Software Requirements Specification

For Online Learning Management System

Name	ID
Ruba jamal alzahrani	2220004549
Budur Khalaf Alqattan	2230040030
Zahra alshuwiki	2220006062
Raghad Lisloum	2220004295
Reef Alwarthan	2220006486
Albandari Aldossary	2220005834

Darah alqahtani	**2210004542**

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a detailed description of the requirements for the Online Learning Management System (OLMS) developed by Group 2 for the CIS 417 course. The document outlines the intended features and functions necessary to address the evolving needs of online education. The purpose of OLMS is to create an accessible, user-friendly platform that overcomes the scalability, usability, and interactivity limitations found in many existing learning management systems, such as Canvas, Blackboard, and Rippling. This SRS details the system's objectives, scope, interfaces, and specific requirements, ensuring the platform meets both educational and operational standards.

1.2 Document Conventions

This document adheres to IEEE standards for SRS formatting. Each requirement is prioritized and presented in structured sections, including functional and non-functional requirements, ensuring readability and ease of navigation for all audiences.

1.3 Intended Audience and Reading Suggestions

The intended audience for this SRS includes project developers, system analysts, educational administrators, and other stakeholders invested in the OLMS. Project managers may focus on the scope and requirements sections, while developers and testers will benefit from the functional and interface requirements. Readers unfamiliar with technical details may start with the overview sections for context before exploring detailed requirements.

1.4 Product Scope

OLMS aims to enhance the online learning experience by addressing critical user pain points in current LMS platforms. This system will provide a robust, scalable environment that supports a range of functionalities, including course catalog browsing, enrollment management, and progress tracking. Designed with mobile compatibility and interactivity in mind, OLMS seeks to improve accessibility and engagement for students, fostering an inclusive and adaptable learning environment.

1.5 References

1) [1] Ruth Thornton | Global Product Manager, "How to create a course catalog system for professional development," *Instructure.* https://www.instructure.com/en-au/resources/blog/how-create-course-catalog-system-professional-development (accessed Nov. 07, 2024).

2) [2]"Catalog Management", *Coursedog*, 2023. https://www.coursedog.com/products/catalog-management (accessed Nov. 07, 2024).
3) [3]Courseleaf, "Publish a Catalog that Drives Student Success.", *Courseleaf*, 2024. https://www.courseleaf.com/software/cat/ (accessed Nov. 07, 2024).
4) [4] OneSeven Tech, "How to Write a Software Requirements Specification (SRS Document)? - OneSeven Tech (OST) | Digital Product Studio," O*neSeven Tech (OST).* https://www.oneseventech.com/blog/how-to-write-software-requirements. (accessed Nov. 08, 2024).
5) [5] Requirements.co.za , "External Interface – Requirements," *Requirements.co.za*, 2024. https://www.requirements.co.za/external-interface/ (accessed Nov. 09, 2024).
6) [6] Jelvix, "The Full Guide To Software Requirements Specification Documentation," *Jelvix*. https://jelvix.com/blog/software-requirements-specification (accessed Nov. 10, 2024).

2. Overall Description

2.1 Product Perspective

The Online Learning Management System (OLMS) main purpose is to help students solve and look for course subjects in an easier way. Also, to keep up with the assignments overall it is intended to enhance the learning systems and tools. This product is enhanced to overcome the limitations of older systems, users of this system are many such as students or tutors. Students can browse courses , check their grades, watch recorded lectures, and more, while teachers can upload large files and videos and give students feedback or grade students assignments and quizzes It makes the whole process straightforward and user-friendly. Unlike many existing systems that are difficult to use and hard to control OLMS boosts modern interfaces with interactive features.OLMS is focused on transforming online education to be more user-friendly, interactive, and available to all individuals. It involves improving the educational experience for both students and teachers.

2.2 Product Functions

The OLMS offers several vital features to improve the online learning process. The main purposes are:

- Course Management

The System can let user browse from available courses catalog, students enrolling in classes of their choice , taking quizzes and tutors can upload assignments.

- *User Interaction*

Users can create or join groups for collaborative work and learning. Also ,Students can participate in discussions to share ideas and ask questions.

- Progress Tracking

Students can check on their performance and grades for both quizzes and assignments. In addition, students can check their deadlines for each course.

- Communication

An integrated messaging system allows users to interact with peers and teachers.

2.3 User Classes and Characteristics

- *Students*

The system is regularly used by students, to do everyday tasks and homework and , to look at available courses, sign up for classes, submit assignments in assignments, and take tests.

- *Instructor*

They can manage courses and upload assignments and keep track of students progress using OLMS .They interact with platform's features since they often use them .

- *Administrator*

Since they frequently use the system for reporting issues and they are essential for the system, they have the greatest degree of access and possess highly developed technical skills.

- *Gust users*

They access the system infrequently since they only might browse the system courses. Also, they have limited access and minimal technical knowledge and experience , they can explore the courses but not enroll in courses.

The most important user of these users are the students and instructor because they are the primary users , the less important users are gust users as they have minimal interaction with the system .

2.4 Operating Environment

The Online Learning Management System is designed as a user-friendly system that works in various hardware and software. For hardware components it can work on many different devices such as desktops, mobile phones and tablets. The operating systems that appropriate to the system are Windows 10 and later, macOS Mojave and later and Linux distributions (Ubuntu 18.04 and later). Also, uses database programs such as, MySQL AND MongoDB to store data and retrieve it or update it for all users .

2.5 Design and Implementation Constraints

There are various issues that developers might face regarding the OLMS such as, language requirements like using a specific programming language or regarding security requirements for implementing security measures such as encryption and user authentication protocols or even issues with the hardware like memory or proccing power .

2.6 User Documentation

First, there is the user manual which guides and explains every feature and functionality in the system also, FAQs which have the most asked questions and their answers to save time for users and answer their concerns.

2.7 Assumptions and Dependencies

Assumptions, first user adoption since we assume that the user is going to use the system . If they don't adopt it, the project success might be affected negatively . Second, internet access users need internet to use the system so if some users cannot connect to the internet, it could limit their ability to use the Online Learning Management System. Also , for dependencies we plan to use a third-party database management system such as MySQL to store data, if there are any issues with this app it might result in delay to our project .Another important factor is the funding availability , the project depends on the funding to improve and implement the features of the system , so if there is any cuts on the budget it will be a problem to continue and hire needed staff .

3. External Interface Requirements

3.1 User Interfaces

Define the logical characteristics of each interface between the software and users, including:

- **Screen Design**: Describe the screen layout, positioning of components, and overall design flow. This can include sample screen images or mockups.
- **Standard GUI Elements**: Specify any GUI standards or style guides, such as standard buttons (e.g., "Help", "Submit") and consistent features across screens.
- **Keyboard Shortcuts**: Define any shortcuts for ease of access and improved user efficiency.
- **Error Messages**: Establish standard formats and display methods for error messages, providing clear guidance for resolving issues.
- **Screen Layout Constraints**: Describe any required or recommended layout structures that should be followed to ensure a cohesive interface.

These details should provide a clear outline but will be elaborated on in a separate user interface specification document.

3.2 Hardware Interfaces

Define the logical and physical characteristics of each interface between the software and hardware:

- **Supported Device Types**: List devices (e.g., printers, scanners, IoT devices) that the software will interact with.
- **Data and Control Interactions**: Describe how data and control signals will be exchanged between the software and each hardware component.
- **Communication Protocols**: Specify the communication protocols (e.g., USB, Bluetooth, or custom protocols) that will be used to interact with the hardware.

3.3 Software Interfaces

Describe the interactions with other software components:

- **System Connections**: List specific software components the product interfaces with, including databases, operating systems, tools, and libraries.
- **Data Exchange**: Define incoming and outgoing data items or messages, along with their purposes.
- **APIs and Protocols**: Refer to application programming interface (API) protocols and standards for communication between components.
- **Shared Data**: Identify shared data across components, specifying any required implementation constraints (e.g., global data areas).

3.4 Communications Interfaces

Define requirements for any communication functions:

- **Types of Communication**: Include e-mail, web-based interactions, network server communications, and other relevant channels.
- **Message Formatting and Standards**: Specify message formatting and any required standards (e.g., FTP, HTTP).
- **Security Considerations**: Describe any encryption, authentication, or data protection standards needed.
- **Data Transfer**: Specify data transfer rates and any necessary synchronization methods.

4. System Features

4.1 Feature :Catalog of Courses and Search Feature

4.1.1 Description and Priority

Description: This feature will enable the user for purposes such as browsing, searching, and filtering of courses available in any category, like subject area, e.g., Math, Science, Humanities, difficulty level, or instructor. A user can locate a certain course by a search bar or by filtering options like course duration, prerequisites, certification, etc.

Priority: High

- Advantage: 9 - Users can find relevant courses quickly, hence increasing course enrollments and overall engagement on the platform.
- Penalty: 8 - Without this users will go elsewhere its too hard to find courses. Fewer courses will be enrolled in.
- Cost: 7 - This requires back-end structures to support hundreds of thousands of records and a front-end design that will make using the search function easy to use.
- Risk: 6 - Medium risk; unless optimized for performance, system performance will degrade with a large catalog of courses.

4.1.2 Stimulus/Response Sequences

- User Action: User selects a course category. For example, Science.
- System Response: Lists all Science courses available, course name, thumbnails, instructor information and course description.
- Unsuccessful Case: In case of no course being available under that category, it says, "No courses available in this category," suggesting related categories or popular courses.

- User Action: User types in a keyword in the search bar, example- "calculus."
- System Response: Lists courses related to "calculus" with course description and enrollment options.
- Unsuccessful Case: In the case of no matching courses for a keyword, the system shows "No courses found for 'calculus'" and suggests similar courses or keywords.
-
- User Action: User selects a filter, e.g., "Beginner level."
- System Response: The system now sifts through the course listing according to the level selected.
- Unsuccessful Case: The system displays, "No courses found for the selected filters," in an event when no course matches the filter, with options to reset or adjust filters.

- User Action: User clicks on a course for details.
- System Response: Open detailed course page - popping up extended information like syllabus, objectives, duration, and requirements.
- Unsuccessful Case: In case details are not available, pop an error message "Course details not available" and route the user to the main catalog.

4.1.3 Functional Requirements

- REQ-1: The system shall allow browsing of courses by category and keyword search.
- REQ-2: The system should permit filters for refining courses by difficulty, duration, and certification.
- REQ-3: Course information-such as syllabus, instructor, and enrollment-should be dynamically presented based on the user input.

4.2 Feature: Course Enrollment and Management

4.2.1 Description and Priority

This feature involves the enrollment of users in courses, managing enrollment in courses, and access to course materials. The user is allowed to add and remove courses in his dashboard, track his progress, and see his grades.

Priority High

- Advantge: 8- Simplifies how the user learns. The user enjoys the process and will be more satisfied with the platform.
- Penalty: 7- Courses are cumbersome for a user to engage in. Few users would complete courses, and their satisfaction with the platform diminishes.
- Cost: 6 - Backend for tracking enrollments and frontend of designing intuitive navigation
- Risk: 5 - Medium; would degrade in performance if many enrollments unless optimized.

4.2.2 Stimulus/Response Sequences

- User Action: A user enrolls in a course.
- System Response: Returns "You have successfully enrolled" and adds the course to the user's dashboard.
- Unsuccessful Case: If enrollment fails, say because the class is full, then the system displays "Unable to enroll" along with suggestions to join the waiting list or similar courses.

- User Action: The user goes to his/her dashboard.

- System Response: The system lists all enrolled courses with status flags such as "In Progress" or "Completed" and links to course materials.
- Unsuccessful Case: In case there is no course enrolled, it displays a message: "No courses enrolled. Browse courses to start learning!"

- User Action: The user requests removing a course from his dashboard.
- System Response: Confirms with "Course removed successfully," updating the dashboard.
- Unsuccessful Case: If the course is not removable-for instance, "locked by the instructor"-an error message occurs: "Unable to remove course. Please contact support."

- User Action: The user completes the course.
- System Response: Marks course as completed and generates a certificate, if any.
- Unsuccessful Case: In case of failure of completion due to system fault, a message appears: "Course completion error. Please retry."

4.2.3 Functional Requirements

- REQ-1: The system shall provide a feature of course enrollment by users and also show course enrollment status on the dashboard.
- REQ-2: The system shall be able to add and remove courses from the dashboard.
- REQ-3: The system should provide the user with the facility of tracking progress of courses and grades or completion status.
- REQ-4: If applicable, the system should provide a completion certificate.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system should respond to the user within 3 seconds under typical network conditions, to ensure that the learning process has minimal disruption.
- Data processing speed should be within 5 minutes to ensure that submission of assignments and grading have minimum processing time.
- The LMS must be able to scale horizontally to handle the increases in the user lode without requiring significant reconfiguration or downtime.

5.2 Safety Requirements

- The General Data Protection Regulation should be followed to ensure data protection and unauthorized access.
- Implement data pick up and recovery protocol to make sure that all data and the system are secure and able to be recovered.
- Ensure that personal data are only accessible to authorized personal.
- Implement clear warnings and confirmation messages before making any irreversible action.

5.3 Security Requirements

- **User Authentication and Authorization**:
  -Require secure login with multi-factor authentication (MFA) for all users to prevent unauthorized access.

-implement role-based access control to limit permissions based on user roles.

- **Data Protection and Encryption**:
  -store sensitive data securely with backup protocols.
  -use end to end encryption to protect data shared across the platform.
- **Incident response and monitoring:**

- implement continuous monitoring and logging of user activities to detect an authorized access and potential security, preaches.
-develop an incident response plan to deal with data breaches.

5.4 Software Quality Attributes

- **Usability**
  -The system must be easy to learn and navigate through with a clear interface and use a friendly design to support users with different technical skill level.
- **Scalability**
  -the most the system should be able to handle the increases in the number of users and courses without facing any problem.
- **Maintainability**
  -Conduct regular system audits and performance testing to identify areas to improve.

5.5 Business Rules

- Only administrators can change the user access and manage the system configuration.
- Instructors can create edit and remove their own courses content, and manage enrolled students access to course materials, and assignments.
- Instructors have control over visibility of their course content.
- Students can enroll in courses based on eligibility set by the institution or instructors.

6. Other Requirements

- **Database structure, and management:**

-scalable secure and high-level performance database to handle large volume of students and courses data.
-Implement database backup and recovery protocols to prevent data loss.

- **User training and support:**
  -Offer video, tutorials, and guides for the user.
  -provide a Helpdesk for any user questions about using the system.

Appendix A: Glossary

- Risk :This is the analysis of the issues that could arise regarding the implementation of the functionality. For example, performance risk due to the size of a large number of courses in the catalog.

- Priority: This is assigned to a feature depending on how much it is needed inside the LMS. The higher the priority, the more basic the feature is in terms of either system functionality or the satisfaction of users.

- Penalty: Is a result, usually negative, from not having a functionality implemented, such as lower engagement times on behalf of the user, or inability to navigate around the platform.

- Cost: Estimated effort to implement this feature in man-hours, including resources in both backend and frontend development.

- Error Message :A message that appears when something cannot be done, giving users additional information as to why it has failed, and what they can and should do next.

Appendix C: To Be Determined List

Priority	*task*	Status
1	*Security Measures*	*In Progress*
2	*Database Design*	*In Progress*
3	*managing enrollment in courses*	*In Progress*
4	*Testing Procedures*	*TBD*
5	*User Interface Guidelines*	*TBD*

First model business process

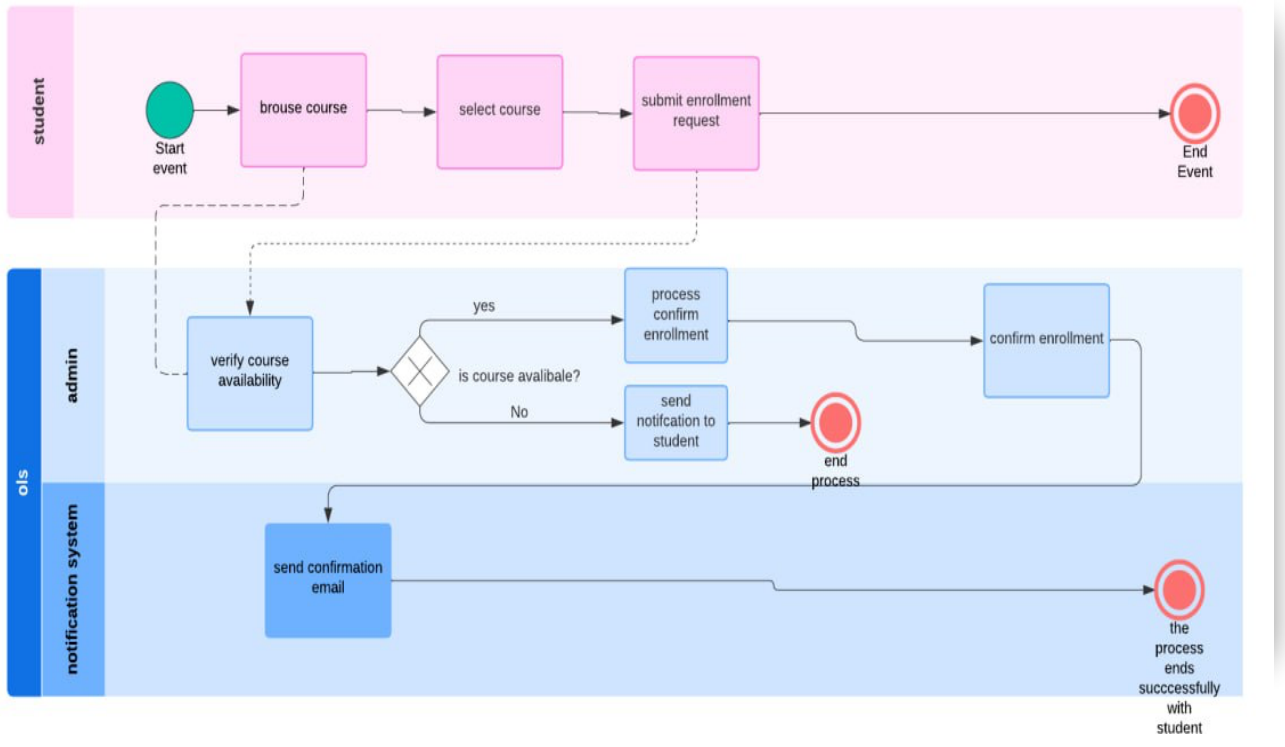
the Course Enrollment Process

That is how the scenario goes:

Students can peruse courses on an online learning platform and sign up for them according to their availability and areas of interest. They receive a confirmation email after their enrollment is completed and they have chosen a course.

Presumptions

To enroll, students need to be signed into their accounts.
Enrollment is limited to courses that have seats available.
An email notification and enrollment confirmation are sent automatically by the system.
Stores of Data:
Course Database: Contains links to browse courses and check for open seats.
Student Enrollment Information: Associated with Process Enrollment to document the status of student enrollment.
Message Flows: The Online Learning System Pool's Verify Course Availability job receives a message from the student's Submit Enrollment Request.



Second business process model

Course Completion and Certification Process

The scenario

To obtain a certification, a student must pass a final exam after finishing a course. A certificate is created by the system and emailed to the learner upon passing.

Presumptions

To obtain certification, a minimum score on the course exam must be achieved.
After passing, a certification is automatically generated.
The student receives an email notification from the system upon certification.Exam Results Database Data Store: Connected to Evaluate Exam Results to document scores.
Store granted certificates in the Certification Data Store, which is connected to the Generate Certificate.
Flows of Messages:
The Evaluate Exam Results task receives an exam data message from the student's Attempt Final Exam.

When the certificate is created, the Send Certificate Email notifies the Student Pool.