**COMSATS University Islamabad,**

**Abbottabad Campus**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**for**

**AI-Assisted Learning Management System**  
Version 0.1

***By***

**Hanzla Nouman CIIT/FA21-BSE-015/ATD**

**Laiba Binta Tahir CIIT/FA21-BSE-019/ATD**

**Arfah Ali CIIT/FA21-BSE-080/ATD**

***Supervisor*****Mr. Mukhtiar Zamin**

***Bachelor of Science in Computer Science (2021-2025)***

**Table of Contents**

**Revision History 3**

**1.** **Introduction 5**

1.1 Purpose 5

1.2 Scope 5

**2.** **Overall description 6**

2.1 Product perspective 6

2.2 Operating environment 6

2.3 Design and implementation constraints 9

**3.** **Requirement identifying technique 9**

3.1 Use Case Diagram 10

3.2 Use Case Descriptions 10

3.2.1 Access Recommended resourses 10

3.2.2 Save recemmondede resourses 12

3.2.3 Create lecture 13

3.2.4 Create Assgnment 14

3.2.5 Create quiz 15

3.2.6 Evaluate assingment 16

**4.** **Functional Requirements 17**

4.1 Feature 1: Personalized learning 21

4.2 Feature 1: Automated Evluations 22

4.3 Feature 1: Analytics 22

4.4 Feature 1: Content Generation 23

**5.** **Non-Functional Requirements 25**

5.1 Usability 25

5.2 Performance 25

5.3 Reliablity 25

5.4 Security 26

5.5 Scalability 26

5.6 Maintainance 26

**7.** **Project Gantt chart 26**

**6.** **References 27**

**List of Figures**

Figure 1 Use case Diagram ………..11

Figure 1 Gantt Chart ………..24

**List of Tables**

Table 1 Access Recommended resourses 10

Table 2 Save recemmondede resourses 12

Table 3 Create lecture 13

Table 4 Create Assgnment 14

Table 5 Create quiz 16

Table 6 Evaluate assingment 17

Table 7 Functional Requirement 1.1 19

Table 8 Functional Requirement 1.2 19

Table 9 Functional Requirement 2.1 19

Table 10 Functional Requirement 2.2 20

Table 11 Functional Requirement 3.1 20

Table 12 Functional Requirement 3.2 20

Table 13 Functional Requirement 4.1 21

Table 14 Functional Requirement 5.1 21

Table 15 Functional Requirement 5.2 22

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**Mr. Mukhtiar Zamin**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction**

This document presents the Software Requirements Specification (SRS) for the AI-Assisted Learning Management System (LMS). It provides an organized and structured overview of the system, its purpose, scope, and features. The document serves as a foundation for all stakeholders, including students, teachers, administrators, and developers, to understand the system requirements and design goals.

The SRS will guide the development, implementation, and validation processes to ensure the system meets user needs and expectations.

**Purpose**

The purpose of this document is to outline the functional and non-functional requirements for the AI-Assisted LMS. This software will be developed to assist educational institutions in managing teaching, learning, and evaluation processes efficiently. The primary users of this system include students, teachers, and administrators.

By leveraging AI tools, the system aims to enhance content delivery, automate administrative tasks, personalize learning experiences, and streamline evaluations. This document ensures that the requirements are clear, measurable, and feasible for implementation.

**Scope**

The AI-Assisted LMS is an advanced learning platform designed to improve the educational process by integrating artificial intelligence into content generation, evaluation, and administrative functions. It supports the following key objectives:

* Content Generation: Assisting teachers with AI-based tools to create high-quality learning materials efficiently.
* Personalized Learning: Providing AI-driven content recommendations and interactive materials tailored to individual students’ needs.
* Automated Evaluations: Enabling self-evaluation tools and standard assessment processes with instant feedback.
* Performance Analytics: Generating performance insights for teachers and administrators to make data-driven decisions.
* Attendance Management: Automating attendance tracking.

The software will be accessible via web browsers and mobile devices, ensuring flexibility for users. It connects with external services such as authentication servers and AI API servers for facial recognition and secure user management. The AI-Assisted LMS will serve educational institutions ranging from small schools to large universities, offering an efficient and intelligent solution for managing learning and teaching processes.

**Overall description**

**Product perspective**

The AI-Assisted Learning Management System (LMS) is a product designed to address the evolving needs of educational institutions by integrating artificial intelligence into traditional LMS platforms. Unlike existing systems, this product incorporates advanced AI-driven capabilities such as content generation, personalized learning recommendations, automated evaluations, and performance analytics.

The system is developed as a standalone solution but can also integrate with existing educational tools, databases, and infrastructure. It serves as a replacement for outdated or manual systems and enhances existing LMS solutions by offering innovative AI-powered features.

**Key Context**

**Origin:** The system was conceptualized to modernize educational processes, providing a scalable, intelligent, and efficient platform that addresses challenges faced by educators and learners.

**New Capabilities:** The AI-Assisted LMS offers functionalities such as:

* AI-powered content generation and recommendations.
* Automated evaluations and real-time feedback.
* Comprehensive performance analytics for decision-making.

**Operating environment**

The AI-Assisted LMS is designed to operate effectively in a variety of hardware platforms, operating systems, and geographical locations. It ensures compatibility with modern client-side and server-side configurations, targeting educational institutions in urban and semi-urban areas of Pakistan and allowing for global scalability.

OE-1: The system shall support client-side devices, including desktop and laptop PCs with a minimum of 4 GB RAM and Intel i3 processors, mobile devices running Android 7.0+ or iOS 12+, and tablets with at least 2 GB RAM.

OE-2: The system will operate correctly on the following operating systems: Windows 10 and 11, macOS 10.15 (Catalina) and above, and Linux distributions such as Ubuntu 18.04+.

OE-3: Server-side operations shall run on cloud-based servers (AWS) with a minimum configuration of 16 GB RAM, quad-core processors, and 1 TB SSD, with auto-scaling for peak load management.

OE-4: The system shall be fully compatible with the following web browsers: Google Chrome (v80+), Mozilla Firefox (v70+), Microsoft Edge (Chromium-based), and Apple Safari (v12+).

OE-5: Databases shall be hosted on MongoDB Atlas (NoSQL) with geographically distributed backup centers for enhanced reliability.

OE-6: Hosting shall be implemented on AWS (EC2 and S3) with CDN support for fast content delivery and secure communication using TLS 1.3.

**Design and implementation constraints**

The development of the LMS is subject to specific technical and resource constraints to ensure cost-effectiveness and alignment with the project’s goals.

CO-1: The frontend of the system shall be developed using TypeScript with Next.js to ensure maintainability and scalability.

CO-2: The backend shall use NestJS to leverage its modular architecture for API design.

CO-3: AI models shall be developed using Python-based libraries, including TensorFlow and PyTorch, and must be optimized to run efficiently on mid-range GPUs to manage costs and low-bandwidth scenarios.

CO-4: The system shall integrate Llama APIs deployed on cloud (AWS Bedrock) for content generation.

CO-5: NoSQL (MongoDB) database shall be utilized for unstructured and structured data storage, respectively.

CO-6: All sensitive data shall be encrypted using AES-256, and the system must comply with GDPR and local data protection laws.

CO-7: The system shall be developed within a zero-budget framework, relying on free-tier cloud services and open-source tools wherever possible.

**Requirement identifying technique**

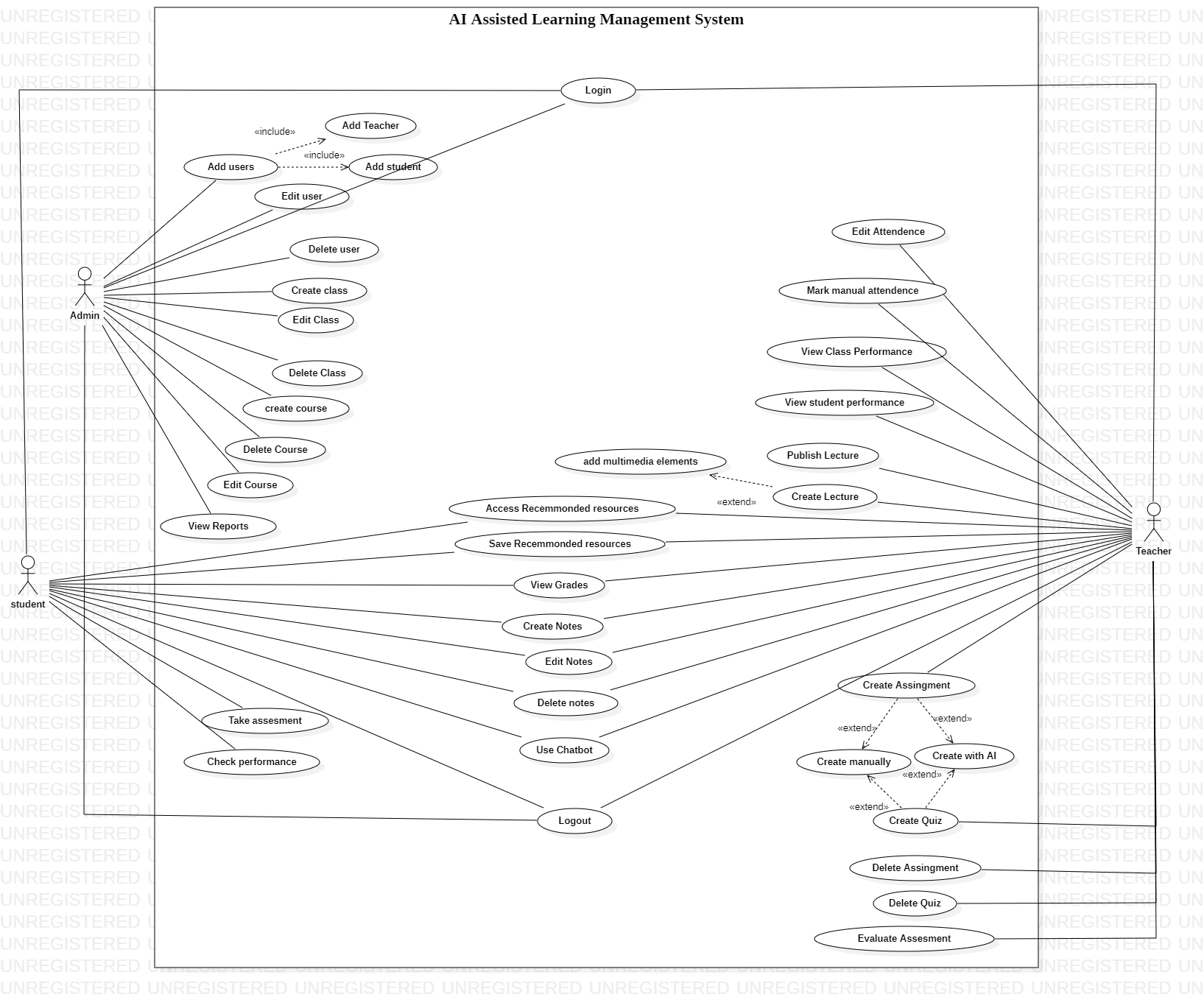
To identify the requirements for the AI-Assisted LMS, we utilized the **Use Case Analysis** technique. This approach is particularly effective for interactive end-user applications like an LMS, as it defines the interactions between various actors (students, teachers, and administrators) and the system.

By employing use cases, we were able to capture the specific functionalities required for different user roles in the LMS. For example:

* Teacher Use Case: Creating AI-assisted lecture content using predefined templates and generating quizzes with automated evaluations.
* Student Use Case: Accessing personalized learning recommendations, submitting quizzes, and receiving instant feedback.
* Admin Use Case: Managing user roles, monitoring system performance, and ensuring data security compliance.

This technique allowed us to clearly outline user-system interactions, ensuring that the LMS is user-centric and supports the needs of all stakeholders effectively. It also provided a solid foundation for deriving functional requirements and guiding system design.

**Use case diagram**

****

# Figure 1 Use case Diagram

**Use case Use case descriptions**

The table below provides a detailed example of a use case for an AI-Assisted Learning Management System (LMS). This structured template helps document and present user interactions and system functionalities in a clear and organized manner. The use case describes how specific system features are designed to meet user needs and outlines the steps, conditions, and outcomes associated with those interactions.

**Use Case Description: Access Recommended Resources**

# Table 1 UC-1 Access Recommended Resources

|  |  |
| --- | --- |
| **Use Case ID** | **UC-1** |
| **Use Case Name** | Access Recommended Resources |
| **Actors** | **Primary Actor**: Student **Secondary Actor**: Content Recommendation Module (System) |
| **Description** | The student accesses the system to view personalized recommendations for study materials based on their individual performance, preferences, and progress in the course. |
| **Trigger** | The student logs into the system and navigates to the "Recommended Resources" section. |
| **Preconditions** | 1. The student must be logged into the system. 2. The student must have completed some assessments or have progress tracked in the system. |
| **Postconditions** | 1. The student can view a list of personalized study materials. 2. The student can access the recommended resources. |
| **Normal Flow** | **1.0 Access Recommended Resources** 1. The student logs into the system. 2. The student navigates to the "Recommended Resources" section, either from the dashboard or the course-specific page. 3. The system analyzes the student's performance, preferences, and progress to generate a list of personalized recommendations. 4. The system presents the recommended resources (e.g., books, videos, quizzes, assignments). 5. The student reviews the recommendations, which may include explanations, additional reading materials, or practice exercises. 6. The student selects a resource to access. 7. The system directs the student to the selected resource or content. |
| **Alternative Flows** | **1.1 No Recommendations Available** 1. If the system cannot generate recommendations due to insufficient data, it displays: "No recommendations available at this time. Please complete more assessments." **1.2 No Internet Connection** 1. If the student experiences an internet issue, the system prompts: "Unable to access resources due to connection issues. Please try again later." |
| **Exceptions** | **1.0.E1 System Error While Accessing Resources** 1. The system displays an error message: "Unable to access the requested resource. Please try again." 2. The student may retry accessing the resource. |
| **Business Rules** | **BR-1**: Recommended resources must align with the student's current course syllabus and performance data. **BR-2**: The system may only recommend resources that are part of the officially approved content library. |
| **Assumptions** | 1. The system continuously tracks and updates the student's performance data. 2. Recommended resources will always be available to students as long as they have internet access. |

**Use Case Description: Save Recommended Resources**

# Table 2 UC-2 Save Recommended Resources

|  |  |
| --- | --- |
| **Use Case ID** | **UC-2** |
| **Use Case Name** | Save Recommended Resources |
| **Actors** | **Primary Actor**: Student **Secondary Actor**: Content Recommendation Module (System) |
| **Description** | The student saves recommended study resources for later review, allowing them to bookmark content they find helpful or wish to return to. |
| **Trigger** | The student clicks on the "Save for Later" button next to a recommended resource. |
| **Preconditions** | 1. The student must be logged into the system. 2. The student must have accessed recommended resources. |
| **Postconditions** | 1. The resource is saved in the student's "Saved Resources" section. 2. The student can access the saved content at any time. |
| **Normal Flow** | **1.0 Save Recommended Resources** 1. The student logs into the system and accesses the recommended resources section. 2. The student reviews the available recommendations and finds a resource they wish to save. 3. The student clicks on the "Save for Later" button next to the resource. 4. The system saves the resource to the student's personalized "Saved Resources" section. 5. The system displays a confirmation message: "Resource saved successfully." 6. The student can later navigate to their "Saved Resources" section to view and access all saved content. |
| **Alternative Flows** | **2.1 Resource Already Saved** 1. If the student tries to save a resource already saved, the system displays: "This resource is already saved in your list." **2.2 Insufficient Data to Save** 1. If the system fails to save the resource, it displays: "Unable to save resource at this time. Please try again later." |
| **Exceptions** | **2.0.E1 System Error During Save Operation** 1. If an error occurs while saving the resource, an error message is displayed: "Error occurred while saving. Please try again later." |
| **Business Rules** | **BR-1**: Resources can only be saved if they are part of the official recommended list. **BR-2**: Saved resources are accessible until the student marks them as "Completed" or removes them manually. |
| **Assumptions** | 1. The student may save multiple resources for later review. 2. Saved resources will not expire unless the course or content is deleted by the administrator. |

**Use Case Description: Create lecture**

# Table 3 UC-3 Create Lecture

|  |  |
| --- | --- |
| **Field** | **Details** |
| **Use Case ID** | UC-3 |
| **Use Case Name** | Create Lecture |
| **Actors** | **Primary Actor:** Teacher  **Secondary Actor:** Content Generation System |
| **Description** | The teacher creates lecture content using the AI-assisted content generation feature. Optionally, the teacher can manually edit AI-generated content. |
| **Trigger** | The teacher clicks the **"Create Lecture"** button in the course module and selects "Create with AI." |
| **Preconditions** | 1. The teacher must be logged into the LMS.  2. The teacher must have access to the specific course within the LMS. |
| **Postconditions** | The lecture is saved as a draft under the "Saved" tab.  The lecture is published under the "Published" tab for student access. |
| **Normal Flow** | 1. The teacher navigates to the "Courses" tab in the dashboard.  2. The teacher selects a course and clicks the "Create Lecture" button.  3. The teacher selects the "Create with AI" option.  4. The system prompts the teacher to enter inputs such as Title, Description, duration, Chapter, Academic Level, Learning Outcomes, Preferred Formats, Attachments. Objective, and Reference.  5. The AI generates lecture content based on the inputs provided and displays it for review.  6. The teacher optionally edits the AI-generated content using the built-in editor.  7. The teacher clicks "Save" to save as a draft or "Save and Publish" to make it available to students. |
| **Alternative Flows** | **AF-1:** Teacher selects "Create Manually":  1. Instead of "Create with AI," the teacher selects "Create Manually."  2. The system displays a form. |
|  | 3. The teacher manually enters all lecture details and content.  4. The teacher clicks "Save" or "Save and Publish" to complete the process. |
| **Exceptions** | **EX-1:** AI Generation Error:  1. The AI fails to generate content due to a technical issue.  2. The system suggests retrying or switching to manual. |
|  | **EX-2:** Save/Publish Error:  1. The system encounters an error while saving or publishing the lecture.  2. The system advises retrying. |
| **Business Rules** | 1. AI-generated content must align with inputs and academic standards.  2. Edited content must pass validation before saving or publishing. |
| **Assumptions** | 1. Teachers have reliable internet access during the creation process.  2. The AI system is functional and accessible. |

**Use Case Description: Create Assignment**

# Table 4 UC-4 create assignment

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Use Case ID** | UC-4 |
| **Use Case Name** | Create Assignment |
| **Actors** | Primary Actor: Teacher Secondary Actor: System |
| **Description** | The teacher creates and schedules an assignment by entering details, uploading resources, and applying rubrics. |
| **Trigger** | The teacher clicks the "Create Assignment" button. |
| **Preconditions** | 1. The teacher must be logged into the system. 2. The teacher must have appropriate permissions to create assignments. |
| **Postconditions** | 1. The assignment is saved in the system. 2. The assignment is scheduled for a specified date and time. 3. A confirmation message is displayed. |
| **Normal Flow** | 1. The teacher logs into the system and navigates to the "Assignments" tab. 2. The teacher clicks "Create Assignment." 3. The system displays the "Create Assignment" form. 4. The teacher fills out the form with details such as Assignment Name, Rubrics, References, Due Date, etc. 5. The teacher uploads additional resources or enters assignment content. 6. The teacher customizes or applies a predefined grading rubric. 7. The teacher reviews the details and selects "Save and Publish" or "Save." |
| **Alternative Flows** | **AF-1: Missing Mandatory Fields** 1. The system highlights incomplete fields and prompts the teacher to fill them in. 2. The teacher enters the missing information and continues. **AF-2: Invalid Due Date** 1. The system shows an error if the due date is invalid. 2. The teacher corrects the date and proceeds. |
| **Exceptions** | **EX-1: System Error** 1. The system fails to save the assignment. 2. The system displays an error message and prompts a retry. |
| **Business Rules** | BR-1: Assignments must include either instructions or attached resources. BR-2: Assignments must have a title and a valid due date. |
| **Assumptions** | 1. Teachers will create assignments within course timelines. 2. Stable internet connectivity during assignment creation. |

**Use Case Description: Create Quiz**

# Table 5 UC-5 Create quiz

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Use Case ID** | UC-5 |
| **Use Case Name** | Create Quiz |
| **Actors** | Primary Actor: Teacher Secondary Actor: System |
| **Description** | The teacher creates and schedules a quiz by entering questions, defining correct answers, and assigning marks. |
| **Trigger** | The teacher clicks the "Create Quiz" button. |
| **Preconditions** | 1. The teacher must be logged into the system. 2. The teacher must have appropriate permissions to create quizzes. |
| **Postconditions** | 1. The quiz is saved in the system. 2. The quiz is scheduled for a specified date and time. 3. A confirmation message is displayed. |
| **Normal Flow** | 1. The teacher logs into the system and navigates to the "Quizzes" tab. 2. The teacher clicks "Create Quiz." 3. The system displays the "Create Quiz" form. 4. The teacher fills out details such as Quiz Title, Rubrics, References, Due Date, etc. 5. The teacher adds questions (e.g., MCQs, Short Questions, Fill-in-the-Blanks) using the question editor. 6. The teacher defines correct answers and assigns marks per question. 7. The teacher reviews the quiz structure and selects "Save and Publish" or "Save." |
| **Alternative Flows** | **AF-1: Missing Mandatory Fields** 1. The system highlights incomplete fields and prompts the teacher to fill them in. 2. The teacher enters the missing information and continues. **AF-2: Invalid Date or Time** 1. The system shows an error if the scheduled date is invalid. 2. The teacher corrects the date and proceeds. |
| **Exceptions** | **EX-1: System Error** 1. The system fails to save the quiz. 2. The system displays an error message and prompts a retry. |
| **Business Rules** | BR-1: Quizzes must include at least one question. BR-2: Quizzes must have a valid scheduled date and time. |
| **Assumptions** | 1. Teachers will create quizzes within course timelines. 2. Stable internet connectivity during quiz creation. |

**Use Case Description: Evaluate assessment**

# Table 6 UC-6 Evaluate assignment

|  |  |
| --- | --- |
| **Use Case ID** | **UC-6** |
| **Use Case Name** | Evaluate assessment |
| **Actors** | Primary Actor: Teacher  Secondary Actor: Standard Evaluation Module (System) |
| **Description** | The teacher evaluates student submissions for assignments or quizzes. The system supports AI-assisted grading and allows manual adjustments to scores and feedback before finalizing and publishing results. |
| **Trigger** | The teacher accesses submitted assessments for evaluation. |
| **Preconditions** | 1. Students have submitted their assignments/quizzes.  2. The teacher is logged into the system and has access to the evaluation module. |
| **Postconditions** | 1. Scores and feedback are recorded and saved.  2. The results are available for publishing to students. |
| **Normal Flow** | 1. The teacher logs into the system and navigates to the "Pending Evaluations" section. 2. The system displays a list of submitted assignments/quizzes. 3. The teacher selects a specific assessment to evaluate. 4. The system performs AI-assisted grading (if enabled) based on predefined rubrics and displays the scores. 5. The teacher reviews the AI-assigned scores and feedback. 6. The teacher performs any of the following actions: Adjust scores manually, Edit or add detailed feedback for each student. 7. The teacher clicks the "Save Evaluation" button. 8. The system saves updated scores and feedback. 9. The teacher clicks "Publish Results" to make scores and feedback visible to students. 10. The system displays a confirmation message: "Results published successfully." |
| **Alternative Flow** | 2.1 No Submissions Available   1. If no submissions are available, the system displays a message: "No submissions found for evaluation."   2.2 Incomplete AI Grading   1. If AI grading fails for any question or response, the system notifies the teacher: "AI grading incomplete. Manual evaluation required." 2. The teacher proceeds to manually grade the affected questions.   2.3 Partial Evaluation   1. If the teacher saves the evaluation without publishing, the system marks the submissions as "Partially Evaluated. |
| **Exceptions** | 2.0.E1 System Error During Grading   1. If an error occurs during AI-assisted grading, the system displays: "Error occurred during grading. Please grade manually." |
| **Business Rules** | BR-1: Scores must be saved before publishing results.  BR-2: Teachers can override AI-assigned scores and feedback.  BR-3: Final scores and feedback cannot be edited after publishing. |
| **Assumptions** | 1. Teachers will cross-check AI-assigned scores for accuracy. 2. Students expect timely results after submission deadlines. |

**Functional Requirements**

This section outlines the functional requirements of the AI-Assisted Learning Management System (LMS), presented in a natural language style. The functional requirements are organized by key features such as Personalized Learning, Automated Evaluations, Performance Analytics, Attendance Management, and Content Generation. These requirements describe how the system will behave, including the responses to user inputs and anticipated error conditions, such as invalid input handling or failure scenarios.

**Functional Requirement**

For each feature, individual functional requirements have been labeled and organized with unique identifiers for clarity. The following template has been used to document these requirements in a clear and consistent manner.

**Feature 1: Personalized Learning**

# Table 7: Functional Requirement 1.1

|  |  |
| --- | --- |
| **Field** | **Details** |
| **Identifier** | FR-PL-001 |
| **Title** | Personalized Content Recommendations |
| **Requirement** | The system shall provide personalized learning material recommendations to each student based on their learning history, preferences, and performance metrics. |
| **Source** | Design Team |
| **Rationale** | To enhance student engagement and learning outcomes by providing relevant content. |
| **Business Rule** | Recommendations must be aligned with the curriculum. |
| **Dependencies** | FR-PL-002, FR-PA-001 |
| **Priority** | High |

# Table 8: Functional Requirement 1.2

|  |  |
| --- | --- |
| **Field** | **Details** |
| **Identifier** | FR-PL-002 |
| **Title** | Interactive Material Generation |
| **Requirement** | The system should allow teachers to generate interactive content such as quizzes and flashcards tailored to students' needs. |
| **Source** | Teacher Feedback |
| **Rationale** | To facilitate active learning through engaging materials. |
| **Business Rule** | Materials must adhere to institutional guidelines. |
| **Dependencies** | FR-CG-001 |
| **Priority** | Medium |

**Feature 2: Automated Evaluations**

# Table 9: Functional Requirement 2.1

|  |  |
| --- | --- |
| **Field** | **Details** |
| **Identifier** | FR-AE-001 |
| **Title** | Instant Feedback Mechanism |
| **Requirement** | The system shall provide instant feedback to students after quiz submissions, highlighting areas of improvement. |
| **Source** | Assessment Tools Research |
| **Rationale** | To support continuous learning through real-time feedback. |
| **Business Rule** | Feedback must include references to relevant study materials. |
| **Dependencies** | FR-PL-001, FR-PA-002 |
| **Priority** | High |

# Table 10: Functional Requirement 2.2

|  |  |
| --- | --- |
| **Field** | **Details** |
| **Identifier** | FR-AE-002 |
| **Title** | Self-Evaluation Tools |
| **Requirement** | The system shall enable students to perform self-assessments using AI-driven evaluations of written and multiple-choice responses. |
| **Source** | Student Feedback Surveys |
| **Rationale** | To empower students with self-directed learning capabilities. |
| **Business Rule** | Self-assessments must not count toward final grades unless explicitly permitted. |
| **Dependencies** | None |
| **Priority** | Medium |

**Feature 3: Performance Analytics**

# Table 11: Functional Requirement 3.1

|  |  |
| --- | --- |
| **Field** | **Details** |
| **Identifier** | FR-PA-001 |
| **Title** | Analytics Dashboard for Teachers |
| **Requirement** | The system shall provide a dashboard with visualized analytics on student performance trends, attendance, and engagement. |
| **Source** | Requirements Analysis Document |
| **Rationale** | To assist teachers in making data-driven decisions. |
| **Business Rule** | Data must be anonymized when aggregated. |
| **Dependencies** | FR-AM-001 |
| **Priority** | High |

# Table 12: Functional Requirement 3.2

|  |  |
| --- | --- |
| **Field** | **Details** |
| **Identifier** | FR-PA-002 |
| **Title** | Predictive Performance Insights |
| **Requirement** | The system shall use AI to predict students’ future performance based on their learning patterns and attendance records. |
| **Source** | Predictive Modeling Research |
| **Rationale** | To enable proactive intervention strategies. |
| **Business Rule** | Predictions must be flagged as estimates and not definitive results. |
| **Dependencies** | FR-PA-001 |
| **Priority** | High |

**Feature 4: Content Generation**

# Table 13: Functional Requirement 4.1

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Identifier** | CG-01 |
| **Title** | AI-Assisted Lecture Creation |
| **Requirement** | The system must allow teachers to generate lecture content automatically by providing inputs such as learning objectives, topics, and references. |
| **Source** | Design Team |
| **Rationale** | Automating lecture content creation helps reduce the workload and improves efficiency for teachers. |
| **Business Rule** | The generated content must align with provided inputs and adhere to academic standards. |
| **Dependencies** | Requires a stable connection to the AI model and teacher access to the LMS. |
| **Priority** | High |

# Table 14: Functional Requirement 4.2

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Identifier** | CG-02 |
| **Title** | Manual Editing of AI-Generated Content |
| **Requirement** | The system must allow teachers to edit and modify the AI-generated lecture content before saving or publishing. |
| **Source** | Teacher Feedback |
| **Rationale** | Teachers may want to customize AI-generated lectures to better fit their teaching style and student needs. |
| **Business Rule** | Edited content should be validated for required fields before saving or publishing. |
| **Dependencies** | Requires successful generation of lecture content using the AI feature. |
| **Priority** | High |

# Table 15: Functional Requirement 4.3

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Identifier** | CG-03 |
| **Title** | Save and Publish Options |
| **Requirement** | The system must provide options to save the lecture as a draft or publish it directly for student access. |
| **Source** | Teacher Feedback |
| **Rationale** | Teachers may want flexibility to refine content progressively or make it immediately available to students. |
| **Business Rule** | Drafts are saved under the "Saved" tab and visible only to the teacher; published lectures appear under the "Published" tab for students. |
| **Dependencies** | Edited or AI-generated content must be validated before saving or publishing. |
| **Priority** | Medium |

**Non-Functional Requirements**

This section outlines the quality requirements that ensure the system meets performance, usability, and reliability standards.

**Usability**

USABILITY-1: The system shall allow users to access the Content Recommendation module and receive recommendations within 2 clicks of the main dashboard.

USABILITY-2: The Standard Evaluation module shall provide clear, step-by-step instructions for completing evaluations to minimize user confusion.

USABILITY-3: The system interface shall support accessibility features compliant with WCAG 2.1 Level AA standards to accommodate users with disabilities.

USABILITY-4: The mobile version of the LMS shall provide consistent and intuitive user experience compared to the web version.

USABILITY-5: Self-Evaluation modules shall include tooltips to guide users on their functionality.

**Reliability**

RELIABILITY-1: The system’s uptime shall be 99% over the 24/7 operational period.

RELIABILITY-2: The Database Server shall recover automatically from failures within 30 seconds.

RELIABILITY-3: The system shall support data backup every 12 hours to prevent data loss.

**Security**

SECURITY-1: The system requires multi-factor authentication (MFA) for users accessing sensitive modules like Analytics and Attendance.

SECURITY-2: Data transferred between Client Devices and servers shall be encrypted using HTTPS/TLS 1.3 protocols.

SECURITY-3: All user data stored in the Database Server shall be encrypted using AES-256 encryption standards.

SECURITY-4: The system shall logs all user activities, including logins, access, and modifications, for auditing purposes.

**Performance**

PERFORMANCE-1: 95% of webpages generated by the LMS shall load within 3 seconds over a 20 Mbps or faster internet connection.

PERFORMANCE-2: The Content Generation module shall process and generate learning content within 10 seconds for standard text inputs.

PERFORMANCE-4: The system shall support a minimum of 500 concurrent users accessing modules like Analytics and Content Recommendation without performance degradation.

**Scalability**

SCALABILITY-1: The system shall be able to scale horizontally to support up to 10,000 concurrent users by adding additional Application Servers.

SCALABILITY-2: The system shall ensure seamless performance even as the database size grows up to 1 million records.

**Maintainability**

MAINTAINABILITY-1: The system shall allow new AI components to be integrated into the Application Server with minimal changes to existing modules.

MAINTAINABILITY-2: Code shall follow industry-standard design patterns and include detailed documentation to ensure maintainability.

MAINTAINABILITY-3: System logs and error messages shall be stored to facilitate troubleshooting and future system updates.

**Project Gantt chart**

A screenshot of a computer

Description automatically generated

Figure 2 Gantt chart

**References**

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.

IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans.

IEEE Std 730.1-1995, IEEE Guide for Software Quality Assurance Planning.

IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans.