**Software Requirement Specification (SRS)**

**Name**: LMS Eklavya Foundation

**Team No.** : 14

**Members:**

Virat Garg

Varun Gupta

Prajas Wadekar

Vishnu Satvik

Vempati Keerti

## Version: 2nd

## Date: 14th Feb 2025

## **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Change Description** | **Reason for Change** |
| 1.0 | 14/02/2025 | Initial Version | Initial Draft |
| 2.0 | 2/03/2025 | Added usecases and user discussions | Feedback from client |
|  |  |  |  |

## **1. Brief Problem Statement**

This document outlines the software requirements for the Learning Management System (LMS) for Eklavya Foundation as part of our DASS course project. LMS aims at providing Higher Studies to all the students in form of different courses from various fields. It enables the teacher to create multiple courses and manage them. Add lectures, pdfs,assignments . The platform would be based on regular feedback mechanism from students, in form of comments and reviews. Different Organizations would be able to independently maintain their functioning and help children in various ways.

**2. System Requirements**

The LMS platform will be developed using the **MERN stack**. Cloudinary will be used for storing files. GitHub will serve as a collaborative tool for code tracking. Communication will be primarily through WhatsApp and biweekly meetings. We would use the server’s provided by our client for using MongoDB and Cloudinary to ensure security and privacy of data.

### **3. Users Profile**

The UI should be designed in such a manner to allow users from any level of software knowledge to easily deal with it.

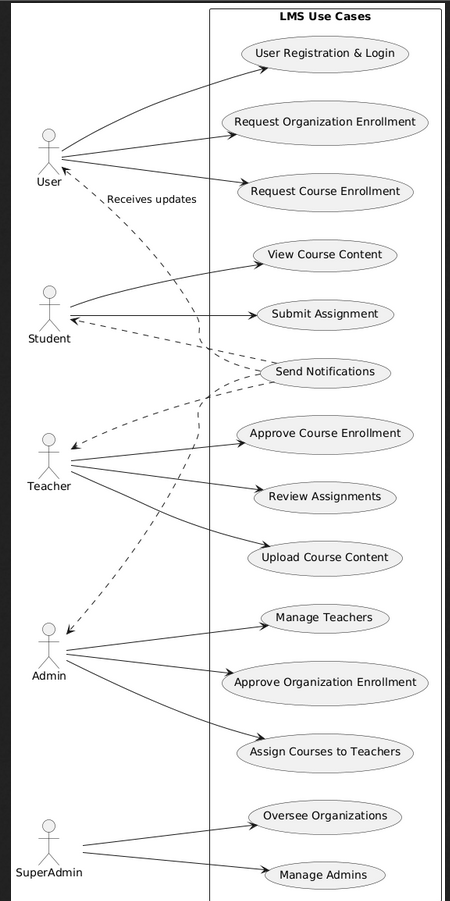
* **User**: Can request to join an organization, or can be added to the organization by the admin and will get the details through mail.
* **Student**: Can enroll in courses, access content, and submit assignments, provide feedback for the course.
* **Teacher**: Can add and manage course content, approve/reject student enrollment.
* **Admin**: Manages organization-related settings, approves user requests, and assigns teachers.
* **Super Admin**: Manages organizations and oversees activities across the platform.

## **3. System Requirements**

### **3.1 Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Use Case Name** | **Description** | **Release** |
| 1 | User Registration & Login | Users must sign up and log in to access the platform. | R1 |
| 2 | Organization ,Course Enrollment | Users request to join an organization ,course approved by the admin, teacher respectively | R1 |
| 3 | Content Management | Teachers upload course content (videos, PDFs, assignments). | R1 |
| 4 | Assignment Submission | Students submit assignments. | R1 |
| 5 | Teacher Management | Admins can edit add teachers, assign them courses. | R2 |
| 6 | Super Admin Controls | Super Admins manage organizations and user activities. | R3 |
| 7 | Teacher invites students. | Teachers invite students to join their courses | R2 |
| 8 | My Courses | Students can easily access the courses he is enrolled in | R1 |
| 9 | Discussions | Students can discuss under each section of their courses and give a feedback. | R1 |

## **4. Use Case Diagram**



## **5. Use Case Descriptions**

### **Use Case: UC-01**

**Use Case Number:** UC-01

**Use Case Name:** User Registration & Login

**Overview:** Allows users to register an account and log in securely.

**Actors:** User,Admin

**Precondition:** User must provide valid registration details.

**Main Flow:**

1. User enters registration details.

He either sign up’s as a new user or is provided by the credentials from the organisation through mail.

1. System validates and creates an account.
2. User logs in using credentials.
3. System verifies and grants access.
4. User can easily edit his profile including the password

**Alternate Flows:**

* Invalid credentials → Display error message.
* Forgotten password → Allow password reset.
* Email already registered->display already registered.

**Postcondition:** User is logged into the system, either as a student or as a user.

### **Use Case: UC-02**

**Use Case Number:** UC-02

**Use Case Name:** Organization, Course Enrollment

**Overview:** Users request to join an organization,C ourse subject to admin approval.

**Actors:** User, Admin ,Teacher

**Precondition:** User must be registered and logged in.

**Main Flow:**

1. User browses available organizations.
2. User submits a request to join an organization.
3. Admin reviews and approves/rejects the request.
4. Once a user has joined an organisation, he can request to join any course of the organization, which can be approved by the teacher as well as the admin.
5. If approved, the user is added to the organization,course and can access all it’s content.

**Alternate Flows:**

* Request rejected → User notified with reason.

**Postcondition:** User is either enrolled or notified of rejection.

### **Use Case: UC-03**

**Use Case Number:** UC-03

**Use Case Name:** Content management

**Overview:** Teacher uploads content in different formats(pdfs,video or assignments) and students can provide feedback.

**Actors:** Teacher, Student

**Precondition:** Teacher and the Student must be pre registerd.

Student is registered under a course taught by that teacher.

**Main Flow:**

1. Teacher uploads the content in a course under a subtopic.
2. Student registered in that course can view it and provide feedback if required.

3. Students can submit the assignments under the deadline decided by the teacher.

**Alternate Flows:**

* Wrong content uplooded->ability to edit the content.
* (If enough time available)Use of abusive words in comments->the comments are censored appropiately.

**Postcondition:** Student is warned regarding use of abusive language.

### **Use Case: UC-04**

**Use Case Number:** UC-04

**Use Case Name:** Assignment submission

**Overview:** Student submits the assignment assigned.

**Actors:** Teacher, Student

**Precondition:** Teacher and the Student must be pre registerd.

Student is registered under a course taught by that teacher.

**Main Flow:**

1. Teacher uploads the assignment in a course.
2. Student can submit the assignment under the deadline decided by the teacher.

3. Teacher can view the submission of each of the student and add comments for each one of them.

**Alternate Flows:**

* Wrong assignment uplooded->ability to edit the assignment
* Very less students submitted->Teacher can extend the deadline according to the course flexibility.

**Postcondition:** Student is notified about the condition and can now submit the assignment if he wants.

### **Use Case: UC-05**

**Use Case Number:** UC-05

**Use Case Name:** Teacher management

**Overview:** Admin has the power to add teachers to the LMS and can also assign them various courses.

**Actors:** Teacher, Admin

**Precondition:** Admin must be registered.

**Main Flow:**

1. Admin add teachers to LMS by adding their emails,teachers are sent a mail regarding this with their assigned passwrod for LMS.

2. Admin assigns course to the teacher.

3.Admin can edit the organisation’s details to make it more presentable.

**Alternate Flows:**

**Postcondition:** Teacher is assigned a particular course and can start to add content to that course.

### **Use Case: UC-06**

**Use Case Number:** UC-06

**Use Case Name:** Super Admin Controls

**Overview:** Super Admin can add various admins

**Actors:** Super Admins,Admins

**Precondition:**

**Main Flow:**

1. Super Admin adds Admin by adding his email and his account details will be mailed to him.
2. **Alternate Flows:**

**Postcondition:** Admin is registered and can add teachers as well as add the organisation’s

### **Use Case: UC-07**

**Use Case Number:** UC-07

**Use Case Name:** Teacher sends invite

**Overview:** Teacher can invite students to join there courses.

**Actors:** Teacher,Students

**Precondition:** Teacher must be logged in and have a course assigned

**Main Flow:**

1.Teacher accesses the list of students under the organisation he works.

2.He can send the invite to those students to join his course.

**Alternate Flows:**

**Postcondition:** Student receives an invite and has the choice to join the course.

### **Use Case: UC-08**

**Use Case Number:** UC-08

**Use Case Name:** My Courses

**Overview:** Student can view all of his courses under My Courses section.

**Actors:** Students

**Precondition:** Student is logged in

**Main Flow:**

1.All the courses in which the student is registered are showed in the MyCourses section.

2.Student can visit any of those courses and easily access there content.

**Alternate Flows:**

**Postcondition:** Student can access and complete the course on his own pace.

### **Use Case: UC-09**

**Use Case Number:** UC-09

**Use Case Name:** Discussions

**Overview:** Students can discuss under a section of any course.

**Actors:** Students.

**Precondition:** Student is logged in and enrolled in any courese

**Main Flow:**

1.Student goes to any of his enrolled courses shown under My Courses and go to any section of that course

2.He can view the public discussion regarding the section and participate in it.

**Alternate Flows:**

1. Student sends a hated comment in the discussion.
2. Teacher takes down that comment

**Postcondition:** Students can easily raise any queries about that section and take part in healthy discussions.