HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Web technologies and e-Services
PROJECT REPORT

Online message and video call

GROUP 4

Nguyen Ngoc Dang	20200149
Nguyen Thanh Dat	20205178
Cao Dang Dat	20205177

Class: 139398
Lecturer: Ph.D. Do Ba Lam

Ha Noi, 6/2032

Table of contents

I. Introduction

Over the past few years, the Covid-19 pandemic has had a profound effect on the way people communicate and work with each other. Online meetings or real-time messaging have become popular as a time- and distance-saving solution.

Based on the knowledge of web-app design in the Web Technology & Electronic Services course, we decided to apply our knowledge and skills to create an Online message and video call website. This website serves similar to Zalo to help people add friends, text, and create online chat rooms.

II. Functionalities

To complete this project, we decided to create 2 subprojects. One is an online message site, the other is for video conferencing.

• Messaging:

- Sign in/sign out: using email and password.
- Find other user.
- Chat real-time with others user.
- Create group chat, chat with group.
- Send the image, video, file

• Video call:

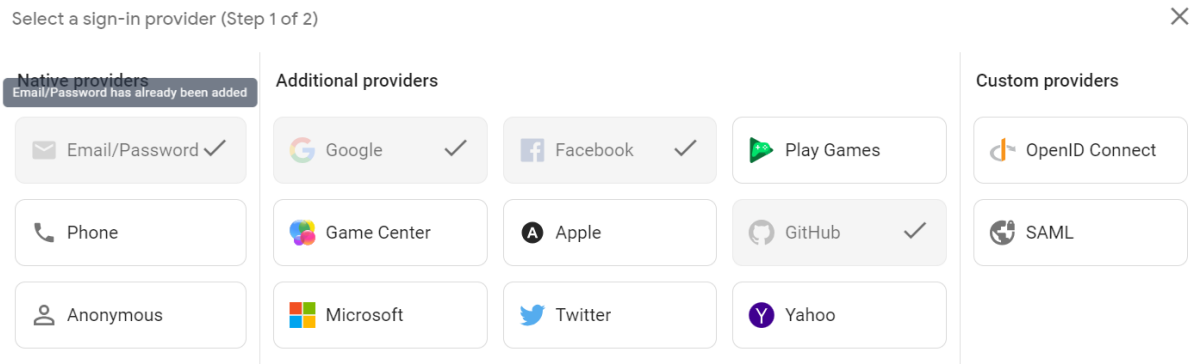
- Call video and sound to each other
- Share the screen
- Real-time chat in a room
- Show all participants in a room
- Control others camera and sound.
- Group call

III. Tech

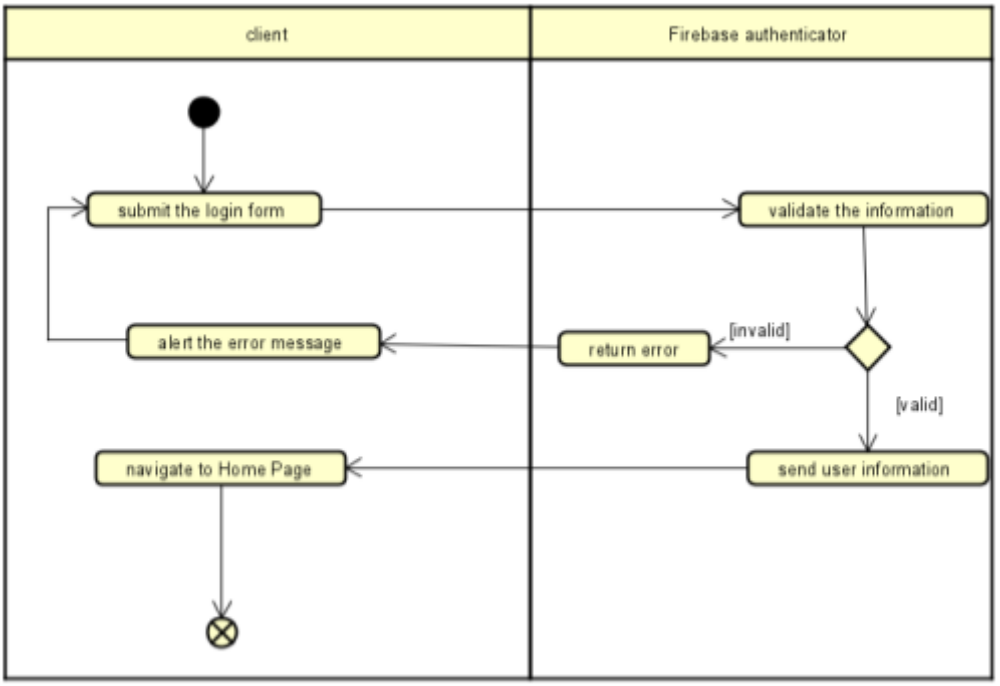
In this project we focus on using Nodejs for the back-end and React-js for the front-end.

• **Login**:

Firebase offers various functions and APIs for authentication, including options to authenticate through Facebook, Google accounts, or email and password.

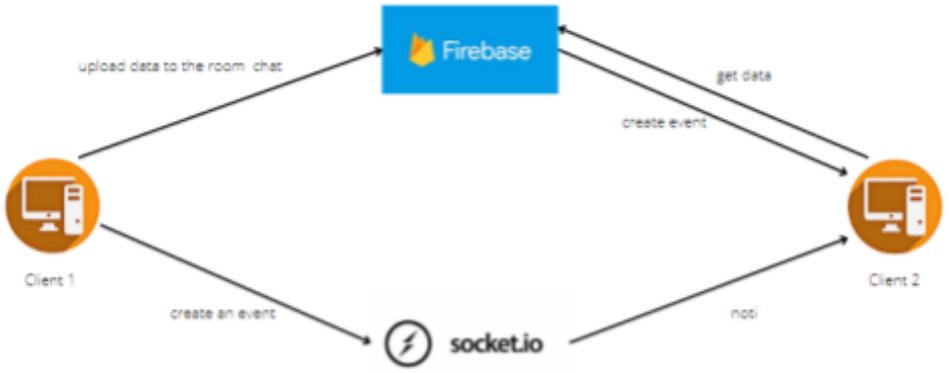


- For example, to connect this app with google account, we need to set up Firebase project (go to firebase.google.com). After that, copy the firebaseConfig object which will help us later to initialize our web app and connect it with firebase.



Login activity diagram

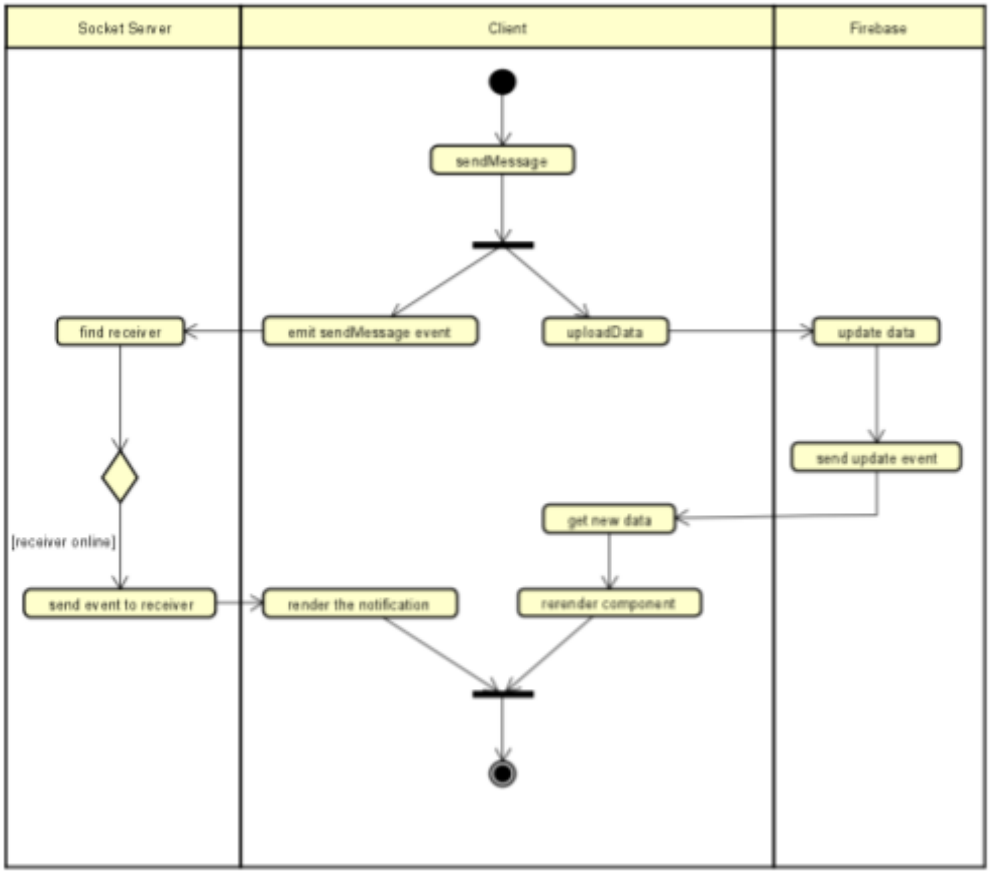
• **Messaging**:



Chat Realtime Feature

Firebase also provides real-time database updates, which are triggered whenever there is a change in the database. Then in both client get the message that database has updated, and reload the data for the newest data.

In addition, we utilize socketIO. Once a user successfully logs in, they will connect to the server's socketIO and listen for events. The user's information is sent to the server, enabling us to track who is online or offline. When the user send a message, client will create an event “send message”, then the server socketIO will send an event to the receiver client (if they are online) then create a notification on the screen.

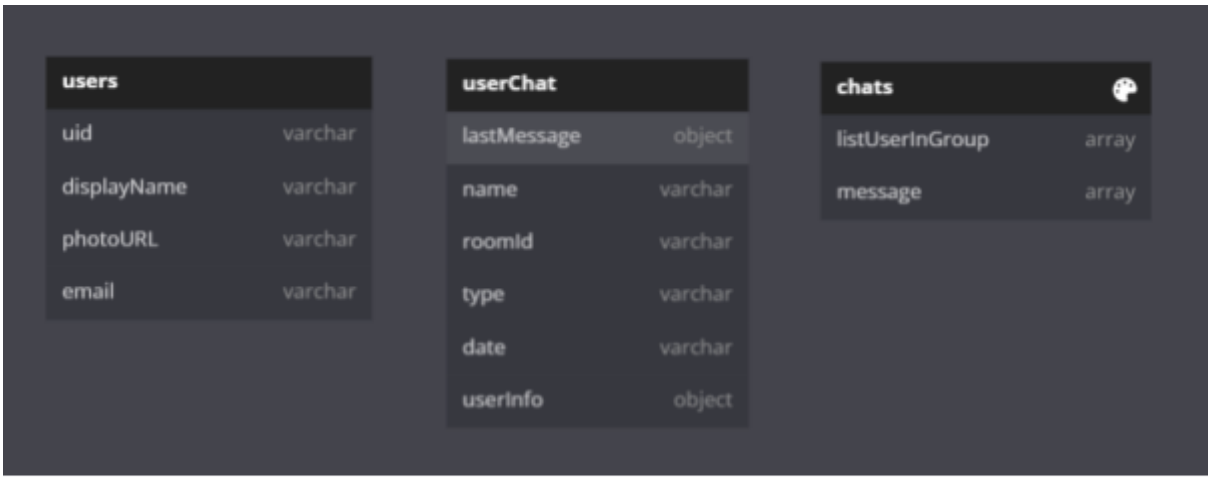


Send Message Activity Diagram

• **Database Design:**

Firebase is a NoSQL database. In this project, we created 3 document: user, userChat, Chats.

- user: store information of user, each user has unique uid, gmail.
- userChat: store information of each room chat that connects with the user. We have two type of room “DirectMessage”, “Group”
- chats: store history of the conservation in each ‘*userChat*’, listUser in that group chat, will be store here

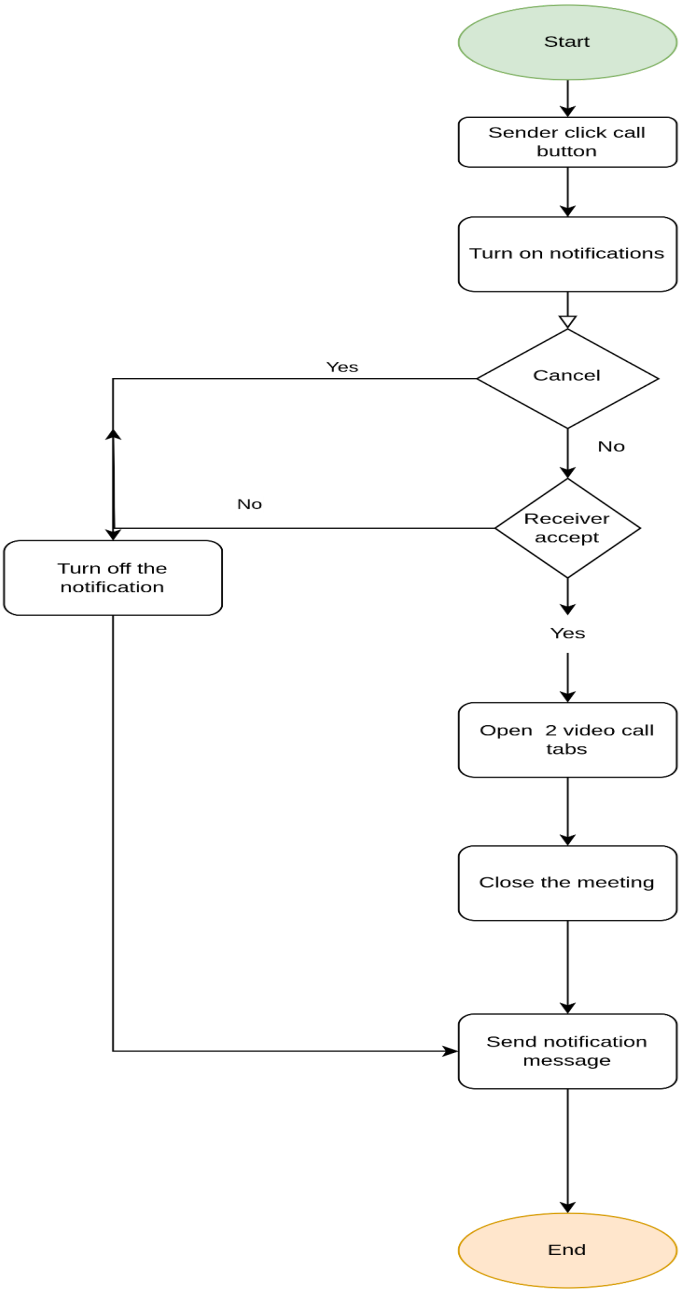


Database Design

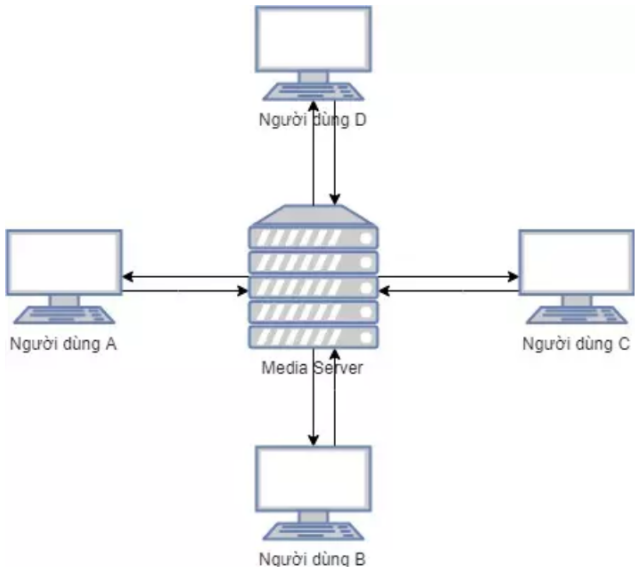
For the upload file (image, video, …), we upload to the Firebase Cloud, and send the link of that file to the “chats”.

• **Video call:**

Initially, when the call button is clicked, the notification event will be fired, sending a call request to the recipient. If the event is canceled, a call end message will be displayed. If the receiver accepts the call, 2 new windows will be opened in sender and receivers. If it is a group call, the sender will automatically enter the room and wait for the others. Below is the flow chart of join video room events.



In the video call part, we mainly use 3 libraries, socket.io, WebRTC, and mediasoup to design the SFU architecture as shown below.

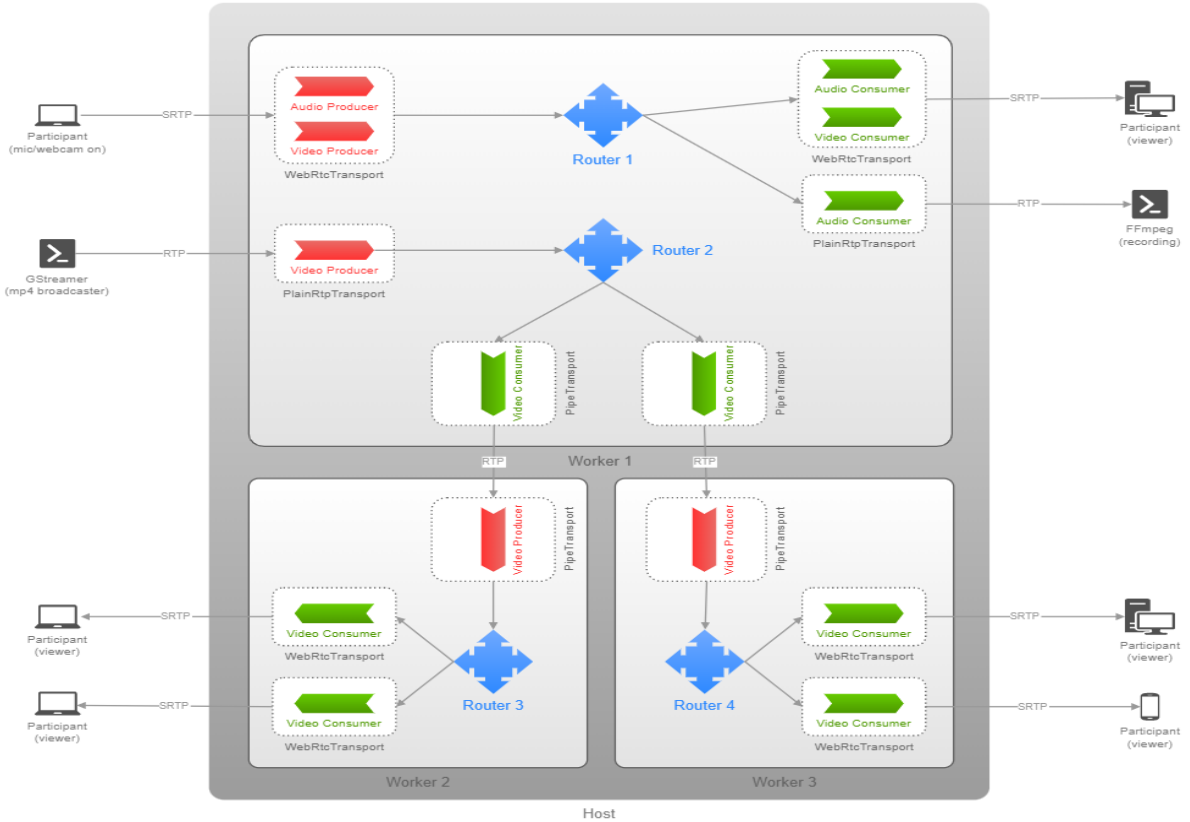


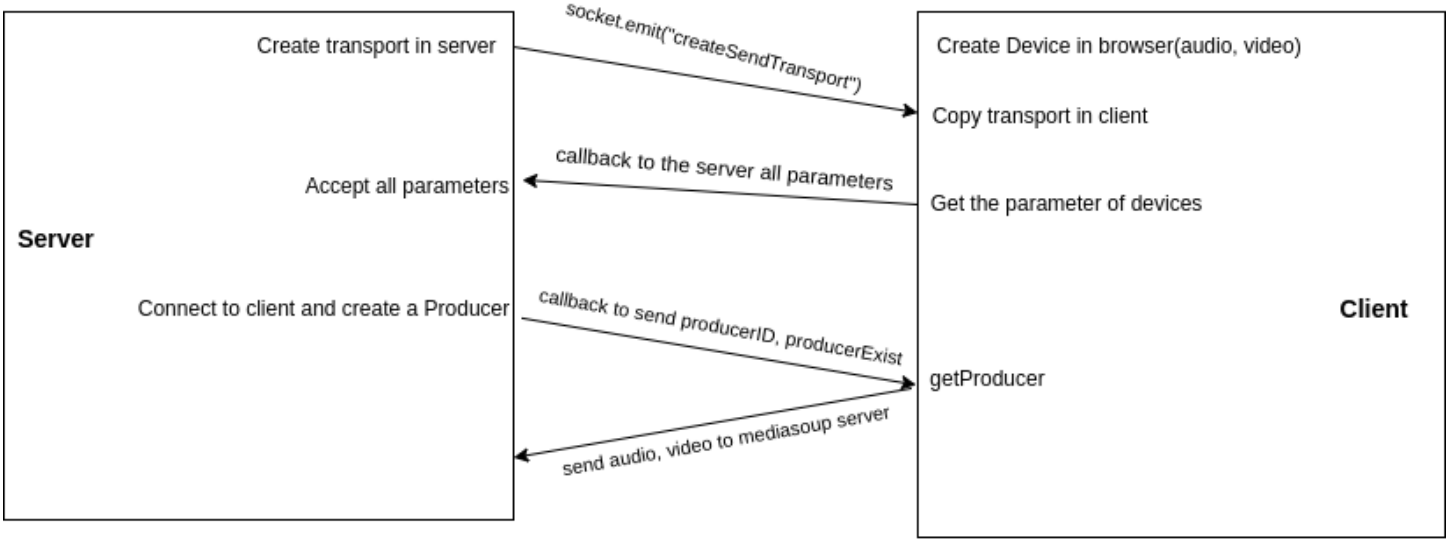
- In which socket.io plays the role of listening to events from both client and server sides to perform tasks such as notifying the server that the client has access, notifying the client server to open a camera,....
- WebRTC listens to IPs from 2 browsers using ICE candidates then encapsulates video and audio from users into packages.
- Mediasoup acts as a media server that speeds up the transport of media packages from WebRTC and to all clients.

Explain detail about the flow of video call:

- When joining a room, a client creates producers and receives consumers of other clients in the room. Producers contain media such as audio, video, and share screen of a client. Consumers are Producers of other clients, view only, not editable.
- To create Producers in a client following these steps:
  - Listen and create device directly in browser by mediaDevices.getUserMedia in webRTC.
  - Create a transport in mediasoup server by createWebRtcTransport()
  - Client sends the parameter of their device to mediasoup server to request server to create a connection and also create a Producer.

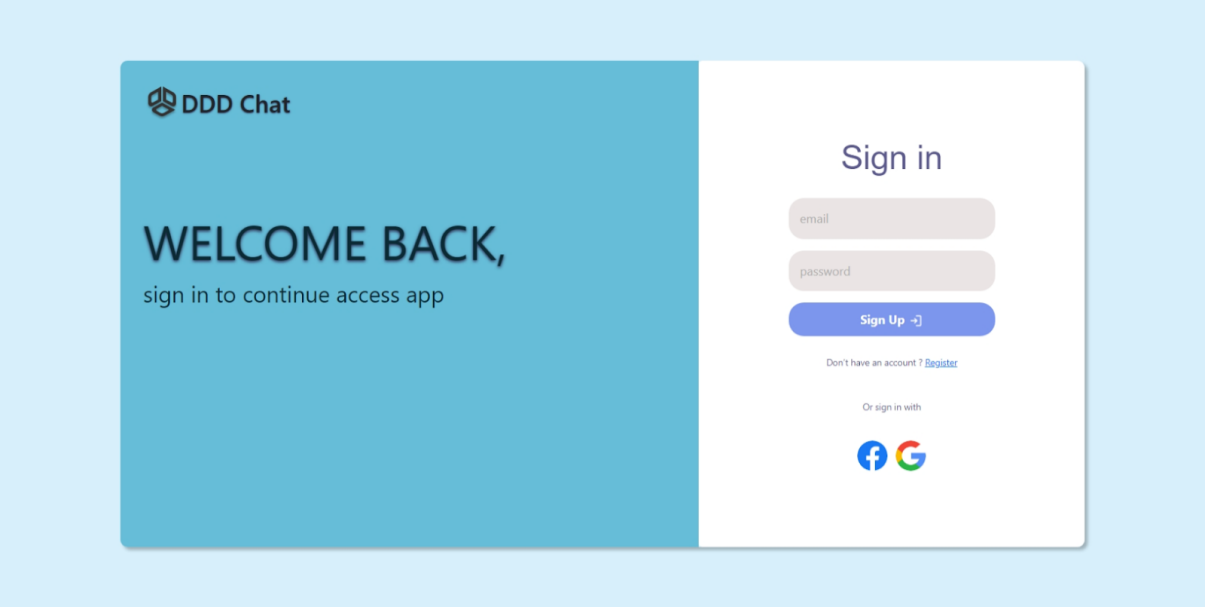
- Server connects and creates a Producer.
- Server uses callback to send *producerId* and *producerExist* back to client.
- If *producerExist* equal True, client getProducer and trigger an event to send media to server.
- Server converts Producer to a Consumer and also create transports and send to other clients.



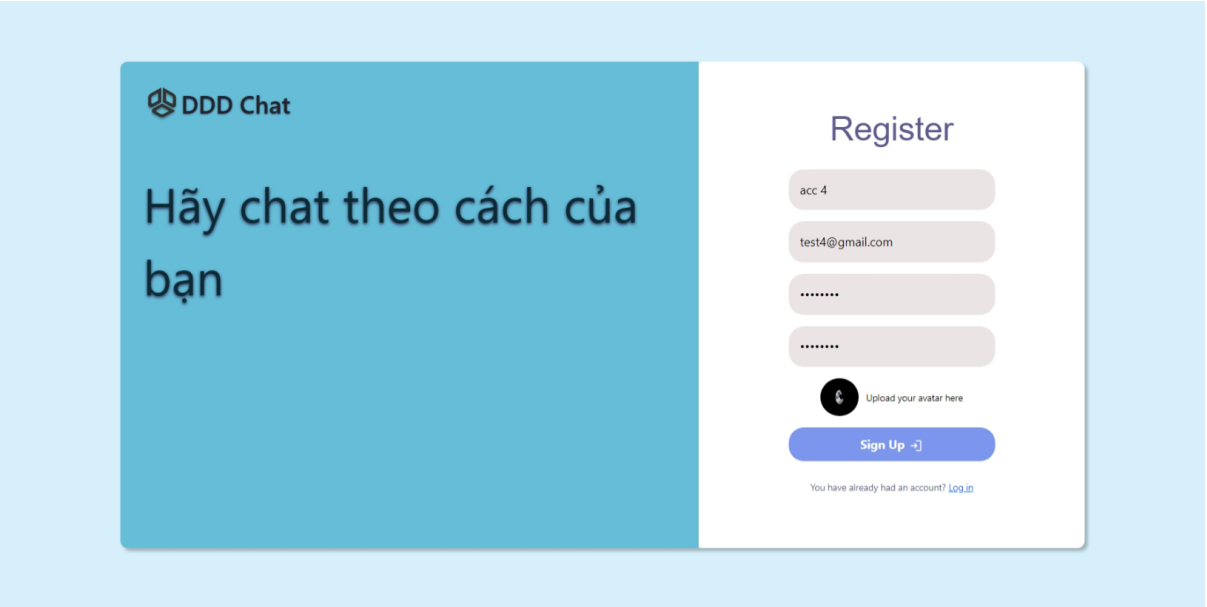


IV. Demonstration

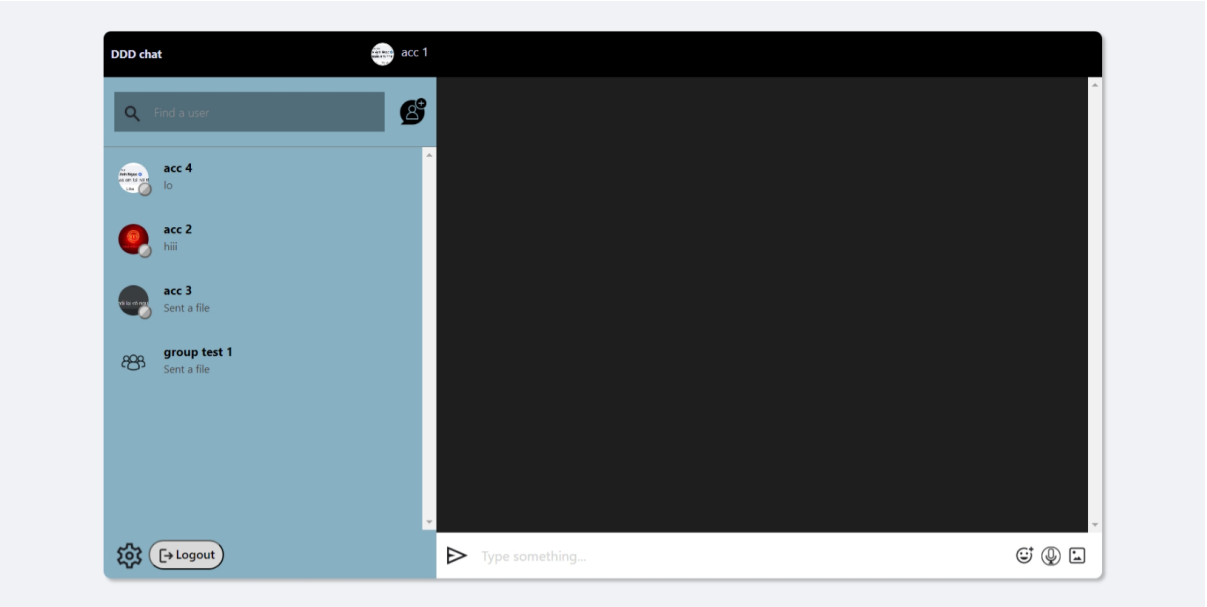
Login Page:



Register Page:

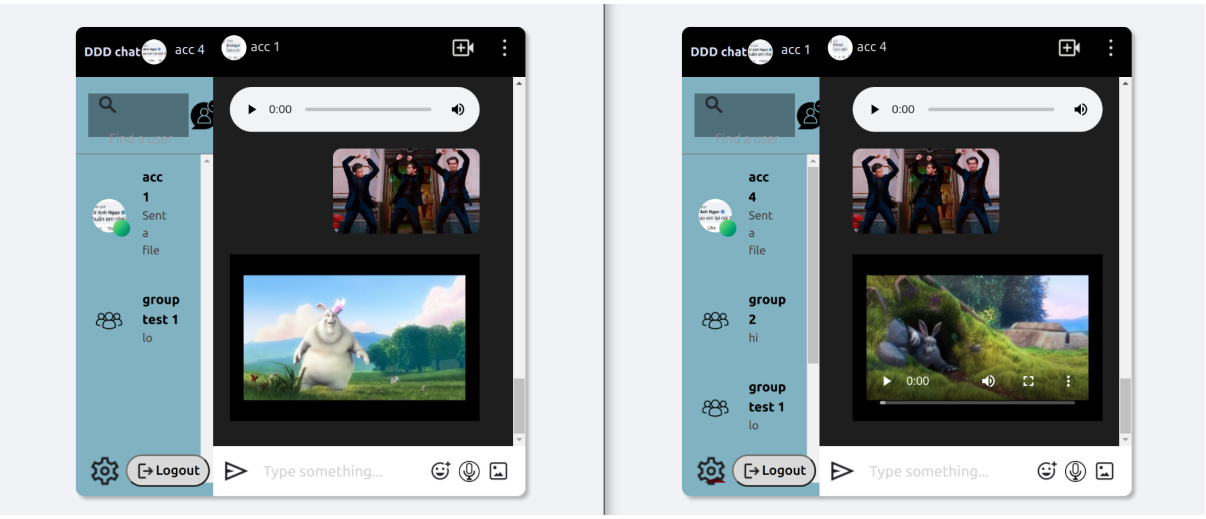


Home Page:

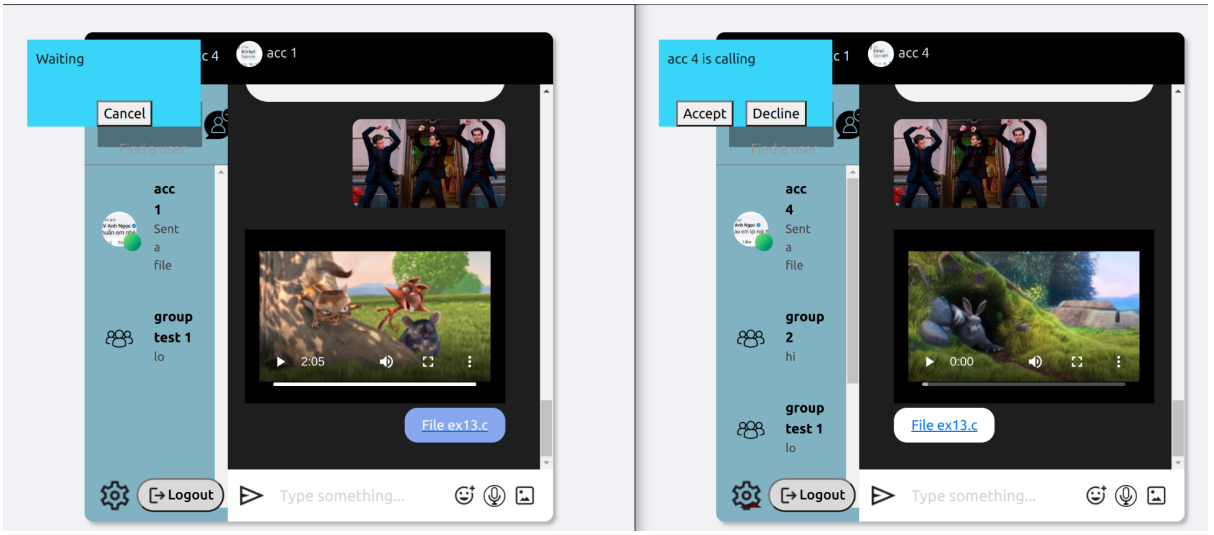


When 2 user is online, a tiny green tick will appear:

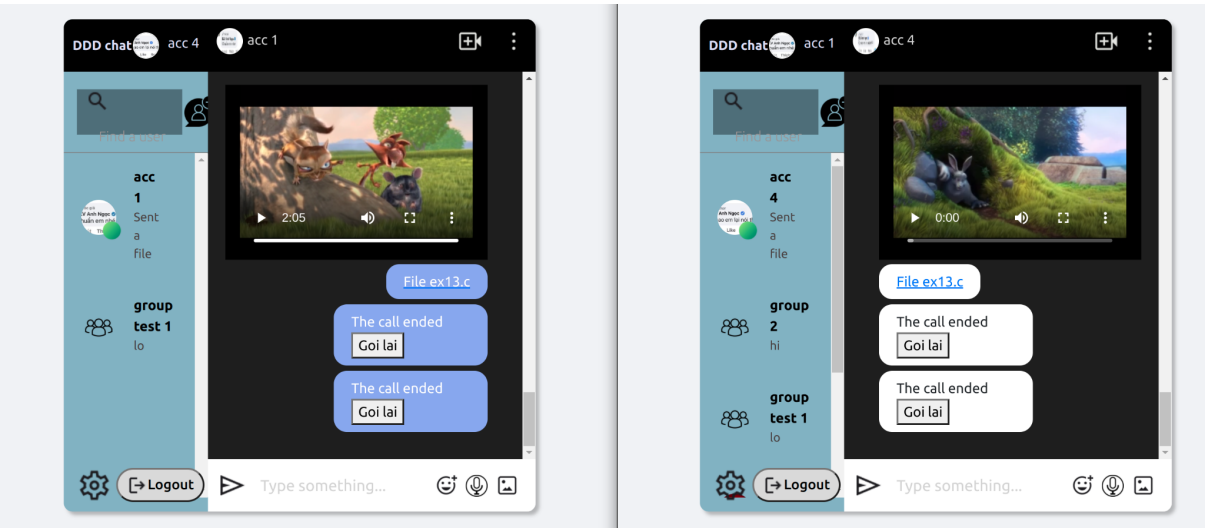
Send audio, image, video:



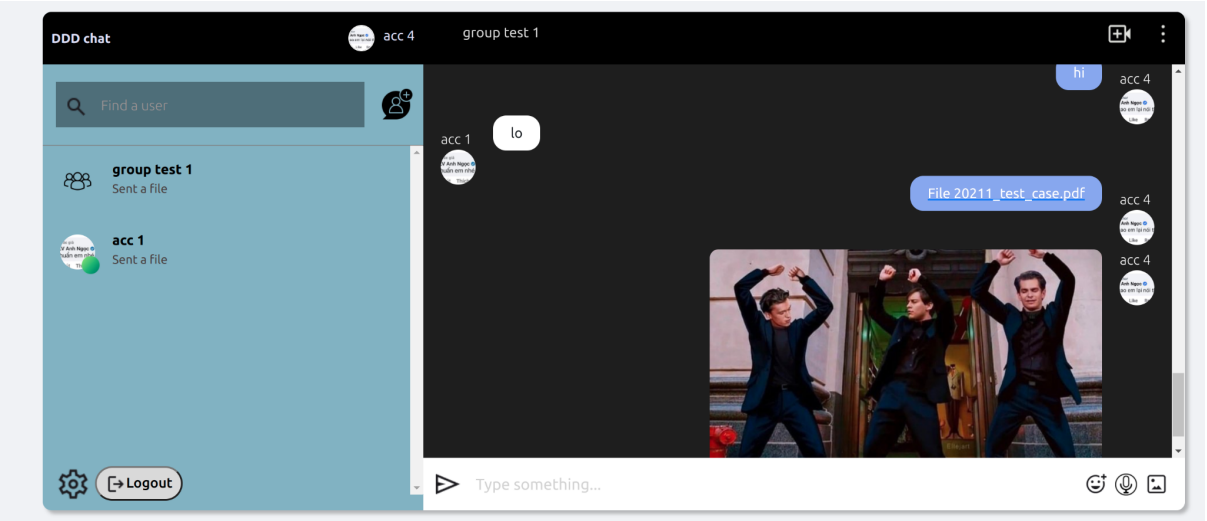
Notification when call:



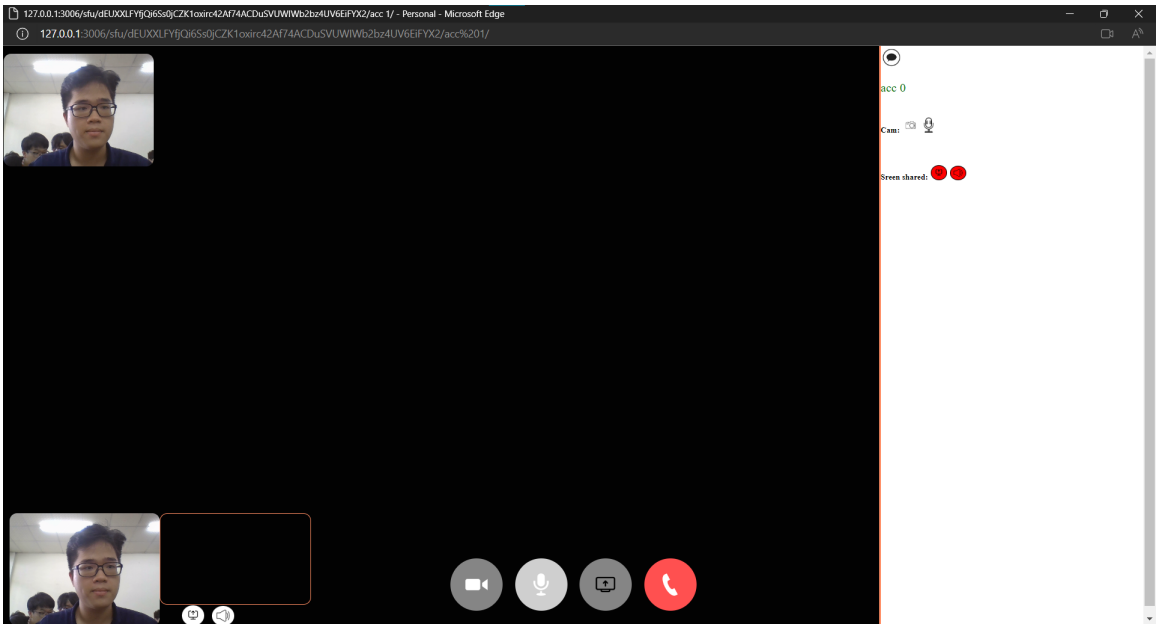
Message notifies an ended call



Group chat:


DDD chat
acc 4
group test 1
Find a user
group test 1
Sent a file
acc 1
Sent a file
Logout
hi
acc 4
acc 1
lo
File 20211_test_case.pdf
acc 4
acc 4
Type something...

Video Call:


127.0.0.1:3006/sfu/dEUXXLFYfjQi6Ss0jCZK1oxirc42Af74ACDuSVUWlWb2bz4UV6EiFYX2/acc 1/ - Personal - Microsoft Edge
127.0.0.1:3006/sfu/dEUXXLFYfjQi6Ss0jCZK1oxirc42Af74ACDuSVUWlWb2bz4UV6EiFYX2/acc%201/
acc 0
Cam:
Sreen shared:

V. Member contribution & Repository

All source code has been pushed to this GitHub repository: Github

Workload	**Member**
Planning and Gathering Requirements	Dang Dat, Ngoc Dang
Design UI	Dang Dat, Ngoc Dang
Frontend Development	Dang Dat, Thanh Dat
Login Feature	Dang Dat, Thanh Dat
Messaging Feature Development	Dang Dat
Set up database	Dang Dat
Video Call Development	Ngoc Dang
Connect Message and Video Call feature	Ngoc Dang, Dang Dat
Integration and Testing	Ngoc Dang, Dang Dat, Thanh Dat
Deployment	Ngoc Dang

VI. Improvement proposal

In this program, we face some bugs about the notification when we use React (the noti component is rendered many times). We will try to figure out the root of that problem and fix it in the future.
As for the video-call part, we will develop more such as applied AI for noise reduction, filter, sound quality, image quality, UI design. Finally, the security and optimization for the website needs to be improved

VII. Conclusion

This project helped us a lot in learning about many aspects of web development. Skills such as API writing, front-end design, back-end, web security were best applied in this project. Thanks to the efforts of all members, we can learn to work with each other and finish this project on time.

VIII. References

Firebase Documentation (google.com)

Introduction | Socket.IO

mediasoup :: Examples

mediasoup :: Documentation

Chat App using React and Firebase | Realtime Private Chat Application - YouTube