

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 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
Description des tests effectués.....	15
Erreurs restantes.....	15
Liste des documents fournis	17
Conclusions	17
Annexes.....	19
Résumé du rapport du TPI / version succincte de la documentation	19
Sources – Bibliographie.....	19
Journal de travail.....	19
Manuel d'Installation	19

Manuel d'Utilisation..... 19

Archives du projet 19

:

ANALYSE PRÉLIMINAIRE

INTRODUCTION

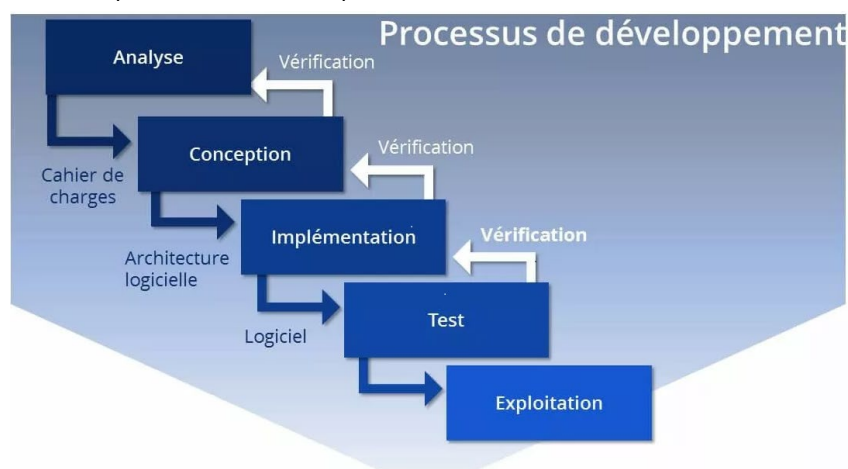
Ce projet est réalisé dans le cadre d'un TPI, il y a 90 heures de travail. Le projet est une application en C# de gestion d'inventaire de matériel informatique de l'école. Elle permettra de parcourir, ajouter, modifier du matériel informatique présent dans une base de données. Pour ce projet, j'ai des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



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 :

Window Name

partie utilisateur

partie administrateur

Window Name

nom d'utilisateur

mot de passe

se connecter

accueil


Window Name


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


Window Name

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

Window Name

 insérer du matériel

 modifier le matériel

 modifier les listes marques, modules et lieux d'emplacement

se déconnecter

Window Name

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

Window Name

RECHERCHER

nom

description

date d'achat

marque ▼

module ▼

lieu de stockage ▼

date de renouvellement

quantité

type

Window Name

Name	date d'achat	type de matériel	quantité
SC02-20	02.03.2022	routeur	1
ordi2.000	01.01.1999	pc	1
SRV02-203-2432	11.10.2000	câble	18
RT-234-34535	31.02.2015	switch	4

i-CQ VD Octobre 2010
02.12.2010

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 texte box.	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 insert 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.

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.

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	84h40(pause de 5min pas prise en compte)							
		SEM	1	2	3	4	5	
1 Analyse								15 h 15
	11 Planification initial	Prévu	1 h 30					1 h 30
		temps réel						
	12 MCD/MLD	Prévu	3 h 45					3 h 45
		temps réel						
	13 Elaboration des stratégies de test	Prévu	1 h 30					1 h 30
		temps réel						
	14 Documentation Initiale	Prévu	8 h 00					8 h 00
		temps réel						
	15 création de la base de données	Prévu	0 h 30					0 h 30
		temps réel						
2 IHM								3 h 00
	21 création des pages de base partie admin/utilisateur	Prévu	1 h 00					1 h 00
		temps réel						
	22 mise en place de tous les boutons/texte	Prévu	1 h 00	0 h 30				1 h 30
		temps réel						
	23 contrôle des normes de nommage CPNV	Prévu		0 h 30				0 h 30
		temps réel						
3 code métier								37 h 15
	31 connexion à la base de données	Prévu		1 h 00				1 h 00
		temps réel						
	32 mise en place des classes et fonction de base pour celle-ci	Prévu		2 h 00				2 h 00
		temps réel						
	33 aide à l'insertion de données	Prévu		10 h 00				10 h 00
		temps réel						
	34 insertion des informations du matériel	Prévu		6 h 30				6 h 30
		temps réel						
	35 historisation du matériel	Prévu			6 h 00			6 h 00
		temps réel						
	36 consultation du matériel avec critère	Prévu			7 h 45	4 h 00		11 h 45
		temps réel						
4 Tests								14 h 30
	41 Tests automatique d'une classe	Prévu				6 h 00		6 h 00
		temps réel						
	42 Tests scénario/ tests unitaires	Prévu				4 h 30		4 h 30
		temps réel						
	43 Correction de bugs	Prévu				4 h 00		4 h 00
		temps réel						
5 Documentation final								10 h 40
	51 Rapport de projet	Prévu		2h00	2h00	3 h 00		3 h 00
		temps réel						
	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						

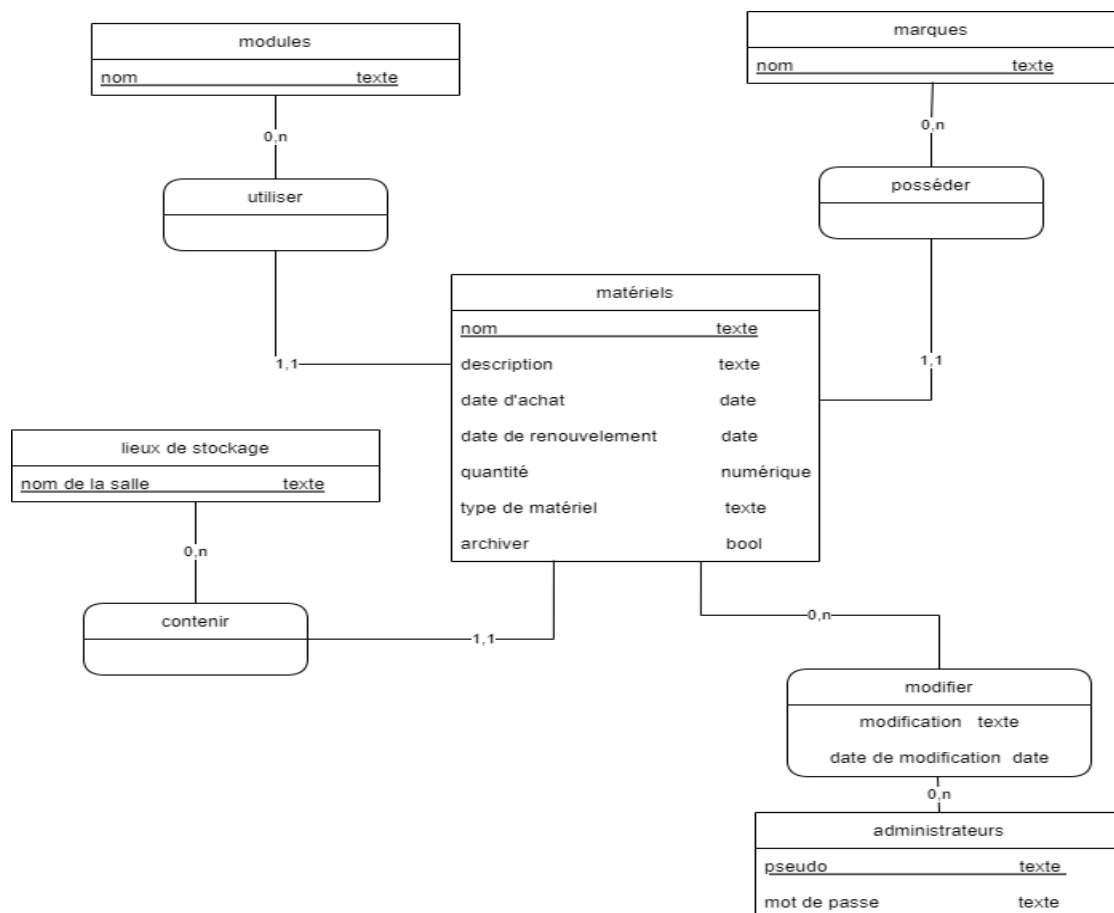
DOSSIER DE CONCEPTION

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

Le système d'exploitation sera Windows 10.

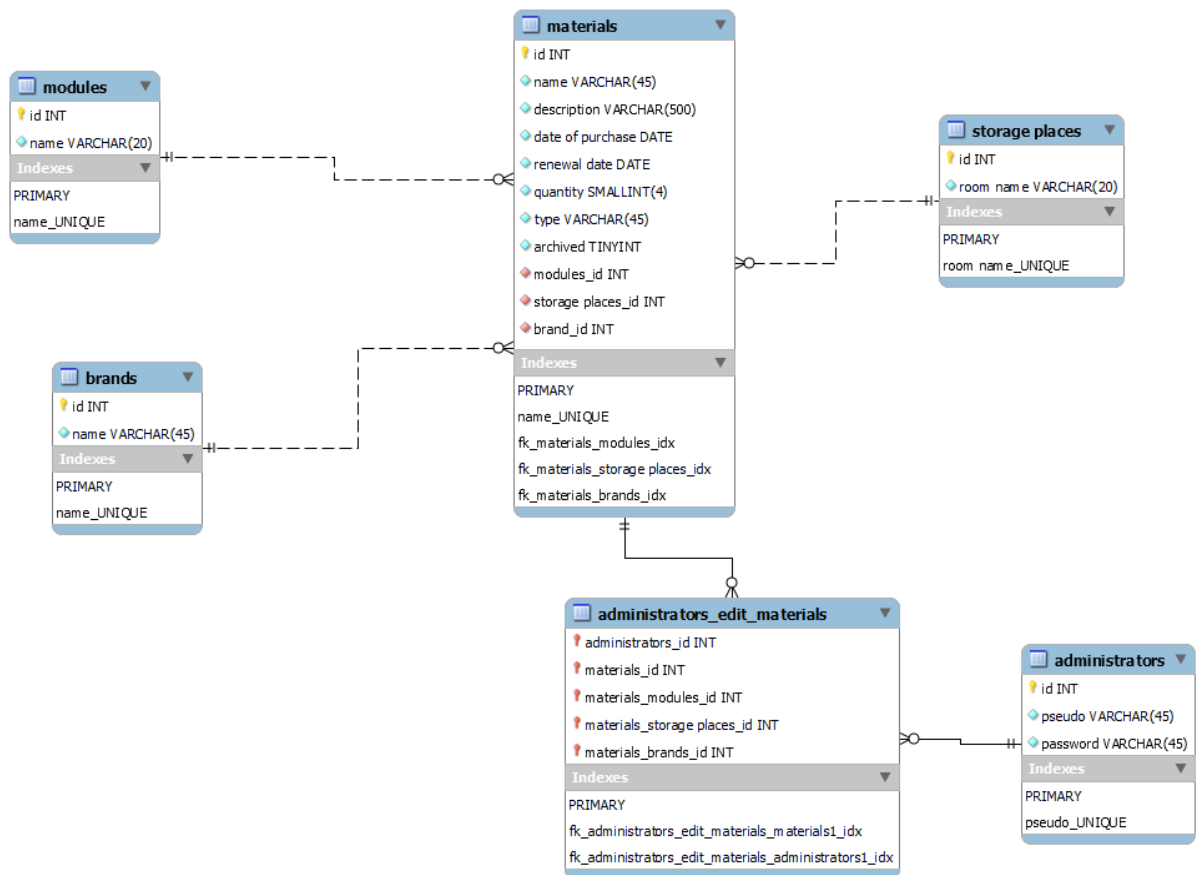
L'outil principale utilisé sera Microsoft Visual studio pour coder en C# ainsi que les applications de réalisation comme balsamiq, Excel, draw.io, MySQL Workbench, mariadb, HeidiSQL. Un outil de gestion de version git.

Voici le MCD de ce projet :



Projet : TPI
Titre : GestionInventaireInformatique
Auteur : Tristan Barras
Version : 1.0 du 02.05.2023

Voici Le MLD de ce logiciel



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...

DESCRIPTION DES TESTS EFFECTUÉS

numéro du test	description	date du test	résultat attendu	résultat
1	Test unitaire automatique <pre> [Test] 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>	08.05.23	La fonction CheckAdmin retourne true si un admin similaire est trouvé dans la bdd	réussit
2	Test unitaire automatique <pre> [Test] 0 references 0 changes 0 authors, 0 changes public void CheckAdminFlaseTest() { //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>	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 GetListTest() { //Arrange ConnectionDB bdd = new ConnectionDB(); List<string> listexpected = new List<string>(); listexpected.Add("SC-C111"); listexpected.Add("SC-C236"); List<string> list = new List<string>(); //Act list = bdd.GetList("storageplaces"); //Assert Assert.AreEqual(listexpected, list); } </pre>	08.05.23	La fonction retourne la liste « storageplaces » disponible dans la bdd actuellement elle contient « SC-C111 » et « SC-C236 ».	réussit
4	<pre> public void InsertDeleteWordTest() { //Add and Delete are together to not leave stuff //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); //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>	08.05.23	La fonction InsertWord va ajouter un mot dans une liste de la bdd	réussit
5	<pre> [Test] 0 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>	08.05.23	La fonction GetId va retourner l'id en int d'un élément d'une liste	réussit

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)*

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)

JOURNAL DE TRAVAIL

Date	Durée	Activité	Remarques

MANUEL D'INSTALLATION

MANUEL D'UTILISATION

ARCHIVES DU PROJET

<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>. 03.05.2023

<https://stackoverflow.com/questions/3061042/how-do-i-set-combobox-read-only-or-user-cannot-write-in-a-combo-box-only-can-sel> 08.05.2023

<https://www.codeproject.com/Questions/646695/how-to-reload-form-or-refresh-form-in-csharp-net> 08.05.2023

<https://stackoverflow.com/questions/29814305/how-to-load-list-data-into-datagridview-in-c-sharp> 09.05.2023

<https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridview.rowheadermousedoubleclick?view=windowsdesktop-7.0&viewFallbackFrom=net-5.0> 09.05.2023

<https://learn.microsoft.com/en-us/answers/questions/372644/pass-from-data-grid-to-textbox-in-c> 09.05.2023

messageInput <https://stackoverflow.com/questions/10797774/messagebox-with-input-field> 09.05.2023