אבישי יניב: 307916023
רון עסיס: 312178825
רון אוחנה: 203314604

Milestone 2-Low Level Design

Brief information on the project:

Creating a virtual environment in which users can post their messages and read the messages written by other users connected to the same server.

ISE main project:

Classes:

User Class

General functionality and brief info:

The class “User” represents an actual person registered to the system or using the chat.
The “user” class has the purpose of setting each person who uses the chat with specific identification properties.

Fields:

log – an object necessary for logging system using log4net.

g_ID – this is the group ID of each user.

Nickname – unique name for the user as an identification mean.

Status – the current status of the user (Online or Offline).

Functionality:

sendMessage – Allows the user to send messages to the chat room.
The method receives a String (content of the message) and returns a full message object containing all the relevant details including the content itself.

logout – logging out of the system.

isOnline - This method is the user’s status in the system at any time.
true – Online.
false – Offline.

Message Class

General functionality and brief info:

The Massage class is a way to represent each massage that goes through the server and to the chat room. Each massage has unique details such as time, date, nickname, guide retrieved from server etc.

Fields:

log – an object necessary for logging system using log4net.

timeStamp – time and date in form of String.

guidNum – The ID number of a message given by the server.

messageBody – A string with the message’s content.

Writer_name – String holding the name of the user who sent the message.

Writer_gID – String holding the group id of the user who sent the message.

אבישי יניב: 307916023
רון עסיס: 312178825
רון אוחנה: 203314604

Milestone 2-Low Level Design

Functionality:

checkContentValidity – a method responsible for checking if a message is valid.
it checks if length is greater than 100 and or the message body is empty.

ChatRoom Class

General functionality and brief info:

ChatRoom class is the "Manager" of a single chat room. It manages the messages and users in form of lists and has the status of all current actors at each moment.

Fields:

log – an object necessary for logging system using log4net.

loggedInUser – Represents the user who is currently connected to the chat room.

messages – List of all the sent messages.

users – List of all the registered users.

MessageHandler – a MessageHandler object used to handle the logic of creating queries from the logic layer.

centerOperatoin – a centerOperation object used to handle the logic of sorting messages from the logic layer.

userListChanged , MessagesListChnaged – Boolean vibrables responsible for alerting when users list or messages list has changed.

Functionality:

registration – This function is handling the process of creating new users.
it receives nickname, group ID, password and password verification and if those details are correct (a verification is needed) , new user is being created.

login - Receives a nickname group ID and password as identification properties and if the detailed are correct, the user can enter the chat room.
the user's status changes to online and loggedinUser (in the chat room) changes as well.

send – Through this method, an online user can send message into the chat room.
It receives the message content and reaches to the user's class to send the message from the user itself (OOP).

retrieveMessages – This method start the process of receiving messages from the server.

getUser – Method used to return a specific user verified by group id and nickname.

updateMessages – calls retrieveMessage.

updateUsers – reaches the sql server to get the users. Afterwards it checks if something has changes in users list and report.

getListsReady – prepare a list of already filtered lists by the user's demands.

passwordIsCorrect – reaches the server to check if a password matches the password given.

saveEditedMessage – reaches the server to edit a message which can be searched by the details given.

updateMessageList – after messages have been received, this method checks if messages list has been changed.

shrink – shrinks the messages list to the size of 200 max. shorten the list from the beginning.

hasBeenEdited – receives body and guid and returns message object if the guid exists and the body is different. used to check if a message has been edited.

אבישי יניב: 307916023
רון עסיס: 312178825
רון אוחנה: 203314604

Milestone 2-Low Level Design

MessageHandler Class

General functionality and brief info:

MessageHandler Class is responsible for handling the messages requests to the sql server.

Fields:

filters – a list of IQueryAction.

Functionality:

clearFilters – removes all the filtering parameters.

addGroupFilter – adding group id filter parameter.

addNickkName – adding nickname filter parameter.

retrreiveMessages – adding the relevant filters to the sql query string.

SQL Module Class

General functionality and brief info:

SQL Module class is used to handle connectivity to the sql server, the class instance is a singleton object that classes related to the Business Logic layer can use to execute commands related to reading and writing to server.

Fields:

instance – An instance of the singletone filehandler.

currDir – The directory manager.

Server_address – An address to the server location.

User_name – user name for the connection to the server.

password –password for the connection to the server.

connection– An objects used to access the sql server.

Data_reader – An object used to read from sql queries results.

Functionality:

passwordOfUser – returning the password of a user from sql server.

idOfUser- returning the id of a user from sql server..

saveNewUser – method responsible for saving new users to the sql server.

loadMsgList- used to load the last updated list of messages from the server.

loadUserList- used to load the last updated list of users from the server**.**

saveMessageToServer – method responsible for saving a given message to the sql server.

saveEditedMessageToServer – method responsible for saving an edited message to the server. The method tracks the old message and update it.

אבישי יניב: 307916023
רון עסיס: 312178825
רון אוחנה: 203314604

Milestone 2-Low Level Design

Query Class

General functionality and brief info:

A class representing the query to the server. It holds all the query attributes and can be added to, removed from or be changed.

IQueryAction Interface

General functionality and brief info:

An interface with a single method responsible for the specific action of each child implementing the interface.

GroupFilter Class

General functionality and brief info:

This object is responsible for creating the relevant part of a query for filtering by group id.

Implements IQueryAction therefore has execute method which does the logic of adding the filter to the query.

NicknameFilter Class

General functionality and brief info:

This object is responsible for creating the relevant part of a query for filtering by nickname.

Implements IQueryAction therefore has execute method which does the logic of adding the filter to the query.

CenterOperation abstract Class

General functionality and brief info:

CenterOperation class is an abstract class used to set default methods for all filter and sort classes. Has a main method called doOperation_NameClass.

SortTime Class

General functionality and brief info:

SortTime class is an object used to order the messages in the chat in a specific order. The class extends CenterOperation and therefor has the method doOperation_NameClass.
This method checks if the demand is ascending or descending and accordingly changes the messages list.
The message list is already sorted by time as a default sort and therefor only ascending or descending action is needed.

SortNickname Class

General functionality and brief info:

SortNickname class is an object used to order the messages in the chat in a specific order. The class extends CenterOperation and therefor has the method doOperation_NameClass.
This method uses a comparator created by us and sorting the messages list by the nickname of the writers.

אבישי יניב: 307916023
רון עסיס: 312178825
רון אוחנה: 203314604

Milestone 2-Low Level Design

SortByGroupIdNickname Class

General functionality and brief info:

SortGidNicknameTime class is an object used to order the messages in the chat in a specific order. The class extends CenterOperation and therefor has the method doOperation_NameClass.
This method uses a comparator created by us and sorting the messages list by the gid, nickname of the messages. The comparator has priority when Gid is top priority followed by nickname.

SortGidNicknameTime Class

General functionality and brief info:

SortGidNicknameTime class is an object used to order the messages in the chat in a specific order. The class extends CenterOperation and therefor has the method doOperation_NameClass.
This method uses a comparator created by us and sorting the messages list by the gid, nickname and timestamp of the messages. The comparator has priority when Gid is top priority followed by nickname and least by time.

CompareGroupIdNickname Class

General functionality and brief info:

CompareGroupIdNickname class is comparator used to sort by more than one parameter(group id and nickname). It implements Icomparer and has the compare method.
This method returns an integer the same as default compare method.

CompareGidNicknameTime Class

General functionality and brief info:

CompareGidNicknameTime class is comparator used to sort by more than one parameter(group id, nickname and time). It implements Icomparer and has the compare method.
This method returns an integer the same as default compare method.

CompareNickname Class

General functionality and brief info:

CompareNickname class is comparator used to sort by one parameter(nickanem) It implements Icomparer and has the compare method.
This method returns an integer the same as default compare method.

Hashing Class

General functionality and brief info:

This class is used to convert the user's password into a much more secure string.
it uses hash method to do so.

אבישי יניב: 307916023
רון עסיס: 312178825
רון אוחנה: 203314604

Milestone 2-Low Level Design

GUI project:

Main_Window Class

General functionality and brief info:

Main_Window class represents the Main window of the program. This is a welcome window for the user which can from there choose to register, login or exit the program.

Main_Window code behind

Each button (login, register, back and exit) has a method that responsible for the action when clicked.
each button that reaches a system property is binded to this property using observable model (explanation later in document).

Registration_Window Class

General functionality and brief info:

Registration_Window class represents the registration window of the program, the user is able to choose a nickname and group id and if those are unknown to the system, the user will be registered, otherwise the user will get an error.

Login_Window Class

General functionality and brief info:

Login_Window class represents the login window of the program, the user is able to enter a nickname and group id and if those are known to the system, chat room window will appear, otherwise the user will get an error massage.

Registration_Window and Login_Window code behind

Each button has a method that responsible for the action when clicked.
each button that reaches a system property is binded to this property using observable model.
In addition to the button clicked methods, there are focus method for instance if user is pressing the choose nickname bar, it will immediately clear.

ChatRoom_Window Class

General functionality and brief info:

ChatRoom_Window class represents the chat window itself, the user can send and watch messages. In addition this window has a menu bar at its right side with filter and sort preferences for the user to enjoy.

ChatRoom_Window code behind

In addition to the window properties, this window holds the timer and initiating process of receiving and displaying messages every specified time.

Functionality:

playDispacherTimer - this method starts the timer, every 2 seconds (as demanded), the method update the timer.

dispacherTime_Tick – every two seconds, this method is doing this process:
- getting new messages from server.
-updating messages list corresponded to the user's choices.
-adding warnings to log.

אבישי יניב: 307916023
רון עסיס: 312178825
רון אוחנה: 203314604

Milestone 2-Low Level Design

ListBox Class

Functionality:

Override handmade property for c# listbox, the addition of this class is the ability to scroll. (shown in practical session.).

editMessageWindow Class

General functionality and brief info:

editMessageWindow class represents the edit message when clicking a valid message.

ChatRoom_Window code behind

Reaching chatroom in business logic in order to update the message in sql server.

Observers:

General functionality and brief info:

Observable model classes are the connection between properties to the gui. In this way there is no direct access to the system properties and by binding a property to its property in the observable model class, one can see the changes in the gui department without reaching the xaml code itself.

Each observable class implements InotifyPropertyChanged which force it to handle changes in the xaml, in that way when a property changes, it will “know”.

ObservableModel and ObserChatRoomModel :

Responsible for login, registration, chatroom and main window.
each property such as textbox observable collection etc, has a property in here.
The property has getter and setter and most important, onPropertyChange that updated when something has changed.
For example, textbox “Choose nickname” in xaml code has a property that returns and receives string object and on property change to know when the user has changed the text box.