

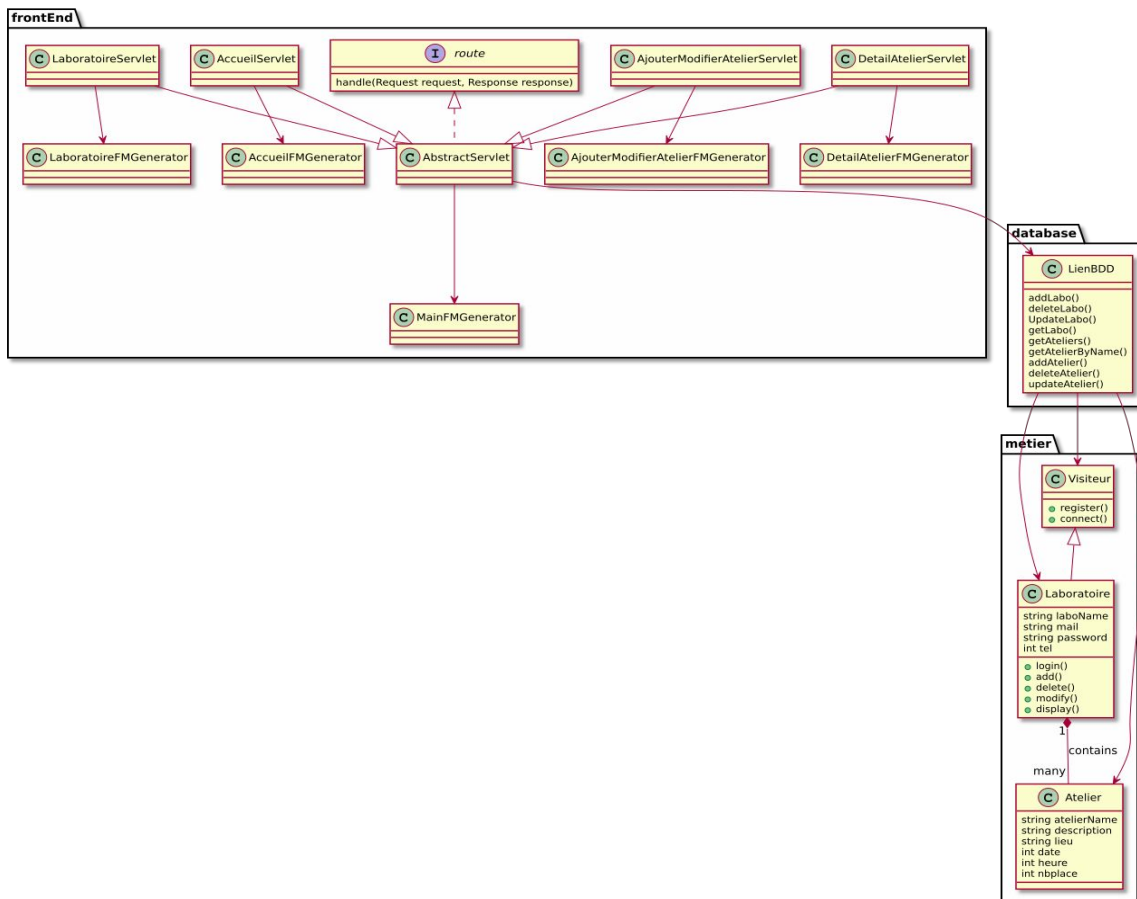
## Conception

On se base sur une architecture du type Client-Server avec une architecture 3 tiers.

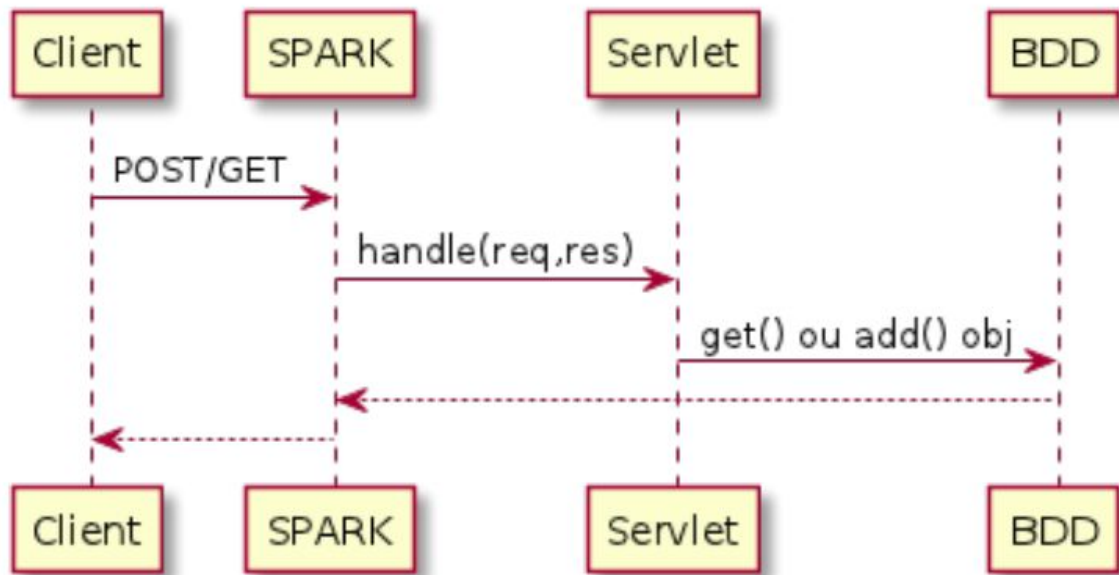
1. Les clients sont représentés dans cette première partie par le front-end c'est à dire les interactions avec les utilisateurs : laboratoires, les visiteurs et les ateliers.
2. Le serveur est celui qui gérera toutes les requêtes du client. Le serveur reçoit des requêtes du client, il les traite et fait des requêtes si besoin à la base de données
3. La base de données reçoit des requêtes de la part du serveur. Elle lui répond.



D'un point de vue de l'architecture statique de notre projet, nous avons réalisé le diagramme de classe suivant :



Un diagramme séquentiel détaille le fonctionnement des servlet des requêtes du client aux accès à la base de donnée :

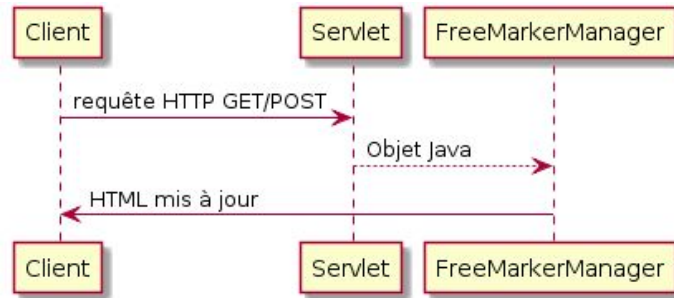


Les vues sont les suivantes :

- la vue accueil qui contient les formulaires de connexion et d'inscription
- la vue laboratoire qui contient la liste des ateliers d'un labo
- la vue detailAtelier qui contient les informations détaillées d'un atelier
- la vue ajouterModifierAtelier qui permettent d'ajouter un atelier ou de modifier un atelier existant.

On utilise le moteur de templates "FreeMarker" qui permet de générer automatiquement des fichiers html à partir d'un modèle de données (fichier java + template). 4 classes Java liées à leur Servlet correspondantes formeront les modèles de données pour les 4 fichiers HTML définissant les vues.

Un dernier fichier HTML sera généré via FreeMarker par le biais de mainFMGenerator et servira de base aux autres fichiers HTML (header, menu et footer).



Le diagramme d'activité suivant est un exemple qui détaille les étapes réalisées par la fonction handle dans la servlet AccueilServlet :

