Marmara University Engineering Faculty
Department of Computer Engineering

CSE3063 – Object Oriented Software Design

Course Registeration System
Requirement Analysis Document (RAD)
Iteration #2

Instructor: Murat Can Ganiz

January 7, 2024

STUDENTS – GROUP 16

BURAK KARAYAĞLI - 150121824
BARIŞ GİRAY AKMAN - 150121822
EFE ÖZGEN - 150121077
HAKKI KOKUR - 150120033
MUSA ÖZKAN - 150121058
TAMER ÜNAL - 150119795
ATİLLA GEL - 150119564

1 Introduction

The name of the project is Course Registration System. Main purpose in the project is providing users to get assigned to the necessary lectures through their advisors and simplifying the registration process of the course registration.

Users will enter the system through the user name and password given during the registration to the system. To enable the course at the system, there must be at least one lecturer to lecture. Otherwise, course won't be enabled for that semester. Each student who have taken any of the courses will be graded through the system at the end of the semester. And these grades will be processed through transcript.

2 Scope

The project's scope is to develop a robust university course registration system, accommodating various user roles: students, advisors, lecturers, and staff members.

The system will offer a range of essential functionalities, including user authentication, course enrollment, grade management, and administrative tools. The core entities within the system, such as Person, Student, Advisor, Lecturer, and Course, will be intricately connected to facilitate operations like course registration, student management, course approval, and transcript viewing. Data management will rely on JSON files and Command-line menus will serve as the user interface, enabling users to interact seamlessly with the system.

3 Constraints of the System

Certain functionalities are explicitly out of scope for this project. For instance, the project does not involve the development of a graphical user interface (GUI) for the system, meaning users will interact solely through command-line interfaces. Additionally, the project uses JSON files for data storage, but it does not implement a full-fledged relational database system. Consequently, advanced database management features, such as complex queries and optimizations, are not part of the scope.

While the project defines various roles, like students, advisors, and lecturers, it does not address the creation or management of user accounts and authentication, assuming that these aspects are managed separately.

Finally, the system does not incorporate real-time or web-based features, meaning that all interactions occur within the confines of the command-line interface.

4 Requirement Analysis

4.1 Vision

Course registration system for computer engineering department. It is a system for student's choosing courses and waiting for approval from the advisors.

4.2 Glossary

4.2.1 Person

Definition: Represents an individual within the system.
Purpose: Serves as the base entity for various user roles like students, staff, advisors, and lecturers.
Example: Contains attributes such as name, surname, username, and password.

4.2.2 Staff

Definition: Represents university staff members.
Purpose: Includes employees like advisors and lecturers who have specific responsibilities related to managing students and courses.
Example: Staff members can set office hours, manage student registrations, and have roles like advisors or lecturers.

4.2.3 Advisor

Definition: A staff member responsible for advising students.
Purpose: Guides and supports students in academic matters, assists with course selection, and manages student course organization.
Example: Advisors maintain lists of assigned students and facilitate course approvals.

4.2.4 Lecturer

Definition: A staff member responsible for teaching courses.
Purpose: Delivers course content, facilitates discussions, and assesses students' performance through assignments and exams.
Example: Lecturers conduct lectures, manage course materials, and evaluate student progress.

4.2.5 Student

Definition: An individual enrolled in the university.
Purpose: Pursues academic programs, selects courses, and maintains academic records within the system.
Example: Students have personal details, academic statuses, and can enroll in or drop courses.

4.2.6 Course

Definition: Represents a university course.
Purpose: Provides information about course content, prerequisites, and scheduling for student enrollment.
Example: Courses have attributes like name, description, prerequisites, and scheduling details.

4.2.7 Mandatory Course

Definition: A type of course that students must take.
Purpose: Essential components of academic programs, ensuring students fulfill degree requirements.
Example: Mandatory courses have specific lecture dates, quotas, and, in some cases, lab hours.

4.2.8 Non-Technical Elective Course

Definition: An elective course unrelated to technical subjects.
Purpose: Offers students flexibility to choose courses outside their core academic disciplines.
Example: Non-technical elective courses have lecture dates, quotas, and may be offered remotely.

4.2.9 Technical Elective Course

Definition: An elective course related to technical subjects.
Purpose: Allows students to choose specialized courses within their academic discipline.
Example: Technical elective courses may have prerequisites, lecture dates, quotas, and required credit values.

4.2.10 Grade

Definition: Represents a student's performance grade in a specific course.
Purpose: Indicates a student's performance and progress within a course.
Example: A grade includes the course and the associated grade (e.g., "A," "B," "C," etc.).

4.2.11 Time Interval

Definition: Represents a time interval, such as lecture times.
Purpose: Used to define course schedules, office hours, and other time-related data.
Example: A time interval includes start and end times and the day of the week.

4.2.12 Course Section

Definition: Details of the class, time, and lecturers for the courses.
Purpose: Course sections provide specific information about when and where a course is held, as well as who is responsible for its instruction.
Example: A student checks the course section to find out the schedule and location of a particular class.

4.2.13 JSON

Definition: Notation used for data storage.
Purpose: JSON (JavaScript Object Notation) is a lightweight data-interchange format used to store and exchange information between systems.
Example: A web application uses JSON to transmit data between the server and client in a structured format.

4.2.14 Course Registration

Definition: The process of courses being selected by students and approved by advisors.
Purpose: Course registration enables students to choose the classes they wish to attend, while also ensuring that advisors can review and approve their selections.
Example: During course registration, a student selects their desired classes through the university's online portal, and the advisor approves the choices.

4.3 Requirements

4.3.1 Functional Requirements

- The system must support different roles, including Students, Advisors, Lecturers, and Staff.
- Students must be able to enroll in various types of courses, including Mandatory Courses, Non-Technical Elective Courses, and Technical Elective Courses.
- Students must be able to view the details of the courses they have enrolled in.
- Advisors must be able to manage a list of students.
- Advisors must be able to organize the course selections of their assigned students.
- Advisors must be able to approve or reject course selections made by students.
- Advisors must be able to add and delete students.
- Lecturers must be able to manage and view the courses they are responsible for.
- Lecturers must be able to add and delete courses they are lecturing.
- Staff members must have access to certain administrative functionalities, including managing user accounts.
- The system must allow for course prerequisites to be defined.
- The system must keep track of grades for individual courses.
- Students must be able to add and drop courses within specific timeframes.
- Students must be able to view their transcripts.
- The system must support the concept of "Sections" for courses, including setting dates and quotas for sections.
- The system must provide user authentication and login functionality.
- The system must ensure data consistency and maintain relationships between various entities, such as students, courses, and advisors.

4.3.2 Non-Functional Requirements

- The system must accept at max 5 courses for each student
- The system must inform the user at the end of operations
- The system should maintain detailed logs os system activities and errors for debugging and auditing purposes
- The system should comply with relevant data protection and privacy regulations
- The system should have at least 40 students from all classes, 5 advisors, lecturers and at least 10 different courses including NTE, FTE, TE at least 5 of them having prerequisites.

5 Use Cases

5.1 Use Case: Attempt to Login to Systems

Specialization of: Authenticate

Actor Actions	System Responses
1. User enters their username and password.	2. System checks the provided credentials.
	3. If credentials are valid, the system retrieves the user's information and displays it along with the main menu.
	4. System responds with a success message: "Authentication successful."
	5. System displays the user's information and the main menu.

5.2 Use Case: Attempt to login with invalid username or password

Extensions of: Authenticate

Actor Actions	System Responses
1. User enters their username and password.	2. System checks the provided credentials.
	3. If credentials are not valid, the system retrieves warning message and return back to login page

5.3 Use Case: Student Enrolls in Course

Specialization of: Registration for the Course

Student Actions	System Responses
1. Student pick the course selection from menu	
2. Student selects courses to enroll in.	3. System checks the whether the student has met the requirements for the selected courses
	4. System adds the selected course to the student's list of selected courses.
	5. System responds with a success message: "Course enrolled successfully."

5.4 Use Case: Student hasn't met the requirements

Extensions of: Registration for the Course

Student Actions	System Responses
1. Student selects courses to enroll in.	2. System checks the whether the student has met the requirements for the selected courses
	3. The system does not add courses that students are not eligible to take.
	4. System responds with a warning message.

5.5 Use Case: Checks the requirement

Includes: Registration for the Course

Student Actions	System Responses
1. The student send the selections	2. The system begins checking the criteria.
	3. The system checks whether the student's credit limit is sufficient to add the selection.
	4. The system checks the maximum course selection limit.
	5. The system checks for overlapping in the selected courses.
	6. The system checks whether the prerequisite courses have been taken or not
	7. The system returns the success or warning message.

5.6 Use Case: Student Drops Courses

Specialization of: Dropping the Course

Student Actions	System Responses
1. Student selects courses to drop.	2. System removes the selected courses from the student's list of selected courses.
	3. System responds with a success message.

5.7 Use Case: Student View Selected Courses

Specialization of: View Selection

Student Actions	System Responses
1. Student requests to view their selected courses.	2. System retrieves and displays the list of courses selected by the student.
	3. System responds by displaying the list of selected courses.

5.8 Use Case: Student Views Transcript

Specialization of: View Transcript Information

Student Actions	System Responses
1. Student requests to view their transcript.	2. System retrieves and displays the student's transcript, including grades for completed courses.
	3. System responds by displaying the student's transcript.

5.9 Use Case: Student Send Approval Request

Specialization of: Sending selection to Advisor

Student Actions	System Responses
1. Student sends an approval request to him/her advisor	2. The system locks the student's course selection.
	3. The system grants the advisor editing permission for that student's course selection.
	4. The system returns a success message to the student.

5.10 Use Case: Lecturer views the courses assigned to him.

Specialization of: View assigned courses

Lecturer Actions	System Responses
1. Lecturer requests to view the courses they are responsible for.	2. System retrieves and displays the list of courses associated with the lecturer.
	3. System responds by displaying the list of courses.

5.11 Use Case: Lecturer views their students

Specialization of: View students

Lecturer Actions	System Responses
1. Lecturer requests to view the list of students they are teaching.	2. System retrieves and displays the list of students associated with the staff member.
	3. System responds by displaying the list of students.

5.12 Use Case: Lecturer add course

Specialization of: Creating Course

Lecturer Actions	System Responses
1. Lecturer select the add course section from menu	
2. Lecturer provides the details of the course (course information).	
	3. System adds the course to the lecturer's list of lectured courses.
	4. System responds with a success message: "Course added successfully."

5.13 Use Case: Advisor Organizes Student Course Selection

Specialization of: Organize Selection

Advisor Actions	System Responses
1. Advisor selects the student he/she wants to edit.	2. System check the advisor have access to edit.
	3. System retrieves students' course selections.
4. Advisor add or remove courses to list and send to system.	5. System edit the student course selection

5.14 Use Case: Advisor Approve/Deny Student Selection

Specialization of: Approve or Reject Course Selections

Advisor Actions	System Responses
1. Advisor selects the students who need approval.	2. System list the student selections.
	3. system prints a course with warnings.
4. Advisor selects the approve or deny button.	5. System write the selection to database and update states
	6. Proceed to step 2 and repeat until all the courses are completed.

5.15 Use Case: Advisor view Students Information

Specialization of: View Student Info

Advisor Actions	System Responses
	1. System retrieves and displays student list.
2. Advisor select a student	3. The system prompts the user to specify the type of information they want (transcript, selected courses, general information, etc.).
4. Advisor select	5. System retrieves the requested information.

5.16 Use Cases: View Information

Generalization of: View Student Info, View students, View assigned courses, View Transcript Information, View Selection

Actor Actions	System Responses
1. The user selects "INFO" from the main menu	2. The system retrieves the information options
3. The user selects one of them	4. The system prints the requested information page.
	5. The system returns to the main menu.

5.17 Use Cases: Going Back to The Previous Menu

Includes: Approve or Reject Course, Organize Selection, Creating Course, Dropping the Course, Registration for the Course

Actor Actions	System Responses
1. The user send "-1" instead of other choice	2. The system back to the previous menu.
	3. The system prints previous page.

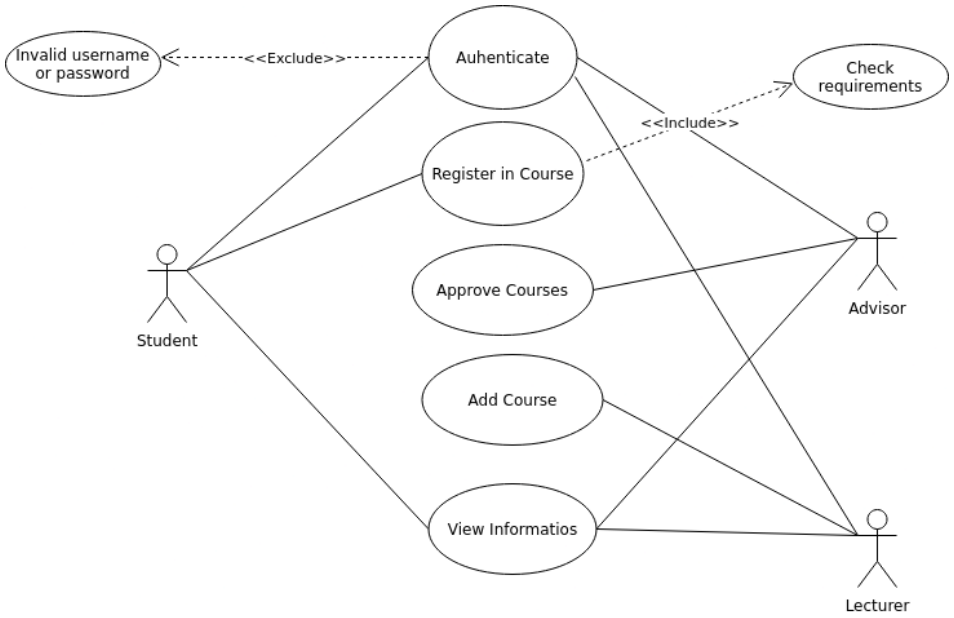

Figure 1: Use Case Diagram