Projet Génie Logiciel :

Application de chat client/serveur

Rapport intermédiaire

Amel Dussier, Batien Clément, Antoine Drabble et Guillaume Serneels

Table des Matières

Fonctionnement général **3**

Partage des résponsabilités clients/serveur **3**

Cas d'utilisation **4**
- **Diagramme général de contexte** **4**
- **Déscription des acteurs** **5**
  - Acteurs principaux 5
  - Acteurs secondaires 5
- **Description des cas d'utilisation** **5**
  - Création de compte 5
  - Connection à l'application 6
  - Rejoindre une discussion publique 6
  - Créer un groupe de discussion public 7
  - Créer une discussion privée 7
  - Supprimer un utilisateur d'un groupe de discussion public 8
  - Consulter les reports d'un groupe publique 8
  - Consulter les reports d'un groupe privé 9
  - Quitter un groupe 9
  - Gérer ses contacts 9
  - Ajouter un utilisateur dans un groupe 10
  - Reporter un autre utilisateur 10
  - Supprimer son compte 11
  - Envoyer un message 11
  - Consulter les messages 12
  - Se déconnecter de l'application 12
  - Charger des messages de l'historique 13

Protocole d'échange client/serveur **13**

Ebauche du modèle de domaine (découpage MVC) **14**
- **Modèle serveur** **14**
- **Modèle client** **15**

Base de données **16**
- **Objectif** **16**
- **Modèle conceptuel** **16**

Ebauche des interfaces utilisateur **17**
- **Fenêtre de connection** **17**
- **Fenêtre d'acceuil** **17**
- **Fenêtre de chat** **18**
- **Fenêtre d'édition de groupe** **18**

Gestion du projet **19**
- **Role des participants** **19**
- **Plan d'itérations** **20**
  - Itération 1 20
  - Itération 2 21
  - Itération 3 22
  - Itération 4 23
  - Itération 5 24
  - Itération 6 25

Glossaire **26**

Fonctionnement général

Nous allons réaliser une application de chat client-serveur ressemblant à Whatsapp et Telegram. La première fonction sera la création de compte, ou simplement la connection à un compte existant (en fournissant un nom d'utilisateur et un mot de passe). Il y aura ensuite trois types de chat possible :

- Un chat privé 1 à 1 qui permettra de communiquer avec ces contacts. La première version permettra de communiquer par message texte puis si le temps le permet, nous étudierons la possibilité d'implémenter la communication audio et les discussions chiffrée de bout en bout avec un système utilisant des clés publiques/privées. Les contacts pourront être ajoutés en les recherchant par leur nom d'utilisateur.

- Un chat privé de groupe. Un utilisateur pourra créer un groupe avec plusieurs de ses contacts et nommer le groupe. Toutes les personnes ajoutées pourront envoyer des messages dans ce groupe qui seront lisible par tous les membres. Le créateur du groupe en sera l'administrateur et pourra y ajouter/supprimer des membres.

.
Finalement, il y aura une fonctionnalité de report de messages qui permettra aux utilisateurs de reporter un message abusif. Tous les reports seront envoyés à l'administrateur de l'application et à l'administrateur du groupe si ce n'est pas une discussion 1 à 1. Les reports envoyés à l'administrateur seront stockés dans une base de données SQL et seront accessible via une interface web.

Un utilisateur pourra en bloquer un autre afin de ne plus recevoir de messages de sa part.

L'application serveur sera codée en Scala, l'application client sera sur Android et les informations seront stockées à l'aide d'une base de données SQL.

Partage des résponsabilités clients/serveur

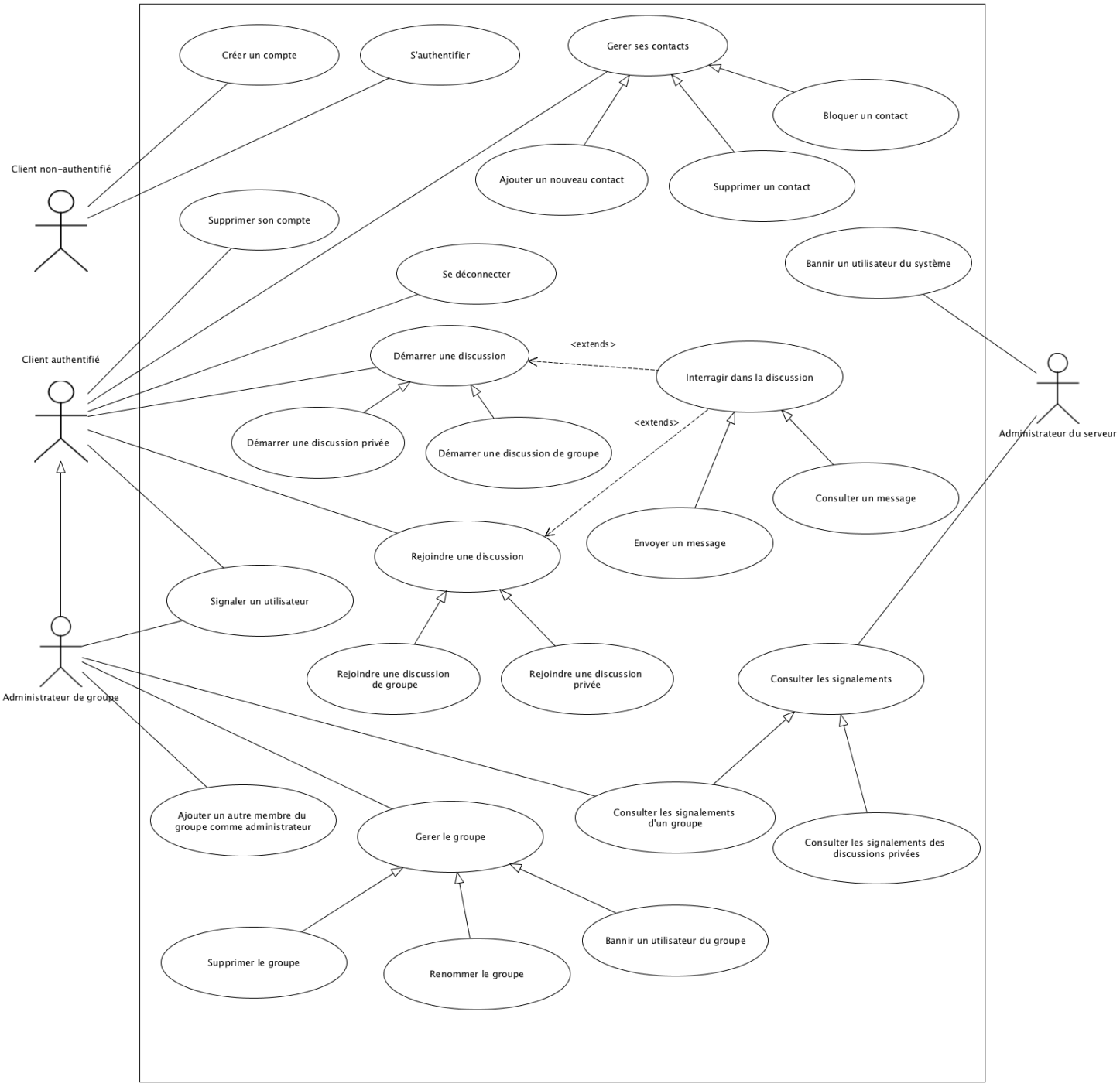
Le serveur s'occupe de gérer l'ensemble des opérations, le client n'est qu'une vue des données du système. Il affiche les informations que l'utilisateur demande et transmet les actions qu'il souhaite effectuer. Le serveur s'assure qu'il en aie l'autorisation.

Le serveur fonctionne en permanence. Le client peut à tout moment s'y connecter en démarrant l'application et en fournissant ses informations de connections (nom d'utilisateur et mot de passe).

Si l'application n'est pas lancée, l'utilisateur est considéré comme déconnecté. Après la première connection, le client maintient un cookie (token) de session lui permettant de se reconnecter rapidement

Cas d'utilisation

Diagramme général de contexte



Déscription des acteurs

Acteurs principaux

- Client non-authentifé
  Il devra soit créer un compte soit s'authentifier pour se transformer en client authentifié.
- Client authentifié
  C'est l'acteur qui représente les utilisateurs qui possèdent un compte et qui sont connectés.
- Administrateur de groupe
  Il est notifié des reports des discussions de son groupe et peut gérer les membres du groupe. Le rôle d'administrateur de groupe étend le rôle de client authentifié.

Acteurs secondaires

- Administrateur du système
  Il est notifié des reports de groupes et de discussions privées et peut bannir des utilisateurs de l'application. Cet acteur n'a pas de rôle de client.

Description des cas d'utilisation

Création de compte

Acteur(s) : Client non-authentifé

Description : Un client non-authentifé peut créer un nouveau compte

Scénario principal :

1. L'utilisateur ouvre l'application
2. L'utilisateur accède à l'interface d'inscription
3. L'utilisateur fournit un pseudonyme et un mot de passe (avec confirmation
4. Le serveur accepte et crée le nouveau compte
5. L'application authentifie l'utilisateur

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonné
b. Le compte existe déjà : l'application signale l'erreur et l'action est abandonnée

Connection à l'application

Acteur(s) : Client non-authentifé

Description : Un client non-authentifé peut se connecter avec un compte existant

Scénario principal :

1. L'utilisateur ouvre l'application
2. L'utilisateur accède à l'interface de connexion
3. L'utilisateur fournit un pseudonyme et un mot de passe
4. Le serveur accepte les informations de connexion
5. L'application authentifie l'utilisateur

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée
b. Les informations de connexion sont fausses : l'application signale l'erreur et l'action est abandonnée

Rejoindre une discussion publique

Acteur(s) : Client authentifé

Description : Un client authentifé peut rejoindre une discussion publique

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur accède à l'interface des conversations
3. L'utilisateur clique sur "Rejoindre une discussion"
4. L'utilisateur entre le nom de la discussion qu'il veut rejoindre
5. Le serveur autorise l'utilisateur à accéder à la discussion
6. L'application ouvre la discussion publique pour l'utilisateur

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée
b. Aucun groupe publique avec ce nom existe : l'application signale l'erreur et l'action est abandonnée
c. Le groupe existe mais l'utilisateur n'est pas autorisé (banni) : l'application signale l'erreur et l'action est abandonnée

Créer un groupe de discussion public

Acteur(s) : Client authentifé

Description : Un client authentifé peut créer une discussion publique

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur accède à l'interface des conversations
3. L'utilisateur clique sur "Créer une discussion publique"
4. L'utilisateur entre le nom de la discussion qu'il veut créer
5. Le serveur autorise l'utilisateur à créer la discussion
6. L'application ouvre la discussion publique pour l'utilisateur

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée
b. Un groupe publique avec ce nom existe : l'application signale l'erreur et l'action est abandonnée

Créer une discussion privée

Acteur(s) : Client authentifé

Description : Un client authentifé peut créer une discussion privée avec un autre utilisateur

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur accède à l'interface des conversations
3. L'utilisateur clique sur "Créer une discussion privée"
4. L'utilisateur choisit un de ces contacts enregistrés, ou entre le pseudonyme de l'utilisateur avec qui il veut avoir une discussion
5. Le serveur autorise l'utilisateur à créer la discussion
6. L'application ouvre la discussion privée pour l'utilisateur

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée
b. Aucun utilisateur avec ce pseudonyme existe : l'application signale l'erreur et l'action est abandonnée

Supprimer un utilisateur d'un groupe de discussion public

Acteur(s) : Administrateur de groupe

Description : Un administrateur de groupe peut supprimer un utilisateur d'une discussion publique

Scénario principal :

1. L'administrateur de groupe accède à une discussion publique qu'il administre
2. L'administrateur de groupe clique sur "Gérer la discussion"
3. L'administrateur de groupe clique sur "Voir les membres"
4. L'administrateur de groupe sélectionne un utilisateur membre du groupe
5. L'administrateur de groupe clique sur "Supprimer de la discussion"
6. L'application demande confirmation
7. Le serveur autorise la suppression de l'utilisateur de la discussion publique
8. L'application met à jour la liste des utilisateurs membres du groupe

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée

Consulter les reports d'un groupe publique

Acteur(s) : Administrateur du système, Administrateur de groupe

Description : Un administrateur peut consulter les reports d'une discussion publique

Scénario principal :

1. L'administrateur accède à une discussion publique qu'il administre
2. L'administrateur clique sur "Gérer la discussion"
3. L'administrateur clique sur "Voir les signalements"

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée

Consulter les reports d'un groupe privé

Acteur(s) : Administrateur du système

Description : Un administrateur du système peut consulter les reports d'une discussion privée

Scénario principal :

1. L'administrateur du système accède à l'interface de gestion du système
2. L'administrateur du système clique sur "Voir les signalements"

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée

Quitter un groupe

Acteur(s) : Client authentifé

Description : Un client authentifé peut quitter une discussion privée ou publique

Scénario principal :

1. L'utilisateur accède à une discussion privée ou publique
2. L'utilisateur clique sur "Gérer la discussion"
3. L'utilisateur clique sur "Quitter la discussion"
4. L'application demande confirmation
5. Le serveur accepte le départ de l'utilisateur du groupe
    5.1 Si la discussion est publique et que l'utilisateur était le dernier administrateur de la discussion, celle-ci passe en mode "lecture seule" pour les membres restants

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée

Gérer ses contacts

Acteur(s) : Client authentifé

Description : Un client authentifé peut gérer ses contacts (ajout, suppression, blocage)

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur accède à l'interface des contacts
3. L'utilisateur peut cliquer sur "Ajouter un contact"

a. L'utilisateur entre le pseudonyme d'un utilisateur
b. Le serveur accepte l'ajout de contact

4. L'utilisateur peut sélectionner un contact, puis cliquer sur "Supprimer" ou "Bloquer"

a. L'application demande confirmation
b. Le serveur accepte la suppression ou le blocage du contact

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée
b. Le pseudonyme de l'utilisateur à ajouter n'existe pas : l'application signale l'erreur et l'action est abandonnée

Ajouter un utilisateur dans un groupe

Acteur(s) : Administrateur de groupe

Description : Un administrateur de groupe peut ajouter un utilisateur au groupe

Scénario principal :

1. L'administrateur de groupe accède à une discussion publique qu'il administre
2. L'administrateur de groupe clique sur "Gérer la discussion"
3. L'administrateur de groupe clique sur "Ajouter un utilisateur"
4. L'administrateur de groupe choisit un de ces contacts enregistrés, ou entre le pseudonyme de l'utilisateur à ajouter
5. Le serveur accepte l'ajout de l'utilisateur

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée
b. Le pseudonyme de l'utilisateur à ajouter n'existe pas : l'application signale l'erreur et l'action est abandonnée

Reporter un autre utilisateur

Acteur(s) : Client authentifé

Description : Un client authentifé peut signaler un autre utilisateur

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur accède à une discussion
3. L'utilisateur sélectionne un ou plusieurs messages
4. L'utilisateur clique sur "Signaler"
5. L'application demande confirmation
6. Le serveur accepte le signalement

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée

Supprimer son compte

Acteur(s) : Client authentifé

Description : Un client authentifé peut supprimer son compte

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur clique sur "Gérer mon compte"
3. L'utilisateur clique sur "Supprimer mon compte"
4. L'application demande confirmation
5. Le serveur accepte la suppression du compte
6. L'application déconnecte l'utilisateur

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée

Envoyer un message

Acteur(s) : Client authentifé

Description : Un client authentifé peut envoyer un message dans une discussion

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur accède à une conversation
3. L'utilisateur saisit un message
4. L'utilisateur clique sur "Envoyer"
5. Le serveur accepte l'envoi du message
6. L'application met à jour le contenu de la discussion

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée

Consulter les messages

Acteur(s) : Client authentifé

Description : Un client authentifé peut consulter les messages d'une discussion

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur accède à une conversation
3. L'application met automatiquement à jour le contenu de la conversation

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée

Se déconnecter de l'application

Acteur(s) : Client authentifé

Description : Un client authentifé peut se déconnecter de l'application

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur clique sur "Gérer mon compte"
3. L'utilisateur clique sur "Se déconnecter"
4. L'application demande confirmation
5. L'application déconnecte l'utilisateur

Scénarios d'échec : aucun

Charger des messages de l'historique

Acteur(s) : Client authentifé

Description : Un client authentifé peut charger l'historique des messages

Scénario principal :

1. L'utilisateur ouvre l'application et se connecte avec un compte valide
2. L'utilisateur accède à une conversation
3. L'utilisateur remonte la conversation affichée
4. L'utilisateur clique sur "Historique", si tous les messages de la discussion ne sont pas déjà affichées
5. Le serveur accepte la demande et retourne un nombre prédéfini (par ex. 100) de messages de l'historique de la discussion
6. L'application met à jour le contenu de la discussion

Scénarios d'échec :

a. Le serveur n'est pas disponible : l'application signale l'erreur et l'action est abandonnée

Protocole d'échange client/serveur

Le protocole de communication client serveur devra permettre les commandes suivantes :

Client -> Serveur

- Demande de création de compte
- Validation/Refus de création de compte
- Demande de tentative de connection
- Validation/Refus de tentative de connection
- Demande d'ajout d'un contact
- Validation/Refus d'ajout de contact (par exemple s'il est bloqué ou s'il y a une erreur) (Ajout d'un lien dans les deux sens entre les deux contacts, création de discussions vide)
- Demande de création de groupe
- Validation/Refus de création de groupe
- Envoi d'un message à un contact/groupe
- Validation/Refus d'envoi du message
- Demande de déconnection
- Validation/Refus de déconnection

Serveur -> Client

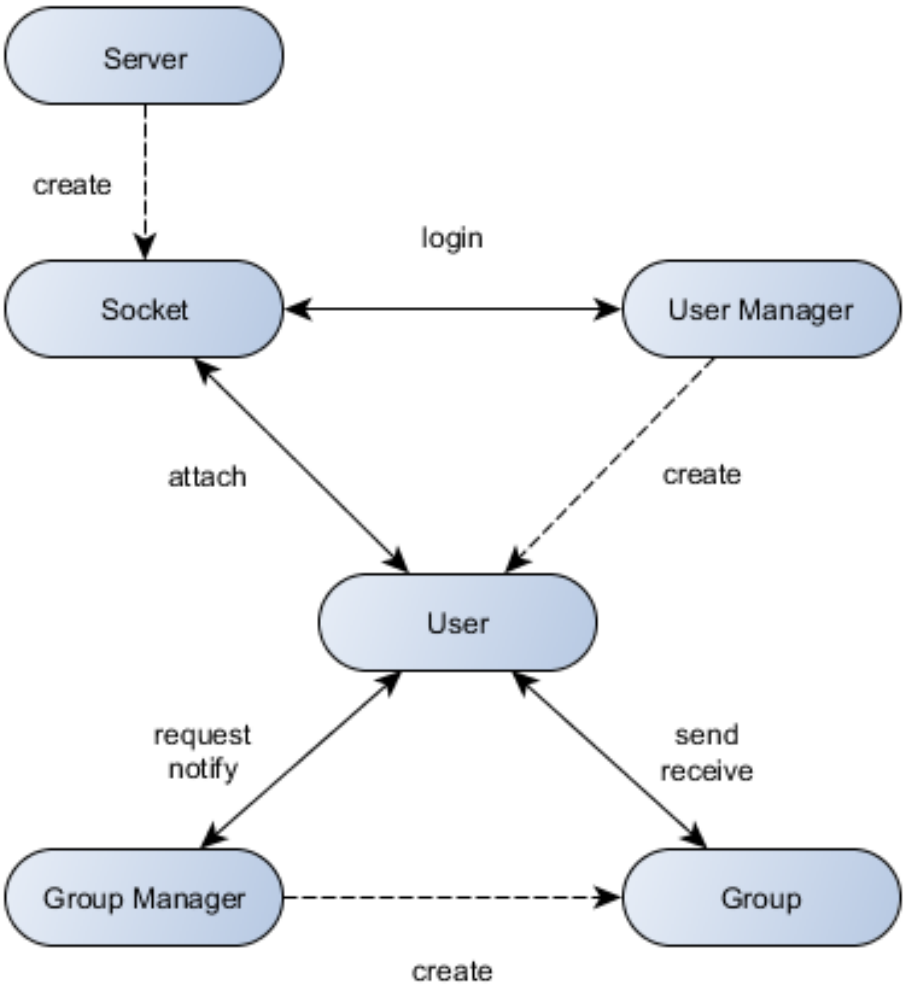
- Envoi d'un message au client
- Acceptation/Refus de réception
- Création d'un groupe dont le client est membre

- Acceptation/Refus de création

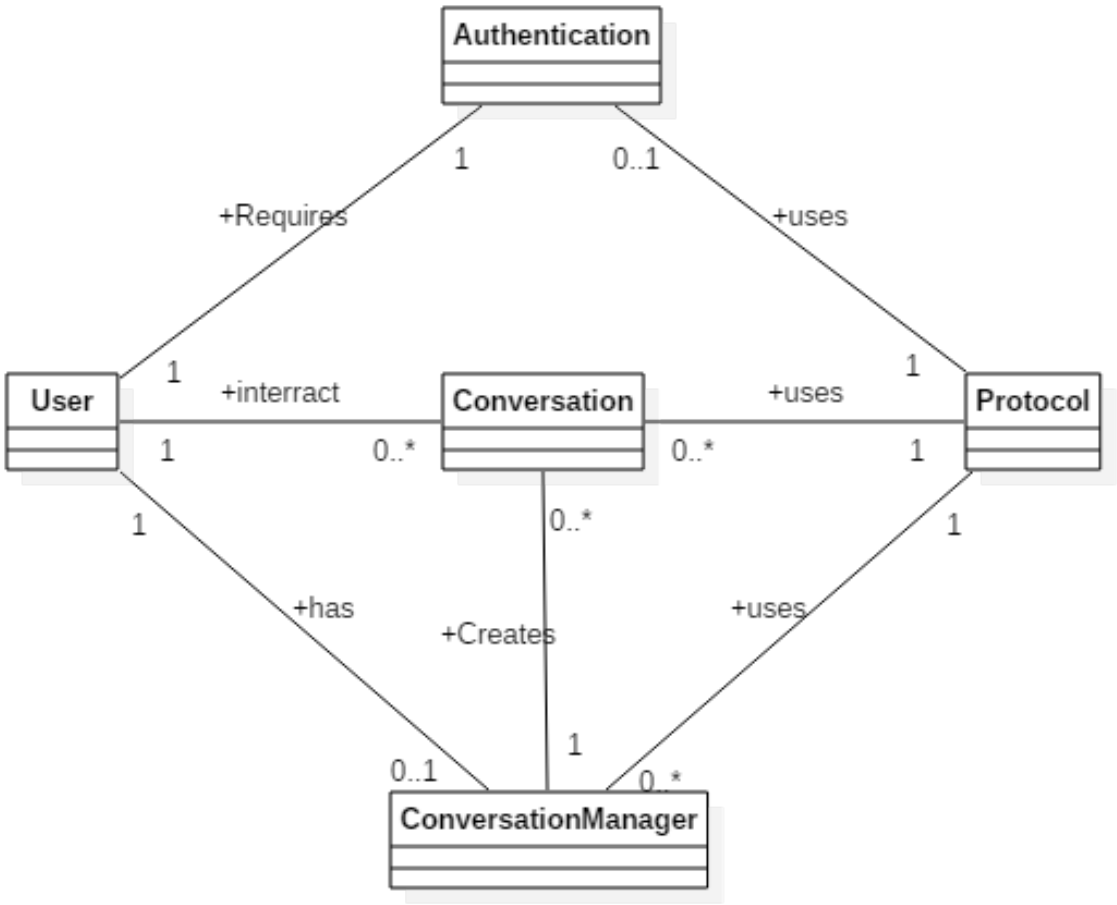
Une classe Java Protocole va définir les différentes commandes. Elle sera utilisée du côté serveur et client. Une commande sera représentée sur 3 bytes. Le premier byte va définir la commande à utilisé. On aura donc 255 commandes différentes possible ce qui devrait suffire. Les 2 bytes suivants définiront la longueur du messages soit une longueur maximum de 65535. Le message sera situé directement après ces 3 bytes.

Ebauche du modèle de domaine (découpage MVC)

Modèle serveur



Modèle client



Base de données

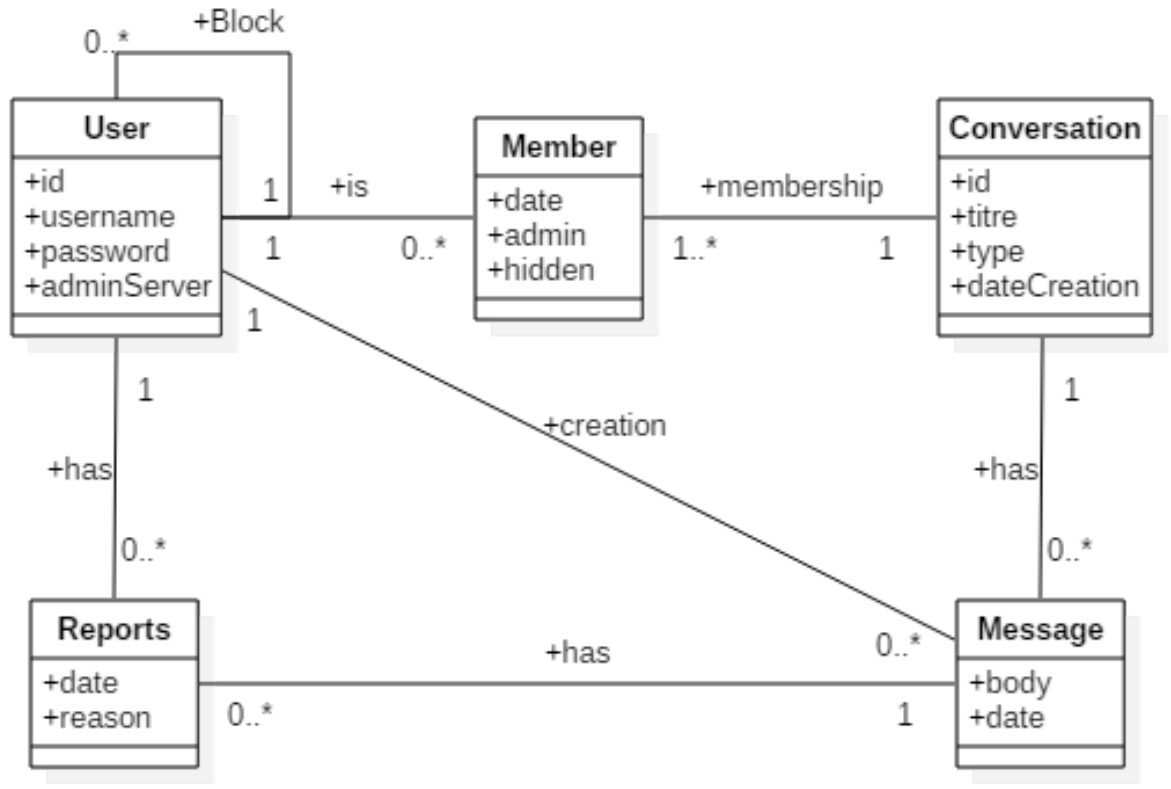
Objectif

La base de données définit des utilisateurs qui peuvent être membre de conversations. Les conversations privées seront différenciées des conversation de groupe par l'attribut type dans conversation.

Une conversation contient plusieurs messages qui sont liés à un utilisateur et les messages peuvent être reportés. Seul les administrateurs de groupes (membre dont l'attribut admin est à true) pourront voir les messages reportés.

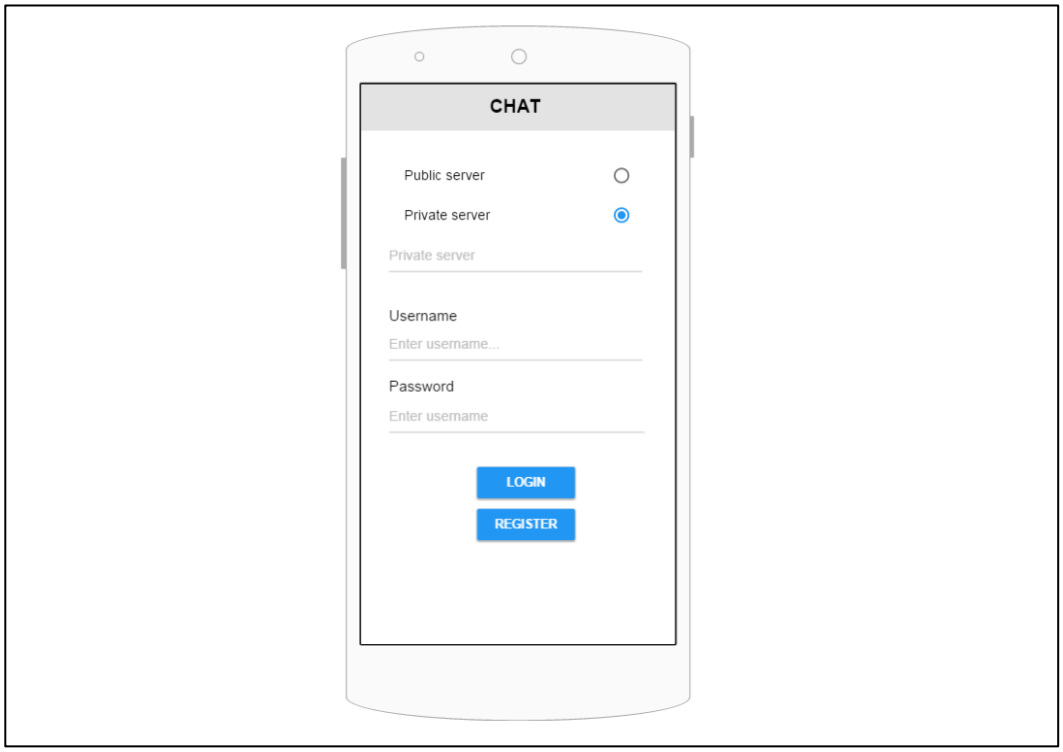
Un utilisateur peut en bloquer un autre via l'association block.

Modèle conceptuel

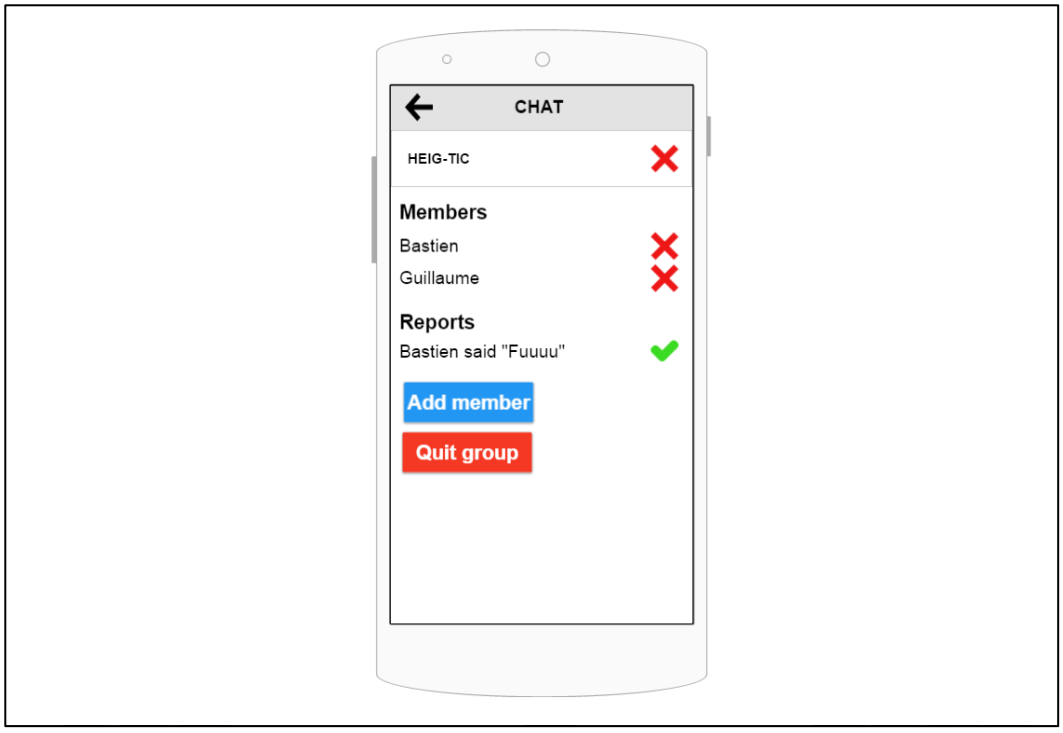


Ebauche des interfaces utilisateur

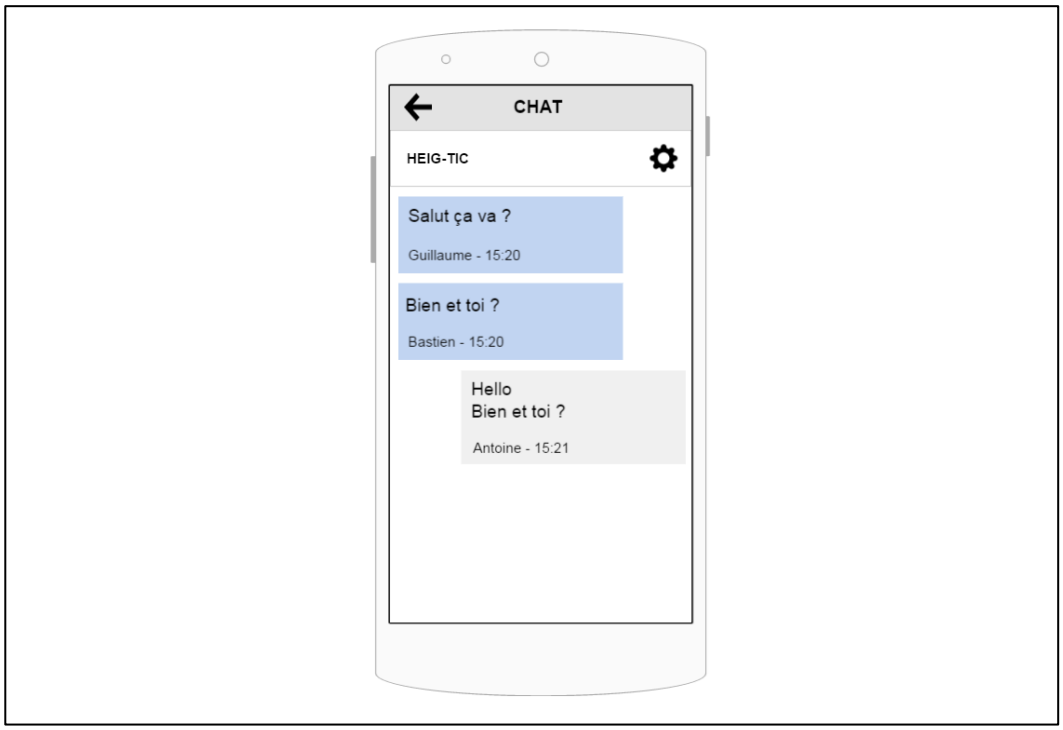
Fenêtre de connection



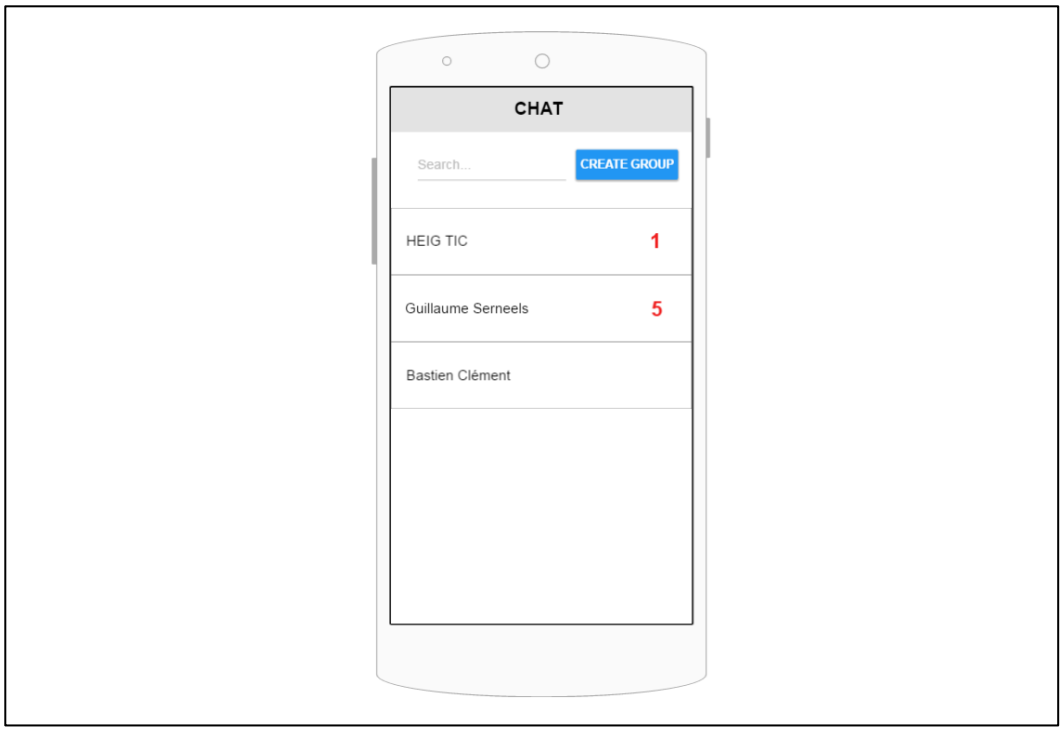
Fenêtre d'acceuil



Fenêtre de chat



Fenêtre d'édition de groupe



Gestion du projet

Role des participants

- **Représentant des utilisateurs** (Client) : Amel Dussier
  - Collecte des besoins
  - Spécification des tests de fonctionnalités
  - Explication des aspects métiers

- **Chef de projet : Project Manager** Amel Dussier
  - Planification
  - Coordination avec les utilisateurs

- **Analyste** Antoine Drabble
  - Spécifications
  - Collecte des demandes de changement

- **Software architect** Bastien Clément
  - Conception de l'architecture du produit

- **Programmeur** Bastien Clément, Antoine Drabble, Guillaume Serneels
  - Participe à la concep- tion du produit
  - Ecrit les tests unitaires
  - Codage

- **Responsable des tests** Guillaume Serneels
  - Participe à l'intégration continue des composants
  - Ecrit les tests fonc- tionnels
  - Met en place l'archi- tecture permettant de lancer régulièrement les tests fonctionnels

- **Responsable de la configuration** Bastien Clément
  - Gestion de la base des artefacts du projet
  - Gestion des releases
  - Allocation des droits
  - Responsable de la configuration (logicielle & matérielle)
  - Intégration des changements

Plan d'itérations

Itération 1

Objectif général : Création de la base de données et mise en place d'une première communication simple entre client-serveur

Objectifs détaillés :

- Apprendre les bases du développement Scala et Android (Gestion)
- Mettre en place la base de données (Développement de l'infrastructure)
- Définir une première version du protocole de communication (Gestion : Conception)
- Réaliser une première communication entre le client et le serveur (Développement des fonctionnalités)
    - Cet objectif correspond à réaliser partiellement tous les cas d'utilisation étant donné qu'ils nécessitent tous une communication client/serveur. A l'issue de l'itération, il sera possible de vérifier qu'un message simple a bien été transmis du client au serveur.

Durée : 2 semaines

Date de début : vendredi 22 avril

Date de fin : lundi 2 mai (Ascension)

Partage du travail :

- Antoine :
    - Apprendre les bases du développement Scala
    - Définition de la première version du protocole de communication
- Bastien :
    - Création de la base de données
    - Mise en place de la communication côté serveur
- Guillaume :
    - Apprendre les bases du développement Android
    - Mise en place de la communication côté client
- Amel :
    - Apprendre les bases du développement Android
    - Première ébauche de l'interface de connexion

Temp consacré : environ 40 heures (5 heures par personne et par semaine)

Itération 2

Objectif général : Mise en place des fonctionnalités de création / suppression de compte et de connexion

Objectifs détaillés :

- Ajouter les fonctionnalités de gestion de compte et de connexion au protocole de communication (Gestion : Conception)
- Interfacer l'application serveur avec la base de données (Développement de l'infrastructure)
- Implémenter la gestion des comptes et de connexion au niveau du serveur et du client (Développement des fonctionnalités)
    - Cas d'utilisation réalisés complètement : Création de compte, Connexion à l'application
    - Cas d'utilisation réalisés partiellement : Supprimer son compte, Se déconnecter
        - Ces fonctionnalités seront implémentées au niveau du code, mais leur intégration à l'interface utilisateur se fera lors des itérations suivantes.
- Commencer la rédaction du rapport final, avec la structure des chapitres (Gestion : Rédaction)

Durée : 1 semaine

Date de début : mardi 3 mai

Date de fin : lundi 9 mai

Partage du travail :

- Antoine :
    - Ajouter les fonctionnalités de gestion de compte et de connexion au protocole de communication
    - Interfacer l'application serveur avec la base de données
- Bastien :
    - Implémenter la gestion des comptes et de connexion au niveau du serveur
- Guillaume :
    - Implémenter la gestion des comptes et de connexion au niveau du client
- Amel :
    - Finaliser l'interface utilisateur pour la création de compte et la connexion
    - Commencer la rédaction du rapport final

Temp consacré : environ 20 heures (5 heures par personne)

Itération 3

Objectif général : Mise en place des fonctionnalités de recherche et de gestion de contacts

Objectifs détaillés :

- Ajouter les fonctionnalités de recherche et de gestion de contacts au protocole de communication (Gestion : Conception)
- Implémenter la recherche et de gestion de contact au niveau du serveur et du client, ainsi que les interfaces utilisateur correspondantes (Développement des fonctionnalités)
    - Cas d'utilisation réalisés complètement : Gérer les contacts
    - Cas d'utilisation réalisés partiellement : Créer une discussion privée, Ajouter un utilisateur dans un groupe public
        - La fonctionnalité de recherche est utilisée dans plusieurs cas d'utilisation. La réalisation complète de ces cas d'utilisation se fera lors des itérations suivantes.
- Continuer la rédaction du rapport final (Gestion : Rédaction)

Durée : 1 semaine

Date de début : mardi 10 mai

Date de fin : lundi 16 mai

Partage du travail :

- Antoine :
    - Ajouter les fonctionnalités de recherche et de gestion de contact au protocole de communication
    - Continuer la rédaction du rapport
- Bastien :
    - Implémenter la recherche et de gestion de contact au niveau du serveur
- Guillaume :
    - Implémenter la recherche et de gestion de contact au niveau du client
    - Ajout de l'interface de recherche de contact
- Amel :
    - Ajout de l'interface de gestion de contact

Temp consacré : environ 20 heures (5 heures par personne)

Itération 4

Objectif général : Mise en place des discussions privées

Objectifs détaillés :

- Ajouter les fonctionnalités de création de discussion privée, d'envoi de messages et d'historique au protocole de communication (Gestion : Conception)
- Implémenter la création / suppression de discussion, l'envoi de messages et l'affichage de l'historique au niveau du serveur et du client, ainsi que les interfaces utilisateur correspondantes (Développement des fonctionnalités)
    - Cas d'utilisation réalisés complètement : Créer une discussion privée, Envoyer un message, Consulter les messages, Charger les messages de l'historique
    - Cas d'utilisation réalisés partiellement : Créer un groupe de discussion public
        - Une discussion publique est une extension d'une discussion privée, donc certaines fonctionnalités seront identiques. La réalisation complète de ce cas d'utilisation se fera lors des itérations suivantes.
- Continuer la rédaction du rapport final (Gestion : Rédaction)

Durée : 1 semaine

Date de début : mardi 17 mai

Date de fin : lundi 23 mai

Partage du travail :

- Antoine :
    - Ajouter les fonctionnalités de création de discussion privée, d'envoi de messages et d'historique au protocole de communication
    - Implémenter la gestion de l'historique de discussion au niveau du serveur
- Bastien :
    - Implémenter la création / suppression de discussion, l'envoi de messages au niveau du serveur
- Guillaume :
    - Implémenter la création / suppression de discussion, l'envoi de messages et l'affichage de l'historique au niveau du client
    - Continuer la rédaction du rapport
- Amel :
    - Ajout de l'interface de création et d'affichage de discussion
    - Ajout de l'interface de saisie de message

Temp consacré : environ 20 heures (5 heures par personne)

Itération 5

Objectif général : Création de discussion publique (discussion de groupe)

Objectifs détaillés :

- Ajouter les fonctionnalités de discussion publique (créer, supprimer, rejoindre), et de gestion des membres (ajouter, supprimer, promouvoir administrateur de groupe) d'une discussion publique au protocole de communication (Gestion : Conception)
- Implémenter toutes les fonctionnalités relatives aux discussions publiques au niveau du serveur et du client, ainsi que les interfaces utilisateur correspondantes (Développement des fonctionnalités)
    - Cas d'utilisation réalisés complètement : Rejoindre une discussion publique, Créer un groupe de discussion public, Supprimer un utilisateur d'un groupe de discussion public, Quitter un groupe, Ajouter un utilisateur dans un groupe
- Continuer la rédaction du rapport final (Gestion : Rédaction)

Durée : 2 semaines

Date de début : mardi 24 mai

Date de fin : lundi 6 juin

Partage du travail :

- Antoine :
    - Ajouter toutes les fonctionnalités relatives aux discussions publiques au protocole de communication
    - Aider Bastien et Guillaume pour implémenter les fonctionnalités au niveau serveur ou client
- Bastien :
    - Implémenter toutes les fonctionnalités relatives aux discussions publiques au niveau du serveur
- Guillaume :
    - Implémenter toutes les fonctionnalités relatives aux discussions publiques au niveau du client
    - Extension de l'interface d'affichage de discussion privée pour gérer les discussions publiques
- Amel :
    - Ajout de l'interface de gestion de groupes
    - Continuer la rédaction du rapport

Temp consacré : environ 40 heures (5 heures par personne et par semaine)

Itération 6

Objectif général : Signalement et blocage

Objectifs détaillés :

- Ajouter les fonctionnalités de signalement de message et de blocage d'utilisateur au protocole de communication (Gestion : Conception)
- Implémenter toutes les fonctionnalités relatives au signalement de message et de blocage d'utilisateur au niveau du serveur et du client, ainsi que les interfaces utilisateur correspondantes (Développement des fonctionnalités)
    - Cas d'utilisation réalisés complètement : Consulter les signalements d'un groupe public, Consulter les reports d'un groupe privé, Reporter un autre utilisateur
- Terminer la première version du rapport final (Gestion : Rédaction)

Durée : 1 semaine

Date de début : mardi 7 juin

Date de fin : lundi 13 juin

Partage du travail :

- Antoine :
    - Ajouter les fonctionnalités de signalement de message et de blocage d'utilisateur au protocole de communication
    - Terminer la première version du rapport final
- Bastien :
    - Implémenter toutes les fonctionnalités relatives au signalement de message et de blocage d'utilisateur au niveau du serveur
    - Finaliser l'application serveur
- Guillaume :
    - Implémenter toutes les fonctionnalités relatives au signalement de message et de blocage d'utilisateur au niveau du client
    - Finaliser l'application cliente et les différentes interfaces
- Amel :
    - Ajouter les options de signalement et de blocage aux différentes interfaces utilisateur
    - Finaliser les différentes interfaces

Temp consacré : environ 20 heures (5 heures par personne)

Glossaire

Discussion privée : Une discussion privée (1 à 1)

Discussion de groupe : Une discussion de groupe privée (3 et +)