Dart / Flutter

Cahier des charges

Gestionnaire de mots de passe

Contexte

Dans le cadre de notre formation en développement web, Clément Prévost et moi-même, Lisa Tallet, devons développer un système permettant de stocker en toute sécurité les identifiants et mots de passe utilisés pour différents services en ligne. L'objectif principal est d'offrir un coffre-fort numérique où les utilisateurs peuvent stocker leurs informations de connexion de manière chiffrée.

Objectifs

- Création d'un compte individuel qui contiendra les données
- Permettre la création et la lecture de manière sécurisée de mot de passe par rapport à l'URL d'un site internet.
- Permettre d'après le lien renseigné par l'utilisateur d'accéder directement au site internet

Acteurs

Un seul acteur

- Utilisateur (1 seul rôle)

Périmètre et cas d'utilisation

- Enregistrer des mots de passe liés à des URL
- Voir les mots de passe
- Aller directement sur le site via l'URL renseigné
- Se connecter
- Se déconnecter
- Créer un compte

Règles métier

- Le stockage des mots de passe doit être totalement sécurisé pour ne pas dévoiler les mots de passe à des individus malveillants
- La confidentialité sur l'interface graphique doit être renforcée
- Les mots de passes doivent être cryptés et stockés dans une base de données

Exigences fonctionnelles

- Le système doit permettre à l'utilisateur de créer un compte avec un identifiant unique et un mot de passe sécurisé. L'utilisateur doit pouvoir se connecter et se déconnecter de son compte.
- Le système doit permettre à l'utilisateur de créer, lire, modifier et supprimer des mots de passe associés à des URL de sites internet. Chaque mot de passe doit être associé à une URL unique.
- Le système doit permettre à l'utilisateur d'accéder directement au site internet associé à un mot de passe en cliquant sur l'URL correspondante.
- Le système doit stocker les mots de passe de manière sécurisée dans une base de données cryptée. Les mots de passe ne doivent jamais être stockés en clair.
- Le système doit renforcer la confidentialité en masquant les mots de passe sur l'interface graphique. L'utilisateur doit avoir la possibilité de les afficher temporairement s'il le souhaite.

Exigences non fonctionnelles

- Le système doit être facile à installer pour les utilisateurs, avec une procédure d'installation simple.
- Le système doit être convivial et simple à utiliser, avec une interface utilisateur intuitive.
- Le système doit être conçu pour fonctionner sur différentes plateformes (Windows, macOS, Linux).
- Le système doit offrir des performances optimales, avec des temps de réponse rapides.
- Le système doit être facile à maintenir et à mettre à jour, avec une documentation claire.

Contraintes

- Le système doit respecter les normes de sécurité les plus élevées pour protéger les données des utilisateurs contre les accès non autorisés et les attaques malveillantes.
- Le système doit être développé en utilisant les langages de programmation Dart et Flutter, ce qui nécessite des compétences et des connaissances spécifiques dans ces domaines.
- Le système doit être compatible avec les principaux navigateurs web) et les différents systèmes d'exploitation (Windows, macOS, Linux, etc.).

Spécifications techniques

Base de donnée :

Stockage des utilisateurs

- id (Int), username (String), password (String : hashPassword)

Stockage des mots de passe

- id (Int), site_name (string), site_url (String), password (String : crypt password), id_user (Int)

Librarie(s) :

Utilisation pour la cryptographie de la librairie : https://pub.dev/packages/cryptography