# Introduction

L’objectif de ce TP de Java final était de mettre en application nos différentes connaissances et de travailler sur un projet concret. Il nous a permis d’utiliser des connaissances en bases de données, en programmation orientée objet et en réalisation d’interface graphique.

Ce rapport a pour but de présenter les différentes fonctionnalités du programme et les axes d’amélioration de celles-ci.

Table des matières

[Introduction 1](#_Toc471371240)

[Présentation 2](#_Toc471371241)

[Résumé : 4](#_Toc471371242)

[Partie 1 : Conception 4](#_Toc471371243)

[Package et interface 4](#_Toc471371244)

[Classes abstraites et internes 4](#_Toc471371245)

[Exceptions 4](#_Toc471371246)

[Partie 2 : Persistance 5](#_Toc471371247)

[Fichier 5](#_Toc471371248)

[Base de données 5](#_Toc471371249)

[Scénario de suppression : 6](#_Toc471371250)

[Partie 3 : Interface graphique 9](#_Toc471371251)

[Affichage d’un tableau de vol 9](#_Toc471371252)

[Interface graphique pour mettre à jour les données de vols par un des acteurs 10](#_Toc471371253)

[Cas d’utilisation : Connexion d’un utilisateur 10](#_Toc471371254)

[Cas d’utilisation : un admin veut créer un avion et tout son personnel. 13](#_Toc471371255)

[Cas d’utilisation : Un manager veut afficher un certain vol et assigné des personnes à ces vols 16](#_Toc471371256)

[Cas d’utilisation : Un membre d’équipage veut afficher ses assignations 17](#_Toc471371257)

# Présentation

Dans la vue principale quatre onglets sont visibles :

* Admin : il a la possibilité de créer un vol, de supprimer un vol, de créer un avion et de supprimer un avion
* Manager : il a la possibilité de rajouter un membre d’équipage à un vol, de supprimer un membre d’équipage d’un vol et d’afficher les données d’un vol
* Membre d’équipage : il permet avec le nom du membre d’équipage de trouver sur quels vols il intervient
* Afficher tableau : il permet d’afficher un récapitulatif de toutes les assignations de tout le personnel

# Résumé :

## Partie 1 : Conception

### Package et interface

Il y a 3 packages : DataBase, Modèle, Vue et 2 sous packages dans le package Vue : FormulaireAdmin et FormulaireManager.

Il y a une interface : InterfaceAeroports

### Classes abstraites et internes

Il y a une classe abstraite : MembreEquipage

Il y a une classe internes : Dans la classe Aéroports il y a la classe Size.

### Exceptions

Il y l’exception EquipageException qui relève plusieurs erreurs :

* Si un membre d’équipage à plus de deux qualifications
* Si une position est déjà affectée
* Si le nombre max de PNC sur un vol est atteint
* Si un membre d’équipage n’est pas qualifié

Une exception InvariantBroken qui vérifie si les paramètres sont valides

## Partie 2 : Persistance

### Fichier

Les aéroports qui sont représenté par un tableau de String dans la classe Aeroports implémente Serializable. Il est enregistré dans le fichier aeroports.txt.

### Base de données

Nous avons une base de données MySql qui fonctionne en local. La base se nomme « betterairmanager » et peut-être crée grâce au script betterairmanager.sql dans le package DataBase.

Il y a 6 tables :

* Assocpncequipage : C’est une table d’association entre equipage et membreequipage
  + Varchar refEquipage, qui est la clé dans la table equipage
  + Varchar nompnc, qui est la clé dans la table membreequipage
* Avion : les informations des avions
  + Varchar TypeAvion, qui est la clé dans la table typeavion
  + Varchar Ref, qui est la référence de l’avion
* Equipage, liée à la table vol
  + Varchar Pilot, qui fait référence à un nom dans la table membreequipage
  + Varchar Copilot, qui fait référence à un nom dans la table membreequipage
  + Varchar volRefNpc, qui est la clé dans la table d’association assocpncequipage
* Membreequipage, les informations des membres d’équipage
  + Varchar Nom, nom du membre d’équipage
  + Varchar Prenom, prenom membre d’équipage
  + Varchar Metier, metier membre d’équipage
  + Varchar qualif1, qualification1 d’un membre d’équipage
  + Varchar qualif2, qualification2 d’un membre d’équipage
* Typeavion, les informations d’un type d’avion
  + Varchar nom, nom du type d’avion
  + Int nbPNCmin, nombre de PNC minimum dans ce type d’avion
  + Int nbPNCmax, nombre de PNC maximum dans ce type d’avion
* Vol, les informations d’un vol
  + Varchar numvol, clé primaire et défini un vol et un équipage
  + Varchar site, site de départ du vol
  + Varchar dest, site de destination du vol
  + Date date, date de départ du vol
  + Varchar avion, référence du l’avion utilisé par le vol

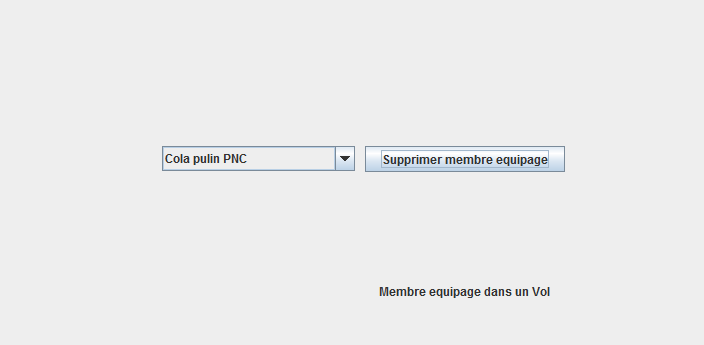
A chaque manipulation de la base de données les interfaces sont mis à jour.

L’intégrité de la base de donnes est préserver à chaque ajout et suppression.

## Scénario de suppression :

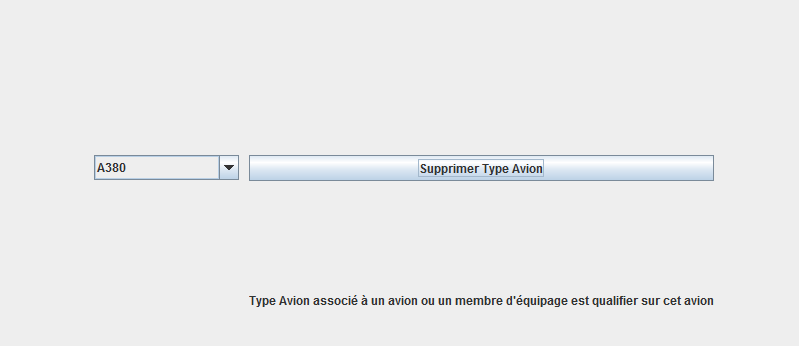
Membre d’équipage

S’il y a une suppression d’un membre d’équipage il y a d’abord une vérification si ce membre d’équipage ne fait pas partie d’un vol. S’il en fait partie la suppression n’a pas lieu et un message d’erreur apparait.



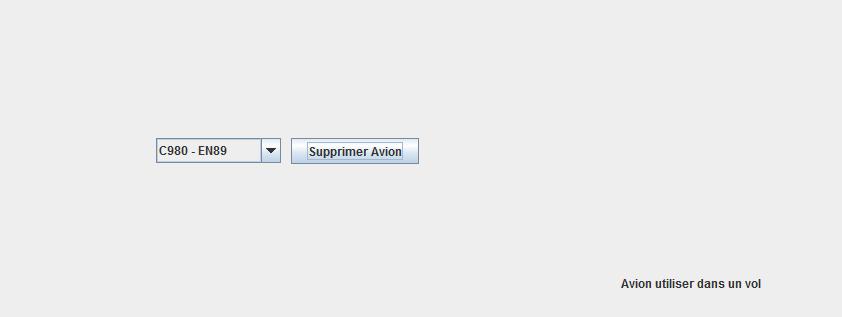
Type Avion

Pour supprimer un type avion il faut que ce type ne soit pas utiliser pour un avion. Si c’est le cas la suppression n’a pas lieu et un message d’erreur apparait



Avion

Pour supprimer un avion, il faut que cet avion ne soit pas utilisé dans un vol. Si c’est le cas la suppression n’a pas lieu et un message d’erreur apparait



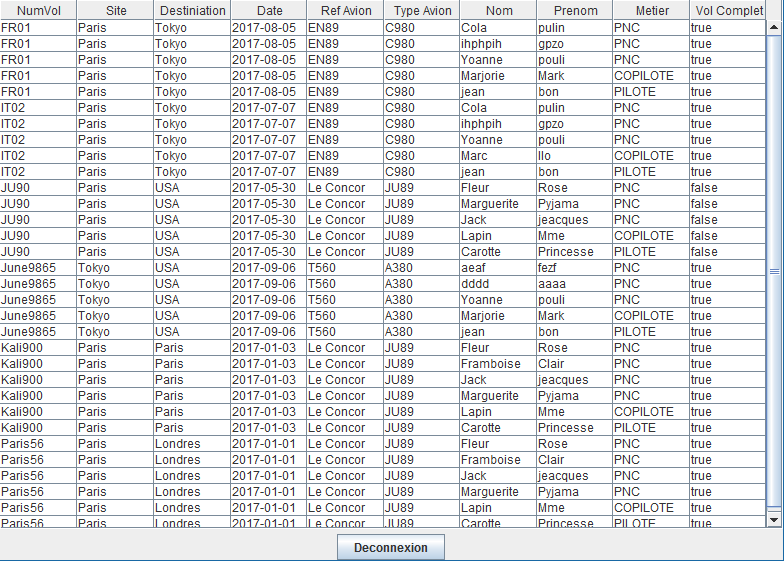
Vol

Il n’y a pas de vérification lors de la suppression d’un vol car elle n’impact l’intégrité de la base de données.

## Partie 3 : Interface graphique

### Affichage d’un tableau de vol

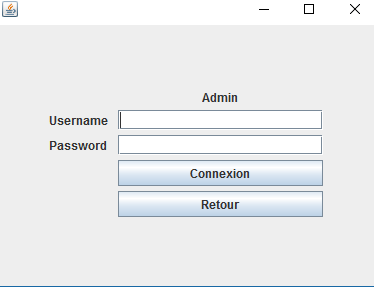




### Interface graphique pour mettre à jour les données de vols par un des acteurs

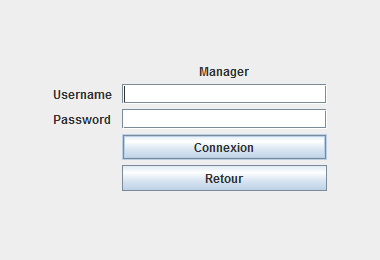
### Cas d’utilisation : Connexion d’un utilisateur

#### Connexion d’un admin (username : admin password : admin)



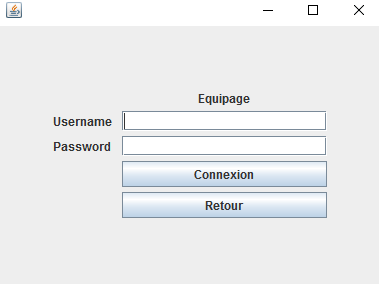
#### Connexion d’un manager (Username : manager Password : manager)





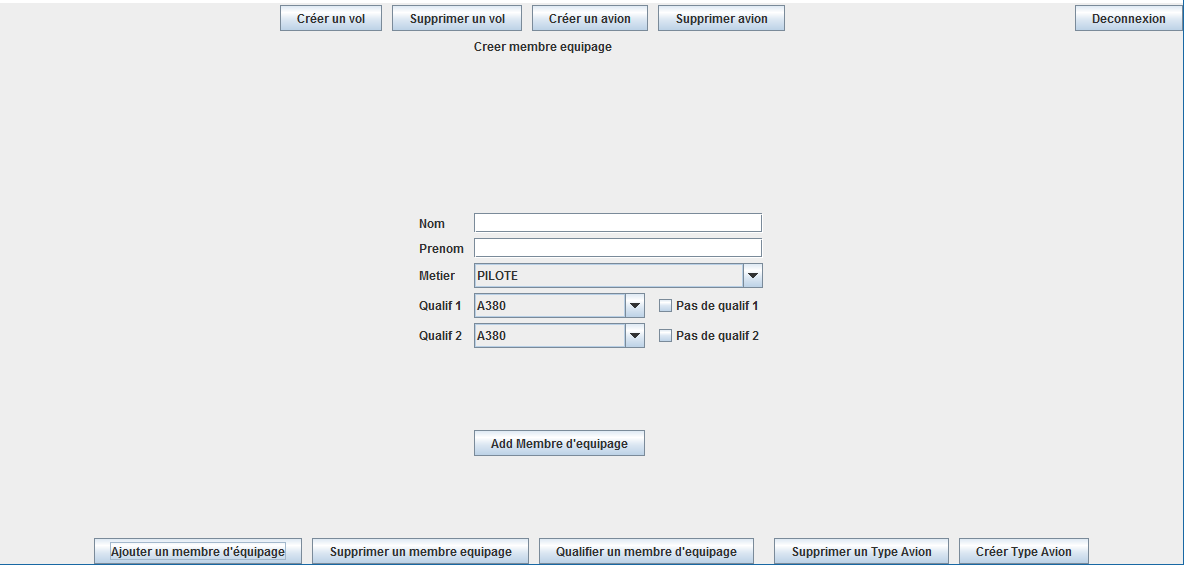
#### Connexion d’un utilisateur (Username : nom d’un membre d’équipage)



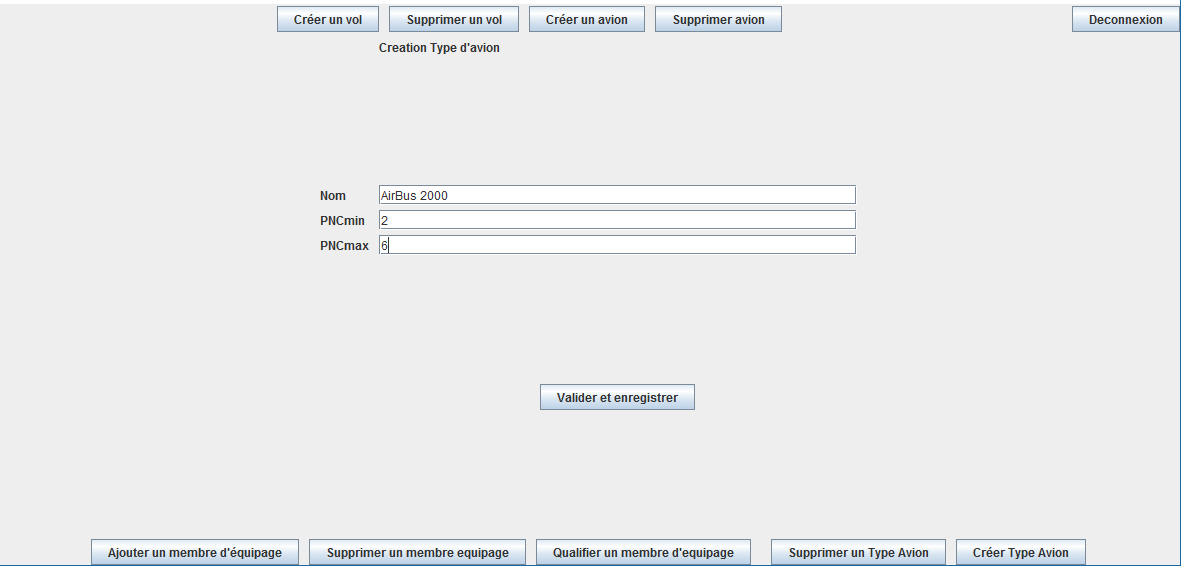


### Cas d’utilisation : un admin veut créer un avion et tout son personnel.

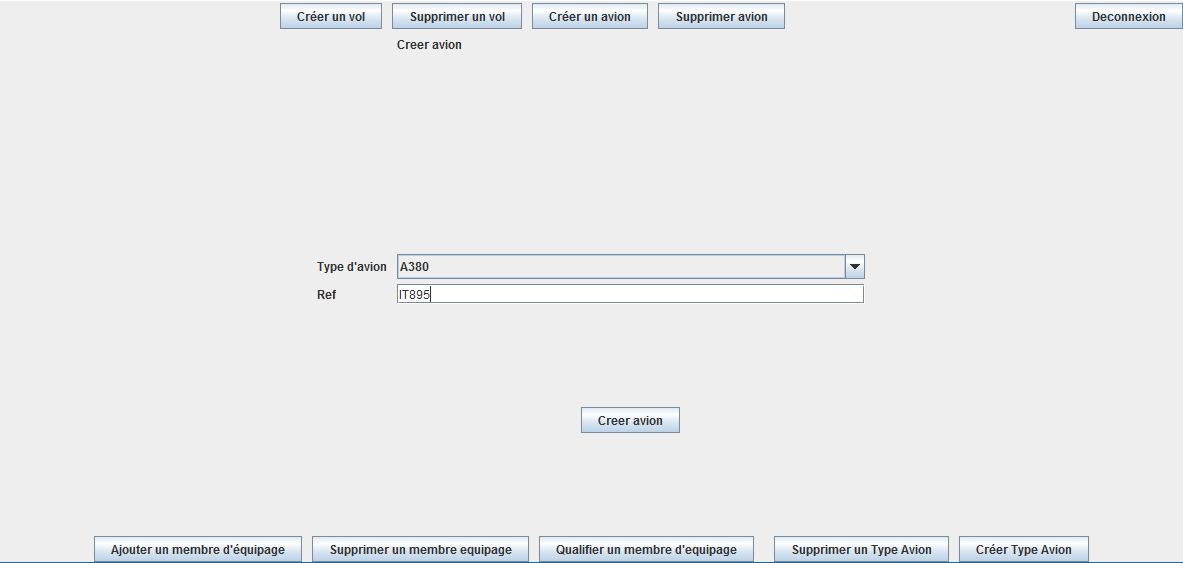
#### Créer des membres d’équipages (nom, prénom, métier, qualification)



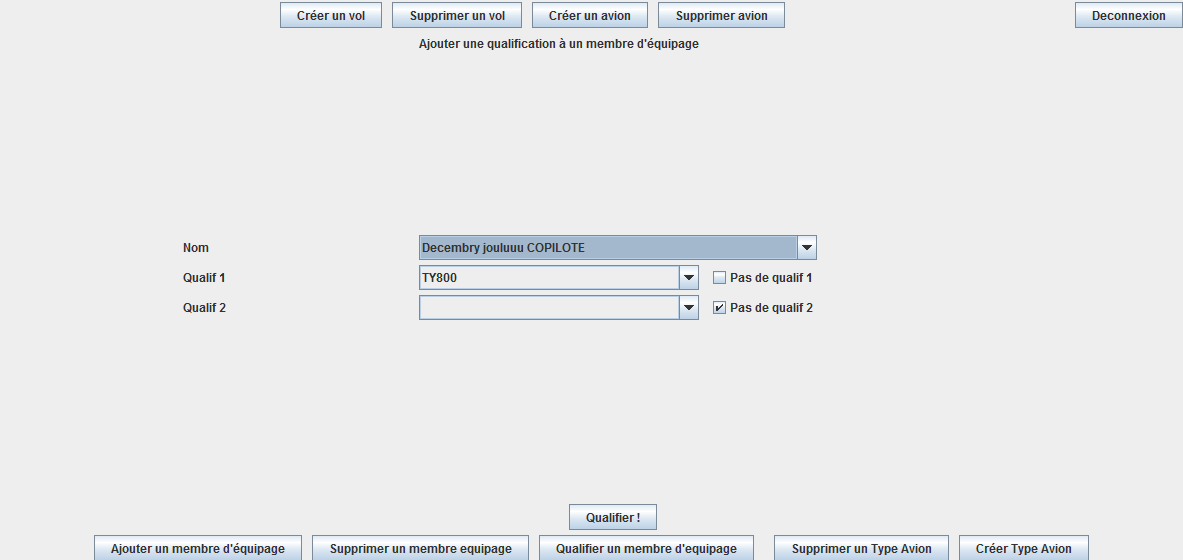
#### Créer un type d’avion (nom, PNC minimum, PNC maximum)



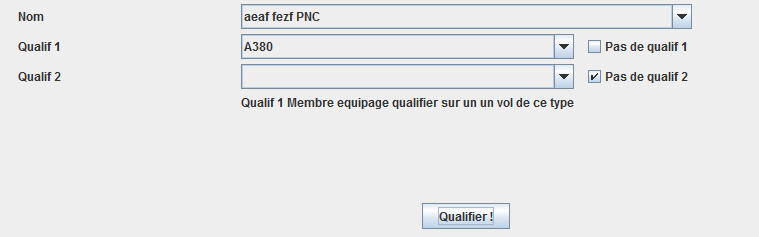
#### Créer un avion (Type, référence de l’avion)



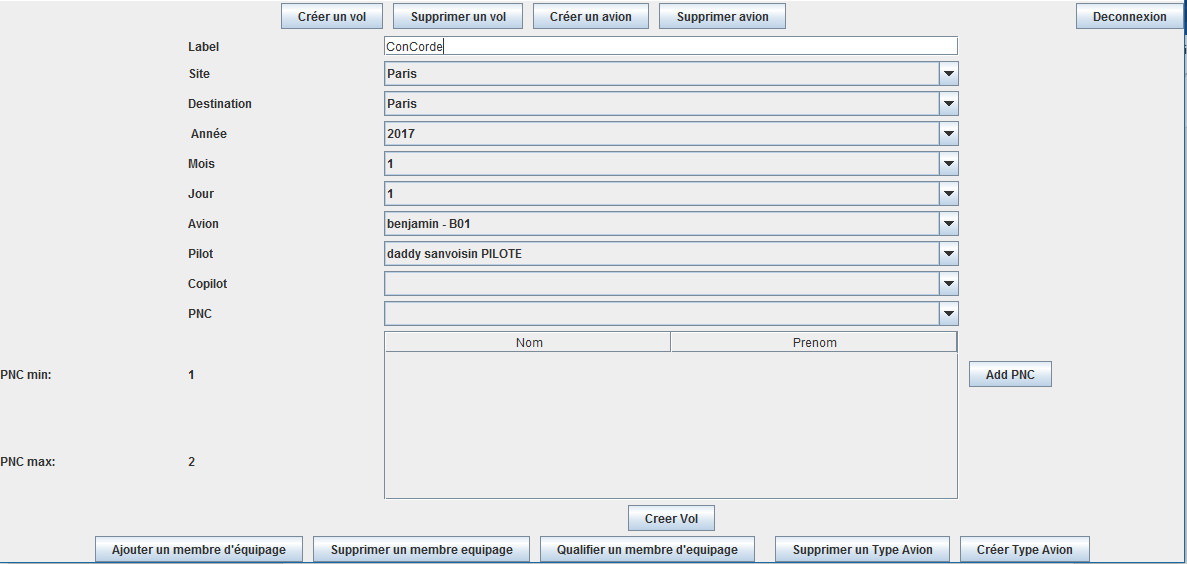
#### Qualifier les membres d’équipage pour ce type d’avion (Nom, qualification1, qualification2)



Il y a une vérification lors de la modification d’une qualification, si le membre d’équipage fait partie d’un équipage qui requière cette qualification alors on ne peut pas la modifier.

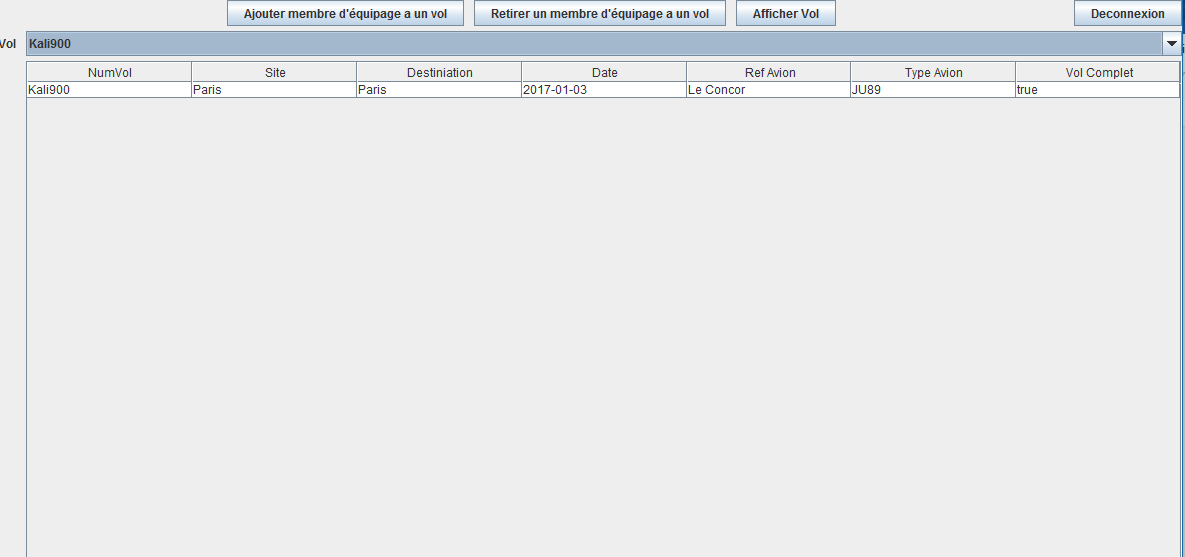


#### Créer un vol (nom du vol, départ, destination, date, type d’avion, pilote, copilote, PNC)

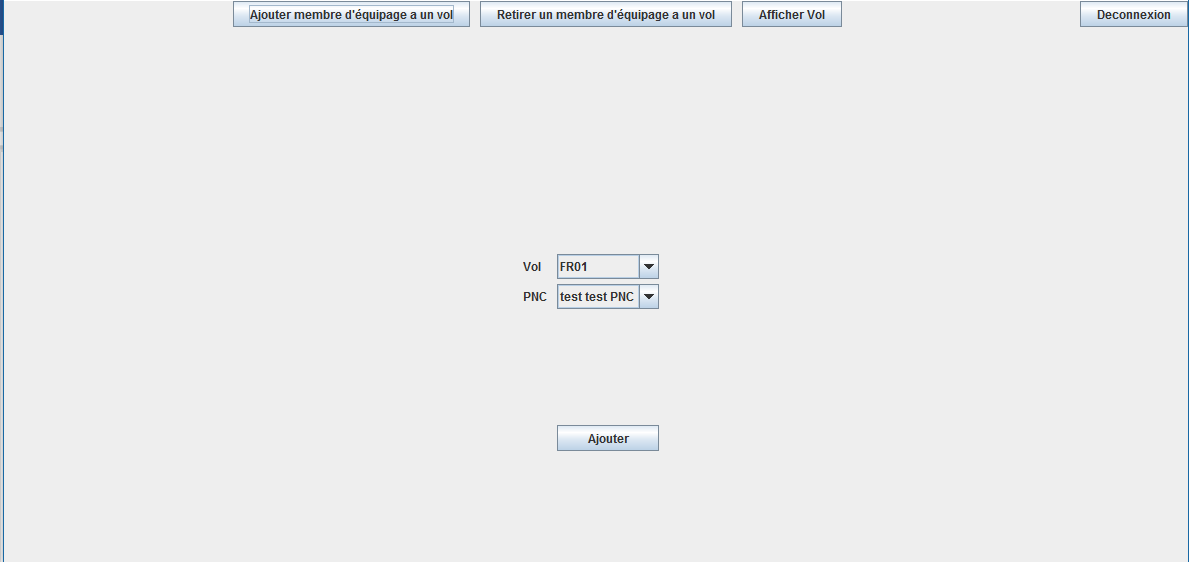


### Cas d’utilisation : Un manager veut afficher un certain vol et assigné des personnes à ces vols

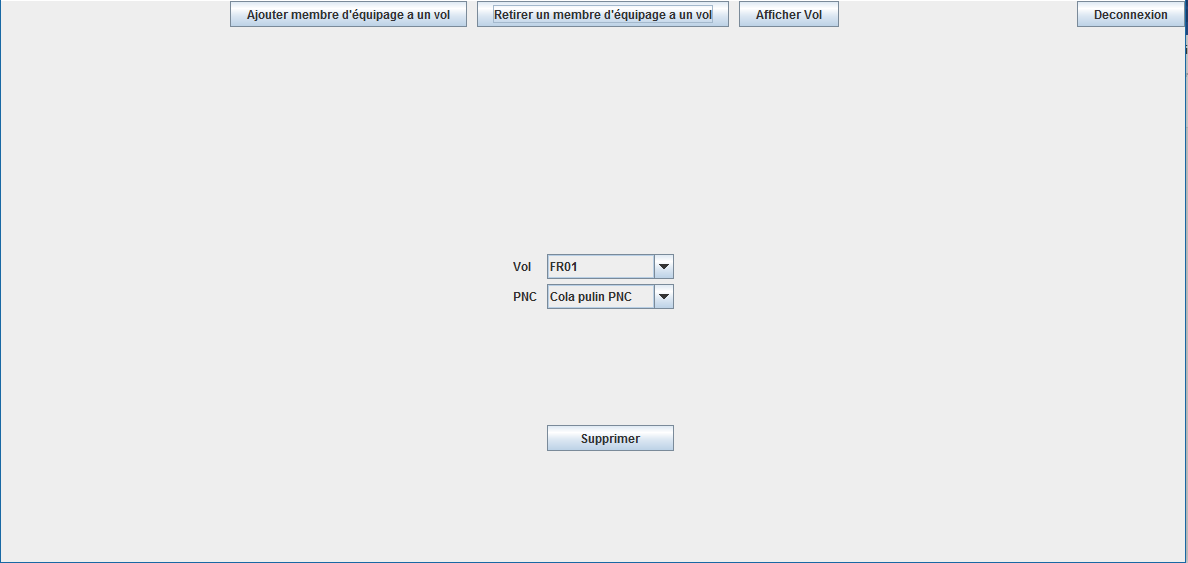
#### Manager – Afficher vol (sélectionner le vol)



#### Ajouter un membre à un équipage



#### Retirer un membre à un équipage



### Cas d’utilisation : Un membre d’équipage veut afficher ses assignations

#### MembreEquipage (Voir connexion membre équipage pour sélectionner une personne, ici Cola)

