



# APPLICATION **AIR CHANCE**

Rapport réalisé par :  
LYASSIMANE  
VIGNE MATHIS  
EL AOUADI RAYHANA

Encadré par :  
M. FABRICE JOUANOT



# Sommaire

- Introduction
- Conception
- Étude technique
- Réalisation
- Comment réaliser l'application

## Introduction

Comme étant des étudiants en troisième année de licence MIAGE (Méthodes Informatiques Appliquées à la Gestion des Entreprises) , et dans le cadre de renforcer nos acquis dans le module de « Base de données » nous avons réalisé un projet qui consiste à créer une application Java qui accède via le protocole JDBC à une base de données sur un serveur distant pour la compagnie aérienne Air Chance.

La Compagnie souhaite informatiser la gestion du planning des vols et la gestion des réservations client. Pour ce fait , nous avons utilisé: Java comme langage de programmation pour la partie application et Oracle (SQL,PL/SQL) JDBC pour la partie base de données.

Nous procéderons dans une première partie de ce rapport à présenter notre conception du programme, ainsi que les modifications appliqués sur notre diagramme, et dans la deuxième partie nous allons décrire les fonctionnalités réalisés l'application.

# Conception

Cette partie a pour objectif de présenter l'ensemble des étapes d'élaboration du projet. Nous avons commencé par l'étude de l'existant, ensuite on a passé à la phase de conception et enfin l'implémentation et le codage.

## Diagramme UML :

Le Diagramme UML sert à assurer une modélisation de très haut niveau indépendamment du langage et des environnements, il fait également collaborer les participants de tous membres du groupe autour d'un même document de synthèse ,il fait des simulations avant de construire l'application et documente le projet.

Après avoir regardé le cahier de charge de notre application, nous avons eu l'uml ci-dessous :

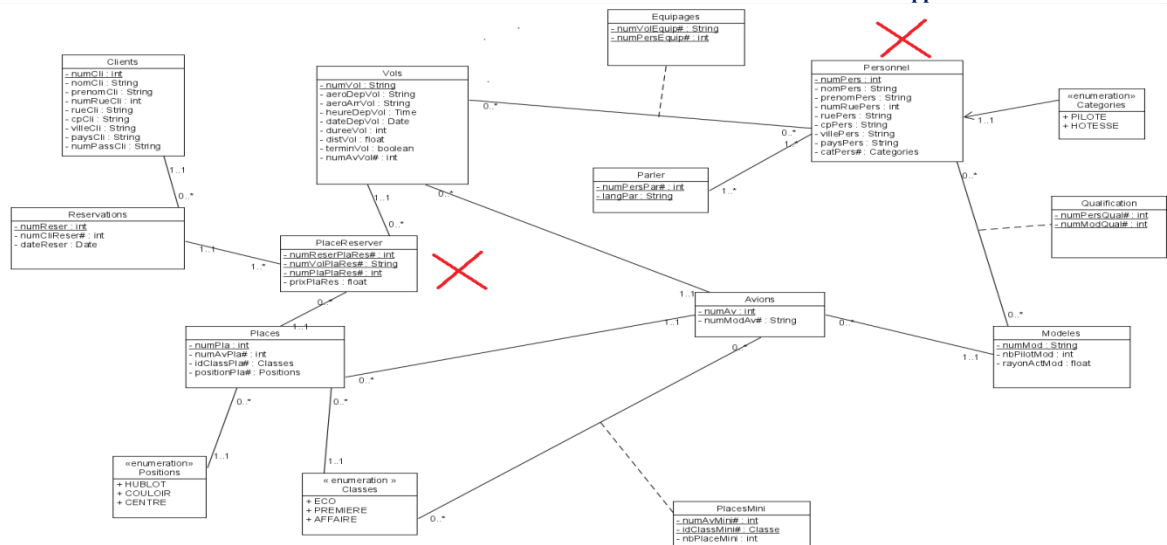


Figure 1 : 1<sup>er</sup> UML réalisé (Rapport intermédiaire)

En suivant les remarques et les instructions de notre professeur, après l'envoi du rapport intermédiaire, on s'est rendu compte que :

- Quelques-unes de nos classe ont beaucoup de propriétés en commun et nous avons donc décidé de faire recours à la notion d'héritage.
- nous avons remarqué qu'il faut créer une classe catalogue qui contiendra le prix pour optimiser la gestion de ce dernier.
- Nous avons rajouter un attribut heure de vol, les heures de vols ne pouvant être calculés en cas de suppression d'un vol ...

Ci-dessous le résultat de notre diagramme UML final :

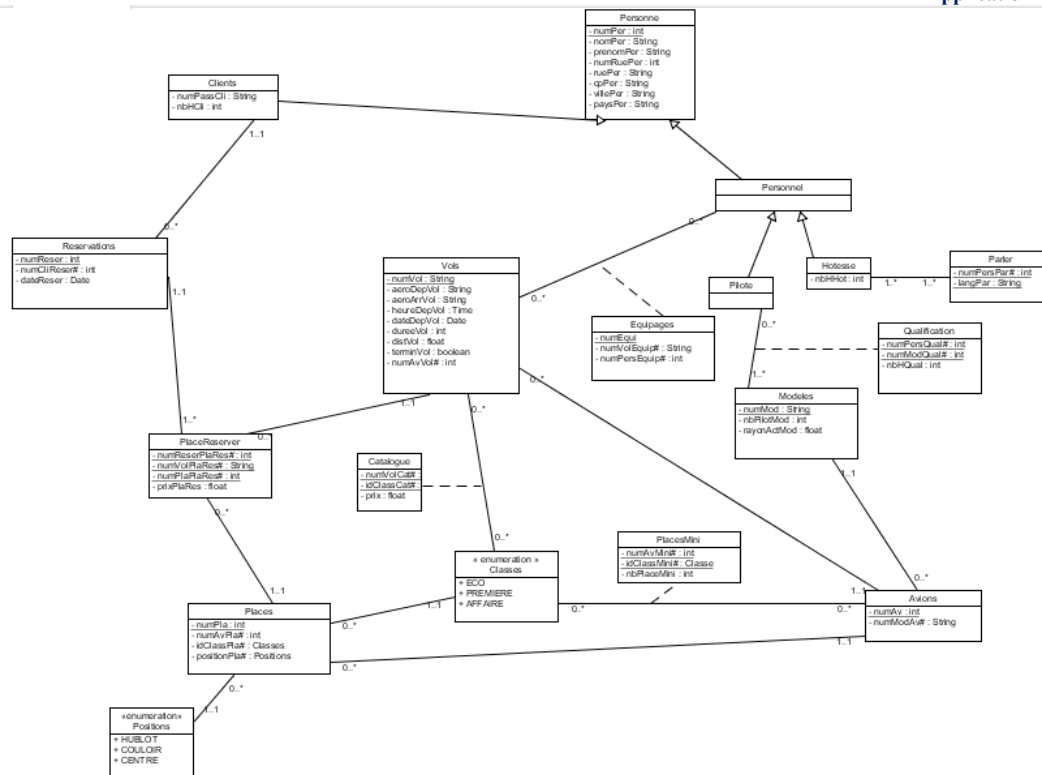


Figure 2 : UML final

## Étude technique :

ce chapitre présente ,Le langage utilisé ensuite décrit le design pattern adopté ,et enfin il présente quelques outils et technologies utilisés dans la mise en œuvre du projet.

## Langage et technologie :

### Le langage Java :

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy, présenté officiellement le 23 mai 1995 au SunWorld. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

### Pourquoi Java ?

Le premier avantage de java c'est qu'il est portable et donc exécutable sur n'importe quel système, puis il s'agit d'un langage très populaire, donc on peut facilement trouver des documentations et de l'aide en ligne pour coder/aider/reprendre du Java. Et d'ailleurs il dispose souvent des API qui correspondent à peu près à tous besoin.

### Oracle :

Oracle Database est un système de gestion de base de données relationnelle qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet.

### DAO :

Le modèle DAO (Data Access Object) est un modèle structurel qui nous permet d'isoler la couche application/métier de la couche de persistance (généralement une base de données relationnelle)

La fonctionnalité de cette API consiste à masquer à l'application toutes les complexités inhérentes à l'exécution d'opérations CRUD dans le mécanisme de stockage sous-jacent. Cela permet aux deux couches d'évoluer séparément sans rien savoir l'un de l'autre.

## Outils :

### Eclipse :



Eclipse IDE est un [environnement de développement intégré](#) libre (le terme Eclipse désigne également le [projet](#) correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel [langage de programmation](#). Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son [architecture](#) totalement [développée autour](#) de la notion de plug-in (en conformité avec la [norme](#) OSGi) : toutes les fonctionnalités de cet atelier [logiciel](#) sont développées en tant que plug-in.

### Sublime text :



Sublime Text est un [éditeur de texte](#) générique codé en [C++](#) et [Python](#), disponible sur [Windows](#), [Mac](#) et [Linux](#). Le logiciel a été conçu tout d'abord comme une extension pour [Vim](#), riche en fonctionnalités<sup>1</sup>.

Depuis la version 2.0, sortie le [26 juin 2012](#), l'éditeur prend en charge 44 [langages de programmation](#) majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares.

## Réalisation :

### 1 – Partie base de données :

#### Création de la base de données :

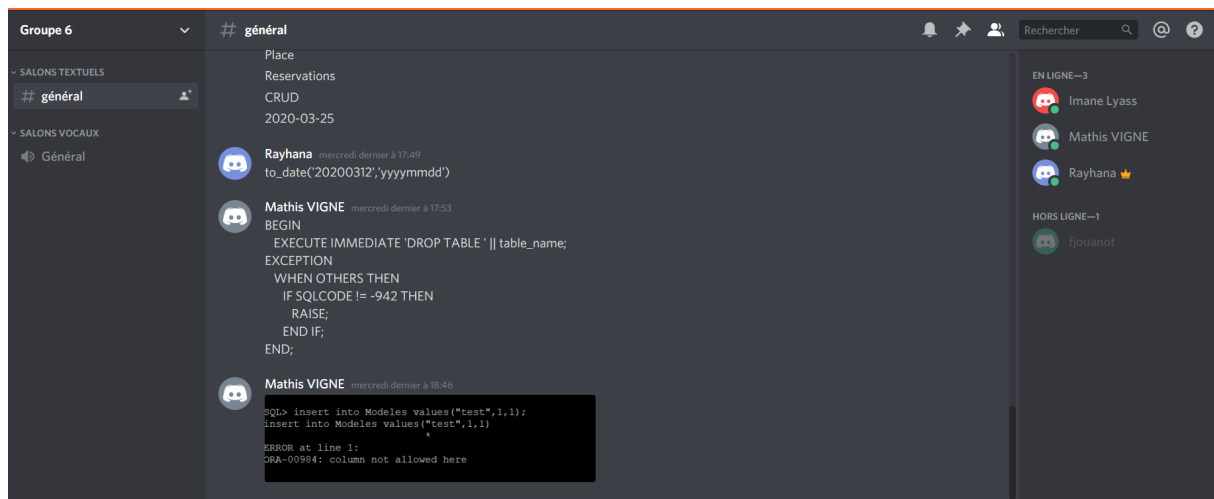
La 1ère étape était de créer notre base de données, nous avons donc créer nos tables en utilisant l'éditeur Sublime Text , nos tables sont les suivants :

- **table Catalogue;**
- **table PlaceReserver;**
- **table Places;**
- **table Reservations;**
- **table Parler;**
- **table Equipages;**
- **table PlacesMini;**



- **table Qualification;**
- **table hotesse;**
- **table pilote;**
- **table Positions;**
- **table Clients;**
- **table Vols;**
- **table Avions;**
- **table Classes;**
- **table Modeles;**

nous avons rencontré quelques difficultés lors de l'exécution de nos scripts pour de se connecter de serveur de l'Imag car les ressources de l'UGA rencontraient des problèmes ce qui nous a fait perdre pas mal de temps, pour ce fait , nous avons créé un groupe sur Discord ou on se réunit chaque jour.



## 1 – Partie code :

### Création des classes et des DAO :

Après avoir réussi la connexion avec notre base de données, nous avons créé toutes nos classes et nos DAO :

## Réalisation des fonctionnalités de l'application :

Nous avons essayé de faire le maximum possible des fonctionnalités demandés, mais en même temps, nous avons décidé de répartir les fonctionnalités entre nous et de les faire parfaitement au lieu d'essayer de tout faire d'une façon absurde.

En vue des problèmes rencontrés avec la connexion de la base de donnée à chaque fois(chose qui empêchait de faire des test et donc qui empêcher d'avancer dans le code) et vu qu'on est que 3 personnes en groupes , nous n'avons pas eu le temps de réaliser toute les fonctions .

En conséquence voici la liste des fonctions réalisées avec leurs explications :

### Choix de la fonctionnalité :

Choisissez une fonctionnalité en tapant de 1 à 5 dans la liste suivante, pour quitter choisissez un numéro ne correspondant à aucune fonctionnalité

- 1 - Ajouter d'un personnel de vol
- 2 - Suppression d'un personnel de vol
- 3 - Confirmation de la terminaison d'un vol
- 4 - Consultations des réservations d'un client
- 5 - Suppression d'une réservation

Faites votre choix :

Lorsque l'on lance le logiciel, il faut commencer par choisir une fonctionnalité parmi les la liste ci-dessus en écrivant le numéro associé.

### Ajout de personnel de vol :

```

||  Souhaitez-vous ajouter un pilote ou une hotesse ?
est Pour ajouter un pilote tapez 1
    Pour ajouter une hotesse tapez 2
    Tout autre choix vous redirigera au menu
    1
    Saisissez son nom :
    Test
    Saisissez son prenom :
    Test
    Saisissez son numéro de rue
    1
    Saisissez sa rue :
    Test
    Saisissez son code postal
    12345
    Saisissez sa ville :
    Test
    Saisissez son pays :
    Test
    Numéro | Nbr. Pilotes | Rayon d'action
    Boeing737 | 1 | 70.89
    Airbus737 | 1 | 80.0
    TBM700 | 1 | 95.69

    Saisissez un numéro de modèle d'avion pour lequel ce pilote est qualifié
  
```

ensuite demandé de choisir si l'on souhaite ajouter un pilote ou un hôtesse.

Les informations communes aux 2 sont ensuite demandé. Puis si l'on a choisi d'ajouter un pilote la liste des modèles d'avion est afficher et il est demander à l'utilisateur de saisir un modèle pour lequel il est qualifié.

Lors du choix d'un modèle, une vérification de son existence dans la base est faite, n que le même modèle n'est pas choisis 2 fois et qu'au moins 1 modèle est choisi.

Dans le cas du choix d'ajout d'hôtesse, la procédures est la même avec les langues qu'elles parlent au lieu des modèles.

```

    Saisissez un numéro de modèle d'avion pour lequel ce pilote est qualifié
    Boeing890
    Ce modèle n'existe pas !
  
```

```

Le pilote doit être qualifié pour au moins 1 modèle
Saisissez un numéro de modèle d'avion pour lequel ce pilote est qualifié
Airbus737
Si vous souhaitez ajouter une autre qualification tapez 0, sinon tapez un autre numéro
0
Saisissez un numéro de modèle d'avion pour lequel ce pilote est qualifié
Airbus737
Ce pilote a déjà cette qualification
Si vous souhaitez ajouter une autre qualification tapez 0, sinon tapez un autre numéro
  
```

### Suppression d'un personnel de vol :

Pour supprimer un personnel de vol l'utilisateur doit commencer par choisir, comme précédemment si il souhaite supprimer un pilote ou une hôtesse.

La liste de ceux-ci est ensuite affichée, il lui suffit de saisir le numéro de celui qu'il souhaite supprimer pour que ce soit fait.

```
Souhaitez-vous supprimer un pilote ou une hotesse ?
Pour supprimer un pilote tapez 1
Pour supprimer une hotesse tapez 2
Tout autre choix vous redirigera au menu
1
num Pilote | nom Pilote | prenom Pilote | num Rue | rue | code postale | vil
4 | Test | Test | 1 | Test | 12345 | Test | Test | 0
1 | Park | Edward | 17 | Rue Gangnam | 1236 | Seoul | Corée du Sud |
2 | Lee | Jessica | 10 | Rue Seoul | 1236 | Busan | Corée du Sud | 0
3 | Croft | Hannah | 25 | Rue NYC | 13965 | California | USA | 0

Saisissez le pilote que vous souhaitez supprimer :
4
num Pilote | nom Pilote | prenom Pilote | num Rue | rue | code postale | vil
1 | Park | Edward | 17 | Rue Gangnam | 1236 | Seoul | Corée du Sud |
2 | Lee | Jessica | 10 | Rue Seoul | 1236 | Busan | Corée du Sud | 0
3 | Croft | Hannah | 25 | Rue NYC | 13965 | California | USA | 0
```

### Terminaison d'un vol :

Pour terminer un vol, nous affichons la liste des vols, l'utilisateur doit ensuite saisir le numéro de vol qui est terminé. Ce vol est donc mis à jour comme terminé et la durée du vol est ajoutée aux passagers et à l'équipage (pour le pilote, le nombre d'heure en vol sur le modèle de l'avion de ce vol est mis à jour et non un nombre d'heure total comme pour les clients et les hôtesse).

Nous vérifions que le vol saisi existe et qu'il n'est pas déjà terminé avant d'appliquer les modifications.

```
ID | Aero. Dep. | Aero. Arr. | Heure Dep. | Duree | Distance | Terminer ? | Avion
19E56 | CasablancaM5 | OrlyParis | 16:00:00 | 2020-02-05 | 3 | 2500 | Non terminé | 6
18B56 | BarajasMadrid | MarrakeshM6 | 18:00:00 | 2020-03-15 | 2 | 1500 | Non terminé | 7
11B00 | TangerIbnBatouta | MilanMalpensa | 12:00:00 | 2020-04-15 | 4 | 3500 | Terminé | 8

Choisissez le vol que vous souhaitez terminer
18B57
java.sql.SQLException: Ce vol n'existe pas !
Ce vol n'existe pas !
```

```
ID | Aero. Dep. | Aero. Arr. | Heure Dep. | Duree | Distance | Terminer ? | Avion
19E56 | CasablancaM5 | OrlyParis | 16:00:00 | 2020-02-05 | 3 | 2500 | Non terminé | 6
18B56 | BarajasMadrid | MarrakeshM6 | 18:00:00 | 2020-03-15 | 2 | 1500 | Terminé | 7
11B00 | TangerIbnBatouta | MilanMalpensa | 12:00:00 | 2020-04-15 | 4 | 3500 | Terminé | 8

Choisissez le vol que vous souhaitez terminer
18B56
java.sql.SQLException: Ce vol est déjà terminé !
Ce vol est déjà terminé !
```

### Consultation des réservations d'un client :

Pour cette fonctionnalité, nous commençons par afficher la liste des clients, l'utilisateur doit ensuite saisir le numéro du client dont il souhaite consulter les réservations. On affiche pour chacune des réservations du client son contenu (place, vol concerné , prix de la place)

```
num Client| nom Client | prenom Client |num Rue|rue|code postale|ville|pays|numPassCli|nbHeures
10 | Lyass | Imane | 17 | Stade de la france | 92000 | Paris | France | HJ12356 | 45
11 | Vigne | Mathis | 14 | Rue de Rome | 14000 | Venise | Italie | AB7896 | 30
12 | EL-Aouadi | Rayhana | 16 | Rue de l'Italie | 23000 | NEW YORK | USA | ZZ1478 | 27
25 | Rodriguez | Camilla | 17 | Rue Georges Gabriel | 16400 | Madrid | Espagne | ET36900 | 35
38 | Escobar | Pablo | 26 | Rue de chili | 18560 | Pogota | Colombie | BY2356 | 105
45 | Imrane | Mariem | 16 | Rue de l'Algérie | 10000 | Rabat | Maroc | RO2586 | 75

Choisissez le clients dont vous voulez consulter les réservations :
11
Reservation n° 2 du 2020-02-08 par le client n° 11 (Vigne Mathis)
Vol n°11B00 - Place n°15 - Prix 2500.0
Vol n°11B00 - Place n°40 - Prix 3500.0
```

### Suppression d'une réservation :

Pour cette fonctionnalité, nous affichons la liste des réservations, l'utilisateur doit ensuite saisir le numéro de celle qu'il souhaite supprimer, puis celle-ci est supprimer (les éléments de la table des places réserver associer aussi évidemment).

```
Num Reservation | numCliReser | dateReser
1 | 10 | 2020-01-05
2 | 11 | 2020-02-08
3 | 12 | 2020-03-04
4 | 25 | 2020-01-10

Choisissez la réservation que vous souhaitez supprimer
4
Num Reservation | numCliReser | dateReser
1 | 10 | 2020-01-05
2 | 11 | 2020-02-08
3 | 12 | 2020-03-04
```

## Comment essayer l'application :

Pour essayer l'application il faut récupérer le projet sur GitHub au lien suivant :

<https://github.com/vignema/Projet-BD-Lyass---EL-Aouadi---Vigne>

Clonez ensuite le projet sur votre ordinateur avec le lien de clonage suivant:

Avec la méthode HTTPS : <https://github.com/vignema/Projet-BD-Lyass---EL-Aouadi---Vigne.git>

Avec la méthode SSH : [git@github.com:vignema/Projet-BD-Lyass---EL-Aouadi---Vigne.git](https://github.com/vignema/Projet-BD-Lyass---EL-Aouadi---Vigne.git)

Une fois cloner, aller dans le dossier src du projet et modifier le fichier BD.properties en y indiquant votre nom d'utilisateur et votre mot de passe nécessaire pour se connecter à votre base de données Oracle.

Pour créer la base de données et avoir un jeu de données, le dossier scripts contient les fichiers sql nécessaires.

Vous pouvez ensuite ouvrir le projet via Eclipse et lancer le fichier squelette\_appli.java situé dans le package main du dossier src du projet.