SOFTWARE REQUIREMENTS ANALYSIS

FOR

<Course Registration Simulation>

prepared by <Group 21>

<13/11/2021>

1. Overall Description

This project aims to simulate a random course registration process based on particular department policies and rules. At each execution of the program there will be generated randomly a sample of students for each class at a random semester and show the possible course schedules that are specific to a student's information taken from the transcript that is generated from random letter grades.All department regulations are applied while both detecting the available courses and generating student transcripts.

2. Requirement Specifications

Requirements will be ranked as

High: will be represented as H,

Medium: will be represented as M and

Low: will be represented as L

by

Risk, Coverage and Criticality.

Functional Requirements

1. Courses and business logic related to these courses in the Computer Engineering Department of Marmara University will be used. - LRisk | HCoverage | HCriticality -
2. There will be 8 semester and for each semester at least 35 students will exist. - LRisk | HCoverage | HCriticality -
3. Students will be generated randomly. – HRisk | HCoverage | LCriticality -
4. Each student will have a transcript and content of these transcripts will be generated randomly. – HRisk | HCoverage | LCriticality -
5. There will be five types of courses. Compulsory courses, TE, NTE, FTE, and UE courses. – LRisk | MCoverage | HCriticality
6. All courses will have a student quota, timeline and credit that represent the course. Some courses will have a lab or problem section and each may have more than one different sessions. These sessions will also have certain quotas. Some courses have prerequisite courses. In addition, some special courses require a certain amount of credit to be reached in order to be taken – MRisk | MCoverage | HCriticality -A prerequisite tree and credit limits of some courses will be provided according to the department policy. – LRisk | HCoverage | HCriticality -
7. A student can choose courses only from the current semester and the previous semesters. – LRisk | LCoverage | MCriticality –

8. No student will fail more than two courses in a semester. – LRisk | LCoverage | MCriticality –
9. If a student has taken two or fewer courses in a semester, they will pass both . – LRisk | LCoverage | MCriticality –
10. Passing the course letter grades will be ranked from AA to DD and failed letter grade will be FF. – LRisk | LCoverage | MCriticality –
11. A student must have a GPA of at least 1.8 in order to take courses from the current semester. – LRisk | LCoverage | MCriticality –
12. If a student's GPA is less than 1.8 only the failed courses and the passed courses with letter grades DC and DD can be choosen by the student. – LRisk | LCoverage | MCriticality –
13. A students will be able to choose the courses whose quota is not filled and for which the prerequisites are met. –MRisk | MCoverage | HCriticality -
14. Elective courses will be chosen randomly for each student. – MRisk | LCoverage | MCriticality -
15. Selected courses will be sent for advisor approval. – LRisk | LCoverage | MCriticality -
16. The advisor checks the conflicts in the courses and whether the credit requirements of the courses that have credit limit are met or not. – HRisk | MCoverage | HCriticality -
17. The courses approved by the advisor are taken and the not approved ones are not taken. – LRisk | MCoverage | HCriticality -
18. For each student, a record of the courses she has taken and the courses she has not taken, together with the reasons, is kept. – HRisk | LCoverage | MCriticality -

For each semester, it is recorded how many students took which course and how many students did not take which course with its reasons. – HRisk | LCoverage | MCriticality -

Non-Functional Requirements

1. Java will be used as programming language.

2.Json files will be used as database.

3.The software will have a command line interface.

4. The software should have a reusable dynamic design for adding new features.

5. The software must be executable on desktop pc platform.

6. Users should be able to easily understand the purpose of the software and what it does.

7. Response time of the software should be as fast as possible.

3. Glossary

Business Logic: Rules and regulations of the department and prerequisites.

GPA: Grade point average. It is ranked from 0 to 4 according to course success.

Transcript: A document that keeps the course grade and information of the student.

Semester: A time period for a single term of a education year.

Letter grade: Equivalent letter to the course mark.

JSON: Short for Java Script Object Notation which is a data format.

Quota: A limitation for the number of students that can register for a course

Timeline: It is the semester when the course is started to be taken.

Credit: It is the numerical value that each course has, showing the required student workload and that a certain value must be reached in order to graduate.

Lab Section: These are the sections where the content of the relevant course is practiced in the laboratory environment.

Problem Section: These are the sections where the content of the relevant course is practiced with the problems.

Compulsory Couses: Courses that must be taken and completed in order to graduate.

TE Courses: It refers to “Techinal Elective Courses”. They are a pool of courses that are technically close to department courses.

NTE Couses: It refers to “Non-Techinal Elective Courses”. They are a pool of non-techinal courses.

FTE Courses: It refers to “Faculty Techinal Elective Courses”. They are a pool of courses that are technically close to faculty courses.

UE Courses: It refers to “University Elective Courses”. They are mixed pool of university courses.

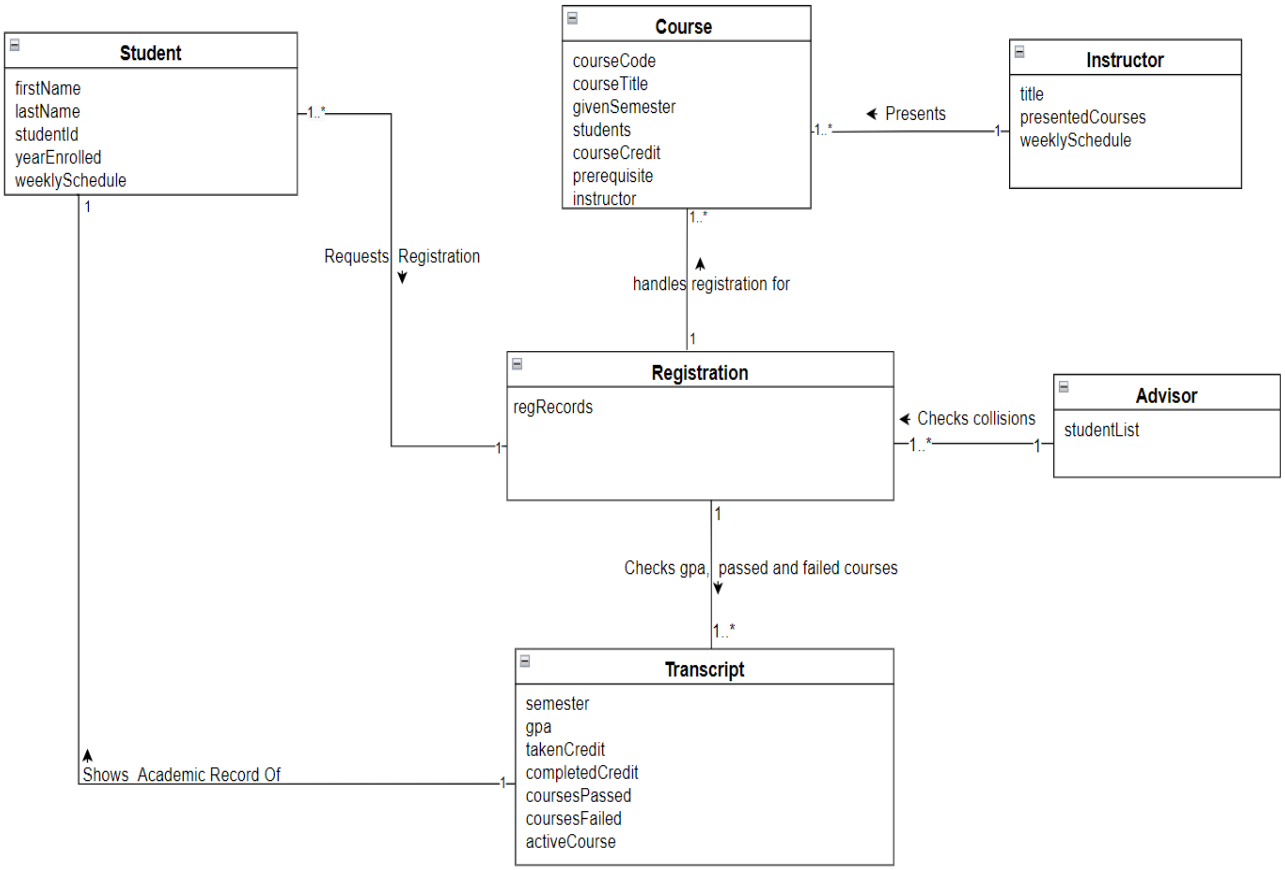
Section: Indicator of whether the object represents a main section of the class or the lab sections.**Schedule:** Implementation of a time table displaying the hours occupied by a specific class or a student.

Collision: The situation where the student tries to take more than one course that coincides with the same time schedule.

Advisor: A lecturer who checks the course schedule conflicts (collisions) and gives the course registration approvals accordingly, and also undertakes the task of consulting students.

Prerequisite: A type of course which a student has to successfully pass in order to take another, chained courses.

4. Domain Model



Use Case: Register For Courses

Actors: Student, Advisor, System

1. System checks transcript of the student and determines courses to be listed in order student to take.

2.Student selects courses available.

3.Student sends list of taken courses to Advisor.

4.Advisor confirms courses can be taken or not.

5.System creates outputs if there was an error while taking courses.

6.If everything is normal, student enrolls in classes taken.

Alternative: System failure

1a. Student may have gpa less than 1.8. If that is the case, he/she can only take

past failed courses and passed courses with score DD or DC.

3a. At step 3 enrolling may fail due to quota or prerequisite issues.

Allow student to make adjustments to the list of courses taken.

Alternative: Confirmation failure

4a. Advisor may deny list of courses taken due to uncompleted credits, technical elective issues or course collisions.

Allow student to make changes on schedule or adjustments.

Use Case: Registration Simulation From Observer Perspective

Actors: Observer, System

1. Observer starts the simulation.
2. System creates json files for each student containing their transcripts and course registration history.
3. Some students failed to register for some courses because they failed to meet some course requirements or course quotas were full.
4. System, classifies the students according to why they could not take the courses and displays the student IDs of these students along with the number of students who could not take the course for the same reason.