

# Fundamentals of Programming Project Documentation

## Client Side:

In this part of the project we have some functions I'd like to explain the reason and use of them;

### `void myconnect():`

This function is the main part of the client connection to the server; It consists of creating a socket and connecting to the sever on the defined port.

### `void create():`

This function is the tool that I used to build the string that we want to send to the server; For example if we want to register a new user to the server by the name of "Mahdi" and password "1234" and we call the create function the produced string would be "register Mahdi, 1234".

### `void chat():`

We use this to receive the resulting string from the server and process this string and answer accordingly.

### `void print():`

This function is used to produce the interactive menu that helps the user to toggle between the parts we have built.

## Server side:

This part of the project has been developed a little less modular and with less functions but still the code is quite comprehensible and easy to understand; So in the following, we will discuss the parts that we coded to this phase.

### `bool checkAuth(char* s):`

In this phase of the project we have to authenticate the user many times, and in order to do that we need to validate the authentication token that we receive; This function looks into the date base and checks if such user exists or not.

In the function `main()` we start the code by creating a socket connection to the defined port and listening to that port;

Then we receive the command string from the client side we are listening to and process the string by the commands you had given us in the project doc sent into the telegram channel.

The rest of the code is quite readable; Actually like English. For each command we do the following:

- 1 - We receive the string,
- 2 - We separate the parts that is in the string,
- 3 - We do the necessary validations(authentication token, channel name or the user name given to us),
- 4 - We edit our database according to received datas,
- 5 - We generate the JSON we need to send to client,
- 6 - We send the JSON to client,
- 7 - Wait for the next commands and repeat :)

```

else if(strcmp(firstCommand , "create") == 0){
    char chname[MAX] = {0}, authTok[MAX] = {0};
    int ptr , cnt = 0;
    for(int i = 15; ; i++) {
        if (buffer[i] == ',') {
            ptr = i + 2;
            break;
        }
        chname[cnt++] = buffer[i];
    }
    strcpy(authTok , buffer + ptr);
    FILE* fp;
    cJSON* ans = cJSON_CreateObject();
    char path[] = "../Resources/Channels/User/";
    strcat(path , chname);
    strcat(path , ".txt");
    if(!checkAuth(authTok)){
        cJSON* type = cJSON_CreateString("error");
        cJSON* content = cJSON_CreateString("such user does not exist");
        cJSON_AddItemToObject(ans , "type" , type);
        cJSON_AddItemToObject(ans , "content" , content);
        goto this;
    }
    if(fp = fopen(path, "r+")){
        fclose(fp);
        cJSON* type = cJSON_CreateString("error");
        cJSON* content = cJSON_CreateString("channel already exists");
        cJSON_AddItemToObject(ans , "type" , type);
        cJSON_AddItemToObject(ans , "content" , content);
        goto this;
    }
    fp = fopen(path , "w+");
    cJSON* type = cJSON_CreateString("Successful");
    cJSON* content = cJSON_CreateString("channel created");
    cJSON_AddItemToObject(ans , "type" , type);
    cJSON_AddItemToObject(ans , "content" , content);
    fprintf(fp , "%s" , authTok);
    this:
    memset(buffer , 0 , sizeof(buffer));
    char* s = cJSON_Print(ans);
    strcpy(buffer , s);
}

```

The code above shows what we do for the create channel command and this is exactly what we have explained above;

We check the string, separate the parts we want, check the token, check if the channel does not exist beforehand, create the channel, and make the JSON file and print it to buffer to send back to client.

And this is it!

This documentation covers everything implemented in this phase and those of phase 1; Still if anything is not clear or you found a bug during working with the app please feel free to contact us at [gheidi@ce.sharif.edu](mailto:gheidi@ce.sharif.edu) .

The code and documentation have been written by  
**Mohammad Mahdi Gheidi**.

Wish you all the best and good luck!