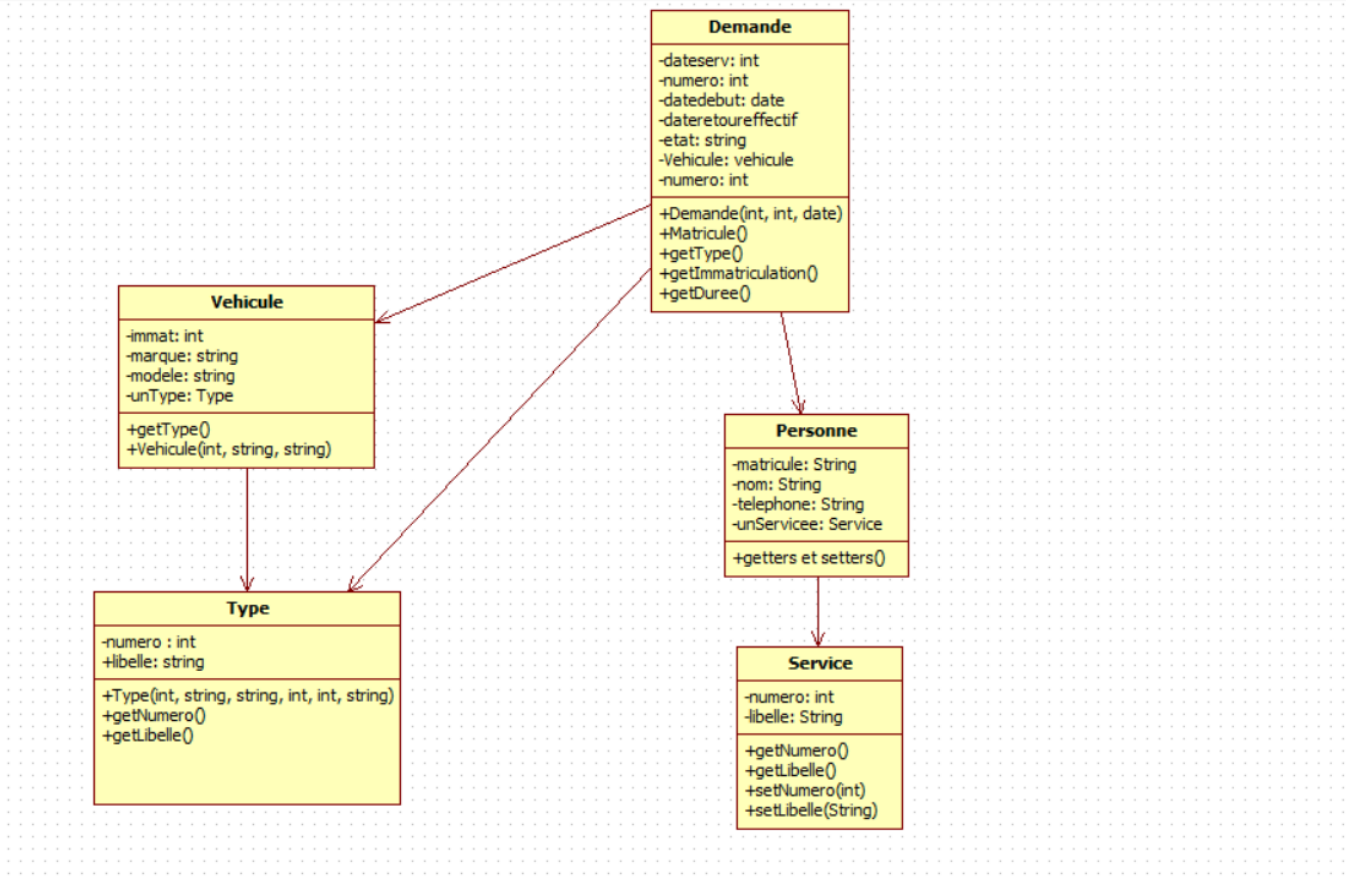
*type(**numero**, libelle)*
*vehicule(**immat**, marque, modele, #notype)*
*service(**numero**, libelle)*
*personne(**matricule**, nom, telephone, #noservice)*
*demande(**datereserv,numero**, datedebut, #matricule, #notype, #immat, duree, dateretoureffectif, etat)*

Diagramme de classes



Pour concevoir l'application, nous avons repris les tables de la base de donnés ci dessus. Pour le code Java, nous avons créé des classes métier qui correspondent à chaque table. Les colonnes d'une table se transforment en attributs de la classe qu'elles illustrent.

Le rôle des classes métier regrouper toute la logique métier dans l'application

Les classes métier (fichiers java) :

- **Demande.java**
- **Personne.java**
- **Service.java**
- **Type.java**

Aymane – Ayyoub – Rayan

- **Vehicule.java**

La classe liée aux données :

- **Passerelle**

L'affichage :

- **Menu**

Exécution de l'application :

- **App.java**

Description des classes métier :

Demande :
La classe Demande va permettre à une personne d'effectuer une demande de réservation d'un véhicule.
Elle contient un constructeur , des attributs : datereserv, numero , datedebut , un objet personne de type personne , le numero du type , un objet véhicule de type véhicule , la durée , la date de retoureffectif , son etat ; ainsi que des accesseurs et mutateurs pour chaque attribut, nécessaire pour consulter ou modifier une demande de réservtion.

Type :
La classe *Type* identifie le type de véhicule (utilitaire ,citadine, camionnette) .

Vehicule :
La classe *Vehicule* représente un véhicule et permet de vérifier sa disponibilité .Elle est utilisée lors de la validation d'une demande de reservation par le service gestionVehicules.

Service
Représente le service auquel appartient une personne.

Personne
Représente un utilisateur (employé) pouvant effectuer une demande de réservation.

Classes d'accès aux données :

Passerelle :
Son rôle est de séparer la logique métier et l'accès aux données, afin de communiquer avec la base de données.

Les fonctions :

- **Connexion à la BDD :** public Passerelle()
  Fonction qui permet de se connecter à la base de données avec trois paramètres : l'URL de connexion, le nom d'utilisateur et le mot de passe.
- **Vérifier la connexion :** public boolean verifierConnexion(int matricule, String mdp)
- **Récupérer le numéro du type :** public Type recupererTypeParNumero(int numero)
- **Modifier une réservation :**
  public void modifierReservation(int numero, LocalDate dateReserv, LocalDate dateDebut, String matricule, int noType, String immat, int duree, LocalDate dateRetourEffectif, String etat)
- **Valider un véhicule :**
  public boolean verifierReservation(String marque, String modele, Type unType, String immat)
  Cette fonction permet de valider une demande de réservation et de choisir un véhicule disponible ou non selon les valeurs passées en paramètres : la marque, le modèle, le type de véhicule et l'immatriculation.

Le menu :

Les fonctions :

- **Fonction d'affichage du menu :** public void afficherMenu(Scanner scanner, Passerelle db)
  En fonction du choix, la personne peut :
  - Faire une réservation d'un ou plusieurs véhicules
  - Vérifier la disponibilité d'un véhicule
  - Modifier une réservation
  - Quitter
- **Fonction de connexion au menu :** public boolean menuConnexion(Scanner s, Passerelle db)
- **Fonction du menu de réservation :** public void menuReservation(Scanner s)
- **Fonction du menu de vérification de disponibilité du véhicule :**
  public void menuVerificationDisponibilite(Scanner s, Passerelle db)
  Cette fonction permet de vérifier la disponibilité du véhicule.
- **Fonction d'affichage du nombre de véhicules réservés :**
  public void menuResumeFinal()