REAL-TIME E-LEARNING PLATFORM

Final Report for Lecture Playback Module

Project ID: 19-079

P.G.R.S.H. Gamlath
IT16124936

Supervisor – Dr. Malitha Wijesundara

Bachelor of Science (Hons) in Information Technology Specialized in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

16th September 2019

DECLARATION

I declare that this is my own work and this dissertation 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 my 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. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

Signature: Date: 16^{th} of September 2019

Signature of the Supervisor: Date: 16^{th} of September 2019

ABSTRACT

Real-time e-learning platform is a WebRTC based live teaching platform which facilitates the e-learning requirements of universities and other institutes. This solution includes lecture live streaming, lecture playback, a vector based interactive whiteboard, chatting and file sharing module and a real-time lecture movement tracking module using a PTZ camera. The system is capable of streaming two simultaneous streams of a 1080p camera and a 720p screen capture seamlessly using a network connection with 256KB/s bandwidth. Lecture playback supports camera video playback in 1080p, 720p and 360p and screen playback in 720p and 360p. This player is integrated with MPEG-DASH technology to support adaptive bitrate streaming. Clients can use this player with an internet connection with minimum of 128 KB/s bandwidth. With these features, this solution is a very cost effective and reliable product compared to existing competitors in the market.

ACKNOWLEDGEMENT

First, I would like to express my gratitude to supervisor Dr. Malitha Wijesundara and co-supervisor Mr. Pramadi Athapaththu for the guidance and mentoring given throughout this period to make this research a success. Also I would like to thank senior lecturer and lecture in-charge for Comprehensive Design and Analysis Projects module, Mr. Jayantha Amararachchi for the valuable assistance given to complete this project.

At last but not least I am also thankful to all my family members and colleagues who have been always helping and encouraging me throughout the research period. I have no valuable words to express my thanks, but my heart is still full of the favors received from every person.

TABLE OF CONTENT

Declaration ... i
Abstract ... ii
Acknowledgement ... iii
Table of Content ... iv
List of Figures ... v
List of Abbreviations ... v
1. Introduction ... 1
1.1. Background Literature ... 1
1.2. Research Gap ... 2
1.3. Research problem ... 2
1.4. Research objectives ... 2
2. Methodology ... 4
2.1. Commercialization aspects ... 6
2.1.1. Cloud based revenue model ... 6
2.1.2. On premises revenue model ... 6
3. Testing and Results ... 12
4. Conclusion ... 14
References ... 16

LIST OF FIGURES

Figure 2.1: High Level Architectural Diagram 4
Figure 2.2: Video Conversion Process 5
Figure 3.1: Encoding Time Comparison 12
Figure 3.2: File Size Comparison 13

LIST OF ABBREVIATIONS

WebRTC	Web Real-Time Communications
RTP	Real-time Transport Protocol
PTZ	Pan Tilt Zoom
KB	Kilo Byte
CPU	Central Processing Unit
MPEG	Moving Picture Experts Group
DASH	Dynamic Adaptive Streaming over HTTP

1. INTRODUCTION

Real-time e-Learning platform is a live lecture streaming web application which is developed focusing on Universities and other educational institutes. Lecturers can stream lectures from or out of the institute premises remotely while students can join relevant live streams from anywhere. This report will describe the live streaming component of the application. Lectures streamed from institute premises will be equipped with a PTZ camera to get a third person perspective camera feed of the lecturer. Lecturers can use their web camera instead of PTZ camera if they are streaming remotely outside of the institute premises. Apart from the camera feed, Lecturer's Computer screen is captured and streamed. Lecturer can select between which screen, which application to share. Also audio feed from a mic attached to the lecturer's PC is broadcasted. The application is capable of handling multiple simultaneous lecture sessions without any interruptions to each other. This component is also responsible for recording live streams to facilitate the lecture playback where students are given the capability to rewind and playback already streamed lectures for future reference.

1.1. Background Literature

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 issue. 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. [1][2]

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 a webcam in real-time and also can use a virtual whiteboard that is provided with the system to describe the question.

1.2. Research Gap

When we consider about existing competitors of this field, they do not support adaptive bitrate streaming for recorded lecture playback. Some competitors allow users to change the quality of the video while others providing a fixed video quality. In both scenarios, a user with either a low bandwidth or varying bandwidth internet connection will experience pauses in the play due to buffering. Also the current competitors do not provide a comprehensive analysis of usage data in order to have a self-assessment for students as well as lecturers to assess students and their own lectures.

1.3. Research problem

The most common teaching and learning practice adopted by many enterprises has always been a classroom with one or more instructors and learners meeting physically and in real-time. But in this teaching model, there are several drawbacks which will be addressed as problems within this research.

A classroom based learning experience means the class schedule is predetermined and not subject to change. Students must shape their personal schedules around school instead of the other way around. If plans unexpectedly change or an emergency comes up, the student cannot adjust the class schedule to turn in the work at a different time. This is one of the main problems that this research will find solutions for.

Content non-reusability is another problem that can be found in classroom learning. Memorizing or writing all the necessary content while listening to a lecture is difficult. Therefore, students might miss many important points that the lecturer is pointing out during a lecture.

1.4. Research objectives

1. Minimize buffering pauses in playback

   A main objective of the player is to have minimum buffering pauses if not nil. The player should automatically change the source to match the source bitrate with the current network bandwidth.

2. Keep the video and data feeds in-sync

   Video inputs for the player are independent feeds. The player should keep the feeds in-sync all the time to deliver the best user experience. Data feeds recorded from the whiteboard and chat window during the live lecture session should also played in sync with the videos.

3. Data collection and analysis

   User interactions with player should be captured relative to the position of the lecture playback. With these data, a detailed report has to be generated for each student, each lecture and each module.

2. METHODOLOGY

In order to develop this real-time application which is usable from multiple devices as laptops, tablets and smartphones we have chosen WebRTC technology. WebRTC is an open source project developed by google to make web browsers support peer-to-peer communications. On top of that, to implement our solution, we used Janus as the media server. Janus is a general purpose WebRTC server which is capable of implementing WebRTC media communications with browsers [3]. Backend application server is developed with Node.JS while frontend application is implemented using React.JS. MPEG-DASH technology is adhered in development of the playback player in order to support adaptive bitrate streaming [4] [5].

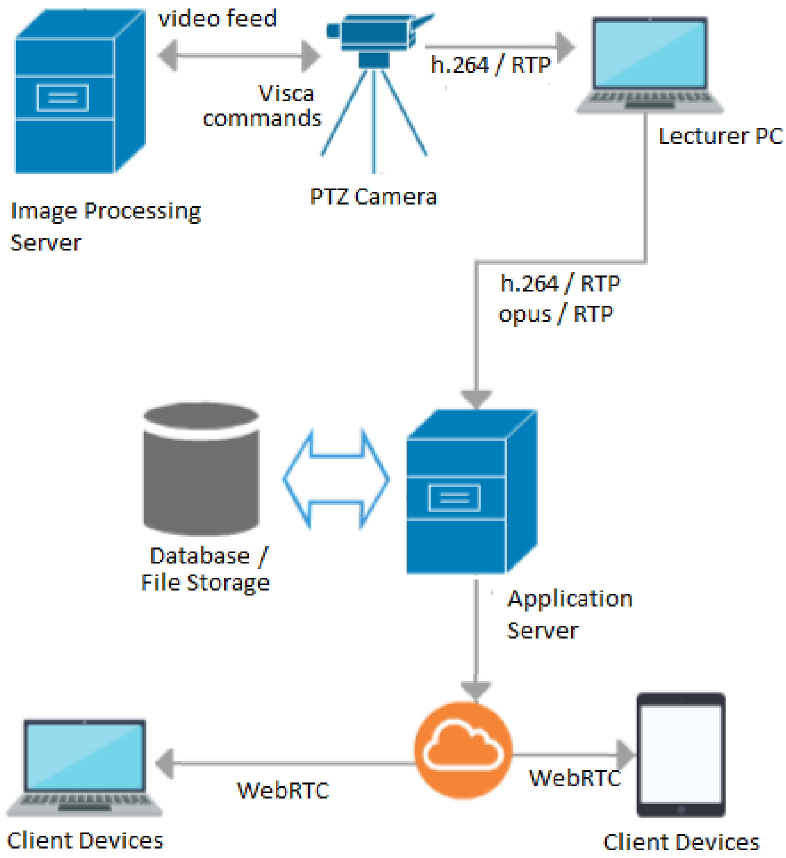

Figure 2.1: High Level Architectural Diagram

Main purpose of the lecture playback component is to provide a complete and uninterrupted playback experience of a past lecture session. Additionally, students, lecturers and management should be able to view a comprehensive analysis of the usage data from the player.

In order to be played in a video player with ability to seek, input videos should possess all the necessary key frames. During the streaming process some key frames might not get created in the recorded stream. Therefore, the post processed video streams are passed through another process to recreate key frames at regular intervals [6].

The playback player is developed to support adaptive bitrate streaming. User of the playback player can explicitly select the video quality in which user wants the videos to be played. In order to cater this requirement, the key frame recreated video streams are automatically converted into 3 different video qualities with different resolutions and different bitrates. (1080p, 720p, 360p). After the conversion, each converted video and the original audio will be spitted into 2 second segments which are indexed in order. These segments are then fed into the player for the playback. If the user opted for automatic video quality, the player will select the suitable video stream according to the current network bandwidth of the user.

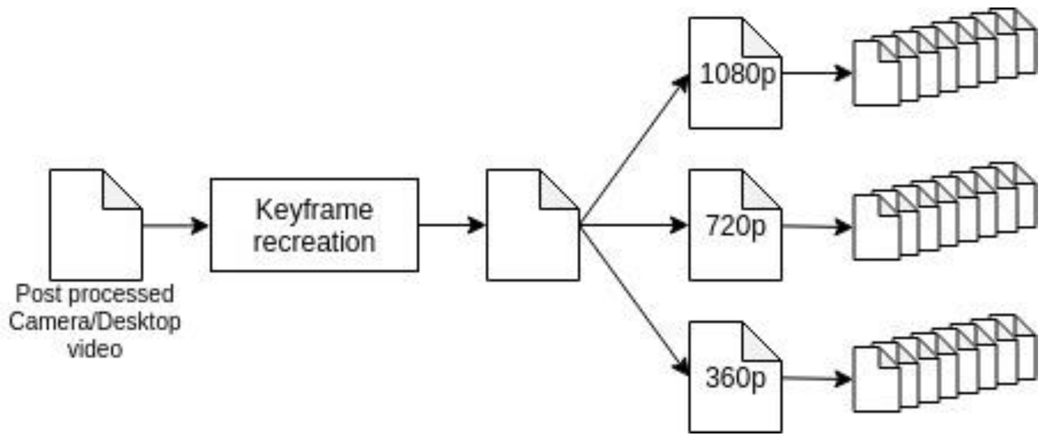

Figure 2.2: Video Conversion Process

In the player, two videos are played in sync along with the audio. Chat data and whiteboard data related to the playing lecture session will be retrieved from the server and displayed at the relevant timestamp giving the original experience of the live play. User can annotate different sections of the video for future references.

Player will keep track of the sections which are played by the user and get sent to the server. These data combined with the user information and lecture session information, will be used generate statistics. With these statistics, lecturers can have a true feedback on each lecture session they conducted. Management can have an assessment on the lecture sessions based on the students’ interactions with the videos. Students can have a self-evaluation based on own interactions with each lecture session.

2.1. Commercialization Aspects

Key stakeholders of this system are:

- University / Educational Institutes / Schools
- Lecturers / Teachers
- Students

Our main goal of the product in the aspect of commercialization is reduce the cost for the usage of the product than the competitors in the current market.

Real-time e-learning platform is commercialized based on two business revenue models.

- Cloud based revenue model
- On premises revenue model

2.1.1. Cloud based revenue model

This revenue model is used if the client agrees to install their application setup in a amazon cloud environment. Amazon will charge for monthly resource usage. Clients are charged with an additional service charge which will be a multiple of the entire usage bill from cloud service.

Total usage bill = Amazon cloud bill x billing factor

2.1.2. On premises revenue model

If the client does not want to use a 3rd party service to host their application and wants to setup the system on premises, this revenue model will be used. Here client will be billed monthly based on the user accounts created in the system.

2.2. Implementation

From the starting point of the project, we followed a waterfall model of development until the end of the project. Unit testing was done to each individual component developed while integration testing also carried out each time n feature integration was carried out to the system. We maintained a development environment which was our local devices which we use to develop as well as a AWS linux server as the production environment. Each component was developed and tested in both of these environments for expected outcomes.

During the implementation period, we used Github for the version controlling of our application. Two Git repositories were maintained as for the frontend developments and backend developments. Clearly identified features and independent development of them in different branches allowed the features to be integrated to the main system successfully without major merge conflicts.

After a successful development stage, following are the outcomes of above implementations in the UI point of view.

Once the encoding process is completed for a recorded lecture, it will appear in the *Recordings* page accessed using side menu.

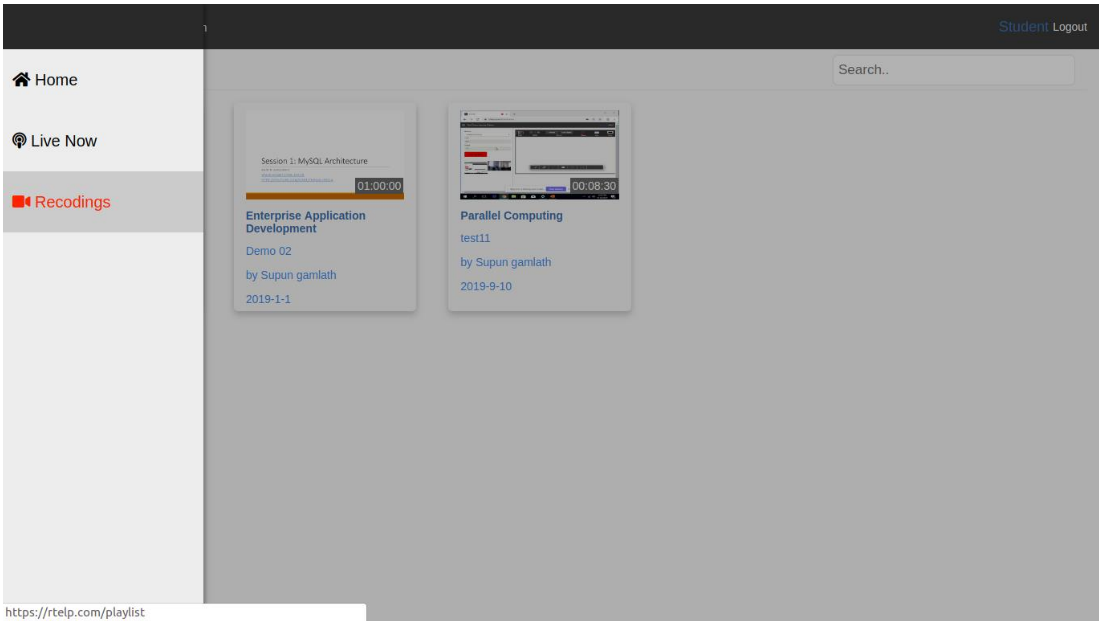

Figure 2.2.1 Recordings Menu

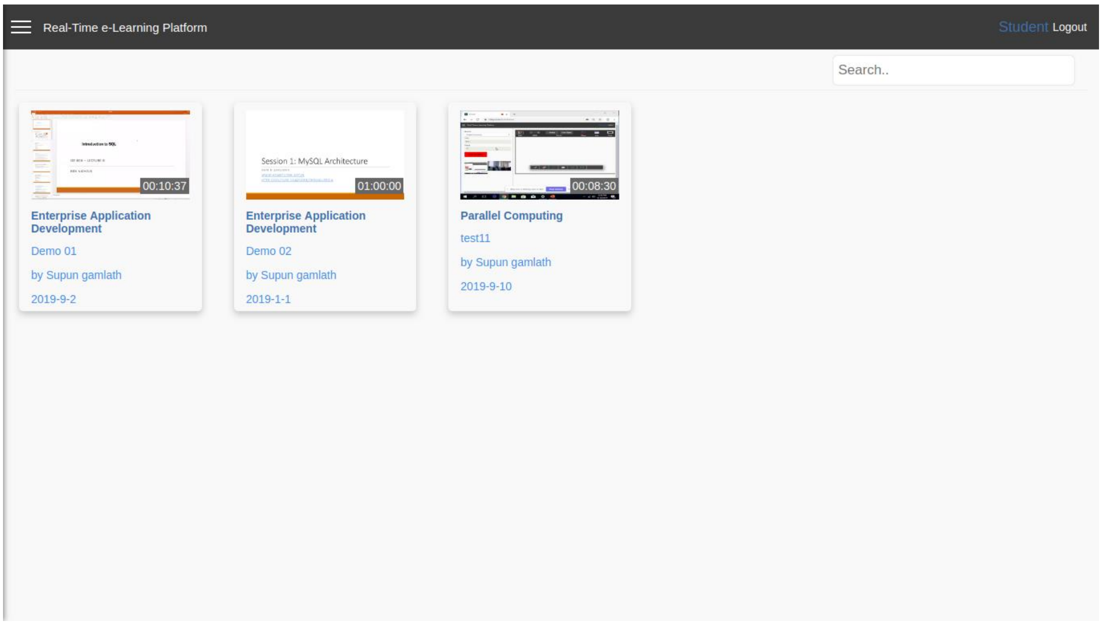

Figure 2.2.2 Recorded Lectures

Student can search for lectures using module name or lecture title. Once a lecture is selected, student will be redirected to the playback player.

Player will automatically start to play the lecture in the default 50:50 layout as shown in the figure 2.2.3.

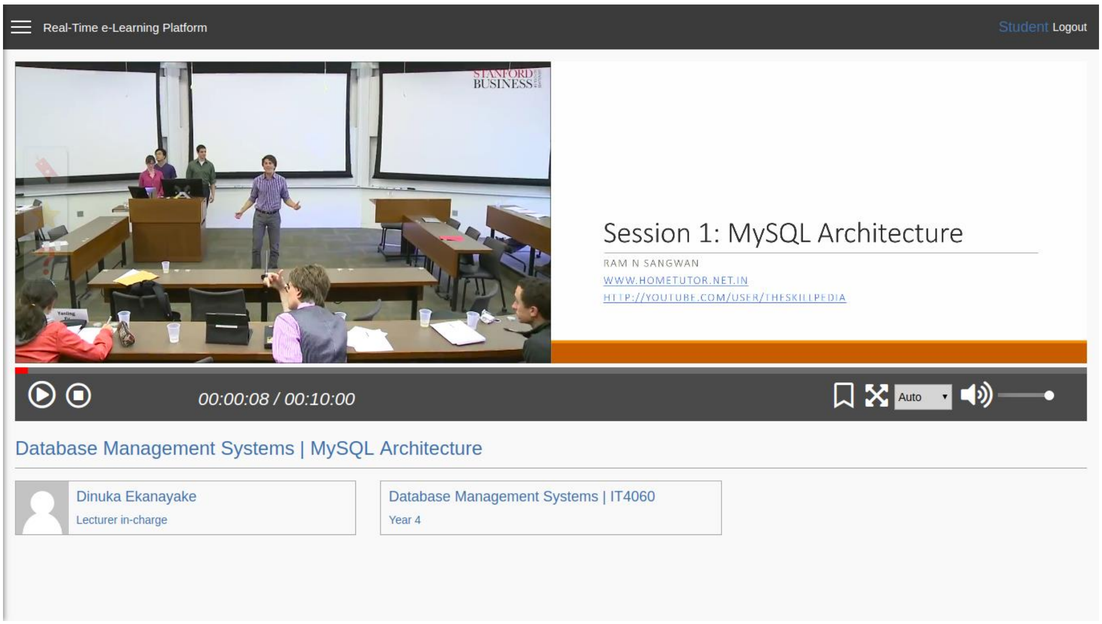

Figure 2.2.3 Playback player in default 50:50 layout

Student do have all the basic functionalities of a video player including *play, pause, stop, seek, volume up, volume down* and *mute.* Apart from the default 50:50 layout, there is a full screen option that can be enabled by double clicking on a video of choice or by clicking the *full screen* button in the controls bar.

Figure 2.2.4 and Figure 2.2.5 shows the full screen layout with 2 videos interchanged. While one video is in full screen mode, other video is in a floating position. This floating video can be resized and moved around the screen.

To exit the full screen mode, student can either double click on the full screen video or by clicking the *full screen* button in the controls bar.

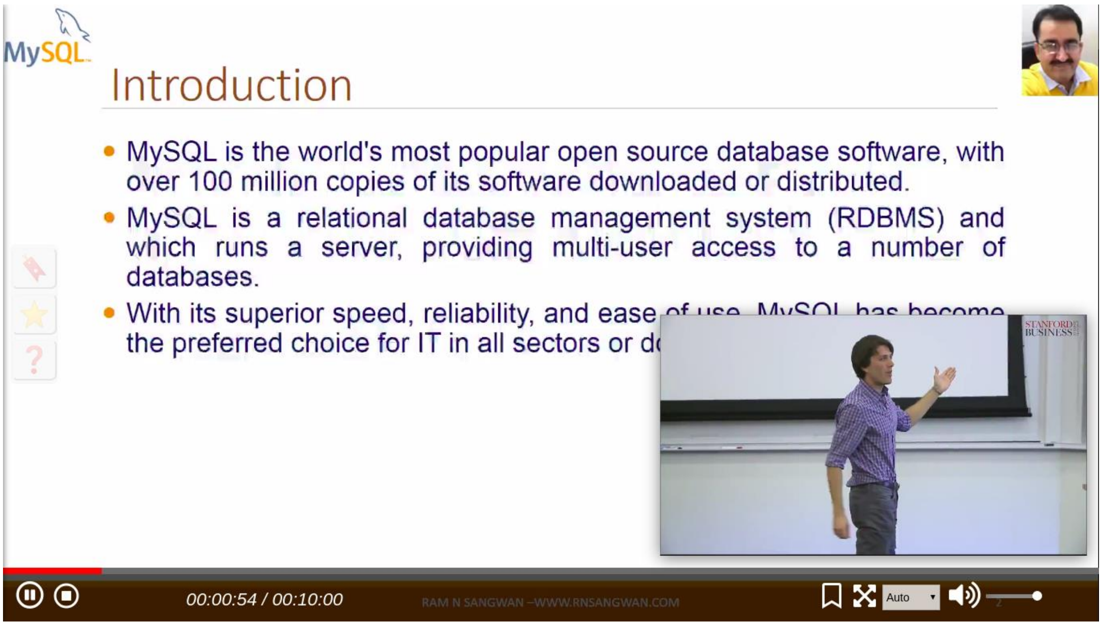

Figure 2.2.4 Playback player in full screen layout

Figure 2.2.5 Playback player in full screen layout

There is an option to change the quality of the video as shown in Figure 2.2.6. Available options are *High*, *Medium* and *Low.* The default value is set to *Auto*, so that the player will adapt to the available bandwidth of the internet connection and change the quality of the video accordingly.

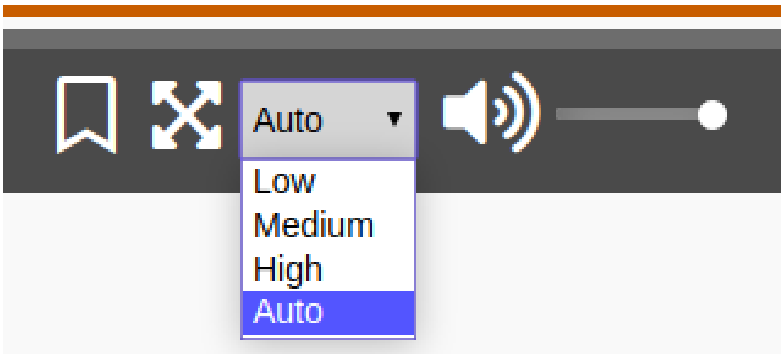

Figure 2.2.6 Quality Selector

Students can annotate the video with 3 different types of annotations. *Important*, *Unclear* and *Bookmark.* This can be done using a floating toolbar on top of the video that is made transparent to make less distracting to the viewer. It becomes opaque when the cursor is moved on top of it. This is shown in Figure 2.2.7.

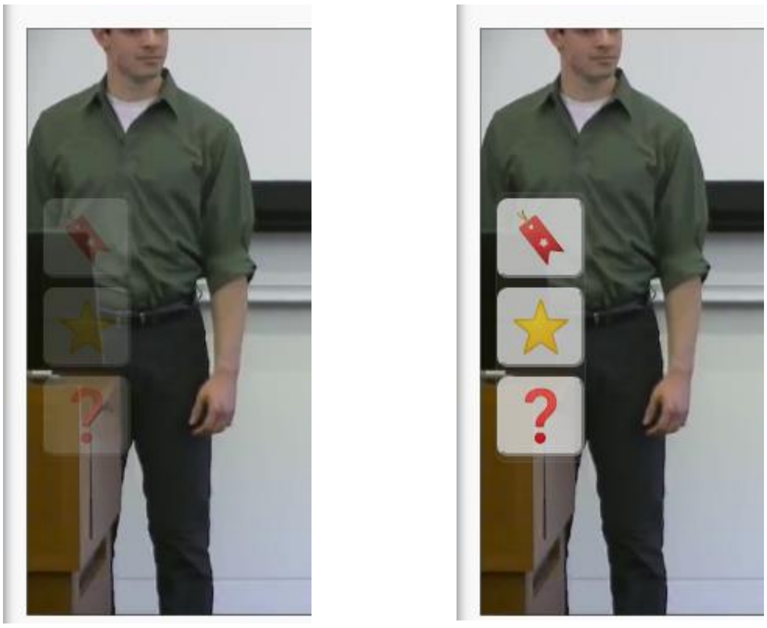

Figure 2.2.7 Annotation Buttons

These annotations are shown in a separate bar aligned with timeline. This bar can be toggled using the *Annotations* button in controls bar. This bar is shown in the figure 2.2.8.

Figure 2.2.8 Annotations bar

These annotations can be used to seek the video by clicking on any of it. It’s timestamp and comment for the bookmark is also displayed as a tooltip on mouse hover. These annotations can be deleted by right clicking.

3. RESULTS AND DISCUSSION

Encoding process was tested on a system with an Intel Core i5 CPU with 4 x 2.50GHz cores, 8GiB memory running 64-bit Ubuntu 18.04.2 LTS. Testing was done using 4 different input files with duration of 10 minutes. Each file was encoded to 1080p, 720p and 360p.

Time taken for each encoding process was compared with the input video's duration. All the encodings took less time than the original video duration. Figure ### shows the comparison with encoding times as a percentage of the original video duration. 1080p encoding time is around 60% of the original video duration and 720p encoding time is around 33% of the original video duration. 360p video takes only around 19% of the original video duration.

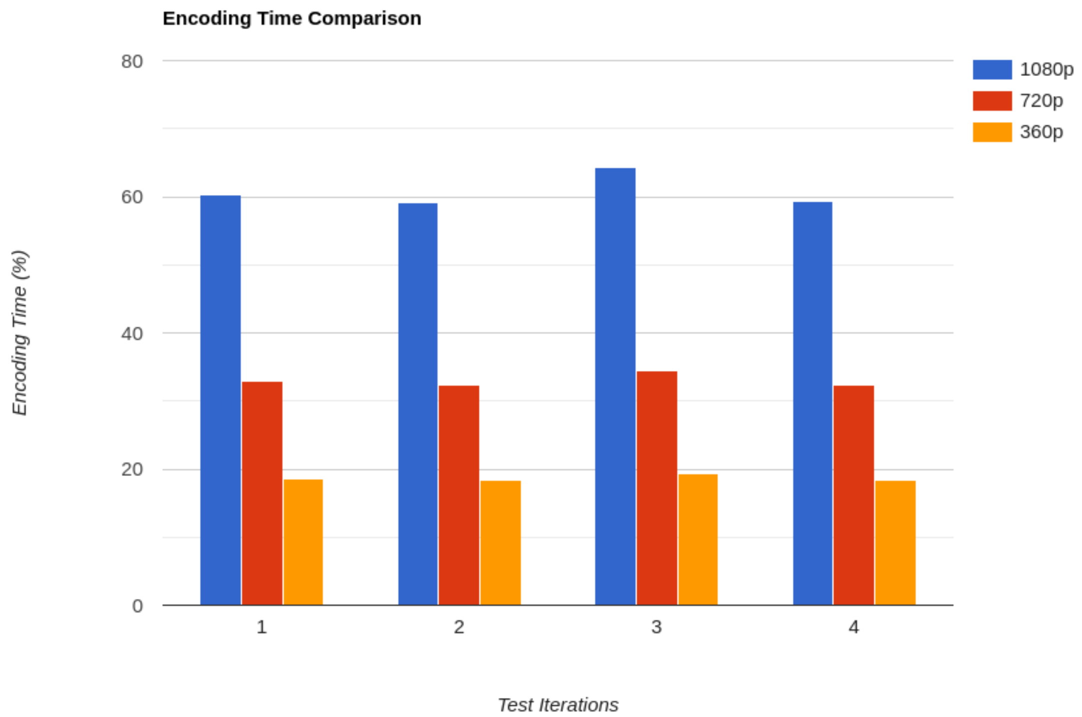

Figure 2.2.1: Encoding Time Comparison

The other aspect tested was the final file size of the encoded videos relative to the original file size. Figure 3.1 shows the comparison with each encoded file size as a percentage of the original file. 1080p output size is around 20% of the original file size and 720p output size is around 4% of the original file size. 360p video takes only around 1% of the original file size. However, this encoding process is very CPU intensive.

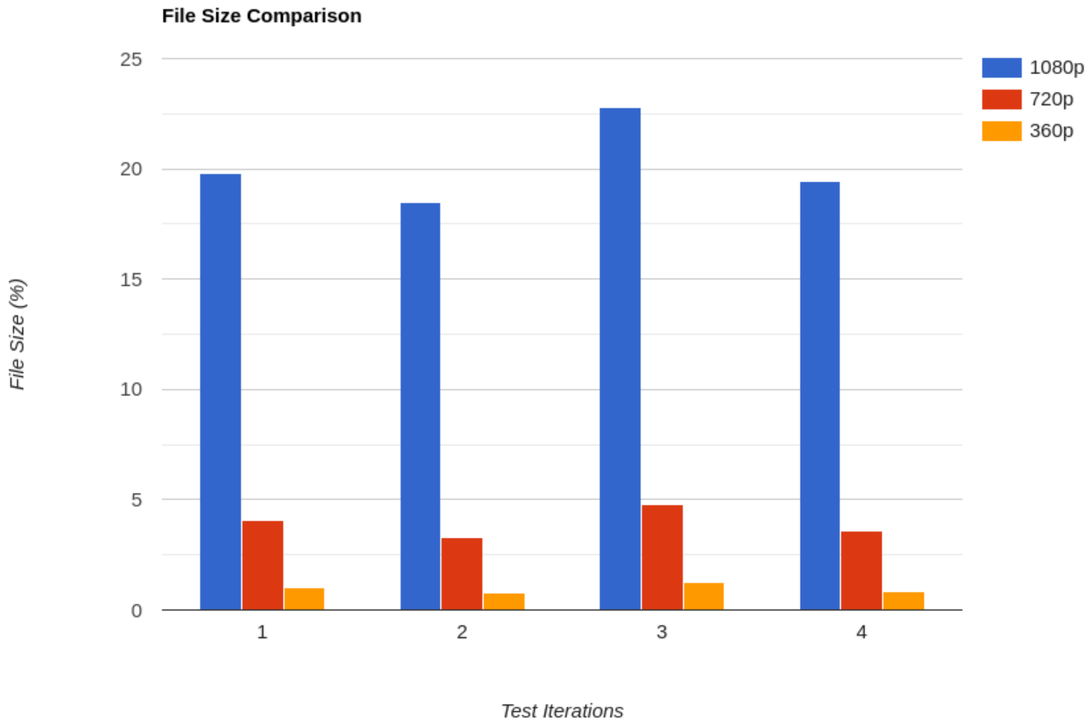

Figure 2.2.2: File Size Comparison

This reduction of file size is great for the users as they can play the videos with a low bandwidth internet connection.

4. CONCLUSION

Implementing an actual real-time e-learning platform is not possible since anyway along the pipe line, we have to do some computational functions in order to deliver media across different clients. But we were able to achieve a near real-time solution for our problem reducing the latency to fractions of a second. All these results were possible because of using WebRTC as the base for the solution. For establishing near real-time remote communications, WebRTC is highly recommended than other technologies out there.

With video quality set to 360p, videos can be played without any buffering pauses given an internet connection with a minimum bandwidth of 128KB/s. With the quality set to *Auto*, videos will be played in highest quality available. If the prevailing bandwidth is not sufficient for the bitrate of the highest quality stream, videos will pause until player select the suitable stream. With a relatively consistent network bandwidth, videos will be played smoothly without any pauses in the middle. Both video streams are played in-sync with each other. If the videos get off-sync, player will automatically correct the sync by bringing back both videos to lower timestamp.

Each time a lecture is played, a list of time segments is stored in the server. These segments represent the parts of the lecture the user actually watched. The list has a reference to the original lecture session, student logged in, time the student logged in. Another list containing annotations added by the student for that particular lecture is also stored in the server with reference to student logged in and the original lecture session. Using these data, a detailed usage report can be generated.

Looking ahead, next steps for this solution will be introducing adaptive bitrate capability during a live stream. This will help the clients to access live streams at any condition of their network bandwidths.

Using automated PTZ camera improves student experience on e-learning platforms. yolo3 can continue tracking even in low light conditions with high accuracy. SORT is also implemented using yolo3 and the combination works well in a GPU with a frame rate around 10. PID controller smoothly drives the PTZ camera to obtain a good video feed of the lecturer.

Looking ahead, even though yolo3 works well in this scenario it is a general purpose object detector with a variety of 80 classes. Well optimized and specific purpose detection model can be trained using the previously recorded videos of the lecturer sessions by annotating frames as the image dataset to the model.

Yolo3 and SORT requires a GPU to get higher FPS value. With the advancement of deep learning and the computing devices, if this algorithm can be more optimized in future versions, so that it can fit into a device like a raspberry pie, it will reduce the overall cost by tremendous amount.

REFERENCES

[1] Chris Curren, “*Strategies for e-Learning in Universities*”,CSHE.7.04,2004

[2] H. M. Truong, "Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities," Computers in Human Behavior, 2015

[3] Meetecho s.r.l, “Meetecho Janus Recordings”, https://janus.conf.meetecho.com/docs/recordings.html [Accessed: 2019-08-06]

[4] "What is Adaptive Bitrate Streaming? - Encoding.com Help", Encoding.com Help, 2019. [Online]. Available: https://help.encoding.com/knowledge-base/article/what-is-adaptive-bitrate-streaming/. [Accessed: 03- Aug- 2019].

[5] "DASH Streaming | Encoding.com", Encoding.com, 2019. [Online]. Available: https://www.encoding.com/mpeg-dash/. [Accessed: 03- Aug- 2019].

[6] "ffmpeg Documentation", Ffmpeg.org, 2019. [Online]. Available: https://ffmpeg.org/ffmpeg-all.html#Advanced-Video-options. [Accessed: 03- Aug-2019].