Université Nice Sophia Antipolis

Membre de UNIVERSITÉ CÔTE D'AZUR

POLYTECH NICE-SOPHIA
Département Sciences Informatiques

Rapport de projet
ISA

Philippe Collet & Guilhem Molines

Equipe I:

Vincent Coppé | Bastien Jard | LI Zeyang | LU Yao | Esther Tchangani

INTRODUCTION

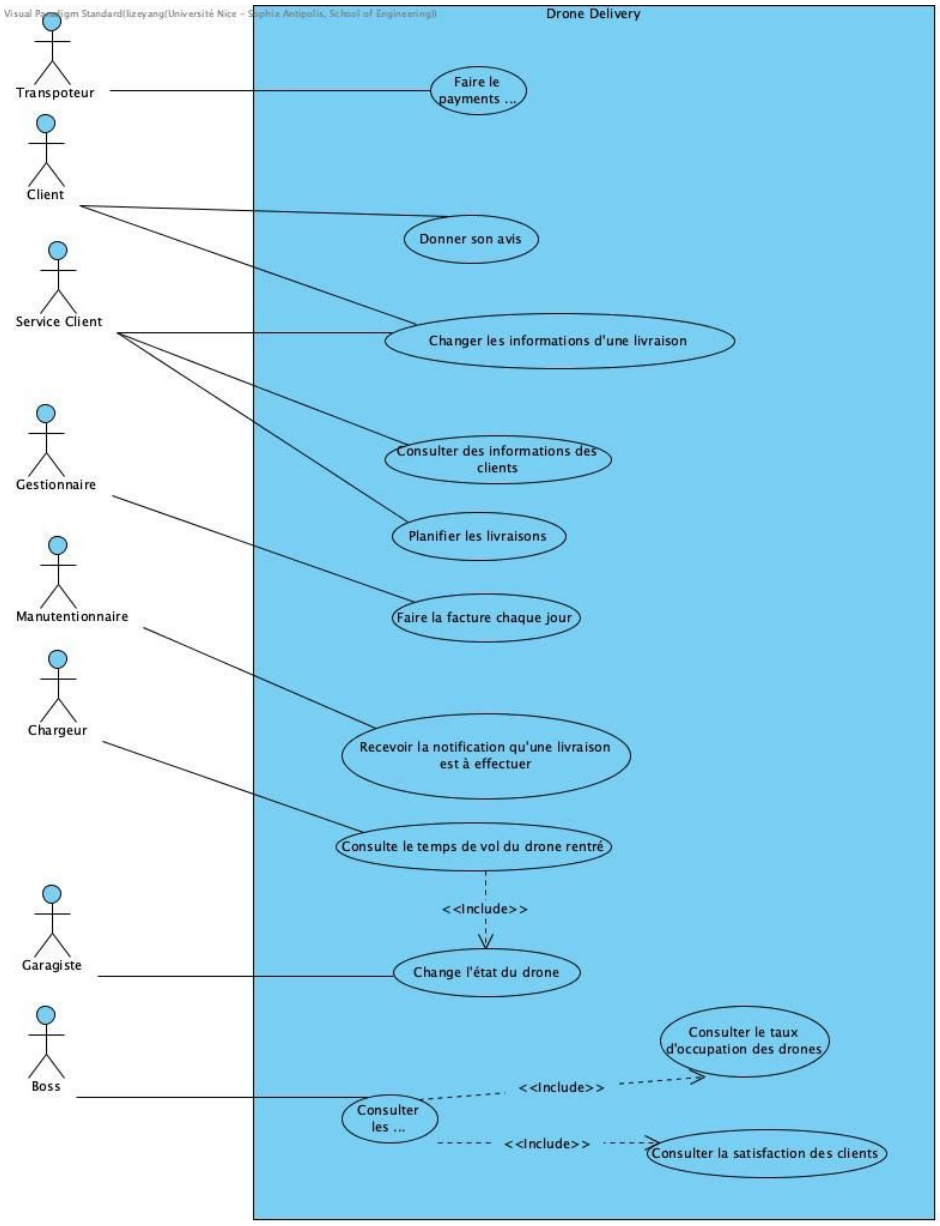
Un système logiciel doit répondre aux exigences des parties prenantes et aux contraintes qui existent durant sa production. Pour ce faire, il est important de partir sur une bonne architecture de manière à répondre aux spécifications.

Dans ce rapport, nous décrivons les différentes étapes qui permettent d'aboutir au système de livraison par drone. Nous présentons dans un premier temps, les acteurs du système ainsi que les différentes interactions de ceux-ci avec le système. Ensuite, nous décrivons la structure du système par un diagramme de composant et nous finissons par la présentation des objets métiers du système.

I. VUE FONCTIONNELLE

1. Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation du système drone delivery est le suivant:



Les principaux acteurs de notre système et le rôle joué par chacun d'eux sont les suivants :

Le client : L'un des indicateurs auxquels s'intéresse Bob est la satisfaction du client. Et pour cela, il est nécessaire que le client donne son avis sur la prestation (délai de livraison par exemple). Le client interagit avec le système Drone Delivery en envoyant un message ou par vote via la plateforme. Ce cas d'utilisation permet la réalisation du cas d'utilisation « Consulter la satisfaction des clients » réalisé par le boss.

Les horaires de livraison sont définis par appel téléphonique entre le client et le service client mais, le client peut renseigner les informations de livraison via la plateforme.

Le boss : Une fois les informations sur la satisfaction du client rentrées, le boss se charge de réaliser des statistiques par rapport à ces informations afin d'améliorer le service de la livraison. Le boss consulte aussi le taux d'occupation des drones.

Le service client : Il est en charge de la planification des livraisons. Il renseigne les informations de livraisons.

Le manutentionnaire : Le chargement et le déchargement des camions est une action réalisée physiquement, donc pas d'interaction avec le système. Cependant, Marcel communique avec le système qui lui envoie une notification lorsqu'une livraison doit être effectuée d'où le cas d'utilisation « Recevoir la notification qu'une livraison est à effectuer. Après avoir chargé le drone, Marcel appuie sur le bouton « Départ » qui lance le drone et qui change l'état de la commande du client, mais l'action d'envoi du drone est faite par le système d'où le fait de ne l'avoir pas renseigné comme un cas d'utilisation correspondant au manutentionnaire.

Le chargeur : Son rôle dans le système est de consulter le temps de vol des drones et de modifier leur état dans le système en fonction de celui-ci.

Le garagiste: Il est chargé de la révision du drone. Après la révision, il interagit avec le système en changeant l'état du drone.

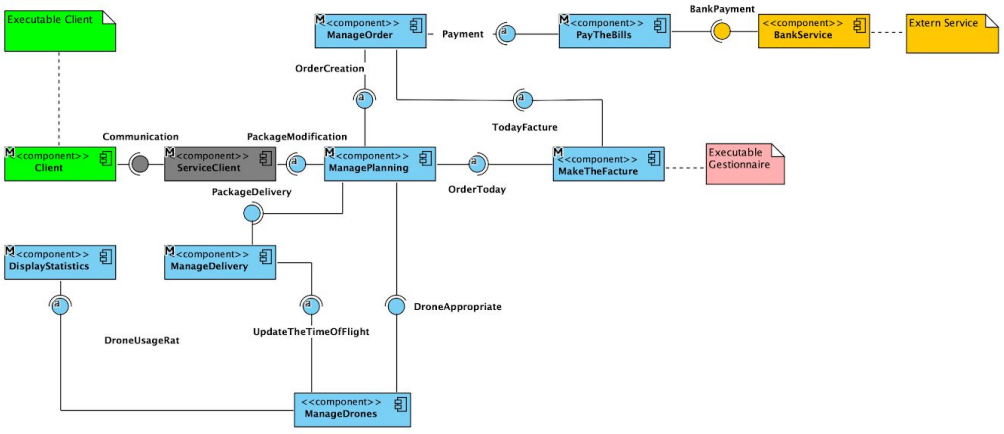
Le gestionnaire: Le système se charge de calculer le nombre de colis livrés, le montant dû par chaque transporteur et d'envoyer une notification à l'approche de

l'échéance de paiement. Le rôle du gestionnaire dans le système est donc de consulter ces informations et en fonction de cela, elle envoie une facture au transporteur.

Le transporteur: Il peut utiliser la plateforme pour payer ses factures à la société

2. Diagramme des composants

Le diagramme des composants montre la structure des composants de notre projet.



Le composant Client : Fournit une interface au client pour qu'il puisse communiquer avec le système.

Le composant ServiceClient : Il fournit aux clients des fonctions de soumission et les clients peuvent soumettre des demandes de livraison au système.Il reçoit et traite les demandes des clients.

Le composant PayTheBills : Il est responsable de la facturation de la commande et d'intégrer l'interface de paiement externe au système .

Le composant BankService : Composant externe qui fournit des fonctions de paiement bancaire.

Le composant MakeTheFacture : Il fournit la fonction de génération de facture. Il accepte les informations de commande du jour et les transmet au OrderManager de commande après avoir généré la facture.

Le composant DisplayStatistic : Il est responsable de la réception et du traitement des statistiques d'occupation du drone et de satisfaction client, ainsi que de fournir une fonction pour les afficher.

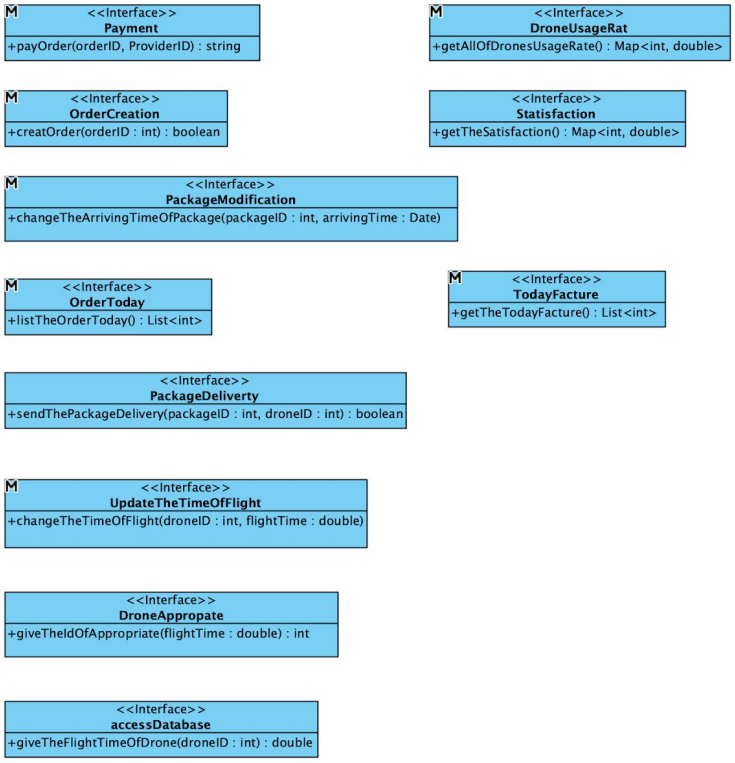
Le composant ManageOrder : Il est de créer les commandes avec les factures qui y sont associés et de fournir celles qui doivent être payées.Il transmet les commandes au **ManagePlanning**.

Le composant ManagePlanning : reçoit les informations de commande du OrderManager.Il est chargé d'assigner un colis à un drone, et de fournir visuellement un planning de vol des drones. Il peut également mettre à jour les informations concernant une commande.

Le composant ManageDelivery : reçoit les informations de commande du **ManagePlanning** et est chargé ensuite de calculer le temps de vol d'un drone pour une livraison, et met à jour le temps total de vol du drone. Il fournit une fonction permettant de signaler que le drone est parti livrer une commande .

Le composant ManageDroneInforamtion : Il est chargé de fournir et de mettre à jour les informations des drones, comme leur état et leur temps de vol. Il fournit l'ID de drone approprié au **ManagePlanning** selon les exigences du drone utilisé par le **ManagePlanning**.Il peut aussi consulter les informations du drone.

3. Interfaces fonctionnelles



Payment: les opérations liées au paiement des factures des transporteurs.

OrderCreation: l'opération pour créer une commande qui contient les paquets que le OrderManager a sélectionné.

PackageModification: l'opération pour modifier l'heure d'arrivée du colis correspondant en fonction de la demande de le client.

OrderToday: l'opération pour obtenir les commandes du jour afin de pouvoir générer les factures correspondantes

PackageDelivery : l'opération pour créer une commande contenant l'identifiant du drone et du colis à transporter par ce drone

UpdateTheTimeOfFlight: l'opération pour changer le temps du vole

DroneAppropriate: l'opération pour fournir un drone approprié selon les exigences du gestionnaire pour le temps de vol du drone

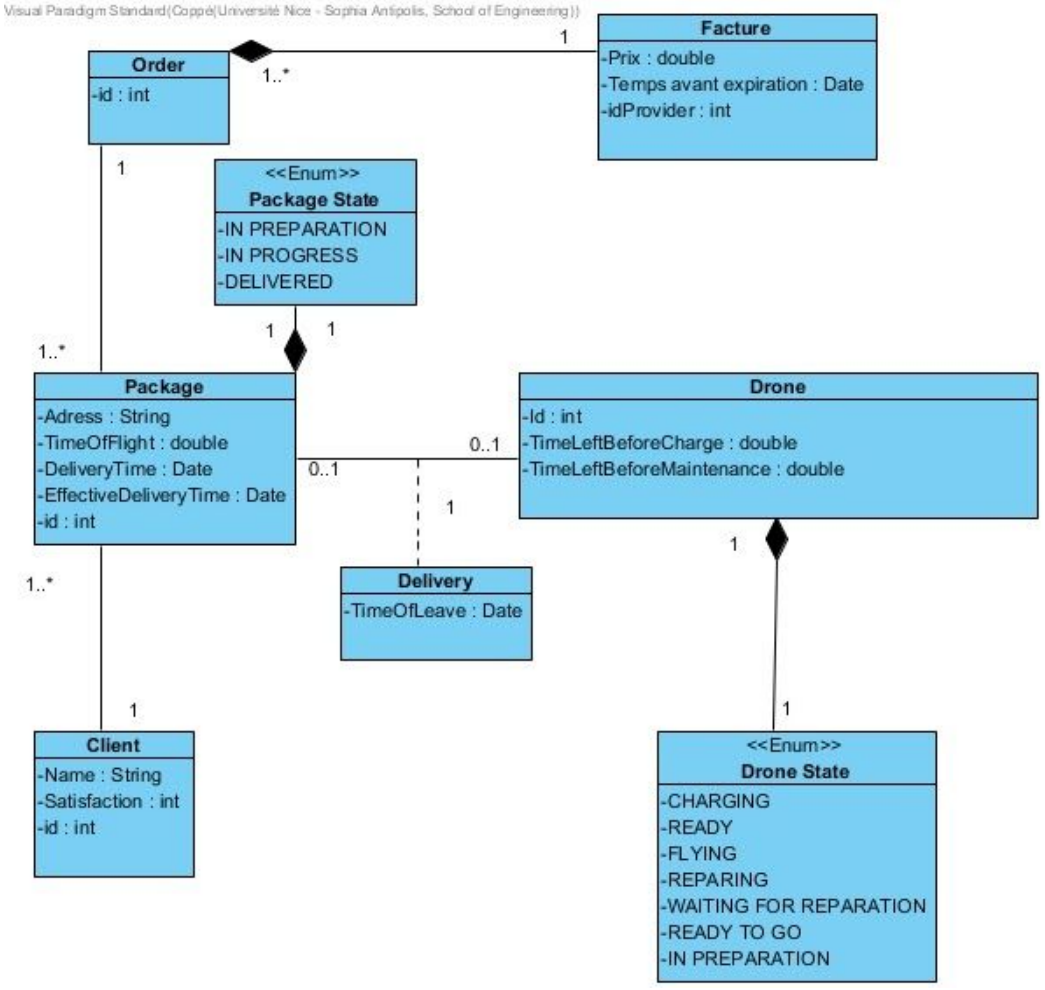
DroneUsageRate: l'opération pour calculer le taux d'occupation des drones

Satisfaction: l'opération pour calculer la satisfaction des clients

TodayFacture: l'opération pour générer les factures du jour à partir des commandes concernées

II. VUE DEVELOPPEMENT

1. Diagramme de classe des objets métiers du système



Nous avons plusieurs objets basique dans notre système, Drone qui représente les drones dans notre projet et qui possède plusieurs arguments tels qu'une Enum Drone State qui permet de savoir si le drone est en Charge, en vol en réparation etc, il possède aussi un package lorsqu'il est READY TO GO ou FLYING. Nous avons un Objet pour le package auquel est associé un client, une commande la date de livraison demandé ainsi que la date de livraison auquel le colis a effectivement été livré et une enum pour savoir si le colis est en train d'être livré, si il a été livré ou si il n'est pas encore parti. Il y a un objet Facture possède son prix l'id du fournisseur associé à celle ci et sa date d'expiration .

CONCLUSION

Cette première étape du projet "Drone Delivery" nous a permis de poser les bases de notre futur système c'est à dire la structure que nous souhaitons donner à notre application.