Simple Login and Registration System

Introduction

In every system, one of their most important parts involves the login and registration system. From reasons ranging from privacy to keeping track of a user's track record or details, the ability for a user to be able to create their own account can be extremely beneficial to both the user and the system itself. A login and registration system are applied in nearly every system in todays modern and digital world. From creating student accounts for university students, to accessing bank account details, to gain access into a social media account or even a operating system, to even keeping track of a user's purchases in order to qualify for a discount, the login and registration system is used in nearly every system these days.

The login and registration system are basically a system whereby an individual user is able to register an account of their own to be able to gain access to the system's full use. The user will then have to login again to gain access to their accounts or the system once they have logged out.

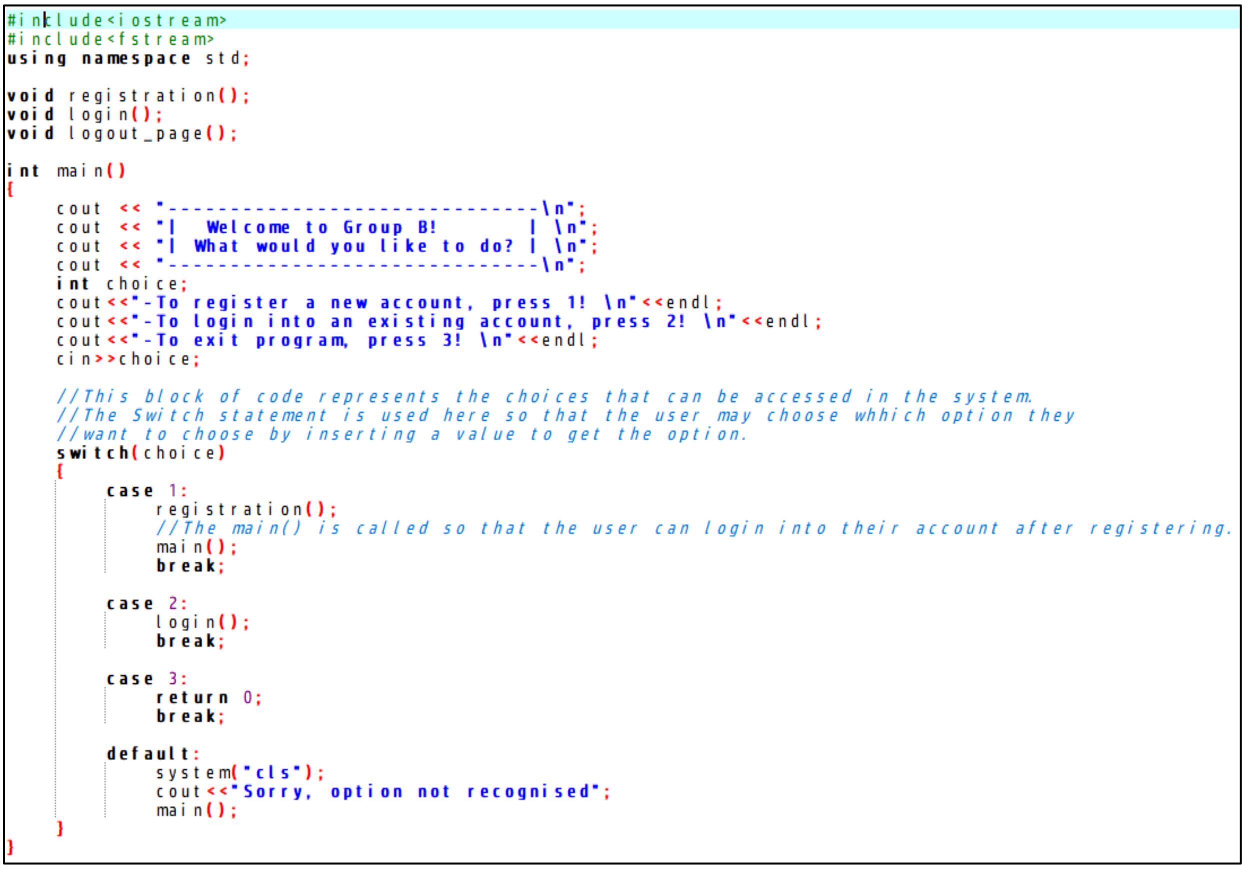
Thus, the main part of our program will mainly focus on enabling the user to be able to create their own account, using specific and unique credentials such as their "username" and "password". Once they have created their account, they will be able to access their account and then logout again, before having to login again to gain access to their account. Our project will be solely focused on the process of registering, logging in and logging out of an account, which will all be programmed by using the C++ programming language.

In order to be able to create this system in the C++ programming language, various applications from the programming language will be implemented to create this program. These implementations involve the input and output of the program, which will mainly be used to allow the user to input their usernames and passwords and for the system to be able to output various instructions for the user. Besides that, if-else and loops will also be used in the program, such as to make sure that every username is unique, to insert the usernames and passwords into their respective .txt files and to check the username and password combination in order to give access to that specific user into their accounts. Besides all of that, array's will also be used, which are again related to the usernames, passwords and their respective .txt files in which the prior two will be saved. And, obviously, the backbone of any program which is Functions will also be heavily used to help complete the login and registration system.

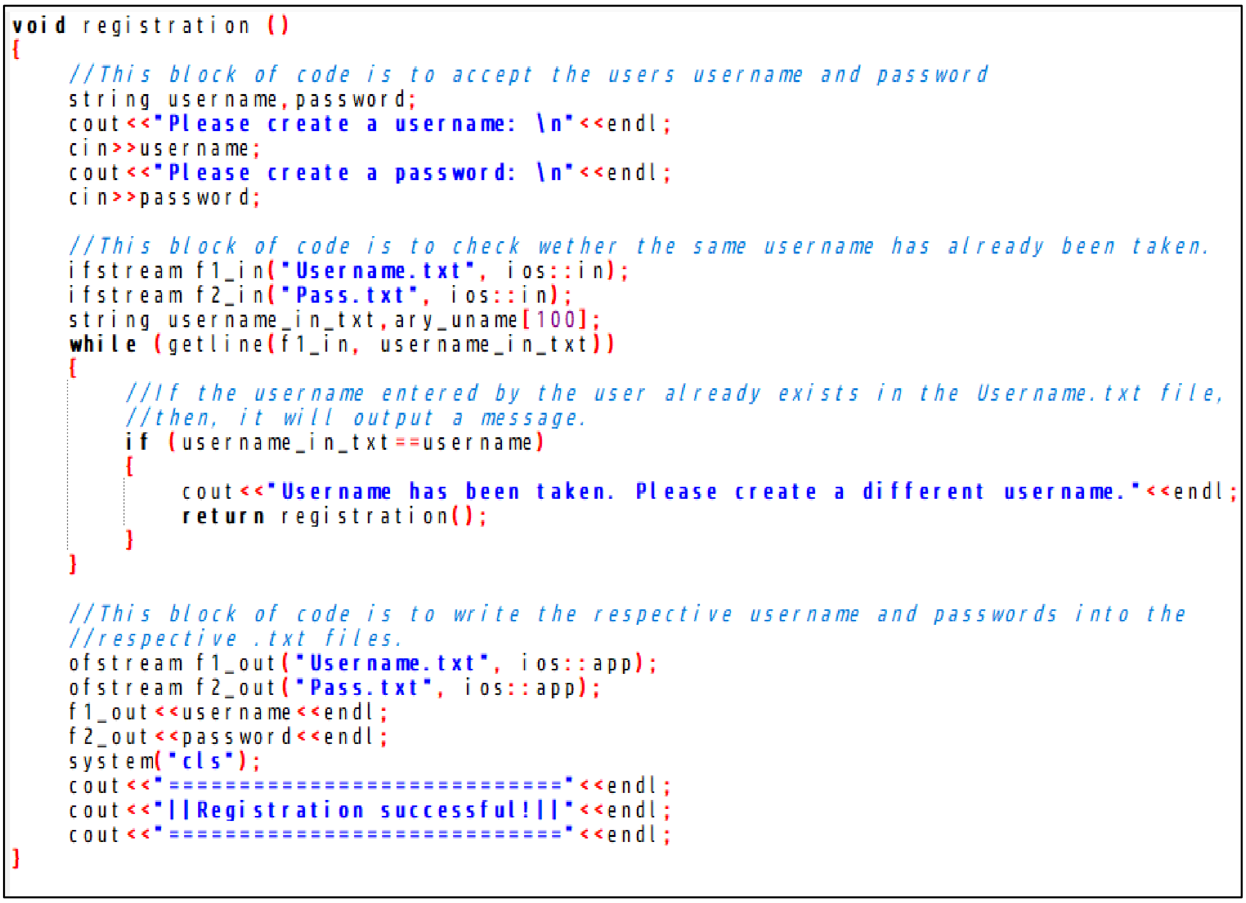
Objectives

- To be able to create a program where the user can register an account using unique username and password.
- To be able to create a program where the user can login into their accounts with only their correct credentials (username and password) to gain access to their account.
- To be able to create a program where the user can logout of their account easily.

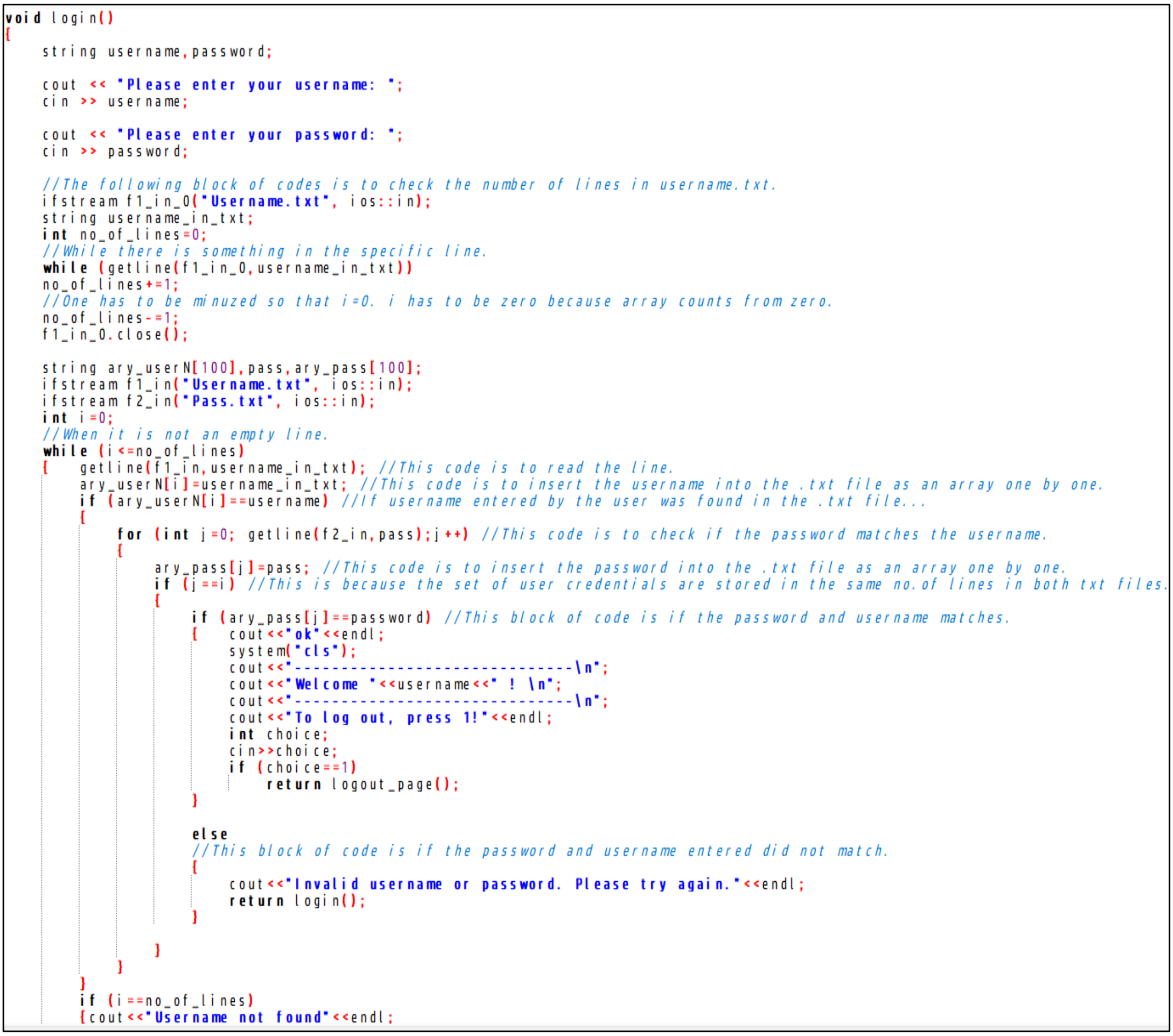
Source Code



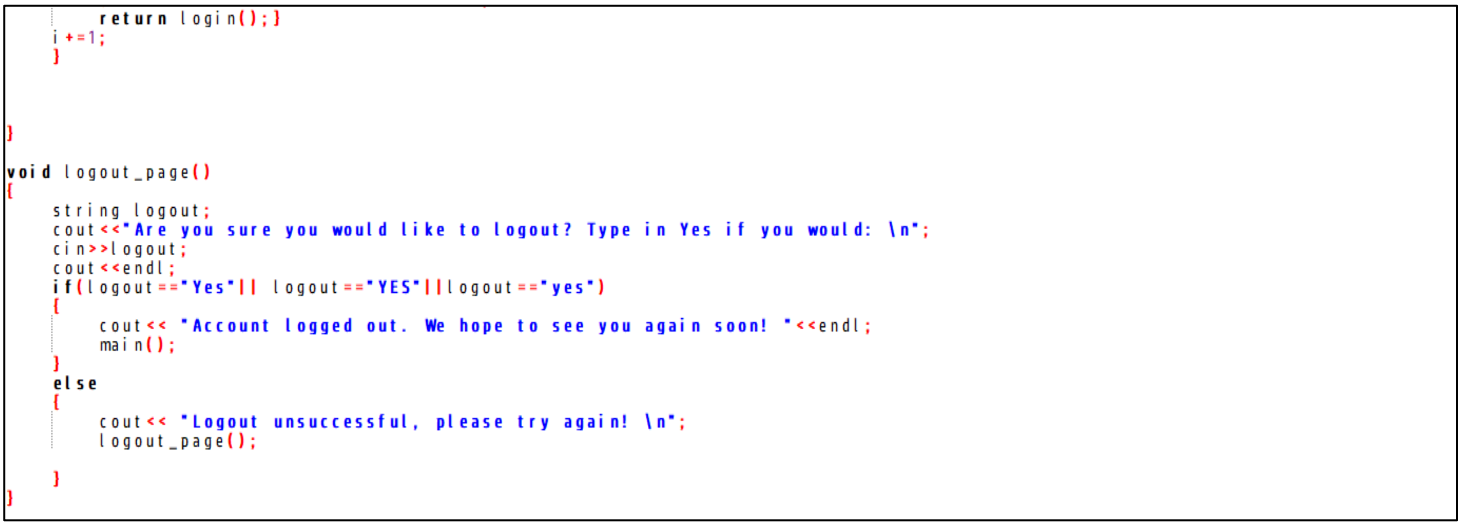
```
#include<iostream>
#include<fstream>
using namespace std;

void registration();
void login();
void logout_page();

int main()
{
	cout << "------------------------------\n";
	cout << "|  Welcome to Group B!       |  \n";
	cout << "| What would you like to do? |  \n";
	cout << "------------------------------\n";
	int choice;
	cout<<"-To register a new account, press 1! \n"<<endl;
	cout<<"-To login into an existing account, press 2! \n"<<endl;
	cout<<"-To exit program, press 3! \n"<<endl;
	cin>>choice;

	//This block of code represents the choices that can be accessed in the system.
	//The Switch statement is used here so that the user may choose which option they
	//want to choose by inserting a value to get the option.
	switch(choice)
	{
		case 1:
			registration();
			//The main() is called so that the user can login into their account after registering.
			main();
			break;

		case 2:
			login();
			break;

		case 3:
			return 0;
			break;

		default:
			system("cls");
			cout<<"Sorry, option not recognised";
			main();
	}
}
```



```
void registration ()
{
	//This block of code is to accept the users username and password
	string username,password;
	cout<<"Please create a username: \n"<<endl;
	cin>>username;
	cout<<"Please create a password: \n"<<endl;
	cin>>password;

	//This block of code is to check wether the same username has already been taken.
	ifstream f1_in("Username.txt", ios::in);
	ifstream f2_in("Pass.txt", ios::in);
	string username_in_txt,ary_uname[100];
	while (getline(f1_in, username_in_txt))
	{
		//If the username entered by the user already exists in the Username.txt file,
		//then, it will output a message.
		if (username_in_txt==username)
		{
			cout<<"Username has been taken. Please create a different username."<<endl;
			return registration();
		}
	}

	//This block of code is to write the respective username and passwords into the
	//respective .txt files.
	ofstream f1_out("Username.txt", ios::app);
	ofstream f2_out("Pass.txt", ios::app);
	f1_out<<username<<endl;
	f2_out<<password<<endl;
	system("cls");
	cout<<"=========================="<<endl;
	cout<<"||Registration successful!||"<<endl;
	cout<<"=========================="<<endl;
}
```

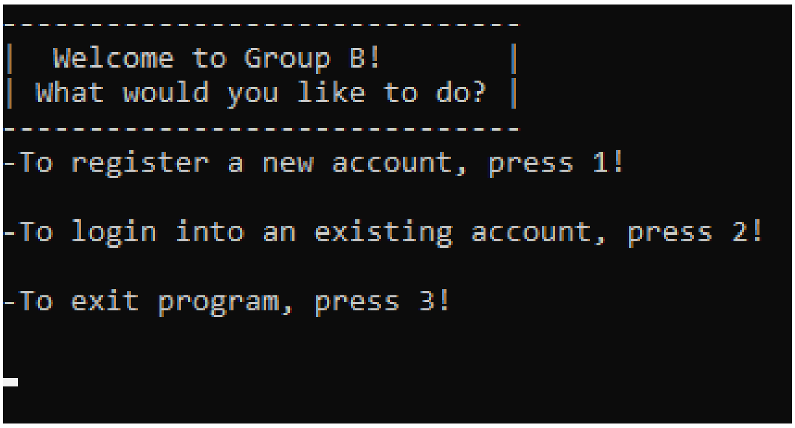


```cpp
void login()
{
    string username,password;

    cout << "Please enter your username: ";
    cin >> username;

    cout << "Please enter your password: ";
    cin >> password;

    //The following block of codes is to check the number of lines in username.txt.
    ifstream f1_in_0("Username.txt", ios::in);
    string username_in_txt;
    int no_of_lines=0;
    //While there is something in the specific line.
    while (getline(f1_in_0,username_in_txt))
    no_of_lines+=1;
    //One has to be minuzed so that i=0. i has to be zero because array counts from zero.
    no_of_lines-=1;
    f1_in_0.close();

    string ary_userN[100],pass,ary_pass[100];
    ifstream f1_in("Username.txt", ios::in);
    ifstream f2_in("Pass.txt", ios::in);
    int i=0;
    //When it is not an empty line.
    while (i<=no_of_lines)
    {   getline(f1_in,username_in_txt); //This code is to read the line.
        ary_userN[i]=username_in_txt; //This code is to insert the username into the .txt file as an array one by one.
        if (ary_userN[i]==username) //If username entered by the user was found in the .txt file...
        {
            for (int j=0; getline(f2_in,pass);j++) //This code is to check if the password matches the username.
            {
                ary_pass[j]=pass; //This code is to insert the password into the .txt file as an array one by one.
                if (j==i) //This is because the set of user credentials are stored in the same no.of lines in both txt files.
                {
                    if (ary_pass[j]==password) //This block of code is if the password and username matches.
                    {   cout<<"ok"<<endl;
                        system("cls");
                        cout<<"--------------------------------\n";
                        cout<<"Welcome "<<username<<" ! \n";
                        cout<<"--------------------------------\n";
                        cout<<"To log out, press 1!"<<endl;
                        int choice;
                        cin>>choice;
                        if (choice==1)
                            return logout_page();
                    }

                    else
                    //This block of code is if the password and username entered did not match.
                    {
                        cout<<"Invalid username or password. Please try again."<<endl;
                        return login();
                    }

                }
            }
        }
        if (i==no_of_lines)
        {cout<<"Username not found"<<endl;
```



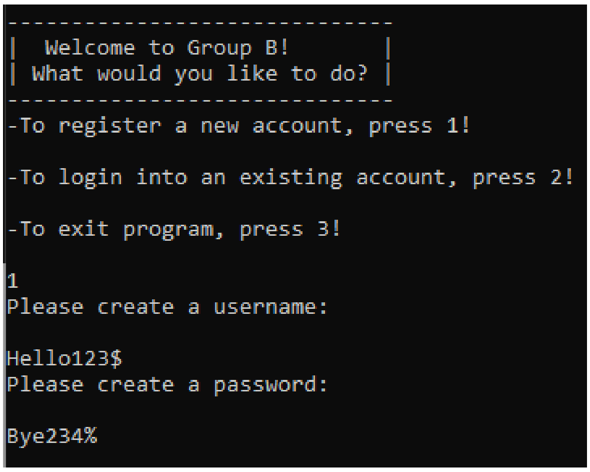
```cpp
        return login();}
    i+=1;
    }



}

void logout_page()
{
    string logout;
    cout<<"Are you sure you would like to logout? Type in Yes if you would: \n";
    cin>>logout;
    cout<<endl;
    if(logout=="Yes"|| logout=="YES"||logout=="yes")
    {
        cout<< "Account logged out. We hope to see you again soon! "<<endl;
        main();
    }
    else
    {
        cout<< "Logout unsuccessful, please try again! \n";
        logout_page();

    }
}
```

Print Screens



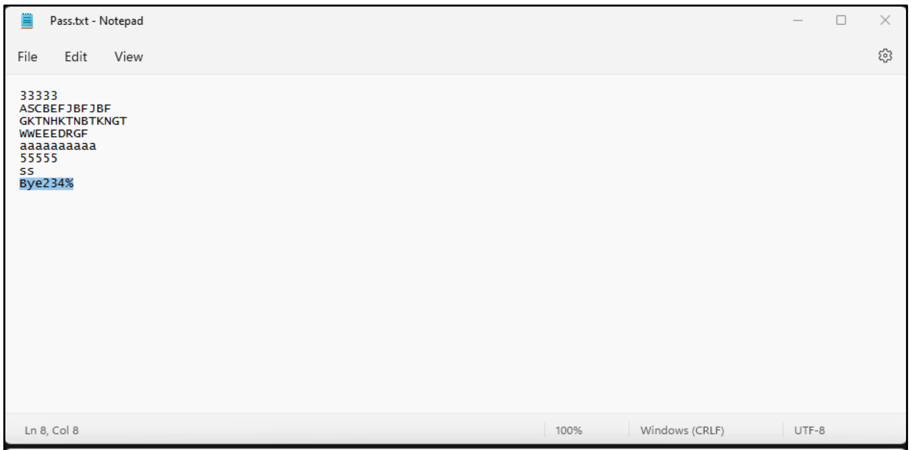
```
------------------------------
|  Welcome to Group B!        |
| What would you like to do? |
------------------------------
-To register a new account, press 1!

-To login into an existing account, press 2!

-To exit program, press 3!
```

The above screenshot shows the main menu of our login system. This main menu will also be the first program outputs that the user will see once they have entered the login system, where the user will be welcomed by a welcome message in the top of the screen followed by a question as to what they would like to do. The user will then have 3 options to choose from as to what they would like to do. Each of those options have been provided a numeric value where the user will have to type in to choose their respective option.

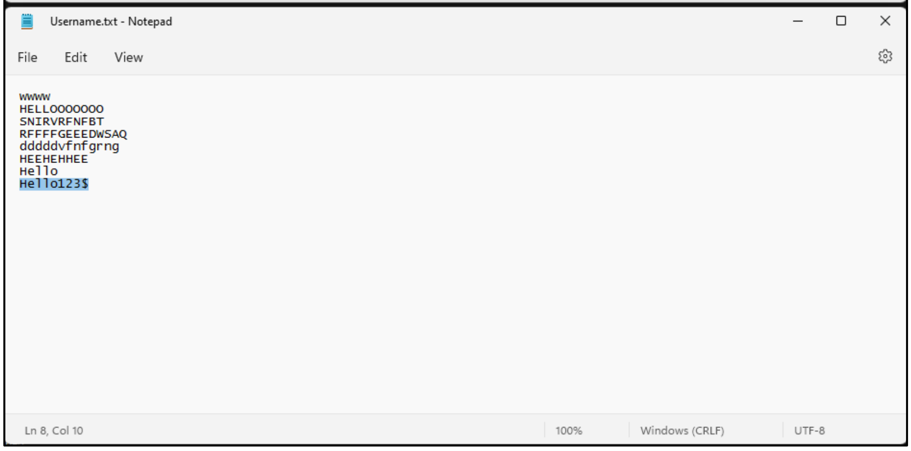


```
------------------------------
|  Welcome to Group B!        |
| What would you like to do? |
------------------------------
-To register a new account, press 1!

-To login into an existing account, press 2!

-To exit program, press 3!

1
Please create a username:

Hello123$
Please create a password:

Bye234%
```

Assuming that the user decides to choose option 1, which is the register option, they will then be presented with an output asking the user to create a username and password so that the user may register into their account. Here, the user may type in their preferred username and password, as in the screenshot above.

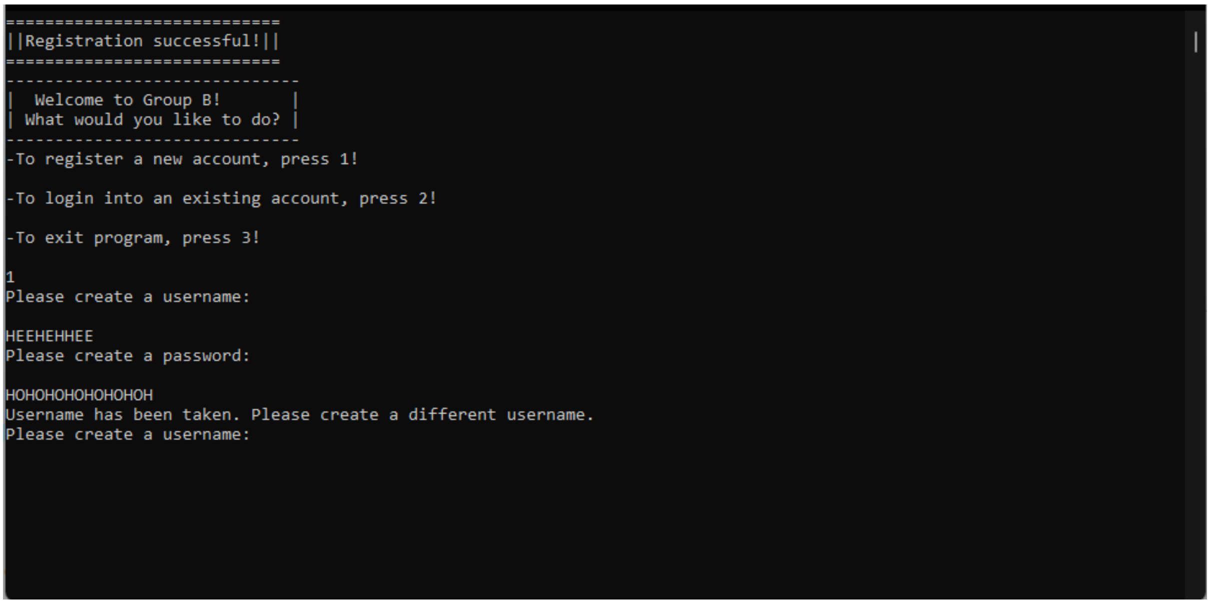


```
33333
ASCBEFJBFJBF
GKTNHKTNBTKNGT
WWEEEDRGF
aaaaaaaaaa
55555
ss
Bye234%
```



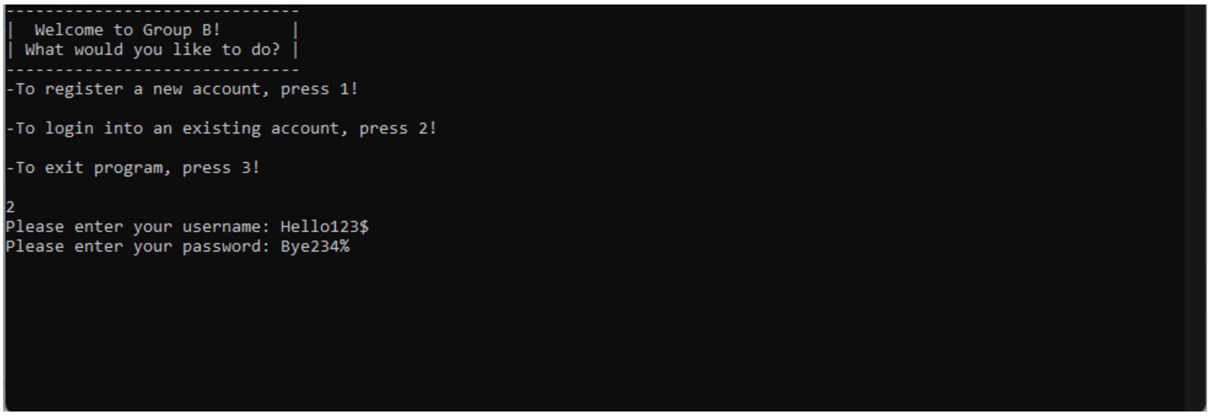
```
wwww
HELLOOOOOOO
SNIRVRFNFBT
RFFFFGEEEDWSAQ
dddddvfnfgrng
HEEHEHHEE
Hello
Hello123$
```

Above, is a screenshot from the two .txt files that have been created for this program, which are the "*Username.txt*" file and the "*Pass.txt*" file, one for saving the username's and the other is for saving the passwords respectively. From the screenshot above of the command prompt, which shows the user type in the username *"Hello123$"* and password "*Bye234%",* the exact usernames and passwords will then be saved into their respective .txt files, as highlighted. Above are various other usernames and passwords that have been tested earlier.



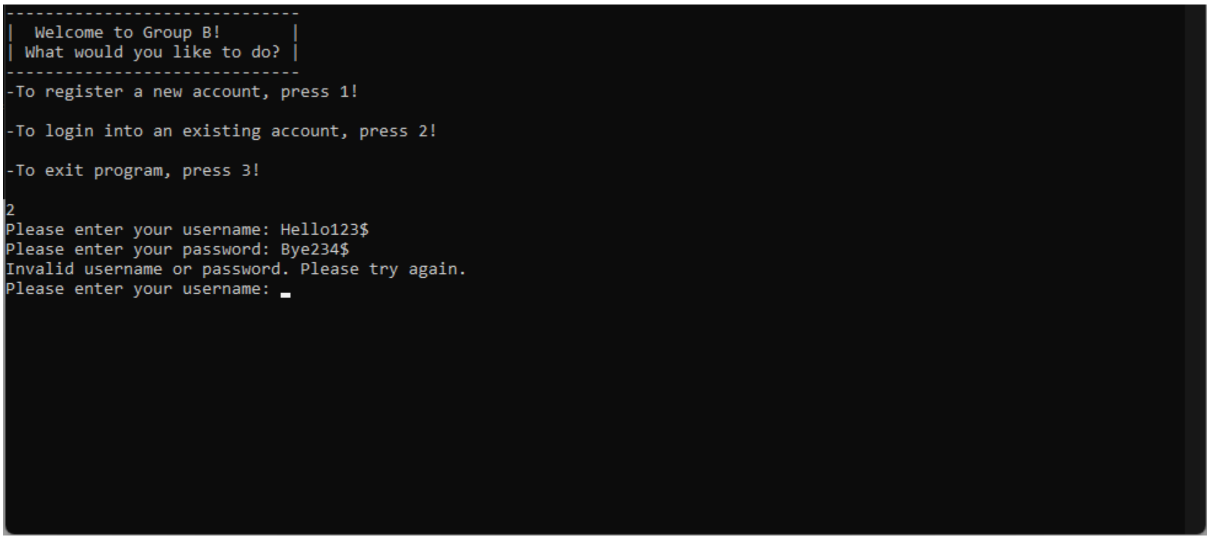
```
============================
||Registration successful!||
============================
------------------------------
|  Welcome to Group B!       |
| What would you like to do? |
------------------------------
-To register a new account, press 1!

-To login into an existing account, press 2!

-To exit program, press 3!

1
Please create a username:

HEEHEHHEE
Please create a password:

HOHOHOHOHOHOHOH
Username has been taken. Please create a different username.
Please create a username:
```

The above screenshot is still related to the registration function; however, this screenshot demonstrates what happens if the user tries to register their account using a username that has already been taken. From the above screenshot, it can be seen that the username “*HEEHEHHEE”* has already been taken as also shown in the screenshot of the .txt file above. Thus, the user will not be able to register their account using this username and will instead have to create a new username, which the program will take them directly to afterward.



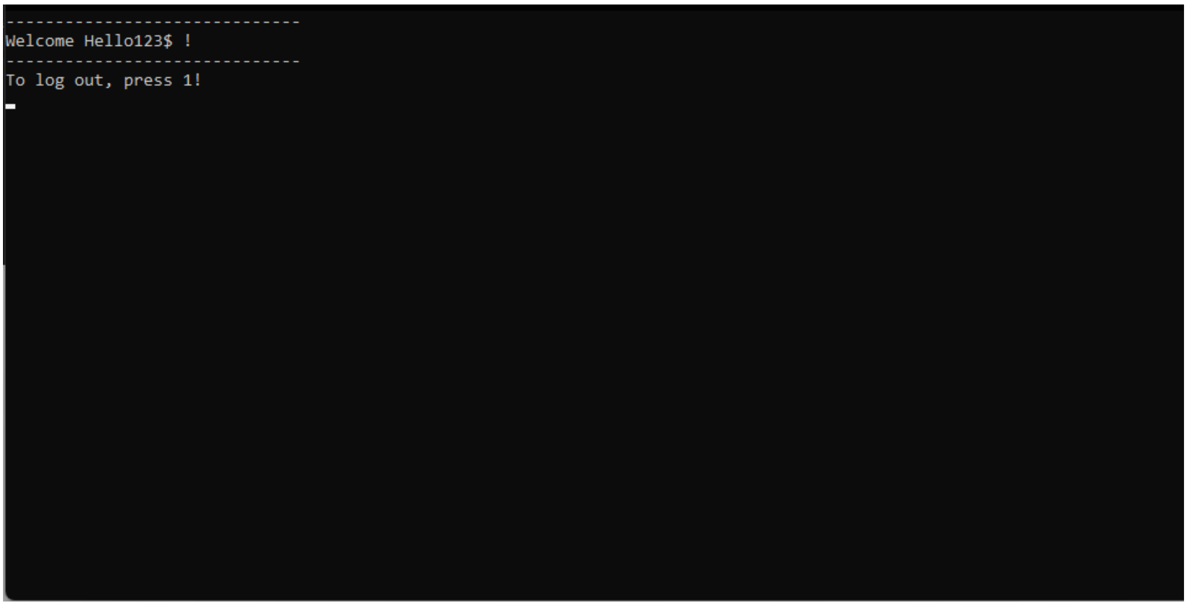
```
-----------------------------
|  Welcome to Group B!       |
| What would you like to do? |
-----------------------------
-To register a new account, press 1!

-To login into an existing account, press 2!

-To exit program, press 3!

2
Please enter your username: Hello123$
Please enter your password: Bye234%
```

The above screenshot now begins demonstrating about the login function. Once the user types in the command “2” into the main menu, which is the option for logging in into their account, the user will then be prompted with output commands to type in their username and password. Here, the user types in the same username and password as seen in the above screenshots when the user was registering.



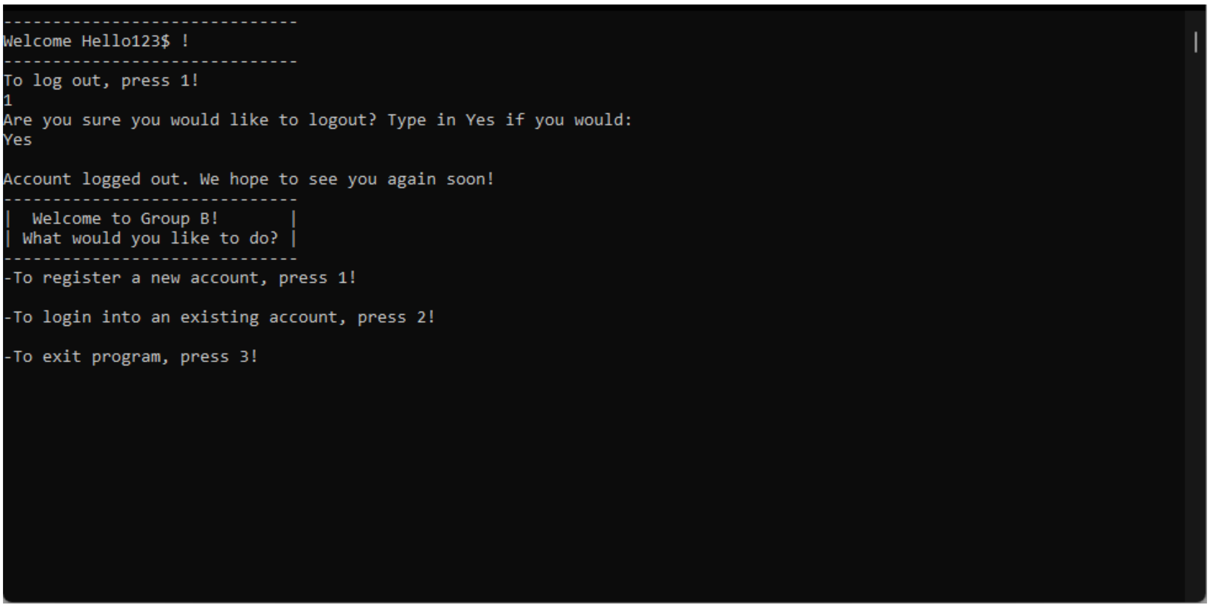
```
-----------------------------
|  Welcome to Group B!       |
| What would you like to do? |
-----------------------------
-To register a new account, press 1!

-To login into an existing account, press 2!

-To exit program, press 3!

2
Please enter your username: Hello123$
Please enter your password: Bye234$
Invalid username or password. Please try again.
Please enter your username: _
```

The above screenshot demonstrates what happens to the program if the user types in a wrong username and password combination. The error is because the user typed in the password “*Bye234$*” when instead the correct password registered was “*Bye234%”.* It can be seen that once a wrong username or password had been entered, an error message will be printed out and the user will be prompted to login again.



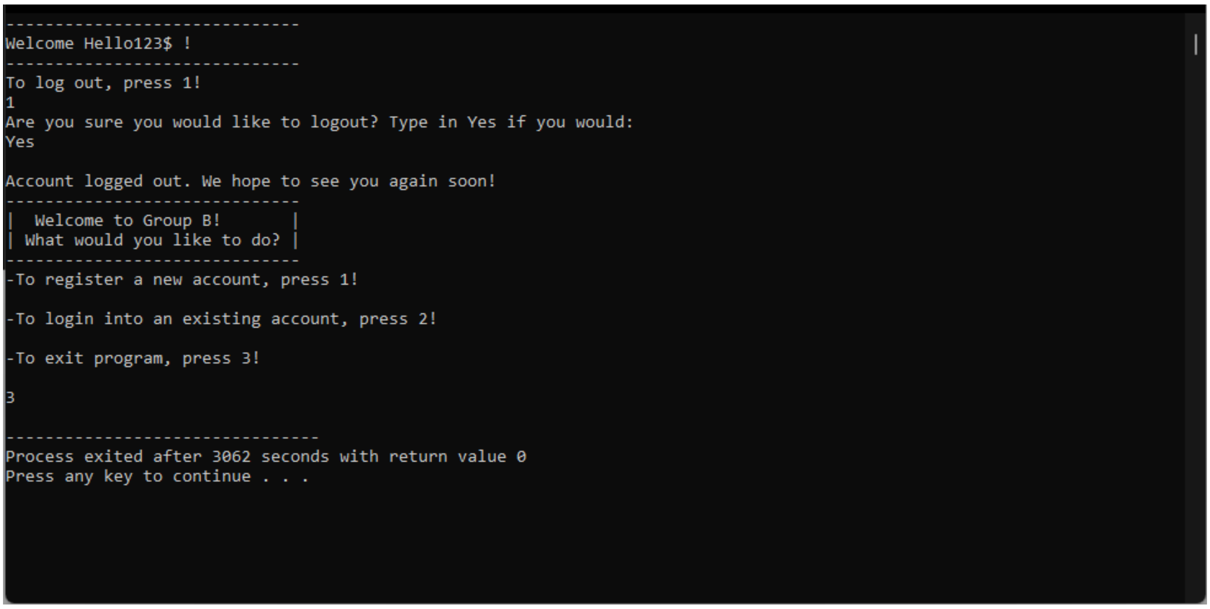
```
------------------------------
Welcome Hello123$ !
------------------------------
To log out, press 1!
```

Thus, once the user (*Hello123$*) has logged into their account using the correct details, they will then be given access to enter their account, where they will see a logout option after a printed welcome message with the user’s username.



```
------------------------------
Welcome Hello123$ !
------------------------------
To log out, press 1!
1
Are you sure you would like to logout? Type in Yes if you would:
Yes

Account logged out. We hope to see you again soon!
------------------------------
|  Welcome to Group B!       |
| What would you like to do? |
------------------------------
-To register a new account, press 1!

-To login into an existing account, press 2!

-To exit program, press 3!
```

The above screenshot demonstrates what happens if the user chooses to logout of their account. Once the user clicks “1”, which is the option to log out of their account, they will then be asked again as another form of confirmation on whether they would really like to logout of their account. Once the user types in “Yes”, they will then be logged out of their accounts.



```
------------------------------
Welcome Hello123$ !
------------------------------
To log out, press 1!
1
Are you sure you would like to logout? Type in Yes if you would:
Yes

Account logged out. We hope to see you again soon!
------------------------------
|  Welcome to Group B!        |
| What would you like to do? |
------------------------------
-To register a new account, press 1!

-To login into an existing account, press 2!

-To exit program, press 3!

3

--------------------------------
Process exited after 3062 seconds with return value 0
Press any key to continue . . .
```

And finally for this program, the screenshot of the exit program will be demonstrated. Once the user has chosen to click on option “3”, which is to exit the program, the program process will end up being terminated and the user will the program will then stop.

References

- Chrysanthus Forcha (2021). How to Use C++ fstream. https://linuxhint.com/use-cpp-fstream/
- Ravikiran A S (2022). C++ Switch Statement: Advantages, Syntax, and Examples. https://www.simplilearn.com/tutorials/cpp-tutorial/cpp-switch#:~:text=The%20switch%20statement%20in%20C%2B%2B,the%20control%20of%20the%20execution.
- Introduction to C / C++ Programming File I/O. https://www.cs.uic.edu/~jbell/CourseNotes/CPlus/FileIO.html#:~:text=ifstream%20is%20an%20input%20file,out%20to%20a%20data%20file.
- Base class for streams (type-dependent components). https://cplusplus.com/reference/ios/ios/
- Introduction to C++ ofstream. https://www.educba.com/c-plus-plus-ofstream/