**PLATFORM FOR IMPROVING SEARCHABILITY AND INTERACTIVITY OF RECORDED LECTURES**

Project ID: 19-087

Software Requirement Specification

**Fernando S. S. M. S.**

**IT16001862**

Bachelor of Science (Honors) in Information Technology

Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2019

**PLATFORM FOR IMPROVING SEARCHABILITY AND INTERACTIVITY OF RECORDED LECTURES**

Project ID: 19-087

Software Requirement Specification

40 mm

20 mm

20 mm

25 mm

Bachelor of Science (Honors) in Information Technology Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2019

# **DECLARATION**

I declare that this is our own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

------------------------------------------------

S. S. M. S. Fernando

The above candidates are carrying out research for the undergraduate Dissertation under my supervision

Supervisor’s Name: Mr. Nuwan Kodagoda Signature: -------------------------------

Co-Supervisor’s Name: Ms. Kushnara Suriyawansa Signature: -------------------------------

**TABLE OF CONTENTS**

[**DECLARATION** 2](#_Toc8678361)

[**1.** **INTRODUCTION** 5](#_Toc8678362)

[**1.1** **Purpose** 5](#_Toc8678363)

[**1.2** **Scope** 5](#_Toc8678364)

[**1.3 Acronyms and Abbreviations** 7](#_Toc8678365)

[**1.4 Overview** 7](#_Toc8678366)

[**2.** **OVERALL DESCRIPTIONS** 8](#_Toc8678367)

[**2.1** **Product Perspective** 10](#_Toc8678368)

[**2.1.1** **System Interfaces** 13](#_Toc8678369)

[**2.1.2** **User Interfaces** 13](#_Toc8678370)

[**2.1.3** **Software Interfaces** 13](#_Toc8678371)

[**2.1.4** **Hardware Interfaces** 13](#_Toc8678372)

[**2.1.5** **Communication Interfaces** 13](#_Toc8678373)

[**2.1.6** **Memory Constraints** 14](#_Toc8678374)

[**2.1.7** **Operations** 14](#_Toc8678375)

[**2.1.8** **Site adaptation requirements** 14](#_Toc8678376)

[**2.2** **Product functions** 14](#_Toc8678377)

[**2.3** **User characteristics** 20](#_Toc8678378)

[**2.4** **Constraints** 20](#_Toc8678379)

[**2.5** **Assumptions and Dependencies** 21](#_Toc8678380)

[**2.6** **Apportioning of requirements** 21](#_Toc8678381)

[**3.** **SPECIFIC REQUIREMENTS** 23](#_Toc8678382)

[**3.1** **External interface requirements** 23](#_Toc8678383)

[**3.1.1** **User Interfaces** 23](#_Toc8678384)

[**3.1.2** **Hardware Interfaces** 25](#_Toc8678385)

[**3.1.3** **Software Interfaces** 25](#_Toc8678386)

[**3.1.4** **Communication Interfaces** 26](#_Toc8678387)

[**3.2** **Classes/Objects** 26](#_Toc8678388)

[**3.3** **Performance Requirements** 26](#_Toc8678389)

[**3.4** **Design Constraints** 26](#_Toc8678390)

[**3.5** **Software system attributes** 27](#_Toc8678391)

[**3.5.1** **Reliability** 27](#_Toc8678392)

[**3.5.2** **Availability** 27](#_Toc8678393)

[**3.5.3** **Security** 28](#_Toc8678394)

[**3.5.4** **Maintainability** 28](#_Toc8678395)

[**4.** **SUPPORTING INFORMATION** 29](#_Toc8678396)

[**5.** **REFERNCES** 31](#_Toc8678397)

**LIST OF FIGURES**

[**Figure 1 LearnWorlds logo 10**](#_Toc8656997)

[**Figure 2 Echo360 Logo 11**](#_Toc8656998)

[**Figure 3 Relay Logo 11**](#_Toc8656999)

[**Figure 4 Use Case Diagram for 'Content Based Video Segmentation into Topic Units' 15**](#_Toc8657000)

[**Figure 5 Activity Diagram for 'Content Based Video Segmentation into Topic Units' 16**](#_Toc8657001)

[**Figure 6 User Interface for home screen 23**](#_Toc8657002)

[**Figure 7 User Interface for uploading video 24**](#_Toc8657003)

[**Figure 8 User Interface for topic segmentation 24**](#_Toc8657004)

[**Figure 9 Cloud Architecture of the System 29**](#_Toc8657005)

[**Figure 10 High level system architecture 30**](#_Toc8657006)

**LIST OF TABLES**

[**Table 1 List of Abbreviations and Acronyms 7**](#_Toc8553283)

[**Table 2 Product feature comparison 12**](#_Toc8553284)

[**Table 3 Use Case Scenario for Uploading course materials 17**](#_Toc8553285)

[**Table 4 Use case scenario for video segmentation 18**](#_Toc8553286)

[**Table 5 Use case scenario for editing video segments 19**](#_Toc8553287)

# **INTRODUCTION**

## **Purpose**

The purpose of this document is to present a detailed overview about the software product that will be developed as our final year research project: A Platform for Improving Search ability and Interactivity of Recorded Lectures. Details regarding software requirements, project scope, target audience, functional requirements, non-functional requirements, user interfaces and design constraints will be addressed throughout this document. Overall this document will give a through explanation of our software product, its parameters and goals.

The main targeted audience of this document is the project supervisor, co-supervisor and team members of this research. However, since this is written in an easily understandable format, any individual will be able to read and understand this document. Also this can be specifically useful for anyone interested in carrying out this research forward. At the same time, this document serves as a key reference document for developers, testers and users throughout the entire software development life cycle.

## **Scope**

This document covers the requirements for release of the proposed system ‘Platform for Improving Search ability and Interactivity of Recorded Lectures’. Mention will be made throughout this document of selected probable features of future releases. The purpose of this is to guide developers in selecting a design that will be able to accommodate the full-scale application. This research is mainly focused on developing a system for automatically improve the interactivity and accessibility of recorded lectures and other supporting course materials in just few clicks. Hence our main goal is to create a web platform which can intelligently add interactivity and accessibility to course materials with minor human intervention thereby improving both learner engagement and user friendliness for lecturers who upload course materials to Learning Management Systems.

According to the identified requirements through the initial study, product can be divided into four main components as follows:

* Matching lines of code from sample code to occurrences in the video
* Content based lecture video segmentation into topic units
* Automatic question generation from lecture content
* Video De-noising and synchronization with lecture slides

At the time of writing this document, to the best of our knowledge, there is no such system that is capable of performing all the above mentioned features at the same time. According to the knowledge that we gained from existing product analysis and literature surveys, content based lecture video segmentation into topic units and automatic question generation from lecture content are some of the novel features that is proposed for this system, and is not available in any of the existing products. The main focus in this document is to present details about the functionality related to Content based lecture video segmentation into topic units.

Content based lecture video segmentation into topic units has many notable benefits such as:

* Makes information search easier
* Reduce learning time
* Enhance overall learning experience

Through this feature lecturers will be able to upload a lecture video, probably of 1-3 duration, and the system will automatically analyze the video content and identify the time frames of possible topic transitions, and split the original video into subcomponents depending on the identified time frames.

## **1.3 Acronyms and Abbreviations**

Table 1 shows the list of abbreviations and acronyms.

|  |  |
| --- | --- |
| SRS | Software Requirements Specification |
| LMS | Learning Management Systems |
| MOOC | Massive Open Online Course |
| API | Application Programming Interface |
| SaaS | Software as a Service |
| AWS | Amazon Web Services |

Table 1 List of Abbreviations and Acronyms

## **1.4 Overview**

The initial release of our product will be deployed as web platform using some of the services provided by cloud computing. The primary users of our system will lecturers, who wish to create course materials, and students/learners who will utilize these materials for learning.

This SRS will cover in detail all the functional and non-functional requirements of the sub component, ‘Content based lecture video segmentation into topic units’. Details will be spread across three chapters each covering a different perspective of the component. The first chapter gives an overall description of the whole system and the subcomponent which will be the main focus in this document, along with a brief description about the purpose and the scope of this document.

The second chapter will present and in-depth overview of the system in in user perspective. Sections such as product perspective, product functions, user characteristics and constraints under which the system will operate, will be covered under this chapter. In the first sub-section ‘product perspective’, the proposed system will be compared with related existing systems. It then moves on to describe details about several interfaces of system, memory constraints and operation of users to provide a better understanding about the product to the readers. This chapter will finally conclude with apportioning of requirements.

The third and the last chapter of this SRS will focus on specific requirements of the product and is primarily written for developers. This chapter will describe about the functionalities mentioned in chapter two in a more technical perspective. The latter part of this chapter provides details about the system attributes such as reliability, maintainability, availability and security.

Both chapter two and three describes about same sub component system but is intended for two different types of target audiences and will have different forms language. The document finally concludes providing supporting information regarding the contents of the document.

# **OVERALL DESCRIPTIONS**

Today, e-learning has become an essential component of higher education for both teachers and students. According to a study on the effectiveness of e-learning on education, it was found that students nowadays are more satisfied with web enhanced learning when compared to a traditional classroom environment [1]. Therefore, it is common to see universities and higher education institutes adopting some form of e-learning to assist their students. Many institutes use their own customized version of a LMS to provide online course material.

Online education is beneficial for both students and teachers in many ways. For instance, as lecture content is always available online, the possibility of missing a lecture is low and teachers can ensure that students have access to course material irrespective of time and location [1]. Recorded lecture videos also enable students with different styles of learning and different levels of understanding to obtain a better grasp of the subject. For example, those who are familiar with the work can skip ahead to a section of interest while those that need more time to understand the concepts can pause and rewind to digest the lecture at their own pace [2] [3].

Usually, many LMSs enable lecturers to upload course material such as tutorials, lab sheets, lecture slides and recorded lecture videos. Whilst videos are more effective because they address both visual and auditory aspects of teaching [4], many students find it tedious to watch recorded lecture videos because of its duration, which normally lasts around 1 - 3 hours and its lack of interconnectivity and relevance to other course material [5]. Although there are many video-creation and editing platforms which allow users to create interactive course material, these methods are time consuming as they require the lecturer to spend a lot of time editing and making the video interactive.

As stated in the above paragraph, one of the major drawbacks identified in recorded lecture videos is its duration, which is relatively long and usually exceeds the attention span of an average student. According to the researches that have been conducted, it was identified that the maximum attention span of an adult is about 20 minutes. Therefore it is highly beneficial for the students if the raw lecture video can be split into subparts of smaller durations. It is even more beneficial if this splitting is done according to the occurrences of the topic transitions in the original video, in other words, splitting the video into cohesive topic units. This is the main objective of the subcomponent: Content based lecture video segmentation into topic units. Hence this one of the most important parts of this research [6].

This component will mainly focus on segmenting videos based on the retrieved transcripts. Out of the main three types of video segmentation features available (textual, visual, audio), textual approach is used, because transcripts are a rich source of information in this specific domain of lecture videos.

As a whole, our system will be primarily deployed as a web application, however, it can be further extended as a plugin for existing MOOCs and LMSs. The system can also be deployed on cloud and provided as a, Software as a Service product, for customer

## **Product Perspective**

According to the background study and existing product analysis that was carried out, following products were identified as systems with similar objectives:

* LearnWorlds
* Echo
* TechSmith Relay

Brief descriptions about each of the above mentioned products are provided below:

1. **LearnWorlds**



Figure LearnWorlds logo

LearnWorlds [7] is a platform that enables users to create and sell online courses. It allows content creators (instructors) to upload video clips, documents, quizzes and other reference materials to an online course and customize by using drag and drop tools to show course outline, branding, etc.

1. **Echo360**



Figure Echo360 Logo

Echo360 [8] is an active learning platform that analyses student engagement data and enables teachers to create and distribute videos. It has features for online and offline video recording as well as streaming. It also includes a built-in video editor. Echo360 provides auto transcribing for lecture videos with automatic speech recognition, linking class presentations and student note-taking to specific moments in the video. Major LMS platforms also can be integrated to Echo360.

1. **TechSmith Relay**



Figure Relay Logo

TechSmith Relay [9] is a video creation and sharing platform that maximizes student engagement in online courses. It allows instructors to create engaging and interactive videos using a web-based editor and integration with TechSmith Camtasia. Relay also provides methods to add captions and embed questions in the video. Integration with popular LMS is an added feature in TechSmith Relay.

Even though there are several products already available with similar objectives, they mostly focus on the use of manual processes that involve human intervention to make the content more interactive and increase the searchability. Our proposed solution aims to reduce the amount of human interaction needed for this process by introducing a platform which will analyze and augment the content automatically.  Table 2 is a comparison of the proposed system with existing systems in the market.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | LearnWorlds | Echo360 | TechSmith Relay | Our Solution |
| Matching lines in code samples to occurrence in recorded lectures | ✗ | ✗ | ✗ | ✔ |
| Automated segmentation of lecture video into topic units | ✗ | ✗ | ✗ | ✔ |
| Matching slides with the lecture video | ✗ | ✔ | ✗ | ✔✔ |
| Automated noise removal from the video | ✗ | ✗ | ✗ | ✔ |
| Automatic question generation | ✗ | ✗ | ✗ | ✔ |
| Embedding questions in video playback | ✔ | ✔ | ✔ | ✔✔ |
| Automated question embedding in video playback | ✗ | ✗ | ✗ | ✔ |

✗ : Not available

✔ : Available

✔✔ : Improved

Table Product feature comparison

When considering about the specific component ‘Automated segmentation of lecture video into topic units’, there has been many research studies conducted in this area, however none of the existing products have this feature. Even though software like Camstatia enables splitting of recorded lecture videos, this quite time consuming and requires human intervention, where the user will have to move the play head to specific points where the video need to be segmented.

* + 1. **System Interfaces**
  + Python 3 sdk
  + PyMongo driver
    1. **User Interfaces**
* Platform Homepage: *list of available lecture videos with a small introduction to the platform.*
* Interface for delete/modify video: *Allows the user to do changes to the uploaded video/ delete the video*
* Interface for manual segmentation of video: Allows the user to select the points of segmentation manually
* Interface for automatic segmentation of the video: *Uploaded lecture video can be submitted for topic segmentation using this interface*
  + 1. **Software Interfaces**
* Tensorflow
* Keras – Deep learning library
* Node js runtime
* FFMPEG library
* AWS Cloud services
* WebStorm
  + 1. **Hardware Interfaces**

No special hardware is required other than a regular PC or a laptop with internet connection to access the web application.

* + 1. **Communication Interfaces**

A high-speed internet connection is preferred (ADSL, HSDPA, 4G LTE or FTTH).

* + 1. **Memory Constraints**

Minimum RAM capacity of 2GB.

* + 1. **Operations**
  + Users should have access to internet to use the system.
  + Login to the system
  + Upload lecture videos and supporting materials
  + Segment long lecture videos into topics
    1. **Site adaptation requirements**

The product will be delivered across the web as a SaaS product. Since the end user interacts with a web-interface and all processing is carried out in the cloud, a JavaScript enabled web browser is all that is required.

## **Product functions**

Product functions of the sub component ‘content-based segmentation of lecture videos into topic units’ will be illustrated by the use case diagram in Figure 4, and activity diagram Figure 5. A further insight into the major functions of this subcomponent is given by the set of use case scenarios in Table 3, Table 4 and Table 5.

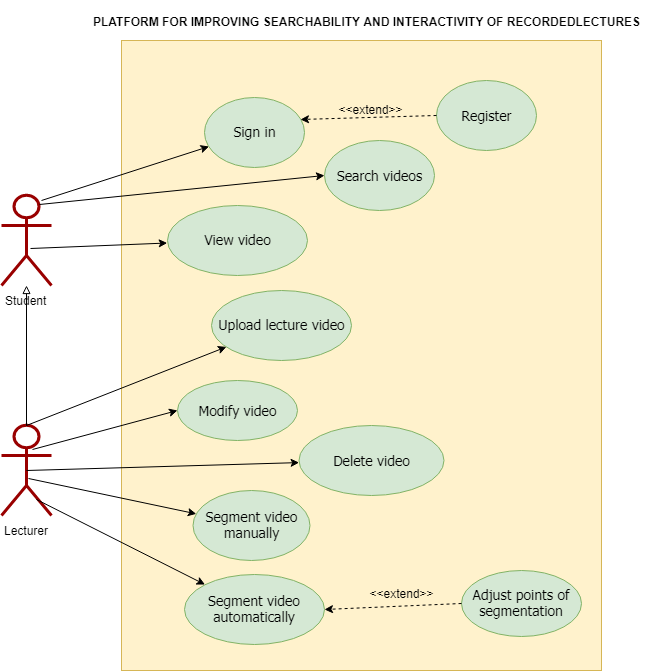


Figure Use Case Diagram for 'Content Based Video Segmentation into Topic Units'

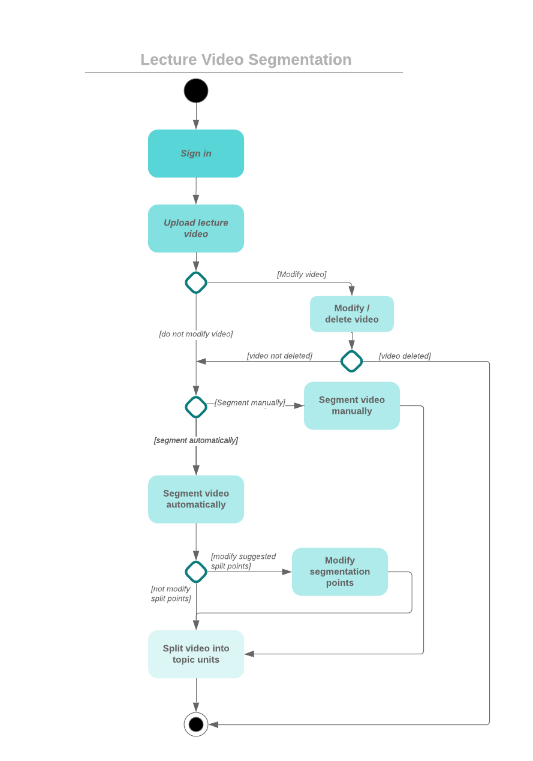


Figure Activity Diagram for 'Content Based Video Segmentation into Topic Units'

|  |  |
| --- | --- |
| **Use case name** | Upload a lecture video and reference materials |
| **Pre-Conditions** | Users need to login to the system using valid credentials. |
| **Post-Conditions** | Upload video and lecture materials to the platform |
| **Actor** | Lecturer/ course creators |
| **Main success scenarios** | 1. Use case starts with user logging into the system. 2. User navigate to upload content page. 3. User clicks ‘upload video’ button 4. User select video files from the computer 5. System displays the uploading progress. 6. User clicks the ‘upload content’ button. 7. User select lecture slides from the computer. 8. System displays the uploading progress. 9. System displays uploaded video and lecture slides. |
| **Extensions** | 5b. Upload fails and system displays error message.  8a. Upload fails and system displays error message. |

Table 3 Use Case Scenario for uploading course materials

|  |  |
| --- | --- |
| **Use case name** | Segment video into topic units |
| **Pre-Conditions** | Users need to login to the system using valid credentials.  There should be an already uploaded lecture video |
| **Post-Conditions** | Set of topically segmented lecture videos |
| **Actor** | Lecturer/ course creators |
| **Main success scenarios** | 1. Use case starts with user logging into the system. 2. User selects the video to be segmented 3. User clicks the ‘Segment into topics’ button. 4. System displays the ‘processing.......’ progress bar. 5. System displays the segmentation points of the video. 6. User clicks the ‘Proceed’ button. 7. System displays the segmented videos |
| **Extensions** | 6a. User replays the video at segmentation points.  6b. User changes the segmentation points manually and clicks the ‘Proceed’ button. |

Table 4 Use case scenario for video segmentation

|  |  |
| --- | --- |
| **Use case name** | Edit video segments |
| **Pre-conditions** | User should login to the system using valid credentials. |
| **Post conditions** | System should modify the lecture video according user input. |
| **Actor** | Lecturer |
| **Main success scenarios** | 1. User navigates to the relevant course page 2. User selects the segment to be edited 3. User selects the action to be performed, delete/modify video 4. User does the necessary modifications and click the ‘save’ button. 5. System displays the modified lecture video segments |

Table 5 Use case scenario for editing video segments

## **User characteristics**

The target audience of our system is universities and higher educational institutes who wish to provide course materials for their students online. Therefore, our users fall into 2 main categories:

* + Lecturers
  + Students

Lecturers will use this system to upload course materials, and make it more interactive for the students, whereas students will use the system for learning and studying purposes. Both the categories of users do not require any prior technical knowledge besides basic interactions with a web application.

## **Constraints**

The proposed system consists of a front-end web application, which communicates with the backend services to perform tasks. All the heavy computational tasks will be carried out by the backend, and the UI interacts with the users to perform operations as requested. Constraints of the system are as follows:

* + There should be a proper internet connectivity for user to access the web application and for communication between frontend and backend process.
  + Angular.js will be used to develop frontend client application while the backend processes are developed using Python and machine learning frameworks.
  + The backend of the system will follow a microservices architecture where several components of the backend will be developed independent of each other.
  + The developers of each microservice will be free to choose whatever technology stack suits their purpose.
  + Backend system must have minimum of 4GB RAM.
  + After performing the necessary analysis on the uploaded video, the final optimized video shall not be stored on the system internally. All temporary files will be removed after the final video is hosted on a suitable video hosting platform.
  + Topic segmentation feature is only available for lectures in English language.
  + The platform may not provide all the functionalities of a typically LMS, as our main goal is to create interactive and easily accessible course materials. The system can be improved to be exposed as a plugin for existing LMSs

## **Assumptions and Dependencies**

* + Users should have basic computer literacy, in order to use the system efficiently.
  + Initial release of platform will be targeted for desktop web browsers. Future releases shall introduce mobile friendly design.
  + All browsers are assumed to be capable of running the frontend JavaScript application.
  + Only screen recorded lectures (voice over presentation) will be supported for the initial release. Future releases will support more lecture video patterns such as talking head, lecture capture, etc.

## **Apportioning of requirements**

The requirements described in the first two sections of this document are referred to as the primary specifications and those in section three are the requirements/functional specification. The two levels of requirements are intended to be consistent with each other. Any inconsistencies will be logged as defects. If a requirement is stated within both the primary specifications and the functional specifications, the application will be built from the functional specification since it is more detailed.

‘Essential requirements’ stated in section three are to be implemented in the initial release of this product. 'Desirable requirements', stated below, will be implemented in this release if possible but are not committed to by the developers. It is anticipated that they will be part of future release. 'Optional requirements' will be implemented at the discretion of developers.

* + Essential Requirements:
    1. The content creator should be able to upload a lecture video and source code file so that it can be processed by the system.
    2. The learner should be able to navigate the lecture video using the lines of code in a source code file as the reference.
    3. The learner should be able to navigate the lecture video using the lecture slides.
    4. The system should automatically generate and suggest suitable questions for each lecture video.
    5. The system should automatically suggest segmentation points where the video can be broken down into smaller sections.
  + Desirable Requirements:
    1. Generate data on popularity of lecture videos.
    2. Have a feedback system where students can reach the lecturer who created the videos.
    3. Generate statistical report for lectures and how learners engage with them.
    4. Maintain a student profile and grading system.
  + Optional Requirements:
    1. Create a search function to search through the indexed video file for code occurrences.

# **SPECIFIC REQUIREMENTS**

## **External interface requirements**

* + 1. **User Interfaces**

The description of user interface in section 2.1.2 showed a basic description of the interfaces to provide product perspective. The interfaces mentioned under section 3.1.1 provide a more detailed representation of the required interfaces and are intended for the developers of the system. Mock user interfaces are shown in Figure 6, Figure 7 and Figure 8.

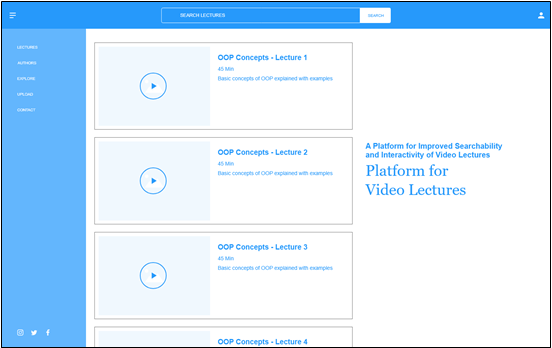


Figure User Interface for home screen

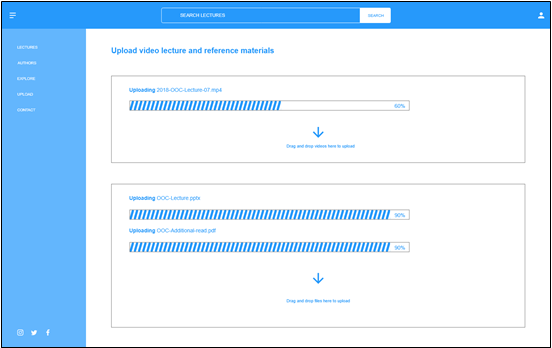


Figure User Interface for uploading video

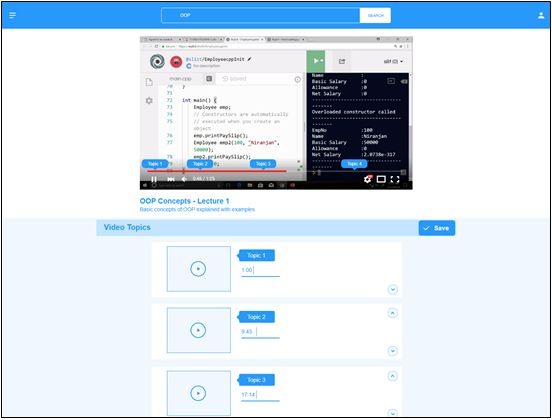


Figure User Interface for topic segmentation

* + 1. **Hardware Interfaces**

No specialized hardware is required for the backend. A regular computer with an internet connection and web browser capable of running JavaScript will be required to access the web application.

* + 1. **Software Interfaces**
  + **TensorFlow** – TensorFlow is an open source machine learning library that is commonly used for building machine learning applications. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets developers to easily build and deploy ML (Machine Learning) Powered applications. In this sub component, TensorFlow will be used for training the algorithm used for topic detection.
  + **Keras** – Keras is an open source library written in python, which is capable of running on top of TensorFlow. Specifically used for building and training neural networks.
  + **Node js runtime** – Node js is an open source, cross-platform for developing server side and network applications and will be mainly used for the backend of our research project.
  + **FFMPEG** – FFMPEG is a command line tool mainly used for audio and video manipulations. It also be used along with languages like node js.
  + **AWS cloud services** – AWS is one of the most popular cloud vendors in the world today. Several services provided by AWS will be used throughout this research for several purposes. Some of the services used will be EC2(Elastic Cloud Compute), RDS (Relational Database Service), S3 (Simple Secure Storage) etc.
  + **Webstorm** – IDE used for modern JavaScript development. Supports JavaScript, Typescript, CSS and HTML. Also supports frameworks such as React, Angular, Vue
    1. **Communication Interfaces**

A stable high-speed internet connection is required.

## **Classes/Objects**

* Essential
  + MediaController – To handle upload and streaming of video and other media content.
  + Authenticator – To generate authentication tokens for valid user credentials.
  + VideoPreProcessor – Handle initial processing of uploaded videos
  + ServiceController – Apply the required algorithm or service when called.
* Optional
  + DataAnalyzer – To Generate analysis reports

## **Performance Requirements**

The proposed platform is expected to run on a standard desktop computer or a laptop computer with minimum requirements to access internet. Considering functional requirements following performance requirements are identified.

* + Platform should support minimum 1000 simultaneous requests.
  + Loading a webpage should not take more than 4000 milliseconds
  + Once a page is loaded, each user interaction should not have a latency of more than 500 milliseconds.
  + Video quality should be at least 480p

## **Design Constraints**

* + The system is designed to mainly focus on computer science domain specific course materials.
  + Topic segmentation of the lecture videos will mainly depend on the video’s textual content (transcript), however visual content will also be taken into consideration

## **Software system attributes**

* + 1. **Reliability**

Reliability is the probability of failure-free software operation for a specified period of time in a specified environment. Software reliability is also an important factor affecting system reliability [10] .

Reliability of this sub component depends on the reliability of the algorithm used for topic segmentation. Since the algorithm depends on the transcripts generated from the video, actual reliability will depend on the audio quality of the uploaded video. Better the audio quality, higher the accuracy of the generated transcripts and higher the reliability.

* + 1. **Availability**

Availability is one of the major system attributes that is considered in the modern world. It can be simply defined as the probability that the system is functioning properly when requested for use. Availability is also a key factor in the proposed system because the system needs to be available for use whenever requested by the lecturers and students. As the backend will be deployed in a cloud environment, the availability of the system will depend on the cloud provider’s services and performance. However, we expect the system to be available about 90% of the time

* + 1. **Security**

Suitable encryption methods should be used to encrypt all data generated by the system. Special attention should be given to maintain the confidentiality of the Users’ personal information if it is stored. The user should be authenticated using a suitable framework before utilizing the system. All passwords must be stored as hashed values. For internal system communication each call which is transmitted to a service over the internet must use token-based authentication based on the OAuth [11] or OAuth2 standard. Furthermore Static Application Security Testing (SAST) methods [12] should be employed in the continuous integration pipeline to identify security vulnerabilities in the source code.

* + 1. **Maintainability**

Maintainability can be defined as the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment. The system is divided into functionally cohesive modules to improve maintainability. Also, we practice software engineering best practices throughout this research, which will improve the overall maintainability of the system. The architecture of the system will be deployed in a manner that will support high maintainability in the cloud environment in which it is deployed.

# **SUPPORTING INFORMATION**

**Appendix A: High-level architecture diagrams**

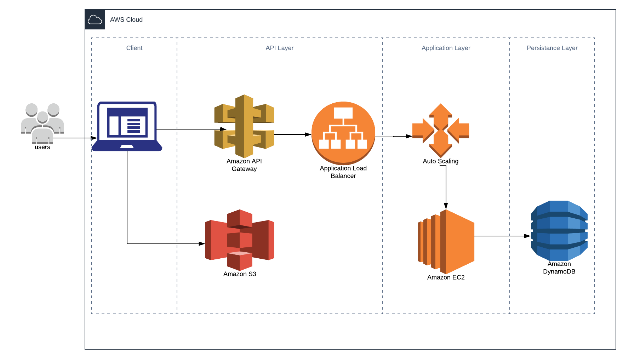
****

Figure Cloud Architecture of the System

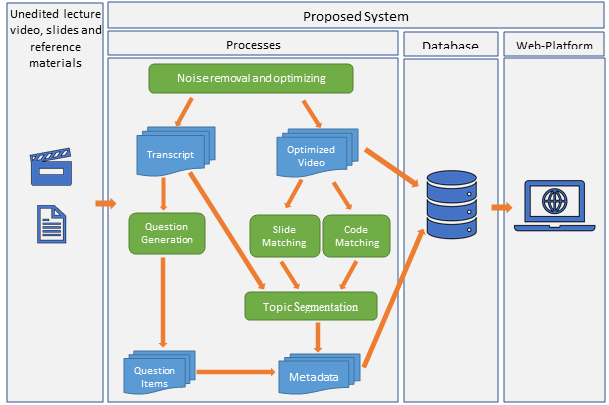


Figure High level system architecture

# **REFERNCES**

|  |  |
| --- | --- |
| [1] | "Importance and Effectiveness of E-learning," 1 December 2015. [Online]. Available: https://higheredrevolution.com/importance-and-effectiveness-of-e-learning-9513046ed46c. [Accessed 6 May 2019]. |
| [2] | T. M. M. S. a. C. H. M. Brandsteidl, "Replacing traditional classroom lectures with lecture videos: an experience report," *Proceedings of the 8th edition of the Educators' Symposium,* 2012. |
| [3] | H. L. a. V. S. M. Maher, Flipped classroom strategies using online videos, North Carolina: UNC Charlotte, 2015. |
| [4] | M. L. J. F. N. H. C. M. Chau, "Segmentation of Lecture Videos Based on Text: A Method Combining Multiple Linguistic Features," *Proceedings of the 37th Hawaii International Conference on System Sciences, Hawaii,* 2004. |
| [5] | Z. Woolfitt, "The effective use of video in higher education," 2015. [Online]. Available: https://www.inholland.nl/media/10230/the-effective-use-of-video-in-higher-education-woolfitt-october-2015.pdf. [Accessed 11 May 2019]. |
| [6] | K. Wilson and J. H. Korn, "Attention During Lectures: Beyond Ten Minutes," in *Teaching of Psychology.* , 2007, pp. 85-89. |
| [7] | "Learnworlds: Create and Sell Online Courses from Your Own Website," Learnworlds, 2019. [Online]. Available: https://www.learnworlds.com/. [Accessed 4 May 2019]. |
| [8] | "About Echo360 - The Smarter Video Platform for Education," Echo360, 2019. [Online]. Available: https://echo360.com/about/. [Accessed 5 May 2019]. |
| [9] | "TechSmith | Create high-quality images and videos for better communicatio," TechSmith, 2019. [Online]. Available: https://www.techsmith.com. [Accessed 4 May 2019]. |
| [10] | J. Pan, "Software Reliability," 1999. [Online]. Available: https://users.ece.cmu.edu/~koopman/des\_s99/sw\_reliability/. [Accessed 10 May 2019]. |
| [11] | "OAuth 2.0 — OAuth," Oauth.net, 2019. [Online]. Available: https://oauth.net/2/. [Accessed 13 May 2019]. |
| [12] | "Static Application Security Testing (SAST) - Gartner IT Glossary," Gartner IT Glossary, 2019. [Online]. Available: https://www.gartner.com/it-glossary/static-application-security-testing-sast. [Accessed 13 May 2019]. |