EavesDrop

Granthali Jadhav[1], **Hardik Yewale**[2], **Akshay More**[3] **and** **Sainath Patil**[4]*

Abstract

Interactive Live streaming is the process of transferring real-time data over the internet. The data can be of type audio, video or any common data type. Interactive live streaming is an important feature of applications and platforms in which the actions of the audience affect the content of the communication. Simultaneously, Podcasting is one of the types of contents that people are attracted to. The idea of podcasting includes recording the audio or video file and uploading it to a platform where people can access it. Now, with the thought of mixing these two, the idea of an interactive podcast was born. We analyze that low delay is a restriction for Web Interactive Streaming. Interactive communication over the internet with the restrictions of real-time boundaries requires a framework that will allow the development of the proposed project. Real Time Communication (RTC) has wide-industry uses. RTC is the standard and it also increases the browsing model which allows access to live streaming systems which consist of social media, television, chatting applications, as well as the communication media. Users are allowed to read comments, write comments, record the sessions/audio/video, and edit audio/videowhich is done within a Cloud Infrastructure which also provides services of quality. The development of the application is the most important part where using different technologies a system can stream audio and video data over the internet, handle a number of users who can make audio rooms to discuss a topic among themselves, where several people can hear them, and they also have a chance to add value to the conversation with their inputs.

Keywords : **Cloud, interactive, live-streaming, podcasting, Real-Time Communication, time-critical**

I. INTRODUCTION

Podcasting, the conception of audio entertainment, where multiple people can hear a discussion the host is having, thus, creating a window for a special type of content where the users become listeners. They do not need to look at a screen to enjoy the content. A new experience is created for an individual as a person doesn't have to check his phone, he just has to play and listen to the podcast whenever he wants like while working, going for a jog etc. Recent social trends as well as technology advances have led to the emergence of varied famous web-based live streaming platforms like Facebook Live, YouTube Live, and Instagram Live. These platforms are aimed at increasing scalability and are therefore, real-time. These also show high delay[1]. Now, one needs to understand that the operation of settling a real-time medium isn't easy and is dependent on real-time architecture which allows inventors to produce real-time terrain where multiple requests can be transferred from multiple sides. These requests can include questions from one end, acknowledgement from multiple lines, audio crimes, and numerous other effects that count during the live streaming of the data. To put effects into perspective, the operation of a real-time terrain can be done by using the RTC framework which is the real-time communication frame that allows the user to do the following:

There are multiple live technologies that give these features and one of the most popular ones is Google's WebRTC frame. The WebRTC is Open-source, recent,

Manuscript Received : March14, 2022 ; Revised : April18, 2022 ; Accepted : April22, 2022. Date of Publication : June 5, 2022.

G. Jadhav*[1], *Graduate*, Email : jadhavgranthali@gmail.com ; ORCID iD : https://orcid.org/0000-0002-0240-0708
H. Yewale[2], *Student*, Email : yewalehardik@gmail.com ; ORCID iD : https://orcid.org/0000-0002-9985-4772
A.More[3], *Student*, Email : moreakshay725@gmail.com ; ORCID iD : https://orcid.org/0000-0003-3359-467X
S. Patil[4], *Assistant Professor*, Email : sainath.patil@vcet.edu.in ; ORCID iD : https://orcid.org/0000-0002-8226-4389
[1,2,3,4]Vidyavardhini's College of Engineering and Technology, Vasai(W) - 401202, Maharashtra.

DOI : https://doi.org/10.17010/ijcs/2022/v7/i3/171268

and an upcoming technology [2]. It allows users to view video content as well as comment on it while streaming it real-time for communication. WebRTC allows real time communication using the API that helps transfer voice and audio through the internet using JavaScript code. Lately, new platforms were introduced for Real-time, that dispatch service browser embedded operation or web operation. Within this operation, WebRTC has stepped into similar interest as there are many new applications that support API. This is common for browsers such as Google Chrome, Mozilla Firefoxetc. HTML5 is supported by WebRTC that establishes and holds the Connection of Peer and Media Stream, as well as it can be used for Chat Rooms using API to have P2P communication with peers [3]. Though the WebRTC frame works well, it comes with a fiscal cost which can be avoided. The AgoraRTC Framework is less expensive and provides a healthy amount of original talk time and it is smarter. Now, as we talk about how we can apply the RTC frame along with live-streaming capabilities, it is important to note that development is the most important. The development of the operation was believed to be effective in Flutter as it supports both Android and iOS. Also, using the Firebase for Database and Server purposes was beginning to be veritably effective as it handed the functionality which was necessary for the operation.

In 2010, Server Socket.io was created. Socket.io works in two ways: (a) It allows bi-directional communication which is done between server and clients. The server Socket.io also has a huge community, which means getting help is fairly easy. This allows the developer to open the connection and this helps in real time communication which is a phenomenon of time. This communication can be done only when Client holds Socket.io within the browser. The server should also have Socket.io package. Data can be transferred in the form of JSON. These days popular web operations which are real-time use PHP which is a traditional method and is difficult. It includes pooling the changes into server, having track for time stamps etc. Sockets play an important role in Real-Time Systems Architectures as well as Bi-Directional Communication within clients through the servers. It also says that server pushes guests into the room. When an event is generated, the users are pushed to particular guests. The Socket.io server is very popular and is used by Zendesk, Yammer, Microsoft Office, Trello, as well as many real-time applications. GitHub is the most important JavaScript which is depended on Node Package Manager Module, i.e., NPM.

II. PROBLEM STATEMENT

Transfer of audio data packets from the source to the destination follows the path of all the participants and those participants can hear the audio packets which eventually mean that they hear the discussion that the hosts are having and thus, serve the idea of Podcasting. Participation of the followers in the discussion can make the participants feel like they are a part of the discussion and thus, creates a positive impact causing the hype of the operation through the audience themselves. The Live-Streaming system solves few problems in the existing Interactive Live Streaming. One of them has the significance that the systems are web based. Hence, at last, there has been an important trend to shift towards web based operations. Few features are similar to multimedia that had limited support traditionally. The operations that were totally dependent had to find another way and had to calculate the non-plugin Standards [1]. Fig. 1 shows importance of latency in real time systems.

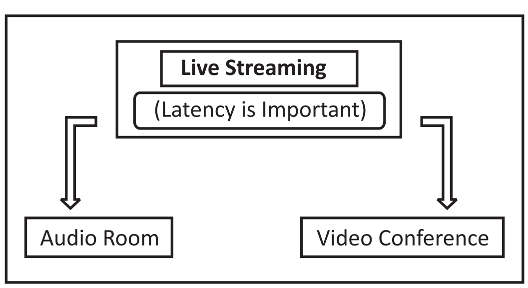


Fig. 1. Importance of Latency in Real-time Systems

III. RELATED WORKS

In the era of the internet the works of similar systems are found in different aspects with different purposes. As everything has faults, systems were developed to make the user use them more efficiently and easily. Thousands of written and videotape content are posted on social media but there is no scope of increase in audio content. There are hardly any platforms available to explore someone's audio skills. Colorful enquiries include system features like chatboxes, audio, and videotape relations in live sessions. The exploration included the armature of WebRTC, Chatbox algorithm, different approaches for interactive communication and much more. The interactive relation through audio approaches is encouraged with the existing systems. Applications of the Live Interactive Streaming are as follows:

- Skype, Google Hangouts, and FaceTime from Apple help in Video Conferencing.
- Users can understand what is passed in the real world through Remote Rendering Systems.
- Surveillance systems allow the client to send videos through servers.

A. Architecture of WebRTC

The most important illustration of a videotape conferencing system is Skype by Microsoft before the of advent of WebRTC. WebRTC provides peer-to-peer browsing [4]. Peer connection enables communication that is direct between all the users. We can also say it is peer-to-peer communication or it can be browser-to-browser communication [5]. The framework allows P2P communication which is translated end-to-end for both audio as well as for video content and the data is transmitted. Fig. 2 shows the architecture of Web RTC. The framework of WebRTC includes three parts:

(1) Layer for web developer

(2) Layer for browser developer

(3) Service layer

This tier contains Video Engine, Voice Engine, and tools that help in Transport Communication.

Voice Engine helps in Audio Processing. It includes Audio processing and decoding. Videotape machine deals with image processing and codec. The transport point includes ciphering the collection of audio and videotape, and transmitting it over SRTP protocol.

It is important to understand that WebRTC isn't just a single API, rather it is a collection of APIs which also include protocols for working groups that are similar to W3C (World Wide Web Consortium) and IEF (Internet Engineering Task Force) [3].

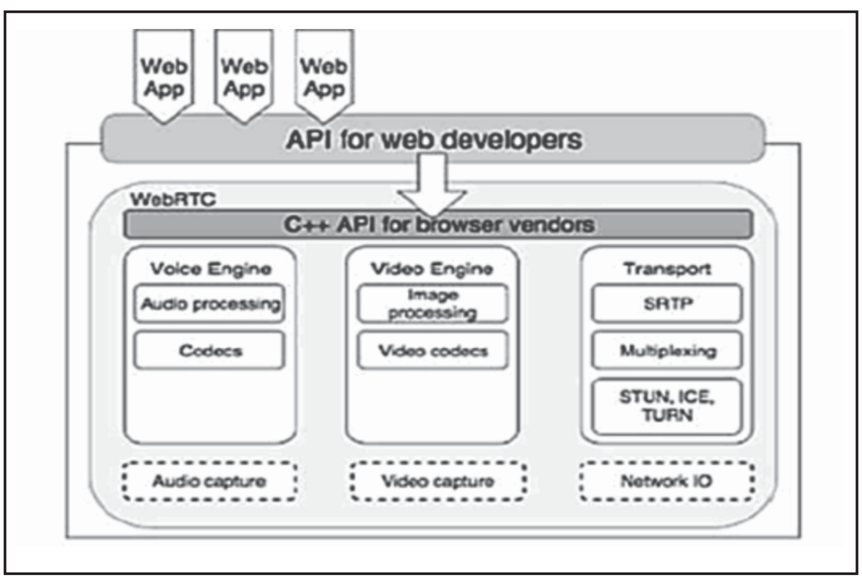


Fig. 2. Architecture of Web RTC

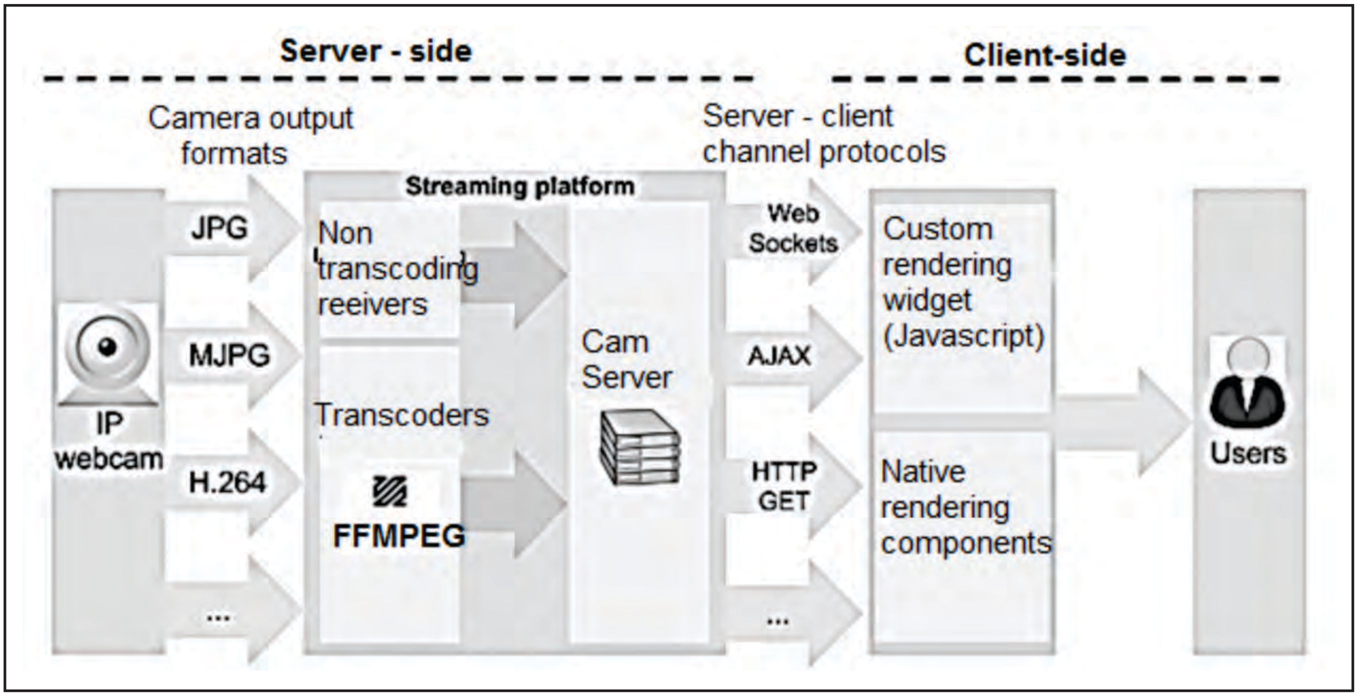


Fig. 3. Live Streaming

Additional features of WebRTC are as follows:

(i) Streaming of audio, video or any data.

(ii) Report errors to coordinate communication.

(iii) Understand information about the network.

(iv) Communicate streaming audio, video, or data.

B. Interactive LiveStreaming Approach

We shall describe and dissect numerous of the colorful approaches available for web-based interactive live streaming. One of them is Refreshing Image and M-JPEG approach which is frequently employed with infrastructures for live-streaming . Also, we include some new approaches which are more infrequently used, several of which have only lately become available, thanks to their reliance on new under-development web norms. Fig. 3 shows the simplified working of Live Streaming interaction from the time it is captured to the time it is displayed. The inflow starts when the IP webcam captures a frame. Different IP webcam models are live, and different models support different formats. The maximum used are JPG (separate images).

The cameras helps us in streaming platform, counting the source and target formats through those formats. It may not be necessary to transcode it into a special format. Transcoding can take a big quantum of processing power and adds some quiescence. The server will store images and stream them for people who are surfing the internet. Various channels of client server protocols are available. If the data is delivered, the surfer will get the data that is handed and then it calculates the native element that will help to reuse, render, and crack using JavaScript [1]. Fig. 3 shows Live streaming.

C. Chatrooms

Whenevera user wants to have the knowledge for any content, the user can look for it by giving a keyword. A user searches for the appropriate room and joins it. If no such room exists, one can create the room of appropriate name. After users join a dialogue room, they can decide whether the dispatches they shoot will be displayed alongside the sender name. If the user wants to be visible to other users also, the dispatches are displayed with the username. If the user does not want the name to be visible, a singular id is generated as a username for that specific dialogue room and thus, the dispatches are displayed with the generated name. Chatrooms can be deleted by admin as well as the user who created it. The subsequent flowchart explains the dialogue room workflow [6]. Fig. 4 shows the algorithm for chatrooms.

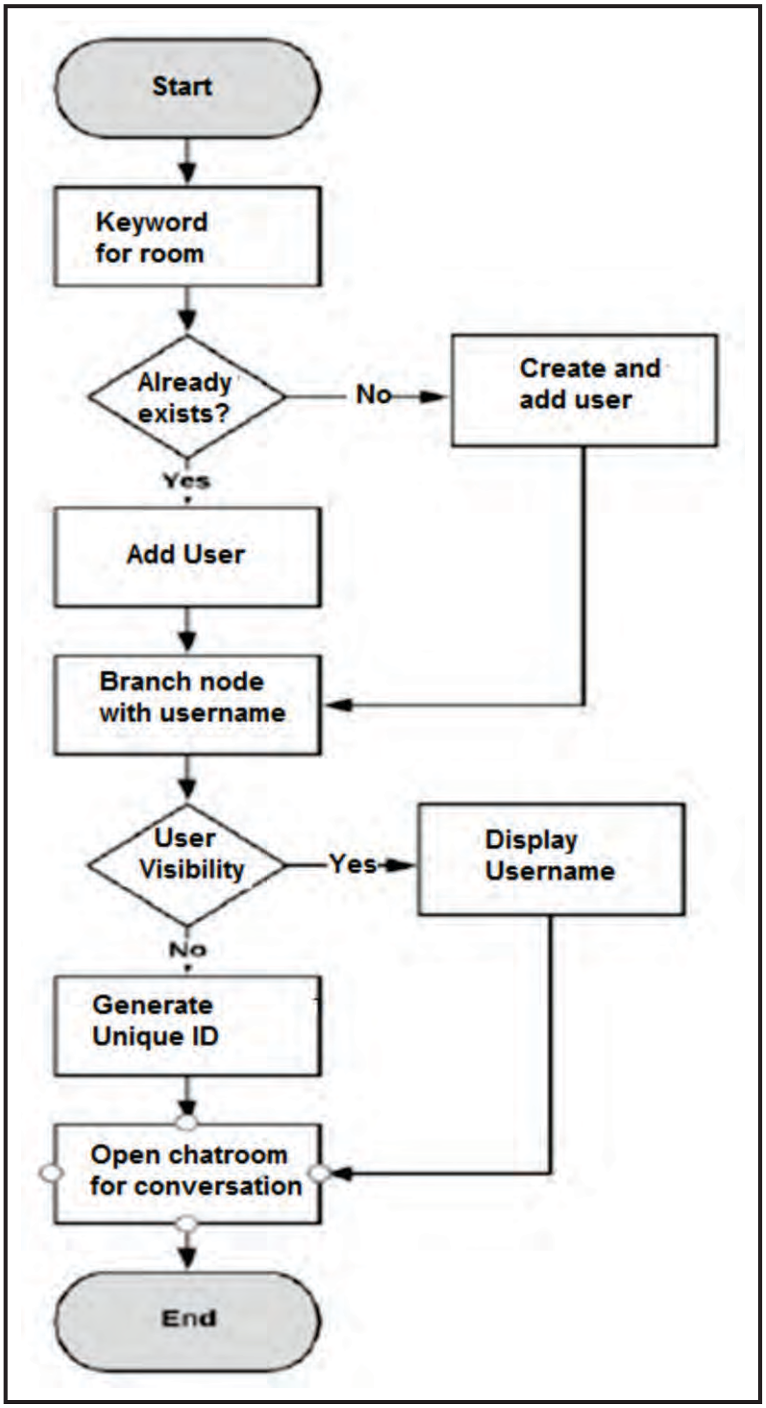


Fig. 4. Algorithm for Chatrooms

IV. COMBINED STUDY

Audiences are increasing day by day. The users are increasing more for radio than for music. Until now, it was considered that number of radio listeners is decreasing day-by-day, instead it is still increasing [7]. Fig.5 shows that 63% of users listen to podcasts at home, 11% at work, 11% in car/truck, 4% while walking, and 8% others [5]. Fig. 6 gives us the details of overall users listening to Podcasts till 2019.

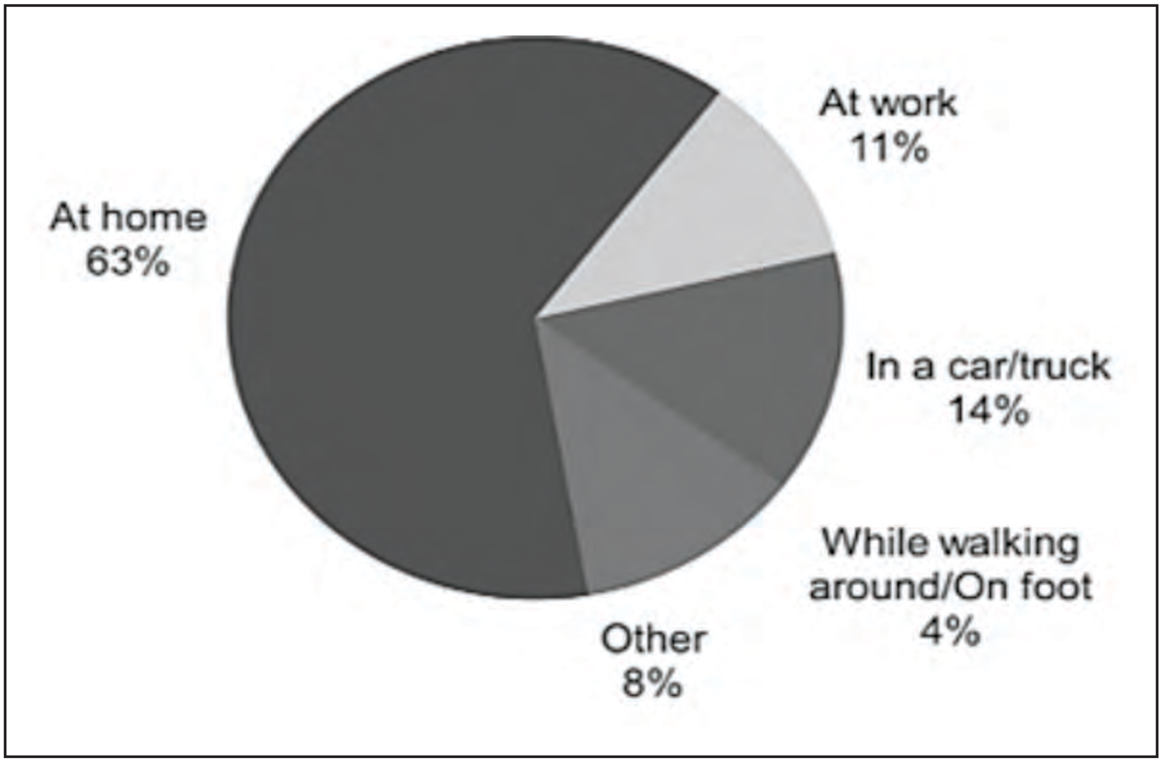


Fig. 5. Podcast Users

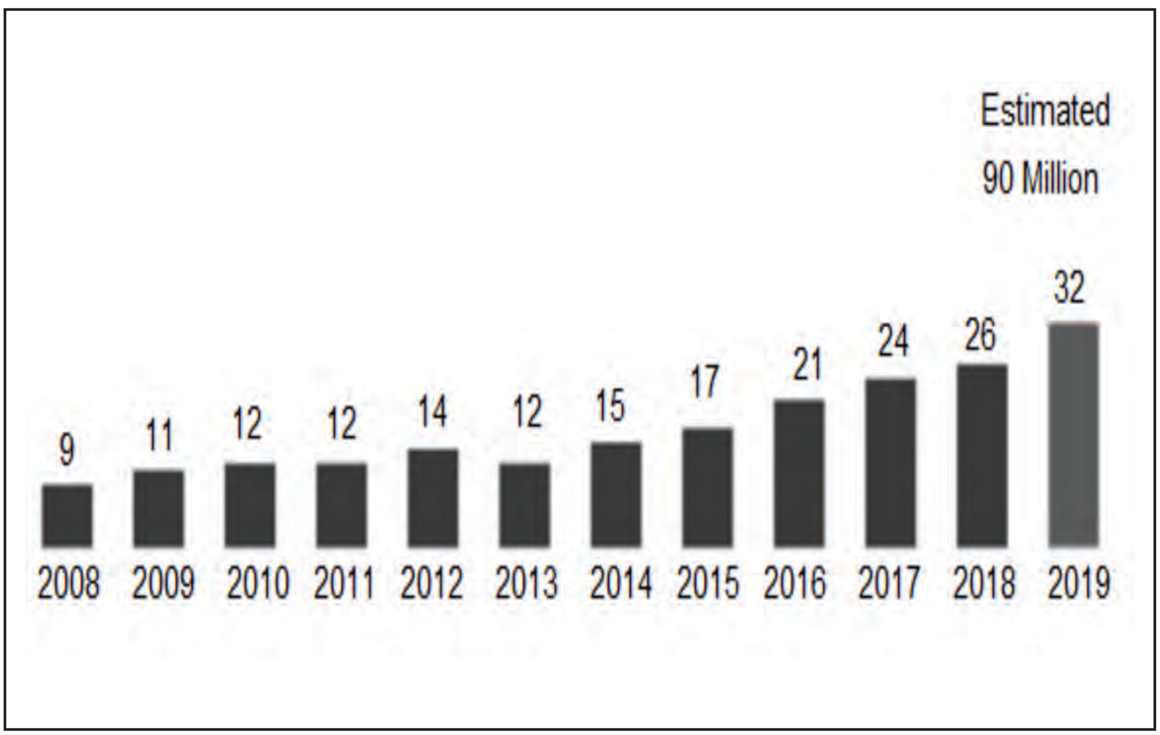


Fig. 6. Overall Listeners of Podcasts

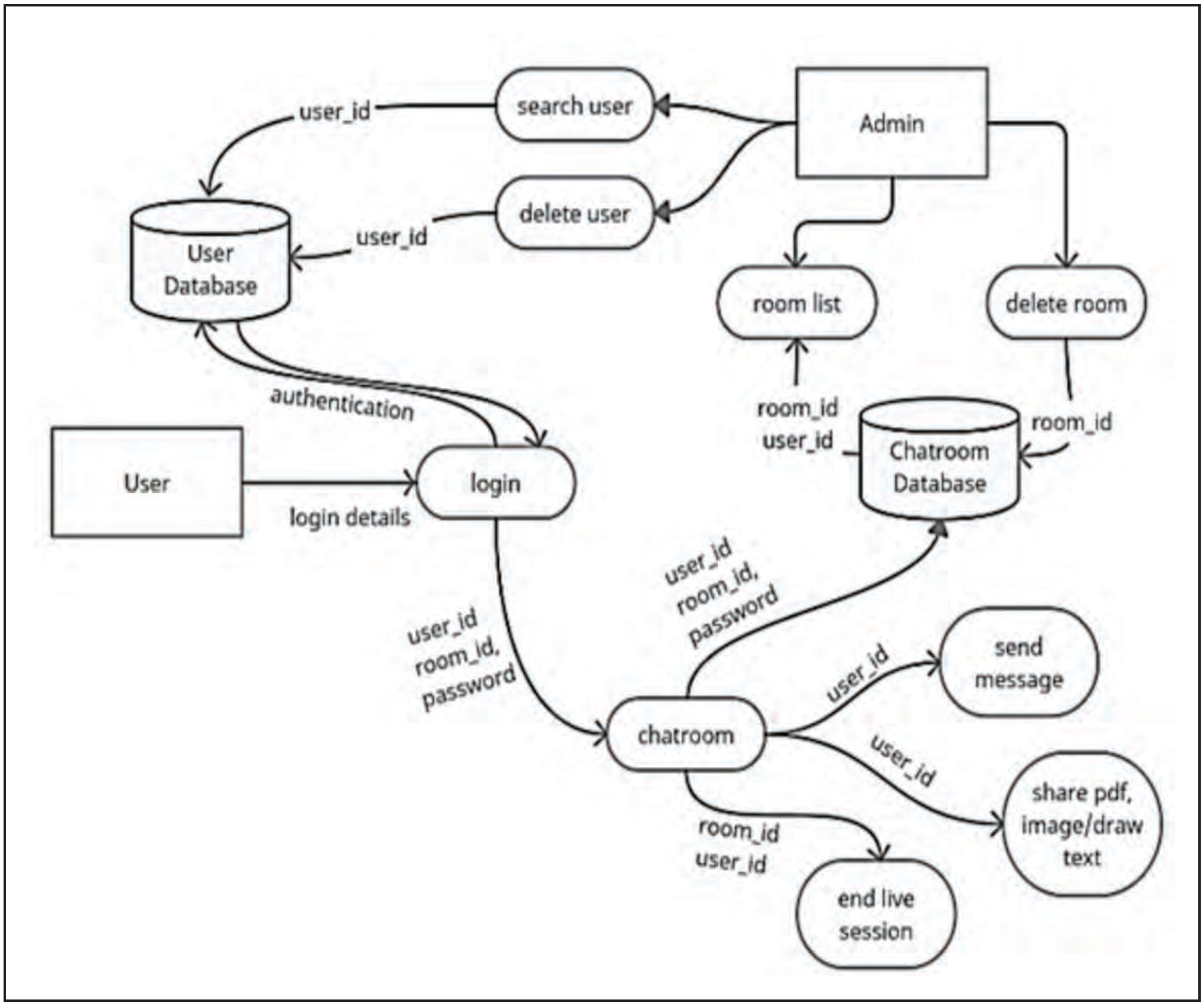


Fig. 7. Dataflow Diagram

V. PROJECT IMPLEMENTATION

A. Data Flow Diagram

Data flow diagrams in simple words shows how the overall data flows within the system. As our system provides the user with the rights of creating a live chat room which can consist of 'n' number of participants, it assigns them rights of logging into the system. Then they can share PDFs, send messages. and also end the live sessions once they enter the chatroom.

The data flow for admin is also very simple. All the data flows through the admin panel where the room list, user-ids, room-ids, and various other uniquely generated information is then forwarded to the database which the admin can access (Fig. 7).

B. Activity Diagram

The Activity diagram (Fig. 8) shows the flow of the system, that is how it will work, the result, and the flow of the projects. The word "data" is plural, not singular.

Use case diagram (Fig. 9) represents users and their respective jobs, as well as the unique duties assigned to them which they have to carry out.

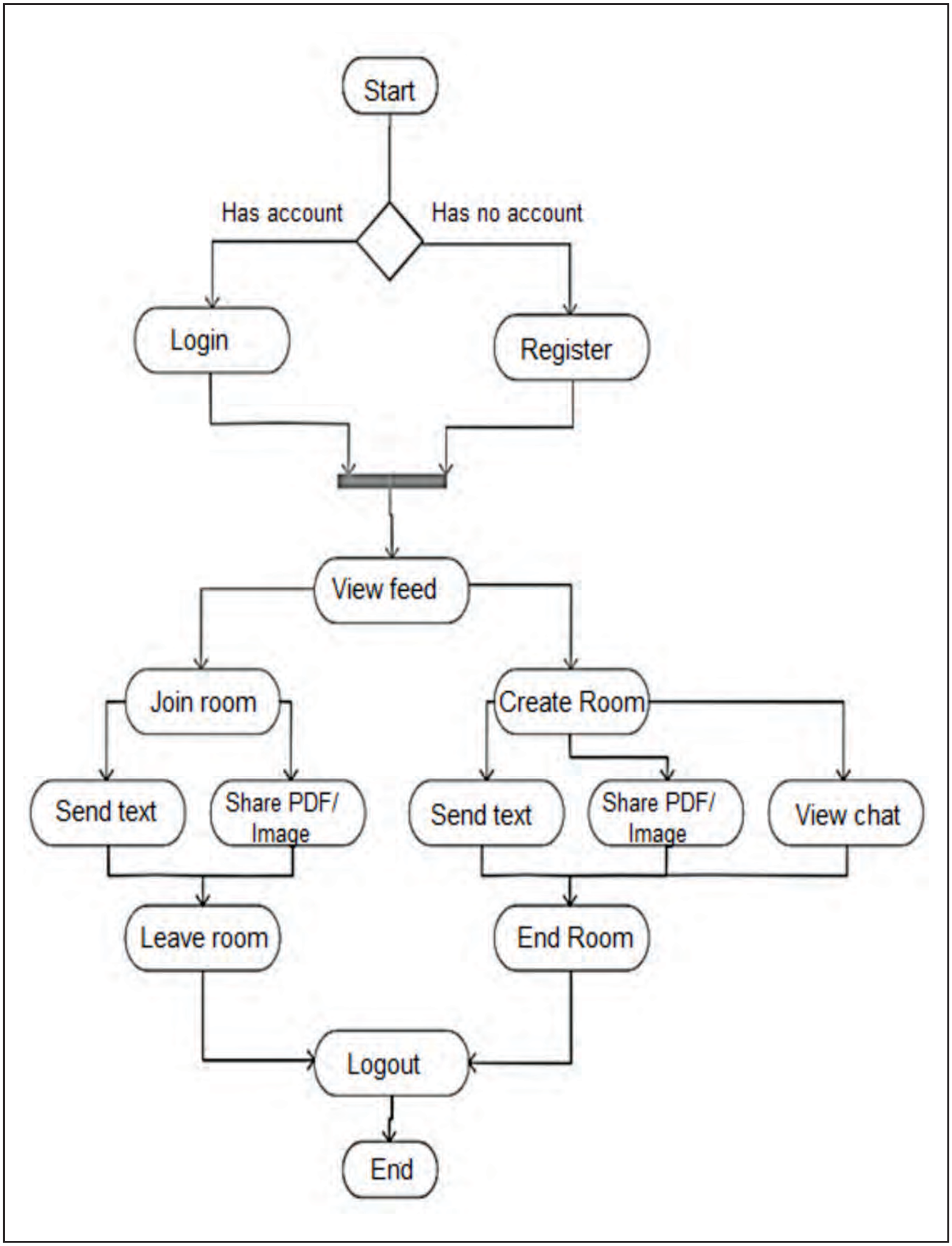


Fig. 8. Activity Diagram

C. Use Case Diagram

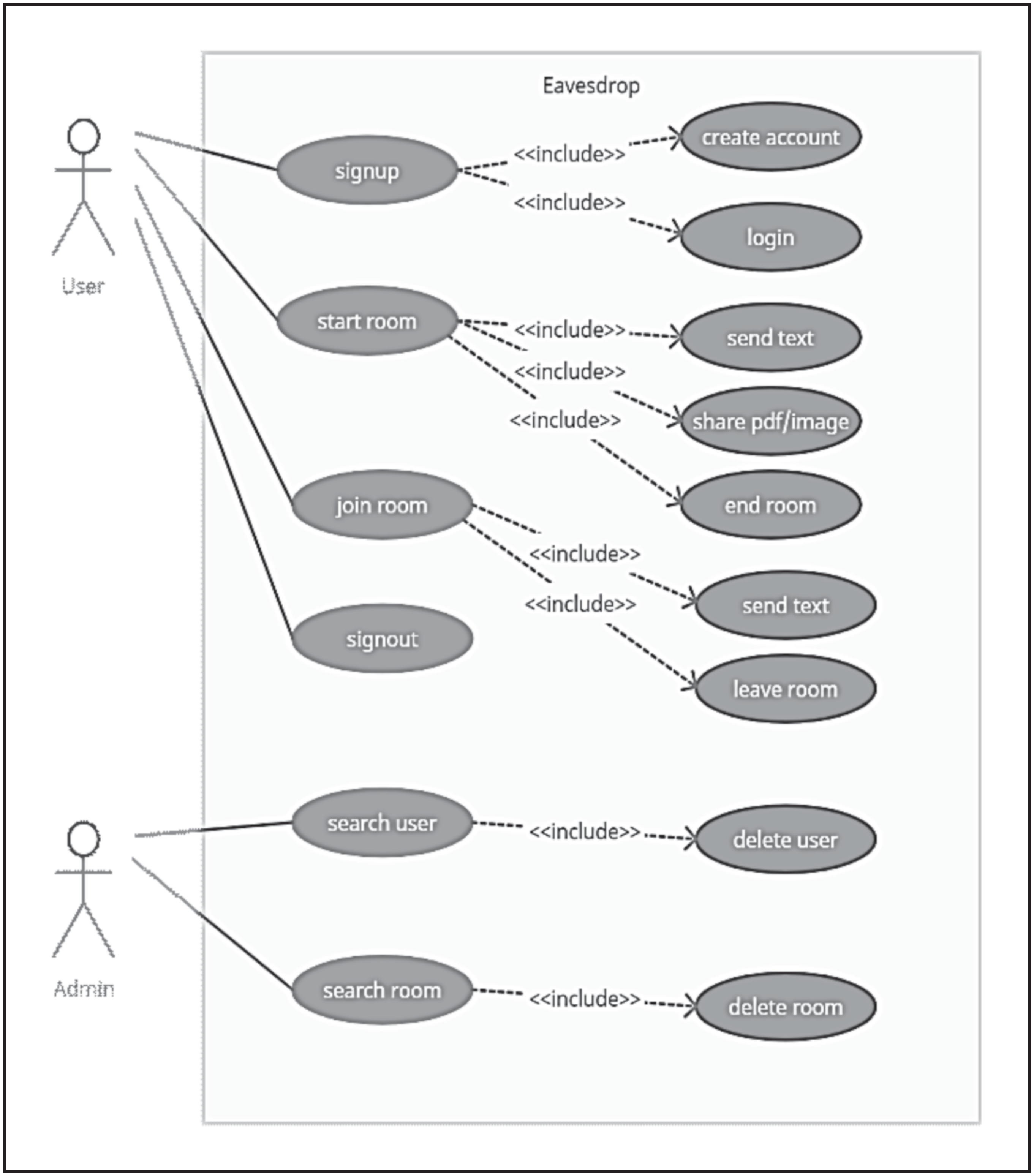


Fig. 9. Use Case Diagram

D. Block Diagram of the System

As shown in the block diagram (Fig. 10), there are four main components of the system,

- Server Socket.io
- RTC Multiconnection Library,
- User
- Admin

The administrator holds together the user as well the chatrooms they create, which obviously puts the admin in a very commanding position. The work involves creating rooms, handling them, and distributing them in two components, i.e. Server Socket.io, and the RTC Multiconnection Library respectively. When the user presses create room button, a request is processed from RTC Library which in return, processes a request to the Server Socket.io, which means it is requesting for a particular socket on the internet that streams audio and video. The request when accepted is processed for the user which results in room creation and various ID generation.

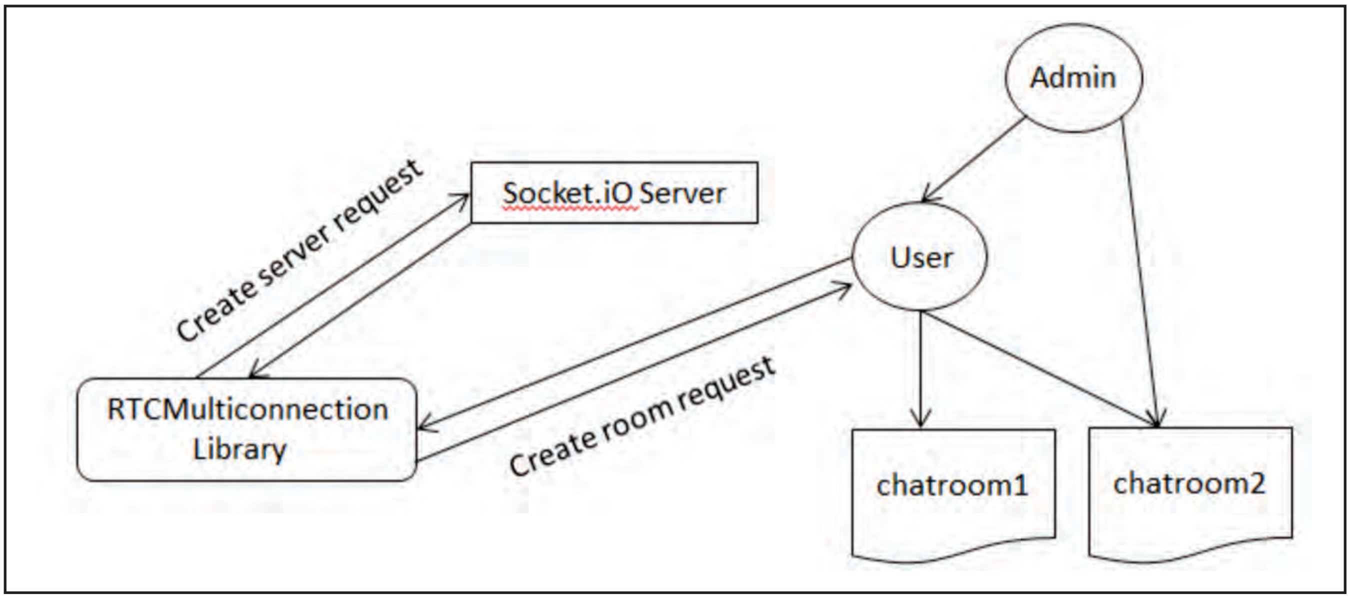
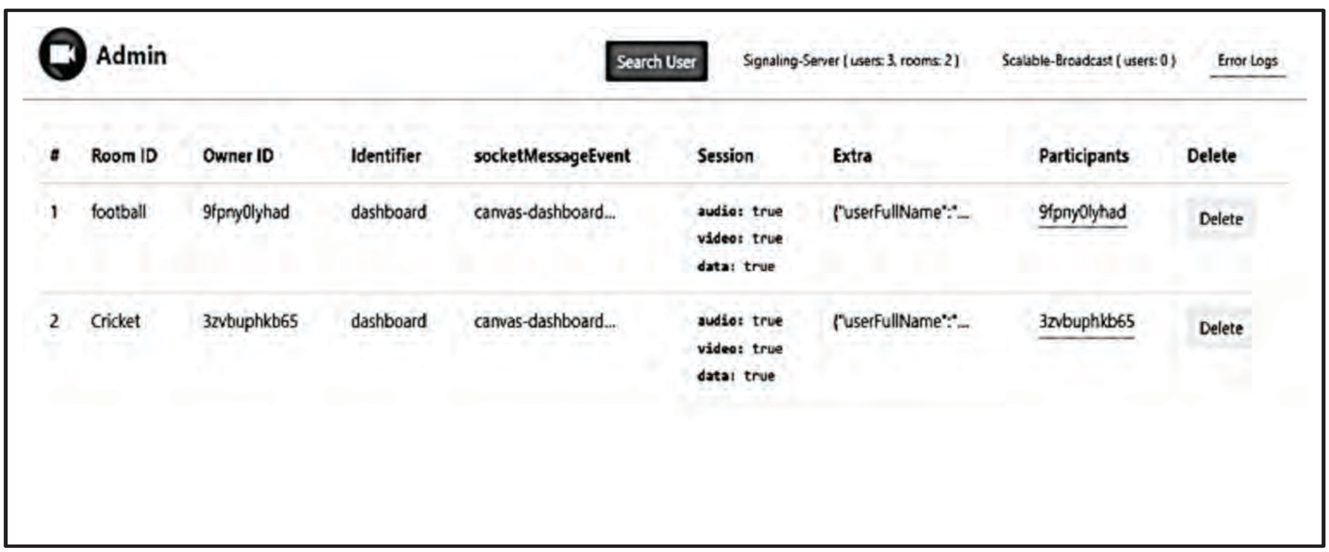


Fig. 10. Block Diagram

Fig. 11. Dashboard

Fig. 12. Conference Room



#	Room ID	Owner ID	Identifier	socketMessageEvent	Session	Extra	Participants	Delete
1	football	9fpny0lyhad	dashboard	canvas-dashboard...	audio: true video: true data: true	{"userFullName":"...	9fpny0lyhad	Delete
2	Cricket	3zvbuphkb65	dashboard	canvas-dashboard...	audio: true video: true data: true	{"userFullName":"...	3zvbuphkb65	Delete

Fig. 13. Room Details

VI. RESULT SNAPSHOTS

Fig. 11 to 14 show the results.

VII. CONCLUSION

The system will be a real time communication application for users so that they can communicate with ease to make sessions more interactive. The users can create their chat rooms for exchange of content and information. This video conferencing technology is increasing day-by-day due to COVID-19. The system includes audio and video medium of interaction with audience, where audience interaction plays a negligible role. With understanding of related works, the demerits of existing systems can be listed. The application includes three main features:

(1) Live streaming

(2) Audience interaction

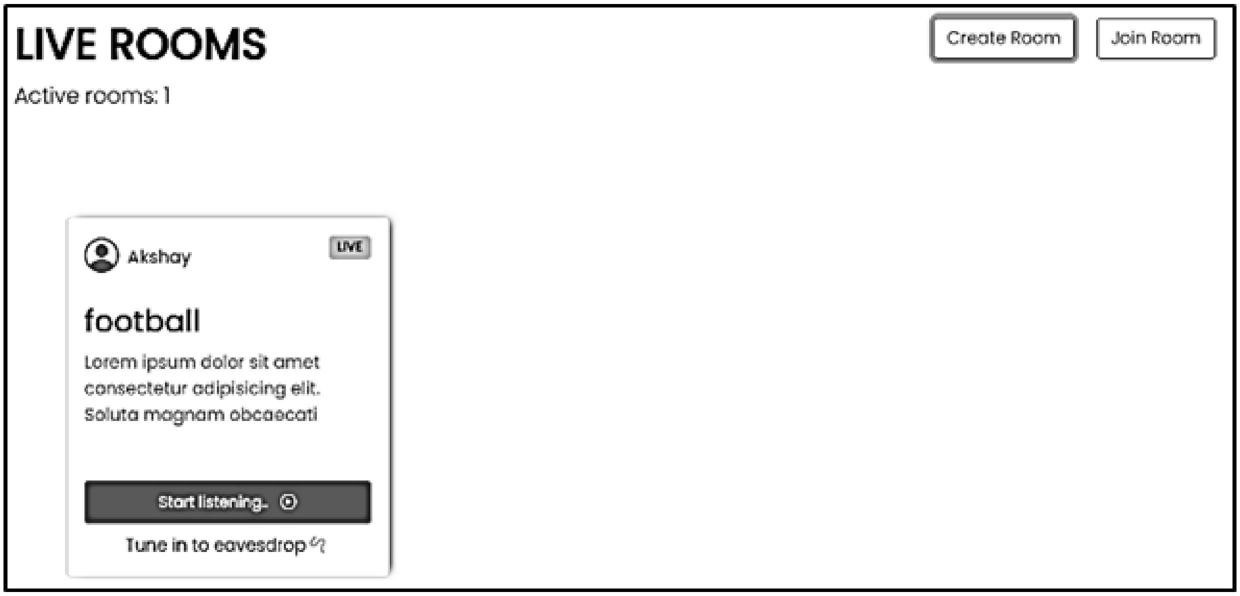


Fig. 14. Live Rooms

(3) Chatrooms

The goal is to implement the EavesDrop streaming application with audio, video, live facilities for the audience, where WEBRTC helps the developer to build better Real-time communication systems. The system can be further improved by streaming the audio interactive sessions on various platforms and much more.

AUTHORS' CONTRIBUTION

The present project was developed under the guidance of Prof. Sainath Patil who continuously motivated the team, provided knowledge, and expertise to build and resolve issues. Hardik Yewale was the Project Lead as well as Project Developer who consistently maintained the scope throughout the project. The task of Live Streaming using WebRTC was successfully implemented by him. Akshay More designed and developed the front end, resolved streaming errors and errors at various logs. Granthali Jadhav started with the literature survey of various papers, defined the scope, and understood the requirements. She designed the logs wherein the admin has control over all the rooms and she was also responsible for documentation.

CONFLICT OF INTEREST

The authors certify that they have no affiliation with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or material discussed in the manuscript.

FUNDING ACKNOWLEDGEMENT

The authors did not receive any financial support for the research, authorship, and/or for the publication of the article.

REFERENCES

[1] L. Rodriguez-Gil, P. Orduña, J. García-Zubia, and D. López-de-Ipiña, "Interactive live-streaming technologies and approaches for web-based applications," *Multimedia Tools Appl.,* vol. 77, pp. 6471–6502, 2018, doi: 10.1007/s11042-017-4556-6.

[2] V.Amaral, S. R. Lima, T. Mota, and P. Chainho, "Exploring WebRTC technology for enhanced real-time services," in Á. Rocha, A.Correia, F. Tan, and K. Stroetmann (eds) New perspectives in Inform. Sys. Technol., vol. 2., *Advances Intell. Sys. Comput.,* vol. 276. Springer, Cham. 2014, doi: 10.1007/978-3-319-05948-8_5.

[3] G. Suciu, S. Stefanescu, C. Beceanu, and M. Ceaparu, "WebRTC role in real-time communication and video conferencing," *2020 Global Internet Things Summit*

(GIoTS), Dublin, Ireland, 2020, pp. 1–6, doi: 10.1109/GIOTS49054.2020.9119656.

[4] M. H. Rahman, "A survey on real-time communication for web," *Scientific Res. J.*, vol. 3, no. 7, pp. 39–45, 2015. [Online]. Available: http://www.scirj.org/papers-0715/scirj-P0715273.pdf

[5] Ediso Researh, "The podcast consumer Canada 2018," EdisonResearch.com. [Online]. Available: https://www.edisonresearch.com/the-podcast-consumer-canada-2018/

[6] S. S. R. Emmadi and S. Potluri, "Android based instant Messaging application using Firebase," *Int. J. Recent Technol. Eng.,* vol. 7, no. 5S2, pp. 352–355, 2019. [Online]. Available: https://www.ijrte.org/wp-content/uploads/papers/v7i5s2/ES2063017519.pdf

[7] J. Moore and O. Moore, "After a breakout year, looking ahead to the futureof podcasting," Techcrunch.com, 2019. [Online]. Available: https://techcrunch.com/2019/08/21/after-a-breakout-year-looking-ahead-to-the-future-of-podcasting/

About the Authors

Granthali Yadav is Graduate in Information Technology from Vidyavardhini College of Engineering and Technology, Vasai. Her experience includes various internships in designing and front end development.

HardikYewale is Engineering Undergraduate (Information Technology) at Vidyavardhini's College of Engineering and Technology. He has done projects related to various domains such as Web Dev, Mysql, and Live streaming framework, etc.

Akshay More is Undergraduate (in Information Technology) from Vidyavardhini College of Engineering and Technology, Mumbai. He is interested in developing new technologies. He has experience in coding in the languages: C++, C#, Java, Python, JavaScript, HTML, CSS, SQL, and has expertise in Reactjs. He also has a keen interest in project management and the product owner field, especially in Agile approach and Scrum framework, which he plans to explore over the next few years.

Sainath Patil is a Research Scholar and is doing research on Cybersecurity at Thakur College of Engineering and Technology, University of Mumbai. He completed Bachelor of Engineering (Electronics Engineering) and Master of Engineering (Computer Engineering) in 2000 and 2005 respectively from University of Mumbai, India. He is Assistant Professor with the Department of Information Technology Vidyavardhini's College of Engineering& Technology, University of Mumbai, India. His research interests include computer networks, cyber security, Big Data analytics, and image processing.