Learning management system

Supplementary Specification

Version 1.0

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 25/11/2024 | 1.0 | Final version | Nguyễn Thị Hiền Anh Trần Thị Kim Ngân Vũ Thị Liên Nguyễn Duy Thùy Dương Đào Quang Khánh Nguyễn Quang Dũng Tạ Thị Tuyết Lan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Functionality 5

2.1 For all role 5

2.2 For teacher/Instructor 5

2.3 For student 6

2.4 For admin 7

3. Usability 8

4. Reliability 8

5. Performance 8

6. Supportability 9

7. Design Constraints 9

8. Online User Documentation and Help System Requirements 10

9. Purchased Components 10

10. Interfaces 11

10.1 User Interfaces 11

10.2 Hardware Interfaces 11

10.3 Software Interfaces 12

10.4 Communications Interfaces 12

11. Licensing Requirements 12

12. Legal, Copyright, and Other Notices 12

13. Applicable Standards 12

Supplementary Specification

# Introduction

## Purpose

The purpose of this **RD\_Supplementary Specification** is to document additional system requirements for the Learning Management System (LMS) that complement the Use-Case and Functional Specifications. This includes requirements related to legal, regulatory compliance, quality attributes (such as usability, reliability, performance, and supportability), and design constraints. This document ensures that the system is designed and implemented to meet not only functional needs but also operational and user experience goals.

## Scope

This specification applies to the **LMS project** and defines the requirements that are not easily captured in the use cases. These include:

**Quality attributes**: Usability, reliability, performance, and supportability.

**Legal and regulatory compliance**: Ensuring the system adheres to applicable laws and standards.

**Design constraints**: Specific requirements for architecture, tools, and technologies used in development.

**System interfaces**: Describing user interfaces, hardware/software interfaces, and communication protocols.

## Definitions, Acronyms, and Abbreviations

N/A

## References

**Business Use-Case Specification: Manage Courses Content**, Group 10, Version 1.0, 25/11/2024.

**Use-Case Specification: Manage Courses Content**, Group 10, Version 1.0, 25/11/2024.

**ISO/IEC 25010:2011**: Software Engineering — Systems and Software Quality Requirements and Evaluation (SQuaRE) — Quality Model.

**WCAG 2.1**: Web Content Accessibility Guidelines 2.1.

**OWASP**: Open Web Application Security Project (OWASP) guidelines.

## Overview

The Supplementary Specification document for the Unica Online Learning Platform is a multifaceted guide, systematically divided into distinct sections to comprehensively address key facets of system development and functionality. Among these sections are Functionality, which elucidates system requirements in a natural language style, organized by features or users; Usability, which delves into factors impacting user training time for optimal website productivity; and Reliability, detailing availability metrics, mean time between failures, system downtime post-failure, accuracy benchmarks, and defect rates. The Performance section outlines key efficiency and responsiveness characteristics, while Design Constraints identifies limitations affecting the system's architecture and development. The Online User Documentation and Help System Requirements segment outlines prerequisites for user support and accessibility. Purchased Components identifies external elements integral to the Unica system, and Licensing Requirements delves into stipulations for system use. Legal, Copyright, and Other Notices addresses pertinent legal considerations, and Applicable Standards ensures alignment with industry norms by referencing specific sections of relevant standards. Together, these sections create a comprehensive framework guiding the development, implementation, and ongoing support of the Learning Management System in accordance with industry standards and user expectations.

# Functionality

## For all role: Login/Logout

Log In: Users (students, instructors, and administrators) must be able to log into the LMS using their valid username and password. Upon logging in, they are directed to their respective home pages based on their roles. If the user fails to enter the correct credentials after 5 attempts, their account will be temporarily locked for 30 minutes to prevent unauthorized access. The system must store passwords securely using encryption, and users must have the option to reset their passwords via email.

Log Out: Users must be able to securely log out of the system, ensuring that their session is terminated. The session should automatically expire after 15 minutes of inactivity. Upon successful logout, users are redirected to the login page, ensuring that no data is left open or accessible by unauthorized individuals.

* 1. **For Teacher/Instructor**

**Manage Course Content:** Instructors must have the ability to create, edit, and delete course modules and their contents. They can add learning materials such as documents, videos, and links to specific course modules. Additionally, instructors can organize these materials into sections (e.g., weekly topics) and update the course description or format. They can also delete outdated or irrelevant content and adjust access permissions for certain materials based on the needs of the course.

**Grade Assignments:** Instructors must be able to review student submissions for assignments and quizzes, and assign grades based on predefined criteria. They can provide written feedback, upload grades manually, and make adjustments if necessary. The grading system must be flexible, supporting numeric, letter, and rubric-based grades. Instructors should also be able to set up grading scales for different assignments.

**Monitor Student Progress**: Instructors must be able to track student activity and performance. This includes viewing detailed reports on student progress, assignment submissions, quiz results, and course completion rates. Instructors should receive alerts for students who are falling behind or have incomplete assignments. The system should provide real-time updates whenever a student submits a quiz, completes a module, or updates their progress.

**Post Announcements:** Instructors must be able to create and post course announcements, such as updates on upcoming deadlines, changes to the course schedule, or other important information. Announcements can be text-based or include multimedia content like images or videos. They should be visible to all enrolled students and should be time-sensitive, with the ability to set an expiration date for each announcement.

**Manage Discussions:** Instructors must have the ability to moderate forums or group discussions. They can review posts, reply to student queries, and remove inappropriate or off-topic content. Instructors should also have the option to set specific discussion topics or questions for students to respond to. This ensures that the discussions are productive and aligned with course objectives.

* 1. **For Student**

**View Course:** Students must be able to access course-related content, including videos, PDFs, presentations, and other materials uploaded by the instructor. Access to these resources should be restricted to enrolled students only, with materials organized by modules or topics. Each resource must be presented in a structured, easy-to-navigate layout.

**Submit Assignments and Quizzes:** Students should be able to submit their work for assignments and quizzes. This includes uploading files such as PDFs or Word documents or entering text directly into the system. The system must support various file formats and set limitations on file sizes (e.g., up to 50MB per file). The system will automatically reject late submissions after the due date. Students will receive confirmation upon successful submission, with a timestamp indicating the exact time of submission.

**Take Test/Quiz:** Students must be able to take quizzes or tests for course evaluation. The system will present quizzes with various question types (e.g., multiple-choice, short-answer, essay). Each quiz will have a time limit (e.g., 60 minutes). Once a test begins, students cannot pause or resume unless explicitly allowed by the instructor. The system will track responses in real-time and prevent submission after the time limit expires.

**Track Progress:** Students should be able to view their grades, track the completion of modules, and monitor their overall progress throughout the course. The system will display progress in terms of percentage completed, grades received for assignments and quizzes, and other relevant metrics. This information should be updated in real-time as the instructor grades assignments or marks modules as completed.

**Participate in Discussions:** Students must have access to course forums or group chats where they can engage in peer-to-peer or instructor-led discussions. Students can post new threads or respond to existing discussions, fostering collaborative learning. Instructors have the ability to moderate the discussions and remove inappropriate posts. Students will receive notifications of new posts and can reply directly to specific threads.

**View Announcements:** Students should be able to view course-specific or system-wide announcements posted by the instructor or administrator. These announcements should be easily accessible from the student dashboard and course page. Announcements will be time-sensitive and should be automatically archived or removed after a specific period, but students should also be notified via email or in-app notifications when a new announcement is made.

* 1. **For admin**

**Manage Users:** Administrators must be able to add, edit, or delete user accounts and assign roles such as student, instructor, or administrator. This includes managing user information such as usernames, passwords, email addresses, and roles. Administrators must also be able to reset user passwords and unlock accounts that are temporarily locked after failed login attempts.

**Manage Course:** Administrators should be able to create new courses, assign instructors to courses, and enroll students. Courses should be categorized (e.g., core, elective) and have specific start and end dates. Administrators can assign instructors to specific courses and manage student enrollment either manually or through self-enrollment.

**Monitor System Performance:** Administrators are responsible for tracking the system’s health, uptime, and performance. They must have access to logs, reports, and other diagnostic tools that provide information on system usage and potential issues. The system should automatically log errors and provide real-time alerts for performance degradation or failures.

**Generate Reports**: Administrators should be able to generate detailed reports about system usage, including active users, course activity, student engagement, and performance metrics. These reports can be used for auditing purposes, improving system functionality, or identifying areas for improvement.

**Configure System Settings:** Administrators should be able to configure system-wide settings, such as themes, permissions, and integrations with third-party services. This includes managing the overall look and feel of the LMS, setting default permissions for different user roles, and configuring external services like video hosting or cloud storage.

**Manage Course Catalog**: Administrators must oversee the course catalog, ensuring courses are available to students and properly categorized. They can create, modify, or remove courses, organize courses into categories, and manage course availability. The course catalog should be organized and easily navigable for both students and instructors.

**Post General Announcements:** Administrators should be able to post system-wide announcements to inform all users about important events or changes, such as system maintenance or major updates. These announcements should be visible on the LMS dashboard and can include text, images, and links.

# Usability

## The LMS must be user-friendly and provide an intuitive interface for both instructors and students.

## The time to train instructors should not exceed 2 hours for basic usage, and 5 hours for advanced course management tasks.

## Task times for common operations, such as uploading materials, creating assignments, and grading, should be optimized to ensure instructors can complete tasks within 10 minutes.

## The system must adhere to industry-standard design guidelines, including IBM CUA standards and Microsoft’s GUI standards.

## The LMS must be fully accessible, meeting the requirements of WCAG 2.1 to support users with disabilities.

# Reliability

## Availability:

## The LMS must ensure 99.5% system availability during business hours (8 AM to 6 PM).

## System should allow planned maintenance during off-peak hours, with a maximum downtime of 2 hours per month.

## Mean Time Between Failures (MTBF):

## The system must have a MTBF of at least 1,000 hours to minimize downtime.

## Mean Time to Repair (MTTR):

## In the event of a failure, the system should be restored within 4 hours.

## Accuracy:

## The system must process and deliver accurate course content, grade submissions correctly, and display activity results without error.

## Bug Rate:

## The LMS should have a maximum of 1 bug per 1,000 lines of code (KLOC) during post-launch operations.

## Critical bugs must be resolved within 24 hours, while minor bugs can be fixed during the next scheduled update.

# Performance

## Response Time

## The LMS should respond to user inputs (e.g., opening a course, submitting an assignment) within 2 seconds on average.

## Course content pages should load within 5 seconds, even with large multimedia resources.

## Throughput:

## The system should handle up to 10,000 concurrent users without noticeable performance degradation.

## Capacity

## The LMS should be capable of managing up to 100,000 courses, with each course hosting 500 resources and 5,000 students.

## Degradation Mode:

## If the system experiences high load, it should continue to function in a degraded mode, allowing users to access previously loaded course materials while new data is processed in the background.

## Resource Utilization:

## The system should optimize memory usage (no more than 1 GB per active session) and disk space for efficient operation.

# Supportability

## Maintainability

## The system should include logging capabilities to track errors and facilitate debugging during maintenance.

## Code should follow established coding standards (e.g., PEP 8 for Python) to ensure maintainability.

## System Monitoring

## The system should provide real-time monitoring of server health, user activity, and resource usage to assist in troubleshooting.

## Documentation:

## The LMS must include comprehensive user manuals for both instructors and administrators, covering all key functionalities and troubleshooting steps.

# Design Constraints

# Technology Stack

# Software languages: javascript

Development tool: Visual Studio Code, Mongodb compass

Class libraries: express, handlebars, mongoose

Architecture: MVC

Tool to work in teams: Microsoft Office, Jira, Git, Balsamiq mockups

# Platform Compatibility:

# The LMS must support the latest versions of major browsers (Chrome, Firefox, Safari, Edge) and be fully compatible with iOS and Android mobile devices.

# Security:

The LMS must comply with **GDPR** regulations to ensure the protection of user data, including encryption of all sensitive data (e.g., passwords, personal information).

# Online User Documentation and Help System Requirements

Online support concerning administration issues & user tutorials including “How to” and “Frequently asked questions” guides will be available.

Readme files and release notes are to be delivered to the customer in each release.

User guides and Administration guides are to be provided per customer request.

# Purchased Components

N/A

# Interfaces

## User Interfaces

## For Students

## View Course: Access course-related files (videos, PDFs, presentations, etc.).

## Submit Assignments and Quizzes: Submit assignment work.

## Take test/quiz: take quizzes and tests for evaluation.

## Track Progress: View grades, completed modules, and overall course progress.

## Participate in Discussions: Engage in forums or group chats for collaborative learning. (Khánh)

## View Announcements: Stay updated on course or system-wide news and events.

## Lecturer/Instructor

## Manage Courses Content: Edit course settings, Add new module, Add new section (Activities/Resources), Delete module/section, Move section/module

## Grade Assignments: Review submissions and assign scores or feedback.

## Monitor Student Progress: View student activity, progress reports, and performance.

## Post Announcements: Share updates or important notifications with enrolled students.

## Manage Discussions: Moderate forums or respond to student queries.

## Administrators

## Manage Users: Add, edit, or delete user accounts and assign roles.

## Manage Course: Create course, add instructors and students

## Monitor System Performance: Track system health, uptime, and logs for troubleshooting.

## Generate Reports: Create detailed usage, performance, or engagement reports.

## Configure System Settings: Set global configurations like themes, permissions, or integrations.

## Manage Course Catalog: Oversee course availability and organize categories.

## Post general announcements

## Hardware Interfaces

Server infrastructure:

Logical Structure:

Server architecture: Load-balanced cluster

Web server software and configuration: Tomcat

Database backend: MongoDB

## Software Interfaces

The system will interact with Linux-based servers for hosting the application and cloud-based storage solutions (e.g., AWS S3) for file management.

## Communications Interfaces

LAN: Local Area Network

HTTPS: Hypertext Transfer Protocol Secure

# Licensing Requirements

N/A

# Legal, Copyright, and Other Notices

© Learning Management System, 2023. All Rights Reserved.

# Applicable Standards

ISO/IEC 9126-1: Software Engineering - Product Quality: The system must meet the quality model standards defined by ISO, ensuring functionality, reliability, and usability.

WCAG 2.1: The LMS must comply with the Web Content Accessibility Guidelines to support accessibility for users with disabilities.