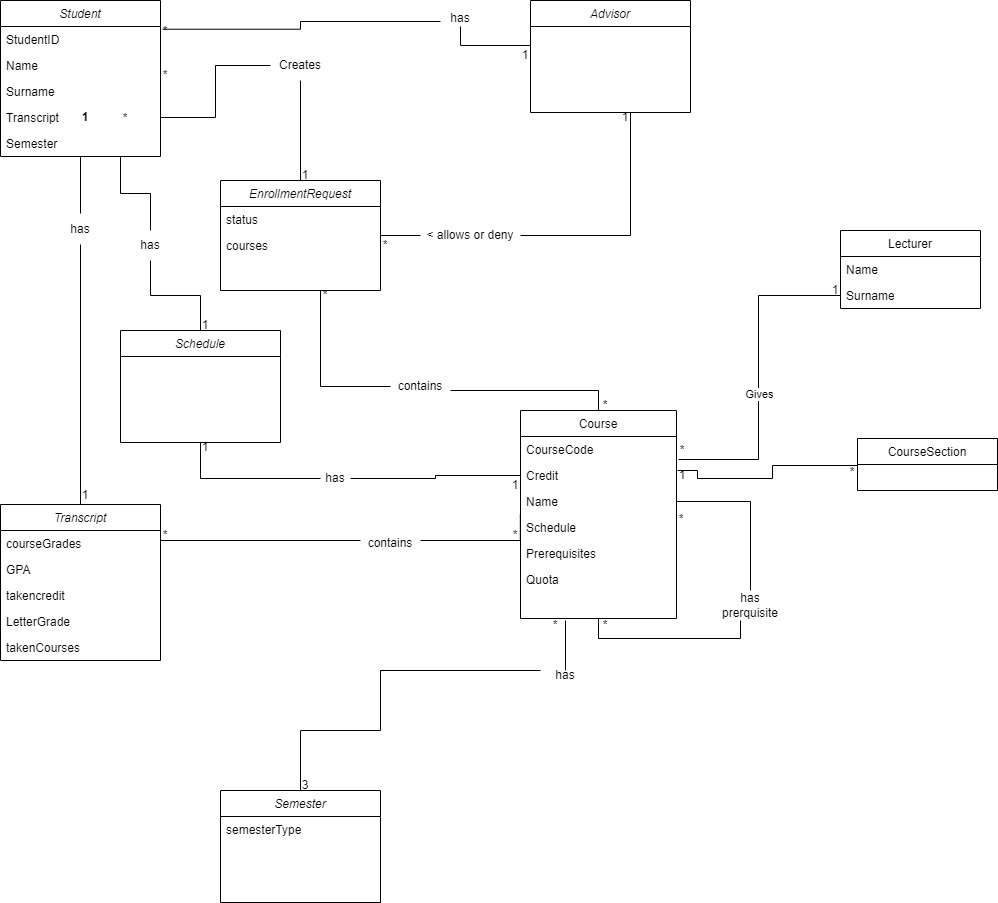
## Maintainability:

The source code must be easy to understand so if any bugs occur developer 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 randomly generated advisor
* Randomly created advisors check 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: