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

sort(ICollection<IMessage> messages, MessageService.Sort SortBy, bool descending)- Sort a message List by the time

requestAllMessagesfromUser(string nickName, int GroupID)- Receive all the messages from a certain user

requestMessagesfromUser(ICollection<IMessage> messages, string nickName, int GroupID)- Receive all the messages from a certain user from a certain collection

requestAllMessagesfromGroup(int GroupID)- Receive all the messages from a certain group

requestMessagesfromGroup(ICollection<IMessage> messages, int GroupID)- Receive all the messages from a certain group from a certain collection
requestMessages(int number) - receive the last n stored messages (private method)

MessageSrvice class

enums

sort- Sort options

Functionality

handle the stored on RAM messages. This class is static

Properties

RamMesseges - get and set the stored messages

Functions

FilterByUser(IUser user)- recive all the messages from a certain user
FilterByGroup(int groupID)- recive all the messages from a certain group

FilterByUser(IUser user, ICollection<IMessage> toFilter)- recive all the messages from a certain user and from a certain colection of messages
FilterByGroup(int groupID, ICollection<IMessage> toFilter)- recive all the messages from a certain group and from a certain colection of messages
sort(ICollection<IMessage> messages, Sort SortBy, bool descending)- Sort a message List by the time
deserialize()-deserializes data.
serialize(ICollection<IMessage> Data)-serializes data
DefaultSort(ICollection<IMessage> Data)- Sorting by time
SaveMessage(IMessage msg)- save a single message to the RAM
EditMessage(Guid ID, string newBody) - edit the message with the given guid to the new content
lastNMessages(int amount) - receive the last n stored messages
SaveLast10FromServer(string url) - request the last 10 messages from the server

UserSrvice class

Functionality

handle the stored on RAM users. This class is static

Properties

RamUsers - get and set the stored users

Functions

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)
deserialize()-erializes data.
serialize(ICollection<IUser> Data)- serializes data.
DefaultSort(ICollection<IUser> Data)-returns the massege list as is

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

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

GeneralHandler class

Fields :
_ramData- Stores a coppy of the data in the ram for a quick acces
Functions:
Start()- initiates the ram's saves from users stored in the disk.
Deserialize()-desirializes data.
Serialize(ICollection<T> Data)-serializes data
DefaultSort(ICollection<T> Data)-sorts the data
UpdateDisk()- updatesthe data stored in the disk after changings in the ram
SetRam()- setting the ram, if null.
MergeIntoFirst(ICollection<T> list2)- merge two collections to the first one without 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
Server(string logMessage)- can be used to put a referece at the begennig of the message – Server
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

OvesrvanleObject class

Properties:

Messages-holdes the messages
messageContent- holds the message content
errorText- holds the exeption error text
mainWindowLoginRadio-is it a login or register
username- holds the username of the logged in user
groupID- holds the groupid of the logged in user
usernameBox- a text box that holds the username filter text
groupidBox- a text box that holds the groupid filter text
filterUsername- Boolean that holdes if the filterusername is checked
filterGroupid- Boolean that holdes if the filtergroupid is checked
filterNone- true if none of the filters are used
filterGroupString-holds the group name the messages sorted by
filterNameString- holds the name the messages sorted by
sortAscending-true if the sort type is ascending
sortDescending- true if the sort type is descending
sortOption- holds the number of the sort type

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.
Printer<T>(ICollection<T> list)- An easy way to print the relevant lists received from tha ChatRoom class

Chatwindow.xaml

Fields :

ObservableObject bindObject- Responsible for all the binding in the project

DispatcherTimer dispatcherTimer- a timer that handeles refreshing the chat
ICollection<IMessage> last20- holdes the last 20 messeges
bool sortChanged-holdes a boolean true if the sort method changed
bool filterApplied- holdes a boolean true if a filter is applied

Functions

dispatcherTimer_Tick(object sender, EventArgs e)-updates the massegas showing on the screen every 2 seconds
sendButton_Click(object sender, RoutedEventArgs e)-sends the message typed in the textbox
logoutButton_Click(object sender, RoutedEventArgs e)- logges out
Printer(ICollection<IMessage> list)-printes a message on the screen
UpdateScreen()-updates the screen
RadioButton_Click(object sender, RoutedEventArgs e)
ComboBox_SelectionChanged(object sender, SelectionChangedEventArgs e)-takes care of the sort type to apply
Filter_Click(object sender, RoutedEventArgs e)- takes care to clicks on the filter button

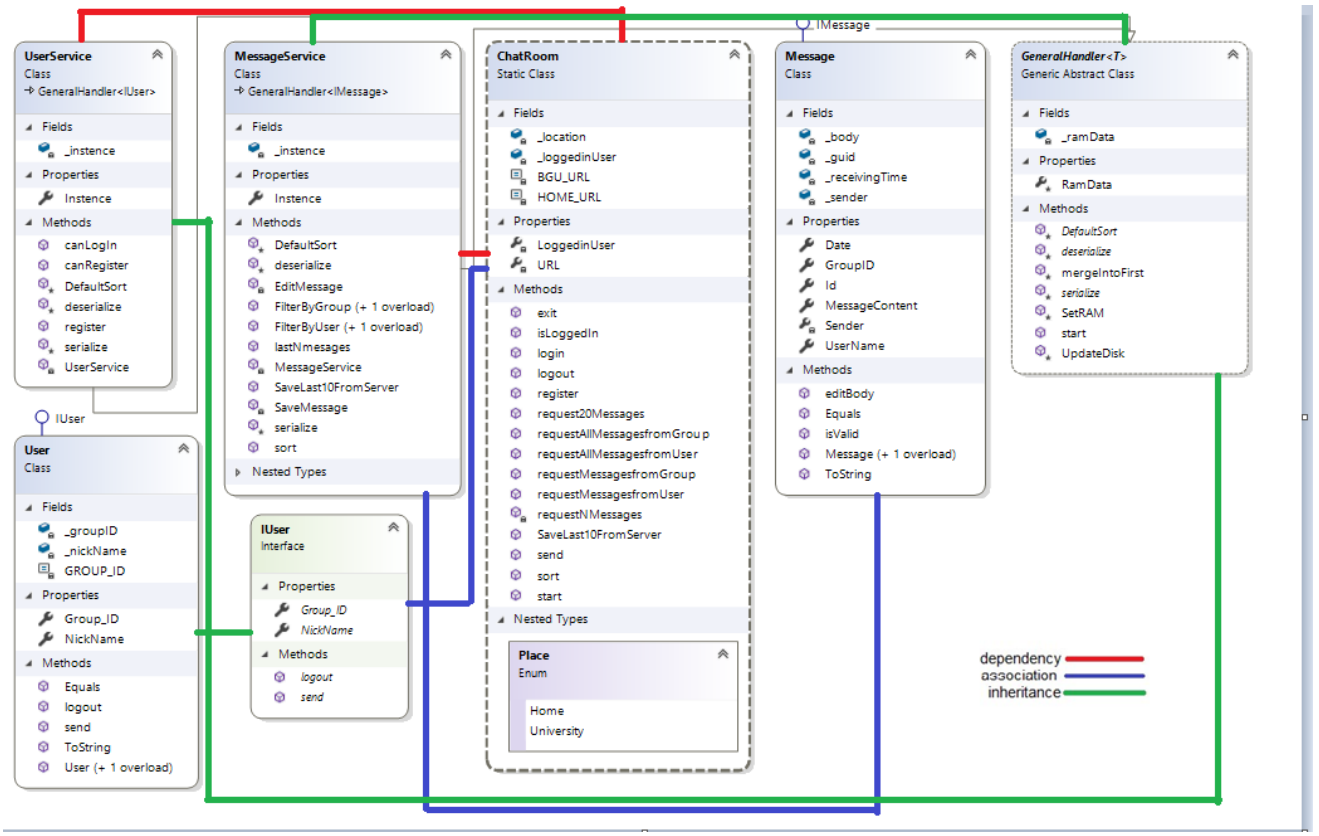
Apply_Click(object sender, RoutedEventArgs e)- takes care to clicks on the apply button

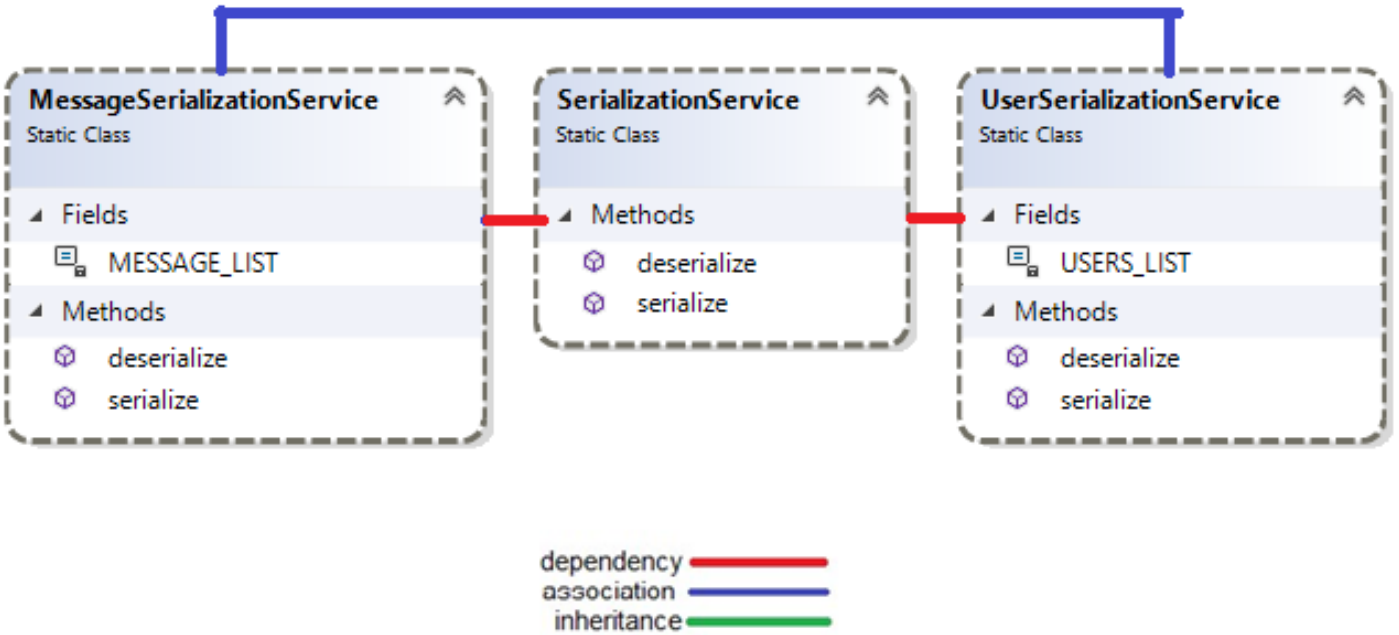
Mainwindow.xaml

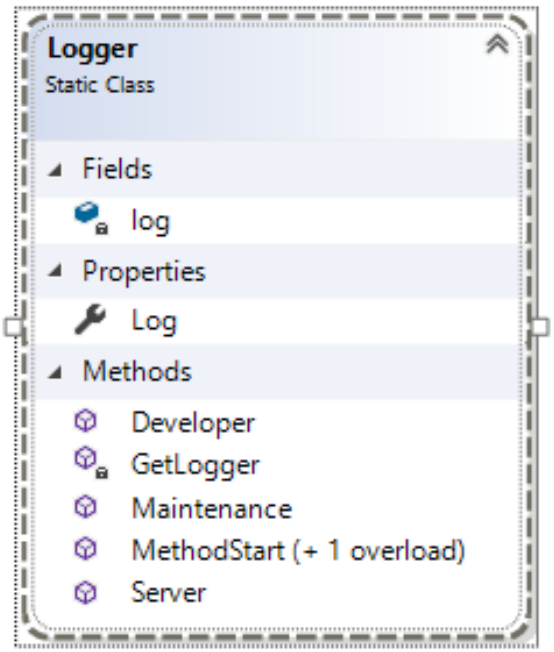
Functions:

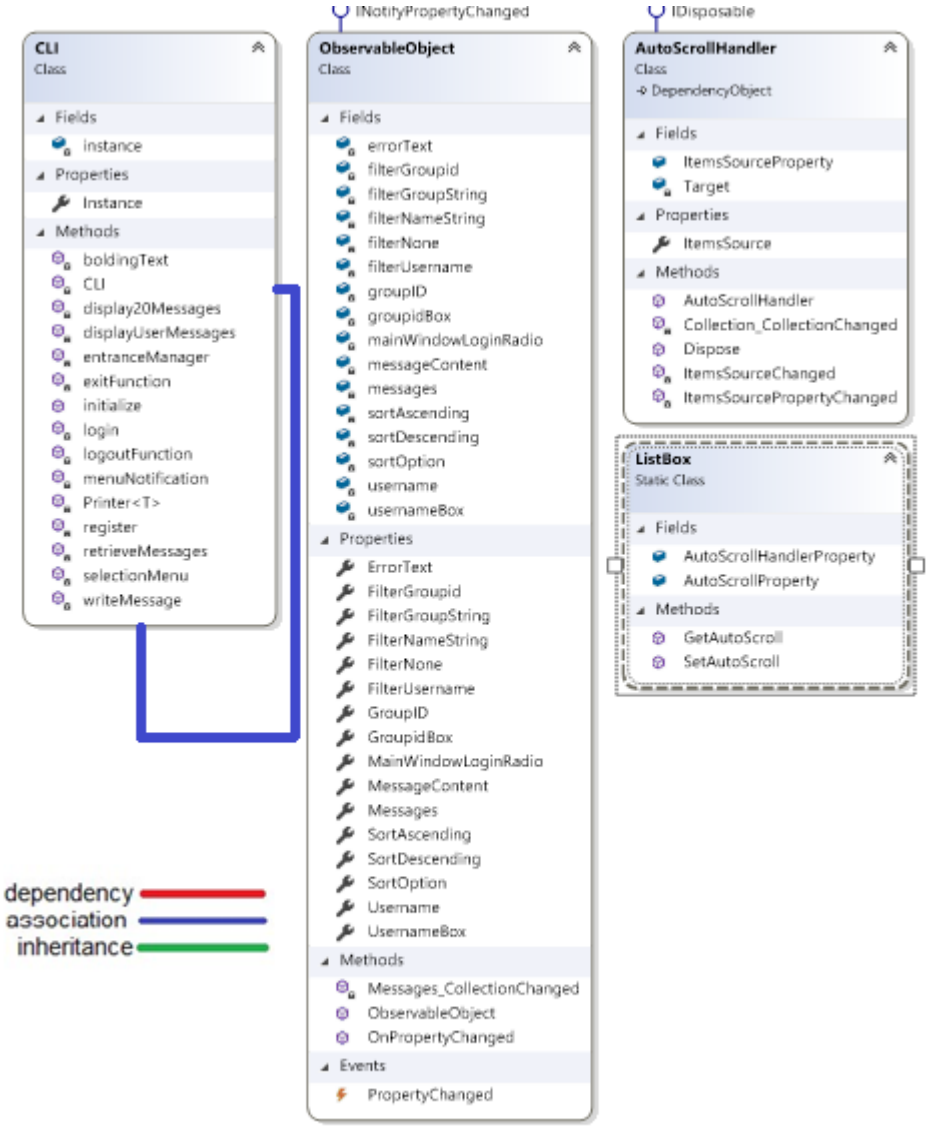
userHandlerButton_Click(object sender, RoutedEventArgs e)- takes care of logging in or registering a user

exitButton_Click(object sender, RoutedEventArgs e)- exsits the window


UserService
Class
GeneralHandler<IUser>
Fields
_instence
Properties
Instence
Methods
canLogIn
canRegister
DefaultSort
deserialize
register
serialize
UserService
IUser
User
Class
Fields
_groupID
_nickName
GROUP_ID
Properties
Group_ID
NickName
Methods
Equals
logout
send
ToString
User (+ 1 overload)
MessageService
Class
GeneralHandler<IMessage>
Fields
_instence
Properties
Instence
Methods
DefaultSort
deserialize
EditMessage
FilterByGroup (+ 1 overload)
FilterByUser (+ 1 overload)
lastNmesages
MessageService
SaveLast10FromServer
SaveMessage
serialize
sort
Nested Types
IUser
Interface
Properties
Group_ID
NickName
Methods
logout
send
ChatRoom
Static Class
Fields
_location
_loggedinUser
BGU_URL
HOME_URL
Properties
LoggedinUser
URL
Methods
exit
isLoggedIn
login
logout
register
request20Messages
requestAllMessagesfromGroup
requestAllMessagesfromUser
requestMessagesfromGroup
requestMessagesfromUser
requestNMessages
SaveLast10FromServer
send
sort
start
Nested Types
Place
Enum
Home
University
IMessage
Message
Class
Fields
_body
_guid
_receivingTime
_sender
Properties
Date
GroupID
Id
MessageContent
Sender
UserName
Methods
editBody
Equals
isValid
Message (+ 1 overload)
ToString
GeneralHandler<T>
Generic Abstract Class
Fields
_ramData
Properties
RamData
Methods
DefaultSort
deserialize
mergeIntoFirst
serialize
SetRAM
start
UpdateDisk
dependency
association
inheritance


MessageSerializationService
Static Class
Fields
MESSAGE_LIST
Methods
deserialize
serialize
SerializationService
Static Class
Methods
deserialize
serialize
UserSerializationService
Static Class
Fields
USERS_LIST
Methods
deserialize
serialize
dependency
association
inheritance


Logger
Static Class
Fields
log
Properties
Log
Methods
Developer
GetLogger
Maintenance
MethodStart (+ 1 overload)
Server


CLI
Class
Fields
instance
Properties
Instance
Methods
boldingText
CLI
display20Messages
displayUserMessages
entranceManager
exitFunction
initialize
login
logoutFunction
menuNotification
Printer<T>
register
retrieveMessages
selectionMenu
writeMessage
INotifyPropertyChanged
ObservableObject
Class
Fields
errorText
filterGroupid
filterGroupString
filterNameString
filterNone
filterUsername
groupID
groupidBox
mainWindowLoginRadio
messageContent
messages
sortAscending
sortDescending
sortOption
username
usernameBox
Properties
ErrorText
FilterGroupid
FilterGroupString
FilterNameString
FilterNone
FilterUsername
GroupID
GroupidBox
MainWindowLoginRadio
MessageContent
Messages
SortAscending
SortDescending
SortOption
Username
UsernameBox
Methods
Messages_CollectionChanged
ObservableObject
OnPropertyChanged
Events
PropertyChanged
IDisposable
AutoScrollHandler
Class
DependencyObject
Fields
ItemsSourceProperty
Target
Properties
ItemsSource
Methods
AutoScrollHandler
Collection_CollectionChanged
Dispose
ItemsSourceChanged
ItemsSourcePropertyChanged
ListBox
Static Class
Fields
AutoScrollHandlerProperty
AutoScrollProperty
Methods
GetAutoScroll
SetAutoScroll
dependency
association
inheritance