Marmara University Engineering Faculty

Department of Computer Engineering

CSE3063 - OBJECT ORIANTED SOFTWARE DESIGN

ITERATION #1

REQUIREMENTS ANALYSIS DOCUMENT (RAD)

Software Requirement Specification Document

1.Introduction

The purpose of the student registration system project is to simulate a course registration system for a particular semester with students of each year. And observe the potential outcome of every student by comparing their transcript before and after the simulated semester

2. Overall Description

We have a student course registration system simulation of CSE Department of Marmara university for a particular semester where random students of different years(freshman, sophomore etc.) and random advisors are created for a particular university semester (spring or fall). Also, the courses are created(including all the elective courses) according to the curriculum of Marmara University CSE department

After students and advisors are created, every student in the registration system is assigned an advisor to check their registration process for the course sections they want to register.

Then, the registration system randomly marks every student as passed, or failed to the courses that the student took before the simulated semester according to the curriculum with a certain probability given as an input to the registration system. The reason for registration system to fail some courses is to simulate the prerequisite conditions, hour collision condition, credit condition etc.

After registration system make every student pass or fail their previously taken courses, it then lists the offered courses for each student in the simulated semester one by one. Offered courses consists of firstly the courses that student took and failed(that is a must) plus, the courses of the current semester. And every student sends those lists to their advisors one by one for the advisor to either approve or disapprove the courses.

Advisors when their advised student sends them a course, will either approve or disapprove that course. Advisors approve the course registration request only if:

- The course section isn't full
- The prerequisite of the requested course is passed by the student
- The required credit for the course is satisfied by the student
- There is no more than one hour collision in student's schedule
- The student registers to courses no more than they can, according to the curriculum

If any of the conditions above isn't satisfied, the advisor don't approve the registration process.

If the disapproved course is an elective course, the student may choose another course inside the pool of electives whether it's a technical, faculty or non-technical elective. But if the course is not elective, student have no chance to register to that course in the simulated semester, so they move on with the next course in their offered course list.

After the registration process for each course for each student is finished, the registration system shows every student's transcript before and after the simulated course registration process(before means, after the registration system marks the students as passed or failed to their past courses) along with the statistics about the overall registration process such as how many students failed to register because of a collision, because of the course section being full etc. .

After the simulation ends, the user can analyze those transcripts and statistics to make assumptions about the registration process.

3. Ranking Requirements

High: Obtaining the offered course list for a student by taking the student's failed courses into consideration.

Medium: Checking the course approval conditions by the advisor.

Low: Listing the each transcript before and after the registration process

4. Fully Dressed Use Case

Use Case: Register to a Course

Actors: Student, Advisor, Registration System, Course

1. The student has a year, semester, and a list of passed courses and failed courses
2. An advisor is assigned to the student by the registration system
3. Registration System adds past courses to student as failed or passed with a certain probability.
4. Registration system lists the offered Mandatory courses and number of corresponding elective courses that is available in student's current semester for the student to request to register
5. Student browses through that list and chooses the courses they want to take
6. Student sends a registration request to their advisor for each course that they chose
7. For all of the courses, advisor checks if there is a collision. And for courses that has a credit condition or prerequisite condition, System checks if the course is approvable by looking at student's completed credits and passed courses accordingly.
8. After getting checked for approvability, student sends request to the corresponding course section in his list, and if the course section isn't full, the registration process is over for that section(student registered successfully).
9. Student requests to register until there is no course left in the offered course list.
10. Student successfully or unsuccessfully registered all the courses which registration system offered for simulated semester.
11. The registration system shows the transcript of the student before and after the registration process

Alternative Flow: There is more than one hour collision and Course is a Mandatory Course

7a. Student can't register that course in simulated semester because a mandatory course section has no alternative in the system unlike electives.

7b. Student moves with the next course in his offered course list.

Alternative Flow: There is more than one hour collision and Course is an Elective Course

7a. Student looks for another random elective course section of the same type from the list of courses in the registration system

7b. Student chooses an alternative course section from that list randomly.

7c. Student sends request for that course to his advisor.

Alternative Flow: Course Section is Full, and Course is a Mandatory Course

8a. Student can't choose another course section because every course has exactly one course section in the system.

8b. Student can't take that course section and moves to the next mandatory in his list.

Alternative Flow: Course Section is Full, and Course is an Elective Course

8a. Student asks the system if there is any course section with the same type(NTE, FTE, TE, UE) that is not full

8b.Registration System says there is at least one elective course section with the same type that is not full

8c. Student sends registration request to a random elective course section of the same type

8d. Advisor approves the new elective course section

Alternative Flow: All of the electives are full

8ba. Registration system says there is no elective course which is not full.

8ca. Student randomly sends registration order to the advisor for every elective course with the same type to get rejected to every one of them.

8cb. After iterating through every elective course section, student looks for another elective course section types in his list to register.

Alternative Flow: Prerequisite or Credit condition isn't met, and course is a Mandatory Course

8a. Student can't register that course in the simulated semester because a mandatory course section has no alternative in the system unlike electives.

8b. Student moves on with the next course in his offered course list.

Alternative Flow: Prerequisite Condition isn't met, and Course is a Technical Elective Course

8a. Student can't take that particular elective course.

8b. Student chooses another random course section of Technical Elective course.

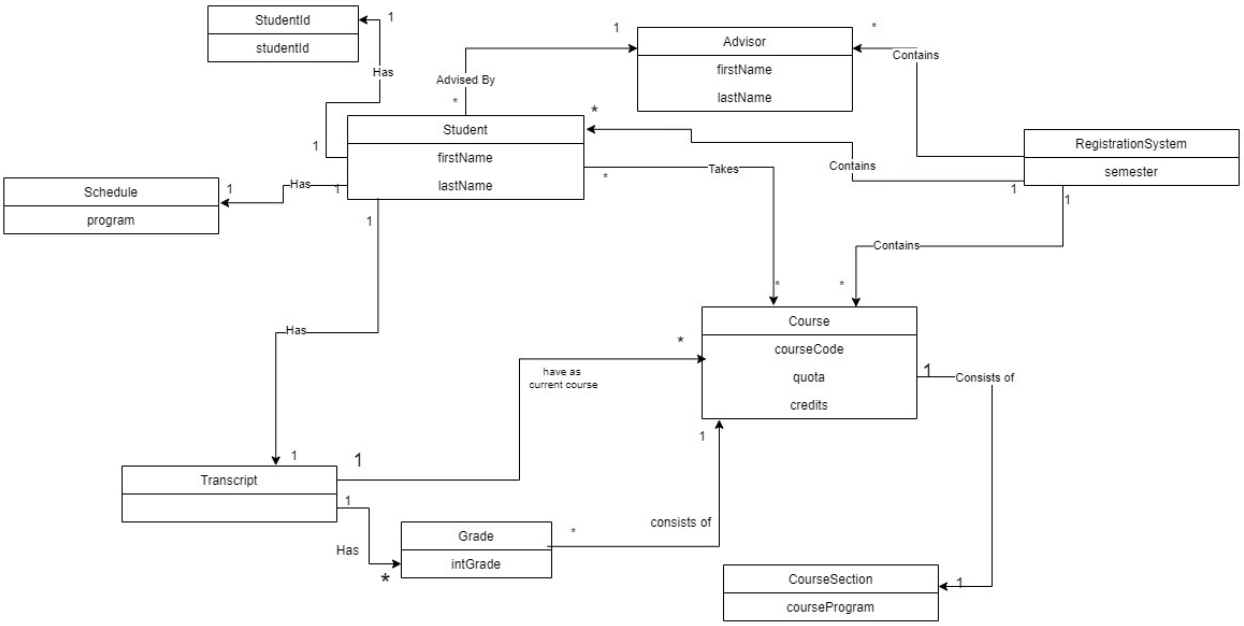
8c. Student sends registration request to his advisor for the chosen course section.

8d. Advisor approves the registration request for the newly chosen course section.

8e. Newly chosen course section isn't full and student successfullly registers to the course section.

8f. Student moves on to the other course sections in his offered course sections list.

5. Domain Class Diagram



6. Functional Requirements

Functional requirements describe system behaviours. Functional Requirements can be divided into three main headings:

1- Priority
2- Critically
3- Risks

The Functional Requirements in the project are listed below:

-The system will allow students to be added to and dropped out of courses.

-The system will allow students to see their grade which including student's all semester.

-The system will provide courses to student that students need to register.

-When system will student to allow register course, then system will provide a quota for all courses.

-When system provide courses to students, the system will check if students has prerequisite course or not.

-When system provide courses to students, the system will check if there is a collision between courses or not.

-The system will provide a schedule to student.

-When system provide courses to students if a course is full the system will not accept student.

- The system will allow advisors to approve courses that students select.

- İf a student cannot complete required course or complete credits the system will prevent students from enrolling to a course.

- The system will allow manage student course, approve, check prerequisite to advisors.

7.Non-Functional Requirements

Non-Functional Requirements describe other desired attributes of overall system. Non-Functional Requirements are divided into six main headings:

- Product Cost
- Performance
- Portability
- Availability
- Security
- Safety

The Non-Functional Requirements in the project are listed below:

- The system will protect the user's privacy

- The system has high performance
- The system will be maintainable.
- The system will have high accessibility.
- The system will be secure.
-The system will be highly portable.
- The system will not have any downtime.
- The system will be scalable.
- The system only will allow advisors to see grades and courses of all students.
- The system will have high efficiency.