FALL 2022

CSE 3063

OBJECT ORIENTED SOFTWARE DESING

ITERATION 2

REQUIREMENT ANALYSIS DOCUMENT (RAD)

Prepared by

GROUP 6

MARMARA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

TABLE OF CONTENTS

Before Begin:

This document was prepared using the Requirement Analysis Document that we prepared for iteration 1. It covers all changes, including additions, deletions, and rephrasing, in the latest document.

The changes in this document are represented by using highlight colors yellow, blue and red.

- Yellow Highlight ( ) stands for additions.
- Blue Highlight ( ) stands for rephrasing/ revising. (Especially for paragraphs)
- Red Highlight ( ) stands for deletions.

---

1. STAKEHOLDERS

- **Customer**
  - Murat Can GANİZ
- **Developers**
  - İsmail ÖKSÜZ
  - Emre SAĞIROĞLU
  - Serkan KORKUT
  - Onur ALKURT
  - Mustafa YANAR
  - Erdem PEHLİVANLAR
  - Ertan KARAOĞLU
  - Yasin ÇÖREKÇİ
  - Mertkan TURAN
  - Mehmet Akif GÜLMÜŞ
  - Yunus KAYA

2. VISION

We created a simulation for the Student Course Registration System that addresses the procedures and guidelines according to the policies and guidelines of the Marmara University Computer Engineering Department. This simulation is intended to help students understand and follow the necessary steps for registering for courses at the university.

3. SCOPE

The scope of this project is to design and implement a simulation of the Student Course Registration System at Marmara University Computer Engineering Department. This simulation will cover the procedures and guidelines that students need to follow in order to register for courses, as well as the policies and guidelines of the department. The simulation will be designed as a learning tool for students, but will not include the development of an actual course registration system or its integration with existing university systems.

4. PROBLEM STATEMENTS

The purpose of this project is to resolve problems that may occur in the student registration system. It checks that students can only enroll in courses that satisfy certain requirements. For example, the course must not be fully enrolled, the student must have completed any necessary prerequisites, and the course schedule must not overlap with other courses the student is taking. If these conditions are met, the student can successfully add the course to their schedule.

5. FUNCTIONAL REQUIREMENTS

- Students are created by using a json file for each semester.
- Advisors are added by using a json file.
- When students are created, an advisor is defined for each student.
- The software will use a json file to read the list of courses, types of courses, prerequisites, quota, and semester (1 to 8).
- When a student wishes to add a course to their calendar, the registration system makes sure there are no conflicts.
- The system must make sure that all prerequisites for a course are taken before adding it to a student.
- A student must be able to enroll in a course as long as there is availability in the course's quota.
- To enroll in technical elective courses or graduation project, a student must have completed the required number of credits. The system must ensure that this requirement is met.

6. NON-FUNCTIONAL REQUIREMENTS

- Java programming language was used for this project.
- Junit 5 was used to prepare Unit Tests.
- Log4j was used to log the actions in a log file.
- Draw.io was used to draw diagrams.
- We distributed tasks by a Kanban Board on monday.com.
- We distributed tasks by a Kanban Board via Trello.
- Discord was used for group meetings.
- Discord was used for group meetings that we made online.
- We set our appropriate meeting time using Doodle.
- We write our documents using Google Docs.
- No Graphical User Interface was used in this project.

7. USE CASES

Actors: Student, Advisor

1- Student enters the course registration system.

2- Student chooses the courses in the course registration system.

2- Student adds the courses in the course registration system.

3- Sends it to the advisor for approval.

3- Student sends the chosen courses to the advisor for approval.

4- Advisor approves the registration.

4- Advisor checks collision, course quota and prerequisites courses.

5- Advisor approves if the student is suitable to take the course.

6- Course registration is completed.

7- The system shows log records.

Extensions:

2.a) If the student has not successfully completed the prerequisite of the course he/she wants to register, he/she cannot take the course.

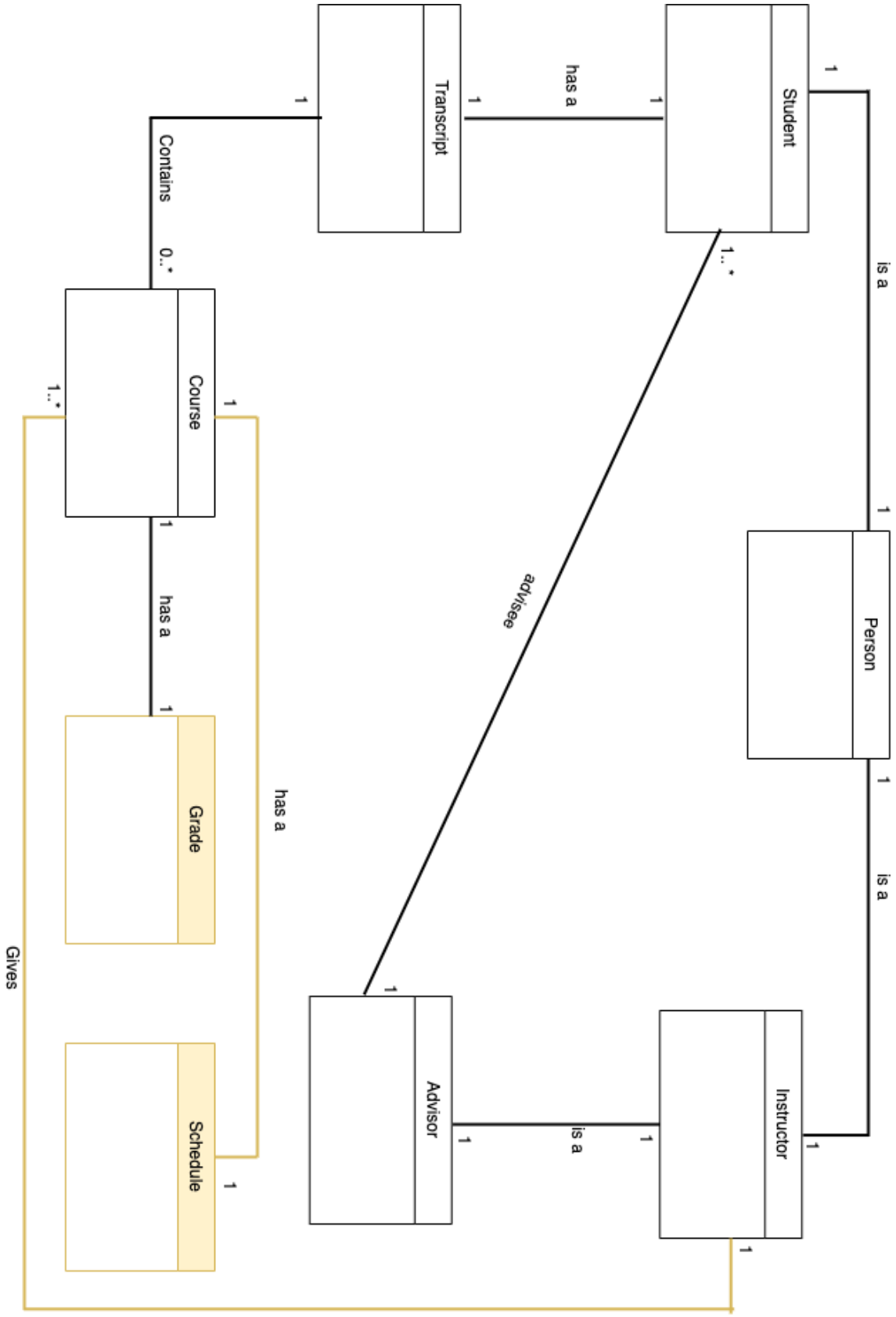
2.a) If the student's credit is less than the minimum required credits to take the course, the student cannot take the course.

4.a) If the student has not successfully completed the prerequisite of the course he/she wants to register, he/she cannot take the course.

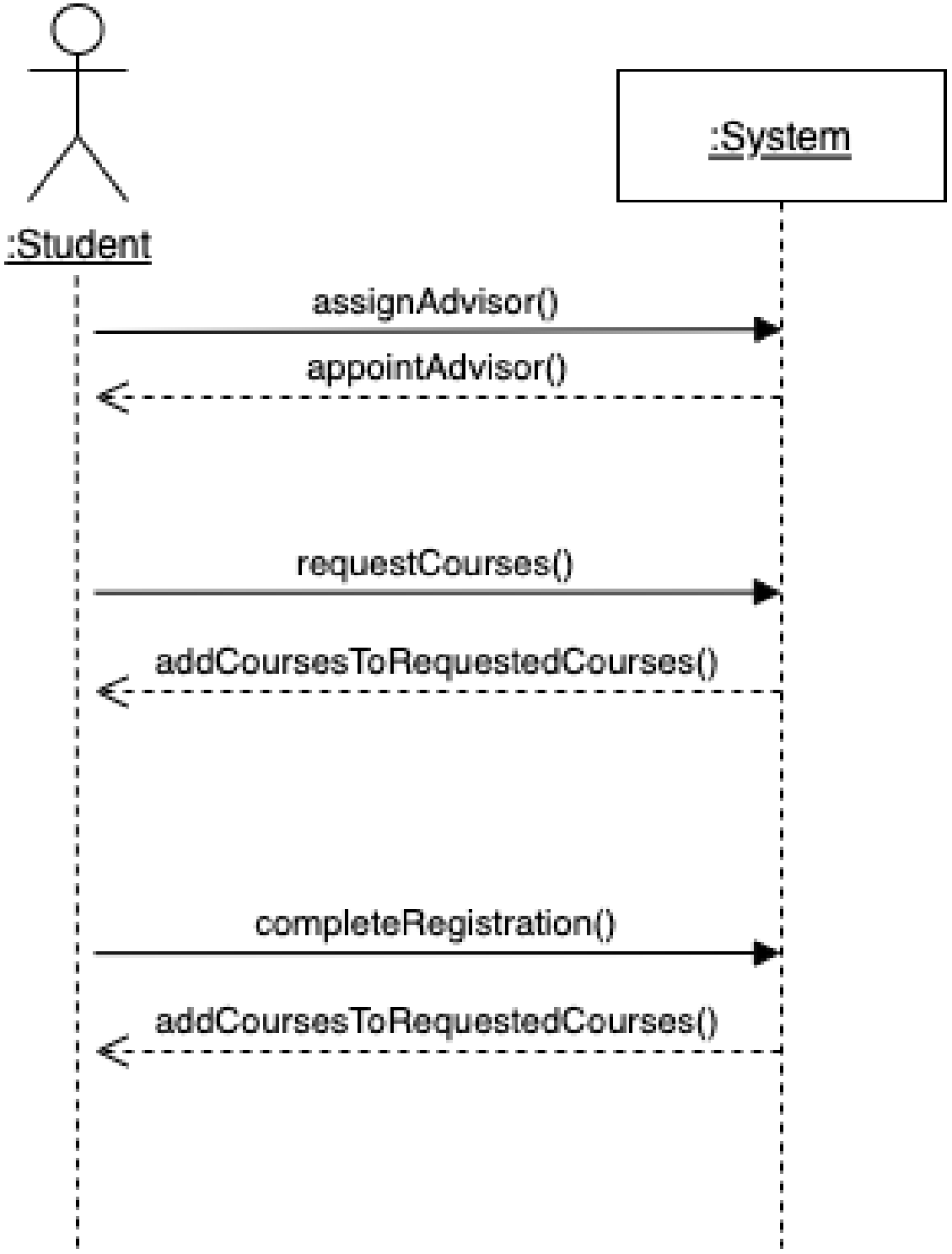
4.b) If the course quota is full, the advisor will not approve the course enrollment.

4.c) If the course that the student wants to take overlaps with another course, the advisor will not approve the course enrollment.

8. DOMAIN MODEL



9. SYSTEM SEQUENCE DIAGRAM:



10. GLOSSARY

- **Advisor:** A person whose duty it is to offer guidance to students on enrolling in courses.
- **Course:** Lessons that students need to pass in order to graduate.
- **Course Grade:** Average of the students grades in the assignments for a specific course.
- **Credit:** The recognition for having taken a course at school or university.
- **Cumulative GPA:** The average of all semester GPAs.
- **Elective Course:** Courses that are optional to take. Students can choose the lessons if they are interested in the topics.
- **Faculty Elective Course:** Subclass of Elective Course. Those courses' topics are related to the department.
- **GPA:** Abbreviation for “Grade Point Average”. GPA is a number representing the average value of the grades earned in courses over time.
- **Grade:** A class that finds letterGrade.
- **Graduation Project:** Courses that a student can enroll in after collecting at least 165 credits. Subclass of Mandatory Course.
- **ICreditRequirement:** An interface checks the required credits.
- **Instructor:** Lecturer of the courses.
- **Java:** Java is a class-based, object-oriented programming language that we use to develop that system.
- **Json File:** Json is a structure used to store data and move it between different platforms. JSON is text written in object notation format.
- **JUnit 5:** A unit testing framework.
- **Letter Grade:** A system used in schools to evaluate the performance of a student on a specific assignment or test.
- **Logging:** Is a method of recording events or actions that take place while a software program is running.
- **Log4j:** A Java-based logging utility.
- **Mandatory Course:** Courses that every student must take.
- **Non-Technical Elective Course:** Subclass of Elective Course. Those courses' topics are not much related to the department according to the Technical Elective Courses.

- **Person:** Superclass of student and advisor with name, surname, email, and phone number.
- **Prerequisite:** Course or other requirement that a student must have taken prior to enrolling in a specific course or program.
- **Quota:** The number or percentage of persons of a specified kind permitted to enroll in a course.
- **Registration:** The action or process of registering or of being registered.
- **Schedule:** A class that holds the day and course time of the courses taken by the student.
- **Semester:** A division constituting half of the regular academic year.
- **Student:** The main character of the course registration system under the Person class.
- **Student ID:** Unique ID that students have.
- **Transcript:** A class that calculates the GPA of the student and shows the courses taken, passed, and failed.
- **Unit Test:** A test that evaluates the individual components of a larger system.