

Dossier de conception

Logiciel : Projet S5

Equipe de développement :

- Vincent FOUGERAS
- Quentin MARTY
- Pierre POMERET-COQUOT

(L3 Info groupe 1)

1. Cahier des charges

Production de trois logiciels :

- Application client
- Application administrateur
- Serveur

1.a. Application client

Interface graphique client.

Fonctionnalités :

- Connexion avec couple identifiant/mot de passe
- Consultation de la liste des groupes
- Création d'un ticket à destination d'un groupe
- Ajout d'un message sur un ticket (créé par l'utilisateur ou a destination de l'un de ses groupes).
- Consultation des tickets et messages

Propriétés des messages :

- Date d'émission
- Statut de lecture (non transmis, reçu, lu)

1.b. Application administrateur

Fonctionnalités :

- Création/suppression de groupes
- Création/suppression de comptes utilisateur
- Création/suppression de liens d'appartenance utilisateur-groupe
- Suppression de tickets / messages

1.c. Serveur

Serveur gérant les applications client et administrateur.

Fonctionnalités :

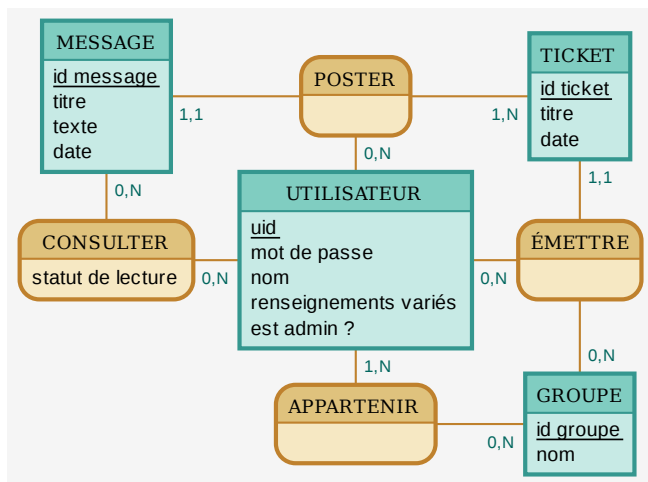
- Permettre la connexion simultanée de multiples applications client
- Répondre aux requêtes des applications client
- Vérifier la validité des requêtes provenant des applications client
- Enregistrer l'information nécessaire aux applications client.
- Permettre la connexion d'une application administrateur
- Exécuter et répondre aux requêtes de l'application administrateur

Le serveur utilisera une base de données

2. Analyse

2.1. Modèle de données

L'information peut être modélisée ainsi :



3. Axes de développement

L'application sera écrite en Java. La base de donnée utilisera le SGDB mySQL.

3.1. Modèle de données

3.1.a. SQL

```

MESSAGE ( id message, titre, texte, date, #uid émetteur, #id ticket )
TICKET ( id ticket, titre, date, #uid créateur, #id groupe )
CONSULTEUR ( #id message, #uid lecteur, statut de lecture )
UTILISATEUR ( uid, mot de passe, nom, renseignements variés, est admin ? )
APPARTENIR ( #uid, #id groupe )
GROUPE ( id groupe, nom )
    
```

3.1.b. Java

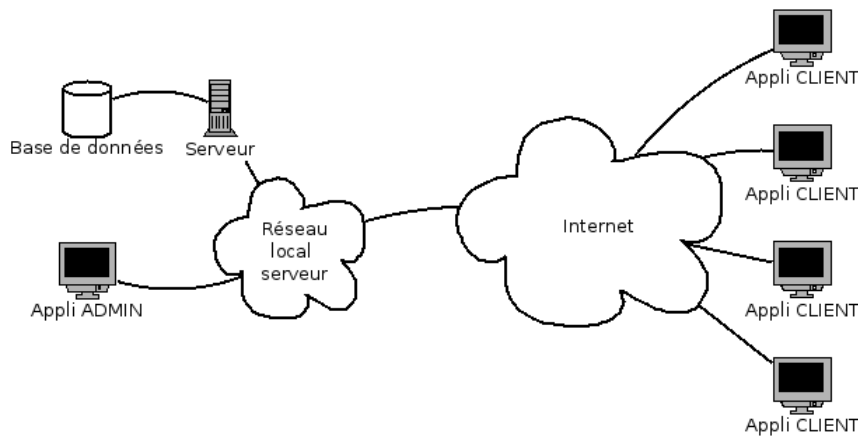
Au sein d'un package *modele* destiné aux applications et au serveur.

L'information sera contenue dans les classes suivantes :

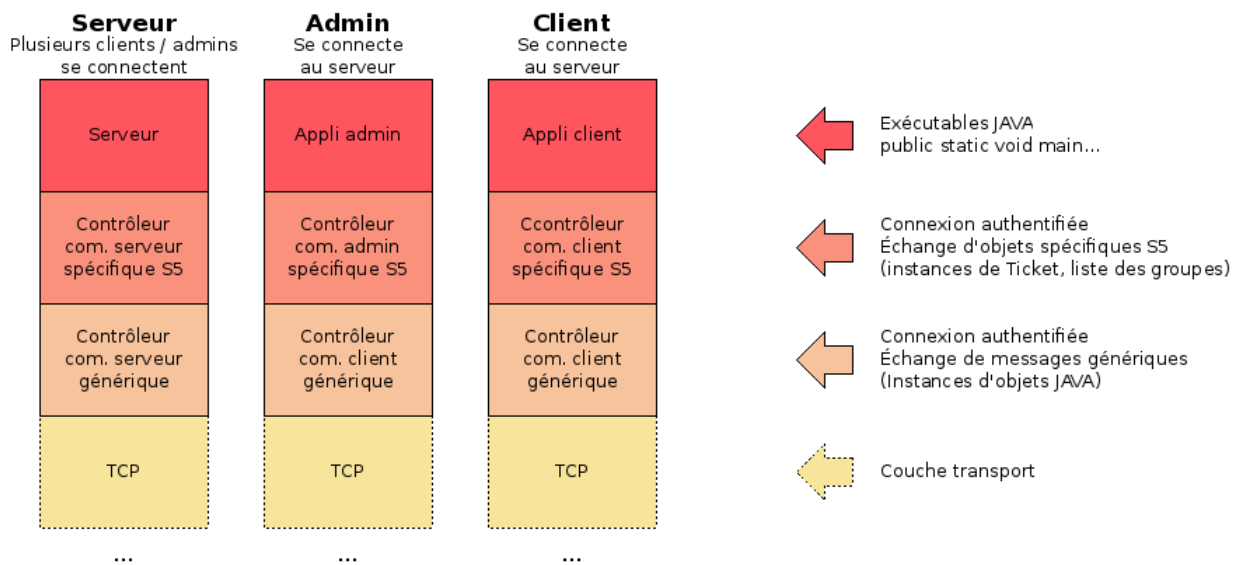
- Groupe (contient des tickets)
- Ticket (contient des messages)
- Message (contient un utilisateur)
- Utilisateur

Chacune de ces classes peut être identifiée par un entier ou une chaîne de caractère unique. Par ailleurs, la classe Ticket pourra être instanciée avec des lacunes (sans savoir les messages contenus dans ce ticket) : ces lacunes seront comblées à la demande par l'application.

Diagramme de classe :



Communication sur TCP :



La communication réseau est assurée par trois packages :

3.2.a. Package *communication*

Interfaces pour un outil générique permettant la connexion authentifiée de clients à un serveur, et leur dialogue.

- **ControleurComClient** : dialogue avec un ControleurComServeur
- **ControleurComServeur** : dialogue avec plusieurs ControleurComClient
- **ObservateurComClient** : est informé des messages reçus par un ControleurComClient
- **ObservateurComServeur** : est informé des messages reçus par un ControleurComServeur

Prévu pour être réutilisé dans d'autres logiciels : ne dépend pas du modèle de données choisi.

3.2.b. Package *communication.simple*

Implémentations concrètes des interfaces définies dans le package *communication*.

Echange de messages contenant un tableau (array) d'objets sérialisables.

Le premier élément du tableau est un énuméré indiquant le rôle du message (message d'accueil, établissement de connexion, réponse, requête, information).

Son but est de savoir comment *caster* les autres éléments si besoin.

```
public enum SimpleTypeMessage implements Serializable {
    BONJOUR,
    IDENTIFICATION,
    CONFIRM_IDENTIFICATION,
    INFORME,
    DEMANDE,
```

```
}          INVALIDE;
```

3.2.c. Package *commChatS5*

Implémentation de contrôleurs spécifiques aux applications développées :

- CtrlComClient
 - Demander la liste des groupes et leurs contenus (tickets)
 - Demander le contenu d'un ticket
 - Créer un nouveau ticket / message
 - Etre informé d'un nouveau ticket / message
- CtrlComAdmin
 - Consulter toutes les données
 - Modifier toutes les données
 - Supprimer toutes les données
- CtrlComServeur
 - Traite les requêtes des clients et admin

Les interfaces suivantes permettent aux applications client, admin et serveur d'être informées par un contrôleur :

- S5Client : méthodes qu'utilise un CtrlComClient pour informer l'appli client
- S5Admin : méthodes qu'utilise un CtrlComAdmin pour informer l'appli admin
- S5Serveur : méthodes qu'utilise un CtrlComServeur pour produire les réponses aux requêtes des clients / admin

Le package utilise *communication.simple*, et donc SimpleMessage pour communiquer : on envoie des tableaux d'objets sérialisable.

Le premier élément du tableau n'est pas visible, et le reste de ce tableau est utilisé de la même manière que précédemment :

- Un énuméré indiquant le type de message (demande de ticket, demande de la liste des groupes, réponse, etc.)
- Eventuelles informations (id du ticket demandé, ticket, liste des groupes, etc.).

L'énuméré sert, une fois encore, à savoir comment *caster* les autres cases du tableau.

```
public enum TypeMessage {
    REQUETE_LISTE_GROUPE,
    REQUETE_TICKET,
    REQUETE_NOUVEAU_TICKET,
    REQUETE_NOUVEAU_MESSAGE,

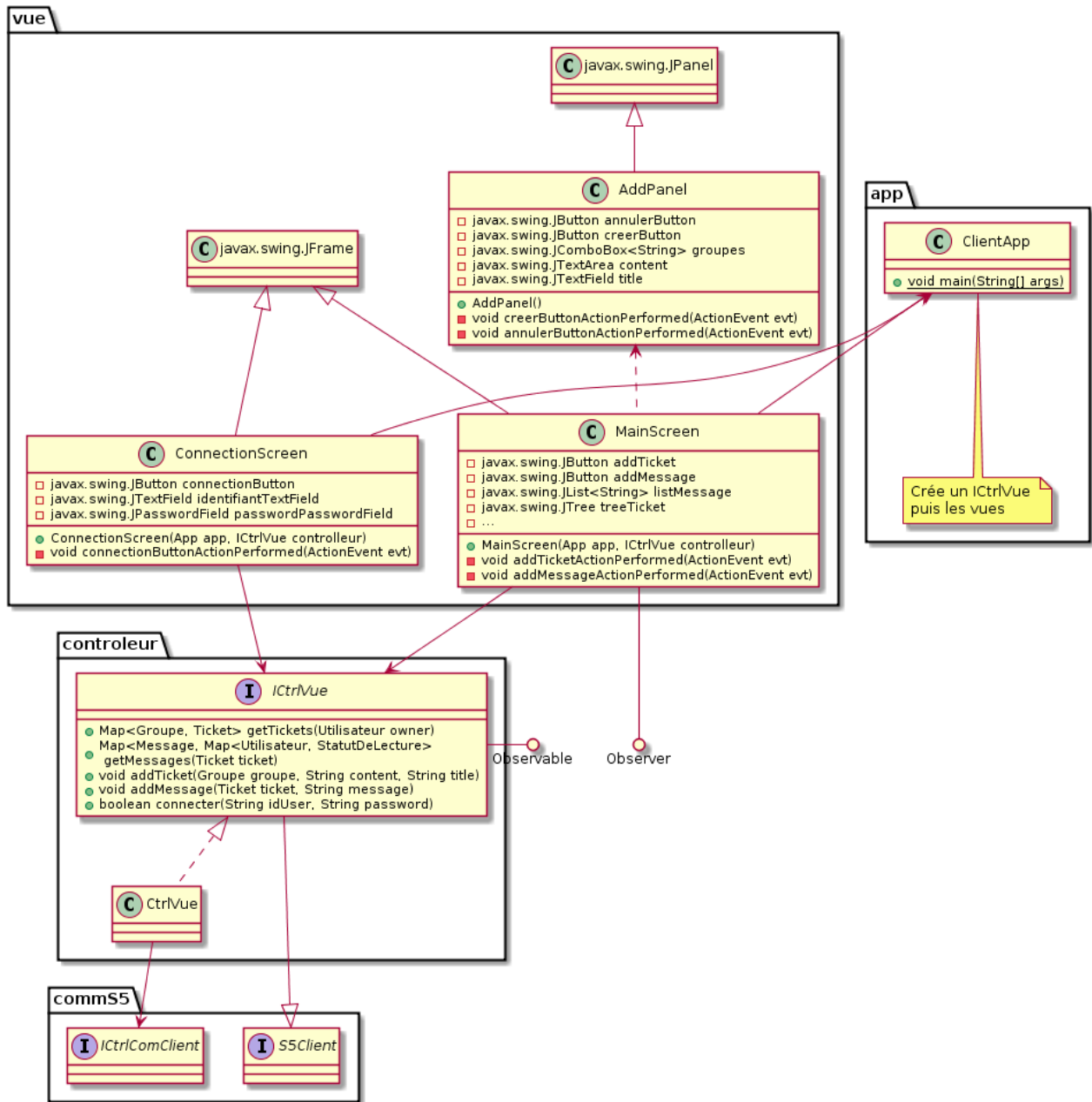
    INFORME_LISTE_GROUPE,
    INFORME_TICKET,
    INFORME_MESSAGE,

    [...]

    INCONNU;
}
```

3.3. IHM et Controleur de l'application client

Diagramme de classe :



Maquettes de l'IHM : Ecran de connexion :

Chat Uni

Connexion

Identifiant

Mot de passe

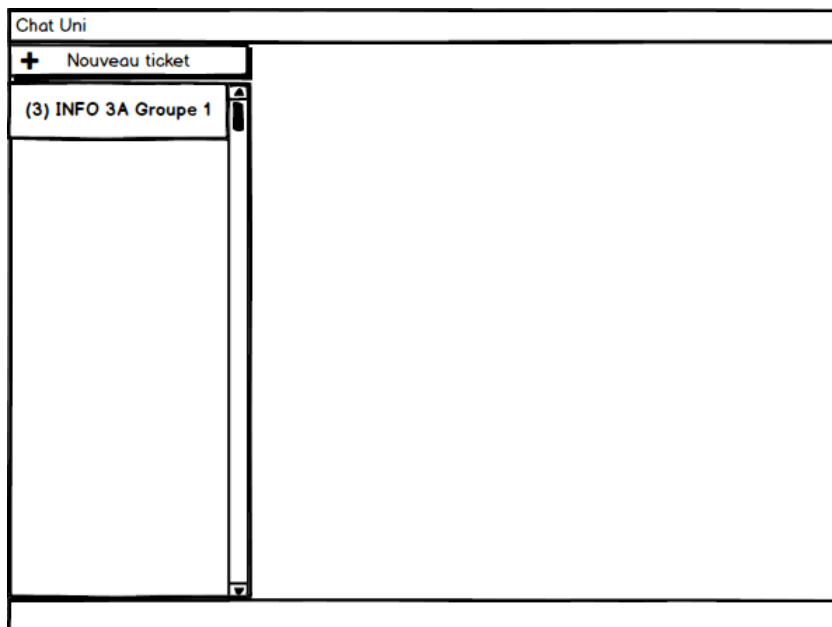
Se connecter

Ouverture de l'application :

Chat Uni

+ Nouveau ticket

(3) INFO 3A Groupe 1



Création d'un nouveau ticket :

Chat Uni

+ Nouveau ticket

(3) INFO 3A Gro

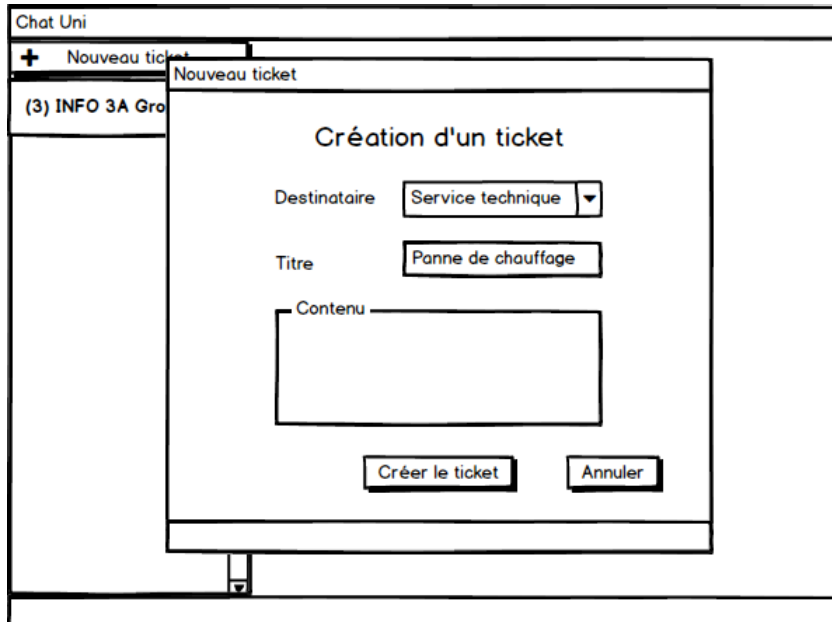
Nouveau ticket

Création d'un ticket

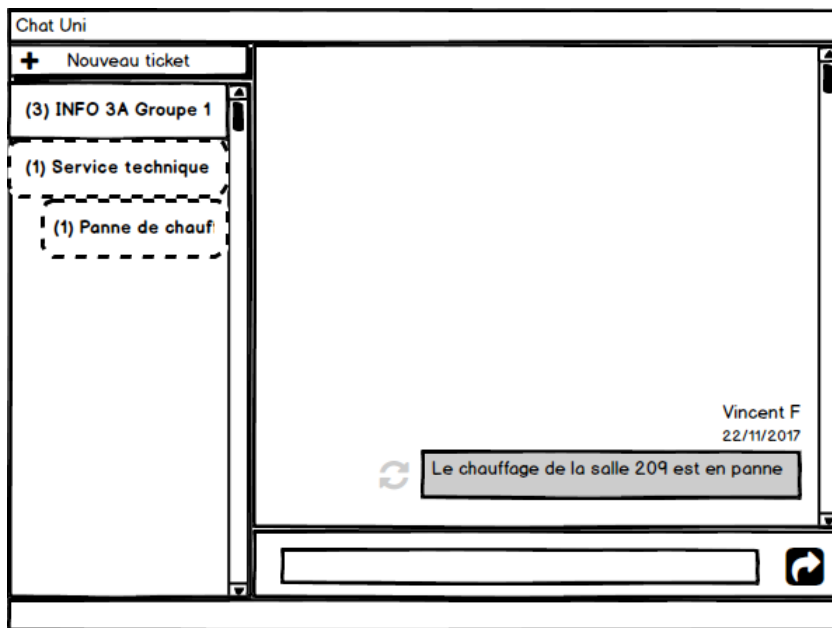
Destinataire

Titre

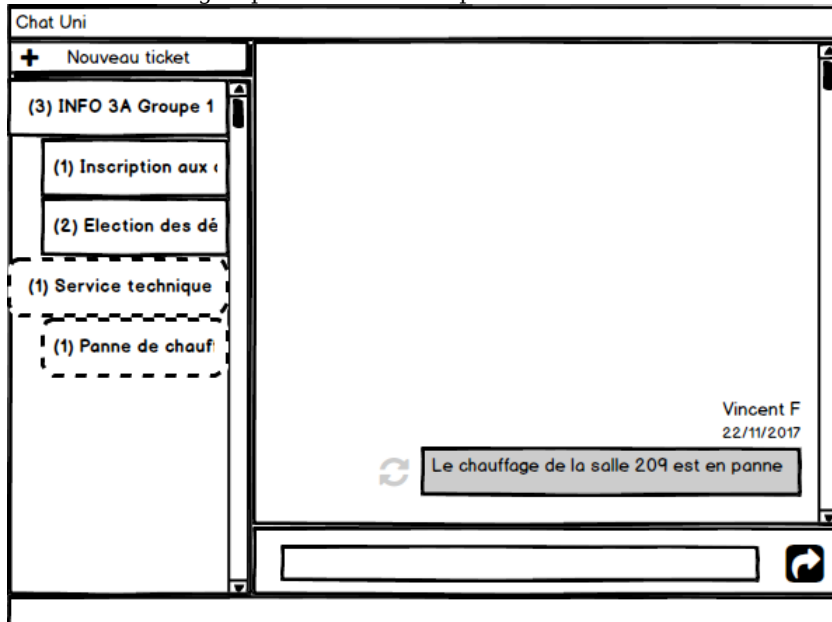
Contenu



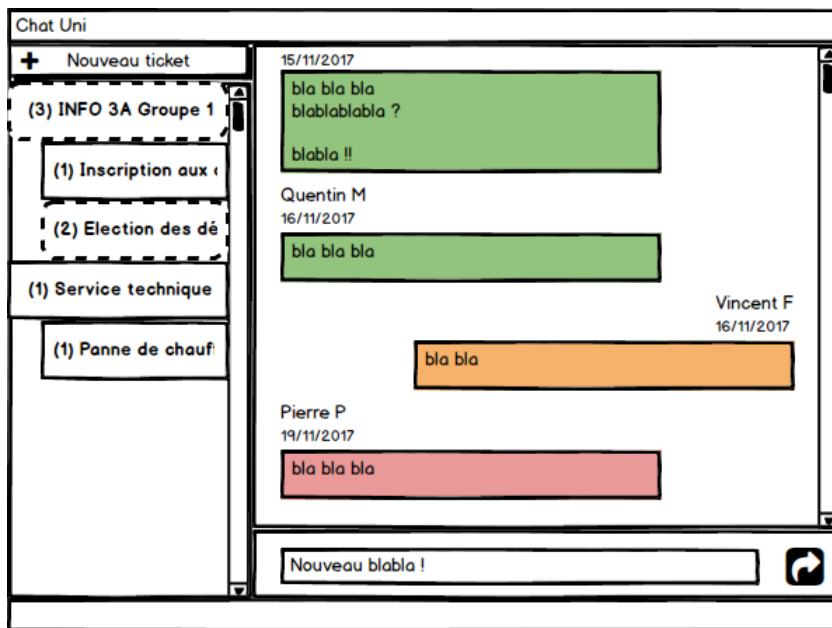
Après la création du ticket :



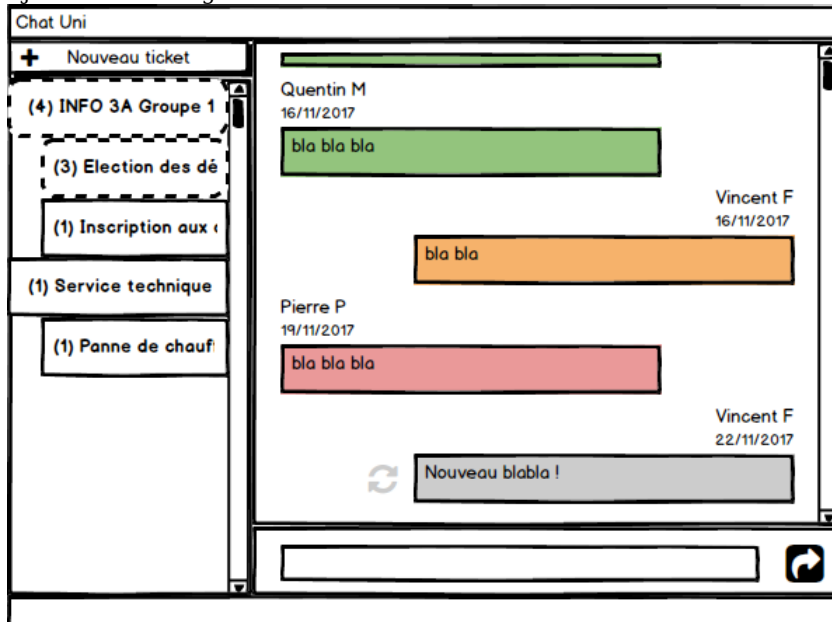
Déroulement du groupe "INFO 3A Groupe 1" :



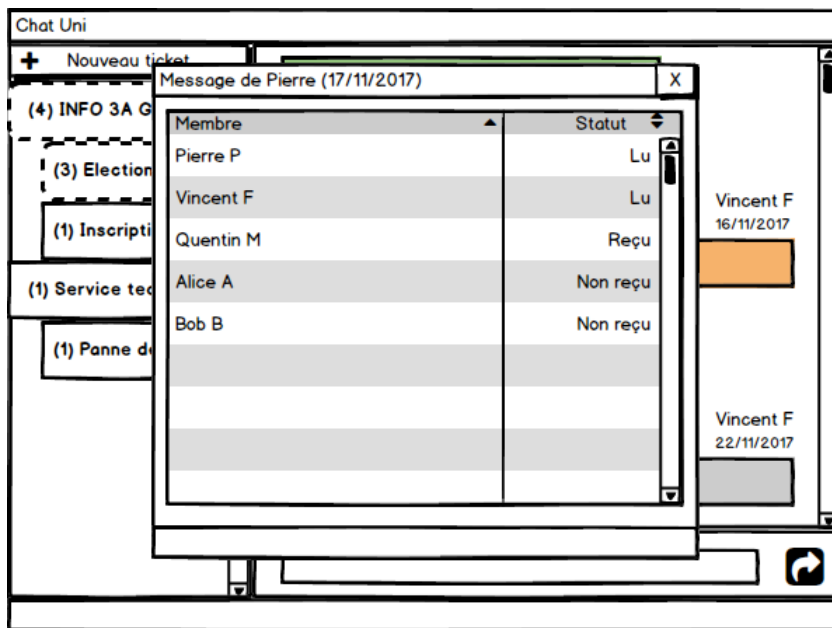
Ouverture d'un ticket contenant plusieurs messages :



Ajout d'un message sur ce ticket :



Détails d'un message :



3.4. IHM et Controleur de l'interface administrateur

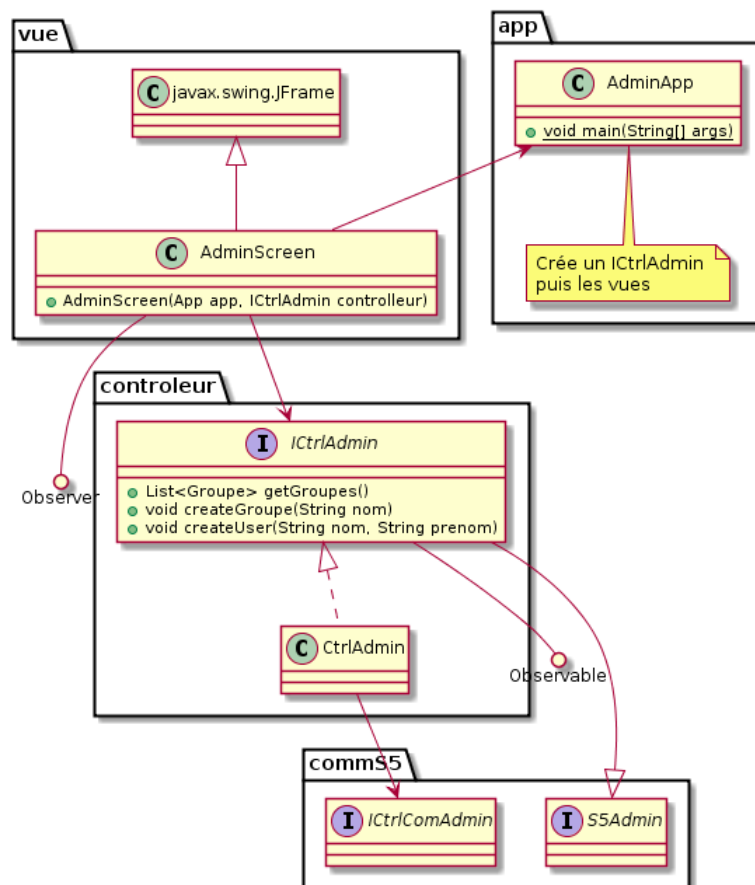


Diagramme de classe :

Maquettes de l'IHM : Onglet des groupes :

Administration

Groupes

Utilisateurs

INFO 3A Gpe 1.1
INFO 3A Gpe 1.2
INFO 3A Promo
Service technique
+ Nouveau groupe

Onglet des utilisateurs :

Administration

Groupes

Utilisateurs

Pierre P
Vincent F
Quentin M
Alice A
Bob B
+ Nouvel utilisateur

Détails d'un groupe :

Administration

Groupes

Utilisateurs

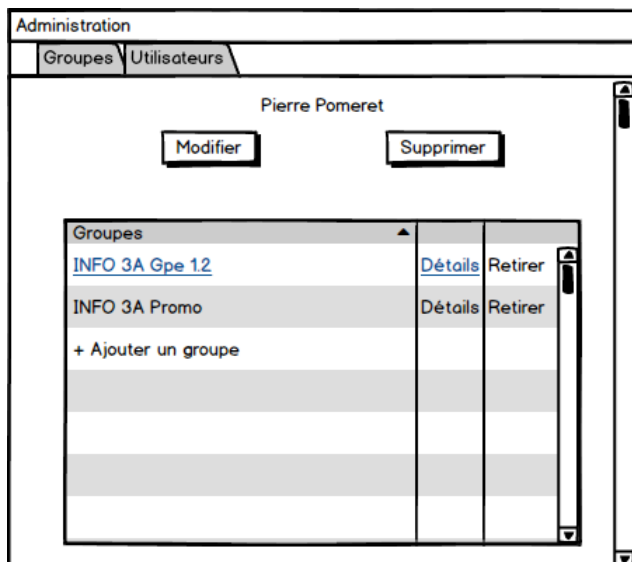
Groupe INFO 3A Gpe 1.2

Renommer

Supprimer

Membres		
Pierre P	Détails	Retirer
Quentin M	Détails	Retirer
Alice A	Détails	Retirer
Bob B	Détails	Retirer
+ Ajouter un utilisateur		

Détails d'un utilisateur :



4. Calendrier

Semaine 47

- Définition modèle de données
- Définition com réseau
- Définition base de donnée
- Définition application client

Semaine 48

- Implémentation com réseau générique
- Implémentation base de données
- Application client : vues

Semaine 49

- Dossier de conception complet
- Implémentation com réseau spécifique simple
- Application client : contrôleur

Semaine 50

- Fermeture du dépôt : dossier de conception (lundi 11/12)
- Com réseau : reprise sur incident
- Serveur : v0

Semaine 51

- Examens

Semaine 52

- Vacances

Semaine 1

- Vacances
- Serveur
- Application admin

Semaine 2

- Finitions
- Relectures et fioritures

Semaine 3

- Soutenance (mercredi 17/01)

Planning

