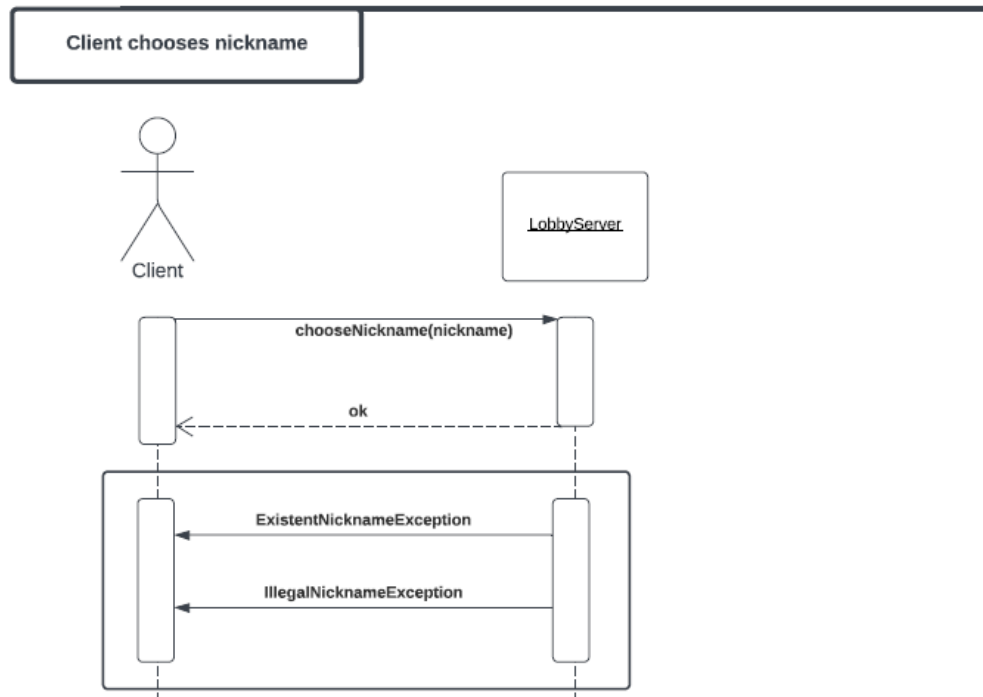


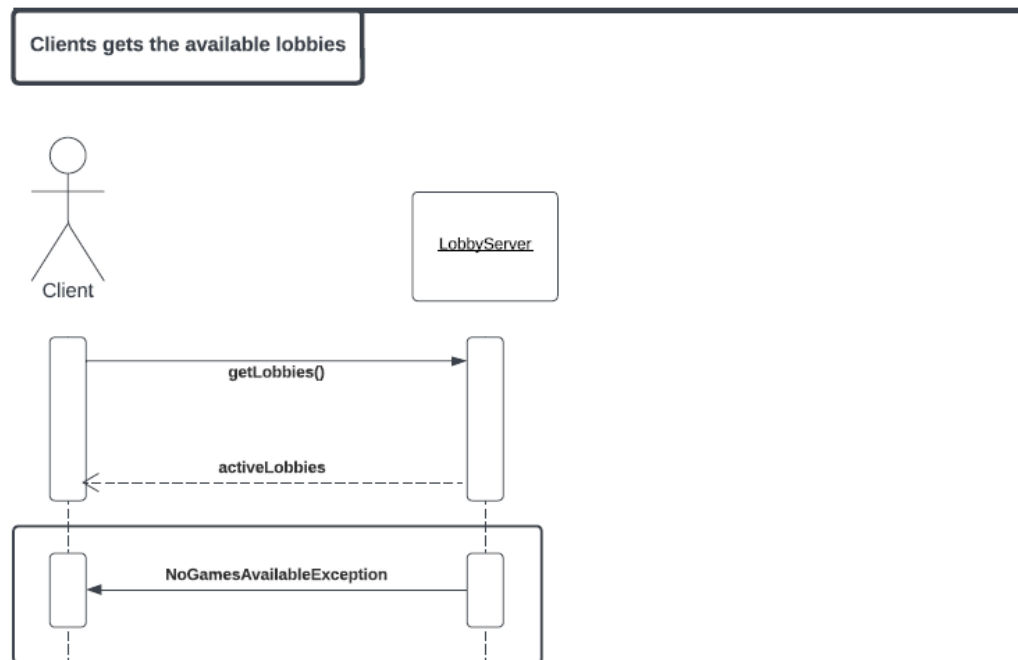
SEQUENCE DIAGRAM

- Client chooses nickname:



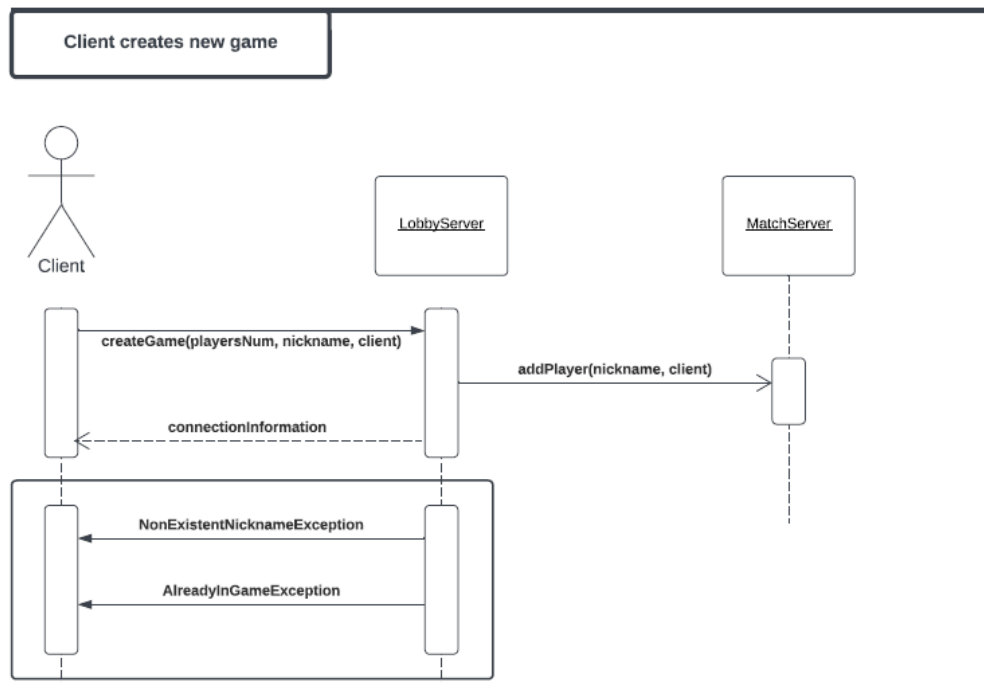
When the player types his nickname the method chooseNickname is called on the server, which will throw an exception if the nickname is invalid (already chosen or banned).

- **Client asks available lobbies**



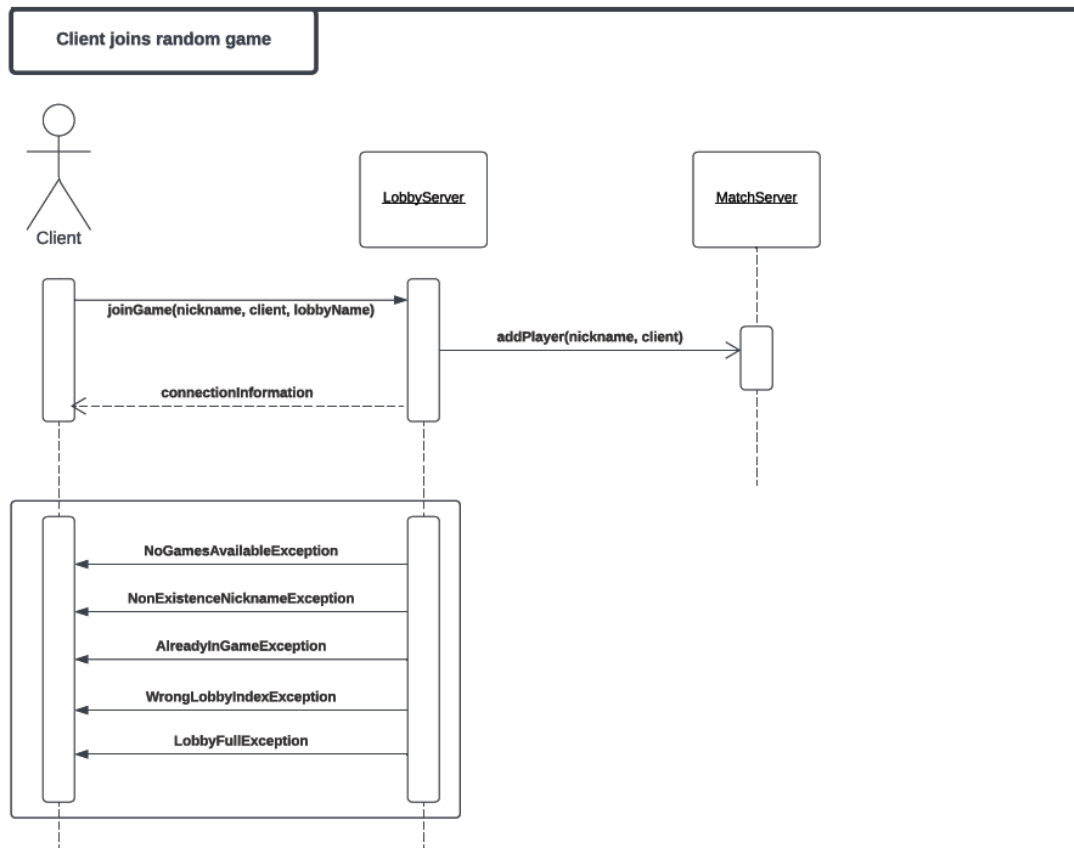
Before joining a game the client must select which lobby he wants to join. With the command “`getLobbies()`” it can ask the lobby server which are the available ones. An Exception is thrown if there are no available games.

- **Client creates a new game:**



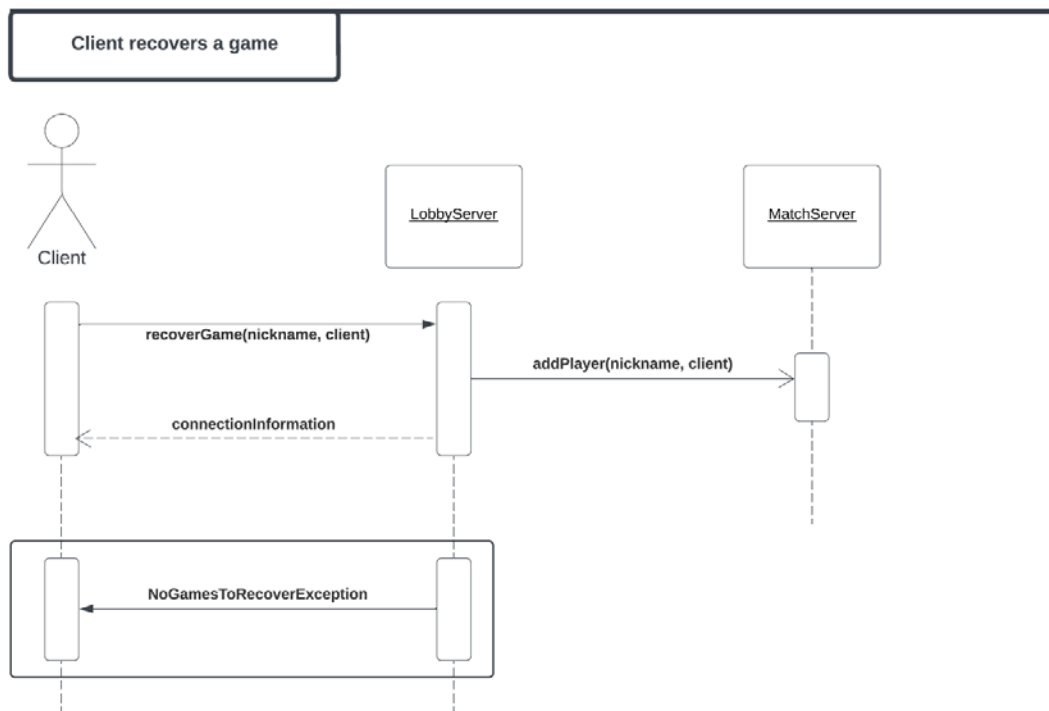
When the player selects the option “create new game” he must select the number of players, then the method `createGame` is called on the `LobbyServer`, which creates a new `Match` server, adds the player to it and returns the connection information for the `MatchServer` to let the client connect. Exceptions are thrown if the player is not registered or if a player with that nickname is already in game.

- Client joins an existing random game:



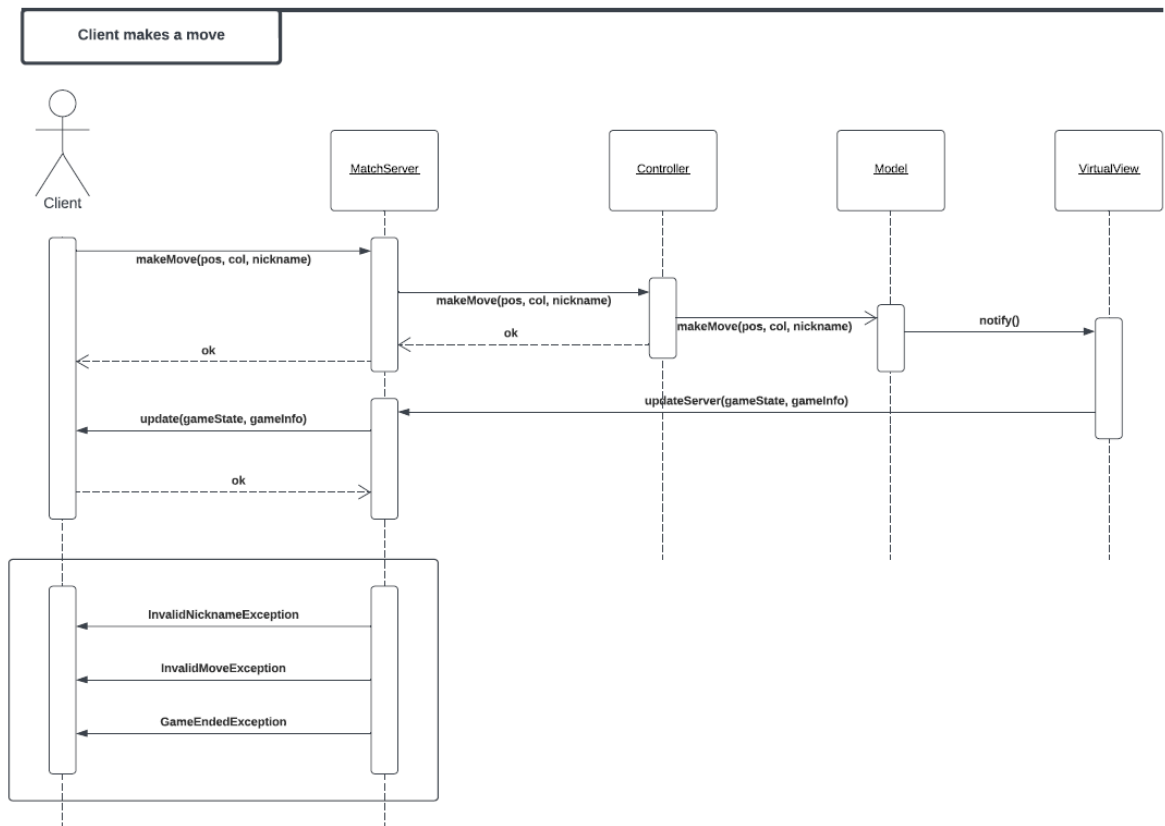
Same as the previous interaction but in this case the player selects the option “join an existing game” and must select the lobby in which to join. Extra Exceptions are thrown if the player selects an index out of bounds or the lobby is already full.

- **Client recovers an existing game:**



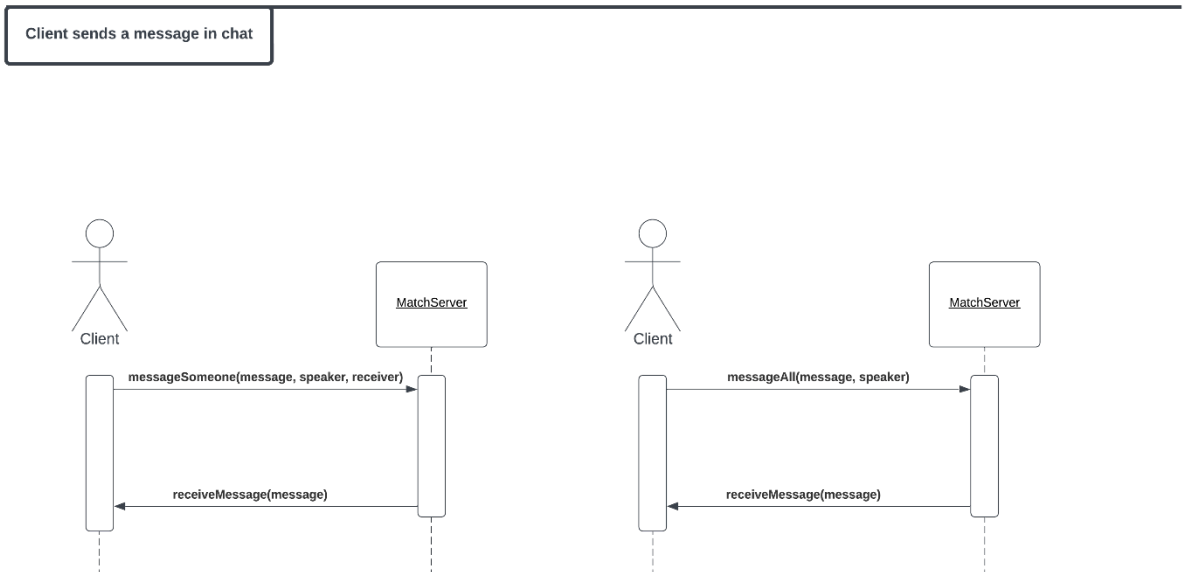
Same as the previous interaction but in this case the player selects the option “recover an existing game”, this can only be done if there is a game to recover (i.e. the server has crashed). If there are no game to be recovered then the `NoGamesToRecoverException` is raised.

- **Client makes a move:**



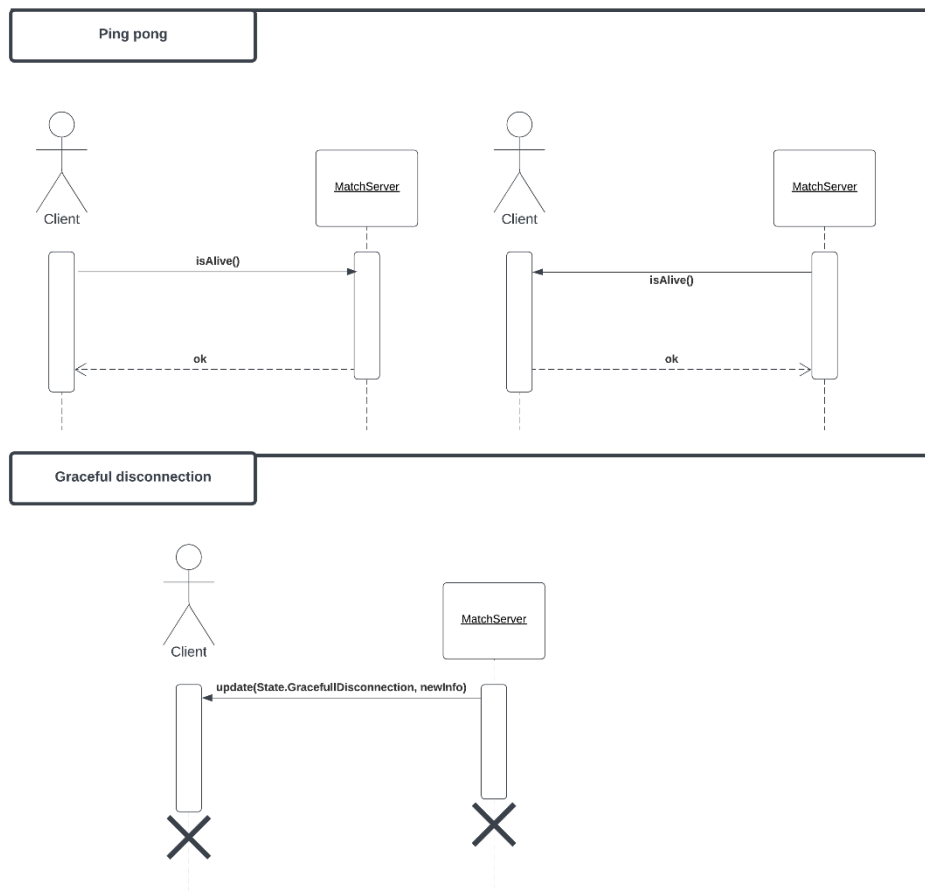
When a player selects a move he calls the `makeMove` method on the `MatchServer`, which calls the corresponding method on the controller and updates the model accordingly. When the model is updated it calls an update method on a `VirtualView` which creates the new `gameState` and `gameInfo` based on the changes the player's move caused and calls the `updateClients` method on the `MatchServer` which will call the update methods on all the clients. Exceptions are thrown if the nickname is invalid, the move is illegal or the game has already ended.

- Client sends a chat message



When the player wants to write a message in the chat he must select if the message is global or directed to a specific player, this will change the method called on the server, in case of “messageAll” the server will call the receiveMessage method on all clients, otherwise only on the player that is supposed to receive the message.

- Ping pong and disconnection



When the connection between the client and the server is established a ping pong starts from both sides calling the `isAlive` method on the other side independently, in this way both are able to detect if the connection is lost, in this case (when the `isAlive` method throws an exception) the server will notify all the other clients that one is not connected anymore and close the game. The disconnected client, thanks to its ping, detects that it can't reach the server and automatically set its state to `State.GRACEFUL_DISCONNECTION` and closes itself.