Requirement Analysis Document

Student Lecture Registration System

	Name	Surname	Number
1.	Ahmet Abdullah	Gültekin	150121025
2.	Ömer	Deligöz	150120035
3.	Melik	Özdemir	150120004
4.	Hamza Ali	İslamoğlu	150120063
5.	Enes	Haksoy	150120024
6.	Mehrin	Kaya	150123824
7.	Ayşe Gülsüm	Eren	150120005

1) Introduction

In this project, our aim is to create a student course registration system. This system will be like the course registration system used by our school. Students will interact with their advisors according to specific rules to successfully complete or not complete the course registration process.

1.1) Purpose of the System

Our system's purpose is to enable students to effectively track their course statuses and enhance communication with student advisors.

1.2) Scope of the System

The system scope includes interactions related to course registrations, encompassing activities such as course enrollment, approval/rejection, and viewing student information.

1.3) Objectives and Success Criteria of the Project

a) Objectives of the Project:

- Developing a simple Command Line Interface for students to manage their course registrations and academic information.
- Making the course registration process more efficient and accessible for students.
- Creating interaction between students and advisors
- Enabling the advisor to access student requests and course statuses.

b) Success Criteria of the Project:

- The system should be accessible accurately and promptly by both students and advisors.
- The system should process login information accurately and present correct data to the users.
- It should accurately reflect user requests, course statuses, and added data in a current and precise manner.
- Ensuring accurate processing of added data and maintaining consistency within the system.

- The system should be user-friendly and effective based on user feedback.
- The system should effectively reflect student and advisor interactions through feedback.

2) Current System

-

3) Proposed System

A system operating in accordance with the principles of Marmara University course registration system.

3.1) Overview

In the project, there are two types of users: advisor and student. Each user type has its own viewing menus with distinct displays. A student can view his/her requests, course registrations, transcripts, advisor information, and check whether the registration is successful or not. Similarly, advisors can view the information and request of students and provide feedback such as approval or rejection of requests.

3.2) Requirements

a) Functional Requirements

1. Student Requirements

- Student must have an advisor.
- Student must have student ID.
- Student must have transcript.
- Student must have class year.
- Student may have request.
- Student must have username to login.
- Student must have password to login.

2. Advisor Requirements

- Advisor must have an ID.
- Advisor must have username to login.
- Advisor must have password to login.
- Advisor must have students that they advise.
- Advisor can have list request(s).

3. Registration Requirements

- Registration system includes a student and student's desired courses.
- Registration system sends the list of selected courses to the advisor.

4. Course Requirements

- Course must have course name.
- Course must have course code.
- Course must have credit score.
- Course may have pre-requisite course(s).
-

b) Non-Functional Requirements

1. Obtaining Student Information from JSON Input:

- Student information should be obtained from JSON files as input.

2. Secure Storage and Transmission of User Data:

- User data should be stored correctly.

3. User Interface:

- The user interface should be easily understandable and user-friendly.
- Layout should be carefully selected to enable users to access information and interact with the system effortlessly.

4. **Implementation of Registration System Program with JAVA:**
    - The registration system program must be implemented using the JAVA programming language.
    - The latest stable version of JAVA and appropriate libraries should be used.
5. **All Steps Should Be Performed on the Command Line:**
    - All registration and other processes should be carried out via the Command Line interface.
    - This allows users to interact with the system through command-line instructions for management purposes.

4) Use Cases

a) Student Use Case

Scope: Course Registration System
Level: User Goal
Primary Actor: Student
Description: This use case outlines the process through which a student registers for a course using the Course Registration System.
Stakeholders and Interests:

• **Student:** Requires efficient and prompt course operations as it directly impacts their academic career.
• **Advisor:** Accepts or rejects students' selected courses in a manner that maximizes productivity for students.

Preconditions:

- Students' and Advisors' information is identified and authenticated.

Success Guarantee:

- The courses chosen by students are accepted.
- Students are successfully enrolled in the chosen courses.
- Transcripts are promptly updated.

Main Success Scenario:

1. Student logs into the system using their username and password.
2. Student views a list of available courses.
3. Student adds or drops courses as needed.
4. Student saves the draft.
5. Student submits the draft to the Advisor.
6. The courses selected and submitted to the Advisor are accepted.
7. Student logs out of the system.

Alternative Scenarios:

1. If the username or password is incorrect, the student cannot log into the system.
2. If a selected course is already full, the system displays an error message, and the registration is canceled.
3. If a course is rejected by the advisor, the system notifies the student.
4. Student exits the system.
5. Student returns to the main menu.
6. Student views their own transcript.
7. Student lists the courses they have chosen.
8. Student checks the status of their request.
9. Student exits without saving the draft.

b) Advisor Use Case

Scope: Course Registration System

Level: User Goal

Primary Actor: Advisor

Description: This use case outlines the process by which an advisor manages course operations for courses chosen by students in the course registration system.

Stakeholders and Interests:

- **Student:** Requires efficient and prompt course operations as it affects their academic career.
- **Advisor:** Accepts or rejects students' chosen courses in a manner that is most productive for students.

Preconditions:

- Students' and Advisors' information is identified and authenticated.

Success Guarantee:

- The courses chosen by the student are accepted.
- Drafts are saved.
- Notifications are sent to students.

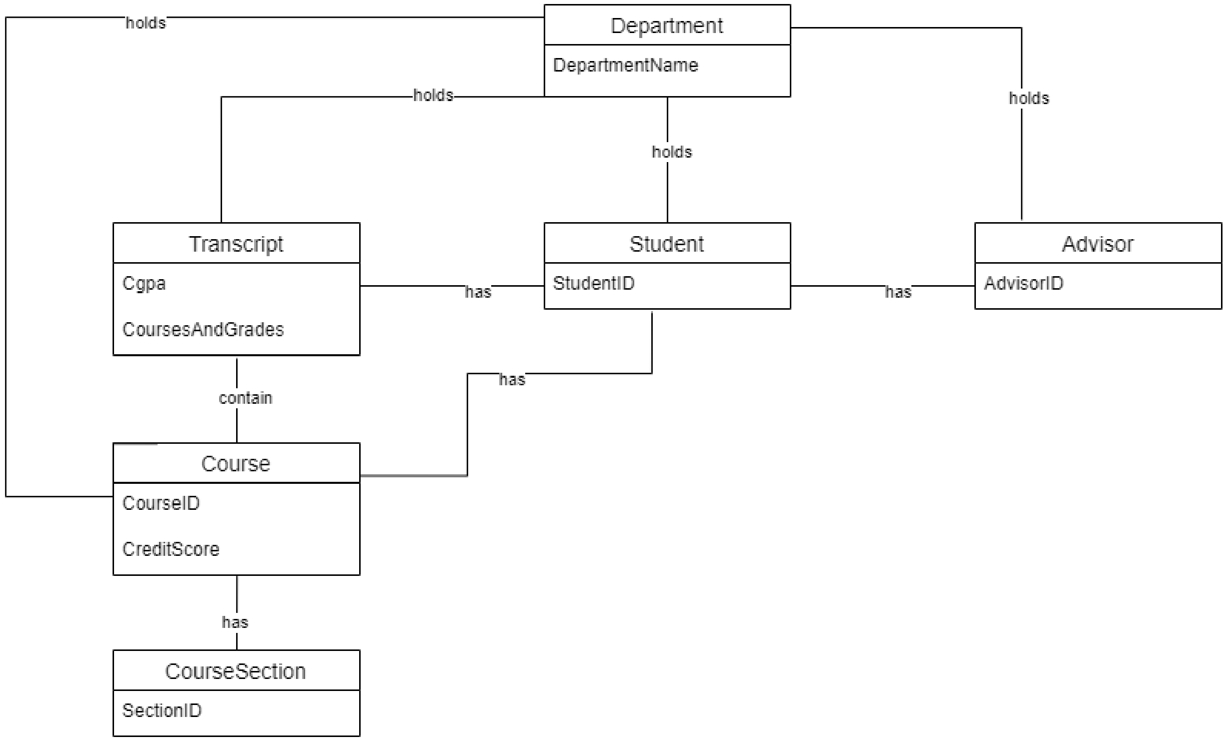
Main Success Scenario:

1. Advisor logs into the system by entering the username and password.
2. Advisor checks the incoming course requests.
3. Advisor accepts the courses chosen by the student.
4. Advisor saves the drafts.
5. The system sends a notification to the student about the request status.
6. Advisor logs out of the system.

Alternative Scenarios:

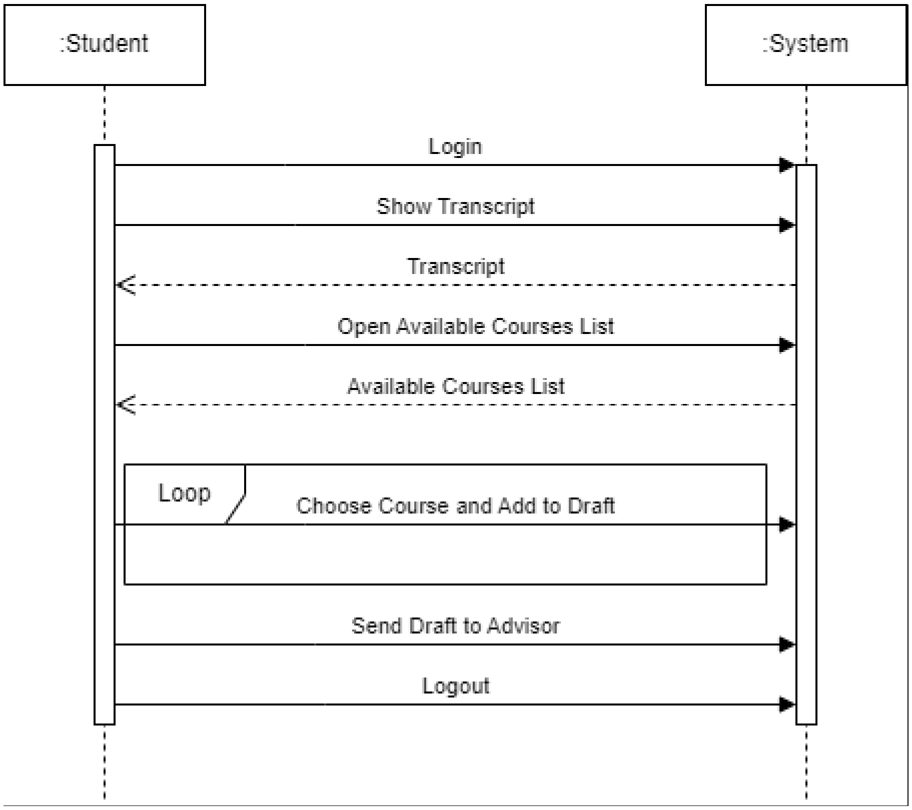
1. If the username or password is incorrect, the system notifies the advisor and provides another chance to enter the system.
2. If the advisor identifies issues with the selected course, the advisor rejects the request, and a notification is sent to the student.
3. If the advisor wants to review the requests of a student chosen by the advisor, they can perform this operation.

5) Domain Model

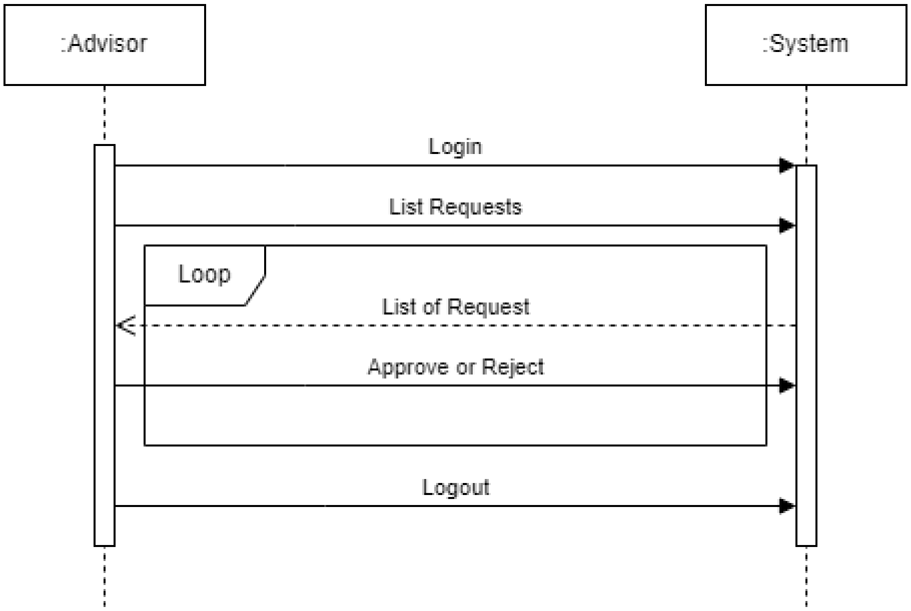


6) System Sequence Diagram (SSD)

a) Student SSD



b) Advisor SSD



7) Glossary

- **Student Lecture Registration System:**
  A software system designed to facilitate the course registration process for students in an educational institution.
- **Advisor:**
  An academic or administrative staff member responsible for guiding and assisting students in academic matters, including course selection and registration.
- **Command Line Interface (CLI):**
  A text-based user interface where users interact with the software by typing commands into a terminal or command prompt.
- **JSON (JavaScript Object Notation):**
  A lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate.
- **Functional Requirements:**
  Specifications describing the functions and capabilities that a system must possess, including both user and system interactions.
- **Non-Functional Requirements:**
  Specifications that describe the qualities and characteristics the system must have, such as security, performance, and usability.
- **JAVA:**
  A high-level, class-based, object-oriented programming language designed for creating robust and secure applications.
- **Use Case:**
  A description of how a system will be used, typically outlining interactions between users (actors) and the system to achieve a specific goal.
- **Transcript:**
  A document that provides a record of a student's academic performance, including grades and courses completed.
- **Course Code:**
  A unique identifier assigned to a specific academic course.

- **Credit Score:**

  A numerical representation of the academic credit value assigned to a course, indicating the workload and difficulty level.
- **Pre-requisite Course(s):**

  Courses that must be successfully completed before a student is allowed to enroll in a particular course.
- **Main Success Scenario:**

  The expected sequence of actions and events leading to the successful accomplishment of a use case.
- **Alternative Scenario:**

  Variations in the sequence of actions and events that may occur in response to different conditions or inputs during the execution of a use case.