Documentation for programmer monitoring interface

Claudel Louis RT212

TABLE OF CONTENTS

This documentation is intended for people who know the basics of programming.

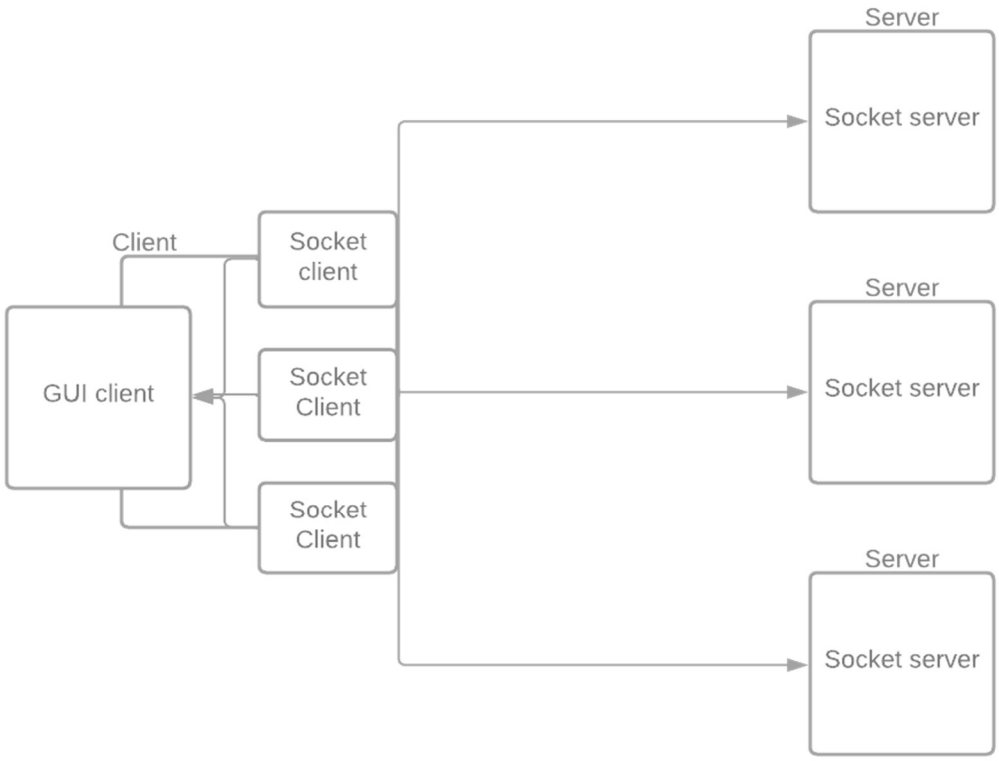
In this document, I will explain the structure of the application.

In a second time, I will explain you how I have structured my code

In a third time, I will explain if I have fully met the specifications.

STRUCTURE OF THE APPLICATION

After this little introduction, I'll show you the structure of the application :

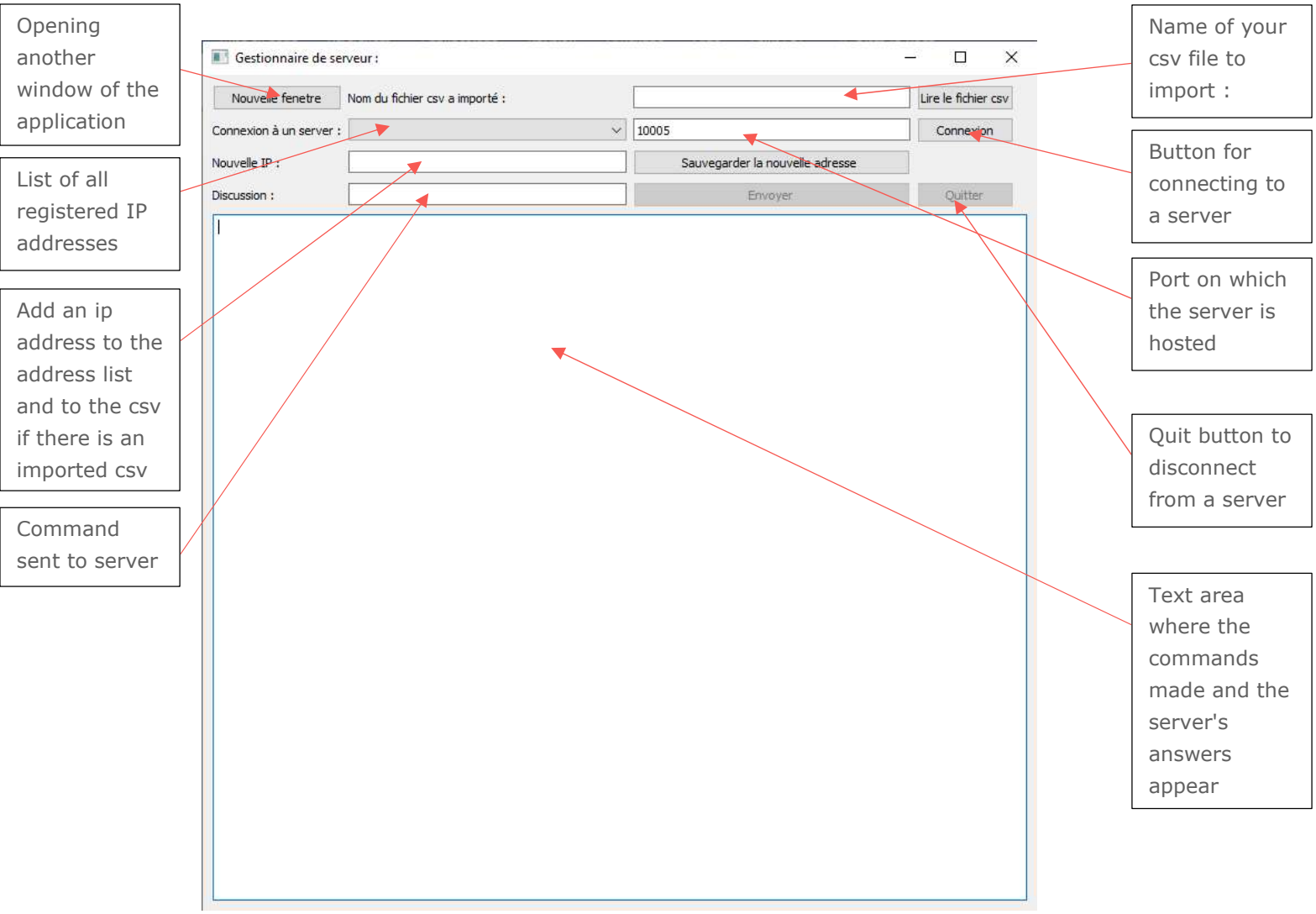


The interface was built using python, the PyQt5 library for GUI, psutil library for server. You have to get the file python client.py and server.py which is on github. link from github :

https://github.com/Grievous400/R309/tree/master/Sae

Client interface architecture :

Let me introduce you to the Mistletoe on the client side:



By default, the port is 10005 for the client and the server. You can change the port if you want.

You can read a csv file with a list of ip addresses. The format of the csv must be one address per line. You can add them via the interface. If a csv has been read when you add ip via the interface, they are directly added to the csv file.

MY CODE

In this part, I will explain my code for client and server. The discussion between my socket server and my socket client is synchronous.

1.Client code

My client code is divided into two main parts. The first part is the socket class and the second part is the MainWindow class which is the GUI.

I'll start by introducing the socket class which is from line eight to forty-five. The socket class is inherited from threading. Thread.It is also composed of four functions which are the __init__ function, the "client_connect" function which is used to connect to a socket server, the "send" function which is used to send and receive messages from the server and finally the close function which is used to disconnect from the server. There are also the getters and the setters for each attribute of the socket class.

The MainWindow class is inherited from QMainWindow. The first function is __init__ in which we can find the declaration of all the elements of the GUI, their placement and the actions related to the different buttons. The other functions of this class are the functions related to actions. We have the connection function which manages the connection to a server, the "send" function which sends a message and receives it, the "quit" function which is used to disconnect from the server, the "newa" function which is used to add a new address to the list, the "lirefichiercsv" function which is used to read a csv file and adds these addresses to the list and finally we have the "create_new_document" function which is used to create other clients.

2.Server code

The server has a great "discussion" function. It processes the various commands requested and returns the answers. In this function, we can see that the messages received are tested several times so that the value returned is the best possible. There is also the declaration of the socket server.

WHAT I DID AND DIDN'T DO WELL

server side:

I managed to set up a server that is able to handle all possible commands. I tested it on windows and on a debian VM. I didn't have time to test it on a mac VM. The only command that doesn't work is the kill command. Only one client can connect to a server.

client side :

My GUI can read a csv file and can add ip addresses to it. From a GUI, you can open several client windows and each of them can connect to a different client. The GUI can send commands and receive their answers. It puts them in a history. When the commands are wrong, the server takes some time to respond which crashes the GUI but there is no need to restart it.

I didn't have time to do the extra options