

---

# Rapport de projet

## Génie logiciel : KitBox

---

Abah Emene, Calugar Finias, Souidi Yassin,  
Taymans Landry, Vandermeersch Thomas

Juin 2020

# Table des matières

Table des matières.....	2
1 Introduction .....	3
2 Analyse des besoins.....	3
2.1 Exigences spécifiques à l'application.....	3
2.2 Exigences spécifiques à un meuble.....	4
2.3 Diagramme des cas d'utilisation.....	5
2.4 Diagramme d'activité .....	9
2.5 Glossaire.....	10
3 Explication du fonctionnement du logiciel.....	11
3.1 Interface client.....	11
3.2 Interface magasinier.....	17
4 Justification des choix architecturaux et technologiques .....	21
4.1 Choix d'architecture.....	21
4.2 Choix technologiques.....	21
5 Tous les diagrammes et documents réalisés .....	23
5.1 Explication succincte du diagramme de classe .....	23
5.2 Schéma relationnel.....	24
5.3 Diagrammes de séquence.....	24
6 Conclusions .....	28
6.1 Objectifs .....	28
6.2 Pistes d'améliorations.....	28
6.3 Collaboration dans le groupe.....	28

# 1 Introduction

Ce rapport est réalisé dans le cadre du cours de "Projet Génie Logiciel" qui nous met dans la situation de développeurs devant répondre aux besoins des clients représentés ici par nos professeurs.

Ce projet consiste en la programmation d'une application en C# permettant la création d'une commande et la gestion des stocks d'un magasin de conception de meubles afin de nous plonger dans une situation du milieu professionnel tout en travaillant en équipe.

Les prochains points vous permettront de comprendre plus amplement ce merveilleux projet que vous pouvez aussi retrouver au lien GitHub suivant :

« <https://github.com/ThomasVECAM/KitBox-Project> »

## 2 Analyse des besoins

### 2.1 Exigences spécifiques à l'application

Avant de commencer la programmation, il est important de se pencher sur les exigences qui nous permettront de mieux comprendre les tenants et les aboutissants du projet.

Liste des exigences :

- L'application doit permettre à l'utilisateur la composition de meubles tout en personnalisant ceux-ci selon les possibilités qu'on retrouve dans les exigences spécifiques aux meubles.
- L'application doit permettre à l'utilisateur d'effectuer une facture récapitulative de la commande.
- L'application doit prévenir au plus tôt l'utilisateur de la disponibilité des pièces dans le stock.
- L'application doit pouvoir proposer régulièrement des commandes automatiques de réapprovisionnement de stock qui seront à valider/ à modifier par le magasinier.
- L'application doit être facile et agréable à utiliser.
- L'application doit afficher un récapitulatif des meubles conçus et une facilité de duplication ou de modification de ceux-ci.
- L'application doit afficher le prix par casier au plus tôt ainsi que le prix total par meuble.

— Le magasinier doit être le seul à pouvoir accéder à l'interface magasinier et à effectuer les fonctionnalités suivantes :

- a) accéder à une liste de commandes non clôturées
- b) accéder à l'historique des commandes clôturées
- c) imprimer une facture lorsqu'un paiement est effectué
- d) afficher une liste complète des pièces composant la commande payée
- e) confirmer manuellement le paiement d'un client
- f) informer le magasinier des pièces non disponibles
- g) modifier le stock manuellement
- h) valider et/ou modifier une commande automatique pour le fournisseur
- i) créer des commandes fournisseur

## 2.2 Exigences spécifiques à un meuble

— Un meuble est composé au maximum de 7 casiers et au minimum de 1 casier

— Un casier est composé de :

- a) 4 tasseaux verticaux
- b) 2 traverses AV (avec 2 rainures pour portes)2
- c) 2 traverses AR (avec 1 rainure pour paroi)
- d) 4 traverses GD (avec 1 rainure pour paroi)
- e) 2 panneaux HB (déposés sur les traverses)
- f) 2 panneaux GD (glissés dans les rainures des tasseaux et traverses)
- g) 1 panneau AR (glissé dans les rainures des tasseaux et traverses)
- h) 2 portes (éventuellement) avec 2 coupelles pour l'ouverture (non disponible pour les portes en verre)

— Il existe la possibilité de choix de couleur pour les cornières, les portes et les panneaux des casiers

— Les longueurs des cornières sont standardisés mais peuvent être coupés à la taille désiré

— Chaque casier peut avoir une couleur différente

— Un casier comporte deux portes de même couleur

— Toutes les caractéristiques dimensionnantes des composants d'un meuble se font à partir d'un catalogue. Des dimensions personnalisées ne peuvent pas être choisies.

## 2.3 Diagramme des cas d'utilisation

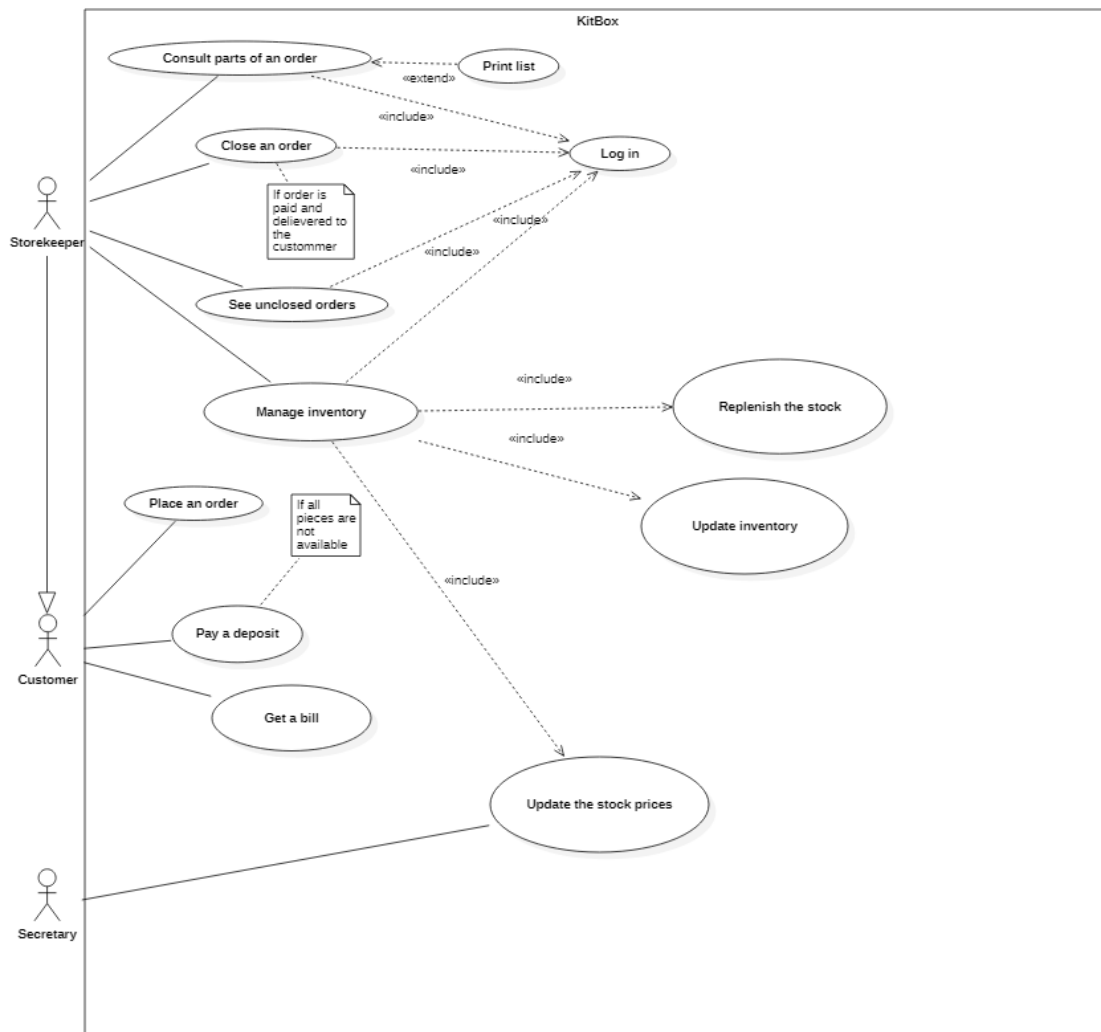


Figure 1

- **Case:** Consult parts of an order

**Objectif :** L'objectif est de permettre au magasinier de voir les pièces d'une commande (afin de les rassembler et les remettre au client).

### Précondition :

- La commande doit exister.
- Le magasinier est connecté.

**Garantie de succès :** Le magasinier voit la liste des composants.

**Flot :**

- Le magasinier recherche la commande du client dans le système.
- La liste des pièces de la commande est affichée.

- **Case** : Print list

**Objectif** : L'objectif est de permettre au magasinier d'imprimer la liste des composants d'une commande client.

**Précondition** : La liste des composants est disponible

**Garantie de succès** : La liste est imprimée.

**Flot** : Le magasinier demande au système d'imprimer la liste

- **Case** : Log in

**Objectif** : L'objectif est de permettre au magasinier ou à la secrétaire de se connecter afin de réaliser des actions administrateurs

**Garantie de succès** : L'utilisateur reçoit les droits administrateurs.

**Flot** :

- L'utilisateur entre son nom d'utilisateur et son mot de passe.
- Mot de passe correct : l'utilisateur est informé du succès de sa connexion.

**Flot alternatif** :

- L'utilisateur entre son nom d'utilisateur et son mot de passe.
- Mot de passe incorrect : le système informe que les données sont incorrectes.

- **Case** : Close an order

**Objectif** : L'objectif est de permettre au magasinier de clôturer une commande une fois que le client a reçu toutes ses pièces.

**Précondition** :

- La commande doit exister.
- Le magasinier est connecté.
- La commande est payée et donnée au client.

**Garantie de succès** : La commande est clôturée.

**Flot** :

- Le magasinier clôture la commande.
- Le magasinier recherche la commande du client dans le système.

- **Case** : See unclosed orders

**Objectif** : L'objectif est de permettre au magasinier de voir quelles commandes ne sont pas terminées.

**Précondition** : Le magasinier est connecté.

**Garantie de succès** : Le magasinier voit les différentes commandes non clôturées.

**Flot** : Le magasinier demande au système d'afficher les commandes non clôturées et celle-ci lui sont affichées à l'écran.

- **Case** : Manage inventory

**Objectif** :

- L'objectif est de permettre au magasinier de mettre à jour les stocks.
- Réapprovisionner le stock.
- Modifier le stock.
- Créer un nouveau composant.

**Précondition** : Le magasinier est connecté.

**Garantie de succès** : Le stock est mis à jour.

- **Case**: Replenish the stock

**Objectif** : L'objectif est de permettre au magasinier de réapprovisionner le stock sur base des prix et délais de livraison.

**Précondition** : Le magasinier reçoit une proposition de commande de la part du système (cette proposition se base sur les prix et délais de livraison inscrit dans la base de données).

**Garantie de succès** :

- Les pièces des commandes en attente (car manque de stock) sont commandées.

**Flot** :

- Le magasinier reçoit une proposition de commande faite par le système.
- Le magasinier confirme la commande.

**Flot alternatif** :

- Le magasinier reçoit une proposition de commande faite par le système.
- Le magasinier modifie la proposition de commande.
- Le magasinier confirme la commande.

- **Case** : Update inventory

**Objectif** : L'objectif est de permettre au magasinier de modifier le stock en cas de casse de matériel (ou autre).

**Garantie de succès** : Le stock est mis à jour.

**Flot** :

- Le magasinier recherche une pièce dans la base de données.
- Le magasinier modifie la valeur en stock de cette pièce.
- Le magasinier confirme les modifications.

- **Case** : Place an order

**Objectif** : L'objectif est de permettre au client de commander les meubles qu'il souhaite et les caractéristiques de ceux-ci.

**Garantie de succès** : Une commande valide est formée.

**Flot** :

- Le client entre ses informations personnelles (nom, Prénom, email, adresse).
- Le client ajoute un ou plusieurs meubles de la manière suivante :
  - Il ajoute 1 à 7 "box" dont il choisit les différentes caractéristiques (dimensions, couleurs, portes)
  - Le système lui propose des cornières compatibles avec la couleur des "box".
- Le client confirme la commande.

• **Case : Pay a deposit**

**Objectif :** Lorsque toutes les pièces ne sont pas disponibles le client est invité à payer un acompte et revenir la semaine suivante.

**Précondition :**

- Une commande est créée et existe.
- Toutes les pièces de la commande ne sont pas disponibles.

**Garantie de succès :** L'acompte est versé.

**Flot :**

- Le système calcule et affiche la valeur de l'acompte a versé.
- (- Le client paye l'acompte)

• **Case : Get a bill**

**Objectif :** L'objectif est de donner au client une facture.

**Précondition :** Une commande doit exister.

**Garantie de succès :** La facture est imprimée.

**Flot :**

- Le système, sur base de la commande compose une facture.
- Le système affiche/imprime la facture.

• **Case : Update the stock prices**

**Objectif :** L'objectif est de permettre à la secrétaire de mettre régulièrement à jour les prix de la base de données sur base des catalogues fournisseurs.

**Précondition :** La secrétaire est connectée.

**Garantie de succès :** Les prix du catalogue sont mis à jour.

**Flot :**

- Rechercher un produit dans la base de données.
- Modification des prix de ce produit sur base des catalogues fournisseurs.
- Confirmer et sauver les modifications.



## 2.4 Diagramme d'activité

### Order activity diagram

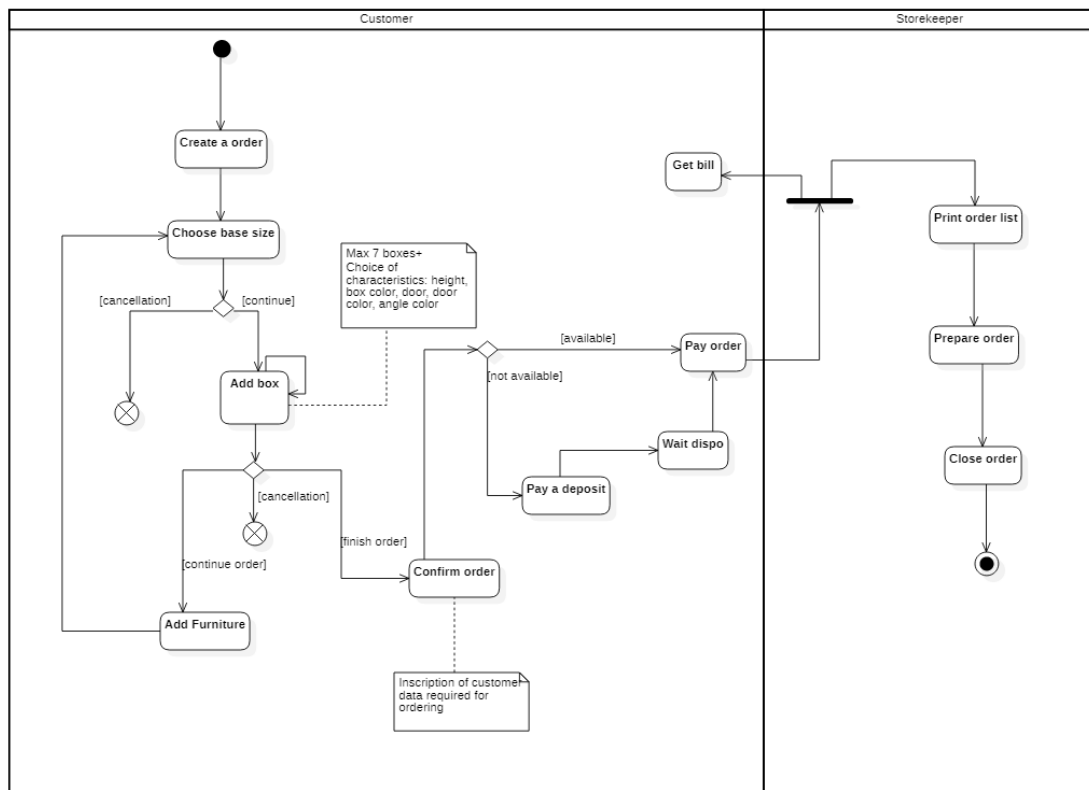


Figure 2

Au lancement de l'application par le client, celui-ci est amené à créer une commande en commençant par choisir la taille de la base du meuble.

Ensuite, il peut soit continuer la commande soit décider de s'arrêter là. Pour la suite de la commande, le client choisi d'ajouter des casiers et de les personnaliser.

Enfin, après avoir fini son premier meuble, il a le choix entre créer un nouveau meuble ce qui le fait revenir à l'étape où il choisit la taille de la base ou de confirmer la commande.

Suite à la confirmation de la commande, celle-ci est soit disponible alors le client paye la commande, soit indisponible alors le client paye un acompte en attendant que les différentes pièces manquantes arrivent pour pouvoir payer la commande.

Pour finir, après que le client ait payé la commande, celui-ci reçoit une facture et le magasinier se charge de l'imprimer, de préparer la commande et de finaliser la commande.

## Stock management activity diagram

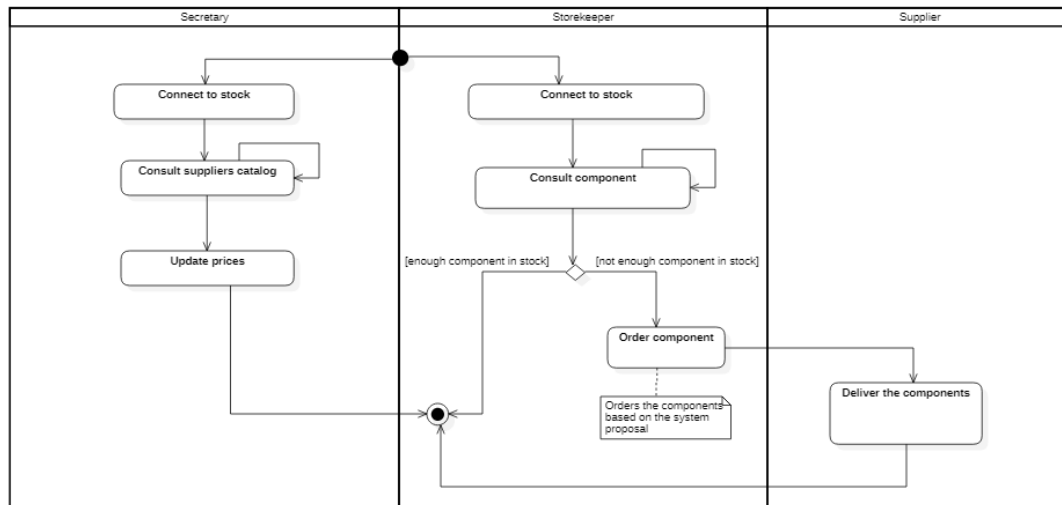


Figure 3

Au lancement de l'application magasinier, celui-ci doit s'identifier afin d'accéder à la plateforme.

Le secrétaire se charge de vérifier les différents catalogues des fournisseurs pour voir si des changements sont à effectuer en ce qui concerne le prix. Le magasinier quant à lui se charge de consulter les composants afin de vérifier s'il y en a assez dans le stock. Dans le cas contraire, il effectue une commande sur base d'une proposition du système

## 2.5 Glossaire

**Utilisateur** : Toute personne, client ou magasinier inclus

**Box** : Un casier

**Casier** : Élément constitutif de l'armoire, comprenant des tasseaux, des traverses, des panneaux, et éventuellement de portes. Les casiers sont empilables

**Cornière en acier** : Profilé en métal en « L » liant les différents casiers superposés

**Coupelle** : Pièce en plastique dans le panneau de porte permettant une ouverture de porte plus aisée

**Traverse** : Pièce en bois horizontale permettant la liaison entre deux panneaux pour rigidifier un casier

**Tasseau** : Pièce en bois verticale à l'intérieur du casier permettant la liaison entre deux panneaux

**HB** : Haut et bas

**AR** : Arrière

**GD** : Gauche et droite

**AV** : Avant

**Commande finalisée** : Commande qui contient un ou plusieurs meubles étant approuvé par le client dans l'application mais sans être forcément payée

**Commande non-clôturée** : Commande finalisée et dont l'acompte a été payé

**Commande clôturée** : Commande finalisée et dont toute la facture a été payé

**Corner Color** : Couleur des cornières

**Add** : Ajouter

**IDE** : Environnement de développement

**Merge** : Mot utilisé dans le "langage GitHub" représentant l'union/la fusion de deux fichiers

**Order** : Commande contenant une liste de meubles, un identifiant et appartenant à une personne

### 3 Explication du fonctionnement du logiciel

#### 3.1 Interface client



*Figure 4*

Appuyer sur start pour accéder à l'interface client et créer des meubles

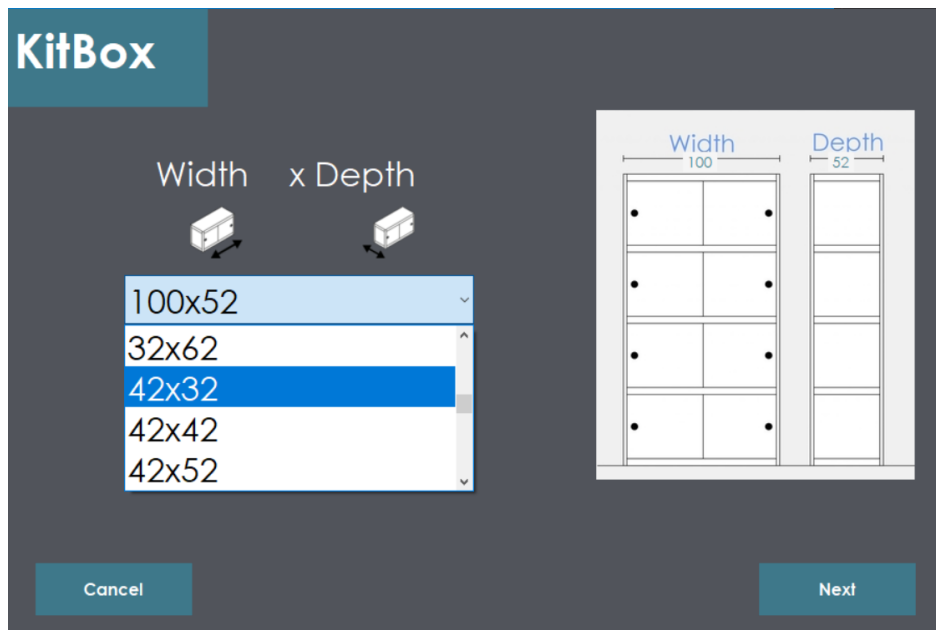


Figure 5

Une liste déroulante vous permettra de choisir les dimensions de la base de votre meuble parmi les dimensions présentes dans le catalogue à la figure 5.



Figure 6

Chaque meuble peut avoir un nom personnalisé ou un nom par défaut afin de mieux s'y retrouver. Vous pouvez choisir les "Corner Color" correspondantes pour tout le meuble. Le bouton "Add" vous permet d'ajouter un nombre maximum de 7 casiers. Pour chaque casier vous avez la possibilité de choisir la hauteur dans une liste déroulante comprenant les dimensions du catalogue.

En fonction des dimensions de la base du meuble, le programme va proposer l'ajout de portes. Il va de même actualiser : le prix dans le coin supérieur de la fenêtre grise, les dimensions ainsi que la disponibilité du casier dans le stock par le carré vert pomme dans le coin supérieur de la fenêtre jaune. Si le carré devient rouge, cela signifie que le casier n'est plus disponible en stock.

Le choix de couleur du casier est également proposé à l'utilisateur. Chaque casier est duplicable, supprimable ou modifiable.

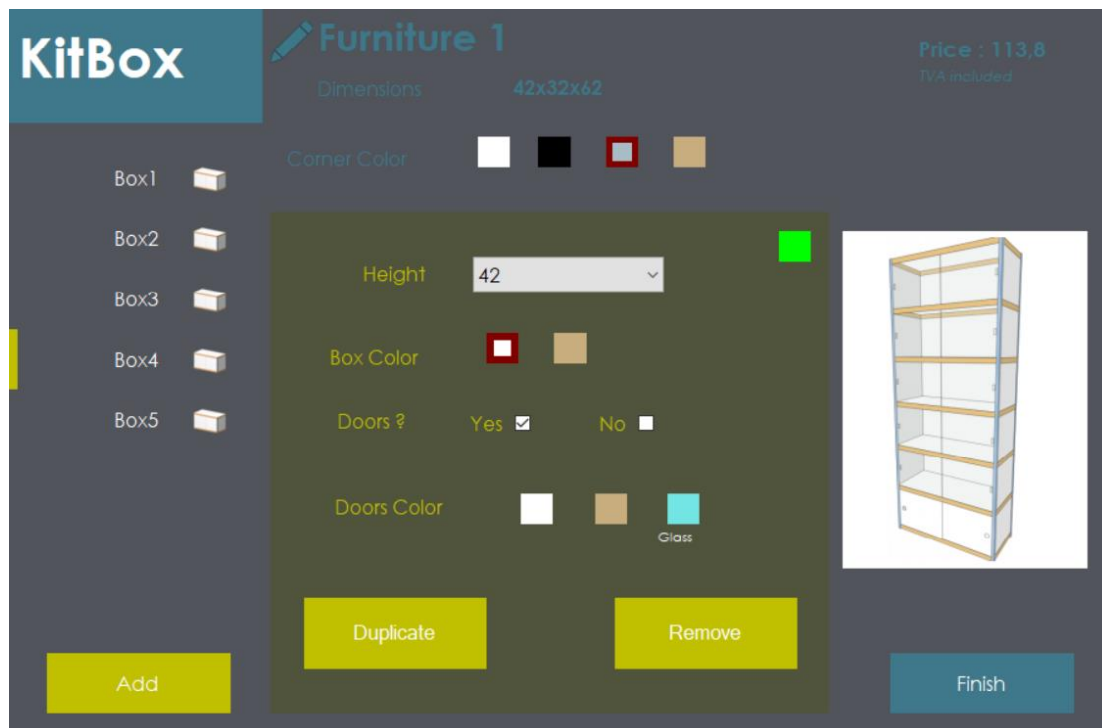


Figure 7

Pour des dimensions de meubles où un choix de porte est possible, le programme propose des portes de couleurs ou des portes en verre. Lorsque vous survolez les couleurs avec la souris, la couleur vous est indiquée à l'écrit comme dans figure 7, "Glass".

Une fois tout votre meuble complet, le bouton finish vous permet d'accéder à un récapitulatif de commande à la figure 8. Si des éléments sont non choisis dans le meuble, le programme vous en préviendra.

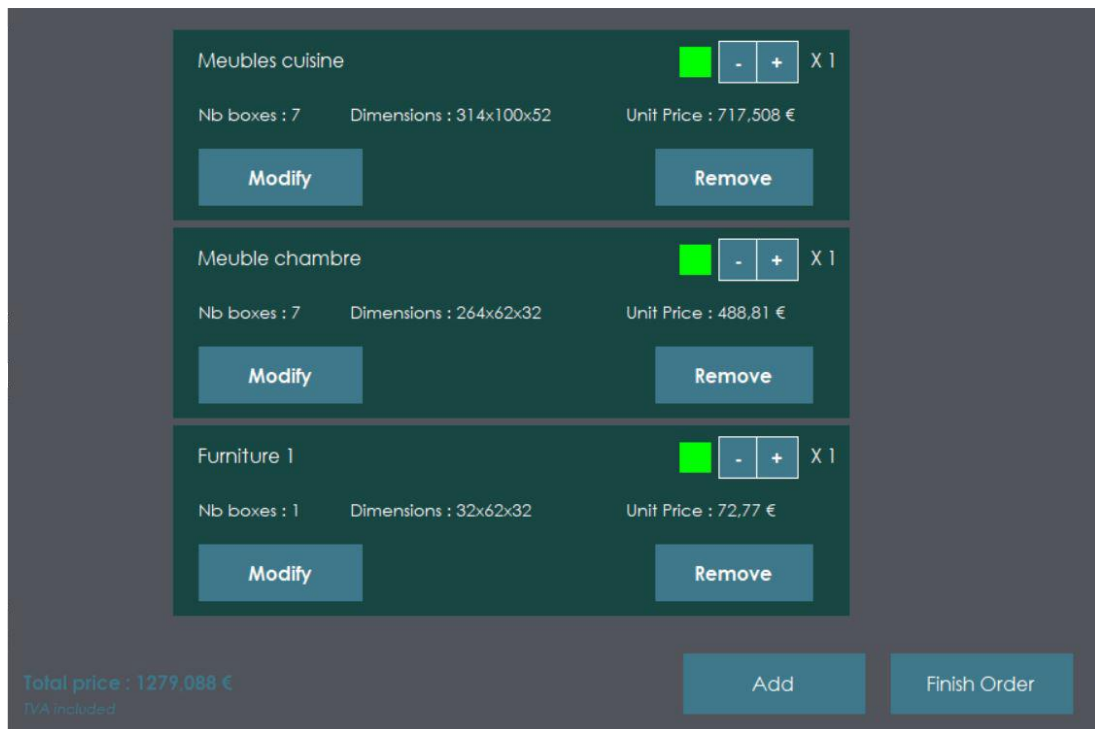


Figure 8

Sur la page récapitulative vous retrouverez un descriptif succinct de votre meuble (nombre de casiers, dimensions, prix du meuble ainsi que la disponibilité dans le stock de toutes les pièces du meuble)

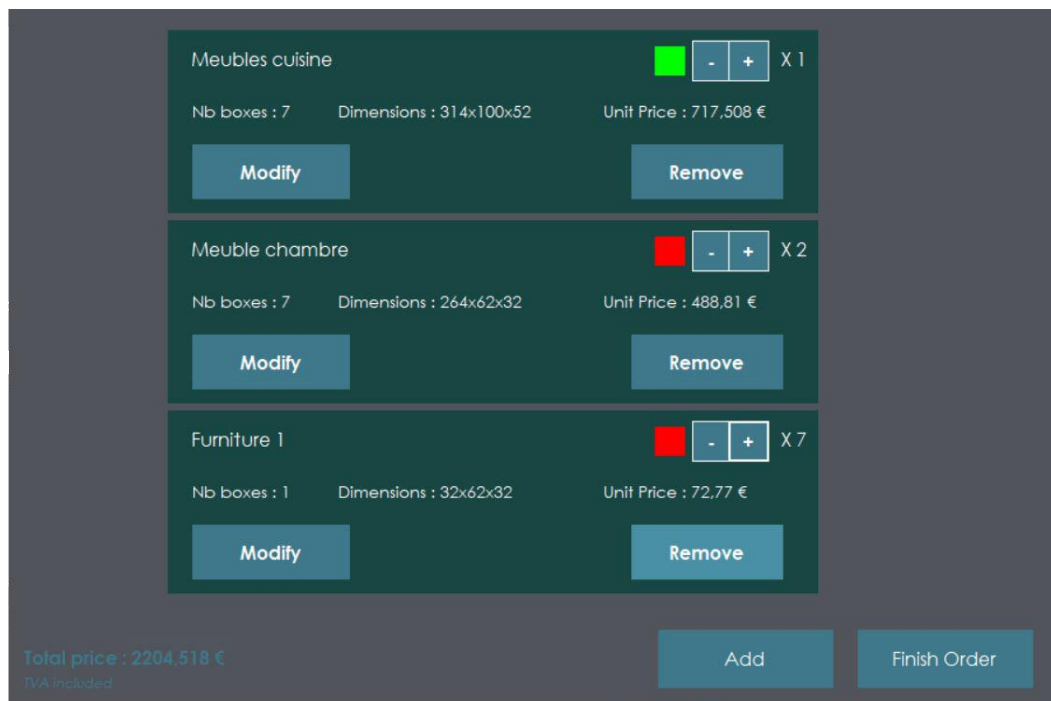


Figure 9

Vous avez la possibilité d'ajouter un nouveau meuble à reconstruire en appuyant sur "Add", cela vous renverra à la figure 5 où vous pouvez débiter votre construction. Vous pouvez rajouter autant de meubles que vous voulez.

Les boutons + et - vous permettront de choisir la quantité de meubles identiques. Une fois que les limites du stock de pièces sont dépassées le carré vert pomme passera au rouge indiquant que certaines pièces ne sont pas directement disponibles dans le magasin (figure 9). Si par exemple, vous voulez changer la couleur d'un meuble déjà validé vous pouvez le faire en appuyant sur "Modify" qui vous renverra à la figure 5 ou 6. Tout en bas de la fenêtre grise vous pouvez retrouver le prix total de vos achats. Le bouton « finish order » vous redirige vers la page des informations relatives à votre facture à la figure 10.

**Order : #10**

Name : Calugar Finias      Adress : Rue de la Reussite

Email : finias@gmail.com      City : Bruxelles

Phone : 0493574      Postal code : 1020

I am :    A particular ☐    A company ☒

TVA number : 654233652

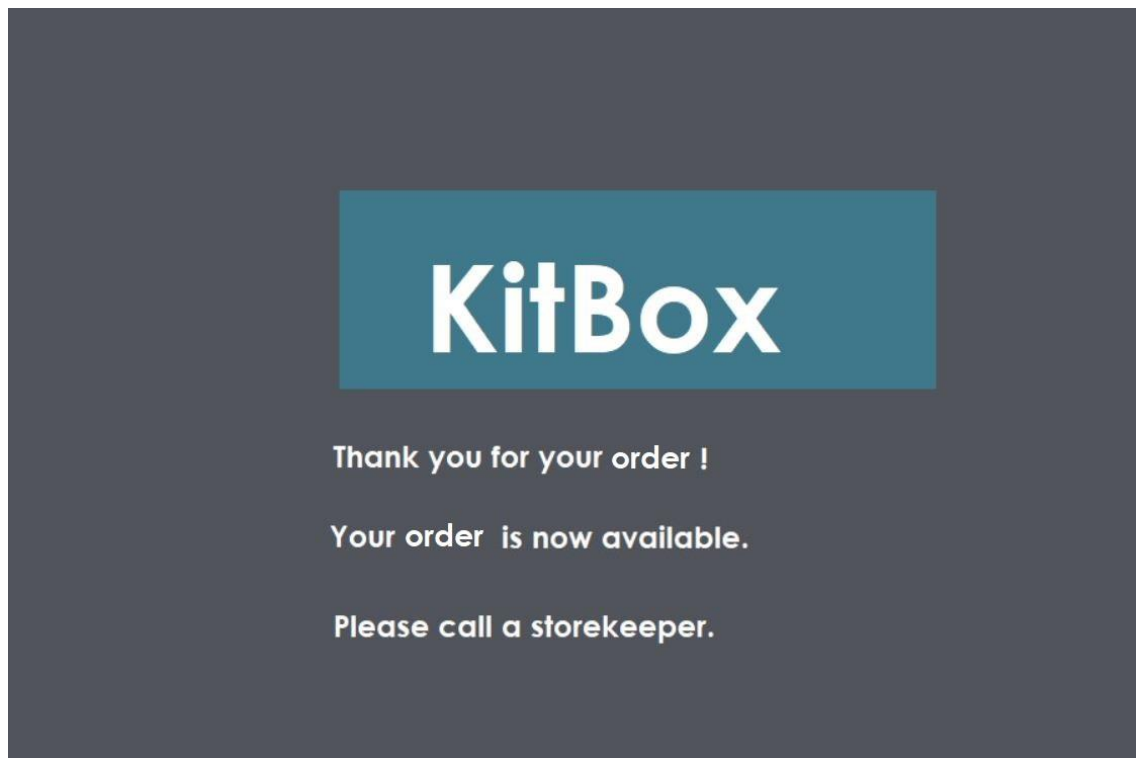
Confirm order

Figure 10

Cette page permet d'insérer les coordonnées essentielles du client et/ou de l'entreprise. Un numéro de commande est assigné à chaque nouvelle commande. En cas d'une entrée incohérente le texte se mettra en rouge afin de prévenir l'utilisateur. Voir un exemple à la figure 10 avec un numéro de GSM invalide sur la commande #10.

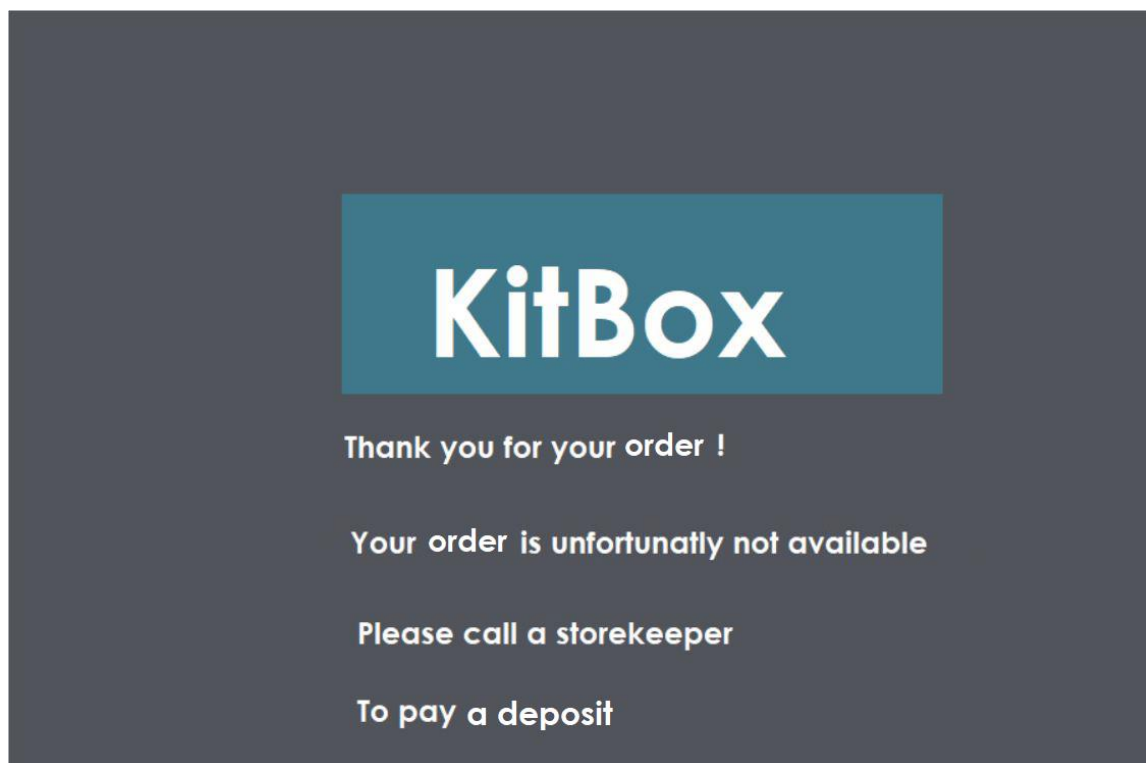
Enfin, en fonction de si toutes les pièces sont disponibles ou pas en stock, le client sera invité soit à payer un acompte (voir figure 12) soit il sera informé que sa commande est disponible (voir figure 11)

Commande disponible :



*Figure 11*

Pièces indisponibles, demande d'un acompte :



*Figure 12*



## 3.2 Interface magasinier

Une page de connexion permet au magasinier de se connecter grâce à un mot de passe.

Figure 13

Une fois connecté, il faut appuyer sur "import" dans la fenêtre de gestion de commande client. Cela permet d'obtenir une liste des numéros des commandes + le nom du client associé dans une table sur la gauche.

En cliquant sur une des commandes dans la table de gauche, on obtient la liste des composants sur la droite. Dans la table en bas de page, sont affichés les informations sur le client qui sont modifiables. Une fois les modifications effectuées, celles-ci peuvent être sauvegardées dans la base de données en appuyant sur « Update Customer ».

ID	Nom
1	Commande Test
2	Vandemeersch ...
4	Vandemeersch ...
7	Calugar dd
8	Test Test
9	Bla Bla
10	Bla Bla
11	sqdsqd qdsqds
12	Bla Bla
13	Bla bla
14	Test Test
15	Bla Bla
16	dsq qsd
17	lkds lkds
18	aze kaze
19	Bla Bla
20	lmsqk lmsqkd
21	sqd qsd
22	Vandemeersch ...

Component_Nombre	Meuble	Box	Ref	Dimension	Couleur	Nb_Pièces_cas
1	1_1	1	Panneau HB	52(p)x100(L)	Blanc	2
2	1_1	1	Panneau Ar	42(h)x100(L)	Blanc	1
3	1_1	1	Panneau GD	42(h)x52(p)	Blanc	2
4	1_1	1	Traverse Ar	100(L)		2
5	1_1	1	Traverse Av	100(L)		2
6	1_1	1	Traverse GD	52(p)		4
7	1_1	1	Tasseau	37(h)x42		4
8	1_1	2	Panneau HB	52(p)x100(L)	Blanc	2
9	1_1	2	Panneau Ar	42(h)x100(L)	Blanc	1
10	1_1	2	Panneau GD	42(h)x52(p)	Blanc	2
11	1_1	2	Traverse Ar	100(L)		2
12	1_1	2	Traverse Av	100(L)		2
13	1_1	2	Traverse GD	52(p)		4
14	1_1	2	Tasseau	37(h)x42		4
15	1_1	3	Panneau HB	52(p)x100(L)	Blanc	2
16	1_1	3	Panneau Ar	42(h)x100(L)	Blanc	1
17	1_1	3	Panneau GD	42(h)x52(p)	Blanc	2
18	1_1	3	Traverse Ar	100(L)		2
19	1_1	3	Traverse Av	100(L)		2
20	1_1	3	Traverse GD	52(p)		4
21	1_1	3	Tasseau	37(h)x42		4
22	1_1	4	Comieres	3x42(h)	Noir	4
23	1_12	1	Panneau HB	52(h)x100(L)	Blanc	2

ID	Nom	Phone	Email	Adresse	Commune	Code_Postal	TVA
18	Bla bla	902813091283	Bla bla@gmail.com	lkjqsldsq	sqds	2929	938012839

Buttons: Import, Component table, Delete, Update Customer, Provider Tables

Figure 14

Le bouton "Component Table" permet d'accéder à la fenêtre affichant la table des composants sur la gauche et la table des commandes fournisseurs sur la droite, comme dans la figure suivante.

Ref	Code	Dimension	Hauteur	Profondeur	Largeur	Couleur
Comieres	COR100BLDEC	100(h)decoupee	100	0	0	Blanc
Comieres	COR100BRDEC	100(h)decoupee	100	0	0	Brun
Comieres	COR100GLDEC	100(h)decoupee	100	0	0	Galvanise
Comieres	COR100NRDEC	100(h)decoupee	100	0	0	Noir
Comieres	COR108BL	3x32(h)	108	0	0	Blanc
Comieres	COR108BR	3x32(h)	108	0	0	Brun
Comieres	COR108GL	3x32(h)	108	0	0	Galvanise
Comieres	COR108NR	3x32(h)	108	0	0	Noir
Comieres	COR112BL	2x52(h)	112	0	0	Blanc
Comieres	COR112BR	2x52(h)	112	0	0	Brun
Comieres	COR112GL	2x52(h)	112	0	0	Galvanise
Comieres	COR112NR	2x52(h)	112	0	0	Noir
Comieres	COR125BLDEC	125(h)decoupee	125	0	0	Blanc
Comieres	COR125BRDEC	125(h)decoupee	125	0	0	Brun
Comieres	COR125GLDEC	125(h)decoupee	125	0	0	Galvanise
Comieres	COR125NRDEC	125(h)decoupee	125	0	0	Noir
Comieres	COR138BL	3x42(h)	138	0	0	Blanc
Comieres	COR138BR	3x42(h)	138	0	0	Brun
Comieres	COR138GL	3x42(h)	138	0	0	Galvanise
Comieres	COR138NR	3x42(h)	138	0	0	Noir
Comieres	COR144BL	4x32(h)	144	0	0	Blanc
Comieres	COR144BR	4x32(h)	144	0	0	Brun
Comieres	COR144GL	4x32(h)	144	0	0	Galvanise
Comieres	COR144NR	4x32(h)	144	0	0	Noir
Comieres	COR150BLDEC	150(h)decoupee	150	0	0	Blanc
Comieres	COR150BRDEC	150(h)decoupee	150	0	0	Brun
Comieres	COR150GLDEC	150(h)decoupee	150	0	0	Galvanise
Comieres	COR150NRDEC	150(h)decoupee	150	0	0	Noir

Figure 15

Le bouton "Suggest" suggère des commandes à faire aux fournisseurs si les stocks sont insuffisants. Ces suggestions sont affichées dans une table à droite. Une fois un composant reçu il faut valider la réception en appuyant sur le composant et ensuite sur "Received" ou supprimer le composant en appuyant sur "Delete". Le composant en question est rajouté au stock.

Le bouton "Update" permet d'envoyer les modifications effectuées à la base de données.

ID_Fournisseur	ID_Composant	Validation	Quantity
2	COR100BLDEC	<input type="checkbox"/>	53
2	COR100BRDEC	<input type="checkbox"/>	52
2	COR100GLDEC	<input type="checkbox"/>	51
2	COR100NRDEC	<input type="checkbox"/>	55
2	COR108BL	<input type="checkbox"/>	52
2	COR108BR	<input type="checkbox"/>	55
2	COR108GL	<input type="checkbox"/>	55
2	COR108NR	<input type="checkbox"/>	54
2	COR112BL	<input type="checkbox"/>	50
2	COR112BR	<input type="checkbox"/>	50
2	COR112GL	<input type="checkbox"/>	56
2	COR112NR	<input type="checkbox"/>	57
2	COR125BLDEC	<input type="checkbox"/>	51
2	COR125BRDEC	<input type="checkbox"/>	52
2	COR125GLDEC	<input type="checkbox"/>	51
2	COR125NRDEC	<input type="checkbox"/>	54
2	COR138BL	<input type="checkbox"/>	51
2	COR138BR	<input type="checkbox"/>	55
2	COR138GL	<input type="checkbox"/>	57
2	COR138NR	<input type="checkbox"/>	51
2	COR144BL	<input type="checkbox"/>	54
2	COR144BR	<input type="checkbox"/>	55
2	COR144GL	<input type="checkbox"/>	52
2	COR144NR	<input type="checkbox"/>	56
2	COR150BLDEC	<input type="checkbox"/>	57
2	COR150BRDEC	<input type="checkbox"/>	55
2	COR150GLDEC	<input type="checkbox"/>	55
2	COR150NRDEC	<input type="checkbox"/>	55

Figure 16

Un bouton "Provider Table" nous affiche une table des composants sur la gauche avec leur prix et leurs délais de livraison, en fonction de chaque fournisseur et une table avec les informations des fournisseurs sur la droite, comme dans la figure suivante.

**Form3**

ID_Fournisseur	Code_composant	Prix	Delai
1	COR100BLDEC	1.08	6
1	COR100BRDEC	0.87	12
1	COR100GLDEC	1.13	5
1	COR100NRDEC	0.77	10
1	COR108BL	0.85	7
1	COR108BR	0.69	7
1	COR108GL	0.84	6
1	COR108NR	0.55	13
1	COR112BL	0.89	12
1	COR112BR	0.74	5
1	COR112GL	0.94	8
1	COR112NR	0.62	6
1	COR125BLDEC	1.28	7
1	COR125BRDEC	1.04	7
1	COR125GLDEC	1.26	6
1	COR125NRDEC	0.83	13
1	COR138BL	1.08	11
1	COR138BR	0.85	12
1	COR138GL	1.02	6
1	COR138NR	0.75	11
1	COR144BL	1.08	3
1	COR144BR	0.82	11
1	COR144GL	1.13	12
1	COR144NR	0.8	5
1	COR150BLDEC	1.62	11
1	COR150BRDEC	1.28	12
1	COR150GLDEC	1.53	6
1	COR150NRDEC	1.13	11
1	COR168R	1.3	5

ID	Nom1	Nom2	Adresse	Code_Postal
1	TRABELBO SA	Les Artisans du B...	Parc industriel 9	6124
2	TratBois SPRL	Ebenistes reunis	Chemin des 2 Ma...	5150

Update Catalogue      Update Provider  
Delete Component      Delete Provider

Figure 17

ID Client : 27	Nom : Vandermeersch Thomas	
Adresse mail : <a href="mailto:thomas@wezembeek.be">thomas@wezembeek.be</a>	N° téléphone : +324739049330	
Adresse : Avenue Saint-Hubert,43	Ville : Wezembeek-Oppem	Code postal : 1970

## Facture de la commande n° 22

	Quantité	Prix (€)
Living Room	2	415,56
Cornière Galvanise	4	5,08
Box 1	1	58,6
Box 2	1	72,05
Box 3	1	72,05
<b>Kitchen</b>	4	3288,656
Cornière Noir	4	0,98
Box 1	1	117,312
Box 2	1	117,312
Box 3	1	117,312
Box 4	1	117,312
Box 5	1	117,312
Box 6	1	117,312
Box 7	1	117,312
	<i>Total (HTVA)</i>	<i>3061,336</i>
	<i>TVA (21%)</i>	<i>642,88</i>
	<b>Prix total</b>	<b>3704,216</b>

Figure 18 Facture d'une commande

La facture (figure 19) quant à elle est créée automatiquement dans un fichier sur l'ordinateur ou l'application été lancée.

## 4 Justification des choix architecturaux et technologiques

### 4.1 Choix d'architecture

Notre programme débute par l'importation de toute la base de données en local afin d'éviter des latences des demandes SQL pendant l'utilisation.

Ensuite, le choix des dimensions de la base du meuble se fait à partir d'une liste déroulante, remplie à partir de la base de données. Débuter par ces dimensions nous semblait très important car elles nous permettent de présélectionner que les produits constituant les casiers disponibles à ces dimensions choisies. Une fois ces dimensions choisies, une nouvelle page est disponible. Nous avons structuré le programmes par pages reprenant chacun des fonctionnalités propres comme par exemple : page création de casiers, page de récapitulatif, page d'inscriptions de données etc.

Pour les pages création des casiers et la page de récapitulatif nous avons décidé de laisser le choix à l'utilisateur en lui donnant de nombreuses possibilités de création (duplication ou suppression des casiers ou des meubles, nommer les meubles...) et cela tout en le prévenant en direct de la disponibilité du stock et du prix.

Enfin, pour une compréhension plus détaillée de l'architecture cfr point 5 (tous les diagrammes).

### 4.2 Choix technologiques

#### **Programmation Orientée Objet**

Afin de répondre au mieux aux besoins des clients nous nous sommes dirigés vers la programmation orientée objet (POO) qui est selon nous la solution idéale pour notre travail. Il s'agit ici, de laisser le choix à l'utilisateur de construire ou de manipuler des objets pour construire des meubles sans se soucier de risques d'erreurs. La POO nous permet donc de créer une large variété d'objets avec des paramètres différents (couleurs, dimensions...) et des méthodes, interagissant entre eux.

La POO permet aussi de se servir d'objets déjà créés pour être la base d'autres objets, comme nous l'avons fait dans notre cas pour construire des casiers. En effet, nous avons utilisés les objets : cornières, tasseaux, traverses, panneaux et portes pour constituer un nouvel objet : casier.

De plus dans la POO, il existe une séparation nette entre les différents composants de notre programme tels que : interface, base de données, créations d'objets. Cette séparation permet un travail d'équipe plus aisé. Finalement, le plus grand avantage de la POO est la modularité car elle permet une amélioration future de l'application et/ou des extensions, des rajouts, des modifications... Laissant un large panel de possibilités futures au client.

## **C#**

Ce langage de programmation nous a été imposé pour réaliser ce travail. Le C# est un langage très répandu et en constante évolution, il permet une programmation aisée de différents softwares, une utilisation de différents *frameworks* et il dispose surtout d'une grande documentation sur internet. Finalement il permet l'utilisation d'IDE professionnels tels que Visual Studio, Mono Develop ou Unity 3D.

## **Trello**

Trello est une interface de gestion de projet en ligne nous ayant permis de facilement nous répartir des tâches de manière visuelle et d'en suivre leur évolution.

## **GitHub**

Malgré des débuts difficiles, GitHub peut très vite nous aider dans la gestion de projets d'équipe. Il permet un versioning automatique, un travail d'équipe en parallèle sur différentes branches, évite la perte de données grâce aux fameux « commits » et permet des retours en arrière en cas de problèmes.

## **Visual Studio**

En ce qui concerne l'IDE, nous avons travaillé sur Visual Studio qui est compatible avec les différentes technologies précitées. Visual Studio permet donc de créer un lien direct avec GitHub pour le versioning de nos codes et permet également la programmation en C# avec une aide pour la création d'interfaces utilisateurs (Windows forms) très utile pour la création des interfaces clients et magasinier.

## **phpMyAdmin**

Cette application web permet de créer une base de données MySQL en ligne permettant de facilement gérer notre base de données MySQL. Malgré sa grande latence, sa gratuité et sa documentation sont des atouts.

## **Jitsi Meet**

Logiciel de conférences en ligne gratuit compatible avec tout Browser et ne nécessitant aucune création de compte.

## 5 Tous les diagrammes et documents réalisés

### 5.1 Explication succincte du diagramme de classe

Un "Order" est à voir comme un ensemble de meubles dont chaque meuble possède ses propres caractéristiques. Une classe "Furniture" est instanciable pour créer autant d'objets meubles que l'on veut. Chaque objet meuble est composé de cornières et de casiers qui à leur tour sont construits à partir de composants divers. Chaque composant hérite de toutes les caractéristiques de la classe abstraite "Component" et rajoute ses caractéristiques propres.

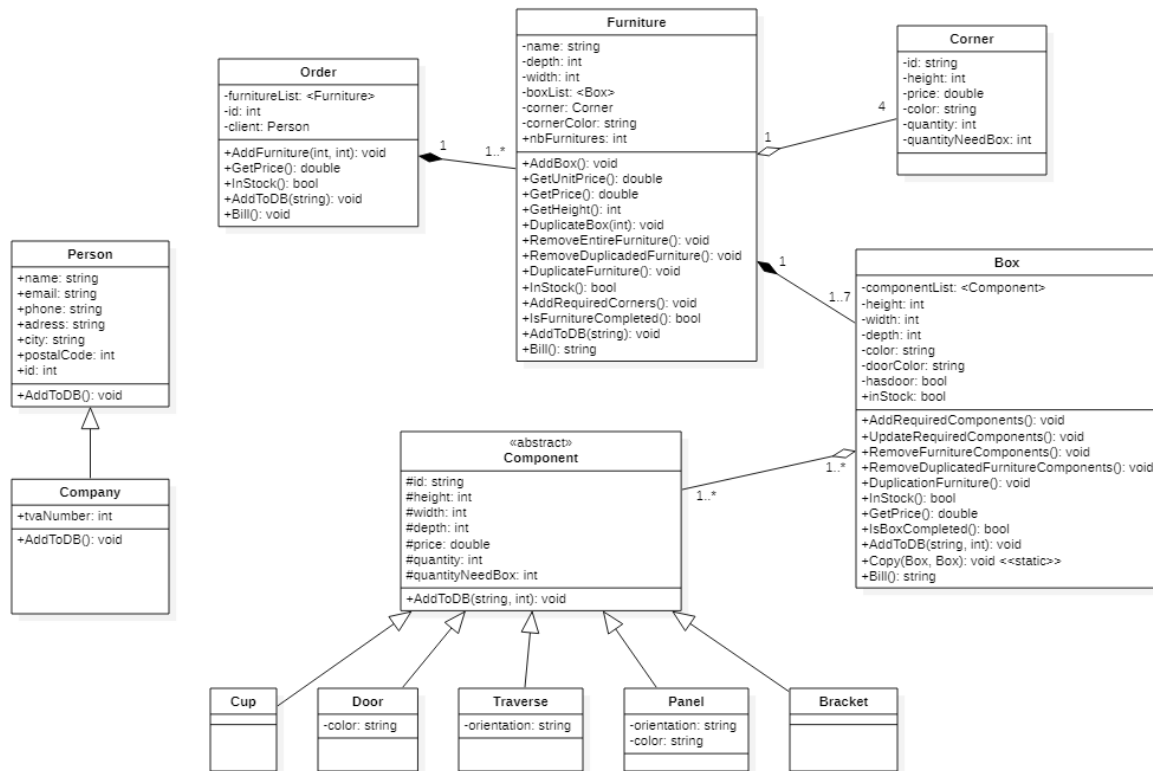


Figure 19

## 5.2 Schéma relationnel

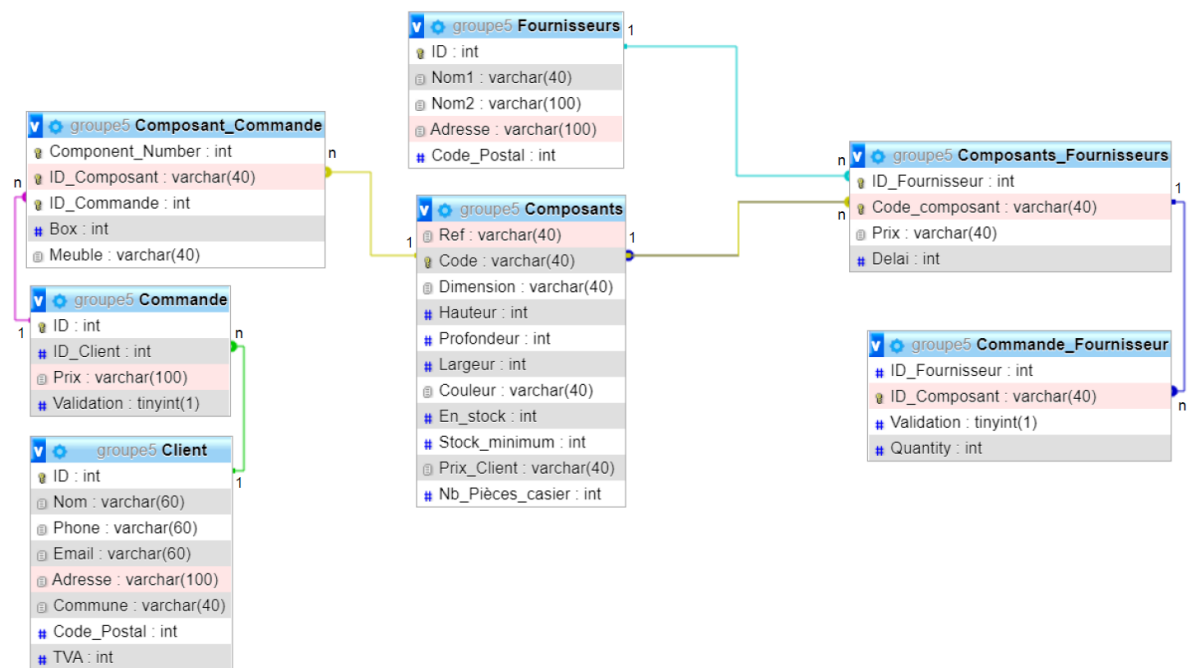


Figure 20 Diagramme de base de données

Ce diagramme est à la base du fonctionnement de notre base de données.

## 5.3 Diagrammes de séquence

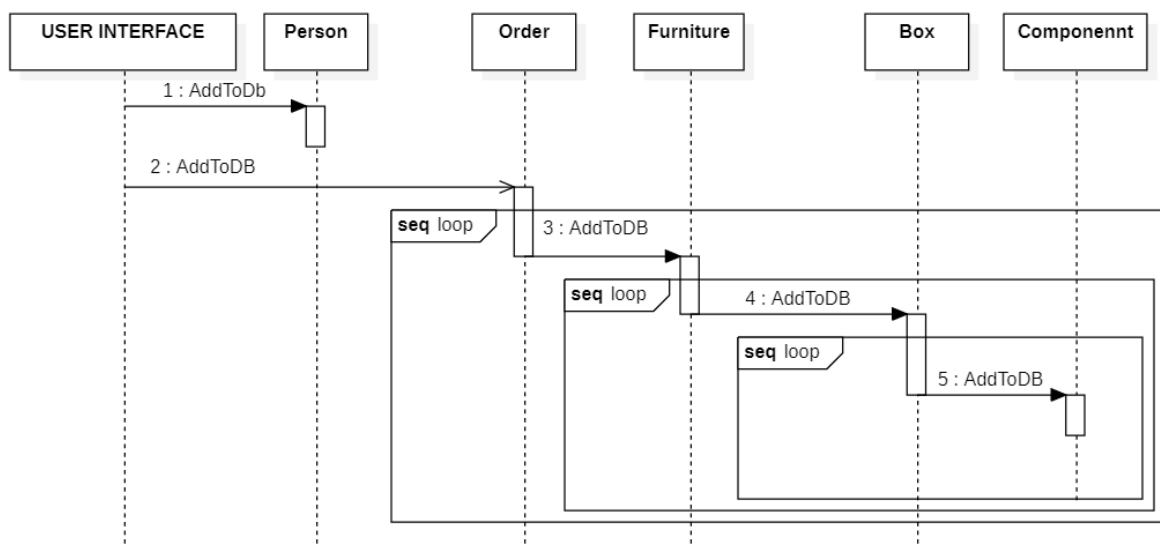


Figure 21 Diagramme de séquence: ajout à la base de donnée



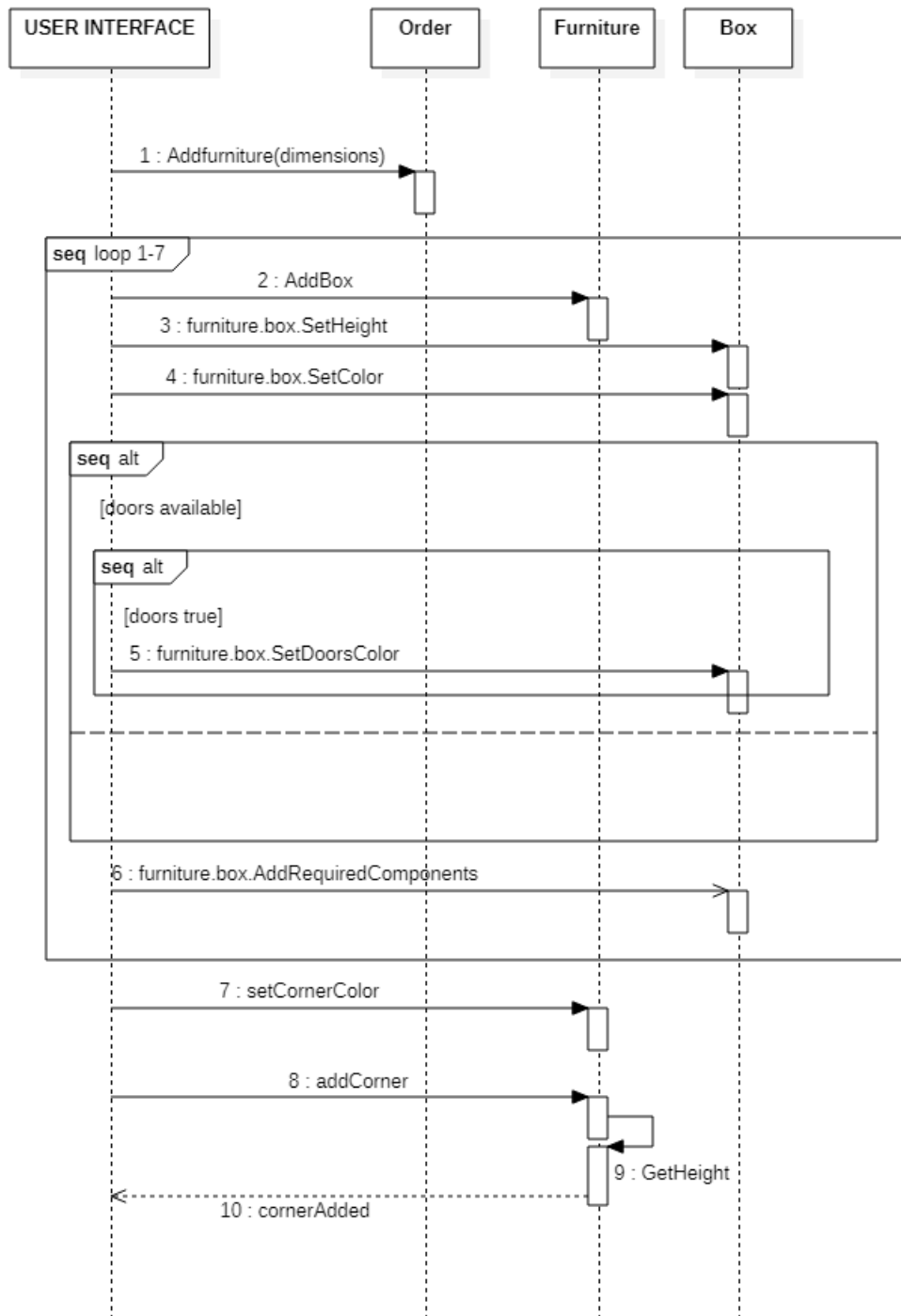


Figure 22 Diagramme de séquence : création de meuble

Le diagramme de séquence de la création d'un meuble est un exemple. En effet, l'utilisateur n'est pas obligé d'entrer les différentes informations (couleur de box, de portes) dans l'ordre établi dans le diagramme de séquence. De plus, la méthode `AddRequiredComponents()`, est exécutée à chaque modification. Cela permet de mettre à jour le prix, les dimensions et le stock en temps réel. Le diagramme est là pour montrer l'idée derrière le code. Il n'a pas pour but de refléter la pure réalité.

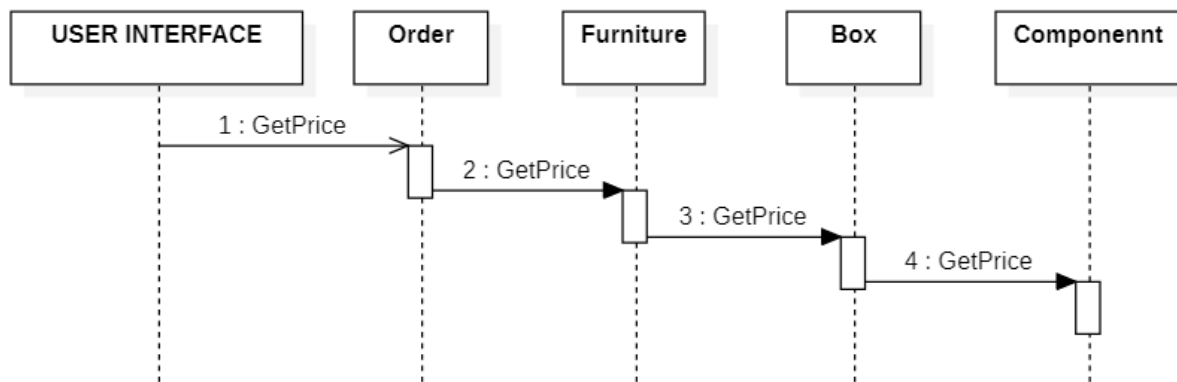


Figure 23 Diagramme de séquence : retour du prix

Le diagramme de séquence ci-dessus est un modèle de l'appel des fonctions `GetPrice ()`, `InStock ()` et `Bill ()` lorsqu'elles sont appelées sur `Order`. Cette architecture permet de facilement retrouver les informations sur une commande en particulier, un meuble en particulier, une box particulière ou encore un composant.

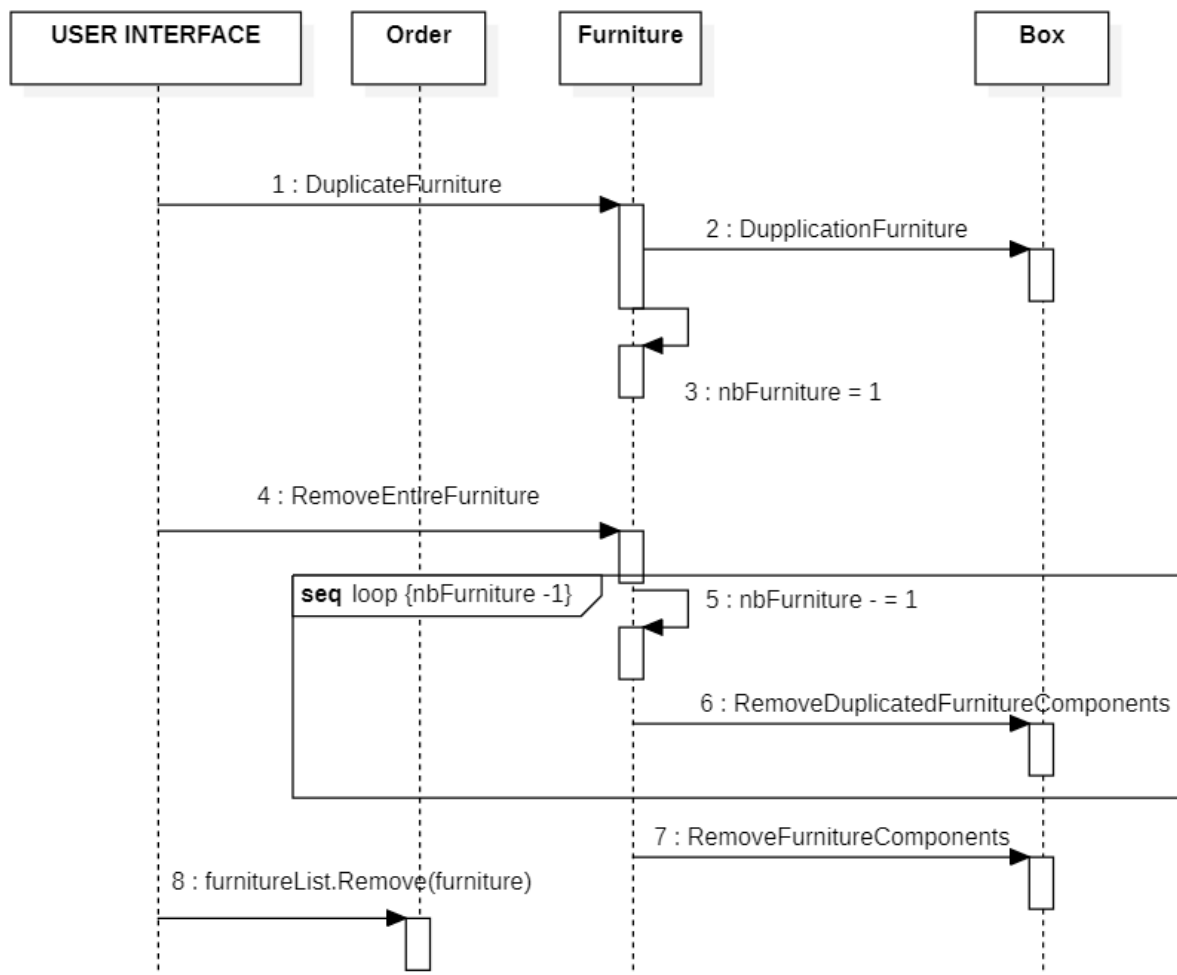


Figure 24 Diagramme de séquence : Supprimer l'entièreté d'un meuble

Lors d'une suppression, il faut "réapprovisionner" le stock. C'est pourquoi, la fonction de suppression doit passer par plusieurs classes.

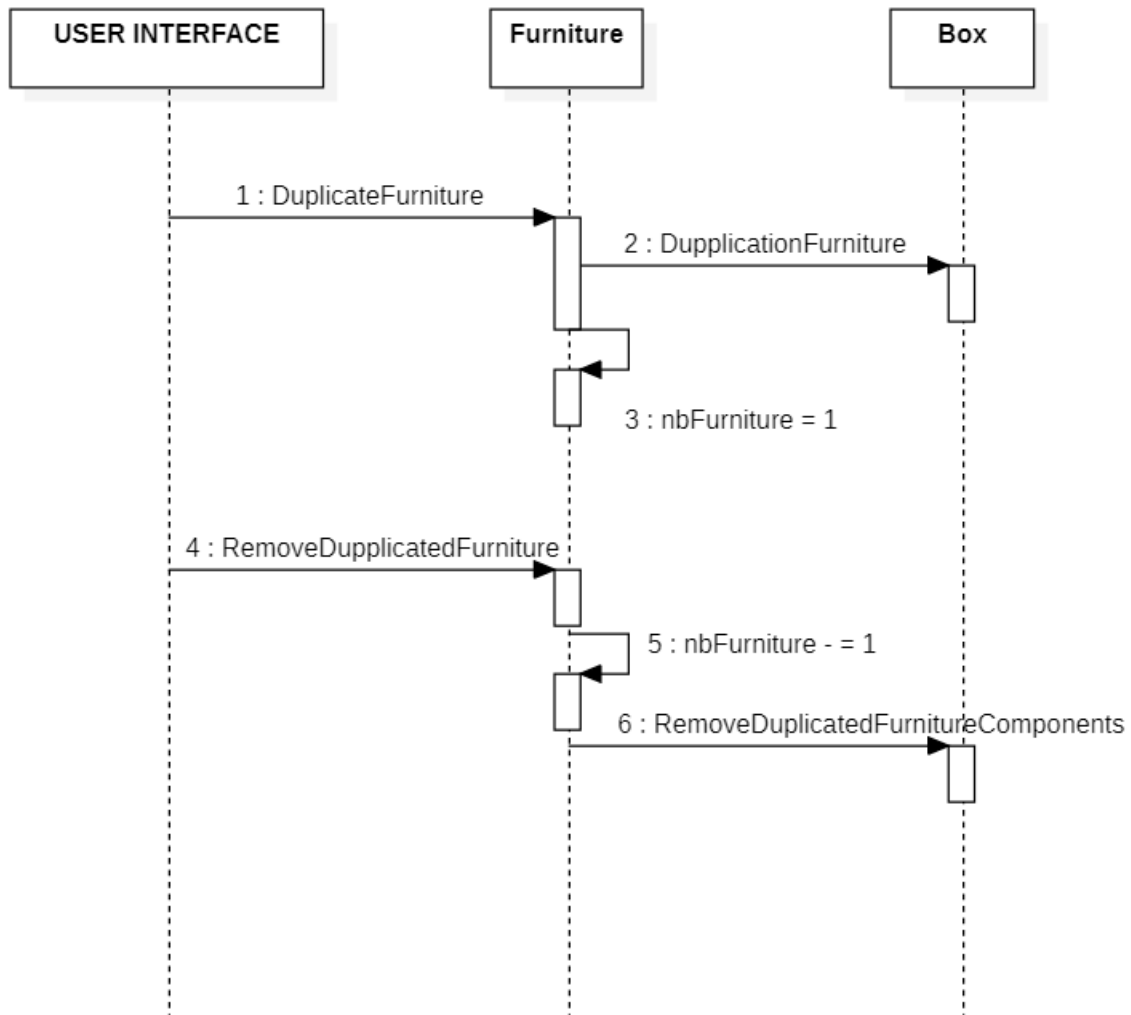


Figure 25 Diagramme de séquence: Supprimer et/ou dupliquer un meuble

## 6 Conclusions

### 6.1 Objectifs

Les objectifs que nous avons atteints correspondent à l'accomplissement des exigences du client :

- Création d'une base de données regroupant tous les produits.
- Création d'une interface client esthétique et « userfriendly » permettant la création de meubles personnalisés.
- Création d'une interface magasinier permettant la gestion des commandes et des stocks.

Nous avons accompli tous les objectifs. Cependant le client a la possibilité d'apporter des améliorations futures à certaines parties.

### 6.2 Pistes d'améliorations

Plusieurs points sont ressortis comme pouvant être améliorés bien que non demandés :

- L'esthétique de l'interface magasinier.
- Le redimensionnement de la fenêtre de l'application.
- Une indication précise des parties manquantes dans le stock.
- Un affichage dynamique du meuble en construction et ses couleurs.
- la latence de l'importation des données.
- un traitement des données de l'utilisation de l'application permettant de se faire une idée du marché et d'améliorer sa stratégie marketing. Des exemples : « le meuble préféré des clients », « le meuble le plus vendu », etc.

### 6.3 Collaboration dans le groupe

Ce projet nous a permis d'apprendre à mieux gérer le travail en équipe et la personnalité de chacun. Comme vu au cours de « Génie Logiciel », nous avons toujours essayé de réaliser les travaux, au moins par binôme, afin de permettre la correction en direct d'erreurs potentielles. Cela fut très bénéfique.

Les plateformes telles que Trello ou Jitsi nous ont permis de poursuivre nos tâches et de rester connectés dans le groupe malgré la situation compliquée due au confinement. Le groupe a de nouveau fait preuve d'entraide en se soutenant dans les tâches même en distanciel.

En effet, à certains moments GitHub ne nous a pas aidé à la collaboration à cause des différents "merge" difficiles à effectuer qui ont fait que le travail a dû par moment être centralisé.

Enfin, nous pensons que chacun a donné le meilleur de soi dans la réalisation de ce projet et cela s'est démontré par l'achèvement de celui-ci.