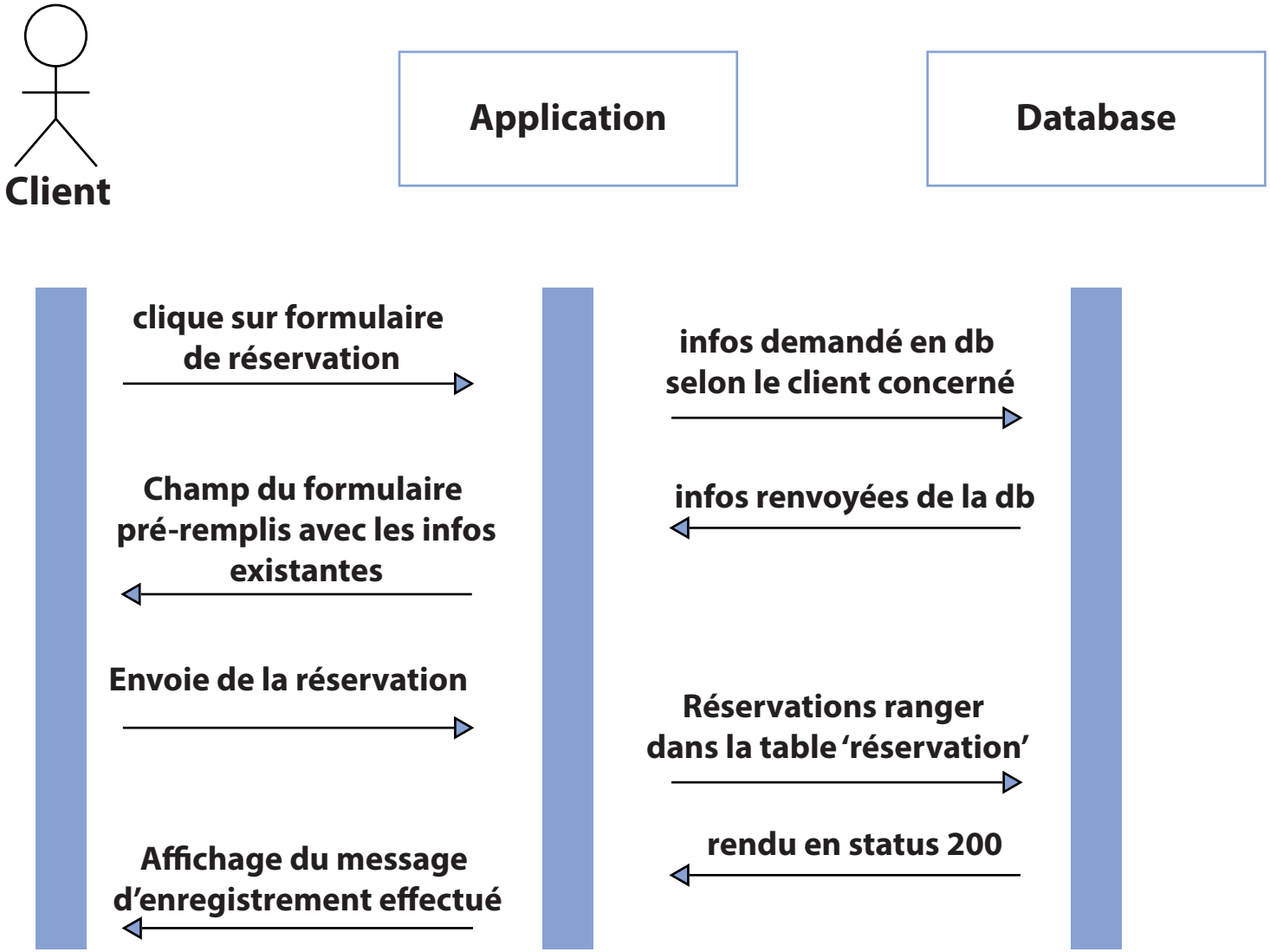
DOCUMENTATION TECHNIQUE

DEVELOPPEUR WEB FULL - STACK

RESTAURANT STUDI

DIAGRAMME DE SÉQUENCE



CHOIX DES TECHNOLOGIES EMPLOYÉES

Maquette

- Photoshop
- Illustrator
- Adobe XD

Front-end

- Html5 / Css3 / Javascript
- Framework ReactJs

Back-end

- MySQL
- NodeJs (ExpressJs)

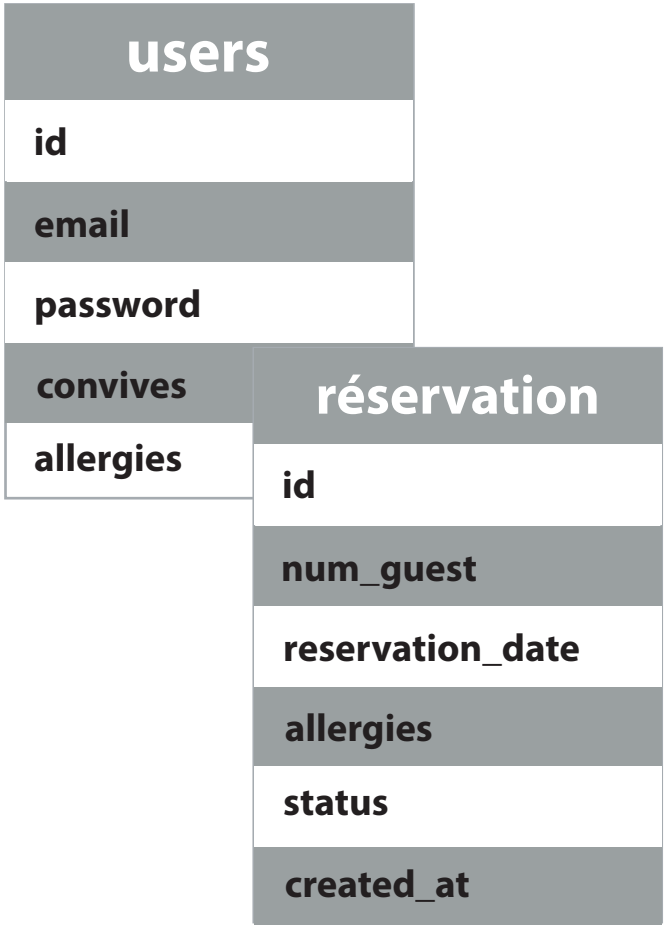
Déploiement

- Ionos avec base de donnée

Gestion / management

- Npm (gestionnaire de package)
- Trello

MLD


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

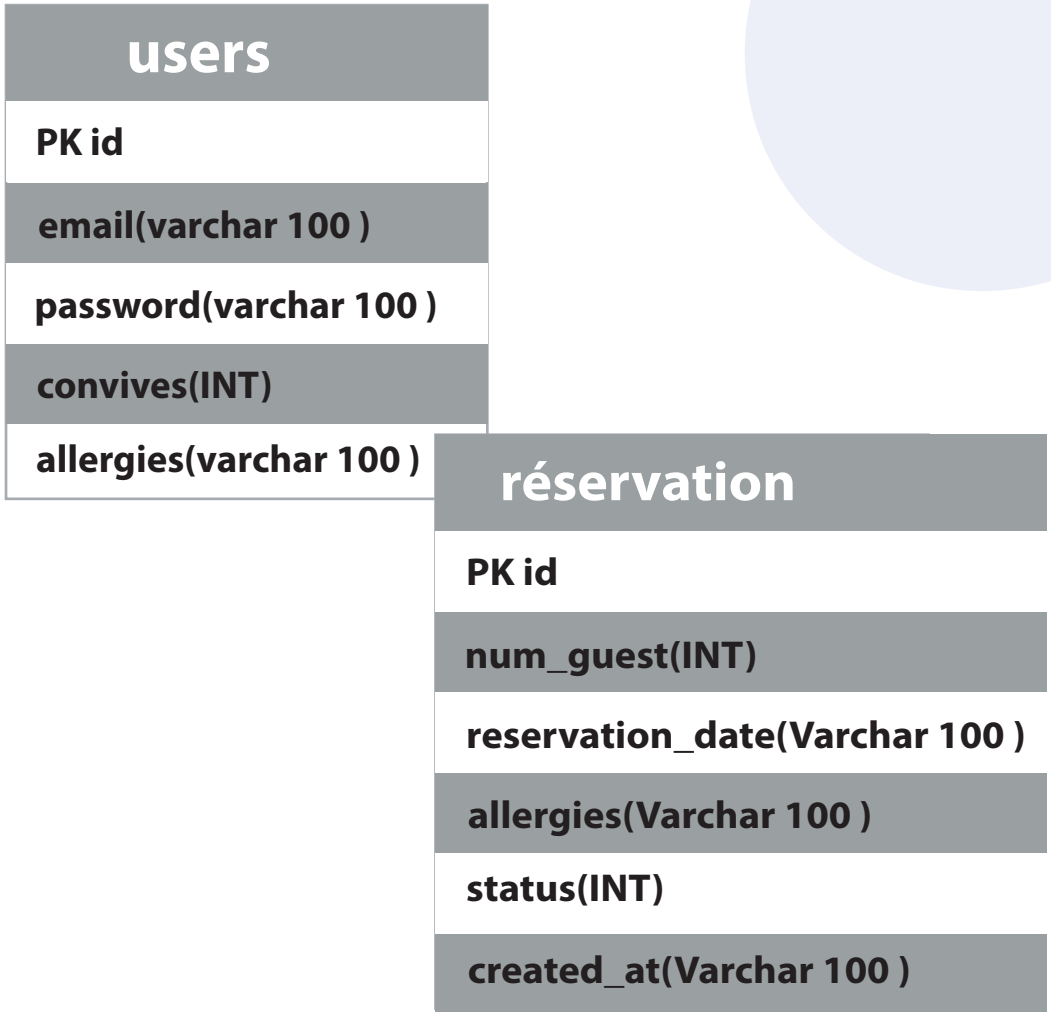
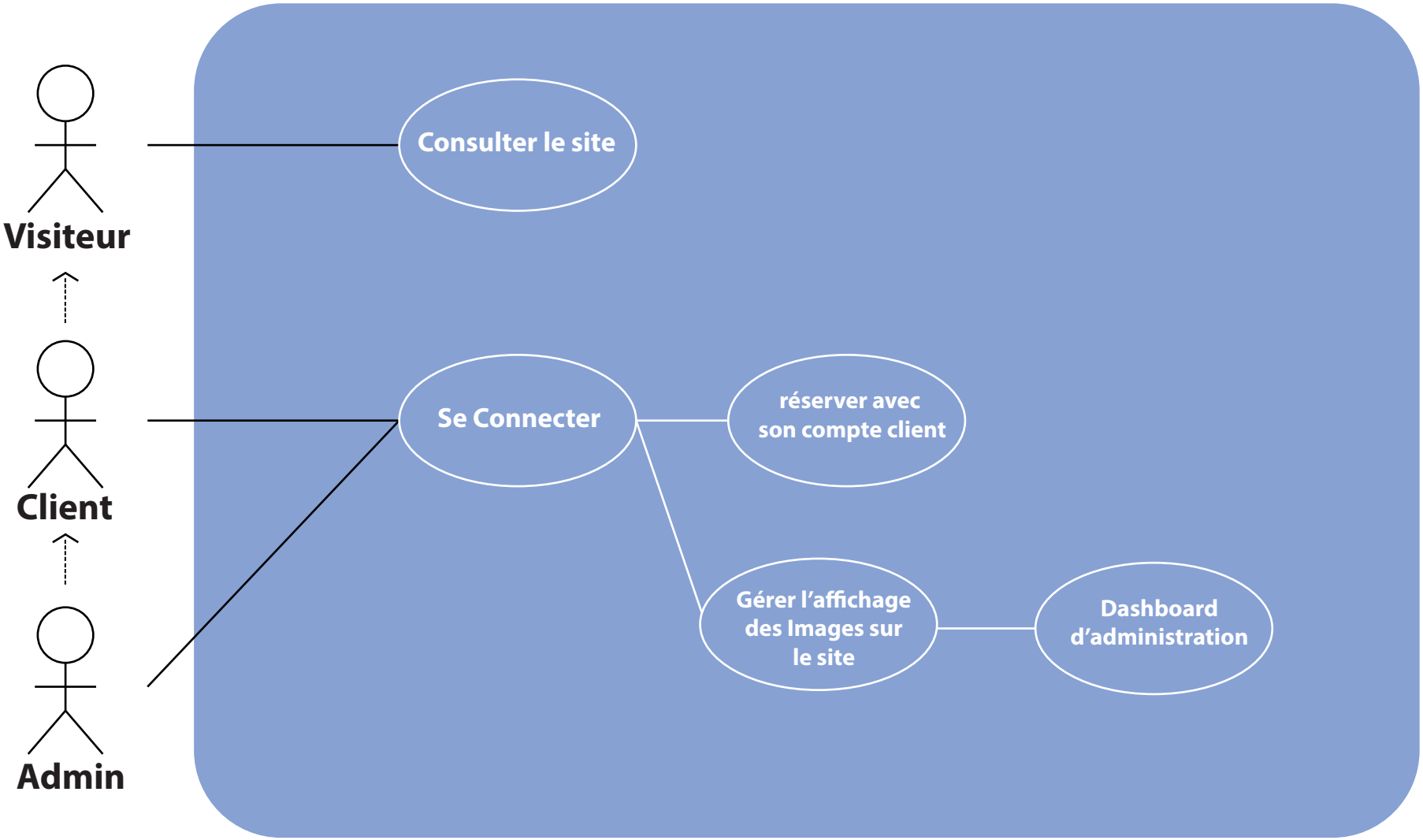

users
PK id
email(varchar 100)
password(varchar 100)
convives(INT)
allergies(varchar 100)
réservation
PK id
num_guest(INT)
reservation_date(Varchar 100)
allergies(Varchar 100)
status(INT)
created_at(Varchar 100)

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