SOFTWARE REQUIREMENT

PROJECT NAME: CHAT APP

Bellow are demanding requirement of the chat app:

1. Basic Requirements  
    - Create a chat room feature.  
    - Display a list of chat rooms.  
    - Implement the ability to join chat rooms.  
    - Enable chat functionality within the rooms.  
     
   2. Advanced Requirements (Bonus Points)  
    - Implement file transfer functionality within the chat rooms.  
    - Enable video conferencing using WebRTC.

The requirements are explained in detail list from height to low priority as following:

|  |  |  |
| --- | --- | --- |
| # | Functions | Tasks |
| 1 | Create sign up function (1) | (credentials: username, name, password) |
| 2 | Create sign in function (2) | With credentials of sign up function |
| 3 | Create room function (3) | 1. Each room has an admin (room creator) 2. Each room has many members |
| 4 | Add a user to a room (4) | 1. A user can be invited by any room’s member  2. The invited user can accept or decline the invitation |
| 5 | Display list of rooms (5) | 1. Only admin and member can see the room  2. Display list of invitation list |
| 6 | Chat room function (6) | 1. In a room, admin and member can send messages in real time  2. The list of messages will be saved in database and display later when user open the room  3. Each one can interact with a message (give like, heart, emotion icon)  4. Each one can reply a message  5. When a member is typing message, other members can see it in real time  6. Each one can recall the message (revoke) in real time  7. Each one can share a file or multiple files in the room 8. Each one can download the file in the chat room |
| 7 | Display list or user (7) | User can login the app and view list of all users signed up |
| 8 | Video conference function (8) | - A user can call another user with or within the room  - A user can make a video conference in a room and invite members (member not invited cannot join the call)  - A member of room will take a notification in realtime when there is a conference happening |
| 9 | Setting function (9) | User can change there name and password |
| 10 | User interface(10) | - Web app using react  - Lint the code to ease development process |
| 11 | Real time technology (11) | - Using socket.io with any back end framework  - Using webRTC to handle video call function |

List the requirements in high to low priority order:

|  |  |
| --- | --- |
| Priority Order | Functions/Tasks |
| High priority | Function 1  Function 2  Function 3  Function 4  Function 5  Task 6.1, 6.2, 6.3, 6.4, 6.6  Function 10  Function 11 |
| Low priority | Task 6.5, 6.7, 6.8  Function 7  Function 8  Function 9 |

Technologies (programing languages and framework):

Backend: Nodejs - Express.js

Frontend: ReactJs - Vite

Database: MongoDB

Realtime: Socket.io

Video Call: WebRTC - simple-peer

Research:

1. Backend:

There are many backend technologies can be used to build this chat app. However Nodejs is one of the best platform technology for this case. Reason are it is created for real time application and lots of big company using it are Netflix, NASA, Trello, Uber,…

1. Database:

Mongodb: noSQL document-oriented database. Using Json like document to store records of entity. It is very flexible and scalable. Because its file read and write mechanism, it is very suit for app that requires high write tendency data even when database is hundred of million documents compare to SQL databases.

MySQL: a very popular relational database management system that used by large corporations such as Facebook, Twitter, Airbnb, Booking.com,… It widely used in most Content Management Systems like WordPress, Drupal, Joomla!, Contao,… arcoding to W3schools (1)

Solution choice: For this application, I will choose the Mongodb as database due to its flexible schema. It help to build the app faster and easy to modify the data when there is a change in app design.

1. Video Call: For the ease of development, I choose not to code video conference function with vanillajs but with a library named “simple-peer”. This is a library that works as webrtc wrapper for create a peer connection between two clients.

Citations:

1. https://www.w3schools.com/mysql/mysql\_intro.asp