Requirement Analysis Document

Student Lecture Registration System

	NAME	SURNAME	NUMBER
1.	Kaan	CAMCI	150119063
2.	Kadir Berk	YAĞAR	150120016
3.	Sena	VATANSEVER	150119755
4.	Fatih	ÖZTÜRK	150119519
5.	Samet	CAN	150120528
6.	Serdar	ALSAN	150120034
~~**7.**~~	~~Erman~~	~~KUNDAKÇIOĞLU~~	~~150122603~~

~~Deleted parts~~

Added parts

1.) Vision

~~The goal is to create the simulation that our department uses for student lecture registration. It will be created in accordance with the department's policies and guidelines for attending lectures. Our system's simulation will require an input file as well as arguments like students, lectures, and simulation runs.~~

The system that our department uses to register students for classes built, and a simulation utilized to see how it works. An input file and independent variables like students, lectures, and simulation runs are needed to simulate our system.

2.) Description

~~All universities, including our university, require this system for student registration. Without this registration system design, it would be far more difficult for students to participate in the lecture-taking process. On the other hand, students may have a variety of issues when attending lectures, including lecture quota, necessary courses, and collision.~~

~~We created this software that simulates course registration in order to avoid these issues. In essence, students register for classes via their curriculum's registration system, while advisors keep track of their requests by taking care of the aforementioned issues.~~

The students would have a very tough time enrolling in the lecture if we hadn't designed this registration method. However, without this system, students could run into issues can not saw lecture quotas, required lectures, and collisions while attending classes.

In order to study how students select their lectures, we have also created software that simulates the lecture enrolling process. In essence, advisors monitor student needs while tracking student enrollment through the registration process for their curriculum. All of these phases may be seen thanks to the simulation system we have set up. In reel life, this simulation allows us to predict how the system will operate.

3.) Requirements

Functional Requirements

a. Lecture requirements

- Lecture must have lecture ID.
- Lecture must have name.
- Lecture must have lecture type.
- Lecture must have credit score.
- Lecture must have quota limit.
- Lecture may have different sessions.
- Lecture may have pre- requisite lecture(s).

b. Student requirements

- Student must have advisor.
- Student must have student ID.
- Student must have transcript.
- Student must have entry date.
- Student must have application registration.
- Student may lecture list.
- Student may have debt.

c. Lecture enrollment requirements

- Selected lectures must not have confliction.
- Student must pay education debt.
- Advisor must approve if requirements met.

Non-Functional Requirements

- Student information is taken from JSON file as an input.
- ~~Registration system program has to be implemented with JAVA.~~
- Every step of program should be performed on Command Line Interface.
- A simulation should be created to observe the lecture selection scenarios.

4.) Use Cases

Use Case Name: Enrolling To lectures

Summary: In order to get lectures, students must enroll to lectures from system and send request to their advisor.

Subject: Student

Basic Flow:

~~1.) Student opens "Enroll to lectures" tab at site to view lectures.~~
~~2.) Student needs to select their lectures from a lecture list according to their current curriculum, current semester and their current status of lecture progression.~~
~~3.) System saves lectures as draft at another tab.~~
~~4.) Student sends their lecture draft list to their advisor for approval.~~
~~5.) Student's enrolling to a lecture process has done, lectures from list added to their syllabus.~~

1.) The student logs into the system to view the lectures.

2.) Students view their lectures according to their current curriculum, semester and current lecture progress.

3.) The student saves the selected lectures as drafts.

4.) The student submits the list of lectures he/she has prepared to the approval of his/her advisor.

5.) Selected lectures are canceled or rejected by the advisor, depending on availability.

6.) Lectures approved by the advisor are displayed in the student's curriculum.

Alternative Flow:

- Step 2: if selected lecture has reached their quota capacity system gives warning to student, then use case returns to step 2.
- Step 4: if student needs to pay education debt and he/she didn't paid it yet, system sends warning to student when he/she sends draft to advisor and student returns to step 2.
- Step 4: if there is conflict at lecture draft student can't send lecture draft to advisor, student returns to step 2.
- ~~Step 5: if student sent TE lecture to advisor for approval, even though they have taken 2 TE lectures in fall semester, advisor rejects that TE lecture~~
- Step 5: If the student sends the 4th TE lecture to the advisor's approval despite having taken 3 TE lectures in the spring semester, the advisor rejects that TE lecture.
- Step 5: if student sent TE lecture to advisor for approval, even though they have taken 3 TE lectures in spring semester, advisor rejects that TE lecture.
- Step 5: if student sent FTE lecture to advisor for approval and if student's graduation is impossible at that semester advisor rejects that FTE lecture.

Use Case Name: Approving/Rejecting lecture Requests

Summary: In order to enregister students to lectures, advisor must enter the system and approve or reject coming requests from their students.

Subject: Advisor

Basic Flow:

~~1.) Advisor enters to systems' site and selects coming requests tab from site for approving or rejecting coming requests.~~
~~2.) Advisor checks coming lecture requests from student.~~
3.) ~~Based on advisor's feedback system sends notification to student.~~

1.) The advisor logs into the system and seen the lectures selected by each student.

2.) The advisor checks the lecture requests from the student whether they are conflict, inadequate or inappropriate for any reason.

3.) The advisor's thanks to the feedback system, it sends a notification to the student in the form of approval or rejection.

Alternative Flow:

- ~~Step 2: if request has TE lecture even though student took 2 TE lectures in fall semester, advisor needs to reject that TE lecture, after that advisor returns to step 2 for another student.~~
- ~~Step 2: if request has TE lecture even though student took 3 TE lectures in spring semester, advisor needs to reject that TE lecture, after that advisor returns to step 2 for another student.~~
- ~~Step 2: if request has FTE lecture and if student's graduation is impossible at that semester advisor needs to reject that FTE lecture, after that advisor returns to step 2 for another student.~~
- Step 2: If the student took 2 TE lectures in the fall but took an on-demand TE lecture, the advisor must decline that TE lecture after returning to step 2 for another student.
- Step 2: If the student is taking 3 TE lectures in the spring semester but there is one TE lecture on demand, the advisor must decline that TE lecture after returning to step 2 for another student.
- Step 2: If the selected lectures are too much or difficult for the student, the advisor rejects the lectures he or she deems appropriate.
- Step 2: If there is an optional FTE lecture and the student cannot graduate at that time, the advisor should decline that FTE lecture after returning to step 2 for another student.

Use Case Diagram

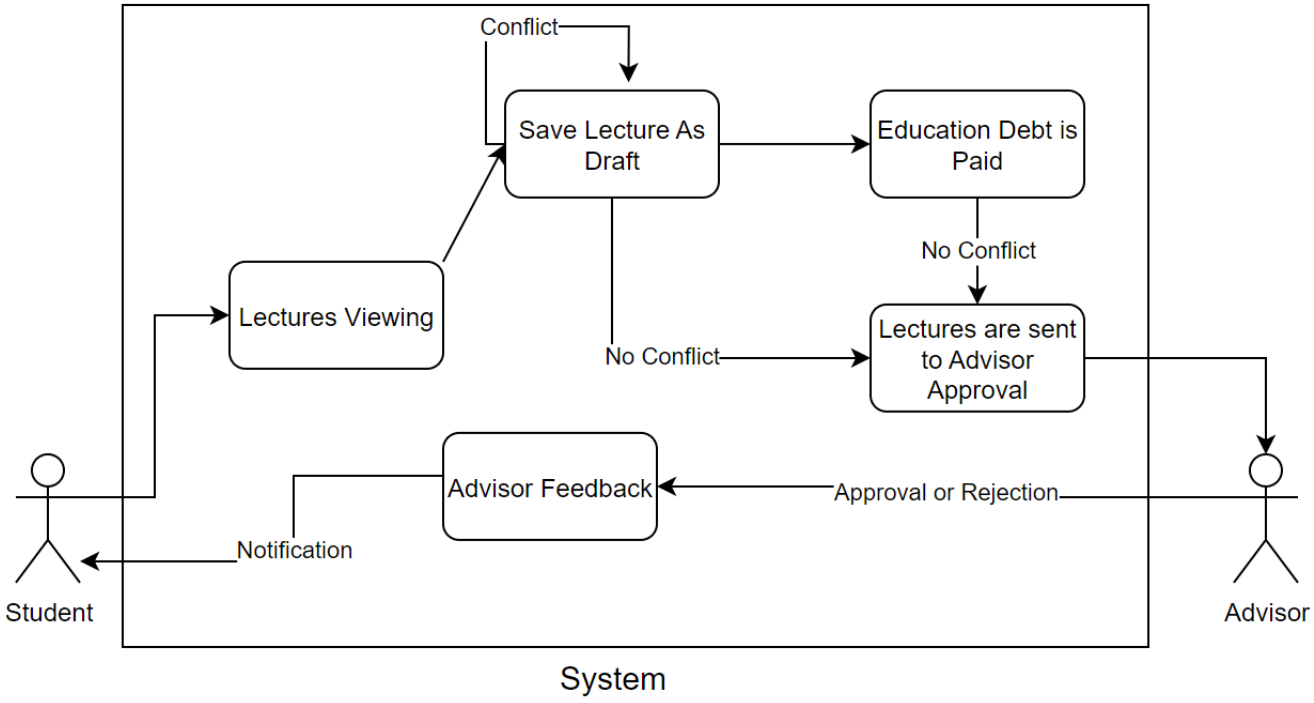
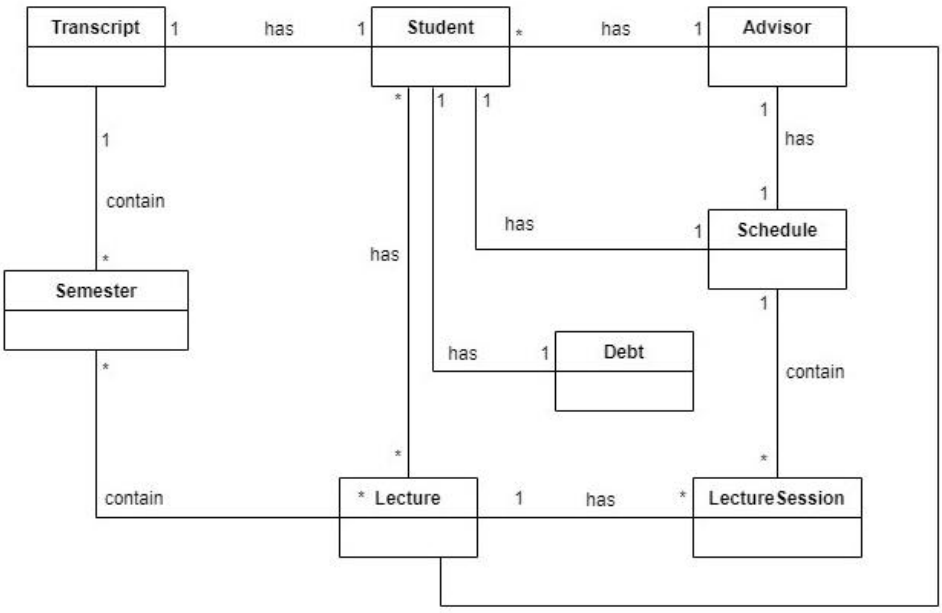
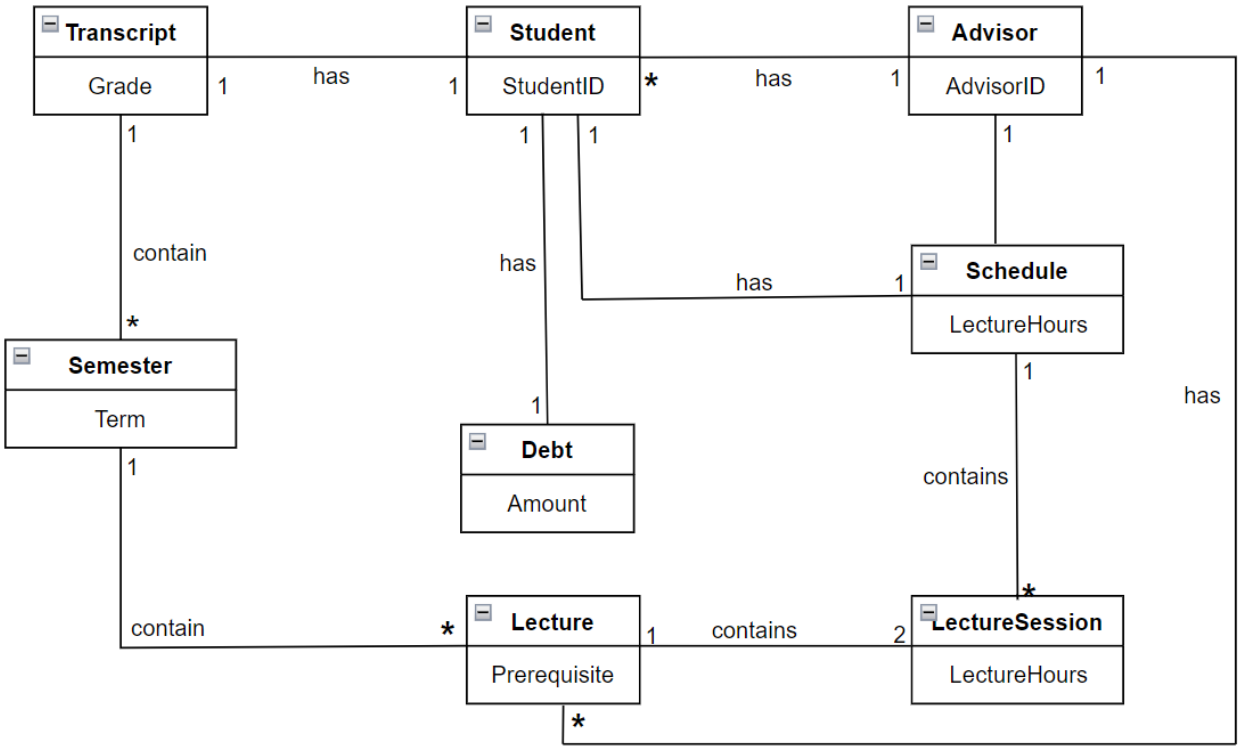


Diagram 1

4.) Domain Model



Old diagram 2



New Diagram 2

5.) System Sequence Diagram (SSD)

1.) ~~After viewing the lectures they want to attend, students save their chosen lectures as drafts.~~ Students view the lectures they want to attend
2.) ~~Students that owe money for tuition repay it. (This is a alternative action.)~~ Students save their chosen lectures as drafts.
3.) The student submits the chosen lectures for approval from the advisor if there are no conflicts.
4.) ~~The advisor reviews the lectures the student has selected.~~ The student submits the chosen lectures to the advisor for approval if there is no conflict.
5.) ~~The chosen lectures have the advisor's approval.~~ The advisor approves or rejects the lectures chosen by the student after reviewing them.
6.) The student views the lectures approved by the advisor in the lectures schedule.

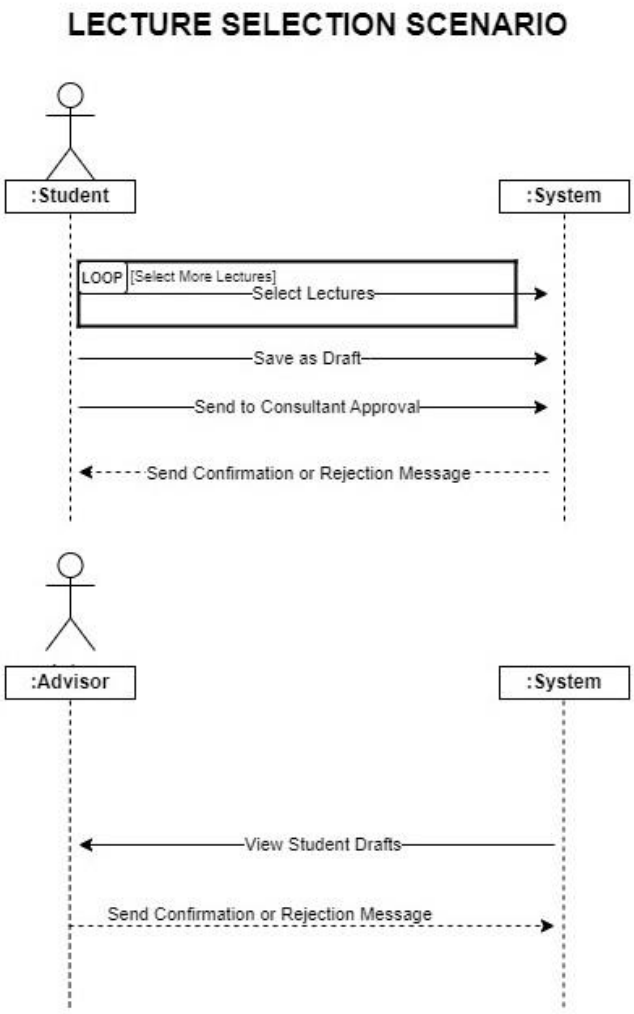


Diagram 3

6.) Glossary

- **Lecture :** Lessons students take
- **Lecture Session :** given lectures by different lecturers
- **Transcript :** Students' grade of lecture records
- **Advisor:** Actor who approves or rejects requests of students for taking lectures , stakeholder Curriculum : All of the semester's lectures
- **Student :** Actor who takes lecture from system , stakeholder
- **Schedule :** Student's weekly plan
- **Semester :** A half-year term in school, university
- **FTE Lecture :** Faculty Technical Elective Lecture
- **NTE Lecture :** Non Technical Elective Lecture
- **TE Lecture :** Technical Elective Lecture
- **Prerequisite Lecture :** Required lecture must be completed prior to take other Lecture
- **UE Lecture :** University Elective Lecture Credits : Is a way of measuring and impression of particular lecture