

Password Manager

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	6
1.2	Objectifs.....	6
1.3	Planification initiale	7
2	Analyse / Conception.....	7
2.1	Concept	7
2.2	Stratégie de test.....	9
2.3	Risques techniques	9
2.4	Planification	9
2.5	Dossier de conception	10
3	Réalisation.....	10
3.1	Dossier de réalisation	10
3.2	Description des tests effectués	11
3.3	Erreurs restantes	11
3.4	Liste des documents fournis	11
4	Conclusions	12
5	Annexes.....	23
5.1	Résumé du rapport du TPI / version succincte de la documentation	26
5.2	Sources – Bibliographie.....	26
5.3	Journal de travail	27
5.4	Manuel d'Installation	27
5.5	Manuel d'Utilisation.....	27
5.6	Archives du projet.....	27

1 **Spécifications**

1.1 **Titre**

Password Manager ou en français, « gestionnaire de mots de passe ».

1.2 **Description**

A compléter, par une explication du contexte, de la situation, des raisons générales de la mise en route d'un tel projet.
Le lecteur doit pouvoir comprendre les motivations du lancement du projet...

La gestion de mot de passe représente un enjeu important au quotidien. En effet, les mots de passe sont là pour protéger des données en ligne, limiter l'accès à celles-ci, qu'importe leur niveau de sensibilité, le but principal est de garantir que seul une personne ou une poignée de personnes puissent lire ces données.

Les logiciels, applications, forums et autres plateformes en ligne, nous offrent pour la majeure partie d'eux cette option : « créer un compte » pour revenir sur nos données personnelles, les modifier ou les supprimer si on le souhaite. De ce fait, une personne peut se retrouver avec une multitude de comptes en ligne, dont il devra se rappeler les identifiants de login pour l'utiliser.

Notre logiciel a pour but, d'alléger l'utilisateur de cette tâche de mémorisation de mots de passe (demande entreprise pour ses employés).

En plus de servir à centraliser les mots de passe des différents sites internet consultés par l'utilisateur au sein d'une entreprise.

Grâce à ce projet l'utilisateur en apprendra plus sur les recommandations pour gérer ses mots de passe.

Contexte : problème de mots de passe récurrent dans une PME (notation sur post-its)

Plus : centralisation mots de passe, génération mots de passe et authentification.

L'authentification impliquant un compte, nous laissons la possibilité de créer un compte.

1.3 **Matériel et logiciels à disposition**

A compléter par ce qui est nécessaire pour le démarrage ...

Windows – pour le travail global

Docker – pour démarrer nos conteneurs phpmyadmin et mysql.

Visual Studio 2022 – pour implémenter notre code.

Word – Pour documenter le rapport.

Excel – pour la planification,

Un utilisateur peut être admin ou utilisateur lambda.

En tant qu'admin, l'utilisateur a la possibilité d'accéder aux mots de passe de tous les employés de l'entreprise. Ceux-ci en cas d'oubli de leur mot de passe peuvent le contacter pour récupérer celui-ci mais celui-ci devra ensuite être modifié. On peut imaginer que l'administrateur bien qu'ayant accès à toutes les données des utilisateurs, ne peut en réalité les voir que si un employé lui a fait une requête pour récupérer son mot de passe et c'est le ticket de celui-ci qui créerait.

1.4 Prérequis

A compléter par une description des compétences, des connaissances et de la formation minimum pour être à même de réaliser le projet ...

La réalisation de ce projet nécessite la connaissance et la maîtrise des modules ICT ci-dessous vus au cours de ma formation :

104 : implémenter un modèle de données

105 : Traiter une base de données avec SQL

114 : Mettre en œuvre des systèmes de codification, de compression et d'encryptage

120 : Développer et implémenter des interfaces graphiques d'utilisateurs pour une application en tenant comptes des exigences ergonomiques et des standards.

226A : Mettre en œuvre sans hérédité les exigences dans un design de classes, implémenter, documenter et tester avec un langage orienté objets.

Ces différentes notions devront s'accorder au. Grâce à la préparation au TPI, la réalisation de ce projet me semble atteignable.

1.5 Cahier des charges

1.5.1 Objectifs et portée du projet (objectifs SMART)

A compléter. Il s'agit d'ébaucher des réponses aux questions de l'acronyme CQQCOQP (Combien, Quoi, Qui, Comment, Où, Quand, Pourquoi)

1.5.2 Caractéristiques des utilisateurs et impacts

A compléter... Il s'agit d'identifier le(s) profil(s) de(s) utilisateur-trice(s) type, et les conséquences que cela va avoir sur la conception (couleurs, ergonomie, utilisation, etc.)

1.5.3 Fonctionnalités requises (du point de vue de l'utilisateur)

A compléter par une espèce de mode d'emploi du produit. S'il s'agissait d'une montre, décrire qu'à part l'heure, il y aura la possibilité d'utiliser un chronomètre, un réveil, ...

1.5.4 Contraintes

Sécurité, backups, disponibilité, système utilisé, interfaces avec autres logiciels, etc.

Ce projet étant fondamentalement basé sur la sécurité. Les données devront être sauvegardées de manière sécurisée, les interactions avec la base de données devront être précautionneusement exécutées, l'accès aux données d'un utilisateur devra faire l'objet d'une authentification.

1.5.5 Travail à réaliser par l'apprenti

Décrire à quoi doit ressembler le travail produit, ce qu'il faudra rendre ...

Le produit attendu est un logiciel qui centralise

1.5.6 Si le temps le permet ...

Objectifs complémentaires au cas où le projet n'est pas assez ambitieux dans le temps imparti...

En plus des éléments demandés à la base dans le cahier des charges, avant validation, ces éléments ont été proposés et finalement acceptés :

La mise en place d'une option d'aide pour permettre à l'utilisateur de comprendre le but de l'application et l'utilité de chaque page sur lesquelles il naviguera.

La mise en place d'une option pour se déconnecter.

La mise en place d'une option pour supprimer une ligne de données du tableau qui affiche les données de mots de passe sauvegardés.

Si le temps me le permet, je trouverais utile d'ajouter des flèches aux bas de la page intitulée « Mon coffre ».

1.5.7 Méthodes de validation des solutions

Comment les tests vont être entrepris, quels tests doivent être entrepris, etc....

1.6 Les points suivants seront évalués

- Le rapport
- Les planifications (initiale et détaillée)
- Le journal de travail
- Le code et les commentaires
- Les documentations de mise en œuvre et d'utilisation

1.7 Validation et conditions de réussite

- Compréhension du travail
- Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
- Etat de fonctionnement du produit livré

2 Analyse préliminaire

2.1 Introduction

Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.

Ces éléments peuvent être repris des spécifications de départ.

Ce projet est réalisé dans le cadre d'un « travail pratique individuel » (TPI). Il porte sur la gestion de mots de passe dans une application prévue à cet effet.

2.2 Objectifs

Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.

Ces éléments peuvent être repris des spécifications de départ.

2.3 Planification initiale

Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées. Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme.

Ces éléments peuvent être repris des spécifications de départ.

3 Analyse / Conception

3.1 Concept

Le concept complet avec toutes ses annexes:

Par exemple :

- *Multimédia: carte de site, maquettes papier, story board préliminaire, ...*
- *Bases de données: interfaces graphiques, modèle conceptuel.*
- *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle...*
- *...*

3.1.1 Maquettes

Les maquettes de notre application ont été réalisées depuis le logiciel « Figma ». Pour représenter notre application, nous en avons faits 6. Elles montrent l'interface de notre application, de son point d'entrée « Authentification » à ses fonctionnalités sous-jacentes : « Générer un mot de passe », « Sauvegarder un mot de passe » et la gestion des données qui auraient été enregistrées dans « Mon coffre ».

La page « Générer un mot de passe » a deux maquettes. Celle qui affiché par défaut, et celle qui est montré à l'utilisateur si l'option « Vérifier la force du mot de passe » est cochée.

Par ces maquettes, on voit également qu'il y a des options qui ne sont visibles qu'à la suite de déclencheurs comme l'appui sur le bouton (d'aide), ou sur le bouton du menu qui affiche les différentes options de navigation.

Pour accéder à l'application l'utilisateur devra soit se connecter à l'application avec des données valides, soit créer un compte conformément aux exigences des bonnes pratiques qu'il aura reçu.

3.1.2 MCD

Notre MCD "Modèle conceptuel des données", se résume en deux entités.

D'un côté, nous avons l'entité "USER", de l'autre l'entité "WEBSITE".

Notre utilisateur est défini par son identifiant, son nom d'utilisateur, son mot de passe maître, son sel et son rôle.

L'attribut "administrator" étant de type booléen, on pourra savoir si l'utilisateur en question a le rôle administrateur ou non.

Un site internet lui est caractérisé par son identifiant, son nom, le nom d'utilisateur utilisé et un mot de passe.

Les deux dernières informations représentent les données de connexion de l'utilisateur de la table "USER".

Ces deux entités sont liées par le type d'association "manage". Ce verbe contient toutes les actions possibles de l'utilisateur :

ajouter, modifier et supprimer des données d'un site internet.

L'entité "USER" est une entité indépendante, tandis que l'entité "WEBSITE" est dépendante de l'entité "USER".

Pour qu'il y ait un site internet, il faut donc qu'il y ait un utilisateur derrière.

Cette relation entre les deux peut résumer en : "un utilisateur gère 0 ou plusieurs site internet" et "un site internet est géré par un ou plusieurs utilisateurs".

Un site internet peut être géré par plusieurs utilisateurs dans le cas, où il y aurait un utilisateur avec un rôle administrateur à 1.

Langage compréhensible pour tous.

Les identifiants des deux tables sont des clés artificielles auto incrementées.

3.1.3 MLD

Langage orienté code.

Le modèle logique des données (MLD), nous permet d'écrire nos données comme nous aimerions les retrouver dans notre code.

Après le MCD, ce modèle varie donc légèrement pour s'adapter aux normes.

Sont mis en gras

3.1.4 MPD (script)

Le MCD et le MLD nous permettent de mieux préparer notre code. Grâce à leur visuel, on peut voir si les données sont logiques, cohérents

et générer ensuite un code adapté au langage visé. Voir démarche ci-dessous :

3.2 Méthode de gestion de projet

La méthode de gestion de projet qui a été utilisée pour ce projet est la méthode des 6 pas, également connu sous le nom de Waterfall.

Celle-ci comprend les étapes ci-dessous :

1. S'informer
2. Planifier
3. Décider
4. Réaliser
5. Contrôler
6. Evaluer

Celle-ci a été choisie car elle est adaptée à la demande de ce projet, c'est-à-dire, réaliser un travail individuel.

La segmentation du projet en tâches distinctes a d'abord permis l'utilisation de cette méthode.

Si on regarde de près nos différentes tâches, celles-ci sont préfixées des termes : analyse, doc, conception, réalisation, test.

L'analyse correspond à l'étape d'information et de planification du projet. Dans cette étape, nous réfléchissons

Les tests permettent de contrôler le code fait. Bien qu'il n'ait pas de tâches prévues à cet effet, après le contrôle du code,

La documentation est une étape annexe qui permet de consigner notre travail.

3.3 Stratégie de test

Décrire la stratégie globale de test:

- *types de des tests et ordre dans lequel ils seront effectués.*
- *les moyens à mettre en œuvre.*
- *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
- *données de test à prévoir (données réelles ?).*
- *les testeurs extérieurs éventuels.*

3.4 Risques techniques

- *risques techniques (complexité, manque de compétences, ...).*

Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, ...).

Comme risques identifiés, il n'y a en qu'un, le calcul de la force du mot de passe. En recevant le cahier des charges et bien des jours après cette étape semblait simple à exécuter jusqu'à sa réalisation.

Le calcul de la force du mot de passe doit tenir compte de plusieurs paramètres (la longueur du mot de passe, la présence de nombres, de lettres majuscules et de caractères spéciaux) à échelonner, d'où le risque.

Pour réduire le risque de ce point technique, un barème a été créé en se basant sur une **heatmap**.

D'autres éléments qui auraient pu causer un problème comme le hachage du mot de passe avec un grain de sel, la génération d'un mot de passe aléatoire et les requêtes MySQL, ont été écartés. Le premier, car il avait été vu dans la phase pré-TPI, le deuxième car la logique coulait de source, le dernier pour les mêmes raisons que le premier.

3.5 Planification

Révision de la planification initiale du projet :

- *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
- *partage des tâches en cas de travail à plusieurs.*

*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le*

responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.

3.6 Dossier de conception

Fournir tous les document de conception:

- *le choix du matériel HW*
- *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
- *le choix des outils logiciels pour la réalisation et l'utilisation*
- *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...*
- *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
- *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...*

Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !

4 Réalisation

4.1 Dossier de réalisation

Décrire la réalisation "physique" de votre projet

- *les répertoires où le logiciel est installé*
- *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
- *les versions des systèmes d'exploitation et des outils logiciels*
- *la description exacte du matériel*
- *le numéro de version de votre produit !*
- *programmation et scripts: bibliothèques externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

NOTE : Evitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...

1. Création d'un git structuré.

- **Etapas du git :**

```
echo "# E_TodoList" >> README.md  
git init
```

```
git add .  
git commit -m "docs : initialize the repository"  
git branch -M main  
git remote add origin  
https://github.com/CielRii/E_TodoList.git  
git push -u origin main
```

2. Création des 6 pages de l'application.
3. Création des modèles MCD, MLDP.
4. Création du script MPD.
5. Script d'initialisation de la base de données.
6. Mise en place du modèle MVC.
7. Implémentation des fonctionnalités de l'application.

4.2 Description des tests effectués

Pour chaque partie testée de votre projet, il faut décrire:

- les conditions exactes de chaque test
- les preuves de test (papier ou fichier)
- tests sans preuve: fournir au moins une description

4.3 Erreurs restantes

S'il reste encore des erreurs:

- Description détaillée
- Conséquences sur l'utilisation du produit
- Actions envisagées ou possibles

4.4 Liste des documents fournis

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

- le rapport de projet
- le manuel d'Installation (en annexe)
- le manuel d'Utilisation avec des exemples graphiques (en annexe)
- autres...

5 Conclusions

Développez en tous cas les points suivants:

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

5.1 Bilan des fonctionnalités demandées

- Il s'agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.
- Si ce n'est pas le cas, estimer en « % » ou en « temps supplémentaire » le travail qu'il reste à accomplir pour terminer le tout.

5.2 Bilan de la planification

- Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différences entre les planifications initiales et détaillées avec le journal de travail.

5.3 Bilan personnel

- Si c'était à refaire:
 - Qu'est-ce qu'il faudrait garder ? Les plus et les moins ?
 - Qu'est-ce qu'il faudrait gérer, réaliser ou traiter différemment ?
- Qu'est-ce que ce projet m'a appris ?
- Suite à donner, améliorations souhaitables, ...
- Remerciements, signature, etc.

Si c'était à refaire, j'hésiterais moins à reprendre des codes précédents. Pour ce projet j'ai eu un peu de mal à clairement le faire, par peur que ça soit contraire à ce qui est attendu et de donner une impression de triche.

Les pages « se connecter » et « créer un compte » avaient déjà été complètement réalisés, de même que ma procédure MySQL de hachage de mot de passe.

Ces éléments contrairement aux requêtes me semblaient moins modifiables et trop gros pour les reprendre tels quels.

Quand on regarde les sources, il y a une différence entre celles notées ici et celles du journal de travail.

Dans notre journal de travail, nous avons notées celles qui nous paraissaient pertinentes sur le moment, mais avec des essais et

la décision d'explorer d'autres voies, ces sources ont été délaissées au profit de sources plus simples. Pour cette raison les sources

mentionnées dans la partie bibliographie sont celles.

Il semblait important pour nous de noter également les documents vus en cours.

J'ai privilégié cette source d'information, pour donner du poids au contenu vu en cours et également éviter d'aller sur internet.

Les différents modèles que nous verrons correspondent aux normes de l'ETML, de même que la manière de nommer les variables

dans les différents bouts de code que vous verrez tout au long de ce rapport.

Les « liens essentiels » pour le TPI n'ont pas été pris en compte.

Bonnes pratiques

Sauvegarde du mot de passe pas contrôlée, mais pour plus de sécurité, nous vous recommandons de suivre cette démarche :

- Longueur : 15 caractères minimum
- Pour le rendre plus robuste, plus complexe : car

Astuce pour créer un mot de passe facile mémoriser : créer une phrase, enlever des lettres dans cette phrase, rajouter des chiffres, caractères spéciaux, etc.

Fonctions CRUD

Les requêtes vers la base de données ne sont possibles que grâce à l'extension.

Pour ça nous utilisons les variables ci-dessous :

```
13 références
public MySqlConnection Connection { get; set; }
private MySqlCommand cmd;
private MySqlDataReader dataReader;
```

Pour se connecter à la base de données, nous avons cette classe.

```
// Connection to the database
9 références
public bool IsConnect()
{
    if (Connection == null)
    {
        string connstring = "server='localhost'; port='6033'; database='setup_database'; UID='root'; password='root'";

        try
        {
            Connection = new MySqlConnection(connstring);
            Connection.Open();
        }
        catch (Exception e)
        {
            MessageBox.Show(e.Message);
        }
    }
    return true;
}
```

Elle retourne une valeur booléenne qui peut être négative ou positive.

Dans la variable « connstring » sont précisés le nom du serveur, le numéro de port de celui-ci, la base de données à utiliser, le nom de l'utilisateur et son mot de passe. Grâce à l'identification de l'utilisateur, on reçoit ses privilèges.

IMAGE À INSÉRER

Une autre classe est instanciée pour se déconnecter :

```
// Disconnection from the database
4 références
public void DisConnect()
{
    Connection.Close();
}
```

Maintenant, un bref aperçu sur l'implémentation des fonctions CRUD :

Dans notre programme, nous créons des utilisateurs mais aussi des données de mot de passe à enregistrer.

La création de nouvelles données se fait grâce à une classe qui suit cet ordre :

```
// Create a new user
1 référence
public bool CreateUser(string username, string masterPassword, byte[] salt)
{
    if (!IsConnect()) return false;

    string query = "INSERT INTO `t_user`(user_id, username, masterPassword, salt) VALUES(NULL, @username, @masterPassword, @salt);";

    cmd = new MySqlCommand(query, Connection);
    cmd.Parameters.AddWithValue("@username", username);
    cmd.Parameters.AddWithValue("@masterPassword", masterPassword);
    cmd.Parameters.AddWithValue("@salt", salt);
    cmd.Prepare();
    cmd.ExecuteNonQuery();
    return true;
}
```

Si la connexion à la base de données est validée (retourne « true »), la requête est exécutée.

Celle-ci suit le pattern « INSERT INTO... VALUES ».

Ajout des valeurs sécurisées grâce à la séparation des paramètres de leurs valeurs. Cette méthode permet de protéger nos données en base contre des injections SQL (autres mesures à mettre en place).

Les données sont ensuite préparées avant d'être exécutées et créés dans la base de données.

La classe « CheckLogin » nous permet de comprendre comment la lecture des données est réalisée.

```
// Check login data validity
1 référence
public bool CheckLogin(string username, string password)
{
    if (!IsConnect()) return false;

    string query = "SELECT * FROM t_user WHERE `username` = @username;"; //Secure request
    cmd = new MySqlCommand(query, Connection); //Send the request to the database
    cmd.Parameters.AddWithValue("@username", username); //Bind the parameters
    dataReader = cmd.ExecuteReader();
    while (dataReader.Read())
    {
        //Retrieve actual user id
        userID = dataReader.GetInt32(0);

        if (dataReader.GetString(1) == username)
        {
            if (dataReader.GetString(2) == password)
            {
                dataReader.Close();
                return true;
            }
        }
    }

    dataReader.Close();
    Disconnect();
    return false;
}
```

Après la validation de la connexion à la base de données, nous avons la requête qui commence par « SELECT.. FROM », ensuite un bind des paramètres avant l'exécution de la requête.

Une fois la requête exécutée, on parcourt chaque données de la variable dataReader qui est de type « MySQLDataReader », en utilisant la fonctionnalité « Read » qui nous permet de lire chaque ligne de résultat. Grâce au « GetString », c'est données sont récupérer selon leur index.

Quand les données sont simplement exécutées sans besoin de les re-utiliser on utilise l'option « ExecuteNonQuery() », sinon « ExecuteReader() » .

Ces options sont propres au type « MySqlCommand ».

Dans notre programme il y a une seule requête pour mettre à jour les données, la voici :

```
// Edition of registered password data
1 référence
public bool EditPasswordData(string newName, string previousName, string username, string password)
{
    if (!IsConnect()) return false;

    string query = "UPDATE `t_website` SET `name` = @newName && `username` = @username `password` = @password " +
        "WHERE `name` = @previousName;";
    cmd = new MySqlCommand(query, Connection);
    cmd.Parameters.AddWithValue("@newName", newName);
    cmd.Parameters.AddWithValue("@previousName", previousName);
    cmd.Parameters.AddWithValue("@username", username);
    cmd.Parameters.AddWithValue("@password", password);
    cmd.Prepare();
    cmd.ExecuteNonQuery();
    return true;
}
```

Pour effacer des données, il n'y a également qu'une seule classe qui s'en occupe, « ErasePasswordData ». Un bin des paramètres est fait, la commande est préparée avant d'être exécutée.

```
// Deletion of registered password data
1 référence
public bool ErasePasswordData(string name)
{
    if (!IsConnect()) return false;

    string query = "DELETE FROM `t_website` WHERE `name` = @name;";
    cmd = new MySqlCommand(query, Connection);
    cmd.Parameters.AddWithValue("@name", name);
    cmd.Prepare();
    cmd.ExecuteNonQuery();
    return true;
}
```

La requête MySQL suit le pattern « DELETE FROM... WHERE.. ». Les paramètres sont bindés, la commande est préparée avant d'être exécutée.

La structure des classes est la même pour 3 des fonctions CRUD, seul la lecture des données nécessite un code légèrement différent.

Gestion des erreurs ?

5.4 Options de sécurité

Pour garantir la sécurité des données de l'utilisateur, le mot de passe enregistré dans la base de données devra être haché et salé.

- Authentification de l'utilisateur (ou le cas échéant « création de compte »)
- Protection contre les injections SQL
- Mots de passe cachés
- Validation du mot de passe s'il
- Absence de pattern pour la génération de mot de passe et pour la validation de mot de passe lors de la création de compte

- Gestion des erreurs avec messages d'erreurs expliquant à l'utilisateur la raison de l'erreur

Génération du mot de passe

La génération de mot de passe est adaptée à la demande de l'utilisateur.

La classe du contrôleur chargée de vérifier ses données prend en paramètres le nombre de caractères qu'il aurait choisi, des variables booléennes pour savoir s'il veut ou non : des chiffres, des lettres majuscules et des caractères spéciaux dans la solution générée. A partir de là, il série de conditions pour savoir les caractères à considérer dans la randomisation.

... position du curseur.

... calcul derrière la position du curseur.

... utilisateur a la possibilité de changer la donnée du champ.

... modification du champ qui fait foi.

```
// Generate a password depending on the user needs
1 référence
public string GeneratePassword(int nbOfCharacters, bool nb, bool capitalLetter, bool specialCharacter)
{
    string characters = "abcdefghijklmnopqrstuvwxyz";

    if (nb)
        characters += "0123456789";
    if (capitalLetter)
        characters += "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    if (specialCharacter)
        characters += "-/#!?";

    var Charsarr = new char[nbOfCharacters];
    var random = new Random();

    for (int i = 0; i < Charsarr.Length; i++)
    {
        Charsarr[i] = characters[random.Next(characters.Length)];
    }

    return new String(Charsarr);
}
```

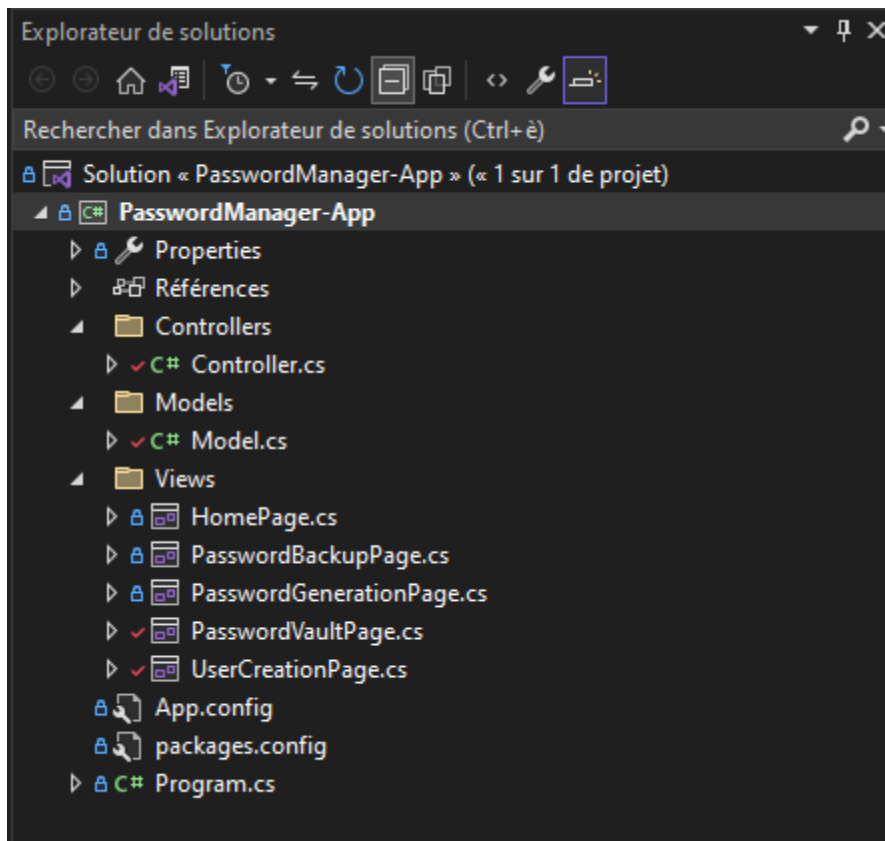
La force du mot de passe est calculée en...

Chiffrement des données

... librairie utilisée.

Explication du modèle MVC

3 dossiers.



Les dossiers « Controllers » et « Models », contiennent tous deux un fichier de type classe. Le « Views » contient 5 fichiers de type form.

Le modèle MVC si l'on regarde au code de la page d'accueil (1^{ère} vue de l'application)

```
namespace PasswordManager_App
{
    7 références
    public partial class HomePage : Form
    {
        // Reference to the controller
        7 références
        public Controller Controller { get; set; }

        1 référence
        public HomePage()
        {
            InitializeComponent();
        }

        1 référence
        private void HomePage_Load(object sender, EventArgs e)
        {
            Controller.ShareAppID(); //Share of app identity
            passwordInsert.UseSystemPasswordChar = true; //Hide password when taping
        }

        1 référence
        private void helpBtn_Click(object sender, EventArgs e)
        {
            Controller.HelpMessage(0);
        }

        1 référence
        private void connexionBtn_Click(object sender, EventArgs e)
        {
            Controller.CheckLogin(userNameInsert.Text, passwordInsert.Text);
            Controller.Redirection("PasswordGenerationPage"); //Redirection to access the app
            Hide();
        }

        1 référence
        private void createAccountRedirection_Click(object sender, EventArgs e)
        {
            Controller.Redirection("UserCreationPage"); //Redirection to create an user
            Hide();
        }
    }
}
```

Quand une action est effectuée sur un élément du form (charge de celui-ci, clic sur le bouton d'aide, sur le bouton de connexion ou pour aller à la page « créer un compte »), le code est bref et redirige vers la classe « Controller ». Cette page contient différentes méthodes.

Ces méthodes contrairement aux méthodes de la page d'accueil comme c'est le cas pour toutes les vues, sont de type privé, donc inaccessibles en dehors de la classe du formulaire.

Les méthodes du contrôleur peuvent soit directement modifier des données (Voir « ShareAppID ») :

```
// Share of app identity through all app pages
1 référence
public void ShareAppID()
{
    PasswordBackupPage.Text = _home.Text;
    PasswordGenerationPage.Text = _home.Text;
    PasswordVaultPage.Text = _home.Text;
    UserCreationPage.Text = _home.Text;

    PasswordBackupPage.Icon = _home.Icon;
    PasswordGenerationPage.Icon = _home.Icon;
    PasswordVaultPage.Icon = _home.Icon;
    UserCreationPage.Icon = _home.Icon;
}
```

Soit faire appel à une méthode du modèle qui se chargera d'exécuter des données.

```
// Check login data
1 référence
public void CheckLogin(string username, string password)
{
    this.username = username;
    if (username != string.Empty && password != string.Empty)
    {
        if (_model.CheckLogin(username, HashPassword(password, true)))
        {
            Redirection("PasswordVaultPage");
        }
        else
        {
            MessageBox.Show("Vos identifiants ne sont pas reconnus. Veuillez les re-vérifier ou créer un nouveau compte.");
        }
    }
    else
    {
        MessageBox.Show("Vos identifiants ne sont pas reconnus. Veuillez les re-vérifier ou créer un nouveau compte.");
    }
}
```

Il gère aussi les erreurs et affiche les messages appropriés selon les erreurs rencontrées.

Et le modèle comme vu précédemment :

```
// Check login data validity
1 référence
public bool CheckLogin(string username, string password)
{
    if (!IsConnect()) return false;

    string query = "SELECT * FROM t_user WHERE `username` = @username;"; //Secure request
    cmd = new MySqlCommand(query, Connection); //Send the request to the database
    cmd.Parameters.AddWithValue("@username", username); //Bind the parameters
    dataReader = cmd.ExecuteReader();
    while (dataReader.Read())
    {
        //Retrieve actual user id
        userID = dataReader.GetInt32(0);

        if (dataReader.GetString(1) == username)
        {
            if (dataReader.GetString(2) == password)
            {
                dataReader.Close();
                return true;
            }
        }
    }

    dataReader.Close();
    Disconnect();
    return false;
}
```

L'initialisation du modèle MVC cependant commence dans la classe « Program » :

```
using System;
using System.Windows.Forms;

namespace PasswordManager_App
{
    0 références
    internal static class Program
    {
        /// <summary>
        /// Main entry point of the app
        /// </summary>
        [STAThread]
        0 références
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            //Initialization of the MCD model
            Model model = new Model();
            HomePage home = new HomePage();
            Controller controller = new Controller(model, home);

            //Initialization of all the page
            PasswordBackupPage passwordBackup = new PasswordBackupPage();
            PasswordGenerationPage passwordGeneration = new PasswordGenerationPage();
            PasswordVaultPage passwordVault = new PasswordVaultPage();
            UserCreationPage userCreation = new UserCreationPage();

            //Controller's initialization of all page depending on it
            home.Controller = controller;
            passwordBackup.Controller = controller;
            passwordGeneration.Controller = controller;
            passwordVault.Controller = controller;
            userCreation.Controller = controller;

            //Initialization of classes in Controller.cs
            controller.PasswordBackupPage = passwordBackup;
            controller.PasswordGenerationPage = passwordGeneration;
            controller.PasswordVaultPage = passwordVault;
            controller.UserCreationPage = userCreation;

            Application.Run(home);
        }
    }
}
```

Déclencheurs suppression

Gestion des versions (versionning) se fait grâce à git, par les commit on peut récupérer et suivre l'évolution du projet. Le message des commit est explicite pour faciliter la compréhension sur l'évolution du travail réalisé.

Le git comprend un fichier README et un fichier .gitignore pour éviter le versionning d'éléments parasites comme des dossiers vides. Des tags existent également pour

mettre en avant les commits importants du projet. Ces tags sont nommés en utilisant le n° de version.

Limites de l'environnement

L'application ayant été créée pour répondre aux besoins de l'entreprise, celle-ci n'est disponible qu'en local et ne pourra pas être utilisée hors du cadre professionnel. Les données enregistrées devront être de type professionnel, aucun compte personnel ne devra être enregistré ici.

Résumé :

Situation de départ – Ce logiciel a été créé pour répondre aux besoins d'une entreprise en matière de gestion de mots de passe. Le but du logiciel est de réduire le risque en matière de sécurité des mots de passe. Et grâce au guide des bonnes pratiques l'utilisateur comprendra mieux les enjeux derrière.

Mise en œuvre – La réalisation de cette application nécessite un environnement...

Résultat – L'application finale permet à l'utilisateur de s'authentifier, de créer un compte, d'ajouter des données de mot de passe, de les afficher, de les gérer et de générer des mots de passe si nécessaire, le tout de manière sécurisé grâce au cryptage de ses données sensibles, à la confidentialité du logiciel dans son ensemble.

Planification

La planification a été segmentée en différentes tâches, chacune servant un objectif à atteindre pour la réalisation du projet dans sa globalité. Le total de 28 séquences réparties sur une durée comprise entre le 08.05.25 et le 02.06.25, comprend.

Journal de travail

Le journal de travail diffère en bien des points de la planification initiale.

Cela vient sans doute d'une envie de bien faire, au point de repartir de quasi zéro alors que des bouts de code avaient déjà été réalisées. Le manque de vérification des différentes pages et de leurs fonctionnalités de façon progressive a également ralenti mon avancée de même que mon environnement de travail et des problèmes personnels.

Bilan

Difficultés à exécuter les tâches de manière routinière, sans trop réfléchir, par peur de simplicité, de

6 Autres

Demandes à ChatGPT

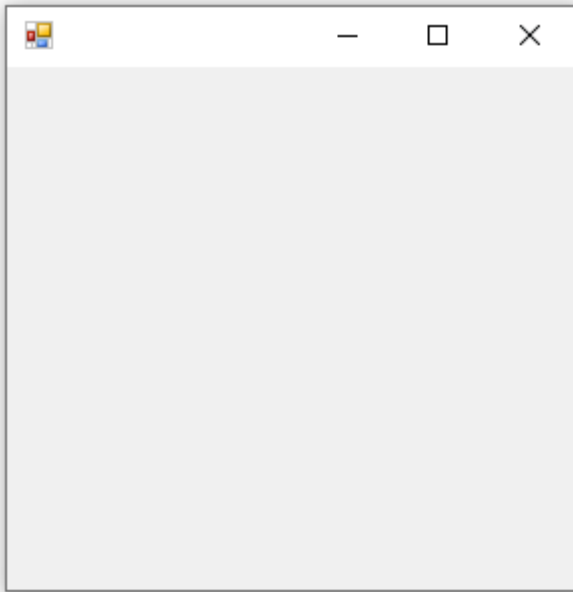
12.05.25 - Données de test

The screenshot displays a web application interface for project management. The top navigation bar includes a search bar, a 'Nouveau' button, and a 'Mettre à niveau' button. The main content area shows a list of tasks under the heading 'À FAIRE'. The tasks are organized into a table with columns for 'Nom', 'Assigné', 'Date d'échéance', and 'Priorité'. The tasks are grouped by a tree view on the left, showing a hierarchy of tasks. The tasks are as follows:

Nom	Assigné	Date d'échéance	Priorité
Page "se connecter" % 1			Urgente
Verification des identifiants			
Page "créer un compte" % 4			Elevée
Verification de la disponibilité du nom de l'utilisateur			
Verification de la concordance des mots de passe			
Exigences du mot de passe			
Accès au reste de l'application			
Page "générer un mot de passe" % 3			Elevée
Contrôle options sélectionnées			
Génération du mot de passe aléatoire			
Contrôle de la force du mot de passe			
Page "sauvegarder un mot de passe" % 1			Normale
Ajout des données dans la base de données (mot de passe haché)			
Page "coffre fort" % 5			Elevée
Récupération des données de la DB			
Affichage des données sous forme de tableau			
Mot de passe masqué			

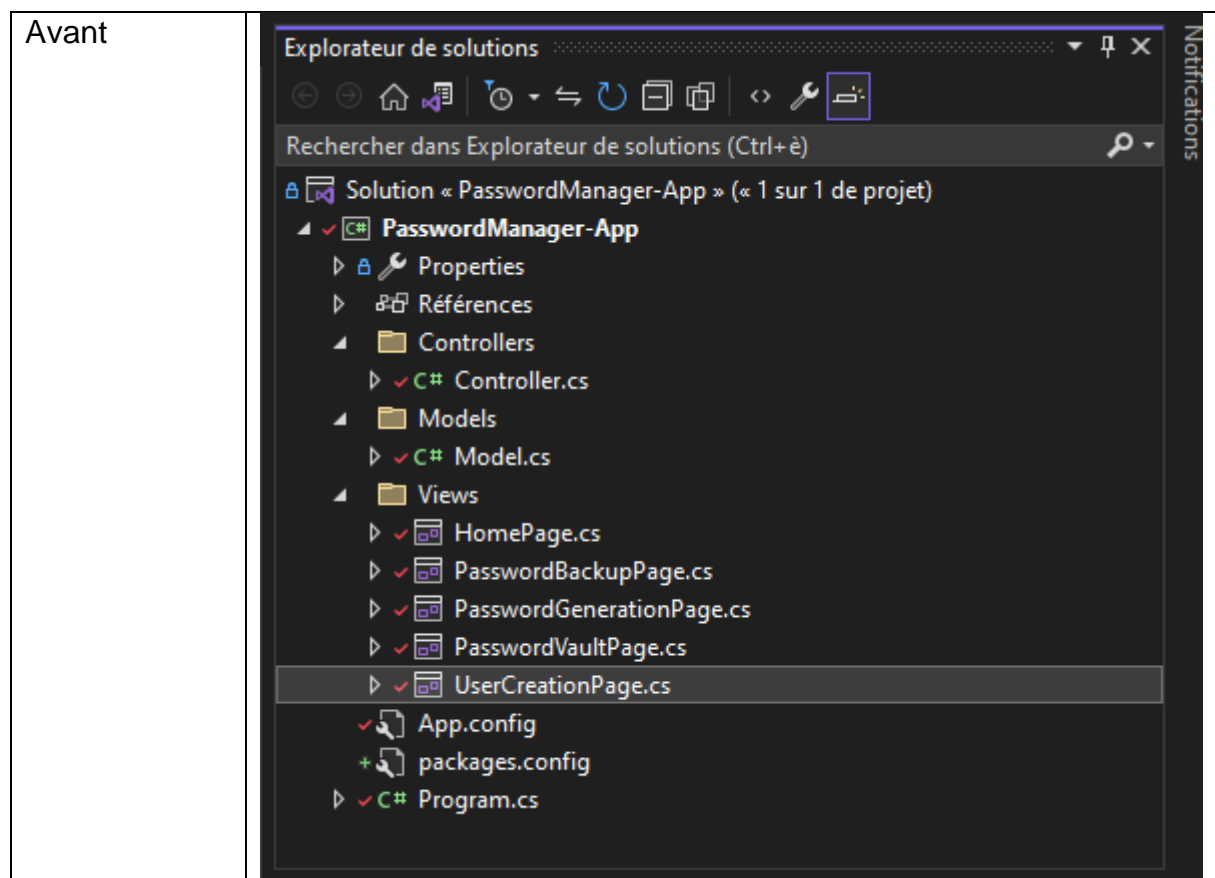
Erreur n° 1

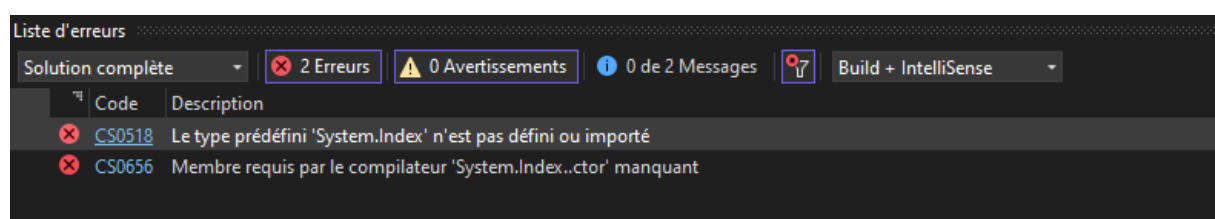
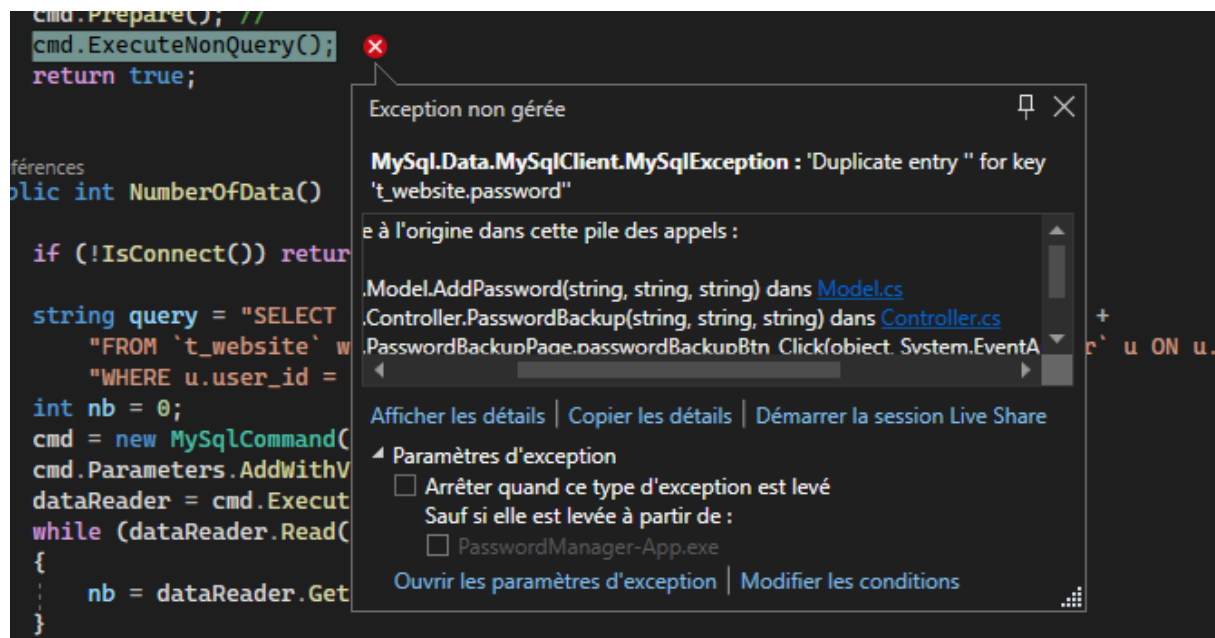
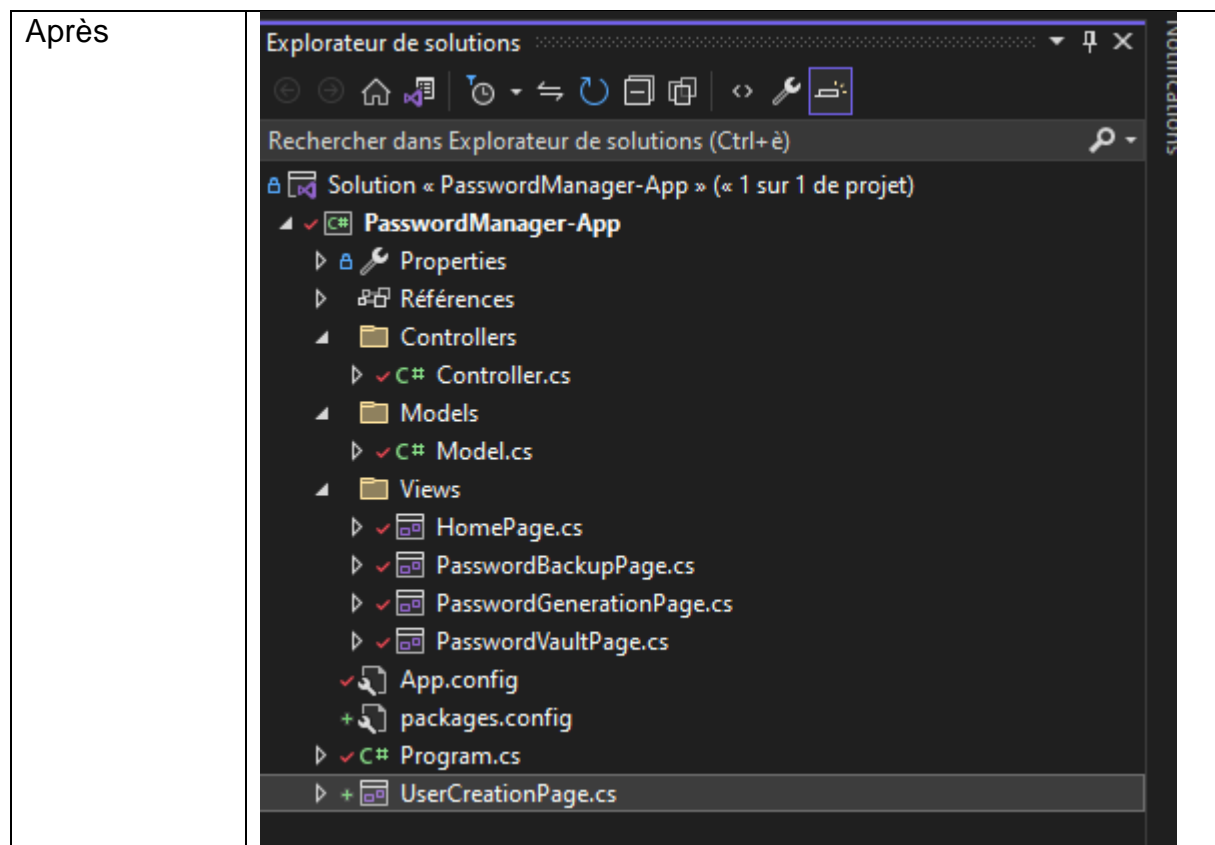
Problème d'affichage du formulaire « UserCreationPage »



Solution n° 1

Créer le formulaire depuis « PasswordManager-App » avant de le déplacer dans le dossier « Views ».





7 Annexes

7.1 Résumé du rapport du TPI / version succincte de la documentation

7.2 Sources – Bibliographie

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)

Vecteezy, <https://www.vecteezy.com/vector-art/21190333-more-vertical-three-dots-settings-filled-icon-in-transparent-background-basic-app-and-web-ui-bold-line-icon-eps10>, 09.05.25
<https://icon-library.com/icon/log-out-icon-16.html>, 09.05.25
<https://icon-library.com/icon/help-icon-png-16.html>, 09.05.25
Vecteezy, <https://www.vecteezy.com/vector-art/15345697-lock-password-password-lock-secure-password-blue-dotted-line-line-icon>, 09.05.25
<https://www.remove.bg/>, 09.05.25
Figma, <https://www.figma.com/>, 09.05.25
<https://blog.cellenza.com/developpement-specifique/les-nouveautes-net-9-et-c-13-avec-des-exemples/>, 09.05.25
<https://dyma.fr/blog/mysql-9-0-decouvrez-les-nouveautes-mais-attention-a-lusage-en-production/>, 09.05.25
<https://dev.to/bytehide/top-10-tips-with-code-examples-c-application-security-47hk>, 09.05.25
Delftstack, <https://www.delftstack.com/howto/csharp/generate-uuid-in-csharp/>, 09.05.25
<https://blog.elmah.io/whats-new-in-net-9-cryptography-improvements/>, 09.05.25
<https://blog.capdata.fr/index.php/nouveautes-mysql-8-0-les-indexes-invisibles/>, 09.05.25
<https://www.angleformation.com/dernieres-fonctionnalites-de-mysql/>, 09.05.25
Freehttps://fr.freepik.com/icone/desinstaller_9498887, 09.05.25
<https://fr.vecteezy.com/art-vectoriel/6082157-coche-icone-vecteur-signe-ou-symbole>, 09.05.25
<https://leprofdegeo.weebly.com/les-cartes-theacutematiques.html>, 09.05.25
Lipsum, <https://fr.lipsum.com/feed/html>, 09.05.25
Looping, <https://www.looping-mcd.fr/>, 12.05.25
<https://chatgpt.com>, 12.05.25
Clickup, <https://app.clickup.com/>, 12.05.25
Online Gantt, <https://www.onlinegantt.com/>, 12.05.25
Convertio, <https://convertio.co/fr/>, 13.05.25
Delftstack, <https://www.delftstack.com/fr/howto/csharp/generate-random-alphanumeric-strings-in-csharp/>, 13.05.25
<https://blog.elmah.io/effective-error-logging-in-windows-forms-applications-with-c/>, 13.05.25

Pngall, <https://th.bing.com/th/id/R.b8e7dfa95ff6ce6e573c6535477d1b29?rik=rtvS%2bmEKj381pg&riu=http%3a%2f%2fwww.pngall.com%2fwp-content%2fuploads%2f5%2fVertical-Line-PNG-Clipart.png&ehk=9ce1mvRIMxAQABgDvFZr6gVo0IAg8MP96dztFw9TQO4%3d&risl=&pid=ImgRaw&r=0>, 14.05.25

C-Sharpcorner, <https://www.c-sharpcorner.com/UploadFile/deveshomar/ways-to-bind-datagridview-in-window-forms-C-Sharp/>, 14.05.25

Github, <https://acceis.github.io/EntroCalc/>, 15.05.25

Archive

<https://web.archive.org/web/20230109120945/https://www.ssi.gouv.fr/administrati-on/precautions-elementaires/calculer-la-force-dun-mot-de-passe/>, 15.05.25

Webdevtutor, <https://www.webdevtutor.net/blog/c-sharp-populate-datagridview>, 15.05.25

Dev FAQ, <https://www.devfaq.fr/question/comment-fusionner-deux-requ-ecirc-tes-select-dans-mysql>, 15.05.25

Learn SQL, <https://learnsql.fr/blog/comment-ecrire-plusieurs-jointures-dans-une-requete-sql/>

ChatGPT, <https://chatgpt.com/>, 15.05.25

C-Sharpcorner, <https://www.csharpcorner.com/UploadFile/mahesh/working-with-listview-in-C-Sharp/>, 16.05.25

Stackoverflow, <https://stackoverflow.com/questions/17381725/c-sharp-listview-item-image>, 16.05.25

Webdevtutor, <https://www.webdevtutor.net/blog/c-sharp-systemdrawingimage>, 16.05.25

TPI VD, <https://www.tpivd.ch/wp/>, 20.05.25

7.3 Journal de travail

Date	Durée	Activité	Remarques

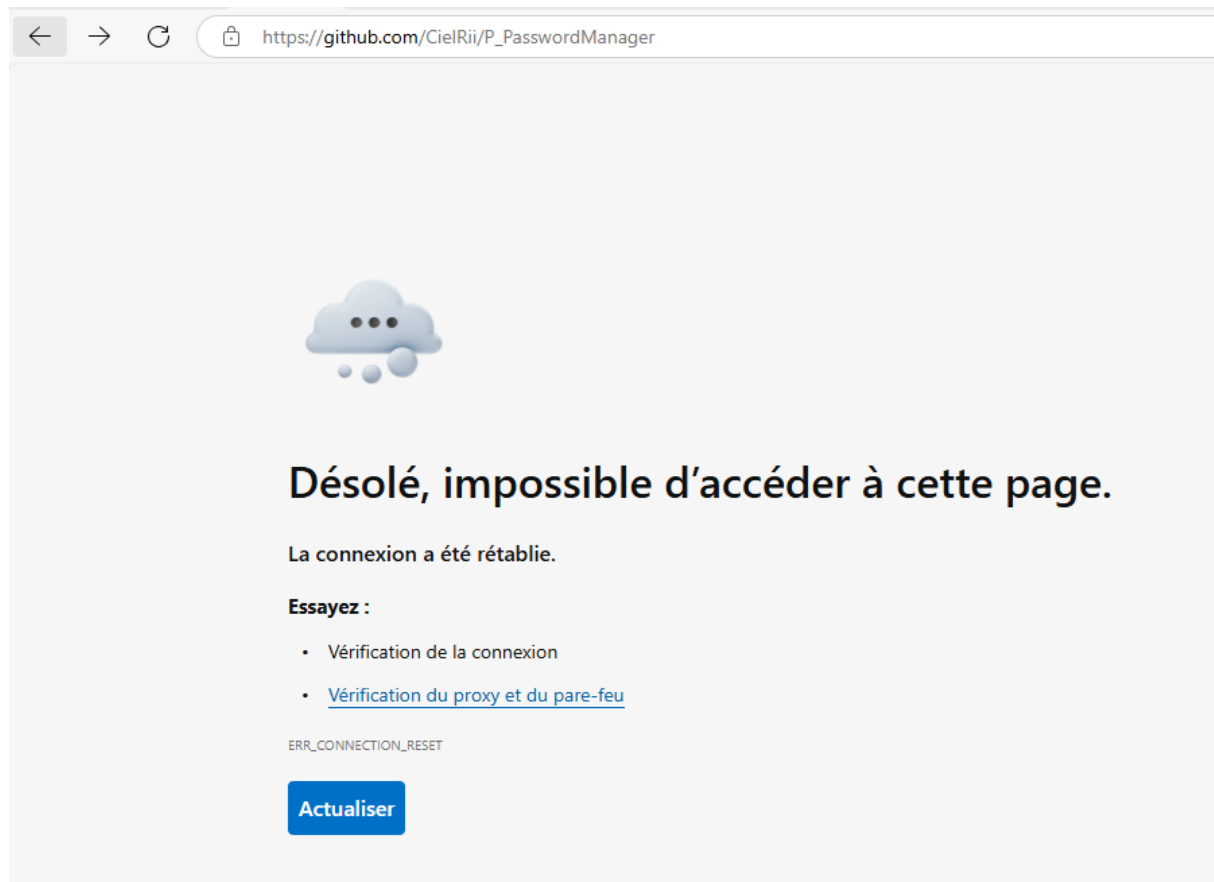
7.4 Manuel d'Installation


7.5 Manuel d'Utilisation


7.6 Archives du projet

Media, ... dans une fourre en plastique

Erreurs :



← → ↻  https://github.com/CielRii/P_PasswordManager



Désolé, impossible d'accéder à cette page.

La connexion a été rétablie.

Essayez :

- Vérification de la connexion
- [Vérification du proxy et du pare-feu](#)

ERR_CONNECTION_RESET

Actualiser

```
sardongmo@INF-A13-M202 MINGW64 /c/Users/sardongmo/Desktop/P_PasswordManager (master)
$ git pull origin master --allow-unrelated-histories
fatal: unable to access 'https://github.com/CielRii/P_PasswordManager.git/': OpenSSL SSL_connect: Connection was reset in connection to github.com:443
```

Sous forme de tableau :

The screenshot shows a window titled "Form4" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a table with the following data:

	Site	Nom d'utilisateur	Mot de passe
▶	Gmail	alice@gmail.com	gmailpass1
	LinkedIn	alice_ln	lnpass1

To the right of the table, there are two icons: a vertical ellipsis (three dots) and a question mark inside a circle. Below the table, a large grey rectangular area occupies most of the window. In the center of this grey area, the text "Mon coffre" is displayed in a white box. At the bottom of the window, a partial label "Opérations sur les résultats de la requête" is visible.