

Cahier des charges simplifié

1. Présentation du projet

- **Nom du projet :**
OrphanCare – Gestion des orphelins et des dons pour une association caritative
- **Brève description :**
Application web destinée à gérer les données des orphelins, les dons reçus, les remarques de suivi, et les notifications internes, au sein d'une association caritative.
- **Objectif principal de l'application :**
Permettre aux responsables de l'association de centraliser, suivre et gérer les informations des orphelins et les dons de manière structurée, sécurisée et accessible.

2. Problèmes à résoudre / besoins exprimés

- **Problèmes identifiés :**
 - Difficulté à stocker et retrouver les informations des orphelins.
 - Absence de suivi clair des dons et des remarques personnalisées.
 - Manque de communication ou de rappels entre les membres de l'équipe.
- **Besoins exprimés :**
 - Centralisation des données dans une plateforme unique.
 - Suivi historique des dons (avec lien vers le responsable et l'orphelin).
 - Ajout et consultation de remarques régulières (médicales, scolaires...).
 - Notifications automatiques par email pour les mises à jour importantes.

3. Utilisateurs cibles

Type d'utilisateur	Description
Administrateur (Directeur)	Gère l'ensemble de l'application, les utilisateurs, les données et les notifications.
Employé (Responsable social)	S'occupe des fiches des orphelins, enregistre les dons, ajoute des remarques.

4. Fonctionnalités attendues

Pour l'administrateur :

- Gérer les utilisateurs et leurs rôles (ajout, modification, suppression).
- Gérer les orphelins (ajouter, modifier, supprimer).
- Suivre les dons, filtrer par date, orphelin ou employé.
- Recevoir des notifications importantes.

Pour l'employé :

- Ajouter/modifier les informations d'un orphelin.
- Enregistrer les dons reçus.
- Ajouter des remarques sur chaque orphelin (état de santé, résultats scolaires...).
- Rechercher et filtrer les orphelins selon plusieurs critères.

5. Contraintes techniques

- **Technologies imposées :**
 - Backend : **Laravel (PHP)**
 - Frontend : **Blade** (ou React en évolution)
- **Base de données prévue :**
 - **MySQL**
- **Hébergement envisagé :**
 - Serveur mutualisé ou VPS (OVH, Hostinger, Laravel Forge...)
- **Compatibilité / Responsiveness :**
 - Interface responsive adaptée aux **ordinateurs, tablettes et smartphones**.



1. Liste des utilisateurs cibles

Nom du type d'utilisateur	Qui sont-ils dans la vraie vie ?
Administrateur	Directeur de l'association, responsable principal Travailleur social, agent chargé de la gestion terrain
Employé	

2. Rôles et permissions

Type d'utilisateur	Actions autorisées
--------------------	--------------------

	- Gérer les comptes utilisateurs (ajout/suppression)
Administrateur	- Ajouter, modifier, supprimer un orphelin
	- Gérer les dons
	- Voir tous les rapports et remarques
	- Gérer les notifications
	- Ajouter/modifier les fiches des orphelins
Employé	- Ajouter des remarques (suivi médical, scolaire...)
	- Enregistrer des dons
	- Consulter les données
	- Ne peut pas gérer les utilisateurs

2. Rôles et permissions

Type d'utilisateur	Fonctions autorisées
Administrateur	<ul style="list-style-type: none"> • Gérer les utilisateurs (ajout, modification, suppression) • Attribuer des rôles • Gérer les orphelins (CRUD) • Gérer les dons (CRUD) • Accéder au tableau de bord/statistiques • Voir toutes les notes • Recevoir toutes les notifications
 Employé / Bénévole 	<ul style="list-style-type: none"> • Ajouter/modifier les informations d'un orphelin • Ajouter des notes de suivi (éducation, santé...) • Rechercher et filtrer les orphelins • Ajouter un don et voir les dons liés • Recevoir des notifications liées à leurs actions

☒ 3. Lien avec les technologies

Technologie	Implémentation prévue
Laravel	<ul style="list-style-type: none"> • Middleware auth pour l'accès sécurisé • Middleware personnalisé <code>isAdmin</code>, <code>isEmployee</code>

- Utilisation de **Gates** ou **Policies** pour contrôler l'accès à certaines ressources (par exemple : un employé ne peut pas supprimer un utilisateur)
| | **React (si interface frontend)** |
- Affichage conditionnel des pages ou composants selon le rôle
- Routes protégées (avec PrivateRoute)
- Utilisation d'un **contexte global d'authentification** pour stocker le rôle et l'état de connexion

✓ 4. Livrable final attendu

Type d'utilisateur	Description	Permissions principales
Administrateur	Responsable système de l'association	<ul style="list-style-type: none"> - Gérer les utilisateurs - Gérer les orphelins - Gérer les dons - Voir le tableau de bord - Accéder à toutes les notes et notifications - Ajouter/modifier des orphelins
Employé / Bénévole	Agent de suivi ou assistant dans l'association	<ul style="list-style-type: none"> - Ajouter des dons - Ajouter des notes de suivi - Rechercher/filtrer - Notifications liées à leurs actions

1. Présentation générale

But :

Développer une application web pour permettre à une association caritative de gérer efficacement les informations des orphelins, les dons reçus, ainsi que la communication interne entre les membres du personnel.

Utilisateurs cibles :

- Administrateurs de l'association
- Employés ou bénévoles
- (Possiblement) les donateurs à l'avenir

2. Liste des fonctionnalités par type d'utilisateur

Administrateur

- Créer, modifier et supprimer des comptes utilisateurs (employés)
- Gérer les rôles et les permissions (Admin / Employé)
- Gérer les informations des orphelins (ajouter, modifier, supprimer)

- Consulter les statistiques globales (nombre d'orphelins, dons, etc.)
- Recevoir des notifications sur les mises à jour importantes











Employé / Bénévole

- Ajouter un orphelin avec ses informations personnelles et médicales
- Modifier les informations d'un orphelin
- Ajouter des notes de suivi (éducation, santé, comportement...)
- Rechercher et filtrer la liste des orphelins
- Enregistrer les dons reçus, avec détail du donateur
- Consulter l'historique des dons d'un orphelin
- Recevoir des notifications importantes par email

3. User Stories (optionnel)

- En tant qu'**employé**, je veux **ajouter un nouvel orphelin**, afin de **mettre à jour la base de données de l'association**.
- En tant qu'**administrateur**, je veux **gérer les comptes utilisateurs**, afin de **contrôler l'accès aux données sensibles**.
- En tant qu'**employé**, je veux **ajouter une note de suivi pour un orphelin**, afin de **documenter son évolution**.
- En tant qu'**administrateur**, je veux **recevoir une notification quand un nouveau don est enregistré**, afin de **suivre les activités de l'association en temps réel**.

4. Priorisation des fonctionnalités

Fonctionnalité	Priorité
Authentification et gestion des rôles (Admin / Employé)	 Essentielle
CRUD complet sur les orphelins	 Essentielle
Ajout et suivi des dons	 Essentielle
Recherche et filtres sur la liste des orphelins	 Essentielle
Ajout de notes de suivi	 Essentielle
Notifications email aux administrateurs	 Secondaire
Tableau de bord avec statistiques	 Secondaire
Historique détaillé des dons par orphelin	 Secondaire
Gestion des comptes utilisateurs (création, suppression, rôles)	 Essentielle
Accès pour les donateurs (interface publique)	 Bonus

✓ 1. Liste des tâches à réaliser (par phase)

🧠 Phase 1 – Analyse & Conception (8 h)

- Étudier le cahier des charges et les besoins utilisateurs
- Créer le modèle de base de données (MCD/MLD)
- Définir les rôles et permissions (Admin, Employé)
- Concevoir les wireframes/pages principales (Accueil, Orphelins, Dons, etc.)

🔗 Phase 2 – Développement Back-end Laravel (20 h)

- Créer le projet Laravel et configurer l'environnement
- Authentification + rôles (Admin / Employé)
- CRUD Orphelins (modèle, contrôleur, migration, vue)
- CRUD Dons (modèle, relation avec orphelins)
- Module de notes de suivi (santé, scolaire)
- Envoi des notifications email (Mailgun ou SMTP)
- Middleware / Policy pour sécuriser les accès

🎨 Phase 3 – Développement Front-end (Blade) (10 h)

- Intégration Blade des vues principales (liste orphelins, formulaire dons, etc.)
- Layout et navigation (menu, tableaux, formulaire)
- Responsive design (Bootstrap ou Tailwind)

Phase 5 – Déploiement (4 h)

Tester en ligne (performance, sécurité, accessibilité)

🕒 2. Estimation de la durée (60 h total)

Phase	Durée estimée
Analyse & Conception	8 h
Développement Back-end	20 h
Développement Front-end	10 h
Tests & validation	8 h
Déploiement	4 h
Documentation & livrables	2 h
Total	52 h

Marge de sécurité / imprévus	8 h
------------------------------	-----

3. Estimation des coûts

Élément	Coût estimé
Nom de domaine (optionnel)	100–150 dh /an
Outils de design (ex. Figma)	Gratuit pour usage individuel
GitHub	Gratuit
Total estimatif	~200dh

4. Outils nécessaires au développement

Catégorie	Outils
Langage / Framework	Laravel (PHP), Blade
Base de données	MySQL (phpMyAdmin / Laravel Migration)
IDE	VS Code
Contrôle de version	Git, GitHub
Design	Figma (wireframes), Whimsical
Test	Postman (API), navigateur