

LIVRABLE MINI PROJET POO -JAVA-

GESTIONNAIRE AEROPORT

Encadrer par

Prof : M.MOUKHAFI

Préparer par

Zakaria El Omari

Mohamed El Ouardi

Informatique S5

REMERCIEMENTS

Nous voulions prendre un moment pour exprimer notre sincère gratitude pour le cours d'Orienté Objet Java que vous avez dispensé. Votre passion, votre expertise et votre approche pédagogique claire ont considérablement enrichi notre compréhension de ce sujet complexe.

Les exemples pratiques et les exercices stimulants que vous avez partagés ont rendu le processus d'apprentissage engageant et instructif. Votre disponibilité pour répondre à nos questions a créé un environnement propice à la collaboration et à la réussite.

Nous sommes reconnaissants d'avoir eu la chance de bénéficier de votre enseignement, et nous sommes convaincus que les compétences acquises auront un impact positif sur nos parcours professionnels respectifs. Merci encore pour votre dévouement envers l'éducation. Nous vous souhaitons tout le succès dans vos projets pédagogiques futurs.

SOMMAIRE

01.

Introduction

02.

Fonctionnalités Principales

- Administration Centralisée.
- Gestion des Employés et des Rôles.
- Planification et Suivi des Vols.
- Gestion des Avions.
- Authentification Multiple.
- Consultation pour les Employés.
- Réservation pour les Passagers.

03.

Les Classes

- Classe Admin.
- Classe Employé.
- Classe Passager.
- Classe Avion.
- Classe Vol.

04.

Base de Données pour l'Application de Gestion d'Aéroport avec MySQL

- Schéma de la Base de Données et Relations entre les Tables.
- Relations entre les Tables.
- Attributs des Tables.

05.

Diagrams

06.

Conclusion

I. Introduction:

L'application de gestion d'aéroport est une plateforme complète conçue pour simplifier et optimiser les opérations au sein d'un aéroport moderne.

Dans cette perspective, l'Application de Gestion d'Aéroport **Arcane** émerge comme une solution intégrale et sophistiquée, conçue spécifiquement pour simplifier et optimiser toutes les facettes des opérations aéroportuaires

II. Fonctionnalités Principales :

- **Administration Centralisée :**
 - L'administrateur, grâce à son accès privilégié, peut gérer l'ensemble de l'aéroport. Cela inclut l'ajout d'employés avec leurs détails spécifiques, la création et l'affectation de vols, l'ajout d'avions avec leurs caractéristiques, ainsi que la génération de rapports exhaustifs sur l'état de l'aéroport.
- **Gestion des Employés et des Rôles :**
 - La plateforme offre un module de gestion des employés permettant d'affecter des emplois du temps, de définir des blocs de travail, et d'associer les employés aux avions en fonction de leurs compétences.
- **Planification et Suivi des Vols :**
 - Les administrateurs peuvent planifier des vols en spécifiant leur destination, origine, pilote, avion, et d'autres détails pertinents. Un suivi en temps réel des vols est également disponible.
- **Gestion des Avions :**
 - Les caractéristiques de chaque avion, telles que le modèle et la capacité maximale de passagers, peuvent être gérées et mises à jour facilement.
- **Authentification Multiple :**
 - L'application prend en charge une authentification multiple, permettant à chaque utilisateur, qu'il soit administrateur, employé ou passager, d'accéder uniquement aux fonctionnalités qui lui sont assignées.
- **Consultation pour les Employés :**
 - Les employés peuvent consulter leurs informations personnelles, leur emploi du temps, le bloc de travail auquel ils sont affectés, et l'avion auquel ils sont assignés.
- **Réservation pour les Passagers :**
 - Les passagers ont la possibilité de consulter les vols disponibles, de choisir leur destination, d'effectuer des réservations et de générer un reçu pour leur vol.

III. Les classes :

Classe Admin :

- Attributs :

- username : Nom d'utilisateur de l'administrateur.
- password : Mot de passe de l'administrateur.
- nom : Nom de l'administrateur.
- prenom : Prénom de l'administrateur.

- Méthodes :

- ajouterEmploye() / modifierEmploye() / supprimerEmploye() : Ajoute un nouvel employé à la base de données avec les détails fournis, le modifier, le supprimer ou afficher ses information.
- ajouterVol() / modifierVol() / supprimerVol() : Ajoute un nouveau vol avec les détails associés, le modifier, le supprimer ou afficher ces information.
- ajouterAvion() / modifierAvion() / supprimerAvion() : Ajoute un nouvel avion avec les informations requise, la modifier, la supprimer ou afficher ses information.
- affecterEmploiDuTempsDepartement() : Affecte un emploi du temps à un employé, selon le departement de l'employe.
- genererRapportAeroport() : Génère un rapport complet sur l'état de l'aéroport, y compris les employés, les vols, les avions, etc.

III. Les classes :

Classe Employe :

- Attributs :

- username : Nom d'utilisateur de l'employé.
- password : Mot de passe de l'employé.
- nom : Nom de l'employé.
- prenom : Prénom de l'employé.
- salaire : Salaire de l'employé.
- role : Rôle de l'employé (pilote, attendance, technicien, etc.).
- departement : Le département de l'employé (Flight Stuff / Ground Stuff).

- Méthodes :

- consulterInformations() : Permet à l'employé de consulter ses informations personnelles, et les modifier.
- consulterAvionAffecte() : Affiche l'avion auquel l'employé est affecté.

III. Les classes :

Classe Passager :

- Attributs :

- username : Nom d'utilisateur de l'employé.
- password : Mot de passe de l'employé.
- nom : Nom de l'employé.
- prenom : Prénom de l'employé.
- num_passport : Numero de passport du client.

- Méthodes :

- consulterVolsDisponibles() : Permet au passager de consulter les vols disponibles.
- choisirVol() : Permet au passager de choisir son vol.
- effectuerPaiement() : Gère le processus de paiement.
- afficherTicket() : Génère une copie du reçu ou du ticket du vol.
- consulterInformations() : Permet au client de consulter ses informations personnelles, et les modifier.

III. Les classes :

Classe Avion :

- Attributs :
 - modele : Modèle de l'avion.
 - capaciteMax : Capacité maximale de passagers de l'avion.

Classe Avion :

- Attributs :
 - destination : Destination du vol.
 - origine : Lieu d'origine du vol
 - depart : Temps de depart du vol.
 - arrive : Temps d'arrive du vol.
 - avion : Avion assigné au vol.
 - pilote : Pilote assigné au vol.

VI. Base de Données pour l'Application de Gestion d'Aéroport avec MySQL :

4.1 Schéma de la Base de Données :

- Création d'une base de données nommée "GestionAeroportDB".
- *Définition des tables principales :*
 - **Admins** : Stocke les informations des administrateurs.
 - **Employes** : Contient les détails des employés.
 - **Passagers** : Enregistre les données des passagers.
 - **Avions** : Stocke les informations sur les avions.
 - **Vols** : Enregistre les détails des vols.

4.2 Relations entre les Tables :

- *Établir des relations appropriées entre les tables :*
 - Une relation entre Vol et Passager pour stocker les passagers d'un vol, avec le choix du classe du vol (business / économie).

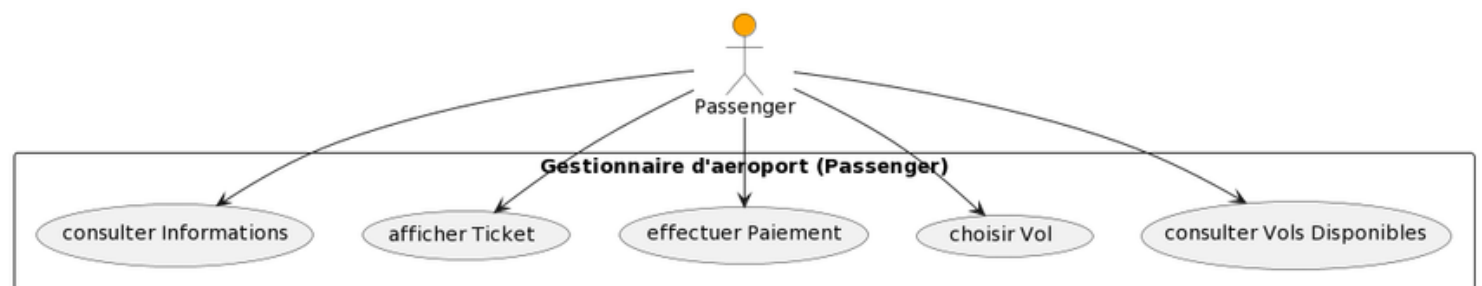
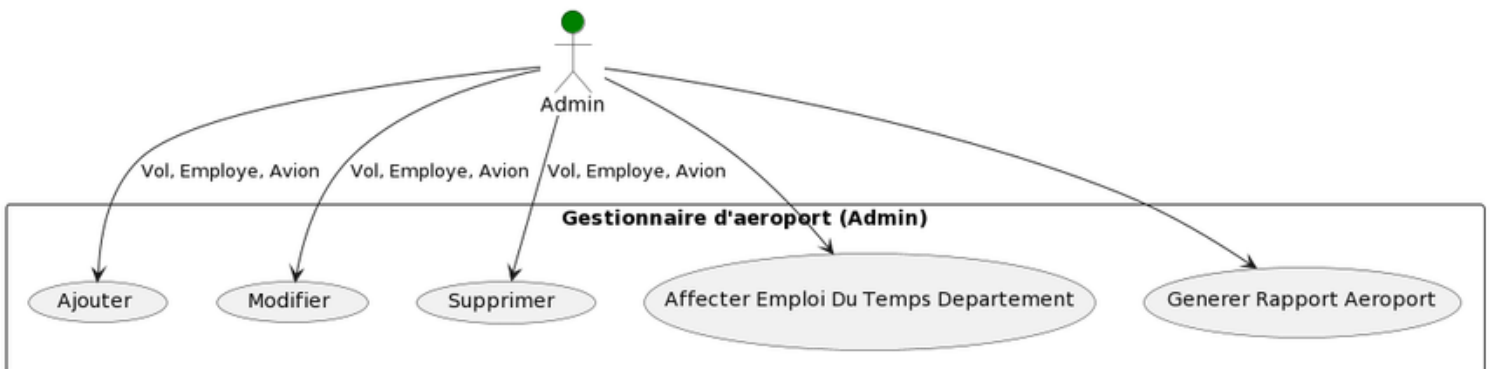
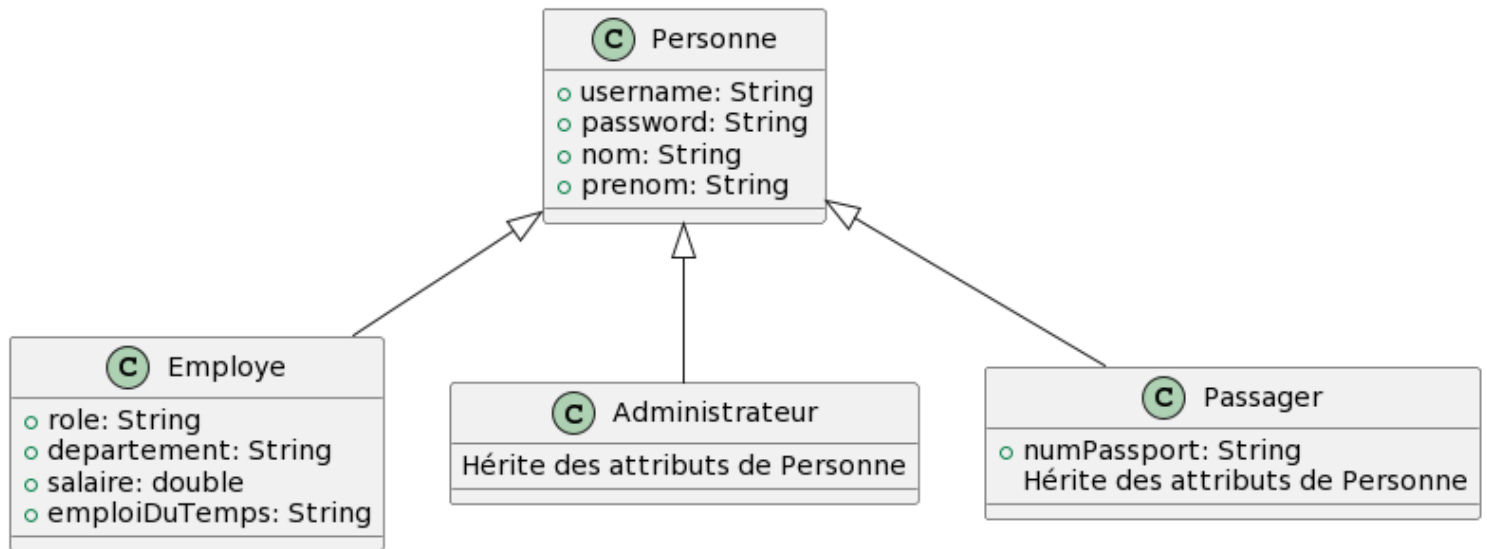
VI. Base de Données pour l'Application de Gestion d'Aéroport avec MySQL :

4.3 Attributs des Tables :

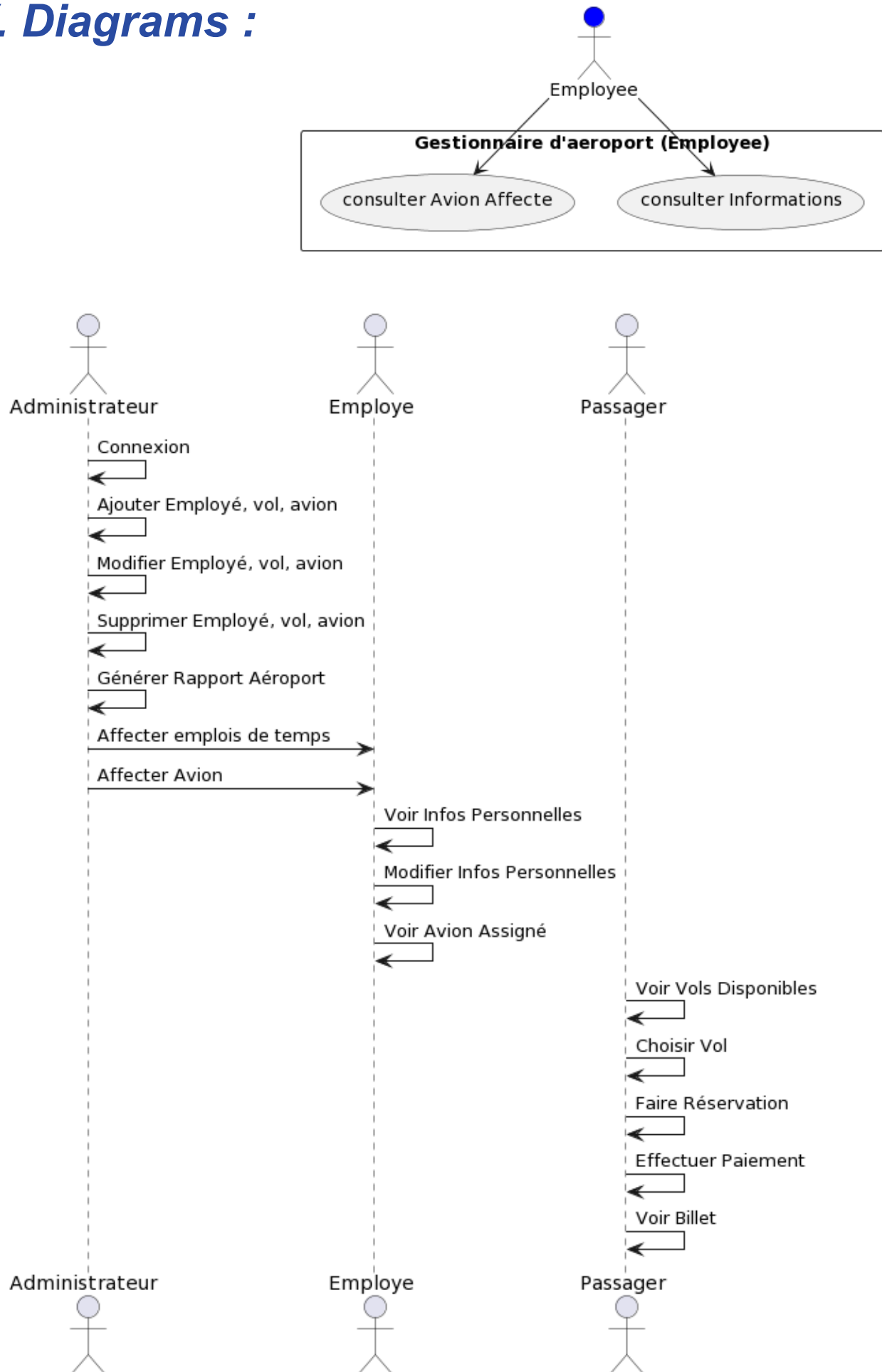
- *Table Admins :*
 - admin_id (clé primaire), username, password, nom, prenom.
- *Table Employes :*
 - employe_id (clé primaire), username, password, nom, prenom, role, emploi_du_temps.
- *Table Passagers :*
 - passager_id (clé primaire), username, password, nom, prenom, salaire, role, departement.
- *Table Avions :*
 - avion_id (clé primaire), modele, capacite_max.
- *Table Vols :*
- vol_id (clé primaire), destination, origine, pilote_id (clé étrangère), avion_id (clé étrangère), prix.

- *Table **Vol_passenger** :*
 - passenger_id (clé primaire), vol_id (clé étrangère), classe.

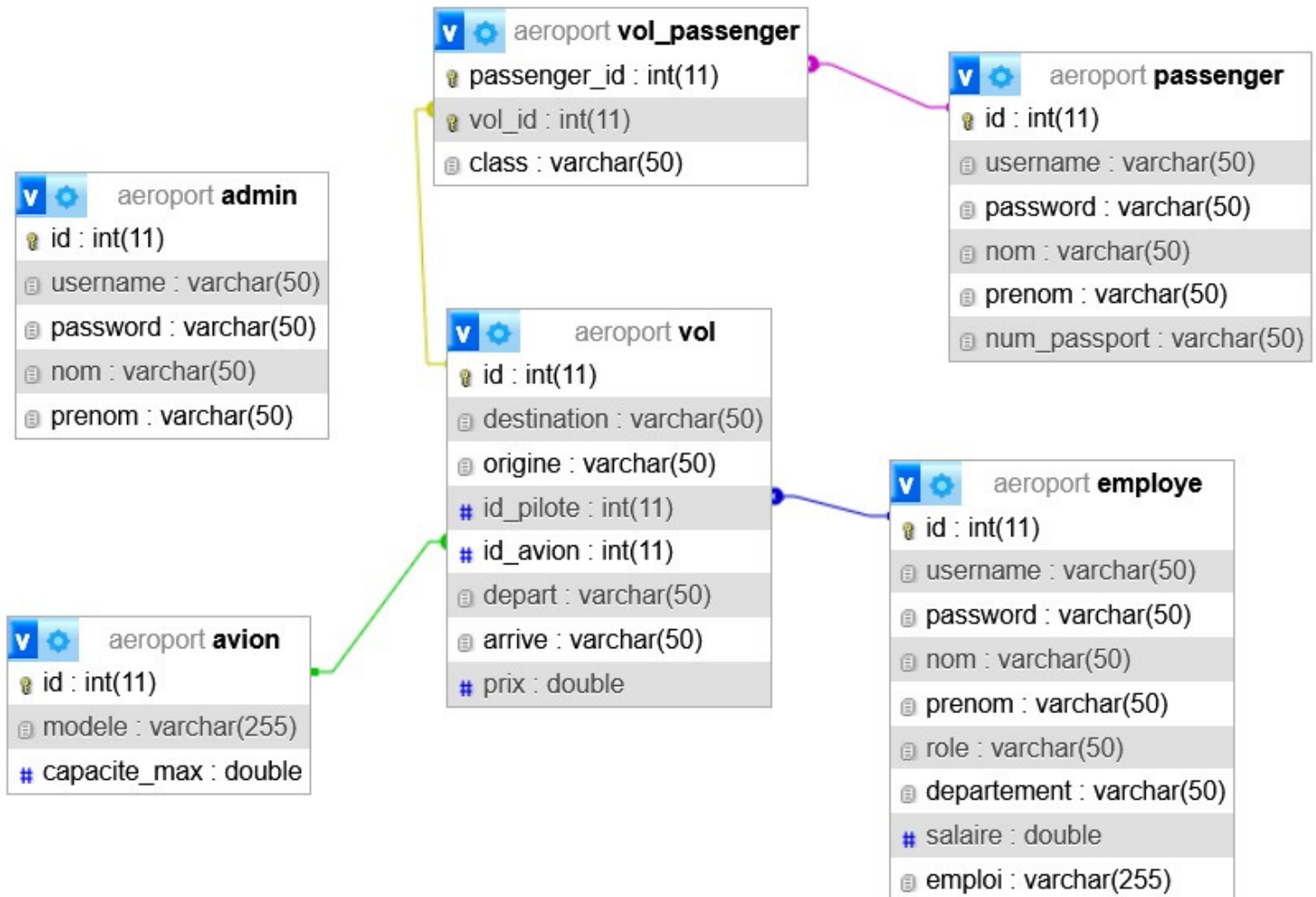
V. Diagrams :



V. Diagrams :



V. Diagrams :



VI. Conclusion :

En conclusion, l'élaboration de l'Application de Gestion d'Aéroport Arcane témoigne de la puissance et de la flexibilité du paradigme orienté objet, avec Java comme langage de programmation principal. L'utilisation judicieuse des concepts orientés objet dans le développement de Arcane a permis de créer une structure modulaire, extensible et facile à maintenir

Exploitation des Principes de LPOO dans Arcane :

- Encapsulation : Les classes telles que Admin, Employe, Passager, Avion, et Vol encapsulent les données et les fonctionnalités qui leur sont spécifiques, renforçant ainsi la sécurité et la modularité du code.
- Héritage : L'utilisation de l'héritage est apparente dans la création de sous-classes spécifiques pour chaque type d'utilisateur (Admin, Employe, Passager), permettant la réutilisation du code tout en respectant les spécificités de chaque entité.
- Abstraction : Les différentes classes et méthodes dans Arcane représentent des abstractions efficaces des entités réelles impliquées dans la gestion aéroportuaire, simplifiant ainsi la complexité du système.