

Logiciel « Chat »

Spécifications

Auteur	Contact
Matthias Brun	matthias.brun@eseo.fr

Relecteur	Contact
Camille Constant	camille.constant@eseo.fr

Testeurs	Contact
Younes Ghoniem	younes.ghoniem@imt-atlantique.net
Matthieu Schmitz	matthieu.schmitz@imt-atlantique.net

Version	Date	Commentaire
1.0	14/04/2023	Ajout Specifications ; Ajout TU ; Ajout Tests Utilisateurs.
0.6	14/02/2022	Modifications Déploiement ; Modifications IHM messages textuels.
0.5	22/01/2018	Modifications Domaine ; Modifications IHM messages du serveur → messages textuels.
0.4	19/02/2017	Ajout Domaine ; Modifications Introduction ; Modifications CU ; Modifications IHM.
0.3	13/01/2013	Ajout CU sortir du chat ; Modification CU lancement client chat → entrer dans le chat ; Modifications IHM : Messages du serveur.
0.2	09/01/2013	Ajout CU lancement client chat ; Ajout pré-conditions aux CU ; Modifications IHM : Messages du serveur.
0.1	08/01/2013	Introduction, Déploiement, CU, IHM.

Table des matières

1	Introduction	3
1.1	Objectif des spécifications	3
1.2	Portée des spécifications	3
1.3	Modification des spécifications	3
2	Déploiement	4
3	Domaine	4
4	Cas d'utilisation	5
4.1	Communiquer avec d'autres utilisateurs	6
4.2	Entrer dans le chat	8
4.3	Demander de l'aide	9
4.4	Échanger des messages	9
4.5	Changer de surnom	9
4.6	Demander des informations	10
4.6.1	Demander ses informations personnelles	10
4.6.2	Demander les informations sur les canaux	10
4.7	Gérer des canaux de discussion	11
4.7.1	Créer un canal de discussion	11
4.7.2	Supprimer un canal de discussion	12
4.7.3	Changer de canal de discussion	12
4.8	Sortir du chat	13
4.9	Saisir une commande invalide	13
5	Interface homme-machine	15
5.1	Interface homme-machine de Felix	15
5.2	Protocole de communication client/serveur	16
5.3	Messages textuels	16

1 Introduction

1.1 Objectif des spécifications

Ce document spécifie un ensemble logiciel composé d'un serveur et de clients chat. Plusieurs clients chat peuvent se connecter au serveur chat et échanger des messages textuels via ce serveur.

Ce document décrit en particulier :

- **le déploiement** prévu des composants serveur et clients chat ;
- **le domaine** d'application du chat ;
- **les cas d'utilisation** attendus pour cet ensemble logiciel ;
- **l'interface homme-machine** envisagée pour un client ainsi que le protocole de communication client/serveur et les messages renvoyés par le serveur.
- **Les spécifications clients** revues et validées par les clients puis appliquées et testées.

1.2 Portée des spécifications

Ce document concerne tous les intervenants du projet.

1.3 Modification des spécifications

Les spécifications pourront être soumises à des modifications. Toute nouvelle version sera signalée par mail à l'ensemble des intervenants du projet.

2 Déploiement

Le chat se compose de deux entités : les composants logiciels **Camix** et **Felix** (cf. figure 1), respectivement composant serveur et client du chat.

Les composants Camix et Felix sont programmés en Java. Leurs exécutions nécessitent une machine virtuelle Java (ou *Java Virtual Machine* - JVM) en version supérieure ou égale 8.

Les composants Felix et Camix communiquent via un réseau TCP/IP (*Transmission Control Protocol/Internet Protocol*).

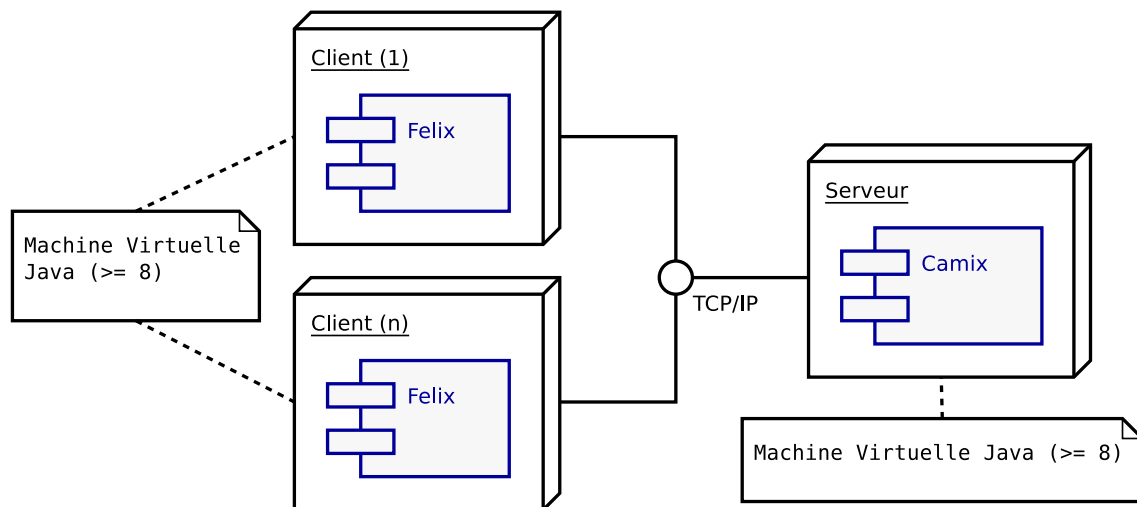


FIGURE 1 – Diagramme de déploiement du chat.

3 Domaine

Un chat est muni de canaux pouvant accueillir des utilisateurs (cf. figure 2). Les utilisateurs peuvent communiquer entre eux au sein d'un même canal. Un utilisateur est reconnu par les autres utilisateurs par son surnom. Le surnom par défaut est un point d'interrogation (?). Un canal est identifié par son nom.

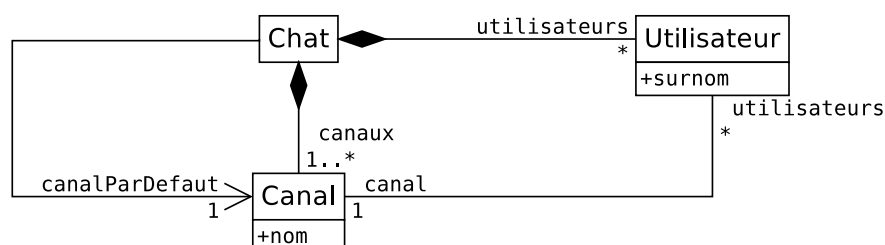


FIGURE 2 – Diagramme de classes du domaine du chat.

4 Cas d'utilisation

La figure 3 présente les cas d'utilisation du chat.

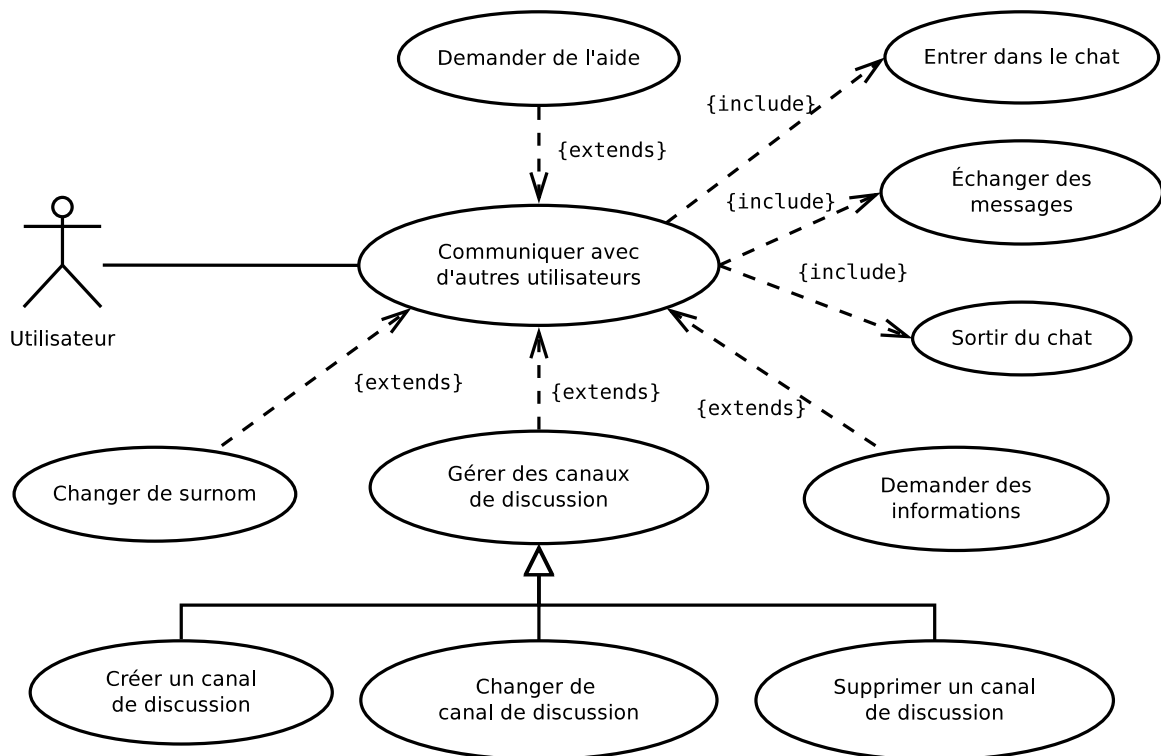


FIGURE 3 – Cas d'utilisation du chat.

Chacun de ces cas d'utilisation est détaillé dans la suite de cette section.

4.1 Communiquer avec d'autres utilisateurs

L'objectif du chat est de permettre à des utilisateurs de communiquer entre eux. Après être entré dans le chat, cette communication se fait par échanges de messages textuels. Chaque utilisateur est identifié par les autres utilisateurs via un surnom qu'il peut changer au cours de son utilisation du chat. Les échanges de messages se font au sein de canaux de discussion dont une gestion est possible par chaque utilisateur (création, suppression et changement de canal). Enfin, le chat fournit une aide aux utilisateurs ainsi que des informations sur les canaux existants et sur les utilisateurs eux-mêmes. La sortie du chat se fait en fermant le client chat (Felix).

Remarque (1) : Un serveur chat (Camix) compatible avec le client chat (Felix) doit être lancé pour permettre aux différents utilisateurs de communiquer.

Remarque (2) : Pour utiliser les différentes fonctionnalités du chat, un protocole est à

la disposition des utilisateurs. Ce protocole (spécifié dans la section 5.2, page 16) définit un ensemble de commandes que peut invoquer un utilisateur.

Cas d'utilisation : Communiquer avec d'autres utilisateurs.

Résumé : Un utilisateur lance un client chat pour entrer dans le chat, il peut ensuite échanger des messages, demander de l'aide sur les commandes du chat, changer de surnom, demander des informations, gérer des canaux de discussion et sortir du chat.

Niveau : Stratégique.

Acteur principal : Utilisateur.

Acteurs secondaires : Les autres utilisateurs du chat.

Pré-conditions : Un logiciel serveur (Camix) est lancé et accessible par le réseau TCP/IP.

Scénario nominal :

1. L'utilisateur entre dans le chat (cf. 4.2, page 8).
2. L'utilisateur échange des messages avec les autres utilisateurs du chat (cf. 4.4, page 9).
3. L'utilisateur sort du chat (cf. 4.8, page 13).

Variantes :

- 2.a L'utilisateur demande de l'aide (cf. 4.3, page 9).
- 2.b L'utilisateur change de surnom (cf. 4.5, page 9).
- 2.c L'utilisateur demande des informations (cf. 4.6, page 10).
- 2.d L'utilisateur gère des canaux de discussion (cf. 4.7, page 11).

Exception [Commande invalide] :

- 2.e L'utilisateur saisit une commande invalide (cf. 4.9, page 13).

4.2 Entrer dans le chat

Cas d'utilisation : Entrer dans le chat.

Résumé : Un utilisateur entre dans le chat en lançant le logiciel client du chat.

Acteur principal : Utilisateur.

Acteurs secondaires : Les autres utilisateurs du chat.

Pré-conditions : Un logiciel serveur (Camix) est lancé et accessible par le réseau TCP/IP.

Scénario nominal :

1. L'utilisateur lance l'exécution du composant Felix.
2. Felix initie la connexion à Camix.
3. Camix inscrit l'utilisateur dans le canal par défaut (place publique).
4. Camix informe les composants Felix des autres utilisateurs inscrits dans le canal par défaut que l'utilisateur arrive dans le chat.
5. Chaque composant Felix concerné affiche un message d'arrivée de l'utilisateur dans le chat.
6. Camix transmet au composant Felix de l'utilisateur un message d'accueil dans le chat.
7. Felix affiche un message d'accueil dans le chat.

4.3 Demander de l'aide

Cas d'utilisation : Demander de l'aide.

Résumé : Un utilisateur peut demander de l'aide sur les commandes du chat.

Acteur principal : Utilisateur.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur saisit la commande d'aide.
2. Felix transmet la commande d'aide à Camix.
3. Camix transmet à Felix les informations sur les commandes disponibles.
4. Felix affiche un message d'aide sur les commandes disponibles.

4.4 Échanger des messages

Cas d'utilisation : Échanger des messages.

Résumé : Un utilisateur échange des messages avec d'autres utilisateurs.

Acteur principal : Utilisateur.

Acteurs secondaires : Les autres utilisateurs du chat.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur saisit un message textuel.
2. Felix transmet le message à Camix.
3. Camix transmet le message aux composants Felix des utilisateurs inscrits dans le même canal que celui de l'utilisateur.
4. Chaque composant Felix concerné affiche le message.

4.5 Changer de surnom

Les utilisateurs peuvent avoir des surnoms pour être plus facilement identifiés dans le chat. Le surnom par défaut d'un utilisateur est un point d'interrogation (?).

Cas d'utilisation : Changer de surnom.

Résumé : Un utilisateur change de surnom.

Acteur principal : Utilisateur.

Acteurs secondaires : Les autres utilisateurs du chat.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur saisit la commande de changement de surnom.
2. Felix transmet la commande de changement de surnom à Camix.

3. Camix change le surnom de l'utilisateur.
4. Camix informe les composants Felix des utilisateurs inscrits dans le même canal que celui de l'utilisateur que ce dernier a changé de surnom.
5. Chaque composant Felix concerné affiche un message de changement de surnom de l'utilisateur.

4.6 Demander des informations

Le serveur chat peut fournir à chaque utilisateur des informations personnelles (surnom de l'utilisateur et canal occupé) et sur les canaux du chat (noms et nombres d'utilisateurs).

Cas d'utilisation : Demander des informations.

Résumé : Un utilisateur peut faire afficher ses informations personnelles ou des informations sur les canaux du chat.

Acteur principal : Utilisateur.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur demande ses informations personnelles (cf. 4.6.1, page 10).

Variante :

- 1.a L'utilisateur demande les informations sur les canaux (cf. 4.6.2, page 10).

4.6.1 Demander ses informations personnelles

Cas d'utilisation : Demander des informations personnelles.

Résumé : Un utilisateur peut obtenir ses informations personnelles (surnom et canal occupé).

Acteur principal : Utilisateur.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur saisit la commande d'informations personnelles.
2. Felix transmet la commande d'informations personnelles à Camix.
3. Camix transmet à Felix les informations sur l'utilisateur (surnom et canal occupé).
4. Felix affiche les informations personnelles de l'utilisateur.

4.6.2 Demander les informations sur les canaux

Cas d'utilisation : Demander des informations sur les canaux.

Résumé : Un utilisateur peut obtenir des informations sur les canaux de discussion (noms et nombres d'utilisateurs).

Acteur principal : Utilisateur.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur saisit la commande d'informations sur les canaux.
2. Felix transmet la commande d'informations sur les canaux à Camix.
3. Camix transmet à Felix les informations sur les canaux (noms et nombres d'utilisateurs).
4. Felix affiche les informations sur les canaux.

4.7 Gérer des canaux de discussion

Le serveur chat offre la possibilité d'échanger des messages dans des canaux de discussion. Les échanges au sein d'un canal restent internes à celui-ci, ils ne sont pas visibles des autres canaux. Le canal par défaut du serveur chat se nomme **place publique**. Le serveur permet de créer et de supprimer des canaux de discussion ainsi que de changer de canal de discussion lors de l'utilisation du chat.

Cas d'utilisation : Gérer des canaux de discussion.

Résumé : Un utilisateur peut créer des canaux de discussion, changer de canal de discussion et supprimer des canaux de discussion.

Acteur principal : Utilisateur.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur crée un canal de discussion (cf. 4.7.1, page 11).

Variantes :

- 1.a L'utilisateur supprime un canal de discussion (cf. 4.7.2, page 12).
- 1.b L'utilisateur change de canal de discussion (cf. 4.7.3, page 12).

4.7.1 Créer un canal de discussion

Cas d'utilisation : Créer un canal de discussion.

Résumé : Un utilisateur crée un canal de discussion sur le serveur du chat.

Acteur principal : Utilisateur.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur saisit la commande de création d'un canal.
2. Felix transmet la commande de création du canal à Camix.
3. Camix crée le canal.
4. Camix informe Felix qu'un canal a été créé.
5. Felix affiche un message de création du canal.

Exception [le canal existe déjà] :

- 3.a.1 Camix informe Felix que le canal existe déjà.
- 3.a.2 Felix affiche un message d'existence du canal.
- 3.a.3 Fin du CU.

4.7.2 Supprimer un canal de discussion

Cas d'utilisation : Supprimer un canal de discussion.

Résumé : Un utilisateur supprime un canal de discussion sur le serveur du chat.

Acteur principal : Utilisateur.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix). Un canal de discussion autre que le canal par défaut a été créé dans le chat.

Scénario nominal :

- 1. L'utilisateur saisit la commande de suppression d'un canal.
- 2. Felix transmet la commande de suppression du canal à Camix.
- 3. Camix supprime le canal.
- 4. Camix informe Felix qu'un canal a été supprimé.
- 5. Felix affiche un message de suppression du canal.

Exception [le canal n'existe pas] :

- 3.a.1 Camix informe Felix que le canal n'existe pas.
- 3.a.2 Felix affiche un message de non existence du canal.
- 3.a.3 Fin du CU.

Exception [le canal n'est pas vide] :

- 3.b.1 Camix informe Felix que le canal n'est pas vide.
- 3.b.2 Felix affiche un message de canal non vide.
- 3.b.3 Fin du CU.

Exception [le canal est le canal par défaut (place publique)] :

- 3.c.1 Camix informe Felix que le canal ne peut pas être supprimé.
- 3.c.2 Felix affiche un message de canal non supprimable.
- 3.c.3 Fin du CU.

4.7.3 Changer de canal de discussion

Cas d'utilisation : Changer de canal de discussion.

Résumé : Un utilisateur change de canal de discussion sur le serveur du chat.

Acteur principal : Utilisateur.

Acteurs secondaires : Les autres utilisateurs du chat.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix). Un canal de discussion autre que le canal par défaut a été créé dans le chat.

Scénario nominal :

1. L'utilisateur saisit la commande de changement de canal.
2. Felix transmet la commande de changement de canal à Camix.
3. Camix informe les composants Felix des utilisateurs inscrits dans le même canal que celui de l'utilisateur que ce dernier quitte le canal.
4. Chaque composant Felix concerné affiche un message de départ de l'utilisateur du canal.
5. Camix change l'utilisateur de canal.
6. Camix informe les composants Felix des utilisateurs inscrits dans le nouveau canal de l'utilisateur que celui-ci arrive dans le canal.
7. Chaque composant Felix concerné affiche un message d'arrivée de l'utilisateur dans le canal.

Exception [le canal n'existe pas] :

- 3.a.1 Camix informe Felix que le canal n'existe pas.
- 3.a.2 Felix affiche un message de non existence du canal demandé.
- 3.a.3 Fin du CU.

4.8 Sortir du chat

Cas d'utilisation : Sortir du chat.

Résumé : Un utilisateur sort du chat en fermant le logiciel client du chat.

Acteur principal : Utilisateur.

Acteurs secondaires : Les autres utilisateurs du chat.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur arrête l'exécution du composant Felix.
2. Felix se déconnecte de Camix.
3. Camix informe les composants Felix des utilisateurs inscrits dans le même canal que celui de l'utilisateur que ce dernier quitte le chat.
4. Chaque composant Felix concerné affiche un message de départ de l'utilisateur du chat.
5. Camix désinscrit l'utilisateur du canal dans lequel il se trouve.
6. Camix ferme la connexion de l'utilisateur.

4.9 Saisir une commande invalide

Cas d'utilisation : Saisir une commande invalide.

Résumé : Un utilisateur peut obtenir de l'aide sur les commandes du chat s'il saisit une commande invalide.

Acteur principal : Utilisateur.

Pré-conditions : Le composant client du chat (Felix) est lancé et connecté par TCP/IP au composant serveur du chat (Camix).

Scénario nominal :

1. L'utilisateur saisit une commande invalide.
2. Felix transmet la commande à Camix.
3. Camix transmet à Felix les informations sur les commandes disponibles.
4. Felix affiche un message d'aide sur les commandes disponibles.

5 Interface homme-machine

5.1 Interface homme-machine de Felix

La figure 4 présente l'interface homme-machine (IHM) du client chat Felix.

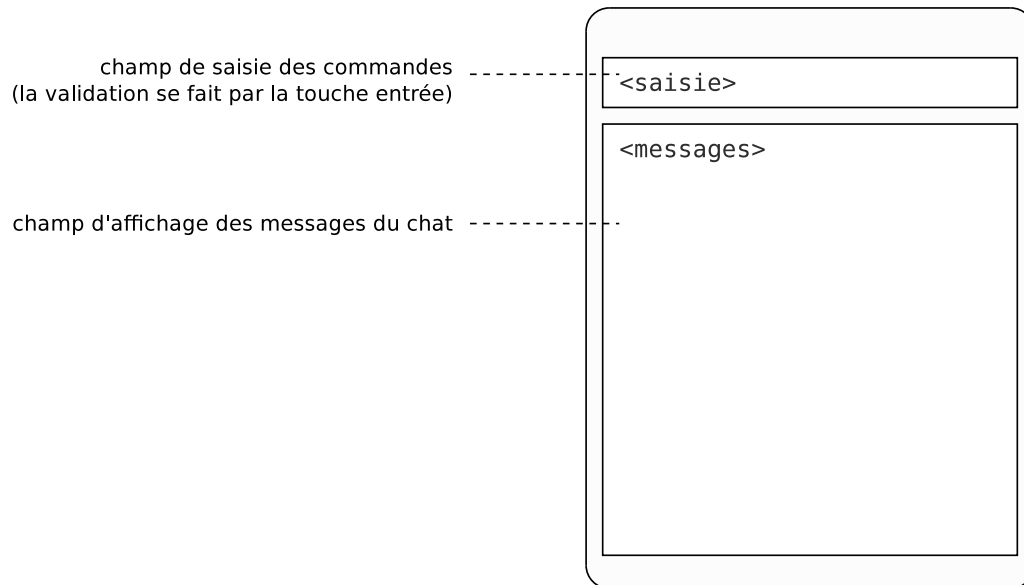


FIGURE 4 – IHM du client chat Felix : vue chat.

5.2 Protocole de communication client/serveur

Pour utiliser les différentes fonctionnalités du chat un protocole est à la disposition des utilisateurs. Ce protocole définit un ensemble de commandes que peut invoquer un utilisateur.

Le tableau 1 énumère les commandes disponibles dans le chat.

/n <i>surnom</i>	Changer de surnom.
/a <i>nom du canal</i>	Créer un canal de discussion.
/r <i>nom du canal</i>	Supprimer un canal de discussion.
/c <i>nom du canal</i>	Changer de canal de discussion.
/l	Afficher la liste des canaux de discussion.
/?	Afficher ses informations personnelles.
/h	Afficher l'aide sur les commandes du chat.

TABLE 1 – Liste des commandes du chat.

5.3 Messages textuels

L'utilisation du chat prévoit des messages communiqués aux utilisateurs par le serveur. Les listings 1, page 16, à 16, page 18, spécifient ces différents messages.

```
* Un nouvel utilisateur est dans le chat (place publique).
```

Listing 1 – Message d'arrivée d'un utilisateur dans le chat (à destination de la place publique).

```
* Taper /h pour avoir de l'aide sur les commandes du chat.
```

Listing 2 – Message d'accueil dans le chat.

```
* Commandes disponibles :  
  /n : changer de surnom ;  
  /c : changer de canal ;  
  /l : afficher les canaux ;  
  /a : créer un canal ;  
  /r : supprimer un canal ;  
  /? : afficher ses informations ;  
  /h : afficher l'aide.
```

Listing 3 – Message d'aide sur les commandes du chat.

```
utilisateur_x > Hello World !
```

Remarque : Un message est préfixé par le surnom de l'émetteur suivi de '>'.

Listing 4 – Exemple de message transmis dans le chat.

```
* utilisateur_x devient utilisateur_y.
```

Listing 5 – Exemple de message de changement de surnom d'un utilisateur.

```
* Informations personnelles :  
  Surnom : utilisateur_x  
  Canal  : place publique
```

Listing 6 – Exemple de message sur des informations personnelles.

```
* Canaux disponibles :  
  - canal_x (3)  
  - place publique (2)
```

Remarque : Le nombre d'utilisateurs dans un canal est donné entre parenthèses.

Listing 7 – Exemple de message sur des informations sur les canaux.

```
* Le canal canal_x a été créé.
```

Listing 8 – Exemple de message de création de canal.

```
* Le canal canal_x existe déjà.
```

Listing 9 – Exemple de message de création impossible de canal (le canal existe déjà).

```
* utilisateur_x quitte le canal place publique.
```

Listing 10 – Exemple de message de départ d'un canal.

```
* utilisateur_x rejoint le canal canal_x.
```

Listing 11 – Exemple de message d'arrivée dans un canal.

```
* Le canal demandé n'existe pas.
```

Listing 12 – Message de non existence d'un canal demandé (pour changement de canal)

```
* Le canal canal_x a été supprimé.
```

Listing 13 – Exemple de message de suppression d'un canal.

```
* Le canal canal_y n'existe pas.
```

Listing 14 – Exemple de message de suppression impossible d'un canal (le canal n'existe pas).

```
* Le canal canal_x n'est pas vide.
```

Listing 15 – Exemple de message de suppression impossible d'un canal (le canal n'est pas vide).

```
* Impossible de supprimer le canal par défaut du chat (place publique).
```

Listing 16 – Message de suppression impossible du canal « place publique ».

```
* utilisateur_x quitte le chat.
```

Listing 17 – Exemple de message de départ d'un utilisateur du chat.