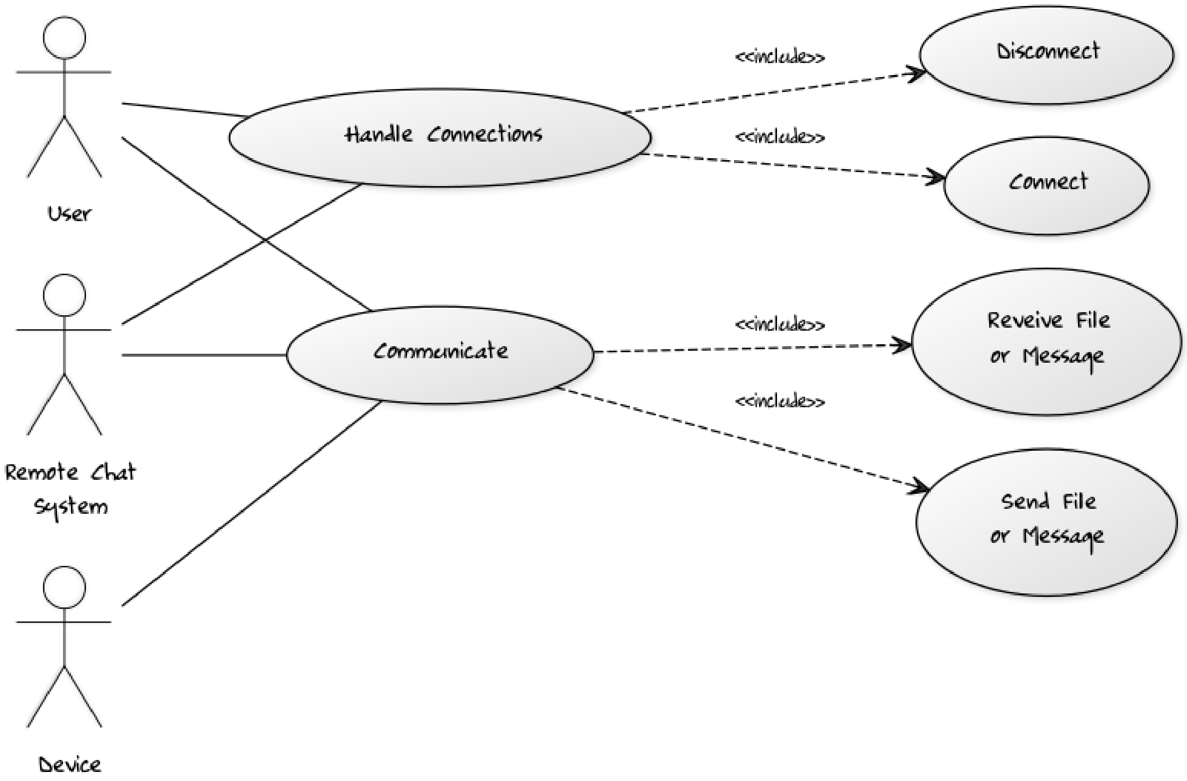
SRS Document

Overview

Here is an informal Software Requirements Specification document which specifies and analyses the requirements of the software to be produced. This P2P software should allow users to communicate by sending and receiving text messages (as well as files) using interconnected devices. When any user connects or disconnects, each host connected to the chat system has to be informed. Thus, each user has a fully updated user list he can use to select people he wants to communicate with. Connected users only can communicate through this chat system. Evidently, when a user receives a message, the chat system informs him about it.

Use case diagram

Analyzing the given SRS permits us to determine that there are three actors and two main uses cases:



(Visible online here : http://yuml.me/5e6e1dca)

Textual use cases

We can describe scenarios from this uses cases. Reminder, there are three ACTORS (User, Remote chat system, Devices (for files)) and four USE CASES : Connect (1) , Disconnect (2), Send (3) & Receive (4).

1) Scenarios for the "Connect" use case

Two scenarios are possible :

a) The user wants to be connected to the chat system : "connect-user"

b) The user is informed that another user is connected : "connect-another"

Use Case Id	1-a
Version	1.0
Name	connect-user
Actors	user & remote chat system
Description	This use case describes the connection to the chat system of a user and the update of the user list.

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 - the user is not already connected (he is disconnected)	
Post-conditions	the user is connected and the list of connected users is updated	

Business rules	- a valid username is an unique identifier for the user - 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: - 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 ● Hello message is sent in UDP broadcast	
Author and date	Jauffret & Gourraud 09/2014	
Basic course of events		
Local User	ChatSystem	Remote Chat System
provides username		
press connect button		
	sends hello messages to other users	
		send ack message to the local user from each remote chat system
	indicates that local user is connected	
	update list with each ack message received	
Alternative path :	indicates that local user is not connected because the username is not valid	

Use Case Id	1-b
Version	1.0
Name	connect-another
Actors	remote chat system
Description	This use case describes the connection to the chat system of a another user, in order to update the local userlist.

Purpose/Overview	This use case is aimed at updating the local user list. When another user connects to the chat system, the local user is informed.	
Triggers		
Preconditions	- another user just connected	
Post-conditions	- the local user list of connected users is updated	
Business rules	- As a user gets connected, the local user already connected has to be informed about the arrival of the new user. On the local user side, a HelloAck message is sent to inform the new user.	
Notes	Hello ack sent in UDP unicast	
Author and date	Jauffret & Gourraud 09/2014	

Basic course of events		
	ChatSystem	Remote Chat System
		sends hello messages to the local user
	Update list	
	send ack message to the new remote chat system	

2) Scenarios for the “Disconnect” use case

Two scenarios are possible :

a) The user wants to be disconnected from the chat system : “disconnect-user”

b) The user is informed that another user is disconnected : “disconnect-another”

Use Case Id	2-a
Version	1.0
Name	disconnect-user
Actors	user & remote chat system
Description	This use case describes the disconnection to the chat system of a user and the update of the user list.

Purpose/Overview	This use case is aimed at allowing the local user to disconnect to the system. The user press the disconnect button or stop the program. In all cases, the program closes after the disconnection.	

Triggers	close button or disconnect button	
Preconditions	- the user isn’t sending/receiving a message or a file (large data size case)	
Post-conditions	the user is disconnected, his program closes and the list of connected users is updated	
Business rules	- when a user gets disconnected, all the other users connected need to be informed about it. A bye message containing the local user name is used.	
Notes	Goodbye message sent in udp broadcast	
Author and date	Jauffret & Gourraud 09/2014	
Basic course of events		
Local User	ChatSystem	Remote Chat System
press disconnect button or close the program		
	sends goodbye messages to other users	
	chat system exits	
Alternative path	indicates that local user is not disconnected because the username is sending/receiving a file/message which isn’t completely received/sent	

Use Case Id	2-b
Version	1.0
Name	disconnect-another
Actors	remote chat system
Description	This use case describes the disconnection to the chat system of a another user and the update of the local user list.

Purpose/Overview	This use case is aimed at updating the local user list when another is disconnecting to the system.	
Post-conditions	the other user is disconnected, his program closes and the local list of connected users is updated	
Business rules	- when a user gets disconnected, all the other users connected need to be informed about it. A bye message containing the local user name is used.	
Author and date	Jauffret & Gourraud 09/2014	
Basic course of events		
Local User	ChatSystem	Remote Chat System
		sends bye messages to other users
	update list	

3) Scenarios for the “Send” use case

Two scenarios are possible :

a) The user sends message to selected user(s) connected to the chat system : “send-message”

b) The user sends file to selected user(s) connected to the chat system : “send-file”

Use Case Id	3-a
Version	1.0
Name	send-message
Actors	local user
Description	This use case describes how a user, through to chat system, sends a message to user(s) of his choice.

Purpose/Overview	This use case is aimed at allowing the local user to select the user(s) he wants to communicate with and then sends him/them a message. The user selects the user(s) from the local connected users list. The user selects the user(s) to communicate with, thanks to checkboxes in front of each person in the connected user list. Then the user informs the text message in the specified box, and then when the user clicks on the “Send message” button, the chat system delivers the message.	
Triggers	send button, checkboxes (one for each connected user)	
Preconditions	- a text message and at least one receiver has been provided - the user is connected	
Post-conditions	the selected user(s) receive(s) the message	

Business rules	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.	
Notes	The message is sent through (several/unique) unicast UDP	
Author and date	Jauffret & Gourraud 09/2014	
Basic course of events		
Local User	ChatSystem	
select the user(s) with checkboxes		
provide message		
press “send message button” button		
	delivers message	
Alternative path		
	if there are problems, indicates that the message hasn’t been sent	

Use Case Id	3-b
Version	1.0
Name	send-file
Actors	local user & device
Description	This use case describes how a user, through to chat system, sends a file to user(s) of his choice.

Purpose/Overview	This use case is aimed at allowing the local user to select the user(s) he wants, and sends him/them a file. The user selects the user(s) from the local connected users list. The user selects the user(s) to communicate with, thanks to checkboxes in front of each person in the connected user list. Then the user informs the file path, and then the device (filesystem) provides the required file to the chat system. when the user clicks on the “Send file” button, the chat system delivers the message.	
Triggers	send button, checkboxes (one for each connected user)	
Preconditions	- a valid file and at least one receiver has been provided - the user is connected	
Post-conditions	the selected user(s) receive(s) the file	
Business rules	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.	
Notes	The message is sent through TCP	

Author and date	Jauffret & Gourraud 09/2014	
Basic course of events		
Local User	ChatSystem	Device
select the user(s) to communicate with checkboxes		
provide filepath		
		provide file selected
press “send file button” button		
	open connexion	
	deliver file	
	close connexion	

4) Scenarios for the “Receive” use case

Two scenarios are possible :

c) The user receives a message from a user connected to the chat system : “receive-message”

d) The user receives a file from a user connected to the chat system : “receive-file”

Use Case Id	4-a
Version	1.0
Name	receive-message
Actors	remote chat system
Description	This use case describes how a user, through to chat system, receives a message.

Purpose/Overview	When the user receives a message, another window is opened, and shows him the message.	
Triggers		
Preconditions	- an user is sending a message to him - the user is connected	
Post-conditions	The user can see the message and the sender receives the notification that the message has been received	
Business rules	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).	
Notes	MessageAck is sent through UDP unicast	

Author and date	Jauffret & Gourraud 09/2014	
Basic course of events		
	ChatSystem	Remote Chat System
		Send a message
	deliver ackRead message	
	Notifies the user : open a window with the message	

Use Case Id	4-b
Version	1.0
Name	receive-file
Actors	remote chat system
Description	This use case describes how a user, through to chat system, receives a file.

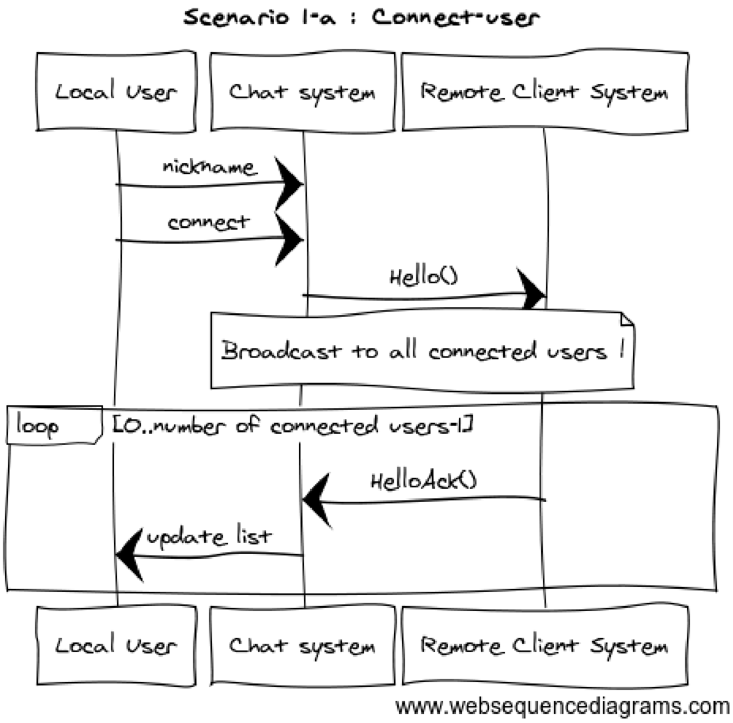
Purpose/Overview	When the user receives a file, another window is opened, and shows him the downloaded file.	
Triggers		

Preconditions	- an user is sending a file to him - the user is connected	
Post-conditions	The user has the file on the filesystem	
Business rules	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).	
Notes	File is sent through TCP : no ack required. We will think about security about file downloading later...	
Author and date	Jauffret & Gourraud 09/2014	
Basic course of events		
	ChatSystem	Remote Chat System
		open connexion
		delivers file
	stores the received file	
		close connexion
	Notifies the user : open a window	

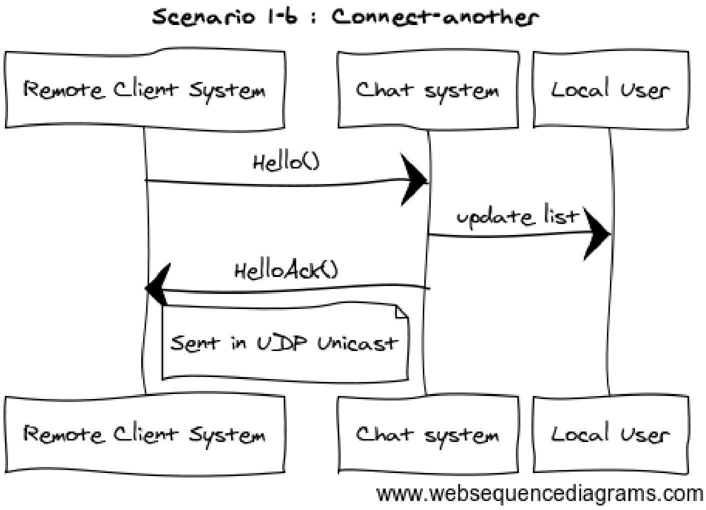
Sequence diagrams

For each scenario, we can draw a UML sequence diagram model.

-------------------------------- CONNECT CASE --------------------------------

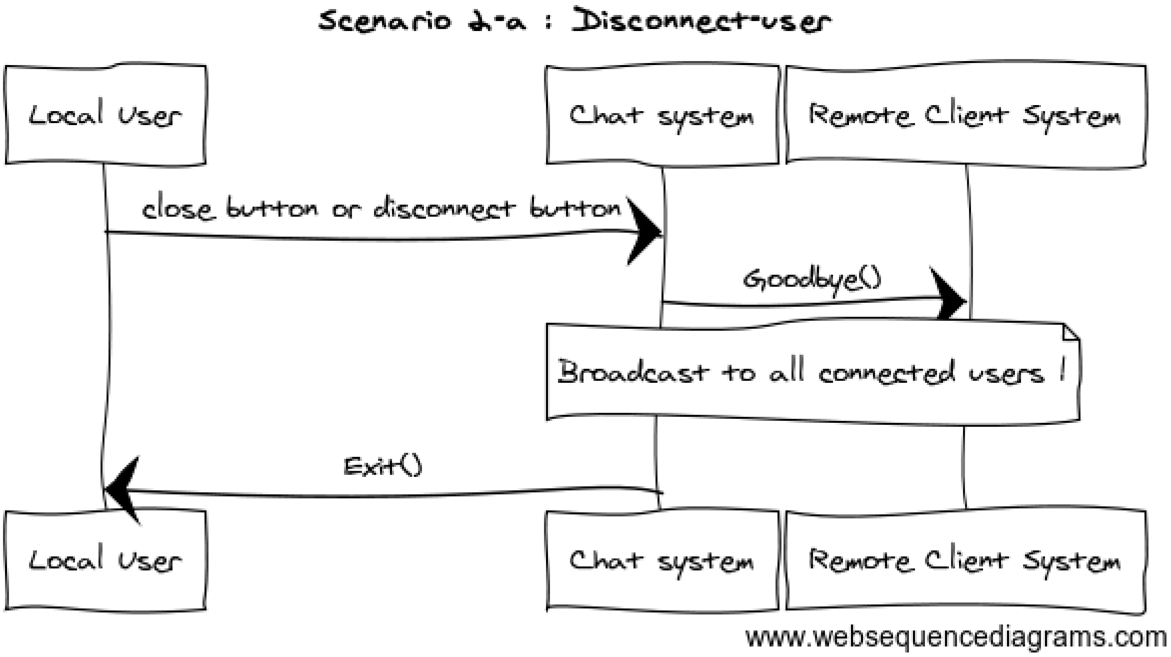

Scenario 1-a : Connect-user
Local User
Chat system
Remote Client System
nickname
connect
Hello()
Broadcast to all connected users !
loop
[0..number of connected users-1]
HelloAck()
update list
www.websequencediagrams.com

(Available here)

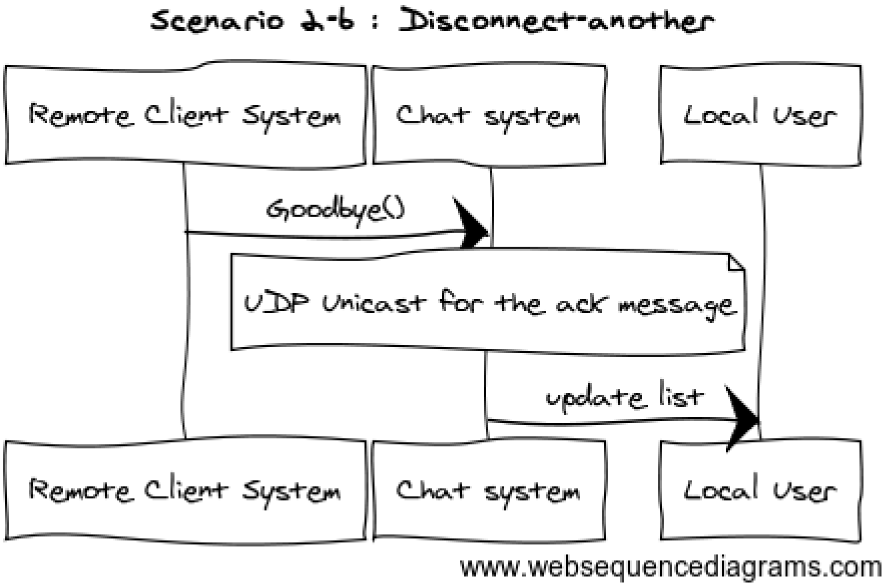


(Available Here)

---------------------------- DISCONNECT CASE ----------------------------

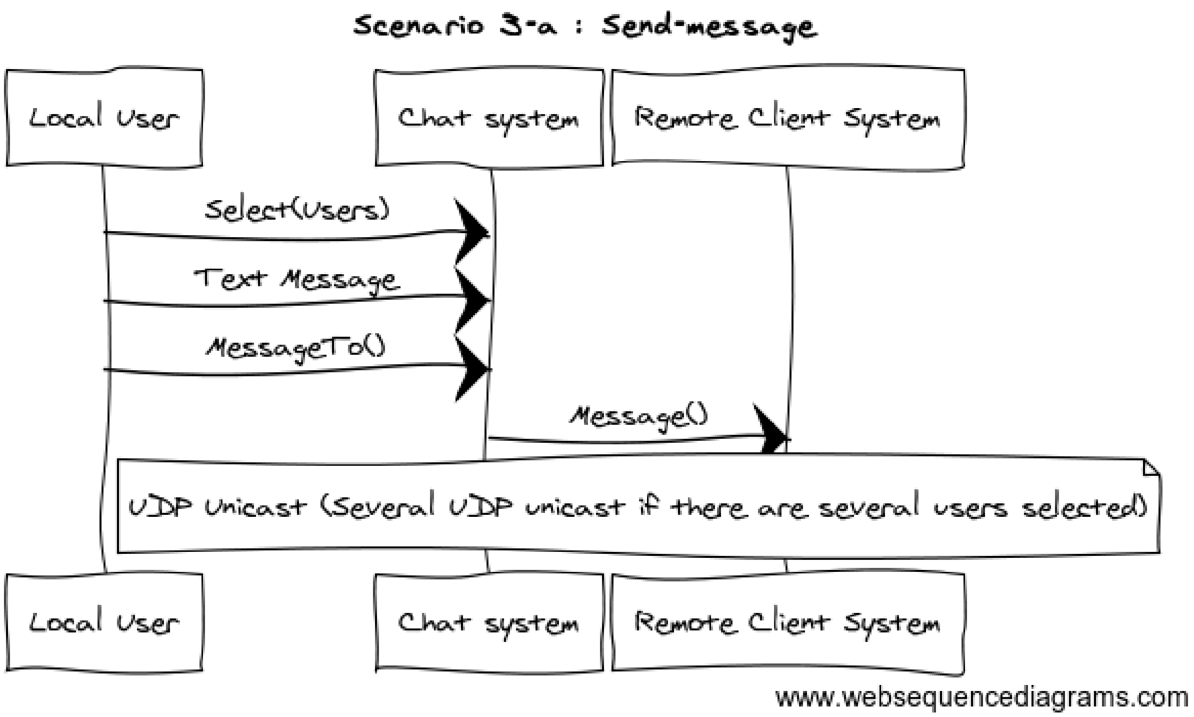

Scenario 2-a : Disconnect-user
Local User
Chat system
Remote Client System
close button or disconnect button
Goodbye()
Broadcast to all connected users !
Exit()
www.websequencediagrams.com

(Available here)

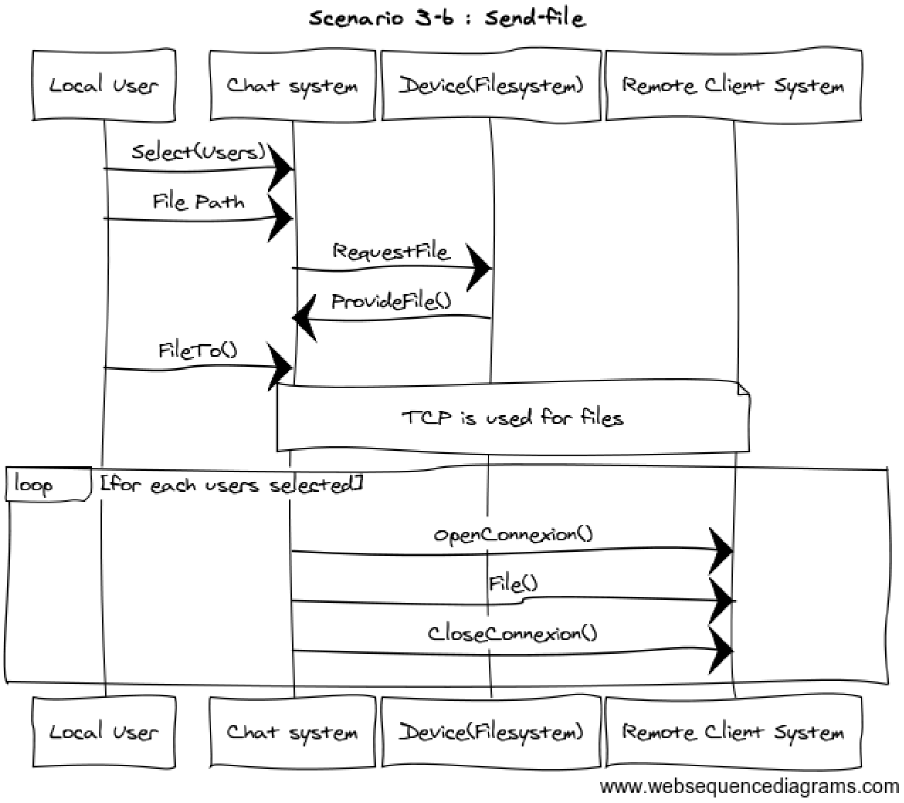

Scenario 2-b : Disconnect-another
Remote Client System
Chat system
Local User
Goodbye()
UDP Unicast for the ack message
update list
www.websequencediagrams.com

(Available here)

-------------------------------- SEND CASE ---



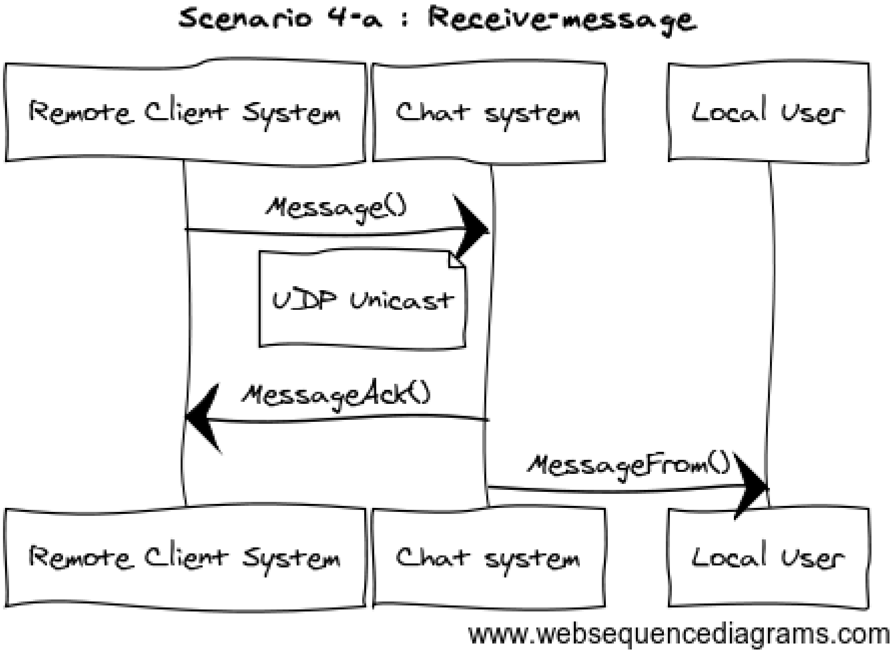
(Available here)

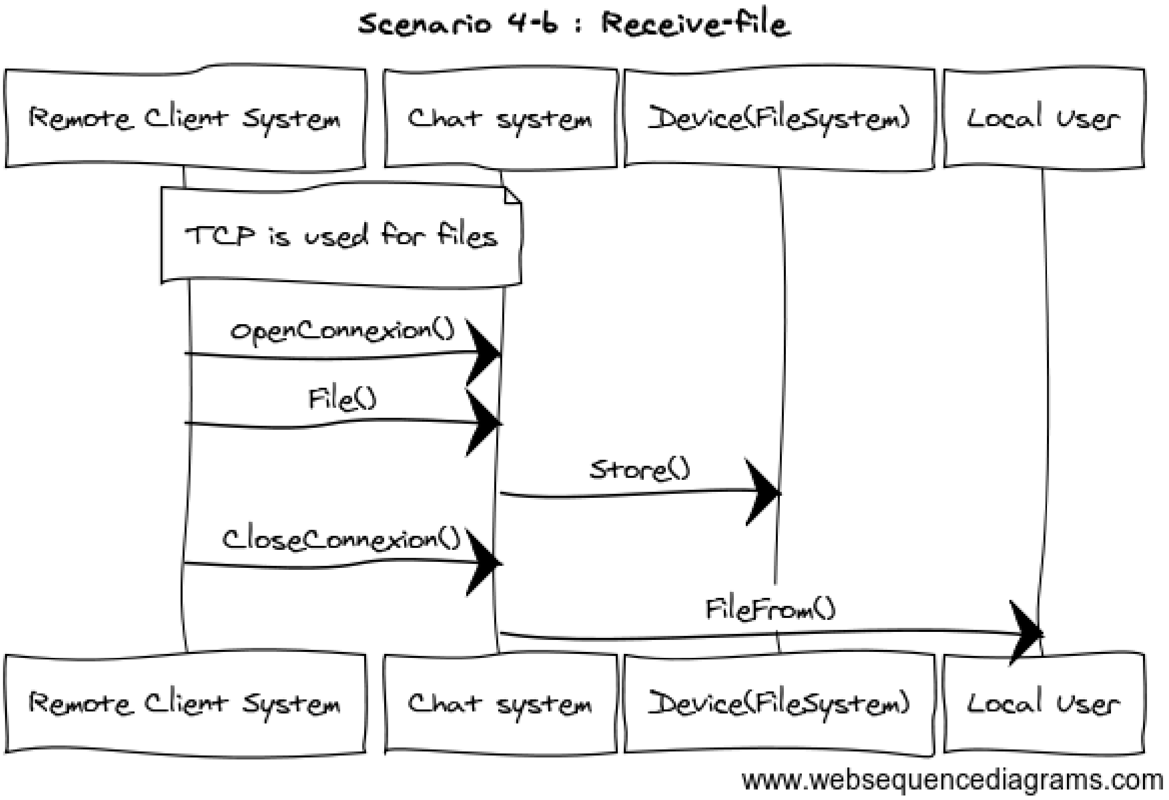


(Available here)

-------------------------------- RECEIVE CASE --------------------------------

(Available here)





(Available here)

Class diagram

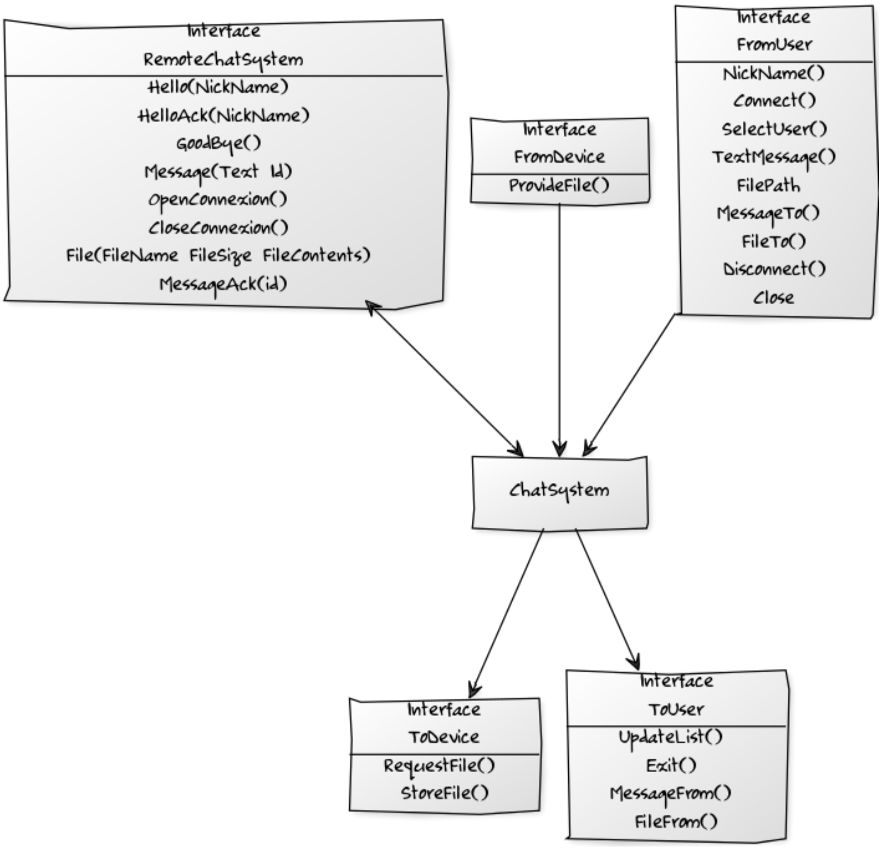
We decided together to set names of messages between Remote chat System and Chat System, in order to test our own softwares together :

- Hello() [Nickname] / Goodbye()
  will use broadcast (UDP)

- HelloAck() [Nickname] / Message() [Text,id] / MessageAck() [id]
  will use unicast (UDP)

- / File() [Filename.Extension,FileSize, FileContents]
  will use TCP (OpenConnexion()/CloseConnexion())

  We always use port number 1337.


Interface
RemoteChatSystem
Hello(NickName)
HelloAck(NickName)
GoodBye()
Message(Text Id)
OpenConnexion()
CloseConnexion()
File(FileName FileSize FileContents)
MessageAck(id)
Interface
FromDevice
ProvideFile()
Interface
FromUser
NickName()
Connect()
SelectUser()
TextMessage()
FilePath
MessageTo()
FileTo()
Disconnect()
Close
ChatSystem
Interface
ToDevice
RequestFile()
StoreFile()
Interface
ToUser
UpdateList()
Exit()
MessageFrom()
FileFrom()