

SRS

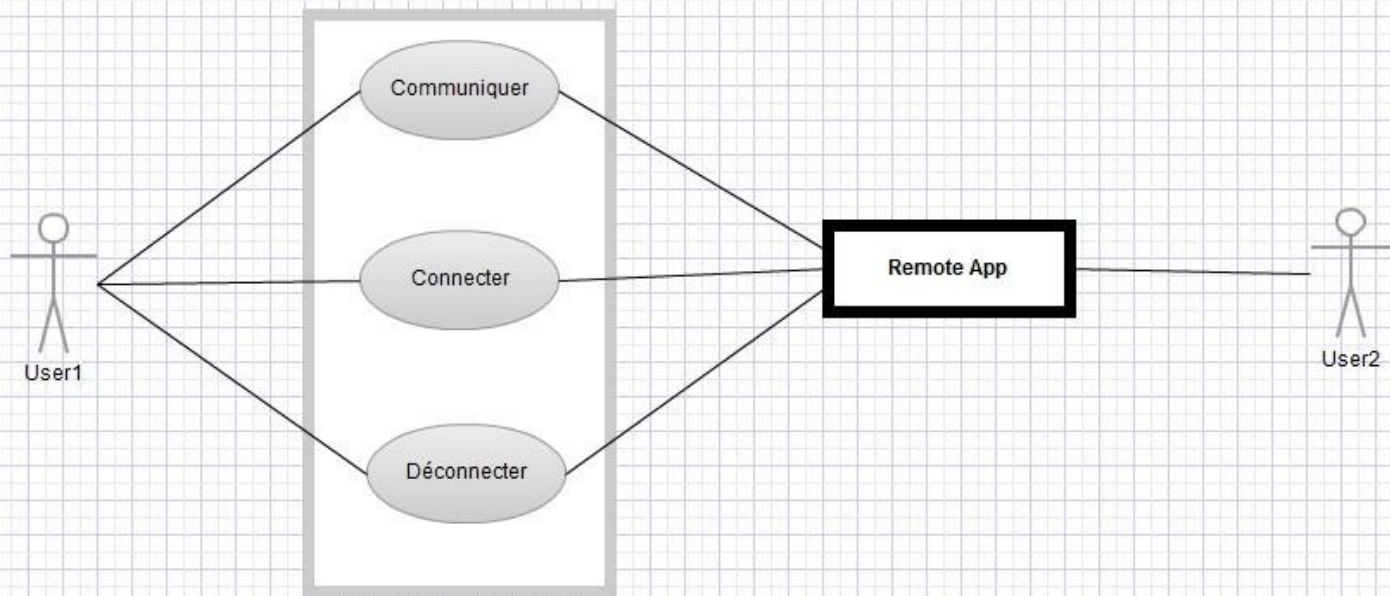
ChatSystem V1.0

Auteur	Version	Date	Modifications
Yingqing & Sofiene	1.0	14/12/2012	

Sommaire

1. Use case.....	3
2. Scénario	3
2.1Connecter	3
2.2Connecter	4
2.3Déconnecter	5
2.4Déconnecter	7
2.5Communiquer.....	8
2.6Communiquer.....	9
2.7communiquer	11
2.8 Communiquer.....	12
3. Diagramme de séquences	14
3.1Connexion.....	14
3.2Connexion.....	15
3.3 Déconnexion.....	15
3.4 Déconnexion :.....	16
3.5 Communiquer.....	17
4. Signaux :.....	20
5. Diagramme de classe.....	20
6. Maquette.....	22

1. Use case



2. Scénario

2.1 Connecter

Use Case Id	1
Version	1.0
Name	Connexion de l'utilisateur local
Actors	Utilisateur, Chat System, Remote System
Description	Cet Use Case décrit la connexion au système par l'utilisateur local.

Purpose/Overview	Cet Use Case a pour but de permettre à l'utilisateur de se connecter au système. L'utilisateur appuie sur le bouton de connexion après avoir saisi son login (son nom d'utilisateur). Le système informe l'utilisateur si la connexion est établie ou pas.
Triggers	Bouton de connexion
Preconditions	-Un nom d'utilisateur valide (disponible) -L'utilisateur est déconnecté

Post-conditions	L'utilisateur est connecté
Business rules	<ul style="list-style-type: none"> - Le nom d'utilisateur est un identifiant unique pour l'utilisateur - Quand un utilisateur est connecté, tous les autres utilisateurs qui sont connecté doivent être informés (message en broadcast).
Notes	<p>Plusieurs possibilités d'offrir un identifiant unique existent:</p> <ul style="list-style-type: none"> - Le login est comparé avec les autres utilisateurs pour vérifier s'il est valide. - Le login est automatiquement converti en un login valide en ajoutant l'adresse de host qui est unique.
Author and date	Sofiene et Yingqing, 26.9.2012
Basic course of events	
Utilisateur local	Chat System
Rentrer son nom d'utilisateur	
Appuyer sur le bouton "Connecter"	
	Envoyer le message "hello" à remote system
	Informé aux autres utilisateurs que l'utilisateur local est connecté
Alternative path	Indiquer que cet utilisateur n'a pas réussi à se connecter car il n'a pas renseigné son login.

2.2Connecter

Use Case Id	2
Version	1.0
Name	Connexion de l'utilisateur distant
Actors	Utilisateur, Remote System
Description	Cet Use Case décrit la connexion au système par l'utilisateur distant.

Purpose/Overview	Cet Use Case a pour but de permettre au remote system de pouvoir communiquer avec l'utilisateur local.
Triggers	Réception de message "hello"
Preconditions	-Un nom d'utilisateur valide (disponible) -L'utilisateur est déconnecté
Post-conditions	L'utilisateur distant communique avec l'utilisateur local.
Business rules	- Le nom d'utilisateur est un identifiant unique pour l'utilisateur -Quand un utilisateur est connecté, tous les autres utilisateurs qui sont connecté doivent être informés (message en broadcast).
Notes	Plusieurs possibilités d'offrir un identifiant unique existent: - Le login est comparé avec les autres utilisateurs pour vérifier s'il est valide. - Le login est automatiquement converti en un login valide en ajoutant l'adresse de host qui est unique.
Author and date	Sofiène et Yingqing, 26.9.2012
Basic course of events	
Local Distant	Chat System
	Recevoir le message "hello" provenant de l'utilisateur distant
	Informé aux autres utilisateurs que l'utilisateur local est connecté
	Ajouter le nouvel utilisateur à la liste
Alternative path	Indiquer que cet utilisateur n'a pas réussi à se connecter car son nom d'utilisateur n'est pas valide.

2.3 Déconnecter

Use Case Id	3
Version	1.0

Name	Déconnexion de l'utilisateur local
Actors	Utilisateur, Chat system, Remote system
Description	Ce Use Case décrit la déconnexion du système par l'utilisateur local.

Purpose/Overview	Cet Use Case a pour but de permettre au Remote System de terminer la connexion et la communication avec l'utilisateur local.
Triggers	Réception de message "byebye"
Preconditions	-L'utilisateur est connecté
Post-conditions	L'utilisateur est déconnecté
Business rules	-Quand un utilisateur est déconnecté, tous les autres utilisateurs qui sont connecté doivent être informés (message en broadcast).
Notes	- En cas d'interruption du programme, les autres utilisateurs ne sont pas avertis de la déconnexion, une requête est alors envoyé toutes les 15mn pour mettre à jour la liste des utilisateurs connectés.
Author and date	Sofiene et Yingqing, 26.9.2012
Basic course of events	
Utilisateur local	Chat System
Appuyer sur le bouton "Déconnecter"	
	Envoyer un message "byebye" aux Remote system et le système affiche de nouveau une page de connexion.
	Informé aux autres utilisateurs que l'utilisateur local est déconnecté.

2.4 Déconnecter

Use Case Id	4
Version	1.0
Name	Déconnexion de l'utilisateur distant
Actors	Utilisateur, Chat system, Remote system
Description	Ce Use Case décrit la déconnexion du système par l'utilisateur distant.

Purpose/Overview	Ce Use Case a pour but de permettre à l'utilisateur distant de se déconnecter du système. L'utilisateur appuie sur le bouton de déconnexion. Une fois déconnecté, le système affiche de nouveau une page de connexion.
Triggers	Bouton de déconnexion
Preconditions	-L'utilisateur est connecté
Post-conditions	L'utilisateur est déconnecté
Business rules	-Quand un utilisateur est déconnecté, tous les autres utilisateurs qui sont connecté doivent être informés (message en broadcast).
Notes	- En cas d'interruption du programme, les autres utilisateurs ne sont pas avertis de la déconnexion, une requête est alors envoyée toutes les 15mn pour mettre à jour la liste des utilisateurs connectés.
Author and date	Sofiène et Yingqing, 26.9.2012
Basic course of events	
Utilisateur local	Chat System
	Réception de message "byebye" en provenant de l'utilisateur distant
	Envoyer un message "byebye" aux Remote system
	Informé aux autres utilisateurs que l'utilisateur local est déconnecté.

	Mettre à jour la liste d'utilisateur
--	--------------------------------------

2.5 Communiquer

Use Case Id	5
Version	1.0
Name	Envoi des messages au Remote System
Actors	Utilisateur, Chat system, Remote system
Description	Cet Use Case décrit l'envoi des messages (message pur) au Remote System.

Purpose/Overview	Ce Use Case a pour but de permettre à l'utilisateur de envoyer des messages au Remote System. Tout d'abord, l'utilisateur sélectionne un autre utilisateur ou un groupe d'utilisateurs connecté(s) de sa liste en appuyant sur son/leurs login(s). En suite, l'utilisateur écrit un message et appuie sur le bouton d'envoi pour envoyer le message. Le message est alors affiché dans le chat.
Triggers	Bouton d'envoi (Envoyer)
Preconditions	-L'utilisateur est connecté -L'utilisateur distant sélectionné doit être connecté
Post-conditions	Messages envoyés
Business rules	Un message vide ne peut être envoyé.
Notes	
Author and date	Sofiène et Yingqing, 26.9.2012
Basic course of events	

Utilisateur local	Chat System
Cliquer le nom d'utilisateur ou un groupe d'utilisateur qu'il souhaite communiquer	
Tapper les textes que l'utilisateur veut envoyer	
Appuyer sur le bouton "envoyer"	
	Envoyer ces messages au Remote system
	Indiquer à l'utilisateur local que les messages est bien reçus

2.6 Communiquer

Use Case Id	6
Version	1.0
Name	Envoi des messages/fichiers au Remote System
Actors	Utilisateur, Chat system, Remote system
Description	Cet Use Case décrit l'envoi des messages avec les fichiers au Remote System.

Purpose/Overview	Cet Use Case a pour but de permettre à l'utilisateur d'envoyer des messages au Remote System. Tout d'abord, l'utilisateur sélectionne un autre utilisateur ou un groupe d'utilisateurs connecté(s) de sa liste en appuyant sur son/leurs login(s). En suite, L'utilisateur appuie sur le bouton "Browse" pour parcourir un fichier (non exécutable) et appuie sur le bouton envoyer pour transmettre le fichier au remote system. Le fichier peut être accompagné d'un message. Le fichier ne peut être envoyé que si l'utilisateur distant accepte de le recevoir. L'utilisateur local envoie donc dans un premier temps une requête pour demander l'acceptation de l'utilisateur distant. En cas d'acceptation, le message commence à être envoyé (suivi de l'envoi dans le chat), en cas de refus (un message de refus dans le chat).
Triggers	Bouton d'envoi (Envoyer)

	Bouton pour joindre un fichier (Browse)
Preconditions	-L'utilisateur est connecté -L'utilisateur distant sélectionné doit être connecté
Post-conditions	Messages/Fichiers envoyés
Business rules	- Une pièce jointe peut être envoyée sans qu'il y ait un besoin d'envoyer un message avec. - L'utilisateur émetteur ne peut envoyer de fichiers que s'il a fini d'envoyer le dernier. Les envois en parallèle de fichiers sont interdits.
Notes	
Author and date	Sofiene et Yingqing, 26.9.2012
Basic course of events	
Utilisateur local	Chat System
Cliquer le nom d'utilisateur ou un groupe d'utilisateur qu'il souhaite communiquer	
Tapper les textes que l'utilisateur veut envoyer	
Cliquer sur le bouton "Browse" pour parcourir le fichier que l'utilisateur veut envoyer	
Sélectionner le fichier	
Appuyer sur le bouton "envoyer"	
	Envoyer ces messages / fichiers au Remote system
	Indiquer à l'utilisateur local que les messages/fichiers sont bien reçus

2.7communiquer

Use Case Id	7
Version	1.0
Name	Réception des messages
Actors	Utilisateur, Chat system, Remote system
Description	Cet Use Case décrit la réception des messages (message pur) vient des utilisateurs distants

Purpose/Overview	L'utilisateur peut recevoir d'un message vient des autres utilisateurs et les affichages.
Triggers	La réception de message (L'interface de communication clignote)
Preconditions	-L'utilisateur est connecté -L'utilisateur distant sélectionné doit être connecté
Post-conditions	Messages reçus
Business rules	
Notes	
Author and date	Sofiène et Yingqing, 26.9.2012
Basic course of events	
Utilisateur local	Chat System
	Recevoir un message vient des autres utilisateurs
	Afficher les messages sur l'interface de communication
Cliquer l'interface et lire les messages	

2.8 Communiquer

Use Case Id	8
Version	1.0
Name	Réception des messages/fichiers
Actors	Utilisateur, Chat system, Remote system
Description	Cet Use Case décrit la réception des messages/fichiers vient des utilisateurs distants

Purpose/Overview	L'utilisateur reçoit un message pour l'acceptation de fichier qui est envoyé par les autres utilisateurs distants. 2 boutons: accepter, refuser permettre au utilisateur de choisir s'il veut l'accepte. S'il clique le bouton accepter, un autre message affiche pour demande s'il veut enregistrer le fichier ou l'ouvert. En cas de refusé, un message de refusions va envoyer automatiquement au émetteur.
Triggers	-La réception de message /fichiers (L'interface de communication clignote) -Bouton "Accepter" -Bouton "Refuser" -Bouton "enregistre" -Bouton "ouvrir "
Preconditions	-L'utilisateur est connecté -L'utilisateur distant sélectionné doit être connecté -Le fichier ne doit pas être exécutable
Post-conditions	Messages/fichiers reçus
Business rules	Si l'utilisateur n'est pas cliqué le bouton "accepter" ni "refuser", le system attend.
Notes	
Author and date	Sofiene et Yingqing, 26.9.2012
Basic course of events	
Utilisateur local	Chat System
	Recevoir des messages/fichiers viennent des autres utilisateurs
	Afficher les messages sur l'interface de communication

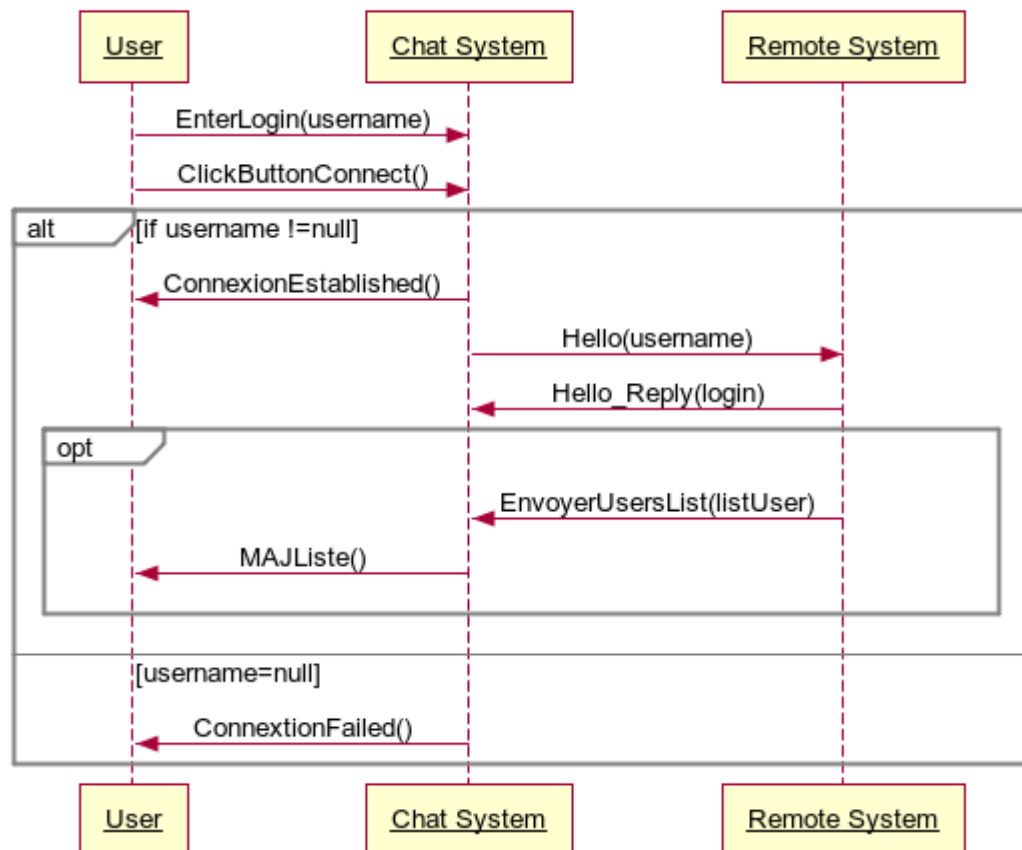
	Afficher la demande d'acceptation, il affiche sur l'écran deux boutons : Accepter et Refuser
Si l'utilisateur clique sur le bouton "Accepter"	
	Afficher demande à l'utilisateur s'il souhaite enregistrer le fichier ou juste l'ouvrir
Si l'utilisateur clique sur le bouton "Refuser"	
	Envoyer un message de refusion au Remote System

3. Diagramme de séquences

3.1 Connexion

La séquence qui suit est assez simple. Nous sommes déconnectés et nous voulons nous connecter. Pour faire cela nous envoyons un signal Hello () en diffusion ayant comme paramètre un Username. En suite, nous recevrons un signal HelloReply () des utilisateurs qui sont déjà connectés. Ici, il y a un contrôle de Username, si l'utilisateur n'a pas saisi son Username, un message d'erreur va afficher en indiquant que l'Username ne doit pas être null.

Diagram Sequence : Connexion de l'utilisateur local

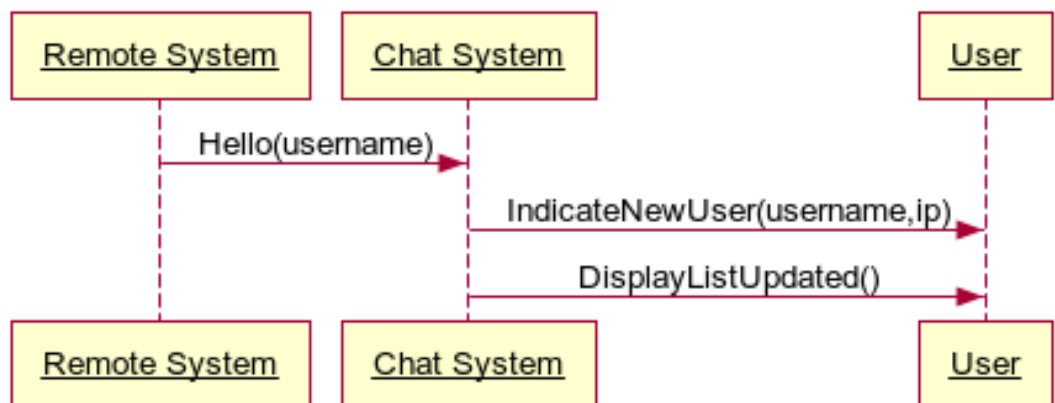


www.websequencediagrams.com

3.2 Connexion

Cette séquence est la réception d'une connexion d'un Remote User. Pour cela, il faut être d'abord connecté. Si nous le ne sommes pas, nous ignorons le signal Hello (). Si nous sommes connectés, nous récupérons l'adresse Ip de l'émetteur. Nous mettons cette adresse dans une table avec l'Username correspondant. Si l'adresse Ip est déjà dans la table nous remplaçons l'Username lié à l'adresse Ip. Si l'adresse Ip n'est pas dans la table, nous l'ajoutons avec l'Username.

Diagram Sequence : Connexion de l'utilisateur distant

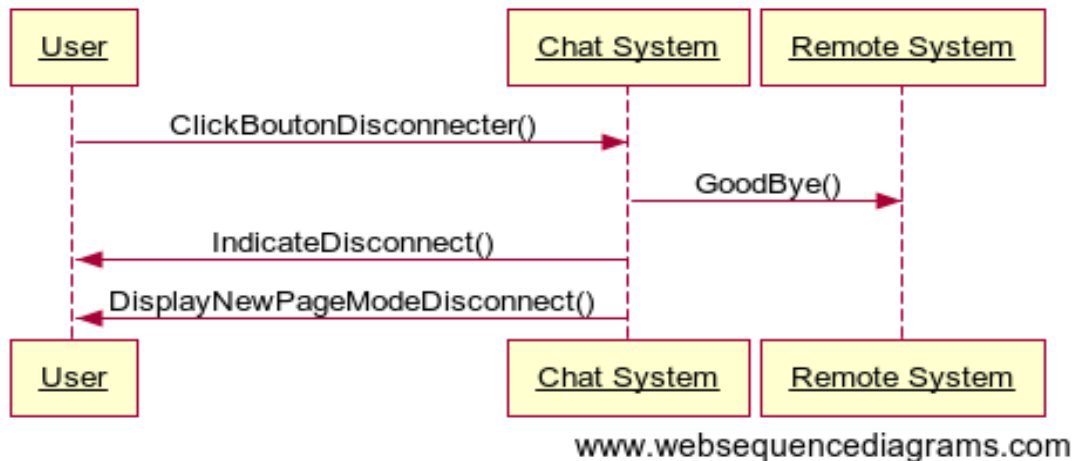


www.websequencediagrams.com

3.3 Déconnexion

Si nous voulons nous déconnecter, il faut être connecté et il suffit d'envoyer un signal GoodBye () en diffusion.

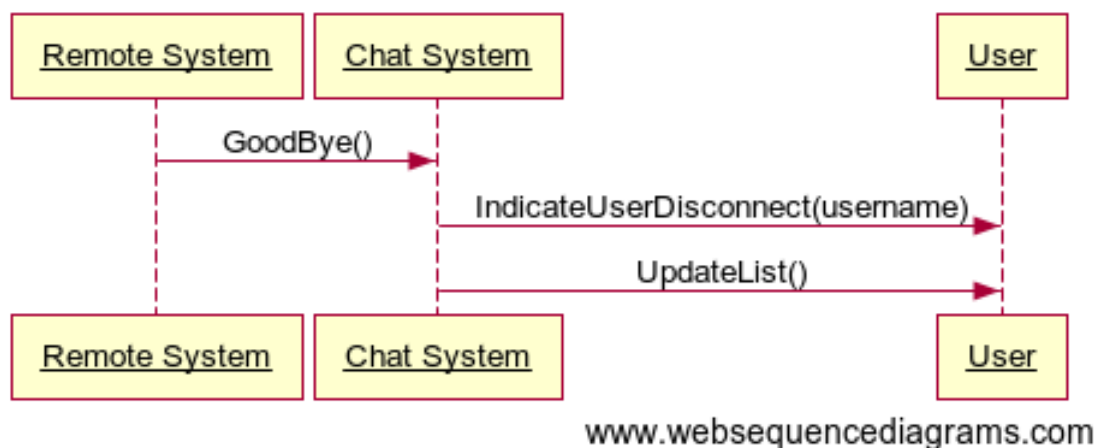
Diagram Sequence : Déconnexion de l'utilisateur local



3.4 Déconnexion :

Quand nous recevons un signal GoodBye (), tout en étant connecté, nous identifions l'émetteur du signal par son adresse Ip et nous le supprimons dans la liste des connectés ayant l'adresse Ip de l'émetteur. Si des échanges de données sont en cours avec l'émetteur nous les coupons immédiatement.

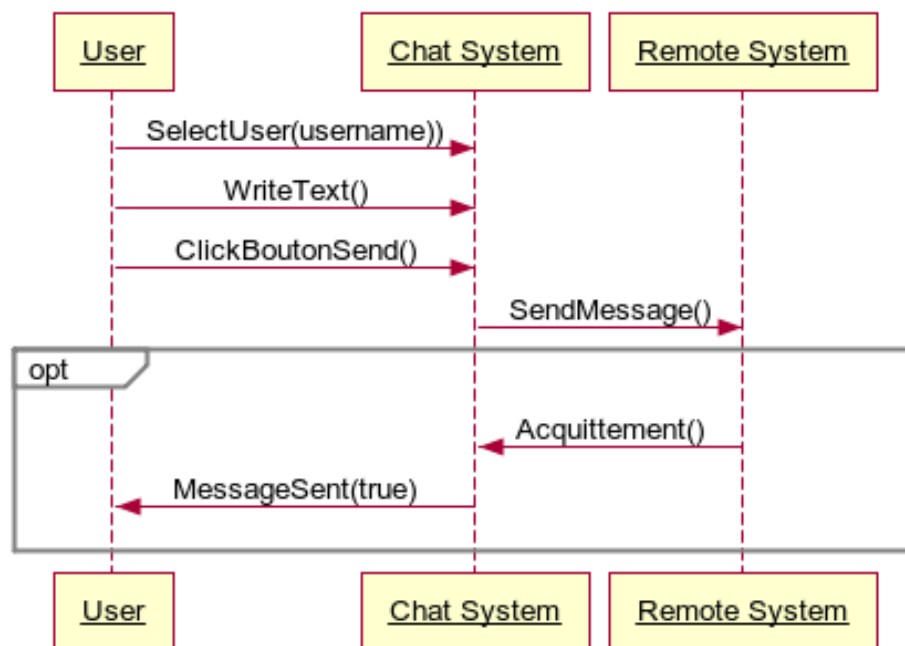
Diagram Sequence : Déconnexion de l'utilisateur distant



3.5 Communiquer

Pour envoyer un message texte à un ou plusieurs utilisateurs. Nous envoyons un signal `SendText()` à chaque utilisateur sélectionné. Pour ce faire, l'utilisateur local doit être connecté au préalable.

Diagram Sequence : Envoi des messages au Remote System

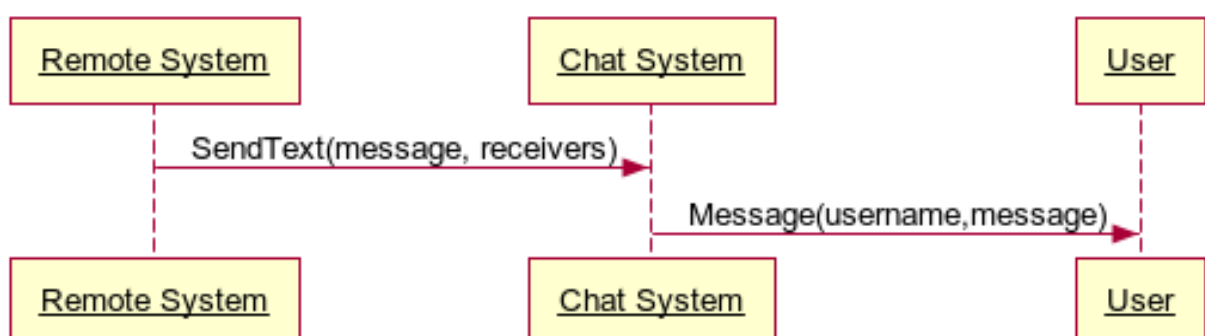


www.websequencediagrams.com

3.6 Communiquer

Pour recevoir un message texte, l'utilisateur local doit être connecté au préalable. Ensuite, l'émetteur du message est identifié par son adresse IP et le message est affiché dans l'onglet qui correspond à cet émetteur. Enfin, l'utilisateur local sélectionne l'onglet pour le visualiser.

Diagram Sequence : Réption des messages

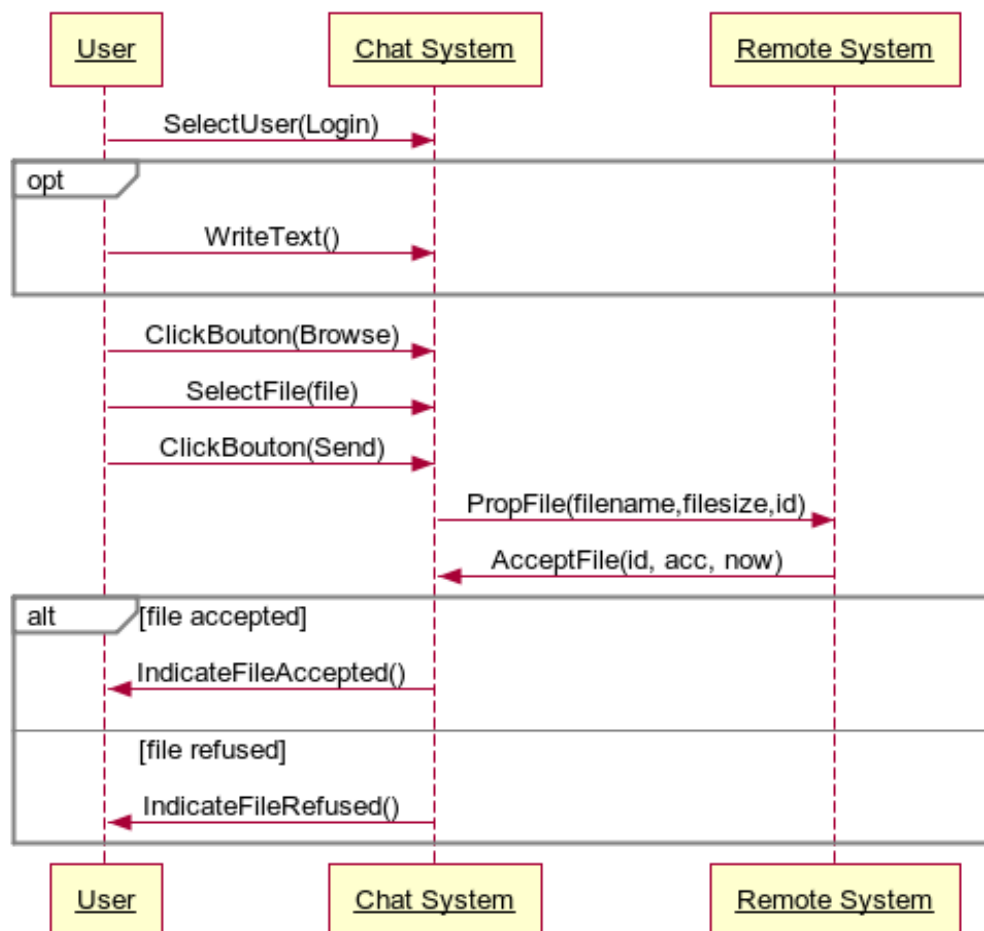


www.websequencediagrams.com

3.7 Communiquer

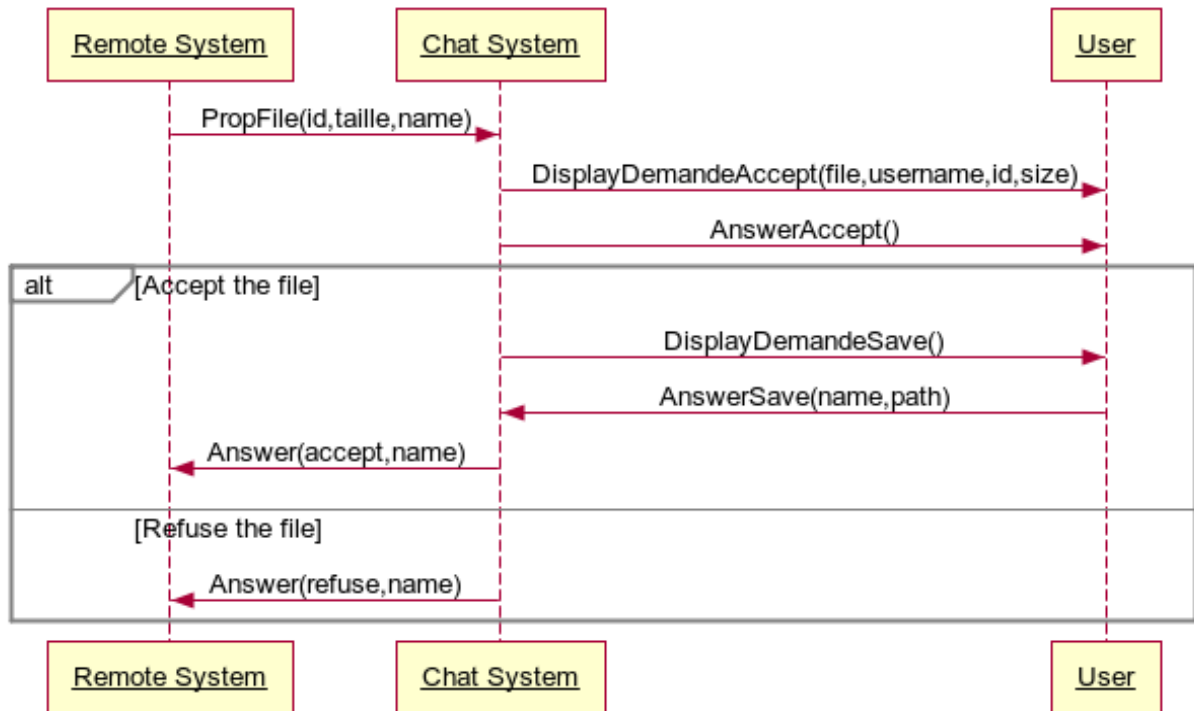
Pour envoyer un fichier, l'utilisateur local doit être connecté au préalable. Ensuite, l'utilisateur local sélectionne sur la liste des contacts l'utilisateur distant auquel il veut envoyer un fichier. Il appuie ensuite sur le bouton « Browse » pour parcourir son fichier. En le choisissant, un message de demande de réception de fichier est envoyé à l'utilisateur distant, qui à son tour, décide d'accepter ou de rejeter le fichier. En cas d'acceptation, il choisit l'endroit où l'enregistrer sur son disque dur et le téléchargement commence à être émis. Une fois le téléchargement fini, un message est affiché sur l'écran de l'utilisateur local.

Diagram Sequence : Envoi des fichiers au Remote System



3.8 Communiquer

Diagram Sequence : Réception des messages/fichiers

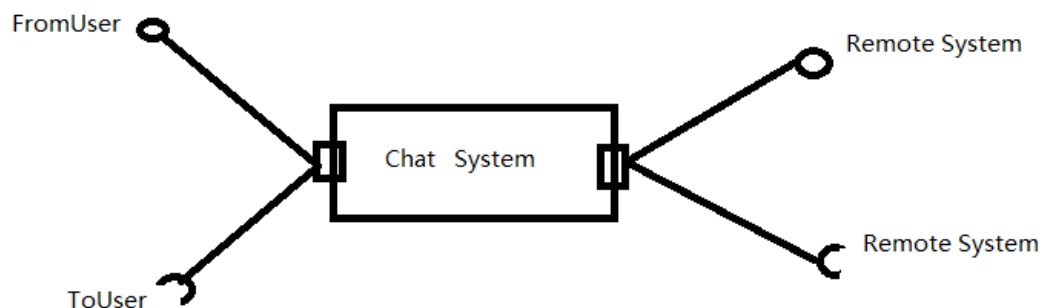


www.websequencediagrams.com

4. Signaux :

Signal	Types	Remarques
AcceptFile(fid,acc,now)	fid : Int acc : boolean now : boolean	
AskLogin()		Utilisation d'adresses IP
GoodBye()		Utilisation d'adresses IP
Hello(username)	Username: string	
HelloReply(login)	Login: string	
PropFile(filename,files,file_id)	Filename: string Files: long File_id: int	
SendText(message,receivers)	message: string receivers: string[]	

5. Diagramme de classe



<<Interface>> FromUser
+Signal EnterLogin(username) +Signal ClickBouttonConnect() +Signal Hello(username) +Signal ClickBouttonDisconnecter() +Signal GoodBye() +Signal SelectUser(username) +Signal WriteText() +Signal ClickBouttonSend() +Signal SendText(message, receivers) +Signal SelectUser(Login) +Signal ClickBouttonBrowse() +Signal SelectFile(file) +Signal PropFile(id, taille, name)

<<Interface>> Remote System
+Signal Hello(username) +Signal EnvoyerUserList(listUser) +Signal GoodBye() +Signal SendMessage() +Signal Acquittement() +Signal SendText(message, receivers) +Signal PropFile(filename, filesize, id) +Signal AcceptFile(id, acc, now) +Signal Answer(accept, name) +Signal Answer(refuse, name)

<<Interface>> ToUser
+Signal ConnexionEstablished() +Signal MAJListe() +Signal ConnexionFailed() +Signal IndicateNewUser(username, ip) +Signal DisplayListUpdated() +Signal IndicateDisconnect() +Signal DisplayNewPageModeDisconnected() +Signal IndicateUserDisconnect(username) +Signal UpdateLis() +Signal MessageSent(true) +Signal Message(username, message) +Signal IndicateFileAccept() +Signal IndicateFileRefused() +Signal DisplayDemandeAccept(file, username, id, size) +Signal AnswerAccept() +Signal DisplayDemandeSave()

6. Maquette

Grâce à Java swings, nous avons réalisé notre maquette ici dessous : l'interface contient deux JLabel (User et Connected users), Trois JTextArea qui permettent de saisir le username d'utilisateur pour se connecter, écrire les textes pour envoyer et afficher les textes provenant de l'utilisateur local et distant. Quatre boutons (Connect, Disconnect, Send, Browse) implémentent respectivement les ActionListener. Un JList permet d'afficher les utilisateurs qui sont connectés.

