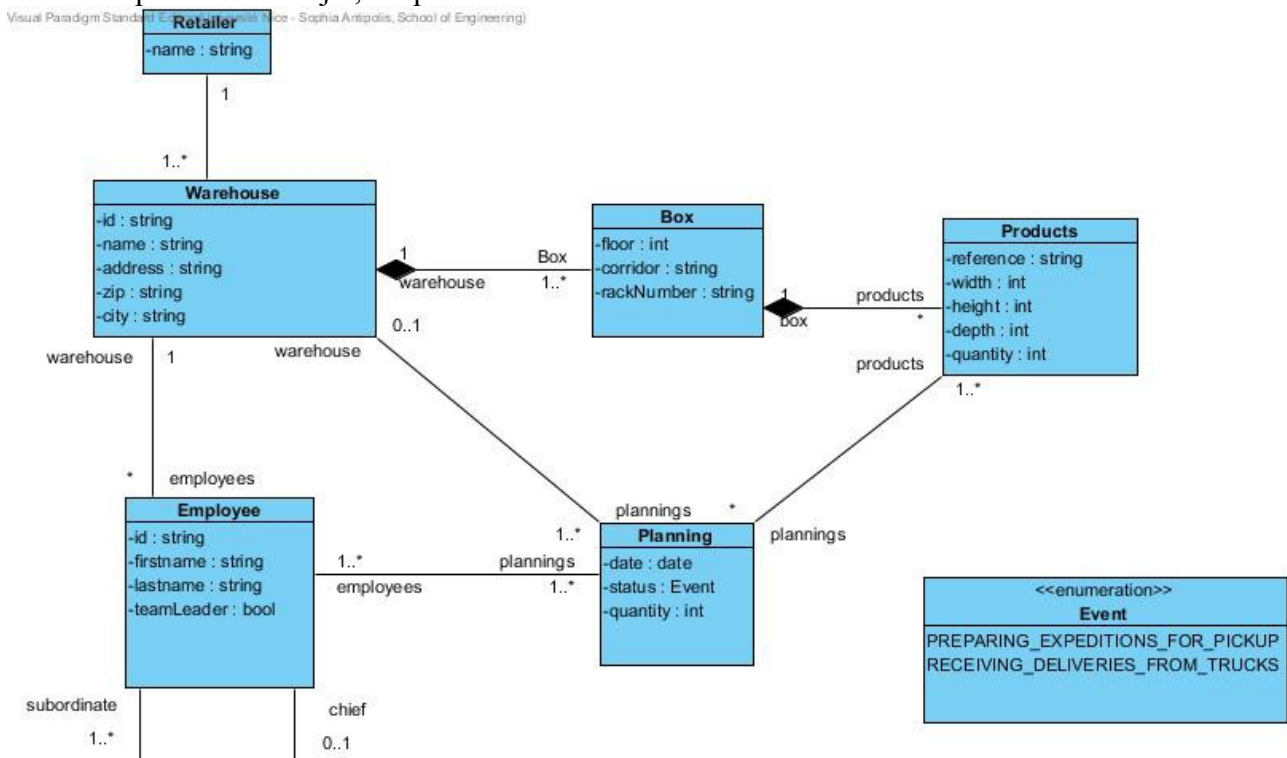


1 Business Model

Durant le parcours du sujet, la spécification client se formalise de cette manière :



En analysant les différentes données, nous nous trouvons avec les différents objets métiers :

- l'objet « *Warehouse* » qui correspond à l'entrepôt où sont stockés les produits
- un entrepôt contient différents « *Box* » de rangement des produits → implique une association de « *Warehouse* » vers « *Box* »
- un « *Box* » contient différents produits : « *Product* » → implique une association de « *Box* » vers « *Product* »
- un entrepôt emploie des employés : « *Employee* » → implique une association de « *Warehouse* » vers « *Employee* »
- un employé peut être « *Team leader* » et donc commander une équipe d'employés → implique une association de type récursive sur l'objet métier « *Employee* »
- un planning est une association de données (entre Warehouse, Products et Employee) permettant de relier les différents tâches à faire dans un warehouse entre un employée et des produits

2 Explications des différents cas d'utilisation

Les différents services sont exposés en REST. Cela permet de facilement s'intégrer au système de données. Les services REST permettent un découplage fort.

Les services créés sont les différentes interfaces accédant aux différents objets du Business Model. On aura alors un service pour l'objet Employee, un pour Warehouse, un pour Product, De plus, chaque service aura une interface pour la partie publique et une autre interface pour la partie privée. (ex : EmployeePrivateRestImpl pour le service privé REST de l'objet Employee)

Les services privés sont utilisés pour la gestion interne des données. Exemple, les gestionnaires

(teamleader) peuvent affecter les plannings aux employés.

URI templates	Méthodes HTTP	Description
<i>Scénario 1</i>		
/rest/redwarehouse/private/warehouses/{id_warehouse}/plannings/{date}	GET	<p>Permet de récupérer les différents plannings liés au Warehouse en fonction d'une date.</p> <ul style="list-style-type: none">Paramètres d'entrées :<ul style="list-style-type: none">id_warehouse → correspond à l'identifiant du warehousedate → correspond à la date à laquelle les plannings sont affectésParamètres de sorties :<ul style="list-style-type: none">un retour HTTP 200 indiquant que la requête a été effectuéeune liste de plannings appartenant au Warehouse durant la date indiquéErreurs pouvant survenir :<ul style="list-style-type: none">Erreur 404 → si le warehouse n'est pas trouvéErreur 500 → si la date n'est pas bien convertie
/rest/redwarehouse/private/employees/{id_employee}/plannings/assign	PUT	<p>Permet d'affecter à un employé des plannings.</p> <ul style="list-style-type: none">Paramètres d'entrées :<ul style="list-style-type: none">paramètres d'URL (PathParam)<ul style="list-style-type: none">id_employee → correspond à l'employé auquel on affecte les planningsparamètres de requête (QueryParam)<ul style="list-style-type: none">planningIds → correspond à une liste d'identifiant des id de plannings qui doivent être affectésParamètres de sorties :<ul style="list-style-type: none">un retour HTTP 200 indiquant que la requête a été effectuéeErreurs pouvant survenir :<ul style="list-style-type: none">Erreur 404 → si l'employé n'est pas trouvéErreur 500 → si la liste des plannings est vide ou inexistante dans les paramètres de requêtes
<i>Scénario 2</i>		
/rest/redwarehouse/private/employees/{id_employee}	GET	<p>Permet de connaître toutes les informations d'un employé ainsi que ses plannings.</p> <ul style="list-style-type: none">Paramètres d'entrées :<ul style="list-style-type: none">paramètres d'URL<ul style="list-style-type: none">id_employee → correspond à l'identifiant de l'employéParamètres de sorties :<ul style="list-style-type: none">un retour HTTP 200 indiquant que la requête a été effectuéeles informations de l'employéune liste de plannings affectés à l'employéErreurs pouvant survenir :<ul style="list-style-type: none">Erreur 404 → si l'employé n'est pas trouvé
<i>Scénario 3</i>		
/rest/redwarehouse/private/employees/{id_employee}/plannings/{id_planning}/register	POST	<p>Permet d'ajouter les produits dans l'entrepôt. Si le produit existe déjà dans l'entrepôt, le système va mettre à jour le compteur du produit, sinon, le système va créer un nouveau produit et rechercher un box adéquat. De plus, le planning sera retiré de l'employé.</p> <ul style="list-style-type: none">Paramètres d'entrées :

		<ul style="list-style-type: none"> paramètres d'URL (PathParam) <ul style="list-style-type: none"> id_employee → correspond à l'identifiant de l'employé id_planning → correspond à l'identifiant du planning que le système doit enregistrer Paramètres de sorties : <ul style="list-style-type: none"> un retour HTTP 200 indiquant que la requête a été effectuée les informations de l'employé une liste de plannings affectés à l'employé Erreurs pouvant survenir : <ul style="list-style-type: none"> Erreur 404 → si l'employé n'est pas trouvé ou si le planning n'a pas été trouvé
--	--	--

Les services publics sont utilisés par les utilisateurs extérieurs (ex : Amazon) à l'entrepôt (ici : RedWarehouse). Exemple, Amazon peut créer un planning pour les employés pour sortir du stock des objets de l'entrepôt (ex : enlever 90 montres du stock).

URI templates	Méthodes HTTP	Description
<i>Scénario 2</i>		
/rest/redwarehouse/public/products/{ref_product}/	GET	Permet de connaître toutes les informations d'un produit et de sa localisation. <ul style="list-style-type: none"> Paramètres d'entrées : <ul style="list-style-type: none"> paramètres d'URL (PathParam) <ul style="list-style-type: none"> ref_product → correspond à la référence du produit que l'on cherche Paramètres de sorties : <ul style="list-style-type: none"> un retour HTTP 200 indiquant que la requête a été effectuée les informations du produit Erreurs pouvant survenir : <ul style="list-style-type: none"> Erreur 400 → si la référence du produit est mal renseignée (ex : vide) Erreur 404 → si le produit n'est pas trouvé
<i>Scénario 4</i>		
/rest/redwarehouse/private/warehouses/plannings	GET	Permet de connaître tous les warehouse ainsi que leurs plannings. <ul style="list-style-type: none"> Paramètres de sorties : <ul style="list-style-type: none"> un retour HTTP 200 indiquant que la requête a été effectuée la liste de Warehouse du système avec les différents plannings pour les différents warehouses
/rest/redwarehouse/private/warehouses/{id_warehouse}/plannings/	GET	Permet de connaître tous les plannings d'un warehouse. <ul style="list-style-type: none"> Paramètres d'entrées : <ul style="list-style-type: none"> id_warehouse → correspond à l'identifiant du warehouse Paramètres de sorties : <ul style="list-style-type: none"> un retour HTTP 200 indiquant que la requête a été effectuée une liste de plannings appartenant au Warehouse durant la date indiqué Erreurs pouvant survenir : <ul style="list-style-type: none"> Erreur 404 → si le warehouse n'est pas trouvé