# Rapport :

# Le développement d'une application de gestion des étudiants : Amélioration de l'expérience utilisateur et optimisation des fonctionnalités


Réalisé par :

- Salma EL HASSNAOUI
- Douae EL HILA


2022/2023

## 1-introduction :

**<Java FX>** est une plateforme de développement d'interfaces graphiques modernes et dynamiques pour les applications Java. Elle offre un ensemble riche de composants graphiques et d'outils de conception qui permettent de créer des interfaces utilisateur attrayantes et interactives. L'un des concepts clés utilisés dans le développement d'applications Java FX est le modèle MVC, un modèle de conception qui divise une application en trois composants distincts : Modèle-Vue-Contrôleur.
L'application de gestion des étudiants présentée dans ce rapport offre une solution pratique et conviviale pour **la gestion efficace des informations des étudiants de l'École Nationale des Sciences Appliquées**. Avec ses fonctionnalités d'ajout, de suppression et de mise à jour des données, ainsi que la possibilité d'afficher les profils détaillés des étudiants, cette application simplifie le processus de suivi et d'organisation des informations des étudiants. Elle offre une interface intuitive et structurée, permettant aux utilisateurs de naviguer facilement entre les différentes fonctionnalités.

Dans cette application, le modèle MVC est mis en œuvre de la manière suivante : le modèle représente les données des étudiants ainsi que la logique métier associée, telles que la connexion à la base de données et la manipulation des données des étudiants. La vue est responsable de l'affichage des informations des étudiants et de l'interaction avec l'utilisateur, comme l'affichage de la table des étudiants et des profils individuels. Enfin, le contrôleur gère les actions de l'utilisateur, en réagissant aux événements tels que les clics de boutons et en coordonnant les interactions entre la vue et le modèle.

## 2-description des scènes :

### ➔ La scène Table :

La scène "Table.fxml" représente une scène graphique qui affiche une table (TableView ) contenant des informations des étudiants .Elle offre des fonctionnalités d'insertion, de suppression et de mise à jour des données.

Le contrôleur <Controller.java > gère les actions des différents éléments de l'interface et les interactions avec la base de données.

Le bouton "*addstudentButton*" est utilisé pour naviguer vers la scène "AddScene.fxml " en appelant la méthode **'goToAddScene'** du contrôleur. Les boutons "deleteButton", "updateButton", et " viewButton" sont créés pour chaque étudiant affiché dans la table. Le bouton "deleteButton" est configuré pour supprimer l'étudiant de la base de données en appelant la méthode '**deleteStudentFromDatabase'** du contrôleur, avec l'affichage d'une fenêtre contextuelle (popup). Le bouton "updateButton" est configuré pour naviguer vers la scène "UpdateScene.fxml " en appelant la méthode '**goToUpdateScene'** du contrôleur. Quand au bouton "viewButton", il a le rôle de naviguer vers la scène "ViewScene.fxml " en appelant la méthode '**goToViewScene** ' du contrôleur.
Les trois méthodes **'goToAddScene'**, '**goToUpdateScene'** et '**goToViewScene'** chargent le fichier FXML, créent la scène correspondante et passent les valeurs de l'étudiant au contrôleur de la scène. La méthode `**loadStudentsFromDatabase**` est utilisée pour charger les données des étudiants à partir de la base de données et les afficher dans la table. Elle utilise la classe <Database.java> pour établir une connexion à la base de données.

### ➔ La scène AddScene :

La scène "AddScene.fxml " représente une interface utilisateur conviviale pour ajouter des étudiants à une base de données.

Elle comprend des champs de saisie pour les informations pertinentes de l'étudiant, une option de sélection d'image de l'étudiant, un `Text` pour afficher des messages d'erreur éventuels, tels que des avertissements concernant des champs vides ou des données incorrectes, et des spinners , initialisés par La méthode '**initialize**',utilisés pour sélectionner la date de naissance de l'étudiant .

Le contrôleur associé à cette scène, <AddController.java >, implémente les actions des différents éléments de l'interface.

Le bouton "*selectimgButton*" ouvre une boîte de dialogue permettant à l'utilisateur de sélectionner une image, tandis que le bouton "*addButton*" ajoute l'étudiant à la base de données en appelant la méthode `**addStudent**`. La méthode ' **goToTable'** permet de naviguer vers la scène "Table.fxml ". Elle est appelée une fois l'ajout est terminé avec succès, ou lorsque Le bouton "backbutton" avec l'icône de flèche vers l'arrière est pressé.

➔ La scène UpdateScene :

La scène "UpdateScene.fxml " représente une interface graphique pour l'édition des informations d'un étudiant. Elle comprend les mêmes champs que la scène "AddScene.fxml ". Le bouton "*editButton*" est utilisé pour mettre à jour les informations de l'étudiant dans la base de données en appelant la méthode '**updateStudent**' et naviguer vers la scène "Table.fxml ".

Le contrôleur associé à cette scène, **<**UpdateController.java **>,** implémente les actions des différents éléments de l'interface .La méthode '**setValues**' est utilisée pour initialiser les valeurs de chaque champs en fonction de l'étudiant sélectionné, La méthode '**getMonthNumber**' est utilisée pour convertir le nom du mois en son numéro correspondant. Les autres méthodes fonctionnent de la même manière que dans la scène "AddScene.fxml ".

➔ La scène ViewScene :

La scène "ViewScene.fxml " offre une interface utilisateur claire et organisée pour afficher les détails d'un étudiant sélectionné. Elle utilise une combinaison d'éléments visuels, d'étiquettes et de boîtes de dialogue pour fournir une expérience conviviale et informative.

Le contrôleur associé à cette scène, <ViewController.java>, est responsable de l'affichage des données de l'étudiant dans les éléments d'interface correspondants. Les valeurs des différentes propriétés de l'étudiant sont récupérées et sont utilisées pour mettre à jour les étiquettes et l'image de l'étudiant.

Le bouton "*printButton*" a le rôle de déclencher la méthode '**handlePrintButtonAction'** du contrôleur. Cette méthode récupère la fenêtre parent de l'événement, créé une instance de PrinterJob (classe de Java FX utilisée pour l'impression), affiche la boîte de dialogue d'impression et procède à l'impression de la fiche étudiante.

## 3-Les points forts du projet :

Ce projet présente plusieurs points forts qui le rendent convivial et efficace pour la gestion des étudiants. Tout d'abord, il offre la possibilité d'ajouter **une image pour chaque étudiant**, ce qui permet une meilleure visualisation et personnalisation des profils. Cela donne aux utilisateurs la possibilité de personnaliser davantage les informations des étudiants en ajoutant une image qui les représente.

En outre, **les boutons " deleteButton ", " updateButton " et " viewButton " sont judicieusement placés à côté de chaque étudiant dans la table**. Cette disposition facilite grandement leur utilisation, car il suffit de cliquer sur le bouton correspondant pour effectuer rapidement des actions telles que la suppression, la mise à jour des informations ou l'affichage du profil complet de l'étudiant correspondant. Cette proximité des boutons permet une interaction fluide et intuitive avec les données des étudiants.

Un autre aspect positif est **la présence de messages d'erreurs** qui s'affichent lors de la saisie de données. Ces messages guident l'utilisateur pour entrer des valeurs correctes et valides, tout en respectant un format spécifique. Cela permet d'éviter les erreurs de saisie et de garantir la cohérence des données enregistrées. Les messages d'erreur fournissent une rétroaction instantanée à l'utilisateur, l'aidant à corriger les erreurs rapidement et efficacement.

Par ailleurs, chaque fonctionnalité**, telle que l'ajout, la modification, l'affichage de la table et l'affichage du profil de l'étudiant, est placée dans une scène distincte**. Cette organisation claire et structurée facilite la navigation et la compréhension des différentes fonctionnalités du projet. Chaque scène se concentre sur une tâche spécifique, ce qui simplifie l'interaction avec l'application et facilite la maintenance. L'architecture de scènes distinctes permet aux utilisateurs de se concentrer sur une fonctionnalité spécifique à la fois, offrant une expérience utilisateur plus fluide et intuitive.

De plus, **les étudiants sont classés par ordre alphabétique dans la table**, ce qui facilite la recherche et la localisation des informations. Cette fonctionnalité permet aux utilisateurs de trouver rapidement un étudiant spécifique en parcourant simplement la liste alphabétique. L'organisation alphabétique facilite la recherche et la gestion des étudiants, améliorant ainsi l'efficacité de l'application.

Un autre point fort de ce projet est **que les scènes ne peuvent pas être redimensionnées**, ce qui contribue à maintenir un style cohérent et esthétique. En limitant la possibilité de redimensionner les scènes, l'interface graphique reste constante et les éléments de l'interface restent bien positionnés et alignés. Cela crée une expérience utilisateur agréable et professionnelle, en maintenant une apparence visuelle cohérente tout au long de l'application.

En outre, **la fenêtre contextuelle (popup)** qui servirait à confirmer l'action de suppression et à s'assurer que l'utilisateur ne le fait pas par accident. Cela éviterait les suppressions

involontaires et permettrait à l'utilisateur de confirmer son choix avant de supprimer définitivement un étudiant.

Un autre avantage de ce projet est **la possibilité pour l'utilisateur d'imprimer le profil d'un étudiant**. Cette fonctionnalité permet à l'utilisateur d'obtenir une version imprimée des informations de l'étudiant, ce qui peut être utile pour les besoins de documentation, les rapports ou simplement pour conserver une copie papier des données.

Enfin, le projet intègre également **un bouton de retour (backbutton)** qui permet de naviguer facilement d'une scène à une autre. Cette fonctionnalité rend l'expérience utilisateur plus fluide et pratique, en offrant la possibilité de revenir rapidement à une scène précédente sans perdre de données ou d'état. L'utilisateur peut ainsi naviguer entre les différentes fonctionnalités de manière intuitive et sans contraintes.

## 4- les améliorations possibles :

Cette application pourrait bénéficier de plusieurs améliorations pour offrir une expérience utilisateur encore plus complète. Une amélioration possible serait d'ajouter une scène de démarrage lors du lancement de l'application, où les utilisateurs auraient le choix entre deux boutons : un bouton "Espace étudiant" et un bouton "Espace professeur".

Si l'utilisateur clique sur le premier bouton "Espace étudiant", il serait redirigé vers la scène de connexion réservée aux étudiants. Une fois connecté, le profil de l'étudiant pourrait être affiché, fournissant ainsi un accès rapide aux informations personnelles de l'étudiant.

D'autre part, si l'utilisateur choisit le deuxième bouton "Espace professeur", il serait dirigé vers la scène de connexion dédiée aux professeurs. Après s'être connecté, il serait redirigé vers la table des étudiants, où il pourrait effectuer des actions spécifiques réservées aux professeurs, telles que l'ajout, la modification ou la suppression d'étudiants.

## 5- Conclusion :

En conclusion, **l'application de gestion des étudiants de l'École Nationale des Sciences Appliquées développée en utilisant Java FX** offre une solution robuste et conviviale pour la gestion efficace des informations des étudiants. Grâce à son interface graphique moderne et interactive, les utilisateurs peuvent facilement ajouter, supprimer et mettre à jour les données des étudiants, ainsi que visualiser les profils détaillés. Le modèle **MVC** a été judicieusement mis en œuvre, permettant une séparation claire des préoccupations et une meilleure visualisation du code.