

Cahier des charges

1. Contexte et présentation du projet	2
2. Périmètre du projet	3
a. Fonctionnalités incluses	3
i. Gestion des utilisateurs et rôles	3
ii. Gestion des élections	3
iii. Paramétrage	4
iv. Contrôle des électeurs	4
v. Gestion des alertes	4
vi. Gestion des résultats	5
vii. Etat d'analyse et statistiques	6
b. Fonctionnalités hors périmètre	6
3. Contrainte technique	6
a. Environnement de développement	6
i. Technologies front-end mobile	6
ii. Technologies front-end back-office	6
iii. Technologie back-end	6
b. Compatibilité et accessibilité	6
i. Compatibilité mobile	6
ii. Fonctionnalité en réseau-local	7
c. Déploiement	7
4. Sécurité et gestion des données	7
a. Authentification	7
b. Accès aux données via l'application mobile	7
5. Test et validation	8
a. Types de tests	8
i. Tests unitaires	8
ii. Tests d'intégration	8
iii. Tests de sécurité	8
iv. Tests de performance	8
b. Validation des résultats	8
i. Validation fonctionnelle	8
ii. Validation utilisateur	8
6. Annexes	9

1. Contexte et présentation du projet

La Commission Électorale Nationale Indépendante (CENI) de Madagascar a entrepris une refonte de ses systèmes de gestion électorale afin de moderniser et d'optimiser les opérations liées à l'organisation des élections. Actuellement, la gestion des processus électoraux est confrontée à de nombreux défis, notamment la complexité des opérations, la multiplicité des acteurs, et les contraintes technologiques adéquate aux situations locales.

Dans ce contexte, la CENI a lancé la mise en place d'une nouvelle plateforme de gestion électoral (application mobile et système back-office). Cette solution a pour but non seulement de centraliser et de sécuriser la gestion des élections, mais également de répondre aux contraintes locales grâce à l'implémentation d'une fonctionnalité en réseau local pour l'application mobile.

2. Périmètre du projet

a. Fonctionnalités incluses

Dans sa globalité, le système peut gérer les élections:

- **Présidentielle**
- **Législative**
- **Communale**

Différents types d'acteur interagissent avec le système:

- **Administrateur**: contrôle total sur le système, pouvant accéder à toutes les ressources.
- **Chef CID (Centre Informatique de District)**: responsable d'envoi des résultats d'élection.
- **Responsable**: enregistrement des électeurs durant l'élection au sein de chaque bureau de vote..
- **Opérateur**: vérification et validation des données récoltées après l'élection.

i. Gestion des utilisateurs et rôles

Cette fonctionnalité est uniquement dans l'application back-office, et seulement les utilisateurs ayant le rôle d'administrateur peuvent y accéder.

Les administrateurs doivent être en mesure de:

- filtrer et voir la liste de tous les utilisateurs, ainsi que le nombre exacte de chaque type d'utilisateur
- créer d'autre utilisateur, modifier ou aussi supprimer un utilisateur:
En effet, il n'y aura pas d'interface de création de compte
- importer une liste d'utilisateur (comme par exemple la liste de tous les chefs CID (Centre Informatique de District))

ii. Gestion des élections

Comme dans la section précédente, seuls les administrateurs auront la possibilité d'utiliser cette fonctionnalité du back-office.

Cette partie permettra aux administrateurs:

- de filtrer et de consulter les historiques d'élection
- de générer les résultats provisoires (sous format pdf)
- de paramétrer l'application selon le type d'élection (qui intégrera les systèmes déjà en place):

Cette étape est la base de tout le système puisque les candidats, électeurs et résultats y seront rattachés

iii. Paramétrage

Toujours dans le rôle d'administrateur, cette section servira de paramétrer et configurer le système dans l'application back-office, comme:

- affectation d'un responsable à un bureau de vote (possibilité d'importation de donnée)
- modification ou insertion de nouveau bureau de vote (on doit donc pouvoir consulter et filtrer les bureaux de vote par régions, district, commune et fokontany)
- possibilité de modifier et ajouter de nouveau région / district / commune / fokontany si besoin

Les circonscriptions électorales seront intégrés dès le début.

iv. Contrôle des électeurs

Cette fonctionnalité a été mise en place spécialement pour les responsables et uniquement sur l'application mobile.

L'application permettra au responsable de chaque bureau de vote:

- de consulter les électeurs inscrits dans le bureau de vote concerné
- de filtrer et d'enregistrer les électeurs inscrits (enregistrement daté)
- d'envoyer la liste des électeurs enregistrés

v. Gestion des alertes

Pour assurer la régularité et un bon suivi des incidents, cette fonctionnalité est nécessaire pour la partie back-office. Les alertes seront classés en fonction du niveau d'incidence.

Une page pour consulter et filtrer les alertes sera accessible pour tout utilisateur, seul les administrateurs peuvent en revanche modifier les états de ces alertes (en attente / en cours de résolution / résolue).

Le système doit être automatiquement en mesure d'alerter:

- les valeurs extrêmes:
Exemple : Dans une région avec une participation électorale habituelle autour de 60%, un bureau de vote rapporte une participation de 95%.
Le degré de sensibilité de la détection des résultats anormaux devrait être configurable.
- les patterns temporels:
Exemple : Un pic soudain de votes est enregistré en dehors des heures d'ouverture des bureaux de vote.
- les incohérences des données:
Exemple : Les résultats électoraux d'un bureau de vote montrent un total de 700 votes, mais les listes électorales ne comptent que 600 électeurs.

vi. Gestion des résultats

La gestion des résultats est le coeur même du projet, elle se déroule après l'enregistrement des votes des électeurs.

Un modèle de PV sera utilisé pour le résultat ainsi que pour le scan. Dans la back-office, les administrateurs doivent pouvoir:

- générer un ou tous les QR codes nécessaires pour les PV de chaque bureau de vote.
Le QR contiendra les informations concernant le bureau de vote ainsi qu'un identifiant unique pour vérifier l'authenticité du document. Le PV sera automatiquement généré avec le code QR sur le modèle du document existant.

Sur l'application mobile, une demande d'authentification sera nécessaire pour les chefs CID, pour pouvoir:

- scanner et saisir les pv de résultats
- envoyer les données récoltées

Après la réception des données, les opérateurs dans l'application back-office devront:

- vérifier et valider les résultats importés (document scanné vs saisie)

Pour finir, seuls les administrateurs pourront dans la back-office:

- exporter les résultats provisoires (sous format pdf)
Les résultats dépendront du type d'élection, respectivement:
 - **présidentielle**: par bureau de vote, par fokontany, par commune, par district, par région et par province
 - **législative**: par bureau de vote, par fokontany, par commune et par district
 - **communale**: par bureau de vote, par fokontany et par commune

vii. Etat d'analyse et statistiques

Cette partie est uniquement dans la back-office et pour les administrateurs. Une page devrait être mise en place pour:

- consulter les résultats statistiques (par région, district, etc...)
- visualiser les taux de participation par zone géographique et par type d'élection

b. Fonctionnalités hors périmètre

La CENI a déjà un système fonctionnel pour certain besoin dans leur gestion électoral.

Le nouveau système intégrera les données essentiels à son fonctionnement en ce qui concerne:

- la liste des candidats et les partis politiques à une élection
- la consultation des électeurs inscrits à une élection

3. Contrainte technique

a. Environnement de développement

i. Technologies front-end mobile

L'application mobile sera développée avec Flutter et une base SQLite intégré. Le choix de Flutter se repose sur sa vitesse de développement et sa capacité à offrir une expérience utilisateur native. Le même code base permettrait aussi de générer la même application sur d'autre plateforme.

ii. Technologies front-end back-office

Le framework Angular avec Typescript sera utilisé pour le front-end du back-office. Il est idéal pour les interfaces utilisateurs complexes et la gestion efficace des composants. Sa structure de nature promets une bonne scalabilité et lisibilité du code.

iii. Technologie back-end

Le serveur de traitement sera basé sur Spring, utilisant Spring Boot pour accélérer le développement. Un framework Java qui permet de créer des applications d'entreprise performantes et sécurisées. Spring est préféré pour son évolutivité et sa capacité à gérer des API REST complexes, ainsi que pour ses outils intégrés de gestion des transactions et de sécurisation des données. Oracle sera utilisé comme base de donnée pour la majorité du système

b. Compatibilité et accessibilité

i. Compatibilité mobile

L'application mobile devra être compatible avec les versions Android minimum versions 7.0, assurant une couverture étendue des appareils mobiles.

ii. Fonctionnalité en réseau-local

L'application mobile devra fonctionner en réseau-local, permettant aux agents de terrain de continuer à utiliser l'application même dans les zones avec des connexions limitées. Une synchronisation sera effectuée dès que la connexion est rétablie.

c. Déploiement

Le back-office sera uniquement déployé sur le serveur local de CENI pour garantir la sécurité et le contrôle total des données électorales. Ce choix permet de protéger les données sensibles contre d'éventuelle intrusion et de s'assurer que les informations ne transitent pas par des serveurs externes. En effet, la CENI a déjà un système de pare-feu en place protégeant le réseau interne de toute attaque extérieur.

4. Sécurité et gestion des données

La communication dans tout le système se fera à travers des web services.

a. Authentification

L'authentification sera basé sur l'utilisation des jetons, précisément le JWT. Après la connexion, chaque utilisateur recevra deux (02) jetons pour la back-office, dont un jeton d'autorisation (pour le contrôle d'accès aux ressources) qui aura un délai de validité de 5 minutes, et un autre jeton de rafraîchissement (pour maintenir la session de l'utilisateur) qui aura un délai de validité de 24 heures. Les utilisateurs de l'application mobile n'auront qu'un seul jeton qui n'aura pas de délai d'expiration pour maintenir la haute disponibilité en réseau-local.

b. Accès aux données via l'application mobile

- Filtre des données

Les données seront filtrés par bureau de vote, afin que les utilisateurs n'aient accès qu'aux informations qui les concernent directement. Cela permettra aussi de limiter l'exploitation des données critiques et à l'optimisation de l'application.

Ces données concerneront généralement les électeurs inscrits au bureau de vote en question.

- Système de cache

Après la connexion, les données devront être enregistrés directement dans l'application elle-même pour assurer le fonctionnement en réseau-local.

5. Test et validation

Le processus de test et validation est essentiel pour garantir que l'application mobile et le système back-office fonctionnent conformément aux spécifications.

a. Types de tests

Les outils à utiliser pour le test sont:

- JUnit pour Spring
- Jasmine pour Angular
- Flutter Test pour Flutter

i. Tests unitaires

Chaque composant du code sera testé individuellement pour s'assurer qu'il fonctionne correctement de manière isolé

ii. Tests d'intégration

Après avoir vérifier que chaque composant fonctionne individuellement, ces tests permettront de s'assurer que les différentes parties du système fonctionnent correctement ensemble. Cela inclut la communication entre le mobile, le back-office, et la base de données

iii. Tests de sécurité

Comme la sécurité est un aspect critique de ce projet, des tests seront effectués pour s'assurer que l'authentification JWT et le filtre des données fonctionnent comme prévu.

iv. Tests de performance

Ces tests vérifieront que le système peut gérer un grand nombre de requête simultanées sans baisse de performance.

b. Validation des résultats

i. Validation fonctionnelle

Après chaque phase de tests, les résultats seront comparés aux exigences initiales pour s'assurer que le système fonctionne comme prévu

ii. Validation utilisateur

Les utilisateurs finaux participeront à des sessions de test pour valider que l'interface utilisateur et les processus métier répondent à leurs besoins.

6. Annexes

Architecture du système

