

Projet : LOGICIEL DE PARTAGE

Introduction

Ce projet consiste à réaliser une application de partage. Elle doit être capable de créer un lien entre les étudiants pour qu'ils puissent s'échanger leurs calculatrices.

L'administrateur crée les comptes de tous les utilisateurs. Il est nécessaire que l'utilisateur se connecte (identifiant, mot de passe) pour avoir accès aux différentes parties de l'application.

L'utilisateur a la possibilité de changer son mot de passe mais pas son nom d'utilisateur.

Supports utilisés :

- **Visual Paradigm 15.2** : pour les diagrammes UML
- **Eclipse** : Pour l'implémentation en JAVA
- **MySQL** : Pour stocker les informations
- **Mysql-connector-java.-5.1.44.jar** : pour établir la connexion avec la base de données.

1. Diagramme de cas d'utilisation

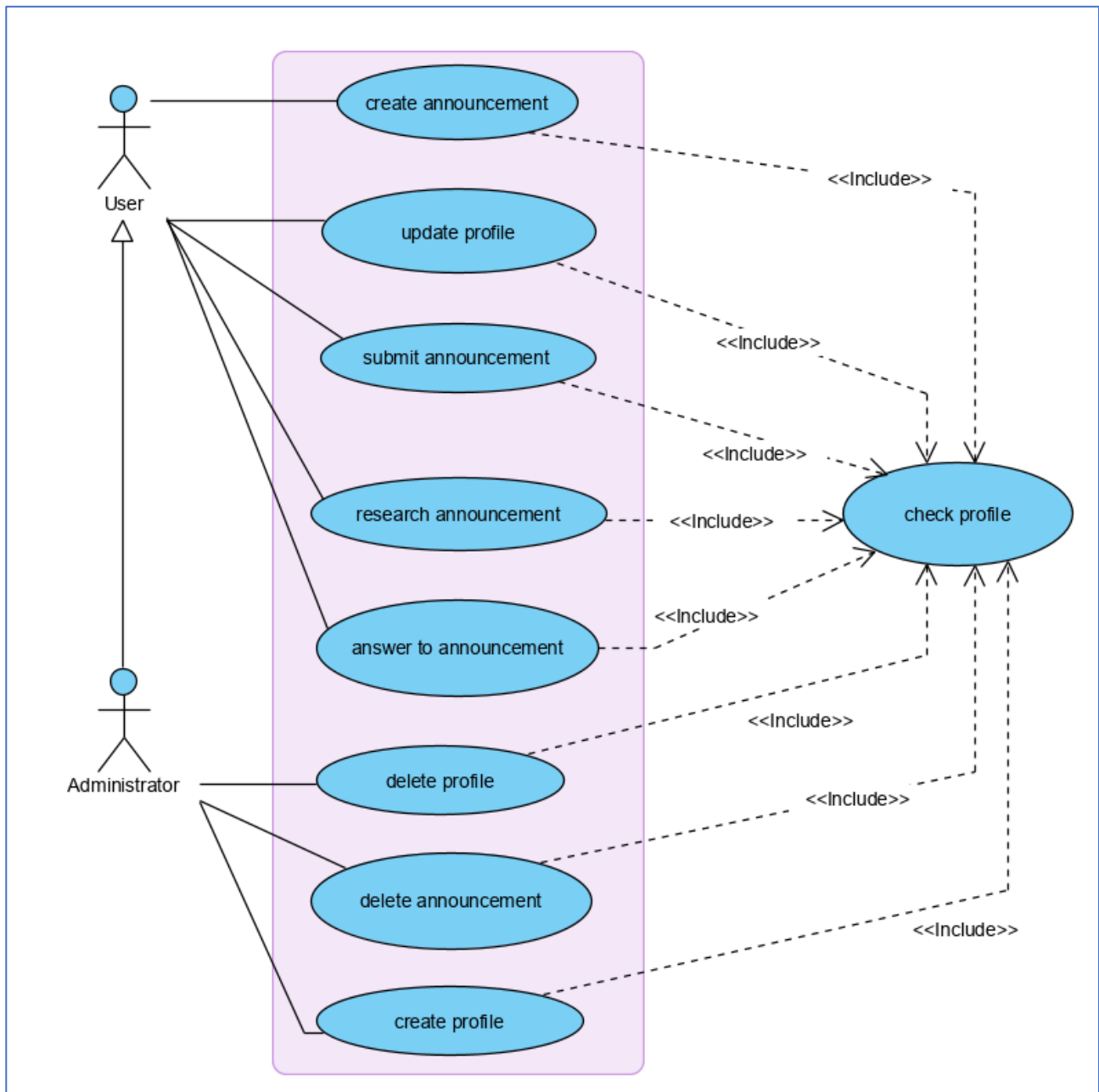


Fig.1 : Diagramme du cas d'utilisation.

Ici, l'administrateur est juste un utilisateur avec plus de rôles, d'où le fait qu'il hérite de « User ». Seul l'administrateur peut créer les différents profils, supprimer les profils et les annonces. Le profil de l'utilisateur doit être vérifié avant toute action de la part de ce dernier.

2. Diagramme de classe

Pour assurer la mise en place du logiciel de partage, il nous faut intervenir un système de gestion des annonces et un système de gestion des utilisateurs afin de pouvoir suivre le partage des informations.

Le diagramme de classe ci-dessous [Fig.2] est un patron **façade**.

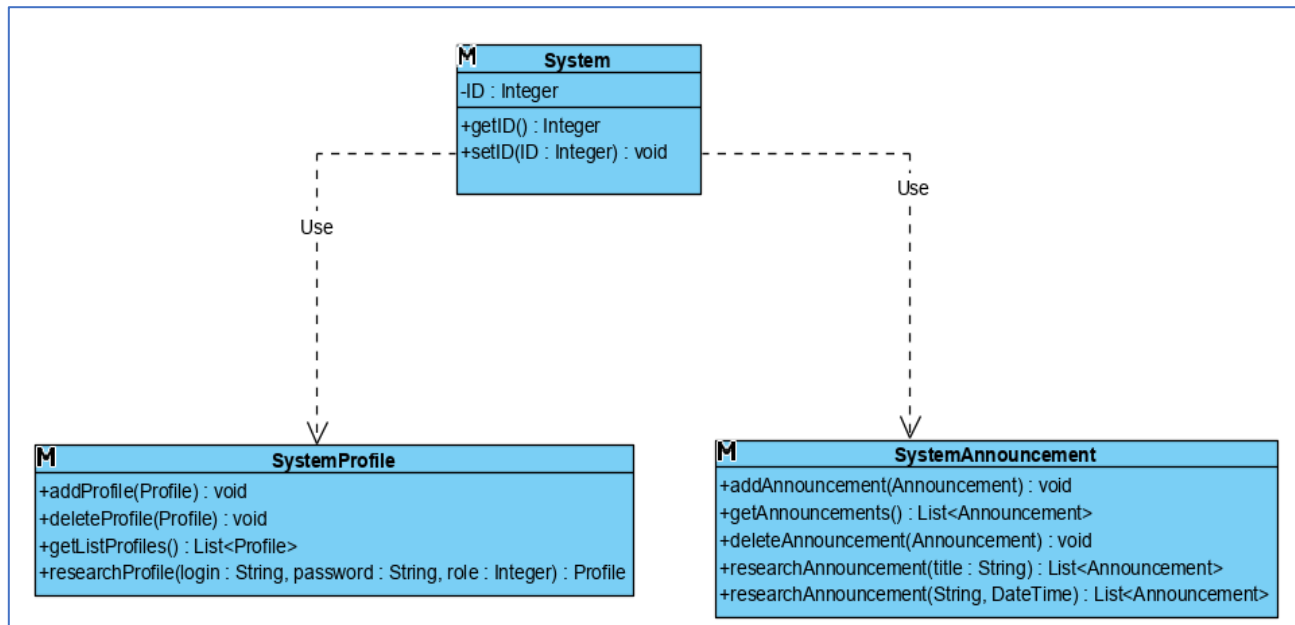


Fig.2 : Patron Façade.

Tous les utilisateurs ont un profil stocké dans le système de profil afin de permettre à l'administrateur de pouvoir contrôler le flux d'information. De même, les annonces sont stockées dans le système des annonces.

Lorsqu'on a besoin des informations sur un profil ou un utilisateur, le système central délègue la tâche au système de profil.

De même, lorsque nous avons besoin de manipuler les annonces, le système central délègue la tâche au système des annonces.

Le diagramme suivant [Fig.3] est un patron factory. On utilise un « patron factory » afin de différencier les utilisateurs de l'administrateur.

L'utilisateur et l'administrateur ont des méthodes en commun. Cependant, l'administrateur a des méthodes que l'utilisateur ne peut pas exécuter.

Aussi, certaines actions de l'utilisateur nécessitent l'autorisation de l'administrateur comme la suppression des annonces et du profil.

D'où l'apparition de la classe abstraite « **Profile** », contenant la méthode abstraite « **updateAnnouncement()** ».

Les méthodes propres aux utilisateurs sont définies dans l'interface « **UserMethods** » et celles de l'administrateur dans l'interface « **AdminMethods** ».

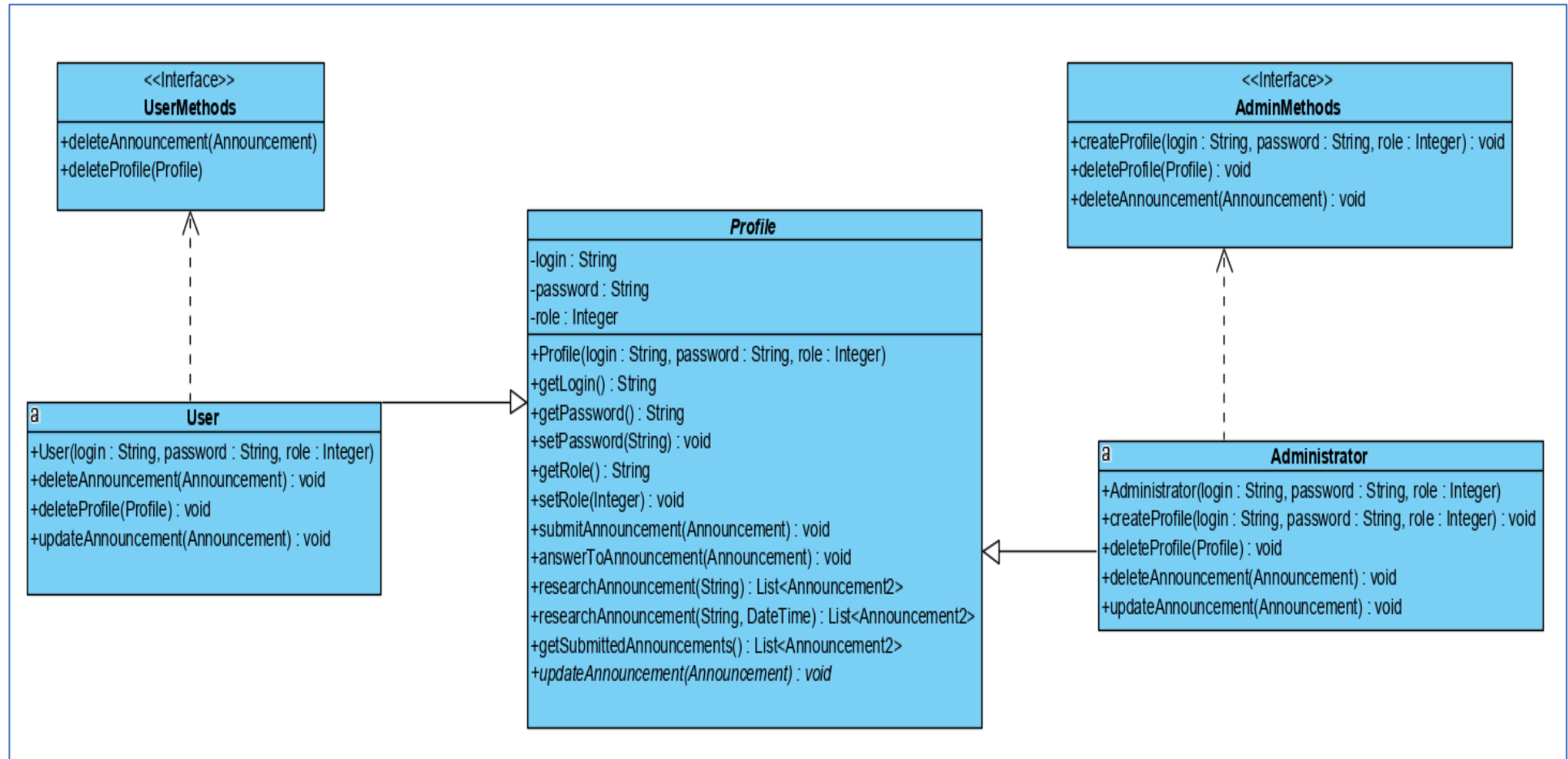


Fig.3. Patron Factory.

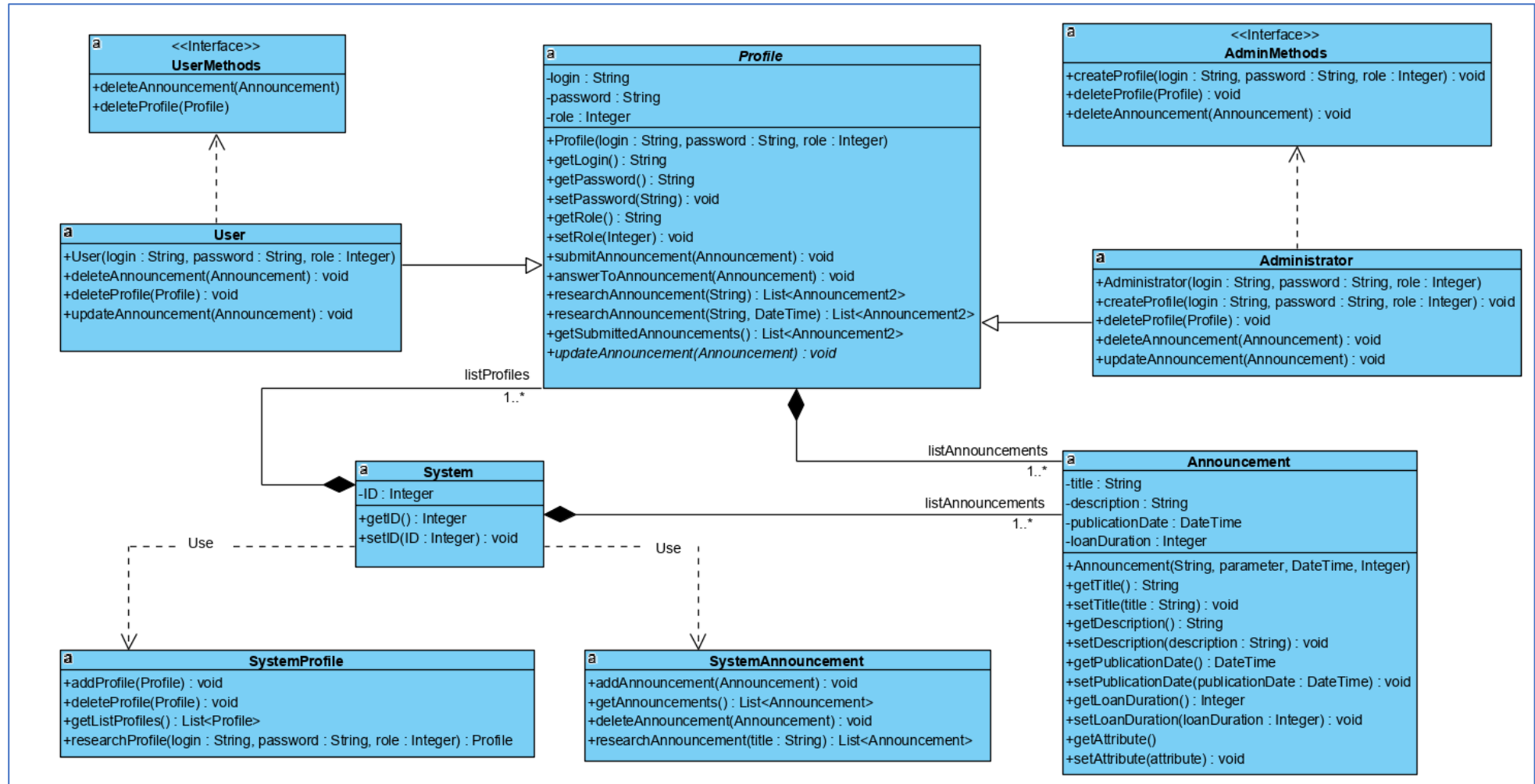


Fig.4. Diagramme de classe du projet.

3. Diagramme de séquence

Les diagrammes de séquences ci-dessous ne sont que des diagrammes approximatifs du fonctionnement du logiciel de partage.

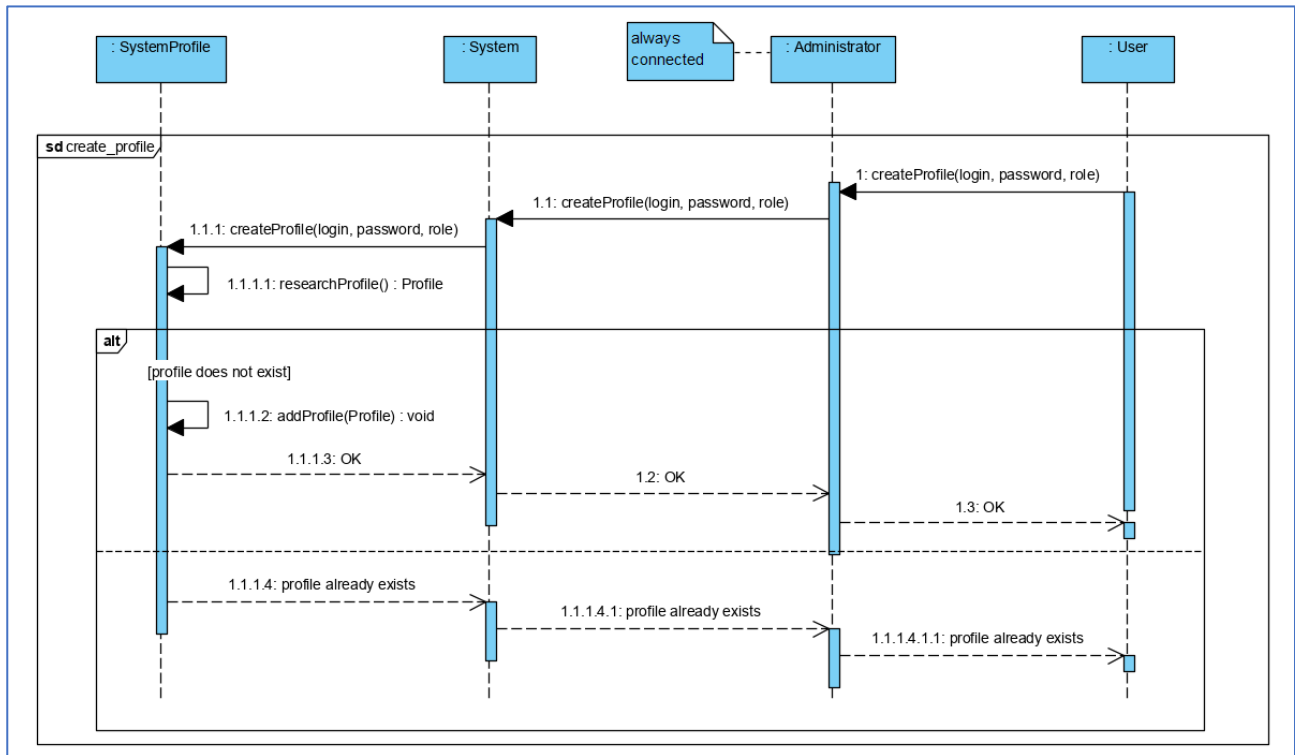


Fig.5 : Création d'un profil.

L'utilisateur demande à l'administrateur de créer son profil, l'administrateur quant à lui, transmet la demande au système central. Le système central demande au système des profils de créer le profil de l'utilisateur. S'il n'y a pas de profil correspondant à l'utilisateur le système profil en crée un et ajoute ce dernier à sa liste de profil. Le système central sera notifié de l'opération et préviendra l'administrateur de la création du profil. Si le profil existe déjà, il envoie une notification à au système centrale qui relaye l'information.

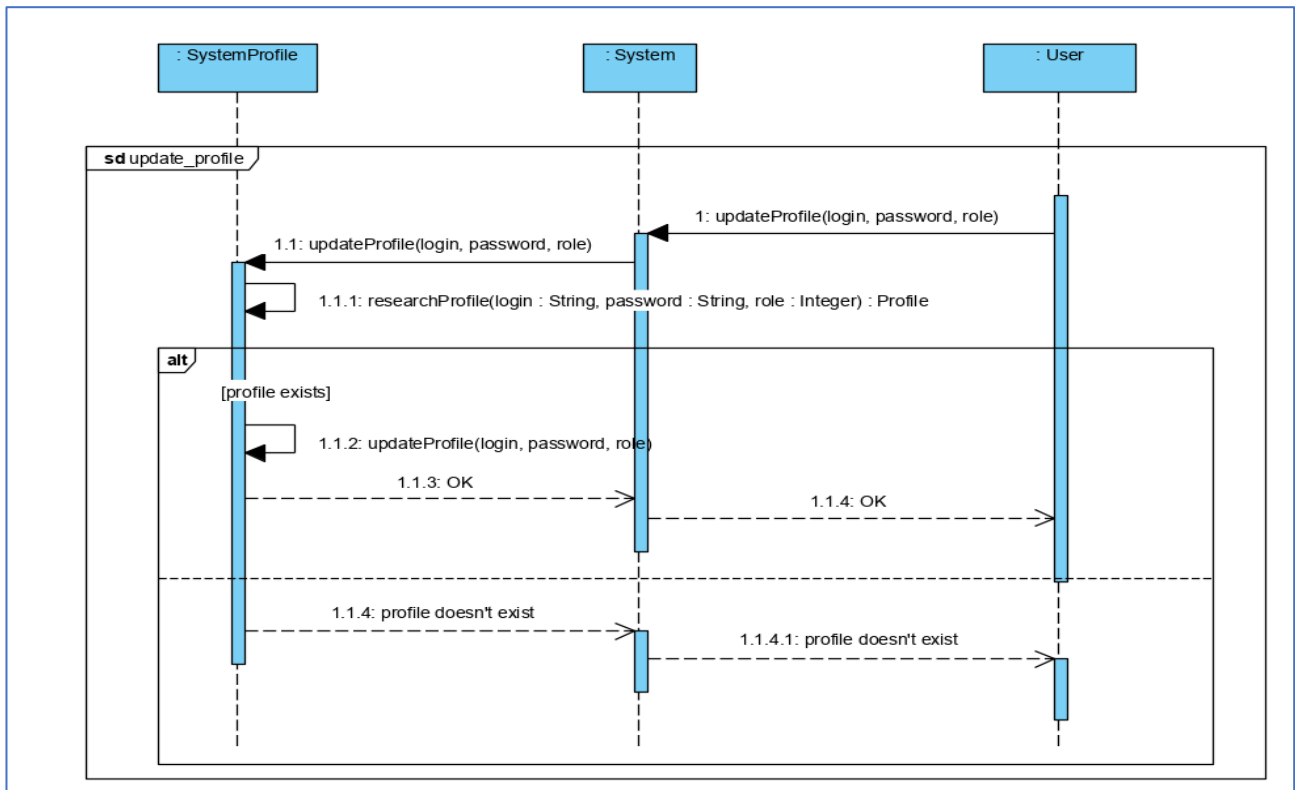


Fig. 6 : Modification d'un profil.

L'utilisateur demande au système central de modifier son profil, le système des profils est informé de cette demande par le système central. Le système des profils vérifie si l'utilisateur a un profil, si tel est le cas il effectue la modification et envoie une notification au système centrale. Le système central confirmera à l'utilisateur que son profil a été modifié.

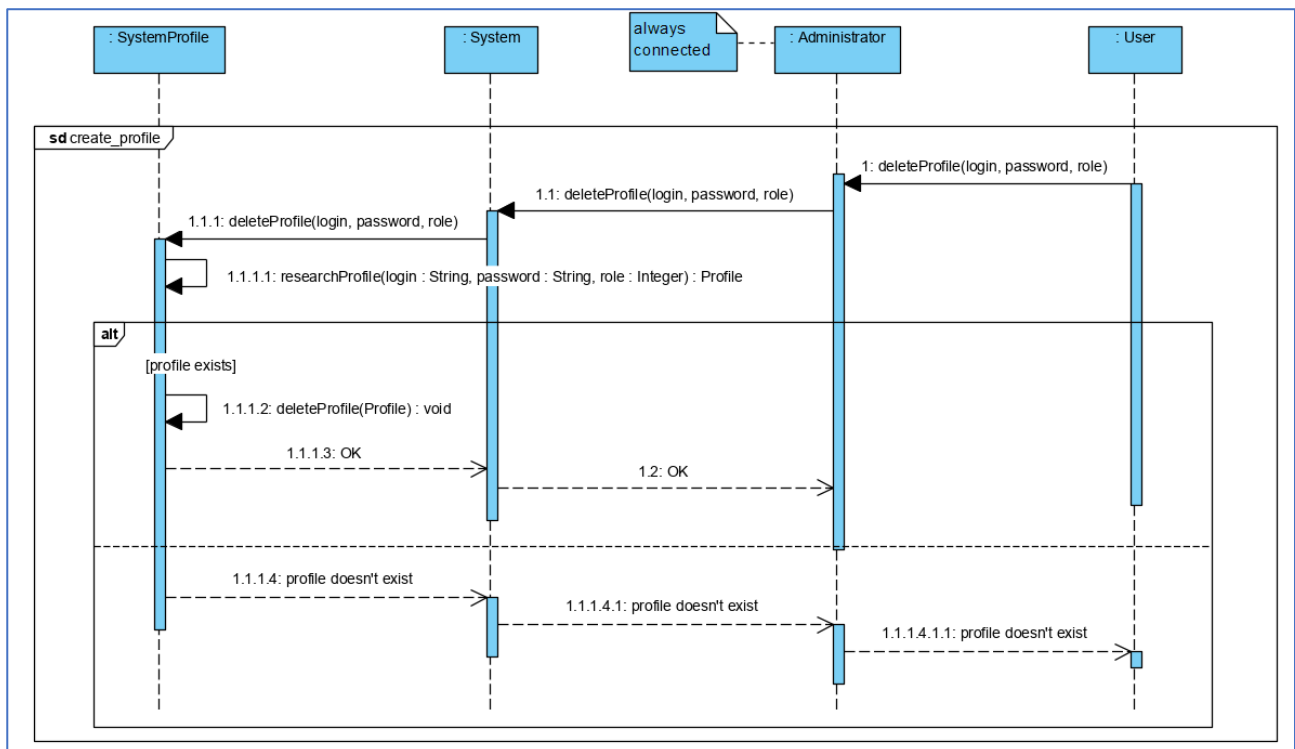


Fig. 7 : Suppression d'un profil.

L'utilisateur demande à l'administrateur de supprimer son profil. L'administrateur transmet cette demande au système central qui à son tour relaie la tâche au système des profils. Le système des profils vérifie que l'utilisateur possède un profil, si tel est le cas il retire le profil de sa liste de profils et notifie le système central de la suppression, le système central confirme la suppression de l'utilisateur à l'administrateur. Si l'utilisateur n'existe pas, le système profil notifiera l'information au système centrale. L'administrateur apprendra la nouvelle par le système central.

Cette opération est possible sans la demande de l'utilisateur.

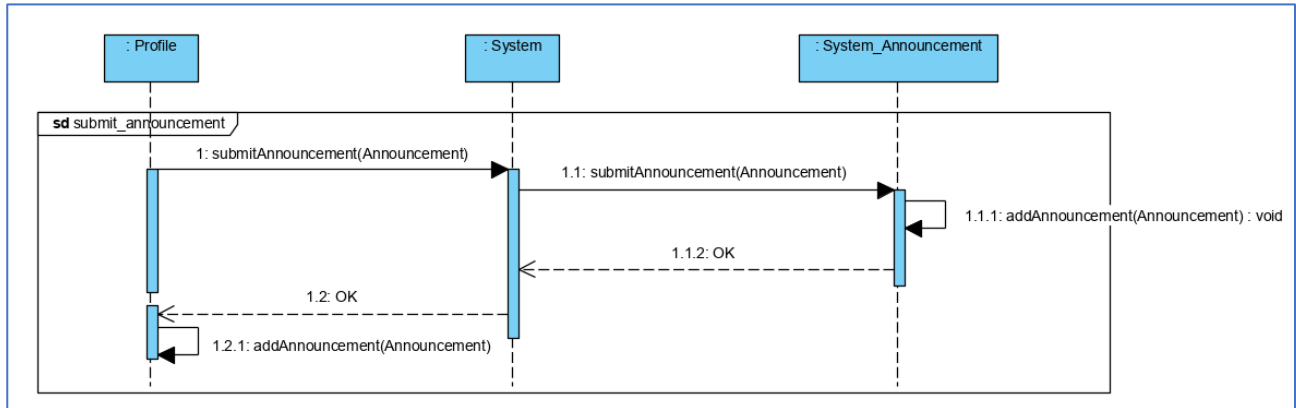


Fig.8 : Création d'une annonce.

L'utilisateur envoie une annonce au système centrale qui informe le système annonce du désir de l'utilisateur de créer une annonce. Le système annonce ajoute l'annonce à sa liste des annonces puis notifie le système central de la réussite de l'opération. L'utilisateur est notifié du succès de l'opération par le système central puis il ajoute l'annonce qu'il a créé à sa liste des annonces émises.

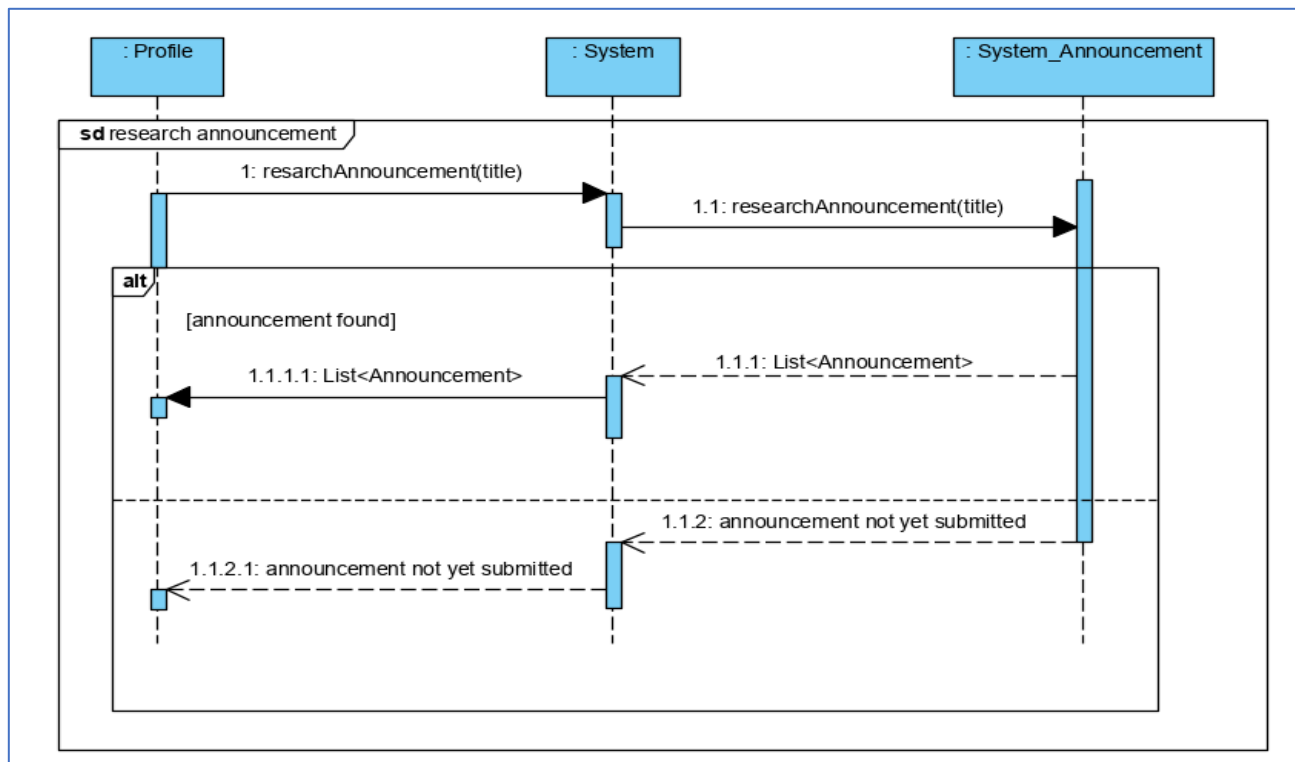


Fig.9 : Recherche d'une annonce par son titre.

L'utilisateur demande au système central qui demande au système des annonces de rechercher toutes les annonces dont les titres correspondent à sa requête, le système annonce renvoie au système centrale la liste de toutes les annonces correspondantes à la requête de l'utilisateur. L'utilisateur reçoit cette liste par le système centrale. S'il y'a aucun résultat, il renvoie une liste vide.

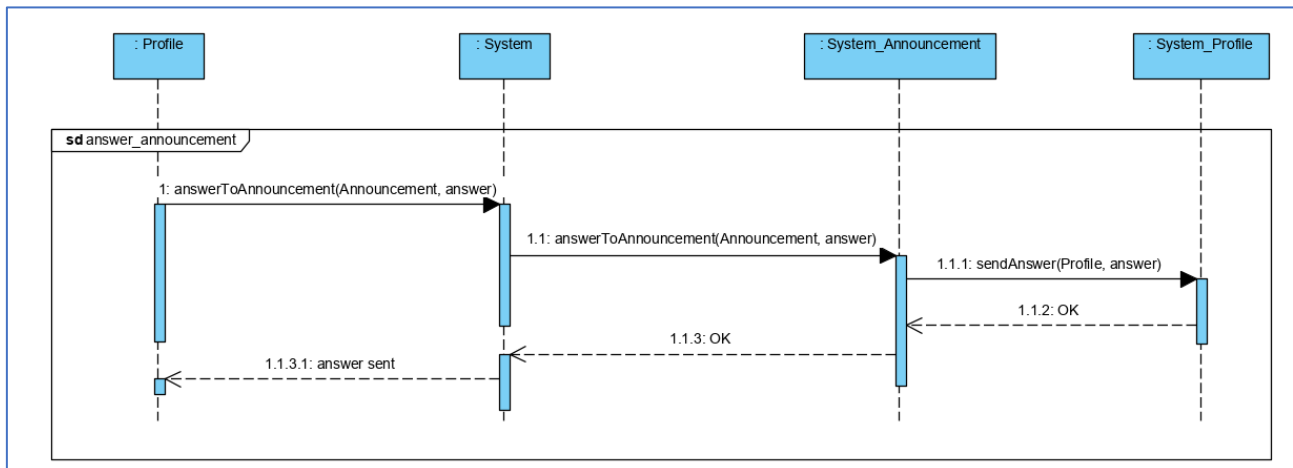


Fig.10 : Réponse à une annonce.

L'utilisateur demande au système central, qui lui-même demande au système des annonces de transmettre sa réponse à l'auteur de l'annonce. Le système des annonces quant à lui demande au système des profils de transmettre le message. Ce dernier transmet le message et notifie le système des annonces qui à son tour notifie le système central avant d'informer l'utilisateur.

Enfin, l'utilisateur ajoute l'annonce à sa liste des annonces auxquelles il a répondu.

Le diagramme ci-dessous [Fig.11] met en évidence la suppression d'une annonce. L'utilisateur demande à l'administrateur de supprimer une annonce. L'administrateur transmet la requête au système des annonces. Ce dernier quant à lui vérifie auprès du système central que c'est bien un administrateur qui lui transmet cette requête. La requête est transmise ensuite au système profil. Si tel est le cas, il recherche l'annonce dans sa liste des annonces, la supprime et notifie la suppression aux autres protagonistes dans l'ordre inverse de la demande de suppression si elle existe.

Si l'annonce n'existe pas, les sujets seront notifiés dans l'ordre suivant : système profil, système centrale, Administrateur et Profil.

Si l'utilisateur n'existe pas, les sujets seront notifiés dans l'ordre suivant : système profil, système centrale, Administrateur et le Profil.

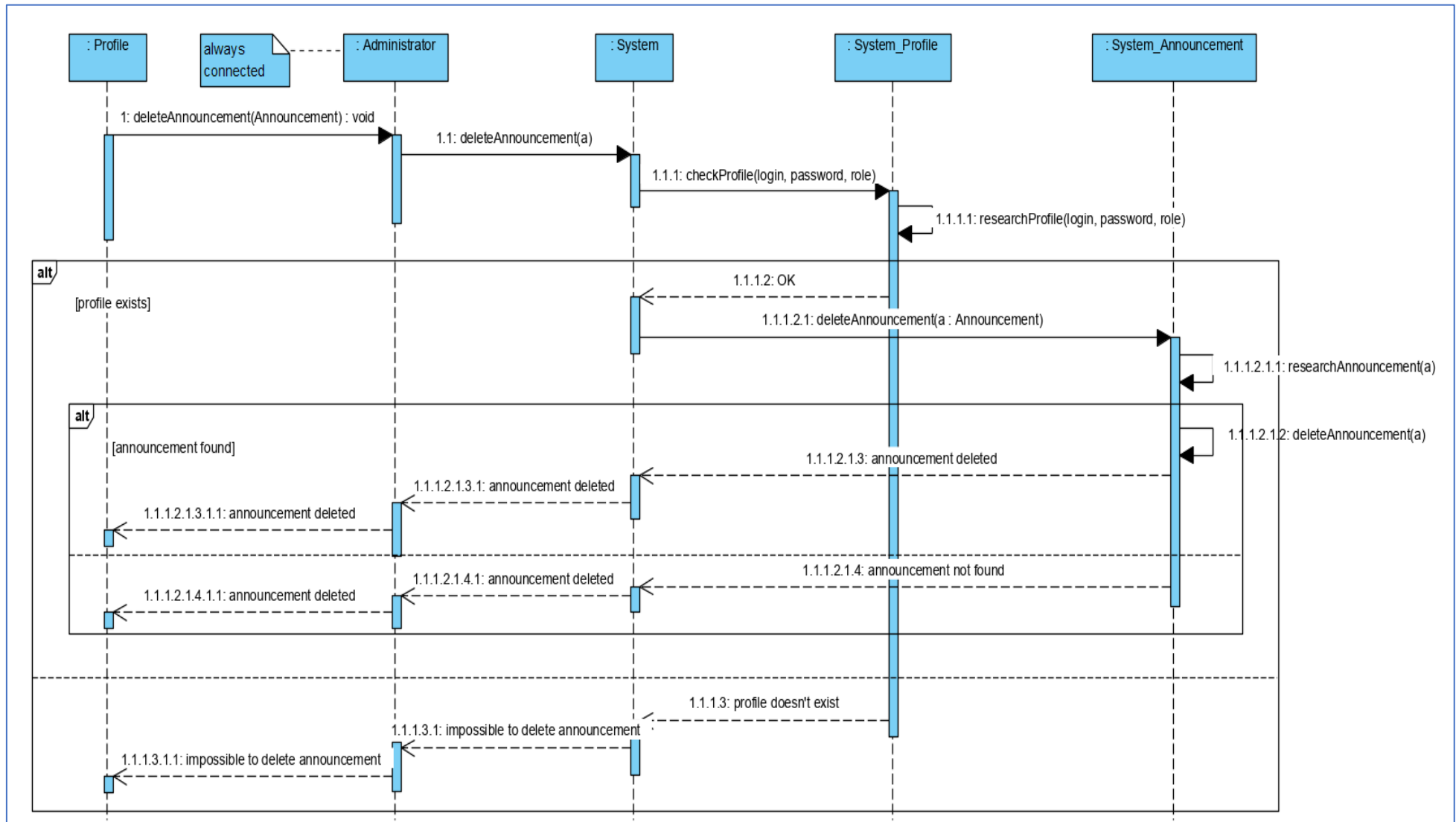


Fig.11 : Suppression d'une annonce.

4. Programme

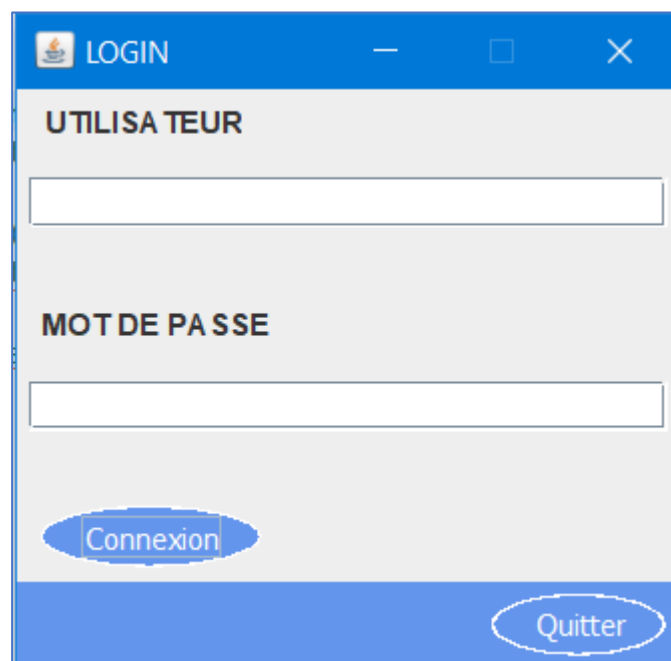


Fig.12. Fenêtre d'accueil.

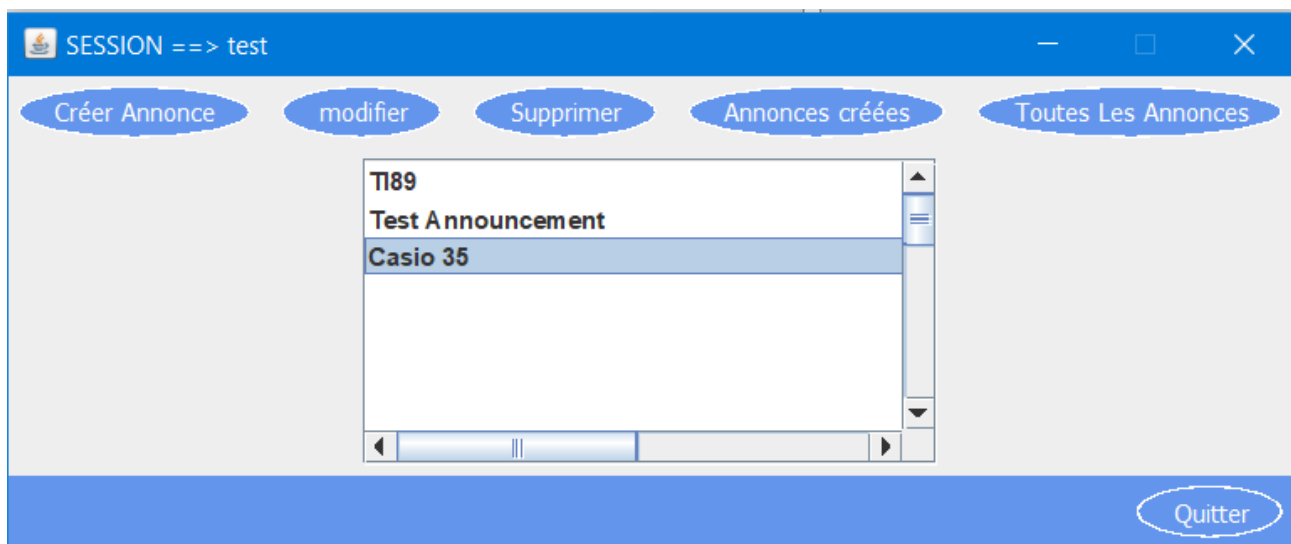


Fig.13. Vue Utilisateur.

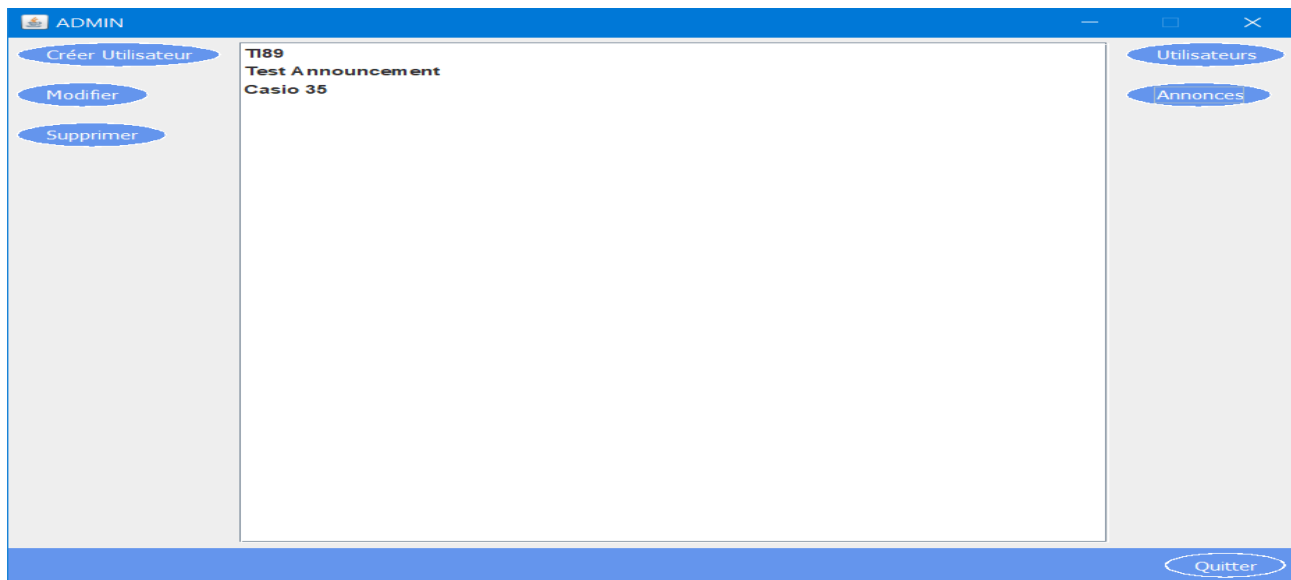


Fig.14. Vue Administrateur avec toutes les annonces créées.

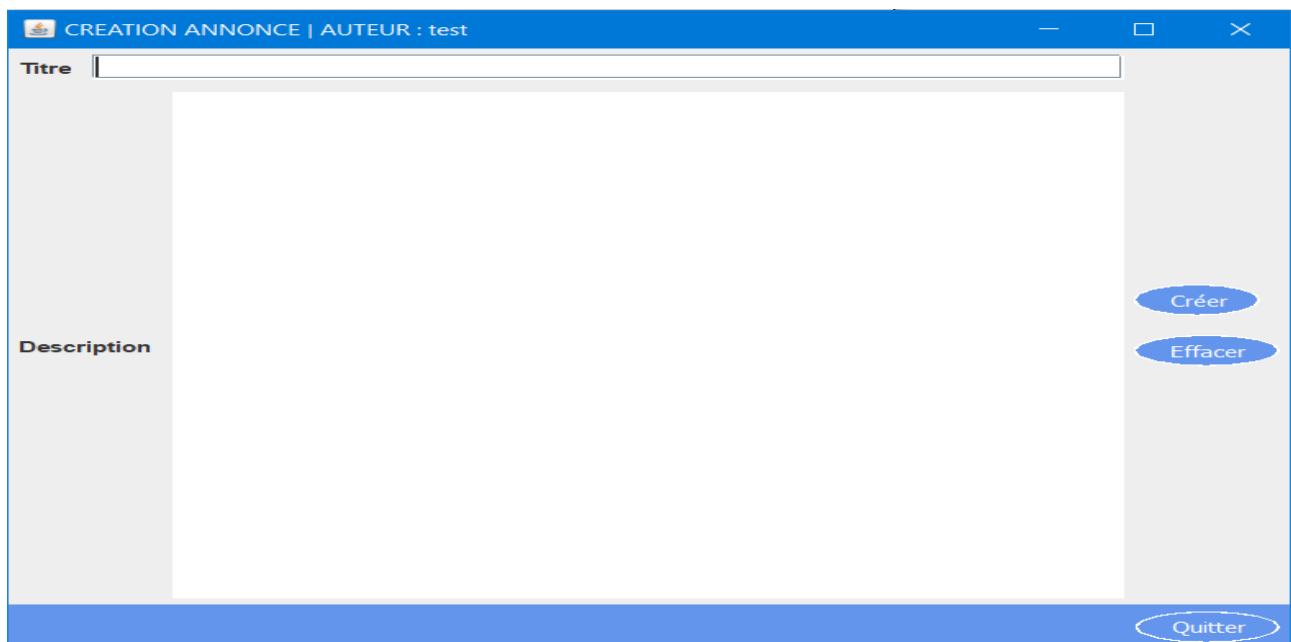


Fig.15. Fenêtre de création des annonces.

5. Organisation des fichiers

Les fichiers sources sont repartis comme suit :

- Package IHM : contient toutes les vues de l'application.
- Package DB : contient la classe « *MySystem* » permettant d'effectuer la connexion avec la base de données. Et la classe « *Requete* » qui est utilisée pour interagir avec la base de données.
- Package gestion : contient les classes « *Profil* », « *Utilisateur* », « *Administrateur* », « *Annonce* » ainsi que les interfaces « *UserMethods* » et « *AdminMethods* ».
- Package test : contient la classe « *AnnonceTest* » utilisée pour réaliser les tests unitaires sur la classe Annonce.
- Package main : contient la classe « *Main* » utilisée comme programme principal pour lancer l'application.

Conclusion

Ce projet nous a permis de mieux comprendre et appréhender les patrons de conception. Ce fut une expérience très enrichissante. Toutefois, nous n'avons pas pu implémenter toutes les idées que nous avions par manque de temps.