# Chat Room LLD

## Terminology

### Chat Room

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

### User

A person who interacts with the system.

### Nickname

A familiar or humorous name the user uses to identify himself.

### 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.Server structure

# Business Logic

## ChatRoom class

### Functionality

Operate the chat room and provides the user with the requested functionalities from the requirements document. This class is static.

### Fields :

**HOME_URL** - holds a static field with the url when running the server locally.
**BGU_URL** - holds a static field with the url when connecting to the server at the university.
_**location** - holds the client's location.
_**loggedinUser** - holds the current user info.

### enums

Place - can get the values of Home or University. helps to easily connect to a server that is running locally. (public)

### Properties

URL - return the server's url according to the client location.
LoggedInUser - get and set the lodged in user.

### Functions

**Start(Place location)** - initialize the ram with the registers users and messages, and set the location of the server according to location.
**exit()** - close the program.
**isLoggedIn()** - return if there is a looted in user
**register(string nickname)** - register with the given nickname and with our's groupID.
**logIn(string nickname)** - login with the given nickname and with our's groupID.

**logout()** - logout the user
**send(string body)** - send a new message with the given body
**SaveLast10FronServer()** - request the last 10 messages from the server
**request20Messages()** - receive the last 20 stored messages
**requestAllMessagesfromUser(string nickname, int GroupID)** - receive all the messages that was sent by the user with the given nickname on the requested group ID
**requestMessages(int number)** - receive the last n stored messages (private method)

# MessageSrvice class

### Functionality

handle the stored on RAM messages. This class is static

### Fields :

_**ramMessages** - hold the messages on the ram.

### Properties

RamMesseges - get and set the stored messages

### Functions

**start()** - initials the stored messaged from disk
**EditMessage(Guid ID, string newBody)** - edit the message with the given guid to the new content
**AllMessagesfromUser(IUser user)** - receive all the messages that was sent by the given  user.
**lastNMessages(int amount)** - receive the last n stored messages
**SaveLast10FronServer()** - request the last 10 messages from the server
**sort(ArrayList messages)** - sort the given messages by their date (private method)
**UpdateDisk()** - updates the disk after a chance was made on the ram (private method)
**SetRAM()** - draw messages from disk into ram (private method)

# UserSrvice class

## Functionality

handle the stored on RAM users. This class is static

## Fields :

**_ramUsers**- hold the users on the ram.

## Properties

RamUsers - get and set the stored users

## Functions

**start()** - initials the stored users from disk
**register(IUser user)** - save the given user to the ram after registration
**CanRegister(IUser user)** - returns if it is possible to register to the given user (not taken already)
**CanLogIn(IUser user)** - returns if it is possible to login to the given user (was registered)
**UpdateDisk()** - updates the disk after a chance was made on the ram (private method)
**SetRAM()** - draw users from disk into ram (private method)

# IUser interface

## Functionality

This is an interface that representing a user

## Properties

NickName - get this user’s nickname
Group_ID - get this user’s nickname

### Functions

**Send(string msg, string url)** - send a new message to the server with the requested content to the given url
**logout()** - logout this user

## User class

### Functionality

Implements IUser interface. represent a user in the chat room

### Fields :

**_groupID** - the group ID of this user.
**_nickName** - the nickname of this user.
**GROUP_ID** – the group ID of our team from the registration sheet.

### Properties

Implements the IUser's properties :
NickName - get this user's nickname
Group_ID - get this user's nickname

### Functions

**Send(string msg, string url)** - send a new message to the server with the requested content to the given url
**logout()** - logout this user
**Equals(object obj)** - return if two users are equals (same group ID and nickname) - override object's Equals method
**ToString()** - return a string that represent this user - override object's ToString method

# Message class

## Functionality

Implements IMessage interface. represent a message in the chatroom

## Fields :

_**guid** - the unique identifier of this message, the guid.
_**recivingTime** - the time this message was received by the server
_**sender** - the user that sent this message
_**body** - the content of this message

## Properties

Implements the properties of IMessage :

Date - get the receiving time

GroupID - get the group ID of the sender

Id - get the guid go this message

MessageContent - get and set (private setter) the contest of this message

UserName - get the sender’s user name

Sender - get the sender of this message (private, not from IMessage)

## Functions

**editbody(string newBody)** - edit the content of this message
**isValid(string body)** -return if the body is valid. A static method.
**Equals(object obj)** - return if two messages are equals (same guid) - override object’s Equals method
**ToString()** - return a string that represent this message - override object’s ToString method

# MessageComparatorByDate class

## Functionality

Implements IComparer interface. compares to messages according to their receiving by server time.

## Functions

**Compare(object x, object y)** - compare two messages x, y. return a positive number if x>y (x was send later), negative number id x<y (x was sent earlier), and 0 if both was sent at the same time

# MergeTwoArrays class

## Functionality

This class merges two ArrayLists into the first one,  and avoids duplications. This class is static.

## Functions

**mergeIntoFirst(ArrayList array1, ArrayList array2)** - merge the two ArrayLists into array1,  and avoids duplications

# Persistent Layer

## SerializationService class

### Functionality

Enable to serialize and deserialize any object to any file. Static class.

### Functions

**serialize(object toSerialize, string fileName)** - serialize the given object to fileName
**deserialize(string fileName)** - deserialize an object from fileName

## UserSerializationService class

### Functionality

Enable to serialize and deserialize the users. This class is static.

### Properties

**USERS_LIST** - hold the location to serialize and deserialize users from

### Functions

**serialize(ArrayList users)** - serialized the users
**deserialize()** -desterilize the uses

## MessageSerializationService class

### Functionality

Enable to serialize and deserialize the messages. This class is static.

### Properties

**Messages_LIST** - hold the location to serialize and deserialize messages from

### Functions

**serialize(ArrayList mesages)** - serialize the  messages
**deserialize()** - deserialize the messages

# Logging Layer

## Logger class

### Functionality

Operate the lodger. This class is static.

### Fields :

log - holds a static field with the logger

### Properties

Log - get the logger

### Functions

**Developer(string logMessage)** - can be used to put a reference at the begetting of the message - “Developer : “ to inform that this logged message is for the developer
**Maintenance(string logMessage)** - can be used to put a reference at the begetting of the message - “Maintenance : “ to inform that this logged message is for the maintenance
**MethodStart(MethodBase method)** - can be used to put a state when a method was entered for debugging
**MethodStart(string methodName, string className)** - can be used to put a state when a method was entered for debugging

# Presentation Layer

## CLI class

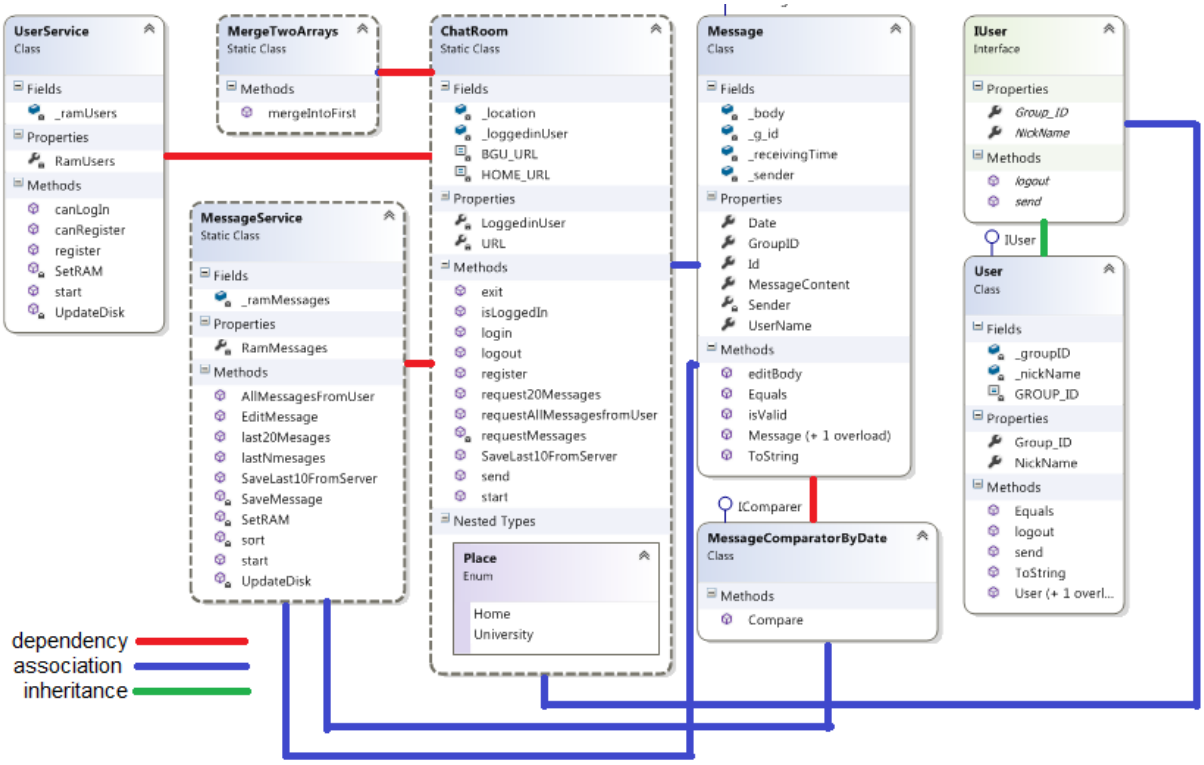
### Functionality

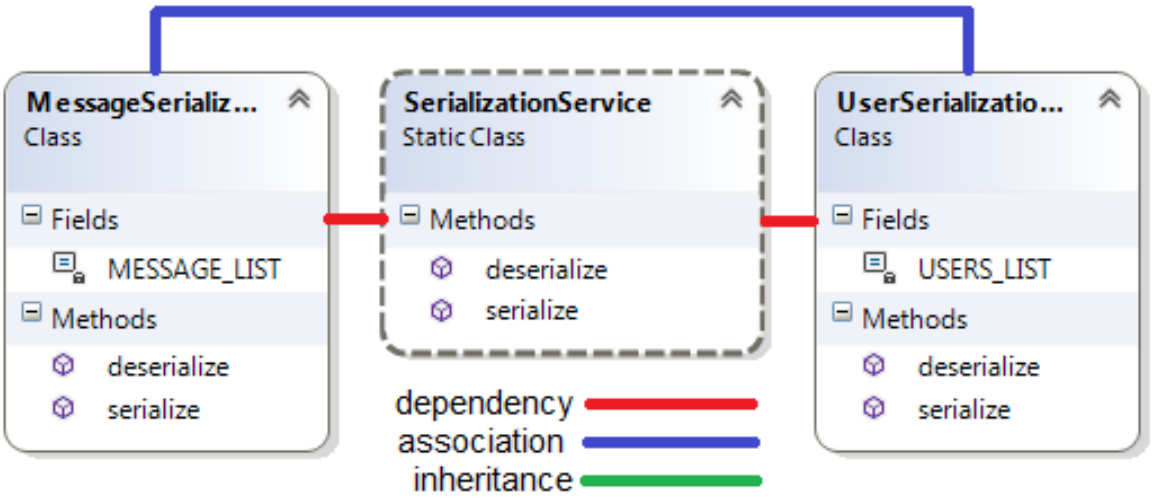
Operate the common-line interface. This class use the singleton design

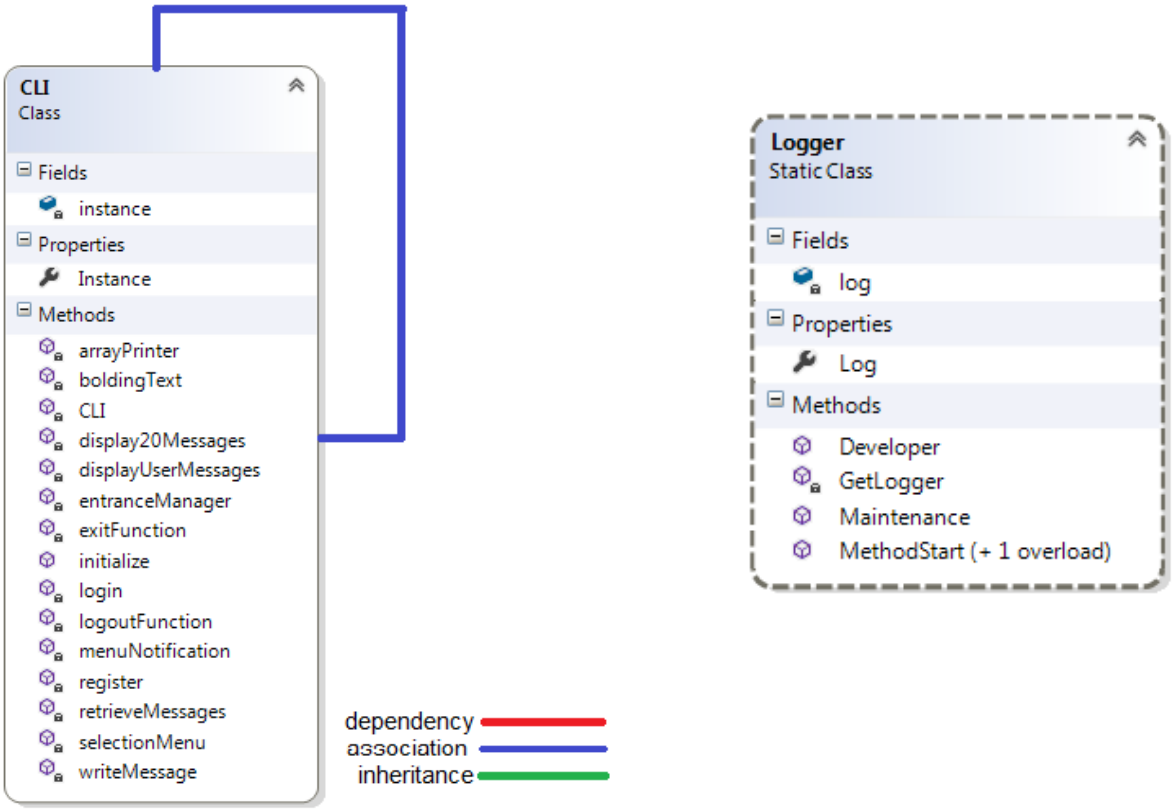
### Functions

**CLI()**- implements the CLI as a singleton.
**initialize()**-initialize(): used in order to initialize the CLI. introducing the user to the chat room instructions and allows him to interact with the relevant menu.
**entranceManager()**-This menu handles a client which isn't logged-in in the chat room.
**selectionMenu()**-This menu handles a client that has logged-in.
**retrieveMessages()-**Trying to retrieve last 10 messages from server, responds accordingly if the attempt was successful or not.
**display20Messages()**-Trying to display last 20 messages from server, responds accordingly if the attempt was successful or not.
**displayUserMessages()**-Trying to display all retrieved messages of a certain user, responds accordingly if the attempt was successful or not.
**writeMessage()-**This function allows a user to send a new message, only if under 150 chars.
**logoutFunction()**-Handles login-out.
**menuNotification()**-A message that pops-up when a user is pressing irrelevant keys (Instructions for menus).
**boldingText(string text, ConsoleColor color)**- Displays the text in the requested color.
**login()**-Handles login-in.
**register()**-Handles registration.
**exitFunction()**-Handles exit request.
**arrayPrinter(ArrayList array)**- An easy way to print the relevant array received from the ChatRoom class.

# UMLs


UserService
Class
Fields
_ramUsers
Properties
RamUsers
Methods
canLogIn
canRegister
register
SetRAM
start
UpdateDisk
MergeTwoArrays
Static Class
Methods
mergeIntoFirst
ChatRoom
Static Class
Fields
_location
_loggedinUser
BGU_URL
HOME_URL
Properties
LoggedinUser
URL
Methods
exit
isLoggedIn
login
logout
register
request20Messages
requestAllMessagesfromUser
requestMessages
SaveLast10FromServer
send
start
Nested Types
Place
Enum
Home
University
Message
Class
Fields
_body
_g_id
_receivingTime
_sender
Properties
Date
GroupID
Id
MessageContent
Sender
UserName
Methods
editBody
Equals
isValid
Message (+ 1 overload)
ToString
IUser
Interface
Properties
Group_ID
NickName
Methods
logout
send
IUser
User
Class
Fields
_groupID
_nickName
GROUP_ID
Properties
Group_ID
NickName
Methods
Equals
logout
send
ToString
User (+ 1 overl...
MessageService
Static Class
Fields
_ramMessages
Properties
RamMessages
Methods
AllMessagesFromUser
EditMessage
last20Mesages
lastNmesages
SaveLast10FromServer
SaveMessage
SetRAM
sort
start
UpdateDisk
IComparer
MessageComparatorByDate
Class
Methods
Compare
dependency
association
inheritance


MessageSerializ...
Class
Fields
MESSAGE_LIST
Methods
deserialize
serialize
SerializationService
Static Class
Methods
deserialize
serialize
UserSerializatio...
Class
Fields
USERS_LIST
Methods
deserialize
serialize
dependency
association
inheritance


CLI
Class
Fields
instance
Properties
Instance
Methods
arrayPrinter
boldingText
CLI
display20Messages
displayUserMessages
entranceManager
exitFunction
initialize
login
logoutFunction
menuNotification
register
retrieveMessages
selectionMenu
writeMessage
Logger
Static Class
Fields
log
Properties
Log
Methods
Developer
GetLogger
Maintenance
MethodStart (+ 1 overload)
dependency
association
inheritance