

For the bonus, we implement the live connection and disconnection of users.

We use two function login and logout.

For the logout, we use the code shows as below to stop the thread but we don't delete the information about the user in the server.

```
GenServer.stop({name_of_node, Node.self()}, :normal, :infinity)
```

And for the login, we start a new node and update the information of the user for the node.

```
{:ok, _} = GenServer.start_link(__MODULE__, server_node_name, name: name_of_node)
GenServer.cast({name_of_node, Node.self()}, {:update_client_state, name_of_node, Node.self()})
#query_latest_tweets(nodename)
```

The complete shows below:

```
def login(nodename, _password, server_node_name)do
  name_of_node = String.to_atom(nodename)
  {:ok, _} = GenServer.start_link(__MODULE__, server_node_name, name: name_of_node)
  GenServer.cast({name_of_node, Node.self()}, {:update_client_state, name_of_node, Node.self()})
  #query_latest_tweets(nodename)
  {name_of_node}
end
```

```
def logout(nodename)do
  name_of_node = String.to_atom(nodename)
  GenServer.stop({name_of_node, Node.self()}, :normal, :infinity)
end
```