Rapport de C#

Fait par Reda Tahiri

I. Class Students:

Propriétés Principales :

StudentID: l’identifiant est attribué à chaque étudiant lors de leur ajout. Il est automatiquement incrémenté.

Password: Mot de passe associé à l'étudiant pour l'authentification.

Name et Prenom: Les noms et prénoms de l'étudiant.

CurrentClass: La classe actuelle de l'étudiant.

Niveau: Le niveau d'études de l'étudiant.

Special: La spécialité, le cas échéant, de l'étudiant.

Méthodes Principales :

Constructeur (): Initialise un nouvel objet Student en attribuant un identifiant unique automatiquement incrémenté à l'étudiant.

StudentsList: Une liste statique partagée qui stocke tous les objets Student créés.

lastAssignedId: Un champ statique privé qui garde la trace du dernier identifiant attribué

Add(): va permettre à l'utilisateur d'ajouter un nouvel étudiant en saisissant les détails tels que le nom, le prénom, la classe, le niveau, la spécialité et le mot de passe. L'étudiant est ensuite ajouté à la liste des étudiants.

Edit(): Permet à l'utilisateur de modifier les détails de l’étudiant existant en saisissant l'identifiant et le mot de passe de l'étudiant .

Delete(): Permet à l'utilisateur de supprimer un étudiant existant en saisissant l'identifiant et le mot de passe de l'étudiant à supprimer.

ViewDetails(): Permet à l'utilisateur de voir les détails d'un étudiant en saisissant l'identifiant et le mot de passe de l'étudiant. Les détails incluent l'identifiant, le nom, le prénom, la classe, le niveau et la spécialité de l'étudiant. De plus, le calendrier associé à l'étudiant est affiché s'il existe.

II. Class Teachers:

TeacherID: l’identifiant est attribué à chaque enseignant lors de leur ajout. Il est automatiquement incrémenté.

Password: Mot de passe associé à l'enseignant pour l'authentification.

Name et Prenom: Les noms et prénoms de l'enseignant.

Prof: La matière que l'enseignant enseigne.

Méthodes :

Constructeur (): Initialise un nouvel objet Teacher en attribuant un identifiant unique automatiquement incrémenté à l'enseignant.

Add(): Permet à l'utilisateur d'ajouter un nouvel enseignant en saisissant les détails tels que le nom, le prénom, la matière enseignée et le mot de passe. L'enseignant est ensuite ajouté à la liste des enseignants.

Edit(): Permet à l'utilisateur de modifier les détails d'un enseignant existant en saisissant l'identifiant et le mot de passe de l'enseignant à éditer. Les nouvelles informations sont ensuite mises à jour.

Delete(): Permet à l'utilisateur de supprimer un enseignant existant en saisissant l'identifiant et le mot de passe de l'enseignant à supprimer.

ViewDetails(): Permet à l'utilisateur de voir les détails d'un enseignant en saisissant l'identifiant et le mot de passe de l'enseignant. Les détails incluent l'identifiant, le nom, le prénom et la matière enseignée par l'enseignant.

lastAssignedId: Un champ statique privé qui garde la trace du dernier identifiant attribué.

TeachersList: Une liste statique partagée qui stocke tous les objets Teacher créés.

III. Class CourseManager:

AddCourse(string courseName, string courseDetails): Cette méthode permet d'ajouter un nouveau cours. Elle prend en paramètre le nom du cours et les détails du cours. Le cours est créé sous forme d'un fichier texte contenant les détails.

UpdateCourse(string courseName, string newCourseDetails): Permet de mettre à jour les détails d'un cours existant. Elle prend le nom du cours à mettre à jour et les nouveaux détails. Le contenu du fichier texte du cours est remplacé par les nouveaux détails.

DeleteCourse(string courseName): prend le nom du cours à supprimer et supprime le fichier texte associé.

ViewCourseContent(string courseName): Affiche le contenu d'un cours spécifique. Elle prend le nom du cours et affiche les détails stockés dans le fichier texte associé.

IV. Class Module:

Add(): Cette méthode permet à l'admin d'ajouter un nouveau module. Elle demande à l'admin de saisir le nom du module et l'ajoute à la liste des modules. Elle crée également un dossier associé au module.

Edit(): Permet de modifier les détails d'un module existant. L'utilisateur doit fournir le nom du module à modifier, puis saisir les nouvelles informations. La méthode met à jour le nom du module et renomme le dossier associé.

Delete(): Supprime un module existant. L'utilisateur doit entrer le nom du module à supprimer. La méthode supprime le module de la liste et supprime le dossier associé.

DisplayDetails(): Affiche les détails de tous les modules existants.

CreateModuleFolder(string ModuleName): Crée un dossier associé au module. Le nom du dossier est basé sur le nom du module.

UpdateModuleFolderName(string newModuleName): Met à jour le nom du dossier associé au module lorsque le nom du module est modifié.

DeleteModuleFolder(string ModuleName): Supprime le dossier associé au module.

AddCoursesToModule(): Permet d'ajouter des cours à un module spécifique. Affiche la liste des cours existants et demande à l'utilisateur de sélectionner un cours à ajouter au module.

V. Class Calendar:

DisplayExamDates(): Affiche les dates des examens stockées dans le dictionnaire examDates.

DisplayCourseDeadlines(): Affiche les dates limites des cours stockées dans le dictionnaire courseDeadlines.

ChangeExamDate(string examName, DateTime newDate, Administration admin): Permet de modifier la date d'un examen. Seul l'administrateur (Administration) a le droit d'apporter des modifications. La nouvelle date est enregistrée et les modifications sont sauvegardées dans le fichier.

ChangeCourseDeadline(string courseName, DateTime newDeadline, Administration admin): Permet de modifier la date limite d'un cours. Tout comme pour les examens, seule l'administration peut effectuer ces modifications. Les nouvelles dates sont enregistrées et les changements sont sauvegardés dans le fichier.

CreateExamDate(Administration admin): Permet à l'administration de créer une nouvelle date d'examen en saisissant le nom de l'examen et la date associée. Les informations sont sauvegardées dans le fichier après la création.

CreateCourseDeadline(Administration admin): De même, l'administration peut créer une nouvelle date limite de cours en fournissant le nom du cours et la date limite. Les détails sont sauvegardés dans le fichier.

SaveCalendarData(): Enregistre les données actuelles (examens et dates limites des cours) dans le fichier calendar.txt.

LoadCalendarData(): Charge les données du fichier calendar.txt lors de la création d'une instance de la classe.

VI. Class Administration:

ManageStudents(): Permet de gérer les opérations liées aux étudiants, telles que l'ajout, la modification, l'affichage des détails et la suppression d'étudiants.

ManageTeachers(): Facilite la gestion des enseignants en offrant des fonctionnalités similaires à celles des étudiants, telles que l'ajout, la modification, l'affichage des détails et la suppression d'enseignants.

CalendarMenu(Calendar calendar, Administration admin): Gère les opérations liées au calendrier, comme la création et la modification de dates d'examens et de dates limites de cours.

CoursesMenu(CourseManager courseManager): Permet d'effectuer des opérations sur les cours, telles que l'ajout, la mise à jour, la suppression et l'affichage des détails des cours.

VII. Class Program:

StudentMenu(Student student): Gère le menu dédié à la gestion des étudiants, offrant des options telles que l'ajout, la modification, la visualisation des détails et la suppression d'étudiants.

TeacherMenu(Teacher teacher): Gère le menu dédié à la gestion des enseignants, proposant des options similaires à celles des étudiants, telles que l'ajout, la modification, la visualisation des détails et la suppression d'enseignants.

AdminMenu(Administration admin, Calendar calendar): Coordination du menu d'administration, permettant à l'administrateur d'effectuer des tâches telles que la gestion des étudiants, la gestion des enseignants, la gestion du calendrier et la gestion des cours.

Main(): Fonction principale où le programme démarre. Elle crée les instances nécessaires, coordonne les différents menus et gère l'ensemble du flux du programme.

VIII. Conclusion

Le système de gestion scolaire que j’ai mis en œuvre dans ce projet est une solution complète qui prend en charge la gestion des étudiants, des enseignants, des cours, des modules et du calendrier.