

# SITUATION INTÉGRATRICE

GESNEL KERRIAN   COLLIN ALEX   NAIGEON ADRIEN   TOUSSAINT JAROD  
GEORGES VALERIAN

## Description du sujet

### Informations générales

- **Équipe** : Les Charlemagnes
- **Titre** : Charl'Agenda

CharlAgenda est une application web qui permet de gérer plusieurs agendas et de les partager entre utilisateurs.

L'application propose plusieurs façons d'interagir : via l'interface graphique classique, par ligne de commande (curl sur l'API), ou en glissant-déposant des rendez-vous directement sur le calendrier.

Les utilisateurs peuvent créer des comptes, gérer plusieurs agendas simultanément, et les visualiser sous différents formats (jour, semaine, mois). Le partage d'agendas entre utilisateurs est facilité par un système de relations (amis) et de droits d'accès (par agenda).

L'application détecte automatiquement les conflits de rendez-vous et propose des outils de recherche avancés (par plage horaire, personnes impliquées, type de rendez-vous).

### Technologies

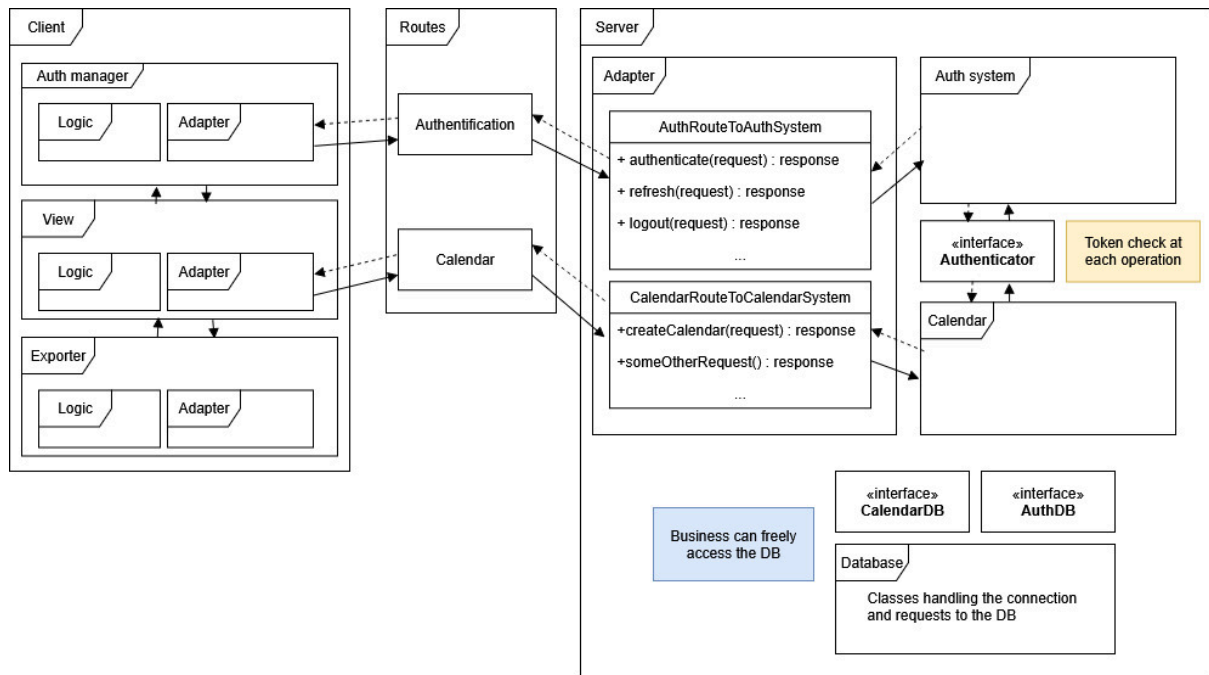
- **Backend**
  - [Node.js](#)
  - [Express.js](#)
  - Système d'authentification : **JWT**
  - Base de données : **MySQL** (pour le relationnel)
- **Frontend**
  - **Vue.js**
  - **Tailwind CSS**
- **Librairies**

## Fonctionnalités

### ⚠ Fonctionnalités minimales (MVP)

- **Gestion des utilisateurs :**
  - s'identifier ⚠
  - créer un compte
  - fermer la session (se déconnecter) ⚠
  - consulter son profil
  - ajouter une relation entre utilisateur (liste d'amis)
- **Gestion des agendas :**
  - créer un agenda ⚠
    - par formulaire
    - par ligne de commande
  - visualiser un nombre quelconque d'agendas ⚠
    - affichage du calendrier
    - visualisation par jour
    - visualisation par semaine
    - visualisation par mois
  - partager des agendas entre utilisateurs ⚠
  - annuler un partage d'agenda ⚠
  - importer un agenda ⚠
    - import via fichier .ics
    - import via URL
  - exporter un agenda ⚠
    - format iCalendar (.ics) (standard sur de nombreuses applications)
- **Gestion des rendez-vous :**
  - ajouter un rendez-vous ⚠
    - par ligne de commande (curl sur l'API)
    - par formulaire
    - par drag-and-drop
  - ajouter un rendez-vous récurrent (qui se répète tous les x temps) ⚠
  - supprimer un rendez-vous ⚠
  - modifier un rendez-vous ⚠
  - partager un rendez-vous avec un utilisateur
  - visualiser les conflits
- **Recherche et tri :**
  - rechercher un rendez-vous par différents critères ⚠
    - par plage horaire
    - personnes
    - type de rendez-vous
    - par tag

## Architecture



Nous avons prévu de construire l'application en suivant une architecture hexagonale. La logique métier (module "Calendar") va interagir avec le reste des composants (Authentication, base de donnée) uniquement via des interfaces et l'interaction avec les routes se fera par le biais d'adaptateurs pour que la logique métier n'ait pas à se soucier des conversions entre données locales et JSON pour les routes.

Le projet va fonctionner à l'aide de 3 routes :

- **/index.html** pour accéder au client
- **/calendar/** pour toutes les interactions métier (nécessite un token d'accès)
- **/auth/** pour toutes les interactions lié à l'authentification