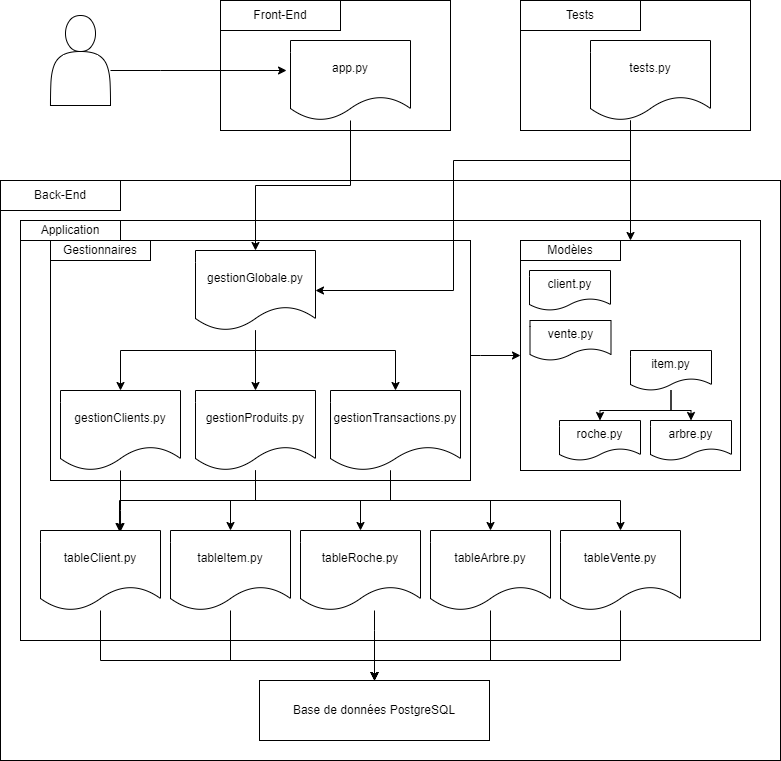
# Modèle Architecture

Les technologies utilisées pour ce projet sont les suivantes :

* Python
* PostgreSQL

L’architecture utilisé pour ce projet est composée des éléments suivants :

* Front-end
* Back-end
  + Application
    - Gestionnaires
      * Global
      * Spécialisé
        + gestionClient.py
        + gestionProduit.py
        + gestionTransaction.py
      * de Table
        + tableClient.py
        + tableItem.py
        + tableArbre.py
        + tableRoche.py
        + tableVente.py
    - Modèle
      * client.py
      * item.py
      * arbre.py
      * roche.py
      * vente.py
    - constantes.py
  + Base de données PostgreSQL
* Tests
  + Test.py



## Front-End

Le front-end consiste en un simple CLI (Command Line Interface) qui permet de communiquer avec le gestionnaire global de la boutique.

Les commandes sont interprétées par le module « argparse » natif de Python.

Le module gère la connexion initiale à la base de données, qu’il utilise pour initialiser le gestionnaire global du programme. Il n’a le droit d’appeler que les fonctions mises disponible dans la couche de Niveau 1 du backend, soit le gestionnaire global. Le module n’a aucune connaissance du fonctionnement du backend.

Pour un front-end plus sophistiqué, la connaissance des modèles d’objet pourrait être utile, mais ce n’est pas nécessaire dans sa forme actuelle.

## Back-End

Le back-end utilise pour la partie application un modèle de couche à 3 niveaux :

**Niveau 1** : Le gestionnaire global est le seul point d’entré de l’application pour le front-end. C’est dans ce module que l’interface se connecte.

**Niveau 2** : Les gestionnaires spécialisé se concentre sur les transactions à la base de données dans leur domaines respectifs. Ils offrent les fonctions que le gestionnaire global peut utiliser. Ils traduisent la représentation interne de la base de données en objet modèle manipulable par la couche supérieure.

**Niveau 3** : Les gestionnaires de table ont pour seul rôle d’effectuer des requêtes à la base de données PostgreSQL et d’offrir leur service aux gestionnaires spécialisés.

**Modèles**: Les modèles sont les objets Python manipulés par les gestionnaires et sont calqué sur la structure de la base de données.

## Tests

Les tests ont accès aux objets modèles ainsi qu’à tous les modules de l’application via le gestionnaire global qui donne accès à tous les autres module.