

Etat : En attente de validation  
V1.0

11/11/2014

AppInfo



Votre logiciel en toute sérénité

# DOSSIER DE CONCEPTION

Projet : 24/24Manager

Client : CPE Lyon



Julien BRIOT – Fabien CHAMPEL – Justine POYARD  
Long LE DAC – Kim HERR

## Objet

Ce document vise à présenter l'architecture du logiciel 24/24 Manager. Ce dernier doit mettre en place le système informatique de gestion d'approvisionnement en pain, boissons et viennoiseries d'un magasin ouvert 24h/24 dans un aéroport, afin de gérer le contenu des présentoirs, leur approvisionnement et les commandes.

Ce document de conception décrit la structure générale du logiciel prochainement développé par l'équipe AppInfo pour son client, l'école CPE. Ce dossier exprime l'architecture du logiciel, son fonctionnement interne. Il s'appuie sur les spécifications des besoins utilisateurs.

Documents Applicables	-
Documents de Référence	- Dossier de Spécifications
Auteurs	<b>L'équipe AppInfo :</b> <ul style="list-style-type: none"> <li>- Julien Briot</li> <li>- Fabien Champel</li> <li>- Kim Herr</li> <li>- Long Le Dac</li> <li>- Justine Poyard</li> </ul>
Clients	<b>L'école de Chimie, Physique et Electronique de Lyon :</b> <ul style="list-style-type: none"> <li>- Xavier Trouillot</li> <li>- Régis Mathieu</li> <li>- Nicolas Padey</li> <li>- Jérôme Thévenon</li> </ul>
Validation Finale du Client	<ul style="list-style-type: none"> <li>- Xavier Trouillot</li> <li>- Nicolas Padey</li> </ul>

## Table des matières

<b>Objet.....</b>	<b>2</b>
<b>Table des matières.....</b>	<b>3</b>
<b>1. Recueil des besoins .....</b>	<b>4</b>
1.1. Connexion / Déconnexion / Quitter .....	4
1.1.1. Principe générale.....	4
1.1.2. Diagramme de cas d'utilisation.....	4
1.1.3. Diagrammes de séquence associés.....	5
1.2. Utilisateur : vendeur .....	6
1.2.1. Principe générale.....	6
1.2.2. Diagramme de cas d'utilisation.....	6
1.2.3. Diagrammes de séquence associés.....	7
1.3. Utilisateur : employé de cuisson .....	9
1.3.1. Principe générale.....	9
1.3.2. Diagramme de cas d'utilisation.....	9
1.3.3. Diagrammes de séquence associés.....	10
1.4. Utilisateur : manager .....	12
1.4.1. Principe générale.....	12
1.4.2. Diagramme de cas d'utilisation.....	12
1.4.3. Diagrammes de séquence associés.....	13
<b>2. Analyse .....</b>	<b>16</b>
2.1. Diagramme de classe .....	16
2.2. Descriptifs des classes.....	16
2.3. Diagramme de séquence .....	17
2.3.1. Connexion / Déconnexion / Quitter.....	17
2.3.2. Interface vendeur .....	18
2.3.3. Interface employé de cuisson .....	21
2.3.4. Interface manager .....	23
<b>3. Conception.....</b>	<b>26</b>
3.1. Complétion de gauche à droite .....	26
3.1.1. Connexion / Déconnexion / Quitter.....	26
3.1.2. Interface vendeur .....	27
3.1.3. Interface employé de cuisson .....	31
3.1.4. Interface manager .....	33
3.2. Cas d'utilisation annexe.....	37
3.2.1. Afficher aide .....	37
3.3. Hors cas d'utilisation .....	37
3.3.1. Thread de récupération des données contenues dans la base .....	37
3.3.2. Thread pour vérifier les produits périmés.....	38
3.3.3. Thread pour vérifier la quantité des stocks congelés .....	39
3.3.4. Thread pour vérifier la quantité des stocks en vente .....	40
3.4. Diagrammes d'activités et d'état transition.....	40
<b>4. Implémentation.....</b>	<b>41</b>
4.1. Partie statique .....	41
4.2. Partie dynamique.....	41
<b>5. Annexe A : Diagramme d'interface .....</b>	<b>42</b>
5.1. Diagrammes .....	42
5.2. Descriptive des classes .....	42

## 1. Recueil des besoins

Dans cette partie nous décrivons les besoins présentés dans le dossier de spécifications. Les besoins se scindent en quatre sous-parties, une première partie qui est commune à tous types d'utilisateur et trois parties spécifiques à chaque type d'utilisateur :

- les besoins liés à la connexion et la déconnexion.
- les besoins liés à l'interface du vendeur,
- les besoins liés à l'interface de l'employé de cuisson,
- les besoins liés à l'interface du manager,

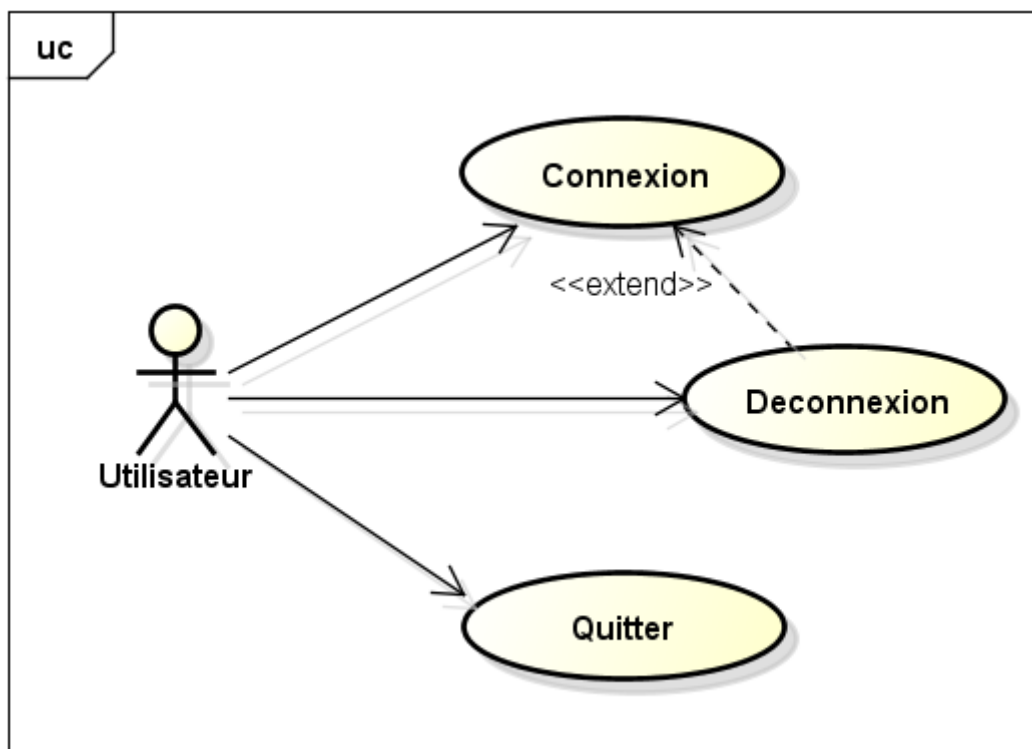
### 1.1. Connexion / Déconnexion / Quitter

#### 1.1.1. Principe générale

Pour utiliser le logiciel 24/24 Manager, l'utilisateur doit dans un premier temps se connecter grâce à ses identifiants. Lorsqu'il a fini de travailler, il peut se déconnecter ou quitter le logiciel.

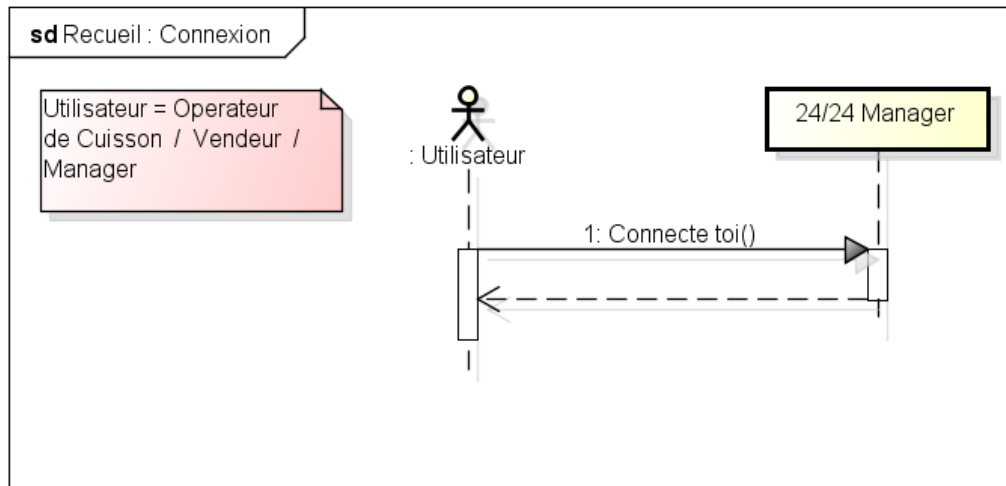
#### 1.1.2. Diagramme de cas d'utilisation

Le cas d'utilisation de la déconnexion étend celui de la connexion étant donné qu'un utilisateur ne peut se déconnecter que s'il est connecté (Lien « extend »). L'action de quitter le logiciel peut être effectuée à tout moment et ne nécessite pas une déconnexion de la part de l'utilisateur.

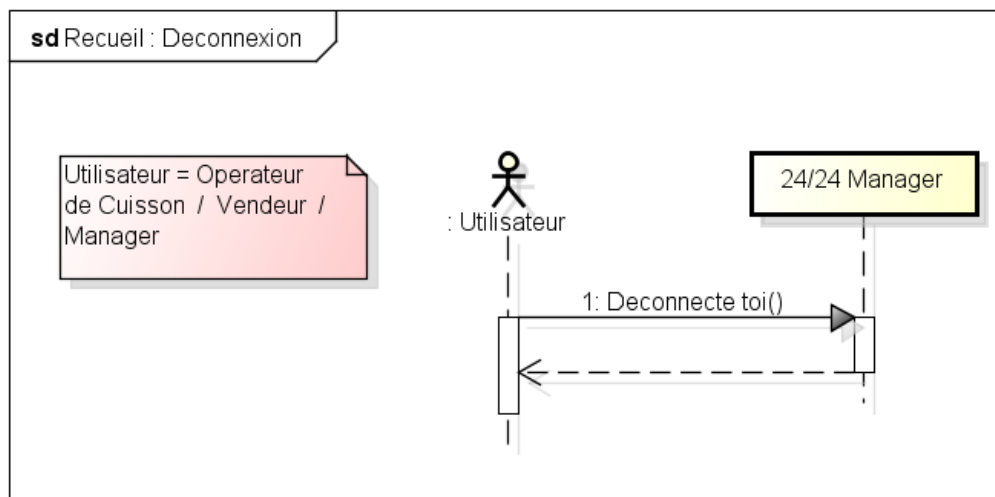


### 1.1.3. Diagrammes de séquence associés

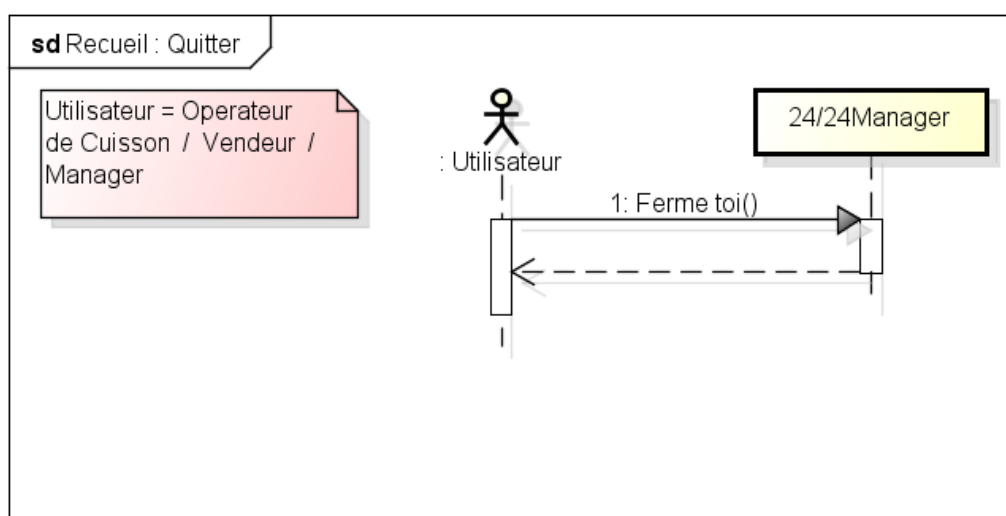
#### Connexion



#### Déconnexion



#### Quitter



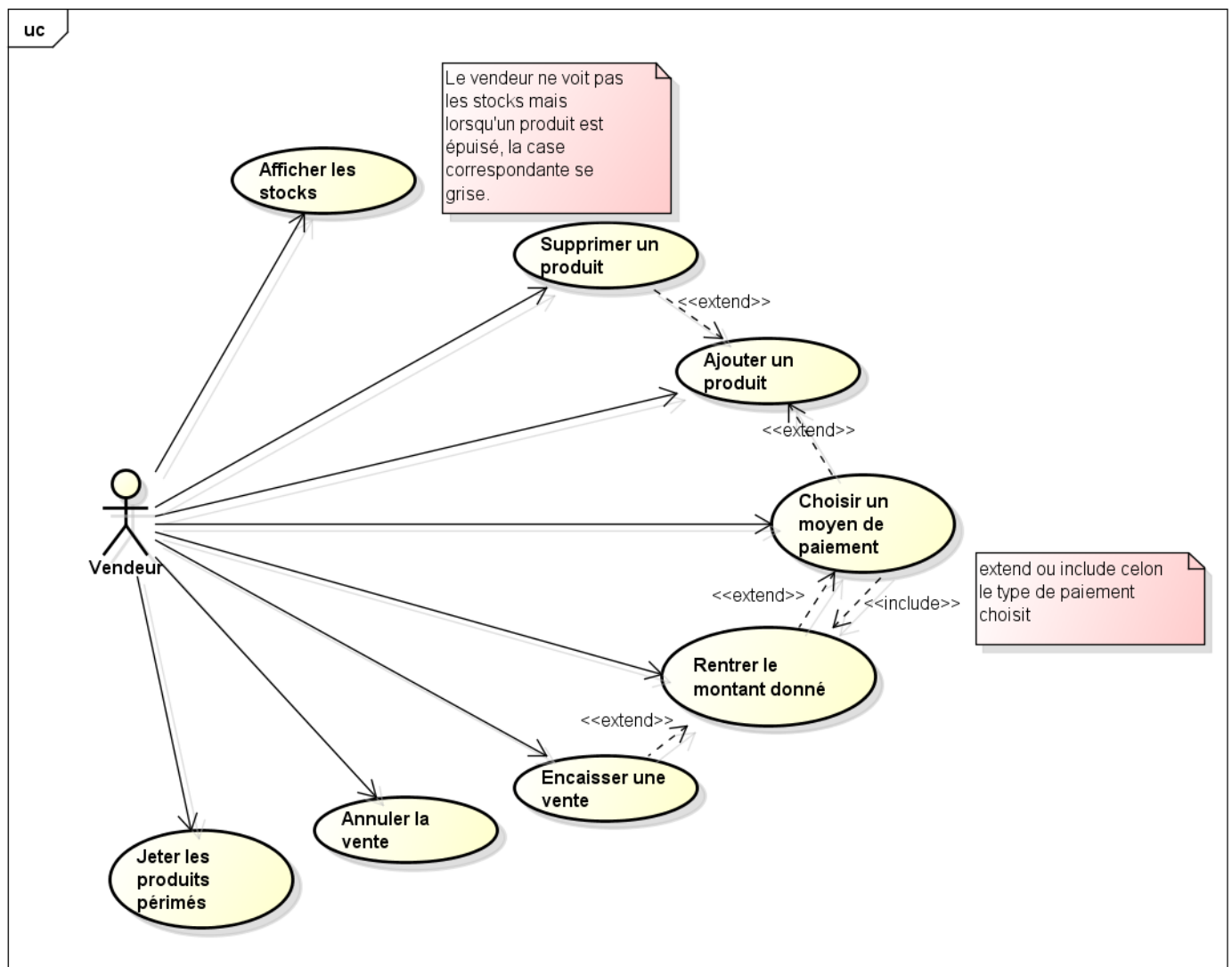
## 1.2. Utilisateur : vendeur

### 1.2.1. Principe générale

Si l'utilisateur a renseigné lors de la connexion qu'il avait pour rôle celui de vendeur alors il accède aux cas d'utilisation spécifiques de ce rôle. Ces cas d'utilisation sont directement reliés à la réalisation d'une vente

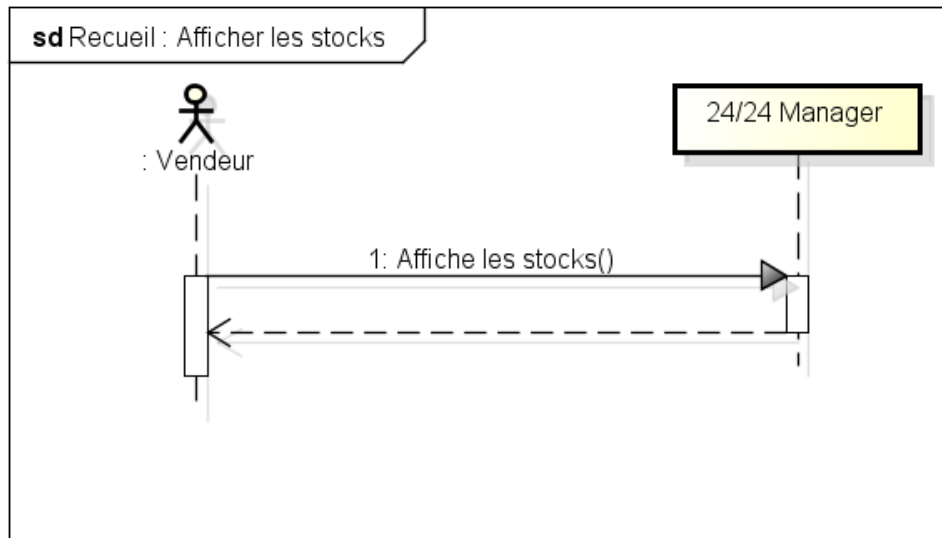
### 1.2.2. Diagramme de cas d'utilisation

Après avoir ajouté un ou des produits à la vente, le vendeur peut supprimer des produits ou sélectionner un type de paiement (Liens « extend » vers « Ajouter un produit »). Ensuite, si le type de paiement est le liquide alors le vendeur peut rentrer le montant donné par le client (Lien « extend » de « Rentre le montant donné » à « Choisir type de paiement »), cependant si le type est la carte bancaire ou le chèque alors le logiciel indique directement le montant donné qui correspond au prix total de la commande (Lien « include » de « Choisir type de paiement » à « Rentre le montant donné »). Enfin, lorsque le montant donné est renseigné, le vendeur peut encaisser la vente (Lien « extend » de « Encaisser » à « Rentrer montant donné »). Les cas « afficher les stocks » et « annuler vente » peuvent être réalisés à n'importe quel moment.

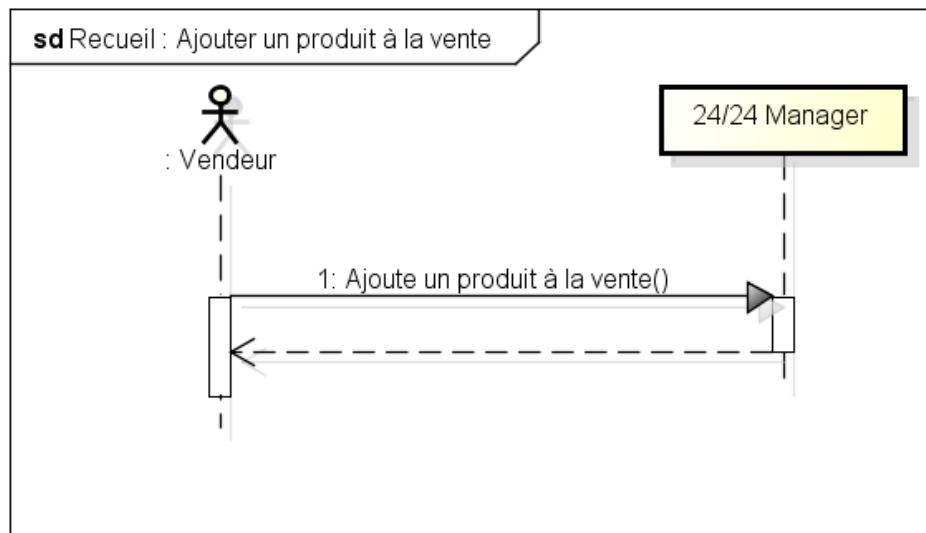


### 1.2.3. Diagrammes de séquence associés

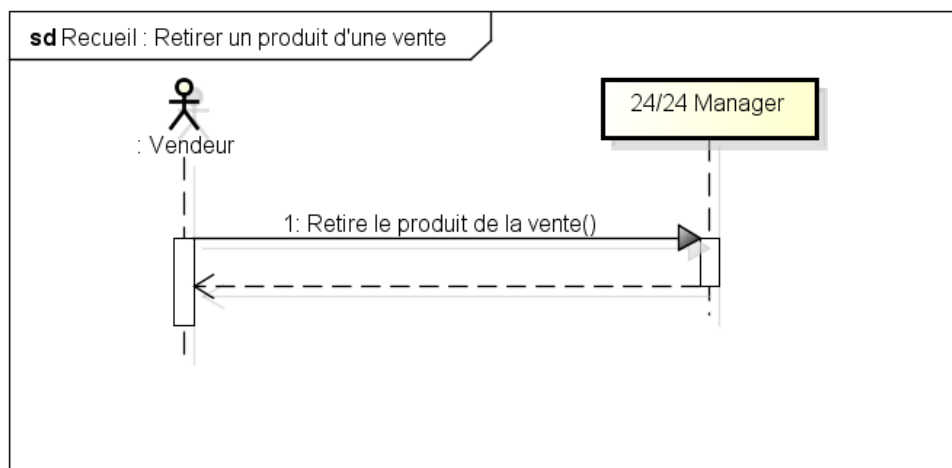
#### Afficher les stocks



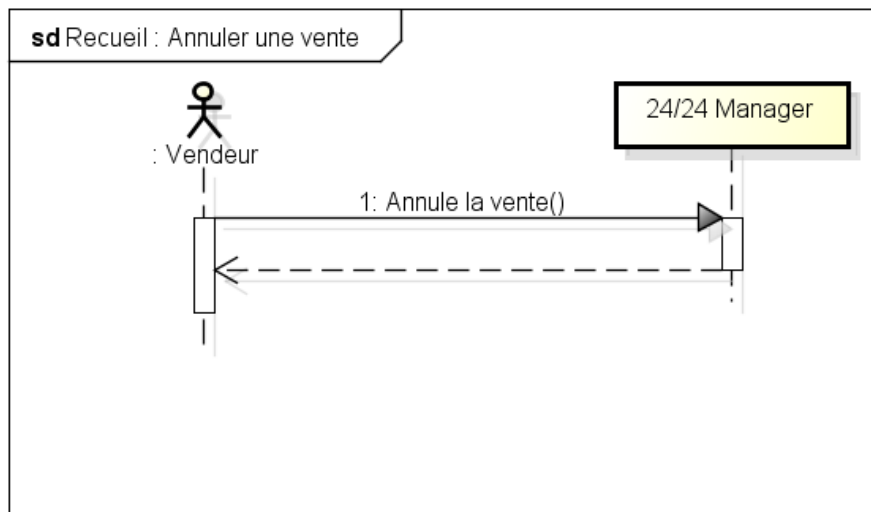
#### Ajouter un produit



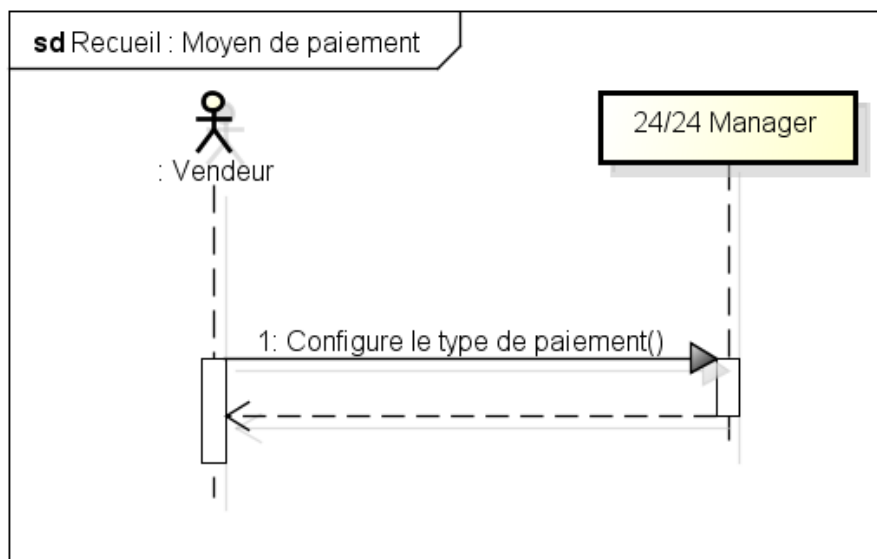
#### Retirer un produit



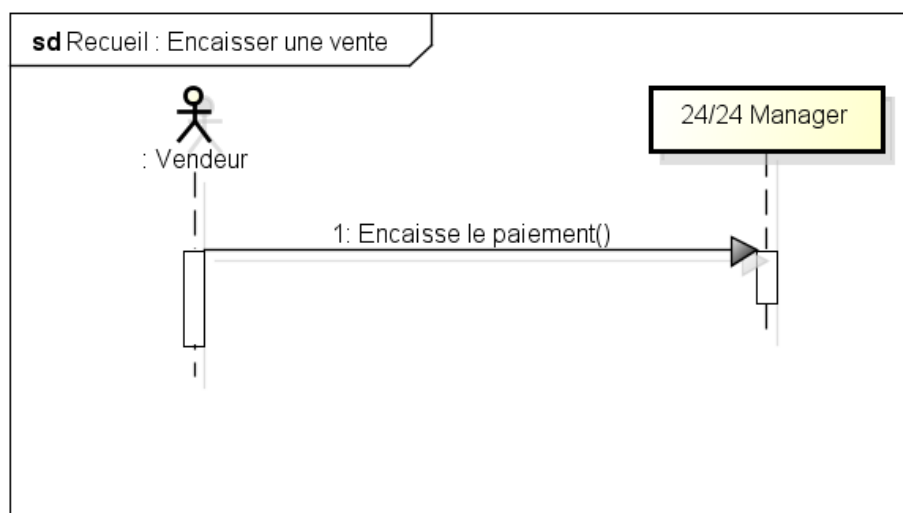
### Annuler la vente en cours



### Choisir un moyen de paiement

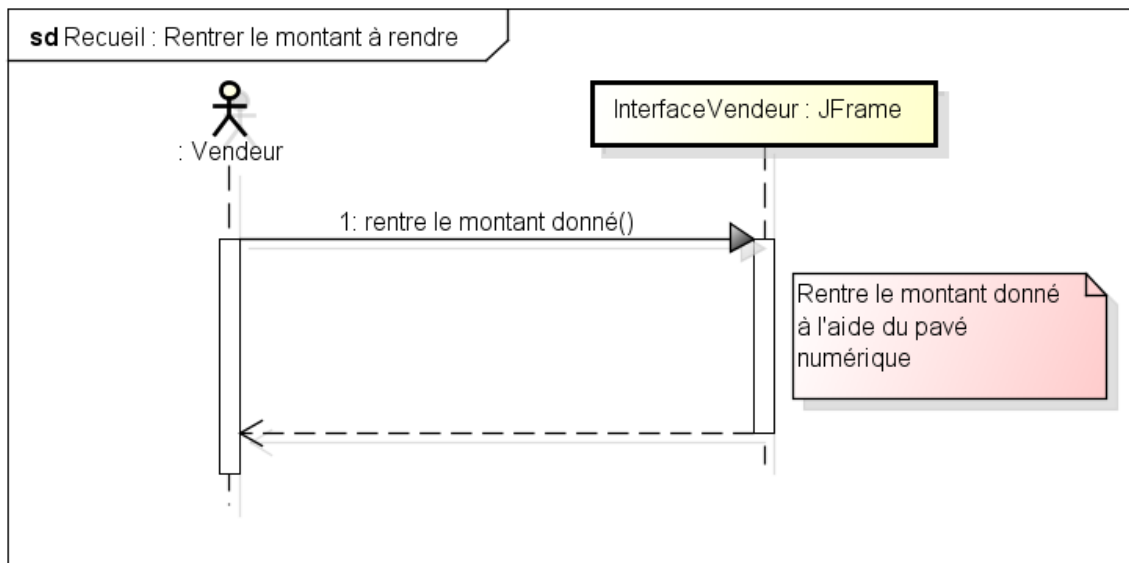


### Encaisser la vente en cours

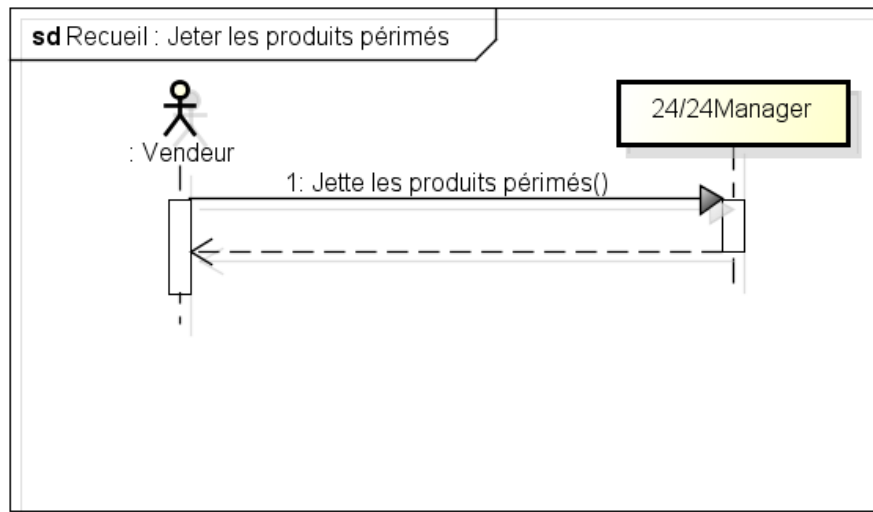




### Rentrer le montant à rendre



### Jeter les produits périmés



## 1.3. Utilisateur : employé de cuisson

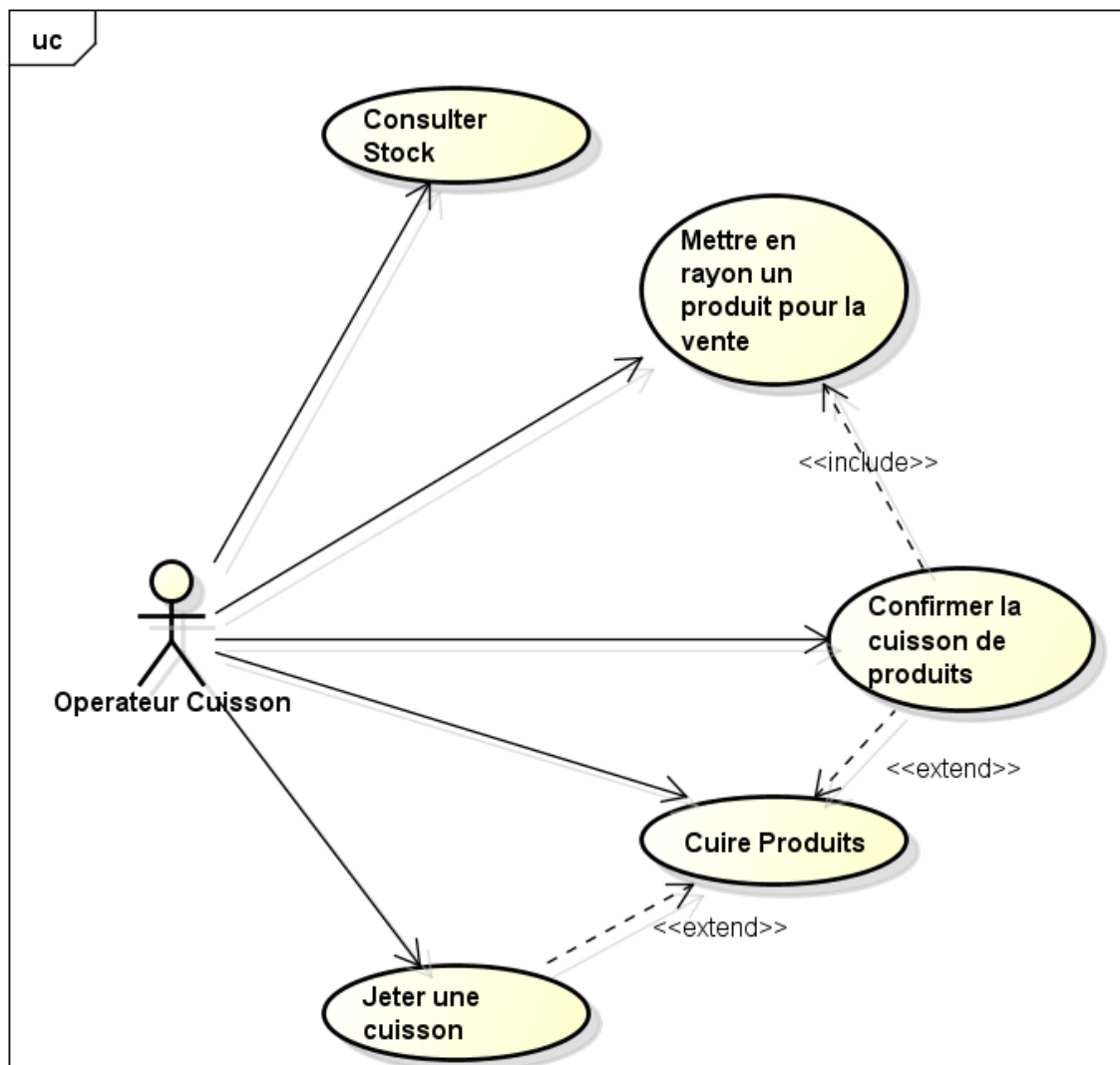
### 1.3.1. Principe générale

Un employé de cuisson possède deux tâches différentes :

- il gère la cuisson des produits et leur mise en vente
- il a aussi la possibilité de consulter le stock réfrigérateur et vente.

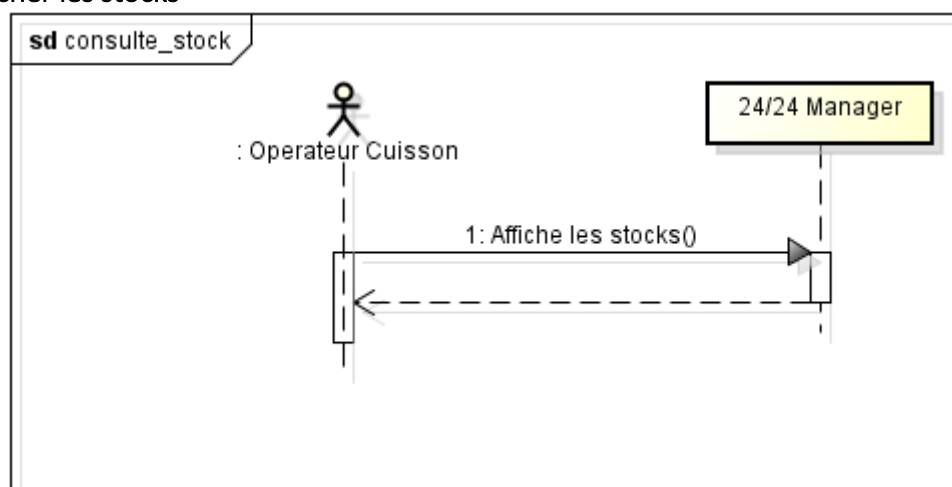
### 1.3.2. Diagramme de cas d'utilisation

En plus de pouvoir consulter les stocks, un employé de cuisson peut mettre en rayon un produit qui ne nécessite pas d'être cuit pour la vente. Pour les autres produits, il doit les mettre à cuire. Cette action lui permet d'utiliser deux fonctionnalités (Liens « extend ») qui sont la possibilité de jeter les produits qui sont non-conformes et la possibilité de confirmer une cuisson. Cette confirmation ou validation inclut le fait de mettre en rayon les produits (Lien « include »).

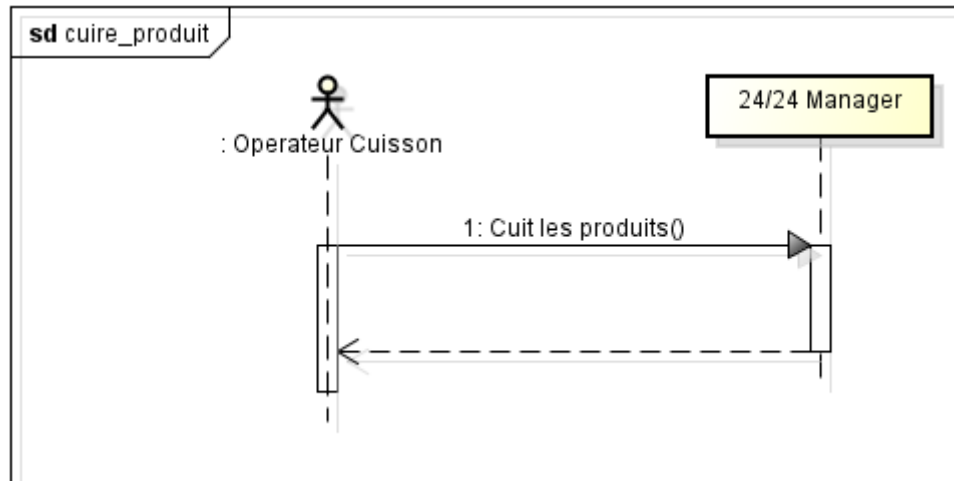


### 1.3.3. Diagrammes de séquence associés

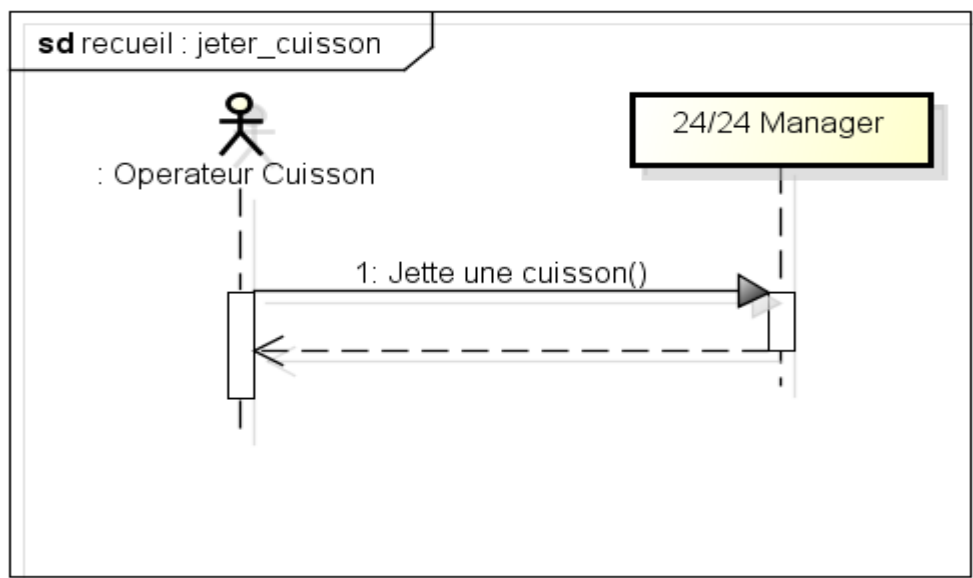
Afficher les stocks



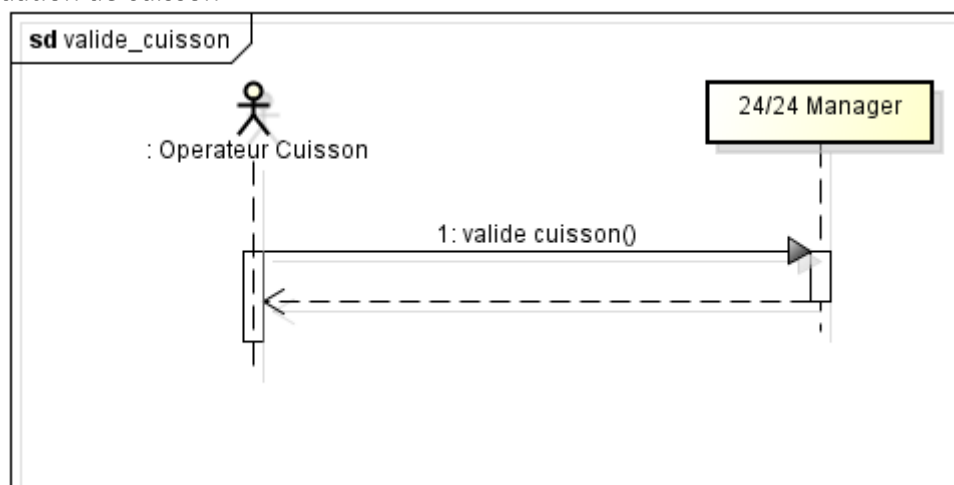
### Cuisson de produits



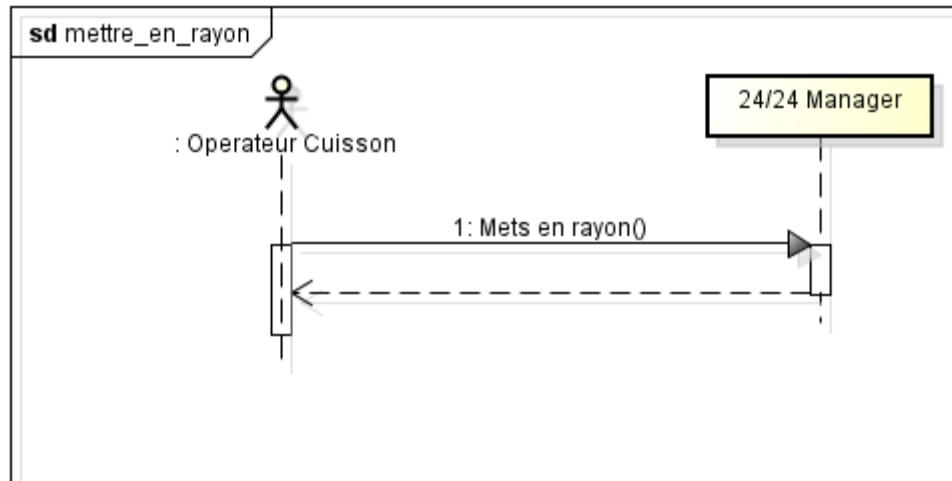
### Rejet de cuisson



### Validation de cuisson



## Mise en rayon



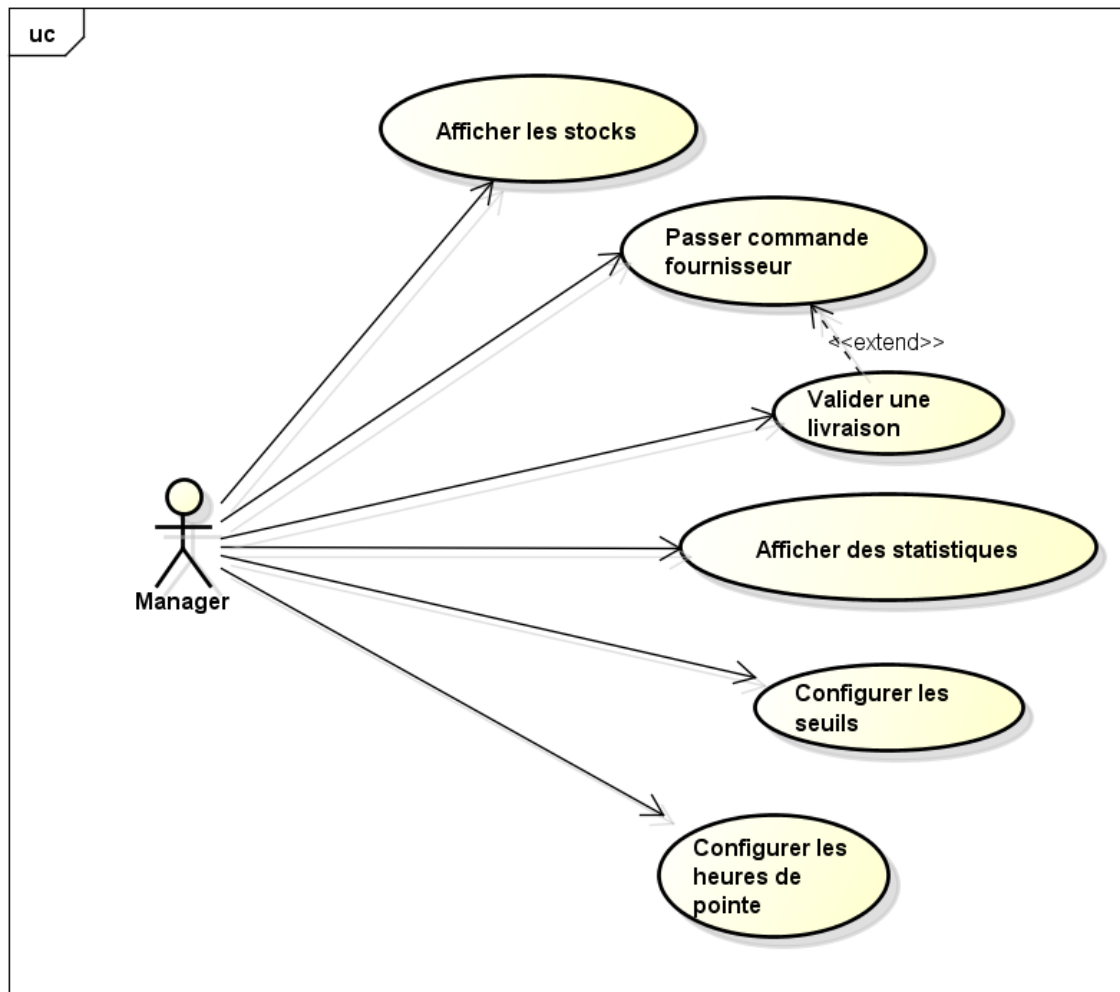
## 1.4. Utilisateur : manager

### 1.4.1. Principe générale

L'interface du manager comporte trois onglets : un pour la configuration du logiciel, un pour gérer les stocks du magasin et un pour prendre connaissance des statistiques du magasin. Dans chaque onglet des fonctionnalités seront implémentées pour répondre aux besoins exprimés par le client.

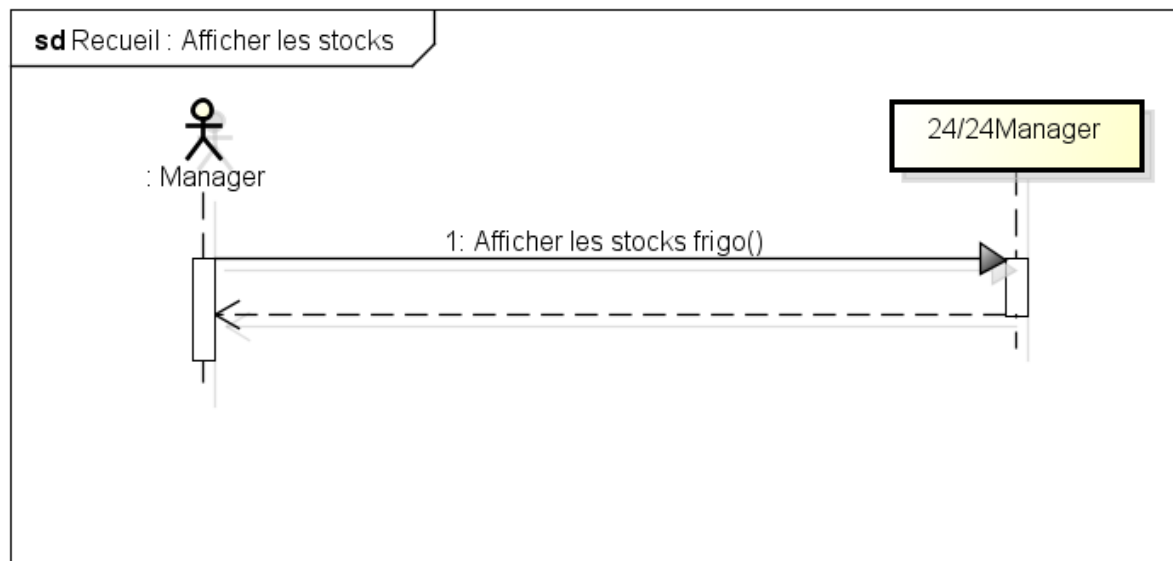
### 1.4.2. Diagramme de cas d'utilisation

Le manager peut afficher les stocks, afin d'avoir une vue d'ensemble sur son magasin et passer des commandes. Pour passer une commande, il faut obligatoirement la valider en cliquant sur le bouton correspondant. Ces deux fonctions ne peuvent fonctionner l'une sans l'autre, ce qui justifie l'inclusion. Il modifie aussi les différents seuils et éléments configurables du logiciel. Le détail sera donné lors du diagramme de séquence correspondant.

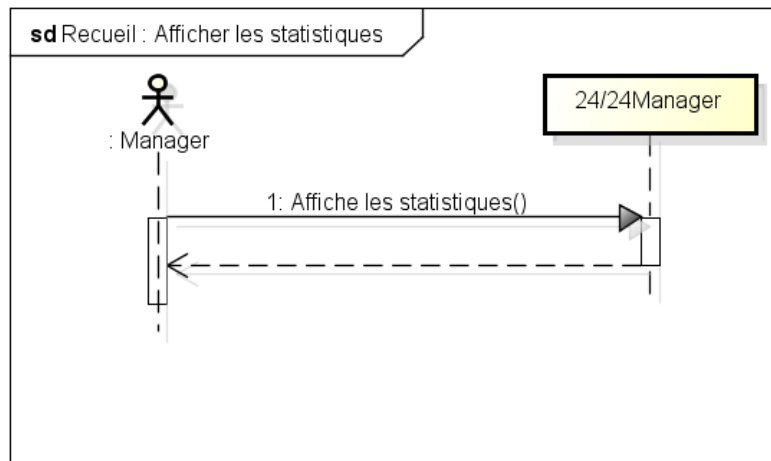


#### 1.4.3. Diagrammes de séquence associés

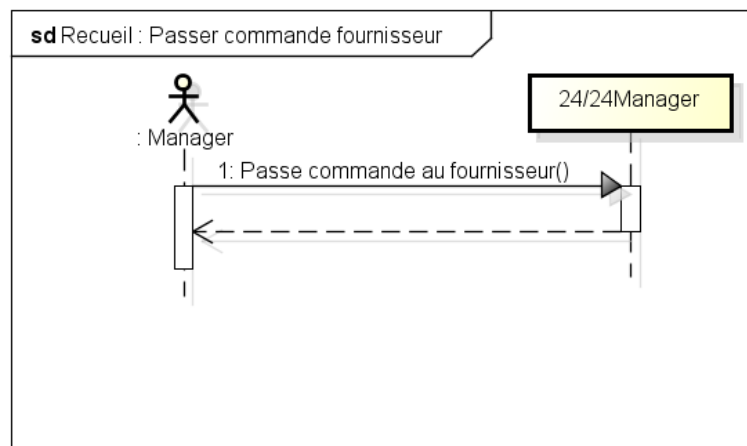
Afficher les stocks du magasin (onglet Gestion des stocks)



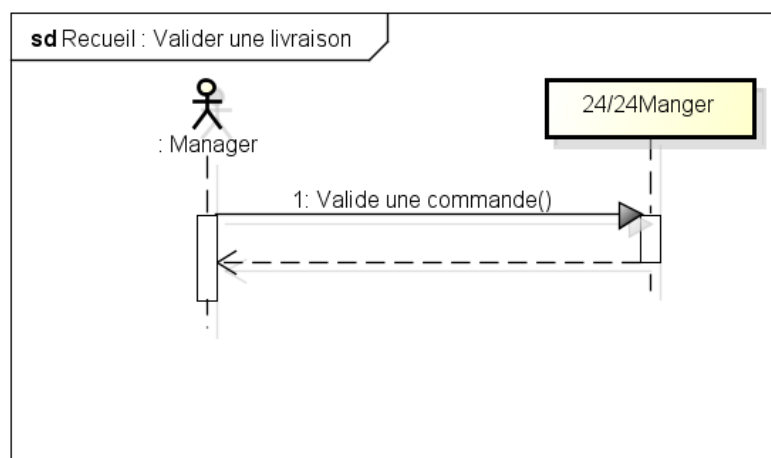
### Consulter les statistiques du magasin (onglet Statistique)



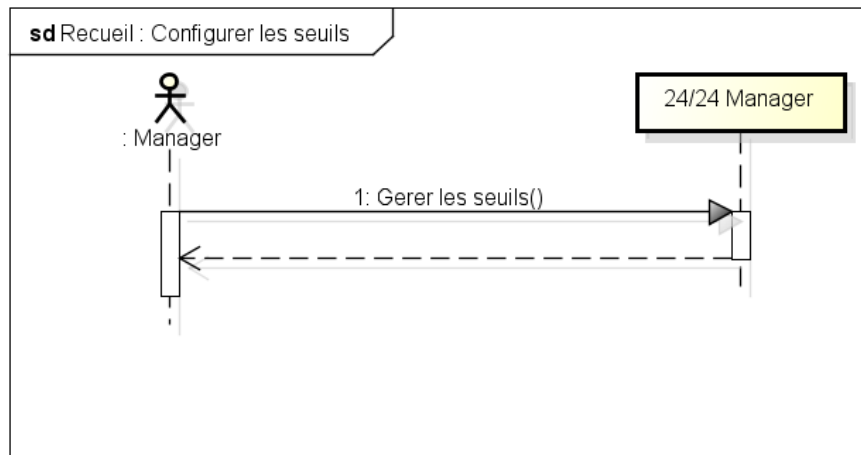
### Passer une commande au fournisseur (onglet Gestion des stocks)



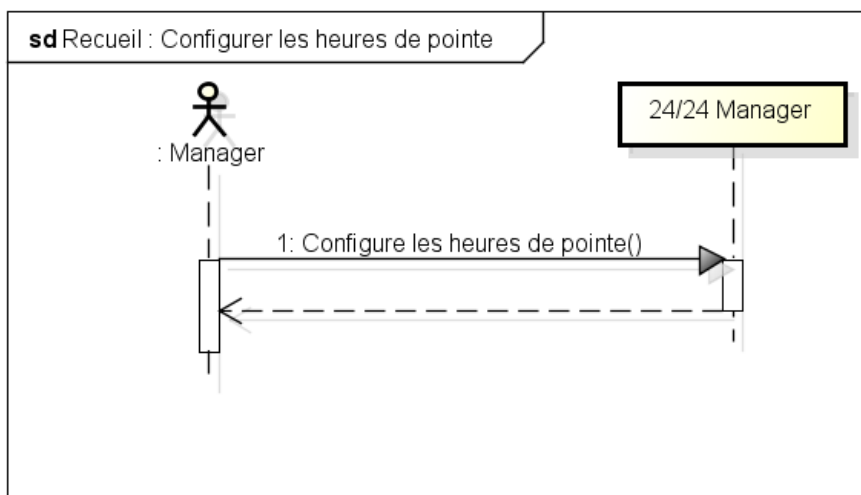
### Validation d'une livraison (onglet Gestion des stocks)



**Configuration des différents seuils et éléments du logiciel :** seuils de stocks, seuils de produits avant de lancer une cuisson hors heure de pointe et en heure de pointe, nombre de produits à cuire hors heure de pointe et en heure de pointe et enfin, configuration des prix par produit. (**Onglet Configuration**)

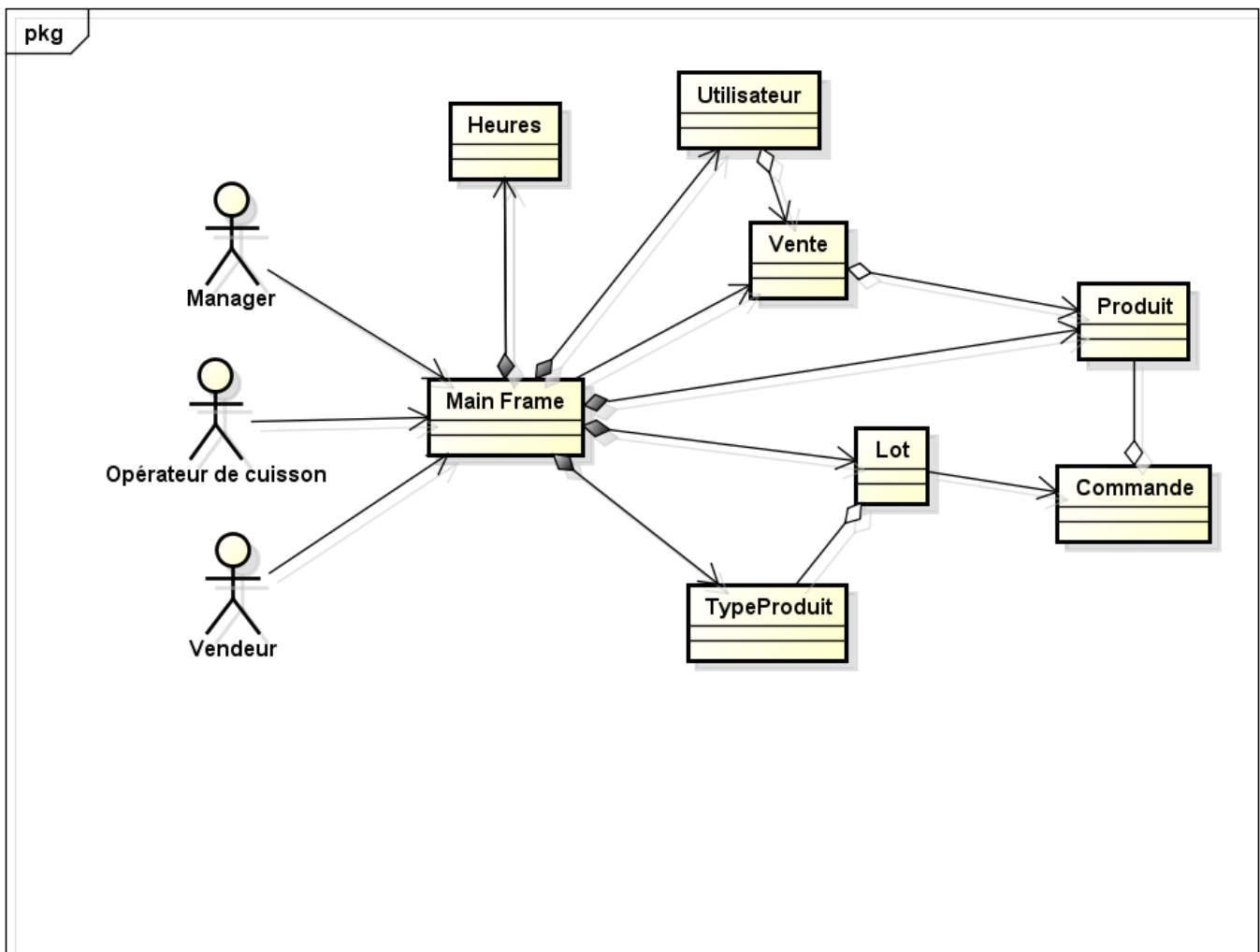


**Gestion des heures de pointes (Onglet Configuration)**



## 2. Analyse

### 2.1. Diagramme de classe



### 2.2. Descriptifs des classes

**Heure** : cette classe permet d'avoir les outils permettant de gérer les heures de pointe du magasin.

**TypeProduit** : cette classe représente chaque type de produit, elle contient les éléments qui définissent les types de produit vendus.

**Statistique** : cette classe a pour rôle de donner les outils permettant de réaliser les statistiques pour le manager.

**Lot** : cette classe permet de gérer les produits des stocks, en les identifiant par lot lors de commande de fournisseur et cela jusqu'à la cuisson des produits.



**Commande** : cette classe permet de créer/gérer les commandes faites par le manager au fournisseur de produit.

**Produit** : cette classe « produit » permet d'identifier chaque produit de façon unique afin de pouvoir gérer les dates de péremption (date de valider de produit après cuisson).

**24/24Manager** : cette classe représente le système, elle permet de lier contenu du système à l'interface de celui-ci.

**Utilisateur** : cette classe représente un utilisateur du système, elle est utilisée pour se connecter au logiciel ou encore pour identifier quel utilisateur a réalisé une vente.

**Vente** : cette classe représente une vente à un client. Elle contient tous les outils permettant de faire la vente (encaissement, création et validation de vente...)

**InterfaceBdd (non représenté sur le diagramme)** : cette classe prépare les requêtes SQL à envoyer à la Base de Données.

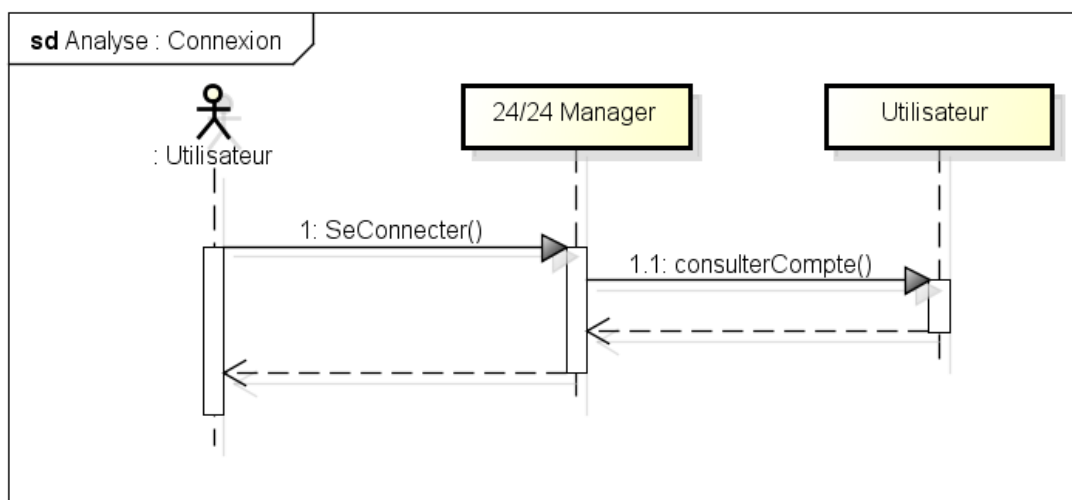
**AccesBdd (non représenté sur le diagramme)** : cette classe permet d'accéder à la base de données, elle permettra de récupérer les données et de modifier les enregistrements de la base de données.

## 2.3. Diagramme de séquence

### 2.3.1. Connexion / Déconnexion / Quitter

Le diagramme présent dans cette partie présente les étapes nécessaires à la connexion d'un utilisateur. Les cas d'utilisation de la déconnexion et de quitter le logiciel ne sont pas présentés ici car leurs traitements respectifs sont uniquement graphiques. Ces derniers seront donc expliqués dans la partie Conception.

#### Connexion

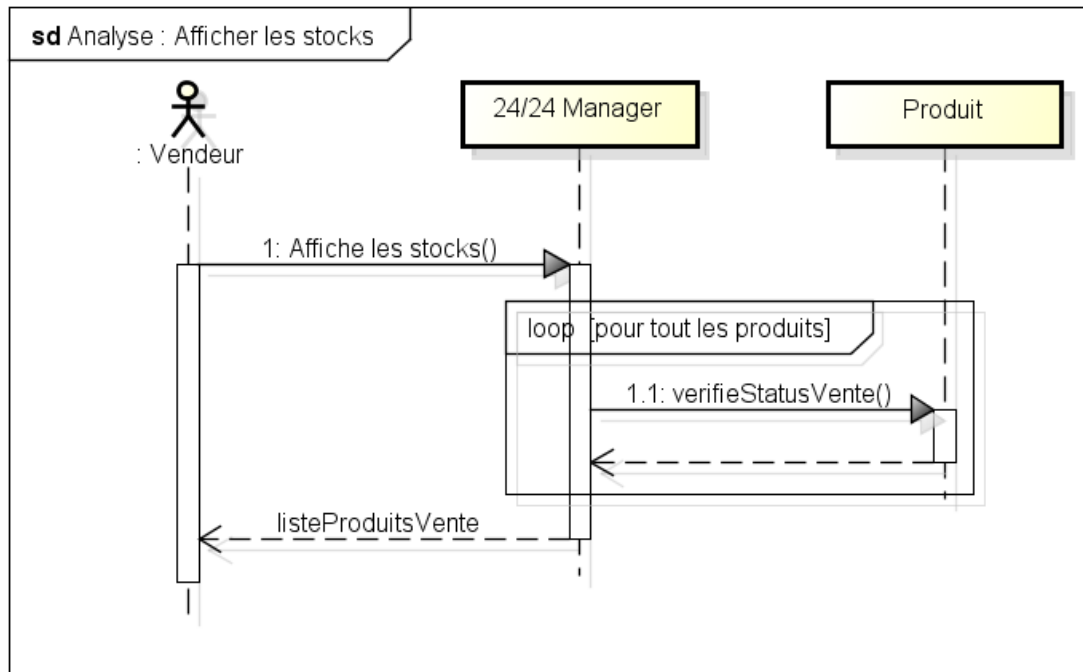


### 2.3.2. Interface vendeur

Les diagrammes présents dans cette partie présentent les étapes nécessaires pour réaliser les cas d'utilisation relatifs au rôle du vendeur.

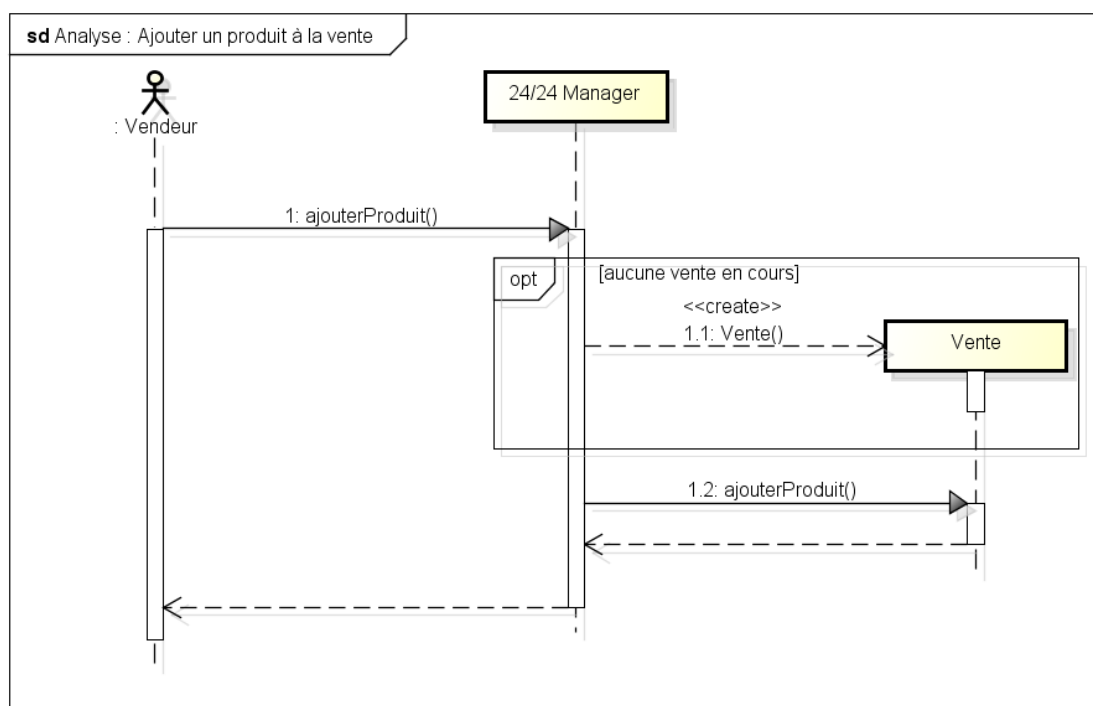
#### Afficher les stocks

Une fois connecté, un vendeur peut directement visualiser le stock en vente.



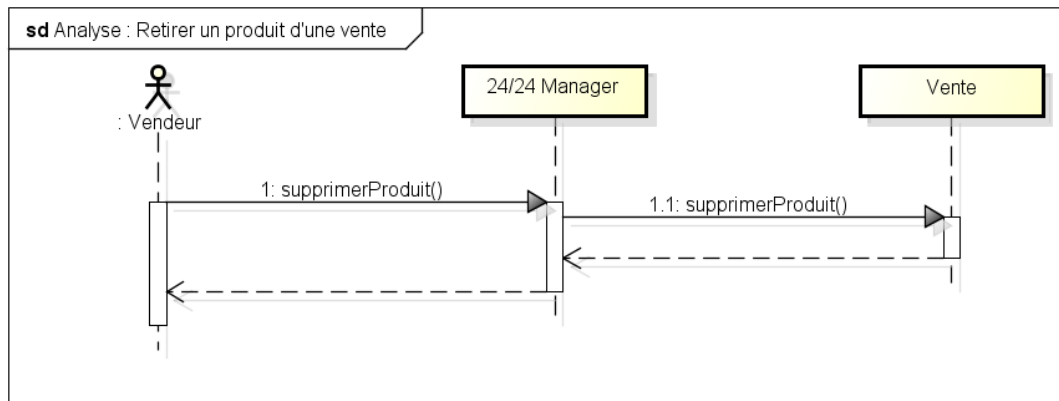
#### Ajouter un produit

Cette action permet au vendeur d'ajouter un produit à la vente en cours.



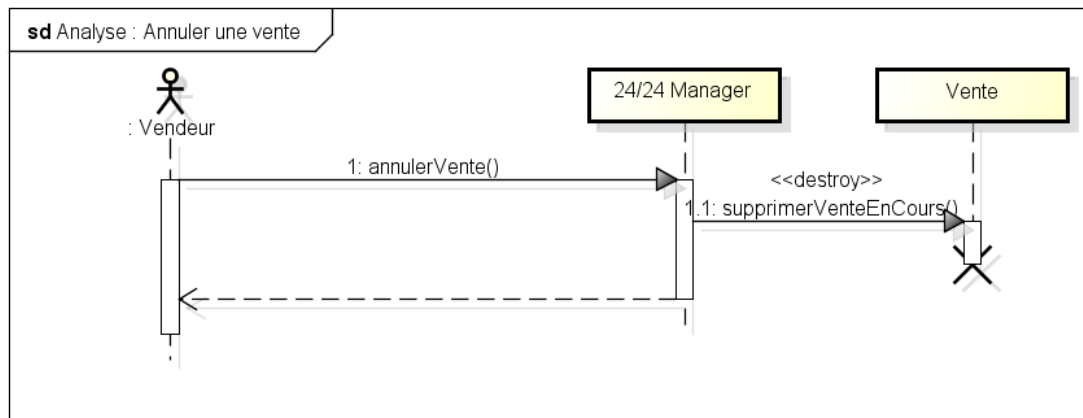
### Retirer un produit

Cette action permet au vendeur de retirer un produit de la vente en cours.



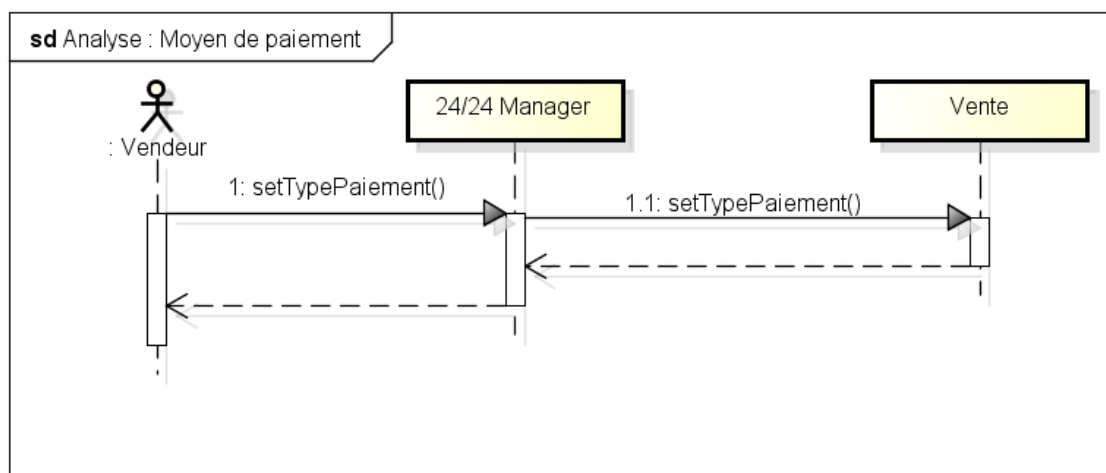
### Annuler la vente en cours

Cette action permet au vendeur d'annuler la vente en cours.



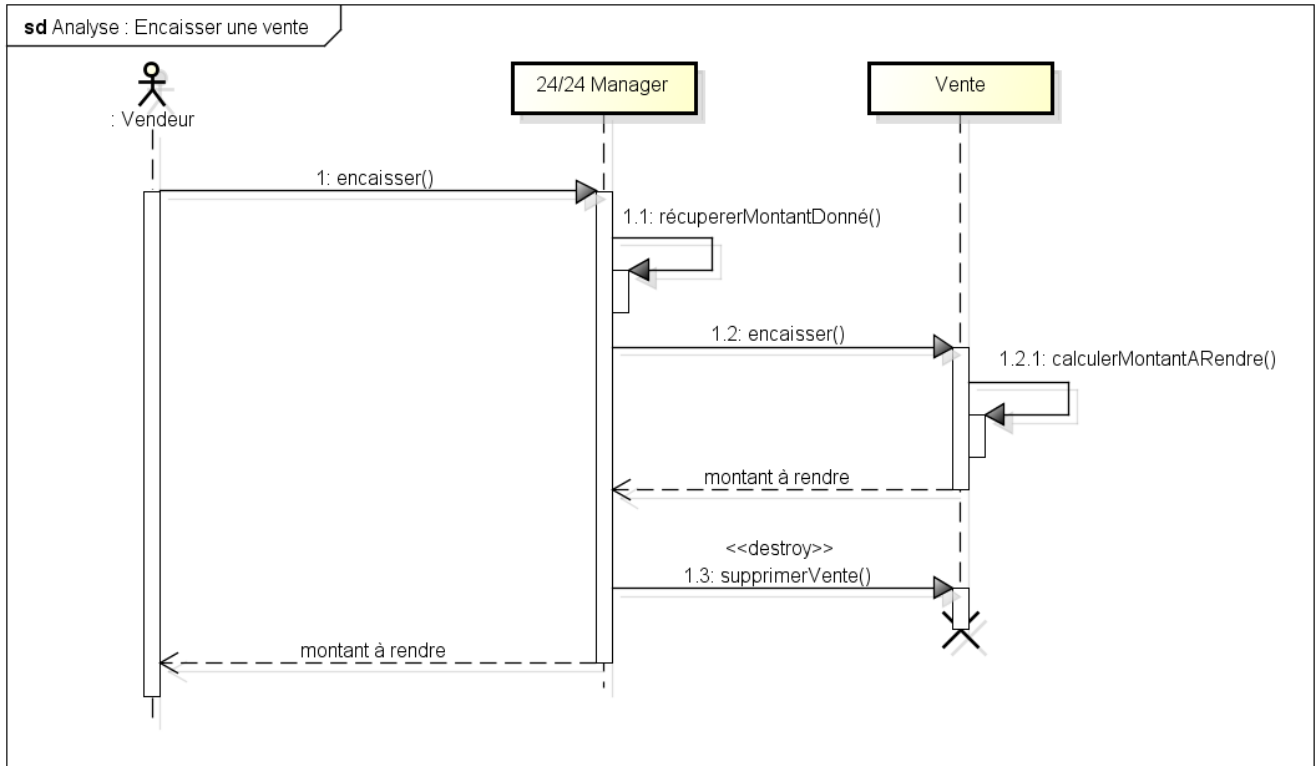
### Choisir un moyen de paiement

Cette action permet au vendeur de choisir un type de paiement pour la vente en cours.



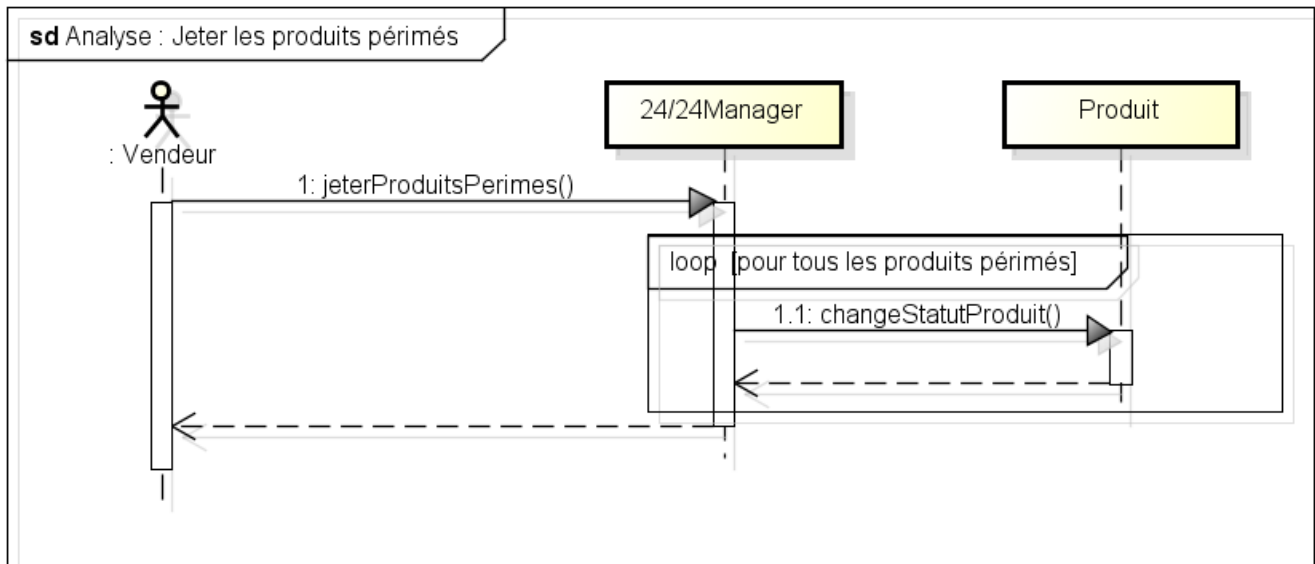
### Encaisser la vente en cours

Cette action permet au vendeur d'encaisser une vente et de connaître le montant à rendre au client.



### Jeter les produits périmés

Cette action permet au vendeur de jeter les produits qui sont périmés.

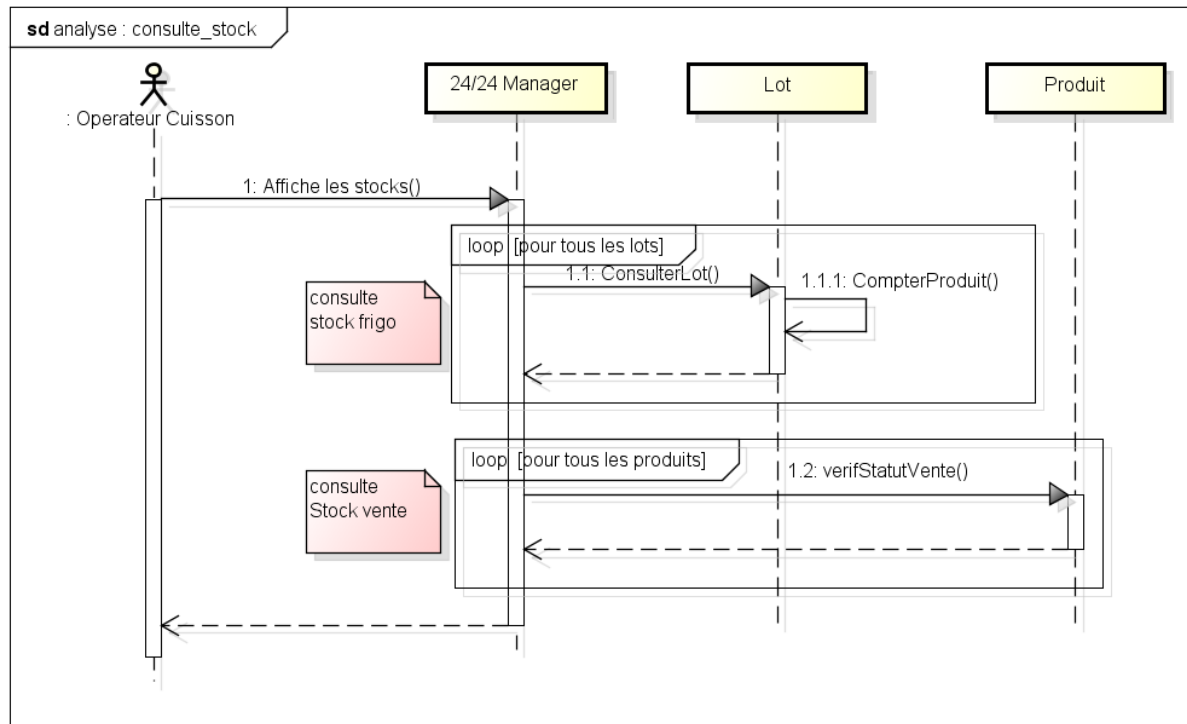


### 2.3.3. Interface employé de cuisson

Les diagrammes présents dans cette partie présentent les étapes nécessaires pour réaliser les cas d'utilisation relatifs au rôle de l'employé de cuisson.

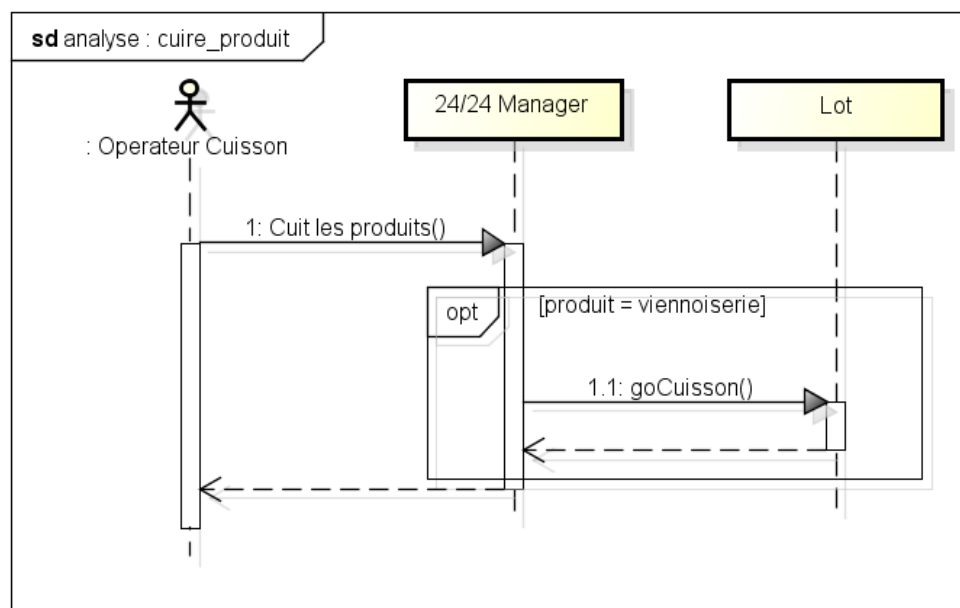
#### Afficher les stocks

Une fois connecté, un employé de cuisson peut directement visualiser le stock en vente et le stock dans le réfrigérateur.



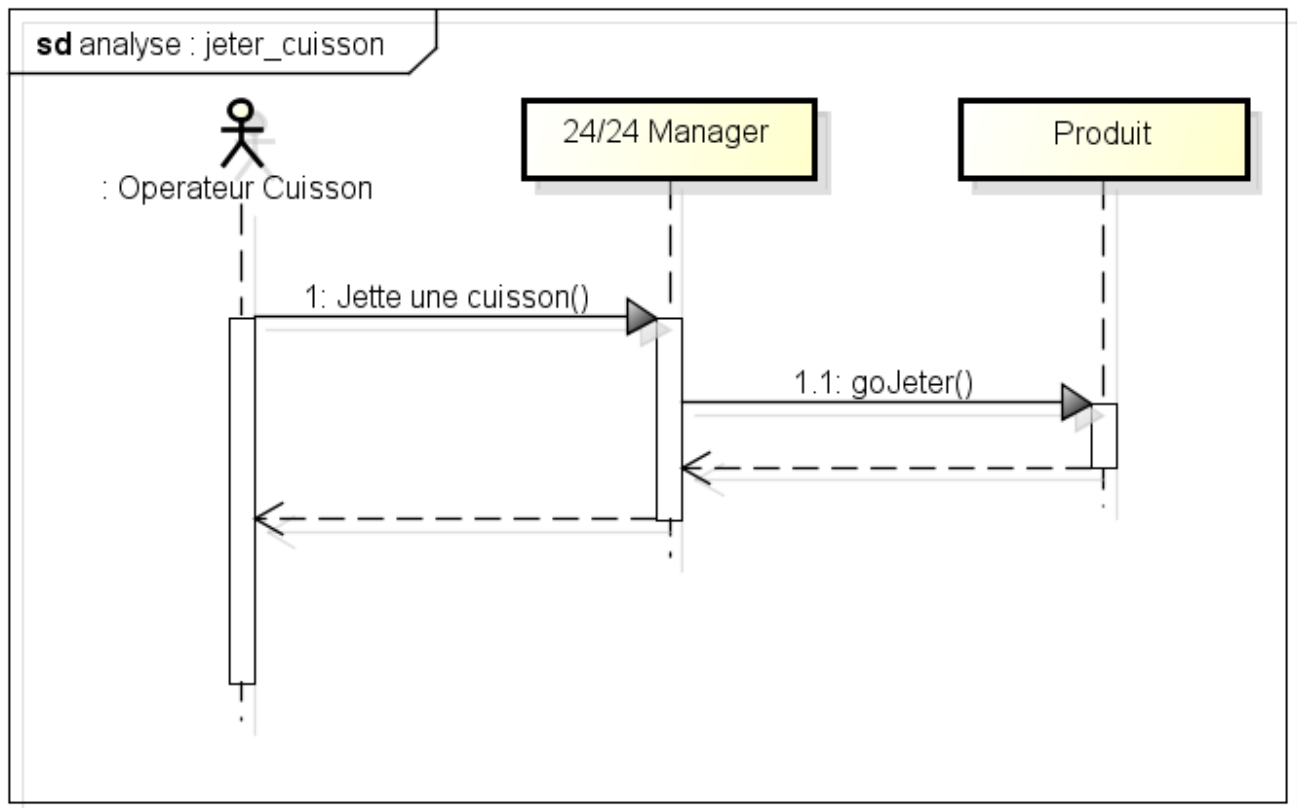
#### Cuisson de produits

Cette action est disponible pour les pains et les viennoiseries.



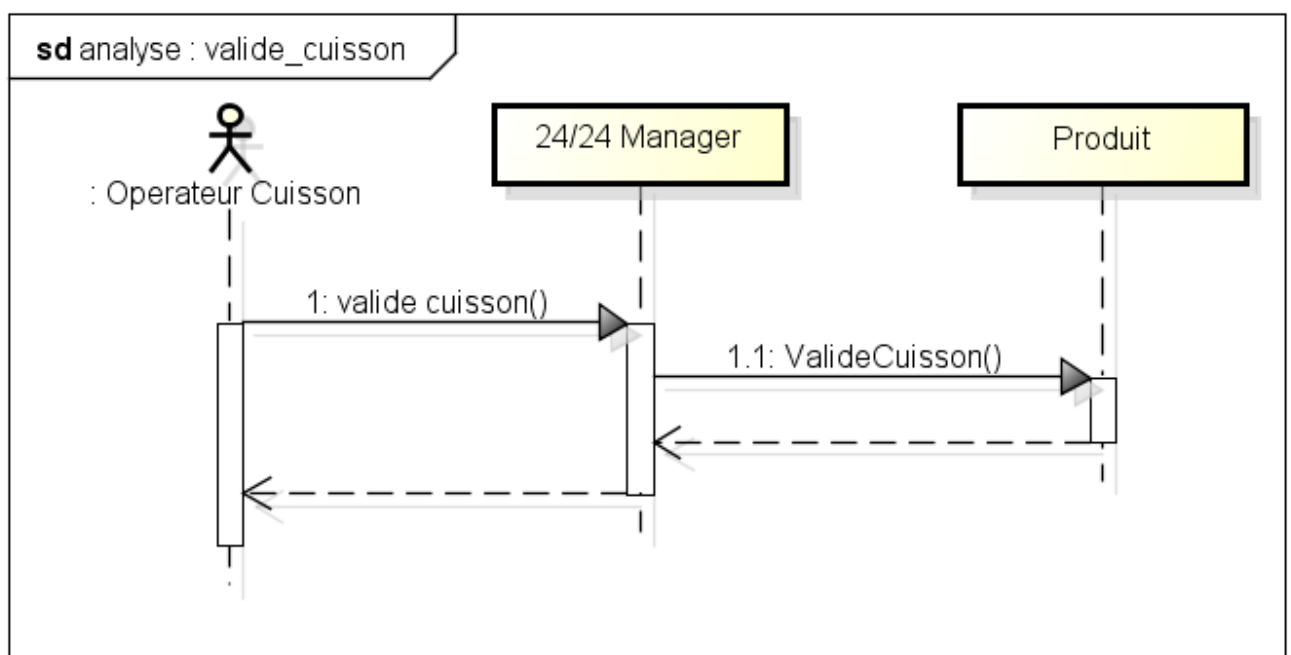
### Rejet de cuisson

Cette action permet à l'employé de cuisson de jeter les cuissons qui sont non conformes.



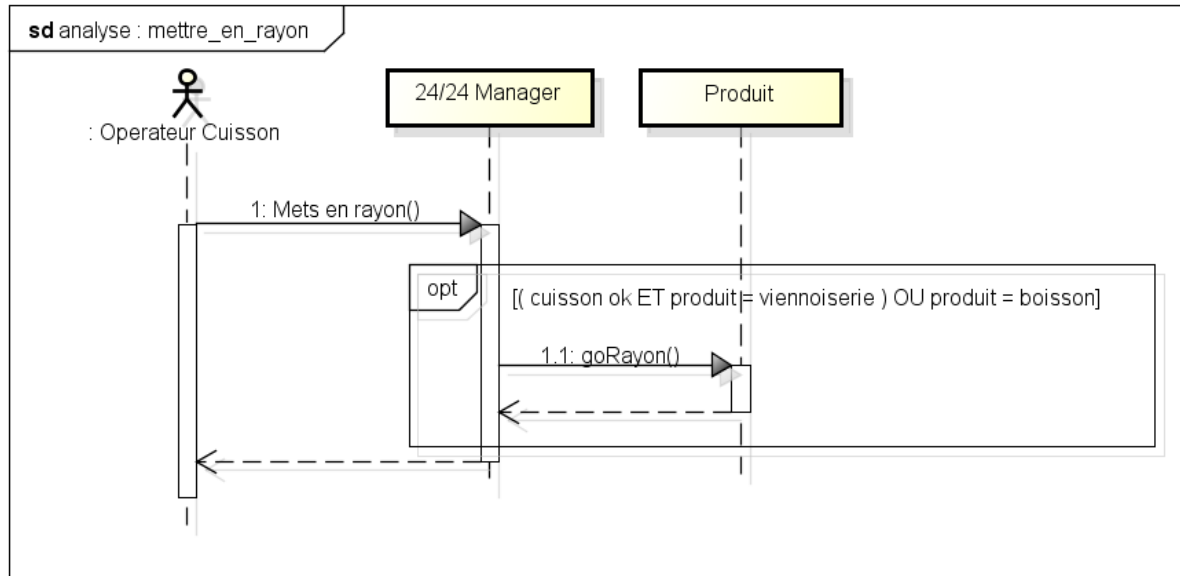
### Validation de cuisson

Cette action permet à l'employé de cuisson de valider la cuisson de produits



### Mise en rayon

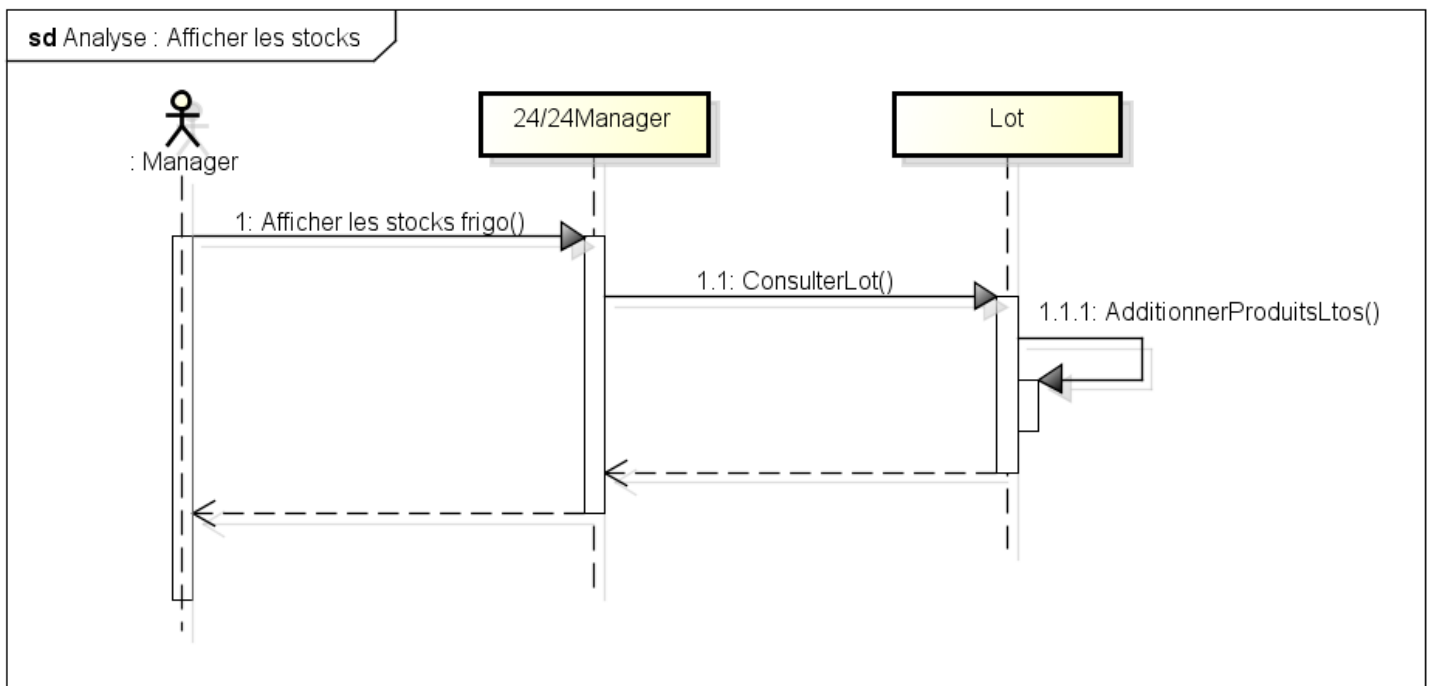
A la fin de la cuisson de produits, l'employé de cuisson peut les mettre en rayon si la cuisson est validée. Les boissons sont directement mises en rayon.



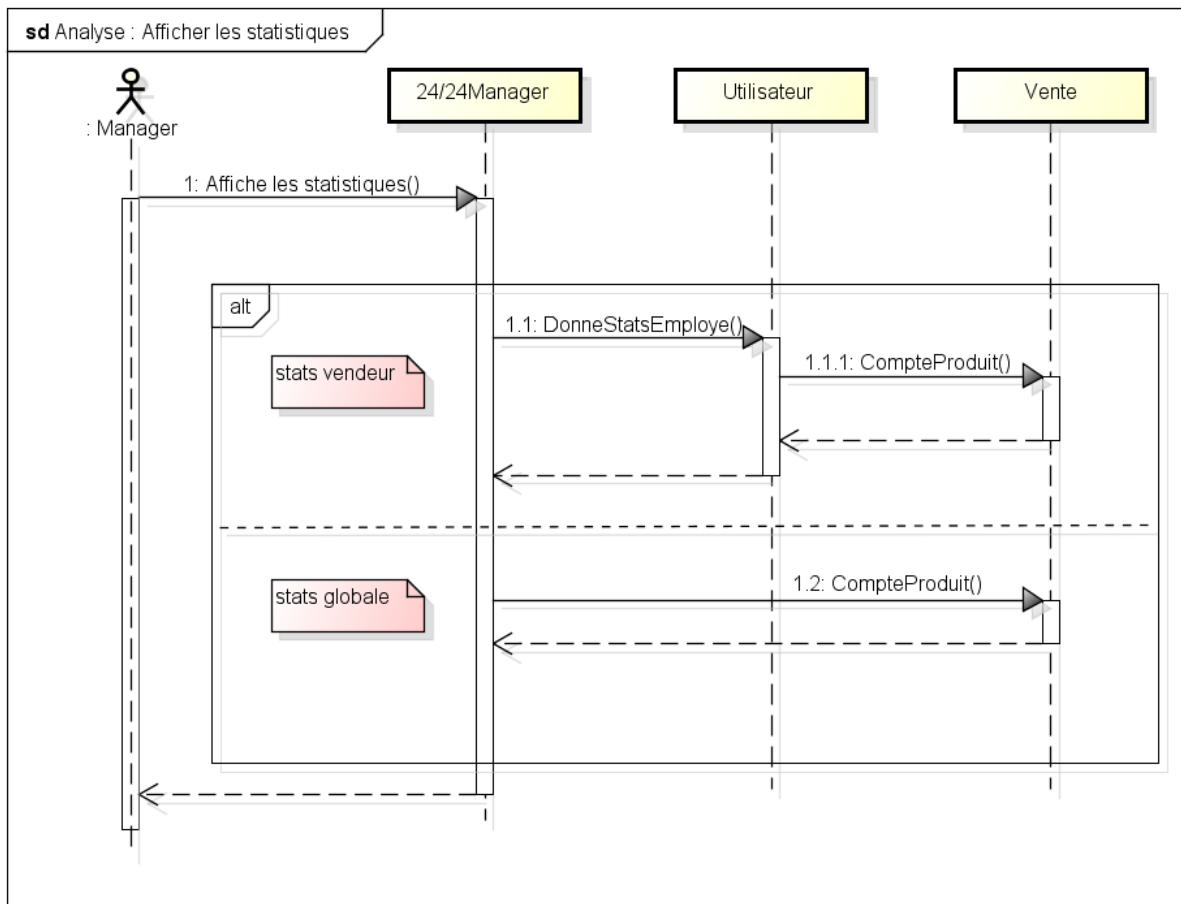
### 2.3.4. Interface manager

Les diagrammes de séquences présentés ci-dessous sont composés des diagrammes du recueil de besoin, un peu plus détaillés.

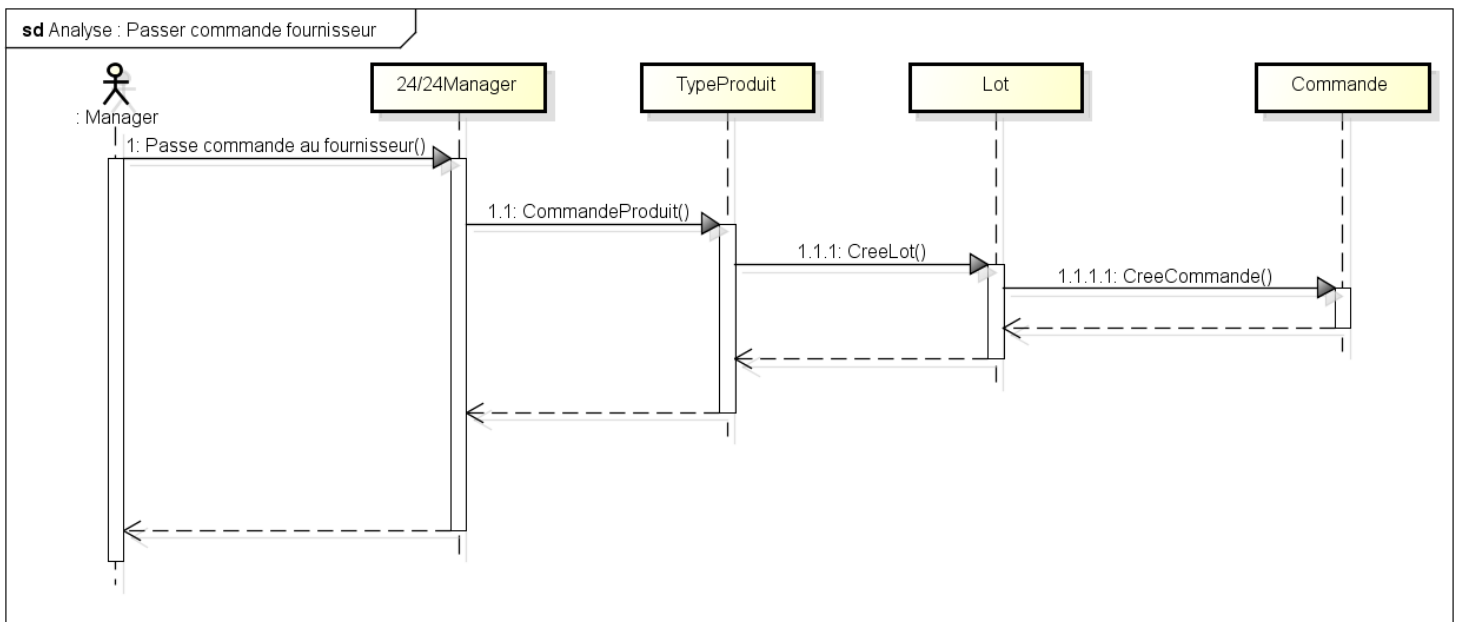
#### Afficher les stocks du magasin (onglet Gestion des stocks)



## Consulter les statistiques du magasin (onglet Statistique)

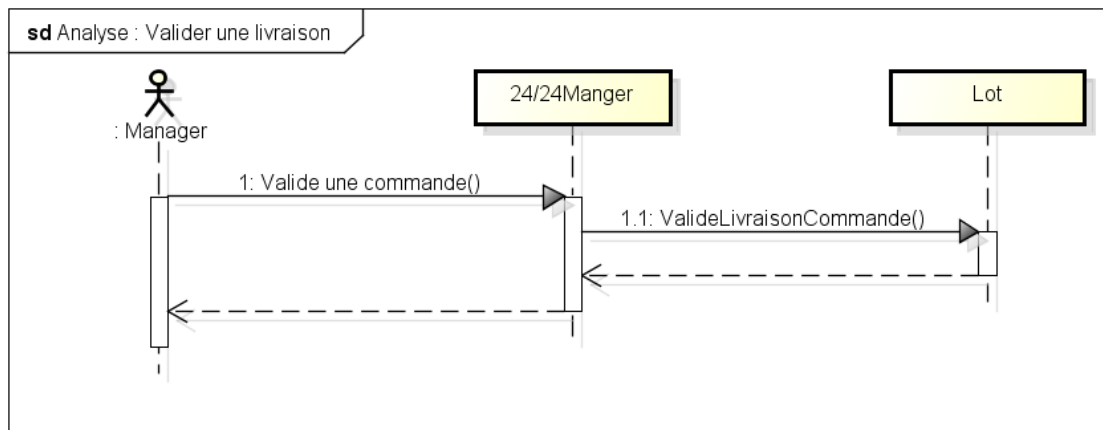


## Passer une commande au fournisseur (onglet Gestion des stocks)

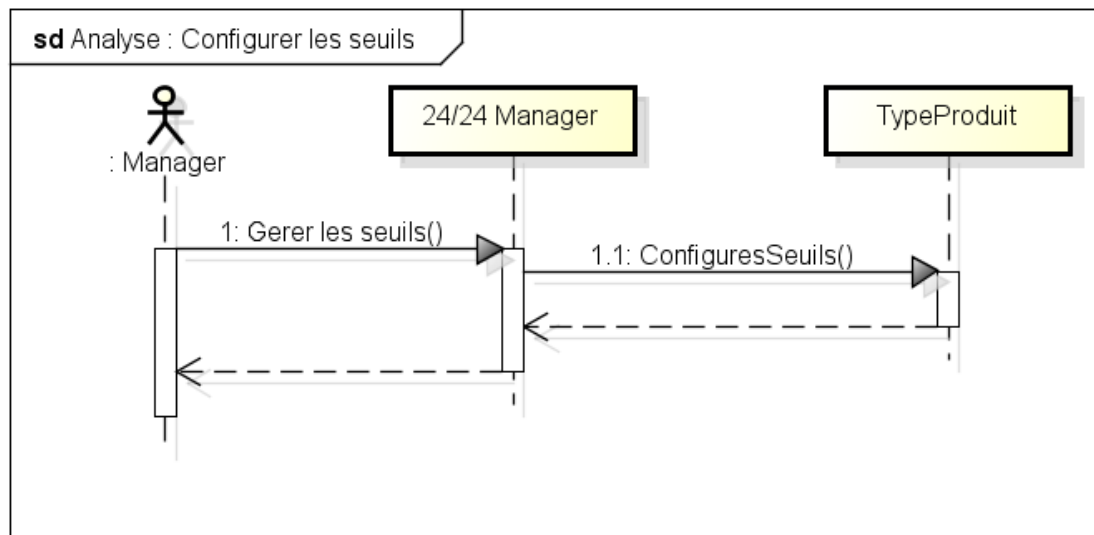




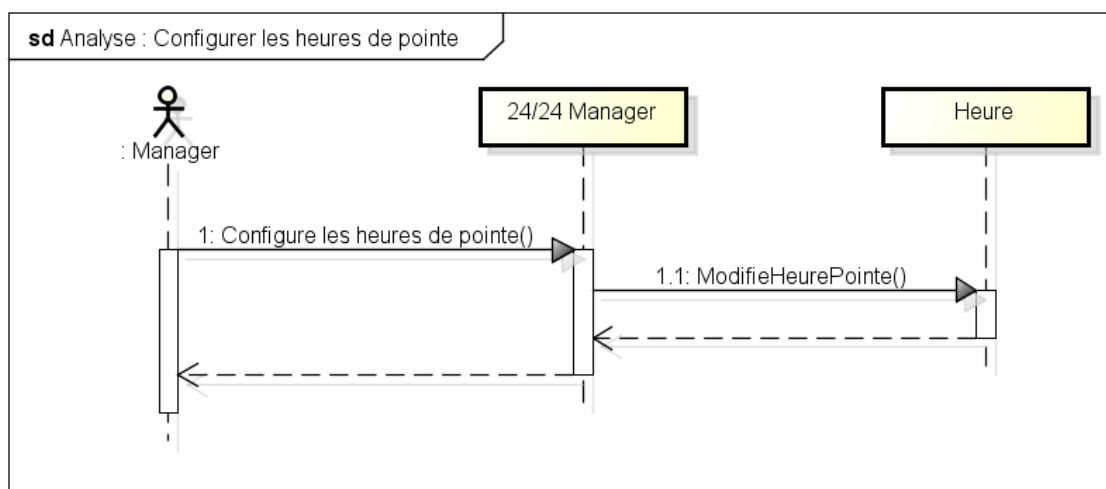
### Validation d'une livraison (onglet Gestion des stocks)



### Configuration des différents seuils et éléments du logiciel (Onglet Configuration)



### Gestion des heures de pointes (Onglet Configuration)



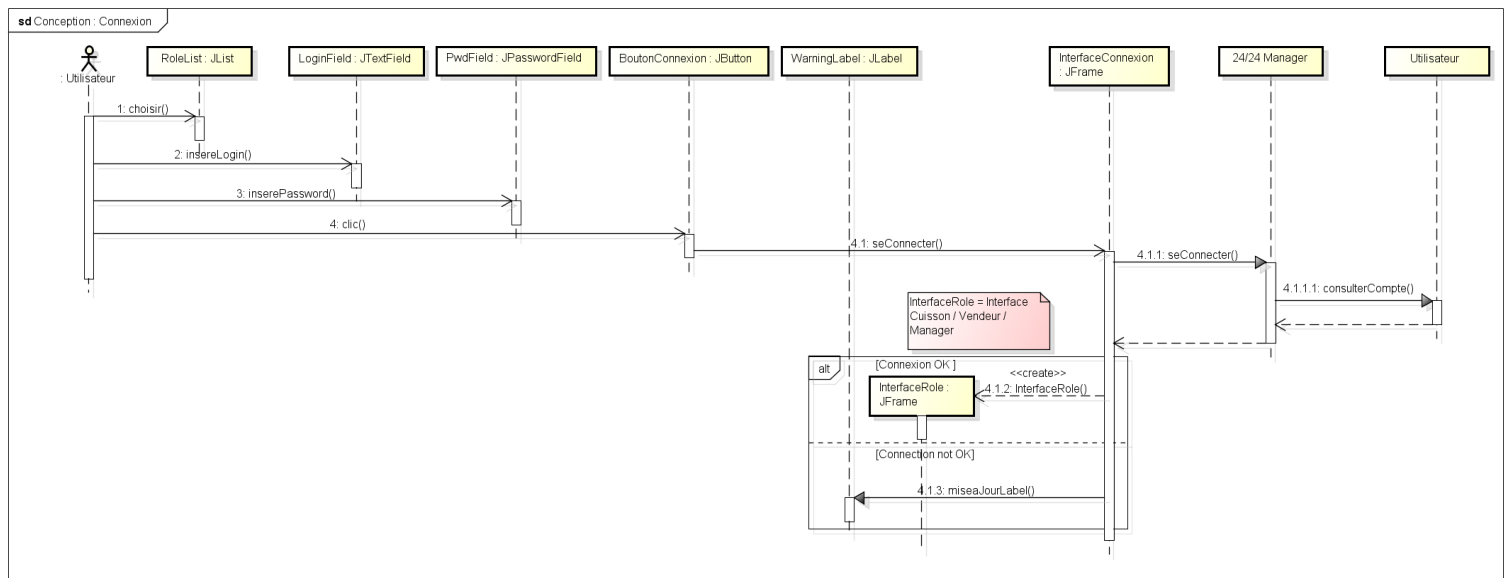
### 3. Conception

#### 3.1. Complétion de gauche à droite

##### 3.1.1. Connexion / Déconnexion / Quitter

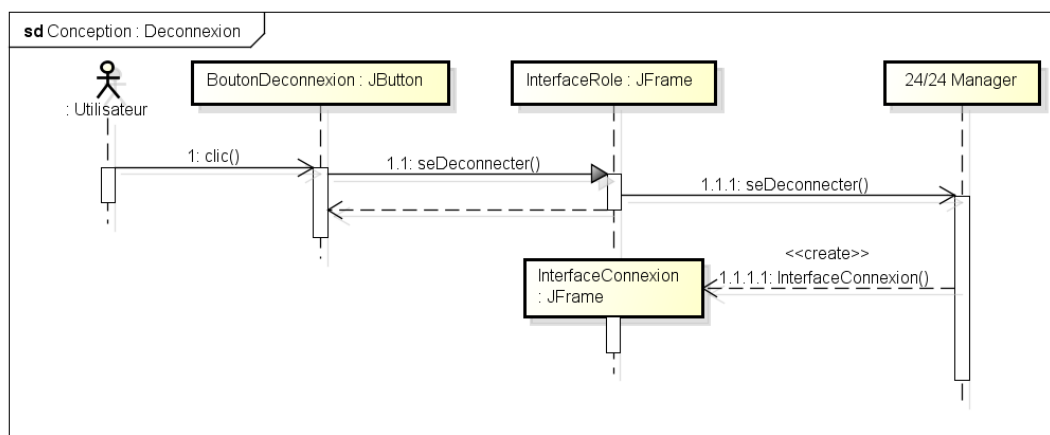
###### Connexion

La connexion nécessite de fournir un identifiant, un mot de passe et de sélectionner un rôle. Ensuite l'utilisateur clique sur le bouton de connexion et le logiciel cherche parmi les utilisateurs si l'un d'eux correspond aux données saisies. Si c'est le cas, le contrôleur ouvre l'interface correspondante, sinon il met à jour un label d'erreur sur l'interface de connexion.



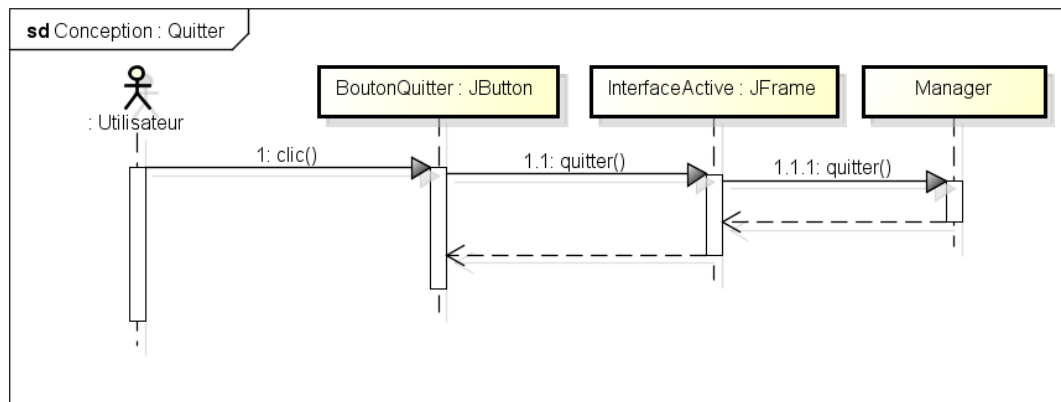
###### Déconnexion

Lors d'une déconnexion, l'utilisateur clique sur le bouton correspondant qui permet de lancer la déconnexion au niveau du contrôleur. Celui-ci va fermer l'interface active et ouvrir l'interface de connexion.



## Quitter

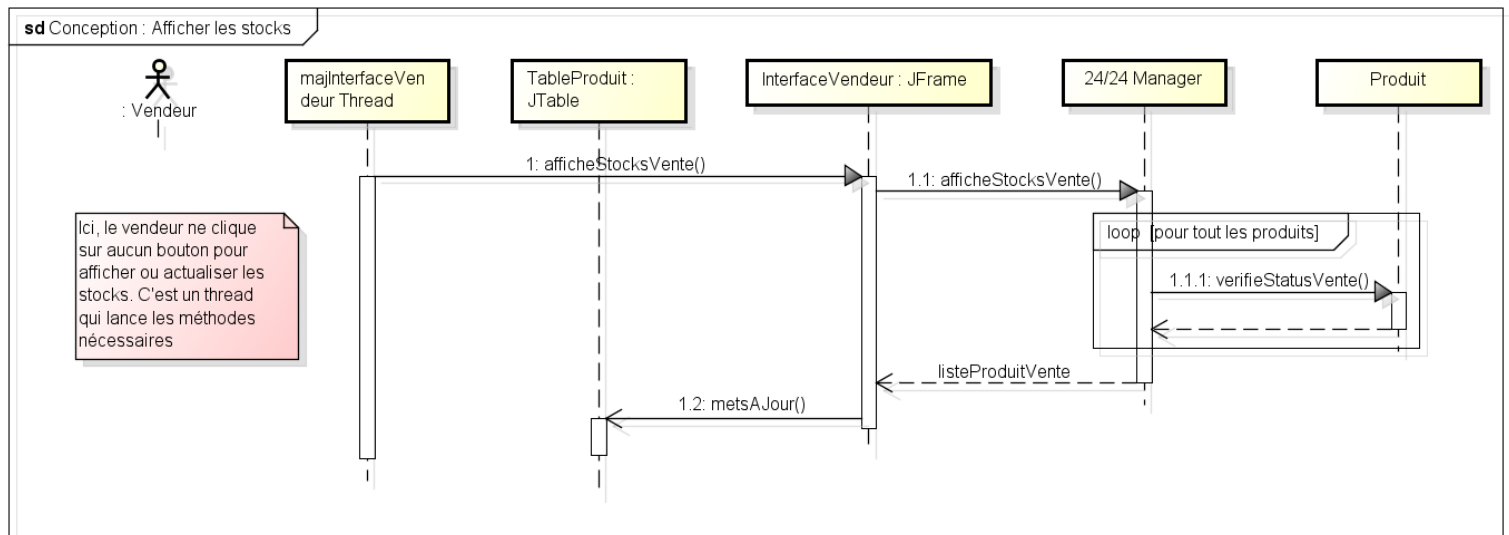
Lorsqu'un utilisateur clique le bouton « Quitter », la classe 24/24Manager se charge de fermer la fenêtre active et de quitter le logiciel.



### 3.1.2. Interface vendeur

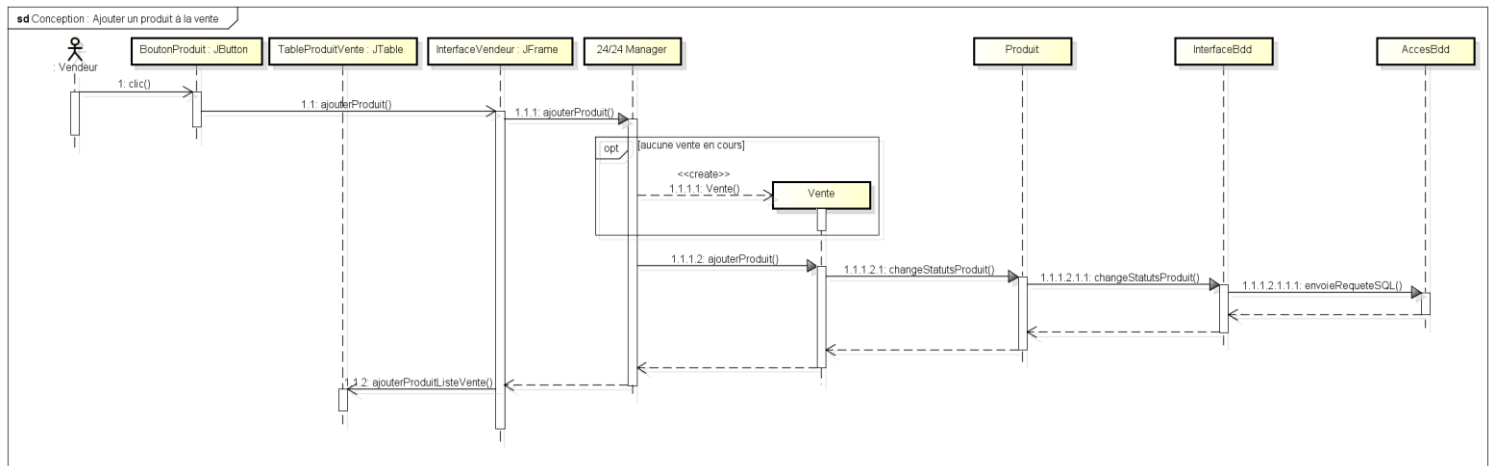
#### Afficher les stocks

Lorsqu'un utilisateur se connecte en tant que vendeur, il a accès directement à une visualisation en temps réel des stocks concernant les produits en vente grâce à l'interface vendeur. Pour cela, il n'a pas à réaliser d'action directement sur le logiciel car c'est un thread qui réalise ce travail. Ce thread va lancer l'action de récupérer les produits dont l'état est « en vente ». L'interface se chargera de mettre à jour la table contenant les différents produits en grisant les types dont le nombre de produit en vente est nul.



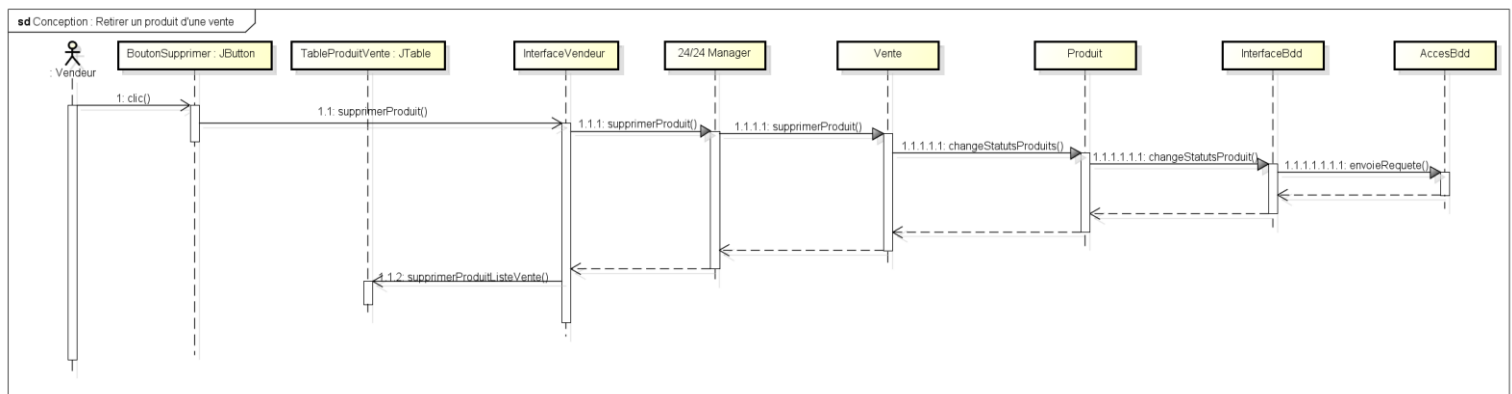
#### Ajouter un produit

Sur la gauche de l'interface, l'utilisateur a accès aux différents types de produit disponible via des boutons. Cliquer sur les boutons permet d'ajouter un produit du même type que le bouton à la vente en cours et de modifier son statut en base de données. Si aucune vente n'est en cours, elle est d'abord créée par le contrôleur 24/24Manager. Pour finir, l'interface met à jour la liste des produits de la vente en cours.



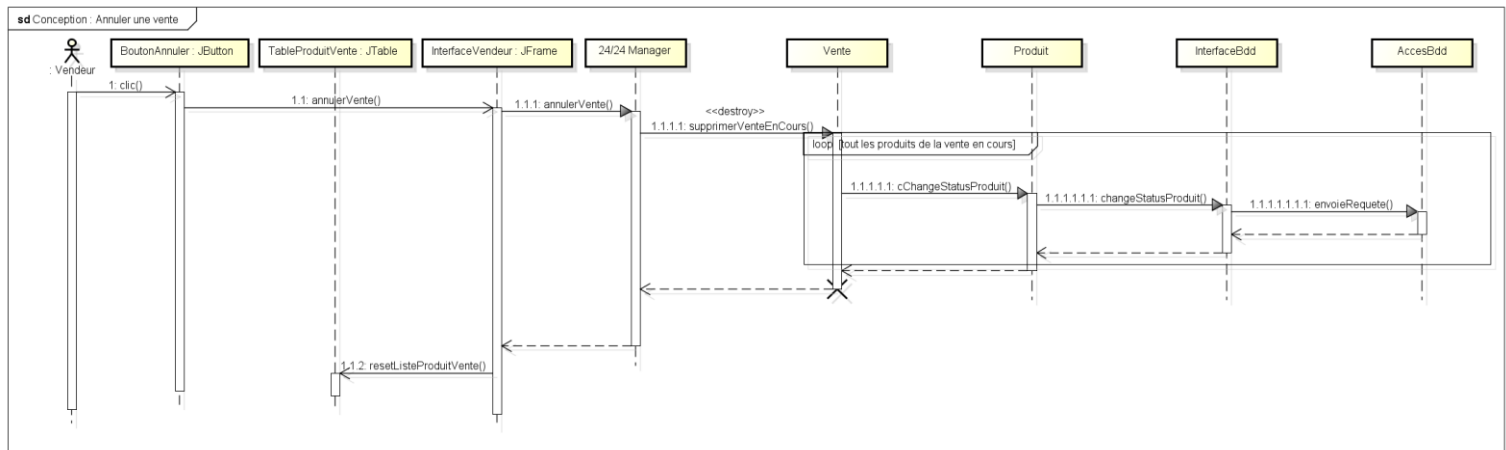
### Retirer un produit

La suppression d'un produit d'une vente en cours fonctionne de la même façon que l'ajout d'un produit. Lors du clic de l'utilisateur sur le bouton correspondant, le produit est ajouté à la vente en cours et son statut est mis à jour en base de données. La liste contenant les produits de la vente en cours peut alors être mise à jour en supprimant le produit concerné.



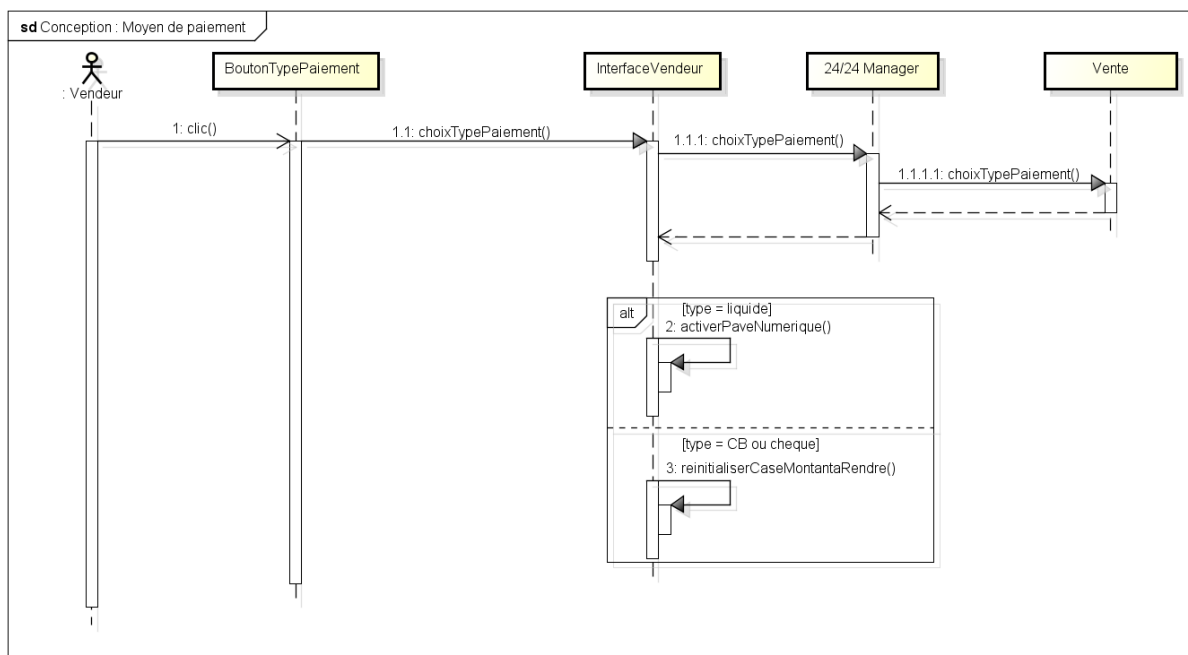
### Annuler la vente en cours

L'annulation d'une vente suit le même schéma que la suppression d'un produit de la vente en cours. Cependant celle-ci impacte tous les produits de la vente et la vente en elle-même. Lorsque l'utilisateur clique sur le bouton Annuler, le contrôleur va modifier le statut de tous produits compris dans la vente en cours puis supprimer la vente en elle-même. Enfin, l'interface va réinitialiser la liste des produits de la vente en cours en supprimant toutes ses entrées.



### Choisir un moyen de paiement

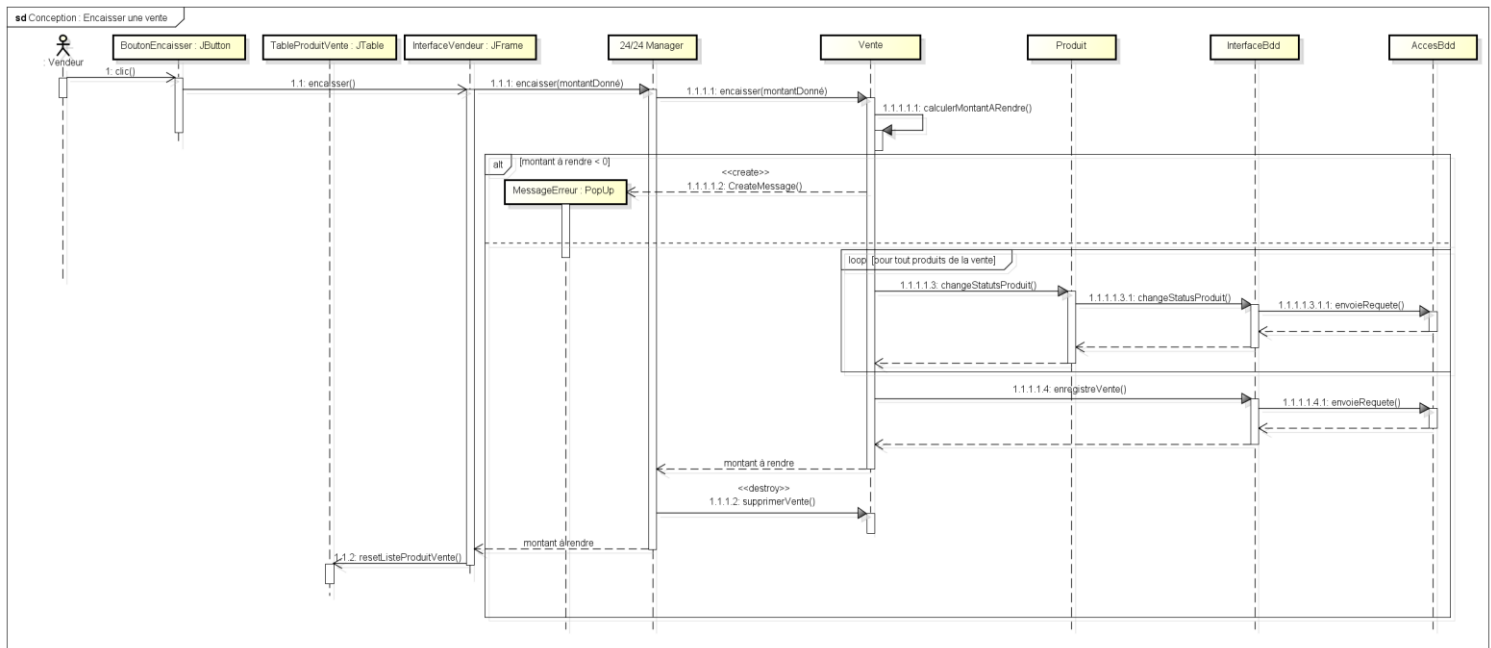
Le vendeur sélectionne le moyen de paiement choisi par le client à l'aide de trois boutons situés sur l'interface. Le clic effectué sur l'un de ces boutons aura pour effet de configurer la vente en cours en lui attribuant un mode de paiement, puis si le mode choisi est le liquide alors le logiciel activera le pavé numérique pour rentrer le montant donné par le client sinon le logiciel indiquera que le montant donné correspond au prix total de la vente en cours.



### Encaisser la vente en cours

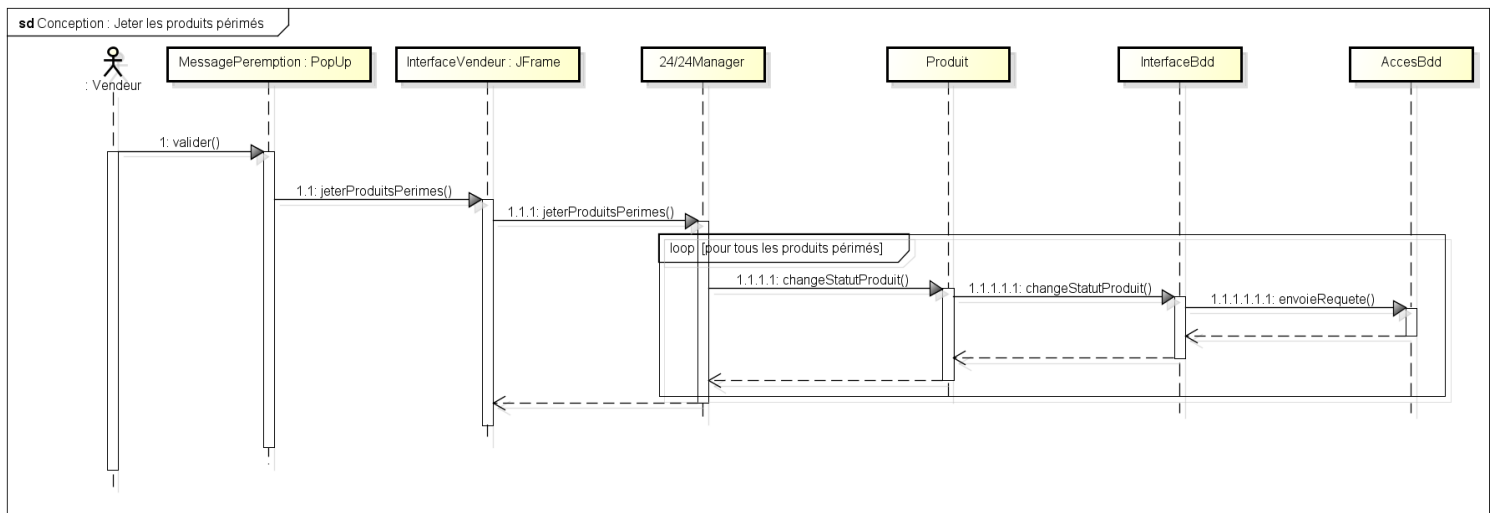
Le clic du bouton d'encaissement entraine le paiement de la vente. L'interface se charge d'envoyer au contrôleur le montant donné qui le transmet à l'objet représentant la vente en cours. Le montant à rendre est calculé. Si ce montant est négatif, il s'agit d'un cas d'erreur qui affiche un message au vendeur et annule l'encaissement de la vente en cours sans la supprimer. Sinon la vente est validée, le programme modifie le statut de tous les produits et enregistre la vente effectuée en base de données. L'objet représentant la vente

en cours est alors supprimé. Au niveau de l'interface, la liste des produits de la vente en cours est réinitialisée et le montant à rendre est affiché.



### Jeter les produits périmés

Lorsque certains produits ont dépassé leur date de péremption, le logiciel averti le vendeur en affichant un message dans un pop-up. Lorsque le vendeur confirme en validant qu'il a bien reçu le message, le contrôleur modifie les informations concernant les produits périmés en mettant à jour leur statut dans la base de données.

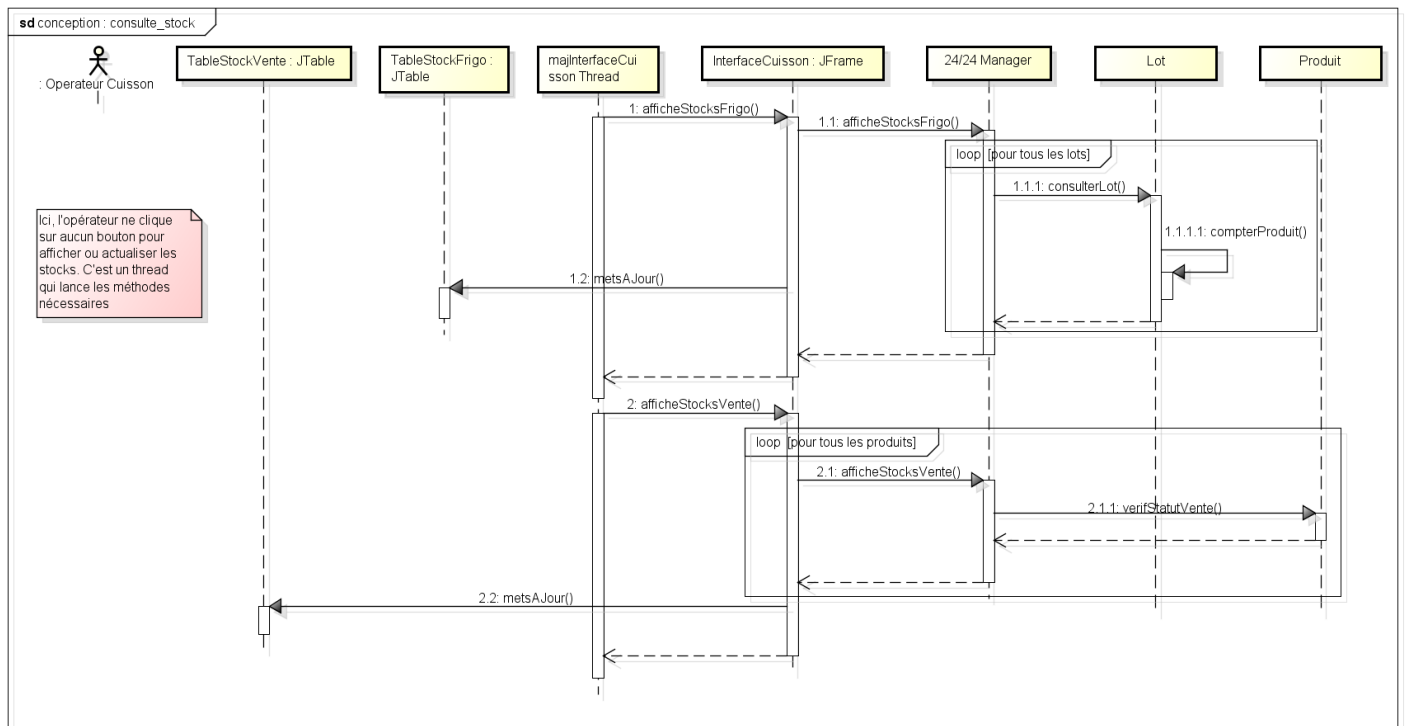


### 3.1.3. Interface employé de cuisson

#### Afficher les stocks

Lorsqu'un utilisateur se connecte en tant qu'employé de cuisson, il a accès, grâce à l'interface de cuisson, à une visualisation en temps réel des stocks des produits en vente et en réfrigérateur. L'interface de cuisson se chargera de mettre à jour les tables contenant les différents produits en affichant leurs noms et leurs quantités grâce à un thread. L'utilisateur n'a donc pas besoin de cliquer sur un bouton pour actualiser l'interface.

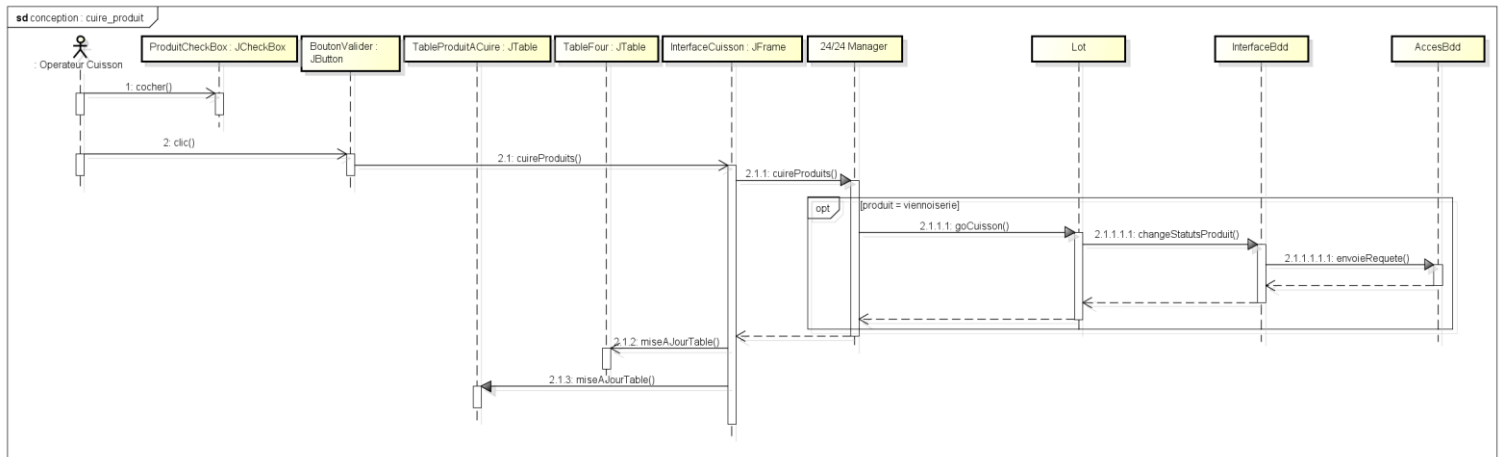
L'employé de cuisson ne pourra pas réaliser d'action sur ces tables, il pourra seulement les consulter.



#### Cuisson de produits

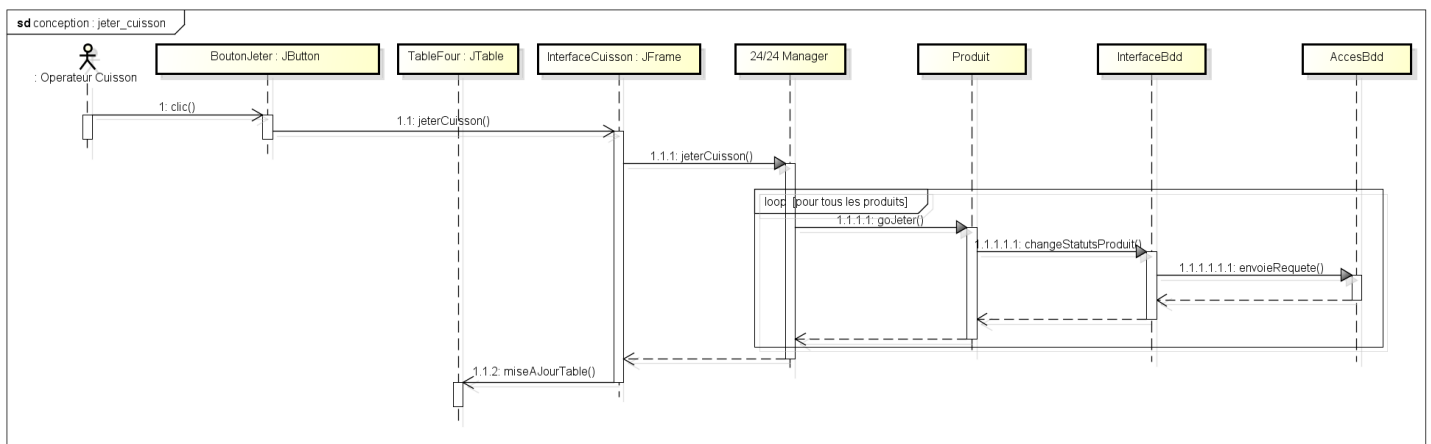
Cette action ne concerne pas les boissons.

Le choix des produits à cuire se fait par sélection des produits correspondants dans un tableau « à cuire ». Il faut ensuite cliquer sur le bouton valider pour confirmer la sélection. Les produits sélectionnés passent ensuite dans le tableau « au four ». Ces deux tableaux sont mis à jour après chaque validation des produits à cuire. Au niveau de la base de données, le statut de chaque produit est également mis à jour.



### Rejet de cuisson

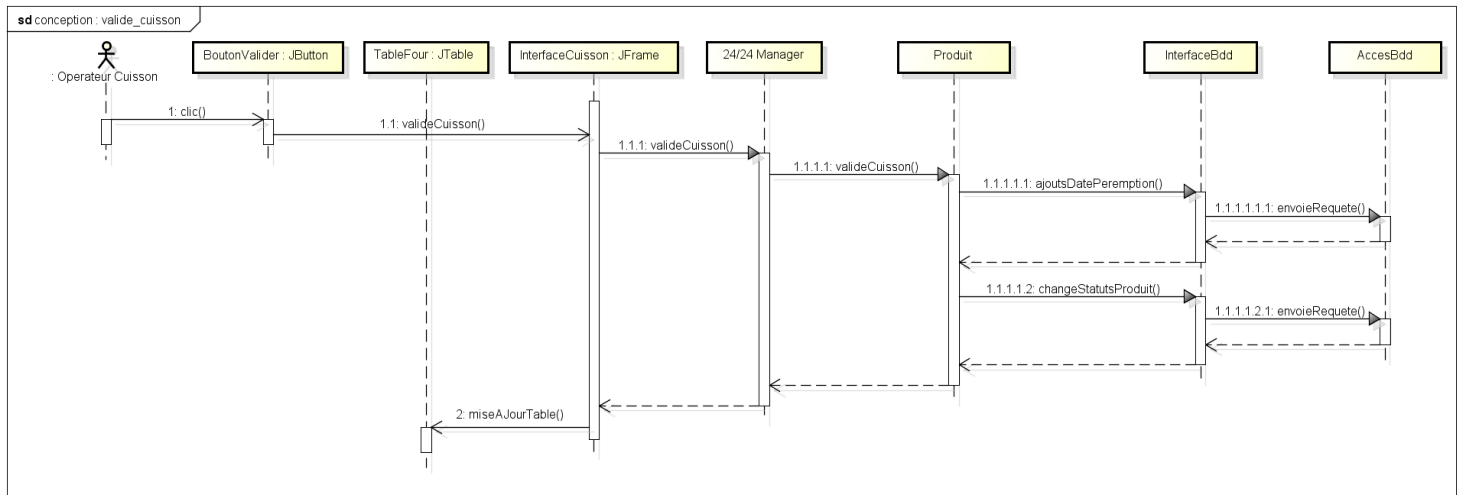
Dans le cas où les produits sont trop cuits (Délai de cuisson est dépassé), ceux-ci peuvent être notifiés comme « jetés » : l'employé de cuisson doit cliquer sur le bouton « à jeter » dans l'interface de cuisson pour effectuer cette action. Le tableau « au four » est mis à jour, les produits jetés sont supprimés de la liste. Au niveau de la base de données, le statut de chaque produit jeté est également mis à jour en modifiant son statut.



### Validation de cuisson

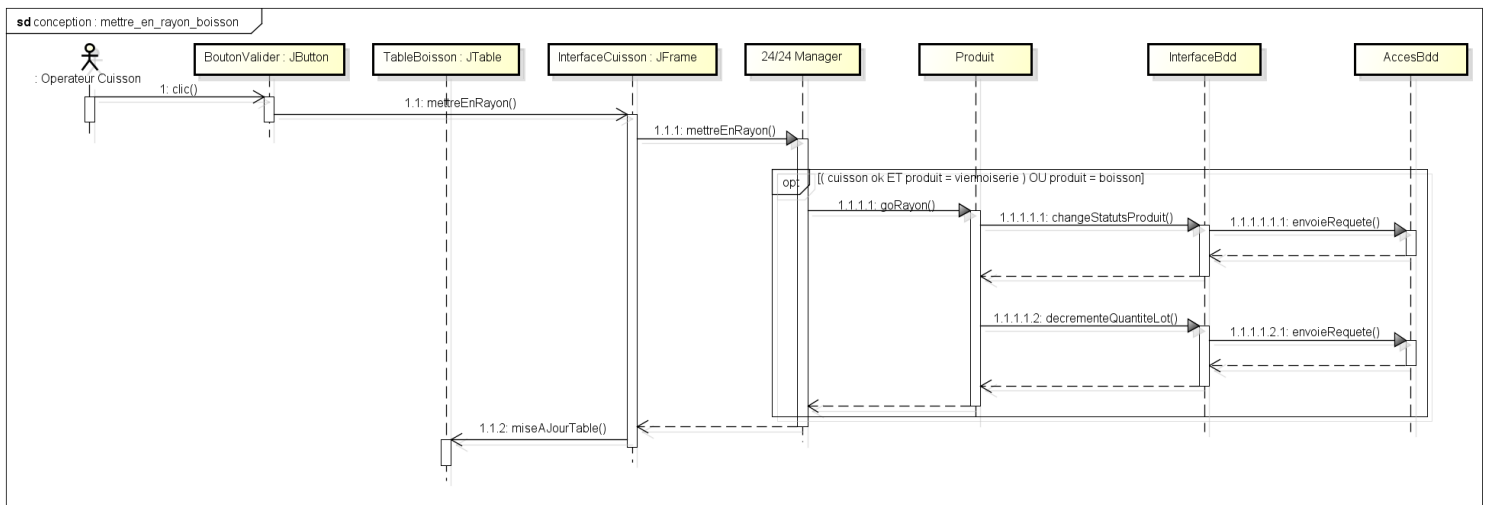
A la fin de la cuisson des produits, l'employé de cuisson peut les mettre en rayon. Pour les pains et les viennoiseries : l'employé de cuisson valide le choix dans le tableau « au four » et les produits concernés disparaissent du tableau « au four ». Au niveau de la base de données, le statut de chaque produit qui passe dans le stock en vente est mis à jour ainsi que sa date de péremption.





### Mise en rayon

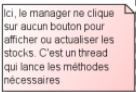
Pour mettre en rayon les boissons : l'employé de cuisson valide le choix dans le tableau « Boisson à mettre en rayon », la boisson concernée apparaît alors dans tableau « stock vente ». Au niveau de la base de données, le statut de chaque produit est également mis à jour et la quantité de produits dans le lot concerné est décrémentée. Enfin, la table « Mettre en rayon » est mise à jour en enlevant les produits qui sont passés dans le stock en vente.



### 3.1.4. Interface manager

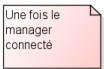
#### Afficher les stocks du magasin (onglet Gestion des stocks)

Lorsque le manager se connecte au logiciel, il arrive sur l'onglet **Gestion des stocks**. On a donc un affichage des stocks. Cet affichage ne va pas chercher les informations dans la base de données puisque les informations sont pré-chargées avant dans des objets Java. Ce sont ces objets que nous consultons directement.



Consulter les statistiques du magasin (onglet Statistique)

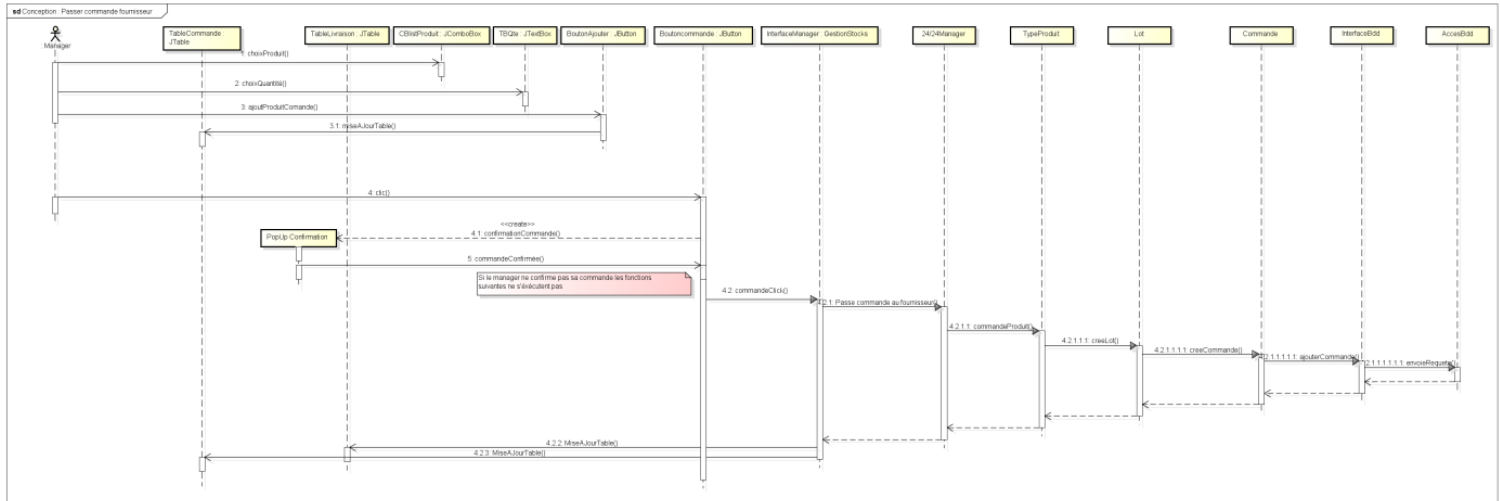
les statistiques par vendeur et les statistiques générales de vente sur une période donnée. Pour les statistiques vendeurs, on compte combien de vente ont été effectuées sur une période donnée (jour, mois ou année), on effectue les calculs nécessaires et on affiche les résultats. Concernant les ventes totales, on récupère le nombre de vente /le nombre de produits jetés / le nombre de produits perdus, dans la base de données pour une période donnée (jour, semaine, mois, année).



### Passer une commande au fournisseur (onglet Gestion des stocks)

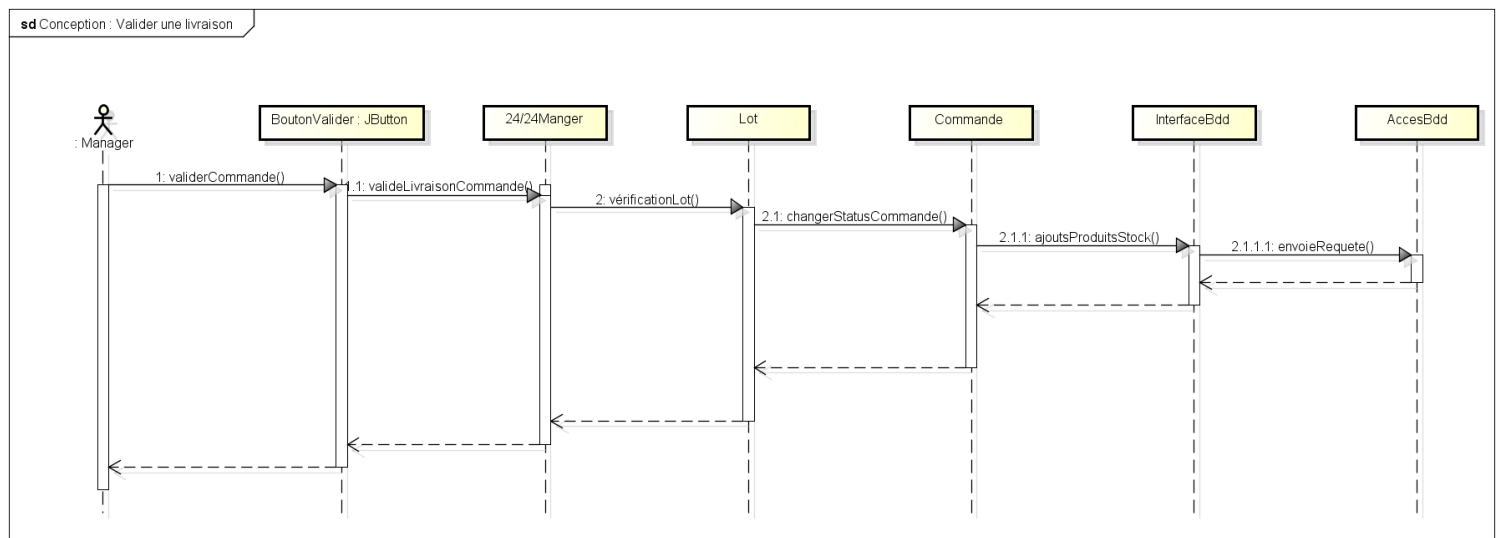
Le manager, pour gérer ses stocks, peut passer des commandes de produit. Il commence par choisir un produit dans une liste déroulante, une quantité correspondante et clique sur « Ajouter ». Cette action provoque une mise à jour de la table de la commande en cours. Une fois sa commande complète, le manager clique sur « Commander » ou « Annuler ». Lorsqu'il clique sur « Commander » et qu'il a confirmé via le pop-up de

confirmation, les objets Lots sont créés et mis en base de données. Une commande est créée et prend le statut « En cours de livraison » puis est ajoutée à la base de données. Si le manager décide d'annuler sa commande, aucun enregistrement n'est fait, ni en objet Java ni en base de données : le seul changement est au niveau de l'affichage du tableau correspondante.



### Validation d'une livraison (onglet Gestion des stocks)

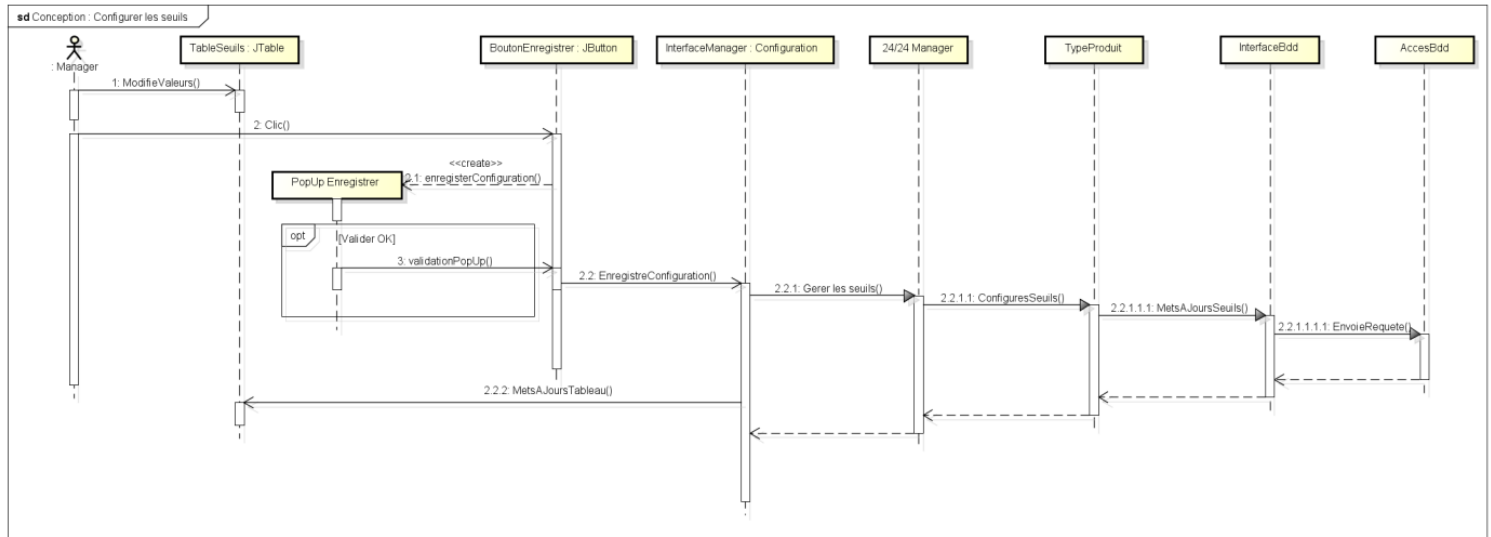
La manager valide ensuite à chaque livraison via le bouton correspondant. Lors de cette validation, nous effectuerons une vérification des lots correspondants en base de données (leur existence notamment), puis le statut de la commande correspondante à la livraison passera à « en stock ». Les produits seront donc en stock réfrigérateur si la validation s'est bien déroulée. Dans le cas d'un problème de livraison, le manager peut annuler la commande en cours de livraison ce qui aura pour effet de supprimer la commande de la base de données.



### Configuration des différents seuils et éléments du logiciel (Onglet Configuration)

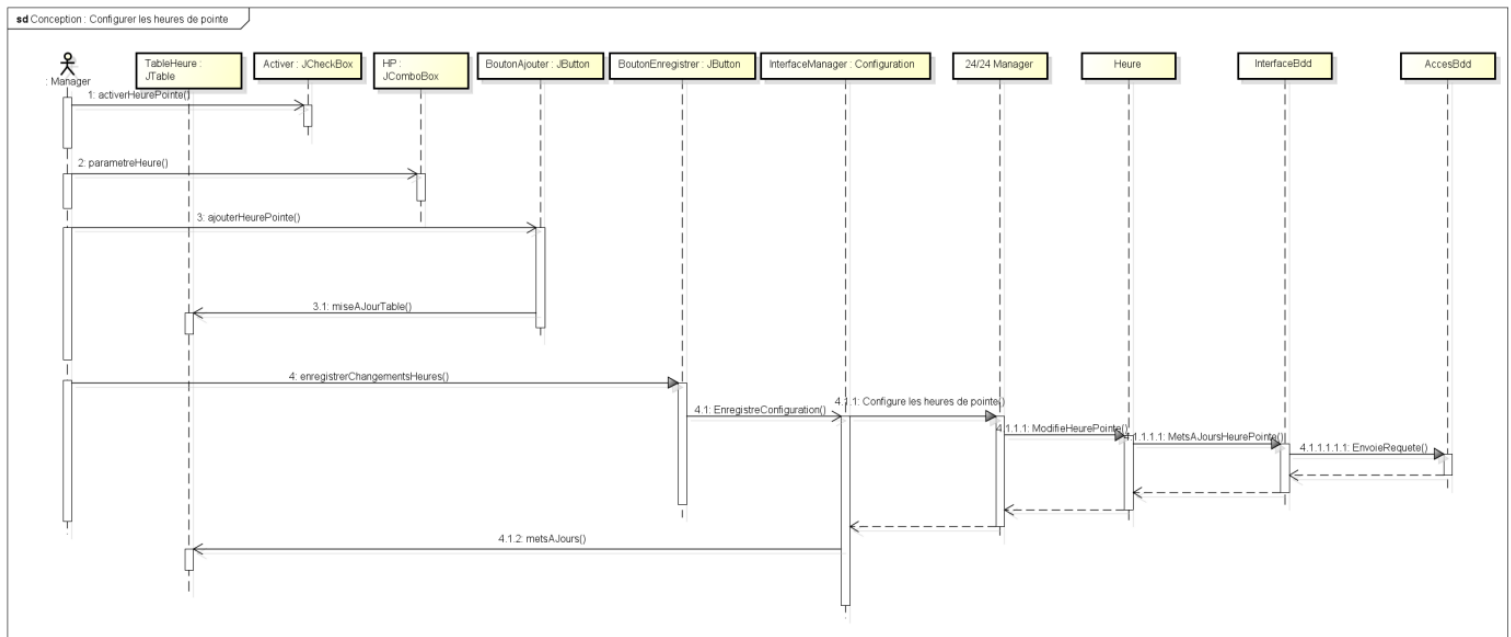
Une fois sur l'interface de configuration, le manager à plusieurs choix : il peut configurer les différents éléments et seuils du logiciel ou ajouter/enlever des heures de pointe.

Lors de la configuration des seuils, le manager ne peut changer d'onglet sans avoir enregistré ou annulé ses modifications via le pop-up de confirmation. Le logiciel récupère toutes les valeurs qui ont changées et les rentre en base de données. Lorsqu'il a terminé, il enregistre son travail via le bouton « Enregistrer ».



### Gestion des heures de pointes (Onglet Configuration)

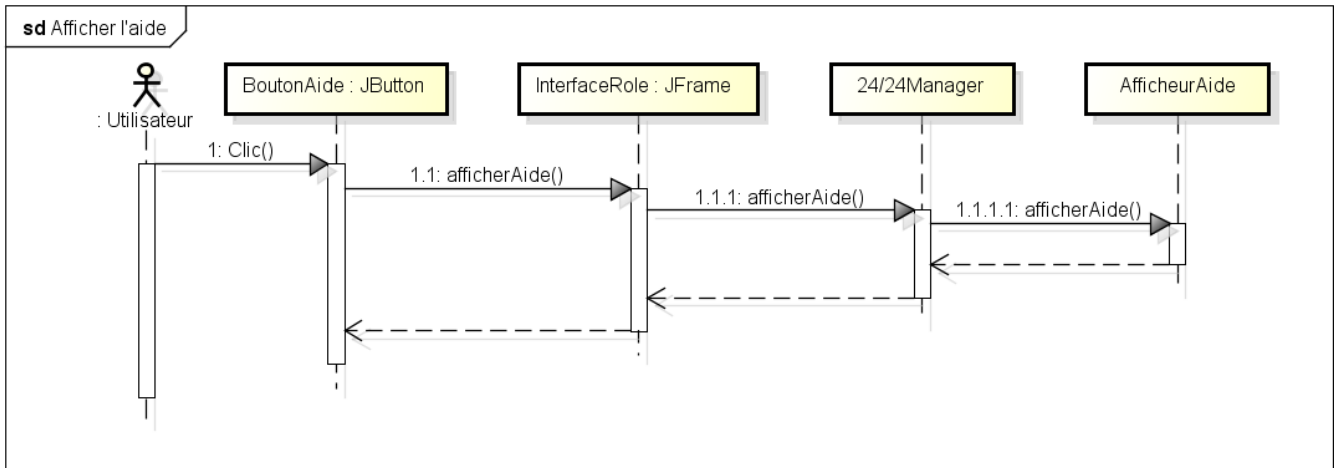
Le manager peut ajouter des heures de pointes, en activer et en désactiver. Pour ajouter une heure de pointe, le manager choisi dans des listes les heures voulues et clique sur « Ajouter ». Cela aura pour effet de mettre à jour la table des heures sur l'interface. Une fois qu'il a terminé il clique sur le bouton « Enregistrer » de l'onglet et les heures sont rentrées en base de données pour traitement. De même, il ne peut quitter cet onglet sans enregistrer ou annuler ses modifications d'heures.



## 3.2. Cas d'utilisation annexe

### 3.2.1. Afficher aide

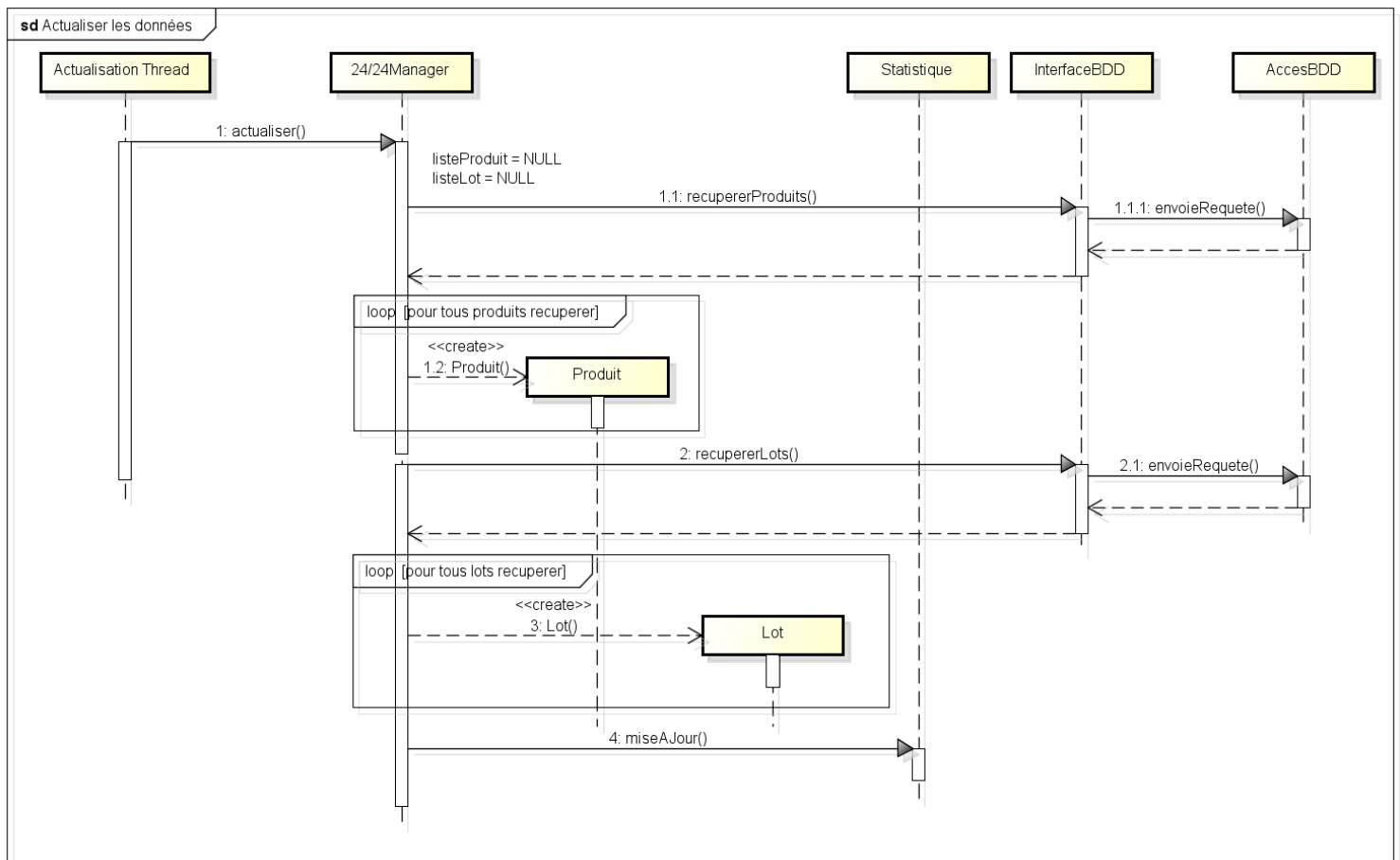
Ce cas d'utilisation permet à un utilisateur, depuis n'importe quelle interface, d'afficher une aide concernant le logiciel. Cette aide sera au format PDF et regroupera sur un même fichier de l'aide par rapport à tout le logiciel. La classe AfficheurAide permettra d'afficher ce fichier.



## 3.3. Hors cas d'utilisation

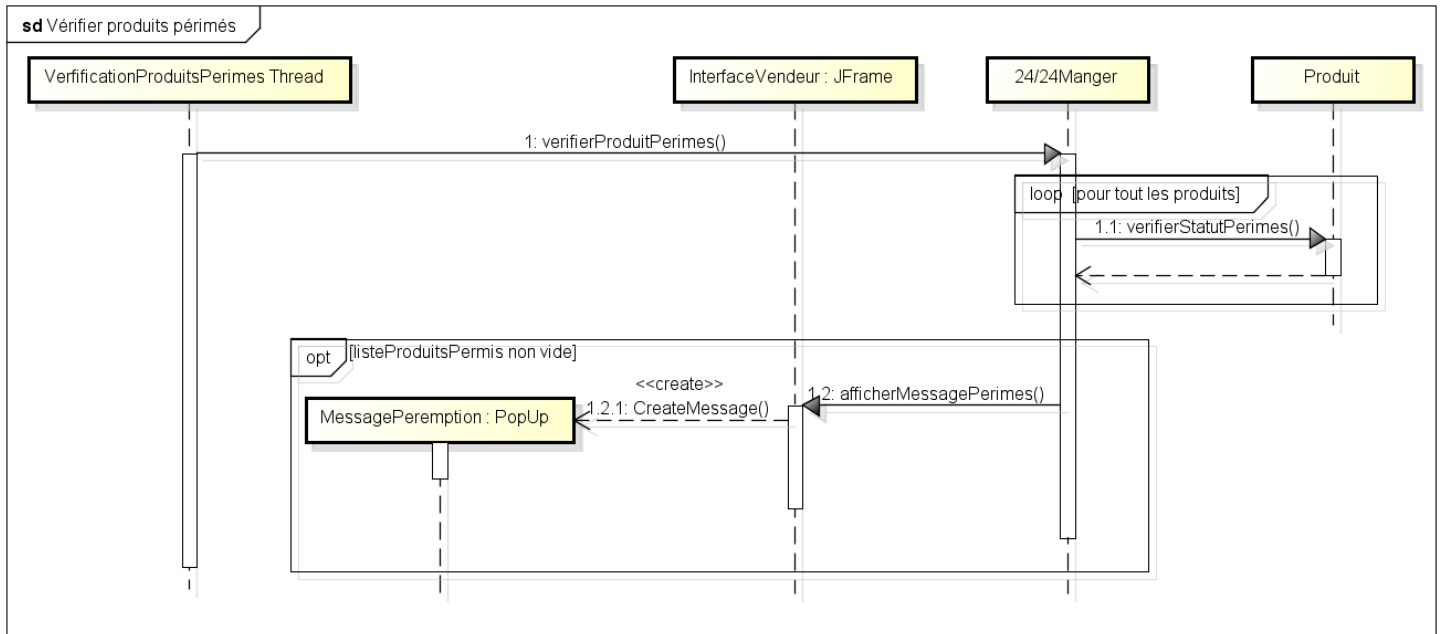
### 3.3.1. Thread de récupération des données contenues dans la base

Certaines informations sont présentes en même temps sur plusieurs interfaces. Pour assurer la cohérence des données entre les différentes interfaces, il faut donc actualiser les données présentes dans nos objets Java. Pour cela, le système exécute un thread qui récupère périodiquement les données dans la base et crée les objets correspondants.



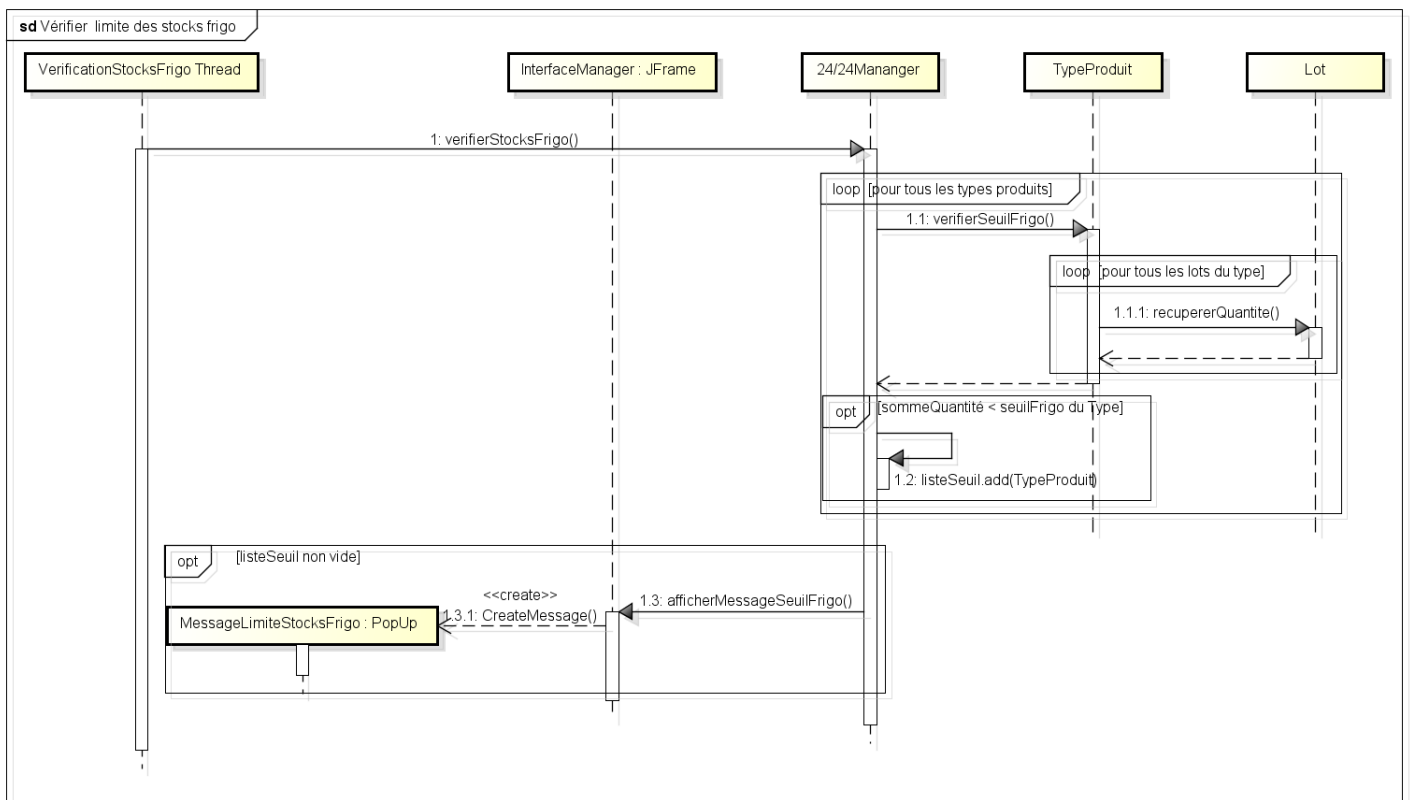
### 3.3.2. Thread pour vérifier les produits périmés

Pour vérifier si des produits sont périmés, le logiciel utilisera un thread. Son but sera d'analyser régulièrement tous les produits et de stocker ceux qui sont périmés. La méthode « verifierStatutPerimes » de la classe Produit compare la date de péremption avec la date courante. Si le produit est périmé, il est stocké dans une liste (non représenté sur le diagramme) et si cette liste n'est pas vide alors le contrôleur commandera à l'interface vendeur d'afficher un pop-up avec le nombre et le type de produit périmés. Valider le pop-up entraîne le cas d'utilisation « Jeter les produits périmés » présent dans les parties concernant l'interface vendeur.



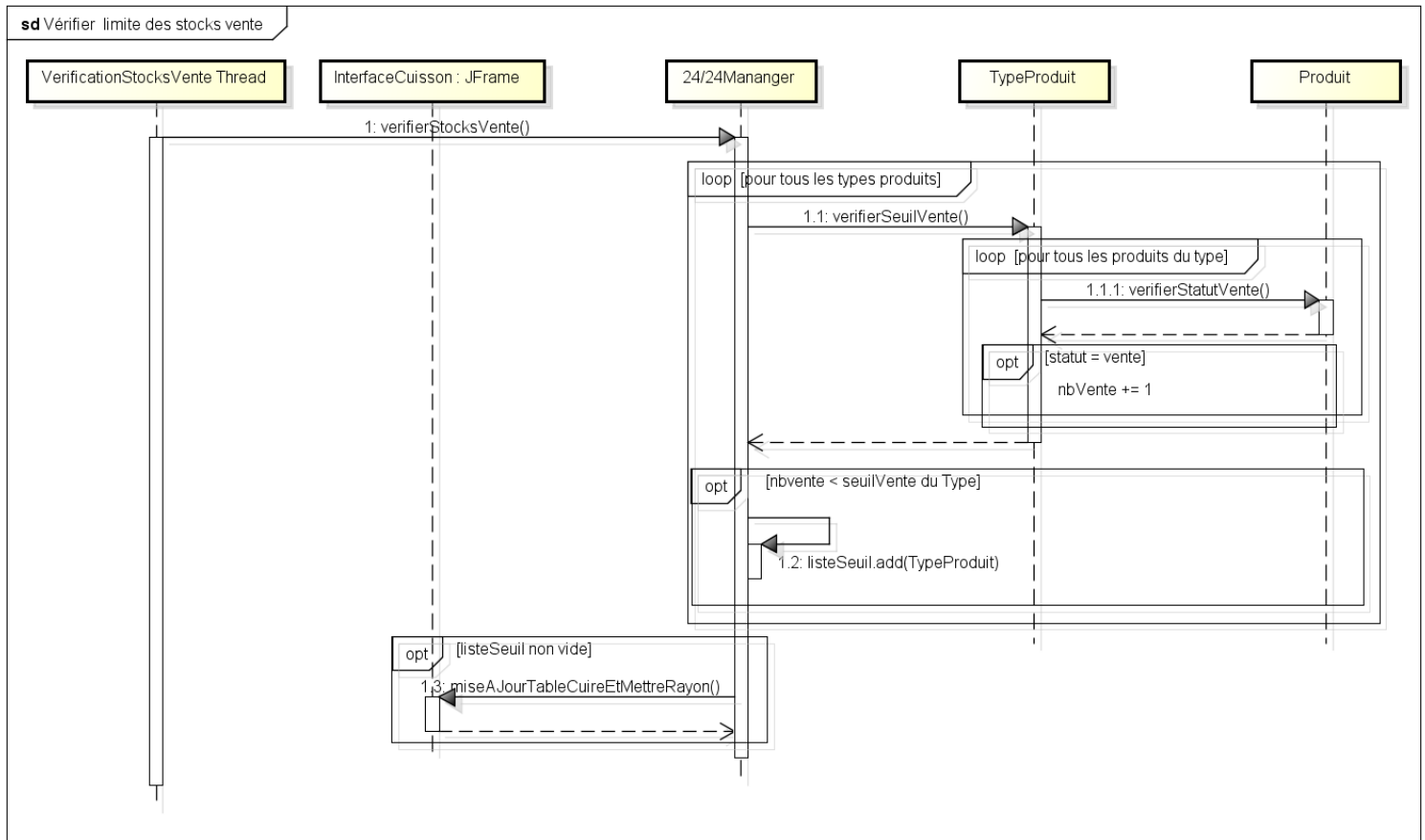
### 3.3.3. Thread pour vérifier la quantité des stocks congelés

Le thread de vérification des stocks en réfrigérateur va vérifier régulièrement l'état des stocks. Pour cela, il appelle une méthode du contrôleur qui, pour chaque type de produit, somme la quantité de produit des lots du même type. Après avoir fait la somme, il compare la valeur avec le seuil défini et ajoute le type de produit à une liste si la somme est inférieure. Le logiciel affiche alors un pop-up pour prévenir le manager qu'il reste peu de stock pour certains types de produit. La validation du pop-up ne lance aucun cas d'utilisation et c'est le manager qui prend la décision de commander des produits.



### 3.3.4. Thread pour vérifier la quantité des stocks en vente

Le thread de vérification des stocks de vente fonctionne de la même manière que celui de vérification des stocks réfrigérateur. Pour tous les types de produits, le logiciel incrémente une variable pour tous les produits qui sont en vente. Si cette variable est inférieure au seuil de produit en vente alors le logiciel ajoutera ce type de produit dans une liste. Si la liste est non vide, les listes représentant les produits à cuire et mettre en rayon sont mises à jour.



### 3.4. Diagrammes d'activités et d'état transition

A ce jour les diagrammes d'activité et d'état transition n'ont pas été décrits, faute de temps. Pour toutes questions concernant ce point, vous pouvez nous contacter via l'adresse mail [groupeg.4irc@gmail.com](mailto:groupeg.4irc@gmail.com).



## 4. Implémentation

Cette partie du travail consiste à exprimer les choix d'implémentation des classes et de leurs attributs. Les attributs ne sont pas visibles sur le diagramme de classe partie Analyse car il reste des modifications à effectuer. Le dossier de conception sera mis à jour lorsque les attributs seront validés par l'ensemble de l'équipe.

### 4.1. Partie statique

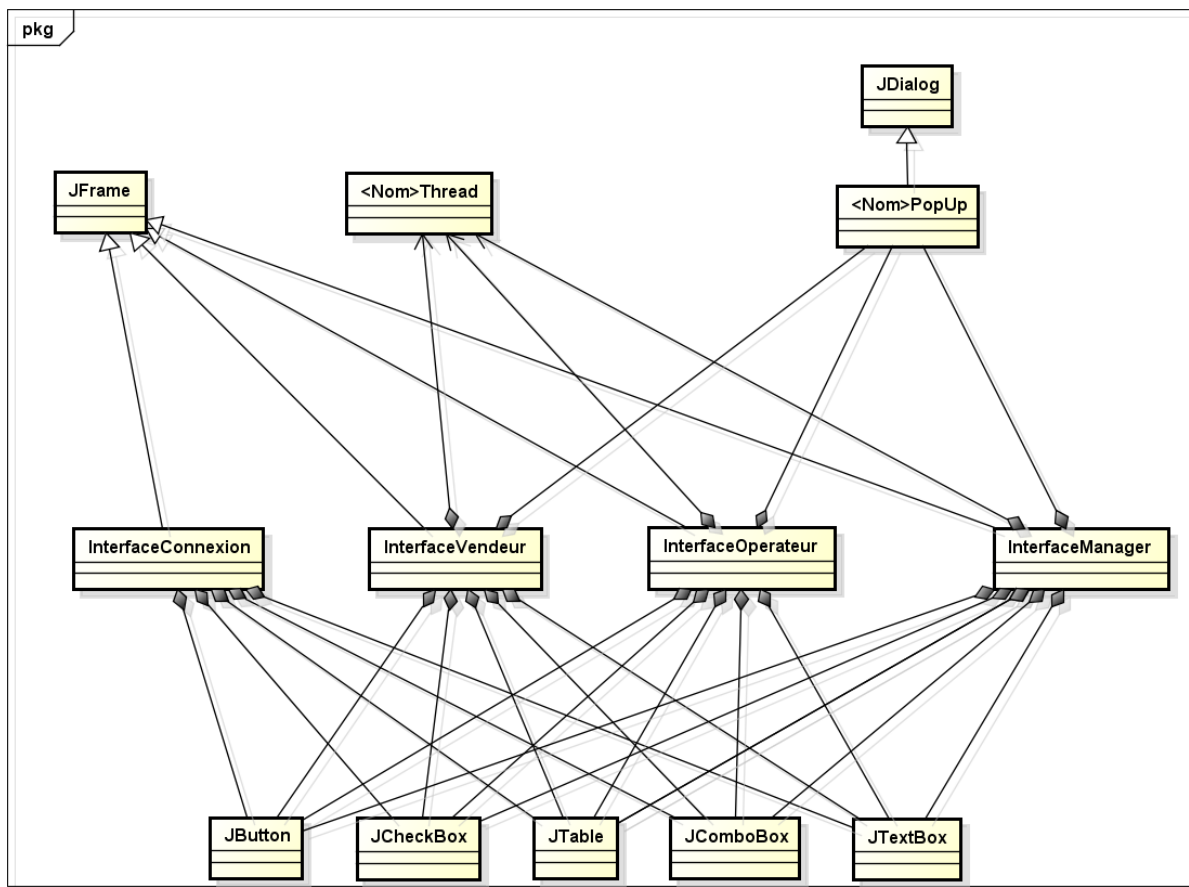
Les liens entre classes sont représentés dans les diagrammes de classe présents dans la phase d'analyse. On observe principalement des liens d'association et d'agrégation.

### 4.2. Partie dynamique

Cette partie n'a pas été réalisée car elle s'appuie sur les diagrammes d'état-transition qui n'ont pas pu être réalisés.

## 5. Annexe A : Diagramme d'interface

### 5.1. Diagramme



### 5.2. Descriptif des classes

**InterfaceConnexion** : Fenêtre du logiciel permettant à l'utilisateur de se connecter au logiciel.

**InterfaceOpérateur** : Il s'agit d'une classe permettant de créer l'IHM de la partie opérateur de cuisson. Elle sert à l'opérateur de cuisson à interagir avec le logiciel.

**InterfaceVendeur** : Classe qui créer l'IHM permettant à un vendeur de communiquer avec le logiciel.

**InterfaceManager** : Cette classe créer l'IHM du manager et permet à celui-ci de discuter avec le logiciel.

**<Nom>Thread** : Cette classe représente toutes les classes dont le nom se termine par Thread dans les diagrammes de séquence. Ces thread sont regroupés dans une même classe pour ne pas surcharger le diagramme. Elles permettront de mettre à jour régulièrement certaines parties des interfaces. Par exemple, elle sera utilisée sur l'interface du vendeur pour afficher les produits disponibles à la vente en temps réel.

**<Nom>Pop-up** : Même principe que pour les classes de type Thread. <Nom>Pop-up regroupe toutes les classes de type pop-up qui peuvent être appelé par les interfaces.

**JButton, JCheckBox, JTable, JComboBox, JTextBox, JFrame, JDialog :**

Ce sont des classes Swing déjà existantes, elles ne seront donc pas développées.