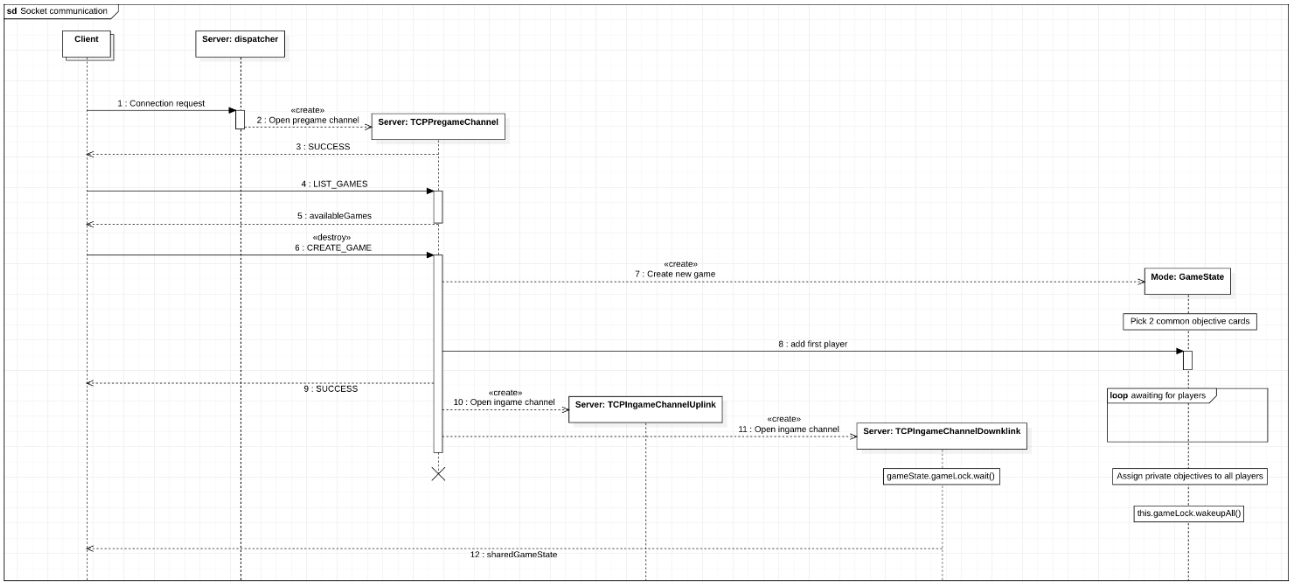
1.1 Access of the player to the game – Create Game



When the Connection request message arrives from the client, the server creates a new TCPPreGameChannel; its function is to communicate with the client before the game starts, sharing on request the list of available games or creating a new game, and after that, it is destroyed. TCPPreGameChannel interacts with the Model sending a <<create>> message that entails the instantiation of a new GameState (class representing a specific game), and a second message that enables the addition of the first player. Messages 10 and 11 involve the creation of TCPIngameChannelUplink (Client to Server communication) and TCPIngameChannelDownlink (Server to Client communication). Downlink channel, synchronized on GameState's lock, awaits until all players are connected and at that point makes every client receive the first personalized SharedGameState.

Additional notes:

- Throughout all these transmissions, the client is also sending regular *keep-alive* pings to the server to avoid channel closure. These messages have been omitted from the diagram to avoid cluttering.
- jRMI initial connections occur by simply requesting the "*remotePreGame*" remote object.
- As opposed to having TCP uplink and downlink channels, jRMI in-game transmissions make use of a specific remote object: "[*gameId*]/[*username*]".
- In order to obtain first SharedGameState, jRMI clients also call the *registerConnection* remote method after succesfully joining a game.
- jRMI clients do not send *keep-alive* pings while in the pre-game phase; after having succesfully connected to a game, however, also jRMI clients start pinging the server regularly.