

---

# CS422 - Final Project - Report

## **Super Group Chat**

---

20125119 - Nguyễn Minh Trí

20125074 - Dương Hoàng Huy

20125107 - Hồ Văn Quân

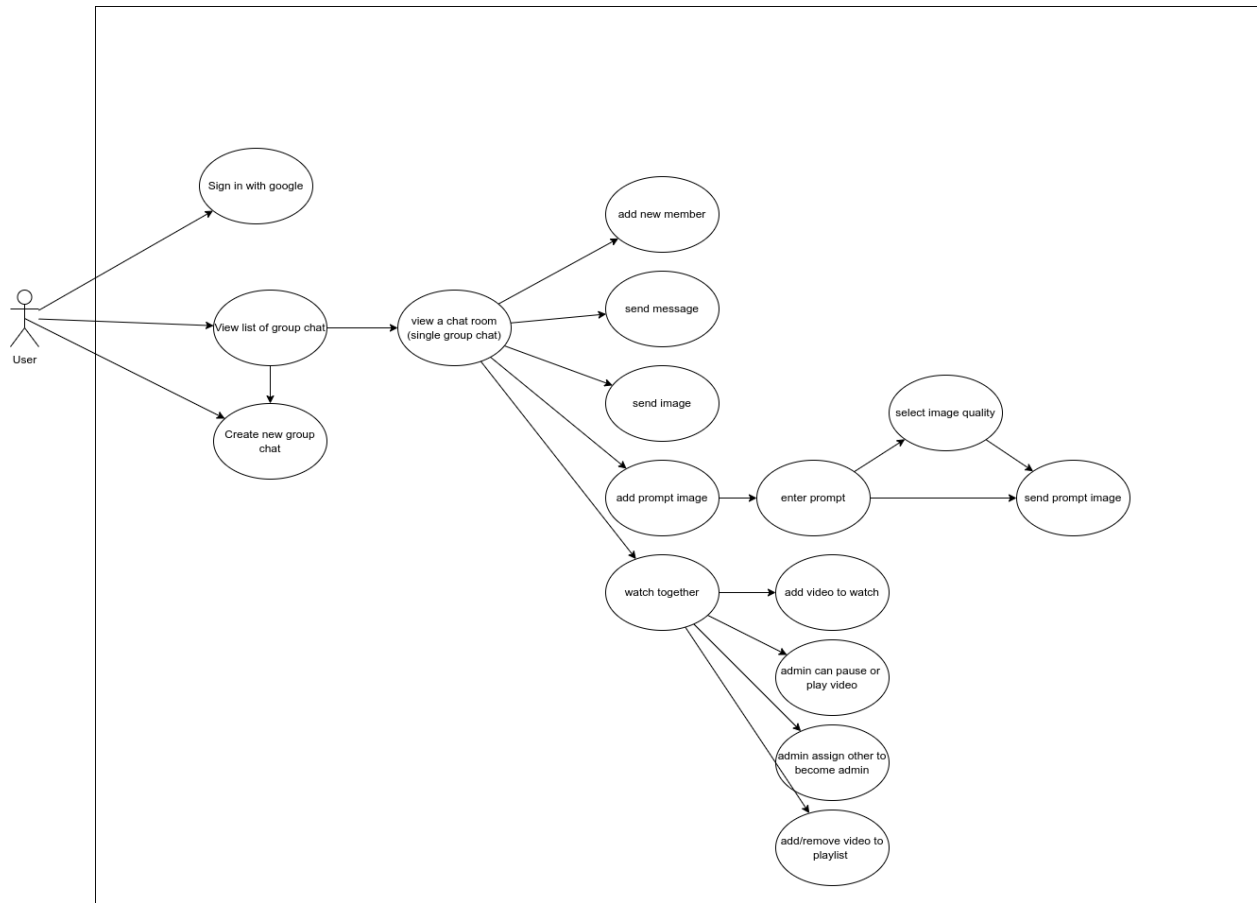
20125108 - Đào Nhật Quang

<b>I. Requirement</b>	<b>1</b>
<b>II. Use-cases</b>	<b>2</b>
<b>III. Class diagram</b>	<b>2</b>
<b>IV. Advanced techniques</b>	<b>3</b>
1. Stable Diffusion - handle return image result	3
2. Observer pattern for event-driven system using web socket	3
<b>V. Main Features</b>	<b>5</b>
1. Sign in with google	5
2. Chat Room	6
3. Generate Image by Stable Diffusion	7
4. Watching Youtube Video Synchronously	9

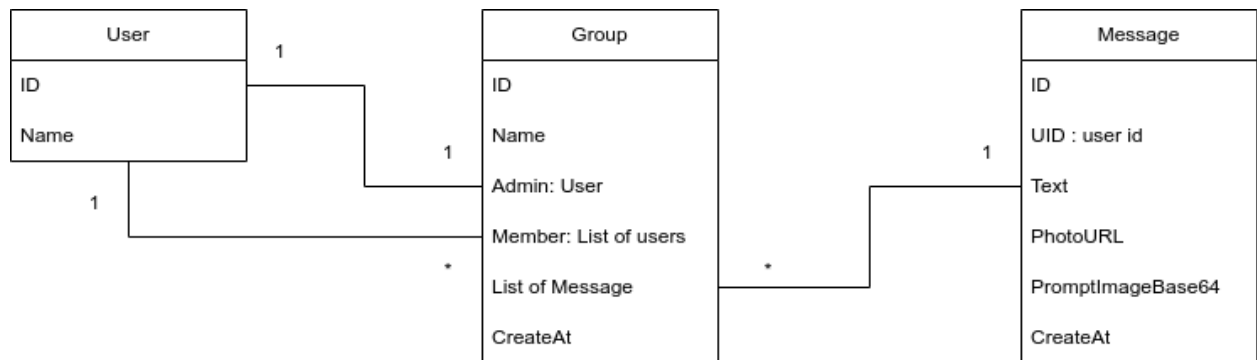
## I. Requirement

A website that allows users to create or participate in a group chat, send messages, images or prompt their own image with the Stable Diffusion model. Moreover they can watch with their friends through the Watch2gether system.

## II. Use-cases



## III. Class diagram



## IV. Advanced techniques

### 1. Stable Diffusion - handle return image result

- Because the speed of using our laptop CPU is very slow (20s/epoch) when compare with GPU on the internet (approximately 1s/epoch)

We allow 2 options:

1. Users can wait until the image is completely generated and send the message with it.
2. Users can prompt the image (usually with a high quality option), while waiting, they can send the image beforehand, later the system will automatically update the image into that message.

In this case, we use the technique to handle the result (the hard part is check whether it is the case user send message beforehand or not, and store the messRef, after the server finish, if we handle the update also inside the get image result (call the api) it will not got the correct messRef(null) because of Asynchronous of Javascript function

1. Store the messageRef that user send (for later add image to it, check whether it in the case or not)
2. When Stable Diffusion Server return the base64 image, with the messageRef, update it with new image data\

### 2. Observer pattern for event-driven system using web socket

- To synchronize between multiple clients through server, they need to agree on a set of necessary events:

```
const EVENTS = Object.freeze({
  ROOM: 'room',
  SYNCHRONIZE: 'sync',
  VIDEO_CONTROLLER: 'control',
  ONLINE: 'online'
});
```

- And their actions:

```
const ACTIONS = Object.freeze({
  // room event
  JOIN: 'join',
  LEAVE: 'leave',
```

```

// control/authorization event
ASSIGN: 'assign',
REMOVE: 'remove'

// broadcast the online list
ONLINE_USERS_LIST: 'onlineuserslist',
});

```

- Both client & server have the **event handler** to classify them into the correct **event controller**. Depending on each type of event, it can be a 1-many (i.e notify a new user) or 1-1 (i.e assign controller to a specific user):

```

// Server
const handleMessage = (req, ws) => {
  // console.log('Incoming request', req)
  let event = req.event;
  if(event === EVENTS.ROOM)
    handleRoomEvent(req, ws);
  else if(event === EVENTS.SYNCHRONIZE)
    handleSyncEvent(req, ws);
  else if(event === EVENTS.VIDEO_CONTROLLER)
    handleControllerEvent(req, ws);
}

```

```

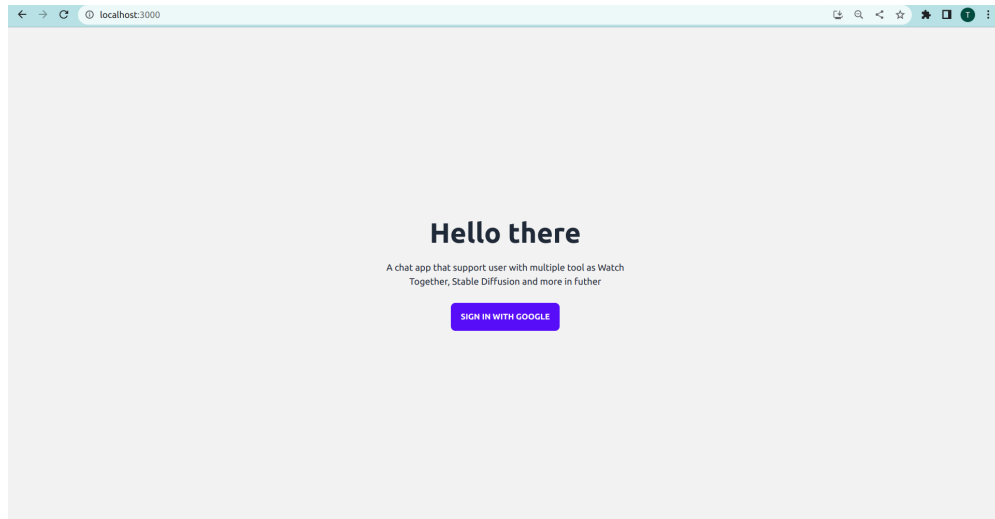
// Client
const handleResponse = res => {
  if(res.event == 'sync'){
    updateVideo(res)
  } else if(res.event == 'control'){
    setController(res.action)
  } else if (res.event == 'online'){
    setOnlineList(res.onlineList)
  }
}

```

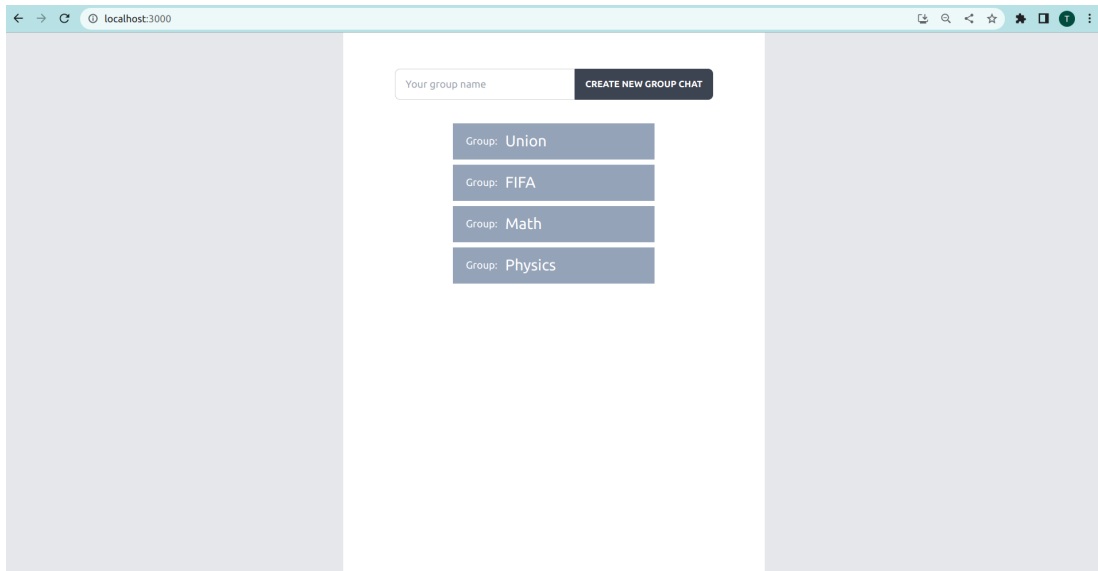
## V. Main Features

### 1. Sign in with google

- User sign in with their Google account (though **Firebase authentication**)

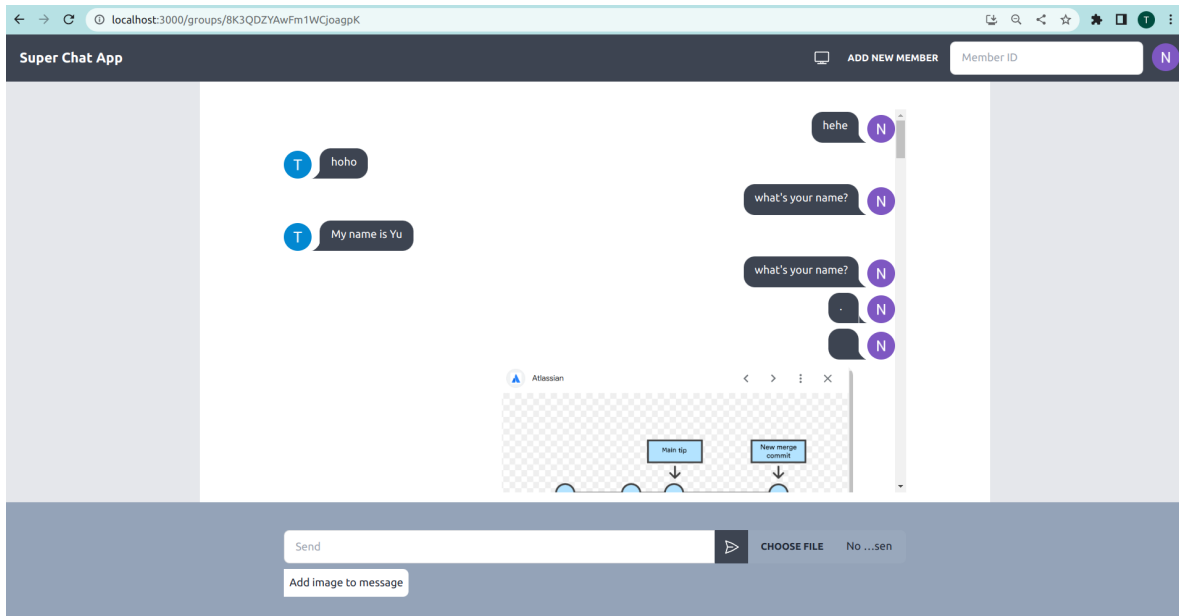


*After sign in user will go to the homepage site*



*Show list of group that user created or invited into*

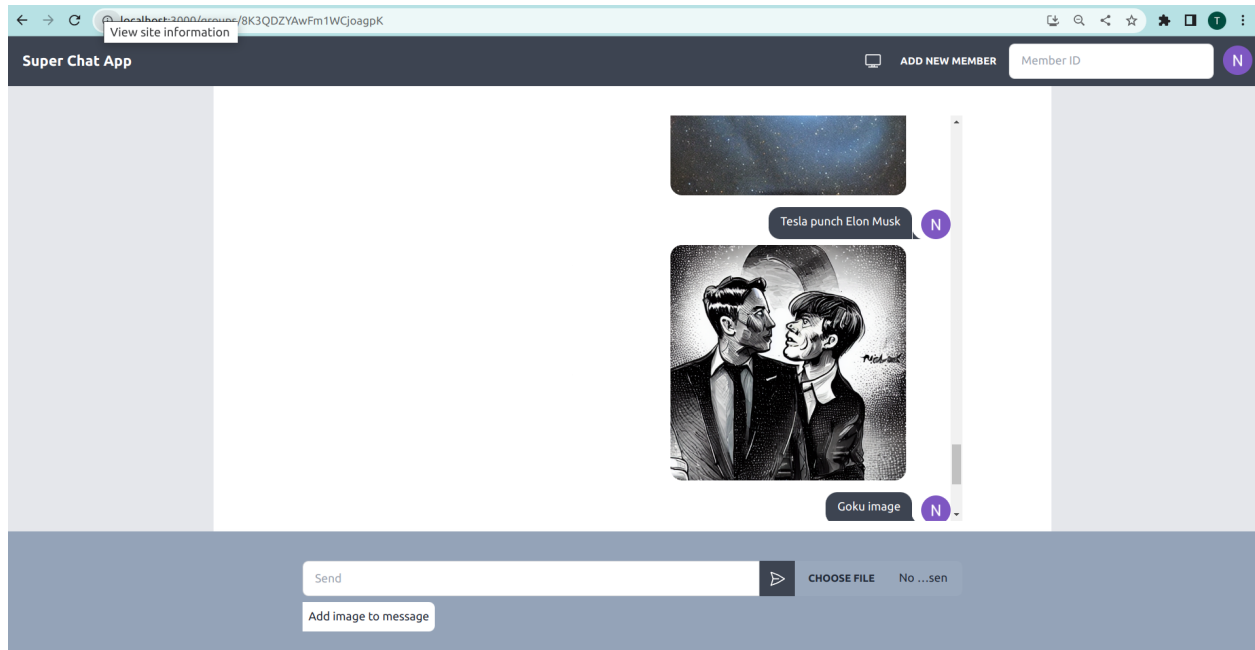
## 2. Chat Room



In this chat room, a single group chat include users, who **are invited** can:

- Send messages
- Send image
- Add prompt image to message

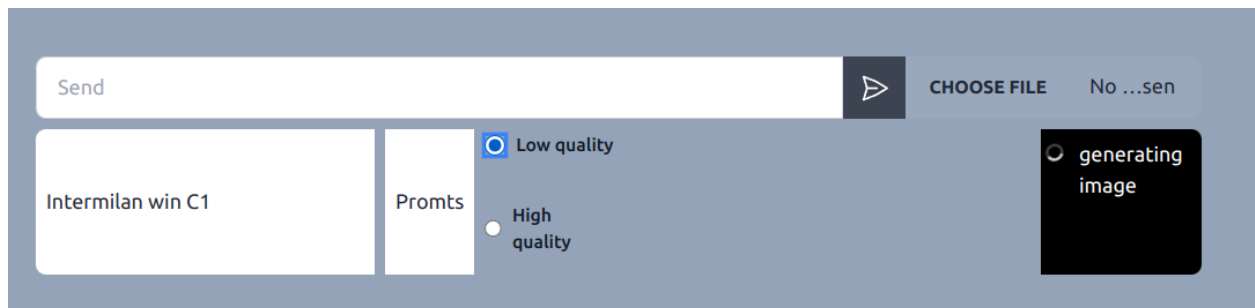
### 3. Generate Image by Stable Diffusion



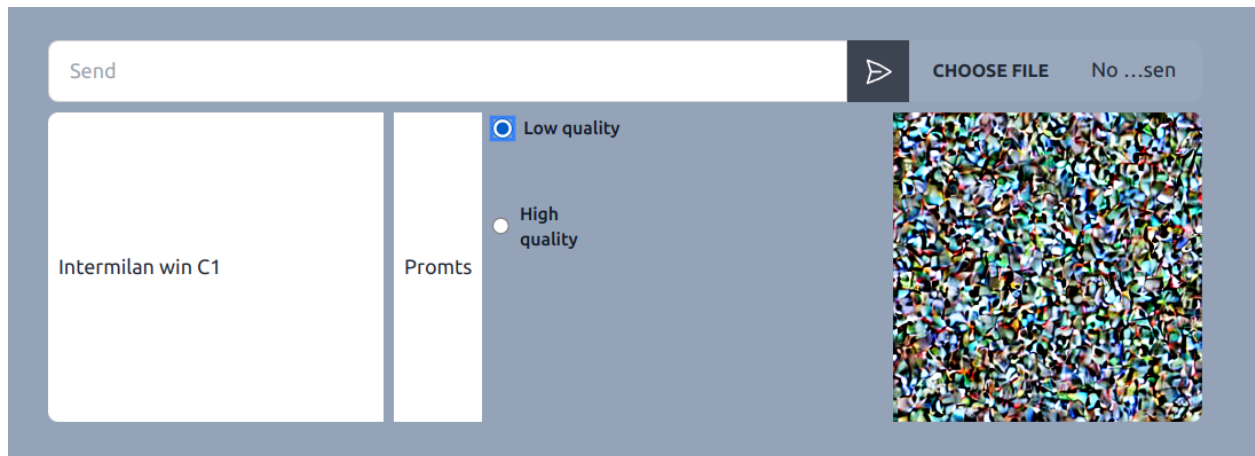
*Result*

#### After select Add image to message

- System will show the input for the user to enter the test, and select Prompts and select the quality to run.





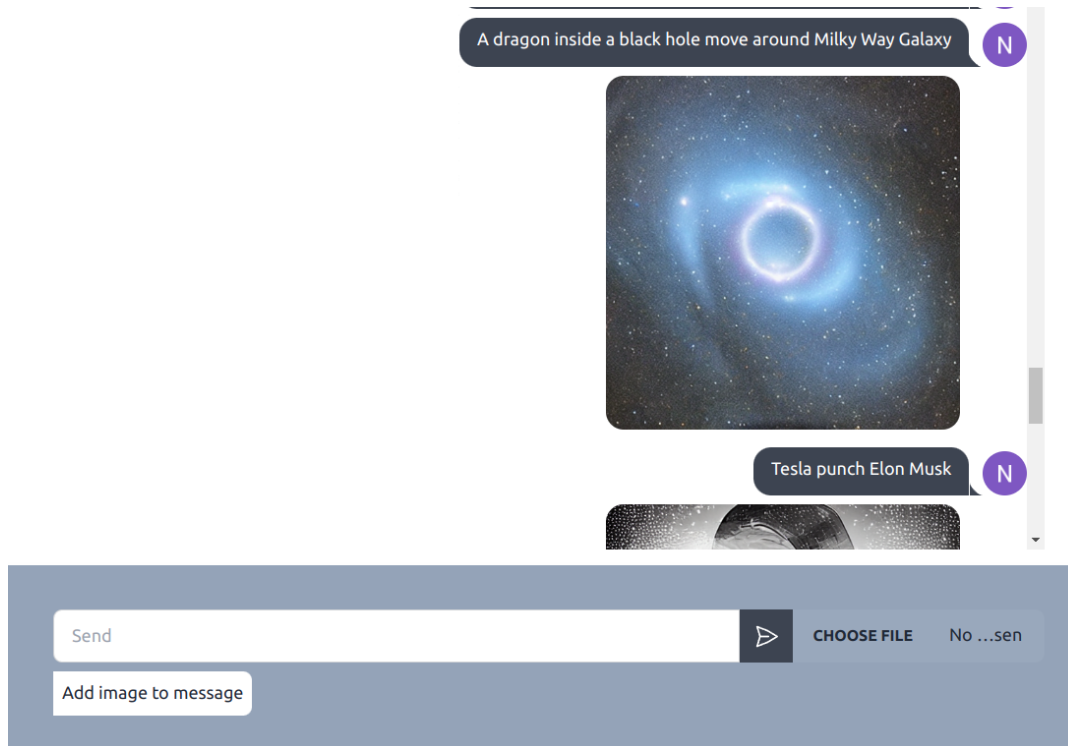


**After that, the user can wait until the image finishes, then send a message with it**

(Or send the message beforehand. After the prompted image finishes, it will auto update to the image.)

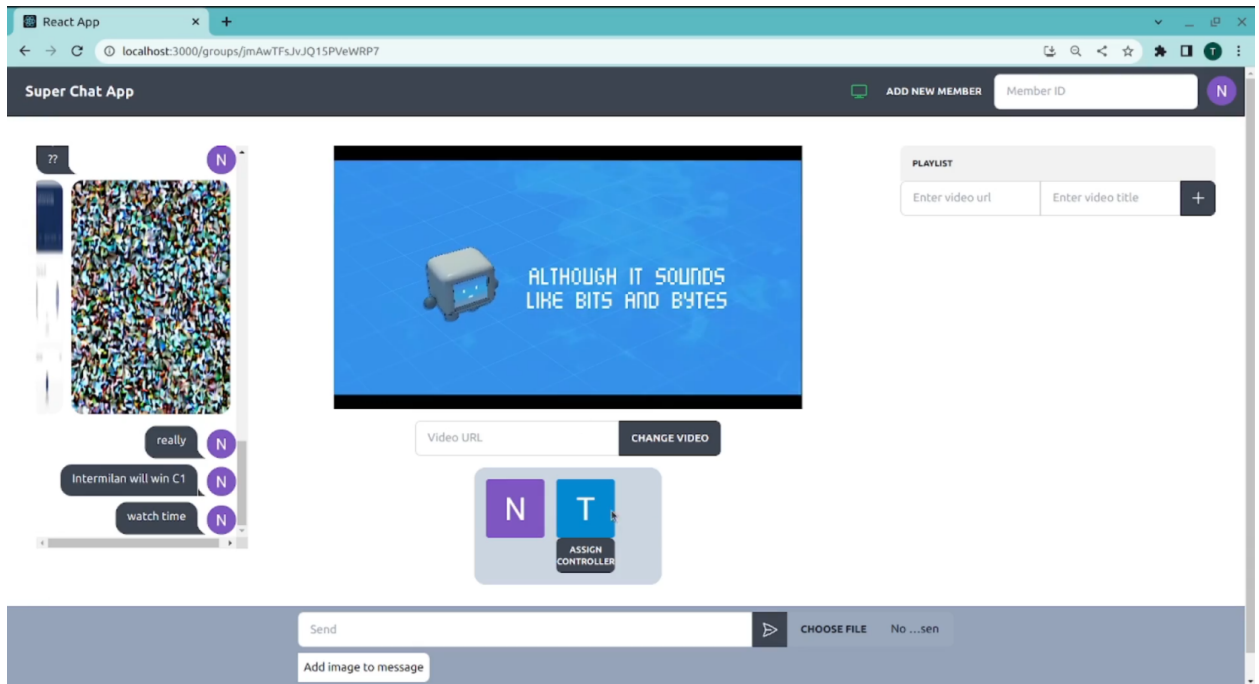


If you choose high quality image it will take much time but the image is more quality

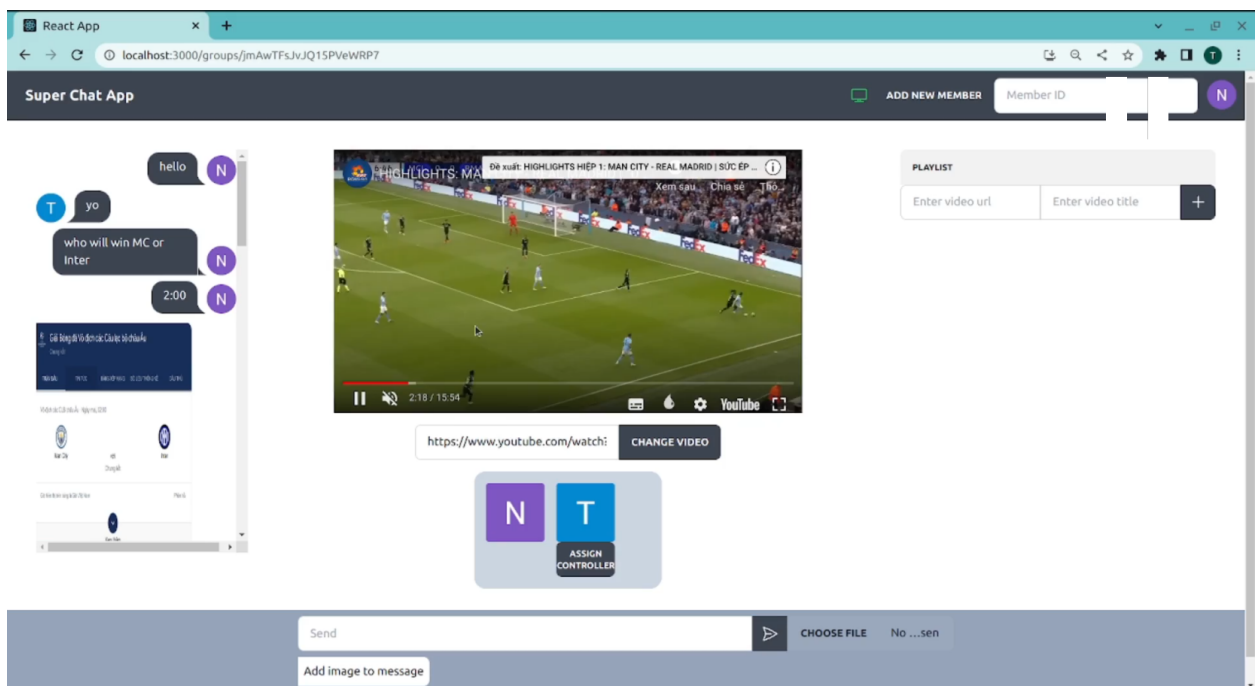


#### 4. Watching Youtube Video Synchronously

- Synchronize the video's state, including pausing, seeking, directly changing a video using a YouTube URL or from the playlist.



- Only a host and controllers can synchronize the video's state among users in the watching room.




- The host can assign or remove a controller for any member in the watching room.
- Add or remove a video from the playlist.

DD NEW MEMBER

Member ID

PLAYLIST



C1

×

Enter video url

Enter video title

+