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

Project ID: 19-087

Software Requirement Specification

Hettiarachchi H. A. I. S.  
IT 15146366

Bachelor of Science (Honors) in Information Technology

Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

April 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

March 2019

**DECLARATION**

I declare that this my own work and this dissertation does not incorporate without acknowledgment of any material previously submitted for a Degree or Diploma in any other university or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

H. A. I. S. Hettiarachchi

IT15146366

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: Date:

Signature of the co-supervisor: Date:

**TABLE OF CONTENTS**

[1 INTRODUCTION 1](#_Toc9421525)

[1.1 Purpose 1](#_Toc9421526)

[1.2 Scope 1](#_Toc9421527)

[1.3 Definitions, Acronyms, and Abbreviations 2](#_Toc9421528)

[1.4 Overview 2](#_Toc9421529)

[2 Overall Descriptions 4](#_Toc9421530)

[2.1 Product Perspective 4](#_Toc9421531)

[2.1.1 System Interfaces 5](#_Toc9421532)

[2.1.2 User Interfaces 5](#_Toc9421533)

[2.1.3 Hardware Interfaces 6](#_Toc9421534)

[2.1.4 Software Interfaces 6](#_Toc9421535)

[2.1.5 Communication Interfaces 6](#_Toc9421536)

[2.1.6 Memory Constraints 6](#_Toc9421537)

[2.1.7 Operations 6](#_Toc9421538)

[2.1.8 Site Adaptation Requirements 7](#_Toc9421539)

[2.2 Product Functions 7](#_Toc9421540)

[2.2.1 Use Case Diagram 8](#_Toc9421541)

[2.2.2 Use Case Scenarios 9](#_Toc9421542)

[2.3 User Characteristics 11](#_Toc9421543)

[2.4 Constraints 12](#_Toc9421544)

[2.5 Assumptions and dependencies 12](#_Toc9421545)

[2.6 Apportion of Requirements 12](#_Toc9421546)

[3 Specific Requirements 13](#_Toc9421547)

[3.1 External Interface Requirements 13](#_Toc9421548)

[3.1.1 User Interfaces 13](#_Toc9421549)

[3.1.2 Hardware Interfaces 15](#_Toc9421550)

[3.1.3 Software Interfaces 15](#_Toc9421551)

[3.1.4 Communication Interfaces 15](#_Toc9421552)

[3.2 Classes/Objects 16](#_Toc9421553)

[3.3 Performance Requirements 16](#_Toc9421554)

[3.4 Design Constraints 16](#_Toc9421555)

[3.5 Software System Attributes 17](#_Toc9421556)

[3.5.1 Reliability 17](#_Toc9421557)

[3.5.2 Availability 17](#_Toc9421558)

[3.5.3 Security 17](#_Toc9421559)

[3.5.4 Maintainability 17](#_Toc9421560)

[4 References 19](#_Toc9421561)

[5 Appendix 20](#_Toc9421562)

**LIST OF FIGURES**

[Figure 2.1: High-level system diagram of automatic question generation process 7](#_Toc9421563)

[Figure 2.2: Use case diagram 8](#_Toc9421564)

[Figure 3.1: User interface for home screen 13](#_Toc9421565)

[Figure 3.2: User interface for uploading content 14](#_Toc9421566)

[Figure 3.3: User interface for video player 14](#_Toc9421567)

[Figure 3.4: User interface for questions embedded in video playback 15](#_Toc9421568)

[Figure 5.1: Activity diagram for question generation and selection 20](#_Toc9421569)

[Figure 5.2: High-level system architecture 21](file:///C:\Users\USER\Documents\SLIIT\CDAP\Documents\Software%20Requirements%20Specification%20-%20IT15146366%20-%2019-097.docx#_Toc9421570)

**LIST OF TABLES**

[Table 2.1: Product feature comparison 5](#_Toc9421571)

[Table 2.2: Use case scenario for upload lecture video and reference materials 9](#_Toc9421572)

[Table 2.3: Use case scenario for generating questions 10](#_Toc9421573)

[Table 2.4: Use case scenario for answering questions 11](#_Toc9421574)

# INTRODUCTION

## Purpose

The purpose of the document is to provide a detail description of requirements for

comprehensive design and analysis project 19-087, A Platform for Improving Searchability and Interactivity of recorded lecturers. The document will explain the purpose, features, interfaces, functional requirements, nonfunctional requirements, outcome of the software and constraints that software should operates.

This document is intended primarily for project supervisor, co-supervisor and team members and will be used as a reference document in development lifecycle. The document is written in a form that any person can read and understand which could help for any future researchers or developers who are interesting in similar kind of application.

## Scope

This document covers the requirements for initial release of automatic question generation component of the proposed platform for improving searchability and interactivity of recorded lectures. Automatic question generation component is used to increase the interactivity of the recorded lectures. In this component, a set of questions will be created automatically and presented to the video uploader to select that will embedded to the video playback upon selection. This will help content creators in creating questions for their lectures in a time efficient way and evaluating end users who watch the lectures while end users to evaluate themselves.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| SDK | Software Development Kit |
| OWL | Ontology Web Language |
| API | Application Programming Interface |
| AWS | Amazon Web Services |
| RAM | Random Access Memory |
| GB | Giga Byte |
| ADSL | Asymmetric Digital Subscriber Line |
| HSDPA | High-Speed Downlink Packet Access |
| FTTH | Fiber To The Home |
| SaaS | Software As A Service |

## Overview

The final product is targeted to be used by any person that creates recorded lecture content who are interested in increasing searchability and interactivity of content when delivering it to the learners. These learners also benefit from the platform by receiving more interactive lecture content with increased searchability. Following four main components will be used to achieve the overall goal of the system.

* Code extraction and source code matching
* Topic segmentation
* Question generation
* Audio enhancement and lecture slides matching

This software requirements specification is intending to cover functional and non-functional requirements automatic question generation component of the proposed system in detail. The first chapter provide and introduction to the system and component as well as the purpose of the document itself.

The second chapter is focused on overall overview of the proposed system that is unlikely to change in future versions. Details provided in this chapter will be focusing on general users of the system with comparing similar products in the market with their features against the proposed system. Further in the chapter, product functionalities of the platform explain in-detail using use cases, user characteristics. Constraints of the system, assumptions and dependencies will be discussed at the end of the chapter.

The third chapter will describe specific requirements of the system considering technical aspects of the requirements mentioned in the second chapter. Since the section is targeted for developers of the system terms used to describe details are technical. The document will conclude with supporting information regarding the content of the document.

# Overall Descriptions

e-Learning has become one of the most popular and essential components in higher education for both students and teachers. Studies show that students are more satisfied with web enhanced learning when compared to traditional classroom studies [1]. Therefore, more universities and higher education institutes are leaning towards to provide lecture content online. Lecture videos that recorded in the classroom or screen captured where a lecturer delivers the lecture using content like lecture slides and other examples is considered to be the most effective way of e-learning.

However, there are some drawbacks in providing lecture content online. Main two problems occurred in this e-learning system are lack of searchability and interactivity with the lecture videos. In lecture videos that are typically longer more than 30 minutes, it is very hard to search for a specific timeframe that explain a specific topic. Learners will have to seek through the video or listen to the entire video. When engaging with these lecture videos students will receive the minimum interactivity with the lecture and lecturer when compared to traditional classroom.

The Platform for Improving Searchability and Interactivity of Recorded Lectures aims to address these drawbacks by introducing an automated process of identifying combining the lecture materials to create an enhanced user experience with better searchability and interactivity with the lecture over web.

## Product Perspective

Since the popularity of e-learning is increased there are several platforms have gained the popularity for providing online courses for students in different models. In the background study carried out for project proposal revealed several notable solutions which have similar feature set and similar goals and objectives. Researches that has been carried out to address the automatic question generation mostly has used Heilman’s and Smith’s work [2] [3] . TedQuiz [4], SeMCQ [5], OntoQue [6] are few of the researches conducted. However, none of the solutions currently available for general usage does not address the generating questions from the lecture content. Most of the solutions support only embedding and prompting question on the playback screen. These questions need to be provided by lecture creators. Table 2.1 shows a comparison of features which are common across the platforms along with those that are improved in the proposed platform.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | LearnWorlds | Echo360 | TechSmith Relay | Our Solution |
| Automatic question generation | ✗ | ✗ | ✗ | ✔ |
| Embedding question on video playback | ✔ | ✔ | ✔ | ✔✔ |
| Automatic question embedding in video playback | ✗ | ✗ | ✗ | ✔ |

Table 2.1: Product feature comparison

✗ : Not available

✔ : Available

✔✔ : Improved

### System Interfaces

* Python SDK
* PyMongod Driver
* OWL API

### User Interfaces

* Platform Homepage: *list of available lecture videos with a small introduction to the platform.*
* Interface for selecting and modifying questions and answers: *automatically generated questions and answers will be showed to the lecturer where he/she can change the content and select which of them to be embedded to the video playback.*
* Interface for video playback: *lecture video will playback and will pause prompting question from the user. When user answers and ‘click’ proceed video will start the playback again.*

### Hardware Interfaces

No special hardware is required. All processing will be carried out in the cloud. A regular computer with an internet connection will be required to access the web application.

### Software Interfaces

* Node.js runtime
* AWS
* Protégé

### Communication Interfaces

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

### Memory Constraints

4GB RAM

### Operations

* Login to the site.
* Upload lecture video and lecture materials.
* Generate questions and answers based on the content of lecture.
* Modify automatically generated questions and answers.
* Select questions to be embedded in video playback.
* Answer questions from video playback.

### Site Adaptation Requirements

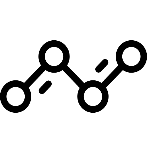
The product will be delivered across the web as a SaaS product. Since the end user interacts with a web-interface and all processing is carried out in the cloud, a JavaScript enabled web browser is all that is required.

## Product Functions



Transcript

Extracted Sentences and Words



Ontology

Reference materials

Question generation and distractor selection

Question Items

Figure 2.1: High-level system diagram of automatic question generation process

High level system diagram of automatic question generation process is show in Figure 2.1. Automatic question generation component of the platform can be discussed as follows.

* Develop a domain specific ontology to represent knowledge
* Extract sentences and words from the transcript of video and other provided reference materials
* Generate questions and answers for the given video using ontology and present to the user
* Embed user selected questions to the video

### Use Case Diagram

A use case diagram for automatic question generation component is shown in Figure 2.2.

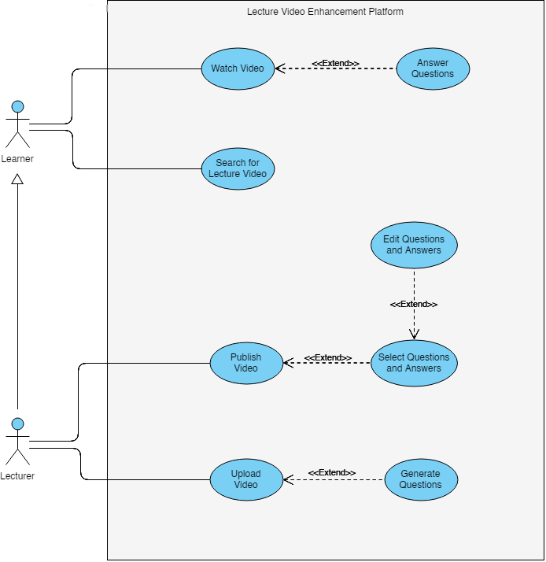


Figure 2.2: Use case diagram

### Use Case Scenarios

|  |  |
| --- | --- |
| **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** | 5a. Upload fails and system displays error message.  8a. Upload fails and system displays error message. |

Table 2.2: Use case scenario for upload lecture video and reference materials

|  |  |
| --- | --- |
| **Use case name** | Generate questions and answers from lecture content |
| **Pre-Conditions** | Users need to login to the system using valid credentials.  There should be an already uploaded lecture video and reference materials |
| **Post-Conditions** | Embed questions and answers to video playback |
| **Actor** | Lecturer/ course creators |
| **Main success scenarios** | 1. Use case starts with user logging into the system. 2. User selects the video to generate questions and answers. 3. User clicks the ‘proceed’ button. 4. System displays the ‘processing.......’ progress bar. 5. System displays the questions and answers generated. 6. User clicks the ‘Proceed’ button. 7. System displays the video with questions embedded. |
| **Extensions** | 6a. User changes the question and answers. |

Table 2.3: Use case scenario for generating questions

|  |  |
| --- | --- |
| **Use case name** | Answer question on video playback |
| **Pre-Conditions** | Users need to login to the system using valid credentials.  Published video should available with questions embedded. |
| **Post-Conditions** | User receives feedback for questions answered. |
| **Actor** | Lecturer/ course creators |
| **Main success scenarios** | 1. Use case starts with user logging into the system. 2. User select a lecture video to play. 3. Video pause and prompt a question. 4. User select the correct answer. 5. User click ‘next’ button. 6. Video unpause and start to play. 7. Video playback finishes. 8. System displays a feedback for questions answered. |
| **Extensions** | 5a. User select the ‘never ask again’ option. |

Table 2.4: Use case scenario for answering questions

## User Characteristics

The platform is targeted for higher education institutes and universities who wish to provide course materials for their students online. Therefore, users of the platform fall into 2 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

* Angular will be used to develop frontend client application and the backend processes will be developed using Python.
* The backend of the system will follow a microservices architecture.
* Internet connectivity is a must to carry out operations.

## Assumptions and dependencies

* Platform will be implemented to use from desktop environment. Future implementation shall introduce mobile friendly design.
* Only screen recorded (voice over presentation) lecture videos will be supported in the platform. Future releases will include the support for different lecture video patterns such as talking head videos, lecture capture, etc.

## Apportion of Requirements

The requirements described in sections 1 and 2 of this document are referred to as primary specifications. Those in section 3 are referred to as requirements (or functional) specifications. These requirements can be considered as essential requirements for the platform. Desirable requirements are mentioned below.

Desirable requirements:

* Generate statistical report for lectures and how leaners used them
* Maintain a learner profile and grading system for that enable the platform as a MOOC

# Specific Requirements

## External Interface Requirements

### User Interfaces

The description of user interfaces in section 2.1.2 explains a basic description of interfaces to provide product perspective. The interfaces mentioned under section 3.1.1 are more detailed representation of the required interfaces and are intended for the developers of the system.

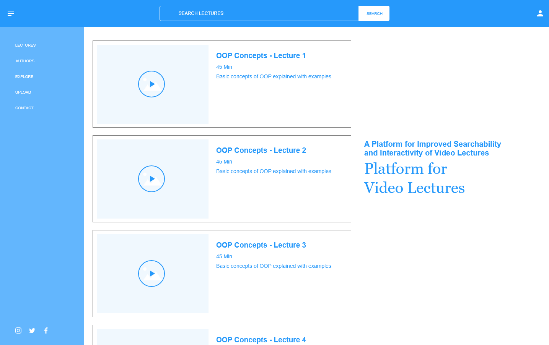
**

Figure 3.1: User interface for home screen

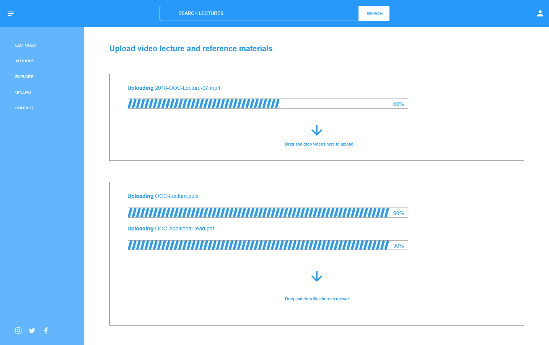
**

Figure 3.2: User interface for uploading content

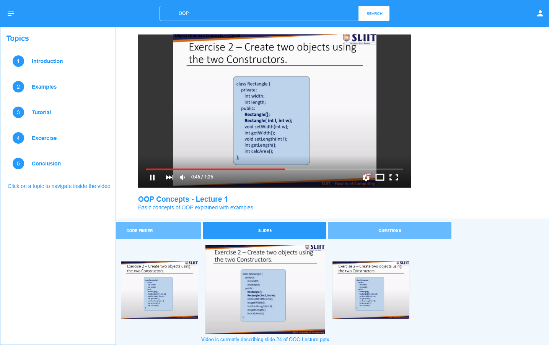
**

Figure 3.3: User interface for video player

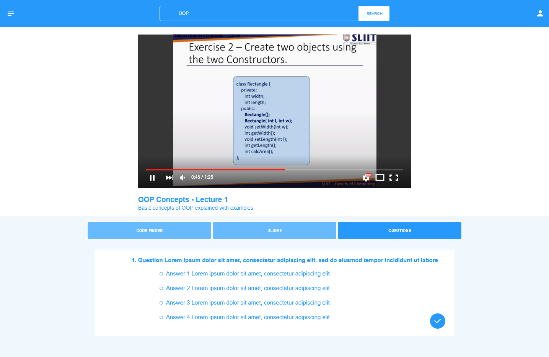
**

Figure 3.4: User interface for questions embedded in video playback

### Hardware Interfaces

No special hardware is required. All processing will be carried out in the cloud. A regular computer with an internet connection will be required to access the web application.

### Software Interfaces

* Node.js runtime
* Ffmpeg – video and audio manipulation
* AWS
* Protege

### Communication Interfaces

Internet connectivity will be required to access the web-application by the users. Standard internet connectivity interface will be required (internet router, cellular data).

## Classes/Objects

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

## Performance Requirements

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

* Time taken to analyze a video and generate questions should not exceed the duration of original video.
* Video quality should be at least 480p.
* System should support minimum 1000 simultaneous requests.
* Questions embedded to video should be presented on video playback in less than 3 seconds.
* Feedback of the questions answered should be presented at the end of playback within 10 seconds.
* More than 100 questions should be able to save for a one video lecture.

## Design Constraints

* The proposed system is mainly focused on playing videos to the users. User interfaces of the platform should be focused on enabling the interactivity and searchability provided by backend processes.
* Architectural design must emphasis on parallel processing since computational heavy tasks of the platform will take significant amount of time to complete.

## Software System Attributes

### Reliability

The system should be implemented using tried and tested libraries wherever possible to minimize the risk of failures. A failure will be defined as a software defect which will cause the entire system to be unavailable.

### Availability

Since all the components of the system are hosted on a cloud environment availability is one of the key system attributes. Any person with an internet connection and proper login should be able to access the platform at any given time of the day as well as system will be accessed by many users at once. System will be designed to carry out task simultaneous with scalability to handle every request as well as will be hosted on an environment that can scale with the requests receive to provide the maximum availability to its users.

### Security

Lecturers that use the platform to upload lecture content will require to login to the platform perform tasks. Login is not required for the learners to watch videos but will require to perform other tasks available for them. Encryption mechanisms will be used for these login credentials avoid unauthorized access to the platform.

### Maintainability

Maintainability of the platform will be highly considered when implementation phase since it is essential correct faults, improve performance or other attributes, add new features in future releases. Best practices in software engineering will be used from the design phase to the end of implementation phase to preserve maintainability.

# References

|  |  |
| --- | --- |
| [1] | "Importance and Effectiveness of E-learning," 1 December 2015. [Online]. Available: https://higheredrevolution.com/importance-and-effectiveness-of-e-learning-9513046ed46c. [Accessed 6 March 2019]. |
| [2] | M. Heilman, "Automatic Factual QUestion Generation from Text," Pittsburgh, 2011. |
| [3] | M. Heilman and N. A. Smith, "Question Generation via Overgenerating Transformations and Ranking," 2009. |
| [4] | Y. T. Huang, Y. M. Tseng, Y. S. Sun and M. Chen, "TEDQuiz: Automatic Quiz Generation for TED Talks VIdeo Clips to Assess Listening," in *2014 IEEE 15th International COnference on Advanced Learning Technologies*, Athens, 2014. |
| [5] | M. Tosic and M. Cubric, "SeMCQ - Protege Plugin for Automatic Ontongly-Driven Multiple Choice Question Tests Generation," in *11th International Protege Conference*, 2009. |
| [6] | M. Al-Yahya, "OntoQue: A Question Generation Engine for Educational Assesment Based on Domain Ontologies," in *2011 IEEE 11th International Conference on Advanced Learning Technologies*, Athens, 2011. |

# Appendix

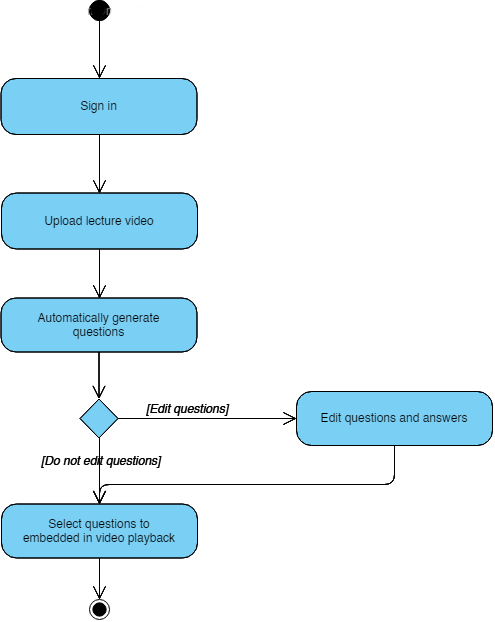


Figure 5.1: Activity diagram for question generation and selection

Figure 5.2: High-level system architecture



Unedited lecture video, slides and reference materials

Noise removal and optimizing

Optimized Video

Transcript

Code Matching

Question Generation

Slide Matching

Topic Segmentation

Metadata



Question Items

Proposed System

Processes

Database

Web-Platform