Report of telemedicine video module

1 Pipeline

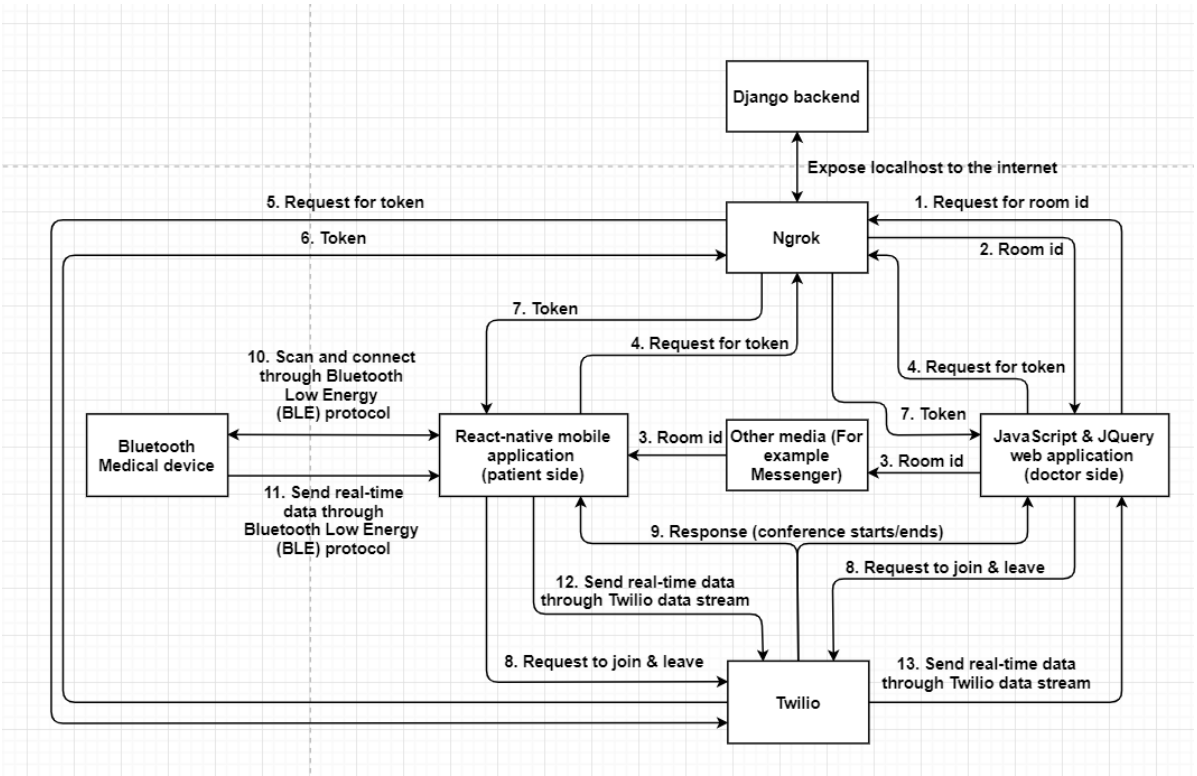

Figure 17: Structure of the system

The system will be constructed by 5 parts:

- Bluetooth telemedicine devices.
- The patient side react-native mobile application on a smartphone/laptop/computer.
- The doctor side Web application.
- A Django backend server.
- Twilio cloud.

The Bluetooth telemedicine device could be any wireless devices using BLE protocol, for example Bluetooth heart rate monitors, Bluetooth blood pressure monitors and Bluetooth

stethoscopes. With the help of these instruments, the system can monitor real-time data like heart rate, blood pressure and auscultation results. This system can be used by smartphones, laptops, or computers as long as they have built-in microphones and cameras and some most used browsers like Foxfire and Chrome. For devices without a built-in camera, the system is also compatible with external microphones and cameras, allowing users to choose the video and audio devices used for video conferences. A Twilio account will be used by the web server to access the services provided by Twilio cloud. The frontend webpages will be implemented using JavaScript and jQuery.

The process of a medical video conferencing between a patient and a doctor:

1. A Doctor uses the browser on his smartphone/laptop/computer to access the web application (a website). Then click a “Create room” button to ask the backend to create a unique room name. The patient side mobile APP does not have this button, only doctors can create a new meeting room.
2. The backend creates a unique room name for the doctor, shows it on the website frontend interface and stores it in the live room name list.
3. The doctor can inform his patients through other social media like email, Messenger or phone.
4. The doctor input the room name and his username in the website frontend and click a “Join call” button, the patient can do the same on the patient side react-native APP. Then the web application and the mobile application will ask the backend for access tokens to enter the meeting room.
5. The backend will ask the Twilio cloud to create access tokens for the doctor and the patient using its Twilio API key.
6. After checking the validity of the API key possessed by the backend server, Twilio cloud will create access tokens and send them back to the server.
7. The server will send the tokens to the doctor side web application and the patient side mobile application.
8. After getting the tokens, patient and doctor side applications will automatically use them to join the meeting room. The token can also be used when leave the meeting room.

9. Twilio establishes the connection of video, audio and data stream after getting the join room request, the meeting starts.

10. When the doctor and the patient are both in the room, the patient could click the “scan for device” button on the mobile application to scan for the Bluetooth medical device he wants to use. All scanned device names or their UUIDs will be shown in the application. After discovering the wanted device, the patient can click on its name to establish a connection with the device. the patient can also choose the service and characteristic of the device he wants to use.

11. After setting up the connection and selected the wanted service and characteristic, the application can start receiving the real-time data sent from the connected medical device.

12. The patient side application will send the real-time medical data to Twilio cloud, which is maintaining the data stream for the video conference. The medical data will be encrypted.

13. The Twilio cloud will forward the medical data to the doctor side web application though data stream of the video conference. The data will be shown in the frontend interface of the doctor side application to allow the doctor provide diagnoses and instructions for the patient.

2 Product

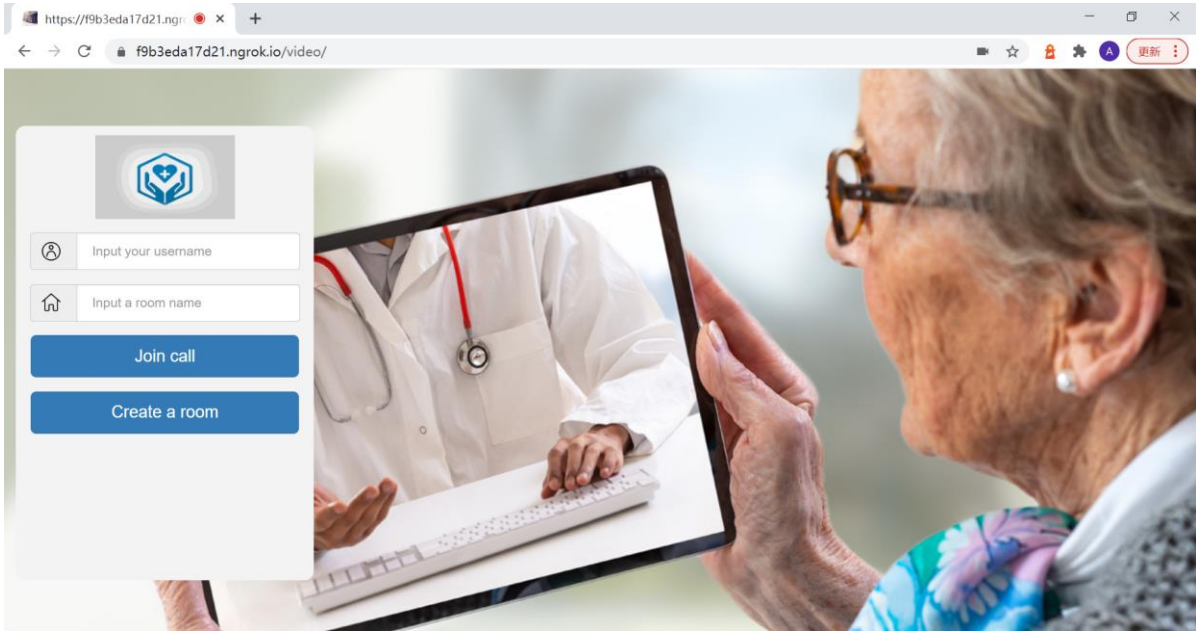

Figure 18: The join room page of the doctor side application

Doctors can open their browser and type in the URL (https://5883372ce004.ngrok/video in this case) to access the doctor-side web application. For now, the system does not have a fixed URL or domain name because it is using ngrok without a paid account, it can be improved in the future.

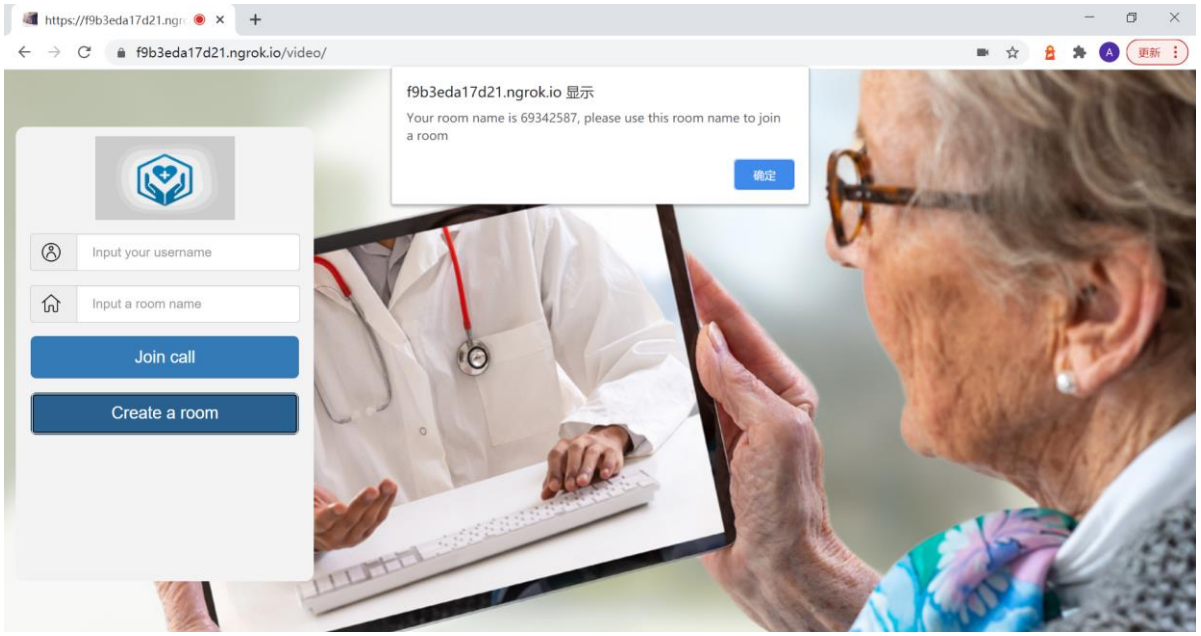

Figure 19: Clicking "Create a room" button in the doctor side application

Doctors can click the "Create a room" button to let the server generate unique names for them and store created names in the system. Because each Twilio room is identified by its name, so doctors are not allowed to create room name by themselves to eliminate duplication.

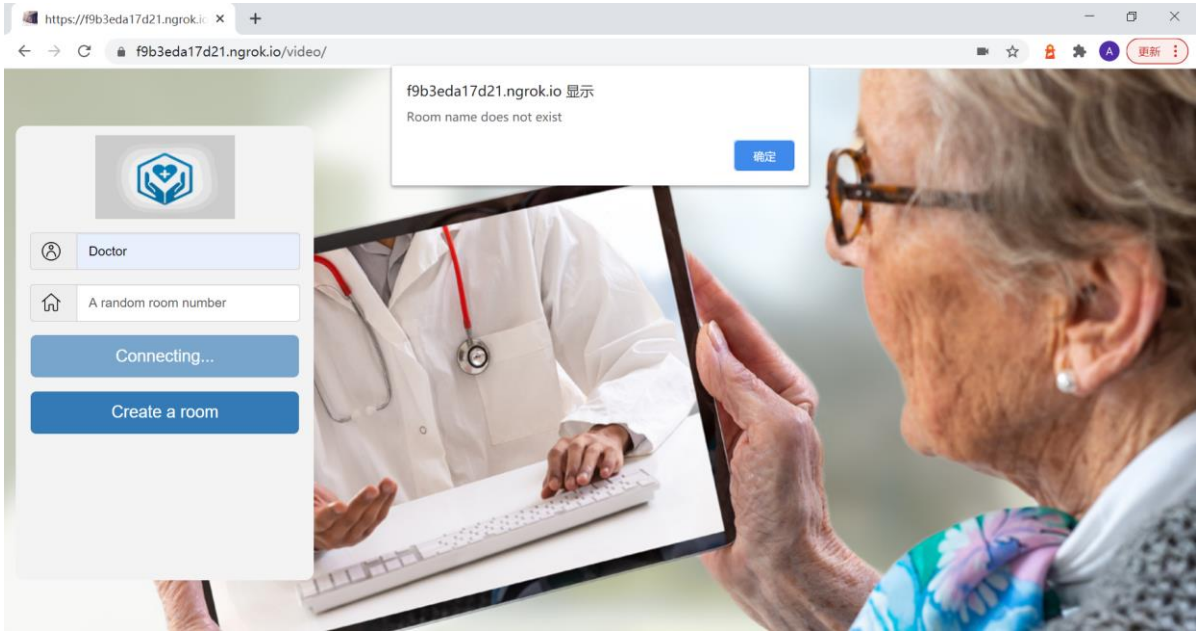

Figure 20: Inputting a random room number in the doctor side application

If a not stored room name is used, then the application will not join the room but report an error.

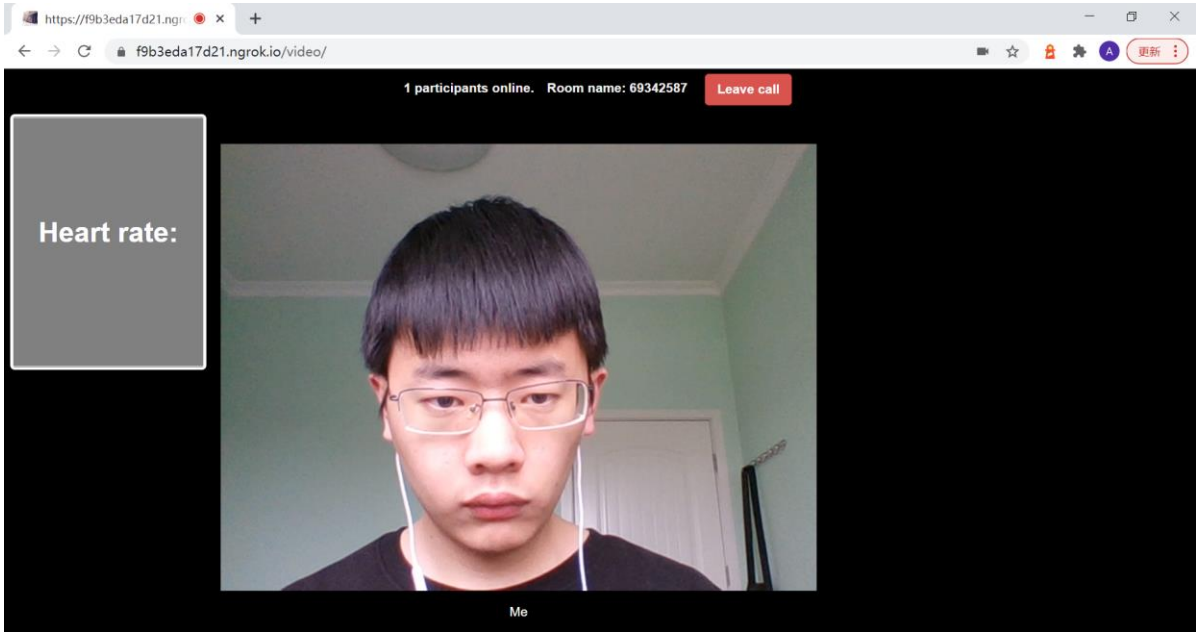

Figure 21: After joining a just created video room in the doctor side application

After joining the room, doctors can see themselves and wait for patients to join in. They have to tell the room name to their patients through other social media channels like email or phones. (it works like Zoom)

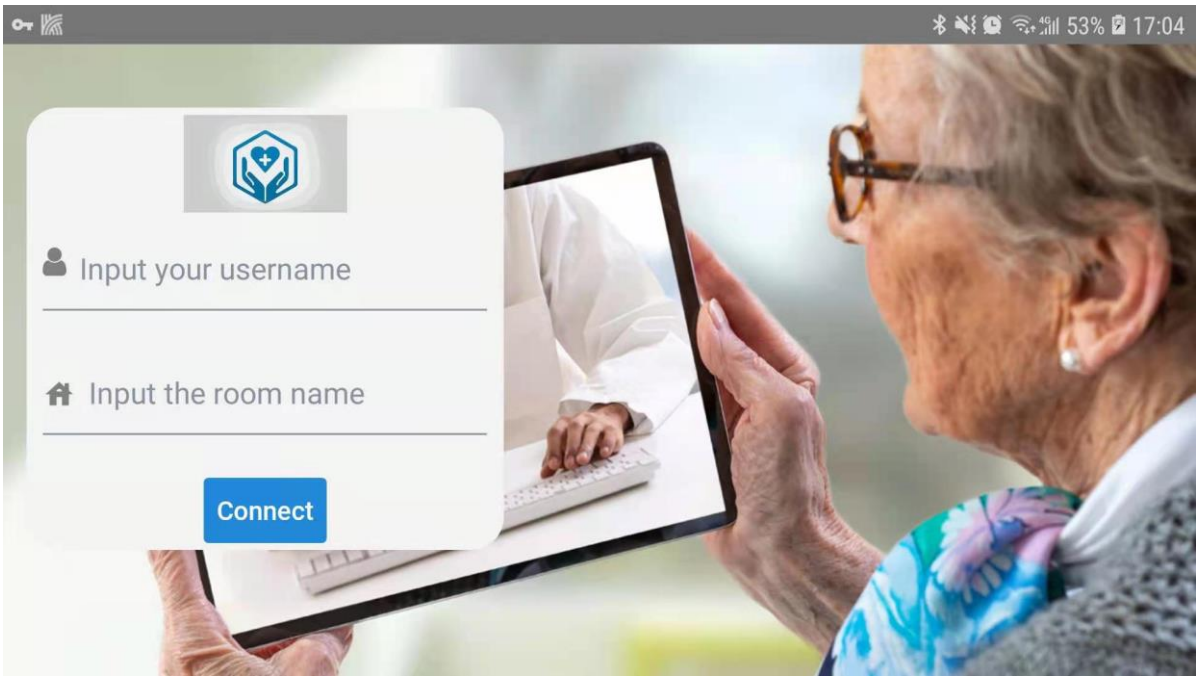

Figure 22: The join room page of the patient side application

Patients can access the patient-side mobile application from their phones or iPads. They do not have the “Create a room” button because only doctors are allowed to create rooms, patients can get room names from their doctors after they create the rooms.

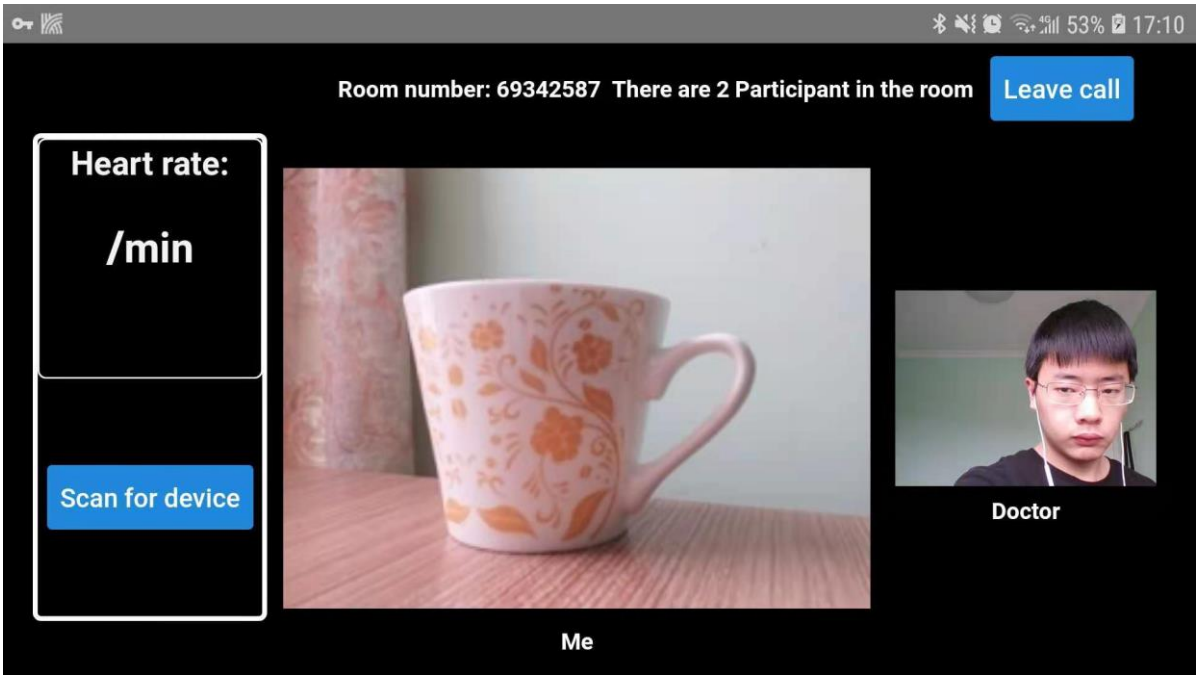

Figure 23: After joining a video room in the patient side application

Figure 24: Adjusting screen size in the patient side application

After joining the video conferencing room, patients can see their doctors' real-time image. They can touch the image of themselves or their doctor to make it bigger or smaller. The reason for not making the two screens the same size is to allow the user to resize the screen as

needed to get a clearer image.

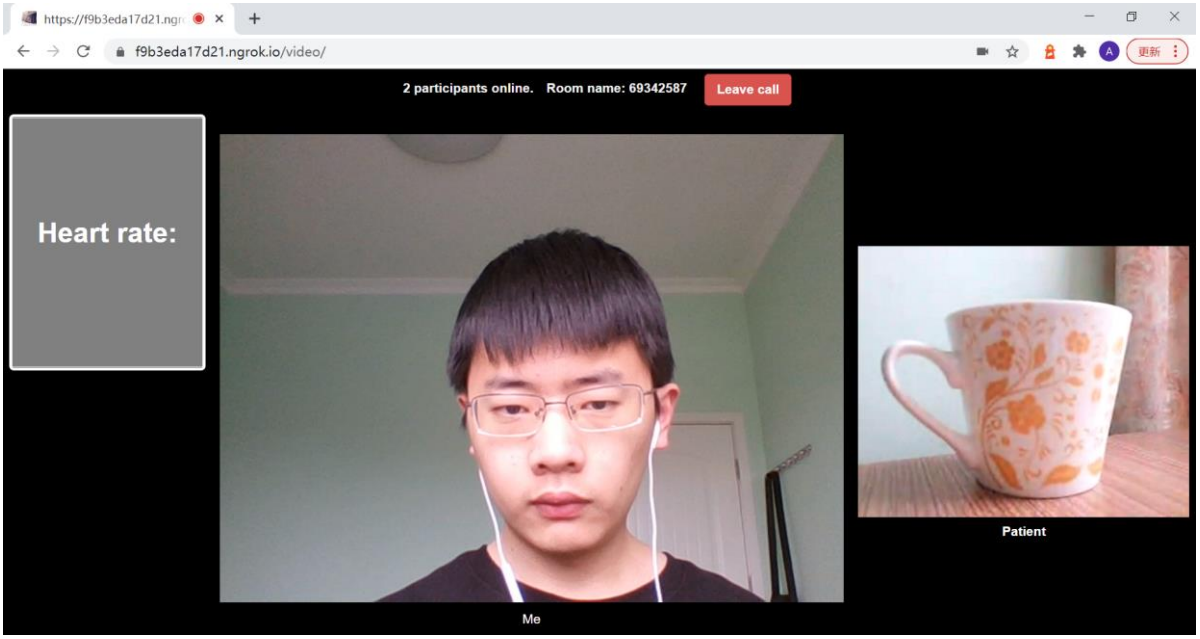

Figure 25: After patient joining the room in the doctor side application

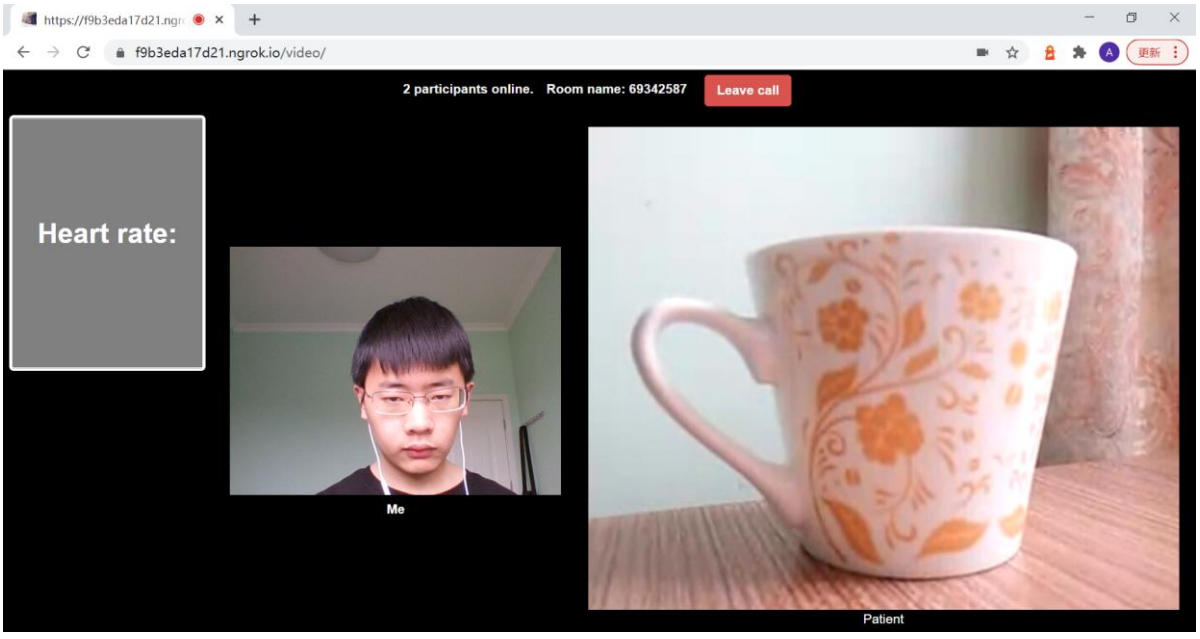

Figure 26: Adjusting screen size in the doctor side application

The doctor-side application also has the same screen adjusting ability.

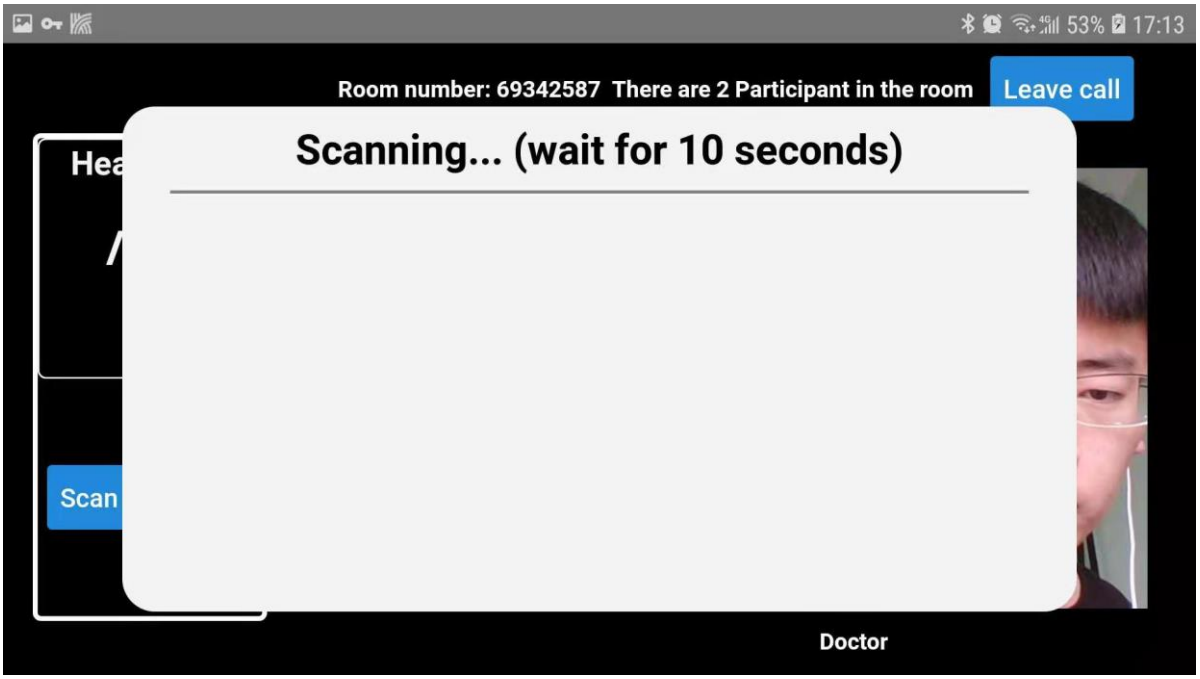

Figure 27: Clicking "Scan for device" button in the patient side application

When both the doctor and the patient are in the room, the patient can start scanning for their Bluetooth medical devices. The app takes a 10-second scan after the patient clicks the button.

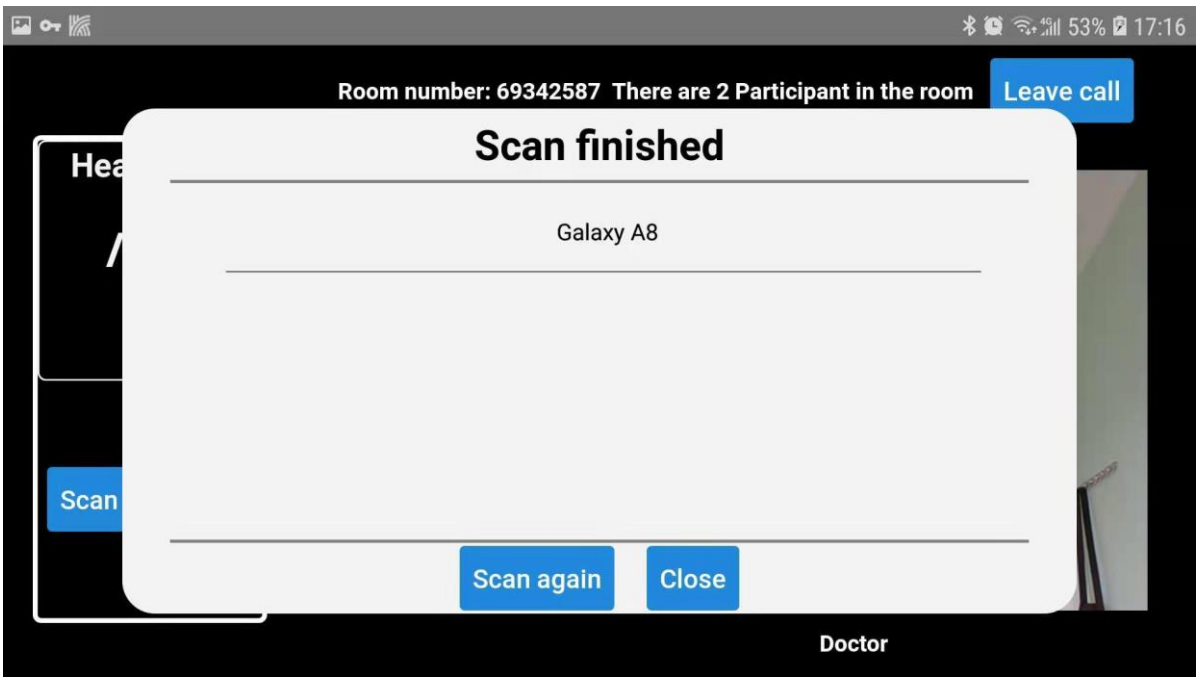

Figure 28: The scanning result of the patient side application

When the scanning is finished, all scanned devices name (or UUID if the device does not have a name) will be listed.

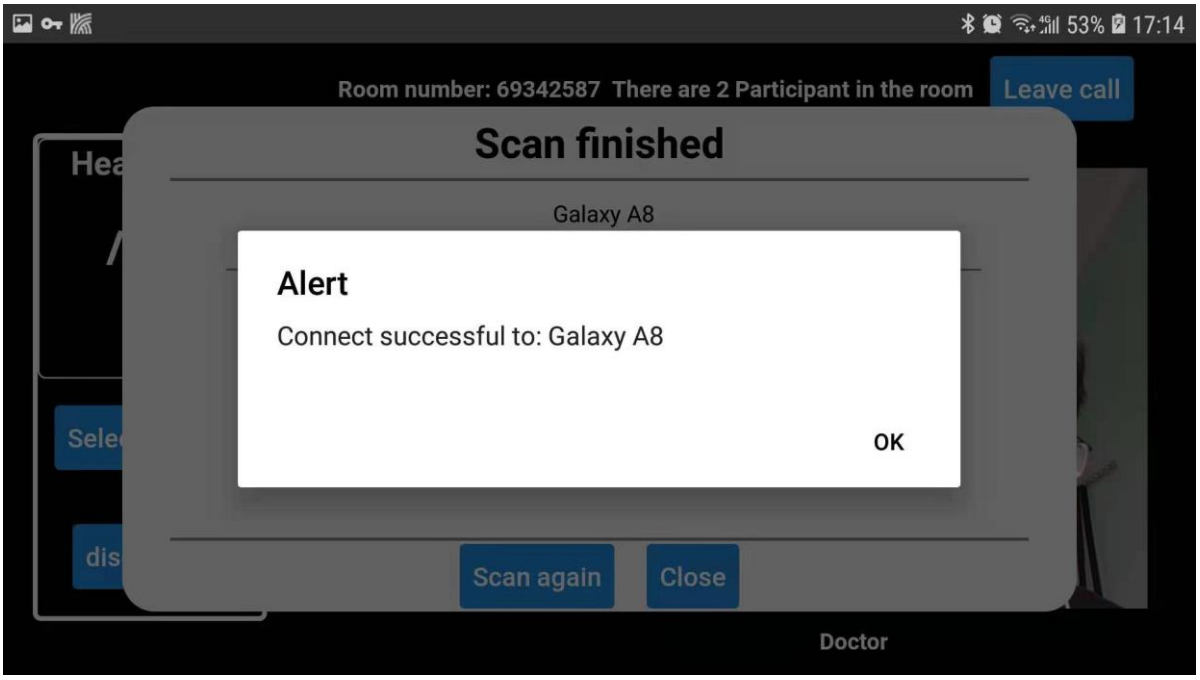

Figure 29: Connecting to a Bluetooth device in the patient side application

The patient can connect to the device by clicking on the listed device name.

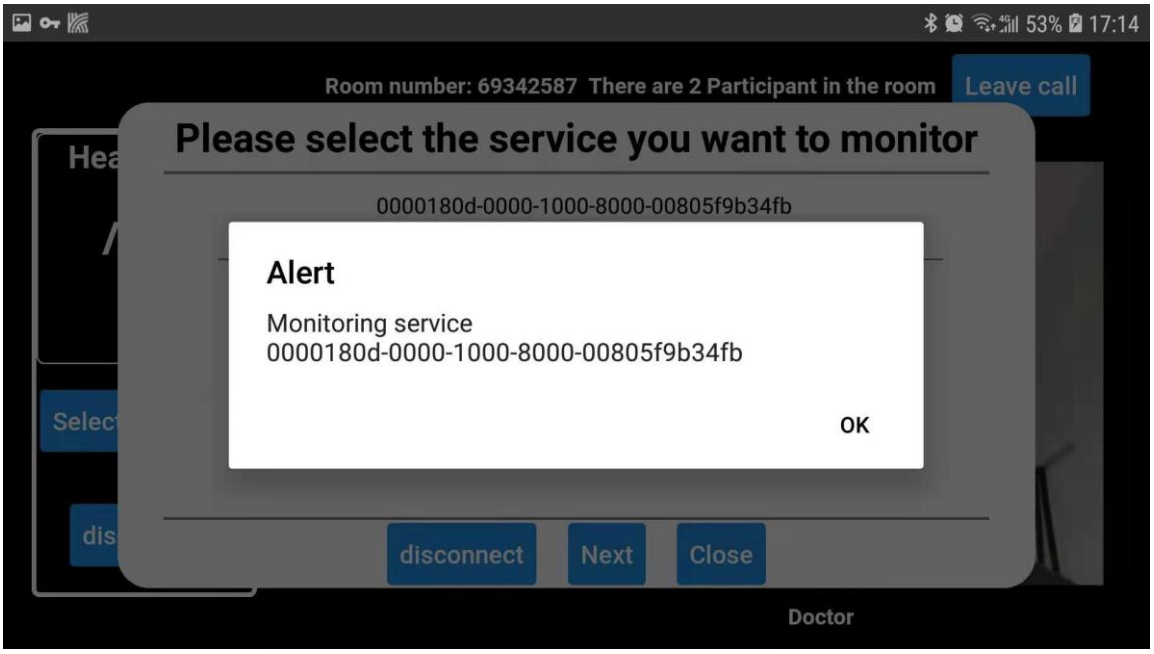

Figure 30: Selecting a BLE service in the patient side application

After connecting to the device, patients should then click the “Select a service” button to choose the service they want to use from the device. All services’ UUIDs will be listed, and patients can select a service by clicking on its UUID.

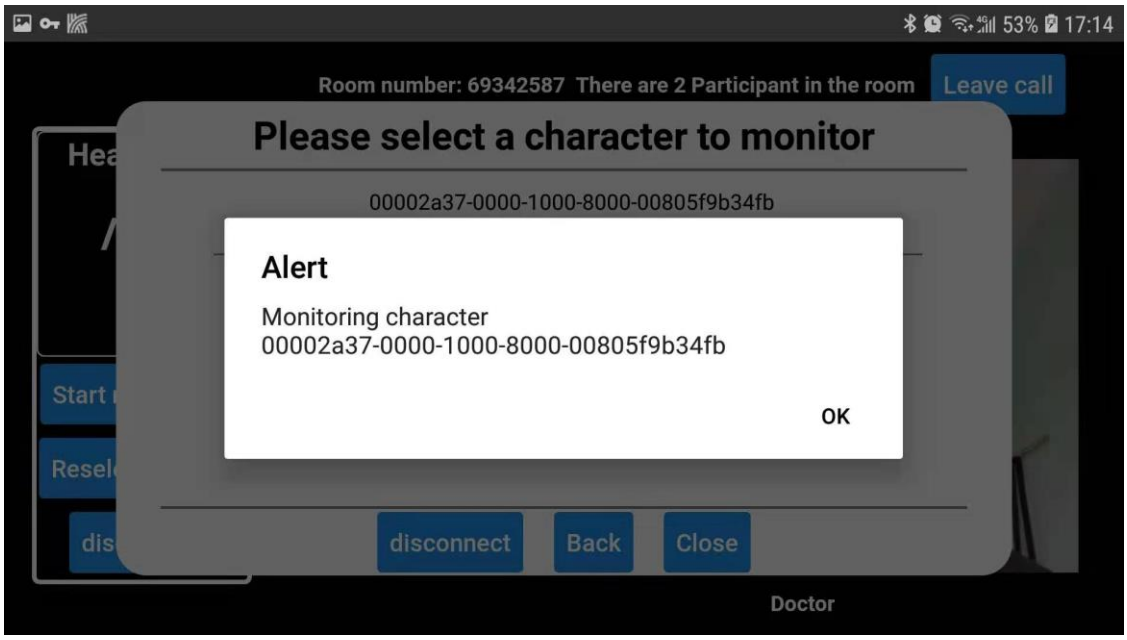

Figure 31: Selecting a BLE characteristic in the patient side application

Then patients can click the “Next” button to choose the characteristic of the service they want to monitor, all characteristics are shown and can be selected same as the service information.

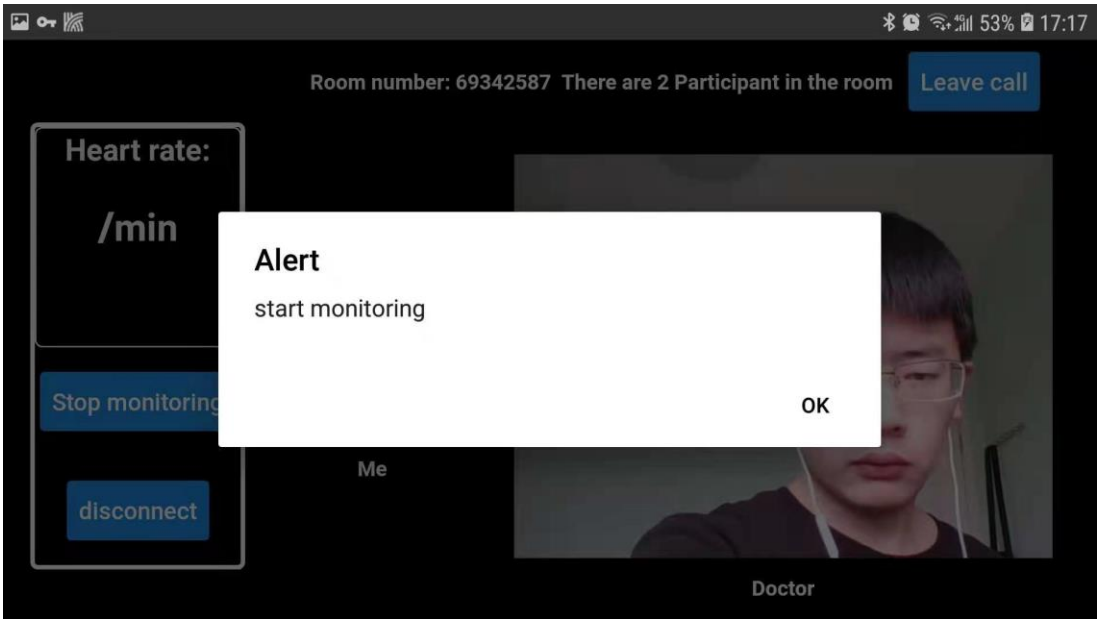

Figure 32: Starting monitoring in the patient side application

After both service and characteristic are selected, a “Start monitoring” button will appear, patients can let their application start to receive data from the connected medical device by clicking this button.

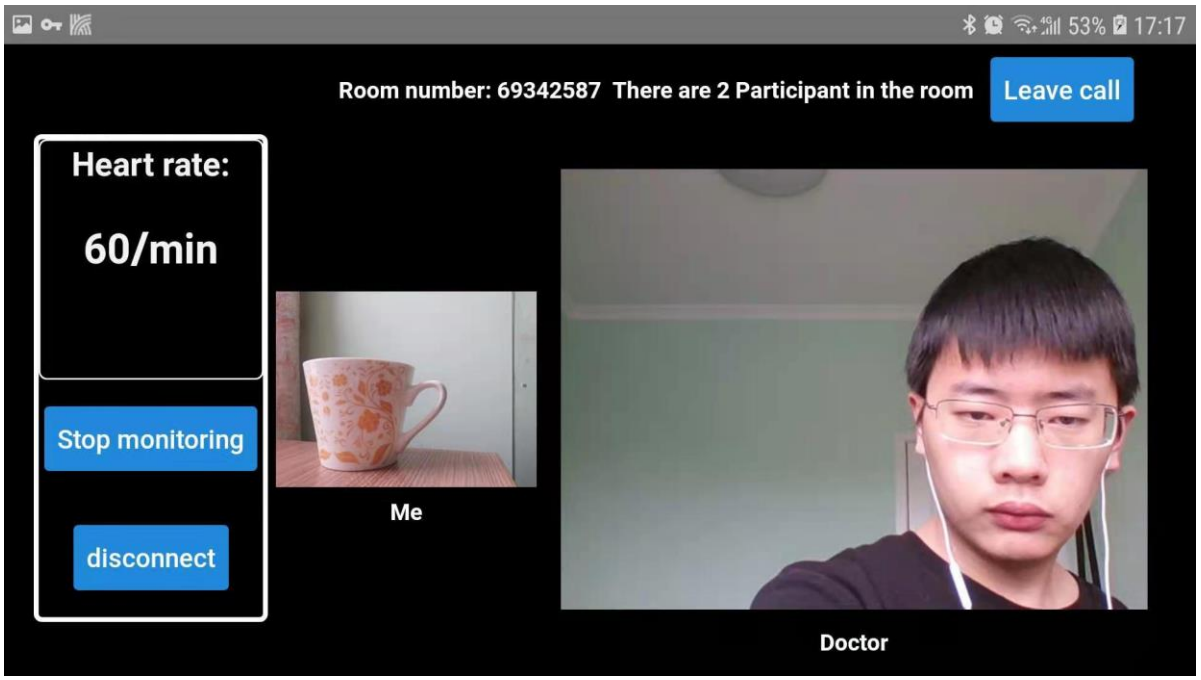

Figure 33: Receiving medical data in the patient side application

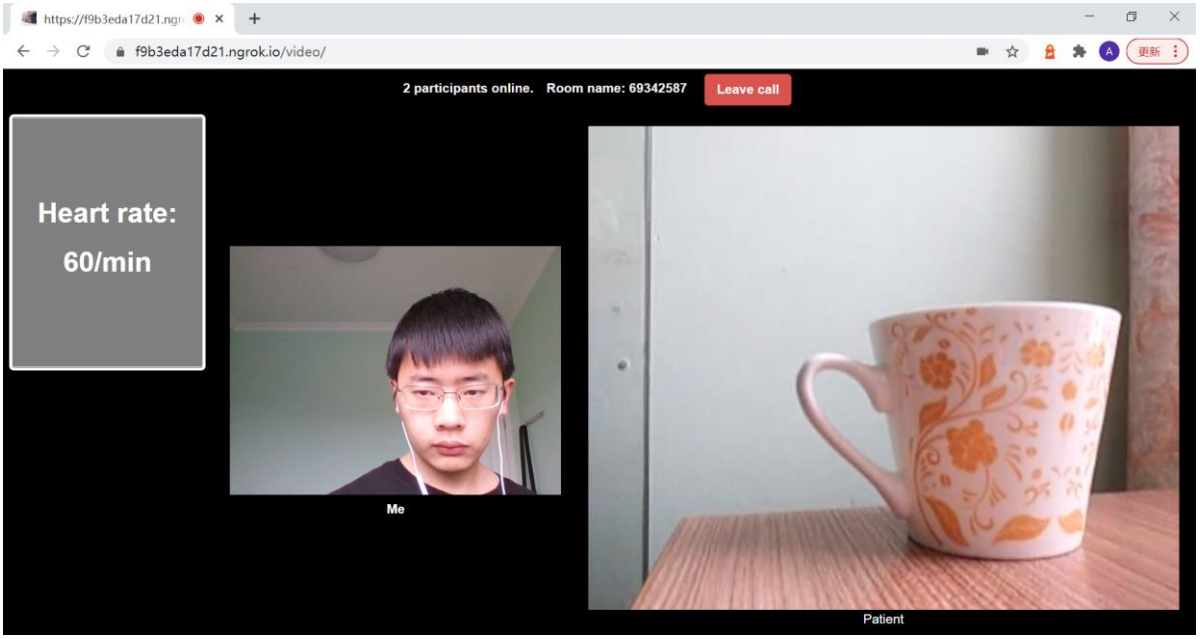

Figure 34: Receiving data shared by the patient side application in the patient side application

When receiving a data from the medical device, the patient side application will send the data to the doctor side through Twilio data stream. The data can be received by the doctor side in half a second.

3 Test results

For all tests, the patient side were all run on an Android system Samsung phone.

3 tested was conducted with doctor side used in different conditions. In all 3 tests the server was located in China, Xian, (the ‘x’ in the following figure) where the patient side was used. In test 1 and 3, doctor sides were used in Sydney, (the square in the following figure) and in test 2 the doctor side was used in China, DongGuan. (the circle in the following figure)

Figure 35: The location of 3 tests

Tests have shown that the system can operate internationally, allowing patients to receive a doctor's diagnosis even when they travel or work abroad.

Test 1:

The doctor side used a Windows system HP Pavllion laptop with a built-in camera and microphone, and the web application was run on the Chrome browser.

Doctor side:

Figure 36: The interface of the doctor side application in test 1

Patient side:

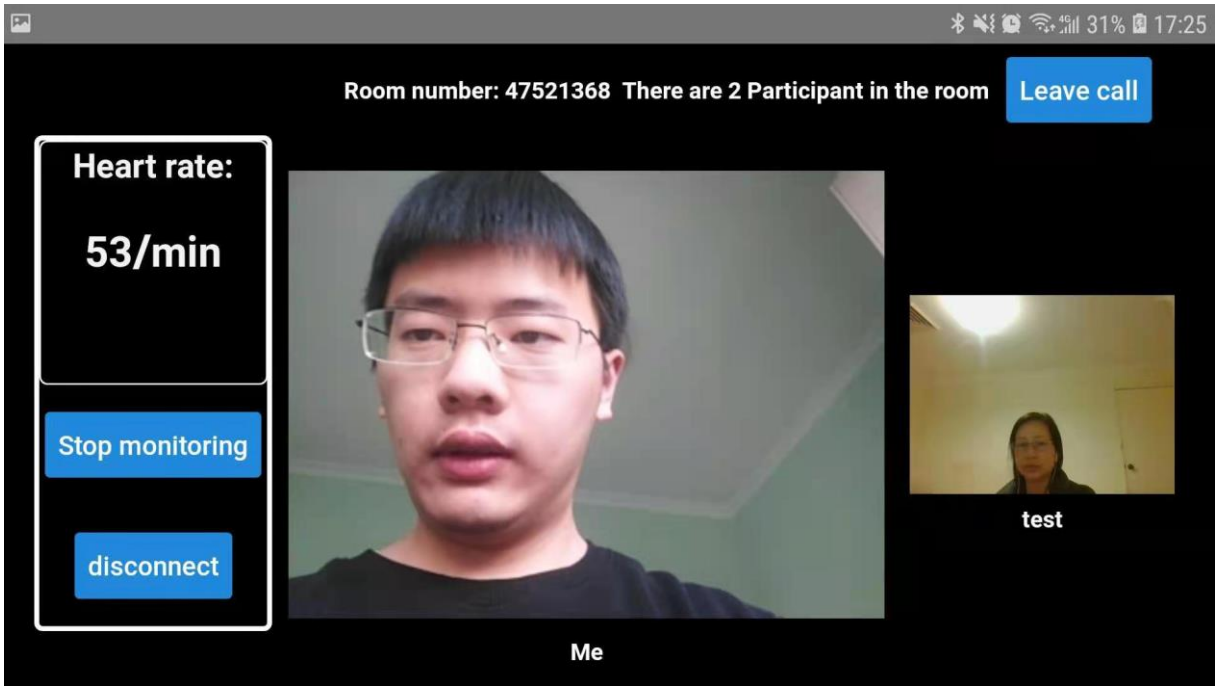

Figure 37: The interface of the patient side application in test 1

3-participants mode was also tested in this test, to simulate the situation of 1 patient use mobile

application and 2 doctors used web applications to join the same room. The following figure is the performance of the patient side in the 3-participants mode, “Me” is the image captured by the phone’s camera, “test” is the image captured by the Pavllion laptop, and “Andrew2” is the second doctor in the conferencing room, it used a Windows ThinkPad laptop with Chrome browser.

Patient side:

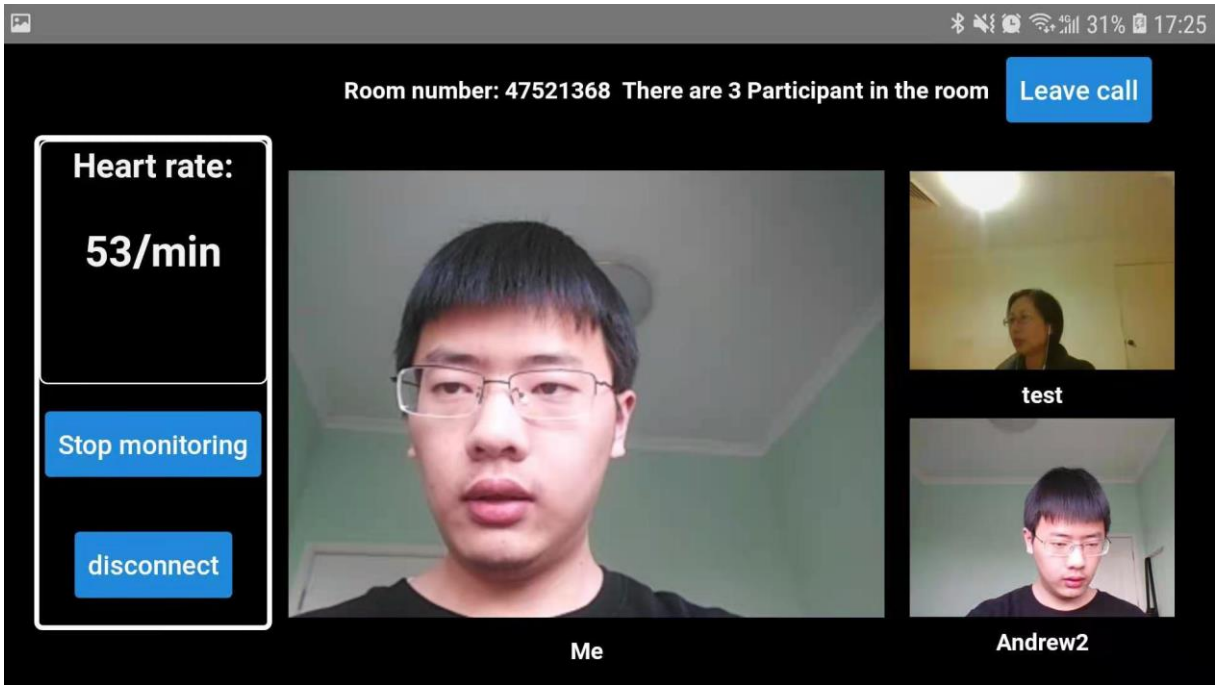

Figure 38: 3 participants mode interface of the patient side application in test 1

Doctor side:

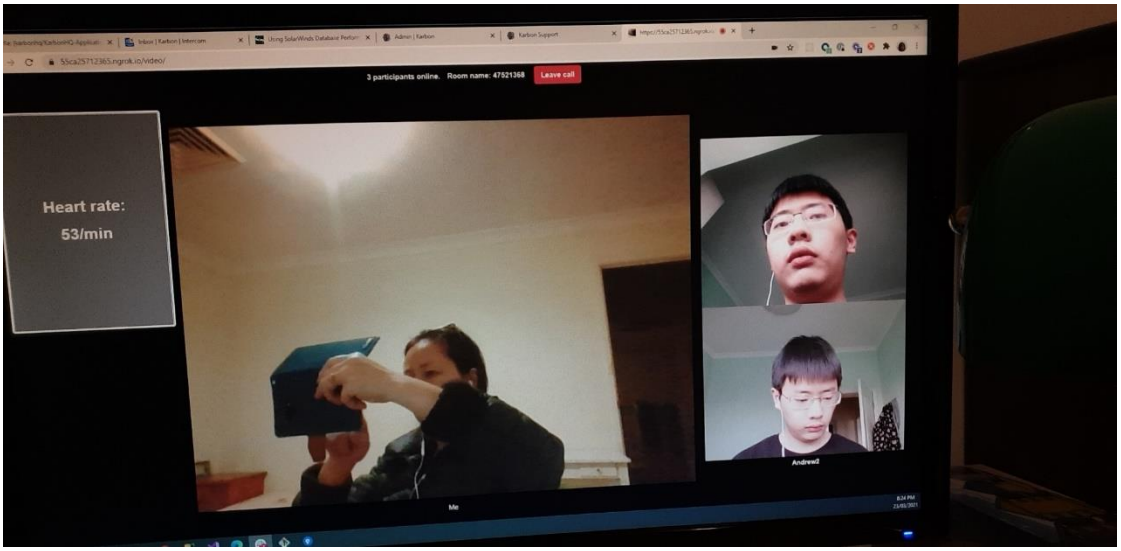

Figure 39: 3 participants mode interface of the doctor side application in test 1

In this test, the picture and sound quality remained clear, with no noise, delay, or blur. The data on the patient side can be received by the doctor side within half a second. When three participants joined the meeting, the picture and sound quality were not affected, and the data transmission time was still less than half a second. Overall, this was a successful test, with both the patient side and the doctor side applications achieving the desired performance.

Test 2

In this test doctor side used a Canon EOS Series external camera, a Windows laptop and the Microsoft Edge browser.

Patient side:

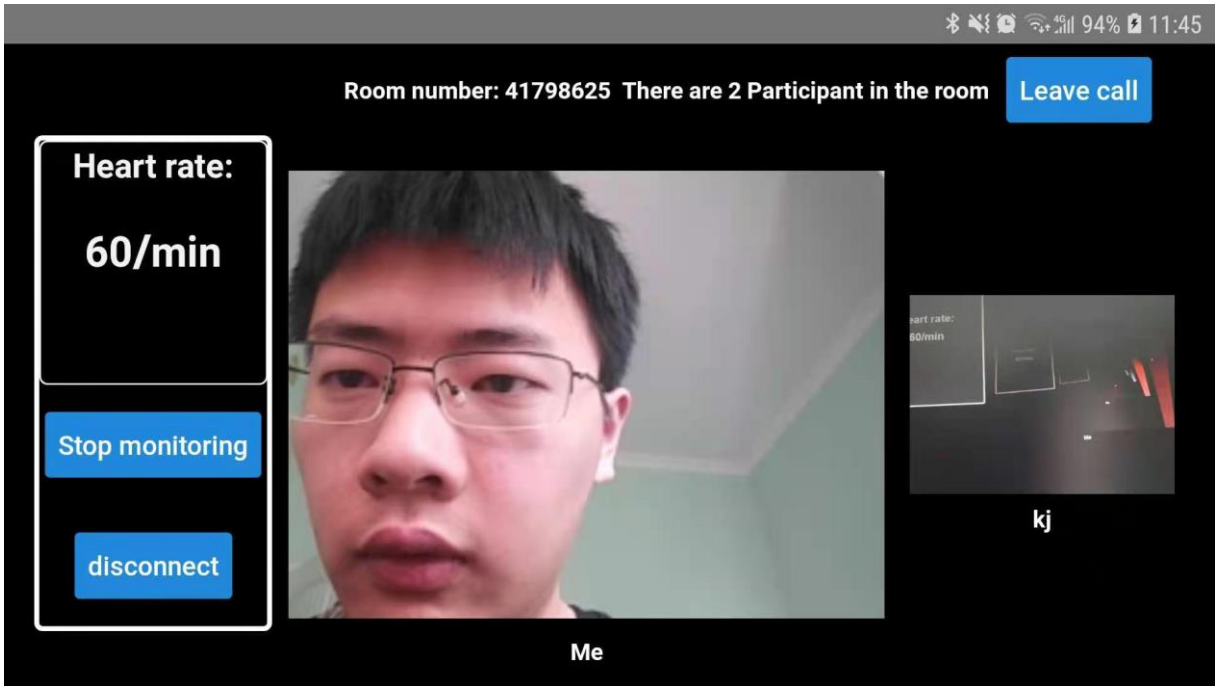

Figure 40: The interface of the patient side application in test 2

Unfortunately, in this test, the person who used the doctor side refused to show his face or room, therefore he pointed the external camera to the laptop screen, but the medical data on the patient side can still be seen matched the data on the doctor side.

The video and audio quality were excellent in this test, without any delays or blurring, and the picture was clearer than the other two tests, thanks to the use of the external camera with better resolution than the laptop embedded camera. The average data transmission time remained in half a second. In general, this test reached the desired result.

It also proved that although the system was designed for software-based video conferencing, it was still compatible for hardware-based video conferencing. This ability allows the system to run within existing hardware-based videoconferencing systems used by many hospitals, reducing the cost of using the system for doctors.

Test 3:

In this test, a registered nurse was asked to use the doctor-side application on an Apple computer, using the Chrome browser with a built-in camera and microphone. The patient-side application used a VPN, which caused the network condition to deteriorate to test the data transmission speed between the doctor and the patient under the low network speed. When using VPN, average download rate and upload rate were 26.83 Mbps and 14.99 Mbps. Without VPN, average rates were 36.98 Mbps and 40.42 Mbps.

Patient side:

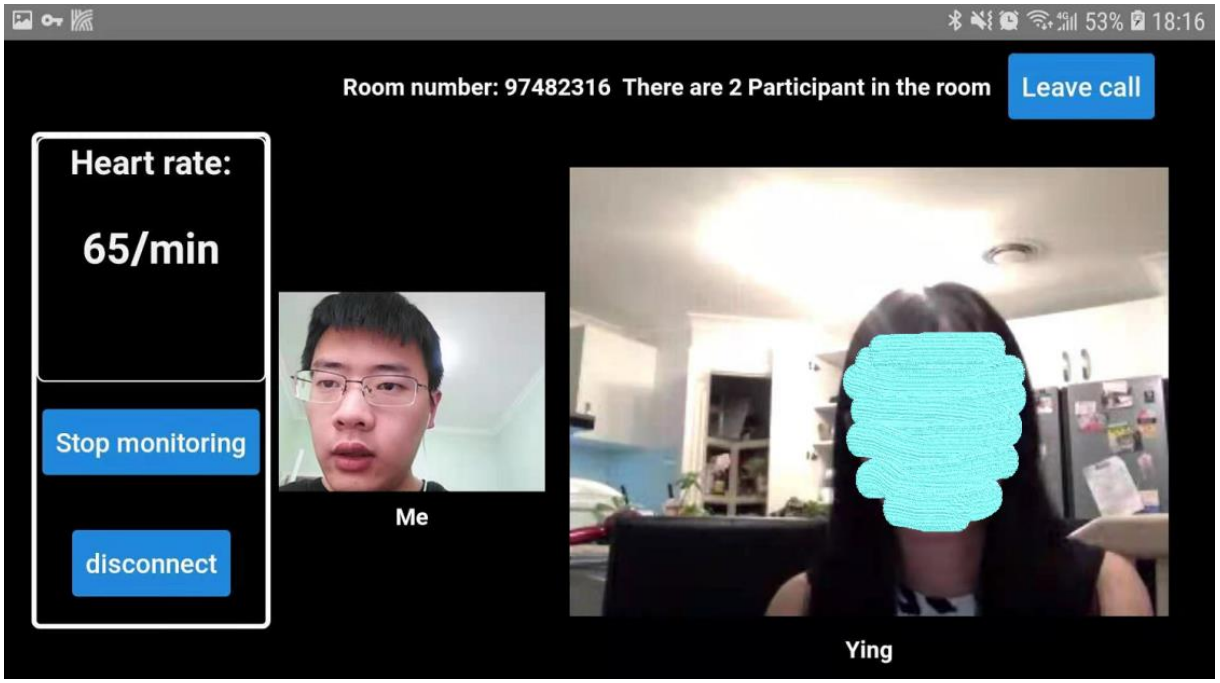

Figure 41: The interface of the patient side application in test 3

Doctor side: (before the patient joined)

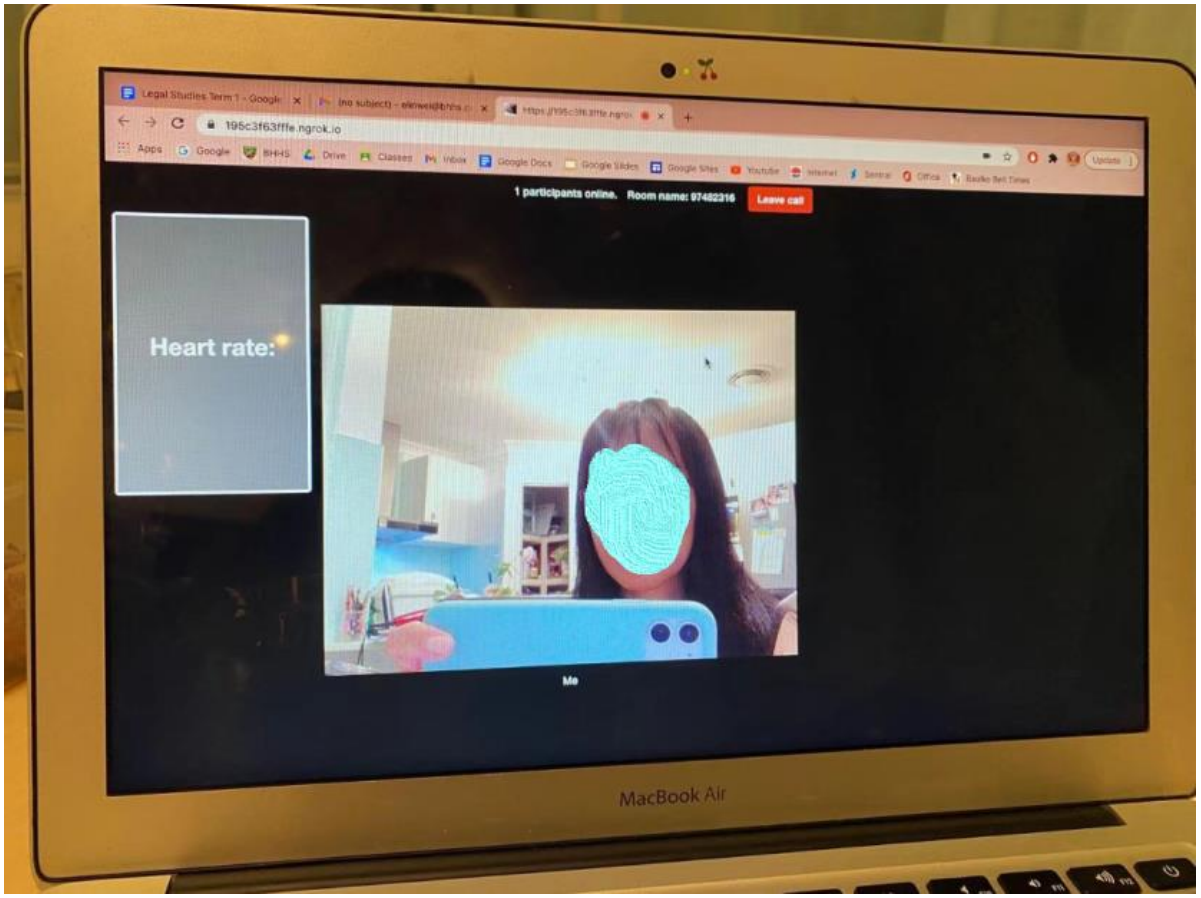

Figure 42: The interface of the doctor side application in test 3

In this experiment, When the doctor's application is running on a Mac, all parts still work properly, demonstrating the system's ability to run across operating systems. Due to the decrease of the network speed, the picture quality began to become unclear and miscellany points began to appear, and the sound was sometimes delayed or stalled. Although affected, the conversation between the doctor side and the patient side applications could take place. The most important data transfer speed can still be maintained in this case, the doctor can still receive the data sent by the patient within half a second to a second. This test proves the ability of the system to operate normally under harsh network conditions.