# SOFTWARE REQUIREMENTS SPECIFICATION

## *GLOBAL OVERVIEW*

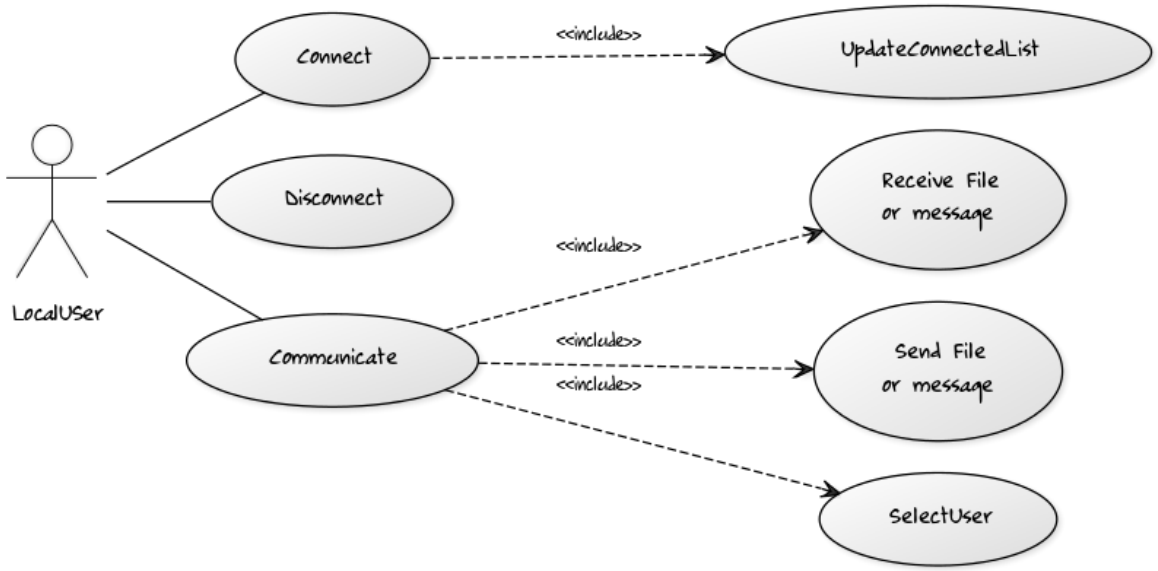
The purpose of this specification is to present the requirements for a distributed (p2p) chat system. This system will allow users to communicate by sending and receiving text messages using interconnected devices. Optionally, users can also communicate by sending and receiving files (i.e. pictures, documents, programs, etc). The following are functional requirements of this system:

- Every user uses an username (or nickname) to connect to the chat system.
- When a user connects to the system, the list of the other connected users is presented. This list includes connected user names and information about their remote system (i.e. remote host information).
- Only connected users are able to communicate using the chat system functions.
- When any user connect (or log on) or disconnect (or log off), the other users have to be informed about it.
- When an user wants to communicate with another user (send a message or send a file), he has to select the remote user from the connected users' list. The message/file to be sent needs to be indicated. Optionally, a group of connected users could be selected as the destination.
- When the system receives a message or file targeted to the connected local user, the user has to be informed about it (i.e. showing the message or an indication about the received file).

Technical requirements such as the kind of terminal, the network environment or the user interface need to be specified and refined during the analysis and design process.

## *USE CASE DIAGRAM*

[LocalUSer]-(Connect)
[LocalUSer]-(Disconnect)
[LocalUSer]-(Communicate)
(Connect)>(UpdateConnectedList)
(Communicate)>(SelectUser)
(Communicate)>(Send File or message)
(Communicate)>(Receive File or message)




***SCENARIOS***

| | |
|---|---|
| Use Case Id | 1.1 |
| Version | 1.0 |
| Name | connection |
| Actors | local user |
| Description | This use case describes the connection to the system of a local user |

| | |
|---|---|
| Purpose/Overview | This use case is aimed at allowing the local user to connect to the system. The user press the connect button after providing a valid login or user name. The system indicates to the user if the connection is sucessful or not. |
| Triggers | connect button |
| Preconditions | - a username has been provided<br>- the user is not already connected (he is disconnected) |
| Post-conditions | the user is connected |
| Business rules | - a valid username is an unique identifier for the user<br>- when a user gets connected, all the other users already connected need to be informed about the arrival of the new user. A hello message containing the local user name is used. |
| Notes | several posibilities to provide an unique identifier exist:<br>- the login is compared with the other users to check is valid |

| | |
|---|---|
| | - the login is automatically converted in a valid login by adding the unique host address |
| Author and date | Neumann, Tribhout, Manolias 15.09.2014 |
| Basic course of events | |
| **Local User** | **ChatSystem** |
| provides username | |
| press connect button | |
| | sends hello messages to other users |
| | indicates that local user is connected |
| **Alternative path** | |
| | indicates that local user is not connected because the username is not valid |

| | |
|---|---|
| Use Case Id | 1.2 |
| Version | 1.0 |
| Name | connection |
| Actors | remote user |
| Description | This use case describes the connection to the system of a local user |

| | |
|---|---|
| Purpose/Overview | This use case is aimed at updating the list of connected people for the local user whenever a remote user is connecting |
| Triggers | a thread is listenning and waiting the “hello” message from remote user |
| Preconditions | - the  remote user is disconnected<br>- the remote user press the connect button |

| | |
|---|---|
| | - the local user is already connected |
| Post-conditions | the local user's list of connected people is updated |
| Business rules | - the remote user send a hello message whenever he's connecting, to inform he's actually connected. Then we are always waiting for remote users connection through a thread. |
| Notes | several posibilities to provide an unique identifier exist:<br>- the login is compared with the other users to check is valid<br>- the login is automatically converted in a valid login by adding the unique host address |
| Author and date | Neumann, Tribhout, Manolias 15.09.2014 |
| Basic course of events | |
| **local system** | **remote system** |
| | send hello |
| update list of connected people, reply with an hello message | |
| | update list of connected people |
| | |
| **Alternative path** | |

| | |
|---|---|
| | indicates that remote user is not connected because the username is not valid |

| | |
|---|---|
| Use Case Id | 2.1 |
| Version | 1.0 |

| | |
|---|---|
| Name | communication |
| Actors | local user |
| Description | This use case describes the send and receive actions between a local and remote users. |

| | |
|---|---|
| Purpose/Overview | Only the connected users are able to communicate. By clicking on a remote username in the connected user list, the local user is able to send messages or files. He is also able to read received messages in his chatbox. The system indicates to the user if the message was successfully sent or received or not. |
| Triggers | users names list button |
| Preconditions | - the remote username is in the connected users list<br>- the local user  is connected |
| Post-conditions | chatbox closed |
| Business rules | - a valid username is required<br>- when a message is sent or received, the local user is notified in his own chatbox |
| Notes | |
| Author and date | Neumann, Tribhout, Manolias 15.09.2014 |
| Basic course of events | |
| **Local User** | **chatsystem** |

| | |
|---|---|
| select connected user | |
| | display a dialog box |
| select a file or/and write a message | |
| | open a file manager / or write the message in the remote dialog box |
| press send button | |
| | the message is sent |
| | the message has been sucessfully received by the remote user |
| **Alternative path** | |
| | indicates that remote user is not connected anymore<br>indicates if the sending has failed |

| | |
|---|---|
| Use Case Id | 2.2 |
| Version | 1.0 |
| Name | communication |
| Actors | remote user |

| | |
|---|---|
| Description | This use case describes the initiation of a connection by the local user |

| | |
|---|---|
| Purpose/Overview | Only the connected users are able to communicate. By clicking on a remote users in their connected userlist |
| Triggers | users names list button |
| Preconditions | - the local username is in the connected users list of the initiator(remote user here)<br>- the initiator is connected |
| Post-conditions | chatbox closed |
| Business rules | - a valid username is required<br>- when a message is sent or received, the local user is notified in his own chatbox |
| Notes | |
| Author and date | Neumann, Tribhout, Manolias 15.09.2014 |
| Basic course of events | |
| **local system** | **remote system** |
| | send a message |
| receive a message/ send ack | |
| display a chatbox for the local user | notify message has been received |

| | |
|---|---|
| display the message in the chatbox | |
| | |
| | |
| | |
| **Alternative path** | |
| | indicates that remote user is not connected anymore<br>indicates if the sending has failed |

| | |
|---|---|
| Use Case Id | 3.1 |
| Version | 1.0 |
| Name | disconnection |
| Actors | local user |
| Description | This use case describes the disconnection to the system of a local user |

| | |
|---|---|
| Purpose/Overview | This use case is aimed at allowing the local user to disconnect to the system. The system indicates to the user if the disconnection is sucessful or not. |
| Triggers | disconnect button |
| Preconditions | - the user is already connected |
| Post-conditions | the user is disconnected |
| Business rules | - when a user gets disconnected, all the other users already connected need to be informed about the departure of the new user. A farewell message containing the local user name is used. |
| Notes | |
| Author and date | Neumann, Tribhout, Manolias 22.09.2014 |

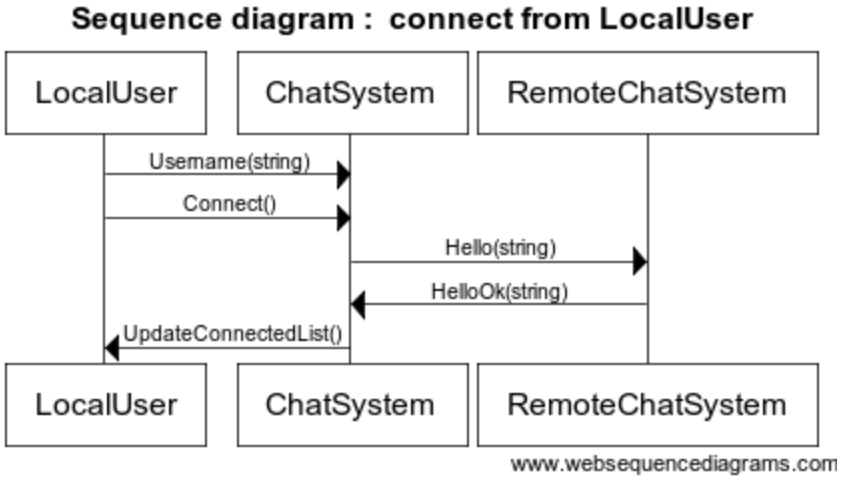
| Basic course of events | |
|---|---|
| **Local User** | **ChatSystem** |
| press disconnect button | |
| | sends farewell message to other users |
| | indicates that local user is disconnected |
| **Alternative path** | |

## *SEQUENCE DIAGRAM*

### Sequence diagram : Connect from LocalUSer

*title Sequence diagram :  connect from LocalUser*

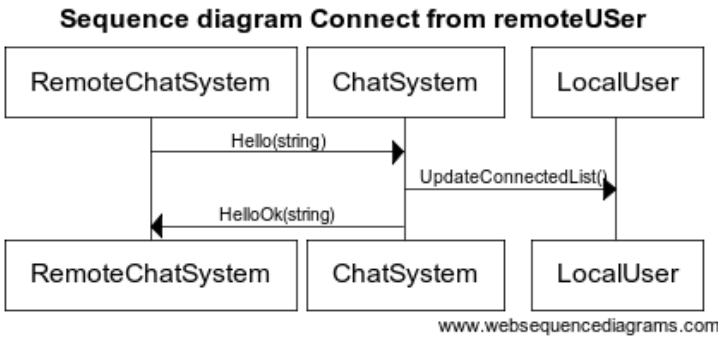
*LocalUser->ChatSystem : Username(string)*
*LocalUser->ChatSystem : Connect()*
*ChatSystem->RemoteChatSystem : Hello(string)*
*UpdateConnectedList(string,boolean)*
*RemoteChatSystem->ChatSystem : HelloOk(string)*
*ChatSystem->LocalUser : UpdateConnectedList()*




### Sequence diagram : receiving connect from remoteChatSystem

*title Sequence diagram Connect from remoteUSer*

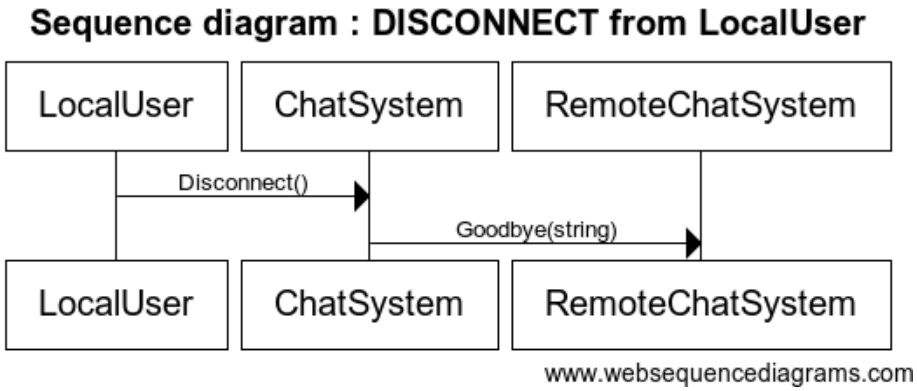
*RemoteChatSystem->ChatSystem : Hello(string)*
*ChatSystem->LocalUser : UpdateConnectedList()*
*ChatSystem-> RemoteChatSystem : HelloOk(string)*




## Sequence diagram : Disconnect from LocalUSer

*title Sequence diagram : DISCONNECT from LocalUser*

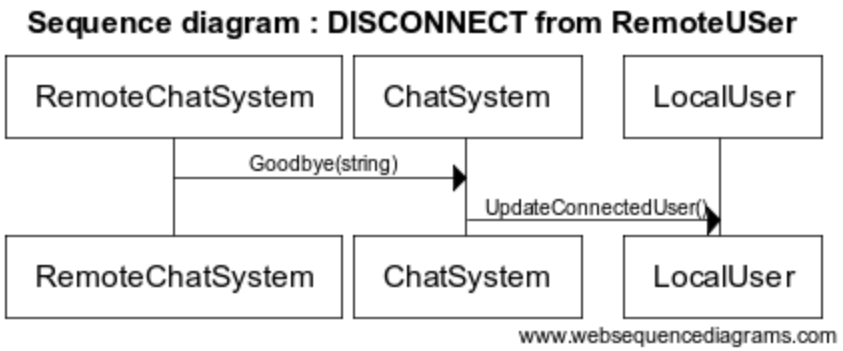
*LocalUser->ChatSystem : Disconnect()*
*ChatSystem->RemoteChatSystem : Goodbye(string)*




## Sequence diagram : receiving Disconnect from RemoteChatSystem

*title Sequence diagram : DISCONNECT from RemoteUSer*

*RemoteChatSystem->ChatSystem : Goodbye(string)*
*ChatSystem->LocalUser: UpdateConnectedUser()*

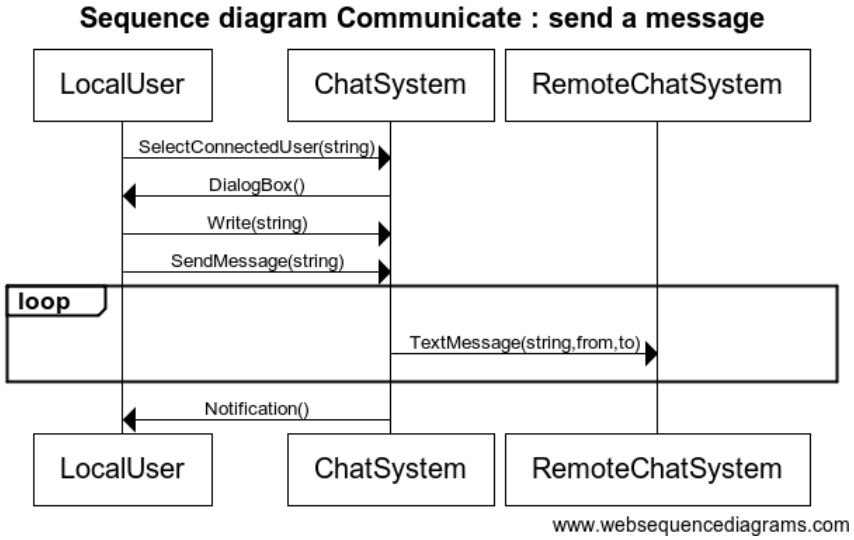



## Sequence diagram : Communicate [Local User]

## Sequence diagram : Communicate : send a message

*title Sequence diagram Communicate : send a message*

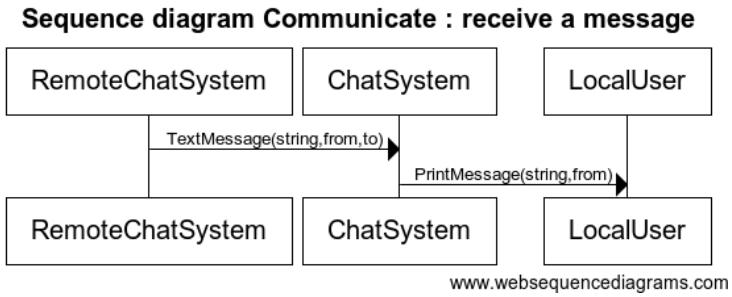
*LocalUser->ChatSystem : SelectConnectedUser(string)*
*ChatSystem->LocalUser : DialogBox()*
*LocalUser->ChatSystem : Write(string)*
*LocalUser->ChatSystem : SendMessage(string)*
*loop*
*ChatSystem->RemoteChatSystem : TextMessage(string,from,to)*
*end*
*ChatSystem->LocalUser : Notification()*




## Sequence diagram : Communicate : receive a message

*title Sequence diagram Communicate : receive a message*

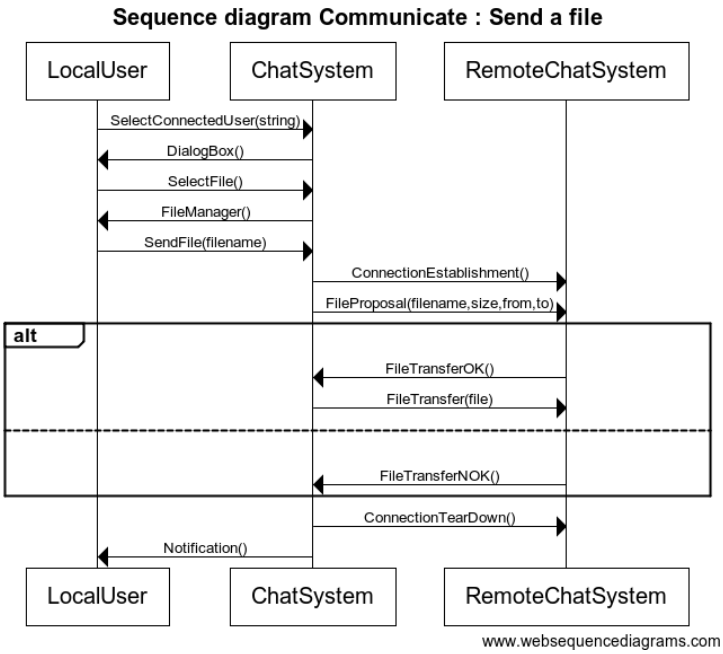
*RemoteChatSystem->ChatSystem : TextMessage(string,from,to)*
*ChatSystem->LocalUser : PrintMessage(string,from)*




## Sequence diagram : Communicate : send a file

*title Sequence diagram Communicate : Send a file*

*LocalUser->ChatSystem : SelectConnectedUser(string)*
*ChatSystem->LocalUser : DialogBox()*
*LocalUser->ChatSystem : SelectFile()*
*ChatSystem->LocalUser : FileManager()*
*LocalUser->ChatSystem : SendFile(filename)*
*ChatSystem->RemoteChatSystem : ConnectionEstablishment()*
*ChatSystem->RemoteChatSystem : FileProposal(filename,size,from,to)*
*alt*
*RemoteChatSystem->ChatSystem : FileTransferOK()*
*ChatSystem->RemoteChatSystem : FileTransfer(file)*
*else*
*RemoteChatSystem->ChatSystem : FileTransferNOK()*
*end*
*ChatSystem->RemoteChatSystem : ConnectionTearDown()*
*ChatSystem->LocalUser : Notification()*

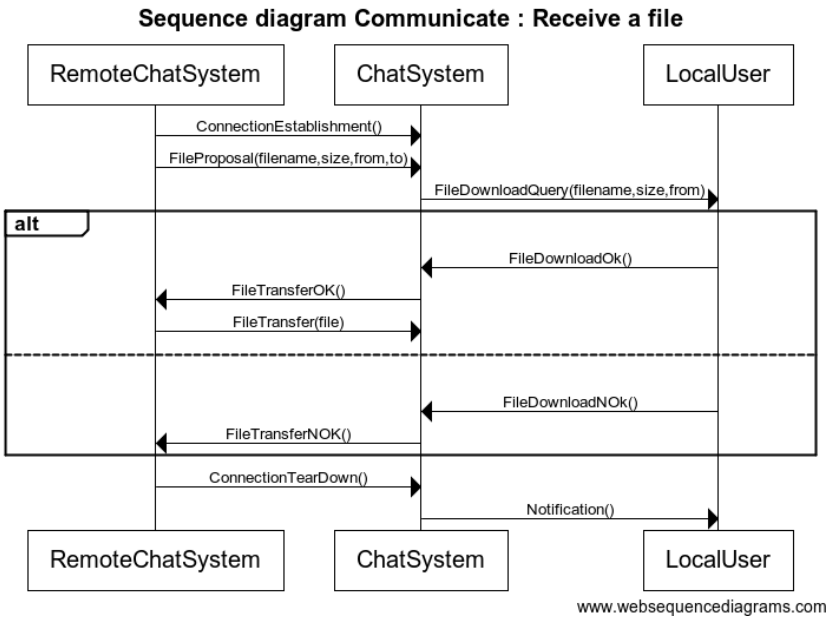



## Sequence diagram : Communicate : Receive a file

*title Sequence diagram Communicate : Receive a file*

*RemoteChatSystem->ChatSystem : ConnectionEstablishment()*
*RemoteChatSystem->ChatSystem : FileProposal(filename,size,from,to)*

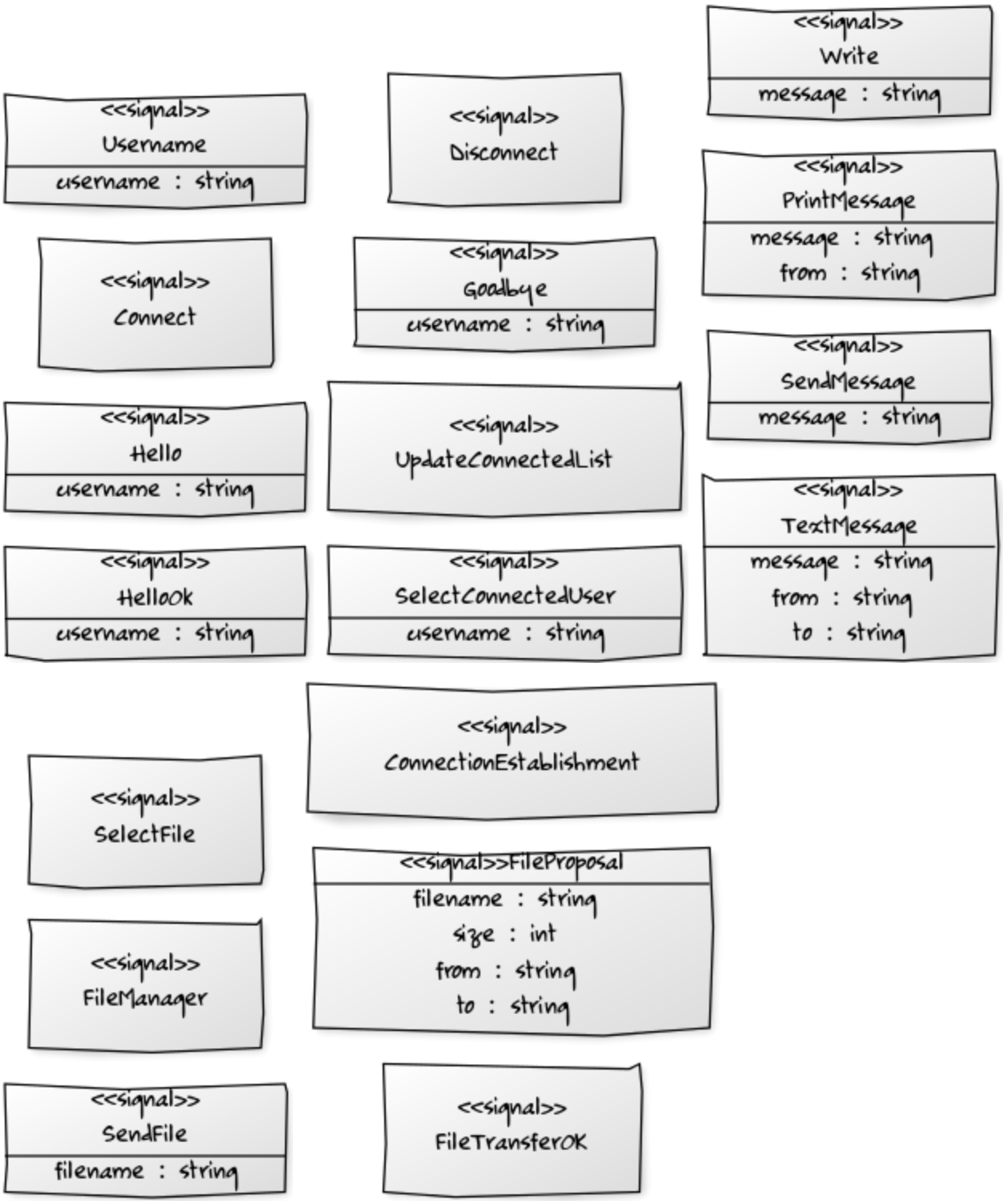
*ChatSystem->LocalUser : FileDownloadQuery(filename,size,from)*
*alt*
*LocalUser->ChatSystem : FileDownloadOk()*
*ChatSystem->RemoteChatSystem : FileTransferOK()*
*RemoteChatSystem->ChatSystem : FileTransfer(file)*
*else*
*LocalUser->ChatSystem : FileDownloadNOk()*
*ChatSystem->RemoteChatSystem : FileTransferNOK()*
*end*
*RemoteChatSystem->ChatSystem : ConnectionTearDown()*
*ChatSystem->LocalUser : Notification()*

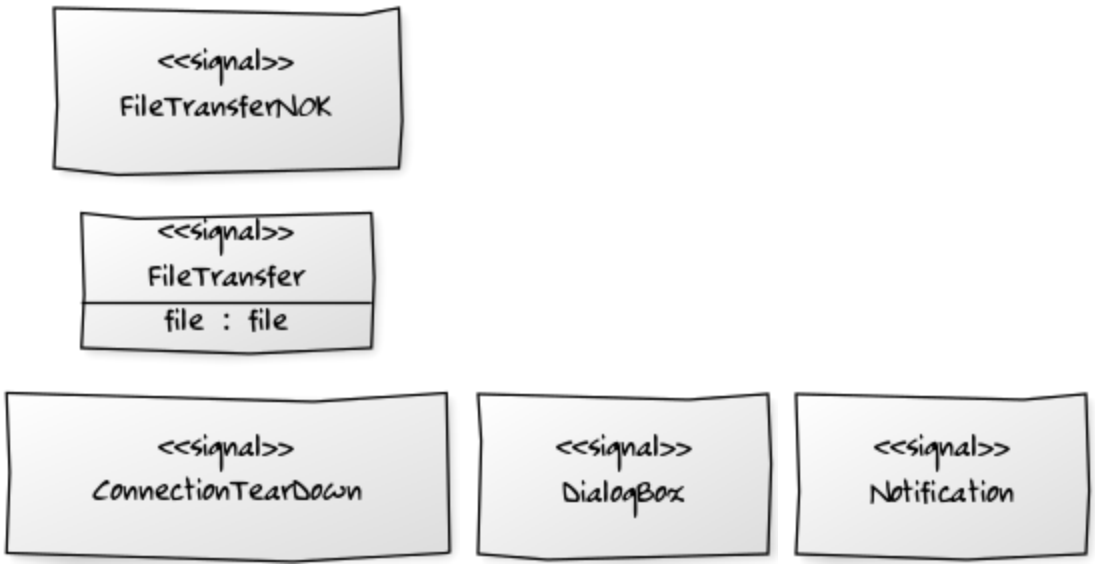



## CLASS DIAGRAM

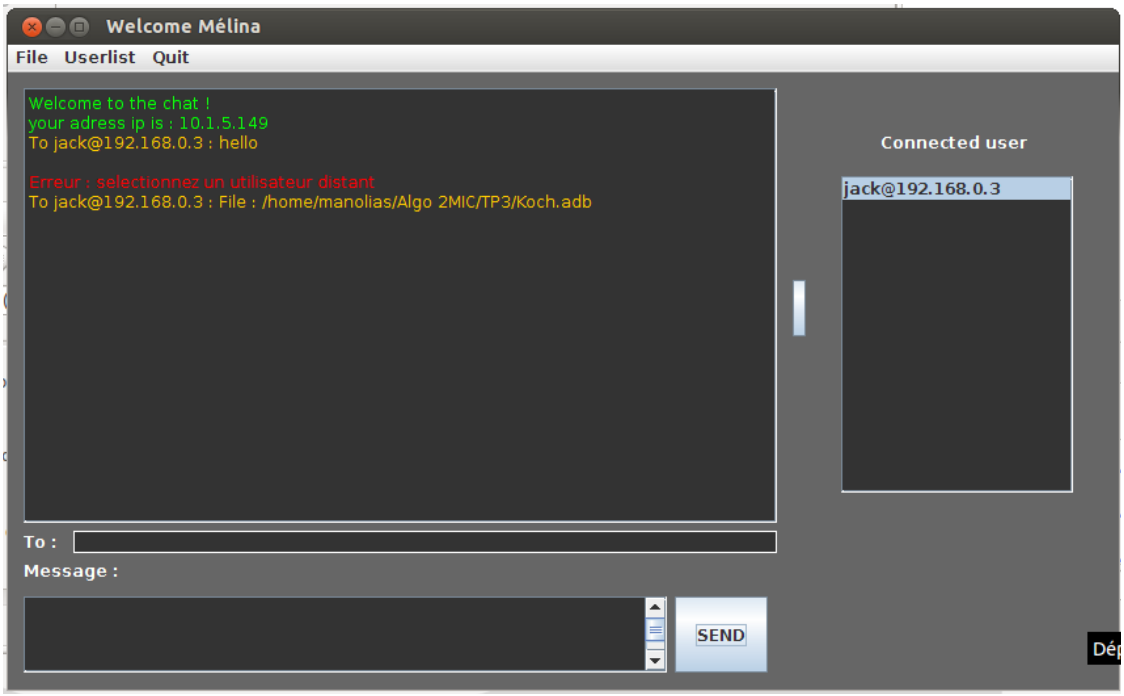
[ChatSystem]
[\<\<signal\>\> Username | username : string]
[<<signal>> Connect]
[\<\<signal\>\> Hello|username : string]
[\<\<signal\>\> HelloOk|username : string]

[\<\<signal\>\> Disconnect]
[\<\<signal\>\> Goodbye|username : string]
[\<\<signal\>\> UpdateConnectedList]
[\<\<signal\>\> SelectConnectedUser | username : string]
[\<\<signal\>\> Write|message : string]
[\<\<signal\>\> PrintMessage | message : string ; from : string]
[\<\<signal\>\> SendMessage | message : string]
[\<\<signal\>\> TextMessage | message : string ; from : string ; to : string]
[<<signal>> SelectFile]
[<<signal>> FileManager]
[\<\<signal\>\> SendFile | filename : string]
[<<signal>> ConnectionEstablishment]
[\<\<signal\>\>FileProposal | filename : string ; size : int ; from : string ; to : string]
[<<signal>> FileTransferOK]
[<<signal>> FileTransferNOK]
[<<signal>> FileTransfer | file : file]
[<<signal>> ConnectionTearDown]
[<<signal>> DialogBox]
[<<signal>> Notification]


<<signal>>
Username
username : string
<<signal>>
Disconnect
<<signal>>
Write
message : string
<<signal>>
Connect
<<signal>>
Goodbye
username : string
<<signal>>
PrintMessage
message : string
from : string
<<signal>>
SendMessage
message : string
<<signal>>
Hello
username : string
<<signal>>
UpdateConnectedList
<<signal>>
TextMessage
message : string
from : string
to : string
<<signal>>
HelloOk
username : string
<<signal>>
SelectConnectedUser
username : string
<<signal>>
ConnectionEstablishment
<<signal>>
SelectFile
<<signal>>FileProposal
filename : string
size : int
from : string
to : string
<<signal>>
FileManager
<<signal>>
SendFile
filename : string
<<signal>>
FileTransferOK


<<signal>>
FileTransferNOK
<<signal>>
FileTransfer
file : file
<<signal>>
ConnectionTearDown
<<signal>>
DialogBox
<<signal>>
Notification

## MOCKUP




## BLACK BOX