IUT Sénart-Fontainebleau, UPEC

Luc Hernandez
Florent Madelaine

IUT Info 2A

2022-23

Projet IHM

Ce projet concerne le cours de DEV.3.1. et une partie importante de votre SAE du S3 (au moins 25 sur 120).

On y retrouve les ingrédients entrevus en TP : une interface homme-machine à coder en java, et une base de donnée à mettre en oeuvre pour assurer la persistance des données.

Il faudra que votre travail s'organise selon le modèle MVC. Par ailleurs, la communication avec le modèle doit passer par une API [1] qui est fournie. Une implantation non persistante du modèle est également fournie.

Le travail se fait à 2 ou 3. Les deux étudiants doivent être dans le même groupe de TP.

Date limite de rendu du rapport et de la réalisation. 21 novembre, minuit, heure française.

Contexte

On s'intéresse à un système de gestion de groupes. Pour simplifier il a une seule promotion et on suppose que les étudiants sont créés.

Il y a trois profils d'utilisateurs. Administrateur, Enseignant et Étudiant. On ne vous demande pas de faire la partie connexion. Il suffit en pratique de livrer trois applis indépendantes.

les fonctionnalités ci-dessous sont indiquées avec la notation MoSCoW.

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 déplacer un individu en validant sa demande (demande de type 1)

S refus d'une demande (de type 1)

C échanger des individus entre 2 groupes à leur demande respectives (demande de type 2)

C refus d'une demande (de type 2)

1. *Application Programing Interface* : ici une collection d'interfaces java et quelques objets concrets qui normalise la communication entre deux parties d'un logiciel.

W généralisation de l'idée ci-dessus mais avec des cycles de taille plus grande que 2.

S fabriquer automatiquement une partition des étudiants en plusieurs groupes

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

Par exemple : à une partition classique sous forme d'arbre Promo, groupes de TD, groupes de TP, groupes de projets, on pourrait ajouter en parallèle de groupes de TD une autre partition en boursier/non boursiers, et des groupes qui ne forment pas de partition par exemple en (aime la) pizza et (aime les) pâtes.

W adapter toutes les opérations à cette arborescence complexe.

NB. la suppression d'un groupe n'entraîne pas la destruction d'un étudiant y appartenant. Il est tout à fait possible d'appartenir à plusieurs groupes.

L'enseignant doit pouvoir

M afficher la liste des groupes

M afficher la liste des étudiant d'un groupe donné

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

L'étudiant doit pouvoir

M afficher la liste des groupes

M afficher la liste des étudiant d'un groupe donné

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).

S demander à passer dans un groupe qui est de même taille ou plus grand en ajoutant une explication pour l'administrateur (demande de type 2 à faire valider).

S voir les demandes de changement de groupe du type 2 (pour favoriser les échanges).

Vous devez programmer 3 prototypes d'IHM, un par utilisateur qu'on va appeler des vues dans la suite de ce sujet.

Vous devez utiliser dans un premier temps une API qui permettra de faire communiquer vos vues avec le modèle pour obtenir des données et envoyer des demandes de mise à jour du modèle.

Modèle persistant des données. Dans un second temps il faudra implémenter un modèle de données persistants. C'est-à-dire qu'il faudra trouver un mécanisme pour permettre de fermer puis de démarrer les vues sans perte de données.

Recommandations. Il faudra tâcher de suivre les principes d'ergonomie vues en IHM. Il faudra fournir un *code fonctionnel*, *agréable à lire*, *documenté* (javadoc) et qui répond à toutes les recommandations et bonnes pratiques vues avec Luc Hernandez, en particulier en ce qui concerne le déploiement avec make.

Finalement, et c'est un aspect essentiel de ce projet, votre code doit s'organiser selon le **modèle MVC**. En particulier, pour la communication entre M et le reste, vous **devez utiliser l'API fournie**.

À rendre

- Le code fonctionnel décrit aux sections précédentes **sur le git d'un étudiant du groupe** avec comme nom de projet **FIprojetIHM2022**.
- Un rapport technique de 3 pages environ explicitant la réalisation, en particulier l'API avec des diagrammes adaptés, détaillant ce qui a été fait et testé avec succès, ce qui a été fait mais n'a pas fonctionné.
- Il faut aussi décrire des cas de tests à réaliser de manière interactive sur votre interface (que ce soit des tests passés avec succès par votre réalisation ou des tests montrant un bug).

Indication sur la notation Une démo démontrant les fonctionnalités de vos applications, leur ergonomie et votre note de rapport fournira votre note de SAE de projet IHM.

Une note technique qui tiendra compte de la qualité du code et du suivi des consignes fournira votre note pour le cours DEV.3.1. avec votre note de préprojet IHM (coef 1 pour la maquette, 4 pour la note technique du projet IHM).

La note technique est fabriquée comme indiqué ci-dessous. La partie vue compte pour 14 points environ. La partie persistance des données compte pour 6 points environ.

Un malus très important sera appliqué si vous n'utilisez pas l'API (moins 10). Un malus important sera appliqué si vous ne documentez pas (moins 4). Un malus très important sera appliqué si il n'est pas possible de déployer votre application sur une machine de l'IUT à partir du git (moins 10 à moins 20). Un malus très important sera appliqué si vous ne donnez pas accès à votre git aux correcteurs Florent Madelaine et Luc Hernandez (moins 20).

En cas de soucis dans un groupe, il faut aviser très rapidement les enseignants.

Il est possible que les membres d'un groupe puisse obtenir des notes différentes en fonction de leur implication dans le projet et en cas de doute il sera possible que nous demandions des détails à individuellement à des membres du groupe pour mieux estimer leur participation au travail d'un groupe.

Globalement, la notation favorisera une réalisation moins complète mais de qualité sur une tentative de résolution complète mais avec des finitions médiocres.

Description de l'API

Au tableau.

L'API vous sera communiquée via git.

Si vous avez besoin de faire évoluer l'API pour répondre à des fonctionnalités avancées ou bien si vous avez des corrections, prière de contacter Florent Madelaine.