**Rapport du projet - Gestion des inscriptions des étudiants à l'ENSAO**

**Réalisé Par les étudiants:**

**- Jidar Ahmed Amin**

**- Ankri Mohamed Khalil**

**- Khartouch Mouad**

* **Introduction:**

**Le présent rapport décrit le développement d'une application desktop en utilisant JavaFX pour la gestion des inscriptions des étudiants à l'ENSAO. L'objectif principal du projet est de permettre aux utilisateurs d'effectuer les opérations CRUD (Create, Read, Update, Delete) pour gérer les étudiants. L'application offre également la possibilité de visualiser le profil d'un étudiant dans une nouvelle scène et d'imprimer une fiche récapitulative en format PDF.**

* **Technologies utilisées:**

**JavaFX: Framework pour le développement d'interfaces utilisateur.**

**Java: Langage de programmation utilisé pour le développement de l'application.**

**JDBC (Java Database Connectivity): API pour la connexion et la manipulation de la base de données.**

**MySQL: Système de gestion de base de données utilisé pour stocker les informations des étudiants.**

* **Vue globale:**

**- Le code commence par déclarer le package nécessaire pour les différentes classes utilisées dans l'application, telles que " LoginScreen  "UpdateWindow", "DataUtils", "PrintWindow" et "Person" (classe descriptive des infos d'un étudiant).**

**Ensuite, la classe principale User étend la classe Application, ce qui indique que c'est une application JavaFX.**

**- L'application offre les fonctionnalités suivantes:**

**\*Authentification Login avec un nom d’utilisateur et mot de passe prédéfinis pour les administrateurs .**

**\* Ajout d'un étudiant: L'utilisateur peut saisir les informations relatives à un nouvel étudiant et les enregistrer dans la base de données.**

**\* Liste des étudiants: L'utilisateur peut afficher une liste de tous les étudiants inscrits à l'ENSAO.**

**- Les informations des étudiants sont affichées dans une table.**

**\* Mise à jour du profil étudiant: L'utilisateur peut sélectionner un étudiant existant et mettre à jour ses informations, par exemple, en modifiant son adresse ou son numéro de téléphone.**

**\* Suppression d'un étudiant: L'utilisateur peut supprimer un étudiant de la base de données en le sélectionnant dans la liste des étudiants et en confirmant la suppression.**

**\* Visualisation du profil étudiant: L'utilisateur peut sélectionner un étudiant dans la liste et afficher son profil complet dans une nouvelle scène.**

**\* Impression de la fiche récapitulative: L'utilisateur peut générer une fiche récapitulative du profil étudiant au format PDF.**

* **Fonctionnalités de l'application (Détaillées):**

1. **L’authentification :**

**Pour garantir l'accès sécurisé et restreint aux fonctionnalités de gestion des inscriptions.**

**Lorsque l'application est lancée, un écran de login apparaît, demandant à l'utilisateur de saisir un nom d'utilisateur et un mot de passe. Les informations de connexion prédéfinies sont utilisées pour valider l'identité de l'administrateur. Si les informations fournies correspondent aux données prédéfinies, l'accès est accordé et l'utilisateur est redirigé vers l'interface principale de gestion des inscriptions.**

**En cas d'erreur d'authentification, un message d'erreur approprié est affiché, informant l'utilisateur que les informations de connexion sont incorrectes. Cela garantit la sécurité et l'accès restreint aux fonctionnalités de l'application.**

**2) La création des colonnes de la TableView:**

**Les colonnes sont définies pour les champs ID, Prénom, Nom, E-mail, Age et Apogée.**

**Chaque colonne est configurée pour extraire la valeur correspondante du modèle de données Person en utilisant les propriétés de type SimpleIntegerProperty ou SimpleStringProperty.**

**Les colonnes sont ajoutées à la TableView.**

**3) La classe interne "Person":**

**Cette classe est utilisée comme modèle de données pour représenter une personne avec des propriétés telles que ID, Prénom, Nom, E-mail, Age et Apogée.**

**Les propriétés sont encapsulées à l'aide des classes SimpleIntegerProperty et SimpleStringProperty pour permettre le suivi des modifications.**

**Des méthodes getter sont définies pour chaque propriété afin de pouvoir les lier aux cellules de la TableView.**

**4) La création des composants de l'interface utilisateur:**

**- Un formulaire avec des champs pour saisir les informations d'une personne (Prénom, Nom, E-mail, Age, Apogée) est créé à l'aide d'une disposition en grille (GridPane).**

**- Des boutons tels que "Register", "Display All", "Delete" et "Modify Coordinates" sont ajoutés à la disposition en grille.**

**\*\* Les composants sont ajoutés à un conteneur vertical (VBox) qui contient également la TableView.**

**5) Les actions des boutons:**

**- Lorsque le bouton "Register" est cliqué, la méthode registerUser est appelée pour enregistrer une nouvelle personne en utilisant les valeurs saisies dans les champs du formulaire.**

**- Lorsque le bouton "Display All" est cliqué, la méthode displayContent de la classe DataUtils est appelée pour afficher toutes les données de la base de données dans la TableView.**

**- Lorsque le bouton "Delete" est cliqué, la méthode deleteObject est appelée pour supprimer la personne sélectionnée dans la TableView de la base de données.**

**- Lorsque le bouton "Modify Coordinates" est cliqué, la méthode navigateUpdate est appelée pour ouvrir une nouvelle fenêtre de mise à jour des coordonnées de la personne sélectionnée.**

**- Lorsque le bouton "View Details" est cliqué, la méthode navigatePrint est appelée et qui permet de naviguer vers une nouvelle fenêtre d'impression pour afficher les détails de l'élément sélectionné dans la table. Cela permet à l'utilisateur de visualiser ou d'imprimer les informations de l'élément sélectionné.**

**6) La connexion à la base de données:**

**Une connexion à la base de données MySQL est établie en utilisant le pilote JDBC.**

**Les informations de connexion, telles que l'URL de la base de données, le nom d'utilisateur et le mot de passe, sont fournies.**

**\*\* Si la connexion échoue, une exception SQLException est gérée et affichée dans la console.**

* **Styliser l'application:**

**via: scene.getStylesheets().add(getClass().getResource("styles.css").toExternalForm());**

**- Style global de l'application (background-color).**

**- Définir la couleur de fond et la couleur de la bordure de la TableView ainsi que la marge et le padding.**

**- Centre le contenu des colonnes et définir la taille de la police et la police à utiliser pour les colonnes.**

**- Définir la police à utiliser pour les cellules.**

**- Personnaliser la couleur de sélection d'un élément de la table.**

**- Personnaliser l'effet de survol des boutons.**

**- Personnalisation de la fenêtre d'impression de la fiche récapitulative en ajoutant le logo de l'université ainsi que celui de notre école.**

* **Conclusion:**

**L'application de gestion des inscriptions des étudiants à l'ENSAO offre une interface conviviale pour effectuer les opérations CRUD de manière efficace. Les fonctionnalités d'ajout, de mise à jour et de suppression des étudiants permettent à l'utilisateur de gérer facilement les inscriptions. La possibilité de visualiser le profil complet d'un étudiant dans une nouvelle scène et d'imprimer une fiche récapitulative au format PDF ajoute de la valeur à l'application.**