

RAPPORT SAÉ PROJET IHM

SOMMAIRE

01

Introduction

02

Aspects techniques

03

Cas de tests

04

Conclusion

Introduction

Bienvenu, dans ce rapport nous allons vous expliquer les différentes étapes de la réalisation de notre Projet IHM. Nous commencerons comme vous avez pu le voir dans le sommaire par vous parler de la partie technique, puis nous vous parlerons des cas de tests avant de conclure ce rapport.

Partie technique

Nous avons essayé de réaliser la totalité des fonctionnalités demandées dans le sujet peu importe leur niveau de notation MoSCoW. Malgré notre bonne volonté, certaines fonctionnalités ont dû être mises de côté comme la barre de recherche pour la vue Professeur. Tout le reste est normalement implémenté dans notre programme et respecte le modèle MVC au mieux.

Nous nous sommes servis avec succès du squelette de l'API imposé et avons réalisés nos différentes classes en fonction de cette API. Étant donné que l'utilisation d'une API n'est pas très intuitive, nous avons désigné Rayan comme garant de celle-ci. Cela signifie pour nous que nous avons pu parfois nous écarter de son utilisation pour faciliter le travail mais tout en le précisant à Rayan qui a pris soin de repasser derrière et de changer notamment les requêtes SQL par l'utilisation de l'API.

Pour le côté persistance des données, nous avons créé une base de données dès le début du projet avec 4 tables :

- Etudiant qui contient le nom, prénom, et qui incrémente un identifiant pour chaque étudiant créé ;
- Groupe qui contient l'identifiant du groupe, son nom, l'identifiant de son père, sa taille minimale et sa taille maximale ainsi que son type qui décrit quels sont les étudiants qui peuvent être ajoutés ;
- Contient qui permet de savoir à quels groupes appartiennent chaque étudiant ;
- Changement qui permet de savoir quelles sont les demandes de changement de groupe effectuées par des étudiants et qui sont en attentes de validation ou de refus. On y retrouve l'identifiant de l'étudiant ainsi que ceux des groupes de départ et d'arrivée et pour finir le type de demande (niveau 1 ou 2) ainsi que les explications. Un identifiant de demande est automatiquement incrémenté.

Nous avons aussi pris la liberté de créer un menu lors du lancement de l'Interface qui permet à l'utilisateur de choisir sur quelle vue il souhaite se connecter. Pour la vue étudiant, ce menu demande aussi le nom de l'étudiant qui souhaite se connecter car nous en avons ensuite besoin pour faire les demandes de changement de groupe. Pour l'administrateur, cela demande un mot de passe qui avec beaucoup de sécurité n'est pas crypté ni visuellement ni dans le code. Le mot de passe est « root ». Ainsi, si le mot de passe n'est pas changé pour root rien ne se passe et l'utilisateur ne peut pas se connecter à la vue Administrateur. Pour finir, la vue Professeur ne demande rien de particulier.

Pour les aspects techniques des fonctionnalités demandées, les trois vues permettent de voir l'arborescence des groupes et les étudiants d'un groupe donné. Pour une partie des fonctionnalités, un bouton est présent et permet d'ouvrir une nouvelle fenêtre pour permettre à l'utilisateur de remplir les arguments demandés.

L'administrateur lui peut pour gérer les groupes :

- Faire des partitions dans un groupe ;
- Créer des sous-groupes
- Ajouter des étudiants dans un groupe;
- Modifier le nom d'un groupe ;
- Supprimer un groupe.

Il peut aussi :

- gérer la création d'un étudiant. C'est une fonctionnalité que nous avons décidé d'ajouter et cela entraîne la création d'un étudiant qui est automatiquement ajouté au groupe Promotion.
- Voir les demandes de changement de groupe effectué par les étudiants. Il peut les accepter ou les refuser.

L'étudiant peut quant à lui faire une demande pour changer de groupe. Il peut aussi voir les demandes de changements de groupe. Ici, nous n'avons pas eu le temps de séparer les demandes de niveau 1 et celles de niveau 2. Cela implique que l'étudiant voit s'afficher l'ensembles des demandes.

Cas de test

Pour tester le bon fonctionnement de notre interface nous vous proposons de lancer la vue administrateur en entrant le mod de passe « root » afin d'utiliser toutes les fonctionnalités qui sont proposées comme :

- Créer un étudiant en entrant son nom et prénom ;
- Modifier un groupe en le renommant ;
- Supprimer un étudiant d'un groupe ;
- Supprimer un groupe ;
- Etc.

Il existe aussi un problème sur la création des groupes. Un groupe peut être créé avec une taille minimale mais aucun étudiant n'est directement ajouté ce qui engendre un groupe vide avec une taille minimale de 2 par exemple.

Vous pouvez aussi lancer la vue Étudiant et tester de demander à changer de groupe. Ainsi, vous pourrez voir qu'un étudiant ne peut pas changer de groupe si le groupe de départ et celui d'arrivée n'ont pas le même père ou encore s'il ne rentre pas d'explication. Vous pouvez aussi afficher les demandes en cours. Vous vous rendrez peut-être compte de deux problèmes ici. Le premier est qu'une demande qui vient d'être faites ne s'affiche pas sans relancer l'interface. Ce bug que nous avons découvert assez tard n'a pas pu être résolu. Le deuxième est que s'il existe plusieurs demandes, la page d'affichage superpose les demandes. Cela n'est pas le cas pour la page qui permet de voir les demandes du côté de la vue administrateur. Nous avons aussi découvert ce problème tardivement et n'avons pas pris le temps de le réparer.

Conclusion

En conclusion la réalisation de ce Projet IHM nous a permis de mettre en application toutes les connaissances ainsi que les différentes méthodes de travail vu en TP. Malgré des manquements quant à certaines fonctionnalités attendues, le plus important a été réalisé. Nous avons su parfaitement compléter l'API fournit et réaliser dans les délais une Interface Homme Machine en Java. Chaque membre du groupe en sort grandi et avec une meilleure compréhension de la réalisation d'une IHM.