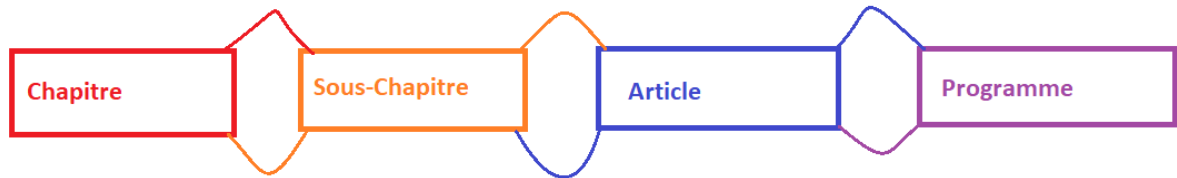
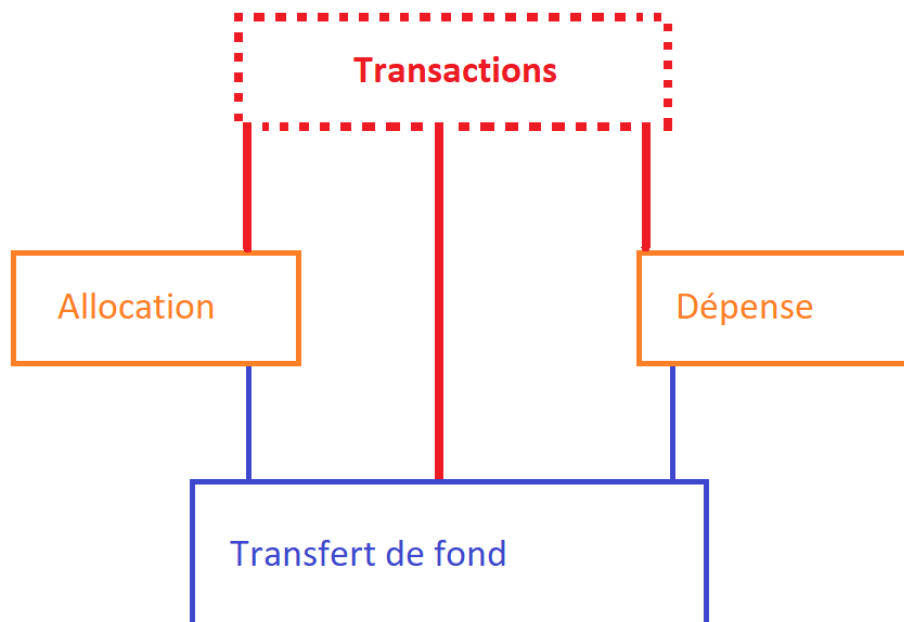


→ Présentation du projet :**- Nomenclature :**

La partie structure de données, ici une nomenclature, se décompose en chapitre qui contiennent des sous-chapitres, qui contiennent des articles, qui peuvent contenir des programmes. Il est possible de parcourir cette arborescence dans les deux sens, par exemple à partir d'un sous chapitre donné, on peut savoir à quel chapitre il est rattaché mais aussi les articles qui dépendent de lui.

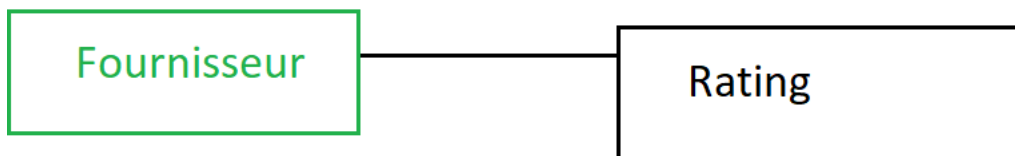
- Transactions :

Pour la partie transactions, nous en avons de trois types. Les deux premières sont les allocations et les dépenses, elles sont assez similaires et représentent un mouvement d'argent dans un sens ou dans l'autre. Le transfert de fond représente un mouvement d'argent entre deux articles ou programmes, il génère donc une allocation ainsi qu'une dépense entre ces deux objets.

- Fournisseurs :

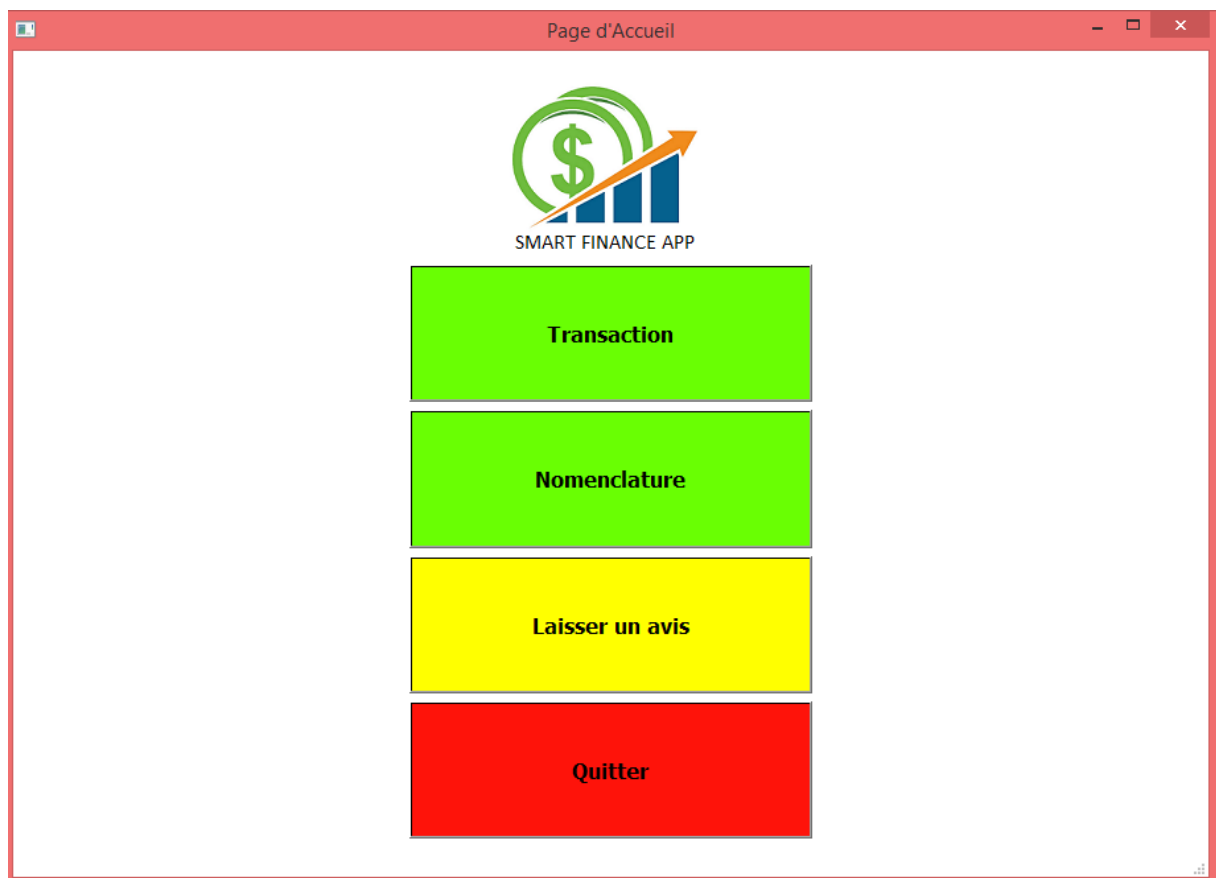


Le fournisseur reçoit l'argent d'une dépense d'un article ou d'un programme. Il est possible d'évaluer un fournisseur.

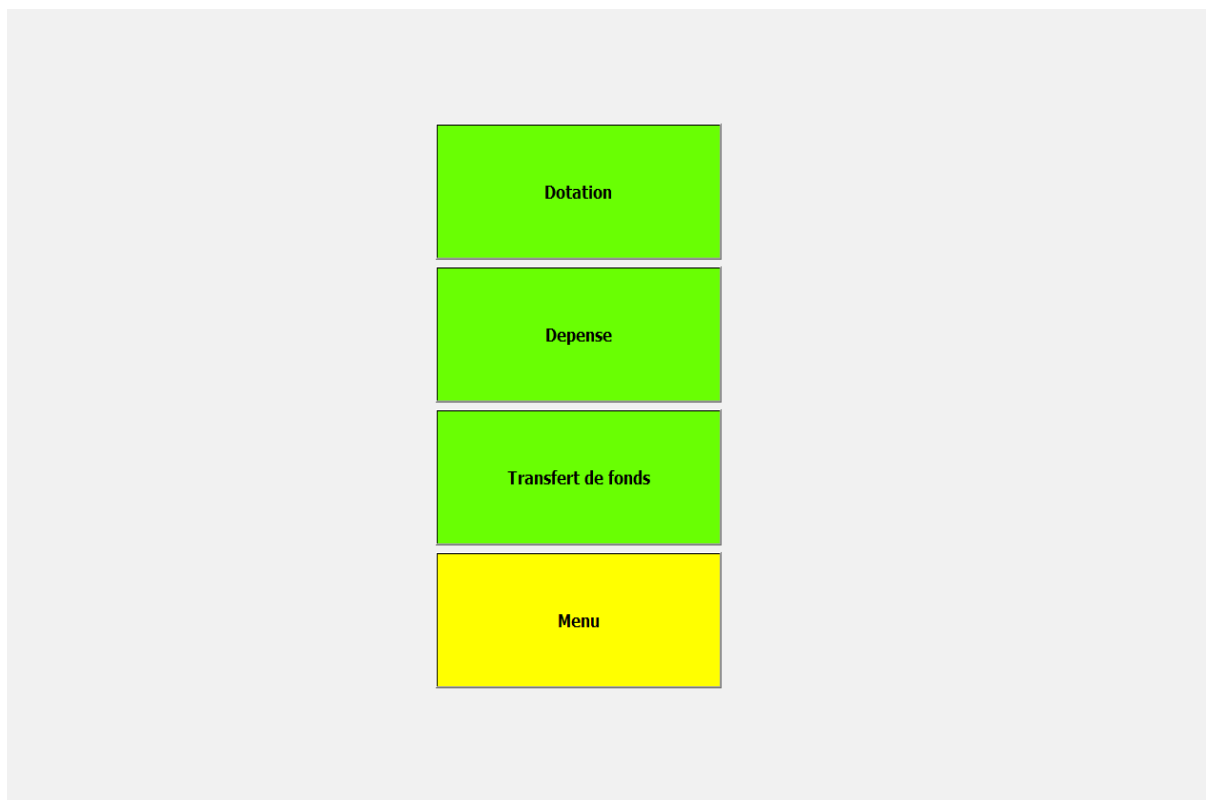


- Base de données :

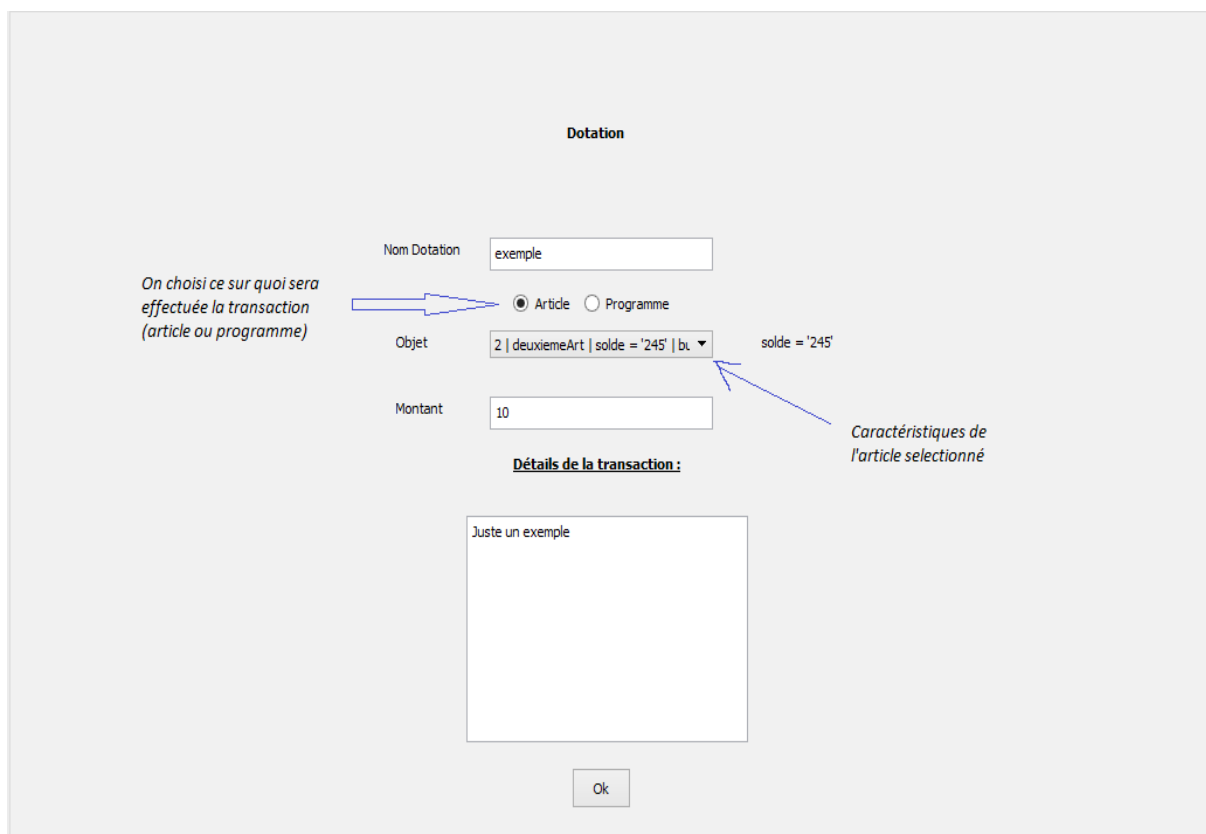
➔ **Implémentation :**



Page d'accueil de l'appli



Page des types de transactions



The form is titled 'Dotation'. It contains the following fields and elements:

- Nom Dotation:** A text input field with the value 'exemple'.
- Article/Programme Selection:** Two radio buttons labeled 'Article' (selected) and 'Programme'.
- Objet:** A dropdown menu showing '2 | deuxiemeArt | solde = '245' | bu.'.
- Montant:** A text input field with the value '10'.
- Annotations:**
 - A blue arrow points from the text 'On choisi ce sur quoi sera effectuée la transaction (article ou programme)' to the radio buttons.
 - Another blue arrow points from the text 'Caractéristiques de l'article selectionné' to the 'Objet' dropdown.
- Détails de la transaction:** A section header followed by a large text area containing the text 'Juste un exemple'.
- Ok Button:** A small button at the bottom center.

Partie dotation

Dotation

Nom Dotation :

☒ Article ☐ Programme

Objet :

Montant :

MessageBox Important

Vous voulez vous imprimer un recapitulatif ?

Juste un exemple

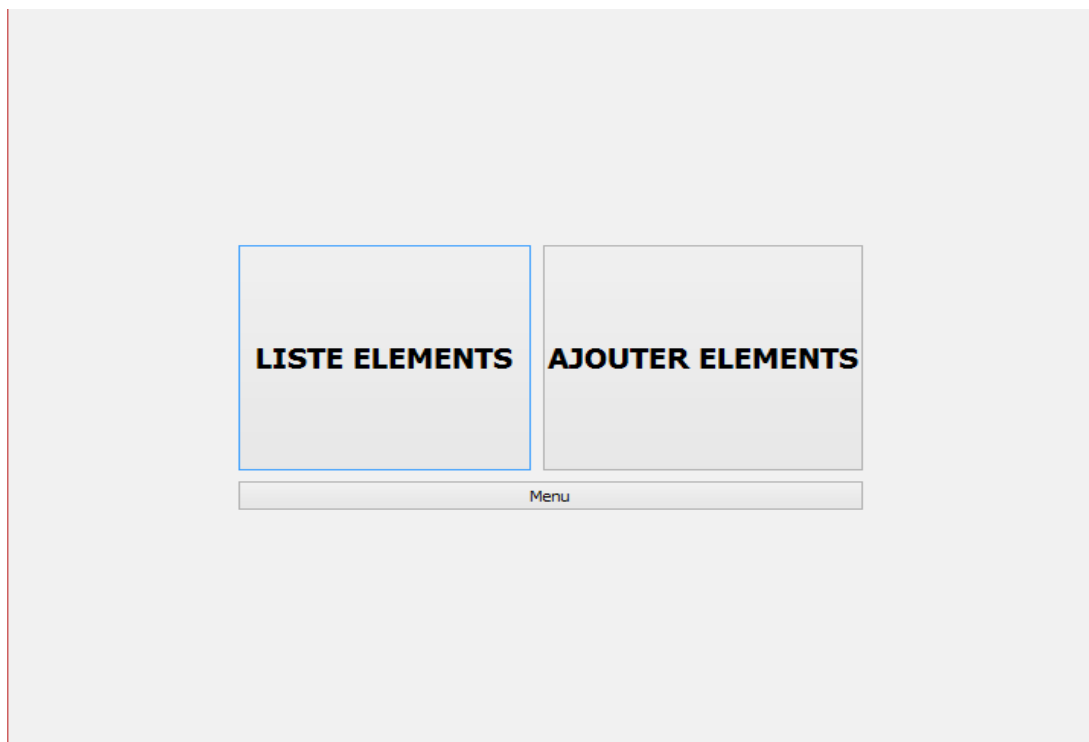
Une Message box pop apres la transaction pour demander si l'user veut imprimer recap

Dialog

Dotation

Nom transaction :	exemple
Montant :	10
Projet Associé :	deuxiemeArt
Description :	Juste un exemple

On arrive sur une page pour imprimer les détails de la transaction



Lorsque nous appuyons sur le bouton « nomenclature », nous arrivons sur cette page

The screenshot shows a light gray background with a red border. At the top left, there is a button labeled 'Retour'. Below it, the page is divided into two main sections: 'AJOUTER' on the left and 'MODIFIER' on the right. Each section contains five buttons stacked vertically, corresponding to the same categories: 'Chapitre', 'SousChapitre', 'Article', 'Programme', and 'Fournisseur'.


Page ou nous pouvons ajouter/modifier les éléments de la nomenclature

Chapitre Ajout

Nom

Description

Done

 Ajout effectué

OK

Ok

Page d'ajout de chapitre, avec une messageBox une fois le chapitre ajouté (meme fonctionnement pour tous les autres éléments de la nomenclature)

Retour

4 | Ayaya

Delete chapitre

Delete sousChap

Delete article

On retrouve notre dans la liste de la nomenclature, avec aucun element dedans

Retour

1 | Chapitre1 Delete chapitre

1 | SousChap1 Delete sousChap

idArticle	nom	
2	deuxiemeArt	255
7	ppp	0

Delete article

La page se met dynamiquement a jour si on trouve des sous_éléments dans l'élément sélectionné

Retour

1 | Chapitre1 Delete chapitre

1 | SousChap1 Delete sousChap

idArticle	nom	
2	deuxiemeArt	255
7	ppp	0

Delete article

idProgramme	nom	
2	c9	9

Delete programme

idTransaction	nom	
3	Laptop	10
4	Important	15
29	aaaa	12
30	aaaa	12
36	azaz	10
53	aeza	15
54	eaze	1

Imprimer recap

Exemple de page avec un élément de chaque

avis

Menu

1 | hello

Avg : 2.6

Avis

Les fournisseurs

Exprimez-Vous

Exemple avis

★ 1 ★ 2 ★ 3 ★ 4 ☆ 5

Ok

La note

Il est possible de laisser un avis sur un fournisseur

Ci dessous la consigne avec :

En vert : Les éléments implémentés sans trop de soucis

En jaune: Les éléments qui ont été implémentés avec une interprétation plus ou moins certaine

En rouge : Les éléments non implémentés

1. Nomenclature :

La nomenclature détermine l'élément sur lequel une transaction sera opérée par exemple l'article à créditer ou à débiter. Elle est structurée en arborescence ou les premiers élément (nœuds racines) contiennent les nœuds feuilles du premier niveau, qui elles contiennent les nœuds feuilles du second niveau etc..

Un exemple d'une arborescence c'est : Les Chapitres (nœuds racines) qui contiennent une liste de sous-chapitres (nœuds feuilles 1er niveau). Les sous-chapitres contiennent une liste d'articles (nœuds feuilles 2nd niveau). Un article contient une liste de programmes.

Il est demandé de réaliser le modèle objet qui permet d'ajouter, de modifier, ou de supprimer un élément de la nomenclature. Un chapitre est caractérisé par son identifiant, son titre, son contenu, et sa description. Pour les nœuds feuilles, ils disposent en supplément de l'id de l'objet hôte. Par exemple, un sous chapitre contient l'id du chapitre auquel il appartient, Un article devra contenir l'id du sous-chapitre etc..

2. Fournisseur :

Il désigne la société qui propose ses prestations à la ville. Par exemple : une entreprise de maintenance de l'état des routes. Il est caractérisé par son id, le nom du gérant, la raison sociale (SSII, personne physique, etc.), description, domaine, année de création, popularité, etc.

On doit pouvoir ajouter, modifier, et supprimer un fournisseur.

3. Transaction :

Une transaction désigne une opération financière menée sur un élément de la Nomenclature comme un article ou un programme. Une transaction peut prendre différentes natures à savoir une transaction dotation, une transaction dépense, une transaction transfert de fond.

3.1 Transaction dotation : Il s'agit d'une opération qui permet de créditer (d'alimenter) le solde d'un élément de la nomenclature comme un article. Ce solde désigne le budget dont dispose l'article sur une période de temps (une année). Il existe plusieurs types de transaction dotations à savoir : (i) Budget initial : qui permet de créditer le solde annuel d'un article ou d'un programme. Un budget initial est caractérisé par son id, année d'exercice financier, sa date de réunion, description, le montant, date rajout (ii) Budget supplémentaire : qui permet de compléter le budget initial en lui rajoutant un montant donné à consommer pendant l'année en cours. Le budget supplémentaire reprend les mêmes caractéristiques.

On doit pouvoir rajouter d'autre modèles de transactions dotations en demandant à l'utilisateur d'introduire leurs attributs. Par exemple : rajouter des dotations location des services de la ville. Dans ce cas-là l'utilisateur introduit les paramètres suivants : année d'exercice financier, date de l'opération, date rajout, type de service, montant. (Il est autorisé de limiter le nombre d'attributs pour une nouvelle dotation à 4 attributs par exemple).

3.2 Transaction Dépense : Il s'agit d'une opération qui permet de débiter le solde d'un élément de la nomenclature comme un article. Une transaction dépense doit être liée à un fournisseur. On peut retrouver plusieurs types de transaction de dépenses chacune décrite par l'utilisateur comme est le cas pour les dotations. (i) Dépense facture proforma : caractérisée par l'id de la transaction, le numéro de la facture proforma, son fournisseur, la description. On doit pouvoir rajouter d'autre modèles de transactions dépenses comme est le cas pour les dotations.

3.3 Transaction Transfert de Fond : Cette opération génère 2 transactions : une dotation et une dépense.

4. Tableau de bord : Une interface graphique sur laquelle on retrouve l'historique des transactions de chaque élément de la nomenclature, ainsi que son solde initial, son solde actuel, et son solde restant.

5. Statistiques et Rapports : Permet d'exporter : (i) les détails d'une transaction, (ii) des historiques des transactions sur une période de temps donnée, (iii) historique des transaction par fournisseur, etc.

Elements en jaune :

1 : Pour ce qui est de la nomenclature, nous n'étions pas sur si elle était un objet au meme titre qu'un chapitre, ou si elle était simplement l'ensemble de l'arbre. Cependant, comme l'exemple d'arborescence donné était conforme au modèle donné (chapitre / sous-chapitre), nous en avons conclu que la nomenclature était simplement l'ensemble de tous ces éléments.

2 : En ce qui concerne les fournisseurs, pour l'implémentation des opérations, nous avons :

- Les dotations qui concernent un article/programme sur lequel une créditation de budget sera effectué.
- Les dépenses où un article / programme verra son solde débité en faveur d'un fournisseur
- Un transfert de fond qui doit implémenter les deux.

Un problème survient alors lors de l'implémentation du transfert. Pour les dépenses, le bénéficiaire de l'argent sera le fournisseur et pour la dotation, le bénéficiaire doit être un article/programme. En effet :

« **Transaction dotation :** Il s'agit d'une opération qui permet de créditer (d'alimenter) le solde d'un élément de la nomenclature comme un article »

« **Transaction Dépense :** Une transaction dépense doit être liée à un fournisseur. »

Il n'y a aucun problème à débiter un article, ni à le créditer. Cependant, lorsqu'on en vient au transfert, qui doit implémenter une dotation et une dépense, il nous est difficile de créditer le solde d'un fournisseur. Par conséquent, les fonctions dépenses et transfert de fond ont été implémentées sur des articles/programme, et non sur des fournisseurs. Il est cependant toujours possible d'utiliser le CRUD sur les fournisseurs, de leur laisser un avis, mais il ne sont pas des éléments de la transaction pour l'instant.

Elements en rouge :

Il a été difficile techniquement de permettre à l'utilisateur d'ajouter de nouveaux modèles de dotation. En effet, entre la gestion de la base de données, de l'interface graphique, des paramètres int/varchar etc., implémenter cette fonctionnalité impliquera d'implémenter QT directement dans le logiciel (Bon on caricature un peu ici, mais dans l'idée, nous n'avons pas trouvé un moyen simple de rendre cela possible).

→ Difficultés rencontrées :

- Connection à la base de données (MYSQL)

Nous avons rencontré de nombreux problèmes de compatibilité avec les drivers qui n'étaient pas ou mal pris en charge pour MYSQL et sommes finalement passé à SQLITE.

- Prise en main de QT

On avait presque l'impression d'aborder un nouveau langage entre le code console écrit au préalable et le code requis par QT, il a fallu passer pas mal de temps le nez dans la documentation de QT pour réussir à appréhender correctement les objets découlant des classes QT.

→ Ressentis personnels :

Antoine : Ce fut déroutant comme épreuve sans cours , mais on a su trouver des tutos utiles et agréables

Gauthier : Cf les difficultés

Marc-Eli : Un sentiment assez mitigé quant à ce projet. Il est certes, toujours gratifiant de mener à bien un projet aussi colossal (à notre échelle).

Cependant, je ne peux m'empêcher de ressentir une certaine frustration à certains niveaux :

1 : Il a été assez laborieux d'utiliser QT (qui est assez lent pour compiler) mais surtout d'utiliser les drivers MYSQL (cf ci dessus les difficultés rencontrées)

2 : QT est un logiciel tellement spécifique avec des Qobjects déjà implémentés qu'il a été difficile de voir un intérêt à implémenter le code réalisé via code block.

3 : Dû au confinement ou pas, les faits sont là : Il y a clairement eu un manque de communication quant aux consignes données pour le projet. Il a été difficile de déchiffrer les consignes données dans un premier temps. Il s'agit

peut être d'une incompréhension de notre part vu que nos camarades auraient, je vous cite, « très bien compris le sujet ».

Encore une fois, il s'agit d'un ressenti personnel, mais le fait d'avoir eu des consignes complémentaires la veille pour le lendemain pour le rendu du main (pas de BDD, main standard code blocks) m'a un peu agacé. Si j'avais su cela plus tôt (car j'étais chargé de l'implémentation de main :)), j'aurais directement amélioré celui préalablement implémenté sur CodeBlocks.

J'ai conscience que ces consignes avaient pour but d'être rassurantes, en nous disant de rester simple, mais elles arrivent cependant un peu tard, sachant qu'on avait déjà initié notre passage sur QT depuis 1 mois en l'absence de nouvelle concernant le cours.

Je tiens tout de même à saluer les délais supplémentaires accordés, compensant en partie les pertes de temps que j'ai cité ci dessus.

Pour conclure, ces épreuves plus ou moins éprouvantes m'auront, je pense, permis de progresser, bien qu'il aurait été bienvenu de pouvoir apprendre à se servir de QT directement, sans passer par un logiciel tiers qui nous est, au final, quasi inutile pour notre sujet. En outre, une meilleure communication au niveau de réponses aux emails/communication des consignes aurait été très appréciée et salutaire au bon déroulement du projet.