

# Rapport de Gestion de Projet pour le Système de Parrainage Électoral

## Table des matières

Introduction.....	3
1. Organisation de l'équipe .....	4
2. Planning de réalisation .....	5
3. Stratégie de développement .....	6
4. Architecture technique .....	7
5. Modèles conceptuels et données .....	8
6. Fonctionnalités du système .....	15
Conclusion .....	20

## **Introduction**

Dans le cadre de la numérisation des procédures administratives liées à la gestion des élections au Sénégal, le projet **Gestion des Parrainages de Candidature d'Élections Présidentielles** a été lancé pour répondre à des besoins essentiels en matière d'efficacité et de transparence. Le processus actuel de parrainage repose sur des méthodes manuelles, impliquant le remplissage de feuilles papier avec des informations personnelles des électeurs, ce qui rend le système à la fois laborieux et sujet à des erreurs. En digitalisant ce processus, nous visons à réduire les délais de traitement, à minimiser les erreurs humaines et à améliorer l'accessibilité pour tous les citoyens.

Ce projet a pour ambition de créer une application web conviviale qui permettra aux électeurs de parrainer les candidats de manière simple et sécurisée. Les différentes phases du projet incluront le chargement de la liste des électeurs, la saisie des informations des candidats, l'ouverture de la période de parrainage, ainsi que le suivi des parrainages. Chaque étape sera soigneusement conçue pour garantir la conformité avec les réglementations en vigueur tout en offrant une expérience utilisateur fluide.

L'objectif principal de ce rapport est de présenter de manière détaillée les différentes étapes de la mise en œuvre de ce projet, ainsi que les méthodes et outils utilisés pour sa réalisation. Nous examinerons l'organisation de l'équipe impliquée dans le développement, le planning de réalisation, la stratégie de développement adoptée, l'architecture technique mise en place, les modèles UML utilisés pour modéliser le système, ainsi que les fonctionnalités clés de l'application développée.

## **1. Organisation de l'équipe**

Le projet **Gestion des Parrainages de Candidature d'Élections Présidentielles** est réalisé par une équipe de quatre membres, chacun ayant un rôle spécifique :

- **Taiba SAKHO :**
  - **Rôle :** Conception du système.
  - **Responsabilités :** Élaboration de l'architecture, de l'ergonomie et des prototypes.
- **Yacine FALL :**
  - **Rôle :** Développement Frontend.
  - **Responsabilités :** Implémentation de l'interface utilisateur et intégration des fonctionnalités.
- **Astou DIOUF :**
  - **Rôle :** Développement Backend.
  - **Responsabilités :** Mise en place de la logique serveur et gestion des API.
- **Mbaye SYLLA :**
  - **Rôle :** Base de données et tests.
  - **Responsabilités :** Conception de la base de données et réalisation des tests fonctionnels.

## **2. Planning de réalisation**

Le projet **Gestion des Parrainages de Candidature d'Élections Présidentielles** a été réalisé sur une période de cinq semaines, débutant le 27 janvier 2025 et se terminant le 3 mars 2025. Le planning a été élaboré en plusieurs phases clés :

### *Semaine 1 (27 janvier - 2 février)*

- **Conception du système** : Élaboration des **diagrammes de classe** pour structurer les classes, attributs, méthodes et relations, création des **diagrammes de cas d'utilisation** pour identifier les interactions entre les utilisateurs et le système, développement du **Modèle Logique de Données** représentant les entités, leurs attributs et les relations, élaboration du **diagramme Entité-Association** pour visualiser les entités, attributs et relations sous forme d'associations, définition du **schéma de base de données** pour préparer la création physique des tables, colonnes, types de données et relations.

### *Semaine 2 (3 février - 9 février)*

- **Choix des technologies** : Sélection des frameworks et bibliothèques pour le développement frontend et backend, choix de la base de données, décision sur les outils de gestion de version et de gestion de projet .
- **Mise en place de l'environnement de développement** : Création d'un **dépôt GitHub** pour héberger le code du projet, configuration de **Jira** pour la gestion des tâches.
- **Développement Frontend et Backend** : Début des travaux de développement en utilisant les technologies sélectionnées.

### *Semaine 3 (10 février - 16 février)*

- **Base de données** : Conception et implémentation de la base de données, intégration avec le backend.
- **Développement** : Création des routes pour le backend, développement des API pour la communication entre le frontend et le backend, intégration des fonctionnalités du frontend avec les API développées, mise en place de la logique métier et des services nécessaires au bon fonctionnement de l'application.

### *Semaine 4 (17 février - 23 février)*

- **Développement** : Poursuite du développement des fonctionnalités, tests unitaires pour assurer la qualité du code, intégration continue pour faciliter le déploiement régulier des mises à jour.

### *Semaine 5 (24 février - 3 mars)*

- **Déploiement** : Préparation et mise en ligne de l'application.
- **PowerPoint** : Création d'une présentation PowerPoint pour présenter le projet et ses résultats.

### **3. Stratégie de développement**

**Méthode adoptée** : Nous avons choisi d'adopter la méthode **Agile** pour le développement de notre projet, ce qui nous a permis de bénéficier d'une grande flexibilité et d'une adaptation rapide aux changements. En suivant les principes Agile, nous avons pu itérer sur notre produit par petites étapes, en nous concentrant sur des fonctionnalités fonctionnelles à chaque cycle de développement. Nous avons également instauré une routine de commits chaque dimanche, garantissant ainsi que toutes les modifications apportées au code soient intégrées régulièrement. Cela a favorisé une meilleure collaboration entre les membres de l'équipe et a permis de détecter rapidement les problèmes potentiels, facilitant la mise en place de solutions avant qu'ils ne deviennent critiques.

**Outils de gestion de projet utilisés** : Pour la gestion de projet, nous avons utilisé **Jira**, un outil puissant qui nous a permis de planifier nos sprints, de suivre l'avancement des tâches et de gérer les priorités de manière efficace. Grâce à Jira, nous avons pu créer des tableaux Kanban pour visualiser le flux de travail, assigner des tâches aux membres de l'équipe et établir des échéances claires. Les réunions régulières de suivi ont été facilitées par la possibilité de consulter les tableaux de bord, ce qui a renforcé notre communication et notre coordination.

En complément, nous avons également utilisé **Google Meet** pour nos réunions virtuelles. Cet outil nous a permis de maintenir des échanges en temps réel, même à distance, et de discuter des avancées et des obstacles rencontrés. Les visioconférences via Google Meet ont été essentielles pour maintenir une bonne dynamique d'équipe et assurer que tous les membres soient alignés sur les objectifs du projet.

## 4. Architecture technique

- Description de l'architecture technique :

- **Architecture générale** : L'architecture de notre application est basée sur une approche en couches, séparant clairement le frontend et le backend. Cela permet une modularité et une maintenabilité du code, tout en facilitant les échanges entre les différentes parties du système.

- **Choix technologiques** :

- **Front-End** :

- **Angular** : Utilisé pour le développement de l'interface BackOFFICE destinée au personnel de la Direction Générale des Élections (DGE). Angular permet de créer une application web dynamique et réactive, facilitant la gestion des parrainages et l'importation de la liste des électeurs. Pour les interfaces FrontOFFICE destinées aux électeurs et aux candidats, des frameworks comme **React Native** ou **Ionic** peuvent être envisagés pour assurer une expérience utilisateur fluide sur mobile.

- **Back-End** :

- **Laravel** : Choisi comme framework PHP pour le développement du backend. Laravel facilite la création d'applications web sécurisées grâce à ses fonctionnalités intégrées, telles que l'authentification, la gestion des sessions et la validation des données. Ce framework permet également de structurer le code de manière modulaire et maintenable.
    - **API REST** : Laravel sera utilisé pour développer des API RESTful, offrant ainsi une communication fluide entre le frontend et le backend. Ces API gèreront les différentes fonctionnalités nécessaires à l'application.

- Base de données :

- **MySQL** : Une base de données relationnelle sera utilisée pour stocker les informations des électeurs, des candidats et des parrainages. MySQL a été choisie pour sa fiabilité, sa performance et sa capacité à gérer des transactions complexes.

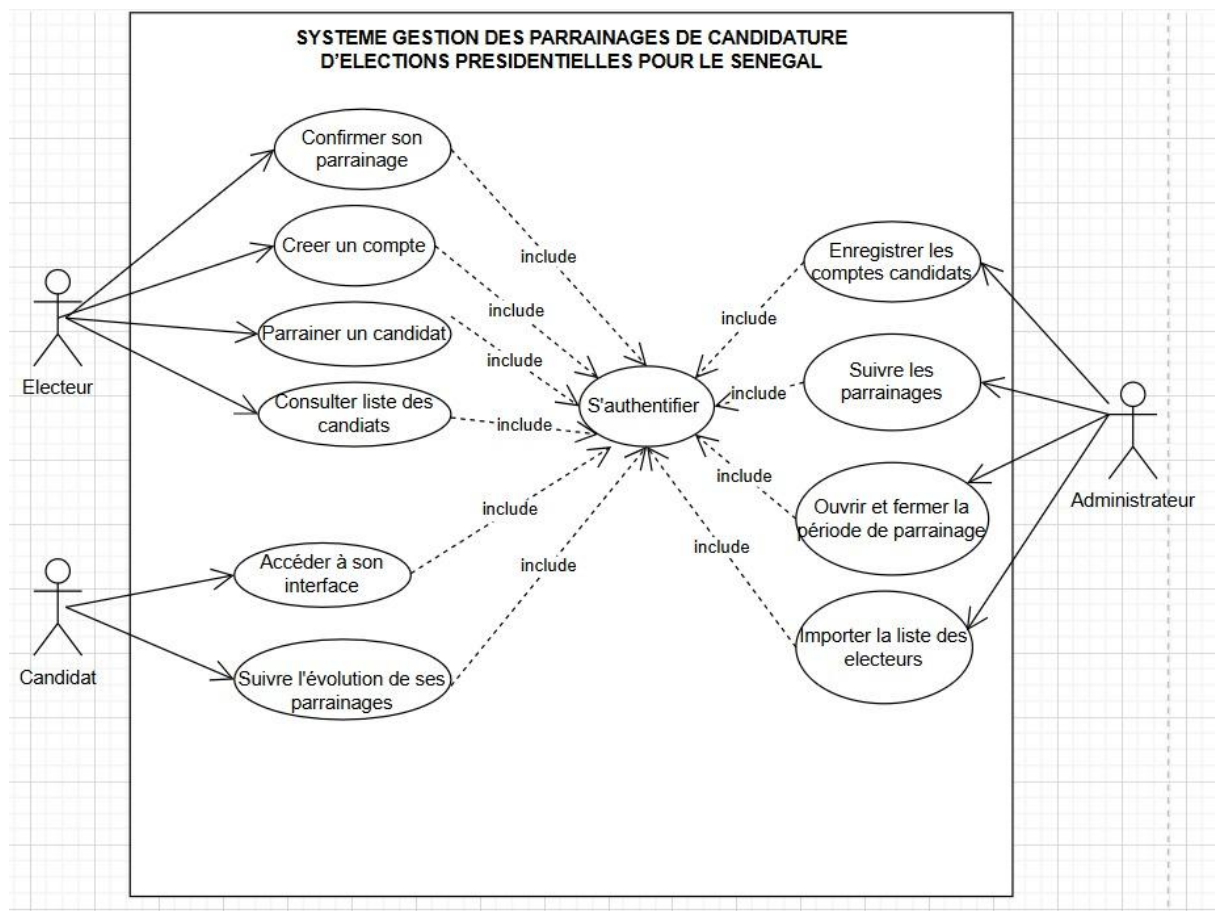
- API :

- **Services REST** : Des API RESTful seront développées pour faciliter l'interaction entre le frontend et le backend. Ces services incluront :
  - **API d'importation** : Pour charger la liste des électeurs et valider leur contenu.
  - **API de parrainage** : Pour enregistrer et suivre les parrainages des candidats.
  - **API de gestion des candidats** : Pour saisir et contrôler les informations des candidats demandant un parrainage.
  - **API d'authentification** : Pour gérer les comptes des électeurs et l'envoi de codes d'authentification par email et SMS.

## 5. Modèles conceptuels et données

Dans le cadre du projet de gestion des parrainages d'élections présidentielles pour le Sénégal, plusieurs diagrammes UML ont été élaborés pour représenter les différentes facettes du système. Ces diagrammes permettent de visualiser les interactions entre les acteurs, les classes et les données, facilitant ainsi la compréhension du fonctionnement et des exigences du système.

### Diagramme de cas d'utilisation (Use Case Diagram)



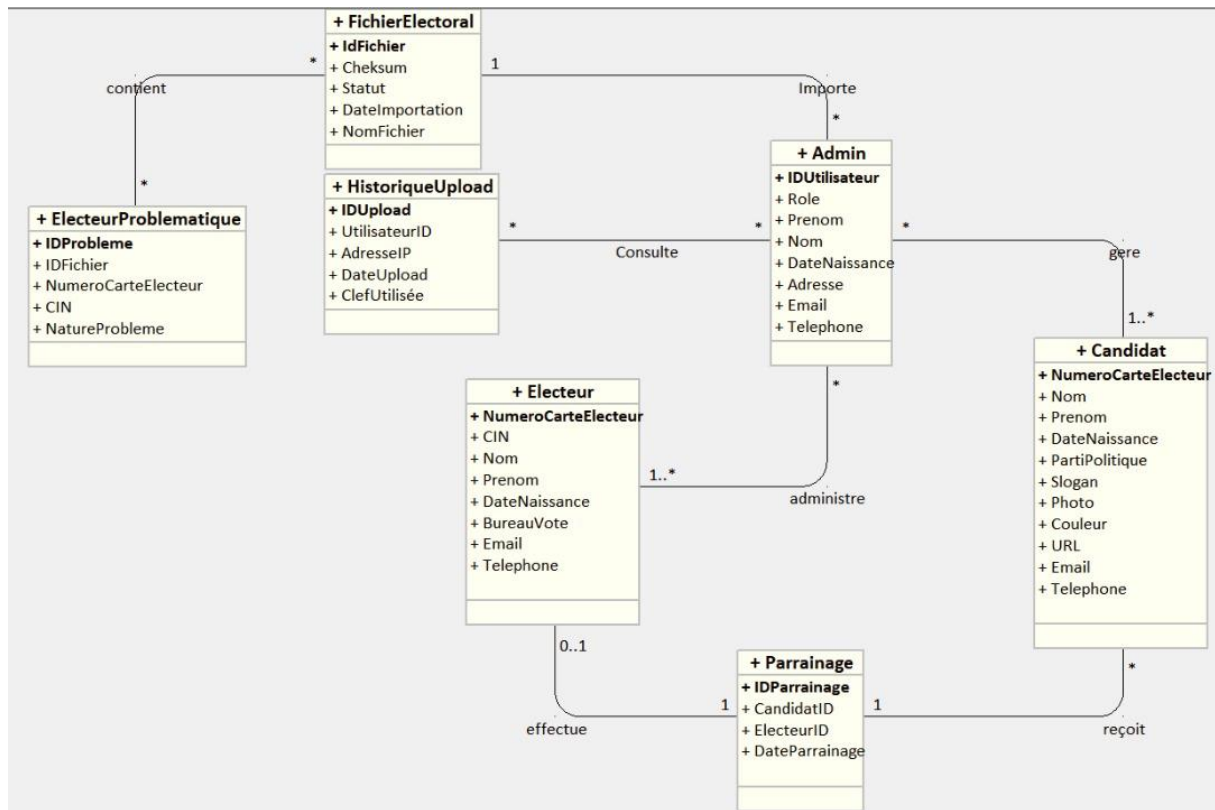
Le diagramme de cas d'utilisation illustre les interactions entre les différents acteurs du système, notamment les électeurs, les candidats et les administrateurs. Chaque acteur est associé à des cas d'utilisation spécifiques, représentant les fonctionnalités offertes par le système. Ce diagramme aide à identifier les besoins fonctionnels et à clarifier les attentes des utilisateurs :

- **Acteurs :**

- **Électeur** : Peut confirmer son parrainage, créer un compte, parrainer un candidat, et consulter la liste des candidats.
- **Candidat** : Accède à son interface, suit l'évolution de ses parrainages, et se fait enregistrer.
- **Administrateur** : Gère les comptes candidats, suit les listes de parrainage, ouvre et ferme les parrainages, et importe la liste des électeurs.



## Diagramme de classes (Class Diagram)



Le diagramme de classes représente la structure statique du système de gestion des parrainages. Il illustre les classes principales, leurs attributs, leurs méthodes, ainsi que les relations entre elles. Ce diagramme est essentiel pour comprendre comment les données sont structurées et interconnectées dans le système.

- **Classes principales :**

- **Electeur :**

- **Attributs :**

- ID : Identifiant unique de l'électeur.
      - NumeroCarteElecteur : Numéro de la carte d'électeur.
      - CIN : Carte d'identité nationale de l'électeur.
      - Nom : Nom de l'électeur.
      - Prenom : Prénom de l'électeur.
      - Email : Adresse email de l'électeur.
      - Telephone : Numéro de téléphone de l'électeur.

- **Méthodes :**

- ConfirmerParrainage() : Méthode pour confirmer un parrainage.

- **Candidat :**

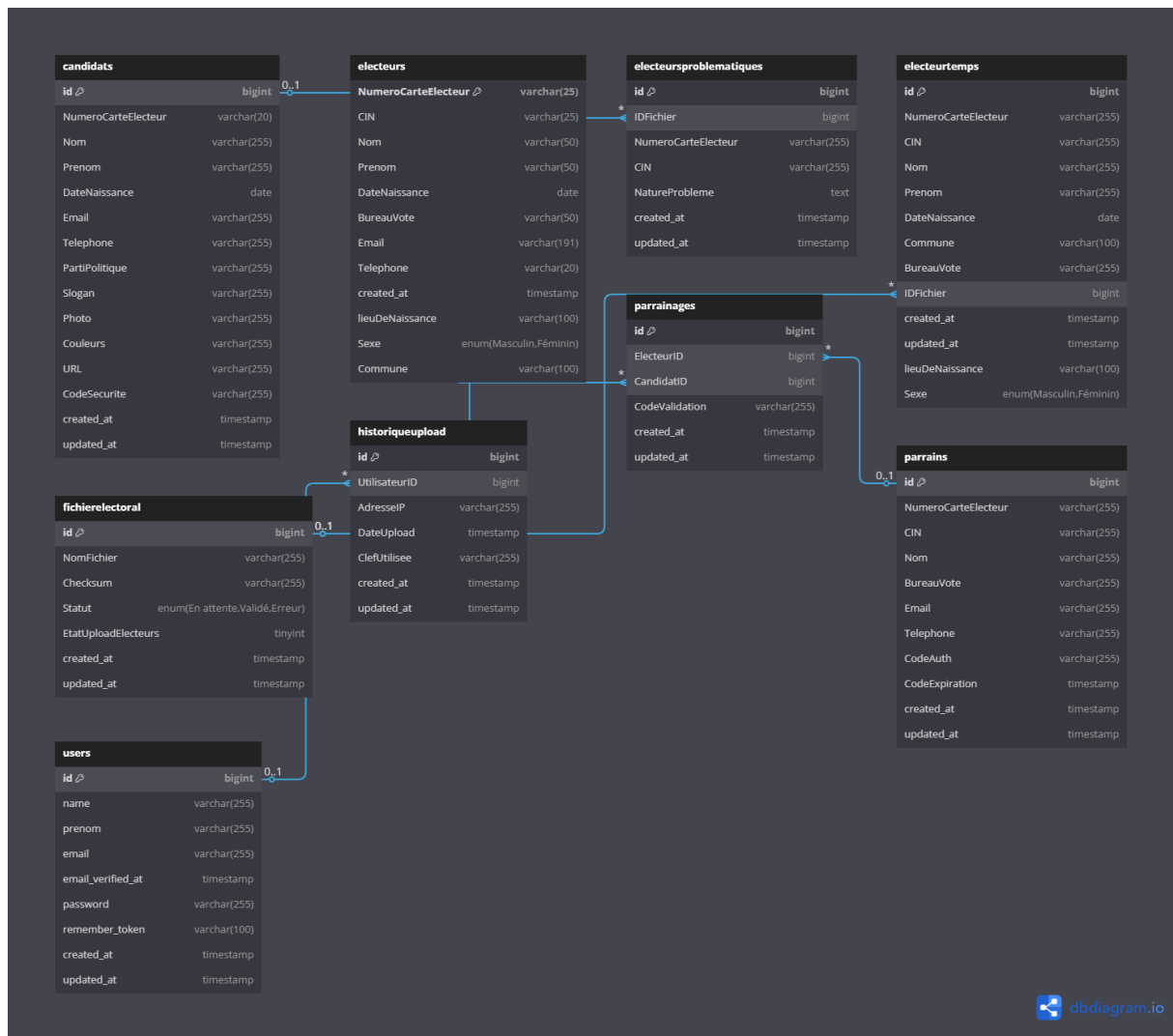
- **Attributs :**

- NumeroCarteElecteur : Numéro de la carte d'électeur du candidat.
      - DateNaissance : Date de naissance du candidat.
      - PartiPolitique : Parti politique auquel le candidat appartient.
      - Nom : Nom du candidat.
      - Prenom : Prénom du candidat.

- **Méthodes :**
  - SuivreEvolutionParrainage() : Permet au candidat de suivre l'évolution de ses parrainages.
- **Parrainage :**
  - **Attributs :**
    - IDParrainage : Identifiant unique du parrainage.
    - DateParrainage : Date à laquelle le parrainage a été effectué.
  - **Relations :**
    - Association avec Électeur (un électeur peut parrainer plusieurs candidats).
    - Association avec Candidat (un candidat peut être parrainé par plusieurs électeurs).
- **HistoriqueUpload :**
  - **Attributs :**
    - IDFichier : Identifiant unique du fichier.
    - NomFichier : Nom du fichier importé.
    - DateUpload : Date à laquelle le fichier a été téléchargé.
    - UtilisateurUpload : Identifiant de l'utilisateur qui a effectué l'upload.
    - NatureProbleme : Description de tout problème associé au fichier.
  - **Méthodes :**
    - ConsulterHistorique() : Permet de consulter les fichiers téléchargés.
- **Admin :**
  - **Attributs :**
    - IDAdmin : Identifiant unique de l'administrateur.
    - Nom : Nom de l'administrateur.
    - Prenom : Prénom de l'administrateur.
    - Email : Adresse email de l'administrateur.
  - **Méthodes :**
    - GererComptesCandidats() : Permet à l'administrateur de gérer les comptes des candidats.
- **Relations :** Le diagramme montre des relations d'association entre les classes, illustrant comment les électeurs, les candidats et les parrainages interagissent. Chaque parrainage relie un électeur à un candidat, et chaque historique d'upload est lié à un utilisateur spécifique.

### Schéma de la base de données

Le schéma de la base de données est conçu pour gérer les informations relatives aux électeurs, candidats, parrainages, utilisateurs, ainsi que les fichiers électoraux. Il se compose de plusieurs tables interconnectées.



## ➤ Tables et descriptions

### • candidats

- **id** : Identifiant unique du candidat (clé primaire).
- **NumeroCarteElecteur** : Numéro de la carte électeur du candidat, unique.
- **Nom** : Nom du candidat.
- **Prenom** : Prénom du candidat.
- **DateNaissance** : Date de naissance du candidat.
- **Email** : Adresse email du candidat.
- **Telephone** : Numéro de téléphone du candidat.
- **PartiPolitique** : Parti politique du candidat.
- **Slogan** : Slogan associé au candidat.
- **Photo** : URL de la photo du candidat.
- **Couleurs** : Couleurs associées au candidat.
- **URL** : Lien vers le site web du candidat.
- **CodeSecurite** : Code de sécurité pour le candidat.
- **created\_at** / **updated\_at** : Dates de création et de mise à jour de l'enregistrement.

### • electeurs

- **NumeroCarteElecteur** : Identifiant unique de l'électeur (clé primaire).

- **CIN** : Carte d'identité nationale, unique.
- **Nom** : Nom de l'électeur.
- **Prenom** : Prénom de l'électeur.
- **DateNaissance** : Date de naissance de l'électeur.
- **BureauVote** : Bureau de vote assigné à l'électeur.
- **Email** : Adresse email de l'électeur.
- **Telephone** : Numéro de téléphone de l'électeur.
- **Commune** : Commune de résidence de l'électeur.
- **created\_at** : Date de création de l'enregistrement.
- **lieuDeNaissance** : Lieu de naissance de l'électeur.
- **Sexe** : Sexe de l'électeur (Masculin, Féminin).
- **electeursproblematicques**
  - **id** : Identifiant unique du problème (clé primaire).
  - **IDFichier** : Référence au fichier électoral (clé étrangère vers fichierelectoral.id).
  - **NumeroCarteElecteur** : Numéro de carte électeur associé au problème.
  - **CIN** : Carte d'identité nationale associée au problème.
  - **NatureProbleme** : Description de la nature du problème.
  - **created\_at** / **updated\_at** : Dates de création et de mise à jour.
- **electeurtemps**
  - **id** : Identifiant unique de l'électeur temporaire (clé primaire).
  - **NumeroCarteElecteur** : Numéro de carte électeur, unique.
  - **CIN** : Carte d'identité nationale, unique.
  - **Nom** : Nom de l'électeur temporaire.
  - **Prenom** : Prénom de l'électeur temporaire.
  - **DateNaissance** : Date de naissance de l'électeur temporaire.
  - **Commune** : Commune de résidence.
  - **BureauVote** : Bureau de vote assigné.
  - **IDFichier** : Référence au fichier électoral (clé étrangère vers fichierelectoral.id).
  - **created\_at** / **updated\_at** : Dates de création et de mise à jour.
  - **lieuDeNaissance** : Lieu de naissance de l'électeur temporaire.
  - **Sexe** : Sexe de l'électeur temporaire (Masculin, Féminin).
- **fichierelectoral**
  - **id** : Identifiant unique du fichier électoral (clé primaire).
  - **NomFichier** : Nom attribué au fichier.
  - **Checksum** : Vérification d'intégrité du fichier.
  - **Statut** : État du fichier (ex. En attente, Validé, Erreur).
  - **EtatUploadElecteurs** : État de l'upload des électeurs.
  - **created\_at** / **updated\_at** : Dates de création et de mise à jour.
- **historiqueupload**
  - **id** : Identifiant unique de l'historique d'upload (clé primaire).
  - **UtilisateurID** : Référence à l'utilisateur ayant effectué l'upload (clé étrangère vers users.id).
  - **AdresseIP** : Adresse IP de l'utilisateur lors de l'upload.
  - **DateUpload** : Date de l'upload.
  - **ClefUtilisee** : Clé utilisée pour l'upload.
  - **created\_at** / **updated\_at** : Dates de création et de mise à jour.
- **parrainages**
  - **id** : Identifiant unique du parrainage (clé primaire).
  - **ElecteurID** : Référence à l'identifiant du parrain (clé étrangère vers parrains.id).

- **CandidatID** : Référence à l'identifiant du candidat (clé étrangère vers candidats.id).
- **CodeValidation** : Code pour valider le parrainage.
- **created\_at / updated\_at** : Dates de création et de mise à jour.
- **parrains**
  - **id** : Identifiant unique du parrain (clé primaire).
  - **NumeroCarteElecteur** : Numéro de carte électeur du parrain, unique.
  - **CIN** : Carte d'identité nationale du parrain, unique.
  - **Nom** : Nom du parrain.
  - **BureauVote** : Bureau de vote assigné au parrain.
  - **Email** : Adresse email du parrain, unique.
  - **Telephone** : Numéro de téléphone du parrain, unique.
  - **CodeAuth** : Code d'authentification du parrain.
  - **CodeExpiration** : Date d'expiration du code d'authentification.
  - **created\_at / updated\_at** : Dates de création et de mise à jour.
- **users**
  - **id** : Identifiant unique de l'utilisateur (clé primaire).
  - **name** : Nom de l'utilisateur.
  - **prenom** : Prénom de l'utilisateur.
  - **email** : Adresse email de l'utilisateur, unique.
  - **email\_verified\_at** : Date de vérification de l'email.
  - **password** : Mot de passe de l'utilisateur.
  - **remember\_token** : Token de session pour l'utilisateur.
  - **created\_at / updated\_at** : Dates de création et de mise à jour.

#### ➤ Relations entre les tables

- **candidats et parrainages** :
  - Relation : Un candidat peut avoir plusieurs parrainages (1 à N).
  - Clé étrangère : CandidatID dans la table parrainages référence id dans candidats.
- **parrains et parrainages** :
  - Relation : Un parrain peut parrainer plusieurs candidats (1 à N).
  - Clé étrangère : ElecteurID dans la table parrainages référence id dans parrains.
- **users et historiqueupload** :
  - Relation : Un utilisateur peut avoir plusieurs uploads (1 à N).
  - Clé étrangère : UtilisateurID dans la table historiqueupload référence id dans users.
- **fichierelectoral et historiqueupload** :
  - Relation : Un fichier électoral peut être associé à plusieurs historiques d'upload (1 à N).
  - Clé étrangère : IDFichier dans la table historiqueupload référence id dans fichierelectoral.
- **fichierelectoral et electeursproblematicques** :
  - Relation : Un fichier électoral peut contenir plusieurs problèmes d'électeurs (1 à N).
  - Clé étrangère : IDFichier dans la table electeursproblematicques référence id dans fichierelectoral.
- **fichierelectoral et electeurtemps** :

- Relation : Un fichier électoral peut être lié à plusieurs électeurs temporaires (1 à N).
- Clé étrangère : IDFichier dans la table electeurtemps référence id dans fichierelectoral.

## **6. Fonctionnalités du système**

Cette section décrit les fonctionnalités principales identifiées dans le système, ainsi que des captures d'écran illustratives pour chaque fonctionnalité. Ces fonctionnalités sont conçues pour répondre aux besoins des utilisateurs et améliorer l'efficacité du processus électoral.

### **a. Liste des fonctionnalités identifiées**

#### ➤ Gestion des candidats

- **Ajout de candidats** : Permet aux administrateurs d'ajouter de nouveaux candidats au système avec toutes leurs informations pertinentes (nom, prénom, parti politique, etc.).
- **Modification de candidats** : Fonctionnalité permettant de mettre à jour les informations d'un candidat existant.
- **Suppression de candidats** : Option pour supprimer un candidat du système.
- **Consultation de la liste des candidats** : Interface pour visualiser tous les candidats enregistrés.

#### ➤ Gestion des électeurs

- **Enregistrement des électeurs** : Permet aux administrateurs d'ajouter de nouveaux électeurs avec toutes leurs informations (nom, prénom, bureau de vote, etc.).
- **Mise à jour des informations des électeurs** : Fonctionnalité pour modifier les détails des électeurs existants.
- **Suppression des électeurs** : Option pour retirer un électeur du système.
- **Liste des électeurs** : Interface pour visualiser tous les électeurs inscrits.

#### ➤ Parrainages

- **Création de parrainages** : Permet aux électeurs de parrainer des candidats.
- **Validation des parrainages** : Fonctionnalité pour vérifier et approuver les parrainages.
- **Consultation des parrainages** : Interface affichant tous les parrainages effectués.

#### ➤ Gestion des fichiers électoraux

- **Téléchargement de fichiers électoraux** : Permet aux utilisateurs de télécharger des fichiers contenant des informations électorales.
- **Vérification de l'intégrité des fichiers** : Système de contrôle pour assurer que les fichiers téléchargés sont valides.
- **Consultation de l'historique des fichiers** : Interface pour visualiser les fichiers électoraux précédemment téléchargés.

#### ➤ Gestion des utilisateurs

- **Création d'utilisateurs** : Permet à un administrateur de créer de nouveaux comptes utilisateurs pour l'accès au système.
- **Gestion des rôles et permissions** : Fonctionnalité pour définir les rôles des utilisateurs et leurs droits d'accès au système.

- **Historique des actions des utilisateurs** : Interface pour consulter les actions réalisées par chaque utilisateur.

### **b. Captures d'écran de quelques fonctionnalités**

- **Capture d'écran** : Interface d'ajout de candidats.

The screenshot shows the 'Enregistrement d'un Candidat' (Candidate Registration) form. The form is part of the 'Système Electoral' application, with navigation links for 'Électeurs', 'Candidats', and 'Calendrier'. A 'Déconnexion' link is in the top right. The form contains the following fields:

- Nom: Ba
- Prénom: Amadou
- Date de naissance: 1970-03-30
- Email
- Téléphone
- Parti Politique
- Slogan
- URL de la Photo
- Couleurs du Parti (ex: #FF0000, #00FF00, #0000FF)
- URL d'Informations

A green button labeled 'Enregistrer le Candidat' is at the bottom left of the form.

- **Capture d'écran** : Liste des électeurs problématiques.

The screenshot shows the 'Électeurs Problématiques' (Problematic Electors) table. The table is part of the 'Système Electoral' application, with navigation links for 'Électeurs', 'Candidats', and 'Calendrier'. A 'Déconnexion' link is in the top right. The table has the following columns:

CIN	N° ELECTEUR	TYPE D'ERREUR	ACTIONS
XY770001	770001122	CIN invalide (doit contenir 13 ou 14 chiffres)	<a href="#">Corriger</a> <a href="#">Supprimer</a>
XY770002	770002233	CIN invalide (doit contenir 13 ou 14 chiffres); Nom ou prénom contient des accents	<a href="#">Corriger</a> <a href="#">Supprimer</a>
XY770003	770003344	CIN invalide (doit contenir 13 ou 14 chiffres)	<a href="#">Corriger</a> <a href="#">Supprimer</a>
XY770004	770004455	CIN invalide (doit contenir 13 ou 14 chiffres)	<a href="#">Corriger</a> <a href="#">Supprimer</a>
XY770005	770005566	CIN invalide (doit contenir 13 ou 14 chiffres)	<a href="#">Corriger</a> <a href="#">Supprimer</a>
XY770006	770006677	CIN invalide (doit contenir 13 ou 14 chiffres)	<a href="#">Corriger</a> <a href="#">Supprimer</a>
XY770007	770007788	CIN invalide (doit contenir 13 ou 14 chiffres); Nom ou prénom contient des accents	<a href="#">Corriger</a> <a href="#">Supprimer</a>
XY770008	770008899	CIN invalide (doit contenir 13 ou 14 chiffres); Nom ou prénom contient des accents	<a href="#">Corriger</a> <a href="#">Supprimer</a>
XY770009	770009900	CIN invalide (doit contenir 13 ou 14 chiffres)	<a href="#">Corriger</a> <a href="#">Supprimer</a>
XY770010	770010011	CIN invalide (doit contenir 13 ou 14 chiffres); Nom ou prénom contient des accents	<a href="#">Corriger</a> <a href="#">Supprimer</a>
JK111223	111223344	CIN invalide (doit contenir 13 ou 14 chiffres)	<a href="#">Corriger</a> <a href="#">Supprimer</a>
KL222334	222334455	CIN invalide (doit contenir 13 ou 14 chiffres)	<a href="#">Corriger</a> <a href="#">Supprimer</a>

- **Capture d'écran** : Interface de création de la période de parrainages.



Système Electoral   Électeurs   Candidats   Calendrier   Déconnexion

### Gestion des Périodes de Parrainage

Date de Début  
02/10/2025

Date de Fin  
02/01/2026

Enregistrer les dates

- **Capture d'écran** : Interface de téléchargement de fichiers.

Système Electoral   Électeurs   Candidats   Calendrier   Déconnexion

### Gestion des Électeurs

#### Chargement du Fichier

Empreinte CHECKSUM (SHA256)

096475d899a0c5a9b03b10d8f47ab887c3c02164bca96a3a920fd1be34d7cf4c

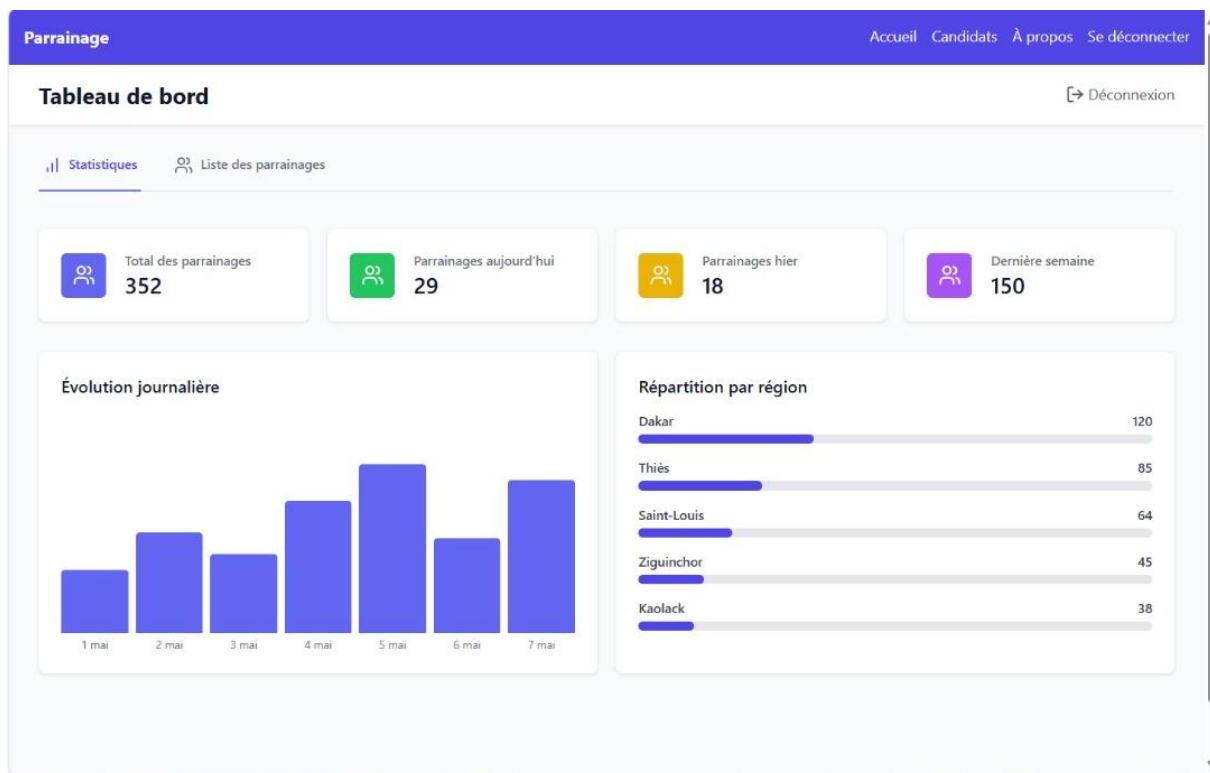
Fichier CSV des électeurs

Sélectionner un fichier ou glisser déposer  
CSV uniquement

Fichier sélectionné: electeur2.csv

Charger le fichier

- **Capture d'écran** : Tableau de bord avec statistiques clés.



- **Capture d'écran** : Liste des parrainages.

**Parrainage** Accueil Candidats À propos Se déconnecter

### Tableau de bord

Statistiques Liste des parrainages

#### Liste des parrainages

Détails des électeurs qui vous ont parrainé

NOM & PRÉNOM	RÉGION	BUREAU DE VOTE	DATE
Diop Mamadou	Dakar	École Primaire Dakar 1	07/05/2023
Ndiaye Fatou	Thiès	Lycée de Thiès	07/05/2023
Sow Abdoulaye	Saint-Louis	Centre Culturel	06/05/2023
Fall Aïssatou	Dakar	École Primaire Dakar 2	06/05/2023
Mbaye Ibrahima	Kaolack	Mairie de Kaolack	05/05/2023
Diallo Mariama	Ziguinchor	Centre de vote Ziguinchor	05/05/2023
Gueye Ousmane	Dakar	École Primaire Dakar 3	04/05/2023
Sarr Aminata	Thiès	Centre de vote Thiès 2	04/05/2023
Fave Moussa	Saint-Louis	École Saint-Louis	03/05/2023

- **Capture d'écran** : Validation de parrainages.



## Validation du parrainage



**Dia Aliou Mamadou**

Les Valeurs aux Services de la Nation !

Code de Validation (5 chiffres)

Saisissez le code à 5 chiffres reçu par email et SMS

[Je n'ai pas reçu de code](#)

[Valider mon parrainage](#)

## **Conclusion**

Ce rapport technique a présenté une vue d'ensemble détaillée du projet, en mettant en lumière l'organisation de l'équipe, le planning de réalisation, la stratégie de développement, l'architecture technique, ainsi que les modèles conceptuels et les données.

Les fonctionnalités du système ont été soigneusement élaborées pour répondre aux besoins des utilisateurs tout en assurant une gestion efficace des informations électorales. Grâce à une structure de base de données bien conçue, le système permet de gérer les candidats, les électeurs, les parrainages et les fichiers électoraux de manière cohérente et sécurisée.

En conclusion, ce projet représente une avancée significative dans la digitalisation des processus électoraux, offrant des outils adaptés pour faciliter la transparence et l'efficacité du système électoral. Les leçons tirées de ce projet serviront de fondement pour de futures améliorations et développements, visant toujours à optimiser l'expérience utilisateur et à garantir l'intégrité des élections.