|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Epreuve intégrée | | | | | | |
| warehousemgmt.io | | | | | | |
| **Version :** | | WareHouseMGMT.IO\_v1.0 | | |  |  |
| **Date de Révision:** | | 19/01/2016 | | |  |  |
|  | | | | | | |
|  |  | |  |  | | |
| **Auteur Ahmed Idoumhaidi**  **Auteur Mikel Cromphoudt**  **Auteur Laurent Demey**  **Auteur Damien Bouillez**  **Auteur Baptiste**  **Auteur Lionel Wéry**  **Auteur Giuseppe**  **Date**  **Signature** | | | | | | |

# Informations

## Révision

|  | | | |
| --- | --- | --- | --- |
| **Version**  **Date** | **Nom de l’auteur** | **Sections** | **Modifications** |
| 1 | Mikel Cromphoudt /  Ahmed Idoumhaidi | Introduction | Insertion de la description du cadre de base du projet. Généralités, objectifs, etc. |
| 2 | Mikel / Ahmed /Damien / Baptiste / Laurent | Les flux niveau macro / quelques règles | Calculs de la surface utile, nombre palettes, rangements, systèmes de gestion (FIFO), évocations des différents scénarii possibles. |
| 3 | Mikel / Ahmed / Lionel / Laurent / Baptiste / Damien | Diagrammes de flux / Description sommaire des fonctionalités | Détail de la zone ‘Relay’, les différents flux |
| 4 | Mikel / Ahmed / Lionel / Laurent / Baptiste / Damien | flux / fonctionnalités | Gestion des espaces de stockage, détail des fonctionnalités, révision des flux, ébauche de la DB, questions à débattre en groupe |
|  |  |  |  |

## Table des matières

1. Informations 2
   1. Révision 2
   2. Table des matières 3
2. Introduction 4
   1. Objectif 4
   2. Contexte et existant 4
   3. Destinataires 4
   4. Les règles implémentées 4
   5. Glossaire 5
3. Formalisation du besoin 6
   1. Description des flux : 6
   2. Description des fonctionnalités : 11
   3. Informations utiles : 13
   4. Story board 14
   5. Données 14
4. Modélisation de la base de données 15
5. Spécifications techniques 16
   1. Spécifications logicielles 16
   2. Interfaces / flux 16
   3. Spécifications des données 16
6. Informations utiles 17
   1. Nombre d’utilisateurs 17
   2. Nombre d’administrateurs 17
   3. Volumétrie 17
   4. Sauvegarde 17
   5. Indisponibilité 17
   6. Sécurité 18
7. Phase de tests 19
   1. Tests à réaliser 19
   2. Résultats attendus 19
   3. Résultats obtenus 19

# Introduction

## Objectif

Ce travail de fin d’étude a pour objectif de procéder à l’analyse des besoins, la conception et la réalisation d’une application permettant la gestion complète du stock d’une centrale d’achat. Par gestion, on entend ici la commande à nos fournisseurs pour alimenter le stock de l’entrepôt, mais aussi, par définition, le stockage de produits ainsi que l’expédition de ces derniers.

## Contexte et existant

(Décrire le contexte général de réalisation du projet : stratégie de refonte informatique, nouveau développement, etc.…)

Au niveau du contexte, on va considérer une situation de départ où l’informatique dans l’entreprise est inexistant. Nous mettons donc en place un système neuf à l’aide des technologies décrites plus bas.

* Les produits qui sont gérés dans le cadre de ce projet sont des produits non périssables.
* Le stock de palettes est pleine propriété de la centrale d’achat. Aucune autre palette ne rentre ou ne sort de l’entrepôt.

Dans le cas de figure qui nous occupe, il n’y aucun existant. Il s’agit d’un projet de mise sur pied d’un nouveau système informatique pour une entreprise de type centrale d’achat/gestion de stock pour des magasins de détail. C’est l’application qui gère les mouvements de palettes entrantes et sortantes correspondant aux approvisionnements depuis nos fournisseurs aux livraisons de nos points de détail. L’application prend en charge non seulement la localisation des produits et palettes mais aussi le réapprovisionnement lorsque le niveau de stock l’exige (en se basant sur la rotation attendue du produit concerné) ou sur base d’une commande dont le volume sort de l’ordinaire.

## Destinataires

(Indiquez ici les directions / services / individus auxquels s’adresse ce document. L’objectif ici est d’impliquer les acteurs.)

L’application est destinée au management du dépôt de la centrale d’achat ainsi qu’aux ouvriers chargés de la préparation des commandes et du réassort au niveau du stock. Elle doit permettre de distribuer l’information utile pour que les ouvriers en chargent de la gestion physique du stock puissent exécuter leurs tâches sur base des instructions générées par l’application. En ce sens, l’application vise donc également les utilisateurs finaux réels même s’ils ne sont pas les commanditaires de cette application ni les clients finaux.

## Les règles implémentées

* Pas plus de 100 articles pour notre gestion de dépôt
* Il n’y a pas d’espace prédéfini par produit, voir système d’attribution dichotomique
* Chaque ligne d’un document facture/commande doit être validée, tâche interactive
* Lors de la mise en stock, la validation du transfert réel d’un article est aussi une tâche interactive
* Un seul article par logette vrac.
* Il faut affecter un état tant à la commande qu’à chacune de ses lignes. Quand une ligne de la commande est vérifiée alors, elle est placée en out et suis le traitement normal pour expédition. Quand une commande est entièrement expédiée (toutes ses lignes ont été traitées) c’est alors que l’état de la commande est modifié à ‘expédiée’ et qu’elle sort du stock de traitement de commande.
* La communication entre module ce fait via des synchronisations automatiques.

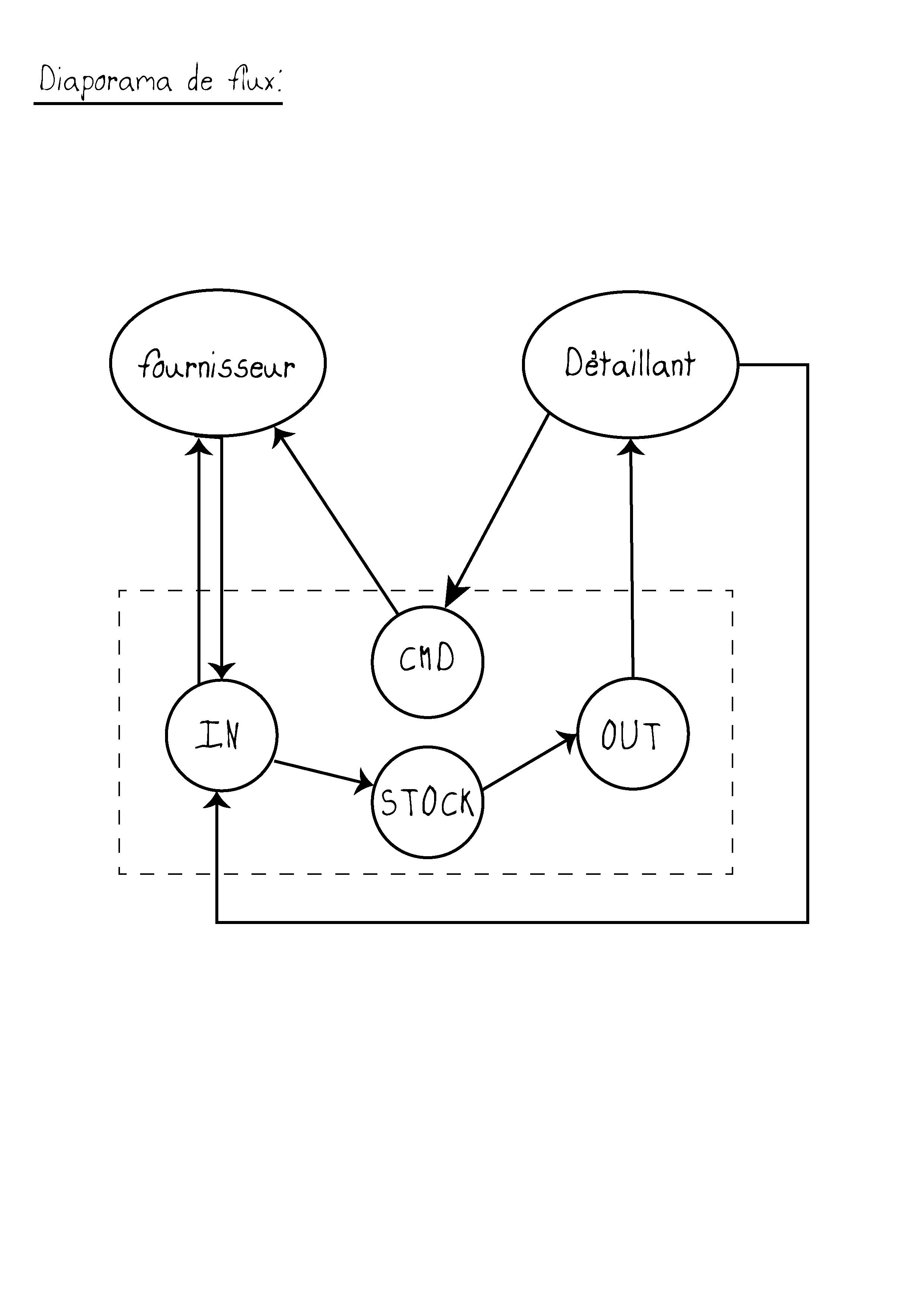
## Glossaire

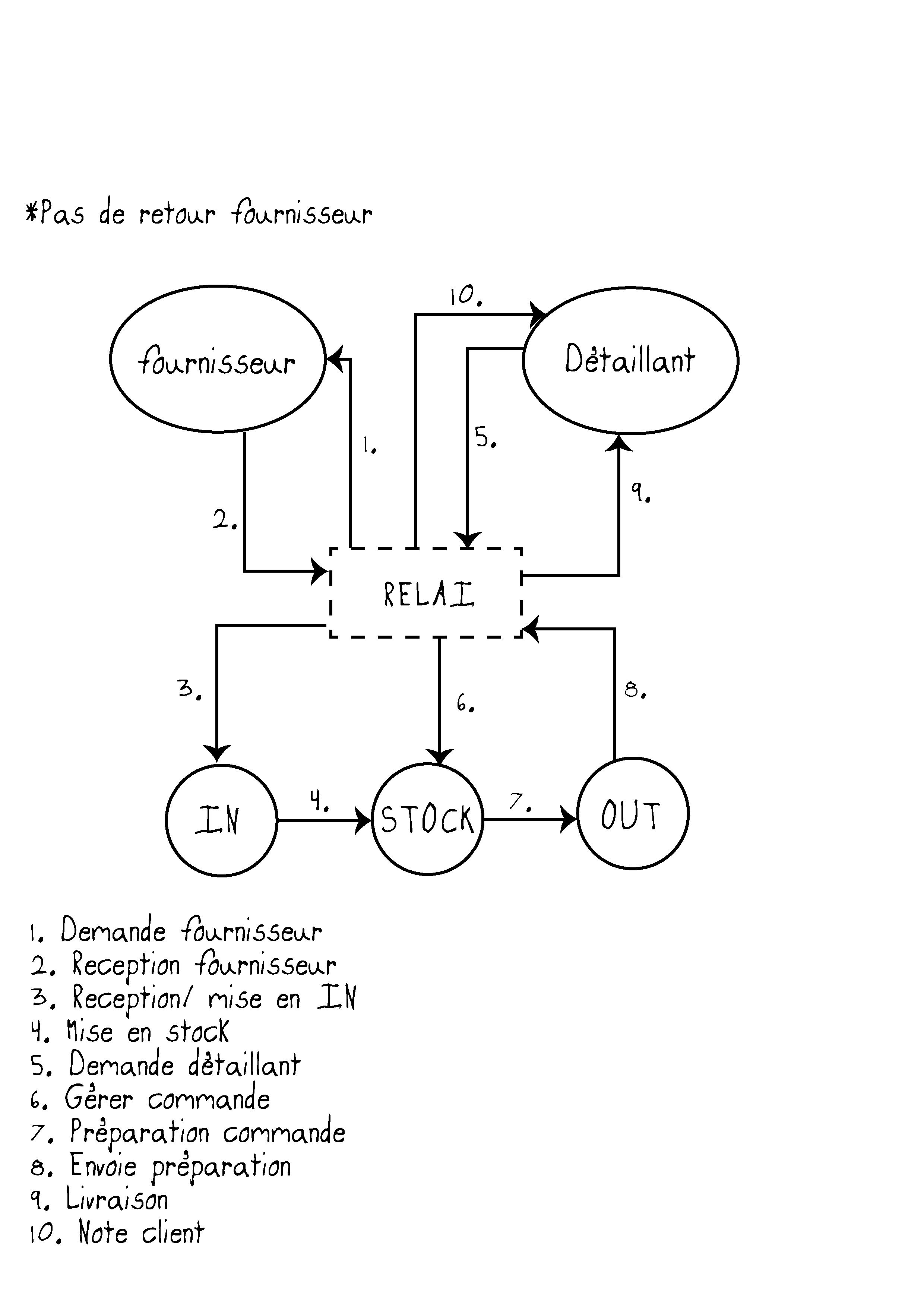
(Donnez ici dans un tableau décrivant le vocabulaire particulier utilisé)

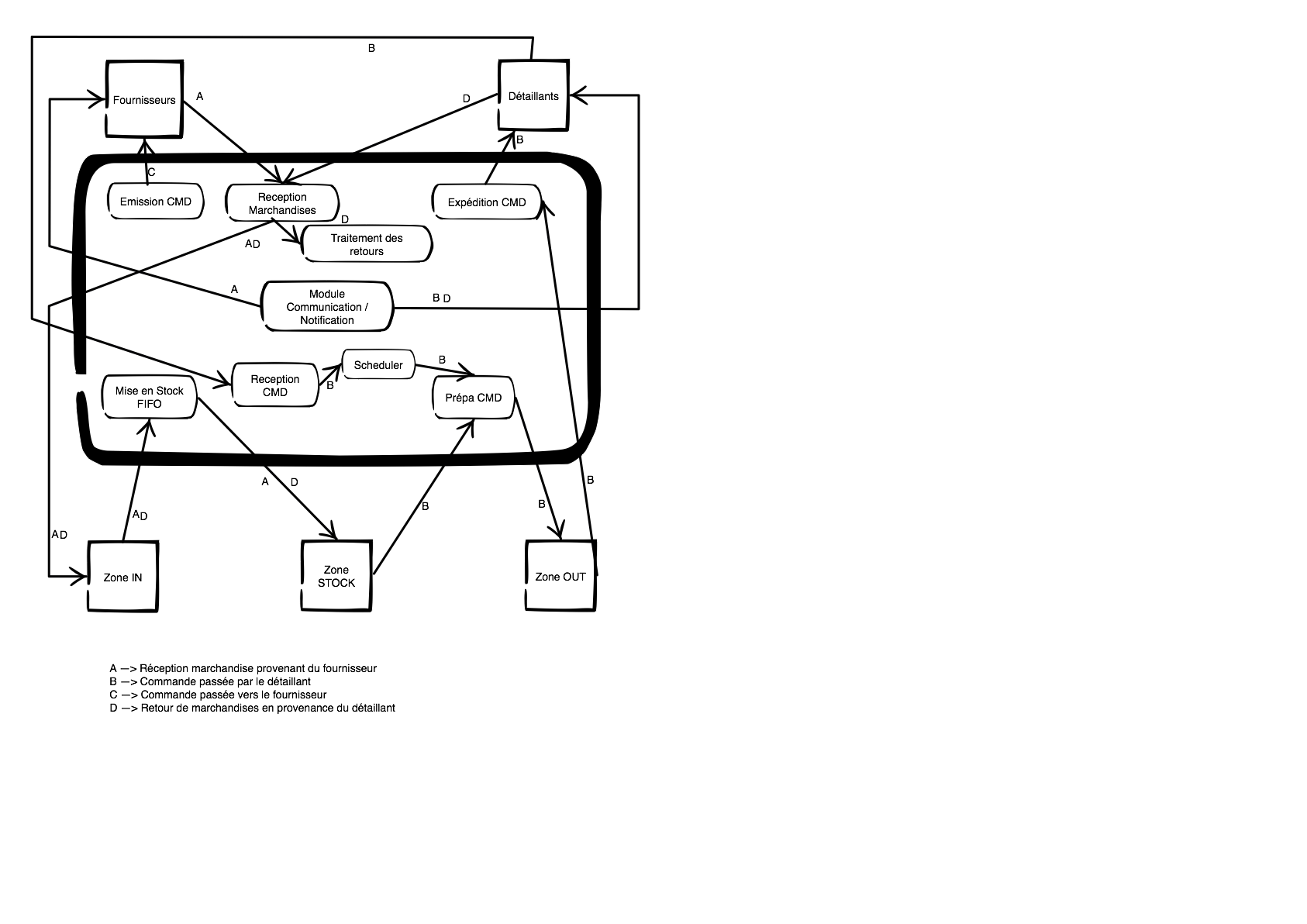
| **Abréviations** | **Description** |
| --- | --- |
|  |  |
| UVC | Unité de Vente Conditionnée |
| Palette Hétérogène | Palette constituée de différents types de produits |
| affectation dichotomique des espaces | Algorithme mis en œuvre pour optimiser la répartition et l’affectation des emplacements de produits dans l’entrepôt. |

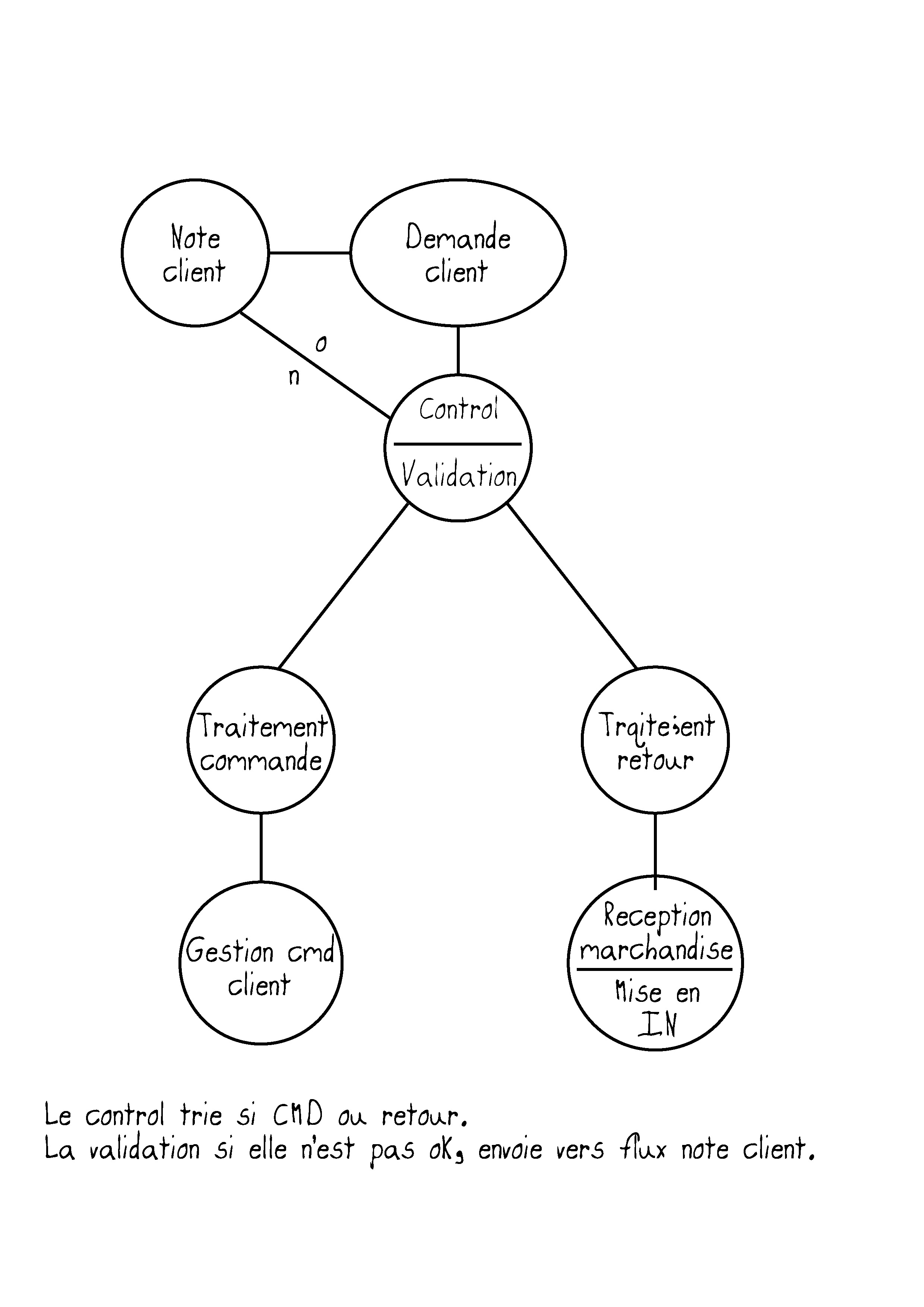
# Formalisation du besoin

## Description des flux :

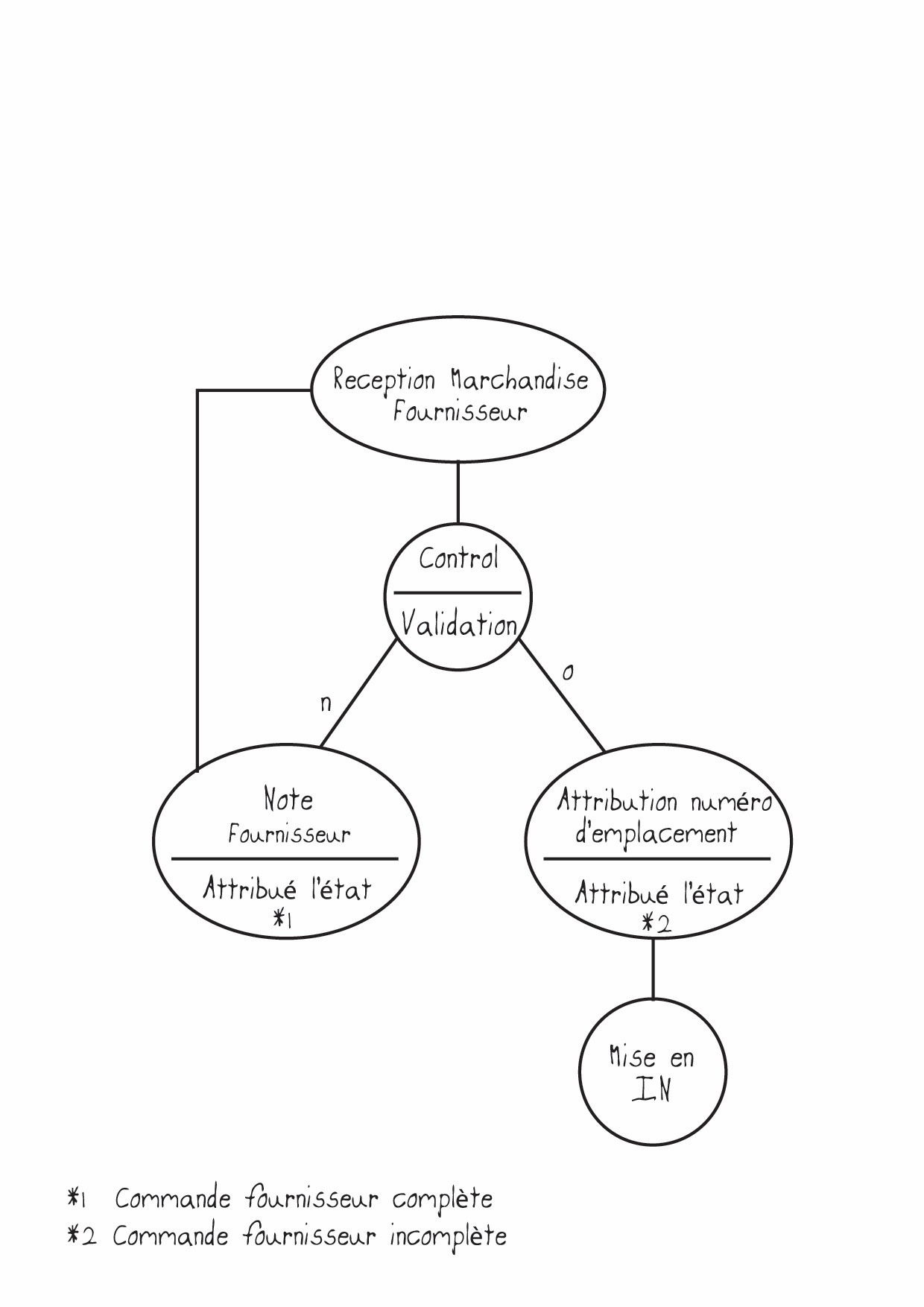
Les flux d’ensemble entre l’entrepôt et les deux entités externes que sont les détaillants (nos clients) et les fournisseurs. La zone délimitée par les pointillés étant l’entrepôt.

Détail des 10 flux entrants et sortants de notre système

Flux de fonctionnement du ‘Relay’

Flux de demande du client/détaillant

flux de réception marchandise fournisseur



## Description des fonctionnalités :

1. Emission commande fournisseur:
   * création de la commande —> via limite stock atteinte ou commande exceptionnelle

Il faut établir un document de type commande fournisseur qui va reprendre les éléments suivants:

* numéro de commande (ID)
* nom du fournisseur
* id fournisseur
* adresse fournisseurs
* libellé article
* quantité (fonction de la limite globale du stock: voir stock dichotomique)
* date de commande
* état de la commande (créé/transmis/complète ou incomplète/traité/archivé)
  + transmission commande —> via module de communication/notification

1. Réception marchandises fournisseur et détaillant:

- administration -> test: réception marchandises fournisseur ou retour détaillant: si retour, voir traitement des retours.

On passe alors par une série de validations afin de déterminer si la réception de marchandises est ok:

* on check les données signalétiques du fournisseurs ou du détaillant de la facture
* on check les articles: présence, quantité, article non-attendu

Si la commande passe les tests alors on passe à la mise en zone IN. Si pas, le statut est modifié à ‘incomplet’. Si ‘incomplet’, on envoie une notification (voir mod de notification), les articles présents suivent le déroulement normal de la mise en zone IN puis mise en STOCK, le reste est en attente sur le document flaggé incomplet.

- mise en zone IN —> attribution emplacement et update du stock

1. Traitement des retours détaillant:

- modification état de la commande (voir sup)

- note de crédit: notification au détaillant et département compta

- réalisation de statistique à destination du management (à définir)

1. Stockage (mise en stock):

- gestion FIFO Stock: on affecte un emplacement lors de la mise en stock avec une date de façon à savoir guider le personnel lors de la préparation de la commande pour un détaillant. (nécessaire à la gestion FIFO du stock)

- mise en zone Stock via un processus d’attribution des espaces palettes et logettes via système dichotomique (voir la définition d’un tel système sous la rubrique glossaire)

- gestion des palettes (split de palettes hétérogènes):

Si palette homogène, on recherche un emplacement pour palette, donc le stock de palette disponible n’est pas affecté. Si, par contre, la palette est hétérogène, on incrémente/décrémente le nombre de palettes dispo en tenant compte de leur type et on passe à l’étape suivante

- gestion des volumes (pour la mise dans les espaces vracs):

On affecte une logette vrac aux articles qui étaient sur la palette hétérogène en fonction de l’espace libre des logettes par rapport au volume des articles.

1. Réception commande détaillant:

- On contrôle les données reçues sur le bordereau de commande du détaillant.

* ID détaillant
* Articles existants
* Commande urgente? Si oui, elle doit être placée tout en haut du stack
* On vérifie que tous les articles soient en stock (ou en IN):

Si oui, on modifie l’état de la commande à ‘en cours de traitement’

- Scheduler (urgence/priorité/disponibilité) : le scheduler vérifie la disponibilité des palettes et leur type et a pour principale fonction de gérer le stack de commande.

1. Préparation de la commande détaillant

- gestion FIFO: on dirige le personnel qui prépare la commande vers la logette ou l’emplacement palette qui présente la date la plus ancienne.

- gestion de la volumétrie des items à destination des palettes sortantes

- gestion poids items et palettes à destination du type de transport

- gestion des commandes partielles: les lignes d’articles qui sont disponibles et prêts à être expédiés seront traités et leur état modifié. La commande restera dans le stack géré par le scheduler aussi longtemps que toutes ses lignes ne sont pas prêtes pour expédition.

1. Expédition:

- matcher les commandes avec le type de camion

- gestion des routes

1. Notifications/Communication:

- notification au client des statuts de commande

- demande d’informations complémentaires

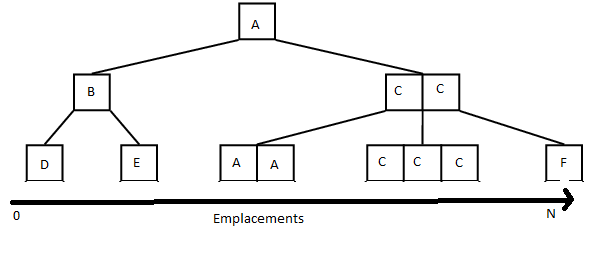
- …

## Informations utiles :

Comment réaliser la numérotation des flux?

* on nomme les acteurs: Fournisseur = F; Détaillant = D -> redondance des lettres, plusieurs lettres sur le même flux avec l’avantage de savoir directement en regardant un flux, tous les acteurs qui sont concernés par ce flux
* on nomme les flux end to end càd que tout ce qui est utilisé lors d’une émission de commande portera le même marqueur. Donc un flux pourrait aussi avoir plusieurs marqueurs selon qu’il est utilisé par différentes fonctionnalités.

Gestions des emplacements dans le stock



Nous prenons la somme totale des emplacements disponibles et divisons celle‐ci par deux pour obtenir la première position (A).

Pour obtenir les suivantes, nous effectuons la même opération mais en reprenant la position du nœud A. Ce qui nous donne la position B.

Et nous répétons l’opération pour le nombre d’éléments à stocker.

Cela permet de former des groupes d’emplacements qui seront flexibles selon le nombre d’articles. Les articles seront rajoutés en début et fin de groupe alternativement par rapport à la position du nœud.

Des déplacements seront possibles pour régulariser les positions mais ils seront de courtes distances.

Il se peut que les groupes se touchent en cas de stockage massif. Dans ce cas, une unité peut‐être placée provisoirement à un emplacement temporaire disponible OU un déplacement rapide d’un élément d’un groupe adjacent peut‐être effectué.

## Story board

*Le story board permet de visualiser rapidement ce que sera l'utilisation de l'application. Souvent réalisé avec un logiciel du style " PowerPoint " de Microsoft, le Story board montre l'enchaînement des écrans entre eux : où cliquera-t-on pour aller d'un écran à l'autre, comment se présenteront les formulaires de saisie, etc... C'est une étape importante pour que la maîtrise d'ouvrage puisse juger de la pertinence de l'ergonomie avec son besoin réel. En voyant ces story boards, il n'est pas rare non plus que d'autres besoins apparaissent.*

## Données

*le coeur de l'informatique reste la donnée en elle même. Il est important que soient décrites précisément les données qui seront manipulées par l'application. Il ne s'agit pas de donner ici une description informatique des données, mais une description " en français ". Il est important de savoir par avance que telle donnée est une date, telle autre une valeur multivaluée et de connaître ces valeurs possibles, telle autre une zone de saisie libre, etc... Ces informations seront bien entendu très utiles lors de la spécification technique.*

*1. Description des données communs pour toutes les Tables/Entités :*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *createdBy* | *Alphanumérique* | *Le responsable de la création de l’information* |
| *createdDate* | *Date* | *La date de la création de l’information* |
| *lastModifiedBy* | *Alphanumérique* | *Le responsable de la mise à jour de l’information* |
| *lastModifiedDate* | *Date* | *La date de la mise à jour de l’information* |

*2. Donnes de la table User/Utilisateur :*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *userId* | *Entier long* | *L’identifiant de la table User* |
| *email* | *alphanumérique* | *Adresse électronique* |
| *password* | *alphanumérique* | *Mot de passe* |
| *enabled* | *boolean* | *Activation du compte de l’utilisateur* |
| *firstName* | *alphanumérique* | *Prénom de l’utilisateur* |
| *LastName* | *alphanumérique* | *Nom de famille de l’utilisateur* |
| *role\_roleid* | *Entier long* | *Clé étrangère référençant la table Role* |

*3. Données de la table Rôle :*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *roleId* | *Entier long* | *L’identifiant de la table Role* |
| *roleName* | *alphanumérique* | *L’intitulé du role* |

*4. Données de la Table Person/Personne*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *personId* | *Entier long* | *L’identifiant de la table Person* |
| *companyName* | *alphanumérique* | *Nom/Raison sociale de la société* |
| *name* | *alphanumérique* | *Nom complet de la personne de contact* |
| *phone* | *alphanumérique* | *Numéro de téléphone* |
| *fax* | *alphanumérique* |  |
| *addressCity* | *alphanumérique* | *Ville du siège sociale* |
| *addressPostCode* | *alphanumérique* | *Code postal* |
| *addressBox* | *alphanumérique* | *Numéro boite* |
| *tva* | *alphanumérique* | *Numéro TVA* |
| *personType* | *MultiValeurs* | *Les valeurs possibles : Client, Fournisseur* |

*5. Données de la Table Orders/Commandes*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *ordersId* | *Entier long* | *L’identifiant de la table Orders* |
| *state* | *Multi-valeurs* | *Les valeurs possibles : Livré, Partiellement livré, En cours de traitement, à traiter* |
| *priority* | *Multi-valeurs* | *Les valeurs possibles : Ordinaire, Urgent, Exceptionnel* |
| *type* | *Multi-valeurs* | *Les valeurs possibles : Client,Fournisseur* |
| *person\_personId* | *Entier long* | *Clé étrangère référençant la table Person* |
| *archived* | *boolean* | *Suppression logique* |
| *delivery\_deliveryId* | *Entier long* | *Clé étrangère référençant la table Delivery* |

*6. Données de la Table OrderDétail/Ligne de commande*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *orderDetailId* | *Entier long* | *L’identifiant de la table OrderDetail* |
| *quantity* | *alphanumérique* | *La quantité demandée* |
| *orders\_ordersId* | *Entier long* | *Clé étrangère référençant la table Orders* |
| *article\_articleId* | *Entier long* | *Clé étrangère référençant la table Article* |
| *orderDetailState* | *Multi-valeurs* | *Les valeurs possibles : Livré, Partiellement livré, En attente.* |
| *archived* | *boolean* | *Suppression logique* |
| *comments* | *Texte* | *Observations* |
| *quantityReceived* | *alphanumérique* | *La quantité reçue* |

*7. Données de la Table Location/Emplacement*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *locationId* | *Entier long* | *L’identifiant de la table Location* |
| *reference* | *alphanumérique* | *La quantité demandée* |
| *articleQuantityAvaibleToStore* | *Entier long* | *Nombre d’emplacement disponible pour un article.* |
| *article\_articleId* | *Entier long* | *Clé étrangère référençant la table Article* |
| *state* | *Multi-valeurs* | *Les valeurs possibles : Vide, Occupé* |
| *zone* | *Multi-valeurs* | *Les valeurs possibles : In, Out, Stock.* |
| *locationType* | *alphanumérique* |  |
| *articleQuantity* | *entier* | *La quantité d’un article en Stock* |
| *completed* | *Boolean* |  |

*8. Données de la Table Delivery/Livraison*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *deliveryId* | *Entier long* | *L’identifiant de la table Delivery* |
| *vehicle\_vehicleId* | *Entier long* | *La clé étrangère référençant la tale Vehicle* |
| *deliveryState* | *Multi-valeurs* | *Valeurs possibles : Livré, En cours de livraison, Prêt à livrer* |

*9. Données de la Table Vehicle/Véhicule*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *vehicleId* | *Entier long* | *L’identifiant de la table Vehicle* |
| *label* | *Alphanumérique* | *Intitulé du véhicule* |
| *vehicleState* | *Multi-valeurs* | *Valeurs possibles : Disponible, Indisponible.* |
| *userSurface* | *décimale* | *La surface du camion occupée* |
| *userWeight* | *décimale* |  |
| *vehicletype\_vehicletypeId* | *Entier long* | *Clé étrangère référençant la table VehicleType* |

*10. Données de la Table VehicleType*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *vehicleTypeId* | *Entier long* | *L’identifiant de la table VehicleType* |
| *label* | *Alphanumérique* | *Intitulé du type de véhicule* |
| *surface* | *décimale* |  |
| *maximumWeight* | *décimale* |  |
| *threshold* | *décimale* |  |

*11. Données de la Relation DeliveryHasPalette*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *deliveryHasPaletteId* | *Entier long* | *L’identifiant de la relation deliveryHasPalette* |
| *delivery\_deliveryId* | *Entier long* | *Clé étrangère référençant la table Delivery* |
| *palette\_paletteId* | *décimale* | *Clé étrangère référençant la table Palette* |
| *paletteCount* | *Entier long* |  |

*12. Données de la table Article*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *articleId* | *Entier long* | *L’identifiant de la relation deliveryHasPalette* |
| *label* | *Entier long* | *Clé étrangère référençant la table Delivery* |
| *buyingUnitPrice* | *décimale* | *Clé étrangère référençant la table Palette* |
| *sellingUnitPrice* | *Entier long* |  |
| *packageQuantity* | *entier* |  |
| *Threshold* | *entier* |  |
| *Quantity* | *entier* |  |
| *Depth* | *décimale* |  |
| *Width* | *décimale* |  |
| *Height* | *décimale* |  |
| *Weight* | *décimale* |  |
| *description* | *alphanumérique* |  |
| *Volume* | *décimale* |  |
| *Archived* | *boolean* | *Suppression logique* |
| *person\_personId* | *Entier long* | *Clé étrangère référençant la table Person* |
| *box\_boxId* | *Entier long* | *Clé étrangère référençant la table Box* |

*13. Données de la Table Boite/Box*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *boxId* | *Entier long* | *L’identifiant de la relation Box* |
| *label* | *Entier long* | *Intitulé de la boite* |
| *height* | *Entier* |  |
| *width* | *Entier* |  |
| *length* | *Entier* |  |

*14. Données de la Table Palette*

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *paletteId* | *Entier long* | *L’identifiant de la relation Palette* |
| *type* | *alphanumérique* |  |
| *surface* | *Entier* |  |
| *paletteCount* | *Entier* |  |

*15. Données de la Table SupplierOrderDetail*

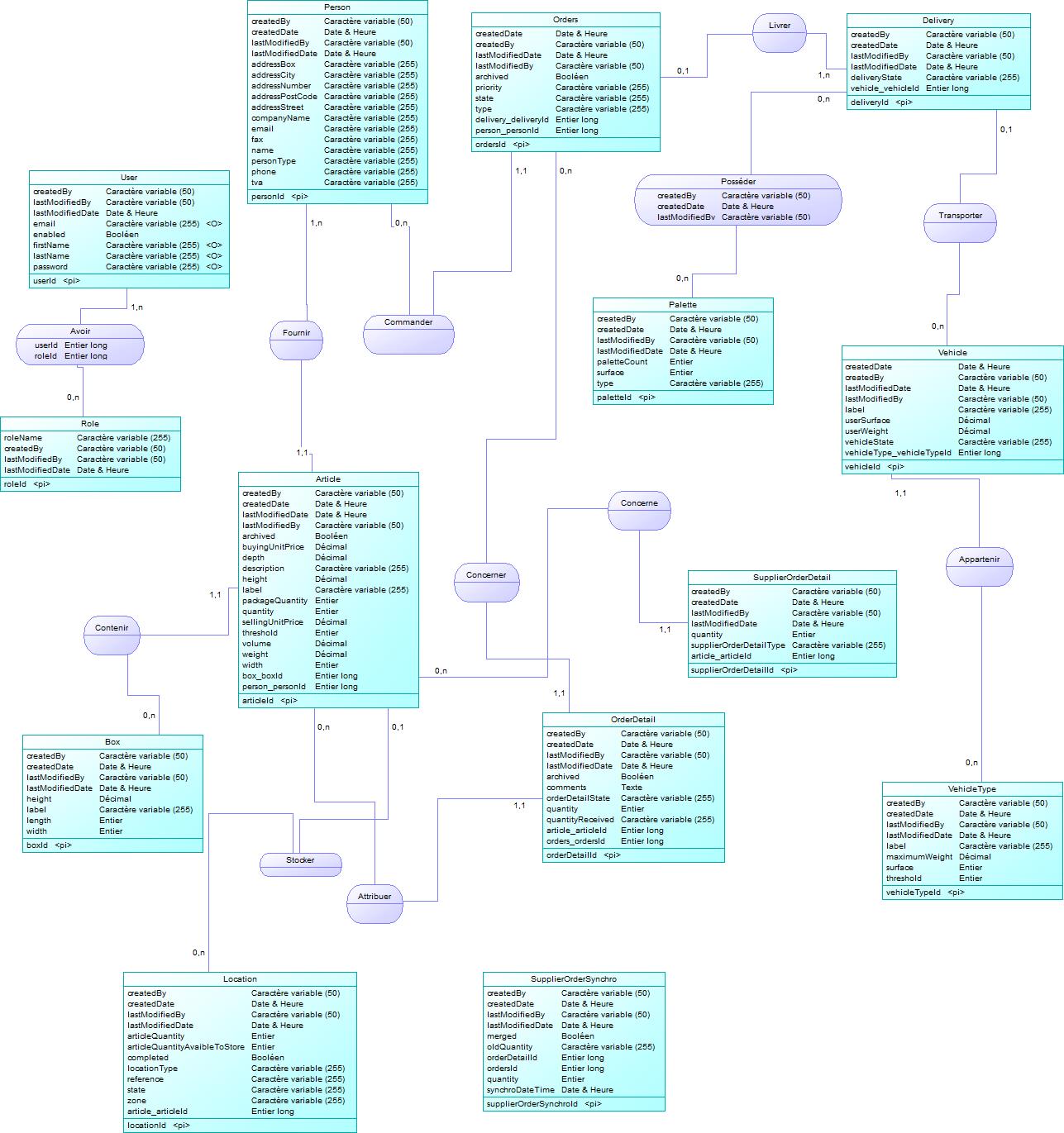
|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *supplierOrderDetailId* | *Entier long* | *L’identifiant de la relation Palette* |
| *article\_articleId* | *Entier long* | *Clé étrangère référençant la table Article* |
| *quantity* | *Entier* |  |
| *supplierOrderDetailType* | *Multi-valeurs* | *Les valeurs possibles : Seuil, Ordinaire* |

*16. Données de la Table SupplierOrderSynchro*

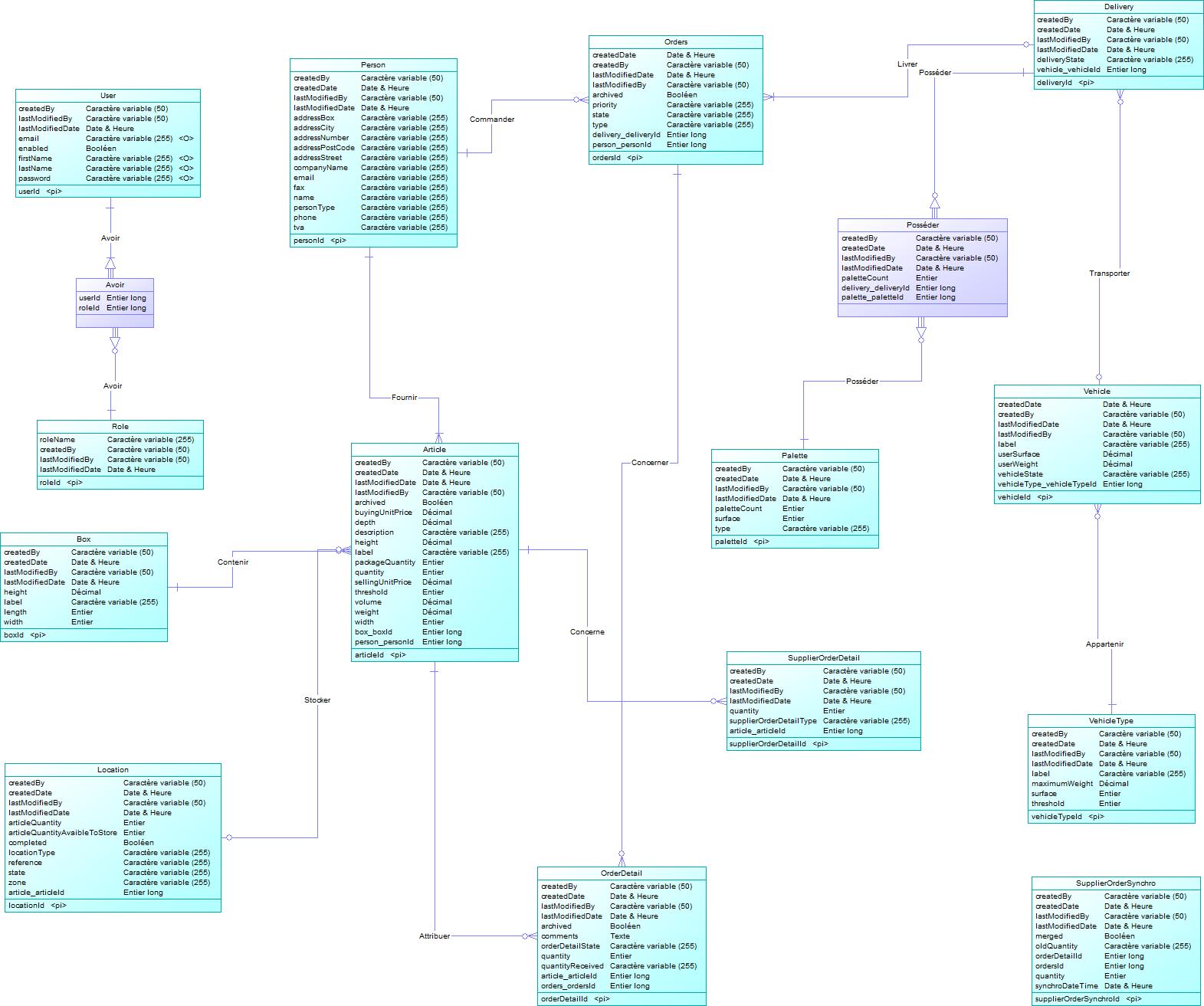
|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| *supplierOrderSynchroId* | *Entier long* | *L’identifiant de la relation SupplierOrderSynchro* |
| *ordersId* | *Entier long* | *Identifiant référençant la table Orders* |
| *orderDetailId* | *Entier long* | *Identifiant référençant la table OrderDetail* |
| *merged* | *boolean* | *Pour savoir si la ligne de commande a été fusionnée avec d’autres.* |
| *oldQuantity* | *alphanumérique* |  |
| *quantity* | *alphanumérique* |  |

# Modélisation de la base de données

* Modèle conceptuel de données :



* Modèle logique de données :



# Spécifications techniques

Après validation du CCD peut démarrer la rédaction du CCT. Ce document est un document technique : il décrit le " comment " on fait alors que les deux premiers décrivaient le " qu’est ce qu’il faut faire ".

## Spécifications logicielles

L’objectif de ce chapitre est de décrire comment sera développé l'application. On peut y décrire les algorithmes, les structures des répertoires, les règles de nommage des fichiers, etc...

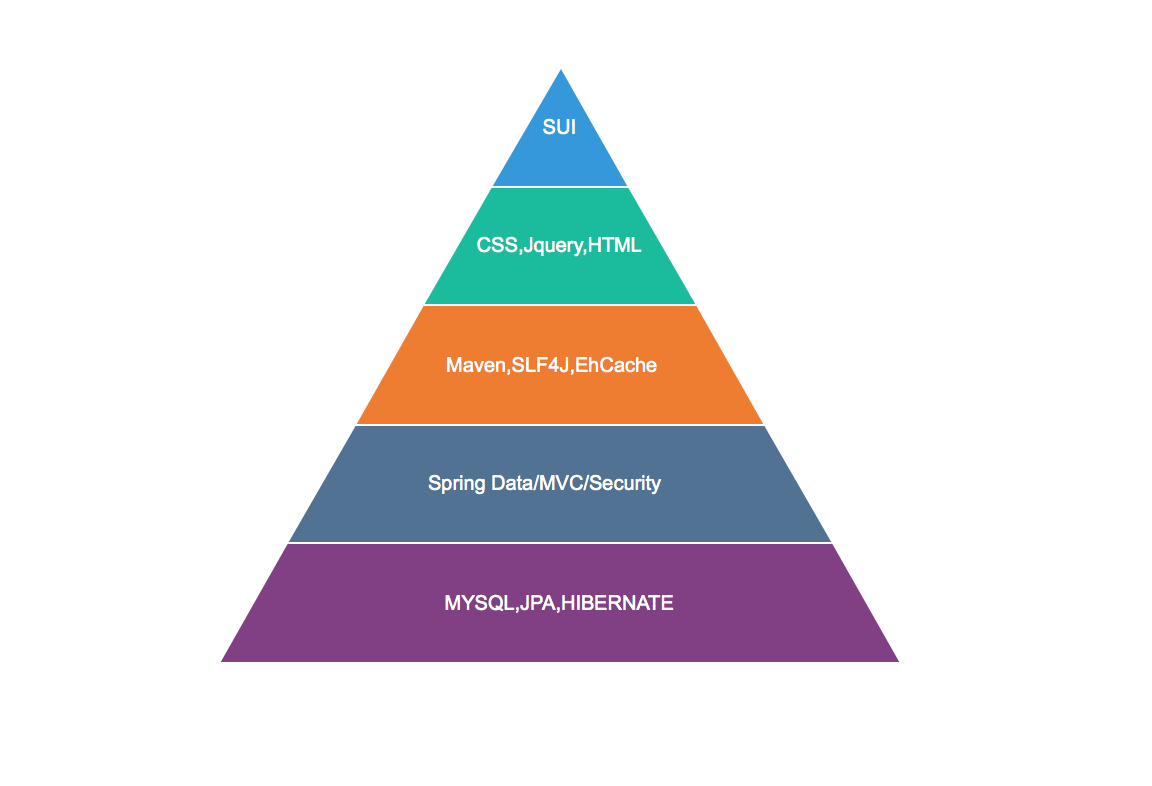
1. L’architecture de l’application :



Les règles de nommages des classes et des packages du projet suivent les conventions de codage Java pour :

* Faciliter la maintenance du projet.
* Améliorer la lisibilité du logiciel.

1. Les technologies utilisées :



**Spring Framework**: est une plate-forme Java qui fournit un soutien d'infrastructure complète pour développer des applications web. Il gère l'infrastructure de l’application afin que nous puissions se concentrer sur notre business, dans le cadre de ce projet on utilise les technologies Spring suivantes :

* Spring Security : Pour l’implémentation de la sécurité applicative de l’application (Gestion des autorisations et d’authentification et cryptage des mots de passes).
* Spring MVC : Pour l’intégration du modèle MVC (Model, View, Controller).
* Spring Core & Context : Ce module fournit la partie fondamentale du Framework Spring qui gère l’implémentation de l’architecture d’inversion de contrôle (IoC) et l’injection des dépendances.
* Spring Data : Une implémentation personnalisée des D.A.O (Data Access Object).

**Java persistent api (JPA)** : Est une interface de programmation Java permettant aux développeurs d’organiser des données relationnelles dans des applications utilisant la plateforme Java.

**Hibernate** : Est un Framework open source gérant la persistance des objets en base de données relationnelles, il apporte une solution aux problèmes d’adaptation entre le paradigme objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau.

**Apache Maven**: Outil de gestion et d’automatisation des dépendances d’un projet avec des modules externes.

**CSS/Html/JavaScript**: Des technologies standards pour le développement Front-End.

**JQuery** : Une bibliothèque JavaScript libre et multiplateforme qui facilite l’écriture des scripts coté client dans le code HTML des pages Web.

**SL4J & Logback**: Api Java et son implémentation utilisées pour la gestion des logs de l’application.

**Semantic UI (SUI)** : Collection d’outils utile à la création d’application web, c’est un ensemble qui contient des codes HTML/CSS et des éléments interactifs ainsi que des extensions JavaScripts en option.

**Apache Tomcat :** Conteneur web de servlets et JSP Java EE.

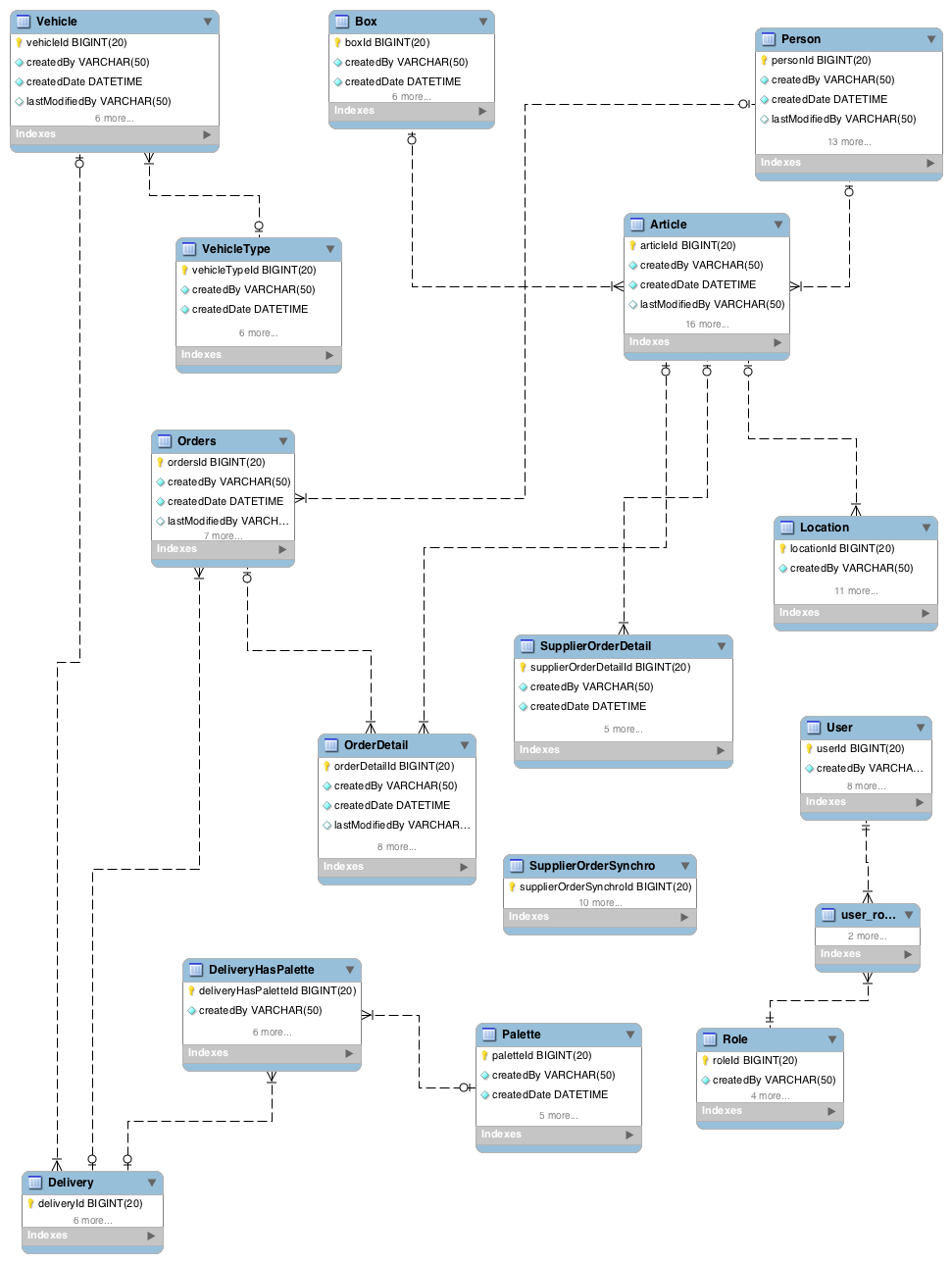
## Interfaces / flux

idem que précédemment, mais pour les interfaces et les flux.

## Spécifications des données

l'objectif de ce chapitre est de décrire ce que sera la base de données. On utilisera par exemple la représentation " Modèle Physique de Données " (MPD) de la méthode Merise : de nombreux logiciels de génie logiciel permettent de le construire très précisément.

Modèle physique de Données :



# Informations utiles

## Nombre d’utilisateurs

Décrire le nombre d’utilisateurs, leur situation géographique (sur le même site, .. ?)

## Nombre d’administrateurs

Combien de personnes auront droit à l’administration des données ? Décrire la population, leur localisation, ...

## Volumétrie

Quelle volumétrie de données ? Parler en terme de nombre de documents, de nombre d’informations, etc...

## Sauvegarde

Quels sont les besoins en terme de sauvegarde ?

## Indisponibilité

Quels sont les besoins en terme d’indisponibilité maximale de l’application dans le cas d’une panne majeure. Selon les besoins (dispo maximum), les moyens mis en œuvre seront différents, et le prix aussi !

## Sécurité

Quels sont les besoins en termes de sécurité ? Faut il mettre en œuvre des moyens de protection et de quel niveau ?

# *Phase de tests*

## Tests à réaliser

Description des tests à réaliser pour cette fonctionnalité. Chaque test est codifié de façon à pouvoir communiquer plus facilement avec mon prestataire

## Résultats attendus

J’indique le résultat que j’attends à l’issue de chaque test. Ce point est important à préciser (1) pour le prestataire pour lui rappeler certaines règles de gestion ou certains détails d’une fonctionnalité (2) pour une personne tiers quand ce n’est pas moi directement qui réalise les tests

## Résultats obtenus

Je reprends chaque code du tests et j’indique les résultats obtenus. J’indique un gros " OK " quand le résultat est concluant. Par contre, je marque un gros " ERREUR " en rouge pour indiquer toute erreur : j’explique le problème rencontré et je mets en annexe éventuellement des impressions écrans des messages d’erreurs rencontrés, avec quelques explications sur les corrections à apporter.

# *gestion DES COMMANDES FOURNISSEUR*

## Objectif

Le module de gestion de commandes fournisseur a pour objectif d’analyser et réaliser les processus et fonctionnalités pour la gestion d’approvisionnement du stock de l’entrepôt en cas de rupture de stock ainsi que le traitement des **Commandes Exceptionnelles \***.

Ce module s’articule autour de trois deux parties principales.

Une première partie sur l’approche conceptuel : identification des besoins fonctionnels, règles et contraintes à respecter ainsi que le modèle organisationnel de traitement (MOT).

Une deuxième partie présentera la phase de réalisation : Description des fonctionnalités implémentées, le résultat obtenu avec la description des interfaces d’utilisateurs.

\* **Commandes Exceptionnelles : il s’agit des commandes dont la quantité commandée dépassent le seuil fixé d’un article, dans ce cas le gestionnaire doit être en mesure d’interagir avec l’application pour établir la commande.**

## Conception

1. Identification des besoins fonctionnels

Pour la gestion de commandes fournisseur, nous allons identifier les besoins suivants :

* + - * 1. L’utilisateur doit pouvoir ajouter/Modifier et supprimer une commande fournisseur, et éventuellement la gestion des lignes de commandes.
        2. L’utilisateur doit pouvoir générer un bon de commande sur base d’un modèle prédéfinis en reprenant les informations enregistrées.
        3. Le système doit conserver les données de Quand et Qui à ajouter/Modifier ou supprimer une commande ou une ligne de commande.
        4. Le système doit pouvoir répondre aux besoins d’approvisionnement en cas de rupture de stock en générant automatiquement des commandes fournisseur d’un manière optimale.

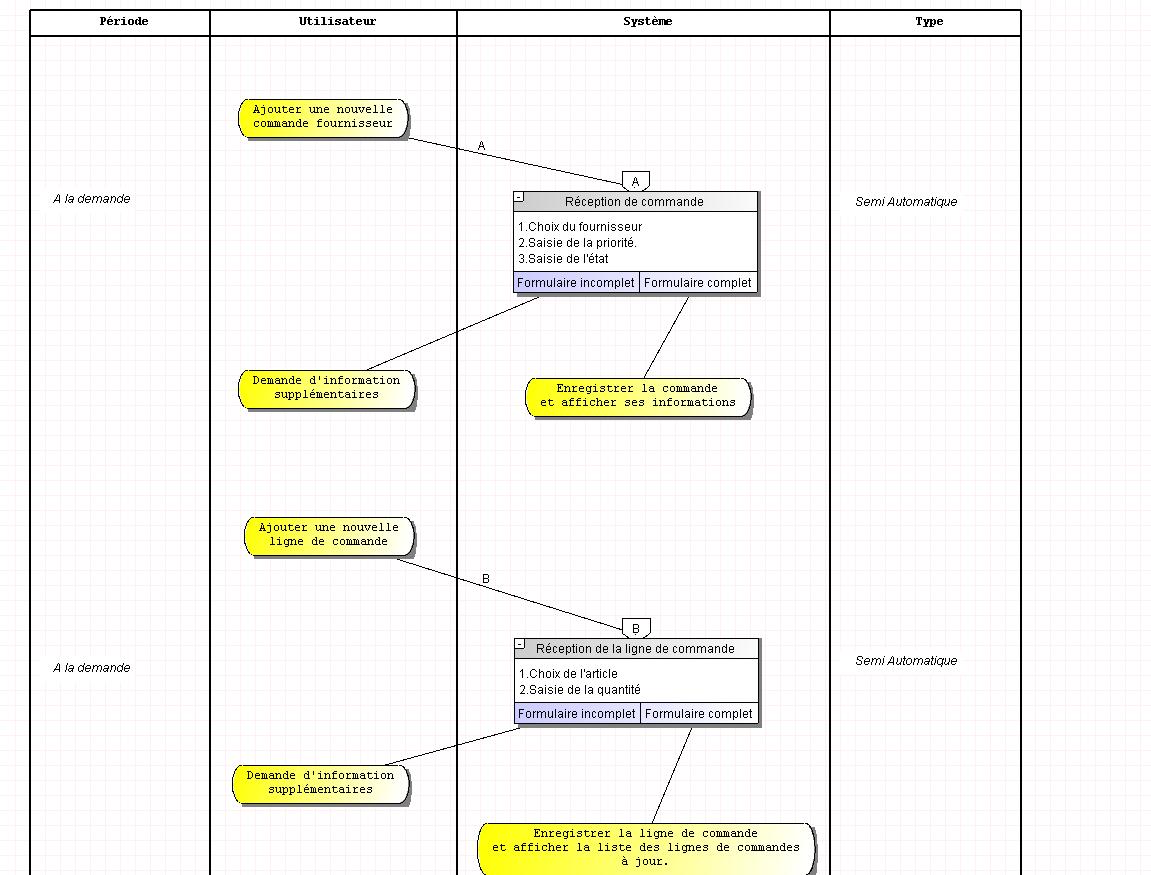
1. Règles et contraintes à implémenter

Les règles suivantes permettent de définir les contraintes et les conditions à prendre en compte lors de l’implémentation des différentes fonctionnalités pour la gestion des commandes fournisseur :

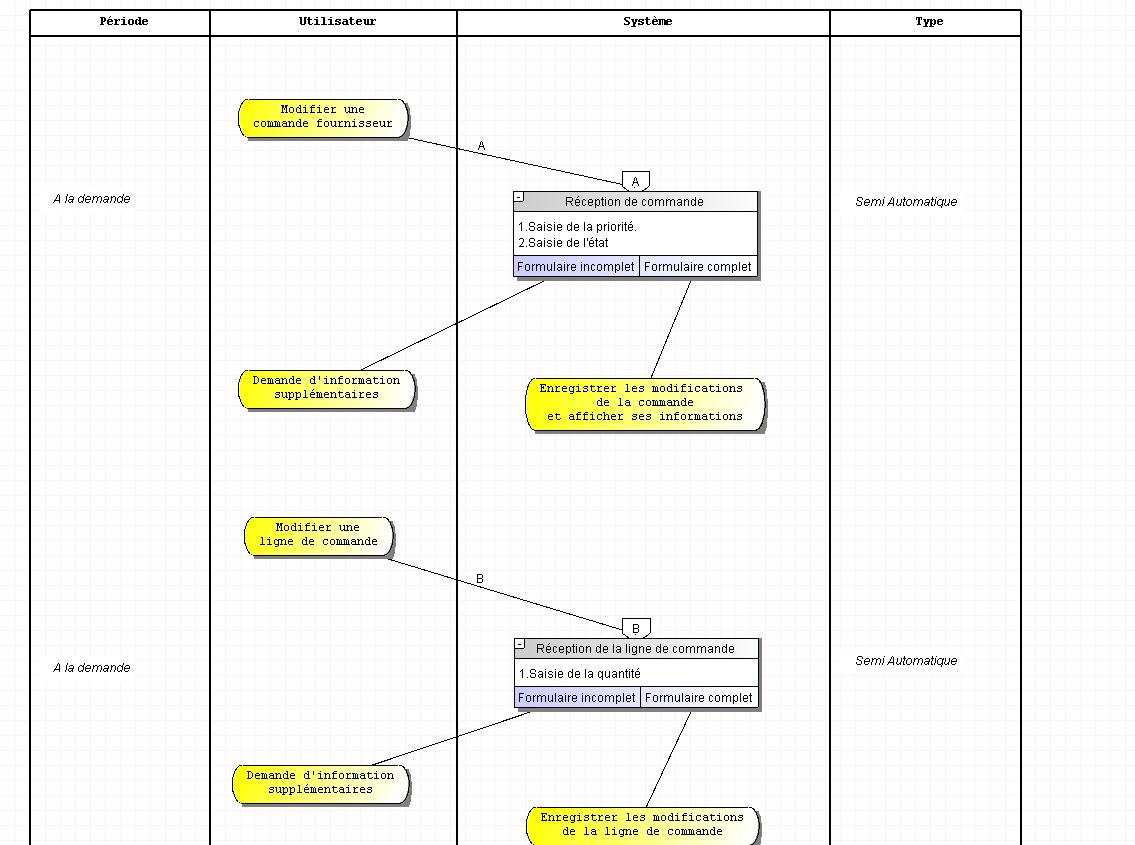
* + - * 1. Uniquement les deux états (A traiter et En cours de traitement) d’une commande seront proposés comme choix possible lors de la création d’une nouvelle commande fournisseur.
        2. L’utilisateur ne doit plus être ne mesure de modifier une commande ou une ligne de commande si sa livraison est complète et vérifiée.
        3. La consultation et/ou la modification des commandes fournisseurs n’est accessible qu’aux utilisateurs ayant le rôle de gestionnaire de commande fournisseur et notamment l’administrateur de l’application.
        4. On ne peut créer qu’une seule ligne de commande par article pour une même commande fournisseur.

1. Modèle organisationnel de traitement (MOT)

* Modèle de la création d’une commande et d’une ligne de commande fournisseur



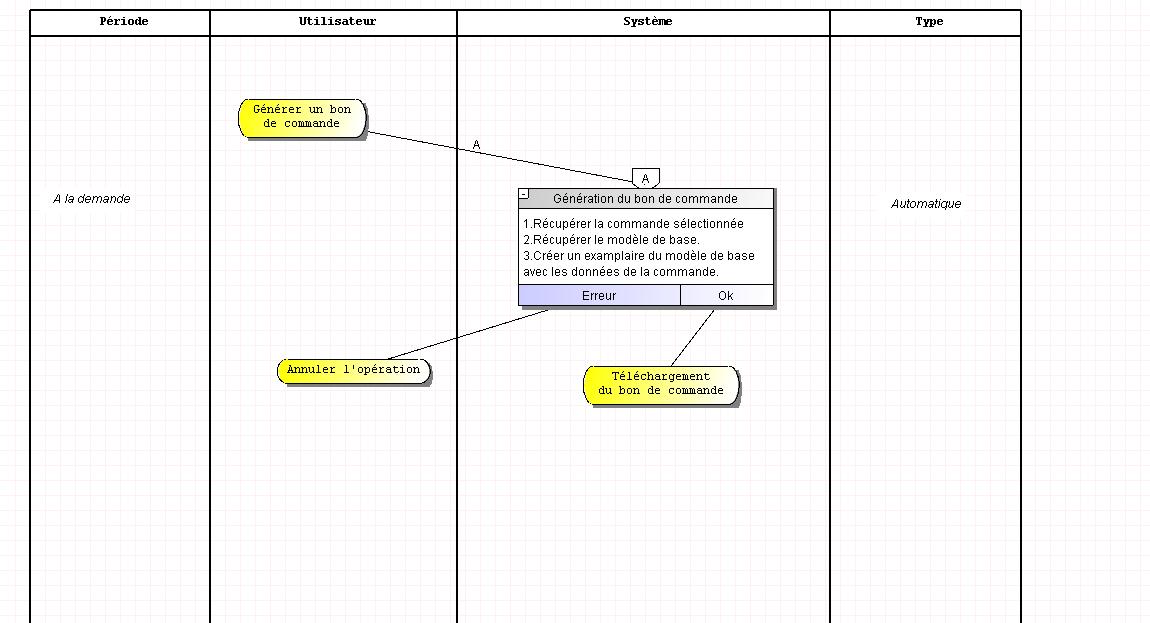
* Modèle de la modification d’une commande et d’une ligne de commande fournisseur



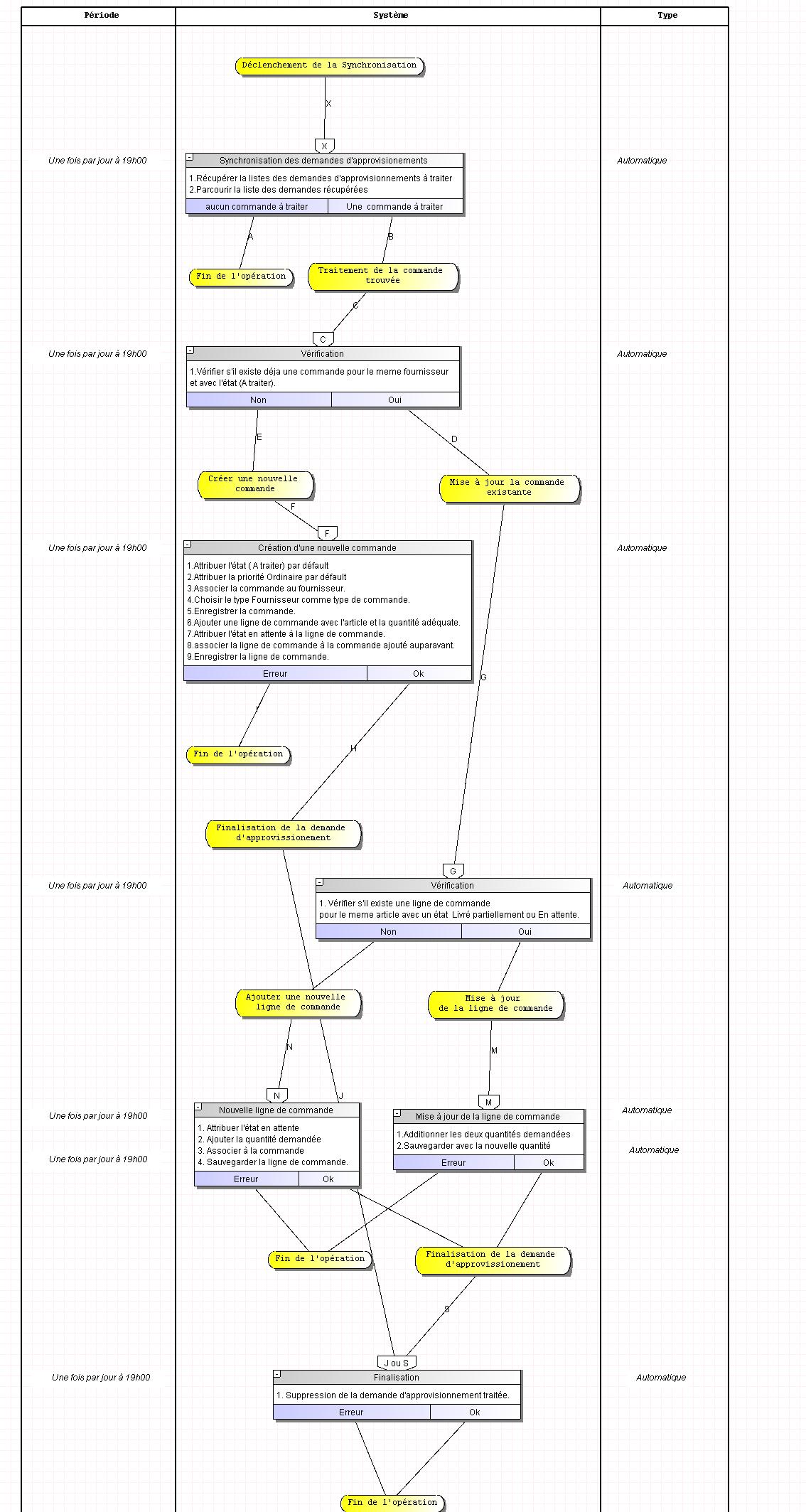
* Modèle de la suppression d’une commande et d’une ligne de commande fournisseur



* Modèle de la génération du bon de commande.



* Modèle de la synchronisation des approvisionnements automatique.



## Réalisation

1. Description des fonctionnalités implémentées

Il s’agit de l’ensemble des fonctionnalités détaillées implémentées par le module de gestion des commandes fournisseur.

**Création, Modification et Suppression d’une commande fournisseur**

Il s’agit des opérations de base pour la gestion des commandes fournisseur, pour la création d’une nouvelle commande, il faut choisir un fournisseur et introduire l’état de la commande parmi les choix possibles (A traiter ou En cours de traitement) ainsi que la priorité (Ordinaire, Urgent ou Exceptionnel).

Quant à la modification, il est possible uniquement si l’état de la commande est à traiter ou en cours de traitement et on peut modifier uniquement l’état et la priorité de la commande.

Enfin La suppression d’une commande consiste à masquer ses informations et les archivées.

**Consultation de l’ensemble des commandes fournisseur**

L’affichage d’un tableau contenant l’ensemble des commandes fournisseur à l’exception des archives, ce dernier fournit des fonctionnalités avancées de recherche et de tri ainsi que la pagination.

**Création, Modification et Suppression d’une ligne de commande fournisseur**

Pour créer une nouvelle ligne de commande il faut sélectionner l’article parmi les choix proposés du fournisseur associer à la commande ainsi que la quantité souhaitée.

Quand à La modification, le gestionnaire peut uniquement modifier la quantité de l’article déjà sélectionner et seulement si son état est en attente.

La suppression d’une ligne de commande à pour conséquent de masquer ses informations et les archivées.

**Détail d’une commande fournisseur**

Une fiche complète des informations d’une commande y compris l’ensembles de ses lignes de commande détaillées.

**Génération du Bon de commande fournisseur**

Cette fonctionnalité permet sur base d’un modèle de base de générer sous format (Docx) un document contenant les informations de la société (Client) et (Fournisseur) ainsi qu’un tableau des lignes de commandes avec le montant TTC de chaque ligne et enfin le montant total TTC et la date de la commande.

**Gestion automatique des demandes d’approvisionnements (Rupture de Stock)**

Il s’agit d’une synchronisation qui se déclenche une fois par jour à 19H00 pour traiter de façon optimale et automatique les demandes d’approvisionnements en cas de rupture de stock ou autres, Par optimale, en entend ici le groupement de l’ensemble des demandes sollicités pour le même fournisseur en additionnant leurs quantités consécutives pour chaque article.

1. Résultat Obtenu (Interfaces & Flux)

Au démarrage de l’application l’utilisateur doit s’authentifier pour consulter le module de gestion de commandes fournisseur.

Une fois connecté, l’utilisateur sera redirigé vers la page d’accueil ou seront afficher les différents modules.

Consulter le module de gestion de commandes fournisseurs permet de visualiser l’ensemble des commandes enregistrées sous forme d’un tableau avec la possibilité d’effectuer des recherches ou trier les commandes et éventuellement ajouter une nouvelle commande ou modifier/supprimer une existante ainsi que télécharger le bon de commande et consulter sa fiche complète.

Pour ajouter une nouvelle commande, il suffit de remplir correctement le formulaire suivant avec les informations décrites sur la figure suivante :

En cas d’erreur ou de manque d’informations le formulaire affichera les messages d’erreurs comme décrites sur la figure suivante.

Pour la modification d’une commande, uniquement la priorité et l’état sont modifiables :

Lors de la suppression d’une commande, une modal de confirmation s’affiche pour demander la confirmation de la suppression :

La fiche complète d’une commande comprend les informations de base ainsi qu’un tableau des lignes de commandes :

Pour ajouter une ligne de commande il faut remplir correctement le formulaire suivant :

1. Perspectives d’évolution

Le module de gestion de commande fournisseur est en mesure d’évoluer et intégrer d’autres fonctionnalités :

* + - * + Notifier le gestionnaire par e-mail pour l’informer sur l’état des opérations de la synchronisation.
        + Malgré que L’architecture de l’application est monolithique cela n’empêche pas qu’au futur le module de gestion de commandes fournisseur peut évoluer sous forme d’un micro service pour gagner en souplesse en terme de choix technologiques.
        + La génération de bon de commande peut être améliorer en intégrant la signature électronique et s’envoyer directement par e-mail via l’application.

## Conclusion

*Ce travail de fin d’étude m’a permis de mettre en œuvre les connaissances acquises en méthodes de conception, et méthodologie de développement.*