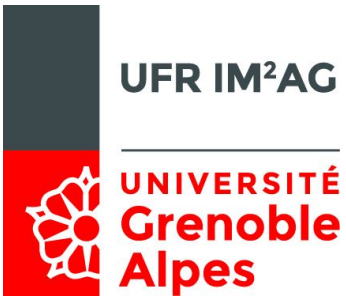
Groupe 6
TROUVIN Paul
BLONDEL Quentin
DEDIEU-MEILLE Andréas
PERRIN Kyllian



Projet Airchance

Rendu final

Des informations complémentaires sont disponibles dans le README.md contenu avec les fichiers sources.

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

Sommaire :

1. Modification depuis le rapport n°1

Nous avons ajouté un attribut idDerniereVille dans les tables Avion et PersonnelNaviguant afin de pouvoir plus simplement travailler sur la cohérence des vols, notamment pour l'écriture des triggers. Cela indique dans quelle ville se trouvent actuellement l'Avion et le PersonnelNaviguant.

2. Fonctionnalités réalisées

Gestion de la planification des vols

1. Planification d'un nouveau vol.
   Pour planifier un nouveau vol, il faut utiliser l'application Manager, puis :
   - Choisir gérer les instances vols dans le menu
   - Choisir ajouter une instanceVol
   - Rentrer la date de départ, la date d'arrivée prévue, le vol correspondant.
   - Une liste d'avions est retournée, correspondant aux informations rentrées precedement
   - Choisir l'avion
   - Une liste de PNC est retournée, pour plusieurs PNC, les mettre à la suite séparés d'une virgule
   - Une liste de PNT est retournée, pour plusieurs PNT, les mettre à la suite séparés d'une virgule
   - Une validation est demandée
   - L'instanceVol est créée !

2. Modification de la planification d'un vol existant.
   Cette fonctionnalité n'est pas implémentée, voir la partie 5.

3. Suppression d'un vol.

La suppression d'un vol est faisable, mais la réaffectation des passagers ne se fait pas (Voir partie 5). On démarre l'application Manager, puis :

- Choisir gérer les instances vols dans le menu
- Choisir supprimer les instanceVol
- La liste des instanceVol est affichée
- Rentrer l'id de l'instanceVol à supprimer
- Une validation est demandée
- L'instanceVol est supprimée

4. Confirmation de la terminaison d'un vol.

Pour confirmer l'arrivée d'un avion, lancer l'application Manager, puis :

- Choisir gérer les instances vols dans le menu
- Choisir confirmation d'une instance vol
- Affiche la liste des instanceVol en cours
- Choisir l'instance Vol à confirmer
- Une validation est demandée
- L'instanceVol est mise à jour, et les triggers ont mis la base de données à jour.

5. Ajout et Suppression d'un personnel de vol.

Pour créer ou supprimer un employé d'airChance, lancer l'application Manager, puis:

- Choisir gérer tous les personnels navigants
- Choisir ajouter ou supprimer
- Les informations nécessaires pour l'ajout sont listés au fur et à mesure
- Les listes des personnels s'affichent dans le cas de la suppression

Gestion des réservations

6. Consultation des commandes d'un client.

Pour consulter des réservations d'un client, il faut lancer l'application Client puis :

- S'identifier
- Choisir Afficher les réservations
- Entrer le type de reservation
- Choisir parmi la liste fournie l'instance vol concerné
- Choisir parmi les places disponibles
- Réservation enregistrée

7. Réservation de la part d'un client.

Pour réserver un billet, il faut lancer l'application Client, puis :

- S'identifier

- Choisir créer une réservation
- La liste des instanceVol

8. Modification de la réservation d'un client.
   Pas pris en charge voir partie 5

9. Suppression de la réservation d'un client.
   Pas pris en charge voir partie 5

14 triggers ont étés écrites et sont disponibles avec une description dans les fichiers dans /BD/InitBD/

UFR IM²AG
UNIVERSITÉ Grenoble Alpes

3. Choix effectués

Fidélité des clients :
Nous avons choisi de ne pas prendre en compte les vols de fret pour la fidélité, cependant, un client achetant plusieurs tickets de passager aura autant de fois le crédit d'heure fidélité accumulé.

Transformation du MCD en MLD :
Nous avons choisi la méthode 3 du cours de LMM afin de transformer notre MCD en MLD. La méthode 3 consiste à prendre la table et ses héritages, et de les fusionner en y ajoutant un attribut type dedans.

Gestion des associations dans l'application :
En vue de simplification, les associations du MCD ont été au maximum rapatriés dans un Objet Table. Ainsi, un PNC est lié à des Langues par l'association LanguePNC sur le MCD, cela se traduit par une liste de Langue présente dans l'objet PNC.
C'est le même cas pour PNT, qui contient un HashMap contenant un ModeleAvion et les heures qu'a passé le PNT sur cet Avion, et pour InstanceVol qui contient une liste de PersonnelNaviguant.
Dans ces classes, la construction de ces listes ne se fait pas par défaut (pour éviter une grande liste d'accès à la BD). Il faut appeler la méthode fill<nomDeLAssociation> sur l'objet en question pour remplir ces listes. Exemple sur PNC :

```
PNC myPnc = new PNC();
myPnc.importFromId("5"); // import du PNC d'id numéro 5
myPnc.fillLanguePNC();
```

Transformation de la gestion des réservations de BD vers objet :
Un client contient un Objet Reservation_Correspondance, qui est lui-même défini par une liste de Réservations (de Fret ou de Passagers).

Sélections depuis l'application :
Dans l'application, lors de consultations, même simple, nous avons choisi de reconstruire les objets associés, afin de pouvoir travailler de manière cohérente ensuite en local, sans avoir à poser les questions. De ce fait, l'import des données dans l'application et l'export dans la base est considérée comme une transaction. (Sauf si il s'agit seulement de consultation bien sûr).

UFR IM²AG
UNIVERSITÉ Grenoble Alpes

4. Elements complexes & solutions

Choix entre trigger et application pour les contraintes :
Nous avons réfléchi pendant une demi-journée pour savoir si chaque contrainte métier était plus ou moins adaptée à leur implémentation en trigger ou dans l'application. Pendant le développement, nous avons beaucoup hésité à changer quelques-uns, mais finalement nous avons choisi de rester sur ce qui était initialement prévu.

Création d'un jeu de données pour la base :
Créer un jeu de données à été assez long et compliqué à faire, même si raccourci par l'utilisation de scripts.

Complexité de l'application :
L'application est complexe dans les appels à la base : elle fait beaucoup de sélection pour pouvoir afficher et affecter ses objets.
En effet, notre erreur a été de vouloir coder chaque import et export pour chaque Objet. Ainsi, pour PNT par exemple, il existe une méthode importTablePNT(), exportTablePNT(PNT pnt), et une autre méthode présente à l'intérieur même de PNT qui s'appelle importFromId(String id). Chaque classe possède ces méthodes, ce qui a été assez long à écrire, d'autant plus qu'on ne les utilise pas toutes dans l'appli.
En résumé, nous avons un backend et une API développée, tandis que les contraintes métier applicatives et l'interface le sont moins.

5. Limites de l'application

À l'heure du rendu, notre application ne prend pas en compte un certain nombre de contraintes, faute de temps.

Liste des contraintes non respectés :

- La pause des employés d'AirChance n'est pas imposée
- Le replacement des client à cause de modification ou d'annulation d'un vol n'est pas effectuée. En effet, la modification des instanceVol n'as pas été implémentée.
- La cohérence du nombre de PN lors de la création d'un avion n'est pas respectée (Nombre de PNT requis pour le modèle, nombre de PNC requis augmenté de 1 à chaque tranche de 50 passagers…)

De plus, on peut créer une nouvelle réservation pour le client, mais cette réservation ne peut être que pour un Avion Passager : par manque de temps la réservation de type Fret n'a pas été codé.

L'application ne permet pas de modifier ou supprimer une reservation.

UFR IM²AG
UNIVERSITÉ Grenoble Alpes

6. Retours sur le projet

Bien que ce projet est intéressant, les interruptions pour les tps de système et réseaux furent plutôt contraignantes, car cela nous empêchait d'être vraiment efficace juste pour une demi-journée. De plus comme nous étions dans des groupes de TP différents, nous ne pouvions pas travailler ensemble la journée du mardi ce qui ne permettait pas une progression optimale du projet.

Nous aurions aussi souhaité des retours plus précis sur les problèmes de rapport, qui nous auraient permis de comprendre plus vite nos erreurs, alors que nous étions déjà un peu cours en temps.

Malgré tout, ce projet est une très bonne conclusion sur l'aspect des bases de données vu en cours même si la partie applicative semble superflue et vraiment difficile à terminer dans le temps imparti.