

A thick dark blue vertical bar runs down the left side of the page. A blue arrow-shaped banner points to the right, containing the date. In the bottom left corner, there are several thin, curved, light blue lines that sweep upwards and to the right.

01/02/2021

# Location de matériel

Gestion de Stocks

Paola Costa

CENTRE PROFESSIONNEL DU NORD VAUDOIS

## Table des matières

1	Analyse préliminaire .....	2
1.1	Introduction .....	2
1.2	Objectifs.....	2
1.3	Planification initiale .....	3
2	Analyse / Conception.....	4
2.1	Concept .....	4
2.1.1	Objectifs.....	4
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.....	8
2.4	Diagrammes .....	9
2.4.1.1	Diagramme de classe .....	9
2.4.1.2	Diagramme de flux.....	10
2.5	Stratégie de test.....	11
2.6	Risques techniques .....	11
2.7	Planification définitive .....	12
2.8	Infrastructure .....	13
2.8.1	Matériel hardware et système d'exploitation .....	13
2.8.2	Outils logiciels.....	13
3	Réalisation.....	13
3.1	Dossier de réalisation .....	13
3.2	Description des tests effectués .....	13
3.3	Erreurs restantes .....	14
3.4	Liste des documents fournis .....	14
4	Conclusions .....	14
5	Annexes.....	15
5.1	Résumé du rapport du TPI / version succincte de la documentation .....	15
5.2	Sources – Bibliographie.....	15
5.2.1	Livres utilisés .....	15
5.2.2	Pages internet consultées .....	15
5.2.3	Personnes consultées .....	15
5.3	Journal de travail .....	15
5.4	Manuel d'Installation .....	15
5.5	Manuel d'Utilisation.....	15
5.6	Archives du projet.....	15
5.7	Table des illustrations .....	16

# **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 complétion 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 distantes.
- 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.

[illegible]

## **2 Analyse / Conception**

### **2.1 Concept**

#### **2.1.1 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 complétion 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.

L'application devra également répondre à plusieurs points techniques spécifiques. Ce sont les suivants :

- 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.

Enfin, afin de rendre l'application accessible, une procédure d'installation et de mise en service seront également fournies.

## 2.1.2 Modèles de données

### 2.1.2.1 Modèle de données conceptuel

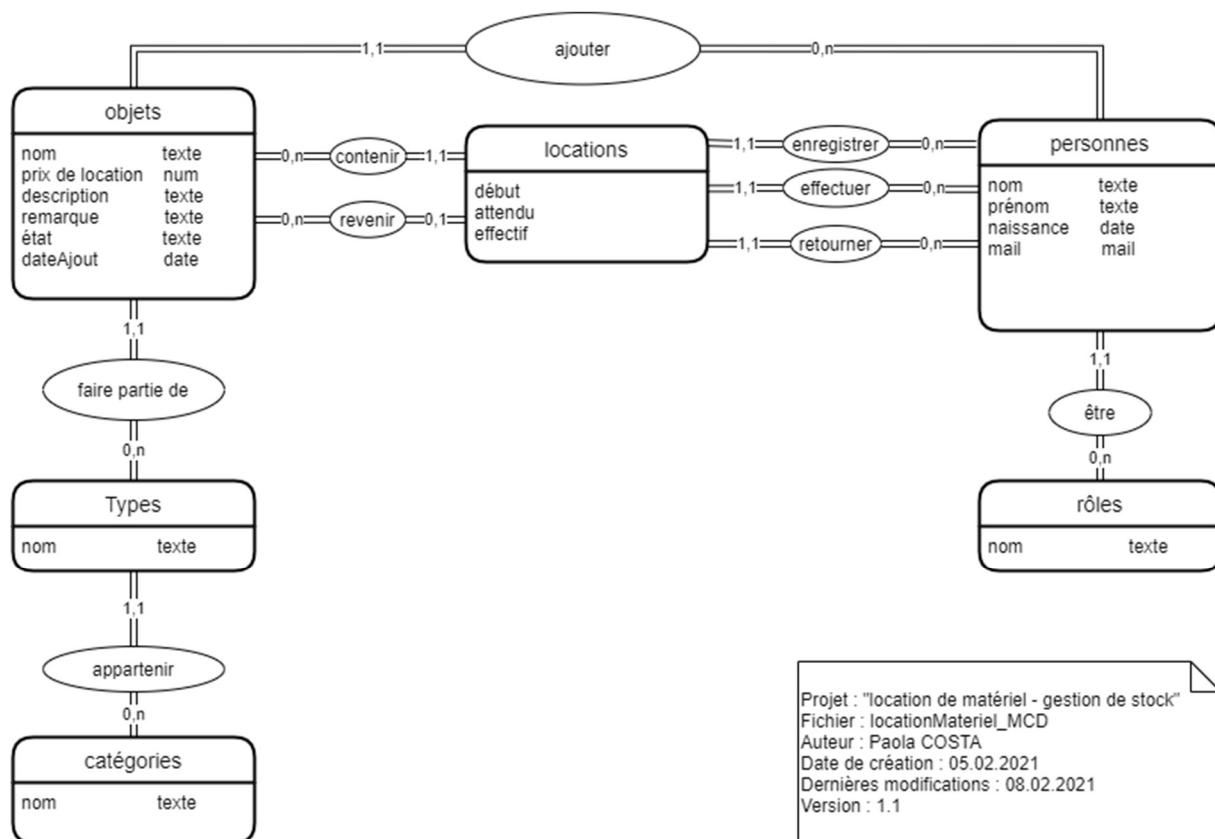


Figure 1 : modèle conceptuel de données

## 2.1.2.2 Modèle de données logique

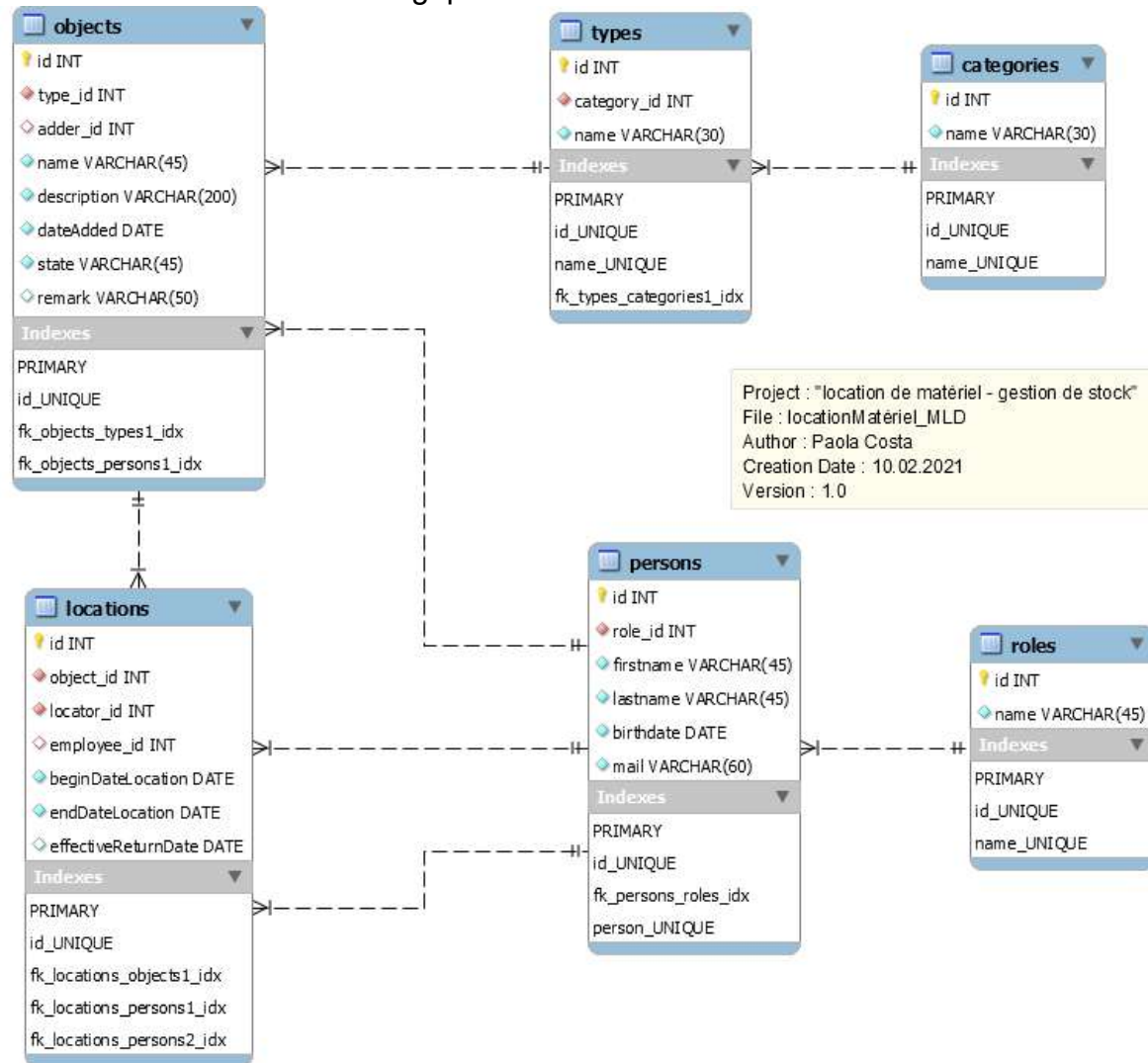


Figure 2 : modèle logique de données

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 classer 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.

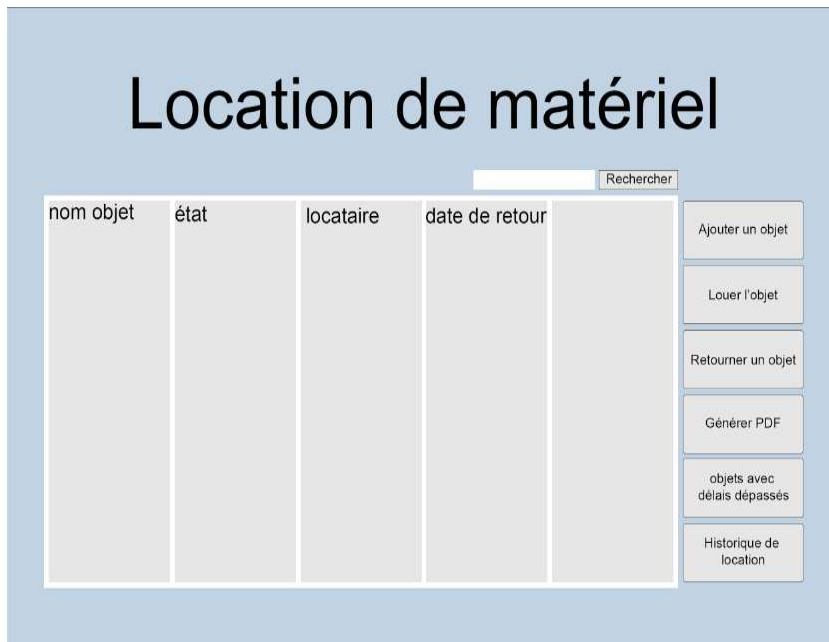


Figure 3 : maquette – vision globale de l'application

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.

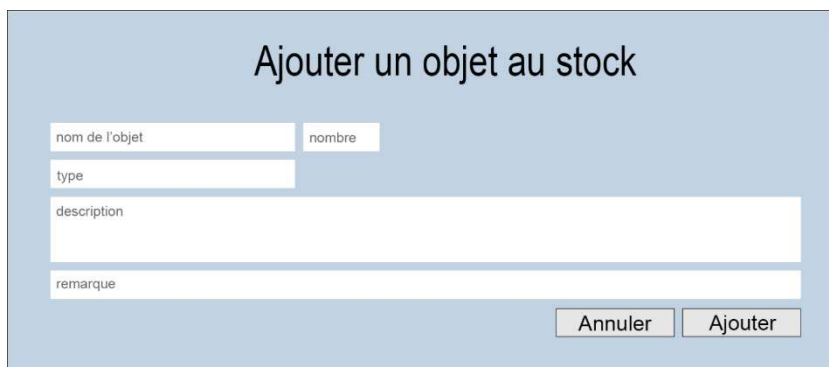


Figure 4 : maquette – formulaire d'ajout d'objet

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.

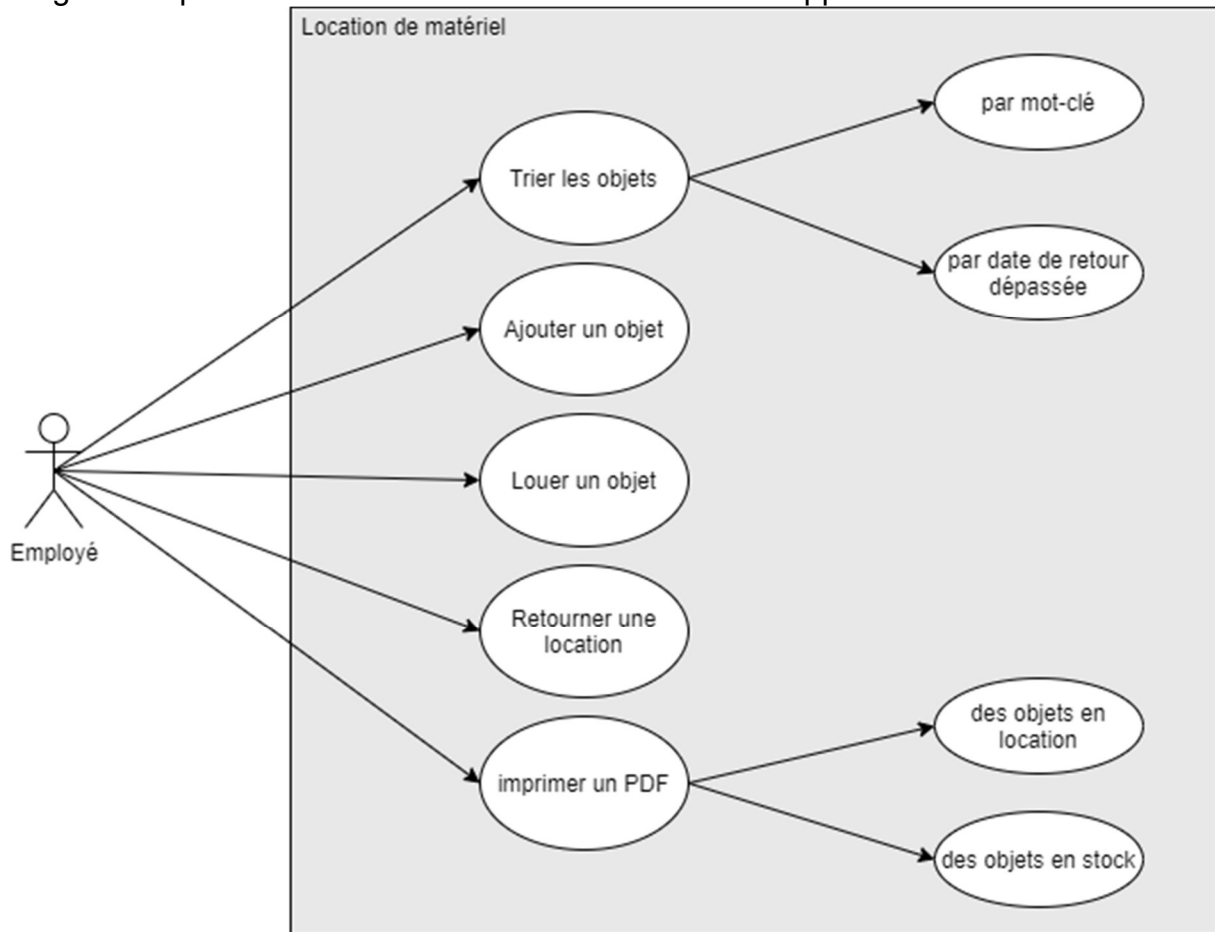


Figure 5 : use cases

### 2.3.2 Scénarii

## 2.4 Diagrammes

### 2.4.1.1 Diagramme de classe

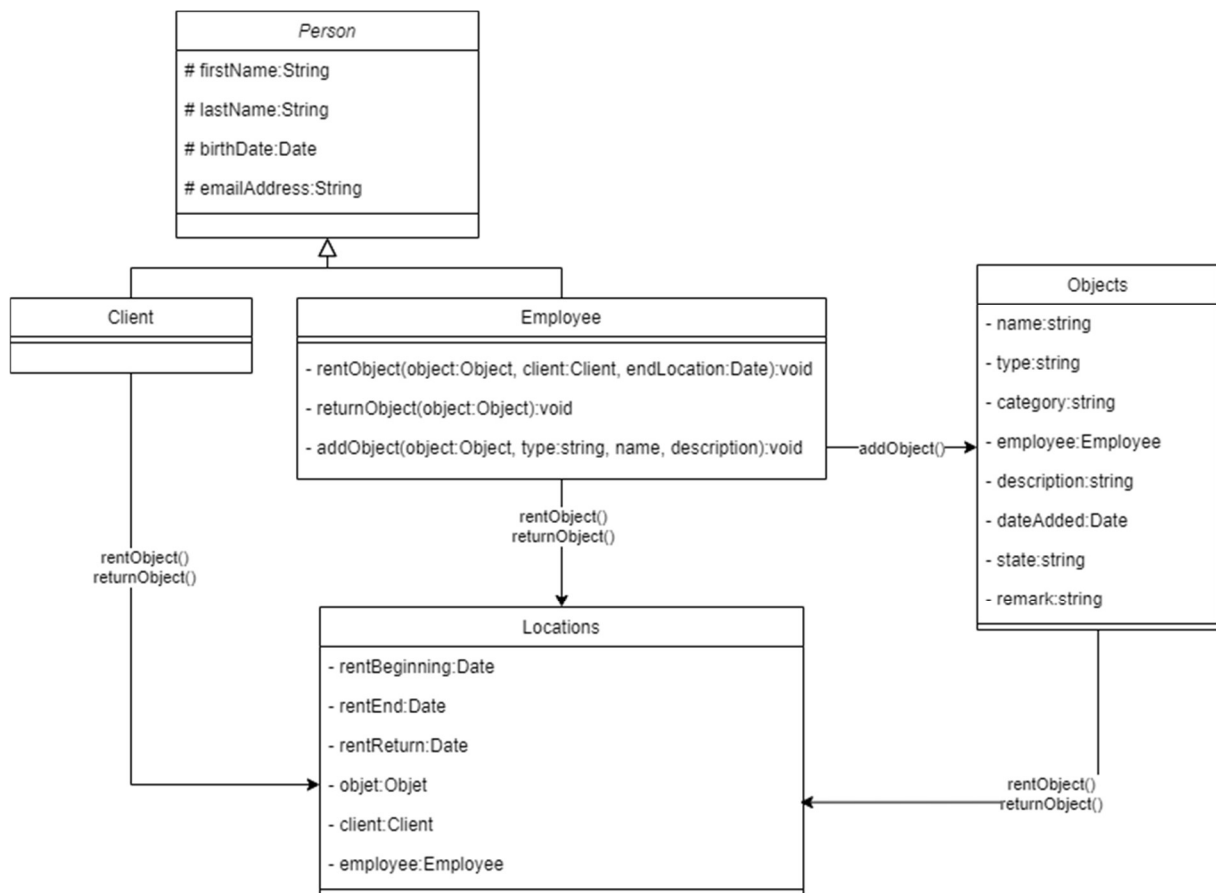


Figure 6 : diagramme de classe

## 2.4.1.2 Diagramme de flux

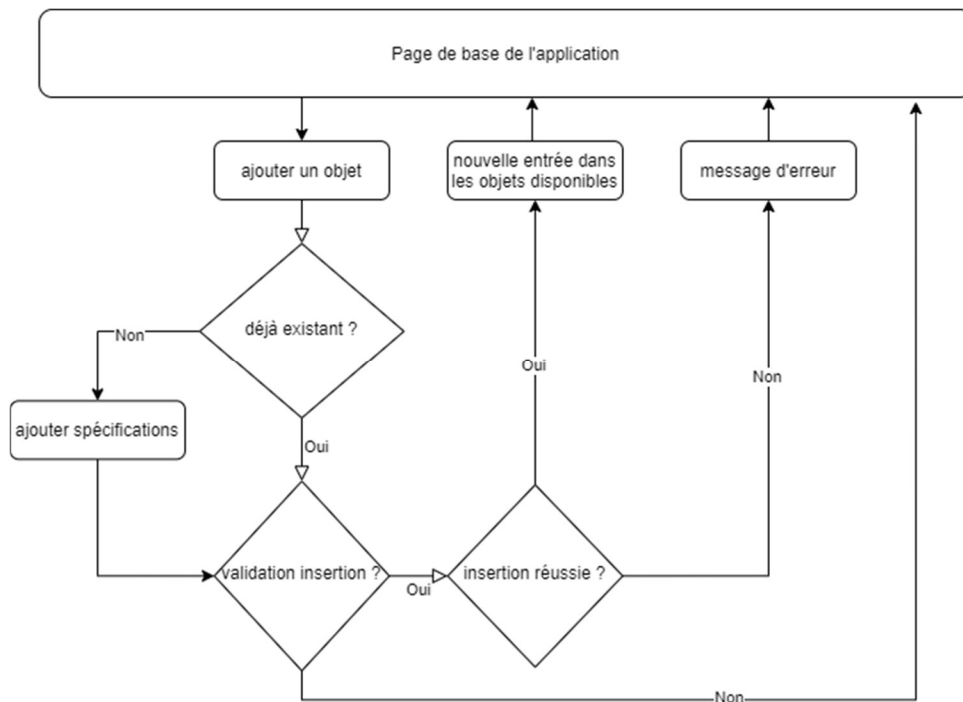


Figure 7 : diagramme de flux – ajouter un objet

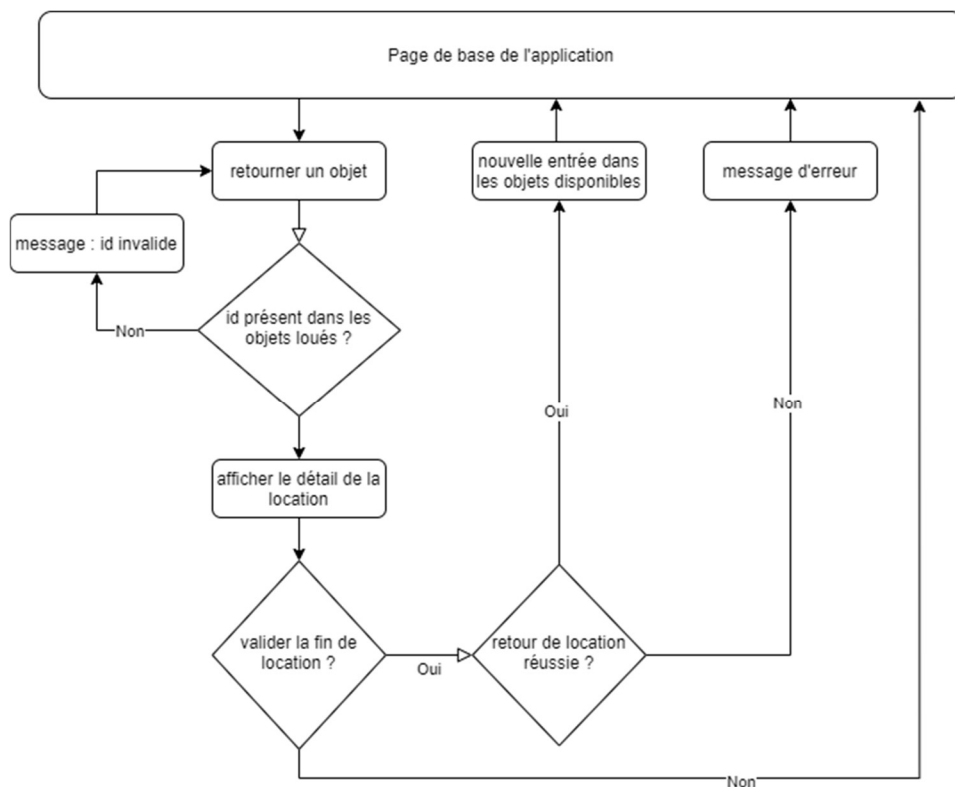


Figure 8 : diagramme de flux – retourner un objet

## 2.5 Stratégie de test

Afin de tester l'application, trois types de tests vont être effectués : les tests unitaires, les tests d'intégration 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. Lors de la liaison de cette fonctionnalité avec une autre, un test d'intégration sera fait. Les tests fonctionnels consisteront à vérifier que les fonctionnalités intégrées à l'application fonctionnent effectivement et retournent ce qu'elles doivent.

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 d'intégration et 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 et d'intégration 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, il existe un risque non-négligeable sur la non-résolution d'un souci d'implémentation d'une fonction spécifique à un élément. 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

- Révision de la planification initiale du projet :
- - planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.
  - partage des tâches en cas de travail à plusieurs.
- 
- Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.

## 2.8 Infrastructure

- 
- Fournir tous les documents de conception:
- - le choix du matériel HW
  - le choix des systèmes d'exploitation pour la réalisation et l'utilisation
  - le choix des outils logiciels pour la réalisation et l'utilisation
  - site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...
  - bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.
  - programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...
- 
- Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !

### 2.8.1 Matériel hardware et système d'exploitation

Machine à disposition : un ordinateur du CPNV sous Windows10 x64

### 2.8.2 Outils logiciels

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

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

MySQL :

## 3 Réalisation

### 3.1 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...

### 3.2 Description des tests effectués

Pour chaque partie testée de votre projet, il faut décrire:

- les conditions exactes de chaque test
- les preuves de test (papier ou fichier)

- tests sans preuve: fournir au moins une description

### **3.3 Erreurs restantes**

S'il reste encore des erreurs:

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

### **3.4 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...

## **4 Conclusions**

- Développez en tous cas les points suivants:
- - Objectifs atteints / non-atteints
  - Maintien des délais prévus
  - Points positifs / négatifs
  - Difficultés particulières
  - Suites possibles pour le projet (évolutions & améliorations)

## **5 Annexes**

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

### **5.2 Sources – Bibliographie**

#### **5.2.1 Livres utilisés**

#### **5.2.2 Pages internet consultées**

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

Recherche d'informations sur le contenu nécessaire dans les +

#### **5.2.3 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

### **5.3 Journal de travail**

Date	Durée	Activité	Remarques

### **5.4 Manuel d'Installation**

### **5.5 Manuel d'Utilisation**

### **5.6 Archives du projet**

Media, ... dans une fourre en plastique



## 5.7 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 .....	9
Figure 7 : diagramme de flux – ajouter un objet .....	10
Figure 8 : diagramme de flux – retourner un objet.....	10