

EGE UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING



OBJECT ORIENTED ANALYSIS AND DESIGN
2017-2018 SPRING SEMESTER
PROJECT REPORT

PROJECT TOPIC: EBYS

PREPARED BY
05140000622- BERKAN AFŞAR
05150000672- FATMA AYSU KÖKSAL
05140000688- GÖKSEL ÇELİK

DEADLINE
11.05.2018

CONTENTS

1- Requirement Analysis.....	2-10
1.1- Vision.....	2
1.2- Use Case.....	2
1.3- Supplementary Specifications.....	9
2- System Sequence Diagrams.....	11-12
3- Domain Model.....	13-14
3- Interaction Diagrams.....	14-15
3- Class Diagram.....	16

1. REQUIREMENT ANALYSIS

We accepted this method to find requirements:

- 1 -Write a brief first draft of the Vision.
- 2 -Identify user goals and supporting use cases.
- 3- Write some use cases and start the Supplementary Specifications.
- 4- Refine the Vision, summarizing information from these.

1.1 – Vision

1- Introduction

We envision a next generation fault-tolerant (EBYS) system. EBYS with the reliability to keep documents safe, compatible with android/ios and offers maximum performance in the minimum time.

2- Positioning

2.1- Business Opportunity

All universities get benefit if a cloudbase system can be produced. Easier to use than systems produced by other companies. It has an user friendly interface. Response quickly to requests.

2.2- Problem Statement

The security of documents is not guaranteed. It is difficult to process the documents. More labor needed. The difficulty of archiving and retrieving documents. Students can not access education information easily and quickly. It is costly for universities to produce their own systems.

2.3- Product Position Statement

The system has the ability to resolve the errors which it has encountered before. The information of the user is always kept confidential. For students, they would like to reach their information which this system uniquely makes possible to do this fastly, easily and the most importantly safe.

1.2- Use Case

We accepted this method to find requirements:

- 1- Choose the system boundary.
- 2- Identify the primary actors.
- 3- For each, identify their user goals.
- 4- Define use cases that satisfy user goals.

1- Boundary

It is web-based software application. Students, teacher and student affairs officers use this applications.

2- Primary Actors

Student, teacher, mentor and student affairs officer.

3- Find User Goals

For Student:

Views syllabus

Views grade status

Views transcript

Re-Registration/ Course selection operations

Re-Information of tuition

Re-Information of examination

Appeal double major / minor program

Views information of preparatory class

Applies to refund tuition

Applies to renew identity

Selects language

Logs in and Logs out from System

Changes Account Password

For Teacher:

Edits grade

Edits absenteesim

Determines precentage of grades

Views academical program

Views syllabus

Lecture / class operations

Edits grade via Excel

Views statistics

Views grafics

Views own lectures

Views exam grades

Views all students

Sends message

Selects language

Logs in and Logs out from System

Changes Account Password

For Mentor:

Confirms registration

Confirms course selection

Selects language

Logs in and Logs out from System

Changes Account Password

For Student Affairs Officer:

Document operations

Diploma operations

Graduation procedures

Freeze enrolment procedure

Allow enrolment procedures

Selects language

Logs in and Logs out from System

Changes Account Password

4- Use Cases

Use Case UC1: Process Enrolment

Primary Actor: Student

Stakeholders and Interests:

-Student: Wants to enroll fast entry no enrolment errors with minimal effort during his/her enrollment process.

-Mentor: Wants to confirm to all request of enrollment without errors.

Preconditions: Student is identified. Mentor is identified and authenticated. Student must have finished first semester. Student administration Office starts the process of enrollment.

Success Guarantee(Postconditions): The enrollment must be completed. Syllabus is created. Enrollment authorization approval are recorded.

Main Success Scenario:

1. Student starts the enrollment process using course identification number and student identification number.
2. System returns the list of semester courses.
3. Student selects courses and adds them into taken courses list according to terms and conditions of E.U.

The students repeats step 3 until students done.

4. System saves the list which selected by student.
5. Student sends the list for get confirmation of his/her mentor.
6. System returns the confirmation.

Extensions(or Alternative Flows):

- a. At any time system fails:

To support recovery and correct enrollment, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Student refreshes the page, logs in and requests recovery of prior state.
2. System reconstructs prior state.
- 2a. System detects anomalies preventing recovery:
 1. System signals error to the students records the error, and enters a clean state.
 2. Students enter a new Re-Registration/ Course selection page.
- 1a. Page not found:
 1. System returns error message and rejects request.
- 2a. System couldn't return the courses because of DB connection errors.
 1. Refresh the page.
 2. Re-entry the system.
- 3a. Courses couldn't be selected.
 1. Try to choose the courses again.
- 4a. The list of courses couldn't save because of DB connection errors.

1. System records selection.
2. Returns the error message.
3. Re-try to save.

5a. Mentor couldn't see the list of courses.

1. Refresh the page.

5b. Mentor couldn't confirm.

1. Sent the error message to the system.

Use Case UC2: Edit Grade by Teacher

Primary Actor: Teacher

Stakeholders and Interests:

-Student: wants to enroll fast entry no enrolment errors with minimal effort during his/her process of showing grades.

-Teacher: wants to record the grades with minimal effort without errors.

Preconditions: Course is identified. Teacher is identified and authenticated. At least one student must enroll this course.

Success Guarantee(Postconditions): Editing is done. Grades transfer to system. System convert to letter grade. System calculates GPA.

Main Success Scenario:

1. The teacher starts a new editing process using student identification number, teacher identification number, grade type and grade.
2. System shows a list of the courses that given by her/him.
3. The teacher selects the course by using course identification number.
4. System returns a list of the students that are taking the course .
5. The teacher selects student using by student identification number and edits his/her grade.
The teacher repeats step 5 until students done.
6. The teacher sends the list with grades.

Extensions(or Alternative Flows):

a. At any time system fails:

To support recovery and correct editing, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. System returns error message.

2. Teacher refresh the page.

2a. System returns a list that has missing courses:

1. The teacher sent request again.

2b. System couldn't show the list because of DB connection errors:

1. Returns the error message.

2. The instructor sent request again to system.

3a. The course which is wanted to select is not in the semester.

1. Teacher should select another course.

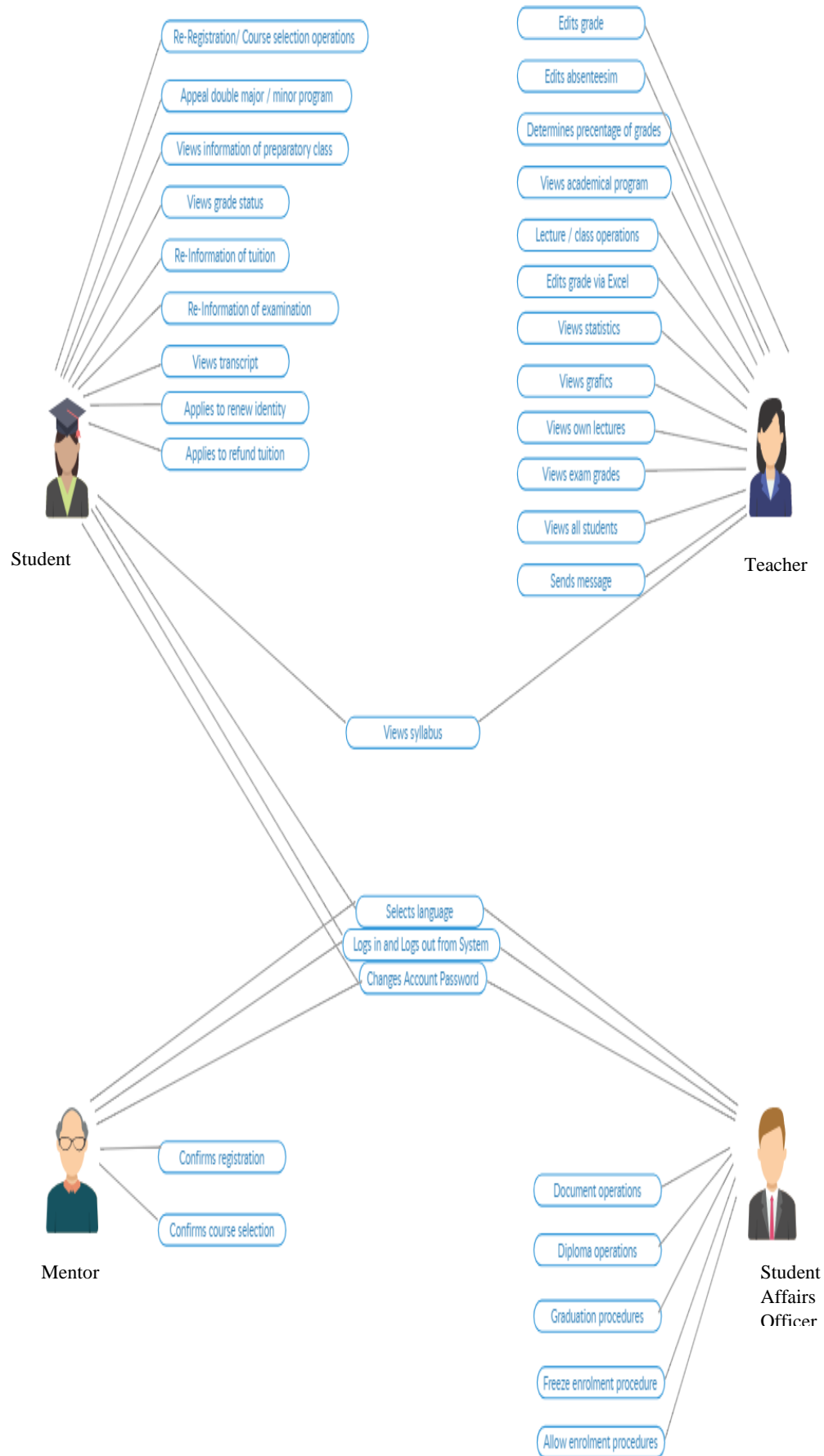
2. Re-entry the system.

4a. A student identified twice.

1. Teacher reports the situation to the student administration officer .

5a. The teacher tries to edit the grade out of valid boundary.

1. Return error message.



1.3 SUPPLEMENTARY SPECIFICATIONS

1- Introduction

This document is the repository of all EBYS requirements not captured in the use cases.

2- Functionality

2.1- Logging and Error Handling

Returns error message. Communicates with the responsible person. If this error was identified already, system find a solution for it. Self- refreshes. Logs all error to permanent storage.

2.2- Pluggable Business Rule

At various scenario points of several use cases support the ability to customize the functionality of the system with a set of arbitrary rules that execute at that point or event.

2.3- Security

Students can not Access the information of other students. Teachers can not Access the information outside their course. Students can not change own grades and absenteeism. Only mentor can confirm the list of the course that wanted to select. Courses only first week can be selected by teacher and student.

3- Usability

3.1- Human Factors

Avoid colors associated with common forms of color blindness. Vocalization system is used. Text is easily visible.

4-Reliability

4.1- Recoverability

If there is a failure to use system, try to solve in order to still complete the process.

5- Performance

Our goal is to achieve our horizontalization in less than 1 minute, 90% of the time. System can response to more than one request at the same time.

6- Supportability

6.1- Adaptability

System can work together with other system that university's uses.

6.2- Configurability

System can be used at mobile platform.

7- Implementation Constraints

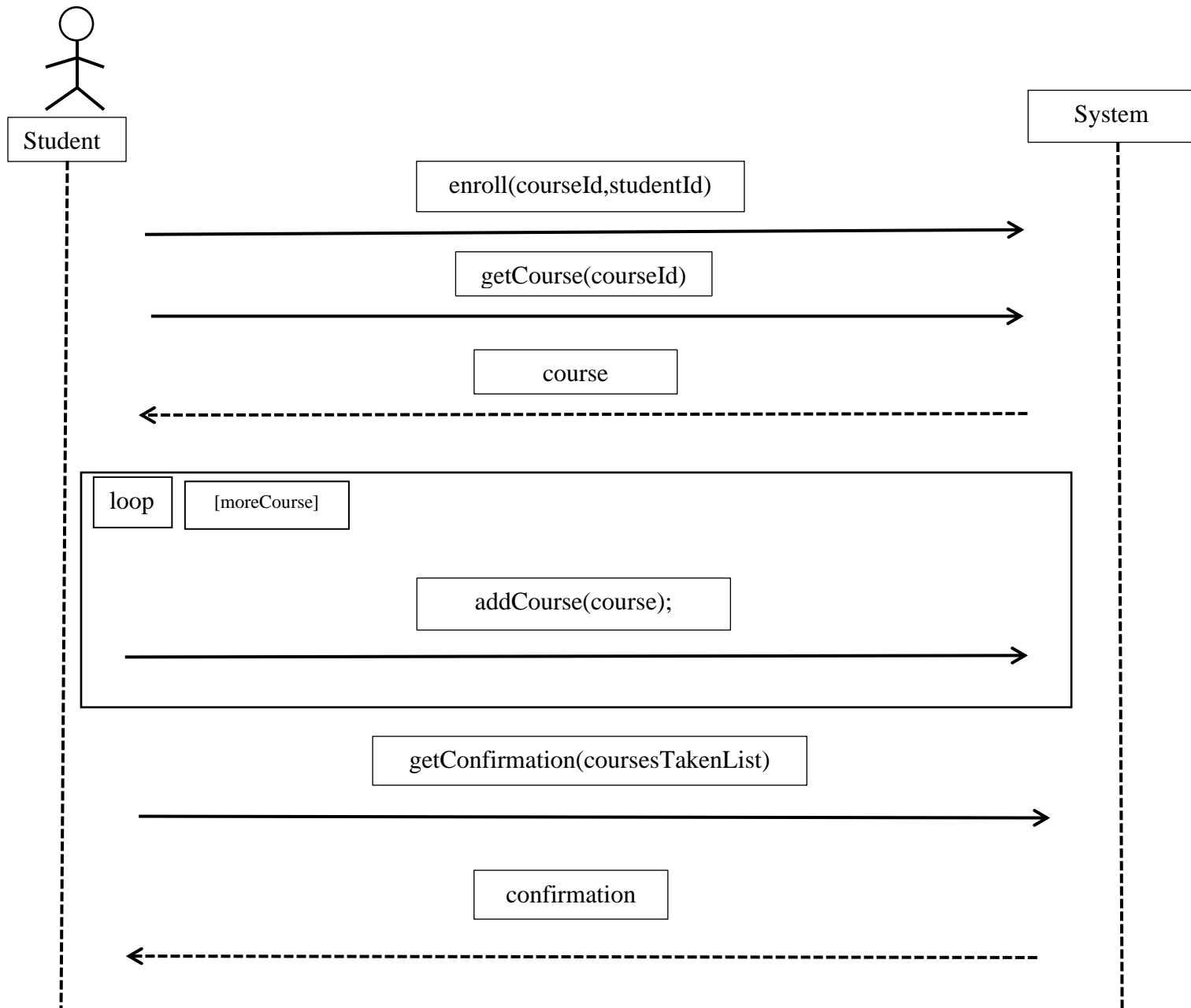
Java Technologies was used for this system because of popularity, adaptability and usability.

8- Business Rules(Domain Rules)

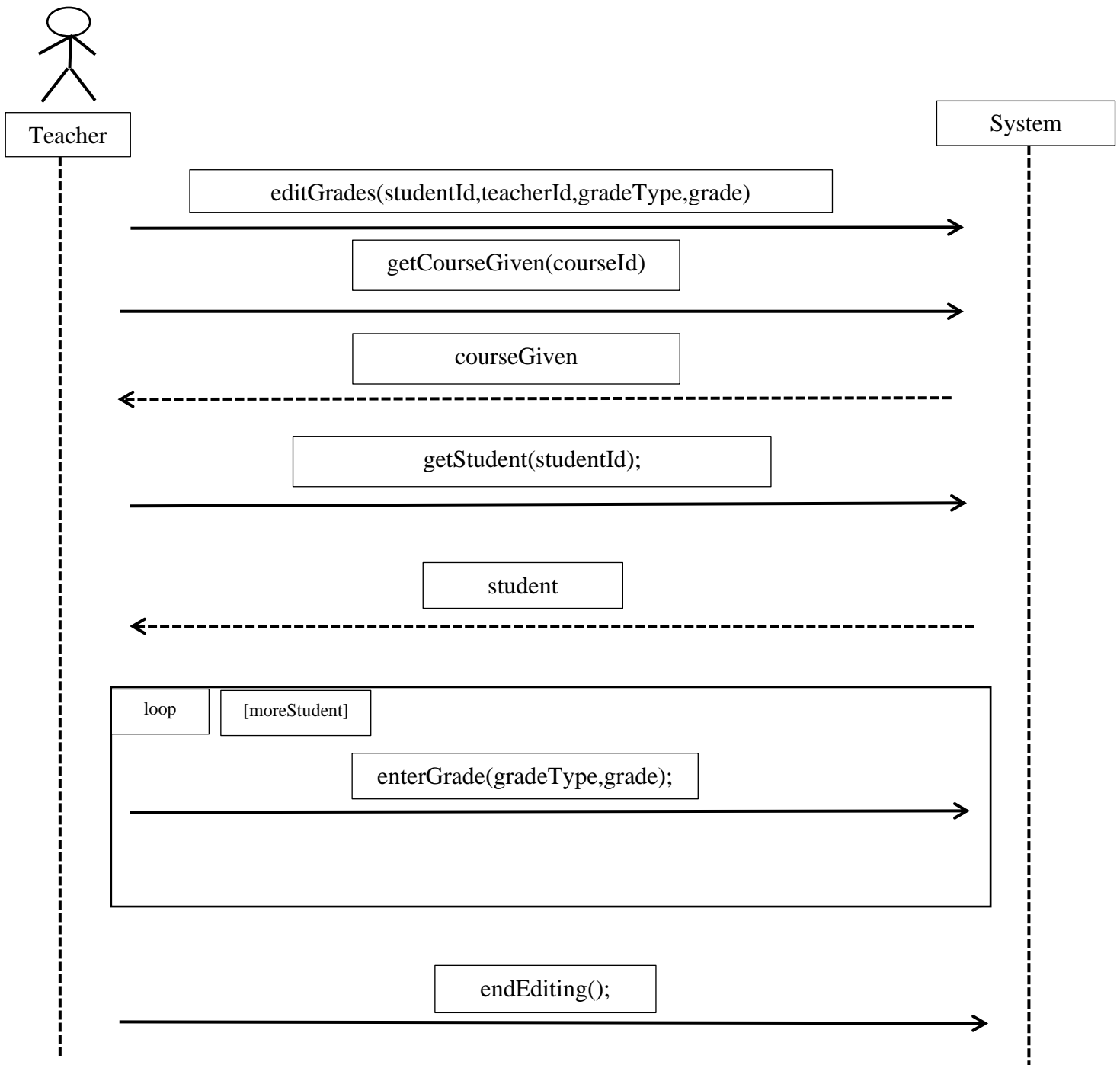
ID	Rule	Changeability	Source
Rule1	Students can take maximum 42 AKTS.	It is replaced by senate judgment.	Regulation of E.U.
Rule2	If the student had a course which is taken FF or taken firstly, student must not take at the same time as this course.	It is replaced by senate judgment.	Regulation of E.U.
Rule3	If the student took letter grade of CC at a course, the students must take this lesson again.	It is replaced by senate judgment.	Regulation of E.U.
Rule4	If there is a pre-condition for a course, the student must take the pre-condition courses first.	It is replaced by senate judgment.	Regulation of E.U.
Rule5	If a student wants to do double major program, the student has least 3.00 GPA or if a student wants to do minor program, the student has least 2.50 GPA.	It is replaced by senate judgment.	Regulation of E.U.
Rule6	%80 attendance is required for all courses. If a student can not ensure this condition for a course, the student fails this course.	It is replaced by teacher who gives this course.	Teacher

2. SYSTEM SEQUENCE DIAGRAMS

1-PROCESS ENROLMENT SSD



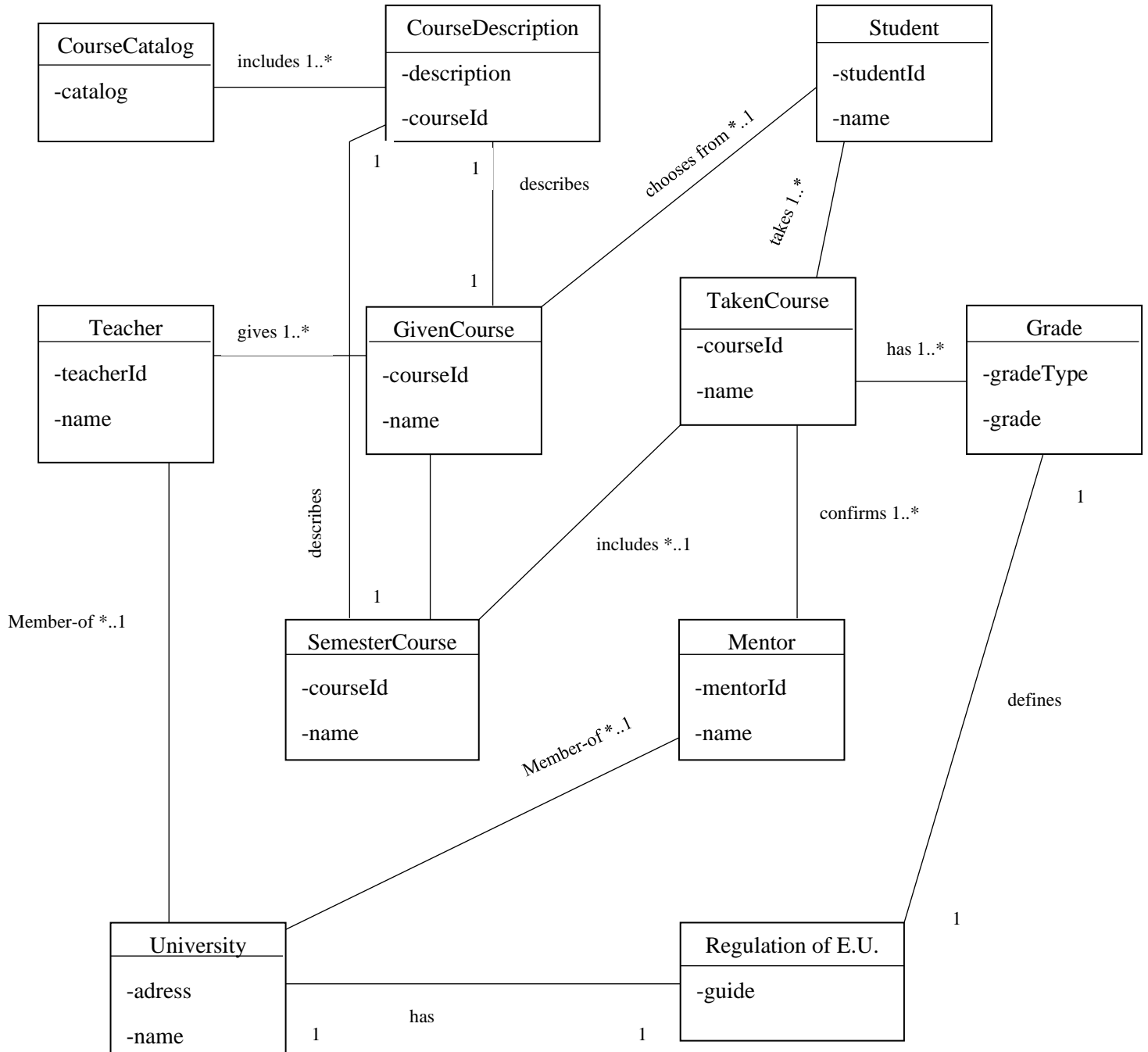
2- EDIT GRADE BY TEACHER



3. DOMAIN MODEL

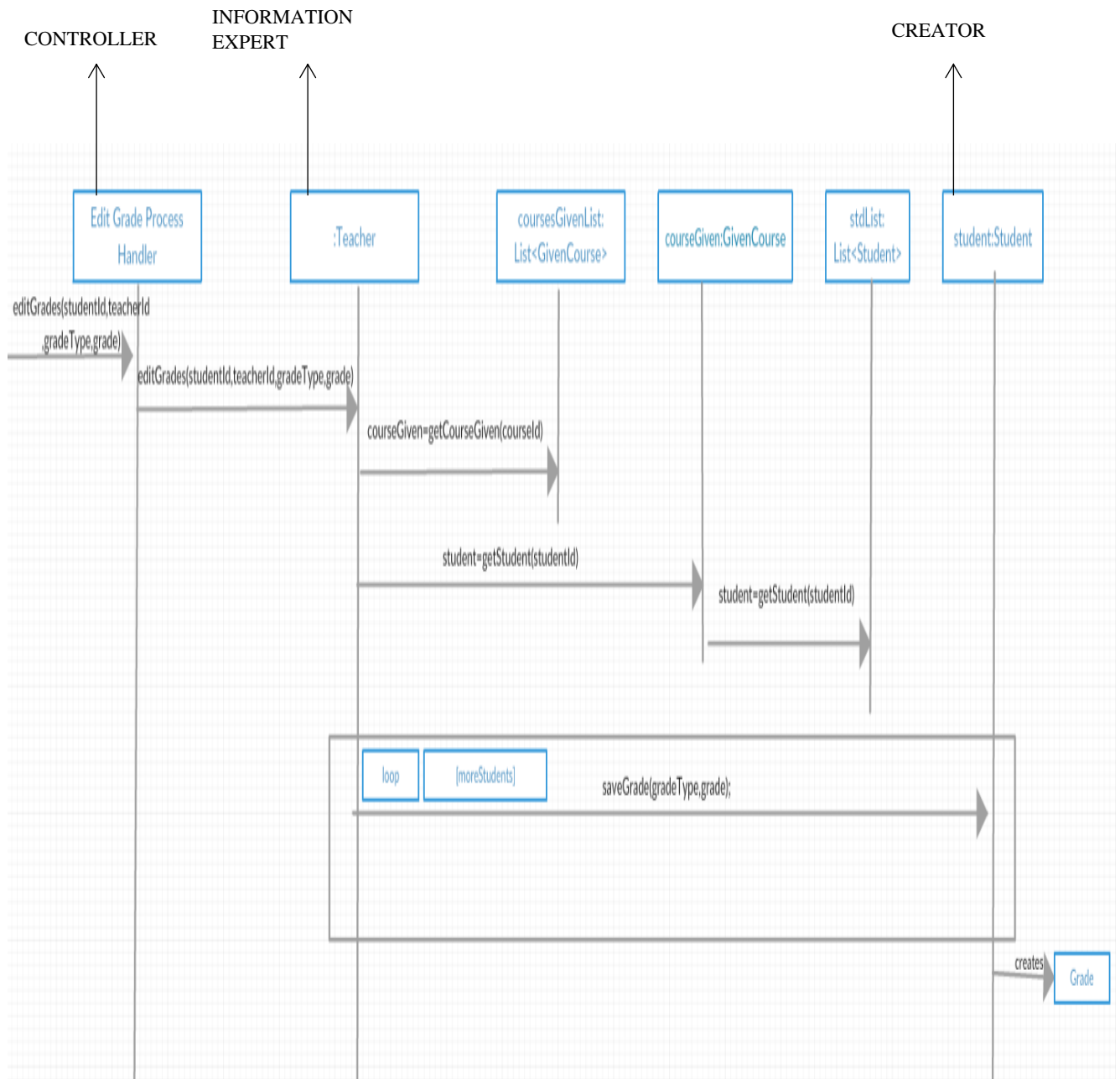
We accepted this method to find requirements:

- 1 – Find conceptual classes(Use noun phrase + category List).
- 2 – Find associations between conceptual classes(Find must to know relations + category list).
- 3- Find attributes of conceptual classes(Find primitive attribute).

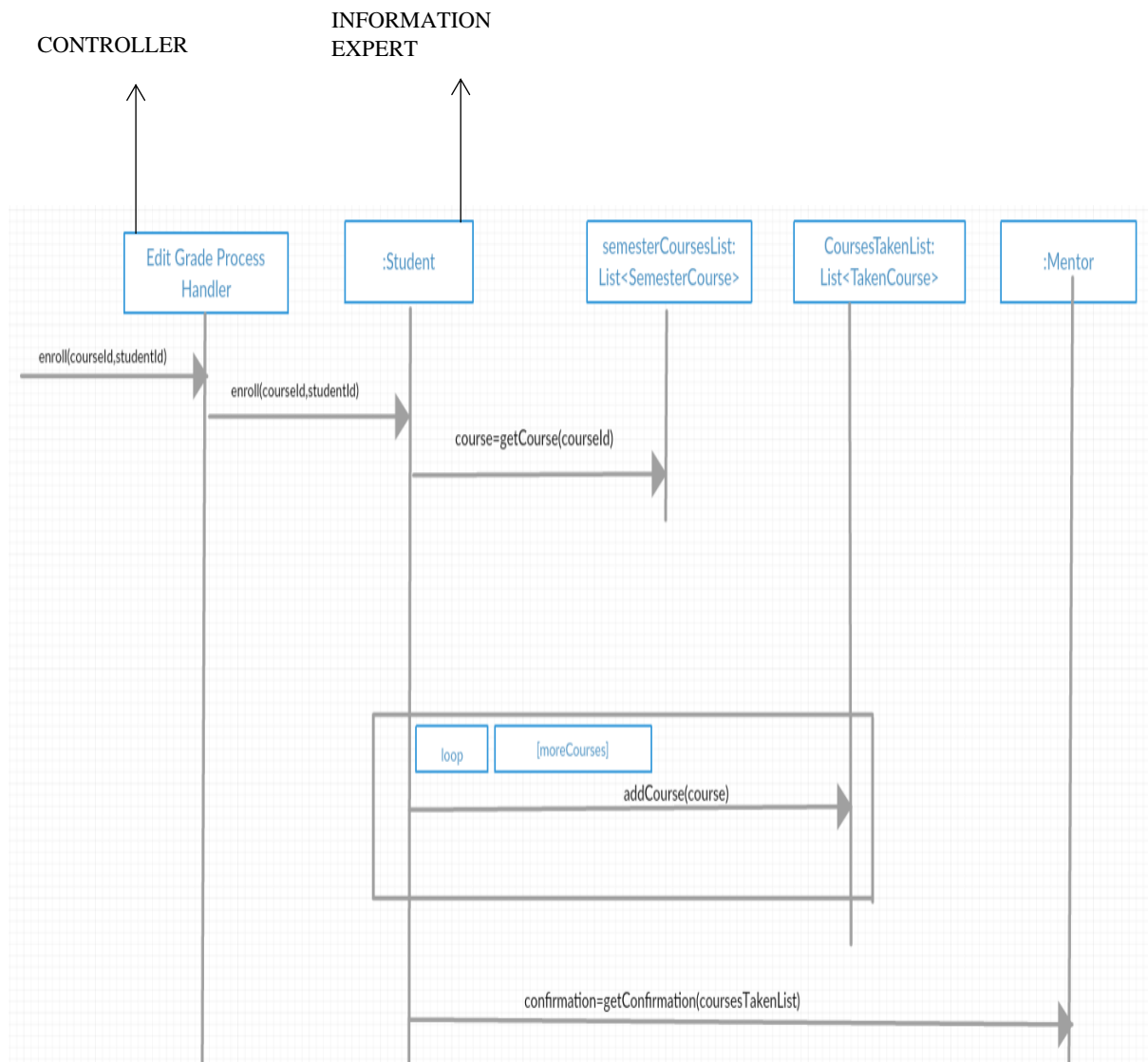


4. INTERACTION DIAGRAMS

1- EDIT GRADE BY TEACHER



2-PROCESS ENROLMENT



5. CLASS DIAGRAM

