

CLIENT-SIDE DESKTOP APPLICATION OF A CHAT ROOM – LLD

Client-side structure:

Namespace - Presentation layer-CLI (command line interface):

❖ **Class GUI:**

- Functionality- handles commands from the user. Initializes by displaying menu bar to the console line (output), and after each "action" (login, sending messages, retrieving messages etc.). Implements action according to the input (numbers menu bar) by calling external functions from the business layer.

Namespace - Business (Logic) layer:

❖ Entity –class User

- Functionality- class per user, all the users identified by a nickname and group id. The user communicates with other users in the chatroom by sending messages.

❖ Entity – class Message

- Functionality- performs actions dictated by a request. A message is limited to 150 characters, identified by GUID and the user who sent it.

❖ Entity -class Chatroom

- Functionality- the chat room manages the interactions with the presentation layer. user and message classes, registers and online users. The chat room holds the URL for server request through communication layer, nicknames list in for duplicates prevention and messages list for efficient message retrieving.

Namespace - Persistent layer:

❖ Class MessageHandler

- Functionality- manages all the data documenting and file handling regarding messages.
- File-Messages- at each message request documents its details, including the message's body.

❖ Class UserHandler

- Functionality- manages all the data documenting and file handling regarding users.
- File-Users- contains all registered users.

❖ Class LogHandler

- Functionality- manages all the operations and errors documenting. In case of errors and exceptions documents the details.
- Files- log- contains all data transitions and system operations.

Namespace - Communication layer:

for communicating with the server.

Requests from server-

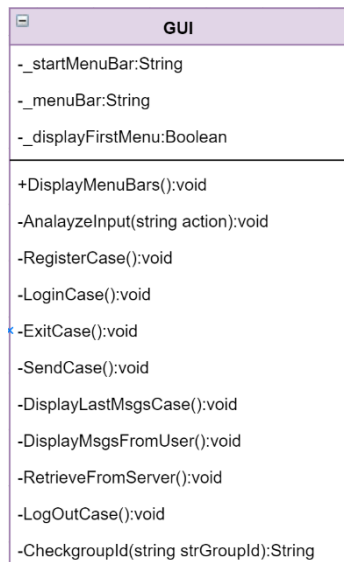
- Send message request

A send message request is initiated by the user, the request is sent to the server, which assigns the message with a unique ID (GUID) and the server's timestamp.

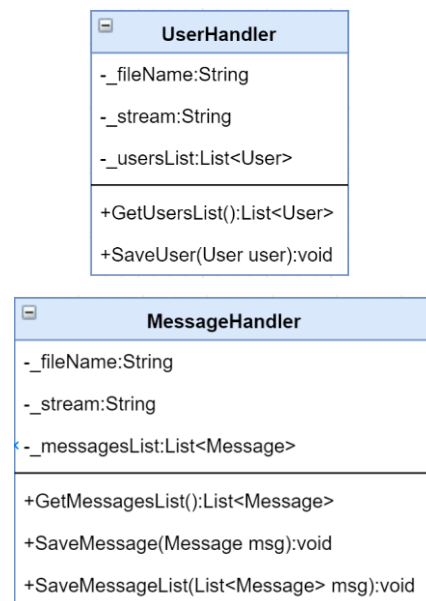
- Get 10 messages request

A get message request is initiated after each "send message request" and potentially can be initiated at any time. This type of request is intended to receive the last 10 messages stored on the server.

Presentation layer



Persistent layer



Logic layer

