

# Conception d'un site web pour un restaurant

# UML

Projet réalisé dans le cadre d'un ECF  
Par Enzo Capitanio  
Année 2023





# Sommaire

## 1. Présentation

- 1.1 Concept
- 1.2 La demande
- 1.3 Spécifications techniques

## 2. Analyse de l'application

- 2.1 Les besoins utilisateurs
- 2.2 Qui sont les utilisateurs
- 2.3 Structure du projet

## 3. Cas d'utilisation: Diagramme

- 3.1 Décomposition du diagramme
- 3.2 Cas d'utilisation

## 4. Diagramme de classe

## 5. Diagramme de séquence

- 5.1 Connection d'un client
- 5.2 Réservation d'un client



# 1. Présentation

## 1.1 Concept

Dans le cadre de mon ECF, nous imaginons qu'un restaurant gastronomique situé à Chambéry(73) fait sa start up et souhaite posséder un site web pour pouvoir enregistrer leurs clients afin de mieux les connaître. Les clients pourront alors s'identifier et réserver une table depuis le site,

Le Quai Antique a donc besoin de quelqu'un pour mener à bien leurs projet.

## 1.2 La demande

Toujours dans l'imagination, je suis alors nouvellement recruté le gérant me demande réaliser le projet et vous trouverez donc à la suite, les différents diagrammes et modèles qui ont permis de structurer ma réflexion.

## 1.3 Spécifications techniques

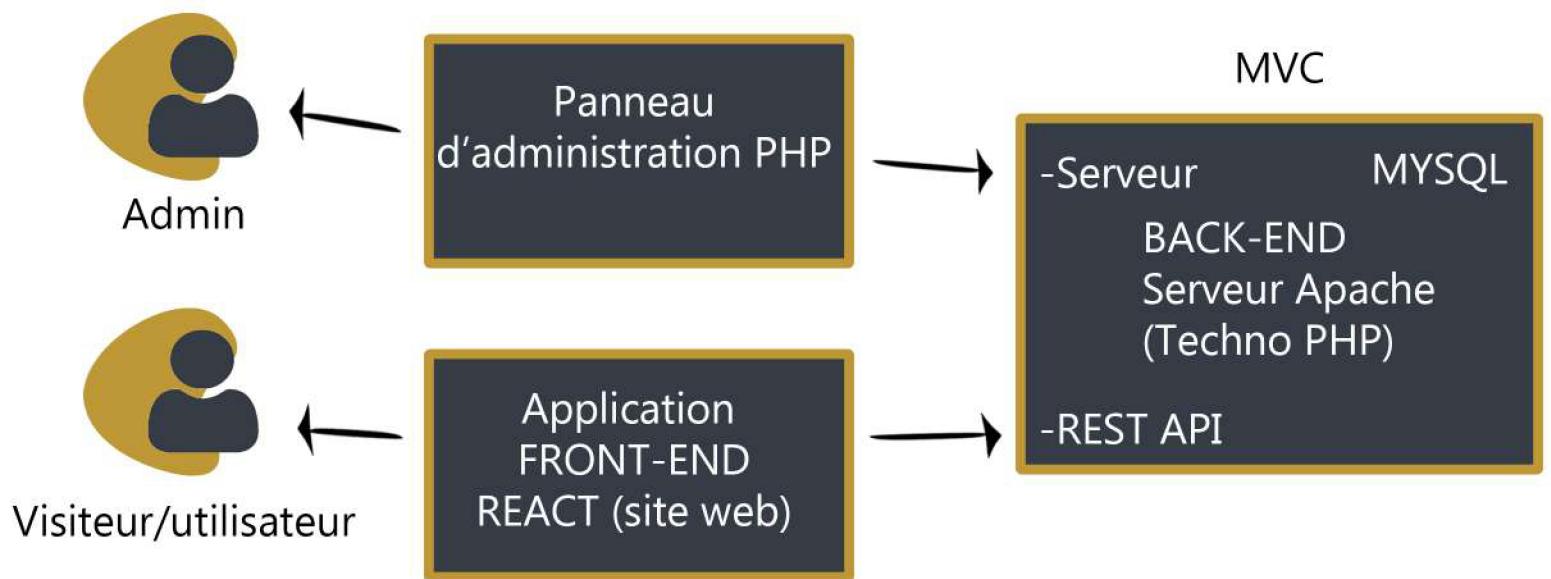
Techno Front-end :  
-REACT JS  
-BOOSTRAP  
-(HTML/CSS)

Techno Back-end :  
-PHP (8.1) / PDO  
-Composer  
-BOOTSTRAP

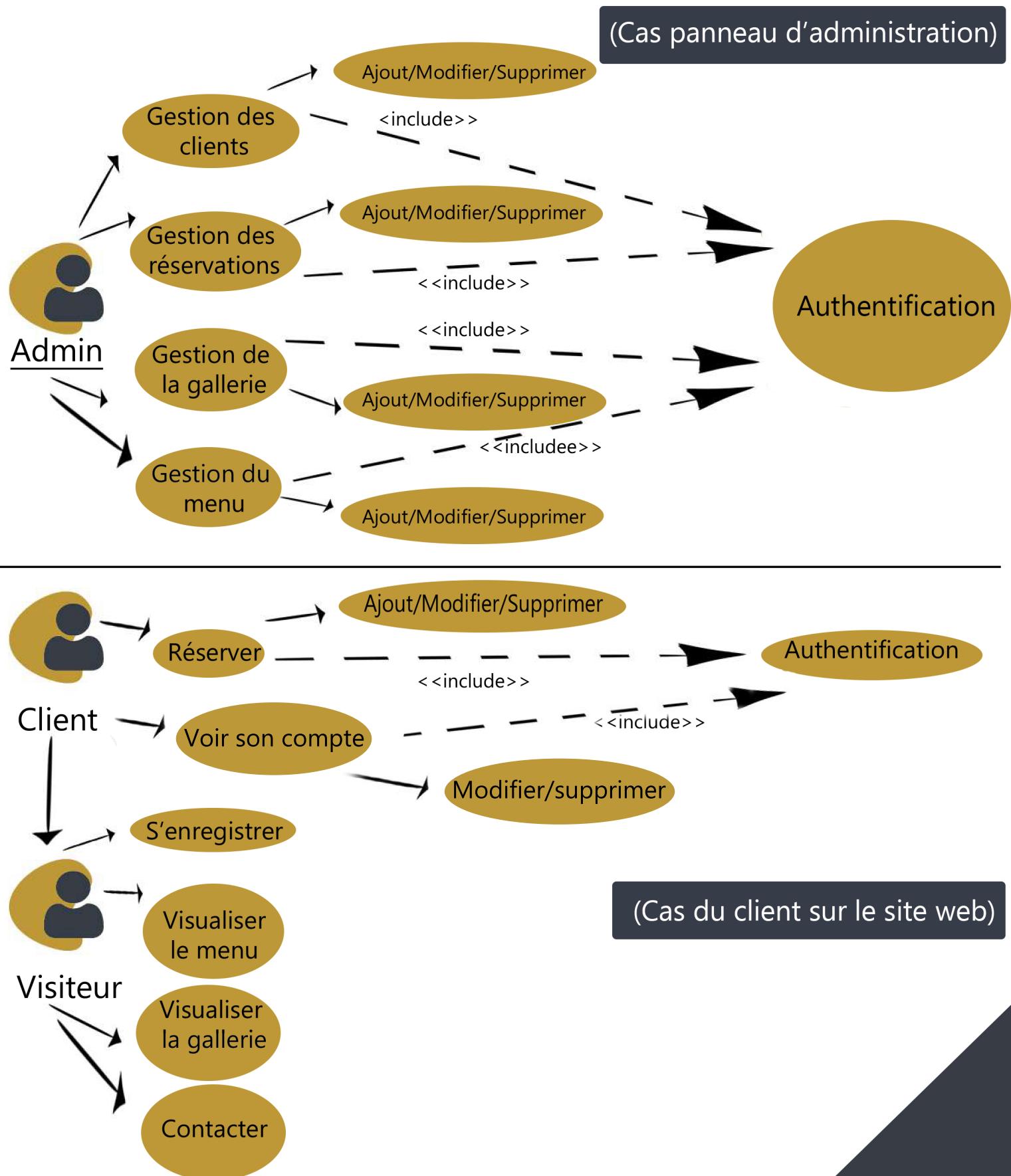
Techno serveur :  
-MYSQL  
-PHP  
-APACHE

## 2.3 Structure du projet

Je vais utiliser un modèle MVC dans le cadre de ce projet et créer un CRUD. Le back end sera composé avec PHP et le front-end sera quant à lui monter avec REACT JS. La partie vue aura en réalité 2 parties, un dédié au panneau d'administration qui servira à gérer les données du site (clients, menu, galerie etc.) et la deuxième qui sera avec REACT pour les clients et visiteurs.



### 3. Cas d'utilisation : Diagramme



## 3.1 Décomposition du diagramme

Le site web comporte différentes parties qui peuvent être analysées séparément.

### **La partie Administration:**

L'administrateur pourra gérer l'application en back office.

Il pourra créer, modifier et supprimer plusieurs éléments tel que les clients, les réservation, la galerie ou encore la carte à condition d'être authentifié.

### **La partie utilisateur (client et visiteur):**

Le visiteur pourra visualiser la carte et la galerie tandis que le client, qui aura donc créer un compte pourra quand à lui s'authentifier pour faire une réservation. Il pourra alors modifier ou supprimer son compte et ou sa réservation.

Client ou visiteur, pourront tout deux contacter l'entreprise grâce à un formulaire de contact.

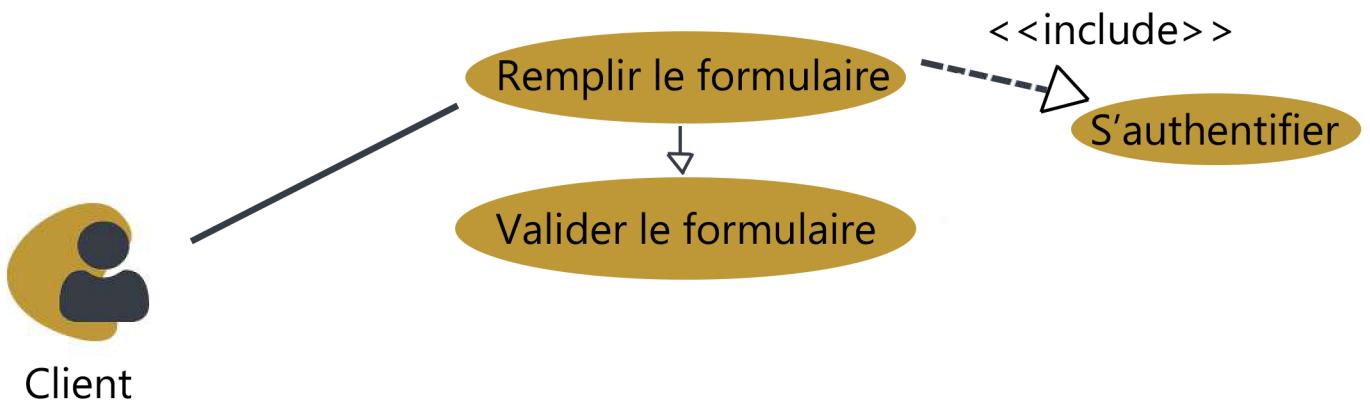
### **Autre:**

Le formulaire de contact enverra un email à l'entreprise avec les informations que l'utilisateur aura rentré.



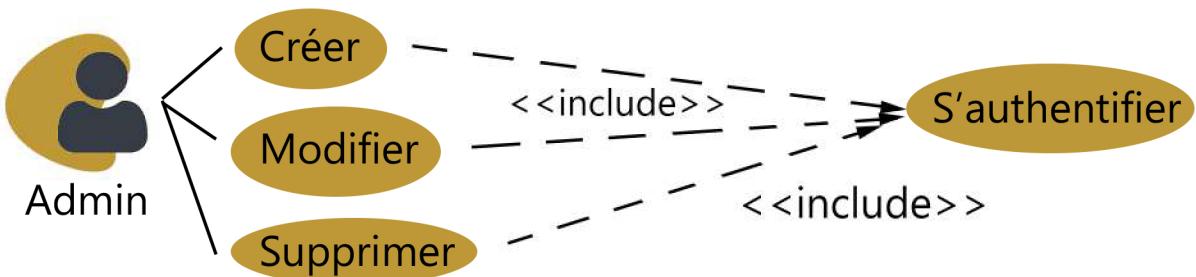
## 3.2 Cas d'utilisation

Le client réserve une table :



1. Le client s'authentifie
2. Il accéde au formulaire de réservation
3. Il remplit les données attendues
4. Il valide le formulaire

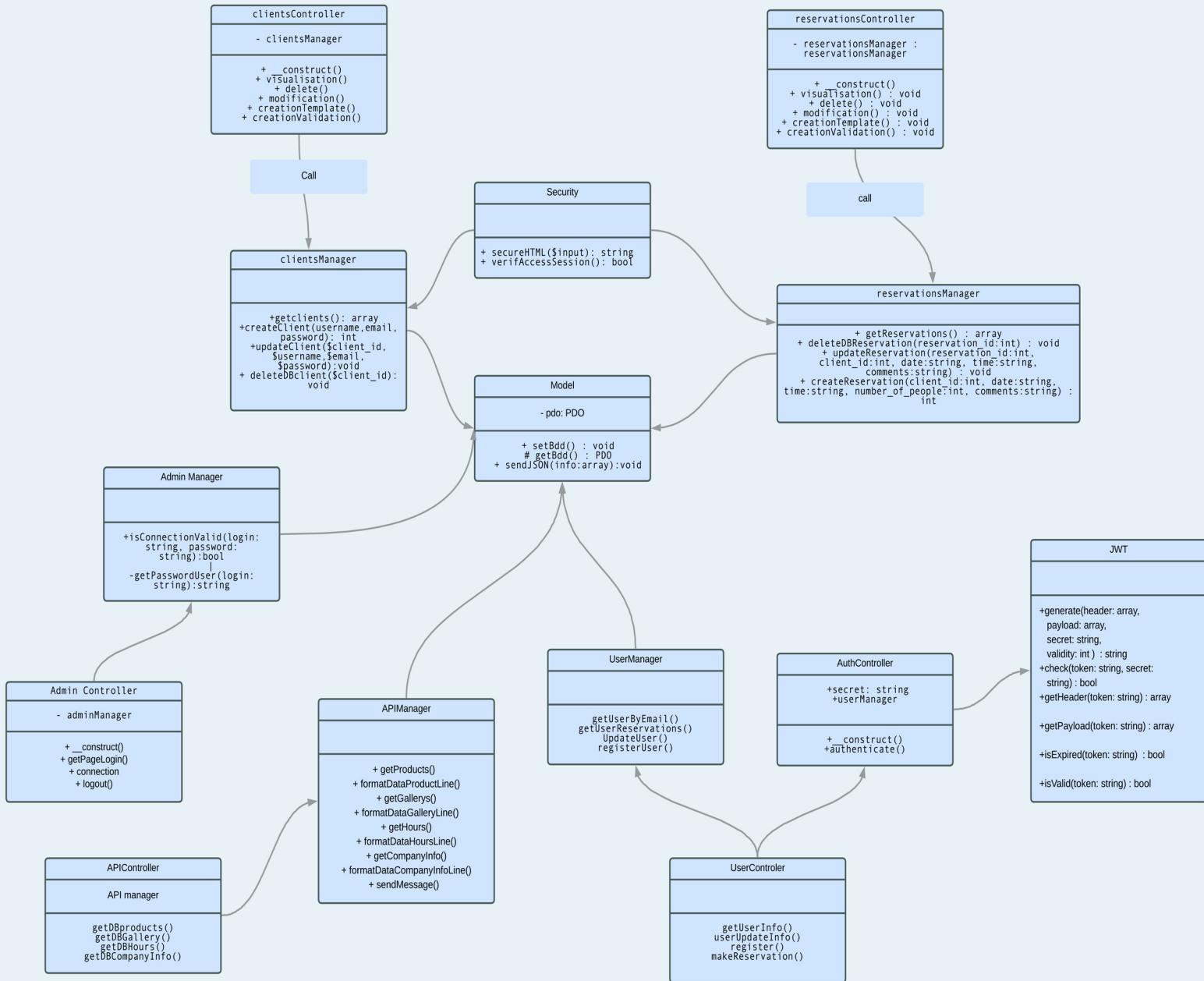
L'administrateur modifie la carte :



1. L'administrateur s'authentifie sur le panneau d'administration
2. Il peut alors aller sur la page listant le menu
3. Il clique sur «modifier»
4. Il change alors les données souhaitées
5. Il valide

Notez que l'opération est la même pour la galerie, les clients, et les réservations.

# 4. Diagramme de classe



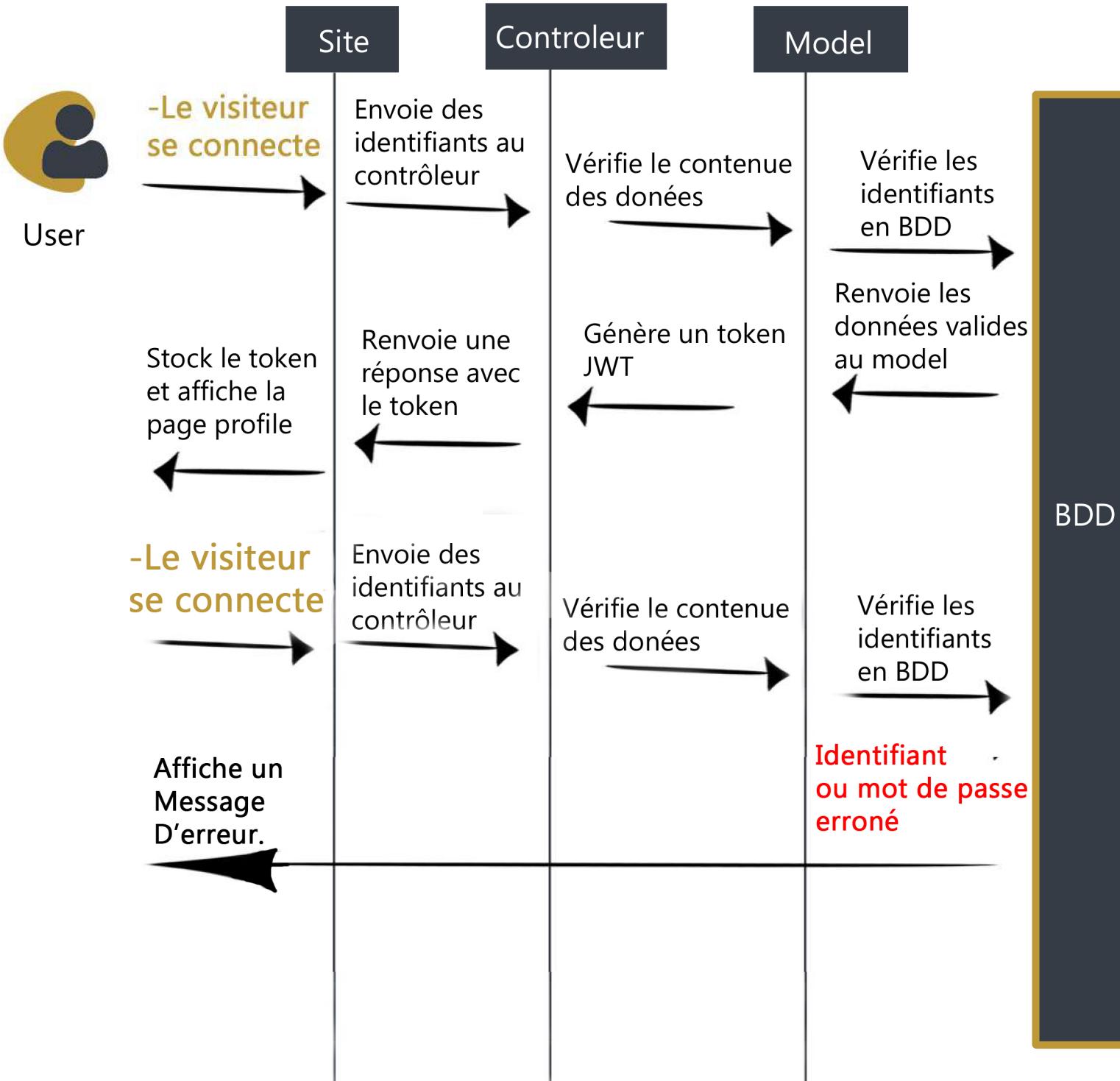
Ici nous pouvons vite nous rendre compte de comment l'application fonctionne, les contrôleurs appellent les «managers» qui sont en l'occurrence les modèles de l'application par fonctionnalité.

Les managers font appels au Model principal de l'application qui aura pour sa part l'unique rôle de faire la connexion à la base de données.

Pour les utilisateurs il y a un système de JWT pour garder une sécurité sur l'authentification des utilisateurs. La classe "security" est utilisé pour vérifier si l'admin a les autorisations pour modifier le contenu du site et donc d'accéder aux pages du panneau d'administration mis en place.

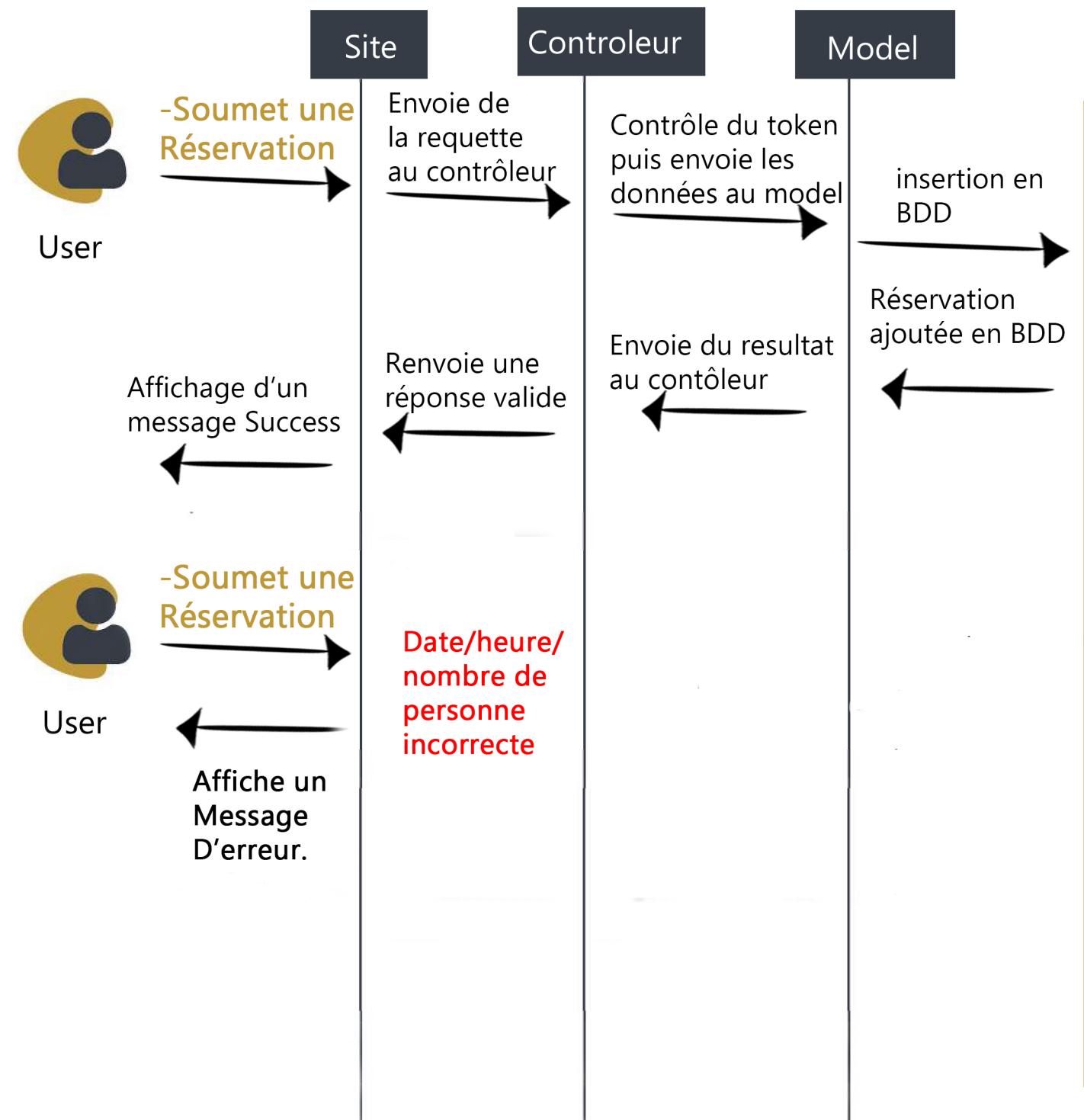
# 5. Diagramme de séquence

## 5.1 Connection d'un client



\*BDD = Base de données

## 5.2 Reservation d'un client



\*BDD = Base de données