

 **La Plateforme**

La grande école du numérique pour tous

Overlook Hotel – Kickoff



Sujet



Tandis que votre maîtrise de Java s'intensifie, une nouvelle aventure s'ouvre à vous : la gestion d'un hôtel. Vous plongerez dans le monde fascinant de la gestion hôtelière en exploitant la puissance du framework Spring.

 **La Plateforme**

La grande école du numérique pour tous

Fonctionnalités



Mise en place de l'environnement et conception

A. Configuration de Spring Boot

→ **Initialisez** un projet **Spring Boot** avec les **dépendances** nécessaires.

B. Conception de la base de données

→ **Modélisez** des **entités** pour les chambres, les réservations, les clients, les employés et les gestionnaires.

→ **Établissez** des **relations** appropriées entre ces entités.

→ Vous devrez fournir votre modélisation dans votre **drive**



 **La Plateforme**

La grande école du numérique pour tous

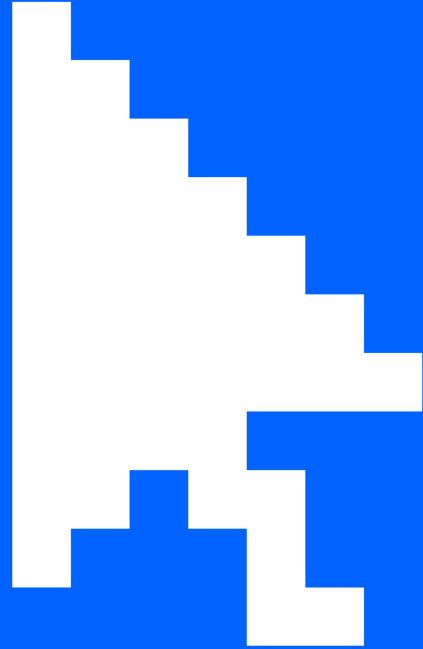
MODÉLISATION D'ENTITÉS

Solutions entreprises

La méthode MERISE ?

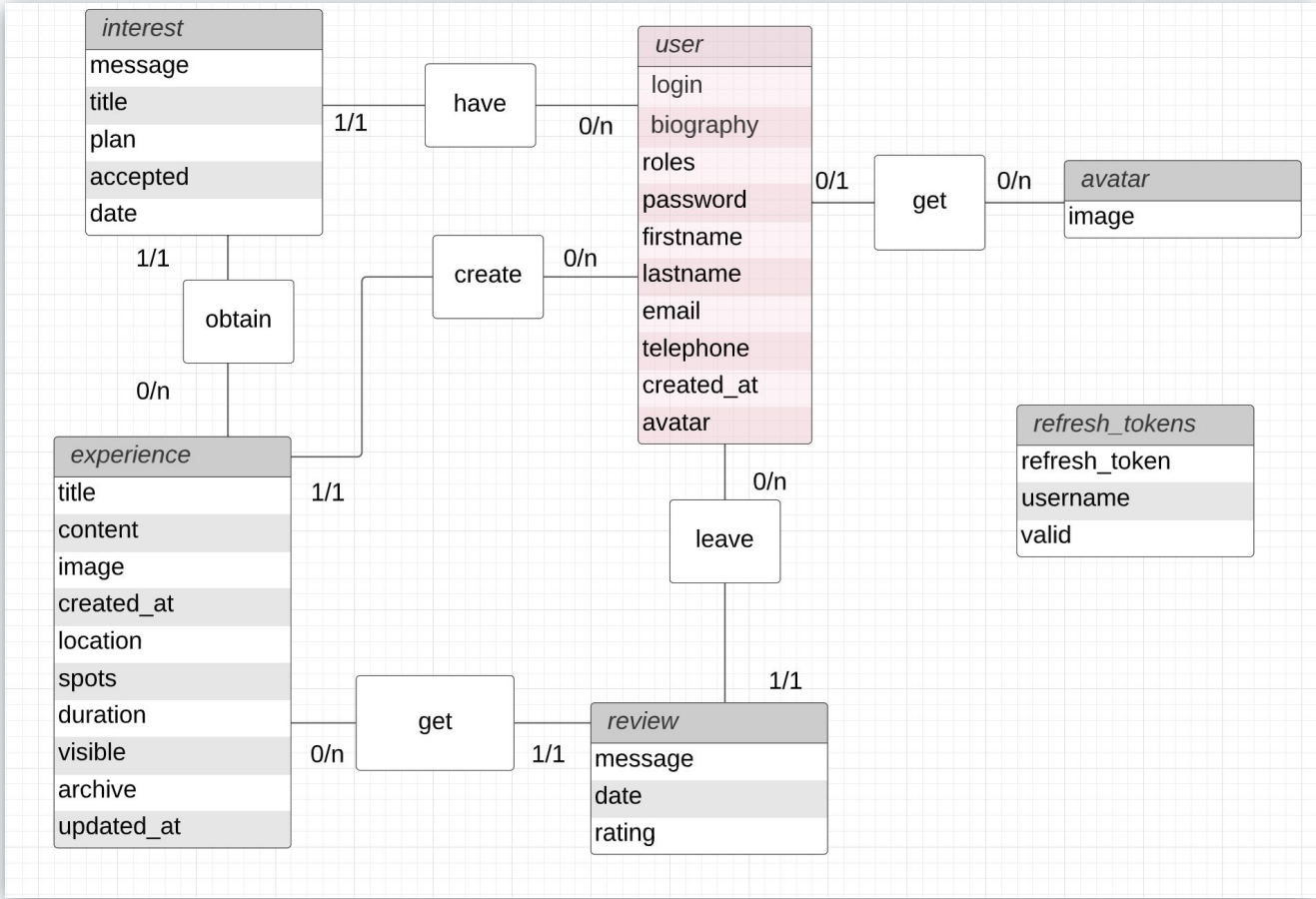
MERISE est une méthode d'analyse et de conception des systèmes d'information basée sur le principe de la séparation des données et des traitements.

Modèles répartis sur 3 niveaux (le niveau conceptuel (MCD), le niveau logique (MLD) le niveau physique (MPD)).



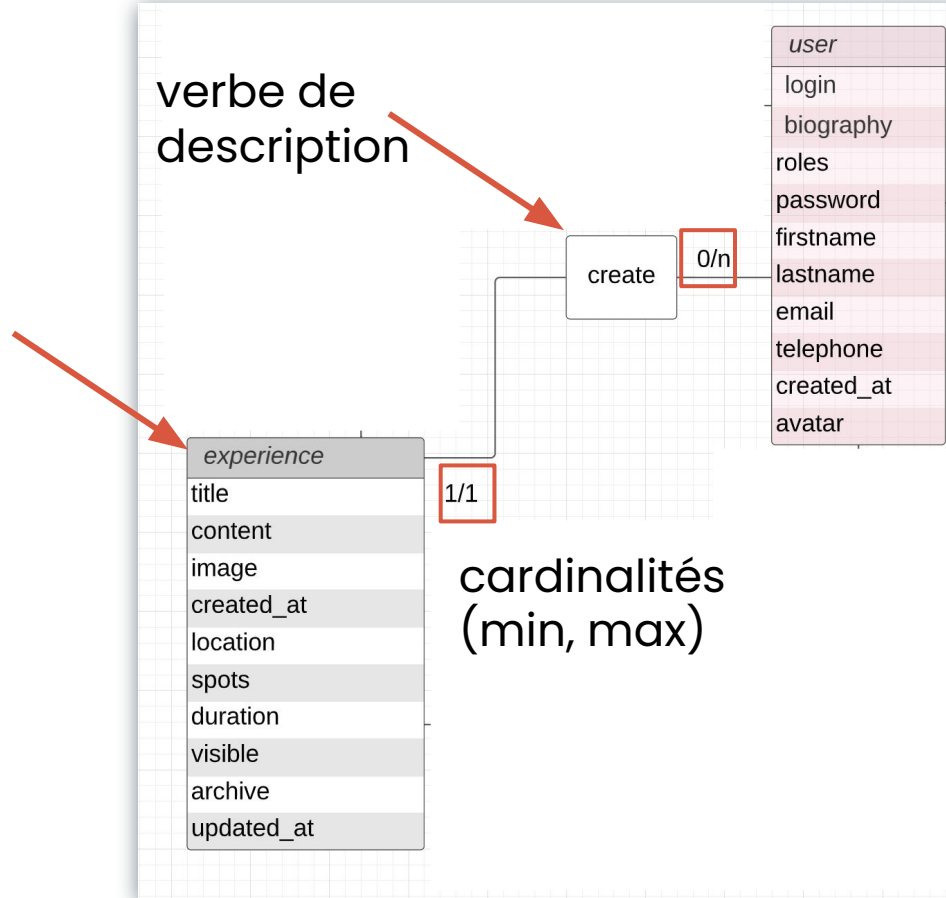
MCD : Modèle conceptuel de données

Représentation graphique et facilement compréhensible des données et de leurs liens.



entité

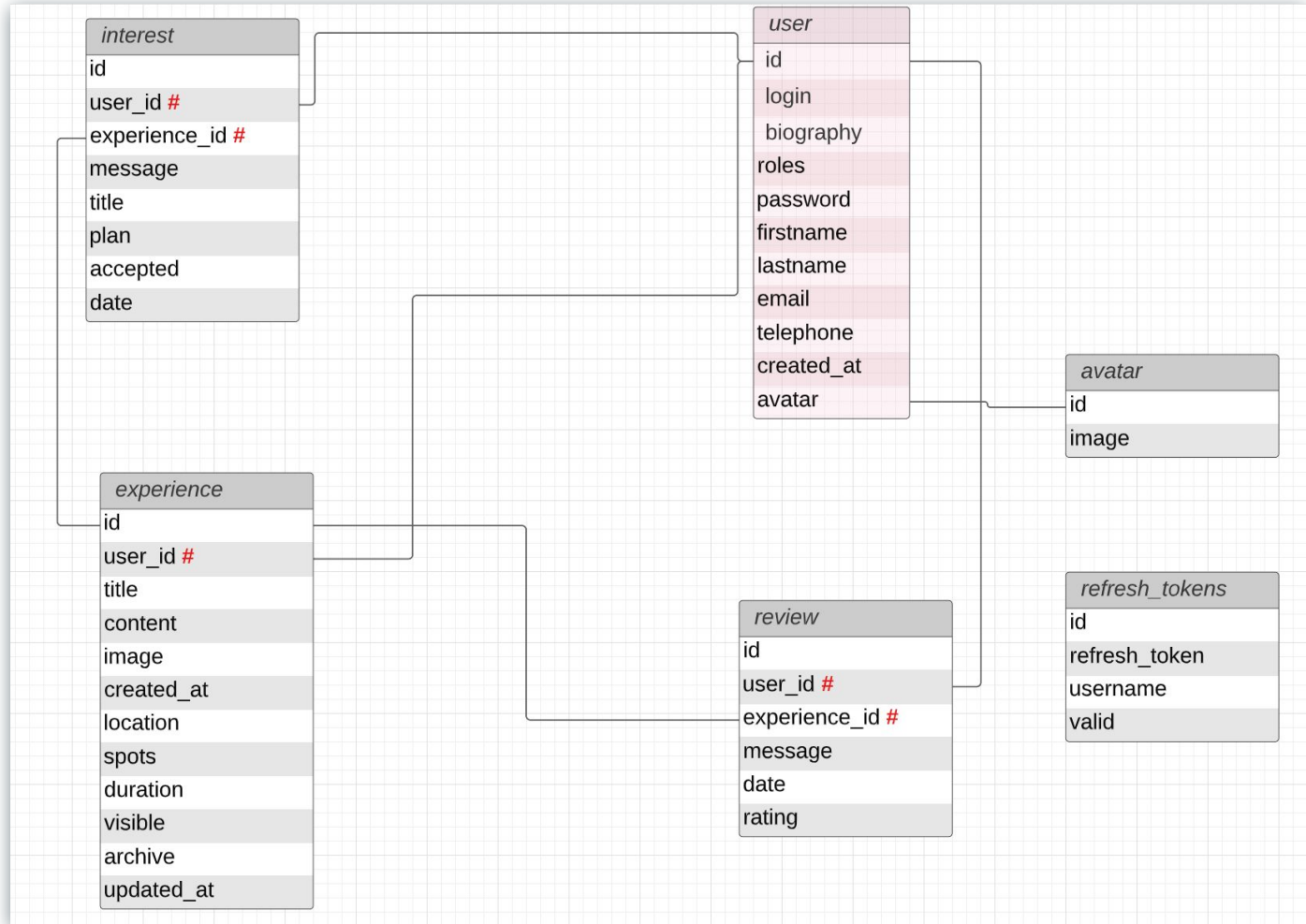
verbe de
description



- Pas d'Id primaire
- Ni de clé étrangère
- Pas d'affichage du type des données

MLD : Modèle Logique de données

Modèle qui se rapproche de la structure réelle d'une base de données.



clé primaire
=
donnée unique

experience	
id	
user_id #	
title	
content	
image	
created_at	
location	
spots	
duration	
visible	
archive	
updated_at	

clé étrangère
=
établit et conserve
liaison entre deux
tables

user	
id	
login	
biography	
roles	
password	
firstname	
lastname	
email	
telephone	
created_at	
avatar	

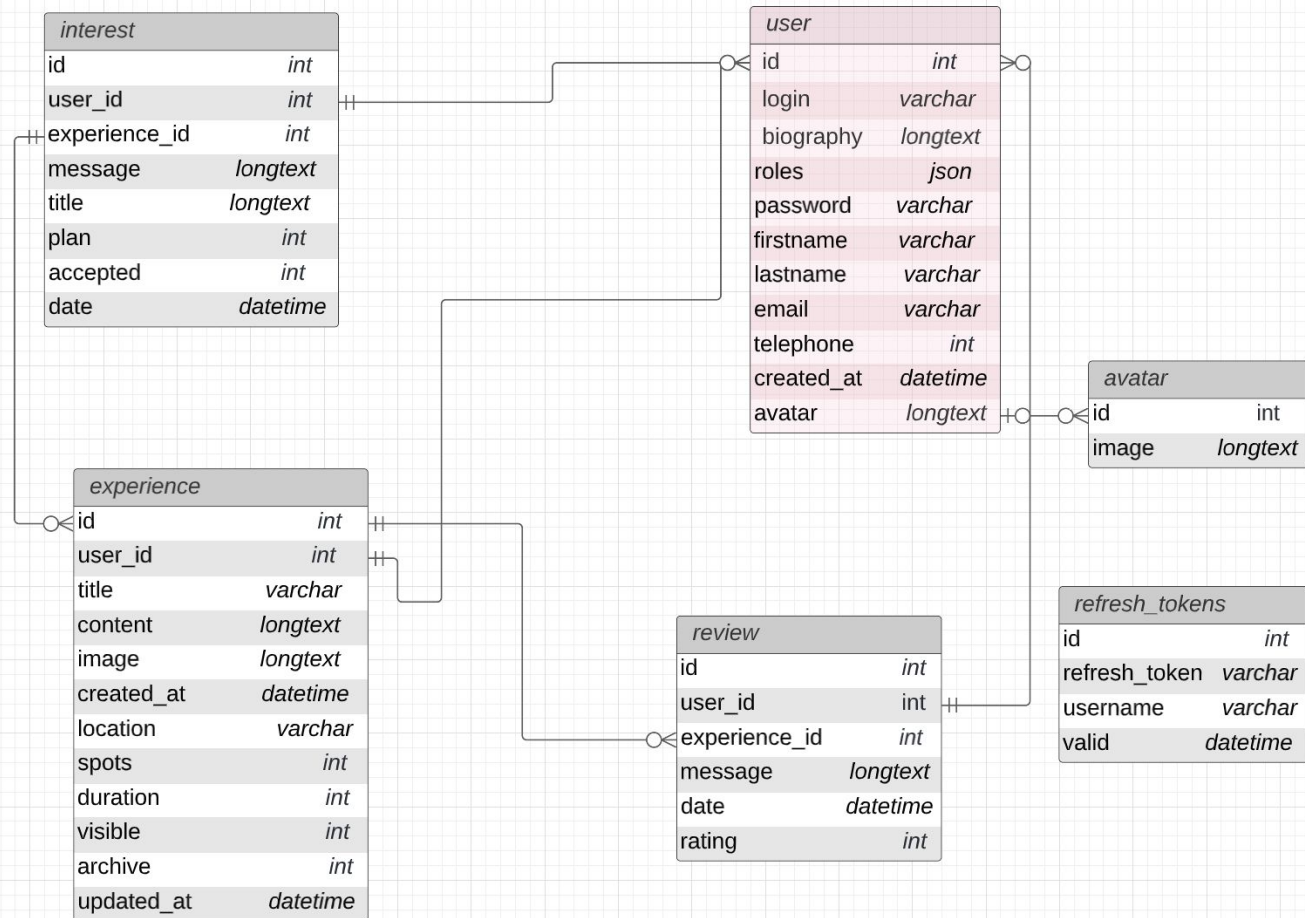
- Suppression des cardinalités et des verbes
- Relations entre les clés
- On enlève les cardinalités
- Si relations n..n n..n dans MCD, ajouter une table de liaisons

MPD : Modèle physique de données

encore plus proche
de la structure
finale de la base de
données.

Il **modélise** et
décrit

complètement son
architecture



tous les types

tous les
champs

experience	
id	int
user_id	int
title	varchar
content	longtext
image	longtext
created_at	datetime
location	varchar
spots	int
duration	int
visible	int
archive	int
updated_at	datetime

pattes
d'oie

user	
id	int
login	varchar
biography	longtext
roles	json
password	varchar
firstname	varchar
lastname	varchar
email	varchar
telephone	int
created_at	datetime
avatar	longtext



La Plateforme

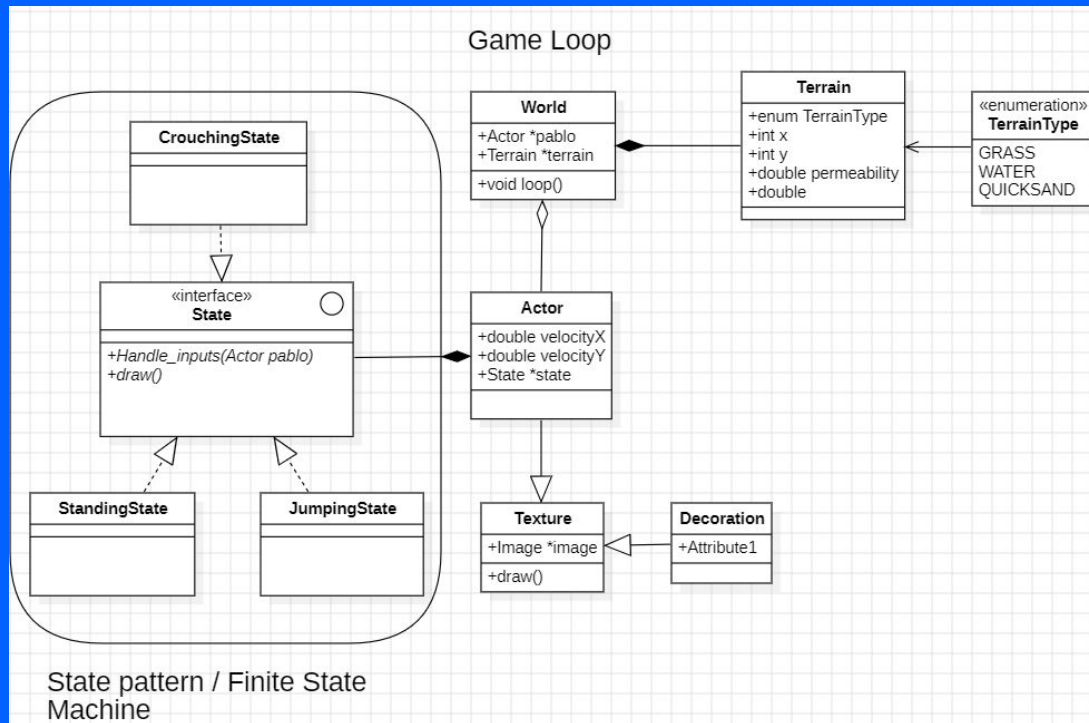
La grande école du numérique pour tous

MODÉLISATION DE DONNÉES

Solutions entreprises

/ UML, c'est quoi ?

- Unified Model Language
- Langage graphique
- Permet de créer plusieurs types de diagrammes, comme :
 - Classes
 - Tables
 - Etats
 - Cas d'utilisations





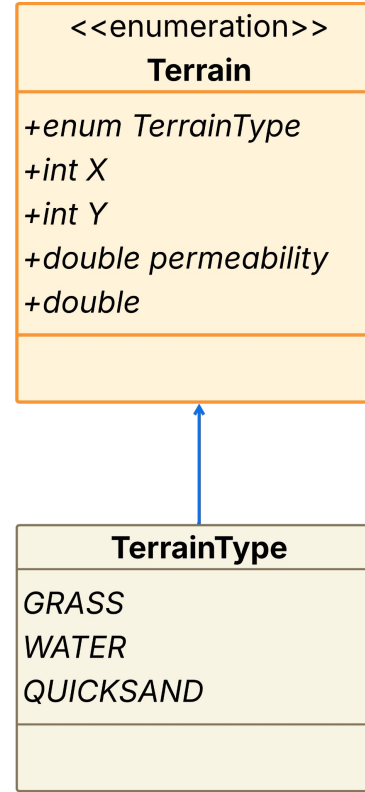
 **La Plateforme**

La grande école du numérique pour tous

Relations

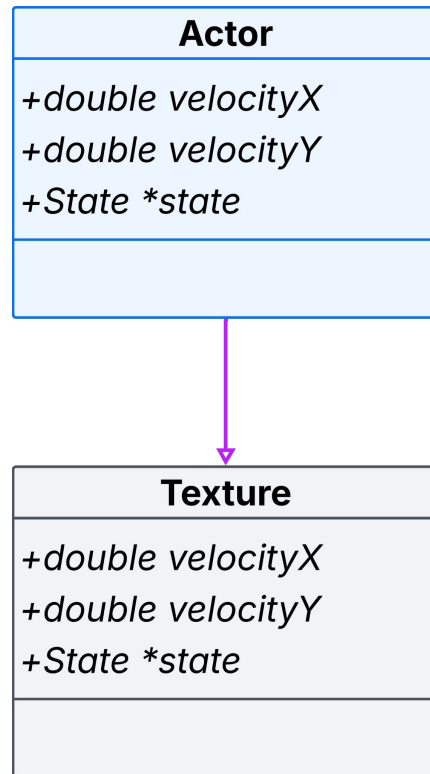
Généralisation / Héritage

Représentation des relations classe parent et classe enfant. Une relation de type **"kind-of" (est de type de)**.



Association simple

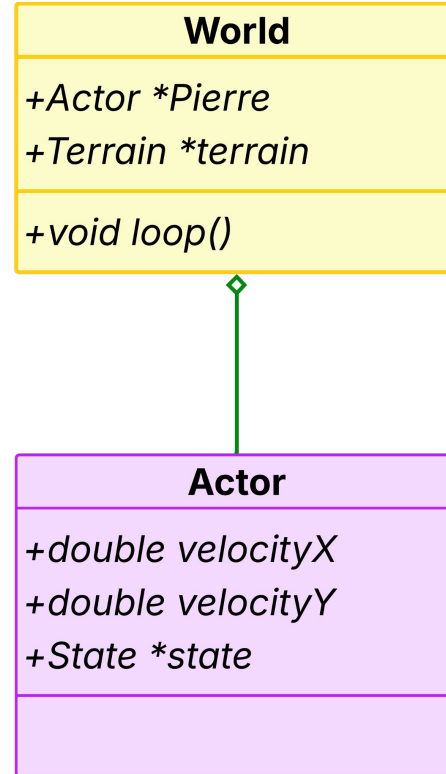
Relation qui **relie deux classes**
ou plus, indiquant qu'il existe
une connexion ou une
interaction entre elles



Agrégation

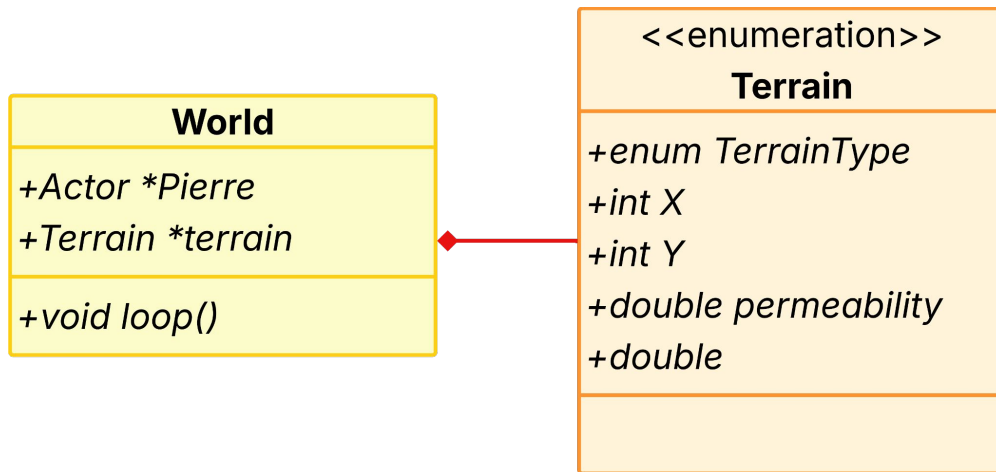
Type **d'association** plus spécifique qui représente la relation "**part-of/**" (**partie-de**) dans les diagrammes.

Actor fait partie de World , mais les deux sont indépendants.



Composition

Représenter la
dépendance des objets
vis-à-vis de l'entité
focale.



Les Terrains font partie de World. Si World est *détruit*, les *instances* de Terrain qui le *composent* le sont aussi.

Système d'authentification et de rôles

A. Backend

→ Mettez en place un **système d'authentification**.

→ Implémentez des **rôles distincts** pour les clients et les gestionnaires, avec des autorisations appropriées pour chaque rôle.

B. Frontend

→ Pages de **connexion distinctes** pour les clients et les gestionnaires.

→ Page **d'inscription** pour les nouveaux clients.

La gestion des chambres

A. Backend

→ **API CRUD** pour ajouter, modifier, supprimer et afficher des chambres.

B. Frontend (gestionnaire)

→ Interface de **gestion** des chambres : ajout, modification, suppression.

→ **Vue d'ensemble** des chambres disponibles.

C. Frontend (client)

→ Vue des **chambres disponibles** avec possibilité de réservation.

La gestion des réservations

A. Backend

→ **API** pour créer, consulter, modifier et annuler des réservations.

B. Frontend (gestionnaire)

→ **Vue d'ensemble des réservations**, avec options pour les modifier ou les annuler.

C. Frontend (client)

→ Interface pour **effectuer une réservation**.

→ Vue des **réservations passées et à venir**.

La gestion des clients et des employés

A. Backend

→ API CRUD pour **gérer les informations des clients et des employés.**

B. Frontend (gestionnaire)

→ **Interfaces** pour gérer les informations des clients et des employés.

Système de feedback et d' évaluation

A. Backend

→ API permettant aux clients de laisser des **commentaires** et des **évaluations**.

B. Frontend (client)

→ Interface pour **soumettre des feedbacks**.

C. Frontend (gestionnaire)

→ **Vue des feedbacks** avec option pour y **répondre**.

Système de fidélité

A. Backend

→ **Système de points** pour **récompenser** les clients fidèles.

B. Frontend (client)

→ **Interface** pour visualiser et utiliser les points de fidélité.

Gestion des événements et des installations

A. Backend

→ API pour **gérer des événements spécifiques ou des installations.**

→ Système pour **envoyer des notifications automatisées.**

B. Frontend (gestionnaire)

→ **Interface** pour **gérer et planifier** des événements ou des installations.

→ **Interface** pour gérer les **préférences de notification** et visualiser les notifications reçues .

C. Frontend (client)

→ Vue des **événements à venir** et option pour **réserver** des installations.

Analyses et rapports

A. Backend

→ **Outils d'analyse** pour évaluer la satisfaction client, l'occupation des chambres, les performances financières.

B. Frontend (client et gestionnaire)

→ **Tableau de bord** montrant diverses métriques (**mesures techniques chiffrées, souvent collectées automatiquement**) et KPI (**Indicateurs Clés de Performance**).

mieux comprendre ...

Métriques = Les "jauges" de l'hôtel

Ce sont des **chiffres qui se mettent à jour automatiquement** :

- Nombre de chambres occupées aujourd'hui
- Nombre de réservations cette semaine
- Argent gagné ce mois
- Nombre de clients connectés sur le site
- Nombre de commentaires reçus

KPI = Les indicateurs les plus importants

Ce sont les **chiffres cruciaux** pour savoir si l'hôtel marche bien :

- **Taux d'occupation** : 75% des chambres sont occupées
- **Note moyenne** : 4,2/5 étoiles
- **Chiffre d'affaires mensuel** : 50 000€
- **Clients fidèles** : 30% de clients qui reviennent



Gestion avancée des employés

A. Backend

→ Fonctionnalités pour **gérer les horaires, les jours de congé, les formations.**

B. Frontend (client et gestionnaire)

→ **Interface** dédiée pour la gestion des ressources humaines.

Gestion de Projet et Collaboration d'Équipe

Utilisation de Trello :

Pour organiser et suivre l'avancement des tâches. Les tâches doivent être clairement définies, assignées à des membres spécifiques de l'équipe et suivies régulièrement.

Réunions d'équipe :

Organisez au moins deux points de rencontre par semaine pour discuter de l'avancement du projet, des défis rencontrés et des prochaines étapes. Ces réunions doivent être planifiées à l'avance et tous les membres de l'équipe doivent y participer activement.

Gestion de Projet et Collaboration d'Équipe

Rapport de Projet :

Rédigez un rapport détaillant :

- les différentes phases du développement, les décisions prises,
- les problèmes rencontrés,
- les solutions adoptées,
- ainsi que les tâches assignées et réalisées par chaque membre chaque semaine.

Stockez ce rapport sur le Google Drive du groupe et partagé avec l'accompagnateur.

Gestion de Projet et Collaboration d'Équipe

Conseils pratiques :

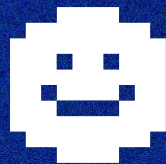
→ **Divisez pour mieux régner** : Abordez chaque étape séquentiellement et assurez-vous de maîtriser chaque fonctionnalité avant de passer à la suivante.

Critères d'évaluation :

→ **Fonctionnalité** : L'application doit être fonctionnelle et refléter toutes les spécifications.

→ **Qualité du code** : Le code doit être propre, bien organisé et suivre les bonnes pratiques de Spring.

→ **Collaboration et gestion de projet** : La manière dont votre équipe collabore et gère le projet sera également évaluée.



 La Plateforme