Miniclip Challenge: Backend Developer

The challenge consists of implementing a simple **chat server** in Erlang/OTP22.3.

Please use:

- rebar3 (https://rebar3.readme.io/) to create the erlang application and build it (please specify which version of rebar3 you used).
- Kerl (https://github.com/kerl/kerl) to install Erlang/OTP22.3. (suggestion)

Attention! A simple client application is needed (nothing fancy, command line is enough if well explained/documented).

The chat server have some mandatory features:

1. Accept multiple TCP/IP client connections at the same time.
2. The user can create a room or list/enter/exit other created rooms.
3. The user can send a message to all the users in a specific room.

… and some optional features (not in a specific order):

4. The user can send a private message to a single user.
5. The user can invite other users to join him in a private room.
6. The network layer is well separated from the chat server layer

… and if you are really fast and getting bored:

7. The messages and the whole communication between clients and server are mediated using Google Protocol Buffer (https://developers.google.com/protocol-buffers, https://libraries.io/hex/gpb)
8. Store the chat messages on a RedisDB (https://libraries.io/hex/mc_eredis)

Delivery Rules

To implement the challenge you must provide a public github repository with the code and the instructions to build/run the challenge (use the README.md file). If no instructions are provided we assume everything will run smoothly creating a release with the **`rebar3 release`** command.

Don’t forget to tag the final commit as “1.0.0”.
Git usage and code organisation will be evaluated as part of the challenge.

You have two weeks from today.
Good Luck!