Software Requirements Specifications

for

Learning Management System

Seekho LMS

i200865 Aans Rehman Khan
i200767 Hassan Rasool
i200954 Huzefa Tanveer

Table of Contents

1. Introduction:

1.1. Purpose:

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. The primary focus of Seekho is to create a dynamic, robust e-learning platform that emphasizes automated video segmentation and dynamic quiz generation from segmented transcripts.

1.2. Scope:

"Seekho" is an e-learning platform aiming to revolutionize the method of content delivery and assessment. By harnessing the power of AI and NLP, Seekho auto-segments video content and generates quizzes dynamically. This will allow educators to deliver content seamlessly and assess learners effectively, providing a holistic e-learning environment. This initiative is in line with the evolving digital transformation in the education sector.

1.3. Definitions, Acronyms, and Abbreviations:

1.4. References:

Natural Language Processing Fundamentals. Smith, J., 2022. O'Reilly Publications.

Effective eLearning Content Delivery. Kumar, R., 2021. EduTech Press.

2. Research on existing products:

There are several LMS that assist students to understand clearly, some of them are as follows:

- EdApp
- Docebo
- Cornerstone OnDemand
- TalentLMS
- Degreed
- Learning Paths
- Absorb
- 360Learning
- Brainly
- Moodle
- Canvas
- Blackboard Learn

These LMSs offer a variety of AI features, such as:

- Personalized learning: AI can analyze each learner's progress and recommend relevant content and activities.
- Adaptive learning: AI can adjust the difficulty and pace of learning based on each learner's performance.
- Real-time feedback: AI can provide learners with immediate feedback on their work, helping them to identify and address any weaknesses.

- Automated tasks: AI can automate tasks such as grading assignments and generating reports, freeing up instructors to focus on teaching.

3. Project Vision:

3.1. Problem Statement:

Videos are one of the most effective modes of communicating ideas and concepts. Videos are an integral part modern society, and their applications range from advertisement and entertainment to education and social communication. Videos play an important role in education sector. Learning management systems utilize video lectures to make learning easy and effective. However, what modern learning management systems lack is a way to make video lectures engaging.

Most of the modern learning management systems such as Google Classroom do provide an option to upload video lectures. The problem is that these video lectures are of little to no use to students not because of the contents of the video lecture, but its time duration. The attention span of people especially youth is decreasing with the passage of time, and lengthy video lectures are often ignored as students often need to cover a particular concept discussed in a video lecture, not the whole lecture. Usually, students have to play the video in 2x to search through the video lecture the particular concept they want to understand. The problem intensifies during or near the exam days when students are in a hurry to cover concepts which are well covered in the video lectures but lost in the lengthy valuable video lectures. Moreover, even though teachers want their students to go through the video lectures and fully understand them. With LMS such as Google Classroom, it's not possible as it lacks any such feature. "Seekho" will help students and teachers overcome the issues mentioned above by allowing students to search through the video lectures and allowing teachers to track the progress and engagement of students with their video lectures.

3.2. Business Opportunities:

Video lectures are an integral part of modern learning management systems. Video lectures allow teachers to convey concepts clearly and effectively. Students on the other hand find it quite easy and interesting to grasp concepts via video lectures instead of navigating and going through lots of articles and web sites. A system which makes it easy for teachers to track students' progress and engagement with their video lectures, and which makes it easy for students to engage, search through, and fully utilize the video lecture will easily be adopted by teachers and students, and hence their educational institutes.

3.3. Objectives:

Here are some of the objectives that we want to achieve:

3.3.1. Enhance Video Lecture Engagement:

The primary objective of "Seekho" is to make video lectures more engaging for students. It aims to overcome the issue of lengthy video lectures that students often find challenging to navigate and understand fully.

3.3.2. Segmentation of Video Lectures:
"Seekho" will automatically segment video lectures into meaningful sections based on the content, making it easier for students to locate and understand specific concepts discussed in the lecture.

3.3.3. Search Functionality:
The system will enable students to search for specific concepts within video lectures using text, image, and voice search. This search feature will enhance accessibility and usability of video content.

3.3.4. Adaptive Learning:
"Seekho" aims to ensure that students have a clear understanding of concepts discussed in video lectures. It will provide adaptive learning by guiding students back to relevant segments of the video based on their quiz performance.

3.3.5. Efficient Quiz Management:
The system offers the ability for teachers to create quizzes efficiently. Smart Quiz Duplication will automatically generate quizzes to save time and prevent academic misconduct.

3.3.6. Progress Tracking:
"Seekho" will track student engagement and performance, providing both teachers and students with analytics on their progress and understanding of course material.

3.3.7. Universal Features:
The system will provide common features found in traditional learning management systems, such as class creation, announcements, assignment submissions, quiz taking, grading, and more.

3.3.8. User Roles:
"Seekho" will have role-based access, including teacher and student roles, with specific privileges and responsibilities for each.

3.3.9. Easy Course Management:
Teachers can create, manage, and organize courses, making it easy for students to access relevant content and interact with the system.

3.3.10. Improving Learning Outcomes:
The main objective is to enhance the learning outcomes of students by making educational video content more accessible and engaging.

3.3.11. Time Efficiency:
By automating tasks like quiz generation and content segmentation, "Seekho" aims to save time for both students and teachers.

3.3.12. Real-time Feedback:

Through quizzes and analytics, students will receive real-time feedback on their performance, allowing them to focus on areas where they need improvement.

3.3.13. Innovation in Education:

"Seekho" seeks to innovate in the education technology space by offering new ways for students to interact with video lectures and instructors to track and support student progress.

3.4. Project Scope:

1. Build a system that can segment the video. We will follow the Linguistics-Based Approach that will help us to segment the video based on the concepts. This will further help us in our major system that is learning management system to search and locate to specific concept in the lecture recording.

2. Our system will also generate quizzes that will assist teachers.

3. Our system will allow the students to attempt a quiz after attending the lecture so that the system can ensure that the student has the complete understanding of the concepts that are discussed in the video lecture. In case, if the student fails to attempt the question, the student will be redirected to that concept so that the student can understand that concept again and perform the quiz. This will help the student to have an understanding about the concept.

4. The system will keep an eye on all the activities of the student and record the engagement time to the system. This will allow teachers to evaluate the student based on the engagement time and marks that students get in the quizzes.

5. Build a Web Interface of a learning management system like Google Classroom's web interface but with the unique and innovative feature of allowing students to learn and understand the lectures more efficiently and increase engagement with the system.

6. Our system will also have a search bot that will allow the student to navigate to any concept that the student wants to search quickly. We will achieve this by text, image, and audio.

3.5. Constraints:

3.5.1. Usability:

It will capture the human aspects while using the interface, aesthetics, consistency in the user interface, online and context sensitive help, wizards and agents, user documentation, and training materials.

3.5.2. Responsiveness:

The performance imposes conditions on functional requirements. For-example, for a given action, it may specify performance parameters: transactions per second, response time, recovery time and resource usage such as (memory, disk, CPU…).

3.5.3. Capacity:

It will capture all the extensibility, adaptability, compatibility, configurability, serviceability, install-ability, localizability.

3.5.4. Portability:

There are applicable portability requirements to implement this functionality.

3.6. Stakeholders Description:

3.6.1. Stakeholders Summary:

There are many stakeholders of our project, that might include development team, quality assurance team, project management and much more. But the key stakeholders of our project are as follows:

Students: The primary users of the system, students, will access the "Seekho" to enhance their learning experience, engage with video lectures, improve their understanding of course material, and track their academic progress.

Teachers/Instructors: Teachers and instructors will utilize the system to create and manage courses, upload video lectures, quizzes, and assignments. Their goal is to improve the quality of education, track student progress, and ensure that students have access to effective learning materials.

Educational Institutions: These institutions, such as schools and universities, will adopt "Seekho" to provide a modern and engaging learning platform for their students. Their primary goal is to offer innovative and efficient education solutions.

Administrators: System administrators will manage user accounts, maintain system functionality, and ensure the security and reliability of the platform. They aim to provide uninterrupted services and maintain data privacy.

3.6.2. Key High-Level Goals and Problems of Stakeholders:

3.6.2.1. Students:

3.6.2.1.1. Goals:

- Enhance their understanding of course content.
- Improve engagement with video lectures.
- Efficiently locate specific concepts within video lectures.
- Track their academic progress and performance.

3.6.2.1.2. Problems:

- Difficulty navigating lengthy video lectures.
- Lack of engagement with course material.
- Need for efficient access to course content.

3.6.2.2. Teachers/Instructors:

3.6.2.2.1. Goals:

- Provide high-quality educational content.

- Track student progress and engagement.
- Reduce the time spent on quiz creation.
- Ensure students understand course concepts.

3.6.2.2.2. Problems:

- Time-consuming quiz creation.
- Limited ability to track student engagement.
- Challenges in ensuring student comprehension.

3.6.2.3. Educational Institutions:

3.6.2.3.1. Goals:

- Implement modern and innovative learning solutions.
- Improve student performance and satisfaction.
- Attract more students through advanced technology.

3.6.2.3.2. Problems:

- Need for engaging and efficient learning platforms.
- Competition for student enrollment.

3.6.2.4. Administrators:

3.6.2.4.1. Goals:

- Maintain system security and reliability.
- Manage user accounts and system functionality.

3.6.2.4.2. Problems:

- Data security and privacy concerns.
- System maintenance and uptime.

4. Software Requirement Specifications:

4.1. List of Features:

4.1.1. Course Management:

- Course creation, editing, and deletion.
- Course enrollment and management for students.
- Course materials upload (text, documents, videos, etc.).

4.1.2. Content Delivery:

- Lecture notes, presentations, and videos.
- Structured course modules and lessons.

4.1.3. Assignment and Assessment:

- Assignment submission and grading.

4.1.4. Progress Tracking:

- Tracking of course progress for students.
- Gradebook and performance analytics.
- Real-time progress tracking with detailed insights into achievements and areas for improvement.

4.1.5. Notifications and Announcements:

- System-generated notifications and announcements from instructors.

4.1.6. Search Functionality:

- Search for courses, materials, and other resources.
- Search for specific keywords or concepts within video lectures.

4.1.7. Video Player:

- Integrated video player for viewing course-related video content within the platform.

4.1.8. Automated Video Segmentation through NLP:

- Utilizes Natural Language Processing (NLP) to automatically segment video content into meaningful sections.

4.1.9. Dynamic Quiz Generation:

- AI algorithms generate quizzes.

4.2. Functional Requirements:

4.2.1. Course Management:

4.2.1.1. Course Creation:

4.2.1.1.1. Description:

Instructors can create, edit, and delete courses.

4.2.1.1.2. Priority:

High

4.2.1.1.3. Stimulus/Response Sequences:

4.2.1.1.3.1. Instructor will first log into the system.
4.2.1.1.3.2. Instructors access the course management interface.
4.2.1.1.3.3. Instructors create a course by clicking the add (+) button.
4.2.1.1.3.4. The new course form will be opened.
4.2.1.1.3.5. Instructors will enter the inputs as required.
4.2.1.1.3.6. After completing the form, the instructor will click on the done button. The new course will be created and returned to the course management interface.

4.2.1.1.4. Requirements:

4.2.1.1.4.1. The system should be connected to the database.
4.2.1.1.4.2. The system should be able to add the course to the database.
4.2.1.1.4.3. The system must provide functionality for course creation.

4.2.1.2. Course Editing:

4.2.1.2.1. Description:

Instructors can edit courses.

4.2.1.2.2. Priority:

High

4.2.1.2.3. Stimulus/Response Sequences:

4.2.1.2.3.1. Instructor will first log into the system.
4.2.1.2.3.2. Instructors access the course management interface.
4.2.1.2.3.3. Instructor will open the course by clicking on the specific course.
4.2.1.2.3.4. Instructors edit a course by clicking the edit button.
4.2.1.2.3.5. The editing form will be opened.
4.2.1.2.3.6. Instructors will make modifications as per their choice.
4.2.1.2.3.7. After making all the modifications, the instructor will click on the done button. The course will be edited.

4.2.1.2.4. Requirements:

4.2.1.2.4.1. The system should be connected to the database.
4.2.1.2.4.2. The system should be able to edit the course in the database.
4.2.1.2.4.3. The system must provide functionality for course creation, editing, and deletion.

4.2.1.3. Course Deletion:

4.2.1.3.1. Description:

Instructors can delete a course.

4.2.1.3.2. Priority:

High

4.2.1.3.3. Stimulus/Response Sequences:

4.2.1.3.3.1. Instructor will first log into the system.
4.2.1.3.3.2. Instructors access the course management interface.
4.2.1.3.3.3. Instructor will open the course by clicking on the specific course.
4.2.1.3.3.4. Instructors can delete a course by clicking the delete button.
4.2.1.3.3.5. The course will be deleted and returned to the course management interface.

4.2.1.3.4. Requirements:

4.2.1.3.4.1. The system should be connected to the database.
4.2.1.3.4.2. The system should be able to add the course to the database.
4.2.1.3.4.3. The system must provide functionality for course creation, editing, and deletion.

4.2.1.4. Course Management for Students:

4.2.1.4.1. Description:

Students can enroll in courses, view their enrolled in the courses and manage their courses.

4.2.1.4.2. Priority:
High

4.2.1.4.3. Stimulus/Response Sequences:
4.2.1.4.3.1. Students log in to their accounts.
4.2.1.4.3.2. Students will click the add button in the course management interface.
4.2.1.4.3.3. The new form will appear.
4.2.1.4.3.4. Student will enter the course code that the instructor has given to the students.
4.2.1.4.3.5. Student will click the add button.
4.2.1.4.3.6. The new course will be added.
4.2.1.4.3.7. The system will return to the main course management interface.

4.2.1.4.4. Requirements:
4.2.1.4.4.1. The system should be able to communicate with the database.
4.2.1.4.4.2. The system should allow the student to enroll in the new courses.
4.2.1.4.4.3. The system should check for the valid course code.
4.2.1.4.4.4. Students should be able to view and manage their enrolled courses.

4.2.1.5. Course Materials Upload (Text, Documents, Videos, etc.):
4.2.1.5.1. Description:
Instructors can upload various types of course materials, including text, documents, and videos.

4.2.1.5.2. Priority:
High

4.2.1.5.3. Stimulus/Response Sequences:
4.2.1.5.3.1. Instructors access the course materials management section.
4.2.1.5.3.2. Instructors will click on the button named as add new material.
4.2.1.5.3.3. The form will open instructor will select which type of file he/she wants to add.
4.2.1.5.3.4. The instructor will attach the file and click the button “Add”.
4.2.1.5.3.5. The new material will be added to the database and will be displayed in the course.

4.2.1.5.4. Requirements:
4.2.1.5.4.1. The system should be able to add new material to the database.
4.2.1.5.4.2. The system should support the upload of different types of course materials.

4.2.1.5.4.3. Instructors must be able to manage and organize course content.

4.2.2. Content Delivery:

4.2.2.1. Lecture Notes, Presentations, and Videos:

4.2.2.1.1. Description:

Students can access lecture notes, presentations, and video content within the platform.

4.2.2.1.2. Priority:

High

4.2.2.1.3. Stimulus/Response Sequences:

4.2.2.1.3.1. Students will log in to the system.
4.2.2.1.3.2. Students navigate to their course materials.
4.2.2.1.3.3. There will be portions of presentations and videos. Students can swap from on to another.
4.2.2.1.3.4. Students access lecture notes, presentations, and videos.

4.2.2.1.4. Requirements:

4.2.2.1.4.1. The system should be able to access the database.
4.2.2.1.4.2. The system should allow students to access various types of course content, including lecture notes, presentations, and videos.

4.2.2.2. Structured Course Modules and Lessons:

4.2.2.2.1. Description:

Courses are organized into structured modules and lessons for effective learning.

4.2.2.2.2. Priority:

High

4.2.2.2.3. Stimulus/Response Sequences:

4.2.2.2.3.1. The user will log in to the system.
4.2.2.2.3.2. Instructors create course modules and lessons.
4.2.2.2.3.3. Students navigate through structured course content.

4.2.2.2.4. Requirements:

4.2.2.2.4.1. The system must support the organization of courses into modules and lessons.
4.2.2.2.4.2. Students should be able to easily navigate through structured course content.

4.2.3. Assignment and Assessment:

4.2.3.1. Assignment Submission and Grading:

4.2.3.1.1. Description:

Students can submit assignments, and instructors can evaluate and grade them.

4.2.3.1.2. Priority:
High

4.2.3.1.3. Stimulus/Response Sequences:
4.2.3.1.3.1. Instructors create assignments and set deadlines.
4.2.3.1.3.2. Students submit assignments.
4.2.3.1.3.3. Instructors review and grade assignments.

4.2.3.1.4. Requirements:
4.2.3.1.4.1. The system should accept zip files, form, images and videos.
4.2.3.1.4.2. The system should support assignment creation, submission, evaluation, and grading.

4.2.3.2. Assessment Tools (Quizzes, Tests, Assignments):
4.2.3.2.1. Description:
Instructors can create quizzes, tests, and assignments as assessment tools.

4.2.3.2.2. Priority:
High

4.2.3.2.3. Stimulus/Response Sequences:
4.2.3.2.3.1. Instructors will log in to the system.
4.2.3.2.3.2. Instructors will select the specific course in which they want to create a quiz, test, or assessment.
4.2.3.2.3.3. Instructors will click on the create quiz button.
4.2.3.2.3.4. Instructors create quizzes, tests, or assignments.
4.2.3.2.3.5. Students access and complete assessments.

4.2.3.2.4. Requirements:
4.2.3.2.4.1. The system should provide tools for instructors to create quizzes, tests, and assignments.
4.2.3.2.4.2. Students should be able to access and complete assessments.

4.2.4. Progress Tracking:
4.2.4.1. Tracking of Course Progress for Students:
4.2.4.1.1. Description:
Students can track their progress within each course.

4.2.4.1.2. Priority:
High

4.2.4.1.3. Stimulus/Response Sequences:
4.2.4.1.3.1. Students will log in to the system.
4.2.4.1.3.2. Students will select the specific course.
4.2.4.1.3.3. Students access their course progress reports.
4.2.4.1.3.4. The system updates progress data based on user activities.

4.2.4.1.4. Requirements:
4.2.4.1.5. The system should track course progress of students.

4.2.4.1.6. The system should show the course progress to the students.
4.2.4.1.7. The system should monitor the total time the student is active on the system.

4.2.4.2. Gradebook and Performance Analytics:

4.2.4.2.1. Description:

Instructors can maintain gradebooks, and both instructors and students can access performance analytics.

4.2.4.2.2. Priority:

High

4.2.4.2.3. Stimulus/Response Sequences:

4.2.4.2.3.1. Instructors input grades and maintain gradebooks.
4.2.4.2.3.2. Students and instructors access performance analytics.

4.2.4.2.4. Requirements:

4.2.4.2.4.1. The system should support gradebook management by instructors.
4.2.4.2.4.2. Both instructors and students should have access to performance analytics.

4.2.4.3. Real-Time Progress Tracking with Detailed Insights into Achievements and Areas for Improvement:

4.2.4.3.1. Description:

Real-time tracking of course progress with insights on achievements and areas for improvement.

4.2.4.3.2. Priority:

High

4.2.4.3.3. Stimulus/Response Sequences:

4.2.4.3.3.1. Users access real-time progress tracking.
4.2.4.3.3.2. The system provides detailed insights based on user activities.

4.2.4.3.4. Requirements:

4.2.4.3.4.1. The system must offer real-time course progress tracking.
4.2.4.3.4.2. It should provide detailed insights into user achievements and areas for improvement.

4.2.5. Notifications and Announcements:

4.2.5.1. System-Generated Notifications and Announcements from Instructors:

4.2.5.1.1. Description:

Users receive system-generated notifications and announcements from instructors.

4.2.5.1.2. Priority:

High

4.2.5.1.3. Stimulus/Response Sequences:

4.2.5.1.3.1. Instructors will upload any new material to the course.
4.2.5.1.3.2. The system will
4.2.5.1.3.3. Users receive notifications.

4.2.5.1.4. Requirements:

4.2.5.1.4.1. The system should support system-generated notifications and announcements for communication.

4.2.6. Search Functionality:

4.2.6.1. Search for Courses, Materials, and Other Resources:

4.2.6.1.1. Description:

Users can search for courses, materials, and other resources within the platform.

4.2.6.1.2. Priority:

High

4.2.6.1.3. Stimulus/Response Sequences:

4.2.6.1.3.1. Users enter search queries.
4.2.6.1.3.2. The system displays search results.

4.2.6.1.4. Requirements:

4.2.6.1.4.1. The system must provide search functionality for courses and materials.

4.2.6.2. Search for Specific Keywords or Concepts within Video Lectures:

4.2.6.2.1. Description:

Users can search for specific keywords or concepts within video lectures.

4.2.6.2.2. Priority:

High

4.2.6.2.3. Stimulus/Response Sequences:

4.2.6.2.3.1. Users enter search keywords.
4.2.6.2.3.2. The system searches for keywords within video content.

4.2.6.2.4. Requirements:

4.2.6.2.4.1. The system shall be able to search for the specific concept by matching the words, phrases and frequency of words used.
4.2.6.2.4.2. The system should offer search functionality within video lectures.

4.2.7. Video Player:

4.2.7.1. Integrated Video Player for Viewing Course-Related Video Content within the Platform:

4.2.7.1.1. Description:

Users can view course-related video content through an integrated video player.

4.2.7.1.2. Priority:

High

4.2.7.1.3. Stimulus/Response Sequences:

4.2.7.1.3.1. The user will first log into the system.
4.2.7.1.3.2. The user will select the course he/she wants to study of.
4.2.7.1.3.3. The user will select the lecture he/she wants to study.
4.2.7.1.3.4. Users access video content.
4.2.7.1.3.5. The system plays videos within the platform.

4.2.7.1.4. Requirements:

4.2.7.1.4.1. The system should have access to the videos that will run in the video player.
4.2.7.1.4.2. The system should include an integrated video player for viewing video content.

4.2.8. Automated Video Segmentation through NLP:

4.2.8.1. Utilizes Natural Language Processing (NLP) to Automatically Segment Video Content into Meaningful Sections:

4.2.8.1.1. Description:

The system uses NLP to automatically divide video content into relevant sections.

4.2.8.1.2. Priority:

High

4.2.8.1.3. Stimulus/Response Sequences:

4.2.8.1.3.1. The system will access video content.
4.2.8.1.3.2. The system will generate the transcript of the video.
4.2.8.1.3.3. The NLP algorithm will generate the segments in the transcript.
4.2.8.1.3.4. The system will match the segments with the timestamps in the video.

4.2.8.1.4. Requirements:

4.2.8.1.4.1. The system should generate the transcript of the video.
4.2.8.1.4.2. The system should employ NLP for automated video segmentation.

4.2.9. Dynamic Quiz Generation:

4.2.9.1. AI Algorithms Generate Quizzes:

4.2.9.1.1. Description:
AI algorithms are used to generate quizzes.

4.2.9.1.2. Priority:
High

4.2.9.1.3. Stimulus/Response Sequences:
4.2.9.1.3.1. Instructors create quiz templates.
4.2.9.1.3.2. AI algorithms generate quizzes based on templates.

4.2.9.1.4. Requirements:
4.2.9.1.4.1. The system should utilize AI algorithms to generate quizzes based on predefined templates.

4.3. Non-Functional Requirements:

4.3.1. Performance Requirements

Load Time: The web application should load within 2 seconds under standard broadband conditions.

Video Stream: Videos should stream with a maximum buffer time of 3 seconds for users with at least a 5 Mbps connection.

Response Time: For user interactions such as enrolling in a course or taking a quiz, the system should respond within 1.5 seconds.

Concurrency: The system should handle a minimum of 10,000 concurrent users without degradation in performance.

Real-time Processing: Automated video segmentation through NLP should process within 5 minutes for an hour-long video.

4.3.2. Safety Requirements

Data Backup: Regular backups of user data, course content, and quiz results should be maintained to prevent data loss.

Operational Safety: Ensure safeguards against potential hazards of prolonged screen time, such as reminders for users to take breaks during long courses.

Disaster Recovery: A disaster recovery plan should be in place to ensure system recovery within 24 hours in case of any catastrophic events.

4.3.3. Security Requirements

Data Encryption: User data, especially personal information, should be encrypted both in transit and at rest.

Authentication: Multi-factor authentication should be available and encouraged for educators and administrators.

Role-Based Access Control (RBAC): Ensure only authorized users can access specific features based on their roles (e.g., students can't delete courses).

Privacy Compliance: Ensure GDPR, CCPA, and other relevant data protection compliance for storing and processing user data.

5. Iteration Plan:

Iteration 1: Project Initiation and Core Features

Duration: 4 weeks

Objectives:

- Project kickoff and team formation.
- Define the project's scope, goals, and constraints.
- Create a detailed project plan, including roles and responsibilities.
- Develop the core features and architecture of the Learning Management System (LMS).
- Set up the basic user roles and access control (e.g., student, teacher, admin).

Tasks:

- Define and document the core features in more detail.
- Develop a basic web interface for the LMS.
- Implement user authentication and role management.
- Create a simple database schema to store essential data.
- Set up the development environment and version control system.

Iteration 2: Enhanced Functionality and User Roles

Duration: 6 weeks

Objectives:

- Enhance LMS functionality.
- Implement adaptive learning and quiz duplication features.
- Refine the user roles and their permissions.
- Introduce video lecture segmentation and searching.

Tasks:

- Develop the adaptive learning feature, allowing quizzes based on video content.
- Implement video lecture segmentation and searching.
- Enhance the user interfaces for teachers and students.
- Improve the database schema to store more detailed information.
- Conduct user testing for core features.

Iteration 3: Advanced Features and User Engagement

Duration: 8 weeks

Objectives:

- Implement advanced features to improve user engagement.
- Develop the progress and engagement tracker.
- Enhance the search functionality.
- Perform rigorous testing and debugging.

Tasks:

- Develop the progress and engagement tracker, including analytics.
- Enhance the search functionality to support text, image, and voice searching.
- Conduct comprehensive testing, including load testing and security testing.
- Gather user feedback and make necessary refinements.
- Prepare documentation for the system.

Iteration 4: Finalization, Deployment, and Maintenance

Duration: 4 weeks

Objectives:

- Finalize the system and address any remaining issues.
- Prepare for deployment.
- Develop maintenance and support plans.

Tasks:

- Address any remaining bugs or issues.
- Optimize system performance and security.
- Prepare the system for deployment on the target server.
- Develop maintenance and support plans for ongoing system updates and user support.
- Conduct training for users and administrators as necessary.
- Finalize project documentation.

6. Implementation Details:

We will develop our web application using MERN stack and for the AI related features such as video segmentation, and quiz generation we will use python.

Some implementation details of our major features are as follows:

6.1. Detailed Searching in a video:

Preprocessing:

Preprocess the video transcript by tokenizing the text, removing stop words, punctuation, and special characters. You may also want to perform stemming or lemmatization to reduce words to their root forms.

Concept Extraction:

Identify the specific concepts or keywords you want to search for in the transcript. These concepts can be entered as prompts by the user or predefined.

Vectorization:

Convert the preprocessed text data into numerical vectors. There are various vectorization techniques you can use, such as:

TF-IDF (Term Frequency-Inverse Document Frequency): Assign numerical weights to terms based on their frequency and importance within the transcript.

Word Embeddings (Word2Vec, GloVe, FastText): Map words to dense vector representations that capture semantic relationships. You can average word vectors to represent entire documents.

Document Embeddings (Doc2Vec): Generate embeddings for the entire transcript or document.

Similarity Measurement:

Calculate the similarity between the concept vector and each segment of the video transcript.

Common similarity metrics include:

Cosine Similarity: Measures the cosine of the angle between two vectors. It's commonly used for text similarity.

Euclidean Distance: Measures the straight-line distance between vectors.

Jaccard Similarity: Measures the intersection over union of sets for binary data.

Ranking and Filtering:

Rank the segments based on their similarity scores with the concept vector.

Set a threshold or use a ranking algorithm to filter and return the most relevant segments.

6.2. Quiz Generation:

Deep learning models, especially sequence-to-sequence models like LSTM or transformers (e.g., GPT-3, T5), can be trained to generate questions from passages. - You frame the task as a sequence-to-sequence problem, where the input is the passage, and the output is the MCQ question. - The model learns to generate question text and answer choices simultaneously.

6.3. Video Segmentation:

Video to audio conversion:

FFmpeg or moviepy

Voice Activity Detection (VAD):

Apply Voice Activity Detection (VAD) algorithms to identify segments of audio that contain speech. VAD can help isolate the speech portions from silence and background noise.

Transcription with ASR:

Once you have segmented the audio into speech segments, you can use ASR models to transcribe the speech into text. This step is similar to traditional ASR but is applied to shorter, segmented audio clips.

Google Cloud Speech-to-Text

Amazon Transcribe

Microsoft Azure Speech Service

Open-source options like Mozilla DeepSpeech or Kaldi

language models such as bert is also used for language modeling

Text segmentation:

Audio Segmentation:

Silence Detection: Audio segments can often be separated by periods of silence. You can use audio processing libraries and techniques to detect long periods of low audio amplitude (silence) as potential segment boundaries.

Audio Energy: Measuring audio energy levels can help identify significant changes in content. Sudden increases or decreases in audio energy can be indicative of segment boundaries.

Speech-to-Text Alignment: If you have transcriptions of the audio, you can align the text with the audio and identify breaks based on changes in spoken content.

Change-Point Detection Algorithms: Change-point detection algorithms like the Bayesian change point detection or CUSUM (cumulative sum) algorithm can be used to identify abrupt changes in time series data, such as audio energy levels or visual feature vectors.

Sentiment Analysis: Perform sentiment analysis on the text to identify changes in sentiment or emotional tone. Significant shifts in sentiment may indicate the need for segmentation.

Named Entity Recognition (NER): Use NER models to identify and extract named entities (e.g., names, dates, locations) within the text. Named entities can represent segments of interest, such as people, events, or places.

Topic Modeling:

To find topics within the text segments, you can employ various topic modeling algorithms and models, such as:

- **Latent Dirichlet Allocation (LDA)**
- **Non-Negative Matrix Factorization (NMF)**
- **BERT-based models fine-tuned for text classification.**
- **Hierarchical Dirichlet Process (HDP)**

- **Latent Semantic Analysis (LSA)**

LDA:

Latent Dirichlet Allocation (LDA) is a probabilistic generative model used in natural language processing (NLP) and text mining. It is primarily used for topic modeling, a technique for discovering hidden topics or themes within a collection of documents. Here's an explanation of LDA, its use cases, strengths, and weaknesses:

How LDA Works:

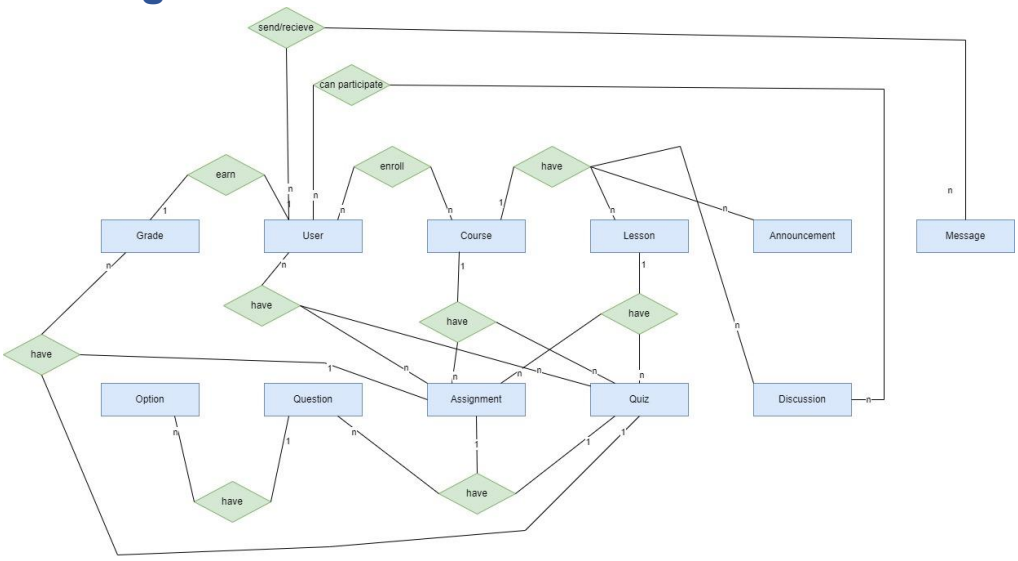
LDA assumes that documents are mixtures of topics, and each topic is a mixture of words. It operates under the following key assumptions:

1-Each document is a mixture of a few topics.

2-Each topic is a mixture of words.

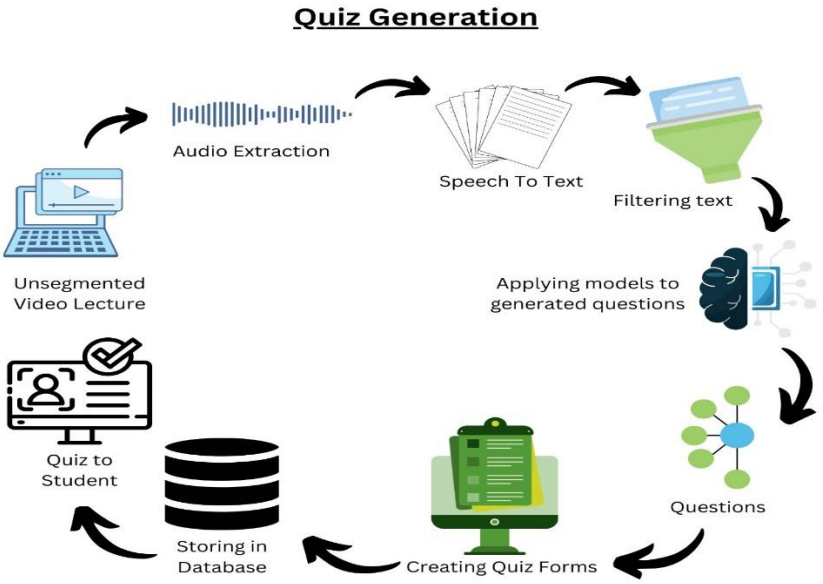
LDA aims to reverse-engineer this process to discover the topics in a collection of documents. It does this through a process of statistical inference, trying to find the hidden topics that are most likely to have generated the observed documents.

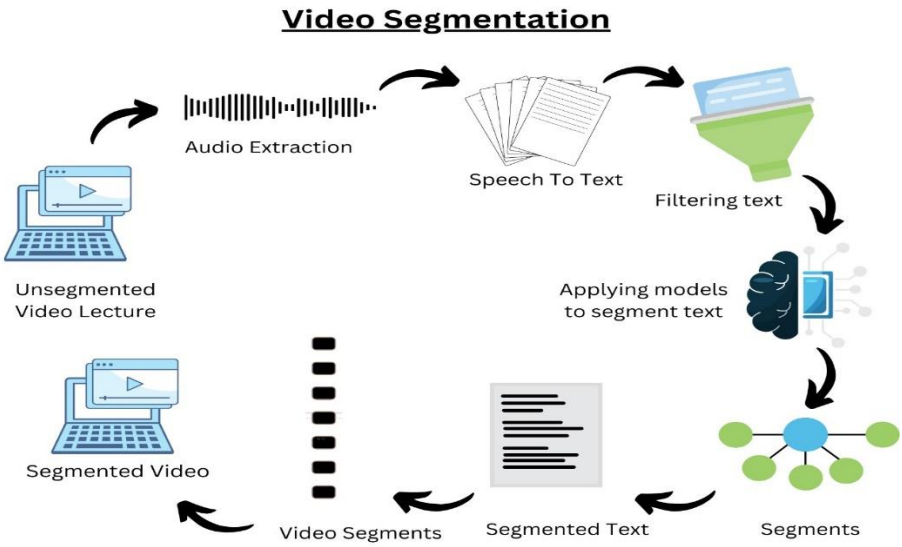
ER Diagram:



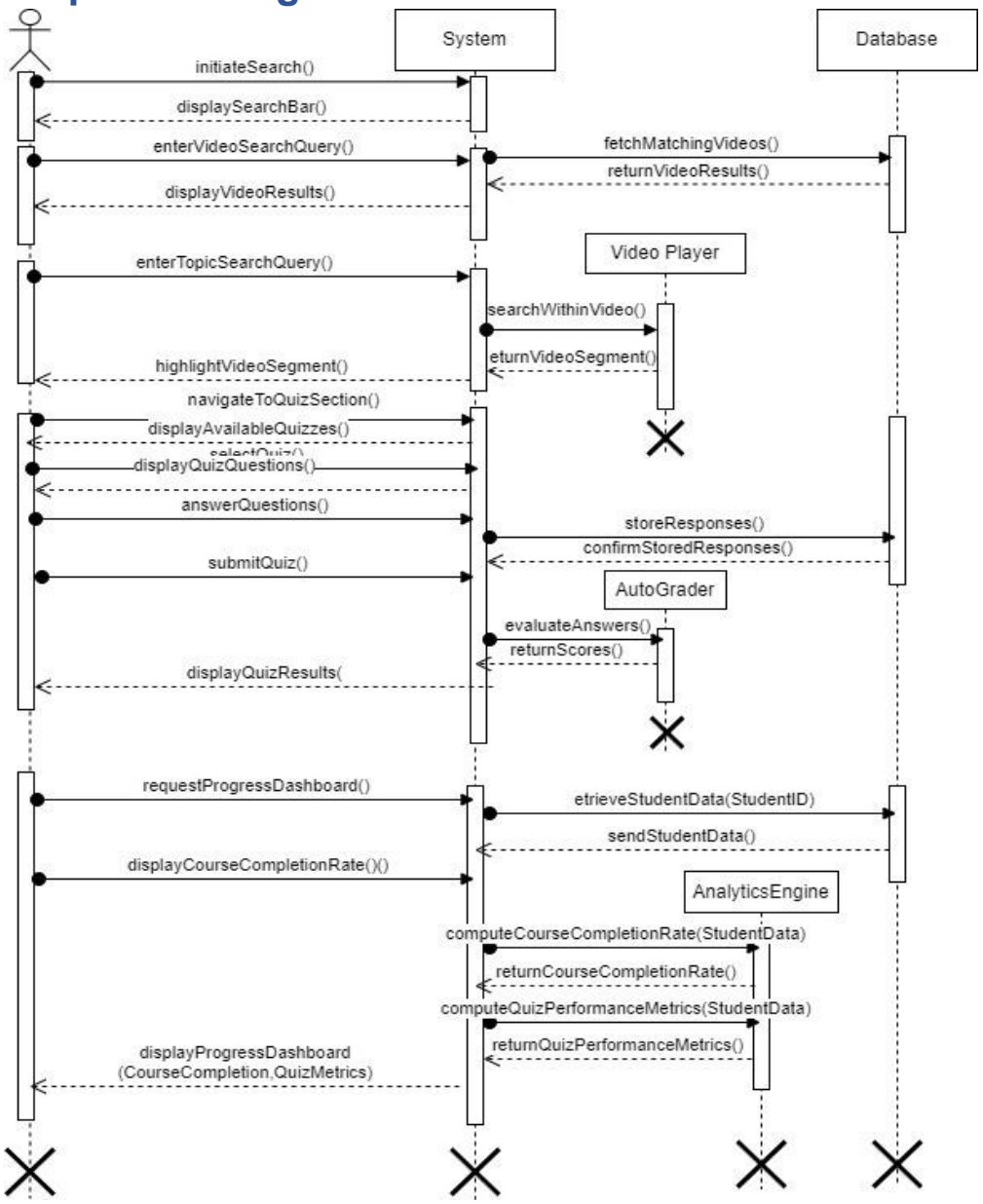
We have also made a JSON file for clarifying the attributes.

Architecture Diagram:

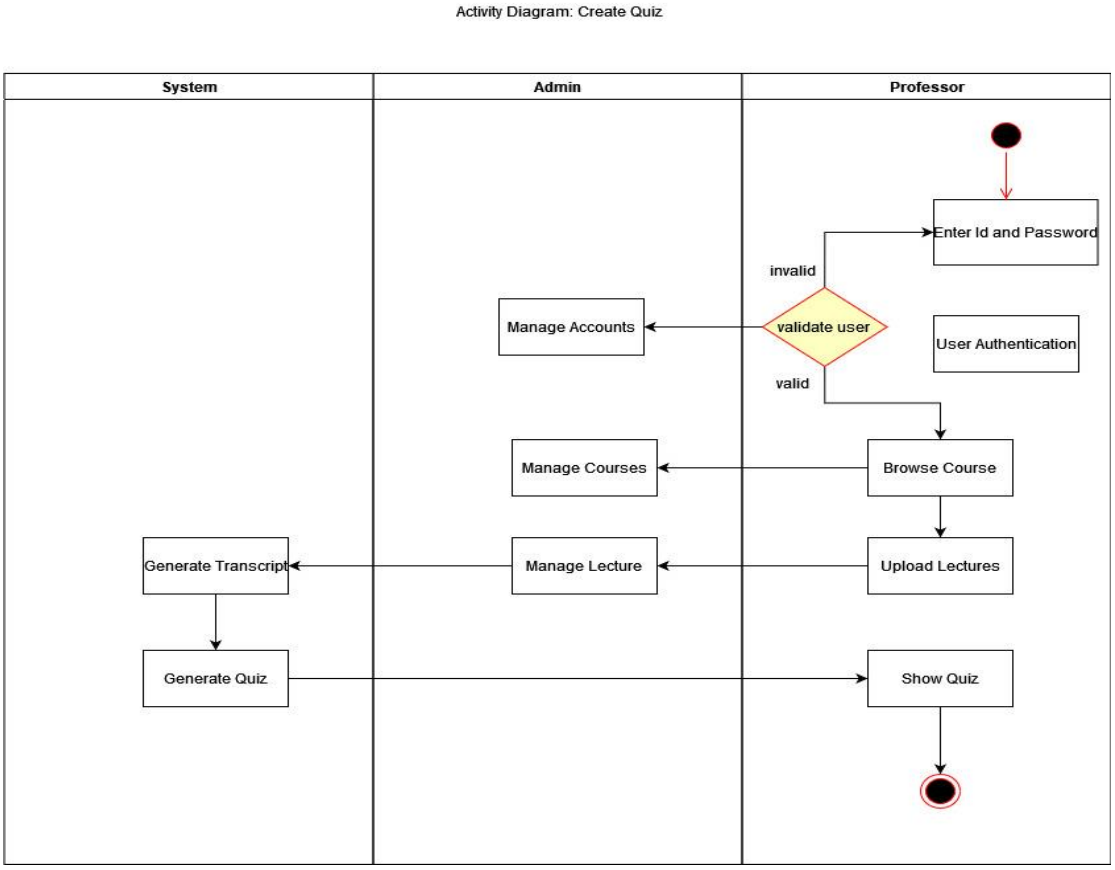




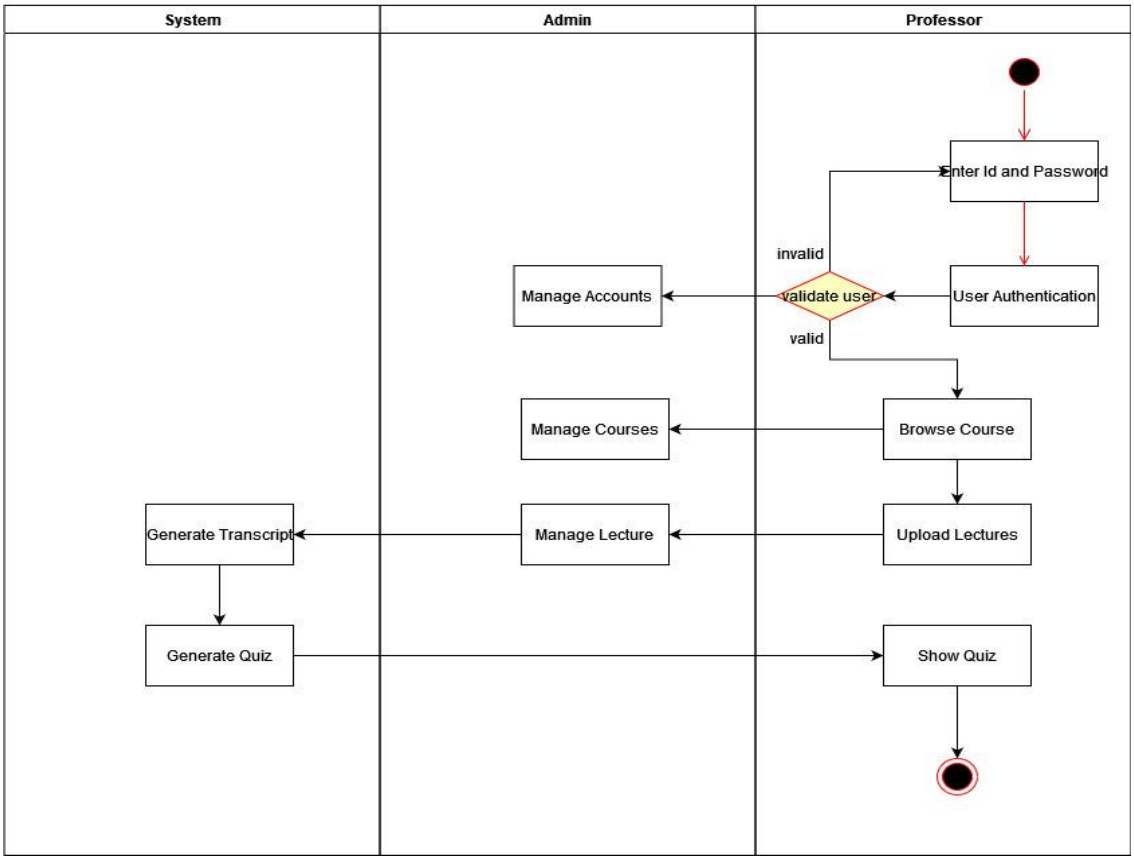
Sequence Diagram:



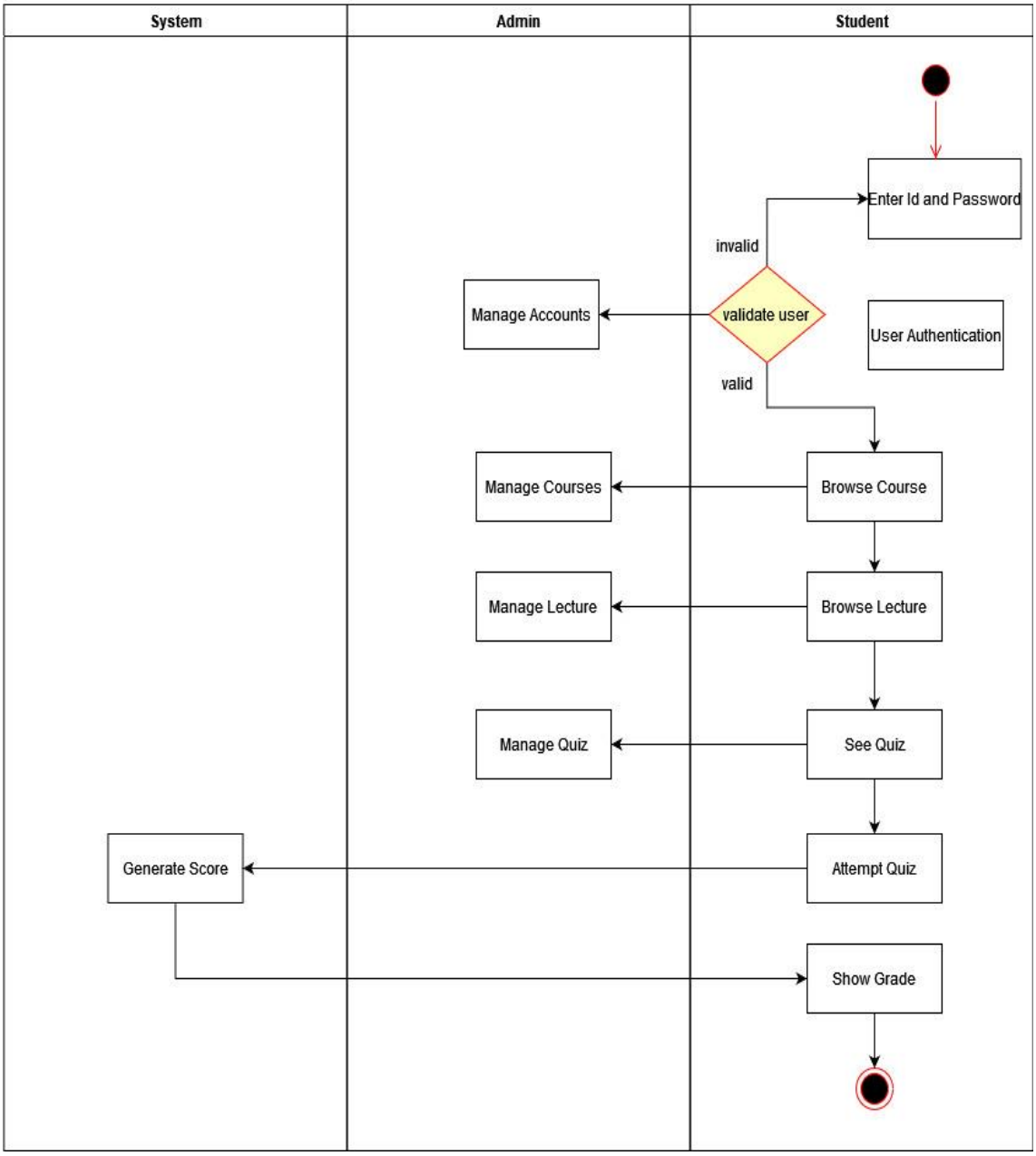
Activity Diagrams:



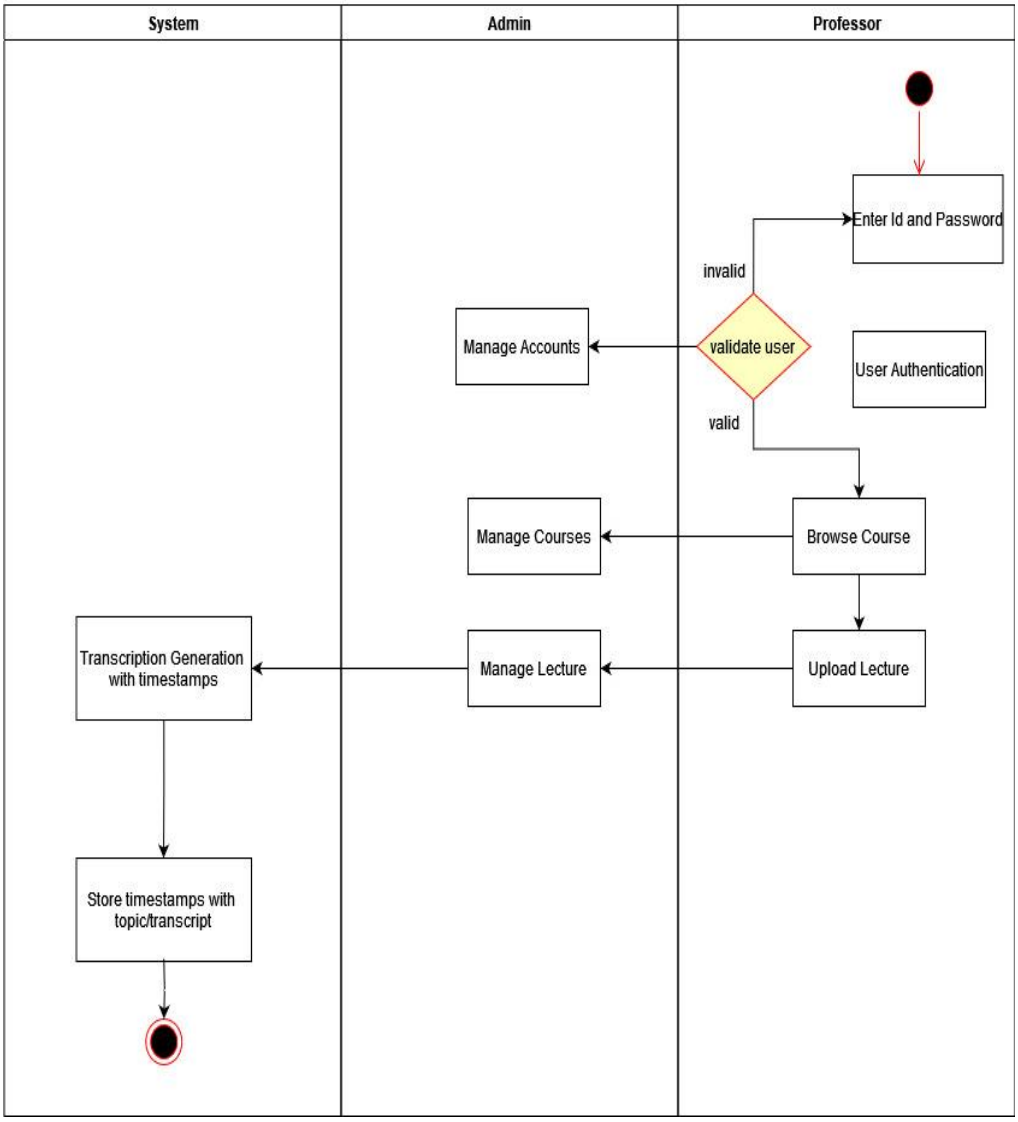
Activity Diagram: Create Quiz



Activity Diagram:Attempt Quiz and Grade



Activity Diagram: Video Segmentation



Use case Diagram