

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 et réservation d'un client
- 5.2 Connection et modification d'un produit par l'administrateur



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 reflexion.

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. Analyse du site web

2.1 Les besoins utilisateurs

Le visiteur doit pouvoir:

- visualiser la carte, la galerie d'image et les horaires
- contacter l'entreprise

Le client doit pouvoir:

- Faire ce qu'un visiteur peut faire
- S'inscrire (Nom, email, mot de passe)
- Modifier/supprimer son compte
- S'authentifier
- Faire/modifier/annuler une réservation

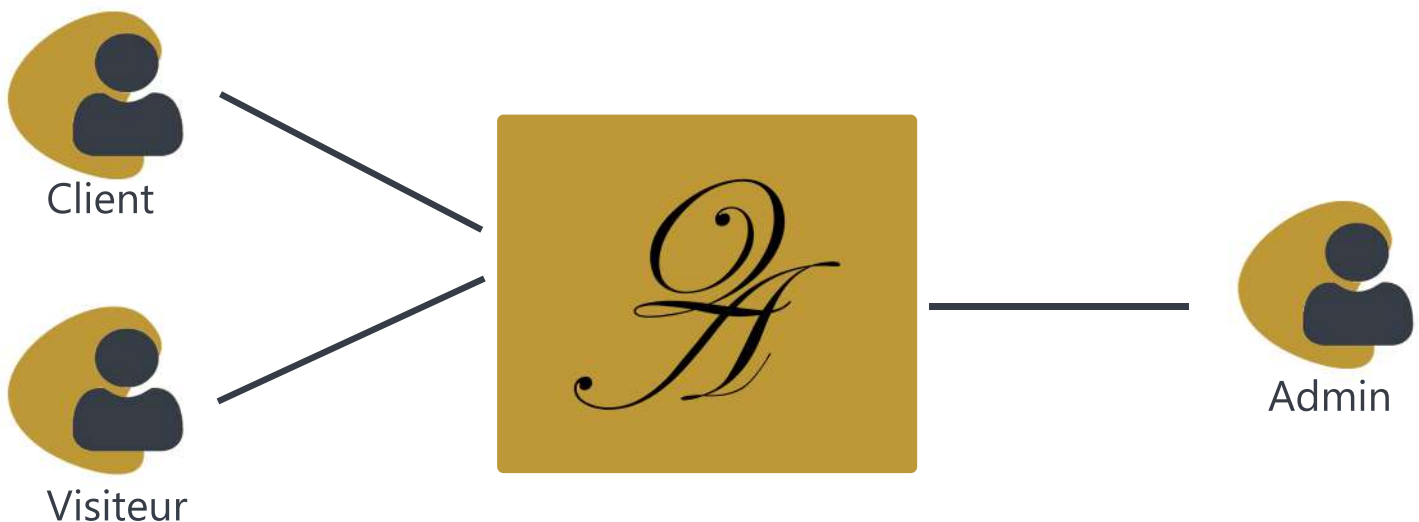
Le responsable et/ou Le chef doit pouvoir:

- Visualiser les clients, les réservations, la galerie, la carte
- Créer/modifier/supprimer les clients, les réservations, la galerie, la carte

2.2 Qui sont les utilisateurs: Diagramme de contexte

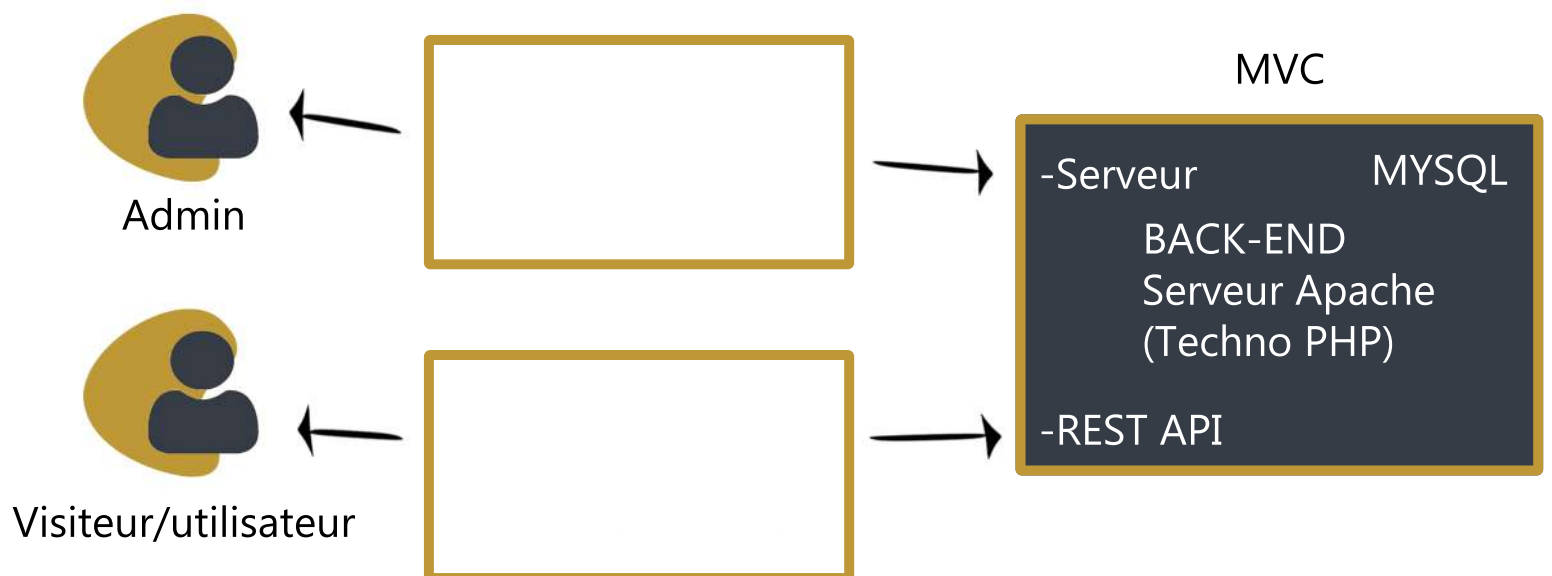
Le container «doré» représente le site web.

- Les acteurs principaux sont : Client et administrateur
- Acteur secondaire; Le visiteur, futur potentiel client

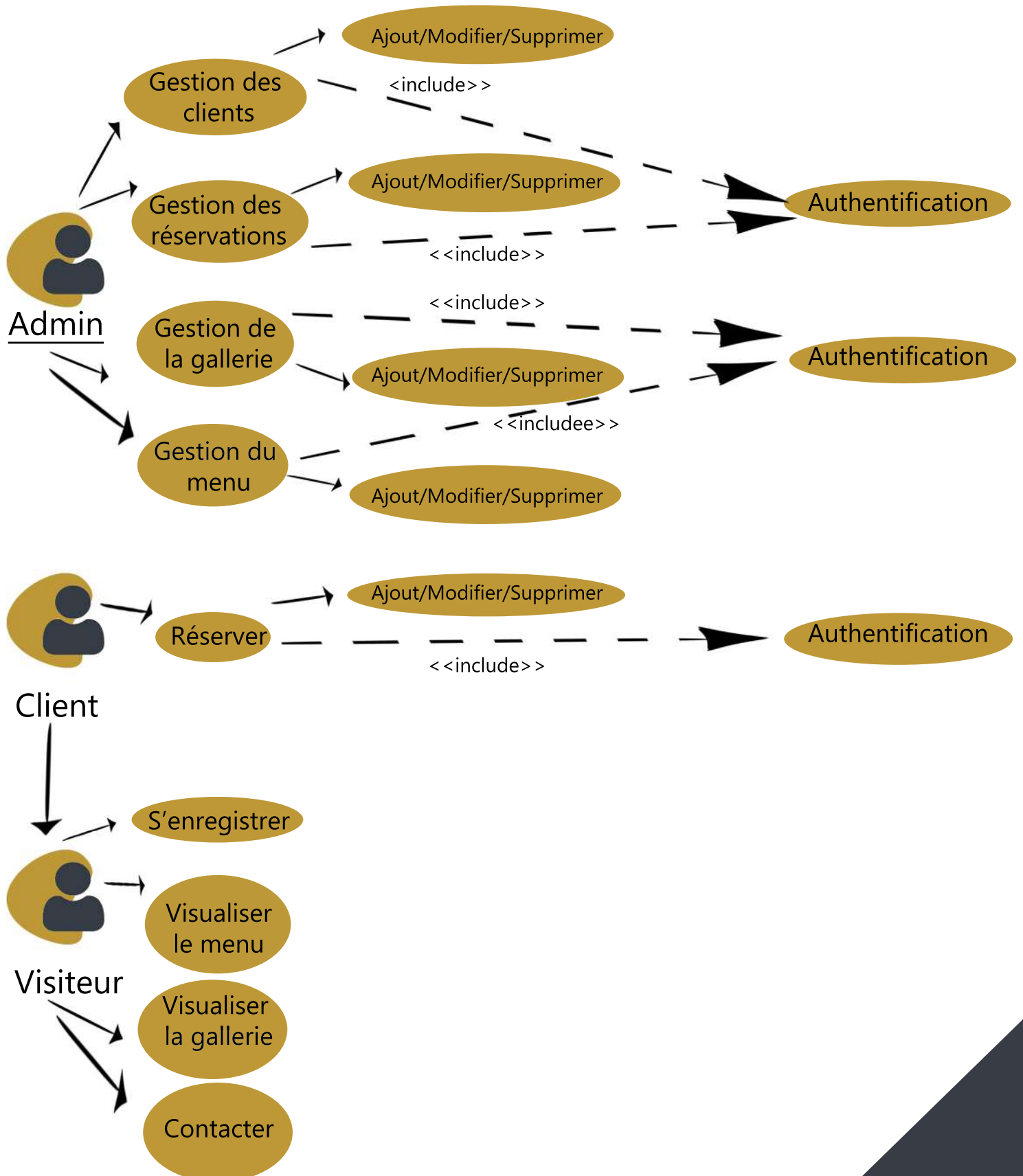


2.3 Structure du projet

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.

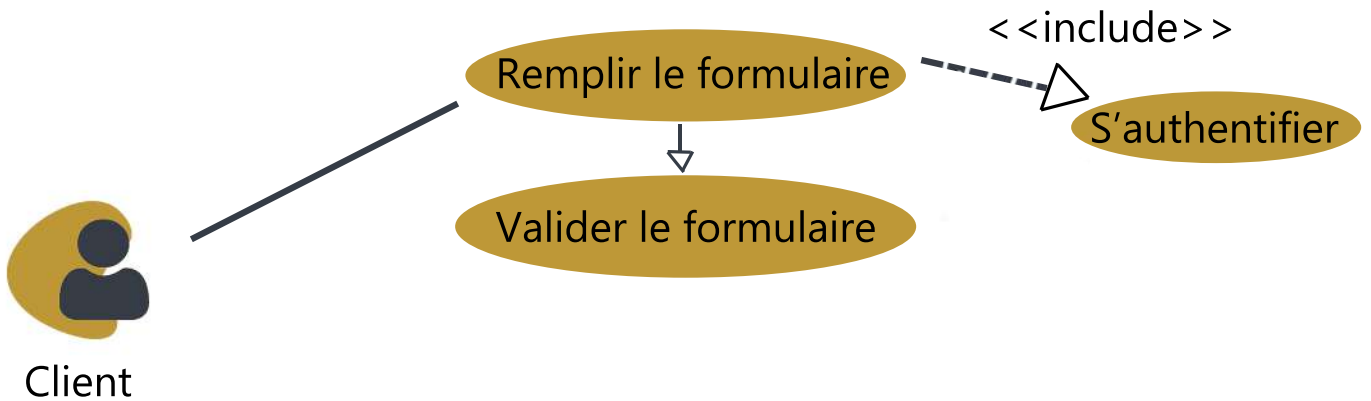
Autre:

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



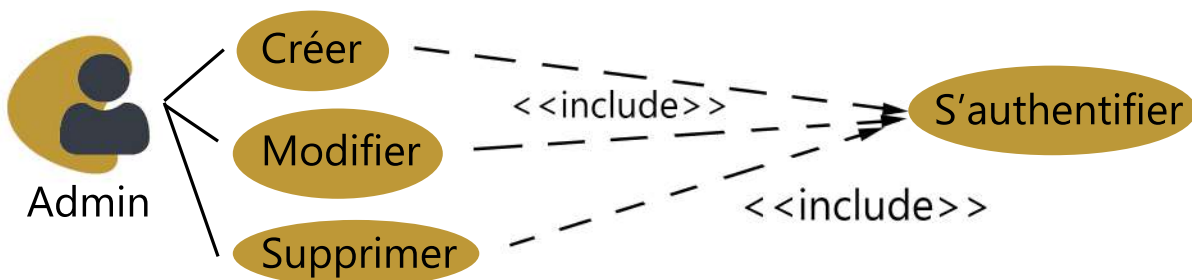
3.2 Cas d'utilisation

Le client réserve une table :



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

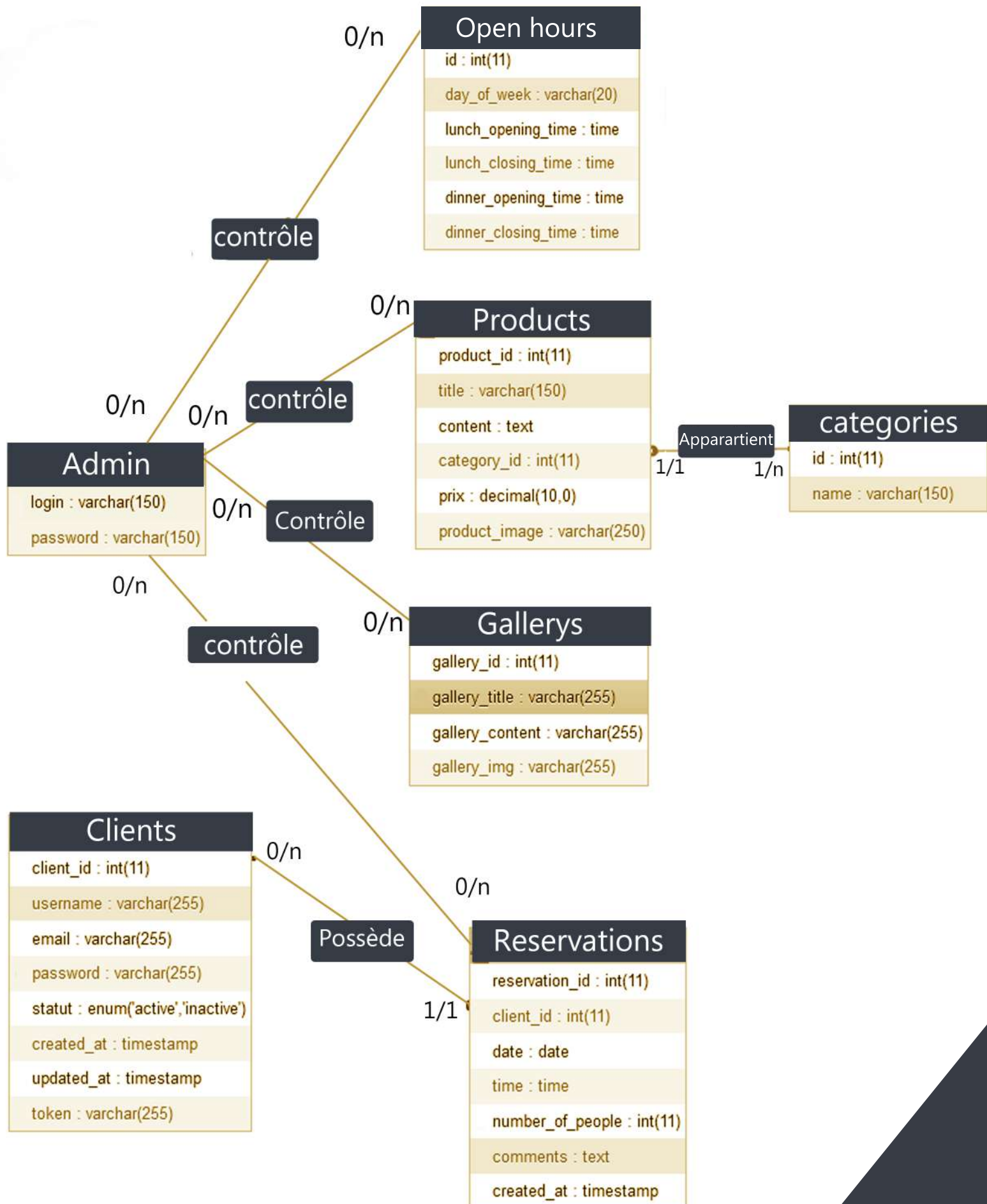
L'administrateur modifie la carte :



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

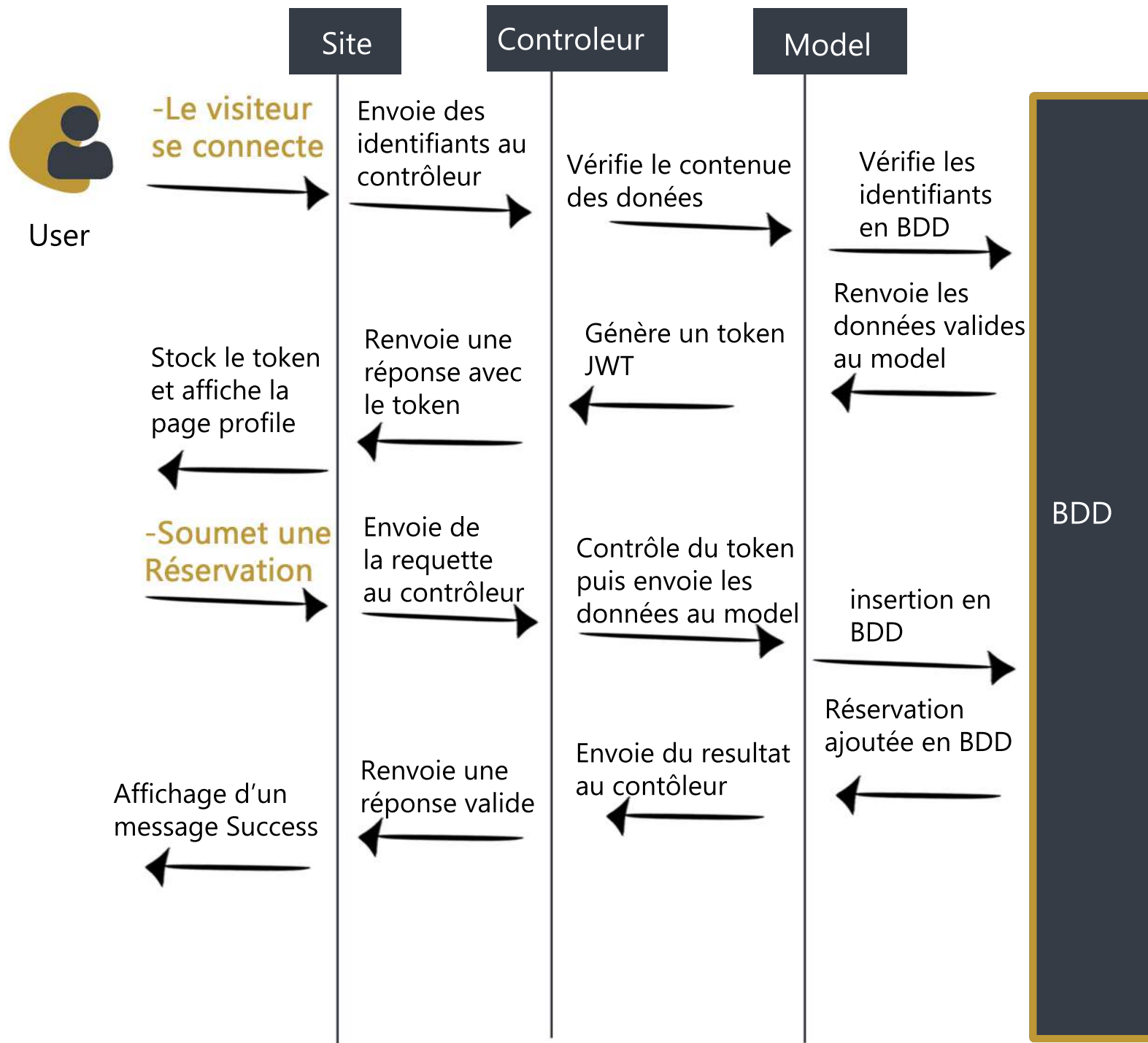
Notez que l'opération est la même pour la galerie, les clients, et les réservations.

4. Diagramme de classe



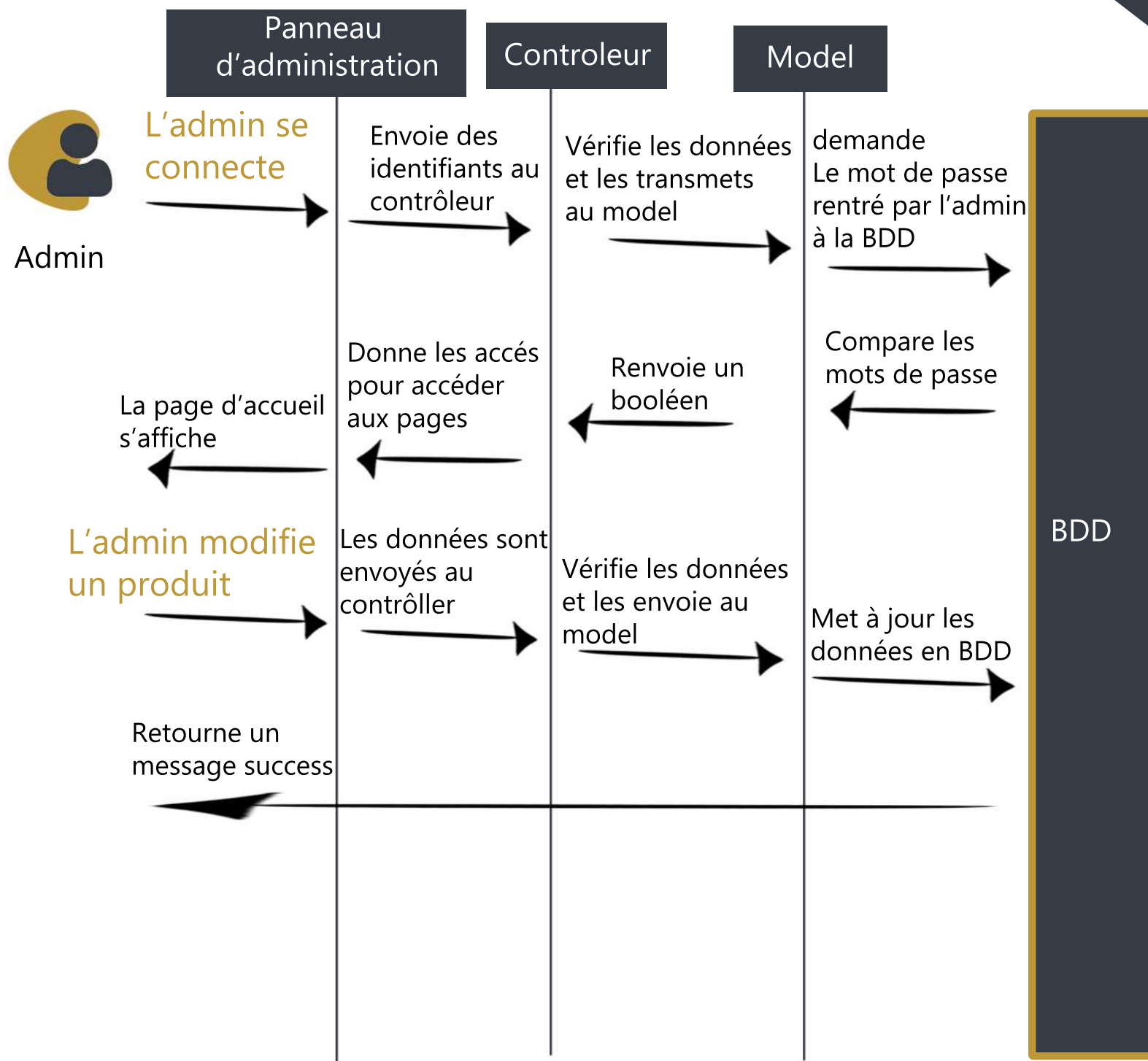
5. Diagramme de séquence

5.1 Connection et réservation d'un client



*BDD = Base de données

5.2 Connection et modification d'un produit par l'administrateur



*BDD = Base de données