



Cahier des Charges

Technique

Mélanie TROUTIER & Mathis MOLLIER & Théo VALADOUX & Yulian GUINAND
23/09/2025

1. Introduction et périmètre technique.....	3
Rappel du contexte.....	3
Positionnement du document.....	3
Public visé.....	3
2. Architecture technique globale.....	4
3. Environnement technologique.....	4
4. Modèle de données.....	5
1. Table : DAYS.....	5
2. Table : INTERVENTIONS.....	6
3. Table : MEMBERS.....	7
4. Table : ASSOCIATIONS.....	8
5. Synthèse des relations entre les tables.....	9
6. MCD:.....	10
7. MPD :.....	10
8. Diagramme de classes.....	11
9. Diagramme ADO.....	12
5. Interfaces techniques.....	12
6. Contraintes techniques.....	12
7. Sécurité.....	14
8. Interface utilisateur (IHM).....	14
9. Communication.....	20
10. Gestion du code et méthodologie.....	20
11. Planning.....	21
12. Signatures.....	22

1. Introduction et périmètre technique

Rappel du contexte

Chaque année le lycée Pasteur Mont Roland organise une journée Santé Citoyenneté et invite des associations à y participer. Mr Pernelle, organisateur de cette journée, nous a demandé de créer une application lui permettant de centraliser, automatiser et sécuriser le processus d'invitation, de réponse et de suivi. L'objectif est de réduire sa charge administrative tout en améliorant la coordination avec les associations partenaires.

Actuellement, la gestion de ces interventions repose principalement sur des échanges de courriels et des documents partagés, ce qui engendre une charge administrative importante pour le professeur principal en charge de la coordination. Cette méthode manuelle présente plusieurs limites : risque d'erreurs, perte d'informations, suivi complexe et absence de centralisation des données.

Positionnement du document

Ce cahier des charges techniques traduit les besoins fonctionnels en spécifications techniques concrètes. Il définit l'architecture logicielle, les technologies utilisées, la gestion des données, la sécurité et les interfaces à développer.

Sont inclus dans ce document : le backend, frontend, la base de données, les API, la sécurité et l'hébergement.

Ne sont pas couverts : la formation des utilisateurs finaux, la rédaction des manuels pédagogiques, ni la gestion du matériel réseau de l'établissement.

Public visé

Ce document s'adresse principalement :

- à l'équipe de développement (conception, codage, intégration),
- aux administrateurs systèmes (installation, déploiement, maintenance),
- aux responsables techniques de l'établissement (validation de la conformité et de la sécurité).

2. Architecture technique globale

L'architecture de l'application repose sur une structure **client-serveur** moderne, permettant une séparation claire entre l'interface utilisateur, la logique métier et la gestion des données. Le **front-end**, développé en React et stylisé avec Tailwind CSS, constitue l'interface principale pour les associations et le professeur responsable. Il communique directement avec le **back-end**, une API développée en Laravel, qui centralise le traitement des données et coordonne les différentes fonctionnalités de l'application.

Le **back-end** interagit avec la **base de données SQL Server**, hébergée sur un serveur Windows sécurisé, afin de stocker, récupérer et mettre à jour toutes les informations liées aux associations, aux interventions et aux réponses aux formulaires. Chaque composant est isolé mais interconnecté via des **protocoles sécurisés** (HTTPS) garantissant la confidentialité et l'intégrité des échanges. Les données transmises entre le front-end et l'API sont formatées en **JSON**, assurant une communication standardisée et facilement exploitable.

Cette architecture assure une modularité optimale, une maintenance simplifiée et une scalabilité future, tout en garantissant un accès rapide et sécurisé aux informations pour tous les utilisateurs autorisés.

3. Environnement technologique

Le backend principal sera réalisé en **C#**, tandis que l'API sera développée en **Laravel**. Le frontend sera conçu avec **React**, associé à **Tailwind CSS** pour la mise en forme et à **Vite** pour optimiser le développement et le déploiement. Les données seront centralisées dans une base **SQL Server**. L'ensemble sera hébergé sur un **serveur Windows**. Enfin, le projet s'appuiera sur divers outils tiers et dépendances, notamment des bibliothèques et packages **npm** pour le frontend et **nuget** pour le backend, afin de simplifier le développement et garantir la maintenabilité du code.

4. Modèle de données

Le modèle de données repose sur quatre tables principales — **DAYS, INTERVENTIONS, MEMBERS et ASSOCIATIONS** — ainsi que sur plusieurs relations d'association entre elles.

1. Table : DAYS

La table **DAYS** regroupe l'ensemble des journées organisées par l'association ou ses partenaires.

Chaque enregistrement correspond à une journée spécifique, identifiée par une clé primaire unique.

- **days_id (INT, clé primaire)** : identifiant unique de la journée.
- **date_day (DATE)** : date précise de la journée (par exemple « 2025-06-14 »).
- **title (VARCHAR)** : titre ou nom donné à la journée (ex. « Journée découverte », « Fête de l'environnement »).

Cette table est centrale dans le modèle : elle est reliée aux interventions qui ont lieu lors de cette journée (**relation HOSTING**) et aux associations qui y participent (**relation AGREE**).

Ainsi, une journée peut accueillir plusieurs interventions et impliquer plusieurs associations partenaires.

2. Table : INTERVENTIONS

La table **INTERVENTIONS** recense toutes les interventions, ateliers ou activités prévues pendant les différentes journées.

Chaque intervention est liée à **une journée** (via la clé étrangère `days_id`) et organisée par une association (via la clé étrangère `asso_id`).

- **inte_id (INT, clé primaire)** : identifiant unique de l'intervention.
- **title (VARCHAR)** : titre de l'intervention (ex. « Atelier cuisine », « Initiation au théâtre »).
- **activity (VARCHAR)** : description ou type de l'activité.
- **needs (VARCHAR)** : liste ou description du matériel nécessaire (ex. « tables, chaises, four, ingrédients »).
- **disposition (VARCHAR)** : précisions sur l'organisation ou la mise en place de l'activité (ex. « en extérieur, par petits groupes »).
- **cost (VARCHAR)** : coût estimé ou réel de l'intervention (peut être un montant ou une mention telle que « gratuit »).
- **nb_meals (INT)** : nombre de repas prévus pour cette intervention (le cas échéant).
- **rest (BOOLEAN)** : indique si une pause est prévue pendant l'intervention.
- **when_rest (VARCHAR)** : moment de la pause, si applicable (ex. « 14h00 »).
- **other (VARCHAR)** : informations complémentaires (remarques, contraintes, etc.).
- **days_id (INT, clé étrangère)** : référence à la journée pendant laquelle l'intervention est organisée.
- **asso_id (INT, clé étrangère)** : référence à l'association qui propose ou encadre l'intervention.

Les relations principales sont donc :

- **HOSTING (DAYS → INTERVENTIONS)** : une journée peut accueillir plusieurs interventions.
- **GOING (ASSOCIATIONS → INTERVENTIONS)** : une association peut proposer plusieurs interventions.
- **PARTICIPATE(INTERVENTIONS → MEMBERS)** : une intervention peut accueillir plusieurs membres

3. Table : MEMBERS

La table **MEMBERS** contient les informations sur les personnes impliquées dans les associations partenaires.

Chaque membre appartient à une seule association, mais une association peut avoir plusieurs membres.

- **memb_id (INT, clé primaire)** : identifiant unique du membre.
- **last_name (VARCHAR)** : nom de famille du membre.
- **first_name (VARCHAR)** : prénom du membre.
- **fonction (VARCHAR)** : rôle ou fonction au sein de l'association (ex. « animateur », « bénévole », « président »).
- **asso_id (INT, clé étrangère)** : référence à l'association à laquelle appartient le membre.

Relation principale :

- **BELONG (MEMBERS → ASSOCIATIONS)** : un membre appartient à une seule association.
- **PARTICIPATE(MEMBERS → INTERVENTIONS)** : un membre peut participer à plusieurs interventions

Cette table permet d'identifier les ressources humaines disponibles, de suivre les rôles des intervenants et de faciliter la coordination des journées.

4. Table : ASSOCIATIONS

La table **ASSOCIATIONS** répertorie l'ensemble des associations partenaires ou organisatrices des journées.

Chaque association peut participer à plusieurs journées et proposer plusieurs interventions.

- **asso_id (INT, clé primaire)** : identifiant unique de l'association.
- **nom (VARCHAR)** : nom de l'association.
- **mail (VARCHAR)** : adresse e-mail de contact.
- **telephone (VARCHAR)** : numéro de téléphone de l'association.
- **url_photo (VARCHAR)** : lien vers une photo ou un logo représentant l'association.

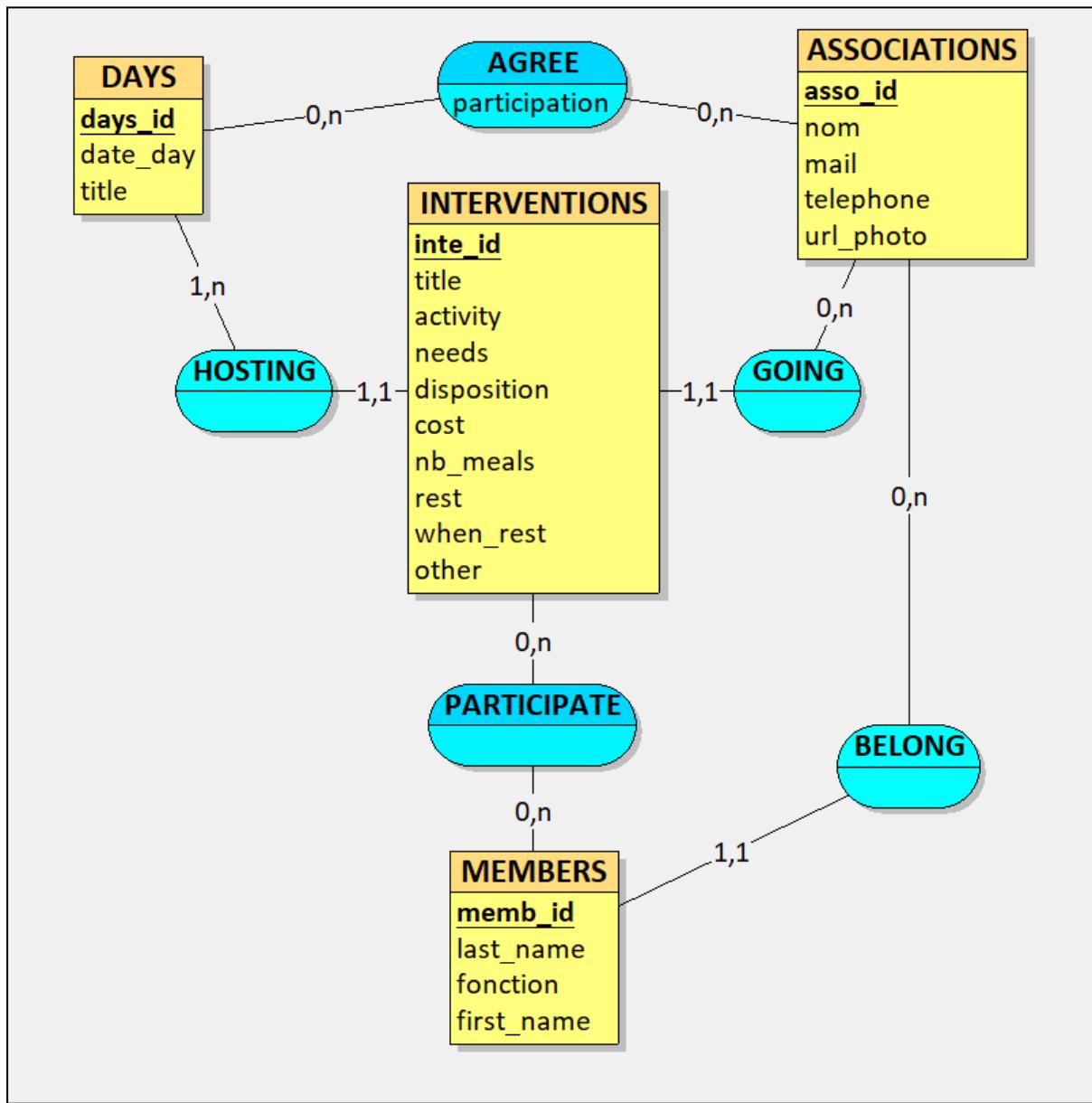
Relations principales :

- **AGREE (ASSOCIATIONS ↔ DAYS)** : une association peut participer à plusieurs journées, et une journée peut accueillir plusieurs associations.
- **GOING (ASSOCIATIONS → INTERVENTIONS)** : une association peut proposer plusieurs interventions.
- **BELONG (ASSOCIATIONS ↔ MEMBERS)** : une association regroupe plusieurs membres.

5. Synthèse des relations entre les tables

Relation	Description	Cardinalité
AGREE	Relie DAYS et ASSOCIATIONS : une journée peut accueillir plusieurs associations et une association peut participer à plusieurs journées.	N:N
HOSTING	Relie DAYS et INTERVENTIONS : une journée peut accueillir plusieurs interventions, mais une intervention appartient à une seule journée.	1:N
GOING	Relie ASSOCIATIONS et INTERVENTIONS : une association peut proposer plusieurs interventions, mais une intervention est organisée par une seule association.	1:N
BELONG	Relie MEMBERS et ASSOCIATIONS : un membre appartient à une seule association, mais une association peut avoir plusieurs membres.	1:N
PARTICIPATE	Relie INTERVENTIONS et MEMBERS : un membre peut participer à plusieurs interventions et une intervention peut accueillir plusieurs membres	N:N

6. MCD:

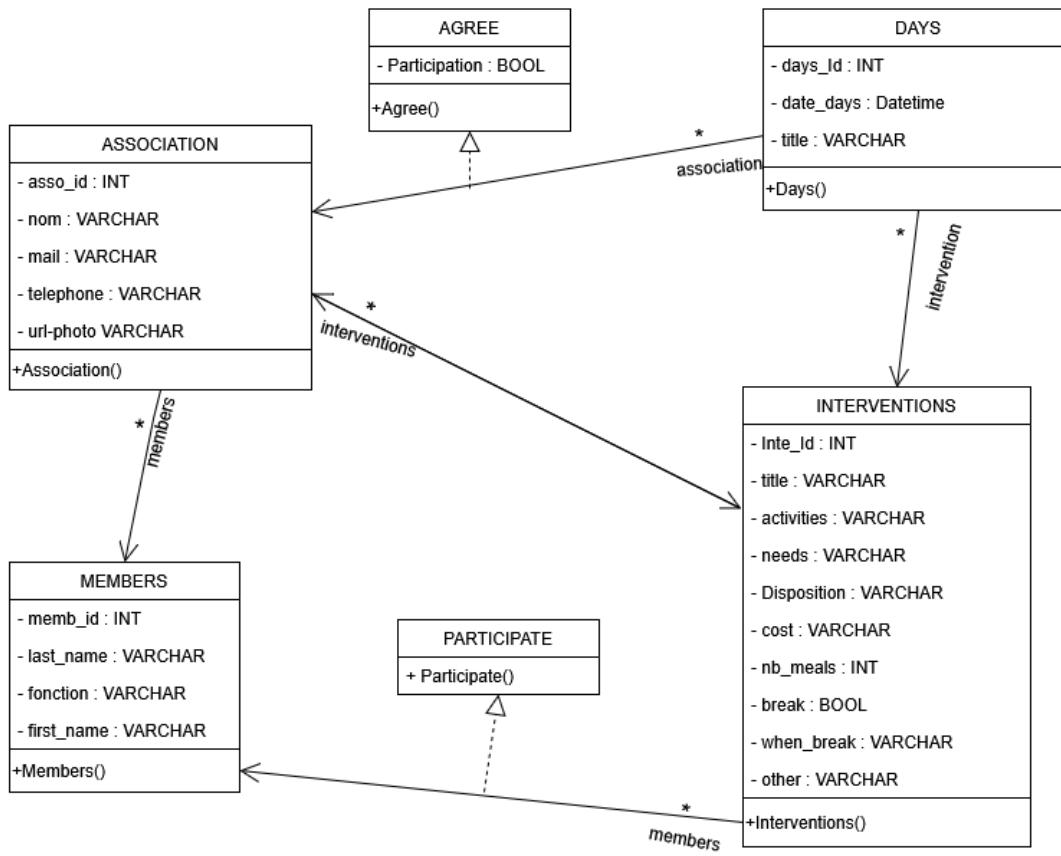


7. MPD :

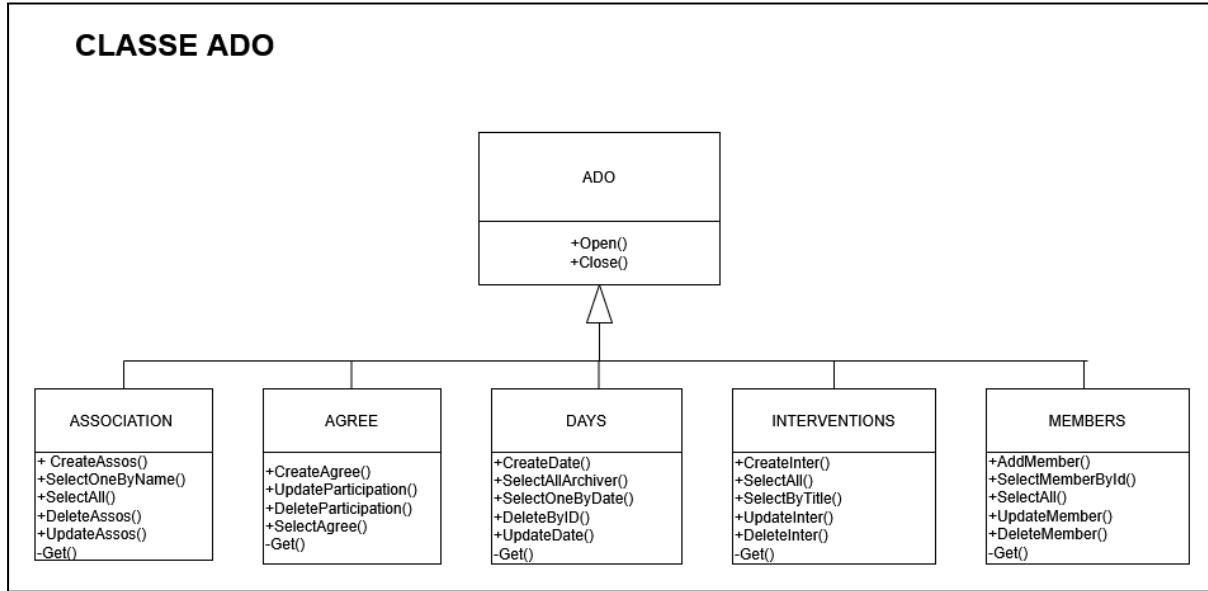
DAY(`days_id`, `date_day`, `title`)
ASSOCIATIONS(`asso_id`, `nom`, `mail`, `telephone`, `url_photo`)
INTERVENTIONS(`inte_id`, `title`, `activity`, `needs`, `disposition`, `cost`, `nb_meals`, `rest`, `when_rest`, `other`, #`days_id`, #`asso_id`)
MEMBERS(`memb_id`, `last_name`, `first_name`, `fonction`, #`asso_id`)
AGREE(`days_id`, `asso_id`, `participation`)
PARTICIPATE(`inte_id`, `memb_id`)

8. Diagramme de classes

Diagramme de classe



9. Diagramme ADO



5. Interfaces techniques

L'API constituera un élément central du projet et comprendra **quatre points d'accès principaux** :

- un premier **GET** pour la récupération des invitations,
- un **PUT** permettant d'apporter une réponse à ces invitations,
- un second **GET** dédié à l'affichage des formulaires,
- et enfin un **POST** destiné à l'enregistrement des réponses transmises.

Ces endpoints devront être définis avec précision et documentés de manière exhaustive afin de garantir un usage simple, cohérent et conforme aux standards de développement

L'API retourne les données sous format JSON et sont ensuite renvoyées.

6. Contraintes techniques

L'application devra être compatible avec les principaux navigateurs modernes, en particulier Google Chrome, Mozilla Firefox et Microsoft Edge, dans leurs versions récentes, afin de garantir une accessibilité optimale pour tous les utilisateurs. Les performances devront être suffisantes pour assurer un temps de réponse inférieur à deux secondes lors du chargement des pages ou de la soumission des formulaires, même en cas de forte sollicitation. Le système devra également être capable de supporter simultanément la connexion et la participation d'une cinquantaine d'associations sans dégradation notable des performances. Enfin, l'hébergement s'appuiera sur un serveur Windows Server disposant d'une configuration minimale de 4 Go de RAM, 4 coeurs CPU et 50 Go d'espace disque, garantissant ainsi la stabilité et la fiabilité de la plateforme.

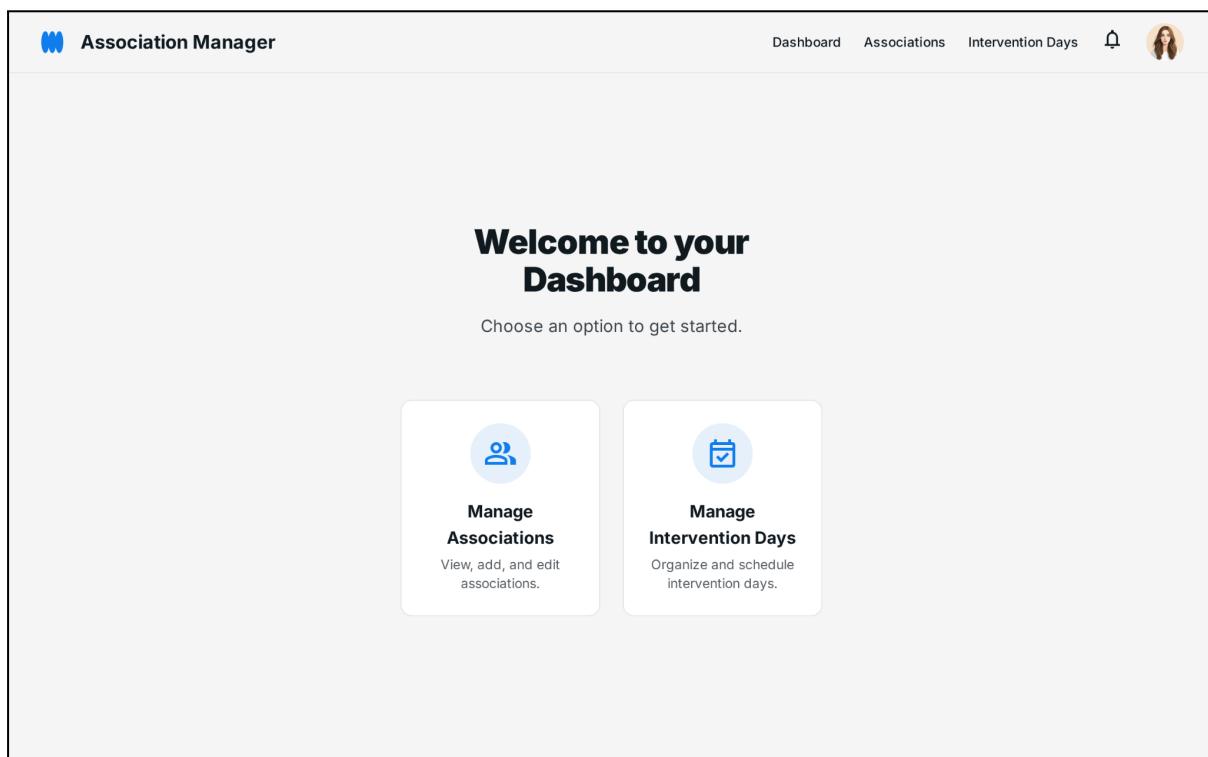
7. Sécurité

L'administrateur se connectera via un bureau à distance sur l'application. Cela lui permettra d'avoir un accès sécurisé et de garantir la confidentialité et l'intégrité des données sensibles. Les autres utilisateurs pourront se rendre sur le formulaire via un lien. Toutes les images seront sauvegardées en base de données pour permettre une meilleure sécurisation des données. En ce qui concerne les données importantes, ils seront chiffrées pour en garantir la sécurité.

8. Interface utilisateur (IHM)

Certaines parties de la maquette sont des placeholders. Sur la maquette, la langue est en anglais mais l'outil sera en français. Il ne faut donc pas prendre en compte la langue et les placeholders.

Page d'atterrissement pour l'outil d'admin :



La page d'atterrissement pour les journées : (Attention la barre de navigation sera comme sur la première image.

CommunityConnect

- Dashboard
- Intervention Days**
- Associations
- Volunteers
- Reports

Intervention Days

+ New Day

Upcoming Day

Street Cleanup Initiative
August 5, 2024 - Downtown Plaza

Planning

Invited Associations

+5

[View Responses](#) [Send Form](#)

Archived Days

DATE	LOCATION	ASSOCIATIONS	VOLUNTEERS	STATUS	ACTION
2024-07-20	City Park	Helping Hands, Community Builders	25	Completed	View Results
2024-06-15	Community Center	Youth Empowerment, Green Team	18	Completed	View Results
2024-05-05	Local Library	Elder Care, Literacy Project	22	Completed	View Results

La page pour voir les interventions en détail (les réponses des formulaires) :

The screenshot displays the 'Intervention Day Details' page from the 'Association App'. At the top, there's a navigation bar with the app logo, 'Association App', and links for 'Dashboard', 'Interventions', 'Associations', and 'Members'. A user profile icon is also present.

Intervention Day Details

Review the details of the intervention day, including participating associations and their submitted forms.

Date
July 15, 2024

Participating Associations

Community Helpers

View Members **View Form**

Form Results

Activity Details Community cleanup and park beautification	Room Needs Large hall with projector
Break Presence Morning and Afternoon	Room Layout Theater style
Cost \$250	People Staying for Meals 30
Additional Details Volunteers will bring their own tools.	

Youth Empowerment Group

View Members **View Form**

Form Results

Activity Details Youth leadership workshop	Room Needs Medium room with tables and chairs
Break Presence Afternoon only	Room Layout Classroom style
Cost \$150	People Staying for Meals 20
Additional Details Workshop materials provided.	



Edit Intervention

Date

08/15/2024



Intervention Details

Title

Workshop on digital tools

Description of the activity

A hands-on workshop to familiarize members with new digital collaboration tools.

Needs

Projector, whiteboard, markers, and a good internet connection.

Presence of a break

Is there a break during this intervention?



Room Layout

Classroom

Cost (€)

150

Number of people eating

12

Save Changes

Pour la page des associations :

 CommunityConnect

Dashboard **Associations** Interventions Reports Settings 

Associations Management

Manage your associations, their members, and related interventions.

Add New Association

Existing Associations			
ASSOCIATION NAME	DESCRIPTION	MEMBERS	ACTIONS
Greenwood Community Group	Focuses on environmental sustainability and community gardening.	25	 
Oakwood Seniors Club	Provides social activities and support for senior citizens.	40	 
Riverdale Youth Initiative	Empowers young people through educational and recreational progr...	30	 

Showing 1 to 3 of 3 results

 Previous  Next

Pour ajouter une association :

The screenshot shows the 'CommunityConnect' application interface. At the top, there is a navigation bar with links for Dashboard, Associations (which is the active tab), Interventions, Members, and Reports. A user profile icon is also present. Below the navigation, the title 'Add New Association' is displayed, followed by a subtitle 'Enter the association's details and manage its members.' The main form is divided into two sections: 'Association Details' and 'Manage Members'. In the 'Association Details' section, there are fields for 'Association Name' (with placeholder 'Enter association name') and 'Email' (with placeholder 'Enter email'). Below these, there is a field for 'Phone Number' (placeholder 'Enter phone number'). In the 'Manage Members' section, there is a table with columns 'FIRST NAME', 'LAST NAME', and 'ROLE'. Three rows of data are shown: Liam Harper (President), Olivia Bennett (Secretary), and Noah Carter (Treasurer). Each row has 'Edit' and 'Delete' buttons. At the bottom of the form, there are 'Cancel' and 'Save Association' buttons.

Une **Interface Homme-Machine (IHM)** sera mise à disposition des associations. Celle-ci leur offrira la possibilité de consulter directement les invitations reçues et d'y répondre via un formulaire en ligne. La technologie choisie pour le développement de cette interface reste laissée à l'appréciation de l'équipe projet, à condition que celle-ci respecte strictement les principes d'ergonomie et de simplicité d'utilisation.

L'application sera responsive ce qui permettra de l'utiliser à la fois sur des écrans d'ordinateur, mais également sur des écrans plus petits comme une tablette ou un smartphone.

Le module frontal, destiné aux associations, sera développé avec **React** pour la logique d'interface, associé à **Tailwind CSS** pour la mise en forme et à **Vite** pour l'optimisation du développement et du déploiement

9. Communication

L'application intègre un système automatisé d'envoi de courriers électroniques afin de faciliter la communication avec les associations. Les e-mails seront envoyés par l'intermédiaire d'un service tiers spécialisé (type Mailtrap) afin d'assurer fiabilité et délivrabilité. Chaque message comprendra un lien généré par l'application, permettant à l'association de répondre directement via le formulaire en ligne.

10. Gestion du code et méthodologie

Le développement s'appuiera sur une **méthodologie GitFlow** afin de garantir une organisation claire et collaborative du code source. Chaque fonctionnalité sera développée dans une **branche dédiée** (*feature branch*) avant d'être intégrée, après revue de code, dans la branche principale (*main*). Cette approche permet d'assurer une traçabilité optimale, une intégration continue maîtrisée et une meilleure qualité du produit final.

Chaque développeur est libre d'utiliser l'IDE de son choix (Visual Studio, Visual Studio Code, etc.). L'ensemble du code source sera versionné et partagé via GitHub afin de centraliser la gestion collaborative du projet

11. Planning

Séance	Durée
18/09	4 heures
23/09	4 heures
14/10	4 heures
04/11	4 heures
25/11	4 heures
02/12	4 heures
05/01	2 heures
06/01	4 heures
12/01	2 heures
13/01	4 heures

Le projet doit être entièrement finalisé pour le **13 janvier**, après un volume estimé à **136 heures** de travail. À cette échéance, une soutenance officielle sera organisée devant l'équipe enseignante afin de présenter l'application, d'en démontrer le fonctionnement, de mettre en avant ses principales fonctionnalités et d'illustrer son apport pour la gestion des interventions associatives au sein du lycée.

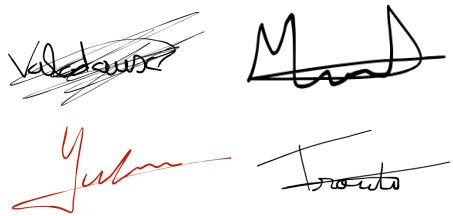
12. Signatures

Signature du client

Date :

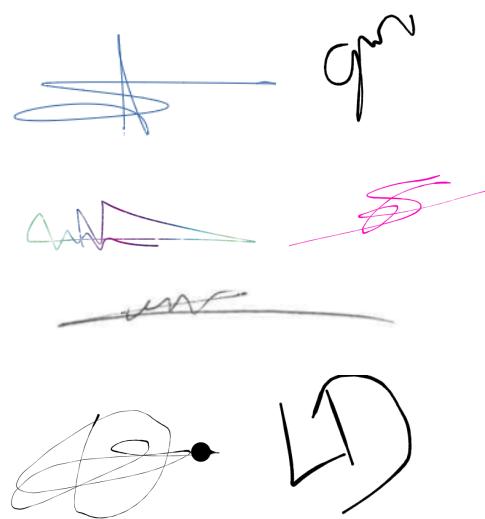
Signature de l'équipe de développement et des administrateurs réseaux

Date: 16/10/2025



Handwritten signatures of development team members, including a black signature that appears to be "Valdovin", a blue signature that appears to be "M", a red signature that appears to be "Yulian", and a black signature that appears to be "Trots".

Date: 16/10/2025



Handwritten signatures of network administrators, including a blue signature that appears to be "g", a green signature that appears to be "S", a black signature that appears to be "V", and a black signature that appears to be "D".