Introduction:

Dans le cadre de nos études, nous avons pour projet la création d'un site web complet.
Afin de développer nos compétences et nos connaissances nous allons recréer le site Airbnb sous le nom de "Hbnb".

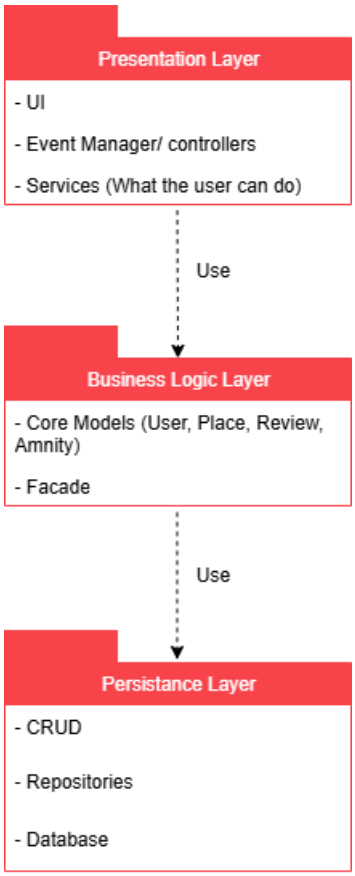
L'objectif de cet exercice est de comprendre et de représenter clairement l'architecture et le fonctionnement interne d'un site web sur le plan technique.
Ce document servira de référence pour la structure du site, facilitant son développement.

Vous retrouverez donc ci-dessous plusieurs diagrammes illustrant les interactions entre les différents éléments du système:

- Diagramme de package
- Diagramme de classes
- Diagramme de séquence

1 -Le diagramme de package:

Ce diagramme représente l'architecture du système dans son ensemble. Il illustre les différentes interactions entre les différentes couches et à quoi elles correspondent.



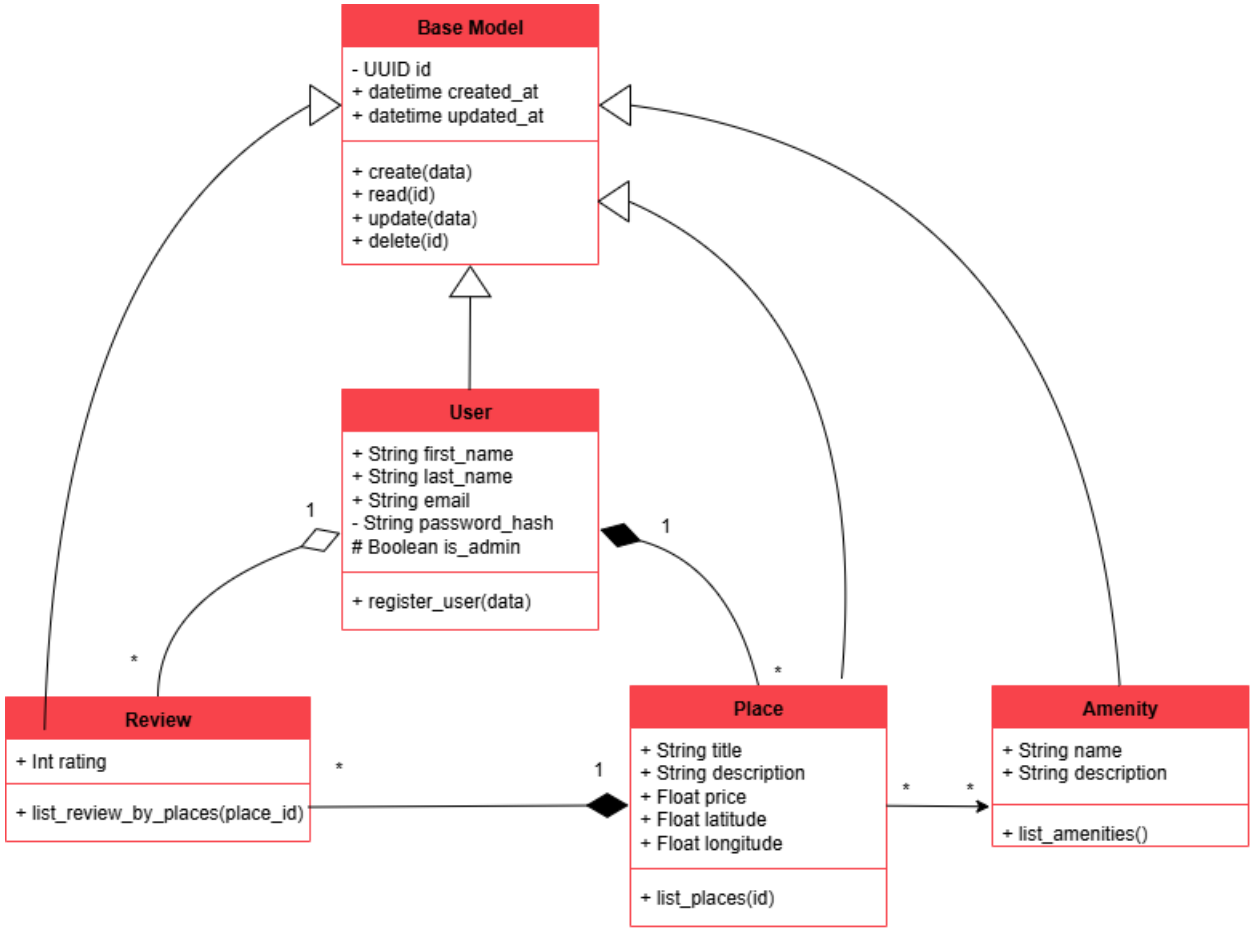
- Presentation layer:
Gère l'interface utilisateur et les contrôleurs API

- Business Layer:
Contient la logique métier, les règles de validation ainsi que les modèles de classes principales pour le bon fonctionnement de l'application.

- Persistance layer:
Gère le contrôle et l'accès des informations dans la base de données via des dépôts.

2- Le diagramme de classes:

Ce diagramme détaille les entités principales du domaine, ainsi que leurs relations.
Il encapsule les règles métiers qui servent à vérifier si les données sont conformes à une demande.



- Base Model: Contient les attributs et les méthodes communes à toutes les classes pour simplifier la lisibilité et la maintenance.
Toutes les autres classes en héritent.

- User:
Représente l'utilisateur et tous ses attributs (nom, prénom, mail...). Certaines de ces données sont privés comme son mot de passe.
Un utilisateur peut avoir plusieurs logement ou plusieurs avis.
L'utilisateur peut mettre à jour son profil comme il le souhaite via les méthodes héritées du base model, qui vont vérifier si les demandes sont valides, avant de communiquer la requête auprès de la couche de persistance.

- Place:
Le logement est lié à un utilisateur qui en est le propriétaire.
Il a ses propres attributs, comme un nom, une description ou encore un prix.
Il peut avoir plusieurs commodités.
Il y a une relation de composition, si le propriétaire disparait, le logement est également supprimé.

Tout comme l'utilisateur, les données du logement peuvent être modifiées ou supprimées de la même manière.

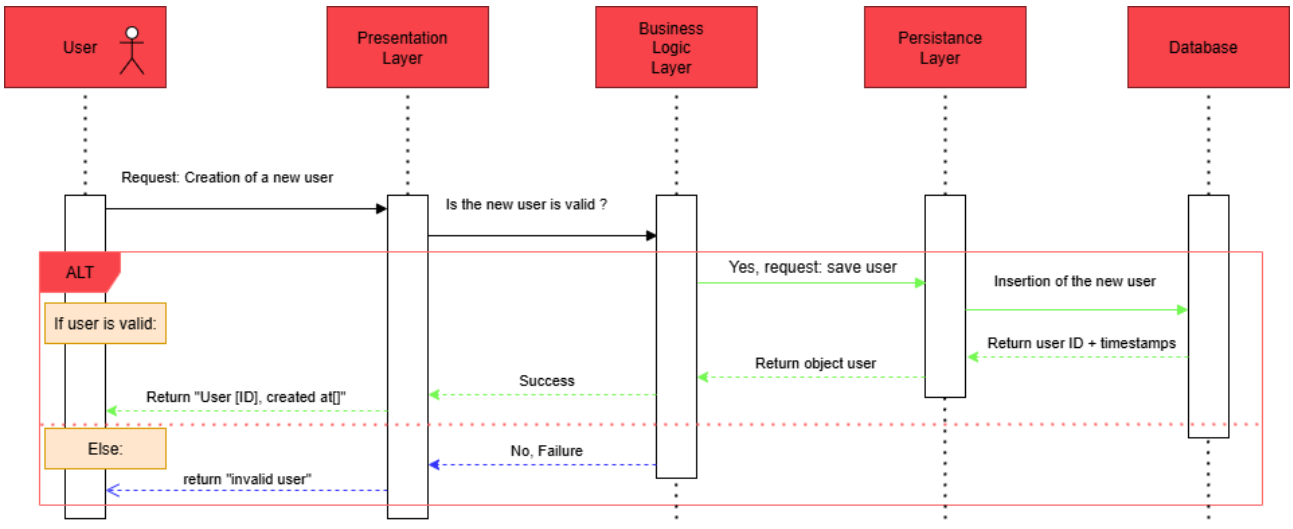
- Amenities:
Contient tout les commodités disponibles (piscine, wifi...).
Cette classe vient souvent compléter la description d'un logement en y apportant plus de précision.
Les commodités peuvent être les mêmes sur différents logements.
Elles ont des données modifiables de la même manière que les deux classes précédentes.

- Review:
Un utilisateur peut en créer plusieurs, et il peut y avoir plusieurs avis sur un seul logement.
Cela correspond aux avis laissés sur un logement. Si celui-ci est supprimé, les avis disparaissent car il y a une relation de composition.
Cependant si un utilisateur supprime son compte, son avis ne sera pas supprimé tant que la page du logement existera, car c'est une relation d'agrégation. L'utilisateur devra demander à le modifier ou à le supprimer avant.

3- Les diagrammes de séquence:

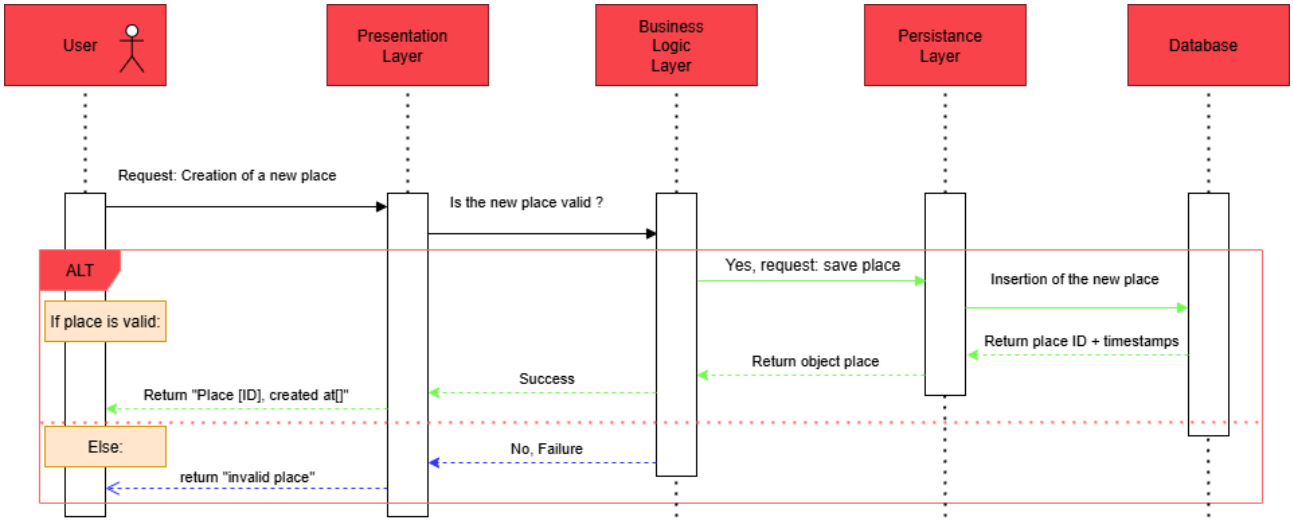
Illustre les flux d'interactions du système entre les différents composants lorsqu'une requête est formulée.
Voici donc quatre diagrammes de séquence avec quatre scénarios différents:

Demande de création d'un utilisateur:



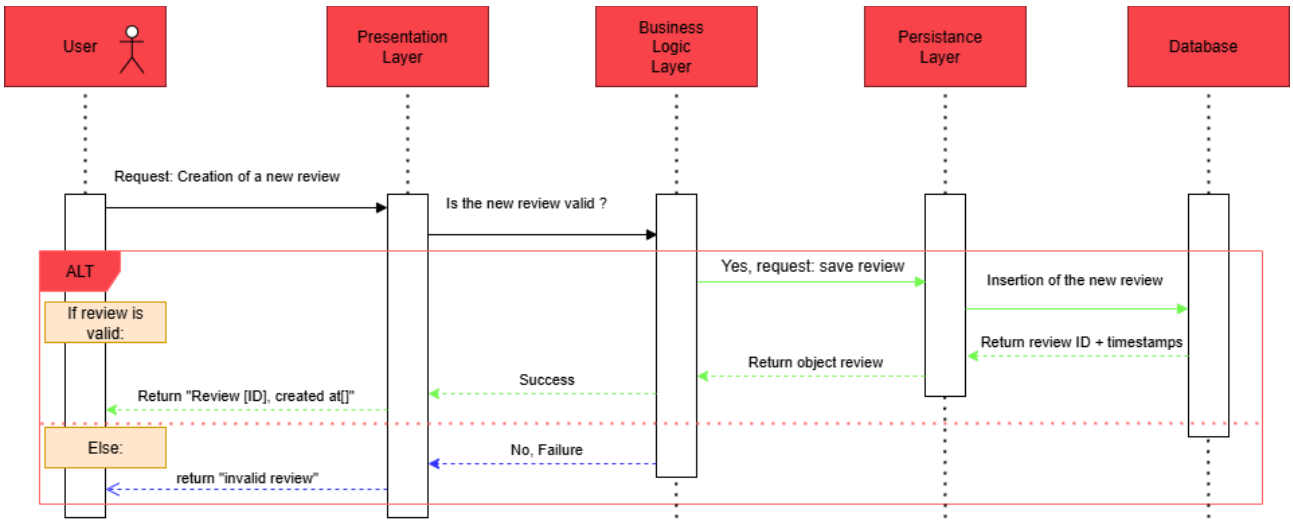
Dans ce diagramme, l'utilisateur demande à créer un nouveau compte. Il va faire une demande à l'API (dans le presentation layer), qui va demander au business logic layer de vérifier si la requête et les données sont valides. Si elles ne le sont pas, le système va renvoyer une erreur.
Si la requête et les données sont valides, alors elles sont envoyées au persistance layer qui va insérer un nouvel utilisateur dans la base de données.
Quand c'est fait, la base données renvoie l'identifiant (ID) de l'utilisateur, ainsi que l'heure de sa création.

Demande de création d'un nouveau logement:



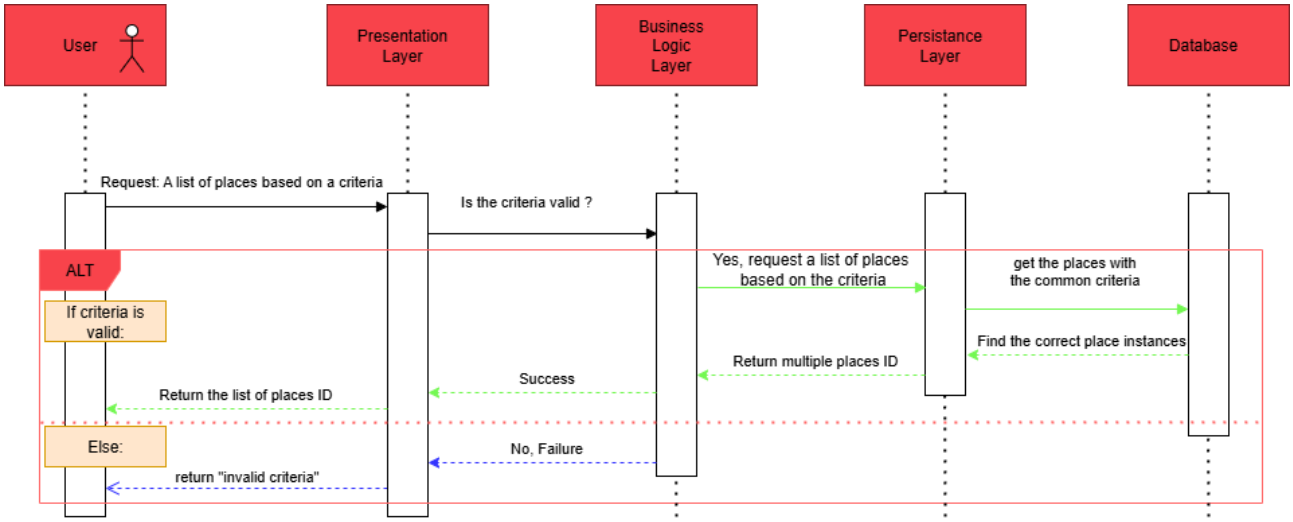
Dans ce diagramme, l'utilisateur demande à créer un nouveau logement. Il va faire une demande à l'API (dans le presentation layer), qui va demander au business logic layer de vérifier si la requête et les données sont valides. Si elles ne le sont pas, le système va renvoyer une erreur.
Si la requête et les données sont valides, alors elles sont envoyées au persistance layer qui va insérer un nouveau logement dans la base de données.
Quand c'est fait, la base données renvoie l'identifiant (ID) du logement, ainsi que l'heure de sa création.

Demande de création d'un avis:



Dans ce diagramme, l'utilisateur demande à créer un nouvel avis sur un logement. Il va faire une demande à l'API (dans le presentation layer), qui va demander au business logic layer de vérifier si la requête et les données sont valides. Si elles ne le sont pas, le système va renvoyer une erreur.
Si la requête et les données sont valides, alors elles sont envoyées au persistance layer qui va insérer un nouvel avis dans la base de données. Quand c'est fait, elle renvoie l'identifiant (ID) du logement, ainsi que l'heure de sa création.

Demande d'une liste de logements basée sur un critère:



Dans ce diagramme, l'utilisateur demande une liste de logements, basée sur un critère. Il va faire une demande à l'API (dans le presentation layer), qui va demander au business logic layer de vérifier si la requête et les données sont valides. Si elles ne le sont pas, le système va renvoyer une erreur.
Si la requête et les données sont valides, alors elles sont envoyées au persistance layer qui va chercher une correspondance avec le critère, dans les logements présents dans la base de données. Quand c'est fait, elle renvoie toutes les instances correspondantes, et renvoie une liste d'ID (d'identifiants) de toutes ces instances.

Conclusion:

Ce projet nous permet de mieux comprendre comment se structure un site web et comment illustrer au mieux son architecture, de façon claire et détaillée.
Cette documentation est essentielle pour mener à bien un projet, et pour la rendre plus maintenable à l'avenir.

Auteurs:

Antoine Coquemont
Maï Le Meur
Joëvin Manceau