

Documentation Technique – Projet Vite et Gourmand

Cyril BOCAGE

TP Développeur Web Web Mobile

Année 2026

SOMMAIRE

1. Introduction aux choix techniques
2. Modèle Conceptuel de Données (MCD)
3. Diagrammes UML Cas d'Utilisation (Use Case)
4. Documentation du déploiement
5. Gestion de projet

Introduction aux choix techniques

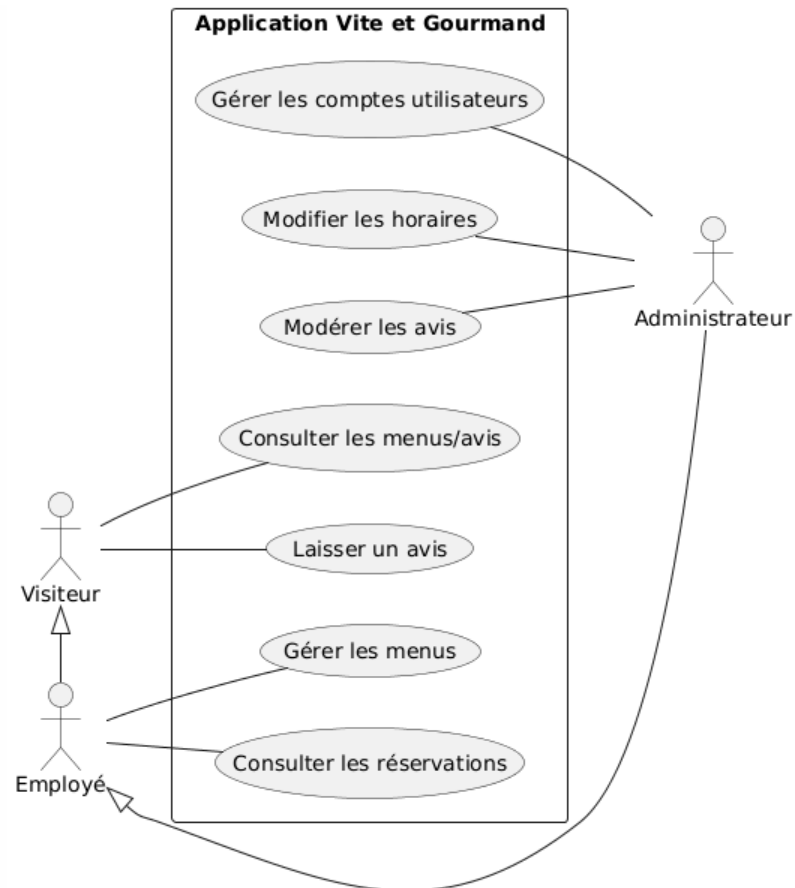
Architecture : J'ai choisi le parti pris de séparé le frontend du backend pour garder une interface dynamique et réactive. Le backend est sécurisé par Symfony en plus des regex et autres mécanismes implanté tout au long du code. Mais Symfony garanti les sécurité contre les XSS (Cross-Scripting), InjectionSQL, et attaque CSRF entre autre de part son ORM Doctrine qui facilité la gestion complexe de la base de données (menus, réservations, avis)

J'ai choisi une base de données non relationnelle MongoDB afi d'y stocjker les statistiques et gagner en réactivité lors de l'affichage dans le dashboard admin.

Environnement de développement :

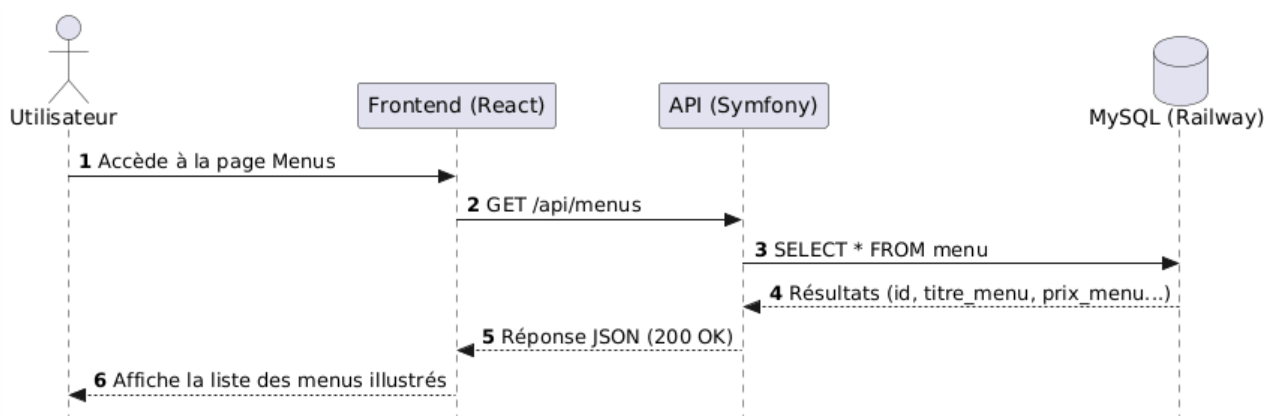
- **OS** : Windows 10/11 (ou Mac/Linux selon ton cas)
- **Serveur Local** : Symfony CLI / PHP 8.2

Diagrammes UML Cas d'Utilisation (Use Case)

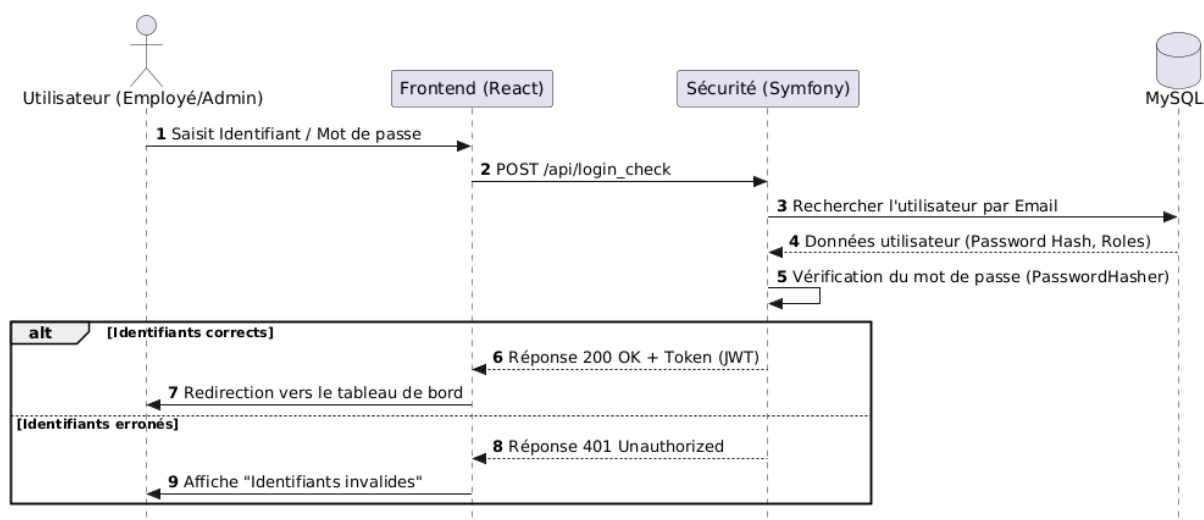


Ce diagramme définit les rôles au sein de l'application. On y distingue l'utilisateur public, dont les actions sont limitées à la consultation et à l'interaction de base, et l'employé possède les droits d'écriture et de modérations sur les données sensibles (menus,

horaires, avis). L'administrateur lui possède les droits de l'employé en plus de consulter les statistiques et de pouvoir créer un compte employé dont le mot de passe n'est communiqué que physiquement.



Ce schéma illustre le cycle de vie d'une requête HTTP. On observe que le Frontend React ne communique pas directement avec la base de données, mais sollicite une API REST développée sous Symfony. Cette architecture garantit une meilleure sécurité et une séparation claire des responsabilités. Les données transitent au format JSON.



Ce diagramme détaille le processus d'authentification sécurisée. L'utilisation du composant de Sécurité de Symfony interroge la base de données pour vérifier les droits. Le système utilise un hashage de Symfony avec l'algorithme 'auto' pour les de passe garantissant qu'aucune donnée sensible n'est stockée en clair.

Documentation du déploiement

Hébergement : J'ai utilisé Railway pour l'automatisation (CI/CD) via GitHub, ce qui permet une maintenabilité et une mise à jour relativement facile.

Configuration : Mise en place de variables d'environnement (DATABASE_URL) pour lier les conteneur PHP à la base de données MySQL.

Migration et synchronisation : La structure de la base de données a été ajustée pour assurer une parfaite cohérence entre les Entités Symfony et le schéma SQL.

Stockage des médias : Les images sont enregistré via des liens dans la base de données.

Gestion de projet

Git : j'ai créé une branche main qui sert de branche principal pour la production. Une branche develop a été créer pour le développement de nouvelles features. Chaque feature (fonctionnalités) possède sa propre branche qui une fois prête sont merger (fusionner) sur la develop afin d'être tester, si le test est concluant alors elle sera merger sur la main.

Trello : J'ai utilisé un trello afin de découper les US (User Story) du cahier des charges en sprint. Je me suis organisé de façon a accomplir de petits objectif car je suis en reconversion professionnelle et je travaille à côté. Je ne pouvais donc pas m'y consacré à temps plein, le mieux pour moi était donc de découper le projet en petite tâche afin de garder la motivation par petit accomplissement.