

Milestone 2

Low Level Design

Presentation

MainWindow : Window

- **Attributes:**
 - **ObservableObject** _main – the binding object to the window's xaml components.
 - **Chatroom** myChatRoom – Stores the chatroom object in use.
 - **ProgramWindow** pw – Stores the main chatroom window.
- **Methods:**
 - **void** copyResources() – a function for showing the images in the windows.
 - **void** DataWindow_Closing(object, CancelEventArgs) – Determines what happens when closing the window.
 - **void** btn_login_Click(object, CancelEventArgs) – A function for the login button.
 - **void** btn_register_Click(object, CancelEventArgs) – A function for the register button.
 - **void** StartProgram() – Starts the main chatroom window.

ProgramWindow : Window

- **Attributes:**
 - **ObservableObject** _main – the binding object to the window's xaml components.
 - **Chatroom** myChatRoom – Stores the chatroom object in use.
 - **MainWindow** main – Stores the chatroom's login window.
 - **DispatcherTimer** dispatcherTimer – Stores the chatroom's timer.
- **Methods:**
 - **void** startWindow() – starts the timer and updates the window's view.
 - **void** UpdateView() – Updates the window's view.
 - **void** btn_logout_Click(object, CancelEventArgs) – A function for the logout button.
 - **void** ComboBox_SelectionChanged(object, CancelEventArgs) – the function triggered when the filter combo box is changed, and updates the editable fields through the binding.
 - **void** dispatcherTimer_Tick(object, CancelEventArgs) – triggered by the timer every 2 second, retrieves new messages and update the display if necessary.

- **void** Button_Click_1(object, CancelEventArgs) – A function for the apply sort and filter button, sends the selection to the chatroom.
- **void** ProgramWindow_Closing(object, CancelEventArgs) – Determines what happens when closing the window, logout and close the chatroom.

ObservableObject : INotifyPropertyChanged

- **Attributes:**
 - **PropertyChangedEventHandler** PropertyChanged – Required event for this binding class.
 - **ObservableCollection<string>** Messages – binding for the messages panel.
 - **string** nicknameR – Binding for the register nickname textbox.
 - **string** nicknameL – Binding for the login nickname textbox.
 - **string** sortCombo – Binding for the sort combo box.
 - **string** isDesc – Binding for the descending checkbox.
 - **string** filterCombo – Binding for the filter combo box.
 - **string** isFilterGroup – Binding for the filter group text box availability.
 - **string** filterGroup – Binding for the filter group text box.
 - **string** isFilterUser – Binding for the filter user text box availability.
 - **string** filterUser – Binding for the filter user text box.
 - **string** messageText – Binding for the send text text box.
- **Methods:**
 - **void** Messages_CollectionChanged(object, NotifyCollectionChangedEventArgs) - Required method for this binding class.
 - **void** OnPropertyChanged([CallerMemberName] string) – Required method for this binding class.

Logic

ChatRoom

- **Attributes:**
 - **int** sortType – Stores what sort type the user chose.
 - **boolean** isAsc – Stores what order the user has asked for.
 - **int** filterType – Stores what filter type the user chose.
 - **string** userFilter – Stores what user to filter by.
 - **string** groupFilter – Stores what group to filter by.
 - **user** _loggedInUser – Stores the data of the user currently used in or NULL otherwise.
 - **Dictionary<Guid, Message>** recivedMessages – Stores all previously received messages.
 - **Dictionary<String, User>** registeredUsers – Stores all registered users.
 - **String** URL – Stores the URL for the server.
 - **messagesHandler** messHandler – Stores the Masseges Handler used to save and load messages on disk.
 - **usersHandler** userHandler – Stores the Users Handler used to save and load users on disk.
 - **Logger** mLogger – Logger object used by the logger.
 - **FileLogger** mFileLogger – FileLogger object used by the logger.
- **Methods:**
 - **boolean** Login(**string** nickname) – gets a user nickname and loges him in to the system, return false if user non-exists.
 - **boolean** Logout() – loges the user out, returns false if no user is logged in.
 - **boolean** register(**string** nickname) – Creates a user.
 - **int** retrive10Messages() – Returns last 10 messages.
 - **void** SetFilterAndSort(int,int,bool,string,string) – Sets the sort and filter properties.
 - **List<String>** GetAllMessages() – Returns all the messages to be shown at the messages panel in the right order and filter.
 - **List<Message>** GetMessagesByAll() – Returns all messages.
 - **List<Message>** GetAllByUser() – Returns all messages by the userFilter.
 - **List<Message>** GetAllByGroup() – Returns all messages by the groupFilter.
 - **List<String>** SortByTimestamp(List<Message>) – Sorts the given messages by the timestamp.
 - **List<String>** SortByNickname(List<Message>) – Sorts the given messages by the nickname.
 - **List<String>** SortByAll(List<Message>) – Sorts the given messages by the timestamp, group and nickname.

- **boolean** writeMessages(**String** message) – sends message, returns true if successful.
- **Boolean** CheckMessageValidity(**String** content) – gets a new message content and return whether or not its valid before sending it.
- **void** exit() – terminates logger.
- **void** ProcessLogMessage(**String** message) – useless function needed to implement the logger.

User

- **Attributes:**
 - **string** nickname – Stores the unique nickname for the user.
 - **int** groupID – Stores the group ID of the user.
- **Methods:**
 - **Imessage** writeMessage(**string** message, **string** url) – asks communication to send a new message to the server.
 - **List<Imessage>** retrieve10Messages(**string** url) – asks communication to retrieve the 10 latest messages from the server.

Message

- **Attributes:**
 - **Guid** _id – Stores the unique id assigned by the server.
 - **string** _messageContent – Stores the content of the message.
 - **string** _userName – Stores the nickname of the user that as written the message.
 - **DateTime** _date – Stores the time when the message was received by the server.
 - **string** _groupID – Stores the id of the user that wrote the message.
- **Methods:**
 - **String** ToString() – returns the messages representation as string.

Persistence

Abstract fileHandler

- **Attributes:**
 - **string** binPath – Stores the path to the bin file.
- **Methods:**
 - **void** save(**object** o) – saves the item to the disk .
 - **object** load() – Returns the items in the bin file.

messagesHandler : fileHandler

- **Attributes:**
 - **string** binPath – Stores the path to the bin file.
- **Methods:**
 - **void** save(**object** o – saves the item to the disk .
 - **object** load() – Returns the items in the bin file.

usersHandler : fileHandler

- **Attributes:**
 - **string** binPath – Stores the path to the bin file.
- **Methods:**
 - **void** save(**object** o) – saves the item to the disk .
 - **object** load() – Returns the items in the bin file.
 - **load()** – Returns the items in the xml file.

ILogger - downloaded

Logger - downloaded

FileLogger – downloaded

UML:

