STUDENT REGISTRATION OF COURSE SYSTEM REQUIREMENTS ANALYSIS DOCUMENT

Vision

Our vision for Student Registeration of Couse System is to create simulation of student course registeration which used for our department. It is modelled as our department rules and regulations for taking courses. Our system will be developed as a simulation that takes arguments of user command line and input files. Simulation runs until all student's transcripts have been produced.

Scope

Student Registeration of Couse System will be simulation fort his Project. This system will be geting a dataset as a Json file. There are prerequisites of the courses and other businnes logic will also modelled after our departments rules and regulations for taking courses. After processing all instances, system will create an output Json file.

Functional Requirements

User can wiew student's transcript. In this transcript there are taken courses, grades, YANO, GANO, given credit, completed credit, offered courses and errors of registerations.

User can recreate students.

Advisor checks the courses and approves schedule.

Nonfunctional Requirements

Software Language: We used phyton programming languge to handle OO Programming efficiently.

Capacity : 70 students for each year. 280 students at total.

Availability : User can wiev the system when a user would like to use it.

Usability : Accesing to student information files and offered courses. The student can complete the registeration process in a short time.

Iteration Plan

As a result of this project, we will have a student registeration system that can work both step by step and as a simulation. We will have equal number of students in each of the 8 semesters. While registering for these students ; There will be restrictions such as completing the prerequisite courses, not exceeding the course qouta. Moreover, students will need to achieve a certain number of credits in order to take final year courses. After the registeration process is completed, user will be informed about the actions taken.

Glossary

Advisor is the lecturer who helps the students with the course registeration.

Course is a set of classes that focus on a certain topic given throughot the semester.

Phyton is an interpreted high-level general-purpose programming language.

Json File is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects.

Offered Courses is the list of courses that the student can enroll in the current semester.

Prerequisite Tree these are the rules that contain the necessary conditions for a student to enrolling in a course.

Quata is the maximum number of students who can enroll in a course.

Registeration system is a system that helps students register to courses.

Semester is a half a year term in a school.

Student is the person that enrolls in a course.

Transcript is the document which shows the student enrolled, completed or failed courses.

Stakeholders

Customer :   Murat Can Ganiz

Developers :   Serkan Koç
               İbrahim Hakkı Candan
               Atila İlhan Yatağan
               Kardelen Kubat
               Rümeysa Ulusoy
               Ramazan Karkin
               Mervan Aburşu
               Erdi Türkay
               Ulaş Deniz Işık

Use Case: Register to Course
Actor : Student, Advisor

1-)Student logs into the system.
2-) The system displays a list of available courses to the student.
3-) The student looks over the courses that have been listed.
4-)A student can only take a maximum of ten courses from the course pool.
5-) The student chooses from a list of courses.
6-) The student submits the courses he or she has chosen for approval to the advisor.
7-) The system displays to the student who selected courses the result of the advisor's approval.

Alternative: Selection Failure
4a. The system issues an error for selecting overlapping courses at step 4. The student can re-select courses from the list using the system.
4b. For selected courses, the system returns an error due to a quota problem.

Alternative: Batch registration
5a. Student selects the batch option and selects all the courses in all of the semesters.

Alternative: Prerequisite Failure
6a. Student selects the batch option and selects all the courses in all of the semesters.
6b. System gives an error to the student who selects the courses which have prerequisite condition . System allows student to re-select the courses in the list.
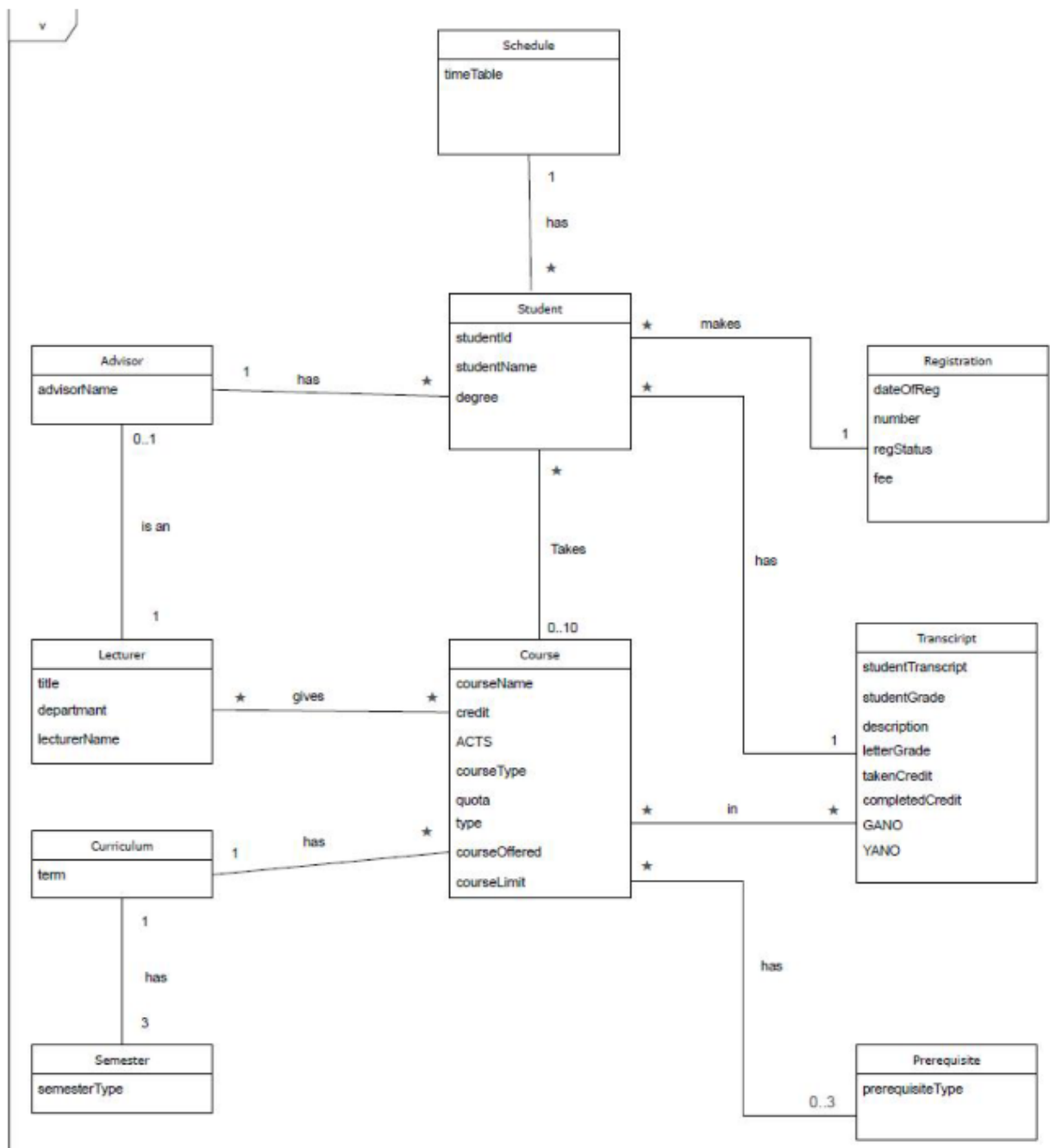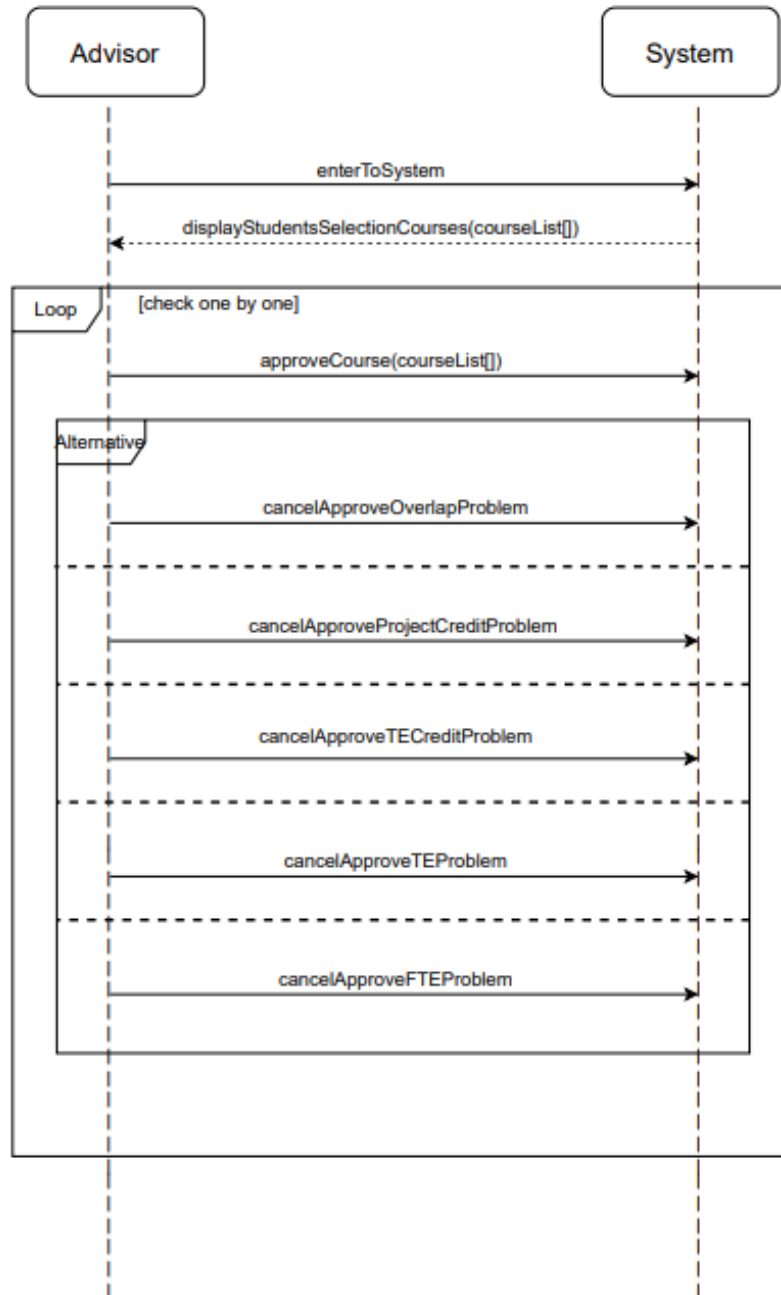
Use Case: Advisor's Approval
Actor: Advisor, System

1-) Advisor enters to the system.
2-) System displays the student's selection course request to the advisor.
3-) Advisor browses the courses selected by the student .
4-) Advisor starts to the process of approving course selection request .
5-) Advisor approves the courses one by one .

Alternative: Disapproval the selection

5a. At step 5 , the advisor do not approve because of overlapping courses.
5b. The advisor do not approve graduation Project because student completed credits 165 .
5c. The student is unable to enroll in TE classes because he or she has finished credits 155.
5d. The advisor do not approve the Technical Elective Course CSE4062 because student already took 2 Technical Elective and in fall semester only 2 Technical Elective can be taken .
5e. The advisor do not approve Faculty Technical Elective course CSE4062 because students can not take Faculty Technical Elective course in fall semester unless they are graduating this semester.

Schedule
- timeTable

1
has
*

Student
- studentId
- studentName
- degree

Advisor
- advisorName

1 — has — *

0..1

is an

1

Registration
- dateOfReg
- number
- regStatus
- fee

* — makes

*

1

has

Lecturer
- title
- departmant
- lecturerName

Takes

0..10

Course
- courseName
- credit
- ACTS
- courseType
- quota
- type
- courseOffered
- courseLimit

* — gives — *

Transciript
- studentTranscript
- studentGrade
- description
- letterGrade
- takenCredit
- completedCredit
- GANO
- YANO

1

* — in — *

Curriculum
- term

1 — has — *

Semester
- semesterType

1
has
3

has

Prerequisite
- prerequisiteType

0..3

Advisor's Approval

Advisor → System: enterToSystem

System --> Advisor: displayStudentsSelectionCourses(courseList[])

Loop [check one by one]

Advisor → System: approveCourse(courseList[])

Alternative

Advisor → System: cancelApproveOverlapProblem

Advisor → System: cancelApproveProjectCreditProblem

Advisor → System: cancelApproveTECreditProblem

Advisor → System: cancelApproveTEProblem

Advisor → System: cancelApproveFTEProblem

Register to Course