Assignment Phase 1

Individual Assignment
Due: 8 am Thursday 8 Sep 2022

Introduction

You are required to build the server and front end for a chat system.

The chat system will allow users to communicate with each other in real-time within different groups and channels. Some users will have admin permission to add users to channels and groups, whilst a super admin has access to the entire site. The solution must be implemented using Node.js, Angular, and sockets.

Terminology

Group

A collection of users that have been given permission from a group admin or super admin to be able to be a member of the group.
Multiple groups can exist.
Each user can exist in multiple groups.

Channel

Each group will have access to several rooms/channels for the purpose of chatting.
The super/group/group assis admin can create the rooms and give/remove permission for existing group member to access them.

Architecture

A global list of groups is required.
A global list of rooms is required for each group.
A list of which users are in each group and rooms.

Users

- UserName
- Email
- ID
- Role

The system will have users which can have the following additional roles:

- Super Admin
- Group Admin
- Group Assis (Assistant)

Super Admin

- A Super Admin can create users with Group Admin role.
- A Super Admin can also remove all users.

- A Super Admin can also upgrade another user to Super Admin role.
- A Super Admin can also do whatever a group admin can.

Group Admin

- A Group Admin can create groups.
- A Group Admin also can create channels (subgroups) within groups.
- A Group Admin can create and or invite users to a channel (if the user has already been created, they will simply be added to the channel).
- A Group Admin can remove groups, channels, and users from channels.
- A Group Admin can also allow a user to become a Group Assis of the group.
- A Group Admin other than a Super Admin **cannot** create a user with Group Admin role.

Group Assis

- A Group Assis of a group can add or remove users in the group from channels within the group.
- A Group Assis of a group can also create channels within the group.
- Group Assis **cannot** do other operations of the Group Admin role.

A user is identified by their username. Initially there is one user called 'super' who is also a Super Admin.
A user also has an email address (no emails are sent to the email address).

The first page of the website requires a user to enter their username, which is remembered in local storage. A user may 'logout' which also clears the username out of local storage.

Once a standard user enters their username the page should display the groups they have been added to and the channels for each group. The user can only access the channels they have been given access to. Once a group and channel combo have been selected the user should be able to begin chatting with other users in that channel.

For Super Admin, Group Admin and Group Assis, the page should also display the input forms according to their roles abilities for adding or deleting users, groups and channels, respectively.
Selecting a channel should display the channel history

A text box should allow for new messages to be sent to the channel. New messages are broadcast to all users currently viewing the channel and added to the history

User Authentication

You should add support for users entering a password. If the password does not match it should ask the user to login again.

The MongoDB, Sockets, Image and Video support are not implemented in assignment phase 1. You will use a serialized JSON file in local storage for storing all the user, group, and channel data as well as chat history.

MongoDB

The Node.js server storage use a MongoDB database. The mongo database should store all of the user, group, and channel data as well as chat history.

Sockets

Sockets will be used to support chat communication in a channel. When the user logs in they can choose from the list of groups they are in. They can then choose a particular channel to communicate in. When they are in a channel the chat history should update as users send messages. The channel should also show in realtime when users join and leave a channel.

Image support

The chat system should allow users to specify a profile image (i.e. avatar). The profile image should be displayed in the chat history alongside their username for messages that they posted. The chat system should also support sending images as a chat message and will appear to all users viewing the chat. Image storage on the server can be as files in a specified directory, with the path to the file stored in the mongo database.

Video support

The chat system should allow users to have video chat. For this purpose, more APIs can be used in both the Angular and server side. For example, PeerJS can be used in Angular side and Peer server can be implemented on your sever side. For demonstrating the function of video chat, you may upload your app on elf.

Git

Git must be used during the development of the chat system. We recommend that you use GitHub and share the repository with your marker. You will be marked on frequent updates to the repository and the usage of git features.

Documentation

Documentation of your implementation is required. You will need to provide the following:

- Describe the organization of your Git repository and how you used it during the development of your solution
- Description of data structures used in both the client and server sides to represent the various entities, e.g.: users, groups, channels, etc.
- Angular architecture: components, services, models, routes.
- Node server architecture: modules, functions, files, global variables.
- A description of how you divided the responsibilities between client and server (you are encouraged to have the server provide a REST API which returns JSON in addition to a static directory)
- A list of routes, parameters, return values, and purpose in the sever side
- Describe the details of the interaction between client and server by indicating how the files and global vars in server side will be changed and how the display of each angular component page will be updated.