** UIDT/MSDA**

**2023/2024**

**Exposants :**

Aliou DIALLO

Abdoul Wahab LY

Sofiatou LO

Coumba CISSE

**Rapport sur le Projet : Gestion des Notes des Étudiants**

**1. Introduction**

**1.1 Contexte**

Ce projet a été développé pour faciliter la gestion des notes des étudiants au sein d'une institution académique. Dans de nombreux établissements, les enseignants ont besoin d'une solution efficace pour suivre et gérer les notes des étudiants, en assurant l'intégrité et l'exactitude des informations. Ce projet offre une interface graphique qui permet l’ajout, la modification, la suppression, ainsi que la consultation des notes des étudiants.

**1.2 Objectifs**

Les principaux objectifs du projet sont :

* Offrir une interface intuitive pour gérer les notes des étudiants.
* Permettre aux enseignants ou gestionnaires de saisir les informations relatives aux étudiants et à leurs notes.
* Garantir la validation des données avant leur insertion dans la base de données.
* Assurer la possibilité de modifier ou de supprimer les informations de manière sécurisée.

**2. Spécifications Techniques**

**2.1 Technologies Utilisées**

Le projet repose sur plusieurs technologies pour son développement :

* **Python (Tkinter)** : utilisé pour créer l'interface graphique (GUI).
* **MySQL** : utilisé pour gérer la base de données des étudiants et de leurs notes.
* **MySQL Connector** : bibliothèque Python utilisée pour interagir avec la base de données MySQL.
* **Tkinter Treeview** : utilisé pour afficher les données des étudiants sous forme de tableau dans l'interface.

**2.2 Structure de la Base de Données**

Le projet repose sur une base de données MySQL appelée **gestion\_notes**. La table principale de cette base de données est liste, qui stocke les informations des étudiants et leurs notes. Voici la structure de la table :

* identifiant\_etudiant (INT, clé primaire) : Identifiant unique de l'étudiant.
* prenom\_etudiant (VARCHAR) : Prénom de l'étudiant.
* nom\_etudiant (VARCHAR) : Nom de l'étudiant.
* matiere (VARCHAR) : Matière dans laquelle la note est attribuée.
* note (FLOAT) : Note obtenue par l'étudiant pour la matière donnée.

**3. Fonctionnalités du Système**

**3.1 Ajout de Notes**

L'utilisateur peut ajouter des informations sur un étudiant, y compris l'identifiant, le prénom, le nom, la matière, et la note. Avant l'ajout, plusieurs validations sont effectuées :

* Vérification que tous les champs sont remplis.
* Validation de l'identifiant (doit être un nombre entier).
* Vérification que l'identifiant et la matière ne sont pas déjà enregistrés dans la base de données pour éviter les doublons.
* Validation que la note est un nombre compris entre 0 et 20.

### ****3.2 Modification des Notes****

Cette fonctionnalité permet de modifier les informations relatives à un étudiant déjà enregistré dans la base de données. L'utilisateur doit sélectionner une note à partir de la table pour activer la modification. Des vérifications similaires à celles de l'ajout de notes sont effectuées.

### ****3.3 Suppression des Notes****

La fonctionnalité de suppression permet de retirer les informations relatives à un étudiant de la base de données. L'utilisateur doit sélectionner une entrée dans la table pour la supprimer. Un message de confirmation s’affiche pour éviter les suppressions accidentelles.

### ****3.4 Affichage des Notes****

Les notes des étudiants sont affichées sous forme de tableau (Treeview) dans l'interface utilisateur. La table est mise à jour dynamiquement après chaque ajout, modification ou suppression de données.

**4. Interface Utilisateur**

**4.1 Conception de l'Interface**

L'interface utilisateur a été construite avec Tkinter, une bibliothèque Python pour créer des interfaces graphiques. Elle est simple et intuitive, avec :

* Des champs de saisie pour entrer l'identifiant, le prénom, le nom, la matière et la note.
* Des boutons pour ajouter, modifier ou supprimer une note.
* Une table Treeview qui affiche toutes les notes enregistrées dans la base de données.

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

### ****4.2 Expérience Utilisateur****

L'expérience utilisateur a été optimisée avec des boîtes de dialogue (messagebox) qui informent l'utilisateur du succès ou de l'échec des actions. Les messages d'erreur s'affichent lorsqu'une saisie incorrecte est faite ou lorsqu'une action non autorisée est tentée (par exemple, la tentative de modification sans sélection préalable d'une note).

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

**5. Sécurité et Validations**

**5.1 Validation des Entrées**

Un ensemble de validations a été mis en place pour s'assurer que les données saisies sont correctes avant d'être insérées dans la base de données. Ces validations incluent :

* La vérification que les champs obligatoires ne sont pas vides.
* La validation du format des données (ex. l'identifiant doit être un entier, la note doit être un nombre compris entre 0 et 20).

### ****5.2 Gestion des Erreurs****

Les erreurs sont gérées de manière à éviter les plantages de l'application. Les blocages éventuels sont gérés avec des structures try-except, et des messages d’erreur sont affichés pour informer l’utilisateur en cas de problème (ex. une erreur de connexion à la base de données).

**6. Difficultés Rencontrées et Solutions**

**6.1 Connexion à la Base de Données**

Une des premières difficultés rencontrées était la gestion de la connexion à la base de données. Une fonction dédiée, connect\_db(), a été créée pour faciliter cette connexion et gérer les erreurs de connexion de manière appropriée.

**6.2 Mise à Jour des Notes en Temps Réel**

Mettre à jour la table affichée après chaque modification ou suppression nécessitait de réinitialiser et de recharger la table après chaque opération sur la base de données, ce qui a été résolu grâce à la fonction charger\_notes().

**7. Améliorations Futures**

Certaines améliorations pourraient être envisagées pour ce projet :

* **Recherche des étudiants** : Ajouter une fonctionnalité de recherche pour retrouver rapidement les notes d'un étudiant en fonction de son nom ou de la matière.
* **Authentification** : Mettre en place un système de connexion pour limiter l'accès aux fonctionnalités de modification ou de suppression des notes.
* **Export des données** : Ajouter la possibilité d'exporter les notes des étudiants dans des formats courants comme PDF ou Excel.

**8. Conclusion**

Ce projet de gestion des notes des étudiants a permis de créer une solution simple mais efficace pour enregistrer, modifier, supprimer et consulter les notes des étudiants via une interface graphique conviviale. Il a également permis d’acquérir des compétences en gestion de bases de données, en développement d'interfaces graphiques et en validation de données.