

Rapport Projet Java

TOSSOUKPE Kossi Serge
AMANKWAH Emmanuel

2025



Contents

1	Introduction	3
2	Description du Projet	3
3	Diagramme UML et Justifications	3
4	Manuel d'Utilisation	5
5	Organisation du Travail	6
6	Difficultés et solutions	6
7	Conclusion et Perspectives	6

1 Introduction

PasswordManager est une application de gestion de mots de passe développée en Java permettant aux utilisateurs de stocker et gérer leurs informations d'authentification de manière sécurisée. Les fonctionnalités incluent la génération de mots de passe forts et de longueurs choisies, le chiffrement des données, suppression d'utilisateur, consultation de la liste des utilisateurs par un administrateur, suppression de données par un utilisateur.

Membres du groupe et contributions :

- **Emmanuel AMANKWAH:**
Développement du module de sauvegarde et de chargement des données.
- **Kossi Serge TOSSOUKPE:**
Développement du module de chiffrement et de génération de mots de passe.

Chacun a un peu touché à tout notamment sur le débbugage, mais chacun s'est respectivement concentré sur les parties sus mentionnées.

2 Description du Projet

Objectifs :

- Permettre aux utilisateurs de stocker leurs mots de passe de manière sécurisée.
- Offrir une génération automatique de mots de passe robustes.
- Faciliter l'accès aux identifiants sans compromettre la sécurité.
- Implémenter des mesures cryptographiques pour la protection des données stockées dans l'ordinateur.

Technologies et langages :

Pour gérer le chiffrement, on a utilisé Cipher, SecretKeySpec, SecretKeyFactory, crypto.spec.PBEKeySpec, charset.StandardCharsets, security.SecureRandom et security.spec.KeySpec via l'API Java Cryptography Extension (JCE).

3 Diagramme UML et Justifications

Diagramme de classes UML :

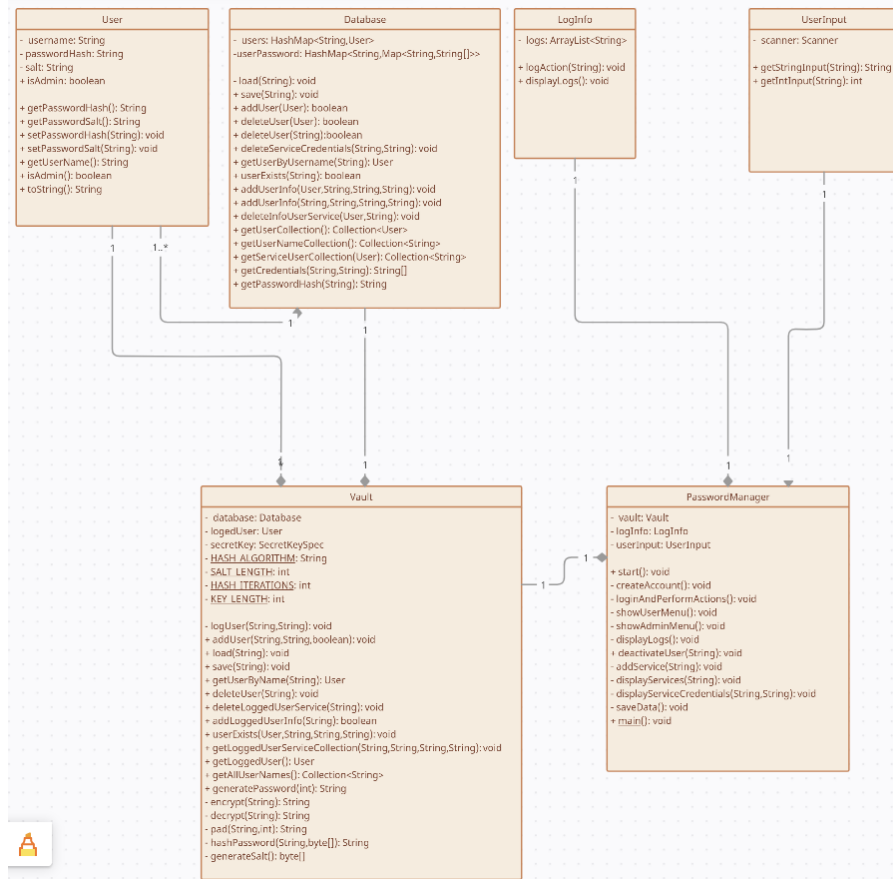


Figure 1: Diagramme de classes

Justifications des choix de conception :

- **PasswordManager** : classe principale de l'application. Elle se charge de présenter à l'utilisateur les actions possibles et de faire exécuter les bonnes procédures en fonction de ce que l'utilisateur a demandé.
- **DataBase** : Gère le stockage des données en cours d'exécution du programme et de les charger ou sauvegarder dans un fichier en début d'exécution du programme ou à la fin.
- **Vault** : Gère l'authentification des divers utilisateurs du programme et implémente les mesures cryptographiques pour stocker de façon sécurisée les diverses données sensibles.
- **UserInput** : classe utilitaire pour demander une information particulière à l'utilisateur
- **User** : Classe qui sert à contenir les informations pertinentes à chaque utilisateur durant l'exécution du programme.
- **LogInfo** : Classe qui devait servir à structurer et stocker les informations sur ce qui a été fait dans l'application pour permettre à un utilisateur administrateur d'avoir accès à un historique détaillé des actions menées sur l'application. (On ne s'en sert pas dans cette version du projet)

Les relations entre ces classes assurent une séparation claire des responsabilités et facilitent l'extensibilité du projet.

4 Manuel d'Utilisation

Interface utilisateur :

- Menu permettant de créer un compte d'utilisateur (simple/admin), de se connecter à un compte existant ou de quitter l'application.

```
Bienvenue dans SecurePasWordManager !

Menu:
1. Créer un compte
2. Se connecter
3. Quitter
Votre choix :
```

Figure 2: Menu 1

- Puis un second menu qui s'affiche après une connection reussie. Ce menu permet les actions suivantes pour un utilisateur simple:

```
Actions disponibles :
1. Ajouter une information de connexion pour un service
2. Afficher les identifiants et mots de passe de tout les services
3. Afficher l'identifiant et le mot de passe d'un service en particulier
4. Supprimer un service en particulier
q. Se déconnecter
Votre choix : █
```

Figure 3: Menu2

Et Pour un administrateur:

```
Actions disponibles :
1. Ajouter une information de connexion pour un service
2. Afficher les identifiants et mots de passe de tout les services
3. Afficher l'identifiant et le mot de passe d'un service en particulier
4. Supprimer un service en particulier
5. Supprimer un utilisateur
6. Afficher tout les utilisateurs
q. Se déconnecter
Votre choix : █
```

Figure 4: Menu3

Instructions d'installation et d'exécution :

1. Cloner le repository.
2. Compiler avec `javac -cp PasswordManager.java`
3. Exécuter avec `java PasswordManager`

Nota Bene:

Pour créer un compte administrateur on a besoin du code secret: "groupe hp".

Les données enregistrées seront sauvegardées dans un fichier (password_manager.txt) qui sera créé automatiquement.

5 Organisation du Travail

Approche de développement :

On a commencé avec un programme minimal viable dont les methodes se concentraient principalement dans la classe PasswordManager. Puis on a déplacé et adapté les methodes selon notre architecture de conception. On s'est assuré du développement de la classe de base de données avant d'intégrer les fonctionnalités d'authentification et de chiffrement.

6 Difficultés et solutions

Problèmes rencontrés :

- Difficulté à intégrer le module de chiffrement avec la base de données.
- Probleme avec le "salt" pour la fonction de hashage, choisir mode de chiffrement AES adapté entre ECB (Electronic Code Book) sans Padding, ECB (Electronic Code Book) avec Padding et CBC (Cipher Block Chaining) avec un IV (Initializer vector). On eu beaucoup de prolemes avec l'IV du CBC et le Padding du ECB qui ne matchaient pas quand on fait un déchiffrement.
- Gestion des erreurs d'authentification d'utilisateurs.

Solutions apportées :

- Pour le mode de chiffrement on a finalement adopté le mode ECB avec padding.
- Utilisation d'un encodage base64 pour le stockage des données chiffrées dans des String et dans un fichier.
- Insertion de messages d'erreur pour le débuge.

7 Conclusion et Perspectives

Résultats obtenus :

Le projet PasswordManager offre une solution fonctionnelle et sécurisée (confidentialité) pour la gestion des mots de passe.

Améliorations futures :

- Ajout d'une interface graphique.
- Ajout de logs permettant à un admin de savoir quel utilisateur a fait quoi et quand
- Integration d'une solution pour le remplissage automatique de champs.