Hayar Malak De Wancker Maddy Margely Gwendal BUT Informatique S2
IUT A de Lille
Groupe B-G3

Rapport IHM SAE 2.01-02

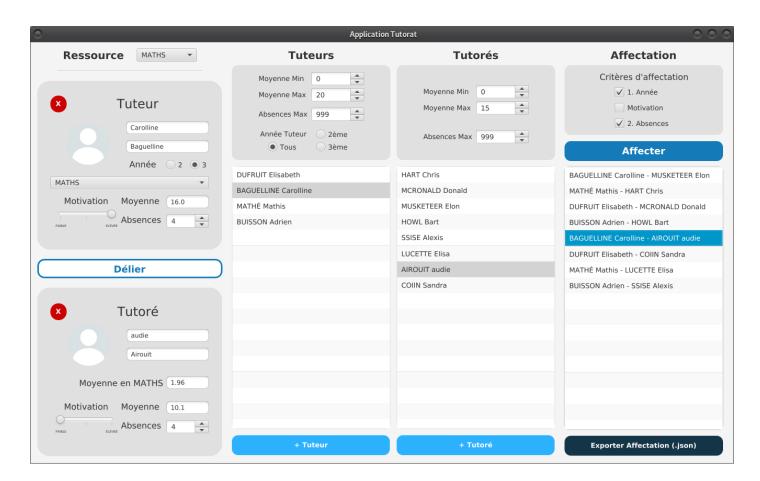
Notre projet a été de concevoir une application de gestion de tutorat. Les étudiants ciblés sont les *Tuteurs* et les *Tutorés*. On ne peut affecter un tuteur à un tutoré que si ce dernier propose un tutorat dans la ressource sélectionnée.

1. Lien vers le dépôt git

https://gitlab.univ-lille.fr/sae2.01-2.02/2022/B-G3

<u>Référence commit</u>: 0574d30b2d3f7ba931f73684186cf92d3b1dcf5e - "Final Commit for the User Interface"

2. Capture d'écran de l'application finale



3. Justification des choix de conception

- Pas de page de connexion, une seule page complète sans être visuellement complexe.
- Cohérence dans l'agencement, l'interface est divisé en trois partie
- Une partie à gauche contenant les information du tuteur/tutoré :

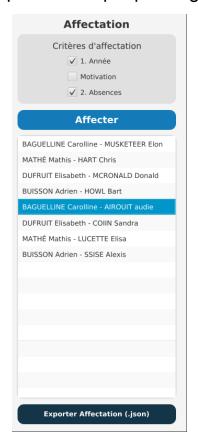




- La partie au milieu des listes des tutorés et des tuteurs de la ressource sélectionnée où il est possible de les filtrer selon plusieurs critères (si aucun critère choisi, l'affectation se base uniquement sur les moyennes des tuteurs et tutorés).



- La partie de droite concerne l'affectation, où on peut pondérer les critères d'affectation pris en compte par l'algorithme.



 Bon choix de widget : textField pour les notes, spinner pour le nombre d'absences, choicebox pour la sélection de la ressource, slider pour le niveau de motivation.

La pondération des affectations en fonction de l'ordre de sélection des critères (si on choisit le critère "motivation" avant "année", il aura un poids plus grand). Nous avons choisi pour cela des checkbox qui affichent l'ordre de choix des critères.



- Les actions de chaque bouton sont simples et reconnaissables : "lier", "+ tutoré", "filtrer"....
- Utilisation d'une charte graphique esthétique, et regroupement par couleur des boutons effectuant le même type d'action (Exemple : les boutons "+ tuteur" et "+ tutoré" sont du même ton de bleu contrairement aux autres).
- Widgets situés aux bons endroits et qui attirent l'attention : widgets de filtrage et d'ajout de tuteur/tutoré situés de manière visible respectivement au-dessus et en dessous de leurs listes.

4. Contribution des membres du groupe

- Malak

J'ai principalement travaillé sur la conception du programme, notamment dans l'implémentation de l'algorithme de calcul d'affectation qui a été un exercice assez difficile, mais aussi dans l'écriture des différentes classes de l'application. J'ai aussi participé à l'élaboration des différents mockups graphiques et sur la conception de l'interface sur SceneBuilder.

- Gwendal

Pour ma part, j'ai aidé à la conception dans l'écriture du code, mais je me suis surtout occupé de l'aspect utilisateur de l'application. J'ai réalisé l'interface, en très grande partie. Pour cela j'ai utilisé SceneBuilder, ainsi que Figma, un outil que je savais déjà utiliser.

- Maddy

Pour la partie dev, j'ai participé à l'élaboration de l'UML et à la création des classes de test. Pour la partie IHM je me suis occupé de rendre l'interface FXML (conçue majoritairement par Gwendal) fonctionnelle avec le code JavaFx. Pendant le processus de programmation de l'interface nous nous sommes rendus compte des failles de l'interface et avons pu l'ajuster sur SceneBuilder en conséquence.