Software Requirements Specification

CENG 396

Software Engineering
Project

Project Group 1

Stage 2- Software Requirements Specification

Names of members

Mert Alp Kuvandık 201911038
Emre Haser 201711407
Süleyman Serdar Erdemir 201811292
Volkan Mazlum 201811045

15/05/2022

Table of Contents

List of Figures

1. Introduction

1.1. Purpose

This project was created for the students' education. They can train themselves for anything that they want and test themselves for how they are (according to test types). They can learn languages, laws, math, physics etc. for their area of interest. They can join lecturers' tests/quizzes to learn how they are improved themselves. This project's aim is the valuable people everywhere. This project will contribute to the future of the world by educating people with high-quality education.

1.2. Scope

- The system will include a database on servers that will store all of the information about users as well as log data about users.
- The system will be built to swiftly give users with accurate responses.
- The system will continuously verify if the response supplied is adequate.
- The system will continuously check if there is a conflict in schedule.
- The system will provide communication between users.
- The system is responsible for managing account. For example, Create, Delete, Modify.
- The system is responsible for feedbacks as using survey.
- The system is not responsible about payment among users. For example, Company-Student internship agreement.
- The system is responsible for creation of environment that will be used between users. For example, Online Classroom
- The system is not responsible grading between users.
- The system will have a voting mechanism for students' answers(results), students' works in internship, and giving a vote lecturer by students which will provide evaluation for each of them.

1.3. Product Overview

1.3.1. Product Perspective

1.3.1.1. System Interfaces

1.3.1.2. User Interface

1.3.1.3. Hardware Interface

1.3.1.4. Software Interface

1.3.1.5. Communication Interface

1.3.1.6. Memory Constraints

1.3.2. Glossary

- **Auditory, Visual, Linguistic, and Logical:** These represent different types of tests. An effort is made to produce a test suitable for each student. That is, tests/quizzes suitable for different types of intelligence such as auditory, visual, linguistic, and logical will be produced by the lecturer. This is a new generation method.
- **Class**: Classes, on the other hand, are like short seminars on current topics that are not continuous like courses. Courses are divided on this subject. It may also contain materials such as courses.
- **Course**: Courses are courses where people can get information about the subject, they want through the materials uploaded by the lecturer.
- **Grade**: It represents the test/quiz score taken by the student according to the results he/she has uploaded.
- **Manage**: It is involved in most parts of the project. It means only one thing. Indicates that it hosts more than one job. For example, manage test/quiz means that you can both upload and delete a quiz on the system.
- **Podcast**: A podcast is a music-like document or audio recording that the student must upload in response to questions. It is a new generation method.
- **Result**: It represents the answers given to the questions in the tests. It does not mean the test result or answer. It represents the answer given by the student.

1.3.3. User Characteristics

There are four types of users in this project. One of them is students which most of the people in this project will use. The second one is the lecturer that is the content producer. They will produce content for students. The third one is the company users, that user is offering internship opportunities for students. And the last one is admins, who will control them.

Firstly, most of the users of the project we created will be students. Students will train themselves via these contents and get some results for their education. And get some internship opportunities for their career life. They will get a certificate if they finish the classes. They can use them in their careers. Students are the most valuable part of this project, so we care about their education, and we will get responses for improving it for them.

The second user is the lecturers, who are going to produce courses, quizzes/tests for students. And they will be always controlled by our admins for the students because they can make and uploads some trash content. Lecturers are the valuable part cause if they will not appear in this project, this project will be un-useful cause there was no content without their courses.

Another type of user is the company users, who will be offering an internship for students. They can also find good employees via these internships. When they offer an internship for students, all types of students they choose, send a notification to students. They will control by our admins also for the unnecessary and trash content for our users.

In a nutshell, the system's admin is going to control the contents for the system's users. S/he will remove some users, contents, etc. for the project positively. If S/he will not be here this project will go negatively. Our admin works for our users.

1.3.4. Overview of Functional Requirements

The user must first register into the system. For registration, name, surname, age, telephone, and user preference (lecturer or student or company) must be entered. After registration, different operations can be performed according to the preferred user type.

In the Student role, the user can enroll in courses, internships, and online classes. He can search for classes, courses, and internships in the desired content from the system and view their contents. It can also view

the profiles of users who posted courses and internships. Afterward, s/he can choose different options (auditory, visual, reading) according to his/her wishes for the exams s/he will attend in the course or internship. They can upload their answers to the system with the podcast method, which is an innovative method.

In the Lecturer role, the user can add courses to the system and delete the added course. It can examine the students registered in the course as well as delete their enrollment from the course. Courses should include quizzes/tests of different types (auditory, visual, reading) for students to try out. They can also create online classes. The difference between these courses and classes is that Classes are hourly and instantaneous.

The user in the role of the Company should add an internship to the system. Students enrolled in these internships can both delete and add new users. Tests and applications should be uploaded to the internships so that students can try themselves as if they were on the job site.

In the Admin role, the user can perform all the functions in other roles. User deletion, adding, adding course deletion, etc.

Briefly,

1. Student role consists of all the features that user can do, which are:
    - Signing up to the SmartEdu (The system must allow the students to self-register to courses, classes, or internships available in the course, classes, or internships catalog of system.)
    - Signing in to the SmartEdu
    - Signing out of the SmartEdu
    - Course Enrollment (The system must give opportunity to student course registration.)
    - Test/Quiz Participation (The system must manage test participation by controlling completed or not courses by student.)
    - Join Online Class (The system must give opportunity to student online class registration.)
    - View Course Content (The system must receive and upload content of classes, courses.)
    - View Previous Courses (The system must provide rich text editor with html capabilities to see previous content, it also applies to other content viewing options.)
    - Manage Give Answers to Quiz/Test (The system must allow uploading the podcast content for results, answers of test or quiz in types of media from a computer or mobile device.)
    - Voting Lecturer
    - Deleting account
2. Company role consists of all the features that a worker can do, which are:
    - Signing up to the SmartEdu
    - Signing in to the SmartEdu
    - Signing out of the SmartEdu
    - Manage Internship (The system must support creating internships.)
    - Manage Duty of Work (The system must provide the company the ability to create different duty for being able to evaluate students that are registered to internship.)
    - Voting Students in internship (The system must provide the company the ability to give a vote for evaluating students' works.)
    - Deleting account
3. Admin role consists of some set of features that an admin can do and some set of features that a student, lecturer and company can do, admin features are:
    - Signing up to the SmartEdu
    - Signing in to the SmartEdu
    - Signing out of the SmartEdu
    - Manage Course
    - Viewing Courses, Students, Company, and Lecturer (The system must create and produce student, courses, company or lecturer reports based on selected data elements (For example, if lecturer is

selected, courses that lecturer created will be displayed in report) including name, address, and student identification number.)
- Manage Schedule
- Announcement (The system must alert announcement of admin to all users into system.)
- Authentication (The system must verify that a student is not enrolled in the same class or system by being twice to avoid duplicate registration.)
- Deleting account
- User Management (Delete, create, modify User) (The system must support management of access, delete, create, modify, views, functionality roles in the system such as student, lecturer, and company.)

4. Lecturer features are:
    - Signing up to the SmartEdu
    - Signing in to the SmartEdu
    - Signing out of the SmartEdu
    - Manage Quiz/Test (The system must allow changes to the test/quiz questions via new questions.)
    - Manage Course Content (The system must connect training material to a course that Lecturer uploaded to course.)
    - Manage Test/Quiz Type (Creating) (The system must provide the lecturer the ability to create different type of course for student.)
    - Create Online Classroom (The system must support creating online class.)
    - Deleting account

1.3.4. Limitations

1.4. Definitions

2. Reference

- IEEE Computer Society. (2009). IEEE Standard for Information Technology—Systems Design—Software Design Descriptions. IEEE. New York: IEEE. IEEE Std 1016
- Sommerville, I. (2016). Software engineering. Boston: Pearson Education Limited

3. Specific Requirements

3.1. External Interfaces

3.2. Analysis- UML

3.2.1. Student

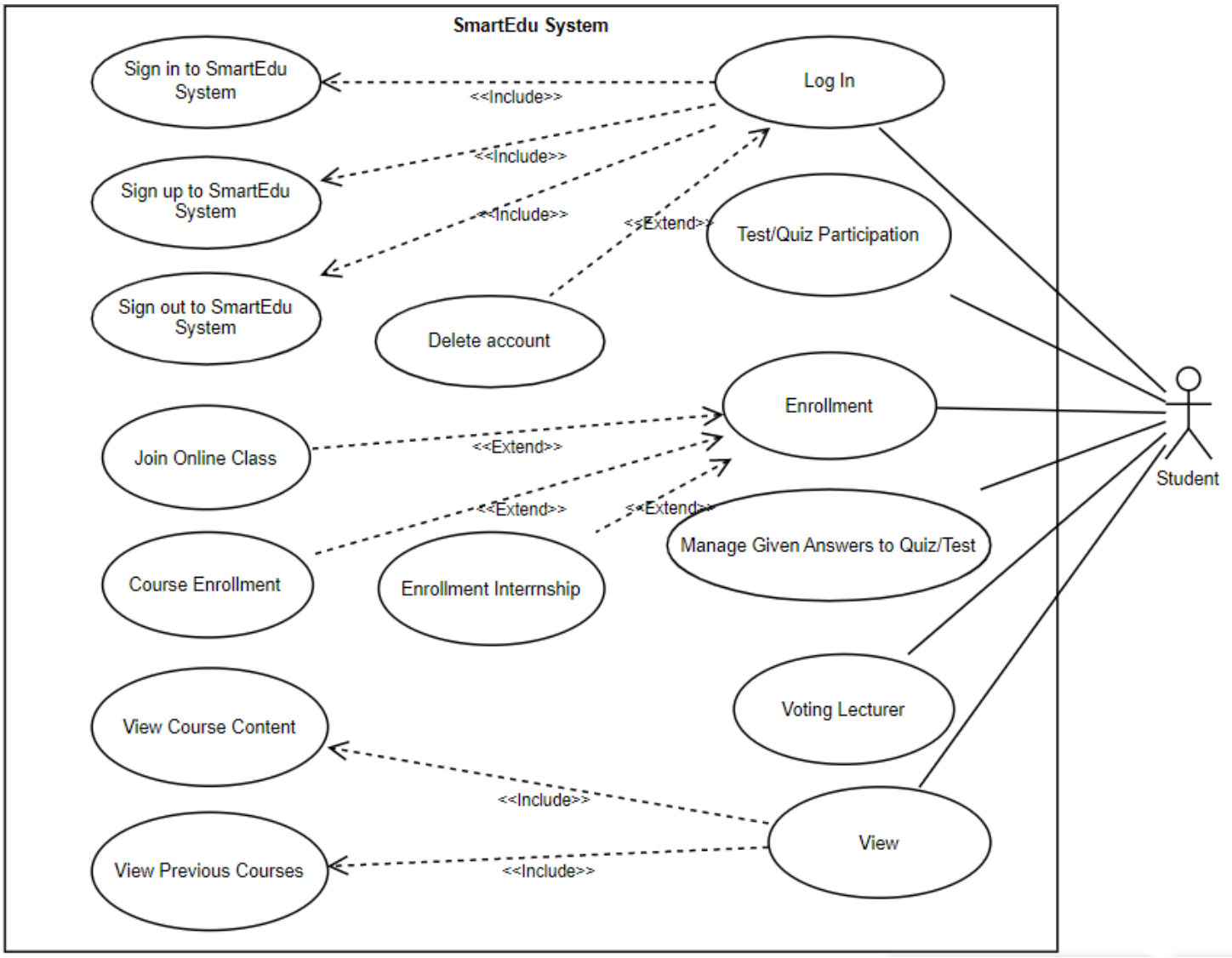

Figure 1: Student Use Case Model

Usecase Id	1
Usecase Name	Sign Up System
Description	Student signs up system to be able to take courses or attend internships.
Precondition	Student that does not exist in the system fills in the blanks with valid first and last name, mail, username and password.
Related Use Cases	-
Postcondition	End User is successfully signed up to Chorus system.
Main Flow	1. Student opens sign up page located at the system URL. 2. Student enters their first and last name, mail, username and password. Student also ticks the "I am 18 years old or older" and "I accept the End-User Agreement" boxes. 3. System verifies entered information. (A1) 4. System validates the student. 5. System successfully signs up the student into online education system.
Alternate flows	<A1 – A. If the first or last name, email, username, or password entered by the observer is/are not valid System displays "first or last name, email, username or password is not valid" error message. Student will enter valid information. Use case returns on step 2 of normal flow. B. If the email entered by the student exist in the system. System shows "The email address you entered already exists in the system. Please enter another email address or click "forgot my password" button" error message. Use case returns on step 2 of normal flow. >

Usecase Id	2
Usecase Name	Sign in System

Description	Student signs into system in order to take courses or attend internships.
Actor	Student
Precondition	Student exists in the system with a valid username and password.
Related Use Cases	UC-1
Postcondition	Student is successfully signed into system.
Main Flow	1. Student enters the system page. 2. Student click the sign in button. 3. Enter his/her own mail and password. 4. System verifies entered information. (A1) 5. Student sign into system.
Alternate flows	<A1 – If the mail and/or password entered by the Student is not valid Google system displays "Invalid mail address or password" error message. Use case returns on step 2 of normal flow. >

Usecase Id	3
Usecase Name	Sign Out
Description	Student signs out of Google system in a successfully way.
Actor	Student
Precondition	Student must be signed into system.
Related Use Cases	UC-1, UC-2
Postcondition	Student is successfully signed out the online education system.
Main Flow	1. Student clicks the button of sign out and request sign out to system. 2. System check information. (A1) 3. Student signs out. 4. System return student to sign in page.
Alternate flows	<A1 – System cannot create session to sign out. System automatically returns to sign out page again. >

Usecase Id	4
Usecase Name	Delete Account
Description	Student deletes his/her account from the system.
Actor	Student
Precondition	Student must be registered in system.
Related Use Cases	UC-1, UC-2
Postcondition	Student deletes successfully his/her account.

Main Flow	1. Student clicks the button of delete and request delete account from system 2. System check information of session. (A1) 3. System deletes account of Student. 4. System returns student to sign up page.
Alternate flows	<A1 – System cannot create session to delete account. System automatically returns to delete page again. >

Usecase Id	5
Usecase Name	Course Enrollment
Description	Student can choose between the courses in the system and can register for the courses in the system to learn something about the subject he/she wants.
Actor	Student
Precondition	Student must sign in to system and must be in course list page.
Related Use Cases	UC-2
Postcondition	Student enrolls successfully in course.
Main Flow	1. Students click the button of course list. 2. Student filters courses according to what he/she wants. (A1) 3. Student can check content of course. 4. Student request to enroll course from system. 5. System check session information. (A2) 6. Student enrolls the course. 7. Return enrollment page.
Alternate flows	<A1- Student may not find course that student wants. A. System can provide opportunity of new course to student. B. Student search new course. > <A2 – System cannot create session to enroll course. System automatically returns to courses page again. >
Usecase Id	6
Usecase Name	Test/Quiz Participation
Description	There are tests created by the teachers in the system. Students can solve them according to their own development and wishes. For example, a student with strong auditory ability can choose

	this option. Also, they can upload podcasts to solutions.
Actor	Student
Precondition	Student must sign into system. Student must enroll the course. Student must finish the course or part of the course. Student must open the test page.
Postcondition	Student finished test/quiz successfully.
Related Use Cases	UC-1, UC-2,UC-5
Main Flow	1. Student enters the course page. 2. Student clicks test/quiz participation part and requests starting quiz/test to system. (A1) 3. System tests session state. (A2) 4. System returns test/quiz page. 5. Student can start quiz/test.
Alternate flows	<A1 – The student may be trying to take the exam for the course that he is not registered for. Therefore, the system does not allow this and the student returns to the course page. > <A2 – System cannot create session to quiz/test participation. System automatically returns to course page again. >

Usecase Id	7
Usecase Name	View Course Content
Description	The student can see the course content they want. You can choose the course accordingly.
Actor	Student
Precondition	Student must sign into system. Student must be in the course page.
Related Use Cases	UC-2
Postcondition	Student viewed successfully course content.
Main Flow	1. Student enters the course page. 2. Student clicks view button of course and requests to view course content to system. 3. System checks session state. (A1) 4. System returns course content that is selected by student. 5. Student can view course content.
Alternate flows	<A1 – System cannot create session to quiz/test participation. System automatically returns to course page again. >

Usecase Id	8
Usecase Name	View Previous Courses

Description	The students can see the courses students finished or took. You can choose the course to see content (grades, uploading, course content, videos, podcasts, etc.), what students did in the course.
Actor	Student
Precondition	Student must sign into system. Student must be in the view previous course page.
Related Use Cases	UC-1, UC-2,UC-5,UC-6
Postcondition	Student viewed successfully course content that he/she studied.
Main Flow	1. Student enters the view previous course page. 2. Student clicks view button of view course and requests to view course content to system. 3. System checks session state. (A1) 4. System returns course content that is selected by student. 5. Student view course content.
Alternate flows	<A1 – System can not create session to quiz/test participation. System automatically returns to course page again. >

Usecase Id	9
Usecase Name	Join Online Class
Description	Apart from the courses, students can also attend the online classes opened by the lecturers. These classes are not like courses. It doesn't include things like quizzes/tests, it's more like a seminar.
Actor	Student
Precondition	Student must sign into system. Student must be in the class page. Lecturer must open the online class.
Related Use Cases	UC-1, UC-2
Postcondition	Student joined successfully class.
Main Flow	1. Student enters the class page. 2. Student looks at the classes that lecturers opened by online. 3. Student clicks button of join class and requests to system. 4. System checks session state. (A1) 5. System returns class session that is selected by student.

	6. Student watch lecture in class.
Alternate flows	<A1 - The system checks the status of the class. A. If the class is full, the student cannot enter the class and the student returns to the view class page. B. If the class ends when the student joins, the student returns to the view class page again. C. If the system cannot enter the class, if the system crashes, the student returns to the view class page again. >

Usecase Id	10
Usecase Name	Manage Give Answers to Quiz/Test
Description	Students are required to answer these when taking tests of the courses they are attending. The podcast method, which is a new method, is used in this answering system. Students answer the tests using their own voices.
Actor	Student
Precondition	Student must sign into system. Student must enroll the course. Student must finish the course or part of the course. Student must open the test page.
Related Use Cases	UC-1, UC-2,UC-5,UC-6
Postcondition	Student uploads successfully the answers by being podcast.
Main Flow	1. Student enters the course page. 2. Student clicks test/quiz participation part and requests starting quiz/test to system. (A1) 3. System tests session state. (A2) 4. System returns test/quiz page. 5. Student can starts quiz/test. 6. Student upload file of podcast to finish test/quiz. 7. System take podcast and control types. (A3) 8. System return message. 9. Student finishes the quiz/test.

Alternate flows	<A1 – The student may be trying to take the exam for the course that he is not registered for. Therefore, the system does not allow this and the student returns to the course page. > <A2 – System can not create session to quiz/test participation. System automatically returns to course page again. > < A3 - The system checks the status of the file. A. The student may upload the wrong file extension while uploading the answers to the questions. The system sends an error message to it and directs the student back to the test page. B. Students can upload blank file. The system sends an error message to it and directs the student back to the test page. C. The student can upload the answer as a podcast, but the system may not detect the audio. The system sends an error message to it and directs the student back to the test page. >

Usecase Id	11
Usecase Name	Enrollment Internship
Description	Student can choose between the internships in the system and can register for the internships in the system to learn something from real worker about the subject he/she wants.
Actor	Student
Precondition	Student must sign in to system and must be in internships list page.
Related Use Cases	UC-1, UC-2
Postcondition	Student enrolls successfully in internships.
Main Flow	1. Students click the button of internships list. 2. Student filters courses according to what he/she wants. (A1) 3. Student can check content of internships. 4. Student request to enroll internships from system. 5. System check session information. (A2) 6. Student enrolls the internships **7.** Return enrollment page.

Alternate flows	<A1- Student may not find internships that student wants. C. System can provide opportunity of new course to student. D. Student search new internships. > <A2 – System cannot create session to enroll internships. System automatically returns to internships page again. >

Use case id	12
Use case name	Voting Lecturer
Description	Lecturer will be graded by student who took courses of lecturer.
Actor	Student, Lecturer
Precondition	Students has to be registered and completed the course whose lecturer published.
Related Use Cases	UC-1, UC-2,UC-5,UC-6
Postcondition	Lecturer is graded by student successfuly.
Main Flow	1. Students click the button of course. 2. Student filters courses that s/he completed. (A1) 3. Student can reach the lecturer information of this course. 4. Student request to give a vote to lecturer from system. 5. System check session information. (A2) 6. Student gave a vote. 7. Student returned course page via system.
Alternative Flow	<A1- Student may not find course that s/he completed that student wants. A. System can provide again course information. > <A2 – System cannot create session to give a vote. System automatically returns to course page again. >

3.2.2. Lecturer

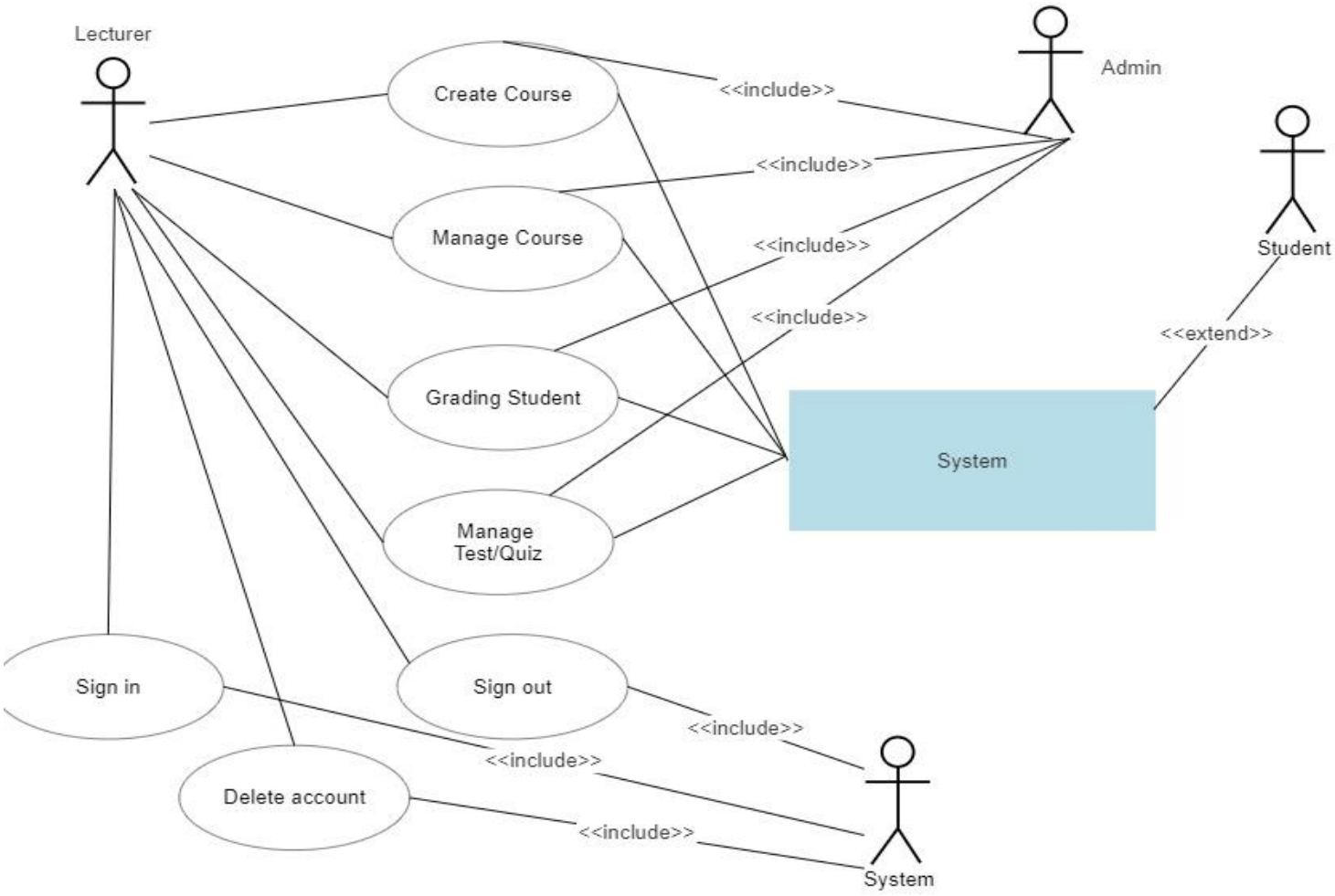

Figure 2: Lecturer Use Case Model

Use case id	1
Use case name	Creating a course.
Description	Lecturer is going to create a course.
Actor	Lecturer, Student, Admin
Precondition	Lecturer has to be approved by admin.
Postcondition	Lecturer will be created a course and students are going to get it.
Main Flow	1. Lecturer will be going to give information about her course to admin. 2. Admin will going to approve the course. <A1> 3. Course is now on system. 4. Students can get the course.

Alternative Flow	<A1> If admin does not approve the course. Lecturer has to be create again.

Use case id	2
Use case name	Managing a course.
Description	Lecturer is going to manage a course add something or change anything.
Actor	Lecturer, Admin
Precondition	Lecturer has to be giving a course
Postcondition	Lecturer changed or added something to a course.
Main Flow	1. Lecturer open a ticket to admin to change or add something to his course. 2. Admin will approve it. <A1> 3. Something changed or added to a course. 4. Student get an announcement about the information what is changed.
Alternative Flow	<A1> If admin will not approve it. Lecturer has to be send a new ticket again.

Use case id	3
Use case name	Grading a student
Description	Students will be graded before the course.
Actor	Student, Lecturer, Admin
Precondition	Students has to be getting the course that lecturer.
Postcondition	Students will be graded.
Main Flow	8. System will be calculate the quiz/homework data for each student. 9. System send the grades to the lecturer. 10. Lecturer will be approve the grades and send the admin to approve. <A1> 11. Admin will be approve the grades.<A2> 12. When lecturer and admin will approve grades, system is going to send the grades to the student.
Alternative Flow	<A1> If lecturer not approve the grades, grades will be sent to admin to look and fix it. <A2> If admin not approve grades, grades will be sent to lecturer to look and fix it.

Use case id	4
Use case name	Manage Test/Quiz

Description	Creating a test/quiz for students.
Actor	Student, Lecturer, Admin
Precondition	Student has to be in the course.
Postcondition	Student can access the test/quiz.
Main Flow	1. Lecturer has to be open a ticket for creating a test/quiz. 2. Admin will approve her ticket. 3. Lecturer send a copy of test/quiz to the admin. 4. Admin is going to approve the test/quiz. <A1> 5. Test/Quiz is published in the course content.
Alternative Flow	<A1> If admin not approve the test/quiz, lecturer has to be change the test/quiz or not going to publish it.

Use case id	5
Use case name	Sign in as a lecturer.
Description	Lecturer will sign in to the system.
Actor	Lecturer, System
Precondition	Lecturer has not to be signed in.
Postcondition	Lecturer will sign in.
Main Flow	1. In the menu lecturer will be click sign in as a lecturer. 2. Lecturer will write her username and password. 3. System checks on her username and password on database. <A1> 4. System approved her to sign in.
Alternative Flow	<A1> If username and password does not match system will be wants to write her username and password again.

Use case id	6
Use case name	Sign up as a lecturer.
Description	Lecturer will sign up to the system.
Actor	Lecturer, System, Admin
Precondition	Lecturer has not to be signed up the system.
Postcondition	Lecturer signed up the system.
Main Flow	1. Lecturer is going to click create an account as a lecturer button. 2. System will send the creating account page to her screen. 3. Lecturer will fill the information page. 4. System is going to send the information to

	the admin. 5. Admin approves him. <A1> 6. Lecturer is now signed up the system.
Alternative Flow	<A1> If admin does not approve him, lecturer has to fill the information page again.

Use case id	7
Use case name	Delete account as a lecturer.
Description	Deleting account.
Actor	Lecturer, System
Precondition	Lecturer has to be an account.
Postcondition	Lecturer's account has been deleted.
Main Flow	1. Lecturer will click the manage account page. 2. Lecturer will click delete account button. 3. System asks to "Are you sure?". 4. Lecturer will click "Yes". <A1> 5. Lecturer's account has been deleted.
Alternative Flow	<A1> If lecturer not click the yes button, system will reload the managing account page.

3.2.3. Company

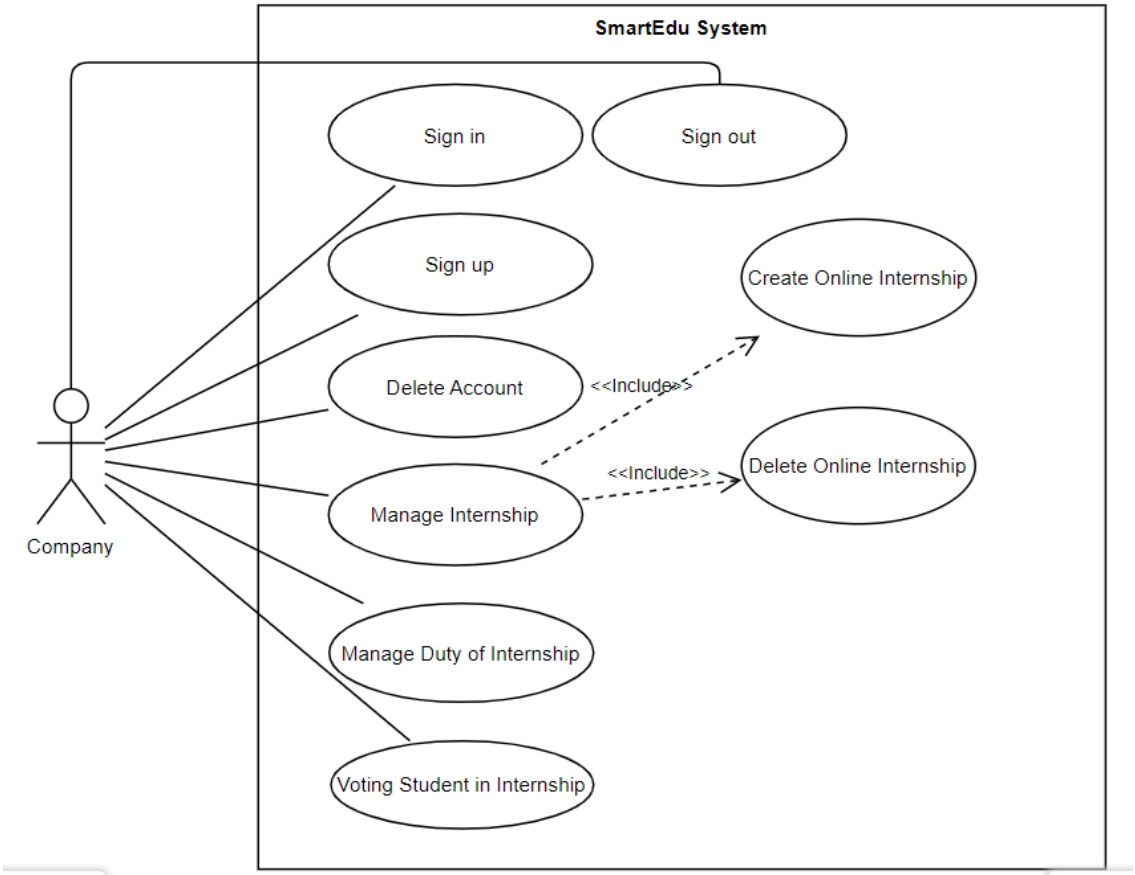

Figure 3: Company Use Case Model

Usecase Id	1
Usecase Name	Manage Internship
Description	It is a form of manage that includes operations such as deleting internships or creating internships that users (Company) have added.
Actor	Company
Precondition	Company must sign in to system. Company must be in the Internship page.
Postcondition	Company deleted successfully Internships that he/she created or Company created successfully

	Internships.
Main Flow	1. The user logs into the system. 2. Goes to the Internship page. 3. From the Internship page, he chooses the manage operation (create, delete) he wants to do. (A1) 4. The user who chooses the Create process creates an internship according to the content he wants and according to the rules presented by the system. 5. After creating it, it sends a request to the system to complete the operation. (A2) 6. The course is successfully created by the system.
Alternate flows	<A1 – 1. The user presses delete button. 2. He chooses the internship he wants to delete from among the internships he has created. 3. It sends the internship deletion request to the system. 4. If everything goes smoothly, the internship is successfully deleted, no longer accessible in the system. > <A2 – System can not create session to create Internship. System automatically returns to Internship page again. >

Usecase Id	2
Usecase Name	Manage Duty of Internship
Description	Users can upload tests and tasks to the courses added by the user (Company) so that students can test themselves and learn how they encounter problems in business life.
Actor	Company
Precondition	Company must sign in to system. Company must be in the Internship page.
Postcondition	Company created successfully works/duties for Internship that he/she created.
Main Flow	1. The user logs into the system. 2. Goes to the Internship page. 3. He chooses from the courses he has added. 4. It sends a request to the system to add work to this internship according to the

	rules determined by the system. It also sends the job content with the request. (A1) 5. If this request is successful, the job content is successfully saved to the internship in the system. 6. The course is successfully created by the system.
Alternate flows	<A1 – System can not create session to create Internship's duties. System automatically returns to Internship page again. >

3.2.4. Admin

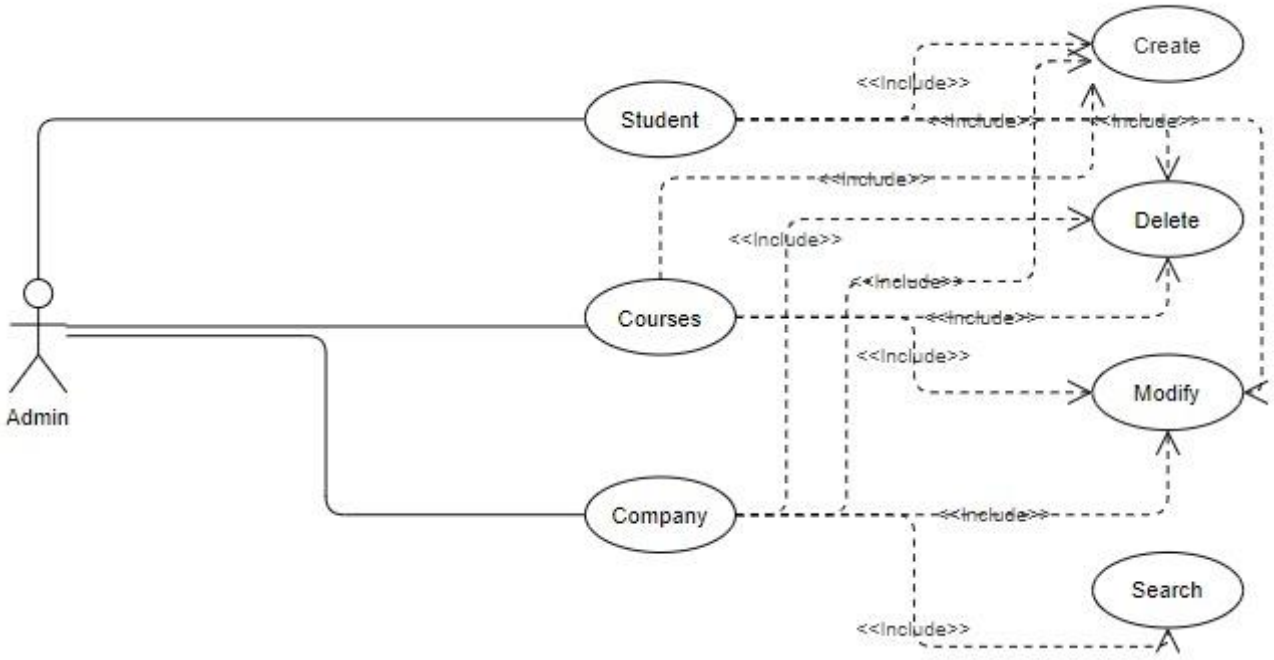

Figure 4: Admin Use Case Model

Usecase Id	1

Usecase Name	Authentication
Description	The users must be authenticated to see or change entities of the system.
Actor	Admin
Precondition	System must be open
Postcondition	Successfully login to the system
Main Flow	1-System displays login page. 2-User enters his credentials. 3-System checks credentials. <A1> 4-System shows dashboard
Alternate Flow	<A1-Check fails system shows login page again>

Usecase Id	2
Usecase Name	Create User
Description	Admin creates a new user.
Actor	Admin
Precondition	Admin must be logged in to system.
Postcondition	New user is created.
Main Flow	1-Admin gives information to system. 2-System automatically creates profile of user. <A1> 3-Admin authenticates user and approves profile. <B1> 4-After profile approval system save user to database.
Alternate Flow	<A1- Because lack of information system can't create profile. System send message "Please fill the required area"> <B1-Authentication fails, and user not approved. System informs the user about approval. Creation must be done again.>

Usecase Id	3
Usecase Name	Search User
Description	Admin searches a user.
Actor	Admin

Precondition	Admin must be logged in to system.
Postcondition	Searched user is shown in page.
Main Flow	1-Admin gives user information to system. 2-System selects user from database. <A1> 3-Database gets the searched user.
Alternate Flow	<A1- Because user doesn't exist in database. System sends message "User not found">

Usecase Id	4
Usecase Name	Delete User
Description	Admin deletes a user.
Actor	Admin
Precondition	Admin must be logged in to system.
Postcondition	Message Return: "User deleted."
Main Flow	1-Admin gives user information to system. 2-System queries select user from database. <A1> 3-Database gets the searched user. 4-System checks, and queries delete user.
Alternate Flow	<A1- Because user doesn't exist in database. System sends message "User not found">

Usecase Id	5
Usecase Name	Modify User
Description	Admin modifies a user.
Actor	Admin
Precondition	Admin must be logged in to system.
Postcondition	Message Return: "User information updated."
Main Flow	1-Admin gives user information to system. 2-System queries select user from database. <A1> 3-Database gets the searched user. 4-System checks, and queries update selected information of user in database.
Alternate Flow	<A1- Because user doesn't exist in database. System sends message "User not found">

Usecase Id	6

Usecase Name	Create Learning Courses
Description	Admin creates new course.
Actor	Admin
Precondition	Admin must be logged in to system.
Postcondition	Course page shown in system.
Main Flow	1-Admin gives information to system. 2-System automatically creates profile of course. <A1> 3-Admin authenticates course information and approves course. <B1> 4-After profile approval system save course to database.
Alternate Flow	<A1- Because lack of information system can't create profile. System send message "Please fill the required area"> <B1-Authentication fails, and course not approved. System informs the user about approval. Creation must be done again.>

Usecase Id	7
Usecase Name	Search Learning Courses
Description	Admin searches new course.
Actor	Admin
Precondition	Admin must be logged in to system.
Postcondition	Searched course page shown in system.
Main Flow	1-Admin gives course information to system. 2-System selects course from database. <A1> 3-Database gets the searched course.
Alternate Flow	<A1- Because course doesn't exist in database. System sends message "Course not found">

Usecase Id	8
Usecase Name	Delete Learning Courses
Description	Admin deletes course.
Actor	Admin
Precondition	Admin must be logged in to system.
Postcondition	Message Return: "Course deleted."

Main Flow	1-Admin gives course information to system. 2-System queries select course from database. <A1> 3-Database gets the searched course. 4-System checks, and queries delete course.
Alternate Flow	<A1- Because course doesn't exist in database. System sends message "Course not found">

Usecase Id	9
Usecase Name	Modify Learning Courses
Description	Admin modifies course.
Actor	Admin
Precondition	Admin must be logged in to system.
Postcondition	Message Return: "Course information updated."
Main Flow	1-Admin gives course information to system. 2-System queries selects course from database. <A1> 3-Database gets the searched course. 4-System checks and queries update selected information of course in database.
Alternate Flow	<A1- Because course doesn't exist in database. System sends message "Course not found">

Usecase Id	10
Usecase Name	Announcement
Description	Admin publishes general message to all users.
Actor	Admin
Precondition	Admin must be logged in to system.
Postcondition	Message published in main page's dashboard
Main Flow	1-Admin gives message to system. 2-System checks message. <A1> 3-System saves the announcement and its date to database.
Alternate Flow	<A1-If there is a problem with message too much or invalid character system shows message of error.>

Usecase Id	11
Usecase Name	Manage Schedule
Description	Admin manages schedule.
Actor	Admin
Precondition	Admin must be logged in to system.
Postcondition	Message Return: “Schedule updated.”
Main Flow	1-Admin gives schedule information to system. 2-System checks date. <A1> 3.System save the schedule date to database. <B1>
Alternate Flow	<A1- If time or date format is invalid system sends message of error> <B1-If in the same date and time there is another course or meeting etc. system send message “Time is taken”.

3.3. Detailed Description of Functional Requirements

User Story	As a User, I must be able to delete account.
Function	Deleting Account
Description	With the data entered by the user, autentication is provided. The system deletes the account via information.
Input	Name, surname, password
Output	Confirmation message
Action	If the user data is entered correctly, the system confirms the deleting account and the user deleted the account. If there is an incorrect entry, the user is shown an error message and asked to enter again.
Requirements	The user must be registered in the system.
Pre-Condition	User being on the deletion page.
Post-Condition	Confirmation of deletion and deletion account from system.

User Story	As a User, I must be able to register or enter the system
Function	Sign Up, Sign in
Description	With the data entered by the user, autentication is provided. The user enters the system.
Input	Name, surname, password, phone number, age
Output	Confirmation message
Action	If the user data is entered correctly, the system confirms the login and the user can use the website. If there is an incorrect entry, the user is shown an error message and asked to enter again.
Requirements	User being on the login or registration page
Pre-Condition	User being on the login or registration page
Post-Condition	Successful data entry, confirmation of registration or logging into the system

User Story	As a User, I must be able to attend a course, internship or class on the website and view content of them.
Function	• Course Enrollment • Join Online Class • Join Internship
Description	The user (Student) can register for internships, online classes, and courses in the system as she wishes.
Inputs	Request to attend
Outputs	Confirmation message, Content of Course or Class or Internship
Action	The user can search for a class, course or internship according to the criteria he wants in the system. When it finds them, it can send a request to the system for registration. If a problem does not occur, the user can register with them and receive his training as he wishes. If the problem occurs, the system redirects the user back to the course, internship selection page.
Requirements	In order to perform these functions, the user role must be selected.
Pre-Condition	The user is required to register and login to the system.
Post-Condition	Successfully enrolling in a course, class or internship.

User Story	As an User (Lecturer), I must be able to delete course or change course content how I want.
Function	• Manage Course • Manage Course Content
Description	The user (Lecturer) can delete the courses I added in the system as I want. Instead of deleting courses, they can update their content.
Inputs	Request to delete course / New Information (Content) of Course
Outputs	Confirmation message, New Content of Course
Action	After the user logs in to the system, he goes to the page where the courses he has added are located. It sends a request to the system to delete the courses they want from among the courses here. In addition, by updating the content of these courses instead of deleting them, they can add new information and not victimize students.

Requirements	In order to perform these functions, the lecturer role must be selected.
Pre-Condition	The user is required to register and login to the system.
Post-Condition	Successfully deleting in a course or changing content of course

User Story	As an User (Company), I must be able to manage internship how I want (deletion, changing or etc.), and I want to add some duty to internships.
Function	• Manage Internship • Manage Duty of Work
Description	The user (Lecturer) can delete the courses I added in the system as I want. Instead of deleting courses, they can update their content.
Inputs	Request to delete internship / New Duty (like test/quiz) for internship
Outputs	Deletion Confirmation message, New Content of Duty,Work
Action	After the user enters the system, they can delete any internships they have added to the system. Apart from that, it can upload different jobs to the already existing courses to try the registered students. It can make evaluations based on the results. When any problem occurs, the system redirects the user to the internship viewing page.
Requirements	In order to perform these functions, the company role must be selected.
Pre-Condition	The user is required to login to the system.
Post-Condition	Successfully deleting in a internship or adding new duty/work to internship page.

User Story	As an User (Company), I must be able to evaluate student's work and give a vote to student.
Function	• Voting Students in Internship
Description	The user (Company) can give points and votes for the work done by the students registered for the internship.
Inputs	Vote to student's work
Outputs	Confirmation message to vote student's work
Action	The user who is logged into the system goes to the page where the internships he/she added are located. By

	entering one of these internships, he sees the list of students registered and doing business. It reviews their work and gives them points.
Requirements	In order to perform these functions, the company role must be selected.
Pre-Condition	The user is required to login to the system.
Post-Condition	Successfully giving vote to student's work in a internship

User Story	As an User (Lecturer), I must be able to add test/quiz with different types.
Function	• Manage Test/Quiz Type (Creating)
Description	The user can add different types (auditory, visual, reading) tests and quizzes to the courses in the system in order to evaluate the students.
Inputs	Request to add new test/quiz to course / New Information (Question) of Test/Quiz
Outputs	Confirmation message, New Information of Test
Action	After the user logs in to the system, he goes to the page where the courses he has added are located. The user sends a request to the system to add tests and quizzes to the selected course. After the system approves this request, the user can add any test or quiz to the system.
Requirements	In order to perform these functions, the lecturer role must be selected.
Pre-Condition	The user is required to register and login to the system.
Post-Condition	Successfully adding new informtion to a course.

User Story	As an User (Student), I must be able to view course content and completed courses by student.
Function	• View Course Content • View Previous Courses
Description	The user can review the contents of the courses in the system. Also, you can view the contents of the courses he has completed in the past, the answers he has uploaded, etc. He can inspect and see things.

Inputs	Request to view content of course or completed course by student
Outputs	Confirmation message and Content of Course
Action	After the user logs in to the system, he goes to the page where the courses are. The user can access the list of both the completed courses and all the courses in the system from this page. You can review the content as you wish. In any problem (deleting the course, failing to load the system, deleting the answers in the user's old courses, etc.), the user is returned to the page with the course list by the system.
Requirements	In order to perform these functions, the student role must be selected.
Pre-Condition	The user is required to register and login to the system.
Post-Condition	Successfully viewing course content.

User Story	As an User (Student), I must be able to attend quiz/test whose I took from system, and give a result by being podcast.
Function	• Test/Quiz Participation • Manage Give Answers to Quiz/Test
Description	The User must also complete the tests in order to complete the contents of the course he attended. A podcast download is required to complete these.
Inputs	Podcast by being result to test/quiz
Outputs	Score
Action	The user logs in to the system and goes to the course they want to complete. Login to complete the tests and quizzes included in this course. Each question must upload an answer as a podcast. By installing them, he can finish the test. After the answers are analyzed, the score is transmitted by the system as a message. In any problem, the process is interrupted, the answers are saved and the user is sent to the test login page.
Requirements	In order to perform these functions, the student role must be selected.
Pre-Condition	The user is required to register and login to the system. Also, the user must take course to attend quiz of course.
Post-Condition	Successfully adding result (podcasts) to every question and got the score.

User Story	As an User (Admin), I must be able to change status of user (delete, create etc.), also look at the information everything in system.
Function	• User Management (Delete, create, modify User) • Viewing Courses, Students, Company and Lecturer
Description	The user can access the information of every user in the system, and can see the information of all courses and classes in the system. You can also delete them from the system, update data, or use a different user, course, etc. can create.
Inputs	User Information or Course Information or Class Information
Outputs	Confirmation message
Action	After logging into the system, the user is directed to the required page according to the operation they want to do. After the user who makes a request according to the desired transaction is approved by the system, the transaction starts to be processed. After successful completion, the system returns a message of successful completion. For example, if the user wants to add a new user to the system, he must enter the user data correctly. To do this, you can go to the user manage page in the admin page.
Requirements	In order to perform these functions, the admin role must be selected.
Pre-Condition	The user is required to register and login to the system.
Post-Condition	-

User Story	As an User (Admin), I must be able to make an announcement to notify user in the system.
Function	• Announcement
Description	The user can broadcast announcements for the things he/she deems necessary, in a way that every user can see throughout the system.
Inputs	Announcement Message
Outputs	Confirmation message
Action	After the user logs in, he goes to the announcement page. Here it creates an announcement message. After completing it, it sends it to the system. The approved message is broadcast throughout the system. When an

	error occurs, the message is recorded in the system and sent back to the user announcement page.
Requirements	In order to perform these functions, the admin role must be selected.
Pre-Condition	The user is required to register and login to the system.
Post-Condition	Successfully made an announcement in the system.

User Story	As an User (Admin), I must be able to make or change an schedule to notify user in the system.
Function	• Manage Schedule
Description	The user (Admin) can assign time to all of the classes in the system to avoid conflicts.
Inputs	Time
Outputs	Confirmation message
Action	The user is sent to the required page after logging in to the system. Here, time is assigned to the desired class. However, the time of each class should be reviewed. It doesn't necessarily have to be at the same time or at different times.
Requirements	In order to perform these functions, the admin role must be selected.
Pre-Condition	The user is required to register and login to the system.
Post-Condition	Successfully made an schedule in the system.

3.4. Non-Functional requirements

- **Performance**

When sending an announcement request by admin, this will be sent to all users in the system.

- **Usability**

For the user to use the system more effectively, its design will be simpler and in a way that everyone can understand. The commands sent by the system will always be at the basic level. Contents and explanations that will create confusion will not be found in the system. People who add courses, classes, and internships will be asked not to add complicated things. Admin will be able to control them. Also, Mobility must be supported by the system (etc. phones and tablets). All current and previously supported versions of contemporary web browsers, such as Internet Explorer, Firefox, Chrome, and Safari, must be supported by the system.

- **Scalability**

The program is scalable and modifiable by the Admin.

- **Interoperability**

The program is based on taking courses from system. When student, lecturer, company information is not correctly loaded, the system should issue an exception message. The system must accept uploads of data. The system must have the ability to authenticate when user wants to sign up or in to system.

- **Maintainability**

The program is maintainable when there is any problem. In addition, external links will not be given outside the site too much. Their numbers will be kept at a reasonable level.

- **Security**

The program is secured from hacking and viruses. The system must integrate with authentication from system. The system must allow for failovers without disrupting service.

- **Manageability**

There is an admin who manages and supervises the system. The system must accept training material in the following file types:

A. Word, Excel, and PowerPoint etc.
B. PDF
C. Video formats (etc. mp4)
D. Image formats (etc. png, jpg, gif)

Also, the system must accept answers from students in the following file types:

A. Podcast (mp3 etc.)

3.5. Functional Modeling (DFD)

3.5.1 DFD Diagrams

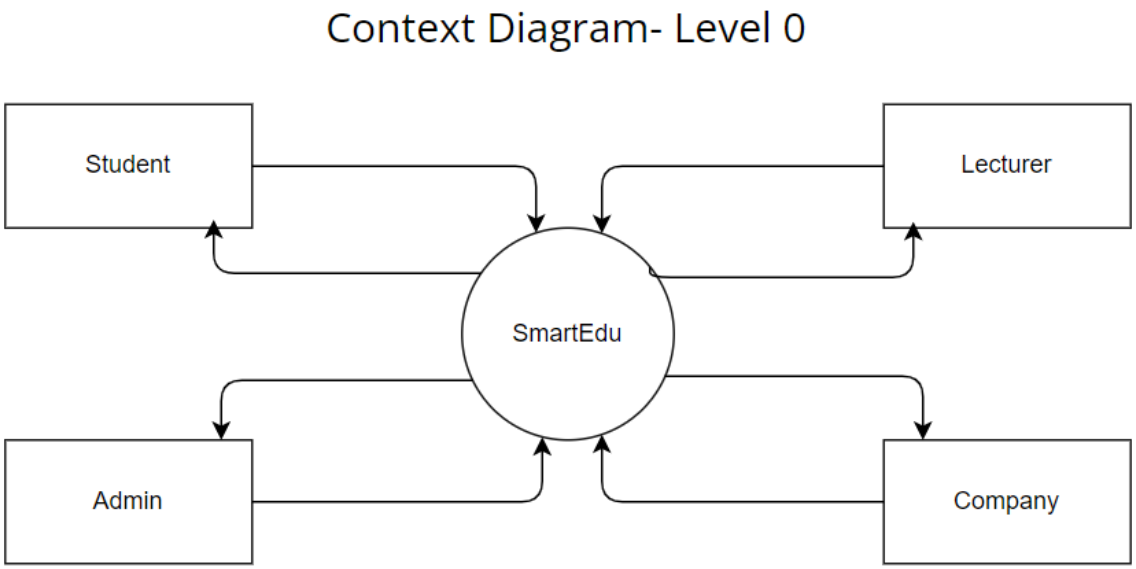

Figure 5: Context Diagram

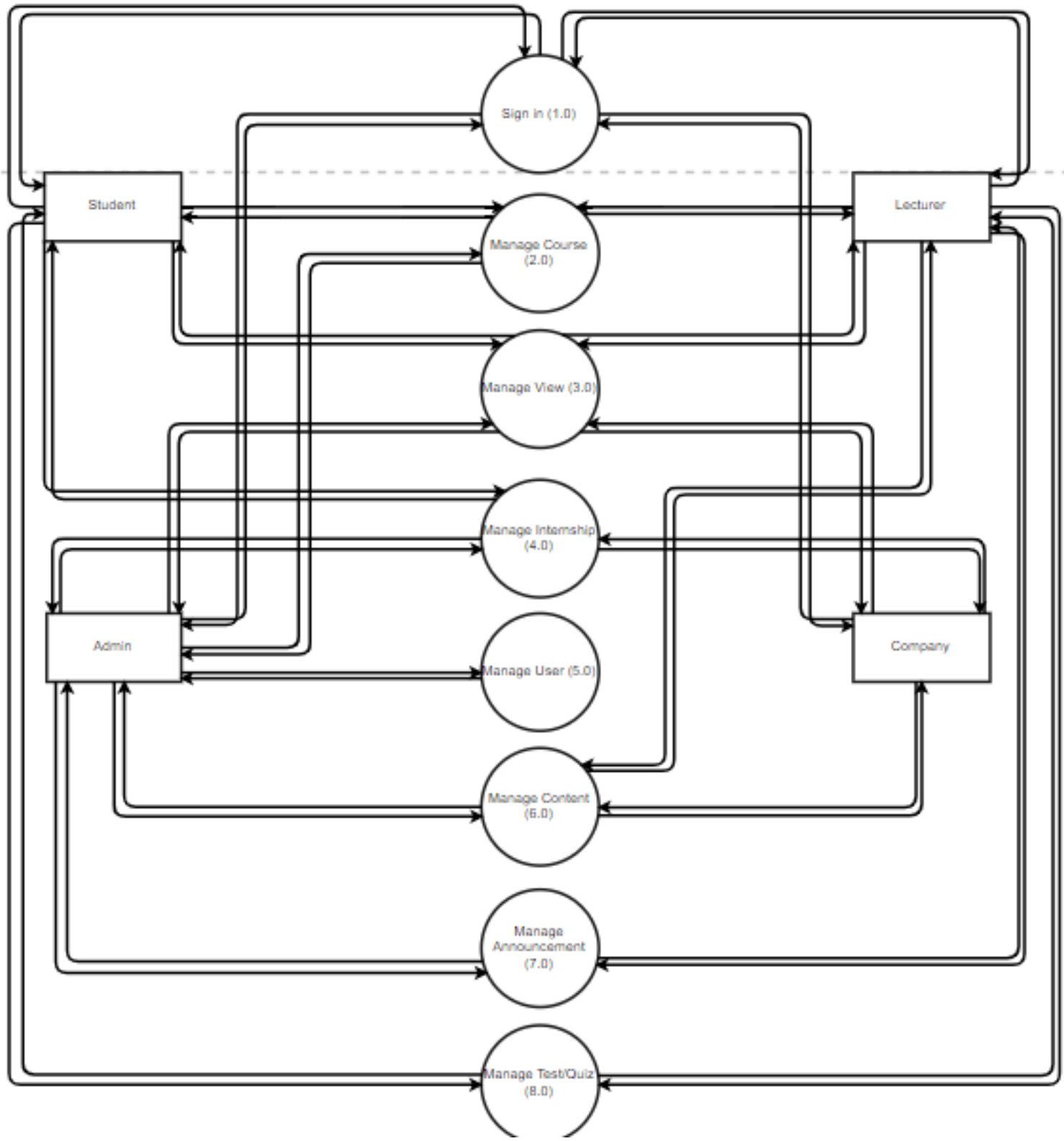

Figure 6: DFD Level-1

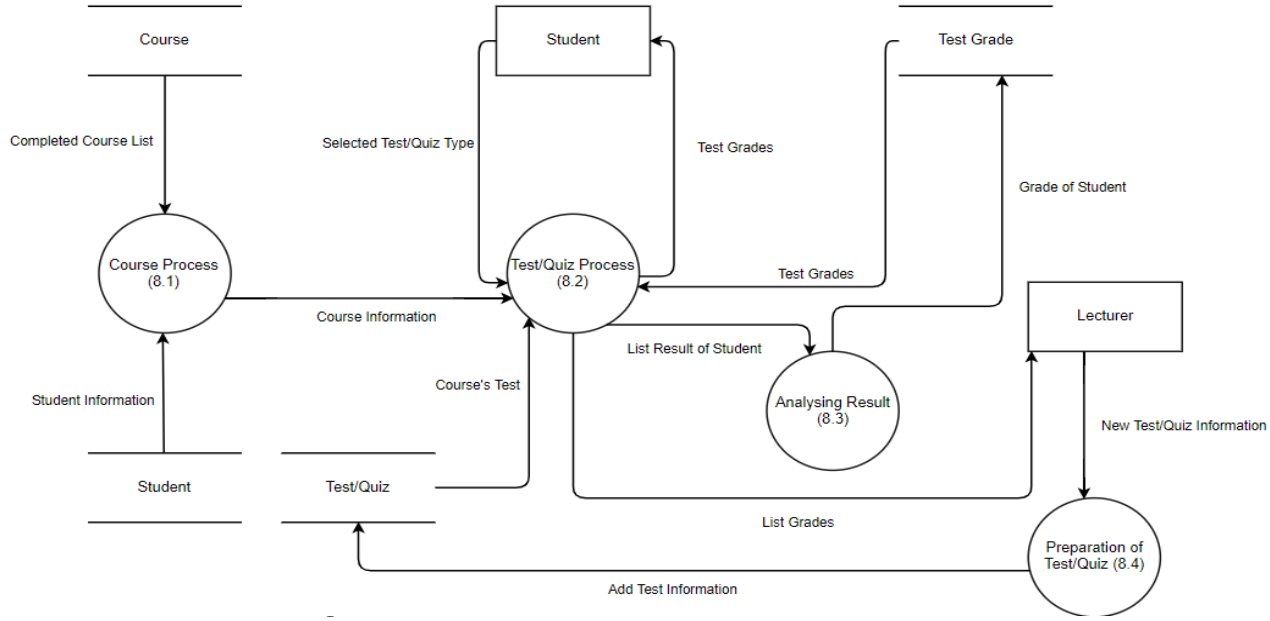

Figure 7: DFD Level-2 Manage Test/Quiz

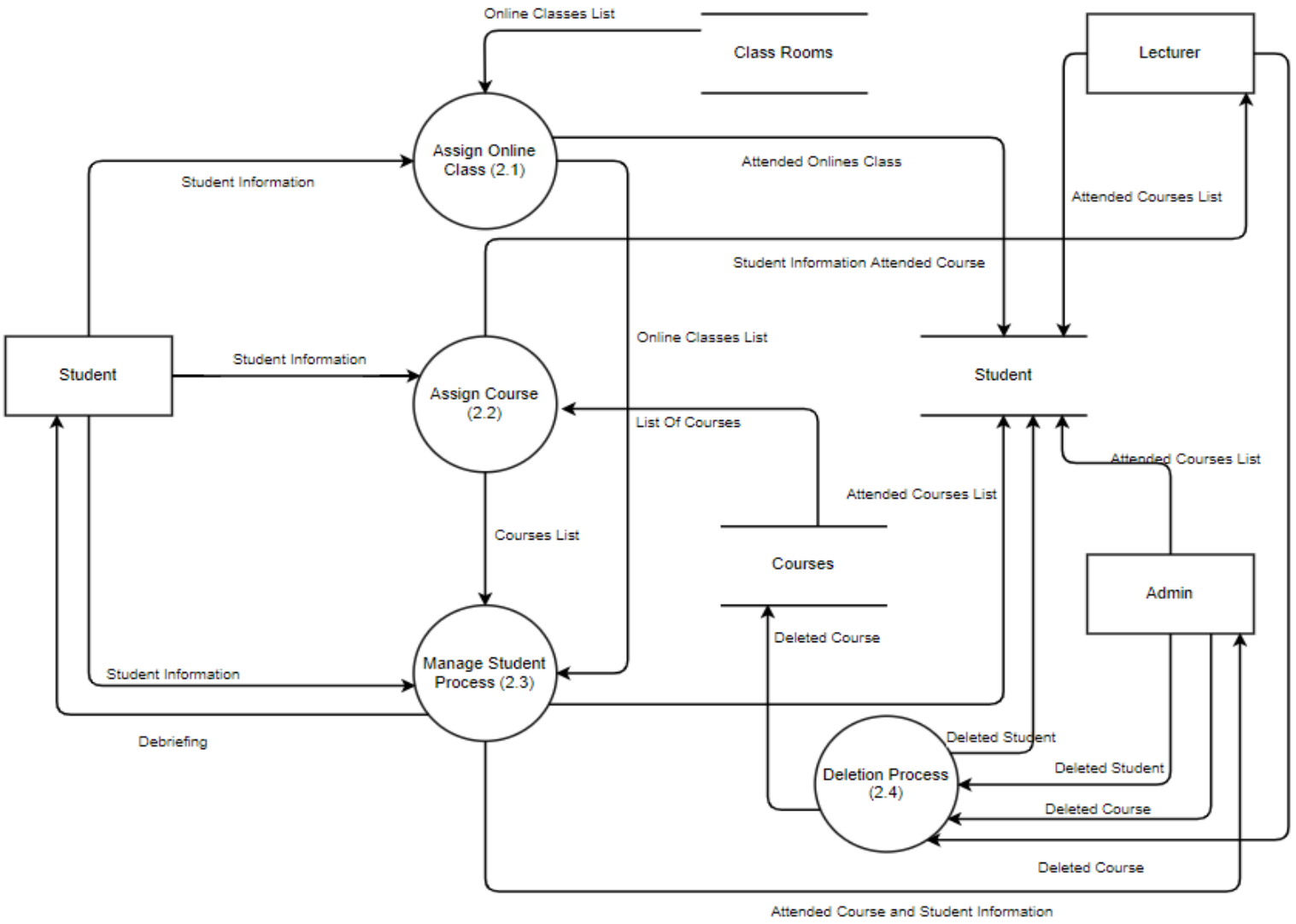

Figure 8: DFD Level-2 Manage Course

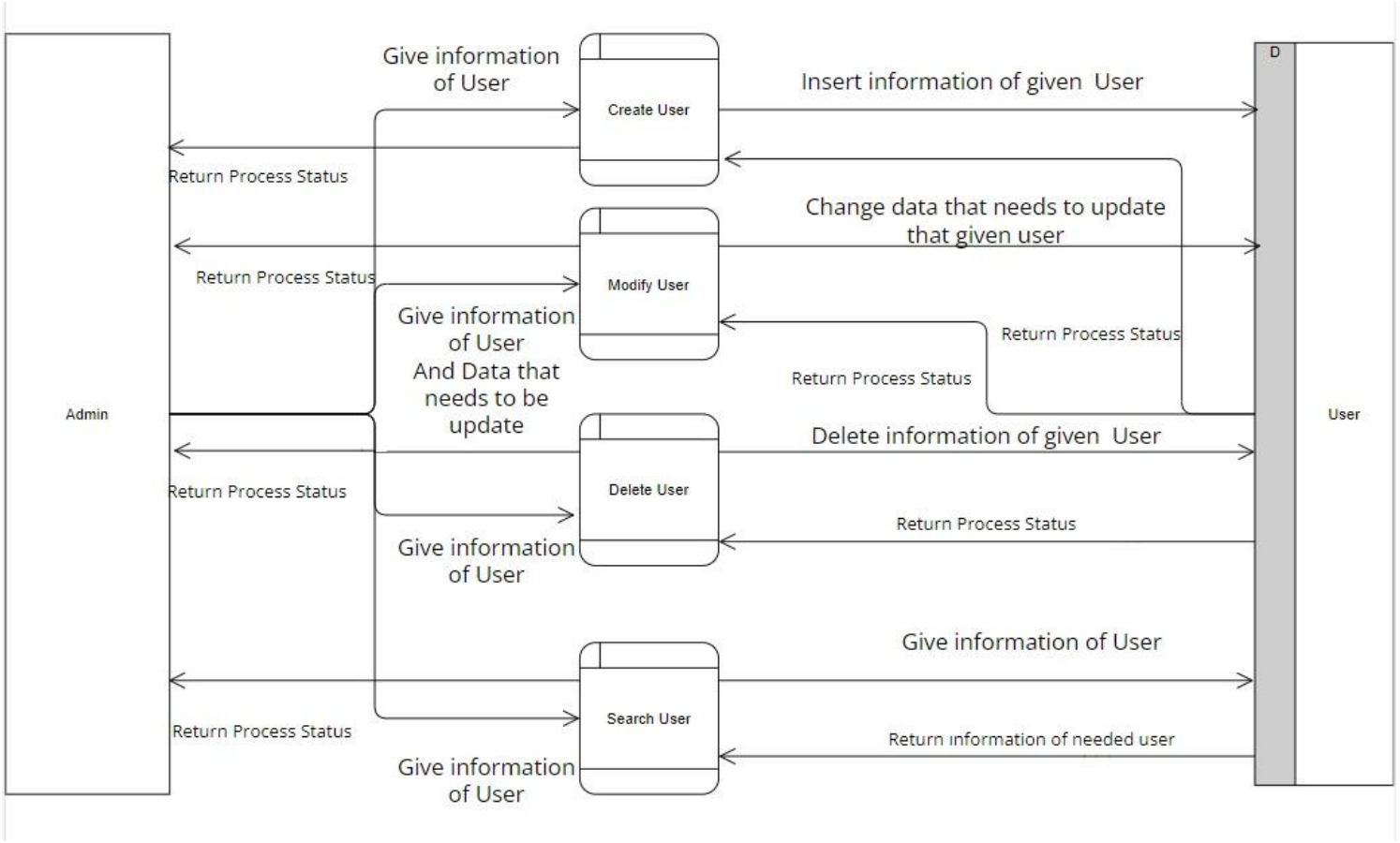

Figure 9: DFD Level-2 Manage User

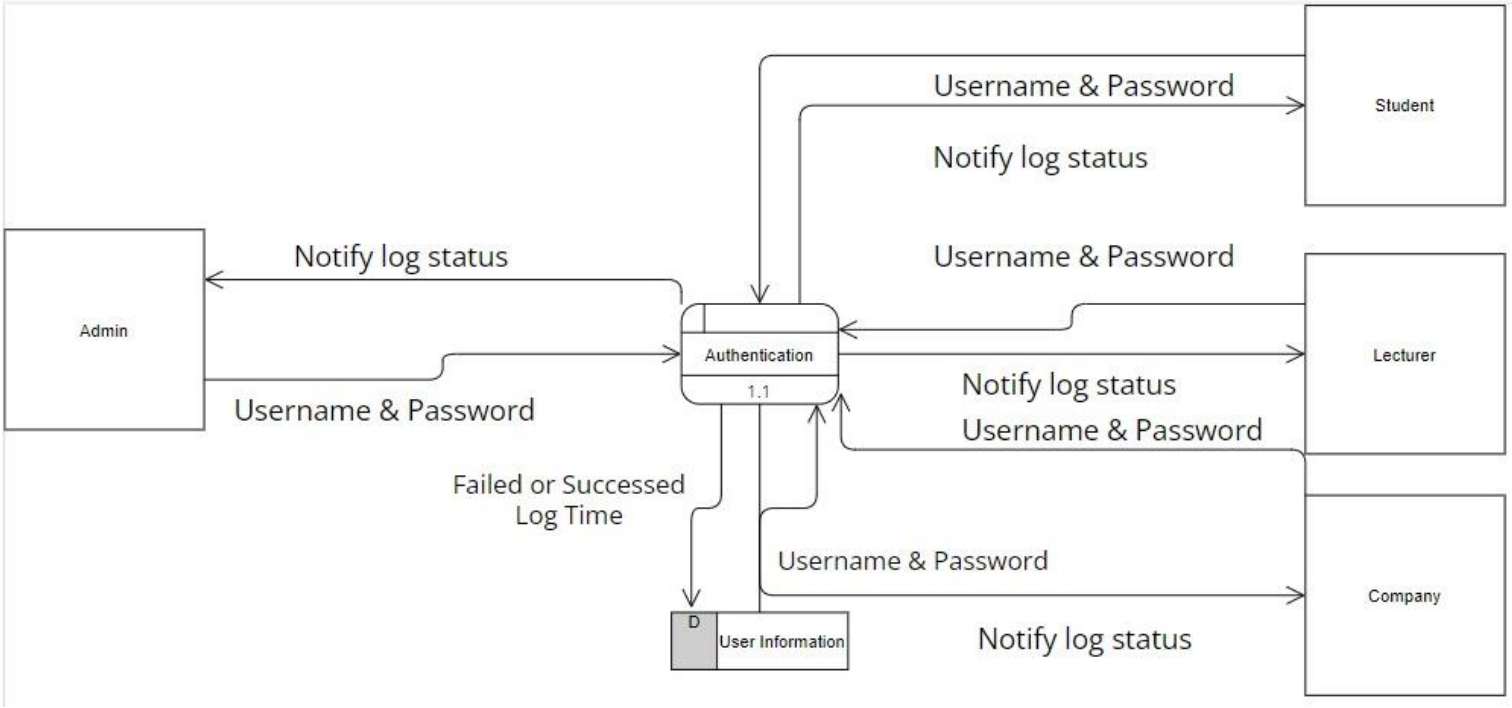

Figure 10: DFD Level-2 Sign in

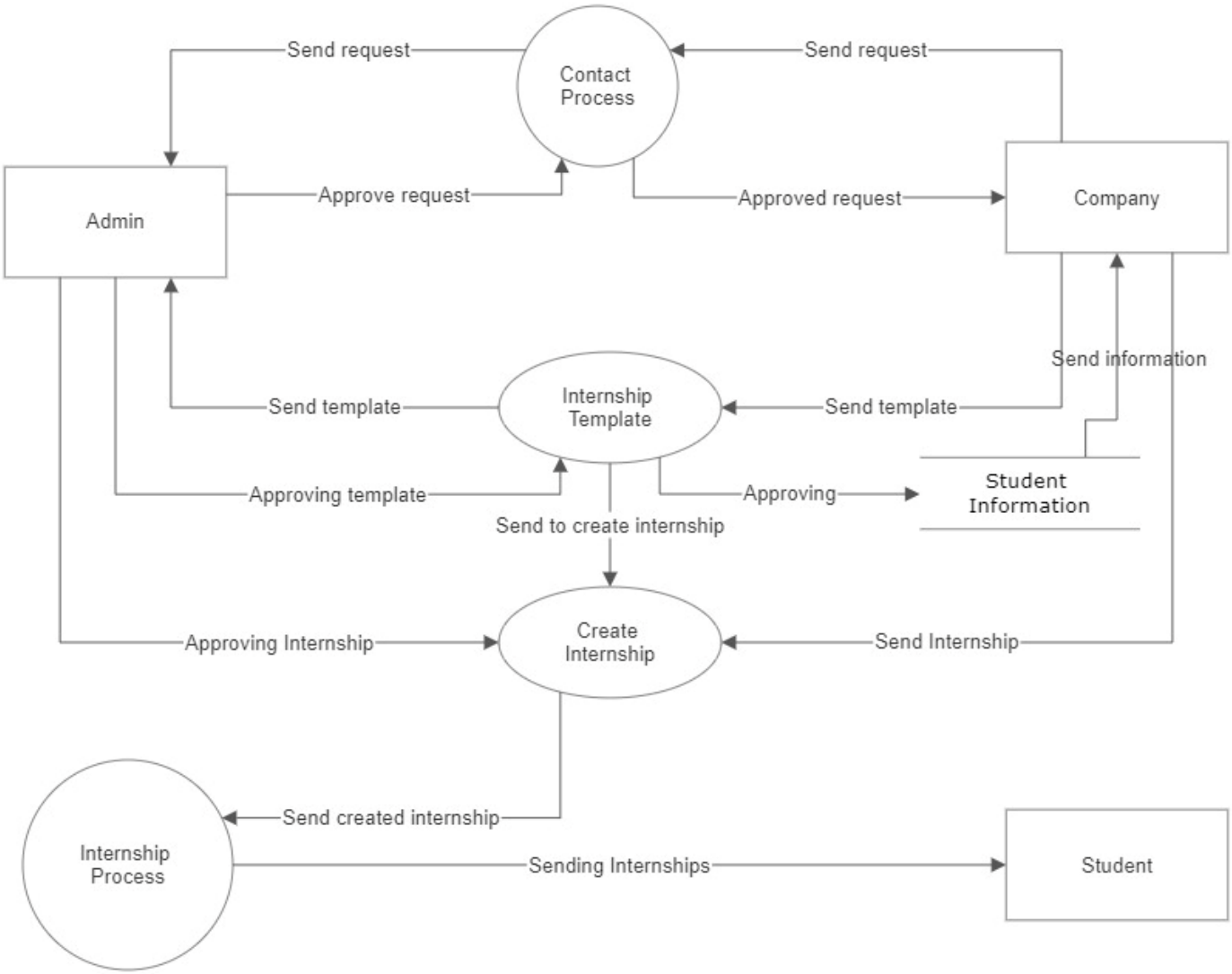

Figure 11: DFD Level-2 Manage Internship

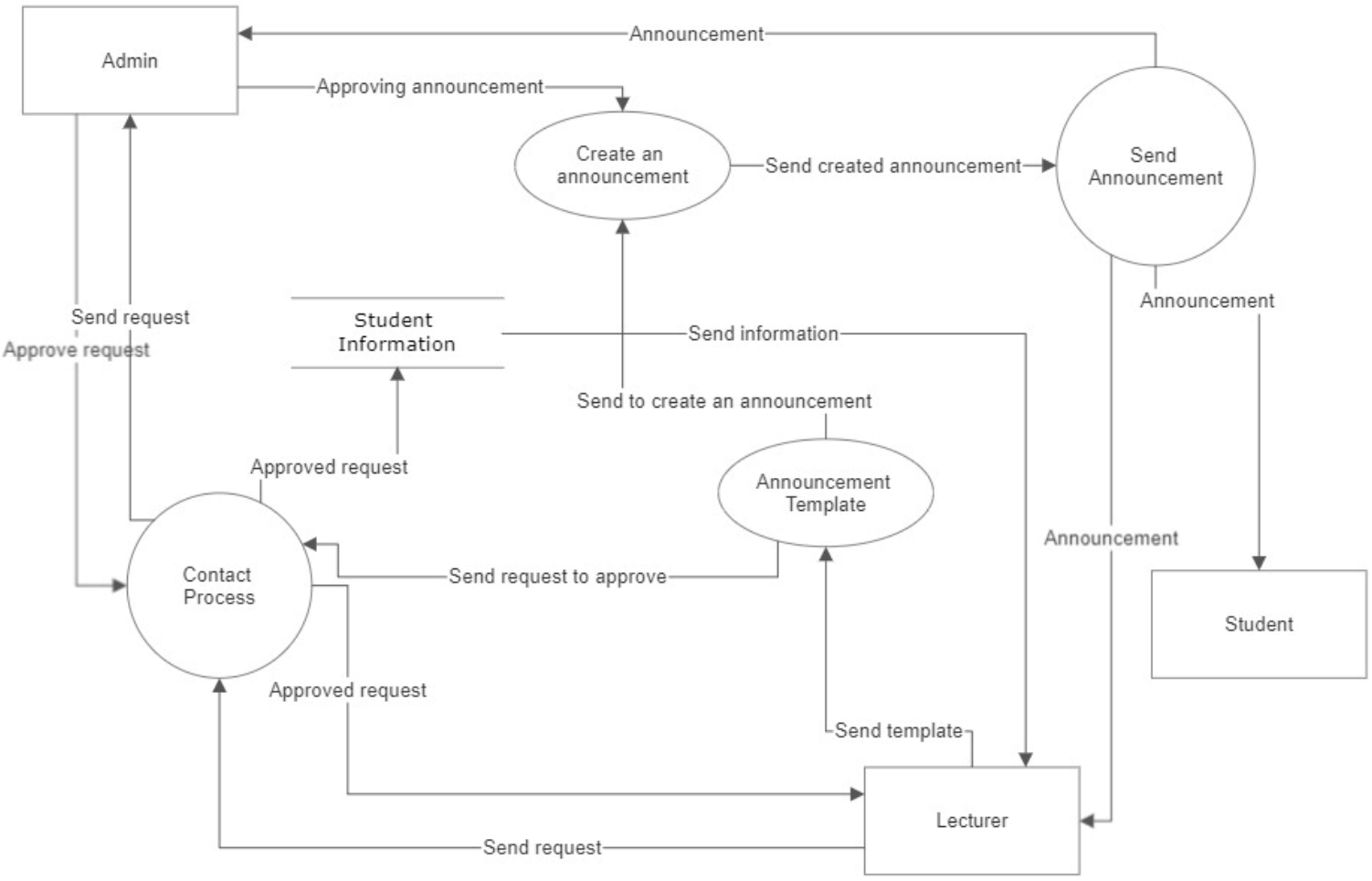

Figure 12: DFD Level-2 Announcement

3.5.2 Data Dictionary

Name:	Completed Course List
Aliases:	Finished Course, Course List
Where/How used:	To Course Process (Input) In Student Information (Output)
Description:	To get the completed courses in the system Course Information + Student_id
Format:	Text

Name:	Student Information
Aliases:	-
Where/How used:	To Student Process (Input) In Course Information with another info. (Output)
Description:	Student_name + Student_surname + Student_id
Format:	Text

Name:	Course Information
Aliases:	-
Where/How used:	To Test_Quiz Process (Input)
Description:	Course information completed by student in system or just Course Information Course_id + Course_name + CourseDetails + Lecturer_name
Format:	Text

Name:	Course's Test
Aliases:	-
Where/How used:	To Analysis Result (Input)
Description:	All courses have different tests. Course_id + Test_id + TestQuestion
Format:	Text

Name:	Course's Test
Aliases:	-
Where/How used:	To Test_Quiz Process (Input)
Description:	All courses have different tests. Course Information + Test_id + TestQuestion
Format:	Text

Name:	List Result of Student
Aliases:	-
Where/How used:	To Test_Quiz Process (Input) In Grade of Student (Output)
Description:	Student's podcast responses to questions Student Information + Result_id, Test_id + Question_id + Podcast
Format:	Text

Name:	Add Test Information
Aliases:	New Test/Quiz Information
Where/How used:	Preperation of Test/Quiz (Output)
Description:	It holds the tests that will be added to the course by the Lecturer. Course's Test + Course Information
Format:	Text

Name:	Grade of Student
Aliases:	Test Grades
Where/How used:	To Anlysing Result (Input) ToTest/Quiz Process (Input)
Description:	Student Information + Course Information + Course's Test +Grade
Format:	Int

Name:	Selected Quiz/Test Type
Aliases:	New Test/Quiz Information
Where/How used:	In Test/Quiz Process (Output)
Description:	It includes options such as seeing, listening, writing and reading to determine the test method. Student Information +Course Information + StudentChoice
Format:	Text

Name:	Deleted Course
Aliases:	-
Where/How used:	To Deletion Process (Input) In Course Database (Output)
Description:	Contains course information deleted by the admin or lecturer. Course Information
Format:	Text

Name:	Deleted Student
Aliases:	-
Where/How used:	To Deletion Process (Input) In Student Database (Output)
Description:	Contains student information deleted by the admin. Student Information
Format:	Text

Name:	Attended Course and Student Information
Aliases:	-
Where/How used:	From Manage Student Process to Admin(Output)
Description:	Contains course information is attended by student and student information. Student Information + Completed Course List
Format:	Text

Name:	Debriefing
Aliases:	-
Where/How used:	From Manage Student Process to Student(Output)
Description:	Content indicating whether the operation is complete
Format:	Bool(True or False)

Name:	List of Courses
Aliases:	Courses List
Where/How used:	To Assign Course (Output) To Manage Student Process (Output)
Description:	Contains all course information in the system. Course Information
Format:	Text

Name:	Attended Courses List
Aliases:	-
Where/How used:	From Manage Student Process to Student(Output)
Description:	Course List is attended by student Completed Course List + Student Information
Format:	Text

Name:	Online Classes List
Aliases:	Courses List, Attended Online Class
Where/How used:	To Assign Online Course (Output) To Manage Student Process (Output)
Description:	Contains all online classes in the system. Class_id + Class_name + Lecturer_name + Class_content
Format:	Text

3.6. Logical database requirements
3.7. Design constraints
3.8. Software system attributes
3.9. A brief description of each Team Member Contribution

Name:	Volkan Mazlum

- Student, Company Use Case Diagram and Tables
- Data dictionary are made by Volkan (only part of himself).
- Overview of Functional Requirement, Non-Functional Requirements and Detailed Functional Requirement parts are made by Volkan.
- Context Diagram- Level 0 and Level-1 are made by Volkan.
- DFD Level-2 of Manage Test/Quiz and Manage Course are made by Volkan.

Name:	Süleyman Serdar Erdemir

Nothing

- **He did not complete the assigned tasks.**
- Normally, Süleyman had to make the Company use case diagram and use cases, as well as the last DFD level-2s.However, he did not that.
- Company use cases were made by Volkan MAZLUM.

Name:	Mert Alp Kuvandık

- Admin Use Case Diagram and Tables
- DFD Level-2 of Manage User and Sign in are made by Mert.
- Scope part is made by Mert

Name:	Emre Haser

- Lecturer Use Case Diagram and Tables
- DFD Level-2 of Announcement and Manage Internship are made by Emre.
- Purpose and User characteristic parts are made by Emre