

# CAHIER DES CHARGES

## Contexte et définition du problème

Pour aider à la prise de décision en matière de vacances, On a pensé à mettre en œuvre une application permettant de location de Bungalow.

## Objectif du projet

Création d'un backend permettant de définir la partie métier d'une application de location de Bungalow.

## Cibles du projet

Cette application s'adresse :

- ✓ Aux clients /internauts consultent les offres.
- ✓ À l'administrateur qui gère les offres.

## Description fonctionnelle des besoins

La future application doit satisfaire la liste des besoins fonctionnels suivants :

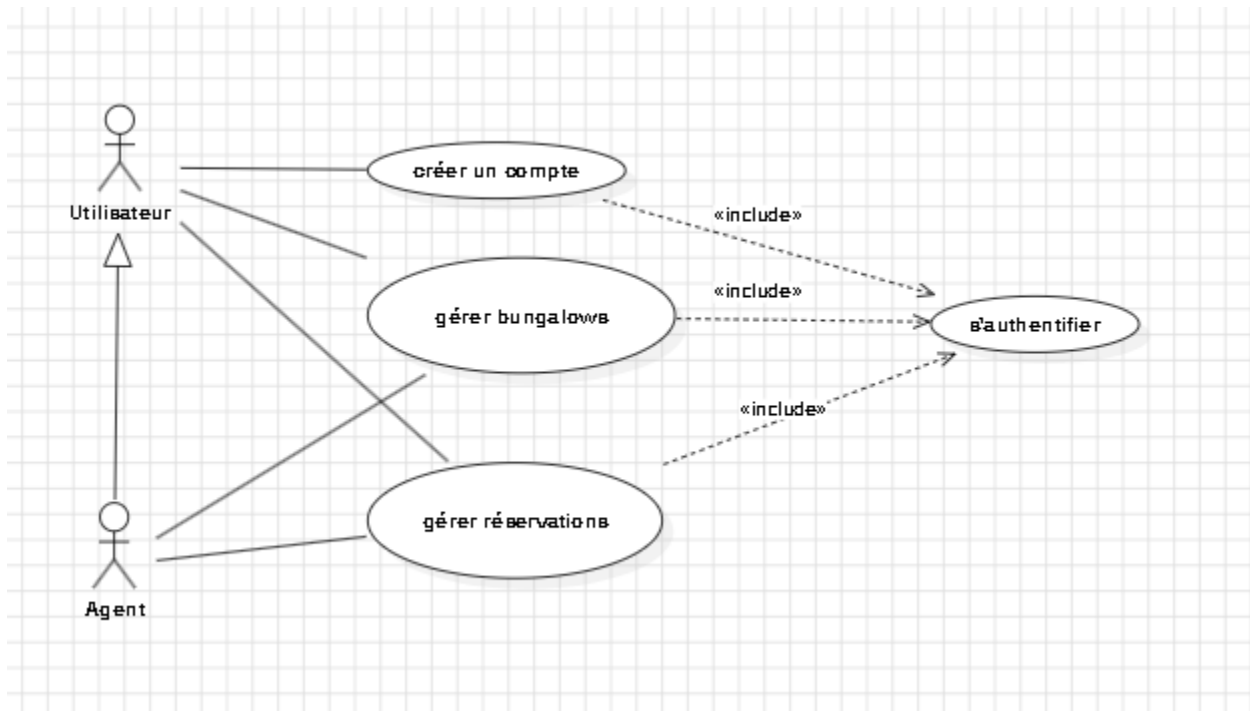
- ✓ Authentification administrative
- ✓ Ajout d'une nouvelle offre
- ✓ Consultation des offres

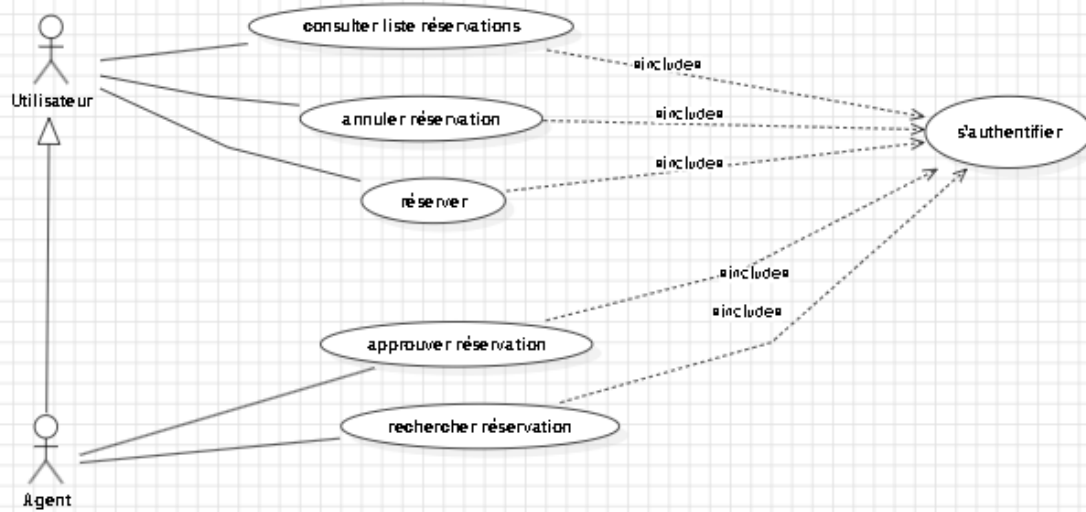
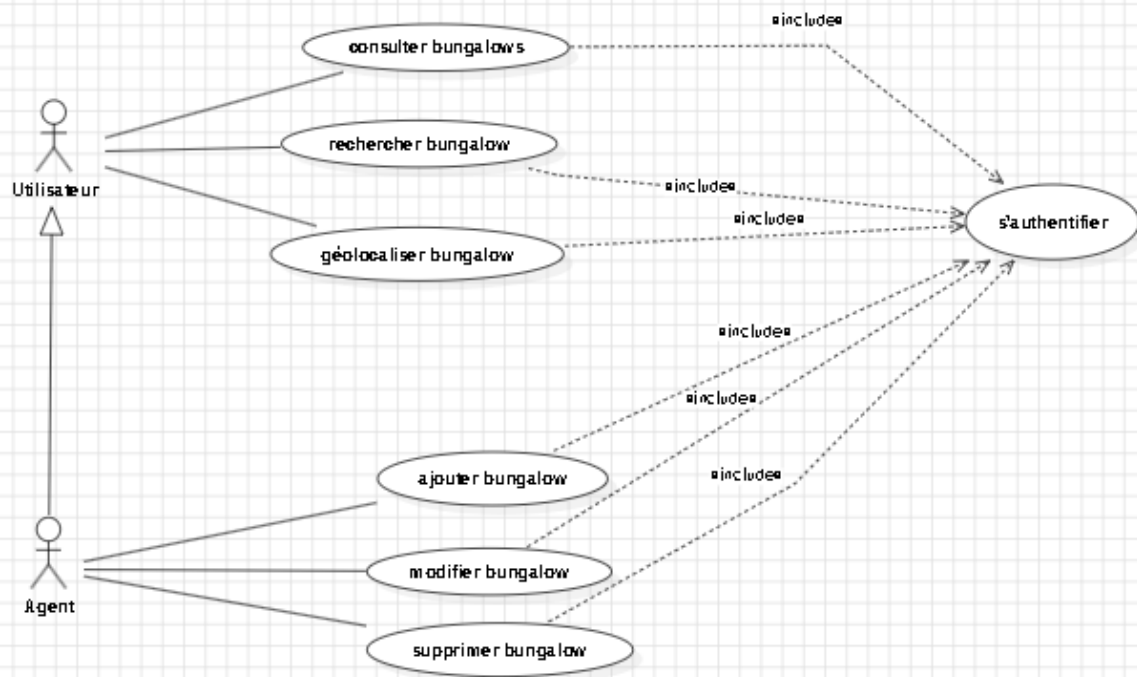
## **Description NON fonctionnelle des besoins**

Les contraintes qui doivent être respectées :

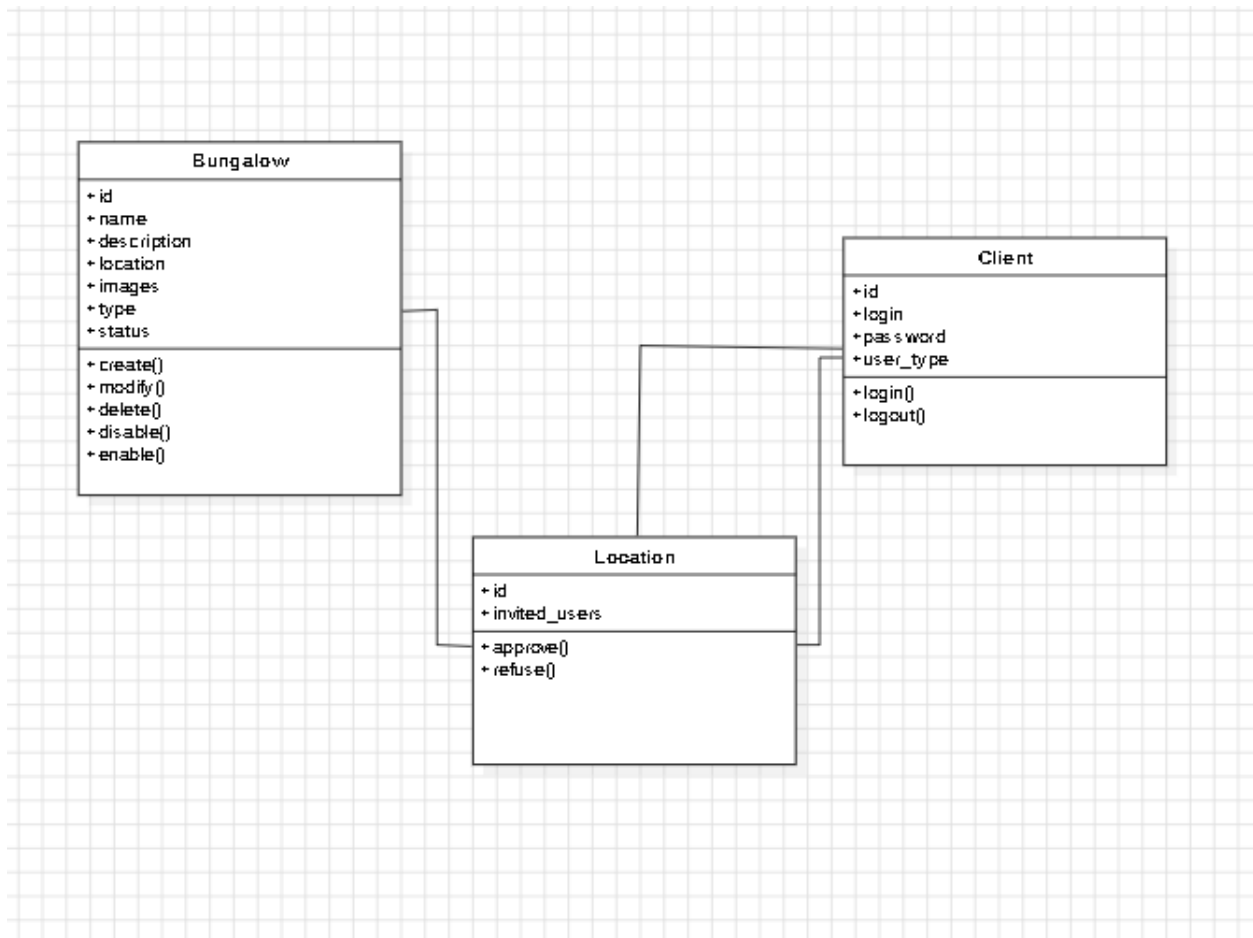
- ✓ L'ergonomie : notre application doit présenter une interface simple pour que l'utilisateur ait une manipulation aisée.
- ✓ La performance : notre application doit être performante c'est-à-dire à travers ses fonctionnalités, répondre à toutes les exigences des usagers d'une manière optimale et dans un temps de réponse raisonnable
- ✓ La convivialité : les interfaces graphiques claires
- ✓ La sécurité

## Diagramme de cas d'utilisation :





## Diagramme de classes :



## Fonctionnalités du Backoffice

1. **Authentification de l'Admin** : Permet à l'admin de se connecter en utilisant son login et mot de passe. Redirige vers le Dashboard après une connexion réussie.
2. **Gestion des Bungalows** :
  - a. **Ajouter un Bungalow** : L'admin peut ajouter de nouvelles Bungalows au système de réservation en spécifiant des détails comme l'adresse, le prix, les images et le statut du Bungalow.
  - b. **Mettre à Jour un Bungalow** : Modifier les informations d'un bungalow qui existe déjà (par exemple : la localisation, le statut).
  - c. **Supprimer un Bungalow** : supprimer un bungalow de la base de données.
  - d. **Voir la Liste des Bungalows** : Afficher une liste de toutes les voitures avec l'option de recherche.
  - e. **Details Bungalow** : Permet au admin de demander les détails d'un Bungalow spécifique.
3. **Gestion des Réservations** :
  - a. **Ajouter une Réservation** : L'admin peut ajouter de nouvelles réservations.
  - b. **Voir les Réservations** : Accéder à une liste de toutes les réservations avec l'option de recherche.
  - c. **Approuver une Réservation** : L'admin peut approuver la demande de réservation d'un client.
  - d. **Modifier une Réservation** : L'admin peut modifier une réservation du bungalow si nécessaire.
  - e. **Supprimer une Réservation** : L'admin peut annuler une réservation du bungalow si nécessaire.
  - f. **Details Réservation** : Permet au admin de demander les détails d'une réservation spécifique.

## API Endpoints

### 1. Authentification Client :

- a. **POST /api/signup** : Permet au client de s'inscrire
- b. **POST /api/login** : Connexion du client en utilisant son login et mot de passe.
- c. **POST /api/logout** : Déconnexion du client.

### 2. Recherche et de Location de Bungalows :

- a. **GET /api/bungalows** : Permet aux clients de consulter la liste des Bungalows.
- b. **GET /api/bungalows/{id}** : Permet aux clients de demander les détails d'un Bungalow spécifique.
- c. **POST /api/bungalows/{id}/geo** : Permet aux clients de demander les coordonnées d'un Bungalow.
- d. **GET /api/reservation/all** : Permet aux clients de consulter toutes leurs réservations.
- e. **POST /api/reservation/reserve/{id}** : Permet aux clients de demander une réservation d'un Bungalow.
- f. **PUT /api/reservation/{id}** : Permet aux clients d'annuler une réservation de location confirmée en spécifiant l'ID de la réservation.