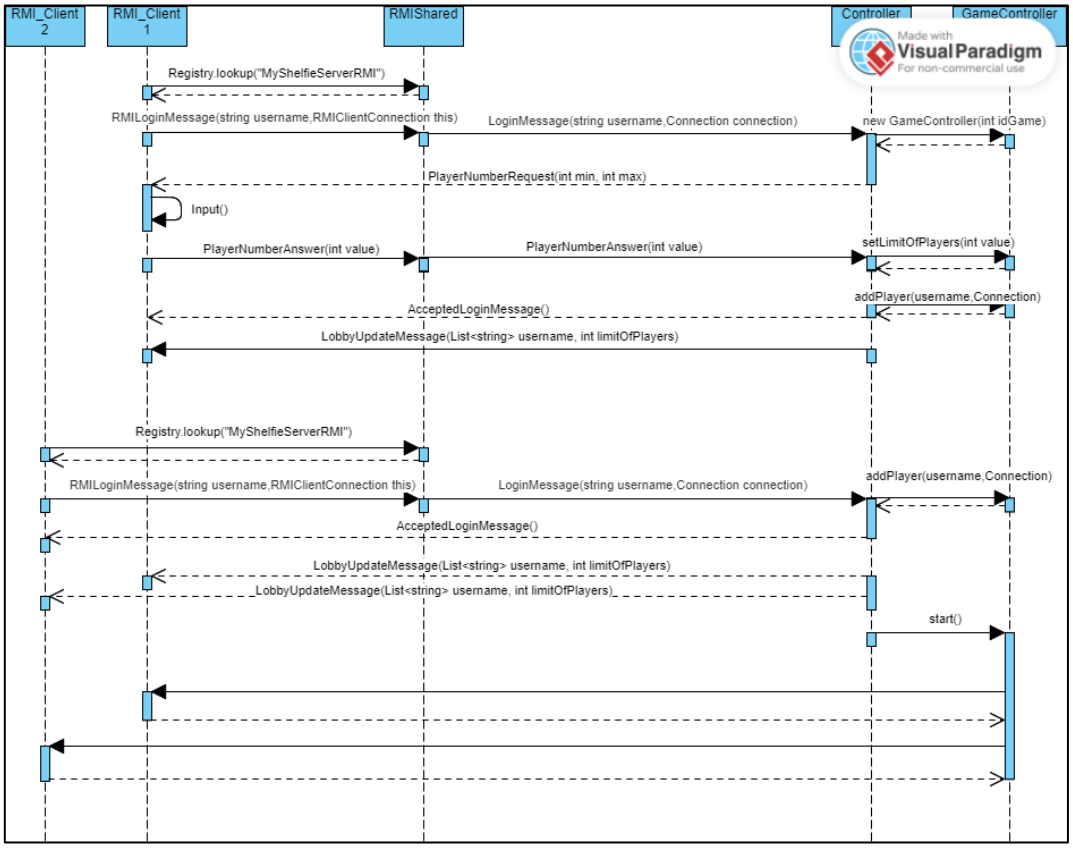
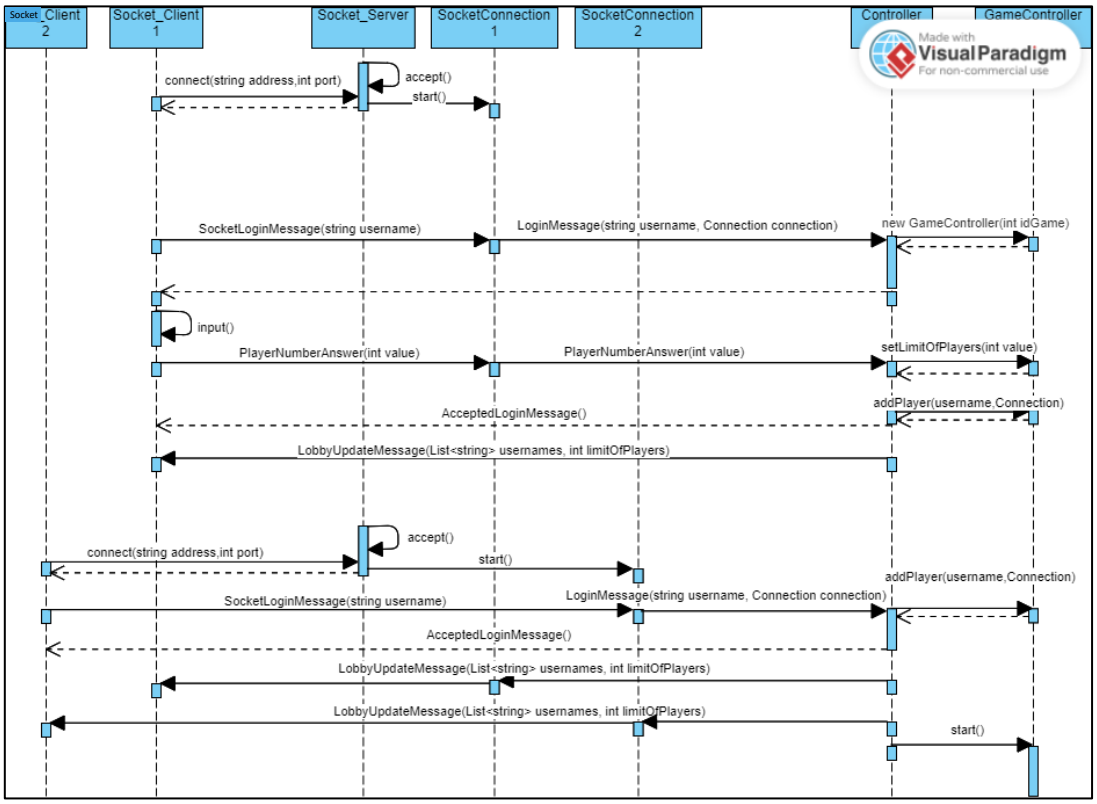
1) A player joins the game:
   The only message we need to integrate are the login messages, infact RMI and Socket have a completely different way to connect beetween client and server.

RMI



In the above diagram we can see the creation on a lobby of 2 players using RMI.
RMI needs to put the client itself in the message in order to be called by the server.
The first player will be asked for the number of players to play (between max and min), as he answers the Controller will add him in the lobby and send an ACK. **(AcceptedLoginMessage)**
Every time a player joins the lobby, the controller will notify all other players joined with the current List of names and the size of the lobby. **(LobbyUpdateMessage)**

SOCKET



Almost the same process is done with socket clients, the only thing that changes is the login process. A Socket_Server is always accepting new connection by new players. It generates a new thread for each of them. This thread will behave as a gateway between client and server.
SocketLoginMessage don't need a Connection interface, it will be built on relative SocketConnection.

The Controller will manage the creation of the lobbies; once a lobby is full the control of the game is delegated to **a Thread(GameController)** who owns all the names and all the connections to talk with its players.
Clients will talk directly to GameController, forgetting about the Controller who only accept login requests and create the lobbies.

NOTE : Obviously the clients will need to change their server references to talk directly to GameController, this process is not shown in the diagram.