

Network Communications

The class diagram of network and controller can be found in the deliverables folder in "UML controller+network.pdf".

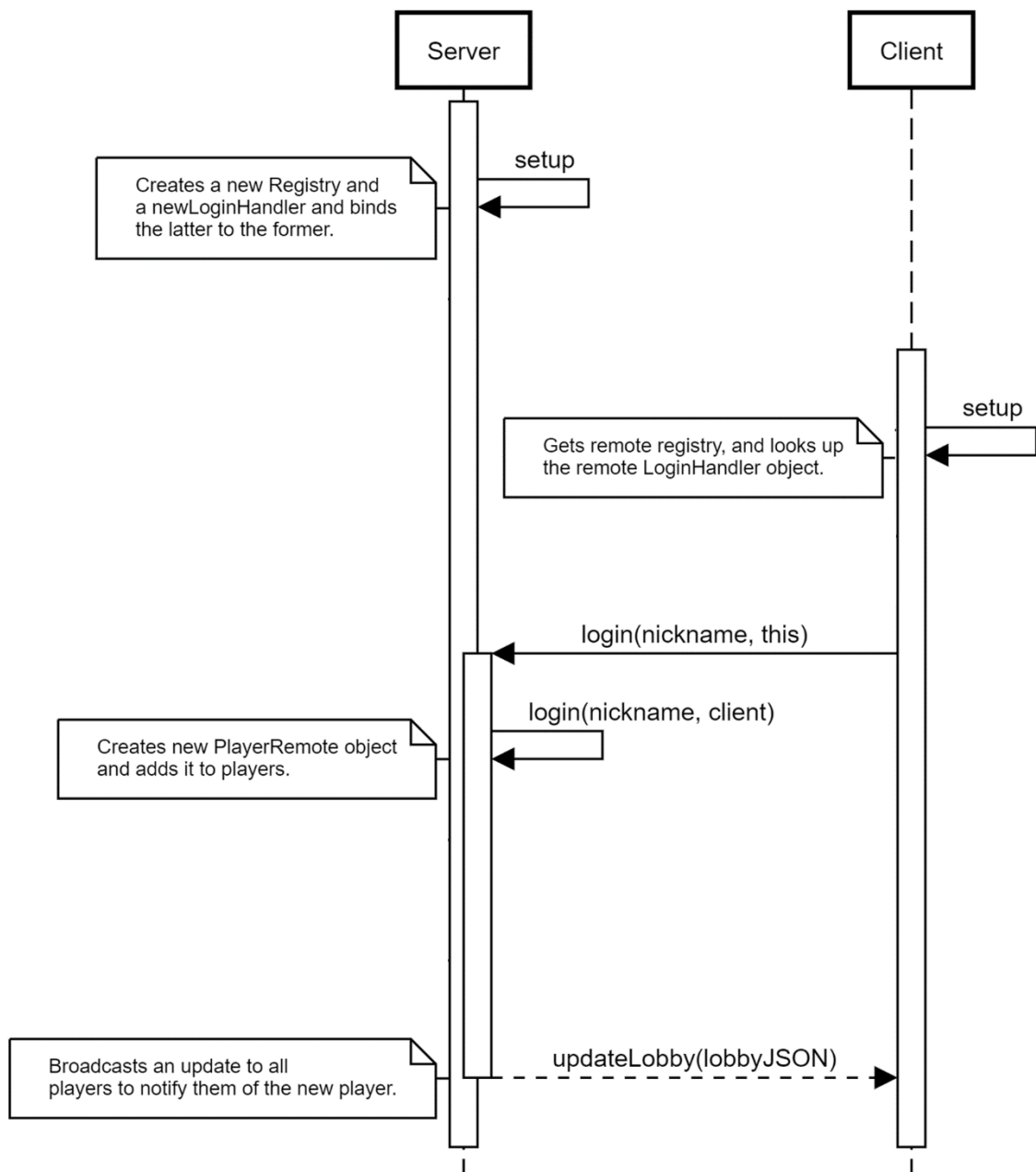
Clients can connect to the Server via both socket and RMI. The two different implementations are masked by the `PlayerConnection` class in the server and by the `ServerConnection` class in the client. The server has a `LoginHandler` object that listens for new clients and creates the right instance of `PlayerConnection` when a new client connects.

The `LoginHandler` allows user to connect to the server and to disconnect from it.

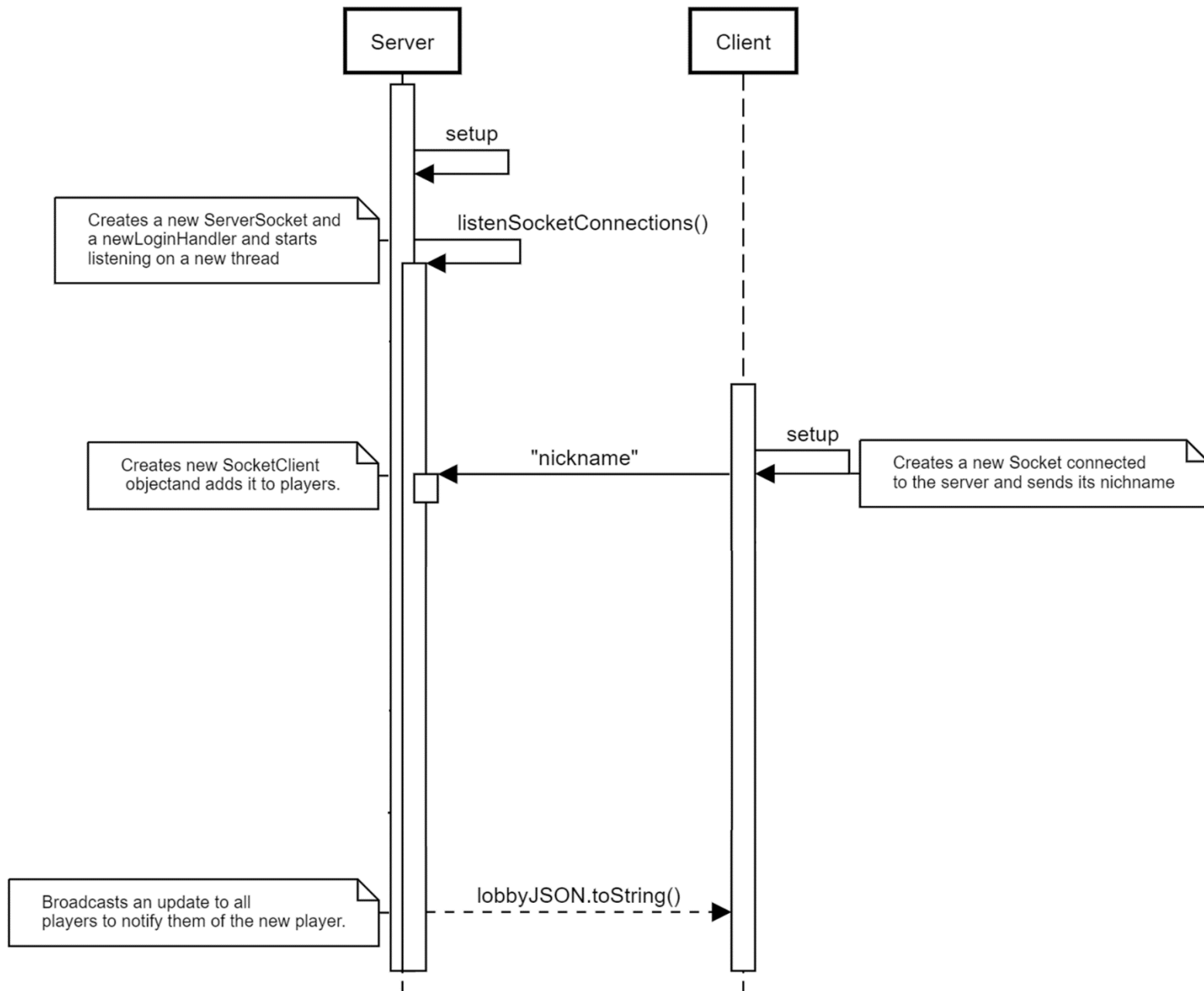
There are two main types of messages that the server can send to the client: selections and updates. The former is used when the controller needs the player to select something, whereas the latter is used when the controller needs to notify clients of an update.

The `ClientPlayer` class implements the `ClientFunctionalities` interface that extends `Remote`. In this way in case of an RMI connection the client passes a reference to self in the login method, allowing the server to call methods from the remote reference. Alternatively, in case of a socket connection the server encodes method execution requests in JSON format and send them to the client. The client parses the messages received from the server and executes the corresponding method of `ClientPlayer` (the same that would be executed directly via RMI). It then encodes the returned value (if present) and sends it to the server. In this way it's guaranteed that the results of a call to a method in `PlayerConnection` are independent of the connection type underneath since the final code executed on the client is the same.

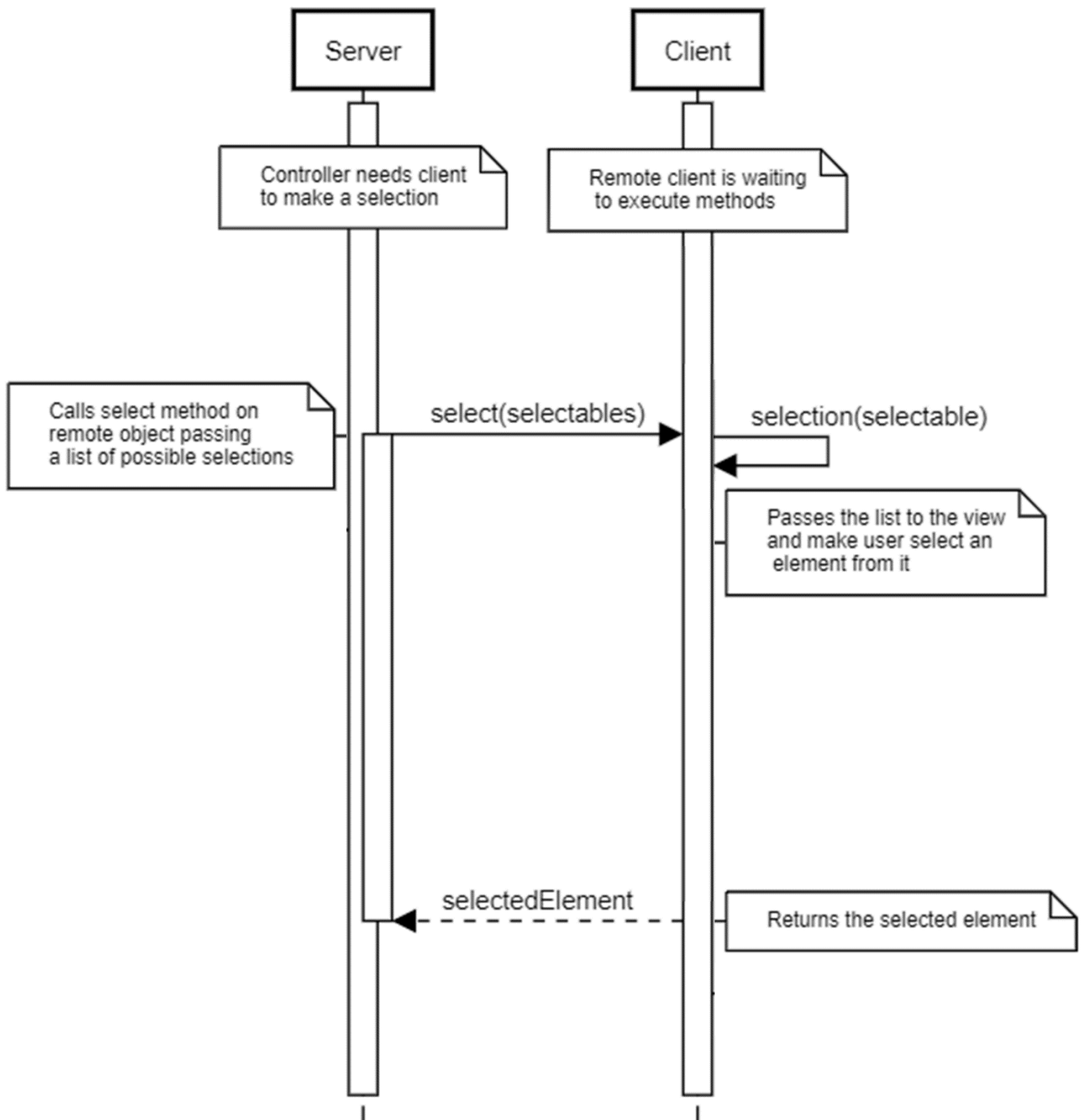
The following is a scheme of the login flow in case of a client connecting via RMI:



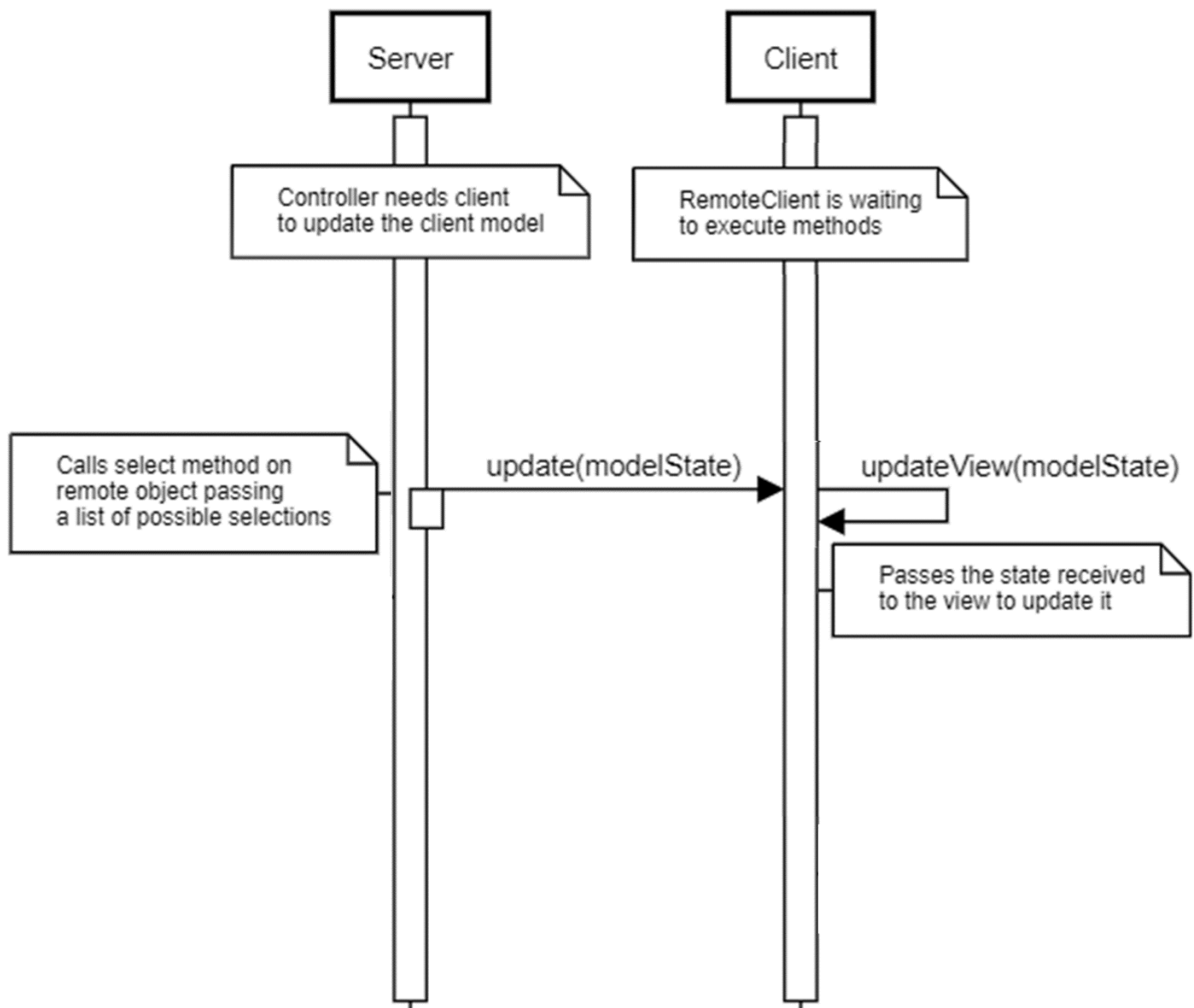
The following is a scheme of the login flow in case of a client connecting via socket:



The following is a scheme of a generic selection method called by the server in case of a client connected via RMI:



The following is a scheme of a generic update method called by the server in case of a client connected via RMI:



The following is a scheme of a generic select/update message sent by the server in case of a client connected via socket:

