High Level Design

Terminology

Chat Room

A virtual environment in which users can post their messages and read the messages
written by other users.

User

A person with a group id and a humorous nickname who interacts with the system.

Registration

The act of recording user details.

Login

The act of signing into the system by the user.

Message

The text which the user delivers. Message content is limited to 150 characters.

Message Frame

A written communication sent between the users of the system. A wrapper for a message.

Communication model

Requests

Send message request

A send message request is initiated by the user, the request is sent to the server, which assinges 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.

Actors

Users

A person connected to the chatroom using a client software, for sending and receiving messages. A user is identified by her group ID and a nickname that is unique to her group.

Milestone 1 Low Level Design

classes

Chat-room
A class of virtual environment in which users can post their messages and read the messages written by other users.
Atribbutes:
(-) logger //will be in charge of logging the data to a txt file
(-) List<User> users
(-) List<Message> messages
(-) User currentUser
(-) const string url
Methods:
(+) Start() - preparing the system to start runing
(+) Registration(int groupId, string nickname)
(+) bool Login(int groupId, string nickname)
(-) bool IsValidNickname(int groupId, string nickname)
(-) bool CheckIfInsertSomething(string str)
(+) Logout()
(+) RetrieveNMessages(int n) - getting N messeges from the server
(+) List<IMessage> DisplayNMessages(int n)
(+) List<IMessage>DisplayAllMessagesFromCertainUser(string groupId, string nickname)
(+) Send(string messageContent)
User
A class of a person who interacts with the system.
Attributes:
(-)string groupId
(-)string nickname
Methods:
(+) Send(string messageContent)

Message
A class which describes an object which includes:
Attributes:
(-)Guid Id   //is determined by the server
(-)string nickname
(-)DateTime Date //is determined by the server
(-)string messageContent //limited to 150 characters
(-)string groupId

<u>methods</u>:
(+) save() //The method is called from the constructor. Every new Message object will be handled by the persistent layer.
(+)CheckValidity(string content)

gui

this class belongs to the presentation layer and will be in charge of all the presentation part to the user and will manage the communication between the user and the logic layer

<u>Attributes</u>:
(-) logger //will be in charge of logging the data to a txt file
(-)ChatRoom chatRoom

<u>methods:</u>
(+) Start() //display the starting information for the client
(+)Register() //display the client the information of the registration process and give the logic layer the details that the client gave the system for handling
(+)Login() //display the client the information of the login process and give the logic layer the details that the client gave the system for handling
(+)AfterLogin() //display the informetion that a loged in user should get
(+) Display(int number) //display a number of meddages
(+)logout()//inform the logic layer that a user asked to log out
(+)send()//inform the logic layer a user tried to send a message and deliver it the message content

MessageHandler

This class belongs to the persistent layer and it will connect between the DataBase and the Logic layer

<u>methods:</u>
(+) saveToFile(Message message)
(+) List <Message> restoreMessages()

UserHandler

This class belongs to the persistent layer and it will connect between the DataBase and the Logic layer

<u>methods:</u>
(+) saveToFile(User user)
(+) List< User> restoreUsers()