



# ANALYSE DU PROJET

---

Université de yaoundé I

Département: **Informatique**

Unité d'enseignement: **INF 4198 - Projet**

Année académique: **2023 - 2024**

Enseignant: **Dr. Azanzi Jiomekong** - [fidel.jiomekong@facsciences-uy1.cm](mailto:fidel.jiomekong@facsciences-uy1.cm)

Superviseur: **Dr. Thomas Messi Nguete** - [thomas.messi@facsciences-uy1.cm](mailto:thomas.messi@facsciences-uy1.cm)

# PLAN

## **I. INGÉNIERIE DES BESOIN**

- A. Le problème à résoudre
- B. Les solutions existantes
  - 1. Présentation
  - 2. Limites
- C. Notre solution
- D. Spécifications

## **II. CONCEPTION DE LA SOLUTION**

- A. Application mobile de réservation en ligne
- B. Application mobile de vérification de réservation
- C. Application web de gestion des trajet
- D. Dashboard web d'administration

## **III. PRODUCT BACKLOG**

- A. Application mobile de réservation en ligne
- B. Application mobile de vérification de réservation
- C. Application web de gestion des trajets
- D. Dashboard web d'administration

## **IV. DÉFINITION DES SPRINTS**

## INDEX

- **Voyage:** Trajet planifié
- **Agenda:** Programme d'un planning

# I. INGÉNIERIE DES BESOIN

## A. Le problème à résoudre

Au Cameroun, le processus de réservation de billets est souvent difficile pour les usagers tels que les agences de voyage et les voyageurs. Ils doivent faire face à de longues files d'attente, ce qui peut entraîner des temps d'attente considérables.

## B. Les solutions existantes

### 1. Présentation

Des solutions telles que **Flixbus**, **Blablacar** ont respectivement été développées par les entreprises **Flix GmbH** et **Comuto** qui sont des plateformes communautaires de covoiturage permettant la gestion des transports d'un lieu de départ vers une destination à travers des réservations sur application mobile ou sur site web. **Flixbus** étant implanté dans plusieurs pays européens, au Maroc et aux Etats-Unis, **BlablaCar** quant-à lui est le leader mondial du covoiturage avec 100 millions d'utilisateurs en Octobre 2021.

### 2. Limites

Le problème majeur rencontré avec les solutions existantes est qu'elles ne sont pas disponibles au Cameroun. De plus, celles-ci ne s'adaptent pas avec les agences de transport déjà présentes car ne fonctionnent qu'avec leur propre agence (agence de l'entreprise).

## C. Notre solution

Nous avons donc décidé de mettre en place une gamme de produit locaux qui non seulement permettra de faire des réservations de billet de voyages en ligne (pour minimiser la taille des files d'attente rencontrées dans les agences de voyages et réduire le temps d'attente des passagers) mais aussi s'adapter aux agences de voyage déjà existantes au cameroun.

## D. Spécifications

### 1. Application mobile de réservation en ligne

#### Spécifications fonctionnelles

- Consulter la liste des agences de voyage et leurs sites.
- Rechercher une agence de voyage
- Rechercher un site
- Rechercher un trajet
- Rechercher un voyage (trajet planifié)
- Consulter les détails d'un trajet
- Consulter les détails d'un voyage (trajet planifié)
- Réserver (payer) un billet de voyage
- Consulter les détails d'une réservation
- Annuler une réservation
- Consulter l'historique de réservation
- Définir un agenda de voyage

#### Spécifications non fonctionnelles

- **Responsive:** Le produit doit pouvoir s'adapter à n'importe quelle taille de smartphones.
- **La confidentialité des données:** Étant un produit à grand public, les données des utilisateurs doivent être confidentielles.
- **L'extensibilité:** Le produit doit pouvoir évoluer en fonction des besoins de la société.
- **La rapidité:** Les requêtes doivent être optimisées.
- **La convivialité:** Les interfaces doivent être intuitives.

### 2. Application mobile de vérification de réservation

#### Spécifications fonctionnelles

- Vérifier une réservation
- Configurer le voyage
- Authentification

#### Spécifications non fonctionnelles

- **Simplicité:** l'interface utilisateur doit être simple et intuitive

### 3. Application web de gestion des trajets

#### Spécifications fonctionnelles

- Lister le personnel
- Ajouter, modifier, supprimer un membre du personnel
- Consulter les détails d'un membre du personnel
- Ajouter, modifier, supprimer et lister les profils
- Assigner des droits à chaque profils
- Ajouter, modifier, supprimer et lister les itinéraires
- Planifier un voyage
- Générer les rapports d'analyse
- Consulter les statistiques
- Créer, modifier, supprimer et lister les sites de l'agence
- Consulter le planning de voyage.
- Enregistrer une réservation

#### Spécifications non fonctionnelles

- **La sécurité:** Cette application va manipuler les données de plusieurs utilisateurs
- **Robustesse:** Étant un centre de plusieurs calculs en temps réel, ce produit doit être bien conçu.
- **Convivialité:** L'interface doit être intuitive; attrayante et facile à prendre en main

### 4. Dashboard web d'administration

#### Spécifications fonctionnelles

- Ajouter, modifier, supprimer et lister les agences de voyage
- Consulter les détails d'une agence de voyage
- Afficher les statistiques sur les agences de voyages

#### Spécifications non fonctionnelles

- **La confidentialité:** Dans le sens où ce produit sera accessible uniquement par les administrateurs des partie prenantes.

## II. CONCEPTION DE LA SOLUTION

### A. Application mobile de réservation en ligne

#### 1. Acteurs et cas d'utilisations

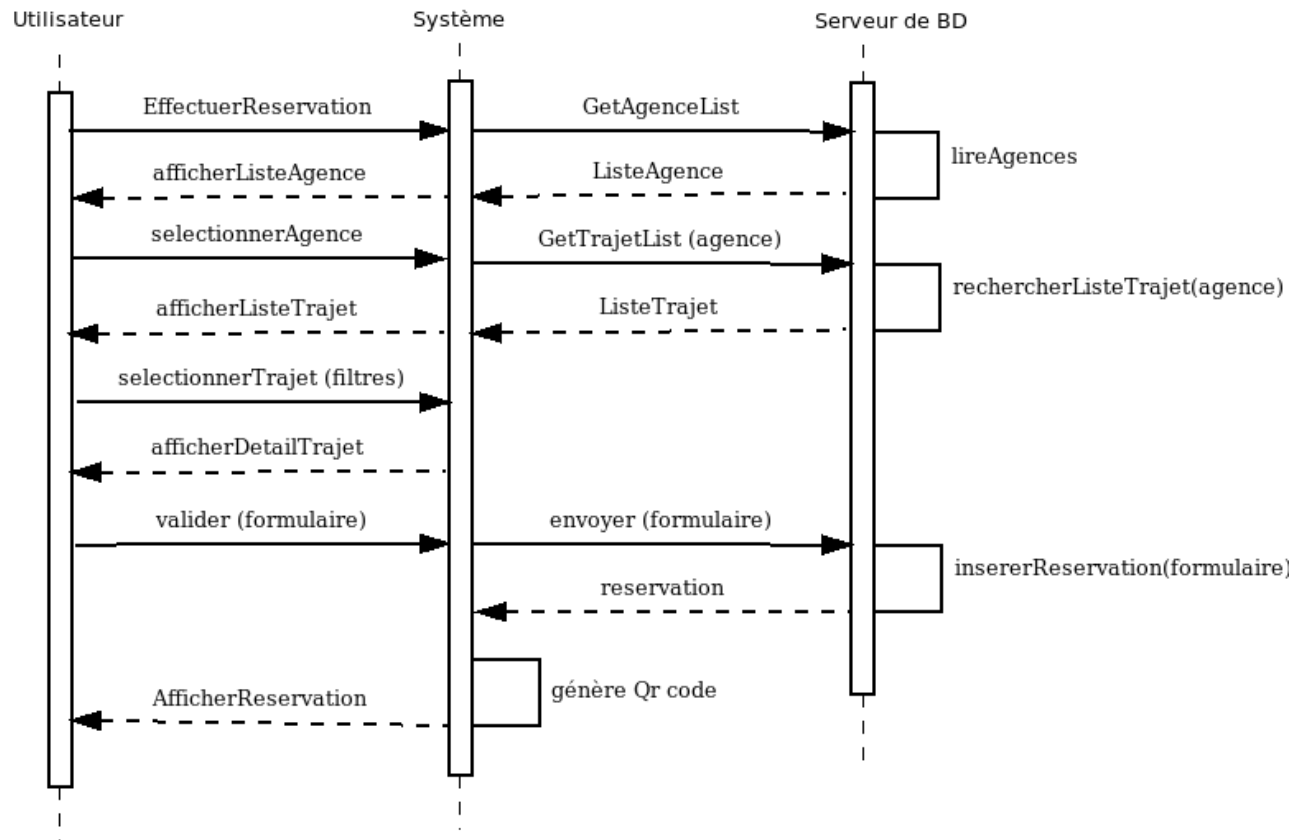
Acteurs	Description	Cas d'utilisations
Client/passager	Toute personnes voulant réserver un billet de voyage en ligne	<ul style="list-style-type: none"> <li>- Consulter la liste des agences de voyage et leurs sites.</li> <li>- Rechercher une agence de voyage</li> <li>- Rechercher un site</li> <li>- Rechercher un trajet</li> <li>- Rechercher un voyage (trajet planifié)</li> <li>- Consulter les détails d'un trajet</li> <li>- Consulter les détails d'un voyage (trajet planifié)</li> <li>- Réserver (payer) un billet de voyage</li> <li>- Consulter le détails d'une réservation</li> <li>- Annuler un réservation</li> <li>- Consulter l'historique de réservation</li> <li>- Définir un agenda de voyage</li> </ul>
Serveur de base de données	Outil permettant d'accéder à la base de données	/

#### 2. Description des principaux cas d'utilisations (textuelle et diagramme de séquence)

- Nom du cas d'utilisation: Réserver (payer) un billet de voyage
- But: Effectuer une réservation pour un trajet planifié dans une agence de voyage
- Acteur: Client/passager, Serveur de base de données
- Pré-condition: Le trajet doit exister.
- Déclencheur: Un bouton "Effectuer une réservation"
- Post-condition: Un E-billet est généré.
- Scénario:
  - L'utilisateur clique sur le bouton "Effectuer une réservation"
  - Le système envoie la requête au serveur de base de données
  - Le serveur de BD retourne la liste des agences
  - Le système affiche la liste des agences
  - L'utilisateur sélectionne l'agence
  - Le système récupère la liste des trajets du serveur de BD et les affiche.
  - L'utilisateur recherche le trajet en fonction des filtres et sélectionne
  - Le système affiche les détails du trajet
  - L'utilisateur saisit ses informations personnelles, de paiement et valide
  - Le système envoie la requête au serveur de BD
  - Le serveur de BD traite la requête et renvoie la réponse
  - Le système notifie le client

Diagramme de séquence correspondant:

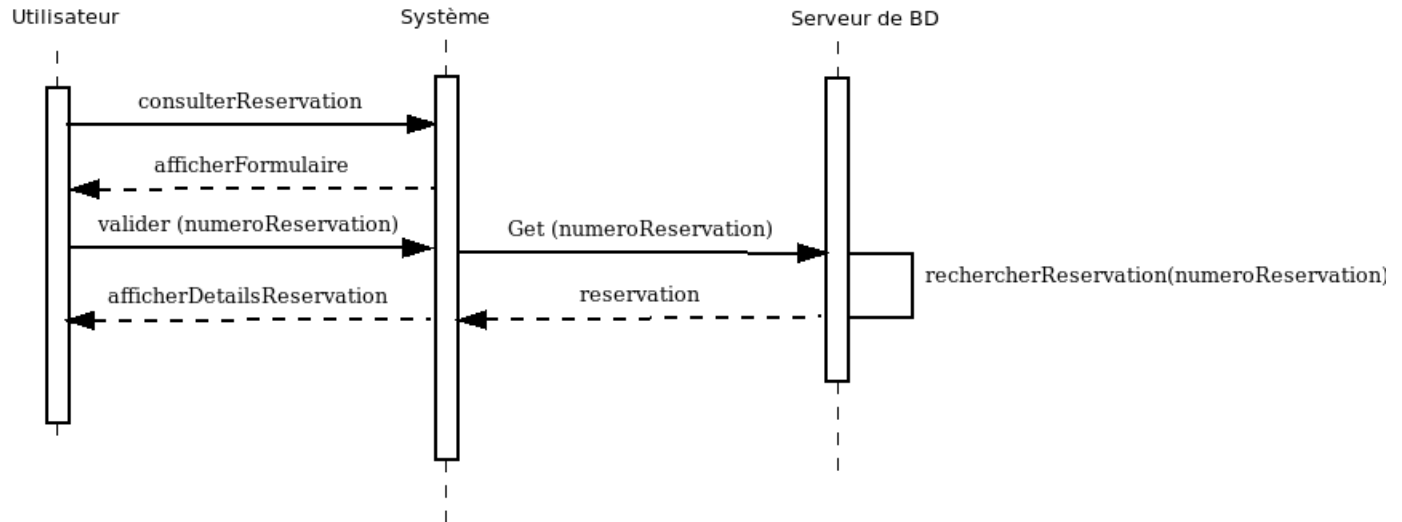




- Nom du cas d'utilisation: Consulter une réservation
- But: Connaître toutes les informations associées à une réservation.
- Acteur: Client/passager, Serveur de base de données
- Pré-condition: La réservation doit exister
- Déclencheur: Un bouton "Consulter une réservation"
- Post-condition: Les informations sont affichées à l'écran
- Scénario:
  - L'utilisateur clique sur le bouton "Consulter une réservation"
  - le système affiche une boîte de dialogue avec un champs de saisie
  - L'utilisateur renseigne le numéro de réservation et clique sur le bouton "valider"
  - Le système envoie la requête au serveur de BD
  - Le serveur de BD traite la requête et retourne le résultat

- Le système affiche une nouvelle page avec les informations de la réservation.

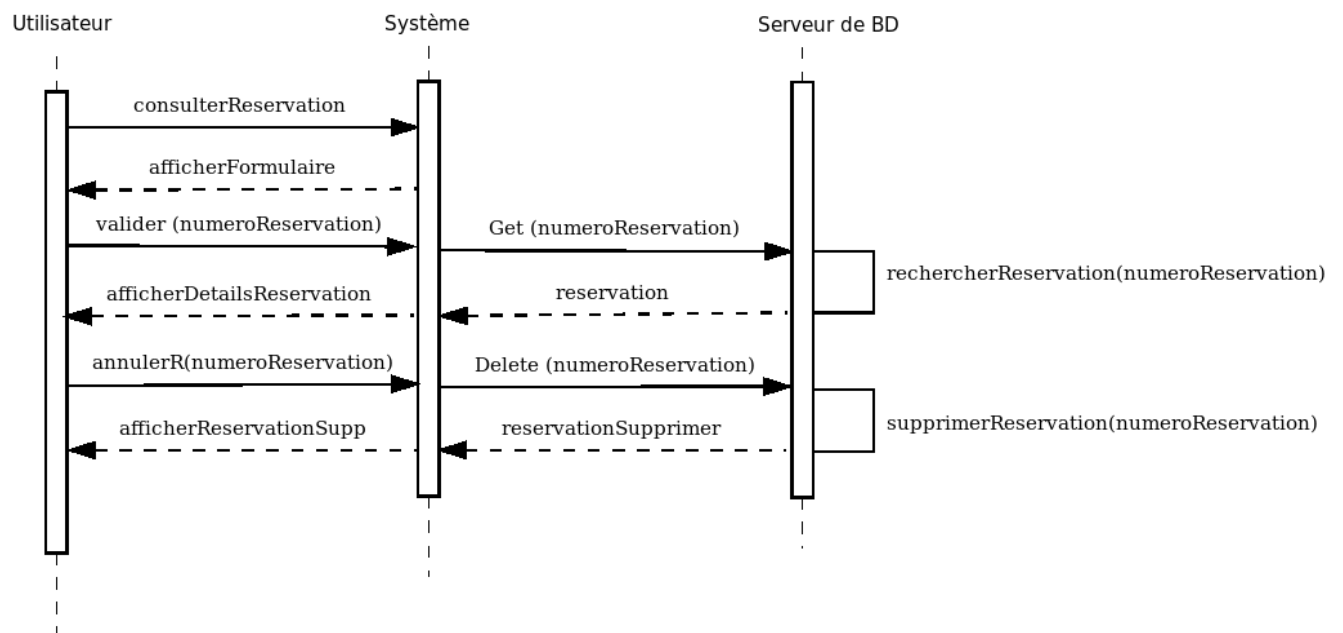
Diagramme de séquence correspondant:



- Nom du cas d'utilisation: Annuler une réservation
- But: Annuler une réservation.
- Acteur: Client/passager, Serveur de base de données
- Pré-condition: La réservation doit exister
- Déclencheur: Un bouton "Annuler la réservation"
- Post-condition: La réservation est supprimée
- Scénario:
  - L'utilisateur clique sur le bouton "Consulter une réservation"
  - le système affiche une boîte de dialogue avec un champs de saisie
  - L'utilisateur renseigne le numéro de réservation et clique sur le bouton "valider"
  - Le système envoi la requête au serveur de BD
  - Le serveur de BD traite la requête et retourne la réservation

- Le système affiche une nouvelle page avec les informations de la réservation.
- Le client clique sur le bouton "Annuler la réservation"
- Le système envoie la requête de suppression au serveur de BD
- Le serveur de BD supprime la réservation et retourne un message
- Le système affiche le message de suppression

Diagramme de séquence correspondant

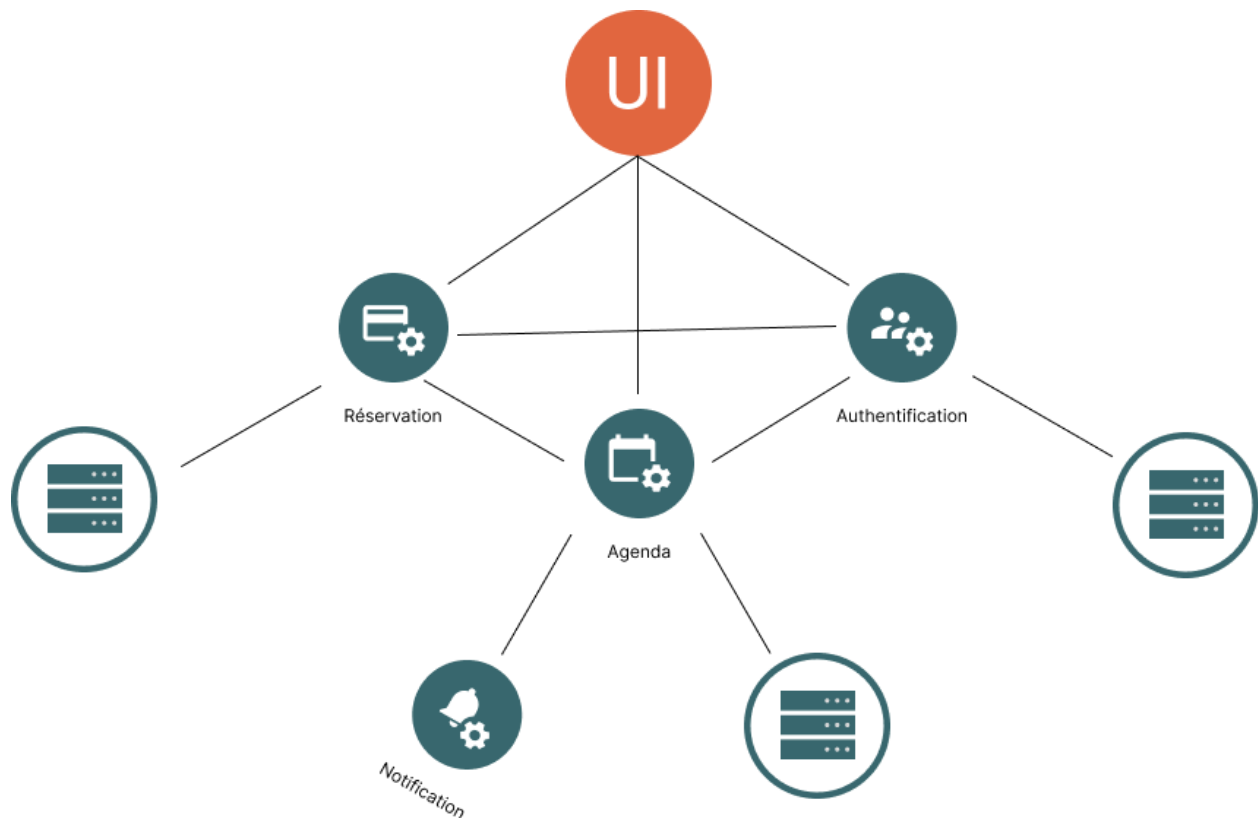


### 3. Mise en place de l'architecture logicielle

Compte tenu des paramètres suivants:

- Le produit sera amené à évoluer en fonction des besoins de la société
- Le produit est une application au grand public, la maintenance doit être simple et facile à gérer et les services doivent être indépendants pour minimiser les surcharges dû au nombre important de requêtes.

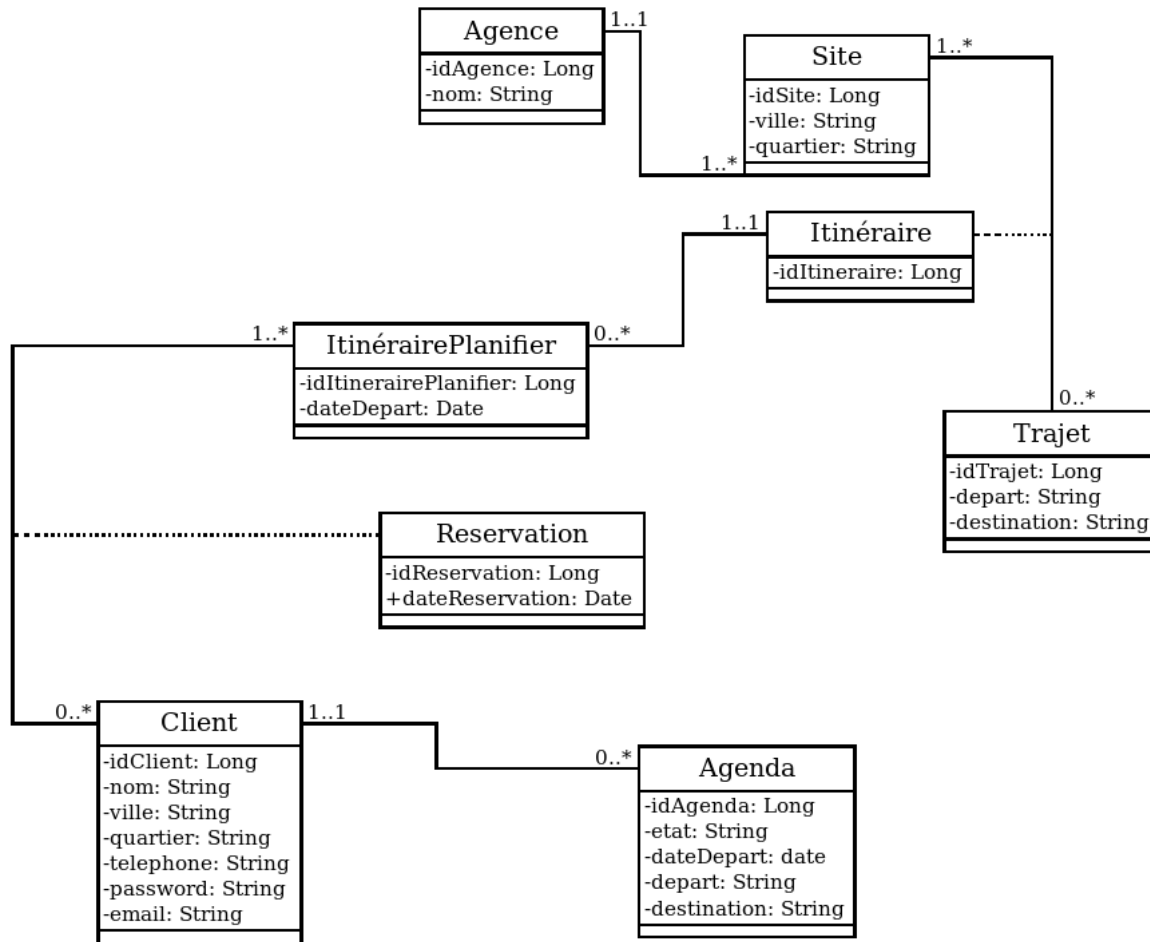
Nous choisissons de mettre en place une architecture microservice



#### 4. Diagramme de classe & schéma relationnel

Classe	Description	Attributs
Agence	Correspond au nom global d'une agence, à qui sera attaché des site	<ul style="list-style-type: none"> <li>- idAgence</li> <li>- nom</li> </ul>
Site	Agence concrète dans laquelle seront effectuées les réservations	<ul style="list-style-type: none"> <li>- idSite</li> <li>- ville</li> <li>- quartier</li> </ul>
Trajet	Entité caractérisé par un lieu de départ et un lieu d'arrivé	<ul style="list-style-type: none"> <li>- idTrajet</li> <li>- destination</li> <li>- depart</li> </ul>
Itinéraire	Classe qui définit les trajet d'un site	<ul style="list-style-type: none"> <li>- idItinéraire</li> </ul>
ItinérairePlanifié	Permet de programmer les voyages du côté des sites	<ul style="list-style-type: none"> <li>- idItinérairePlanifie</li> <li>- dateDepart</li> </ul>

Réservation	Définit la structure d'une réservation effectuée en ligne	<ul style="list-style-type: none"><li>- dateReservation</li><li>- idReservation</li><li>- nbrPlace</li><li>- nbrSiege</li></ul>
Agenda	Permet de définir un planning de voyage du côté des passagers	<ul style="list-style-type: none"><li>- idAgenda</li><li>- etat</li><li>- dateDepart</li><li>- depart</li><li>- destination</li></ul>
Client	Classe définissant la structure d'un utilisateur primaire dans le système; celui-là qui enclenche le processus de réservation	<ul style="list-style-type: none"><li>- idClient</li><li>- nom</li><li>- ville</li><li>- quartier</li><li>- telephone</li><li>- password</li><li>- email</li></ul>



## Schéma relationnel

- AGENCES (**idAgence**, nom)
- SITES (**idSite**, #idAgence, ville, quartier)
- TRAJETS (**idTrajet**, depart, destination)
- ITINERAIRES (**idItineraire**, #idTrajet, #idSite)
- ITINERAIRESPLANIFIER (**idItinerairePlanifier**, #idItineraire, dateDepart)
- RESERVATIONS (**idReservation**, #idItinerairePlanifier, #idClient, dateReservation)
- AGENDAS (**idAgenda**, #idClient, etat, dateDepart, depart, destination)
- CLIENTS (**idClient**, nom, ville, quartier, telephone, password, email)

## 5. Définition de l'environnement de développement

- Système d'exploitation: Linux (Ubuntu 22)

- Frontend: React native, Expo (Typescript)
- Backend: Spring Boot (java)
- SGBD: H2 Database
- Gestion de version de code: Github
- Gestion des containers d'images: Docker

## B. Application mobile de vérification de réservations

### 1. Acteurs et cas d'utilisations

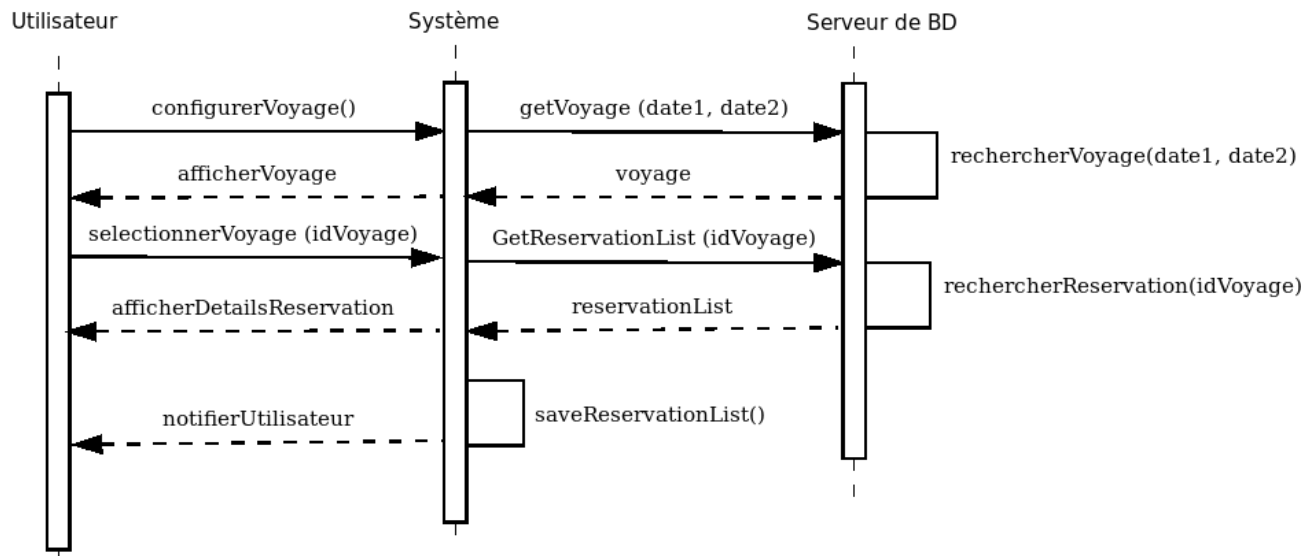
Acteurs	Description	Cas d'utilisations
Personnel Agence	Personnel particulier d'une agence chargé des vérifications des réservations	<ul style="list-style-type: none"> <li>- Scanner une réservation</li> <li>- Configurer un voyage</li> </ul>
Serveur de base de données	Outil permettant d'accéder à la base de données	/

### 2. Description des principaux cas d'utilisations (textuelle et diagramme de séquence)

- Nom du cas d'utilisation: Configurer un voyage
- But: Sélectionner le voyage pour lequel on voudrait vérifier les réservations.
- Acteur: Personnel agence (utilisateur), Serveur de base de données
- Pré-condition: Le voyage doit exister..
- Déclencheur: Un bouton "Configurer le voyage"
- Post-condition: La vérification des réservations peut commencer.
- Scénario:
  - L'utilisateur clique sur le bouton "Configurer le voyage"
  - Le système envoi une requête au serveur de BD pour récupérer le 1er voyage dont la date de départ est comprise dans un certain intervalle
  - Le serveur de BD traite la requête et retourne le résultat.

- Le système affiche une page avec le voyage retour.
- L'utilisateur sélectionne le voyage
- Le système envoie la requête au serveur de BD pour récupérer et stocker en local les réservations de ce voyage.
- Le système notifie l'utilisateur.

Diagramme de séquence correspondant:

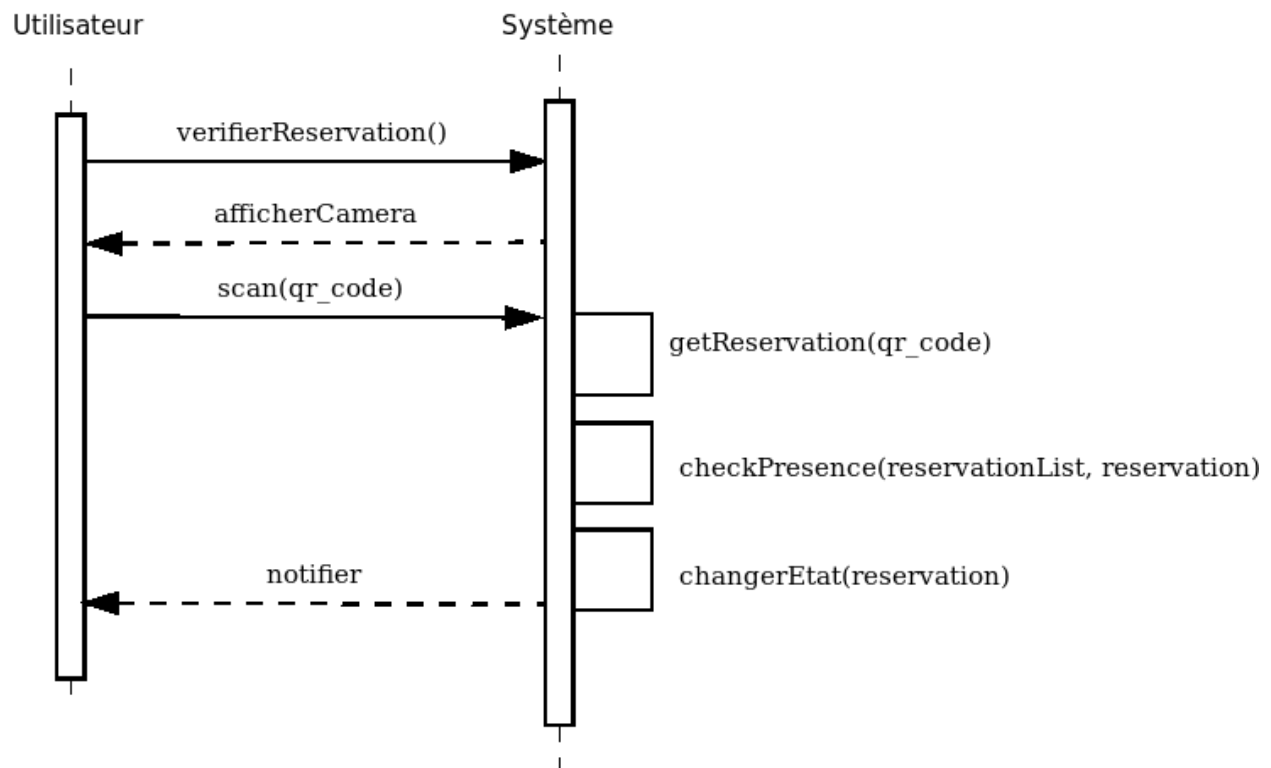


- Nom du cas d'utilisation: Vérifier une réservation
- But: Vérifier si le voyage associé à une réservation est identique au voyage configuré.
- Acteur: Personnel agence (utilisateur)
- Pré-condition: Le voyage doit être configuré.
- Déclencheur: Un bouton "Vérifier une réservation"
- Post-condition: La réservation est valide
- Scénario:
  - L'utilisateur clique sur le bouton "Vérifier une réservation"
  - Le système affiche une page avec la vue de l'appareil photo
  - L'utilisateur positionne l'appareil photo sur le Qr code de la réservation
  - Le système scan le Qr code et récupérer la réservation



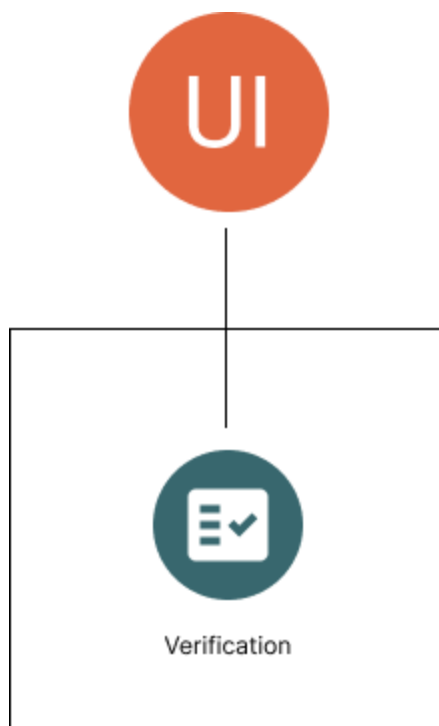
- Le système vérifie que la réservation récupérée est dans la liste de voyages qui a été stocké en local
- Le système change l'état de la réservation
- Le système notifie l'utilisateur.

Diagramme de séquence correspondant:



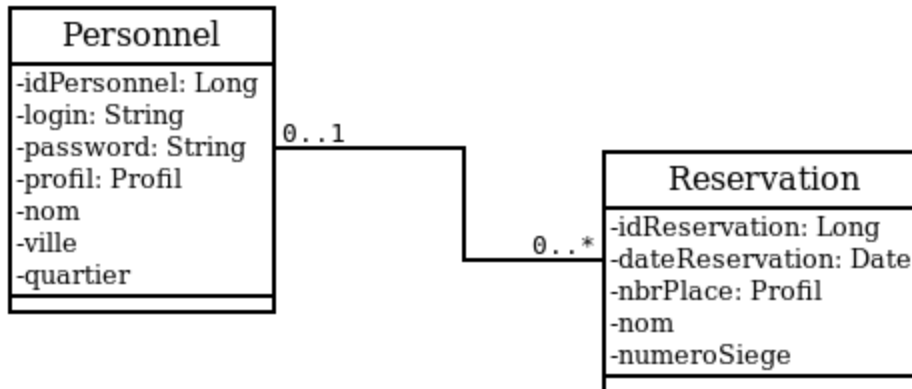
### 3. Mise en place de l'architecture logicielle

Le produit est léger et possède peu de fonctionnalités, d'où la mise en place d'une architecture monolithique.



#### 4. Diagramme de classe et schéma relationnel

Classe	Description	Attributs
Personnel agence	Classe représentant la structure du personnel d'une agence chargé des vérifications des réservations	<ul style="list-style-type: none"> <li>- idPersonnel</li> <li>- login</li> <li>- password</li> <li>- profil</li> <li>- nom</li> <li>- ville</li> <li>- quartier</li> </ul>
Réservation	Définit la structure d'une réservation effectuée en ligne	<ul style="list-style-type: none"> <li>- dateReservation</li> <li>- idReservation</li> <li>- nbrPlace</li> <li>- numerosSiege</li> <li>- nom</li> </ul>



### Schéma relationnel

- PERSONNELS (idPersonnel, nom, #profil, password, login, ville, quartier)
- RESERVATIONS (idReservation, #idPersonnel, dateReservation, nbrPlace, numerosSiege, nom)

### 5. Définition de l'environnement de développement

- Système d'exploitation: Linux (Ubuntu 22)
- Frontend: React native, Expo (Typescript)
- Backend: Spring Boot (java)
- SGBD: H2 Database
- Gestion de version de code: Github
- Gestion des containers d'images: Docker

## C. Application web de gestion des trajets

### 1. Acteurs et cas d'utilisations

Acteurs	Description	Cas d'utilisations
Gestionnaire	Personnel particulier de l'agence chargé de la planifications des itinéraires	<ul style="list-style-type: none"> <li>- Consulter les détails d'un membre du personnel</li> <li>- Consulter les statistiques</li> </ul>

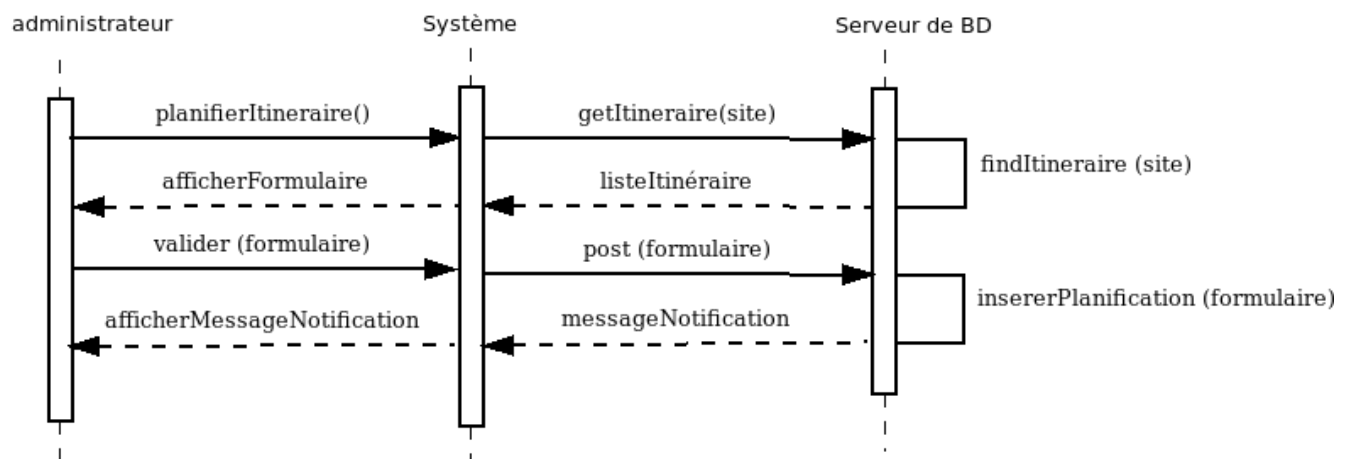
		<ul style="list-style-type: none"> <li>- Consulter le planning de voyage</li> <li>- Enregistrer une réservation</li> </ul>
Administrateur	/	<ul style="list-style-type: none"> <li>- Consulter les détails d'un membre du personnel</li> <li>- Lister le personnel</li> <li>- Ajouter, modifier, supprimer un membre du personnel</li> <li>- Ajouter, modifier, supprimer et lister les profils</li> <li>- Assigner des droits à chaque profils</li> <li>- Consulter les statistiques</li> <li>- Créer, modifier, supprimer et lister les sites de l'agence</li> <li>- Planifier un voyage</li> <li>- Consulter le planning de voyage.</li> <li>- Générer les rapports d'analyse</li> <li>- Ajouter, modifier, supprimer et lister les itinéraires</li> </ul>
Serveur de BD	Composant permettant d'accéder à la base de données	/

## 2. Description des principaux cas d'utilisations (textuelle et diagramme de séquence)

- Nom du cas d'utilisation: Planifier un itinéraire
- But: Définir une date de départ pour un itinéraire
- Acteur: Administrateur, serveur de BD
- Précondition: Le l'itineraire doit exister
- Déclencheur: Un bouton "planifier un itinéraire"

- Post-condition: Les passagers peuvent réserver pour cet itinéraire
- Scénario:
  - L'administrateur clique sur le bouton "Planifier un itinéraire"
  - Le système envoie une requête au serveur de BD pour récupérer tous les itinéraires du site.
  - Le serveur de BD récupère en BD les itinéraires du site et retourne le résultat.
  - Le système affiche une page avec un formulaire
  - L'administrateur sélectionne l'itinéraire, remplit le formulaire et valide
  - Le système envoie la requête d'ajout au serveur de BD pour ajouter la planification
  - Le serveur de BD insère la planification en BD et retourne le message de succès.
  - Le système notifie l'administrateur.

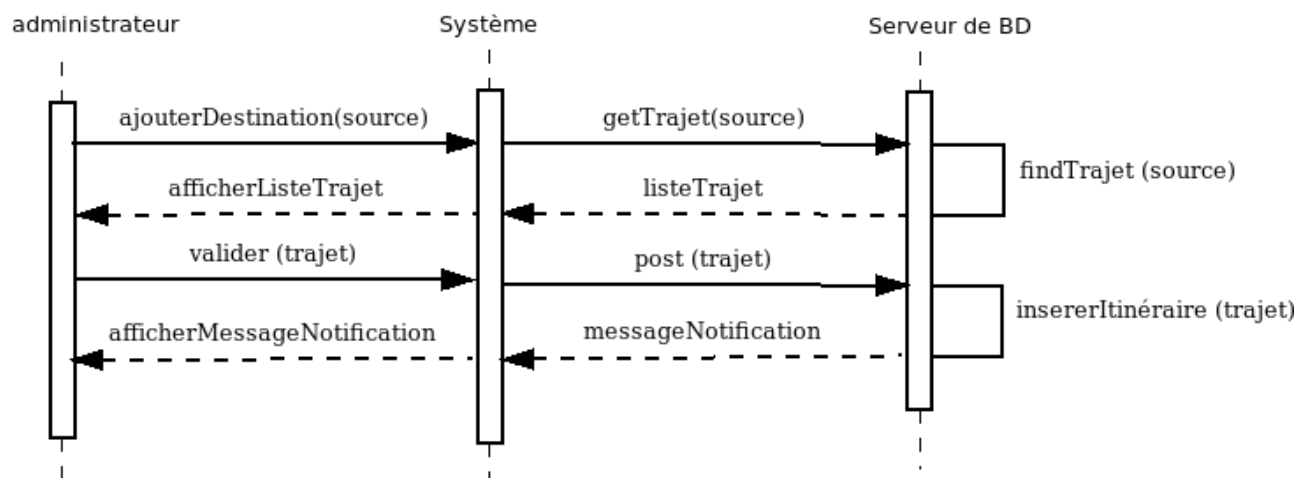
Diagramme de séquence correspondant



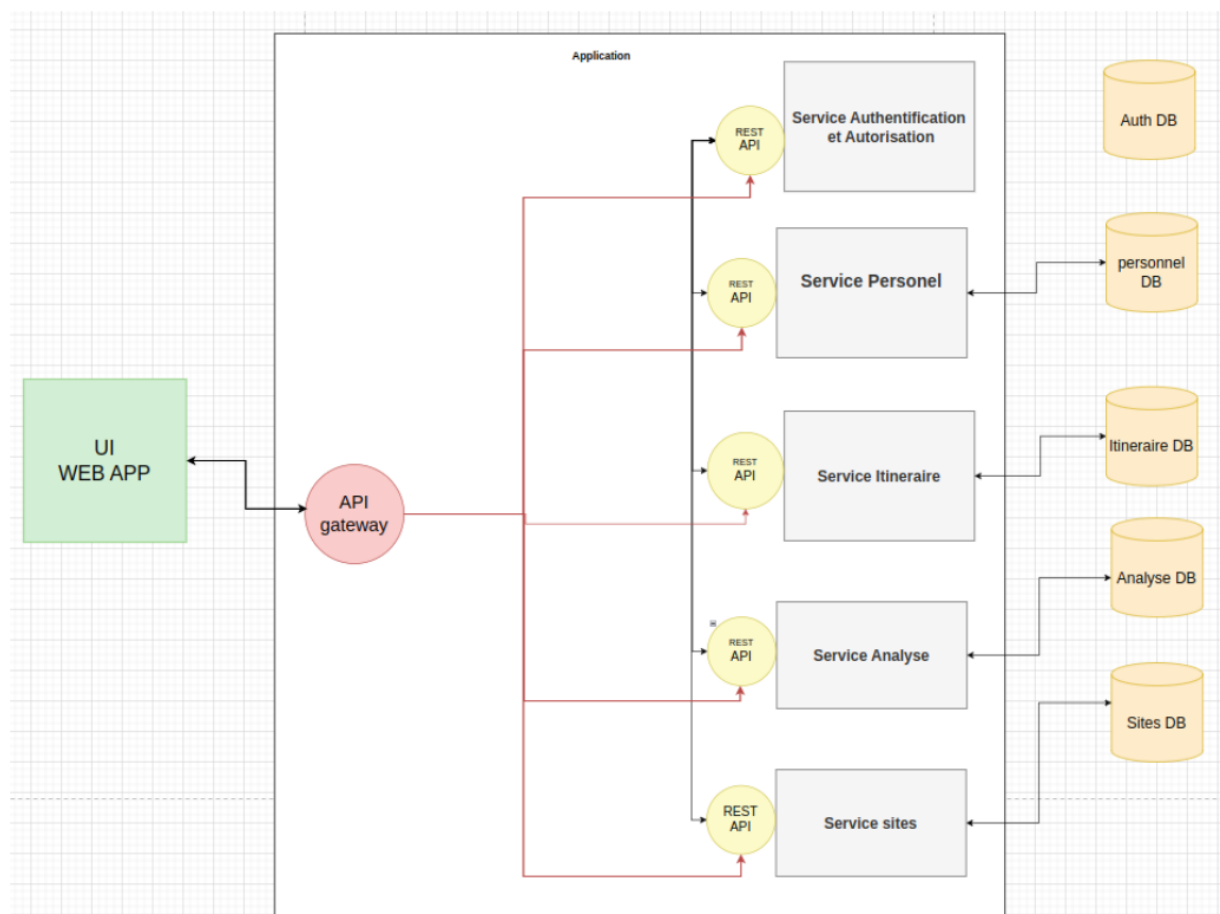
- Nom du cas d'utilisation: Ajouter un itinéraire
- But: Ajouter un trajet à une agence
- Acteur: Administrateur, serveur de BD
- Précondition: Le trajet doit exister

- Déclencheur: Un bouton "Ajouter une destination"
- Post-condition: L'itinéraire est visible dans le système
- Scénario:
  - L'administrateur clique sur le bouton "Ajouter une destination"
  - Le système envoie la requête au serveur de BD pour récupérer tous les trajets dont source est la ville où est situé le site
  - Le serveur de BD recherche les trajets en fonction de la source et retourne le résultat.
  - Le système affiche les destinations
  - L'administrateur sélectionne une destination et valide
  - Le système envoie la requête d'ajout au serveur de BD
  - Le serveur de BD insère l'itinéraire en BD et retourne le message d'ajout
  - Le système notifie l'administrateur

Diagramme de séquence correspondant



### 3. Mise en place de l'architecture logicielle

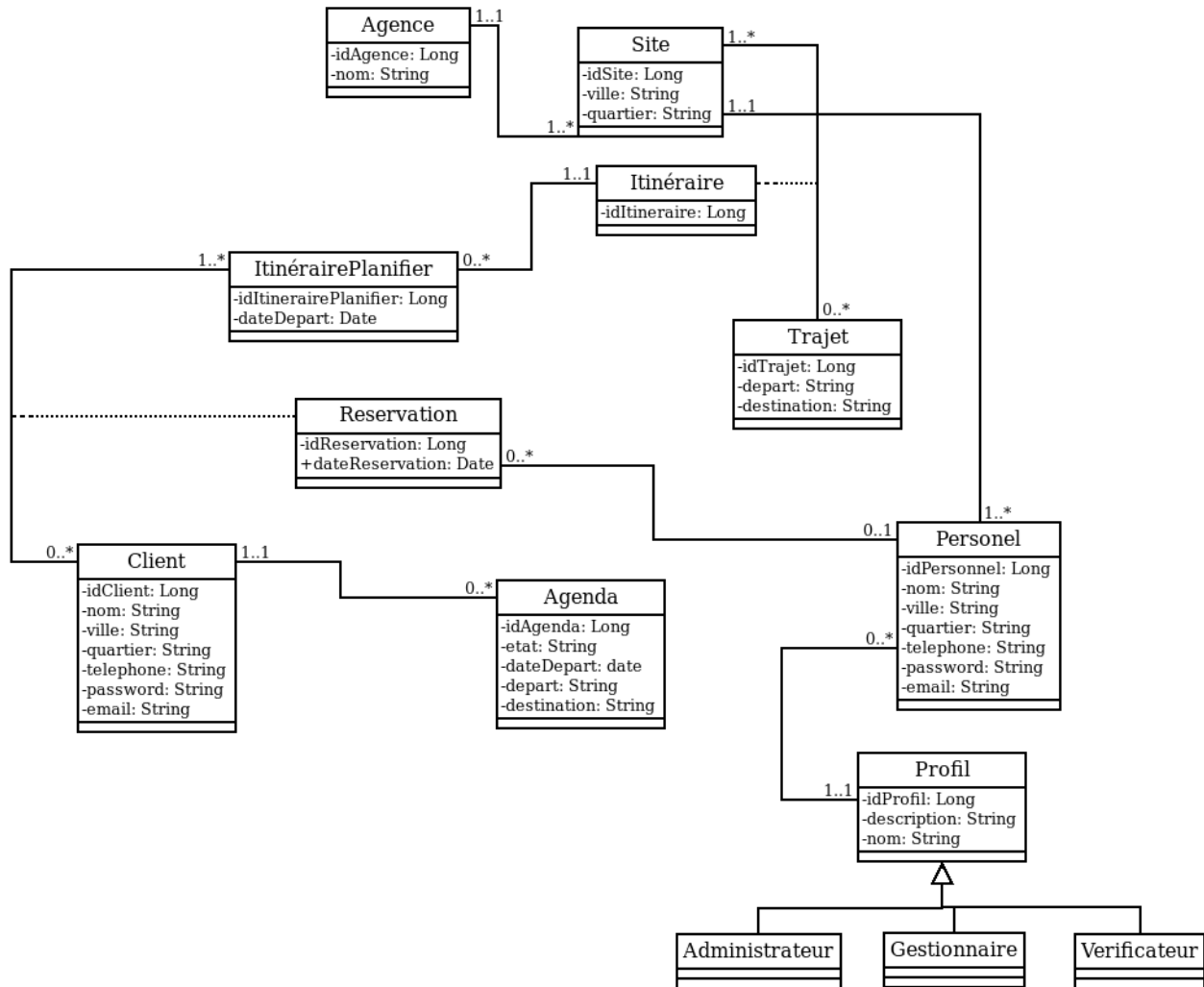


#### 4. Diagramme de classe et schéma relationnel

Classe	Description	Attributs
Agence	Correspond au nom global d'une agence, à qui sera attaché des site	<ul style="list-style-type: none"> <li>- idAgence</li> <li>- nom</li> </ul>
Site	Agence concrète dans laquelle seront effectuées les réservations	<ul style="list-style-type: none"> <li>- idSite</li> <li>- ville</li> <li>- quartier</li> </ul>
Trajet	Entité caractérisé par un lieu de départ et un lieu d'arrivé	<ul style="list-style-type: none"> <li>- idTrajet</li> <li>- destination</li> <li>- depart</li> </ul>
Itinéraire	Classe qui définit les trajet d'un site	<ul style="list-style-type: none"> <li>- idItinéraire</li> </ul>
ItinérairePlanifié	Permet de programmer les voyages du côté des sites	<ul style="list-style-type: none"> <li>- idItinérairePlanifie</li> </ul>

		<ul style="list-style-type: none"> <li>- dateDepart</li> </ul>
Réservation	Définit la structure d'une réservation effectuée en ligne	<ul style="list-style-type: none"> <li>- dateReservation</li> <li>- idReservation</li> <li>- nbrPlace</li> <li>- nbrSiege</li> </ul>
Agenda	Permet de définir un planning de voyage du côté des passagers	<ul style="list-style-type: none"> <li>- idAgenda</li> <li>- etat</li> <li>- dateDepart</li> <li>- depart</li> <li>- destination</li> </ul>
Client	Classe définissant la structure d'un utilisateur primaire dans le système; celui-là qui enclenche le processus de réservation	<ul style="list-style-type: none"> <li>- idClient</li> <li>- nom</li> <li>- ville</li> <li>- quartier</li> <li>- telephone</li> <li>- password</li> <li>- email</li> </ul>
Personnel	Classe représentant le Personnel particulier de l'agence chargé de la définition et la planification des itinéraires.	<ul style="list-style-type: none"> <li>- idPersonnel</li> <li>- nom</li> <li>- ville</li> <li>- quartier</li> <li>- telephone</li> </ul>
Profil	Modèle qui définit la structure d'un profil	<ul style="list-style-type: none"> <li>- idProfil</li> <li>- nom</li> <li>- description</li> </ul>





## Schéma relationnel

- AGENCES (**idAgence**, nom)
- SITES (**idSite**, #idAgence, ville, quartier)
- TRAJETS (**idTrajet**, depart, destination)
- ITINERAIRES (**idItineraire**, #idTrajet, #idSite)
- ITINERAIRESPLANIFIER (**idItinerairePlanifier**, #idItineraire, dateDepart)
- RESERVATIONS (**idReservation**, #idItinerairePlanifier, #idClient, dateReservation)
- AGENDAS (**idAgenda**, #idClient, etat, dateDepart, depart, destination)
- CLIENTS (**idClient**, nom, ville, quartier, telephone, password, email)
- PERSONNEL (**idPersonnel**, #idSite, #idProfil, nom, ville, quartier, telephone, password, email).

- PROFILS (idProfil, description, nom)

## 5. Définition de l'environnement de développement

- Système d'exploitation: Linux (Ubuntu)
- Frontend: Angular (Typescript)
- Backend: Spring Boot (java)
- SGBD: H2 Database
- Gestion de version de code: Github
- Gestion des containers d'images: Docker

### D. Dashboard web d'administration

#### 1. Acteurs et cas d'utilisations

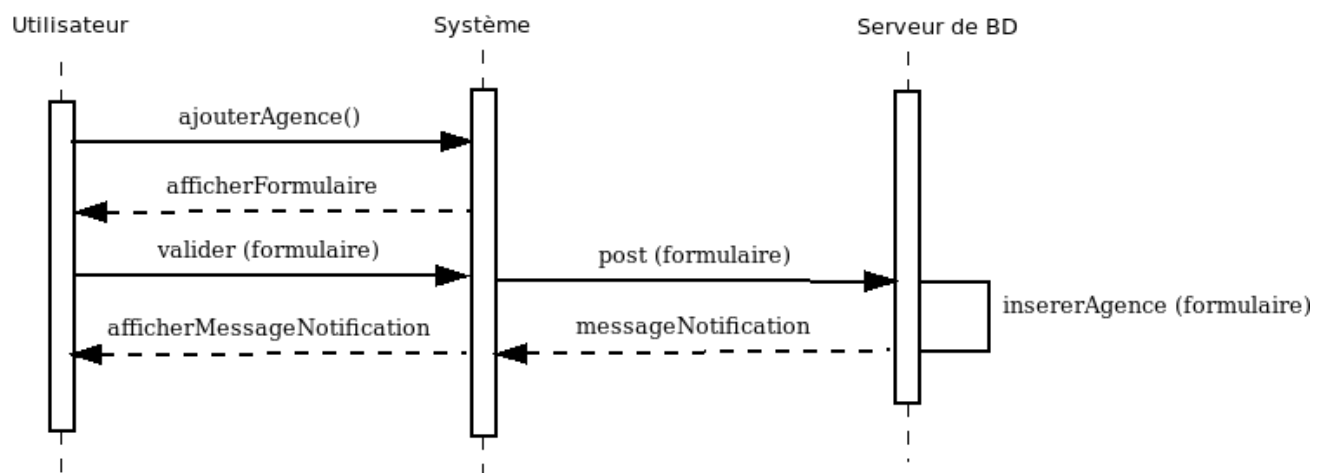
Acteurs	Description	Cas d'utilisations
Administrateurs	ensemble des administrateurs des agences de voyages, y compris l'équipe de développement	<ul style="list-style-type: none"> <li>- Ajouter, modifier, supprimer et lister les agences de voyage et leurs sites</li> <li>- Consulter les détails d'une agence de voyage</li> <li>- Afficher les statistiques</li> </ul>
Serveur de base de données	Outil permettant d'accéder à la base de données	/

#### 2. Description des principaux cas d'utilisations (textuelle et diagramme de séquence)

- Nom du cas d'utilisation: Ajouter une agence de voyage
- But: Permettre à l'agence d'être visible dans le système
- Acteur: Administrateur, serveur d BD
- Précondition: L'agence doit être existante

- Déclencheur: Un bouton "Ajouter une agence"
- Post-condition: L'agence est visible dans le système
- Scénario:
  - L'administrateur clique sur le bouton "Ajouter une agence"
  - Le système affiche une page avec un formulaire
  - L'administrateur remplit le formulaire et valide
  - Le système envoie la requête d'ajout au niveau du serveur de BD
  - Le serveur de BD insère l'agence dans la base de données et retourne un message
  - Le système affiche le message de création à l'administrateur

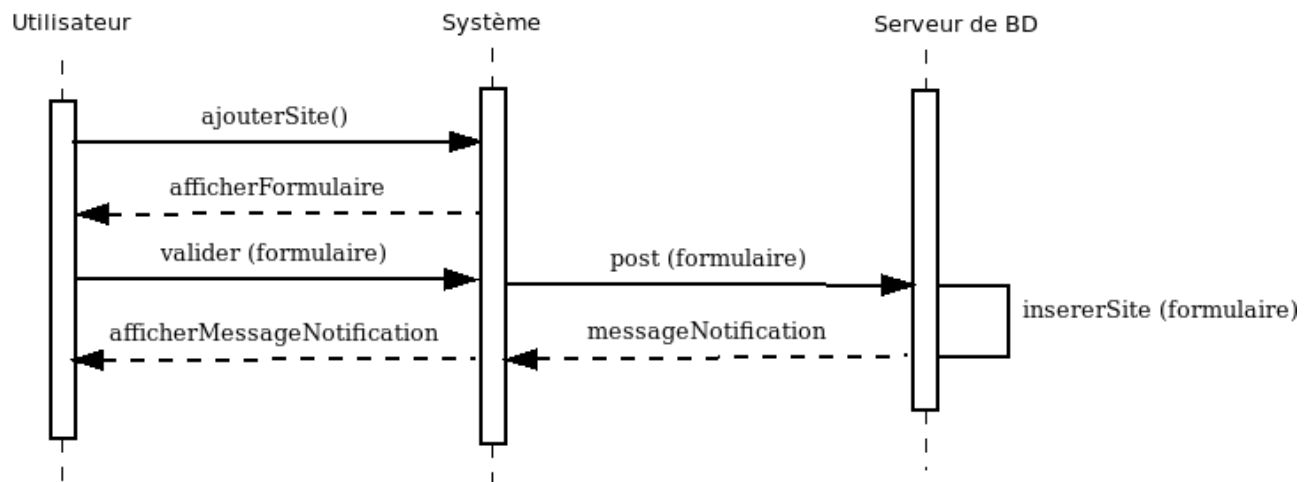
Diagramme de séquence correspondant



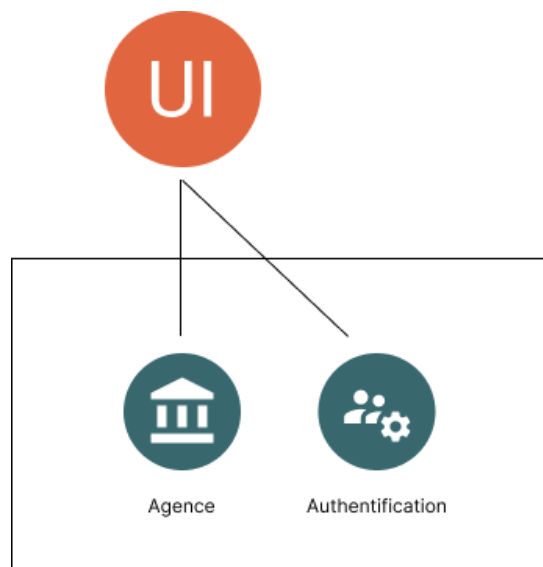
- Nom du cas d'utilisation: Ajouter un site à une agence
- But: Permettre à l'agence de gérer leur trajet en fonction de leur sites
- Acteur: Administrateur, serveur d BD
- Précondition: L'agence doit être existante dans le système
- Déclencheur: Un bouton "Ajouter un site"
- Post-condition: Le site de l'agence est visible dans le système
- Scénario:
  - L'administrateur clique sur le bouton "Ajouter un site"

- Le système affiche une page avec un formulaire
- L'administrateur remplit le formulaire; sélectionne l'agence et valide
- Le système envoie la requête d'ajout au niveau du serveur de BD
- Le serveur de BD insère le site dans la base de données et retourne un message
- Le système affiche le message de création à l'administrateur

Diagramme de séquence correspondant

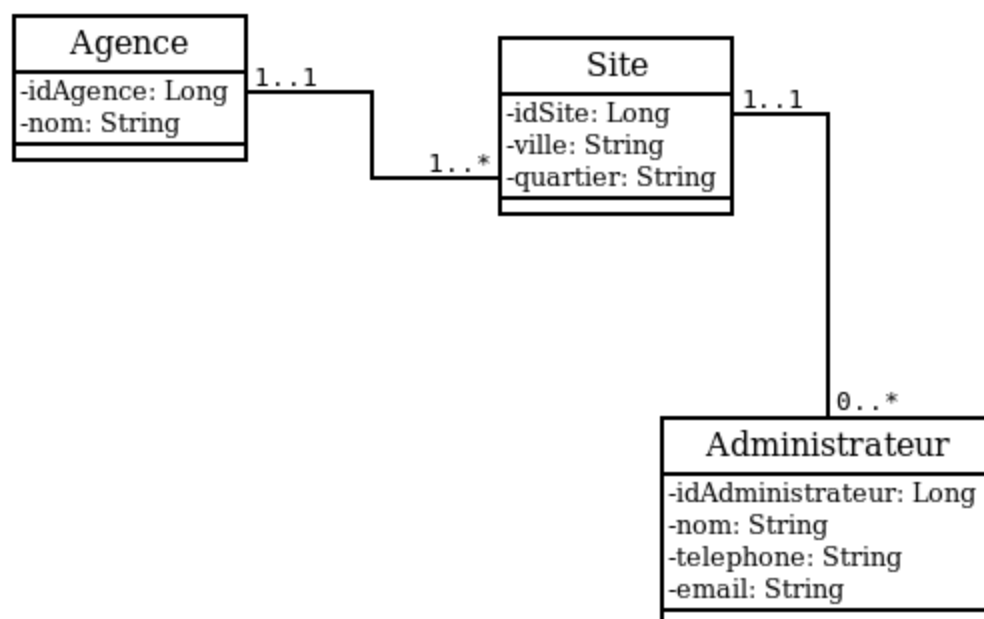


### 3. Mise en place de l'architecture logicielle



### 4. Diagramme de classe et schéma relationnel

Classe	Description	Attributs
Administrateur	Classe représentant la structure du personnel d'une agence chargé des vérifications des réservations	<ul style="list-style-type: none"> <li>- idPersonnel</li> <li>- login</li> <li>- password</li> <li>- profil</li> <li>- nom</li> <li>- ville</li> <li>- quartier</li> </ul>
Agence	Définit la structure d'une réservation effectuée en ligne	<ul style="list-style-type: none"> <li>- dateReservation</li> <li>- idReservation</li> <li>- nbrPlace</li> <li>- numerosSiege</li> <li>- nom</li> </ul>
Site		



### Schéma relationnel

- AGENCES (idAgence, nom)
- SITES (idSite, #idAgence, #idAdministrateur, ville, quartier)
- ADMINISTRATEURS (idAdministrateur, nom, telephone, email)

## 5. Définition de l'environnement de développement

- Système d'exploitation: Linux (Ubuntu)
- Frontend: React (Javascript)
- Backend: Spring Boot (java)
- SGBD: H2 Database
- Gestion de version de code: Github
- Gestion des containers d'images: Docker

## III. PRODUCT BACKLOG

### A. Application mobile de réservation en ligne

1. Mettre en place l'environnement de développement
  - Installer les outils
  - Créer et configurer le projet (frontend et backend)
  - Faire la connexion a la base de données
2. Réserver (payer) un billet de voyage
  - Concevoir la page de réservation
  - Implémenter le design de la page de réservation
  - Définir le modèle d'une réservation
  - Écrire les algorithmes de paiement et les implémenter
  - Intégrer les API de paiements
  - Tester
3. Consulter les détails d'une réservation
  - Concevoir la page des détails d'une réservation
  - Implémenter le design de la page des détails
  - Écrire les algorithmes de récupération en base de données et les implémentés
  - Tester
4. Annuler une réservation
  - Ajouter un bouton 'Annuler la réservation' sur la page de détails d'une réservation (2)

- Écrire les algorithmes d'annulation de réservation (changement de l'état de la réservation, remboursement) et les implémenter.
  - Tester
5. Consulter la liste des agences de voyage et leurs sites.
    - Concevoir le squelette de la page d'accueil
    - Ajouter la section exploration
    - Concevoir les composants de présentation et de détails (agence, site) et les implémenter sur la page d'accueil.
    - Écrire les codes de récupération au niveau du backend.
    - Tester.
  6. Rechercher une agence de voyage
    - Ajouter l'option de recherche sur la page d'accueil.
    - Concevoir le formulaire de recherche
    - Ecrire le code de recherche avec filtre d'une agence
    - Tester
  7. Rechercher un site
    - Ecrire le code de recherche avec filtre d'une site
    - Intégrer ce code sur le composant de recherche (5)
    - Tester
  8. Rechercher une itineraire
    - Ecrire le code de recherche avec filtre des itinéraires
    - Intégrer ce code sur le composant de recherche (5)
    - Tester
  9. Rechercher un voyage
    - Ecrire le code de recherche avec filtre des voyages
    - Intégrer ce code sur le composant de recherche (5)
    - Tester
  10. Consulter les détails d'un voyage
    - Concevoir et implémenter la page de détails d'un voyage
    - Ecrire les codes de recuperation des donnees
    - Tester
  11. Consulter l'historique de réservation
    - Concevoir le design de la page d'historique et implémenter

- Ecrire les codes de recuperation de donnees et tester
- Tester

#### 12. Définir un agenda de voyage

- Concevoir le design de la page agenda
- Implémenter le design
- Ecrire le code d'ajout et d'affichage
- Tester

### **B. Application mobile de vérification des réservations**

#### 1. Mettre en place l'environnement de développement

- Installer les outils
- Créer et configurer le projet (frontend et backend)
- Faire la connexion a la base de données

#### 2. Scanner une réservation

- Mettre en place l'environnement de développement
- Concevoir la maquette de la page d'accueil et implémenter
- Concevoir la maquette de la page de caméra et implémenter
- Concevoir le composant de notification
- Ecrire et implémenter les algorithmes de scan
- Tester

#### 3. Configurer un voyage

- Ajouter le menu "configurer le voyage" sur la page d'accueil
- Concevoir les maquettes des composants (liste des voyage, details du voyage, notification) et les implémenter
- Implémenter les fonctions correspondantes et tester.

### **C. Application web de gestion des trajets**

#### 1. Mettre en place l'environnement de développement

- Installer les outils
- Créer et configurer le projet (frontend et backend)
- Faire la connexion a la base de données

#### 2. Ajouter, modifier, supprimer et lister les membres du personnel



- Concevoir le design des pages d'ajout, de modification de suppression et d'affichage de la liste des membres du personnel
  - Création du modèle de donnée d'un membre du personnel
  - Ecrire les code CRUD et tester
  - Tester sur les pages
3. Consulter les détails d'un membre du personnel
- Ajouter un bouton "voir" sur chaque ligne de la liste des membres du personnel
  - Concevoir le design de la page des détails et implémenter.
  - Tester
4. Ajouter, modifier, supprimer et lister les profils
- Concevoir le design des pages d'ajout, de modification de suppression et d'affichage des profil
  - Création du modèle de donnée d'un profil
  - Ecrire les code CRUD et tester
  - Tester sur les pages
5. Assigner des droits à chaque profils
- Concevoir la page de détails d'un profil
  - Ecrire la fonction qui permet de recuperer un profil et tester
  - Ecrire le code qui permet d'ajouter un droit à un profil et tester
  - Ecrire les code de vérification des accès d'un profil et tester sur le système
6. Ajouter, supprimer et lister les itinéraires
- Concevoir le design des pages d'ajout, de suppression et d'affichage des itinéraires
  - Création du modèle de donnée pour une itinéraire
  - Ecrire les code d'ajout, de suppression, d'affichage et tester
  - Tester sur les pages
7. Planifier un voyage
- Concevoir le formulaire de planification
  - Créer le modèle de données pour un voyage
  - Ecrire les codes de planification et tester
  - Intégrer les codes de planifications sur la page et tester.
8. Consulter le planning de voyage.
- Concevoir la page d'accueil des plannings de voyage
  - Implémenter

- Ecrire les code d’affichage et tester
- Tester le code d’affichage sur la page
- 9. Consulter les statistiques
  - Concevoir le design du tableau de bord et implémenter
  - Ecrire les codes de récupération et d’analyse des données et tester
  - Afficher les données sur le tableau de bord
- 10. Générer les rapports d’analyse
  - Ajouter un bouton “Générer un rapport d’analyse” sur le tableau de bord (9)
  - Ecrire la fonction qui permet de générer un rapport d’analyse
  - Tester
- 11. Créer, modifier, supprimer et lister les sites de l’agence //
- 12. Enregistrer une réservation
  - Concevoir le design du formulaire d’enregistrement d’une réservation
  - implémenter le design
  - Créer le modèle de donnée pour représenter une réservation
  - Ecrire la requête d’ajout au niveau du frontend et la fonction d’insertion au niveau du backend et tester
  - Tester cs fonctions sur la page

#### **D. Dashboard web d’administration**

1. Mettre en place l’environnement de développement
  - Installer les outils
  - Créer et configurer le projet (frontend et backend)
  - Faire la connexion a la base de données
2. Ajouter, modifier, supprimer et lister les agences de voyage
  - Concevoir le design des page d’ajout, de modification, de liste et la boîte de dialogue de suppression
  - Implémenter le design
  - Créer le modèle d’une agence
  - Ecrire les code CRUD pour une agence et les intégrés sur les pages
  - Tester
3. Consulter les détails d’une agence de voyage
  - Ajouter un bouton “voir” sur les lignes de la liste des agences (2)

- Concevoir le design de la page de détails d'une agence et implémenter.
  - Ecrire le code d'affichage des informations
  - Tester
4. Afficher les statistiques du système
- Concevoir le design du tableau de bord et implémenter
  - Ecrire les codes de récupération et tester
  - Afficher les informations récupérées sur la page

## IV. DÉFINITION DES SPRINTS

18 sprints à partir du 3 janvier (avec prévision du temps pour la session normale)

Nos sprints seront organisés chacun sur un intervalle de 7 jours. Ce qui fera environ 18 sprints à partir du 3 Janvier 2024. Conformément à la méthodologie Scrum, les daily meetings sont effectués chaque matin (en ligne) et à la fin de chaque sprint, un sprint review (en présentiel) est organisé pour valider les fonctionnalités développées.

Sprints	Période	Product goals
1	03/01/2024 - 09/01/2024	<ul style="list-style-type: none"> <li>- Mettre en place l'environnement de développement (App mobile de vérification)</li> <li>- Mettre en place l'environnement de développement (App web de gestion des trajets)</li> <li>- Mettre en place l'environnement de développement (App web dashboard d'administration).</li> </ul>
2	10/01/2024 - 17/01/2024	<ul style="list-style-type: none"> <li>- Réserver (payer) un billet de voyage</li> </ul>

		<ul style="list-style-type: none"> <li>- Consulter les détails d'une réservation</li> <li>- Consulter les détails d'un voyage</li> </ul>
3	18/01/2024 - 24/01/2024	<ul style="list-style-type: none"> <li>- Annuler une réservation</li> <li>- Consulter la liste des agences de voyage et leurs sites.</li> </ul>
4	25/01/2024 - 31/01/2024	<ul style="list-style-type: none"> <li>- Ajouter, supprimer et lister les itinéraires</li> <li>- Planifier un voyage</li> </ul>
5	01/02/2024 - 07/02/2024	<ul style="list-style-type: none"> <li>- Ajouter, modifier, supprimer et lister les profils</li> <li>- Ajouter, modifier, supprimer et lister les membres du personnel</li> </ul>
6	08/02/2024 - 14/02/2024	<ul style="list-style-type: none"> <li>- Configurer un voyage</li> <li>- Scanner une réservation</li> </ul>
7	15/02/2024 - 21/02/2024	<ul style="list-style-type: none"> <li>- Ajouter, modifier, supprimer et lister les agences de voyage</li> <li>- Consulter les détails d'une agence de voyage</li> </ul>
8	22/02/2024 - 28/02/2024	<ul style="list-style-type: none"> <li>- Rechercher une agence de voyage</li> <li>- Rechercher un site</li> <li>- Rechercher une itineraire</li> <li>- Rechercher un voyage</li> </ul>

9	01/03/2024 - 07/03/2024	<ul style="list-style-type: none"><li>- Consulter les détails d'un membre du personnel</li><li>- Assigner des droits à chaque profils</li></ul>
10	08/03/2024 - 14/03/2024	<ul style="list-style-type: none"><li>- Consulter le planning de voyage.</li><li>- Consulter les statistiques (App web de gestion des trajets)</li></ul>
11	15/03/2024 - 21/03/2024	<ul style="list-style-type: none"><li>- Afficher les statistiques (dashboard web)</li></ul>
12	22/03/2024 - 28/03/2024	<ul style="list-style-type: none"><li>- Consulter l'historique de réservation (App mobile)</li><li>- Définir un agenda de voyage</li></ul>
13	01/04/2024 - 07/04/2024	<ul style="list-style-type: none"><li>- Générer les rapports d'analyse</li><li>- Enregistrer une réservation</li></ul>
14 - 18	08/04/2024 - 30/05/2024	<ul style="list-style-type: none"><li>- Test d'intégration et correction des bugs</li></ul>