

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

30/05/2023

Application C# de gestion d'inventaire informatique

Several thin, curved lines in dark blue and light grey that originate from the bottom left and sweep upwards and to the right.

Candidat : BARRAS Tristan

CHEF DE PROJET : ANDOLFATTO FRÉDÉRIQUE

TABLE DES MATIERES

Analyse préliminaire	3
Introduction	3
Objectifs	3
Méthode de gestion de projet	3
Planification initiale.....	4
Analyse / Conception	5
Concept	5
Stratégie de test	11
Planification	12
Dossier de conception.....	13
Réalisation.....	14
Dossier de réalisation.....	14
Classes	15
Description des tests effectués	17
Tests unitaire/integration	17
Tests système	20
Erreurs restantes.....	20
Liste des documents fournis	20
Conclusions	21
Objectifs atteints / non-atteints.....	21
Points positifs / négatifs.....	21
Difficultés particulières	21

Suites possibles pour le projet (évolutions & améliorations)	21
Annexes.....	22
Glossaire.....	22
Résumé du rapport du TPI / version succincte de la documentation	23
Situation de départ	23
Mise en œuvre	23
Résultat	23
Sources – Bibliographie.....	24
Internet	24
Journal de travail.....	25
Manuel d'Installation	27
Manuel d'Utilisation.....	28
Partie admin.....	28
Partie User.....	32
Table des illustration.....	35
Archives du projet	37
MCD	37
Cahier des Charges.....	38

ANALYSE PRÉLIMINAIRE

INTRODUCTION

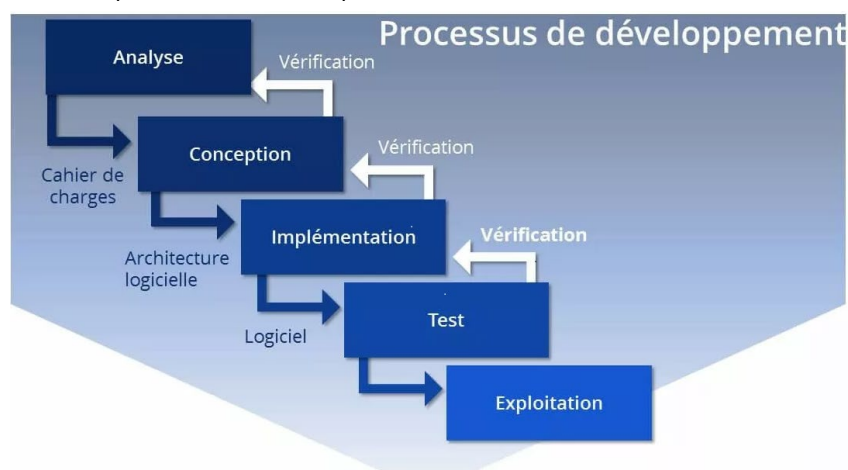
Ce projet est réalisé dans le cadre d'un TPI, il y a 90 heures de travail. Le projet est une application en C# de gestion d'inventaire de matériel informatique de l'école. Elle permettra de parcourir, ajouter, modifier du matériel informatique présent dans une base de données. Pour ce projet, j'ai des 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 effectuée pour voir concrètement l'apparence de l'application :

Accueil

partie utilisateur

partie administrateur

Accueil 2

LoginAdmin

nom d'utilisateur

mot de passe

se connecter

accueil

login 3

Ajouter/ModifierMatériels

nom	<input type="text"/>
description	<input type="text"/>
date d'achat	<input type="text"/>
marque	ComboBox ▼
module	ComboBox ▼
lieu de stockage	ComboBox ▼
date de renouvellement	<input type="text"/>
quantité	<input type="text"/>
type	<input type="text"/>
archiver	<input type="checkbox"/>

ajouter/modifier

accueil

Ajouter/modifier 4

ModifierListeAdmin

marques	modules	lieux	types	
ComboBox ▼	ComboBox ▼	ComboBox ▼	ComboBox ▼	effacer
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	insérer

accueil

Modifier les listes 5


ListMatérielAdmin


Name	date d'achat	module	archiver	modifier
SC02-20	02.03.2022	ICT146	<input type="checkbox"/>	modifier
ordi2.000	01.01.1999		<input checked="" type="checkbox"/>	modifier
SRV02-203-2432	11.10.2000	ICT110	<input type="checkbox"/>	modifier
RT-234-34535	31.02.2015	ma04	<input checked="" type="checkbox"/>	modifier


accueil

Liste matériel admin 7

HomeAdmin

insérer du matériel

modifier le matériel

modifier les listes marques, modules et lieux d'emplacement

se déconnecter

Accueil admin 6

Recherche

MatérielUtilisateur

nom

description

date d'achat

marque

ComboBox

module

ComboBox

lieu de stockage

ComboBox

date de renouvellement

quantité

type

chercher

accueil

Recherche matériel 9

[illegible]

Liste matériel utilisateur 8

DétailMatérielUtilisateur								
Name	description	date d'achat	marque	module	lieux de stockage	date de renouvellement	quantité	type de matériel
ordi2.000	ordinateur dernier cri	01.01.1999	DELL	ICT146	c236	2025	1	pc

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.

tristan Barras	description	condition	réaction
Use case: l'admin modifie les listes de mot disponible			
	1) lancement de l'application		
1			
	l'admin double-click sur le .exe	L'utilisateur doit avoir téléchargé le projet et y avoir accès.	L'application est lancée et l'admin voit la fenêtre d'accueil.
	2) enregistrement de l'admin		
2			
	L'admin suit le scénario 1 et appuie sur le bouton disponible "partie administrateur" .		Une nouvelle fenêtre s'ouvre et les textbox pseudo et mot de passe apparaissent.
	L'admin entre son pseudo et son mot de passe.	L'admin doit avoir un compte dans la base de donnée.	L'admin est connecté et voit 4 boutons, insérer du matériel, modifier le matériel, modifier les listes marques, modules et lieux d'emplacement et un bouton accueil.
	3) modifier les listes		
3			
	L'utilisateur suit le scénario 1 et 2 puis appuie sur le bouton modifier les listes marques, modules et lieux d'emplacement.		Une nouvelle fenêtre s'ouvre et l'on voit trois textbox, trois combobox, un bouton effacer, un bouton insérer ainsi qu'un bouton accueil.
	L'admin sélectionne un mot disponible dans la combobox marques puis appuie sur effacer.		Le mot sélectionné est effacé de la base de données et n'est plus disponible dans la liste utilisée.
	L'admin écrit un mot dans la textbox modules puis appuie sur insérer.		Le mot qui est écrit dans la textbox est inséré dans la base de données et l'admin peut le voir dans la combobox.
	L'admin appuie sur effacer sans sélectionner un mot ou appuie sur insérer sans écrire un mot dans une textbox.	Aucun mot ne doit être sélectionné et les textbox doivent être vides.	Une messagebox apparaît et indique à l'admin qu'il doit sélectionner/écrire un mot.
Use case: l'admin insère du matériel dans la base de données			
	4) insérer le matériel		
4			
	Les scénarios 1 et 2 ont été effectués et l'admin appuie sur le bouton insérer du matériel.		Une nouvelle fenêtre apparaît et on voit tous les critères qui représentent un objet informatique et 2 boutons : ajouter/modifier et accueil.
	L'admin appuie sur ajouter sans remplir tous les champs.	L'admin n'a pas rempli tous les champs.	Un pop-up avertit l'admin qu'il n'a pas remplis tous les champs nécessaires.
	L'admin appuie sur ajouter en remplissant tous les champs .	Tous les champs doivent être remplis.	Un nouveau matériel informatique a été ajouté dans la base de données.

Scénario partie 1/2 11

Use case: l'admin modifie/archive le matériel			
	5) modifier le matériel		
5	Les scénarios 1 et 2 ont été effectués et l'admin appuie sur le bouton modifier le matériel.		Une nouvelle fenêtre s'ouvre avec la liste de tout le matériel disponible dans la base de données.
	L'admin appuie sur le mot modifier à droite du matériel qu'il veut modifier.		Une nouvelle fenêtre apparaît et on voit tous les critères remplis qui représentent l'objet informatique et 2 boutons : ajouter/modifier et accueil.
	L'admin modifie le module attribuer à l'objet puis appuie sur ajouter/modifier.		La fenêtre avec toute la liste du matériel s'ouvre et l'on peut voir que la modification a été effectuée.
	6) archiver le matériel		
6	Les scénarios 1 et 2 ont été effectués et l'admin appuie sur le bouton modifier le matériel.		Une nouvelle fenêtre s'ouvre avec la liste de tout le matériel disponible dans la base de données.
	L'admin appuie sur le mot modifier à droite du matériel qu'il veut modifier.		Une nouvelle fenêtre apparaît et on voit tous les critères remplis qui représentent l'objet informatique et 2 boutons : ajouter/modifier et accueil.
	L'admin va cocher sur le radio bouton archivé puis va appuyer sur ajouter/modifier.		La fenêtre avec la liste de tout le matériel s'ouvre et l'on peut voir que l'objet a été modifié.
Use case: un utilisateur veut consulter des informations sur le matériel			
	7) aller au menu utilisateur		
7	Le scénario 1 doit être fait, l'utilisateur va appuyer sur le bouton partie utilisateur.		Une nouvelle fenêtre s'ouvre avec tous les critères disponibles du matériel informatique.
	8) consultation du matériel		
8	Les scénarios 1 et 7 doivent être faits, l'utilisateur appuie sur chercher.	L'utilisateur ne doit avoir choisi aucun critère.	Un pop-up apparaît en disant qu'il faut choisir au moins un critère.
	Les scénarios 1 et 7 doivent être faits, l'utilisateur appuie sur chercher après avoir choisi un type de matériel.	L'utilisateur doit avoir choisis au moins un critère.	Une nouvelle fenêtre s'ouvre avec les critères de base nom, date d'achat, type de matériel, quantité et les critères spécifiques choisis par l'utilisateur.
	L'utilisateur va cliquer sur un élément de la liste.		Une nouvelle fenêtre s'ouvre avec tous les critères sur le matériel, les modifications effectuées sur le matériel avec les dates et un bouton retour.

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 écrits 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

Projet	Barras Tristan		Planification					
Application C# pour gestion d'inventaire informatique			02.05.23	08.05.23	15.05.23	22.05.23	30.05.23	
	Total							
Prévu (périodes)	84h40(pause de 5min pas prise en compte)		17h15	22h30	15h45	22h30	6h40	
temps réel	90 h 00		18 h 40	24 h 10	16 h 15	23 h 50	7 h 05	
		SEM	1	2	3	4	5	
1 Analyse								15 h 15
								13 h 15
	11 Planification initial	Prévu	1 h 30					1 h 30
		temps réel	1 h 30					1 h 30
	12 MCD/MLD	Prévu	3 h 45					3 h 45
		temps réel	1 h 30					1 h 30
	13 Elaboration des stratégies de test	Prévu	1 h 30					1 h 30
		temps réel	0 h 30					0 h 30
	14 Documentation Initiale	Prévu	8 h 00					8 h 00
		temps réel	9 h 00					9 h 00
	15 création de la base de donné	Prévu	0 h 30					0 h 30
		temps réel	0 h 45					0 h 45
2 IHM								3 h 00
								1 h 50
	21 création des pages de base partie admin/utilisateur	Prévu	1 h 00					1 h 00
		temps réel	0 h 30					0 h 30
	22 mise en place de tous les boutons/texte	Prévu	1 h 00	0 h 30				1 h 30
		temps réel	1 h 05					1 h 05
	23 contrôle des normes de nommage CPNV	Prévu		0 h 30				0 h 30
		temps réel	0 h 15					0 h 15
3 code métier								37 h 15
								22 h 55
	31 connexion à la base de donné	Prévu		1 h 00				1 h 00
		temps réel	0 h 30					0 h 30
	32 mise en place des classes et fonction de base pour celle-ci	Prévu		2 h 00				2 h 00
		temps réel	0 h 35					0 h 35
	33 aide à l'insertion de données	Prévu		10 h 00				10 h 00
		temps réel	1 h 15	2 h 45				4 h 00
	34 insertion des informations du matériel	Prévu		6 h 30				6 h 30
		temps réel		5 h 15				5 h 15
	35 historisation du matériel	Prévu			6 h 00			6 h 00
		temps réel		3 h 05				3 h 05
	36 consultation du matériel avec critère	Prévu			7 h 45	4 h 00		11 h 45
		temps réel		5 h 05	4 h 25			9 h 30
4 Tests								14 h 30
								16 h 05
	41 Tests automatique d'une classe	Prévu				6 h 00		6 h 00
		temps réel		1 h 30	2 h 10			3 h 40
	42 Tests scénario/ tests unitaires	Prévu				4 h 30		4 h 30
		temps réel			1 h 35			1 h 35
	43 Correction de bugs	Prévu				4 h 00		4 h 00
		temps réel		4 h 00		6 h 50		10 h 50
5 Documentation final								10 h 40
								20 h 15
	51 Rapport de projet	Prévu		2h00	2h00	3 h 00		3 h 00
		temps réel		1 h 15	4 h 25	8 h 20		14 h 00
	52 guide d'installation de l'application	Prévu				1 h 00		1 h 00
		temps réel						
	53 finalisation de la documentation	Prévu					6 h 40	6 h 40
		temps réel					6 h 15	6 h 15
	Autre	Prévu (périodes)						
		temps réel	1 h 15	1 h 15	3 h 40	8 h 40	0 h 50	15 h 40

planification 13

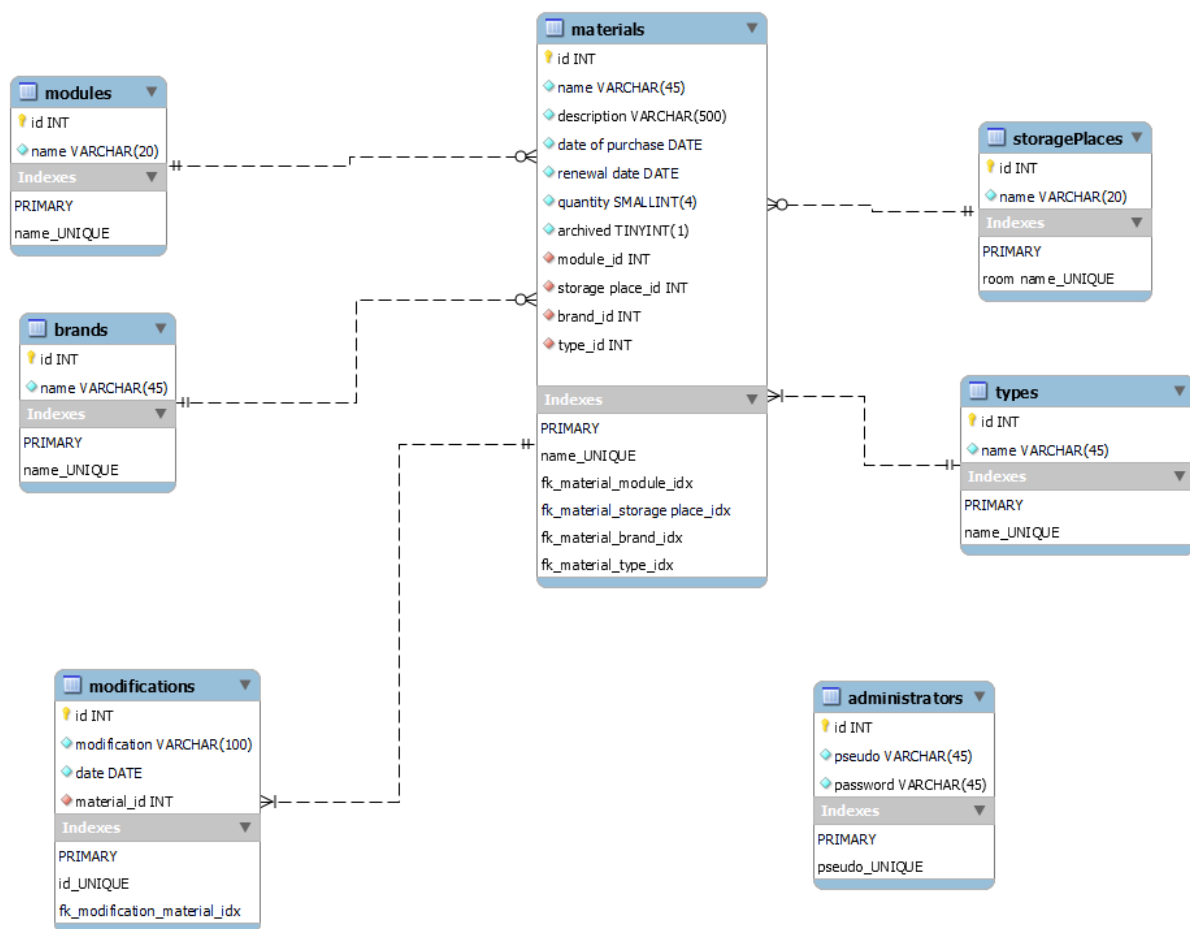
DOSSIER DE CONCEPTION

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.

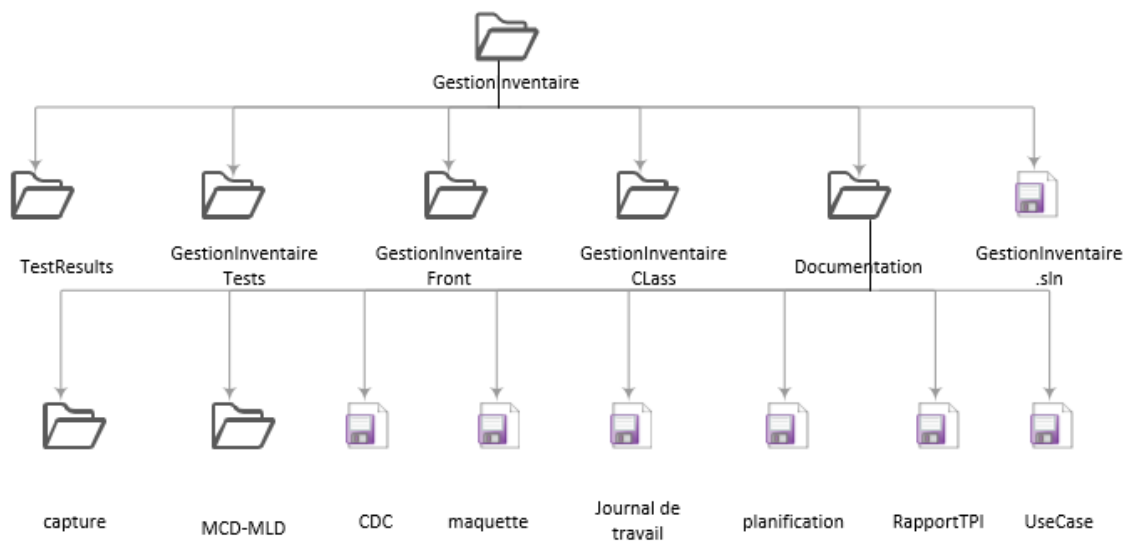
Voici le MLD de ce projet :



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 scripts 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

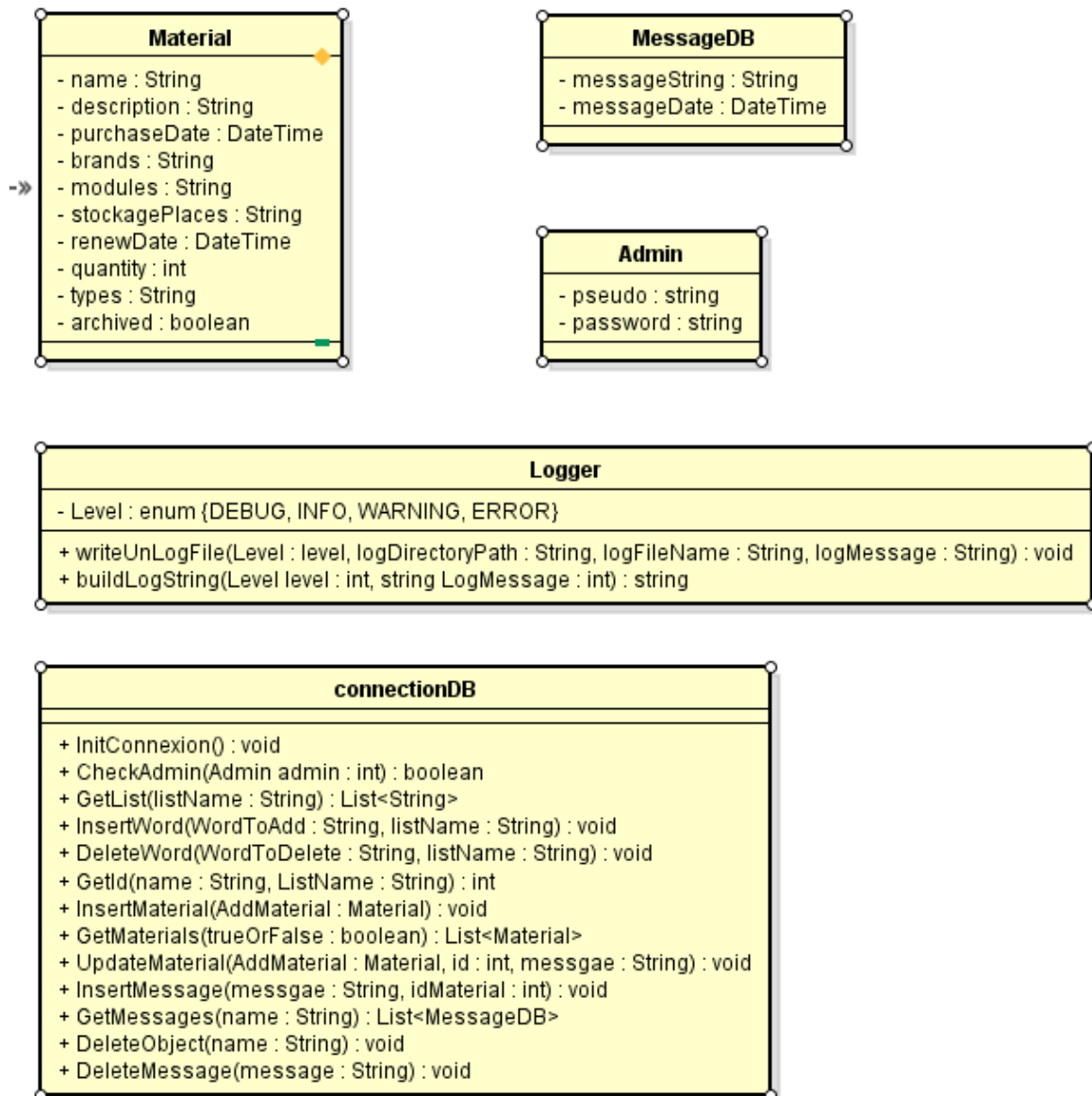


Diagramme de classe 16

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 contenu 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	<pre> [Test] 0 0 references Tristan, 41 minutes ago 1 author, 1 change public void CheckAdminTest() { //Arrange ConnectionDB bdd = new ConnectionDB(); Admin tritri = new Admin("tristan", "Pa\$\$w0rd"); bool resActual = false; //Act resActual = bdd.CheckAdmin(tritri); //Assert Assert.IsTrue(resActual); } </pre> <p>CheckAdminTest 17</p>	08.05.23	La fonction CheckAdmin retourne true si un admin similaire est trouvé dans la bdd	réussit
2	<pre> [Test] 0 0 references 0 changes 0 authors, 0 changes public void CheckAdminFalseTest() { //Arrange ConnectionDB bdd = new ConnectionDB(); Admin tritri = new Admin("hello", "Pa\$\$w0rd"); bool resActual = false; //Act resActual = bdd.CheckAdmin(tritri); //Assert Assert.IsFalse(resActual); } </pre> <p>CheckAdminFalseTest 18</p>	08.05.23	La fonction CheckAdmin retourne false si il n'y pas d'admin similaire dans la bdd	réussit
3	<pre> [Test] 0 0 references 0 changes 0 authors, 0 changes public void InsertDeleteWordGetListTest() { //Add and Delete are together to not leave stuff in the DB //Add Section //Arrange ConnectionDB bdd = new ConnectionDB(); List<string> listAddxpected = new List<string>(); listAddxpected = bdd.GetList("storageplaces"); listAddxpected.Add("SC-C444"); List<string> listAdd = new List<string>(); //Act bdd.InsertWord("SC-C444", "storageplaces"); listAdd = bdd.GetList("storageplaces"); //Assert Assert.AreEqual(listAddxpected, listAdd); } </pre> <p>InsertDeleteWordGetListTest 1/2 19</p>	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

	<pre>//Delete Section //Arrange List<string> listDeleteExpected = new List<string>(); listDeleteExpected = bdd.GetList("storageplaces"); listDeleteExpected.Remove("SC-C444"); List<string> listDelete = new List<string>(); //Act bdd.DeleteWord("SC-C444", "storageplaces"); listDelete = bdd.GetList("storageplaces"); //Assert Assert.AreEqual(listDeleteExpected, listDelete);</pre> <p>InsertDeleteWordGetListTest 2/2 20</p>		La fonction DeleteWord va enlever un mot d'une liste dans la bdd	
4	<pre>[Test] 0 references 0 changes 0 authors, 0 changes public void GetIdTest() { //Arrange ConnectionDB bdd = new ConnectionDB(); string storageplaces = "SC-C111"; int idStoragePlaces = -1; int idExpected = 2; //Act idStoragePlaces = bdd.GetId(storageplaces, "storageplaces"); //Assert Assert.AreEqual(idExpected, idStoragePlaces); }</pre> <p>GetId 21</p>	08.05.23	La fonction GetId va retourner l'id en int d'un élément d'une liste	réussi
5	<pre>public void CheckMaterialInsertAndGetMaterial() { //Arrange ConnectionDB bdd = new ConnectionDB(); List<material> listMaterialTest = new List<material>(); material materialTest = new material(); materialExpected.Name = "PCTEST"; materialExpected.Description = "PC de test"; materialExpected.PurchaseDate = DateTime.Now.Date; materialExpected.Brands = "HP"; materialExpected.Modules = "ICT-160"; materialExpected.StockagePlaces = "SC-C236"; materialExpected.RenewDate = DateTime.Now.Date; materialExpected.Quantity = 1; materialExpected.Types = "pc"; materialExpected.Archived = false; //Act bdd.InsertMaterial(materialExpected); listMaterialTest = bdd.GetMaterials(false); //get the last created material materialTest = listMaterialTest.Last(); //delete the material to let the DB clean bdd.DeleteObject("PCTEST"); //Assert Assert.AreEqual(materialExpected.Name, materialTest.Name); }</pre> <p>CheckMaterialInsertAndGetMaterial 22</p>	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	<pre> [Test] 0 references 0 changes 0 authors, 0 changes public void CheckMaterialUpdate() { //Arrange ConnectionDB bdd = new ConnectionDB(); List<material> listMaterialTest = new List<material>(); material materialTest = new material(); materialExpected.Name = "PCTEST"; materialExpected.Description = "PC de test"; materialExpected.PurchaseDate = DateTime.Now.Date; materialExpected.Brands = "HP"; materialExpected.Modules = "ICT-160"; materialExpected.StockagePlaces = "SC-C236"; materialExpected.RenewDate = DateTime.Now.Date; materialExpected.Quantity = 1; materialExpected.Types = "pc"; materialExpected.Archived = false; //Act bdd.InsertMaterial(materialExpected); int id = bdd.GetId(materialExpected.Name, "materials"); //Update materialExpected.Name = "PCTEST2"; bdd.UpdateMaterial(materialExpected, id, "messgae Test Update"); listMaterialTest = bdd.GetMaterials(false); //get the last created material materialTest = listMaterialTest.Last(); //delete the material to let the DB clean bdd.DeleteMessage("messgae Test Update"); bdd.DeleteObject("PCTEST2"); //Assert Assert.AreEqual(materialExpected.Name, materialTest.Name); } </pre> <p>CheckMaterialUpdate 23</p>	16.05.23	<p>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</p>	réussi
7	<pre> [Test] 0 references 0 changes 0 authors, 0 changes public void InsertMessageAndGetMessage() { //Arrange ConnectionDB bdd = new ConnectionDB(); List<MessageDB> ListMessage = new List<MessageDB>(); MessageDB messageExpected = new MessageDB(); messageExpected.MessageDate = DateTime.Now; messageExpected.MessageString = "messgae Test2 Update"; //create a material to be able to crate a messgae List<material> listMaterialTest = new List<material>(); material materialTest = new material(); materialExpected.Name = "PCTEST"; materialExpected.Description = "PC de test"; materialExpected.PurchaseDate = DateTime.Now.Date; materialExpected.Brands = "HP"; materialExpected.Modules = "ICT-160"; materialExpected.StockagePlaces = "SC-C236"; materialExpected.RenewDate = DateTime.Now.Date; materialExpected.Quantity = 1; materialExpected.Types = "pc"; materialExpected.Archived = false; bdd.InsertMaterial(materialExpected); int id = bdd.GetId(materialExpected.Name, "materials"); //Act bdd.InsertMessage("messgae Test2 Update", id); ListMessage = bdd.GetMessages("PCTEST"); MessageDB messageObtened = new MessageDB(); messageObtened = ListMessage.Last(); bdd.DeleteMessage("messgae Test2 Update"); bdd.DeleteObject("PCTEST"); //Assert Assert.AreEqual(messageExpected.MessageString, messageObtened.MessageString); } </pre> <p>InsertMessageAndGetMessage 24</p>	16.05.23	<p>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.</p>	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 scripts 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 donc 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 précédent 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	Élément 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, langage 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 évènement 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 récupérer 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](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

Barras Tristan		Gestion d'inventaire de matériel informatique de l'école		Aujourd'hui
Journal		script insertion de données		
Date	Sem	Activité	Heures	Remarques
02.05.2023	1	compréhension du cahier des charges	08h00-08h45	discussion sur le cahier des charges et du TPI en générale
		11 Planification initial	08h45-10h30	
		14 Documentation Initiale	10h30-12h15	création de la maquette, partie admin bientôt fini
		14 Documentation Initiale	13h30-14h30	maquette v1 complète faite sur balsamiq
		12 MCD/MLD	14h30-15h00	création du MCD
		14 Documentation Initiale	15h00-16h55	documentation: introduction/objectif/planification initiale
03.05.2023	1	14 Documentation Initiale	08h00-08h45	paragraphe méthode de travail dans le rapport de projet
		14 Documentation Initiale	08h45-11h20	Use case et scénario
		12 MCD/MLD	11h20-12h00	MLD v1 finit
		14 Documentation Initiale	12h00-12h15	MLD, planification et diagramme de Gantt intégrer
04.05.2023	1	13 Elaboration des stratégies de test	08h00-08h30	écrit dans le rapport la stratégie de test
		14 Documentation Initiale	08h30-09h15	rapport, insérer des éléments, scénarios, maquette
		15 création de la base de donné	09h15-10h00	un peu de problème, normes cpnv était fausse, les FK était des PK
		21 création des pages de base partie admin/utilisateur	10h00-10h30	création deu projet C#, form et class de base
		22 mise en place de tous les boutons/texte	10h30-11h00	mise en place de bouton avec nommage CPNV
			11h00-11h20	mail, réponse + github
		12 MCD/MLD	11h20-11h40	correction MCD-MLD
		22 mise en place de tous les boutons/texte	11h40-12h00	création de bouton, recherche data grid
		14 Documentation Initiale	12h00-12h15	journal de travail +doc
		22 mise en place de tous les boutons/texte	13h30-13h45	fin de mise en place datagrid vide
		23 contrôle des normes de nommage CPNV	13h45-14h00	
		31 connexion à la base de donné	14h00-15h05	j'ai créer le login qui me permet de tester la connexion, j'ai créer la classe logger pour me renvoyer des erreurs de connexion à la db et j'avais une faute d'orthographe dans la connexion ce qui m'a fait perdre 20 min
		32 mise en place des classes et fonction de base pour celle-ci		
		33 aide à l'insertion de données	15h20-16h35	fonction GetList créer il permet de récupérer des éléments et d'en faire une liste, insérer les données dans les combobox
			16h35-16h55	journal de travail + mail
08.05.2023	2	33 aide à l'insertion de données	09h50-11h00	création de fonction InsertWord et DeleteWord qui permettent de gérer les listes dans la db
		gestion des excption	11h00-11h45	gestion des exceptions en cas de double insérion et en cas d'échec de connexion à la db
		41 Tests automatique d'une classe	11h45-12h15	TDD et mise en place des tests dans la doc
		41 Tests automatique d'une classe	13h30-14h15	la classe connection DB sera la classe tester automatiquement car cela permet de tester toutes mes fonctions, les 4 premiers tests sont créer et effectif
		51 Rapport de projet	14h15-14h45	insertion des tests unitarie dans le rapport
		34 insertion des informations du matériel	14h45-15h05	création de la commande SQL plus création de la fonction InsertMaterial
		34 insertion des informations du matériel	15h20-16h40	création de la fonction getId, la fonction InsertMaterial terminer, la forme AddModifyMaterials est fonctionnel
		41 Tests automatique d'une classe	16h40-16h55	mise en place du test GetId et début du test InsertMatériel, doc sur le test GetId

Journal de travail 1/3 25

09.05.2023	2	34 insertion des informations du matériel	08h00-09h35	création de la fonction GetMaterial qui permet de récupérer dans une liste des objets matériel, data grid de listadmin est fonctionnel
		34 insertion des informations du matériel	09h50-11h50	recherche sur le datagrid pour pouvoir récupérer les données en double cliquant sur une ligne, et la création de cette fonctionnalité. Modification de la form addModifyAdmin pour s'adapter entre créer et modifier, en modifiant les données sont préinsérer, lien principal pour la recherche: https://learn.microsoft.com/en-us/answers/questions/372644/pass-from-data-grid-to-textbox-in-c
		51 Rapport de projet	11h50-12h15	doc + journal de travail
		35 historisation du matériel	13h30-16h35	création des fonctions UpdateMaterial et InsertMessage, modification de la form addModifyAdmin pour la partie modification, mise en place de la modification et de l'historique des messages
		51 Rapport de projet	16h35-16h55	journal de travail + mail
10.05.2023	2	43 Correction de bugs	08h00-10h00	correction de bug avec exception, commentaire dans le code, bloquer pour le teste de InsertMaterial et GetMaterial, amélioration du code supprimer des doublons exc...
		43 Correction de bugs	10h00-12h15	Théo mon camarade de classe à tester les fonctions admin et à trouver des bugs, correction de bug notamment de les excpetions concernant le format des formulairee
11.05.2023	2	33 aide à l'insertion de données	08h00-09h35	mise en place de restriction en cas de suppression d'un éléments d'une liste utilisé par un materiel
		36 consultation du matériel avec critère	09h50-12h00	mise en place des éléments de base pour cet objectif modification de la fonction getMatériel, recherche sur des fonctions lier au datagrid notmanet le fait de rendre invisible certaine colone, correction d'un bug avec getId
			12h00-12h15	journal de travail
		36 consultation du matériel avec critère	13h30-15h05	envoi de données entrez dans la recherche jusqu'au dataGrid, dataGrid affiche seulement les critères important et critère choisis
		36 consultation du matériel avec critère	15h20-16h40	les lignes sont toutes invisible jusqu'a qu'il y aie une des caractéristiques qui sont similaire, alors les lignes concerné sont visible, recherche pour les data gridsview notamment une exception qui se lance quand la première ligne est rendu invisible
			16h40-16h55	journal de travail + mail
15.05.2023	3	36 consultation du matériel avec critère	09h50-12h00	mise en place concrète de l'affichage par recherche faite dans la partie ListUser, fonction GetMessage créer qui permet de voir la liste des messgae d'un materiel, affichage des modifications dans le show détail, affichage des détails dans show détail, l'affichage des matériels avec crière affiche tous les éléments qui ont au moins un critères et non pas les éléments qui ont les 2
			12h00-12h15	journal de travail + doc
		36 consultation du matériel avec critère	13h30-16h45	mise en place de l'agoritmie de trie pour les recherches de matériel lorsque plusieurs critères étaient choisis tous ceux qui avait au moins un critère de juste s'affichait, j'ai pris pas mal de temps pour trouver une solution viable car au début je partai sur un switch avec pour chaque cas minimum 15 lignes. J'ai finis les grosses parties du code mais il manque les commentaires, le
			16h45-16h55	journal de travail + git
16.05.2023	3	51 Rapport de projet	08h00-08h30	mise en forme
		visite de l'expert	08h30-08h50	
		41 Tests automatique d'une classe	08h50-09h35	test de insertiongetMaterial modifier et cration du
		41 Tests automatique d'une classe	09h50-11h15	finalisation de la classe de test tous les tests sont créer et effectif
		51 Rapport de projet	11h15-12h15	insertion des tests unitaires dans le rapport + journal de travail
		refactorisation du code	13h30-16h00	J'ai essayé de refactoriser une partie de code qui permet de trier le matériel qui correspond au critère car j'ai 100 lignes de code alors que je suis sûr qu'il y a une manière de faire bien plus court, le code que j'ai fait est disponible en commentaire dans ListMaterialUser
		51 Rapport de projet	16h00-16h45	la maquette à été mise a jour avec les titres en dessus de chaque fenêtre, finalisation des tests.
			16h45-16h55	journal de travail + mail

17.05.2023	3	42 Tests scénario/ tests unitaires	08h00-09h35	réalisation des tests système et finalisation des tests unitaire
		51 Rapport de projet	09h50-12h00	partie réalisation, tous les tests + dossier de réalisation, changement des titres de la maquette
			12h00-12h15	journal de travail + mail
22.05.2023	4	script insertion de données	09h50-12h15	création du script avec des données à insérer
		43 Correction de bugs	13h30-15h05	correction d'un bug avec le login qui marchait plus avec les nouvelles données insérer
		51 Rapport de projet	15h20-16h40	list des documents fournis mise en forme des images maquette, mise en forme général du rapport
			16h40-16h55	journal de travail
23.05.2023	4	43 Correction de bugs	08h00-09h35	le test getList n'était pas utile car elle pouvait être intégrer à AddDeleteGetListWordTest ce qui permet de pouvoir modifier la liste de storage place sans faire un échec dans le teste
		refactorisation du code	09h50-12h15	écriture des commentaires et effacer des doublons maintenant je touche plus au code et les tests sont toujours fonctionnel
		51 Rapport de projet	13h30-16h30	erreurs retantes, sources, anuel d'utilisation, MCD
			16h30-16h55	mail + journal de travail
24.05.2023	4	43 Correction de bugs	08h00-12h00	bug pour l'aide à l'insertion de données dans lequel, on ne pouvait que rajouter/supprimer un seul éléments
			12h00-12h15	mail + journal de travail
25.05.2023	4	51 Rapport de projet	08h00-12h00	conclusion, résumé, diagramme de classe
			12h00-12h15	journal de travail
		51 Rapport de projet	13h30-16h30	description classe, glossaire, correction orthographe, petite modification
			16h30-16h55	mail + journal de travail
30.05.2023	5	53 finalisation de la documentation	08h00-12h15	finalisation des la table des illustration, manuel d'installation
		53 finalisation de la documentation	13h30-16h00	vérification que tous les points soit présent
			16h00-16h50	rendu du document

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 contenu 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

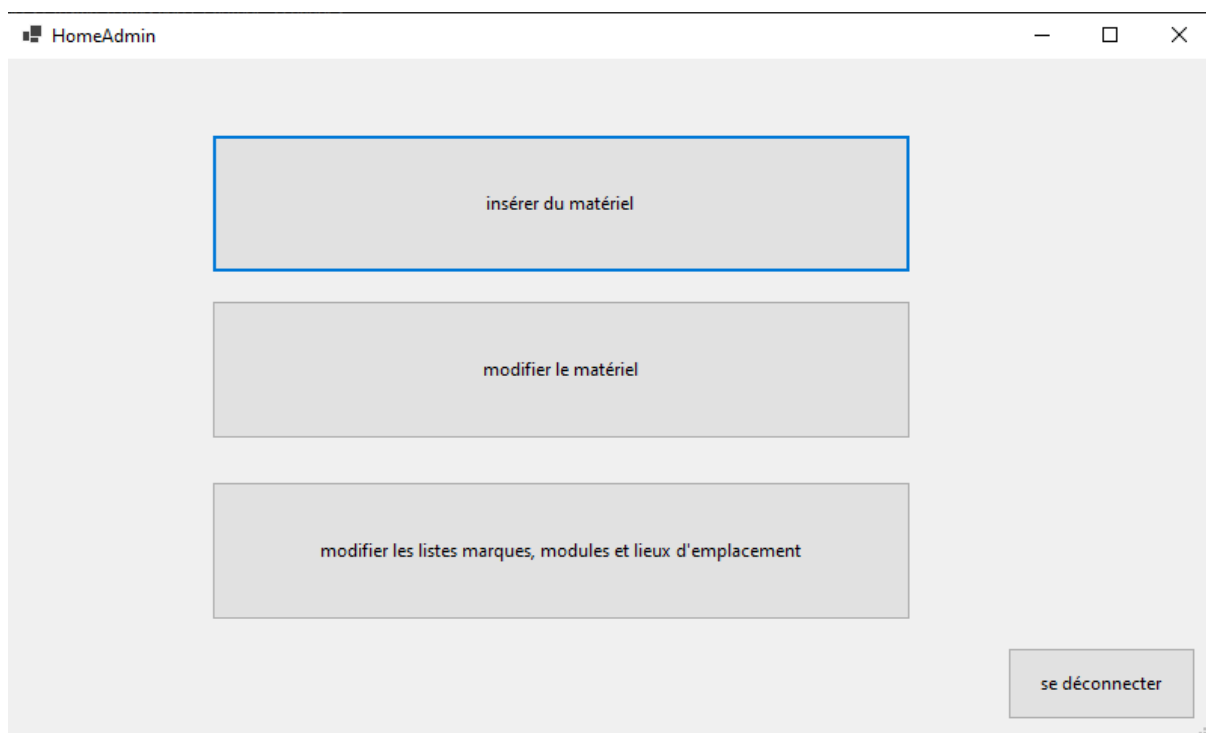
Lorsque vous double cliquez sur GestionInventaire.exe vous allez arriver sur cette page



Accueil manuel 29

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.

Vous allez à présent vous retrouver sur cette page :



HomeAdmin

insérer du matériel

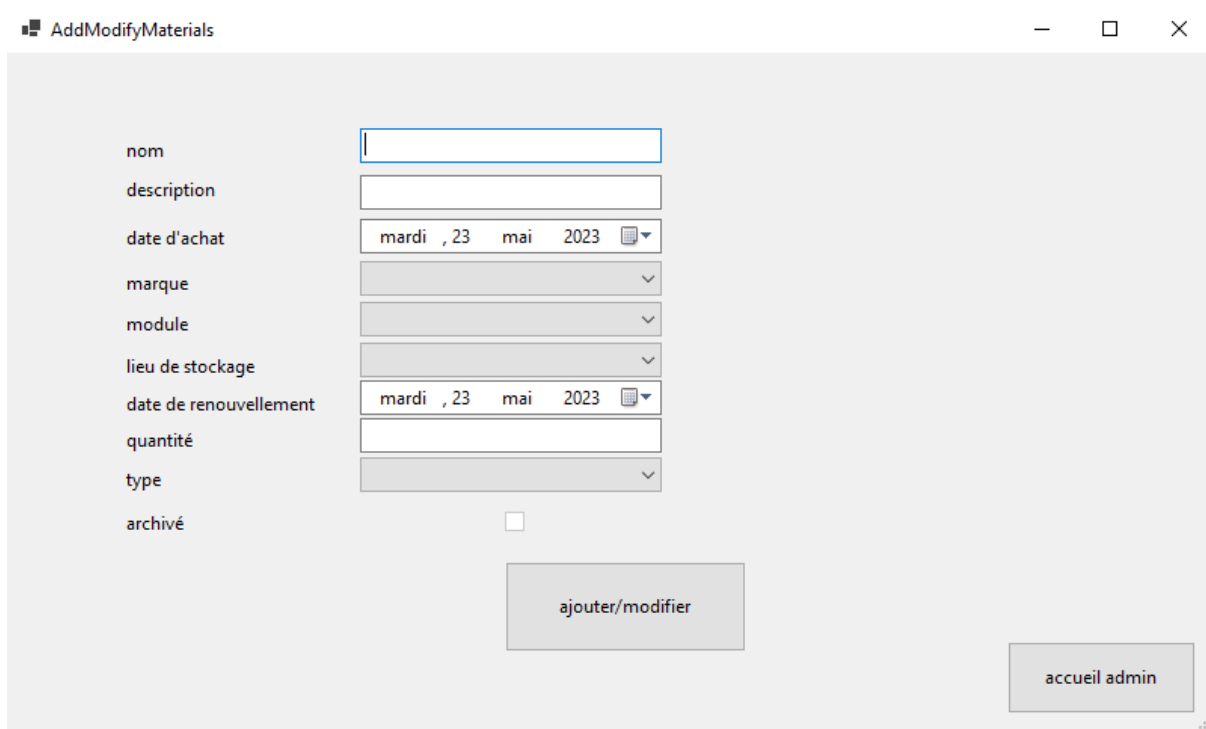
modifier le matériel

modifier les listes marques, modules et lieux d'emplacement

se déconnecter

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 retrouver sur cette page :



AddModifyMaterials

nom

description

date d'achat

marque

module

lieu de stockage

date de renouvellement

quantité

type

archivé

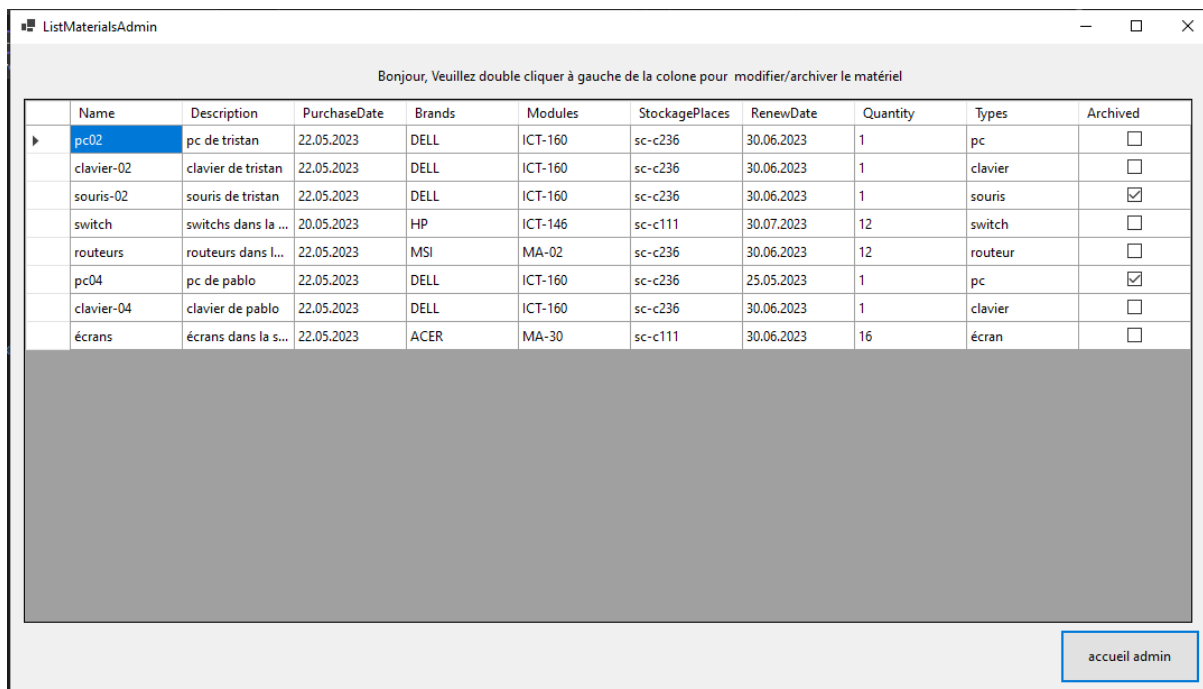
ajouter/modifier

accueil admin

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 renouvellement 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 :



	Name	Description	PurchaseDate	Brands	Modules	StockagePlaces	RenewDate	Quantity	Types	Archived
▶	pc02	pc de tristan	22.05.2023	DELL	ICT-160	sc-c236	30.06.2023	1	pc	<input type="checkbox"/>
	clavier-02	clavier de tristan	22.05.2023	DELL	ICT-160	sc-c236	30.06.2023	1	clavier	<input type="checkbox"/>
	souris-02	souris de tristan	22.05.2023	DELL	ICT-160	sc-c236	30.06.2023	1	souris	<input checked="" type="checkbox"/>
	switch	switchs dans la ...	20.05.2023	HP	ICT-146	sc-c111	30.07.2023	12	switch	<input type="checkbox"/>
	routeurs	routeurs dans l...	22.05.2023	MSI	MA-02	sc-c236	30.06.2023	12	routeur	<input type="checkbox"/>
	pc04	pc de pablo	22.05.2023	DELL	ICT-160	sc-c236	25.05.2023	1	pc	<input checked="" type="checkbox"/>
	clavier-04	clavier de pablo	22.05.2023	DELL	ICT-160	sc-c236	30.06.2023	1	clavier	<input type="checkbox"/>
	écrans	écrans dans la s...	22.05.2023	ACER	MA-30	sc-c111	30.06.2023	16	écran	<input type="checkbox"/>

accueil admin

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.

Modifier manuel 33

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.

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

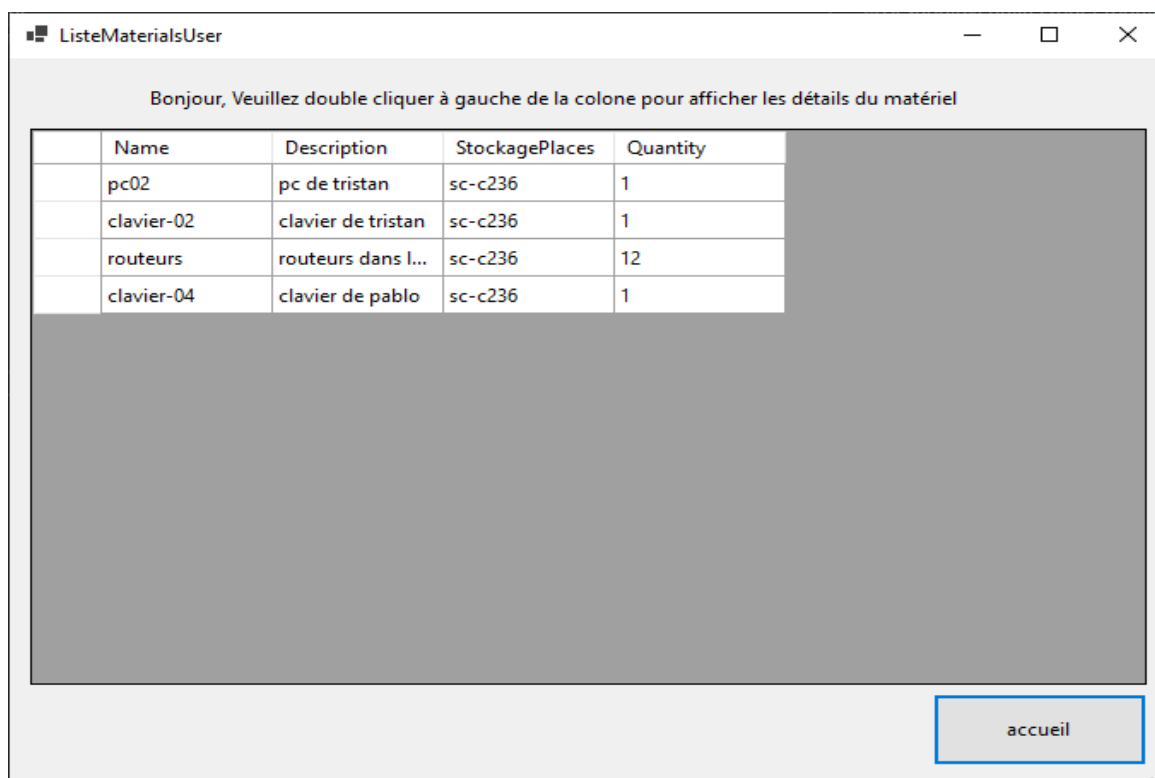
Pour cette partie il faut sélectionner partie utilisateur dans l'accueil. Puis vous allez arriver sur cette page :

The screenshot shows a window titled "SearchMaterialsUser" with standard Windows window controls (minimize, maximize, close). The main content area has a light gray background and contains the following elements:

- A message at the top: "Voici le menu de recherche! Veuillez sélectionner au moins un critère!"
- A list of search criteria on the left, each with a corresponding input field on the right:
 - nom: text input field
 - description: text input field
 - date d'achat: date picker showing "lundi , 1 janvier 1900" with a calendar icon
 - marque: dropdown menu
 - module: dropdown menu
 - lieu de stockage: dropdown menu
 - date de renouvellement: date picker showing "lundi , 1 janvier 1900" with a calendar icon
 - quantité: text input field
 - type: dropdown menu
- At the bottom center, a button labeled "chercher".
- At the bottom right, a button labeled "accueil".

Ajouter manuel 35

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

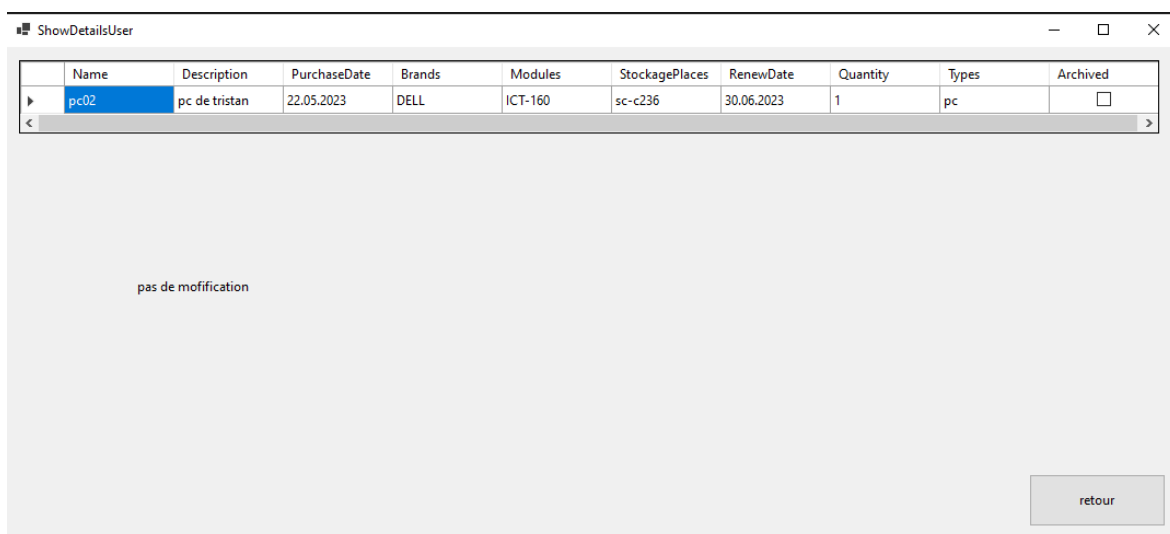


Name	Description	StockagePlaces	Quantity
pc02	pc de tristan	sc-c236	1
clavier-02	clavier de tristan	sc-c236	1
routeurs	routeurs dans l...	sc-c236	12
clavier-04	clavier de pablo	sc-c236	1

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



The screenshot shows a window titled 'ShowDetailsUser' with a table containing the following data:

	Name	Description	PurchaseDate	Brands	Modules	StockagePlaces	RenewDate	Quantity	Types	Archived
▶	pc02	pc de tristan	22.05.2023	DELL	ICT-160	sc-c236	30.06.2023	1	pc	<input type="checkbox"/>

Below the table, the text 'pas de modification' is displayed. A 'retour' button is located in the bottom right corner.

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.

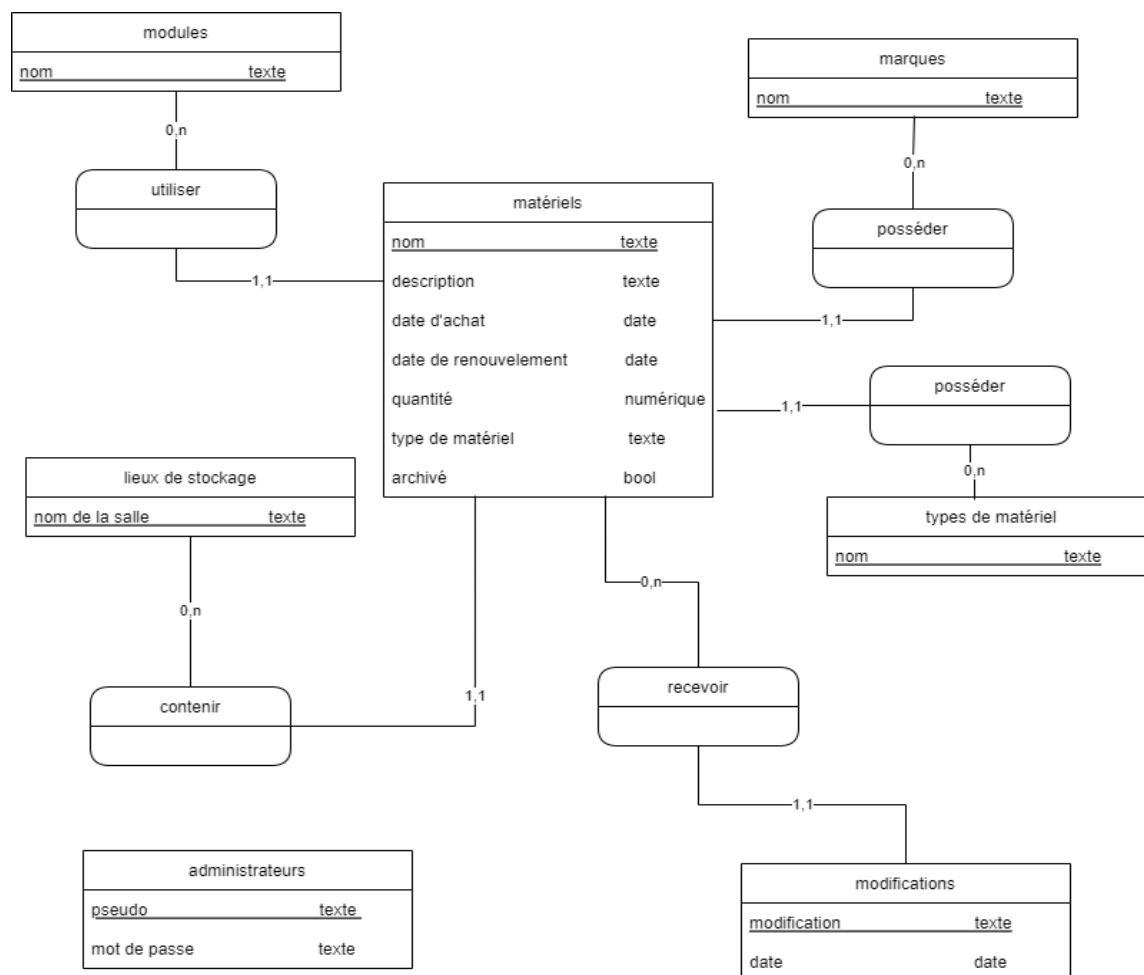
TABLE DES ILLUSTRATION

Méthode de gestion, en cascade1	3
Accueil 2	5
login 3.....	5
Ajouter/modifier 4	6
Modifier les listes 5	6
Accueil admin 6.....	7
Liste matériel admin 7.....	7
Liste matériel utilisateur 8	8
Recherche matériel 9	8
Détail matériel 10.....	9
Scénario partie 1/2 11.....	10
Scénario partie 2/2 12.....	11
planification 13	12
MLD 14	13
Dossier de réalisation 15.....	14
Diagramme de classe 16	15
CheckAdminTest 17	17
CheckAdminFalseTest 18	17
InsertDeleteWordGetListTest 1/2 19	17
InsertDeleteWordGetListTest 2/2 20	18
GetId 21.....	18
CHeckMaterialInsertAndGetMaterial 22	18
CheckMaterialUpdate 23	19
InsertMessageAndGetMessage 24	19
Journal de travail 1/3 25	25
Journal de travail 2/3 26	26
Journal de travail 3/3 27	27
Figure 28	28
Accueil manuel 29.....	28
Accueil admin manuel 30.....	29
Ajouter/modifier manuel 31	29
Liste matériel admin manuel 32.....	30
Modifier manuel 33	31
Modifier liste manuel 34.....	31
Ajouter manuel 35	32

Liste matériel utilisateur 36	33
Détail matériel 37.....	34
MCD 38.....	37
CDC 1/3 39	38
CDC 2/3 40	39
CDC 3/3 41	39

ARCHIVES DU PROJET

MCD



Projet : TPI
 Titre : GestionInventaireInformatique
 Auteur : Tristan Barras
 Version : 2.0 du 15.05.2023

CAHIER DES CHARGES

CPNV

Filière informatique

Examen - TPI

1 INFORMATIONS GENERALES

Candidat :	Nom : BARRAS	Prénom : TRISTAN																				
	✉ : tristan.barras@cpnv.ch	☎ : 077 468 56 86																				
Lieu de travail :	<input checked="" type="checkbox"/> CPNV, Rue de la Gare 14, 1450 Sainte-Croix																					
Orientation :	<input type="checkbox"/> 88601 Développement d'application <input checked="" type="checkbox"/> 88602 Informatique d'entreprise <input type="checkbox"/> 88603 Technique des systèmes																					
Chef de projet :	Nom : ANDOLFATTO	Prénom : Frédérique																				
	✉ : frederique.andolfatto@cpnv.ch	☎ : 077 206 66 45																				
Expert 1 :	Nom : Ruchat	Prénom : Laurent																				
	✉ : laurent.ruchat@bluewin.ch	☎ : 079 203 85 42																				
Expert 2 :	Nom : Berney	Prénom : Daniel																				
	✉ : daniel.berney@heig-vd.ch	☎ : 079 209 87 93																				
Période de réalisation :	Du mardi 2 mai 2023 à 8h00 au mardi 30 mai 2023 à 16h50																					
Horaires de travail :	<table border="0"> <tr> <td>Lundi</td> <td>09h50-12h15</td> <td>13h30-16h55</td> <td>Pentecôte 29 mai</td> </tr> <tr> <td>Mardi</td> <td>08h00-12h15</td> <td>13h30-16h55</td> <td></td> </tr> <tr> <td>Mercredi</td> <td>08h00-12h15</td> <td>-</td> <td></td> </tr> <tr> <td>Jeudi</td> <td>08h00-12h15</td> <td>13h30-16h55</td> <td>Ascension 18 mai</td> </tr> <tr> <td>Vendredi</td> <td>-</td> <td>-</td> <td>Pont de l'Ascension 19 mai</td> </tr> </table> <p>Toutes les demi-journées ont une pause obligatoire de 15 minutes, sauf si elles commencent à 09h50.</p>		Lundi	09h50-12h15	13h30-16h55	Pentecôte 29 mai	Mardi	08h00-12h15	13h30-16h55		Mercredi	08h00-12h15	-		Jeudi	08h00-12h15	13h30-16h55	Ascension 18 mai	Vendredi	-	-	Pont de l'Ascension 19 mai
Lundi	09h50-12h15	13h30-16h55	Pentecôte 29 mai																			
Mardi	08h00-12h15	13h30-16h55																				
Mercredi	08h00-12h15	-																				
Jeudi	08h00-12h15	13h30-16h55	Ascension 18 mai																			
Vendredi	-	-	Pont de l'Ascension 19 mai																			
Nombre d'heures :	90 heures																					
Planning (en H ou %)	Analyse : 15h, Implémentation : 40h, Tests : 18h, Documentation : 17h																					
Présentation :	Dates retenues : 12 ou 13 juin 2023																					

2 PROCÉDURE

Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1^{er} jour.

Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.

Le candidat a connaissance de la feuille d'appréciation avant de débiter le travail.

Le candidat est entièrement responsable de la sécurité de ses données.

En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son CdP.

Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier.

A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L'un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.

3 TITRE

Création d'une application en C# de gestion d'inventaire du matériel informatique de l'école

4 MATÉRIEL ET LOGICIEL À DISPOSITION

1 ordinateur en configuration standard CPNV avec accès à internet (windows 10)
Environnement de développement Microsoft Visual Studio
Outil de modélisation de base de données
Base de données
Outil de gestion de versions tel git
Liste non exhaustive pouvant dépendre des choix techniques effectués et de l'expérience du candidat

5 PRÉREQUIS

Compétences en développement C#
Compétences en modélisation et gestion de base de données

6 DESCRIPTIF DU PROJET

Le candidat devra dans un premier temps procéder à l'analyse et la conception des différents modules de l'application. Cette analyse comprendra la modélisation de la base de données ainsi qu'une maquette de l'application. Ensuite, il commencera la réalisation et effectuera les tests nécessaires.

L'application doit couvrir les cas suivants :

- Partie « administrateur » (avec authentification) :
 - o Insertion des informations du matériel :
L'administrateur, par l'intermédiaire d'une interface graphique qu'il devra créer, pourra enregistrer les informations sur le matériel. Les données à entrer manuellement dans l'application, si elles sont pertinentes, sont les suivantes : nom du matériel, la description, la date d'achat, la marque, le module dans lequel il est utilisé, le lieu du stockage, la date de renouvellement, la quantité, le type de matériel (routeur, beamers, utilitaires, câbles, NAS....). Toute autre information utile pourra être ajoutée par le candidat.
 - o Historisation du matériel :
L'administrateur aura la possibilité de modifier les informations sur le matériel et d'archiver le matériel
 - o Aide à l'insertion des données :
Une interface graphique permettra d'insérer manuellement dans la base de données la liste des marques utilisées pour le matériel, la liste des modules ainsi que l'emplacement où le trouver. Ces informations seront utilisées lors de l'insertion des informations sur le matériel. A l'aide de l'interface graphique mentionnée précédemment, la liste de ces données (marques du matériel, modules et emplacement) sera saisie manuellement par le candidat qui s'inspirera du matériel utilisé lors de sa formation ou des modules qu'il a suivis lors de sa formation pour les trouver.
- Partie « utilisateur » :
 - o Consultation du matériel :
Un utilisateur pourra consulter les informations sur le matériel référencé. Une interface sera créée et permettra de rechercher, en fonction de critères, le matériel. Chaque information concernant le matériel (nom, description, date d'achat, marque,

module, lieu de stockage, date de renouvellement, quantité, type de matériel...) pourra être un critère de recherche. Un critère au moins est obligatoire pour pouvoir effectuer une recherche. Une liste contenant les informations principales sur le matériel correspondant aux critères de recherche s'affichera. Le candidat définira lors de son analyse les données principales à afficher lors du résultat de sa recherche. L'utilisateur, en cliquant sur un élément de cette liste, aura la possibilité d'accéder à toutes les informations sur le matériel ainsi que l'ensemble des modifications qui ont été effectuées sur celui-ci, y compris la date à laquelle la modification a eu lieu.

Le candidat sera libre de choisir sa méthode de gestion de projet ainsi que le système de gestion de base de données. Il devra justifier ses choix.

Le candidat devra écrire une classe de tests automatiques sur une classe au choix de son projet.

7 LIVRABLES

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

- Une planification initiale sous forme électronique au format PDF le 02 mai 2023.
- Un rapport de projet sous forme électronique au format PDF deux fois par semaine, le mardi soir et le jeudi soir
- Un journal de travail sous forme électronique au format PDF deux fois par semaine, le mardi soir et le jeudi soir
- A la fin du TPI, son rapport de projet final et son journal de travail sous forme électronique au format PDF
- A la fin du TPI, une archive contenant :
 - Un script de création de la base de données
 - Un dossier contenant le code de l'application complète
 - Une procédure installation de l'application
 - Un exécutable de l'application

8 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20) :

1. Choix, modélisation et intégration de la base de données
2. Qualité et lisibilité du code source, respect des conventions de nommage standards et du CPNV
3. Architecture du code source, séparation IHM et code métier
4. L'insertion réussie des informations sur le matériel
5. La recherche réussie du matériel selon les critères demandés
6. L'archivage réussi des informations sur le matériel lorsqu'il ne fonctionne plus ou lorsqu'il est remplacé
7. La complétude de la classe de tests par rapport à la classe productive