
Software Requirements Specification

for

Learning Management System (LMS)

Version 1.2 approved

**Prepared by Ece Büşra Civelek, Betül Taşkiran,
Deniz Ziştoğlu, Gamze Demir**

Işık University

13/06/2024

Table of Contents

1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope	2
1.5 References	2
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product Functions.....	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment	4
2.5 Design and Implementation Constraints.....	5
2.6 User Documentation	6
2.7 Assumptions and Dependencies	6
3. External Interface Requirements	6
3.1 User Interfaces	6
3.2 Hardware Interfaces.....	12
3.3 Software Interfaces	13
3.4 Communication Interfaces.....	13
4. System Features	14
4.1 Submit Homework.....	14
4.2 Enroll in course.....	15
4.3 View Course Grades.....	16
4.4 Take Quiz	17
4.5 Edit Course Material.....	18
4.6 Create A Quiz	19
4.7 Edit Grades	20
5. Other Nonfunctional Requirements.....	20
5.1 Performance Requirements.....	20
5.2 Safety Requirements.....	21
5.3 Security Requirements.....	21
5.4 Software Quality Attributes.....	21
5.5 Business Rules.....	22
6. Other Requirements	23

Revision History

Name	Date	Reason For Changes	Version
First Documentation	06/05/24-12/05/2024	First addition	1.0
Second Documentation	13/05/24-20/05/24	Update-add	1.1
Third	21/05/24-31/05/2024	Update	1.2
Fourth	01/06/2024-13/06/2024	Final Updates	1.3

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) documents the software requirements of the educational software tools targeting K12 and high schools developed by developers at Işık (Dal). Scope of the product has two main focuses as SIS and LMS being integrable to existing systems focusing on LMS to function independently allowing a “plug and play” approach. Another focus is to integrate the already existing online teaching tool ODA into LMS. This is the version 1.4.

1.2 Document Conventions

In this document titles are bold and in Times text type with font size of fourteen and explanations under every title are written in Times New Roman with font size eleven.

References are made by using section number and titles.

In a use case; preconditions are denoted by PRE-X, postconditions are denoted by POST-X and business rules are denoted by BR-X.

1.3 Intended Audience and Reading Suggestions

1. Developers:

- Interest: Detailed technical requirements, system architecture, integration points.
- Purpose: To understand what they need to build, how the components interact, and constraints.
- Relevant Sections: Introduction, Overall Description, Functional Requirements, Non-Functional Requirements, External Interface Requirements, System Features, Data Requirements.

2. Project Managers:

- Interest: Project scope, timeline, milestones, risk management, and resource management.
- Purpose: To plan, manage, and track the project's progress and ensure it goes on smoothly.
- Relevant Sections: Introduction, Other Requirements, Other Nonfunctional Requirements.

3. Marketing Staff:

- Interest: Product features, marketable points, competitive advantages, and customer needs.
- Purpose: To develop marketing strategies, promotional materials, and to communicate the product's value proposition.
- Relevant Sections: Introduction, Product Features, Use Case Scenarios.

4. Users:

- Interest: System functionalities, ease of use, capabilities.
- Purpose: To determine if the product meets their needs and how it can be implemented within their already existing systems.
- Relevant Sections: Functional Requirements, Use Case Scenarios, Mockups, Integration Requirements.

5. Testers:

- Interest: Detailed testable requirements, test scenarios, and quality assurance standards.
- Purpose: To develop and execute test plans ensuring the software meets all specified requirements.
- Relevant Sections: Functional Requirements, Non-Functional Requirements, Other Requirements, Other Nonfunctional Requirements, Use Cases.

6. Documentation Writers:

- Interest: Understanding the product to create user manuals, guides, and technical documentation.
- Purpose: To provide clear, accurate, and user-friendly documentation for end-users and administrators.

- Relevant Sections: Functional Requirements, User Interface Mockups, System Operations, Integration Requirements.

1.4 Product Scope

The System is designed to allow students to easily and efficiently access their course grades online. The primary purpose of this software is to enable students to view both their current and past course grades, facilitating easy tracking of their academic progress. The main benefits of this system include:

Purpose and Objectives:

Purpose: To enhance the management of educational processes for students and instructors.

Objectives:

- Provide easy access to educational materials.
- Simplify the management of student records and grades.
- Effectively organize and manage online, and hybrid courses.
- Offer "plug and play" features for seamless integration with existing systems.
- Allow students to analyze their academic performance.

Relevant Benefits:

For Students:

- Improved Accessibility:** Students can access their grades from any location with internet access, whether on campus or at home.
- Historical Data Access:** Students can view grades from previous terms, allowing them to analyze their academic performance.

For Instructors:

- Simplified uploading and updating of course content.
- Enhanced processes for monitoring and evaluating student performance.

For Administration:

- Increased Efficiency:** The system eliminates the need for manual grade distribution and provides instant updates, allowing real-time access to the latest grade information.
- A comprehensive overview of the educational process for better management.
- Improved processes in terms of efficiency and effectiveness.
- User-Friendly Interface:** The interface is intuitively designed, making it easy for students to quickly find the information they need.

Alignment with Corporate Goals:

This software aligns with DaI's strategy to extend its success in K12 and high school educational software into the higher education market. The modular structure of the software, which allows customization and independent use according to customer needs, aligns with strategies to expand market share. This system supports the institution's business strategy in the following ways:

Enhancing Student Services: Providing a digital solution that meets the needs of tech-savvy students.

Increasing Operational Efficiency: Reducing administrative burden associated with grade distribution.

Supporting Data-Driven Decisions: Enabling students and advisors to make informed decisions based on accessible academic performance data.

1.5 References

This Software Requirements Specification (SRS) references other documents and websites listed below. Each reference includes enough information for the reader to access it, including title, author, version number, date, and source or location.

1. Developers at Işık (DaI) Vision and Scope Document
Author: DaI Product Management

- Date: May 15, 2024
Version: 1.0
Location: DaI Internal Wiki
2. Işık'ta Geliştiriciler K12 Tools User Interface Style Guide
Author: DaI User Experience Design
Date: November 14, 2023
Version: 2.2
Location: DaI Internal Wiki
 3. Developers at Işık Student Information System Integration Technical Specification
Author: DaI Integration Engineering
Date: April 1, 2022
Version: 4.3
Location: DaI Internal Wiki
 4. Developers at Işık Learning Management System API Specification
Author: DaI Software Development
Date: August 10, 2021
Version: 1.2
Location: DaI Internal Wiki
 5. Developers at Işık ODA Online Learning Component User Manual
Author: DaI Instructional Content Development
Date: September 18, 2020
Version: 3.7
Location: DaI Internal Wiki

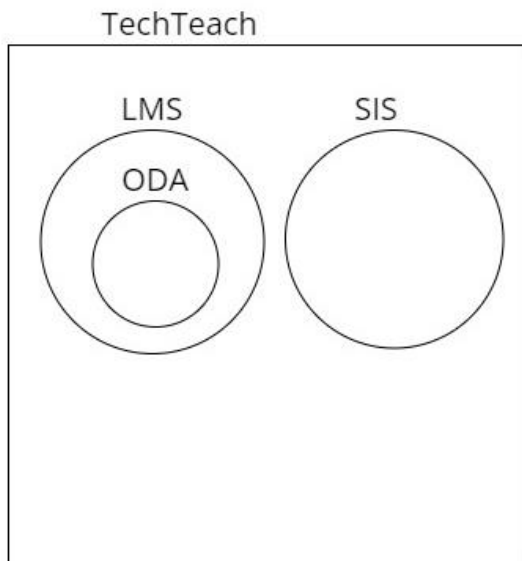
Note: The references above may not be exhaustive. Please contact DaI Product Management if any references are missing or outdated.

2. Overall Description

2.1 Product Perspective

LMS is a new self-contained product designed because of the emerging needs of tools in higher education. It is not a replacement but focuses on integrating with existing systems.

Simple diagram showing major components:



Existing ODA will be integrated with LMS to provide online teaching functionalities. LMS will exchange data with SIS. Both LMS and SIS will be integrable with existing systems supporting “plug and play”.

2.2 Product Functions

Modular Structure: The product should offer two main modules: Student Information System (SIS) and Learning Management System (LMS). These modules can be purchased bundled or separately.

Online, Face-to-Face, and Hybrid Support: The LMS should support online and hybrid modes of instruction, accommodating various teaching methodologies.

Integration Capability: Whether purchased individually or as a bundle, each module must be integrable with existing systems, ensuring a "plug and play" functionality.

Utilization of the existing online teaching component: The LMS should incorporate the existing online teaching component (ODA) to expedite marketing efforts and enhance functionality.

2.3 User Classes and Characteristics

1. Students:

- **Frequency of Use:** Students would use this system daily, for accessing course materials, participating in classes, submitting assignments.
- **Technical Expertise:** Basic skills to navigate the system and participate in online classes.
- **Importance:** Students are the primary users of the educational software and their satisfaction is absolutely necessary for the success of the product.

1. Instructors/Teachers:

- **Frequency of Use:** Regularly, for course planning, delivery, and assessment.
- **Subset of Product Functions Used:** Access to course creation, content management, communication tools, grading, and assessment features.
- **Technical Expertise:** Moderate level of technical proficiency to use teaching tools and manage course content.
- **Security or Privilege Levels:** Moderate access rights limited to course-related functions.
- **Importance:** Instructors play a very important role in delivering educational content.

2. IT Administrators:

- **Frequency of Use:** Periodically, for system maintenance, updates.
- **Subset of Product Functions Used:** Access to system for, security management, integration with existing systems, and technical support.
- **Technical Expertise:** High level of technical proficiency in system administration.
- **Security or Privilege Levels:** Highest access rights for system-level security management.
- **Educational Level or Experience:** Background in IT administration and infrastructure management.
- **Importance:** IT administrators ensure the reliability, security, and interoperability of the system

2.4 Operating Environment

The System operates within a versatile operating environment. The software is designed to function seamlessly across various hardware platforms, operating systems, and web browsers.

In terms of **hardware platform compatibility**, the system is compatible with modern devices such as desktop computers, laptops, tablets, and smartphones. This allows students to access their grades from a variety of devices.

Operating system requirements encompass widely used operating systems such as Windows, MacOS, and Linux. This ensures that the software operates across different platforms commonly used by students.

Regarding web browsers, the system is compatible with popular browsers like Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple Safari. This compatibility provides flexibility for users who prefer different web browsers.

Other software components include the need for a web server, database management system, and various programming languages. Commonly used web servers such as Apache HTTP Server or Nginx can serve as the web server. Database management systems like MySQL, PostgreSQL, or Microsoft SQL Server are preferred choices for data storage. The system is built using programming languages and frameworks such as HTML, CSS, JavaScript, and PHP.

Integration requirements include interfacing with existing Student Information Systems (SIS), authentication systems, and email services. The system integrates with these components to provide a comprehensive experience for students and administrators.

2.5 Design and Implementation Constraints

The design and implementation of the System is subject to many restrictions that developers may encounter:

Corporate Policies: The development process must comply with corporate policies, such as the company's privacy policies, data management standards, and intellectual property protection policies. Additionally, education industry-specific regulations and data protection laws must be adhered to.

Hardware Limitations: The system must run smoothly on a variety of devices, which requires adapting to different operating systems (Windows, macOS, Linux), browsers (Chrome, Firefox, Safari) and screen sizes. It should also be active on devices with low memory capacity or limited processing power.

Interfaces with Other Applications: The System must integrate with other institutional applications such as existing Student Information Systems, authentication systems and email services. These integrations enable syncing student information, single sign-on experience, and managing email notifications.

Specific Technologies and Tools: Specific technologies and tools must be selected for the development and distribution of the software. For example, a SQL-based database may be used for data storage, while a JavaScript framework such as React.js may be preferred for the user interface. These choices will have a direct impact on system performance, security and scalability.

Security Considerations: System should be equipped with security measures such as strong authentication mechanisms, data encryption techniques and session management. Additionally, security settings such as password reset policies and session timeout should be offered to increase users' data security.

Design Conventions and Programming Standards: During the software development process, established design traditions and programming standards must be followed. This includes topics such as coding style, file configuration, documentation guidelines, and error handling practices. Compliance with these standards increases the readability, maintainability, and overall quality of the software.

Maintenance Responsibility: The customer's organization may be responsible for the long-term maintenance of the software. Therefore, the design and implementation of the software should be modular and well documented to facilitate future updates and maintenance. Additionally,

extensibility features should be offered to make it easier for the customer to customize and integrate the software.

2.6 User Documentation

Delivered User Documentation Components:

User Guide: This guide will explain the product's core functionalities, installation, and usage step-by-step. It will be written in clear language and include plenty of screenshots.

Online Help: Users can access online help to troubleshoot issues and learn more about the product. This will include features like contextual help, a search function, and FAQs.

Training: DaI will offer a series of short video tutorials on how to use the product. These tutorials will help new users get started quickly.

Delivery Formats:

The user guide will be available in both online and printed formats, in PDF format.

Online help will be offered in HTML format, accessible through web browsers.

Training videos will be delivered in MP4 format, watchable on the DaI website or YouTube.

Standards:

All user documentation will be written in clear and concise language, using a consistent voice and tone.

Screenshots and other visuals will be high-quality and clear.

Training videos will be short, focused, and have an engaging visual style.

2.7 Assumptions and Dependencies

Assumptions:

Users are assumed to have basic computer literacy skills.

Users are assumed to have an internet connection.

Users are assumed to have JavaScript and cookies enabled in their browsers.

Dependencies:

This product requires integration with a Student Information System (SIS) and a Learning Management System (LMS).

The product relies on utilizing the already existing online learning component (ODA).

The product depends on specific third-party libraries and APIs.

Impact:

If the above assumptions or dependencies are incorrect, not shared, or change, the project could be impacted. This could lead to delays, reduced functionality, or performance issues in the product.

Risk Mitigation:

DaI plans to mitigate these risks by:

Documenting system requirements to ensure users have basic computer skills and internet access.

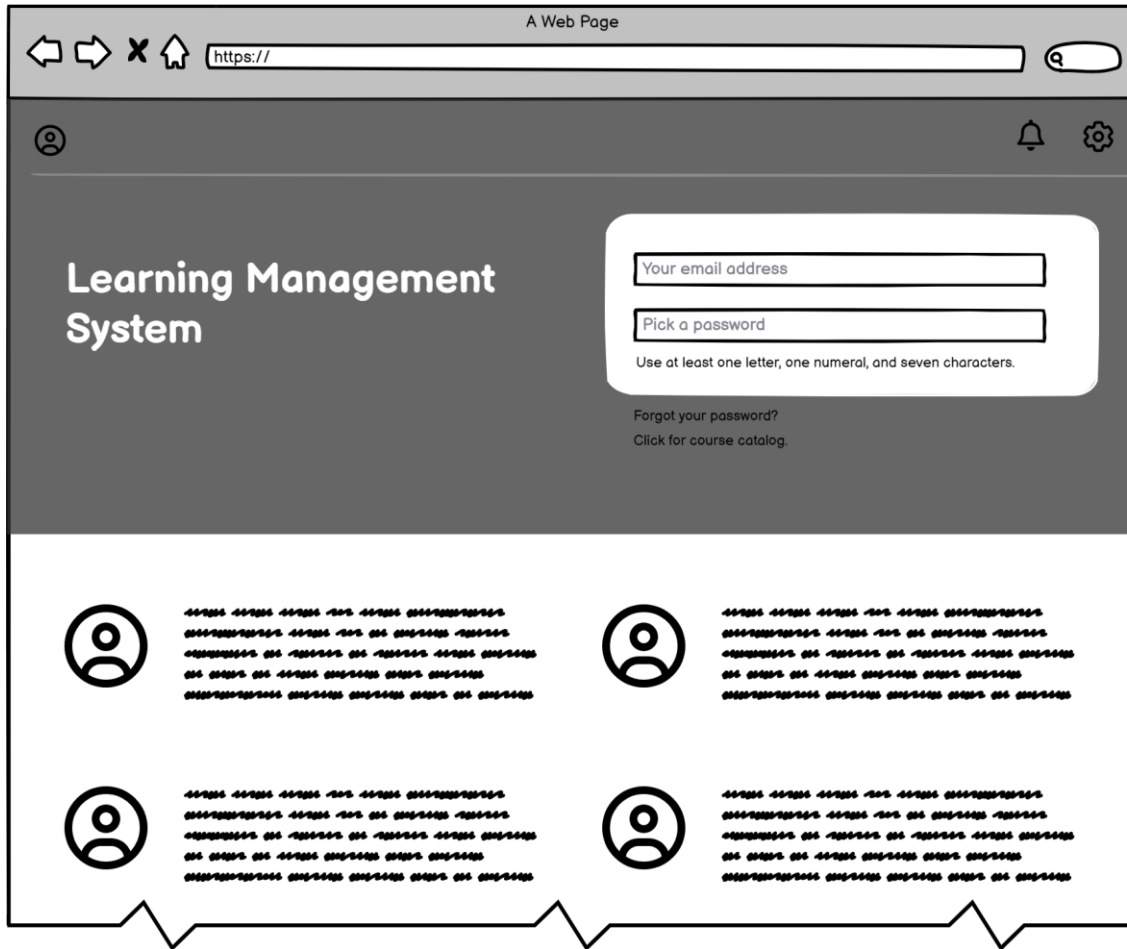
Testing compatibility with SIS, LMS, and third-party components.

Clearly outlining assumptions and dependencies in documentation and training materials.

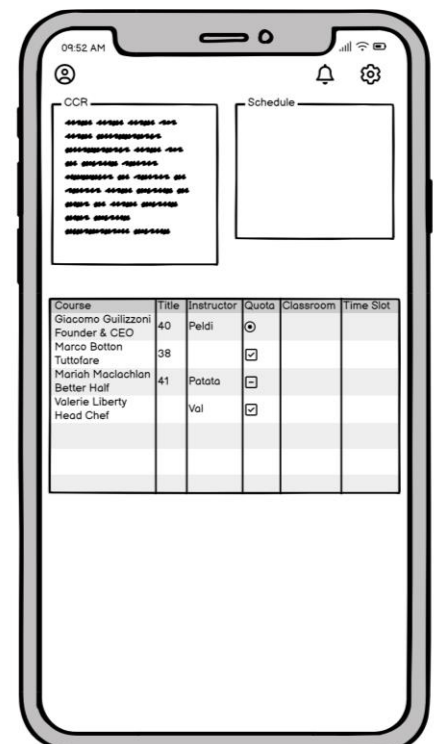
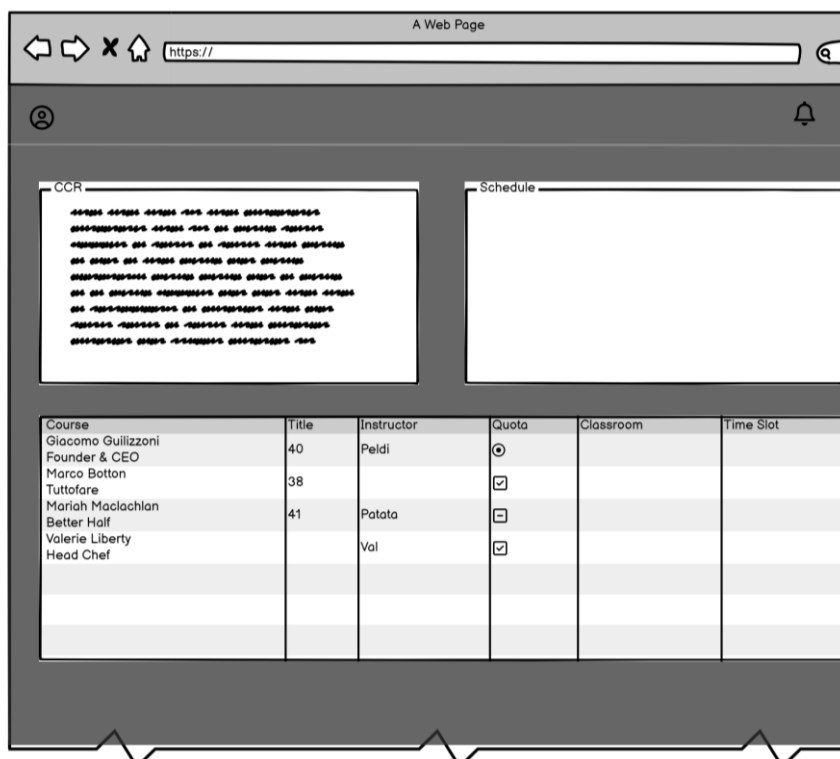
Collecting user feedback and incorporating it into product development.

3. External Interface Requirements

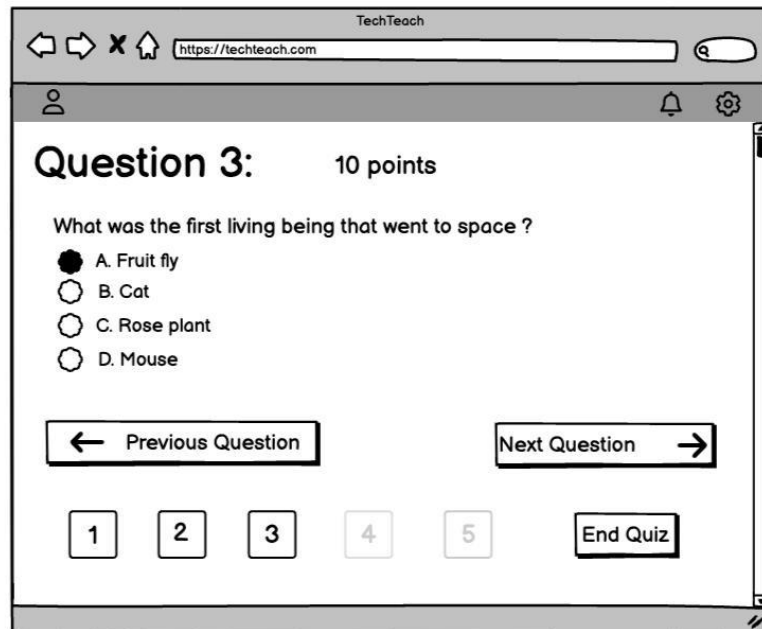
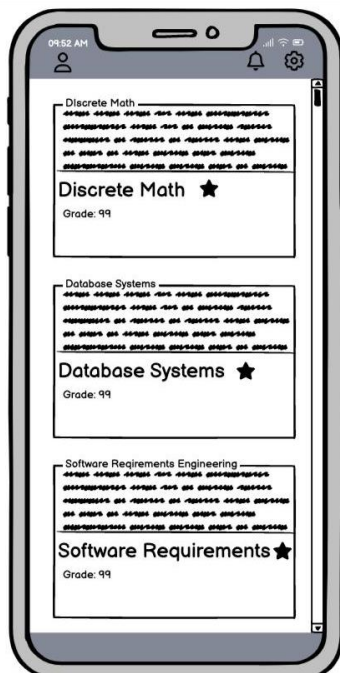
3.1 User Interfaces



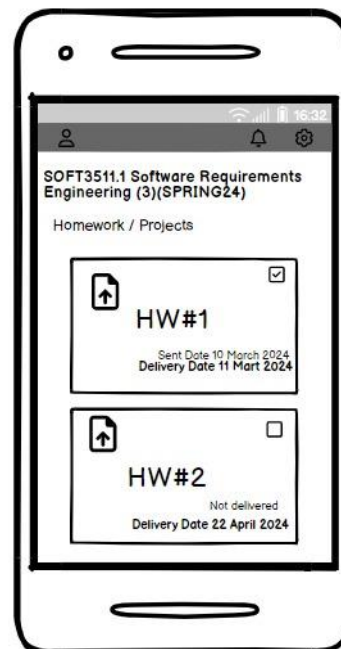
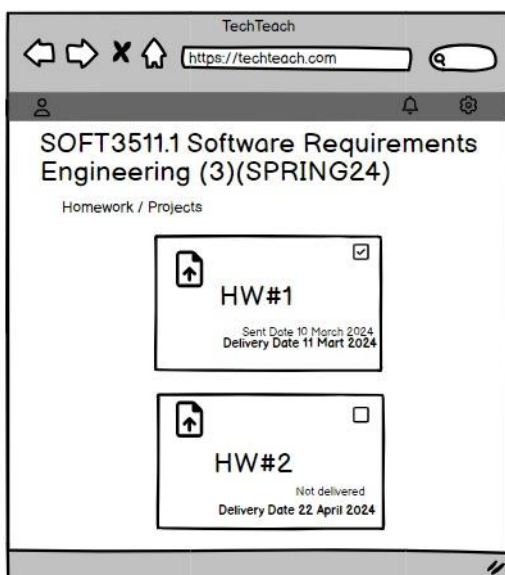
Enrollment in Course Interface

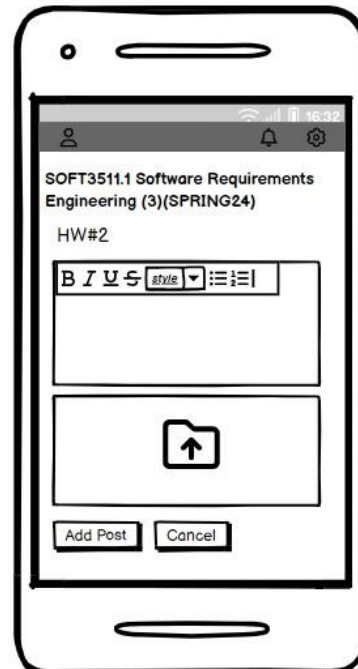
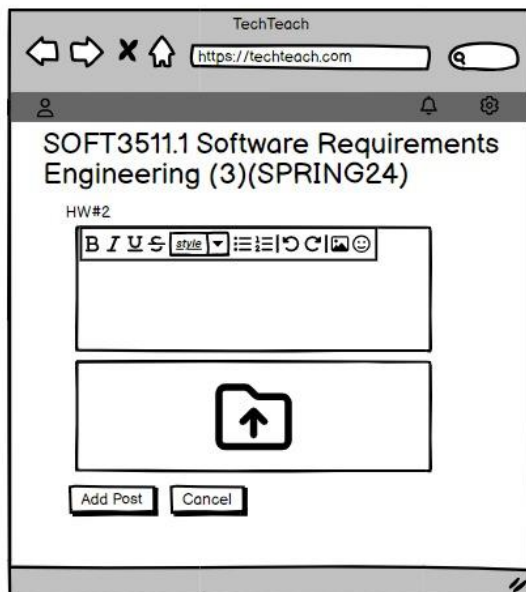
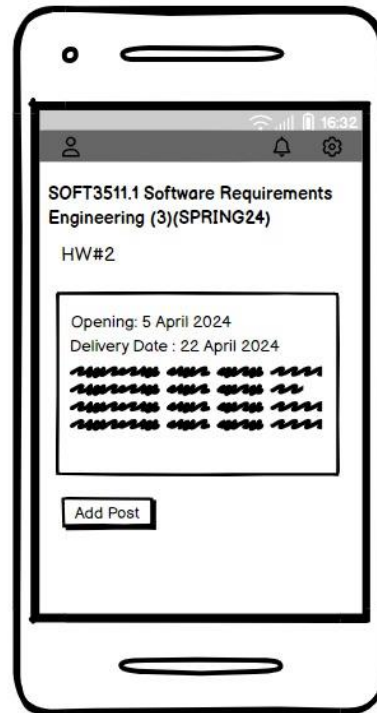


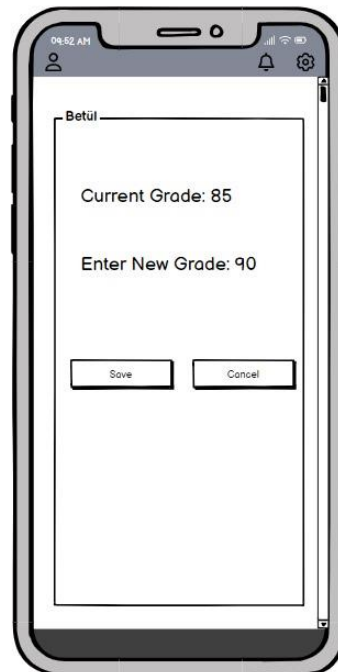
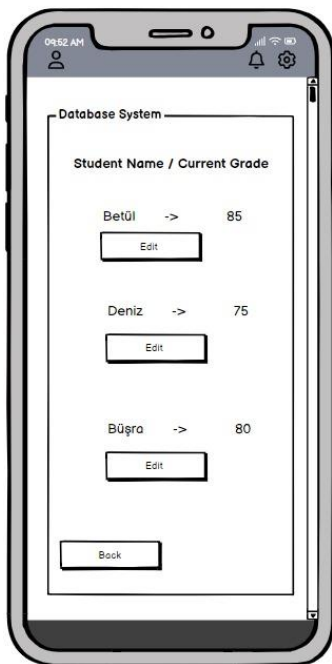
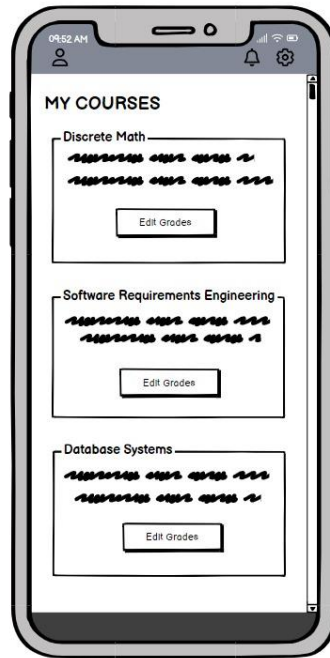
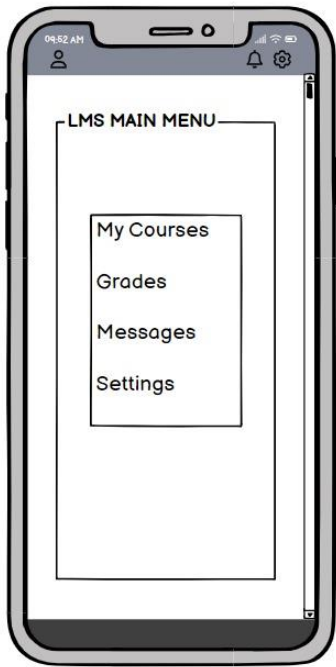
Take Quiz Interface

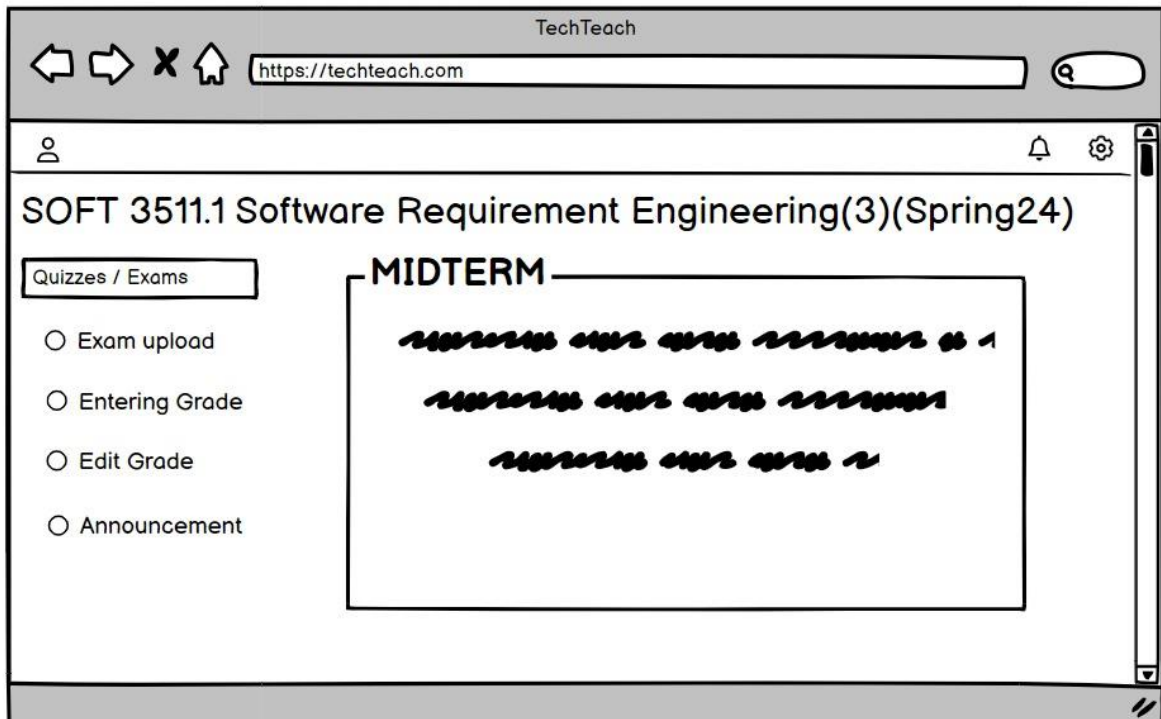
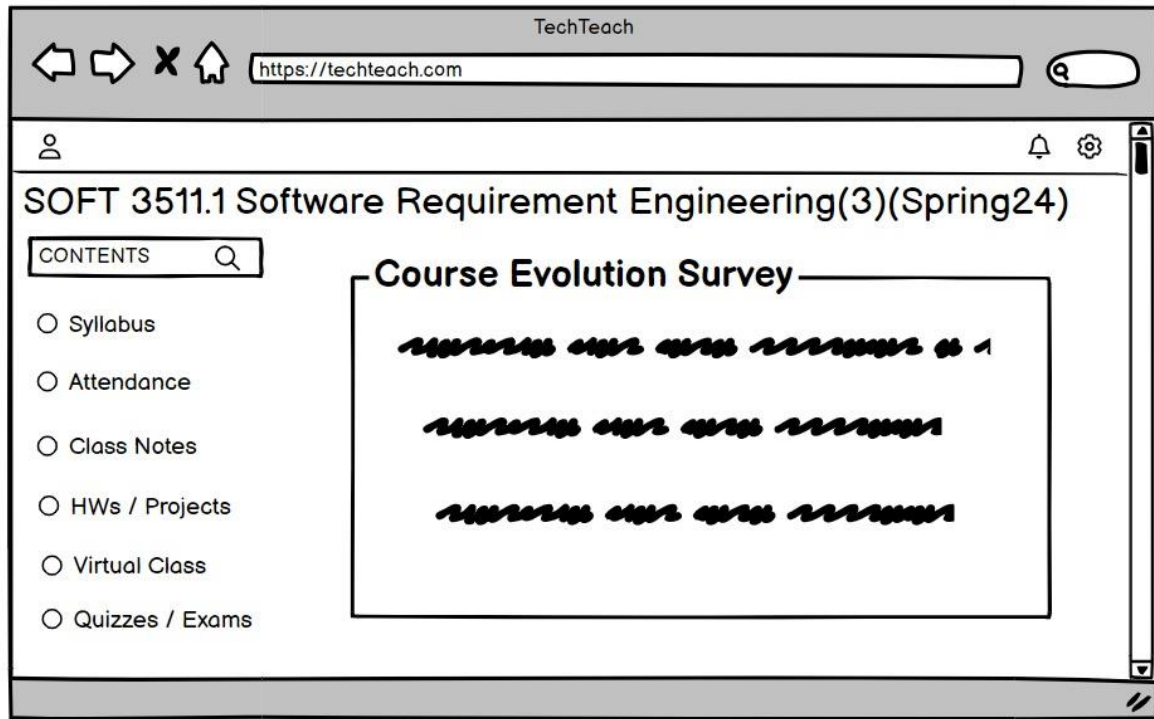


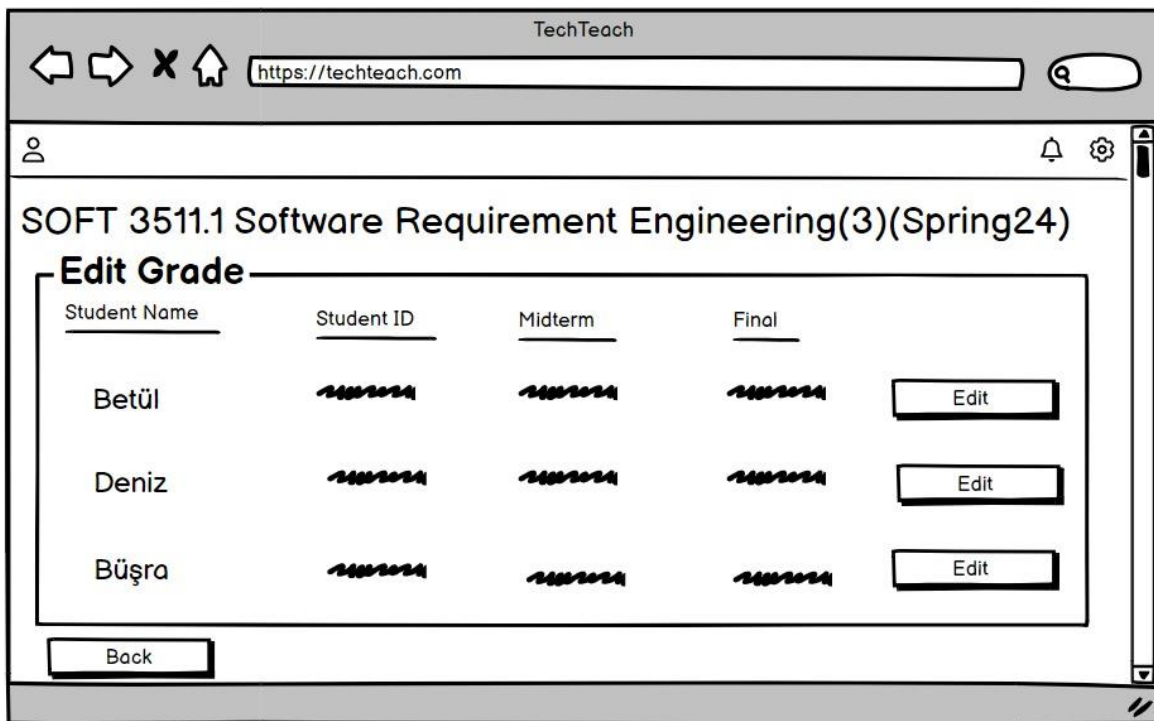
Submit Homework Interface











3.2 Hardware Interfaces

LMS will interact with different hardware components to provide a user friendly, secure and reliable platform. Here are some characteristics of interfaces:

- **Logical Characteristics:**
 - LMS will run on servers that handle web pages and application logic.
 - LMS will connect to servers that store data such as course materials and student information. (Other requirement section provides a data flow diagram for better understanding.)
 - LMS will use large storage systems for saving every course content and student-instructor data.
 - **Supported Devices:** Desktops, laptops, tablets, and smartphones.
 - **Operating Systems:** Windows, macOS, Linux, iOS, Android.
 - **Browsers:** Compatible with major web browsers like Chrome and Firefox.
- **Physical Characteristics:**
 - Different devices used by students, instructors and administrators including computers and mobile devices.

- Keyboards, mice, and touchscreens for user interaction.
- Monitors, projectors and printers for instructor's usage.
- Tools like digital whiteboards for classroom activities.

3.3 Software Interfaces

This section describes connections between LMS and other specific software components. Here are some data items for the Student Information System (SIS): Student(sName, sId, contactInfo, coursesEnrolled), Faculty(fName, fId, department), Course(cName, cId instructor), Enrollment (student, course, semester). Messages of SIS will include student information, course information, enrollment status. Communication is done through API calls. Purpose of SIS is to accommodate LMS by providing user and course information to registrate, grade and enroll. SIS will need a service to access students records, course schedules and enrollment statuses. LMS will be in data exchange with SIS by API (Application Programming Interface.) Some data items for ODA: CourseContent (lectures, presentations, assignments), OnlineAssessments (quiz, exam, homework). Messages of ODA will include course contents. Services needed will be access to course content. Communication is done through API calls and message passing. Purpose of ODA is to provide online learning support for the LMS. LMS will be again in data exchange with ODA through an API. Implementation constraint for both of these will be result of API, because it works as a layered approach so SIS can't be implemented before ODA.

3.4 Communication Interfaces

For the new educational software tool developed by DaI, communication interfaces will play a crucial role in ensuring seamless integration, data exchange, and interaction between different modules and external systems. Here are some requirements associated with communication functions:

Integration with Existing Systems:

The Student Information System (SIS) and Learning Management System (LMS) modules should be designed with open and accessible connection points to facilitate easy integration with existing educational management systems used by customers.

Compatibility with widely accepted connection standards and methods should be ensured.

Inter-module Communication:

The LMS and SIS modules must effectively communicate with each other to exchange student information, course data, enrollment details, grades, and other necessary information.

The formats used for communication should be standardized to ensure easy understanding and use.

Communication Protocols:

Communication between software modules and external systems should be secured to protect data transmission and ensure data integrity.

Secure communication methods may be required for specific functionalities, such as file transfers or email notifications.

Particularly, the transfer of student information, grades, and other sensitive data should be highly secure.

Real-time Collaboration:

For real-time collaboration features in the LMS, such as live chats, discussions, or video conferencing, fast and bidirectional communication methods should be utilized.

Data Transfer Rates:

The system should handle varying data transfer rates depending on the volume of concurrent users, especially during peak usage times.

Bandwidth optimization techniques, such as data compression and caching mechanisms, should be implemented to enhance performance and reduce latency.

Authentication and Authorization:

Reliable authentication mechanisms should be implemented to ensure that only authorized users can access the system and its functionalities.

Integration with centralized login providers or support for secure authentication methods may be necessary to streamline user authentication across different platforms.

Error Handling and Logging:

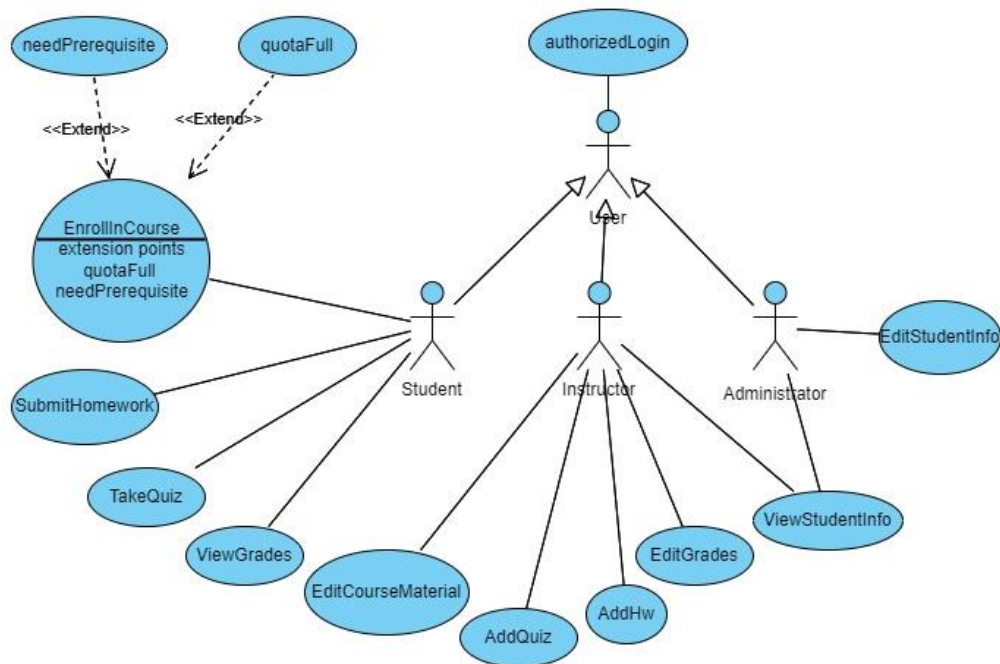
Robust error handling mechanisms should be in place to gracefully manage communication failures, such as server downtime or network disruptions.

Logging of communication activities and error messages should be implemented for troubleshooting and auditing purposes.

By adhering to these communication interface requirements, the educational software tool developed by DaI can ensure interoperability, security, and reliability in various deployment scenarios, including online, face-to-face, and hybrid modes of instruction.

4. System Features

Use case diagram:



4.1 Submit Homework

ID and Name:	UC-1 Submit Homework		
Created By:	Gamze Demir	Date Created:	06/05/24

Primary Actor:	Student	Secondary Actors:	System
Description:	This use case describes the process of a student uploading a completed assignment to the system.		
Trigger:	Student wants to submit an assignment.		
Preconditions:	-Student is logged in to the system. -The assignment is open for submission (active deadline).		
Postconditions:	-Assignment file is uploaded to the system. -Student receives confirmation with details like uploaded file name and timestamp.		
Normal Flow:	1. Student selects the assignment they want to submit from a list. 2. The system prompts the student to upload a file. 3. Student selects the file containing their assignment. 4. The system validates the file format (e.g., docx, pdf). 5. The system uploads the file and displays a progress bar. 6. Upon successful upload, the system displays a confirmation message with details like uploaded file name and timestamp.		
Alternative Flows:	Student cancels upload: - During steps 3 or 4, the student can choose to cancel the upload process. The system returns the student to the assignment details page.		
Exceptions:	1. Network error: - During step 5, a network error occurs. The system displays an error message and allows the student to retry the upload. 2. Invalid file format: - During step 4, the uploaded file format is invalid. The system displays an error message specifying the supported formats and prompts the student to select a valid file. 3. File size limit exceeded: - During step 3, the selected file size exceeds a predefined limit. The system displays an error message specifying the limit and prompts the student to select a smaller file.		
Priority:	Normal		
Frequency of Use:	Varies depending on course workload but generally frequent during assignment deadlines.		
Other Information:	- The system should support large file uploads efficiently.		
Assumptions:	1. Students have access to a device and internet connection to upload files. 2. Students are familiar with uploading files on a computer system.		
Business Rules:	1. Only assignments open for submission can be uploaded. 2. File size limitations may apply (depending on system storage capacity).		

4.2 Enroll in course

ID and Name:	UC-2 Enroll in course		
Created By:	Ece Büşra Civelek	Date Created:	07/05/24
Primary Actor:	Student	Secondary Actors:	LMS
Description:	This use case represents when a student enrolls to a course. Student selects a course, requests to enroll and receives a message according to		

	confirmation.
Trigger:	Student requests to enroll to a course.
Preconditions:	PRE-1: Student logged into LMS.
Postconditions:	POST-1: Student is enrolled into a course. POST-2: Student enrollment status is updated. POST-3: A confirmation or an error message is displayed.
Normal Flow:	2.0 Enroll in course 1. System shows available courses. 2. Student selects a course. 3. System shows information of each course as name, course id, instructor, schedule. 4. Student requests to enroll to the course. 5. System checks for quota and student's information for prerequisites 6. System displays an appropriate message according to successful enrollment or full quota or need for prerequisite.
Alternative Flows:	None
Exceptions:	2.0.E1: The course has full quota. 2.0.E2: Student doesn't meet prerequisites.
Priority:	High
Business Rules:	BR-5 and BR-6

4.3 View Course Grades

ID and Name:	UC-3 View Course Grades		
Created By:	Betül Taşkıran	Date Created:	08/05/24
Primary Actor:	Student	Secondary Actors:	Student Information System
Description:	A student accesses the Course Grades System from the corporate intranet or from home, views the grades of the desired course and, if necessary, has the opportunity to review the grades of previous semesters.		
Trigger:	A student indicates that they want to view course notes.		
Preconditions:	PRE-1. The student must be logged into the Course Grades System. PRE-2. The student must be registered as a valid user.		
Postconditions:	POST-1. Course notes are stored in COS in a format that the student can view. POST-2. Course notes are presented in a format convenient for student viewing. POST-3. The student can successfully view course notes.		
Normal Flow:	3.0 View Notes for a Single Lesson 1. The student selects the relevant course to view its grades. 2. COS retrieves and presents the grades of the selected course to the student. 3. The student can review and examine the grades in detail.		

	4. After reviewing the grades, the student can exit from COS or perform another action.
Alternative Flows:	1.1 View Grades for Past Semesters 1. The student chooses the option to view grades for past semesters. 2. COS brings and presents the grades for the requested past semester to the student. 3. The student can review and download the grades for past semesters if needed.
Exceptions:	1.0.E1 Previous Term Grades Not Available -When the student wants to view past semester grades, there is no grade information for this semester in the system. -COS displays an error message to the student stating that grades for the relevant semester are not available. -Alternatively, the student may choose to view grades for another term or exit COS.
Priority:	High
Frequency of Use:	Approximately 300 users per day on average. Peak usage load for this scenario could be between 9:00 A.M. and 10:00 A.M. local time.
Business Rules:	BR-10, BR-11 and BR-12
Other Information:	-The student can end the process of reviewing grades at any stage. -The student can view all the courses he/she has ordered over the past six months and repeat one of these courses as a new order.
Assumptions:	-The student must be logged in to COS to view past semester grades. -COS must be able to access grade data for the relevant semester in order to provide the grades for the semester requested by the student.

4.4 Take Quiz

ID and Name:	UC-4 Take Quiz		
Created By:	Deniz Ziştoğlu	Date Created:	09/05/24
Primary Actor:	Student	Secondary Actors:	System, Instructor.
Description:	A student accesses the Quiz System from the corporate intranet or from home, takes the desired quiz and, if necessary, has the opportunity to review the grade.		
Trigger:	An instructor activates the quiz in the system.		
Preconditions:	PRE-1. The student must be logged into the Quiz System. PRE-2. The student must be registered as a valid user. PRE-3. The instructor must have uploaded the quiz itself.		
Postconditions:	POST-1. Quiz is locked, student cannot take it anymore. POST-2. Quiz can be graded by the instructor or the system if the instructor		

	has uploaded the answers previously. POST-3. The student can successfully view course notes.
Normal Flow:	4.0 Take Quiz and got it graded by the system. <ol style="list-style-type: none"> 1. The student selects the relevant quiz. 2. Student goes thru the questions one by one selecting their desired answers. 3. The System records each answer. 4. Student approves the answers and ends the exam. 5. If the instructor had uploaded the answers to system, System checks the answers and shows the grade immediately.
Alternative Flows:	1.1 Take Quiz and got it graded by the instructor. <ol style="list-style-type: none"> 1. The student selects the relevant quiz. 2. Student goes thru the questions one by one selecting their desired answers. 3. The System records each answer. 4. Student approves the answers and ends the exam. 5. If the instructor did not upload the answers to system, Instructor checks the answers manually. 6. System shows the grade.
Priority:	Normal
Frequency of Use:	Approximately 300 users per month at the same time on average. Peak usage load for this scenario could be between quiz taking time designated by instructor
Business Rules:	BR-10, BR-11 and BR-12
Other Information:	-The student can end the process of quiz at any stage if they wish. -The student can view all the previous quizzes the past six months and repeat one of these courses as a new order.
Assumptions:	-The student must be logged in to System to view past quizzes. -System must be able to access question data for the relevant quiz in order to provide the quiz requested by the student. -System must be able to access answer data for the relevant quiz in order to provide the grades for the quiz requested by the student.

4.5 Edit Course Material

ID and Name:	UC-5 Edit Course Material		
Created By:	Ece Büşra Civelek	Date Created:	19/05/24
Primary Actor:	Instructor	Secondary Actors:	LMS, student
Description:	This use case represents when an instructor wants to add or update or delete material and make it visible to the student.		
Trigger:	An instructor wants to add-update-change-delete material.		
Preconditions:	PRE-1: Instructor is logged into the system.		
Postconditions:	POST-1: Material is added to the system.		

	POST-2: Students can view the material.
Normal Flow:	6.0 Edit Course Material <ol style="list-style-type: none"> 1. Instructor selects a specific course to edit its material. 2. System displays current materials. 3. Instructor choses the material that she wants to add-update-delete. 4. An interface is provided for instructor to put-delete the material. 5. System saves the changes and updates the material. 6. Students are notified by this update-add.
Alternative Flows:	Add New Course Material <ol style="list-style-type: none"> 1. Instructor chooses a course to add material. 2. Instructor chooses a material and puts it into required area provided by the interface. 3. System adds the material to the course and saves the change. 4. Students are notified with a new course material.
Exceptions:	2.0.E1: System gives error and doesn't add-update the material because of the size of the material. 2.0.E2: System gives error and doesn't add-update the material because of the type of the material.
Priority:	High
Business Rules:	BR-13 and BR-14

4.6 Create A Quiz

ID and Name:	UC-6 Create A Quiz		
Created By:	Deniz Zıstoğlu	Date Created:	19/05/2024
Primary Actor:	Instructor	Secondary Actors:	System
Description:	Instructor wants to quiz their students; they create a specific quiz for a course.		
Trigger:	Instructor decides to create a quiz.		
Preconditions:	PRE-1. Instructor is logged in to the System.		
Postconditions:	POST-2. Students can access and take the quiz.		
Normal Flow:	5.0 Create Quiz <ol style="list-style-type: none"> 1. Instructor decides for the date and time of quiz. 2. Instructor uploads the questions. 3. Instructor decides if students can move freely between questions. 4. Instructor decides and sets the duration of the quiz 5. Instructor uploads the answers. 		
Alternative Flows:	5.2. Instructor does not upload the answers.		
Exceptions:	5.2 If Instructor does not upload the answers, Instructor should check the quiz after all of the students take it.		
Priority:	Normal		

4.7 Edit Grades

ID and Name:	UC-7 Edit Grades		
Created By:	Betül Taşkıran	Date Created:	20/05/24
Primary Actor:	Instructor	Secondary Actors:	Administrator
Description:	This use case allows instructors to edit grades for students enrolled in a course.		
Trigger:	Instructor initiates the process to edit grades.		
Preconditions:	PRE-1: Instructor is logged into the system and has appropriate permissions.		
Postconditions:	POST-1: Changes made to grades are successfully saved in the system. POST-2: Students are notified of any grade modifications. POST-3: Gradebook is updated to reflect the edited grades.		
Normal Flow:	8.0 Edit Grades 1. Instructor selects the course for which grades need to be edited. 2. System displays the list of enrolled students and their current grades. 3. Instructor selects a student to edit their grade. 4. System presents the current grade for the selected student. 5. Instructor modifies the grade as needed. 6. System verifies the changes and prompts the instructor to confirm. 7. If confirmed, system saves the changes and updates the gradebook.		
Alternative Flows:	None		
Exceptions:	2.0.E1: If the instructor tries to edit grades for a course they are not assigned to, system displays an error message. 2.0.E2: If the system encounters an error while saving changes, it notifies the instructor and prompts them to retry.		
Priority:	Normal		
Business Rules:	BR-10, BR-11 and BR-16		

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- 1. Response Time:** The system must respond to user interactions within acceptable timeframes.
- 2. Scalability:** The system must be able to handle increasing numbers of users and data without significant degradation in performances.
- 3. Reliability:** The system must maintain high availability and uptime, minimizing downtime for maintenance or unexpected issues.
- 4. Load Balancing:** The system should distribute workload evenly across servers to ensure optimal performance and prevent any single server from becoming overloaded.

5. Integration Performance: When integrated with existing systems, the module should seamlessly exchange data and communicate with other software components.

6. Resource Utilization: The system should utilize hardware resources efficiently, such as CPU, memory, and storage, to maximize performance while minimizing costs.

7. User Experience: The system should provide a responsive and seamless user experience across different devices and network conditions.

5.2 Safety Requirements

1. Data Security and Privacy:

- The system must comply with relevant data protection regulations.
- Personal data of students, instructors and faculties must be securely stored.
- Access to data must be restricted based on user roles.

2. System Reliability:

- The software must be reliable and stable, with minimized downtime to ensure uninterrupted access to educational resources.
- Regular backups of data must be performed to prevent data loss in case of system failures or errors.

3. Authentication and Authorization:

- Strong authentication mechanisms (e.g. multi-factor authentication) must be implemented to prevent unauthorized access to the system.
- Role-based access control must be enforced to ensure that users can only access resources and functionalities appropriate to their roles.

4. Secure Communication:

- All communications between users and the system, including data transmission, must be encrypted to prevent leakage of sensitive information.

5. User Safety Guidelines:

- Clear guidelines and instructions must be provided to users on how to safely use the system.
- Educational content must match to ethical standards and guidelines to ensure the safety and well-being of students and instructors.

6. Emergency Procedures:

- Procedures must be in place to handle emergencies such as system failures and possible data leaks.
- Contact information for technical support and emergency response must be available to users.

5.3 Security Requirements

Data Protection: Ensure the security and privacy of user data utilized or generated by the product, complying with relevant data protection regulations.

User Authentication: Implement robust user identity authentication mechanisms to verify the legitimacy of users accessing the system.

Compliance: Adhere to external policies or regulations related to security and privacy issues, ensuring the product meets required standards.

Certifications: Obtain necessary security or privacy certifications to demonstrate compliance with industry standards and reassure customers about the product's security measures.

5.4 Software Quality Attributes

Software Quality Attributes define additional attributes that are critical to users and developers of the product. These attributes include factors such as software usability, reliability, accuracy, flexibility, and maintainability. These features ensure that the product is easily adopted and maintained by users, enhancing the overall user experience while allowing developers to manage the software effectively.

Usability:

The user interface should enable easy viewing and management for students and lecturers. This should make it easier for users to quickly access to system and make necessary edits. Additionally, operations such as filtering and sorting should be simple and intuitive for users. User-friendly error messages should make it easy for users to understand and solve problems they encounter.

Reliability:

The system should allow users to access to the system reliably. This starts with loading correctly. The system must be constantly available without any interruptions, so users can always access to system.

Correctness:

The system must display users' notes correctly. This includes ensuring that notes are gathered correctly from the database. Additionally, the system must process input data correctly and provide accurate results to users.

Flexibility:

The system should be usable on different devices and platforms (mobile phones, computers), allowing students and lecturers to access to system from different devices. Additionally, the system should be available in different languages and allow users to choose their preferred language.

Ease of Maintenance:

The code base of the software must be easily accessible so that developers can read, understand and modify the code. The system must have the ability to adapt to new requirements or technological changes so that future updates and improvements can be easily implemented. These quality features increase the value of the product to users and developers.

5.5 Business Rules

1. Only authorized users (student, instructor, administrator) can access LMS.
2. Different users (student, instructor, administrator) have different distinct permissions in LMS.
3. Only registered students can enroll courses from LMS.
4. Students can only access the courses that they are enrolled.
5. Course quotas must be checked before enrollment is confirmed.
6. Student prerequisites must be verified before enrollment is confirmed.
7. All works must stick to academic policy including plagiarism and cheating.
8. Course materials are only available for one semester.
9. Homework submissions must be checked for the current date and time with the assignment deadline.
10. System must verify student's information before showing grades.
11. Grades will only be available when the instructor releases them.
12. System must provide access to grade data until the end of each semester.
13. Only authenticated and authorized instructors can edit the course materials.
14. Every change in courses will be notified to students.
15. LMS must follow all regulations regarding to data privacy
16. System must check the new grades to ensure it is between an acceptable range for the course.

6. Other Requirements

Appendix A: Glossary

API: Application programming interface, is a way for two or more computer programs or components to communicate with each other.

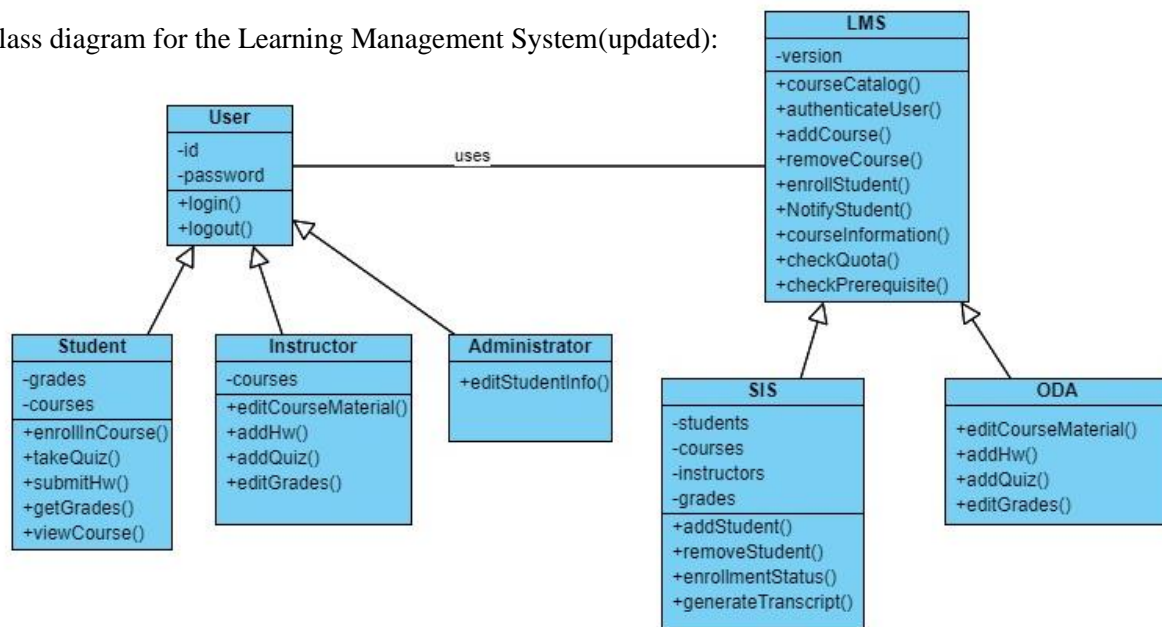
BR: Business Rules, covers all the kinds of guidance needed for day-to-day business operations.

POST: Postcondition, is a condition or predicate that must always be true just after the execution of some section of code or after an operation in a formal specification.

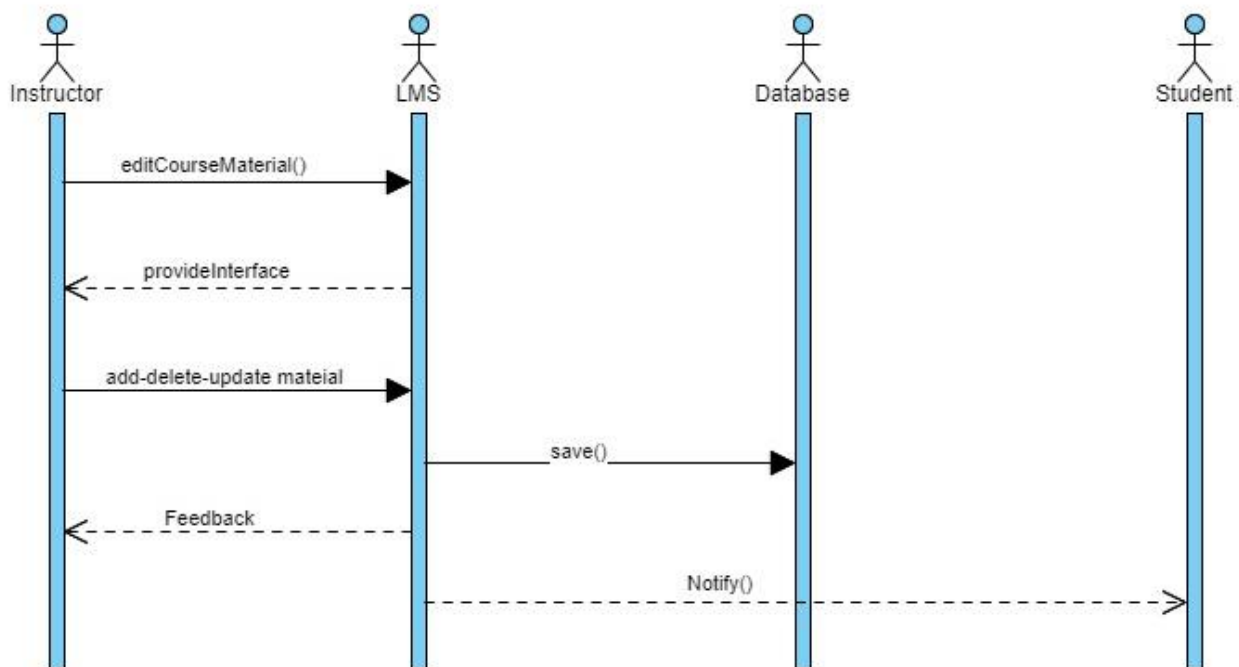
PRE: Precondition, is something that must happen or be true before it is possible for something else to happen.

Appendix B: Analysis Models

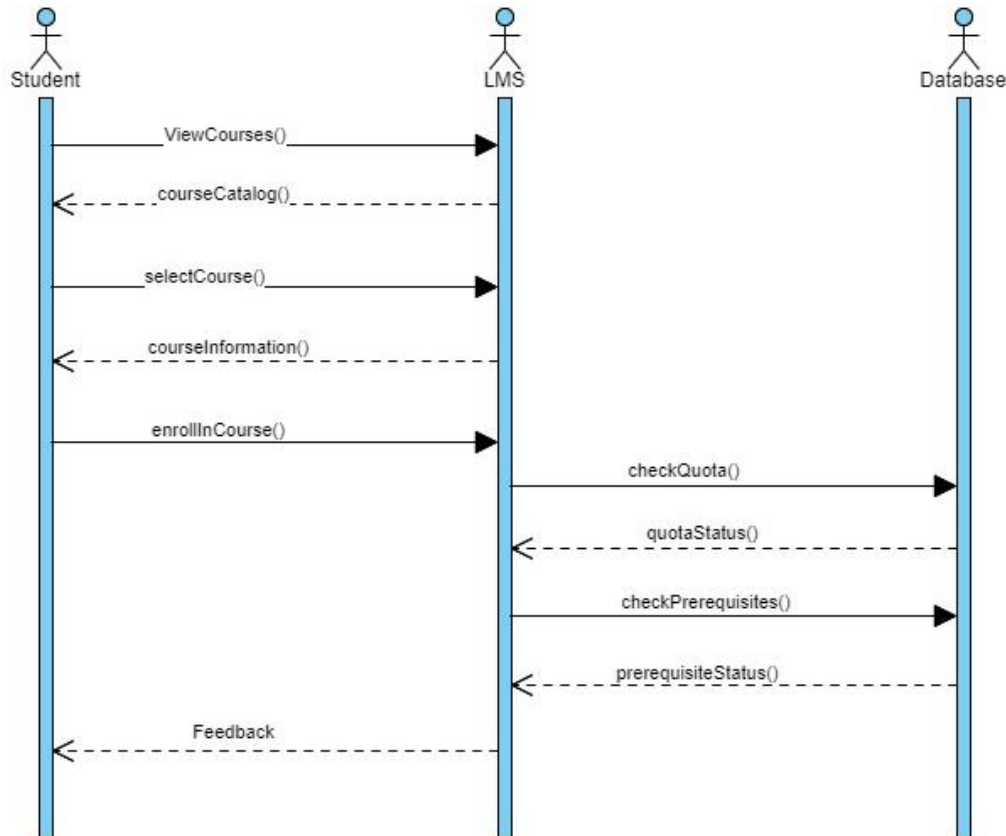
Class diagram for the Learning Management System(updated):



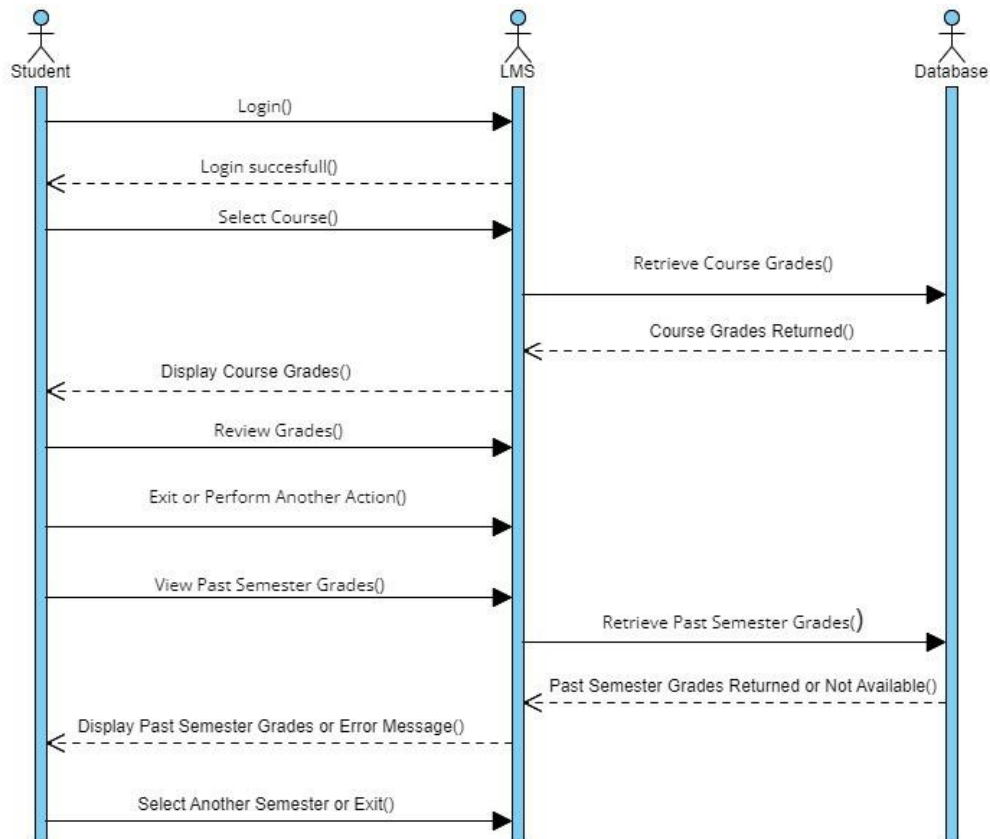
Sequence diagram for Edit Course Material high priority use case:



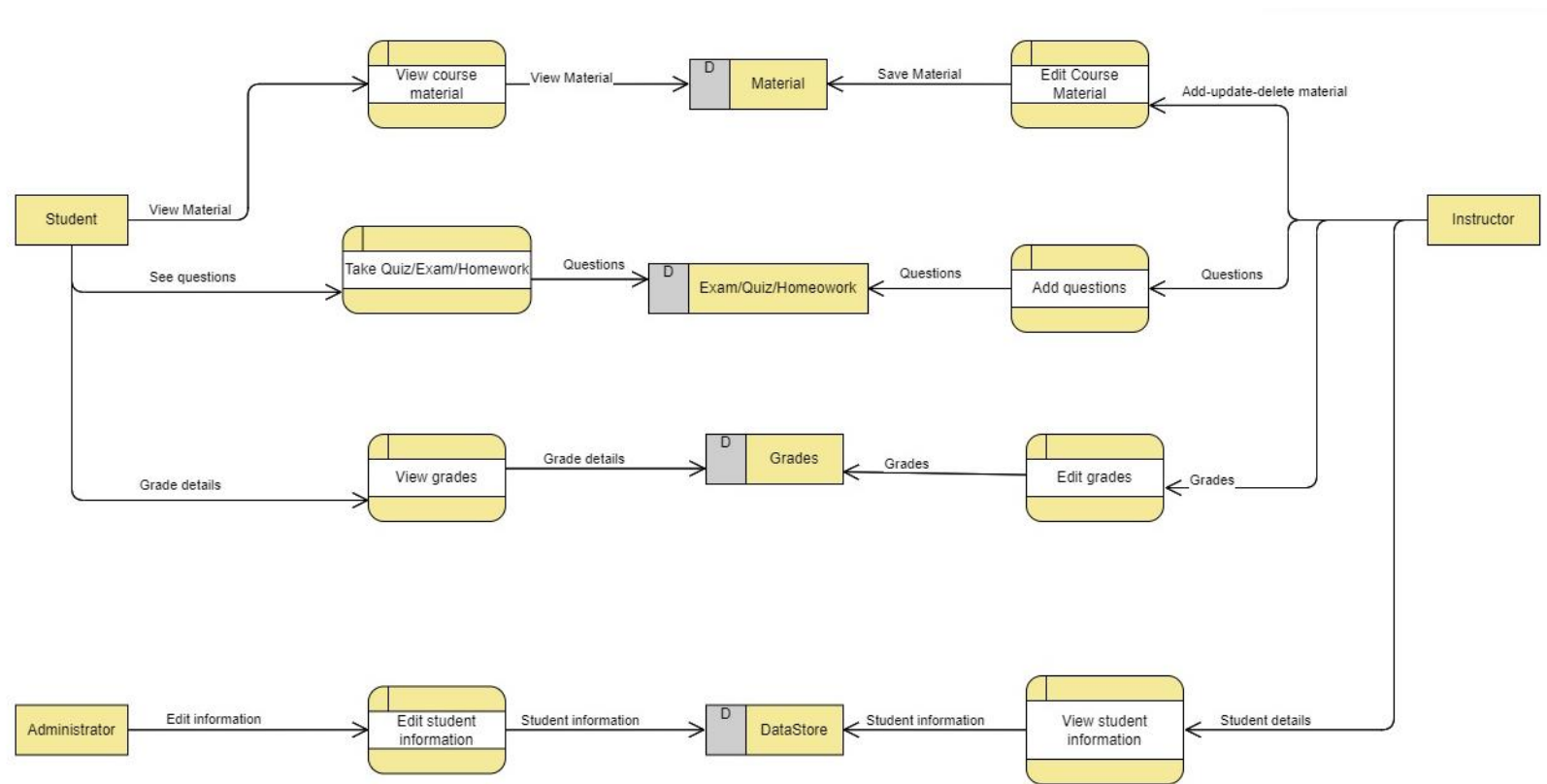
Sequence diagram for Enroll in Course high priority use case:



Sequence diagram for View Course Grades high priority use case:



Data flow diagram of the Learning Management System focusing more on interactions between users:



Appendix C: To Be Determined List

- Name of the tool: Determined to be TechTeach.
- Priority levels of the use cases: All use cases are reconsidered and updated and determined their priorities.
- Business rules: Are determined after use cases were decided precisely.