

WEB SPHERE(FLUTTER)

Rapport projet :

***Password manager***

Asmae MOUBARRIZ

Achraf AKRACHE

5iir7-G1

Table des matières

[Introduction 2](#_Toc185587223)

[Contexte du projet : 2](#_Toc185587224)

[Objectifs du projet : 2](#_Toc185587225)

[Méthodologie utilisée : 2](#_Toc185587226)

[Chapitre 1 : Conception et développement de l’application 3](#_Toc185587227)

[Étude des besoins et fonctionnalités à intégrer 3](#_Toc185587228)

[Gestion des utilisateurs 3](#_Toc185587229)

[Gestion des mots de passe 3](#_Toc185587230)

[Choix des technologies 4](#_Toc185587231)

[Structure de l’application 4](#_Toc185587232)

[Chapitre 2 : Fonctionnalités principales et démonstration 4](#_Toc185587233)

[Présentation des fonctionnalités implémentées 4](#_Toc185587234)

[1. Connexion et authentification utilisateur 4](#_Toc185587235)

[2. Liste des comptes enregistrés 4](#_Toc185587236)

[3. Ajout de compte avec génération automatique de mots de passe 5](#_Toc185587237)

[4. Mise à jour et suppression des comptes 5](#_Toc185587238)

[5. Chiffrement des mots de passe 5](#_Toc185587239)

[Chapitre 3 : Défis, solutions et résultats 5](#_Toc185587240)

[Défis rencontrés 5](#_Toc185587241)

[Solutions adoptées 5](#_Toc185587242)

[Résultats obtenus 5](#_Toc185587243)

[Conclusion 6](#_Toc185587244)

[Résumé des points clés 6](#_Toc185587245)

[Perspectives d’amélioration 6](#_Toc185587246)

# Introduction

## Contexte du projet :

Pourquoi une application de gestion de mots de passe ?

Dans un monde où les utilisateurs doivent gérer de nombreux comptes en ligne pour des services tels que les réseaux sociaux, les plateformes professionnelles ou encore les achats en ligne, il devient essentiel de garantir la sécurité des mots de passe. Cependant, la majorité des utilisateurs utilisent des mots de passe simples, souvent réutilisés sur plusieurs plateformes, ce qui augmente les risques de piratage et de vol de données.

Pour répondre à ces défis, l’idée d’une application de gestion de mots de passe est née. Ce projet vise à offrir aux utilisateurs un outil simple, sécurisé et pratique pour gérer leurs mots de passe de manière centralisée et automatisée. Il garantit également une protection des données personnelles grâce à l’utilisation d’algorithmes de chiffrement modernes.

## Objectifs du projet :

Sécurisation et gestion des données personnelles

L’objectif principal de ce projet est de concevoir une application répondant aux attentes suivantes :

• Offrir un moyen sûr pour stocker et gérer les identifiants des utilisateurs.

• Automatiser la génération de mots de passe forts, répondant aux normes de sécurité modernes.

• Proposer une interface intuitive et ergonomique pour une utilisation simplifiée.

• Garantir la confidentialité et la sécurité des données grâce au chiffrement des mots de passe.

## Méthodologie utilisée :

Collaboration et outils techniques

Ce projet a été réalisé en binôme, mettant en avant un travail collaboratif basé sur :

• Les cours et travaux pratiques suivis en classe, qui ont fourni une base théorique solide.

• La documentation technique officielle de Flutter et Firebase, utilisée pour approfondir les connaissances et résoudre des problèmes spécifiques.

• Git pour la gestion du code source et la facilitation de la collaboration. Grâce à Git, nous avons pu partager, synchroniser et gérer efficacement les modifications apportées au projet, réduisant ainsi les conflits de version.

Cette méthodologie nous a permis de structurer et d’organiser notre travail pour aboutir à une application fonctionnelle et sécurisée.

# Chapitre 1 : Conception et développement de l’application

## Étude des besoins et fonctionnalités à intégrer

Avant de commencer le développement, nous avons mené une analyse approfondie des besoins des utilisateurs. Les fonctionnalités suivantes ont été identifiées comme essentielles :

• Authentification sécurisée pour protéger l’accès aux données.

• Gestion complète des comptes enregistrés, incluant la création, l’affichage, la mise à jour et la suppression.

• Génération automatique de mots de passe forts, pour encourager les utilisateurs à ne pas réutiliser les mêmes mots de passe.

• Affichage sécurisé des mots de passe, masqués par défaut pour éviter tout accès non autorisé en cas de consultation publique.

Gestion des utilisateurs (authentification et comptes)

L’authentification est au cœur de l’application. Les utilisateurs peuvent :

• Créer un compte avec une adresse email et un mot de passe sécurisé.

• Se connecter pour accéder à leurs informations.

Ces données sont stockées dans Firebase Authentication, garantissant une protection renforcée grâce à des règles de sécurité.

### Gestion des mots de passe

Une fois connectés, les utilisateurs peuvent :

• Ajouter des comptes en renseignant des informations comme :

• Nom de l’application (ex. : Gmail).

• Plateforme associée (ex. : Web, Mobile).

• Identifiant de connexion (ex. : user123@gmail.com).

• Mot de passe, généré automatiquement ou saisi manuellement.

• Mettre à jour ou supprimer des comptes existants.

Tous les mots de passe sont chiffrés avant leur stockage dans Firebase, garantissant leur confidentialité même en cas d’accès non autorisé à la base de données.

## Choix des technologies

• Flutter : utilisé pour développer une interface utilisateur intuitive et multiplateforme (iOS et Android).

• Firebase : choisi pour ses fonctionnalités d’authentification, de base de données en temps réel, et de gestion des utilisateurs.

## Structure de l’application

L’application est organisée en trois couches principales :

• Interface utilisateur : Écrans de connexion, liste des comptes, et formulaires.

• Logique métier : Gestion des fonctionnalités (CRUD) et génération automatique de mots de passe.

• Firebase : Stockage et sécurisation des données.



# Chapitre 2 : Fonctionnalités principales et démonstration

## Présentation des fonctionnalités implémentées

### 1. Connexion et authentification utilisateur

• Les utilisateurs saisissent leur email et mot de passe pour accéder à l’application. En cas d’erreur, un message clair s’affiche.

• Exemple : Connexion d’un utilisateur enregistré avec l’email example@gmail.com.

### 2. Liste des comptes enregistrés

• Présentation des comptes sauvegardés sous forme de liste.

• Exemple :

• Application : Gmail.

• Identifiant : user123@gmail.com.

• Mot de passe : \*\*\*\*\*\*\*\* (masqué).

### 3. Ajout de compte avec génération automatique de mots de passe

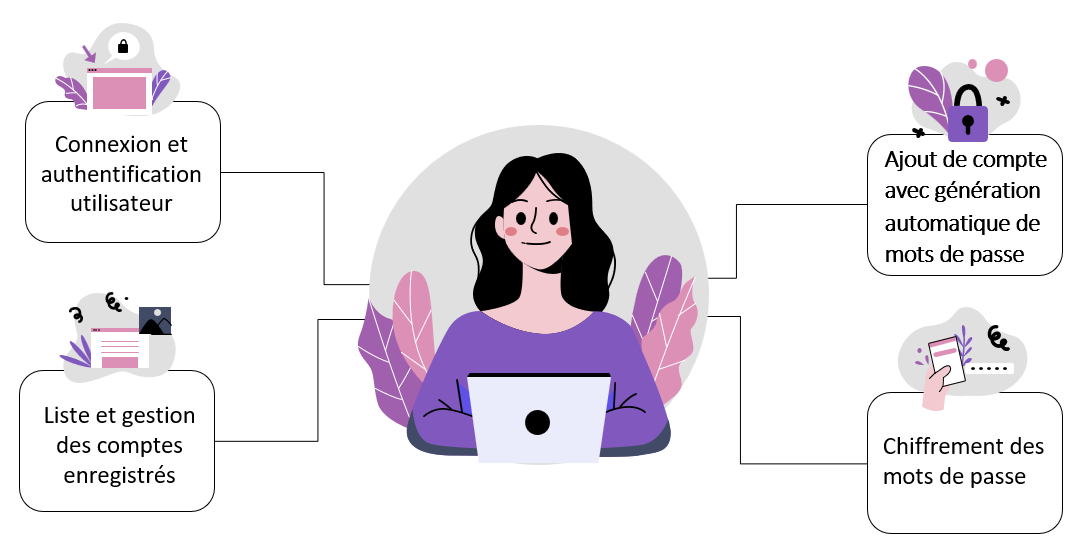
• Fonctionnalité permettant de générer un mot de passe complexe et unique, tel que : Ht@93X!kz.

### 4. Mise à jour et suppression des comptes

• Exemple : Modification d’un mot de passe ou suppression d’un compte inutilisé.

### 5. Chiffrement des mots de passe

• Les mots de passe sont chiffrés avant leur enregistrement dans Firebase, garantissant leur sécurité même en cas d’accès non autorisé à la base de données.



# Chapitre 3 : Défis, solutions et résultats

## Défis rencontrés

1. Intégration de Firebase avec Flutter

• Configuration initiale complexe, nécessitant des ajustements dans les fichiers de projet.

2. Mise en place du chiffrement des mots de passe

• Recherche d’une bibliothèque de chiffrement adaptée à Flutter.

3. Collaboration en binôme avec Git

• Gestion des conflits de fusion lors de la synchronisation des modifications.

## Solutions adoptées

• Utilisation de tutoriels et de documentation pour comprendre la configuration de Firebase.

• Adoption de bibliothèques comme encrypt pour implémenter le chiffrement.

• Communication régulière et gestion des branches pour éviter les conflits Git.

## Résultats obtenus

• Une application fonctionnelle et sécurisée.

• Une base solide pour intégrer des fonctionnalités supplémentaires, comme l’authentification biométrique.

# Conclusion

## Résumé des points clés

Ce projet nous a permis de concevoir une application complète, combinant :

• Sécurité grâce au chiffrement et à Firebase.

• Ergonomie grâce à Flutter.

• Collaboration efficace avec Git.

Bilan personnel

Nous avons développé des compétences en :

• Développement d’applications mobiles multiplateformes.

• Gestion de bases de données et intégration de services cloud.

• Travail collaboratif et gestion de projet.

## Perspectives d’amélioration

• Ajout de fonctionnalités avancées, telles que :

• Authentification biométrique.

• Partage sécurisé de mots de passe.

• Optimisation de l’interface utilisateur pour une expérience encore plus fluide.

En conclusion, ce projet a été une expérience enrichissante, tant sur le plan technique que sur le plan professionnel, nous préparant à relever des défis plus complexes dans le futur.