REAL TIME E-LEARNING PLATFORM

Final Report for Student Questioning with Whiteboard Module

19-079

Supervised By -:

Dr. Malitha Wijesundara

Authors -:

IT16132306 H.M.S.S. Herath

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

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

September 2019

DECLARATION

I declare that this is my 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.

Name	Student ID	Signature	Date
H.M.S.S.Herath	IT16132306		

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

Signature of Supervisor Date

ABSTRACT

e-Learning is simply meaning using electronic devices and technologies for educational purposes. With the growing use of these technologies on universities suggest the future of the classroom will rely heavily on e-learning. RTELP is a product for e-Learning that makes student university life more reliable and easy. Like technology ease the life of people, this product will ease the student academic life and helps students in different ways.

This solution includes a platform for e-learning by providing virtual classroom for the students with real classroom facilities. It will cover a normal classroom scenario by streaming the lectures so that students can interact with the lecture from anywhere around the world. From this system we are suggesting more interaction among publisher and the connected users. With interactive whiteboard, polling system, offline playback system users will have a special interaction to the real time streaming as well as offline playback. Using of this product is simple as the user interfaces are simply designed and usability of the product is improved.

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. Live streaming component is very less CPU intensive and it use around 14% of the CPU for streaming 10 simultaneous sessions with 10 listeners per each on a AWS t2.micro instance with 1 vCPU 2.5 GHz, Intel Xeon Family, 1 GiB memory. Because of that, this solution is a very cost effective product compared to existing competitors in the market.

ACKNOWLEDGEMENT

It always pleasures to remind the fine people in SLIIT for their sincere supervision and assistance I received to uphold this research and I would like to express my deepest appreciation to all those who guidance me to make this research success.

I would like to express my special thanks of gratitude to our final year research supervisor, Dr. Malitha Wijesundara and co-supervisor, Mr. Pramadhi Athapattu for their able guidance and support throughout the research.

I would also like to extend my gratitude to my research team members A.M.H.B. Athapaththu, W.A. Geeth Sameera and P.G.R.S.H. Gamlath for their cooperation and encouragement in completing this project.

Finally, I thank everyone who support me and encourage me to finish this research successfully.

H.M.S.S. Herath

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Acronym/Abbreviation	**Definition**
SQHW	Student Questioning Handling & Whiteboard

Table 1: Acronym/Abbreviation

1. INTRODUCTION

The main 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.

Student Questions Handling & Whiteboard (SQHW) is one of the major component of the main real time e-Learning system. This module enhances the interaction between lecturer and the student connected via internet by introducing real time questioning facility. Students can ask questions in different ways. Using webcam and the microphone, using chat box type and send the question. There is a whiteboard facility which can be used by the students to ask questions more clearly by drawing or writing and this is a special feature with vector based drawing window. This whiteboard is light weight and the processor usage when using this feature is very low and can be ignore it.

1.1. Research Problem

For a university student attending lectures is a must to understand and study the subject materials. Here the most common teaching model of the world is meeting the instructors and students in real time and learn. In this teaching model there are several drawbacks that has to be concerned and they will be addressed as the research problems. One of them is student has to shape their personal schedules around university as the class based learning schedule not changed according to student personal needs. When he cannot attend due to some reasons, he won't able to cover the lectures again and it would be a huge loss considering from the student side. Even if the student attends the lecture he can't memorizing or write all the necessary content and sometimes may forget things memorized with the time.

Furthermore, there are several problems to be concerned even after the real time e-Learning system developed. One is to create the real classroom scenario considering the student questioning. So it will be useless if there is no way to communicate with the lecturer and the student in real time to ask questions from the lecturer. And the other one is to enhance the interaction between lecturer and student. In real classroom scenario student can use whiteboard or paper to ask questions more clearly by drawing or writing. Those are the identified problems when developing e-Learning system.

1.2. Research Objectives

The developed solution for the identified problems is an e-Learning system to overcome lecture missing problems as well as content non-reusability problems. The overall expectation of the system is to provide better experience as a real class room scenario with many useful functionalities. The system developed by splitting to four components as

- Real time audio/video transcoding component
- Face tracking component
- Student Question Handling & Whiteboard (SQHW) component
- Video player with data analytics component

This document covers the development of the SQHW component. The main objective of this component is to build up a way to handle the real time questioning facility with the ongoing lecture. Student can mainly ask questions using his webcam and microphone through video and audio medias. Rather student can use chat box to interact with lecturer with text and uploading files when necessary.

The developed system is a web based application and SQHW component is focused with three main functions.

1 Controlling the questioning function
2 Whiteboard function
3 Polling function

The functionalities are described below more clearly.

1 Controlling the questioning function

When we consider the procedure to ask a question using system, first the student has to notify the lecturer that he has to ask a question. When student click the button to ask a question, request will be sent to the lecturer showing that the specific student is willing to ask a question. With the confirmation of the request by lecturer, student able to ask the question with his webcam and microphone. If the student wants to ask the question using chat box procedure would be different. Here no need to send a request to the lecturer, instead student can type the question in chat box and send. So the lecturer and other students in the session would be notified if any new messages received and the chat box is a broadcast chat box. Other connected students also could able to see the chat and they also can interact using chat box.

2 White board function

When student ask the question after lecturer granted the permission, whiteboard editable function will be allocated automatically for the student who asking the question. When the student writes on whiteboard it will be broadcasts to the lecturer as well as all the connected students. This whiteboard creates with specific functions that ease the using of whiteboard. Student can use different colors, eraser with different sizes, and the object eraser which means erase whole line you draw. All the line coordinates drawing here will be send to the database for future usage of analytics component. Rather there are special functions to create straight line, circles, ellipse, rectangles. This will easy the usage of whiteboard whenever one's needed to draw circles, lines or rectangles.

Apart from the above functions users can add images to the whiteboard to explain the things more clearly. If there is unclear parts or lecturer like to explain that part by drawing something this feature is very useful. The usage will be discussed later on this document.

3 Polling function

Polling is a significant function that implemented in this product. This will keep the students motivated towards the lecture. Here the lecturers would be able to creates polls to get an idea how students know about the subject matters. Once lecturer creates the poll, students would be able to see the poll inside the chat window. And they have access to answer with their preferred answer. Poll results have plotted in a graphical way to understand the results clearly. Lecturer can see the current voting details while students voting and later results can be shows to the students after closing the poll.

1.3. Background Literature

e-Learning is a rapidly developing industry and it is adopted by both corporates and academic institutes. [1]

As an impact of early stages of e-learning, universities and lecturers are used to communicate with students online, share and provide access to learning materials, external resources etc. As a result of further more use of e-learning, web based video courses are being published and can access over internet. [2]

Many platforms have been implemented on this domain to accomplish above requirements. Those services are implemented focusing a wide range of end users other than e-Learning requirements for universities.

As the awareness of availability of these technologies spread among universities, many universities have taken measures to incorporate these technologies in their teaching-learning process.

One of the strategies is to monitor student’s learning style to provide valuable advices and instructions for students and lecturers to optimize student’s learning process. [3]

Another frequently used strategy is to enhance the lecture hall experience for the students by giving the opportunity to interact with the lecturer by sharing presentations, quizzes, feedbacks, etc... using technologies like Bluetooth and Wi-Fi. [4]

Universities have taken measures to livestream lectures for the accessibility of both on-campus and off-site students. [5]

Most recently there is a growing demand for self-paced e-learning where students can follow the learning materials at their own pace. This is important because 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. The ideal current solution is to provide interactive content using different approaches such as video streams, quizzes, polls, etc...

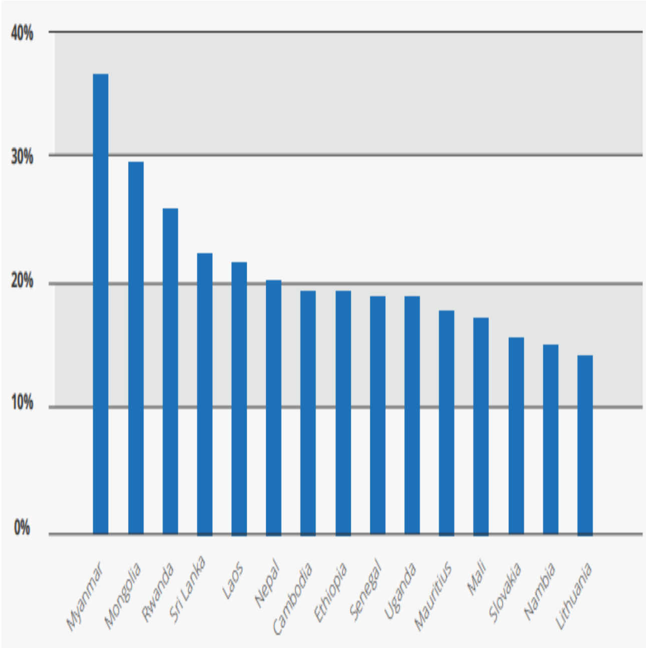

Figure 1.3.1: 2016-2021 Worldwide e-learning 5-year Growth Rates By Country

Figure 1.3.1 shows the predicted growth rate of demand for e-Learning industry up to 2021 according to a research carried out by Docebo.[1] So clearly there is a business value for our proposed solution, which is a fully optimized real-time e-Learning platform that can be adopted by universities and will be profitable as well as beneficial for both the institute and students

1.4. Research Gap

The implemented e-Learning system is a product with more newly introduced features which resolves various problems existing with students relating the studies. When considering the research gap there are systems that trying to overcome those problems, but the solutions target only on solving one or rare scenarios which are not fully helpful for the students. The proposed SQHW component is a very rare solution which will be implemented on RTELP app comparing to the other systems.

Eduscope is a recently deployed real time e-Learning system by the students of SLIIT. It consists with more features compared to the other systems. But it also didn’t cover the fully requirement of the students. Here is the comparison with Eduscope along with our proposed real time e-Learning system.

Systems / **Feature**	**Eduscope**	**Proposed system**
Vector based Whiteboard	No	Yes
Text-based lecturer student interactions	Yes	Yes
Video-based lecturer student interactions	No	Yes
Poll feature	No	Yes

Table 1.4.1: Comparison with Eduscope

2. METHODOLOGY

This system is used by the students of the university to access lecture recordings and live streams. Lecturers of the university interacts with the system during live lecture streaming. Administration panel use this system to monitor activities and behaviors related to students and lectures within the application.

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 [11]. 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 [12] [13]. Socket connection used develop the chat and the whiteboard and plotlyJS is used to plot the poll graphs.

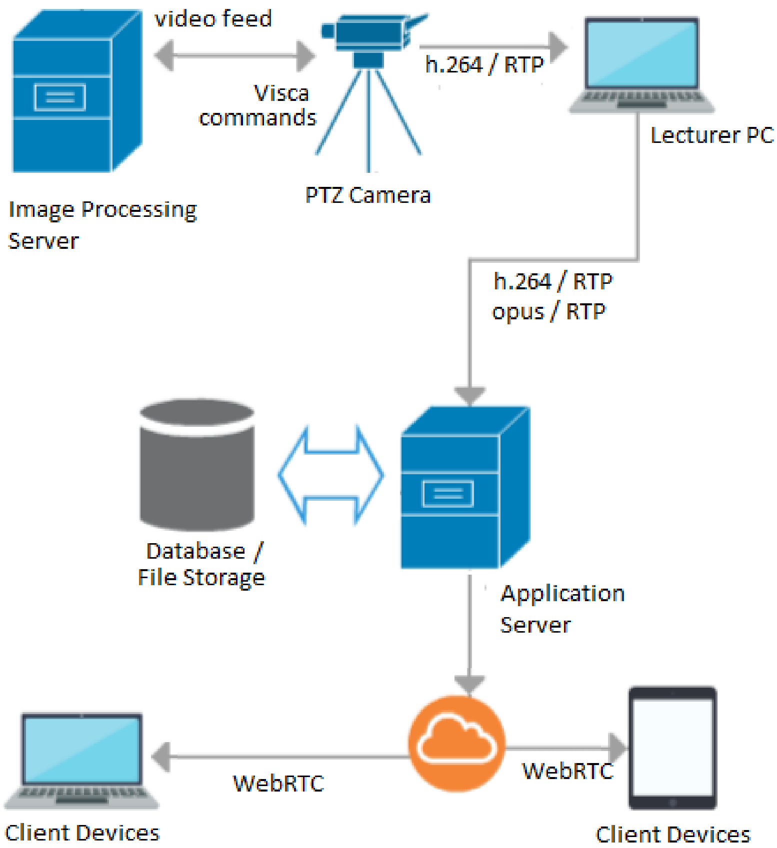

Figure 2.1: High Level Architecture Diagram

Main purpose of this component is to create live interaction with the lecturer and the student. This component developed considering two main parts.

1) Student Questioning Handling with Whiteboard

Whiteboard was implemented using vectors. So only the line coordinates will be transferred among users and created a vector based whiteboard. When a line draws, the coordinates of that drawing goes to the server. Then through the server socket connection that line coordinates are streaming to the all users of the current session. As this whiteboard is a light weight board run with vectors, small process will be used not like video based whiteboards. And also lecturer and the student both can use and draw with this whiteboard as a real time whiteboard.

2) Live chat with poll

The main purpose of this module is to allow chat with each other who connected to the same session. Whether students can ask questions from the lecturer using chat or chat among other users is allowed. Users also can share images or pdf documents among others using the chat window.

Rather than this there is a specific feature that allows lecturer to create polls. So the poll will be displayed to students and they will be able to pick the preferred answer. After the vote is done results will be analysis in the server according to the poll id and lecturer will able to see the results in a graphical way. For the graph drawings plotlyJS is used as a library. Finally, the vote graph can be shared among students. So the students also able to see the results in graphical way.

Both whiteboard and the chat with live poll implemented using socket connection. For publishers and listeners one socket is created each and will be adding the necessary properties for the socket in server. Here the values of the properties of the socket given as pingInterval: 65000ms and pingTimeout: 60000ms on the front end. So it will give a more persistence connection that suitable for the sockets. Drawings in the whiteboard streams with minimum or no delay. Mostly it depends with the internet connection, because socket has no or very low latency.

All these drawings and chats will be saved in a database for the usage of offline playback.

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

2.1.2. Cloud based revenue model

This revenue model is used if the client agrees to install their application setup in an 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

2.1.3. 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.

Current SQHW component UI's are as follows,

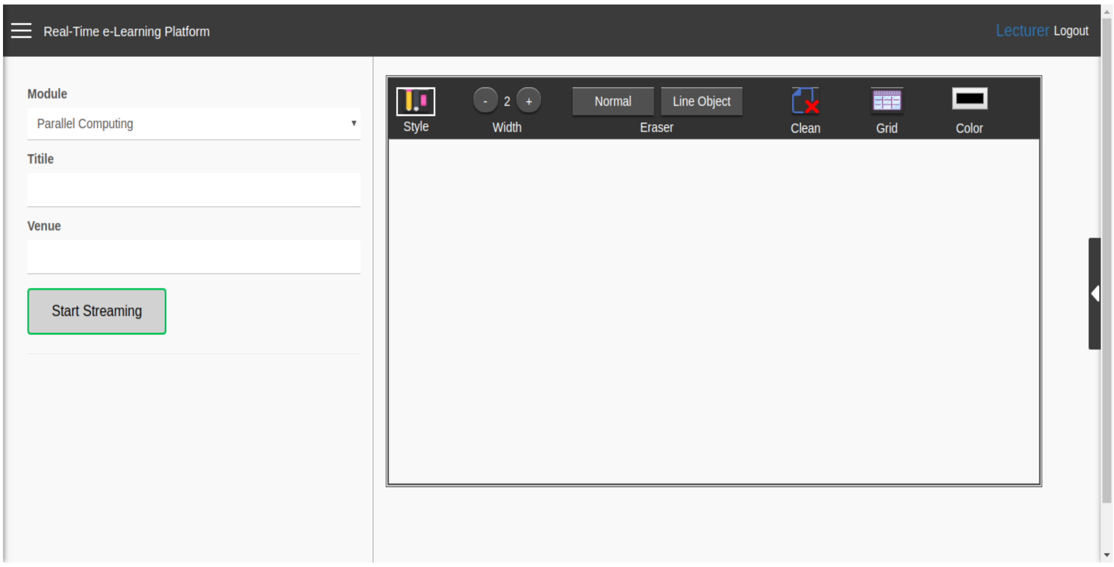

Figure 2.2.1: Lecturer Whiteboard UI

Above Figure 2.2.1 shows the current UI of the lecture's whiteboard. Here the whiteboard is fixed and display in a side. Any drawings by a student will be seen here.

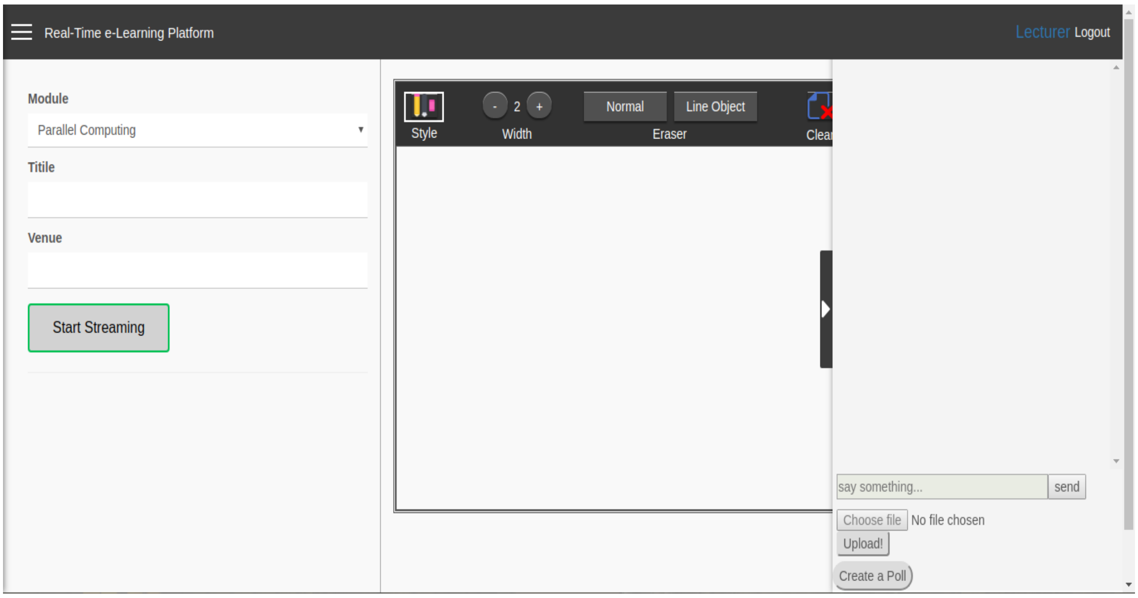

Figure 2.2.2: Lecturer Chat UI

In figure 2.2.2 shows the chat box appearance. For the convenience of the lecturer it has been collapsed and whenever needed it can be expanding like in the figure. If a message comes there was a notification sound as well as blink in the expand button with a different color.

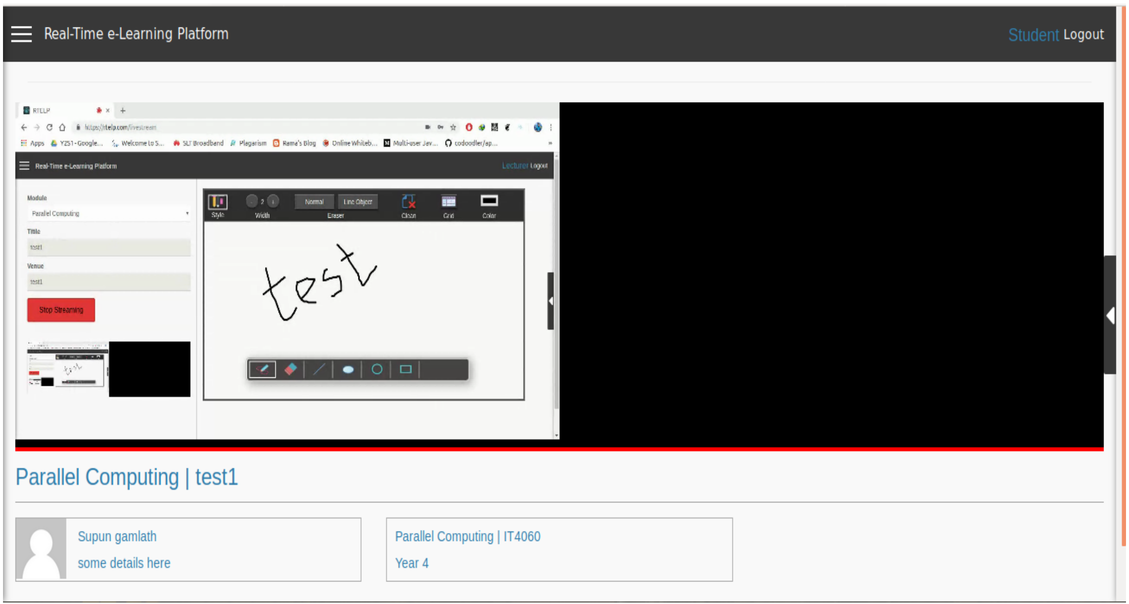

Figure 2.2.3: Student Live UI

Here in figure 2.2.3 shows the screenshot of student side. At the beginning the whiteboard is not showing in the screen. If the student wants to see the whiteboard, he/she has to click the box sign at the beginning of the player controls. Lecturer screen video and the video feed from the camera appears on each sides.

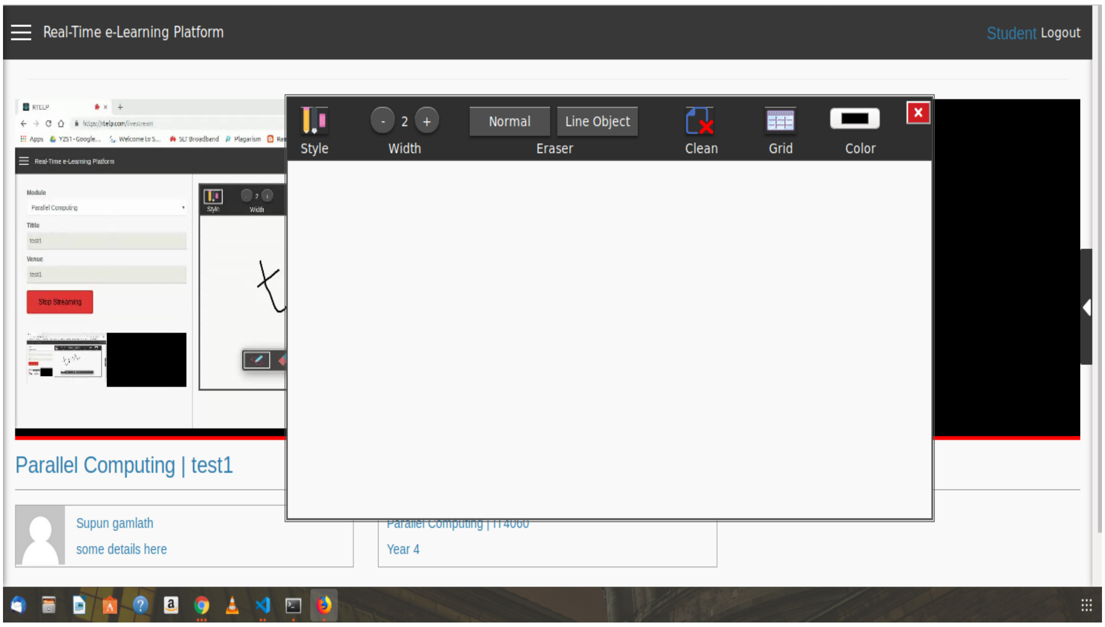

Figure 2.2.4: Student Whiteboard UI

In figure 2.2.4 shows the whiteboard of the student side when it visible. Here in student whiteboard is drag gable. Student can move the whiteboard to anywhere as he/she wish. After using the whiteboard, student can hide it and continue watching the video feeds.

Student chat also same as the lecturer chat in figure 2.2.2.

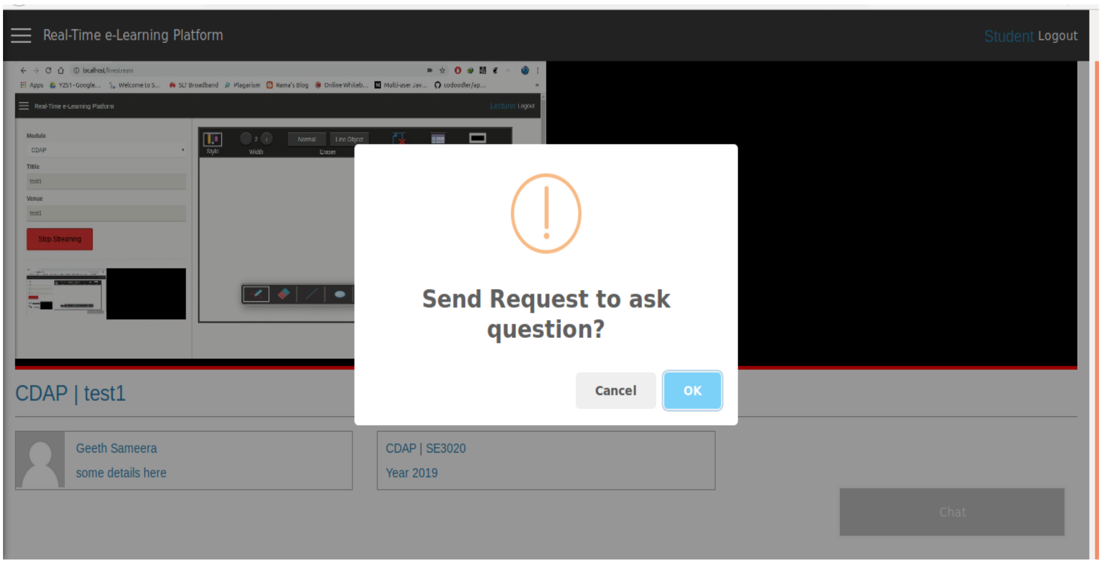

Figure 2.2.5: Raise hand send request

When student wants to ask a question, first he has to send the request using Raise Hand method. When student clicks the button to ask a question confirmation box is popup and with the confirmation of that request will be send to the corresponding lecturer. (figure 2.2.5)

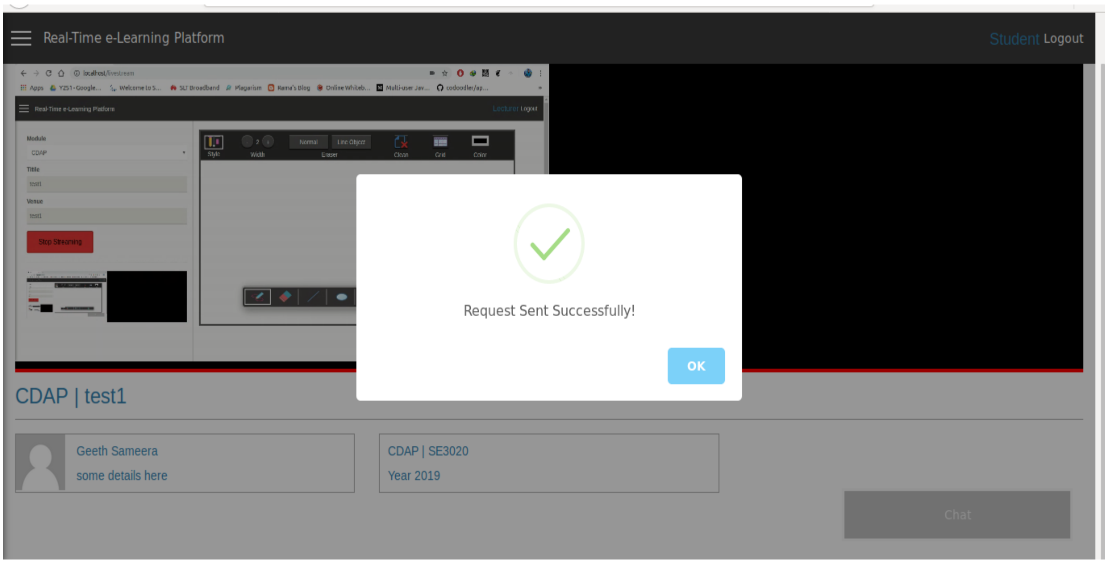

Figure 2.2.6: Raise hand successful

Figure 2.2.6 shows the successful request sent message. If there is something wrong or request doesn't send to the lecturer, sending failed message will be popup.

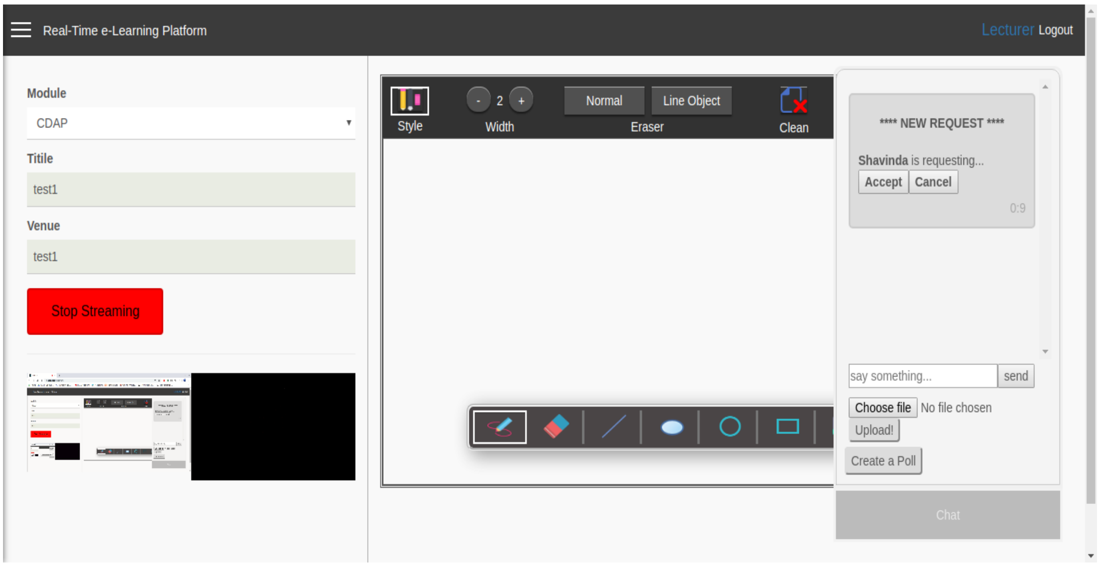

Figure 2.2.7: Raise hand lecturer view

The request from the student to ask a question is appearing on the lecturer side chat box (figure 2.2.7). Lecturer can confirm of reject the request.

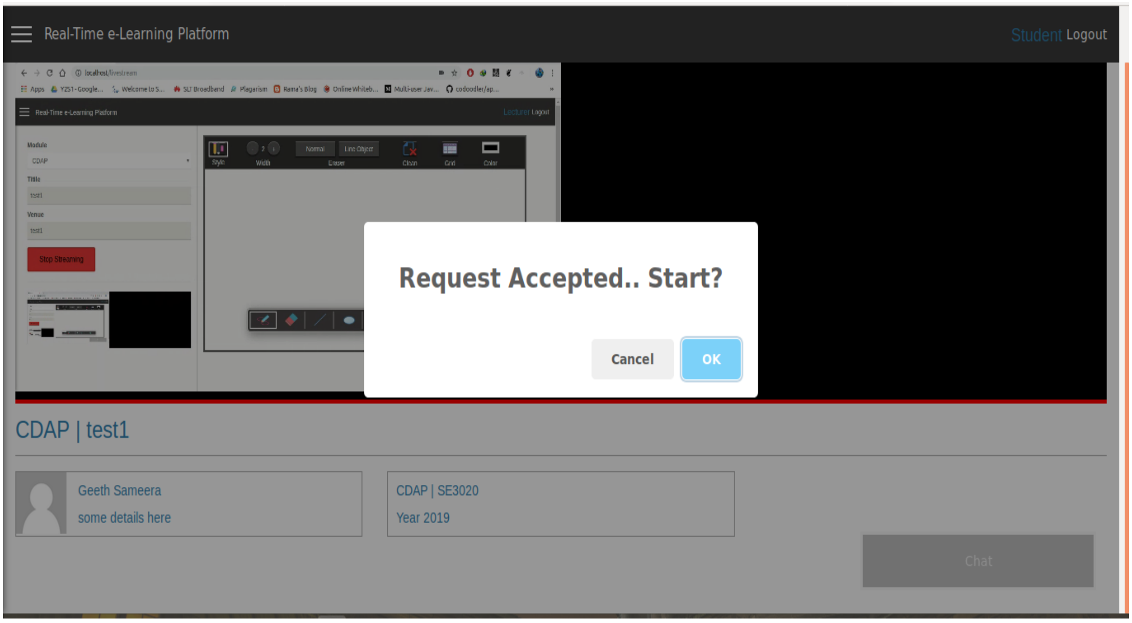

Figure 2.2.8: Raise hand student starts Q&A

After the confirmation of the lecturer popup box appears on the corresponding side and student can start the questioning when he clicks ‘ok’ button. (figure 2.2.8)

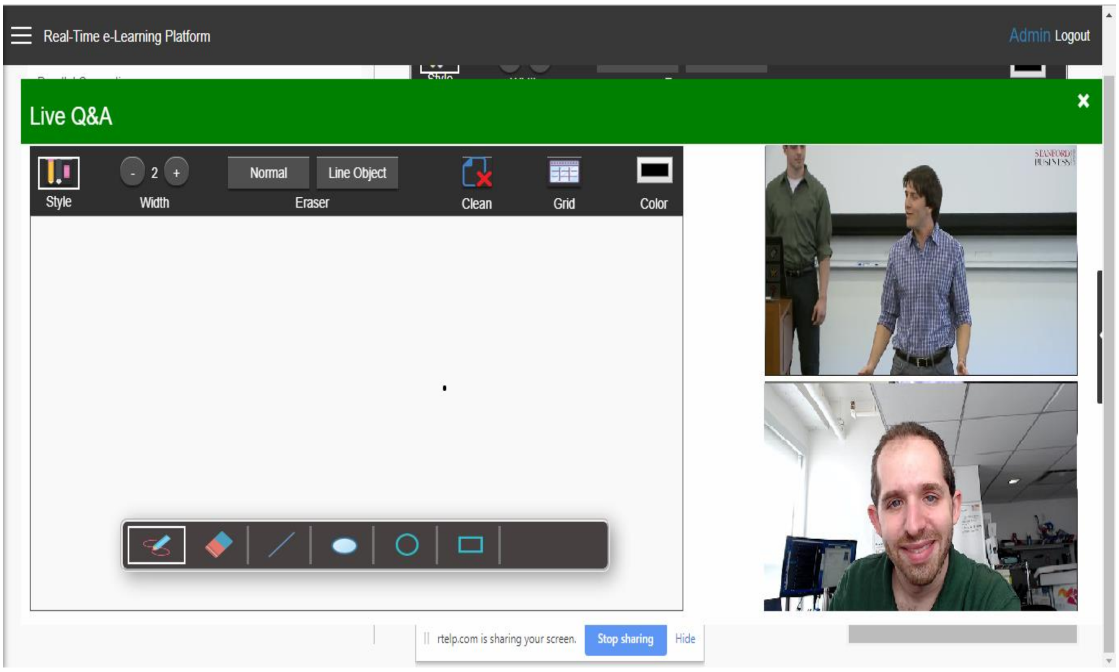

Figure 2.2.9: Live Questioning

After starting the questioning by the student lecturer and students both popup a window and through that live questioning can be done. (figure 2.2.9)

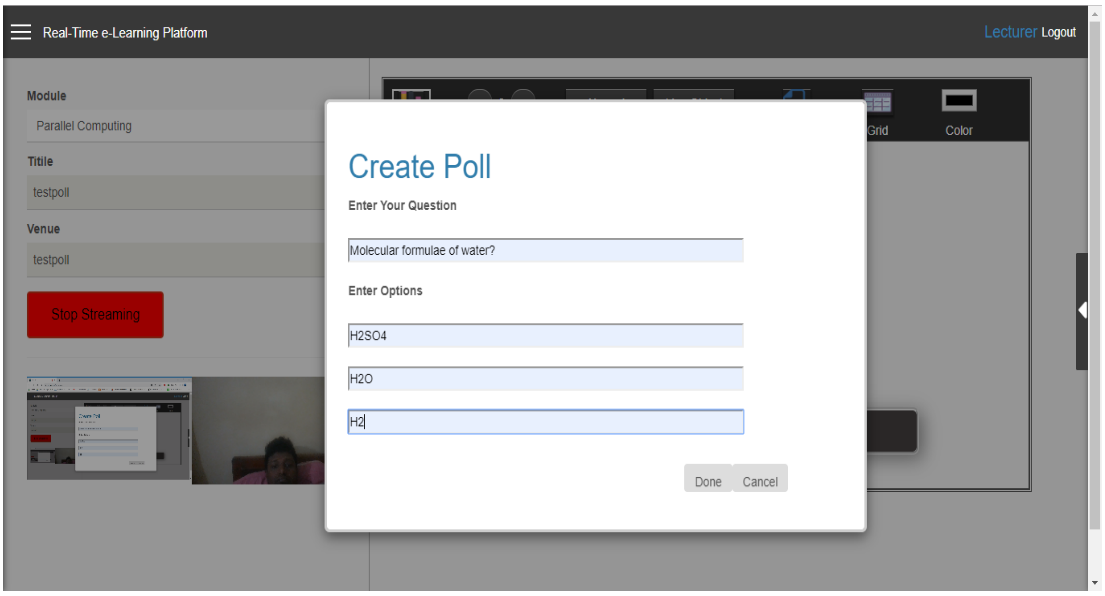

Figure 2.2.10: Create Poll

Figure 2.2.10 shows the creating of a poll by the lecturer. After creating the poll question and the options were transmitted to the students of the respective session.

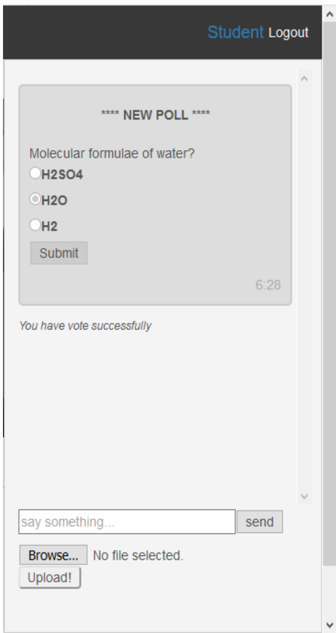

Figure 2.2.11: Student Poll

After lecturer start the poll, question and the options will appear on the chat box of the students (figure 2.2.11). Students can vote their preferred answer here.

Figure 2.2.11: Poll Results

Lecturer can watch the poll results while students voting. Results are appearing in a graphical way. After students finished voting, Lecturer can share the results and then students also able to see the results.

3. TESTING

Testing was done according to the documented requirements in requirement analysis phase. After the finishing of each and every sprint, each member performed unit testing to check whether the module is working according to the requirements. Thereafter the integration of each module system testing was done to check whether the whole system working according to the requirements.

Tracking algorithm was tested on both CPU and GPU to measure the FPS (Frames Per Second) given by each configuration. RTMP Video stream from the PTZ camera is tested under different network bandwidths for any delay.

Live streaming component is tested for the capability of high quality video transmission with minimum latency. Lecturer and students interact with each other during the live session through the chat and the whiteboard features were tested by creating different sessions. That’s to ensure drawings and chat were transmitted only among the respective sessions.

Finally, all the functionalities of the system are tested against different devices and browsers.

4. RESULTS AND DISCUSSION

CPU performance of the tracking algorithm was around 2-3 FPS. Same algorithm was tested in NVidia GTX 1050 and the FPS was around 10-15. GPU performance is sufficient to run the algorithm in real time. Higher network bandwidth is required to send the PTZ Camera stream to the image processing server with a minimum delay.

The livestream component was capable of handling simultaneous streams while maintaining the quality of the streams and with very low latency depending on the network bandwidth allocated for the system.

Functionalities of the whiteboard and chat features were functioning as expected. Application is capable of communicating between lecturer's end and the students end of the application with minimum latency.

Playback videos responded to the variations of the network as expected with a minimum buffer delay. Automatic bitrate changes occur smoothly.

System also supports all major browsers such as Google Chrome, Mozilla Firefox and Safari and devices which had above mentioned browsers.

5. 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.

6. REFERENCE LIST

[1] Docebo, *ELEARNING MARKET TRENDS AND FORECAST 2017-2021*, Debeco, 2016

[2] Chris Curren, "*Strategies for e-Learning in Universities*",CSHE.7.04,2004

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

[4] I U Cooray, W & M P Abhayawickrama, D & Ragel, Roshan. "Real time interactive lecture delivery system." 91-96. 10.1109/ICIAFS.2010.5715641.

[5] McCrohon M, Lo V, Dang J, Johnston C. "Video streaming of lectures via the Internet: An experience" ERIC Clearinghouse; 2001 Dec 9.

[6] Amir Ghahrai, "*Software Development Life Cycle - SDLC Phases*",testing excellence,2018 https://www.testingexcellence.com/software-development-life-cycle-sdlc-phases/. [Accessed: 2019-03-05]

[7] "Requirement Analysis Techniques", *Requirement Analysis Techniques,* visual-paradigm.com, [HTML]. Available: https://www.visual-paradigm.com/guide/requirements-gathering/requirement-analysis-techniques/. [Accessed: 2019-03-02]

[8] D. Ehlers, "*Quality in e-Learning from a Learner's Perspective*", 2004,7

[9] Amir Ghahrai, "*Software Development Life Cycle - SDLC Phases*",testing excellence,2018 https://www.testingexcellence.com/software-development-life-cycle-sdlc-phases/. [Accessed: 2019-03-05]

[10] "Requirement Analysis Techniques", *Requirement Analysis Techniques,* visual-paradigm.com, [HTML]. Available: https://www.visual-paradigm.com/guide/requirements-gathering/requirement-analysis-techniques/. [Accessed: 2019-03-02]

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

[12] "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].

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