

# Rapport du projet IHM

Adrien Dos Santos

Kayyissa Haissous

Claire Gobert

Etudiants en BUT Informatique

Groupe 1

## Introduction :

Le but de ce projet était de réaliser une application de gestion de groupes à l'aide d'une API fournie.

Nous nous sommes grossièrement réparti le travail ainsi :

Lien avec la base de données pour le modèle persistant : Adrien

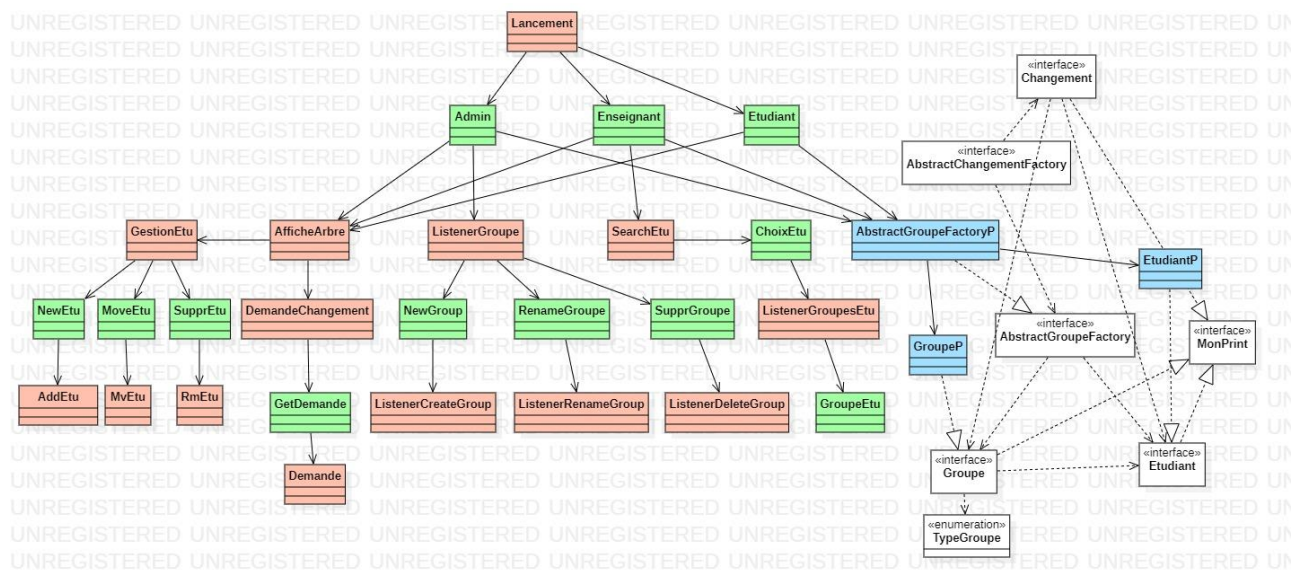
Vues graphiques : Claire

Lien entre les vues et la manipulation des données : Kayyissa

## L'API :

L'API fournie contient diverses classes que nous avons implémenté pour manipuler une base de données. Quelques modifications ont été apportées, notamment l'ajout d'une méthode `getGroupeByName` dans la classe `AbstractGroupeFactory`.

## Structure du code :



Légende :

Blanc -> API

Bleu -> Modèle persistant

Vert -> Vues

Rouge -> Contrôleurs

Le modèle persistant est actuellement un modèle non persistant car ce dernier possède bien plus de fonctionnalités. Un avantage à utiliser une telle API est qu'il suffit de changer l'appel à la classe `AbstractGroupeFactory` dans les fenêtres admin enseignant ou étudiant pour changer de modèle.

## Les fonctionnalités :

Plusieurs fonctionnalités étaient demandées suivant un ordre de priorité (MoSCoW). Voici celles que nous avons réalisées et testé :

L'administrateur doit pouvoir :

M créer, supprimer et renommer un groupe

M ajouter un individu dans un groupe

M déplacer un individu dans un autre groupe

S fabriquer automatiquement une partition des étudiants en plusieurs groupes

C Les groupes forment une arborescence. Tout en haut toute la promo, un groupe indestructible ensuite on peut soit ajouter des groupes qui ne forment pas une partition, soit un groupe qui forme une partition.

L'enseignant doit pouvoir :

M afficher la liste des groupes

M afficher la liste des étudiants d'un groupe donné

L'étudiant doit pouvoir :

M afficher la liste des groupes

M afficher la liste des étudiants d'un groupe donné

En plus de réaliser ces fonctionnalités nous nous sommes assurés du mieux possible de la pertinence de ces actions. En effet, nous n'avons pas permis d'ajouter un groupe à une partition déjà créée, ou encore de déplacer un étudiant dans un groupe où il se trouve déjà. La suppression d'un étudiant d'un groupe qui est fils d'une partition, supprimera aussi l'étudiant de la partition. A la création ou la suppression d'un groupe l'affichage est mis à jour au même titre qu'en cas de modification des groupes des étudiants.

Nous avons réalisé d'autres fonctionnalités mais de façon partielle :

L'enseignant doit pouvoir :

S chercher le groupe d'un étudiant à partir des premières lettres de son nom

L'étudiant doit pouvoir :

M demander à passer dans un groupe qui est moins plein que le sien en ajoutant une explication (demande de type 1 à faire valider).

En effet, actuellement la recherche des groupes auxquels appartient un étudiant se fait à partir du nom complet. Nous savons comment le faire mais nous nous interrogeons sur l'interface qui va avec. Est-ce à l'utilisateur ou à l'application d'estimer la précision de sa recherche ?

Pour ce qui est des demandes de changement de groupes, elles se font dans le cas où le groupe destination souhaité n'a pas atteint sa taille maximale et non s'il est moins plein que celui de l'étudiant. Nous savons également comment le faire mais notre application manque de gestion de demandes de changement de manière générale.

Les autres fonctionnalités n'ont pas été réalisées.

## Conclusions :

**Conclusion générale :** Ce projet nous a appris l'intérêt d'une API : avoir un squelette commun pour que chacun puisse fournir du code compatible avec celui des autres membres. Nous sommes un peu déçus de ne pas avoir eu le temps d'approfondir la gestion des demandes de changement de groupe, car nous avons beaucoup de travail à réaliser en ce moment, mais nous sommes plutôt satisfaits de notre application.

**Claire :** Ce projet était plus long que prévu, de plus nous avons commencé en retard, mais nous avons malgré tout réussi à faire à peu près toutes les fonctionnalités les plus importantes. Une meilleure organisation aurait sûrement permis de faire le projet dans sa totalité.

**Adrien :** Pour ma part ce projet fut une bonne démonstration de l'utilité d'une API (livrée en état de marche avant les vacances) qui simplifie l'utilisation du code pour les autres.

**Kayyissa :** Ayant réalisé ce projet dans un état de fatigue indécemment, je préfère ne pas me prononcer.