|  |
| --- |
| Password Manager |

Table des matières

[1 Analyse préliminaire 3](#_Toc499021832)

[1.1 Introduction 3](#_Toc499021833)

[1.2 Objectifs 3](#_Toc499021834)

[1.3 Planification initiale 4](#_Toc499021835)

[2 Analyse / Conception 4](#_Toc499021836)

[2.1 Concept 4](#_Toc499021837)

[2.2 Stratégie de test 4](#_Toc499021838)

[2.3 Risques techniques 4](#_Toc499021839)

[2.4 Planification 5](#_Toc499021840)

[2.5 Dossier de conception 5](#_Toc499021841)

[3 Réalisation 5](#_Toc499021842)

[3.1 Dossier de réalisation 5](#_Toc499021843)

[3.2 Description des tests effectués 6](#_Toc499021844)

[3.3 Erreurs restantes 6](#_Toc499021845)

[3.4 Liste des documents fournis 6](#_Toc499021846)

[4 Conclusions 6](#_Toc499021847)

[5 Annexes 7](#_Toc499021848)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 10](#_Toc499021849)

[5.2 Sources – Bibliographie 10](#_Toc499021850)

[5.3 Journal de travail 11](#_Toc499021851)

[5.4 Manuel d'Installation 11](#_Toc499021852)

[5.5 Manuel d'Utilisation 11](#_Toc499021853)

[5.6 Archives du projet 11](#_Toc499021854)

# Spécifications

## Titre

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

## Description

Application desktop permettant de gérer les mots de passe de manière sécurisée et centralisée pour une petite PME.

Bonnes pratiques

Explication du contexte – mauvaise gestion des mots de passe (problème sécurité, confidentialité, sensibilité des données, confiance en l’entreprise)

Explication de la situation

Raisons du projet (moivations)

## Matériel et logiciels à disposition

* 1 ordinateur type de la section informatique (Windows 10, avec suite Office)
* Microsoft Visual Studio 2022

## Prérequis

La réalisation de ce projet nécessite la connaissance et la maîtrise des modules ICT ci-dessous, enseignés à l’ETML :

* 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 servent à attester des différentes compétences acquises tout au long de la formation suivie entre juillet 2023 et mars 2025.

## Cahier des charges

### 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)

Combien

Quoi

Qui

Comment

Où

Quand

Pourquoi

### 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.)

Cette application étant conçue pour les employés d’une petite PME,

### Fonctionnalités requises (du point de vue de l’utilisateur)

L’utilisateur devra pouvoir :

* S’authentifier
* Créer un compte
* Accéder à ses données enregistrées
* Sauvegarder de nouvelles données de mot de passe
* Générer un mot de passe aléatoire
* Obtenir de l’aide s’il a besoin d’être guidé

### Contraintes

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

Git régulier pour sauvegarder le travail en cours, avoir du versionning.

Ce projet étant fondamentalement basé sur la sécurité. Les données devront être sauvegardées de manière sécure, 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.

### Travail à réaliser par l'apprenti

En plus d’une application permettant de gérer les mots de passe pour les employés d’une petite PME, le travail rendu comprendra un document décrivant la politique de mot de passe choisi pour l’entreprise et un mode d’emploi pour faciliter l’utilisation de l’application par l’utilisateur.

### Si le temps le permet …

Lors de la validation du cahier des charges, ces éléments ont été proposés et finalement validés :

* La mise en place 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 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 sauvegardées.

Lors de la conception des maquettes, il a été jugé plus judicieux d’avoir un bouton menu qui faciliterait la navigation entre les pages de l’application et d’ajouter l’option de déconnection dans ce menu.

Si le temps le permet, il serait utile d’ajouter des flèches aux bas de la page intitulé « Mon coffre ». Cela permettrait de gérer un plus grand volume d’identifiants, si l’entreprise venait à densifier ses activités. Et de permettre à l’utilisateur lors de son authentification et de la création de son compte de gérer la visibilité de son mot de passe.

### Méthodes de validation des solutions

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

Les tests suivants ont été définis :

Pour l’authentification :

Pour la création de compte :

Pour la sauvegarde de mot de passe :

## 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

## 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

# Analyse préliminaire

## 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.

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 personne 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.

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.

L’application devra permettre :

De centraliser les identifiants et mots de passe de l’entreprise dans une base de données sécurisée.

De générer automatiquement des mots de passe robustes afin de décourager les mauvaises pratiques (comme l’utilisation du même mot de passe partout).

D’authentifier l’utilisateur

## 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.*

## Planification initiale

Le projet a été segmentée en différentes tâches, chaque tâche permettant d’atteindre la réalisation complète du projet. Les séquences de travail se comptent au nombre de 28, réparties sur une durée comprise entre le 08.05.25 et le 02.06.25.

IMAGE : Voir annexe

Les tâches s’enchaînent les unes à la suite des autres. Il y a une dépendance entre certaines phases de la planification en raison de l’utilisation de la méthode de gestion de projet «Waterfall». Pour commencer la réalisation, il faut avoir terminé la conception et avant ça l’analyse. Les tests unitaires et la documentation ont été laissés pour la fin, pour éviter de revenir sur du travail déjà réalisé. Toutes ces tâches ont pour but de remplir l’ensemble des objectifs du projet.

IMAGE : Planification sous forme de diagramme de GANTT

Chaque tâche a une durée définie selon sa complexité et l’effort qu’il faudra mettre pour en venir au bout. Ainsi les premières étapes du projet : analyse et conception prennent moins de temps que la réalisation ou la documentation. Dans cette planification, beaucoup de temps a été laissé pour revoir le travail réalisé et l’amélioré si nécessaire. Le projet comprenant plusieurs points à documenter, la documentation est le point qui prend le plus de temps après l’amélioration de la documentation.

IMAGE : Planification sous forme de camembert

# Analyse / Conception

## 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…*
* *…*

### Maquettes

Les maquettes de l’application ont été réalisées depuis le logiciel « Figma ».

Pour modéliser celle-ci, nous en avons faits six. 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.

### 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 booléan, 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 pas 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 incrementés.

### 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

### 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 :

## Méthode de gestion de projet

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

Celle-ci comprendre les étapes ci-dessous :

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

Celle-ci a été choisi car elle est adaptée à la demande de ce projet, c'est-à-dire, réalisé 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 réfléchis

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.

## 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.*

## Risques techniques

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.

## 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.*

## 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 !***

# Réalisation

## 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: librairies 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.*

* **Etapes 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*

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

## 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*

## Erreurs restantes

*S'il reste encore des erreurs:*

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

## 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…*

# Conclusions

*Développez en tous cas les points suivants:*

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

## 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.

## 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 différence entre les planifications initiales et détaillées avec le journal de travail.

## 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 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 réalisé, 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 :

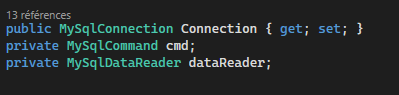
* Longueur : 15 caractères minimum
* Pour le rendre plus robuste, plus complexe : car

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 :



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



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

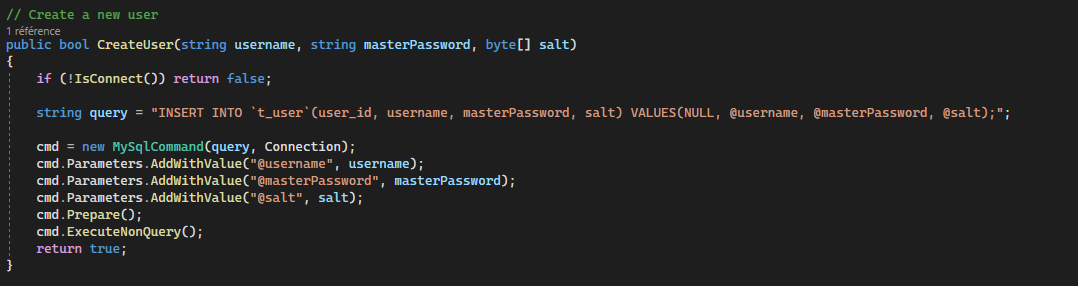
Une autre classe est instanciée pour se déconnecter :



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 :



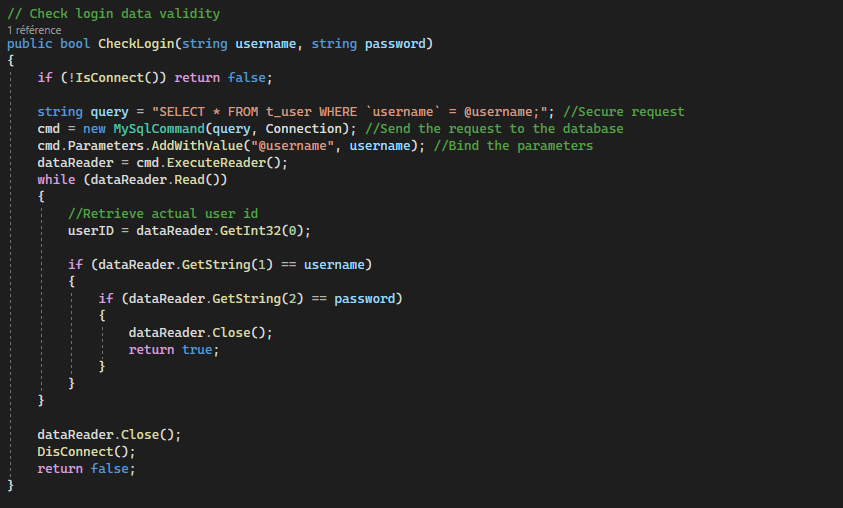
Si la connexion a la base de données est validée (retourne « true »), la requête est exécutée.

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.



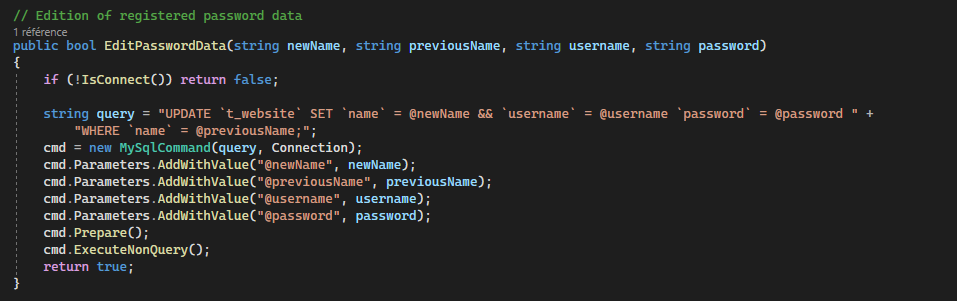
Après la validation de la connexion à la base de données, nous avons la requête qui commence par « SELECT.. FROM », ensuite un bind des paramètres avant l’exécution de la requête.

Une fois la requête exécutée, on parcours 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.

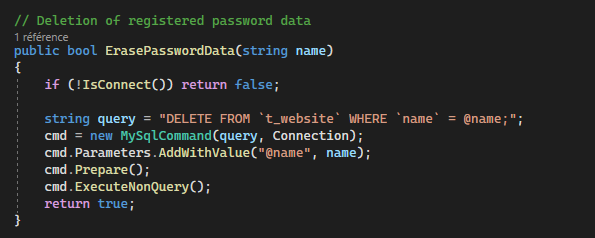
Quand les données sont simplement exécutées sans besoin de les re-utiliser on utilise l’option « ExecuteNonQuery() », sinon « ExecuteReader() » .

Ces options sont propres au type « MySQLCommand ».

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



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.



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

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

Gestion des erreurs ?

## 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.

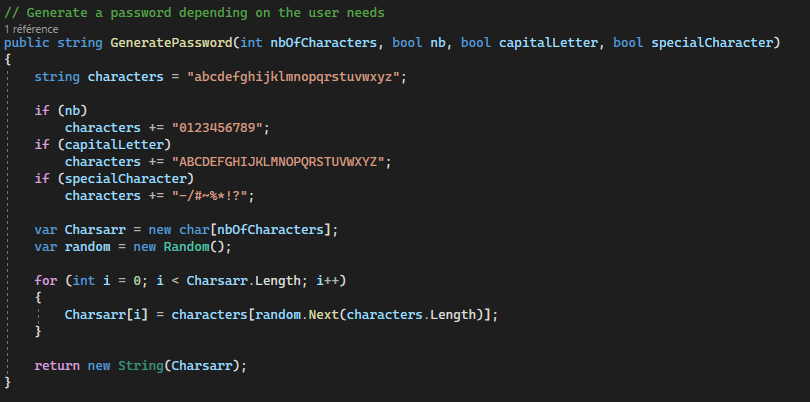
La classe du contrôleur chargée de vérifier ses données prend en paramètres le nombre de caractères qu’il aurait choisi, des variables booléennes pour savoir s’il veut ou non : des chiffres, des lettres majuscules et des caractères spéciaux dans la solution générée. A partir de là, il série de conditions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



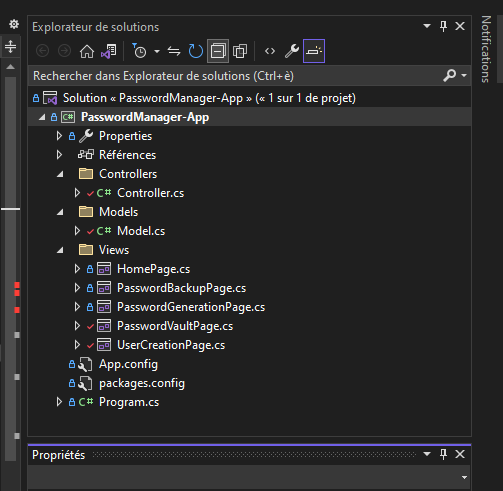
La force du mot de passe est calculée en…

Chiffrement des données

… librairie utilisée.

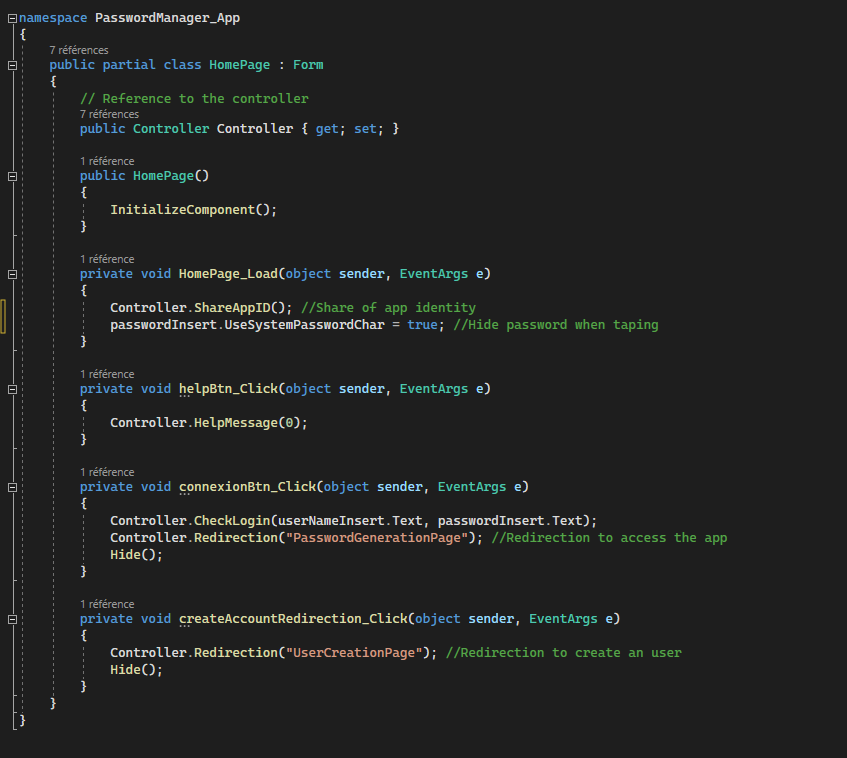
Explication du modèle MVC

3 dossiers.



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

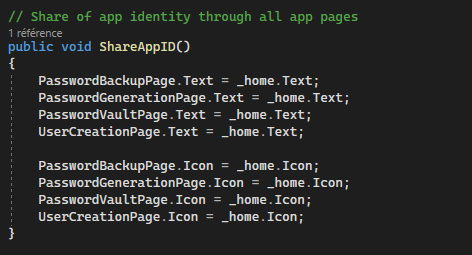
Le modèle MVC si l’on regarde au code de la page d’accueil (1ère vue de l’application)



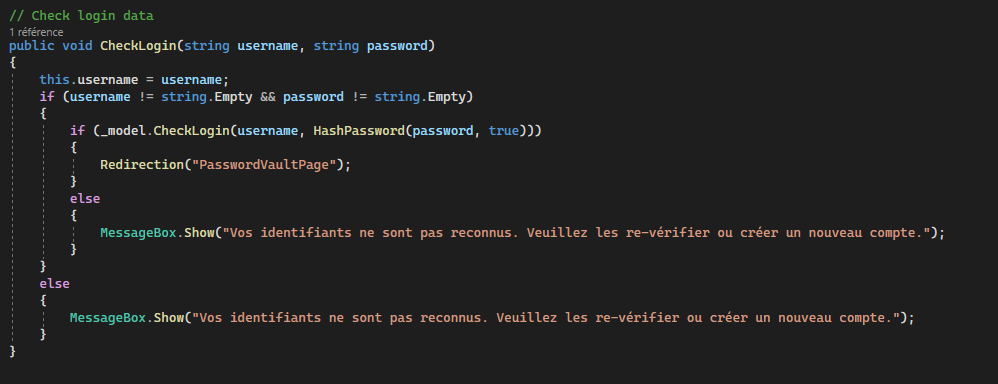
Quand une action est effectuée sur un élément du form (charge de celui-ci, clic sur le bouton d’aide, sur le bouton de connexion ou pour aller à la page « créer un compte »), le code est bref et redirige vers la classe « Controller ». Cette page contient différentes méthodes.

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 ») :

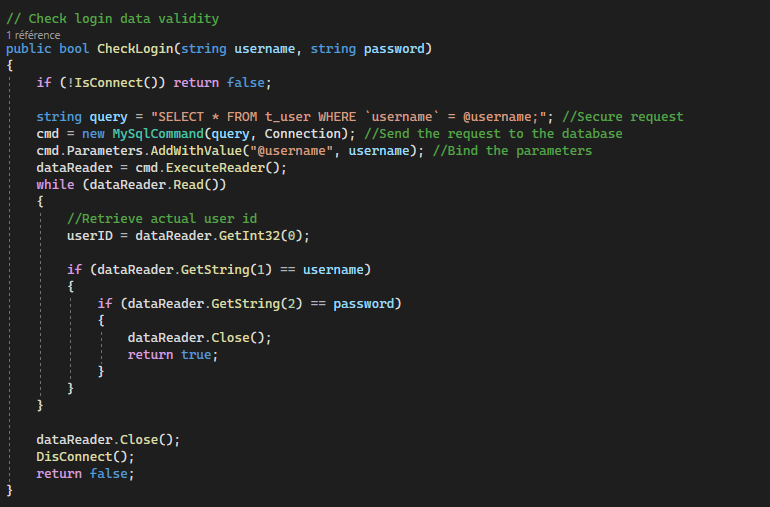


Soit faire appel à une méthode du modèle qui se chargera d’exécuter des données.

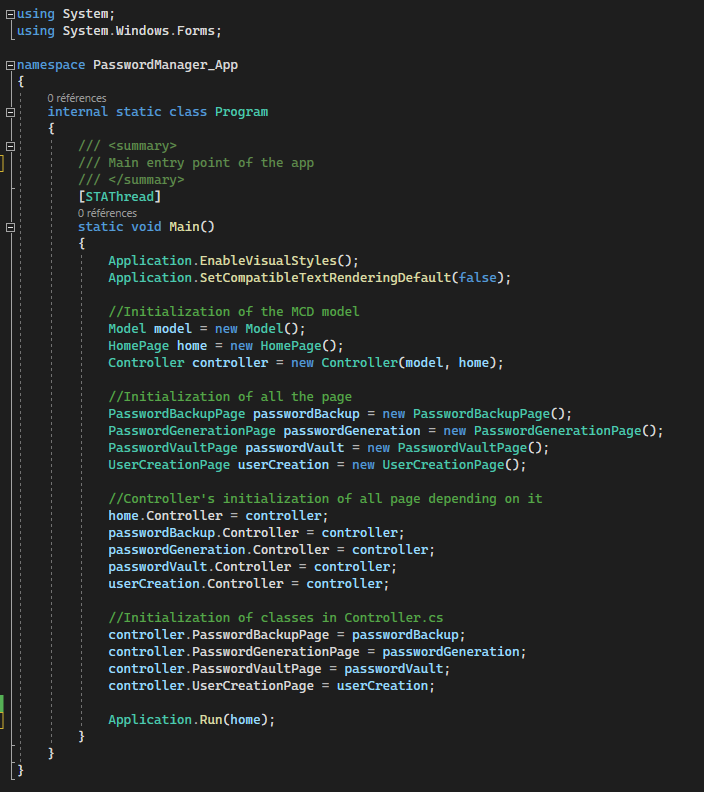


Il gère aussi les erreurs et affiche les messages appropriés selon les erreurs rencontrées.

Et le modèle comme vu précédemment :



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



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.

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.

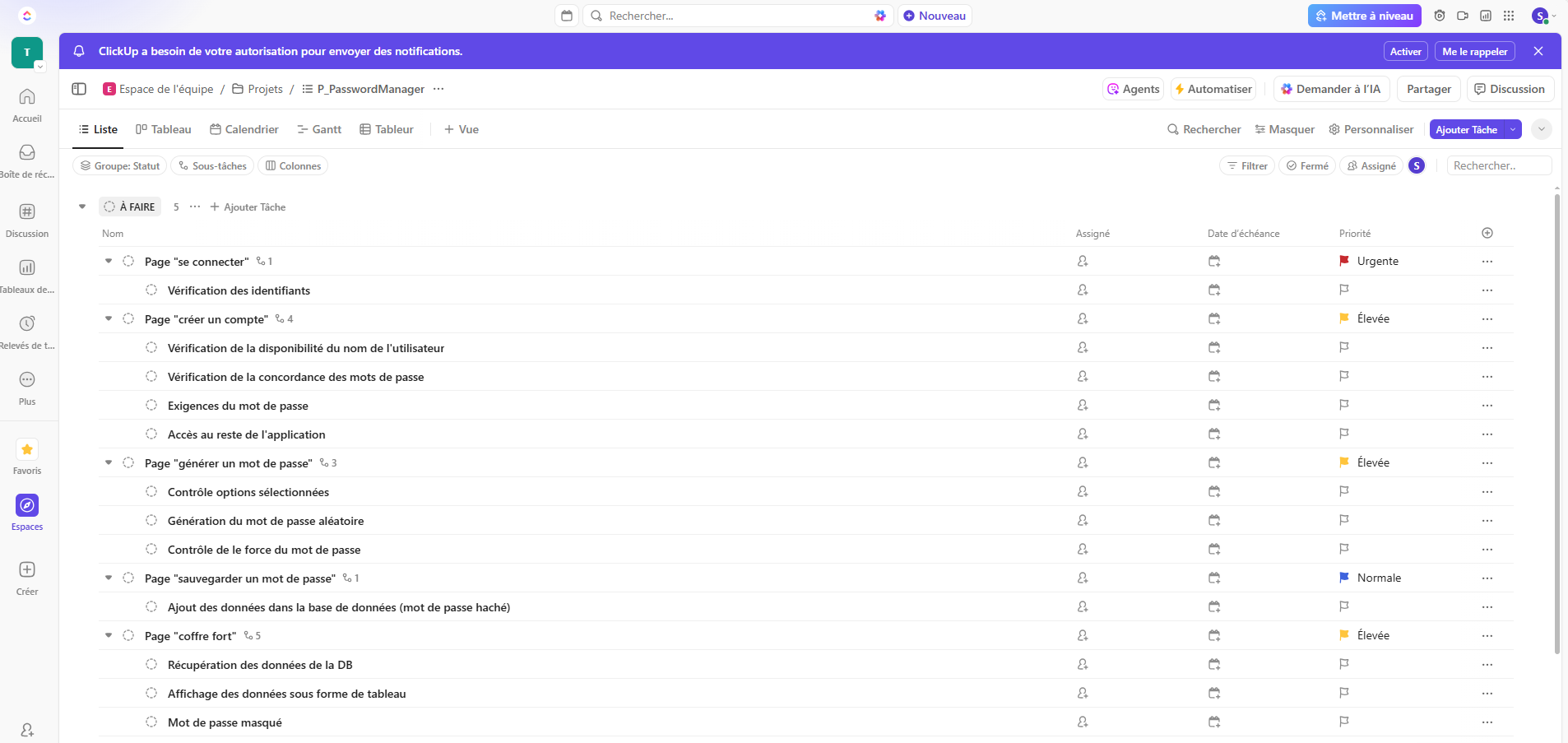
Bilan

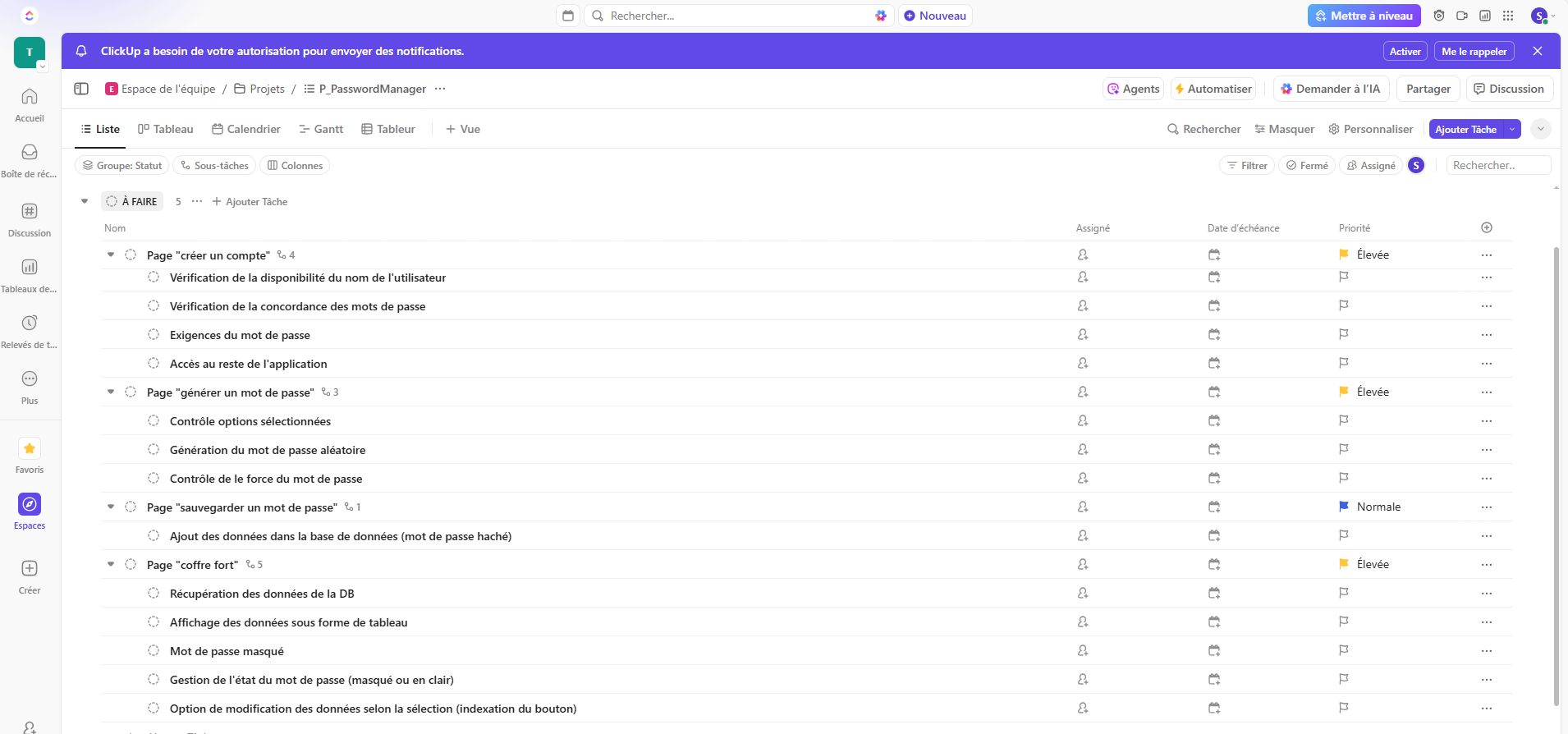
Difficultés à exécuter les tâches de manière routinière, sans trop réfléchir, par peur de simplicité, de

# Autres

Demandes à ChatGPT

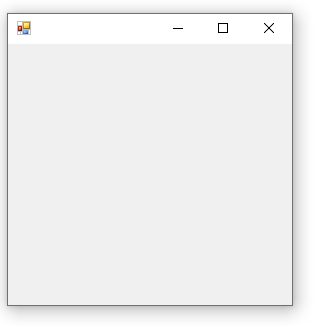
12.05.25 - Données de test





Erreur n° 1

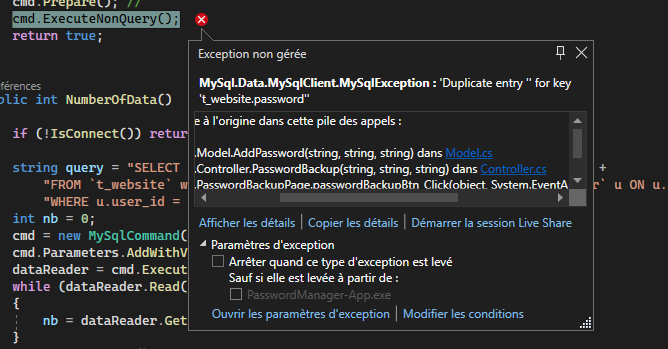
Problème d’affichage du formulaire « UserCreationPage »

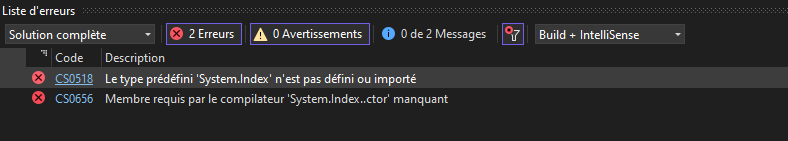


Solution n° 1

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

|  |  |
| --- | --- |
| Avant |  |
| Après |  |





# Annexes

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

## 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*](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*](https://icon-library.com/icon/log-out-icon-16.html)*, 09.05.25*

[*https://icon-library.com/icon/help-icon-png-16.html*](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*](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/*](https://www.remove.bg/)*, 09.05.25*

*Figma,* [*https://www.figma.com/*](https://www.figma.com/)*, 09.05.25*

[*https://blog.cellenza.com/developpement-specifique/les-nouveautes-net-9-et-c-13-avec-des-exemples/*](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/*](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*](https://dev.to/bytehide/top-10-tips-with-code-examples-c-application-security-47hk)*, 09.05.25*

*Delfstack,* [*https://www.delftstack.com/howto/csharp/generate-uuid-in-csharp/*](https://www.delftstack.com/howto/csharp/generate-uuid-in-csharp/)*, 09.05.25*

[*https://blog.elmah.io/whats-new-in-net-9-cryptography-improvements/*](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/*](https://blog.capdata.fr/index.php/nouveautes-mysql-8-0-les-indexes-invisibles/)*, 09.05.25*

[*https://www.angleformation.com/dernieres-fonctionnalites-de-mysql/*](https://www.angleformation.com/dernieres-fonctionnalites-de-mysql/)*, 09.05.25*

*Free*[*https://fr.freepik.com/icone/desinstaller\_9498887*](https://fr.freepik.com/icone/desinstaller_9498887)*, 09.05.25*

[*https://fr.vecteezy.com/art-vectoriel/6082157-coche-icone-vecteur-signe-ou-symbole*](https://fr.vecteezy.com/art-vectoriel/6082157-coche-icone-vecteur-signe-ou-symbole)*, 09.05.25*

[*https://leprofdegeo.weebly.com/les-cartes-theacutematiques.html*](https://leprofdegeo.weebly.com/les-cartes-theacutematiques.html)*, 09.05.25*

*Lipsum,* [*https://fr.lipsum.com/feed/html*](https://fr.lipsum.com/feed/html)*, 09.05.25*

*Looping,* [*https://www.looping-mcd.fr/*](https://www.looping-mcd.fr/)*, 12.05.25*

[*https://chatgpt.com*](https://chatgpt.com/)*, 12.05.25*

*Clickup,* [*https://app.clickup.com/*](https://app.clickup.com/)*, 12.05.25*

*Online Gantt,* [*https://www.onlinegantt.com/*](https://www.onlinegantt.com/)*, 12.05.25*

*Convertio,* [*https://convertio.co/fr/*](https://convertio.co/fr/)*, 13.05.25*

*Delftstack,* [*https://www.delftstack.com/fr/howto/csharp/generate-random-alphanumeric-strings-in-csharp/*](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/*](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=9ce1mvRlMxAQABgDvFZr6gVo0IAg8MP96dztFw9TQO4%3d&risl=&pid=ImgRaw&r=0*](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=9ce1mvRlMxAQABgDvFZr6gVo0IAg8MP96dztFw9TQO4%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/*](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/*](https://acceis.github.io/EntroCalc/)*, 15.05.25*

*Archive* [*https://web.archive.org/web/20230109120945/https://www.ssi.gouv.fr/administration/precautions-elementaires/calculer-la-force-dun-mot-de-passe/*](https://web.archive.org/web/20230109120945/https://www.ssi.gouv.fr/administration/precautions-elementaires/calculer-la-force-dun-mot-de-passe/)*, 15.05.25*

*Webdevtutor,* [*https://www.webdevtutor.net/blog/c-sharp-populate-datagridview*](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*](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/*](https://learnsql.fr/blog/comment-ecrire-plusieurs-jointures-dans-une-requete-sql/)

*ChatGPT,* [*https://chatgpt.com/*](https://chatgpt.com/)*, 15.05.25*

*C-Sharpcorner,* [*https://www.csharpcorner.com/UploadFile/mahesh/working-with-listview-in-C-Sharp/*](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*](https://stackoverflow.com/questions/17381725/c-sharp-listview-item-image)*, 16.05.25*

*Webdevtutor,* [*https://www.webdevtutor.net/blog/c-sharp-systemdrawingimage*](https://www.webdevtutor.net/blog/c-sharp-systemdrawingimage)*, 16.05.25*

*TPI VD,* [*https://www.tpivd.ch/wp/*](https://www.tpivd.ch/wp/)*, 20.05.25*

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## Manuel d'Installation

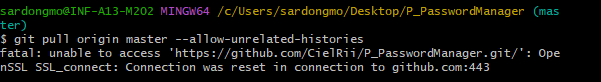
## Manuel d'Utilisation

## Archives du projet

*Media, … dans une fourre en plastique*

Erreurs :





Sous forme de tableau :

