

# Report-Chat App

10/01/2021

## Overview

The project was made for the course of C# in the semester 7 at EFREI PARIS. I made a Chat app with TCP that can send messages, create topics, join topics, and send private messages.

## Functions implemented

1. **Login** : The user can login to his account previously added.
2. **Register** : The user can register easily by filling their pseudo, mail and password.
3. **Connect** : The user can connect to the server, all the users and topics will be automatically added.
4. **Create topics** : The user can write in the chat /create [name] or click the '+' button to add a topic.
5. **Join topics** : The user can write in the chat /join [name] or double click on a topic to join the current topic with a new window that will open. It is possible to have multiple topics open at the same time.
6. **Send a message on a topic** : On the topic's window, it is possible to write a message to all users in the topic chat.
7. **Send private message** : It is possible to send a private message to another user by writing on the chat /private [name] or by double clicking on the name of the user in the users list. A new window will be created.

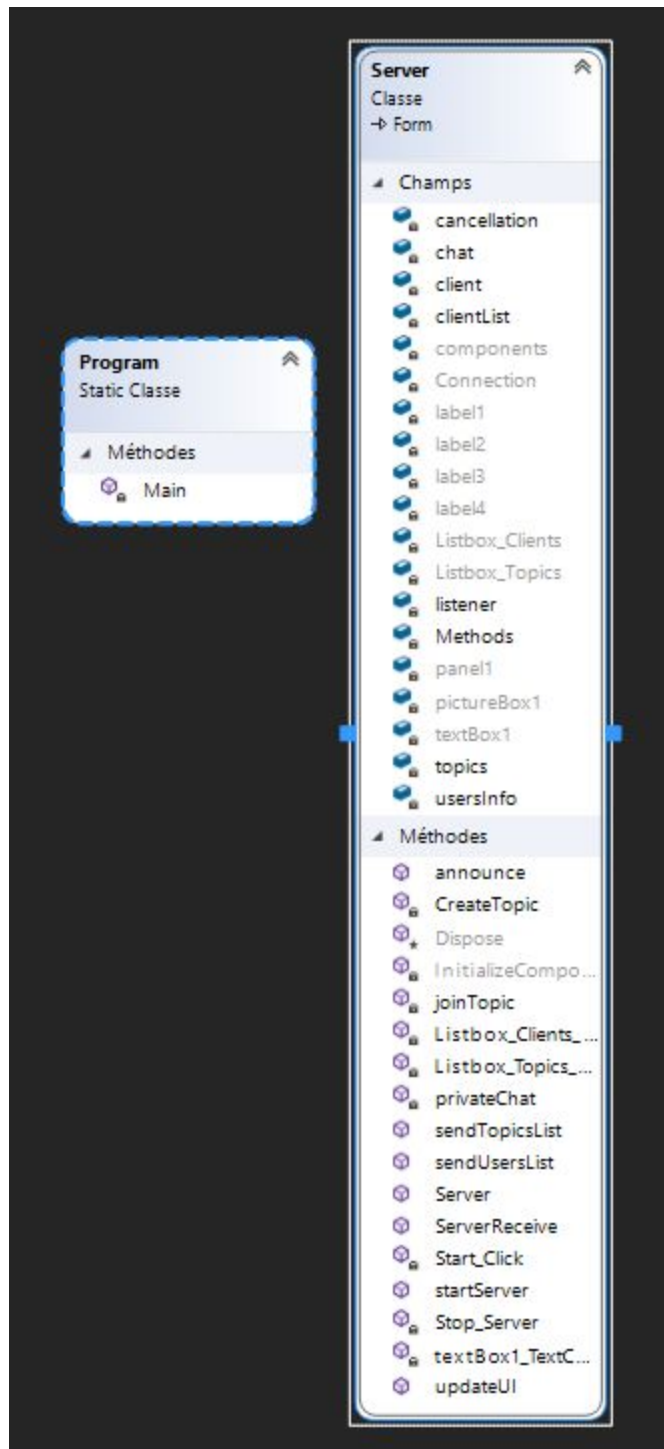
8. **Status of the connection :** A panel is here to tell you if you are still connected to the server, if it is green you are connected, if not the panel will be red.
9. **About :** By clicking on the button “About” in the register window, you will be able to see all the information about the App and the GitHub link.
10. **Graphical interfaces :** All the app was done with a graphical interface so it will be easier for the user to navigate through the app.

## Possible improvement

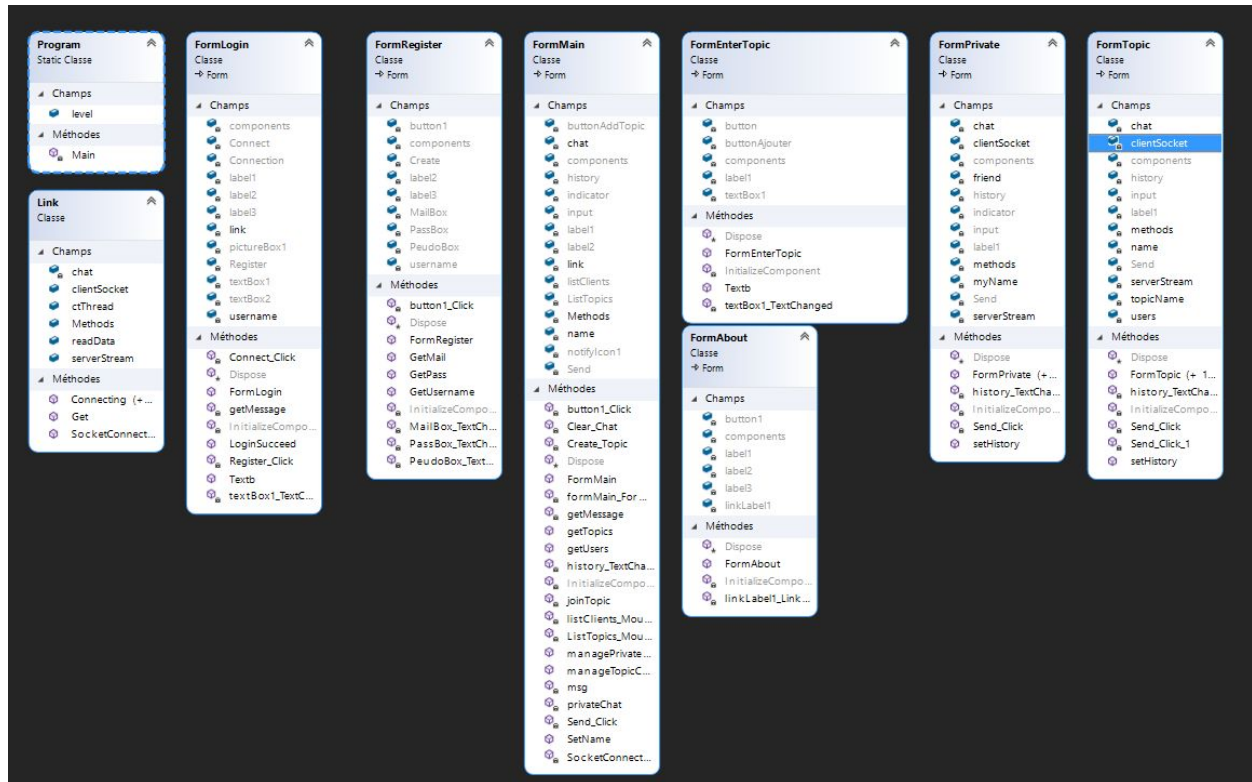
1. The graphical interfaces interface could be better.
2. Some new things could be added to have a better user experience.
3. All the users registered could be added to a local files to save all the datas in case of a crash from the server. For now, the server is supposed to run 24/24h and never stop. Also in the project description, there was no instruction about that so that is why i made it like that.

## The design and the implementation

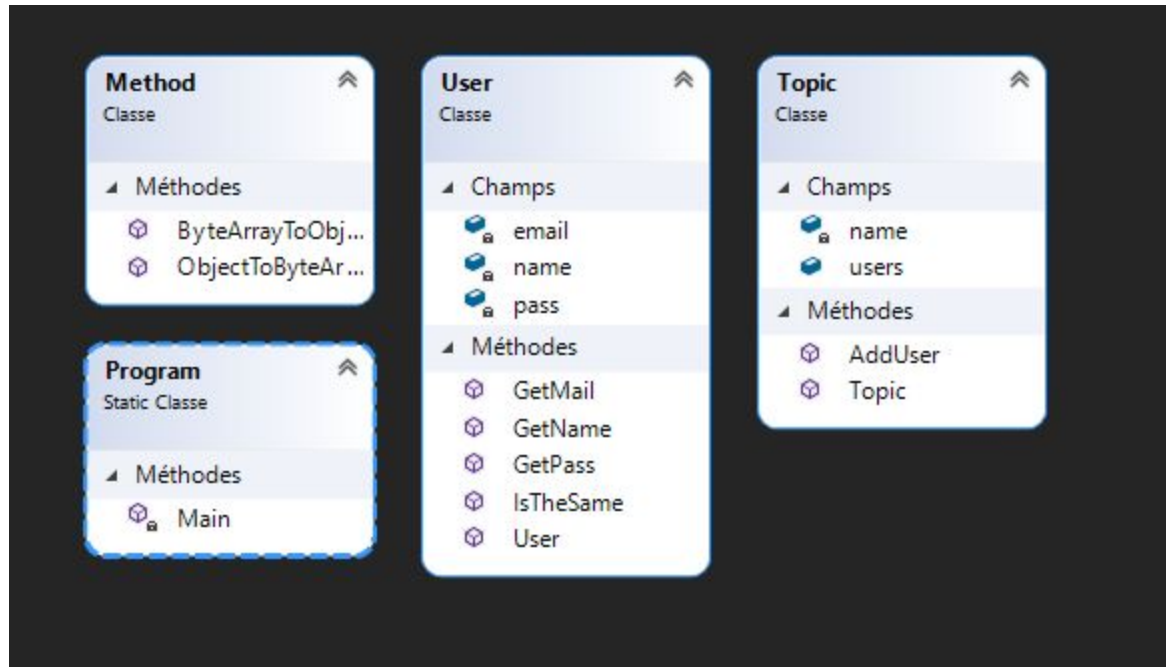
First, here is the diagram server, with a system of multithreading it will organize all the clients behaviour.



Second, the client visual diagram :

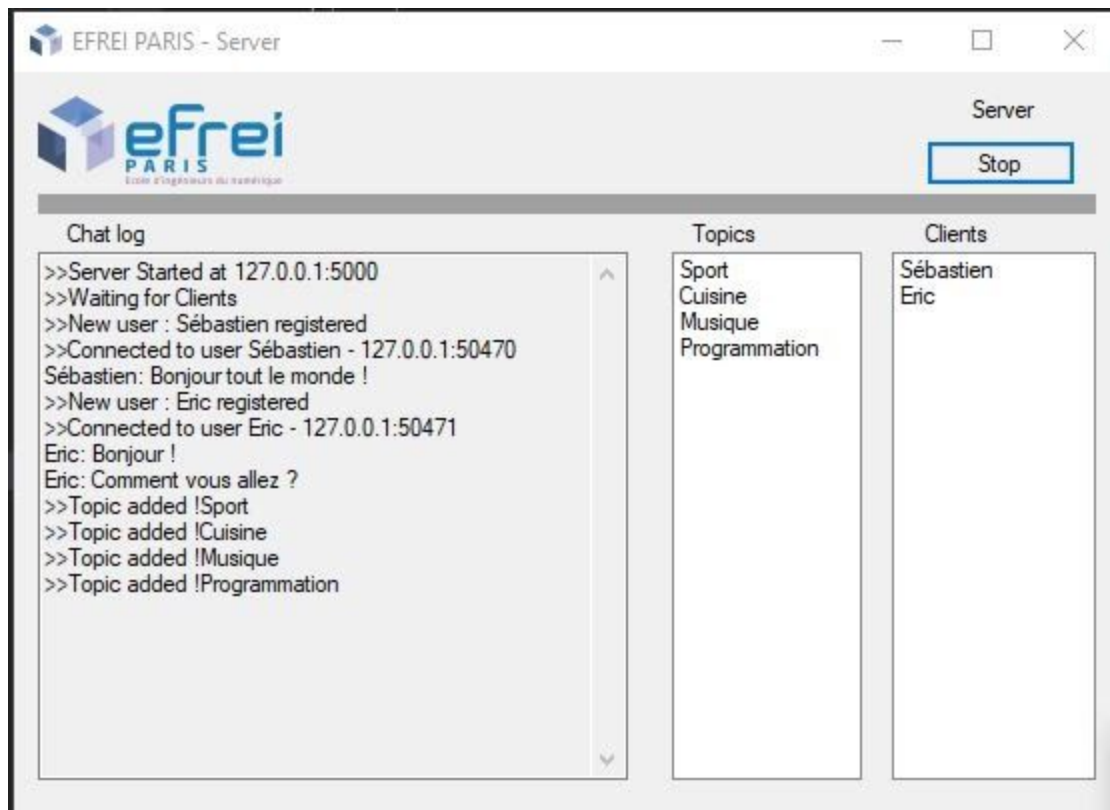


Third, the communication class

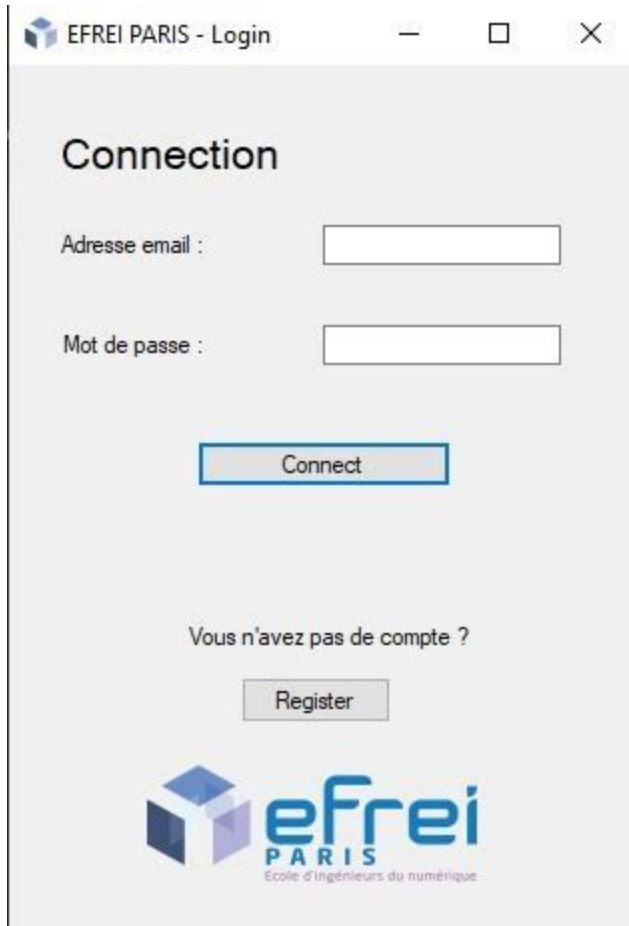


## Screenshot from the app

Here is a screenshot from server :



Login :



The image shows a web browser window titled "EFREI PARIS - Login". The window has a light gray background and contains the following elements:

- Header:** The title bar shows "EFREI PARIS - Login" with standard window control buttons (minimize, maximize, close).
- Section Header:** The word "Connection" is displayed in a large, bold, black font.
- Form Fields:** There are two input fields. The first is labeled "Adresse email :" and the second is labeled "Mot de passe :". Both fields are empty and have a light gray border.
- Buttons:** Below the password field is a blue button with the text "Connect". Below the "Connect" button is a link that says "Vous n'avez pas de compte ?". Below this link is a gray button with the text "Register".
- Logo:** At the bottom of the window is the EFREI PARIS logo, which consists of a stylized blue and white cube icon followed by the text "efrei PARIS" in a bold, sans-serif font. Below the logo, the text "Ecole d'Ingénieurs du numérique" is written in a smaller font.

Register :



A screenshot of a Java Swing window titled "FormRegister". The window has a standard title bar with a minimize button, a maximize button, and a close button. The main content area is light gray and contains three text input fields stacked vertically. Each field is preceded by a label: "Pseudo :", "Adresse email :", and "Mot de passe :". Below the input fields is a blue "Create" button. At the bottom of the window is a gray "About us" button.

FormRegister

Pseudo :

Adresse email :

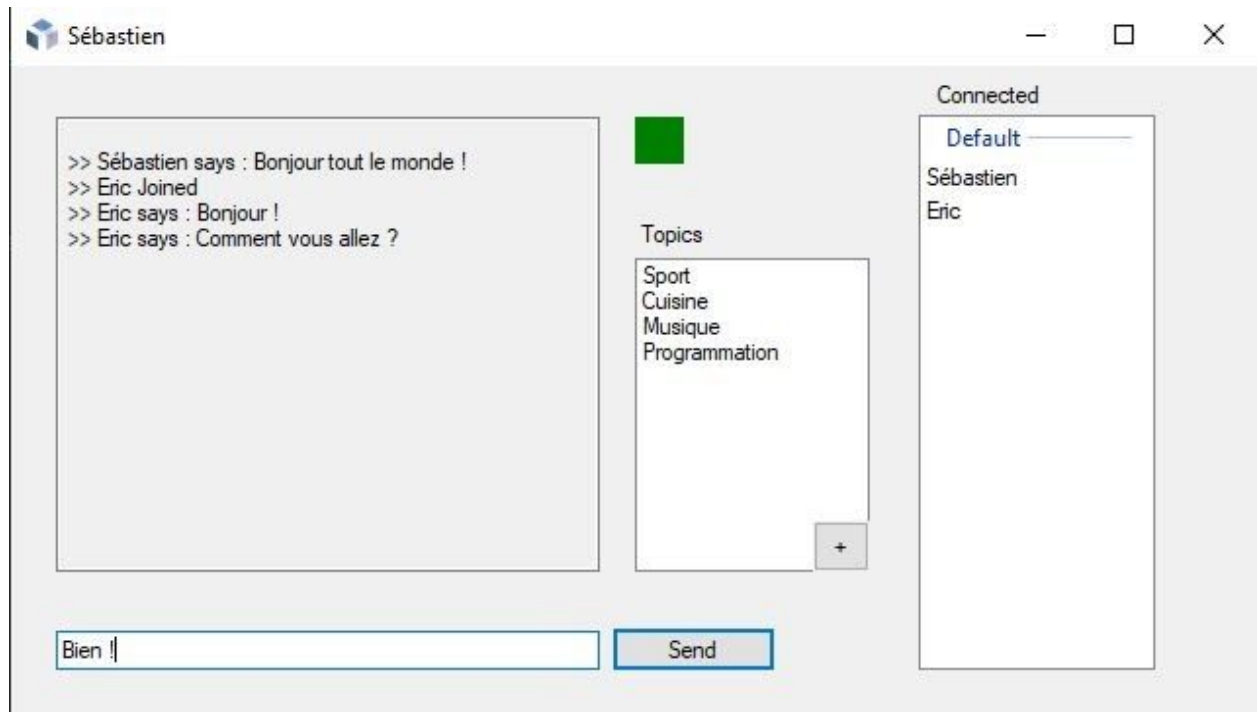
Mot de passe :

Create

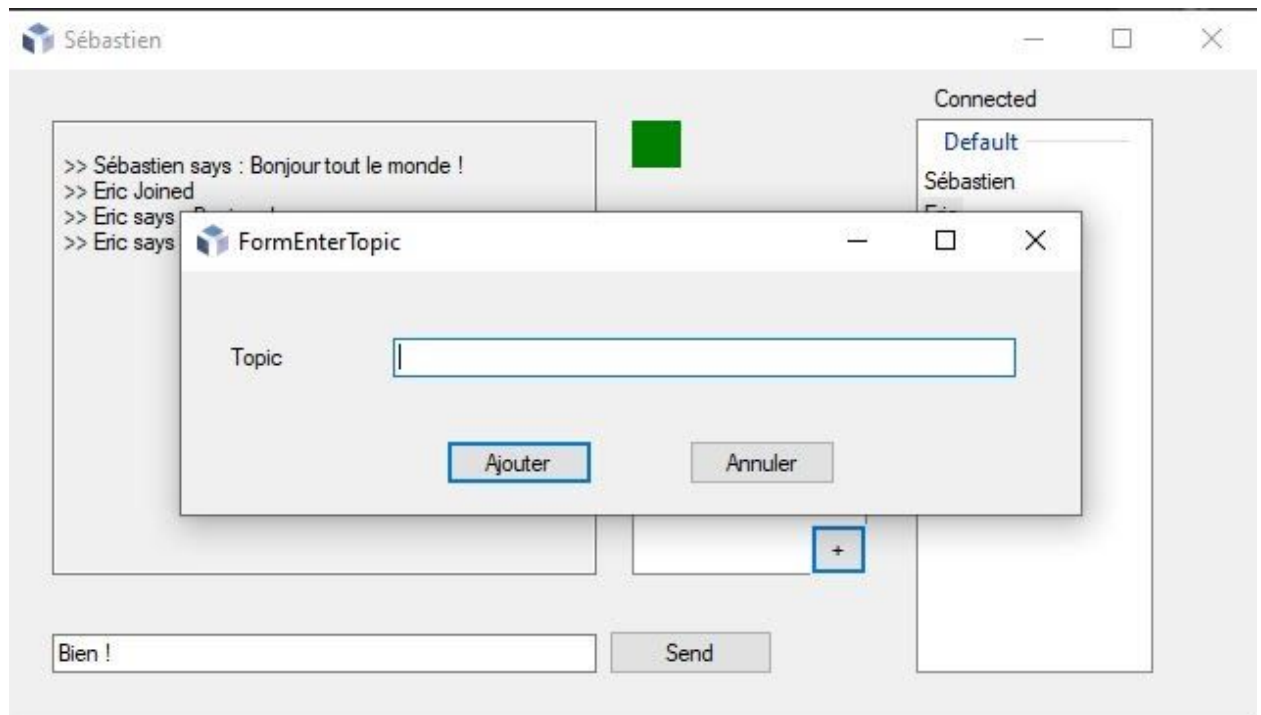
About us



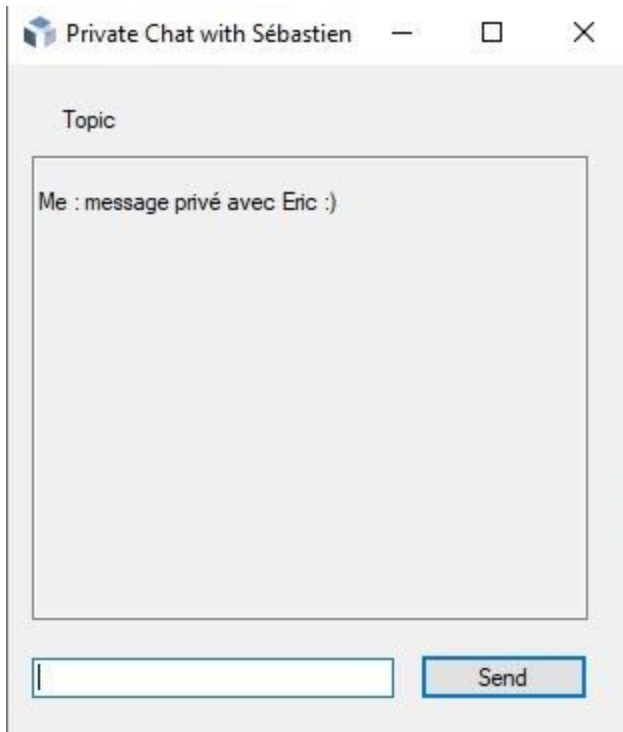
The client app :



Create a topic :



Create a private message :



The image shows a screenshot of a web application window titled "Private Chat with Sébastien". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there is a section labeled "Topic" which contains a text area with the message "Me : message privé avec Eric :)". Below the text area, there is a text input field with a cursor and a "Send" button.

The topic :

