# 3813ICT Software Frameworks

Assignment Phase 1

Zhaocheng Dong(Chris)s5172215

## Table of Contents

**Description**

**Plan**

**Data structures**

**Rest API**

**Angular Architecture**

### ****Description****

The main purpose of this software is to develop a new web chat software. Its development cycle is divided into two main phases. The first phase is aimed at building the framework of the software and defining the main state of the software, while the second phase is aimed at improving the server functionality of the web chat software as well as the database and other more detailed features. The development of this software is based on the use of the latest Vscode for development and testing. Every effort has been made to ensure that the software will be fully functional and tested by the end of the second phase.

### Plan

Phase 1.  
1. Have a preliminary software plan and a name for the software (initially decided on 'webchat').  
2. Build the main framework of the software.  
3. Define the functionality (applicable to the first phase).  
4. Test the software regularly.  
5. Complete a summary and upload it to Github.  
Phase 2.  
1. refine any unfinished development work from phase 1.  
2. Secure a server for the software.  
3. Build the database for the software.  
4. Develop more personalised features.  
5. Complete the summary and upload it to Github.

### Data structures

There are four main data structures in this program. They are users, groups, channels and permissions. A user is an array, consisting of a user object. A user has an ID, a username, an email address and a role. A group is an array consisting of a group object. A group has an ID and a group name. Channel is an array consisting of a channel object. A channel has an ID and a group name.

{groupId: number, members: [

{userId: number, channelIds: [number,…]

]}

### **Rest API**

**Identify the user**

export const rolePermission = () => {

    filterRouter(router, roleRouter);   router.addRoutes(router); }

**Distinguishing user permissions**

export const permission = {

    member:["Home"],

    admin:["Home" ,"Notify"],

    super\_admin:["Home" ,"Notify","Manage"]

}

**Instances (users with different permissions)**

[

    { "id":1 ,"user":"Admin" , "email":"Admin@gmail.com" , "password":"Admin", "Permission":"Admin"},

    { "id":2 ,"user":"Chris" , "email":"Chris@gmail.com" , "password":"Chris", "Permission":"super\_admin"},

    { "id":3 ,"user":"Mike" , "email":"Mike@gmail.com" , "password":"Mike", "Permission":"member"}

]

**User permissions(example)**

    {

        "home": {

          "id":"100",

          "name":"home",

          "desc":"home",

          "value":true,

          "children": []

        }

      }

**Adding and removing user permissions(example)**

          "options": {

            "create": true,

              "delete": true,

              "update": true

        }

### Angular Architecture

Login  
This is the default path for the program. It consists of a simple login form with a username input and a submit button. When the submit button is clicked it calls the login function, which makes an HTTP POST request to API/auth with the username. If the response returns ok: true, then the returned user data will be set to the user in local Storage, the permission service's update User function will be called with the return value of what is stored for the user in local Storage, and the application will navigate to the chat route. If the response returns ok: false then the local error variable is set to true and a prompt is displayed on the front end stating that the username or password is incorrect.  
Chat  
This is the heart of the application. It consists of a side navigation panel and a content area. The side navigation is populated on load by an HTTP POST request /get User Channels which returns a hierarchical array of groups and channels to the group local variable.