

15/10/2025

CAHIER DES CHARGES TECHNIQUE

SAÉ 5 - Développement avancé

Chayimaa MEHRAZ / Zaynab RIFI

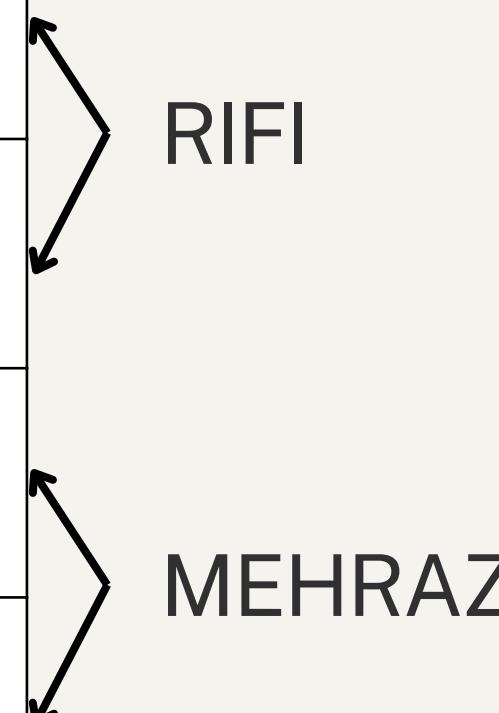
SOMMAIRE

- 1 - Organisation de l'équipe**
- 2 - Besoins client**
- 3 - Architecture générale**
- 4 - Technologies utilisées**
- 5 - Outils de développement**

- 6 - Hébergement / Déploiement**
- 7 - Environnement de test**
- 8 - Carte / Itinéraire / Calendrier**
- 9 - Sécurité**

1 - ORGANISATION DE L'ÉQUIPE

Méthodologie : Scrum (agile)	Répartition des rôles
Sprint de X semaines	Frontend (Blade, Tailwind, JS), UX/UI
Réunions régulières (daily / point d'avancement)	Backend Laravel (logique métier, tournées, calendrier)
Attribution des tâches via GitHub	Base de données (SQLite, migrations, modèles)
Revues de code (pull requests)	Carte / itinéraire (API), intégration et tests



RIFI

MEHRAZ

2- BESOINS CLIENT

Gérer les adhérents

L'inscription, la connexion et la gestion des abonnements.

Planifier les dates de livraison

Définir un calendrier de livraisons avec des fréquences et exceptions.

Offrir un tunnel de commande simple côté adhérent

L'adhérent choisit un abonnement et un point de dépôt puis valide.

Organiser les livraisons

Créer les tournées, choisir les dépôts et afficher l'itinéraire sur une carte.

Gérer les paniers et le stock

Répartir le stock de légumes dans différents types de paniers.

Interface claire et responsive

Application facile à utiliser sur ordinateur et mobile.

3 - ARCHITECTURE GÉNÉRALE

Architecture MVC (Laravel)

Frontend

Blade + Tailwind

Backend

Laravel (Contrôleurs, Modèles, Routes)

Base de données

SQLite

Environnement

Docker

Utilisateur → Front → Backend → DB

4 - TECHNOLOGIES UTILISÉES



Laravel

Framework PHP MVC pour structurer, sécuriser et faciliter le développement.



SQLite

Base de données légère, fichier unique, simple à utiliser.



JavaScript

Gestion des interactions dynamiques (carte, itinéraire, calendrier...).



Bootstrap

Bibliothèque de composants prêts à l'emploi pour une interface responsive.



Tailwind CSS

Framework utilitaire pour le design sans écrire de CSS manuel.

5- OUTILS DE DÉVELOPPEMENT



Docker

Conteneurs pour exécuter l'app dans un environnement identique.



VS Code

Éditeur de code léger et personnalisable avec extensions (Laravel, Tailwind, Git...)



Github

Plateforme en ligne pour héberger le projet, collaborer et suivre l'avancement.

6 - HÉBERGEMENT/DÉPLOIEMENT

1



Serveur web : Apache

- Compatible avec Laravel (PHP)
- Gestion des routes et du .htaccess

2



Conteneurisation : Docker

- Environnement reproductible
- Configuration identique dev / prod

3



Base de données : SQLite

- Stockée sur le serveur
- Sauvegardes régulières

7- ENVIRONNEMENT DE TEST

- | | |
|---|--|
| 1 | Serveur local Apache (via Laravel / Docker) |
| 2 | Base de données SQLite dédiée aux tests |
| 3 | Tests automatisés avec PHPUnit (intégré à Laravel) |
| 4 | Données fictives via seeders / factories |
| 5 | Tests manuels via Postman / navigateur |

8 - CARTE / ITINÉRAIRE / CALENDRIER



Carte

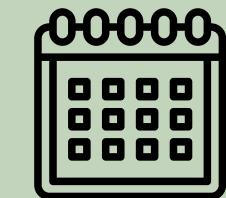
API utilisée : (Google Maps ou Leaflet + OpenStreetMap)

Affichage des points de retrait
Interaction : clic pour infos
Intégration via JavaScript dans
Blade



Itinéraire

API de calcul d'itinéraire (ex :
Directions API, Leaflet Routing
Machine)
Position utilisateur → point de retrait
Affichage graphique sur la carte
Mise à jour dynamique



Calendrier

Outil : FullCalendar (JS) ou calendrier
personnalisé

Affichage des dates de livraison
Interaction (détails d'une date)
Synchronisation avec la base de
données

9 - SÉCURITÉ

Authentification & rôles

→ Accès limité selon le type d'utilisateur (admin / adhérent).

Validation des données

→ Contrôle des champs côté backend et frontend.

Protection des formulaires

→ CSRF, mots de passe hashés.

Sauvegarde des données (SQLite)

→ Backups réguliers pour éviter la perte d'informations.

