CSE3063 PROJECT 1 GROUP 3 ITERATION 3 REQUIREMENT ANALYSIS DOCUMENT 1 Introduction

1.1 Problem Statement

To fix the problems with course management, we propose to develop an extensive online course registration and management system simulation that complies with Marmara University rules.

1.2 Product Scope

This project aims to overcome the challenges of students and advisors in selecting and managing courses by following the conditions set out in the Marmara University regulation. This system will include functions such as course selection, management of courses received, sending selected courses to advisor approval, managing enrolled students and creating student transcripts.

1.3 Vision

This project aims to accomplish a successful simulation of a university's course registration system by using the benefits object oriented software design. The project is inherently targeted at the students of Computer Science and Engineering department of Marmara University.

2 Overall Description

2.1 Product Functions

This system’s purpose is to allow students to register to courses under some restrictions which is controlled by the system itself. After the student selects a course, the system controls the restrictions for that student to register to that particular course and decides to approve or reject the registeration process. A transcript is created for each student that can be accessed by them.

2.2 User Constraints and Characteristics

All students and advisors will be able to use this software but, students won’t have access permission to all parts of it.

Students: They can enroll to courses in new semesters and access their transcripts.

Advisors: They can control their students' course enrolling period. They can approve or disapprove students’ schedule for the semester.

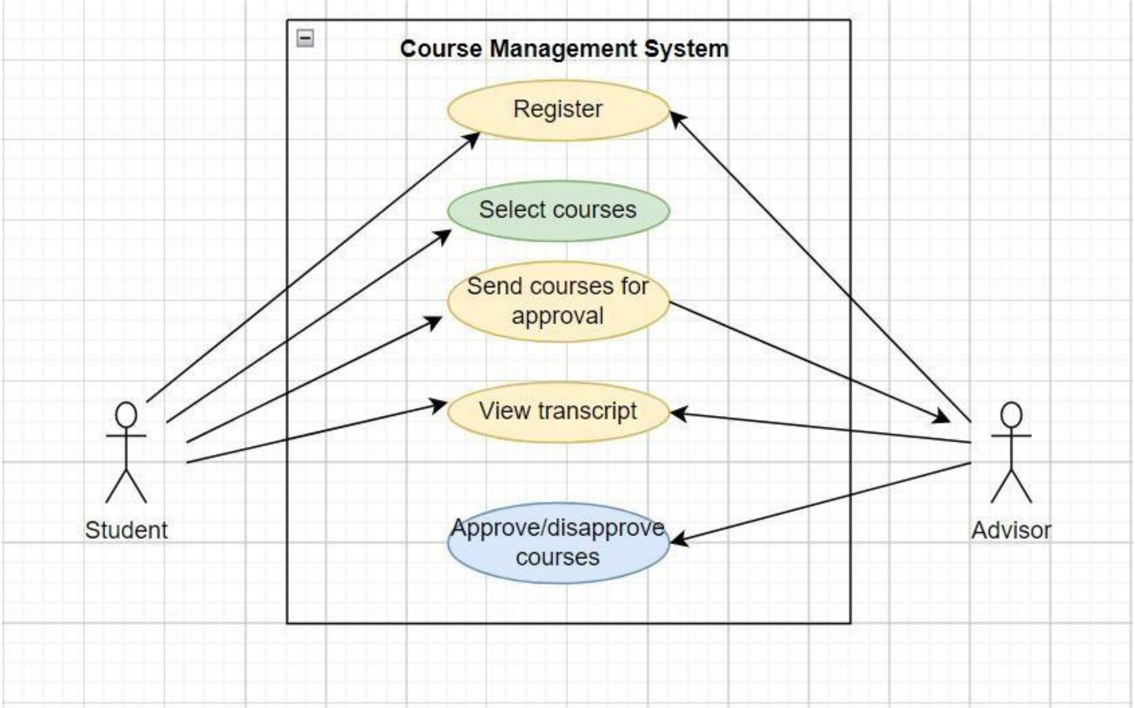


Figure 1 : Use case diagram

3 Requirements

3.1 Functional Requirements

FR1 - The system shall provide the transcript

FR2 - The system shall enable students to subscribe/unsubscribe to courses

FR3 - The system shall provide a notification in case there is a collision

FR4 - The system shall provide a notification in case a requested prerequisite is not met

FR5 - The system shall provide a notification in case of lack of sufficient capacity

FR6 - The system shall enable advisors to approve/disapprove the courses

FR7 - The system shall enable users to view timetable for classes

3.2 Nonfunctional Requirements

3.2.1.Availability:

Course management system will be a software product produced in Python programming language. It runs on the ide console and can be used on many platforms such as: Visual Studio Code.The operating environments for development and product use is as follows:

- Development environment of the product: Python

- Database: consists of JSON files that contain data of students and courses.

- Operating Systems: Windows and Linux

The number 1 requirement for users is a computer with a Python program, they can run the software with its help.A mouse is optional but a keyboard is a mandatory need.

The user needs a computer which can edit a json file, a simple text editor which comes alongside with operating systems like Windows, Mac or Linux will be enough.

3.2.2 Performance:

The system should give an output at a reasonable time.

Since all code is in front of the user, when the user writes extra code or changes anything and runs the program every change will be visible at the moment of execution.

3.2.3 Reliability:

The code will be customizable according to the user's request. Every part can be changed and improved using ide or a code editor.This feature gives the user full authority, so this situation can lead to some dangers.The user may unintentionally cause very large errors and may not be able to solve the problem, so they should be very careful.

Just like the source code, JSON files which contain properties of students and courses will be available to change. Users can change anything about any student's information, they can be created, changed, or deleted. This feature is also vulnerable to any bad intentions. Users can see and observe everything but the system could get corrupted at any moment.

Users may be only interested in executing the code and don’t care about changing it, the software will run smoothly so its priority is low, but for security it's very high.

3.2.4 Security:

Our information and source code will probably hold in a file, and all information will be there. Our program gives the user all possible permission on everything. The files could be deleted by the user and there won't be any warnings since it's like any other file on the computer. All the information about courses and students may be lost with simple actions. All the information needs to be backed up somewhere, we suggest multiple HDD like devices since they are not connected to any network and portable and easy to protect.

Glossary

JSON Configuration file: It is a modifiable input file that contains information of department, curricilum, lecturers & teaching assistants, students and courses.

Registration Message: Registration message is a simple message that states whether the student registered successfully or not, if the registration is unsuccessful it states the reason.

JSON Output file: JSON Output file: There is an output file for each student that uses the registration system. The name of the file is the student number and it contains student transcript that contains all the necessary information.

Advisor: Advisors are selected among department lecturers. Each student is assigned a random advisor who can approve or dismiss student’s course registration requests.

Department Rule: Department rules determine if a student can register to the chosen course.

Prerequisites: A course can have many prerequisites and be prerequisite to many courses, as well as having no prerequisites or not being prerequisite to any courses.

Course Types: A course can be one of the following: mandatory course, elective course or lab course. Lab courses and elective courses have a capacity. Lab courses have sections ans are teached by teaching assistants

USE CASE

Scope: Student Information System

Level: User Goal

Primary Actor: Lecturer

Stakeholders and Interests:

- **Department:** Department is collection of students, lecturers, and teaching assistants. Each department can have their custom configuration file. Configuration file contains information about department, lab section capacity limit, elective course limit, given courses in that department and lecturers of department.
- **Curriculum:** Curriculum determines which courses can be taken on which semester according to faculty syllabus.
- **Courses:** Courses may in type of mandatory or elective. We have several elective course types such as non-technical, faculty technical, university etc. If a course contains lab hours, it can have more than one section. Each lab section instructed by a teaching assistant. A course can have pre-requisite courses, or it can be prerequisite for another courses. If a course have a pre-requisite course, student must have succesfully pased that pre-requisite course to take that course.
- **Lecturer:** A lecturer may teach more than one course. Also they may supervise students.
- **Teaching Assistant:** Teaching assistants are instructors of lab sections. Each lab section can have only one teaching assistant, but one course can have many lab sections.
- **Student:** Student is the one who takes courses according to curriculum. They can not take course from previous semesters or coming semesters. They can select courses which has collision, but these selections not approved by their advisors.
- **Transcript:** Transcript is the document which is showing that the student's taken courses and grades from these courses and GPA of his/her.

Preconditions: Lecturer is identified and authenticated.

1. Lecturer runs the code.
2. The program creates random students.
3. For each course taken by a student, it assigns random point between 0-100. By using that grade it calculates letter grade and grade in [0, 4] format. For elective courses, it will select a random course from elective course pool. Also, for the lab sections a random section will be selected from lab sections.
4. After course registration process, system will generate an output file for each student. That file will contain student information and 2 transcripts in JSON format. First one will contain courses from previous semesters and the other one will contain all courses inluding the ones recently taken.
5. After student output files generated, system will generate an output file for department. It will contain information about which course could not be taken by how many students.

Technology and Data Variations List:

- This program will create new random students every time it runs.

Alternative Flows:

3a

If capacity of all elective courses is full student will not be able to register any elective course.

3b

If lab section capacity is full for each section student will not be able to take a lab section.

3c

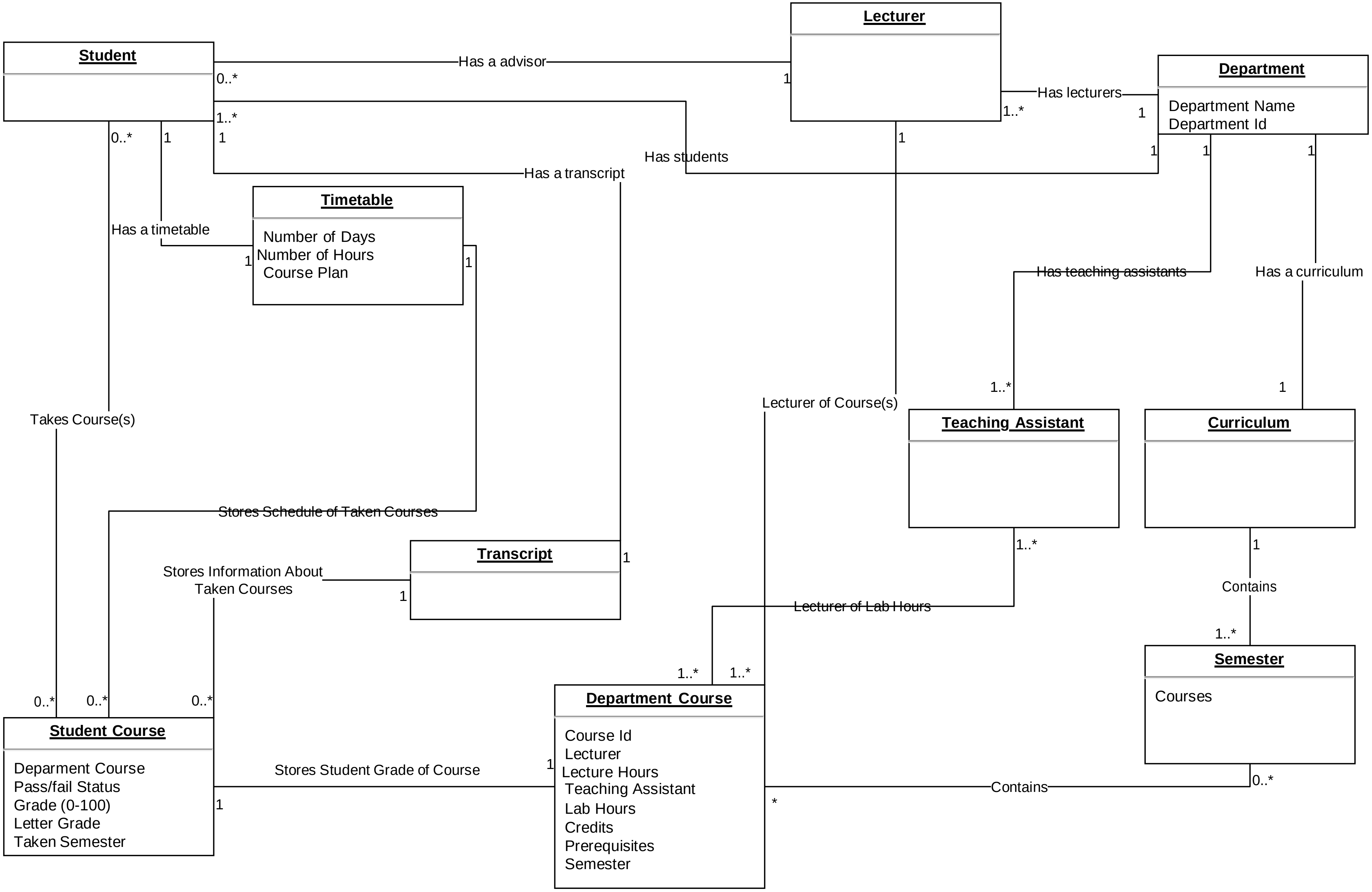
If student has any collision (taken two or more courses with a same schedule, all lecture hours does not need to be the same) in his timetable, advisor won’t approve registration of the second course.

3d

If course grade less than 1.5/4, student will fail from the course. And he/she can not take other courses if the failed course is a prerequisite for them.

References

https://bys.marmara.edu.tr
https://cse.eng.marmara.edu.tr/dosya/eng/cse/documents/general/PrerequisiteTree_2020.pdf
https://www.marmara.edu.tr/dosya/www/mevzuat/2021/mu_yonerge_basari_degerlendirme_2 020_v2 04.02.2021.pdf?_t=1612473513


Lecturer
Student
Department
Department Name
Department Id
Has a advisor
0..*
1
Has lecturers
1..*
1
1..*
1
0..*
1
1
1
1
1
Has students
Has a transcript
Timetable
Number of Days
Number of Hours
Course Plan
Has a timetable
1
1
Has teaching assistants
Has a curriculum
1..*
1
Lecturer of Course(s)
Takes Course(s)
Teaching Assistant
Curriculum
Stores Schedule of Taken Courses
Transcript
1
1..*
1
Stores Information About
Taken Courses
1
Contains
Lecturer of Lab Hours
1..*
Semester
Courses
1..*
1..*
0..*
0..*
0..*
Student Course
Deparment Course
Pass/fail Status
Grade (0-100)
Letter Grade
Taken Semester
Department Course
Course Id
Lecturer
Lecture Hours
Teaching Assistant
Lab Hours
Credits
Prerequisites
Semester
Stores Student Grade of Course
1
1
*
Contains
0..*