30/05/2023

Candidat : BARRAS Tristan

Chef de projet : ANDOLFATTO Frédérique

Application C# de gestion d’inventaire informatique

Table des matières

[Analyse préliminaire 3](#_Toc136355238)

[Introduction 3](#_Toc136355239)

[Objectifs 3](#_Toc136355240)

[Méthode de gestion de projet 3](#_Toc136355241)

[Planification initiale 4](#_Toc136355242)

[Analyse / Conception 5](#_Toc136355243)

[Concept 5](#_Toc136355244)

[Stratégie de test 11](#_Toc136355245)

[Planification 12](#_Toc136355246)

[Dossier de conception 13](#_Toc136355247)

[Réalisation 14](#_Toc136355248)

[Dossier de réalisation 14](#_Toc136355249)

[Classes 15](#_Toc136355250)

[Description des tests effectués 17](#_Toc136355251)

[Tests unitaire/integration 17](#_Toc136355252)

[Tests système 20](#_Toc136355253)

[Erreurs restantes 20](#_Toc136355254)

[Liste des documents fournis 20](#_Toc136355255)

[Conclusions 21](#_Toc136355256)

[Objectifs atteints / non-atteints 21](#_Toc136355257)

[Points positifs / négatifs 21](#_Toc136355258)

[Difficultés particulières 21](#_Toc136355259)

[Suites possibles pour le projet (évolutions & améliorations) 21](#_Toc136355260)

[Annexes 22](#_Toc136355261)

[Glossaire 22](#_Toc136355262)

[Résumé du rapport du TPI / version succincte de la documentation 23](#_Toc136355263)

[Situation de départ 23](#_Toc136355264)

[Mise en œuvre 23](#_Toc136355265)

[Résultat 23](#_Toc136355266)

[Sources – Bibliographie 24](#_Toc136355267)

[Internet 24](#_Toc136355268)

[Journal de travail 25](#_Toc136355269)

[Manuel d'Installation 27](#_Toc136355270)

[Manuel d'Utilisation 28](#_Toc136355271)

[Partie admin 28](#_Toc136355272)

[Partie User 32](#_Toc136355273)

[Table des illustration 35](#_Toc136355274)

[Archives du projet 37](#_Toc136355275)

[MCD 37](#_Toc136355276)

[Cahier des Charges 38](#_Toc136355277)

# Analyse préliminaire

## Introduction

Ce projet est réalisé dans le cadre d’un TPI, il y a 90 heures de travail. Le projet est une application en C# de gestion d’inventaire de matériel informatique de l’école. Elle permettra de parcourir, ajouter, modifier du matériel informatique présent dans une base de données. Pour ce projet, j’ai des compétences en développement sur du C# ainsi que des compétences en modélisation et gestion de bases de données. J’ai fait un projet similaire le pré-TPI qui a duré 72 heures, ce qui me permet d’avoir un peu d’expérience pour la gestion d’un projet comme le TPI.

## Objectifs

Les objectifs de ce projet sont en premier de créer une partie administrateur disponible avec une authentification et qui pourra à l’aide d’une interface graphique insérer dans une base de données des informations sur du matériel informatique. Le deuxième objectif est de pouvoir créer/modifier des listes pour les marques du matériel, les modules ainsi que l’emplacement où ce qu’il se trouve. Le troisième objectif est le fait de pouvoir modifier les données du matériel ainsi que l’archivé. Le quatrième objectif est de faire une interface graphique pour que les utilisateurs puissent consulter le matériel informatique grâce à des critères et il peut voir toutes les informations sur le produit en cliquant dessus.

## Méthode de gestion de projet

La méthode de gestion de projet qui est utilisé est la méthode dite en cascade. Cette méthode a été choisie par confort en effet, ma méthode de travail correspond mieux à cette méthode que par exemple la méthode agile. Je me retrouve plus dans cette méthode qui me permet de bien suivre le projet étape par étape. La méthode se divise en 5 phases qui doivent se suivre successivement. La première phase l’analyse est dans ce cas l’analyse du cahier des charges normalement cette étape comprend la conception du cahier des charges, mais dans cette situation le cahier des charges a été reçu le premier jour. La seconde étape la conception comprend dans ce cas l’analyse initiale, la modélisation de la base de données, la maquette de l’application ainsi que le point analyse/conception dans le rapport du projet. La troisième étape est l’implémentation comprend la partie programmation. La quatrième partie, l’étape teste comprend tous les tests prévus dans la partie conception puis l’exploitation est la première version du produit est finie.

Méthode de gestion, en cascade1

## Planification initiale

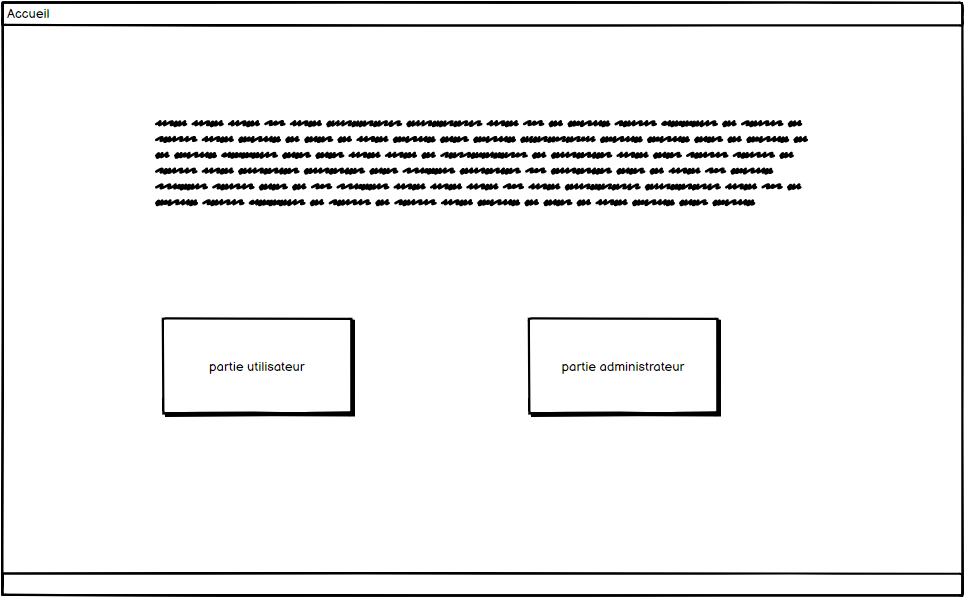
Voici un diagramme de Gantt pour représenter ma planification initiale.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **02.05.2023 - 30.05.2023** | | | | |
| 02.05.23 | 08.05.23 | 15.05.23 | 22.05.23 | 30.05.23 |
| Analyse |  |  |  |  |  |
| IHM |  |  |  |  |  |
| Code métier |  |  |  |  |  |
| Tests |  |  |  |  |  |
| Documentation final |  |  |  |  |  |

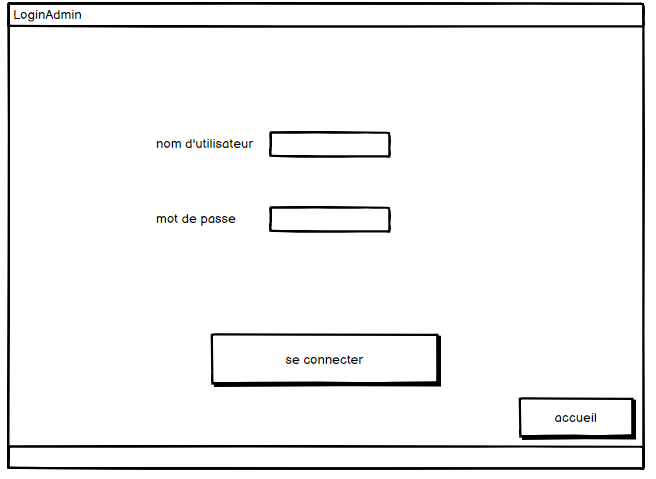
# Analyse / Conception

## Concept

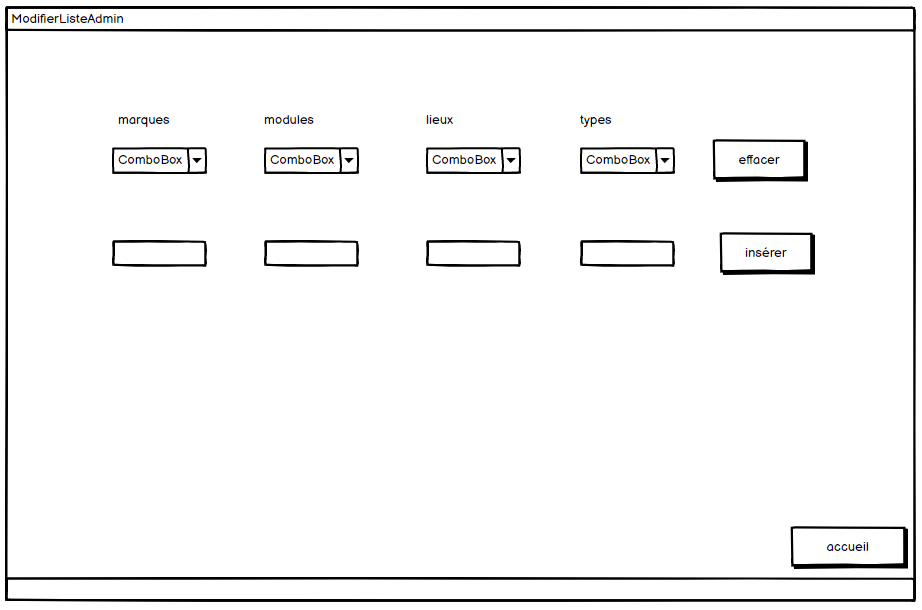
Voici la maquette effectué pour voir concrètement l’apparence de l’application :



Accueil 2

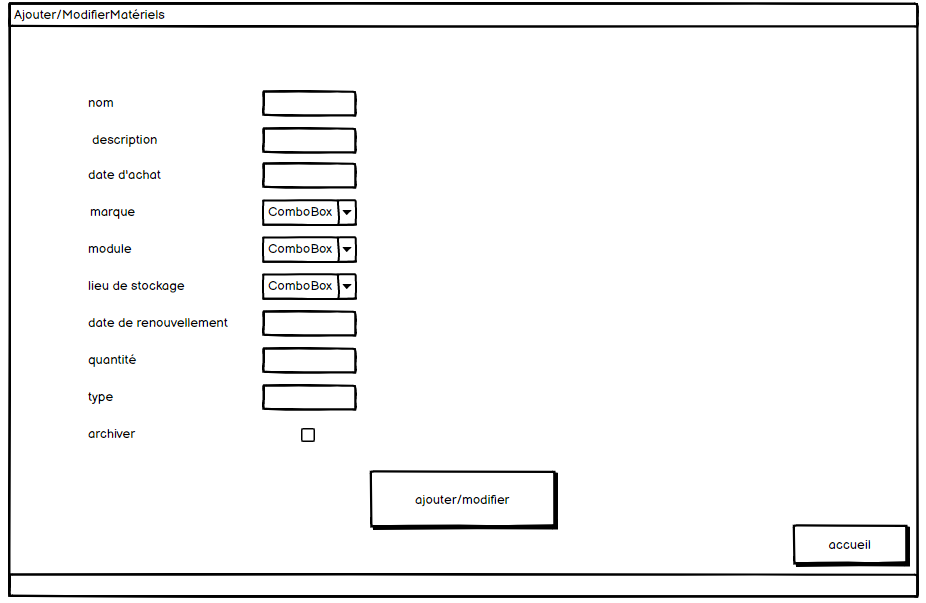


login 3

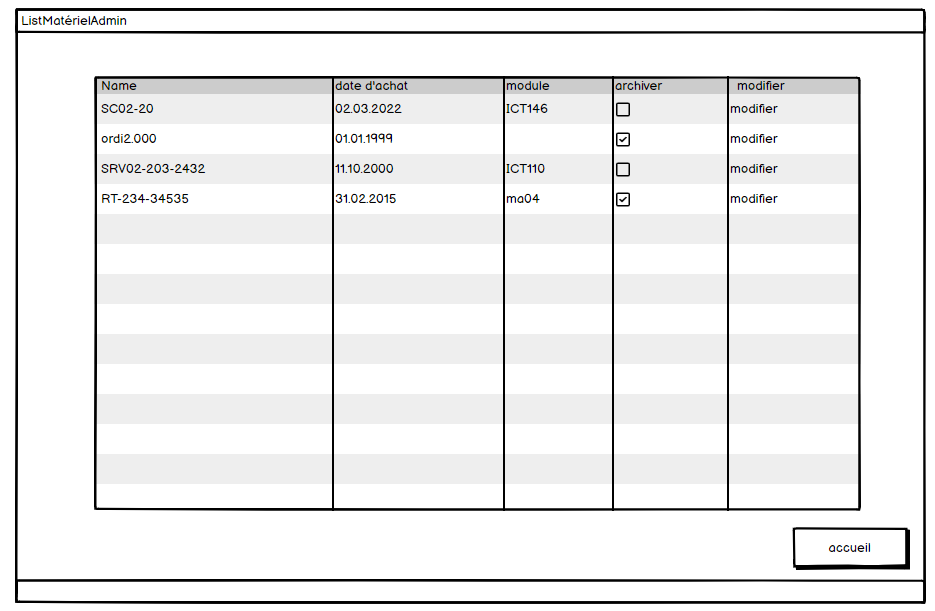


Ajouter/modifier 4

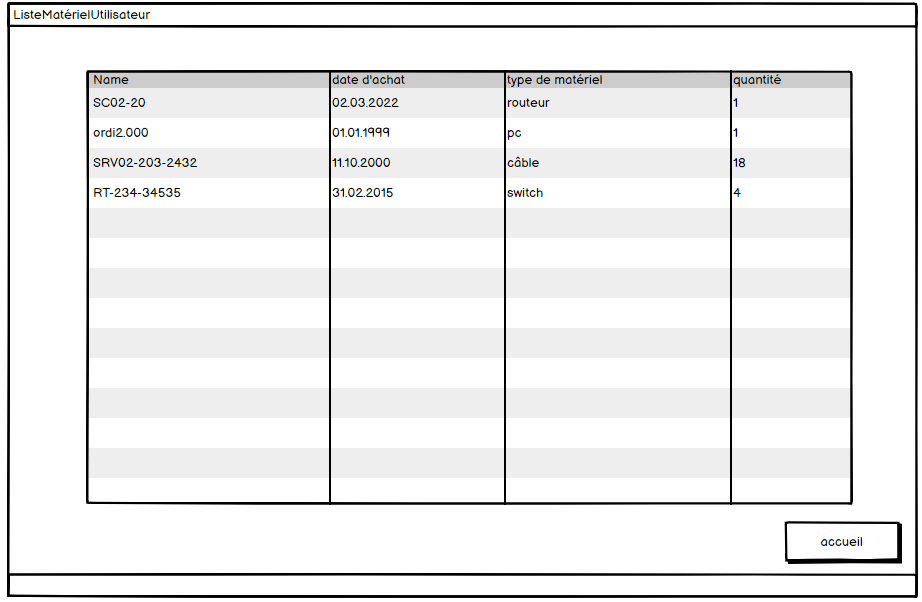
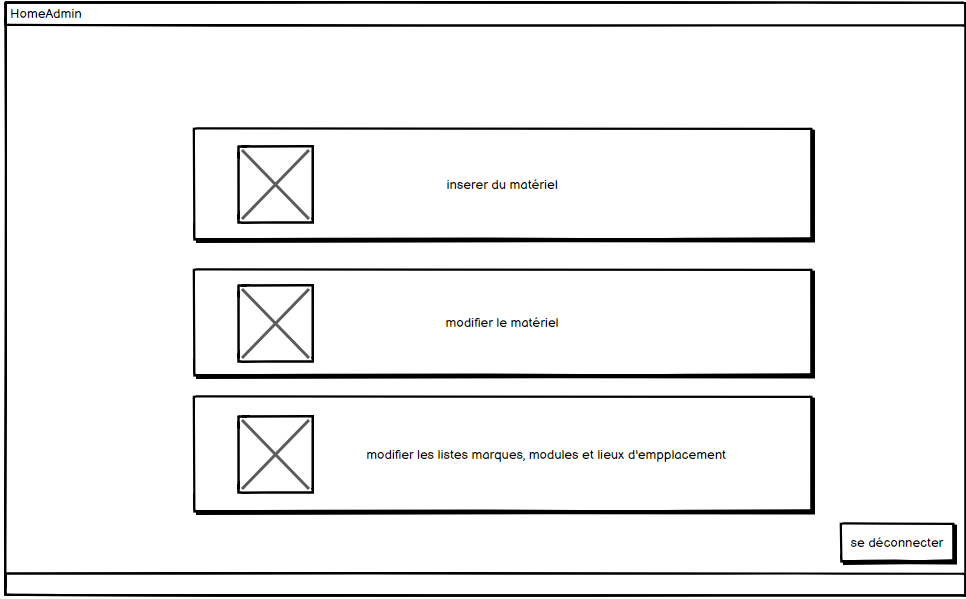
Modifier les listes 5



Accueil admin 6

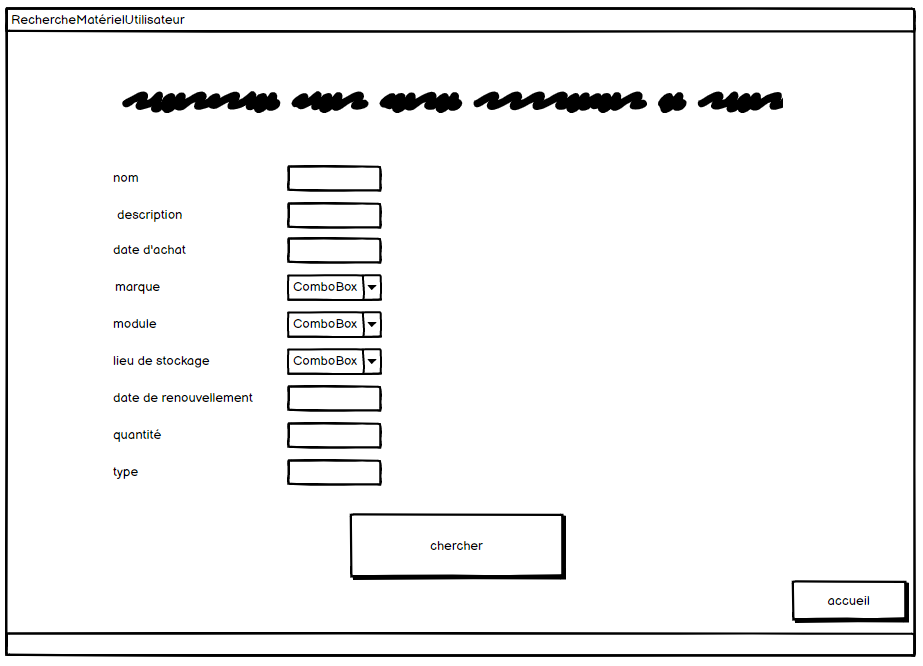


Liste matériel admin 7

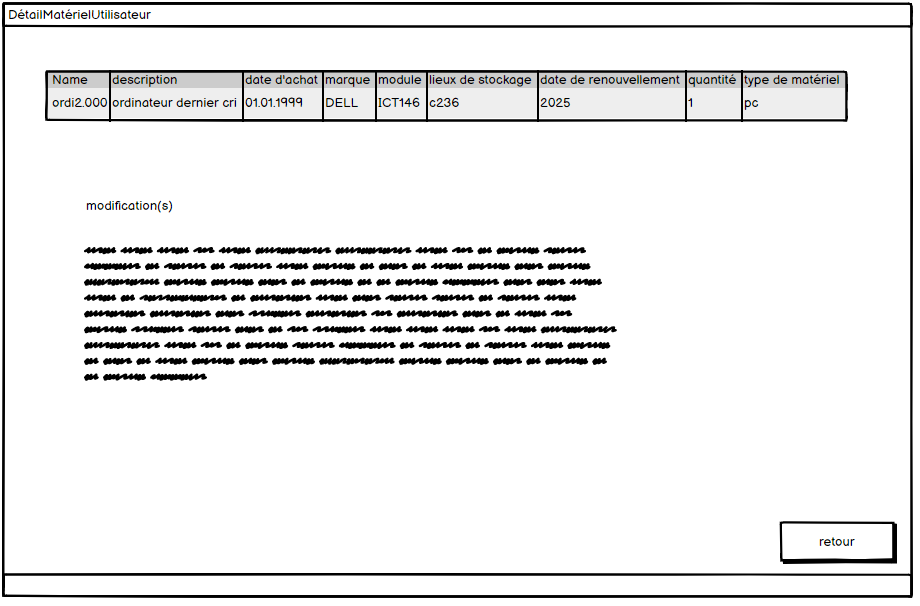


Liste matériel utilisateur 8

Recherche matériel 9



Détail matériel 10



Voici les scénarios et Use case pour montrer les fonctionnalités de l’application, les scénarios seront utilisé pour effectuer les tests système car ils assurent que toutes les fonctionnalités marchent.



Scénario partie 1/2 11



Scénario partie 2/2 12

## Stratégie de test

La stratégie de teste sera faite en pyramide, en premier les tests unitaires pour chaque fonction importante, les tests d’intégrations seront fait quand plusieurs éléments interagissent entre eux pour s’assurer du fonctionnement et enfin les tests systèmes qui permettent de tester les scénarios écris dans la partie concept de ce projet. Une classe de test automatique sur une classe prévue pour ce projet devra être fait. Les données seront fictives mais elles seront inspirées des cours que j’ai effectué. Les tests seront effectués par moi-même.

## Planification



planification 13

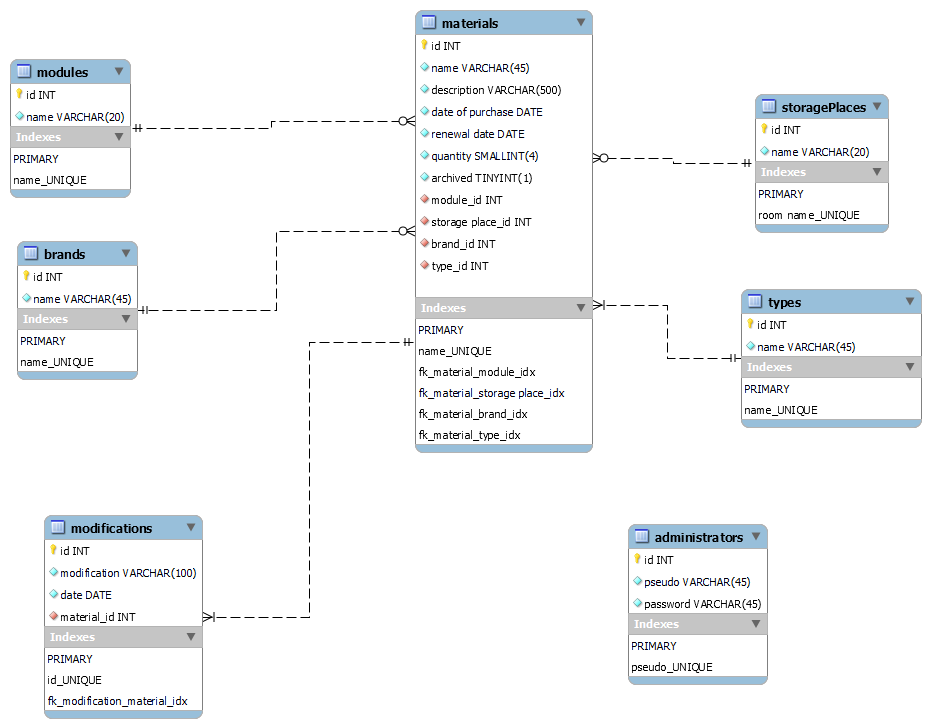
## Dossier de conception

Le matériel utilisé sera 1 PC en configuration standard CPNV avec accès à internet.

Le système d’exploitation sera Windows 10.

L’outil principale utilisé sera Microsoft Visual studio pour coder en C# ainsi que les applications de réalisation comme balsamiq, Excel, draw.io, asta, MySQL Workbench, mariadb, HeidiSQL. Un outil de gestion de version git.

Voici le MLD de ce projet :



MLD 14

# Réalisation

## Dossier de réalisation

Dossier de réalisation 15

Le dossier est composé de deux parties, la première, c’est le projet, code, donc qui contiennent Front qui contient l’affichage du projet, Classe qui contient les classes utiliser dans le code, Tests qui contient les tests automatiques effectuer et le .sln qui est la solution du projet. La deuxième partie concerne l’analyse/conception du projet avec comme contenue les User cases/scénarios, le rapport du TPI, la planification, le journal de travail, la maquette, le cahier des charges, le MCD/MLD, avec les scriptes de créations de base de données et d’ajout de données, et les captures d’écran.

## Classes

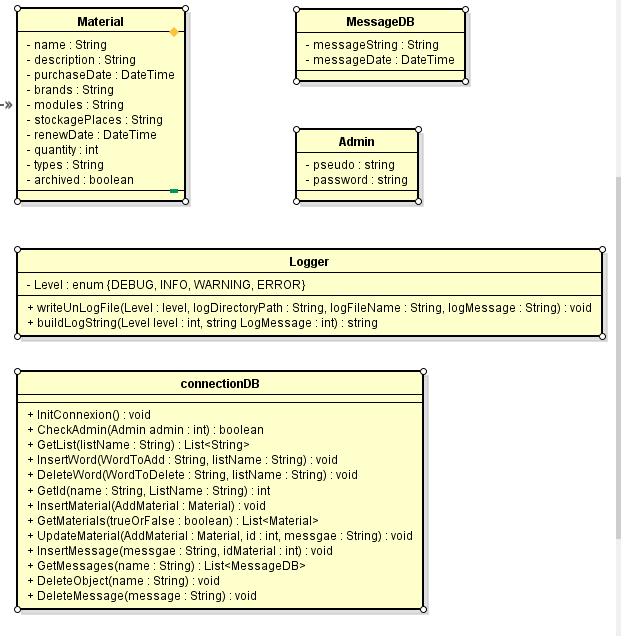
Voici le digramme de classe de ce projet

Diagramme de classe 16

La classe material me permet de créer un objet qui est similaire au material dans la base de données ce qui me permet de manipuler plus simplement les caractéristiques d’un matériel.

La classe MessageDB permet de pouvoir manipuler plus facilement les modifications dans la base de données.

La classe Admin permet de pouvoir gérer les logins de l’admin

La classe Logger permet de pouvoir écrire les erreurs de connexion dans un .txt, la fonction buildLogString va écrire une phrase avec tous les paramètres cette fonction est utilisé dans writeUnLogFile qui permet d’écrire la phrase dans un fichier .txt

La classe connection permet de faire des requêtes SQL à la base de données et de récupérer les données :

* InitConnexion permet d’établir la connexion avec la base de données, c’est ici que l’on peut modifier les paramètres de connexion a la bdd.
* CheckAdmin permet de vérifier si l’admin entrez est similaire a un admin dans la base de données si c’est le cas alors la fonction va renvoyer true sinon elle rendra false.
* GetList permet de récupérer des listes de string, cette fonction est utilisée pour afficher le contenue dans les listes déroulantes.
* InsertWord permet d’ajouter des mots dans les listes déroulantes.
* DeleteWord permet de supprimer des mots dans les listes déroulantes.
* GetId permet de récupérer l’id d’un mot dans une liste qui est utile dans l’insertion de matériel car les listes sont liées avec un clé étrangère donc pour simplifier la requête, l’on va directement insérer l’id.
* InsertMaterial permet d’ajouter du matériel dans la base de données.
* GetMaterial permet de retourner la liste de matériel dans la base de données, si un true est insérer alors les matériaux archivés sont récupéré si c’est un false alors les matériaux archivés ne sont pas récupérés.
* UpdateMaterial permet de mettre à jour un matériel que l’on souhaite modifier.
* InsertMessage permet d’insérer une modification à un matériel.
* GetMessage permet de récupérer la liste de toutes les modification d’un matériel et de les retourner.
* DeleteObject permet de supprimer un matériel dans la base de données, cette fonction est utilisé dans les tests.
* DeleteMessage permet de supprimer une modification, cette fonction est utilisée dans les tests.

## Description des tests effectués

### Tests unitaire/integration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| numéro du test | description | date du test | résultat attendu | résultat |
| 1 | CheckAdminTest 17 | 08.05.23 | La fonction CheckAdmin retourne true si un admin similaire est trouvé dans la bdd | réussit |
| 2 | CheckAdminFalseTest 18 | 08.05.23 | La fonction CheckAdmin retourne false si il n’y pas d’admin similaire dans la bdd | réussit |
| 3 | InsertDeleteWordGetListTest 1/2 19 | 08.05.23 | La fonction InsertWord va ajouter un mot dans une liste de la bdd et la fonction getList va récupérer la liste. | réussi |
| InsertDeleteWordGetListTest 2/2 20 | La fonction DeleteWord va enlever un mot d’une liste dans la bdd |
| 4 | GetId 21 | 08.05.23 | La fonction GetId va retourner l’id en int d’un élément d’une liste | réussi |
| 5 | CHeckMaterialInsertAndGetMaterial 22 | 16.05.23 | La fonction InsertMaterial va insérer le matériel dans la base de données et la fonction GetMaterial va retourner une liste de matériel dont le dernier de la liste est le matériel qu’on vient de créer | Réussi |
| 5 | CheckMaterialUpdate 23 | 16.05.23 | La fonction UpdateMaterial va mettre à jour le matériel que l’on a modifier puis avec la fonction getMaterial l’on récupère le dernier matériel de la liste qui vient d’être modifier | réussi |
| 7 | InsertMessageAndGetMessage 24 | 16.05.23 | La fonction InsertMessage qui va être liée avec le matériel « PCTEST » va insérer un message de modification dans la base de données, la fonction getMessages va rendre une liste de messages et la dernière insérer va rendre le message inséré. | réussi |

### Tests système

|  |  |  |
| --- | --- | --- |
| **Numéro du scénario** | **date** | **résultat** |
| 1 | 17.05.2023 | réussi |
| 2 | 22.05.2023 | réussi |
| 3 | 17.05.2023 | réussi |
| 4 | 17.05.2023 | réussi |
| 5 | 17.05.2023 | réussi |
| 6 | 17.05.2023 | réussi |
| 7 | 17.05.2023 | réussi |
| 8 | 17.05.2023 | réussi |

## Erreurs restantes

Il n’y a pas d’erreur connus seulement des optimisations possibles.

## Liste des documents fournis

Voici la liste de tous les documents fournis qui sont disponible dans le dépôt GitHub <https://github.com/Tristan1403/GestionInventaireInformatiqueTPI.git> :

Le rapport du TPi

La documentation initiale avec toutes la partie analyse, conception

Un manuel d’installation et d’utilisation

Les scriptes pour créer la base de données et y insérer des données

L’application avec un .exe pour l’ouvrir

# Conclusions

### Objectifs atteints / non-atteints

Tous les tests ont été réussis, les scénarios ainsi que les tests unitaires. Les points techniques ont été fait et respecté donc mon application est opérationnel. Les points un peu moins réussit sont que le code n’est pas optimisé partout, j’aurai encore pu faire de réunissage du code. La partie des tests automatique n’est pas forcément optimisée comme je le voudrai.

### Points positifs / négatifs

La documentation initiale s’est faite assez naturellement et rapidement don ça ne m’a pas posé de problème. Le code est toujours agréable, c’est pour cela que j’ai choisi de faire de mon TPI, du code. Le fait d’avoir fait un pré TPI a rendu les deux points précèdent bien plus simple à effectuer. Avec ce temps gagner ça m’a permis de passer plus de temps sur la documentation qui est pour moins le point qui me passionne le moins est donc j’ai pu compenser. Il y pas plus de points négatifs, car le TPI, c’est bien passé.

### Difficultés particulières

Les seules difficultés se trouvaient dans le code et dans les tests automatiques, pour le code les dataGridView étaient totalement nouveau pour moi alors j’ai dû en premier lieux faire les recherches puis commencer à tester différentes choses ce qui a finis par payer. Pour les tests automatiques, j’avais aussi moins l’habitude et je trouvais que pour y ai un minimum de cohérence, il fallait que les tests ne laissent pas des traces dans la base de données donc j’ai dû créer des fonctions supplémentaires pour effacer les données qui étaient insérées dans la base de données.

### Suites possibles pour le projet (évolutions & améliorations)

Les améliorations possibles seraient du peaufinage, car c’est de l’optimisation du code, mettre plus de commentaire, éviter les doublons, créer des boucles au lieux d’enchaîner les if ou bien améliorer l’apparence de l’application. Pour les évolutions du projet, on pourrait ajouter des critères pour mieux décrire le matériel. On pourrait héberger la base de données pour avoir des données communes, car pour l’’instant, c’est une application locale.

# Annexes

## Glossaire

|  |  |
| --- | --- |
| combobox | Liste déroulante |
| C# | Langage de programmation orienté Object |
| dataGridView | Elément utiliser en C# pour afficher un tableau |
| if | Utilisé dans la programmation permet de signifier une condition |
| MCD | Modèle Conceptuel de Données, représentation logique du stockage de données |
| MLD | Modèle Logique de Données comme un MCD mais adaptation pour le modèle technologique |
| SQL | Structured Query Language, language qui permet de communiquer avec une base de données |
| Terme | Définition |
| TPI | Travail Pratique Individuel |

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

### Situation de départ

Au début, il a fallu commencer par analyser le cahier des charges puis créer une planification initiale. Les objectifs du cahier ont été retranscrits dans le rapport. La méthode utilisée, en cascade pour ce projet a été expliquée. Puis la documentation initiale à commencer avec les Use case/scénarios, la maquette, le MCD/MLD, la stratégie de test, la planification puis la partie réalisation a été commencé.

### Mise en œuvre

Pour la réalisation, il se devait d’avoir un moyen de sauvegarde, GitHub a été utilisé pour ce projet, il est disponible dans les documents fournis. La base de données a dû être créée et des données de base en été inséré. Il a fallu commencer par la partie IHM avec la mise en place des formes décrite dans la maquette. Puis il y a la partie des classes avec la connexion à la base de données et la création d’objet avec les constructeurs. Vient la réalisation des objectifs écrit dans la planification donc l’aide à l’insertion de données, l’insertion des informations du matériel, l’historique du matériel et la consultation du matériel avec critère. Une fois, ces objectifs terminés, On en vient à la partie tests avec la création d’une partie de test automatique de la classe de connexion à la base de données. Ces tests ont été effectués en parallèle avec la réalisation du projet, ces tests se nomment des tests unitaires. Il y a par la suit les tests système qui sont plus généraux et qui suivent la stratégie de teste. Ils consistent à suivre les scénarios/Use case et si le résultat est similaire alors le teste est réussi.

### Résultat

Pour le résultat, vu que tous les tests sont passés les objectifs sont atteints. Les bugs encore présent et connus doivent être expliqué avec une solution, il n’y a pas de bug connu dans mon cas donc il reste la conclusion du rapport dans lequel je décris que le fait d’avoir fait un pré TPI m’a donné une bonne expérience et j’ai pu donc corriger la plupart de mes erreurs, je suis aussi bien plus à l’aise en C# que je l’étais avant ce qui m’a permis de gagner du temps pour pouvoir passez plus de temps dans le rapport.

## Sources – Bibliographie

### Internet

Méthode en cascade consulté le 03.05.2023 :

<https://www.ionos.fr/digitalguide/sites-internet/developpement-web/modele-en-cascade/#:~:text=Le%20mod%C3%A8le%20en%20cascade%20reposant,%2C%20impl%C3%A9mentation%2C%20test%20et%20exploitation>.

Combobox en read-only consulté le 08.05.2023

https://stackoverflow.com/questions/3061042/how-do-i-set-combobox-read-only-or-user-cannot-write-in-a-combo-box-only-can-sel

Recharger une forme C# consulté le 08.05.2023

https://www.codeproject.com/Questions/646695/how-to-reload-form-or-refresh-form-in-csharp-net

Comment mettre des données dans un DataGridView consulté le 09.05.2023

<https://stackoverflow.com/questions/29814305/how-to-load-list-data-into-datagridview-in-c-sharp>

Créer un évente en double cliquant sur le datagrid consulté le 09.05.2023

https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridview.rowheadermousedoubleclick?view=windowsdesktop-7.0&viewFallbackFrom=net

Comment récupéré les données d’un dataGridView consulté le 09.05.2023

https://learn.microsoft.com/en-us/answers/questions/372644/pass-from-data-grid-to-textbox-in-c

Message Input, consulté le 09.05.2023

https://stackoverflow.com/questions/10797774/messagebox-with-input-field

Pour éviter une exception quand la première ligne est rendue invisible, consulté le 11.05.2023

<https://www.developpez.net/forums/d2100367/dotnet/langages/csharp/rendre-invisible-ligne-datagridview/>

Faire un retour à la ligne dans un label, consulté le 15.05.2023

<https://www.developpez.net/forums/d511035/dotnet/langages/csharp/retour-chariot-string/>

Enlever les doublons dans une liste, consulté le 15.05.2023

<https://www.tutorialspoint.com/remove-duplicates-from-a-list-in-chash#:~:text=Use%20the%20Distinct()%20method,from%20a%20list%20in%20C%23>

## Journal de travail



Journal de travail 1/3 25



Journal de travail 2/3 26



Journal de travail 3/3 27

## 

## Manuel d'Installation

1. Pour l’installation de l’application, il faut en premier lieux cloner ou télécharger le contenue dans le GitHub : <https://github.com/Tristan1403/GestionInventaireInformatiqueTPI>
2. Il faut maintenant récupérer le script d’installation de base de données et créer la bdd avec comme admin de la bdd « AdminGestionInventaire » et le mot de passe doit être « Pa$$w0rd ». Puis insérer les données avec le script d’insertion de données.
3. Il faut maintenant compiler la solution qui se trouve dans GestionInventaire\GestionInventaireInformatique\GestionInventaire.sln
4. Maintenant un .exe est créer dans GestionInventaire\GestionInventaireInformatique\GestionInventaireFront\bin\Debug\net6.0-windows\GestionInventaireFront.exe

## Manuel d'Utilisation

### Partie admin

Lorsque vous double cliquer sur GestionInventaire.exe vous allez arriver sur cette page

Veuillez cliquer sur « partie administrateur » puis veuillez entrer le pseudo « admin » et le mot de passe « Pa$$w0rd » pour pouvoir accéder à la partie administrateur.

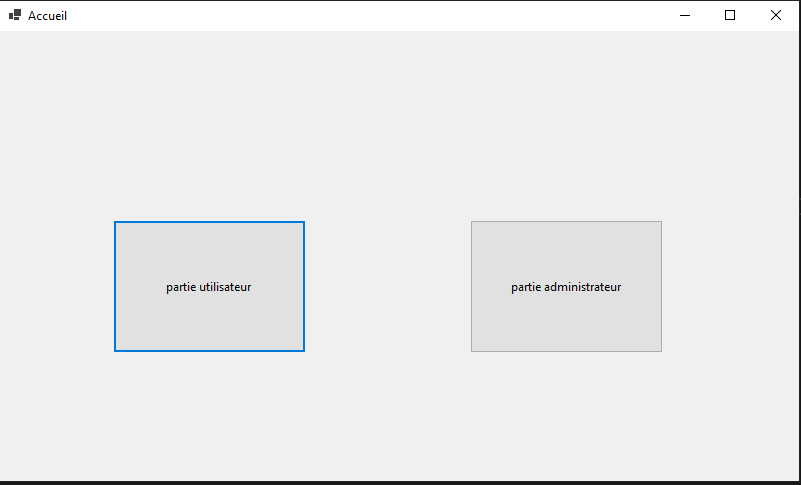
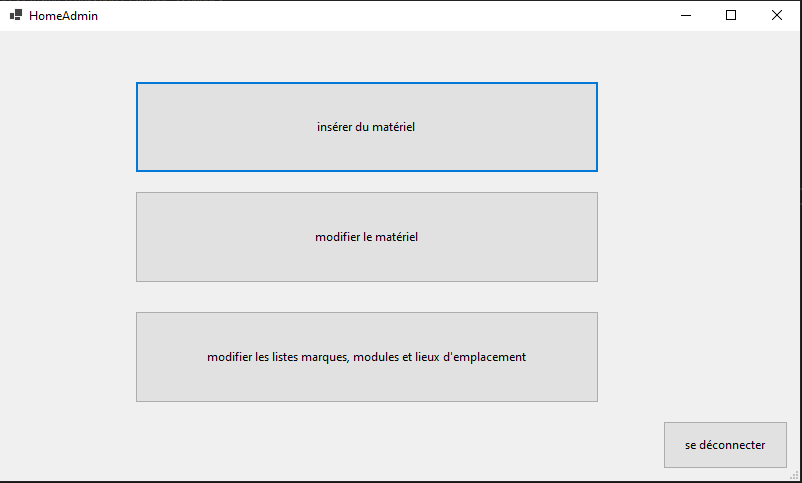


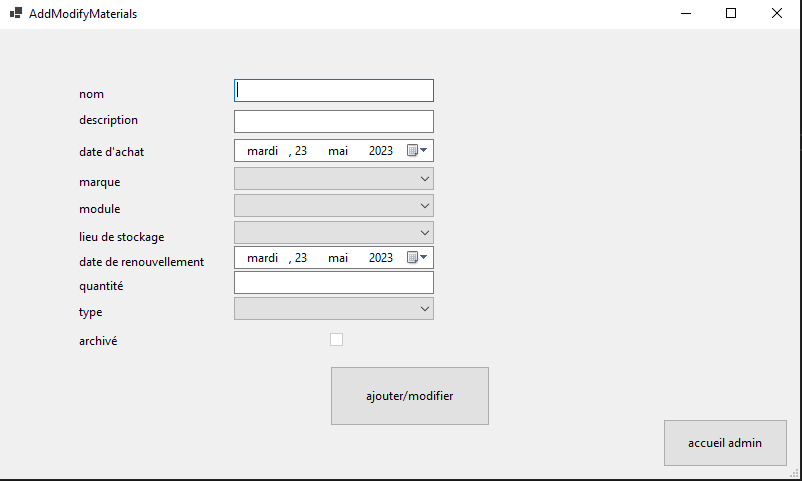
Figure 28

Accueil manuel 29

Vous allez à présent vous retrouver sur cette page :

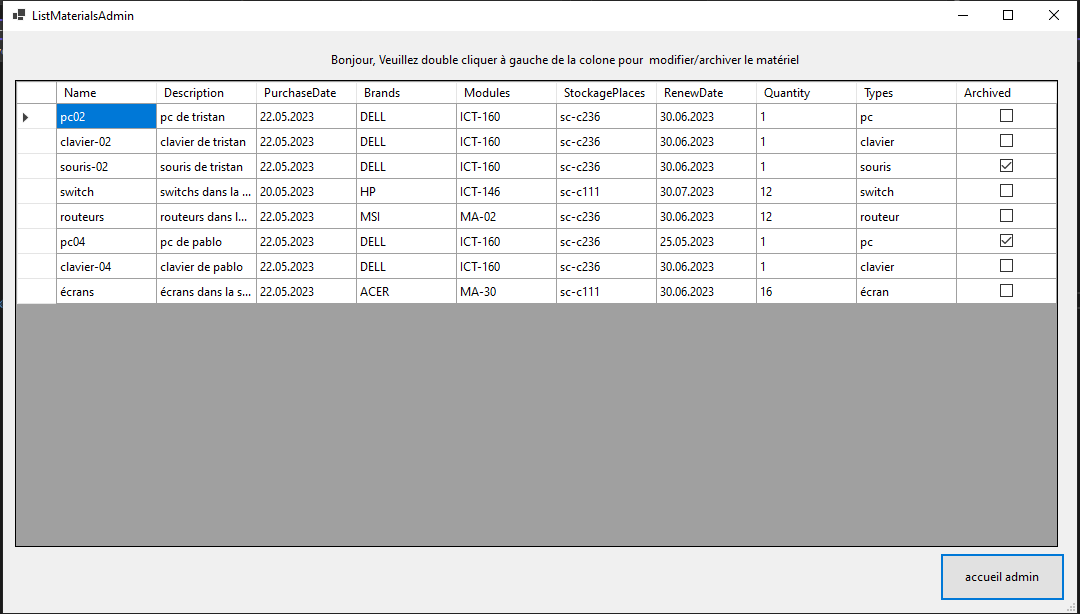
Accueil admin manuel 30



Vous avez maintenant le choix entre 3 types d’action on va commencer par l’insertion de matériel, pour cela veuillez cliquer sur « insérer du matériel ». Vous allez vous retrouvez sur cette page :

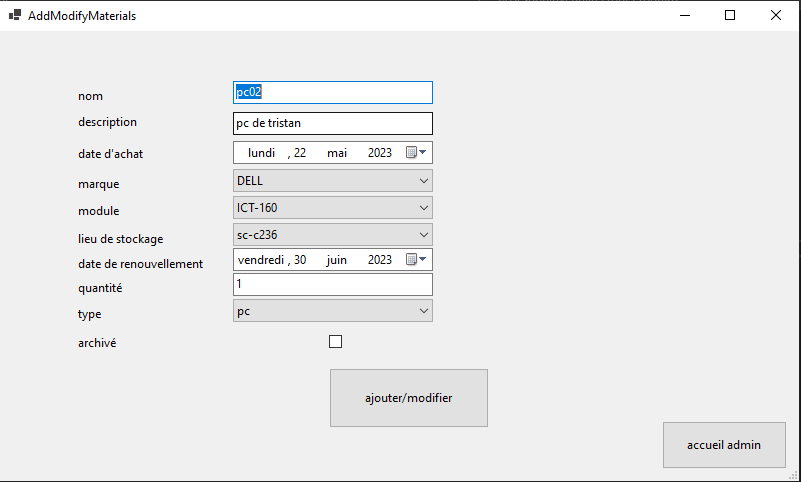
Ajouter/modifier manuel 31

Pour insérer il y a des règles, tous les champs doivent être remplis, la quantité doit être un chiffre, la date de renouvèlement doit être après la date d’achat. Puis il faut appuyer sur ajouter/modifier.

Pour la partie modification du matériel, veuillez revenir à l’accueil et appuyer sur « modifier le matériel » puis vous allez arriver sur cette page :

Liste matériel admin manuel 32

Vous pouvez voir tous le matériel dans la base de données, pour modifier il faut double cliquer à gauche du matériel que l’on veut modifier.

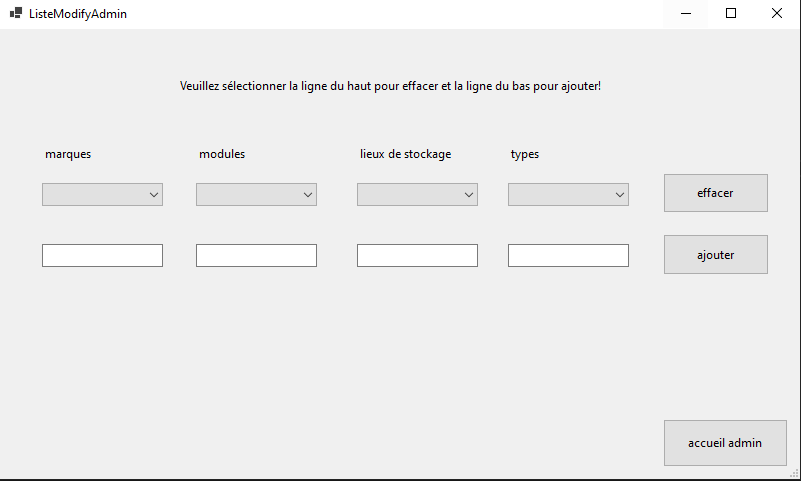
Puis l’on arrive sur cette page avec les données du matériel écrite.

Pour modifier le matériel il faut simplement faire les changements puis appuyer sur ajouter/modifier, il est possible d’archiver le matériel en cochant archiver.

Modifier manuel 33

Pour la partie modification des listes marques, modules, lieux de stockage et types, il faut revenir à l’accueil et appuyer sur modifier les listes. Vous allez arriver sur cette page :

Modifier liste manuel 34

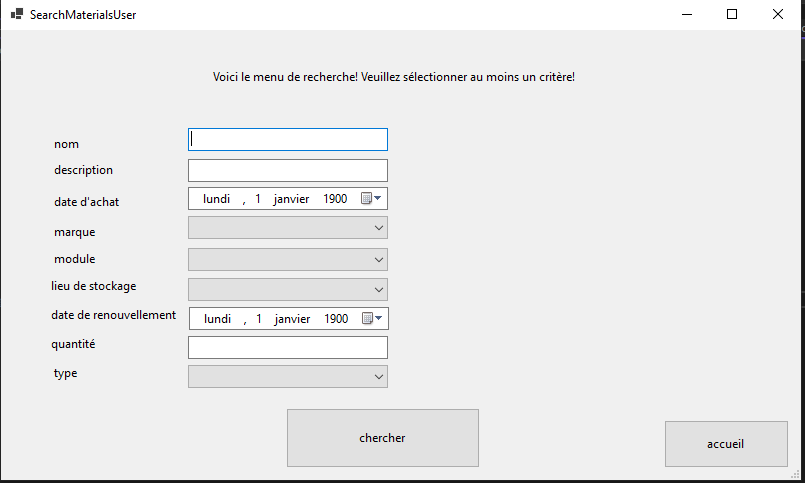


Pour ajouter un mot dans une liste, il faut écrire en bas et pour effacer, il faut sélectionner un mot dans la combobox.

### Partie User

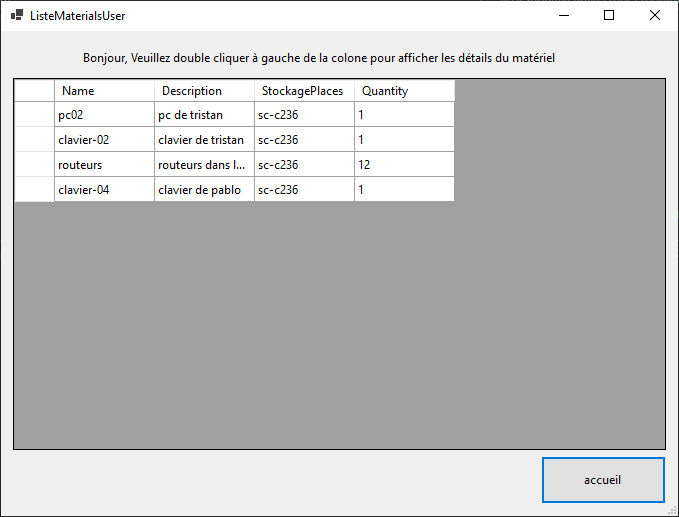
Pour cette partie il faut sélectionner partie utilisateur dans l’accueil. Puis vous allez arriver sur cette page :

Ajouter manuel 35



Puis cette page affiche les critères principaux qui sont le nom, la description et la quantité puis affiche les critères rechercher.

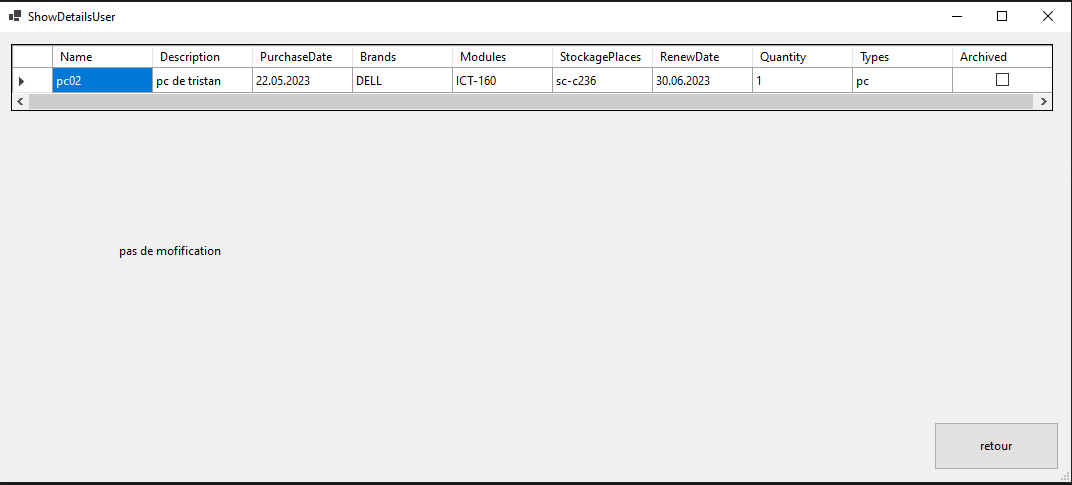
Liste matériel utilisateur 36



Pour voir les détails et modifications effectués sur le matériel, il faut double cliquer à gauche du matériel.

Puis si l’on double clic sur La partie à gauche du matériel l’on va arriver sur cette page

Détail matériel 37



On voit sur cette page tous les critères du produit ainsi que les modifications effectuées. Pour revenir à la recherche effectuer il faut appuyer sur retour.

## Table des illustration

[Méthode de gestion, en cascade1 3](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355197)

[Accueil 2 5](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355198)

[login 3 5](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355199)

[Ajouter/modifier 4 6](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355200)

[Modifier les listes 5 6](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355201)

[Accueil admin 6 7](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355202)

[Liste matériel admin 7 7](#_Toc136355203)

[Liste matériel utilisateur 8 8](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355204)

[Recherche matériel 9 8](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355205)

[Détail matériel 10 9](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355206)

[Scénario partie 1/2 11 10](#_Toc136355207)

[Scénario partie 2/2 12 11](#_Toc136355208)

[planification 13 12](#_Toc136355209)

[MLD 14 13](#_Toc136355210)

[Dossier de réalisation 15 14](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355211)

[Diagramme de classe 16 15](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355212)

[CheckAdminTest 17 17](#_Toc136355213)

[CheckAdminFalseTest 18 17](#_Toc136355214)

[InsertDeleteWordGetListTest 1/2 19 17](#_Toc136355215)

[InsertDeleteWordGetListTest 2/2 20 18](#_Toc136355216)

[GetId 21 18](#_Toc136355217)

[CHeckMaterialInsertAndGetMaterial 22 18](#_Toc136355218)

[CheckMaterialUpdate 23 19](#_Toc136355219)

[InsertMessageAndGetMessage 24 19](#_Toc136355220)

[Journal de travail 1/3 25 25](#_Toc136355221)

[Journal de travail 2/3 26 26](#_Toc136355222)

[Journal de travail 3/3 27 27](#_Toc136355223)

[Figure 28 28](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355224)

[Accueil manuel 29 28](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355225)

[Accueil admin manuel 30 29](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355226)

[Ajouter/modifier manuel 31 29](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355227)

[Liste matériel admin manuel 32 30](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355228)

[Modifier manuel 33 31](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355229)

[Modifier liste manuel 34 31](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355230)

[Ajouter manuel 35 32](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355231)

[Liste matériel utilisateur 36 33](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355232)

[Détail matériel 37 34](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355233)

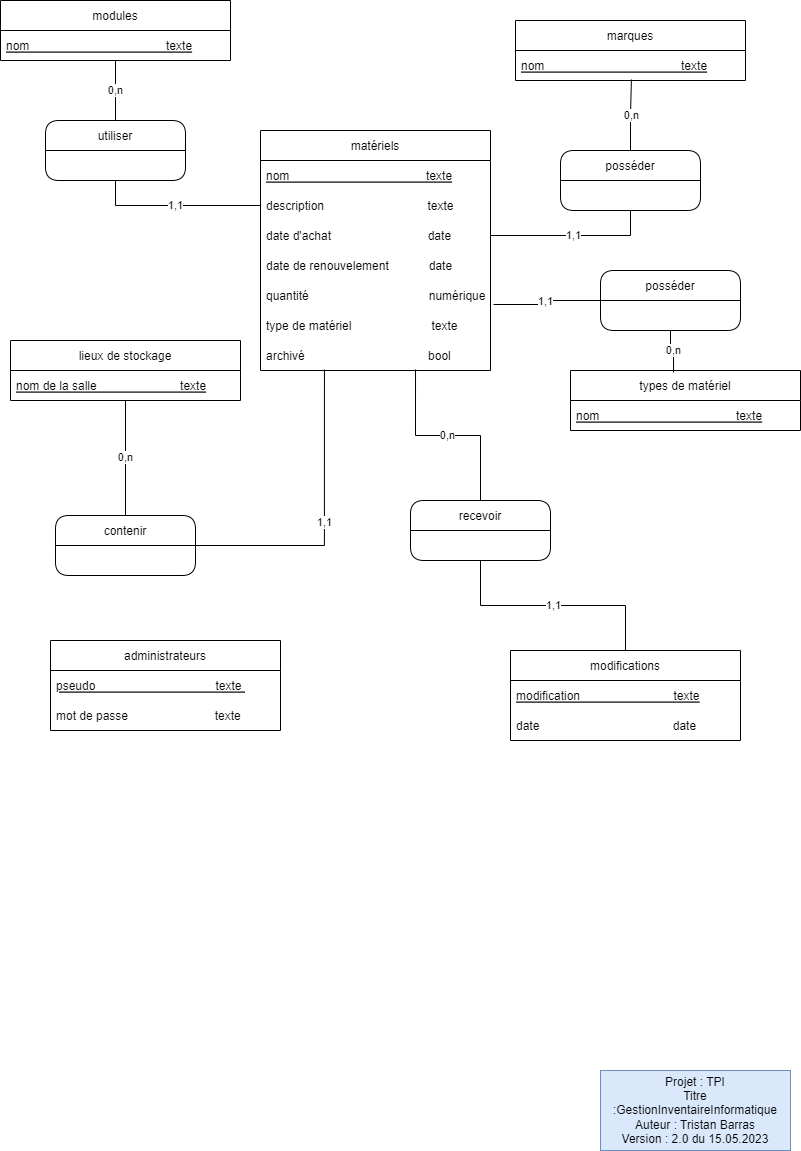
[MCD 38 37](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355234)

[CDC 1/3 39 38](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355235)

[CDC 2/3 40 39](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355236)

[CDC 3/3 41 40](file:///C:\Users\tristan.barras\source\repos\GestionInventaire\documentation\RapportTPITristanBarras.docx#_Toc136355237)

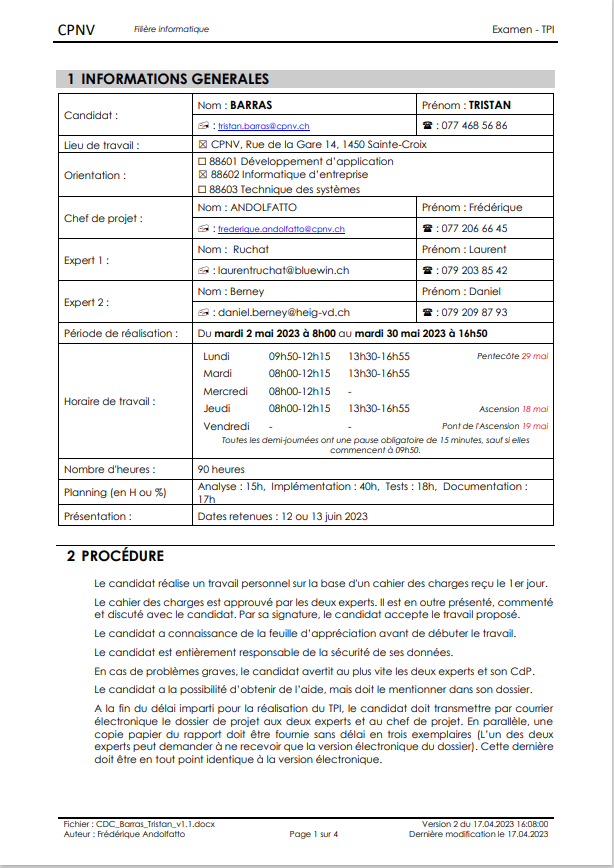
## Archives du projet



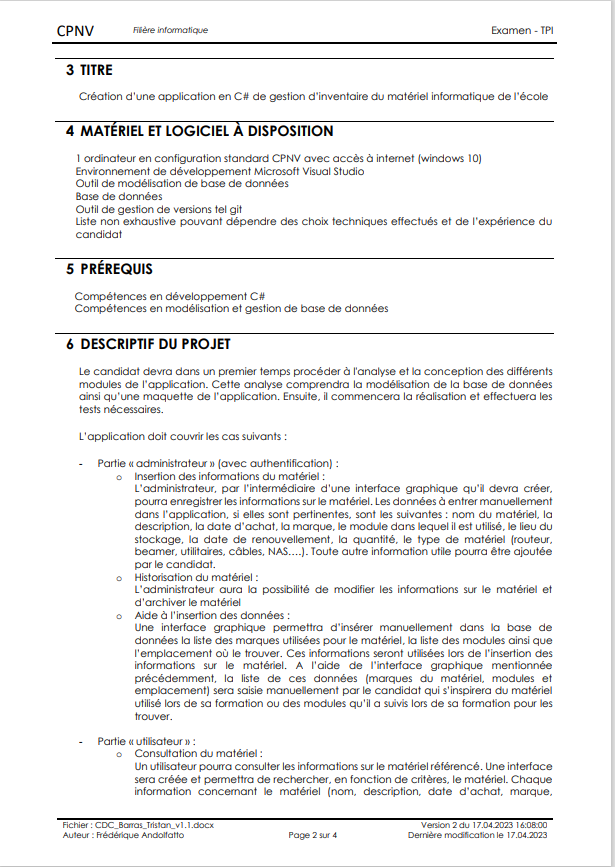
### MCD

MCD 38

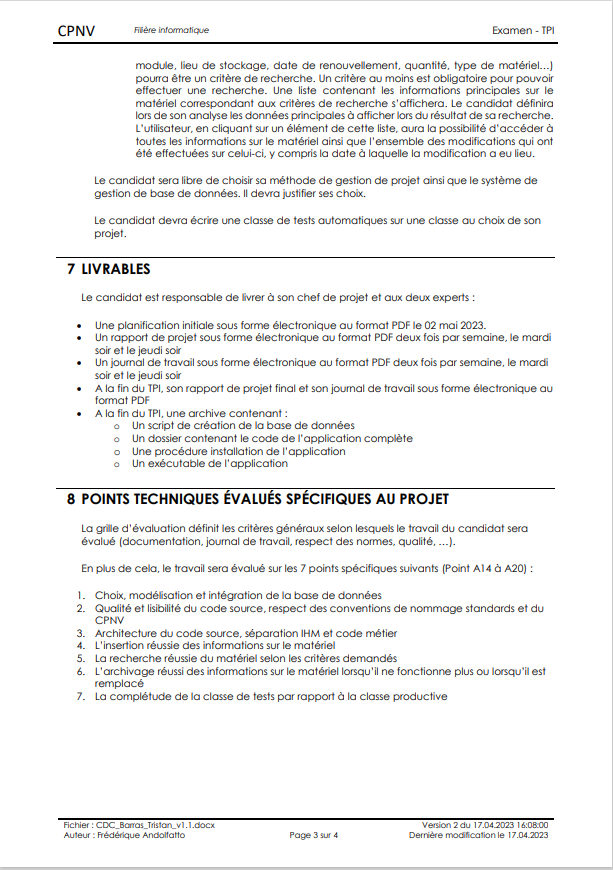
### Cahier des Charges



CDC 1/3 39



CDC 2/3 40



CDC 3/3 41