GESTIONNAIRE DE MOTS DE PASSE : SECUREPASS

REALISE PAR : NGO MBEDEG LE-NYE ESPERANCE AUDREY

Cursus : Master 2 CYBER

I. Description Détaillée du Projet et Objectifs

Le projet consiste en la conception et le développement d'un gestionnaire de mots de passe sécurisé, destiné à offrir une solution pratique et fiable pour gérer les informations de connexion de manière sécurisée. Le gestionnaire de mots de passe sera conçu pour être autonome, il fonctionnera localement sur l'appareil de l'utilisateur, sans nécessiter de connexion Internet afin de garantir la confidentialité des données, offrant ainsi une solution pratique et sécurisée pour stocker et accéder aux informations de connexion.

Les informations seront stockées sur une base de données locale, l'objectif étant de créer une interface utilisateur conviviale permettant aux utilisateurs de stocker, d'organiser et d'accéder facilement à leurs identifiants et mots de passe, tout en assurant la sécurité des informations sensibles.

Le gestionnaire de mots de passe comprendra les fonctionnalités suivantes :

- La création d'un compte utilisateur avec un nom d'utilisateur unique et un mot de passe maître sécurisé. Il est possible de créer plusieurs comptes.

- Le stockage sécurisé des identifiants de connexion pour différents sites Web, applications ou services en ligne.

- L'utilisation de mots de passe forts avec un format de 14 caractères minimum composés d'au moins une minuscule, une majuscule, un chiffre et un caractère spécial.

- Une interface conviviale et sécurisée permettant à l'utilisateur de visualiser, ajouter, modifier et supprimer des entrées de mot de passe en vérifiant l'identité de l'utilisateur.

- Une sécurité renforcée par le chiffrement des données sensibles et l'utilisation de techniques de hachage pour stocker les mots de passe de manière sécurisée.

- L'option de récupération de mot de passe via une question secrète prédéfinie

- La gestion des erreurs au cas où un utilisateur existe déjà et si l'utilisateur se trompe de mot de passe ou de la réponse à la question secrète

- Une interface graphique intuitive et esthétique pour une expérience utilisateur optimale : ce point sera vu en dernier en tenant compte de sa faisabilité.

II. Planning du Projet

Le développement du projet sera planifié sur une période de 45 heures de cours dont 20h supplémentaires où nous travaillerons à la finalisation du rapport final. Le planning sera organisé comme suit :

- Semaine 1-2 : Analyse des besoins et des exigences fonctionnelles du gestionnaire de mots de passe.

- Semaine 3-4 : Apprentissage du langage python pour le développement du gestionnaire de mots de passe, en tenant compte de la logique applicative et une bibliothèque GUI pour l'interface utilisateur.

- Semaine 5-6 : Conception détaillée de l'architecture logicielle, y compris la modélisation des données, des fonctionnalités, des interactions utilisateur et la rédaction du cahier des charges complet.

- Semaine 7-8 : Développement des fonctionnalités essentielles du gestionnaire de mots de passe, en commençant par l'implémentation de la gestion des comptes utilisateurs et du stockage sécurisé des données.

- Semaine 9-10 : Intégration des fonctionnalités avancées telles que l'amélioration de l'interface utilisateur.

- Semaine 11-12 : Tests intensifs pour garantir la fiabilité, la sécurité et les performances du logiciel. Correction des bugs et optimisation des performances.

- Semaine 13 : Finalisation de la conception du projet avec une révision complète de toutes les fonctionnalités et une préparation pour la phase de présentation.

III. Organisation du Travail

En tant que développeur travaillant seule sur le projet, une organisation efficace du travail est essentielle pour atteindre les objectifs fixés dans le temps imparti. Pour ce faire, les tâches seront réparties de manière équilibrée sur les différentes périodes de travail, en tenant compte des exigences spécifiques de chaque étape du développement. Une planification minutieuse sera effectuée pour maximiser l'utilisation du temps disponible et garantir une progression régulière du projet.

La communication avec les différents intervenants du module de « Projet de Master » sera maintenue régulièrement lors des cours afin de discuter des progrès, résoudre les problèmes potentiels et obtenir des conseils et des retours d'expérience.

En outre, des séances de travail régulières seront planifiées, en alignement avec les plages horaires disponibles. Nous nous sommes fixés au moins une séance de 4h de travail deux fois par semaine pendant les heures de cours et quatre heures le weekend. Cela garantira une utilisation efficace du temps et permettra de maintenir un bon équilibre entre les exigences du projet et les autres engagements académiques.

IV. Evolution et Limites

1. *Partie Conceptuelle*

Au cœur de notre logiciel ***SecurePass*** se trouvent deux entités principales : **User** et **Password**. Chaque utilisateur, identifié par un **username**, peut stocker plusieurs enregistrements de mots de passe, chacun lié à des sites spécifiques. Pour garantir la sécurité, nous avons implémenté un système de hachage pour les mots de passe utilisateur avec un sel unique et une question secrète, solution de base pour la récupération du compte, anticipant les besoins de sécurité et de confidentialité.

Pour gérer les interactions utilisateur, nous avons développé **PasswordManagerApp**, l'interface entre l'utilisateur et la base de données. Cette classe contrôle non seulement le processus de connexion et de création de compte mais aussi la gestion des mots de passe stockés, offrant des fonctions pour ajouter, visualiser et supprimer les mots de passe.

Ci-dessous la Figure 1 qui illustre le modèle conceptuel de notre projet :

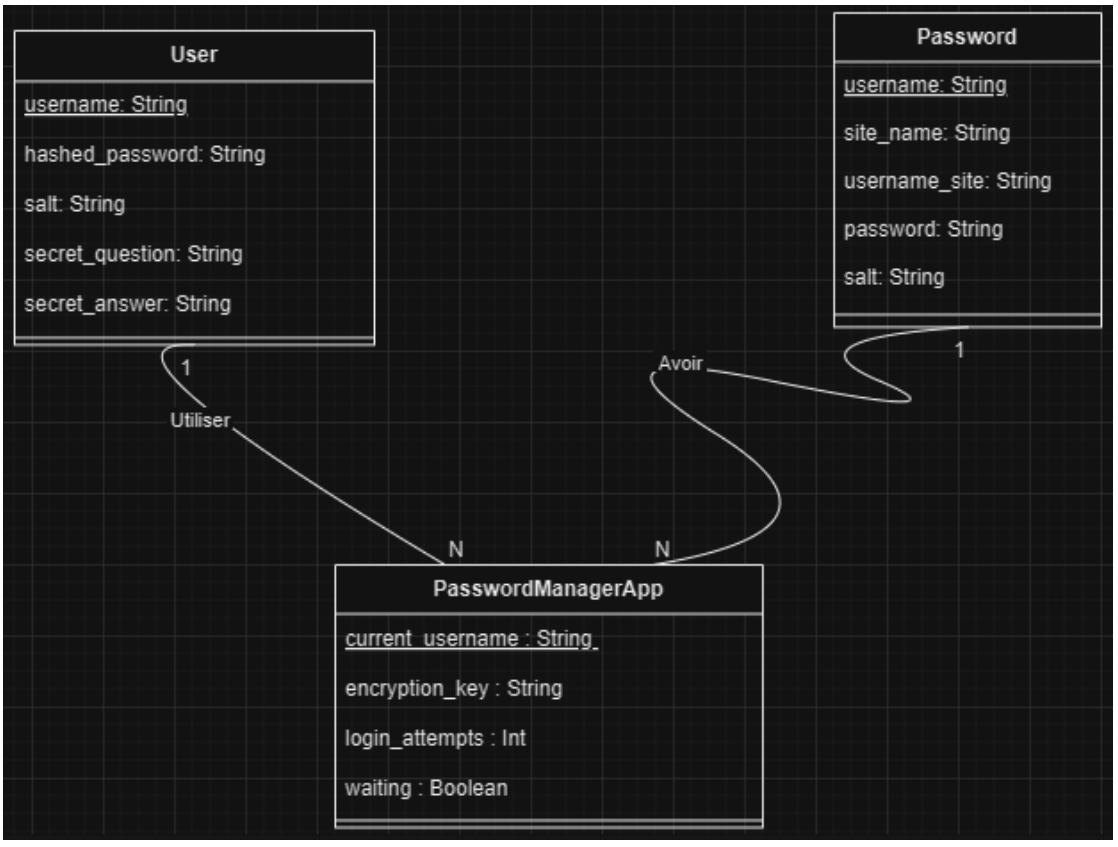

Figure 1: Modèle Conceptuel de SecurePass

Les méthodes regroupées autour du PasswordManagerApp sont les suivantes :

- **clear_screen()**: Supprime les widgets de la fenêtre principale.
- **show_login_screen()**: Affiche l'écran de connexion.
- **show_countdown(remaining)**: Affiche un compte à rebours en cas de multiples tentatives de connexion échouées.
- **ask_secret_question()**: Pose la question secrète à l'utilisateur.
- **login()**: Connecte l'utilisateur à l'application.
- **show_create_account_screen()**: Affiche l'écran de création de compte.
- **create_account()**: Crée un nouveau compte utilisateur.
- **show_main_menu()**: Affiche le menu principal de l'application.
- **add_password_ui()**: Affiche l'interface pour ajouter un nouveau mot de passe.
- **add_password()**: Ajoute un nouveau mot de passe à la base de données.
- **view_passwords()**: Affiche tous les mots de passe enregistrés par l'utilisateur.
- **delete_password_ui()**: Affiche l'interface pour supprimer un mot de passe existant.
- **delete_password()**: Supprime un mot de passe de la base de données.
- **logout()**: Déconnecte l'utilisateur et revient à l'écran de connexion.

➢ ***Relations et Interactions***

✓ ***Création de compte :*** L'utilisateur remplit les champs nécessaires et crée un nouveau compte. La validation des entrées et le hachage sécurisé du mot de passe sont effectués.

✓ ***Connexion :*** L'utilisateur saisit ses identifiants pour se connecter. Après trois tentatives échouées, une question secrète est posée.

- ✓ ***Gestion des mots de passe :*** Une fois connecté, l'utilisateur peut ajouter, visualiser et supprimer des mots de passe stockés pour différents sites.

- ✓ ***Déconnexion :*** L'utilisateur peut se déconnecter de l'application, ce qui efface les données temporaires et retourne à l'écran de connexion.

2. *Limites du projet*

Malgré ces avancées, nous reconnaissons que l'application a ses limites et qu'il faudra optimiser surtout la partie sécurisée aussi bien côté utilisateur que base de données.
Par exemple, bien que l'application utilise le cryptage pour protéger les mots de passe stockés, la clé de cryptage est générée et stockée localement, ce qui pourrait poser un risque si un attaquant parvient à y accéder. À l'avenir, nous envisageons d'explorer des méthodes plus sécurisées pour gérer les clés de cryptage, peut-être en utilisant un service de gestion des clés ou en implémentant un système de clé publique/privée.
De plus, la méthode de récupération de mot de passe amène des questions de sécurité et nous allons également travailler sur l'amélioration future de cette partie.
Enfin la partie front-end niveau UX Design n'est pas optimale et nous verrons également comment améliorer le visuel pour qu'au-delà de la praticité, le logiciel soit tout autant attrayant.