Requirement Analysis Document

Marmara University - Faulty of Engineering

Department of Computer Engineering

CSE1242 Object Oriented Software Design (Autumn 2023)

Submit Date: 20/12/2023.

Course Registration System (RAD)

Student Number (ID)	Name	Surname
150120026	Ardacan	ÖZENER
150121921	Leen	I. A. SHAQALAIH
150121071	Ufuk	ACAR
150120018	Ömer Faruk	ATLIOĞLU
150120060	Ömer Faruk	GEYSOĞLU
150120998	Abdelrahman	Zahran
150121534	Furkan	Acar

Sections Of the Report: -

Important Notes:

- RAD for Iteration (3)

Section (1): Vision

Introduction

Welcome to the Course Registration System Project, an innovative and meticulously designed software solution aimed at streamlining the course registration process for both students and university staff. This project adheres to the highest standards of software design and development, ensuring a reliable, user-friendly, and efficient system that will enhance the overall academic experience within the university.

System Overview:

The Course Registration System is a comprehensive software application designed to facilitate the course registration process for students and empower university staff with the tools they need to manage course offerings seamlessly. This system will encompass a user-friendly web-based interface, ensuring that the registration process is straightforward and intuitive for all users.

Some of the key features of the system will include:

1. **User Authentication:** Robust user authentication and authorization mechanisms will be implemented to ensure data security and user privacy.
2. **Course Search and Selection:** Students will have access to an extensive catalog of courses, allowing them to search, filter, and select the courses that align with their academic goals.
3. **Real-Time Availability:** The system will provide real-time information on course availability, ensuring students can make informed decisions.
4. **Administrative Tools:** University staff will have access to powerful administrative tools for managing course offerings, class sizes, and accommodating special requests.

And more!

User Benefits:

The Course Registration System is expected to bring numerous benefits to both students and university staff:

For Students:

1. **Simplified Registration:**

The system will streamline the course registration process, making it more straightforward and less time-consuming for students.

2. **Real-Time Information:**

Students will have access to real-time course availability, ensuring they can make informed decisions about their academic journey.

3. **Conflict Avoidance:**

The system will automatically detect scheduling conflicts, reducing the likelihood of registration errors, and improving overall satisfaction.

For University Staff:

1. **Efficient Course Management:**

The administrative tools in the system will enable staff to manage course offerings, class sizes, and special requests with greater efficiency.

2. **Reduced Administrative Workload:**

Automation of various processes will lead to reduced administrative workloads and errors, allowing staff to focus on more critical tasks.

In conclusion, the Course Registration System is a promising project that aligns with the principles of modern software design and development. It is poised to revolutionize the course registration process, benefiting both students and university staff by enhancing efficiency, providing real-time information, and reducing administrative burdens. This system will serve as a valuable asset in the ever-evolving landscape of education, supporting the academic journeys of students and the operational excellence of educational institutions.

Our Vision for the Course Registration System is to create a digital ecosystem that fosters academic excellence by harnessing the power of technology. We aim to provide students with a user-friendly platform that empowers them to take control of their academic journey while offering university staff the tools they need to provide efficient support. Ultimately, we aspire to see a university environment where course registration is a seamless, stress-free process that supports the academic aspirations of every student, enabling them to make the most of their educational experience. This system will not only save time and reduce administrative burdens but also contribute to the university's reputation as a leader in modern education administration.

Thank you!

Section (2): Description of the Problem

The University Course Registration System is a comprehensive project aimed at streamlining and automating the process of course registration for students at a university. This system will initially utilize a Command Line Interface (CLI) to provide a user-friendly interface for students, administrators, and faculty members. The primary objective of the system is to enhance the efficiency and accuracy of the course registration process, reducing the manual workload on both students and university staff.

For students, the CLI will offer a straightforward way to browse available courses, view course details, and register for classes. The system will also incorporate features such as checking course prerequisites and displaying real-time seat availability. To ensure a smooth registration experience, the CLI will provide clear prompts and instructions.

Administrators and faculty members will benefit from functionalities that allow them to manage course offerings, monitor enrollment statistics, and make adjustments to the course catalog. The system will maintain a centralized database to store and update course information, student registrations, and other relevant data. This database will enable efficient data retrieval and reporting, supporting decision-making processes within the university.

**Future iterations of the project may include the development of a graphical user interface (GUI) and integration with other university systems for a more seamless experience. Overall, the University Course Registration System aspires to be a robust, user-friendly tool that enhances the efficiency of course management and registration processes within the university setting. **

Section (3): Glossary

A

Academic Advisor: A faculty member responsible for guiding students in their academic pursuits, including course selection and degree requirements.

- **Aliases**: Counselor, Academic Counselor, Guidance Counselor
- **Description**: Academic advisors play a crucial role in helping students navigate their academic path, ensuring they fulfill their degree requirements and make the most of their educational experience.
- **Format (Type, Length, Unit)**: N/A
- **Relationships to Other Elements**: Academic advisors are closely related to students, as they offer personalized guidance to individual students. They may also have a connection with the university's degree program and course catalog.
- **Range of Values**: N/A
- **Validation Rules**: N/A

Advisor ID: A unique identification number or code assigned to academic advisors. This identifier is used for administrative and record-keeping purposes, allowing the university to associate students with their respective academic advisors**.**

- **Aliases**: Academic Advisor Identifier, Advisor Code
- **Description**: Advisor IDs streamline administrative processes, helping the university track student-advisor relationships and maintain accurate advising records.
- **Format (Type, Length, Unit)**: Alphanumeric (e.g., ADV123, A4567)
- **Relationships to Other Elements**: Advisor IDs are closely associated with academic advisors, students, and academic advising records.
- **Range of Values**: Each advisor is assigned a unique ID.
- **Validation Rules**: Advisor IDs should be unique and follow specific formatting conventions established by the university. Additionally, they may be linked to specific academic departments or advising units.

C

Course Catalog: A comprehensive list of all available courses, including their descriptions, prerequisites, and credit hours.

- **Aliases**: Course Schedule, Course Listing, Course Offerings
- **Description**: The course catalog serves as a valuable resource for students, faculty, and administrators, enabling them to understand the curriculum, plan course selections, and ensure academic progress.
- **Format (Type, Length, Unit)**: Document (Digital or Print)
- **Relationships to Other Elements**: The course catalog is linked to specific courses, their descriptions, and academic departments.
- **Range of Values**: N/A
- **Validation Rules**: N/A

Course Registration: The process by which students enroll in courses for an upcoming academic term.

- **Aliases**: Enrollment, Class Registration
- **Description**: Course registration is a critical activity for students, as it determines their course load and progress toward degree completion. It is also essential for the university to manage class sizes and allocate resources.
- **Format (Type, Length, Unit)**: Process
- **Relationships to Other Elements**: Course registration is directly related to students, course sections, and the registrar's office.
- **Range of Values**: N/A
- **Validation Rules**: N/A

Course Section: A specific instance of a course offered in a particular term, indicating the time, location, and instructor.

- **Aliases**: Class Section, Course Offering
- **Description**: Course sections provide the details needed for students to select specific times and instructors for courses that match their schedules and preferences.
- **Format (Type, Length, Unit)**: Structured Data
- **Relationships to Other Elements**: Course sections are linked to the corresponding course, instructor, and classroom.
- **Range of Values**: Information about available seats and scheduling details.
- **Validation Rules**: Seat availability must be within the specified capacity of the assigned classroom.

D

Drop/Add Period: A designated timeframe during which students can make changes to their course schedule, such as dropping or adding courses.

- **Aliases**: Add/Drop Period, Course Adjustment Period
- **Description**: The drop/add period allows students to fine-tune their course selections, accommodating changes in their preferences, availability, or academic goals.
- **Format (Type, Length, Unit)**: Timeframe
- **Relationships to Other Elements**: Directly linked to course registration and student schedules.
- **Range of Values**: Start and end dates for the drop/add period.
- **Validation Rules**: N/A

F

Full-Time Student: A student who is enrolled in a sufficient number of credit hours to be considered full-time, typically 12 or more credit hours per term.

- **Aliases**: FT Student, Full-Time Enrollment
- **Description**: Determining full-time or part-time status affects tuition costs, eligibility for scholarships, and access to university resources.
- **Format (Type, Length, Unit)**: Numeric
- **Relationships to Other Elements**: Full-time student status affects financial aid eligibility, student services, and the calculation of tuition fees.
- **Range of Values**: Typically, 12 or more credit hours.
- **Validation Rules**: A student's course load should meet or exceed the institution's definition of full-time enrollment.

Faculty: The academic staff responsible for teaching courses and conducting research.

- **Aliases**: Instructors, Educators
- **Description**: Faculty members are at the heart of the educational process, delivering lectures, grading assignments, and contributing to the university's research endeavors.
- **Format (Type, Length, Unit)**: N/A
- **Relationships to Other Elements**: Faculty members are linked to specific courses they teach, academic departments, and students in the context of advising.
- **Range of Values**: N/A
- **Validation Rules**: N/A

G

Grade Point Average (GPA): A numerical representation of a student's academic performance, calculated by averaging the grades received in all completed courses.

- **Aliases**: GPA Score, Academic Average
- **Description**: The GPA serves as a key indicator of a student's overall academic achievement and is used for various purposes, including eligibility for honors, scholarships, and graduate programs.
- **Format (Type, Length, Unit)**: Numeric (e.g., 4.0 scale)
- **Relationships to Other Elements**: GPA is related to individual courses, the student's transcript, and academic standing.
- **Range of Values**: Typically, GPA scores range from 0.0 to 4.0 or equivalent scales.
- **Validation Rules**: GPA calculation rules may include weighted GPA for honors courses or pass/fail grading policies.

Graduation Requirements: The set of criteria that students must fulfill to obtain their degree, including completing specific courses and achieving a minimum GPA.

- **Aliases**: Degree Requirements, Graduation Criteria

- **Description**: Graduation requirements are a roadmap that students follow to ensure they meet the academic standards necessary for degree conferral.
- **Format (Type, Length, Unit)**: N/A
- **Relationships to Other Elements**: Graduation requirements are directly tied to degree programs and the student's academic record.
- **Range of Values**: Varies depending on degree program and university policies.
- **Validation Rules**: Meeting all specific requirements, such as minimum GPA and course credits, is mandatory for degree conferral.

I

Instructor: The person responsible for teaching a specific course section.

- **Aliases**: Teacher, Professor, Educator
- **Description**: Instructors are vital to the learning process, guiding students through the course content, providing feedback, and fostering an engaging learning environment.
- **Format (Type, Length, Unit)**: N/A
- **Relationships to Other Elements**: Instructors are directly associated with specific course sections and academic departments.
- **Range of Values**: N/A
- **Validation Rules**: N/A

Internship: A practical work experience related to a student's field of study, often undertaken for academic credit.

- **Aliases**: Work Placement, Cooperative Education
- **Description**: Internships provide students with valuable insights into their chosen profession, helping them build practical skills and establish connections in the industry.
- **Format (Type, Length, Unit)**: N/A
- **Relationships to Other Elements**: Internships are linked to degree programs, courses, and academic departments offering experiential learning opportunities.
- **Range of Values**: Varies by program and university policy.
- **Validation Rules**: N/A

L

Lecture: A type of course instruction that typically involves a one-way delivery of content from the instructor to the students.

- **Aliases**: Lecture Session, Classroom Presentation

- **Description**: Lectures are a fundamental component of many courses, providing students with foundational information and concepts.
- **Format (Type, Length, Unit)**: Teaching Method
- **Relationships to Other Elements**: Lectures are associated with course sections and academic content.
- **Range of Values**: N/A
- **Validation Rules**: N/A

Lab: A type of course instruction that involves hands-on learning and practical exercises in a controlled environment.

- **Aliases**: Laboratory Session, Practical Class
- **Description**: Labs are critical for courses that require experimentation, data collection, and the development of practical skills.
- **Format (Type, Length, Unit)**: Teaching Method
- **Relationships to Other Elements**: Labs are linked to course sections, equipment, and academic departments.
- **Range of Values**: N/A
- **Validation Rules**: Labs often have limited capacity due to available equipment or safety considerations.

M

Major: The primary area of study a student chooses to pursue within their degree program.

- **Aliases**: Major Field, Major Concentration
- **Description**: The major is a significant part of a student's academic journey, shaping their expertise and career prospects.
- **Format (Type, Length, Unit)**: Text, Variable (e.g., Computer Science, Psychology)
- **Relationships to Other Elements**: Majors are linked to degree programs, courses, and graduation requirements.
- **Range of Values**: Varied based on the university's offerings.
- **Validation Rules**: Meeting specific course and credit requirements associated with the chosen major.

P

Prerequisite: A course or requirement that must be completed before a student can enroll in a more advanced course.

- **Aliases**: Precondition, Entry Requirement
- **Description**: Prerequisites serve as a structured sequence of learning, guiding students through a curriculum in a logical and progressive manner.
- **Format (Type, Length, Unit)**: Course Title or Requirement

- **Relationships to Other Elements**: Prerequisites are directly related to the courses they precede and are crucial for academic planning.
- **Range of Values**: Specific course titles or requirements.
- **Validation Rules**: Students must successfully complete all specified prerequisites to enroll in advanced courses.

R

Room Capacity (Quota): The maximum number of students that a specific classroom or lecture hall can accommodate.

- **Aliases**: Seating Capacity, Maximum Occupancy
- **Description**: Room capacity information helps determine the suitability of a location for hosting classes, examinations, and events.
- **Format (Type, Length, Unit)**: Numeric (e.g., 50 students, 150 seats)
- **Relationships to Other Elements**: Room capacity is associated with specific course sections and scheduling.
- **Range of Values**: Varies by room or venue and may depend on factors such as seating arrangement.
- **Validation Rules**: Course sections should not exceed the specified room capacity to ensure safety and comfort.

S

Semester: A half of an academic year, typically divided into fall and spring terms.

- **Aliases**: Term, Academic Semester
- **Description**: Semesters provide a standardized framework for organizing coursework and assessing academic progress.
- **Format (Type, Length, Unit)**: Timeframe
- **Relationships to Other Elements**: Semesters are closely tied to course scheduling, registration periods, and academic calendars.
- **Range of Values**: Typically two semesters per academic year, but may vary based on the university's academic calendar.
- **Validation Rules**: Courses and registrations should align with the designated semester.

Student ID: A unique identification number assigned to each student for administrative and record-keeping purposes.

- **Aliases**: ID Number, Student Identification
- **Description**: Student IDs streamline administrative processes, including registration, record-keeping, and authentication of student identities.
- **Format (Type, Length, Unit)**: Alphanumeric or Numeric (e.g., ABC12345, 123456)

- **Relationships to Other Elements**: Student IDs are associated with individual students and their academic records.
- **Range of Values**: Unique for each student.
- **Validation Rules**: Student IDs should be unique, and their format may follow specific patterns or standards.

Syllabus: A document outlining the course objectives, content, reading materials, assignments, and assessment methods.

- **Aliases**: Course Outline, Class Syllabus
- **Description**: The syllabus provides students with essential information about what to expect in the course, helping them prepare and manage their time effectively.
- **Format (Type, Length, Unit)**: Document (Digital or Print)
- **Relationships to Other Elements**: Syllabi are directly linked to course sections and instructional content.
- **Range of Values**: Content can vary by course and instructor, but typically includes course objectives, reading lists, assessment methods, and grading scales.
- **Validation Rules**: N/A

T

Transcript: An official record of a student's academic performance, including courses taken, grades earned, and degree(s) conferred.

- **Aliases**: Academic Record, Educational Transcript
- **Description**: Transcripts are vital for verifying a student's academic history, and they are often required for graduate school applications, job opportunities, and professional licensing.
- **Format (Type, Length, Unit)**: Document (Digital or Print)
- **Relationships to Other Elements**: Transcripts are directly linked to students, courses, and graduation records.
- **Range of Values**: Various data points, including course titles, grades, credits, and degree information.
- **Validation Rules**: Transcripts should accurately reflect a student's academic record and comply with university policies.

Term: An academic period within the university year, which can be a semester, quarter, or other specified time frame.

- **Aliases**: Academic Term, School Year Segment
- **Description**: Terms provide a structured timeline for the organization of courses, student registration, and administrative processes.
- **Format (Type, Length, Unit)**: Timeframe
- **Relationships to Other Elements**: Terms are associated with course scheduling, registration periods, and academic calendars.
- **Range of Values**: The university defines the number and structure of terms in an academic year.

- **Validation Rules**: Courses and academic activities should align with the designated term.

Data Dictionary Attributes

- **Aliases**: Alternative names or terms that may be used to refer to the same concept.
- **Description**: Additional information or context about the term to provide a deeper understanding.
- **Format (Type, Length, Unit)**: The data type, character length, and measurement unit if applicable, for data associated with the term.
- **Relationships to Other Elements**: How the term is related to other terms or entities within the system, such as the relationship between a student and their course registration.
- **Range of Values**: The allowable values or acceptable range for data associated with the term, for example, the range of possible credit hours.
- **Validation Rules**: Specific rules and criteria that data associated with the term must meet to be considered valid, which can include format, range, or other constraints.

These terms and attributes collectively serve as the foundation for developing and maintaining the university course registration system, guiding the system's behavior, and ensuring that it meets the requirements of both students and administrative staff.

Section(3): Functional & Non-Functional Requirements

Functional Requirements:

1. **Transcript Processing:**
    - The system must be able to read student transcripts from JSON files (e.g., "150119055.json").
    - It should update these transcript files at the end of each run to reflect the latest information on student registrations and academic progress accurately.
2. **System Configuration:**
    - The system must include a "parameters.json" file to read and configure system parameters.
    - This file allows for easy customization and adaptation of the system without the need to modify the source code.
3. **User Authentication:**
    - The system must provide a secure user authentication mechanism.
    - Users should be able to enter valid credentials (name and password) registered in the system for secure access.
4. **Course Prerequisites:**
    - The system must calculate and verify course prerequisites.

- It should determine whether a student meets the necessary requirements to register for a specific course, ensuring they have the required background for successful enrollment.

5. **Advisory Approval:**
    - The system must enable academic advisors to either approve or deny a student's request to enroll in a particular course.
    - This feature ensures that advisors have control over students' course selections based on academic progress and advisability.
6. **Week Schedule:**
    - The system should provide a feature for students to view and manage their weekly schedule.
    - Users should be able to add, remove, or modify courses within their schedule, and the system should reflect these changes accurately.
7. **Notifications:**
    - The system must have a notification mechanism to alert users about important events, such as course approvals, registration status, and other relevant updates.
    - **Notifications can be sent through email, in-app messages, or other communication channels, ensuring users stay informed about their academic status. **

Non-Functional Requirements:

1. **Clarity in Outputs and Logs:**
    - All outputs, system logs, and variable names must be clear, understandable, and use easy-to-comprehend names for both users and developers. This enhances the system's usability and maintainability.
2. **Bug-Free Integration:**
    - The system must seamlessly integrate new courses, advisors, and students without introducing any bugs. This ensures that the system remains robust and reliable as it evolves with the addition of new data.
3. **Error-Free Source Code:**
    - The source code must be thoroughly tested, and it should not include any bugs. This ensures the reliability and stability of the system, minimizing the risk of errors during operation.
4. **JSON Data Storage:**
    - All data, including student transcripts, system parameters, and other relevant information, must be stored in JSON files. This choice of data storage format ensures consistency and ease of data retrieval.
5. **Code Readability:**
    - The source code must be easy to understand, facilitating maintenance and updates by developers. Clear and well-documented code enhances collaboration and reduces the learning curve for those working on the system.
6. **Error Logging:**

- All possible errors encountered during system operation must be logged to a file. This aids in troubleshooting and debugging, ensuring that issues are identified and addressed promptly for a smooth user experience.

7. **Additional Feature (Elective NTE, FTE, TE):**

    - The system must support the addition and management of elective courses categorized as Non-Technical Electives (NTE), Free Technical Electives (FTE), and Technical Electives (TE). This enhancement broadens the scope of course offerings and academic flexibility for students.

Section(5): Domain Model

The domain model for the University Course Registration System is represented by several key entities and their relationships:

1. **Student:**

Represents an individual enrolled in the university. A student has personal information, a unique student ID, and an associated transcript that tracks their academic history.

2. **Advisor:**

Represents an academic advisor who guides and advises students in their course selections. Advisors have the authority to approve or deny course registration requests from students.

3. **Course:**

Represents an academic course offered by the university. Each course has a unique identifier, a name, and information about its prerequisites, credit hours, and other relevant details.

4. **Course Section:**

Represents a specific instance or section of a course offered during a particular semester or term. Each course section is associated with a classroom and has information about its schedule, capacity, and enrolled students.

5. **Classroom:**

Represents a physical or virtual space where course sections are conducted. It includes information such as the room number, capacity, and any additional resources available.

6. **Transcript:**

Represents an academic record for a student, documenting their course enrollments, grades, and overall academic performance. The transcript is associated with a specific student and is updated as the student progresses through their academic journey.

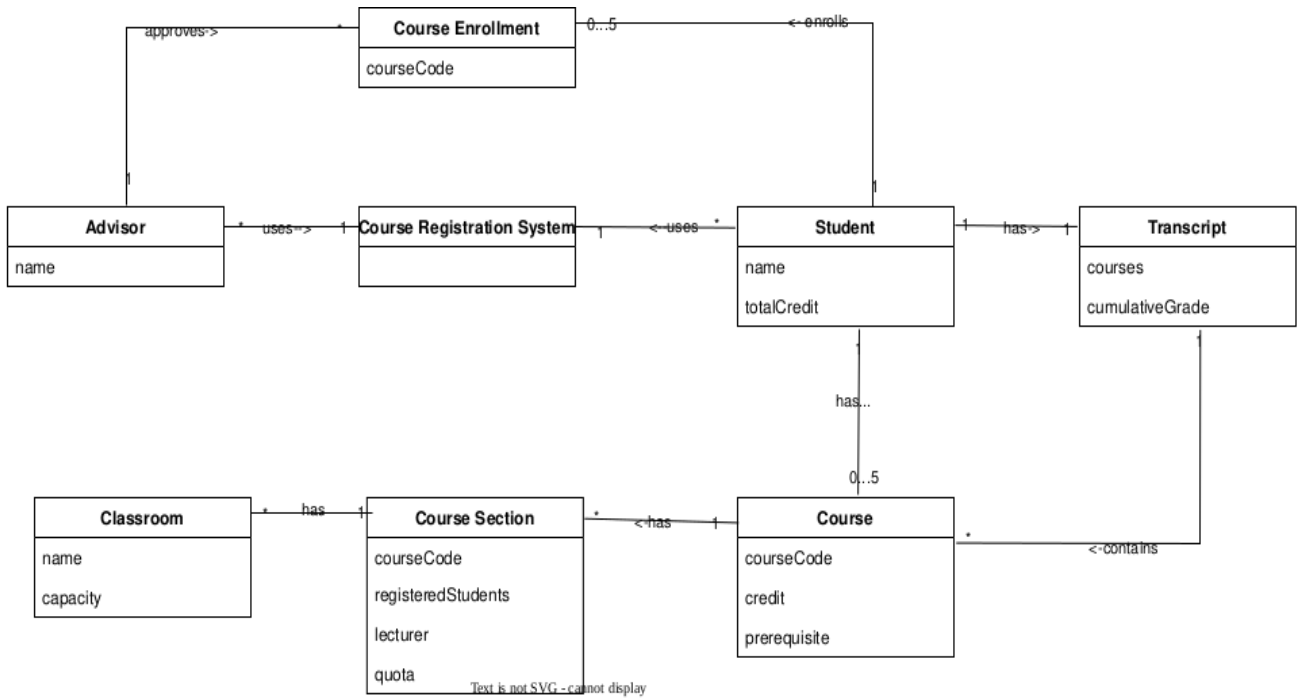
7. **Course Enrollment:**

Represents the act of a student registering for a particular course section. It includes information about the student, the course section, and any approval status from the advisor.

8. **Course Registration System:**

Represents the overall system responsible for managing the course registration process. It includes functionalities for reading transcripts, processing course enrollments, managing advisors, and handling system configurations.

This domain model captures the key entities and their relationships within the University Course Registration System. Students enroll in courses, guided by their advisors, and the system manages the registration process, course sections, classrooms, and academic records through transcripts.



Section (6): Use Cases (1 & 2)

Use Case #1

Description

This usage scenario explains how a student enrolls in courses through the system and how the process progresses.

Stakeholders and Interests

1. **Student**: The student wants to register for selected courses, check the suitability, prerequisites and Quota of the courses, view their course schedule, and access an updated transcript.
2. **Course Registration System**: The system aims to securely verify student identities and accurately process course registrations and updates.
3. **Academic Advisors**: Advisors play a role in approving or guiding students in their course selections. They want to assist students in making appropriate course choices and review and approve course registration requests.

Actors

Primary Actor: Student

Supporting Actors: Course Registration System, Database (JSON files)

Preconditions

- The student must be a registered and active student in the system.
- The student needs to log in with their username and password.
- The course registration period must be open.
- The student must meet the registration requirements, such as prerequisites and credit limits.

Success Guarantee (Postconditions)

- The student's course registration is successfully completed, and the student can access their course schedule, the list of registered courses, and an updated transcript.

Basic Flow

1. System Login:

1.1. The student logs into the system using their username and password.

1.2. The system verifies the student's identity and redirects them to the main page upon successful login.

2. View Course List:

2.1. The student selects the "Course Registration" module.

2.2. The system displays a list of courses available for registration for the current term.

3. Course Selection:

3.1. The student selects the course(s) they wish to register for from the list.

3.2. The system checks the prerequisites of the selected course(s) and the student's previous course records.

4. Conflict Check and Approval:

4.1. The system checks if there are any time conflicts between the selected courses.

4.2. If there are no conflicts or prerequisites issues, the system presents the option to confirm the selection to the student.

5. Registration Process:

5.1. The student confirms their selection.

5.2. The system records the student's registration and updates the registration in the JSON database.

6. Registration Result and Transcript Update:

6.1. The system sends a successful registration message to the student.

6.2. The student's transcript and course schedule are updated, and this information is made available to the student.

Alternative Flows

1.1.a) The student uses unwanted special characters in the username.

1.1.a.1) The system detects unwanted special characters in the username and informs the student with an error message.

1.1.a.2) The student realizes that a valid username is required and retries with a corrected username.

1.1.a.3) The system accepts the corrected username and proceeds to the password request step.

1.2.a) If the system detects an incorrect username or password, it displays an error message and redirects to the login screen.

1.2.a.1) The system identifies an incorrect username or password combination and informs the student with an error message.

1.2.a.2) The student understands that they need to enter the correct username and password and retries with corrected information.

1.2.a.3) The system accepts the correct username and password and proceeds to the course list viewing step.

2.2.a) If the courses for the term have not opened yet or the course list is empty, it displays an error message.

2.2.a.1) The system detects that the term has not started or the course list is empty and informs the student with an error message.

2.2.a.2) The student understands that they should check again later and is redirected to the main page.

3.2.a) If one or more of the selected courses do not meet prerequisites, it displays an error message.

3.2.a.1) The system identifies that one or more of the selected courses do not meet prerequisites and informs the student with an error message.

3.2.a.2) The student understands that they should choose appropriate courses and returns to the course selection step.

4.2.a) If there is a time conflict among the selected courses, the system displays conflicting courses and provides an opportunity for the student to resolve the conflict.

4.2.a.1) The system detects a time conflict among the selected courses and displays this conflict.

4.2.a.2) The student makes changes to the conflicting courses or selects a different time.

4.2.a.3) The system confirms the conflict-free course selection and proceeds to the approval step.

5.1.a) If the student chooses to cancel their confirmation before completing the registration, the registration process is cancelled.

5.1.a.1) The student chooses to cancel their confirmation before finalizing the registration.

5.1.a.2) The system informs the student that the registration process has been canceled and redirects them to the course selection step.

5.2.a) If an error occurs during the registration process, the system sends an error message and suggests repeating the registration process.

5.2.a.1) The system detects an error during the registration process and informs the student with an error message.

5.2.a.2) The student corrects the error and chooses to repeat the registration process.

5.2.a.3) The system accepts the corrected registration process and proceeds to the registration result and transcript update step.

Special Requirements

• The system should efficiently and securely handle user login and course registration processes.

Frequency

• This process occurs at the beginning of each academic term and during specified registration periods.

Use Case #2

Description:

This use case explains the process of an advisor who approves registrations for courses for each student through the system and outlines how this process will operate.

Stakeholders and Interests:

1. **Student**: The student wants to register for selected courses, check the suitability and prerequisites of the courses, view their course schedule, and access an updated transcript.
2. **Course Registration System**: The system aims to securely verify student identities and accurately process course registrations and updates.
3. **Academic Advisors**: Advisors play a role in approving or guiding students in their course selections. They want to assist students in making appropriate course choices and review and approve course registration requests.

Actors:

- Primary Actor: Advisor

- Supporting Actors: Course Registration System, Database (JSON files)

Preconditions:

- The advisor must be registered and an active advisor within the system.

- The advisor must be able to log in to the system with a username and password.

- The course registration period must be open for approval.

- The advisor must check the necessary conditions for registration for each student's registration request.

Postconditions:

- The advisor’s course registration approvals are successfully completed, and the transcript has updated.

Basic Flow:

1. System Login:

1.1. The advisor logs into the system by using their username and password.

1.2. The system authenticates the advisor’s identity and redirects to the homepage if the login is successful.

2. Viewing Student List and Student Selection:

2.1. The advisor selects the “Student List" module and selects student(s) from the student list.

2.2. The system displays a list of students which the advisor is responsible for the current term.

3. Checking and Confirmation:

3.1. The system checks for any time conflicts between selected courses and prerequisite information against the student’s academic history for the selected student.

3.2. If no conflicts or prerequisite issues are detected, the system is going to prompt an approve button to the advisor's screen to confirm the student's each course selection.

4. Registration Approval Process:

4.1. The advisor approves or denies each student’s course registration requests.

4.2. The system processes the registration and updates the records in the JSON database.

5. Registration Outcome and Transcript Update:

5.1. The system sends a successful approval message to the advisor.

5.2. The advisor updates the student's transcript.

Alternative Flows:

1.1.a) The advisor uses unwanted special characters in the username.

1.1.a.1) The system detects unwanted special characters in the username and informs the advisor with an error message.

1.1.a.2) The advisor realizes that a valid username is required and retries with a corrected username.

1.1.a.3) The system accepts the corrected username and proceeds to the password request step.

1.2.a) If the system detects an incorrect username or password, it displays an error message and redirects to the login screen.

1.2.a.1) The system identifies an incorrect username or password combination and informs the advisor with an error message.

1.2.a.2) The advisor understands that they need to enter the correct username and password and retries with corrected information.

1.2.a.3) The system accepts the correct username and password and proceeds to the course list viewing step.

2.2.a) If the courses for the term have not opened yet or the student list is empty, it displays an error message:

2.1.a.1) The system detects that the term has not started or the student list is empty and informs the advisor with an error message.

2.1.a.2) The advisor understands that they should check the student list again later on.

3.2.a) If one or more of the selected courses do not match prerequisites or there is a time conflict, it displays an error message.

3.2.a.1) The system identifies that one or more registration requests do not match with the prerequisites and informs the advisor with an error message.

3.2.a.2) The advisor understands that students' requests got denied by the system and returns to the student list.

4.2.a) If an error occurs during the approval process, the system displays an error message.

4.2.a.1) The system detects an error during the approval process and informs the advisor with an error message.

4.2.a.2) The advisor understands what causes an error message and denies the requests.

4.2.a.3) The system accepts the advisor’s feedback and updates the transcript.

Special Requirements:

- The system must perform securely and swiftly during user login and course registration processes.

Frequency:

- This process will occur at the beginning of each academic term and during designated registration periods.

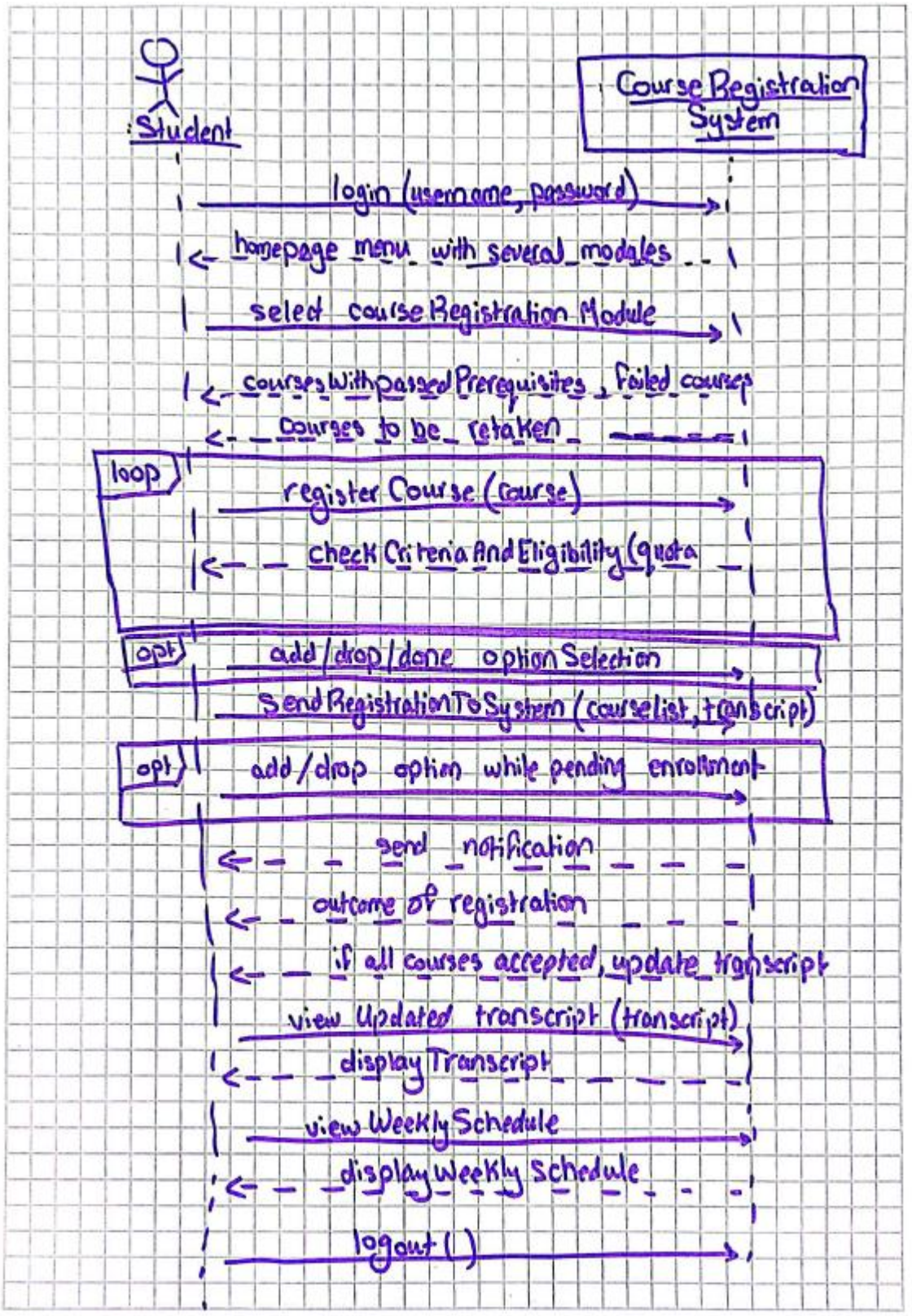
Section (7): System Sequence Diagram

- Student System Sequence Diagram (SSD)

A System Sequence Diagram (SSD) provides a visual representation of the interactions between external actors and a system in response to specific actions or events. In the context of a university course registration system, let's consider a simple SSD for the scenario of a student registering for a course.

1. **Actor: Student**
    - The student initiates the registration process by sending a "Register" message to the system.
2. **System Response: Course Registration System**
    - The Course Registration System receives the "Register" message and processes it.
3. **System Behavior:**
    - The system checks the student's eligibility, prerequisites, and course availability.
    - If the conditions are met, the system processes the registration and updates the student's record.
4. **System Response: Confirmation Message**
    - The system sends a "Registration Successful" message to the student.

This sequence diagram illustrates a basic interaction between the student and the course registration system. It captures the flow of messages between the external actor and the system, highlighting the key steps involved in the course registration process.

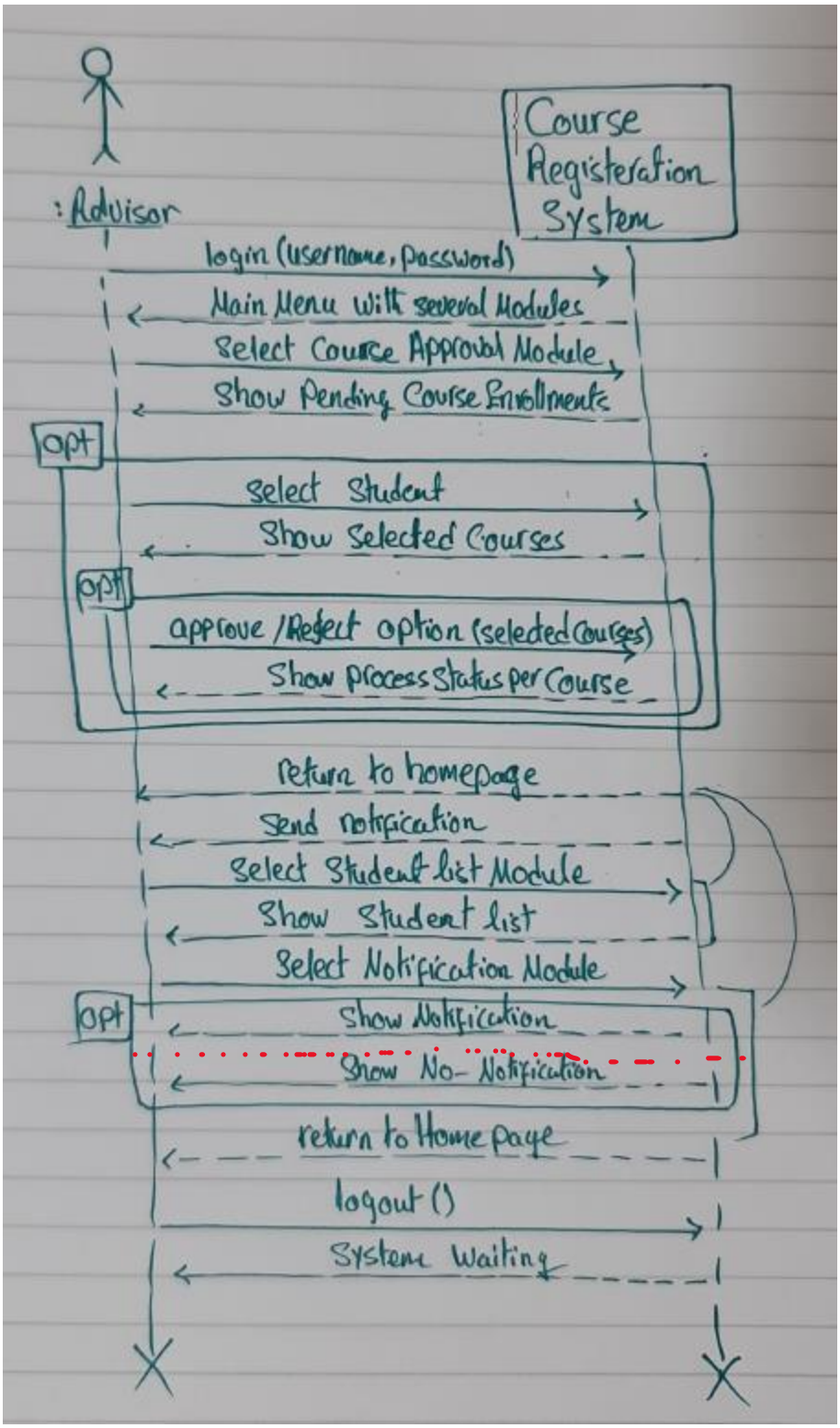

Student
Course Registration System
login (username, password)
homepage menu with several modules
select course Registration Module
courses with passed Prerequisites, failed courses
Courses to be retaken
loop
register Course (course)
check Criteria And Eligibility (quota
opt
add/drop/done option Selection
Send Registration To System (courselist, transcript)
opt
add/drop option while pending enrollment
send notification
outcome of registration
if all courses accepted, update transcript
view Updated transcript (transcript)
display Transcript
view Weekly Schedule
display Weekly Schedule
logout ()

- Advisor System Sequence Diagram

A System Sequence Diagram (SSD) for an advisor approving a student's course registration in a university registration system is described below:

1. **Actor: Advisor**
    - The advisor initiates the approval process by sending an "Approve Registration" message to the system.
2. **System Response: Course Registration System**
    - The Course Registration System receives the "Approve Registration" message and processes it.
3. **System Behavior:**
    - The system retrieves the list of students awaiting approval and presents it to the advisor.
    - The advisor selects a student from the list for registration approval.
4. **System Response: Confirmation Message**
    - If the student's registration is approved, the system sends a "Registration Approved" message to the advisor.
5. **System Response: Denial Message**
    - If the student's registration is denied (due to conflicts, prerequisites, etc.), the system sends a "Registration Denied" message to the advisor.

This SSD outlines the interaction between the advisor and the course registration system during the approval process. It focuses on the key messages exchanged and the system's responses as the advisor reviews and approves/denies student registrations.


Course Registeration System
:Advisor
login (username, password)
Main Menu with several Modules
Select Course Approval Module
Show Pending Course Enrollments
opt
Select Student
Show Selected Courses
opt
approve / Reject option (selected courses)
Show Process Status per Course
return to homepage
Send notification
Select Student list Module
Show Student list
Select Notification Module
opt
Show Notification
Show No- Notification
return to Home page
logout ()
System Waiting

Workflow Report

Teamwork in Iteration (1):

Each team member handled a particular part of the Requirement Analysis Design, but we helped one another by brainstorming all possible ideas. The diversity of perspectives not only fueled creativity but also laid the groundwork for a robust project framework:

- Abdelrahman wrote the vision and glossary.

-Ardacan mentioned the functional and non-functional requirements.

-Ömer Geysoğlu described the student's use case

-Ufuk described the advisor's use case

-Furkan sketched the domain model

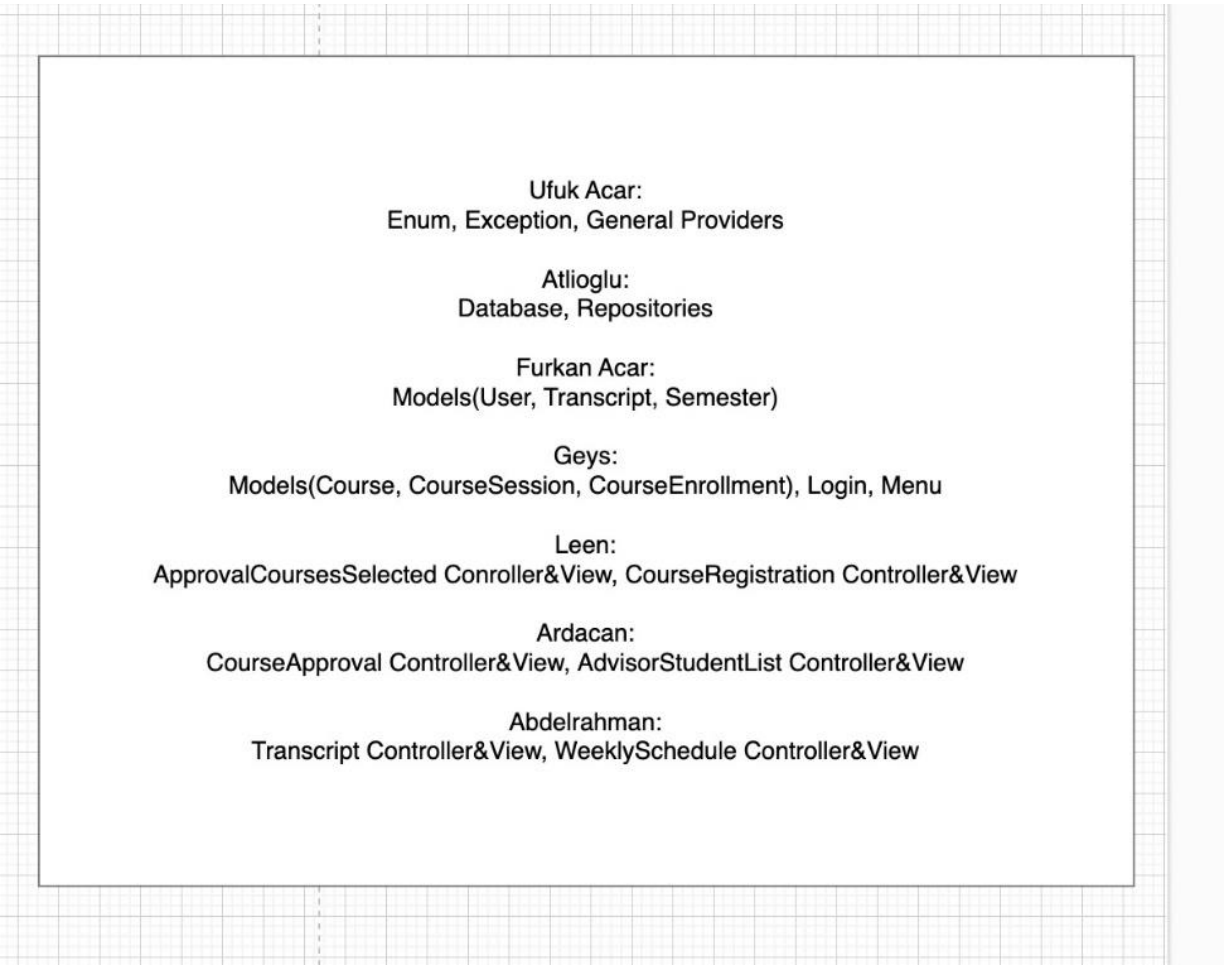
-Leen drew the student's system sequence diagram

-Ömer Atlıoğlu drew the advisor's system sequence diagram

In the Design Phase:

We divided the group into 2, Furkan, Ömer Atlıoğlu, Ömer Geysoğlu, and Ufuk worked on the DCD; while Ardacan, Leen, and Abdelrahman worked on the DSD. Both teams opened collaborative workspace on draw.io where we were able to work together to manage the diagrams. We were facing some difficulties and confusion because we made a single DSD for the whole project. After the demo, we recognized our misunderstanding of the DSD, and hopefully, we were able to fix it in the second iteration. We did several online and face-to-face meetings to discuss what we've done so far.

In Coding Phase:



Teamwork in Iteration (2):

In this iteration, everyone made some modifications to their parts in the Requirement Analysis Design. We added more functionality to our course registration system.

In the Design Phase: We had a meeting where we discussed which features we should add to this iteration. Since we got a lot of ideas, we filtered out which of these features should be added to this iteration. Upon dividing the parts among us, everyone started thinking about how they can modify their DCDs. For the DSD, since we misunderstood it in the first iteration, we divided the DSDs to:

- User Layer (handled by Leen)
- Domain Model:

Course Registration (handled by Ufuk)

Transcript and Weekly Schedule (handled by Abdelrahman)

Approval courses selected and Course approval (handled by Arda)

Notification (handled by Ömer Atlıoğlu)

In the coding part:

Leen: In this iteration, I handled several features of the course registration controller. I first made the course registration more flexible by allowing the students to choose as many numbers of courses as they want. I also handled the situation where the advisor accepts some courses of the enrollment while rejecting the others. In this case, the student can either finalize their course enrollment with the accepted courses, or he/she can add more courses and send them to the advisor to check. I considered the courses that the student has failed, got DD, or DC to also be shown to the student to select. There were some bugs in this part, so I worked on fixing them. Since we need to make our system run without breaking apart, I revised the try/catch.

Ufuk: In this iteration, I implemented a series of control mechanisms to facilitate students' registration for courses and add-drop processes. These controls aimed to guide students in selecting courses by regularly checking the specified quota and real-time availability of courses.

Firstly, during the course registration process, I implemented a quota check for each selected course through a method called checkQuota. If the quota for a chosen course is full, an error message is displayed to the student, prompting them to consider alternative course options. This allows students to successfully identify courses with full quotas during their course selections and update their choices with suitable alternatives.

A similar check was performed during add-drop processes. At each step of adding or removing courses, students used the checkQuota method to verify course quotas. This provided guidance to students in adding or dropping courses based on the current quota status.

When course selections were submitted for advisor approval, I updated the currentQuota value based on whether the advisor approved the course selections or not. If the advisor approved the course selections, I increased the quotas for the respective courses; otherwise, I made no changes. This allowed students to make their course selections based on the updated quotas after they were approved by the advisor.

These controls and updates increased sensitivity to quota issues during the course selection processes, enabling students to manage course registration and add-drop processes more efficiently.

Ardacan: Logger object's purpose is log the user's choice by writing it to a file. File's content consists lines in format of ({user's choice} is user's choice in method {methodName} in class {className}{\n}) and for exceptions (Exception {Exception type}).

Attributes:

fileName:Path => is used for log file's path. Path type is in java.nio.file.Path. In DCD showed as String not to confuse you.

className:String => stores the class name who called the caller of Logger.

methodName:String => stores the method name who called the caller of Logger.

Methods:

Logger(): => constructor with no parameters. Sets fileName to ./Iteration2/code/CSE3063GRP12/database/log/log.txt and both methodName and className to null. Calls AppConstant.getInstance().getBasePath() for default path.

Logger(String): => constructor with parameters. Sets fileName to input and both methodName and className to null

write(string):void => APPENDS to log file in fileName. Format specified above. Does not call any other methods written in project. Only calls methods in built in Java libraries.

writeException(string):void => APPENDS to log file in fileName. Format specified above. Does not call any other methods written in project. Only calls methods in built in Java libraries.

Furkan: Conducted and implemented unit tests.

Repository updates: Methods for Transcript Handling and Student Handling were altered.

Changes were made in Course Registration handling.

Provided necessary feedback by testing the Pull Requests opened by our team members.

Ömer Atlıoğlu: A notification feature operates when a Student performs a course approval or when an Advisor approves or rejects courses sent by the Student. If a course approval action is initiated by a Student in the CourseRegistrationController class, this notification will be added to the Advisor's notification JSON file.

In the ApprovalCoursesSelectedController class, when an Advisor approves, rejects, or partially approves a course, this notification will be added to the Student's notification JSON file. The NotificationResponse class holds a Notification ArrayList and the containsNew property, indicating the presence of new notifications.

The NotificationController and View display notifications to the user through the menu. The 'read' field within the Notification class provides information on whether the notification has been previously read. If there's a new notification, the text in the menu appears bold, and the new notification is displayed in bold as well.

Abdelrahman:

Creating 65 transcript.json

Creating 7 advisors.json

Improving transcript controller and viewer functionalities

Improving weekly schedule controller and viewer

Improving error handling and exception handling regarding the previous classes

Better views for registration view and student list at advisor side

Ömer Geysoğlu: In the improvements I made, I transitioned from a simple system where academic advisors could only approve or reject selected courses in bulk to a more realistic structure where they can make individual decisions for each course. In this context, I added approvedCourses and rejectedCourses fields to the Course class in the MVC model to keep records of approved and rejected courses. Additionally, I implemented some bug fixes in login and menu controls to enhance user experience and system stability. These enhancements ensured that the system is user-friendly and functional.

Teamwork in Iteration (3):

- Update all artifacts.
- Convert Codes from java to python.

- All Team members contributed to this phase.

Thank You!