

# **Rapport d'analyse : réservation d'un billet d'avion**

## **SOMMAIRE**

<b>I-</b>	<b>Contexte et rappel du sujet</b>	<b>2</b>
<b>II-</b>	<b>Hypothèses et reformulation</b>	<b>2</b>
	<b>Diagramme de cas d'utilisation</b>	<b>4</b>
	<b>Les hypothèses</b>	<b>5</b>
	<b>La hiérarchie des fonctionnalités</b>	<b>6</b>
<b>III-</b>	<b>Diagramme de classe et modélisation</b>	<b>7</b>
<b>IV-</b>	<b>Modélisation dynamique</b>	<b>10</b>
	<b>Diagramme de séquence</b>	<b>11</b>
	<b>Diagramme d'activité</b>	<b>13</b>
	<b>Diagramme d'état-transition d'un avion</b>	<b>14</b>
<b>V-</b>	<b>Conclusion</b>	<b>14</b>

## I- Contexte et rappel du sujet

Dans un contexte mondial de démocratisation des échanges des hommes et des marchandises grâce notamment aux routes aériennes, la mise en place d'un système de réservation efficace est essentielle pour faciliter les besoins des clients et des compagnies.

Dans ce but, il nous a été demandé de faire une application de réservation de billets d'avion destinée à être utilisée par les compagnies aériennes pour gérer les réservations mais également par les clients pour qu'ils saisissent les critères de vol eux-mêmes et choisissent le vol qui leur convient le mieux. De plus si possible il serait pertinent d'avoir une visualisation de quelques fonctionnalités comme les différents vols disponibles, un plan de vol ... Il s'agit donc de faire un produit s'apparentant à un CRS<sup>1</sup> « computer reservations system » ou un GDS<sup>2</sup> « Global distribution system » pour une utilisation du client ou de la compagnie aérienne.

## II- Hypothèses et reformulation

Afin de comprendre au mieux les problèmes possibles liés à une réservation nous avons listé les différents besoins des deux acteurs qui nous ont semblés importants et ce qu'il serait intéressant de visualiser. Nous avons regroupé ce que nous avons retenu dans le tableau suivant :

Client : Réservations de billets	Compagnie : Gérer la compagnie	Visualiser
<ul style="list-style-type: none"><li>- Proposer des choix (dates, destinations, sièges,...)</li><li>- Faire un billet</li><li>- Gérer sa réservation (modifier, annuler...)</li></ul>	<ul style="list-style-type: none"><li>- Affecter des avions (capacité, autonomie, coût : remplissage, consommation ...)</li><li>- Gérer les routes (taux remplissage, taxes..)</li><li>- Gérer un plan de vol</li><li>- Gérer les passagers (affecter aux vols...)</li><li>(- Promos)</li></ul>	<ul style="list-style-type: none"><li>- Vols disponibles (en fonction de destination, dates, promos..)</li><li>- Routes (sur la carte)</li><li>- Trajectoire des avions (plan de vol, position, en fonction du temps)</li></ul>

Etant tous les deux assez familiers avec les réservations de billet d'avion grâce à nos voyages

<sup>1</sup> CRS : ils permettent de réserver un service dans le temps pour une personne ou un groupe. Ces systèmes s'intègrent dans plusieurs domaines notamment celui du transport tel que l'avion. Ils peuvent être directement accessibles aux personnes non professionnelles grâce au développement de serveurs web accessibles depuis un navigateur.

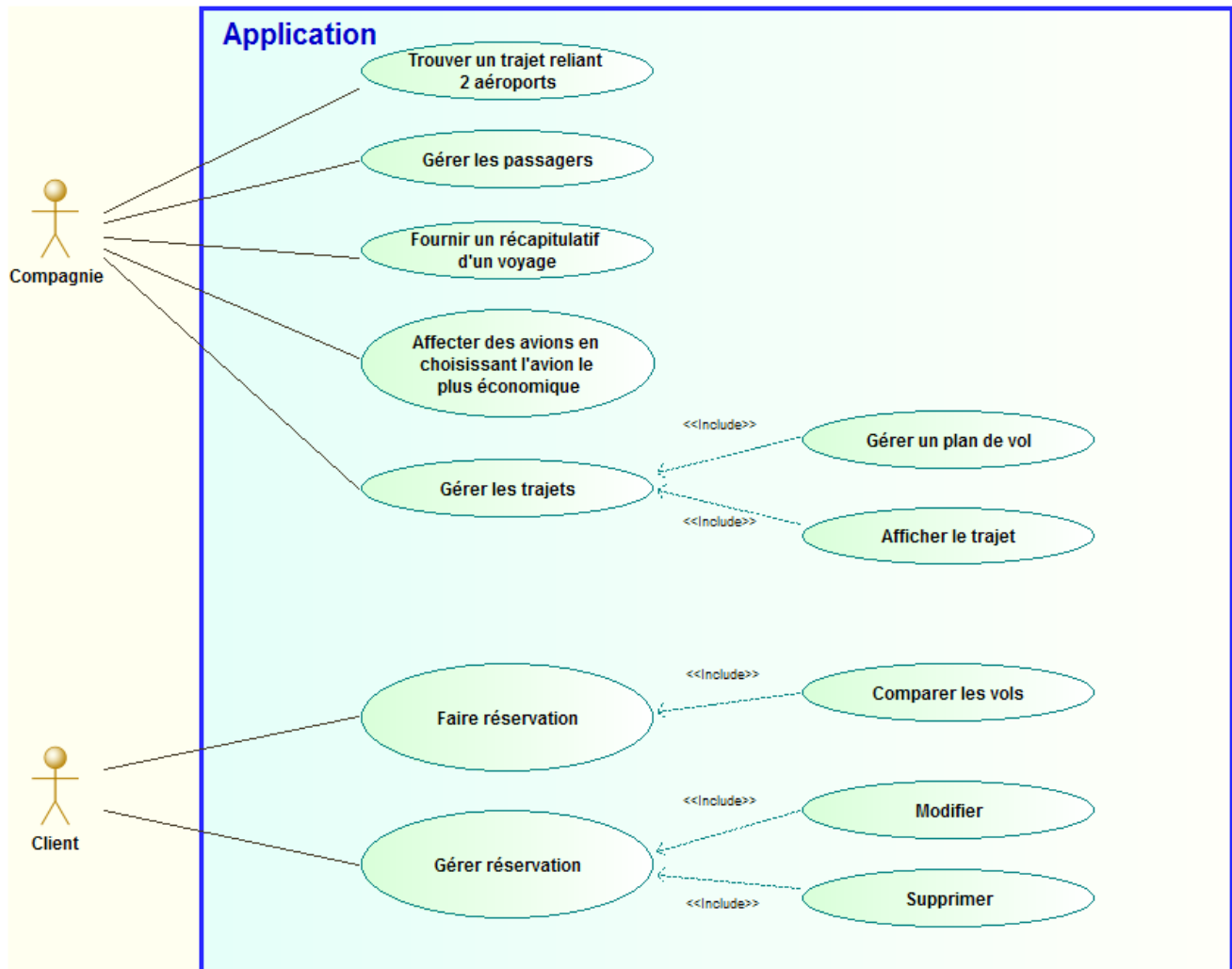
<sup>2</sup> GDS : ces plates-formes électroniques de gestion des réservations qui permettent aux agences de voyages de connaître en temps réel l'état du stock des différents fournisseurs de produits touristiques comme les compagnies aériennes et de réserver à distance.

nous avons eu certaines idées pour améliorer les réservations et les rendre ainsi plus intéressantes avec une fidélisation des clients. Ces fonctionnalités pourraient être mises en place si nous avons assez de temps :

- fidéliser son client, en lui attribuant des points (comme des miles)
- faire des promos lors de période de creux (hors vacances par exemple)
- proposer des hôtels avec les destinations choisies pour faciliter les recherches du client et une voiture si besoin
- proposer une assurance en cas d'imprévus
- proposer de la flexibilité au niveau des dates et d'autres options mais cela pourrait engendrer des frais supplémentaires (repas particulier, ...)
- il pourrait être astucieux de montrer le nombre de points que la réservation permet de gagner et de montrer l'itinéraire aller-retour

## Diagramme de cas d'utilisation

Nous avons ainsi réalisé le diagramme d'activité suivant en intégrant seulement les fonctionnalités de base que devrait faire l'application:



Les deux acteurs principaux ont chacun des actions réalisables distinctes. Le client s'occupe exclusivement de sa réservation : une fois la réservation faite, il peut la modifier ou la supprimer. Tandis que la compagnie peut effectuer plusieurs actions liées aux vols, aux avions, aux routes ... La compagnie peut trouver un trajet reliant 2 aéroports, gérer les passagers, fournir un récapitulatif de voyage avec toutes les informations concernant sa destination, son vol etc.. La compagnie peut affecter des avions en choisissant l'avion le plus économique et gérer les trajets en générant un plan de vol et en l'affichant.

## Les hypothèses

Au moment-même de la rédaction de notre rapport d'analyse toutes nos hypothèses ne sont pas encore définitives, certaines nécessitent une réflexion plus poussée en relation avec l'avancement de la programmation pour se rendre réellement compte des hypothèses nécessaires, en voici une partie non-exhaustive :

- Pour l'instant la gestion des avions au tout début du programme n'est pas encore bien définie, nous supposons donc que dans un premier temps lors du lancement de l'application, il y a assez d'avions, là où il faut pour les réservations.

- De plus nous supposons que le client débutant une réservation va nécessairement la mener à son terme, c'est-à-dire sans s'arrêter en plein milieu de la réservation.

- Les avions se déplacent le long d'une ligne géodésique et à vitesse constante.

- Nous travaillons avec une base de données où nous avons seulement gardé les grands et moyens aéroports.

- Nous avons les horaires que de certaines compagnies aériennes, nous les regroupons donc en grande banque d'horaire où les horaires sont identiques au sein d'une même banque.

- Le prix du billet sera fixé en partie par les aéroports : plus l'aéroport a un nombre de routes qui lui sont connectées élevé, plus il est intéressant donc plus le prix sera plus élevé, les compagnies lowcost utilisent par exemple des aéroports moins grands puisqu'ils sont moins chers.

- Nous hésitons encore sur le « temps » de l'application : est-ce que nous appliquons une horloge interne différente à l'horloge absolue de la Terre, ou bien tout revient à 0 à l'arrêt de l'application, ou bien encore est-ce que l'on fait marcher l'application en continu. Nous penchons plus vers l'idée d'une horloge interne (c'est-à-dire une échelle de temps propre au programme) à l'application, elle serait la meilleure solution en attendant de trouver une autre façon de penser.

- Nous nous sommes limités à la création d'une vingtaine de types d'avions avec leurs agencements intérieurs.

## La hiérarchie des fonctionnalités

Les fonctionnalités essentielles que permet l'application sont :

- pour la compagnie de :
  - trouver un trajet reliant 2 aéroports
  - gérer les passagers
  - fournir un récapitulatif d'un voyage
  - affecter des avions en choisissant l'avion le plus économique
- pour le client de :
  - faire une réservation
  - gérer sa réservation

Les fonctionnalités secondaires sont :

- pour les compagnies de :
  - gérer les trajets : gérer un plan de vol, afficher le trajet

Les fonctionnalités facultatives que l'on aimerait apporter sont :

- fidéliser son client, en lui attribuant des points (comme des miles)
- faire des promos lors de période de creux (hors vacances par exemple)
- proposer des hôtels avec les destinations choisies pour faciliter les recherches du client et une voiture si besoin
- proposer une assurance en cas d'imprévus
- proposer de la flexibilité au niveau des dates et d'autres options mais cela pourrait engendrer des frais supplémentaires (repas particulier, ..)
- il pourrait être astucieux de montrer le nombre de points que la réservation permet de gagner et de montrer l'itinéraire aller-retour

# III- Diagramme de classe et modélisation

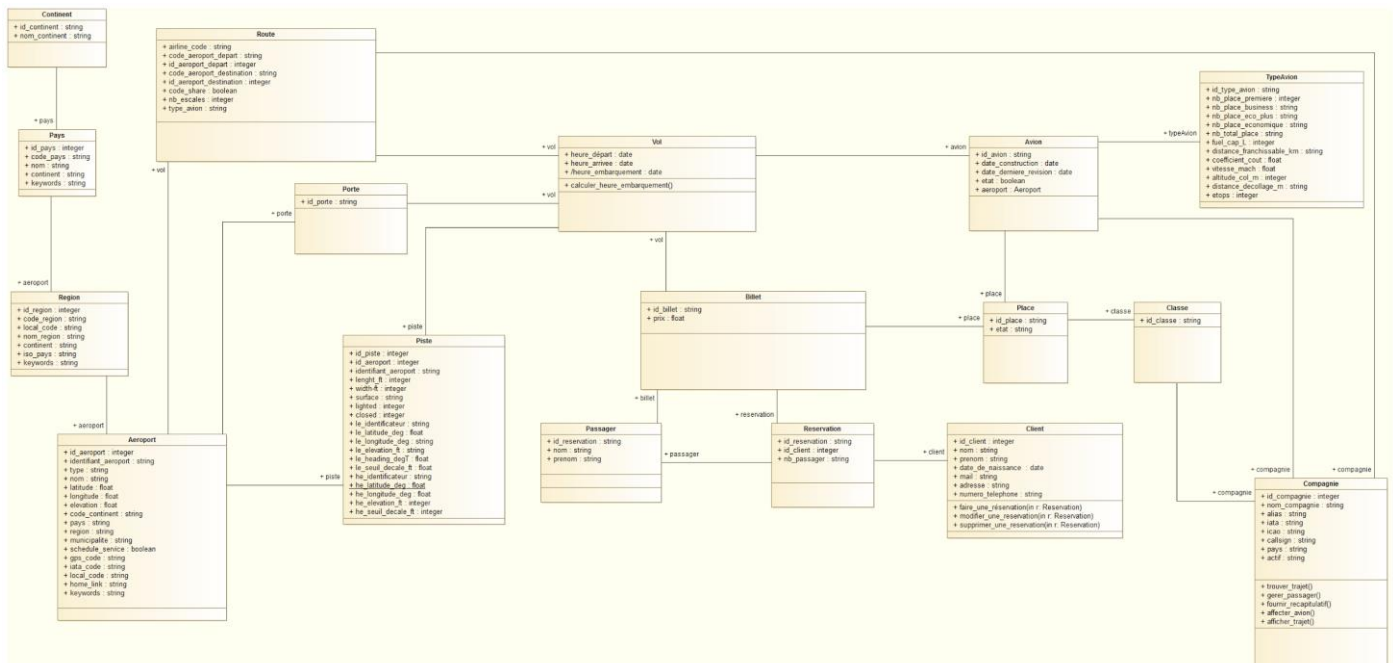


Diagramme de classe version 1

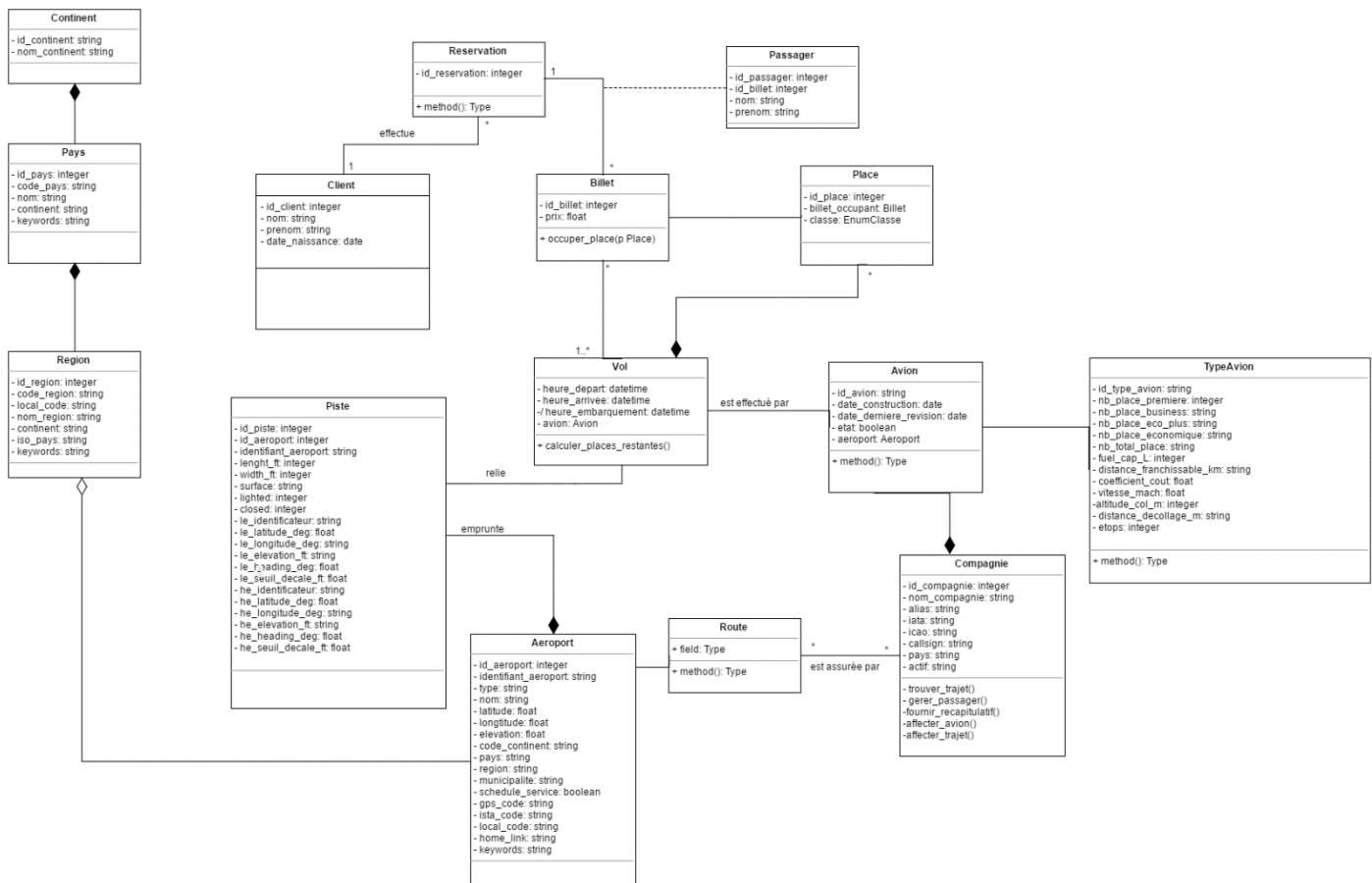


Diagramme de classe version 2

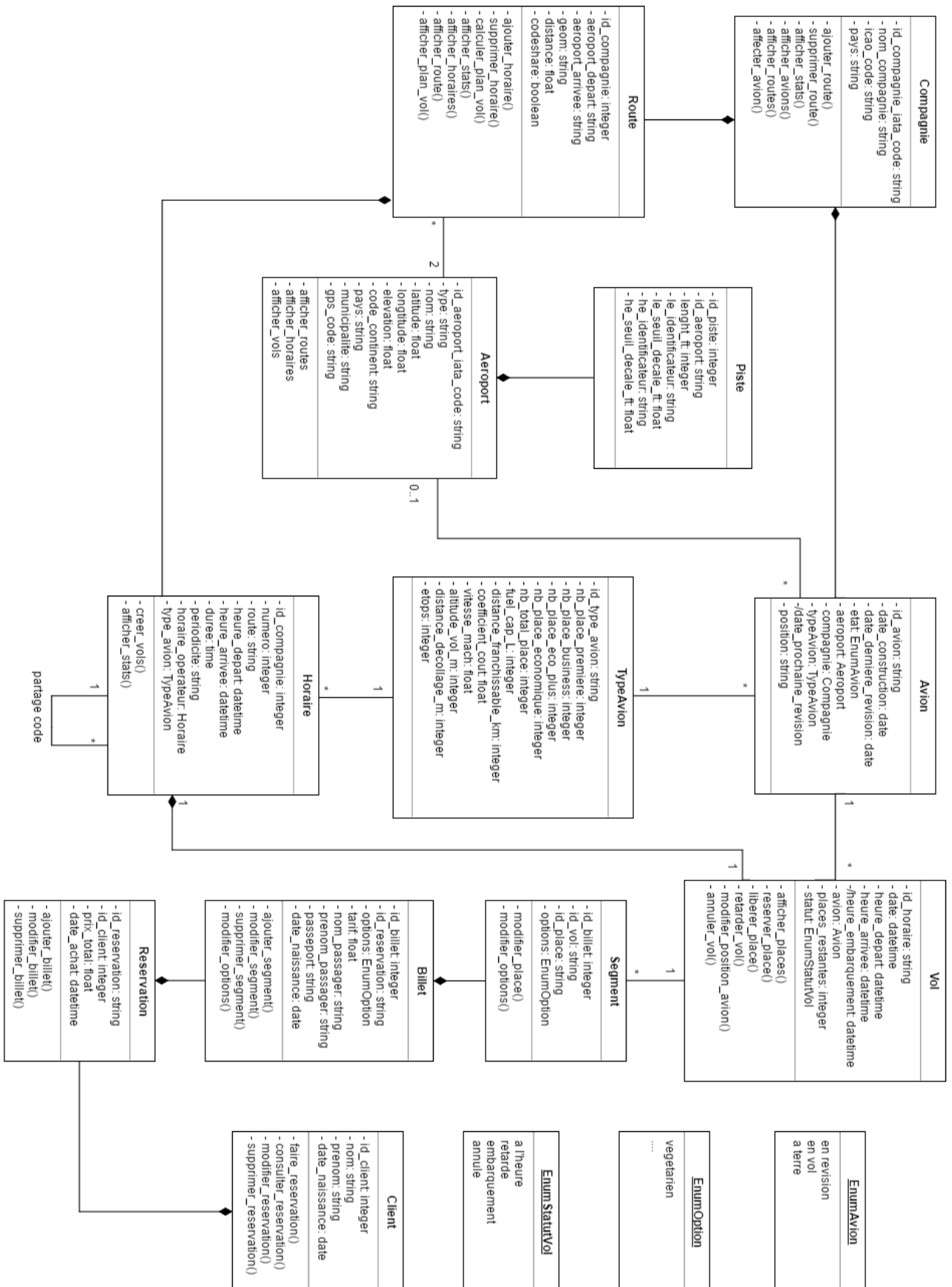


Diagramme de classe version 3 (dernière version)



Au fur et à mesure de notre réflexion et de notre modélisation nous avons changé petit à petit notre diagramme de classe. On peut ainsi voir 3 grandes phases évolutives avec la dernière qui est à terme celle que nous avons gardée. Cependant avec l'avancé de notre programmation elle ne correspondra pas exactement à notre code puisque l'on s'est rendu compte que quelques ajouts étaient encore nécessaires.

L'organisation est la suivante : l'application est divisée en deux pôles, d'un côté la **Compagnie** et de l'autre le **Client**.

Ne dépendant ni de l'un ni de l'autre sont les **Aéroports**, qui ont chacun une ou plusieurs **Pistes** de longueurs différentes.

- La **Compagnie** gère des **Avions**, qui ont une configuration de la cabine particulière (**ConfigAvion**) propre à un type d'avion donné (**TypeAvion**). Un Avion est à un moment donné soit en vol, soit dans un **Aéroport**.

- La Compagnie gère ensuite des **Routes** qui relient deux **Aéroports** (les éventuelles escales techniques sont ignorées).

- Une **Route** se décompose en **Horaires**, qui portent le numéro de vol, l'heure prévue de départ ainsi que celle d'arrivée, et une configuration de cabine donnée (**ConfigAvion**).

Dans le cas d'un partage de code, l'Horaire possède un numéro de vol propre (généralement à 4 chiffres) mais renvoie pour les autres informations vers l'Horaire de la Compagnie qui assure véritablement le **Vol**.

- Un **Horaire** est assuré par des **Vols**, qui sont uniques et ont lieu à une date et à une heure données. Un Vol donné est assuré par un **Avion** précis (celui-ci peut être déterminé à la dernière minute avant le vol). Chaque Vol présente un certain nombre de sièges disponibles en fonction de la configuration (**ConfigAvion**) associée à l'Horaire.

- Un **Client** réalise des **Réservations**, qui concernent l'achat d'autant de **Billets** qu'il y a de voyageurs concernés.

- Chaque **Billet** contient autant de **Segments** qu'il y a de **Vols** réservés. Le Segment contient l'information de la place réservée pour le voyageur dans le Vol précis, ainsi que les diverses options demandées.

- Un **Billet** a un coût particulier, qui est fonction des **Segments** concernés, c'est-à-dire de la classe du siège, du **Vol** lui-même (distance, taxes d'aéroport et de carburant, période...) et des options demandées.

- Une **Réservation** peut ajouter à la somme des coûts des **Billets** quelques frais de traitement.

## IV- Modélisation dynamique

Pour comprendre en quoi consistait l'enchaînement logique d'une réservation faite par un client nous avons fait plusieurs simulations de réservations ce qui a abouti à la stratégie suivante :

- Demander le lieu (aéroport) de départ, demander le lieu (aéroport) d'arrivée
- Demander une date de départ, une date de retour
- Demander le nombre de passagers (adulte/enfant)
- Demander quel type de classe (eco, eco plus, business, première)
  - x heure de départ/ heure de retour
  - x nombre d'escales
  - x compagnie voulue
- Affichage des différents vols possibles avec les différents horaires
- Remplissage des informations des passagers une fois le choix effectué
  - x choisir le siège dans l'avion avec affichage du plan
  - x rentrer un code promo
  - x accepter les conditions
  - x proposer une assurance de voyage/ assurance de flexibilité
- Demander si le passager a des besoins particuliers (voyager avec un animal, mobilité réduite, mineur non accompagné, autre demande ...)
- Demander le paiement
- Envoyer le compte-rendu de la réservation

Les étapes ayant « -x » devant ne sont pas des étapes essentielles à la réservation mais des options qui nous sont venues en tête pour affiner la réservation et correspondre au mieux aux besoins du client et qui seront des fonctionnalités avancées.

## Diagramme de séquence

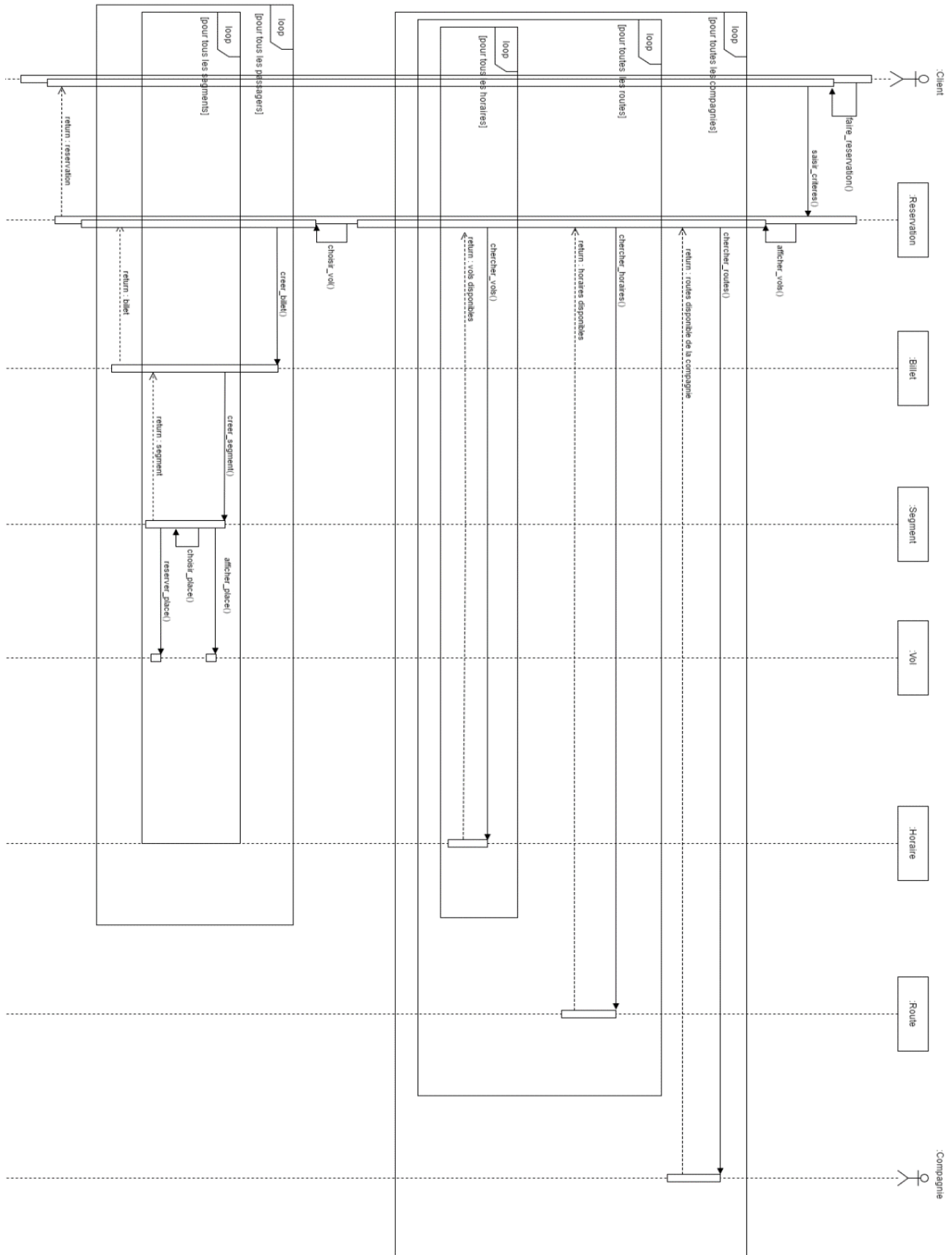


Diagramme de séquence d'une réservation

Ce diagramme montre l'enchaînement des fonctions appelées lors de l'exécution de la fonction "faire\_reservation" par un Client.

- La première action réalisée par cette fonction est de créer un objet Réservation (vide pour l'instant).

- Sur cet objet, elle appelle une fonction saisir\_criteres(), faisant appel à l'interface homme-machine pour récupérer de l'utilisateur les différents critères de recherche de vol.

- Une fois les critères connus, elle appelle sa fonction rechercher\_vols().

Cette fonction va parcourir toutes les compagnies (on les suppose accessibles depuis un objet Reservation).

- Pour chaque compagnie, elle va chercher dans sa liste de routes celles qui permettent de réaliser le chemin demandé, éventuellement avec des changements de vols (escales).

- Pour chaque ensemble de routes ainsi récupéré, elle va chercher les horaires permettant de réaliser le voyage, en faisant attention aux enchaînements de vols s'il y a des escales.

- Pour chaque horaire obtenu de cette manière, elle va interroger les vols aux dates demandées pour retenir ceux qui ont encore des places libres dans la classe demandée.

On considère que des vols disponibles ont été trouvés. Autrement, la fonction demande de revoir les critères ou d'abandonner la recherche.

- L'utilisateur est alors appelé à choisir un vol (ou ensemble de vols) par la fonction choisir\_vol().

- Une fois le choix effectué la fonction se charge de créer un billet par passager, et pour chaque billet, un segment par vol.

- La création d'un segment entraîne le choix et la réservation effective d'une place dans le vol sélectionné.

- Une fois tous les billets complétés, la réservation est prête et s'ajoute à celles du client.

## Diagramme d'activité

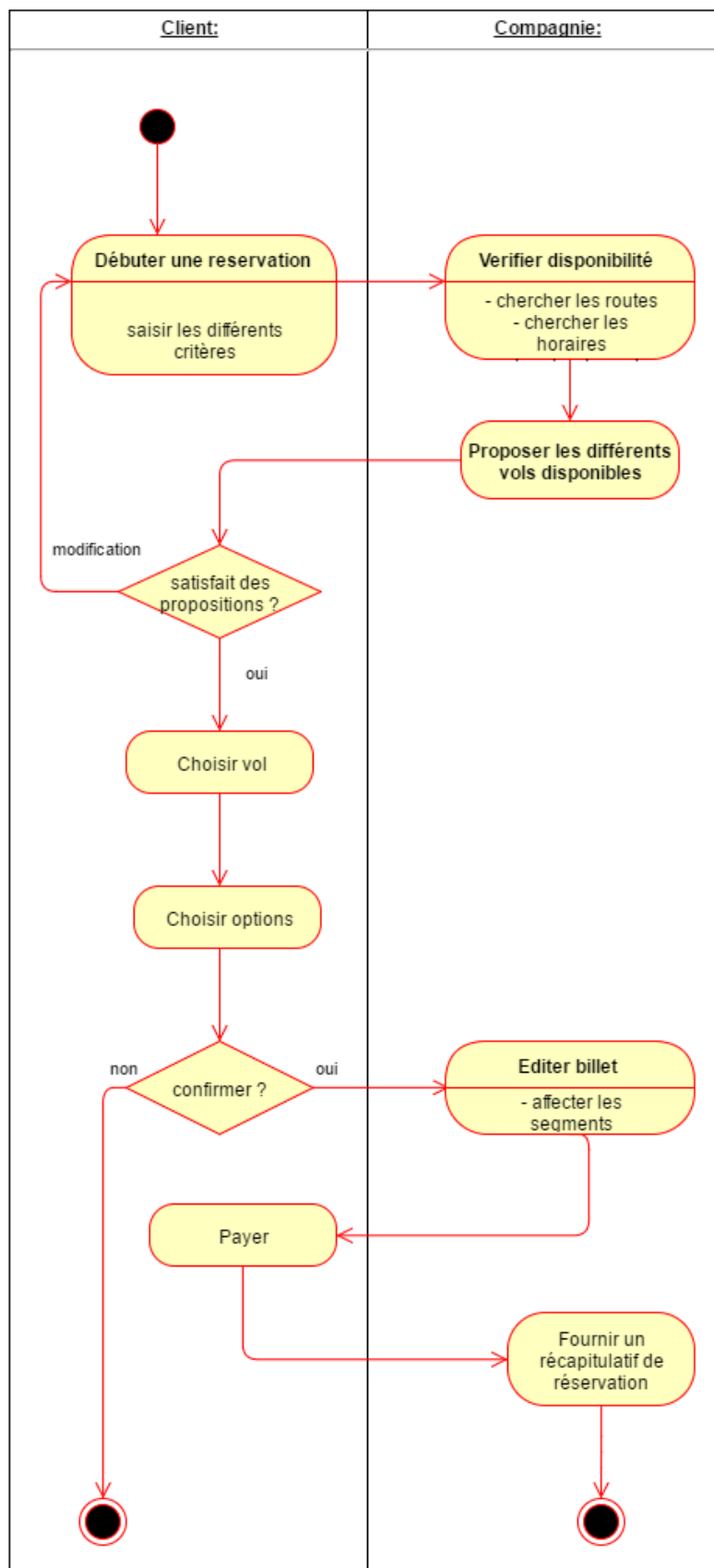


Diagramme d'activité d'une réservation

Le diagramme d'activité correspond à la réservation d'un billet d'avion par un client.

Tout d'abord le client débute la réservation en inscrivant les différents critères qui lui sont demandés. C'est alors au tour de la compagnie aérienne de vérifier les différents vols disponibles pour les critères voulus.

Après vérification, la compagnie propose différents vols disponibles.

Si le client est satisfait par les résultats il continue en choisissant un vol, sinon il recommence depuis le début en changeant les critères de base.

Une fois le vol choisi, il faut choisir des options (être servi normalement, végétarien, besoin d'assistance ...)

Arrive enfin le moment de la confirmation du vol, si le client ne veut pas confirmer, cela mène à la fin de la réservation, s'il confirme la compagnie va éditer son billet en lui affectant les différents segments.

Enfin, le client paye, ce qui amène la compagnie à lui fournir (le plus souvent par mail) un récapitulatif de réservation.

## Diagramme d'état-transition d'un avion

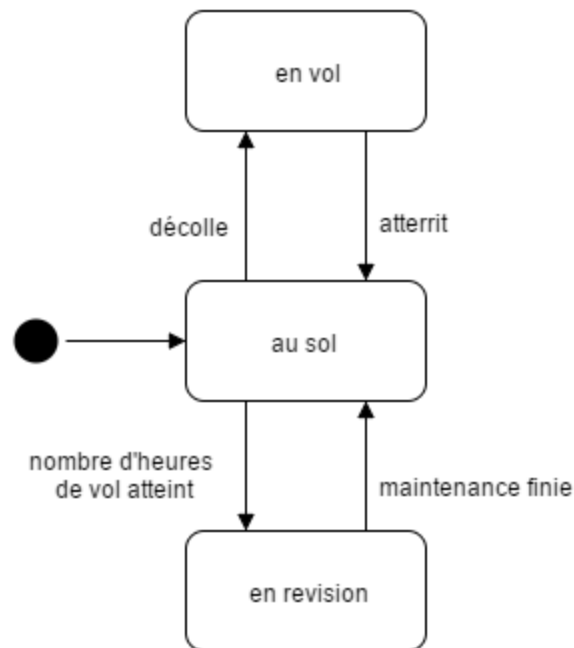


Diagramme d'état-transition d'un avion

Le diagramme d'états-transitions, ici représentant l'état d'un avion, si il est au sol, en vol ou en révision permet de visualiser plus clairement les différents états possibles et surtout quels sont les moyens de passer de l'un à l'autre. Ainsi l'avion ne peut être que dans 3 états : en vol, au sol, ou en maintenance. La majorité du temps l'avion est soit en vol, soit au sol. Ce changement se fait lorsqu'il décolle ou atterrit. Il arrive parfois qu'arrivé à un certain nombre d'heures que l'avion doit avoir une révision, il est alors inactif car il ne peut être utilisé pour un vol. Une fois la maintenance finie, il est de nouveau actif pour réaliser des vols.

## V - CONCLUSION

Cette analyse, même si elle est appelée à évoluer avec le code, a permis de donner à notre programme son organisation principale, centrée sur les deux pôles "Client" et "Compagnie". Les fonctionnalités majeures de réservation et de gestion de réseau et de flotte ont été explicitées, ainsi que les fonctionnalités supplémentaires possibles (programme de fidélité, ventes annexes, gestion des plans de vols...). Enfin, les diagrammes de séquence et d'activité ont permis de détailler la réalisation d'une réservation. Toutes ces informations vont nous permettre de nous guider dans la suite du projet et la réalisation de notre programme.