REAL-TIME E-LEARNING PLATFORM

19-079

Software Requirements Specification

A.M.H.B. Athapaththu

Dr. Malitha Wijesundara

B.Sc. Special (Honors) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

March 2019

REAL-TIME E-LEARNING PLATFORM

19-079

Software Requirements Specification

A.M.H.B. Athapaththu

Dr. Malitha Wijesundara

B.Sc. Special (Honors) Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

March 2019

DECLARATION

I declare that this is my own work and this system requirement specification 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.

Name	Student ID	Signature	Date
A.M.H.B.Athapaththu	IT16131002		

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

Signature of Supervisor Date

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Purpose

The purpose of this System Requirement Specification(SRS) is to provide a detailed view of the Video Transcoding component of the Real-time e-Learning Platform. This will allow to understand completely about the expectations from this system component highlighting all the key features, functional and non-functional requirements, system flow, interfaces, dependencies, user characteristics of this component of the system in details. It will also be a reference for the system design and implementation phases and for the overall research success. This document is intended primarily for reference of the supervisor, Dr. Malitha Wijesundara, co-supervisor Mr. Pramadhi Atapattu, research group members and any other personal interested in this field.

1.2 Scope

Primarily, purpose Real-time e-learning platform is to provide its users a real-life lecture hall experience through a web service that broadcast lectures and facilitate other features that are related to a physical lecture hall scenario. Video transcoding module of this product is in charge of converting an incoming RTMP stream into WebRTC format with multiple variants. These variants have different bitrates so that viewers with different levels of download bandwidth are able to consume live video streams at the best possible quality for their connection. Also this module will store live streamed video feeds to facilitate users with video playback for later.

1.3 Definitions, Acronyms and Abbreviations

Term	Definition
Latency	Delay over network

Table 1.3 1 Definitions for terms used in SRS

Acronym / Abbreviation	Definition
SRS	System Requirements Specifications
RTMP	Real-Time Messaging Protocol
PC	Personal Computer

Table 1.3 2 Glossary of Acronyms

1.4 Overview

Real-time e-Learning platform is a live lecture streaming web application which will stream lecture modules in real-time. There are many lecture modules to be streamed simultaneously and each lecture module contains two video feeds which are the video feed of the lecturer and the video feed from the lecturer's computer. The video transcoding component should be able to transcode, and stream these video feeds in real-time. In order to achieve this goal, transcoding server has to be in a very resourceful environment where it can perform well.

1.4.1 Main Goal

The main goal of the transcoding server is to transcode videos with minimum latency at all stages of data acquisition, data processing and broadcasting.

1.4.2 Specific Goals

Specific goals of the transcoding component are:

- Support multiple bitrates of video streams
- Support video playback by storing live streams
- Support 50 simultaneous video transcoding's

1.4.3 Users

This web service will be used by the students of the university as an aid for their academic activities and lecturers and administration to monitor students and lecturers to provide a better educational experience to students.

1.4.4 Organization of SRS

This document is mainly focused on Video Transcoding module of the Real-time e-learning platform.

Chapter two of this document will discuss more detailed description of the product comparing and clarifying the product services with other similar products, product interfaces etc. while chapter 3 describes requirement analysis of the system. It will focus on all functional and non-functional requirements to be implemented to the final product.

2 Overall Descriptions

Virtual education is an emerging concept around the world. Students are starting to adapt to learn more productively through internet than traditional classroom due to many reasons. One of the main reasons is that every student has a different pace of catching-up with the teaching where some of them can be considered as fast learners while others may be slow. In traditional classroom scenarios, there is no solution to this fact. Another reason is the schedule flexibility. Through virtual learning, students can access their courses at any time anywhere with internet giving students the full

control of their schedule of the day. Supporting the facts, universities provide students with the access to course materials via internet. Going beyond this, Real-time e-Learning Platform is a solution for the shortcomings in conventional education system mainly with universities.

This proposed system is a web-based application where students can watch live lectures and interact with the lecturer and also playback previous lectures online. A tracking camera will be tracking the lecturer movements while recording and both lecturer's video feed and the lecturer's computer screen feed will be streamed. Also this system contains features to interact with the lecture. Student can ask questions from the lecturer remotely via his or her webcam in real-time and also can use a virtual whiteboard that is provided with the system to describe any misunderstandings.

Video Transcoding

Video transcoding simply is the process of converting a video file from one format to another. This is very important to this e-learning platform because we target a wide range of devices and browsers to support this application. In this system, Transcoding server will receive two video feeds from tracking camera and the lecturer's PC using Real-Time Messaging Protocol (RTMP) in h.264 video standard. Transcoding server is now in charge of converting these RTMP feeds to WebRTC with minimum latency in order to broadcast to the clients. WebRTC is an open project that provides browsers and mobile applications with real-time communication capabilities. Therefore, broadcasting feeds in WebRTC will increase the supportive devices for the System. Also if a student wants to ask a question through webcam, that video feed will be sent back to lecturer through the transcoding server.

Another critical factor that has to be considered when trying to transmit data in real-time is network bandwidth which as developers we do not have the control of. So as a solution for this, we have introduced an adaptive bitrate feature which will request different bitrates of streams according to the strength of the user's network connection. Providing these multiple variants of the video streams is also another responsibility if the transcoding server.

Transcoding server will also store live streams to provide the playback capability for the users to watch lectures online.

2.1 Product Perspective

There are few other similar products that are currently available in the market. Eduscope is one of them.

Features	**Eduscope**	**Proposed System**
Real-Time Transcoding	Yes with considerable latency	Yes
Adaptive Bitrate Capability	No	Yes
Webcam support student questioning facility	No	Yes
Video playback	Yes	Yes

Table 2.1 1 Comparison with existing systems

2.1.1 System Interfaces

Transcoding module does not interact with any other external systems and hence there are no system interfaces.

2.1.2 User Interfaces

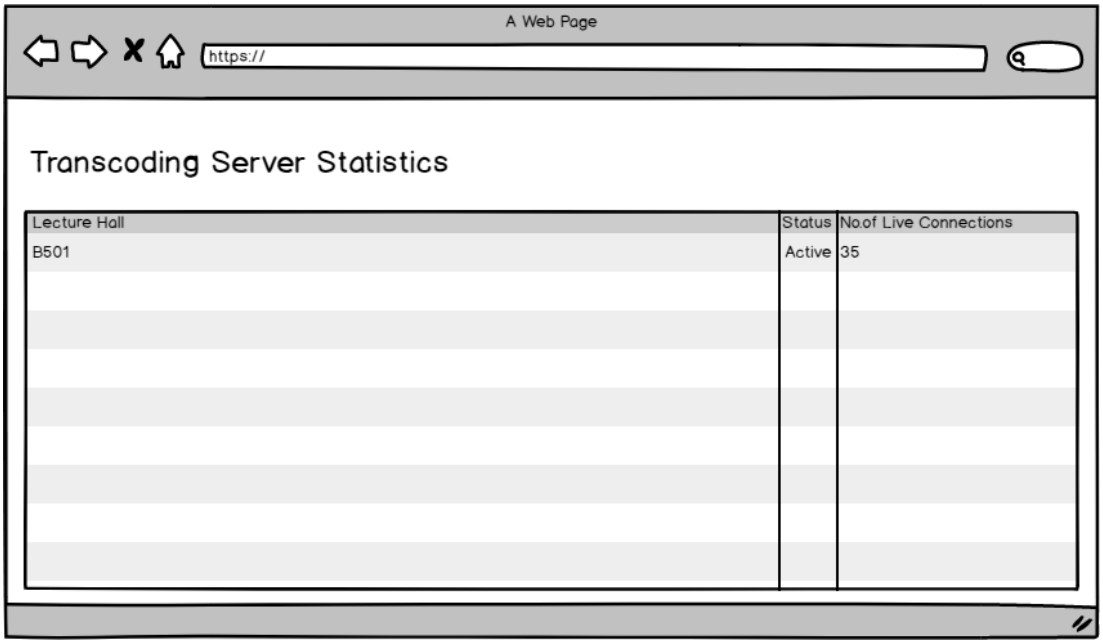

Figure 2.1.2 1 Server Statistics interface

Transcoding module is a backend server component. Statistics of the server will be show in the application for system administrator's usage.

2.1.3 Hardware Interfaces

Transcoding module acquires video feeds from a tracking camera and a desktop feed of a PC. Amplified audio from the lecture hall will also be received via an amplifier.

2.1.4 Software Interfaces

The Transcoding module will use kurento media server 6.9.0 for transcoding purposes and MySQL 5.7 as the database.

‘

2.1.5 Communication Interfaces

Communications between the lecture hall and the transcoding server requires an internet connection or a local network connection and for the communication between transcoding server and the student's PC will require internet connection.

2.1.6 Memory Constraints

Since video transcoding is a heavy computing process, the computer containing the transcoding server will require at least 6GB of RAM and 8GB is recommended.

Server is storing all the streamed video feeds to provide video playback for late viewers to watch lectures. Since videos are heavy in size, and there are many lectures to be recoded per day, this system will require approximately 10TB of storage capacity to store lectures per month.

2.1.7 Operations

- Student should log in to the website via internet
- Student should allow access to webcam and microphone of the PC

2.1.8 Site Adaptation Requirements

Students does not have to do any initial configurations to use this system. Giving web browser the permission access, the webcam and microphone of the device is enough for students to experience the application. Lecturers will have to log in to the system to start the live stream. All the configurations regarding cameras and PC's will be pre-configured.

2.2 Product Functions

When a lecturer starts streaming, Tracking camera feed will be use to analyze and track the lecturer's movements while same feed is to send to transcoding module as well. Transcoding module receives two H.264 video stream from tracking camera and Lecturer PC and will convert them into WebRTC and broadcast through the web app for the students to watch in real-time.

High level diagram of transcoding process

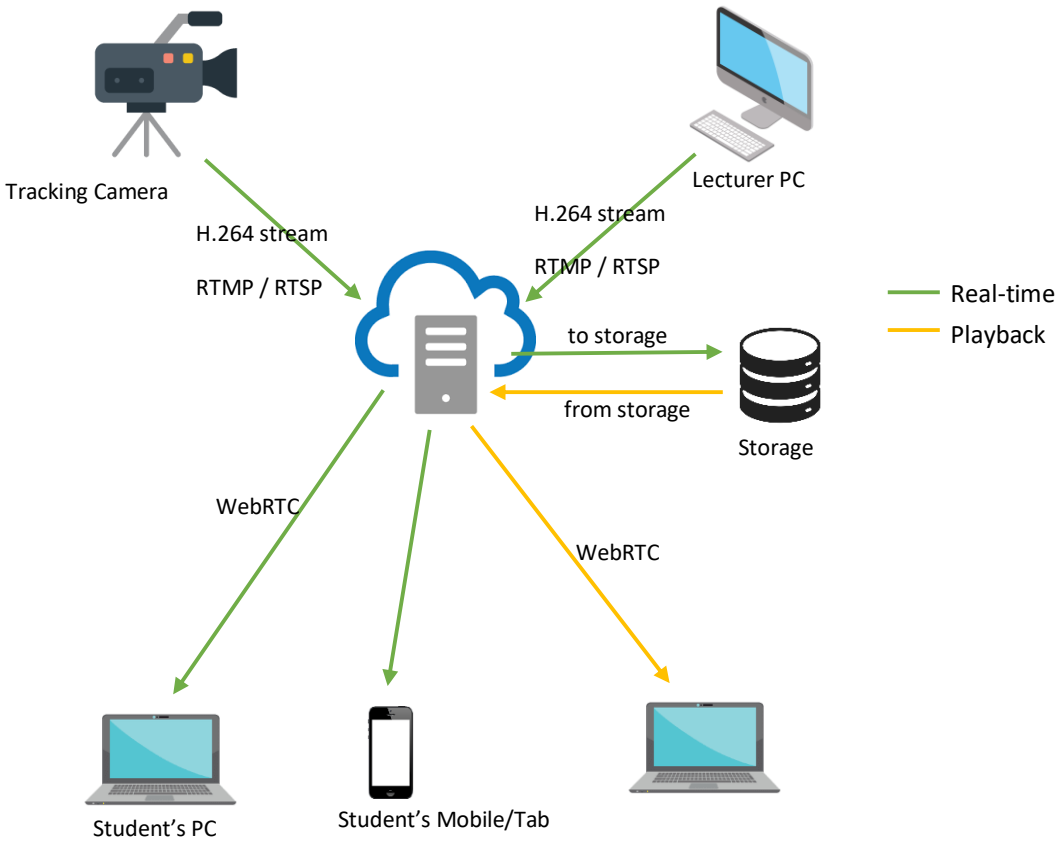

Figure 2.2.1 1 High level diagram of transcoding process

2.2.1 Use Case Diagram

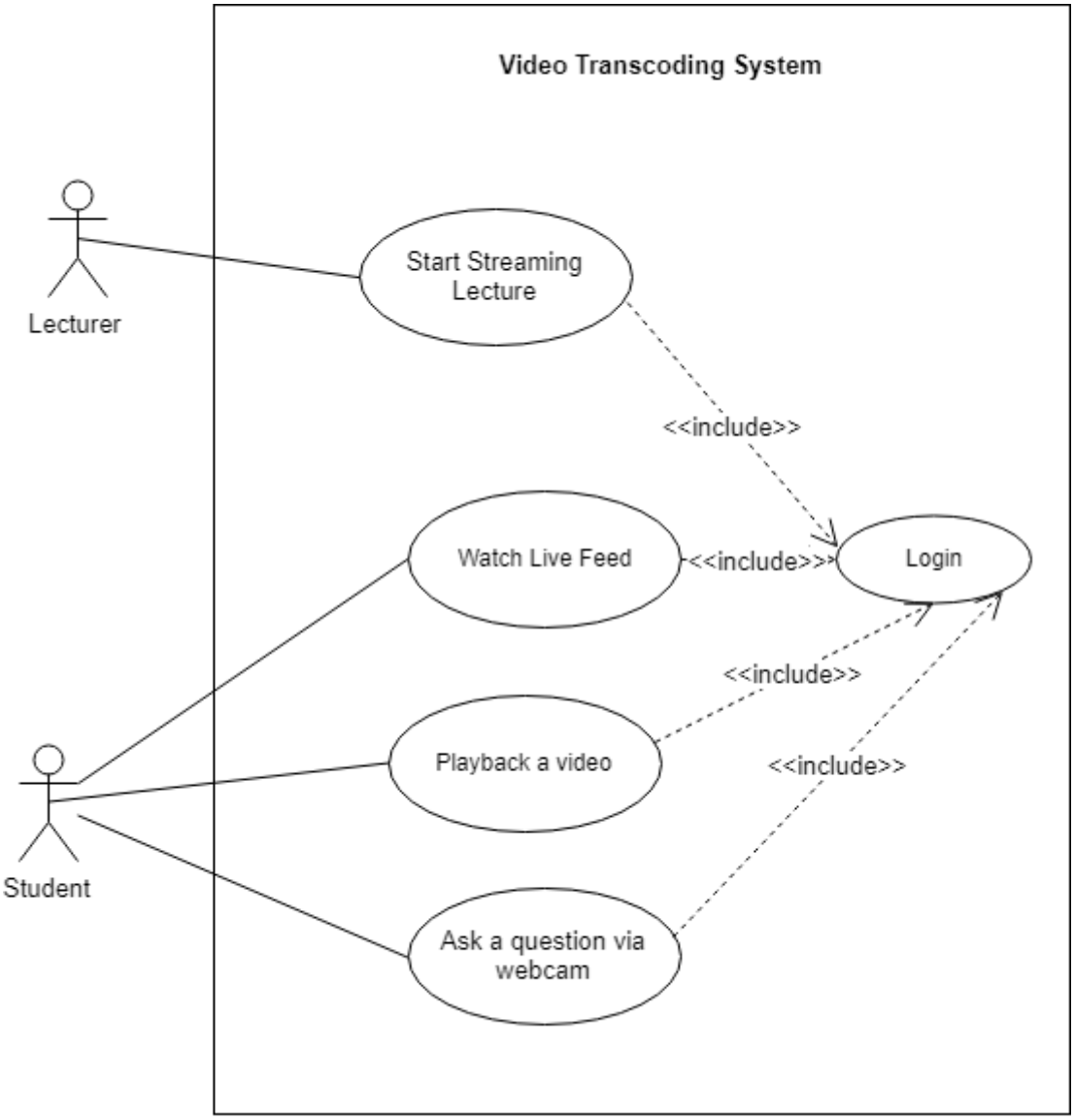

Figure 2.2.2 1 Use case diagram

2.2.2 Use Case Scenarios

Use case ID	UC_1
Use case name	Start streaming lecture
Goal in context	Streaming lecture

Pre-condition	Lecturer should be logged in	
Post-condition	None	
Primary actor	Lecturer	
Secondary actor	None	
Main flow	**Step**	**Action**
	1	The use case begins when Lecturer click on start streaming button on the application.
	2	Server will receive video feeds from camera and PC.
	3	Server will start transcoding video feeds and broadcast .
	4	Server will store streaming lecture locally for playback purposes.
	5	The use case ends when the lecturer click on stop streaming button on the application.

Table 2.2.3 1 Use case scenario - Start streaming lecture

Use case ID	UC_2	
Use case name	Watch live stream	
Goal in context	Broadcasting lecture	
Pre-condition	Student should be logged in	
Post-condition	None	
Primary actor	Student	
Secondary actor	None	
Main flow	**Step**	**Action**
	1	The use case starts when Student click on the link shared to watch the video.
	2	Server will verify the validity of the connection
	3	Server will start transcoding video feeds and broadcast.
	4	The use case ends when student click on stop button/ exit from the browser tab

Table 2.2.3 2 Use case scenario - Watch live stream

Use case ID	UC_3
Use case name	Playback video
Goal in context	Broadcasting a recorded lecture
Pre-condition	Student should be logged in
Post-condition	None

Primary actor	Student	
Secondary actor	None	
Main flow	**Step**	**Action**
	1	The use case starts when Student click on the thumbnail of a previous lecture in the application
	2	Server will verify the validity of the connection
	3	Server will start transcoding video feeds and broadcast.
	4	The use case ends when student click on stop playback button or exit the browser

Table 2.2.3 3 Use case scenario - Playback video

Use case ID	UC_4	
Use case name	Ask a question via webcam	
Goal in context	Provide live Q&A functionality to student	
Pre-condition	Student should be logged in Student should have a connected webcam and a microphone	
Post-condition	None	
Primary actor	Student	
Secondary actor	None	
Main flow	**Step**	**Action**
	1	The use case starts when Lecturer accepts students request to ask a question.
	2	Application ask for permission to use the webcam and microphone from the student.
	3	Student provides the permission.
	4	Server receive feed from students PC and send it to lecturer .
	5	Application display the students video feed to lecturer in real-time.
	6	Use case ends when student ends the questioning session.

Table 2.2.3 4 Use case scenario - Ask a question via webcam

2.3 User Characteristics

Users of the Real-time e-Learning platform are University students and lecturer's and administration panel who has the authorization to access the application. Users will require to have simple computer skills or should be familiar with smart devices.

Application will be based on English and users should have knowledge on English language to understand the instructions.

2.4 Constraints

- There will be a wide range of devices and browsers that this application will support but with few exceptions.
- Users require internet access and for better real-time experience users should have a high speed internet connection to reduce network delay.
- Since the application servers require large processing power, a high performance computer is required to deploy the application. A cloud computer is preferred.
- Since application stores videos of every lecture stream, servers should have a large storage.
- For the users to fully experience the application features, user devices should have a webcam and a microphone.

2.5 Assumptions and Dependencies

- Users have the authorization to access the web application.
- Users have a high speed internet connection.
- User devices have the capability of a webcam and a microphone.

2.6 Apportioning of Requirements

Main requirements mentioned above in the document will not be changed during the development phase. However, the technologies and some approaches will be changed in order to make the application better.

Requirements mention in chapter 3 are desirable requirements that are likely to be change in the future with the progress of implementation of the application. And these optional requirements will be implemented and released as several versions of the product.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Transcoding module is a pre-configured server to provide the required services for the other modules of the application. Interface shown in *Figure 2.1.2.1* will be available for administrative access. It shows details of each streaming setup configured to the application.

3.1.2 Hardware Interfaces

- Tracking camera to acquire lecture tracking video feed.
- Laptop / Desktop / Tablet of the lecturer
- Microphone

3.1.3 Software Interfaces

- Kurento media server 6.9.0 for transcoding purposes
- MySQL 5.7 as the database to store necessary information related to the service.

3.1.4 Communication Interfaces

Video feeds from the tracking camera and lecture's PC will be sent to transcoding server via local network connection or via internet. Using internet will require an internet connection with high bandwidth.

In order for users to watch lecture live feeds or playbacks, they also require an internet connection with high bandwidth.

3.2 Classes/Objects

There will not be any classes to be designed to implement this component of the system.

3.3 Performance Requirements

Transcoding is a very resource consuming process. It requires high processing power and high memory as well. This particular application, requires to transcode 2 videos per lecture for approximately 50 lectures simultaneously. And these should be delivered to approximately 100 online viewers per lecture.

Each and every lecture which were live streamed has to be stored to support to playback. Therefore, this application should have a larger storage space to store those videos. Required storage is approximately 10TB per month.

3.4 Software System Attributes

3.4.1 Reliability

Reliability of this component depends on the time taken to deliver a live lecture stream to the viewers. These streams should be very close to real-time which means with very minimum latency. In order to achieve that, this application has to be deployed in a high

performance server. And also we have introduced adaptive bitrate feature as well to the application.

Synchronization between video streams and the audio stream is also take part in the reliability of this application. Out of sync videos are very distracting for this kind of application.

3.4.2 Availability

This service will be generally available 99% of the time except for scheduled downtimes for the maintenance purposes.

3.4.3 Security

Communication between transcoding server and the system application will be encrypted to restricting the access to the malicious parties. All streams will require an access code which will be emailed to students of the specific class in order to access a live stream.

3.4.4 Maintainability

As system handles large video files, it collects large amount of data over time. As a result, these data should back up to redundant servers frequently. Transcoding server will also receive frequent updates improving the performance as a result of further development and research of this project.

3.5 Other Requirements

A Real-time captioning functionality will be implemented to provide live translates of the lectures and will be released with later versions of the module.

4 Supporting Information

4.1 References

[1] Yueshi Shen, "*Live Video Transmuxing/Transcoding:FFmpeg vs TwitchTranscoder Part I* " https://blog.twitch.tv/live-video-transmuxing-transcoding-ffmpeg-vs-twitchtranscoder-part-i-489c1c125f28. [Accessed: 2019-05-13]

[2] Kurento, "Kurento Media Server Development Documentation" https://doc-kurento.readthedocs.io/en/6.10.0/#dev-docs [Accessed: 2019-05-13]