



Rapport Général

Groupe EF-3

Sommaire

1	Introduction	2
2	Fonctionnalités principales	2
3	Architecture	3
3.1	Structure des Dossiers	3
3.2	Diagramme UML	4
4	Points forts de la plateforme	6
5	Organisation de l'Équipe et Méthodes Agiles	7
5.1	Réunions Hebdomadaires	7
5.2	Sprints	7
5.3	Utilisation d'Outils de Gestion de Projet	7
5.4	Code Reviews	7
5.5	Communication	8
5.6	Documentation	8
5.7	Retrospectives	8
6	Difficultés rencontrées	8
7	Conclusion	8

1 Introduction

La plateforme d'apprentissage interactif, développée dans le cadre du projet long, a pour objectif de fournir une expérience éducative enrichissante. Elle permet aux utilisateurs d'accéder à divers cours en ligne couvrant des domaines tels que les sciences numériques, la littérature et les mathématiques. En utilisant Java et JavaSwing, cette application propose une interface utilisateur conviviale et dynamique

2 Fonctionnalités principales

1. Interface Utilisateur :

- Développement d'une interface utilisateur intuitive et ergonomique avec Java Swing et AWT.
- Utilisation d'icônes et d'images pour une expérience visuelle enrichie.

2. Gestion des Données :

- Utilisation d'un fichier de propriétés pour le stockage local des données des utilisateurs et des cours.
- Classe `GestionneurData` pour la gestion des opérations de lecture et d'écriture des données.

3. Authentification et Inscription :

- Classes `Login` et `Inscription` pour gérer les processus de connexion et d'inscription.
- Séparation des processus pour les étudiants et les enseignants.

4. Chargement et Initialisation :

- Classe `LoadingScreen` pour afficher un écran de chargement interactif avec des messages de progression.
- Initialisation de l'application avec vérification des modules et des bases de données.

5. Gestion des Cours :

- Ajout, modification et suppression de cours, avec catégorisation par thème et niveau de difficulté.
- Suivi de la progression des étudiants dans les cours.

6. Interactions Utilisateur :

- Utilisation de boutons stylisés pour les actions principales comme la connexion et l'inscription.
- Interface principale avec options pour se connecter ou s'inscrire, dirigée par des boutons interactifs.

7. Classements et Récompenses :

- Affichage de classements quotidiens et hebdomadaires, avec des récompenses pour les meilleurs élèves

3 Architecture

3.1 Structure des Dossiers

L'architecture du projet est organisée en plusieurs dossiers principaux, chacun ayant un rôle spécifique dans le développement de l'application.

Voici la structure des dossiers et une brève description de leur contenu :

- **src/**
 - **projetlong/**
 - **Main.java** : Initialise et lance l'application avec une interface utilisateur principale.
 - **GestionneurData.java** : Gère le stockage et la récupération des données des utilisateurs et des cours à partir de fichiers de propriétés.
 - **Inscription.java** : Fournit une interface utilisateur pour l'inscription des étudiants et des enseignants.
 - **Login.java** : Fournit une interface utilisateur pour la connexion des étudiants, des enseignants et des administrateurs.
 - **Student/**
 - **ConnexionEtudiant.java** : Gère la connexion des étudiants à l'application.
 - **ConsulterProfil.java** : Permet aux étudiants de consulter leur profil utilisateur.
 - **EnrollCourse.java** : Permet aux étudiants de s'inscrire à des cours.
 - **GererProfil.java** : Permet aux étudiants de gérer et de modifier leur profil utilisateur.

- **Inbox.java** : Gère la messagerie des étudiants.
- **InscriptionEtudiant.java** : Gère l'inscription des étudiants à l'application.
- **Message.java** : Représente les messages échangés entre les utilisateurs.
- **Student.java** : Représente les données d'un étudiant.
- **StudentAccountDetails.java** : Gère les détails du compte des étudiants.
- **StudyCours.java** : Permet aux étudiants de suivre et d'étudier les cours inscrits.
- **SupprimerCompte.java** : Permet aux étudiants de supprimer leur compte utilisateur.
- **ViewParticipants.java** : Permet de voir les participants inscrits à un cours.
- **Exam/**
 - **AffichageExamenQCM.java** : Gère l'affichage des examens sous forme de QCM.
 - **App.java** : Classe principale de l'application de gestion des examens.
 - **Examen.java** : Représente un examen général.
 - **ExamenQCM.java** : Représente un examen sous forme de QCM.
 - **Exercice.java** : Représente un exercice général.
 - **ExerciceQCM.java** : Représente un exercice sous forme de QCM.
 - **ResultatExamen.java** : Gère l'affichage des résultats des examens.

3.2 Diagramme UML

Le diagramme UML suivant vous aidera à bien visualiser la structure des dossiers.

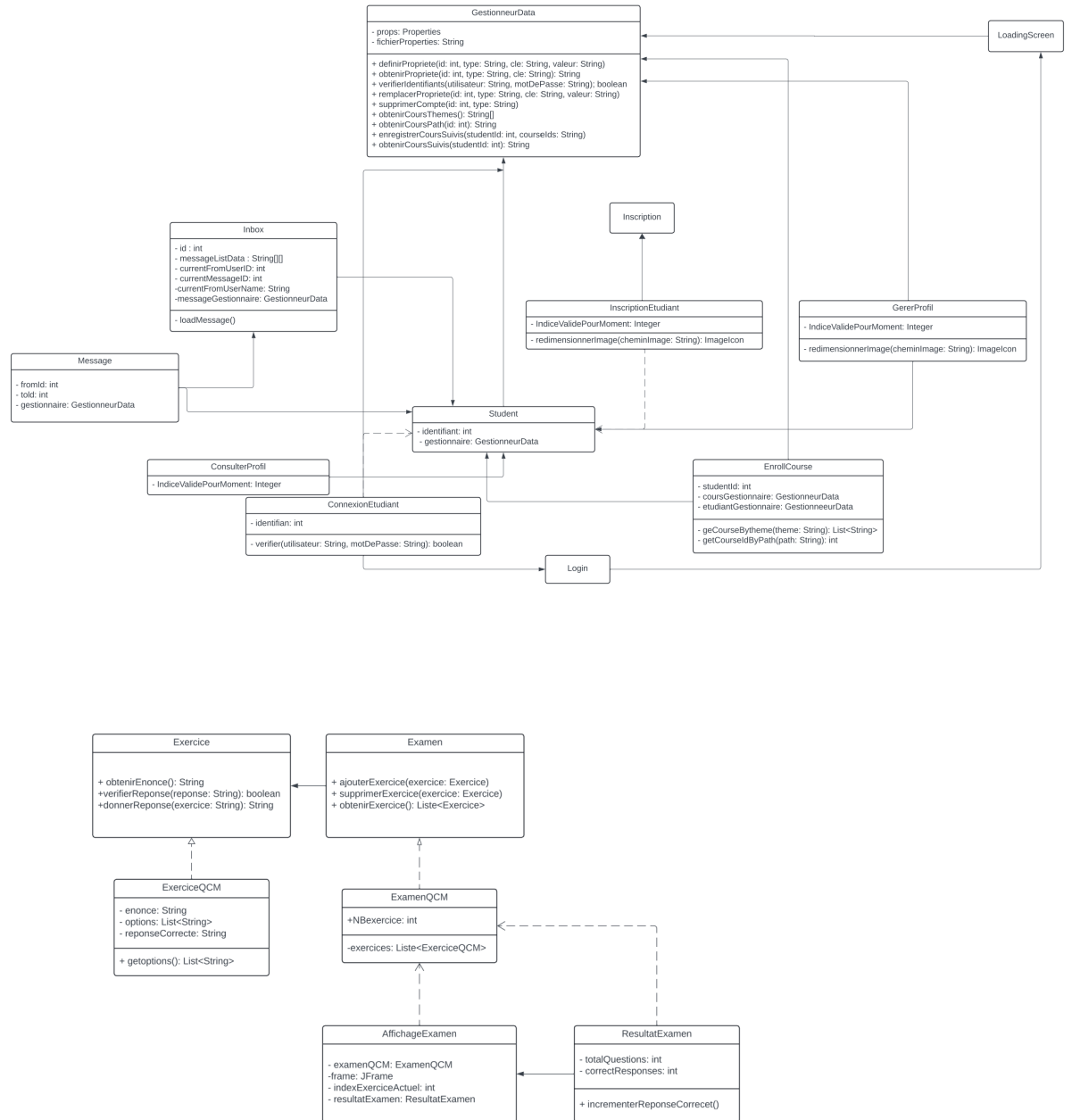


FIGURE 1 – Diagramme UML

4 Points forts de la plateforme

Un des points forts majeurs de notre projet est la capacité de stocker des informations dans plusieurs bases de données en fonction du type de données (cours, examens, messages, données des utilisateurs, etc.). Cette fonctionnalité est rendue possible grâce aux différentes méthodes implémentées dans la classe `GestionneurData`.

La méthode `enregistrerDonnees` permet de sauvegarder les données des utilisateurs dans un fichier de propriétés, assurant ainsi une gestion efficace et sécurisée des informations. La méthode `definirPropriete` facilite la mise à jour des propriétés spécifiques en définissant des paires clé-valeur et en enregistrant ces modifications.

```
1 public void enregistrerDonnees() {
2     try (OutputStream output = new
3         FileOutputStream(fichierProperties)) {
4         props.store(output, "Donnees des Utilisateurs");
5     } catch (IOException ex) {
6         System.out.println("Erreur lors de l'enregistrement
7             des donnees: " + ex.getMessage());
8         ex.printStackTrace();
9     }
10 }
11
12 public void definirPropriete(int id, String type, String
13     cle, String valeur) {
14     props.setProperty(type + "." + id + "." + cle, valeur);
15     enregistrerDonnees();
16 }
```

Ces méthodes permettent de gérer de manière flexible et efficace les différentes informations essentielles à l'application, garantissant une structure de données organisée et facilement accessible. Cette modularité dans le stockage des données améliore considérablement la robustesse et la fiabilité de notre système.

De plus, **des badges** sont désormais attribués aux utilisateurs en fonction du score qu'ils ont obtenu. Ces badges sont affichés dans la classe `ConsulterMonProfil`. Le score obtenu lors des examens est enregistré dans la base de données, assurant un suivi précis des performances des utilisateurs. Lorsque vous appuyez sur "Participants", un classement de tous les utilisateurs de la base de données est affiché, basé sur leurs scores. Ce classement permet de motiver les étudiants à améliorer leurs performances en créant une saine

compétition académique.

5 Organisation de l'Équipe et Méthodes Agiles

L'équipe a suivi des méthodes agiles pour organiser le développement du projet. Voici comment nous nous sommes organisés :

5.1 Réunions Hebdomadaires

L'équipe tenait des réunions hebdomadaires pour discuter de l'avancement du projet, des problèmes rencontrés et des solutions possibles. Ces réunions permettaient également de redéfinir les priorités et de répartir les tâches pour la semaine suivante.

5.2 Sprints

Le travail était divisé en sprints de deux semaines. À la fin de chaque sprint, une revue était effectuée pour évaluer les progrès réalisés et planifier le sprint suivant. Cette approche permettait de maintenir un rythme de développement constant et d'apporter des améliorations continues au projet.

5.3 Utilisation d'Outils de Gestion de Projet

Des outils comme Trello et Git étaient utilisés pour suivre les tâches et le code. Chaque membre de l'équipe avait des responsabilités claires, et l'avancement des tâches était régulièrement mis à jour dans l'outil de gestion. Cela facilitait la collaboration et assurait une transparence totale sur l'état d'avancement du projet.

5.4 Code Reviews

Les revues de code étaient effectuées pour s'assurer de la qualité du code et pour partager les connaissances au sein de l'équipe. Cela permettait également de détecter et de corriger les erreurs rapidement. Les revues de code favorisaient une culture de la qualité et de l'amélioration continue.

5.5 Communication

Une communication constante était maintenue via un groupe Discord dédié au projet, où les membres pouvaient poser des questions, partager des mises à jour et discuter des problèmes en temps réel. Cette communication régulière aidait à résoudre rapidement les problèmes et à maintenir une cohésion d'équipe.

5.6 Documentation

Une documentation complète et à jour était maintenue pour faciliter la prise en main du projet par les nouveaux membres de l'équipe et pour assurer la continuité du développement. La documentation incluait des guides d'installation, des manuels d'utilisation et des spécifications techniques détaillées.

5.7 Retrospectives

À la fin de chaque sprint, une rétrospective était organisée pour discuter des points forts et des points à améliorer. Les retours d'expérience étaient pris en compte pour optimiser les processus de travail et augmenter l'efficacité de l'équipe.

6 Difficultés rencontrées

Malheureusement, nous n'avons pas eu le temps de terminer entièrement notre plateforme. En particulier, nous n'avons pas encore implémenté la classe **Professeur**, bien que sa structure soit très similaire à celle des classes pour les étudiants, avec des fonctionnalités adaptées aux enseignants.

7 Conclusion

En conclusion, ce projet a permis de créer une application robuste et intuitive, offrant une variété de fonctionnalités destinées à enrichir l'expérience éducative des utilisateurs. Grâce à l'utilisation de bibliothèques telles que JavaSwing, nous avons développé une interface utilisateur conviviale et dynamique. Notre application se distingue par sa capacité à gérer efficacement les données des utilisateurs et

des cours, offrant une expérience d'apprentissage personnalisée et engageante.

Les méthodes de gestion agiles ont joué un rôle crucial, permettant une organisation efficace de l'équipe et une adaptation rapide aux besoins changeants du développement.

En somme, ce projet a jeté les bases d'une plateforme éducative prometteuse, capable de s'adapter aux besoins des étudiants et des enseignants tout en offrant un environnement d'apprentissage stimulant et interactif.