Software Frameworks Documentation – s5289789 Thomas Marion

**Git Hub repository layout:**  
My Git repository is separated into several branches where I can make relevant changes that won’t affect my source version on main. Once the changes on the branches were tested and fully functional, I would then merge them into main. This always ensured when making new features I would always have a clean source version to go back to if things went very wrong. In terms of version control I would push new versions of my code to these side branches every 10 minutes or so depending on how much I had added/changed.

Data Structures:

Server Data:  
 server data just contains fields for users and groups.

User:

Contains all user information like username, email, password etc as well as lists for groups and roles they are apart of

Group:

Group contains fields for its name, as well as lists for channels it might parent as well as a user list, and the ID of the user who created it

Channel:

Channel contains fields for the channel name, creator of it, as well as a list of all current users in it.

**REST API documentation:**

ROUTES:

/api/auth – POST:

This route receives a user input username and password and checks if it exists on the server side instance of users. If it does not exist it returns a valid Boolean of false. If it returns valid, it generates a JWT token containing the USER ID of the now signed in user as well as a valid Boolean of true.

/api/auth – POST

This route receives a username, email and password from a super administrator and checks if the username already exists from the existing set of users. If its unique the new user will be created and added to the server-side list meaning someone can now log in with the given username and password.

/api/updateProfile – POST

Takes in values in all fields of an existing profile besides password as well as a token stored locally on the users browser. If the token is valid and the user ID from the token matches an existing user it will change their server side details with the provided ones.

/api/verifyToken – POST

This takes in a locally stored JWT token and returns whether its valid or not

**Angular Architecture:**

COMPONENTS:

Chat-Room

Component that displays the chatroom as well as additional features dependant on the user role

Create-Account

Sub component displayed to super admins on the chatroom page to create new users

Group-UI

Sub component to display a list of all current groups to users in the chatroom page

Home

Basic home page for the site

Login-page

Page to enter username/password and auth to a valid user server side

Nav-bar

Standard bootstrap nav bar with working links to redirect to relevant pages

Profile

Page to update current user info fields

SERVICES:

API – service:

Service responsible for handling all route calls to the server. Middleman handler between client and the server

Socket-service:

Service responsible for handling all socket related requests and responses to and from the server to update UI components user side.