

Rapport de projet de Base de Données

« Application Air Chance »

Groupe 3 :

Idriss SAGARA

Abderrahmane BEKKOUCH

Soufiane KABAD



I. Introduction :

L'objectif du projet est la réalisation d'une application Java qui accède via le protocole JDBC à une base de données sur un serveur distant.

L'application est constituée de deux grandes parties : Gestion de la planification des vols et la Gestion de réservation

Pour ce qui est de l'organisation du projet et des fichiers le composant, l'application possède 4 packages :

- Un Package « airChance »,
- Un Package « connection »,
- Un Package « dao » et
- Un Package « data ».

- Etape à suivre pour l'exécution du projet:

Après avoir récupéré le dépôt du projet sur git:

Dans le dossier de l'application veuillez trouver deux fichiers de scripts SQL :

Aero.sql : contient le schéma relationnel.

Insertion.sql : contient les données à insérer dans la base de données.

- il faut exécuter le fichier **Aero.sql** et **insertion.sql** : pour charger les tables, procédures, triggers. et faire des insertions pour peupler la base .
- Dans le package "airChance" vous trouvez le fichier `squelette_appli.java`, ce fichier contient un menu pour toutes les fonctionnalités du projet, et chaque fonctionnalité est identifiée via un numéro à choisir pour la tester.
- Durant le test nous avons pris soin de tracer toutes les tâches après exécution de chaque requête, en montrant avant et après exécution les questions

II. Schéma conceptuel et relationnel :

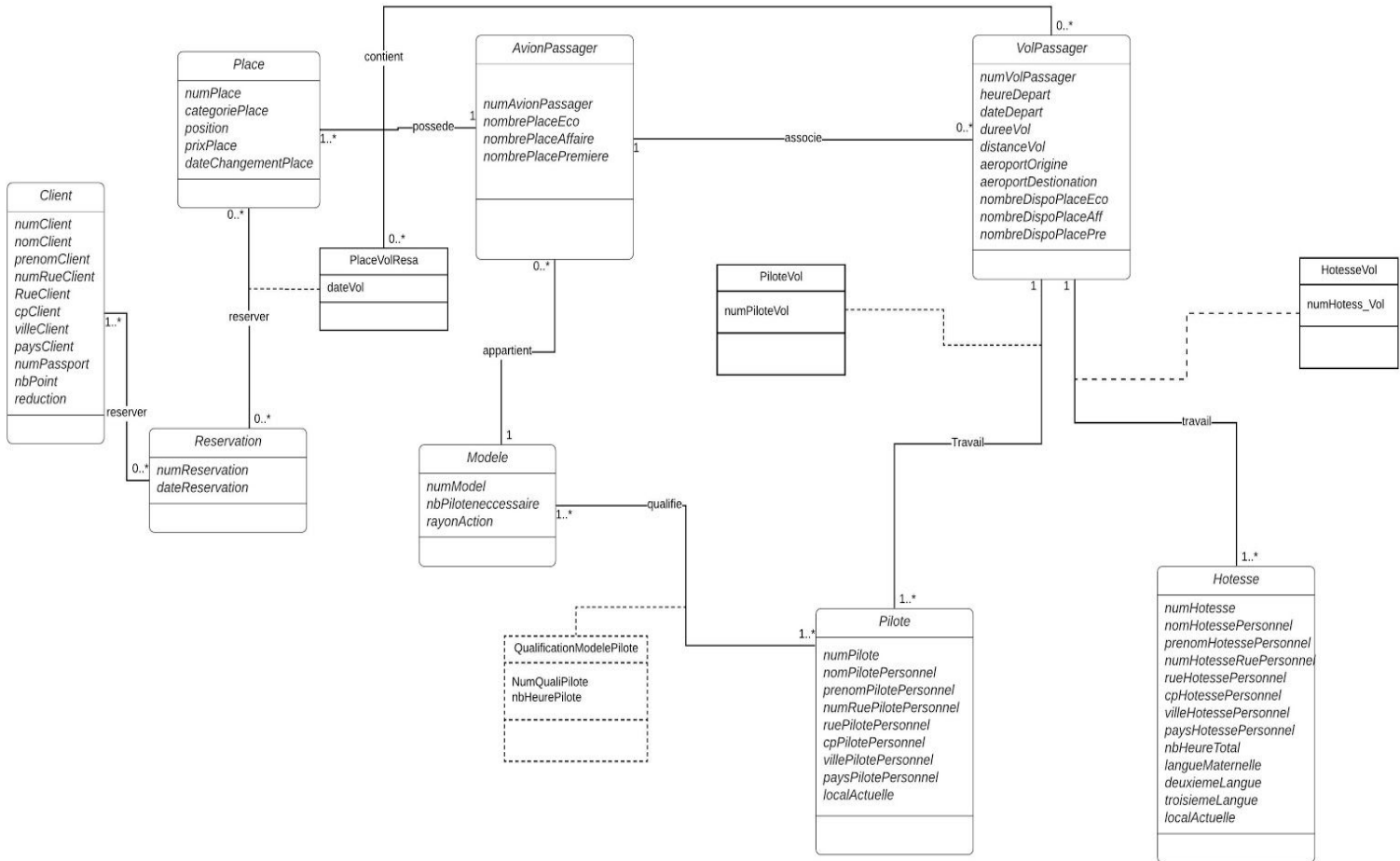
- Schéma conceptuel :

Pour le Schéma conceptuel nous avons optés pour une modélisation UML et depuis le rapport intermédiaire ont eu certains changements parmi lesquelles nous avons :

- ❖ la revue du schéma du modèle de la base de données;
- ❖ la suppression l'héritage entre personnel et pilote et hôtesse.
- ❖ ajout d'autres classes qui ont facilitées le déroulement du projet : HotesseVol, PiloteVol, QualificationModelePilote, PiloteVolResa.

Schéma conceptionnel

Soufiane Kabad | March 31, 2020



- **Modèle relationnel :**

Modele(nummodel, nombrepilotenecessaire, rayonaction)

AvionPassager(numAvionPassager, ____, nombrePlaceEco, nombrePlaceAff, nombrePlacePre, #nummodel).

Place(numplace, position, typeclasse, prixplace, datechangmentplace, #nummodel)

Client(numclient, nomclient, prenomclient, rueclient, cpclient, villeclient, paysclient, numpassport, nombrepont, reduction).

VolPassager(numvolpassager, dateEnregistrementVol, Datedepart, dureeVol, DistanceVol, aeroportorigine, aeroportdestination, nombrePlaceDispoEco, nombrePlaceDispoAff, nombrePlaceDispoPre, EtatVol, Terminer, #numAvionPassager).

Reservation(numReservation, DateReservation, EtatReservation, #numplace, #numclient, #numvolpassager)

Pilote(numPilote, nomPersonnelPilote, PrenomPersonnelPilote, PaysPersonnelPilote, LocalisationActuelle,nbHeure)

Hotesse(numHotesse, nomPersonnelHotesse, LangueMaternelle, deuxiemeLangue, troisiemeLangue, LocalisationActuelle)

QualificationPilote(NumQualiPilote, nbHeurePilote , #nummodel, #numPilote)

PiloteVol(NumPiloteVol,Pr #NumVolPassager, #NumPilote)

HotesseVol(NumHotesse_Vol, #NumVolPassager, #NumHotesse)

PlaceVolResa(dateVol , #numplace , #numvolpassager, #numreservation)

III. L'état d'avancement de projet :

1. l'état du projet

Toutes les fonctionnalités demandent ont été réalisés.

pour aller vite nous n'avons pas pris compte de certain détails comme

appliquer des triggers pour recalculer le prix et mettre à jour le nombre de point de fideliter .

nous avons fait quatre procédure et trois (3) triggers qui sont :

❖ procédures

- terminerVolPassager :qui permet de finaliser un vol;
- reassigner : qui permet de réassigner d'un vol supprimé à un autre vol ;
- updateVol : qui permet de mettre à jour les informations d'un vol;
- updatepilote: qui permet de mettre à jour les informations d'un pilote ;
- updateHotesse: qui permet de mettre à jour les informations d'une hôtesse.

❖ triggers

- PILOTE_VOL: qui permet d'incrémenter le NUMPILOTEVOL dans la table pilvol ;

- PILOTE_SEQ: qui permet d'incrémenter le NUMPILOTE dans la table PILOTE;
- HOT_VOL: qui permet d'incrémenter le NUMHOTESS_VOL dans la table HOTESSE_VOL

2. l'architecture et technologie utilisées :

a. l'architecture

Pour l'architecture du projet nous avons jugé nécessaire de faire usage du **“PATRON DAO”**.

b. les technologies:

les technologies utilisées sont :

- ❖ Java et JDBC pour la partie applicative ;
- ❖ Oracle (SQL, PL/SQL) pour la partie base de données.

IV. La gestion de projet :

Pour la gestion de projet nous avons divisé les tâches comme ci-dessous :

conception du projet

- Schéma conceptuel et relationnel : pour une meilleure compréhension du projet et pour effectuer un travail cohérent chacun de notre côté on a décidé qu'il valait mieux qu'on fasse cette partie ensemble.

Réalisation du projet

- Création de la base de données ;
- Peuplement de la base de données;
- découpage du backlog (cahier de charge) en sprint et des sprints en users -stories ;
- exécution des sprints se faisaient dans l'ordre du backlog (cahier de cahier);
- réalisation des users- stories se faisaient par affectation entre les membres du groupe ;
- affectation des users-stories se fait par consultation entre nous en tenant compte de la capacité de la personne qui nous a permis d'aller plus vite , le tableau contient le partage les sprints ;

Sprints	réalisé par
1 ^{er} sprint	Idriss SAGARA
2eme sprint	Idriss SAGARA
3eme sprint	Idriss SAGARA
4eme sprint	Soufiane KABAD

5eme sprint	Soufiane KABAD
6eme sprint	Abderrahmane Bekkouch
7eme sprint	Abderrahmane Bekkouch
8eme sprint	Soufiane KABAD

V. Conclusion :

Ce projet nous a été d'une grande aide pour comprendre le côté pratique des bases de données, l'usage des connaissances acquises en UEs de SI et de POO, aussi le travail sur une nouvelle plateforme (Gitlab) nous a permis en tant que groupe d'apprendre un nouvel outil de collaboration à distance surtout en ces temps difficiles.

Annexe

Ancien schéma de la base de données

