Marmara University

Engineering Faculty - Computer Engineering

CSE3063 – Object-Oriented Software Design

Project Iteration-1

Requirements Analysis Document

Group-5 Members

Havva Nur Özen 150119771

İlker Keklik 150120074

Yasin Küçük 150120062

Kerim Hasan Yıldırım 150120040

Sarp Sıraç Müftüler 150119717

Yusuf Duman 150120023

Table of Contents

Description of The Project

The project aims to simulate a course registration system for Computer Engineering students at Marmara University. The registration system will contain the students, lecturers, advisors, course, and the Department. To simulate the system, students, lecturers, advisors, and courses will be created manually considering the necessary number of instances to access all the functionalities of the project.

Every person in the system has a randomly generated person ID assigned to them. Besides person IDs, all staff and students have randomly generated student IDs and staff IDs. To differentiate the ID types, the prefixes of the IDs are assigned differently for person ID, staff ID, and student ID.

All students have an advisor assigned to them to approve the requested courses by the student. Every course request needs approval from the advisor for the student to take the course.

All courses have semester information, and some courses have prerequisite courses.

There are conditions students must consider before requesting a course:

1) The student must have successfully completed prerequisite courses of the course.
2) The student must have enough completed credits to enroll for the course depending on the course's semester information.
3) The course the student wants to choose must be a valid course in the department.
4) The total of active courses and sent course requests of the student can't be more than the maximum allowed number of courses for a semester.

If all the conditions are not met by the student, the system will not let the student send the request to his/her advisor. If the student meets all the conditions, he/she can send a request to the advisor to take the course for its approval.

Advisors possess the ability to review and approve/reject course requests from students and access a list of students under their supervision. It has also more functional abilities for the courses he teaches and the students he supervises.

In addition to course operations, students can utilize the system to view their transcripts, encompassing information on total completed credits, both successful and unsuccessful course attempts, and corresponding letter grades.

Use Case for Course Registration System

Actors: Student, Advisor, System

1. Student accesses the Course Registration System and enters their student ID.
2. Student lists the courses he/she has taken.
3. Student displays the Transcript
4. Student lists courses that he/she has not completed and is not enrolled in.
5. Student sends a request to register for a new course.
6. Advisor accesses the Course Registration System and enters their staff ID.
7. Advisor lists active registrations under his/her responsibility.
8. Advisor lists the students he/she advises.
9. Advisor evaluates a registration request.
10. Advisor lists the courses he/she teaches.
11. Advisor assigns grades to the students of the courses he/she teaches on the next screen after listing their courses.

Alternative

1a. If an invalid student ID is entered, the system displays an error message and prompts the user to re-enter a valid ID.

1b. Upon entering a valid student ID, the student is successfully logged in.

Alternative

5a. If the user enters a course code that does not exist or wants to register for a course that cannot be registered for another reason, an error message is returned along with the reason.

5b. If eligible for registration, the Registration Request is sent to the Student's Advisor.

Alternative

6a. If the user enters an invalid staff number, an error message appears and the options are repeated.

6b. Advisor is logged in when the valid staff number is entered.

Functional Requirements

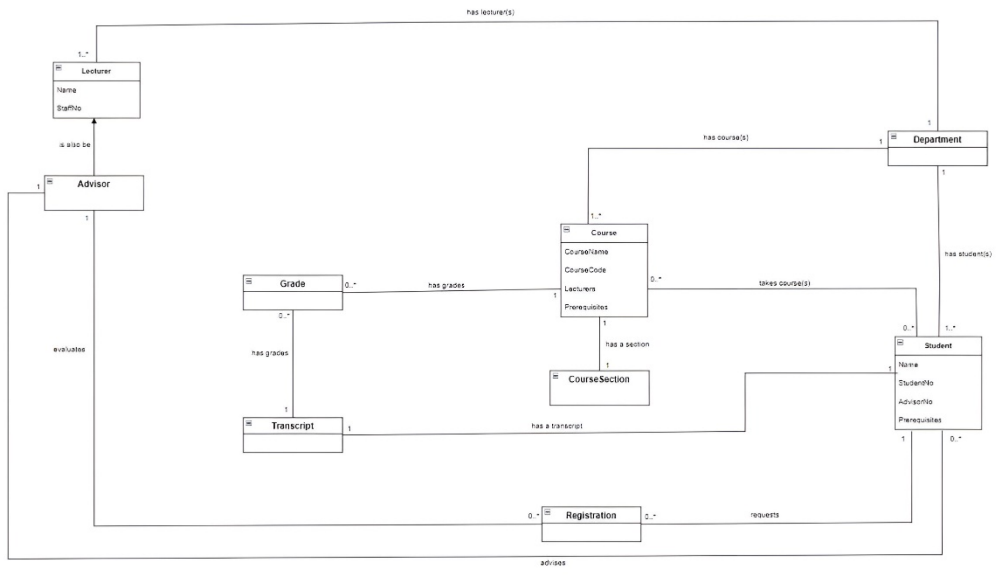
1. The system must ensure that each student and advisor is assigned an identical student number and staff number, respectively.
   - **Priority**: High
   - **Criticality**: It is crucial for both students and advisors to log in to the system and perform essential operations.
   - **Risks**: Failure to provide identical student or advisor numbers will result in the absence of a private account for users, preventing them from carrying out necessary operations.
2. The system must provide the available courses for students to send course registration requests.
   - **Priority**: High
   - **Criticality**: This functionality is crucial for students to make informed decisions about their course registration, impacting the core user experience.
   - **Risks**: If the available courses are not accurately presented, students might request courses that are not being offered, leading to confusion and potential enrollment issues.
3. The system must provide a transcript for each student.
   - **Priority**: High
   - **Criticality**: While important for students to track their academic progress, it is not as time-sensitive as some other functionalities.
   - **Risks**: Not providing transcripts can cause students not to complete any course.
4. The system must control the prerequisites of courses when students send registration requests.
   - **Priority**: High
   - **Criticality**: Essential for ensuring that students meet necessary requirements before attempting a course, preventing enrollment in classes for which they are not adequately prepared.
   - **Risks**: Without proper prerequisite control, students may register for courses they are not qualified to take, leading to difficulties and disruptions in the learning process.
5. The system must provide advisors with the sent course registrations.
   - **Priority**: High
   - **Criticality**: Critical for advisors to review and manage student course registrations, ensuring proper academic guidance and resource allocation.
   - **Risks**: If advisors cannot access sent course registrations, there is a risk of delayed or inefficient approval processes, impacting the overall efficiency of the academic advising system.

6. The system must assign Lecturers to courses.
    - **Priority**: High
    - **Criticality**: Important for providing students with grades to complete the courses.
    - **Risks**: Without proper advisor assignments, students cannot receive grades which means they cannot complete the courses.
7. The system must allow advisors to give grades to students in the selected course.
    - **Priority**: High
    - **Criticality**: Essential for students to complete the courses.
    - **Risks**: If advisors cannot assign grades, students can not receive proper feedback, and academic records may be incomplete or inaccurate.
8. The system must provide advisors with the courses and students of the courses.
    - **Priority**: High
    - **Criticality**: Crucial for advisors to have access to the necessary information for effective academic supervision and support.
    - **Risks**: Without proper access to course and student information, advisors cannot reach the students of the courses and provide them with grades, potentially impacting students' transcripts.
9. The system must save all data before terminating the program.
    - **Priority**: High
    - **Criticality**: Critical for data integrity and system reliability, ensuring that no information is lost during or after system shutdown.
    - **Risks**: Failure to save data before termination may result in the loss of critical information, impacting the overall functionality and reliability of the system.

Non-Functional Requirements

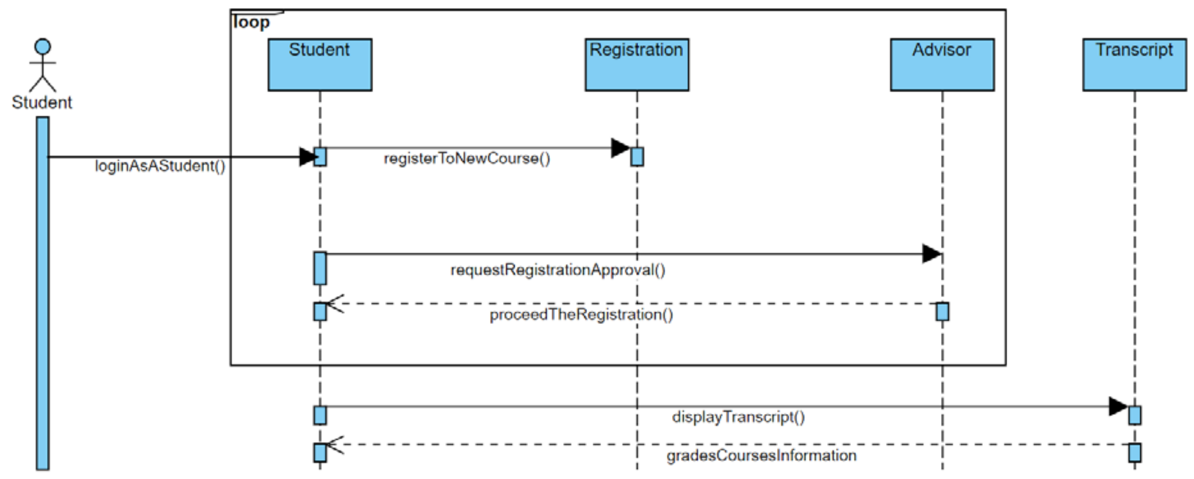
1. **Security:** The system must implement robust security measures to protect user data and prevent unauthorized access.
2. **Performance:** The system should provide efficient response times, ensuring a smooth user experience.
3. **Availability:** The system must have minimal downtime, ensuring continuous availability for users.
4. **Maintainability:** The system should be designed for easy maintenance, with considerations for extensibility and reusability of code components.
5. **Portability:** The system should be compatible with standard web browsers, ensuring accessibility across different platforms.
6. **Product Cost:** The system development and maintenance costs should be kept within a defined budget.

Domain Model

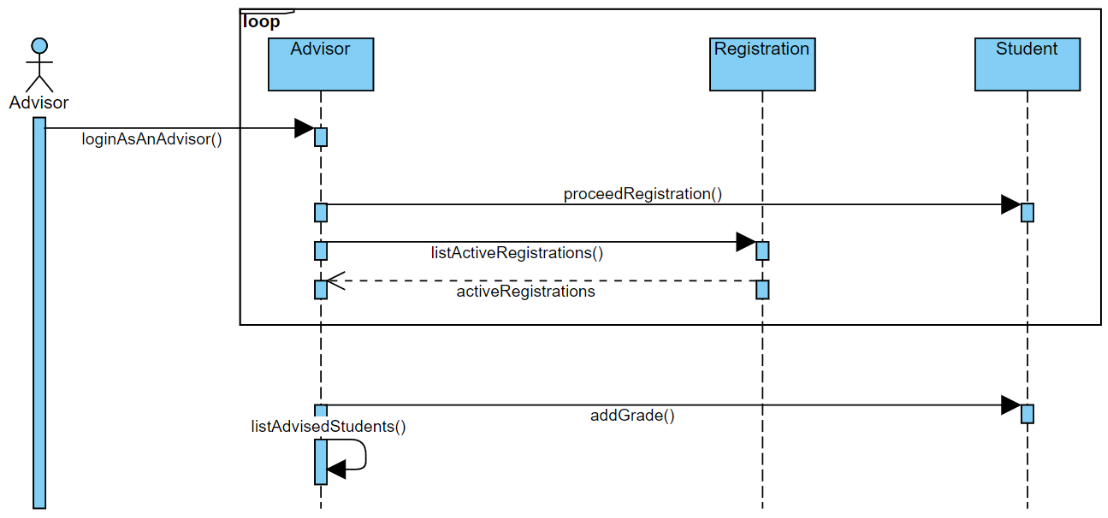


System Sequence Diagram (SSD)

1) Actor: Student



2) Actor: Advisor



Glossary

1. **Advisor:** An individual assigned to guide and approve course requests for students, assisting in academic planning.
2. **Course Registration System:** A simulated platform designed to mimic the process of registering for courses, involving students, lecturers, advisors, courses, and the Department.
3. **Prerequisite Courses:** Courses that must be successfully completed before a student can enroll in a specific course.
4. **Transcript:** A record of a student's academic history, including completed and failed courses, total credits, and letter grades.
5. **Criticality:** The importance of a requirement to the overall system functionality and user experience.
6. **Priority:** The ranking order of features or functionalities based on their importance.
7. **Risk:** Potential issues or challenges that may arise in meeting a specific requirement, along with strategies to mitigate them.
8. **Semester:** One half of an academic year.
9. **Credit:** A numerical representation of the value assigned to a course.