



FLIGHTS

UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

Compagnie Aérienne Belfortaine

GL52 - P2013

Etudiants:

MAKKI Hala

GHAMRAWI Safwan

GHAZAYEL Jad

LE GUELVOUIT Samuel

Enseignant

M. Abderrafiaa KOUKAM

Responsable

M. Nicolas GAUD



Sommaire

I.	Introduction	4
II.	Description	5
A.	Contexte du sujet.....	5
B.	Diagramme de contexte	5
C.	Domaine étudié	6
D.	Fonctionnement du système	7
E.	Acteurs.....	7
F.	Périmètre du système.....	7
G.	Extensions possibles	8
III.	Dossier de conception	9
A.	Modèle Conceptuel	9
B.	Caractéristiques des utilisateurs.....	11
C.	Contraintes de développement.....	12
D.	Hypothèses de travail	12
IV.	Besoins fonctionnels	13
A.	Diagramme de cas.....	13
E.	Diagramme de séquence	26
F.	Domain model	36
G.	Diagramme de classe.....	40
H.	Diagramme d'état.....	42
V.	Spécification des interfaces	44
A.	Interface matériel/logiciel	44
B.	Interface logiciel/logiciel.....	44
C.	Interface Homme/logiciel	44
VI.	Besoins en performance	47
VII.	Contraintes de développement	47
I.	Fiabilité et tolérance aux fautes	47
II.	Le comportement du système dans des situations anormales (les exceptions critiques)	47
III.	Sécurité	48
IV.	Conclusion	49
V.	Références	50
VI.	Index	50
VII.	Annexes	50

I. Introduction

Buts et destinataires du document

Le projet qu'on est amené à réaliser au sein des UV BD50 et GL52, titré « COMPAGNIE AERIENNE BELFORTAINE », concerne au gros terme la gestion des transactions de réservation de billets avions en ligne par des clients. Ce Projet consiste à réaliser à partir d'un dossier de spécification la conception de la base donnée ainsi que les différentes interfaces nécessaires au bon fonctionnement du site Web.

Il faut noter que pour développer l'application conçue, on part du zéro, en utilisant PL/SQL pour réaliser le site web, et Oracle comme SGDB. La partie conception sera représentée par UML : un langage de modélisation international. Ce dernier modélisera les différents aspects du système et les interactions entre la compagnie, l'administrateur et leurs clients sous forme de diagrammes.

Le présent document a donc pour but de rendre compte des besoins nécessaires à la réalisation de ce projet. Il servira de base à l'équipe informatique en charge du développement mais il permettra également au client de vérifier que ses attentes ont correctement été retranscrites.

Nous commencerons tout d'abord par décrire le contexte du projet, ensuite nous aborderons la partie conception ainsi que la partie réalisation du projet et nous finirons par la présentation de la maquette du site internet.

Définitions - Abréviations

Flotte	Ensemble d'avions.	
Type avion		Airbus A300,
Vol commercial	Les vols prévus, pas de dates, seulement les heures et les jours du vol.	
Vol réel	Et toutes ses informations dans la réalité, avec les dates.	
Classe		Business, economy, first class, ...
Périodes	Périodes de réduction/promotion.	
Transaction	Représentant une opération de paiement ou de remboursement.	

II. Description

Dans ce rapport, nous allons d'abord détailler les spécifications qu'on a respectées dans la conception et la réalisation du site web.

A. Contexte du sujet

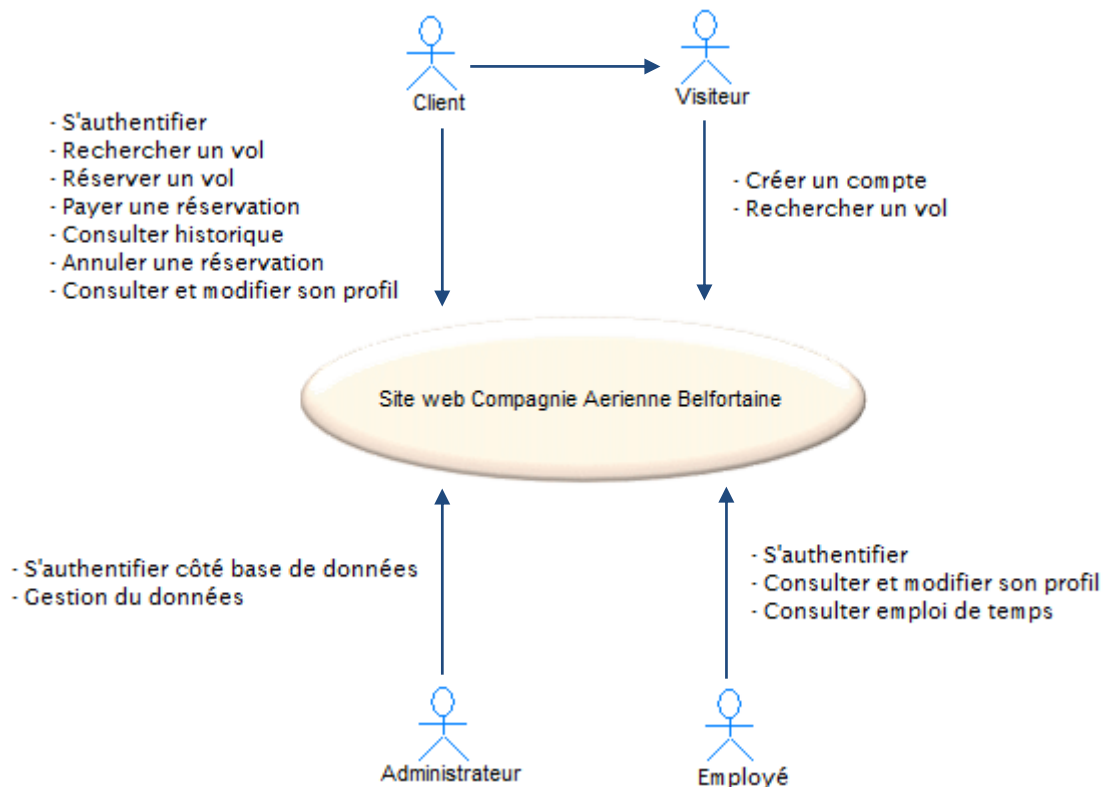
Le sujet que nous avons choisi est le développement du site d'une Compagnie Aérienne. Vu que ce type de site peut comporter de nombreux paramètres, Il n'y a pas qu'un seul but pour permettre aux clients de consulter et réserver les vols en ligne, quel que soit la compagnie :

- Du côté client, c'est considéré comme service rapide, accessible 24/7.
- Du côté compagnie, ça réduit du temps, et du personnel, donc de coût.
- Toutes les compagnies ont des sites web à cette époque, c'est une nécessité pour rivaliser tout concurrent.

B. Diagramme de contexte

Le système interagit avec des éléments externes, les acteurs, avec lesquels il échange d'informations (flux).

Voici le diagramme de contexte décrivant les besoins des utilisateurs.



C. Domaine étudié

L'application va se composer de deux interfaces, la première est celle accessible par les clients et les employés et les internautes.

Elle sert à la navigation sur le site dans le but de :

- Rechercher un vol selon certains critères (destinations, horaires, avec ou sans escale...).
- Donner toute information qui semble être intéressante ou indispensable au client (des vols, avion et équipe personnel, promotion).
- Créer son compte personnel.
- Réserver des billets, paiement, et réception de ces billets par courrier électronique.
- Eventuellement modifier ou annuler sa réservation.

L'administrateur a une autre interface, qui va s'occuper de la partie gestion de données.

Une personne naviguant sur le site est un visiteur simple qu'on nommera « Internaute ». Elle ne sera considérée par la compagnie comme un client qu'une fois un compte est créé ; A partir de là elle aura accès à d'autres outils autre que la recherche des vols, comme la réservation, bénéficier des promotions, éditer son compte...

L'internaute peut alors rechercher des vols, il saisit simplement tous les détails concernant son vol comme : La ville de départ, celle d'arrivée, l'option aller-retour ou aller simple, la date de départ et celle d'arrivée, avec ou sans escale... etc.

Ensuite, si l'internaute trouve son vol visé, il doit créer un compte et saisir ses coordonnées personnelles pour pouvoir y réserver des places, et éventuellement payer contre les billets réservés par l'intermédiaire d'un tiers (Banque, PayPal...).

Une fois inscrit, l'internaute devient un client, et peut alors consulter son historique, l'état de son vol, et peut également modifier ou annuler sa réservation.

Il faut noter qu'on n'a pas implémenté toutes ces fonctionnalités, mais la partie du site qu'on a réalisée respecte bien tout ce qui est conçu dans ce rapport.

D. Fonctionnement du système

Concernant le fonctionnement du système, le site se chargera de gérer les différents utilisateurs ainsi que leurs niveaux d'accès. Les utilisateurs se décomposent en deux groupes possédant des droits différents :

- ✓ **Le visiteur (Internaute) :**
Aucun droit d'accès particulier, il peut seulement rechercher des vols d'après des critères visés.
- ✓ **Le client:**
Il a la possibilité d'effectuer également des recherches multicritères d'un vol. Il pourra éventuellement réserver des vols et en payer.
- ✓ **L'employé :**
Cet utilisateur peut accéder à son compte pour consulter ses horaires de travail.
- ✓ **L'administrateur :**
Par rapport à l'utilisateur, il n'accède pas au site de même façon que les autres utilisateurs, il est un utilisateur coté serveur, il possède tous les privilèges de consultation, et manipulation de la base de données, comme l'insertion, la mise à jour des vols...

E. Acteurs

Dans notre projet, nous avons identifié 2 acteurs principaux :

- ✓ **Visiteur (Internaute):**
Le site web est conçu pour que n'importe quelle personne puisse rechercher un vol.
- ✓ **Client :**
Le client peut réaliser les mêmes actions que l'internaute. Cependant il a aussi accès à des fonctionnalités plus avancées après inscription afin d'avoir la possibilité de réserver un vol, choisir une classe (Business ou Economie), consulter les vols déjà réservés, et consulter des offres promotionnels s'il y en a.

F. Périmètre du système

- ✓ Une flotte contient plusieurs avions, caractérisés par un type.
- ✓ Un avion contient un certain nombre de places pour une classe.
- ✓ Un employé, qui est une personne, a une certaine fonction, et est dédié à travailler dans une flotte, et effectue son travail durant les vols (commerciaux).
- ✓ Un vol réel correspond à un vol commercial, et est effectué par un avion.



- ✓ Un vol commercial concerne le départ d'un aéroport dans une ville, située dans un pays, de même l'arrivée à un autre aéroport, et peut avoir plusieurs escales dans des autres aéroports.
- ✓ Un vol commercial peut comporter une ou plusieurs promotions, tout dépendant d'une période.
- ✓ Un client, qui est une personne, peut réserver plusieurs billets (et la classe correspondante pour chaque billet) pour des passagers dans un vol réel.
- ✓ Les informations sur la transaction d'une réservation sont alors sauvegardées.

G. Extensions possibles

Vu qu'on n'avait pas le temps pour réaliser tout ce qui a été conçu, voilà les extensions possibles :

- Vu qu'on n'avait pas le temps pour intégrer les fonctionnalités des employés dans notre site, fournir à eux plusieurs fonctionnalités est considéré comme extension ; la plus importante fonctionnalité à mettre en œuvre sera la possibilité de consulter leur emploi de temps, à l'aide d'un compte pour chacun.
Il faut noter que cette extension ne nécessite aucun changement dans la conception de notre système.
- Vu qu'on n'a pas mis en œuvre la fonctionnalité « choisir place » pour un vol, on peut après ajouter une sorte de carte pour chaque type d'avion, pour que le client puisse choisir la place qu'il veut pour le vol.
- Permettre au client de choisir pareillement ses plats pour le vol en ligne, ainsi que de préciser combien de poids en bagages (amener des équipements sportifs, des véhicules).
- Implémenter une fonctionnalité pour envoyer automatiquement des mails promotionnels aux clients les plus actifs.

III. Dossier de conception

A. Modèle Conceptuel

Voici les différents sous modèles :

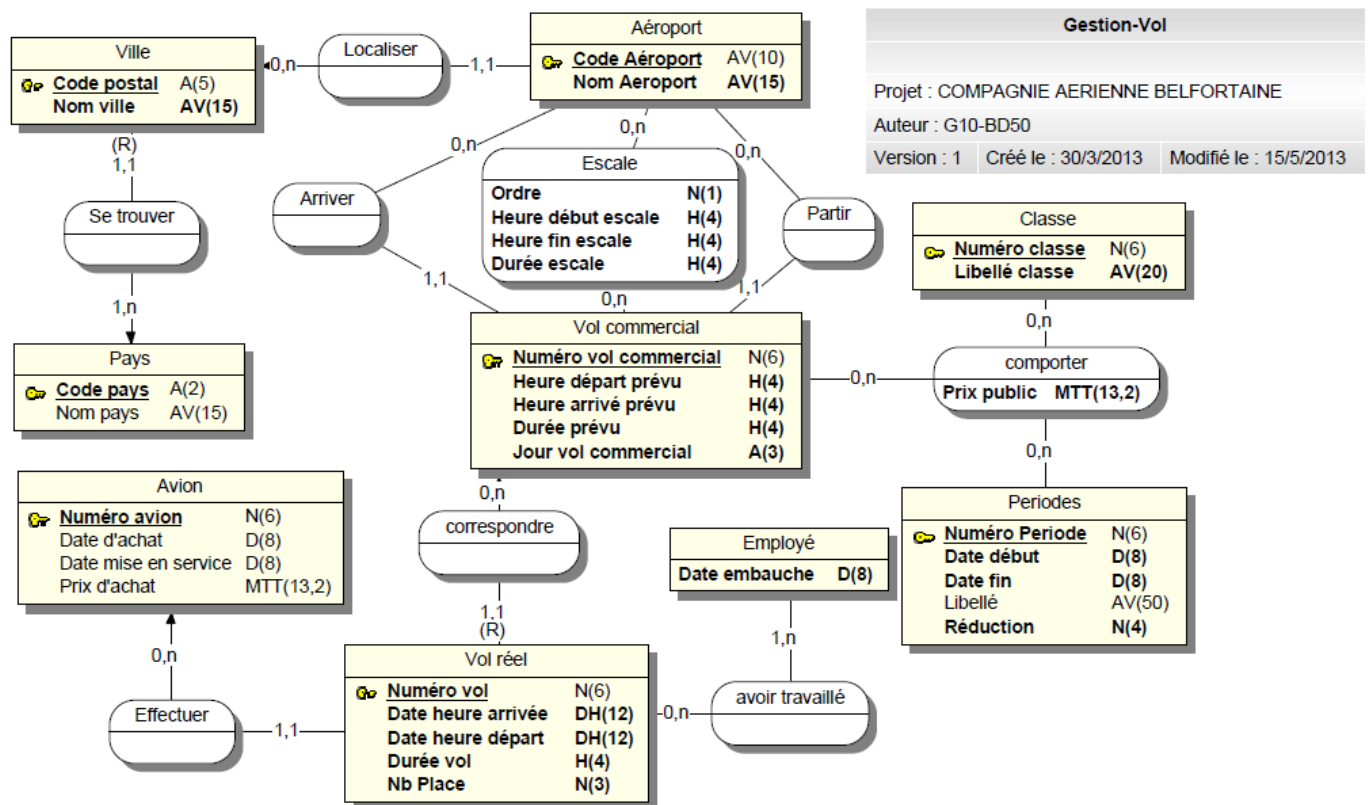


Fig. 1: MCD Gestion Vol

Dans notre premier sous-modèle « Gestion-Vol », nous avons des vols commerciaux ; chacune de ces vols commerciaux a un aéroport de départ, un aéroport d'arrivée et peut avoir plusieurs escales dans différents aéroports (en indiquant les ordres de ces escales). Les vols réels sont des vols commerciaux réalisés à une date précise, en indiquant pour chaque vol la date et l'heure d'arrivée et départ, la durée du vol effectué et le nombre de place des voyageurs. De plus, on des promotions qui seront effectuées pour différentes périodes et différentes classes.

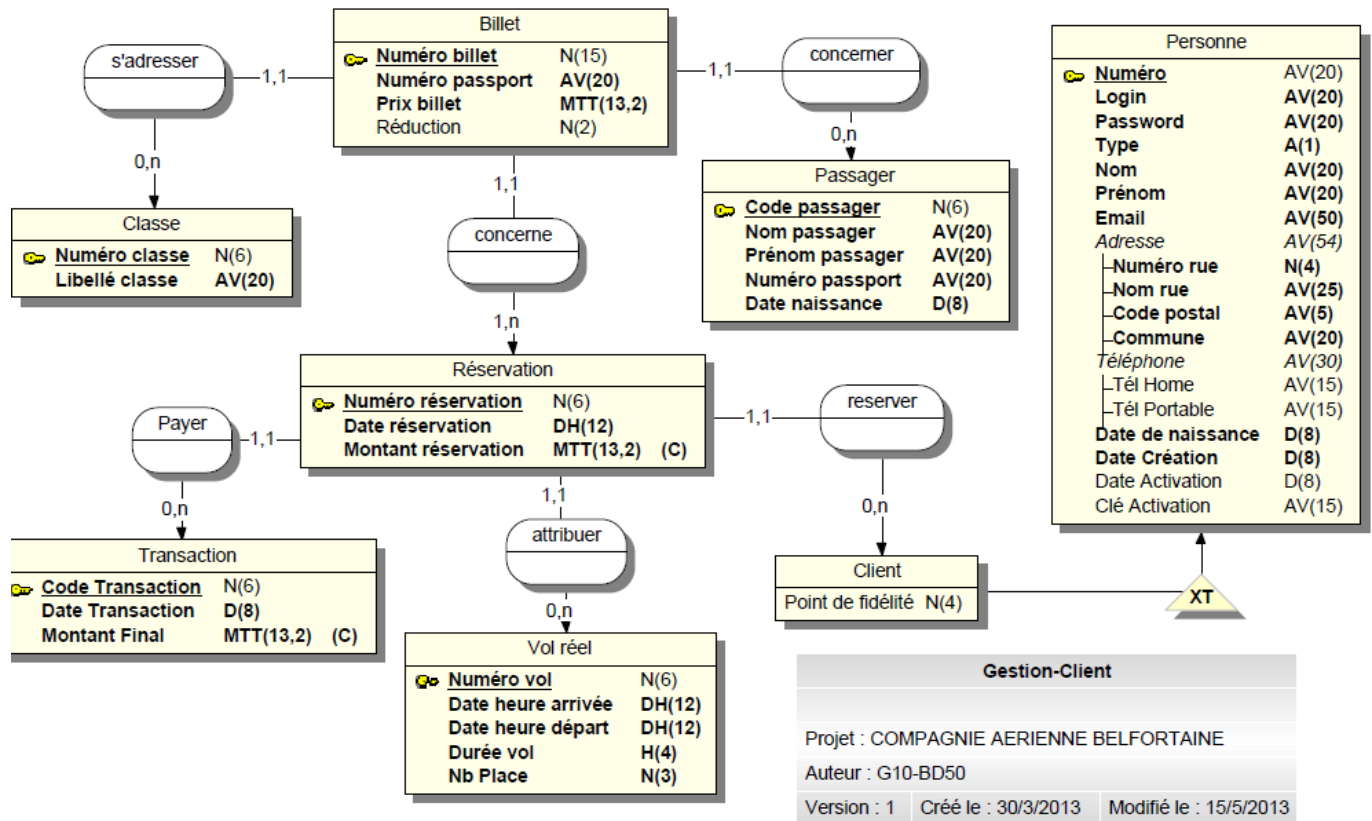


Fig. 2: MCD Gestion Client

Voici notre deuxième sous-modèle « Gestion Client ».

Dans ce modèle on a des clients qui héritent de la classe Personne. Ces clients peuvent effectuer des différentes réservations pour différents passagers. Chaque réservation est composée de différents billets, chacun de ces billets est pour un seul passager. Un passager peut avoir plusieurs Billets pour différentes réservations. Chaque billet est d'un certain type Classe (première classe, économique, business).

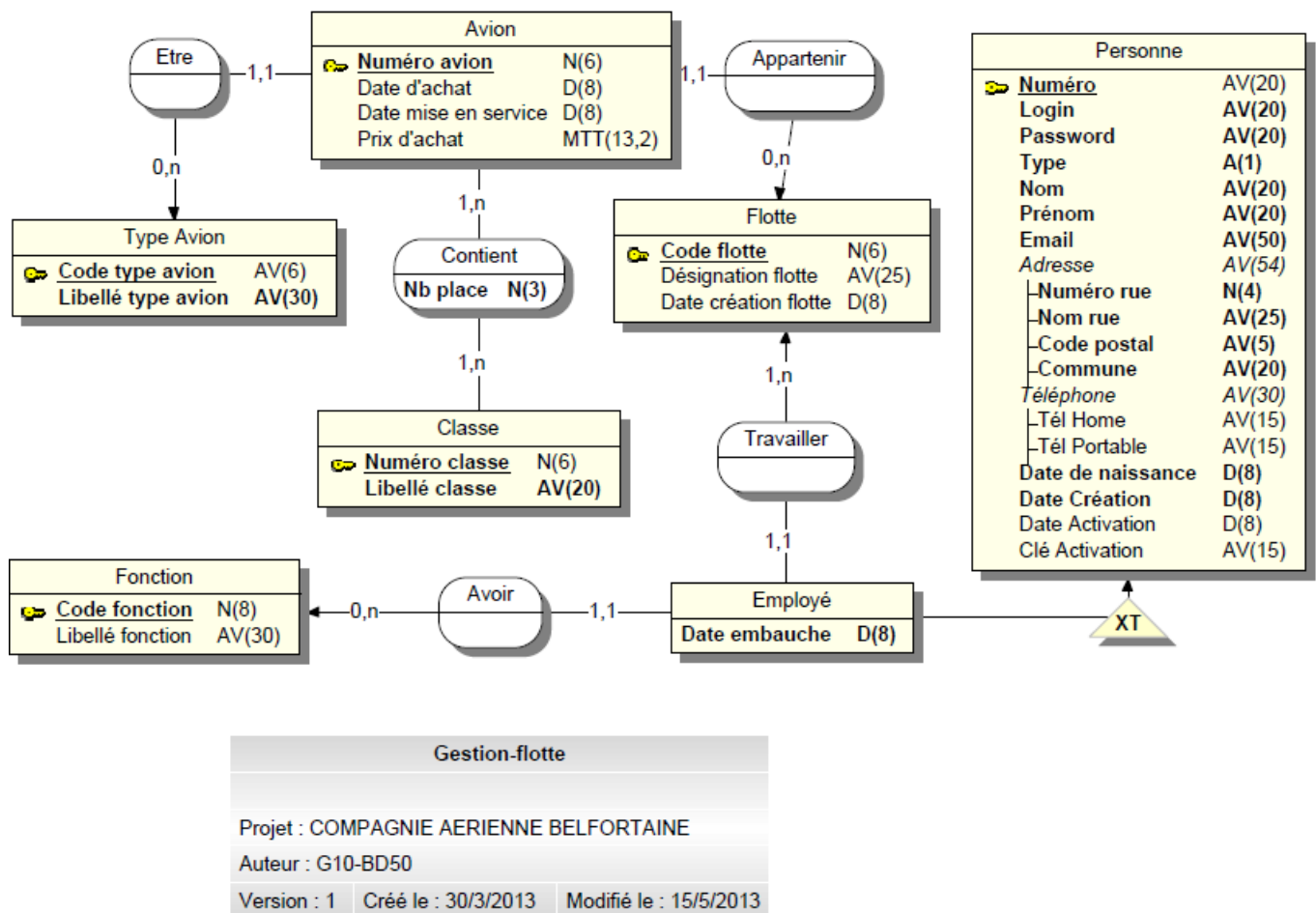


Fig. 3:MCD Gestion Flotte

Dans notre dernier sous-modèle « Gestion Flotte », on a l'entité flotte, composée de plusieurs avions, ainsi plusieurs employés sont consacrées à travailler sur cette flotte. Chacun de ces employés, héritant de l'entité personne, a une fonction dans la flotte.

Les avions sont caractérisés par leurs types, et chaque avion contient un nombre de places pour les différentes classes.

B. Caractéristiques des utilisateurs

Le système interagit avec des éléments externes, les utilisateurs, avec lesquels il échange les flux d'informations.

Nous disposons de 3 différents types d'utilisateurs, chacun possède ses propres interactions avec le system et consultent les données qui sont fournies par un 4^{ème} type d'utilisateurs.

Un premier utilisateur est l'internaute, il possède le moindre nombre de droits de consultation. Son utilisation du système exige le minimum en termes de connaissances informatiques, c'est pourquoi une interface graphique simple et claire est une nécessité pour répondre à ses besoins.



Un deuxième utilisateur est le client, il est l'utilisateur pour lequel on réalise le système; cet utilisateur est en premier temps un internaute, mais afin de réserver un vol, il doit passer par des étapes séquentielles, ces étapes seront guidées pour que ce « future client » ait des informations justes et conformes aux exigences du système.

Cet utilisateur est un utilisateur occasionnel, fidèle à la compagnie. Il consulte le système à chaque fois qu'il a envie de réserver un vol.

Le troisième utilisateur est l'employé, son rôle majeur est de consulter son emploi de temps, il est classifié comme un utilisateur occasionnel en fonction de son rôle qui est la consultation de ses données propres à lui.

Et enfin, le quatrième utilisateur est l'administrateur, son utilisation du système n'exige pas des connaissances avancées dans l'utilisation des systèmes informatiques, même s'il a comme rôle la gestion des données, car l'interface dédiée à lui est aussi simple et claire ; son travail consiste alors à remplir la base de données et d'en faire des mises à jours régulières sans la moindre faute.

C. Contraintes de développement

L'interface web du système sera développée avec le langage PL/SQL en utilisant Oracle Web Toolkit pour la partie utilisateur, et Apex (Oracle Application Express) pour la partie administrateur de base de données. Les pages générées devront respecter les normes (x)HTML et CSS du W3C (World Wide Web Consortium) afin de garantir un affichage optimal sur la majorité des configurations possibles (système d'exploitation et navigateur internet).

Le système sera implanté sous Oracle 11g, un Système de Gestion de Base de Données Relationnelles (SGBDR) fiable. L'Atelier de Génie Logiciel (AGL) utilisé pour la conception de la base de données est Win'Design (version 12), et Astah Professional pour la modélisation des différents diagrammes UML.

D. Hypothèses de travail

D'un point de vue purement technique, un ajout de nouvelles spécifications peut entrainer des problèmes dans la BDD si le système de gestion de base de données n'est pas identique. En effet, certaines taches de développement sont propres à Oracle et le passage sous un SGBD différent nécessiterait de redévelopper certaines parties.

Si de nouveaux besoin venaient à apparaître, les spécifications pourront être amenées à changer. Dans ce cas il sera nécessaire de repenser le système pour satisfaire la demande.

IV. Besoins fonctionnels

A. Diagramme de cas

1. ACTOR DOCUMENTATION TEMPLATE “Personne”

UUID/Itération

AC01

Name

Personne

Stereotype

<<humanbeing>>

Type

Abstract

Form/Responsabilité

(Primary, sender, receiver, UC01 : Logout)

(Primary, sender, receiver, UC02 : Consulter compte)

(Primary, sender, receiver, UC03 : Modifier détails personnels)

Définition/Description

Un acteur abstrait avec les fonctionnalités communes entre les deux acteurs AC02 : Client, et AC03 : Employé.

Relationships

Il est parent de l'acteur AC02 : Client.

Il est parent de l'acteur AC03 : Employé.

2. ACTOR DOCUMENTATION TEMPLATE “Client”

UUID/Itération

AC02

Name

Client

Stereotype

<<humanbeing>>

Type

Concrete

Form/Responsabilité

(Primary, sender, receiver, UC04 : Consulter historique)
(Primary, sender, receiver, UC05 : Annuler réservation)
(Primary, sender, receiver, UC06 : Modifier réservation)
(Primary, sender, receiver, UC07 : Consulter reservation en cours)
(Primary, sender, receiver, UC08 : Désactiver compte)
(Primary, sender, receiver, UC09 : Payer)
(Primary, sender, receiver, UC10 : Résérvé vol)

Définition/Description

Il est l'utilisateur principal du système, il utilise le système pour réserver et payer des places sur des vols, il a besoin de s'authentifier pour faire ceci.

Relationships

Il hérite de l'acteur AC04 : Internaute.
Il hérite de l'acteur AC01 : Personne.

3. ACTOR DOCUMENTATION TEMPLATE “Employé”

UUID/Itération

AC03

Name

Employé

Stereotype

<<humanbeing>>

Type

Concrete

Form/Responsabilité

(Primary, sender, receiver, UC11 : Consulter emploi de temps)

Définition/Description

Il utilise le système pour consulter son emploi de temps sur les vols, après authentification.

Relationships

Il hérite de l'acteur AC01 : Personne.

4. **ACTOR DOCUMENTATION TEMPLATE “Internaute”**

UUID/Itération

AC04

Name

Internaute

Stereotype

<<humanbeing>>

Type

Concrete

Form/Responsabilité

(Primary, sender, receiver, UC12 : Créer un compte)

(Primary, sender, receiver, UC13 : Consulter vol internaute)

(Primary, sender, receiver, UC14 : Rechercher vols)

Définition/Description

Il représente chaque personne qui utilise le système pour rechercher des vols, il n’a pas besoin de s’authentifier, mais c’est un potentiel Client.

Relationships

Il est parent de l’acteur AC02 : Client.

5. **ACTOR DOCUMENTATION TEMPLATE “Banque”**

UUID/Itération

AC05

Name

Banque

Stereotype

<<software>>

Type

Concrete

Form/Responsabilité

(Primary, sender, receiver, UC15 : Payer)

(Primary, sender, receiver, UC16 : Rembourser)

Définition/Description

Acteur externe utilisé pour effectuer les opérations financières.

6. ACTOR DOCUMENTATION TEMPLATE “PayPal”

UUID/Itération

AC06

Name

PayPal

Stereotype

<<software>>

Type

Concrete

Form/Responsabilité

(Primary, sender, receiver, UC15 : Payer)

(Primary, sender, receiver, UC16 : Rembourser)

Définition/Description

Acteur externe utilisé pour effectuer les opérations financières.

7. ACTOR DOCUMENTATION TEMPLATE “Administrateur”

UUID/Itération

AC07

Name

Administrateur

Stereotype

<<humanbeing>>

Type

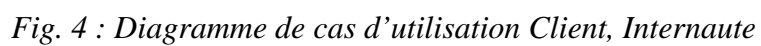
Concrete

Form/Responsabilité

(Primary, sender, receiver, UC01 : Logout)
(Primary, sender, receiver, UC17 : Accéder Vols commerciaux)
(Primary, sender, receiver, UC18 : Accéder Vols réels)
(Primary, sender, receiver, UC19 : Accéder Escales)
(Primary, sender, receiver, UC20 : Accéder Périodes)
(Primary, sender, receiver, UC21 : Accéder Villes)
(Primary, sender, receiver, UC22 : Accéder Aéroports)
(Primary, sender, receiver, UC23 : Accéder Flottes)
(Primary, sender, receiver, UC24 : Accéder Avions)
(Primary, sender, receiver, UC25 : Accéder Types d'avions)
(Primary, sender, receiver, UC26 : Accéder Fonctions)
(Primary, sender, receiver, UC27 : Accéder Classes)
(Primary, sender, receiver, UC28 : Accéder Employés)
(Primary, sender, receiver, UC29 : Consulter Clients)
(Primary, sender, receiver, UC30 : Consulter Réservations)
(Primary, sender, receiver, UC31 : Consulter Transactions)
(Primary, sender, receiver, UC32 : Consulter Billets)
(Primary, sender, receiver, UC33 : Consulter Passagers)

Définition/Description

Responsable de la base de données, tout ce qui concerne l'ajout et la suppression des informations nécessaires au bon fonctionnement du système.



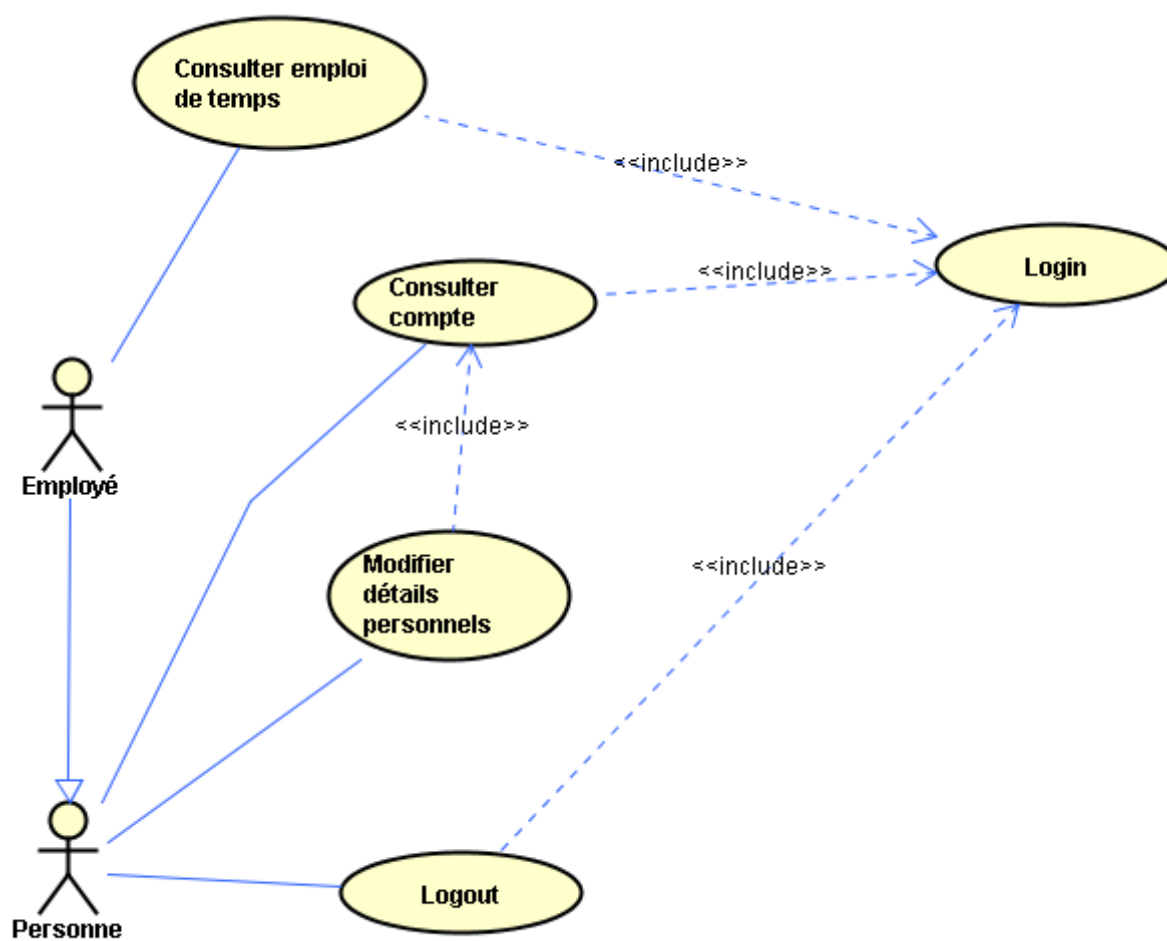


Fig. 5 : Diagramme de cas d'utilisation Employé

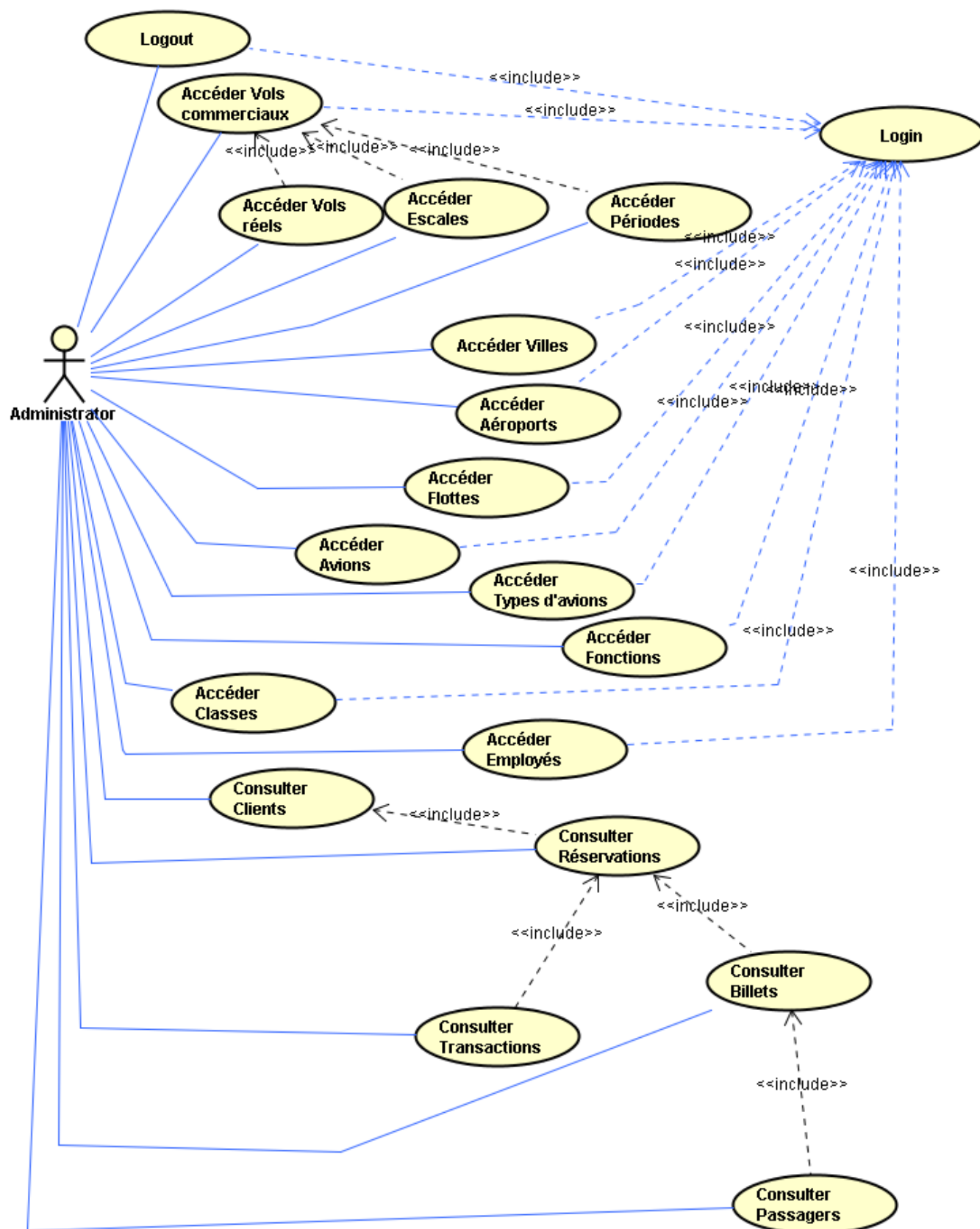


Fig. 6 : Diagramme de cas d'utilisation Administrateur

1. TEMPLATE FOR DOCUMENTING USE CASES “Reserver vol”

UUID/Itération

UC10

Name

Réserver vol

Type

Concrete

Description

Opération de réservation d’un billet.

Actors

AC03 : Client

Conditions

Pre : Client est authentifié
Places disponibles sur le vol
Vol est choisi

Post : Enregistrement de la réservation dans la base de données
Décrémenter des places disponibles sur le vol
Enregistrement billets dans la base de données
Envoie billets par mail au client

Flow

1. Client appuie sur Réserver.
2. Le système affiche résumé vol choisi (dates et heures, aéroports).
3. Client saisit NbBillets.
4. Le système affiche un formulaire pour que le client puisse remplir les coordonnées des passagers.
5. Le client saisit ainsi leurs coordonnées (nom, prénom, passport, date naissance).
6. Il saisit aussi le type de classe pour chaque billet (économique, business, ...).
7. Le système choisit les places sur le vol par défaut.
- 8.
9. a. Client appuie sur Valider.
b.
10. Le système enregistre la réservation dans la base de données.
11. Le système décrémenter le nombre de places disponibles sur le vol concerné.
- 12.
13. le système enregistre les passagers dans la base de données.
14. Le système enregistre les billets dans la base de données.
15. Le système envoie les billets par mail au client.
16. Le système affiche « Réservation OK »

Exception/Variation Flow

8. Le client choisit une place pour chaque billet.
9. b. 16. Client appuie sur Annuler.
17. le système affiche la page d’accueil.
12. Le système calcule la réduction potentielle selon les points de fidélité du client.

2. TEMPLATE FOR DOCUMENTING USE CASES “Payer”

UUID/Itération

UC09

Name

Payer

Type

Concrete

Description

Paielement d’une réservation.

Actors

AC03 Client
AC05 Banque
AC06 PayPal

Conditions

Pre : Client est authentifié
Réservation effectuée

Post : paiement est effectué
Ajout transaction dans la base de données
Incrémentation point de fidélité du client

Flow

1. Client appuie sur PayerCommande.
2. Le système affiche le résumé de l’opération à exécuter (montant à payer est affiché).
3. Le système affiche les différentes méthodes de paiement possibles (carte de crédit et PayPal dans notre exemple).
4. a. Le client choisit carte de crédit comme méthode de paiement.
b.
5. a. Le client valide.
b.
6. Le système affiche un formulaire pour que le client saisisse ses coordonnées bancaires (numéro carte crédit, code sécurité, date expiration, date naissance, nom personne).
7. Le client saisir ses coordonnées.
8. Le système envoie les informations nécessaires pour effectuer l’opération à la banque.
9. La banque renvoie un message résultat d’opération au système.
10. Le système affiche alors ce message au client.
- 11.
- 12.
- 13.
- 14.
- 15.
- 16.
- 17.
- 18.
- 19.
20. Le système enregistre la transaction dans la base de données.
21. Il associe cette transaction à la réservation correspondante.
22. Le système change l’état paiement de la transaction.
23. Il incrémente les points de fidélité du client.



Exception/Variation Flow

- 4. b. Le client choisit PayPal comme méthode de paiement.
 - 11. Le système se connecte avec PayPal en envoyant les infos compte entreprise et le montant et le type d'opération.
 - 12. PayPal affiche la page saisir coordonnées.
 - 13. Le client saisit ses coordonnées (login et mot de passe).
 - 14. PayPal affiche le résumé d'opération à effectuer.
 - 15. Le client valide.
 - 16. PayPal se connecte avec la banque.
 - 17. La banque renvoie le résultat d'opération à PayPal.
 - 18. PayPal renvoie le message au système.
 - 19. Ce dernier l'affiche au client.
- 5. b.
 - 23. Le client annule.
 - 24. Le système affiche « Paiement annulé ».

3. TEMPLATE FOR DOCUMENTING USE CASES “Annuler réservation”

UUID/Itération

UC05

Name

Annuler réservation

Type

Concrete

Description

Païement d’une réservation.

Actors

AC03 Client

Conditions

Pre : Client est authentifié
 Réservation existe et est sélectionnée

Post : suppression de la réservation de la base de données
 Incréméntation des places disponibles sur le vol concerné

Flow

1. Client appuie sur « annuler réservation ».
2. Le système affiche un message « êtes-vous sur ? ».
3. a. Client appuie sur « Oui ».
 b.
4. Le système supprime les billets de la réservation de la base de données.
5. Il marque la réservation comme annulée.
6. Il incrémente les places disponibles sur le vol concerné.
7. Il affiche un message « annulation effectuée »

Exception/Variation Flow

3. b. 8. Client annule.
9. le système affiche la page d’accueil.

4. TEMPLATE FOR DOCUMENTING USE CASES “Rembourser”

UUID/Itération

UC16

Name

Rembourser

Type

Concrete

Description

Remboursement d’une réservation annulée.

Actors

AC03 Client
AC05 Banque
AC06 PayPal

Conditions

Pre : Client est authentifié
 Réservation est annulée
 Paiement de la réservation annulée est effectué

Post : remboursement est effectué
 Ajout transaction dans la base de données
 Décrémentation point de fidélité du client

Flow

1. a. Le système envoie les informations nécessaires pour effectuer l’opération à la banque.
 b.
2. La banque renvoie un message résultat d’opération au système.
3. Le système affiche alors ce message au client.
- 4.
- 5.
- 6.
- 7.
- 8.
9. Le système modifie les attributs de la transaction correspondante à la réservation annulée.
10. Il décrémente les points de fidélité du client.

Exception/Variation Flow

1. b.
 4. Le système se connecte avec PayPal en envoyant les infos compte entreprise et le montant et le type d’opération.
 5. PayPal se connecte avec la banque.
 6. La banque renvoie le résultat d’opération à PayPal.
 7. PayPal renvoie le message au système.
 8. ce dernier l’affiche au client.



E. Diagramme de séquence

1. Diagrammes de séquences niveau système

D'après ce diagramme, on peut comprendre explicitement la relation entre le client et le système de réservation.

Plusieurs contraintes entre dans la réservation afin de réserver un ou plusieurs vols comme le choix de classe, les coordonnées des passagers, et les points de fidélité, Etc...

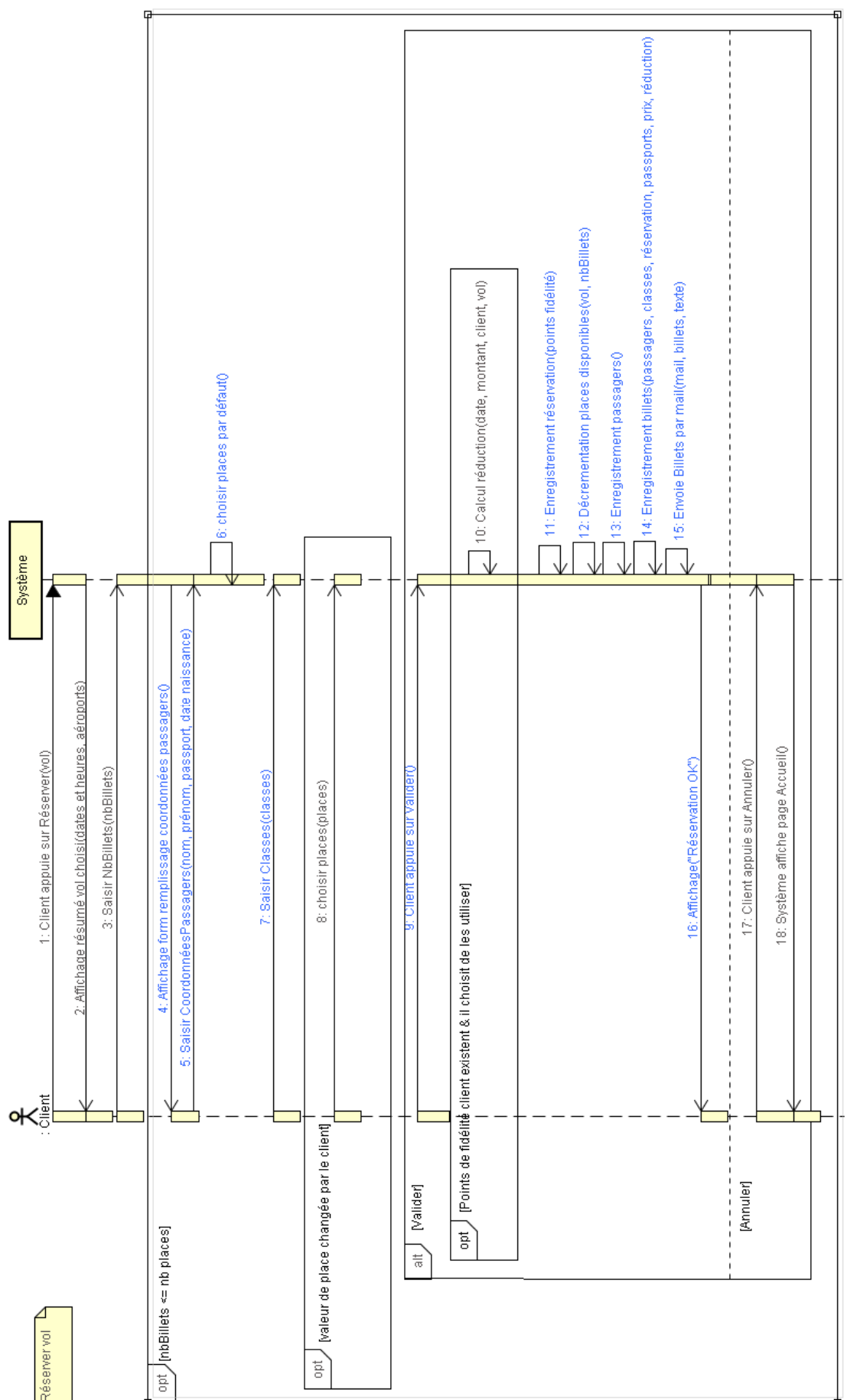


Fig. 7 : Diagramme de séquence niveau système Réserver vol



Le paiement se fait soit par paypal ou carte bancaire. Le client tout d'abord le client choisit une méthode pour le paiement d'une réservation.

Pour payer par carte bancaire, le système affiche une forme pour insérer les données de sa carte bancaire puis le système vérifie auprès de la banque si les données sont bonnes. Puis le système affiche le résultat de l'opération.

Pour payer par paypal, de même le client saisit les données du compte paypal pour vérifier si tout est bon. Une fois les données insérées, on envoie les données vers le site paypal pour la vérification. Après le système affiche le résultat de l'opération.

Une fois paiement fait, on enregistre les données de la transaction (la date, montant). Et mise à jour l'état paiement à PAYEE.

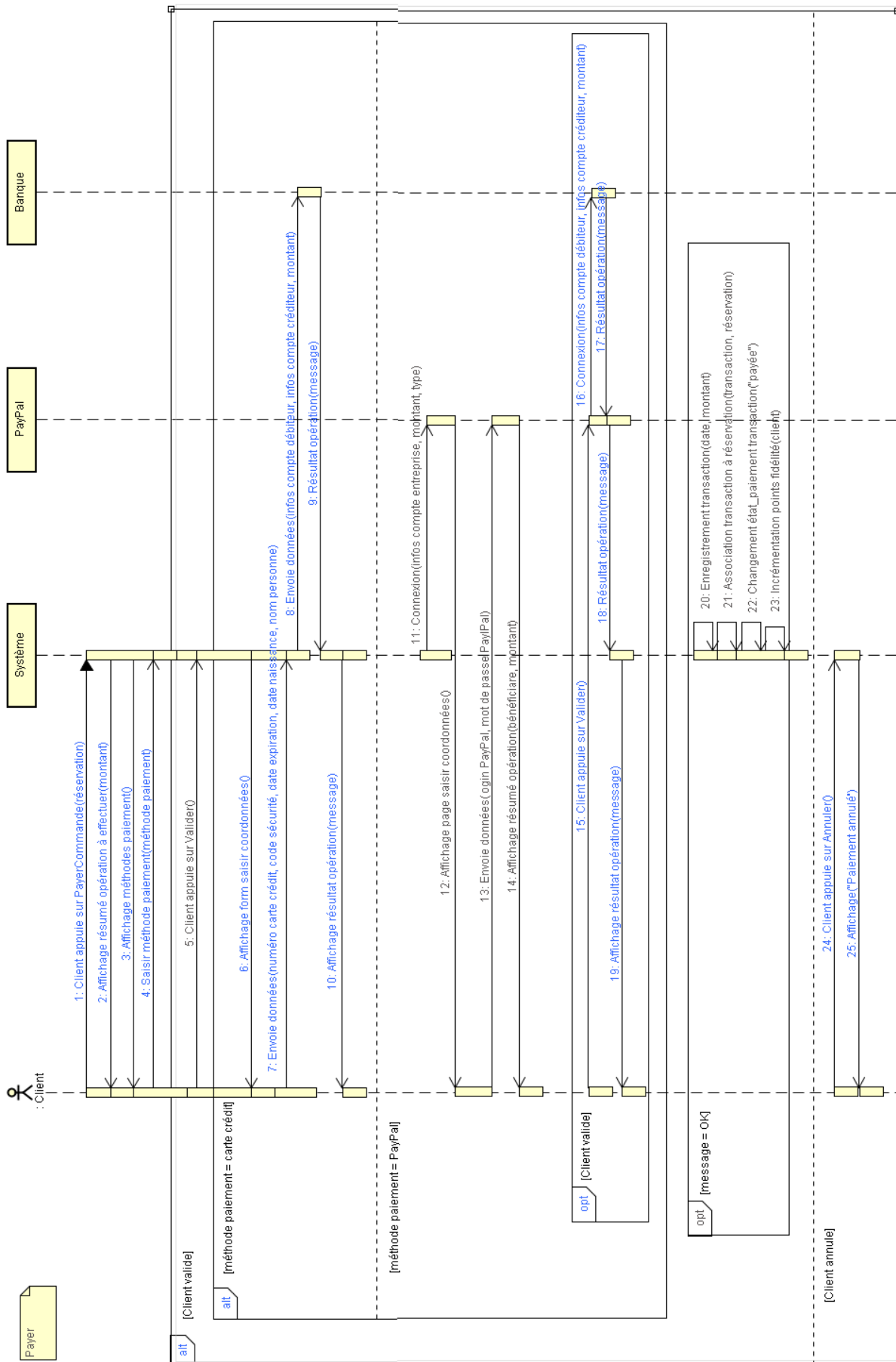


Fig. 8 : Diagramme de séquence niveau système Payer



Le client peut annuler une réservation tant qu'il reste au moins 15 jours pour le vol. Dès que le client appuis sur le bouton annuler, un message s'affiche en demandant s'il est sûr qu'il va annuler la réservation. S'il appuie sur le bouton valider pour annuler la commande et on supprime ou modifier les données. La deuxième option s'il appuie le bouton annuler. Le système affiche la page accueil du site.

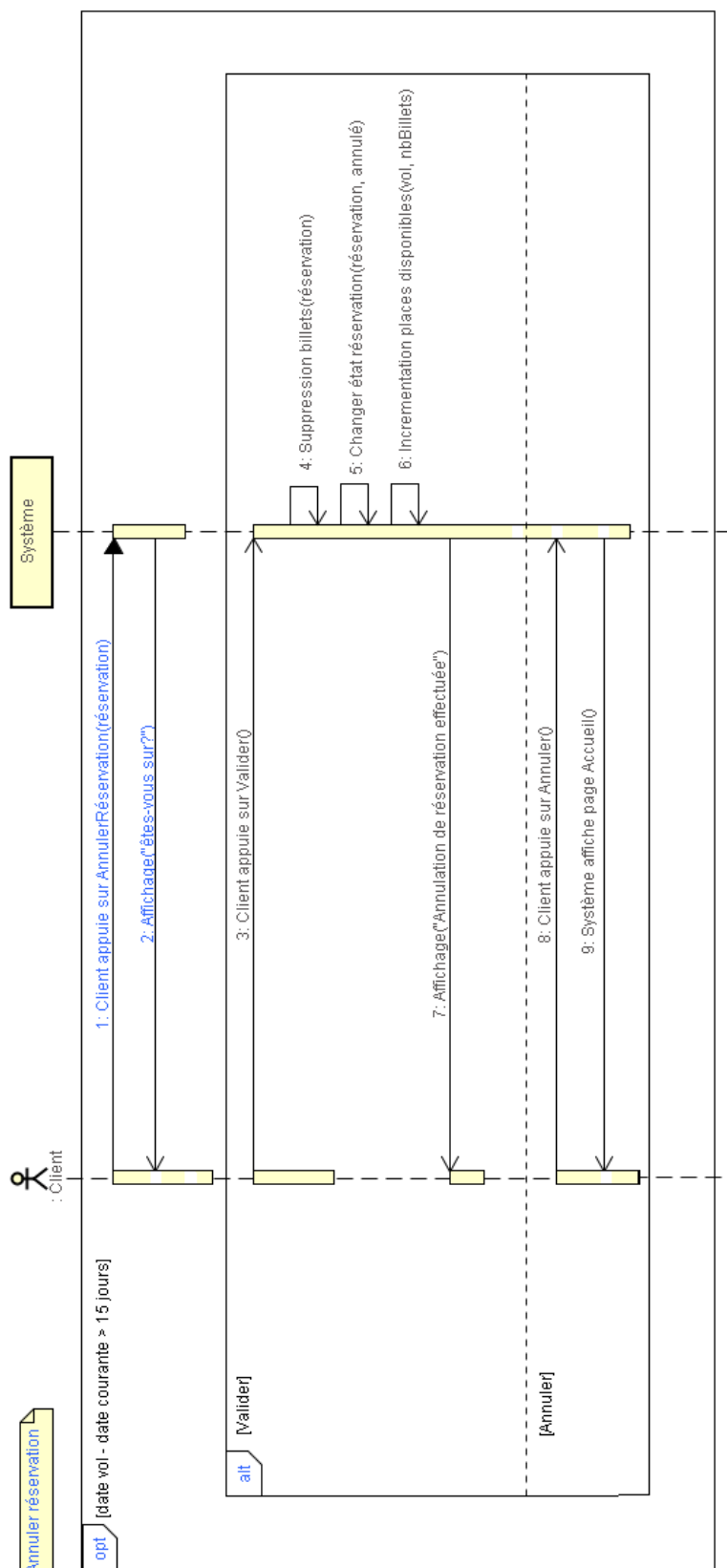


Fig. 9 : Diagramme de séquence niveau système Annuler réservation

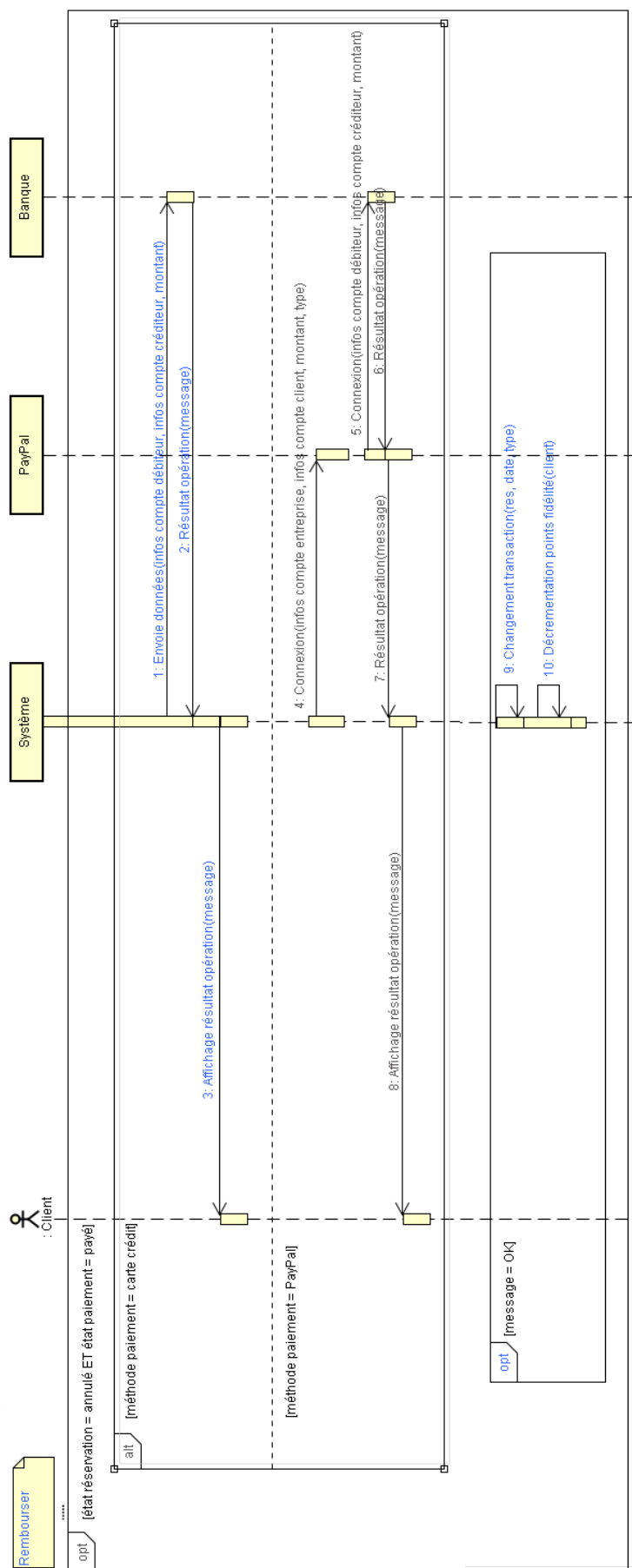


Fig. 10 : Diagramme de séquence niveau système Rembourser



2. Diagrammes de séquences niveau objet

La réservation d'un vol est la principale tâche de notre système, elle entraîne également une interaction avec les différentes parties de notre SGBD, ce diagramme décrit la succession des procédures afin qu'un client qui a déjà créé un compte sur notre site puisse réserver son vol visé.

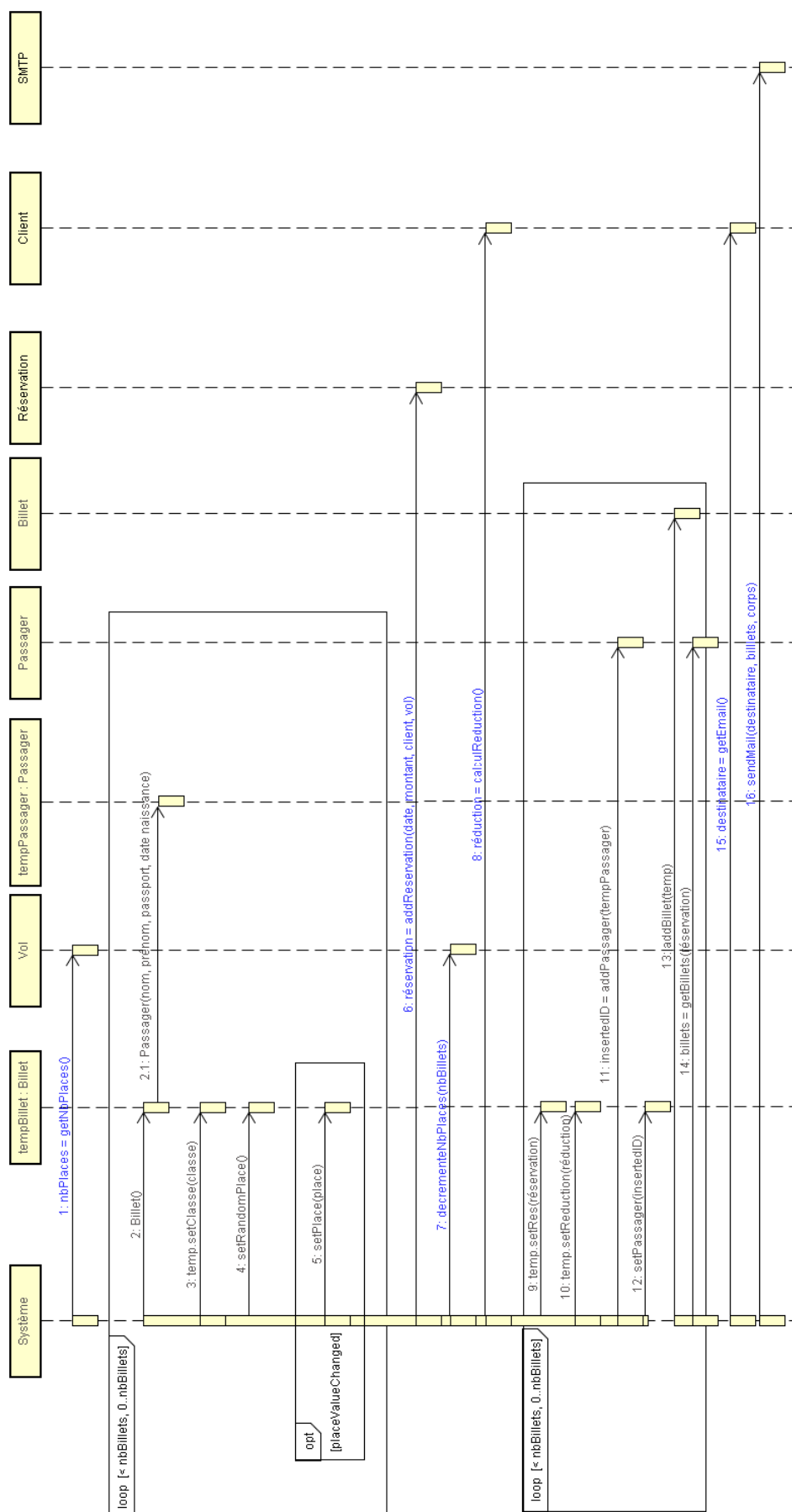


Fig. 11 : Diagramme de séquence niveau objet Réserver vol

Ce diagramme indique la confirmation du paiement par le système et éventuellement incrémenter les points de fidélité du client.

Une fois le paiement est fait, le système doit insérer et modifier des données dans la base de données. Tout d'abord, le système ajout une ligne dans la table transaction (le montant et date). Ensuite, on fait l'affectation de la transaction à la réservation convenable. Puis la mise à jour de l'état paiement. Et comme le client a des points de fidélité on ajout des points au client.

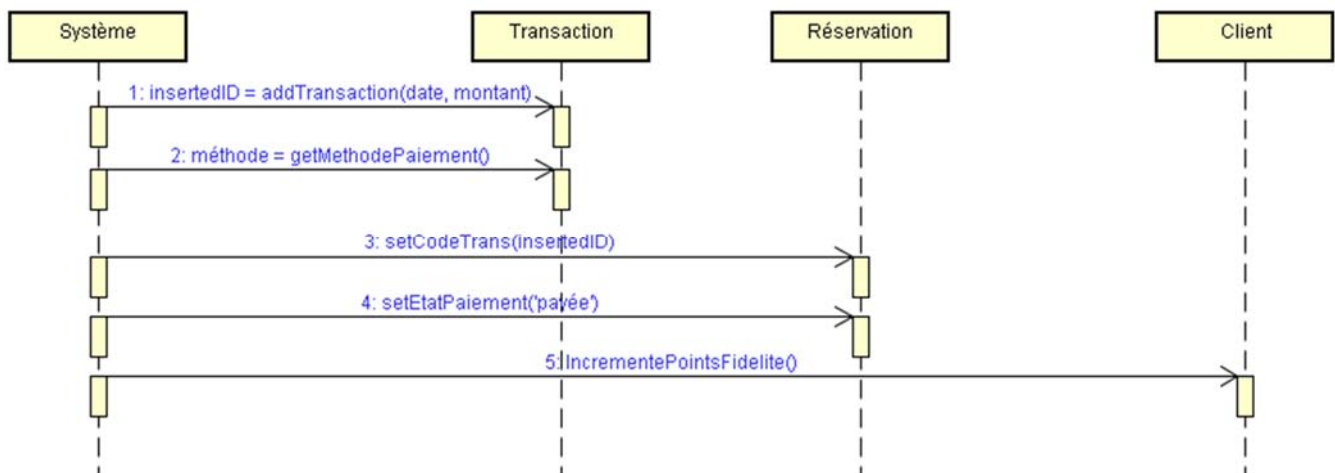


Fig. 12 : Diagramme de séquence niveau objet Payer

Ce diagramme explicite la démarche de réservation avec les tiers qui sont la banque et le système de paiement « Paypal », et comment se fait le péage du ticket.

Le fonctionnement de l'annulation d'une réservation au niveau objet. Le système demande la date du vol pour vérifier s'il reste plus de 15 jours pour le vol. Puis le système supprime les billets de cette réservation. Ensuite on fait la mise à jour de l'état de la transaction. Et finalement on incrémente le nombre de place qui ont été annulés.

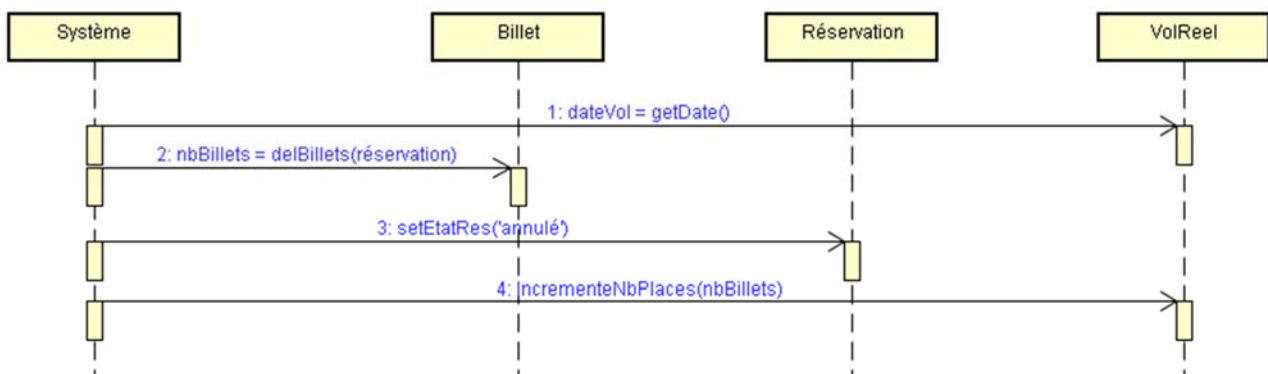


Fig. 13 : Diagramme de séquence niveau objet Annuler réservation

Suite à une raison quelconque, un client peut annuler sa réservation, et notre système peut interagir avec les différents besoins de notre clients, ce diagramme montre la démarche afin de rembourser l'argent au client qui a annulé sa réservation.

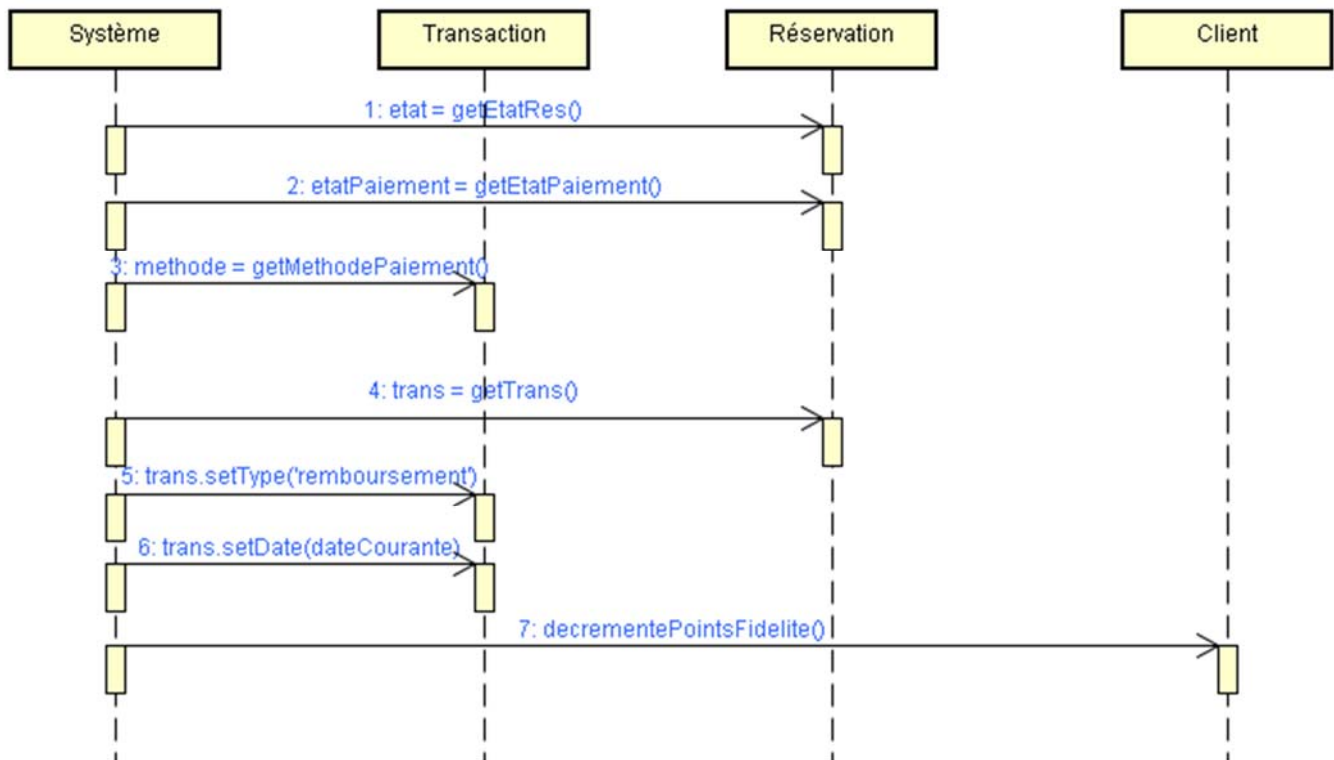


Fig. 14: Diagramme de séquence niveau objet Rembourser

F. Domain model

Un modèle de domaine dans la résolution de problèmes et le génie logiciel est un modèle conceptuel de tous les sujets liés à un problème spécifique. Il décrit les différentes entités, leurs attributs, les rôles et les relations, ainsi que les contraintes qui régissent le domaine du problème.

Dans notre modèle de domaine on trouve qu'une réservation est composée de plusieurs billets. Chacun de ces billets concerne un seul passager. Ces réservations seront effectuées par les clients du site. Et une réservation sera attribuée à un seul vol réel.

Une flotte est composée de plusieurs employés et avion. L'employé hérite de la table personne.

Les vols commerciaux sont composés des vols réels. Chaque vol réel sera effectué par un avion. Chaque vol commercial a obligatoirement un aéroport de départ et d'arriver et il peut faire un ou plusieurs escales dans différentes aéroport. L'aéroport est localisé dans une ville.

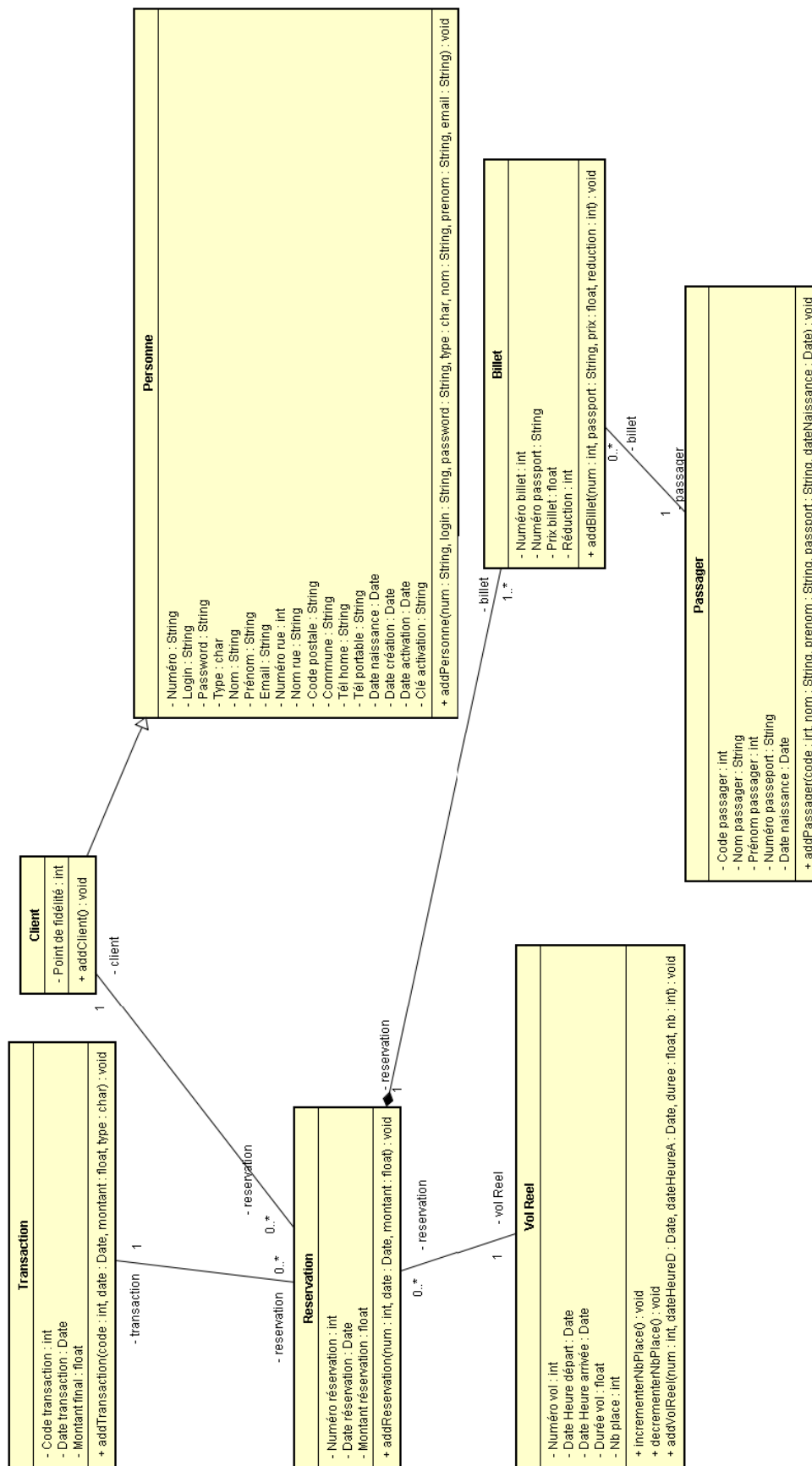


Fig. 15 : Domain model Client

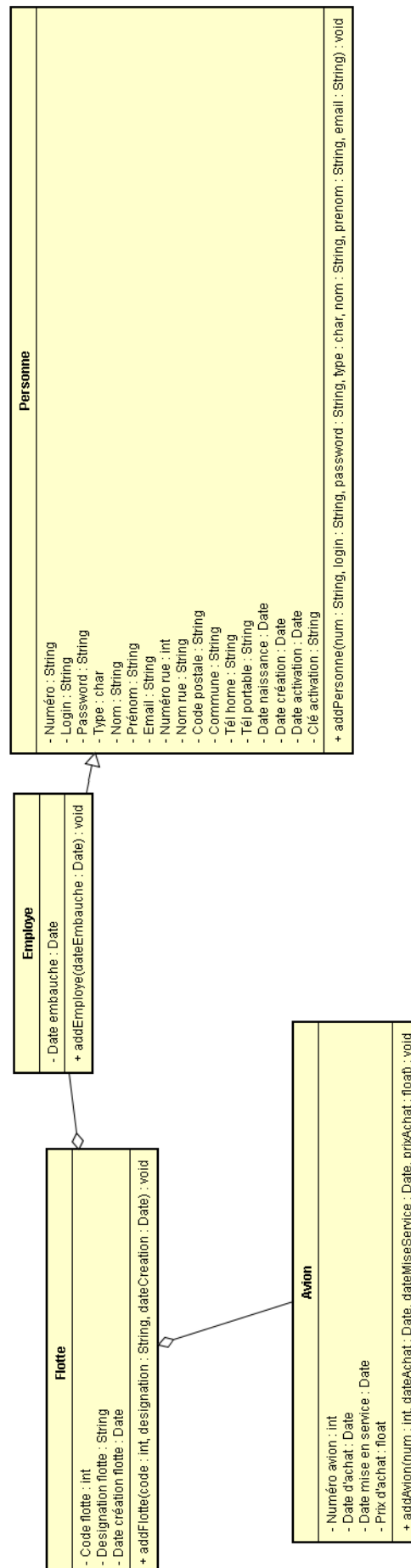


Fig. 16 : Domain model Flotte

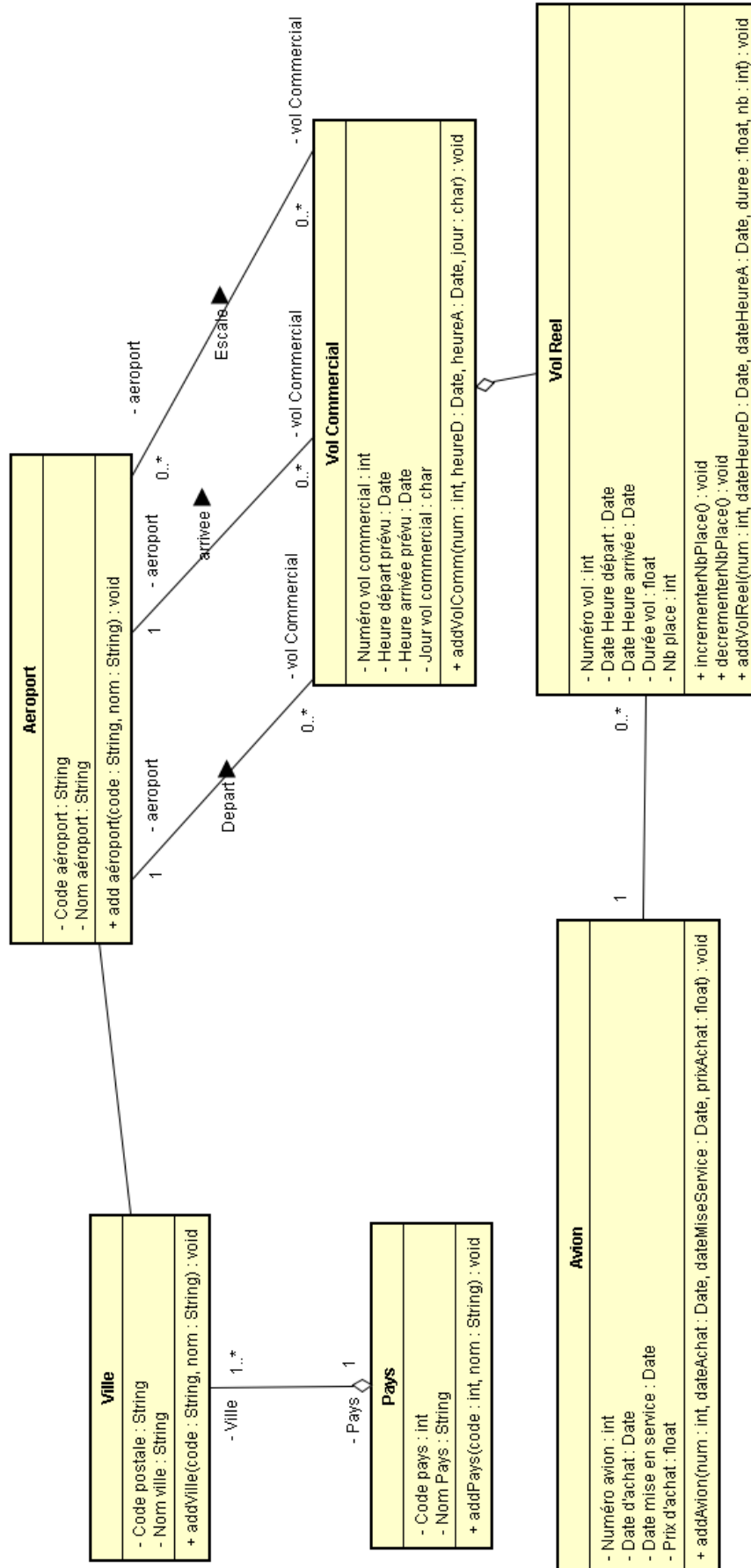


Fig. 17 : Domain model Vol



G. Diagramme de classe

Voici notre diagramme de classe pour le système. On trouve dans ce diagramme qu'une réservation sera demandée par un seul client. Et un client peut effectuer plusieurs réservations. De plus, une réservation est composée de plusieurs billets. Un billet concerne un et un seul passager. Le passager peut avoir plusieurs billets. Une réservation est faite pour un seul vol réel. Un vol réel peut concerner 0 ou plusieurs réservations.

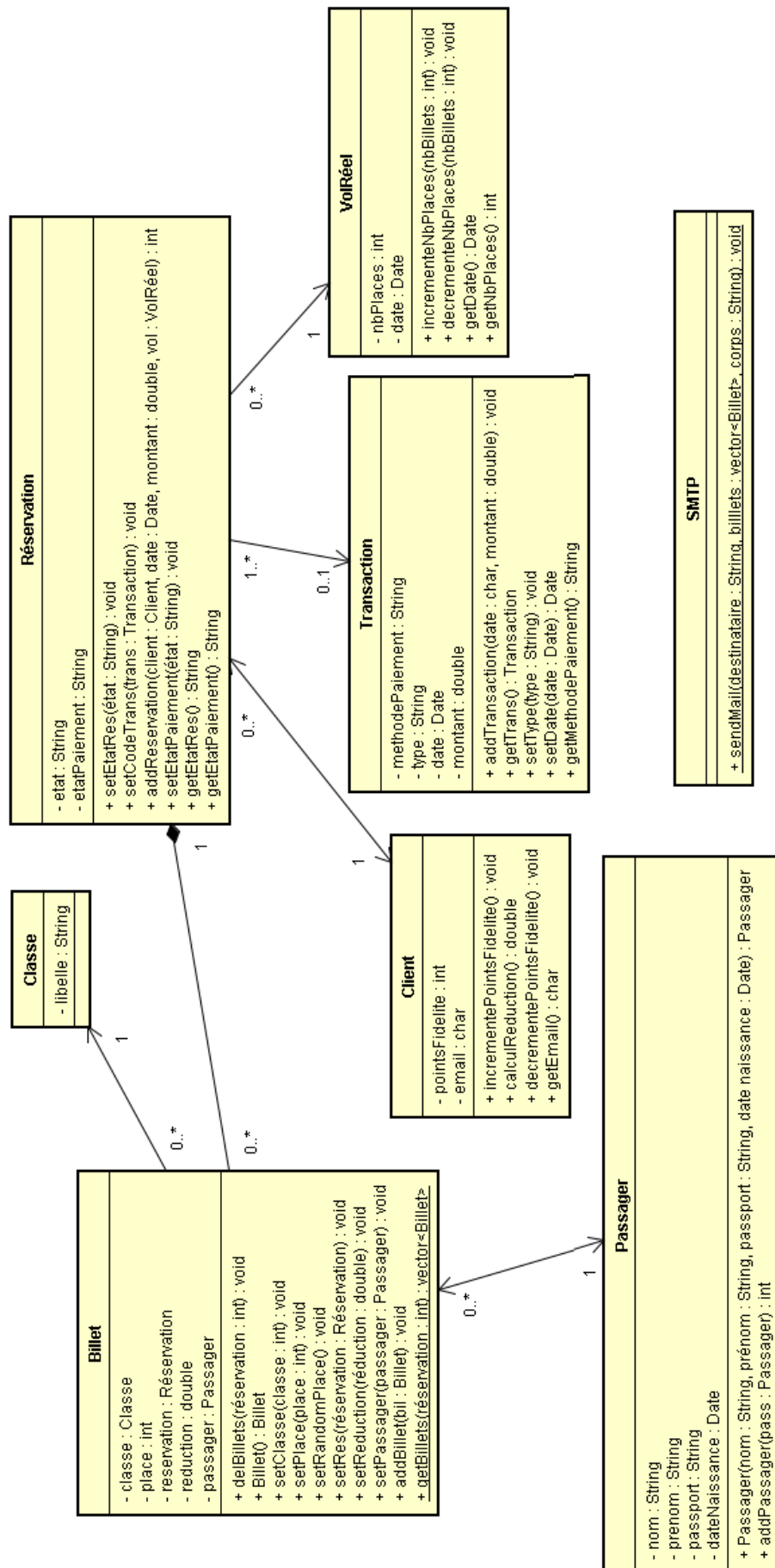


Fig. 18: Diagramme de classe



H. Diagramme d'état

Ce diagramme représente les interactions asynchrones au sein du système de réservation. L'appel synchrone de fonctionnalité n'est pas toujours un moyen de communication satisfaisant, notamment dans notre système.

Ce diagramme détermine également les états stables par lesquels passe le système.

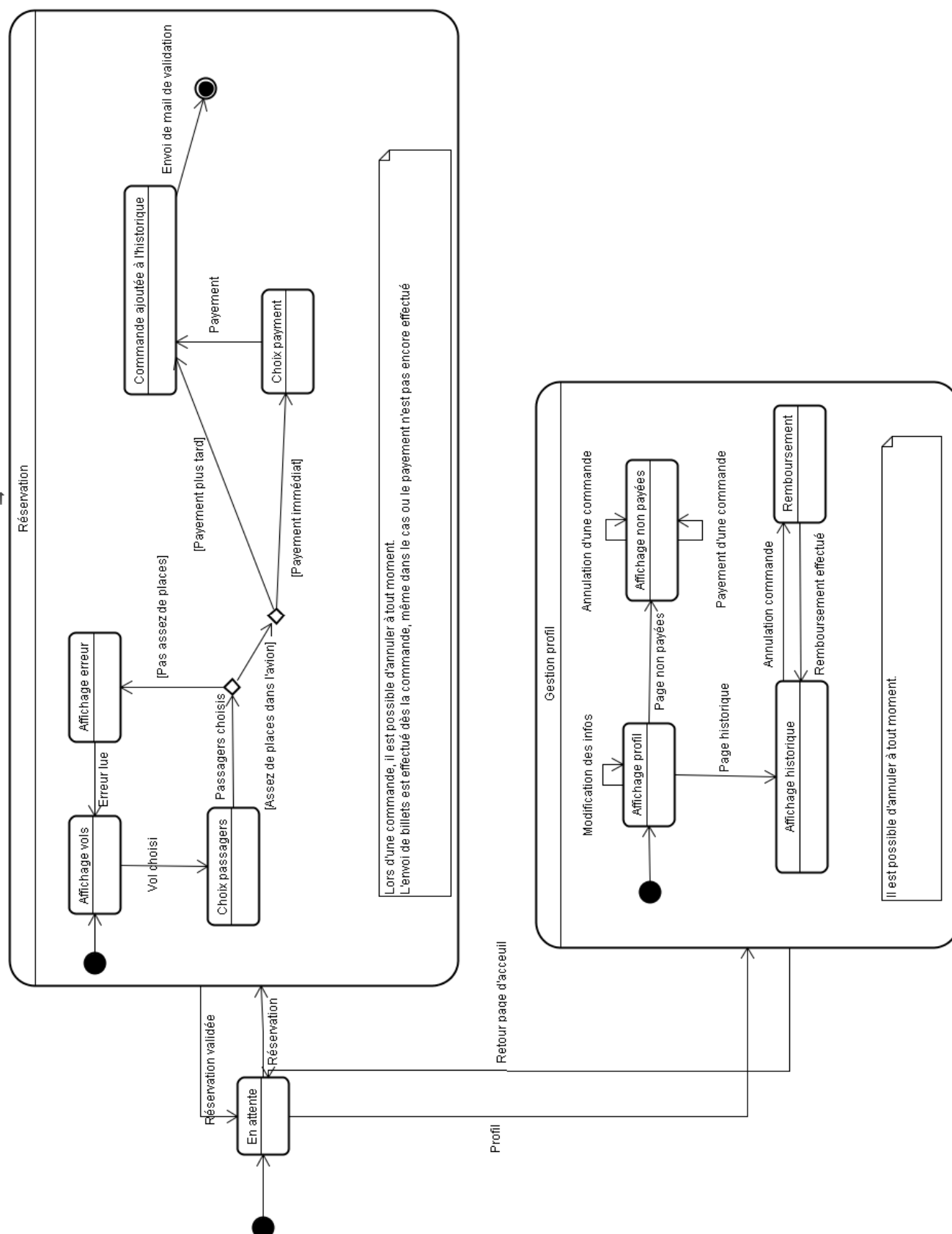


Fig. 19: Diagramme d'états de transitions

V. Spécification des interfaces

A. Interface matériel/logiciel

Pour utiliser cette application, nous avons besoin d'un poste client et d'un poste serveur.

Configuration minimale requise :

- ✓ Serveur :
Configuration minimale pour Oracle 11g. A savoir : un PC Windows processeur 1Ghz avec 256 Mo de mémoire vive (512 recommandé). L'espace disque dépend des options que l'on souhaite installé mais ici 3 Go suffiront.
- ✓ Client :
Configuration minimale standard. Un PC Windows équipé d'un processeur à 1Ghz avec 512 Mo de mémoire vive est suffisant.

Afin d'établir une communication entre le serveur et le client chacun doit disposer d'une connexion à internet de type Ethernet, WIFI ou tous autres du même genre.

B. Interface logiciel/logiciel

- ✓ Serveur :
Cette application devra fonctionner avec le logiciel Oracle 11g qui sera chargé de gérer la base de données. Pour ce faire, il faut installer sur le poste serveur, le logiciel Oracle 11g serveur ainsi que la base de données et l'application en elle-même. Pour finir, les manipulations sur la base de données s'effectueront à l'aide de logiciels développés par Oracle tel que SQL Plus ou SQL Développeur, et Apex.
- ✓ Client :
Concernant le client, il suffit juste d'avoir un poste ayant un accès internet ainsi qu'un navigateur web. Il est conseillé d'utiliser Google Chrome comme navigateur web pour une meilleure fiabilité.

C. Interface Homme/logiciel

Nous allons maintenant présenter les deux interfaces disponibles. La première a été développée en PL / SQL, et la 2^{ème} en Apex.

Il faut noter qu'il n'y a pas de consolidation entre les deux applications.

Fig. 20: Interface développée en PL / SQL

G10_FLY_ADMINISTRATION

Welcome: G10_FLY Logout

Gestion vols

Gestion avions

Gestion employés

Consultation personnes

Vols commerciaux

Cancel

Delete

Submit

<input type="checkbox"/>	Numéro vol commercial	Code aéroport arrivée	Code aéroport départ	Heure départ prévue	Heure arrivée prévue	Durée prévue	Jour
<input type="checkbox"/>	1	BFT	CDG	12:00	18:00	02:00	LUN
<input type="checkbox"/>	2	TPL	AMD	00:00	00:00	00:00	LUN
<input type="checkbox"/>	3	AMD	TPL	00:00	00:00	00:00	LUN
<input type="checkbox"/>	4	ZGT	BFT	00:00	00:00	00:00	MER
<input type="checkbox"/>	5	RTD	AMD	00:00	00:00	00:00	VEN
<input type="checkbox"/>	6	CDG	TPL	00:00	00:00	00:00	JEU
							LUN
							MAR
							MER
							JEU
							VEN
							SAM
							DIM

Add Row

VI. Besoins en performance

I. Nombre maximum de terminaux

La recherche de vols doit être accessible à 5000 personnes simultanément minimum. Ceci pour assurer une réponse rapide du système même en cas de forte utilisation.

II. Nombre maximum de transactions simultanées

Le système devra être capable de traiter au moins 100 transactions simultanées (réservation d'un vol), Il y a toujours plus de recherche de vols que de réservation.

III. Nombre de fichiers et leurs tailles

Vu que la génération des billets (en extension .pdf) se fait d'une manière temporaire et à chaque fois que le client demande cette action, il n'y a pas beaucoup de fichiers à sauvegarder à part les fichiers .css et les images de l'interface.

IV. Temps de réponse souhaité

La recherche d'un vol doit être rapide : un temps de réponse inférieur à la seconde est souhaitable. La réservation d'un vol, est par contre moins critique et nous pouvons autoriser un temps de réponse d'une dizaine de secondes.

V. Les contraintes liées à l'environnement (temps réel)

Quand un nouveau vol est rajouté, il doit apparaître immédiatement dans les recherches. Le nombre de places restantes doit aussi être maintenu à jour en fonction des réservations et désistements.

VII. Contraintes de développement

I. Fiabilité et tolérance aux fautes

Les données affichées doivent être fiables. Il faudrait dédoubler la base de données pour ne jamais la perdre en cas de défaut matériel. Ceci aidera aussi à garder le système en marche en cas de panne.

Une sauvegarde de la base de données sera faite quotidiennement, ceci en prévision du cas exceptionnel d'une défaillance simultanée de tous les serveurs contenant la base de données. Nous aurons toujours une base de données de secours (bien que pas entièrement à jour) dans ce cas.

II. Le comportement du système dans des situations anormales (les exceptions critiques)

Différentes situations anormales peuvent se produire :

- Charge très importante du système, dans ce cas le système refusera des requêtes pour maintenir sa stabilité et garder un temps de réponse acceptable pour les requêtes qui ont été acceptées. Les utilisateurs dont les requêtes ont été refusées seront invités à réessayer ultérieurement si leurs requêtes sont valides. Dans le cas contraire, les IP à l'origine de ces requêtes seront bannies temporairement pour réduire la charge.
- Malgré le dédoublement prévu de la base de données, les différents serveurs contenant la BDD tombent, dans ce cas le système ne peut plus continuer à fonctionner. Il devra donc se couper automatiquement lorsque l'accès à la BDD est impossible.

III. Sécurité

a. Restriction sur l'utilisation de certaines commandes

Certaines commandes ne sont pas faites pour être utilisées par de « simples utilisateurs » :

- Ajout / modification / suppression d'un vol.
- Ajout / modification / suppression d'un aéroport.
-

Ces commandes sont réservées au personnel qualifié pour faire ce genre de modification (administrateur avec Apex).

b. Contrôle des accès aux données

Tous les accès et toutes les modifications du système seront logués pour éviter les abus et en cas de présence de données falsifiées, nous pourrions savoir comment elles sont apparues et ainsi les corriger plus facilement.

En plus, l'accès est géré pour chaque page, en vérifiant si l'utilisateur est logué ou pas avant l'affichage du contenu, et c'est à l'aide de la déclaration d'un contexte, dans lequel on ajoute l'ID de l'utilisateur comme variable globale après une authentification successive.

c. Utilisation de mot de passe

Les commandes restreintes seront protégées par un mot de passe, qui ne sera pas lui-même sauvegardé dans la base de données, mais un hachage de lui pour des raisons de sécurité.

d. Utilisation de standards en ce qui concerne les méthodes, outils et langages de développement.

L'analyse du projet est faite en UML pour assurer une conception solide.

La base de données sera déployée après une conception dictée par MERISE.



IV. Conclusion

Dans le cadre de ce projet de l'UV GL52, nous avons conçu et développé un site web d'une compagnie aérienne en partant d'un dossier de spécification afin de réaliser un site web complet qu'on en a pu enfin satisfaire toute exigence conceptuelle et technique.

Ce projet nous a permis de mettre en pratique les notions acquises au cours du semestre de Printemps 2013. Nous avons ainsi pu suivre les étapes d'analyse des besoins et de création des modèles basés sur le langage UML 2.0 , et implémenté la base de donnée en utilisant Oracle comme système de gestion de base de données(DBMS).

Nous avons pu constater que la définition du projet, de ses besoins et des différents diagrammes nous ont permis d'accélérer la phase de codage car tout était établi et il suffisait de suivre le plan fixé. On a ainsi pu se rendre compte de l'importance de cette phase dans un projet.

V. Références

<http://psoug.org/reference/http.html>

http://www.oracle-base.com/articles/misc/dbms_session.php

<http://stackoverflow.com/questions/5897438/md5-in-oracle-dbms-obfuscation-toolkit-md5>

VI. Index

Fig. 1 :	MCD Gestion Vol
Fig. 2 :	MCD Gestion Client
Fig. 3 :	MCD Gestion Flotte
Fig. 4 :	Diagramme de cas d'utilisation Client, Internaute
Fig. 5 :	Diagramme de cas d'utilisation Employé
Fig. 6 :	Diagramme de cas d'utilisation Administrateur
Fig. 7 :	Diagramme de séquence niveau système Réserver vol
Fig. 8 :	Diagramme de séquence niveau système Payer
Fig. 9 :	Diagramme de séquence niveau système Annuler réservation
Fig. 10:	Diagramme de séquence niveau système Rembourser
Fig. 11:	Diagramme de séquence niveau objet Réserver vol
Fig. 12:	Diagramme de séquence niveau objet Payer
Fig. 13:	Diagramme de séquence niveau objet Annuler réservation
Fig. 14:	Diagramme de séquence niveau objet Rembourser
Fig. 15 :	Domain model Client
Fig. 16 :	Domain model Flotte
Fig. 17 :	Domain model Vol
Fig. 18 :	Diagramme de classe
Fig. 19 :	Diagramme d'états de transitions
Fig. 20 :	Interface développée en PL / SQL
Fig. 21 :	Interface développée avec Apex

VII. Annexes

UML : Unified Modeling Language (Last version UML 2.4.1).

HTML: HyperText Markup Language (HTML 5).

CSS: Cascading Style Sheets

Merise : une méthode d'analyse, de conception et de gestion de projet informatique.

MCD : Modèle Conceptuel du Donnée.

MLR : Modèle Logique Relationnel

PL/SQL : Procedural Language/Structured Query Language

Apex : Oracle Application Express

Htp et **htf** : hypertext procedures and hypertext functions packages generate HTML tags.

W3C : World Wide Web Consortium