

SOFTWARE CONSTRUCTION

(CS-322-SC)

**Software Requirements
Specifications (SRS) Document**

CASE STUDY # 4

ASSIGNMENT SYSTEM

SUBMITTED TO: Dr. ONAIZA MAQBOOL

GROUP NO. 10

TEAM MEMBERS:

M JUNAID FIDA-04072113048 (Lead)

SHOAIB UD DIN - 04072113053

M KHALIL AFZAL -04072113031

Version no. 1.2

mswordcoverpages.com

Signature Page

The submitted document titled “Assignment System” is authentic work submitted by **Muhammad Junaid Fida, Shoaib Ud Din** and **Muhammad Khalil Afzal** conducted at Quaid e Azam University, Islamabad. The document strictly follows the IEEE guidelines and ensures secure coding practices. All data from any source is clearly referenced. This document has not been submitted prior in any other award or degree program and is a result of genuine self-efforts to the best of our knowledge and belief.

Signature:

Muhammad Junaid Fida

Shoaib Ud Din

Muhammad Khalil Afzal

Dated: 06-11-2023

Sign of Instructor:

Preface

This Software Requirements Specification document serves as a comprehensive guide for the Assignment System project undertaken by the Department of Computer Science. This document provides the project's purpose, scope, overall descriptions and key considerations, guiding us towards a successful implementation. The document adheres to internationally recognized standards, including ISO/IEEE/IEC, to ensure the highest quality in system development and implementation.

- **Change History:**

- Version 1.0 (Use Case Diagram and Use Cases Document)
- Version 1.1(Domain Model and System Sequence diagrams document)
- Version 1.2(Complete SRS document including all the use cases and their texts.)

Table of Contents

	Page no.
1. Introduction	05
1.1 Purpose	05
1.2 Scope	05
1.3 Definitions, Acronyms, and Abbreviations	06
1.4 References	06
1.5 Overview	06
2. Overall description	06
2.1 Product perspective	06
2.2 Product functions	08
2.3 User characteristics	08
2.4 Constraints	09
2.5 Assumptions and dependencies.....	09
3. Specific Requirements	10
3.1 External Interfaces.....	10
3.2 Functions.....	12
3.3 Performance Requirements.....	13
3.4 Software System Attributes.....	13
3.5 Use Cases.....	14

List of Figures:

1. Use Case Diagram	23
2.Doamin Model	24
3. System Sequence Diagrams	25

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for developing the Assignment System for the Department of Computer Science. It provides a detailed description of the system's functionality and behavior to guide the development process.

The intended audience includes project managers, developers, designers, testers, and members of the Department of Computer Science. It also serves as a reference for teachers and students who will use the system. The document is written in a language for individuals without extensive technical knowledge.

1.2 Scope

The software product to be produced is the Assignment System, which is designed to facilitate the management of course material, including lectures and assignments, for the Department of Computer Science.

The Assignment System will:

- Allow teachers to upload course material, including lectures and assignments.
- Notify students registered for a course when new lectures or assignments are uploaded.
- Enable students to access the system and view their class schedule, lectures, and assignments.
- Allow students to submit their assignments within the allocated time.
- Support the submission of multiple versions of an assignment before the deadline.

The Assignment System will not:

- 1) Handle the process of course allocation or course evaluation. It will rely on data provided by the Course Allocation System and Course Evaluation System.
- 2) do user authentication and signup option will be managed separately. Our system will only provide login or sign in functionality.

The Assignment System is designed to make it easier for the Department of Computer Science to handle course material. Its main objectives are to improve communication between teachers and students, create a central place where students can access and submit assignments, and make the process of submitting and grading assignments more time efficient. With features like automated notifications and submission, the system aims to reduce the amount of manual work required.

1.3 Definitions, Acronyms, and Abbreviations

RMMM – Risk Mitigation Monitoring and Management

ISO - International Standard Organization

IEC - International Electrotechnical Commission

IEEE – Institute of Electrical and Electronics Engineers

UC – Use case

1.4 References

Use case text Documentation: The documentation and diagram of all the use cases that are contained in the assignment

Course Allocation System Documentation: The documentation of the course allocation system data, which the assignment system will integrate with.

Course Evaluation System Documentation: Similarly, the documentation of the course evaluation system data, which the assignment system will integrate with.

Book: Systems and Software Engineering - Life Cycle Processes - Project Management (International Standard ISO/ IEC/ IEEE 16326 2nd Edition- 2019)

1.5 Overview

The Software Requirements Specification (SRS) document outlines the detailed requirements for the assignment system of the Department of Computer Science. This system will integrate with the course allocation and course evaluation systems to facilitate the management of course material. Teachers can upload lectures and assignments, and students will be notified of new uploads. Students can access the system to view their schedule, lectures, and assignments, and submit their work. The SRS document provides a comprehensive description of the system's purpose, scope, and user characteristics, serving as a reference for stakeholders involved in the development process.

2. Overall Description

2.1 Product Perspective

The assignment system will be integrated with the course allocation system and the course evaluation system, retrieving information about courses, students, and teachers. It will serve as a standalone system accessible to teachers and students.

2.1.1 System Interfaces

The assignment system will interface with external systems and components as follows:

1-) Course Allocation System: The assignment system will communicate with the course allocation system to retrieve information about course schedules, student enrollments, and teacher assignments. This interface will ensure that the assignment system has up-to-date information about courses and student registrations.

2-) Course Evaluation System: The assignment system will interact with the course evaluation system to gather feedback from students regarding the courses they are enrolled in. This interface will enable the assignment system to provide relevant notifications and updates based on the course evaluations

2.1.2 User Interfaces

The assignment system will have the following logical characteristics for its user interfaces:

Screen Formats and Layouts: The user interfaces should provide clear and intuitive screen formats and layouts to facilitate easy navigation and interaction. The system should use consistent design patterns and formatting across all screens.

Menus: The system should provide appropriate menus to allow users to perform their tasks efficiently. Menus should provide easy access to system functions.

The interface design should be optimized to ensure that even a less technical sound user can properly understand the system. This can be achieved by considering the following aspects:

1-) Use clear and concise language in menus and error messages.

2-) Provide meaningful and informative feedback to users when performing actions.

3-) Use appropriate visual cues, such as icons or color coding, to convey information or status.

2.1.3 Hardware Interfaces

The assignment system will have logical characteristics for its hardware interfaces, including:

Supported Devices: The system should be compatible with various devices, such as computers, laptops, or mobile devices, that have a web browser to access the assignment system.

Configuration Characteristics: The system should adapt to different hardware configurations, such as screen resolutions, input methods (keyboard, mouse, touch), and device capabilities (e.g., camera for document submission).

2.1.4 Software Interfaces

The assignment system will utilize the following required software products and interfaces:

Data Management System: The system will interface with a data management system to store and retrieve course information, student data, and assignment submissions. The specific data management system, including its name, mnemonic, specification number, version number, and source, will be determined during the system design phase. Which mostly likely would be a local data storage.

Operating System: The system will be developed to run on a specific operating system or a set of supported operating systems. The exact operating system(s) to be supported will be determined during the system design phase.

Other Application Systems: The system may have interfaces with other application systems, such as the course allocation system or the course evaluation system, to exchange relevant data. The details of these interfaces will be defined in coordination with the respective systems.

2.1.5 Communications Interfaces

The assignment system will utilize various communications interfaces, such as local network protocols, to enable communication between different system components and facilitate data exchange, such as data from course allocation system and course evaluation system. The specific communication protocols and interfaces will be determined during the system design phase.

2.1.6 Memory Constraints

The assignment system should consider memory constraints to ensure efficient memory utilization. The system should be designed to optimize memory usage, considering factors such as the number of concurrent users, the size of course material and assignment submissions, and any limitations imposed by the hardware or operating system.

2.2 Product Functions

The assignment system will provide the following functions:

- 1-) Teachers can upload course material, including lectures and assignments.
- 2 -) Students registered for a course will be notified when new lectures or assignments are uploaded.
- 3 -) Students can view the class schedule, lectures, and assignments for their registered courses.
- 4 -) Students can submit their assignments within the allocated time.
- 5 -) Multiple versions of an assignment can be submitted before the deadline.

2.3 User Characteristics

The intended users of the Assignment System are primarily teachers and students in the Department of Computer Science. The educational level of the users varies, ranging from

undergraduate to graduate level. Some users may have previous experience with similar systems, while others may be relatively new to using technology for educational purposes.

The technical expertise of the users may also vary. While some users may have a strong technical background and be comfortable with using software systems, others may have limited technical knowledge and require a user-friendly interface that is easy to navigate.

The assignment system will be used by two main user roles:

1-) Teachers: They will upload course material, manage assignments, and interact with the system to communicate with students.

2-) Students: They will access the system to view course information, submit assignments, and receive notifications.

2.4 Constraints

There will be the following constraints for this system:

1. **Time Constraint:** Four-month time duration for project completion.
2. **Budget Constraint:** There is a fixed budget xyz allocated for the project.
3. **Resource Constraint:** Limited skilled human resources available.
4. **Software and Language Constraints:**
 - Java language will be used for programming.
 - Project libre will be used for Gantt Chart.
 - Argo UML will be used for diagrams and models.
 - MS word will be used for documentation.

2.5 Assumptions and Dependencies

There will be the following assumptions for this system.

1-) The Course Allocation System and Course Evaluation System are already in place and accessible for integration.

2 -) Users will need compatible devices (computers, laptops, or mobile devices) with a web browser to access the assignment system.

3-) User authentication, that is teacher and student accounts, is managed separately and not within the scope of this project.

4-) The project team possesses the required skills and expertise to complete the tasks effectively.

5-) All the necessary resources are available for system development.

3. Specific Requirements:

3.1 External Interfaces

The detailed description of all inputs into the Assignment System is as follows:

1. **Registration Number:**

- A. Name of item: Registration Number
- B. Description of Purpose: To check if the user is registered in the system and ensure authorization.
- C. Source of input: Keyboard
- D. Valid range, accuracy and tolerance: 11-digit Numerical string
- E. Unit of measurement: no unit
- F. Relationship to other inputs: Password must match the specified Registration Number to login the system.
- G. Screen: Login Screen
- H. End Messages: If Registration number is incorrect, "Re Enter your registration number" otherwise "Enter your Password"

2. **Password:**

- A. Name of item: Password
- B. Description of Purpose: To check if authorized person is accessing the system and ensure system security.
- C. Source of input: Keyboard
- D. Valid range, accuracy, and tolerance: minimum 5 characters, including Upper- and Lower-case letters, numerical value and a special character.
- E. Unit of measurement: no unit
- F. Relationship to other inputs: Must be valid for the Registration number for authorized login.
- G. Screen: Login Screen
- H. End Messages: If Password is incorrect, give 3 chances and display the message "re-enter your password", otherwise disable the input and display the message, "Contact administrator for login details"

3. **Course Information:**

- A. Name of item: Course information
- B. Description of Purpose: to facilitate the accurate provision of course details to students and teachers who are specifically assigned to a particular course. This ensures that the right course-related information is accessible to the relevant individuals.
- C. Source of input: Course Allocation System

D. Unit of measurement: no unit

E. Screen: User profiles

4. **Course Material:**

A. Name of item: Course material

B. Description of Purpose: to facilitate students with relevant course material such as class schedule, lectures etc.

C. Source of input: File

D. Valid range, accuracy, and tolerance: (.pdf,.docx,.png,.jpg)

E. Unit of measurement: no unit

F. Relationship to other inputs: Must be valid for the specified course.

G. Screen: User profiles

H. End Messages: If there is no course material upload, display message, "No course material uploaded yet" otherwise display the list of course material.

5. **Assignment:**

A. Name of item: Assignment

B. Description of Purpose: To maintain assignment record and facilitate teacher to evaluate students.

C. Source of input: File

D. Valid range, accuracy, and tolerance: (.pdf,.docx,.png,.jpg)

E. Unit of measurement: no unit

I. Relationship to other inputs: Must belong to exact the specified course.

F. Screen: User Profiles

J. End Messages: If there is no assignment scheduled, display message, "No assignment added yet" otherwise display the list of assignments along with due date.

The detailed description of all inputs into the Assignment System is as follows:

1. **Evaluation Results:**

A. Name of item: Evaluation results

B. Description of Purpose: To facilitate students see the marks of submitted assignments and go through teacher's feedback.

C. Destination of output: Assignment Screen where students have uploaded the assignment.

D. Valid range, accuracy, and tolerance: 0-100, numerical value followed by a string.

E. Unit of measurement: no unit

F. Relationship to other inputs: evaluation results must correspond to correct assignment with respect to specified student.

G. Screen: User profiles

- H. End Messages: If evaluation results are not yet received, display the message; “Result pending”

2. Notifications:

- A. Name of item: notifications
- B. Description of Purpose: To facilitate students and teacher to receive real-time updates from the system.
- C. Destination of output: Screen/Monitor
- D. Valid range, accuracy, and tolerance: alpha-numeric string
- E. Unit of measurement: no unit

3.2 Functions:

The system shall import course information, including course details, registered students, and teacher information, from the Course Allocation System.

Teachers shall be able to upload lecture materials, which may include documents, presentations, or videos.

The system shall allow teachers to upload assignments, which include assignment instructions, files, and submission deadlines.

The system shall maintain a clear relationship between the inputs (e.g., assignment submissions) and outputs (e.g., recorded submissions).

The system shall send notifications to registered students when new lectures or assignments are uploaded.

The system shall allow teacher to configure parameters, such as assignment submission deadlines and notification settings.

The system shall handle situations such as network communication failures, and unexpected errors and provide informative error messages.

The system shall validate assignment submissions, checking for its format and size.

The system shall allow multiple versions of an assignment to be submitted before the deadline.

The system shall maintain a submission history for each student, recording the submission time and version of each assignment.

3.3 Performance Requirements:

- The system shall provide response with in 10 seconds for both normal and maximum user interactions.
- The system shall display error messages within 3 seconds.
- The system should allow multiple students to access their accounts simultaneously and system should be designed to handle large number of users approximately 5000.
- The alert notifications should be forwarded to teacher and students with in 1 hour.

3.4 Software System Attributes:

Reliability:

- The system must ensure data integrity with regard to student assignments and course materials. All data must be accurately stored, transmitted, and retrieved.
- The system should have robust error-handling mechanisms to manage unexpected errors and provide informative error messages to its users.

Security:

- The system should only allow department admin, faculty and students to access the system by their departmental IDs.
- The system should restrict access to sensitive system functions and data to authorized personnel only (i.e., University Administration).

Maintainability:

- The system shall be developed in modules to facilitate future updates and maintenance.
- Each module's functions and responsibilities should be well-documented.
- Version control systems shall be used to manage changes to the system's source code.

Portability:

- The system shall be designed to work consistently on modern web browsers like Chrome, Firefox, and Safari.
- The system should be accessible from operating systems such as Windows, macOS, and Linux.
- The system should be accessible through any device such as laptop, PC, smart phone, tablet, etc.

Availability:

- The system should be available 24/7 to its users from anywhere.
- There should be a load balancing mechanism to evenly distribute incoming user requests enhancing system performance and availability.

3.5 Use Cases:

SHOAIB UD DIN: 04072113053

Use Case UC 1: Login

Primary Actor:	User (Student or Teacher)
Stakeholders and Interests	Student: wants to access the assignment system to view their courses, lectures, assignment and submit assignments. Teacher: wants to access the assignment system to view their courses, lectures, assignments, and upload course materials Admin: wants to ensure the security and integrity of the assignment system
Pre-conditions:	<ul style="list-style-type: none">• The user must have a valid account in the system.• The user must have access to a device with an internet connection.
Post-conditions:	<ul style="list-style-type: none">• The user is successfully authenticated and gains access to the system's features.
Basic Flow:	1) The user opens the assignment system's website or application. 2) The system presents the login page to the user. 3) The user enters their username and password. 4) The system verifies the entered credentials. 5) The system grants access to the user. 6) The system redirects the user to the dashboard or homepage, where they can access their courses, lectures, and assignments based on their role (teacher or student). 7) The user can now view the class schedule, lectures, and assignments specific to their registered courses. 7.1) The user is a teacher, they can upload course materials, including lectures and assignments, which will be made available to the registered students. 7.2) The user is a student; they can view the uploaded lectures and assignments and submit their solutions within the allocated time. 8) The user can also access other features and functionalities provided by the assignment system.
Alternative Flows:	4) The entered credentials are invalid: 4.1) The system displays an error message indicating that the login attempt has failed. 4.2) The user is prompted to re-enter their username and password. 6) The user's role is not recognized or authorized: 6.1) The system displays an error message indicating that the user does not have the necessary permissions. 6.2) The user is redirected to an appropriate page or denied access to the system's features. 8.) The user encounters an error while uploading course materials: 8.1) The system displays an error message indicating the issue encountered. 8.2) The user is prompted to retry the upload or contact the system administrator for assistance.

	10) The user encounters any technical issues or errors while using the system: 10.1) The system displays an error message indicating the issue encountered. 10.2) The user is advised to try again later or contact the system administrator for support.
Special Requirements:	<ul style="list-style-type: none"> The system should have a secure authentication mechanism to ensure the privacy and integrity of user accounts and data
Frequency of Occurrence:	<ul style="list-style-type: none"> Range from daily usage during the active semester to less frequent usage during breaks or vacations. The frequency of usage might be higher during assignment submission periods when students need to submit their solutions within the allocated time.

Use Case UC 2: Upload Course Material

Primary Actor:	Teacher
Stakeholders and Interests:	<p>Teacher: Wants to provide lecture materials to students in fast and organized manner.</p> <p>Student: Wants to access and view lecture and other course material in order to stay updated with class progress.</p> <p>Admin: Wants progress report of students enrolled in offered courses and to facilitate their teachers and students.</p>
Pre-conditions:	<ul style="list-style-type: none"> The teacher is authenticated and identified as a course instructor. The course has been created and assigned to the teacher.
Post-conditions:	<ul style="list-style-type: none"> The course material (lectures and assignments) is uploaded. Registered students are alerted about the uploaded course material (lectures and assignments).
Basic Flow:	1.) The system displays the teacher's dashboard/homepage. 2.) The teacher navigates to the course section. 3.) The system retrieves the list of courses assigned to the teacher. 4.) The teacher selects the desired course from the list. 5.) The system presents options to upload course material, including lectures and assignments. 6.) The teacher chooses to upload either a lecture or an assignment. 7.) The teacher wants to upload a lecture: 7.1.) The teacher provides the lecture title, description, date, and any additional details. 7.2.) The teacher attaches the lecture files (e.g., slides, documents, multimedia). 7.3.) The teacher submits the lecture for upload. 7.4.) The system saves the lecture and associates it with the selected course. 7.5.) The system alerts the registered students of the uploaded lecture.

	8.) The teacher wants to upload an assignment: 8.1.) The teacher provides the assignment title, description, due date, and any additional instructions. 8.2.) The teacher attaches any necessary files or reference materials. 8.3.) The teacher submits the assignment for upload. 8.4.) The system saves the assignment and associates it with the selected course. 8.5.) The system alerts the registered students of the uploaded assignment.
Alternative Flows:	3.) There are no courses available for the teacher 3.1.) The system may display a message indicating that the teacher is not currently assigned to any courses. 6.) Teacher encounters any error during the upload process 6.1.) The system displays an error message 6.2.) System prompts the teacher to correct the issue.
Special Requirements:	<ul style="list-style-type: none"> • System must support different file formats for uploading course material. • System must enforce file size limits to prevent storage.
Frequency of Occurrence:	<ul style="list-style-type: none"> • May range from a few times per week to once every few weeks; depending on the course and the teacher

Use Case UC 3: Update Course Info

Primary Actor:	Teacher
Stakeholders and Interests:	Teacher: wants to efficiently review and evaluate student assignments, provide feedback, and assign grades. Students: wants to receive latest and accurate course material.
Pre-conditions:	<ul style="list-style-type: none"> • The teacher is logged into the assignment system. • The teacher has access to the relevant course.
Post-conditions:	<ul style="list-style-type: none"> • The selected course material has been successfully updated and uploaded. • Registered students receive notifications about the updates.
Basic Flow:	<ol style="list-style-type: none"> 1. The teacher selects the course for which they want to update course materials from their dashboard. 2. The system presents a list of existing course materials associated with the selected course. 3. The teacher selects the specific course material they wish to update, a lecture or an assignment. 4. The system displays the details of the selected course material, including its current content, title, and other related information. 5. The teacher makes the necessary updates to the course material, which may involve modifying content, attaching files, or changing deadlines. 6. The teacher confirms the changes they made and uploads the updated course material.

	<ol style="list-style-type: none"> 7. The system validates the changes and stores the modified course material. 8. The system notifies registered students of the changes through the alert notification. 9. The teacher receives a confirmation that the update was successful and that students have been alerted to the changes.
Alternative Flows:	<ol style="list-style-type: none"> 1. There are no courses available for the teacher <ol style="list-style-type: none"> 1.1) The system may display a message indicating that the teacher is not currently assigned to any courses. 3. There is no course material uploaded for the selected course. <ol style="list-style-type: none"> 3.1) The system provides option to upload material. 6. Teacher encounters any error during the upload process <ol style="list-style-type: none"> 6.1.) The system displays an error message 6.2.) System prompts the teacher to correct the issue.
Special Requirements:	<ul style="list-style-type: none"> • System must support different file formats for uploading course material. • System must enforce file size limits to prevent storage.
Frequency of Occurrence:	<ul style="list-style-type: none"> • May range from a few times per week to once every few weeks.

MUHAMMAD KHALIL AFZAL: 04072113031

Use Case UC 4: View Lectures

Primary Actor:	Student
Stakeholders and Interests:	<p>Student: Wants to view the course material and lectures in order to stay updated with class progress.</p> <p>Teacher: Wants to ensure that the lecture uploaded are accessible to students in an organized and easy manner.</p> <p>Admin: Wants to ensure quality education by facilitating students and teachers.</p>
Pre-conditions:	<ul style="list-style-type: none"> • The student is authenticated and identified. • The student has registered that course.
Post-conditions:	<ul style="list-style-type: none"> • The student has successfully viewed the course lectures.
Basic Flow:	<ol style="list-style-type: none"> 1.) The system authenticates the student's credentials. 2.) The system displays the student's dashboard/homepage. 3.) The student navigates to the course section.

	<p>4.) The system retrieves the list of courses in which the student is enrolled.</p> <p>5.) The student selects the desired course from the list.</p> <p>6.) The system displays the course details, including the schedule, lectures, and assignments.</p> <p>7.) The student selects the lecture they want to view.</p> <p>8.) The system presents the selected lecture including relevant details such as title, description and any attached files.</p> <p>9.) The student can view the lecture material.</p>
Alternative Flows:	<p>4.) There are no courses available for the student</p> <p>4.1.) The system may display a message indicating that the student is not currently enrolled in any courses.</p> <p>7.) The course does not have any lectures or assignments uploaded yet</p> <p>7.1.) The system may display a message indicating that no content is available</p>
Special Requirements:	<ul style="list-style-type: none"> The system should only allow registered students to access and view course contents.
Frequency of Occurrence:	<ul style="list-style-type: none"> Depends on the number of classes taken by teacher.

Use Case UC 5: View Schedule

Primary Actor:	Student
Stakeholders and Interests:	<p>Students: accessing and staying updated with the course schedule to manage their time and prepare for lectures and assignments.</p> <p>Teachers: ensure that the schedule is accurately represented to provide timely information to students.</p> <p>Admin: ensure its smooth operation, including the schedule functionality.</p>
Pre-conditions:	<ul style="list-style-type: none"> The student is logged into the assignment system. The student is registered for at least one course. The course schedule has been uploaded by the respective teacher.
Post-conditions:	<ul style="list-style-type: none"> The student has successfully viewed the schedule of the selected course.
Basic Flow:	<ol style="list-style-type: none"> 1) The student accesses the assignment system website or application. 2) The student logs into their account using their credentials. 3) The system validates the student's credentials and grants access to their account. 4) The student navigates to the "Courses" section. 5) The system displays a list of courses in which the student is registered. 6) The student selects the desired course from the list. 7) The system retrieves and displays the schedule of the selected course. 8) The schedule includes information such as lecture timings, assignment deadlines, and any other relevant course-related events. 9) The student reviews the schedule to gather necessary information about lectures and assignments.

	10) The student may take note of important dates and deadlines for future reference. 11) The system allows the student to navigate back to the main menu or perform other actions.
Alternative Flows:	3.) The student's credentials are invalid or authentication fails: 3.1) The system displays an error message indicating the login failure. 3.2) The student is prompted to re-enter their credentials or initiate the password recovery process. 7.) The schedule for the selected course is not available: 7.1) The system displays a message indicating that the schedule is not yet uploaded or is unavailable. 7.2) The student can retry viewing the schedule later.
Special Requirements:	<ul style="list-style-type: none"> The system should only allow registered students to access and view course contents.
Frequency of Occurrence:	<ul style="list-style-type: none"> Depends on the academic calendar and the number of courses taken by the student.

Use Case UC 6: View Assignment for Teacher

Primary Actor:	Teacher
Stakeholders and Interests:	Teacher: wants to view the uploaded assignment in order to evaluate and give feedback on submitted work. Students: want their assignments to be reviewed and graded in a timely manner, and they are interested in receiving constructive feedback to improve their work. Admin: wants an effective and streamlined system for managing assignments, grading, and feedback.
Pre-conditions:	<ul style="list-style-type: none"> The teacher is logged into the assignment system. The teacher has access to the relevant course and assignments for evaluation. Students have submitted their assignments within the allocated time.
Post-conditions:	<ul style="list-style-type: none"> The student receives the evaluation, feedback, and grade for their assignment. The teacher has provided feedback of assignments.
Basic Flow:	1) The teacher accesses the assignment system and navigates to the relevant course. 2) The system presents a list of assignments for the selected course. 3) The teacher selects the assignment to review. 4) The system displays a list of student submissions for the selected assignment. 5) The teacher selects a student submission to evaluate. 6) The system presents the student's submission, including any attached files or additional information. 7) The teacher evaluates the assignment based on the provided criteria, considering the quality, correctness, and adherence to guidelines. 8) The teacher provides comments and feedback to the student, highlighting strengths and areas for improvement. 9) The teacher grades the assignment using the specified grading scale. 10) The teacher submits the evaluation and feedback. 11) The system updates the student's record with the evaluation, feedback, and grade.
Alternative Flows:	4.) No assignments have been submitted for the selected course: 4.1) The system informs the teacher that no assignments have been submitted yet. 6.) The teacher encounters technical issues or difficulties accessing the student's submission, 6.1) The teacher may request technical support from the system administrator for assistance.

Special Requirements:	<ul style="list-style-type: none"> The system should allow teachers to provide detailed comments and feedback to students, including the ability to highlight specific sections or points in the assignment.
Frequency of Occurrence:	<ul style="list-style-type: none"> Vary from daily to weekly, according to the assignment submission schedule

MUHAMMAD JUNAID FIDA: 04072113048

Use Case UC 7: View Assignment for Student

Primary Actor:	Student
Stakeholders and Interests:	<p>Student: Wants to access and view their assigned assignments and any feedback provided by the teacher.</p> <p>Teacher: Needs to provide assignment details, evaluate submissions, and provide feedback to students.</p> <p>Admin: Aims to facilitate effective assignment management and enhance the learning experience for students.</p>
Pre-conditions:	<ul style="list-style-type: none"> The student is registered for the course. The student is logged into the assignment system.
Post-conditions:	<ul style="list-style-type: none"> The student has successfully viewed the assignment details and any feedback provided by the teacher. The student can take appropriate actions based on the assignment information, such as making revisions or seeking clarification if necessary.
Basic Flow:	<ol style="list-style-type: none"> 1) The student navigates to the "Assignments" section within the dashboard. 2) The system retrieves and displays a list of assignments relevant to the courses the student is registered for. 3) The student selects an assignment they wish to view in detail. 4) The system presents the assignment details, including the assignment name, description, due date, and any additional instructions provided by the teacher. 5) The system displays the submission status (submitted or not submitted). 6) The system shows the feedback provided by the teacher, including comments, grades, or any other relevant information.
Alternative Flows:	<p>5. The student has not yet submitted any versions of the assignment:</p> <ol style="list-style-type: none"> 5.1) The system displays a message indicating that no submissions have been made. 5.2) The student can proceed to submit their assignment as per the submission guidelines.
Special Requirements:	<ul style="list-style-type: none"> The system should only allow registered students to access and view assignment.
Frequency of Occurrence:	<ul style="list-style-type: none"> vary depending on the course schedule and the number of assignments per course range from daily to weekly

Use Case UC 8: Submit Assignment

Primary Actor:	Student
Stakeholders and Interests:	Student: Wants to submit an assignment.

	<p>Teacher: Wants to receive assignment submission from student in timely manner and assess the submitted assignment for grading and feedback.</p> <p>Admin: Wants to ensure smooth operation of assignment submission and manage submission and evaluation process.</p>
Pre-conditions:	<ul style="list-style-type: none"> • The student has registered that course. • The assignment is created and assigned by the teacher. • The assignment submission period is open.
Post-conditions:	<ul style="list-style-type: none"> • The assignment is successfully submitted. • The teacher is alerted about the student's assignment submission.
Basic Flow:	<ol style="list-style-type: none"> 1.) The student logs into the assignment system through id and password. 2.) The system authenticates the student's credentials. 3.) The system displays the student's homepage. 4.) The student navigates to the course section. 5.) The system retrieves the list of courses in which the student is enrolled. 6.) The student selects the desired course from the list. 7.) The system displays the course details, including lectures, assignments, and submission deadlines. 8.) The student selects the assignment they want to submit. 9.) The system presents the assignment details, including the title, description, due date, and any attached files. 10.) The student uploads their assignment file/files to the system. 11.) The system confirms the successful submission. 12.) The system displays a notification for conformation of the submission 13.) The system alerts the teacher about the student's assignment submission.
Alternative Flows:	<ol style="list-style-type: none"> 8.) The submission period has ended, <ol style="list-style-type: none"> 8.1.) The system prevents the student from submitting their assignment. 8.2.) The system displays a message that the submission is closed. 10.) The student wants to resubmit their assignment before the deadline: <ol style="list-style-type: none"> 10.1.) The student selects the assignment they want to resubmit. 10.2.) The system allows the student to upload a new version of their assignment. 10.3.) The system overwrites the previous submission 12.) The system fails to upload and submit assignment. <ol style="list-style-type: none"> 12.1) The system shows notification for unsuccessful submission of assignment. 12.2) The system shows alert message to reupload assignment files.

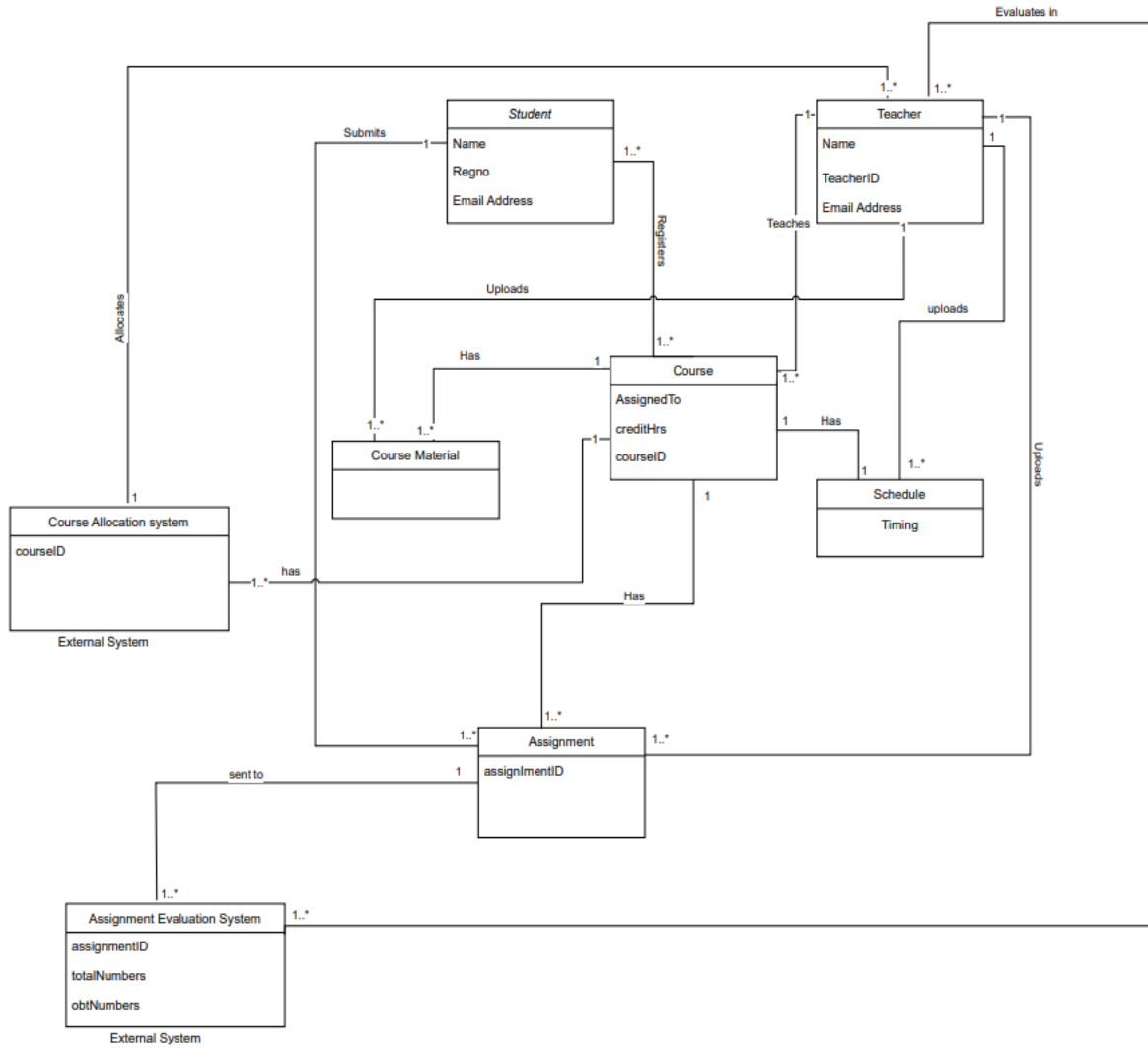
Special Requirements:	<ul style="list-style-type: none"> • The system should manage assignment submission deadlines, preventing late submissions and notifying students timely. • The system should provide secure file upload and stores submitted assignment files. • The system should manage multiple submissions for single assignment.
Frequency of Occurrence:	<ul style="list-style-type: none"> • May range from a few times per week to once every few weeks; depending on the course and the teacher

Use Case UC 9: Update Assignment

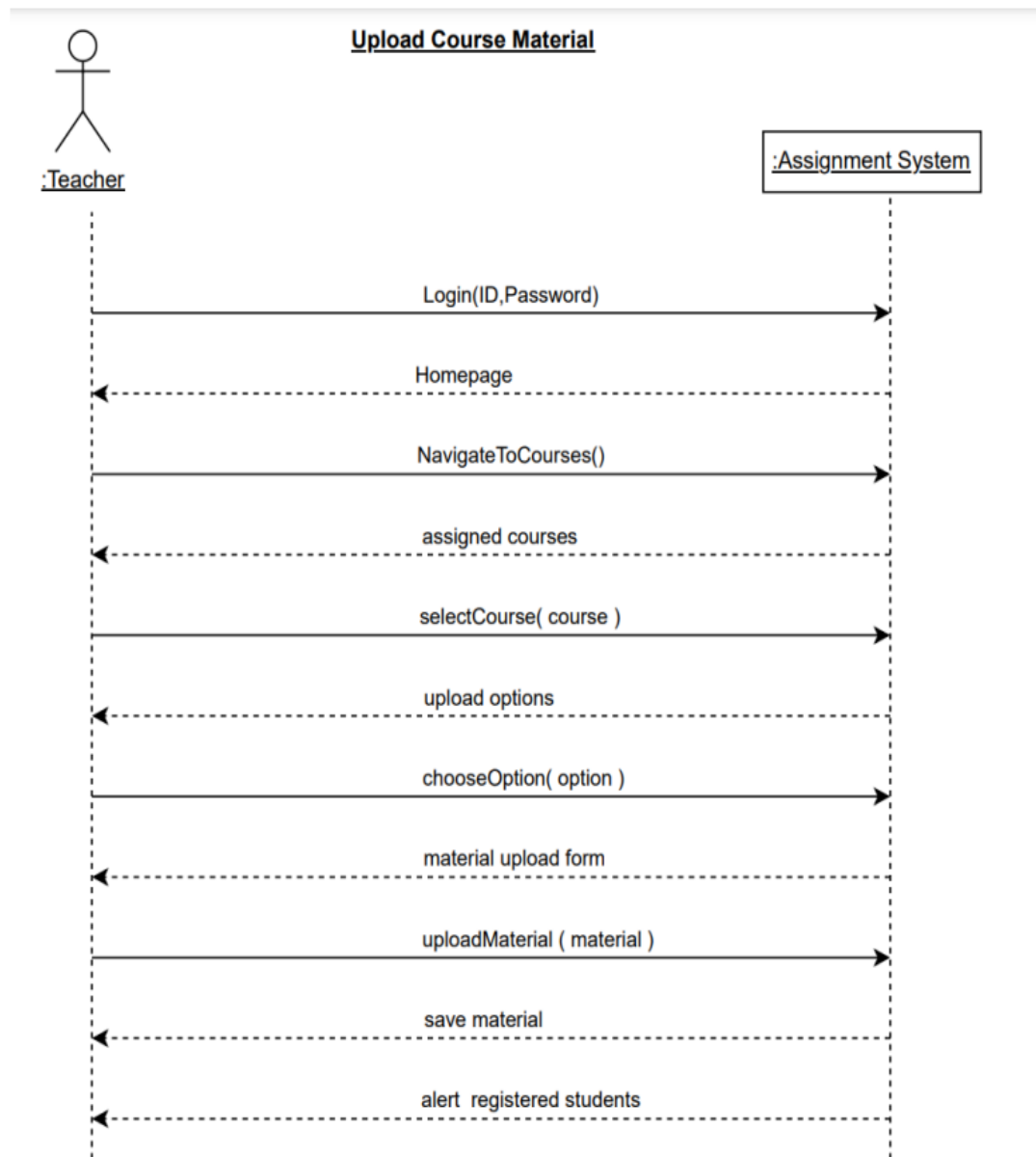
Primary Actor:	Student
Stakeholders and Interests	<p>Student: wants to update their already existing assignment by submitting a revised version before the deadline.</p> <p>Teacher: wants to receive the updated assignment from the student, review the changes made, and provide feedback or grade accordingly.</p> <p>Admin: wants to provide effective assignment management and enhance the learning experience for students.</p>
Pre-conditions:	<ul style="list-style-type: none"> • The student is logged into the assignment system. • The student has already submitted an assignment for a specific course. • The assignment update deadline is not passed.
Post-conditions:	<ul style="list-style-type: none"> • The updated assignment is successfully submitted by the student.
Basic Flow:	<ol style="list-style-type: none"> 1) The student navigates to the assignment system's website or application. 2) The system presents the student's dashboard or homepage. 3) The student selects the course for which they want to update the assignment. 4) The system displays the assignments page for the selected course, showing the already submitted assignment. 5) The student selects the option to update the assignment. 6) The system allows the student to upload the revised version of the assignment. 7) The student selects the updated file from their device and uploads it. 8) The system verifies that the updated assignment file meets the submission requirements. 9) The system replaces the previous submission with the new version. 10) The student receives confirmation that the updated assignment has been successfully submitted.
Alternative Flows:	<p>8.) The updated assignment file is invalid or does not meet the submission requirements:</p> <ol style="list-style-type: none"> 8.1) The system displays an error message indicating the issue with the updated file. 8.2) The student is prompted to re-upload a valid and compliant version of the assignment. <p>10.) There are any technical issues during the submission process:</p> <ol style="list-style-type: none"> 10.1) The system displays an error message indicating the problem encountered. 10.2) The student can retry the update process or contact the system administrator for assistance.

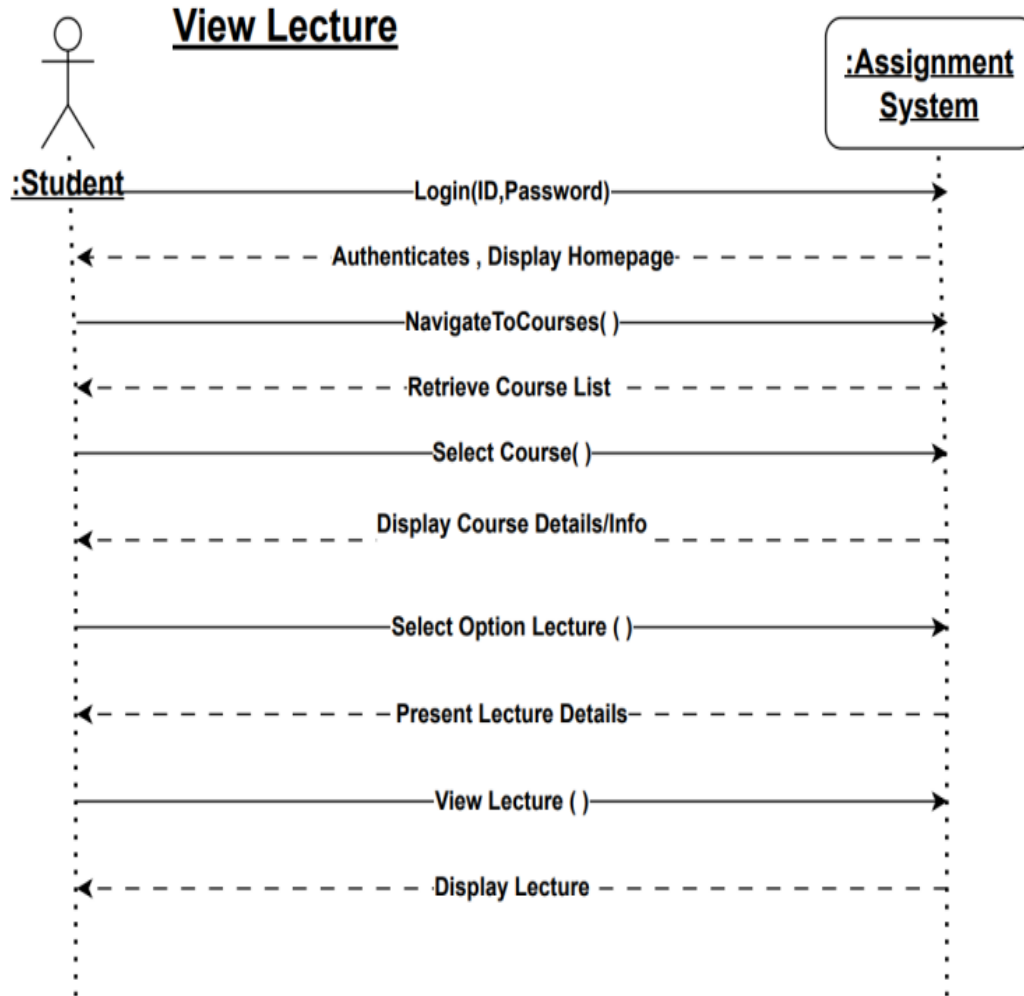
Domain Model:

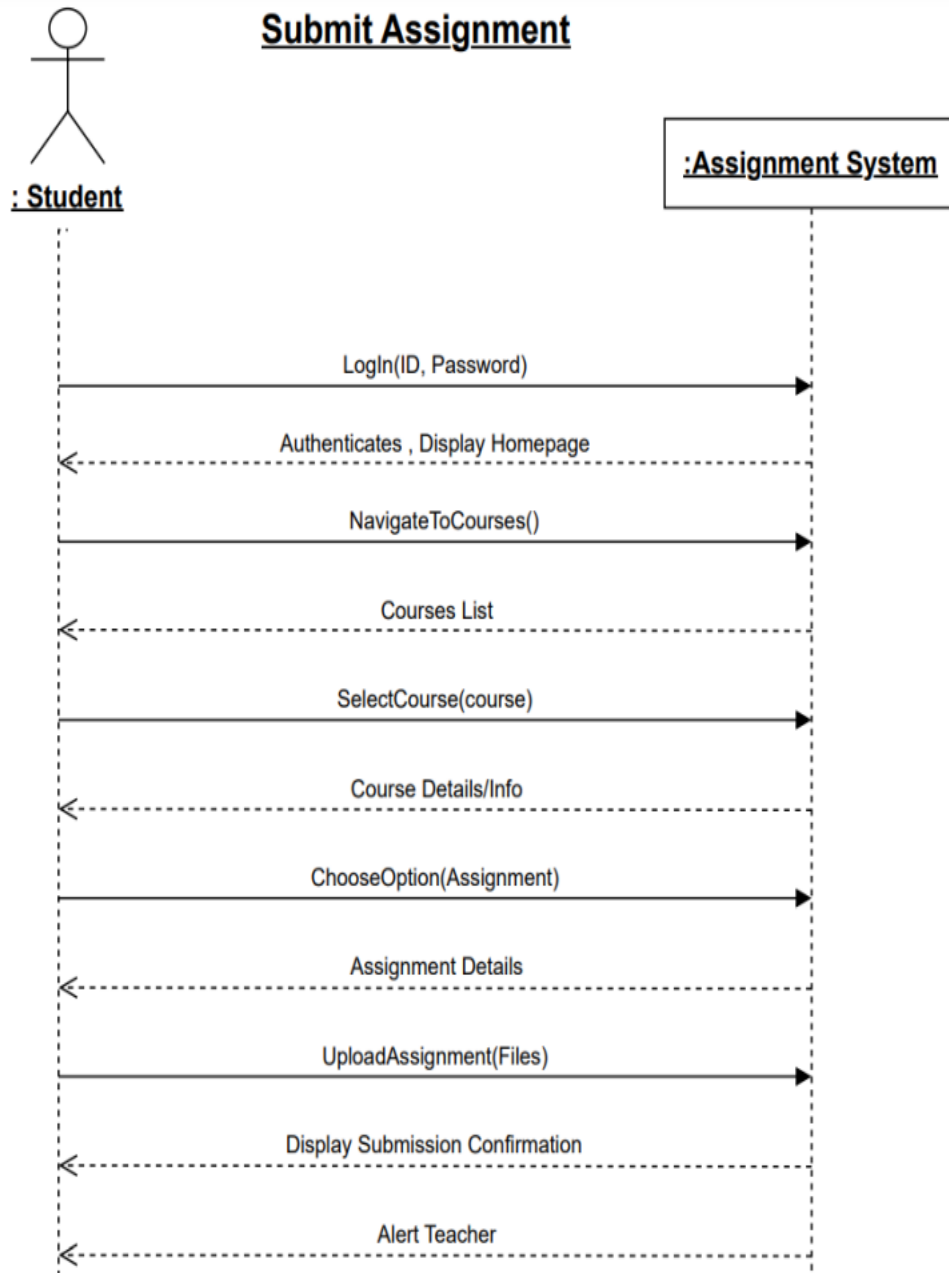
Domain Model : Assignment System



System Sequence Diagrams:







← THE END →