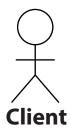


DOCUMENTATION TECHNIQUE

DEVELOPPEUR WEB FULL - STACK

RESTAURANT STUDI

DIAGRAMME DE SÉQUENCE



Application

Database

Champ du formulaire pré-remplis avec les infos existantes

Envoie de la réservation

Affichage du message d'enregistrement effectué

infos demandé en db
selon le client concerné
infos renvoyées de la db

Réservations ranger
dans la table 'réservation'
rendu en status 200

CHOIX DES TECHNOLOGIES EMPLOYÉES

Maquette

- Photoshop
- Illustrator
- Adobe XD

Front-end

- Html5 / Css3 / Javascript
- Framework ReactJs

Back-end

- MySQL
- NodeJs (ExpressJs)

Déploiement

■ lonos avec base de donnée

Gestion / management

- Npm (gestionnaire de package)
- Trello
- GitBash => GitHub

MLD

users id email password convives réservation allergies id num_guest reservation_date allergies status created_at

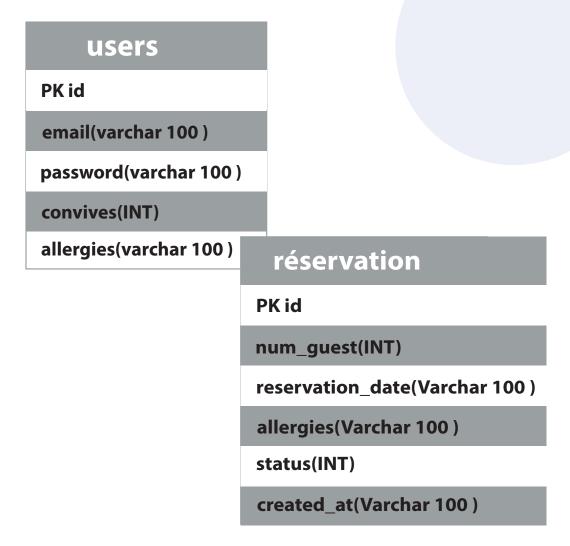
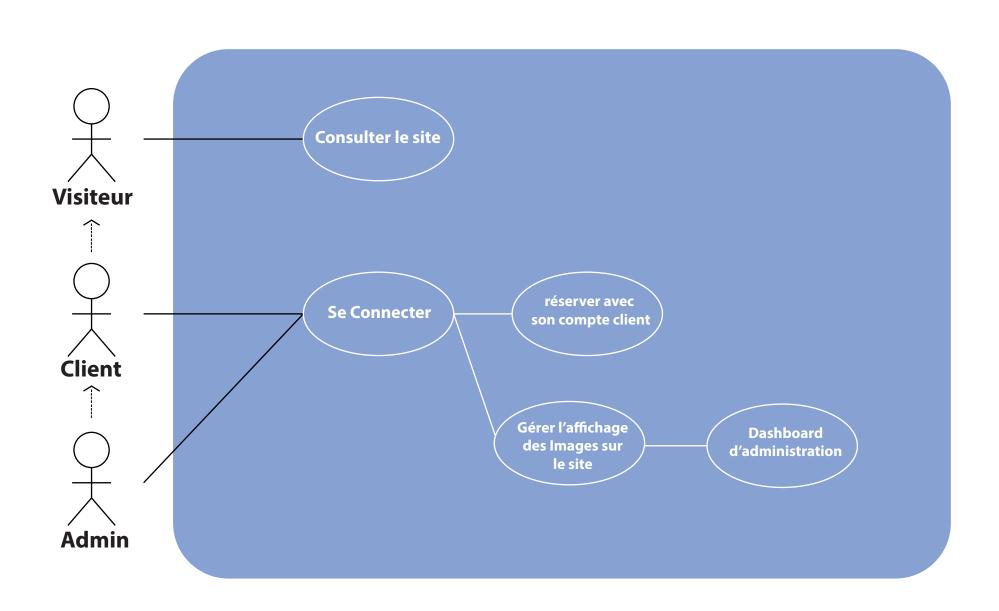


DIAGRAMME DE CAS D'UTILISATION



LES PRATIQUES DE SÉCURITÉ

Afin de sécuriser l'application, j'ai utilisé des contraintes de validation au niveau des formulaire de création et de modification.

Par exemple: avoir un e-mail valide-> vérifie la longueur de la saisie

J'ai mis en place un moyen de crypter les mots de passe des utilisateurs avant qu'ils soient insérés de manière visible dans la base de donnée.

J'ai utilisé la méthode bcrypt comme ceci : bcrypt.hash(mot_de_passe, 10) qui hashera avec un salt factor de 10 le mot de passe, la valeur 10 peux etre changé