

Compte rendu du projet SAE3.02

Par HARTMANN Matthias et SANDRA Valentin

Table des matières

1. Gestion du projet
 1. Organisation interne de l'équipe
 2. Organisation des tâches
 3. Support de communication
2. Fonctionnalités
 1. Rappel du cahier des charges
 1. Cahier des charges obligatoire du projet
 2. Cahier des charges de la solution
 3. Améliorations éventuelles
 2. Protocole mis en place
 1. Mots clés du protocole
 2. Diagramme des échanges

Gestion du projet

Organisation interne de l'équipe

Afin de produire efficacement l'application, nous avons décidé de diviser le travail en deux parties.

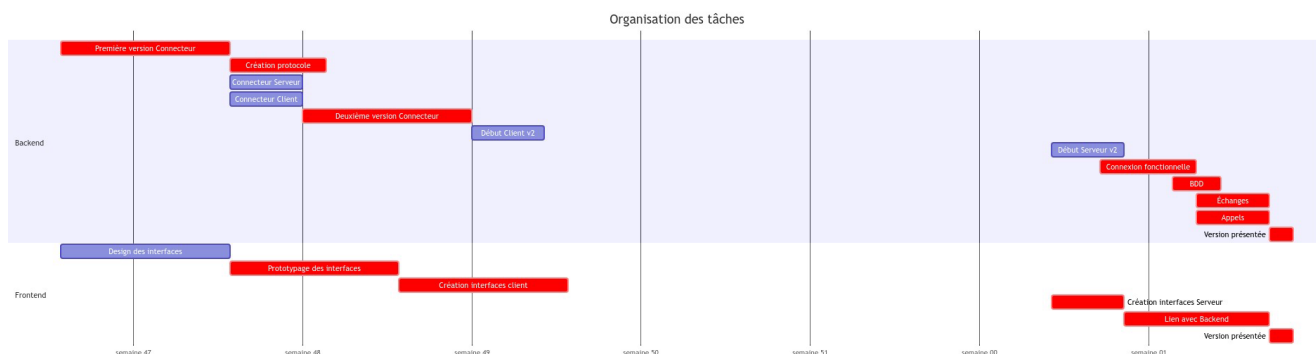
- Le Front End
- Le Back End

Le front end, c'est à dire toutes les interfaces utilisateurs ont été faites par Valentin. Le back end, c'est à dire la gestion de connexions, des bases de données et du serveur, a été fait par Matthias.

La mise en relation des deux parties a été faite majoritairement par Valentin car la charge de travail du front end était la plus petite.

Organisation des tâches

Vous retrouverez ci-dessous un digramme Gantt montrant l'avancée du projet en fonction des semaines et de la division du travail.



Support de communication







Pour assurer la communication interne à l'équipe nous avons utilisé le logiciel Discord, nous faisons nos réunions dessus pendant les heures d'autonomie. De plus nous avons utilisé le logiciel de versionnage github afin d'enregistrer notre travail et de plus facilement voir son évolution.

Fonctionnalités












Rappel du cahier des charges

Les éléments présents dans la solution commencent par , les éléments non présents commencent par 






Cahier des charges obligatoire du projet

-  Création d'une application client/serveur basé sur un ou plusieurs protocols de transport (TCP/UDP)
-  Utilisation d'un protocole applicatif, d'une base de donnée ~~ou d'un système de fichier.~~
-  Programmation en Python (*Ici en Python 3.9*)
-  Application coté client unique pour tous les clients
-  Création d'une interface graphique avec une option pour les paramètres les éléments réseaux
-  Utilisation du processus de sérialisation et désérialisation.

Cahier des charges de la solution

-  Un client doit pouvoir ouvrir le SoftPhone, s'enregistrer sur le serveur ou bien se connecter s'il est déjà enregistré
-  Le client doit pouvoir avoir accès à la liste des clients connectés sur le serveur
-  Le client doit pouvoir changer son mot de passe ou bien son nom/pseudo
-  Le client ne doit pas pouvoir être en mesure d'obtenir l'adresse ip d'un autre client
-  Le client doit pouvoir se déconnecter
-  Le client doit recevoir des informations lorsqu'il est en appel. (temps d'appel, nom du correspondant...)
-  Le client doit pouvoir créer un annuaire téléphonique lui permettant d'enregistrer le nom de certains clients afin de facilement les rappeler
-  Le serveur doit pouvoir enregistrer les actions opérées (LOG)
-  Le serveur doit pouvoir obtenir des informations sur un client
-  Le serveur doit pouvoir être contrôlé à l'aide de commandes (Graphique ~~ou Non~~)
-  Le serveur doit pouvoir gérer plusieurs clients en même temps (au moins 2)

Améliorations éventuelles

-  Possibilité de faire sonner un client (appel non immédiat)
 -  Interface permettant d'accepter ou non un appel entrant
 -  Interface permettant de mettre en attente un appel et de décrocher un autre
-  Mettre en place un système de statut d'apparence (Disponible, Absent, Occupé..)
-  Mise en place de ConfCall (Appel avec plus de 2 interlocuteurs)

Protocole mis en place

Nous avons créer notre propre protocole, vous retrouverez la liste des mots clés ainsi que des diagrammes d'échange ci-dessous.

Mots clés du protocole

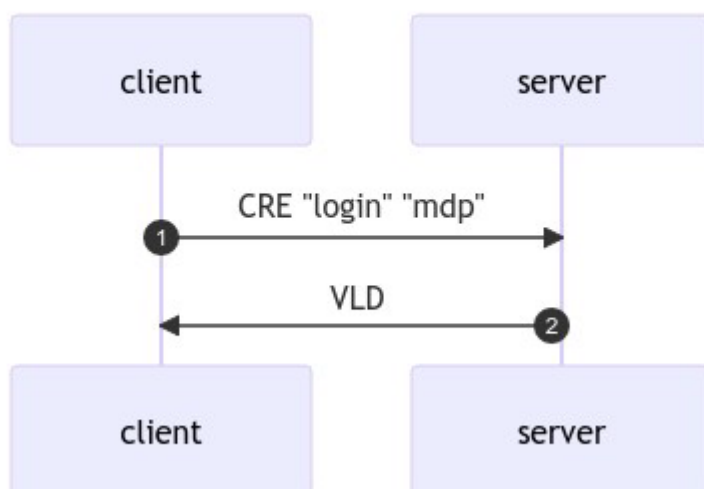
```

REG → Lance le processus d'authentification
LOG → Donne le nom d'utilisateur "log <username>"
PSS → Donne le mot de passe "pss <password>"
AUT → Informe que le client est authentifié
CRE → Créer un utilisateur "cre <username> <password>"
# Qualité de service
VLD → Retour Valide
REF → Retour erreur "ref [pourquoi]"
ENT → Ferme la connexion TCP
NUL → Mot clé par défaut
# Commands
LSR → Retourne la liste des clients connectés "lsr <username> <username> ..."
LSD → Demande la liste des clients connectés "lsd <username> <username> ..."

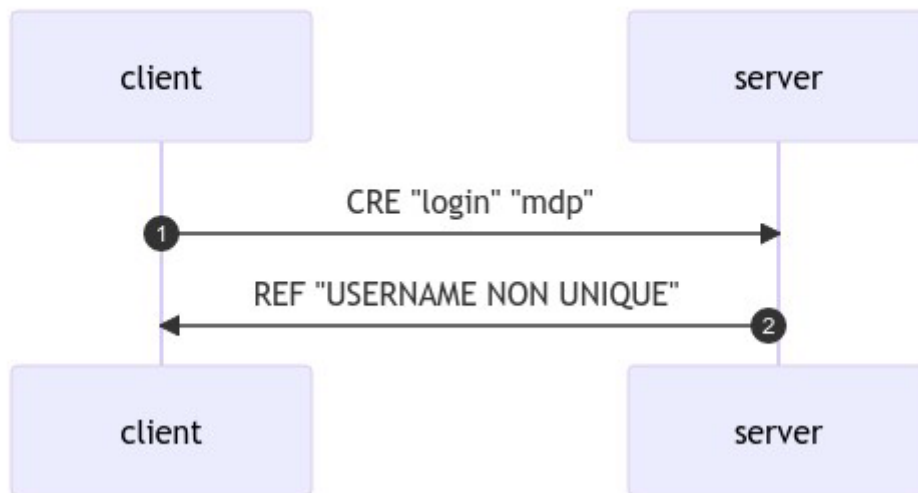
CAL → Prépare un appel "cal <username> [usernames...]"
ASK → Notifie un client qu'un appel lui est assigné "ask <callName>"
RES → Accepte ou non l'appel "res <bool:1|0>"
STA → Démarre l'appel
INF → Informations générales à l'appel "info time:00h00m00s act:usr1,usr2,usr3,..."
FIN → Termine l'appel
  
```

Diagramme des échanges

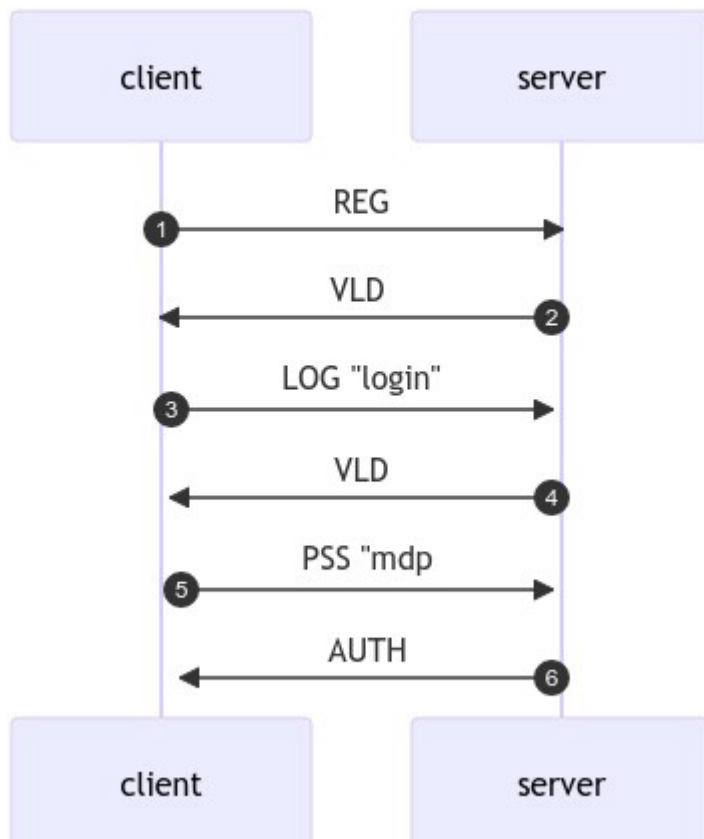
Création d'un compte (Valide)

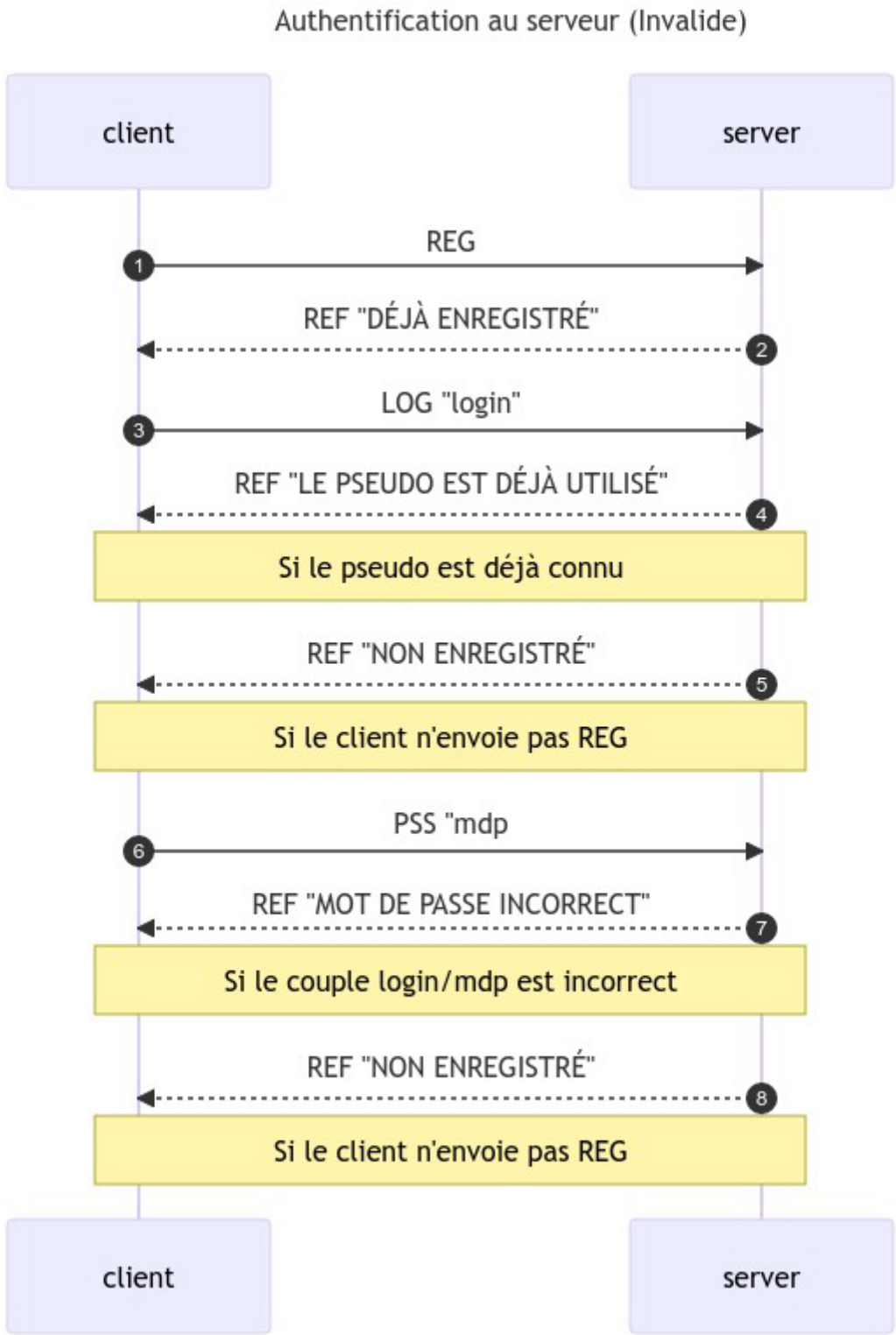


Création d'un compte (Invalide)

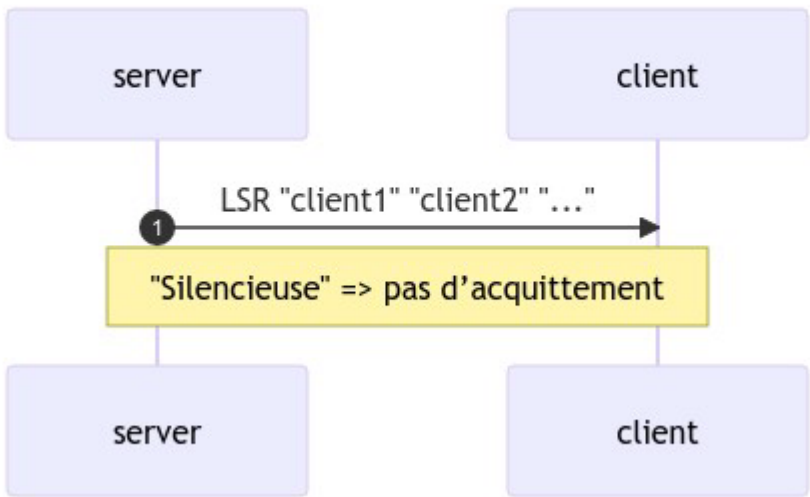


Authentification au serveur (Valide)

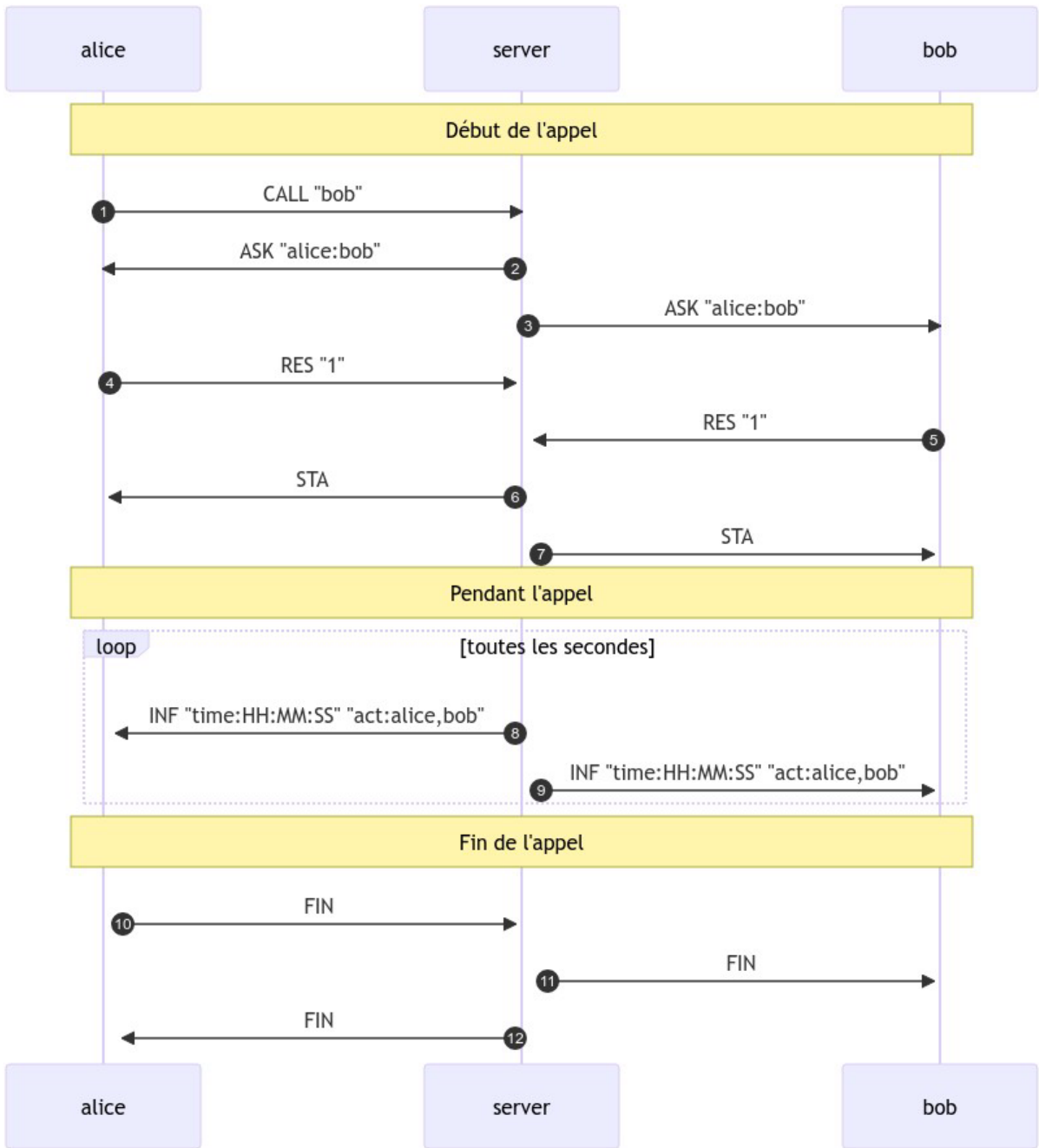




Récupération de la liste des clients



Procédure d'appel



Fermeture de la connexion

