MARMARA UNIVERSITY

FACULTY OF ENGINEERING

GROUP 2

COURSE REGISTRATION SYSTEM- ITERATION 1

Student Names:

1- Yusuf Demir – 150120032
2- Sena Ektiricioğlu – 150120047
3- Beyza Nur Kaya – 150120077
4- Muhammed Hayta – 150121068
5- Selin Aydın – 150120061
6- Eren Duyuk – 150120509
7- Mustafa Emir Uyar - 150120007

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Vision

The vision of the project is to create a course registration system that will simplify the process of enrolling in courses and submitting them for approval to advisors. The proposed system will provide students with a command line interface that allows them to easily browse through available courses, select the ones they wish to enroll in and submit their selections and send the selections to approval. This will significantly reduce the time and effort required for students to manage their course schedules while ensuring the accuracy of their course records.

1.2 Scope

The student module offers various features, including course search and selection, course updates, and the ability to send course selections for approval to an advisor. It also provides a streamlined registration process with clear instructions. On the other hand, the advisor module offers course management functionality for students and includes the ability to approve course enrollment requests.

1.3 System Constraints

• The system must be compatible with the existing university JSON file.

• The system should have scalability to cater to medium-sized student populations.

• Basic security measures must be implemented to safeguard user information.

• The system must comply with university guidelines.

1.4 Stakeholders

- **Students:** End-users responsible for course selection and registration.
- **Advisor:** Users who provide management and functionality to course enrollment approval for students.

1.5 Problem Statements

- Issue 1: Selecting courses can be a challenging and time-consuming task for students.
  Impact: It often leads to delays and errors.

- Issue 2: Advisors encounter manual processes when approving courses.
  Impact: It can result in reduced efficiency, higher workloads, and an increased likelihood of errors.

2.REQUIREMENTS

2.1 Functional Requirements

- Both students and advisors must have secure login credentials to access the system.
- Students can view a list of available courses
- Students can select and enroll in available courses.
- Students can remove themselves from selected courses while in draft status.
- Advisors can see the courses that students want to enroll in.
- Advisors can approve or reject the course enrollment requests from students.
- The system should display detailed information about each course, including the course name, lecturer name etc.
- The system should ensure seamless integration with existing university JSON files.

2.2Non-Functional Requirements:

- The system should ensure the security of user data, and secure authentication methods.
- The command line interface should be intuitive and easy to use for both students and advisors.
- The system should be stable and available for use.

3.GLOSSARY

Term	Definition and Information	Format	Validation Rules
User	User are stands for the people who are primary stakeholders to the system. It is also a superclass of advisor and student.	Abstract class	User must be an advisor or a student of Marmara University.
Advisor	Advisor is the subclass of User class which has a list of students as attributes and several methods.	Class	An advisor must have a unique username.
Student	Student is the subclass of User class which has several attributes and methods.	Class	A student must have a unique username and must be a third-grade student at Marmara University.
Username	It is the unique name for each user to login to their account.	String type attribute	Student’s username must start with “o”.
Course	Course class is for creating a course with attributes such as courseCredit,courseName ,prerequisiteInformation and courseSection.	Class	A student can take a course if they have passed all prerequisite courses, and any section of the course is available.
Prerequisite	It contains courses that are prerequisites for taking courses. It checks whether prerequisite	Class	

	courses are passed using the methods in it.		
Course status	It contains status draft, pending, approved, and denied for courses.	Enum	
Course result	It contains the situation of the course such as passed, failed, or active.	Enum	
Database manager	Database manager provides integration with university database file.	Class	It must be compatible with Java Json library named Jackson.
Authenticate service	Authenticate service has methods to verify usernames and passwords.	Class	
Authenticate user	Authenticate user is the method for password and username verification using database information.	Method that returns User	
Transcript	It contains calculations such as completed credits and GPA.	Class	
CLIAdvisor	It is a command line interface (CLI) that has pages for monitoring students and their courses.	Class	It must be available only for advisors.
CLIStudent	It is a command line interface (CLI) that has pages for monitoring available courses.	Class	It must be available only for students.
CLILogin	It is a command line interface (CLI) that contains the login page.	Class	

4.USE CASES

4.1 USE CASE UC1: COURSE SELECTION

Scope: University Course Registration System

Level: User-goal

Primary Actor: Student

Stakeholders and Interests:

-Student: wants to view the courses that they can select and choose according to that information, send the courses they had selected to their advisor.

Preconditions: Student is identified before the registration and successfully login to the system.

Success Guarantee or Postconditions: Student can view the courses they can take. They can correctly select the courses. They also need to send them to the advisor with no issues.

Main Success Scenario or Basic Flow:

1. Student opens the system.
2. Student sees the courses available for himself/herself.
3. Student select courses s/he needs to get with their section if there are any provided.
4. Student saves the selected courses with their section information.
5. Student sends the selected courses to his/her advisor.
6. Student closes the system.

Extensions or Alternative Cases:

*a. At any time, student may want to exist the system.

1. Student select the exit command.

2. Student exits the system.

3a. Student may select courses s/he not needed.

1. Student removes these courses from the selected ones.
2. Return back to step 2.
   4.a Student chooses the course that s/he cannot get, therefore while saving there is an error.
   4a1. Student may enter non existing course code.

1. Save operation cannot be done.

2. System gives the error message.

3. Return back to step 2.

4a2. If the section is full, it cannot be selected.

1. System does not save the course.

2. System gives the error message.
3. Return back to step 2.

4.2 USE CASE UC2: COURSE APPROVALS

Scope: University Course Registration System

Level: User-goal

Primary Actor: Advisor

Stakeholders and Interests:

-Advisor: wants to view their students they advise, see what courses they selected, review elected courses, and approve or reject their application status.

Preconditions: Advisor is identified before the approval and successfully login to the system.

Success Guarantee or Postconditions: Advisor views the students s/he advises. Advisor views the courses selected by the student s/he advises. Advisor reviews the courses selected and approves or rejects accordingly.

Main Success Scenario or Basic Flow:

1. Advisor opens the system.
2. Advisor views the students that s/he is responsible for.
3. Advisor selects the student and sees the courses that the student selected.
4. Advisor reviews the courses that student selected before.
5. Advisor approves the courses that student selected.
6. Advisor closes the system.

Extensions or Alternative Cases:

*a. At any time, advisor may want to exist the system.

1. Advisor select the exit command.

2. Advisor exits the system.

3a. Advisor may select non-existing student id.

1. System cannot select the student.
2. System gives the error message.
3. Return back to step 2.

5a. Advisor can reject the courses that the student selected partially or all of them.

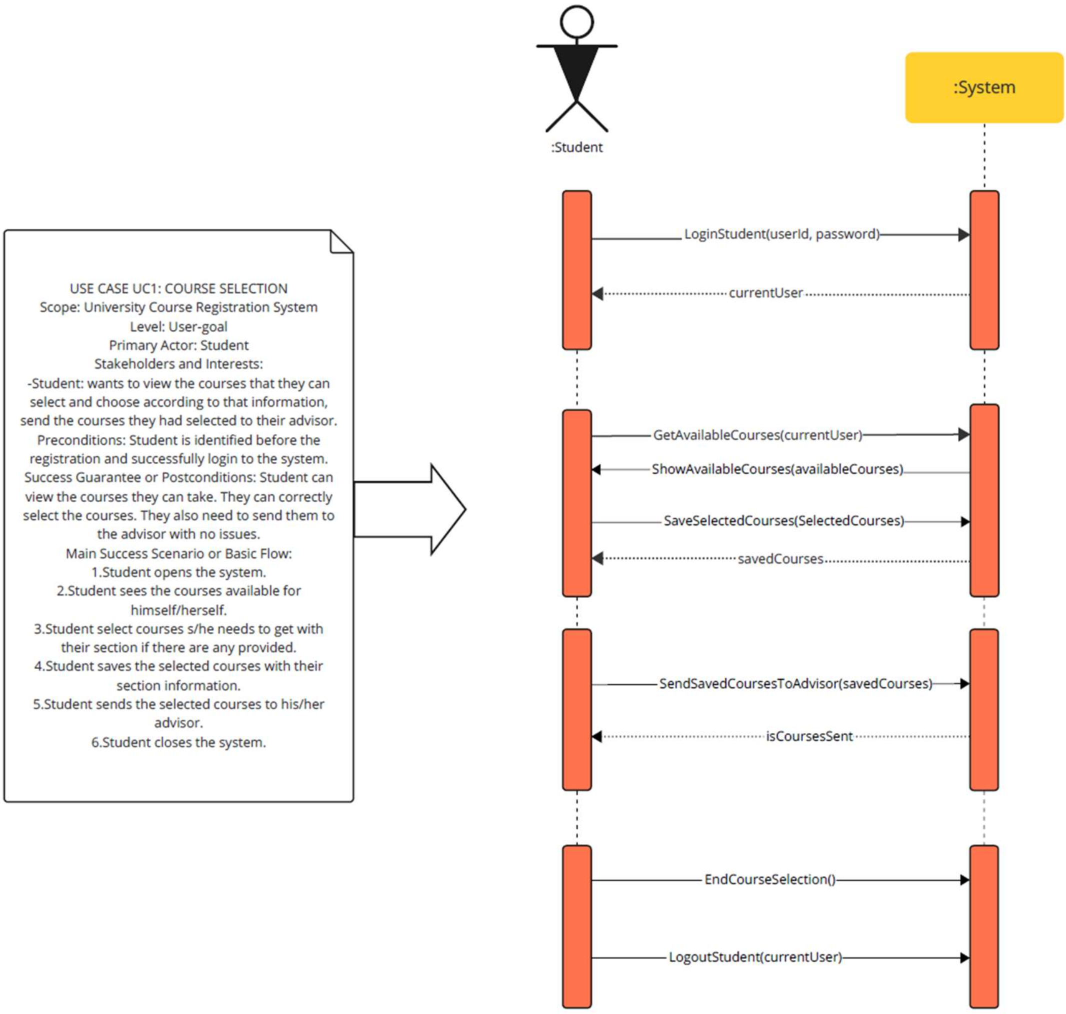
1. The system changes the status of student's course registration by rejecting.
2. System gives the error message to student.

6a. Advisor can choose another student instead of closing the system.

1. Return back to the step 2.

5.SSD DIAGRAMS

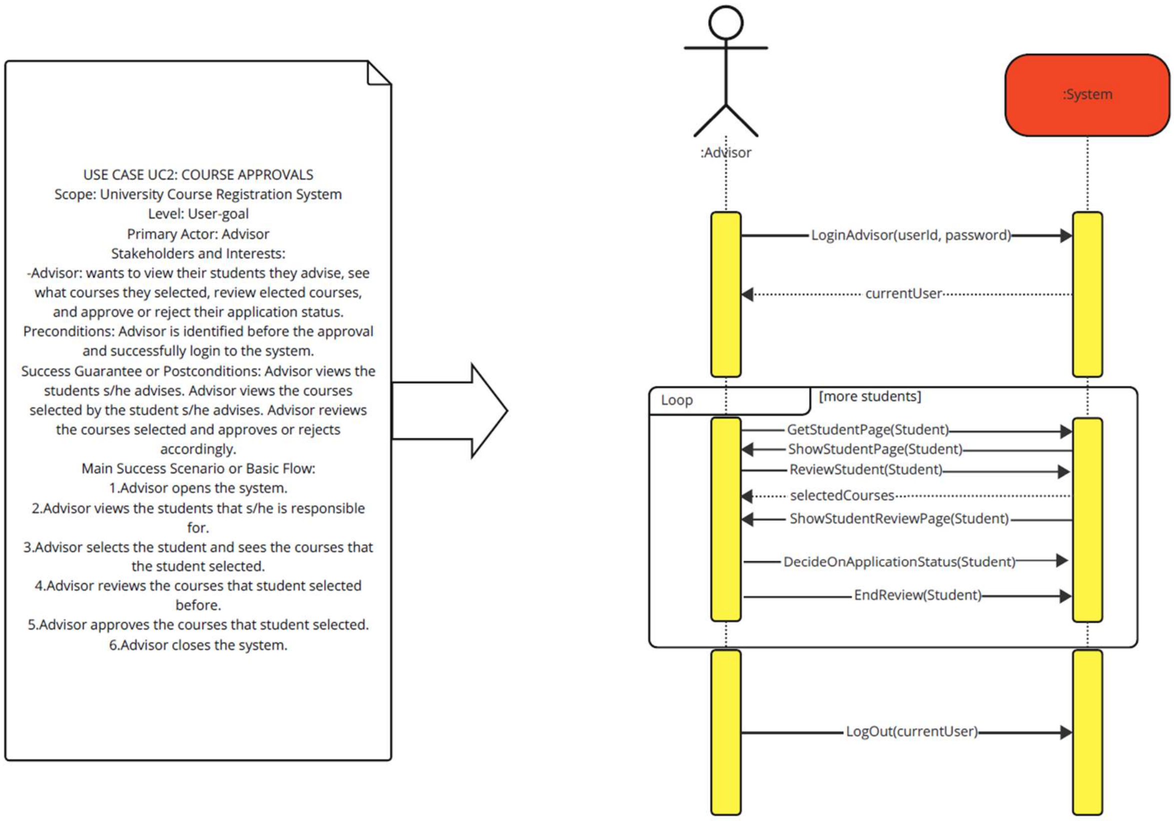
5.1 Student SSD



USE CASE UC1: COURSE SELECTION
Scope: University Course Registration System
Level: User-goal
Primary Actor: Student
Stakeholders and Interests:
-Student: wants to view the courses that they can select and choose according to that information, send the courses they had selected to their advisor.
Preconditions: Student is identified before the registration and successfully login to the system.
Success Guarantee or Postconditions: Student can view the courses they can take. They can correctly select the courses. They also need to send them to the advisor with no issues.
Main Success Scenario or Basic Flow:
1.Student opens the system.
2.Student sees the courses available for himself/herself.
3.Student select courses s/he needs to get with their section if there are any provided.
4.Student saves the selected courses with their section information.
5.Student sends the selected courses to his/her advisor.
6.Student closes the system.

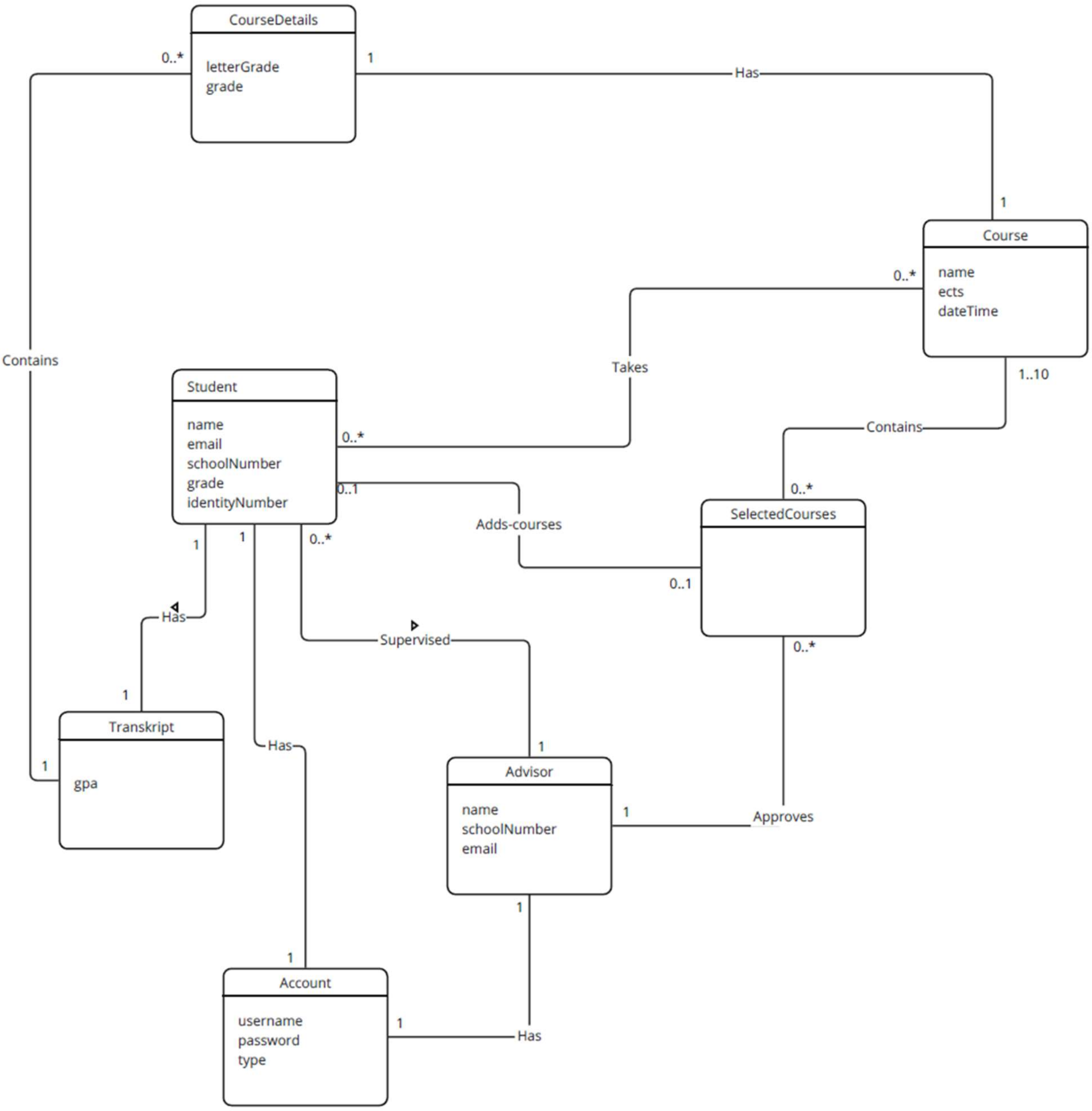
:Student
:System
LoginStudent(userId, password)
currentUser
GetAvailableCourses(currentUser)
ShowAvailableCourses(availableCourses)
SaveSelectedCourses(SelectedCourses)
savedCourses
SendSavedCoursesToAdvisor(savedCourses)
isCoursesSent
EndCourseSelection()
LogoutStudent(currentUser)

5.2 Advisor SSD



USE CASE UC2: COURSE APPROVALS
Scope: University Course Registration System
Level: User-goal
Primary Actor: Advisor
Stakeholders and Interests:
-Advisor: wants to view their students they advise, see what courses they selected, review elected courses, and approve or reject their application status.
Preconditions: Advisor is identified before the approval and successfully login to the system.
Success Guarantee or Postconditions: Advisor views the students s/he advises. Advisor views the courses selected by the student s/he advises. Advisor reviews the courses selected and approves or rejects accordingly.
Main Success Scenario or Basic Flow:
1.Advisor opens the system.
2.Advisor views the students that s/he is responsible for.
3.Advisor selects the student and sees the courses that the student selected.
4.Advisor reviews the courses that student selected before.
5.Advisor approves the courses that student selected.
6.Advisor closes the system.

6. DOMAIN MODAL


CourseDetails
letterGrade
grade
0..*
1
Has
1
Course
name
ects
dateTime
0..*
Takes
1..10
Contains
Contains
Student
name
email
schoolNumber
grade
identityNumber
0..*
0..1
Adds-courses
0..1
0..*
SelectedCourses
1
1
0..*
Has
Supervised
0..*
1
Transkript
gpa
1
Has
1
Advisor
name
schoolNumber
email
1
Approves
1
1
Account
username
password
type
1
Has