Software Requirements Specification for P2P Chat System

Requirements for Version 1.0

ASMA SOUFI, YUANBO WANG

4IR Group A2, 2014/2015

Table of contents

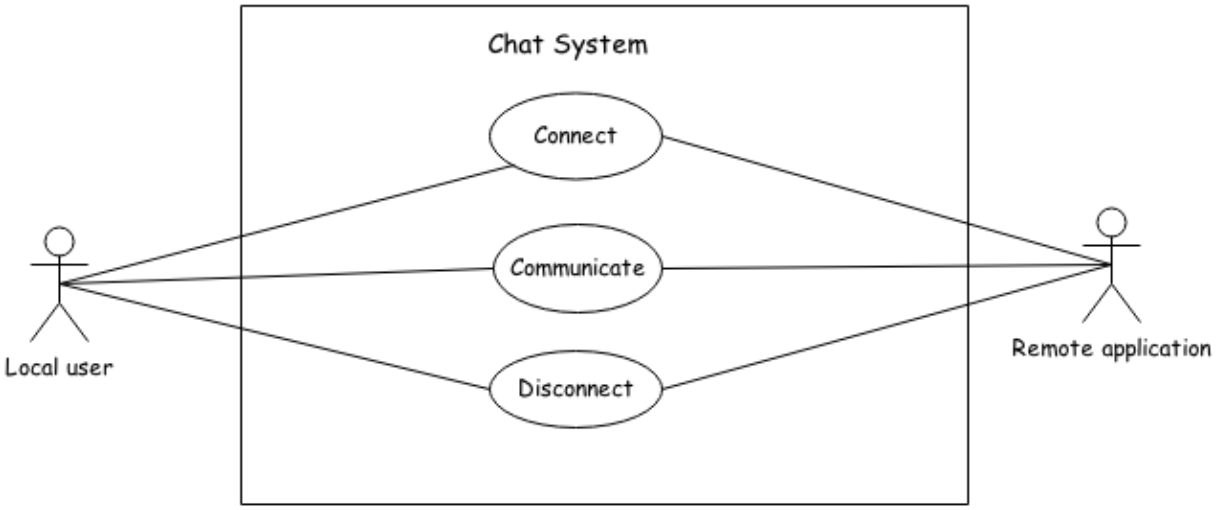
Overview of requirements

The Peer-to-Peer Chat system will should 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 a 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).

Use case diagram

For the Peer-to-Peer Chat system, we have defined two actors which are “Local user” and “Remote application”. The use cases are respectively “Connect”, “Communicate” and “Disconnect”, where “Communicate” can be devided into more scenarios like “send” or “receive”.



Scenarios

Use case ID	Use case name	Scenario ID	Scenario name	Actors involved
1	Connect	**1.1**	connection initiated by local user	local user
1	Connect	**1.2**	connection initiated by remote user	remote application
2	Disconnect	**2.1**	disconnection initiated by local user	local user
2	Disconnect	**2.2**	disconnection initiated by remote user	remote application
3	Communicate	**3.1**	local user sends messages	local user, remote application
3	Communicate	**3.2**	local user receives messages	local user, remote application
3	Communicate	**3.3**	local user sends files	local user, remote application
3	Communicate	**3.4**	local user receives files	local user, remote application

Use Case ID	1
Scenario ID	1.1
Version	1.0
Name	connection initiated by the local user
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 presses the connect button after providing a valid login or user name. The system indicates the user if the connection is successful or not.
Triggers	connect button
Preconditions	1. A username has been provided 2. The user is not already connected (he is disconnected)
Post-conditions	the user is connected

Business rules	1. A valid username is an unique identifier for the user 2. 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 information (IP address) is used.	
Notes	To identifier the unicity of every user name, the login offered by the new user is compared with the other users to assure its validity.	
Author and date	Yuanbo WANG, Asma SOUFI 14.9.2014	
Basic course of events		
Local user	**Chat system**	**Remote application**
provides a valid username (nickname)		
presses connect button		
	sends a hello message containing the local user's nickname to all other connected users(broadcast)	
		receives the hello message, and sends back a helloACK containing the nickname of remote user
	receives the helloACK, adds the remote user name to the connected user list of the local user, then displays the connected user list	
Alternative path		
	indicates that local user is not connected because the username is not valid	

Use Case ID	1
Scenario ID	1.2
Version	1.0
Name	connection initiated by the remote user
Actors	remote application
Description	This use case describes the connection to the system of a remote user

Purpose/Overview	This use case is aimed at allowing the remote user to connect to the system.
Triggers	connect button

Preconditions	- A username has been provided - The remote user is not already connected (he is disconnected)
Post-conditions	the remote user is connected
Business rules	- A valid username is an unique identifier for the user - When the remote user gets connected, all the other users already connected need to be informed about the arrival of the remote user. A hello message containing the remote user name is used.
Notes	To identifier the unicity of every user name, the login offered by the new user is compared with the other users to assure its validity
Author and date	Yuanbo WANG Asma SOUFI, 14.9.2014
Basic course of events	
Chat system	**Remote application**
	sends a hello message to local chat system containing its IP address
receives the hello message from remote application and sends the helloACK back. Then adds the remote user's nickname to local user's connected user list	
	receive ACK of hello message from local chat system and adds the local user name to its connected user list

Use Case ID	2
Scenario ID	2.1
Version	1.0
Name	disconnection initiated by the local user
Actors	local user
Description	This use case describes how the local user disconnects from the chat system

Purpose/Overview	This use case is aimed at allowing the local user to disconnect from the chat system. The user presses the disconnect button, a window pops up and let the local user confirm if he really wants to leave. If the local user presses "yes", the system indicates to the user if the disconnection is successful or not.
Triggers	disconnect button
Preconditions	the user is 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 leaving of the user. A goodbye message containing the local user name is used. - When the user presses the disconnect button, he is automatically removed from other connected users' list.	
Notes		
Author and date	Yuanbo WANG,Asma SOUFI 14.9.2014	
Basic course of events		
Local user	**Chat system**	**Remote application**
presses disconnect button		
	pops up a window to ask if the user really wants to disconnect from the chat system	
presses "yes"		
	sends a goodbye message("User xx is leaving the chat.") to other users containing its user name	
		receives the goodbye message and removes this user from its connected user list
	pops up a window to indicate that the local user has been successfully disconnected : "you are now offline"	
Alternative path		
	If user presses "no" in the confirming window, then local user will be still connected.	

Use Case ID	2
Scenario ID	2.2
Version	1.0
Name	disconnection initiated by the remote user
Actors	remote application
Description	This use case describes how the remote user disconnects from the chat system

Purpose/Overview	This use case is aimed at allowing the remote user to disconnect from the system. The procedure is similar to local user disconnection scenario.
Triggers	disconnect button
Preconditions	the user is 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 leaving of the user. A goodbye message containing the remote IP adress is used.
Notes	
Author and date	Yuanbo WANG, Asma SOUFI 17.9.2014
Basic course of events	
Chat system	**Remote application**
	sends a goodbye message(“User xx is leaving the chat.”) to other users containing its IP address
receives the leaving message and removes this user from connected user list	
	indicates that remote user has been disconnected : “you are now offline”.

Use Case ID	3
Scenario ID	3.1
Version	1.0
Name	local user sends messages
Actors	local user, remote application
Description	This use case describes how a user can send a message to others.

Purpose/Overview	This use case is aimed at allowing the local user to send messages to others. The local user selects the users to whom the message will be sent, then presses the send button in the dialog window, and the message is sent to the selected user(s).
Triggers	send button
Preconditions	user has selected one or multiple users to send the message
Post-conditions	the message is sent to selected users

Business rules	The message to send should not be null, if so, nothing will happens.	
Notes		
Author and date	Yuanbo WANG, Asma SOUFI 14.9.2014	
Basic course of events		
Local user	**Chat system**	**Remote application**
selects one or several users, writes a message in the message window, and presses “send message” button		
	the message is sent to selected users using UDP protocol	
		the remote application receives the message and sends back ACK

Use Case ID	3
Scenario ID	3.2
Version	1.0
Name	local user receives messages
Actors	local user, remote application
Description	This use case describes how a local user can receive a message from other(s)

Purpose/Overview	This use case is aimed at allowing the local user to receive messages from other users. If local user receives a message from a remote user, a notification should appear in red color beside the message sender, e.g. : “Peter [new message]”. local user can click on the user name, a window pops up and show the message received.
Triggers	none
Preconditions	the user is connected
Post-conditions	notification appears in red color beside the name of the message sender in the connected user list. e.g. : “Peter [new message]”
Business rules	
Notes	
Author and date	Yuanbo WANG, Asma SOUFI 14.9.2014

Basic course of events	
Local user	**Chat system**
	receives the message from other users, then shows a notification in red color beside the sender's user name
presses the user name	
	the message received will be shown in the message frame

Use Case ID	3
Scenario ID	3.3
Version	1.0
Name	local user send files
Actors	local user, remote application
Description	This use case describes how the local user can send files to other users

Purpose/Overview	This use case is aimed at allowing the local user to send a file. The local user select one or multiple users, then presses "choose file to send" button to choose the file to send in local disk, then presses "send" button. The file is sent to selected remote user.
Triggers	send button
Preconditions	the user is connected
Post-conditions	the file is sent
Business rules	
Notes	If a remote user disconnects from the chat system when the dialog window is open for local user, a notification will show automatically to inform the local user that this dialog is no longer available.
Author and date	Yuanbo WANG, Asma SOUFI 14.9.2014
Basic course of events	
Local user	**Chat system**
selects one or multiple users, then presses "choose file to send" button	
	pops up a window to choose file to send

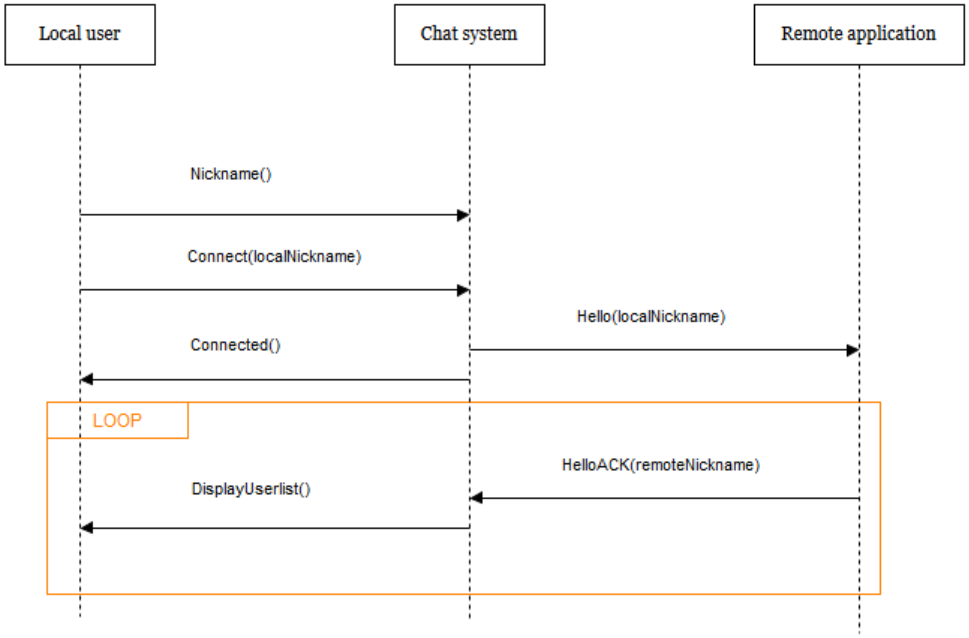
presses “send” button	
	the file is sent to selected users in TCP packet

Use Case ID	3
Scenario ID	3.4
Version	1.0
Name	local user receives files
Actors	local user, remote application
Description	This use case describes how the local user can receive a file from other users

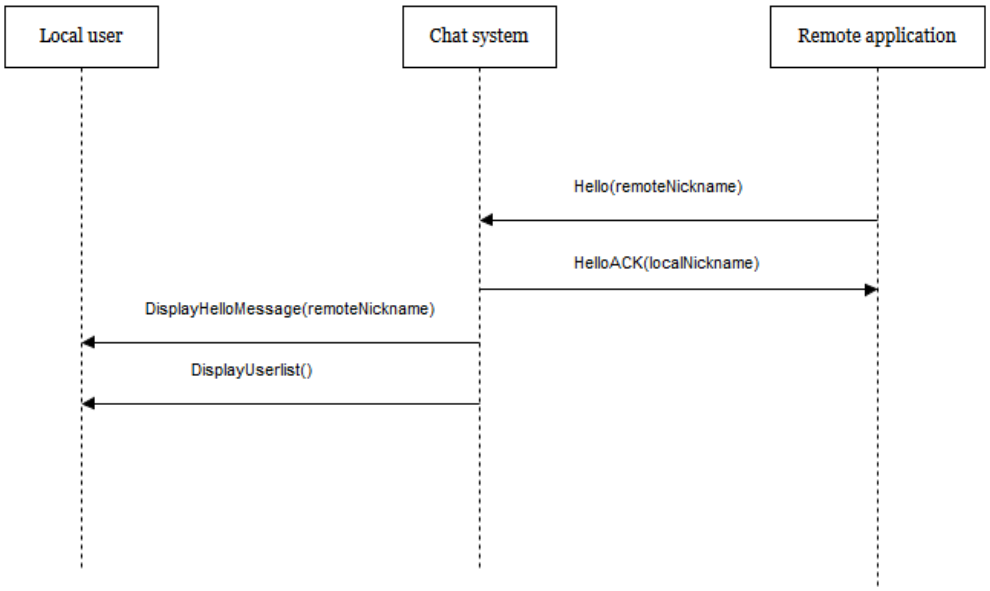
Purpose/Overview	This use case is aimed at allowing the local user to receive files using TCP protocol.
Triggers	none
Preconditions	the user is connected
Post-conditions	the file is received
Business rules	
Notes	
Author and date	Yuanbo WANG, Asma SOUFI 18.9.2014
Basic course of events	
Chat system	**Remote application**
	the file to send is selected by user and sent to selected users
receives the file from remote user, and shows a notification of newly received file in red color	

Sequence diagram (black box approach)

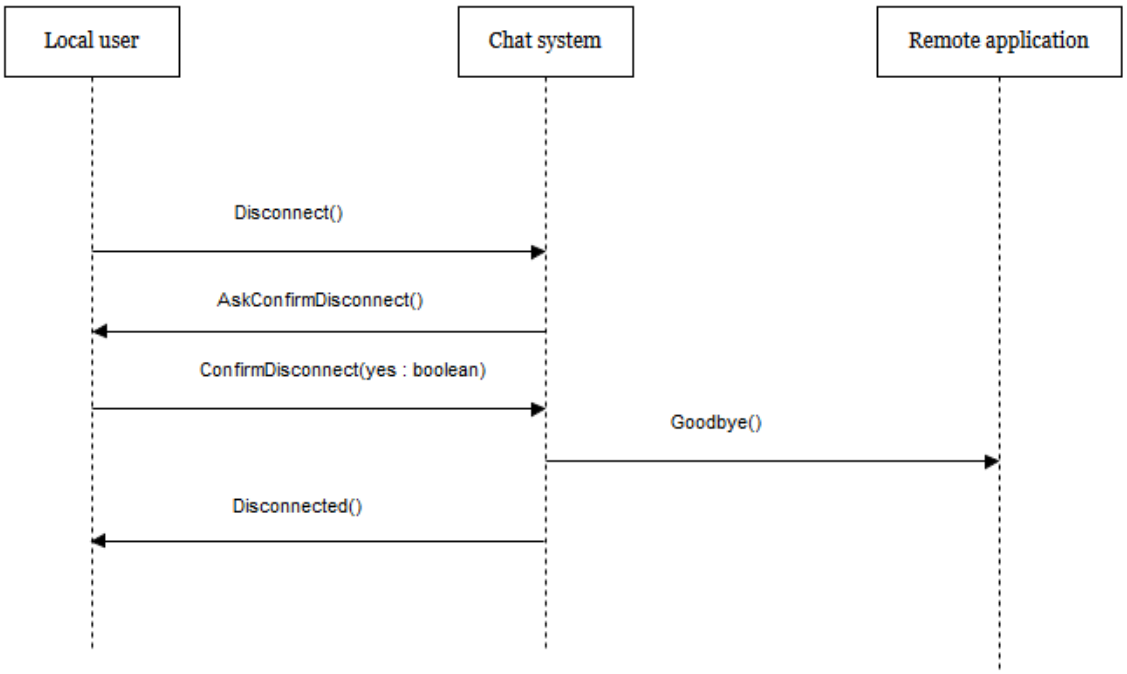
Scenario 1 : Connection initiated by local user



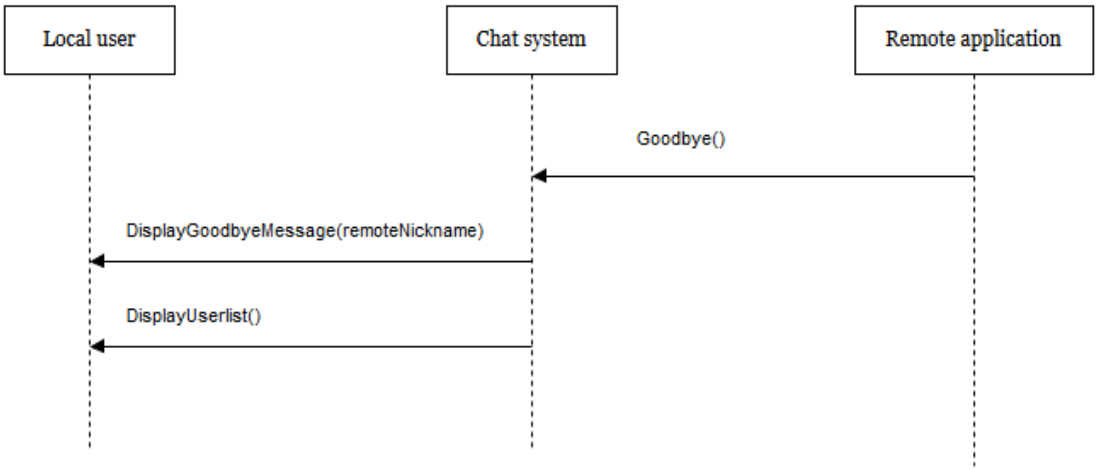
Scenario 2 : Connection initiated by remote user



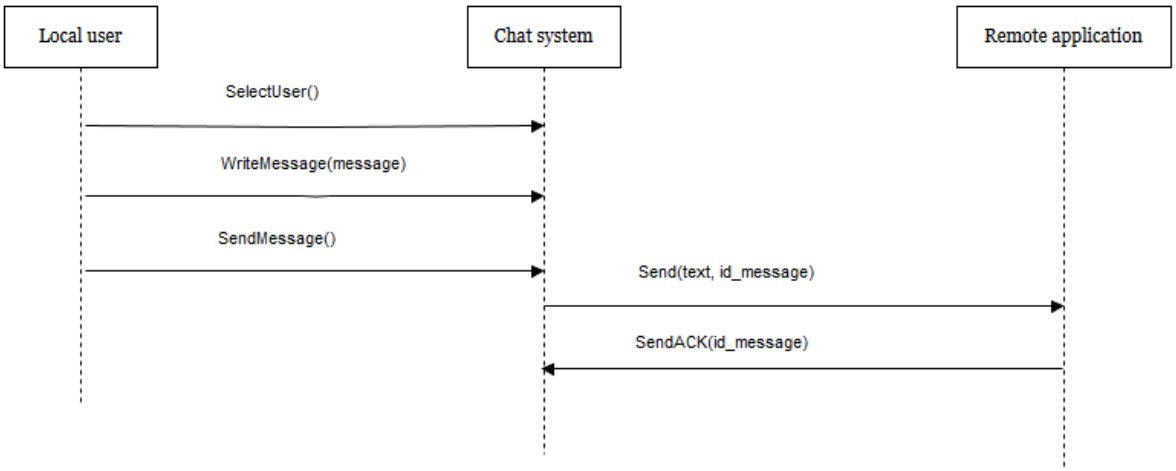
Scenario 3 : Disconnection initiated by local user



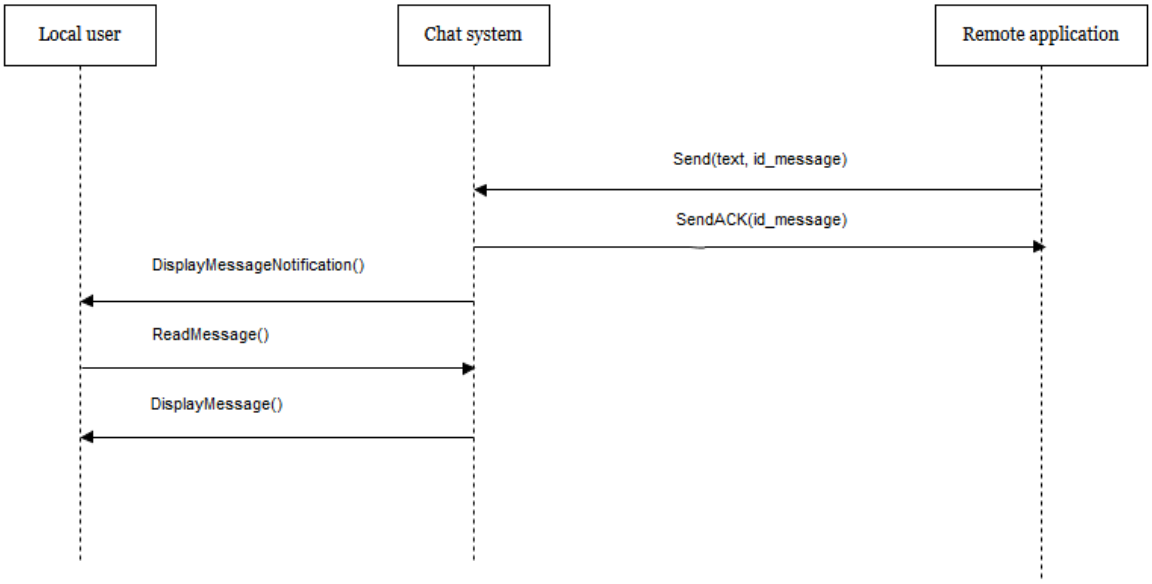
Scenario 4 : Disconnection initiated by remote user



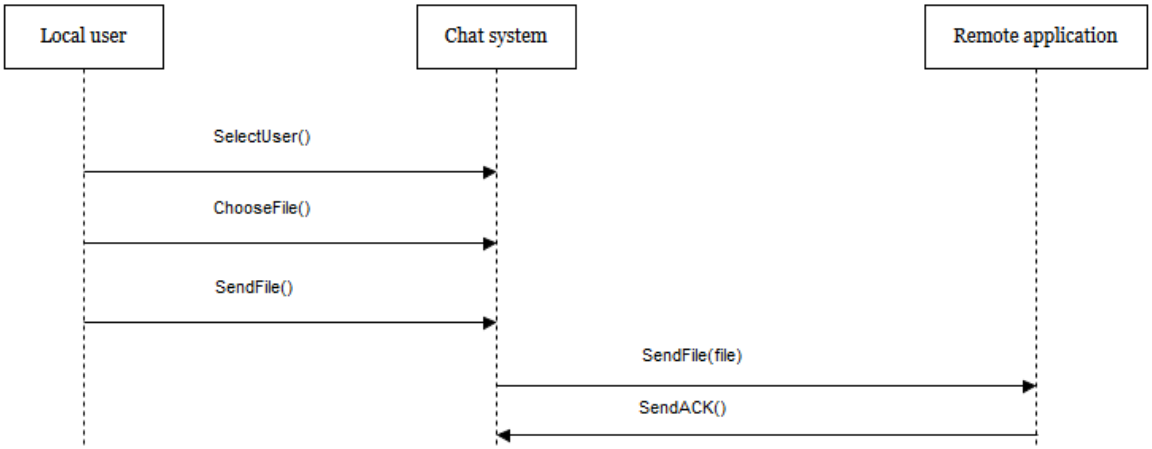
Scenario 5 : Local user sends messages



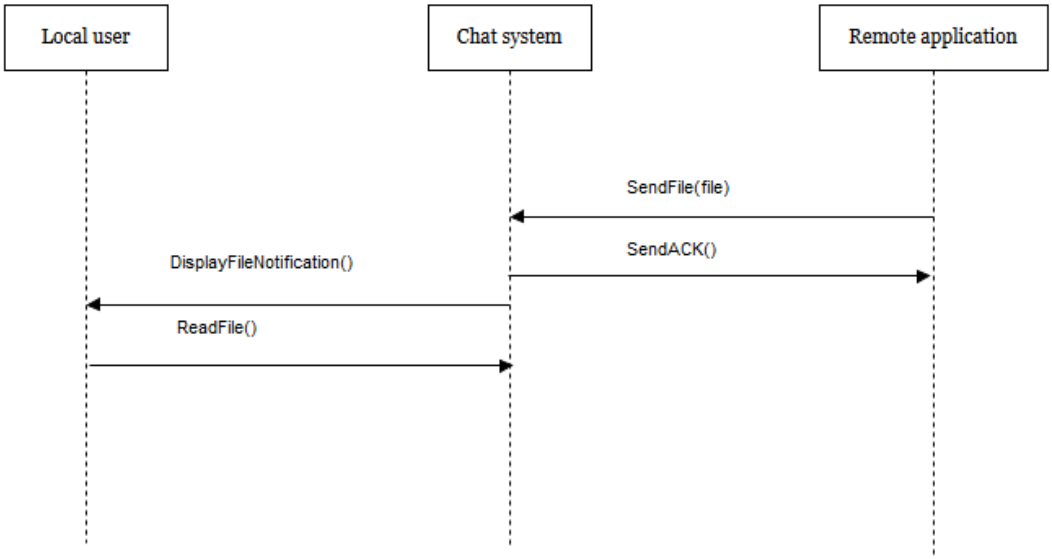
Scenario 6 : Local user receives messages



Scenario 7 : Local user send files



Scenario 8 : Local user receives files



Class diagram (black box approach)

Finally, we constructed a class diagram in a black box approach, showing three interfaces representing all messages that can be sent or received by the Chat System, *to User*, *from User* and *processRemoteChatSystem* which is the equivalent of *from/to remote Chat System*.