**Happy chatting room**

**Description:**

This is a virtual chatting room serving for two or three people communicating with each other. The chatting pairs must run this program on their computer simultaneously. Once they establish connections with each other, they can share text messages.

**User Interface:**

* **Part one (Establish connections):**

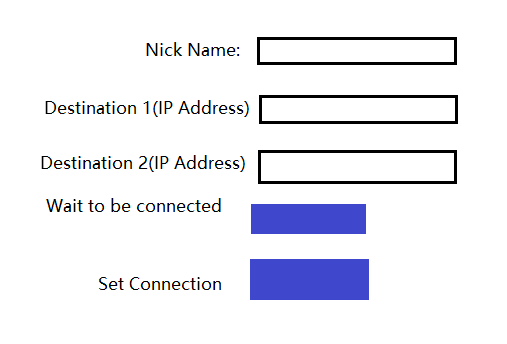
There should be the following functions:

1: Input a nick name representing the host when chatting with others.

2: Input other users’ IP addresses to be connected to.

3: There should be at least two buttons. One button is for initializing connections with other users. Another is for waiting the connection request from other users. These two buttons corresponding to different state of your host.

It looks like the following figures:



* **Part two (virtual chatting room):**

There should be the following functions:

1: Display the message that the users share with each other.

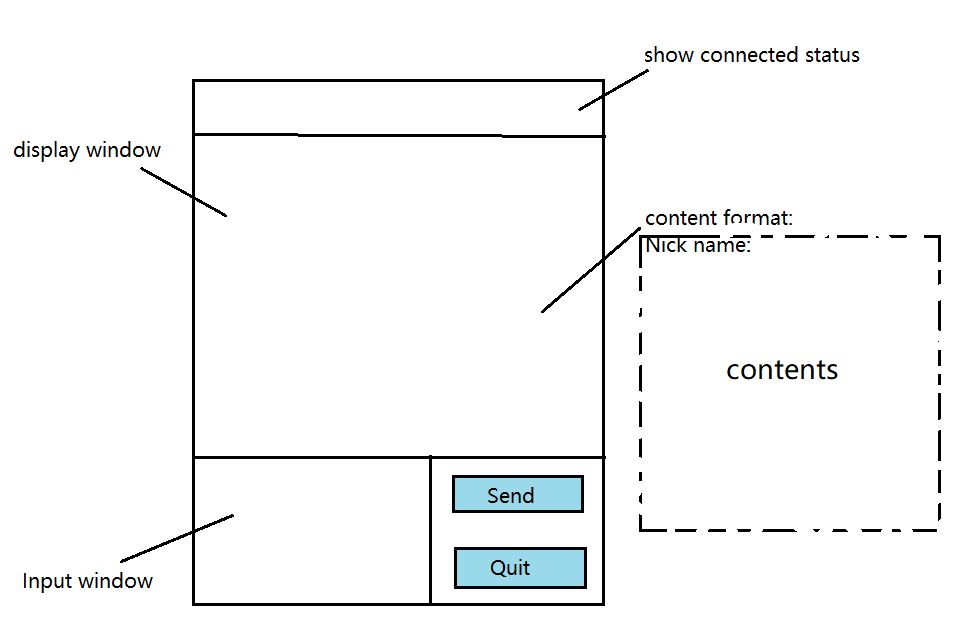
2: Display the connection status. ( who are active in this chatting room now.)

3: Input the text message and send it to other users.

4: The format of the message should be: Sender’s nick name + contents

5: At least two buttons, one is for sending the message that user inputs, another is for quitting the chatting room

It looks like the following figure:



**How does this application work?**

Three users must run this application simultaneously. They would enter the first part of the UI. Two users need to offer their nick name and click on “Wait to be connected” button. Another user must be the initializer who need to offer the nick name, IP addresses of other users, and click on Connect button. Once they are connected, they would all enter into the second part of the UI. Then they can start sharing message!

**How to set up connections among these three users?**

The initializer would use TCP protocol to set up connections with other two users respectively.

**How to transport packets among them?**

* **For the initializer:**

All the packets that the initializer received or packaged are putting into a queue. And the initializer periodically pulls a packet from the queue and sends that packet to other two users. Then the initializer extracts data from that packet and displays the data on the UI.

* **For other two users:**

(1) Once a user sends a message out, the message would be packaged into one or multiple packets being sent to the initializer.

(2) Once a user receives a packet from the initializer, it will extract the data from that packet and display it on the UI.

This strategy is designed to guarantee the messages displayed in each user’s UI are same to each other.

**Error recovery:**

If the initializer can only get connected to one user, they can still enter into the virtual room. And the initializer will continue trying to connect to another user.

If the initializer cannot get connected to any users, there should be an alert: connection failed!