

01/02/2021

# Location de matériel

Gestion de Stocks

Paola Costa

CENTRE PROFESSIONNEL DU NORD VAUDOIS

## Table des matières

1	Analyse préliminaire .....	3
1.1	Introduction.....	3
1.2	Objectifs.....	3
1.3	Planification initiale .....	4
2	Analyse / Conception.....	5
2.1	Concept .....	5
2.1.1	Fonctionnalités.....	5
2.1.2	Modèles de données .....	5
2.1.2.1	Modèle de données conceptuel .....	5
2.1.2.2	Modèle de données logique.....	6
2.2	Maquettes.....	7
2.3	Use Cases & Scénarii.....	8
2.3.1	Use Cases .....	8
2.3.2	Scénarii.....	9
2.3.2.1	Ajouter un objet.....	9
2.3.2.2	Afficher les objets dont le délai est dépassé .....	9
2.3.2.3	Générer un PDF .....	9
2.3.2.4	Louer un objet.....	10
2.3.2.5	Retourner un objet loué .....	11
2.3.2.6	Rechercher par mot-clé .....	11
2.4	Diagrammes .....	12
2.4.1	Diagramme de classe .....	12
2.4.2	Diagramme de flux.....	13
2.5	Stratégie de test.....	14
2.6	Risques techniques .....	14
2.7	Planification définitive .....	15
2.8	Infrastructure .....	16
2.8.1	Matériel hardware et système d'exploitation .....	16
2.8.2	Outils logiciels .....	16
3	Réalisation.....	17
3.1	Dossier de réalisation .....	17
3.1.1	Répertoires et fichiers du projet .....	17
3.1.1.1	Répartition physique des fichiers .....	17
3.1.1.2	Fichiers et description .....	17
3.1.2	Produit fini.....	18
3.2	Liste des documents fournis .....	18
3.2.1	Programmation et scripts .....	18
3.3	Description des tests effectués .....	19
3.3.1	Tests unitaires .....	19
3.3.2	Tests fonctionnels.....	19
3.3.2.1	Ajouter un objet.....	20
3.3.2.2	Louer un objet.....	20
3.3.2.3	Retourner un objet .....	21
3.3.2.4	Délai dépassé .....	21

3.3.2.5 Rechercher .....	21
3.3.2.6 Générer le PDF.....	21
3.3.2.7 Historique de location .....	21
3.4 Problèmes rencontrés et résolution .....	22
3.5 Erreurs restantes .....	23
<b>4 Conclusions .....</b>	<b>25</b>
4.1 Atteinte des objectifs.....	25
4.2 Maintien des délais .....	25
4.3 Points positifs et négatifs .....	26
4.4 Difficultés particulières.....	26
4.5 Évolutions et améliorations.....	26
<b>5 Annexes.....</b>	<b>27</b>
5.1 Sources – Bibliographie .....	27
5.1.1 Pages internet consultées .....	27
5.1.2 Personnes consultées .....	27
5.2 Journal de travail .....	28
5.3 Mode d'emploi utilisateur .....	32
5.4 Manuel d'installation .....	33
5.5 Archives du projet.....	33

## Table des illustrations

Figure 1 : modèle conceptuel de données.....	5
Figure 2 : modèle logique de données .....	6
Figure 3 : maquette – vision globale de l'application .....	7
Figure 4 : maquette – formulaire d'ajout d'objet.....	7
Figure 5 : use cases .....	8
Figure 6 : diagramme de classe .....	12
Figure 7 : diagramme de flux – ajouter un objet .....	13
Figure 8 : diagramme de flux – retourner un objet.....	13
Figure 9 : tests unitaires de la classe "Objects" .....	19
Figure 10 : test "Ajouter un objet" avec des données valides .....	20
Figure 11 : test "Louer un objet" avec des données valides .....	20
Figure 12 : test "Retourner un objet avec des données valides.....	21
Figure 13 : visuel global de l'application .....	32
Figure 14 : ajout d'un objet .....	32
Figure 15 : location d'un objet.....	32
Figure 16 : retourner un objet .....	32
Figure 17 : sélectionner le type de PDF .....	32

# **1 Analyse préliminaire**

## **1.1 Introduction**

Avec les avancées actuelles de la technologie, il n'a jamais été aussi simple de gérer des stocks d'une entreprise. Cependant, tout le monde n'est pas encore équipé pareillement. Dans le but de préparer mon TPI, je vais donc réaliser une application de gestion de stocks pour mon pré-TPI. Celle-ci est créée dans le cadre d'une PME, spécialisée dans la location de matériel d'extérieur. Elle permettra à une personne, même novice en informatique, d'insérer ou sortir des pièces du stock (selon les arrivages ou les locations), de créer un PDF de l'inventaire d'un simple clic ainsi que de rechercher de plusieurs manières dans les données. L'intégralité des données propres à l'application seront stockées dans une base de données.

Ce projet servira tout d'abord à reprendre en main des technologies qui ont été abordées précédemment dans différents modules. Il s'agit ici du C# et du MySQL. Les mêmes technologies serviront à réaliser le TPI. Ensuite, le fait de compléter un projet individuel dans des conditions semblables à celles du TPI permet également de se projeter et de se rendre compte de comment le projet final (TPI) va se dérouler.

Enfin, aucun travail n'a été effectué eu préalable pour préparer le pré-TPI. L'intégralité du développement se fera lors du module et du temps mis à disposition.

## **1.2 Objectifs**

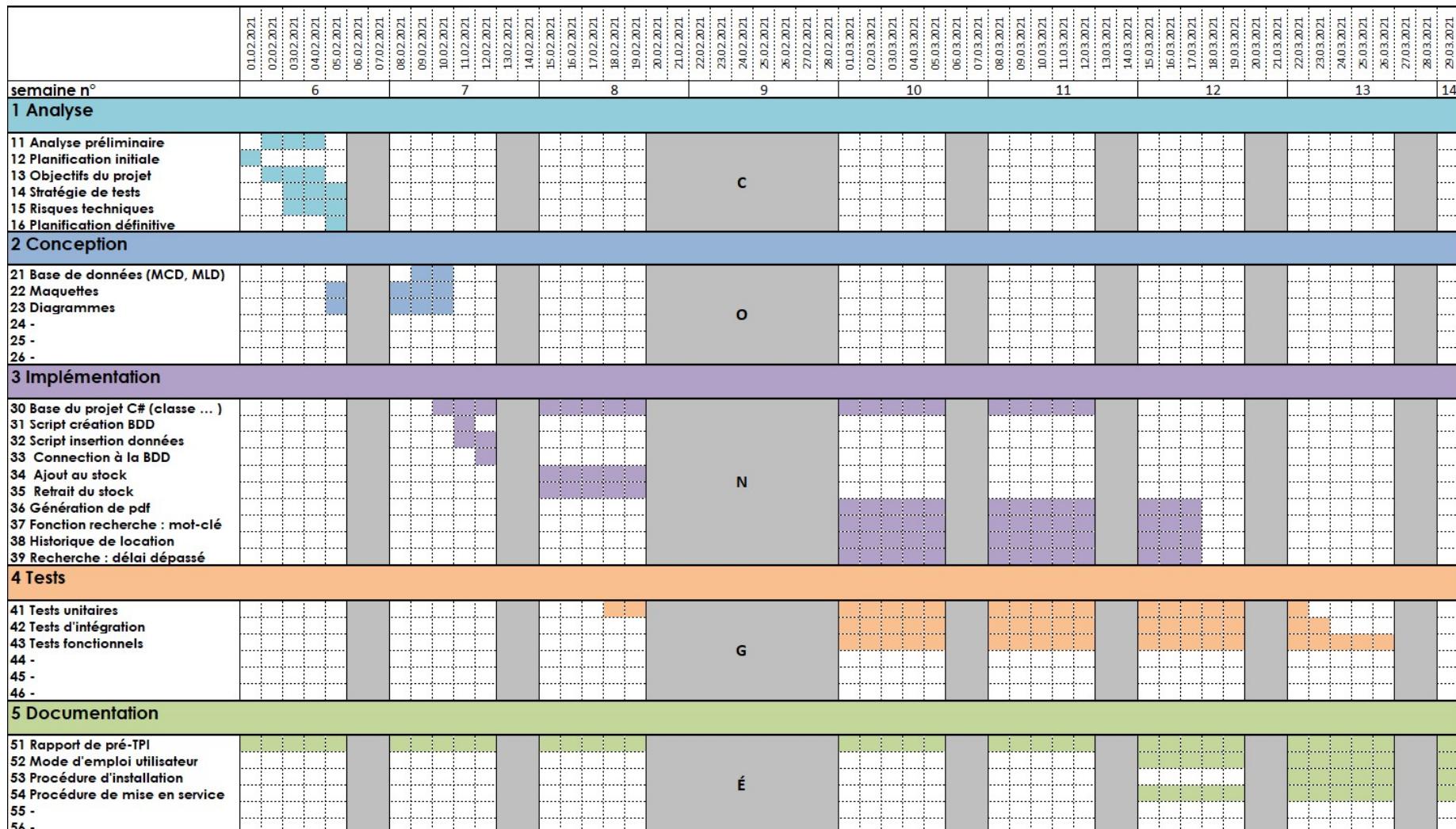
Afin de mener à bien ce projet, de nombreux objectifs sont à compléter. La validation de ceux-ci permettra de déterminer le degré de compléction du projet. L'élément principal de ce projet consiste à créer une application « clé en main ». Cela signifie que l'application sera fonctionnelle sans investissements ultérieurs. De plus, l'application doit être accessible à des novices en informatique. Un mode d'emploi sera donc mis à disposition, afin de faciliter l'utilisation.

De plus, l'application contiendra plusieurs fonctionnalités. Il s'agit de celles qui sont citées ci-dessous :

- L'application devra permettre d'ajouter des objets aux stocks.
- L'application devra permettre de louer et retourner une location.
- Il sera possible d'effectuer une recherche sur l'application.
- Les données de l'application seront stockées dans une base de données distante.
- L'application permettra de générer des rapports au format PDF, afin de réaliser des inventaires.
- L'application permettra d'effectuer un tri dans les données.
- L'application devra permettre d'afficher l'historique d'une pièce spécifique.

Enfin, afin de faciliter l'accessibilité à l'application, l'application sera accompagnée d'une procédure d'installation et de mise en service.

### **1.3 Planification initiale**



Comme le cadre de travail est clairement défini dès le départ, le projet se déroulera sur un mode cascade. La répartition des tâches est faite dès le départ et une comparaison des diagrammes de Gant sera réalisée à la fin du projet.

## 2 Analyse / Conception

### 2.1 Concept

#### 2.1.1 Fonctionnalités

En plus des différents éléments abordés dans le point [1.2 Objectifs](#), plusieurs points techniques spécifiques doivent être respectés. Ils sont explicités ci-dessous :

- Les données internes à l'application sont stockées dans une base de données.
- Au lancement de l'application, une connexion à une base de données distante est établie.
- Chaque pièce présente dans le stock est unique. Cela s'exprime grâce à son ID, qui est un code numérique unique.
- Il est possible d'ajouter une pièce dans le stock grâce à son ID. Celui-ci peut être déjà existant ou pas. S'il est nouveau, les informations propres à la pièce sont requises avant l'ajout.
- Il est possible de sortir une pièce du stock grâce à son ID. Il faut la durée de la location et les informations sur le loueur pour que cela s'effectue.
- Il est possible de générer deux types de rapports PDF différents, afin de réaliser un inventaire. Le premier regroupe le contenu actuellement en stock, le deuxième le matériel loué.
- Une fonction de recherche par mot-clé est disponible sur l'application.
- Une fonction de tri permet de voir les pièces dont la date de retour de location est dépassée.
- Il est possible de consulter l'historique de location d'une pièce.

#### 2.1.2 Modèles de données

##### 2.1.2.1 Modèle de données conceptuel

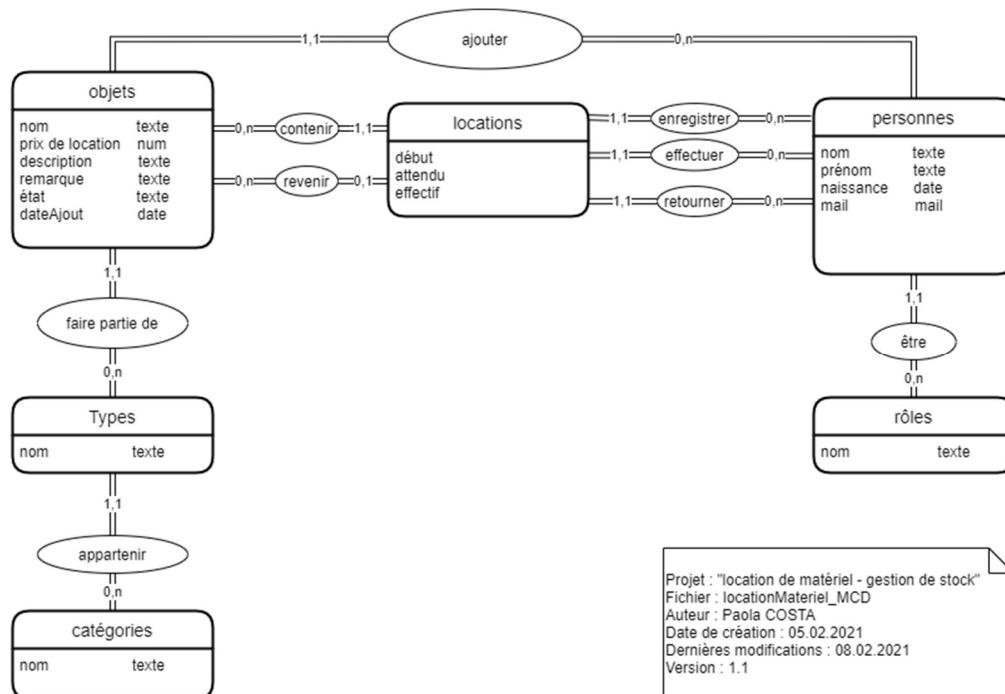


Figure 1 : modèle conceptuel de données

### 2.1.2.2 Modèle de données logique

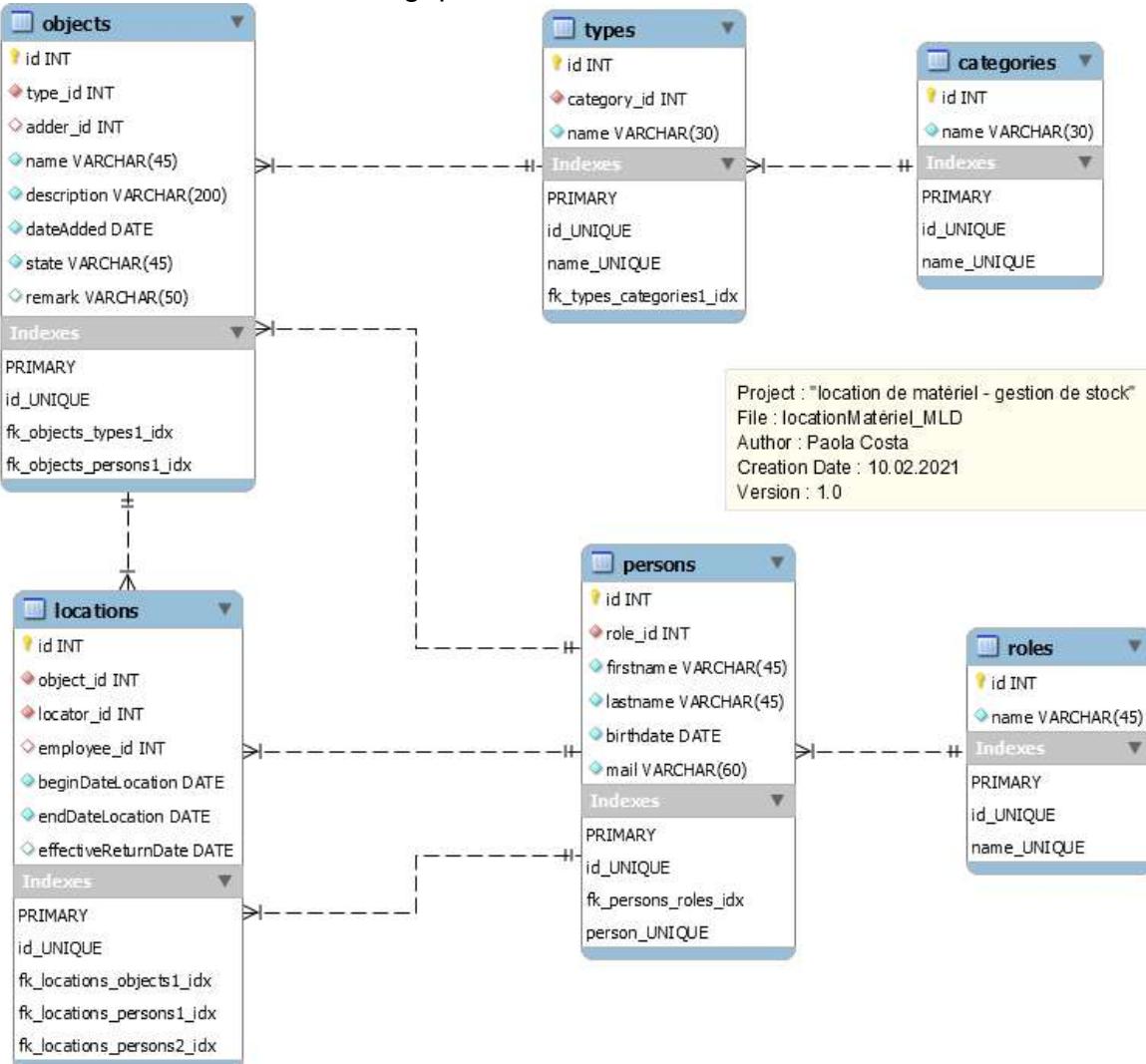


Figure 2 : modèle logique de données

Dans la table « locations », il y a trois clés étrangères :

- object\_id représente l'objet qui est loué
- locator\_id représente la personne qui loue l'objet
- employee\_id représente la personne de l'entreprise qui enregistre la location.  
Cette information est présente afin de pouvoir créer des logs plus complets.

La table « roles » a pour but de classifier les personnes en trois groupes : utilisateur, administrateur et client. Les clients n'ont pas accès à l'application. Les administrateurs gèrent les utilisateurs.

La fonctionnalité de gestion n'est pas encore mise en place, la présence de cette table a pour but de permettre une évolution future de l'application.

Dans la table « objects », la colonne adder\_id sert à identifier la personne qui a ajouté l'objet dans le stock. Encore une fois, il s'agit de pouvoir créer un log complet.

La raison pour laquelle adder\_id (table « objects ») et employee\_id (table « locations ») peut être nul est pour permettre la suppression d'un employé (rôle utilisateur) sans être impacté parce qu'il aurait procédé à une location non finie. Cela aura pour conséquence d'afficher « NULL » à l'emplacement de son nom si on récupère les logs.

## 2.2 Maquettes

L'intégralité des maquettes se rapportant à l'application seront présentées ici. Celles-ci permettent de visualiser l'apparence du produit fini.

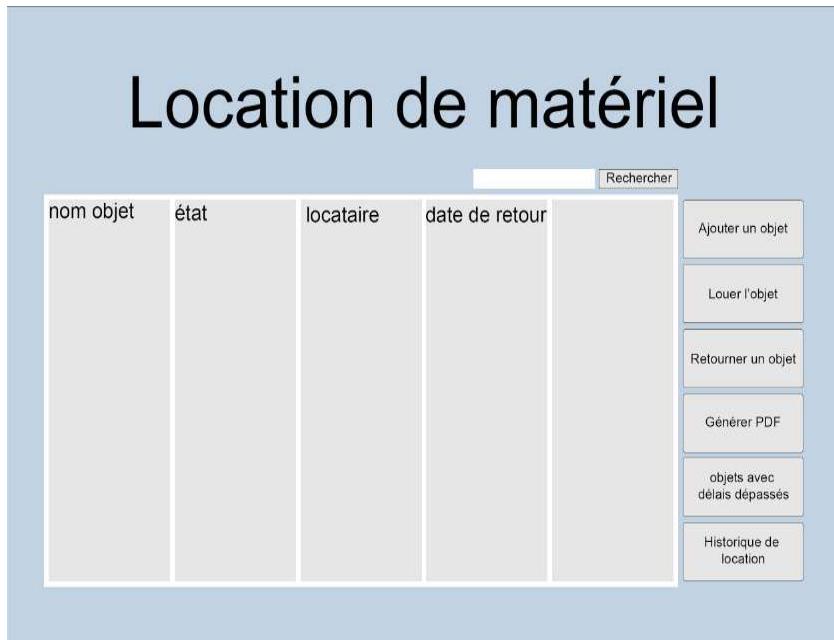


Figure 3 : maquette – vision globale de l'application

Cette maquette représente le visuel global de l'application, avec l'intégralité des fonctionnalités.

Les boutons à droite servent à effectuer les actions sur l'application. Celles-ci sont variables.

Certains des boutons permettent d'ouvrir des formulaires à remplir, afin de renseigner des détails supplémentaires.

Figure 4 est un formulaire intitulé 'Ajouter un objet au stock'. Il contient les champs suivants : 'nom de l'objet' (placeholder), 'nombre' (placeholder), 'type' (placeholder), 'description' (placeholder), 'remarque' (placeholder). En bas, il y a deux boutons : 'Annuler' et 'Ajouter'.

Figure 4 : maquette – formulaire d'ajout d'objet

Cette maquette montre l'apparence d'un formulaire. Il s'agit ici de celui d'ajout d'un objet au stock.

Les textes en gris clair représentent des placeholders. Ils sont visibles tant que le curseur n'est pas dans le champ texte.

Tout le projet va fonctionner sur le principe de ces deux maquettes. Dès qu'un formulaire doit être utilisé, le visuel sera semblable à celui de la Figure 2. Un pop-up de confirmation valide chaque clic sur un bouton, la sélection par défaut de la confirmation est « Valider ».

## 2.3 Use Cases & Scénarii

### 2.3.1 Use Cases

Comme montré par le schéma, l'intégralité des actions se font en tant qu'employé. Ce diagramme permet de montrer les cas d'utilisation de l'application.

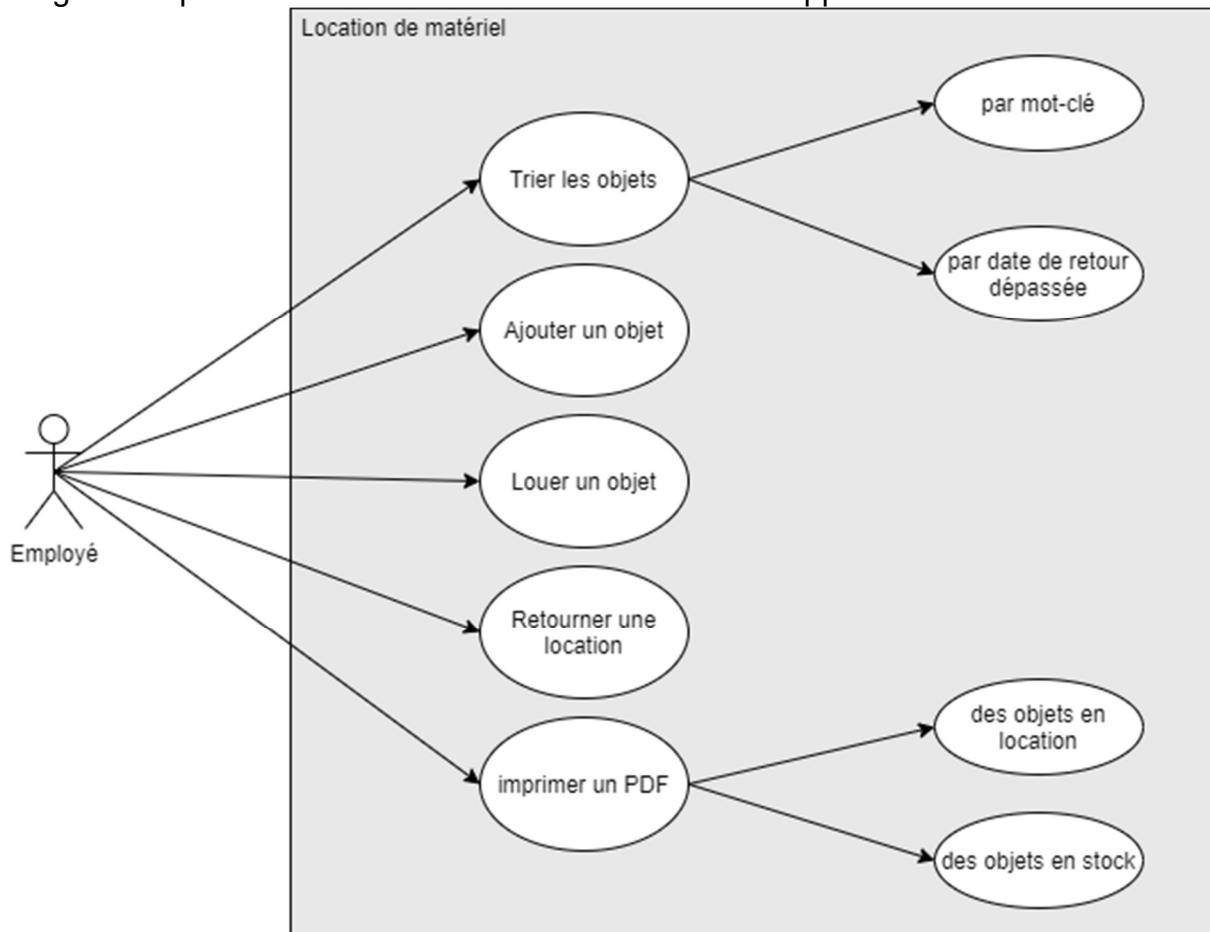


Figure 5 : use cases

### 2.3.2 Scénarios

#### 2.3.2.1 Ajouter un objet

Action	Condition	Réaction
L'employé clique sur « ajouter un objet »		
L'employé donne un nom générique à l'objet (clé à molette)	Le nom n'existe pas du tout	Un formulaire s'ouvre, il faut remplir les spécificités de l'objet
	Le nom existe déjà (clé à molette 2)	Le système demande une confirmation qu'il s'agit du même objet
Validation du pop-up	il s'agit du même objet	L'objet est ajouté avec un nouveau numéro aux objets en stock
	Il ne s'agit pas du même objet	Un formulaire s'ouvre, il faut remplir les spécificités de l'objet
L'employé entre les spécificités et valide et ajoute l'objet	Toutes les données sont entrées	L'objet est ajouté au stock, message comme quoi ça a bien fonctionné, retour à l'accueil
	Des données sont manquantes, de type erroné	Un message d'erreur apparaît, retour à la page d'accueil de l'application

#### 2.3.2.2 Afficher les objets dont le délai est dépassé

Action	Condition	Réaction
L'employé clique sur « afficher les objets avec délai dépassé »		Seul les objets actuellement en location et dont la date de retour prévue est antérieure à la date actuelle sont affichés à l'écran

#### 2.3.2.3 Générer un PDF

Action	Condition	Réaction
L'employé clique sur « générer un PDF »		Un pop-up avec 3 options s'affiche : « annuler », « location », « en stock »
L'employé sélectionne « location »		Un PDF avec tous les objets loués est généré
L'employé sélectionne « en stock »		Un PDF avec tous les objets en stock est généré
		Retour à la page d'accueil
L'employé sélectionne « annuler »		Retour à la page d'accueil

## 2.3.2.4 Louer un objet

Action	Condition	Réaction
L'employé clique sur « louer un objet »		
L'employé tape le nom ou l'id de l'objet	Le nom ou l'id existe dans la base de données et l'objet n'est pas loué	Un pop-up de confirmation apparaît, demandant si c'est le bon objet à louer
	Le nom ou l'id sont incorrects l'objet avec ce nom est déjà loué	Un message d'erreur apparaît, retour à la page d'accueil
	Le nom ou l'id sont incorrects l'objet avec cet ID est déjà loué	
L'employé invalide la confirmation de l'objet sélectionné		Retour à la page d'accueil
L'employé valide la confirmation		Un formulaire de location s'ouvre, pour spécifier les données du locataire
L'employé remplit le champ n° de client et la date de retour de location. Le reste du formulaire est grisé	Le client existe	Un pop-up de confirmation apparaît
	Le client n'existe pas	Retour à l'accueil
L'employé remplit tout le formulaire, sauf le champ n° de client	Un client identique existe déjà	Un pop-up pour savoir si le client voulu est celui affiché apparaît
	Le client n'existe pas	Un pop-up de confirmation de la location apparaît
Le client est bien celui indiqué		Un pop-up de confirmation de la location apparaît
Le client n'est pas le même		Un message d'erreur apparaît, retour à l'accueil
Validation de la confirmation de location	Le client existait déjà	Message de réussite, retour à l'accueil
	Le client n'existant pas	Message de réussite, message d'ajout du client à la base de données, retour à l'accueil

## 2.3.2.5 Retourner un objet loué

Action	Condition	Réaction
L'employé clique sur « retourner un objet »		Un formulaire s'affiche
L'employé entre le n° ou le nom de l'objet et valide	Le nom ou l'id est valide	La liste des éléments correspondants loués apparaît sur le formulaire
	Le nom et/ou l'id sont invalides	Message d'erreur. Retour à la page d'accueil
L'employé sélectionne l'objet qu'il doit retourner et valide		Un pop-up s'affiche pour valider
L'employé valide		Message de succès et retour à la page d'accueil
L'employé annule.		Retour à la page d'accueil

## 2.3.2.6 Rechercher par mot-clé

Action	Condition	Réaction
L'employé tape du texte dans le champ de recherche et appuie sur le bouton « Rechercher »	Le champ de texte est vide	Rien ne se passe
	Le champ de texte contient du texte	Un tri s'effectue. Seules les données avec une correspondance partielle s'affichent
	Il n'y a aucune correspondance entre les données et le texte de recherche	Aucune entrée n'est présente dans le tableau
	Toutes les données correspondent à la recherche	Toutes les données s'affichent dans le tableau
L'employé tape un autre texte et appuie sur « Rechercher »		La recherche repart de zéro, la liste affichée à l'écran n'impacte pas le résultat

## 2.4 Diagrammes

### 2.4.1 Diagramme de classe

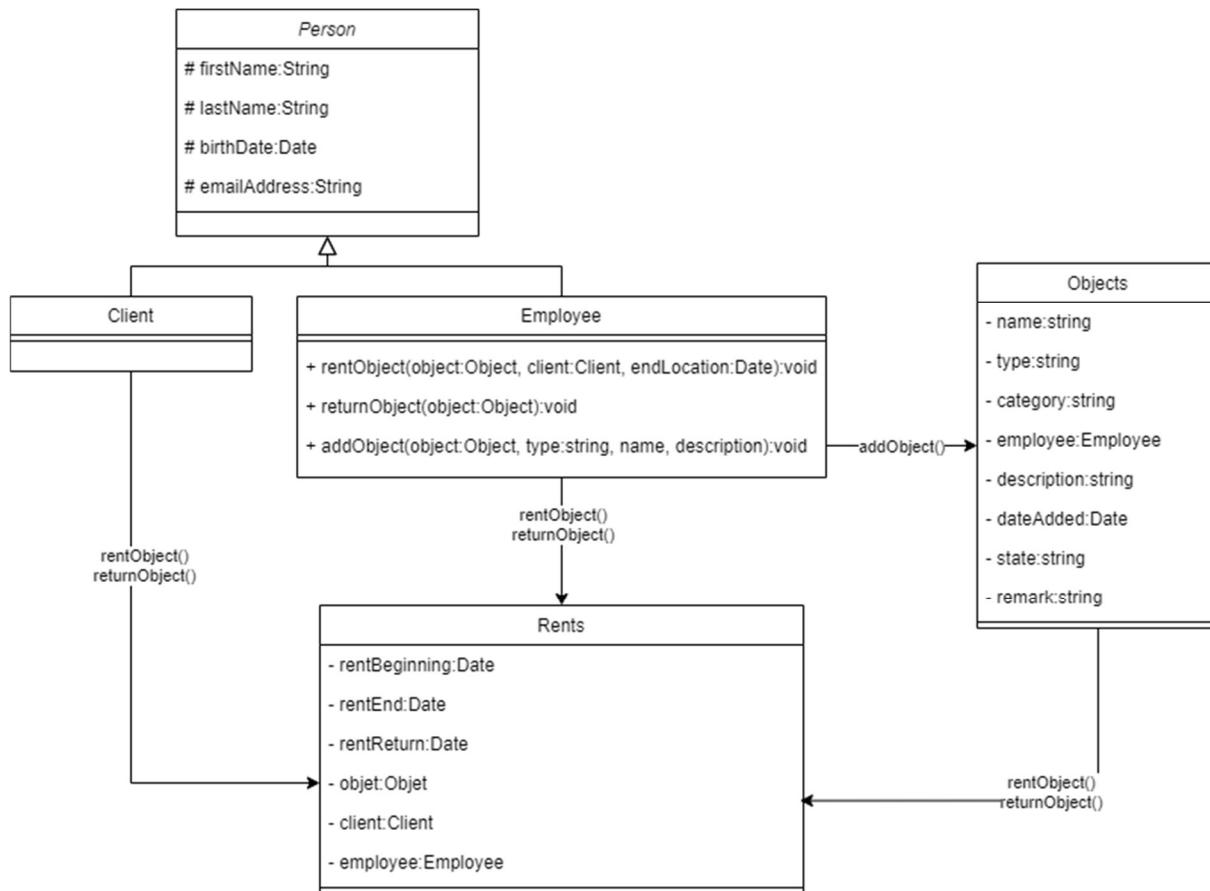


Figure 6 : diagramme de classe

### 2.4.2 Diagramme de flux

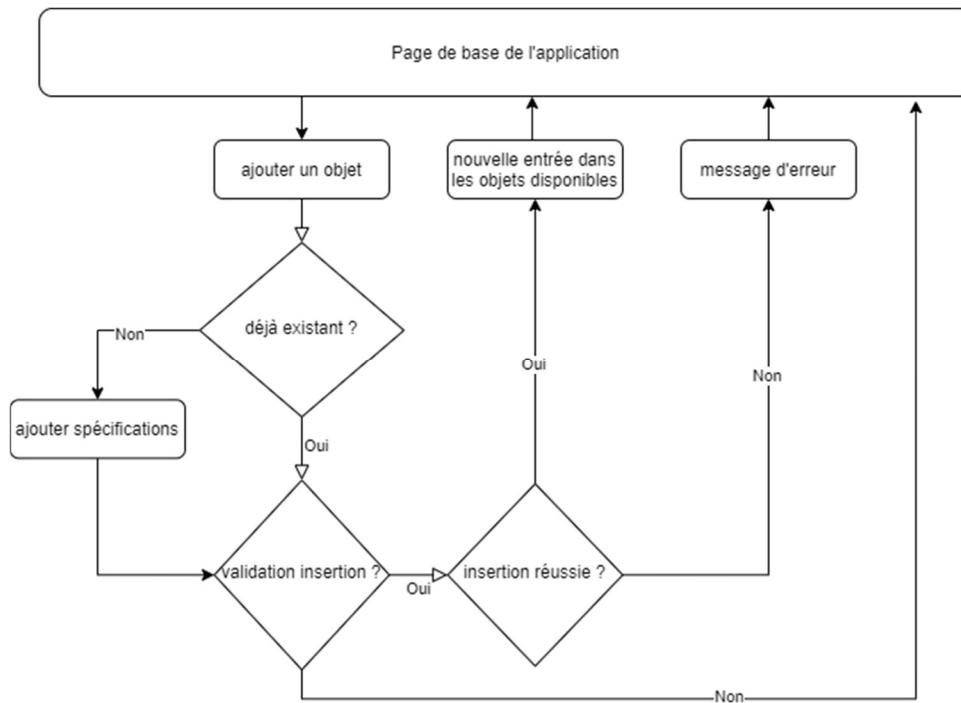


Figure 7 : diagramme de flux – ajouter un objet

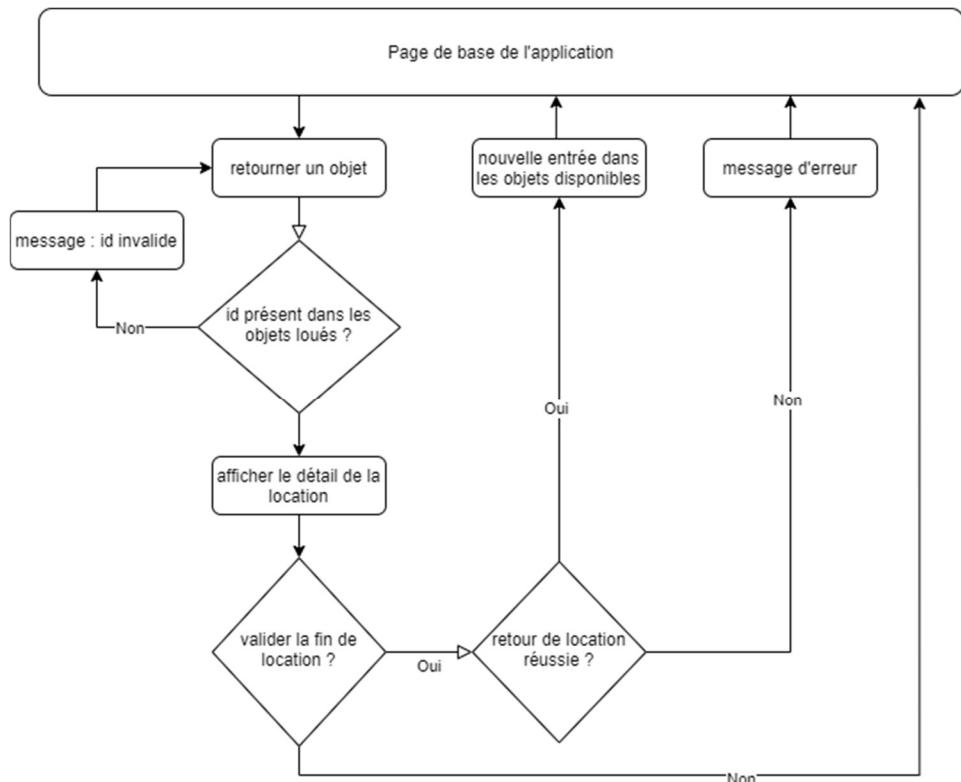


Figure 8 : diagramme de flux – retourner un objet

## 2.5 Stratégie de test

Afin de tester l'application, deux types de tests vont être effectués : les tests unitaires et les tests fonctionnels. Ils sont réalisés de manière automatique grâce à des classes de tests spécifiques sur Visual Studio 2019. Les tests d'intégration s'effectuent entre différentes unités et valide le bon fonctionnement de ces unités entre elles. Les tests fonctionnels permettent de valider que l'application fonctionne comme elle doit fonctionner et fait ce qui est attendu.

Pour pouvoir détecter les problèmes au plus vite, les tests seront réalisés tout au long du développement. Chaque fonctionnalité subira tout d'abord un test unitaire. Les tests fonctionnels consisteront à vérifier que les fonctionnalités intégrées à l'application fonctionnent effectivement et retournent ce qu'elles doivent. Ils seront effectués régulièrement à l'aide du bouton « démarrer » en mode débug sur VisualStudio.

Les tests unitaires se doivent d'être exhaustifs, afin de détecter les soucis avant d'intégrer l'élément au projet. Autant que possible, les tests fonctionnels seront eux aussi exhaustifs.

Dans le cadre des tests fonctionnels, les données à utiliser pour réaliser les tests seront les données réelles, fournies avec le script de création de la base de données. Les tests unitaires se feront directement sur Visual Studio 2019, avec des classes de tests spécifiquement créées.

La majorité des tests étant automatisée, aucun testeur externe ne sera utilisé.

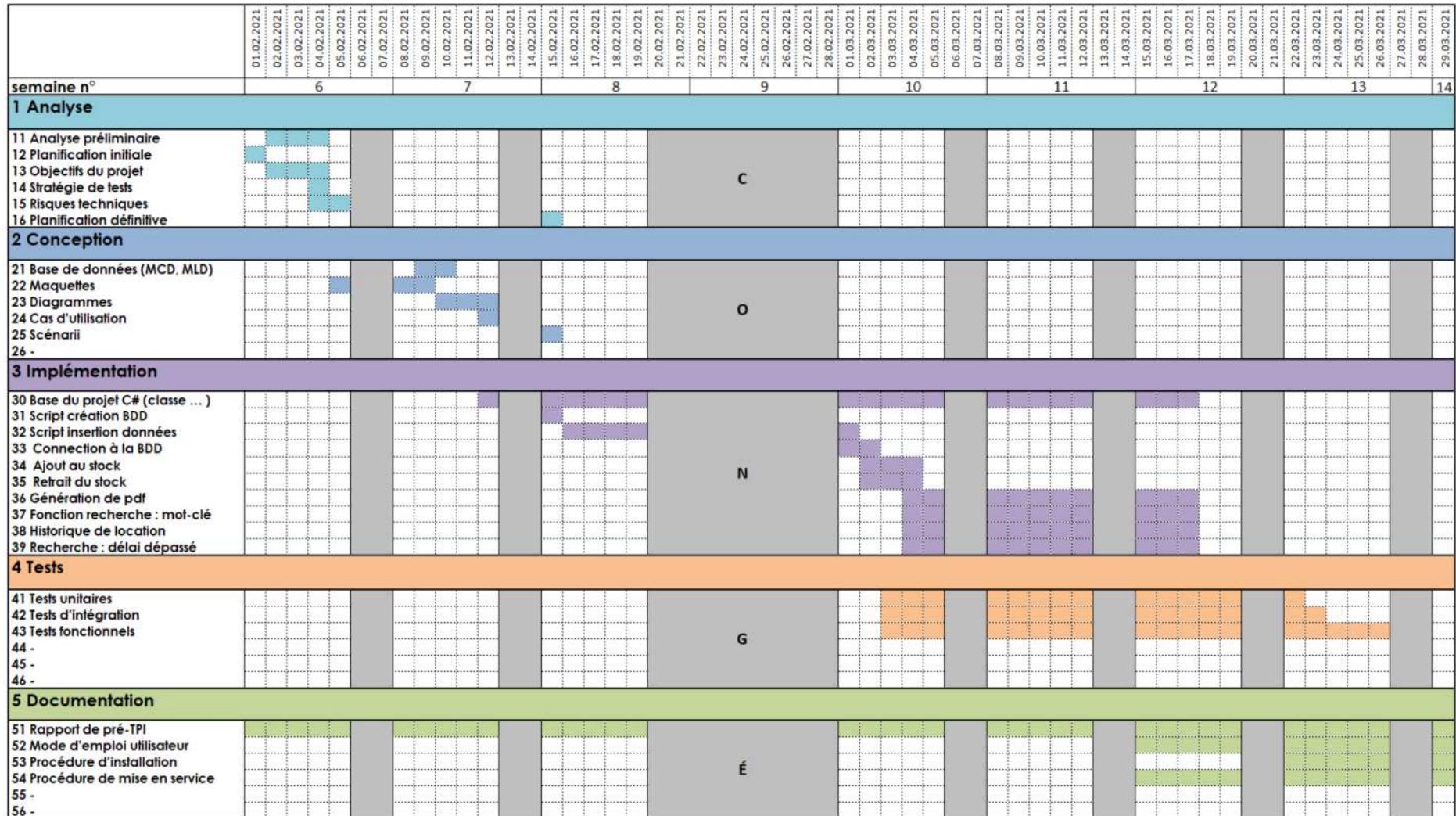
## 2.6 Risques techniques

Malheureusement, le risque zéro n'existe pas. Voici donc les principaux risques liés à ce projet, ainsi que les solutions utilisées pour garantir la meilleure réalisation possible. Tout d'abord, le risque de ne pas réussir à résoudre un souci d'implémentation est relativement élevé. Afin de garantir la détection la plus rapide et le temps de débogage le plus court possible, des tests unitaires seront mis en place. Cette mise en place est prioritaire, car elle permettra d'éviter une perte de temps conséquente par la suite.

Ensuite, la liaison de la base de données à l'application C# est un élément technique spécifique, qui n'a pas été abordé en cours. Pour éviter une perte de temps liée à l'apprentissage de l'implémentation de cet élément, l'aide de Mme F.Andolfatto sera demandée. Il s'agit d'une enseignante d'informatique au CPNV.

Enfin, la longueur du projet peut être un souci majeur. Afin d'éviter cela, la planification proposée respecte des délais assez serrés, laissant cependant un peu de marge en cas de besoin. Cela permet d'empiéter sur cette marge si une tâche en vient à s'éterniser.

## 2.7 Planification définitive



Les variations par rapport à la planification initiale proviennent d'un manque de temps estimé pour la partie analyse. Cela a donc décalé l'intégralité des tâches de développement afin qu'elles se situent à la suite de l'analyse/conception.

## 2.8 Infrastructure

### 2.8.1 Matériel hardware et système d'exploitation

L'infrastructure hardware utilisée pour réaliser ce projet consiste uniquement en un ordinateur du CPNV, installé avec un système d'exploitation Windows10 x64.

### 2.8.2 Outils logiciels

Développement en C# : Microsoft Visual Studio Enterprise 2019, Version 16.8.5

Paquet : MySql.Data version 8.0.23

Paquet : iTextSharp version 5.5.13.2(création de PDF)

Base de données : MySQLWorkbench, version 6.3.6 build 511 CE (64bits)

Historisation des données : L'intégralité du pré-TPI est disponible sur GitHub. Les données sont présentes à l'adresse suivante :

[https://github.com/AhVen98/gestionStocks\\_preTPI](https://github.com/AhVen98/gestionStocks_preTPI)

GitHub est la méthode de versioning utilisée dans ce projet. Il s'agit du moyen mis en place pour pouvoir récupérer les anciennes versions du projet en cas de soucis/besoins.

### **3 Réalisation**

#### **3.1 Dossier de réalisation**

##### **3.1.1 Répertoires et fichiers du projet**

###### **3.1.1.1 Répartition physique des fichiers**

Le projet C# consiste en une solution globale, « locationMateriel ». Celle-ci comporte 3 parties, appelés projets.

- La première, « locationMaterielLibr », contient l'intégralité des classes d'objets ainsi que les méthodes utilisées.
- La seconde, « locationMaterielView », regroupe l'intégralité des WindowsForms. C'est cette partie qui contient tout ce qui a trait au côté graphique du projet.
- La troisième et dernière partie est « locationMaterielTests ». Il s'agit du projet qui contient tous les tests unitaires effectués lors du projet, afin de s'assurer du bon fonctionnement des éléments individuellement.

###### **3.1.1.2 Fichiers et description**

###### **Partie 1 : locationMaterielLibr**

- Persons.cs : il s'agit du fichier où les attributs représentant une personne sont définis.
  - Clients.cs : contient les informations propres aux clients. Elle hérite de « Persons.cs »
  - Employees.cs : contient les informations contenants les employés. Il y a également toutes les méthodes permettant d'ajouter, louer et retourner des objets. Cette classe hérite également de « Persons.cs ».

Clients n'est actuellement pas utilisée, sa mise en place provient de l'analyse. La classe n'a pas été supprimée, car dans le cas où l'application serait poursuivie, sa présence permettra un système d'authentification et d'ajout de client.

- Objects.cs : ce fichier contient la classe avec les objets. Un objet représente n'importe quel élément du stock. Il s'agit du point névralgique de l'application. Sans cette classe, aucune donnée ne s'affiche sur l'application. Les seules méthodes mises en place sont les constructeurs et les getters.
- ConnectionDB.cs : il s'agit du fichier contenant les méthodes génériques reliées à la base de données. Il permet d'éviter de copier-coller sans arrêt les mêmes lignes de code.
- RequestDB.cs : ce fichier contient toutes les méthodes liées aux requêtes à la base de données. Il s'agit d'un point central de l'application. Sans ce fichier, aucune information ne peut être lue, ajoutée ou modifiée dans la base de données.

###### **Partie 2 : locationMaterielView**

Chaque fichier cité ci-dessous contient en réalité deux fichiers. Le premier, « .Designer.cs », comprend la partie graphique du fichier. Le second, « .cs », comprend les méthodes liées à cette partie graphique.

- AddObjectForm : il s'agit du formulaire permettant d'ajouter un objet à la base de données
- ChoicePDFForm : ce formulaire permet de sélectionner le type de PDF à imprimer. Cette partie de l'application n'est pas encore fonctionnelle pour le moment, mais le formulaire est mis en place.

- locMateriel : il s'agit du visuel principal. C'est à partir de celui-ci que tous les autres formulaires sont accessibles par des boutons, placés sur l'application. Toutes les fonctionnalités demandées dans le cahier des charges sont placées sur le visuel, mais toutes ne sont pas fonctionnelles.
- RentObjectForm : il s'agit du formulaire permettant d'effectuer la location d'un objet.
- ReturnObjectForm : ce formulaire permet de rendre un objet déjà loué.

### Partie 3 : locationMaterielTests

Cette partie est censée comporter l'intégralité des tests unitaires effectués dans ce projet. Au final, suite à une conception incompatible avec cette méthode, les tests unitaires n'ont pas été automatisés. Les fonctions ont cependant été testées lors des tests fonctionnels.

- ObjectsTests.cs : ce fichier contient les tests liés à la classe Objects.

### **3.1.2 Produit fini**

Le produit est actuellement en version 0.7. Il s'agit d'une version partiellement utilisable. Elle est simple d'utilisation, mais manque encore de certains éléments nécessaires. De même, certains des boutons affichés n'exécutent rien du tout.

Les logiciels utilisés ont les versions suivantes :

- Microsoft Visual Studio Enterprise 2019, Version 16.8.5
  - 1. MySql.Data version 8.0.23
  - 2. iTextSharp version 5.5.13.2(création de PDF)
- MySQLWorkbench, version 6.3.6 build 511 CE (64bits)

La version du système d'exploitation est du Windows10 x64. Étant donné le projet et les fichiers qui ont dû être créés, il est nécessaire d'être attentif au fait que l'application est fonctionnelle uniquement sur un système d'exploitation Windows.

## **3.2 Liste des documents fournis**

Les documents fournis aux clients en complément de l'application sont les suivants :

- Un rapport de projet, version 1.0
- Un manuel d'installation, version 1.0 (voir annexe [5.4 Manuel d'installation](#))
- Un mode d'emploi, version 1.0 (voir annexe [5.5 Mode d'emploi](#))

Tous ces fichiers seront fournis en version PDF.

### **3.2.1 Programmation et scripts**

#### **Interaction avec la base de données :**

La base de données s'intitule locationMateriel. Afin de générer la base de données, deux fichiers existent, « createDatabase.sql » et « insertData.sql ». Le 1<sup>er</sup> fichier permet de générer la structure de la base de données. Le 2<sup>ème</sup> contient les données afin de pouvoir tester la base.

Même si les infos sont peu nombreuses, elles sont suffisantes pour permettre de tester les différentes fonctionnalités.

Afin d'interagir avec la base de données, il est nécessaire d'installer le paquet MySql.Data sur Visual Studio. Cela permet d'obtenir les méthodes nécessaires à faire les requêtes.

Les deux fichiers concernant les requêtes à la base de données, dans le projet, sont les suivants :

- ConnectionDB.cs
- RequestDB.cs

Le premier (ConnectionDB) contient l'intégralité des méthodes nécessaires pour pouvoir effectuer l'entièreté des étapes d'une requête SQL. Le deuxième (RequestDB) contient les requêtes à proprement parler.

### 3.3 Description des tests effectués

Les tests sont tous réalisés dans l'environnement sur lequel le développement a été réalisé. Les preuves sont apportées par capture d'écran, jointes au document ci-dessous.

Dans certains cas, les tests ont été mis en place et ont dû être supprimés par la suite, car leur réalisation n'était au final pas faisable. Cela est expliqué au point [3.2.1 Tests unitaires](#).

#### 3.3.1 Tests unitaires

Les tests unitaires sont des tests effectués sur les éléments de manière séparée. Ils sont présents afin d'éviter les soucis lorsque tout est mis en commun par la suite.

- Tests de la classe « « Objects » »

Les tests de cette classe permettent de déterminer que les différents constructeurs récupèrent bien les bonnes valeurs.

- Tests de la classe « « Employees », « Clients », « Persons » »

Les tests ne sont pas concluants et irréalisables de manière automatisée, car les getters (afin d'accéder aux données depuis l'extérieur de la classe) ne sont pas mis en place.

- Tests de la classe « « RequestDB » » et « « ConnectionDB » »

Il n'existe pas vraiment de manière de tester les deux classes, car il s'agit d'appels sur une base de données. Si des soucis avaient lieu dans ces méthodes, le programme ne s'exécuterait tout simplement pas.

Pour tester les requêtes, il faudrait les exécuter sur MySQLWorkbench et vérifier que le retour soit bien celui attendu.

Les différentes vues de l'application (fichiers WindowsForms) ne sont pas testées. Il s'agit du côté graphique de l'application.

Pour les tests unitaires qui n'ont pas pu être réalisés, le bon fonctionnement des éléments sera prouvé dans la partie [3.2.3 Tests fonctionnels](#). Il n'est en effet pas possible de faire fonctionner un bloc de l'application si une unité est défectueuse.

#### 3.3.2 Tests fonctionnels

Pour réaliser les tests fonctionnels, la méthode utilisée est toujours la même :

- Insérer/sélectionner les données pertinentes (si nécessaire)
- Cliquer sur le bouton correspondant
- Insérer les données demandées (si nécessaire) deux fois
  - o Données valides
  - o Données invalides
- Confirmer l'action testée
- Vérifier le résultat
  - o Il doit correspondre à ce qui est attendu

Test	Durée	Ca
locationMaterielTests (3)	34 ms	
locationMateriel.Tests (3)	34 ms	
ObjectsTests (3)	34 ms	
Objects_AllParam_OK	34 ms	
Objects_AllParamNoDate_OK	< 1 ms	
Objects_AllParamNoRenter.OK	< 1 ms	

Figure 9 : tests unitaires de la classe "Objects"

### 3.3.2.1 Ajouter un objet

Avec des données valides, le résultat escompté est obtenu. Les données sont bien ajoutées à la base de données :

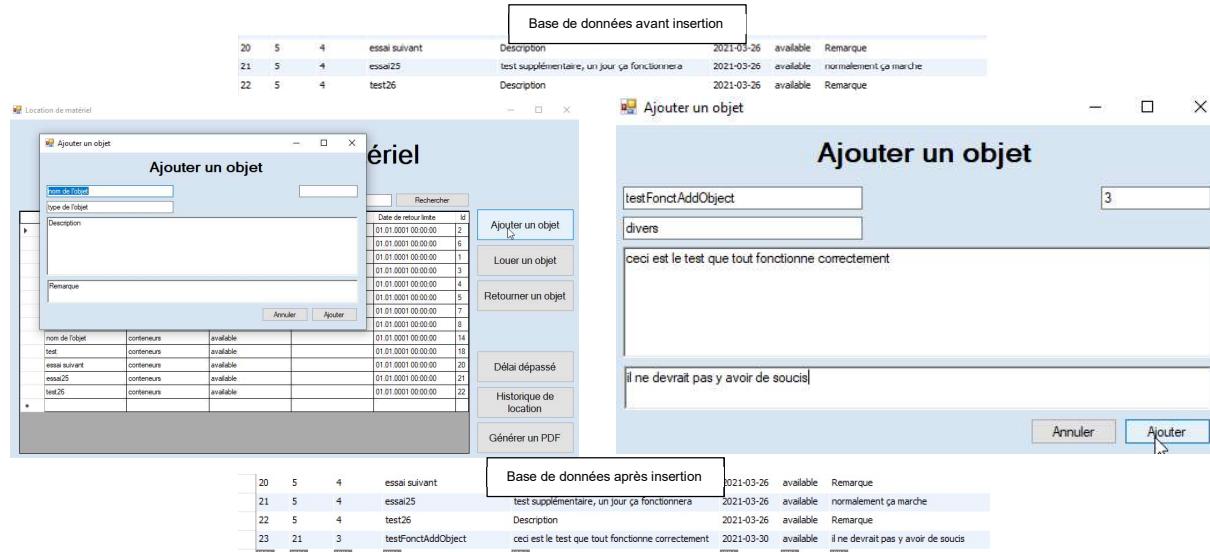


Figure 10 : test "Ajouter un objet" avec des données valides

En mettant un mauvais type d'objet, cela fonctionnera toujours : la valeur par défaut a été réglé sur « Divers ». Par contre, en mettant un numéro d'utilisateur qui n'est pas un entier (ici, chaîne de texte), aucune modification n'a lieu dans la base de données.

### 3.3.2.2 Louer un objet

La location d'objet a actuellement un souci -> la récupération de l'état ne fonctionne pas correctement, ce qui résulte en l'impossibilité de louer l'objet, comme l'action par défaut est d'annoncer à l'employé que l'objet n'est pas disponible à la location.

La fonction permettant d'effectuer la location fonctionne, le souci se trouve au niveau de la récupération de l'état de l'objet.

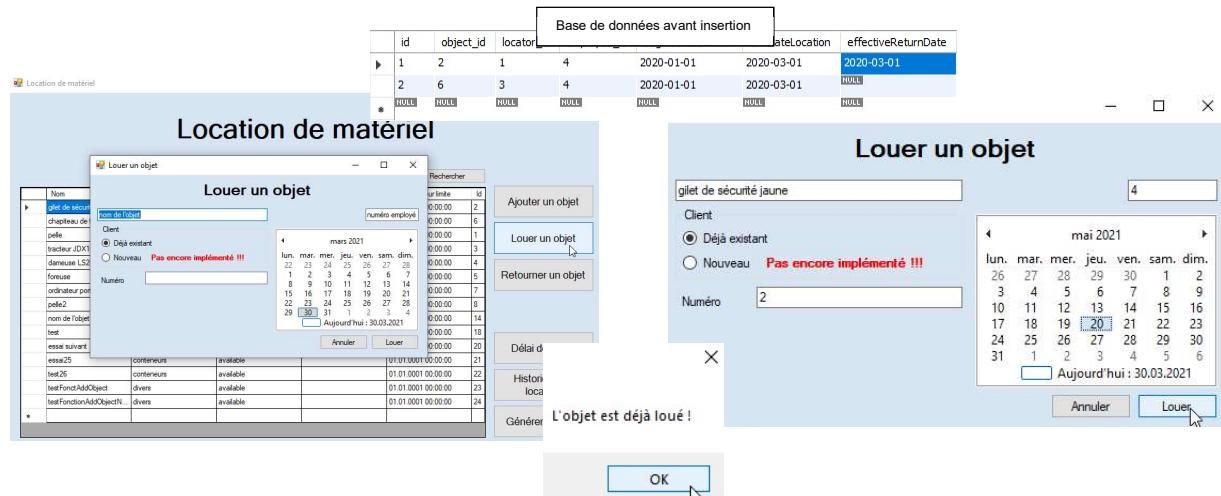


Figure 11 : test "Louer un objet" avec des données valides

Le fait d'insérer des données non valides fonctionnent, car il s'agit de l'action par défaut réalisée.

### 3.3.2.3 Retourner un objet

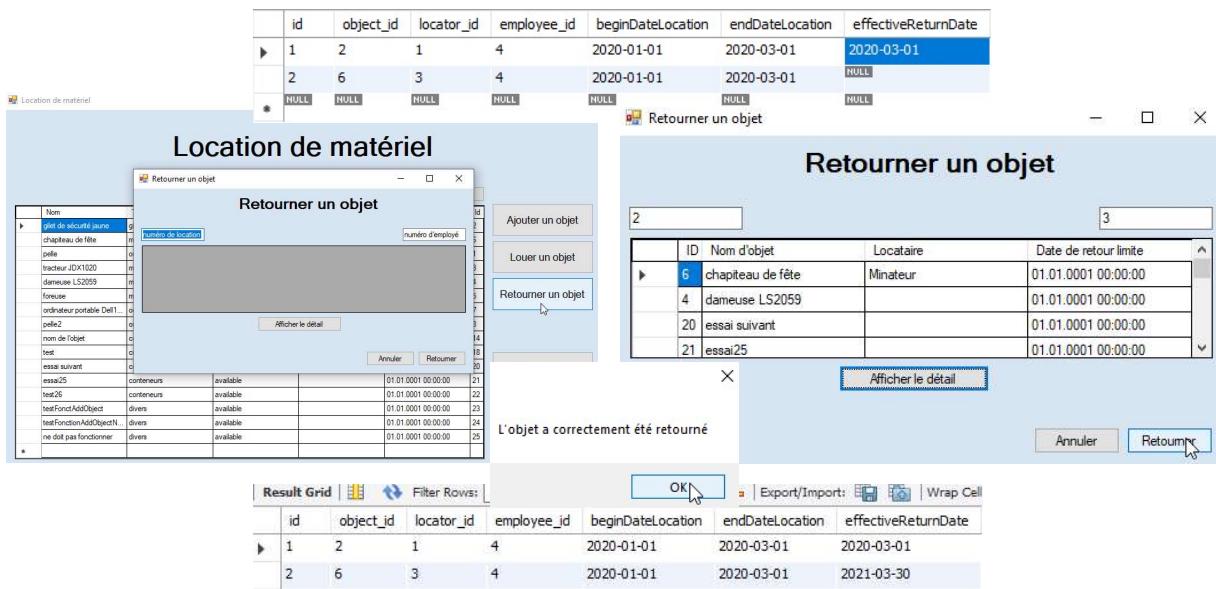


Figure 12 : test "Retourner un objet avec des données valides"

On peut voir qu'il y a un souci au niveau de la fonction « Afficher le détail ». C'est censé montrer uniquement l'objet loué et cela retourne l'intégralité de la liste des objets de la base de données.

En mettant un numéro de location invalide, le message comme quoi l'objet a été correctement s'affiche également, même si aucune modification n'a lieu au niveau de la base de données. Il s'agit d'un souci à régler, même si l'erreur n'a pas de conséquences actuellement.

### 3.3.2.4 Délai dépassé

En cliquant sur le bouton « Délai dépassé », seuls les objets loués dont le délai n'est pas respecté sont censés s'afficher. Actuellement, rien ne s'affiche, car le code derrière le bouton n'est pas encore développé.

### 3.3.2.5 Rechercher

Après avoir entré du texte dans le champ de texte et cliqué sur « Rechercher », seuls les lignes où le nom ou la description correspond sont censés s'afficher. Actuellement, il n'y a aucun changement sur les données affichées, comme le code derrière le bouton n'est pas développé.

### 3.3.2.6 Générer le PDF

Après avoir cliqué sur « Générer le PDF » et sélectionné l'option souhaitée sur le formulaire, rien ne se passe. Le code censé permettre générer le PDF n'est pas développé et ne retourne donc rien.

### 3.3.2.7 Historique de location

En sélectionnant un objet et cliquant « Historique de location », un formulaire avec le détail de l'objet est censé s'afficher. Rien ne s'affiche sur le programme et le formulaire n'apparaît, car cette partie n'a pas été développée.

### 3.4 Problèmes rencontrés et résolution

#### [18.02.2021] Script d'insertion des données

Lors de la création du script d'insertion des données, deux soucis majeurs se sont posés. Le premier a été de trouver des données pertinentes. Le but est d'avoir des données suffisamment variées pour que toutes les possibilités soient clairement visibles, sans pour autant avoir 1'500 entrées dans la base de données.

Ce souci a été résolu à l'aide de Wikipédia, où je me suis inspirée pour nommer les catégories et les types.

#### [01.03.2021] Mise en place des classes pour les tests unitaires

Lors de la mise en place des classes des tests unitaires le lundi matin 1<sup>er</sup> mars 2021, un souci s'est présenté : les classes appelées depuis le reste du projet n'étaient pas reconnues par la classe de test.

L'aide de l'enseignant J.Ithurbide a permis de résoudre le problème (la classe « Objects » n'était pas déclarée publique et le deuxième projet de la solution n'était pas référencé dans le projet de test) et a prévenu un souci qui aurait pu se poser au futur : l'organisation des fichiers.

L'organisation qu'il a proposée suit une architecture MVC, qui permet de séparer en 3 blocs distincts l'application.

#### [05.03.2021] Interaction entre plusieurs fichiers

Lors de l'implémentation des premières requêtes (fichier « RequestDB »), au moment de faire appel à des méthodes présentes dans un autre fichier, des erreurs de référence sont apparues.

L'aide de F.Andolfatto a permis de corriger le problème très rapidement : il s'agissait d'un oubli de ma part. Aucun objet n'était initialisé au début de la classe.

#### [05.03.2021] Faire appel à un autre projet dans une solution unique

La modification de la structure du projet en architecture MVC a vu un nouveau problème apparaître : la présence de types de projets différents. L'entièreté des données sont stockées dans un projet de type librairie, alors que les visuels et le contrôleur sont dans un projet de type WindowsForms. Hors, dans un projet de type « Bibliothèque de classe (.NET standard) », aucune référence ne s'affichait.

Avec F.Andolfatto, plusieurs tests successifs de création de projets ont permis de démontrer que le type de projet utilisé ne permettait pas le référencement. Afin de pouvoir avoir des références, il a été nécessaire de déplacer les fichiers dans un nouveau projet avec un type différent : « Bibliothèque de classe (.NET Framework) ». Ce type permet de référencer d'autres projets. Il est cependant incompatible avec des systèmes IOS ou Android.

#### [08.03.21] Débug des requêtes SQL

La création de nombreuses méthodes utilisées pour exécuter une seule requête augmente la probabilité d'avoir des erreurs/bugs lors de l'exécution du programme. Plusieurs de ces soucis se sont déclarés aujourd'hui. Entre eux, des messages de fonctionnement impossible, car un processus restait ouvert.

L'aide de X.Carrel a permis de résoudre l'entièreté de ces erreurs. La principale était une étape oubliée dans le fonctionnement d'une requête en C#. Il est nécessaire de supprimer le DataReader une fois qu'il a été exécuté et cette étape n'avait pas été faite.

#### [12.03.21] Absence

Un état grippal, avec une difficulté à respirer (nez et gorge pris) et un mal de crâne carabiné, ont fait qu'aucune avancée n'a été réalisée sur le projet ce vendredi. Il y a donc 2 périodes « à vide » cette semaine-là.

Aucune période de rattrapage n'est prévue, il y aura juste 90 minutes de moins sur le temps total, avec l'impact que cela peut avoir sur l'avancée du projet.

#### [26.03.21] Affichage de données en multiples colonnes

Un souci majeur, qui a posé problème tout le long du projet, a été l'affichage des données. Cela a posé comme problème majeur des retards accumulés tout le long du projet avec une application qui, ce vendredi 26 mars 2021, n'affichait encore rien du tout.

L'aide de F.Andolfatto a permis de trouver une solution. Il s'agissait de l'utilisation d'un mauvais outil d'affichage (ListView au lieu de DataGridView) et une complexité trop élevée au niveau des redirections dans les fichiers. La simplification qui a été apportée a permis de dépanner la majeure partie de l'application et d'avoir enfin un résultat qui affiche quelque chose de concret.

Il reste cependant beaucoup de travail et certaines données sont encore manquantes.

### **3.5 Erreurs restantes**

Arrivé à la fin du projet, de nombreuses erreurs sont encore présentes. Les voici présentées ci-dessous, avec une description associée, quelles conséquences en découlent et des moyens de le corriger.

Il est clair qu'il suffit de temps pour résoudre la majeure partie de ces erreurs, il n'est pas censé y avoir d'éléments qui pourrait rendre le projet infaisable.

De plus, en se basant sur les scénarios, les messages de confirmation ne sont pas présents. Il s'agit d'un élément supplémentaire qui n'a pas été implémenté.

Ces informations découlent principalement du point [3.2.2 Tests fonctionnels](#).

#### Fonction de location des objets :

La fonction de location des objets indique systématiquement que l'objet est déjà loué. Il s'agit d'un souci sur la requête pour récupérer l'état de l'objet.

La conséquence pour le projet est assez importante, car une fonctionnalité centrale du projet ne fonctionne pas correctement.

Afin de corriger cet élément, il faut aller débugger la partie qui récupère l'état d'un objet à partir d'un ID.

#### Affichage des dates :

L'affichage des dates présente un souci. Peu importe si une date est donnée ou pas, on voit systématiquement « 01.01.0001 00:00:00 » affiché. Le souci est que la variable fournie n'a aucun paramètre, cela a pour conséquence de mettre une valeur par défaut. On parle de « l'hydratation des données ».

Cela n'a aucune conséquence réelle sur l'application, ce n'est que de l'affichage.

Afin de régler ce souci, il va falloir gérer l'absence de valeur de la colonne des dates de retour. C'est cette absence de valeur qui génère cette date complètement absurde et sans pertinence.

Bouton de tri :

Le bouton permettant de trier par délai dépassé n'est pas fonctionnel à cause d'un manque de temps. Il n'a pas été possible d'arriver jusque-là dans l'application. Il est cependant possible de trier par la colonne « date de retour attendue », même si, avec le problème d'affichage de dates, ça n'a actuellement aucune utilité.

Par rapport aux fonctionnalités, cela n'a pas de conséquences autres qu'empêcher de trier les données pour voir uniquement les objets loués avec un délai dépassé.

Il n'y a rien à corriger à cet endroit, il faut juste implémenter le tout.

Bouton Rechercher :

Le champ texte et le bouton Rechercher à côté ne sont actuellement pas fonctionnels. Encore une fois, il s'agit d'un manque de temps pour la réalisation. Comme il ne s'agit pas d'une des fonctionnalités de base de l'application, nécessaire au bon fonctionnement, cela a été repoussé et n'a au final pas été fait.

La seule conséquence liée à ce souci est une absence de recherche par terme, cela n'impacte pas le reste de l'application.

L'implémentation du bouton est tout ce qui est nécessaire pour que cette partie soit fonctionnelle.

Accès à l'historique d'un objet :

Cette fonctionnalité, bien que centrale, n'a pas d'impact réel sur les autres. De plus, elle nécessite que l'intégralité des autres fonctionnalités soient en place pour pouvoir être développée. Vu le retard accumulé sur le reste du projet, il n'y a donc pas eu de temps pour la réaliser. Le bouton est présent, mais n'a aucun effet.

À part l'absence de la fonction d'historique de l'objet, il n'y a pas de conséquences négatives sur l'application.

Le développement du code derrière le bouton suffira à créer l'historique d'un objet.

Génération du fichier PDF :

La génération d'un PDF est un point technique très spécifique, qui nécessite, encore une fois, que le reste des fonctionnalités de base soit en place. Il n'a donc pas été possible de le développer, car le reste ne fonctionnait pas à temps.

L'absence de cet élément n'a aucun impact sur le reste de l'application.

Le formulaire pour le choix du type de PDF est prêt, ce qui manque est le code derrière ces boutons. Cela peut cependant être compliqué, il s'agit d'un paquet à part entière qui a été ajouté pour le permettre.

Affichage statique

Pour le moment, l'affichage sur la page principale est un affichage statique. Cela veut dire que, quand une modification est réalisée dans la base de données, cela ne modifie en rien ce qui s'affiche sur l'application.

D'un point de vue fonctionnalité, cela n'a aucun impact. Par contre, il n'est donc pas possible de se reposer sur ce qui est visible sur l'application pour savoir ce qui est réel comme information.

Afin de corriger ce souci, il faut régler le souci de recharger les données à chaque fois qu'une action est réalisée. Cela n'est pas très compliqué en théorie, mais peut vite prendre du temps.

## 4 Conclusions

### 4.1 Atteinte des objectifs

Comme spécifié au début du projet, dans [1.2 Objectifs](#) et [2.1.1 Fonctionnalités](#), de nombreux objectifs ont été émis. Tous n'ont pas pu être remplis, voici un récapitulatif rapide de l'atteinte de ces objectifs :

Le stockage des données a bien lieu dans une base de données.

La connexion à la base de données s'effectue bien au lancement de l'application. Il ne s'agit cependant pas d'une base de données distantes pour le moment, mais d'une base de données locales.

Toutes les pièces (objets) du stock sont uniques et définis comme tel par un ID.

L'ajout d'une pièce au stock peut être fait. Une pièce ajoutée par un ID déjà existant est un retour de location. Un ajout d'une nouvelle pièce demande de préciser tous les paramètres.

La location d'une pièce se fait par l'ID de l'objet et le numéro de client.

La génération du rapport en PDF n'est pas mise en place pour le moment, mais est prévue dans l'application.

La recherche par mot-clé est présente, mais n'a pas encore été développée.

La fonction de tri par date de retour dépassée est prévue (bouton déjà en place), mais n'a pas été implémentée actuellement.

L'historique d'une pièce n'est pas disponible, mais le bouton est prêt pour pouvoir le faire.

L'application est simple d'utilisation et convient tout à fait à des personnes débutantes dans le domaine de l'informatique.

L'application n'est pas « clé en main ». Il reste du travail de développement à réaliser pour pouvoir considérer l'application comme terminé.

D'un point de vue global, même si l'entièreté des objectifs n'ont pas été réalisés, les points les plus importants selon moi sont complétés et l'avancée du projet est significative malgré tout. Il reste cependant de la place pour de l'amélioration.

### 4.2 Maintien des délais

Malgré la fin du délai imparti, il reste encore de nombreuses erreurs et de nombreux soucis présents dans l'application.

La raison principale de ce manquement a été une erreur d'analyse de ma part : lors de l'analyse, il m'était complètement sorti de tête que j'étais absente trois jours la semaine du 15 mars 2021. La résultante en a été presque 8h de moins à accorder au projet.

Une deuxième raison, qui a d'ailleurs été énoncé au tout début dans les risques techniques, a été la mise en place des requêtes liées à la base de données. De nombreuses erreurs et faux-pas à plusieurs reprises ont représenté une perte de temps conséquente.

Ensuite, ce projet s'est révélé être beaucoup plus complexe que prévu, intégrant des notions qui n'avaient pas été abordées en cours. Bien que cela ait été un plaisir de les découvrir, ainsi que de se rendre compte de la réalité du monde du travail, où l'on ne connaît pas forcément tout de la technologie que l'on doit utiliser, cela a rendu la réalisation de ce projet très compliquée et l'a énormément ralenti.

La présence de nombreuses erreurs tout au long du projet a également causé une baisse de motivation en fin de semaine 4, ce qui fait que du temps a été perdu à ce moment-là, car l'efficacité de travail n'était plus la même.

### 4.3 Points positifs et négatifs

Ce projet m'a permis de découvrir beaucoup de choses. D'un certain côté, j'ai appris énormément sur l'importance d'une analyse et conception complète et détaillée. J'ai également découvert l'importance de connaître la technologie utilisée autant que possible. Ainsi, malgré une connaissance importante des WindowsForms et des projets de type librairie, je n'étais pas du tout préparée à devoir intégrer les deux éléments ensemble et cela m'a causé de nombreux soucis.

Ce projet a donc un gros point positif : pour mon TPI, qui sera le prochain projet, je suis bien plus informée sur les risques techniques, le temps nécessaire à la réalisation des différentes tâches, ainsi qu'à la manière de débugger de manière optimale un code. Un élément supplémentaire très positif a été que je me suis rendue compte de mes capacités. Je ne suis clairement pas la plus douée en développement, mais je suis suffisamment compétente pour faire une application qui tient la route, bien documentée avec tous les éléments nécessaires pour que le projet soit repris par quelqu'un d'autre si besoin est.

Malgré cela, il y a eu un côté un peu plus noir à ce projet.

Un des points négatifs du projet a été son avancée. Malgré tous les efforts fournis, le projet n'est pas complété et c'est un élément qui m'embête particulièrement. Un autre point négatif a été mon incapacité à régler certains problèmes seuls et d'avoir dû demander de l'aide pour le faire. Les résoudre seule aurait pu, potentiellement, se faire, mais cela aurait demandé un temps impressionnant et aurait donc diminué l'avancée du développement de manière significative.

### 4.4 Difficultés particulières

Comme tout projet, il arrive que des difficultés se présentent. Lors de celui-ci, certains points spécifiques ont été problématiques. L'élément le plus marquant à ce sujet a été l'intégration d'un projet de type librairie ainsi qu'un projet de type WindowsForms dans la même solution. En combinant ces éléments avec la gestion des appels à une base de données, les compétences nécessaires étaient bien supérieures à ce qui était prévu et a compliqué le développement de l'application.

La gestion du temps est également une difficulté marquée dans le projet. Il est complexe de se projeter correctement afin de savoir quelle tâche nécessite quel temps pour la réalisation. C'est ce point qui a majoritairement contribué au fait que le projet soit rendu alors qu'il n'est pas terminé.

### 4.5 Évolutions et améliorations

Au vu de l'avancée actuelle de l'application, les premières améliorations à apporter au projet serait de finir de développer les fonctionnalités de base annoncées. Une fois cela fait, de nombreuses possibilités existent. En voici quelques-unes ci-dessous :

- Modifier la méthode « Louer un objet » pour louer plusieurs objets d'un coup
- Implémenter l'ajout de clients à la base de données
- Transformer le champ « type » lors de l'ajout d'objet en drop-down, avec les options disponibles
- Mettre un système d'authentification en place (cela permettrait d'enlever le champ « Numéro d'employé » sur les formulaires)
- Ajouter des messages de confirmation ou d'erreurs aux endroits pertinents

Ces possibilités d'améliorations ne sont pas les seules existantes, mais il s'agit de possibilités à partir de l'application actuelle.

## 5 Annexes

### 5.1 Sources – Bibliographie

#### 5.1.1 Pages internet consultées

Page des TPI vaudois : <http://www.tpivd.ch/> [12.02.21]

Recherche d'informations sur le contenu nécessaire dans le rapport de projet

Documentation Microsoft, pour la création de tests unitaires [01.03.2021]

<https://docs.microsoft.com/en-us/visualstudio/test/walkthrough-creating-and-running-unit-tests-for-managed-code?view=vs-2019>

Recherche de la manière correcte de mettre en place un projet de tests dans une solution

Forum [commentcamarache.net](http://commentcamarache.net), pour la syntaxe d'appel de formulaire [04.03.2021]

<https://codes-sources.commentcamarache.net/forum/affich-150911-appel-d-un-form-a-partir-d-un-autre-en-c>

Recherche de la syntaxe pour faire un appel sur un autre formulaire, afin d'avoir un code le plus propre possible

Article sur [c-sharpcorner.com](http://c-sharpcorner.com), pour la mise en place du PDF [15.03.21]

<https://www.c-sharpcorner.com/UploadFile/f4f047/generating-pdf-file-using-C-Sharp/>

Recherche d'un paquet nugget pour pouvoir créer des fichiers PDF à partir d'une application C#

Documentation Microsoft, sur le fonctionnement des listViews [26.03.21]

<https://docs.microsoft.com/en-us/dotnet/desktop/winforms/controls/how-to-add-and-remove-items-with-the-windows-forms-listview-control?view=netframeworkdesktop-4.8>

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.listviewitem?view=net-5.0>

Recherche sur la manière de faire fonctionner les listViews. Cet élément a finalement été abandonné pour le DataGrid

#### 5.1.2 Personnes consultées

Mme F.Andolfatto

- [05.02.21] Précisions sur la liaison des bases de données à un projet en C#
- [12.02.21] Retour sur la pertinence et l'exactitude d'un diagramme de classe
- [05.03.21] Interactions entre différents fichiers et projets
- [26.03.21] Résolution du problème d'affichage des données, à l'aide de l'élément « DataGrid »

M J.Ithurbide

- [01.03.21] Résolution du problème de mise en place du projet de test et explication sur l'architecture MVC dans un projet incluant WindowsForms et librairie
- [08.03.21] Conversion des données d'un type à un autre (ici, string à int) en C#

M X.Carrel

- [08.03.21] débug de requêtes SQL en C#

## 5.2 Journal de travail

Ceci est le journal de travail de la semaine 6 de 2021. Il s'agit de la semaine n°1 du projet, du 1<sup>er</sup> février 2021 au 5 février 2021.

Date	Semaine	Activité	Avancée	Heures	Remarques
01.02.2021	6	55 -		0:30	Discussion du déroulement du projet, reçu du CdC, réponses aux
01.02.2021	6	12 Planification initiale	En cours	1:00	Mise en place du fichier excel de la planification
01.02.2021	6	51 Rapport de pré-TPI		0:10	Mise en place du canevas pour le rapport complet
01.02.2021	6	12 Planification initiale	Terminé	0:30	complétion de la planification initiale
03.02.2021	6	13 Objectifs du projet	Terminé	0:30	rédaction des objectifs
03.02.2021	6	11 Analyse préliminaire	En cours	0:30	réflexion sur l'introduction au projet et comment la rédiger, prise de notes -> structure du paragraphe
04.02.2021	6	11 Analyse préliminaire	Terminé	0:30	Rédaction de l'introduction
04.02.2021	6	14 Stratégie de tests	Terminé	0:30	Rédaction de la stratégie de test
04.02.2021	6	22 Maquettes	En cours	0:20	début des maquettes
05.02.2021	6	12 Planification initiale	Terminé	0:45	correction d'une erreur -> passage d'un tableau d'heures à un diagramme de Gant
05.02.2021	6	51 Rapport de pré-TPI		0:10	ajout de la planification initiale au dossier
05.02.2021	6	15 Risques techniques	Terminé	0:30	rédaction des risques techniques du projet
05.02.2021	6	22 Maquettes	En cours	0:35	avancée sur maquette de l'application
05.02.2021	6	55 -		0:10	retour sur le rendu de la planification discussion du CdC
05.02.2021	6	22 Maquettes	En cours	0:20	avancée sur les maquettes -> la maquette de globale de l'application est terminée
05.02.2021	6	51 Rapport de pré-TPI		0:45	travail sur le visuel du document, avec mise en place de sections, d'une page de garde, d'une table des matières
05.02.2021	6	21 Base de données (MCD, MLD)	En cours	0:20	mise en place d'un fichier pour le MCD et réflexion sur les entités et relations

Ceci est le journal de travail de la semaine 7 de 2021. Il s'agit de la semaine n°2 du projet, du 8 février 2021 au 12 février 2021.

Date	Semaine	Activité	Avancée	Heures	Remarques
08.02.2021	7	22 Maquettes	En cours	0:45	maquette du formulaire d'ajout d'objet faite
08.02.2021	7	51 Rapport de pré-TPI		0:45	ajout de la maquette à la conception fermeture du point maquette correction d'une erreur de pied-de-page
08.02.2021	7	21 Base de données (MCD, MLD)	En cours	0:20	avancée du MCD : ajout table locataires, objets, types création des relations
08.02.2021	7	21 Base de données (MCD,	En cours	1:10	fin du MCD
08.02.2021	7	51 Rapport de pré-TPI		0:15	ajout du MCD au rapport (insertion image) rédaction d'un paragraphe d'explication
10.02.2021	7	21 Base de données (MCD,	Terminé	1:00	création du MLD sur la base du MCD
11.02.2021	7	51 Rapport de pré-TPI		0:20	le paragraphe concernant le MLD était hors-sujet -> suppression compléter de la partie MCD/MLD (insertion images, rédaction du texte)
11.02.2021	7	23 Diagrammes	En cours	0:50	création du diagramme de flux "ajouter objet" et "retourner objet"
11.02.2021	7	51 Rapport de pré-TPI		0:20	insertion des diagrammes dans le rapport ajout des légendes aux différentes figures du document
12.02.2021	7	55 -		0:15	discussion sur la pertinence du MCD/MLD retour sur le rendu du rapport de la semaine précédente -> les objectifs sont mal situés, leur précision les emmène dans le concept, ceux de l'analyse préliminaire sont à refaire
12.02.2021	7	23 Diagrammes	En cours	0:30	création du diagramme de classe -> l'aide de Mme Andolfatto sera nécessaire pour certifier l'exactitude du diagramme
12.02.2021	7	30 Base du projet C# (classe ... )	En cours	0:15	mise à jour de l'application création des classes et du windows form principal
12.02.2021	7	23 Diagrammes	Terminé	0:10	discussion avec Mme Andolfatto pour valider l'exactitude du diagramme
12.02.2021	7	24 Cas d'utilisation	Terminé	0:20	Création du diagramme pour les use cases
12.02.2021	7	51 Rapport de pré-TPI		0:45	ajout du diagramme des Use Cases insertion des légendes aux figures insertion des sources dans le rapport déplacement des objectifs dans le paragraphe correct, rédaction d'un texte simplifié pour les objectifs de l'analyse préliminaire début de la rédaction de l'infrastructure utilisée pendant le projet

Ceci est le journal de travail de la semaine 8 de 2021. Il s'agit de la semaine n°3 du projet, du 15 février 2021 au 19 février 2021.

Date	Semaine	Activité	Avancée	Heures	Remarques
15.02.2021	8	25 Scénarioii	En cours	0:50	Rédaction des scénarioii
15.02.2021	8	25 Scénarioii	Terminé	0:45	Fin de la rédaction des scénarioii
15.02.2021	8	16 Planification définitive	Terminé	0:25	Ajout de la planification définitive -> l'analyse/conception est terminé dans sa majorité, seul du détail au niveau du hardware est manquant -> il sera précisé au fil de la semaine, quand les systèmes seront tous en place les uns par rapport aux autres
15.02.2021	8	31 Script création BDD	Terminé	0:25	Le script de création de la base de données a été générés sur la base du MCD et du MLD
15.02.2021	8	30 Base du projet C# (classe ... )	En cours	0:30	Réflexion sur la meilleure façon de mettre en place les fichiers de l'application
17.02.2021	8	30 Base du projet C# (classe ... )	En cours	0:50	Mise en place du visuel de la page d'accueil de l'application, avec création des boutons, champs de recherche, etc création des constructeurs des classes Persons, Employées et des méthodes
17.02.2021	8	23 Diagrammes		0:05	Correction d'une erreur d'accessibilité sur les méthodes elles sont publiques et non pas privées
18.02.2021	8	30 Base du projet C# (classe ... )		0:10	Finition de l'implémentation de l'héritage sur "Clients"
18.02.2021	8	32 Script insertion données	En cours	0:40	Mise en place de l'insertion des données
18.02.2021	8	51 Rapport de pré-TPI		0:05	Correction d'une erreur de numérotation remplacement du diagramme de classe -> erreur de typo
18.02.2021	8	32 Script insertion données	En cours	0:25	Avancée de l'insertion des données prend plus de temps que prévu, à cause de la recherche pour trouver des informations concrètes
19.02.2021	8	55 -		0:15	discussion avec le chef de projet sur l'avancée et réponses aux questions de compréhension dans le rapport validation de l'avancée + mise en avant des corrections à apporter
19.02.2021	8	51 Rapport de pré-TPI		0:05	correction de termes inadéquats dans la partie analyse
19.02.2021	8	32 Script insertion données	En cours	0:30	réflexion sur les données à mettre dans les tables
19.02.2021	8	32 Script insertion données	En cours	1:00	avancée du script d'insertion des données -> pas de soucis pour les tables de références -> fini -> erreur de format de date pour les dates de naissance, je séche et donc je perds du temps
19.02.2021	8	32 Script insertion données	En cours	1:00	erreur trouvée et corrigée avancée sur les données insérées

Ceci est le journal de travail de la semaine 10 de 2021. Il s'agit de la semaine n°4 du projet, du 1<sup>er</sup> mars 2021 au 5 mars 2021. La semaine non-présente (semaine 9) découle des vacances de carnaval.

Date	Semaine	Activité	Avancée	Heures	Remarques
01.03.2021	10	32 Script insertion données	Terminé	0:20	fin du script d'insertion -> toutes les tables contiennent des données. Afin d'augmenter la pertinence du projet, des données seront probablement ajoutées par la suite, mais ce qui y figure maintenant suffit pour pouvoir tester et implémenter le reste
01.03.2021	10	33 Connection à la BDD	En cours	0:15	la connexion à la bdd est mise en place, elle devrait être utilisable -> sera testé dès que possible
01.03.2021	10	41 Tests unitaires	En cours	0:20	mise en place du projet de test, afin de pouvoir ajouter tous les tests par la suite
01.03.2021	10	41 Tests unitaires	En cours	0:40	souci avec l'initialisation des données -> le projet de test ne reconnaît pas mes classes courantes -> impossible de trouver l'erreur, ça risque de prendre du temps
01.03.2021	10	30 Base du projet C# (classe ...)	En cours	0:55	grâce à J.Ithurbide et par ses explications, l'erreur a été trouvé -> le projet a également été remanié afin de respecter une structure MVC.
01.03.2021	10	51 Rapport de pré-TPI		0:10	ajout du problème avec les tests unitaires au projet
01.03.2021	10	34 Ajout au stock		0:15	réflexion sur la meilleure manière de formuler la méthode d'ajout à la BDD
03.03.2021	10	33 Connection à la BDD	En cours	0:25	restructuration du code lié à la base de données -> les méthodes récurrentes (ouverture et fermeture de la connexion, ...) sont laissées dans le fichier connexionDB. Un fichier pour contenir les requêtes a été créé à part.
03.03.2021	10	30 Base du projet C# (classe ...)	En cours	0:20	début de la mise en place d'un redirecteur dans le contrôleur pour sélectionner la bonne méthode en fonction du bouton appuyé sur l'application.
03.03.2021	10	30 Base du projet C# (classe ...)	En cours	0:15	suite de la mise en place du redirecteur
03.03.2021	10	30 Base du projet C# (classe ...)	En cours	0:05	liaison des boutons de l'application sur le redirecteur
03.03.2021	10	34 Ajout au stock	En cours	0:25	mise en place du formulaire d'ajout d'un objet
04.03.2021	10	34 Ajout au stock	En cours	0:35	recherche de liaison de formulaire -> en cliquant sur le bouton correspondant, un formulaire supplémentaire s'ouvre. La méthode étant trouvée, la liaison des autres formulaires sera beaucoup plus simple
04.03.2021	10	30 Base du projet C# (classe ...)	En cours	0:20	mise en place des premières requêtes SQL pour afficher des données. Recherche nécessaire pour trouver comment afficher des informations multiples sur une même ligne de données
04.03.2021	10	44 -		0:35	recherche sur la façon d'afficher plusieurs infos sur une seule ligne dans une application
05.03.2021	10	34 Ajout au stock	En cours	0:50	essais et avancée sur la mise en place de la méthode "ajouter un objet" -> le temps nécessaire est élevé, mais il s'agit de la première fonction. Une fois qu'elle est en place, le reste ira beaucoup plus vite, car un exemple sera disponible
05.03.2021	10	30 Base du projet C# (classe ...)	En cours	0:35	avancée sur le contrôleur, modification sur le visuel (bouton manquant), avancée des requêtes SQL

Ceci est le journal de travail de la semaine 11 de 2021. Il s'agit de la semaine n°5 du projet, du 8 mars 2021 au 12 mars 2021.

Date	Semaine	Activité	Avancée	Heures	Remarques
08.03.2021	11	34 Ajout au stock	En cours	1:10	essais multiples sur l'avancement de la fonctionnalité d'ajout au stock -> un souci de référence avait lieu le problème de référencement a été détecté et une solution autre devra être mise en place pour régler le problème
08.03.2021	11	55 -		0:20	aide de X.Carrel pour dépanner le SQL -> une incompréhension du fonctionnement d'une requête a causé des soucis de mon côté résolution des soucis initiaux, le problème est désormais dans les données
08.03.2021	11	34 Ajout au stock	En cours	0:30	bidouillage des requêtes SQL pour continuer d'essayer de régler les soucis liés -> problème trouvé : je n'ai pas encore envoyé les paramètres présents sur le formulaire dans la requête
08.03.2021	11	35 Retrait du stock	En cours	1:00	début de la mise en place du formulaire de location avancée du code et des requêtes propres à la location d'un objet -> actuellement, une erreur de DateTime est présente dans le contrôleur
08.03.2021	11	51 Rapport de pré-TPI		0:25	Avancée du rapport : ajout de X.Carrel aux personnes consultées et rédaction des problèmes rencontrés
08.03.2021	11	37 Fonction recherche : mot-clé	En cours	0:40	implémentation de la requête de recherche par mot-clé liaison de l'action du bouton dans le redirecteur
08.03.2021	11	39 Recherche : délai dépassé	En cours	0:40	mise en place de la requête pour trouver tous les objets loués avec délai dépassé liaison de la requête sur le redirecteur
10.03.2021	11	35 Retrait du stock	En cours	1:20	finition du formulaire, tous les éléments sont désormais placés les premières liaisons entre les différentes méthodes ont été faites, les radiobuttons sont fonctionnels reste à finaliser les méthodes, les requêtes et les interactions
11.03.2021	11	34 Ajout au stock	En cours	1:00	mise en place du formulaire de retour d'objet dans le stock premier redirecteur
11.03.2021	11	35 Retrait du stock	En cours	0:20	avancées des requêtes propres au retrait d'un élément au stock -> il faut absolument que je mette en place les trigger dans ma base de données, pour que ça fonctionne correctement

Ceci est le journal de travail de la semaine 12 de 2021. Il s'agit de la semaine n°6 du projet, du 15 mars 2021 au 19 mars 2021. La raison pour laquelle le journal est aussi court vient du fait qu'une formation à l'extérieur a été suivie du mercredi au samedi de cette semaine. Ainsi, seul le lundi contenait des heures mises à disposition pour avancer sur le TPI.

L'analyse présente de ce fait une erreur au tout début, comme l'intégralité des heures de cette semaine ont été prises en compte pour effectuer la répartition des tâches.

Date	Semaine	Activité	Avancée	Heures	Remarques
15.03.2021	12	36 Génération de pdf	En cours	0:30	mise en place du formulaire pour le choix de pdf voulu liaison des boutons souhaités sur le redirecteur
15.03.2021	12	36 Génération de pdf	En cours	1:00	recherche sur l'enregistrement automatique de PDF avec le C#
15.03.2021	12	30 Base du projet C# (classe ... )		1:15	recherche sur la manière d'afficher des données dans la page -> le souci se pose toujours, je n'arrive pas avoir de données récupérées depuis la base de données visibles
15.03.2021	12	38 Historique de location	En cours	0:20	réflexion sur la meilleure manière de faire un log, aucune idée pertinente. En suspens pour le moment, il reste du boulot sur les autres éléments
	0				

Ceci est le journal de travail de la semaine 13 de 2021. Il s'agit de la semaine n°7 du projet, du 22 mars 2021 au 26 mars 2021. Il s'agit de la dernière semaine « pleine » du projet, le rendu étant prévu le lundi 29 mars 2021.

Date	Semaine	Activité	Avancée	Heures	Remarques
22.03.2021	13	30 Base du projet C# (classe		1:00	recherche de comment afficher les objets sur la page principale
22.03.2021	13	41 Tests unitaires		0:30	avancée des tests unitaires pour les différentes fonctions
22.03.2021	13	30 Base du projet C# (classe ... )		0:10	discussion et recherche avec J.Ithurbide, pour trouver une solution viable pour afficher plusieurs colonnes de données sur la page principale
15.03.2021	12	52 Mode d'emploi utilisateur	En cours	1:00	début de la mise en place du mode d'emploi utilisateur le visuel n'est pas encore en place, mais l'idée globale est présente
24.03.2021	13	41 Tests unitaires	Terminé	0:40	Tous les tests ayant pu être réalisé le sont, ils sont désormais en place
24.03.2021	13	51 Rapport de pré-TPI		0:15	Les tests unitaires qui ont été réalisés ont été documentés. Ceux qui n'ont pas pu être faits ont également été documentés
24.03.2021	13	34 Ajout au stock	Terminé	0:10	La méthode permettant d'ajouter un ajout au stock fonctionne correctement, le souci qu'il restait était un oubli d'une ouverture de connexion pour l'interaction avec la base de données
24.03.2021	13	35 Retrait du stock	Terminé	0:30	Le retrait des objets (location) fonctionne désormais. Cette partie est également fonctionnel. Le souci encore présent est que visuellement, rien ne change sur l'application
24.03.2021	13	37 Fonction recherche : mot-	En cours	0:05	Mise en place de la méthode pour la requête servant à effectuer la
25.03.2021	13	30 Base du projet C# (classe		0:10	modifications des visuels pour corriger les petits soucis
25.03.2021	13	51 Rapport de pré-TPI		0:30	vérification du rapport avec la grille d'évaluation, afin de compléter les éléments manquants. Même si l'application n'est pas finie, la documentation est en place et aussi complète que possible
25.03.2021	13	43 Tests fonctionnels	En cours	0:50	tests de l'application -> tous les essais possibles sont faits, afin de trouver tous les endroits où des erreurs existent encore. EDIT : un oubli de ma part -> je n'ai pas fait de captures d'écran, il va falloir refaire les tests pour les ajouter au rapport
26.03.2021	13	55 -		0:10	discussion avec C.Egger, chef de projet. Cela a porté sur l'avancée du projet, la date de rendu. Rendez-vous pour une démonstration
26.03.2021	13	52 Mode d'emploi utilisateur	Terminé	0:40	rédaction d'un mode d'emploi utilisateur. Le mode d'emploi montre le fonctionnement de l'application dans son ensemble, même les éléments qui ne fonctionnent pas pour le moment
26.03.2021	13	53 Procédure d'installation	En cours	0:25	recherche sur comment créer un exécutable pour un projet. Les recherches ont été infructueuses et vu l'état de l'application (incomplète,...), rendre un exécutable de cela semble inutile
26.03.2021	13	31 Script création BDD		1:30	avec l'aide F.Andolfatto, il a finalement été possible de corriger une erreur majeure dans la base de la conception du projet. Cela permet désormais d'afficher des données concrètes sur la première page, même si elles sont partielles

### 5.3 Mode d'emploi utilisateur

L'application fonctionne sur le principe que le client n'a pas d'accès aux données. Seuls les employés y ont accès. Le numéro d'employé, demandé à chaque action, permet de savoir qui a effectué quelle tâche.

Ajouter un objet (bouton 1)

- Cliquer « Ajouter un objet » (bouton 1)
- Remplir les champs disponibles
  - ➔ Type : type de l'objet, voir liste
  - ➔ Nom : nom de l'objet
- Valider avec le bouton « Ajouter » (cf. Figure 2)

Louer un objet (bouton 2)

- Cliquer « Louer un objet » (bouton 2)
- Remplir les champs disponibles
  - ➔ Nom : nom de l'objet
  - ➔ Numéro : numéro de client
  - ➔ Calendrier : sélectionner la date à laquelle l'objet doit être rapporté **AU PLUS TARD** (cf. Figure 3)
- Valider avec le bouton « Louer »

Retourner un objet (bouton 3)

- Cliquer « Retourner un objet » (bouton 3)
- Remplir les champs disponibles
  - ➔ Numéro de location
- Vérifier s'il s'agit du bon objet avec « Afficher le détail » (cf. Figure 4)
- Valider avec le bouton « Retourner »

Trier les données (bouton 4 et 7)

- Cliquer « Délai dépassé » (bouton 4)
  - ➔ Seuls les objets loués et qui auraient déjà dû être retournés s'affichent

Afficher l'historique de location

- Sélectionner un objet sur l'application
- Cliquer « Historique de location » (bouton 5)
  - ➔ Un formulaire s'ouvre, avec l'historique des actions ayant eu lieu pour l'objet en question

Créer un PDF

- Cliquer « Générer un PDF » (bouton 6)
- Sélectionner « Objets loués » (bouton central) ou « Objets disponibles » (bouton de droite)
  - ➔ Seuls les objets ayant l'état « loué » ou « disponible » seront sélectionnés
- Choisir l'emplacement d'enregistrement et renommer le fichier
- Valider la génération du PDF

Figure 13 : visuel global de l'application

Figure 14 : ajout d'un objet

Figure 15 : location d'un objet

Figure 16 : retourner un objet

- Entrer un mot-clé
- Cliquer sur « Rechercher » (bouton 7)
  - ➔ Seuls les objets où le terme/bout de mot se trouve dans le nom ou la description s'affiche

Figure 17 : sélectionner le type de

## 5.4 Manuel d'installation

Le projet n'étant pas terminé, aucun exécutable n'a été créé. La démarche qui suit permet de mettre en place le projet et de pouvoir le tester. Il ne s'agit aucunement d'une solution fonctionnelle pour commercialiser/utiliser réellement l'application.

- Installer MySQLWorkbench
  - o Configurer une connexion localhost, sur le port 3306, avec comme login root et comme mot de passe root
  - o Exécuter les scripts createDatabase.sql et insertData.sql
- Installer VisualStudio 2019
  - o Ouvrir le fichier « locationMateriel.sln » situé dans le dossier locationMateriel avec visualStudio
- Dans la solution
  - o Faire un clic droit sur le nom de la solution dans visualStudio -> gérer les paquets Nuget
    - Ajouter les paquets MySql.Data et iTextSharp
- Démarrer le projet en appuyant sur le bouton démarrer

## 5.5 Archives du projet

L'archive du projet se situe dans un dossier archivé .zip. Il contient :

- Le rapport de projet en format .PDF
- Le journal de travail en format .PDF
- L'application en elle-même, qui doit être lancée avec Visual Studio (vu l'avancée du développement, cela ne fait pas de sens de passer à une version exécutable)
- Les deux scripts permettant de créer la base de données
- Le mode d'emploi en format .PDF
- Le manuel d'installation en format .PDF