Low Level Design

Presentation Layer

GUI

Action
+action(List<GuiMessage> msgs) : List<GuiMessage>

LoginWindow.xaml.cs

- chatRoom : ChatRoom
- mainWindow : MainWindow
- obs : ObservableObject
- hashing : Hashing
- HashedPassword : string
- salt : string

- isPassOnlySpaces(String pb) : bool
- PasswordValidity (String pb) : bool

GuiMesaage

- body : string
- userName : string
- g_id : string
- dateTime : DateTime
- id : string

+ toString() : string

ChatRoomWindow.xaml.cs

- msgLength : int
- chatRoom : ChatRoom
- mainWindow : MainWindow
- obs : ObservableObject
- isOptionVisible : bool
- op : Options
- orderChoice : string
- filterChoice : string
- sortChoisce : string
- sortAction : Action
- filterInfo : string[]
- order : int
- msgs : List<GuiMessage>
- groups : List<string>
- dispatcherTimer : DispatcherTimer
- listBox : ListBox

+ getMembersOf(string g_id) : List<string>
- getMessagesList() : void
- isMsgOnlySpaces() : bool

Message.xaml.cs

- chatRoom : ChatRoom
- obs : ObservableObject
- msg : GuiMessage

Options.xaml.cs

- chatRoomWindow : ChatRoomWindow
- obs : ObservableObject
- orderChoice : string
- filterChoice : string
- sortChoisce : string
- tOrderChoice : string
- tFilterChoice : string
- tSortChoisce : string
- prevOrder : string
- prevFilter : string
- prevSort : string
- users : List<string>
- groups : List<string>
- userChoice : string
- groupChoice: string

RegisterWindow.xaml.cs

- chatRoom : ChatRoom
- mainWindow : MainWindow
- obs : ObservableObject
- hashing : Hashing
- HashedPassword : string
- salt : string
- nickname : string
- groupId : string
- correctPass : bool

- isPassOnlySpaces(String pb) : bool
- PasswordValidity (String pb) : bool

ObservableObject

+ PropertyChanged : PropertyChangedEventHandler
- txtSendContent : string
- isOptionVisible : object
-btnSendIsEnabled : bool
- listBoxSelectedIndex : int
- groupIdContent: string
- nicknameContent: string
- lblAddRegContent : string
- btnRegisterVisibility : string
- lblAddRegVisibility : string
- btnLoginIsEnabled : bool
-btnRegIsEnabled: bool
-groupIdText : string
- nicknameText: string
-lblRegErrorVisibility: string
-lblRegErrorContent: string
-btnLoginVisibility: string
-lblAddLoginContent : string
-lblAddLoginVisibility : string
-ascendingIsChecked : bool
-descendingIsChecked: bool
- selectedGroup : int
- isGroupFiltered : string
- isUserFiltered: string
- f_NoneIsChecked : bool
- f_UserIsChecked: bool
- f_GroupIsChecked : bool
-s_AllIsChecked : bool
-s_TimeIsChecked : bool
- s_NicknameIsChecked: bool

+ OnPropertyChanged([CallerMemberName] string propertyName= null) : void

Business (Logic) Layer

ChatRoom

- currUser : User
- url : string
- allMessages : MessageHandler
- allUsers : UserHandler
- prevMsgs : List<GuiMessage>
- sort : PresentationLayer.Action
- filter : string[]

+ login(string nickname, string g_id, string pass) : bool
+ findUser(string nickname, string g_id) : User
+ getMembersOf(string g_id) : List<string>
+ register((string nickname, string g_id, string pass) : bool
+ getGroups() : List<string>
+ send (string message) : bool
+ updateMessage(string newMsg, GuiMessage msg) : bool
+ getMessages(int order, PresentationLayer.Action sortAction, string[] filterInfo) : List<GuiMessage>
+ retieveMessages : void
- updatePresMessages : void
+ logOut() : void

Message

- body : string
- user : User
- dateTime : DateTime
- id : Guid

+ IEquatable<Message>.Equals(Message other) : bool
+ ToString() : string
+ CompareTo(Object other) : int

Hashing

+ GetHash(string inputString) : byte[]
+ GetHashString(string inputString) : string

SortByGNT

+ action(List<GuiMessage> msgs) : List<GuMessage>
+ Compare(GuiMessage msg1, GuiMessage msg2) : int

User

- nickname : string
- g_id : string
- loggedIn : bool
- password : string
- user_id : string

+ logout() : void
+ isEqual(string nickname, int g_id, string pass) : bool
+ toString() : string

SortByName

- comparator : MessageComparator

+ action(List<GuiMessage> msgs) : List<GuiMessage>
+ sortRange(int i, int range, List<GuiMessage> msgs) : void
+ Compare(GuiMessage msg1, GuiMessage msg2) : int

SortByTime

- comparator : MessageComparator

+ action(List<GuiMessage> msgs) : List<GuiMessage>
+ sortRange(int i, int range, List<GuiMessage> msgs) : void
+ Compare(GuiMessage msg1, GuiMessage msg2) : int

Persistent Layer

HandlerFactory

- connection : SqlConnection

+ createUserHandler() : UserHandler
+ createMessageHandler() : MessageHandler

MessageHandler

- list : List<Message>
- _name : string
- _id : string
- connectionFail : bool

+ filterByNone() : bool
+ send(Message msg) : bool
+ updateMessage(string guid, string body, DateTime time) : bool
+ retrive(DateTime time) : List<Message>
+ FilterByGroup(string g_id) : bool
+ FilterByUser(string nickname, string g_id) : bool

UserHandler

- allUsersList : List<User>
- userExist : User
- exist : bool
- connectionFail : bool

+ addUser(User user) : bool
+ getAllUsers : void
+ doesExist(string nickname, string g_id) : bool
+ getMembers(string g_id) : List<User>
+ getUserById(string user_id) : User
+ FilterByUser(string nickname, string g_id) : bool

ConnectionHandler

- connection : SqlConnection

+ connect() : void
+ disconnect () : void
+ setCon(SqlConnection con) : void